

병렬 어레이 프로세싱을 위한 반집합 대수 LDPC 부호의 구성

정회원 이 광 재*, 이 문 호**, 학생회원 이 동 민***

Construction of Semi-Algebra Low Density Parity Check Codes for Parallel Array Processing

Kwang-jae Lee*, Moon-ho Lee** *Regular Members* Dong-min Lee*** *Student Member*

요 약

본 논문에서는 이중 대각 부행렬에 근거한 결정적 LDPC 부호의 일종인 반집합 대수 LDPC 행렬이라는 새로운 LDPC 부호의 구성을 제시한다. 이러한 구성 방법에 의해 일종의 high rate LDPC 부호를 얻게 되며, 이 부류의 부호들은 큰 girth와 양호한 최소거리 특성을 갖는다. 또한, 순환 시프트 레지스터를 이용한 단순한 병렬 어레이 구조로 구현할 수 있으며, 반복 복호에 의해 용이하게 처리될 수 있다.

Key Words . LDPC, semi-algebra, cyclic-shift, parallel array; dual-diagonal

ABSTRACT

In this paper, we present a novel LDPC code construction called as semi-algebra low density parity check(LDPC) codes which is one kind of deterministic LDPC code based on dual-diagonal sub-matrix. The constructing method results in a class of high rate LDPC codes. Codes in this class have a large girth and good minimum distances. Furthermore, they can be implemented by simple parallel array architecture using cyclic shift register and perform well with the iterative decoding.

I. 서 론

본 논문에서는 이중 대각 부행렬에 근거한 결정적(deterministic) LDPC 부호의 일종인 반집합 대수(semi-algebra) LDPC(low density parity check) 행렬이라는 새로운 LDPC 부호의 구성을 제시한다. 이러한 패턴은 Li Ping, WK Leung 및 Nam Phamdo에 의해 소개되었으며[1], Rich Echard 및 SC Chang 등은 π -rotation 구조를 이용하여 하드웨어 구현이 용이한 π -rotation LDPC 부호를 제시하였다[2]. 또한 Darush Divsalar, Huyi Jin 및 Robert J McEliece에

의해 소개된 "RA-turbo code"의 일반형으로 고려할 수도 있다[3]. 이러한 구조는 단순한 부호화 기법에 적합하며, 부호 거리 및 행렬 girth의 추정에 대한 이론적 기초해석을 수행하기 위한 충분한 대칭성(symmetry)을 포함하고 있다. 이하 본 논문에서는 반집합 대수 LDPC 부호의 기본적 구성과 특성에 대해 고찰하며, 이의 성능 및 하드웨어 구조를 제시한다. 시뮬레이션 결과에서 반집합 대수 LDPC 부호는 랜덤 부호[4] 및 quasi-cyclic LDPC 부호[5]와 비교할 때 high rate LDPC 부호로서 우수한 특성을 보여주고 있으며, 패리티 검사 행렬의 각 부행렬은 단지 순

* 한려대학교 멀티미디어 정보통신공학과 (kjee@haniyo.ac.kr),

** 전북대학교 전기전자공학부 (dmlee74@hotmail.com), ** 전북대학교 전기전자공학부 (moonho@chonbuk.ac.kr)
논문번호 030459-1014, 접수일자 2003년 10월 13일

※ This work was supported in part by Univ IT Research Center Project, Ministry of Information&Comm, and Ministry of Commerce, Industry &Energy, Korea

환 시프트된 항등행렬로 구성되므로 정보벡터와 부행렬간의 연산, 즉 부호화 과정은 복잡한 행렬연산 대신 단순한 시프트만으로 처리할 수 있어 간단한 하드웨어를 이용하여 부호화기를 구현할 수 있는 이점이 있다. 또한, 각 부행렬 단위로 독립 연산이 가능하고, 각 부행렬에 대한 연산구조가 동일하기 때문에 고속 병렬처리 구조로의 확장이 용이할 뿐만 아니라, 순환 시프트의 특성상 왼쪽 시프트와 오른쪽 시프트를 적절히 활용하면 최대 시프트 수를 절반으로 줄일 수 있으므로, 이러한 특성을 이용하여 연산 속도를 두 배로 높일 수 있는 이점이 있다.

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

II. 반집합 대수 LDPC 부호의 설계

H 행렬은 H^p 와 H^d 두 개의 부행렬로 구성되며, 각 부행렬은 부벡터 c^p 및 부벡터 c^d 와 관련된다. 이하에 서는 c^p 를 패리티 벡터, c^d 는 정보 벡터라 부르기로 한다

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = [H^p H^d] \quad (1)$$

$$c = \begin{bmatrix} c^p \\ c^d \end{bmatrix} \quad (2)$$

H^p 는 $(n-k) \times (n-k)$ 정방행렬이며, H^d 는 $(n-k) \times k$ 행렬이다 패리티 검사는 $H \cdot c = 0$ 의 제약조건을 만족하여야 하므로 다음의 관계를 얻을 수 있다.

$$H^p c^p = H^d c^d \quad (3)$$

2.1 H^p 행렬

H^p 부행렬은 소위 이중 대각(dual-diagonal) 행렬이며, 다음 식으로 나타낼 수 있다. 여기서 I는 대각 행렬을, D는 대각 행렬의 첫 행을 제거하고 마지막 행에 0으로 이루어진 행을 추가한 것이다

$$H^p = I + D \quad (4)$$

예를 들어, 10×10 의 간단한 D 행렬의 경우 다음과 같다

$$H^p = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

주어진 임의의 정보 벡터에 대한 부호어 벡터를 구하기 위해 식 (2), (3)의 관계로부터 다음과 같은 프로젝션(projection) 벡터를 정의한다

$$H^p c^p = v = H^d c^d \quad (5)$$

$[H^p]^{-1} = U^p$ 의 관계에 주목하면,

$$c^p = U^p v \quad (6)$$

여기서, U^p 는 상삼각(upper triangular) 행렬이다. 따라서, 프로젝션 벡터를 먼저 계산한 후, 식 (6)에 따라 이를 변환하여 c^p 를 구할 수 있으며, 식 (2)의 관계로부터 최종적인 부호어를 얻게 된다.

2.2 H^d 행렬

본 절에서는 대수적 접근에 근거한 H^d 행렬의 구성법에 대하여 고찰한다 H^d 행렬은 $t \times t$ 부행렬들로

분할되며, 여기서 t 는 정수이다. $t \times t$ 길이의 (j, k) - H^d 행렬(단, j 는 column weight, k 는 row weight)의 구성은 다음과 같은 과정으로 나타낼 수 있다.

- 1 패리티 검사 행렬의 j 번째 행 블록, k 번째 열 블록에 위치하는 부행렬은 $I_{t \times t}$ 대각행렬에서 $(\text{mod } t)$ 의 관계를 이용하여 각 행을 오른쪽 시프트 시켜 얻게 되며, 이를 $B_{j,k}$ 라 하자

$$\text{단, } i \in S = \{0, 1, 2, \dots, t-1\} \quad (7)$$

2. $q^k \equiv 1 \pmod{t}$ 를 만족하는 q 가 존재하면, 집합 S 는 다수의 집합 C 로 분할할 수 있으며, s 를 포함 한 하나의 집합은 다음과 같다.

$$\{s, sq, sq^2, \dots, sq^{m_s-1}\} \quad (8)$$

여기서, m_s 는 다음을 만족하는 가장 작은 양의 정수이다.

$$sq^{m_s} \equiv s \pmod{t} \quad (9)$$

3. H^d 행렬에서 1의 위치는 집합 C_1, \dots, C_t 와 파라미터 t 를 이용하여 결정할 수 있다

본 논문에서는 파라미터 i, j, k 로부터 $B_{j,k}$ 를 구성하기 위해 다음과 같은 반복(recursive) 알고리즘을 제안한다.

$B_{j,k}$ (Recursive algorithm)

$p=i$ (shift number)

for $j_1 = 1$ to t (t is block size of)

for $j_2 = 1$ to t

$B[j_1][j_2] = 0$ (initialization)

$j_2 = (j_1+p) \text{ mod } t$

$B[j_1][j_2] = 1$ (shifted procession)

$B_{j,k}$ 에 대한 하드웨어 구성도를 그림 1에 나타내었으며, i (shift number)를 변경시킴으로써 H^d 행렬의 구성을 위한 또 다른 시프트 행렬을 얻을 수 있다. 그림 1의 시프트 위치를 위한 회로도에서 j_1 과 p 의 가산 결과와 t 를 비교하여 (j_1+p) 가 t 에 비해 작은 경우 시프트 위치 n 은 곧바로 구해지지만, 반대의 경

우 $(j_1+p)-t$ 를 이용하여 시프트 위치 n 을 구하게 된다. 이로써 시프트 위치에 따라 $B_{j,k}$ 를 구성하는 회로를 구현할 수 있으며, 전체 $t \times t$ 행렬을 RAM에 저장하게 된다.

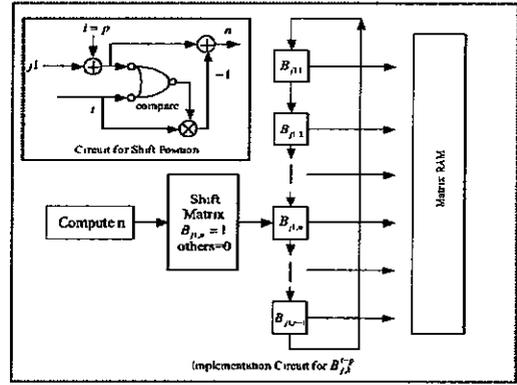


그림 1 시프트 위치를 위한 회로 및 $B_{j,k}$ 행렬을 위한 하드웨어 구성도

2.3 반집합 대수 LDPC

반집합 대수 LDPC의 일반식은 다음과 같이 나타낼 수 있다^[6].

$$H_{New} = \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & H(p_{0,0}) & H(p_{0,1}) & \dots & H(p_{0,k-1}) \\ 0 & \dots & 0 & 0 & H(p_{1,0}) & H(p_{1,1}) & \dots & H(p_{1,k-1}) \\ 0 & 0 & \dots & 1 & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 & H(p_{j-1,0}) & H(p_{j-1,1}) & \dots & H(p_{j-1,k-1}) \end{array} \right] \quad (10)$$

제안 행렬의 부호율(code rate)은 다음과 같으며(n, m 은 각각 H^d 행렬의 열 번호와 행 번호를 나타낸다), high rate LDPC 부호의 일종임을 알 수 있다.

$$R_{New} = \frac{n_{New} - m}{n_{New}} = \frac{(m+n) - m}{(m+n)} = \frac{1}{1 + \left(\frac{m}{n}\right)} = \begin{cases} 1/2, & n=m \\ (1/2, 1) & n > m \\ 1, & n \rightarrow \infty \end{cases} \quad (11)$$

예를 들어, $j=3, k=5, t=31$ 인 경우의 반집합 대수 LDPC 부호를 설계하는 경우, $q^k \equiv 1 \pmod{t}$ 에 따라 $q=2$ 가 되며, 패리티 검사 행렬은

$$H = [H_{93 \times 93}^b | H_{93 \times 135}^d]$$

$$R = \frac{1}{1 + m/n} = \frac{1}{1 + 93/155} = 0.625 \quad (12)$$

H^d 행렬 구성의 두 번째 단계에 따른 집합 C_i ($i=0,1,2,\dots,12$)의 분포를 표 1에 나타내었다 H^d 의 1의 위치는 C_1, C_2, C_3 에 의해 결정된다.

표 1 집합 C_i 의 분포

S	$sq^{m_i} \equiv s \pmod{t}$	m_i	$C = \{s, sq, sq^2, \dots, sq^{m_i-1}\}$
0	$0 \cdot 2^{m_i} = 0 \pmod{31}$	1	{0}
1	$1 \cdot 2^{m_i} = 1 \pmod{31}$	5	{1, 2, 4, 8, 16}
3	$3 \cdot 2^{m_i} = 3 \pmod{31}$	5	{3, 6, 12, 24, 17}
5	$5 \cdot 2^{m_i} = 5 \pmod{31}$	5	{5, 10, 20, 9, 18}
6	$6 \cdot 2^{m_i} = 6 \pmod{31}$	5	{6, 12, 24, 17, 3}
7	$7 \cdot 2^{m_i} = 7 \pmod{31}$	5	{7, 14, 28, 25, 19}
9	$9 \cdot 2^{m_i} = 9 \pmod{31}$	5	{9, 18, 5, 10, 20}
10	$10 \cdot 2^{m_i} = 10 \pmod{31}$	5	{10, 20, 9, 18, 5}
11	$11 \cdot 2^{m_i} = 11 \pmod{31}$	5	{11, 22, 13, 26, 21}
12	$12 \cdot 2^{m_i} = 12 \pmod{31}$	5	{12, 24, 17, 3, 6}
13	$13 \cdot 2^{m_i} = 13 \pmod{31}$	5	{13, 26, 21, 11, 22}
14	$14 \cdot 2^{m_i} = 14 \pmod{31}$	5	{14, 28, 25, 19, 7}
15	$15 \cdot 2^{m_i} = 15 \pmod{31}$	5	{15, 30, 29, 27, 23}

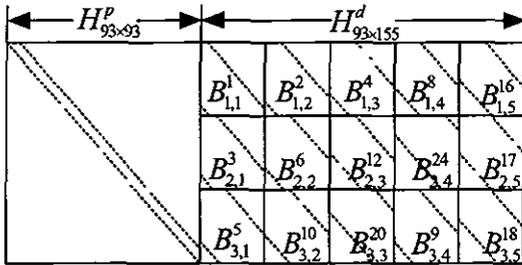
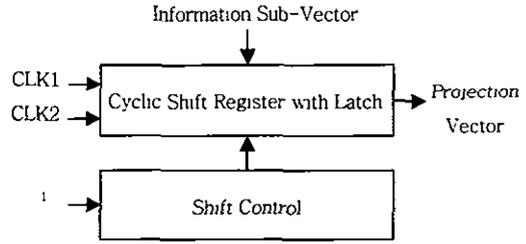


그림 2 $j=3, k=5, t=31$ 인 경우의 반집합 대수 LDPC 부호

반집합 대수 행렬을 구현하기 위해 {1,2,4,8,16}, {3,6,12,24,17}, {5,10,20,9,18}을 선택하면, 그림 2와 같이 되며, 그림에서 점선으로 나타낸 부분은 1의 위치를, 그 외의 부분은 모두 0으로 구성되는 저밀도 행렬이다. 이하에서는 패리티 검사 행렬을 구성하고 있는 부행렬들의 열을 열 블록, 부행렬들의 행을 행 블록으로 부르기로 한다.



CLK1 clock for loading information sub-vector
CLK2 clock for shift right operation

그림 3 부행렬 연산을 위한 PE 구조

2.4 SA-LDPC 부호를 이용한 부호화기 구성

앞 절에서 시프트 레지스터를 이용하여 $t \times t$ 부행렬 $B_{j,k}$ 을 생성하는 하드웨어의 블록도를 나타내었다. 본 절에서는 부행렬 $B_{j,k}$ 을 생성하지 않고 $B_{j,k}$ 의 시프트 특성만을 이용하여 프로젝션 벡터를 생성하고 이를 부호어로 변환함으로써 행렬 연산 없이 부호화를 실현하는 반복 및 병렬처리 구조의 하드웨어 구성에 대해 고찰한다 제안 방식에서 모든 부행렬은 항등행렬 $I_{t,t}$ 의 모든 열을 오른쪽 시프트시켜 얻게 되며, 시프트된 항등 행렬과 정보 부벡터 간의 연산은 정보 부벡터를 순환 시프트(cyclic shift) 시킨 것과 동일하다. 즉, $I_{t,t}$ 에 대한 시프트 회수만큼 정보 부벡터를 단지 오른쪽(또는 왼쪽) 순환 시프트 시킴으로써 부행렬에 대한 연산이 완료됨을 의미한다 그림 3에 부행렬 $B_{j,k}$ 와 정보 부벡터 간의 연산을 위한 기본적인 처리요소인 PE (processing element)의 구조를 나타내었으며, pre-set 기능을 갖는 양방향 시프트 레지스터와 1 값에 따라 시프트 회수를 결정하는 제어부로 구성된다.

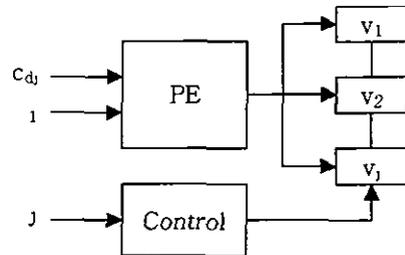


그림 4 반복 연산 구조

그림 4는 하나의 PE만을 이용하여 부호화를 수행

하는 반복 연산구조를 나타내며, 이 구조에서는 패리티 검사 행렬의 부행렬들의 열 단위로 반복 연산을 수행하기 때문에 프로젝션 벡터의 출력 방향을 결정하기 위한 추가의 제어부가 필요하다

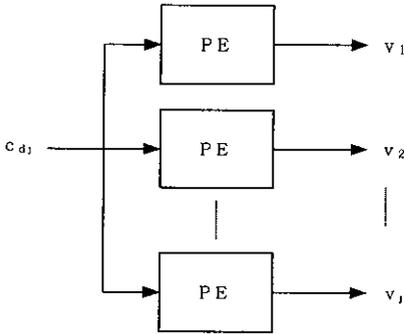


그림 5 선형 배열 구조에 의한 병렬처리

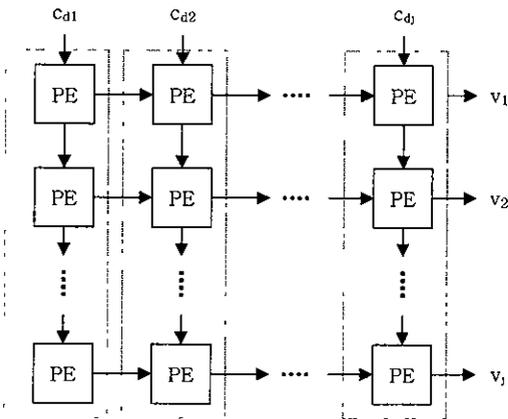


그림 6 병렬 배열 구조에 의한 병렬처리

그림 5는 선형 배열에 의한 병렬처리 구조를 나타내며, 임의 프로젝션 벡터는 다음과 같이 구성된다.

$$\begin{aligned}
 v_1 &= v_{1,1} + v_{1,2} + \dots + v_{1,k} \\
 v_2 &= v_{2,1} + v_{2,2} + \dots + v_{2,k} \\
 &\vdots \\
 v_j &= v_{j,1} + v_{j,2} + \dots + v_{j,k}
 \end{aligned}
 \tag{13}$$

패리티 검사 행렬의 열 블록 즉, 열을 구성하는 부행렬들에 대한 연산이 동시에 수행되며, 각 PE에 i값을 미리 저장시켜 두거나, 시프스 생성기 등을 이용함으로써 외부 제어신호를 필요로 하지 않는 효율적인 구

조를 얻을 수도 있다

그림 6은 선형 배열 구조를 확장시킨 병렬처리 구조이며, 각 PE는 패리티 검사 행렬의 특정 부행렬에 대한 연산을 특정 PE가 독립적으로 수행하기 때문에 i값을 제어신호로 요구하지 않는다. 이는 PLD, FPGA, ASIC 등에 의한 VLSI 구현시 매우 유용한 특성이다.

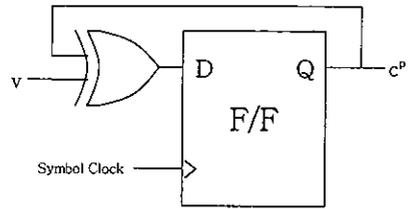


그림 7 프로젝션 벡터에서 부호어를 구하기 위한 회로

그림 7은 식 (6)에 따라 부호어를 구하기 위한 회로의 구성으로 문헌[2]에 제시되어 있으며, 플립플롭을 이용하여 상삼각 행렬에 대한 연산을 대신하고 있다.

III. 반집합 대수 LDPC 부호의 특성

본 절에서는 반집합 대수 행렬에 대한 해석적 결과를 기술한다. 특히, 제안 부호의 거리와 행렬의 girth 특성에 대해 고찰한다 LDPC 부호군에 대한 이 두 가지 특성은 우수한 반집합 대수 LDPC 부호를 선택하는데 있어 유용하다

3.1 거리 특성

하드웨어 구현을 고려하여 설계된 π rotation LDPC 부호^[2]와 제안 방식인 반집합 대수 LDPC 부호의 행렬이 갖는 특성을 비교할 때, 다음과 같은 차이점을 나타낸다 반집합 대수 LDPC 부호의 행렬은 임의의 두 행에 대한 열의 구성 요소로서 단지 하나의 "1"을 포함하며, 각 열의 무게(weight)는 j이며, 여기서 j는 정수(integer)이다 본 제안 부호의 최소거리를 구하기 위해서는 먼저 식 (1)의 패리티 검사 행렬 H와 생성행렬(generator matrix) G의 관계를 고찰할 필요가 있다 행렬 G의 각 행은 $[c^j \ c^d]$ 로 구성된다. 여기서 c^j 는 무게 1인 벡터를 나타내며, c^d 와 관련된 c^j 는 H^T 행렬에서 하나의 열을 식 (6)의 관계에 따라 변환한 것에 지나지 않는다. 따라서 H^T

행렬의 각 열을 변환하여 얻어진 행렬을 R이라 하면, G 행렬은 다음과 같이 쓸 수 있다.

$$G = [R | I] \tag{14}$$

일반적으로 선형 블록 부호의 최소거리는 G 행렬의 최소 행 무게로서, 또는 H 행렬에서 각 열의 mod 2 합이 0인 열의 수로서 구할 수 있지만, LDPC 부호의 경우 상당히 복잡한 문제가 될 수 있다. 그러나, H 행렬에 이중 대각 행렬을 포함하는 RA-부호, π -rotation LDPC 부호 및 본 제안 부호의 경우에는 H^d 행렬에 대한 거리 d_2 를 이용하여 부호어의 최소 거리를 결정할 수 있다[7] 여기서, 거리 d_2 는 무게 2인 정보벡터를 입력으로 가정할 때 패리티 벡터 c^T 의 최소 무게를 의미하며, 식 (14)로부터 최소거리는 $d_2 + 1$ 이 된다.

이러한 방식으로 구한 본 제안 부호의 거리는 다음과 같이 주어지며, 이의 계산 예를 이하에서 간략히 살펴보기로 한다.

$$d_{min} = j+1, (j>2) \tag{15}$$

$j=4, k=4, t=5$ 일 경우, H^d 행렬을 구성하기 위해, 다음과 같이 4개의 시프트 수를 선택할 수 있다.

$$\begin{aligned} &\{1, 3, 4, 2\} (s=1) \\ &\{2, 1, 3, 4\} (s=2) \\ &\{3, 4, 2, 1\} (s=3) \\ &\{4, 2, 1, 3\} (s=4) \end{aligned} \tag{16}$$

부행렬 $B_{11}(1)$ 은 다음과 같다.

$$B_{11}(1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

$B_{11}(1)$ 의 최하위 행부터 1의 위치를 이용하여 표시하면 다음과 같다

$$B_{11}(1) = [15432] \tag{18}$$

이를 이용하면 H^d 행렬은 다음과 같이 나타낼 수 있다

$$H^d = \begin{bmatrix} [15432] & [32154] & [43215] & [21543] \\ [21543] & [15432] & [32154] & [43215] \\ [32154] & [43215] & [21543] & [15432] \\ [43215] & [21543] & [15432] & [32154] \end{bmatrix} \tag{19}$$

예를 들어, 열 1과 13에 대한 최소거리 d_{mn} 은 다음과 같이 계산된다.

$$d_{mn} = |1-2| + |2-1| + |3-5| + |4-4| + 1 = 5$$

이는 H^d 행렬 내에서 최소 값이며, 따라서 $d_{mn} = 5$ 가 된다.

$$\begin{matrix} \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{matrix} \begin{matrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{matrix} \end{matrix} \rightarrow \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{matrix}$$

$$\begin{matrix} \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{matrix} \end{matrix} \leftarrow \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{matrix}$$

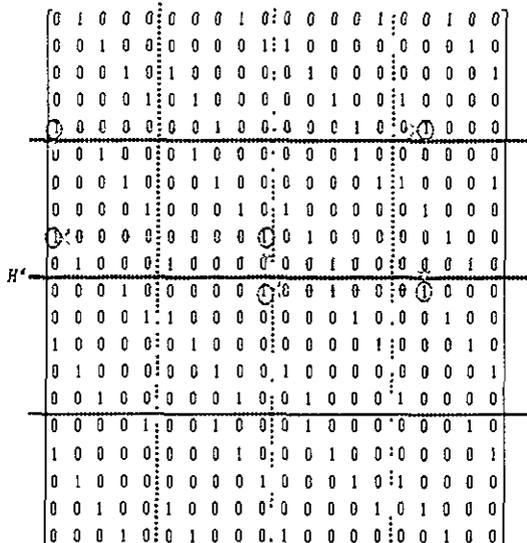


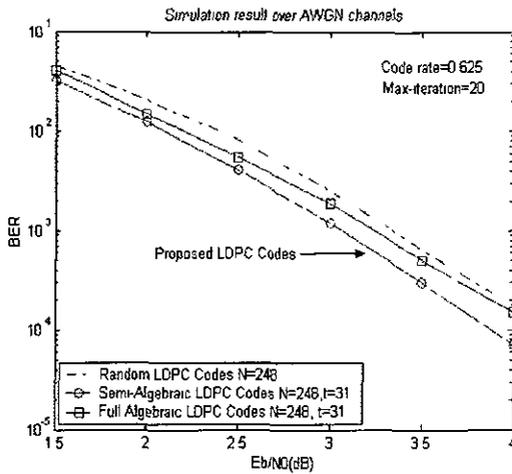
그림 8 SA-LDPC 부호의 girth 특성

3.2 Girth 특성

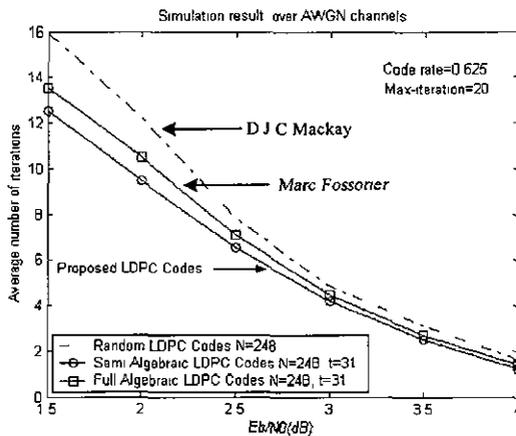
Girth 만큼의 최대 사이클을 호출할 수 있으므로, H^d 행렬의 girth를 최대화함으로써 제안 부호의 성능을 개선할 수 있다. 제안 부호의 H^d 행렬은 대수적 접근에 근거하고 있으며, 문헌[8]에 제시된 방법을 이용하여 구한 본 대수적 LDPC 부호의 girth는 6 이상이며, 그림 8에 이의 예를 나타내었다. 또한, girth 향상을 위한 다른 대수적 기법을 이용한다면 12까지 개선할 수 있을 것으로 기대된다^[7]. 이는 girth 6인 랜덤 부호와 girth 4인 π -rotation LDPC 부호에 비해 보다 개선된 특성이다^[2]

IV. 시뮬레이션 및 결론

Sum product 알고리즘^[3]에 근거한 제안 LDPC 부호의 시뮬레이션 결과를 그림 9에 나타내었으며, E_b/N_0 대 BER 특성, 그리고 평균 반복 회수 대 E_b/N_0 특성을 포함하고 있다. 평균 반복 회수는 부호 속도와 전력 소비를 나타내는 중요한 파라미터이다. BER이 10^{-4} 인 경우, 랜덤 부호에 비해 0.25dB, quasi-cyclic LDPC(full-algebra LDPC) 부호에 비해 0.22dB의 부호화 이득을 얻을 수 있음을 알 수 있다^[8]. 또한, 제안 LDPC 부호의 패리티 검사 행렬은 각 부행렬 단위로 독립 연산이 가능하고, 각 부행렬에 대한 연산구조가 동일하기 때문에 고속 병렬처리 구조로의 확장 및 VLSI의 구현이 용이할 뿐만 아니라, 순환 시프트의 특성 이용하여 연산 속도를 배가할 수 있는 하드웨어 구현 측면에서의 이점을 갖는다.



(a) E_b/N_0 vs BER



(b) 평균 반복 회수 vs E_b/N_0

그림 9 반집합 대수 LDPC 부호의 성능

참고 문헌

- [1] Li Ping, W. K. Leung, and Nam Phamdo, Low density parity check codes with semi-random parity check matrix, Electronics Letters 35 (1999), No. 1, 38-30
- [2] Rich Echard and S. C Chang, Deterministic π -rotation low-density parity check codes, Electronics Letters 38 (2002), No. 10, 464-465.
- [3] Dariush Divsalar, Hui Jin, and Robert J. McEliece, Coding theorems for turbo-like codes, Proceedings of the 36th Allerton Conference on Communication, Control and Computing (1998), 201-210.
- [4] D J. C. Mackay, "Good Error-Correcting Codes Based on Very Sparse Matrices," IEEE Trans. Information. Theory, vol 45, pp. 399-431, Mar. 1999
- [5] Kou, Y., Lin, S. and Fossorier. M P C, "Low density parity check codes onstruction based on finite geometries," Global Telecommunications Conference, 2000 GLOBECOM '00 IEEE, Volume: 2, 27 Nov -1 Dec 2000, Page(s) 825 -829 vol 2
- [6] Yu Yi and Moon Ho Lee, "Semi-Algebraic Low-Density Parity-Check SA-LDPC) Codes,"

3rd International Symposium on Turbo Codes and Related Topics, Brest, France, Sept 5th, 2003. Page(s): 379-382.

- [7] Rich Echard, On the Construction of Some Deterministic Low-Density Parity-Check Codes, Ph.D. thesis, George Mason University, Semester 2002.
- [8] Marc. Fossorier, "Quasi-Cyclic Low-Density Parity-Check Codes," International Symposium on Information Theory, Japan, June 4th, 2003.

이 동 민(Dong-min Lee)



2002년 2월 : 전북대학교 정보통신공학과 졸업
2004년 8월 : 전북대학교 정보통신공학과 석사수료

<관심분야> 차세대 이동통신, 멀티미디어 통신

이 광 재(Kwang-jae Lee)

정회원



1986년 2월 : 전북대학교 전자공학과 졸업
1990년 2월 : 전북대학교 전자공학과 석사
1995년 2월~현재 : 전북대학교 전기공학과 박사수료

1995년 3월~현재 : 한려대학교 멀티미디어정보통신공학과 전임강사

<관심분야> 이동통신, RF-ID, 전력선통신

이 문 호(Moon-ho Lee)

정회원



1967년 2월 : 전북대학교 전기공학과 졸업
1976년 2월 : 전북대학교 전기공학과 석사
1990년 2월 : 일본 Tokyo University 정보통신공학과 공학박사

1981년 3월~현재 : 전북대학교 전기전자공학부 교수

<관심분야> 이동통신, 정보이론, UWB