

저밀도 패리티 검사부호에서 오류마루 감소를 위한 수정 합-곱 알고리즘

정회원 유 석 근*, 강 석 근**, 종신회원 주 언 경*°

A Modified Sum-Product Algorithm for Error Floor Reduction in LDPC Codes

Seog Kun Yu*, Seog Geun Kang** *Regular Members*, Eon Kyeong Joo*° *Lifelong Member*

요 약

본 논문에서는 저밀도 패리티 검사부호의 복호에서 발생하는 트래핑 세트에 갇힌 오류를 정정할 수 있는 수정 합-곱 알고리즘을 제안한다. 원래 합-곱 알고리즘과는 달리 제안된 방법은 반복복호의 실패 원인이 트래핑 세트인지를 판단하는 과정과 트래핑 세트에 갇힌 오류를 정정하는 2단계 복호과정으로 구성된다. 수정 합-곱 알고리즘에서는 트래핑 세트에 포함된 변수노드를 검색하기 위하여 실패 검사노드 집합과 경관정 비트들의 천이패턴을 이용한다. 그리고 검색된 변수노드의 정보를 반전시킨 다음 합-곱 알고리즘을 수행함으로써 트래핑 세트에 갇힌 오류를 정정한다. 모의실험 결과 제안된 복호알고리즘은 신호 대 잡음비의 증가에 따라 지속적으로 향상되는 오류성능을 보인다. 따라서 수정 합-곱 알고리즘은 저밀도 패리티 검사부호에서 발생하는 오류마루 현상을 현저히 감소시키거나 거의 제거할 수 있는 것으로 사료된다.

Key Words : LDPC Code, Modified Sum-Product Algorithm, Error Floor, Trapping Set

ABSTRACT

In this paper, a modified sum-product algorithm to correct bit errors captured within the trapping sets, which are produced in decoding of low-density parity-check (LDPC) codes, is proposed. Unlike the original sum-product algorithm, the proposed decoding method consists of two stages. Whether the main cause of decoding failure is the trapping sets or not is determined at the first stage. And the bit errors within the trapping sets are corrected at the second stage. In the modified algorithm, the set of failed check nodes and the transition patterns of hard-decision bits are exploited to search variable nodes in the trapping sets. After inverting information of the variable nodes, the sum-product algorithm is carried out to correct the bit errors. As a result of simulation, the proposed algorithm shows continuously improved error performance with increase in the signal-to-noise ratio. It is, therefore, considered that the modified sum-product algorithm significantly reduces or possibly eliminates the error floor in LDPC codes.

※ 본 연구의 일부는 BK21 지원사업의 결과임. 또한 일부는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 결과임(NIPA-2010-C1090-1031-0007).

* 경북대학교 IT대학 전자공학부(amatess@ee.knu.ac.kr, ekjoo@ee.knu.ac.kr) (°:교신저자)

** 경상대학교 전기전자공학부(sgkang@gnu.ac.kr)

논문번호 : KICS2010-01-007, 접수일자 : 2010년 1월 6일, 최종논문접수일자 : 2010년 4월 27일

I. 서 론

저밀도 패리티 검사부호(low-density parity-check code)는 1960년대 초 Gallager^[1]에 의해 처음 소개되었다. 이 부호는 선형 블록부호의 일종으로 패리티 검사행렬이 매우 적은 수의 1로 구성되어 있다. 1990년대에 들어 Mackay 등^[2]은 부호어의 길이가 매우 긴 저밀도 패리티 검사부호의 오류 성능이 Shannon 한계에 근접할 수 있음을 입증하였다. 이에 따라 이 부호는 유럽의 위성 디지털 비디오 방송(digital video broadcasting-satellite; DVB-S)과 휴대 인터넷 표준 가운데 하나인 WiMAX에 채택되는 등 차세대 무선 통신시스템에의 응용이 확대되고 있다^{[3],[4]}.

Richardson 등^{[5],[6]}은 저밀도 패리티 검사부호에서도 오류마루(error floor) 현상이 발생할 수 있음을 보였다. 터보부호와는 달리 저밀도 패리티 검사부호의 오류마루 현상은 합-곱 알고리즘(sum-product algorithm; SPA)의 특수한 구조와 잡음 패턴의 복합적인 영향에 의하여 발생하는 트랩핑 세트(trapping set)가 주된 원인이다^{[5],[12]}. 이에 따라 최근에는 오류마루 현상이 나타나는 비트오율(bit error rate; BER) 영역을 감소시킬 수 있는 여러 가지 방법들이 연구되고 있다^{[8],[12]}. [8]과 [9]에서는 트랩핑 세트들의 사이클(cycle)에 따른 오류마루 현상을 분석함으로써 패리티 검사행렬 설계 시 가능한 한 최소 사이클의 크기를 늘리는 방법을 제시하였다. 또한 트랩핑 세트의 크기는 부호어의 길이에 비하여 매우 작으므로 저밀도 패리티 검사부호와 높은 부호율을 가지는 외부부호를 연결함으로써 트랩핑 세트를 해결하는 방법도 제안된 바 있다^[10]. 그리고 [11]과 [12]에서는 복호 알고리즘을 수정하여 트랩핑 세트에 의한 오류마루 현상을 억제하는 방법들이 제안되었다. 이와 같은 기존 복호 알고리즘들은 패리티 검사행렬에서 발생하는 트랩핑 세트의 구조와 위치에 대한 사전정보가 필요하다. 하지만 이 정보를 얻기 위해서는 사전에 방대한 모의실험이나 패리티 검사행렬을 이용한 검색이 요구된다. 이러한 문제점에 대처하기 위하여 [13]에서는 사전정보 없이 오류마루 현상을 감소시킬 수 있는 방안 대하여 연구하고 그 결과를 제시하였다. 하지만 이 기존 알고리즘은 복호실패의 원인이 트랩핑 세트에 의한 것인지 아닌지를 구분하지 않는다. 따라서 트랩핑 세트에 의한 오류가 아닐 경우에는 필요없는 추가 복호과정에 의해 복잡도가 증가할 가능성이 있다.

또한 부호어 길이와 부호율에 따라 트랩핑 세트의 검출을 위한 문턱값이 변할 수 있으므로 적합한 문턱값을 찾는 과정이 추가로 필요하다.

본 논문에서는 트랩핑 세트의 구조와 위치에 대한 사전정보 없이 저밀도 패리티 검사부호의 복호에서 발생하는 트랩핑 세트에 의한 오류마루 현상을 감소시킬 수 있는 수정 합-곱 알고리즘을 제안한다. 이 방법은 트랩핑 세트에 의한 오류가 아닐 경우 추가적인 복호과정을 거치지 않으며 문턱값이 저밀도 패리티 검사부호의 부호어 길이와 부호율에 상관없이 일정한 특징을 가지는 점에서 기존 방법과는 근본적으로 차이가 있다. 이러한 과정을 위하여 제안된 수정 합-곱 알고리즘은 반복복호에서 실패 검사노드(failed check node; FCN)와 비트오류들의 위치와 수, 그리고 비트 확률의 경관정 천이패턴을 이용한다. 제안된 알고리즘은 2단계의 합-곱 복호과정으로 구성된다. 우선 1단계에서는 원래 합-곱 복호과정을 수행하면서 실패 검사노드를 검색하고 비트 확률의 경관정 정보를 저장한다. 다음으로 2단계에서는 저장된 정보를 이용하여 트랩핑 세트에 포함된 변수노드(variable node; VN)를 탐색하여 이들의 정보를 반전시킨 다음 다시 합-곱 복호를 수행하여 트랩핑 세트에 갇힌 오류를 정정한다. 그 결과 제안된 알고리즘은 기존에 최적 문턱값을 사용하여 가장 우수한 성능을 보인 방법^[13]처럼 오류마루 현상을 현저히 감소시키거나 거의 제거하였다. 하지만 알고리즘의 구현에 필요한 메모리 수의 측면에서는 기존의 방법에 비하여 43%이상 감소시키는 장점이 있는 것으로 나타났다.

본 논문의 구성은 다음과 같다. 2장에서는 저밀도 패리티 검사부호에서 발생하는 트랩핑 세트와 이로 인한 오류마루 현상의 발생 원인에 대하여 소개한다. 3장에서는 저밀도 패리티 검사부호의 반복복호에 트랩핑 세트가 미치는 영향에 대하여 서술한다. 4장에서는 본 논문에서 제안하는 수정 합-곱 알고리즘을 제시하고 동작원리를 상세히 소개한다. 그리고 5장에서는 가산성 백색 가우시안 잡음(additive white Gaussian noise; AWGN) 환경에서 제안된 알고리즘과 기존 알고리즘들의 모의실험 수행 결과를 제시하고 성능을 비교 분석한다. 마지막으로 6장에서 본 논문의 결론을 제시한다.

II. 트랩핑 세트

저밀도 패리티 검사부호는 선형 블록부호로서 1

의 수가 0의 수에 비하여 매우 적은 패리티 검사행렬을 사용한다. 패리티 검사행렬은 일반적으로 다음과 같이 정의된다.

$$\mathbf{H} = [h_{ij}], \quad 1 \leq i \leq M, 1 \leq j \leq L, \quad (1)$$

여기서 M 과 L 은 패리티 비트의 수와 부호어의 길이를 각각 나타낸다. 또한 패리티 검사행렬은 복호된 부호어에서 비트오류의 유무를 판별하기 위하여 (2)와 같은 신드롬 검사(syndrome test)에도 사용된다.

$$\mathbf{H}\mathbf{c}^T = \mathbf{0}, \quad (2)$$

여기서 \mathbf{c} 는 $1 \times L$ 부호어 벡터, \mathbf{c}^T 는 \mathbf{c} 의 전치행렬(transpose), 그리고 $\mathbf{0}$ 는 $M \times 1$ 영행렬(zero matrix)이다. 이러한 \mathbf{H} 행렬은 패리티 비트의 수에 해당하는 검사노드와 부호어의 길이에 해당하는 변수노드를 가지는 태너 그래프(Tanner graph)를 이용하여 표현할 수 있다^[4].

저밀도 패리티 검사부호의 복호기는 합-곱 알고리즘을 이용한 반복복호를 통하여 수신 신호를 복호한다. 여기서 복호 복잡도를 감소시키기 위하여 터보부호에 사용되는 최대가능성복호기(maximum-likelihood decoder)와는 달리 태너 그래프의 주변확률(marginal probability)은 일반적으로 정확하게 계산하지 않는다^[7]. 이로 인하여 합-곱 알고리즘에서는 낮은 무게(low weight)를 갖는 부호어와 \mathbf{H} 행렬의 구조적 특징에 의하여 복호에 실패하는 경우가 발생될 수 있다. 이와 같이 반복복호가 종료되어도 수정되지 않은 비트오류들의 집합을 실패집합(failure set)이라고 한다^[5]. 반복복호가 성공적으로 완료되면 실패집합은 공집합(null set)이 된다. 만약 실패집합이 공집합이 아닌 경우 반복복호는 신드롬 검사에 의하여 실패집합이 공집합이 될 때까지 계속 수행되며 미리 정해진 최대 횟수에 도달하면 종료된다. 그리고 최종적으로 복호에 실패한 비트오류들의 집합을 트래핑 세트라고 정의하며 (a, b) 트래핑 세트로 표시한다^[5]. 여기서 a는 태너 그래프에서 트래핑 세트에 해당하는 하위 그래프의 변수노드의 수를 나타내고 b는 변수노드와 이웃하면서 홀수 차수(degree)를 가지는 검사노드의 수이다. 차수는 트래핑 세트 내에서 임의의 노드에 연결된 노드의 수를 나타낸다.

트래핑 세트의 구조적인 특징과 이의 영향에 대한 이해를 위한 예로써 (5, 2) 트래핑 세트를 그림 1에 나타내었다. 여기서 $v_1 \sim v_5$ 는 트래핑 세트에 갇힌 변수노드들이고 이에 연결된 검사노드들은 $c_1 \sim c_6$ 이다. 검사노드들 가운데 c_1 과 c_6 는 각각 변수노드 v_1 과 v_5 와만 연결되므로 차수가 1이다. 여기서는 모든 비트가 0인 부호어를 전송하였으나 트래핑 세트에 속한 5개의 변수노드에 1이 수신된 것으로 가정한다.

변수노드들은 수신된 데이터를 연결된 검사노드로 보내고 검사노드는 이 데이터들을 모듈러-2 합(modular-2 sum)함으로써 합-검사를 수행한다. 모든 검사노드에서 합-검사 결과가 0이 되면 복호된 비트열을 합법적인 부호어로 판단하여 반복복호를 중단한다. 이를 위해서는 변수노드에 오류가 없거나 검사노드의 차수가 모두 짝수이어야 한다. 하지만 그림 1의 예에서 검사노드 c_1 과 c_6 는 1개의 변수노드와 연결되므로 합-검사 결과는 1이 된다. 따라서 이 검사노드는 트래핑 세트 외부에서 연결된 변수노드에서 받은 정보를 v_1 과 v_5 변수노드로 보낸다. 반면 검사노드 c_2, c_3, c_4, c_5 는 짝수 차수를 가지므로, 즉 연결된 변수노드가 짝수 개이므로 합-검사 결과는 0이 된다. 따라서 이들은 트래핑 세트 내부의 변수노드들에게 오류값인 1을 보낸다. 그 결과 변수노드 v_1 과 v_5 는 검사노드 c_1, c_6 으로부터 0을 수신하더라도 나머지 검사노드로 부터는 1이 수신되므로 오류가 수정되지 않는다. 이에 비하여 v_2, v_3, v_4 노드는 이들과 연결된 모든 검사노드들이 짝수 차수를 가지므로 합-검사 결과가 0이 된다. 이로 인하여 변수노드들에는 비트오류가 저장되어 있음에도 불구하고 복호기는 합-검사 결과에 따라 이를 정정하지 않으므로 오류 상태가 계속 유지된다. 따라서 트래핑 세트에 갇힌 변수노드의 오류는 반복복호를

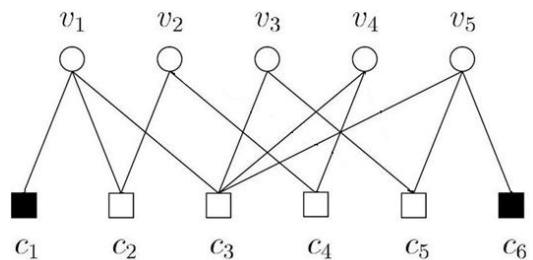


그림 1. (5, 2) 트래핑 세트의 예
Fig. 1. An example of (5, 2) trapping set

수행하더라도 수정되지 않는다. 이는 저밀도 패리티 검사부호의 오류성능에서 오류마루 현상이 발생하는 주된 원인이다.

III. 반복복호에 따른 트래핑 세트의 영향

수신된 부호어에 트래핑 세트가 존재하면 반복복호를 수행하는 동안 트래핑 세트 내부의 오류정보는 치수가 홀수인 검사노드와 연결된 변수노드에 의하여 외부로 전파될 수 있다. 이로 인하여 실패 검사노드의 수는 증가한다. 하지만 반복복호가 수행됨에 따라 트래핑 세트 외부의 비트오류는 정정되므로 실패 검사노드의 수와 비트오류의 수는 다시 감소되어 트래핑 세트 내부의 변수노드에서만 오류가 발견된다. 따라서 비트오류의 수와 실패 검사노드의 수는 반복복호 수행 횟수에 따라 증감을 반복하는 형태로 나타날 것으로 예상된다.

이를 확인하기 위해서 부호어의 길이가 1024비트이고 부호율이 3/4인 비균일 랜덤 저밀도 패리티 검사부호(irregular random LDPC codes)에 대하여 합-곱 알고리즘을 이용한 반복복호를 수행하여 트래핑 세트 발생 유무를 조사하였다. 여기서 열무게(column weight)의 분포는 무게 2인 열이 30%, 3인 열이 60%, 그리고 7인 열이 10%가 되도록 정하였다. 그 결과 신호 대 잡음비(signal-to-noise ratio; SNR)가 $E_b/N_0=4.4\text{dB}$ 일 때 (14, 4) 트래핑 세트가 발생되었다. 이 트래핑 세트로 인한 실패 검사노드와 비트오류의 패턴은 그림 2와 같다. 여기서 가로축은 수행된 반복복호의 횟수를 나타낸다. 그리고 그림 2(a)와 (b)의 세로축은 검사노드의 위치와 부호어에서 비트의 위치를 각각 나타낸다. 실패 검사노드는 그림 2(a)에서 점으로 표시하였으며 따로 표시된 부분은 트래핑 세트를 나타낸다. 그리고 그림 2(b)에서 점으로 표시된 것은 오류를 가지는 변수노드이다.

예상된 바와 같이 반복복호의 횟수에 따라 실패 검사노드의 수가 크게 증가하는 부분과 20개 이하의 적은 수를 유지하는 부분이 반복적으로 나타나는 것을 그림 2(a)에서 확인할 수 있다. 특히 적은 수를 유지하는 부분에서 실패 검사노드의 대부분은 트래핑 세트에 속한다. 또한 비트오류 수의 증감 패턴도 실패 검사노드 수의 증감과 거의 일치함을 그림 2(b)를 통하여 알 수 있다. 그리고 비트오류의 수가 적은 부분에서 나타나는 대부분의 오류는 트래핑 세트에 포함된 변수노드인 것으로 나타났다.

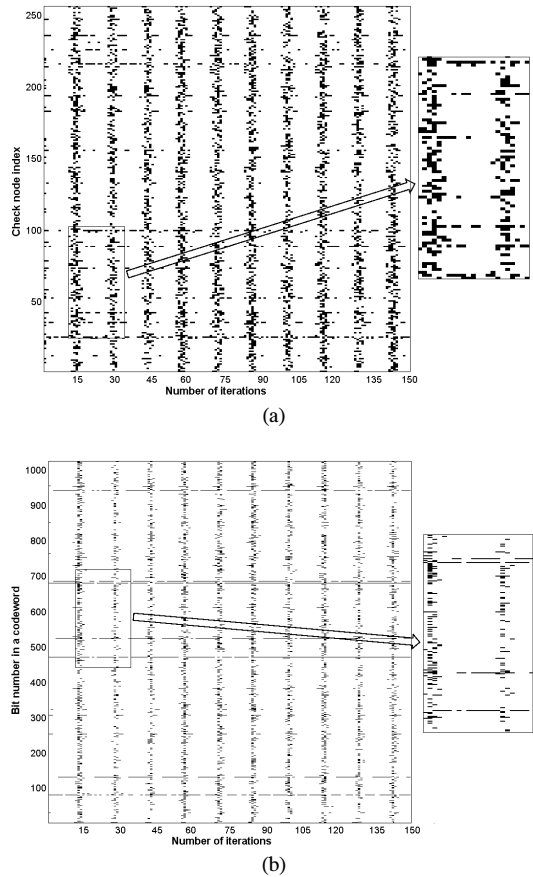


그림 2. 반복복호에 따른 (14, 4) 트래핑 세트의 영향 (a) 실패 검사노드 패턴, (b) 비트오류 패턴
Fig. 2. The influence of a (14, 4) trapping set according to the iterations (a) The patterns of FCNs, (b) The patterns of bit errors

트래핑 세트 내의 오류는 합-곱 복호 알고리즘에 의해서도 수정되지 않는다. 반복복호를 수행함에 따라 트래핑 세트 외부의 오류는 수정되면서 비트오류의 수가 급격히 감소한다. 하지만 실패한 검사노드를 통하여 트래핑 세트 내부의 오류정보가 외부로 전파되면서 비트오류의 수가 다시 증가하는 현상이 계속하여 반복된다.

만약 그림 2(b)에 나타난 비트오류 패턴에 대한 정보를 수신기가 활용할 수 있으면 트래핑 세트에 속한 변수노드들을 탐지하는 것이 가능하다. 하지만 수신기는 비트오류의 위치를 알 수 없으므로 이 패턴의 정보를 직접적으로 이용할 수 없다. 이에 대한 대안으로 수신기는 비트 확률의 경관정 천이패턴을 이용하여 그림 2(b)에 나타난 비트오류 패턴과 유사한 정보를 얻을 수 있다. 이는 반복복호가 수행됨에 따라 트래핑 세트에 포함된 변수노드들의 비트 확

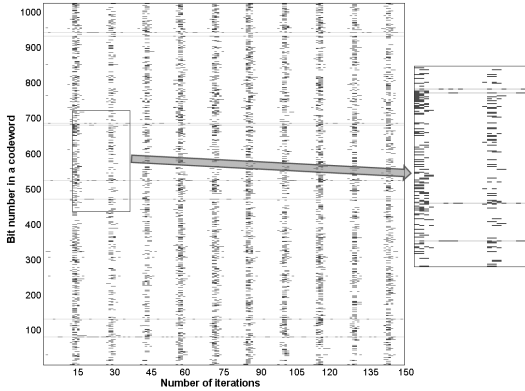


그림 3. 반복복호에 따른 경관정 비트들의 천이패턴
Fig. 3. The transition patterns of hard-decision bits according to the iterations

률에 대한 경관정 값의 변화가 빈번하게 발생하는 특성을 이용한다.

예를 들어 그림 2(b)에 보인 오류패턴에 대한 비트 확률의 경관정 천이패턴을 그림 3에 나타내었다. 여기서 점으로 나타낸 것은 현재 복호에서의 경관정 결과가 이전 복호 때와 서로 다른 경우를 의미한다.

트래핑 세트에 속하는 변수노드들은 비트오류의 수가 매우 적은 부분에서도 경관정 값의 천이가 자주 발생함을 알 수 있다. 따라서 실패 검사노드의 수가 현저히 적은 부분에서 이에 연결된 변수노드 가운데 천이를 일으키는 변수노드를 선택하면 트래핑 세트에 포함된 변수노드의 일부 또는 전부를 검출할 가능성이 매우 높다. 이러한 변수노드는 반복 복호에도 불구하고 정정되지 않는 오류정보를 가지고 있으므로 이들을 반전시켜 합-곱 알고리즘을 수행하면 트래핑 세트 구조를 해체할 수 있을 것으로 예상된다.

IV. 수정 합-곱 알고리즘

원래의 합-곱 알고리즘은 반복복호를 수행하더라도 트래핑 세트에 갇힌 비트오류를 정정하지 못한다. 이러한 문제를 해결하여 오류마루 현상이 발생되지 않거나 발생되더라도 오류마루가 나타나는 비트오류 영역을 현저히 감소시킬 수 있는 방안이 필요하다. [11]과 [12]에서는 충분한 모의실험을 통해 발생할 수 있는 트래핑 세트의 구조와 위치에 대한 사전정보를 얻는 방법을 이용하였다. [13]에서는 변수노드가 오류일 경우의 초기 확률값 특성과 비트

확률의 경관정 천이패턴을 이용하는 방법이 제시되었다. 본 논문에서 제안하는 수정 합-곱 알고리즘은 트래핑 세트에 대한 사전정보를 필요로 하지 않는 방법이며 또한 [13]의 방법과는 달리 오류 변수노드의 특징인 경관정 천이패턴만 뿐만 아니라 실패 검사노드의 패턴도 동시에 이용한다. 이는 그림 4의 흐름도에 나타낸 바와 같이 2단계 복호과정으로 이루어진다. 여기서 I_{max1} 과 I_{max2} 는 1단계와 2단계 복호과정에서 반복복호의 최대 수행 횟수를 각각 나타낸다. 그리고 n 은 수행된 복호 횟수를 나타내는 카운터이다. 원래의 알고리즘에서 수정되었거나 추가된 부분은 점선 블록으로 표시하였다.

수정 합-곱 알고리즘은 원래 알고리즘처럼 각 변수노드의 초기 확률값을 계산하는 것으로 시작한다. 하지만 제안된 알고리즘에는 합-곱 알고리즘 수행 후 실패 검사노드와 비트 확률의 경관정 천이결과를 저장하는 과정이 추가되었다. 이 정보는 2단계 복호과정에서 트래핑 세트에 포함된 변수노드를 검출하는데 이용된다. 다음으로 신드롬 검사를 수행하여 그 결과가 0이면 (2)가 만족되므로 반복복호를 종료한다. 하지만 신드롬 검사결과가 0이 아니면 I_{max1} 으로 정해진 1단계의 최대 횟수까지 반복복호를 수행한다. 제안된 알고리즘은 기존에 제시된 SCMG_{multi} 기법^[13]과는 달리 1단계의 반복복호가 완료되었음에도 불구하고 (2)가 만족되지 않으면 그 원인이 트래핑 세트인지를 판단하여 트래핑 세트가 원인인 아니면 1단계 과정만으로 복호를 종료한다.

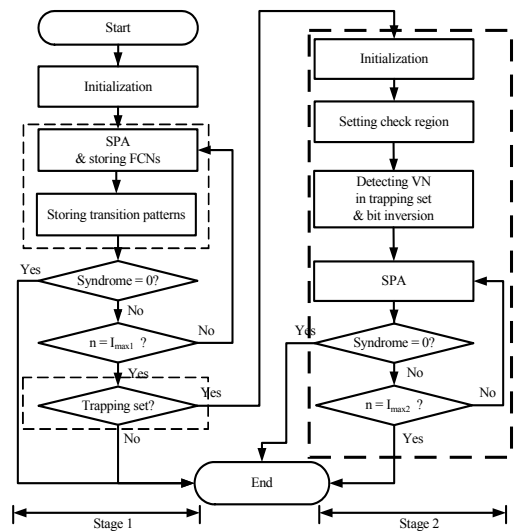


그림 4. 수정 합-곱 알고리즘의 흐름도
Fig. 4. Flow chart of the modified sum-product algorithm

이는 트래핑 세트에 의한 복호오류가 아닐 경우 불필요한 추가 연산에 의한 복호 복잡도의 증가와 시간지연을 방지한다. 하지만 트래핑 세트가 원인인 것으로 판단되면 2단계 복호과정이 시작된다. 제안된 알고리즘에서는 이 판단을 위한 기준으로 실패 검사노드의 최소 개수를 이용한다.

실패 검사노드의 개수가 가장 적은 경우 비트 오류의 수도 가장 적어지는 경향이 있으므로 2단계 복호과정은 모든 변수노드의 비트 확률값을 이때의 값으로 초기화시키는 것으로 시작된다. 그리고 트래핑 세트에 해당하는 변수노드를 찾기 위하여 1단계에서 저장된 실패 검사노드 패턴과 경관정 천이패턴에 대한 적절한 검사영역(check region)을 설정한 다음 해당 영역에 존재하는 실패 검사노드들의 집합을 만든다. 그리고 비트 확률값의 경관정 천이패턴을 이용하여 데이터의 천이가 발생하면서 실패 검사노드와 연결된 변수노드들의 집합을 만든다. 트래핑 세트에 포함된 변수노드들은 이 세트에 속한 검사노드에만 연결되는 특성이 있으므로 검색된 변수노드에 연결된 모든 검사노드가 실패 검사노드 집합의 원소이면 이 변수노드는 트래핑 세트에 포함된 것으로 간주한다. 이러한 방법으로 구한 변수노드들의 비트를 반전시켜 오류의 일부를 인위적으로 수정한 다음 반복복호를 수행함으로써 트래핑 세트 구조를 해체하고 이에 포함된 오류를 정정한다.

V. 모의실험 및 분석

제안된 복호알고리즘의 성능 평가를 위하여 몬테카를로(Monte-Carlo) 방식으로 모의실험을 수행하였다. 여기서는 부호어 길이가 512비트이면서 1024비트와 동일한 무게 분포와 부호율을 가지는 저밀도 패리티 검사부호를 사용하였다. 그리고 변조 방식은 2진 위상편이변조(binary phase shift keying; BPSK)를 이용하였고 전송환경은 평균이 0이고 분산이 σ^2 인 가산성 백색 가우시안 잡음환경으로 하였다.

원래 알고리즘의 최대 반복복호 횟수는 150회로 충분히 크게 설정하였다. SCMG_{multi} 기법은 1단계 반복복호 150회 동안 오류가 수정되지 않으면 천이에 대한 문턱값 N 과 초기 확률값에 대한 문턱값 mag 을 이용하여 선택된 변수노드의 초기 확률값을 0.5로 설정하였다^[13]. 이 때 초기 확률값에 대한 문턱값은 $mag=3,4,5$ 로 하였다. 그리고 천이에 대한 문턱값은 부호어 길이가 512비트인 경우 $N_{512}=65$

로, 1024비트인 경우에는 $N_{1024}=128$ 로 하였다. 이 값들은 $E_b/N_0=4.6dB$ 에서 부호어 길이에 따라 문턱값을 변형하며 모의실험을 수행하여 최적의 성능을 보인 경우이다. 그 후, 각 mag 에 따라 10회씩의 최대 반복복호수를 설정하여 최대 30회의 추가 반복복호를 수행하도록 구성하였다. 제안된 알고리즘 역시 1단계 최대 반복복호 횟수는 $I_{max1} = 150$ 회로 원래 복호알고리즘과 동일하게 정하였으며 2단계에서는 $I_{max2} = 30$ 회로 정하였다. 또한 수정 합-곱 알고리즘의 2단계 복호과정 수행을 결정하기 위하여 트래핑 세트가 반복복호 실패의 원인인지를 판단하는 기준인 실패 검사노드의 최소 개수는 [6]의 실험 결과에 따라 5로 정하였다. 그리고 2단계 과정에서 검사영역을 결정하는 기준도 실패 검사노드의 수가 5보다 작은 영역으로 설정하였다.

5.1 오류성능 분석

제안된 알고리즘을 이용하여 복호한 오류패턴 결과를 그림 5에 나타내었다. 그림 2(b)의 결과와 비교하면 반복복호가 150회에 도달할 때까지는 제안된 알고리즘에서도 1단계 복호과정이 수행되므로 비트오류 패턴의 결과들이 서로 일치한다. 하지만 제안된 알고리즘에서는 2단계 복호가 시작되어 겨우 2회의 반복복호만 수행하여도 트래핑 세트 내부의 오류를 포함한 모든 오류가 수정되는 것을 확인할 수 있다. 따라서 제안된 수정 합-곱 알고리즘은 트래핑 세트로 인한 오류마루 현상을 현저히 감소시키거나 제거할 수 있을 것으로 예상된다.

이를 확인하기 위하여 기존 알고리즘과 제안된 알고리즘을 적용한 저밀도 패리티 검사부호의 비트

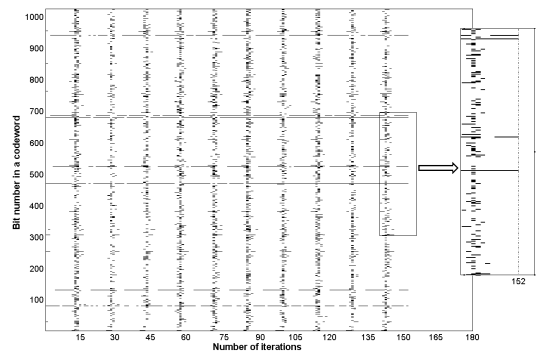


그림 5. 수정 합-곱 알고리즘에서 반복복호에 따른 비트오류 패턴
Fig. 5. The patterns of bit errors in the modified SPA

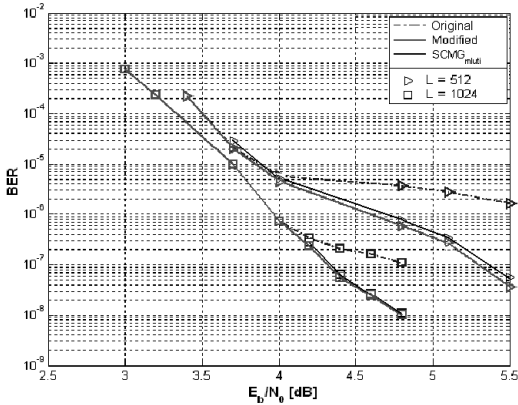


그림 6. 복호알고리즘에 따른 저밀도 패리티 검사부호의 오류성능
 Fig. 6. Error performance of the LDPC codes according to decoding algorithms

오율을 그림 6에 나타내었다. 여기서 오류마루 현상이 발생하지 않은 신호 대 잡음비 영역인 $E_b/N_0 = 3.0 \sim 4.0 \text{ dB}$ 에서는 10^5 개 부호어를 모의실험에 사용하였다. 오류마루 현상이 발생하는 $4.1 \sim 5.5 \text{ dB}$ 영역에서는 복호오류가 발생한 부호어 표본을 충분히 얻기 위하여 3×10^8 개의 부호어를 사용하였다.

신호 대 잡음비가 낮은 영역에서는 원래의 합-곱 알고리즘과 기존 SCMG_{multi} 기법, 그리고 제안된 알고리즘이 거의 동일한 비트오율을 보인다. 이는 트래핑 세트에 의한 오류보다는 잡음이 오류성능에 미치는 영향이 더 크기 때문인 것으로 분석된다. 하지만 E_b/N_0 가 증가됨에 따라 트래핑 세트 외부의 비트오류는 반복복호에 의하여 정정되지만 트래핑 세트 내부의 오류는 정정되지 않는다. 이에 따라 원래 알고리즘이 적용된 저밀도 패리티 검사부호의 비트오율은 부호어의 길이가 512비트인 경우 약 4.0dB부터, 그리고 1024비트인 경우에는 약 4.2dB부터 오류마루 현상을 보인다. 이에 비하여 SCMG_{multi} 기법 및 수정 알고리즘이 적용된 부호는 신호 대 잡음비가 증가됨에 따라 지속적으로 성능향상이 이루어지는 것을 확인할 수 있다. 그 결과 부호어의 길이가 512인 경우 약 5.1dB에서, 그리고 1024인 경우는 약 4.8dB에서 제안된 알고리즘의 비트오율은 원래의 방법에 비하여 거의 1/10 수준으로 감소함을 알 수 있다. 이러한 비트오율의 경향은 SCMG_{multi} 기법에서도 그대로 나타난다. 그리고 성능차이는 미미하지만 제안된 수정 합-곱 알고리즘은 SCMG_{multi} 기법에 비하여 약간 우수한 성능을 가짐을 확인할 수 있다. 따라서 SCMG_{multi} 기법과 수정 합-곱 알고

리즘은 저밀도 패리티 검사부호의 반복복호에서 발생하는 트래핑 세트 구조를 효과적으로 해제함으로써 오류마루 현상을 현저히 완화시키거나 제거할 수 있는 것으로 판단된다. 하지만 본 논문에서 제안된 알고리즘은 트래핑 세트의 크기에 대한 문턱값을 사용한다. 따라서 부호어 길이와 부호율에 따라 문턱값을 변화시킬 필요가 없다. 반면에 기존 SCMG_{multi} 기법은 문턱값으로 변수노드의 개수를 사용하기 때문에 부호어 길이와 부호율에 따라 최적의 성능을 보이는 문턱값을 찾기 위한 추가적인 과정이 필요하다.

5.2 구현 복잡도 분석

기존 SCMG_{multi} 기법과 제안된 알고리즘 역시 1단계 복호과정은 원래 합-곱 알고리즘과 동일하므로 반복복호에 필요한 연산량도 동일하다. 그리고 2단계 복호과정의 반복복호는 두가지 모두 최대 30회로 제한되므로 이에 해당되는 연산량이 추가될 수 있다. 하지만 잘 알려져 있듯이 합-곱 알고리즘에서 요구되는 연산복잡도가 크지 않을 뿐만 아니라 그림 5에서도 확인할 수 있듯이 2단계 복호과정에서 실제로 수행되는 반복복호 횟수가 설정된 최대 횟수에 비하여 매우 적으므로 수정 합-곱 알고리즘에서 요구되는 연산량의 증가는 미미할 것으로 판단된다.

원래 알고리즘에 비하여 SCMG_{multi} 기법과 수정 합-곱 알고리즘을 구현하기 위하여 추가로 요구되는 메모리 크기를 부호어의 길이에 따라 계산하여 표 1에 나타내었다.

SCMG_{multi} 기법의 경우 모든 변수노드의 경관정천이 회수를 저장한 다음 선택된 N 개의 변수노드의 초기 확률값의 크기를 비교한다. 따라서 천이 결과에 따라 선택된 변수노드의 위치를 기록하기 위하여 N 개의 메모리가 필요하다. 또한 모든 변수노드의 초기 확률값도 저장되어야 하므로 SCMG_{multi} 기법을 구현하기 위해서는 부호어 길이의 2배에 해

표 1. SCMG_{multi} 기법과 수정 합-곱 알고리즘에서 추가로 필요한 메모리 크기 [바이트]
 Table 1. The additional memory size in bytes for implementing the SCMG_{multi} method and the modified sum-product algorithm

Method	SCMG _{multi}	Proposed
Codeword length		
512 [bits]	1089	612
1024 [bits]	2176	1124

당하는 메모리와 변수노드의 위치 저장을 위한 N 개의 메모리가 사용된다. 그리고 실험결과 SCMG_{multi} 기법은 부호어 길이가 길어지고 부호율이 줄어들수록 문턱값의 크기인 N 이 커지는 것으로 나타났다.

제안된 알고리즘은 1단계에서 반복복호가 수행될 때마다 실패 검사노드의 위치와 천이된 변수노드의 위치를 저장한다. 그러나 트래핑 세트의 크기는 [6]에서처럼 최대 수십개의 변수노드와 그 보다 적은 검사노드로 이루어져있다. 그리고 2단계 복호과정에서 검사영역의 결정 기준으로 실패 검사노드의 수를 5개로 정하였다. 따라서 본 논문에서는 매 반복 복호 마다 실패 검사노드의 수가 5개 이하일 때 그 위치를 중복되지 않게 저장하였으며 변수노드 역시 실패 검사노드의 수가 5개 이하일 때에 천이되는 변수노드들만 저장하였다. 실패 검사노드 집합과 천이 변수노드 집합은 1~30개의 크기를 가지지만 본 논문에서는 이들을 충분히 저장하기 위하여 각각 50개의 메모리를 설정하였다. 그리고 1개의 메모리를 1바이트로 설정하였기 때문에 1단계 복호과정에 필요한 메모리는 100바이트가 된다. 2단계 복호과정에서는 초기화를 위하여 부호어의 길이와 동일한 메모리가 요구된다. 따라서 제안된 알고리즘은 부호어 길이와 동일한 크기의 메모리와 추가로 100 바이트가 요구된다. 그러므로 부호어 길이가 512비트인 경우 SCMG_{multi} 기법은 문턱값이 $N_{512}=65$ 이므로 $512 \times 2 + 65 = 1089$ 바이트의 메모리가 필요하다. 이에 비하여 제안된 알고리즘은 $512 + 100 = 612$ 바이트 만으로 구현할 수 있다. 그리고 부호의 길이가 1024비트인 경우에도 같은 방식으로 계산하면 SCMG_{multi} 기법의 문턱값 $N_{1024} = 128$ 이므로 총 2176 바이트가, 제안된 알고리즘은 1124바이트가 필요하다. 그러므로 제안된 알고리즘은 기존 알고리즘 보다 부호어 길이가 512비트에서 구현시에는 약 43%, 1024비트에서는 약 48%의 메모리 절감효과를 가진다.

VI. 결 론

본 논문에서는 저밀도 패리티 검사부호에서 오류 마루 현상이 발생하는 비트오율 영역을 현저히 감소시키거나 이를 제거할 수 있는 수정 합-곱 알고리즘을 제안하였다. 여기서는 오류마루 현상의 주요 원인인 트래핑 세트를 탐색하고 이에 갇힌 비트오류를 정정하기 위하여 실패 검사노드의 집합과 비트 확률의 경관정 천이패턴을 이용하였다. 기존의

합-곱 알고리즘과는 달리 제안된 알고리즘은 2단계의 복호과정 중에서 1단계에서는 반복복호에 실패한 원인이 트래핑 세트에 의한 것인지를 판단한다. 그리고 2단계에서는 실패 검사노드와 변수노드들이 가지는 오류정보의 경관정 천이패턴을 이용하여 트래핑 세트에 포함된 변수노드를 파악하여 이들의 정보를 반전시킨 다음 합-곱 알고리즘을 수행한다. 가산성 백색 가우시안 잡음 환경에서 모의실험을 수행한 결과 제안된 복호알고리즘에서는 오류마루 현상이 거의 나타나지 않은 것을 확인하였다. 그리고 최적의 문턱값을 이용한 SCMG_{multi} 기법과 비교한 결과 제안된 알고리즘은 부호어의 길이에 따라 거의 동일하거나 미미하지만 조금 더 우수한 오류 성능을 가지는 것으로 나타났다. 이로 인하여 수정 합-곱 알고리즘은 신호 대 잡음비의 증가에 따라 원래의 알고리즘을 통한 복호방법에 비하여 비트오율을 현저히 감소시킨다. 이와 함께 제안된 알고리즘은 반복복호 과정에서 발생하는 정보만을 이용하여 트래핑 세트에 갇힌 변수노드들을 검출하므로 트래핑 세트에 대한 사전정보가 필요없다. 또한 구현에 필요한 추가적인 메모리의 수 측면에서 제안된 수정 알고리즘은 기존 SCMG_{multi} 기법에 비하여 약 43% 이상 감소시킬 수 있는 장점이 있음을 확인하였다.

참 고 문 헌

- [1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, Vol.8, No.1, pp.21-28, Jan., 1962.
- [2] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, Vol.32, No.18, pp.1645-1646, Aug., 1996.
- [3] ETSI TS 102 377 V1.1.1, *Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines*, ETSI, Feb., 2005.
- [4] J. Lin, Z. Wang, L. Li, J. Sha, and M. Gao, "Efficient shuffle network architecture and application for WiMAX LDPC decoders," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, Vol.56, No.3, pp.215-219, Mar., 2009.
- [5] C. Di, D. Proietti, E. Telatar, T. Richardson, and R. Urbanke, "Finite length analysis of low-density parity-check codes on the binary

erasure channel," *IEEE Trans. Inform. Theory*, Vol.48, No.6, pp.1570-1579, June, 2002.

[6] T. Richardson, "Error floors of LDPC codes," *Proc. 41st Allerton Commun. Contr. Comput. Monticello, IL*, pp.1426-1435, Oct., 2003.

[7] C. A. Cole, S. G. Wilson, E. K. Hall, and T. R. Giallorenzi, "Analysis and design of moderate length regular LDPC codes with low error floors," *Proc. IEEE Inform. Sci. Syst. 2006*, Charlottesville, VA, pp.823-828, Mar., 2006.

[8] M. Ivkovic, S. K. Chilappagari, and B. Vasic, "Eliminating trapping sets in low-density parity-check codes by using Tanner graph covers," *IEEE Trans. Inform. Theory*, Vol.54, No.8, pp.3763-3768, Aug., 2008.

[9] X. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graph," *IEEE Trans. Inform. Theory*, Vol.51, No.1, pp.336-398, Jan., 2005.

[10] Y. Zhang and W. Ryan, "Toward low LDPC-code floors: A case study," *IEEE Trans. Commun.*, Vol.57, No.6, pp.1566-1573, June, 2009.

[11] E. Cavus and B. Daneshrad, "A performance improvement and error floor avoidance technique for belief propagation decoding of LDPC codes," *Proc. IEEE PIMRC 2005*, Berlin, Germany, Vol.4, pp.2386-2390, Sep., 2005.

[12] Y. Han and W. Ryan, "Low-floor decoders for LDPC codes," *IEEE Trans. Commun.*, Vol.57, No.6, pp.1663-1673, June, 2009.

[13] H. S. Park, *LDPC Decoding Schemes with Post-Processing for Lowering Error Floors*, M. S. Thesis, Seoul National University, Feb., 2009.

[14] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, Vol.27, No.5, pp.533-547, Sep., 1981.

유 석 근 (Seog Kun Yu)

정회원



2004년 2월 경북대학교 전자전기공학부 학사
 2007년 2월 경북대학교 전자전기공학부 석사
 2007년 3월~현재 경북대학교 전자전기컴퓨터 학부 박사과정

<관심분야> 디지털통신, 디지털신호처리, 오류정정부호

강 석 근 (Seog Geun Kang)

정회원



1988년 2월 경북대학교 전자공학과 학사
 1993년 2월 경북대학교 전자공학과 석사
 1999년 8월 경북대학교 전자공학과 박사
 1993년~1994년 국방과학연구소 연구원

2000년~2003년 National University of Singapore, Research Fellow

2003년~현재 경상대학교 전기전자공학부 부교수
 2003년~현재 경상대학교 공학연구원 책임연구원

<관심분야> 디지털통신, 무선통신, 오류정정부호, 디지털신호처리, 통계적신호처리

주 언 경 (Eon Kyeong Joo)

종신회원



1976년 서울대학교 공과대학 전자공학과 학사
 1984년 Ohio State University 전기공학과 석사
 1987년 Ohio State University 전기공학과 박사
 1976년~1979년 해군 통신전자 기술장교

1979년~1982년 한국과학기술원 연구원

1987년~현재 경북대학교 IT대학 전자공학부 교수

<관심분야> 디지털통신시스템, 정보이론 및 부호화, 디지털신호처리 등