Irish Signal and Systems Conference, Dublin, June 28–30, 2006

Construction of Girth 8 LDPC Codes based on Finite Geometries

Mr John Craddock,[†] **Dr Mark Flanagan and Prof Anthony Fagan**

[†]*School of Electronic,*
Electrical and Mechanical Engineering
University College Dublin
IRELAND

E-mail: [†]`john.craddock@ee.ucd.ie`

Abstract — **This paper presents a novel method for constructing Low Density Parity Check (LDPC) codes with a girth of up to 8. These codes are based on the structural properties of finite lattice geometries. Results are presented which show that these codes perform well over awgn channels with iterative decoding. These codes also have several implementation advantages, which are presented.**

I Introduction

Low Density Parity Check (LDPC) codes were first introduced by Robert Gallager of MIT in 1960 [1]. They remained unused until they were rediscovered by MacKay in 1997 [2]. LDPC codes are proven to be very good, in that sequences of codes exist which, when optimally decoded, achieve information rates up to the Shannon limit.

LDPC codes are a type of block code defined as the null space of a binary parity check matrix H. The term 'low density' refers to the sparseness of the H matrix. LDPC codes are divided into two categories, regular (Gallager) codes and irregular codes. The column and row weights of regular LDPC codes are constants, ρ and γ respectively where both ρ and γ are small compared to the codeword length n. An LDPC code is irregular if its row or column weights vary. Irregular LDPC codes have been shown to outperform regular LDPC codes, however regular codes have some implementation advantages.

LDPC codes can be represented by a Tanner graph. Each column in the H matrix represents a variable node j and each row a constraint node k. Nodes j and k are connected by an edge iff the element of the H matrix (k, j) is non-zero. Fig. 1 shows the tanner graph of the code given by;

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad (1)$$

A cycle in the code is a closed path along edges. The cycle $f_0, c_0, f_1, c_4, f_2, c_1, f_0$ is highlighted in fig 1. The girth of the code is the length of the shortest cycle. Most randomly constructed codes have a girth of four. For codes with short cycles in their Tanner Graph, iterative decoding becomes correlated after a small number of iterations and decoding may not converge or may converge more slowly.

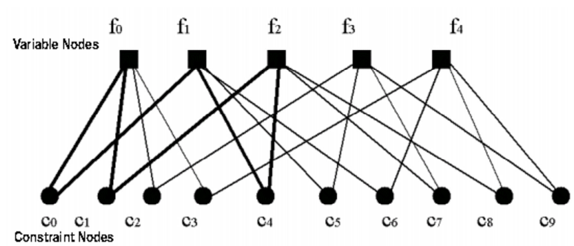

Fig. 1: The Tanner graph of H given in (1).

LDPC codes are usually decoded using a message passing algorithm. The algorithm acts locally. Each variable and constraint node gathers information from and passes information to its neighbouring nodes. This process continues until it converges to a valid codeword or a maximum number of iterations is reached. The message passing algorithm is optimum in the case of a graph with a tree structure i.e. containing no cycles. One means of improving the performance of the message passing algorithm to decode LDPC codes in to increase the girth of the Tanner graph defined by the code. This increases the number of iterations necessary for information to be passed around a cycle in the code and makes the code more tree-like.

Several geometric methods of producing LDPC codes have been presented, see for example [3] and [4]. Codes produced in this manner have a girth of 6.

In section 2 a novel method of constructing LDPC codes based on a finite lattice geometry is presented. These codes have a girth of up to 8. This is an improvement on previous geometric constructions whose girth was limited to 6. Section 3 contains details of the advantages this construction yields in practical implementations. Section 4 contains results which show that these codes perform well over an AWGN channel. Section 5 concludes the paper.

II Code Construction

The code construction proposed is based on a regular finite lattice, see Fig 2 and 3. Points in this lattice represent variable nodes. Bundles of parallel lines are drawn on the lattice, representing constraint nodes. A variable j participates in the constraint k iff the point j intersects the line k. Choosing bundles of parallel lines such that no two points have more than one line in common ensures that no cycles of length four exist. If the bundles of parallel lines are chosen such that no group of three bundles of parallel lines are linearly dependant then the girth of the code rises to 8.

Examples

Example 1

Fig 2 shows a 2 dimensional finite lattice of size 4×4. 3 bundles of parallel lines are drawn. As this is a finite lattice with each side of length 4, arithmetic is conducted modulo 4. Lines wrap around the lattice, as shown in the case of the line (2,0), (3,1), (0,2), (1,3) highlighted. There are 12 lines drawn in 3 bundles of 4. Each line contains 4 points. This example gives a regular code with 16 variable nodes and 12 constraint nodes. The column and row weights are 3 and 4 respectively. As no two points are contained together in more than one line, no cycle of length 4 can exist. The girth of this code is 6 and an example of a loop of length 6 is highlighted.

Example 2

Fig 3 shows a 2 dimensional finite lattice of size 4×4. 2 bundles of parallel lines are drawn. Each line contains 4 points. This is example gives a regular code with 16 variable nodes and 8 constraint nodes. The column and row weights are 2 and 4 respectively. In this case the bundles of parallel lines chosen are linearly independent so no loop of length 6 can exist, see theorem 2.1. The shortest cycle in this code is of length 8 and a cycle of length 8 is highlighted.

Theorem 2.1

A code constructed by this method using bundles of parallel lines where any set of three bundles is linearly independent has a girth of at least 8.

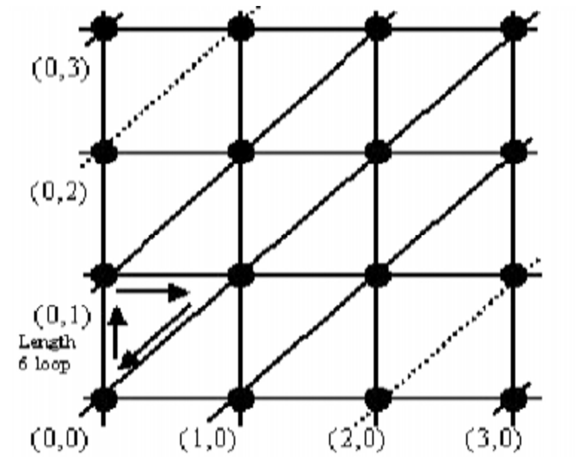

Fig. 2: A finite lattice geometry of size 4×4 showing three bundles of parallel lines. One line is highlighted as an example of how lines wrap around the finite lattice.

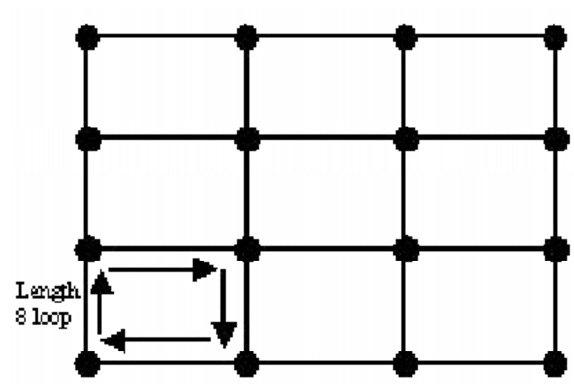

Fig. 3: A finite geometry of size 4×4 showing two linearly independent bundles of parallel lines.

Proof:

Assume that the code contains a length 6 *cycle.*

Three non-zero vectors in the direction of the lines in the geometry must sum to zero for the cycle to be closed.

This violates the condition that any three vectors are linearly independent.

Therefore the original assumption must be false and the code cannot contain a length 6 *cycle.*

The condition that no two points have more than one line in common ensures that no cycles of length 4 *exist in the code.*

The code then must have a girth of at least 8

In both examples shown a regular code was constructed by choosing a cubic lattice, i.e. a lattice with the same number of points in each dimension. The row weight of the code can be varied by varying the number of points in each dimension. The column weights of the H matrix can be varied by omitting some of the lines from the construction, this is the equivalent of removing rows from the H matrix or constraint nodes from the Tanner Graph. Irregular codes can be constructed by this method by varying the row or column weights or a combination of both.

III Decoding

Parallel hardware implementation of a message passing decoding algorithm requires each node in the Tanner Graph of the code to be implemented physically on a chip. As the length of the code increases, implementing the edges connecting the nodes as connections on the chip represents an increasingly difficult routing problem. This limits the length of a code which can be implemented. In other implementations the interconnections are stored in memory. The amount of memory this requires can limit the size of a code.

Geometrically constructed regular LDPC codes mitigate against both these problems. As these codes are constructed according to geometric rules rather than randomly constructed, the interconnections between the nodes in the code's Tanner Graph do not need to be stored individually in a decoder's memory. The Tanner Graph can be constructed knowing only the nature of the geometry representing the variable nodes and which parallel bundles of lines representing constraint nodes are to be used.

The regular LDPC codes presented here can be written in the form:

$$\begin{bmatrix} H_1 \\ H_2 \\ \dots \\ H_\rho \end{bmatrix} \tag{2}$$

where each H_n is a matrix with column weight of 1 and a constant row weight. This allows partly parallel implementation of the turbo decoding algorithm presented in [5]. This algorithm offers reduced memory requirements, reduced routing complexity and improved decoder throughput over message passing decoders.

IV Results

This code construction was tested over an AWGN channel and decoded using a message passing algorithm with a maximum of 20 iterations, see Fig. 4 The codes tested were constructed using four bundles of parallel lines in an $8\times8\times8$, 3 dimensional finite lattice. In the case of the girth 8 code, the four parallel bundles of lines were chosen such that any three were linearly independent, they were chosen randomly in the case of the girth 6 code, subject to the condition that no two points had more than one line in common. Both of these codes had 301 information bits giving a rate of 0.59. For comparison, a code was constructed randomly with a row weight of 8 and column weights of 3 and 4 to give a length 512, rate 0.59 code. These results show that a Lattice geometry code of girth 8 outperforms a randomly constructed code by over 0.3dB at a BER of 10^{-3}.

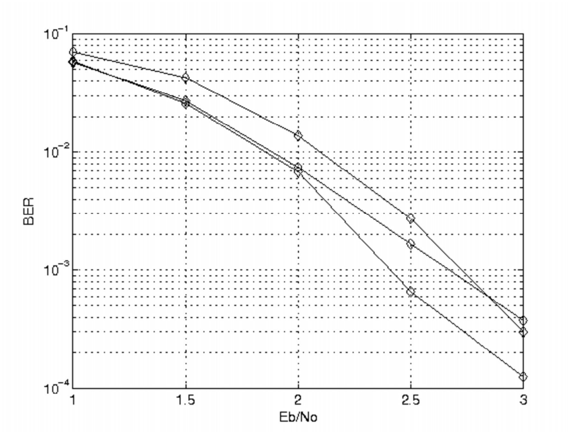

Fig. 4: BER for length 512 rate 0.59 codes. The lines, from left to right are Finite Lattice codes of Girth 8 and Grith 6 and a randomly constructed code.

V Conclusion

This paper presented a novel method for constructing LDPC codes based on finite lattice geometries. These codes have a girth of up to 8. It was shown that these codes have some implementation advantages and that they perform well over AWGN channels.

References

[1] R. Gallager "Low-density parity-check codes". *Information Theory, IEEE Transactions on*, 8:21–28, 1962.

[2] D.J.C. MacKay "Good error-correcting codes based on very sparse matrices". *Information Theory, IEEE Transactions on*, 45:399–431, 1999.

[3] Heng Tang, Jun Xu, Yu Kou, Shu Lin and Khaled Abdel-Ghaffar "On Algebraic Construction of Gallager and Circulant Low-Density Parity-Check Codes". *Information Theory, IEEE Transactions on*, 50:1269 - 1279, 2004.

[4] Y. Kou, S. Lin and M Fossorier "Low-density parity-check codes based on finite geometries: A rediscovery and new results". *Information Theory, IEEE Transactions on*, 47:2711 - 2736, 2001.

[5] Mohammad M. Mansour "High-Throughput LDPC Decoders". *Very Large Scale Integration Systems, IEEE Transactions on*, 11:976–996, 2003.