제33회 한국정보처리학회 춘계학술발표대회 논문집 제17권 제1호 (2010. 4)

여러 부행렬들의 무작위 조합으로 만든 Quasi-Cyclic LDPC 부호

황용수*, 오상헌*, 전문구*
*광주과학기술원 정보통신공학과
e-mail : {yshwang, oosshoun, mgjeon}@gist.ac.kr

Quasi-Cyclic LDPC Codes by random combination of multiple sub-matrices

Yongsoo Hwang*, Sanghoun Oh*, Moongu Jeon*
*Department of Information and Communications,
Gwangju Institute of Science and Technology

요 약

기존의 Quasi-Cyclic LDPC 부호는 하나의 기본행렬의 순환행렬을 부행렬로 사용하여 패리티 검사 행렬을 만든다. 본 논문에서는 무게가 서로 다른 두 개의 기본 행렬의 순환행렬들과 영행렬을 부행렬로 사용하고, 이 세 개의 부행렬들을 주어진 조건하에서 무작위로 조합하여 패리티 검사 행렬을 만드는 방법을 제안한다. 제안된 LDPC 부호는 girth가 6이상인 Irregular LDPC 부호이다.

1. 서론

1960년대 초반 R. G. Gallager에 의해서 처음 제안된 Low Density Parity Check (LDPC) 부호는 이론적으로 Shannon Limit에 근접하는 뛰어난 성능에도 불구하고, 여러 가지 문제점들과 오해로 인해 실제로 활용되지 못하였다. 이후 R. M. Tanner를 비롯하여 David. J. C. Mackay와 Radford M. Neal 등 여러 학자들에 의해 그 효용성이 재발견되었고, 현재는 가장 각광을 받고 있는 부호이며, 다양한 통신환경에서 활용되기 위해 다양한 연구가 계속 진행되고 있다 [1~6].

LDPC 부호는 블록 선형 부호로 0에 비해 1이 상대적으로 매우 적은 패리티 검사 행렬을 갖는 부호이다. LDPC 부호는 패리티 검사 행렬을 만드는 방법론의 관점에서 무작위 LDPC 부호와 구조적 LDPC 부호로 구분하고, 패리티 검사 행렬의 행과 열에 1 의 개수의 고정유무에 따라 Regular LDPC 와 Irregular LDPC 로 나뉜다. 일반적으로 복호화의 관점에서 무작위 LDPC 부호와 Irregular LDPC 부호의 성능이 더 우수한 것으로 알려져 있다.

Quasi-Cyclic LDPC 부호는 구조적 LDPC 부호의 대표적인 예로 일반적으로 LDPC 부호가 갖는 높은 부호화 비용의 단점을 shift register를 이용함으로 인하여 해소할 수 있다 [1] [6]. 본 논문에서는 하나의 행렬의 순환행렬을 부행렬로 사용하는 기존의 Quasi-Cyclic LDPC와 달리 무게를 달리하는 두 개 행렬의 순환행렬들과 영행렬을 부행렬로 사용하는 방법을 제안한다. 또한 이 세 개의 부행렬들을 주어진 조건 하에서 무작위로 조합하여 패리티 검사 행렬을 만든다.

다음 장에서는 기본적인 Quasi-Cyclic LDPC 부호에 대해 간단히 언급하며, LDPC 부호가 갖는 몇 가지 특성에 대해 언급한다. 3장과 4장에서는 제안하는 방법의 구체적인 내용과 실험결과에 대해서 다루며, 5장에서 결론을 맺는다.

2. Quasi-Cyclic LDPC 부호

Quasi-Cyclic LDPC 부호는 단위행렬 같은 기본행렬의 순환행렬을 부행렬로 사용하여 전체 패리티 검사 행렬을 만든 부호화 방법이며, 패리티 검사 행렬은 아래와 같은 모습이다 [1] [3].

$$H=\begin{bmatrix} I^{C_{0,0}} & I^{C_{0,1}} & I^{C_{0,2}} & \cdots & I^{C_{0,L-1}} \\ I^{C_{1,0}} & I^{C_{1,1}} & I^{C_{1,2}} & \cdots & I^{C_{1,L-1}} \\ I^{C_{2,0}} & I^{C_{2,1}} & I^{C_{2,2}} & \cdots & I^{C_{2,L-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I^{C_{J-1,0}} & I^{C_{J-1,1}} & I^{C_{J-1,2}} & \cdots & I^{C_{J-1,L-1}} \end{bmatrix}$$

수식 1. 일반적인 QC LDPC 부호의 패리티 검사 행렬

$I^{C_{i,j}}$ 는 단위행렬 I 의 각 요소들이 $C_{i,j}$ 만큼 shift 된 행렬을 의미한다. 따라서 $H_{shift}=[C_{i,j}]$ 같은 shift index 행렬이 정의되어야 한다. 이때 패리티 검사 행렬의 크기는 사용된 단위행렬의 크기 p에 비례하는데, $(Jp)\times(Lp)$이다. 또한 부호 율은 $K/N \simeq (Lp-Jp)/Lp = 1-J/L$ 이다. "=" 기호 대신 "≃" 를 사용한 이유는 H 행렬이 full rank 인지의 여부에 따라 K 값이 달라지기 때문이다.

LDPC 부호는 블록 선형 부호이기 때문에, 인접한 두 부호 워드(codeword)간의 최소거리, minimum distance 라는 속성을 갖는데, 이 최소거리가 멀수록 더 많은 오류를 검출하고 수정할 수 있다.

LDPC 부호의 패리티 검사 행렬은 Tanner 그래프라는 Bipartite 타입의 그래프로 표현이 가능하다. 이때 그래프에 여러 사이클(cycle)이 존재하는데, 가장 짧은 길이의 사이클을 Girth 라고 하며 LDPC 부호의 성능을 좌우하는 중요한 속성이다. 긴 Girth를 갖는 LDPC 부호일수록 좋은 성능을 갖는다. 특히, 길이가 4인 Girth 만은 반드시 피하는 것이 좋다.

3. 다양한 부 행렬들의 임의의 조합

본 논문에서 제안하는 LDPC 부호는 무게를 각기 달리하는 아래의 행렬들의 순환행렬과 영행렬을 부행렬로 사용하여 만든 패리티 검사 행렬을 갖는 Quasi-Cyclic LDPC 부호이다.

$$O=\begin{vmatrix}0&0&0&\cdots&0\\0&0&0&\cdots&0\\0&0&0&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&0\end{vmatrix},\ C_1=\begin{vmatrix}1&0&0&\cdots&0\\0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\end{vmatrix},\ C_2=\begin{vmatrix}1&0&0&\cdots&1\\1&1&0&\cdots&0\\0&1&1&\cdots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\end{vmatrix}$$

수식 2. 부행렬들

임의의 어떤 행렬들도 부행렬로 사용될 수 있지만, 본 논문에서는 위에서 정의된 영행렬과 2개의 기본행렬을 부행렬로 사용한다. 무게가 3이상인 행렬을 부행렬로 사용하려면, 고려해야 할 사항이 있는데 이는 추후에 언급하도록 한다.

2장에서 언급한 대로 일반적인 Quasi-Cyclic LDPC 부호의 패리티 검사 행렬을 만들기 위해서는 부행렬들을 얼마만큼 shift 할 것인지를 나타내는 shift index 행렬이 필요하다. 그런데 본 논문에서 관심을 갖는 부분은 부행렬들을 얼마나 shift 할 것인가 하는 부분이 아니고, 세 개의 부행렬들을 어떤 조합으로 배치할 것인가이다. 따라서 shift 값은 특별히 고려하지 않고 열에 따라서 일정한 값을 부여한다.

부행렬들의 배치하기 위해서 부행렬들의 조합을 알려주는 lookup 테이블인 부행렬 index 행렬이 필요한데, 여기에서는 그 index 행렬을 무작위로 만드는 방법을 이용한다. 이때 반드시 지켜야 하는 조건들이 아래와 같이 세 가지 존재한다.

1) C_1과 C_2의 조합으로 그림.1 과 같이 index 행렬 내에 사각형을 구성하지 않도록 한다.

2) C_2는 index 행렬 내에서 각 열과 행에 많아야 하나 존재하도록 한다.

3) index 행렬의 각 열에는 최소한 C_1과 C_2가 하나씩 또는 세 개 이상의 C_1이 존재하도록 한다. 즉, 각 열에는 1이 세 개 이상 존재해야 한다.

$$H=\begin{pmatrix}0&2&0&0&1&1&0&0&0\\1&0&0&\textcircled{2}&0&0&\textcircled{1}&0&0\\0&1&0&\textcircled{1}&0&0&\textcircled{1}&0&1\\1&0&0&0&2&0&0&1&0\\1&0&0&0&0&1&0&0&2\end{pmatrix}$$

그림 1. 인덱스 행렬내의 사각형

$$H=\begin{pmatrix}\textcircled{1}&1&0&0&0&\textcircled{1}&0&0&0\\1&0&0&1&0&0&1&0&0\\0&1&0&0&0&0&1&0&1\\0&0&1&0&1&0&0&1&0\\\textcircled{1}&0&0&0&0&\textcircled{1}&0&0&1\end{pmatrix}$$

그림 2. 패리티 검사 행렬내의 사각형

위에서 정의한 세 가지 조건이 필요한 이유는 아래에서 언급하는데, 요약하여 말하면 1)과 2)는 LDPC 부호에서 길이가 4인 girth를 피하기 위함이고, 3)은 LDPC 부호가 유효한 부호가 되게 하기 위함이다.

1) 패리티 검사 행렬에서 길이가 4인 girth 가 존재하는지 여부를 판별하는 방법은 1이 패리티 검사 행렬에 그림.2 에 존재하는 것 같은 사각형의 존재 여부인데, 만약 C_1과 C_2의 조합으로 그림.1 과 같이 index 행렬에 사각형을 구성하게 되면 반드시 패리티 검사 행렬에 사각형이 존재하게 된다.

2) 1)에서와 마찬가지 이유로 C_2 가 하나의 행이나 열에 두 개 이상 존재하면 패리티 검사 행렬에 1로 만든 사각형이 존재하게 된다. 이와 같은 이유로 인해 무게가 3이상인 기본행렬을 사용할 경우 C_1과 C_2와 같은 형태가 아닌 적절히 치환된 형태의 행렬을 사용하여야 한다.

3) 효용성이 있는 LDPC 부호를 만들기 위해서는 패리티 검사 행렬의 각 행과 열에 아래 조건의 각각 k개와 j개의 1이 존재해야 한다. $k>j\geq 3$. 이 조건을 만족해야 블록의 길이가 길어질수록 LDPC 부호의 minimum distance도 블록의 길이에 따라서 증가하고 [2], LDPC 부호의 복호화 알고리즘으로 사용되는 Sum product algorithm을 이용하여 복호화 할 때 좋은 성능을 보여준다.

$$H=\begin{pmatrix}C_2&C_1&&&&C_1&&&&C_1\\&C_1&C_2&&C_1&&C_1&&&\\&&C_1&&C_1&C_2&C_1&&C_1&\\C_1&&&C_2&C_1&&&&&C_1\\&&C_1&&&C_1&&C_2&C_1&\\C_1&&&&C_1&&&C_1&&C_2&C_1\\&&&C_1&&&C_2&C_1&C_1&\\&&C_2&&&&&C_1&C_1&C_1\end{pmatrix}$$

수식 3. 부행렬들의 index 행렬.

수식.3 에서 무작위로 만들어진 부행렬 index 행렬의 예를 볼 수 있는데, 이 index 행렬에서 행과 열의 크기는 상호간의 의존적이다. 즉, index 행렬을 만들기 위해서 열의 길이가 주어지면, 그에 따라 행의 길이가 가능한 범위가 정해진다. 위의 예에서 index 행렬의 사이즈가 8×14 인데, 이는 행의 길이가 8 일 때, 가능한 열의 길이가 최대 14 이다. 위의 세 조건을 만족하면서 열의 길이가 15 이상인 index 행렬은 만들 수가 없다. 이를 역으로 말하면, 열의 크기가 14 가 주어진다면 행의 크기는 8 보다는 크고 14 보다 작아야 한다는 것이다. 위에서 보인 예는 각 열에는 1이 세 개이지만 각 행에는 5~6개의 1이 존재하는 Irregular LDPC를 만드는 index 행렬이다. 하지만 이

결과는 index 행렬의 각 열마다 1의 개수가 3인 경우이고, 1의 개수가 늘어나게 되면 행의 길이는 더 짧아지게 된다.

아래의 그래프에서는 각 열에서 1의 개수가 3개와 4개로 고정했을 때, 행과 열의 크기의 상호 관계에 대해서 살펴볼 수 있는데, 행의 크기가 커질수록 행과 열의 크기의 비율이 점점 커짐을 알 수 있다. 따라서 부호율도 역시 점점 커지게 된다.

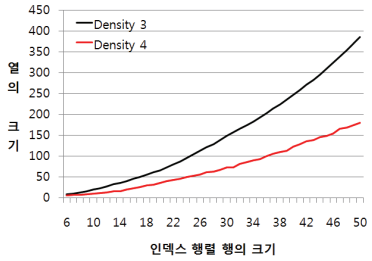

그림 3. 행과 열의 크기 관계

무작위로 만들어진 부행렬 index 행렬을 바탕으로 해서 패리티 검사 행렬을 만든다. 이때, 기본 행렬들 O, C_1 그리고 C_2의 크기 p는 3이상으로 모두 같아야 한다. p에 대한 다른 제약 조건이 존재하지 않으므로 패리티 검사 행렬의 크기를 유연하게 조절할 수 있다. 부행렬 인덱스 행렬의 크기가 $J \times L$일 때, 패리티 검사 행렬의 크기는 $Jp \times Lp$이고, 부호율은 $K/N = (Lp - Jp)/Lp = 1 - J/L$ 이다. 여기서 앞장에서의 Quasi-Cyclic LPDC 경우와는 다르게 "=" 부호를 사용한 이유는 여기서 제안한 방법으로 만들어진 패리티 검사 행렬은 full rank 이기 때문이다.

4. 모의실험 결과

이장에서는 앞장에서 제안한 Quasi-Cyclic LDPC 부호를 Binary Symmetric Channel (BSC)에서 복호화한 모의실험 결과를 다룬다. Sum Product Algorithm을 이용하여 복호화 하였고, 최대 반복 횟수는 20번이다. BSC에서 사용된 에러율은 다음과 같다.

$$\Pr(c_i = b | y_i) = \begin{cases} 1-\epsilon & when\ y_i = b \\ \epsilon & when\ y_i = b^c \end{cases}$$

수식 4. BSC에서 사용된 Cross Over 에러율

그림 4, 5, 6에서는 부호의 길이가 각각 500, 1000, 그리고 2000인 경우의 모의실험 결과를 보여준다. 부호의 길이가 각각 500, 1000, 2000 이 되도록 부행렬의 index 행렬의 크기와 기본행렬의 크기를 적절히 결정하고 BSC 의 에러율(0.01~0.1)에 맞춰 모의실험을 각각 10000 번씩 수행하였다. 복호화 성공률은 다음과 같이 정의하였다.

$$\text{복호화 성공율} = \frac{\text{에러 수정 성공 횟수}}{\text{전체 모의 실험 횟수}}$$

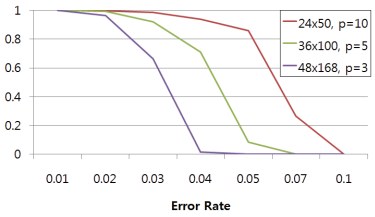

그림 4. 부호 길이 500 인 경우, 복호화 성공률과 BSC 의 에러율

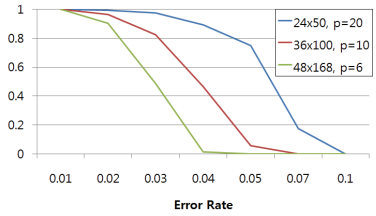

그림 5. 부호 길이 1000 인 경우 복호화 성공율

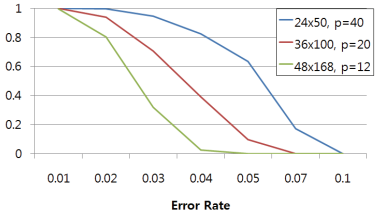

그림 6. 부호 길이 2000 인 경우 복호화 성공률

5. 결론

본 논문에서는 Quasi-Cyclic LDPC 부호의 패리티 검사 행렬을 만들기 위해서 영행렬을 포함하여 세 개의 부행렬들을 무작위로 조합하였다. 이렇게 생성된 LDPC 부호는 각 열에만 1의 개수가 고정되어 있는 일종의 Irregular LDPC 부호이다. 또한 주어진 조건을 만족시키면 길이가 6 이상인 Girth를 갖는 Tanner Graph를 가진다.

참고문헌

[1] Marc P. C. Fossorier "Quasi-Cyclic Low-Density Parity-Check Codes From Circulant Permutation Matrices" IEEE Trans. Inform. Theory, Vol. 50, No. 8, pp. 1788~1793.

[2] R. G. Gallager "Low-Density Parity-Check Codes" IRE Trans. Inform. Theory 1962. pp. 21~28.

[3] Manabu HAGIWARA and Hideki IMAI "Quantum Quasi-Cyclic LDPC Codes" Proc. ISIT 2007, France 2007.

[4] David J.C. Mackay "Good Error-Correcting Codes Based on Very Sparse Matrices" IEEE Trans. Inform. Theory, Vol. 45, No. 2, 1999, pp. 399~431.

[5] David. J.C. Mackay and Radford M. Neal "Near Shannon Limit Performance of Low Density Parity Check Codes" Electronics Letters Vol. 32, No. 18, 1996, pp. 1645~1646.
[6] 김준성, 배슬기, 정비웅, 송홍엽 "IEEE 802.16e의 LDPC 부호화 기술 분석" 대한전자공학회 텔레콤 제20권 제2호, 2004, pp. 27~33.