A Euclidean Geometry Based Algebraic Construction Technique for Girth-8 Gallager LDPC Codes

Mark F. Flanagan, John Craddock, Colm P. Fewer and Stephen J. Redmond
Dept. of Electronic and Electrical Engineering
University College Dublin
{mark.flanagan, john.craddock, cfewer, stephen.redmond}@ee.ucd.ie

Abstract — **A construction technique is proposed for low-density parity-check (LDPC) codes based on finite Euclidean geometries $EG(m, 2^s)$. These codes are shown to be regular Gallager codes with Tanner graphs of girth eight. The minimum distance of these codes is shown to be lower-bounded by 2^m. The codes are also amenable to an efficient partly parallel decoder implementation, which may be used in conjunction with the turbo decoding message passing (TDMP) algorithm for LDPC decoding. Finally, simulation results show that these codes have very good error-correcting performance.**

I. Introduction

Low-density Parity Check (LDPC) codes, first proposed 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 sum-product algorithm (SPA). Algebraic and combinatorial constructions of LDPC codes have received much attention over the past decade since such constructions generally yield codes with lower implementation complexity than do pseudorandom constructions [2, 3]. Of these, codes based on finite geometries have exhibited performance closest to capacity. Finite geometry codes are a well-established class of linear block code, where points of the geometry are associated with the code bits, and lines of the geometry are associated with parity checks. These codes are traditionally decoded using majority-logic [4]; however, in [5], these codes were shown to be LDPC codes which yield high performance using sum-product decoding. This class of LDPC codes was extended in [6] by selecting for parity checks only bundles of parallel lines in the geometry. Comprehensive results on both the original and extended class of finite geometry LDPC codes were recently presented in [7].

A shortcoming of [6] and [7] is that the bundles of parallel lines were chosen at random, i.e. no criterion was offered for their choice. In this paper we propose a judicious choice of parallel line sets which benefits the code structure in two ways: firstly, the resulting code Tanner graph has a girth of 8, and secondly, the minimum distance of the code is bounded below by 2^m. Another shortcoming of [6] and [7] is that a fully parallel SPA decoder architecture was assumed. It is well known that fully parallel decoder implementations are prone to routing congestion problems. In this paper we describe a more realistic partly parallel decoder architecture which uses the turbo decoding message passing algorithm for LDPC decoding. Significantly, we provide code performance results for the proposed implementation as well as for the parallel SPA decoder implementation.

The paper is organized as follows. Section II provides a unified framework for finite Euclidean geometry codes and the proposed codes, and proves results on the benefits of the proposed codes. Section III proves a result on the sparsity of proper eight-cycles in the proposed codes. Section IV describes the partly parallel decoder architecture, section V presents simulation results, and section VI concludes this work.

II. Euclidean Geometry Codes and Reduced Euclidean Geometry Codes

A low-density parity-check code is defined as the null space of an $q \times n$ 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 nodes, corresponding to the bits, and q check 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 recently 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 [8, 9].

An m-dimensional finite Euclidean geometry (EG) over the field $GF(2^s)$, denoted $EG(m, 2^s)$, may be represented as a set of $n = 2^{ms}$ m-tuples $(\alpha_1, \alpha_2, \ldots \alpha_m)$ where $\alpha_i \in GF(2^s)$ for each $i = 1, 2, \ldots m$. The all-zero m-tuple is called the *origin* of the geometry and is denoted o. The set of m-tuples, called *points* of the geometry, form a vector space over $GF(2^s)$ according to the usual rules of entry-wise addition and multiplication for m-tuples. For any point $x \in EG(m, 2^s)$ and any nonorigin point $v \in EG(m, 2^s)$, the line through x parallel to v

is comprised of $\rho = 2^s$ points:

$$L_x(v) = \{x + \beta v | \beta \in GF(2^s)\}$$

Any nonorigin point $v \in EG(m, 2^s)$ induces a partition of $EG(m, 2^s)$ into $2^{(m-1)s}$ subsets, i.e. the lines parallel to v. The proof of this statement is trivial. It is easy to show that two points x and y lie on the same line parallel to v if and only if $x + y = \beta v$ for some $\beta \in GF(2^s)$. This statement will be important in proofs in this paper.

An LDPC code may be constructed using the geometry $EG(m, 2^s)$ as follows. First, a set of nonorigin points $B = \{v_1, v_2, \ldots v_l\} \subset EG(m, 2^s)$ is chosen. Then, a set of lines S is defined as the following disjoint union:

$$S = \bigcup_{i=1}^{l} S(v_i)$$

where $S(v_i)$ denotes the set of all lines parallel to v_i. Finally, a parity check matrix $\mathbf{H}$ is constructed such that the rows of $\mathbf{H}$ correspond to the lines in S and the columns of $\mathbf{H}$ correspond to the points in $EG(m, 2^s)$. 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 $l \cdot 2^{(m-1)s}$ rows and 2^{ms} columns. Note that the parity check matrix may be written as the vertical concatenation of the matrices $\mathbf{H_1}, \mathbf{H_2}, \ldots \mathbf{H_l}$, where the parity checks in $\mathbf{H_i}$ correspond to the lines in $S(v_i)$. 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 S.

If B consists of all nonorigin points in $EG(m, 2^s)$, then S contains all the lines in the geometry. Parity check codes thus formed are, with slight modification, the Type-I finite Euclidean geometry codes of [5], which we denote as $C_{EG}(m, 2^s)$. The modification is that, in the formation of the parity check matrix, the origin is excluded from the set of points, and lines through the origin are excluded from the set of lines; the purpose of this is to obtain a cyclic code[1]. The resulting codes are regular with column weight $(2^{ms} - 1) / (2^s - 1) - 1$ and row weight $\rho = 2^s$.

We shall impose the following two conditions on the set B:

Condition 1: Any γ-element subset A of B is linearly independent.
Condition 2: The elements of B span $EG(m, 2^s)$.

Here γ is a design parameter and is chosen to take the value 3 or 4. When the set B satisfies the above two conditions, we shall call the resulting codes reduced Euclidean geometry (REG) codes, and denote them as $C_{REG}(m, 2^s, B)$. Specifically, REG codes with $\gamma = 3$ are denoted $C^{(3)}_{REG}(m, 2^s, B)$, and REG codes with $\gamma = 4$ are denoted $C^{(4)}_{REG}(m, 2^s, B)$. A simple example of a REG code is where B is chosen to be a *basis* for $EG(m, 2^s)$. In this paper we shall derive some results on cycles in the Tanner graph of REG codes. To this end we define a *proper* $2n$-cycle to be a $2n$-cycle which consists of $2n$ *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 [11].

Theorem 1 *REG codes $C_{REG}(m, 2^s, B)$ are regular Gallager codes with column weight l and row weight 2^s, and their Tanner graphs have girth 8.*

Proof
For any nonorigin point $v \in EG(m, 2^s)$, the lines in $S(v)$ have no points in common, but contain all points in $EG(m, 2^s)$. Therefore, every column in the submatrix $\mathbf{H_i}$ (corresponding to the lines in $S(v_i)$) have weight 1. It follows that all columns in $\mathbf{H}$ have weight l, and that all rows have weight $\rho = 2^s$. Now, a 4-cycle in the Tanner graph would imply that two distinct points lie on two distinct lines, which violates the axioms of the Euclidean geometry; therefore the girth of the Tanner graph is at least 6. Assume there is a proper 6-cycle in the Tanner graph. This must correspond to a non-degenerate triangle in the geometry. Therefore there exist three distinct points x, y, z such that x and y lie on a line L_1, y and z lie on a line L_2 and z and x lie on a line L_3. Therefore $x + y = \beta_1 v_1$, $y + z = \beta_2 v_2$ and $z + x = \beta_3 v_3$ for some $v_1, v_2, v_3 \in B$ and some scalars $\beta_1, \beta_2, \beta_3 \in GF(2^s)$. Adding these three equations yields $\beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3 = 0$. By Condition 1, $\{v_1, v_2, v_3\} \subset B$ is a linearly independent set, which implies $\beta_1 = \beta_2 = \beta_3 = 0$. This is a contradiction, since the points x, y, z are distinct. This proves that the Tanner graph has girth 8. ■

Theorem 2 *The minimum distance of the REG code $C_{REG}(m, 2^s, B)$ is bounded by $d_{min} \geq 2^m$.*

Proof
By Condition 2, B contains a basis D; we prove the stronger result that the REG code $C_{REG}(m, 2^s, D)$ satisfies $d_{min} \geq 2^m$. Also, without loss of generality we may take $D = D_m = \{e_1, e_2, \ldots e_m\}$, where the i^{th} entry of e_i is equal to 1, and all other entries of e_i are equal to 0. The minimum distance of the code is equal to the minimum number of columns of $\mathbf{H}$ which sum to zero. This is equal to the size of the smallest nonempty subset X of the code bits which has the property that all check nodes connect into X an even number of times. Therefore, it suffices to prove the following statement for each $m \geq 1$:

[1] The advantage of this, as stated in [5], is that cyclic codes admit low-complexity linear-time encoding. The codes proposed here are not cyclic but approximately linear-time encoding may be achieved using the technique of [10].

If X is a nonempty subset of the points in $EG(m, 2^s)$ such that all lines parallel to vectors in D_m are incident on X an even number of times, than $|X| \geq 2^m$.

We prove this statement by induction on the geometry dimension m. Consider the case $m = 1$. The m-dimensional geometry consists of simply $\rho = 2^s$ points which lie on a single line. A nonempty subset X of the points must contain at least one element. The line incident on this point must be incident on X an even number of times, *i.e.* at least twice. Therefore $|X| \geq 2$ proving the case $m = 1$.

Now assume the result holds for some $m = j \geq 1$, and let X be a nonempty subset of the $(j+1)$-dimensional geometry $EG(j+1, 2^s)$ such that all lines parallel to vectors in D_{j+1} are incident on X an even number of times. Since X is nonempty, $\exists$ some $x \in X$; suppose $x = \left(\alpha_1^*, \alpha_2^*, \ldots \alpha_j^*, \alpha_{j+1}^*\right)$. Consider the set of points $\left\{\left(\alpha_1, \alpha_2, \ldots \alpha_j, \alpha_{j+1}^*\right)\right\}$ with $\alpha_i \in GF(2^s)$ for each $i = 1, 2, \ldots j$. This is simply $EG(j, 2^s)$; therefore, considering lines parallel to vectors in $D_j = \{e_1, e_2, \ldots e_j\} \subset D_{j+1}$, by the inductive hypothesis $\exists$ a subset of X, say X_j, of points lying in this subgeometry, of size at least 2^j. Now, for any $x \in X_j$, consider the line $L_x(e_{j+1})$. This line is incident on X at least twice, but cannot be incident on points in X_j. Therefore X contains at least one more point $x' \notin X_j$. Applying this argument to all $x \in X_j$ yields $|X| \geq 2|X_j| \geq 2^{j+1}$, and the result follows by the principle of induction. ■

We note that [6] cited the bound $d_{min} \geq l + 1$, where l denotes the column weight of $\mathbf{H}$. This bound also applies to REG codes.

III. Comparison of the number of Proper Eight-cycles in EG and REG codes

Theorem 3 *The number of proper eight-cycles in the Tanner graph of the REG code $C_{REG}^{(4)}(m, 2^s, B)$ is*

$$N_{REG}^{(8)} = l(l-1)\, 2^{ms} \left(2^s - 1\right)^2 / 8 \qquad (1)$$

Proof

Consider any point $x \in EG(m, 2^s)$, and any two points $v_1, v_2 \in B$. Pick $y \in L_x(v_1) \setminus \{x\}$ and $w \in L_x(v_2) \setminus \{x\}$. Then $x + y = rv_1$ and $x + w = sv_2$ for some elements $r, s \in GF(2^s) \setminus \{0\}$. Suppose that $xyzw$ forms a proper 8-cycle, for some point $z \in EG(m, 2^s)$. This must correspond to a non-degenerate quadrilateral in the geometry. Therefore we must have $z + y = pv_i$ and $z + w = qv_j$ for some $v_i, v_j \in B$, and for some $p, q \in GF(2^s) \setminus \{0\}$. Adding these four equations yields $rv_1 + sv_2 + pv_i + qv_j = 0$. Now, since v_1 and v_2 are distinct points in B, r and s are nonzero, and no four elements of B are linearly dependent (by Condition 1), the only way this equation may be satisfied is if pv_i and qv_j are equal to rv_1 and sv_2 in some order. If we suppose $pv_i = rv_1$ then $z + y = rv_1 = x + y$, implying $x = z$. This is an improper 8-cycle. Therefore we must have $pv_i = sv_2$ and $qv_j = rv_1$, giving $z + y = sv_2 = x + w$ and $z + w = rv_1 = x + y$. Both of these equations state that $w = x + y + z$. The important thing to note is that the solution for w is unique, and is given by the "parallelogram law" for the geometry. The expression for the number of eight-cycles may now be explained factor by factor. Firstly, there are 2^{ms} ways of choosing the initial point x. Then, there are l ways of choosing the point v_1. This choice being made, there are $l - 1$ choices for v_2. Then, there are $2^s - 1$ ways of choosing each of the points y and z. Given these choices, the solution for w is unique. Finally, since cycles are invariant to cyclic shift and reversal of the ordering $xyzw$, each cycle is counted 8 times. Therefore the total number of eight-cycles in the graph is $N_{REG}^{(8)} = l(l-1)\, 2^{ms} \left(2^s - 1\right)^2 / 8$. ■

Theorem 4 *The number of proper eight-cycles in the EG code $C_{EG}(m, 2^s)$ is*

$$N_{EG}^{(8)} = (n-1)(n-\rho)(n-2\rho+1)(n-5\rho+6)/8 \qquad (2)$$

where $n = 2^{ms}$ and $\rho = 2^s$.

Proof

As in the previous proof, the expression will be explained factor by factor. For the first point p_1 of the cycle, we may choose any nonorigin point in the geometry, hence there are $n - 1$ choices. For the second point p_2, we must exclude the line op_1; this leaves $n - \rho$ points. For the third point p_3 , all points on the lines op_2 and p_1p_2 must be excluded; this leaves $n - 2\rho + 1$ points. For the fourth point p_4, all points on the lines op_1, op_3, p_1p_2, p_2p_3 and p_1p_3 must be excluded; this leaves $n - 5\rho + 6$ points. Finally, cycles are counted with a multiplicity of 8 as in the proof of theorem 3 above. Therefore the total number of eight-cycles in the graph is $N_{EG}^{(8)} = (n-1)(n-\rho)(n-2\rho+1)(n-5\rho+6)/8$. ■

Note that a direct numerical comparison of the results provided by theorems 3 and 4 requires that the number of parity checks be taken into account. The main point is that the number of eight-cycles in REG codes is vastly reduced since these cycles must correspond to parallelograms rather than quadrilaterals in the geometry. Also note that there are $N_{EG}^{(6)} = (n-1)(n-\rho)(n-3\rho+3)/6$ six-cycles in the Type-I EG codes [5], whereas REG codes contain no six-cycles.

IV. Partly Parallel LDPC Decoder Implementation for REG codes

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.

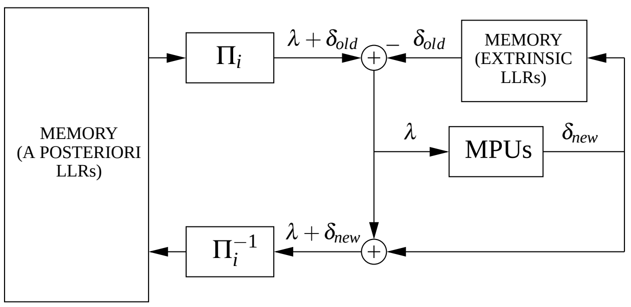

Figure 1: Partly parallel architecture for decoding of REG LDPC codes.

This leads to a partly parallel decoder architecture depicted in figure 1. The decoder operates on a blockrow-by-blockrow basis and therefore contains $2^{(m-1)s}$ 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 2). 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 $i = 1, 2, \ldots p$ completes one iteration of LDPC decoding. Decisions may be made on the *a posteriori* LLRs after any number of these iterations.

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 $c_1, c_2, \ldots c_n$. The parity-check equation $c_1 \oplus c_2 \oplus \ldots \oplus c_n = 0$ may be written in terms of a binary-valued "state" variable s_k as follows: $s_0 = 0$, and for each $k = 1, 2, \ldots n$, $s_k = s_{k-1} \oplus c_k$. Thus $s_n = 0$ and the factor f may be further factored as $f = \prod_{k=1}^{n} T_k \left(s_k, s_{k-1}, c_k\right)$, where T_k denotes the indicator function $[s_k = s_{k-1} \oplus c_k]$. This factorization of f leads to the factor graph transformation depicted in figure 2. 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].

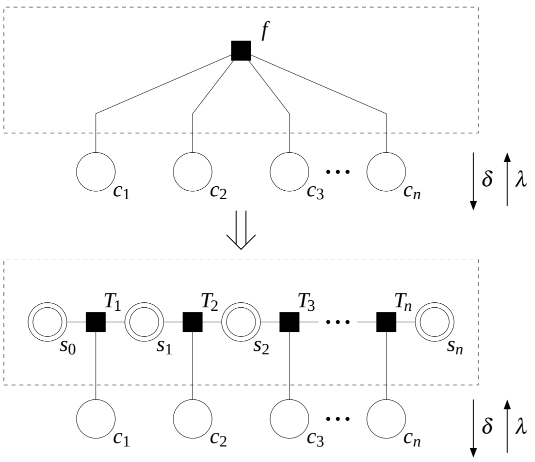

Figure 2: Factor graph transformation to facilitate the TDMP algorithm for decoding of REG LDPC codes.

V. Simulation Results

In this section, error correcting performance is demonstrated for two example girth-8 REG codes $C_{REG}^{(3)}(m, 2^s, B)$. 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 N_{iter} iterations were completed.

The first code was constructed using code construction parameters $m = 3$, $s = 3$ and $B = \{(1,0,0), (0,1,0), (0,0,1), (1,1,1)\}$. The resulting 256×512 parity check matrix has rank 208. The bit error rate (BER) and frame (codeword) error rate (FER) performance of this $(512, 304)$ LDPC code is shown in figure 3 for $N_{iter} = 10$. It may be seen from the figure that the partly parallel TDMP-based implementation loses little of the performance of the fully parallel SPA decoder. The second code was constructed using the parameters $m = 4$, $s = 3$ and $B = \{(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)\} \cup \{(0,1,1,1), (1,0,1,1), (1,1,0,1), (1,1,1,0)\}$. The resulting 4096×4096 parity check matrix has rank 2492. The BER and FER performance of this $(4096, 1604)$ LDPC code is shown in figure 4 for both decoder implementations. Also shown is the BER performance for parallel SPA decoding with $N_{iter} = 100$. As may be seen, little is gained by using the larger value of N_{iter}. We see that 10 iterations of TDMP-based partly parallel decoding compares very favourably with 100 iterations of the fully parallel decoder.

VI. Conclusion

A class of regular Euclidean geometry based LDPC codes has been proposed. These codes were shown to exhibit desirable LDPC code properties such as girth eight

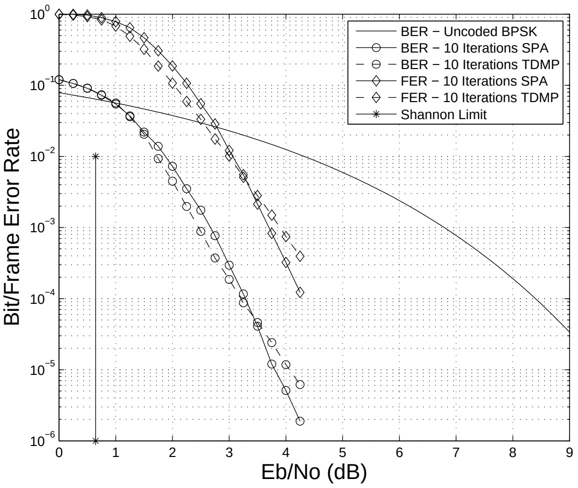

Figure 3: Error performance of the $(512, 304)$ REG code constructed using $m = 3$, $s = 3$ and $B = \{(1,0,0), (0,1,0), (0,0,1), (1,1,1)\}$.

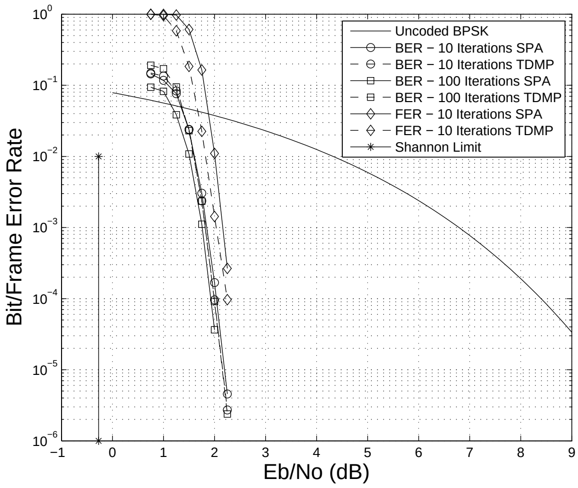

Figure 4: Error performance of the $(4096, 1604)$ REG code constructed using $m = 4$, $s = 3$ and $B = \{(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)\} \cup \{(0,1,1,1), (1,0,1,1), (1,1,0,1), (1,1,1,0)\}$.

in the Tanner graph, sparsity of eight-cycles in the Tanner graph, and a lower bound on the code minimum distance. A partly parallel decoder architecture was outlined for these codes and error rate performance using a small number of iterations with this architecture was demonstrated to be near-Shannon limit, and competitive with the error rate performance of a fully parallel decoder.

Acknowledgments

The authors would like to acknowledge the support of Enterprise Ireland.

References

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

[2] J. M. F. Moura and J. Lu and H. Zhang, "Structured Low-density parity-check codes," *IEEE Signal Processing Magazine*, pp. 42–55, January 2004.

[3] B. Vasic and O. Milenkovic, "Combinatorial constructions of low-density parity-check codes for iterative decoding," *IEEE Transactions on Information Theory*, pp. 1156–1176, vol. 50, no. 6, June 2004.

[4] S. Lin and D. Costello, "Error control Coding: Fundamentals and Applications," Prentice Hall, Englewood Cliffs, N. J., 1983.

[5] Y. Kou and 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*, pp. 2711–2736, vol. 47, no. 7, November 2001.

[6] H. Tang and J. Xu and Y. Kou and S. Lin and K. Abdel-Ghaffar, "On algebraic construction of Gallager and circulant low-density parity-check codes," *IEEE Transactions on Information Theory*, pp. 1269–1279, vol. 50, no. 6, June 2004.

[7] H. Tang and J. Xu and S. Lin and K. A. S. Abdel-Ghaffar, "Codes on Finite Geometries," *IEEE Transactions on Information Theory*, pp. 572–596, vol. 51, no. 2, February 2005.

[8] S. M. Aji and R. J. McEliece, "The Generalized Distributive Law," *IEEE Transactions on Information Theory*, vol. 46, pp. 325–343, March 2000.

[9] F. R. Kschischang and 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.

[10] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 638–656, February 2001.

[11] R. J. McEliece and 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, no. 2, pp. 140–152, February 1998.

[12] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC Decoders," *IEEE Transactions on Very Large Scale Integrated Systems*, vol. 11, no. 6, pp. 976–996, December 2003.

[13] L. R. Bahl and J. Cocke and 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.