

# 검사노드 갱신 과정에서 발생하는 중복 연산을 제거한 수정된 Extended Min-Sum LDPC 복호기

김민호, 박경빈, 정기석\*  
한양대학교

minhokim1000@hanyang.ac.kr, lay1523@hanyang.ac.kr, \*kchung@hanyang.ac.kr

## Modified Extended Min-Sum LDPC Decoder Eliminating Overlapped Operations of Check Node Update

Kim Min-Ho, Park Kyeong Bin, Chung Ki-Seok\*  
Hanyang University, Seoul, Korea

### 요 약

크기가 2 보다 큰 유한체를 갖는 비이진 저밀도 패리티 검사 부호 (Non-Binary Low Density Parity Check Codes, NB-LDPC)는 채널통신과정에서 발생하는 데이터 오류를 정정하는 기법 중 하나로 뛰어난 오류정정 성능을 갖고 있다. 하지만 검사노드의 매우 높은 연산 복잡도 때문에 이를 줄이기 위한 연구가 활발히 이루어지고 있다. 확장된 최소합 (Extended Min-Sum, EMS) 알고리즘은 NB-LDPC 복호 알고리즘 중 하나로 검사노드에서 신뢰도가 높은 원소들만으로 연산을 함으로써 연산 복잡도를 낮춘 것이 특징이다. 하지만 고려되지 않은 나머지 원소들로 인해 전체 메시지를 모두 채우지 못하는 문제가 발생한다. 이를 해결하기 위해 EMS 알고리즘은 채우지 못한 메시지에 대해 전체 원소들의 신뢰도를 이용하여 다시 한 번 연산을 수행한다. 본 논문에서는 이러한 과정에서 발생하는 중복 연산을 줄이면서 복호 성능은 그대로 유지하는 방법을 제안한다.

### I. 서 론

유한체 (Galois field, GF)의 크기가 2, 즉 GF(2) 인 이진 LDPC 부호는 채널 용량에 근접하는 오류정정 성능을 보여주지만 [1], 부호어의 길이가 짧을 때에는 오류정정 능력이 현저히 감소한다. 이에 반해 유한체의 크기가 2 보다 큰 (GF(q),  $q > 2$ ) 비이진 LDPC 부호는 부호어의 길이가 짧아도 오류정정 능력이 뛰어나다는 장점이 있다 [2].

하지만 비이진 LDPC 부호는 연산의 복잡도가 매우 높는데, 이를 보완하기 위해 제안된 EMS 알고리즘은 메시지 벡터의 전체 심볼  $q$  개 중에서 신뢰도가 높은  $n_m$  개의 심볼들만 절단 (truncation)하여 연산을 수행함으로써 연산의 복잡도를 낮추었다 [3]. 하지만 메시지 절단 과정에서 잃어버린  $q - n_m$  개의 심볼들의 정보로 인해 메시지 벡터의 전체 심볼  $q$  개 중 신뢰도를 갱신하지 못한 심볼이 생기고, 이러한 심볼을 empty entry 라 한다. EMS 알고리즘에서는 empty entry 를 채우기 위해 다시 메시지를 절단하기 전인 전체 심볼  $q$  개를 모두 고려하여 다시 한 번 연산을 수행하는데, 이때 메시지 절단 전의 연산과 중복되는 부분이 발생하게 된다.

본 논문에서는 기존 EMS 알고리즘에서의 empty entry 를 채우기 위한 방식을 개선하여 중복되는 연산량은 줄이고 오류정정 능력은 전혀 하락하지 않는 연산 방식을 제안한다.

본 논문의 구성은 다음과 같다. II절에서는 EMS 알고리즘 과정과 수정된 EMS 알고리즘 (MEMS)의 중복연산 제거 방법에 대해 설명한다. 이후 실험을 통해 두 알고리즘의 실행시간 및 복호성능을 비교한다. III절에서는 결론을 맺는다.

### II. 본 론

#### 1. EMS 알고리즘

비이진 LDPC 부호의 복호를 위한 EMS 알고리즘은 초기화, 신드롬 확인, 검사노드 연산, 변수노드 연산 네 단계로 구성된다. 이때 모든 연산은 각 노드의 메시지 벡터를 기준으로 이루어지는데, 메시지 벡터는 유한체의 원소, 즉 심볼과 그 심볼이 갖는 신뢰도 값으로 구성된다. 심볼의 신뢰도는 Log-Likelihood Ratio (LLR)로 표현된다.

GF(q)인 비이진 LDPC 부호의 변수노드와 검사노드의 개수가 각각  $N = \{n | n \in [0, N-1]\}$ ,  $M = \{m | m \in [0, M-1]\}$ 일 때, 초기화 단계에서는  $N$  개의 변수노드로 입력되는  $N$  개의 메시지 벡터를 생성한다. 이때  $n$  번째 변수노드로 입력되는 메시지 벡터는 심볼  $a$  ( $a \in [0, q-1]$ )와 심볼의 LLR  $\gamma_n(a)$ 으로 이루어지는데 이는 수식 1 을 통해 얻을 수 있다.

$$\gamma_n(a) = \log \frac{P(x_n=0|y_n)}{P(x_n=a|y_n)}$$

수식 1  $n$  번째 변수노드로 입력되는 메시지 벡터 초기화

이 때 수식 1 의  $x_n$ 은 전송된 실제 부호어의  $n$  번째 심볼을 의미하고,  $y_n$ 은 채널을 통해 전달받은  $n$  번째 noisy 심볼을 의미한다. 수식 1 의  $P(x_n=a|y_n)$ 부분은 채널로부터 전달받은 값이 실제로 전달한 심볼 값일 확률을 계산한 것이다. 여기서  $P(x_n=0|y_n)$ 은 심볼이 0 으로 고정된 값을 갖는다. 따라서  $P(x_n=a|y_n)$ 의 전달받은 심볼  $a$  가 전송하고자 했던 심볼  $a$  일 때 확률은 1 에 가까워지며, log 값은 작아지는 모습을 보인다. 따라서 LLR 이 작을수록 신뢰도 값이 높아진다.

초기화 이후 검사노드에서는 각 검사노드와 연결된 변수노드들로부터 전달받은 메시지 벡터들  $\alpha_{m,n}(a)$ 를 기반으로 다시 변수노드로 보낼 메시지 벡터들  $\beta_{m,n}(a)$ 를 생성한다. 검사노드의 연산은 그림 1 과 같이 이루어지는데, 이때 메시지 벡터들은 신뢰도가 높은  $n_m$  개의 심볼만으로 연산을 수행한다.

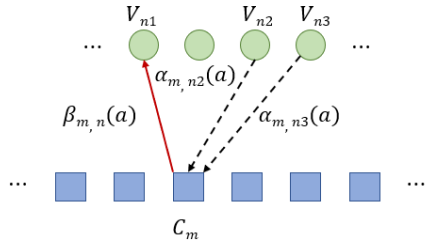


그림 1  $d_c=3$  인 검사노드 메시지 연산

그림 1 과 같이 검사노드와 연결된 변수노드의 개수를  $d_c$  라 할 때,  $i$  번째 ( $i \in [0, d_c-1]$ ) 변수노드로 전달할 메시지는  $i$  번째 변수노드로부터 전달받은 메시지를 제외한 나머지  $d_c-1$  개의 메시지만으로 연산한다. 이러한 전체 하에 검사노드에서 출력 메시지는 각 심볼에 대한 LLR 값을 갱신하는데, 이 때 출력 메시지의 심볼은 수식 2 와 같은 패리티 검사 방정식을 만족하는 조합에 따라 연산한다. 이때  $h_{mn}$  은 패리티 체크 행렬  $H$  의  $m$  행  $n$  열의 0 이 아닌 원소 값을 의미하고,  $v_n$  ( $v_n \in [0, q-1]$ )은 부호어의  $n$  번째에 위치한 변수노드의 심볼 값을 나타낸다.

$$\sum_0^{d_c-1} h_{mn} v_n = 0$$

수식 2 패리티 검사 방정식

수식 2 에서의 덧셈과 곱셈 연산은 유한체 덧셈을 의미하고, 수식 2 를 만족하는 심볼의 조합에 따라 수행되는 검사노드의 연산은 수식 3 과 같다. 수식 3 의  $\beta_{m,n}(a)$ 는  $n$  번째 변수노드로 보내는 출력 메시지 값이다. 여기서  $H(m) \setminus \{n\}$ 은  $m$  번째 검사노드와 연결된  $d_c$  개의 변수노드들 중에서  $n$  번째 변수노드를 제외한 나머지 변수노드들을 의미한다. 연산 대상은  $n'$ 번째에 있는 변수노드들이 수식 2 의 패리티 검사 방정식을 충족할 때 심볼  $a_{n'}$ 와 LLR  $\alpha_{m,n'}(a_{n'})$  값에 대해 각각 덧셈 연산을 수행한다. 이후 출력된 심볼에 대응하는 LLR 값들 중 최솟값을 뽑아 심볼과 함께 출력 메시지로 보내게 된다.

$$\beta_{m,n}(a) = \min[\sum_{n' \in H(m) \setminus \{n\}} \alpha_{m,n'}(a_{n'})]$$

수식 3 EMS 의 검사노드 연산

각 검사노드에서 수식 2 를 만족하는 심볼의 조합을 local configuration set 이라 하고,  $n_m$  개로 절단된 메시지에 대한 local configuration set 을  $\text{conf}(n_m)$ 이라 표현한다.

$\text{conf}(n_m)$ 을 기준으로 검사노드 연산을 수행할 때, 해당 조합으로 생성되지 않는 심볼(empty entry)이 발생할 수 있는데 EMS 알고리즘에서는 이러한 심볼들을 다시  $\text{conf}(q)$ 를 기준으로 연산하여 신뢰도 값을 갱신한다. 자세한 내용은 [3]에서 알 수 있으며, 이후의 변수노드와 신드롬 확인 과정은 생략한다.

본 논문에서는 이러한 empty entry 를 채우는 과정에서 발생하는 불필요한 중복 연산을 없앤 MEMS 알고리즘을 제안한다.

## 2. MEMS 알고리즘

MEMS 알고리즘은 기존 EMS 알고리즘의 복호 과정을 기반으로 한다. 그러나 EMS 알고리즘은 empty entry 문제가 발생했을 때 검사노드 갱신 단계에서  $\text{conf}(q)$ 에 대해 다시 한 번 연산함으로써 앞서 수행한  $\text{conf}(n_m)$  만큼의 불필요한 중복 연산이 발생한다. 이를 해결하고자 MEMS 알고리즘에서는 기존 EMS 알고리즘의  $\text{conf}(q)$ 의  $q$  개의 원소 값에서 이미 연산과정에 사용한  $n_m$  개의 메시지를 제외한  $q-n_m$ 에 대해서만 검사노드 연산을 수행한다.

$$\left\lfloor \frac{q-n_m}{q} \right\rfloor \times \# \text{ of Empty Entry}$$

수식 4 empty entry 발생시 추가되는 검사노드 연산량

수식 4 는 MEMS 알고리즘에서 empty entry 문제가 발생한 경우에 기존 EMS 알고리즘에서  $\text{conf}(q)$ 를 고려해준 것과 다르게  $\text{conf}(q-n_m)$ 의 검사노드 연산과정을 수행했을 때 연산량 감소 비율을 구하기 위한 식이다.

## 3. 실험결과 및 분석

본 논문에서는 Intel 社의 i7-6700 에서 실험을 진행하였다. 해당 플랫폼은 3.40GHz 로 동작하는 쿼드코어 프로세서를 포함하고 있으며, 16GB 메모리를 사용하였다.

MEMS 및 EMS 알고리즘을 구현하여 실험을 진행하였으며, 각각 OpenMP 를 적용하였다.

두 알고리즘은 GF(64) (576, 480) 정규 (2, 12) 비이진 LDPC 코드에서  $n_m$  을 15, 최대 frame 을 100000, 그리고 최대 반복 횟수를 15 로 설정하였고, 이때 MEMS 및 EMS 알고리즘의 실행시간 및 성능을 평가하였다.

전체 실행시간은 GF(q) 값이 정해졌을 때,  $n_m$  의 크기가 커질수록 실행시간의 개선 정도가 높아진다. 여기서  $n_m=15$  이고 GF(64) 일 때 MEMS 알고리즘은 EMS 알고리즘과 비교해서 약 2% 정도의 전체 실행시간 개선을 보였다.

그림 2 를 통해 두 알고리즘의 Signal to Noise Ratio (SNR) 변화에 따른 Bit Error Rate (BER)이 동일한 것을 확인할 수 있다. MEMS 알고리즘에서는 중복연산만 제거하였기 때문에 복호 성능은 EMS 알고리즘의 성능을 그대로 유지하였다.

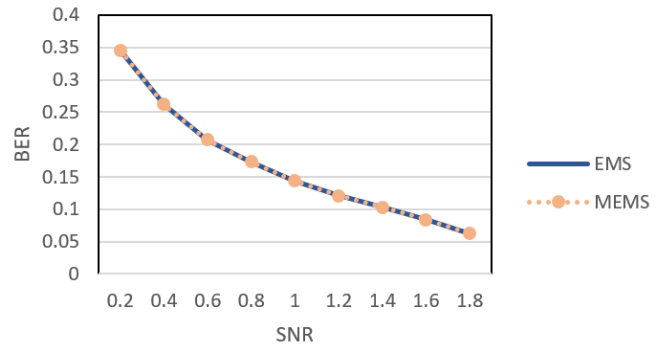


그림 2 제안된 알고리즘의 Error Rate 성능

## III. 결론

본 논문에서는 EMS 알고리즘의 반복 복호 과정에서 발생한 중복 연산을 제거하면서도 EMS 알고리즘과 동일한 복호 성능을 유지하는 MEMS 알고리즘을 제안하였다. MEMS 알고리즘은 검사노드 연산과정에 사용한 신뢰도 높은 원소들을 제외한 나머지 원소들에 대해서 다시 한 번 연산을 수행한다. 실험을 통해, MEMS 알고리즘은 기존 EMS 알고리즘과 동일한 오류정정 성능을 보여주면서도 연산량은 효과적으로 줄어든 것을 확인할 수 있었다.

## ACKNOWLEDGMENT

이 논문은 2019 년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임 (N0001883, 2019 년 산업전문인력역량강화사업)

## 참고 문헌

- [1] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching low-density parity check codes", IEEE Trans. Inf. Theory, vol. 47, pp. 619-637, Feb. 2001
- [2] M. Davey, D. MacKay, "Low density parity check codes over GF(q)," IEEE Commun. Lett., vol 2, no. 6, Jun. 1998.
- [3] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," IEEE Trans. Commun., vol. 55, no. 4, pp. 633-643, Apr. 2007.