

LDPC 부호를 위한 복잡도와 대기시간을 낮춘 VCRBP 알고리즘

준회원 김정현*, 종신회원 송홍엽*

Reduced Complexity-and-Latency Variable-to-Check Residual Belief Propagation for LDPC Codes

Jung-Hyun Kim* Associate Member, Hong-Yeop Song* Lifelong Member

요약

본 논문에서는, LDPC 부호를 위한 node-wise VCRBP의 개선된 기법인, 강제수렴 node-wise VCRBP와 부호 기반 node-wise VCRBP를 제안한다. 두 가지 기법 모두 node-wise VCRBP에 비하여 매우 적은 오류 정정 성능 열화만으로 복호 복잡도와 대기시간을 현저하게 줄인다.

Key Words : LDPC Codes, VCRBP, Fast Convergence Decoding, Informed Dynamic Scheduling

ABSTRACT

This paper proposes some new improved versions of node-wise VCRBP algorithm for low-density parity-check (LDPC) codes, called forced-convergence node-wise VCRBP algorithm and sign based node-wise VCRBP, both of which significantly reduce the decoding complexity and latency, with only negligible deterioration in error correcting performance.

I. 서론

LDPC 부호^[1]는 낮은 복호 복잡도로 Shannon의 채널 용량 한계에 거의 근접하는 성능을 갖는 오류 정정 부호로 다양한 통신 분야에서 주목받고 있다. 대표적인 복호 알고리즘으로 모든 변수노드와 체크노드의 메시지를 동시에 전달하는 표준 신뢰 전달 (belief propagation 이하 BP) 알고리즘이 있으며, 이를 반복 복호 횟수 면에서 개선시킨 순차적 스케줄링 BP 알고리즘이 있다. 순차적 스케줄링 기법은 변수노드 방향으로 업데이트하는지 체크노드 방향으로 업데이트 하는지에 따라 Shuffled BP (SBP)와 Layered BP (LBP)로 구분된다. 이러한 순차적 스케줄링 기법들은 표준 BP 알고리즘 보다 두 배의

복호 수렴속도를 갖는다^{[2][3]}. SBP와 LBP 기법이 소개된 이후로, Replica SBP^[4], 개선된 SBP^[5], 결합 열-행 복호^[6], 에지기반 스케줄링 BP^[7] 등과 같은 더욱 향상된 기법들이 연구되고 있다.

최근 레지듀얼 BP (Residual Belief Propagation 이하 RBP)^[8] 불리는 매우 효과적인 동적 스케줄링 기법이 소개되었으며, 또한 이를 LDPC 부호에 적용시킨 RBP 복호 알고리즘^{[9][10]} Variable-to-Check RBP(VCRBP) 복호 알고리즘^{[11][12]} 제안되었다. 위의 두 가지 동적 스케줄링 기법들은 모두 기존의 정적 스케줄링 기법들에 비하여 더욱 빠른 복호 수렴속도를 갖는다. 또한 트래핑 셋 (trapping set)을 효과적으로 해결함으로써 정적 스케줄링 기법들에 비하여 현저한 성능 개선을 보인다^{[9][10]}. 그러나 RBP

※ 본 연구는 한국과학재단 특정기초연구(R01-2008-000-20578-0)지원으로 수행되었음.

* 연세대학교 전기전자공학부 부호및암호 연구실({jh.kim06, hysong}@yonsei.ac.kr)

논문번호 : KICS2009-04-152, 접수일자 : 2009년 4월 13일, 최종논문접수일자 : 2009년 5월 22일

복호 알고리즘은 그리디니스 (greediness) 특성으로 인해 성능 면에서 결점을 갖을 뿐만 아니라^{9)[10]}, 복잡도 면에서도 결점이 있다. 이러한 RBP 복호 알고리즘의 문제점들을 개선시킨 것이 VCRBP 복호 알고리즘이다^{11)[12]}. RBP 복호 알고리즘은 체크-변수 메시지 값의 차이로 정렬 기준값인 레지듀얼을 계산하는 반면 VCRBP 복호 알고리즘은 변수-체크 메시지 값의 차이로 레지듀얼을 계산한다. 이러한 변화로 인하여 작은 반복 복호 횟수 만으로 탁월한 성능 개선을 얻을 수 있으며, RBP 복호 알고리즘에서의 불필요한 연산을 피함으로써 복잡도를 현저하게 줄일 수 있다. 또한 node-wise VCRBP (NVCRBP) 복호 알고리즘인¹²⁾ VCRBP 복호 알고리즘에 비해 보다 낮은 복호 복잡도로 거의 근접한 성능을 보인다. 그러나 RBP 스케줄링 기법을 사용하는 모든 동적 스케줄링 복호 알고리즘들은 매 반복 복호 시 추가적으로 레지듀얼 값을 계산하고 정렬해야하므로 표준 BP 복호 알고리즘에 비하여 복호 복잡도와 실제 복호가 이루어지기 전까지의 대기시간이 증가한다.

본 논문에서는 이러한 문제점들을 최소화하기 위하여 NVCRBP 복호 알고리즘의 복잡도와 대기시간을 줄인 두 가지 새로운 기법인 강제수렴 NVCRBP (forced-convergence NVCRBP, 이하 FC-NVCRBP) 복호 알고리즘과 부호기반 NVCRBP (sign-based NVCRBP, 이하 S-NVCRBP) 복호 알고리즘을 제안한다. FC-NVCRBP 복호 알고리즘은 이미 수렴되었다고 판단되는 메시지의 업데이트를 생략함으로써 복잡도와 대기시간을 줄인다. 반면, S-NVCRBP 복호 알고리즘은 메시지의 부호에 근거한 새로운 정렬 기준값을 사용하여 복잡도와 대기시간을 줄인다.

본 논문의 II절에서 LDPC 부호를 위한 NVCRBP 복호 알고리즘을 소개하고, III절에서는 제안하는 FC-NVCRBP 복호 알고리즘과 S-NVCRBP 복호 알고리즘을 설명한다. 그리고 IV절에서는 모의실험을 통한 성능 비교를 보여주고 V절에서 결론을 맺는다.

II. LDPC 부호를 위한 NVCRBP 알고리즘

2.1 Dual-min-sum 복호 알고리즘

LDPC 부호를 위한 표준 BP 복호 알고리즘은 변수 노드와 체크노드 간의 반복적 메시지 전달을 통해 이루어진다. 체크노드 c_i 와 이웃 변수노드 v_j 에 대해 메시지 업데이트 함수는 다음과 같이 정의된다¹⁾.

$$m_{v_j \rightarrow c_i} = \sum_{c_a \in N(v_j) \setminus c_i} m_{c_a \rightarrow v_j} + C_{v_j} \quad (1)$$

$$m_{c_i \rightarrow v_j} = 2 \arctan h \left(\prod_{v_b \in N(c_i) \setminus v_j} \tanh \left(\frac{m_{v_b \rightarrow c_i}}{2} \right) \right) \quad (2)$$

여기서 r_j 를 수신 신호라고 할 때, v_j 에 대한 채널 정보를 $C_{v_j} = \log \left(\frac{p(r_j | v_j = 0)}{p(r_j | v_j = 1)} \right)$ 로 정의한다. 또한 $N(v) \setminus c$ 는 c 노드를 제외한 v 노드의 이웃 노드들의 집합을 의미하고 $N(c) \setminus v$ 는 v 노드를 제외한 c 노드의 이웃 노드들의 집합을 의미한다.

표준 BP 복호 알고리즘에서 체크-변수 메시지의 계산이 연산량의 대부분을 차지하므로 우리는 복잡도를 줄이기 위하여 다음과 같은 dual-min-sum 알고리즘¹³⁾ 사용한다. dual-min-sum 알고리즘에서 수식 (2)가 다음과 같이 대체된다 :

$$m_{c_i \rightarrow v_j} = f(m_{\min}, m_{\min}^*) \prod_{v_b \in N(c_i) \setminus v_j} \text{sign}(m_{v_b \rightarrow c_i}), \quad (3)$$

여기서

$$f(m_{\min}, m_{\min}^*) = \begin{cases} m_{\min} - \alpha & \text{if } m_{\min}^* - m_{\min} < 2, \\ m_{\min} & \text{otherwise,} \end{cases}$$

$$m_{\min} = \min_{v_b \in N(c_i) \setminus v_j} |m_{v_b \rightarrow c_i}|,$$

$$m_{\min}^* = \min_{v_b \in N(c_i) \setminus (v_j, v_b)} |m_{v_b \rightarrow c_i}|,$$

α 는 성능을 최적화 시키는 적당한 실험값이다¹⁴⁾. 본 논문의 이하 실험 결과들은 모두 위의 dual-min-sum 알고리즘을 사용하였으며 α 값을 0.5로 고정하였다.

2.2 NVCRBP 복호 알고리즘

RBP는 레지듀얼 (residual) 이라 불리는 정렬 기준 값을 가장 크게 갖는 메시지를 우선 업데이트하는 동적 스케줄링 기법이다. 메시지 m_k 에 대하여 레지듀얼은 다음과 같이 정의된다⁸⁾.

$$r(m_k) = \| m_k^* - m_k \|, \quad (4)$$

여기서 m_k 와 m_k^* 는 각각 k 번째 반복 복호에서 식 (1) 또는 (2)와 같은 메시지 업데이트 함수에 의한 업데이트 전후 메시지 값이다.

직관적으로, 업데이트 전후의 메시지 값의 차이가

0에 가깝다는 것은 그 메시지가 거의 수렴되었다고 볼 수 있다. 반대로 큰 레지듀얼 값을 갖는 메시지는 전체 부호 그래프에서 아직 수렴하지 않은 부분을 나타낸다고 볼 수 있다. 따라서 가장 큰 레지듀얼 값을 갖는 메시지를 먼저 업데이트 하는 것은 전체 부호가 더욱 빠르게 복호될 수 있도록 해 준다^[8].

RBP 기법은 LDPC 부호의 복호에 다양한 형태로 적용될 수 있다^{[9][10][11][12]}. VCRBP 복호는 이중가장 좋은 성능을 보이며 NVCRBP 복호는 보다 낮은 복잡도로 거의 근접한 성능을 보인다^{[11][12]}. 따라서 본 논문에서는 NVCRBP 복호에 대해서만 다룬다. VCRBP 복호는 선택된 에지에 대응되는 하나의 변수-체크 메시지를 업데이트 하는 반면, NVCRBP 복호는 선택된 변수노드에 대응되는 모든 변수-체크 메시지들을 동시에 업데이트 한다. 이러한 점에서 NVCRBP 복호는 SBP 복호와 유사하다. 주요 차이점은 SBP 복호는 미리 정해진 순서에 의하여 변수노드를 순차적으로 업데이트 하는 반면 NVCRBP 복호는 레지듀얼에 근거하여 다음 업데이트 될 변수노드를 동적으로 선택한다는 점이다. NVCRBP 복호 알고리즘의 자세한 과정이 알고리즘 1에 표현되어 있다.

알고리즘 1 NVCRBP 복호 알고리즘^[12]

- 1: 모든 $m_{c \rightarrow v} = 0$ 로, $m_{v \rightarrow c} = C_n$ 로 초기화
- 2: 모든 $r(m_{v \rightarrow c})$ 을 비교
- 3: 가장 큰 $r(m_{v \rightarrow c})$ 에 대응되는 $m_{v_i \rightarrow c_j}$ 를 선택
- 4: **for** 모든 $c_k \in N(v_i)$ **do**
- 5: $r(m_{v_i \rightarrow c_k}) = 0$ 로 정함
- 6: **for** 모든 $v_a \in N(c_k) \setminus v_i$ **do**
- 7: $m_{c_k \rightarrow v_a}$ 를 생성하고 전달
- 8: **for** 모든 $c_b \in N(v_a) \setminus c_j$ **do**
- 9: $m_{v_a \rightarrow c_b}$ 를 생성하고 전달
- 10: $r(m_{v_a \rightarrow c_b})$ 를 계산
- 11: **end for**
- 12: **end for**
- 13: **end for**
- 14: **if** 정지 조건이 만족되지 않으면 **then**
- 15: 2번째 줄로 돌아가
- 16: **end if**

2.3 복호 복잡도와 대기시간

지금까지의 동적 스케줄링 복호 알고리즘을 다룬

표 1. 근사 복호 복잡도

	BP	NRBP	NVCRBP
매 반복 복호 시 $m_{c \rightarrow v}$ 의 연산 횟수	1	$(d_c - 1)(d_v - 1) + 1$	$d_c - 1$
그림 1에서의 실제 사용된 값	1	$(6.33-1)(3.17-1)+1=12.57$	$6.33-1=5.33$
근사 복호 복잡도 = 최대 반복 복호 횟수 $\times m_{c \rightarrow v}$ 의 연산 횟수	100	$8 \times 12.57 \approx 100$	$19 \times 5.33 \approx 101$

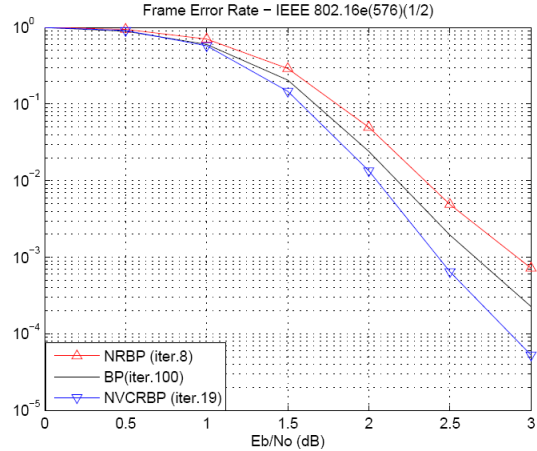


그림 1. BP, NRBP, NVCRBP 복호의 프레임 오류율 성능 비교 (각각 최대 반복횟수 100, 8, 19회 사용)

논문들은 오직 같은 반복 복호 내에서 복호 알고리즘들의 성능을 비교하였다. 그러나 RBP 복호 알고리즘과 VCRBP 복호 알고리즘은 표준 BP 복호 알고리즘에 비해 추가적으로 레지듀얼을 계산하고 그 값을 서로 비교해야 하므로 일회 반복 복호 내에서 더 많은 복잡도와 대기시간을 필요로 한다. 따라서 우리는 실제 같은 복잡도 내에서 BP, NRBP, NVCRBP 복호 알고리즘의 성능을 비교 분석하고자 한다.

표 1은 일회 반복 복호 내에서 근사 복잡도를 보여준다. LDPC 부호의 복호 복잡도의 대부분을 체크-변수 메시지 연산이 차지하므로 일회 반복 복호 내에서 체크-변수 메시지의 연산 횟수를 통해 근사 복잡도를 구한다. 여기서 d_v 와 d_c 는 각각 변수노드와 체크노드의 평균 디그리 (degree), 즉, 에지로 연결된 이웃노드의 평균 개수를 말한다. 또한 이를 통해 실제 부호의 근사 복잡도를 구하고 같은 근사 복잡도를 갖기 위한 반복 복호 횟수의 비율을 구하였다.

그림 1은 표 1에 의해 산출된 근사 복호 복잡도에 근거하여 같은 복잡도 내에서 BP, NRBP,

NVCRBP 복호 알고리즘의 성능을 비교한 것이다. 이를 통해 NRBP 복호는 같은 반복 복호 횟수에서 BP 복호 알고리즘 보다 나은 성능을 보이지만 같은 복잡도 내에서는 나은 성능을 보장하지 못함을 알 수 있다. 반면 NVCRBP 복호는 같은 반복 복호 횟수에서 뿐만 아니라 같은 복잡도 내에서도 탁월한 성능을 보임을 확인할 수 있다.

III. 감소된 복호 복잡도와 대기시간을 갖는 NVCRBP 복호 알고리즘

3.1. FC-NVCRBP 복호 알고리즘

강제수렴 기법은 매우 작은 횟수의 반복 복호 내에서 많은 수의 변수노드들이 높은 신뢰도로 수렴한다는 점을 이용하여 복호 복잡도를 낮추는 기법이다. 즉, 이러한 변수노드들은 이미 높은 신뢰도로 복호 되었으므로 그들의 메시지 값을 남은 반복 복호 내에서 업데이트하지 않는다. 이전의 논문들^{[15][16][17]}에서는 메시지 값의 크기로 수렴 여부를 판단하였다. 또한 수렴되었다고 판단된 메시지를 남은 반복 복호 내에서 특정 값으로 고정하였다. 이로 인해 만약 고정된 메시지가 오류 값을 가졌다면 전체 부호가 복호되는데 심각한 악영향을 끼치게 된다. 따라서 우리는 메시지의 수렴 판단 값으로 레지듀얼을 사용한 FC-NVCRBP 복호 알고리즘을 제안한다. 만약 어떤 메시지의 레지듀얼이 실험적 상수값 r_{th} 보다 작다면 그 메시지를 수렴되었다고 판단하고 남은 반복 복호내에서가 아닌 현재 반복 복호에서만 업데이트 및 비교 대상에서 제외시킨다. 이를 통해 불필요한 연산량과 비교 대상을 줄임으로써 복호 복잡도 뿐만 아니라 대기시간도 감소시킬 수 있다. 알고리즘 2는 FC-NVCRBP 복호의 상세과정이다.

그런데 r_{th} 는 실험적인 값이기 때문에 목표로 하는 성능을 얻기 위하여 얼마나 큰 r_{th} 를 설정해야 하는지 예측하기는 쉽지 않다. 그러므로 우리는 근사 FC-NVCRBP 복호 알고리즘을 다음과 같이 제안한다. 다수의 실험 결과에서 많은 수의 메시지들이 반복 복호 동안 전체 복호에 크게 영향을 미치지 않는 굉장히 작은 레지듀얼 값을 갖는다. 따라서 우리는 충분히 큰 레지듀얼 값을 갖는 일부의 메시지들만을 사용하여, 즉, 실제 복호에 사용되는 메시지의 양을 조절하여 더 낮은 복잡도로 전체 메시지를 사용했을 때의 성능에 거의 근접한 성능을 얻을 수

있다. 게다가 사용할 메시지의 양을 때 반복 복호 시 조절하여 성능은 유지하면서 더욱 복호 복잡도를 낮출 수 있다.

알고리즘 2 FC-NVCRBP 복호 알고리즘

- 1: 모든 $m_{c \rightarrow v} = 0$ 로, $m_{v \rightarrow c} = C_n$ 로 초기화
- 2: 모든 $r(m_{v \rightarrow c}) > r_{th}$ 을 비교
- 3: 가장 큰 $r(m_{v \rightarrow c})$ 에 대응되는 $m_{v \rightarrow c_j}$ 를 선택
- 4: **for** 모든 $c_k \in N(v_i)$; **do**
- 5: $r(m_{v_i \rightarrow c_k}) = 0$ 로 정함
- 6: **for** 모든 $v_a \in N(c_k) \setminus v_i$ **do**
- 7: $m_{c_k \rightarrow v_a}$ 를 생성하고 전달
- 8: **for** 모든 $c_b \in N(v_a) \setminus c_j$ **do**
- 9: $m_{v_a \rightarrow c_b}$ 를 생성하고 전달
- 10: $r(m_{v_a \rightarrow c_b})$ 를 계산
- 11: **end for**
- 12: **end for**
- 13: **end for**
- 14: **if** 정지 조건이 만족되지 않으면 **then**
- 15: 2번째 줄로 돌아감
- 16: **end if**

3.2 S-NVCRBP 복호 알고리즘

NVCRBP 복호와 FC-NVCRBP 복호는 정적 스케줄링 복호 방식들에 비하여 탁월한 성능을 보이지만 레지듀얼을 계산하고 그 값의 크기에 따라 정렬해야 하는 추가적인 연산이 필요하다. 따라서 이러한 단점을 최소화하기 위하여 새로운 동적 스케줄링 복호 기법인 S-NVCRBP 복호 알고리즘을 제안한다. 알고리즘을 간략화하기 위하여 다음과 같은 새로운 정렬 기준값인 $s(\cdot)$ 를 다음과 같이 정의한다 :

$$s(m_k) = \text{sign}(m_k^*) \times \text{sign}(m_k), \quad (5)$$

여기서 m_k 와 m_k^* 는 각각 k 번째 반복 복호에서 업데이트 전후의 메시지 값을 말한다. m_k 가 양수 값을 가지면 $\text{sign}(m_k) = 1$, m_k 가 음수 값을 가지면 $\text{sign}(m_k) = -1$ 로 정한다. 만약 어떤 에지에서 $s(m_k) = -1$ 이면 그 에지에 대응되는 메시지 값이 업데이트 후에 부호가 반전되었음을 의미하고 $s(m_k) = 1$ 이면 그 에지에 대응되는 메시지 값이 업데이트 후에 부호가 반전되지 않았음을 의미한다.

S-NVCRBP 복호는 $s(\cdot)$ 가 음수인 메시지를

포함하는 변수노드를 먼저 업데이트 한다. 이는 부호가 반전되는 메시지는 높은 확률로 큰 레지듀얼을 갖는다는 점으로 미루어 볼 때 타당하다. 만일 더 이상 $s(\cdot)$ 가 음수인 메시지가 없으면 미리 정해진 순서에 의해 나머지 변수노드들을 업데이트 한다. 본 논문에서는 미리 정해진 순서로 변수노드 그리의 내림차순 정렬을 사용하였다. 따라서 일회 반복 복호 내에 모든 변수노드들이 한 번씩 업데이트 된다. S-NVCRBP 복호 알고리즘의 상세과정이 알고리즘 3에 표현되어있다.

알고리즘 3 S-NVCRBP 복호 알고리즘

- 1: 모든 $m_{c \rightarrow v} = 0$ 로, $m_{v \rightarrow c} = C_n$ 로 초기화
- 2: $s(m_{v \rightarrow c}) = -1$ 이 되는 $m_{v \rightarrow c_j}$ 를 선택
- 3: **for** 모든 $c_k \in N(v_i)$ **do**
- 4: $s(m_{v_i \rightarrow c_k}) = 0$ 로 정함
- 5: **for** 모든 $v_a \in N(c_k) \setminus v_i$ **do**
- 6: $m_{c_k \rightarrow v_a}$ 를 생성하고 전달
- 7: **for** 모든 $c_b \in N(v_a) \setminus c_j$ **do**
- 8: $m_{v_a \rightarrow c_b}$ 를 생성하고 전달
- 9: $s(m_{v_a \rightarrow c_b})$ 를 계산
- 10: **end for**
- 11: **end for**
- 12: **end for**
- 13: **if** 정지 조건이 만족되지 않으면 **then**
- 14: 2번째 줄로 돌아감
- 15: **end if**

S-NVCRBP 복호는 모든 레지듀얼 값을 계산할 필요가 없었던래 레지듀얼 값을 서로 비교하지 않아도 된다. 그러므로 S-NVCRBP 복호는 NVCRBP 복호의 탁월한 성능은 유지하면서 복잡도와 대기시간을 현저히 감소시킬 수 있다.

IV. 실험결과

모의실험은 IEEE 802.16e 표준^[18]에 설계된 다양한 LDPC 부호를 사용하였다. AWGN 채널을 가정하고, 부호길이는 576, 1152, 2304 그리고 부호율은 1/2, 3/4 각각에 대하여 실험하였다. 실험결과 모든 부호에서 유사한 형태를 보이므로 부호길이 576, 부호율 1/2의 결과만 보이도록 하겠다. 또한 FC-NVCRBP 복호는 앞에서 정의한 근사 FC-NVCRBP 복호를 적용하였으며 매 반복 복호시 전체 노드의 절반만을 사용하였다.

그림 2에서 최대 8번 반복 복호 하에서 FC-NVCRBP 복호가 NRBP 복호보다 절반 이하의 낮은 복잡도와 대기시간으로 더 나은 성능을 보인다. 또한 S-NVCRBP 복호 역시 복잡도와 대기시간을 현저히 줄이면서 NVCRBP 복호에 매우 근접한 성능을 보인다.

그림 3은 신호대 잡음비 (SNR) 을 2.5dB로 고정하고 반복 복호 횟수를 최대 50회까지 증가시키면서 오류 정정 성능을 확인한 것이다. 이를 통하여 FC-NVCRBP 복호와 S-NVCRBP 복호는 작은 반복 복호 횟수에서 뿐만 아니라 많은 반복 복호 후에도 NVCRBP 복호에 비하여 낮은 복잡도와 대기시간으로 근접한 성능을 보임을 확인할 수 있다.

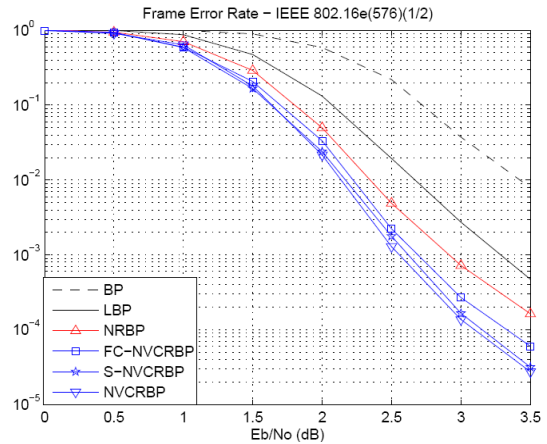


그림 2. BP, LBP, NRBP, FC-NVCRBP (1/2), S-NVCRBP, NVCRBP 복호의 SNR에 따른 프레임 오류율 성능 비교 (최대 반복횟수 8회 사용)

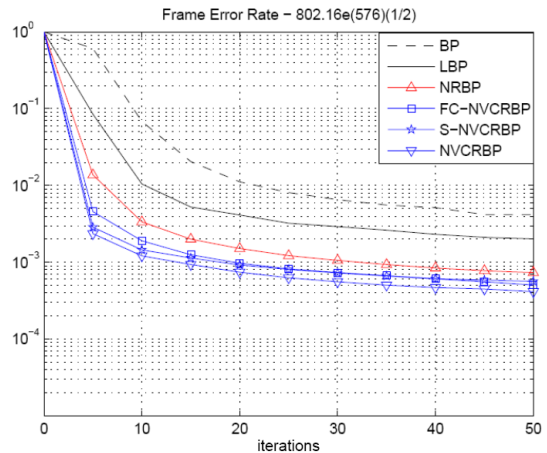


그림 3. BP, LBP, NRBP, FC-NVCRBP (1/2), S-NVCRBP, NVCRBP 복호의 반복 횟수에 따른 프레임 오류율 성능 비교 (SNR 2.5dB로 고정)

V. 결 론

NVCRBP 복호 알고리즘의 복잡도와 대기시간을 줄이기 위하여 FC-NVCRBP 복호 알고리즘과 S-NVCRBP 복호 알고리즘을 제안하였다. 실험결과 제안하는 두 알고리즘 모두 작은 반복 복호 횟수에서 뿐만 아니라 적당히 큰 반복 복호 횟수에서도 NVCRBP 복호 알고리즘 보다 낮은 복잡도와 대기시간으로 거의 근접한 성능을 보였다.

참 고 문 헌

- [1] R. G. Gallager, "Low density parity check codes," MIT press, 1963.
- [2] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," *IEEE Trans. on Comm.*, 53:209-213, February, 2005.
- [3] M. Rovini, F. Rossi, P. Ciao, N. L'Insalata, and L. Fanucci, "Layered Decoding of Non-Layered LDPC Codes," In Proc. 9th EUROMICRO Conference on Digital System Design, pages 537-544, August, 2006.
- [4] Zhang, J., Wang, Y., Fossorier, M., and Yedidia, J.S., "Replica Shuffled Iterative Decoding," *IEEE ISIT 2005*, pages 454-458, September, 2005.
- [5] Junsheng Han, Siegel, P. H., Lee P., Lin jia-ru, "improvement of Shuffled Iterative Decoding," *IEEE ITW 2006*, Chengdu, China, October, 2006.
- [6] Zhiyong He, Sebastien Roy, and Paul Fortier, "Lowering error floor of LDPC codes using a joint row-column decoding algorithm", *ICC 2007*, Glasgow, Ecosse, 24-28 June, 2007.
- [7] Oren Golov and Ofer Amrani, "Edge-based Scheduled BP in LDPC Codes," *ISIT2007*, Nice, France, 24-29 June, 2007
- [8] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: informed scheduling for asynchronous message passing," In Proc. 22nd Conference on Uncertainty in Artificial Intelligence, MIT, Cambridge, MA, July, 2006.
- [9] A. I. Vila Casado, M. Griot, and R. D. Wesel, "Informed Dynamic Scheduling for Belief-Propagation Decoding of LDPC Codes," In Proc. *IEEE ICC 2007*, Glasgow, Scotland, June, 2007.

- [10] A. I. Vila Casado, M. Griot, and R. D. Wesel, "Improving LDPC Decoders via Informed Dynamic Scheduling," *IEEE Information Theory Workshop 2007*, Lake Tahoe, CA, September, 2007.
- [11] Jung-Hyun Kim, Mi-Young Nam, and Hong-Yeop Song, "Variable-to-Check Residual Belief Propagation for LDPC Codes," *IEE Electronic Letters*, vol. 45, no. 2, pages 117-118, January 2009.
- [12] Jung-Hyun Kim, Mi-Young Nam and Hong-Yeop Song, "Variable-to-Check Residual Belief Propagation for Informed Dynamic Scheduling of LDPC Codes," *ISITA 2008*, The Langham Hotel, Auckland, New Zealand, December 7-10, 2008.
- [13] J. Li and X. D. Zhang, "Reduced-complexity belief propagation decoding for low-density parity-check codes," *IEE Electronic Letters*, vol. 44, no. 3, pages 220-222, January 2008.
- [14] Eleftheriou, E., Mittelholzer, T., and Dholakia, A., "Reduced-complexity decoding algorithm for low-density parity-check codes," *IEE Electronic Letters*, vol. 37, no. 2, pages 102-104, January 2001.
- [15] E. Zimmermann, P. Pattisapu, P. K. Bora, and G. Fettweis, "Reduced Complexity LDPC Decoding using Forced Convergence," in *Proceedings of the 7th International Symposium on Wireless Personal Multimedia Communications (WPMC04)*, Abano Terme, Italy, pages 12-15, September, 2004.
- [16] E. Zimmermann, W. Rave, and G. Fettweis, "Forced Convergence Decoding of LDPC Codes - EXIT Chart Analysis and Combination with Node Complexity Reduction Techniques," in *Proceedings of the 11th European Wireless Conference (EW'2005)*, Nicosia, Cyprus, pages 11-13, April, 2005.
- [17] E. Zimmermann, P. Pattisapu, and G. Fettweis, "Bit-Flipping Post-Processing for Forced Convergence Decoding of LDPC Codes," in *Proceedings of the 13th European Signal Processing Conference (EUSIPCO'05)*, Antalya, Turkey, pages 04-08, September, 2005.
- [18] IEEE C802.16e-05/0066r3, "LDPC coding for OFDMA PHY".

