

# Decoding of LDPC Convolutional Codes with Rational Parity-Check Matrices from a New Graphical Perspective

Jian-Jia Weng, Chih-Chieh Lai, and Chung-Hsuan Wang  
 Department of Electrical Engineering, National Chiao Tung University  
 Hsinchu, Taiwan 30010, R.O.C.  
 chaselweng.cm95g@nctu.edu.tw, chwang@mail.nctu.edu.tw

**Abstract**—Previous studies on low-density parity-check convolutional codes (LDPC-CC) reveal that LDPC-CC with rational parity-check matrices (RPCM) suffer from the unaffordable decoding latency/complexity due to the infinite memory order and the poor bit-error-rate performance due to the existence of length-4 cycles in the Tanner graph. However, in this paper, we show that every LDPC-CC with RPCM can be associated with an equivalent Tanner graph which can avoid the infinite memory order and undesired short length cycles but still implements the same constraints specified by the RPCM. Together with the iterative decoding based on belief propagation with proper scheduling, simulation results indicate that LDPC-CC with RPCM can also provide satisfactory decoding performance.

## I. INTRODUCTION

Low-density parity-check (LDPC) codes were originally invented by Gallager in early 1960s [1]. Due to no practical soft decoding strategy at that time, this class of codes were ignored about thirty years. Until 1996, LDPC codes were rediscovered by Mackay and Neal with a graphical representation introduced by Tanner [2]. In their pioneer studies [3], [4], LDPC codes have been shown to achieve capacity-approaching performance with sufficiently long block length and the iterative decoding based on belief propagation (BP) [5]. Recently, a variant class of LDPC codes, called LDPC convolutional codes (LDPC-CC), were proposed in [6], [7]. With the quasi-cyclic LDPC codes (QC-LDPC) [8]-[11] as the counterparts, LDPC-CC can possess the structural properties on their parity-check matrices, which lend themselves well to an efficient high-speed very-large-scale-integration (VLSI) implementation. In addition, various algebraic constructions of LDPC-CC in [12]-[14] also provide significant improvement of the code performance.

In the literature, studies on LDPC-CC are focused on polynomial parity-check matrices (PPCM). A special type of pipeline decoder suitable for practical implementation was proposed in [13] for LDPC-CC with PPCM, which behaves similarly to the sliding-window decoder for ordinary convolutional codes but performs the iterative BP decoding on the Tanner graph. Design guidelines for LDPC-CC with monomial/binomial-term PPCM also guarantee the remarkable bit-error-rate (BER) performance for data protection [12]-[14]. However, LDPC-CC with rational parity-check matrices (RPCM) have been ignored for a long time due to the following fatal defects. First, from the viewpoint of conventional pipeline decoder, LDPC-CC with RPCM will incur an infinite memory order, hence resulting in unaffordable decoding latency and complexity. Moreover, author in [13] demonstrated that the Tanner graph of LDPC-CC with any trinomial (or

higher order)-term parity-check matrix will has a girth less than or equal to 6. This result implies that the rational entries in a parity-check matrix will generate cycles of length 4 in the Tanner graph, thus ruining the performance of iterative decoding.

In this paper, different from the previous results, we discover LDPC-CC with RPCM can still achieve good decoding performance as long as a suitable graph is provided for iterative decoding. To generate the desired graph for decoding, a general procedure is proposed to transform the Tanner graph obtained by the direct mapping from RPCM into an equivalent graph, which can avoid the infinite memory order and undesired short cycles. Based on the equivalent graph, we also show that the conventional pipeline decoder can be used to decode LDPC-CC with RPCM without the fatal defects mentioned above. Finally, simulation results are given to verify the superiority of our proposed scheme over the conventional one.

The rest of this paper is organized as follows. In Section II, a brief review of LDPC-CC is given. The new perspective for decoding LDPC-CC with RPCM is described in Section III. Several examples as well as their simulation results are provided for verification. Finally, a summary is drawn in Section IV to conclude this work.

## II. REVIEW OF LDPC-CC

Let  $F$  be a finite field and  $F((D))$  be the field consisting of all one-sided formal Laurent series of the form  $\sum_{i \geq m} a_i D^i$  with the indeterminate  $D$  standing for time delay, where  $a_i \in F$  for all  $i$  and  $m$  can be any finite integer. The set of all polynomials over  $F$  is denoted by  $F[D]$ . For  $x(D) \in F[D]$ , the weight of  $x(D)$ , denoted by  $w(x(D))$ , is defined as the number of the nonzero coefficients in  $x(D)$ . Also, denote by  $\deg(x(D))$  the degree of  $x(D)$ . Every rational function  $p(D)/q(D)$ , where  $p(D), q(D) \in F[D]$  and  $q(D) \neq 0$ , has a unique Laurent series expansion and is called a rational Laurent series. The rational subfield of  $F((D))$  consists of all rational Laurent series and is denoted by  $F(D)$ . An LDPC-CC  $\mathcal{C}$  over  $F$  is defined as the null space of a parity-check matrix  $\mathbf{H}(D)$  which is in general over  $F(D)$  and has a sparse scalar form. For  $\mathcal{C}$  with PPCM, its  $\mathbf{H}(D)$  has  $c$  columns and  $(c-b)$  rows and is of the following form:

$$\mathbf{H}(D) = \begin{pmatrix} h_{1,1}(D) & h_{1,2}(D) & \cdots & h_{1,c}(D) \\ h_{2,1}(D) & h_{2,2}(D) & \cdots & h_{2,c}(D) \\ \vdots & \vdots & \ddots & \vdots \\ h_{c-b,1}(D) & h_{c-b,2}(D) & \cdots & h_{c-b,c}(D) \end{pmatrix} \quad (1)$$

where  $h_{i,j}(D)$  is in  $F[D]$ ,  $\forall i, j$ . Let  $m_s$  denote the memory order of  $\mathbf{H}(D)$ , i.e.,  $m_s = \max_{i,j} \deg(h_{i,j}(D))$ . A vector  $\mathbf{v}(D) = (v_1(D), v_2(D), \dots, v_c(D))$  (in general over  $F(D)$ ) is a codeword of  $\mathcal{C}$  if and only if  $\mathbf{v}(D) \cdot \mathbf{H}^T(D) = \mathbf{0}$ . Without loss of generality, we assume that the encoding process begins at time-0. Alternatively, we can decompose  $\mathbf{H}(D)$  into a superposition of  $m_s$   $(c-b) \times c$  scalar matrices with different time delays, i.e.,  $\mathbf{H}(D) = \mathbf{H}_0 + \mathbf{H}_1 D + \dots + \mathbf{H}_{m_s} D^{m_s}$ , and then obtain the following scalar parity-check matrix:

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_0 & & & & & & \\ \mathbf{H}_1 & \mathbf{H}_0 & & & & & \\ \vdots & \mathbf{H}_1 & \mathbf{H}_0 & & & & \\ \mathbf{H}_{m_s} & \vdots & \mathbf{H}_1 & \ddots & & & \\ & \mathbf{H}_{m_s} & \vdots & \ddots & & & \\ & & & \mathbf{H}_{m_s} & \ddots & & \\ & & & & & \ddots & \\ & & & & & & \ddots \end{pmatrix} \quad (2)$$

where the blank area stands for the zero entries for convenience. By the substitution of  $\mathbf{v}(D) = \sum_{t \geq 0} \mathbf{v}_t D^t$ , where  $\mathbf{v}_t = (v_t^{(0)}, v_t^{(1)}, \dots, v_t^{(c-1)})$  denotes the vector of  $c$  output coded bits at time- $t$ , the constraint  $\mathbf{v}(D) \cdot \mathbf{H}^T(D) = \mathbf{0}$  implies

$$\mathbf{v}_t \mathbf{H}_0^T + \mathbf{v}_{t-1} \mathbf{H}_1^T + \dots + \mathbf{v}_{t-m_s} \mathbf{H}_{m_s}^T = \mathbf{0}, \forall t \geq m_s. \quad (3)$$

To achieve good performance for data protection, LDPC-CC are usually designed with  $m_s$  larger than one hundred [12]-[14]. With such a choice of  $m_s$ , the rapid growth of the number of the states on the trellis of LDPC-CC makes the optimal decoding infeasible. Conventional studies suggest that the iterative decoding with belief propagation on the Tanner graph obtained from the scalar parity-check matrix  $\mathbf{H}$  in (2) serves as an effective alternative for the decoding of LDPC-CC. Since the scalar parity-check matrix of the LDPC-CC is infinite and repeated, the corresponding Tanner graph extends infinitely with repeated nodes in the same structure that satisfies (3). As observed in (2), every parity-check equation checks the coded bits only within the interval of  $(m_s + 1)$  time units. Hence, the messages exchanged in one iteration are only across an interval that consists of at most  $(m_s + 1)$  time units. This nature gives the well-known pipeline decoder [12] for LDPC-CC. To perform  $I$  iterations of decoding, the pipeline decoder are equipped with a serial concatenation of  $I$  processors; each processor updates the messages of the variable nodes and check nodes within the interval of  $(m_s + 1)$  time units in the Tanner graph. After the received symbols are successively updated by  $I$  processors, the decoded results after  $I$  iterations are then obtained.

For example, consider a binary LDPC-CC with the following PPCM of  $m_s = 3$  [13]:

$$\mathbf{H}(D) = \begin{pmatrix} 1 & D & D^3 \\ D^3 & D^2 & 1 \end{pmatrix}. \quad (4)$$

Decomposing (4) into  $\mathbf{H}_0 + \mathbf{H}_1 D + \mathbf{H}_2 D^2 + \mathbf{H}_3 D^3$ , we can obtain the scalar parity-check matrix  $\mathbf{H}$  by (2). By (3),  $v_t^{(i)}$ 's

now satisfy

$$\begin{cases} v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus v_{t-3}^{(2)} = 0 \\ v_t^{(2)} \oplus v_{t-2}^{(1)} \oplus v_{t-3}^{(0)} = 0 \end{cases} \quad (5)$$

where  $\oplus$  stands for the modulo-2 addition. In this case, each processor operates on a subgraph of the Tanner graph corresponding to (5) that consists of 12 variable nodes and 8 check nodes as well as their edges within an interval of 4 time units, as shown in Fig. 1. Particularly, only the first 2 check nodes and the last 3 variable nodes in each processor can be activated to exchange the messages with the other nodes in the same processor. This feature makes that the decoder can be implemented in a parallel fashion. In addition, each processor has to store the updated messages of the related variable and check nodes. Each time every 3 received symbols are fed into the memory elements of the first 3 variable nodes in processor-1. Simultaneously, the memory contents already existing in the decoder are shifted left with one time unit. Then, all messages along the edges connected to the active check nodes and the active variable nodes are successively updated by the standard BP updating equations. Once the last 3 active variable nodes in processor- $I$  are processed, we then output the decoded bits by making hard decisions.

Consequently, there are two important parameters affecting the performance of the pipeline decoder for LDPC-CC. One is the value of  $m_s$ , which determines the storage requirement and the decoding latency, as illustrated in the above example. For  $\mathbf{H}(D)$  of a large  $m_s$ , the decoder will require a large number of memory elements and induce a long decoding latency. The other is the girth of the Tanner graph. Since the decoder executes the BP algorithm on the Tanner graph, if the graph contains too many short cycles, the BER performance will suffer a serious degradation. Some researchers have shown that a PPCM contains entries with weight larger than two can not have a Tanner graph with girth larger than six [13]. For this reason, most of the studies on LDPC-CC only pay attention to PPCM with monomials or binomials to avoid the undesired short-length cycles.

### III. A NOVEL VIEWPOINT ON TANNER GRAPH FOR DECODING LDPC-CC WITH RPCM

To decode LDPC-CC with RPCM, conventional studies suggest that rational entries in the parity matrix are first expanded to the form of Laurent series. However, such a viewpoint induces the following two critical problems impairing the decoding performance. First, once a rational entry is expanded into the form of Laurent series, the resulting  $m_s$  of parity-check matrix becomes infinity such that the pipeline decoder will require unaffordable decoding latency and complexity, even for  $I = 1$ . In addition, the corresponding Tanner graph will have girth 4 and is not suitable for the iterative BP decoding. Therefore, LDPC-CC with RPCM have been ignored in the previous works.

In this section, we try to give a first attempt to show that LDPC-CC with RPCM can still achieve good decoding performance as long as a suitable graph is provided for iterative decoding. To avoid decoding on the original Tanner graph with the undesired short cycles and infinite memory order, a graph

transformation as illustrated below is first introduced to obtain an equivalent graph with a larger girth and a finite memory order but still implements the same constraints specified by the RPCM. Consider a binary LDPC-CC  $\mathcal{C}_1$  with the following RPCM:

$$\begin{pmatrix} 1 & D & \frac{D^3}{1+D^3} \\ D^3 & D^2 & 1 \end{pmatrix}. \quad (6)$$

By expanding  $D^3/(1+D^3)$  into the form of Laurent series, i.e.,  $\sum_{i \geq 1} D^{3 \cdot i}$ , the RPCM in (6) can be rewritten as

$$\begin{pmatrix} 1 & D & \sum_{i \geq 1} D^{3 \cdot i} \\ D^3 & D^2 & 1 \end{pmatrix} \quad (7)$$

from the conventional viewpoint. By (7),  $v_t^{(i)}$ 's are required to satisfy

$$v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus \left( \sum_{i \geq 1} v_{t-3 \cdot i}^{(2)} \right) = 0 \quad (8)$$

and

$$v_{t-3}^{(0)} \oplus v_{t-2}^{(1)} \oplus v_t^{(2)} = 0 \quad (9)$$

$\forall t \geq 3$ . Denote by  $c_t$  the check node corresponding to the parity-check equation in (8) in the Tanner graph. Due to the infinite series in (8), unfortunately, not only is an infinite  $m_s$  obtained, but also many length-4 cycles are contained in the Tanner graph based on (8), as shown in Fig. 2. The pipeline decoder is thus unworkable in this case. To eliminate the undesired short cycles in Fig. 2 but maintains the same constraints specified by the RPCM, we define  $d_t = \sum_{i \geq 1} v_{t-3 \cdot i}^{(2)}$  and then split (8) into the following two parity-check equations:

$$\begin{cases} v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus d_t & = 0 \\ d_t \oplus \left( \sum_{i \geq 1} v_{t-3 \cdot i}^{(2)} \right) & = 0. \end{cases} \quad (10)$$

By (9) and (10), an equivalent Tanner graph can be constructed with a larger girth 8 and a smaller memory order 3, as shown in Fig. 4.

To illustrate how to transform the Tanner graph, we first inspect on the parity-check equations corresponding to  $c_t$  and  $c_{t-3}$  for simplicity. The two parity-check equations associated with  $c_t$  and  $c_{t-3}$  have a common part  $\sum_{i \geq 2} v_{t-3 \cdot i}^{(2)}$ , which causes the undesired length-4 cycles in the original Tanner graph. We observe that removing the common part in the parity-check equations is equivalent to eliminating the short-length cycles in the Tanner graph. With the aid of  $d_t$ 's, the parity-check equations corresponding to  $c_t$  and  $c_{t-3}$  can be rewritten as

$$v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus v_{t-3}^{(2)} \oplus d_{t-3} = 0 \quad (11)$$

and

$$v_{t-3}^{(0)} \oplus v_{t-4}^{(1)} \oplus d_{t-3} = 0 \quad (12)$$

respectively. Since the two parity-check equations in (11) and (12) have no two (or more) terms in common, we are sure that all the length-4 cycles are eliminated. On the other hand,

substituting  $d_t$  by  $v_{t-3}^{(2)} \oplus d_{t-3}$  into  $v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus d_t = 0$  in (10), we have

$$\begin{aligned} 0 &= v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus \left( v_{t-3}^{(2)} \oplus d_{t-3} \right) \\ &= v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus \left( v_{t-3}^{(2)} \oplus v_{t-6}^{(2)} \oplus d_{t-6} \right) \\ &\vdots \\ &= v_t^{(0)} \oplus v_{t-1}^{(1)} \oplus \sum_{i \geq 1} v_{t-3 \cdot i}^{(2)}. \end{aligned} \quad (13)$$

Indeed, this is exactly the same as (8). The correctness of decoding  $\mathcal{C}_1$  based on the equivalent graph is hence assured.

In addition, we can combine (9) and (10) to form the following equivalent parity-check matrix:

$$\mathbf{H}_{\text{eq}}(D) = \begin{pmatrix} 1 & D & 0 & 1 \\ 0 & 0 & D^3 & 1+D^3 \\ D^3 & D^2 & 1 & 0 \end{pmatrix} \quad (14)$$

which generates another LDPC-CC  $\mathcal{C}_2$  of lower code rate and longer length. Let  $d(D) = \sum_{t \geq 0} d_t D^t$  and  $\mathbf{v}_{\text{eq}}(D) = [v(D) \ d(D)] = [v^{(0)}(D) \ v^{(1)}(D) \ v^{(2)}(D) \ d(D)]$ . It is clear that  $\mathbf{v}_{\text{eq}}(D)$  is a legitimate codeword of  $\mathcal{C}_2$ . Furthermore,  $v(D)$  can be obtained from  $\mathbf{v}_{\text{eq}}(D)$  by puncturing  $d(D)$  away.  $\mathcal{C}_1$  can hence be viewed as a child code punctured from the mother code  $\mathcal{C}_2$ . Suppose we now decode  $\mathcal{C}_1$  based on the Tanner graph corresponding to  $\mathbf{H}_{\text{eq}}(D)$  of  $\mathcal{C}_2$ . Conventional iterative decoding scheme with proper scheduling [15] for punctured LDPC codes can then be employed here to improve the decoding performance. Revealed by the simulation results in Fig. 3, the decoding based on the parity-check matrix in (7) has a very poor performance. In contrast, decoding on the equivalent graph can attain an apparent performance improvement.

As shown in the above example, the graph transformation is to generate an equivalent lower rate mother code, which has a Tanner graph not only satisfying the constraints specified by the original RPCM but also guaranteeing the advantage of finite memory order and a large girth. LDPC-CC with RPCM can then be successfully decoded based on the new graph without the fatal drawbacks mentioned above. Below, a general procedure performing the graph transformation is given. Without loss of generality, we assume that the RPCM  $\mathbf{H}(D)$  contains only one rational entry, say  $h_{1,c}(D) = p(D)/q(D)$  in (1), where  $p(D) = \sum_i p_i D^i$  and  $q(D) = \sum_i q_i D^i$ .

#### Procedure 1:

- Step 1. Set  $\mathbf{H}^* = \mathbf{H}(D)$ . Let  $h_{i,j}^*(D)$  represent the  $(i, j)$ -entry of  $\mathbf{H}^*(D)$ ,  $\forall i, j$ . Let  $\{\kappa^{(t)}(D)\}_{t \geq 0}$  be a sequence of polynomials, in which  $\kappa^{(0)}(D) = q(D)$ . Set  $s = 0$ .
- Step 2. Let  $\boldsymbol{\alpha} = (1, 0, \dots, 0)^T$  be a  $(c-b) \times 1$  column vector and  $\boldsymbol{\beta} = (0, \dots, 0, p(D), q(D))$  be a  $1 \times (c+1)$  row vector. Set

$$\mathbf{H}^*(D) = \begin{pmatrix} \mathbf{H}^*(D) & \boldsymbol{\alpha} \\ \boldsymbol{\beta} & \end{pmatrix}. \quad (15)$$

Furthermore, let  $h_{1,c}^*(D) = 0$  in (15).

- Step 3. Let  $j$  be the lowest power of  $D$  of the nonzero coefficient in  $\kappa^{(s)}(D)$ . Set  $s = s + 1$  and  $\kappa^{(s)}(D) =$

$\kappa^{(s-1)}(D) - D^j$ . Let  $\alpha = (0, \dots, 0, 1)^T$  be a  $(c-b+s) \times 1$  column vector and

$$\beta = (\underbrace{0, \dots, 0}_c, \kappa^{(s)}(D), \underbrace{0, \dots, 0}_{s-1}, 1)$$

be a  $1 \times (c+s+1)$  row vector. Obtain a new  $\mathbf{H}^*(D)$  by (15). Set  $h_{c-b+s, c+1}^*(D) = D^j$ . Repeat Step 3 until  $w(\kappa^{(s)}(D)) \leq 1$ .

*Step 4.* If  $w(h_{c-b+1, c+1}^*(D)) = 1$ , then go to Step 6. Otherwise, let  $j$  be the lowest power of  $D$  of the nonzero coefficient in  $p(D)$ . Set  $s = s + 1$  and  $\kappa^{(s)}(D) = p(D) - D^j$ . Let

$$\alpha = (\underbrace{0, \dots, 0}_{c-b}, 1, \underbrace{0, \dots, 0}_{s-1}, 0)^T$$

be a  $(c-b+s) \times 1$  column vector and

$$\beta = (\underbrace{0, \dots, 0}_{c-1}, \kappa^{(s)}(D), \underbrace{0, \dots, 0}_s, 1)$$

be a  $1 \times (c+s+1)$  row vector. Obtain a new  $\mathbf{H}^*(D)$  by (15). Set  $h_{c-b+1, c}^*(D) = D^j$ . If  $w(\kappa^{(s)}(D)) = 1$ , then go to Step 6.

*Step 5.* Let  $j$  be the lowest power of  $D$  of the nonzero coefficient in  $\kappa^{(s)}(D)$ . Set  $s = s + 1$  and  $\kappa^{(s)}(D) = \kappa^{(s-1)}(D) - D^j$ . Let  $\alpha = (0, \dots, 0, 1)^T$  be a  $(c-b+s-1) \times 1$  column vector and

$$\beta = (\underbrace{0, \dots, 0}_{c-1}, \kappa^{(s)}(D), \underbrace{0, \dots, 0}_s, 1)$$

be a  $1 \times (c+s+1)$  row vector. Obtain a new  $\mathbf{H}^*(D)$  by (15). Set  $h_{c-b+s+1, c}^*(D) = D^j$ . Repeat Step 5 until  $w(\kappa^{(s)}(D)) \leq 1$ .

*Step 6.* Set  $\mathbf{H}_{\text{eq}}(D) = \mathbf{H}^*(D)$ .

Once the equivalent parity-check matrix  $\mathbf{H}_{\text{eq}}(D)$  is obtained, we can employ the standard BP algorithm or the decoding algorithms for punctured LDPC codes which can boost the convergence speed and enhance the BER performance [15] for the decoding of LDPC-CC with RPCM. For the case that  $\mathbf{H}(D)$  contains more rational entries, the graph transformation can also be completed by running Procedure 1 with respect to all rational entries one by one as long as every rational entry is moved to the  $(1, c)$  position with proper permutation of rows and columns. Below, two examples are presented to verify the advantages of our proposed method.

*Example 1:* Consider an LDPC-CC with the following RPCM:

$$\begin{pmatrix} D^{400} & D^{181} & D^{144} & D^{82} & \frac{D^{35}}{1+D^{138}} \\ D & D^{252} & D^{354} & D^{377} & D^{279} \\ D^{20} & D^{409} & D^{344} & D^{383} & D^{107} \end{pmatrix}. \quad (16)$$

By Procedure 1, we have

$$\mathbf{H}_{\text{eq}}(D) = \begin{pmatrix} D^{400} & D^{181} & D^{144} & D^{82} & 0 & 1 & 0 \\ D & D^{252} & D^{354} & D^{377} & D^{279} & 0 & 0 \\ D^{20} & D^{409} & D^{344} & D^{383} & D^{107} & 0 & 0 \\ 0 & 0 & 0 & 0 & D^{35} & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & D^{138} & 1 \end{pmatrix}. \quad (17)$$

In this case, the girths of the Tanner graphs corresponding to (16) and (17) are 4 and 10, respectively. Also, the memory order is decreased from  $\infty$  to 409 by our method. Observed from the simulation results in Fig. 5, the decoding performance of the conventional method based on (16) is worse than the uncoded system, but our method can provide better performance which achieves BER  $10^{-5}$  at signal-to-noise ratio (SNR) 1.9dB.

*Example 2:* Consider an LDPC-CC with the following RPCM:

$$\begin{pmatrix} 1+D^{194} & D^{158} & D^{166} & D^{144} & 0 & D^{65} \\ D^{97} & D^{49} & \frac{1}{1+D^{20}+D^{76}} & D^{203} & D^{65} & D^{37} \\ 0 & D^{166} & D^{83} & D^{138} & D^{48} + D^{132} & 1 \end{pmatrix}. \quad (18)$$

By Procedure 1,  $\mathbf{H}_{\text{eq}}(D)$  can be constructed as

$$\begin{pmatrix} 1+D^{194} & D^{158} & D^{166} & D^{144} & 0 & D^{65} & 0 & 0 & 0 \\ D^{97} & D^{49} & 0 & D^{203} & D^{65} & D^{37} & 1 & 0 & 0 \\ 0 & D^{166} & D^{83} & D^{138} & D^{48} + D^{132} & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & D^{20} & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & D^{76} & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (19)$$

With our graph transformation, the girth is increased from 4 to 8. In addition, the memory order is decreased from  $\infty$  to 203. As shown in Fig. 6, our method can still outperform the original one with a significant SNR gain.

The principle of the graph transformation mentioned above has also been successfully applied to construct good LDPC-CC with RPCM. The simulation results show that the LDPC-CC with RPCM can acquire better BER performance than some well-constructed LDPC-CC with PPCM. Moreover, we have extended the proposed idea to the QC-LDPC whose parity-check matrices containing trinomial or higher order terms to achieve a remarkable performance improvement.

#### IV. CONCLUSION

Among rich studies in the literature, the LDPC-CC with RPCM are not considered because this class of LDPC-CC suffers several drawbacks from their viewpoints. In this paper, we propose a general procedure to transform the Tanner graph obtained by the direct mapping from RPCM into an equivalent graph, which can avoid the infinite memory order and undesired short length cycles. From the simulation results, the decoding performance based on the equivalent graph shows a significant improvement.

#### REFERENCES

- [1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21-28, Jan. 1962.
- [2] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, IT-27, pp. 533-547, Sept. 1981.
- [3] D. J. C. Mackay and R. Neal, "Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 32, pp. 1645-1646, Aug. 1996.
- [4] D. J. C. Mackay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399-431, Mar. 1999.
- [5] J. Pearl, "Probabilistic reasoning in intelligent systems: networks of plausible inference, San Francisco, CA: Morgan Kaufmann, 1988.
- [6] K. Engdahl and K. Sh. Zigangirov, "To the theory of low-density convolutional codes I," *Probl. Inf. Trans.*, vol. 35, no. 4, pp. 295-310, Dec. 1999.
- [7] A. J. Felström and K. Sh. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no.6, pp. 2181-2191, Sep. 1999.

- [8] R. Smarandache and P. O. Vontobel, "On regular quasi-cyclic LDPC codes from binomials," in *Proc. IEEE Int. Symp. Inform. Theory*, Chicago, IL, USA, Jun. 27 - Jul. 2 2004, pp. 274.
- [9] L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near-Shannon-limit quasi-cyclic low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1038-1042, Jul. 2004.
- [10] H. Tang, J. Xu, Y. Kou, S. Lin, and K. Abdel-Ghaffar, "On algebraic construction of Gallager and circulant low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 50, pp. 1269-1279, Jun. 2004.
- [11] Marc P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, NO. 8, pp. 1788-1793, Aug. 2004.
- [12] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966-2984, Dec. 2004.
- [13] A. Sridharan, "Design and analysis of LDPC convolutional codes," PhD Dissertation, University of Notre Dame, Notre Dame, Indiana, 2005.
- [14] C. J. Wu, Y. C. Chou, C. H. Wang, and C. C. Chao, "New construction of LDPC convolutional codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Toronto, Canada, Jul. 6-11 2008, pp. 1040-1044.
- [15] J. Kwon, D. Klinc, J. Ha, and S. W. McLaughlin, "Fast decoding of rate-compatible punctured LDPC codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Nice, France, Jul. 24-29 2007, pp. 1040-1044.

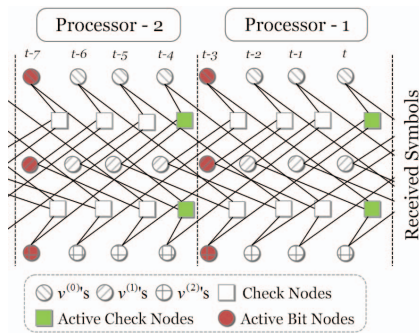


Fig. 1. The pipeline decoder on the Tanner graph.

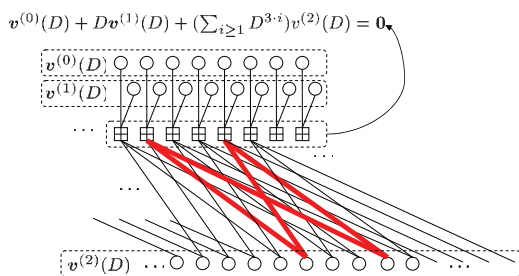


Fig. 2. The Tanner graph corresponding to (8), where the cycle in red has length 4.

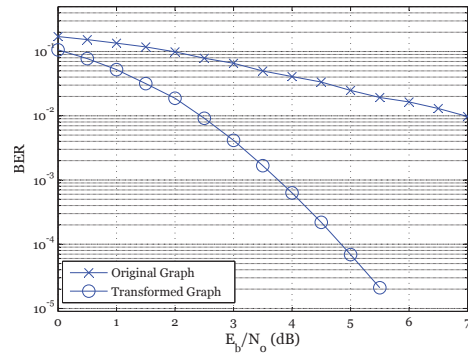


Fig. 3. Performance plots of decoding on various graphs in AWGN channels, where we use the BPSK transmission with block length  $10^3$  and employ the pipeline decoder with the number of iterations 50.

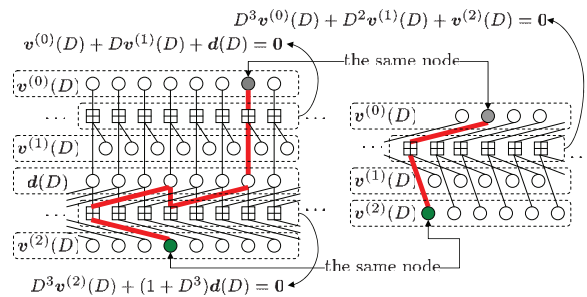


Fig. 4. The equivalent Tanner graph, where the cycle in red has length 8.

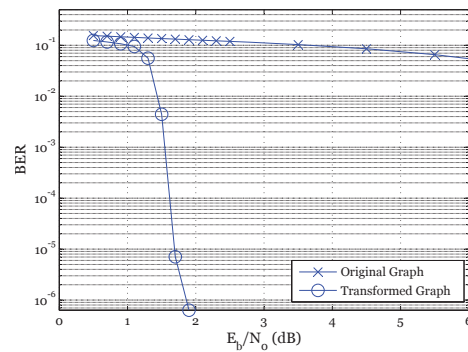


Fig. 5. Performance plots of decoding on various graphs in AWGN channels, where we use the BPSK transmission with block length  $10^4$  and employ the pipeline decoder with the number of iterations 50.

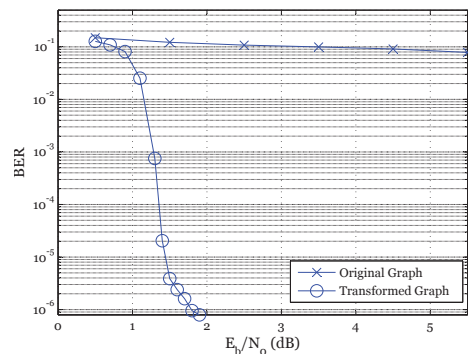


Fig. 6. Performance plots of decoding on various graphs in AWGN channels, where we use the BPSK transmission with block length  $10^6$  and employ the pipeline decoder with the number of iterations 100.