

# LDPC 부호의 적응적 반복 복호수 설정 방식 및 복호기 구조

\*이정훈, \*\*장진수, \*\*\*정영일, \*이문호

\* 전북대학교 정보통신 공학과, 정보통신 연구소

\*\* 전북대학교 컴퓨터 공학과, \*\*\* 전주공업대학 컴퓨터 정보과

## Adaptive Iteration Decoding Preset Method of LDPC Codes by SNR Estimation & Decoder Structure

\* Jeong Hun Lee, \*\* Jin Su Chang, \*\*\* Young Il Jeong, \* Moon Ho Lee

\* Dept. of Inform. & Comm. Eng. Institute of Inform. & Comm. Chonbuk University

\*\* Dept. of Computer Engineering Chonbuk University

\*\*\* Dept. of Computer Engineering, Chonju Technical Collage

E-Mail : moonho@moak.chonbuk.ac.kr

**요약** - 열악한 전송 환경에서 고품질, 고신뢰성 통신을 지속적으로 하기 위해서 오류 정정 부호는 필수적이다. 최근에 반복 복호를 통해 사선의 채널 용량 한계에 근접하는 터보 부호와 LDPC 부호가 가장 관심을 불러일으키고 있다. 반복 복호법은 성능 면에서는 우수해 지나 이에 따른 계산량 증가와 지연이 수반된다. 따라서 본 논문에서는 모의 실험을 통한 수신 데이터를 이용, SNR을 추정하여 LDPC 부호의 최대 반복 복호수에 따른 계산량과 지연을 효과적으로 줄일 수 있는 적응적 반복 복호수 설정 방식을 제안한다.

### 1. 서론

오류 정정 코드(ECCs)는 데이터를 부호화 시켜 전송하고, 전송된 데이터를 복호화 알고리즘을 통해 복호해 오류를 정정함으로써 오류 없는 통신을 가능하게 한다. 최근 가장 관심을 가지는 ECCs로서는 IMT-2000 표준으로 채택된 터보코드를 들 수가 있다. 그러나 터보코드는 반복 복호시 계산량 증가와 지연의 문제가 발생한다. 이에 대한 해결책으로 LDPC 코드가 거론되고 있는데, LDPC 코드는 1962년 Gallager에 의해 처음 제안되어 패리티 검사행렬 H의 원소들의 대부분이 0인 선형 블록 부호로서 당시의 기술로 구현이 불가능한 복호의 복잡도로 인해 오랫동안 잊혀져왔다. [1] 그러나 1995년 Mackay와 Neal은 이를 재발견하고, Gallager의

간단한 확률적 복호법을 이용하여 성능이 매우 우수함을 보였다. [2][3] LDPC 부호는 터보 부호에 비해 정정되지 않는 오류들을 대부분 검출할 수 있고, 복호화의 복잡도가 낮을 뿐 아니라 좋은 거리 특성으로 오류마루 현상이 나타나지 않고, 완전 병렬 처리로 고속 처리가 가능한 장점이 있다. 반면에 부호화 부분의 높은 복잡도가 LDPC코드의 중요한 문제점이었으나 최근에 삼각행렬 분해법등 부호화 방법이 제안되고 있다. LDPC 부호의 복호화 부분에서는 터보코드와 마찬가지로 반복 복호 기법을 사용하고 있다. 본 논문에서는 LDPC 부호의 복호기 구조를 보이고 모의 실험을 통한 수신 데이터를 이용, SNR을 추정하여 LDPC 부호의 최대 반복 복호수에 따른 계산량과 지연을 효과적으로 줄일 수 있는 적응적 반복 복호수 설정 방식을 제안한다.

본 논문의 구성은 2 절에서는 LDPC 부호의 특성, 부호화 복호화 방법과 LDPC 복호기의 하드웨어 구조를 살펴보고, 3절에서는 제안된 SNR 추정에 의한 적응적 반복 복호수 설정 방식에 대해 설명하고, 4절에서는 모의 실험을 통한 결과를 보이고, 5절에서 본 논문의 결론을 맺는다.

### 2. LDPC(Low Density Parity Check Code

LDPC 부호는 매우 sparse 한 랜덤 패리티 검사행렬 H에 의해 정의된다. 패리티 검사 행렬 H의 구조적 특징은 다음과 같다. 1) 각 행(row)은 무게 k의 1로 구

성하며(행의 무게 k는 가능한 균일하게 구성한다.), 2) 각 열(column)은 무게 j개의 1이 되게 구성하며, 임의의 두 열사이의 내적은 1보다 크지 않게 랜덤 하게 구성한다. (열의 무게는 1의 수) 따라서 부호길이에 비해 열과 행의 무게가 매우 작다. 이렇게 패리티 검사 행렬 H에 의해 구성되어 지므로 LDPC (Low Density Parity Check) 부호라 한다. LDPC 부호의 패리티 검사행렬은 1의 개수가 매우 적기 때문에 매우 큰 블록크기에서도 반복 복호를 통하여 복호가 가능하며 블록크기가 매우 커지면 터보 부호처럼 Shannon의 채널용량 한계에 근접하는 성능을 보인다.

### (a) 부호화

부호화에서는 길이 k인 2진 데이터 벡터가 주어지고 길이 n인 벡터를 전송한다. 이때 부호율(R)은 k/n이 된다. 여기서 m=n-k가 parity check bit가 된다. 전송되어서는 벡터 t는 생성 벡터 G의 전치 행렬과 소스 벡터의 곱으로 구할 수 있다. parity check 행렬(P)과 생성행렬을 표현하면 다음과 같다.[2]

$$H = [P | I_m] \quad (I_m : m \times m \text{의 항등행렬}) \quad (1)$$

$$G^T = \begin{bmatrix} I_k \\ P \end{bmatrix} \quad (I_k : k \times k \text{의 항등행렬}) \quad (2)$$

### (b) 복호화

LDPC 부호의 복호를 위해 사용되는 알고리즘은 Sum-product 알고리즘이다. [3] sum-product 복호기는 한적인 거리 복호기(Bounded distance decoder)보다 성이 우수하며, 채널로부터 얻는 likelihood를 이용할 있는 연관성 복호기이다. 연결 오류가 발생하는 채널인 경우에도 연결 오류를 정정할 수 있도록 일반화될 수 있다. sum-product 알고리즘을 설명하기 위한 패리티 검사 행렬 표기와 Bipartite 그래프는 그림 2와 같다.[2]

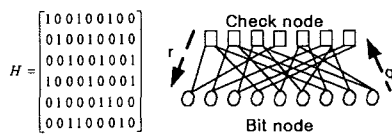


그림1 Parity Check Matrix 와 Bipartite Gr

그래프 상에서 위쪽 사각형 노드를 체크 노드, 아래쪽 원노드를 비트 노드라고 하다. 체크 노드는  $r_{ij}$  를 생성하고, 만약 data 비트가 0이라면  $r_{ij}^0$  를 만족하고 data bit가 1이라면  $r_{ij}^1$  를 만족한다. 이러한 확률은 (3), (4)를 통해 구해진다.

$$r_{i,j}^0 = \frac{1}{2} [1 + \prod_{i \in \text{row}[j] \setminus i} (q_{i,j}^0 - q_{i,j}^1)] \quad (3)$$

$$r_{i,j}^1 = \frac{1}{2} [1 - \prod_{i \in \text{row}[j] \setminus i} (q_{i,j}^0 - q_{i,j}^1)] \quad (4)$$

이때  $i \in \text{row}[j] / i$  는 단지 행 j에서 1을 갖는 모든 비트의 인덱스를 나타내는 것이고 현재 비트 포함하지 않는다. 식 (3), (4)는 그림 2와 같이 표현된다.

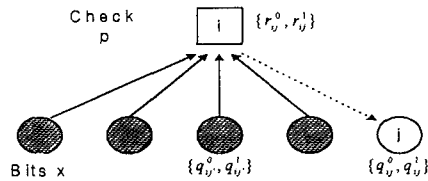


그림 2 체크 노드

비트 노드는  $q_{ij}$  값을 생성하고, 체크 노드와 마찬가지로  $q_{ij}^0, q_{ij}^1$  값을 식 (5), (6)를 통해 구한다. 식 (5) (6)는 그림 3과 같이 표현된다.

$$q_{i,j}^0 = a_{ij} p_i^0 \prod_{j \in \text{col}[i] \setminus j} r_{ij}^0 \quad (5)$$

$$q_{i,j}^1 = a_{ij} p_i^1 \prod_{j \in \text{col}[i] \setminus j} r_{ij}^1 \quad (6)$$

$p_i^0, p_i^1$ 은 각각의 비트 당 현재 사후 확률 추정치 이고,  $j \in \text{col}[i] / j$  는 단지 열 i에서 1을 갖는 모든 체크의 인덱스를 나타내는 것이고 현재 비트 j는 포함하지 않는다. 비트 노드는 그림 3과 같이 구할 수 있다.

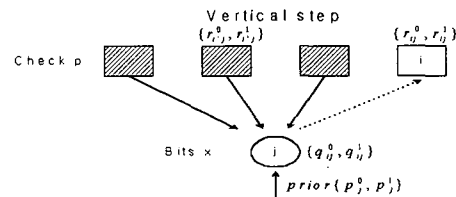


그림 3 비트 노드

끝으로 각 비트의 APP(a Posterior Probability) 값을 정 하고 복호 된 입력 벡터는 패리티 검사행렬 H와 검사된다. 만약 H와 복호 된 입력 벡터와의 곱이 0이라면 최종 출력이 복호 된 입력 벡터가 되고 그렇지 않으면 각각을 다시 계산한다.

### (c) LDPC 부호의 복호기 구조

앞 절에서의 LDPC 부호화 과정을 통해 LDPC 복호기를 설계할 수가 있다. [5][7] LDPC 복호기는 그림 4에서

는 바와 같이 병렬식으로 구성된 네 개의 PU(Processor Unit)과 세 개의 내부 연결망으로 구성된다.

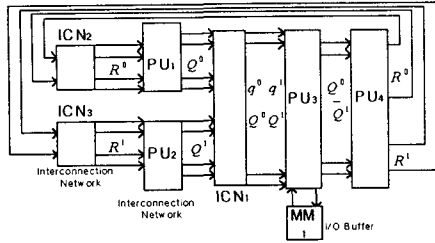


그림 4 LDPC 복호기 구조

PU3은  $Q^0, Q^1$ 의 매트릭스 사이의 차이를 구해준다. 그림 5는 PU1과 PU2의 신호흐름과 앞에서 살펴 본 비트 노드에서 q값을 구하는 과정을 나타낸다.

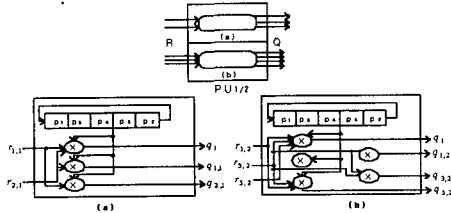


그림 5 비트 노드

(a)는 열의 무게가 2일 때 (b)는 열의 무게가 3일 때를 각각 나타낸다. 그림 6을 통해 PU4에서의 신호의 흐름 알 수가 있다. 이 과정을 통해 체크 노드로부터 r 값을 구 할 수 있다.

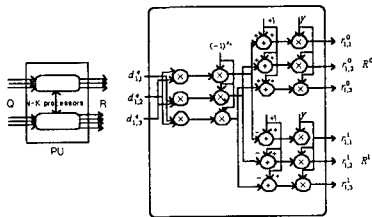


그림 6 체크 노드

### 3. SNR 추정에 의한 반복 복호수 설정

정보 블록의 길이가 각각 256, 512 부호율(R) 이 1/인 LDPC 부호의 SNR에 따른 평균 반복 복호수를 그림 3에 나타낸다. 모의실험에서는 어느 유효한 부호율에 도달할 때까지 반복적으로 복호 동작을 수행한다. 그림 7에서 알 수 있듯이 낮은 Eb/No에서는 미리 지정한 최대 반복 복호수를 모두 실행하지만 높은 Eb/No로

갈수록 적은 반복 복호에서 반복 복호를 멈춘다. 또한 3dB이상의 높은 Eb/No에서는 최대 반복 복호수를 가변하여도 거의 같은 반복 복호수로 수렴함을 알 수 있다. 따라서 필요 이상의 반복 복호를 할 경우에는 계산량의 증가와 지연이 되므로 실험을 통해 반복 복호수를 적응적으로 설정할 수가 있다.

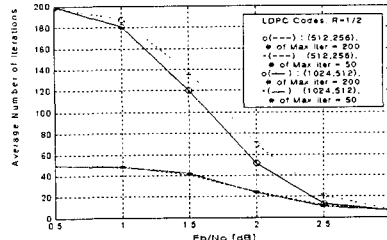


그림 7 각 부호율에서 평균 반복 복호수

제한한 SNR 추정에 의한 적당한 최대 반복수를 설정하기 위해서 수신 데이터 값을 이용, SNR을 추정 한 후 실험을 통해 이미 얻어진 결과를 look up table에 저장하고, 이를 이용하여 요구하는 성능을 만족하는 최대 반복 복호수를 가변적으로 미리 설정해 놓는다. 그러므로 불필요한 반복 복호수를 줄여 지연을 효과적으로 감소 시킬 수 있다. 제안한 구조의 블록도는 그림4와 같다.

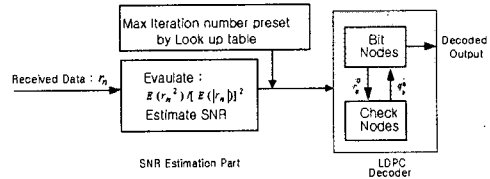


그림 8 적응적 반복 복호수 설정 방식

#### (a) SNR 추정

LDPC 부호는 AWGN 채널에서 최소 에러를 만들어 내는 채널상태 정보에 의존한다. 수신된 데이터가 복호기 입력 전에 채널상태 값을 정확히 알 수 있다면 복호기 입력 전에 요구하는 성능을 만족하도록 최대 반복 복호수를 미리 설정하여 계산 복잡도를 줄일 수 있을 것이다. [6] SNR을 추정하는 방법은 다음과 같다. 부호 비트를 전송하기 위해 AWGN 채널에서 BPSK (binary shift keying) 변조를 가정하면 수신된 데이터는 식(7) 같이 표현할 수 있다.

$$r_n = \pm\sqrt{E_s} + n_n = \pm\mu + n_n \quad (7)$$

$n_n$ 은 평균이 0이며 분산은  $\sigma^2 = N_0/2$ 인 가우시안 잡음을 나타낸다. 심볼 에너지와 비트당 에너지의 관계는

$E_s = E_b R$  과 같으며, R은 부호율을 나타낸다. SNR 을 추정하기 위해 식(7)의 제곱의 평균과 절대값의 평균을 고려하면 다음과 같다

$$E(r_n^2) = E_s + \sigma^2 \quad (8)$$

$$E(|r_n|) = \sigma \sqrt{\frac{2}{\pi}} e^{-(E_s/2\sigma^2)} + \sqrt{E_s} \operatorname{erf}\left(\sqrt{\frac{E_s}{2\sigma^2}}\right) \quad (9)$$

식(8)와 식(9)의 관계를 표현하면

$$\begin{aligned} \frac{E(r_n^2)}{[E(|r_n|)]^2} &= \frac{1 + \frac{E_s}{\sigma^2}}{\left\{ \sqrt{\frac{2}{\pi}} e^{-(E_s/2\sigma^2)} + \sqrt{\frac{E_s}{\sigma^2}} \operatorname{erf}\left(\sqrt{\frac{E_s}{2\sigma^2}}\right) \right\}^2} \\ &= f\left(\frac{E_s}{\sigma^2}\right) = f(\beta) \end{aligned} \quad (10)$$

식 (10) 에서  $\beta = \mu^2/\sigma^2 = 2E_s/N$ 를 나타내며 식 (10) 의 평균값의 비를 변수 z 라 정의하면 다음 식과 같다.

$$z = E(r_n^2)/[E(|r_n|)]^2 \quad (11)$$

즉 식 (10)과 (11)를 이용하여  $\beta$  를 알 수 있으며 SNR을 추정할 수 있게 된다. 위 식에서 식(12)과 같이 간략화 하여  $\beta$ 를 구할 수 있다.

$$\beta \approx -34.0516z^2 + 65.9548z - 23.6184 \quad (12)$$

#### 4. 모의 실험

제안한 SNR 추정에 의한 적응적 최대 반복 복호수 감소 방식을 그림 8에 따라서 모의 실험하였다. LDPC 부호의 블록 크기는 256, 부호율(R)은 1/2로 하였다. 그림 9에 제안한 구조를 적용한 SNR에 따른 BER 성능을 보여주고 있다. 적은 반복 복호수에서 기존 구조와 비교하여 거의 같은 성능을 나타냄을 알 수 있다. 표 1은 실제 SNR과 제안 방식에 의한 SNR 추정 값을 비교하고 제안된 방법에 의한 반복 복호수를 나타낸다.

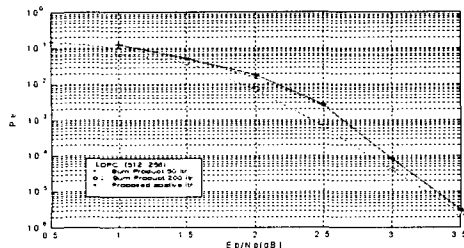


그림 9 SNR에 따른 제안 구조의 BER 성능

표 1 실제 SNR , 추정된 SNR & 평균 반복 복호수

실제 SNR	추정된 SNR	기존 평균 반복 복호수 (최대 50)	기존 평균 반복 복호수 (최대 200)	제안한 최대 반복 복호수
0.5	0.7852	49,8540	198,589	13,1020
1.0	1.3428	48,5120	187,6340	15,1040
1.5	1.8712	40,5230	138,0660	16,0297
2.0	2.1148	24,8780	69,2190	14,0070
2.5	2.5145	11,1510	20,4490	9,3415
3.0	3.0562	6,5497	6,9867	6,4824

#### 5. 결론

본 논문에서는 LDPC (Low Density Parity Check Codes) 부호의 복호기 구조를 보이고 수신데이터를 이용하여 SNR(Signal to Noise Ratio)을 추정하고 반복 복호수 적응적으로 결정하는 방식을 제안하였다. 제안된 방안은 SNR을 추정한 후 모의실험을 통해 이미 얻어진 결과를 look up table 에 저장하고 이를 이용하여 요구하는 성능을 만족하는 최대 반복 복호수를 가변적으로 설정함으로써 불필요한 반복 복호를 없애 전체적으로 평균 반복 복호수를 줄여서 계산량과 지연을 효율적으로 감소시킬 수 있다.

#### [참고 문헌]

- [1] R.G. Gallager, "Low-density parity-check codes *Trans. Inform. Theory*, vol. IT-8, pp21-28, Jan.
- [2] D.J.C. Mackay, "Good error-correcting codes has very sparse matrices, " *IEEE Trans. on Inform Theory*, vol. 45, pp.399-431, Mar.1999.
- [3] David J.C. MacKay 외 2인, "Comparison of Constructions of Irregular Codes," *IEEE Trans. on Commun*, Vol.47, No 10, Oct 1999.
- [4] Zining Wu, Coding and Iterative Detection for Multic Recoding Channels, Kluwer Academic Publishers 2000
- [5] Vladislav Dorokine, Frank R. Kschischang " Gall code for CDMA Applications-Part II " *IEEE Transactions on Commun*, vol. 48, No 11, NOVEMBER 2000
- [6] Soonyoung Kim, Jinsu Chang, Moon Ho Lee, "A Low Delay Stop Criterion of Turbo Decoding, *2nd International Symposium On Turbo Codes*, Sep. 4~2000, Brest, France.
- [7] 이문호, 갈루이스필드·리드솔로몬·비터비·터보호기의 설계, 도서출판 영일, 2000.