

길쌈부호를 수평부호로 가지는 곱부호의 복호를 위한 유한 연판정 데이터 결합

양 필 웅*, 박 호 성*, 홍 석 범*, 전 보 환*, 노 종 선*, 신 동 준**

Finite Soft Decision Data Combining for Decoding of Product Codes With Convolutional Codes as Horizontal Codes

Pilwoong Yang*, Hosung Park*, Seokbeom Hong*, Bohwan Jun*, Jong-Seon No*, Dong-Joon Shin**

요 약

본 논문에서는 곱부호를 위한 연판정 데이터의 결합을 이용한 복호 기법을 유한 비트 연판정만을 적용하여 사용할 수 있도록 새로운 연판정 데이터의 결합 법칙을 제시한다. 기존의 무한 연판정을 이용한 복호 기법은 복잡한 tanh 연산에 기반을 두고 있으므로 높은 연산 복잡도와 함께 하드웨어에 적용하기 어려운 단점이 있다. 따라서 기존 복호 기법에서 이용되는 연산들을 분석하여 작성한 간단한 연산 테이블을 이용하여 복호하는 방법을 제시한다. 또한 연판정 데이터를 적용하기 용이한 길쌈부호를 곱부호의 수평부호로 이용하였을 때 연판정 데이터의 결합 방식을 찾는 방법을 제시하고 모의 실험을 통해 성능을 검증한다. 제안된 알고리즘은 4비트 유한 연판정을 적용한 길쌈부호를 곱부호의 수평부호로 이용하였을 때 무한 연판정을 적용한 경우의 성능에 근접함을 보였다.

Key Words : channel coding, convolutional codes, product codes, soft decision data, Viterbi algorithm

ABSTRACT

In this paper, we propose feasible combining rules for a decoding scheme of product codes to apply finite soft decision. Since the decoding scheme of product codes are based on complex tanh calculation with infinite soft decision, it requires high decoding complexity and is hard to practically implement. Thus, simple methods to construct look-up tables for finite soft decision are derived by analyzing the operations of the scheme. Moreover, we focus on using convolutional codes, which is popular for easy application of finite soft decision, as the horizontal codes of product codes so that the proposed decoding scheme can be properly implemented. Numerical results show that the performance of the product codes with convolutional codes using 4-bit soft decision approaches to that of same codes using infinite soft decision.

I. 서 론

무선 통신 시스템에서 전송되는 신호는 다른 신호와의 간섭, 페이딩 및 전파 방해 등의 효과로 인해 왜곡 된다. 이러한 신호 왜곡 현상을 효과적으로

해결하기 위해 순방향 오류 정정(forward error correction, FEC) 방식이 사용된다. 전진 오류 정정 방식은 정보 시퀀스에 잉여 비트를 추가하여 송신함으로써 수신측에서 이를 이용해 오류를 검출하고 정정하는 방식을 말한다. 길쌈부호(convolutional

※ 본 연구는 방송통신위원회의 차세대 통신 네트워크 원천기술 개발 사업 (KCA-2012-08-911-04-003)과 2012년도정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2012-0000186)

◆ 주저자 : 서울대학교 전기·컴퓨터 공학부 및 뉴미디어통신공동연구소, yangpw@ccl.snu.ac.kr, 정희원

* 서울대학교 전기·컴퓨터 공학부 및 뉴미디어통신공동연구소, lovepark98@ccl.snu.ac.kr, 정희원, fousbyus@ccl.snu.ac.kr, netjic@ccl.snu.ac.kr, 비희원, jsno@snu.ac.kr, 종신회원

** 한양대학교 전자통신컴퓨터공학부, djshin@hanyang.ac.kr, 종신회원

논문번호 : KICS2012-04-188, 접수일자 : 2012년 4월 14일, 최종논문접수일자 : 2012년 6월 13일

codes)는 이러한 방식을 위해 사용되는 오류 정정 부호로서 우수한 오류 정정 능력과 더불어 연판정 (soft decision) 복호를 적용하기 용이하므로 비터비 복호기와 함께 위성·이동 통신에 널리 이용되어진다. 곱부호(product codes)는 여러 개의 부호어가 일렬로 연결된 부호로서 두 개 이상의 짧은 길이의 블록 부호들을 연결하여 긴 블록 부호를 구성 하는데 이용된다. 작은 최소 해밍 거리(minimum Hamming distance)를 갖는 짧은 부호들을 결합하여 큰 최소 해밍 거리를 갖는 긴 부호를 생성할 수 있으므로 곱부호 또한 다양한 분야에 널리 응용되어 왔다^[1,2].

이러한 곱부호를 위한 새로운 복호 기법이 최근에 제안 되었다^[3]. 이는 곱부호의 부호어 행렬 (codeword matrix)내의 수평부호(horizontal codes)를 기준으로 오류가 발생한 수평부호어들의 연판정 데이터를 결합하여 결과적으로 하나의 수평부호어를 기준으로 두 개의 독립적으로 수신한 연판정 데이터를 얻어냄으로써 성능 이득을 얻는 기법이다. 이 기법의 성능 및 복잡도는 하나의 부호어 행렬 내에서 발생한 수평부호어 오류 개수에 영향을 받으므로 특히 높은 신호대 잡음비(signal-to-noise ratio, SNR)에서 적은 복잡도 증가만으로 큰 성능 이득을 얻을 수 있다. 이 때 수평부호로 이용되는 부호는 선형이고 오류 검출 능력이 있는 어떠한 부호도 사용 가능하며, 수직부호(vertical codes)로 사용되는 부호들은 SPC 부호나 해밍 부호와 같은 간단한 부호가 사용될 수 있다.

앞서 제안된 복호 기법은 적은 복잡도 증가로 성능 이득을 나타내는 반면, 무한한 연판정 데이터를 이용한 로그우도비율(log-likelihood ratio, LLR) 덧셈과 tanh 연산에 기반을 두고 있으므로 실제 하드웨어에서 구현되기에 어려움이 있다. 따라서 3비트, 4비트 등의 유한 연판정만을 허용하는 환경에서 이 복호 기법을 간단하게 적용할 수 있는 방법에 대한 고찰이 필요하다.

본 논문에서는 이 복호 기법에 이용되는 두 연산을 분석하여 유한 비트 연판정 데이터만을 이용할 때 복잡한 계산을 거치지 않고 간단한 연산 테이블을 적용하여 결합 법칙을 나타낼 수 있는 방법을 제시한다. 간단한 대수 부호들이 수직부호로 이용될 수 있으나 여기서는 가장 명료하게 기법을 적용할 수 있는 단일 패리티 체크(single parity-check, SPC)부호를 기준으로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 최근

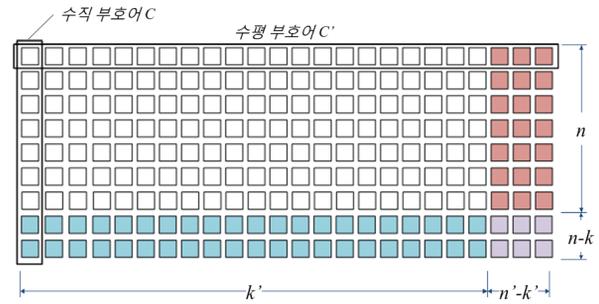


그림 1. 곱부호의 부호어 행렬 구조
Fig 1. Structure of a codeword matrix of a product code

에 소개된 곱부호의 복호 기법에 대해 설명한다. 3장에서는 앞서 소개한 복호 기법에 유한 비트 연판정을 적용하고, 이를 토대로 4장에서는 유한 비트 연판정을 이용한 길쌈부호를 수평부호로 사용할 때의 복호 기법을 유도한다. 5장에서 모의 실험을 통해 성능을 확인하고 마지막으로 6장에서 결론을 맺는다.

II. 곱부호를 위한 복호 기법

이 장에서는 [3]에서 소개된 곱부호를 위한 복호 기법에 대해 소개한다. 이 복호 기법은 동일한 수평 부호어를 위한 두 개의 독립적으로 수신된 연판정 데이터를 결합하는 것을 기반으로 한다. 수신한 곱부호의 부호어 행렬 내에 오류가 발생한 수평부호어만을 결합하므로 이용되는 수평부호는 선형이고 오류 검출 능력이 있어야 한다.

부호의 길이를 $n(n')$, 정보 비트수를 $k(k')$, 최소 해밍 거리를 $d_{\min}(d_{\min}')$ 이라 할 때, 각각 (n, k, d_{\min}) , (n', k', d_{\min}') 의 파라미터를 갖는 두 종류의 블록 부호 C, C' 을 고려한다. 이러한 부호 C 와 C' 을 각각 수직부호, 수평부호로 가지는 곱부호의 구조는 그림 1과 같다.

곱부호의 부호어 행렬이 그림 1과 같이 $n \times n'$ 행렬이라 할 때, 이 부호는 n 개의 수평부호어가 쌓여 있는 행렬이 된다. $\mathbf{c}_i = (c_{i1}, c_{i2}, \dots, c_{in'})$, $1 \leq i \leq n$ 가 i 번 째 수평부호어를, $\mathbf{r}_i = (r_{i1}, r_{i2}, \dots, r_{in'})$, $1 \leq i \leq n$ 가 \mathbf{c}_i 를 전송했을 때 채널로부터 수신한 값을 연판정한 데이터를 의미한다 하자. 연판정 데이터 \mathbf{r}_i 는 \mathbf{c}_i 의 수신값으로부터 얻어진 LLR값으로 이루어진 벡터이다. 곱부호의 수직부호로 SPC 부호가 이용되면 수직부호의 패리티 체크 행렬 H 는 다음과 같이 나타내진다.

$$H = [1 \ 1 \ 1 \ 1 \ \dots \ 1] \quad (1)$$

따라서 패리티 체크 행렬의 성질에 의해 다음 식이 성립한다.

$$\sum_{i=1}^n c_i = 0 \quad (2)$$

즉, 하나의 수평부호어 c_i 는 나머지 다른 수평부호어들의 이진 합(exclusive OR, XOR)과 동일하다.

c_i 의 연관성 데이터 r_i 를 결합하기 위해 아래와 같은 식을 이용한다. 두 개의 부호어가 합해진 결과값의 LLR값은 각각의 부호어의 LLR값을 이용하여 다음과 같이 계산되어 진다^[4].

$$r_{ij} \boxplus r_{i'j} := 2 \tanh^{-1} \left(\tanh \left(\frac{r_{ij}}{2} \right) \tanh \left(\frac{r_{i'j}}{2} \right) \right) \quad (3)$$

임의의 한 수평부호어를 \bar{c}_i 라 하고 이의 LLR값들로 이루어진 벡터를 \bar{r}_i 라 하자. 이 때 부호어 \bar{c}_i 가 정상적으로 복호된 부호어라면 이의 각 비트의 LLR값 \bar{r}_{ij} 은 \bar{c}_{ij} 가 1일 때 $-\infty$, 0일 때 ∞ 가 된다. 이처럼 결합하는 두 개의 부호어 중 하나의 부호어를 이미 알고 있을 때 (3)의 LLR 결합은 아래와 같이 간단히 부호 변화로 나타낼 수 있다.

$$\begin{cases} r_{ij} \boxplus \bar{r}_{ij} = r_{ij} & , \quad \text{if } \bar{r}_{ij} = \infty \\ r_{ij} \boxplus \bar{r}_{ij} = -r_{ij} & , \quad \text{if } \bar{r}_{ij} = -\infty \end{cases} \quad (4)$$

위와 같은 특성들을 이용하여 제시된 복호 방법은 곱부호의 부호어 행렬 내의 수평부호어 오류 개수에 따른 정정 기법이다. 총 n 개의 수평부호어 중 오류가 발생한 수평부호어의 개수가 e 일 때, 세부적인 복호 과정은 아래와 같다.

2.1. $e=1$ 인 경우:

부호어 행렬 내의 수평부호어 오류 개수가 1개인 경우, 수직부호어로 사용된 SPC 부호의 성질에 따라 이 수평부호어는 부호어 행렬 내의 다른 모든 수평부호어들의 합과 같다. 즉, 오류가 발생한 수평부호어를 c_1' 이라 하면 아래와 같은 식이 성립한다.

$$c_1' = c_2 \oplus c_3 \oplus \dots \oplus c_n \quad (5)$$

c_1' 을 제외한 모든 수평부호어들은 정상적으로

복호되었으므로 이들의 이진 합으로 c_1' 을 항상 복호할 수 있다. 따라서 $e=1$ 인 경우 언제나 복호가 가능하다.

2.2. $e=2$ 인 경우:

부호어 행렬 내의 수평부호어 오류 개수가 2개인 경우, 오류가 발생한 수평부호어들을 각각 c_1' , c_2' 이라 하면 SPC 부호의 성질에 따라 아래의 식이 성립한다.

$$c_1' = c_2' \oplus c_3 \oplus \dots \oplus c_n \quad (6)$$

부호어 행렬내에 두 개 이상의 수평부호어 오류가 존재하므로 $e=1$ 인 경우와 같이 경관성 데이터만을 이용한 복호는 사용할 수 없다. 따라서 (6)에 대한 연관성 데이터 계산을 고려한다. c_1' 의 LLR벡터는 r_1' 으로 주어지고, (6)의 우변의 LLR벡터는 아래와 같이 주어진다.

$$r_2' \boxplus r_3 \boxplus \dots \boxplus r_n \quad (7)$$

위 식에서 r_2' 을 제외한 r_3, r_4, \dots, r_n 은 c_3, c_4, \dots, c_n 이 모두 정상적으로 복호된 수평부호어들이므로, 벡터 내의 모든 성분이 $-\infty$ 혹은 ∞ 값을 가진다. 따라서 (7)의 결과값은 (4)에 언급하였듯 r_2' 벡터내의 모든 성분들에 간단한 부호 변화만 일어난 결과, 즉 모든 성분이 r_{2j}' 혹은 $-r_{2j}'$ 값을 가지는 벡터가 된다. 이렇게 얻어진 결과값과 c_1' 의 연관성 데이터 r_1' 은 (6)의 좌우변이 동일하므로 동일한 부호어에 대한 연관성 데이터라 생각할 수 있다. 결과적으로 (6)의 좌변 c_1' 을 기준으로 볼 때, LLR벡터 r_1' 뿐 아니라 (7)의 결과도 c_1' 의 연관성 데이터로 간주할 수 있으므로 이 두 LLR벡터를 단순히 더하여 얻어진 연관성 데이터를 이용해서 c_1' 을 재 복호함으로써 높은 성능 이득을 얻을 수 있다. 재 복호에 성공한다면 곱부호의 부호어 행렬에 남아있는 수평부호어 오류 개수는 1개 이므로 $e=1$ 인 경우로 돌아가 항상 복호가 가능해지고 이로써 부호어 행렬은 모두 복호 된다.

2.3. $e \geq 3$ 인 경우:

3개 이상의 수평부호어 오류를 부호어 행렬이 가지고 있는 경우 오류를 포함한 수평부호어들을 각각 c_1', c_2', \dots, c_h' 이라 한다. ($h \leq n$) 마찬가지로 아래

의 식이 성립한다.

$$c_1' = c_2' \oplus \dots \oplus c_h' \oplus c_{h+1} \oplus c_{h+2} \oplus \dots \oplus c_n \quad (8)$$

$e=2$ 인 경우와 마찬가지로 c_1' 을 위한 두 연판정 데이터는 다음과 같다.

$$\begin{cases} r_1' \\ r_2' \boxplus \dots \boxplus r_h' \boxplus r_{h+1} \boxplus r_{h+2} \boxplus \dots \boxplus r_n \end{cases} \quad (9)$$

$c_{h+1}, c_{h+2}, \dots, c_n$ 은 정상 복호되었으므로 $r_{h+1} \boxplus r_{h+2} \boxplus \dots \boxplus r_n$ 벡터 내의 모든 성분은 $-\infty$ 혹은 ∞ 이다. 따라서 c_1' 을 위한 두 번째 연판정 데이터는 채널에서 얻은 r_2', r_3', \dots, r_h' 을 (3)에 의해 계산한 $r_2' \boxplus r_3' \boxplus \dots \boxplus r_h'$ 을 (4)에 따라 부호 변화함으로써 얻어낼 수 있다. 이 두 LLR 벡터를 더한 결과를 이용하여 c_1' 을 재 복호함으로써 성능 이득을 얻을 수 있고, 성공 시 부호어 행렬에 남은 수평부호어 오류 개수가 하나씩 감소하므로 계속해서 재 복호를 시도할 수 있다.

III. 유한 비트 연판정을 위한 결합 법칙

이 장에서는 II장에서 설명된 기존의 복호 방식을 유한 비트 연판정만이 허용된 환경에 적용하는 방법에 대해 설명한다. 앞선 복호 기법에 사용되는 연산은 하나의 수평부호어를 나타내는 두 개의 연판정 데이터를 합치는 LLR 덧셈 계산과, 부호어 행렬 내의 오류 개수가 3개 이상일 때 사용되는 (3)의 LLR tanh 계산으로 두 가지이다. 따라서 두 가지 계산이 유한 비트 연판정에서 어떻게 표현되는지 도출함으로써 결합 법칙을 구성할 수 있다. 무한 연판정에서 연판정 데이터를 LLR값을 이용한 것과 달리 유한 비트 연판정에서는 하드웨어에서 표현할 수 있는 한계가 존재하므로 모든 실수범위의 수신값을 해당 비트에 맞게 양자화(quantization)하고 각 양자화 레벨의 대푯값으로부터 LLR을 얻는 것으로 가정하였다.

3.1. LLR 합 연산

II장에서 언급했듯이, 부호어 행렬 내의 수평부호어 오류 개수 e 가 2 이상인 경우 연판정 데이터 LLR값의 합 연산이 필요하다.

LLR 합 연산은 근본적으로 해당하는 비트의 신뢰도를 더하는 과정이므로 유한 비트 연판정에서

사용하는 각 양자화 레벨의 대푯값들이 0을 기준으로 대칭이라면 대푯값을 그대로 더하는 것이 결과적으로 신뢰도를 더하는 것이 된다. 따라서 부호어 행렬 내의 수평부호어 오류 개수 e 가 2일 때는 각 비트의 유한 비트 연판정 데이터, 즉 양자화 레벨의 대푯값들을 더한 결과를 복호함으로써 좀 더 높은 신뢰도의 결과를 얻을 수 있다.

하나 e 가 3 이상일 때는 e 가 2인 경우와 달리, 더해야 하는 두 개의 연판정 데이터 중 하나의 신뢰도가 더 저하된 상황이므로 두 연판정 데이터를 그대로 더해서는 안 된다. 즉 II.3절에서 c_1' 에 해당하는 (9)에 나타난 두 연판정 데이터를 더할 때 LLR값은 그대로 더하는 것이 최대비율 결합(maximal ratio combining, MRC)을 만족하지만, 연판정 데이터를 각 양자화 레벨의 대푯값으로 사용하는 유한 비트 연판정 상황에서는 덧셈을 아래와 같이 보정해 줌으로써 MRC를 만족할 수 있다.

$$s_{c_{ij}} + \frac{\sigma^2}{\sigma'^2} s_{(c_2 \boxplus c_3 \boxplus \dots \boxplus c_{ij})} \quad (10)$$

이 때, c_1, c_2, \dots, c_h 는 오류가 발생한 수평 부호어들이고, s_i 는 i 비트의 연판정 데이터인 각 양자화 레벨의 대푯값, σ 는 채널 표준 편차를 나타낸다. σ' 은 \boxplus 연산에 의해 변경된 표준 편차를 의미한다. 수평부호어들 사이의 연산은 각 비트마다 독립적으로 이루어지므로 j 는 수평부호어의 길이 내부의 한 지점이다. tanh 연산을 거친 결과인 $s_{(c_2 \boxplus c_3 \boxplus \dots \boxplus c_{ij})}$ 는 다음 절에서 계산된다.

3.2. LLR tanh 연산

부호어 행렬에 3개 이상의 수평부호어 오류가 발생했을 때는 LLR tanh 연산이 필요하다. 따라서 두 개의 연판정 데이터, 즉 두 개의 대푯값들의 tanh 결과가 어느 양자화 레벨에 존재하는지 판정을 내리는 것이 목적이다.

먼저, (3)의 tanh 연산이 어떤 성질을 가지는지 확인한다. 결합할 비트를 각각 A, B 라 하고 이들의 연판정 데이터는 각각 s_A, s_B , 결합 결과값을 $s_{A \boxplus B}$ 라 한다.

i) $s_{A \boxplus B}$ 의 부호는 s_A 와 s_B 의 부호가 같을 때 양수, 다를 때 음수가 된다.

tanh 연산의 의미는 두 개의 부호어를 이진 합

연산한 결과의 신뢰도를 나타낸다. 이진 합 연산은 두 비트의 부호가 같을 때 양수, 다를 때 음수가 되므로 연관성 데이터의 부호도 마찬가지로 설정된다.

ii) $|s_{A\oplus B}| \leq |s_A|, |s_B|$ 이다.

연관성 데이터의 절대값은 신뢰도의 크기 나타낸다. 즉, 신뢰도가 높은 데이터와 신뢰도가 낮은 데이터를 결합했을 때 결과의 신뢰도는 둘 중 신뢰도가 낮은 데이터 보다 높을 수 없다. 100% 신뢰도를 가진 데이터와 낮은 신뢰도를 가진 데이터를 결합했을 때 결과의 신뢰도는 낮은 신뢰도와 동일하다. 이는 (4)에서 보인 결과와 같다.

iii) $0 \leq s_A \leq s_B$ 일 때

$s_A - s_{A\oplus A} \geq s_A - s_{A\oplus B}$ 이 성립한다.

ii)에서 연산 결과는 항상 신뢰도가 감소함을 언급하였으므로 이제 신뢰도의 감소크기를 고려한다. 신뢰도 s_A 가 있을 때, $s_{A\oplus B}$ 의 신뢰도의 감소크기는 결합 되는 데이터의 신뢰도 s_B 가 높을수록 적다. s_B 가 0이면 $s_{A\oplus B}$ 는 0, s_B 가 ∞ 이면 $s_{A\oplus B}$ 는 s_A 가 된다. 따라서 위의 식이 성립한다.

iv) $0 \leq s_A \leq s_B$ 일 때

$s_B - s_{B\oplus B} \geq s_A - s_{A\oplus A}$ 이 성립한다.

iii)에서 하나의 신뢰도가 고정 되어 있을 때 결과값의 신뢰도의 감소크기는 결합되는 데이터의 신뢰도가 높을수록 적다는 것을 보였다. 이번에는 신뢰도가 높은 데이터 둘을 결합한 결과와 신뢰도가 낮은 데이터 둘을 결합한 결과의 신뢰도 감소크기를 비교한다. s_i 의 우도비율(likelihood ratio, LR)을 L_i 라 하면, (3)에서 LR값은 아래와 같이 계산된다.

$$L_A \oplus L_B = \frac{L_A L_B + 1}{L_A + L_B} \quad (11)$$

$r_A = \log(L_A)$ 이므로, $r_A - r_{A\oplus A}$ 는 다음과 같다.

$$\begin{aligned} & \log(L_A) - \log(L_A \oplus L_A) \\ &= \log(L_A) - \log\left(\frac{L_A L_A + 1}{L_A + L_A}\right) \quad (12) \\ &= \log\left(2 - \frac{2}{L_A^2 + 1}\right) \end{aligned}$$

$s_A \leq s_B$ 이면 $r_A \leq r_B$ 이고, (12)에서 신뢰도가 큰 두 데이터를 결합한 결과는 신뢰도가 작은

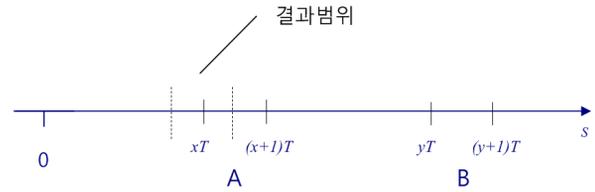


그림 2. 양자화 레벨 A와 레벨 B를 田연산한 결과범위
Fig 2. Result range of combining Level A and B

두 데이터를 결합한 결과보다 신뢰도 감소크기가 더 크다는 것을 확인할 수 있으므로 위의 식이 만족한다. 여기서 중요한 점은 신뢰도 감소크기가 r_A 가 커질수록 점점 커지다가 일정해진다는 점이다. 신뢰도 감소크기의 최댓값은 (12)에 의하면 $\log 2$ 가 되며, 이는 LLR값이므로 채널 수신값으로 변환하면 채널 표준 편차가 σ 인 가산 백색 가우시안 잡음(additive white Gaussian noise, AWGN) 채널일 때 $(\sigma^2/2)\log 2$ 가 된다. 이를 이용하여 유한 비트 연관성에 사용되는 비트수가 정해졌을 때 간단한 연산이 가능한 dB범위를 계산할 수 있다.

위의 성질들을 이용하여 s_A 와 s_B 로부터 $s_{A\oplus B}$ 를 계산한다. 무한 연관성과는 다르게 유한 비트 연관성은 n 비트 연관성일 때 2^n 개의 양자화 레벨이 존재하므로, 두 개의 양자화 레벨을 田연산했을 때 결과값이 어느 양자화 레벨로 결정되는지 판단해야 한다. 그림 2는 양자화 레벨 A와 B가 결합되었을 때의 결과값이 존재할 수 있는 범위를 나타낸다. 결과범위의 좌측 경계는 양자화 레벨 A와 레벨 B의 좌측경계를 결합한 결과로, 우측 경계는 각 양자화 레벨의 우측 경계를 결합한 결과로 결정된다. 또한 iv)의 성질에 의해 결과범위의 크기는 한 양자화 레벨의 크기를 T 라 했을 때 T 를 넘을 수 없으므로 어떠한 결과범위도 최대 두 양자화 레벨에만 걸쳐 존재할 수 있다. 따라서 이러한 결과범위가 두 양자화 레벨에 걸쳐 존재한다면 좀 더 높은 확률을 가진, 즉 결과범위의 중간값을 포함하고 있는 양자화 레벨이 결정레벨이 될 것이다.

여기서 고려할 점은 양자화 레벨 A와 레벨 B의 田연산이 항상 낮은 신뢰도를 갖는 양자화 레벨 A로 결정되는 경우이다. 위의 iv)에서 언급하였듯 신뢰도 감소크기의 최댓값은 채널 표준 편차가 σ 인 AWGN 채널일 때 $(\sigma^2/2)\log 2$ 이다. 만약 이 신뢰도 감소크기의 최대치가 한 양자화 레벨의 크기 T 의 절반을 넘지 않는다면, i) ~ iv)의 성질에 따라 어떠한 두 양자화 레벨 A와 B를 田연산한 결과

는 항상 낮은 신뢰도를 갖는 양자화 레벨로 결정될 것이다. 신뢰도 감소크기의 최댓값은 σ 에 따라 달라지므로, 계산에 의해 이러한 상황이 발생하는 dB 구간을 찾아낼 수 있다.

이외의 경우 아래와 같은 과정을 거쳐 연산 테이블을 작성할 수 있다.

i) 타깃이 되는 dB의 채널 표준 편차 σ 를 계산한다.

ii) 각 양자화 레벨의 대푯값을 설정하고, (일반적으로 중간값) 설정된 대푯값을 LLR로 변환한다. AWGN 채널에서 $r_A = 2s_A/\sigma^2$ 으로 쉽게 변환된다.

iii) 변환된 LLR값들로 가능한 모든 대푯값의 쌍에 대해 (3)의 연산을 수행한다.

iv) iii)의 결과값 들을 $s_A' = (\sigma^2/2)r_A'$ 로 재 변환 하고, 변환한 결과가 위치하는 양자화 레벨을 결정레벨로 설정한다.

이 때 iii)에서 (3)의 연산을 수행한 결과값들은 신뢰도의 크기가 감소하였으므로 변경된 채널 표준 편차 σ' 을 갖지만, iv)에서 재변환시 σ' 이 아닌 σ 를 다시 이용함으로써 두 연판정 데이터를 더하는 과정인 (10)에서 언급한 MRC를 아래와 같이 만족할 수 있다.

$$s_{(c_2 \oplus c_3)} = \frac{\sigma'^2}{2} r_{(c_2 \oplus c_3)} \quad (13)$$

$$\begin{aligned} s_{c_1'} &= s_{c_1} + \frac{\sigma^2}{\sigma'^2} s_{(c_2 \oplus c_3)} \\ &= \frac{\sigma^2}{2} r_{c_1} + \frac{\sigma^2}{\sigma'^2} \frac{\sigma'^2}{2} r_{(c_2 \oplus c_3)} \\ &= \frac{\sigma^2}{2} r_{c_1} + \frac{\sigma^2}{2} r_{(c_2 \oplus c_3)} \end{aligned} \quad (14)$$

결과적으로 계산 후 얻은 연판정 데이터와 원래의 연판정 데이터를 더함으로써 MRC를 만족하는 연산을 수행하였다. 이렇게 얻어진 연판정 데이터를 이용하여 해당 수평부호어를 재 복호함으로써 II장의 복호 기법을 이용할 수 있다.

IV. 길쌈부호를 이용한 곱부호 복호

이 장에서는 III장에 언급한 결합 법칙을 이용하여 3비트, 4비트 연판정을 적용한 길쌈부호를 수평

부호로 하는 곱부호의 복호방법을 알아본다.

4.1. 3비트 연판정 길쌈부호를 수평부호로 하는 곱부호

3비트 연판정을 이용하는 길쌈부호의 경우 일반적으로 이용되는 양자화 레벨의 크기는 $T = 0.6\sigma$ 이다. $2^3 = 8$ 개의 양자화 레벨이 존재하므로 이의 대푯값들을 각각 $-7, -5, -3, -1, 1, 3, 5, 7$ 으로 설정하였고, 이는 각 비트의 연판정 데이터로 사용된다. 부호어 행렬 내에 오류가 발생한 수평부호어의 개수 e 가 1이면 수직부호인 SPC 부호의 성질에 따라 항상 복원 가능하다. e 가 2인 경우 대푯값으로 설정된 연판정 데이터를 더한 값을 복호함으로써 신뢰도를 향상시키는 효과를 얻을 수 있으며, e 가 3 이상인 경우 III장에 제시한 방법을 이용하여 타깃으로 하는 dB의 표를 작성하여 간단히 복호할 수 있다.

가장 먼저 고려할 점은 (3)의 田계산을 해야할 경우, 어떠한 두 양자화 레벨을 田연산 하더라도 항상 낮은 신뢰도를 갖는 양자화 레벨에 놓이게 되는 ‘단순 田연산’ 이 가능한 구간을 찾아내는 것이다. 앞서 언급하였듯 이 구간은 최대 신뢰도 감소크기가 구간의 길이의 절반이하로 수렴하는 dB구간을 찾아내는 것과 동일하다. 즉, 아래와 같이 나타난다.

$$\max\left(\frac{\sigma^2}{2}(r_A - r_{A \oplus A})\right) \leq \frac{T}{2} \quad (15)$$

$r_A - r_{A \oplus A}$ 의 최댓값은 III장에서 \log_2 임을 밝혔으므로 한 양자화 레벨의 크기 $T = 0.6\sigma$ 를 고려하면 σ 의 범위는 아래와 같이 정리된다.

$$\sigma \leq \frac{0.6}{\log_2} \quad (16)$$

따라서 (16)은 $\sigma \leq 0.8656$, $E_b/N_0 \geq 1.25dB$ 일 때 성립한다. 실제로 부호가 적용되는 채널 환경이 이 범위를 벗어나는 경우는 드물기 때문에 표 1과 같은 단순 田연산이 가능하다. 표 1은 $-7 \sim 7$ 을 대푯값으로 갖는 3비트 연판정 데이터를 田연산한 결과를 나타낸다.

결과적으로 3비트 유한 연판정을 이용한 길쌈부호를 수평부호로 이용했을 경우 3개 이상의 수평부호어 오류가 부호어 행렬 내에 발생했을 때 田연산의 결과값은, 부호는 연산할 모든 연판정 데이터의

표 1. 단순 田연산 결과
Table 1. Simple combined results

	-7	-5	-3	-1	1	3	5	7
-7	7	5	3	1	-1	-3	-5	-7
-5	5	5	3	1	-1	-3	-5	-5
-3	3	3	3	1	-1	-3	-3	-3
-1	1	1	1	1	-1	-1	-1	-1
1	-1	-1	-1	-1	1	1	1	1
3	-3	-3	-3	-1	1	3	3	3
5	-5	-5	-3	-1	1	3	5	5
7	-7	-5	-3	-1	1	3	5	7

부호의 곱으로, 절대값은 모든 연판정 데이터 중 절대값이 가장 작은 데이터의 절대값으로 결정되어진다. 이렇게 얻은 결과와 다른 하나의 연판정 데이터를 (14)와 같이 더하여 재 복호함으로써 성능 이득을 얻을 수 있다.

4.2. 4비트 연판정 길쌈부호를 수평부호로 하는 곱부호

4비트의 경우 양자화 레벨의 개수가 3비트 보다 두 배 늘어나므로 $T = 0.3\sigma$ 로 줄어든다. (15)에 의해, 이 경우 표 1에 나온 단순 田연산이 가능한 구간은 $E_b/N_0 \geq 7.27dB$ 이다. 따라서 III장에서 언급한대로 적용하는 채널의 dB에 따라 적합한 田연산 표를 작성할 수 있다. 표 2 ~ 표 5는 0.5 ~ 4.0dB의 범위에서 0.5dB단위마다 얻어낸 연산 테이블의 예시이며, 각 양자화 레벨의 대푯값은 -15 ~ 15로 설정되었다. 연산 결과의 부호는 두 연판정 데이터의 부호가 같으면 양수, 다르면 음수이므로 절대값만 표로 나타내었다.

음영으로 처리된 부분은 표 1의 단순 田연산 결과와 다른 결과값 임을 나타낸다. 채널의 정보를 미리 알고 있다고 가정 할 때 4비트 연판정의 경우 위와 같은 연산 테이블을 이용해 단순 田연산 보다 더욱 정밀한 연산을 수행할 수 있다. 마찬가지로 이렇게 얻어낸 결과와 다른 연판정 데이터를 더한 결과를 이용하여 재 복호를 시도 할 수 있다.

V. 모의 실험 결과

이 장에서는 모의 실험을 통하여 제안된 유한 비트 연판정을 적용했을 때의 성능을 보이고, 무한 연판정을 그대로 사용했을 때의 성능과 비교하여 보인다. 모의 실험에는 3비트·4비트 테일바이팅

표 2. 4비트 연판정 田연산 결과, $E_b/N_0=0.5-1dB$
Table 2. Combined results of 4-bit soft decision at $E_b/N_0=0.5-1dB$

	15	13	11	9	7	5	3	1
15	13	11	11	9	7	5	3	1
13	11	11	9	9	7	5	3	1
11	11	9	9	7	7	5	3	1
9	9	9	7	7	5	5	3	1
7	7	7	7	5	5	3	3	1
5	5	5	5	5	3	3	1	1
3	3	3	3	3	3	1	1	1
1	1	1	1	1	1	1	1	1

표 3. 4비트 연판정 田연산 결과, $E_b/N_0=1.5dB$
Table 3. Combined results of 4-bit soft decision at $E_b/N_0=1.5dB$

	15	13	11	9	7	5	3	1
15	13	11	11	9	7	5	3	1
13	11	11	9	9	7	5	3	1
11	11	9	9	7	7	5	3	1
9	9	9	7	7	5	5	3	1
7	7	7	7	5	5	3	3	1
5	5	5	5	5	3	3	3	1
3	3	3	3	3	3	3	1	1
1	1	1	1	1	1	1	1	1

표 4. 4비트 연판정 田연산 결과, $E_b/N_0=2dB$
Table 4. Combined results of 4-bit soft decision at $E_b/N_0=2dB$

	15	13	11	9	7	5	3	1
15	13	11	11	9	7	5	3	1
13	11	11	9	9	7	5	3	1
11	11	9	9	7	7	5	3	1
9	9	9	7	7	5	5	3	1
7	7	7	7	5	5	5	3	1
5	5	5	5	5	5	3	3	1
3	3	3	3	3	3	3	1	1
1	1	1	1	1	1	1	1	1

(tailbiting) 길쌈부호가 수평부호로, SPC 부호가 수직부호로 적용된 곱부호가 사용되었다. 부호율을 높이기 위해 사용된 테일바이팅 길쌈부호를 복호하기 위해 랩-어라운드 비터비 복호기(wrap-around Viterbi decoder)가 이용되었다³⁾. 수평부호어로 사용된 길쌈부호는 부호율 1/2 길쌈부호에서 12/23

표 5. 4비트 연판정 田연산 결과, $E_b/N_0=2.5\text{-}4\text{dB}$
 Table 5. Combined results of 4-bit soft decision at $E_b/N_0=2.5\text{-}4\text{dB}$

	15	13	11	9	7	5	3	1
15	13	11	11	9	7	5	3	1
13	11	11	11	9	7	5	3	1
11	11	11	9	9	7	5	3	1
9	9	9	9	7	7	5	3	1
7	7	7	7	7	5	5	3	1
5	5	5	5	5	5	3	3	1
3	3	3	3	3	3	3	1	1
1	1	1	1	1	1	1	1	1

양식의 삭제행렬(deleting matrix)을 이용하여 천공(punctured)된 부호율 12/23인 길쌈부호가 사용되었으며, 부호어·메시지 길이는 (552,288), 구속장 길이(constraint length K)는 9가 사용되었다. 수평부호어 복호를 위한 램-어라운드 비터비 복호기의 반복 횟수(iteration)는 두 번으로 설정하였다. 수직부호로 이용된 SPC 부호는 부호어·메시지 길이가 (24,23)로 설정되어 부호율은 23/24이고, 결과적으로 곱부호의 총 부호율은 1/2 이다. 실험환경은 AWGN 채널을 가정하였다.

그림 3은 3비트 유한 연판정을 적용한 길쌈부호를 수평부호로 갖는 곱부호의 성능을 보여준다. 곱부호 # 뒤의 숫자는 복호를 시도하는 수평부호어의 최대 오류 개수를 말한다. 즉, 곱부호 #3은 부호어 행렬 내의 수평부호어 중 3개 이하의 수평부호어에 만 오류가 발생 했을 시 복호를 시도하고 4개 이상의 수평부호어에 오류가 발생하면 복호를 포기한다. 좀 더 많은 수의 수평부호어 오류를 복호 시도 하는 것이 물론 좋은 성능을 보이나, 그만큼 복잡도도 증가하는 결과를 가져온다. 이 때 최대로 고려하는 수평부호어 오류 개수의 증가에 따른 성능 증가폭이 E_b/N_0 가 높아짐에 따라 점점 감소하는 것도 동시에 관찰 된다. 곱부호 #2, #3, #4, #24의 성능이 점점 높은 E_b/N_0 로 갈수록 수렴해 가는 것을 볼 수 있는데, 이는 E_b/N_0 가 높아질수록 곱부호의 부호어 행렬 내의 많은 수의 수평부호어에서 오류가 발생할 확률이 점점 낮아지기 때문이다. 성능 비교를 위해 동일한 부호율을 가진 단일 길쌈부호가 같이 비교되었으며 곱부호가 더 좋은 성능을 보이는 것을 확인하였다.

그림 4는 3비트, 4비트 유한 연판정 길쌈부호와 무한 연판정 길쌈부호를 각각 수평부호로 갖는 곱

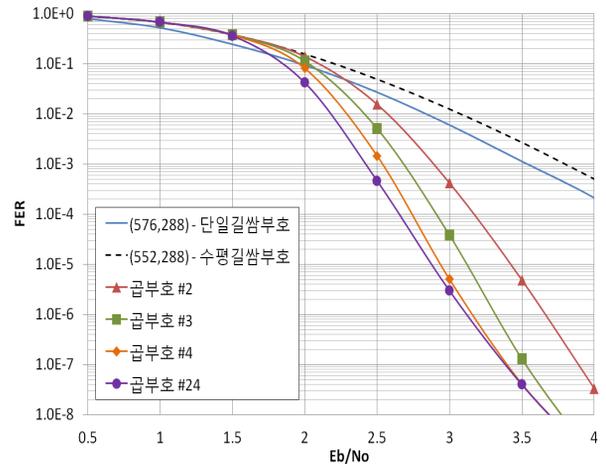


그림 3. 3비트 유한 연판정 테일비iting 길쌈부호를 수평부호로 가지는 곱부호의 성능
 Fig 3. Performance of a product code with 3-bit soft decision TB convolutional codes as horizontal codes

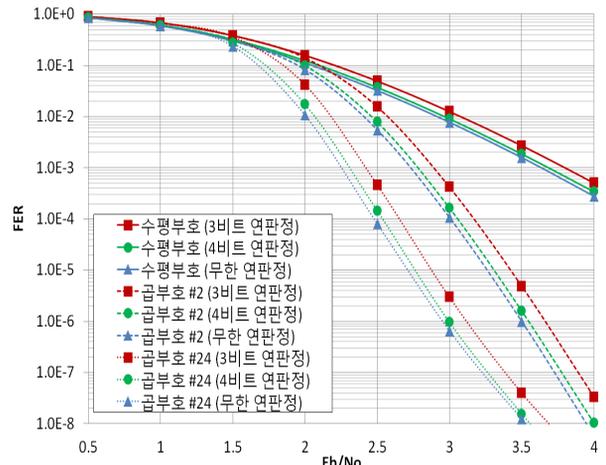


그림 4. 유한 비트 연판정 결합과 무한 연판정 결합 시 곱부호의 성능 비교
 Fig 4. Comparison of product codes with finite and infinite soft decision

부호들을 비교 하였다. 3비트 유한 연판정에는 표 1의 단순 田연산이 사용되었고 4비트 유한 연판정에는 표 2 ~ 표 5의 연산 테이블이 적용 되었다. 무한 연판정에는 채널에서 수신한 값을 그대로 연산에 이용하였고 (3)을 이용하여 田연산을 수행하였다. 결과적으로, 4비트 유한 연판정만을 이용한 길쌈부호의 곱부호가 무한 연판정을 이용한 길쌈부호의 곱부호의 성능에 근접함을 확인 하였다. 또한, 매번 연판정 데이터의 결합을 시도할 때 마다 (3)의 복잡한 tanh계산을 수행해야 하는 무한 연판정의 성능에 근접한 성능을 간단한 연산 테이블 만으로 수행함으로써 복잡도를 낮출 수 있었다.

VI. 결 론

본 논문에서는 곱부호를 위한 연판정 데이터의 결합을 이용한 복호 기법을 유한 비트 연판정만을 적용하여 사용하도록 새로운 연판정 데이터의 결합 법칙을 제시하였다. 무한 연판정을 이용한 높은 연산 복잡도를 요구하는 기존의 복잡한 tanh 연산을 분석하여, 유한 비트 연판정만을 이용한 연산 테이블을 작성하는 방법이 제시 되었다. 또한 연판정 데이터 적용의 용이성 때문에 널리 이용되고 있는 길쌈부호를 곱부호의 수평부호로 이용하였을 때, 제안된 방법을 적용하여 연산 테이블을 작성하였다. 모의 실험을 통해 확인한 성능은, 제안된 알고리즘에 의해 작성한 연산 테이블로 복호를 진행한 4비트 유한 연판정의 길쌈부호를 수평부호로 갖는 곱부호의 성능이 복잡한 연산을 그대로 수행한 무한 연판정의 곱부호의 성능에 근접함을 확인하였다.

References

[1] C. Yang, Y. Emre, and C. Chakrabarti, "Product code schemes for error correction in MLC NAND flash memories," *IEEE Trans. VLSI Systems*, accepted for publication.

[2] C. Yi and J. H Lee, "Interleaving and decoding scheme for a product code for a mobile data communication," *IEEE Trans. Comm.*, vol. 45, no. 2, pp. 144-147, Feb. 1997

[3] B. Shin, S. Hong, H. Park, J. No, D. Shin, "New Decoding Scheme for LDPC Codes Based on Simple Product Code Structure," [Online] <http://arxiv.org/pdf/1203.4040.pdf>

[4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996.

[5] R.Y. Shao, S. Lin, and M. P. C Fossorier, "Two decoding algorithms for tailbiting codes," *IEEE Trans. Comm.* vol. 51, no. 10, pp. 1658-1665, Oct. 2003

[6] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*.

Amsterdam, The Netherlands: NorthHolland, 1978.

[7] P. Elias, "Error-free coding," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 29-37, Sep. 1954.

[8] R. V. Cox and C. E. W. Sundberg, "An efficient adaptive circular Viterbi algorithm for decoding generalized tailbiting convolutional codes," *IEEE Trans. Veh. Technol.*, vol. 43, no. 11, pp. 57-68, Feb. 1994.

[9] H. H. Ma and J. K. Wolf, "On tail biting convolutional codes," *IEEE Trans. Comm.*, vol. 34, no. 2, pp. 104-111, Feb. 1986.

[10] Y. Yasuda, K. Kashiki, and Y. Hirata, "High-rate punctured convolutional codes for soft decision Viterbi decoding," *IEEE Trans. Comm.*, vol. 32, no. 3, pp. 315-319, Mar. 1984.

[11] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599 - 618, Feb. 2001.

[12] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. NJ: IEEE Press, 1999.

[13] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2nd Ed. NJ: Prentice Hall, 2004.

양 필 웅 (Pilwoong Yang)



2010년 2월 포항공과대학교 전
자전기공학과 공학사
2012년 2월 서울대학교 전기·
컴퓨터공학부 공학석사
2012년 3월~현재 서울대학교
전기·컴퓨터공학부 박사과정
<관심분야> LDPC 부호, 오류

정정부호, 통신시스템

박 호 성 (Hosung Park)



2007년 2월 서울대학교 전기공학부 공학사
2009년 2월 서울대학교 전기·컴퓨터공학부 공학석사
2009년 3월~현재 서울대학교 전기·컴퓨터공학부 박사과정
<관심분야> 압축 센싱, LDPC

부호, 통신시스템

홍 석 범 (Seokbeom Hong)



2007년 2월 서울대학교 전기공학부 공학사
2009년 2월 서울대학교 전기·컴퓨터공학부 공학석사
2009년 3월~현재 서울대학교 전기·컴퓨터공학부 박사과정
<관심분야> 네트워크 코딩, 압축 센싱, 오류정정부호

전 보 환 (Bohwan Jun)



2011년 2월 서울대학교 전기공학부 공학사
2011년 3월~현재 서울대학교 전기·컴퓨터공학부 석사과정
<관심분야> LDPC 부호, 오류정정부호, 통신시스템

노 종 선 (Jong-Seon No)



1981년 2월 서울대학교 전자공학과 공학사
1984년 2월 서울대학교 대학원 전자공학과 공학석사
1988년 5월 University of Southern California 전기공학과 공학박사

1988년 2월~1990년 7월 Hughes Network Systems, Senior MTS

1990년 9월~1999년 7월 건국대학교 전자공학과 부교수

1999년 8월~현재 서울대학교 전기·컴퓨터공학부 교수

<관심분야> 시퀀스, 협력통신, 시공간부호, 네트워크 코딩, LDPC 부호, OFDM, 이동통신, 암호학

신 동 준 (Dong-Joon Shin)



1990년 2월 서울대학교 전자공학과 공학사

1991년 12월 Northwestern University, 전기공학과 공학석사

1988년 12월 University of Southern California 전기공학과 공학박사

1999년 1월~1999년 4월 Research Associate (USC)

1999년 4월~2000년 8월 Hughes Network Systems, MTS

2000년 9월~현재 한양대학교 전자통신컴퓨터공학부 부교수

<관심분야> 디지털통신, 이산수학, 시퀀스, 오류정정부호, 암호학