

# LDPC 부호의 효율적인 복호 방법에 관한 연구

준희원 김 준 성\*, 정희원 신 민 호\*, 종신회원 송 흥 업\*

## An Efficient Decoding Algorithm of LDPC codes

Joon-Sung Kim\*, Min-Ho Shin\*, Hong-Yeop Song\* *Regular Members*

### 요 약

본 논문에서는 LDPC 부호의 복호 알고리즘의 하나인 Normalized-BP 알고리즘의 각각의 비트에 업데이트되는 값에 대한 신뢰도에 따라 복호 알고리즘에 필요한 표준화 인수값을 달리함으로써 성능을 향상시킬 수 있는 복호 알고리즘을 제안하였다. 제안된 복호 알고리즘은 기존의 Normalized-BP 알고리즘에 비하여 복잡도를 거의 증가시키지 않으면서 LLR-BP 알고리즘의 BER 성능에 약 0.02dB 정도까지 근접하는 성능을 나타냄을 컴퓨터 모의 실험을 통하여 확인하였다.

Key Words : LDPC codes; Normalized-BP algorithm; normalization factor

### ABSTRACT

In this paper, we propose a modified Normalized-BP algorithm by changing the normalization factor according to the reliability of updated messages. Proposed algorithm has almost same decoding complexity as Normalized-BP algorithm and achieves a bit-error probability of  $10^{-4}$  within 0.02dB away from compared with LLR-BP algorithm.

### 1. 서 론

1962년 Gallager에 의해 처음 제안된 LDPC 부호(low density parity check codes)는 1990대 중반에 Mackay와 Neal에 의해 재발견된 이래 Shannon의 이론적 채널 용량 한계에 근접하는 성능을 가진 부호로서 많은 연구가 진행되어 왔다<sup>[1][2][3]</sup>. LDPC 부호는 패리티 검사 행렬 H의 원소들이 대부분 '0'인 선형 블록 부호(linear block code)로서, 간단한 패리티 검사식들을 이용하여 확률적인 반복 복호 방법을 사용함으로써 성능의 향상을 가져온다. LDPC 부호는 Shannon의 채널 용량 한계에 거의 근접하는 성능을 보이기 때문에 BER  $10^{-6}$  이하의 고품질의 신뢰도를 요구하는 4세대 이동통신 시스템에 적합한 부호화 방식이다. 현재 LDPC 부호의

용량에 대한 분석과 임의 생성 방법이 아닌 유한체 GF(q) 상에서 정의된 설계 방법에 대한 연구와 복호의 복잡도를 줄이는 여러 가지 방법에 대한 연구가 활발히 진행되고 있다<sup>[4][5][6][7][8]</sup>.

Chen과 Fossorier는 AWGN 채널 환경에서 반복 복호의 복잡도를 줄이면서 기존의 복호 방법인 BP(Belief Propagation) 알고리즘에 비하여 적은 성능저하를 보이는 UMP-BP 알고리즘과 Normalized-BP 알고리즘을 제안하였다<sup>[8]</sup>. Normalized-BP 알고리즘은 UMP-BP 알고리즘에 비하여 복잡도는 거의 증가시키지 않으면서 BER  $10^{-4}$ 에서 BP 알고리즘의 성능에 0.1dB 정도의 차이로 근접하는 성능을 나타낸다. 본 논문에서는 LDPC 부호의 복호 알고리즘으로 LLR-BP 알고리즘과 UMP-BP 알고리즘, Normalized-BP 알고리즘에 대하여 소개하고, 기존

\* 연세대학교 전기전자공학과 부호 및 정보이론 연구실(js.kim, mh.shin, hy.song}@coding.yonsei.ac.kr)

논문번호 030584-1231, 접수일자 2004년 1월 2일

※본 연구는 삼성종합기술원의 "4G wireless system"의 연구 개발 과제의 지원에 의해 이루어졌음.

의 Normalized-BP 알고리즘의 반복 복호 과정을 분석하여 성능을 향상시키기 위한 방법을 제안한다 또한 컴퓨터 모의 실험을 통하여 기존의 방법과의 성능을 비교한다.

## II. 복호화 방법

### 1. 채널 모델

본 논문에서는 BPSK 변조 방식을 가정한다. LDPC 부호로 부호화된 부호어  $c = (c_1, c_2, \dots, c_N)$ 이라 할 때 송신되는 부호열  $s = (s_1, s_2, \dots, s_N)$ 이며 여기서  $s_n = 2c_n - 1, n = 1, \dots, N$ 이다.  $s$ 가 AWGN 채널 환경에서 송신된다고 하면 수신된 신호는  $y_n = s_n + v_n$ 이 되며 여기서  $v_n$ 은 mean이 0이고 variance가  $\frac{N}{2}$ 인 가우시안 랜덤 변수이다.

### 2. LLR-BP 알고리즘

Log-Likelihood Ratio-Belief Propagation(LLR-BP) 알고리즘에서는 Gallager가 제안한 "probabilistic decoding"에서처럼 실제의 확률을 사용하는 대신에 log-likelihood ratio를 사용한다는 점을 제외한다면 사실상 동일한 알고리즘이라고 볼 수 있다 [8] LLR-BP 알고리즘에 쓰이는 파라미터를 다음과 같이 정의하였다.

- $F_n$  : 수신된  $y_n$ 으로부터 얻어지는 비트 n의 log-likelihood ratio.

$$F_n = \ln \frac{p(c_n = 1 | y_n)}{p(c_n = 0 | y_n)} = \frac{4}{N_0} y_n \quad (2-1)$$

- $L_{mn}$  : 패리티 검사 방정식 m에서 비트 n으로 가는 비트 n의 log-likelihood ratio
- $z_{mn}$  : 비트 n에서 패리티 검사 방정식 m으로 가는 비트 n의 log-likelihood ratio
- $z_n$  : 매번 반복 복호시에 계산되는 비트 n의 a posteriori log-likelihood ratio
- $N(m) = \{n : H_{mn} = 1\}$  : 패리티 검사 방정식 m에 관여하는 비트의 집합
- $M(n) = \{m : H_{mn} = 1\}$  : 비트 n에 관여하는 패리티 검사 방정식의 집합

- $N(m) \setminus n$  : 비트 n을 제외한  $N(m)$
- $M(n) \setminus m$  : 패리티 검사 방정식 m을 제외한  $M(n)$

LLR-BP 알고리즘의 순서는 다음과 같다

#### 1) 초기화

각각의 m, n에 대하여

$$z_{mn} = F_n \quad (2-2)$$

#### 2) 행 방향 반복 복호

각각의 m, n에 대하여

$$T_{mn} = \prod_{n' \in N(m) \setminus n} \frac{1 - \exp(z_{mn'})}{1 + \exp(z_{mn'})} \quad (2-3)$$

$$L_{mn} = \ln \frac{1 - T_{mn}}{1 + T_{mn}} \quad (2-4)$$

#### 3) 열 방향 반복 복호

각각의 m, n에 대하여

$$z_{mn} = F_n + \sum_{m' \in M(n) \setminus m} L_{m'n} \quad (2-5)$$

$$z_n = F_n + \sum_{m \in M(n)} L_{mn} \quad (2-6)$$

#### 4) 잠정 복호

$$\hat{c} = [\hat{c}_n], \quad \begin{cases} \hat{c}_n = 1 & \text{if } z_n > 0 \\ \hat{c}_n = 0 & \text{if } z_n < 0 \end{cases} \quad (2-7)$$

- $H\hat{c} = 0 \Rightarrow \hat{c}$ 를 부호어로 결정
- $H\hat{c} \neq 0 \Rightarrow$  2)번으로 돌아가 반복 복호 수행
- 정해진 반복 복호 숫자까지 결과가 나오지 않을 때 복호 실패 선언

### 3. UMP-BP 알고리즘

LLR-BP 알고리즘은 반복 복호시 2)의 수식 계산을 수행함에 있어 높은 복호 복잡도를 요구한다. Uniformly Most Probable (UMP) BP 알고리즘은 LLR-BP 알고리즘의 복잡도를 줄이기 위한 방법으로 제안되었다[7]. LLR-BP 알고리즘에 비해 약 1dB 정도의 성능 저하를 보이지만 복잡도가 상당히 줄어들게 되며 채널 추정을 필요로 하지 않는다는 장

접이 있다

- 1) 초기화  
각각의 m, n에 대하여

$$z_{mn} = y_n \quad (2-8)$$

- 2) 행 방향 반복 복호  
각각의 m, n에 대하여

$$\sigma_{mn} = \begin{cases} 1, & \text{if } z_{mn} > 0 \\ 0, & \text{if } z_{mn} \leq 0 \end{cases} \quad (2-9)$$

$$\sigma_m = \sum_{n \in N(m)} \sigma_{mn} \pmod{2} \quad (2-10)$$

$$L_{mn} = (-1)^{(\sigma_m \oplus \sigma_{mn})} \min_{n' \in N(m) \setminus n} |z_{mn'}| \quad (2-11)$$

- 3) 열 방향 반복 복호  
각각의 m, n에 대하여

$$z_{mn} = F_n + \sum_{m' \in M(n) \setminus m} L_{m'n} \quad (2-12)$$

$$z_n = F_n + \sum_{m \in M(n)} L_{mn} \quad (2-13)$$

- 4) 잡정 복호  
LLR-BP에서의 방법과 동일하다.

#### 4. Normalized-BP 알고리즘

Normalized-BP 알고리즘은 LLR-BP 알고리즘과 UMP-BP 알고리즘의 차이점을 분석함으로써 UMP-BP 알고리즘의 성능을 향상시키기 위하여 제안되었다<sup>8)</sup>. LLR-BP 알고리즘과 UMP-BP 알고리즘의 차이는  $L_{mn}$ 을 구하는 방법에 있다.  $L_{mn\_LLR}$ 와  $L_{mn\_UMP}$ 을 비교해 보면 둘의 부호는 같고 크기는 항상  $L_{mn\_UMP}$ 이  $L_{mn\_LLR}$ 보다 더 크다. 그러므로 표준화 인수  $\alpha$ 를 다음과 같이 정할 수 있다.

$$\alpha = \frac{E [|L_{mn\_UMP}|]}{E [|L_{mn\_LLR}|]} \quad (2-14)$$

이 값으로  $L_{mn}$ 을 나눠주면 최종적으로

Normalized-BP 알고리즘에서의  $L_{mn}$ 은 다음과 같다.

$$L_{mn} = (-1)^{(\sigma_m \oplus \sigma_{mn})} \min_{n' \in N(m) \setminus n} |z_{mn'}| / \alpha \quad (2-15)$$

이러한  $\alpha$ 의 값은 매 반복 복호 때마다 또한 SNR 값이 변함에 따라 달라져야 하지만  $\alpha$ 값에 의해 가장 많은 영향을 받는 BER  $10^{-3}$ ~ $10^{-4}$  사이에서의 SNR값에 대응하는  $\alpha$ 값을 선택하여 고정시키는 것만으로도 충분한 효과를 얻을 수 있다.

### III. Modified Normalized-BP 알고리즘

UMP-BP 알고리즘은 LLR-BP 알고리즘에 비해  $L_{mn}$ 을 구하는 방법을 근사화 시킴으로써 성능은 조금 떨어지지만 복잡도를 크게 줄일 수 있었다. 이러한 UMP-BP 알고리즘의 성능을 좀더 좋게 하기 위해 표준화 인수  $\alpha$ 라는 값을 도입한 Normalized-BP 알고리즘이 제안되었다. Normalized-BP 알고리즘은 복잡도는 거의 증가하지 않으면서 LLR-BP 알고리즘에 거의 근접하는 우수한 성능을 보여주고 있다. 이러한 Normalized-BP 알고리즘의 성능을 더욱 개선시키기 위해서는 반복 복호 과정을 보다 세밀하게 분석해 볼 필요성이 있다.

#### 1. 행 방향 반복 복호 과정의 분석

UMP-BP 알고리즘의 경우 행 방향 반복 복호 과정은 다음과 같다.

$$\sigma_{mn} = \begin{cases} 1, & \text{if } z_{mn} > 0 \\ 0, & \text{if } z_{mn} \leq 0 \end{cases} \quad (3-1)$$

$$\sigma_m = \sum_{n \in N(m)} \sigma_{mn} \pmod{2} \quad (3-2)$$

$$L_{mn} = (-1)^{(\sigma_m \oplus \sigma_{mn})} \min_{n' \in N(m) \setminus n} |z_{mn'}| \quad (3-3)$$

여기서  $\sigma_m$ 이라는 값은 패리티 검사 방정식에 관련된 모든 비트들에 대하여 판정을 내려서 그 값들을 modulo 2 sum을 취한 것이다. 다시 말하면 반복 복호 과정에서 나온 신드롬(syndrome)이라고 볼 수 있다.  $\sigma_m$ 은 0 또는 1의 값을 가질 수 있으며

이 값이 행 방향 반복 복호 과정에 어떠한 영향을 주는지 살펴 볼 것이다.

1.1  $\sigma_m=0$ 일 때

$\sigma_m$ 이 0이라는 말은 현재 반복 복호 과정에서  $z_{mn}$ 으로 판정을 내린 결과 중에 오류가 없거나 짝수개의 오류가 있음을 나타낸다. 예를 들어 부호어  $c=(c_1, c_2, \dots, c_N)$ 이라 할 때 송신되는 부호열  $s=(s_1, s_2, \dots, s_N)$ 이며 여기서  $s_n=2c_n-1, n=1, \dots, N$ 이라고 하자. BPSK 변조방식을 가정하고 모두 0인 부호어를 보냈다고 하면 열무게는 4이고  $\sigma_m=0$ 인 경우에 대하여 비트의 구성은 다음과 같은 세가지 경우를 생각할 수 있다.

- $(c_1, c_2, c_3, c_4)=(-, -, -, -)$
- $(c_1, c_2, c_3, c_4)=(-, -, +, +)$
- $(c_1, c_2, c_3, c_4)=(+, +, +, +)$

이 중 오류가 두 개일 경우를 분석해보면 다음과 같다

1)  $c_1, c_2$

자신을 제외한 나머지 비트들이 -, +, +의 값을 가지고 있으므로 modulo 2 sum으로 0을 만족하기 위하여 부호를 -로 결정한다.

2)  $c_3, c_4$

자신을 제외한 나머지 비트들이 -, -, +의 값을 가지고 있으므로 modulo 2 sum으로 0을 만족하기 위하여 부호를 +로 결정한다.

위에서 살펴본 바와 같이  $\sigma_m=0$ 이고 자신에게 오류가 없는 경우에는 주위로부터 자신의 올바른 비트값을 업데이트 할 수 있고, 자신에게 오류가 있는 경우에는 주위로부터 잘못된 비트값을 업데이트 하게 된다.

1.2  $\sigma_m=1$ 일 때

$\sigma_m$ 이 1이라는 말은 현재 반복 복호 과정에서  $z_{mn}$ 으로 판정을 내린 결과 중에 한 개 이상의 홀수개의 오류가 있음을 나타낸다. 위에서와 같이 열무게 4인 부호를 예를 들 때 비트의 구성은 다음과 같은 두가지 경우를 생각할 수 있다.

- $(c_1, c_2, c_3, c_4)=(-, -, -, +)$
- $(c_1, c_2, c_3, c_4)=(-, +, +, +)$

이 중 오류가 한 개일 경우를 분석해보면 다음과 같다.

1)  $c_1, c_2, c_3$

자신을 제외한 나머지 비트들이 -, -, +의 값을 가지고 있으므로 modulo 2 sum으로 0을 만족하기 위하여 부호를 +로 결정한다.

2)  $c_4$

자신을 제외한 나머지 비트들이 -, -, -의 값을 가지고 있으므로 modulo 2 sum으로 0을 만족하기 위하여 부호를 -로 결정한다.

$\sigma_m=1$ 인 경우에는  $\sigma_m=0$ 일 때와는 반대로 자신에게 오류가 없는 경우에는 주위로부터 잘못된 비트값을 업데이트 하게 되며, 자신에게 오류가 있는 경우에는 주위로부터 오류를 바로 잡을 수 있는 올바른 비트값을 업데이트 하게 된다.

1.3  $\sigma_m$ 의 값을 이용한 성능 향상 방법

앞에서의 분석을 바탕으로  $\sigma_m$  값을 이용하여 복호 성능을 향상시키기 위해서  $\sigma_m=1$ 인 경우에 주목해 보자. 부호율이 1/2이고  $E_b/N_o=1$ 인 경우의 오류 확률은 대략 0.13정도가 되므로 사실상  $\sigma_m=1$ 인 경우에는 한 개의 오류가 있을 경우가 다른 홀수개의 오류가 있을 경우에 비해 훨씬 큰 확률을 가지게 된다. 그러므로 한 개의 오류가 있을 경우에 대하여 생각해보면 LDPC 부호의 행무게가 k라고 할 때 k개의 비트의 log-likelihood ratio 중에서 가장 작은 절대값을 가지는 비트의 경우 가장 오류가 있을 확률이 높다. 이것은 출력이 ±1인 BPSK 변조방식에 대하여 평균이 0이고 분산이  $\sigma^2$ 인 AWGN 채널 환경을 통하여 모두 0인 부호어를 보낸다고 할 때 송신된 두 개의 값  $y_1$ 과  $y_2$  ( $|y_1| > |y_2|$ )에 대해 다음과 같이 증명할 수 있다 주어진 두 개의 값의 오류 확률은 Sum-Product 알고리즘에서  $f_n^1$ 를 구할때와 같은 경우라고 생각할 수 있으므로 다음과 같다.

$$\Pr(c_n = 1 || y_1) = \frac{1}{1 + \exp \left[ -\frac{2|y_1|}{\sigma^2} \right]} \quad (3-4)$$

$$\Pr(c_n = 1 || y_2) = \frac{1}{1 + \exp \left[ -\frac{2|y_2|}{\sigma^2} \right]} \quad (3-5)$$

위에서 지수함수는 증가함수이므로 다음과 같이

$\exp \left[ -\frac{2|y_1|}{\sigma^2} \right] < \exp \left[ -\frac{2|y_2|}{\sigma^2} \right]$ 가 성립하고 결과적으로 다음 식이 성립하게 된다.

$$\Pr(c_n = 1 || y_1) < \Pr(c_n = 1 || y_2) \quad (3-6)$$

그러므로 LLR 값이 더 작은 비트의 오류가 있을 확률이 더 높다고 보는 것이 타당하다.

오류가 한 개 있을 경우에 패리티 검사 방정식을 구성하는 비트들 중에서 가장 작은 LLR 값을 가진 비트에 오류가 있다고 생각되어지므로 이러한 비트에 대해서는 업데이트 되는 값의 크기를 더 크게 해주어 올바른 정보를 크게 반영될 수 있도록 하는 것이 성능 향상에 도움을 줄 것이다. 또한 나머지 오류가 없다고 생각되어지는 비트들에 대해서는 업데이트되는 값의 크기를 더 작게 해주어 잘못된 정보를 적절 차단시킴으로써 성능을 향상시킬 수 있을 것이다. 이것은 결과적으로 Normalized-BP 알고리즘의 표준화 인수  $\alpha$  값을 한 개가 아니라 세 개로 사용하는 것과 같다.

## 2. 행 방향 반복 복호 과정에서 $\alpha$ 값의 선택에 대한 분석

UMP-BP 알고리즘의 행 방향 반복 복호 과정에서는 복잡도를 줄이기 위하여  $T_{mn}$  값을 구하는 과정을 다음과 같이 근사화한다.

$$T_{mn} = \prod_{n' \in \mathcal{N}(m) \setminus n} \frac{1 - \exp(z_{mn'})}{1 + \exp(z_{mn'})} \\ \approx (-1)^{\sigma_n \oplus \sigma_{mn}} \min_{n' \in \mathcal{N}(m) \setminus n} \left| \frac{1 - \exp(z_{mn'})}{1 + \exp(z_{mn'})} \right| \quad (3-7)$$

실제로 어떠한 방법으로 복호가 이루어지는지 예를 들어 살펴보기로 하자. 모두 0인 부호어를 보냈

다고 가정하면 열무게가 4인 경우에 대하여 비트의 구성을 각각의 비트의 절대값의 크기 순서에 따라 배열하여 다음과 같이 생각할 수 있다.

$$c_1 \ c_2 \ c_3 \ c_4, \quad (c_1 > c_2 > c_3 > c_4) \quad (3-8)$$

위와 같은 경우에 실제로 각각의 비트에 대하여 어떠한 절대값을 선택하게 되는지 복호 과정을 살펴 보면 다음과 같다.

### 1) $c_1$

자신을 속한 패리티 검사 방정식에서 자신을 제외하고 가장 작은 LLR 값을 가진  $c_4$ 를 선택하고 나머지  $c_2, c_3$ 는 생략된다.

### 2) $c_2$

자신을 제외하고 가장 작은 LLR 값을 가진  $c_4$ 를 선택하고 나머지  $c_1, c_3$ 는 생략된다.

### 3) $c_3$

자신을 제외하고 가장 작은 LLR 값을 가진  $c_4$ 를 선택하고 나머지  $c_1, c_2$ 는 생략된다.

4)  $c_4$  (주어진 패리티 검사 방정식에서 가장 작은 LLR 값인 경우)

자신을 제외하고 가장 작은 LLR 값을 가진  $c_3$ 를 선택하고 나머지  $c_1, c_2$ 는 생략된다

위와 같이 자신의 값이 주어진 패리티 검사 방정식에서 가장 작은 LLR 값을 가진 비트일 경우에는 근사화하기 위하여 생략되는 값은 다른 비트들에 비하여 가장 큰 LLR 값이 된다. LLR 값이 큰 경우는 다음과 같이  $T_{mn}$  값을 구할 때 실제로 곱해지는 값은 1에 가까운 값이 된다.

$$\frac{1 - \exp(|z_{mn}|)}{1 + \exp(|z_{mn}|)} \approx 1, \quad |z_{mn}| \gg 1 \quad (3-9)$$

그러므로 가장 작은 LLR 값을 가진 비트의 경우에는 근사화로 인해 생략된 값들이 다른 비트에 비하여 더 1에 가까운 값을 가지고 있게 된다. 이 경우 주위의 다른 비트들에 비하여  $E[|L_{mn\_LLR}|]$ 과  $E[|L_{mn\_UMP}|]$  값의 차이는 그다지 크지 않게 되고 따라서 더 작은  $\alpha$  값을 사용하는 것이 타당하다. 또한 나머지 비트들에 대

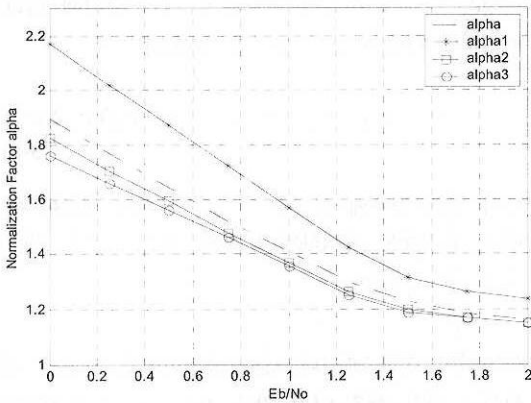


그림 1. (20000, 3, 6) 균일 LDPC 부호의  $E_b/N_o$ 에 따른  $\alpha$  값의 변화

해서는 가장 작은 LLR 값을 가진 비트를 제외하고 생각하면 전체적으로 기대값을 구할 때보다  $E [|L_{mn\_LLR}|]$ 과  $E [|L_{mn\_UMP}|]$  값의 차이가 더 커지게 된다. 그러므로 더 큰  $\alpha$  값을 사용하는 것이 타당하다.

위와 같이 행 방향 반복 복호 과정에서 각각의 비트에 대한 표준화 인수  $\alpha$  값의 분석에 의해서도 성능을 향상시키기 위해 앞에서와 동일한 결과를 얻을 수 있다.

Normalized-BP 알고리즘에서 표준화 인수  $\alpha$ 의 기대값을 구할 때 위에서 제안한 방법대로  $\alpha$ 의 값을 식 (3-10)과 같이 세분화하여 각각에 대한 기대값을 따로 구하여 보았다.

$$\begin{aligned} \alpha_1 &: \sigma_m = 1, z_{mn} \neq \text{minimum value} \\ \alpha_2 &: \sigma_m = 0 \\ \alpha_3 &: \sigma_m = 1, z_{mn} = \text{minimum value} \end{aligned} \quad (3-10)$$

실제로 컴퓨터 모의 실험을 통하여 (20000, 3, 6) 균일 LDPC 부호에 대하여 위에서 제안한 세 개의  $\alpha$  값을 찾아보면 그림 1과 같다.

위와 같이  $\alpha$ 의 값을 세분화하여 Monte Carlo 컴퓨터 모의 실험으로 각각의 기대값을 구해 보았을 때 제안한 방법에서처럼 실제로  $\alpha_1 > \alpha_2 > \alpha_3$  인 크기를 가진다는 것을 확인할 수 있다. 예를 들어 실제로 성능을 보기 위하여 컴퓨터 모의 실험을 할 때 기존의 Normalized-BP 알고리즘에서는 BER  $10^{-3}$ ~ $10^{-4}$  사이의 성능을 가지는  $E_b/N_o = 1.4$  dB

에서의 표준화 인수  $\alpha$  값인 1.25를 사용하였는데 제안한 새로운 알고리즘에서는  $\alpha_1 = 1.37$ ,  $\alpha_2 = 1.23$ ,  $\alpha_3 = 1.21$  값을 사용한다.

### 3. 제안된 알고리즘의 행 방향 반복 복호

Normalized-BP 알고리즘의 성능을 개선하기 위하여 기존의 표준화 인수  $\alpha$  값을  $\alpha_1, \alpha_2, \alpha_3$  ( $\alpha_1 > \alpha_2 > \alpha_3$ ) 세 개의 값으로 확장시켜서 이용한다.

$$L_{mn} = \begin{cases} (-1)^{\overline{(\sigma_m \oplus \sigma_{mn})}} \min_{n' \in N(m) \setminus n} |z_{mn'}| / \alpha_1 & \text{if } \sigma_m = 1, z_{mn} \neq \text{minimum value} \\ (-1)^{\overline{(\sigma_m \oplus \sigma_{mn})}} \min_{n' \in N(m) \setminus n} |z_{mn'}| / \alpha_2 & \text{if } \sigma_m = 0 \\ (-1)^{\overline{(\sigma_m \oplus \sigma_{mn})}} \min_{n' \in N(m) \setminus n} |z_{mn'}| / \alpha_3 & \text{if } \sigma_m = 1, z_{mn} = \text{minimum value} \end{cases} \quad (3-11)$$

위와 같이 변형된 알고리즘의 경우 기존의 알고리즘에 비하여 복잡하게 보이지만 실질적으로는  $\sigma_m$ 의 값은 복호과정에서 항상 계산되어지는 값이고  $z_{mn}$ 이 주어진 패리티 검사 방정식에서 가장 작은 값인지 아닌지는  $L_{mn}$ 을 구하는 과정에서 쉽게 판단할 수 있으므로 복잡도는 그다지 증가하지 않는다.

기존의 알고리즘과 비교하여 실제로 복호화 과정의 복잡도를 계산하여 보면 논문에서 제안된 Modified Normalized-BP 알고리즘의 경우는  $\sigma_m$ 이 0일 때와 1일 때를 나누어 생각하며 각각의 경우에 두 개의  $z_{mn}$  값을 선택하여 적절한 표준화 인수로 나누어주게 되므로 Normalized-BP 알고리즘의 경우에 비해 적절한 표준화 인수값을 선택하는 과정을 제외하고는 추가적인 곱셈 연산을 필요로 하지 않는다. 그러므로 Modified Normalized-BP 알고리즘은 실제적으로 한번의 반복 복호를 수행할 때 Normalized-BP 알고리즘에 비하여 단지 선택하는 부분만을 더 필요로 할 뿐 연산의 복잡도를 증가시키지 않는다는 것을 확인할 수 있다.

## IV. 제안된 알고리즘의 성능 분석

제안된 알고리즘에 대하여 Monte Carlo 컴퓨터 모의 실험을 수행하여 기존의 LLR-BP 알고리즘과

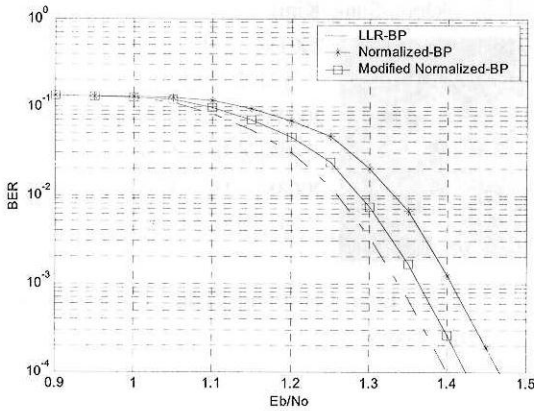


그림 2. (20000, 3, 6) 균일 LDPC 부호의 성능 곡선

Normalized-BP 알고리즘과 비교하였다. 그림 2는 (20000, 3, 6) 균일 LDPC 부호에서의 제안된 알고리즘의 성능을 보여주고 있다. 실험에 사용된 LDPC 부호의 부호율은 1/2이며 표준화 인수  $\alpha$ 는 1.25를 사용하였으며 제안한 새로운 알고리즘에서는  $\alpha_1=1.37$ ,  $\alpha_2=1.23$ ,  $\alpha_3=1.21$  값을 사용하였다. 위의 그림에서 기존의 Normalized-BP 알고리즘은 BER  $10^{-3}$ 에서 LLR-BP 알고리즘에 비해 약 0.07dB 정도의 성능 저하를 나타내는데 비해 본 논문에서 제안된 알고리즘은 기존의 알고리즘보다 향상되어 약 0.02dB 정도까지 근접하는 성능을 보이는 것을 알 수 있다.

그림 3은 (9999, 3, 5) 비균일 LDPC 부호에서의 제안된 알고리즘의 성능을 보여주고 있다. 부호율은 1/3이며 Normalized-BP 알고리즘의 표준화 인수  $\alpha$ 는 1.22를 사용하였으며 제안한 새로운 알고리즘에서는  $\alpha_1=1.33$ ,  $\alpha_2=1.20$ ,  $\alpha_3=1.19$  값을 사용하였다. Normalized-BP 알고리즘의 경우는 LLR-BP 알고리즘에 비해 BER  $10^{-3}$ 에서 약 0.1dB 정도의 성능 저하를 보이지만 논문에서 제안된 알고리즘의 경우는 약 0.04dB 정도까지 LLR-BP 알고리즘에 더 근접하는 성능을 보이는 것을 알 수 있다. 위의 컴퓨터 모의 실험 결과로부터 제안된 알고리즘이 복잡도 면에서는 그다지 큰 차이를 보이지 않으면서 좀더 좋은 성능을 보임을 알 수 있다.

### V. 결론

고품질의 신뢰도를 요구하는 4세대 이동통신 시

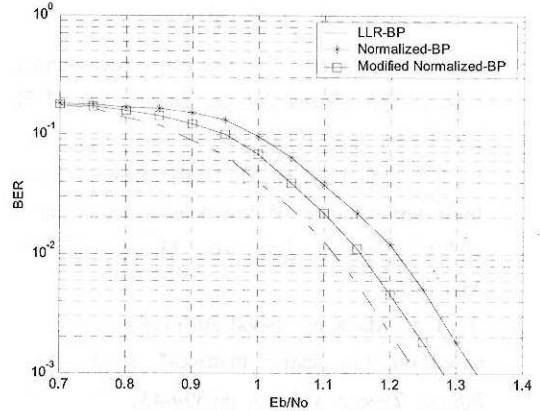


그림 3. (9999, 3, 5) 비균일 LDPC 부호의 성능 곡선

스템에 적합한 부호화 방식으로 많은 연구가 진행되어 오고 있는 LDPC 부호는 반복 복호 방법을 사용하여 Shannon의 이론적 한계에 접근하는 매우 우수한 성능을 나타낸다. Shannon의 한계에 접근하는 또 다른 부호화 방식인 터보 부호와 비교해볼 때 반복 복호를 통하여 정정되지 않은 오류들을 대부분 검출할 수 있고, 부호의 최소거리가 길이에 따라 커지므로 터보 부호와 달리 오류 마루 현상이 나타나지 않는다. 또한 터보 부호에 비해 부호율의 가변이 매우 간단하고, 그래프를 기반으로 한 반복 복호 방법을 사용함으로써 완전 병렬처리로 고속 처리가 가능하다.

본 논문에서는 LDPC 부호의 여러 가지 특징들을 설명하고 LDPC 부호의 생성 방법에 대하여 살펴보았다. 또한 LDPC 부호의 여러 가지 복호 알고리즘에 대하여 설명하고, 각각의 복호 방법의 특징과 성능을 비교 분석함으로써 기존의 방법에 비하여 성능을 향상시킬 수 있는 복호 방법을 제안하였다. 복호 성능을 향상시키기 위하여 Normalized-BP 알고리즘의 행 방향 반복 복호 과정을 분석하여 신드롬이 1이 나올 경우에 대하여 각각의 비트에 업데이트되는 값에 대한 신뢰도에 따라 표준화 인수 값을 달리함으로써 성능을 향상시킬 수 있는 Modified Normalized-BP 알고리즘을 제안하였다. 그림 2, 그림 3를 통하여 제안된 방법은 기존의 Normalized-BP 알고리즘에 비하여 복잡도를 거의 증가시키지 않으면서 LLR-BP 알고리즘에 대하여 BER 성능은 약 0.02dB 정도까지 근접하는 것을 확인할 수 있다.

참 고 문 헌

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theorz*, vol. IT-B, pp. 21-28, Jan. 1962.

[2] D. J. C. MacKay, R. M. Neal, Near Shannon limit performance of low-density parity-check codes," *Electron. Lett.*, vol. 33, pp. 457-458, Mar. 1997

[3] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices", *IEEE Trans. Inform. Theorz*, vol. 45, pp.399-431, Mar. 1999

[4] T. J. Richardson, R. L. Urbanke, "The Capacity of Low-Density Parity- Check Codes Under Message-Passing Decoding," *IEEE Trans. Inform. Theorz*, vol. 47, pp. 599-618, Feb. 2001

[5] M. C. Davey, "Error-correction using low-density parity check codes, degree of doctor of Philosophy", *Universitz of Cambridge*, Dec. 1999.

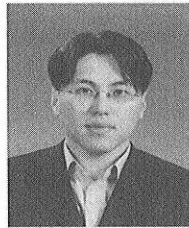
[6] Y. Kou, S. Lin, M. P. C. Fossorier, Low-Density Parity- Check Codes Based on Finite Geometries: A Rediscovery and New Results," *IEEE Trans. Inform. Theorz*, vol. 47, pp. 2711-2736, Nov. 2001

[7] M. Fossorier, M. Mihaljevic and H. Imai, Reduced complexity iterative decoding of low density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, pp. 673-680, May 1999

[8] J. Chen, M. P. C. Fossorier, "Near Optimum Universal Belief Propagation Based Decoding of Low-Density Parity Check Codes," *IEEE Trans. Commun.*, vol. 50, pp. 406-414, Mar. 200

김 준 성(Joon-Sung Kim)

준회원

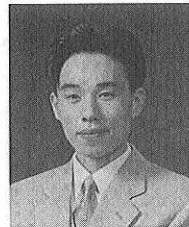


2001년 2월 : 연세대학교 전파  
공학과 졸업  
2003년 2월 : 연세대학교 전기  
전자공학과 석사  
2003년 3월~현재 : 연세대학교  
전기전자공학과 박사과정

<관심분야> Error Correcting Codes, PN Sequences,  
LDPC Codes

신 민 호(Min-Ho Shin)

정회원

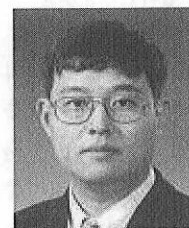


1996년 2월 : 연세대학교 전자  
공학과 졸업  
2000년 2월 : 연세대학교 전기  
전자공학과 석사  
2000년 3월~현재 : 연세대학교  
전기전자공학과 박사과정

<관심분야> Error Correcting Codes, PN Sequences,  
LDPC Codes

송 홍 엽(Hong-Yeop Song)

중신회원



1984년 2월 : 연세대학교 전자  
공학과 졸업  
1986년 5월 : USC 대학원 전  
자공학과 석사  
1991년 12월 : USC 대학원 전  
자공학과 박사  
1992년~1993년 : Post Doc.,

USC 전자공학과

1994년~1995년 : Qualcomm Inc., 선임연구원  
1995년 9월~현재 : 연세대학교 전기전자공학과 교수  
<관심분야> PN sequences, Error Correcting  
Codes, Spread Spectrum Communication Systems,  
Stream Cipher Systems