

DVB-S2 시스템을 위한 저복잡도 LDPC 복호 알고리즘

Low Computational Complexity LDPC Decoding Algorithms for DVB-S2 Systems

정 지원

Ji-Won Jung

요 약

본 연구에서는 DTV, HDTV 서비스를 동시에 제공하는 차세대 위성 방송시스템의 표준안인 DVB-S2에서 채널 부호화 알고리즘으로 채택한 LDPC 부호의 복호 알고리즘에 대해 연구를 하였다. 사논의 한계에 근접하기 위해서는 큰 블록 사이즈의 LDPC 부호어 길이와 많은 반복 횟수를 요구한다. 이는 많은 계산량을 요구하며, 그리고 이에 따른 전력 소비량(power consumption)을 야기시키므로 본 논문에서는 세 가지 형태의 low complexity LDPC 복호 알고리즘을 제시한다. 첫째로 큰 블록 사이즈와 많은 반복 회수는 많은 계산량과 power 소모량을 요구하므로 성능 손실 없이 반복 횟수를 줄일 수 있는 SUBSET 방법을 이용한 복호 알고리즘, 둘째로 early stop 알고리즘에 대해 연구하였고, 셋째로 비트 노드 계산과 체크 노드 계산시 일정한 신뢰도 값보다 크면 다음 반복 시 계산을 하지 않는 early detection 알고리즘에 대해 연구하였다.

Abstract

In this paper, we first review LDPC codes in general and a belief propagation algorithm that works in logarithm domain. LDPC codes, which is chosen for second generation digital video broadcasting standard, are required a large number of computation due to large size of coded block and iteration. Therefore, we presented two kinds of low computational algorithm for LDPC codes. First, sequential decoding with partial group is proposed. It has same H/W complexity, and fewer number of iteration's are required at same performance in comparison with conventional decoder algorithm. Secondly, early detection method for reducing the computational complexity is proposed. Using a confidence criterion, some bit nodes and check node edges are detected early on during decoding. Through the simulation, we knew that the iteration number are reduced by half using subset algorithm and computational complexity of early detected method is about 50 % offs in case of check node update, 99 % offs in case of check node update compared to conventional scheme.

Key words : Low Density Parity Check Code, DVB-S2 Standard, Sequential Decoding, Early Detection Algorithm

I. 서 론

신뢰성이 낮은 광대역 위성방송 채널에서 다채널 및 고품질의 서비스를 제공하기 위해서는 채널 상태에 대한 강력한 오류 제어 방식을 사용해야 한다. 이를 위한 오류 정정 방식으로 실제적인 복호 알고리

즘에서 사논의 채널 용량 한계에 근접한 LDPC(Low Density Parity Check) 코드 방식이 관심의 대상이 되어지고 있다. LDPC 부호는 터보 부호에 비해 복호화의 복잡도가 낮은 뿐 아니라 좋은 거리 특성으로 오류 마루 현상이 나타나지 않고, 완전 병렬 처리로 고속 처리가 가능한 장점이 있다. 실제로 위성

한국해양대학교 전파공학과(Department of Radio Science & Engineering, Korea Maritime University)

· 논문 번호 : 20050510-054

· 수정완료일자 : 2005년 9월 26일

크기 $c = (i_0, i_1, \dots, i_{K-1}, p_0, p_1, \dots, p_{M-1})$ 에 정보 블록크기 $(i_0, i_1, \dots, i_{K-1})$ 를 부호화 한다. 여기서 N 은 부호화의 크기이며, K 는 정보원의 크기이다. M 은 잉여비트의 크기이며 $N-K$ 개를 가진다. 부호화하는 절차는 아래와 같다.

[단계 1] 모든 parity bit를 초기화 시킨다.

$$p_0 = p_1 = \dots = p_M = 0$$

[단계 2] 모든 패리티 비트 주소를 더한다.

$$P_{M_Address} = P_{M_Address} \oplus i_M$$

[단계 3] B submatrix인 low triangular parity check matrix를 구하기 위해 마지막으로 다음과 같은 공식을 적용시킨다.

$$p_n = p_n + p_{n-1} (n=1, 2, \dots)$$

단계 2에서 $M_Address$ 는 검사행렬 H 에서 각 K 개의 정보비트별 '1'의 위치를 저장해 놓은 어드레스를 나타내며, 이는 참고문헌 [1]에 나타내었다.

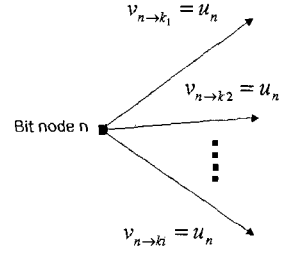
2-2 LDPC 복호화 알고리즘

LDPC 복호기는 전송되어진 심볼을 비트 노드와 체크 노드에서 각각의 확률값을 구하여 반복을 통해 전송되어진 비트를 결정하는 것이다. DVB-S2에 제시한 복호 알고리즘은 기존의 Log Domain에서의 복호 알고리즘과 거의 동일하나, LUT를 만드는 방법 및 check node 확률을 구하는 방법상에서 약간의 차이를 가진다. LDPC의 복호 순서는 다음 그림 2와 같이 크게 세 가지로 나눌 수 있다. 수신비트에다가 채널 추정 값을 구하는 초기화 과정, check node 확률을 구하는 CNU(Check Node Update), 비트 확률을 구하는 BNU(Bit Node Update)로 세 가지 단계로 나눌 수 있다.

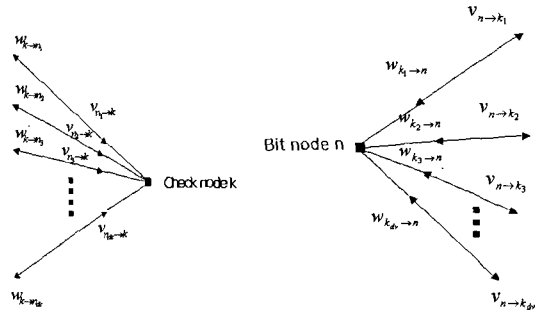
단계 1. 초기화 과정

채널이 AGWN이고 ISI memory less 채널이라고 하면 channel transmit 확률은 다음과 같이 나타낼 수 있다.

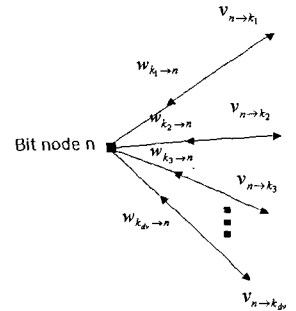
$$P(r | x_n = b) = P(r_n = b | x_n = b) \quad (1)$$



(a) 초기화 과정
(a) Initialize process



(b) CNU 계산 과정
(b) CNU calculating process



(c) BNU 계산 과정
(c) BNU calculating process

그림 2. LDPC 복호화 과정

Fig. 2. Decoding process of LDPC.

여기서, r_n 은 received vector r 의 특정 위치 n 에 위치한 원소를 뜻한다. $x_n=0, 1$ 일 때 각각 1, +1로 매핑하는 BPSK 변조를 한다면 $P(r_n=b | x_n=b)$ 은 가우시안 PDF를 이용해 다음 식과 같이 나타낼 수 있다.

$$f_n^b = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(r_n + (-1)^b)^2}{2\sigma^2}\right)$$

이를 구현을 위해 log-domain으로 바꾸면 다음과 같다.

$$u_n = -L_c \cdot r_n \left(L_c = \frac{2}{\sigma^2} \right), n = (0, 1, \dots, N-1) \quad (2)$$

단계 2. Check Node Update(CNU)^[1]

다음 그림 2는 check node 확률을 구하는 CNU에서 하나의 체크 노드에서 올 수 있는 비트들의 확률을 구하는 그림이다. n_{dc} 개의 row weight를 가진다고 가정할 때, 각각의 체크 노드로 들어오는 비트들의 확률 $w_{k \to n_1}, w_{k \to n_2}, \dots, w_{k \to n_{dc}}$ 은 아래 식과 같이 유

도된다.

$$w_{k \rightarrow n_i} = g(v_{n_1 \rightarrow k}, v_{n_2 \rightarrow k}, \dots, v_{n_{i-1} \rightarrow k}, v_{n_{i+1} \rightarrow k}, \dots, v_{n_{d_c} \rightarrow k})$$

$$g(a, b) = \text{sign}(a) \times \text{sign}(b) \times \{\min |a|, |b|\}$$

$$+ LUT_g(a, b)$$

$$LUT_g(a, b) = \log(1 + e^{-|a+b|}) - \log(1 + e^{-|a-b|}) \quad (3)$$

실제로 $LUT_g(\cdot)$ 는 look up table로 구현될 수 있다.

단계 3. Bit Node Update(BNU)

CNU에서 각 체크 노드로 연결된 비트들에 대한 업데이트가 이루어진 후, 각 비트노드에 연결된 d_c 개의 확률, 즉, 각 컬럼에 해당하는 d_c 개의 비트 노드, $v_{n \rightarrow k_1}, v_{n \rightarrow k_2}, \dots, v_{n \rightarrow k_{d_c}}$ 의 확률을 다음 식에서 구한다.

$$v_{n \rightarrow k_i} = u_n + \sum_{j \neq i} w_{k_j \rightarrow n}$$

2-3 DVB-S2 시스템에서 제공하는 부호율에 대한 성능 분석

DVB-S2 표준안 기반의 시스템은 대역폭 효율이 0.5 bits/symbol부터 4.5 bits/symbol까지 지원하며, 10개의 서로 다른 부호화를 1/4, 1/3, 1/2, 3/5, 2/3, 3/4, 4/5, 8/9 그리고 9/10를 지원하며, 각각의 부호화율에서는 네 개의 서로 다른 변조 방식 QPSK, 8-PSK, 16-APSK 그리고 32-APSK를 지원한다. 그리고 성능 향상을 위해 irregular LDPC 부호를 사용한다. DVB-S2에서 제공하는 각 부호화에 따른 부호화어 크기 N , 정보원의 크기 K 그리고 각 행과 열에서 '1'의 개수를 나타내는 row weight와 column weight의 값수는 아래 표 1과 같다. 그림 3은 가우시안 잡음에서 부호화 블록 사이즈 N 이 64800이고 변조 방식이 QPSK일 때 성능 그래프이다.

III. Low Complexity Algorithm

3-1 Sequential Decoding with Partial Group

앞장에서 살펴보았듯이 일반적인 LDPC decoder는 bit to check block과 check to bit block으로 구성된

표 1. DVB-S2에서 제공하는 각 부호화율에 따른 파라메타

Table 1. Parameters of various coding rates provided by DVS-S2 specification.

Coding rate R	Coded word N	Information size K	Row weight d_c	Column weight d_r
1/ 2	64800	32400	8	7
2/ 3	64800	43200	13	10
3/ 4	64800	48600	12	14
4/ 5	64800	51840	11	18
5/ 6	64800	54000	13	22
7/ 8	64800	57960	4	24
8/ 9	64800	57600	4	27
9/10	64800	58320	4	30

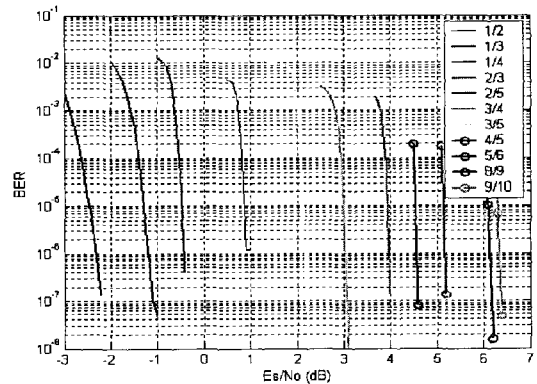


그림 3. 다양한 부호화율에서 BER 성능 곡선(반복 횟수=40, N=64800, QPSK 변조 방식)

Fig. 3. BER performance curves for various coding rates(iteration=40, N=64800, QPSK modulation scheme, Gaussian noise).

다. Bit to check block 블록은 동시에 병렬로 계산된다. 허나 그림 4에서와 같이 sequential decoding algorithm을 이용함으로써 동일한 성능에서 계산량의 감소라는 두 가지 이득을 얻을 수 있다.

먼저 check node를 p 개의 부분집합으로 나눈다. 각각의 부분집합은 n 개의 check node를 가지고 있다. $P=1$ 이면 부분집합으로 나누지 않은 기존의 복호 방식과 동일하다. 이때 총 check node의 수 $M=n \times p$ 이다. 그 다음 첫 번째 부분 집합의 check node를 업데이트 한다. 그 다음 업데이트된 check node를 통해서 bit node를 업데이트 하며 그 뒤 이때의 bit node를 통하여 다른 부분집합의 check node를 업데이트 하게

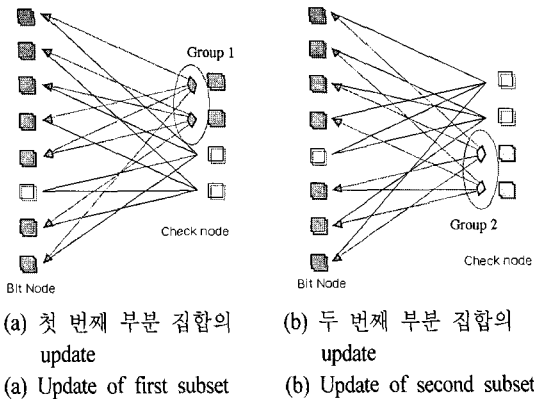


그림 4. Sequential 복호 과정
Fig. 4. Sequential decoding process.

된다. 이때 부분집합을 나눌 때 각각의 check node 들과 bit node들과의 edge를 광범위하게 분포하는 것이 중요하다.

직렬로 연결된 구조에서 첫 번째 복호기가 복호를 한 후 외부 정보(extrinsic information)를 두 번째 복호기에 전달되면서 복호 과정이 진행된다. 마찬가지로 sequential decoding 과정 또한 p 개의 복호 모듈로 나뉘어 각각의 복호를 마친 후(bit node, check node), 다음 복호 모듈로 전달된다. 이는 이미 bit node update 결과 후 다음의 부분집합으로 전달하기 때문에 기존의 방식에 비해 성능열화 없이 같은 복호 시간에 $N/2$ 만큼의 반복 회수만 필요하다(N : 기존 방식의 반복 횟수).

다음 그림 5는 부분집합(subset=2) 경우의 sequential decoder와 일반적인 decoder와의 BER 성능 곡선이다. Sequential decoding algorithm을 사용하였을 경우 2-3절의 부호화율 1/2에서 가우시안 잡음 채널 환경 하에서 시뮬레이션 한 그림이다. 기존 방식의 70 회의 반복 횟수와 sequential decoding algorithm 적용 시 35회의 반복시 성능이 일치함을 알 수 있다.

3-2 Early-stop Algorithm

LDPC와 같은 반복 부호는 일정한 반복 후에는 더 이상 성능이 개선되지 않는 현상이 있으며, 이는 복호속도 저하 및 power 소모량만 증가시키고는 결과만 맞고 있다. 따라서 적절한 early-stop 알고리즘을 이용하여 더 이상의 성능 개선이 없으면 반복을 중지시키는 알고리즘을 그림 6과 같이 구성해야 한다.

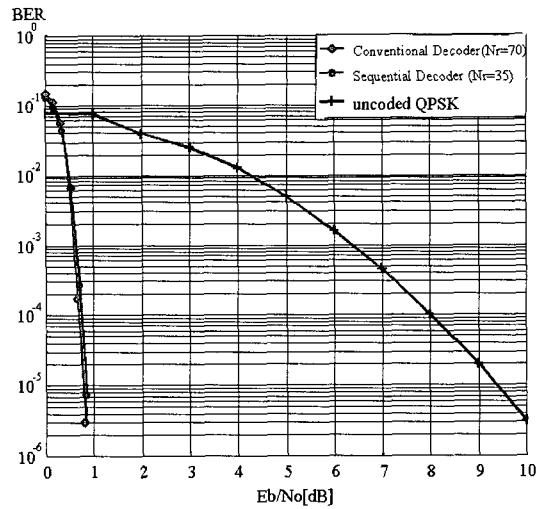


그림 5. Sequential decoder와 기존 복호기의 성능 비교
Fig. 5. Performance comparison of sequential decoder and conventional decode.

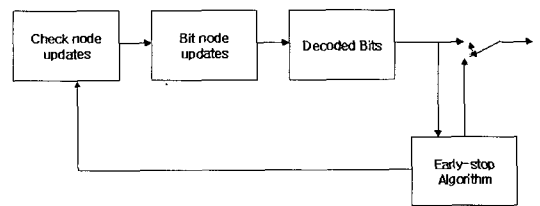


그림 6. Early-stop 알고리즘 블록도
Fig. 6. Block diagram of early-stop algorithm.

본 논문에서는 두 가지 알고리즘을 제시한다. 각 iteration 순간에 나온 LLR 값을 hard decision 하여 비교하는 방식과 parity check equation을 이용하는 방식이다.

3-2-1 Hard Decision Aided Algorithm(HDA)

각 부호화율에 따른 iteration을 마치면 LLR을 hard decision하여 decoded bit를 판정한다. Iteration을 거치면서 정정된 bit가 있다면 LLR의 비트에 변화가 있을 것이고 더 이상 정정할 것이 없다면 iteration이 계속되더라도 더 이상의 변화는 없을 것이다. 따라서 각 iteration일 때의 LLR을 HD하여 그 다음 iteration과 비교 더 이상의 변화가 없을 경우 decoding을 종료하고 최종적으로 나오는 LLR을 이용하여 디코딩한다.

3-2-2 Parity Check Equation Algorithm(PCE)

표 2. E_b/N_0 에 따른 평균 반복 횟수와 복호 속도 증가율

Table 2. Average iteration number and decoding speed improvement rate according to E_b/N_0 .

E_b/N_0	HDC		PCE	
	Iteration	복호속도 향상도	Iteration	복호속도 향상도
0.8	49.52	17.46 %	48.56	19.07 %
0.825	45.88	23.53 %	45.04	24.93 %
0.85	43.11	28.15 %	42.15	29.75 %
0.875	40.53	32.45 %	38.86	33.57 %
0.9	38.53	35.78 %	37.79	37.07 %

Bit node 업데이트 후 각 비트 노드에서 체크 노드 로가는 값과 체크 노드에서 그 비트 노드로 가는 값의 합을 hard decision 한다. 구해진 N 개의 값 중 정보 비트 부분(K 개)을 다시 reencoding 하면 새로운 parity bit(M 개)가 생성된다. 이 parity bit와 hard decision으로 구해진 parity bit를 비교하여 같다면 iteration을 중지한다.

위의 표 2는 두 가지 방식의 early stop algorithm을 적용하였을 시 iteration 수이며 그에 따른 속도의 증가량을 나타내고 있다. Parity check equation 방식을 사용하였을 때 약 1회 iteration을 적게함을 알 수 있다. 그림 7에서 보듯이 iteration 60회한 BER 성능과

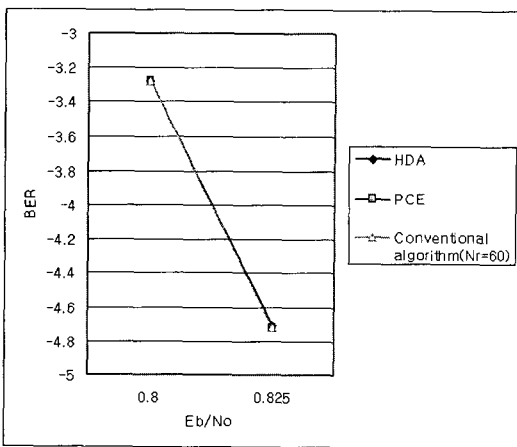


그림 7. Early stop algorithm의 BER 성능곡선($N=64800$, $R=1/2$, irregular LDPC, 가우시안 잡음)

Fig. 7. BER performance curves of early stop algorithm($N=64800$, $R=1/2$, irregular LDPC, Gaussian noise).

위의 2가지 알고리즘을 적용하였을 때의 성능 곡선이 거의 일치함을 알 수 있다.

3-3 Early Detection Method

Decoding 과정의 복잡한 계산량을 줄이는 또 다른 방법은 early detection이다. 이 early detection 방식은 높은 log likelihood 값을 지닌 bit node나 check node는 신뢰할 수 있다는 개념을 기반으로 한다. 따라서 detected된 bit node나 check node는 다음 iteration에서 계산 과정을 거치지 않는다. LDPC decoder에 사용되는 종전의 sum product algorithm은 뛰어난 BER 성능을 가지나 많은 계산 과정이 필요하다. 따라서 계산량을 줄일 수 있는 방법이 필요하게 된다. Bit node나 check node를 업데이트 할 때 일정한 기준 값을 넘게 되면 그 값은 더 이상 iteration 이후의 업데이트를 하지 않더라도 신뢰할만한 값으로 사용 가능하게 된다.

Step 1. Initialization

모든 detected check node, bit node를 0으로 초기화 시킨다.

Step 2. Check Node Updates

Check node의 값이 일정한 기준 값(T_c) 이상일 때 detected check를 1로 설정하고 그렇지 않을 경우 check node update 공식에 따라 계산한다.

Step 3. Bit Node Updates

Bit node의 값이 일정한 기준 값(T_b) 이상일 때 detected node를 1로 설정하고 그렇지 않을 경우 bit node update 공식에 따라 계산한다.

그림 8에서 $T_c=10$ 으로 고정시키고 T_b 값을 변화시키면서 BER 성능을 나타낸 그림이다. $T_b=18$ 때 일반적인 LDPC decoder와 성능이 거의 일치함을 알 수 있다.

표 3은 LDPC decoder에서 요구되는 계산량의 나열한 것이다.

여기서 d_c^* 는 각 check node당 early detected한 edge의 평균 숫자이고, N^* 는 bit node를 early detected한 평균 숫자이다.

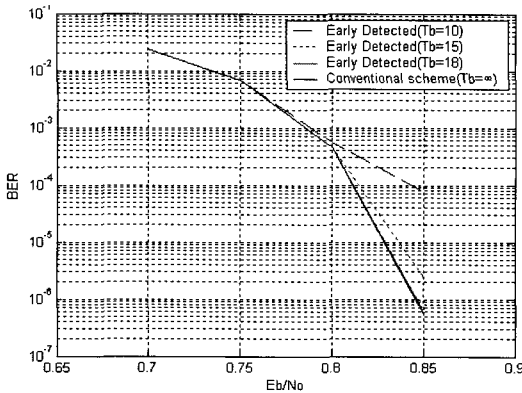


그림 8. $T_c=10$ 으로 고정시 T_b 값 변화에 따른 BER 성능곡선($N=64800$, $R=1/2$, irregular LDPC, 가우시안 잡음)

Fig. 8. Fixed on $T_c=10$, BER performance curves for variation of value of T_b ($N=64800$, $R=1/2$, irregular LDPC, Gaussian noise).

표 3. LDPC decoder에 요구된 계산량

Table 3. Required number of computation for LDPC decoder.

	Conventional decoder	Early detection method
Check nodes($g(a, b)$)	$(M \times (d_c - 1)) \times N_r$	$(M \times (d_c - d_c^* - 1)) \times N_r$
Bit nodes(additions)	$(N \times (d_v - 1)) \times N_r$	$((N - N^*) \times d_v) \times N_r$

표 4. Early detected 된 수

Table 4. The number of early detected.

	N_r					
	10	20	30	40	50	60
d_c^*/M	0.387	1.484	2.619	3.29	3.6	3.8
N^*	491	19048	62372	64334	64335	64335

$T_b=18$, $T_c=10$ 으로 고정하였을 때 표 4는 E_b/N_0 가 1 dB일 때 각 iteration 횟수에 따른 d_c^* 와 N^* 의 숫자를 보여준다. 이 때 $N=64800$, $K=32400$, $M=32400$ 이며 $d_c=7$, $d_v=13$ 이다. 표 3에서 알 수 있듯이 각 check node당 7개의 edge가 있는데 $N_r=60$ 회일 때 약 3.8개를 줄일 수 있으므로 check node update 계산시 약 50% 이상의 계산량을 감소할 수 있으며, 비트 노드는 $N_r=60$ 일 때 기존 방식을 64800이나 early detect한 비

트노드는 64355이므로 약 95% 이상의 계산량을 감소할 수 있다.

IV. 결 론

반복 부호에서는 사noise의 한계에 접근하기 위해서는 큰 블록 사이즈와 많은 반복 횟수를 필요하며, 이는 많은 계산량을 요구하므로 성능 손실 없이 반복 횟수를 줄일 수 있는 알고리즘을 제시하였다. 반복 횟수를 기존의 방식에 비해 성능 열화 없이 줄이는 방법은 SUBSET 방법을 이용한 복호 알고리즘, 그리고 early stop 알고리즘에 대해 연구하였고, 그리고 각각의 반복시 계산량을 줄이기 위해, 비트 노드 계산과 체크 노드 계산시 일정한 신뢰도 값보다 크면 다음 반복시 계산을 하지 않는 early detection 알고리즘에 대해 연구하였다. DVB-S2 규격에 제시된 $N=64800$, $K=32400$, $d_c=7$, $d_v=13$, $R=1/2$ 부호에서 시뮬레이션한 결과 subset을 이용하여 $p=2$ 일 때 성능 열화가 거의 없이 반복 회수를 절반으로 줄일 수 있었으며, early stop 적용 결과, 약 30~40% 정도의 복호 속도 향상을 보였음을 알 수 있다. 계산량을 줄일 수 있는 early detect 방식을 적용시킨 결과 check node 계산량은 약 50%, bit node 계산량은 약 95% 이상 감소함을 알 수 있다. 본 논문의 결과를 토대로 향후 DVB-S2 기반의 LDPC 복호 알고리즘에 본 논문에서 제시한 계산량 및 복잡도 감소 알고리즘을 적용 가능하리라 사료된다.

참 고 문 헌

- [1] "Satellite broadcasting system of integrated service digital broadcasting", ITU-R BO, 1227-2.
- [2] "Digital video broadcasting(DVB): Framing structure, channel coding and modulation for digital satellite news gathering(DSNG) and other contribution applications by satellite", ETSI EN 301 210: European Standard.
- [3] R. G. Gallager, "Low-density parity-check codes", IRE Trans., Information Theory, vol. 8, pp. 21-28, 1962.
- [4] D. J. C. Mackay, R. M. Neal, "Near shannon limit performance of low-density parity-check codes",

Electron. Letter, vol. 32, pp. 1710-1722, Aug. 1996.

[5] T. Richardson, R. Urbanke, "Efficient encoding of low-density parity-check codes", *IEEE Trans. Information Theory*, vol. 47, pp. 638-656, Feb. 2001.

[6] Brendan J. Frey, Frank R. Kschischang, "Early detection and trellis splicing: reduced-complexity iterative decoding", *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, Feb. 1998.

[7] 김성환 외 2인, "체크노드 부분집합에 대한 순차적 복호를 이용한 LDPC 복호 방법", JCCI2004, 2004년.

[8] E. Eleftheriou, T. Mittelholzer, and A.Dholakia, "Reduced complexity iterative decoding of low density parity check codes based on belief propagation", *IEE Electron Letter*, vol. 37, pp. 102-104, Jan. 2001.

정 지 원



1989년 2월: 성균관대학교 전자공학과 (공학사)
 1991년 2월: 성균관대학교 전자공학과 (공학석사)
 1995년 2월: 성균관대학교 정보공학과 (공학박사)
 1991년 1월~1992년 2월: LG 정보

통신연구소 연구원

1995년 9월~1996년 8월: 한국통신 위성통신연구실 선임 연구원

1997년 3월~1998년 12월: 한국전자통신연구원 초빙 연구원

1996년 9월~현재: 한국해양대학교 전파공학과 부교수

2001년 8월~2002년 8월: 캐나다 NSERC Fellowship(Communication Research Center 근무)

[주 관심분야] 위성통신, 이동통신, 변·복조기술, 채널코딩, FPGA 기술 등