LOW DENSITY PARITY CHECK CODES BASED ON FINITE GEOMETRIES: A REDISCOVERY AND MORE

Yu Kou, Shu Lin
and
Marc Fossorier

October 20, 1999

Low Density Parity Check Codes Based on Finite Geometries: A Rediscovery and More*

Yu Kou, Shu Lin and Marc Fossorier
Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, HI 96822
Email: ykou, slin, marc@spectra.eng.hawaii.edu

Submitted to IEEE Transactions on Information Theory

October 12, 1999

Abstract

Low density parity check (LDPC) codes with iterative decoding based on belief propagation achieve astonishing error performance close to Shannon limit. No algebraic or geometric method for constructing these codes has been reported and they are largely generated by computer search. As a result, encoding of long LDPC codes is in general very complex. This paper presents two classes of high rate LDPC codes whose constructions are based on finite Euclidean and projective geometries, respectively. These classes of codes are cyclic and have good constraint parameters and minimum distances. Cyclic structure allows the use of linear feedback shift registers for encoding. These finite geometry LDPC codes achieve very good error performance with either soft-decision iterative decoding based on belief propagation or Gallager's hard-decision bit flipping algorithm. These codes can be punctured or extended to obtain other good LDPC codes. A generalization of these codes is also presented.

Key Words: Low density parity check codes, Euclidean geometry, projective geometry, puncturing, column splitting, iterative decoding, bit flipping decoding.

---

*This research was supported by NSF under Grants NCR 94-15374, CCR 97-32959, CCR 98-14054 and NASA under Grant NAG 5-8414.

1. Introduction

LDPC codes were first discovered by Gallager [1,2] in early sixties and have recently been rediscovered and generalized [3-11]. These codes with iterative decoding based on belief propagation achieve astonishing error performance close to Shannon limit [4,7,8]. A LDPC code is defined by a parity check matrix **H** with the following structural properties: (1) each row consists of ρ "ones"; (2) each column consists of γ "ones"; (3) the number of "ones" in common between any two columns, denoted λ , is no greater than 1; and (4) both ρ and γ are small compared to the length of the code. The code is simply the null space of **H**. Since ρ and γ are small, **H** has a small density of "ones". For this reason, the code specified by **H** is called a LDPC code. Although LDPC codes have been shown to achieve outstanding error performance, no analytic (algebraic or geometric) method has been found for constructing these codes. Codes that have been found are largely computer generated, especially long codes. Encoding of these long computer generated codes is very complex.

In this paper, we present two classes of high rate LDPC codes whose constructions are based on two-dimensional finite Euclidean and projective geometries, respectively. The parity check matrix **H** is a square matrix whose columns correspond to the points of a geometry. Each row of the matrix is the incidence vector of a line in the geometry. The number of "ones" in each row is equal to the number of points on a line. The number of "ones" in each column of **H** is equal to the number of lines that intersect at the point corresponding to the location of the column. Any two columns have exact one "1" in common, i.e., λ =1. This ensures that there is no cycle of length 4 in the graph (called Tanner graph [3]) that represents the code. This last structural property is very important for a LDPC to achieve good error performance with interative decoding based on belief propagation.

These two class of codes are small but infinite. Long LDPC codes can be constructed algebraically and easily. Codes in these two classes are cyclic and therefore, their encoding can be achieved with linear shift registers with feedback connections based on their generator polynomials.

A LDPC code constructed based on finite geometry can be punctured by removing the columns of the parity-check matrix that correspond to the points of a set of lines. This punctua-

tion results in a shortened LDPC code with at least the same minimum distance as the original code. A finite geometry LDPC code can also be extended by splitting each column of its parity check matrix into multiple columns. This column splitting results in a new parity check matrix with lower density of "ones". Finite geometry LDPC codes, their shortened and extended codes achieve both good bit and word error performances with either soft-decision iterative decoding based on belief propagation or Gallager's hard-decision bit flipping algorithm.

Also presented in this paper are a generalization of the code construction based on multi-dimensional finite geometries and a connection between LDPC codes and balanced incomplete block designs.

2. LDPC Codes Constructed Based on Two-dimensional Euclidean Geometry $EG(2, 2^s)$

Consider the Galois field $GF(2^{2s})$ which is regarded as an extension field of $GF(2^s)$. Let α be a primitive element of $GF(2^{2s})$. Then the powers of $\alpha, \alpha^\infty = 0, \alpha^0 = 1, \alpha^1, \alpha^2, ..., \alpha^{2s-2}$, form all the elements of $GF(2^{2s})$. Each element α^i can be expressed in polynomial form,

$$\alpha^i = b_{i,0} + b_{i,1}\alpha$$

and uniquely represented by a two-tuple $(b_{i,0}, b_{i,1})$ with $b_{i,0}$ and $b_{i,1}$ in $GF(2^s)$. Each two-tuple $(b_{i,0}, b_{i,1})$ over $GF(2^s)$ forms a point in the two-dimensional Euclidean geometry over $GF(2^s)$, denoted $EG(2, 2^s)$, [12-14]. Therefore, Galois field $GF(2^{2s})$ may be regarded as the two-dimensional Euclidean geometry $EG(2, 2^s)$ over $GF(2^s)$. The elements of $GF(2^{2s})$ form all the points of $EG(2, 2^s)$. The zero two-tuple, (0,0), (or the zero element $0=\alpha^\infty$ of $GF(2^{2s})$) is called the origin of $EG(2, 2^s)$.

$EG(2, 2^s)$ consists of points and lines. Let $\mathbf{p}_0$ and $\mathbf{p}_1$ be two linearly independent points in $EG(2, 2^s)$. Then the collection of following 2^s points,

$$\{\mathbf{p}_0 + \beta\mathbf{p}_1\} \tag{2.1}$$

with β in $GF(2^s)$, constitutes a line in $EG(2, 2^s)$ that passes through the point $\mathbf{p}_0$. Of course, this line also passes through other $2^s - 1$ points. Let $\mathbf{p}_2$ be a point which is linearly independent

of $\mathbf{p}_1$ and also linearly independent of $\mathbf{p}_0$. Then the lines, $\{\mathbf{p}_0 + \beta\mathbf{p}_1\}$ and $\{\mathbf{p}_0 + \beta\mathbf{p}_2\}$ have one and only one point, $\mathbf{p}_0$, in common. We say that these two lines intersect at the point $\mathbf{p}_0$. Two lines with two common points are identical. Given a point $\mathbf{p}_0$ in EG$(2, 2^s)$, there are

$$\frac{2^{2s} - 1}{2^s - 1} = 2^s + 1 \tag{2.2}$$

lines in EG$(2, 2^s)$ that intersect at $\mathbf{p}_0$, including the line $\{\beta\mathbf{p}_0\}$ that passes through the origin.

Two lines are said to be parallel if they have no common points. Each line in EG(2,2^s) has $2^s - 1$ other lines parallel to it. No two lines can have more than one point in common. The total number of distinct lines in EG(2,2^s) is

$$2^s(2^s + 1). \tag{2.3}$$

Let $\mathbf{v} = (v_0, v_1, ..., v_{2^{2s}-2})$ be a $(2^{2s} - 1)$-tuple over the binary field GF(2). Number the components of $\mathbf{v}$ with the nonzero elements of GF(2^{2s}) (or the non origin points in EG(2,2^s)) as follows: the component v_i is numbered α^i for $0 \leq i \leq 2^{2s} - 2$. Hence, α^i is the location number of v_i. Let $\mathcal{L}$ be a line in EG$(2, 2^s)$ that does not pass through the origin (0,0) (or α^∞). Based on $\mathcal{L}$, form a binary $(2^{2s} - 1)$-tuple as follows:

$$\mathbf{v}_\mathcal{L} = (v_0, v_1, ..., v_{2^{2s}-2})$$

whose i-th component v_i is 1 if its location number α^i is a point on $\mathcal{L}$; otherwise, v_i is 0. That is, the location numbers for the nonzero components of $\mathbf{v}_\mathcal{L}$ form the points of $\mathcal{L}$. This vector $\mathbf{v}_\mathcal{L}$ is called the incidence vector of line $\mathcal{L}$.

Form a $(2^{2s} - 1) \times (2^{2s} - 1)$ matrix $\mathbf{H}$ whose rows are the incidence vectors of the $2^s(2^s + 1) - (2^s + 1) = 2^{2s} - 1$ lines in EG(2,2^s) that do not pass the origin. It follows from the structural properties of lines in EG(2,2^s) that matrix $\mathbf{H}$ has the following structures: (1) Each row has $\rho = 2^s$ "ones"; (2) Each column corresponds to a non origin point in EG(2,2^s) and has $\gamma = 2^s$ "ones"; and (3) Any two columns have one and only one "1" in common, i.e., $\lambda = 1$. The ratio of the total number of "ones" to the total number of entries in $\mathbf{H}$ matrix, called the density of $\mathbf{H}$, is $r = 2^s/(2^{2s} - 1)$. For large s, this density is very small. Therefore, $\mathbf{H}$ has a very low density of "ones". Actually, the $\mathbf{H}$ matrix can be constructed easily by taking the incidence vector $\mathbf{v}_\mathcal{L}$ for a line $\mathcal{L}$ in EG$(2, 2^s)$ which does not pass through the origin and then cyclically

shifting this incidence vector $\mathbf{v}_{\mathcal{L}}$ $2^{2s}-2$ times. This results in $2^{2s}-1$ incidence vectors for $2^{2s}-1$ distinct lines in EG$(2,2^s)$ that do not pass the origin. The incidence vector $\mathbf{v}_{\mathcal{L}}$ and its $2^{2s}-2$ cyclic shifts form the rows of H matrix.

As an example, consider the Galois field GF$(2^{2\times 2})$ generated by the primitive polynomial $p(X)=1+X+X^4$ which is given in Table 1. Regard this field as the two-dimensional Euclidean geometry EG$(2,2^2)$ over GF(2^2). Let α be a primitive element of GF$(2^{2\times 2})$ and $\beta=\alpha^5$. Then $\{0,1,\beta,\beta^2\}$ form the subfield GF(2^2). Every line in EG$(2,2^2)$ consists of 4 points. The set of 4 points $\{\alpha^{14}+\eta\alpha\}$ with $\eta\in$GF(2^2) is $\{\alpha^7,\alpha^8,\alpha^{10},\alpha^{14}\}$ which forms a line in EG$(2,2^2)$. The incidence vector for this line is (0 0 0 0 0 0 0 1 1 0 1 0 0 0 1). This vector and its 14 cyclic shifts form the matrix H as shown in Figure 1.

Let C be the null space of H. Then C is a LDPC code of length $n=2^{2s}-1$. It turns out that C is the $(0,s)$-th order EG (Euclidean Geometry) code and is cyclic [14]. It is also a type-1 DTI (doubly transitive invariant) code discovered by Lin and Markowshy [16]. The generator polynomial of this code is completely characterized by its roots in GF(2^{2s}). Let h be a nonnegative integer less than 2^{2s}. Then h can be expressed in radix-2^s form as follows:

$$h=\delta_0+\delta_1 2^s$$

where $0\le\delta_i<2^s$ for i=0 and 1. The 2^s-weight of h, denoted $W_{2^s}(h)$, is defined as the following sum,

$$W_{2^s}(h)=\delta_0+\delta_1. \tag{2.4}$$

Let $h^{(l)}$ be the remainder resulting from dividing $2^l h$ by $2^{2s}-1$, i.e.,

$$2^l h=q(2^{2s}-1)+h^{(l)} \tag{2.5}$$

with $0\le h^{(l)}<2^{2s}-1$. Let g(X) be the generator polynomial of C. Then α^h is a root of g(X) if and only if [14]

$$0<\max_{0\le l<s} W_{2^s}(h^{(l)})\le(2^s-1). \tag{2.6}$$

The number of roots can be enumerated and is equal to 3^s-1. C has the following parameters:

Length	$n = 2^{2s} - 1$,
Number of parity bits	$n - k = 3^s - 1$,
Dimension	$k = 2^{2s} - 3^s$,
Minimum distance	$d_{min} = 2^s + 1$.

Example 1: Let $s = 7$. The LDPC code constructed based on EG$(2, 2^7)$ is a (16383, 14197) code with $d_{min} = 129$. The rate of this code is $R = 0.867$. Its parity-check matrix **H** has the following parameters: $\rho = \gamma = 128$ and $\lambda = 1$. The density of **H** is $r = 2^s/(2^{2s} - 1) = 0.007813$. The error performance of this code with various decoding algorithms is shown in Figure 5. $\Delta\Delta$

The third property of a low density parity check matrix **H**, $\lambda = 1$, ensures that there is no rectangle in **H** with four "ones" at its four corners. This implies that the Tanner graph [3] for the code generated by **H** has no cycle of length 4, i.e., no two code bits are checked by the same two parity constraints (or parity check sums as Massey called them [15]).

It follows from the structural properties of matrix **H** that for any bit position α^i with $0 \leq i < 2^{2s} - 1$, there are exact $\gamma = 2^s$ rows in **H** which have the following properties: (1) Each row has a "one" at position α^i; and (2) For $j \neq i$, there is at most one row with a "one" at position α^j. In Messay's terminology, these 2^s rows are said to be orthogonal on the bit position α^i. Parity-check sums formed by these orthogonal vectors can be used to estimate the code bit v_i based on majority-logic rule in one step [14,15]. Therefore, the LDPC code generated by matrix **H** is one-step majority-logic decodable and is capable of correcting 2^{s-1} errors.

A list of EG-LDPC codes with their important parameters is given in Table 2.

3. LDPC Codes Constructed Based on Projective Geometry PG$(2, 2^s)$

Construction of LDPC codes based on a projective geometry is very similar to that based on an Euclidean geometry. In this section, we consider construction of LDPC codes based on the two-dimensional projective geometry PG$(2, 2^s)$ over GF(2^s).

Consider the Galois field GF(2^{3s}) which is regarded as an extension field of GF(2^s). Let α be a primitive element of GF(2^{3s}). Let

$$n = \frac{2^{3s} - 1}{2^s - 1} = 2^{2s} + 2^s + 1. \tag{3.1}$$

Then the order of $\beta = \alpha^n$ is $2^s - 1$. The elements $0, 1, \beta, \beta^2, ..., \beta^{2^s-2}$ form all the elements of the subfield GF(2^s). Consider the set

$$\Gamma = \{\alpha^0, \alpha^1, \alpha^2, ..., \alpha^{n-1}\}. \tag{3.2}$$

Partition the nonzero elements of GF(2^{3s}) into n disjoint subsets as follows:

$$\begin{aligned}
&\{\alpha^0, \beta\alpha^0, \beta^2\alpha^0, \ldots, \beta^{2^s-2}\alpha^0\}, \\
&\{\alpha^1, \beta\alpha^1, \beta^2\alpha^1, \ldots, \beta^{2^s-2}\alpha^1\}, \\
&\{\alpha^2, \beta\alpha^2, \beta^2\alpha^2, \ldots, \beta^{2^s-2}\alpha^2\}, \\
&\qquad\vdots \\
&\{\alpha^{n-1}, \beta\alpha^{n-1}, \beta^2\alpha^{n-1}, \ldots, \beta^{2^s-2}\alpha^{n-1}\}.
\end{aligned} \tag{3.3}$$

We readily see that no element in one set can be a product of an element of GF(2^s) and an element from a different set. Represent each set by its first element as follows:

$$(\alpha^i) \triangleq \{\alpha^i, \beta\alpha^i, \beta^2\alpha^i, \ldots, \beta^{2^s-2}\alpha^i\} \tag{3.4}$$

with $0 \leq i < n$. For any α^j in GF(2^{3s}), if $\alpha^j = \beta^l\alpha^i$ with $0 \leq i < n$, then α^j is represented by (α^i). The n elements

$$(\alpha^0), (\alpha^1), (\alpha^2), ..., (\alpha^{n-1}) \tag{3.5}$$

form the points of the two-dimensional projective geometry PG($2, 2^s$) over GF(2^s) [12,14]. Note that all the $2^s - 1$ elements in (α^i) are regarded as the same point in PG($2, 2^s$).

Let (α^i) and (α^j) be any two distinct points in PG($2, 2^s$). Then the line $\mathcal{L}$ passing through (α^i) and (α^j) consists of points of the following form:

$$(z_1\alpha^i + z_2\alpha^j) \tag{3.6}$$

where z_1 and z_2 are elements from GF(2^s) and are not both equal to zero. Since $(z_1\alpha^i + z_2\alpha^j)$ and $(\beta^l z_1\alpha^i + \beta^l z_2\alpha^j)$ are the same point, the line $\mathcal{L}$ consists of

$$\frac{(2^s)^2 - 1}{2^s - 1} = 2^s + 1 \tag{3.7}$$

points. Let (α^k) be a point not on the line $\{z_1\alpha^i + z_2\alpha^j\}$. Then the line $\{z_1\alpha^i + z_2\alpha^j\}$ and the line $\{z_1\alpha^k + z_2\alpha^j\}$ have (α^j) as a common point (the only common point). We say that these

two lines intersect at the point (α^j). The number of lines in PG$(2, 2^s)$ that intersect at a given point (α^j) is

$$\frac{2^{2s}-1}{2^s-1} = 2^s + 1. \tag{3.8}$$

There are $2^{2s} + 2^s + 1$ distinct lines in PG(2,2^s) [12-14].

Let $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$ be an n-tuple over GF(2). Number the components of $\mathbf{v}$ with the elements in Γ (or the points in PG(2,2^s)) as follows: v_i is numbered α^i for $0 \leq i < n$. Let $\mathcal{L}$ be a line in PG$(2, 2^s)$. Define the incidence vector for $\mathcal{L}$ as follows:

$$\mathbf{v}_{\mathcal{L}} = (v_0, v_1, \ldots, v_{n-1})$$

where

$$v_i = \begin{cases} 1, & \text{if } (\alpha^i) \text{ is a point on } \mathcal{L} \\ 0, & \text{otherwise.} \end{cases}$$

Form a $(2^{2s} + 2^s + 1) \times (2^{2s} + 2^s + 1)$ matrix $\mathbf{H}$ whose rows are the incidence vectors of the $2^{2s} + 2^s + 1$ lines in PG(2,2^s). This matrix $\mathbf{H}$ has the following structural properties: (1) Each row has $\rho = 2^s + 1$ "ones"; (2) Each column corresponds to a point in PG(2,2^s) and has $\gamma = 2^s + 1$ "ones"; and (3) Any two columns have exact one "1" in common, i.e., $\lambda = 1$. The density of this matrix is $r = (2^s + 1)/(2^{2s} + 2^s + 1)$. Again $\mathbf{H}$ can be obtained by cyclically shifting an incident vector $2^{2s} + 2^s$ times. Let $\mathbf{C}$ be the null space of $\mathbf{H}$. Then $\mathbf{C}$ is a LDPC code of length $n = 2^{2s} + 2^s + 1$. This code turns out to be the $(1, s)$-th order PG (projective geometry) code [14] and also is a difference-set code [14, 17]. It is cyclic. Let h be a nonnegative integer less than $2^{3s} - 1$. The generator polynomial $\mathbf{g}(X)$ of $\mathbf{C}$ has α^h as a root if and only if h is divisible by $2^s - 1$ and

$$0 \leq \max_{0 \leq l < s} W_{2^s}(h^{(l)}) = j(2^s - 1) \tag{3.9}$$

with $0 \leq j \leq 1$ [14]. The number of roots in $\mathbf{g}(X)$ is $3^s + 1$. $\mathbf{C}$ has the following parameters:

Length	$n = 2^{2s} + 2^s + 1$,
Number of parity bits	$n - k = 3^s + 1$,
Dimension	$k = 2^{2s} + 2^s - 3^s$,
Minimum distance	$d_{min} = 2^s + 2$.

Example 2: Let $s = 6$. The LDPC code constructed based on the lines of PG$(2, 2^6)$ is a $(4161, 3431)$ code with minimum distance $d_{min} = 66$ and rate $R = 0.82456$. The parity check matrix $\mathbf{H}$ has the following parameters: $\rho = 65, \gamma = 65, \lambda = 1$ and density $r = 0.0156$. The error performance of this code with iterative decoding based on belief propagation is depicted in Figure 4. $\Delta\Delta$

Again Tanner graphs for LDPC codes constructed based on the lines of PG$(2, 2^s)$ do not have cycles of length 4 and the codes are also one-step majority-logic decodable.

A list of PG-LDPC codes with their important parameters is given in Table 3.

4. Puncturing and Extension

Finite geometry codes can be punctured in various ways to obtain good shortened LDPC codes. Consider an EG-LDPC code $\mathbf{C}$ generated by a low density parity check matrix $\mathbf{H}$ whose columns correspond to the $2^{2s} - 1$ non origin points in EG(2,2^s). Let $\mathcal{L}$ be a line in EG(2,2^s) that does not pass through the origin of EG(2,2^s). Then the incidence vector $\mathbf{v}_{\mathcal{L}}$ of $\mathcal{L}$ is a row in $\mathbf{H}$. Remove the columns of $\mathbf{H}$ that correspond to the 2^s points on $\mathcal{L}$. This results in a matrix $\mathbf{H}'$ with $2^{2s} - 2^s - 1$ columns. The row in $\mathbf{H}$ that is the incidence vector $\mathbf{v}_{\mathcal{L}}$ of $\mathcal{L}$ becomes a zero row in $\mathbf{H}'$. Removing this zero row from $\mathbf{H}'$, we obtain a $(2^{2s} - 2) \times (2^{2s} - 2^s - 1)$ matrix $\mathbf{H}_s$. Each column of $\mathbf{H}_s$ still has $\gamma = 2^s$ "ones" as the original matrix $\mathbf{H}$. Removing a column of $\mathbf{H}$ that corresponds to a point α^i on $\mathcal{L}$ will delete a "one" from $2^s - 1$ rows in $\mathbf{H}$ which are the incidence vectors of the lines that intersect with line $\mathcal{L}$ at the point α^i. Therefore, there are $2^s(2^s - 1)$ rows in $\mathbf{H}_s$ with weight $\rho_1 = 2^s - 1$. There are $2^s - 2$ lines not passing through the origin of EG(2,2^s) that are parallel to $\mathcal{L}$. Deleting the columns of $\mathbf{H}$ that correspond to the points on $\mathcal{L}$ does not change the weights of the rows which are the incidence vectors of the $2^s - 2$ lines parallel to $\mathcal{L}$. Therefore, there are $2^s - 2$ rows in $\mathbf{H}_s$ with weight $\rho_2 = 2^s$. Any two columns in $\mathbf{H}_s$ still have exactly one "1" in common, i.e., $\lambda = 1$. The density of $\mathbf{H}_s$ is $r = 2^s/(2^{2s} - 2)$. Therefore $\mathbf{H}_s$ is still a low density matrix.

Let $\mathbf{C}_s$ be the null space of $\mathbf{H}_s$. Then $\mathbf{C}_s$ is a LDPC code of length $n = 2^{2s} - 2^s - 1$. Clearly, $\mathbf{C}_s$ is obtained from $\mathbf{C}$ by deleting the components of each codeword at the positions that correspond to the points on the line $\mathcal{L}$. Since the rows of $\mathbf{H}_s$ do not have the same weight,

$\mathbf{C}_s$ is an irregular LDPC code. However, all the columns of $\mathbf{H}_s$ have the same weight which is still $\gamma = 2^s$. Therefore, we still can form 2^s parity-check sums orthogonal on any code bit. Consequently the majority-logic correcting capability is still 2^{s-1} and hence its minimum distance is at least $2^s + 1$ [14,15].

Example 3: Let $s = 4$. The EG-LDPC code with $s = 4$ is a (255, 175) code (the third code in Table 2) with rate $R = 0.6863, d_{min} = 17, \rho = \gamma = 16$ and density $r = 0.0627$. A line in EG(2,2^4) has 16 points. Puncturing this EG-LDPC code based on a line in EG(2,2^4) not passing through the origin results in a (239,160) LDPC code with rate $R = 0.667$, minimum distance at least 17 and density $r = 0.0630$. Note that the punctuation removes 15 information bits and one parity check bit from the (255,175) EG-LDPC code. Figure 6 shows that the error performance of this punctured code is slightly better than that of the original code. $\Delta\Delta$

An EG-LDPC code can be punctured based on the points of a set of q parallel lines with $1 \leq q \leq 2^s - 1$. This results in a shortened LDPC code $\mathbf{C}_s$ of length $n = 2^{2s} - q \cdot 2^s - 1$. Its parity check matrix $\mathbf{H}_s$ has $2^{2s} - q - 1$ rows and $2^{2s} - q \cdot 2^s - 1$ columns. Each column of $\mathbf{H}_s$ still has weight 2^s but its rows have different weights.

Example 4: If we puncture the (255,175) EG-LDPC code (given in Example 3) based on the points of two parallel lines in EG(2,2^4), we obtained a (223,145) LDPC code with rate $R = 0.650$ and $d_{min} \geq 17$. The parity-check matrix $\mathbf{H}_s$ of this code is a 253×223 matrix. Thirteen rows of $\mathbf{H}_s$ have weight 16 and all the other rows have weight 14. All the columns of $\mathbf{H}_s$ have weight 16. The density of $\mathbf{H}_s$ is r=0.0632. The puncturing removes 30 information bits and 2 parity check bits from the original code. $\Delta\Delta$

Puncturing can also be achieved with combination of removing columns and rows of the low density parity check matrix $\mathbf{H}$. For example, let $\mathbf{Q}$ be a set of l lines in EG(2,2^s) not passing through the origin that intersect at a common point α^i, where $1 \leq l \leq 2^s$. Let $\mathbf{P}$ be the set of lines in EG(2,2^s) that are parallel to the lines in $\mathbf{Q}$. Suppose we puncture $\mathbf{H}$ as follows: (1) remove all the rows in $\mathbf{H}$ that are the incidence vectors of the lines in $\mathbf{Q}$ and $\mathbf{P}$; and (2) remove the columns that correspond to the points on the lines in $\mathbf{Q}$. The total number of distinct points on the lines in $\mathbf{Q}$ is $l \cdot (2^s - 1) + 1$. The total number of lines in $\mathbf{Q}$ and $\mathbf{P}$ is $l \cdot (2^s - 1)$. Therefore, puncturing results in a $(2^{2s} - l \cdot 2^s + l - 1) \times (2^{2s} - l \cdot 2^s + l - 2)$ matrix $\mathbf{H}_s$. Both columns and rows of $\mathbf{H}_s$ do not have constant weights any more. The null space of $\mathbf{H}_s$ gives an irregular

LDPC code.

Example 5: Consider puncturing the (255,175) EG-LDPC code. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two lines in EG(2,2^s) not passing through the origin that intersect at the point α^i. There are 28 lines not passing through the origin parallel to either $\mathcal{L}_1$ or $\mathcal{L}_2$. Puncturing the parity check matrix $\mathbf{H}$ of the (255,175) EG-LDPC code based on $\mathcal{L}_1$, $\mathcal{L}_2$ and their parallel lines results in a 225×224 matrix $\mathbf{H}_s$. Fourteen rows in $\mathbf{H}_s$ have weight 15 and all the other rows have weight 14. Twenty-nine columns of $\mathbf{H}_s$ have weight 15 and all the other columns have weight 14. The density of $\mathbf{H}_s$ is $r = 0.06278$. The LDPC code generated by $\mathbf{H}_s$ is a (224,146) code with minimum distance at least 15. $\Delta\Delta$

All the above example shortened EG-LDPC codes have about the same error performance as shown in Figure 6.

PG-LDPC codes can be shortened in a similar way. Consider the two dimensional projective geometry PG(2,2^s) over GF(2^s). Let $\mathcal{L}$ be a line in PG(2,2^s). If all the points on $\mathcal{L}$ are removed from PG(2,2^s), we obtain the two dimensional Euclidean geometry EG(2,2^s) [12,13]. This fact gives a unique relationship between PG-LDPC codes and EG-LDPC codes. Let $\mathbf{C}$ be a PG-LDPC code of length $2^{2s} + 2^s + 1$ with low density parity check matrix $\mathbf{H}$. Recall that $\mathbf{H}$ is a $(2^{2s} + 2^s + 1) \times (2^{2s} + 2^s + 1)$ square matrix whose columns correspond to the points in PG(2,2^s) and whose rows are the incidence vectors of the lines in PG(2,2^s). Suppose we delete the columns from $\mathbf{H}$ that correspond to the points on a line $\mathcal{L}$ in PG(2,2^s). The column removal results in a zero row. Remove this zero row. We obtain a $(2^{2s} + 2^s) \times 2^{2s}$ matrix $\mathbf{H}_0$. The columns of $\mathbf{H}_0$ correspond to the 2^{2s} points in EG(2,2^s) (including the origin) and the rows $\mathbf{H}_0$ are the incidence vectors of all the $2^s(2^s + 1)$ lines in EG(2,2^s) (including the lines passing through the origin). This matrix $\mathbf{H}_0$ is also a low density parity check matrix with $\rho = 2^s, \gamma = 2^s + 1, \lambda = 1$ and $r = 1/2^s$. The null space of $\mathbf{H}_0$ gives a LDPC code $\mathbf{C}_0$ with minimum distance $2^s + 2$. $\mathbf{C}_0$ is called a zero-type EG-LDPC code and is not cyclic.

The first column of $\mathbf{H}_0$ corresponds to the origin α^∞ of EG(2,2^s). Suppose we remove from $\mathbf{H}_0$ the first column and the rows which are the incidence vectors of the $2^s + 1$ lines in EG(2,2^s) that pass through the origin. We obtain a $(2^{2s} - 1) \times (2^{2s} - 1)$ square matrix which is the low density parity check matrix of the EG-LDPC code of length $2^{2s} - 1$. Therefore we may regard that the EG-LDPC codes are descendants of the PG-LDPC codes. For example, the (255,175)

EG-LDPC code is a descendant of the (273,191) PG-LDPC code. Note that puncturing 18 components from the (273,191) PG-LDPC code removes 16 information bits and 2 parity check bits from each codeword.

Clearly, puncturing a PG-LDPC code can be achieved based on a set of lines in PG(2,2^s).

A finite geometry LDPC code C of length n can be extended by splitting each column $\mathbf{h}$ of its parity check matrix $\mathbf{H}$ into q columns with $2 \leq q \leq 2^s$ (for an EG-LDPC code) and $2 \leq q \leq 2^s + 1$ (for a PG-LDPC code) which are of the same length as $\mathbf{h}$. Consider the parity check matrix $\mathbf{H}_{EG}$ of an EG-LDPC code $\mathbf{C}_{EG}$. Dividing 2^s by q, we have

$$2^s = \gamma_{ext} \times q + b$$

where $0 \leq b < q$. Split each column $\mathbf{h}_i$ of $\mathbf{H}_{EG}$ into q columns $\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, ..., \mathbf{h}_{i,q}$ such that b columns, $\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, ..., \mathbf{h}_{i,b}$, have weight $\gamma_{ext} + 1$ and $q - b$ columns, $\mathbf{h}_{i,b+1}, ..., \mathbf{h}_{i,q}$, have weight γ_{ext}. The distribution of 2^s "ones" of $\mathbf{h}_i$ into $\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, ..., \mathbf{h}_{i,q}$ is carried out in a rotating manner, i.e., the first "1" of $\mathbf{h}_i$ is put in $\mathbf{h}_{i,1}$, the second "1" of $\mathbf{h}_i$ is put in $\mathbf{h}_{i,2}$, and so on. This column splitting results in a $(2^{2s} - 1) \times q(2^{2s} - 1)$ low density parity check matrix $\mathbf{H}_{ext,EG}$ which has the following structural properties: (1) each row has weight 2^s; (2) the minimum column weight is $\lfloor 2^s/q \rfloor$ (if 2^s is divisible by q, each column has weight $2^s/q$, otherwise there are two different column weights, $\lfloor 2^s/q \rfloor$ and $\lfloor 2^s/q \rfloor + 1$); (3) any two columns have at most one "1" in common; and (4) the density of the matrix is $r_{ext,EG} = 2^s/q(2^{2s} - 1)$. The null space of $\mathbf{H}_{ext,EG}$ gives an extended EG-LDPC code $\mathbf{C}_{ext,EG}$ of length $n = q(2^{2s} - 1)$.

Example 6: Let $s = 5$. There exists a (1023,781) EG-LDPC code (the fourth code given in Table 2). Suppose we choose q=8. Then column splitting results in a (8184,7162) extended EG-LDPC code of rate $R = 0.875$ whose parity check matrix has the following parameters: $\rho = 32, \gamma_{ext} = 4, \lambda = 1$ and $r = 0.0039125$. The error performance of this extended code is shown in Figure 7. We see that the extension results in an increase of code rate. ΔΔ

PG-LDPC codes can be extended in the same manner.

Puncturing and extension give more choices of LDPC codes. In fact, puncturing and column splitting can be combined to obtain LDPC code. Proper extension also increase the code rate.

5. Generalization

The construction of LDPC codes based on two-dimensional finite geometries over GF(2^s), EG(2,2^s) and PG(2,2^s)), can be generalized to construction based on m-dimensional finite geometries over GF(2^s), EG($m, 2^s$) and PG($m, 2^s$)) with $m \geq 2$.

Consider the m-dimensional Euclidean geometry over GF(2^s), EG($m, 2^s$). This geometry consists of 2^{ms} points and

$$\frac{2^{(m-1)s}(2^{ms}-1)}{2^s-1} \tag{5.1}$$

lines. Each point in EG($m, 2^s$) is represented by an m-tuple over GF(2^s) and the zero m-tuple (0, 0, ..., 0) is the origin of the geometry. The Galois field GF(2^{ms}) as an extension field of GF(2^s) may be regarded as a representation of EG($m, 2^s$) [14]. Let α be a primitive element of GF(2^{ms}). Then the 2^{ms} elements, $\alpha^\infty = 0, \alpha^0, \alpha, \alpha^2, ..., \alpha^{2^{ms}-2}$, represent the 2^{ms} points of EG($m, 2^s$) and the element $0 = \alpha^\infty$ represents the origin of the geometry. Each line in EG($m, 2^s$) is a collection of 2^s points as defined by (2.1) for the two-dimensional case. For any point p in EG($m, 2^s$), there are

$$(2^{ms}-1)/(2^s-1) \tag{5.2}$$

lines in EG($m, 2^s$) that intersect at p. Let $\mathcal{L}$ be a line in EG($m, 2^s$) which does not pass through the origin. The incidence vector of $\mathcal{L}$ is defined as a $(2^{ms}-1)$-tuple over GF(2), $\mathbf{v}_\mathcal{L} = (v_0, v_1, ..., v_{2^{ms}-2})$, such that for $0 \leq i \leq 2^{ms}-2$, $v_i = 1$ if and only if α^i is a point in $\mathcal{L}$. Therefore the weight of $\mathbf{v}_\mathcal{L}$ is 2^s.

Let H be a matrix whose rows are the incidence vectors of the lines in EG($m, 2^s$) that do not pass through the origin. H consists of $2^{ms}-1$ columns and

$$(2^{(m-1)s}-1)\frac{2^{ms}-1}{2^s-1} \tag{5.3}$$

rows. The columns correspond to the $2^{ms}-1$ non-origin points of EG($m, 2^s$). Each row of H consists of 2^s "ones" and each column of H consists $(2^{ms}-1)/(2^s-1)$ "ones". The number of "ones" in each column of H is simply equal to the number of lines in EG($m, 2^s$) that intersect at a point in EG($m, 2^s$). Any two columns of H have exactly one "1" in common. The density

of the matrix is

$$r = \frac{2^s}{2^{ms} - 1}.$$

If m is large, r is very small. Therefore $\mathbf{H}$ is a low density matrix. The m-dimensional EG-LDPC code of length $2^{ms} - 1$ is simply the null space of $\mathbf{H}$. It is a cyclic and one-step majority-logic decodable. Let h be a nonnegative integer less than $2^{ms} - 1$. The generator polynomial $\mathbf{g}(X)$ of this code has α^h as a root if and only if [14]

$$0 < \max_{0 \le l < s} W_{2^s}(h^{(l)}) \le (m-1)(2^s - 1).$$

Since $(2^{ms} - 1)/(2^s - 1)$ orthogonal check-sums can be formed for majority-logic decoding of each code bit, its minimum distance is at least

$$\frac{2^{ms} - 1}{2^s - 1} + 1. \tag{5.4}$$

Example 7: Let $s = m = 3$. The three-dimensional EG-LDPC code constructed based on the lines of EG(3,2^3) is a (511,139) code with $d_{min} = 73$ and rate $R = 0.272$. Its parity check matrix $\mathbf{H}$ is a 4599×511 matrix with $\rho = 8, \gamma = 72, \lambda = 1$ and density $r = 0.01565$. The error performance of this code is shown in Figure 8. ΔΔ

The above generalization results in a large class of EG-LDPC codes. For $m \ge s$, EG-LDPC codes in general have rates less than $1/2$.

An m-dimensional EG-LDPC code can also be punctured to obtain a shortened EG-LDPC code. Let EG$(m-1, 2^s)$ be an $(m-1)$-flat (or $(m-1)$-dimensional subspace) of EG$(m, 2^s)$ that does not contain the origin. We delete the columns of $\mathbf{H}$ that corresponds to the points in EG$(m-1, 2^s)$. This results in a matrix $\mathbf{H}'$ with $2^{ms} - 2^{(m-1)s} - 1$ columns. The rows in $\mathbf{H}$ that correspond to the incidence vectors of lines contained in EG$(m-1, 2^s)$ becomes zero rows in $\mathbf{H}'$. Removing these zero rows in $\mathbf{H}'$, we obtain a low density matrix $\mathbf{H}_s$ with $2^{ms} - 2^{(m-1)s} - 1$ columns and

$$(2^{ms} - 2^{(m-2)s} - 1)\frac{2^{(m-1)s} - 1}{2^s - 1} \tag{5.5}$$

rows. Each column of $\mathbf{H}_s$ still have $\gamma = (2^{ms} - 1)/(2^s - 1)$ "ones". But the rows of $\mathbf{H}_s$ do not have uniform weight any more, in fact, there are two different weights, $2^s - 1$ and 2^s. The null

space of $\mathbf{H}_s$ gives a shortened m-dimensional EG-LDPC code. This shortened code is not cyclic any more.

Similarly, we can construct an m-dimensional PG-LDPC code based on the lines of the m-dimensional projective geometry over GF(2^s), PG($m, 2^s$). An m-dimensional LDPC code is simply the null space of a matrix $\mathbf{H}$ whose rows are the incidence vectors of the lines in PG($m, 2^s$). The columns of $\mathbf{H}$ correspond to the $(2^{(m+1)s}-1)/(2^s-1)$ points of PG($m, 2^s$). $\mathbf{H}$ has the following parameters: (1) each row has 2^s+1 "ones"; (2) each column has $(2^{ms}-1)/(2^s-1)$ "ones"; (3) any two columns have exactly one "1" in common; and (4) the density of the matrix is $(2^{2s}-1)/(2^{(m+1)s}-1)$. The m-dimensional PG-LDPC code is cyclic. Let h be a nonnegative integer less than $2^{(m+1)s}-1$. The generator polynomial of the code has α^h as a root if and only if h is divisible by 2^s-1 and

$$0 \leq \max_{0 \leq l < s} W_{2^s}(h^{(l)}) \leq j(2^s-1) \tag{5.6}$$

with $0 \leq j \leq m-1$ [14]. The minimum distance of the code is at least

$$(2^{ms}-1)/(2^s-1)+1. \tag{5.7}$$

Example 8: Let $m = s = 3$. The 3-dimensional PG-LDPC code is a (585,184) code with $d_{min} = 74$. ΔΔ

Again for $m \geq s$, m-dimensional PG-LDPC codes in general have rates less than 1/2.

Multidimensional finite geometry LDPC codes can be extended by splitting the columns of their parity check matrices.

Example 9: Consider the three-dimensional (511,139) EG-LDPC code given in Example 7. Suppose we extend this code by splitting each column of its parity check matrix $\mathbf{H}$ into 24 columns. Then the extended code $\mathbf{C}_{ext,EG}$ is a (12264,7665) LDPC code with rate $R_{ext,EG} = 0.625$. The extension results in a high rate code. The density of the parity check matrix $\mathbf{H}_{ext,EG}$ of this extended code is $r_{ext,EG} = 0.000652$ and $\gamma_{ext,EG} = 3$. The bit error performance of this extended EG-LDPC code with iterative decoding based on belief propagation is shown in Figure 9, which is only 1.0 dB away from Shannon limit. ΔΔ

Examples 6 and 9 and Figures 7 and 9 show that column splitting is a powerful method for contructing long and high-rate LDPC codes with good error performance.

6. Relation to Balanced Incomplete Block Designs

Analysis and design of experiments is an important subject in combinatorial mathematics. The objective of this subject is to design experiments systematically and with a view to their statistical analysis. One such design is called the balanced incomplete block design[12,13]. This type of design was used for block code construction in late sixties by some coding theorists [18-22].

Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a set of n objects. A balanced incomplete block design (BIBD) of $\mathbf{X}$ is a collection of b ρ-subsets of $\mathbf{X}$, denoted by $B_1, B_2, ..., B_b$ and called the blocks, such that the following conditions are satisfied:

(1) Each object appears in exactly γ of the b blocks;

(2) Every two objects appear simultaneously in exactly λ of the b blocks; and

(3) $\rho < n$.

So, a BIBD is characterized by the five parameters b, n, γ, ρ and λ , it is also called a $(b, n, \gamma, \rho, \lambda)$-configuration. Instead of a list of the ρ-subsets, a BIBD can also be described by its incidence matrix $\mathbf{Q}$, which is a $b \times n$ matrix with $0's$ and $1's$ as entries. The columns of the matrix correspond to the objects and the rows of the matrix correspond to the blocks. The entry in the i-th row and j-th column of $\mathbf{Q}$ is a 1 if the object x_j is in the block B_i and is 0 otherwise.

The matrix given by Figure 1 is actually the incidence matrix of a BIBD with $b = n = 15, \rho = \gamma = 4$ and $\lambda = 1$. Therefore, the low density parity check matrices constructed in sections 2 and 3 based on EG$(2, 2^s)$ and PG$(2, 2^s)$ give two special classes of BIBD's with λ =1. From opposite point of view, if a BIBD with small ρ and γ and λ =1, then its incidence matrix can be used as the parity check matrix to generate a LDPC code. Over the years, there are many such BIBD's that have been constructed [12,13]. This construction may yield good LDPC codes. This should be a direction for further investigation. For example, for any positive integer t such that $4t+1$ is the power of a prime, there exists a BIBD with $n = 20t+5, b = (5t+1)(4t+1), \gamma = 4t+1, \rho = 5$ and $\lambda = 1$. The incidence matrix of this BIBD has density $r = 5/(20t + 5)$ [12,13]. Is the code generated with this matrix as the parity check matrix good? This question should be answered.

7. Decoding of Finite Geometry LDPC Codes and Simulation Results

Finite geometry LDPC codes and their shortened and extended codes can be decoded with the following algorithms: (1) soft-decision iterative decoding based on belief propagation (IDBP) [8]; (2) Gallager's hard-decision bit flipping (BF) iterative algorithm [1,2]; and (3) majority-logic decoding [13,14]. The soft-decision IDBP algorithm gives the best error performance but it requires very large computational complexity and decoding delay. Majority-logic decoding requires the least decoding complexity but its error performance is relative poor compared to the IDBP algorithm. The BF decoding algorithm provides a very good trade-off between the error performance of the soft-decision IDBP algorithm and the decoding complexity of the majority-logic decoding. It performs better than the majority-logic decoding and not far from the soft-decision IDBP algorithm, which will be shown later by simulation results.

The IDBP algorithm is well explained in [8] and will not be repeated in this paper. The BF algorithm is described in [1,2,24,25], but it is not very well known and not being used for decoding LDPC codes. For this reason, we give a brief review of this algorithm. Majority-logic decoding can be found in [13,14].

Let $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$ be a codeword in a low-density parity-check code C. Let $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_J$ denote the rows of H matrix. Since C is the null space of H, the inner product $\mathbf{v} \cdot \mathbf{h}_j$ must be zero for $1 \leq j \leq J$, i.e.,

$$\mathbf{v} \cdot \mathbf{h}_j = 0 \tag{7.1}$$

for $1 \leq j \leq J$. Let

$$\mathbf{h}_j = (h_{j,0}, h_{j,1}, ..., h_{j,n-1}). \tag{7.2}$$

Then from (7.1) and (7.2), we have the following J parity-check equations (or sums):

$$\sum_{i=0}^{n-1} v_i h_{j,i} = 0 \tag{7.3}$$

for $1 \leq j \leq J$.

From the second structural property of H, we readily see that every code bit of a codeword in C is checked exactly by γ parity-check equations. Consider the set of γ parity-check equations that check a particular code bit v_i. Since $\lambda = 1$, none of the other $n - 1$ code bits are checked

by more than one of these γ parity-check equations. From this fact, we see that if there is a single error in the received vector $\mathbf{r}$ at position i, then the γ parity-check equations that check the bit at position i will fail, i.e.,

$$\mathbf{r} \cdot \mathbf{h}_j = 1 \tag{7.4}$$

with $h_{j,i} = 1$ and $j \in \{1, 2, \cdots, J\}$. If there are two errors in the received vector $\mathbf{r}$, because of $\lambda \ll \gamma$, the number of parity check failures will increase, and the number of failed parity-check equations that contain either of the two erroneous bits will be greater than $\lfloor \frac{\gamma}{2} \rfloor$, provided that the error correcting capability of the code $t = \lfloor (d_{min}(C) - 1)/2 \rfloor \geq 2$. The number of parity-check failures increase as the number of errors in $\mathbf{r}$ increases until it reaches the error correcting capability t of the code. When the number of errors exceeds t, there will be error patterns which result in a decrease in the number of parity-check failures. In this case, an erroneous bit position may be checked by less than $\lfloor \gamma/2 \rfloor$ failed parity-check equations or by none of the failed parity-check equations. From the above analysis, we find that if the number of errors in the received vector is t or less, changing an erroneous bit results in a decrease in the number of parity-check equation failures. However, if the number of errors exceeds the error correcting capability t, changing an erroneous bit may result in an increase in the number of parity-check failures.

Based on the above analysis of the parity-check equation failures, Gallager devised the following decoding scheme for LDPC codes. The decoder computes all the parity-check sums based on (7.3) and then changes any bit in the received vector $\mathbf{r}$ that is contained in more than some fixed number δ of unsatisfied parity-check equations. Using these new values, the parity check sums are recomputed, and the process is repeated until the parity-check equations are all satisfied.

The above decoding is an iterative decoding algorithm. The parameter δ, called threshold, is a design parameter which should be chosen to optimize the error performance while minimizes the number of computations of parity-check sums. It is clear that the value of δ depends on the code parameters $\rho, \gamma, \lambda, d_{min}(C)$ and the signal-to-noise ratio (SNR).

If decoding fails for a given value of δ, then the value of δ should be reduced to allow further decoding iterations. For error patterns with number of errors less than or equal to the error

correcting capability of the code, the decoding will be completed in one or a few iterations. Otherwise, more decoding iterations are needed. Therefore, the number of decoding iterations is a random variable and is a function of the channel SNR. A limit maybe set on the number of iterations. When this limit is reached, the decoding process is terminated to avoid excessive computations.

Due to the nature of low-density parity checks, the above decoding algorithm corrects many error patterns with number of errors exceeding the error correcting capability of the code.

A very simple BF decoding algorithm is given below:

Step 1 Compute the parity-check equations. If all the parity-check equations are satisfied, stop.

Step 2 Find the number of unsatisfied parity-check equations for each bit, denoted f_i, $i = 0, 1, ..., n-1$.

Step 3 Identify the set $\mathcal{S}$ of bits for which f_i is the largest.

Step 4 Flip the bits in set $\mathcal{S}$.

Step 5 Repeat steps 1 to 4 until all the parity-check equations are satisfied (for this case, we stop the iteration in step 1) or a predefined maximum iteration number is reached.

The above simple BF decoding algorithm can be improved by using adaptive thresholds δ's. Of course, this improvement is achieved at the expense of more computations.

Bit and word error probabilities of many finite geometry LDPC codes, their punctured and extended codes have been computed based on IDBP, BF and majority-logic decoding algorithms as shown in Figures 2-10. From these figures, we first notice that the soft-decision IDBP algorithm gives the best error performance among the three decoding algorithms at the expense of extensive computational complexity. However, Gallager's hard-decision BF algorithm achieves relatively good error performance with much less computational complexity. It outperforms the simple majority-logic decoding. Therefore, it provides a good trade-off between the error performance of the IDBP algorithm and the complexity of majority-logic decoding algorithm. Next we notice from Figures 2, 3 and 4 that PG-LDPC codes and their descendant EG-LDPC codes have (almost) identical error performance. Figure 6 shows that the (255,175) EG-LDPC code and its punctured codes have almost the same error performance. Figure 7 shows that the (8184,7162) LDPC code obtained by extending the (1023,781) EG-LDPC code with column

splitting achieves an error performance within 0.9 dB from Shannon limit. Figures 9 gives the error performance of the (12264,7665) LDPC code obtained from the 3-dimensional (511,139) by column splitting. We see that its error performance is about 1 dB away from Shannon limit. Figures 7 and 9 suggest that proper column splitting of finite geometry codes may result in very good LDPC codes.

Figures 2 and 3 also give a comparison of the error performance of the finite geometry LDPC codes of lengths 255 (or 273) and 1023 (or 1057) to that of some best computer generated Gallager LDPC codes of the same lengths and same rates. We see that these finite geometry codes outperform their corresponding computer generated Gallager LDPC codes. However, for large code lengths, the best computer generated LDPC codes should perform better than the finite geometry codes because they are chosen from larger code ensembles. Finally Figure 10 shows how fast the (4161,3431) PG-LDPC code converges to its ultimate error performance using IDBP algorithm. We see that 20 iterations are enough to terminate the decoding iteration process for this code. For all the finite geometry codes that we have simulated, the IDBP algorithm converges very fast.

8. Conclusion

In this paper, we have reported two classes of LDPC codes that are constructed based on the lines of two-dimensional Euclidean and projective geometries. These finite geometry codes can be decoded with soft-decision IDBP algorithm, Gallager's BF algorithm and majority-logic decoding. Simulation results show that they perform very well and close to their Shannon limits with the IDBP algorithm. These codes also perform well with Gallager's BF algorithm. The BF algorithm provides a good trade-off between the good error performance of the IDBP algorithm and the simple complexity of majority-logic decoding. Furthermore, they are cyclic and hence can be easily encoded with feedback shift registers in contrast to the complex encoding of computer generated LDPC codes. We have also shown that these finite geometry codes can be punctured and extended to obtain good LDPC codes. Proper extension by column splitting results in good high rate codes.

We have also extended the two-dimensional finite geometry LDPC codes to multidimensional

finite geometry codes. This extension gives a large class of LDPC codes with various rates. A close relationship between a low density parity check matrix and a balanced incomplete block design has also been presented. This relationship may allow us to construct LDPC codes based on many existing balanced incomplete designs.

The results presented in this paper show that algebraic or geometric construction of LDPC codes are possible and the constructed codes are good. Therefore, further investigation is needed and we should give algebraic construction a chance as Tanner suggested[23].

Acknowledgement

The authors wish to thank David J.C. MacKay for providing the parity check matrices of the Gallager LDPC codes used in this paper.

References

[1] R. G. Gallager,"Low Density Parity Check Codes," *IRE Transactions on Information Theory*, IT-8, pp. 21-28, January 1962.

[2] R. G. Gallager, *Low Density Parity Check Codes*, MIT Press, Cambridge, Mass., 1963.

[3] R. M. Tanner,"A Recursive Approach to Low Complexity Codes,"*IEEE Transactions on Information Theory*, IT-27, pp. 533-547, September 1981.

[4] D. J. C. MacKay and R. M. Neal,"Near Shannon Limit Performance of Low Density Parity Check Codes," *Electronics Letters* 32 (18): 1645-1646, 1996.

[5] M. C. Davey and D. J. C. MacKay,"Low Density Parity Check Codes over GF(q)," *IEEE Communications Letters*, June 1998.

[6] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved Low-Density Parity-Check Codes Using Irregular Graphs and Belief Propagation," *Proceedings of 1998 IEEE International Symposium on Information Theory*, pp. 171, Cambridge, Mass., August 16-21, 1998.

[7] D. J. C. MacKay,"Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Transactions on Information Theory*, IT-45, pp.399-432, March 1999.

[8] R. J. McEliece, D. J. C. MacKay, and J. -F. Cheng,"Turbo Decoding as an Instance of Pearl's Belief Propagation Algorithm," *IEEE Journal on Selected Areas*, Vol. 16, pp. 140-152, February 1998.

[9] M. Fossorier, M. Mihaljevic, and H. Imai,"Reduced Complexity Iterative Decoding of Low Density Parity Check Codes," *IEEE Transactions on Communications*, Vol. 47, pp. 673-680, May 1999.

[10] R. Lucas, M. Fossorier, Y. Kou, and S. Lin,"Iterative Decoding of One-Step Majority Logic Decodable Codes Based on Belief Propagation," *submitted to IEEE Transactions on Communications*, 1999.

[11] D. J. C. MacKay,"Sparce Graph Codes,"" *Proceedings of the 5-th International Symposium on Communication Theory and Applications*, pp. 2-4, Ambleside, UK, July 11-16, 1999.

[12] H. B. Mann, *Analysis and Design of Experiments*, Dover Publications, New York, 1949.

[13] A. P. Street and D. J. Street, *Combinatorics of Experimental Design*, Oxford Science Publications, Clarendon Press, Oxford, 1987.

[14] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1983.

[15] J. L. Massey, *Threshold Decoding*, MIT Press, Cambridge, Mass. 1963.

[16] S. Lin and G. Markowsky,"On a Class of One-Step Majority-Logic Decodable Cyclic Codes," *IBM Journal of Research and Developments*, January 1980.

[17] E. J. Weldon, Jr.,"Difference-Set Cyclic Codes," *Bell System Technical Journal*, 45, pp. 1045-1055, September 1966.

[18] S. Lin,"Some Codes Which are Invariant under a Transitive Permutation Group and Their Connection with Balanced Incomplete Block Designs," *Combinatorial Mathematics and Its Applications*, edited by R. C. Bose and T. A. Dowling, Chapter 24, University of North Carolina Press, Chapel Hill, North Carolina 1969.

[19] V. Pless,"On a New family of Symmetry Codes and Related New Five Designs," *Bulletin of American Mathematics Society*, 75, 1339-1342.

[20] N. V. Semakov and V. A. Zinov'ev,"Balanced Codes and Tactical Configurations," *Problems of Information Transmission*, 5, pp 22-28, 1969.

[21] N. Hamada, "On the p-rank of the Incidence Matrix of a Balanced or Partially Balanced Incomplete Block Design and Its Applications to Error-Correcting Codes," *Hiroshima Mathematic Journal*, 3, pp. 153-226, 1973

[22] I. Blake and R. C. Mullin, *The Mathematical Theory of Coding*, Academic Press, New York, 1975.

[23] M. Tanner, "On Quasi-Cyclic Repeat Accumulate Codes", *17th Annual Allterton Conference on Communication, Control, and Computing*, Allerton House, Monticello, IL. September 22-24, 1999.

[24] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710-1722, Nov. 1996.

[25] Albert M. Chan and Frank R. Kschischang, "A Simple Taboo-Based Soft-Decision Decoding Algorithm for Expander Codes", *IEEE Communication Letters*, Vol. 2, pp.183-185, No. 7, July 1998.

List of figure captions

Figure 1. A low density parity check matrix.

Figure 2. Bit-error probabilities of the (255, 175) EG-LDPC code (maximum 50 iterations), (273,191) PG-LDPC code(maximum 50 iterations) and two (273,191) Gallager codes generated by computer (maximum 200 iterations) based on IDBP algorithm.

Figure 3. Bit- and word-error probabilities of the (1023,781) EG-LDPC code, the(1057,813) PG-LDPC code and two (1057,813) Gallager codes generated by computer .

Figure 4. Bit- and word-error probabilities of the (4095,3367) EG-LDPC code and the(4161,3431) PG-LDPC code.

Figure 5. Bit- and word-error probabilities of the (16383,14179) EG-LDPC code.

Figure 6. Bit-error probabilities of the (255,175) EG-LDPC code and its (239,160), (223,145) and (224,146) punctured codes.

Figure 7. Bit- and word-error probabilities of the (8184,7162) LDPC code obtained by extending the (1023,781) EG-LDPC code using column splitting with q=8.

Figure 8. Bit- and word-error probabilities of the (511,139) EG-LDPC code.

Figure 9. Bit- and word-error probabilities of the (12264,7665) LDPC code obatined by extending the (511,139) EG-LDPC code by usuing column splitting with q=8.

Figure 10. Convergence of the IDBP algorithm for the (4161,3431) PG-LDPC code.

Table 1: The elements of $GF(2^4)$ generated by $p(X) = 1 + X + X^4$

Power representation	Polynomial representation	4-Tuple representation
0	0	(0 0 0 0)
1	1	(1 0 0 0)
α	α	(0 1 0 0)
α^2	α^2	(0 0 1 0)
α^3	α^3	(0 0 0 1)
α^4	$1 + \alpha$	(1 1 0 0)
α^5	$\alpha + \alpha^2$	(0 1 1 0)
α^6	$\alpha^2 + \alpha^3$	(0 0 1 1)
α^7	$1 + \alpha + \alpha^3$	(1 1 0 1)
α^8	$1 + \alpha^2$	(1 0 1 0)
α^9	$\alpha + \alpha^3$	(0 1 0 1)
α^{10}	$1 + \alpha + \alpha^2$	(1 1 1 0)
α^{11}	$\alpha + \alpha^2 + \alpha^3$	(0 1 1 1)
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	(1 1 1 1)
α^{13}	$1 + \alpha^2 + \alpha^3$	(1 0 1 1)
α^{14}	$1 + \alpha^3$	(1 0 0 1)

Table 2: List of EG-LDPC codes

n	k	d_{min}	ρ	γ	r
15	7	5	4	4	0.267
63	37	9	8	8	0.127
255	175	17	16	16	0.0627
1023	781	33	32	32	0.0313
4095	3367	65	64	64	0.01563
16383	14197	129	128	128	0.007813

Table 3: List of PG-LDPC codes

n	k	d_{min}	ρ	γ	r
21	11	6	5	5	0.2381
73	45	10	9	9	0.1233
273	191	18	17	17	0.0623
1057	813	34	33	33	0.0312
4161	3431	66	65	65	0.0156
16513	14326	130	129	129	0.0078

$$\mathbf{H} = \begin{bmatrix}
0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1 \\
1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0 \\
0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0 \\
0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0 \\
0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1 \\
1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0 \\
0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1 \\
1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1 \\
1\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\
0\,1\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\
0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0 \\
0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \\
0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0 \\
0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,0\,0 \\
0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,0
\end{bmatrix}$$

Figure 1: A low density parity check matrix.

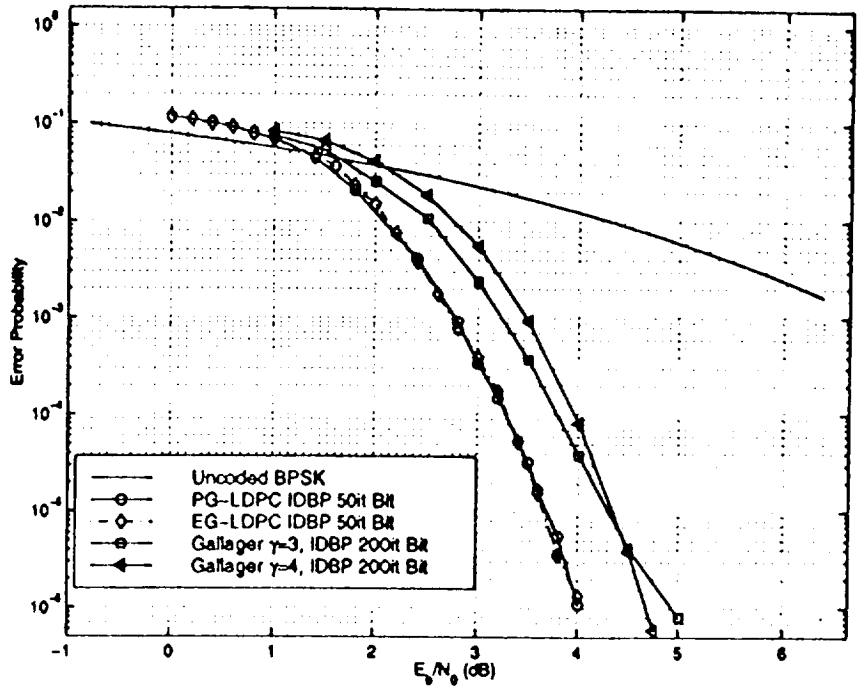

Figure 2: Bit-error probabilities of the (255, 175) EG-LDPC code (maximum 50 iterations), (273,191) PG-LDPC code(maximum 50 iterations) and two (273,191) Gallager codes generated by computer (maximum 200 iterations) based on IDBP algorithm.

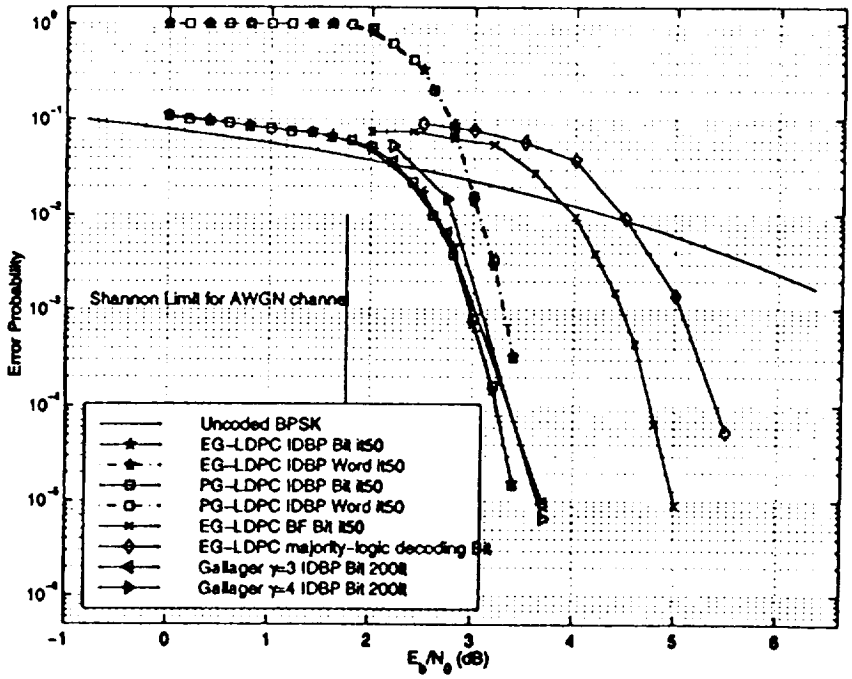

Figure 3: Bit- and word-error probabilities of the (1023,781) EG-LDPC code, the(1057,813) PG-LDPC code and two (1057,813) Gallager codes generated by computer .

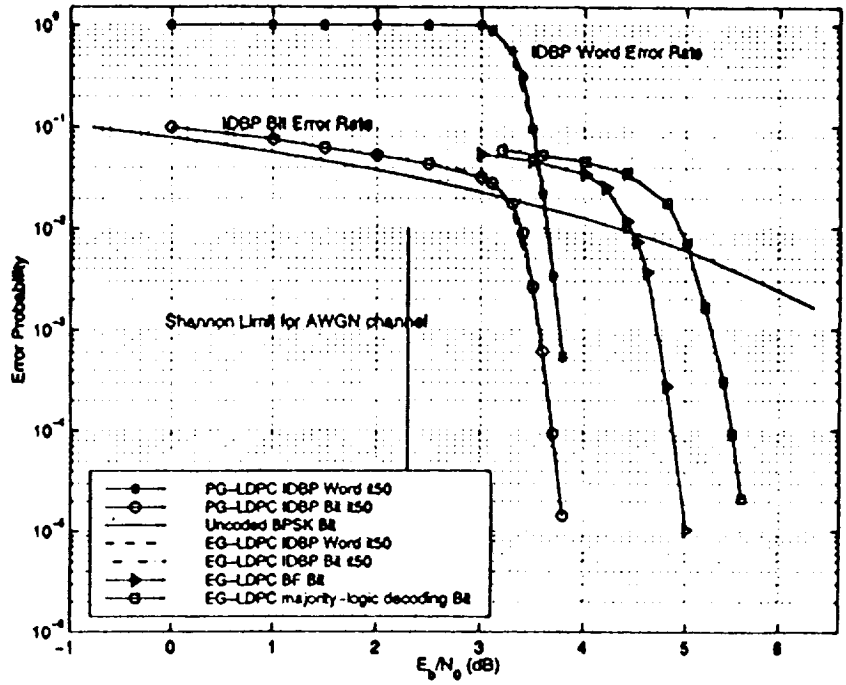

Figure 4: Bit- and word-error probabilities of the (4095,3367) EG-LDPC code and the(4161,3431) PG-LDPC code.

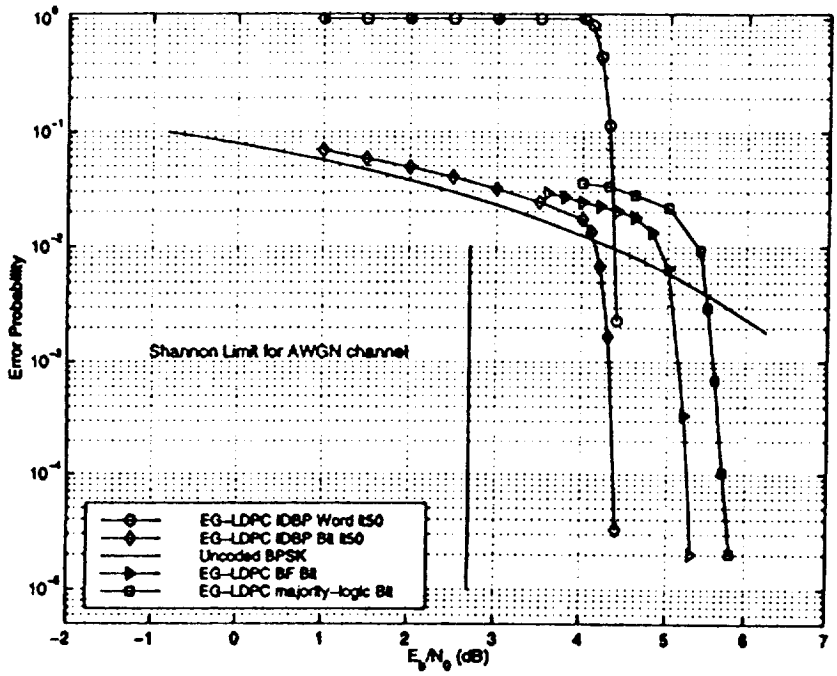

Figure 5: Bit- and word-error probabilities of the (16383,14179) EG-LDPC code.

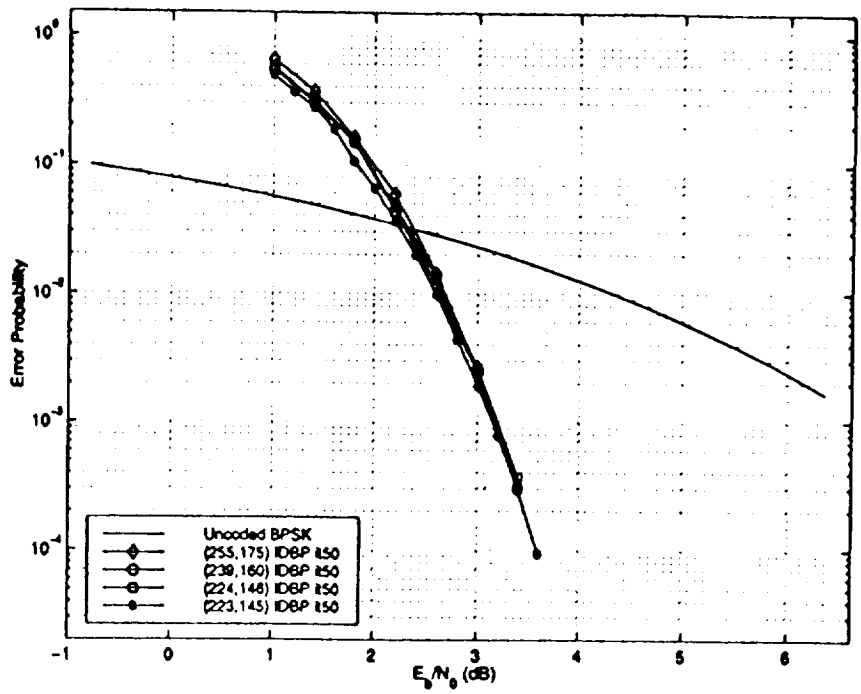

Figure 6: Bit-error probabilities of the (255,175) EG-LDPC code and its (239,160), (223,145) and (224,146) punctured codes.

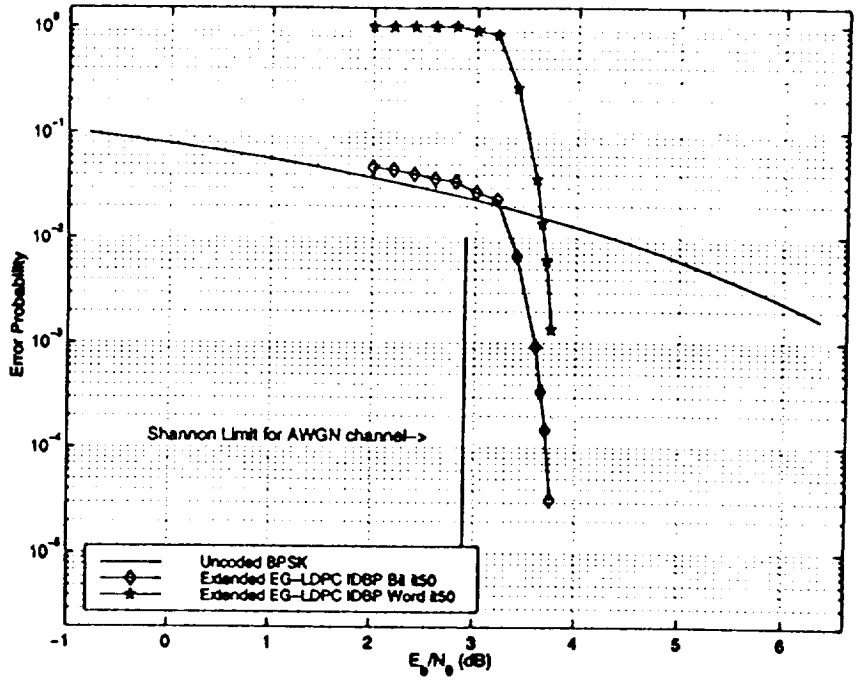

Figure 7: Bit- and word-error probabilities of the (8184,7162) LDPC code obtained by extending the (1023,781) EG-LDPC code using column splitting with q=8.

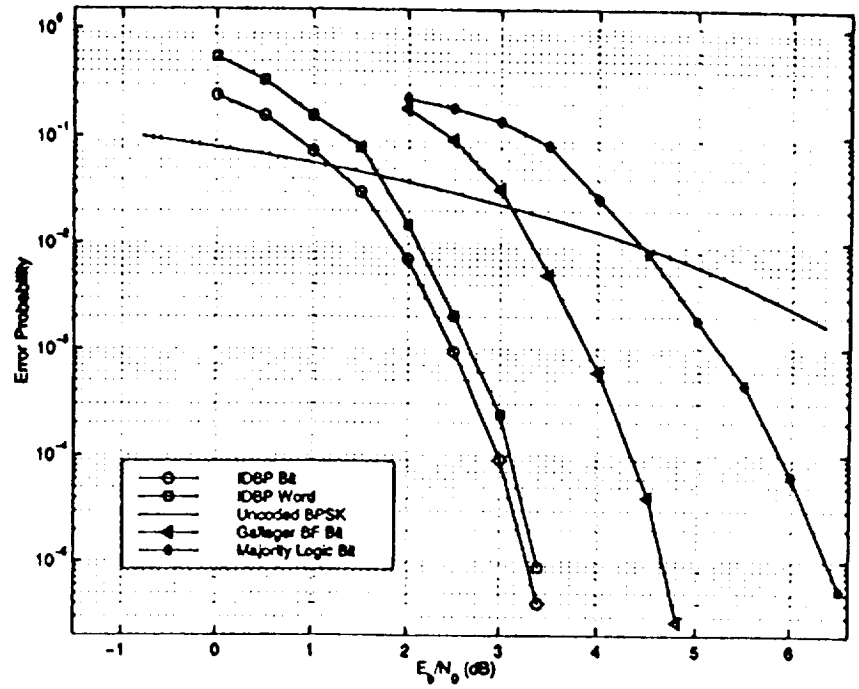

Figure 8: Bit- and word-error probabilities of the (511,139) EG-LDPC code.

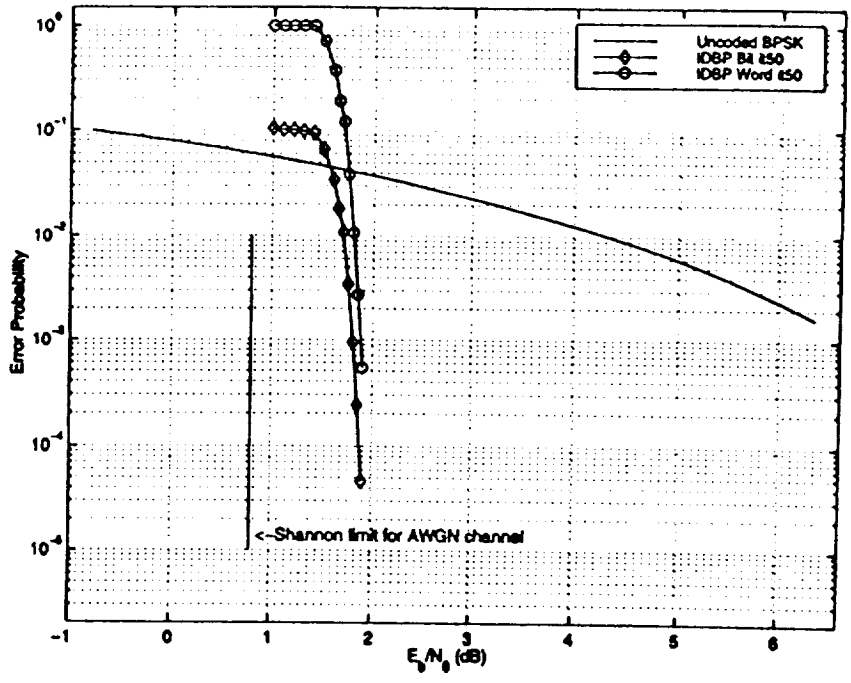

Figure 9: Bit- and word-error probabilities of the (12264,7665) LDPC code obatined by extending the (511,139) EG-LDPC code by usuing column splitting with q=8.

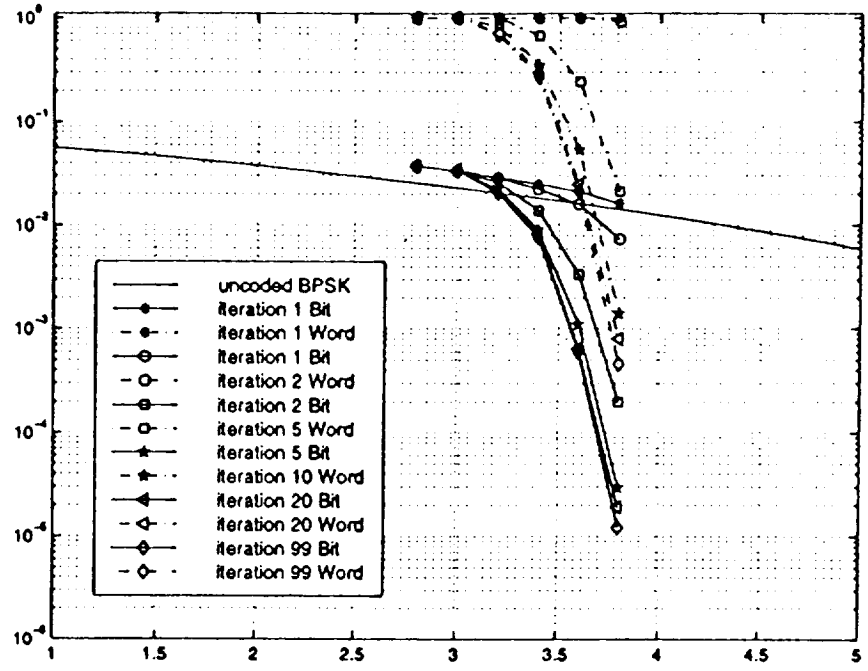

Figure 10: Convergence of the IDBP algorithm for the (4161,3431) PG-LDPC code.