Research Letter

Design LDPC Codes without Cycles of Length 4 and 6

Jun Fan,[1] Yang Xiao,[2] and Kiseon Kim[3]

[1] *Office of Academic Affairs, Central University for Nationalities, Beijing 100081, China*
[2] *Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China*
[3] *Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju 500712, South Korea*

Correspondence should be addressed to Yang Xiao, yxiao@bjtu.edu.cn

We present an approach for constructing LDPC codes without cycles of length 4 and 6. Firstly, we design 3 submatrices with different shifting functions given by the proposed schemes, then combine them into the matrix specified by the proposed approach, and, finally, expand the matrix into a desired parity-check matrix using identity matrices and cyclic shift matrices of the identity matrices. The simulation result in AWGN channel verifies that the BER of the proposed code is close to those of Mackay's random codes and Tanner's QC codes, and the good BER performance of the proposed can remain at high code rates.

1. Introduction

LDPC codes can be described by a bipartite graph called Tanner graph [1], and the girth of a Tanner graph is the length of the shortest cycle in the graph. Girths in the Tanner graphs of LDPC codes prevent the sum-product algorithm from converging [2–10]. Further, cycles, especially short cycles, degrade the performance of LDPC decoders, because they affect the independence of the extrinsic information exchanged in the iterative decoding [2, 3]. Hence, LDPC codes with large girth are desired. Most methods for designing LDPC codes are based on random construction techniques, the lack of structure implied by this randomness presents serious disadvantages in terms of storing and accessing a large parity-check matrix, encoding data, and analyzing code performance (e.g., determining a code's distance properties). Recent research results [4–10] about LDPC codes with large girths show that there are many possible ways to construct the LDPC codes with large girths. However, some of the code constructions [4–10] are not satisfied for application due to the complicated constraints for the structures of parity-check matrices. To solve the problem, this letter provides a different construction of LDPC codes with large girth. The column weight of our codes is 3 and the row weight is v ($v > 3$). The BER performance of the proposed codes is near to those of Mackay's random codes and Tanner's QC codes.

This letter is structured as follows. In Section 2, we analyze the figures of 4-cycles and 6-cycles in the parity-check matrix.

Section 3 presents the design algorithm of the parity-check matrix. Section 4 provides three lemmas show that the proposed codes do not contain 4-cycles and 6-cycles. Section 5 evaluates the BER performance of the proposed codes for AWGN channels via computer simulations. Finally, In Section 6 we make some conclusions.

2. Definitions of the Cycles

An (n, j, k) LDPC code has the parity-check matrix H of n columns, where H has j ones in each column, k ones in each row, and zeros elsewhere, then the (n, j, k) parity-check matrix has $m = nj/k$ rows [1, 2]. A bipartite graph with m check nodes and $n - m$ bit nodes can be created with the edges between the bit and check nodes if there are corresponding 1s in the parity-check matrix H. Such a graph is frequently called a Tanner graph [1, 11]. A cycle in a Tanner graph refers to a finite set of connected edges, the edge starts and ends at the same node, and it satisfies the condition that no node (except the initial and final node) appears more than once [4–10, 12]. The length of a cycle is simply the number of edges of the cycle.

A cycle also can be shown in a tree. Figure 1 shows a 4-cycle in a tree, and Figure 2 shows a 6-cycle in a tree. Figures 1
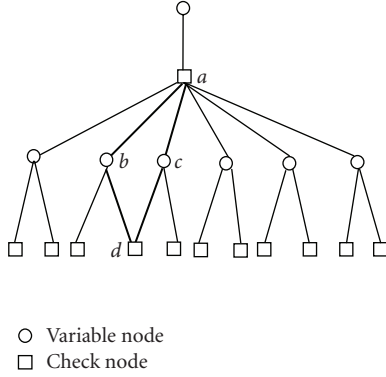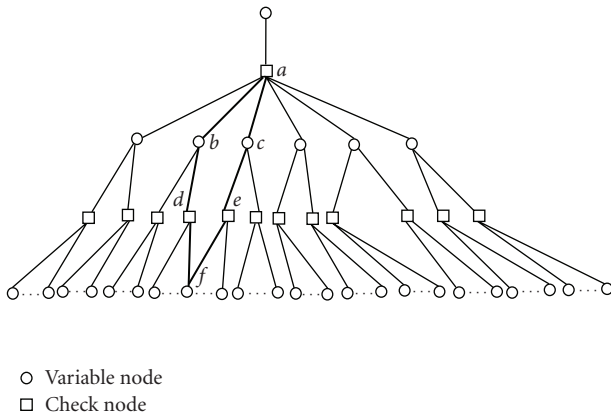
Figure 1: A 4-cycle in tree.

Figure 2: A 6-cycle in tree.

and 2 are not general descriptions for a 4-cycle and 6-cycle in a tree. It is also possible that a 4-cycle or a 6-cycle starts from a variable (bit) node, similar to the trees in Figures 1 and 2. In a tree, check nodes represent rows in parity-check matrix, and variable (bit) nodes represent columns. We try to design a parity-check matrix without 4-cycles and 6-cycles.

2.1. Figures of 4-Cycles in Parity-Check Matrix

In Figure 1, two 1s (we mark them as m_1 and m_2) in row a belong to column b and c, respectively, two 1s (we mark them as m_3 and m_4) in row d belong to column b and c, respectively, so the figure of the 4-cycle in the parity-check matrix as shown in Figure 3. If there is no figure as shown in Figure 3 in parity-check matrix, there is no 4-cycle in the LDPC code. We denote 4-cycles as $G_4(m_1, m_2, m_3, m_4)$.

2.2. Figures of 6-Cycles in Parity-Check Matrix

Figure 4 shows that m_1 and m_2 in row a belong to column b and c, respectively, m_3 in column b belongs to row d, m_4 in column c belongs to row e, m_5 in row d and m_6 in row e belong to column f, where $m_1, m_2, m_3, m_4, m_5, m_6$ represent
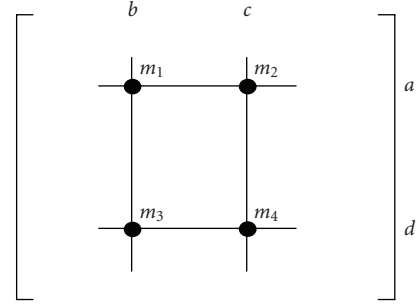
six 1s in 6-cycle, as shown in Figure 4. We denote 6-cycles as $G_6(m_1, m_2, m_3, m_4, m_5, m_6)$.

From Figure 4 we know that six 1s in a 6-cycle belong to 3 rows and 3 columns equally, so we can get the number of 6-cycles with different figures to be 6 from Figure 5.

Figure 5 shows six different kinds of figures of 6-cycles. If there is no figure as shown in Figure 5 in the parity-check matrix, there is no 6-cycle in the LDPC code.

3. Design Algorithm of Parity-Check Matrix

If there is a parity-check matrix H, row weight is v ($v > 3$), column weight is 3. We design three submatrices D, E, F, combine them into matrix H_1, $H_1 = [D\ E\ F]$, transpose H_1 into H_2, expand H_2 into desired parity-check matrix H using identity matrices and cyclic shift matrices of the identity matrix randomly.

3.1. Design Algorithm of Submatrix D

The design algorithm of D is as following.

(1) Design a matrix D_0 with the dimension $v \times v^2$. $D_0(1,1) = D_0(2,1) = D_0(3,1) = \cdots = D_0(v,1) = 1$, other elements are "0"

$$D_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{v \times v^2}. \quad (1)$$

(2) Let $D_1 = D_0^1$, $D_2 = D_0^2$, $D_3 = D_0^3, \ldots, D_{v^2-1} = D_0^{v^2-1}$, where D_0^k represents circularly shift D_0 in (1) for k right-shifting steps circularly.

(3) Let $D = [D_0, D_1, D_2, \ldots, D_{v^2-1}]^T$ and its dimension is $v^3 \times v^2$.

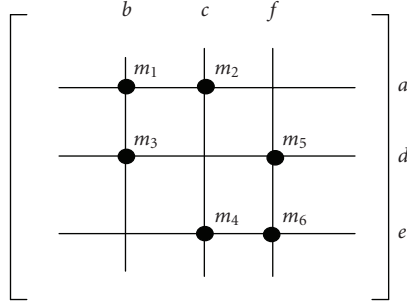Figure 3: A 4-cycle in a parity-check matrix.

FIGURE 4: A 6-cycle in a parity-check matrix.

FIGURE 5: Six kinds of 6-cycles with different figures.

3.2. Design Algorithm of Submatrix E

The design algorithm of E is as following.

(1) Design a matrix E_0 with the dimension $v \times v^2$. $E_0(1,1) = E_0(2,2) = E_0(3,3) = \cdots = E_0(v,v) = 1$, other elements are 0s

$$
E_0 = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{v \times v^2}. \tag{2}
$$

(2) Let $E_1 = [E_0, E_0, \ldots, E_0]^T$, the number of E_0 included in E_1 is v.

(3) Let $E_2 = E_1^v$, $E_3 = E_1^{2v}, \ldots, E_v = E_1^{(v-1)v}$, where E_1^k represents circularly right-shifting E_1 for k steps.

(4) Let $E = [E_1, E_2, \ldots, E_v]^T$ with the dimension $v^3 \times v^2$.

3.3. Design Algorithm of Submatrix F

The design algorithm of F is as following.

(1) Designing a matrix F_0 with dimension $v \times v^2$. $F_0(1,1) = F_0(2, v+1) = F_0(3, 2v+1) = \cdots = F_0(v, v^2 - v + 1) = 1$, other elements are "0." For $v = 4$, we can get F_0 by the following way:

$$
F_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}_{v \times v^2}. \tag{3}
$$

(2) Let $F_1 = F_0^1$, $F_2 = F_0^2$, $F_3 = F_0^3, \ldots, F_{v-1} = F_0^{v-1}$, where F_0^k represents F_0 with circularly right-shifting k steps.

(3) Let $F_v = [F_0, F_1, \ldots, F_{v-1}]^T$.

(4) Let $F = [F_v, F_v, \ldots, F_v]^T$, there are v copies of F_v in F, so the dimension of F is $v^3 \times v^2$.
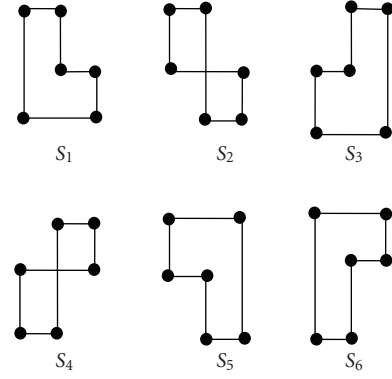
We can get H_1 by the following way:

$$
H_1 = \begin{bmatrix} D_0 & E_0 & F_0 \\ \vdots & \vdots & \vdots \\ D_{v-1} & E_0 & F_{v-1} \\ D_v & E_2 & F_0 \\ \vdots & \vdots & \vdots \\ D_{v^2-1} & E_v & F_{v-1} \end{bmatrix}_{v^3 \times 3v^2}. \tag{4}
$$

Let $H_2 = H_1^T$, then the dimension of H_2 is $3v^2 \times v^3$.

3.4. Expansion of H_2

The expansion algorithm is as follows.

(1) Select an identity matrix I with dimension $p \times p$,

$$
I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{p \times p}. \tag{5}
$$

(2) Let $I_1 = I^1$, $I_2 = I^2, \ldots, I_{p-1} = I^{p-1}$, where I^k represents the matrix I with circularly right-shifting k steps.

(3) Exchanging the 1s in H_2 by the elements in the matrix set $\{I, I_1, I_2, \ldots, I_{p-1}\}$ randomly and exchanging 0s by null matrices with the same dimension as I, then we get the parity-check matrix H, whose dimension is $3pv^2 \times pv^3$.

From the structure of the parity-check matrix H, we can see that the proposed code is a deterministic LDPC code, so the proposed code has the advantage for encoding. To get the LDPC code with the large length $(3pv^2)$, we can expand the H by selecting the desired dimension p of the identity matrix I in (5) with stochastically shifting.

4. The Demonstration of the Code with Girth 8

In Section 3, we present a method for constructing parity-check matrix H with girth 8 based on the submatrices'

shifting. In this section, we add three lemmas with theoretic proofs showing that the parity-check matrix H does not contain 4-cycles and 6-cycles.

Lemma 1. *If a matrix H_1 is constructed by* (4)*, there is no 4-cycle in H_1.*

Proof. We can know from Figure 3 that if there are 4-cycles in the matrix H_1 in (4), the edges connecting $\{m_1, m_3\}$ and $\{m_2, m_4\}$ must be in two different submatrices of D, E, F, because m_1 and m_2 are in the same row. The following formula must be true:

$$L(m_1, m_3) = L(m_2, m_4). \tag{6}$$

In the submatrix D, the lengths $L(m_1, m_3)$ and $L(m_2, m_4)$ could be $1, 2, \ldots, v - 1$. In the submatrix E, the lengths $L(m_1, m_3)$ and $L(m_2, m_4)$ could be $v, 2v, \ldots, (v - 1) \times v$. In the submatrix F, the lengths $L(m_1, m_3)$ and $L(m_2, m_4)$ could be $v^2, 2v^2, \ldots, (v - 1) \times v^2$. We can easily know that $L(m_1, m_3) \neq L(m_2, m_4)$, so there is no 4-cycle in the matrix H_1. $\square$

Lemma 2. *If a matrix H_1 is constructed by* (4)*, there is no 6-cycle in H_1.*

Proof. As shown in Figure 4, if there are 6-cycles in the matrix H_1, the edges connecting $\{m_1, m_3\}$, $\{m_2, m_4\}$, and $\{m_5, m_6\}$ must be in D, E, and F, respectively. Without loss of generality, we assume that $L(m_2, m_4)$ is the longest length among the three lengths. The following formula must be true:

$$L(m_1, m_3) + L(m_5, m_6) = L(m_2, m_4). \tag{7}$$

In the submatrix D, the lengths $L(m_1, m_3)$, $L(m_5, m_6)$ and $L(m_2, m_4)$ could be $1, 2, \ldots, v - 1$. In the submatrix E, the length of $L(m_1, m_3)$, $L(m_5, m_6)$, and $L(m_2, m_4)$ could be $v, 2v, \ldots, (v - 1) \times v$. In the submatrix F, the lengths $L(m_1, m_3)$, $L(m_5, m_6)$, and $L(m_2, m_4)$ could be $v^2, 2v^2, \ldots, (v - 1) \times v^2$. We can know that the edge of $L(m_2, m_4)$ must be in F, or else

$$L(m_1, m_3) + L(m_5, m_6) > L(m_2, m_4). \tag{8}$$

We know that

$$1 + v \leq L(m_1, m_3) + L(m_5, m_6) \leq (v - 1)(v + 1), \tag{9}$$

and $L(m_2, m_4) \geq v^2$, so

$$L(m_1, m_3) + L(m_5, m_6) < L(m_2, m_4). \tag{10}$$

Thus there is no 6-cycle in the matrix H_1, which means H to be free from 6-cycles. $\square$

We know that $H_2 = H_1^T$, so the matrix H_2 is also free of 4-cycles and 6-cycles.

Lemma 3. *If there is no 4-cycle and 6-cycle in the matrix H_2, there is no 4-cycle and 6-cycle in the parity-check matrix H.*

TABLE 1: Parameters of example codes with rate-1/2.

	k	j	N	R	g
New · 1080 · 540	6	3	1080	1/2	8
Mac · 1074 · 537	6	3	1008	1/2	6
QC · 1074 · 537	6	3	1002	1/2	8

TABLE 2: Parameters of our codes with high-rate.

	k	j	N	R	g
New · 5103 · 1701	9	3	5103	0.67	8
New · 5000 · 1500	10	3	5000	0.7	8
New · 5184 · 1296	12	3	5184	0.75	8

Proof. If there is a 4-cycle $G_4(m_1, m_2, m_3, m_4)$ in the parity-check matrix H, from Figure 3 we can know that m_1, m_2, m_3, and m_4 must be in four different identity matrices or cyclic shift matrices of identity matrices. Since an identity matrix I and a cyclic shift matrix of the I in H means a "1" in H_2, there is a 4-cycle in H_2. However, we know that there is no 4-cycle in H_2, so there is no 4-cycle in the parity-check matrix H. Using the same method, we can know that there is no 6-cycle $G_6(m_1, m_2, m_3, m_4, m_5, m_6)$ in the parity-check matrix H. $\square$

5. Performance Evaluation

To demonstrate the error-correcting performance, we constructed two rate-1/2 LDPC codes by the proposed method. For the purpose of comparison, we also construct two classes of Tanner's QC codes and Mackay's random codes [10, 11, 13]. We get Mackay's random codes from [13]. Tanner's QC codes are constructed by the method introduced in Tanner's paper [10, 11]. Both Mackay's random codes and constructed Tanner's QC codes for the comparison are good codes. The selected codes' parameters are given in Table 1.

Table 1 lists the typical values of the row weight k, the column weight j, the code length N, the code rate R, and the girth g for the three codes. The girths of the LDPC codes in Table 1 are tested by the approach of [12]. Table 2 lists the typical values of the proposed codes for different high code and code lengths. In Tables 1 and 2, the New · n · m means the proposed code with the code length n and m parity check bits, so do the Mac · n · m and Qc · n · m.

We simulate the proposed code's error-correcting performance with the assumption that each code is modulated by BPSK and transmitted over additive white Gaussian noise (AWGN) channel. All the codes are decoded with the sum-product algorithm [2, 3]. Figure 6 shows the simulated BER versus signal-to-noise ratio (E_b/N_0). As shown in Figure 6. The BER performance of the proposed codes is very close to MacKay's random codes and Tanner's QC codes. From Figure 7, it can be seen that the proposed codes also have a good BER performance at high code rates 0.67, 0.7, and 0.75. The BER curves of the two codes of at high code rates 0.67, 0.7 are close since two LDPC codes with different code lengths may have approximate minimum weights [10]. In the
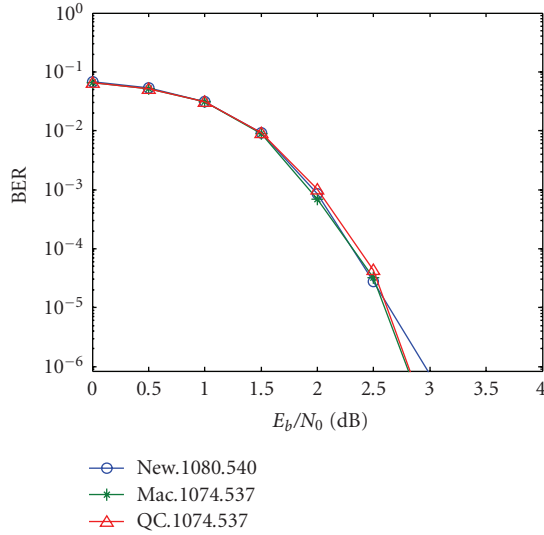
FIGURE 6: BER performance comparison of our code, Mackay's random code and Tanner's QC code.
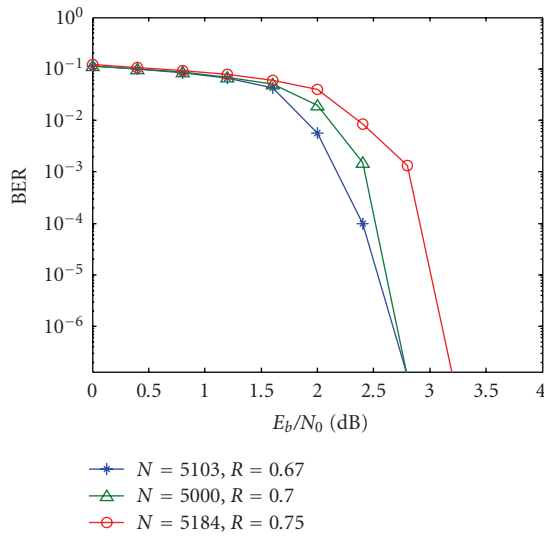

FIGURE 7: BER performance of our codes at high code rate.

expansion algorithm of the proposed code, we replace "1" in H_2 by the elements in the set $\{I, I_1, I_2, \ldots, I_{p-1}\}$ randomly, which will lead to the minimum weights of the obtained codes to be different, the shorter codes may have larger minimum weights.

6. Conclusions

In this paper, we proposed a QC LDPC code without girth-4 and girth-6, three lemmas are provided to prove the short girths' properties of the proposed codes. Simulation verified the good error-correcting performance of the proposed code, whose BER performance is near to those of Tanner's QC codes and MacKay's random codes [10, 11, 13]. The good BER performance can remain at high code rates.

References

[1] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[3] R. Lucas, M. P. C. Fossorier, Y. Kou, and S. Lin, "Iterative decoding of one-step majority logic decodable codes based on belief propagation," *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 931–937, 2000.

[4] S. Kim, J.-S. No, H. Chung, and D.-J. Shin, "Girth analysis of Tanner's (3,5) QC LDPC codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '05)*, pp. 1632–1636, Adelaide, SA, Australia, September 2005.

[5] J. Lu, J. M. F. Moura, and U. Niesen, "Grouping-and-shifting designs for structured LDPC codes with large girth," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '04)*, p. 236, Chicago, Ill, USA, June-July 2004.

[6] B. Vasic, K. Pedagani, and M. Ivkovic, "High-rate girth-eight low-density parity-check codes on rectangular integer lattices," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1248–1252, 2004.

[7] H. Song, J. Liu, and B. V. K. Vijaya Kumar, "Large girth cycle codes for partial response channels," *IEEE Transactions on Magnetics*, vol. 40, no. 4, pp. 3084–3086, 2004.

[8] J. Lu, J. M. F. Moura, and U. Niesen, "A class of structured LDPC codes with large girth," in *Proceedings of IEEE International Conference on Communications*, vol. 1, pp. 425–429, Paris, France, June 2004.

[9] H. Zhang and J. M. F. Moura, "The design of structured regular LDPC codes with large girth," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 7, pp. 4022–4027, San Francisco, Calif, USA, December 2003.

[10] Y. Xiao and M. H. Lee, "Construction of good quasi-cyclic LDPC codes," in *Proceedings of IET International Conference on Wireless Mobile and Multimedia Networks (ICWMMN '06)*, p. 172, Hangzhou, China, November 2006.

[11] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. Costello Jr., "LDPC block and convolutional codes based on circulant matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 2966–2984, 2004.

[12] J. Fan and Y. Xiao, "A method of counting the number of cycles in LDPC codes," in *Proceedings of the 8th International Conference on Signal Processing (ICSP '06)*, vol. 3, pp. 2183–2186, Guilin, China, November 2006.

[13] D. J. C. MacKay, "Encyclopedia of sparse graph codes [EB/OL]," http://www.inference.phy.cam.ac.uk/mackay/codes/data.html .