DBpia

LDPC 부호와 RA 부호의 최소 거리 검색 알고리즘

정회원 정 규 혁*

Minimum Distance Search Algorithms of LDPC Codes and RA Codes

Kyuhyuk Chung* ***Regular Member***

요 약

본 논문은 반복 부분을 이용하여 단지 유효한 부호어만을 검색함으로서 RA 부호의 최소 거리를 구하기 위한 계산량을 줄인다. LDPC 부호도 RA 부호와 같이 반복 부분을 가지므로 제안된 알고리즘은 LDPC 부호의 최소 거리 계산에도 적용된다. 최소 거리는 높은 신호대 잡음비에서 부호의 성능을 결정한다. 따라서 오류 마루를 추정하는 것을 가능하게 한다. 제안된 알고리즘은 부호 구조에 어떠한 제한도 두지 않고 최소 거리를 구할 수 있다. 실제적 의미가 있는 큰 길이의 인터리버를 가진 LDPC 부호와 RA 부호의 최소 거리가 본 논문에서 구해지며 이에 따른 오류 마루를 구하며 또한 이 오류 마루는 반복 복호의 성능과 비교된다.

Key Words : LDPC 부호, RA 부호, 최소 거리, 상향 한계, 오류 마루

ABSTRACT

In this paper, we reduce the computational complexity to find the minimum distance of RA codes by searching only valid codewords using repetition part. Since LDPC codes have repetition part like RA codes, we also apply this algorithm for computing the minimum distance of LDPC codes. The minimum distance dominates the code performance at high signal-to-noise ratios(SNRs) and in turn allows an estimate of the error floor. The proposed algorithm computes the minimum distance without any constraint on code structures. The minimum distances of LDPC codes and RA codes with large interleavers of practical importance are computed and used to obtain the error floor, which is compared with the performance of the iterative decoding.

Ⅰ. 서 론

터보 부호와 직렬 결합 길쌈 부호(Serially Concantenated Convolutional Code(SCCC))의 출현은[1, 2] 반복 복호나 그래프 위의 부호 등 새로운 개념과 기술을 이끌어 내었다. 한편, Low Density Parity Check (LDPC) 부호는[3, 4] 탁월한 성능과 하드웨어 구현의 용이성 때문에 많은 연구가 진행되고 있다. 다수의 모의실험과 성능 한계들이 이러한 부호들의 탁월한 성능을 입증 해주고 있다[5-9].

특히 Repeat Accumulate(RA) 부호는[10] 간단한 구조와 탁월한 성능 때문에 많은 관심을 받고 있다. 터보 부호와 SCCC 부호에 대해서는 최소 길이를 계산하는 좋은 알고리즘이 제시되었다[11]. 그러나 실제적으로 중요한 의미를 가지는 큰 인터리버를 가진 SCCC 부호에 대해 최소 길이를 구하는 데에는 검색 입력 시퀀스의 수가 과도하게 증가되는 결과를 낳게 된다.

본 논문에서는 반복 부분을 사용하여 단지 유효한 부호어만을 검색하여 RA 부호의 최소 길이를

* 단국대학교 정보컴퓨터학부 (khchung@dku.edu)
논문번호 : KICS2006-01-006, 접수일자 : 2006년 1월 3일, 최종논문접수일자 : 2006년 3월 15일

www.dbpia.co.kr

계산하는 량을 감소시켰다. RA 부호가 반복 부호와 축적기의 직렬연결로 이루어진 간단한 구조를 가지고 있음에 착안하였다. 제안된 알고리즘은 부호에 어떠한 제한도 두지 않고 최소 길이를 계산한다. 실제적 중요성을 가지는 인터리버 길이의 RA 부호의 최소 거리가 본 논문에서 구해지며 에러 마루를 얻기 위해 사용되어진다. 이 에러 마루는 반복 복호의 성능과 비교된다. LDPC 부호 또한 RA 부호와 마찬가지로 반복 부분을 가지므로 제안된 알고리즘을 LDPC 부호의 최소 거리를 계산하는 데 적용하였다.

본 논문은 다음과 같이 구성되었다. II장에서는 SCCC 부호의 최소 거리를 구하는 기존의 연구를 소개하고 III장에서는 LDPC 부호와 RA 부호의 관계를 설명하고 IV장에서는 LDPC 부호와 RA 부호의 최소 거리를 계산하는 계산량을 감소시키는 알고리즘을 제안한다. V장에서는 최소 거리 계산 결과와 시뮬레이션 성능을 보이고 VI장에서 결론을 맺는다.

II. SCCC 부호의 최소 거리 계산에 대한 기존의 연구

SCCC 부호의 최소 거리를 계산하는 알고리즘이 제안되어진 바 있다[11]. 입력 정보 블록 길이가 K이고 인터리버 길이가 L이고 전체 (N,K) 직렬 결합 길쌈 부호 C에 대해 부호율이 $r=(K/L)(L/N)=K/N$로 주어졌다고 가정할 때 두개의 길쌈 부호 E_{outer}와 E_{inner}가 길이가 L인 인터리버로 직렬로 연결되어 있다고 가정한다. 심벌 Γ_{outer}와 Γ_{inner}는 길쌈 부호기 E_{outer}와 E_{inner}에 각각 상응하는 트렐리스를 나타낸다. 길이가 $i \le K$인 이진 입력 시퀀스는 $\mathbf{u}^{(i)}=(u_0,u_1,\cdots,u_{i-1})$로 나타낸다. 외부 부호기 E_{outer}가 $\mathbf{u}^{(i)}$를 입력으로 받아 외부 부호기 부분 부호어 $\mathbf{c}_{(outer)}(\mathbf{u}^{(i)})$를 출력한다. 외부 부호기 부분 부호어$\mathbf{c}_{(outer)}(\mathbf{u}^{(i)})$는 인터리버로 위치를 뒤섞은 다음 내부 트렐리스 Γ_{inner}에 제한(constraint)을 두게 된다. 이렇게 Γ_{inner}의 제한된 부부호(sub-code)의 모든 부호어들 중에서 최소 거리를 $\nu(\mathbf{u}^{(i)})$로 나타낸다[11]. 제한된 부부호(sub-code)는 $\mathbf{u}^{(i)}$와 처음 i개의 비트가 동일한 K개의 비트의 정보 프레임 $\mathbf{u}^{(K)}=(u_0,u_1,\cdots,u_{K-1})$에 의해 생성되는 모든 부호어의 최소 거리를 구하는 데 사용된다. 그러한 예가 그림 1에 도시되어 있다. 그림 1에서는

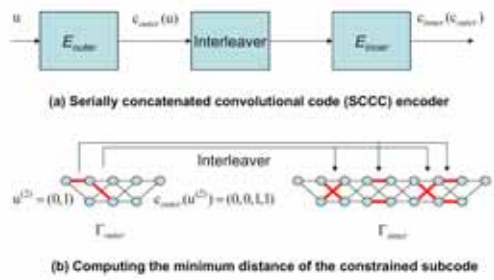

그림 1. SCCC 부호와 제한된 부부호(sub-code)의 최소 거리 계산의 예

부호율 $r_{outer}=r_{inner}=1/2$인 두개의 이진 규칙적인(systematic) two-state 길쌈 부호를 가진 SCCC 부호를 생각한다. 그림 1에 Part(b)는 $\nu(\mathbf{u}^{(2)})$를 계산하기 위하여 길이가 $i=2$인 입력 시퀀스를 나타내고 있다. 주어진 예에선 $\mathbf{c}_{(outer)}(\mathbf{u}^{(2)})=(0,0,1,1)$를 생성하기 위해서 $\mathbf{u}^{(2)}=(0,1)$를 E_{outer}롤 부호화하였다. $\mathbf{c}_{(outer)}(\mathbf{u}^{(2)})$의 인터리버를 통한 4개의 비트는 Γ_{inner}에 다중 제한(constraint)을 만들게 된다. 제한된 부부호(sub-code)의 최소 길이 $\nu(\mathbf{u}^{(2)})$의 계산은 길쌈 부호의 프리 길이를 계산할 때 사용되어지는 통상적인 Viterbi 알고리즘[13] 같은 기술에 몇 가지 수정을 하여 계산되어진다.

완전한 알고리즘은 다음과 같이 가장 간단한 형태로 기술되어질 수 있다. 우선 pseudo-code로 기술하면 다음과 같다.

```
FOR 1 ≤ i ≤ K−1
  compute ν(u^(i))
  IF ν(u^(i)) > d_min THEN
    discard u^(i)
  ELSE
   keep u^(i)
  ENDIF
        obtain u^(K)=(u_0,u_1,・・・,u_{K−1})
            by adding (K−i) zeros
  IF Hamming weight of
   c_inner(c_outer(u^(K))) < d_min THEN
    update d_min
  ENDIF
ENDFOR
```

위에서 기술된 pseudo-code를 상술하면 다음과

www.dbpia.co.kr

같다. 길이 $i=1$인 두개의 입력 시퀀스 $\mathbf{u}^{(1)}=(0)$과 $\mathbf{u}^{(1)}=(1)$로 시작한다. $1 \le i \le K-1$를 만족하는 어떤 길이 i에 대해서도 검색되는 입력 시퀀스 $\mathbf{u}^{(i)}$들의 각각에 대해 $\mathrm{v}(\mathbf{u}^{(i)})$를 계산한다. 만약 $\mathrm{v}(\mathbf{u}^{(i)}) > \mathrm{d}_{\min}$이면 $\mathbf{u}^{(i)}$는 제거된다. 그렇지 않으면 $\mathbf{u}^{(i)}$는 계속 유지한다. 여기에 $(K-i)$개의 제로를 추가하여 얻어진 K-비트의 정보 프레임 $\mathbf{u}^{(K)}=(u_0,u_1,\cdots,u_{K-1})$를 고려하고 SCCC 부호어 $\mathbf{c}_{\mathbf{inner}}(\mathbf{c}_{outer}(\mathbf{u}^{(K)}))$의 해밍 무게를 계산한다. 만약 $\mathbf{c}_{\mathbf{inner}}(\mathbf{c}_{outer}(\mathbf{u}^{(K)}))$의 해밍 무게가 $d_{\min}$보다 작다면 $d_{\min}$을 갱신한다. $i=K$일 때 알고리즘은 완료되고 $d_{\min}$을 계산한다.

Ⅲ. LDPC 부호와 RA 부호의 관계

3.1 RA 부호를 LDPC 부호로 표현

RA 부호가 직렬 결합 부호이고 LDPC 부호는 선형 블럭 부호이다. 그러나 두 부호는 반복 연산과 축적 연산 같은 공통된 부분을 가지고 있다. RA 부호는 무작위 인터리버로 연결된 반복 부호와 축적 부호로 구성되어져 있다. 한편 LDPC 부호는 패리티 체 매트릭스 H에 정렬되어 있는 다수의 싱글 패리티 체 식들로 구성되어 있다. RA 부호를 LDPC 부호로 표현하는 데에는 패리티 체 식이 제로 시작 상태와 제로 종결 상태를 가지는 축적기로 표현될 수 있다는 사실에 기초하고 있다. 만약 LDPC 부호의 패리티 체 행렬 H가 이중 대각 부분 $\mathrm{H_p}$를 가진다면 RA 부호는 LDPC 부호로 변환될 수 있다. 구체적으로 행렬 H속의 이중 대각 부분 $\mathrm{H_p}$는 주대각과 부대각에 1을 가지고 나머지 부분은 제로를 가지는 정방 행렬이다. RA 부호의 반복 부분은 패리티 체 행렬 H속의 규칙적인(systematic) 정보 부분 $\mathrm{H_d}$의 열 무게로 쉽게 표현될 수 있다. 마지막으로 변환되어야 할 부분은 RA 부호의 인터리버이다. 이 변환은 RA 부호의 인터리버의 순서에 따라 H 행렬의 1을 위치시킴으로 쉽게 가능하다. 그러한 예가 그림 2에 도시되어 있다. 그림 2에서는 부호율 $r=1/2$이고 반복 인수 $q=4$인 RA 부호는 반복되고 인터리버를 통과한 4개의 축적 중에 1개의 패리티 비트를 생성한다. 이러한 패리티 비트들이 규칙적인(systematic) 비트와 함께 부호어를 이룬다. 여기서 H 행렬은 $\mathrm{H_p}$와 $\mathrm{H_d}$의 두 분분으로 이루어져 있으며 각각 규칙적인(systematic) 비트와 패리티 비트에 상응한다. (N,K) LDPC 부호의 패리티 체 행렬 $\mathrm{H}=[h_{i,j}]$에 대해 H 행렬의 행의 수는 $N-K$이다. 각각의 i-번째 행은 다음과 같은 패리티 체 식을 나타낸다.

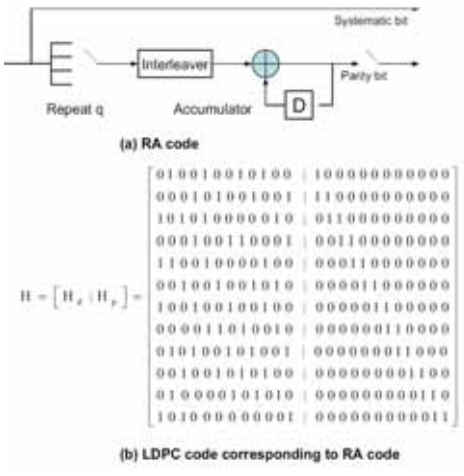

그림 2. RA 부호를 LDPC 부호로 변환하는 예

$$\sum_{j=0}^{K-1} h_{i,j} d_j + \sum_{j=K}^{N-1} h_{i,j} p_{j-K} = 0 \qquad (1)$$

여기서 $h_{i,j}$의 i, j 첨자는 H 행렬의 i-번째 행과 j-번째 열을 나타내고 d_j는 규칙적인(systematic) 비트를 p_j는 패리티 비트를 나타낸다. 만약 LDPC 부호의 H 행렬이 RA 부호로부터 얻어 졌다면 부호화 과정은 생성 행렬 G 대신에 H 행렬을 통해 다음과 같이 이루어질 수 있다.

$$p_0 = \sum_{j=0}^{K-1} h_{0,j} d_j \ , \ p_{i+1} = p_i + \sum_{j=0}^{K-1} h_{i,j} d_j \qquad (2)$$

여기서 $i=0,1,\cdots,N-K-2$이다. 더욱이 H 행렬이 저밀도이므로 부호화 과정이 선형 시간에 이루어질 수 있다. 여기서 한 가지 주목할 점은 RA 부호의 인터리버와 연관된 H 행렬의 각각의 행의 순서이다. 수식 (1)의 교환 법칙에 의해 규칙적인(systematic) 비트들의 순서는 자유롭게 바뀔 수 있다. 결과적으로 같은 패리티 체 식들에 상응하는 여러 개의 인터리버들이 같은 H 행렬로 나타내어지게 된다.

위의 내용들을 기초로 RA 부호는 LDPC 부호의 부분 집합으로 생각될 수 있다. 그러므로 어떤 제한을 가지는 LDPC 부호들은 RA 부호처럼 표현될

www.dbpia.co.kr

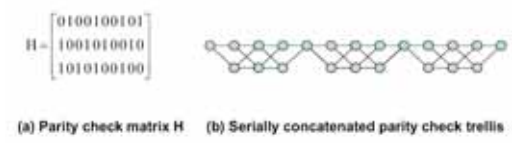

그림 3. 패리티 체 행렬과 상응하는 직렬 연결 조합된 트렐리스의 예

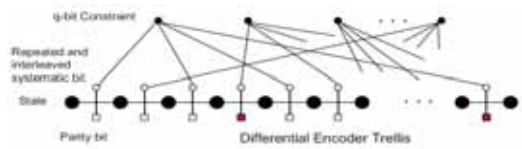

그림 4. 규칙적인(systematic) RA 부호의 유효 부호 표현

수 있다. 이러한 내용은 다음 절에서 다루게 된다.

3.2 LDPC 부호를 RA 부호로 표현

최소 거리를 계산하고자 할 때 부호어 집합을 신중하게 정의할 필요가 있다. 그렇지 않으면 계산량이 지나치게 많아지게 된다. 최소 거리를 계산하는데 있어 부호어 집합이 잘 정의되어 있으면 실제적인 부호화 과정은 반드시 필요한 것은 아니다. 그러므로 최소 거리를 얻기 위해 LDPC 부호를 효과적으로 RA 부호와 같이 표현하려고 한다. 이전 절로부터 패리티 체 식이 제로 시작 상태와 제로 종결 상태를 갖는 축적기로 표현되어질 수 있다는 것을 알았다. H 행렬 속에 구조화된 H_p 부분을 가지지 않는 일반적인 LDPC 부호에 대해 H 행렬 안에 있는 패리티 체 식들에 대한 이러한 짧은 축적기들은 H 행렬의 각각의 행 무게마다 제로 상태로 종결시킴으로서 직렬로 연결할 수 있다. 이것이 LDPC 부호를 RA 부호처럼 보이게 한다. 그러한 예가 그림 3에 도시되어 있다. LDPC 부호의 RA 부호와 같은 표현을 이용하여 최소 거리 계산량이 트렐리스 표현의 효율성으로 인해 상당히 감소될 수 있다.

Ⅳ. 유효 부호어(Valid Codeword (VCW)) 검색 알고리즘

터보 부호와 SCCC 부호의 주요한 차이점은 터보 부호의 각각의 부호화기는 입력 시퀀스 $\mathbf{u}^{(K)}=(u_0,u_1,\cdots,u_{K-1})$와 입력 시퀀스의 인터리버를 거친 시퀀스에 의해 이루어지는 반면 SCCC 부호의 바깥쪽 부호는 입력 시퀀스 $\mathbf{u}^{(K)}=(u_0,u_1,\cdots,u_{K-1})$에 의해 부호화되고 SCCC 부호의 안쪽 부호는 길이가 $L \geq K$인 바깥쪽 부호화기 E_{outer}의 출력 시퀀스 $\mathbf{c}^{(L)}=(c_0,c_1,\cdots,c_{L-1})$에 의해 부호화된다는 점이다. 이러한 사실로 Γ_{inner}의 모든 경로의 수는 2^L이지만 SCCC 부호의 모든 부호어의 수는 $2^K \leq 2^L$이므로 안쪽 트렐리스 Γ_{inner}의 모든 경로가 모두 부호어가 되지는 않는다. 다시 말해 Γ_{inner}의 2^L-2^K개의 경로는 유효한 부호어가 아니다. 그리므로 본 논문에서는 특별히 가장 간단한 형태의 SCCC 부호인 RA 부호에 대해 하나의 알고리즘을 제안하고자 한다. 먼저 규칙적인(systematic) RA 부호를 하나의 또 다른 graphical 모델로 그림 4에서 표현하였다. 주어진 모델에서는 RA 부호의 직렬 결합 표현이 하나의 유효 부호 표현으로 변환되어졌다. 주어진 모델의 각각의 제한(constraint)은 q를 반복 횟수라 할 때 q 번 반복된 규칙적인(systematic) 비트들이 같게 되도록 만든다. 여기서 유효 부호의 입력 시퀀스 $\mathbf{u}^{(L)}=(u_0,u_1,\cdots,u_{L-1})$는 $\mathbf{u}^{(K)}=(u_0,u_1,\cdots,u_{K-1})$의 q 번 반복되고 인터리버를 통과한 시퀀스가 된다. 이러한 모델에 기초하여 완전한 알고리즘은 가장 간단한 형태로 다음과 같이 기술될 수 있다. 우선 pseudo-code로 기술하면 다음과 같다.

```
FOR 1 ≤ i ≤ L
  FOR s_i = {0,1}
    FOR s_{i-1} = {0,1}
      FOR survivor u^(i-1) in s_{i-1}
      IF c_VCW(u^(i)) is valid
        compute I(u^(i)) and D(u^(i))
        IF D(u^(i)) > d_min THEN
        discard u^(i)
        ELSE
        keep u^(i)
        ENDIF
        obtain u^(L) by adding (K-i) zeros
        IF D(u^(L)) < d_min THEN
       update d_min
        ENDIF
      ENDIF
      ENDFOR
    ENDFOR
  ENDFOR
ENDFOR
```


www.dbpia.co.kr

위에서 기술된 pseudo-code를 상술하면 다음과 같다. 최소 거리가 어떤 값 d^*와 같거나 작다고 가정한다. 최소 거리 $d_{\min}$은 d^*로 초기화한다. 제로 시간에서의 초기 상태는 제로 상태이다. 초기 상태 s_0의 길이 제로인 survivor는 입력 무게 $I(\mathbf{u}^{(0)})$가 제로이고 출력 무게 $D(\mathbf{u}^{(0)})$가 제로인 $\mathbf{u}^{(0)}$이다. 여기서 유효 부호어 트렐리스의 부분 시퀀스를 $\mathbf{c}_{\mathrm{VCW}}(\mathbf{u}^{(i)})$라 정의하면 입력 무게 $I(\mathbf{u}^{(i)})$는 길이가 i인 부분 시퀀스 $\mathbf{u}^{(i)}$안에 있는 1의 개수이고 출력 무게 $D(\mathbf{u}^{(i)})$는 유효 부호어 트렐리스의 부분 시퀀스 $\mathbf{c}_{\mathrm{VCW}}(\mathbf{u}^{(i)})$안에 있는 1의 개수이다. 시간 i에서의 상태를 s_i라 정의한다. 그럴 경우 알고리즘은 다음과 같다.

- $1 \le i \le L$인 어떤 i에 대해서도,
- $s_i = \{0,1\}$인 어떤 s_i에 대해서도,
- $s_{i-1} = \{0,1\}$인 어떤 s_{i-1}에 대해서도,
- s_{i-1}안 어떤 survivor $\mathbf{u}^{(i-1)}$에 대해서도,

만약 $\mathbf{c}_{\mathrm{VCW}}(\mathbf{u}^{(i)})$가 유효 부호가 아니면, 다시 말해 $\mathbf{u}^{(i)}$에 있는 규칙적인(systematic) 비트들 중 하나의 제한(constraint)에 연결되어 있는 비트들이 같지 않으면, 버리고 유효 부호면 $I(\mathbf{u}^{(i)})$와 $D(\mathbf{u}^{(i)})$를 계산한다.

- 만약 $D(\mathbf{u}^{(i)}) > \mathrm{d}_{\min}$이면 $\mathbf{u}^{(i)}$를 버린다.
- 그렇지 않으면 $\mathbf{u}^{(i)}$를 s_i에 저장한다.
- q 번 반복된 규칙적인(systematic) 비트들의 제한(constraint)에 따라 $\mathbf{u}^{(i)}$의 반복을 채우고 그 외는 제로로 채워서 생성한 입력 시퀀스 $\mathbf{u}^{(L)}$에 대해 $D(\mathbf{u}^{(L)})$이 $d_{\min}$보다 작으면 $d_{\min}$을 $D(\mathbf{u}^{(L)})$로 update한다.

유효 부호어 검색 알고리즘은 RA 부호의 다른 graphical 모델을 기초로 하여 단지 유효 부호어에 대해서만 최소 거리를 검색하기 때문에 기존의 알고리즘보다[11] 계산량이 상당히 감소되었음은 주목할 만하다. 기존 알고리즘과 제안된 알고리즘의 복잡도에 대한 비교가 표 1에 되어 있다. 표 1에서 보여준 바와 같이 복잡도 감소는 인터리버의 길이가 증가함에 따라 지수적으로 더 감소하는 것을 알 수 있다.

표 1. 기존 알고리즘과 제안된 알고리즘의 계산량 비교 (단위는 검색 대상이 되는 부호어들의 개수의 상향 한계임, $q=4$)

입력블록길이, 인터리버길이	기존 알고리즘	제안된 알고리즘	복잡도 감소
100,400	2^{400}	2^{100}	$2^{400}-2^{100}$
500,2000	2^{2000}	2^{500}	$2^{2000}-2^{500}$
K,L	2^L	2^K	2^L-2^K

Ⅴ. LDPC 부호와 RA 부호의 최소 거리 결과

축적기와 4번 반복 부호를 가진 규칙적인(systematic) RA 부호의 최소 거리가 계산되어졌다. 패리티 비트가 puncturing된 후의 RA 부호의 부호율은 7/8과 1/2이다. 인터리버의 크기 L은 2000이고 pseudo random 인터리버가 사용되어졌다. 그림 5에서는 부호율 7/8에서 최소 거리항과 낮은 거리항을 포함하는 유니온 바운드와 20번의 고정 반복 복호 시뮬레이션 성능이 비교되어 있다. 표 2에는 부호율 7/8과 1/2에 대하여 최소 거리를 포함한 무게 분포가 요약되어 있다.

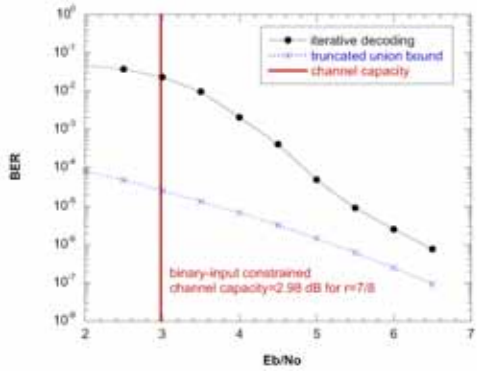

그림 5. 부호율 7/8, $L=2000$인 RA 부호에 대한 최소 거리항과 낮은 거리항을 포함하는 유니온 바운드와 고정 20번 반복 복호 시뮬레이션 성능 비교

표 2. 부호율 7/8과 1/2인 RA 부호에 대한 최소 거리항과 낮은 거리항을 포함하는 무게 분포

Rate r	Distance d	Number of codewords	Sum of input weights
7/8	2	1	1
7/8	3	8	12
7/8	4	8	16
7/8	5	4	9
1/2	7	1	2
1/2	10	2	6

www.dbpia.co.kr

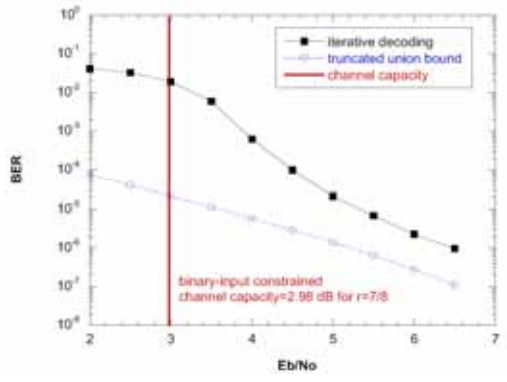

그림 6. $(N,K)=(800,700)$, 부호율 7/8인 LDPC 부호에 대한 최소 거리항과 낮은 거리항을 포함하는 유니온 바운드와 고정 50번 반복 복호 시뮬레이션 성능 비교

제안된 알고리즘을 $(N,K)=(800,700)$인 일반적인 LDPC 부호에 또한 적용하였다. LDPC 부호의 부호율은 7/8이 사용되었고 무작위(random) 숫자 생성기를 사용하여 패리티 행렬의 1의 위치를 정하는 방식으로 LDPC 부호를 생성하였다. 본 LDPC 부호에 대해 $d_{\min}$은 2이고 $d_{\min}$=2인 부호어의 개수는 2개로 구하여 졌다. $d_{\min}$=4인 부호어의 개수는 38개 이었다. 그림 6에서는 부호율 7/8에서 최소 거리항과 낮은 거리항을 포함하는 유니온 바운드와 50번의 고정 반복 복호 시뮬레이션 성능이 비교되어 있다.

제안된 알고리즘은 RA 부호와 유사한 여러 종류의 부호들에 쉽게 적용될 수 있다. 예를 들면 다수의 (보통 4개) 근간 zigzag 부호가 브로드캐스터로 병렬 결합된 zigzag 부호에서는 브로드캐스터가 반복 부호의 역할을 하며 각각의 근간 zigzag 부호는 puncturing을 한 축적기로 나타내어진다[12]. 그러므로 결합된 zigzag 부호의 최소 거리도 또한 제안된 알고리즘으로 계산될 수 있다.

Ⅵ. 결 론

본 논문은 반복 부분을 이용하여 단지 유효한 부호어만을 검색함으로서 RA 부호의 최소 거리의 계산량을 줄였다. LDPC 부호도 RA 부호와 같이 반복 부분을 가지므로 제안된 알고리즘은 LDPC 부호의 최소 거리 계산에도 적용되었다. RA 부호와 LDPC 부호의 관계를 자세하게 살펴보았다. LDPC 부호와 RA 부호의 최소 거리를 계산하였고 에러 마루를 얻기 위해 사용되었으며 이 에러 마루를 반복 복호의 성능과 비교하였다.

참 고 문 헌

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261-1271, October 1996.

[2] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, pp. 909-926, May 1998.

[3] R. Gallager, "Low Density Parity Check Codes," MIT press, 1963.

[4] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *Electronic Letters*, vol. 33, pp. 457-458, March 1997.

[5] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *Communication letters*, vol. 5, pp. 58-60, February 2001.

[6] T. M. Duman and M. Salehi, "New performance bounds for turbo codes," *IEEE Trans. Commun.*, vol. 46, pp. 565-567, May 1998.

[7] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1284-1292, July 1994.

[8] I. Sason and S. Shamai, "Improved upper bounds on the decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum," *IEEE Trans. Inform. Theory*, vol. 46, pp. 24-47, January 2000.

[9] A. M. Viterbi and A. J. Viterbi, "Improved union bound on linear codes for the input-binary awgn channel with applications to turbo codes," *Proc. IEEE Symposium on Information Theory*, p. 29, August 1998.

[10] H. J. D. Divsalar and R. J. McEliece, "Coding Theorems for "Turbo-like" Codes," *2Pro. 36th Annual Allerton Conf. on Commun.,*

www.dbpia.co.kr

Contol, and Comp, *Monticello, IL, USA,* pp. 201-210, September 1998.

[11] R. Garello, P. Pierleoni, and S. Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: Algorithms and applications," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 800-812, May 2001.

[12] L. Ping, and K. Y. Wu, "Concatenated tree codes: A low-complexity, high-performance approach," *IEEE Trans. Inform. Theory*, pp. 791-799, February 2001.

[13] G. D. Forney, "The Viterbi Algorithm," *Proc. of IEEE,* pp. 268-278, March 1973.

정 규 혁 (Kyuhyuk Chung) 정회원

1997년 2월 성균관대학교 전자공학과 졸업

1998년 12월 University of Southern California 전기 공학과 석사

2003년 12월 University of Southern California 전기 공학과 박사

1999년 8월~2000년 5월 미국 Integrated Device Technology, Inc., Member of Technical Staff

2001년 5월~2002년 5월 미국 TrellisWare Technology, Inc., Senior Engineer

2004년 2월~2005년 8월 LG전자 이동통신기술연구소 표준화그룹 선임연구원

2005년 9월~현재 단국대학교 정보컴퓨터학부 전임강사

<관심분야> Channel Coding, Iterative Detection, IEEE 802.16e LDPC code, Mobile Internet, Contents Technologies (CT)

www.dbpia.co.kr