

# Low Complexity Regular LDPC codes for Magnetic Storage Devices

Gabofetswe Malema, and Michael Liebelt

**Abstract**—LDPC codes could be used in magnetic storage devices because of their better decoding performance compared to other error correction codes. However, their hardware implementation results in large and complex decoders. This one of the main obstacles the decoders to be incorporated in magnetic storage devices. We construct small high girth and rate 2 column-weight codes from cage graphs. Though these codes have low performance compared to higher column weight codes, they are easier to implement. The ease of implementation makes them more suitable for applications such as magnetic recording. Cages are the smallest known regular distance graphs, which give us the smallest known column-weight 2 codes given the size, girth and rate of the code.

**Keywords**— Structured LDPC codes, cage graphs.

## I. INTRODUCTION

LOW-density parity-check (LDPC) codes have been shown to perform close to the Shannon limit. It was proved in [1] that LDPC codes with column weight (number of 1's in code matrix column)  $j \geq 3$  have a minimum distance that grows linearly with the block length (number of columns)  $n$  for given  $j$  and row-weight (number of 1's in matrix row)  $k$  and that the minimum distance for codes with  $j = 2$  grows logarithmically with  $n$ . However, compared with  $j \geq 3$  codes, codes with  $j = 2$  are easier to implement and require less storage making them a target for applications such as magnetic recordings [2]. It is also reported that codes with column weight of 2 exhibits block error statistics more compatible with an outer Reed-Solomon (RS) code.

Reed-Solomon(R-S) codes are widely used in current storage devices, mainly because of their relatively low hardware-complexity and high capacity to detect and correct burst errors. They also lend themselves to high-speed encoding and decoding required for high bandwidth disk drives [3]. These applications require short codeword lengths, which allow easier encoding, less implementation memory and low decoding complexity. LDPC codes offer significant performance advantage over R-S codes; however, it comes at

Manuscript received August 21, 2005.

G. Malema is a PhD Student in the Electrical and Electronic Engineering Department, The University of Adelaide, Australia. ( phone: 618-8303-4707; fax: 618-8303-4360; e-mail: gmalema@eleceng.adelaide.edu.au).

M. Liebelt is a senior lecturer and head in the Electrical and Electronic Engineering department, The University of Adelaide, Adelaide, South Australia, 5000 (phone: 618-8303-5057 e-mail: mike@eleceng.adelaide.edu.au).

the expense of increased complexity. LDPC codes can be used as second level codes in concatenated error correction architectures as shown in Figure 1 [2]. For this architecture to be practical we need small decoder VLSI to fit both encoder and decoder making it easier to integrate in magnetic storage devices. LDPC codes with column weight of 2 have low implementation complexity and good performance for magnetic recording applications[2].

There are several ways of constructing column-weight 2 LDPC codes. In [2], 4 and 6 cycle-free codes are constructed

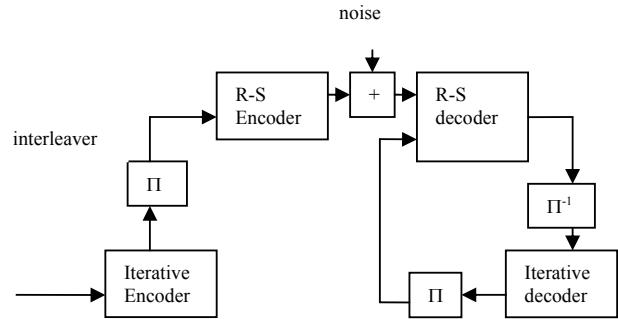


Fig. 1 Serial concatenation of partial response channel encoding and decoding

using disjoint difference sets and in [4] codes with large girth are constructed based on geometry. Though the codes in [4] have large girth, they are not the smallest. The construction method also does not produce high girths and rates codes. For example, codes of girths 16 and 20 are of rate 1/3. In this paper we construct low complexity codes that could be incorporated in magnetic storage devices from cage graphs, with high girth and high rates. Since cages are the smallest known graphs of a given vertex degree and girth, the resulting codes are the smallest one could get for the same girth and vertex degree.

This paper is organized as follows. Section II describes cages and construction of codes based on cages. Section III concludes.

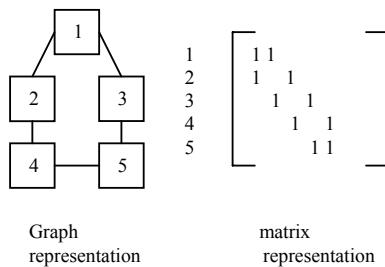


Fig. 2 Graph and matrix representation of 10-cycle

## II. CONSTRUCTING LDPC CODES FROM CAGES

A LDPC code matrix could be represented by a bipartite graph on which one set of vertices represents rows and another set represent columns. The girth,  $g$  is the smallest circle in the graph. A large girth improves the decoding performance of the sum-product algorithm used in LDPC codes. The code matrix could also be represented by a graph with vertices as rows only. In this type of graph, the vertices are rows whereas the columns are represented as edges. The rows connected to the same column form a complete graph. The complete graph in the graph representation forms a column in the matrix. For column weight 2 codes the column is formed by a connection between 2 vertices. Figure 2 shows a graph with a cycle of length 5. Each edge represents a column. The corresponding matrix representation, on which the vertices are rows and edges are columns has a cycle length of 10. The distance in the graph is half what it is in the code matrix. Therefore to construct a code of girth  $g$  we can construct a graph of girth  $g/2$ .

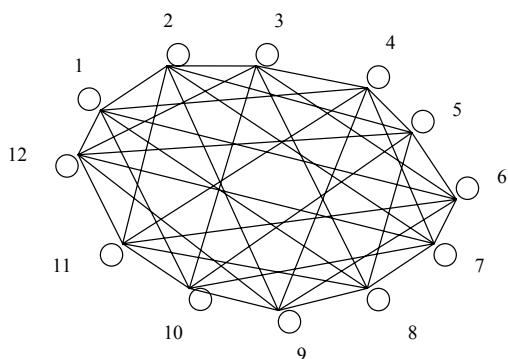


Fig. 3 A (6,4) cage graph

### A. Cages

A  $(k,g)$ -cage is a regular graph of valency (edge degree)  $k$  and girth  $g$  with minimal number of vertices. A lower bound on  $n(k,g)$ , the number of vertices of a cage, depends on whether  $g$  is even or odd. If  $g$  is odd then

$$n(k,g) \equiv I + k + k(k-I) + \dots + k(k-I)^{(g-3)/2} \quad (1)$$

[5][6] and if  $g$  is even, then

$$n(k,g) = 1 + k + k(k-1) + \dots + k(k-1)^{g/2} - 2 + (k-1)^{g/2} - 1 \quad (2)$$

The lower bounds, known as Moore bounds are met very infrequently [6]. Though there isn't a uniform approach to constructing arbitrary cages, there are many cages constructed for some valencies and girths. Cages with valency of 3 are called cubic cages and are easier to find compared to cages of higher valencies [7]. Examples of cubic cages of varying girths and construction methods could be found in [5][6] and [8]. These graphs produce codes with a rate of  $1/3$ . In general the rate is  $1-2/k$ . The size of the matrix is the number of vertices by the number of edges in the graph.

Cages of higher valencies are desirable in that they increase decoding performance. More nodes (rows and columns) contribute towards estimating the received bit. They also improve the data rate. Though they are harder to find at higher girths compared to cubic cages, there are many examples and constructions methods in literature. Some examples could be found in [9][10]. In [11] ways of generating regular graphs and cages are described. There is also associated software by the same author at [12] that generates cages.

We use cages to construct regular LDPC codes with column weight of 2. The vertices of the graph represent the rows of the code matrix whereas the edges are the columns. That is, each edge is a connection of two rows in the code matrix. Since only two vertices/rows are connected, the column weight is 2. Cages are the smallest (number of vertices) known regular graphs of a given valency and girth. Therefore, they give us the smallest known regular column-weight 2 LDPC codes with a given rate (vertex degree) and girth. Figure 3 shows a (6,4) cage graph. There are 12 vertices and 36 edges. A corresponding adjacency matrix is shown in Figure 4 with girth 8.

Figure 5 shows the performance curve for a (2,11) code of a high rate of 9/11 with girth of 10. Though the performance of the code is not as good as other codes, it is good enough for magnetic storage systems [2].

Fig. 4 Structured matrix for the (6,4) cage graph

### B. Hardware Implementation

Structured codes reduce hardware complexity both in the encoder and decoder. The structure or pattern in the code makes it easier to multiply elements of the generator matrix. The cage graph of figure 3 produces a 12x36 code matrix with a girth of 8. The matrix has a defined structure in that odd vertices are connected to even vertices and vice versa. As an example the edges (columns) could be divided into six groups as shown in figure 4. Edges from each odd vertex could form a group. By doing that we get 6 groups of 6 edges each. A similar division with edges from even vertices works in a similar way. In figure 4 columns (edges) for each row (vertex) are treated as a group. In each group connections are between the odd row and all even rows. So the connections could be calculated rather than storing the destination addresses. The grouping reduces the number of interconnections. The number of groups is much smaller than the number of individual rows or columns. Structuring also simplifies addressing. Addresses are generated according to the code pattern. The addressing in each group is determined by the pattern, which are all odd or even rows. In random codes the random communication makes exchange of messages between processing nodes complicated. If the communication network is not hardwired an address table is needed.

Column-weight 2 codes simplify the encoder and decoder complexity further by reducing the number of elements in the code matrix. This means less computations and memory in the encoder. Besides reduced memory, the variable nodes (column computations) complexity of the decoder is reduced. The variable node computation involves the summation of the incoming messages and the channel estimate of the information (received) bit. With two incoming messages, the computation is reduced to exchanging incoming messages and adding them to the channel estimation before sending them as outgoing messages.

However, as noted above, cage graphs are not constructed with one method. Therefore their structure will differ from graph to graph. It is important then to analyze each graph to best exploit its structure for hardware implementation. Decoding performance characteristics for each graph also need to be analyzed before implementation.

### III. CONCLUSION

An approach for constructing LDPC codes with column weight of 2 has been described. Cage graphs are used to represent the code matrix, where vertices are rows and edges are columns. Since cages are the smallest known regular graphs of a given girth and valency, the resulting codes are the smallest one could obtain. These codes have low implementation costs and could therefore be incorporated in magnetic storage devices. Non-regular distance graphs could also be used to construct column-weight 2 codes in a similar way. The resulting codes would have varying rows valencies.

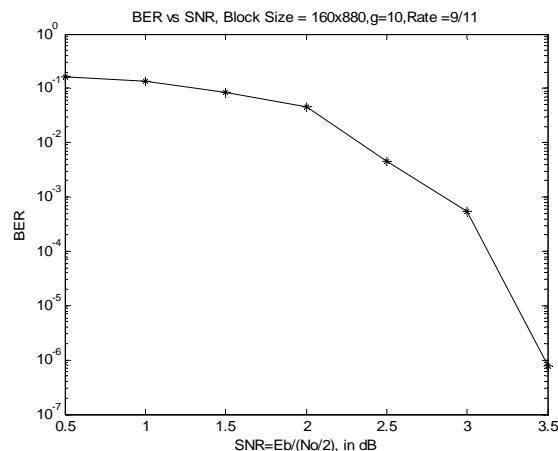


Fig. 5 BER performance curve of a (2,11) code with  $g=10$

### REFERENCES

- [1] R.G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, 1964, pp21-28.
- [2] H.Song, "Low complexity LDPC codes for magnetic recordings," *IEEE Globecom 2002*, November 2002.
- [3] C.Riggle and S. McCathy, "Design of error correction systems for disk drives," *IEEE Transactions Magazine*, July 1998, vol. 34, pp .2362-2371.
- [4] H. Zhang and J.M. Moura , "Large-girth LDPC codes based on graphical models , " *IEEE Workshop on Signal Processing*., 2003, pp-100-103.
- [5] N.L Biggs, "Constructions for Cubic Graphs of large Girth," *Electronic Journal of Combinatorics*, vol. 5, 1998.
- [6] G. Exoo , "A Simple Method for Constructing Small Cubic Graphs of Girths 14,15 and 16 , " *Electronic Journal of Combinatorics*, Vol. 3,1996.
- [7] P.K Wong, "Cages—A Survey," *Journal of Graph Theory*., vol. 3 1982, pp-1-22.
- [8] N.L Biggs, "Cubic Graphs with Large Girth," *Combinatorial Mathematics: Proceedings of the Third International Conference*,1998, pp.56-62.
- [9] E.W Weisstein , "Cage Graph , " From MathWorld—A Wolfram Web Resource. <http://www.mathworld.wolfram.com/CageGraph.html>
- [10] G. Royle, "Cages of higher valency," [Online] Available:<http://www.cs.uwa.edu.au/gordon/cages/allcages.html>
- [11] M. Meringer, "Fast generation of regular graphs and construction of cages," *Journal of Graph Theory*, 1999, vol.30, pp.137-146.
- [12] M. Meringer, "Genreg-download and manual," [Online] Available: <http://www.mathe2.uni bayreuth.de/markus/manual/genreg.html>.

**G. Malema** received BS degree Computer Engineering from Valparaiso University (97) and MS degree Electrical and Computer Science form the University of Illinois at Chicago (2000). He worked at the University of Botswana as a lecturer in the department of Computer Science. He is currently a PhD student at the University of Adelaide looking at low density parity check codes design and implementation.

**M. Liebelt** is a senior lecturer and head of department, Electrical and Electronic Engineering, at the University of Adelaide. He has a B.E(Hons) Electrical Engineering (1979) and M.Eng.Sc (1982) from the University of Adelaide.