Interleaver Design of Punctured RA-Type LDPC Codes

Yong-Chun Piao*, Dong-Joon Shin* ***Regular Members***

ABSTRACT

In this paper, we analyze the cycle structure of punctured finite-length RA-type LDPC codes and design interleavers which have good memory efficiency and avoid short cycles. Furthermore, we introduce the check-node merging scheme for punctured finite-length RA-type LDPC codes and design simple interleavers. Simulation results show that punctured finite-length RA-type LDPC codes using the proposed simple interleavers have better performance than those with random and S-random interleavers.

Key Words : RA-type LDPC codes, Cycle, Girth, Interleaver, Puncturing

Ⅰ. Introduction

LDPC codes have been the main research topic because they show the capacity approaching performance with feasible decoding complexity. To achieve fast encoding, many RA-type LDPC codes have been proposed such as RA, IRA, concatenated zigzag (CZZ) and irregular CZZ (ICZZ) codes[1]-[4].

The construction methods of LDPC codes can be divided into two main classes: random (or pseudorandom) constructions and algebraic constructions. The S-random construction is a pseudorandom construction with the restriction that any two input positions within distance S cannot be permuted to two output positions within distance S. Generally, we choose $S < \sqrt{N/2}$ where N is the interleaver size[5]. Note that it becomes pure random interleaver if $S=1$. Random and S-random interleavers have to be found by computer search and stored in large memory, which implies more complex implementatin. Another problem with S-random interleavers is that their existence is not always guaranteed. To avoid these problems, researchers have considered deterministic constructions that can generate codes on the fly, such that the codes can be analyzed a priori and perform as well as random interleavers. Block interleavers[6] and linear interleavers[7] are well-known deterministic interleavers. For some component codes and short frame size, block and linear interleavers, with properly chosen parameters, perform better than random interleavers[8].

Puncturing is the widely used scheme to achieve various code rate for LDPC codes, which has many applications such as hybrid automatic repeat request (HARQ), adaptive modulation and coding (AMC), etc. However, punctured LDPC codes show worse performance degradation as the puncturing ratio becomes higher. Recently, a systematic puncturing scheme for LDPC codes with finite block lengths was proposed[9]. Short cycles in Tanner graphs prevent the sum-product decoding algorithm from converging to the optimal decoding result and produce error floor. Therefore, in this paper we analyze the cycle structures of punctured RA-type LDPC codes and design simple interleavers which can avoid short cycles to improve the performance of punctured codes.

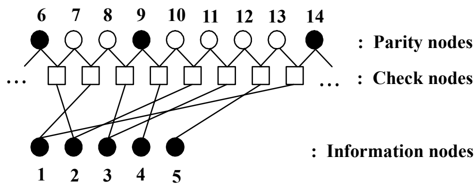
The outline of this paper is as follows. In Section II, we will introduce the check-node merged Tanner graph and analyze the short cycle structure of punctured RA-type LDPC codes. In Section III, we design simple interleavers of punctured finite-length RA-type LDPC codes to avoid short cycles. Simulation results are presented in Section IV and the conclusion is given in Section V.

※ 본 연구는 정보통신부 ITRC와 삼성전자 SKYPASS4G 과제 지원으로 수행되었습니다.
* 한양대학교 전자통신컴퓨터공학과 (pyc0928@ccrl.hanyang.ac.kr, djshin@hanyang.ac.kr)
논문번호 : KICS2006-09-389, 접수일자 : 2006년 9월 22일, 최종논문접수일자 : 2006년 12월 5일

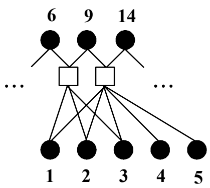
Ⅱ. Cycle Analysis of Punctured RA-type LDPC Codes

Tanner graph of punctured RA-type LDPC code can be simplified by removing punctured parity nodes and merging check nodes, that is called the check-node merged Tanner graph[10]. For the RA-type LDPC codes, the check nodes connected to the punctured degree-2 parity nodes can be merged as in Fig. 1. Thus, it is possible for the punctured RA-type LDPC codes to be decoded using the sum-product algorithm on the check-node merged Tanner graph as in Fig. 1 (b) instead of Fig. 1 (a).

We denote a puncturing pattern by $(d_1, d_2, \cdots, d_k)$ where d_i is the distance between the ith and the $(i+1)$ st unpunctured parity nodes. Then, the puncturing pattern in Fig.1 (a) becomes $(3,5)$. In this paper, for simplicity, we will consider only the regular puncturing pattern (d). Note that if the check node degrees are almost the same, this regular puncturing gives the best performance. Next, we will analyze the cycle structures of punctured RA, IRA, CZZ and ICZZ codes.



(a) Tanner graph of punctured code.



(b) Check-node merged Tanner graph.

Fig. 1. Two equivalent Tanner graphs of punctured RA-type LDPC code. (White circle denotes the punctured node.)

2.1 Cycle Structure of Punctured RA and IRA Codes

An RA code of L information bits and qL codeword bits is denoted by (qL, L)-RA code[1]. Fig. 2 shows the check-node merged Tanner graph of a (qL, L)-RA code punctured by (d) where π denotes the interleaver, i.e., permutation from Z_{qL} to Z_{qL} with Z_{qL} being the integers modulo qL.

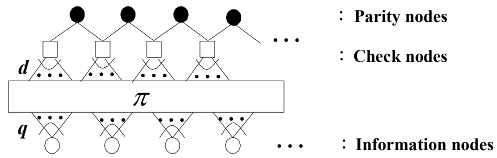


Fig. 2. Check-node merged Tanner graph of (qL, L)-RA code punctured by (d).

In IRA codes, the degrees of information nodes are not constant q but the cycle structure of IRA code can be similarly analyzed as RA code.

Next we will analyze the cycles of RA codes up to length 6. From now on, to calculate the absolute value of $\pi(x)-\pi(y)$, the values of $\pi(x)$ and $\pi(y)$ are regarded as the ordinary integer numbers, even though $\pi(x)$ and $\pi(y)$ are calculated using modulo qL arithmetic.

2.1.1 Cycles of length 2

The necessary condition for the existence of a cycle of length 2 is as follows.

$$|\pi(x)-\pi(y)| < d \quad (1)$$

where $nq \le x \ne y < (n+1)q$ for some $n \in \{0, 1, \cdots, L-1\}$.

2.1.2 Cycles of length 4

There are two types of cycles of length 4 and the necessary conditions for the existence of a cycle of length 4 are as follows.

$$\text{Type-I: } |\pi(x)-\pi(y)| < d \quad \text{and} \quad |\pi(x')-\pi(y')| < d,$$
$$\text{Type-II: } |\pi(x)-\pi(x')| < 2d \quad (2)$$

where $nq \le x \ne x' < (n+1)q$ and $mq \le y \ne y' < (m+1)q$ for some $n \ne m \in \{0, 1, \cdots, L-1\}$.

2.1.3 Cycles of length 6

There are three types of cycles of length 6 and the necessary conditions for the existence of a cycle of length 6 are as follows.

Type-I: $|\pi(x)-\pi(y)|<d$, $|\pi(y)-\pi(z)|<d$, and $|\pi(z)-\pi(x)|<d$,
Type-II: $|\pi(x)-\pi(y)|<d$ and $|\pi(x')-\pi(y')|<2d$,
Type-III: $|\pi(x)-\pi(x')|<3d$ (3)

where $nq \le x \ne x' < (n+1)q$, $mq \le y \ne y' < (m+1)q$ and $kq \le z < (k+1)q$ for some $n \ne m \ne k \in \{0,1,\cdots,L+1\}$.

2.2 Cycle Structure of Punctured CZZ and ICZZ Codes

Fig. 3 shows the check-node merged Tanner graph of $(N_c, l+2)$-CZZ code punctured by (d) where N_c is the number of component zigzag codes and $l+2$ is the check node degree. π_i denotes the interleaver for the ith zigzag code, $1 \le i \le N_c$, i.e., permutation from Z_L to Z_L where L is the number of information bits. ICZZ codes[4] are different from CZZ codes in the sense that the number of information bits entering into each zigzag encoder can be different and also different zigzag codes can be used as component codes. However, cycle structures of ICZZ codes can be similarly analyzed as CZZ codes. Next we will analyze the cycles of CZZ codes up to length 6.

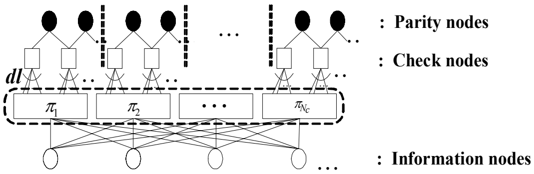


Fig. 3. Check-node merged Tanner graph of $(N_c, l+2)$-CZZ code punctured by (d)

In CZZ codes, since there are no parallel edges between an information node and a check node, punctured CZZ codes do not have a cycle of length 2.

2.2.1 Cycles of length 4

The necessary condition for the existence of a cycle of length 4 is as follows.

$$|\pi_i(x)-\pi_i(y)|<d \text{ and } |\pi_j(x)-\pi_j(y)|<d \quad (4)$$

for $i \ne j$ and some $x \ne y \in \{0,1,\cdots,L-1\}$.

2.2.2 Cycles of length 6

There are two types of cycles of length 6 and the necessary conditions for the existence of a cycle of length 6 are as follows.

Type I: $|\pi_i(x)-\pi_i(y)|<d$, $|\pi_j(y)-\pi_j(z)|<d$, and $|\pi_k(x)-\pi_k(z)|<d$,
Type II: $|\pi_i(x)-\pi_i(y)|<2d$ and $|\pi_j(x)-\pi_j(y)|<d$ (5)

for $i \ne j \ne k$ and some $x \ne y \ne z \in \{0,1,\cdots,L-1\}$.

Ⅲ. Simple Interleaver Design

The interleaver plays a fundamental role for the performance of LDPC codes. In this section, by using the permutation polynomials such as those given in [8] and [11] we design simple polynomial interleavers for punctured finite-length RA-type LDPC codes such that short cycles can be avoided.

***Theorem* 1 [8]**:
(a) $\pi(x)$ is a permutation polynomial of degree k over the integer ring Z_{p^n} if and only if $\pi(x)$ is a permutation polynomial over Z_p and $\pi'(x) \ne 0 \bmod p$ for all integers $x \in Z_{p^n}$.
(b) For any $N=\prod_{i=1}^{m} p_i^{n_i}$ where p_i's are distinct prime numbers, $\pi(x)$ is a permutation polynomial modulo N if and only if $\pi(x)$ is also a permutation polynomial modulo $p_i^{n_i}$ for all i.

***Theorem* 2 [11]**: If $d>1$ is a divisor of $k-1$, then there exists no permutation polynomial $\pi(x)$ of degree d.

Let $N=qL$ where L is the number of information bits and q is the repetition number.

3.1 Interleavers for Punctured RA and IRA Codes

Let $N=qL$ where $L=p^n$ and p is any prime number. Then we can design polynomial interleavers of length N for $(q \times p^n, p^n)$-RA codes by using

Theorem 1. The following Fact 1 is a special case of this theorem when $p=2$ and $q=2^m$.

Fact 1: For the $(2^m \times 2^n, 2^n)$-RA code, $\pi(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0 \mod (2^m \times 2^n)$ can be an interleaver if $a_k \neq 0$, a_1 is odd, $a_2 + a_4 + \cdots$ is even, and $a_3 + a_5 + \cdots$ is even.

Interleaver design for RA codes using Fact 1.

(a) If there is no solution of (1), the interleaver π can avoid the cycles of length 2.

(b) If there is no common solution of (2), the interleaver π can avoid the Type-I and Type-II cycles of length 4.

(c) If there is no common solution of (3), the interleaver π can avoid the Type-I, Type-II and Type-III cycles of length 6.

To remove cycles of length 2, let $\Lambda = \{|\pi(i) - \pi(j)| | x \leq i \neq j \leq x+q-1\}$, where $x \in 0, q, \cdots, (l-1)q$, and design π such that there is no solution of (1). Namely, $\Lambda \cap \{1, 2, \cdots, d-1\} = \phi$.

To remove cycles of lengths 2, 4 and 6, the interleavers of punctured RA and IRA codes can be designed similarly to the above method.

3.2 Interleavers for Punctured CZZ and ICZZ Codes

Fact 2: $\pi_i(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0 \mod L$ where $1 \leq i \leq N_c$ can be interleavers for CZZ codes if $\pi_i(x)$ is a permutation polynomial modulo L for all i.

Interleaver design for CZZ codes using Fact 2.

(a) If there is no common solution of (4) for all $i \neq j$, then $\pi_1, \cdots, \pi_{N_c}$ can avoid the cycles of length 4.

(b) If there is no common solution of (5) for all $i \neq j \neq k$, then $\pi_1, \cdots, \pi_{N_c}$ can avoid the Type-I and Type-II cycles of length 6.

To remove cycles of length 4, first define two sets $\Lambda_i = \{|\pi_i(x) - \pi_i(y)| | 0 \leq x \neq y < L\}$, $\Lambda_j = \{|\pi_j(x) - \pi_j(y)| | 0 \leq x \neq y < L\}$.

Design π_i and π_j such that there is no solution satisfy (4). Namely, Λ_i and Λ_j are disjoint for any pair (x, y).

To remove cycles of lengths 4 and 6, the interleavers of punctured CZZ and ICZZ codes can be designed similarly to the above method.

Ⅳ. Simulation Results

As mother codes, systematic $(4L, L)$-RA and $(4, 3)$-CZZ codes of rate $1/5$ are used and they are punctured by the puncturing patterns (2), (4), and (8) to achieve the code rates $1/3$, $1/2$ and $2/3$. All degree-2 polynomial interleavers in Table I achieve the girth 10 for mother codes of rate $1/5$, but for punctured codes of rates $1/3$, $1/2$ and $2/3$, they can have cycles of lengths 2, 4, 6 and 8. All degree-1 polynomial interleavers in Table II achieve the girth 10 for mother codes of rate $1/5$, but for punctured codes of rates $1/3$, $1/2$ and $2/3$, they can have cycles of lengths 4, 6 and 8. For the simulation, BPSK modulation, sum-product algorithm, maximum 30 iterations, and AWGN channel are assumed. In Fig. 4, RA code using $R1$ polynomial interleaver in Table I shows better performance than the other polynomial, random, and S-random interleavers. In Fig. 5, CZZ code using $C1$ polynomial interleaver in Table II shows better performance than the other polynomial, random and S-random interleavers.

Table 1. Girth Of Punctured Systematic $(4L, L)$-Ra Codes With L=1024 For Degree-2 Interleavers

	1/5	1/3	1/2	2/3
$R1$: $32x^2+15x+1$	10	8	8	8
$R2$: $40x^2+9x+1$	10	6	6	4
$R3$: $128x^2+63x+1$	10	6	4	2

Table 2. Girth Of Punctured $(4,3)$-Czz Codes With L=1000 For Degree-1 Interleavers

	1/5	1/3	1/2	2/3
$C1$: $3x+1$, $11x+1$, $39x+1$, $141x+1$	10	8	8	8
$C2$: $3x+1$, $11x+1$, $79x+1$, $109x+1$	10	8	6	6
$C3$: $x+1$, $3x+1$, $17x+1$, $19x+1$	10	6	4	4

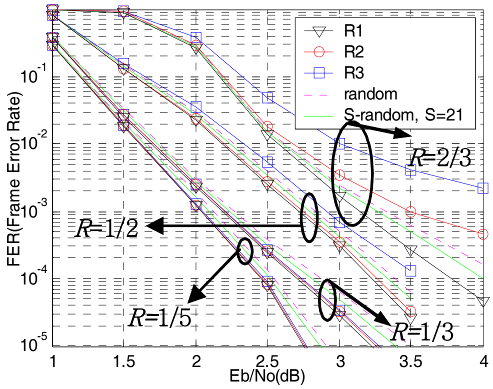


Fig. 4. Performance of punctured systematic (4096,1024)-RA codes using various interleavers.

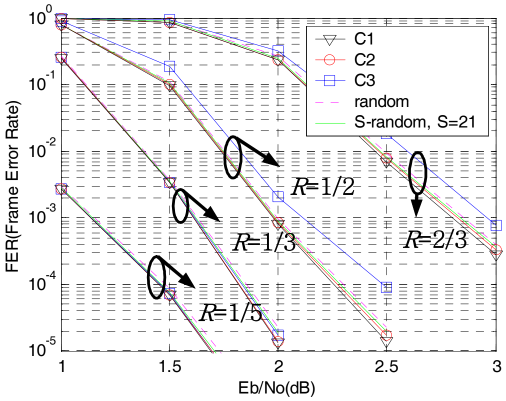


Fig. 5. Performance of punctured (4,3)-CZZ codes with *L*=1000 using various interleavers.

V. Conclusion

In this paper, we analyze the cycle structure of various punctured finite-length RA-type LDPC codes and design simple polynomial interleavers which have good memory efficiency and avoid short cycles. Simulation results show that well-designed simple polynomial interleavers have better performance than the random and S-random interleavers. As a future work, it is interesting to find permutation polynomials for the various code lengths, which are suitable as the interleavers of RA-type LDPC codes.

References

〔1〕 D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'Turbo-like' codes," in *Proc. 36th Allerton Conf. on Communitation, Control and Computing*, pp. 201-210, 1998.

〔2〕 H. Jin, A. Khandekar, and R. J. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. on Turbo Codes and Related Topics*, pp. 1-8, Brest, France, Sep. 2000.

〔3〕 L. Ping, X. L. Huang, and N. Phamdo, "Zigzag codes and concatenated zigzag codes," *IEEE Trans. Inform. Theory,* vol. 47, pp. 800-807, Feb. 2001.

〔4〕 S. N. Hong and D. J. Shin, "Design of irregular concatenated zigzag codes," in *Proc. IEEE int. Symp. Inform. Theory (ISIT'2005)*, pp. 1363-1366, Sep. 2005.

〔5〕 D. Divsalar and F. Pollara, "Turbo codes for PCS applications," *IEEE ICC'95*, pp. 54-59, Jun. 1995.

〔6〕 L. Perez, J. Seghers, and D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1698-1709, Nov. 1996.

〔7〕 O. Y. Takeshita and D. J. Costello, "New deterministic interleaver designs for turbo codes," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 1988-2006, Sep. 2000.

〔8〕 J. Sun and O. Y. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 101-119, Jan. 2005.

〔9〕 J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE Trans. Inform. Theory*, pp. 728-738, Feb. 2006.

〔10〕 S. N. Hong, H. G. Joo, and D. J. Shin, "Optimal puncturing of block-type LDPC codes and their fast convergence decoding," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT'2006)*.

〔11〕 W. S. Diestelkamp, S. G. Hartke and R. H. Kenney, "On the degree of local permutation polynomials," *Journal of Combinatorial Mathematics and Combinatorial Computing*, Vol. 50, pp. 129-140, 2004.

박 영 춘 (Yong-Chun Piao) 정회원

2001년 8월 중국 동북 사범대학교 수학과 학사

2004년 8월 한양대학교 전자통신컴퓨터공학 석사

2004년 9월~현재 한양대학교 전자통신컴퓨터공학 박사과정

<관심분야> 디지털통신, 오류정정부호

신 동 준 (Dong-Joon Shin) 정회원

1990년 2월 서울대학교 전자공학 공학사

1991년12월 Northwestern University, 전기공학 석사

1998년 12월 University of Southern California, 전기공학과 공학박사

1999년 1월~1999년 4월 Research Associate (USC)

1999년 4월~2000년 8월 Hughes Network Systems, MTS

2000년 9월~현재 한양대학교 전자통신컴퓨터공학부 부교수

<관심분야> 디지털통신, 오류정정부호, 이산수학, 시퀀스