

분산 동영상 부호화 시스템을 위한 LDPC 부호 설계 및 성능 평가

준회원 노 현 우*, 종신회원 이 창 우**

LDPC Code Design and Performance Analysis for Distributed Video Coding System

Hyeun-woo Noh* Associate Member, Chang-woo Lee** Lifelong Member

요 약

LDPC(low density parity check) 부호는 낮은 복잡성과 Shannon의 한계에 근접하는 오류 정정 능력을 보이기 때문에 turbo 부호와 함께 많은 응용 분야에 적용되고 있다. 최근에는 분산 동영상 부호화(distributed video coding: DVC) 시스템의 Wyner-Ziv 프레임 복호를 위해서도 LDPC 부호가 많이 사용되고 있다. 본 논문에서는 DVC 시스템을 위한 LDPC 부호를 설계하기 위해 패리티 체크 행렬 H를 설계하고 부호율 적응적인(rate adaptive) 특성을 만족하기 위해 H 행렬의 패리티 점검 노드를 효율적으로 병합하는 방법을 제안한다. 이를 위해 cycle의 연결성을 고려한 ACE(approximation cycles EMD) 알고리즘을 기반으로 효율적인 LDPC 부호를 설계하고 부호율 적응적인 특성을 갖도록 하기 위해 H 행렬의 크기와 압축율을 고려하여 병합 범위를 지정하고 지정된 범위에 따라 패리티 점검 노드를 병합한다. 그리고 ACE 알고리즘의 계수와 차수 분포를 변화시키면서 설계한 LDPC 부호의 성능을 해석한다.

Key Words : Low density parity check(LDPC) code, Distributed video coding(DVC), Rate adaptivity, ACE(approximation cycles EMD) algorithm, Parity check

ABSTRACT

Low density parity check (LDPC) code is widely used, since it shows superior performance close to Shannon limit and its decoding complexity is lower than turbo code. Recently, it is used as a channel code to decode Wyner-Ziv frames in distributed video coding (DVC) system. In this paper, we propose an efficient method to design the parity check matrix H of LDPC codes. In order to apply LDPC code to DVC system, the LDPC code should have rate compatibility. Thus, we also propose a method to merge check nodes of LDPC code to attain the rate compatibility. LDPC code is designed using ACE algorithm and check nodes are merged for a given code rate to maximize the error correction capability. The performance of the designed LDPC code is analyzed extensively by computer simulations.

I. 서 론

Gallager에 의해 처음 제안된 LDPC 부호는 패리티 검사 행렬의 원소가 대부분 0인 선형 블록 부호(linear

※ 이 논문은 2011년도 가톨릭대학교 교비 연구비 지원 및 2011년도 정부의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 임 (No.2011-0005202)

* 가톨릭대학교 정보통신전자공학과(munlan4u@naver.com, changwoo@catholic.ac.kr), (°: 교신저자)
논문번호 : KICS2011-09-386, 접수일자 : 2011년 9월 5일, 최종논문접수일자 : 2011년 12월 14일

block code)로서 turbo 부호와 마찬가지로 Shannon의 채널 용량 한계에 근접하는 성능을 보이지만 당시에는 구현이 불가능한 복잡도로 인해 오랫동안 사용되지 않았다[1]. 1993년 Berrou 등에 의해 turbo 부호가 제안된 이후에 LDPC 부호는 MacKay와 Neal에 의해 재발견되었으며 최근에는 DVC 시스템의 Wyner-Ziv 프레임 복호를 위해서도 turbo 부호와 더불어 LDPC 부호가 많이 사용된다^[2,3].

DVC 시스템은 상관관계를 갖고 있는 신호들을 독립적으로 부호화하고 복호기에서 신호들의 상관성을 이용하여도 부호기에서 신호들의 상관성을 이용한 경우와 유사한 부호화 성능을 보일 수 있다는 Slepian-Wolf와 Wyner-Ziv 부호화 이론에 기반을 두고 있다[2]. DVC 시스템의 기본 원리는 그림 1과 같이 원 신호 X로부터 생성된 패리티 비트 S와 원 신호의 추정치인 부가정보 Y를 이용하여 원 신호 X를 복원하는 것이다.

원 신호 X와 부가 정보 Y의 상관 관계가 낮으면 복호화할 때 필요한 패리티 비트의 수가 늘어나기 때문에 DVC 시스템에서 사용되는 오류 정정 부호는 X와 Y의 상관 관계에 따라 패리티 비트 수가 변하는 부호율 적응적인 특성을 가져야 한다. 이를 위해 turbo 부호에서는 간단한 천공(puncturing) 방법을 제공하여 부호율 적응적인 특성을 갖게 되지만 상대적으로 LDPC 부호에서는 부호율 적응적인 부호 구현이 어렵다. 하지만 Ascenso 등이 설계한 DVC 시스템을 위한 부호율 적응적인 LDPC 부호가 좋은 성능을 보임을 입증하였다^[3].

DVC 시스템을 위한 LDPC 부호를 설계하기 위해서는 부호율 적응적인 특성을 가진 패리티 검사 행렬인 H 행렬을 설계해야 한다. 본 논문에서는 cycle의 연결성을 고려한 ACE 알고리즘[4]을 바탕으로 H 행렬을 설계한다. 특히, 부호율 적응적인 특성을 만족하기 위해서 먼저 저압축율 LDPC 부호를 생성하고 체크 노드 병합 과정에서 생기는 edge들의 중복 연결을 피하

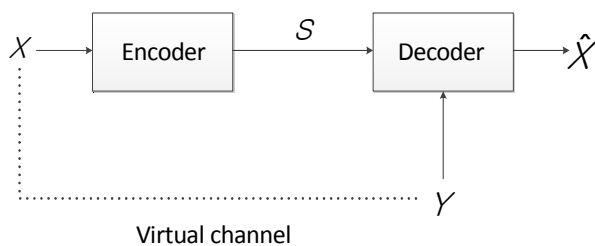


그림 1. DVC 시스템의 기본 개념
Fig. 1. Basic concept of DVC system

고 체크 노드의 차수 분포를 일정하게 유지하도록 체크 노드를 병합함으로써 부호율 적응적인 LDPC 부호의 성능을 최대화하도록 한다. 또한, H 행렬 설계 과정에서 성능 저하의 요인이 되는 4 cycle을 제거하여 성능을 향상시키도록 한다. LDPC 부호의 ACE 알고리즘의 계수와 차수 분포를 변화시키면서 LDPC 부호를 설계하고 성능을 해석한다. 본 논문의 구성은 다음과 같다. II장에서는 LDPC 부호와 ACE 알고리즘에 대하여 설명한다. III장에서는 DVC 시스템을 위한 LDPC 부호 설계 방법을 설명하고 IV장에서는 ACE 알고리즘의 계수와 차수 분포를 변화시키면서 설계한 LDPC 부호의 성능을 모의실험을 통하여 분석한 결과를 기술한다. 마지막 V장에서는 결론을 맺는다.

II. LDPC 부호

2.1 LDPC 부호 설계를 위한 ACE 알고리즘

LDPC 부호의 패리티 검사 행렬 H는 0과 1로 구성되어 있는 행렬인데 이 행렬의 원소 중에서 1의 숫자가 매우 작기 때문에 sparse한 행렬이 되므로 low density parity check 부호라 불린다. 이러한 행렬은 Tanner 그래프로 표현할 수 있는데 그림 2에 행렬 H와 대응하는 Tanner 그래프의 보기를 제시하였다.

H 행렬의 열의 개수만큼의 변수 노드와 행의 개수만큼의 체크 노드가 존재하는데 H 행렬의 (i, j)번째 원소의 값이 1이면 j번째 변수 노드와 i번째 체크 노드가 연결되어 있음을 의미한다. H 행렬의 열과 행에 존재하는 1의 개수가 일정하게 존재한다면 균일(regular) LDPC 부호라고 하며 일정하지 않게 존재한다면 비균일(irregular) LDPC 부호라고 한다. 일반적으로 비균일 LDPC 부호가 균일 LDPC 부호보다 더 좋은 에러 정정 능력을 갖고 있다. 본 논문에서는 비균일 LDPC 부호를 설계한다.

H 행렬의 크기가 $m \times n$ 일 때 n은 부호화된 전체 비트의 길이이고 m과 $n-m$ 은 각각 검사 비트의 길이와 정보 비트의 길이를 나타내며 $(n-m)/n$ 는 부호율이 된다. LDPC 부호에서는 H 행렬로부터 유도된 생성행렬 G를 이용하여 부호화 과정을 수행한다. 이때 부호화된 정보는 정보 비트와 검사 비트로 구성되어 있고 부호 워드 S라고 정의한다. 부호 워드는 $HS^T = 0$ 를 만족하게 된다. 즉, 같은 check 노드에 연결되어 들어오는 비트를 더하면 0이 된다. 복호 과정에서는 H 행렬을 이용한 SPA(sum product algorithm) 과정을 통하여 부호 워드 S는 원 신호 X로

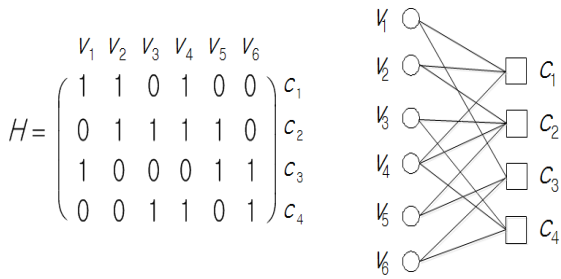


그림 2. 패리티 검사 행렬 H와 Tanner 그래프
Fig. 2. Parity check matrix and Tanner graph

복원된다.

LDPC 부호는 Shannon의 채널 용량의 한계에 근접하는 성능을 가지고 있다고 알려져 있다. LDPC 부호의 성능에 영향을 미치는 중요한 요인으로 각 노드의 차수 분포, cycle의 영향과 block size 등이 있다. 일반적으로 H 행렬 안의 짧은 cycle이 성능 저하의 요인으로 알려져 있다. Cycle은 한 노드에서 시작된 연결선이 다른 노드들을 거친 후 다시 그 노드에서 끝나게 되는 연결선의 경로를 일컫는다. 하지만 모든 짧은 cycle이 똑같이 악영향을 미치는 것은 아니다.

본 논문에서는 cycle의 연결성에 따라서 LDPC 부호의 성능을 향상 시킬 수 있는 특성을 이용한 ACE 알고리즘을 이용하여 H 행렬을 설계하였다. ACE 알고리즘은 다음과 같은 정의를 이용하여 H 행렬을 설계한다⁴⁾.

2.1.1. 정지 집합 (Stopping sets)

패리티 검사 행렬 H를 형성하는 과정에서 정지 집합은 특정 변수 노드들이 두 번 이상 체크 노드들과 연결되어 만들어 진다. 일반적으로 정지 집합은 다수의 cycle로 구성되어 있다. 하지만 차수가 2인 변수 노드는 단일 cycle를 형성한다. 예를 들면 그림 2에서 변수 노드의 부분집합 $A = \{v_3, v_5, v_6\}$ 는 체크 노드의 부분집합 $\{c_2, c_3, c_4\}$ 와 두 번씩 연결되었기 때문에 정지 집합이 된다. 또한, 차수가 2인 변수 노드로 구성되어 있기 때문에 단일 cycle을 형성한다. 따라서 정지 집합은 cycle를 포함하는 상위 개념이다. 하지만 변수 노드의 부분집합 $B = \{v_1, v_2, v_3\}$ 는 연결된 c_3 와 c_4 가 한번만 연결되었기 때문에 정지 집합이 될 수 없다. 또한, 정지 집합은 정지 집합의 각 변수 노드를 구성하는 0과 1의 합으로 결정될 수 있다. 정지 집합의 열들의 합은 1이 되지 않는다. 예를 들면 정지 집합 A의 부분집합 $\{v_3, v_5, v_6\}$ 의 합은

$[0+0+0, 1+1+0, 0+1+1, 1+0+1]^T = [0\ 2\ 2\ 2]^T$ 로 나타낼 수 있고 1 값을 갖고 있지 않는다. 따라서 A는 정지 집합이다⁵⁾.

2.1.2. EMD (Extrinsic message 차수)

변수 노드 집합과 단일 연결된 체크 노드를 외적(extrinsic) 체크 노드라고 한다. 변수 노드 집합과 연결되는 외적 체크 노드의 수를 EMD라고 한다. 따라서 정지 집합의 EMD는 0을 나타내고 변수 노드 집합의 열의 1의 개수는 EMD와 같다. 예를 들면 집합 $B = \{v_1, v_2, v_3\}$ 의 EMD 값의 합은

$[1+1+0, 0+1+1, 1+0+0, 0+0+1]^T$ 로 나타낼 수 있다. 즉, $[2\ 2\ 1\ 1]^T$ 가 된다. 따라서 B의 EMD 값은 2이다.

2.1.3. ACE (Approximate cycle EMD)

만약 cycle 안의 어떠한 변수 노드도 cycle 밖에서 공유한 체크 노드가 없다면 이 cycle의 EMD는 $\sum_i (d_i - 2)$ 이다. 2d cycle 길이의 ACE는 $\sum_i (d_i - 2)$ 이다. 이때 d_i 는 cycle 안의 i번째 변수 노드의 차수이다. 단일 cycle 안에서의 EMD 값은 ACE 값과 같지만 여러 cycle로 구성되어 있을 때는 EMD보다 큰 ACE를 갖게 된다. 예를 들면 그림 2에서 변수 노드 $\{v_2, v_3, v_4\}$ 와 연결된 체크 노드는 6 cycle을 구성하고 ACE는 1이다. 하지만 이 변수 노드의 6 cycle은 4 cycle을 포함하고 있기 때문에 단일 연결된 체크 노드가 없으므로 EMD는 0이다. 따라서 2d cycle 길이의 ACE는 $\sum_i (d_i - 2)$ 이다. 이때 d_i 는 cycle 안의 i번째 변수 노드의 차수이다. 따라서 큰 ACE 값을 가지는 변수 노드 집합은 정지 집합이 되기 위해 추가적인 노드가 필요하게 된다. 이러한 정지 집합과 ACE가 성능에 큰 영향을 미치므로 ACE 알고리즘을 이용하여 최소 정지 집합의 크기를 증가시킴으로써 더 나은 성능을 얻게 된다.

2.2. DVC 시스템을 위한 LDPC 부호의 복호화 과정

일반적인 LDPC 부호에서 부호화된 전체 비트는 정보 비트와 검사 비트로 구성되고 부호 워드는 $HS^T = 0$ 를 만족하게 되며 복호 과정에서 이런 특성을 이용하여 원 신호를 복원하게 된다. 이와 다르게 DVC 시스템을 위한 LDPC 부호의 경우에는 원 신호 X와 곱하여 부호 과정을 수행한다. 즉, $S = HX^T$ 를 구하는데 부호화된 정보는 검사 비트로만 구성되어 있고 패리티 비트 S라고도 정의한다. 이렇게 구해진 패리티 비트 S는 원 신호 X의 추정치인 부가 정보

Y와 함께 복호 과정에 이용되어 원 신호 X를 복원한다. 이때 일반적인 LDPC 부호의 복호 과정과는 달리 체크 노드에서의 값이 0 혹은 1이기 때문에 패리티 비트 S는 LDPC 부호의 복호과정에 이용된다.

DVC 시스템을 위한 LDPC 부호의 복호를 위하여 log 영역 SPA가 많이 사용되고 이를 설명하기 위해 그림 3과 복호 과정의 파라미터를 다음과 같이 정의하였다.

- $x_i, y_i \in \{0, 1\}, i = 1, 2, \dots, n$: i번째 v_i 연산에 적용된 X_i, Y_i 의 현재 값
- $l_i \in \{2, 3, \dots\}, i = 1, 2, \dots, n$: v_i 의 차수
- $q_{i,m}^{out}(q_{i,m}^{in}) \in R, i = 1, 2, \dots, n, m = 1, 2, \dots, l_i$: v_i 에서 m번째 edge를 통해서 들어오고 나가는 log likelihood Ratio(LLR)
- $s_j \in \{0, 1\}, j = 1, 2, \dots, n-k$: j번째 체크 노드 연산에 적용된 패리티 비트 S의 값
- $r_j \in \{2, 3, \dots\}, j = 1, 2, \dots, n-k$: c_j 의 차수
- $t_{j,m}^{out}(t_{j,m}^{in}) \in R, j = 1, 2, \dots, n-k, m = 1, 2, \dots, r_j$: c_j 에서 m번째 edge를 통해서 들어오고 나가는 LLR

DVC 시스템을 위한 LDPC 부호의 log 영역 SPA는 다음과 같고, 체크 노드의 패리티 값을 제외하면 일반적인 LDPC 부호의 복호에도 동일한 방법이 적용될 수 있다.

2.2.1. 초기화 과정

각 변수 노드 i에서 $q_{i,0}$ 를 식 (1)의 LLR로 초기화한다. 이때 $q_{i,0}$ 는 그림 3과 같이 i번째 변수 노드에 입력된다.

$$q_{i,0} = \log \frac{\Pr[x_i = 0|y_i]}{\Pr[x_i = 1|y_i]} = (1 - 2y_i) \log \frac{1-p}{p} \quad (1)$$

$$i = 1, 2, \dots, n, \quad p = \Pr[x_i \neq y_i] < 0.5$$

2.2.2. 변수 노드 연산과정

각 v_i 에서는 iteration 순서에 따라 $q_{i,m}^{out}$ 계산 방법이 달라진다. 첫 번째 iteration에서는 식 (2-1)과 같이 v_i 에 초기 값 $q_{i,0}$ 만이 입력되고 이후에는 식 (2-2)와 같이 v_i 에 초기 값 $q_{i,0}$ 과 v_i 와 연결된 c_j 에서 들어오는 $q_{i,j}^{in}$ 값들을 합한다. 이때 합하는 과정에서 주의해야 할 점은 c_j 에서 들어오는 $q_{i,j}^{in}$ 값들 중 j가 m일 때의 값은 제외하고 합하는 과정을 수행해야 한다.

$$q_{i,m}^{out} = q_{i,0} \quad (2-1)$$

$$q_{i,m}^{out} = q_{i,0} + \sum_{j=1, j \neq m}^{l_i} q_{i,j}^{in}$$

$$m = 1, 2, \dots, l_i, \quad i = 1, 2, \dots, n, \quad q_{i,j}^{in} = 0 \quad (2-2)$$

2.2.3. 체크 노드 연산과정

일반적인 LDPC 부호와는 다르게 그림 3과 같이 c_j 에 s_j 값이 입력되고 식 (3)을 이용하여 체크 노드 연산 과정을 수행한다.

$$t_{j,m}^{out} = (1 - 2s_j) \cdot \ln \frac{1 + \prod_{i=1, i \neq m}^{r_j} \tanh\left(\frac{t_{j,m}^{in}}{2}\right)}{1 - \prod_{i=1, i \neq m}^{r_j} \tanh\left(\frac{t_{j,m}^{in}}{2}\right)} \quad (3)$$

$$m = 1, 2, \dots, r_j, \quad j = 1, 2, \dots, n-k$$

이때 s_j 는 j번째 패리티 비트의 값으로서 s_j 가 1이면 $(1 - 2s_j)$ 값이 -1이므로 -1을 곱하게 되고 0이면 1을 곱하게 된다.

2.2.4. 복원 과정

앞서 1), 2), 3)과정을 거친 후 식 (4)를 통하여 원 신호 X를 복원하게 된다. 이러한 과정은 지정된 iteration 횟수까지 수행된다.

$$\hat{x} = \begin{cases} 0, & \text{if } q_{i,0} + \sum_{m=1}^{l_i} q_{i,m}^{in} \geq 0 \\ 1, & \text{if } q_{i,0} + \sum_{m=1}^{l_i} q_{i,m}^{in} < 0 \end{cases} \quad (4)$$

DVC 시스템에서는 복호기에서 추정된 부가 정보 Y와 부호기에서 전송 받은 패리티 비트 S를 이용하여 원 신호 X를 복원한다. 부가 정보 Y와 원 신호 X의 차이가 크다면 그에 따라 오류정정에 필요한 패리티 비트의 양도 증가하기 때문에 압축률이 줄어들게 된다. 따라서 원 신호 X를 복원해낼 때 필요한 패리티 비트의 수를 최소한으로 할 수 있도록 부호를 적응적인 특성을 갖도록 LDPC 부호를 설계해야 한다.

III. DVC 시스템을 위한 LDPC 부호 설계

3.1. LDPC 부호 설계

본 논문에서는 DVC 시스템을 위한 LDPC 부호를 설계하기 위해서 부호를 적응적인 특성을 갖는 효율적인 비균일 $n \times n$ H 행렬을 설계하였다. 이를

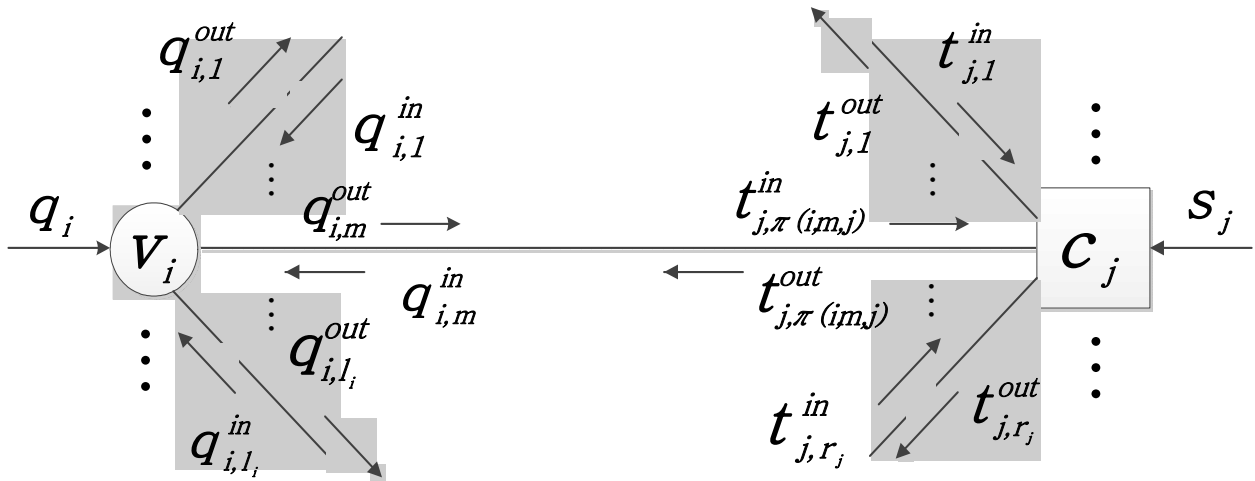


그림 3. I번째 변수 노드와 j번째 체크 노드의 복호 과정
 Fig. 3. Decoding procedure of the ith variable node and jth check node

위하여 Ascenso 등이 제안한 DVC 시스템을 위한 LDPC 부호 설계 방법[3]을 기반으로 H 행렬을 설계하고 ACE 알고리즘을 이용하여 LDPC 부호의 성능을 최대화하기 위한 다음과 같은 방법을 제안한다.

- 1) 주어진 변수 노드의 차수 분포 $\lambda(x)$ 를 토대로 낮은 차수를 갖는 변수 노드를 위한 H 행렬의 열부터 생성하여 압축률 1:1의 기본 H 행렬을 생성한다. 또한 부호율 적응적인 특성을 위해 우선 저압축률의 기본 부호를 생성하고 체크 노드를 병합하는 과정을 통해서 만들어지는 고압축률 부호에서 생길 수 있는 중복된 연결로 인한 edge의 손실이 생기지 않도록 기본 부호를 생성한다.
- 2) H 행렬의 설계 과정에서 ACE 알고리즘을 적용하여 d_{ace} 보다 작은 길이의 모든 cycle은 ACE 값이 최소한 η_{ACE} 값이어야 하는 조건을 만족할 수 있도록 한다⁴⁾.
- 3) 부호율 적응적인 특성을 위한 체크 노드 병합 과정을 수행한 후 고압축률 부호에서는 변수 노드의 수는 유지되지만 체크 노드의 수는 적어진다. 따라서 성능 저하의 요인이 되는 4 cycles이 불가피하게 발생하게 된다. 성능을 최대화하기 위하여 가장 고압축률 부호에 대해서 2 차수를 가진 변수 노드에서 4 cycles이 발생되지 않도록 한다.
- 4) 1)~3) 과정은 모든 변수 노드가 생성될 때까지 반복 수행한다.

기본 H 행렬을 생성하는 과정 1)에서 변수 노드에 대한 H 행렬의 각 열을 차수 분포를 기준으로 만들어 주는데 이때 체크 노드의 차수 분포를 고려하여 H 행렬을 생성하여야 성능을 향상시킬 수 있다. 단지 변수 노드의 차수 분포만을 고려하여 H 행렬을 만들어 준다면 체크 노드의 차수의 종류는 무수히 많아진다. 따라서 성능 향상을 위해서 체크 노드 차수의 종류를 일정하게 만들어야 한다. 본 논문에서는 다음과 같이 체크 노드의 차수 분포를 일정하게 유지하는 방법을 제안한다.

- a) H 행렬의 크기를 고려하여 병합 범위를 지정한다. 즉, 행 또는 열의 개수는 병합 범위의 배수가 되어야 한다.
 - b) H 행렬의 전체 행을 지정한 병합 범위의 크기만큼 묶어서 집합을 만들고 각 집합에 순번을 부여한다.
 - c) 배정한 순번을 랜덤하게 선택하고 선택한 순번에 따라서 지정된 집합의 모든 체크 노드들의 현재 차수의 값을 점검하여 가장 작은 차수 값을 가진 노드를 선택한다.
 - d) 다음 노드를 선택할 때는 이전에 선택된 순번을 다시 선택하지 않는다. 즉, 같은 집합의 체크 노드를 선택하지 않는다.
 - e) 모든 변수 노드를 생성할 때까지 위의 과정을 수행한다.
- a), b) 과정에서 H 행렬을 고려하여 병합 범위를 지정하고 병합 범위를 단위로 전체 행들을 묶어서 순번을 배정한 후에 이를 랜덤하게 선택한다. 예를

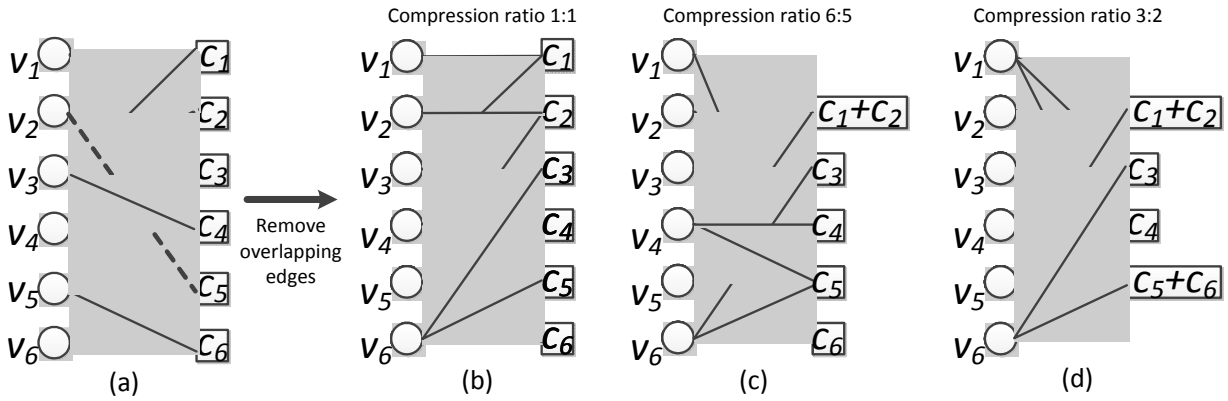


그림 4. 저압축율 H 행렬의 Tanner 그래프와 체크 노드 병합을 통한 부호율을 조절 (a) 체크 노드 병합 과정을 고려하지 않은 일반적인 H 행렬의 Tanner 그래프 (b) 체크 노드 병합 과정을 고려하여 중복 연결된 edge를 제거한 H 행렬의 Tanner 그래프 (c) 체크 노드 c_1 과 c_2 를 병합한 다음의 $CR = 6 : 5$ 인 H 행렬의 Tanner 그래프 (d) 체크 노드 c_1 과 c_2 , c_5 와 c_6 을 각각 병합한 다음의 $CR = 6 : 4$ 인 H 행렬의 Tanner 그래프

Fig. 4. Tanner graph for H matrix with low compression ratio and adaptation of compression ratio by merging check nodes (a) Tanner graph for a normal H matrix (b) Tanner graph for H matrix after removing overlapping edges (c) Tanner graph for H matrix after merging check node c_1 and c_2 ($CR=6:5$) (d) Tanner graph for H matrix after merging check node c_1 and c_2 , c_5 and c_6 ($CR=6:4$)

들면, 396×396 크기의 H 행렬에서 체크 노드의 병합 범위를 6으로 지정하였다면 $396/6 = 66$ 개의 행들의 집합으로 묶어서 순번을 지정할 수 있다. 이때 b) 과정에서 행들의 집합에 0 ~ 65의 순번을 배정하고 순번을 랜덤하게 선택한다. c) 과정에서는 지정된 순번 안의 체크 노드의 차수를 점검하고 가장 작은 차수 값을 가진 노드를 우선 선택함으로써 선택되지 않는 체크 노드가 없고 고르게 선택될 수 있도록 한다. d) 과정을 통하여 병합 범위 안의 체크 노드는 병합과정 후에도 중복 연결로 인한 손실이 없도록 한다.

과정 1)을 수행하면서 형성된 변수 노드는 과정 2)와 3)에서 제시하는 조건을 만족해야 한다. 먼저 과정 2)의 ACE 알고리즘에서 제시하는 계수들을 이용하여 최소 정지 집합의 크기를 증가시킴으로써 성능을 향상 시킨다. 또한 과정 3)에서는 부호율 적응적인 특성을 만족하기 위해서 H 행렬을 부호율에 맞게 압축하게 된다. 이때 edge가 중복되지 않기 때문에 H 행렬의 크기가 작아지는 상황에서 1의 수는 유지되기 때문에 4 cycles의 형성을 피할 수가 없게 된다. 본 논문에서는 설계하는 비균일 H 행렬에서 2 차수를 가진 변수 노드가 가장 고압축율의 H 행렬일 때 4 cycles을 형성하지 않도록 하여 성능을 향상 시킨다. 과정 2), 3)의 조건을 만족시키지 못하면 다시 변수 노드를 발생하기 위해 과정 1)

을 수행한다.

3.2 부호율 적응적인 LDPC 부호를 위한 체크 노드 병합 방법

DVC 시스템을 위한 LDPC 부호에서는 부호율 적응적인 특성을 갖도록 하기 위하여 체크 노드를 병합할 수 있는데[3] 이를 설명하기 위하여 그림 4와 같이 $n \times n$ 저압축율 패리티 검사 행렬 H에 대응하는 Tanner 그래프를 제시하고 체크 노드 병합 과정을 통한 부호율 조절 과정을 도시하였다.

DVC 시스템을 위한 LDPC 부호에서는 저압축율의 기본 패리티 검사 행렬에서 각 체크 노드를 병합하는 방식으로 원하는 부호율을 위한 H 행렬을 구한다. 저압축율 기본 패리티 검사 행렬 H의 압축율 (compression ratio: CR)은 1:1이 된다. 부호율 적응적인 특성을 갖도록 하기 위해서 2개의 체크 노드를 병합하는데 체크 노드를 병합하면 H 행렬의 행의 수가 작아지고 고압축율 부호가 생성된다. 즉, H 행렬의 두 개의 행이 더해지는 과정과 같은데 처음에 그림 4(b)와 같이 $CR = 1 : 1$ 의 6×6 크기의 H 행렬을 생성하고 그림 4(c)와 같이 C_1 과 C_2 을 병합하여 $CR = 6 : 5$ 의 H 행렬을 만들고 최종적으로 그림 4(d)와 같이 C_5 과 C_6 을 병합하여 $CR = 3 : 2$ 의 행렬을 만드는 과정을 수행한다. 이러한 방법을 이용하면 하나의 기본 H 행렬을 이용하여 여러 부호율의 행렬을 생성할 수 있다.

고압축을 부호를 생성하는 과정에서 병합되는 체크 노드들과 연결되어 있는 변수 노드들이 중복 연결되지 않도록 그림 4(b)에 대응하는 H 행렬을 설계해야 한다. 만약 그림 4(a)에 대응하는 H 행렬을 설계 하면 체크 노드 병합 과정을 수행하면서 변수 노드와의 연결이 끊어지게 된다. 즉, 병합 범위 안의 병합 가능한 행들 중에서 같은 열에 변수 노드와 체크 노드의 연결 선을 나타내는 1 값을 가질 수 없다. 그러므로 그림 4(b)와 같이 중복 연결된 edge를 없애고 H 행렬을 설계하면 병합 범위의 어떠한 체크 노드들을 병합하더라도 노드의 연결을 나타내는 edge의 손실을 막을 수가 있다.

DVC 시스템을 위한 LDPC 부호에서는 부호를 적용적인 특성을 갖도록 하기 위하여 병합 대상인 체크 노드를 각각 선택하게 된다. 일반적인 병합 방법으로는 병합 범위 안에서 순차적으로 체크 노드를 병합한다. 그러나 이러한 방법에서는 병합하는 과정에서 체크 노드의 차수를 반영하지 않았기 때문에 고압축을 부호로 갈수록 체크 노드의 차수의 규칙성을 보장받지 못하게 된다. 또한, 4 cycle의 성능 저하 영향을 받게 된다. 따라서 본 논문에서는 다음과 같은 체크 노드를 병합 방법을 제안한다.

- 1) 병합 범위 안의 체크 노드의 차수 값을 점검한다. 즉, H 행렬의 각각 행을 구성한 1의 개수를 구한다.
- 2) 가장 큰 차수 값을 가진 체크 노드를 우선 선택

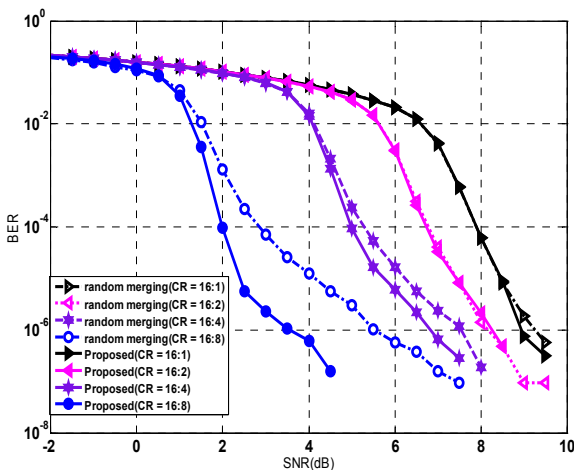


그림 5. H 행렬을 생성할 때 랜덤하게 만드는 방법과 규칙적으로 만드는 제안하는 방법에 따른 부호를 적용적인 LDPC 부호의 BER 성능 비교 (block size : 1584, iteration : 50, 식(5)의 차수 분포 사용, ACE (6, 2), 병합 범위 16을 사용한 경우)
 Fig. 5. BER performance comparison between rate adaptive LDPC codes with the conventional H matrix and proposed H matrix (block size: 1584, iteration: 50, ACE(6,2), using the degree distribution of (5))

한다. 즉, 각 행의 1의 합을 비교하여 가장 큰 수를 가진 행을 선택한다.

- 3) 가장 작은 차수 값을 가진 체크 노드를 우선 선택한다. 즉, 각 행의 1의 합을 비교하여 가장 작은 수를 가진 행을 선택한다.
- 4) 만약 가장 작은 값을 가진 체크 노드가 여러 개라면 체크 노드들 중에서 현재 H 행렬과 비교하여 4 cycle 형성이 가장 작은 체크 노드를 선택한다.
- 5) 선택된 두 체크 노드를 병합한다.

저압축을 부호에서는 다양한 체크 노드의 차수 값을 갖게 된다. 일반적인 병합 방법으로는 다양한 체크 노드의 차수 값을 그대로 갖게 되지만 본 논문에서는 제안하는 병합 방법을 통해서 고압축을 부호로 갈수록 체크 노드의 차수 값을 최대한 일정하게 분포하도록 할 수 있다. 또한, 4 cycle의 수를 점검하여 최대한의 성능 향상을 보장하였다. 하지만 어떠한 방법으로 병합 과정을 수행해도 가장 큰 압축을 부근에 도달하면 비슷한 차수를 갖게 되고 4 cycle을 피하지 못한다.

IV. 성능평가

본 논문에서는 DVC 시스템을 위한 LDPC 부호의 원 정보 X에 AWGN 잡음을 첨가하여 생성한 부가 정보 Y와 부호기에서 생성된 패리티 비트를 가지고 복호화 과정을 수행하여 여러 부호율의 LDPC 부호에 대한 BER 성능을 측정하였다. 저압축을 패리티 검사 행렬 H의 체크 노드 차수를 랜덤하게 생성하고 순차적으로 병합하여 랜덤하게 만들어진 각 요소 H 행렬을 생성하는 방법과 체크 노드의 차수를 규칙적으로 생성하는 제안 방법을 적용하여 저압축을 패리티 검사 행렬 H를 생성하고 제안하는 병합 방법을 적용하여 각 요소 H 행렬을 만들어 성능을 비교하였다. LDPC 부호의 복호 알고리즘으로는 DVC 시스템을 위한 LDPC 부호의 SPA(sum product algorithm)를 사용하였다^[6]. 패리티 검사 행렬을 만드는데 사용한 변수 노드를 위한 차수 분포는 다음과 같다^[7,8].

$$\lambda(x) = 0.38354x + 0.04237x^2 + 0.57409x^3 \quad (5)$$

$$\lambda(x) = 0.23802x + 0.20997x^2 + 0.03492x^3 + 0.12015x^4 + 0.01587x^6 + 0.0048x^{13} + 0.37627x^{14} \quad (6)$$

$$\lambda(x) = 0.131x + 0.26x^2 + 0.187x^6 + 0.115x^7 + 0.08x^{18} + 0.227x^{20} \quad (7)$$

이때 ACE 알고리즘을 위한 d_{acc} 와 η_{acc} 값은 (6, 2), (10, 5), (13, 7)을 사용하였다. 또한 체크 노드 병합 범위는 16을 사용하였다.

그림 5에서는 식 (5)의 차수 분포와 ACE 알고리즘의 계수 (6, 2)를 이용하여 H 행렬을 생성하는 방법에 따라서 설계된 LDPC 부호의 성능을 비교하였다. 각 부호율에서 제안하는 방법으로 생성된 H 행렬이 체크 노드에 대한 차수 분포의 규칙성을 유지하고 cycle의 특성을 개선하여 오류 마루 현상을 완화하여 랜덤하게 만들어진 H 행렬을 생성하여 순차적으로 병합하여 생성된 H 행렬보다 더 나은 성능을 보이고 있다. 고압축률 부호로 갈수록 성능의 차이는 줄어들지만 CR이 16:8인 경우에는 최대 3 dB 정도 성능 차이를 보였다. 그림 6은 식 (6)의 차수 분포와 ACE 알고리즘의 계수 (10, 5)를 이용하여 H 행렬을 생성하는 방법에 따라서 설계된 LDPC 부호의 성능을 비교하였다. 각 부호율에서 제안하는 방법으로 생성된 H 행렬이 체크 노드에 대한 차수 분포의 규칙성을 유지하고 cycle의 특성을 개선하여 오류 마루 현상을 완화하여 랜덤하게 만들어진 H 행렬을 생성하여 순차적으로 병합하여 생성된 H 행렬보다 더 나은 성능을 보이고 있다. 하지만 고압축률로 갈수록 랜덤하게 만든 H 행렬도 병합 과정을 수행하면서 자연스럽게 규칙적인 체크 노드의 차수를 갖게 되어 비슷한 성능을 보이게 된다.

그림 7은 식 (7)의 차수 분포와 ACE 알고리즘의 계수 (13, 7)를 이용하여 H 행렬을 생성하는 방법에 따라서 설계된 LDPC 부호의 성능을 나타낸다. 먼저 체크 노드의 차수를 랜덤하게 만드는 방법과 제안하는 방법을 이용하여 각각의 기본 H 행렬을 생성한다. 성능 비교를 위해 랜덤하게 만든 기본 H 행렬은 순차적으로 병합하여 랜덤한 각 요소 H 행렬을 만들고 체크 노드의 차수 분포를 유지하여 만든 기본 H 행렬은 본 논문에서 제안한 병합 방법을 이용하여 CR이 각각 16:8과 16:4인 경우의 H 행렬을 설계하였다. 각 부호율에서 제안하는 방법으로 생성된 H 행렬이 체크 노드에 대한 차수 분포의 규칙성을 유지하고 cycle의 특성을 개선하여 오류 마루 현상을 완화하여 랜덤하게 만들어진 H 행렬을 생성하여 순차적으로 병합하여 생성된 H 행렬보다 더 나은 성능을 보이고 있다. 특히, CR이 16:8인 경우에 더욱 나은 성능을 보이는데 고압축률 부호로 갈수록 랜덤하게 만든 H 행렬도 체크 노드의 규칙성을 갖게 되고 제안하는 방법도 짧은 cycle에 대한 성능 저하를 피할 수 없게 되어 고압축률로 갈수록 성능 차이는 거의 없게 된다.

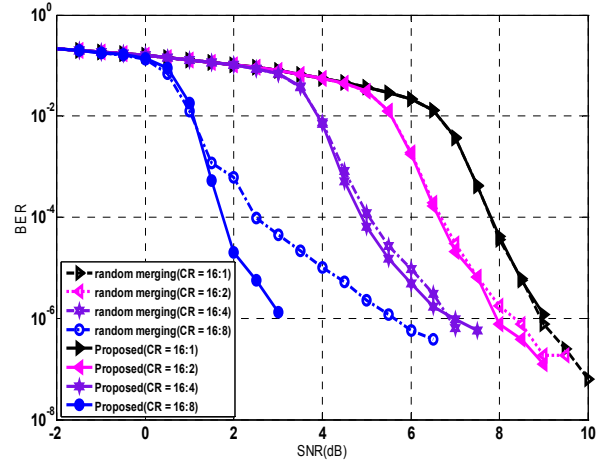


그림 6. H 행렬을 생성할 때 랜덤하게 만드는 방법과 규칙적으로 만드는 제안하는 방법에 따른 부호율 적응적인 LDPC 부호의 BER 성능 비교 (block size : 1584, iteration : 50, 식 (6)의 차수 분포 사용, ACE (10, 5), 병합 범위 16을 사용한 경우) Fig. 6. BER performance comparison between rate adaptive LDPC codes with the conventional H matrix and proposed H matrix (block size: 1584, iteration: 50, ACE(10,5), using the degree distribution of (6))

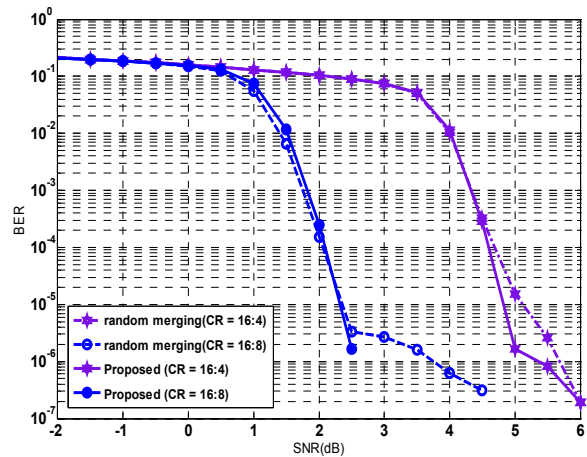


그림 7. H 행렬을 생성할 때 랜덤하게 만드는 방법과 규칙적으로 만드는 제안하는 방법에 따른 부호율 적응적인 LDPC 부호의 BER 성능 비교 (block size : 1584, iteration : 50, 식 (7)의 차수 분포 사용, ACE (13, 7), 병합 범위 16을 사용한 경우) Fig. 7. BER performance comparison between rate adaptive LDPC codes with the conventional H matrix and proposed H matrix (block size: 1584, iteration: 50, ACE(13,7), using the degree distribution of (7))

V. 결론

본 논문에서는 DVC 시스템을 위한 부호율 적응적인 LDPC 부호를 설계하였다. LDPC 부호의 성능에 큰 영향을 미치는 cycle과 차수 분포를 감안하여 CR이 1:1인 기본 비균일 H 행렬을 설계하였다. 최적의 성능을 보이는 LDPC 부호의 설계를 위해서 체크 노

드의 차수 분포를 최대한 일치시키기 위한 방법으로 각 체크 노드의 차수를 점검하여 가장 작은 차수 값을 갖는 노드를 우선 선택하여 체크 노드의 차수를 고르게 분포시키는 방법을 제안하였고 생성된 기본 비균일 H 행렬을 대상으로 병합범위 안에서 체크 노드를 병합하는 과정을 수행하여 설계한 LDPC 부호가 부호율 적응적인 특성을 갖도록 하였다. 이때 제안하는 병합 방법을 이용하여 체크 노드를 병합하여 설계한 LDPC 부호가 일반적인 병합 방법을 사용하여 설계한 LDPC 부호보다 더 나은 성능을 보임을 입증하였다.

참 고 문 헌

- [1] R. G. Gallager, Low density parity check codes, MIT Press, Cambridge, MA, 1963.
- [2] J. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, pp. 471-480, July 1973.
- [3] J. Ascenso, C. Brites and F. Pereira, "Design and performance of a novel low-density parity-check code for distributed video coding", *IEEE International Conference on Image Processing*, Los Angeles, America, 2008.
- [4] T. Tian, C.R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction", *IEEE Trans. on Communications*, vol. 52, No. 8, pp. 1242-1247, Aug. 2004.
- [5] S. H. Kim, J. S. Kim, D. S. Kim, and H. Y. Song, "LDPC code construction with low error floor based on the IPEG algorithm," *IEEE Commun. Lett.*, vol. 11, no. 7, pp. 607 - 609, July 2007.
- [6] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 10, pp. 440 - 442, Oct. 2002.
- [7] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check nodes", *IEEE Trans. Inform. Theory*, vol. 47, No. 2, pp. 619-637, Feb. 2001.
- [8] D. Varodayan, A. Aaron and B. Girod, "Rate-Adaptive Codes for Distributed Source

Coding," *EURASIP Signal Processing Journal, Special Section on Distributed Source Coding*, vol. 86, no. 11, pp. 3123 - 3130, Nov. 2006.

노 현 우 (Hyeun-woo Noh)

준회원



2009년 2월 가톨릭대학교 정보통신전자공학부 학사 졸업
2010년 3월~현재 가톨릭대학교 정보통신전자공학과 석사과정
<관심분야> LDPC 부호, 영상통신

이 창 우 (Chang-woo Lee)

중신회원



1988년 서울대학교 제어계측공학과(공학사)
1990년 서울대학교 제어계측공학과 석사졸업
1996년 서울대학교 제어계측공학과 박사 (영상신호처리 전공)
1996년 3월~1997년 8월 삼성

전자 신호처리연구소 선임연구원
현재 가톨릭대학교 정보통신전자공학부 교수
<관심분야> 영상 신호처리, 영상 통신, LDPC 부호