

# 저밀도 패리티 검사부호의 복호 알고리즘에 따른 성능 비교 분석

윤 태 현\*, 박 진 태\*, 주 언 경°

## Analysis of Performance according to LDPC Decoding Algorithms

Tae Hyun Yoon\*, Jin Tae Park\*, Eon Kyeong Joo°

### 요 약

LDPC(low density parity check)부호는 합-곱 알고리즘 기반의 반복복호를 통해 Shannon 한계에 근접하는 성능을 보인다. 합곱 알고리즘에서 체크노드와 비트노드의 확률 및 부가정보의 갱신 순서는 스케줄링 방법에 따라 달라지며 그에 따라 오류정정능력이나 반복복호의 횟수가 달라진다. 기존에 제안된 순차 BP 알고리즘을 사용한 LDPC의 복호는 표준 BP 알고리즘을 바탕으로 복호를 수행했을 경우에 비해 적은 평균반복복호 횟수에도 불구하고 좋은 성능을 가진다고 알려져 있다. 하지만 기존의 연구들에서는 이러한 성능 차이의 원인에 대한 연구는 미비하다. 따라서 본 논문에서는 두 알고리즘의 적용에 따른 LDPC 복호의 성공 여부에 따라 4가지 경우로 분류하고 각 경우를 비교한다. 이를 통해 두 알고리즘의 성능 차이의 원인을 분석하고 그 결과로 성능 차이의 원인이 패리티 검사행렬 내부의 사이클을 구성하는 비트노드들의 확률 값을 갱신할 때 알고리즘에 따른 갱신 과정의 차이에 있음을 보인다.

**Key Words** : LDPC code, sum-product algorithm, standard BP, shuffled BP, parity-check matrix

### ABSTRACT

LDPC (low density parity check) code shows near Shannon limit performance by iterative decoding based on sum-product algorithm (SPA). Message updating procedure between variable and check nodes in SPA is done by a scheduling method. LDPC code shows different performance according to scheduling schemes. The conventional researches have been shown that the shuffled BP (belief propagation) algorithm shows better performance than the standard BP algorithm although it needs less number of iterations. However the reason is not analyzed clearly. Therefore the reason of difference in performance according to LDPC decoding algorithms is analyzed in this paper. 4 cases according to satisfaction of parity check condition are considered and compared. As results, the difference in the updating procedure in a cycle in the parity check matrix is considered to be the main reason of performance difference.

### I. 서 론

1962년 Gallager에 의해 처음 제안된 저밀도 패리티검사(low density parity check; LDPC)<sup>[1]</sup> 부호

는 반복복호를 이용하여 Shannon 한계에 근접하는 성능을 보이며<sup>[2-3]</sup> 그 강력한 오류정정능력으로 인해 DVB-S2, WiMaX, Wi-Fi 등 차세대 통신의 표준으로 채택되는 등 현재 가장 각광받는 오류정정부호

※ 이 논문은 2012학년도 경북대학교 학술연구비에 의하여 연구되었음.

◆ 주저자 : 경북대학교 전자공학부, thyoon@ee.knu.ac.kr, 준회원

° 교신저자 : 경북대학교 전자공학부, ekjoo@ee.knu.ac.kr, 종신회원

\* 경북대학교 전자공학부, jintae08@ee.knu.ac.kr, 준회원

논문번호 : KICS2012-06-275, 접수일자 : 2012년 6월 7일, 최종논문접수일자 : 2012년 11월 7일

이다.

LDPC 부호의 복호는 합과 곱의 간단한 두 연산만으로 복호가 가능한 합-곱 알고리즘(sum-product algorithm; SPA)을 기반으로 한다<sup>4)</sup>. 표준 BP(standard belief propagation) 알고리즘<sup>4)</sup>은 LDPC 부호의 복호 시 합-곱 알고리즘에 따른 확률 및 부가정보 값의 계산과 그 값들의 갱신을 병렬적으로 처리하는 방식이다. 그러나 표준 BP 알고리즘을 기반으로 복호기를 구성할 경우 병렬 연산을 위해 많은 수의 곱셈기와 덧셈기가 필요할 뿐만 아니라 이를 구동하기 위해 많은 소비 전력이 요구된다. 이는 차세대 통신 시스템에 LDPC 부호를 적용하는데 있어 문제점으로 지적된다<sup>5-6)</sup>. 따라서 표준 BP 알고리즘의 단점을 보완하기 위해 체크노드와 비트노드의 연산을 순차 또는 동적 처리하는 복호 알고리즘들이 제안되었으며 이 알고리즘들은 오류정정능력이나 반복복호 횟수에서 서로 차이를 보인다<sup>7-8)</sup>. 그러나 기존의 연구들에서는 복호 알고리즘에 따른 오류정정능력이나 평균 반복복호 횟수, 복잡도를 비교를 위주로 연구가 진행되었을 뿐<sup>7-14)</sup> 그 성능차이의 원인에 대한 연구는 미비하다.

제안된 여러 알고리즘 중 순차 BP 알고리즘은 확률 값의 갱신 순서만 다를 뿐 표준 BP 알고리즘과 동일한 연산을 통해 복호를 수행함에도 표준 BP 알고리즘에 비해 좋은 성능을 보이는 것으로 알려져 있다<sup>7)</sup>. 즉, 순차 BP 알고리즘을 표준 BP 알고리즘과 비교할 경우 다른 알고리즘들과는 다르게 추가적인 연산 및 처리에 의한 성능 차이를 배제하고 확률 값 갱신 순서에 따른 성능 차이의 원인을 분석할 수 있으리라 기대된다. 따라서 본 논문에서는 순차 BP 알고리즘을 표준 BP 알고리즘과 비교하여 복호 알고리즘에 따른 성능 차이의 원인을 분석한다.

본 논문은 I장의 서론에 이어 II장에서는 BP 알고리즘과 LDPC 부호의 복호 알고리즘들 중 본 논문에서 대상으로 하는 표준 BP 알고리즘과 순차 BP 알고리즘을 소개한다. III장에서는 오류정정능력의 차이 확인을 위한 모의실험 환경에 대해 소개하고 그 결과를 보인다. 이를 바탕으로 두 알고리즘에 따른 오류정정능력 차이의 원인을 분석하고, 마지막으로 IV장에서 결론을 맺는다.

## II. 저밀도 패리티 검사부호의 복호 알고리즘

LDPC 부호는 합-곱 알고리즘 혹은 신뢰도 전파

(belief propagation) 알고리즘으로 알려진 확률 전파 방법을 반복적으로 수행함으로써 복호할 수 있다. 이는 Tanner 그래프<sup>4)</sup>를 통해 효과적으로 표현할 수 있다. Tanner 그래프 상에서 체크노드는 패리티 검사행렬의 각 행에 해당하고, 비트노드는 패리티 검사행렬의 각 열, 즉 부호어의 각 비트에 해당한다. 그리고 패리티 검사행렬의 원소가 1인 경우 그 위치에 해당하는 체크노드와 비트노드를 연결한다. 복호는 연결된 각각의 노드가 패리티검사 조건을 만족시키도록 확률 및 부가정보 값을 갱신하며 수행한다<sup>4)</sup>.

### 2.1. 표준 BP 알고리즘

표준 BP 알고리즘은 LDPC 복호를 위해 최초 제안된 복호 방법으로, 앞서 설명한 합-곱 알고리즘에 사용 되는 체크노드와 비트노드의 연산을 병렬 처리한다. 즉, 표준 BP 알고리즘에서는 모든 비트노드들에서 모든 체크노드들에게 확률 값이 동시에 전달되고 이후 모든 체크노드들에서 모든 비트노드들에게 부가정보 값이 동시에 전달된다. 이후 비트노드에서는 0 또는 1일 확률 값을 연산 후 결정판을 통해 비트노드의 한번 복호 후의 값을 결정한다. 이때 반복복호 중 한번 복호란 모든 비트노드의 확률 값과 체크노드의 부가정보 값이 한번 갱신되는 것을 의미한다. 표준 BP 알고리즘은 이러한 병렬처리로 인해 연산 속도가 빠른 장점이 있지만, 각 노드에서 요구하는 연산을 동시에 처리하기 위해 많은 수의 곱셈기와 덧셈기가 요구되는 단점도 있다.

### 2.2. 순차 BP 알고리즘

순차 BP 알고리즘은 LDPC 복호에 사용되는 합-곱 알고리즘의 체크노드와 비트노드의 연산을 순차적으로 처리한다. 즉 순차 BP 알고리즘에서는 우선 첫 번째 비트노드의 확률 값이 연결된 체크노드에 전달되며 각 체크노드에서는 전달된 확률 값을 바탕으로 계산된 부가정보 값을 연결된 비트노드들에게 전달한다. 이렇게 전달된 부가정보 값을 바탕으로 앞의 갱신 과정을 두 번째 비트노드에서 마지막 비트노드까지 순차적으로 수행하여 전체 비트노드의 확률 값을 갱신한다. 이와 같이 순차 BP 알고리즘은 순차적으로 비트노드를 갱신하므로 한번 복호 과정 중 특정 비트노드를 갱신 할 때, 이미 갱신되어있는 비트노드의 확률 값도 사용할 수 있는 특징이 있다. 또한 하나의 비트노드씩 순차적으로 갱신

함으로 하나의 검사노드와 비트노드를 위한 덧셈기와 곱셈기만으로 복호가 가능한 장점이 있지만, 한번 복호를 위한 시간이 표준 BP 알고리즘에 비해 부호어 길이에 해당하는  $N$ 배만큼 증가하는 단점도 있다.

### III. 모의실험 결과 및 분석

두 알고리즘에 따른 성능 차이를 확인하기 위해 모의실험을 수행하였다. 모의실험 환경은 IEEE 802.16e<sup>[15]</sup>와 IEEE 802.16n<sup>[16]</sup>에서 고려되는 비균일 LDPC 부호와 부호어 길이 672, 1248, 1920을 대상으로 하였고, 부호율은 공통적으로 많이 사용되는 1/2, 2/3, 3/4, 5/6을 사용하였다. 실험의 편의상 AWGN(additive white Gaussian noise) 채널에서 BPSK(binary phase shift keying) 변조하는 것으로 가정하였다. 그리고 최대 반복복호 횟수는 각 알고리즘의 확률 변화를 충분히 관찰하기 위해서 200회로 설정하였다. 각 실험에 사용된 프레임의 수는 1,000,000개를 사용했다.

두 알고리즘의 성능차이의 원인을 분석하기 위해서 패리티검사 조건의 만족 여부를 기준으로 모두 4 가지 경우로 분류했다. 패리티 검사조건이란 한번 복호 후 복호된 부호어가 오류가 없는 합법적인 부호어라 판단되면 복호를 종료하는 것이다. 이 조건을 만족하는 경우는 설정된 최대 반복복호 횟수 이전에 모든 오류를 정정하여 복호를 끝낸 것이고, 이 조건을 만족하지 않는 경우는 최대 반복복호 횟수에 도달할 때까지도 오류를 정정하지 못하여 복호를 성공하지 못한 것이다. 나누어진 경우는 표 1과 같고, 표 2 는 부호어 길이가 672이고 부호율 1/2 일때의 신호대잡음비에 따른 각 경우의 발생 빈도이다. 아래의 결과에서 보인 것과 같이 일반적으로 경우 2가 3의 경우보다 빈번하게 발생하였다.

표 1. 패리티 검사조건 만족 여부에 의해 분류된 4가지 경우  
Table 1. The classified 4 cases according to satisfaction of parity check condition

Case \ Algorithm	Standard BP	Shuffled BP
Case 1	Satisfied	Satisfied
Case 2	Not satisfied	Satisfied
Case 3	Satisfied	Not satisfied
Case 4	Not satisfied	Not satisfied

부호어 길이나 부호율에 따라 차이는 있었지만 전반적인 경우 발생 빈도의 추이는 유사하여 다른 부호율 및 부호어 길이 1248과 1920에서의 결과는 생략하였다. 두 경우의 발생 원인은 LDPC 패리티 검사행렬에서 생기는 짧은 길이의 사이클 내에 존재하는 비트노드의 확률 값을 갱신할 때 복호 알고리즘에 따른 갱신 방법의 차이에 의한 것으로 판단된다.

표 2. 부호어 길이 672, 부호율 1/2에서 경우 발생 빈도  
Table 2. Number of cases at codeword length of 672 with code rate of 1/2

Eb/N0 \ Case	Case 1	Case 2	Case 3	Case 4
2.1	989997	2901	243	6859
2.2	994070	2073	174	3683
2.3	996706	1169	87	2038
2.4	998049	871	42	1039
2.5	998947	557	32	464
2.6	999388	368	15	229
2.7	999679	212	10	99
2.8	999853	103	5	39
2.9	999931	52	5	12
3.0	999978	21	0	1

그러나 일반적으로 LDPC 부호가 적용되는 모의 실험 환경에서 사용되는 패리티 검사행렬은 그 크기로 인해 너무 많은 수의 노드간 연결과 사이클이 존재하므로 이러한 요인에 대한 분석이 굉장히 복잡하고 어렵다. 따라서 본 논문에서는 일반적으로 가장 우수한 성능을 내는 것으로 알려진 패리티 검사행렬 생성 방법<sup>[2][17]</sup>을 기반으로 할 경우 생성할 수 있는 가장 작은 크기인 행의 길이가 28, 열의 길이가 56인 패리티 검사행렬을 사용하였다. 실험 환경은 앞선 모의실험과 동일하다. 표 3은 앞서 생성된 (28×56) 크기의 패리티 검사행렬을 이용해 각각 2~5dB의 신호대잡음비에 따른 각 경우의 발생 빈도를 나타낸 것이다. 발생 횟수의 차이는 있지만 패리티 검사행렬의 크기가 작음에도 불구하고 경우 2가 경우 3 보다 더 빈번하게 발생하는 추이는 앞선 경우들에서와 동일하다. 따라서 이 패리티 검사행렬을 이용하여 경우 2와 3을 분석하고 두 알고리즘의 오류정정능력 차이의 원인을 확인한다.

표 3. 2dB~5dB에서의 경우 발생 빈도  
Table 3. Number of cases at 2dB~5dB

Eb/N0 \ Case	Case 1	Case 2	Case 3	Case 4
2.0	767522	5491	2106	224881
2.5	867532	7011	1904	123553
3.0	932018	5534	1119	61329
3.5	969524	4249	706	25521
4.0	989211	2483	431	7876
4.5	996635	812	145	2408
5.0	998768	409	36	787

먼저 순차 BP가 표준 BP 알고리즘에 비해 좋은 오류정정능력을 가지는 것을 확인할 수 있는 것은 경우 2를 통해서이다. 경우 2는 초기 수신된 오류 비트가 많고, 그 오류 비트들이 10번째 비트와 45번째 비트 사이에 주로 위치할 경우에 발생한다. 이 경우 표준 BP 알고리즘은 비트의 확률 값들을 10번째 비트 앞쪽 또는 45번째 비트 뒤쪽에 위치한 비트로 오류를 전달하고 전달받기를 반복하는 반면, 순차 BP 알고리즘은 10번째 비트 앞과 45번째 비트 뒤에 위치한 비트들이 먼저 옳은 값으로 수렴해 가면서 패리티 검사조건을 만족한다. 경우 3은 순차 BP가 표준 BP 알고리즘에 비해 좋은 성능을 가진다는 일반적인 성능에 반대 되는 경우이다. 이러한 경우가 발생하는 원인은 패리티 검사행렬 구조에 두 알고리즘의 갱신 방식의 차이가 복합적으로 영향을 미친 것으로 판단된다. 따라서 동일한 패리티 검사행렬 구조에서 두 알고리즘을 이용해 복호를 수행했을 때의 특징을 살펴보기 위해 각 비트노드 위치에 따라 발생한 오류를 누적시킨 분포특성을 살펴보았다. 그림 1은 위 표 3에서의 경우 3, 즉 일반적으로 알려진 것과는 다르게 동일한 오류가 발생했을 때 표준 BP 알고리즘으로는 수정이 가능하지만 순차 BP 알고리즘으로는 이를 수정하지 못하는 경우에 2dB의 신호대잡음비에서 발생된 오류를 누적하여 나타낸 것이다. 그림 1에서 보는 바와 같이 표준 BP 알고리즘은 순차 BP 알고리즘에 비해 상대적으로 더 많은 위치에 오류가 발생한 반면, 순차 BP 알고리즘은 3, 13, 20, 27, 35번째 비트에 집중적으로 오류가 발생하였다.

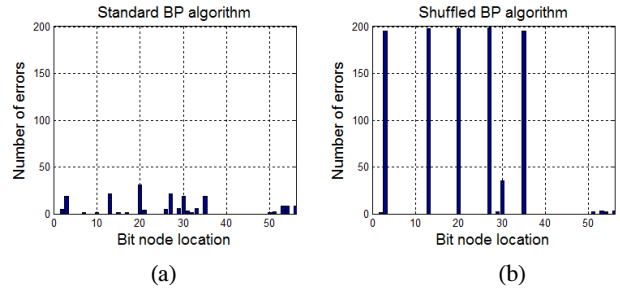


그림 1. 2dB에서 비트 위치별 오류분포  
(a) 표준 BP 알고리즘 (b) 순차 BP 알고리즘  
Fig. 1. Number of errors according to bit location at 2dB.  
(a) Standard BP algorithm (b) Shuffled BP algorithm

이 비트들을 Tanner 그래프로 표현한 결과 그림 2에서와 같이 비트노드들은 길이가 6과 8인 사이클을 형성하고 있었다.

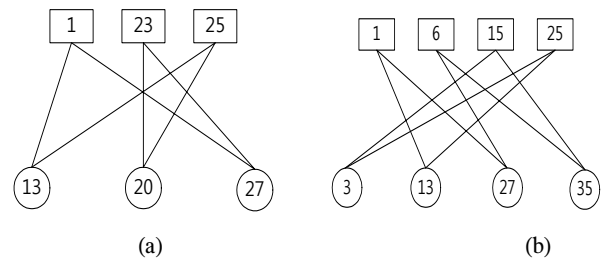


그림 2. 비트노드의 연결  
(a) 길이가 6인 사이클을 구성하는 13, 20, 27번 비트노드  
(b) 길이가 8인 사이클을 구성하는 3, 13, 27, 35번 비트노드  
Fig. 2. Connection of bit node  
(a) 6 cycle with the 13th, 20th, and 27th bit node  
(b) 8 cycle with the 3rd, 13th, 27th and 35th bit node

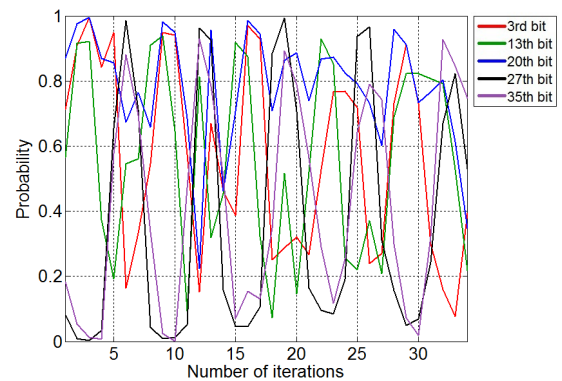
그림 1과 그림 2를 통해 순차 BP 알고리즘은 반복복호 수행 중 사이클 구조에 오류가 집중되어 버린 경우 패리티 검사조건을 만족하지 않음을 알 수 있으며 이는 순차 BP 알고리즘이 표준 BP 알고리즘에 비해 사이클 구조에 취약할 수 있음을 보여주는 예이다. 순차 BP 알고리즘이 반복복호 수행 중 사이클 구조에 오류가 집중되어 버린 경우 제대로 된 오류정정을 할 수 없는 이유는 사이클의 구조와 순차 BP 알고리즘의 갱신 순서의 영향으로 사료된다. 예를 들어 그림 2(a)와 같은 사이클이 6인 구조에 오류가 집중된 경우 순차적으로 13번째 비트를 갱신하면, 13번째 비트는 이미 오류인 20, 27번째 비트로부터 부가정보 값을 전달받아 갱신하므로 오류가 된다. 20번째 비트도 오류인 확률 값으로 갱신된 13번째 비트와 오류인 27번째 비트의 확률 값을 전달 받아 갱신하므로 역시 오류가 된다. 따라서 반복복호를 할수록 패리티 검사를 만족하지 않는 오류 값으로 수렴한다. 반면 표준 BP 알고리즘은 사이클에 오류가 집중 되더라도 모든 비트노

드들을 동시에 갱신하므로 사이클 구조 외부로부터 옳은 확률 값을 전달받아 사이클 내부의 오류는 줄어들지만, 내부의 오류가 외부로 전달되면서 사이클 외부의 새로운 비트위치에 오류가 발생한다. 따라서 반복복호 횟수가 증가함에 따라 비트 확률 값이 지속적으로 진동한다. 이러한 차이로 표준 BP 알고리즘은 순차 BP 알고리즘 보다 더 많은 비트위치에 오류가 발생한다. 그러나 이 경우 표준 BP 알고리즘은 위 과정을 반복하는 과정에서 비트 확률 값들이 활발하게 변하게 되는데, 이때 몇몇 비트의 확률 값이 옳은 확률 값을 가질 때 드물게 패리티 검사를 만족하는 것이다.

그림 3은 이를 확인하기 위해 경우 3일 때 각 알고리즘의 반복복호 횟수에 따른 확률 값 변화를 그래프로 나타낸 것으로 (28×56) 크기의 패리티 검사행렬을 사용하여 2dB에서 실험한 결과이다. 확률 값 변화 추이는 가독성을 높이기 위해 집중적으로 오류가 발생한 3, 13, 20, 27, 35번째 비트의 확률 값 변화 추이를 나타냈다. 그림 3 (a)는 표준 BP 알고리즘의 확률 값 변화 추이로서 처음 패리티 검사조건을 만족하는 34번 반복복호까지를 나타내었고, 그림 3 (b)는 순차 BP 알고리즘의 확률 값 변화 추이로서 끝까지 복호가 성공하지 않아서 최대 반복복호 횟수인 200번 반복복호까지를 나타내었다. 그림에서 볼 수 있듯이 표준 BP 알고리즘은 반복복호 횟수에 따라 지속적으로 진동을 반복하다가 34번 반복복호 후 패리티 검사 조건을 만족하여 복호를 정지한다. 반면, 순차 BP 알고리즘은 각 비트의 확률 값들이 복호가 성공한 것처럼 수렴하는 형태를 가지지만 그 값은 합법적인 부호어가 아닌 값으로 수렴하여 반복복호 횟수가 증가하여도 복호가 성공하지 않는다. 이는 실제로는 오류이나 오류로 판단하지 못하고 올바른 비트로 판단하는 미검출 오류이다. 복호 성공 여부만으로 이를 비교했을 때는 마치 순차 BP 알고리즘으로 정정하지 못하는 오류를 표준 BP 알고리즘으로는 정정할 수 있는 것처럼 보이지만 그림 3의 (a)에서 확인 할 수 있듯이 표준 BP 알고리즘은 적절한 값으로 수렴하여 복호가 성공하는 제대로 된 복호 성공이 아니고 확률 값이 계속해서 변화하다가 특정 순간에 단지 패리티 검사조건이 만족됨에 의하여 마치 오류가 정정된 것처럼 여겨지는 특수한 상황이다. 이는 적절한 복호 성공이라 보기 힘들다.

그림 4는 그림 3의 (a)의 경우가 34번째 반복복호 이후에도 적절한 확률 값으로의 수렴하지 않음

을 확인하기 위해 적절한 복호라 판단될 경우 복호를 정지하도록 하지 않고 최대 반복복호 횟수인 200회까지 반복복호를 수행하였을 때의 확률 변화를 나타낸 것이다. 즉 그림 4에서는 34번째 반복복호까지는 그림 3의 (a)의 경우와 동일하지만 그 이후의 확률의 변화를 보여준다. 그림 4에서 보는 바와 같이 패리티 검사조건을 만족한 34번째 반복복호 이후에도 계속 확률 값이 진동해 패리티 검사조건을 만족하지 못했다. 이후 109 번째에서 다시 패리티 검사조건을 만족하는 경우가 발생하고 그 후에는 역시 확률 값이 변화해 패리티 검사조건을 만족하지 못했다. 최대 반복복호 횟수까지 동일한 현상이 비주기적으로 142, 176번째에서 2번 더 발생했고, 그림 4에서 굵은 실선으로 표시 한 부분이 그 시점이다. 하지만 이는 일반적으로 최초 패리티 검사조건을 만족한 후 확률 값이 더 이상 변화하지 않고 적절한 값으로 수렴하는 제대로 된 복호 성공과는 다른 특수한 경우로서 제대로 된 복호 성공이라 보기 어렵다. 또한 이러한 경우는 표 2와 3에서 볼 수 있듯이 경우 2에 비해 낮은 발생 빈도를 가진다.



(a)

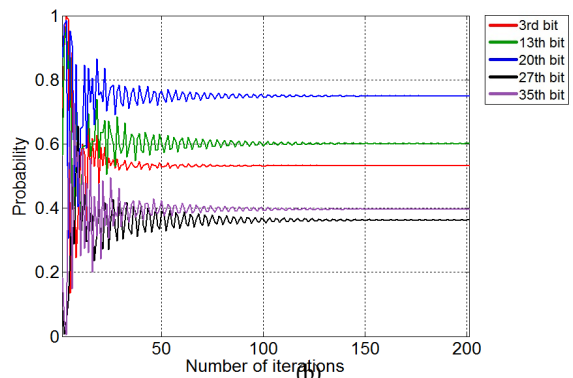


그림 3. 확률 값 변화 추이  
(a) 표준 BP 알고리즘 (b) 순차 BP 알고리즘  
Fig. 3. Transition of probability.

(a) Standard BP algorithm (b) Shuffled BP algorithm

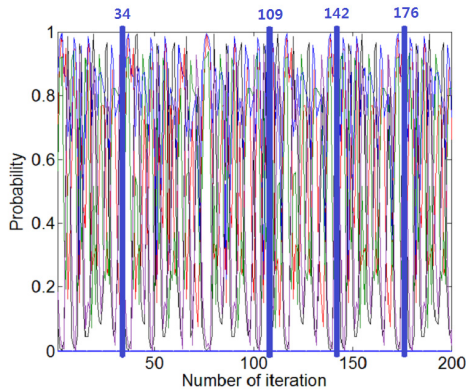


그림 4. 경우 3 발생 시 표준 BP 알고리즘의 확률 값 변화 추이  
 Fig. 4. Transition of probability with the standard BP algorithm in case 3

### V. 결 론

기존 LDPC 부호의 복호 알고리즘에 관한 연구들은 두 개 혹은 그 이상의 알고리즘들의 오류정정 능력과 반복복호 횟수, 복잡도를 단순 비교한 것에 반해 본 논문에서는 표준 BP와 순차 BP 알고리즘을 대상으로 오류정정능력 차이의 원인을 분석하였다.

LDPC 부호는 복호를 수행할 때, 동일한 패리티 검사행렬과 초기 값을 사용하더라도 알고리즘에 따라 패리티 검사 조건에 대해 서로 다른 결과를 나타낼 수 있다. 이러한 경우를 4가지로 나누고 각 경우를 신호대잡음비에 따른 발생 빈도를 살펴보았다. 그 결과 신호대잡음비가 증가 할수록 두 알고리즘 모두 패리티 검사조건을 만족하는 경우는 지속적으로 증가하였고, 그 다음 두 알고리즘 모두 패리티 검사조건을 만족하지 않는 경우와 두 알고리즘 중 하나만 만족하는 경우는 지속적으로 감소하였다. 두 알고리즘 모두 패리티 검사조건을 만족하는 경우와 표준 BP 알고리즘은 만족하지 않지만 순차 BP 알고리즘은 만족하는 경우를 통하여 일반적으로 알려진 것과 같은 순차 BP 알고리즘이 표준 BP 알고리즘 보다 더 좋은 오류정정능력을 가짐을 확인할 수 있었다. 하지만 일반적으로 알려진 것과 다르게 표준 BP 알고리즘은 패리티 검사조건을 만족하고 순차 BP 알고리즘은 만족하지 않는 경우도 있었다. 이 경우 각 비트위치 별로 발생한 오류의 개수를 살펴본 결과 오류가 특정비트들에서 집중적으로 발생하였고, 그 비트들의 특징은 서로 6 혹은

8 사이클을 형성하고 있었다. 이러한 사이클을 구성하는 비트들에 오류가 집중적으로 발생했을 때 순차 BP 알고리즘은 특정 값으로 수렴되어 반복복호를 수행하여도 오류가 정정되지 않았다. 반면 표준 BP 알고리즘은 사이클 내부에서 외부로, 또 다시 외부에서 내부로 오류 전달이 반복되면서 비트 확률 값들이 활발하게 바뀌었다. 이 과정에서 일반적으로 반복복호 횟수가 증가함에 따라 비트 확률 값이 수렴하는 것과는 다르게 경우의 수는 적었지만 우연히 패리티 검사조건을 만족하였다.

본 논문에서는 표준 BP와 순차 BP 알고리즘만을 대상으로 하였지만, 최근 LDPC 복호를 위한 다양한 알고리즘이 추가적으로 제안되고 있다. 따라서 본 논문에서 제시한 성능차이 비교 분석을 이용하여 알고리즘의 특징 및 장단점을 파악하고, 이를 통해 특정 통신 환경이나 시스템에 적합한 LDPC 복호 알고리즘의 선택에 참고 자료로서 활용 가능할 것으로 사료된다.

### References

- [1] R. G. Gallager, "Low-density parity-check code", *IRE Trans. Inform. Theory*, vol. 8, no. 1, pp. 21-28, Jan. 1962.
- [2] S. Y. Chung, Jr. G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity check codes within 0.0045dB from the Shannon limit", *IEEE Commun. Lett.*, vol. 5, pp. 58-60, Feb. 2001.
- [3] D. J. C. Mackay, "Good error-correcting codes based on very sparse matrices", *IEEE Trans. Inform. Theory*, vol. 45, pp. 399-431, Mar. 1999.
- [4] F. R. Kschischang, "Factor graphs and the sum-product algorithm", *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498-519, Feb. 2001.
- [5] J. Jin and C. Y. Tsui, "A low power layered decoding architecture for LDPC decoder implementation for IEEE 802.11n LDPC codes". in *Proc. IEEE ISLPED 2008*, Bangalore, India, pp. 253-258, Aug. 2008.
- [6] X. Y. Hu, E. Eleftheriou, D. M. Arnold and A. Dholakia, "Efficient implementations

of the sum-product algorithm for decoding LDPC codes”, in *Proc. IEEE GLOBECOM 2001*, San Antonio, TX, vol. 2, pp. 1036-1036E, Nov. 2001.

[7] J. Zhang and M. P. C. Fossorier, “Shuffled iterative decoding”, *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209-213, Feb. 2005.

[8] A. I. V. Casado, M. Griot and R. D. Wesel, “Informed dynamic scheduling for BP decoding of LDPC codes”, in *Proc. IEEE ICC 2007*, Glasgow, Scotland, pp. 932-937, June 2007.

[9] S. K. Yu, S. G. Kang and E. K. Joo, “A modified sum-product algorithm for error floor reduction in LDPC codes”, *J. KICS*, vol. 35, no. 5, pp. 423-431, May. 2010.

[10] J. Zhang, Y. Wang and M. Fossorier, “Replica shuffled iterative decoding”, in *Proc. IEEE ISIT 2005*, Adelaide, Australia, pp. 454-458, Sep. 2005.

[11] Z. C. Gang, Y. J. Sheng, L. X. Hong and L. J. Ru, “Improvement of shuffled iterative decoding”, in *Proc. IEEE ITW 2006*, Beijing, China, pp. 114-116, Oct. 2006.

[12] A. I. V. Casado, M. Griot and R. D. Wesel, “Improving LDPC decoders via informed dynamic scheduling”, in *Proc. IEEE ITW 2007*, Lake Tahoe, California, pp. 208-213, Sep. 2007.

[13] G. Han and X. Liu, “An efficient dynamic scheduling for layered belief propagation decoding of LDPC codes”, *IEEE Commun. Lett.*, vol. 13, no. 12, Dec. 2009.

[14] J. H. Kim, M. Y. Nam, H. Y. Song and K. M. Lee, “Variable to check residual belief propagation for informed dynamic scheduling of LDPC codes”, in *Proc. IEEE ISITA 2008*, Auckland, New Zealand, pp. 1-4, Dec. 2008.

[15] IEEE Standard for local and metropolitan area networks part 16 : Air Interface for fixed and mobile broadband wireless access systems amendment 2 : Physical and medium access control layers for combined fixed and mobile operation in licensed

bands and corrigendum 1, *IEEE Std. 802.16e*, 2005.

[16] IEEE P802.11n/D3.07, Draft amendment to standard for Information exchange between systems local and metropolitan networks specific requirements part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications : Enhancements for higher throughput, *IEEE Std. 802.16n*, 2008.

[17] T. Richardson, A. Shokrollahi, and R. Urbanke, “Design of probably good low density parity check codes”, in *Proc. IEEE ISIT 2000*, Sorrento, Italy, pp. 202, Jun. 2000.

윤 태 현 (Tae Hyun Yoon)



부호이론

2005년 2월 경북대학교 전자  
전기공학부 학사 졸업  
2007년 2월 경북대학교 전자  
공학과 석사 졸업  
2007년 3월~현재 경북대학교  
전자공학과 박사과정  
<관심분야> 채널코딩, 변/복조

박 진 태 (Jin Tae Park)



2008년 2월 위덕대학교 정보통신공학과 학사 졸업  
2012년 2월 경북대학교 전자공학과 석사 졸업  
<관심분야> 채널코딩, 변/복조, 부호이론

주 언 경 (Eon Kyeong Joo)

한국통신학회 논문지 35권 5호 참조