SPCPC에서 LDPC부호를 이용한 오류 정정

*김성만 **오태석 ***김범곤 ****송희근 *****김용철
서울시립대학교

*sungmankim@gmail.com

Error correction using LDPC Code in SPCPC

*Sungman Kim **Taesuk Oh ***Bum Gon Kim ****Hee Keun Song *****Yong Cheol Kim
University Of Seoul

요약

본 논문은 AWGN 채널상의 Single Parity Check(SPC) 다차원 product부호에서 LDPC(Low Density Parity Check)부호를 이용한 오류 정정의 성능을 제시한다. 기존 방법인 터보 부호 방식을 이용한 오류 정정과 비교하여 LDPC부호가 갖는 장점을 기술하고 실험을 통해 LDPC 부호를 이용한 오류 정정 성능도 터보부호와 대등함을 보인다.

1. 서론

일반적인 통신 시스템에서 송신 측의 데이터와 채널상의 잡음이 더하져서 수신된 데이터에 오류가 발생할 수 있다. 채널상의 잡음으로 부터 강인성을 갖기 위하여 보내고자 하는 데이터를 부호화하는 것을 채널코딩이라 한다. 채널코딩의 종류 중 하나인 SPC(Single Parity Check) 부호는 데이터 비트들에 한 개의 패리티 비트를 추가하여 코드워드를 생성하는 블록 코드이다.

Product Code는 1954년에 Elias에 의해 처음 소개 되었고 SPC를 다차원 product code로 확장한 것이 SPCPC(Single Parity Check Product Codes)이다. Rankin은 SPCPC에서 soft input, soft-output 기반의 LLR(Log likelihood ratio)을 뽑아내는 터보부호 방식의 오류 정정 알고리듬을 제안하였다.

SPCPC는 블록 코드의 일종이므로 구조적으로 LDPC(Low Density Parity Check) 부호를 이용한 오류 정정도 적용 가능하다. LDPC 부호를 이용한 오류 정정은 일반적으로 부호화와 복호화가 단순하고 직관적이며 사용되는 합 곱(Sum Product) 알고리듬은 하드웨어 구현 시에 병렬 처리 수행이 가능하므로 복호화 및 오류 정정의 수행 속도를 향상 시킬 수 있다.

2장에서는 SPCPC의 구조와 부호화 및 복호화에 대해 서술하고, 3장에서는 LDPC 부호의 구조와 특성을 서술하며, 4장에서는 SPCPC에서 LDPC를 이용한 오류 정정 알고리듬을 제안한다. 5장에서는 컴퓨터 실험에서 hard decision과 soft decision을 이용한 오류정정 알고리듬의 결과를 AWGN 채널의 BER을 통해 분석하고 6장에서 결론을 맺는다.

2. SPCPC

SPC(Single Parity Check)부호는 정보비트에 패리티 1비트를 추가하여 코드워드를 생성한다. (4, 3) SPC 부호의 예는 다음과 같다.

정보비트, b = [1 0 1] 일 때, 코드워드, c = [1 0 1 0] 이다. 각 행렬과의 부호 연산은 GF(2)를 이용한다. 생성 행렬과 복호행렬 그리고 패리티 검사 행렬은 각각 수식 (1), (2), (3)과 같이 G, D, H이다. 코드워드의 모든 비트들의 합은 0이 되어 짝수 패리티를 만족한다.

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

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (2)$$

$$H = [1\ 1\ 1\ 1] \quad (3)$$

SPC Product Codes는 SPC 부호를 다차원으로 확장한 부호이다. (4, 3) SPC 부호를 2차원으로 확장한 (16, 9) SPCPC의 예를 들어 설명한다.

정보비트 와 코드워드는 각각 b = [1 0 1 1 1 0 0 0 1], c = [1 0 1 0 1 1 0 0 0 0 1 1 0 1 0 1] 이다. 이를 4 by 4 행렬로 구성 시 수식 (4)과 같은 형태를 갖고 그림 1과 같이 패리티 비트로 인해 수평 및 수직 방향 모두 짝수 패리티가 성립한다. 그림1.의 Parity on Parity는 수직(Parity on Rows) 방향으로 또는 수평 방향(Parity on columns)으로 모두 같은 값을 갖는다.

$$C = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad (4)$$

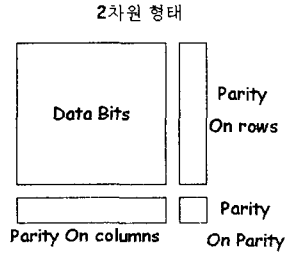

그림 1.

그림 2는 3차원의 SPCPC를 나타낸다.

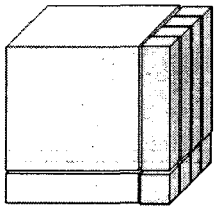

그림 2.

SPCPC는 d 차원의 하이퍼 큐브 형태로 확장되고 확장에 따른 블록 크기는

$$R=\left(\frac{n-1}{n}\right)^{d} \tag{6}$$

이고, 코드 율은

$$N = n^{d} \tag{7}$$

이며, 최소 거리는

$$d_{\min} = 2^{d} \tag{8}$$

이다.

3. LDPC 부호

1962년 Gallager에 의해서 처음 제안된 LDPC 부호는 패리티 검사 행렬(parity check matrix)의 원소들의 대부분이 0인 선형블록부호로서 터보부호와 마찬 가지로 Shannon의 채널용량 한계에 가까운 성능을 보였다.

당시의 기술력으로는 구현이 불가능한 복잡도로 인해 오랫동안 잊어져 왔고 1995년 Mackay와 Neal은 이를 재발견하였고, Gallager의 간단한 확률적(probabilistic) 복호법을 이용하여 LDPC 부호의 성능이 매우 우수함을 보였다.

터보 부호와 비교하였을 때, LDPC부호의 부호화기와 복호화기의 복잡도가 상대적으로 낮고 터보 부호는 병렬 처리 복호화기 하드웨어로 설계가 불가능하지만 LDPC부호는 가능하여 병렬 처리로 인해 수행 속도를 향상 시킬 수 있다.

LDPC부호의 기본적인 특성은 다음의 패리티 관계로부터 얻을 수 있다. v는 송신벡터 즉 코드워드이고, H^T, r 그리고 e는 각각 패리티 검사 행렬의 전치행렬, 수신벡터 그리고 오류벡터이다. 수신 벡터 r과 H^T의 곱 s를 신드롬이라 하고 수식 (9), (10), (11)을 통해 오류를 검출할 있음을 알 수 있다.

$$c\cdot H^{T}=0 \tag{9}$$

$$e = r + v \tag{10}$$

$$\begin{aligned} s &= r\cdot H^{T} \\ &= (v + e)\cdot H^{T} = v\cdot H^{T} + e\cdot H^{T} \\ &= 0 + e\cdot H^{T} \\ &= (s_1, s_2, s_3, \ldots, s_J) \end{aligned} \tag{11}$$

4. SPCPC에서의 LDPC부호를 이용한 오류정정

LDPC부호의 오류 정정 과정을 Hard decision 복호화방식인 Bit-flipping알고리듬과 Soft decision 복호화방식인 Sum-product 알고리듬을 2차원 SPCPC에 적용하는 예를 통해 설명한다.

1) Bit-flipping알고리듬

코드워드가 c = [1 0 1 0 1 1 0 0 0 0 1 1 0 1 0 1]이고 잡음으로 인해 수신벡터 r = [1 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1]의 3번째 비트에 오류가 있다고 가정한다.

$$c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{12}$$

$$r\cdot H^{T} = [1\ 0\ 0\ 0\ 0\ 0\ 1\ 0] \tag{13}$$

$$H = \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\ 0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \end{bmatrix} \tag{14}$$

$$\begin{aligned} y1 &= x1 \oplus x2 \oplus x3 \oplus x4 \\ y2 &= x5 \oplus x6 \oplus x7 \oplus x8 \\ y3 &= x8 \oplus x9 \oplus x11 \oplus x12 \\ y4 &= x13 \oplus x14 \oplus x15 \oplus x16 \\ y5 &= x1 \oplus x5 \oplus x9 \oplus x13 \\ y6 &= x2 \oplus x6 \oplus x10 \oplus x14 \\ y7 &= x3 \oplus x7 \oplus x11 \oplus x15 \\ y8 &= x4 \oplus x8 \oplus x12 \oplus x16 \end{aligned} \tag{15}$$

수신벡터 r을 수식 (12)와 같이 4 by 4 비트 패턴으로 구성 시 수평 및 수직 패리티가 성립하지 않는 1행과 3열을 통해 3번째 비트에 오류가 발생함을 알 수 있다. 이는 식 (13)의 신드롬을 통해 검사하면 1번째와 7번째 비트에서 1, 즉 패리티 실패인 것을 알 수 있다.

이는 곧 식 (14)의 H가 1번째와 7번째 행의 검사행이 오류를 검출한 것임을 알 수 있다. H를 식(15)의 패리티 검사 식으로 변환하면, 3번째 비트만이 유일하게 y1과 y7의 두 검사 식 모두에 포함 되어 있음을 알 수 있다. 즉 실패한 패리티 검사 식을 가장 많이 갖는 3번째 비트를 정정한다.

BF복호화 알고리듬은 다음과 같다.

1단계. 패리티 검사 합 (parity-check sums) 계산을 통해 신드롬을 얻는다. 신드롬이 0이면 복호화를 중지한다.
2단계. 각 비트별로 실패한 패리티 검사 식의 개수를 구하고 각 비트별 식의 개수를 fi로 나타낸다.
이때 i=0,1, ... ,n-1이다.
3단계. fi가 가장 큰 비트들을 찾아서 비트 값을 변경시킨다.
4단계. 패리티 검사 합이 모두 0이 아니고 최대반복 회수 이하이면 2단계를 다시 수행한다.

SPCPC(4, 3)부호의 각 차원별 BF복호화 알고리듬의 최대 오류정정가능 비트는 1, 4, 16비트이고 최대 반복회수는 1회, 3회, 5회가 된다.

2) Sum-Product알고리듬

그림 3과 같이 패리티 검사 행렬, H로 부터 Bipartite graph를 생성해 낼 수 있다. H의 행과 열이 각각 검사 노드와 비트 노드가 된다.

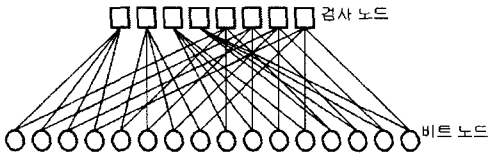

그림 3.

본 graph에서 검사노드와 비트 노드는 Bayesian Network를 형성하여 비트 노드의 비트는 검사노드를 형성한다. 즉 모든 비트 노드는 체크 노드의 부모노드이고 각 체크 노드는 비트 노드들의 자식 노드이다.

Sum-Product알고리듬은 다음과 같다.

1단계. 변수 노드의 초기화

Baye's rule에 따라 식 (16)과 식 (17)로 부터 AWGN 확률밀도함수에 의해 초기화를 얻을 수 있다. 이때, w는 w=1-2x(BPSK 변조시스템의 출력결과)이고 y는 정합필터출력의 샘플링된 결과이다. 또한 AWGN 확률밀도함수를 이용하는 식 (18), (19), (20), (21)을 통해 을 도출해 낼 수 있다.

$$p(y \mid w) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp^{\left(-\frac{(y-w)^2}{2\sigma^2}\right)} \tag{16}$$

$$q_0 = \log \frac{p(x=0 \mid y)}{p(x=1 \mid y)} = \log \frac{\frac{1}{(2\pi\sigma^2)^{1/2}} \exp^{\left(-\frac{(y-1)^2}{2\sigma^2}\right)}}{\frac{1}{(2\pi\sigma^2)^{1/2}} \exp^{\left(-\frac{(y+1)^2}{2\sigma^2}\right)}} = \frac{2}{\sigma^2} y \tag{17}$$

$$\log \frac{f_x^0}{f_x^1} = \frac{2}{\sigma^2} y \tag{18}$$

$$f_x^0 + f_x^1 = 1 \tag{19}$$

$$f_n^1 = 1/(1 + \exp(2ay_n/\sigma^2)) \tag{20}$$

$$f_n^0 = 1 - f_n^1 \tag{21}$$

2 단계. 검사 노드에서 비트 노드로의 메시지 전달

q^x_{ij}는 i를 제외한 검사노드로부터 비트가 x값을 갖을 확률이다. r^x_{ij}는 수신벡터 r의 값이 x일 때, 검사 식을 만족할 확률이다. 검사 노드에서 모든 입력 정보 q^x_{ij}를 저장하고 검사 노드에 연결된 다른 비트들에 기초하여 r^x_{ij}를 생성한다. 데이터 비트가 0이면 r^0_{ij}를 만족하고 1이면 r^0_{ij}을 만족한다. 각 비트 노드에 연결된 체크 노드로부터 확률 정보를 모아서 q^x_{ij}값을 생성한다.

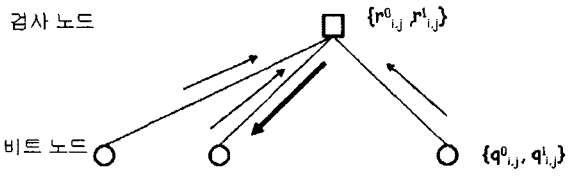

그림 4.

3 단계. 비트 노드에서 체크 노드로의 메시지 전달

p^0_j ,p^1_j는 각 비트 당 현재 사후 확률 추정치, APP(a posterior probability)로써 두 번째 부터의 반복을 위한 값으로 사용된다.

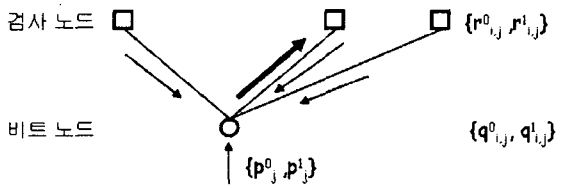

그림 5.

4 단계. 경 판정

각 비트의 APP값을 식(22)으로 경 판정하여 복호된 입력 벡터를 코드워드로 사용하여 H와의 곱으로 0을 확인한다. 패리티 실패가 있는 경우 2단계로 이동하여 다시 수행하게 된다.

$$m=1 \text{ if } q1n>0.5 \tag{22}$$

5. 실험 결과

Eb/N0의 범위가 -2에서 5(dB)까지의 범위를 갖는 AWGN(Additive White Gaussian Noise) 채널에서 BPSK 변조 신호를 이용하여 BER을 측정하였다.

실험 대상 부호 및 복호화 알고리듬은 Uncoded BPSK와 2,3,4차원 SPCPC부호에 대한 hard decision 복호화 그리고 2,3차원 SPCPC 부호에 대한 soft decision 복호화이고 이때 soft decision 복호화의 최대 반복회수는 30회이다. 각 차원별 부호율은 표1과 같다.

차원	2	3	4
부호율	9/16	27/64	81/256

표 1.

SPCPC에서의 LDPC hard decision과 soft decision 복호화에 대한 BER은 그림 6과 같다.

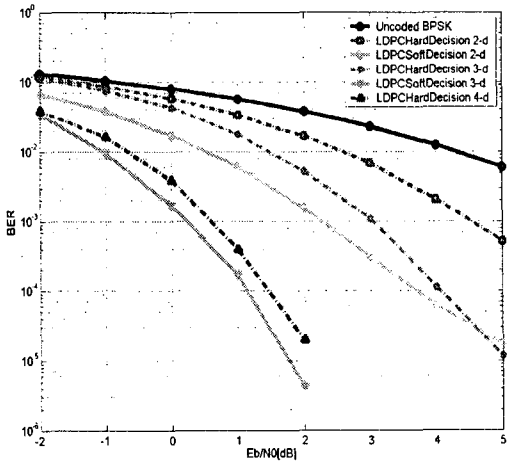

그림 6.

6. 결론

BER이 10^{-5}일때, LDPC부호는 터보부호 방식보다 요구되는 Eb/N0가 각각 2차원에서는 0.9dB높고 3차원에서는 2.2dB낮음을 알 수 있다.

차원	터보부호	LDPC부호
2	4.6dB	5.3dB
3	3.8dB	1.6dB

표2.

표2와 같이 SPCPC에서 LDPC부호는 터보 부호와 대등한 오류 정정 성능을 발휘한다. 그 외에도 LDPC부호는 부호화기와 복호화기의 복잡도가 낮은 직관적인 설계가 가능하고 병렬회로로 설계할 경우 수행 속도를 향상시킬 수 있다는 장점이 있다.

참고 문헌

[1] R.G. Gallager, Information Theory and Reliable Communication, John Wiley, New York, 1968.

[2] P. elias, "Error free coding," IRE Trans. Inform. Theory, vol IT-4, pp. 29-37, sept. 1954

[3] D.Rankin and T.Guliver, "Single parity check product codes." IEEE Trans. Commun., vol 49, pp. 1354-1362, aug. 2001.

[4] R.G Gallager, "Low Density Parity Check Codes," MIT Press, Cambridge, 1963

[5] MacKay, D.J.C., "Good error-correcting codes based on very sparse matrices" Information Theory, IEEE Trans. Vol 45 Issue: 2, Page(s): 399 -431, March 1999

[6] Lin, Error Control Coding 2nd Ed, chapter 17, Pearson Prentice Hall,