

부호어의 무게 분포를 통한 터보 인터리버의 성능 분석

정희원 고태환*, 김주민*, 정덕진*

The Performance Estimation of Turbo Internal Interleaver Using Weight Distribution of Codewords

Taehwan Ko*, Jumin Kim*, Duckjin Chung* *Regular Members*

요약

본 논문에서는 유니온 상한과 같은 성능 한계(Performance bound)와 부호어의 무게 도수(Weight frequency)에 기반을 둔 인터리버의 성능 분석 기법을 제안한다. 새로운 기법의 적용을 위해 블록, 의사 랜덤, 그리고 3GPP 규격의 프라임 인터리버를 설계, 비교하였으며, 특히 S-랜덤 인터리버의 경우 검색 창 크기를 다르게 하여 분석 기법을 적용하여 보았다. 성능 분석을 위해 3GPP 규격에 부합되는 터보 부호화기 및 복호기를 MATLAB을 이용하여 설계한 후 AWGN 채널 환경에서 시뮬레이션 하였다. 분석 결과에 의하면, 두 기준을 모두 고려하였을 때 빠른 분석 시간과 함께 좀더 명확한 인터리버의 성능을 예측할 수 있음이 확인되었다.

ABSTRACT

In this paper, we suggest more precise performance analysis method of turbo interleavers based on two criteria; performance bounds like Union Bound and weight frequency of codewords. In order to present our new method, we employ block, pseudo random, and so-called prime interleavers in compliance of 3GPP standard, respectively. We also applied this method to S-random interleavers that have different window size, S. 3GPP complied turbo encoder, decoder, and AWGN channel are implemented by using MATLAB for our performance analysis. According to our analysis, both criteria should be taken into account coincidentally to predict the performance of newly designed interleavers.

1. 서론

최근 주목 받고 있는 IMT-2000의 표준 중 하나인 W-CDMA 시스템에서는, 멀티미디어 데이터 송수신의 신뢰성 있는 전송을 위해서 강력한 채널 코딩기법인 터보 코드^[1]가 채택되었다. 우수한 성능의 인터리버 설계는 좋은 성능을 가지는 터보 코드를 얻기 위해 고려되어 지는 요소 중에 하나이다. 더 나은 인터리버의 설계를 위해선 좋은 해밍 무게(good Hamming Weight)를 가지는 부호어를 생성해 낼 수 있는 인터리빙 패턴을 찾는 것이 요구된다.

부호어의 무게 분포를 이용한 인터리버의 성능 분석에는 크게 두 가지가 있다. 그 중 하나는 성능 한계를 이용한 분석 방법^{[2][3]}이고 다른 하나는 부호어의 도수 분포를 이용한 방법이다^[4]. 성능 한계의 관점에서, 부호화된 부호어의 무게는 커질수록 더 나은 BER 성능을 가지게 된다고 알려져 있다. 일반적으로 성능 한계를 통한 분석을 위해서는 무게가 1부터 4인 정보 열의 부호어의 무게 분포를 파악해야 하며, 이를 위해서는 매우 긴 분석 시간이 필요로 한다. 한편, 부호어의 무게 분포 도수 분포 관점에서, 같은 무게를 가지는 부호어의 누적된 형태를 이용하여 성능 분석을 수행한다. 동일 무게를 가지는 부호어가 국부적으로 많이 나타날 경우 인

* 인하대학교 전자전기컴퓨터공학부 집적회로연구실(kthinha@dreamwiz.com)

논문번호 : 010155-0626, 접수일자 : 2001년 6월 26일

※본 연구는 2001년도 시스템 IC 2010사업의 지원으로 수행되었음.

터리버의 성능에 안 좋은 영향을 주게 됨이 발표되었다. 이 경우 보통 무계가 2인 입력 열에 대해서 분석이 이루어진다.

따라서 본 논문에서는 3GPP 규격의 터보 부호에 대하여 무계가 2인 입력 열에 대한 부호어의 무계 분석을 앞에서 말한 두 가지 분석 방법이 적절히 고려되어진 인터리버 분석을 수행하였다. 터보 부호의 성능을 알아보기 위해 3GPP 규격의 터보 복호기 및 AWGN 채널을 MATLAB을 이용하여 디자인하여 시뮬레이션을 수행하였다. 블록, 의사 랜덤, 프라임^[5], 그리고 S-랜덤 인터리버^[6]와 같은 여러 인터리버를 사용하여 수행된 시뮬레이션 결과로부터, 본 논문에서 제안한 분석 방법이 인터리버의 성능 분석을 행함에 있어 좀더 명확하면서도 효율적임을 알 수 있다.

II. 이론적 배경

1. 터보 부호

터보 부호는 두개 또는 그 이상의 병렬 연쇄 조직 부호(concatenation systematic code)이며, 일반적인 터보 부호기를 그림 1에 도시하였다. 그림에서 길이가 k 비트인 데이터 블록, u 가 부호화에 입력되고, PAD블록에서 이 데이터 블록에 $N - k$ 개의 꼬리 비트를 추가하여 시퀀스 x_0 가 된다. 이 N 비트 시퀀스는 병렬로 M 개의 인터리버, I_i 와 구성 부호기(constituent encoder), ENC_i 에 공급된다. 각 인터리버는 정해진 인터리빙 패턴을 가지고 x_0 를 혼합한 후, 혼합된 출력을 구성 부호기에 공급한다. 각각의 구성 부호기는 출력 패리티 시퀀스 x_i 를 생성한다. M 개의 패리티 시퀀스를 더한 정보 시퀀스 x_0 는 연쇄적으로 출력되어 부호어를 형성하게 된다.

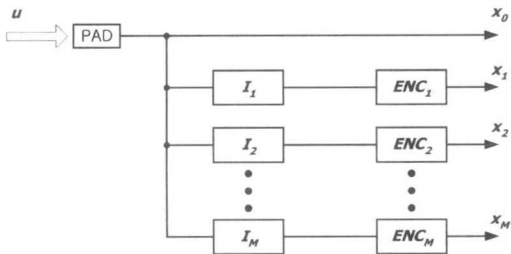


그림 1. 부호율 1/(M+1)의 터보 부호화기의 구조

일반적으로 사용되는 부호화기는 회귀적인 조직 길쌈(Recursive Systematic Convolutional, RSC) 부

호화기이다. 다시 말해 터보 부호화기는 일종의 길쌈 부호기이므로 터보 복호기가 수정된 형태의 비터비 알고리즘인 SOVA(Soft Output Viterbi Algorithm)를 사용하는 것이 가능하게 되며, 또한 회귀적이므로 비반복 부호기를 사용하였을 때 발생하는 출력 부호어의 좋지 않은 거리 특성을 보상할 수 있다.

터보 부호의 복호에는 반복 복호 방식이 이용되며, 이를 위해 주로 MAP(Maximum a posteriori)과 앞에서 언급한 SOVA 알고리즘이 복호화 알고리즘으로 사용된다. 각각의 계산된 비트에 대해 길쌈 부호의 복호에 사용되는 비터비 알고리즘의 출력이 0, 혹은 1인데 반해, 터보 부호 복호화 알고리즘의 출력은 각각의 비트에 대해 연속적인 값으로 나타난다. 비터비 복호기의 목표가 전송된 부호어의 최대 근사 계산을 찾음으로써 부호어의 오류를 최소화하는데 있다면, 터보 복호기는 부호어의 각각의 비트에 대한 사후 확률(posterior probabilities)을 계산함으로써 비트 오류의 최소화를 시도한다.

터보 복호기는 터보 부호화기에 있는 각각의 내부 구성 부호기를 위해 M 개의 내부 복호기로 구성되어 있다. 각각의 기본 복호기는 복호기로 전송되는 각 비트 열에 대해 연 판정을 얻기 하기 위한 복호를 수행하며 복호 과정의 반복 후, 모든 내부 복호기는 연 판정 출력을 다른 $M-1$ 개의 내부 복호기들과 공유한다.

2. 터보 내부 인터리버

터보 부호화기 및 복호기에서 인터리버는 연접 오류가 발생하였을 때 이것의 영향을 분산시켜서 오류 정정 능력을 향상시키는 역할을 한다. 인터리버의 크기가 커질수록 연접 오류를 분산시키는 능력이 증가하기 때문에 성능 향상이 있으며, 사용한 인터리버의 구조에 따라서 큰 성능 차이를 나타낸다. 특히 SNR이 낮은 경우에 있어 인터리버는 터보 부호의 성능을 크게 좌우하게 된다^[7]. 다시 말해 터보 내부 인터리버는 터보 부호의 특성상 복호기의 입력들간의 상관관계를 줄이기 위해 구성 부호기 사이에 위치하게 되는데 부호어 간의 상관 관계를 줄이기 위해선 부호어의 무계를 좀더 크게 하여야 하며 이는 최소 해밍 거리를 증가 시킴으로써 실현 가능하다^[3]. 무계 2인 입력열에 의해 구성 부호기가 낮은 무계의 부호어를 생성하는 과정과 동일한 입력열에 대해 내부 인터리버를 통한 좀더 큰 무계의 부호어 생성 과정을 그림 2에 나타내었다.

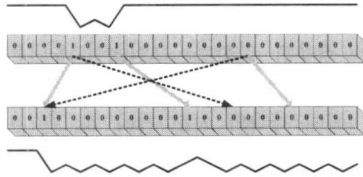


그림 2. 무게 2인 입력열에 의한 작은 해밍 거리 사건과 인터리버에 의한 해결 방법

터보 부호에 사용되는 인터리버의 종류에는 크게 블록, 대각, 순환 쉬프트 인터리버와 같이 일정한 간격으로 인접 비트의 순서를 바꾸는 균일한 구조를 가지는 형태의 인터리버와 의사 랜덤, S 랜덤 인터리버와 같이 인터리빙 패턴이 균일하지 않은 형태의 인터리버로 나눌 수 있다. 일반적으로 프레임 길이가 짧은 경우 균일한 인터리버가, 길 경우 비 균일 인터리버가 성능이 우수하다고 알려져 있다. 프레임 길이가 100 비트일 때 여러 형태의 인터리빙 패턴을 그림 3에 도시하였다.

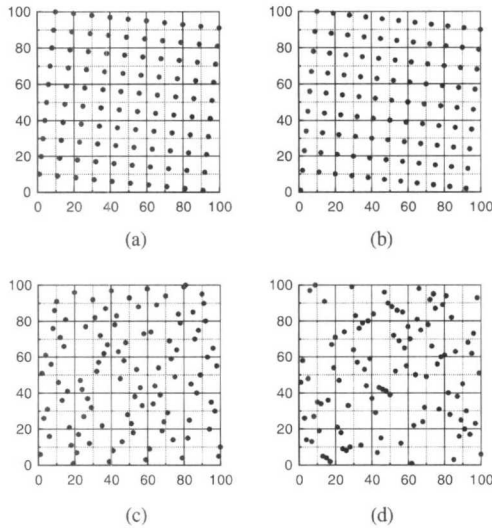


그림 3. 프레임 길이가 100 비트일 때 터보 인터리버의 인터리빙 매트릭스 (a) Block Interleaver (10X10)(b) Helical Interleaver (10 X 10) (c) Prime Interleaver (p = 11, v = 2)(d) S-Random Interleaver (S = 7)

3. 부호 특성에 기반 한 성능 한계

통신 시스템의 성능은 수신자가 얼마나 성공적으로 송신된 메시지를 구별해내는가와 관계된다. 채널 코드에서 부호어간의 차이의 정도는 해밍 거리 스펙트럼(hamming distance spectrum)에 의해, 선형 코드에서는 무게 분포에 의해 요약된다. 두 부호어 사이의 해밍 거리는 부호어의 서로 다른 위치나 심벌의 수를 가리킨다. 부호어의 무게는 부호어의 영

이 아닌 심벌의 수를 가리킨다. 코드의 무게 분포란 부호어들이 가질 수 있는 무게인 0부터 n까지 각각의 무게 별 부호어 개수의 리스트를 말한다. 이러한 수들을 $a_d, d = 0, 1, \dots, n$ 으로 표시하며 다양성(multiplicities)이라 한다. 또한 a_0 0인 가장 작은 거리 $d > 0$ 을 코드의 최소 거리, d_{min} 이라 한다. 이는 부호의 성능에 매우 큰 영향을 미친다. 일반적으로 코드 설계 시 큰 최소 거리를 만드는 것을 목표로 삼는다.

코드의 거리 스펙트럼에 기반 한 몇몇 유용한 성능 한계들이 있는데, 이러한 한계가 최우(Maximum Likelihood) 복호 알고리즘을 사용하여 복호 한다고 가정하면, AWGN 채널에서 최우 복호기가 수신된 부호어 c_i 에 대해 c_j 를 선택할 확률은 다음과 같다^[8].

$$P_r(C_i \rightarrow C_j) = Q\left(\sqrt{\frac{2d_{ij}RE_b}{N_0}}\right) \tag{1}$$

여기서 d_{ij} 는 부호어 c_i 와 c_j 사이의 해밍 거리를 말하고, R은 부호율, $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$ 는 정규화 된 가우시안 랜덤 변수(normalized Gaussian random variable)의 상위 꼬리 확률(upper tail probability)이다.

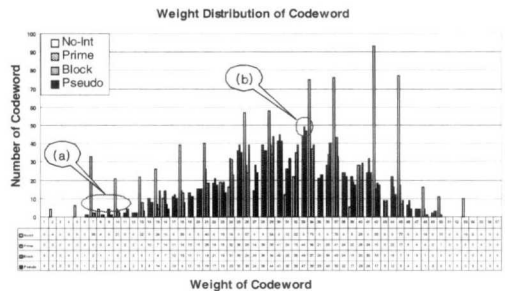


그림 4. 프레임 길이가 40 비트일 때 무게가 2인 입력 열에 대한 부호어의 무게 분포

4. 유니온 상한

수신된 부호어가 다른 부호어 중 하나로 잘못하여 복호 될 확률에 기반을 둔 복호 오류 확률의 상한을 유니온 상한(Union bound)이라 한다. 유니온 상한은 모든 $j \neq i$ 에 대해 전송된 부호어 c_i 가 c_j 로 잘못 선택할 오류 확률을 더함으로 형성된다.

$$P_e \leq \sum_{\substack{j=1 \\ j \neq i}}^M P_r(c_i \rightarrow c_j) = \sum_{\substack{j=1 \\ j \neq i}}^M Q\left(\sqrt{\frac{2d_{ij}RE_b}{N_0}}\right) \tag{2}$$

여기서 M 은 코드 내 부호어의 개수를 말한다. 코드의 무게 분포를 이용하면,

$$P_{\epsilon} \leq \sum_{d=d_{\min}}^n a_d Q\left(\sqrt{\frac{2dRE_b}{N_0}}\right) \quad (3)$$

큰 E_b/N_0 에서 유니온 상한은 식 (3)의 첫 번째 항이 지배적인데, 이는 거리 스펙트럼의 낮은 부분에 의해 주로 영향을 받음을 의미한다. 낮은 E_b/N_0 에서는 더 많은 항에 영향을 받으며, 특히 부호율이 순차적 복호기(Sequential decoder)가 동작할 수 있는 가장 높은 비율의 실질적인 한계인 연산을 통한 차단율(computational cutoff rate), R_0 에 도달하게 되면 지나치게 빠른 속도로 증가한다. AWGN 채널에서 R_0 은 다음과 같다[9].

$$R_0 = 1 - \log_2(1 + e^{-RE_b/N_0}) \quad (4)$$

III. 터보 코드의 부호어 무게 분석

터보 코드에 사용되는 인터리버에 관한 연구는 크게 두 가지로 나눌 수 있다. 하나는 어떻게 하면 성능이 우수하게 알려진 랜덤 인터리버와 비슷하거나 더욱 향상된 성능을 가지는 인터리버를 구조적으로 설계할 수 있을 것인가 하는 연구와 다른 하나는 부호화기에서 출력되는 부호어를 통해 성능에 영향을 미치는 요소를 분석하여 어떻게 하면 인터리빙 패턴을 최적화할 것인가에 관한 연구이다. 전자의 경우 S-랜덤 인터리버와 같이 설계 시 부호어의 최소 자유 거리를 크게 할 수 있도록 제한을 두어 설계하였을 때 매 프레임마다 새롭게 생성되는 랜덤 인터리버를 사용할 경우보다 성능이 우수함이 발표되었고, 후자의 경우에는 부호어의 무게 분포 분석을 통해 인터리버의 상한, 혹은 하한에 영향을 미치는 부호어, 혹은 부호어의 분포 형태에 따른 인터리버와의 관계를 파악하여 더 나은 성능의 인터리버를 제안하기도 하였다. 그러나 이와 같은 기존의 분석은 그 계산량이 매우 복잡하거나 명확하지 못하다는 단점이 있다. 따라서 본 논문의 목적은 무게가 2인 입력 열에 대한 부호어의 무게 분포를 통하여 성능 한계에 영향을 주는 최소 거리 및 도수 분포 등을 동시에 고려하여 인터리버의 성능을 좀 더 명확하고 효율적으로 분석하는데 있다.

이를 위해 프레임 길이가 40bit일 때 터보 코드의 성능 분석을 위한 부호어의 무게 분포를 그림 4에

나타내었다. 여기서 No-Int.이란 인터리버를 사용하지 않았을 경우를 말한다. 그림에서 유니온 상한에 주요한 영향을 주는 낮은 무게의 부호어를 (a)로 표시하였다. 이때 각각의 최소 거리, d_{\min} 은 프라임 인터리버가 9, 블록과 의사 랜덤 인터리버가 7이다. 따라서 유니온 상한의 관점에서 볼 때, 블록 인터리버와 의사 랜덤 인터리버의 성능이 프라임 인터리버 보다는 좋지 않을 것임을 유추해 낼 수 있다. 한편 같은 무게를 가지는 부호어의 누적이 가장 큰 부분, 다시 말해 부호어의 도수 분포가 가장 높은 부분을 (b)로 나타내었으며, 이때 각각의 도수 분포, $a_{d,\max}$ 는 무게가 33일 때 프라임이 44, 블록이 49, 의사 랜덤이 47을 가진다. 따라서 도수 분포의 관점에서는 프라임 인터리버의 성능이 가장 우수할 것이다.

IV. 실험 및 고찰

앞 절에서 설명한 터보 내부 인터리버의 성능과 부호어의 무게 분포와의 관계를 알아보기 위해 블록 인터리버, 의사 랜덤 인터리버, 그리고 3GPP 규격으로 채택된 프라임 인터리버와 같은 여러 가지 구조의 인터리버를 설계한 후 무게가 2인 입력 시퀀스에 대한 부호어의 무게 분포를 분석하였다. 그림 5와 6는 프레임 길이가 각각 192 비트와 1024 비트에 대한 부호어의 무게 분포이며, 표 1에 분석 결과를 도시하였다. 여기서 d_{\min} 은 부호어의 최소 해밍 거리를 의미하며, $a_{d,\max}$ 는 부호어 중 최대 도수, 그리고 d_{\max} 는 최대 도수의 해밍 거리를 의미한다. 이 값들을 유니온 상한에 영향을 주는 요소로 보고 이들이 터보 부호의 성능에 좋은 영향을 줄 경우를 'O'로, 그렇지 않을 경우를 'X'로 나타내었다. 분석 결과로부터, 192 비트의 경우, 의사 랜덤 인터

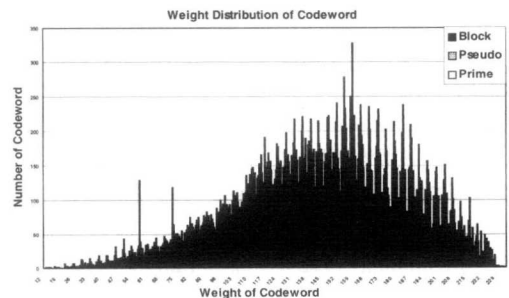


그림 5. 프레임 길이가 192 비트일 때 무게가 2인 입력 열에 대한 부호어의 무게 분포

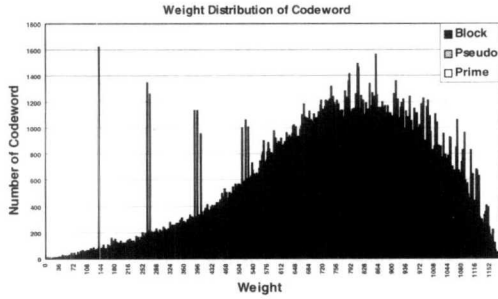


그림 6. 프레임 길이가 1024 비트일 때 무게가 2인 입력 열에 대한 부호어의 무게 분포

리버보다는 블록과 프라임 인터리버가, 1024 비트의 경우 의사 랜덤과 프라임 인터리버가 성능이 우수할 것임을 알 수 있다.

표 1. 부호어의 무게 분포를 통한 인터리버 성능 지표 분석

기	프레임 길이가 192 비트일 때			프레임 길이가 1024 비트일 때		
	블록	의사 랜덤	프라임	블록	의사 랜덤	프라임
d_{min}	26 (O)	13(X)	22(Δ)	45(O)	14(X)	22(Δ)
$d_{t,max}$	235(O)	328(X)	241(Δ)	1626(X)	1212(O)	1567(Δ)
d_{max}	168(O)	160(Δ)	152(X)	138(X)	756(Δ)	856(O)

이와 같은 예측의 정확성을 확인하기 위해 3GPP 규격의 터보 복호기 및 AWGN 채널을 MATLAB을 이용하여 디자인하여 시뮬레이션을 수행하였다. 이때 제한 길이는 4, 생성 다항식은 [13/15]₈, 복호 알고리즘으로는 Log-MAP 알고리즘, 반복 복호는 5회 실시하였고, 몬테카를로 방법에 의거하여 각각의 SNR point 마다 프레임 오류가 50회 일어날 때까지 시뮬레이션을 수행하였다. 그림 7과 8에 시뮬레이션 결과를 나타내었다.

한편, 제안된 분석 방법을 기존의 S-랜덤 인터리

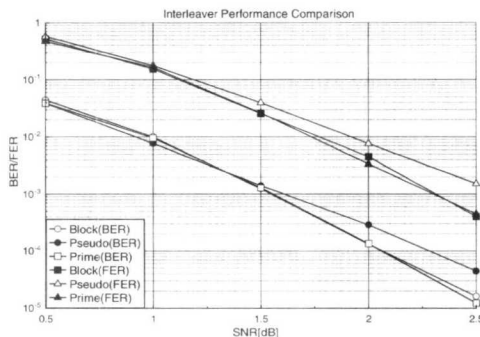


그림 7. 프레임 길이가 192 비트일 경우 각 인터리버에 대한 터보 부호의 성능 곡선

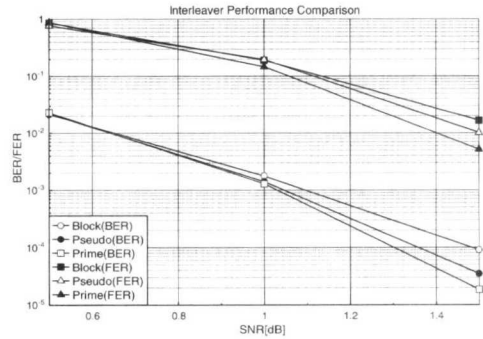


그림 8. 프레임 길이가 1024 비트일 경우 각 인터리버에 대한 터보 부호의 성능 곡선

버에도 적용해 보기 위하여 프레임 길이가 192 비트인 인터리버를 조사 창(Searching window), S가 각각 10, 30, 35가 되도록 설계한 후 앞서의 마찬가지로 무게 분석을 수행한 결과를 표 2에 나타내었다.

표 2 부호어의 무게 분포를 통한 S-랜덤 인터리버 성능 지표 분석표

기	프레임 길이가 192 비트일 때		
	S = 10	S = 30	S = 35
d_{min}	14 (O)	14(O)	14(O)
$d_{t,max}$	261(Δ)	265(X)	254(O)
d_{max}	156(Δ)	140(X)	172(O)

분석된 결과로부터 S=35인 인터리버의 성능이 가장 우수할 것으로 보이며, S=10인 인터리버가 그 다음으로 성능이 우수할 것이다. 따라서 S값을 증가시킨다 하더라도 반드시 인터리버의 성능이 향상되는 것은 아니며 상대적으로 인터리빙 패턴을 찾기 쉬운 작은 S값의 인터리버의 무게 분석을 통해 성

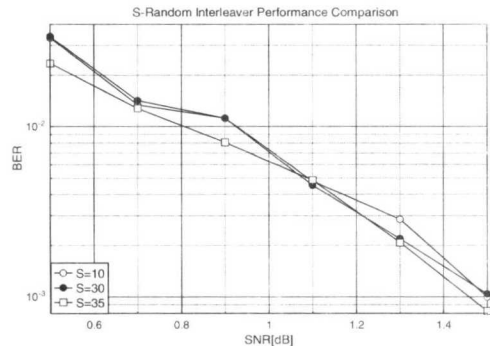


그림 9. 프레임 길이가 192 비트일 경우 S-랜덤 인터리버에 대한 터보 부호의 성능 곡선

능이 우수한 인터리버를 찾을 수 있을 것으로 사료된다. 이와 같은 분석 결과의 정확성을 확인 하기 위하여 앞에서와 같은 방법으로 S-랜덤 인터리버에 대한 시뮬레이션을 수행하여 보았으며, 결과를 그림 9에 나타내었다. 앞에서 예측한 바와 같이 S가 35 일 때 가장 좋은 성능을 보였으며 S가 10일 때 S가 30 일 때 보다 더 나은 성능을 보였다. 이로부터 본 논문에서 제안한 분석 방법이 인터리버의 성능을 예측함에 있어 좀더 명확하면서도 효율적임을 알 수 있다.

V. 결론

터보 코드의 성능을 결정하는 가장 중요한 요소로 인터리버를 들 수 있다. 인터리버를 적절히 설계함으로써 더욱 향상된 성능을 가질 수 있고, 그러기 위해서는 인터리버의 성능과 관계되는 요소를 찾아 최적화하여야 할 것이다. 본 논문에서는 무게가 2인 입력열에 대한 부호어의 무게 분석을 통해 인터리버의 성능 예측 기법을 제시하였고, 제안된 기법의 검증을 위해 3GPP 규격의 터보 복호기 및 AWGN 채널을 MATLAB을 이용하여 디자인하여 시뮬레이션을 수행하였다. 블록, 의사 랜덤, 프라임, 그리고 S-랜덤 인터리버와 같은 여러 인터리버를 사용하여 수행된 시뮬레이션 결과로부터, 본 논문에서 제안한 분석 방법이 인터리버의 성능 분석을 행함에 있어 좀더 명확하면서도 효율적임을 알 수 있었다. 또한 이와 같은 성능 예측 방법을 바탕으로 하여 좀더 성능이 우수한 인터리버를 설계할 수 있을 것으로 사료된다.

참고 문헌

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: turbo code," in Proc. ICC'93, (Geneva, Switzerland, May 1993), pp. 1064-1070.

[2] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," IEEE Trans. on Information Theory, pp. 409-428, Mar. 1996.

[3] P. Robertson, "Illuminating the structures of parallel concatenated recursive systematic (TURBO) codes," Proc. GLOBECOM'94, San

Francisco, CA, pp. 1298-1303, Nov. 1994.

[4] P. Jung and M. Na han, "Performance evaluation of turbo codes for short frame transmission system," Electron. Letter, vol 30 pp. 111-113, Jan. 1994.

[5] A. Shibutani, H. Suda, and Y. Yamao, "Performance of W-CDMA mobile radio with turbo codes using prime interleaver," Proc. IEEE VTC 2000-Spring, pp. 946-950, Tokyo, 2000.

[6] S. Dolinar and D. Divsalar, "Weight distributions for Turbo codes using random and nonrandom permutations," JPL TDA Progress Report, 42, August 1995, pp. 56-65.

[7] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolution codes," IEEE Trans. Comm. vol. 44, no. 5, pp. 591-600, May. 1996.

[8] S. Lin and D. J. Costello, JR. "Error Control Coding," Prentice-Hall, Inc., New Jersey, 1983.

[9] G. C. Clark and J. B. Cain, "Error Correction Coding for Digital Communications," Plenum press, New York, 1988.

고 태 환(Taehwan Ko) 정회원
 2000년 2월 : 인하대학교 전자재료 공학과졸업
 2002년 2월 : 인하대학교 전자재료 공학과 공학석사
 2002년 3월~현재 : 삼성정보통신 연구원



<주관심 분야> FEC 알고리즘, VLSI 구현, 신호처리

김 주 민(Jumin Kim) 정회원
 1993년 2월 : 인하대학교 전자재료공학과졸업
 1995년 8월 : 인하대학교 전자재료공학과 공학석사
 1999년 3월~현재 : 인하대학교 전자재료공학과 박사과정



<주관심 분야> FEC 알고리즘, VLSI 구현, 신호처리

정 덕 진(Duckjin Chung)

정회원



1970년 2월 : 서울대학교 전기공학
학과졸업

1984년 8월 : Utah State Univ.
전기공학과 공학석사

1988년 8월 : Univ. of Utah
전기공학과 공학박사

1980년 ~ 1988년: 한국전자통신
연구원 책임연구원

1989년 ~ 현재: 인하대학교 정보통신공학부 교수

<주관심 분야> FEC 알고리즘, 인공지능, Genetic
Algorithm, VLSI 구현, 신호처리