On a Class of High-Girth LDPC Codes Based on Finite Multidimensional Lattices

John T. Craddock[1], Mark F. Flanagan[2], Anthony D. Fagan[3]

Department of Electronic and Electrical Engineering, University College Dublin, Belfield, Dublin 4, Ireland

[1] john.craddock@ee.ucd.ie
[2] mark.flanagan@ieee.org
[3] tony.fagan@ucd.ie

Abstract

An LDPC code construction technique is proposed based on the structural properties of finite m-dimensional lattices. The Tanner graph of any code from this class is shown to have a girth of eight, and the number of proper eight-cycles in the graph is enumerated. The minimum distance of the codes is shown to be lower bounded by 2^m. The codes are also shown to be highly flexible in terms of code length and rate, and compatible with a low-complexity serial-parallel decoder implementation based on the turbo-decoding message passing algorithm. Finally, simulation results over the AWGN channel demonstrate that these codes have good error-correcting performance.

1 Introduction

Low-density parity check (LDPC) codes, first introduced in [1], are a class of linear block code which have been shown to achieve near-capacity performance on the AWGN channel when decoded using the 'belief propagation' (or 'message passing') algorithm. The first LDPC codes studied were based on pseudorandom constructions [1], [2]; however recently there has been much interest in LDPC codes based on algebraic and combinatorial constructions, for two main reasons. First, the underlying structure (algebraic or combinatorial) often means that it is possible to prove that the derived codes have certain desirable properties (e.g. good girth and minimum distance); second, the structural regularity of these codes often leads to low-complexity implementations for the LDPC decoder. Previously it has been difficult to reap these benefits in a joint manner while retaining a wide choice of code parameters (length and rate).

Belief propagation is known to be BER-optimum when the Tanner graph is a tree (contains no cycles). It has also been shown that short cycles in the Tanner graph inhibit the performance of the decoding algorithm by introducing interdependence between messages [3]. Much research effort, therefore, has gone into the area of LDPC code design with high girth in the Tanner graph (an excellent overview is given in [4]). Our work in [5] proposed a generalization of the Finite Geometry LDPC codes of [6], [7], [8] to a class of girth eight LDPC codes. This paper may be regarded as an extension of the work of [5] to multidimensional lattice codes. This extension is not trivial since finite multidimensional lattices do not possess as much algebraic structure as geometries; for example, they are not vector spaces, in general. However, as we show in the paper, the class of LDPC codes derived from finite multidimensional lattices possesses much more flexibility in terms of code length and rate. We note also that this work may be viewed as a generalization of the two-dimensional lattice LDPC codes described in [9].

2 Mathematical Preliminaries

2.1 Multidimensional Lattices and their Properties

We define the infinite m-dimensional lattice $\mathcal{L}^i$ as

$$\mathcal{L}^i = \mathbb{Z} \times \mathbb{Z}_{p_2} \times \mathbb{Z}_{p_3} \times \cdots \times \mathbb{Z}_{p_m}$$

where $p_2, p_3, \ldots p_m$ are positive integers, i.e. $\mathcal{L}^i$ is the set of m-tuples $(x_1, x_2, x_3, \ldots x_m)$ such that each coordinate $i \neq 1$ is reduced modulo p_i. We shall refer to the elements of the lattice as *points*. We define addition of elements of $\mathcal{L}^i$ as simply a coordinate-wise addition; this is an integer addition in the first coordinate and an addition modulo p_i in coordinate $i \neq 1$. We also define multiplication of elements of $\mathcal{L}^i$ by integers (scalars) as simple coordinate-wise integer multiplication, with reduction modulo p_i in the i^{th} coordinate, $i \neq 1$. We also define the finite m-dimensional lattice $\mathcal{L}$ as

$$\mathcal{L} = \left\{ x = (x_1, x_2, x_3, \ldots x_m) \in \mathcal{L}^i | 0 \leq x_1 \leq c-1 \right\}$$

where c is a positive integer and we assume $c \leq \min\{p_2, p_3, \ldots p_m\}$. The finite lattice $\mathcal{L}$ contains $c \times p_2 \times p_3 \times \ldots \times p_m$ points. The all-zero m-tuple shall be referred to as the *origin* of both $\mathcal{L}$ and $\mathcal{L}^i$ and shall be denoted o. It is worth noting that neither $\mathcal{L}$ nor $\mathcal{L}^i$ is a vector space, since $\mathbb{Z}$ is not a field. However, we

shall take a notational liberty and define a notion of linear dependence of a set of points as follows. For this definition we let $\mathbb{Z}^{\pm c}$ denote the set of integers j with $|j| < c$.

Definition 1: A subset $\{z_1, z_2, \ldots z_r\}$ of $\mathcal{L}^i$ is said to be *linearly dependent* if there exist integers $\alpha_1, \alpha_2, \ldots \alpha_r \in \mathbb{Z}^{\pm c}$, not all zero, such that

$$\sum_{i=1}^{r} \alpha_i z_i = o$$

Otherwise the set $\{z_1, z_2, \ldots z_r\}$ is said to be *linearly independent*. Also, we refer to the expression $\sum_{i=1}^{r} \alpha_i z_i$ as a *linear combination* of the points $z_1, z_2, \ldots z_r$.

2.2 Lines and Slopes in the Lattice

For any point $x \in \mathcal{L}$, we call x a *base point* iff $x_1 = 0$, i.e. iff

$$x = (0, x_2, x_3, \ldots x_m)$$

We also call x a *slope* iff $x_1 = 1$, i.e. iff

$$x = (1, x_2, x_3, \ldots x_m)$$

Now, for any base point $v \in \mathcal{L}$ and for any slope $s \in \mathcal{L}$, we define the line through v with slope s as follows.

$$L_v(s) = \{v + \alpha s | 0 \leq \alpha \leq c - 1\}$$

By considering the first coordinate, it is easy to see that this line contains c distinct points, and is contained in $\mathcal{L}$. Each slope $s \in \mathcal{L}$ induces a partition of the finite lattice $\mathcal{L}$ into $p_2 \times p_3 \ldots \times p_m$ disjoint subsets, i.e. the lines with slope s.

Theorem 1: Two points $x, y \in \mathcal{L}$ lie on a line with slope s if and only if

$$x - y = ks$$

for some $k \in \mathbb{Z}^{\pm c}$.

Proof: First, let $x, y \in \mathcal{L}$ and let $x, y \in L_v(s)$. Then

$$x = v + k_1 s \quad \text{for some } 0 \leq k_1 < c$$

$$y = v + k_2 s \quad \text{for some } 0 \leq k_2 < c$$

so

$$x - y = (k_1 - k_2)\, s$$

thus proving the first part of the theorem. Secondly, let $x, y \in \mathcal{L}$ and suppose

$$x - y = ks$$

for some $k \in \mathbb{Z}^{\pm c}$. If the first coordinate of x is x_1 and the first coordinate of y is y_1, then we must have $x_1 - y_1 = k$ and so

$$x - y = (x_1 - y_1)\, s \tag{1}$$

Also note that since $x, y \in \mathcal{L}$, both x_1 and y_1 lie in the range $0, 1, \ldots c - 1$. Now let $v = x - x_1 s$. It follows from (1) that $v = y - y_1 s$. By considering the first coordinate, it is easy to see that v is a base point. So we have $x = v + x_1 s$ and $y = v + y_1 s$, and the result is proved. ■

3 LDPC Code Class: Construction and Properties

A low-density parity-check code is defined as the null space of a $q \times n$ parity check matrix $\mathbf{H}$, i.e. the set of codewords $\mathbf{x}$ which satisfy $\mathbf{H}\mathbf{x}^{\mathbf{T}} = \mathbf{0}$. The term "low-density" refers to the fact that the parity check matrix is sparse. Associated with any LDPC code is a bipartite graph called the Tanner graph. This consists of n bit (or variable) nodes, corresponding to the code bits, and q check (or factor) nodes, corresponding to the parity checks. Check node i is connected to bit node j if and only if $H_{ij} = 1$. Decoding of an LDPC code typically uses a two-phase message passing (TPMP) algorithm on the Tanner graph, whose two phases are: message passing from bit nodes to check nodes, and vice versa [1]. This was also shown to be an instance of a more general message passing algorithm which operates on a more general form of a Tanner graph known as a factor graph [10], [11].

An LDPC code may be constructed using the finite multidimensional lattice $\mathcal{L}$ as follows. Let $B = \{s_1, s_2, \ldots s_\rho\}$ be a ρ-element set of slopes in the lattice satisfying the following two conditions:

Condition 1: Any η-element subset A of B is linearly independent.
Condition 2: B contains at least m linearly independent slopes.

Here η is a design parameter and is chosen to take the value 3 or 4.

Then, a set of lines X is defined as the following disjoint union:

$$X = \bigcup_{i=1}^{\rho} X(s_i)$$

where $X(s_i)$ denotes the set of all lines with slope s_i. Finally, a parity check matrix $\mathbf{H}$ is constructed such that the rows of $\mathbf{H}$ correspond to the lines in X and the columns of $\mathbf{H}$ correspond to the points in $\mathcal{L}$. The (i, j) entry in $\mathbf{H}$ is equal to 1 if the i^{th} line contains the j^{th} point, and is equal to 0 otherwise. The resulting parity check matrix has $q = \rho \times p_2 \times p_3 \times \ldots \times p_m$ rows and $n = c \times p_2 \times p_3 \times \ldots \times p_m$ columns. Note that the parity check matrix may be written as the vertical concatenation of the matrices $\mathbf{H_1}, \mathbf{H_2}, \ldots \mathbf{H}_\rho$, where the parity checks in $\mathbf{H_i}$ correspond to the lines in $X(s_i)$. Note also that in the code's Tanner graph, bit nodes correspond to points in the geometry, while check nodes correspond to the lines in X.

We refer to these codes as finite multidimensional lattice (FML) LDPC codes. In this paper, we prove some results on cycles in the Tanner graphs of these

FML codes. To this end we define a *proper* $2r$-cycle to be a $2r$-cycle which consists of $2r$ *distinct* vertices (because the Tanner graph is bipartite, all cycle lengths are even). From the point of view of message-passing decoding, only proper cycles are of interest; short proper cycles in the Tanner graph lead to performance degradation of LDPC decoding through introduction of interdependence between messages [3].

Theorem 2: The FML LDPC codes are regular Gallager codes, and their Tanner graphs contain no proper four-cycles.

Proof: First, each line contains c points, and each point is contained in ρ lines. Therefore, the codes are (ρ, c)-regular. Now, suppose two points x, y lie on two distinct lines, then we must have

$$x - y = \alpha s_1$$

and

$$x - y = \beta s_2$$

for some $\alpha, \beta \in \mathbb{Z}^{\pm c}$ and for some $s_1, s_2 \in B$ ($s_1 \neq s_2$). So

$$\alpha s_1 = \beta s_2 \tag{2}$$

but this is impossible by Condition 1. ■

Note that (2) implies $\beta = \alpha$ (check first coordinate), and is therefore equivalent to the condition

$$\alpha\left(s_{2i} - s_{1i}\right) \equiv 0 \pmod{p_i} \qquad \forall i = 2, 3, \ldots m \tag{3}$$

where s_{1i} and s_{2i} denote the i^{th} entries of s_1 and s_2 respectively. Since $0 < \alpha < c$ and $c \leq p_i$ for all i, condition (3) may be avoided simply by ensuring that for every pair of slopes $s_1, s_2 \in B$, the point $s_2 - s_1$ contains at least one entry relatively prime to its corresponding modulus.

Theorem 3: The Tanner graph of an FML code contains no proper six-cycles.

Proof: Suppose the Tanner graph contains a proper six-cycle. Then there must exist three distinct points $x, y, z \in \mathcal{L}$ and three distinct slopes $s_1, s_2, s_3 \in \mathcal{L}$ such that

$$x - y = \alpha s_1$$

$$y - z = \beta s_2$$

$$z - x = \gamma s_3$$

for some nonzero integers $\alpha, \beta, \gamma \in \mathbb{Z}^{\pm c}$. It follows that $\alpha s_1 + \beta s_2 + \gamma s_3 = o$; but this is impossible by Condition 1. ■

Theorem 4: The number of proper eight-cycles in the Tanner graph of an FML code with $\eta = 4$ is given by

$$N_{FML}^{(8)} = n\rho\left(\rho - 1\right)\left(c - 1\right)^2/8 \tag{4}$$

Proof: A variant of the proof of ([5], Theorem 3) can be used to prove this result; however we provide here an alternative method of proof. Assume that there exists a path of length four in the Tanner graph from some variable node x to another variable node y, passing through some variable node z. Then we must have

$$z = x + \alpha s_1$$

and

$$y = z + \beta s_2$$

for some $\alpha, \beta \in \mathbb{Z}^{\pm c}$ and for some $s_1, s_2 \in B$ ($s_1 \neq s_2$). Suppose now that there exists another (distinct) path of length four in the Tanner graph from x to y. This occurs if and only if there exists $w \neq z$ such that

$$w = x + \gamma s_3$$

and

$$z = w + \delta s_4$$

for some $\gamma, \delta \in \mathbb{Z}^{\pm c}$ and for some $s_3, s_4 \in B$. Now, this implies

$$\alpha s_1 + \beta s_2 - \gamma s_3 - \delta s_4 = o$$

and by Condition 1 (recall $\eta = 4$), this equation may only be satisfied if γs_3 and δs_4 are equal to αs_1 and βs_2 in some order. This yields exactly one solution with $w \neq z$, i.e. $\gamma = \beta$, $s_3 = s_2$, $\delta = \alpha$, $s_4 = s_1$. Note that in this solution, w and z cannot be equal because this would imply $\alpha s_1 = \beta s_2$, contradicting Condition 1. Now, we draw a section of the Tanner graph of the code as a rooted tree with some variable node y as root, and consider only the first four tiers T_1, T_2, T_3, T_4 (y itself constitutes tier T_0, and each tier T_i consists of the neighbours to the nodes in T_{i-1} - see figure 1). The first tier T_1 consists of ρ factor nodes. The second tier T_2 consists of $\rho\left(c - 1\right)$ variable nodes; these must all be distinct since the Tanner graph has no cycles of length four. The third tier T_3 consists of $\rho\left(\rho - 1\right)\left(c - 1\right)$ factor nodes; these must all be distinct since the Tanner graph has no cycles of length six. Finally, the fourth tier T_4 consists of $\rho\left(\rho - 1\right)\left(c - 1\right)^2$ variable nodes. By the preceding argument, these may be split into pairs such that each pair corresponds to the same variable. Since each pair corresponds to a proper eight-cycle, the number of such cycles is

$$N_{FML}^{(8)} = n\rho\left(\rho - 1\right)\left(c - 1\right)^2/8 \tag{5}$$

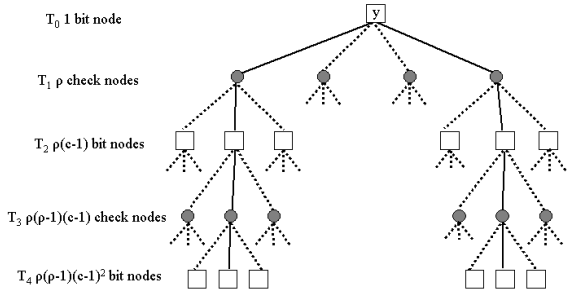

Fig. 1. Section of the Tanner graph drawn as a local rooted tree with variable y as root. This is used in the proof of theorem 4.

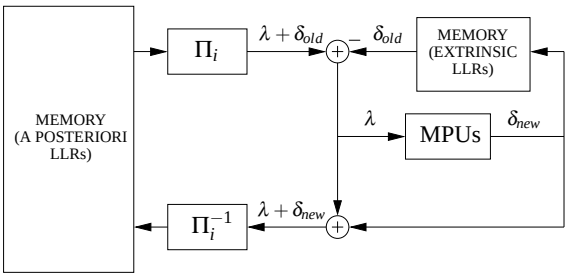

Fig. 2. Partly parallel architecture for decoding of FML LDPC codes.

The denominator corresponds to a division by two, corresponding to the pairs, and a division by four, since each eight-cycle is counted four times. ■

Theorem 5: The minimum distance of the FML LDPC code is bounded by $d_{min} \geq 2^m$.

Proof: By condition 2, B contains m linearly independent slopes; call these $s_1, s_2, \ldots s_m$. Consider the minimum weight codeword. This has positive weight; so some point x corresponds to a bit with value 1. Now draw the line $L_x(s_1)$; this line corresponds to a satisfied parity check, so it contains a point $y \neq x$ which corresponds to a bit with value 1. Now draw $L_x(s_2)$ and $L_y(s_2)$; these lines must contain points w and z respectively which do not lie on $L_x(s_1)$. These points must again correspond to bits with value 1. We continue this argument, at each stage doubling the number of points; the new points are not on the lines already drawn since this would contradict the linear independence of $s_1, s_2, \ldots s_m$. This proves that this codeword must have weight at least 2^m. ■

4 Partly Parallel LDPC Decoder Implementation

A) Decoder Implementation Overview

The decoder architecture discussed in this section is based upon ideas presented in [12]. Recall that for any i, the rows in the i^{th} block row $\mathbf{H_i}$ of the parity check matrix do not overlap and thus may be decoded in parallel. This leads to a partly parallel decoder architecture depicted in figure 2. The decoder operates on a blockrow-by-blockrow basis and therefore contains $p_2 \times p_3 \times \ldots \times p_m$ message processing units (MPUs) corresponding to check nodes for the parity checks in $\mathbf{H_i}$. The *a posteriori* LLRs for the code bits are stored in memory in the natural order. These are interleaved by the interleaver Π_i corresponding to the particular ordering of the 1's in submatrix $\mathbf{H_i}$, and then extrinsic LLRs δ_{old} from the previous MPU output (LLRs corresponding to messages pending at the bit nodes from the check nodes in $\mathbf{H_i}$) are subtracted to form the extrinsic input λ to the MPUs (see also the upper half of figure 3). The new MPU outputs δ_{new} are used to update the extrinsic LLR memory. These MPU outputs are also added to the MPU inputs, and the result is deinterleaved and stored in memory as the updated *a posteriori* LLRs for the code bits. This procedure constitutes the i^{th} subiteration; its application for each $i = 1, 2, \ldots \rho$ completes one iteration of LDPC decoding. Decisions may be made on the *a posteriori* LLRs after any number of these iterations.

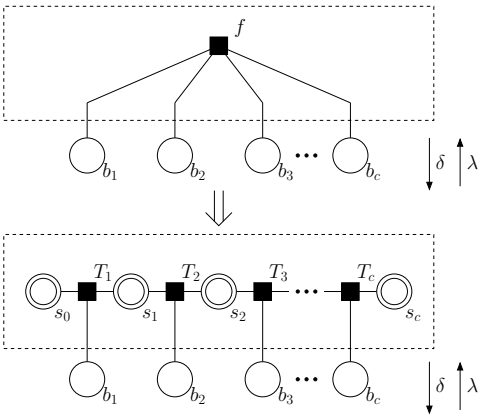

Fig. 3. Factor graph transformation to facilitate the TDMP algorithm for decoding of FML LDPC codes.

B) Trellis-Based LDPC Decoding Algorithm

This section describes an alternative implementation for the MPUs discussed in the previous section. Here the MPU functionality is implemented using the turbo decoding message passing (TDMP) algorithm, first proposed in [12]. Suppose f is a check node of the Tanner graph connected to bit nodes $b_1, b_2, \ldots b_c$. The parity-check equation $b_1 \oplus b_2 \oplus \ldots \oplus b_c = 0$ may be written in terms of a binary-valued "state" variable s_k as follows: $s_0 = 0$, and for each $k = 1, 2, \ldots c$, $s_k = s_{k-1} \oplus b_k$. Thus $s_c = 0$ and the factor f may be further factored as $f = \prod_{k=1}^{c} T_k(s_k, s_{k-1}, b_k)$, where T_k denotes the indicator function $[s_k = s_{k-1} \oplus b_k]$. This factorization of f leads to the factor graph transformation depicted in figure 3. Computationally, this transformation replaces the factor node computation of [1] by a log-APP algorithm [13] which executes on the two-state trellis. Furthermore, it is shown in [12] that the computations may be accomplished using only metric differences which may be implemented using simple "max-quartet" operations. The TDMP algorithm has been shown to yield lower memory requirements and faster decoding convergence [12].

5 Flexibility of Code Length and Rate

The length of a codeword is given by $n = c \times p_2 \times p_3 \times \ldots \times p_m$ and the number of parity checks in the code is given by $q = \rho \times p_2 \times p_3 \times \ldots \times p_m$; thus $q = \rho n/c$ and therefore the rate r of the code is lower bounded by $r \geq (c - \rho)/c$, with equality iff all the rows of H

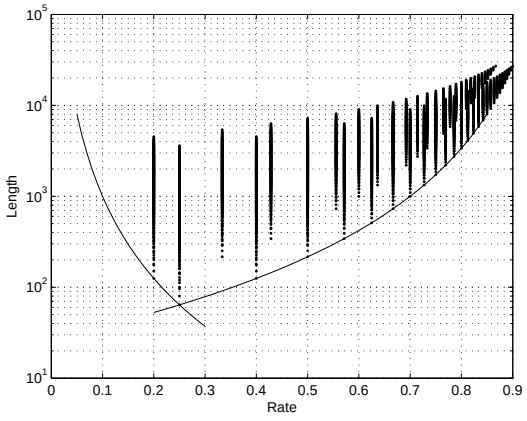

Fig. 4. Achievable length and rate with three-dimensional finite lattice LDPC codes.

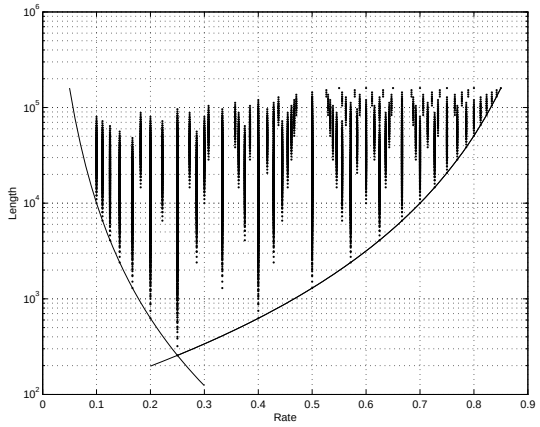

Fig. 5. Achievable length and rate with four-dimensional finite lattice LDPC codes.

are linearly independent.

Lattice construction gives rise to a large family of codes. As there are no restrictions placed on the values $c, p_2, p_3, \ldots, p_m$ other than that they are greater than or equal to 4, a large number of values of n are achievable. Once a designer has chosen values for c and $p_2, p_3, \ldots, p_m$ a rate can be chosen by selecting a value of ρ. The length versus rate characteristics of codes derived from 3 and 4 dimensional lattices are given in figures 4 and 5, respectively.

Theorem 6: The codeword lengths n achievable at a given rate r and for a given lattice dimension m are approximately lower bounded by $n \geq 1/r^m$ and $n \geq 3^m/(1-r)^m$.

Proof: If we assume that almost all parity checks are linearly independent, then the rate of the code is given by $r \approx (c-\rho)/c$. Good LDPC codes require a column weight of at least three, therefore $\rho \geq 3$. The maximum rate achievable for a given value of c is achieved by setting $\rho = 3$ giving $r \leq (c-3)/c$, or equivalently $c \geq 3/(1-r)$. The minimum codeword length is given by $n \geq c^m$. Combining these inequalities gives $n \geq c^m \geq 3^m/(1-r)^m$. The minimum non-zero rate achievable at a given value of c is achieved by setting $\rho = c-1$ giving a rate of $r = 1/c$. Combining this with $n \geq c^m$ gives $n \geq 1/r^m$. These bounds are illustrated in figures 4 and 5. ■

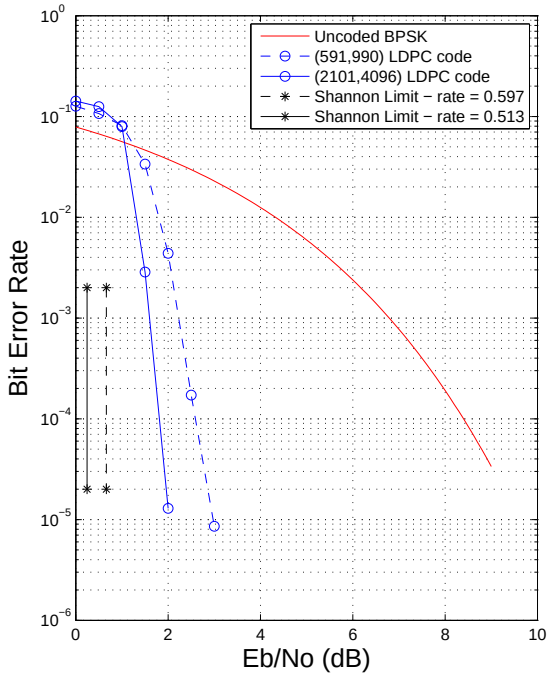

Fig. 6. BER and FER performance of the two example LDPC codes generated from finite multidimensional lattices. 20 iterations of the partly parallel decoder with the turbo decoding message passing algorithm are used in decoding.

6 Error Correcting Performance

In this section, error correction capability is demonstrated for two example girth-8 FML codes. BPSK transmission over the AWGN channel was assumed, and in each case decoding continued until either a valid codeword was detected (via syndrome check) or a maximum of 20 iterations were completed.

The first code was constructed from a three-dimensional finite lattice with $c = 9$, $p_2 = 10$ and $p_3 = 11$. A slope set $B = \{s_1, s_2, s_3, s_4\} = \{(1,0,0), (1,1,0), (1,0,1), (1,1,1)\}$ of size $\rho = 4$ was used to generate parity checks; this slope set has $\eta = 3$. It may easily be checked that B satisfies Condition 1; for example, $\alpha s_1 + \beta s_2 + \gamma s_3 = 0$ implies $\alpha + \beta + \gamma = 0$, p_2 divides β and p_3 divides γ, which can only hold for $\alpha = \beta = \gamma = 0$ since $|\beta|, |\gamma| < c \leq p_2, p_3$. The linear independence of the other 3-element subsets of B may be shown in a similar manner. The resulting LDPC code has length 990 and consists of 440 parity checks 399 of which are linearly

independent, thus giving this code an approximate rate of 0.6.

The second code was constructed from a four-dimensional finite lattice with $c = p_2 = p_3 = p_4 = 8$. A slope set $B = \{s_1, s_2, s_3, s_4, s_5\} = \{(1,0,0,0), (1,1,0,0), (1,0,1,0), (1,0,0,1), (1,1,1,1)\}$ of size $\rho = 4$ was used to generate parity checks; this slope set has $\eta = 4$. Again it may be checked that this set B meets Condition 1. The resulting LDPC code has length 4096 and consists of 2560 parity checks 1995 of which are linearly independent, yielding an approximate code rate of 0.5.

The bit error rate (BER) performance of these codes is shown in figure 6. The curves shown are for the practical low-complexity serial-parallel implementation with the turbo decoding message passing algorithm. Both sample codes are seen to exhibit steep 'waterfall' effects near to their respective Shannon limits, with a steepening of the waterfall and a tightening of the gap to capacity as the code length increases.

7 Conclusion

A class of LDPC codes based on finite multidimensional lattices has been proposed. This code class was shown to exhibit many desirable properties, including: a girth of eight in the Tanner graph; an exact enumeration of the number of proper eight-cycles in the graph; a lower bound on the code minimum distance; good flexibility of code parameters (code length and code rate); low implementation complexity; and finally, good error-correcting performance.

References

[1] R. G. GALLAGER, *Low density parity check codes, IRE Transactions on Information Theory,* vol. IT-8, pp. 21–28, January 1962.

[2] D. J. C. MACKAY, *Good error correcting codes based on very sparse matrices, IEEE Transactions on Information Theory,* vol. 45, pp. 399–431, March 1999.

[3] 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 in Communications,* vol. 16, pp. 140–152, February 1998.

[4] J. M. F. MOURA, J. LU, AND H. ZHANG, *Structured low-density parity-check codes, IEEE Signal Processing Magazine,* pp. 42–55, January 2004.

[5] M. F. FLANAGAN, J. CRADDOCK, C. P. FEWER, AND S. J. REDMOND, *A Euclidean geometry based algebraic construction technique for girth-8 Gallager LDPC codes,*
Proc. IEEE Information Theory Workshop, ITW '06, 22–26 October 2006.

[6] Y. KOU, S. LIN, AND M. P. C. FOSSORIER, *Low density parity check codes based on finite geometries: A rediscovery and new results, IEEE Transactions on Information Theory,* vol. 47, pp. 2711–2736, November 2001.

[7] H. TANG, J. XU, Y. KOU, S. LIN, AND K. A. S. ABDEL-GHAFFAR, *On algebraic construction of Gallager and circulant low-density parity-check codes, IEEE Transactions on Information Theory,* vol. 50, pp. 1269–1279, June 2004.

[8] H. TANG, J. XU, S. LIN, AND K. A. S. ABDEL-GHAFFAR, *Codes on finite geometries, IEEE Transactions on Information Theory,* vol. 51, pp. 572–596, February 2005.

[9] B. VASIC AND O. MILENKOVIC, *Combinatorial constructions of low-density parity-check codes for iterative decoding, IEEE Transactions on Information Theory,* vol. 50, pp. 1156–1176, June 2004.

[10] S. M. AJI AND R. J. MCELIECE, *The generalized distributive law, IEEE Transactions on Information Theory,* vol. 46, pp. 325–343, March 2000.

[11] F. R. KSCHISCHANG, B. J. FREY, AND H-A. LOELIGER, *Factor graphs and the sum-product algorithm, IEEE Transactions on Information Theory,* vol. 47, pp. 498–519, February 2001.

[12] M. M. MANSOUR AND N. R. SHANBHAG, *High-throughput LDPC decoders, IEEE Transactions on Very Large Scale Integrated Systems,* vol. 11, pp. 976–996, December 2003.

[13] L. R. BAHL, J. COCKE, F. JELINEK, AND J. RAVIV, *Optimal decoding of linear codes for minimizing symbol error rate, IEEE Transactions on Information Theory,* vol. 20, pp. 284–287, March 1974.