
재킷 패턴 기반의 F-LDPC 부호

이광재* · 강승선**

An F-LDPC Codes Based on Jacket Pattern

Kwang-Jae Lee* · Seung-Son Kang**

요약

본 논문에서 우리는 LDPC 부호의 부호화 문제를 고려한다. 특히 재킷 패턴과 순환 치환 행렬을 사용함으로써 리차드슨의 하삼각 행렬과 유사한 간단한 LDPC 부호를 제안한다. 이 부호화 기법은 다양한 부호율을 갖도록 확장할 수 있다. 또한 단순한 행렬 처리에 근거하여 다양한 부호율을 갖는 복잡도가 낮고 간단한 부호기를 설계할 수 있다.

ABSTRACT

In this paper, we consider the encoding scheme of Low Density Parity Check codes. In particular, using the Jacket Pattern and circulant permutation matrices, we propose the simple encoding scheme of Richardson's lower triangular matrix. These encoding scheme can be extended to a flexible code rate. Based on the simple matrix process, also we can design low complex and simple encoders for the flexible code rates.

키워드

LDPC, Jacket pattern, Lower triangular matrix, Hadamard
저밀도 패리티 검사 부호, 재킷 패턴, 하삼각 행렬, 아다마르

1. 서론

LDPC(Low Density Parity Check) 부호에서 다루어야 될 주요 문제 중 하나는 명백히 부호화 복잡도를 개선하는 것이다[1-3]. 이에 리차드슨은 LDPC 부호의 근사 하삼각 분해에 근거하여 효율적인 일반적 부호화 알고리즘을 소개했다[4]. 이러한 리차드슨 형태의 계산은 여러 서브 행렬의 연산 과정들에 근거한다. 그래서 만약 패리티 검사 행렬 H의 선택이 적절하게 되지 않으면 변환과 행렬 처리과정의 복잡도는 더 높아진다[5-7]. 따라서 cycle-4를 쉽게 피하고 유

동적인 부호율과 부호어의 길이를 지원하는 재킷 패턴(Jacket pattern)과 단위행렬로부터 열 시프트를 통해 얻을 수 있는 순환행렬(CPM, Circulant Permutation Matrices)로부터 간단한 H 패리티 검사 행렬을 제안한다. 또한 가변 부호율 F-LDPC(Flexible LDPC) 부호를 연구한다. 2장에서 재킷 행렬의 간단한 소개와 재킷 순환 행렬을 통한 기존의 부호화 방법을 간단히 알아보고, 3장에서는 확장이 가능한 재킷 순환 행렬에 대해 알아본다. 그리고 4장에서 이에 따른 결론을 알아본다.

* 한려대학교 멀티미디어정보통신공학과(kjlee@hanlyo.ac.kr)

** 광양보건대학교 컴퓨터정보과(sskang@kwangyang.ac.kr)

접수일자 : 2012. 01. 31

심사(수정)일자 : 2012. 03. 19

게재확정일자 : 2012. 04. 07

II. 재킷 패턴과 순환 치환 행렬에 의한 부호화

재킷 정방행렬을 $[A]_N = [a_{ij}]_N$ 로 볼 때, 그것의 역행렬은 단지 각 성분들의 역인 특징을 가진다. 즉, $[A]_N^{-1} = \frac{1}{N} [1/a_{ij}]_N$, $1 \leq i, j \leq N$, 이것을 재킷 행렬이라 부른다. 아다마르 행렬, DFT 등의 대다수의 흥미로운 행렬들 역시 재킷 행렬 계열에 포함된다[8-9]. 본 논문에서는 복소수 필드 상에서 재킷 행렬의 계열을 보인다. P 를 홀수 소수라 하고 $\alpha = e^{\sqrt{-1}(2\pi/P)}$. 그래서 $\alpha^P = 1$ 이고 $\langle a \cdot b \rangle$ 연산을 하는 $F_p = \{0, 1, 2, \dots, p-1\}$ 는 유한체이고 여기서

$$\langle a \rangle^{\Delta} = a \bmod p \tag{1}$$

$a \in F_p$ 라고 하고, 다음과 같이 함수를 정의하면 $f_a(x) = \langle a \times x \rangle^{\Delta}$ 이다. $\vec{V} = (\alpha^{v_0}, \alpha^{v_1}, \dots, \alpha^{v_{p-1}})$ 는 벡터이고 여기서 $v_i \in F_p$ ($i = 0, 1, \dots, p-1$) 이다.

먼저 다음의 벡터를 정의한다.

$$\vec{V}_a = (\alpha^{f_a(v_0)}, \alpha^{f_a(v_1)}, \dots, \alpha^{f_a(v_{p-1})}) \tag{2}$$

그러면 $p \times p$ 재킷 행렬의 형태는 식 (3)과 같다.

$$[J]_p = \begin{bmatrix} \vec{V}_0 \\ \vec{V}_1 \\ \vdots \\ \vec{V}_{p-1} \end{bmatrix} \tag{3}$$

예를 들어 행렬의 크기 $p = 5$ 라 하면 재킷 행렬의 구성은 식 (4)와 같다.

$$[J]_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ 1 & \alpha^2 & \alpha^4 & \alpha & \alpha^3 \\ 1 & \alpha^3 & \alpha & \alpha^4 & \alpha^2 \\ 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha \end{bmatrix},$$

$$([J]_5)^{-1} = \frac{1}{5} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha \\ 1 & \alpha^3 & \alpha & \alpha^4 & \alpha^2 \\ 1 & \alpha^2 & \alpha^4 & \alpha & \alpha^3 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \end{bmatrix} \tag{4}$$

여기서 재킷 행렬을 $[J]_p$ 라 하고 α^i 를 크기 $p \times p$ 의 순환 행렬 E^i 로 교체하면 저밀도 재킷 행렬을 얻을 수 있다. 예를 들어 행렬 $[J]_5$ 는 식 (5)와 같이 바꿀 수 있다.

$$[J]_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 \\ 1 & \alpha^2 & \alpha^4 & \alpha & \alpha^3 \\ 1 & \alpha^3 & \alpha & \alpha^4 & \alpha^2 \\ 1 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha \end{bmatrix} \Rightarrow \begin{bmatrix} I & I & I & I & I \\ I & E & E^2 & E^3 & E^4 \\ I & E^2 & E^4 & E & E^3 \\ I & E^3 & E & E^4 & E^2 \\ I & E^4 & E^3 & E^2 & E \end{bmatrix},$$

$p = 5$ (5)

여기서 행과 열의 소거(canceling)에 의해 쉽게 비정방 패리티 검사 행렬 H를 얻을 수 있다. 예를 들어 크기 $3p \times kp$ 인 $(3, k)$ 균일 LDPC 부호는 $[J]_p$ 패턴의 저밀도 행렬에 의해 설계되고 $p \geq k$ 이면 그림 1에서와 같이 보여진다.

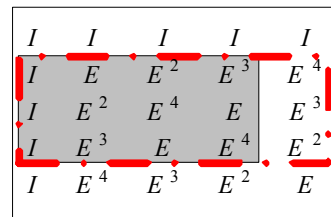


그림 1. 저밀도 행렬의 일부로부터의 H 행렬
Fig. 1 The H matrix from a part of the low density matrix.

이를 바탕으로 리차드슨 부호화 방법에 적용 될 수 있다.

$$\begin{bmatrix} I & 0 \\ ET^{-1} & I \end{bmatrix} \cdot \begin{bmatrix} A & B & T \\ C & D & E \end{bmatrix} = \begin{bmatrix} A & B & T \\ -ET^{-1}A+C & -ET^{-1}B+D & 0 \end{bmatrix} \quad (6)$$

패리티 검사 행렬에서 T 는 하삼각 형태가 된다. 여기서 T 는 $P \times P$ 단위행렬이고, 나머지 $\{A, B, C, D, E\}$ 는 좀 더 간단한 연산의 여러 순환 행렬에 의해서 구해진다.

예를 들어 $p=5$ 인 경우 재킷 패턴 LDPC 부호는 다음에 의해서 구해진다. 여기서 그림 1의 일부를 취함으로써 원하는 부호율에 맞는 행렬을 구성할 수 있다.

$$\begin{bmatrix} I & I & I & I & I \\ I & E & E^2 & E^3 & E^4 \\ I & E^2 & E^4 & E & E^3 \\ I & E^3 & E & E^4 & E^2 \\ I & E^4 & E^3 & E^2 & E \end{bmatrix} \Rightarrow \begin{bmatrix} I & E & E^2 & E^3 \\ I & E^2 & E^4 & E \\ I & E^3 & E & E^4 \end{bmatrix} \xrightarrow{\text{Reverse}} \begin{bmatrix} E^3 & E^2 & E & I \\ E & E^4 & E^2 & I \\ E^4 & E & E^3 & I \end{bmatrix} \quad (7)$$

여기서 각각의 부행렬은 식 (8)과 같이 구성된다.

$$\begin{aligned} A &= E^3, & B &= \begin{bmatrix} E^2 & E \end{bmatrix}, & T &= I_5 \\ C &= \begin{bmatrix} E \\ E^4 \end{bmatrix}, & D &= \begin{bmatrix} E^4 & E^2 \\ E & E^3 \end{bmatrix}, & E &= \begin{bmatrix} I_5 \\ I_5 \end{bmatrix} \end{aligned} \quad (8)$$

그에 따른 패리티 검사 비트는 식 (9)에 의해 얻어진다.

$$\begin{aligned} p_1^T &= -\phi^{-1}(-ET^{-1}A+C)s^T, \\ p_2^T &= -T^{-1}(As^T + Bp_1^T), \end{aligned} \quad (9)$$

여기서 $\phi = -ET^{-1}B + D$ 이고, $T = I$ 이다. 따라서 다음과 같이 간단한 형태로 요약된다.

$$p_1^T = -\phi^{-1} \left(- \begin{bmatrix} A \\ \vdots \\ A \end{bmatrix} + C \right) s^T, \quad p_2^T = -(As^T + Bp_1^T) \quad (10)$$

$$\phi = -EB + D = - \begin{bmatrix} B_1 \dots B_{(m-p)/p} \\ \vdots \\ B_1 \dots B_{(m-p)/p} \end{bmatrix} + D \quad (11)$$

$$B = \begin{bmatrix} B_1 & \dots & B_{(m-p)/p} \end{bmatrix} \quad (12)$$

여기서 각 요소 B_i 는 $p \times p$ 치환행렬이다 ($1 \leq i \leq (m-p)/p$). 리차드슨 유형의 재킷 패턴 LDPC 부호는 순환 행렬과 그것들의 곱셈을 사용함으로써 간단히 부호화 되어질 수 있다. 또한 이런 계열의 LDPC 부호화의 계산 복잡도는 낮아진다.

한편 비균일 LDPC 부호들도 위와 유사하게 간단히 부호화 되어 질 수 있다. 가령 $p=3$ 이라 하고, 재킷 행렬을 식 (13)이라 한다.

$$\text{Jacket} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \end{bmatrix} \quad (13)$$

여기에 부호율에 맞게 자른 후 0을 추가하여 확장 행렬을 얻는다. 여기에서 0의 삽입은 저밀도 행렬을 생성하면서 리차드슨이 제안한 부호화 과정에서 요구되는 역행렬의 복잡도를 감소화 시키는 역할을 한다. 리차드슨의 하삼각 구조에서 볼 때, T의 부행렬을 단위행렬로 놓아 역행렬의 복잡도를 줄이기 위해 좌우 패턴을 바꾸어 H행렬을 생성하면 다음과 같다.

$$\text{Jacket}_{\text{Expansion}} = \begin{bmatrix} 1 & \alpha & 0 & \alpha^2 \\ 1 & 0 & \alpha^2 & \alpha \end{bmatrix} \quad (14)$$

$$H = \begin{bmatrix} \alpha^2 & 0 & \alpha & 1 \\ \alpha & \alpha^2 & 0 & 1 \end{bmatrix} \quad (15)$$

$$H = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

따라서 H행렬을 다음과 같이 분할할 수 있다. 여기서

$$T = 1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$A = [\alpha^2 \ 0] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \alpha = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$C = [\alpha \ \alpha^2] = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix},$$

$$D = 0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad E = 1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

같은 방법으로 $-ET^{-1}A+C$ 과 $-ET^{-1}B+D$ 을 이용하여 p1과 p2를 계산할 수 있다.

III. 확장된 재킷 패턴 LDPC 부호화

재킷 패턴 부호화 방법으로 간단하게 H행렬로부터 부호화를 가능하게 할 수 있었다. 하지만 H행렬 특성상 p를 소수로 정의했기 때문에 원하는 부호어의 길이와 부호율이 맞지 않을 수 있다. 부호화 하는 과정에서 T와 E의 형태가 p1과 p2를 계산하는데, 중요하게 작용하게 되는데, T와 E의 형태가 $p \times p$ 의 형태로 구성되어진다. 물론 열의 무게를 증가시키도록 E

의 형태가 $\alpha p \times p$ (여기서 α 은 자연수)로 만들 수도 있다. 이러한 경우에는 영행렬을 삽입하여 Φ^{-1} 의 연산을 쉽게 할 수 있다. 어떠한 경우라도 n-k의 경우가 p의 배수 이어야 함은 분명하다. 이러한 점에서 재킷 패턴 LDPC 부호의 부호율은 자유롭지 못하다.

이하에서는 재킷 패턴 LDPC를 확장시키는 방법에 대해 살펴본다. H행렬의 기본 구조는 다음과 같다.

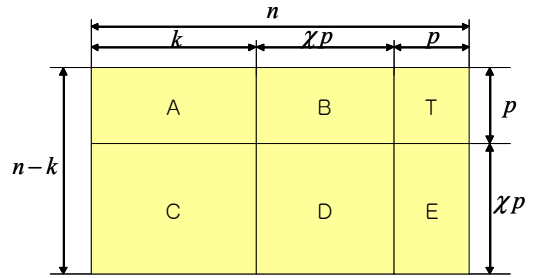


그림 2. H행렬의 기본 구조
Fig. 2 Primary structure of H matrix

여기서 n은 부호 길이를 의미하고, k는 정보길이를 의미한다. 또한 p는 H행렬을 구성하는 기본행렬의 크기이고 또한 소수이다. 여기서 부행렬 B와 E행렬은 χ 개의 기본행렬로 구성되어있고 부행렬 D는 $\chi p \times \chi p$ 의 크기를 갖는 행렬이다. 임의의 k가 정보 길이로 볼 수 있는지 알아보기 위해 다음 행렬 크기로 전개해 볼 수 있다. 앞에서 보인 바와 같이 패리티 비트는 다음과 같이 구해진다.

$$p_1^T = -\phi^{-1}(-ET^{-1}A+C)s^T$$

$$p_2^T = -T^{-1}(As^T + Bp_1^T) \quad (17)$$

여기서 T=I이고 모듈로 2 연산을 하기 때문에 다음과 같이 간단하게 표현할 수 있다.

$$p_1^T = \phi^{-1}(EA+C)s^T$$

$$p_2^T = As^T + Bp_1^T \quad (18)$$

이를 바탕으로 임의의 k로 부호화가 이루어지는지 알아보기 위해 행렬의 크기만을 고려해보면 다음과 같다.

$$\begin{aligned}
 [\phi]_{\text{dim}} &= [EB + D]_{\text{dim}} \\
 &= [E]_{\chi p \times p} [B]_{p \times \chi p} + [D]_{\chi p \times \chi p} \\
 &= [\phi]_{\chi p \times \chi p} \\
 [\phi^{-1}]_{\text{dim}} &= [\phi]_{\text{dim}} = [\phi]_{\chi p \times \chi p}
 \end{aligned} \tag{19}$$

이므로 다음이 성립한다.

$$\begin{aligned}
 [p_1^T]_{\text{dim}} &= [\phi^{-1}]_{\chi p \times \chi p} \{ [E]_{\chi p \times p} [A]_{p \times k} + [C]_{\chi p \times k} \} [s^T]_{k \times 1} \\
 &= [p_1^T]_{\chi p \times 1} \\
 [p_2^T]_{\text{dim}} &= [A]_{p \times k} [s^T]_{k \times 1} + [B]_{p \times \chi p} [p_1^T]_{\chi p \times 1} \\
 &= [p_2^T]_{p \times 1}
 \end{aligned} \tag{20}$$

따라서 임의의 k 에도 부호화가 가능함을 알 수 있다. 이때, $1 \leq k \leq n - (\chi + 1)p$ 이 성립해야 한다. 따라서 영행렬을 삽입하지 않았을 경우 $\max(n) = p^2$ 이어야 하므로 $1 \leq k \leq p^2 - (\chi + 1)p$ 이어야 하고, 영행렬을 삽입하였을 경우 $\max(n) = p(p + 1)$ 이므로 $1 \leq k \leq p^2 - \chi p$ 이어야 한다.

임의의 k 가 주어졌다고 가정할 경우, 부호율은 $R = \frac{k}{n}$ 이다. 여기서 n 을 줄이면 앞의 행렬 계산은 더 이상 성립하지 않게 되므로 부호화가 불가능하게 된다. 하지만 식 (21)와 같이 만들 수 있다면 부호율의 변경이 가능하다고 할 수 있다.

$$R_{\text{new}} = \frac{k}{n + \varepsilon} = \frac{k}{n'} \tag{21}$$

이전의 부호화로 얻어진 부호어는

$cw_{\text{old}} = \{s, p_1, p_2\}$ 이다. 여기서 s 는 정보 비트이고 p_1 과 p_2 는 부호화로 얻어진 패리티 비트이다. 이를 바탕으로 새로운 패리티 비트 p_3 를 간단한 RA(repeat accumulate) 부호기로 생성할 수 있다.

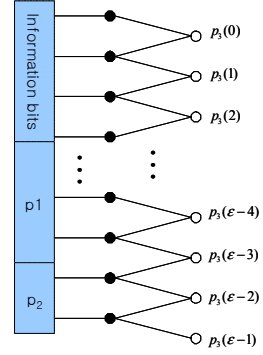


그림 3. RA 부호기에 대한 Tanner 그래프 (형태 1)
Fig. 3 Tanner graph for RA encoder (type 1)

이전 부호어의 각 비트를 표본으로 놓고 ε 개의 제 3 패리티 검사 비트를 생성하기 위해 부호어에 대한 표본 비트의 위치를 균일한 간격으로 놓을 때, 추출한 비트간의 간격은 $\{k + p(\chi + 1)\} / \varepsilon$ 이 될 것이다. 이때의 새로운 부호어는 다음과 같이 구성된다.

$$cw_{\text{new}} = \{cw_{\text{old}}, p_3(0), p_3(1), p_3(2), \dots, p_3(\varepsilon - 1)\} \tag{22}$$

이러한 경우 재귀 공식은 (23)과 같이 나타낼 수 있다.

$$p_3(j) = p_3(j - 1) + v(j) \tag{23}$$

이전 부호어의 샘플의 위치를 $\Lambda(x)$, $x \in \{0, 1, 2, \dots, \varepsilon - 1\}$ 로 정의할 때, 다음이 성립한다.

$$\Lambda(x) = \frac{k + p(\chi + 1)}{\varepsilon} x + \alpha \tag{24}$$

여기서 α 는 $0 < \alpha \leq \frac{k + p(\chi + 1)}{\varepsilon}$ 의 범위에 존재한다.

그리고 $\Lambda(x)^*$ 를 $\Lambda(x)$ 의 위치에 존재하는 샘플의 값이라고 정의한다면, 다음 식이 성립한다.

$$p_3(x) = \Lambda(x)^* + p_3(x - 1), \text{ 단 } p_3(-1) = 0 \tag{25}$$

부호기를 그림으로 표현하면 다음과 같다.

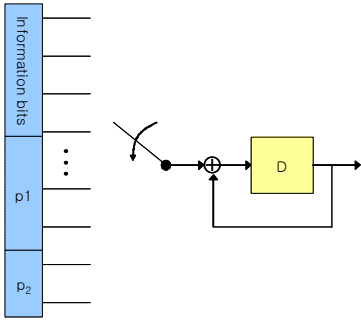


그림 4. 확장된 재킷 패턴의 LDPC 부호기
Fig. 4 Extend jacket pattern LDPC encoder

이러한 방법은 이전 부호어 CW_{old} 로부터 \mathcal{E} 개의 표본을 일정한 간격으로 추출한 경우이며, 일반적으로 인터리버가 이전 단계에서 첨가될 수 있다. 그러한 경우 인터리버 크기는 \mathcal{E} 가 될 것이다.

같은 방법으로 이전 부호어로부터 \mathcal{E} 개의 샘플을 추출하는 대신 모든 부호어를 전부 표본으로 쓸 수 있다. 다음 그림과 같이 다른 종류의 RA 부호를 생각해볼 수 있다[10].

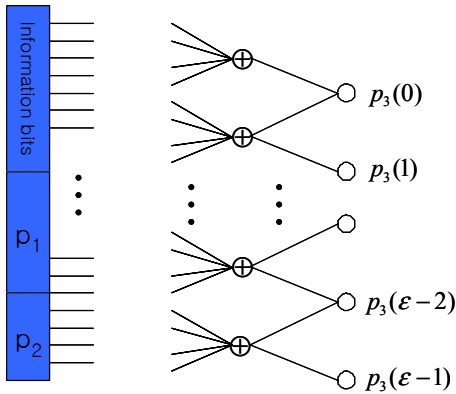


그림 5. RA 부호기에 대한 Tanner 그래프 (형태 2)
Fig. 5 Tanner graph for RA encoder (type 2)

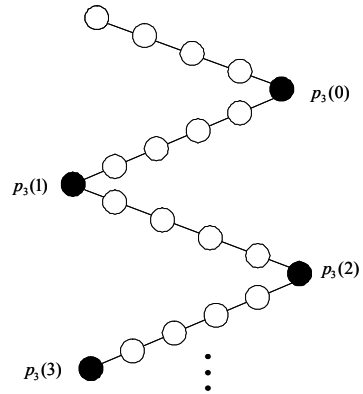


그림 6. 지그재그 부호기에 대한 Tanner 그래프
Fig. 6 Tanner graph for zigzag encoder

위의 Tanner 그래프는 $n_{old} = (s, p_1, p_2)$ 인 부호어와 \mathcal{E} 개의 패리티 비트를 보여주고 있다. 그리고 세 번째의 패리티 비트는

$p_3 = \{p_3(0), p_3(1), \dots, p_3(\mathcal{E}-1)\}$ 가 될 것이다. 패리티 비트의 값은 모듈로 2연산에 의해 결정된다. 이러한 경우 다음과 같은 순환 공식이 성립하게 된다.

$$p_3(j) = p_3(j-1) + \sum_{i=1}^a v((j-1)a+i) \quad (26)$$

같은 방법으로 지그재그 부호를 고려할 수 있다 [11]. 같은 라인에 있는 모든 변수들의 모듈로 2 연산의 합은 0이 된다. 이러한 경우 패리티 비트는 반복적으로 다음 식과 같이 쓸 수 있다.

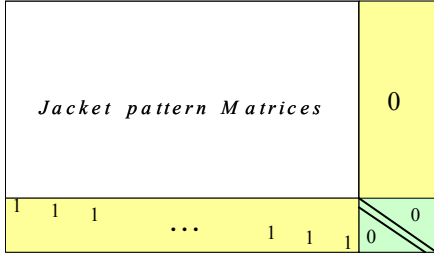
$$p(0) = \sum_{j=1}^J d(i, j) \bmod 2$$

$$p(i) = \sum_{j=1}^J d(i, j) + p(i-1) \bmod 2 \quad (27)$$

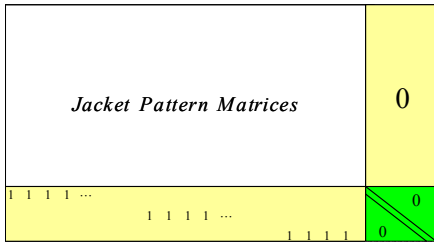
지금까지 확장된 재킷 패턴 LDPC 부호에 대한 부호기를 고려했다. 이에 따라 패리티 검사 행렬 H를 설계할 수 있다. 이를 그림 7에 나타내었다.

형태 1을 분석해보면, \mathcal{E} 개의 비트를 부호어로부터 추출할 때, 재킷 패턴 행렬의 하단은 식 (28)과 같이 구성되어진다. 이러한 형태는 인터리버를 무시하고,

패리티 검사 비트 사이의 간격을 동일하게 했을 경우의 형태이다.



(a)



(b)

그림 7. 확장된 재킷 패턴 H 행렬 (a) 형태1 (b)형태 2
Fig. 7 Extend jacket pattern H matrix (a) type 1 (b) type 2.

$$H_1 = \begin{bmatrix} 1 & 0 \dots & 0 & 0 \dots & 0 & 0 \dots \\ 0 & 0 \dots & 1 & 0 \dots & 0 & 0 \dots \\ 0 & 0 \dots & 0 & 0 \dots & 1 & 0 \dots \end{bmatrix} \quad (28)$$

형태 2를 대상으로 했을 때, H_1 은 식 (29)과 같이 구성되어진다.

$$H_1 = \begin{bmatrix} 1111 \dots & 0 & 0 \\ 0 & 1111 \dots & 0 \\ 0 & 0 & 1111 \dots \end{bmatrix} \quad (29)$$

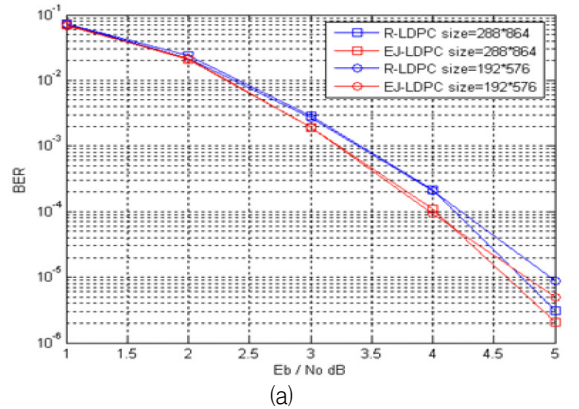
여기서 H_1 의 행렬 크기는 $[H_1]_{\mathcal{E} \times n_{old}}$ 이다. 재킷 패턴 행렬의 오른쪽 부분은 영행렬로 구성된다. 행렬 크기는 $[Zeros]_{(n_{old}-k) \times \mathcal{E}}$ 가 된다. H 행렬을 살펴보면, $n_{new} = n_{old} + \mathcal{E}$ 관계식을 쉽게 얻을 수 있고,

H 행렬의 열의 길이는 $n_{old} - k + \mathcal{E}$ 이다. 여기서 k 는 정보어의 길이이다. 일반적으로 열의 길이와 행의 길이의 차이가 정보어의 길이이다. 따라서 확장된 재킷 행렬의 정보어의 길이를 구하면

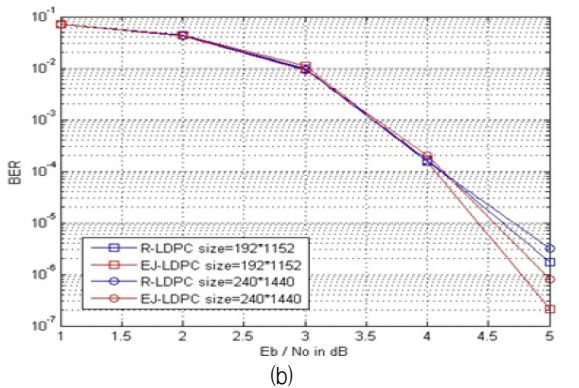
$n_{new} - (n_{old} - k + \mathcal{E}) = k$ 으로 정보어의 길이가 확장 전과 동일함을 알 수 있다.

그림 7에서의 우측 하단 부분은 식 (30)과 같은 이중 대각행렬(dual diagonal matrix)로 구성된다.

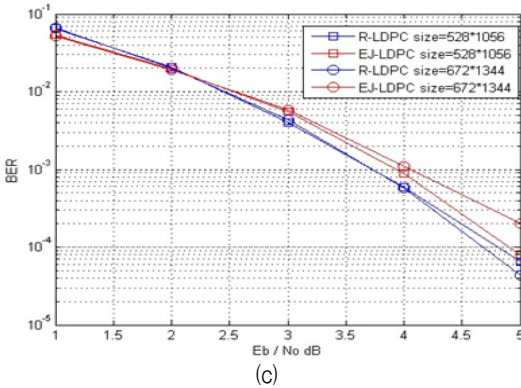
$$H_2 = \begin{bmatrix} 1 & & & & 0 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & 1 \\ 0 & & & & 1 & 1 \end{bmatrix} \quad (30)$$



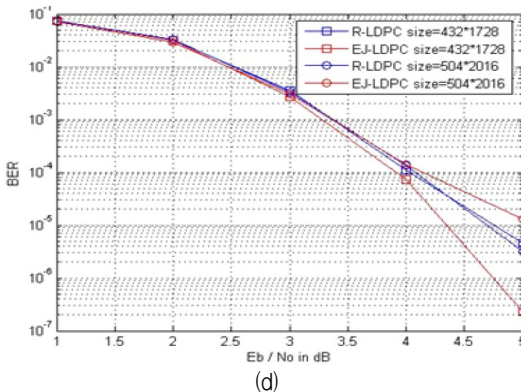
(a)



(b)



(c)



(d)

그림 8. 전산 실험 결과
Fig. 8 Simulation results.

IV. 전산 실험

랜덤 설계 LDPC를 R-LDPC로 정의하고, 이것이 (3,k) 균일 LDPC 부호라고 한다. 재킷 패턴 LDPC를 J-LDPC로 정의한다. 같은 방법으로 확장된 재킷 LDPC를 EJ-LDPC로 정의한다.

모든 실험은 5000 프레임으로 얻어진 결과이다. 또한 다양한 부호율 R=2/3,5/6,1/2,3/4에 대해 시도하였다. 순수 J-LDPC로는 원하는 부호율을 완성할 수 없기에 EJ-LDPC를 사용하였다. 그림 8에서 실험 결과를 보여준다. 행렬 크기가 (288*864), (192*576)이고 부호율이 2/3인 경우 대략 0.1~0.2dB의 성능 향상을 보였고, 행렬 크기가 (192*1152), (240*1440)이고 부호율이 5/6인 경우 BER=10⁻⁵에서 0.25dB 정도의 성능 향상을 보였다. 그리고 행렬 크기가 (528*1056),

(672*1344)이고 부호율이 1/2인 경우 10⁻³에서 0.3dB 정도의 성능 열화를 보인다. 그리고 (432*1728), (504*2016)이고 부호율이 3/4인 경우에는 흥미로운 결과를 보인다. BER이 10⁻⁵에서 전자는 0.5dB 성능 향상을 보이고 후자는 0.3dB 정도의 성능 열화를 보인다. 이는 \mathcal{E} 의 값과 부호어의 길이가 성능 차이를 발생시킨 것으로 보인다.

V. 결론

이 논문에서는 적은 복잡도를 가진 부호기 설계와 재킷 패턴을 사용한 유연한 부호어와 부호율을 연구했다. 재킷 패턴의 특징상 cycle-4가 만들어지지 않고, RA 모델과 Zigzag 모델은 유연한 부호율을 결정할 수 있다. 또한 RA 모델의 추가적인 인코딩으로 인한 cycle-4가 발생하지 않는다.

재킷 패턴 LDPC 부호와 확장된 재킷 LDPC 부호의 주요 관심은 낮은 복잡도를 가진 빠른 부호기 설계, 변경 가능한 부호율 및 부호어의 길이이다. 일반적으로 RA코드를 추가시킴으로써 마지막 패리티 검사 비트의 열 무개가 1이 되어 성능 열화를 가져올 수 있지만, 이러한 설계의 장점으로 유연한 부호어 길이 조절이 가능하다는 점과 원하는 부호율을 얻을 수 있다는 점이다. 또한 부호화 방법에서도 기존의 부호기에 단순한 RA 부호기를 덧붙임으로써 효과적인 인코딩이 가능하다.

참고 문헌

- [1] Y. Kou, S. Lin, and M. Fossorier, "Low density parity check codes based on finite geometries: A rediscovery," Proc. of ISIT 2000, Sorrento, Italy, pp. 22-30, 2000.
- [2] P. Li, W.K. Leung, and N. Phamdo, "Low density parity check codes with semi-random parity check matrix," IEE Electronics Letters, Vol. 35, No. 1, pp. 38-39, 1999.
- [3] M. Yang, Y. Li, and W.E. Ryan, "Design of efficiency encodable moderated-length high rate irregular LDPC codes," IEEE Trans. Comm., Vol. 52, pp. 564-571, 2004.

[4] T.J. Richardson and R.L. Urbanke, "Efficient Encoding of Low Density Parity Check Codes", IEEE Trans. IT, Vol. 47, pp. 638-856, 2001.

[5] 유진우, 임형규, 박세원, "정규성을 허용하는 특별한 부호화 행렬의 구성", 한국전자통신학회논문지, 6권, 2호, pp. 193-198, 2011.

[6] 서희중, "Node Monitoring 알고리즘과 NP 방법을 사용한 효율적인 LDPC 복호방법", 한국전자통신학회논문지, 6권, 1호, pp. 20-26, 2011.

[7] 홍완표, "데이터통신 전송효율과 ASCII 부호체계 고찰", 한국전자통신학회논문지, 6권, 5호, pp. 657-664, 2011.

[8] M.H. Lee, "A New Reverse Jacket Transform and Its Fast Algorithm", IEEE Trans. On Circuit and System 2, Vol. 47, No. 1, pp. 39-47, 2000.

[9] M.H. Lee, B.S. Rajan, and J.Y. Park, "A Generalized Reverse Jacket Transform," IEEE Trans. Circuits Syst. II, Vol. 48, No. 7, pp. 684-690, 2001.

[10] Hui Jin, Aamod Khandekar, and Rovert McEliece "Irregular Repeat-Accumulate Codes", 2nd Int. symp. on Turbo codes and Related Topics, Brest, France, pp. 1-9, 2000.

[11] Li Ping, Xiaoling, N.Phamdo, "Zigzag Codes and Concatenated Zigzag Codes", IEEE Transactions on Information Theory, Vol. 47, No. 2, pp. 800-807, 2001.



강승선(Seung-Son Kang)

1992년 전북대학교 정보통신공학과 졸업(공학사)

1995년 전북대학교 대학원 정보통신공학과 졸업(공학석사)

1998년 전북대학교 대학원 컴퓨터공학과 박사과정 수료

1995년~현재 광양보건대학교 컴퓨터정보과 교수

※ 관심분야 : 디지털신호처리, 부호이론

저자 소개



이광재(Kwang-Jae Lee)

1986년 전북대학교 전자공학과 졸업(공학사)

1990년 전북대학교 대학원 전자공학과 졸업(공학석사)

2006년 전북대학교 대학원 전기공학과 졸업(공학박사)

1995년~현재 한려대학교 멀티미디어정보통신공학과 교수

※ 관심분야 : 부호이론, 이동통신