

Quasi-cyclic LDPC codes from difference families

Sarah J. Johnson¹

School of Electrical Eng'g & Computer Science
University of Newcastle
Callaghan, NSW 2308
Australia

e-mail: sarah@ee.newcastle.edu.au

Steven R. Weller²

School of Electrical Eng'g & Computer Science
University of Newcastle
Callaghan, NSW 2308
Australia

e-mail: steve@ee.newcastle.edu.au

Abstract — We consider in this paper regular and nearly regular quasi-cyclic low-density parity-check (LDPC) codes, derived from families of difference sets. The codes have girth at least 6, and sparse parity-check matrices. They are designed to perform well when iteratively decoded with the sum-product decoding algorithm and to allow low complexity encoding.

I. INTRODUCTION

Low-density parity-check (LDPC) codes were first presented by Gallager [?] in 1962 and have created much interest recently when rediscovered and shown to perform remarkably close to the Shannon limit [?], [?]. The decoding algorithm presented by Gallager, since shown to be a special case of the sum-product decoding algorithm [?], has the advantage of decoding complexity linear in the code length. Consequently, error-correcting codes with very long code lengths are feasible.

One of the main hurdles in the implementation of LDPC codes is the computational complexity of the encoding algorithm for these codes. Encoding is, in general, performed by matrix multiplication and complexity grows with the square of the code length. One class of codes for which very low complexity encoding schemes have been demonstrated are cyclic codes. However low decoding complexity is only guaranteed when the codes have very sparse parity-check matrices and for most known cyclic codes this is not the case. Some of the finite geometry codes from [?], [?], which show remarkable iterative decoding performance, are cyclic and have reasonably sparse parity check matrices. However, the cyclic finite geometry codes exist for only a limited range of code lengths and rates, and column weight increases with code length leading to increased decoding complexity.

An alternative to cyclic codes with encoding complexity almost as low are *quasi-cyclic* codes, first presented in [?] and [?]. A code is quasi-cyclic if for any cyclic shift of a codeword by c places the resulting word is also a codeword, so a cyclic code is a quasi-cyclic code with $c = 1$.

We consider in this paper quasi-cyclic codes described by a parity check matrix

$$H = [A_1, A_2, \dots, A_l]$$

where A_1, \dots, A_l are $v \times v$ circulant matrices. To see that H is quasi-cyclic with $c = l$ simply consider the columns of each circulant matrix interleaved as

$$H = [A_1(1), A_2(1), \dots, A_l(1), \dots, A_1(v), A_2(v), \dots, A_l(v)]$$

where $A_j(i)$ is column i of the j th circulant.

Provided that one of the circulant matrices is invertible (say A_l) the generator matrix for the code can be constructed in systematic form

$$G = \begin{bmatrix} & & & (A_l^{-1} A_1)^T \\ & & & (A_l^{-1} A_2)^T \\ & & & \vdots \\ I_{v(l-1)} & & & (A_l^{-1} A_{l-1})^T \end{bmatrix}$$

resulting in a $(vl, v(l-1))$ quasi-cyclic code. Encoding can be achieved with linear complexity using a $(v-l)$ -stage shift register in much the same way as cyclic codes [?].

The algebra of $v \times v$ binary circulants is isomorphic to the algebra of polynomials modulo $x^v - 1$ with coefficients over GF(2) [?]. A circulant matrix A is completely specified by the polynomial $a(x)$ with coefficients from its first row. In polynomial representation we have a codeword $c(x) = i(x), p(x)$ with coefficients in GF(2), where $i(x)$ is the polynomial representation of the information bits and $p(x)$ is given by

$$p(x) = \sum_{j=1}^{l-1} i_j(x) * (a_l^{-1}(x) * a_j(x))^T.$$

The polynomial $i_j(x)$ is the representation of the information bits $v(j-1)$ to vj and polynomial multiplication ($*$) is mod $x^v - 1$.

Fig. ?? shows the parity check matrix, generator matrix and Tanner graph of a small quasi-cyclic code of rate-1/2 with the first circulant $a_1(x) = 1 + x$, and the second $a_2(x) = 1 + x^2 + x^4$. (Note that for this example the code is not 4-cycle free.)

The aim of this work is to develop quasi-cyclic codes suitable for iterative decoding using the sum-product algorithm. To this end, quasi-cyclic codes with sparse parity-check matrices, large girth and good minimum distance are sought. A sparse parity-check matrix is essential for workable decoding complexity while large minimum distance improves the error floor performance of the code. Large girth results in reduced dependence in the

¹Work supported in part by a CSIRO Telecommunications & Industrial Physics postgraduate scholarship

²Work supported by the Centre for Integrated Dynamics and Control

message passing and so to more efficient iterative decoding. The avoidance of 4-cycles in an LDPC code, to give a girth of at least 6, has often been proposed to improve the performance of the sum-product decoding algorithm [?], [?], [?]. A 4-cycle exists in a code where two parity check sums check on the same pair of code bits. A 4-cycle in a code causes successive iterations of the sum-product decoding algorithm to be highly correlated after only two iterations.

We propose that good quasi-cyclic LDPC codes without 4-cycles can be derived by considering difference families and we present in Section ?? background on difference families before describing the codes and their performance in Sections ?? and ??.

II. DIFFERENCE SETS AND DIFFERENCE FAMILIES

In the following section we describe difference systems and demonstrate that they can be used to construct quasi-cyclic matrices which are 4-cycle free. We begin with the more specific difference sets and generalize to difference families. Our treatment of these difference systems follows that of Anderson [?].

Defⁿ 1: Consider an arbitrary Abelian group \mathcal{G} of order v . A (v, k, λ) *difference set* is a set $D = \{d_1, \dots, d_k\}$ of elements from \mathcal{G} such that each non-zero element g of \mathcal{G} has exactly λ representations as a difference $g = d_i - d_j$.

The set $D + g = \{d_1 + g, d_2 + g, \dots, d_k + g\}$, $g \in \mathcal{G}$, is a translate of D . Let the translates $D, D + g_1, \dots, D + g_{v-1}$ make up the columns of a matrix N , where the position (i, j) in N is a one if the i th element in \mathcal{G} is in the set $D + g_j$, and zero otherwise:

$$A_{i,j} = \begin{cases} 1 & \text{if } g_i \in D + g_j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Theorem 1: The set of translates of a difference set (v, k, λ) in any Abelian group will give a regular square matrix N which is 4-cycle free.

Proof: Take as columns all the translates $D + g$, $g \in \mathcal{G}$ of a (v, k, λ) difference set. Consider the elements $a, b \in \mathcal{G}$. Now $a = d_i + (a - d_i)$ for each i , so a occurs in the translates $D + (a - d_i)$. Similarly, the element b occurs in the translates $D + (b - d_i)$. Thus a, b occur together in a translate $D + h$, $h \in \mathcal{G}$ precisely when $h = a - d_i = b - d_j$ for some i, j . But $a - d_i = b - d_j \Leftrightarrow a - b = d_i - d_j$ and $a - b$ is an element of \mathcal{G} . There are λ pairs (d_i, d_j) for which the differences $d_i - d_j$ give an element of \mathcal{G} and we have that the elements a, b occur together in exactly λ translates. With $\lambda = 1$ each pair of elements occur in exactly one column together so the matrix N is free of 4-cycles. Each set and hence each translate have weight k , and since two elements must appear in exactly λ translates together, the row weight $r = \lambda(v - 1)/k$, and regularity is also proved. ■

Difference sets defined on any group isomorphic to Z_v produce cyclic matrices since $D + j(\text{mod } v)$, $j \in \{0, 1, 2, \dots, v - 1\}$ is a cyclic shift of D . These sets, called *perfect* difference sets, exist for all $k = p^s + 1$, p a prime and s an integer [?]. Matrices from difference sets were originally proposed for use as the parity-check

matrix of cyclic error correcting codes by Weldon [?] and at the same time in the form of projective geometries by Rudolph [?]. More recently, these difference set cyclic codes have been shown to perform well when iteratively decoded with the sum-product decoding algorithm [?], [?].

The matrices obtained from difference sets in a group of order v are always square since there are v translates of a difference set, one for each of the v elements in \mathcal{G} , and hence v rows and columns in the matrix. To construct 4-cycle free matrices which are not square but which are constructed from cyclic translates we need the concept of difference families.

Defⁿ 2: Let D_1, \dots, D_t be sets of size k in an additive Abelian group \mathcal{G} of order v such that the differences arising from the D_i give each nonzero element of \mathcal{G} exactly λ times. The sets $D_1 \dots D_t$, which need not be disjoint, then form a (v, k, λ) *difference family* in \mathcal{G} .

Theorem 2: Take the $(v, k, 1)$ difference family D_1, \dots, D_t in Z_v . The sets $D_i + j$, $1 \leq i \leq t, 0 \leq j \leq v - 1$ will give a $v \times vt$ regular matrix N from circulants without 4-cycles.

Proof: Let a, b be any two elements of Z_v , $a = d_{ih} + (a - d_{ih})$ so a occurs in the translate $D_i + (a - d_{ih})$, $1 \leq h \leq k$. Similarly, b occurs in the translates $D_i + (b - d_{ih})$. Thus a, b occur together in a translate $D_i + j$ precisely when $j = a - d_{ih} = b - d_{il}$ for some i, h, l . By the definition of difference families there are precisely λ choices of i, h, l for which $a - b = d_{ih} - d_{il}$ and so a, b occur together in exactly λ translates. Since $\lambda = 1$, 4-cycles are avoided. The sets are all of size k and there are t of them. Each set has v translates so we have $b = vt$ columns in N . Two elements must be in exactly one translate together and so row weight $r = kt$. ■

Wilson [?] gives the conditions required for a $(v, k, 1)$ difference family.

Lemma 1: Let $v \equiv 1 \pmod{k(k-1)}$ be a prime power such that

$$\exists B = \{b_1, \dots, b_k\} \subset \text{GF}(v) : \Delta^+ B$$

is a complete system of representatives for the cosets of $H^{k(k-1)/2}$ in H , the multiplicative group of $\text{GF}(v)$. Then the family

$$F = \{w^{k(k-1)i/2} B \mid 1 \leq i \leq t\}$$

is a $(v, k, 1)$ difference family.

The existence of $(v, 3, 1)$ difference families has long been proven for all $v \equiv 1 \pmod{6}$, v a prime power [?]. Recently, existence results for $(v, 4, 1)$ and $(v, 5, 1)$ difference families, $v \equiv 1 \pmod{12}$ and $v \equiv 1 \pmod{20}$ respectively, have been proven for all v a prime power [?].

All the sets of a $(v, k, 1)$ difference system and their translates make up the blocks of a combinatorial design called a cyclic Steiner 2-design. A Steiner 2-design is a collection of points and blocks with k points in every block and every pair of points in exactly one block together. For a set of v points there are exactly

$$b = v(v-1)/k(k-1)$$

blocks that make up the Steiner 2-design and each point is incident on

$$r = (v - 1)/(k - 1)$$

blocks. The Steiner 2-designs from difference sets are cyclic as whenever a, b, c is a triple $a + 1, b + 1, c + 1$ is also a triple.

III. QUASI-CYCLIC LDPC CODES

A set in the difference system and its translates make up the blocks of a square $v \times v$ circulant matrix, so the translates from l sets of a $(v, k, 1)$ difference system can be used to produce a $v \times lv$ parity check matrix

$$H = [A_1, A_2, \dots, A_l] \quad (2)$$

for a quasi-cyclic LDPC code with rate $(l - 1)/l$, and no 4-cycles. It is easy to see that H is regular, with v parity checks lv codeword bits, column weight k and row weight kl . Of course if we choose the maximum possible l , namely

$$l = t = (v - 1)/k(k - 1),$$

then H is the incidence matrix of the corresponding Steiner 2-design.

To generate the circulant matrices for our codes we consider first difference systems from triples ($k = 3$) before considering difference systems with $k = 4, 5$.

A. Quasi-cyclic codes from triple difference systems

The use of the incidence matrix of cyclic Steiner triple systems (STS) for iterative decoding has been investigated in [?] for codes of maximum possible rate while avoiding 4-cycles. If very high rates are not desired we can select a subset of the blocks of a cyclic STS in much the same way that a subset of the resolution classes of a Kirkman triple system can be chosen to give a regular LDPC code [?]. Although the cyclic STS designs are not strictly resolvable, the blocks of a cyclic STS can be grouped into classes of v blocks (which are exactly the translates of one difference set) with each point incident on exactly three of the blocks in a class. So constructing H as in (??) with sets from a $(v, 3, 1)$ difference family is equivalent to choosing l of the classes of a cyclic STS. One method to choose the best subset of the sets in a difference family is to apply Tanner's parity check bound [?] to each possible combination of sets.

Kirkman triple systems can be used to construct codes of any rate

$$R = \frac{r - 3}{r}, \quad r \in \{4, 5, \dots, (v - 1)/2\}$$

while the cyclic STS designs allow for codes of any rate

$$R = (l - 1)/l, \quad l \in \{2, 3, \dots, (v - 1)/6\},$$

so there is a penalty of fewer available rates associated with the benefit of a quasi-cyclic code structure.

To construct cyclic STS designs for all $v \equiv 1 \pmod{6}$ we find m triples $\{a, b, c\}$ which partition the set $\{1, \dots, 3m\}$ such that $a + b = c$ or $a + b + c = 0 \pmod{6m + 1}$. Then the triples $\{0, a, a + b\}$ form a $(6m + 1, 3, 1)$ difference system

and so lead to the construction of a cyclic STS($6m + 1$). The differences arising from the triple $\{0, a, a + b\}$ are $\pm a, \pm b, \pm(a + b)$ i.e. $\pm a, \pm b, \pm c$. So altogether the differences are just the non-zero elements $\pm 1, \dots, \pm 3m$ of Z_{6m+1} . For $m = 0$ or $1 \pmod{4}$ the triples are Skolem triple system of order m and for $m = 2$ or $3 \pmod{4}$ O'Keefe triple systems will provide the required triples (see [?]).

Cyclic Steiner triple systems can also be constructed for all $v \equiv 3 \pmod{6}$ using a construction due to Rosa (see [?]). Although not strictly difference families the modified difference systems from Rosa triples provide exactly the properties required to make up the circulants of a quasi-cyclic code. The differences obtained from a Rosa triple system are precisely all the non-zero elements of Z_{6m+3} other than $2m + 1$ and $4m + 2$. Take also the first $2m + 1$ translates of $\{0, 2m + 1, 4m + 2\}$ and a cyclic STS($6m + 3$) is produced.

B. Code girth

Since 4-cycles are avoided in H the girth of the quasi-cyclic codes from difference families is at least 6. Recently in [?] Tanner showed that all codes with parity check matrices from circulants with column weight ≥ 3 must contain a 6-cycle. So the girth of all the quasi-cyclic codes from subsection ??, is exactly 6. What may also be significant is the number of cycles in the code of length 6 and the structure of the circulant matrices allows us to say something about the number of these cycles (N_6).

Suppose that the cycle occurs through positions $(i_1, j_1), (i_1, j_2), (i_2, j_2), (i_2, j_3), (i_3, j_3), (i_3, j_1)$ in a circulant matrix A with $k \geq 3$. Then, as A is circulant, there is also a 6-cycle through the positions $(i_1 + 1, j_1 + 1), (i_1 + 1, j_2 + 1), (i_2 + 1, j_2 + 1), (i_2 + 1, j_3 + 1), (i_3 + 1, j_3 + 1), (i_3 + 1, j_1 + 1)$ in A , and so on for $\{+1, +2, \dots, +(v - 1)\}$ (addition mod v). So that a circulant matrix A that has a 6-cycle must have at least v of them. If there is another 6-cycle through the points in column $A(1)$ other than the 3 through column $A(1)$ from the v above it must also occur in $v - 1$ cyclic shifts by the same argument.

Lemma 2: For a circulant matrix A from a set in $(v, 3, 1)$ difference family,

$$N_6(A) \in \{v, 2v, 3v, 4v\}$$

Proof (sketch): As above, for each 6-cycle through the points in column $A(1)$ there are a total of $v/3$ 6-cycles in A . Further, as $r = 3$ a pair of points can be involved in a maximum of four 6-cycles without also forming 4-cycles, and we have $\max\{N_6(A)\} = 3.4.v/3$. ■

Only if the circulant matrix is from the $(v, 3, 1)$ difference set will the maximum of $4v$ 6-cycles occur (to see this note that for the translates of a difference set every pair of points occurs exactly once in A).

Lemma 3: For a quasi-cyclic matrix $H = [A_1, \dots, A_l]$ from l sets of $(v, 3, 1)$ difference family,

$$N_6 \in \{lv, (l + 1)v, \dots, 2t(3l - 1)v\}.$$

Proof (sketch): Again, for each 6-cycle through the points in column $A(1)$ there are a total of $v/3$ 6-cycles in H . A 6-cycle in H may involve multiple circulants,

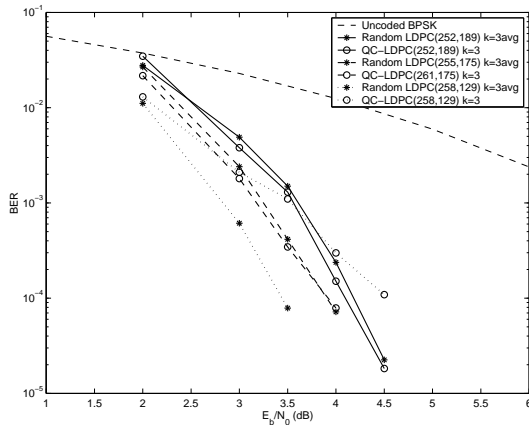


Fig. 1. BER vs. E_b/N_0 for small LDPC codes, max. its. = 50

however the $v - 1$ shifts associated with it will always include three pairs of points from the set of columns $H_1 = \{A_1(1), \dots, A_l(1)\}$. Now $r = 3l$ and a pair of points can be involved in a maximum of $2(3l - 1)$ 6-cycles without causing a 4-cycle in the matrix. Thus we have a maximum of $3l \cdot 2(3l - 1)$ 6-cycles through the points in the combined set of columns H_1 each of which results in $v/3$ 6-cycles in H . ■

The maximum number of 6-cycles will always occur if we take the maximum l , in which case H is the incidence matrix of a Steiner system. The incidence matrix of any Steiner triple system has exactly $(k - 1)(r - 1)b$ 6-cycles. (To see this note that every pair of points must occur exactly once in H so each pair is involved in exactly $(k - 1)(r - 1)$ 6-cycles.)

C. Quasi-cyclic codes from column splitting

6-cycles can be avoided in a circulant matrix if the column weight is less than 3. However, LDPC codes with many columns of weight 2 have bad distance properties and are generally believed not to perform well when iteratively decoded. For this reason MacKay and Neal in [?] minimize the number of weight two columns in their randomly constructed sparse parity-check matrices. However, recently MacKay has suggested that it is possible that codes with many more weight 2 columns can be good codes [?]. Here we propose that column splitting of $(v, k, 1)$ difference systems $k > 3$ be used to give quasi-cyclic LDPC codes with a portion of columns of weight 2. The resulting codes we call *nearly regular* LDPC codes as the columns can be one of two weights.

We take the difference set $D = \{d_1, d_2, \dots, d_k\}$ and divide the circulant $a(x) = x^{d_1} + x^{d_2} + \dots + x^{d_k}$ into two circulants

$$\begin{aligned} a_1(x) &= x^{d_1} + x^{d_3} + \dots + x^{d_{k_1}} \\ a_2(x) &= x^{d_2} + x^{d_4} + \dots + x^{d_{k_2}} \end{aligned}$$

where $k_1 + k_2 = k$. Encoding is performed in exactly the same way as for the regular codes and decoding differs only in that a fixed number of the columns in H will be weight 2, so calculation of bit probabilities for those codeword bits will require fewer computations. The row

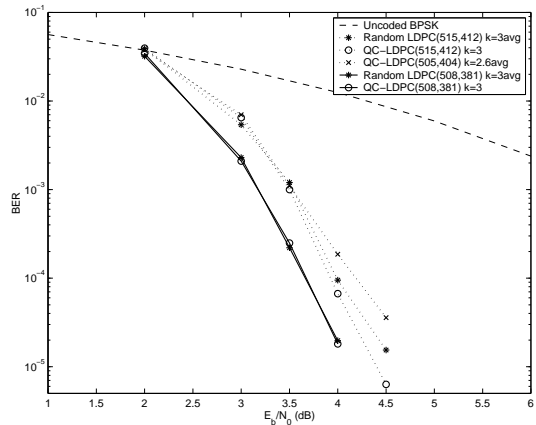


Fig. 2. BER vs. E_b/N_0 for LDPC codes, max. its. = 50

weight of the parity check matrix will be constant for all rows but 1 less for every weight 3 circulant matrix in the code replaced by a weight 2 circulant matrix. Consequently calculation of parity check probabilities will require fewer computations as well. How to optimally select the required circulants is less strait forward for the nearly regular codes since we can no longer apply Tanners parity bound as we did for the regular codes. One option is to choose those circulant matrices which produce only v 6-cycles. Preliminary computer simulations suggest that this strategy does indeed produce codes with significantly fewer than the average number of 6-cycles.

We use $(v, 4, 1)$ and $(v, 5, 1)$ difference families from Buratti [?] for our nearly regular LDPC codes. Buratti defines sets $B = \{1, 0, b, b^2\}$ and $B = \{1, 0, b, b^2, b^3\}$ for $k = 4$ and 5 respectively that provide the system of representatives necessary for Lemma ??.

D. Minimum distance

The avoidance of 4-cycles in the quasi-cyclic LDPC codes guarantees a minimum distance of at least $k_0 + 1$ if each bit in the code is checked by at least k_0 parity checks [?]. If all circulants are weight 3 we have $d_{\min} \geq 4$, however the inclusion of weight two circulants gives $d_{\min} \geq 3$. Unfortunately, for quasi-cyclic codes of rate-1/2 the number of ones in a row of the parity check matrix is an upper bound on the minimum distance of the code [?]. So we have $d_{\min} \leq k_0 + k_1$ where k_0 and k_1 are the weights of the two circulants in a rate half quasi-cyclic code. This limit on the minimum distance of a rate-1/2 code is a possible explanation for the poor error correcting performance of quasi-cyclic codes of this rate.

IV. SIMULATION RESULTS USING ITERATIVE DECODING

We employed belief propagation decoding, also known as sum-product decoding, as presented in [?]. This is similar to Gallager's original algorithm, with the difference that log-likelihood metrics are employed in place of probabilities, reducing the influence of numerical problems on the decoding process.

In the simulation results shown in Fig. 2–4 a number of randomly generated LDPC codes have been created using the construction method from [?], [?]. We see that

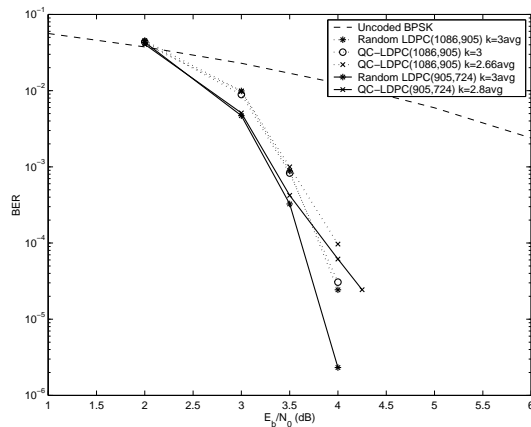


Fig. 3. BER vs. E_b/N_0 for high rate LDPC codes, max. its. = 50

for codes of rate-1/2 the quasi-cyclic codes do not perform as well as the randomly constructed codes. However, for rates-2/3 and above the performance of the quasi-cyclic codes is comparable to that of the random codes of the same rate, codeword length and density.

The quasi-cyclic codes with circulants of weight 2 perform as well as the more dense codes at low signal to noise ratios (SNR) despite their lower decode complexity. Perhaps at these lower SNR values fewer 6-cycles in the codes compensate for the reduction in information passed between nodes in a decode iteration. However, as the SNR increases the codes with circulants of weight 2 do not perform as well as those with all circulants of all weight 3. This may be due to an increase in low weight codewords in the codes with weight 2 circulants. A strategy for selecting circulants based on the weight enumerator of the resulting code may prove useful.

V. CONCLUSION

A large class of quasi-cyclic LDPC codes which perform well when iteratively decoded with the sum-product algorithm have been described. The quasi-cyclic LDPC codes presented show a comparable decoding performance to the randomly constructed LDPC codes with the advantage of a significantly reduced encoding complexity.