

# 3차원 알고리즘을 이용한 랜덤(or s-랜덤) 인터리버를 적용한 터보코드의 성능분석

공 형 윤<sup>†</sup> · 최 지 웅<sup>††</sup>

## 요 약

본 논문에서는 3차원 입출력 알고리즘을 랜덤 인터리버와 s-랜덤 인터리버에 적용하였으며, 이를 터보코드 인터리버에 적용하여 성능을 분석하였다. 인터리버의 성능은 인접 데이터간 최소 거리에 의해 결정되므로, 인접 데이터간의 최소거리를 증가시키는 방법으로 인터리버의 성능을 향상 시켰다. 3차원 알고리즘을 적용한 인터리버는 3차원 저장공간을 이용해 입력 데이터를 저장하고 랜덤하게 추출하는 방식이다. 이러한 방식은 기존의 랜덤 인터리버와 s-랜덤 인터리버에 비해 인접 데이터간 최소거리와 평균거리를 증가시킨다. 컴퓨터 시뮬레이션을 이용하여 3차원 알고리즘을 적용한 터보코드의 성능을 분석하였으며, 전송 환경을 가우시안 채널로 설정하였다.

## Performance Analysis of Turbo-Code with Random (and s-random) Interleaver based on 3-Dimension Algorithm

Hyung-Yun Kong<sup>†</sup> · Ji-Woong Choi<sup>††</sup>

### ABSTRACT

In this paper, we apply the 3-dimension algorithm to the random interleaver and s-random interleaver and analyze the performance of the turbo code system with random interleaver (or s-random interleaver). In general, the performance of interleaver is determined by minimum distance between neighbor data, thus we could improve the performance of interleaver by increasing the distance of the nearest data. The interleaver using 3-dimension algorithm has longer minimum distance and average distance compared to existing random-interleaver (s-random interleaver) because the output data is generated randomly from 3-dimension storage. To verify and compare the performance of our proposed system, the computer simulations have been performed in turbo code system under gaussian noise environment.

**키워드:** 3차원 알고리즘(3-Dimension algorithm), 최소거리(minimum distance), s-랜덤 인터리버(s-random Interleaver)

### 1. 서 론

차세대 이동 통신은 고속의 데이터와 영상정보를 전송할 수 있는 고효율 통신 시스템이어야 하며, 고도의 유무선 통합 망을 구현하여 고품질, 다기능 서비스를 제공할 수 있어야 한다. 디지털 통신에 있어서 통신 채널에서 첨가되는 피할 수 없는 무 기억 채널에 의한 랜덤 잡음, 기억채널에 의한 페이딩, 다중경로에 따른 위상 왜곡 등으로 인하여, 송신되는 신호는 랜덤 및 군집 오류를 포함하여 수신하게 된다. 그 영향으로 도착된 신호들이 서로 다른 위상을 갖게 되고 수신 신호가 왜곡된다. 높은 주파수와 Tropospheric 전파 라디오 채널들에 그러한 현상들이 일어난다. 채널이 메모리를 가졌다는 가정 하에서 에러는 더 이상 비트와 비트

사이에 서로 독립적으로 하나씩 랜덤하게 분포된 비트 에러들처럼 성격 되어질 수 없다. 대부분의 블록 부호나 길쭉 부호들은 랜덤하게 독립적인 에러들에 대항할 수 있게 설계된다. 이렇게 부호화 된 신호에 있어 채널이 메모리를 가졌 다함은 에러 발생 빈도가 감소함을 의미한다. 메모리를 가진 채널들에서 부호화 기술이 계속 제안되어져 왔으나, 이러한 부호화에 있어 가장 큰 문제점은 채널에서 자주 발생 되는 시변 통계의 정확한 모델을 얻기가 어렵다는데 있다. 정확한 통계적 성질이 아니라 오직 채널 기억의 기간(duration) 또는 span의 지식만을 요구하는 하나의 기술이 인터리빙이라는 시간의 차이를 이용하는 방법이다. 인터리빙이란 기억장치를 여러 부분으로 나누고 그 동작 주기에 조금씩 차이를 두어 동기적으로 고속화시키는 것을 말한다. 전송하기 전 부호화된 메시지를 끼워 넣고(interleaving), 수신한 후 빼냄으로써 주어진 시간동안 채널 에러를 연립으로 내보내

<sup>†</sup> 정 회 원 : 울산대학교 전자공학과 교수

<sup>††</sup> 준 회 원 : 울산대학교 대학원 전자공학과

논문접수: 2002년 1월 5일, 심사완료: 2002년 8월 5일

고 복호기에서는 마치 랜덤 한 에러처럼 조절할 수 있다. 실제의 경우에 있어 채널 메모리가 시간 분리를 감소시키므로 인터리빙 아이디어는 시간 내에서 부호 어 심볼들을 분리해 내는데 있다. 따라서 고속 통신의 멀티 미디어 서비스를 위한 시스템에서 인터리버는 중요한 역할을 담당하고 있다[1, 2]. 본 논문에서는 고용량의 데이터에서 우수한 성능을 보이는 랜덤 인터리버와 s-랜덤 인터리버에 3차원 알고리즘을 적용하여 성능을 개선시킨 3차원 랜덤 인터리버와 3차원 s-랜덤 인터리버에 관하여 연구하였으며, 또한 터보코드 시스템[3]에 3차원 알고리즘을 적용한 인터리버를 적용하여 성능을 분석하였다. 3차원 랜덤 인터리버는 입력 데이터를 x, y, z 세 개의 축으로 구성되는 배열 공간 안에 저장하고 무작위로 데이터를 추출하는 방식이고 3차원 s-랜덤 인터리버는 데이터를 무작위로 추출 할 때 저장공간의 z축 주소 값을 2보다 크게 제한하여 무작위로 추출한 데이터가 이전의 데이터와 비교해 제한한 값보다 큰 주소의 데이터를 추출하여 데이터간의 비트 거리를 크게 만드는 방식이다. 본 논문의 구성은 아래와 같다. 기존의 랜덤 인터리버와 s-랜덤 인터리버에 대한 기본 개념을 2장에서 소개하고, 3장에서는 3차원 랜덤 인터리버와 3차원 s-랜덤 인터리버의 구조와 동작에 대해 설명하였다. 4장에서는 컴퓨터를 이용해 성능 분석을 위한 모의 실험을 하였으며, 5장에서는 본 논문의 결론을 내린다.

**2. 랜덤 인터리버와 s-랜덤 인터리버의 소개**

인터리버는 입력된 데이터의 거리분포를 개선시켜 전송 채널에 의한 연접 오류(burst error)를 최소화시키는 방식이다. 입력 데이터가 많을 경우 블록 인터리버 같은 구조적 인터리버보다는 랜덤 인터리버나 s-랜덤 인터리버를 사용하는 것이 시스템의 성능을 향상시킨다.

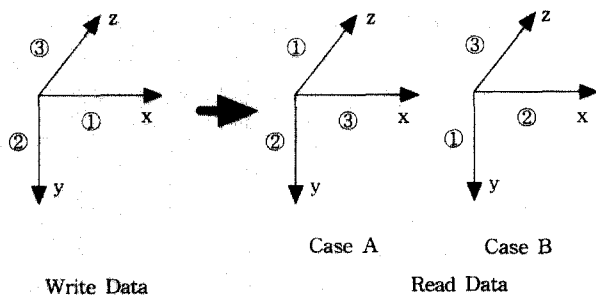
**2.1 랜덤 인터리버, s-랜덤 인터리버의 특징**

랜덤 인터리버와 s-랜덤 인터리버는 인터리빙이 무작위적인 랜덤을 의미하는 것이 아니라 랜덤 변수의 발생에 따라 블록단위의 전송을 하는데 시행착오를 거쳐서 성능이 좋은 랜덤 알고리즘에 따르는 것을 의미한다[1]. 하지만 실시간 형태의 인터리버는 구현이 불가능하며, 단지 랜덤 수열 발생기에 의해 발생하는 수열의 저장을 통해 랜덤 인터리버를 구성하게 된다. 랜덤 인터리버의 문제점은 랜덤 성 자체에 기인한다. 즉 무작위로 인터리빙을 위한 데이터를 선택하다보면 오히려 성능이 나쁜 랜덤 인터리버를 얻을 가능성도 있다. 따라서 연속하는 몇 개의 데이터들을 인터리빙한 후에 특정거리(s값) 이상으로 떨어지도록 인터리버를 구

성함으로써 성능이 나쁜 인터리버를 얻을 확률을 줄일 수 있다. s-랜덤 인터리버는  $S < \sqrt{N/2}$ 가 되도록 데이터간의 거리를 조절함으로써 타당한 시간 안에 만족할 만한 결과를 얻을 수 있다.

**3. 3차원 알고리즘**

본 장에서는 3차원 알고리즘을 적용하여 성능을 개선시킨 3차원 랜덤 인터리버와 3차원 s-랜덤 인터리버의 구조와 동작 특성에 대하여 알아본다. 터보코드는 인터리버에 의해 전송 데이터의 부호화처리가 이루어지기 때문에 인터리버의 성능이 부호기 전체의 성능을 좌우시키는 중요한 요소가 된다. 터보코드의 성능을 좌우할 수 있는 중요한 요소 중의 하나로 인터리버의 종류와 블록의 크기를 들 수 있다[4, 5]. 인터리버의 크기가 커질수록 부호기의 성능을 향상시키게 된다. 3차원 알고리즘은 2차원의 인터리버 개념에 하나의 차원을 더한 것으로 만약 이를 블록 인터리버에 적용한다면  $k \times l \times m$ 의 축을 가지는 3차원 공간상의 메모리가 쌓여진 구조가 된다[6]. 3차원 알고리즘을 적용한 인터리버는 2차원 인터리버와 구조적인 특성은 비슷하지만 동작의 특성이 간단하다. 3차원 블록 인터리버의 경우에는 출력되어지는 방향((그림. 1) 참조)에 따라서 비트거리, 블록거리가 달라져서 성능을 변화시키는 요인이 된다. 인터리버는 인접한 비트간의 연속적인 오류(burst error)에 대하여 우수한 성능을 가지도록 설계를 하여야 한다. 3차원 블록 인터리버는 3차원의 축(x, y, z)을 가지고 어느 하나의 축을 이용하여 입력과 출력을 하는 것으로 기존의 2차원 알고리즘을 이용한 블록 인터리버와 비교하여 간단한 구조를 가지면서, 비트간의 최소거리를 최대화 할 수 있는 방식이다.

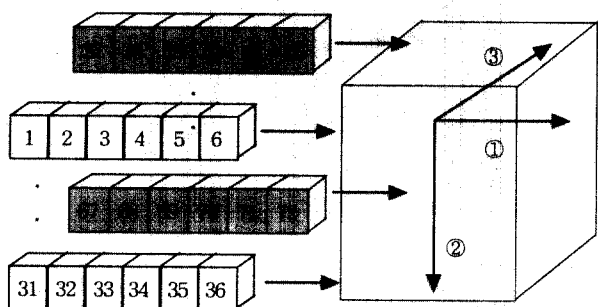


(그림 1) 3차원 블록 인터리버의 데이터 입·출력 순서

**3.1 3차원 알고리즘을 적용한 random-Interleaver 설계**

(그림 2)에서 입력 시퀀스를 저장하는 배열공간은  $x \times y \times z$  구조가 되고, 거기서 입력시퀀스를 랜덤하게 추출해 낸다. 입력 시퀀스의 3차원 구조의 저장순서는 x, y, z축 순서이

다. 3-D 랜덤 인터리버의 알고리즘은 3차원 구조의 저장공간에 입력 데이터 시퀀스를 입력하고 각각의 x, y, z축 주소 값에 랜덤 값을 발생시켜 입력 데이터를 추출해낸다. 데이터를 추출할때 이전의 데이터와 현재 추출한 데이터의 동일 여부에 따라 만약 이전의 데이터와 동일한 주소 값이 나왔다면 다시 랜덤 값을 발생시켜 데이터를 추출해 내는 방법이다. 즉 입력은 3차원 구조에 1, 2, 3, 4, ...의 순서로 메모리에 입력되고 출력은 3차원 구조의 x축, y축, z축 각각의 주소 값을 랜덤 수로 생성시켜 입력 시퀀스를 랜덤한 데이터 배열로 바꾸는 방식이다. 여기서 숫자가 나타내는 것은 입력 데이터의 순서를 나타낸 것이다.



(그림 2) 3차원 블록 인터리버의 데이터 입력 순서

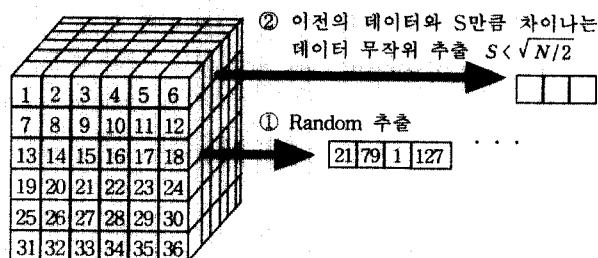
### 3.2 3차원 알고리즘을 적용한 s-random Interleaver 설계

3차원 s-랜덤 인터리버는 입력 데이터 시퀀스를 3차원 구조에 쓰는 방식은 3차원 랜덤 인터리버와 동일하지만 랜덤 인터리버와는 달리 인접 데이터간의 최소 거리는 유지하도록 제한하게되어 있다. s-랜덤 인터리버에서 s는 어떠한 정수를 의미하는데, 이 s만큼을 최소 거리로 유지하면서 랜덤 인터리빙이 이루어짐을 의미한다. 즉, 추출되어질 데이터의 간격이 최소  $\pm s$  가 되도록 데이터를 랜덤으로 추출한다. 만약  $\pm s$  안에 랜덤으로 추출한 데이터가 있다면 이 값은 버리고 다시 추출하는 형식이다. 본 논문에서는 두 가지의 3차원 s-랜덤 인터리버를 제안한다. 첫 번째는 3차원 구조에 입력 데이터를 저장하고 기존의 s-랜덤의 특성과 같은 알고리즘을 가지고 데이터를 추출하는 구조를 가지는 인터리버이며 (Mode A), 두 번째는 일정거리 이상으로 추출하는 3차원만의 독특한 알고리즘으로 데이터를 추출하는 인터리버 (Mode B)이다.

#### 3.2.1 3차원 알고리즘을 적용한 s-랜덤 인터리버 Mode A

3차원 s-랜덤 인터리버는 3차원 랜덤 인터리버의 알고리즘과 비슷하나 제약 조건을 추가한다((그림 3) 참조). 입력 데이터 시퀀스를 랜덤하게 추출하는 기본 알고리즘에 추출한 데이터가 어느 위치에서건 s이상의 차이가 나게 제약조

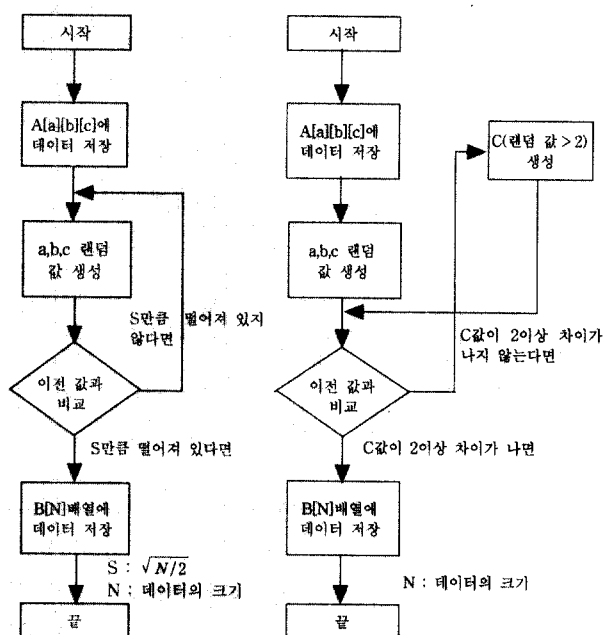
건을 두어 나쁜 랜덤 데이터를 얻을 가능성을 줄이는 것이다. 몇 개의 데이터를 랜덤 인터리빙한 후에 특정거리 이상 떨어지게 하는 s 값은 기존의 s-랜덤 인터리버에서 제안된  $S < \sqrt{N/2}$  값으로 지정하였다.(여기서 N은 입력 데이터 시퀀스의 수이다.)



(그림 3) 3차원 알고리즘을 적용한 s-랜덤 인터리버의 출력 (Mode A)

#### 3.2.2 3차원 알고리즘을 적용한 s-랜덤 인터리버 Mode B

두 번째로 제안하는 3차원 s-랜덤 인터리버는 3차원 구조의 특성을 이용한 알고리즘이다((그림 4) 참조). 입력 데이터를 3차원 구조에 저장하고 랜덤한 데이터를 추출하는 것은 3차원 랜덤 인터리버의 구조와 같고 랜덤 한 s값을 정하는데 있어 3차원적인 구조를 이용한 방식이다. 3차원 s-랜덤 인터리버의 Mode B의 알고리즘은 다음과 같다. 우선 입력 데이터를 3차원 구조의 저장공간에 저장하고 각각의 x, y, z축의 주소 값을 랜덤 수로 발생시킨다. 일정한 데이터를 랜덤으로 추출한 뒤 z 축상의 주소 값  $c > 2s$  값으로 지정하고 이전 데이터와 비교해 랜덤 데이터가 제한하는 주소



(그림 4) 3-D S랜덤 인터리버의 순서도 (Mode A, Mode B)

값 이상으로 나오도록 구성한다. 즉 3차원 구조 행렬을 구성한 후 z 축 상의 주소 값을 제한하는 방식이다. 이때 입력 데이터의 시퀀스의 길이 N이다. z 축 상의 주소 값을 2 이상으로 제한한 이유는 만약 랜덤한 데이터가 5×5×5 3차원 구조에서 이전의 값이 A(5)(5)(1)이 나왔을때 현재의 데이터가 A(1)(1)(2)과 같은 인접한 랜덤 데이터가 나오지 않도록 하기 위한 조건이다. 3-D s-랜덤(Mode B)가 기존의 s-랜덤이나 3-D S-랜덤(Mode A)와 다른 점은 기존의 알고리즘은 현재의 데이터와 이전의 데이터의 차이를 제한하는 방식이고 3-D s-랜덤(Mode B)는 z축 상의 주소 값을 제한하는 방식이다.

4. 모의 실험

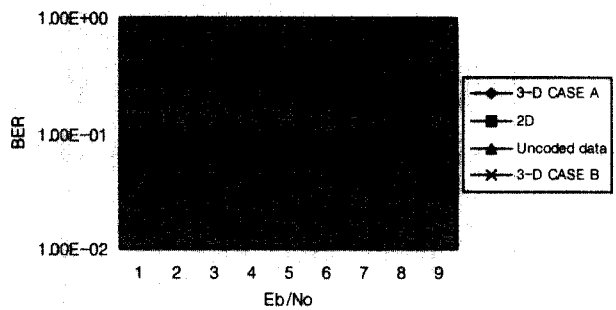
본 장에서는 3차원 랜덤 인터리버와 s-랜덤 인터리버를 터보코드에 적용하여 SOVA 디코더를 이용하여 모의실험을 하였다. 이때 데이터의 길이 N = 27000, N = 64000, N = 125000개로 나누어 실험하였으며, 채널 환경은 gaussian 환경으로 설정하였다. <표 1>은 입력 데이터의 길이에 따른 비트 거리를 평균적으로 나타낸 것이다. 표에서와 같이 s-랜덤과 3-D(3차원 알고리즘) s-랜덤 인터리버는 비교적 랜덤 인터리버나 3-D 랜덤 인터리버에 비해 많은 비트 거리를 가지며, 특히 3-D s-랜덤 인터리버(Mode B)는 다른 인터리버에 비해 데이터의 비트 거리가 더 큰 것을 알 수 있다. 이와 같이 랜덤 인터리버나 3-D 랜덤 인터리버는 랜덤성 자체에 그 특성이 좌우되기 때문에 나쁜 랜덤 특성이 나올 수도 있고 N = 27000일 경우와 같이 아주 좋은 특성이 나올 가능성도 있다. 하지만 평균적으로 볼때 일정 거리 이상의 비트 거리를 제한하는 s-랜덤이나 3-D s-랜덤 인터리버 보다는 성능이 다소 떨어진다.

<표 1> 인터리버에 따른 비트거리(평균) 특성

N / 종류	랜덤 인터리버	S랜덤 인터리버	3-D 랜덤 인터리버	3-D S랜덤 인터리버 (Mode A)	3-D S랜덤 인터리버 (Mode B)
27000 S = 160, c = 3	2061	2887	2375	2411	3034
64000 S = 250, c = 3	6517	6889	6687	7315	7858
125000 S = 350, c = 3	12590	13204	12700	13321	14094

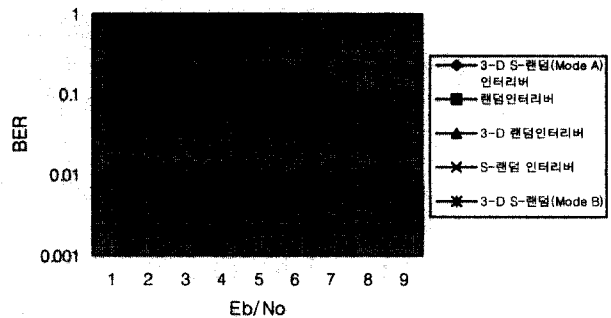
(그림 5)는 3-D 알고리즘을 이용한 블록 인터리버(Case A, & Case B, (그림 1) 참조)와 기존의 블록 인터리버를 터보코드에 적용하여 BER 성능을 비교한 것이다. 디코딩 방법으로는 MAP방식을 이용하였다. CASE A는 최소 비트 거리가 2차원 만큼의 거리를 가지게 되기 때문에 기존의 블

록 인터리버와 비교하여 좋은 결과를 얻을 수 있다. 그러나, x, y, z의 입력에 y, x, z의 출력(CASE B)은 블록거리, 비트거리 모두 기존의 2차원 블록 인터리버보다 적은 것을 알 수 있다. CASE B는 블록거리, 비트거리 모두 1차원의 성질을 가지게 되고, 만약 같은 크기의 인터리버를 구성한다면 하나의 차원이 가지는 거리는 2차원 블록 인터리버보다 3차원 블록 인터리버의 크기가 작기 때문에 상대적으로 거리가 작아진다. CASE B의 성능분석에서는 기존의 2차원 블록 인터리버방식 보다 0.2dB 정도 나쁜 부호화이득을 보인다.



(그림 5) 가우시안 채널에서의 3차원 블록 인터리버의 성능분석

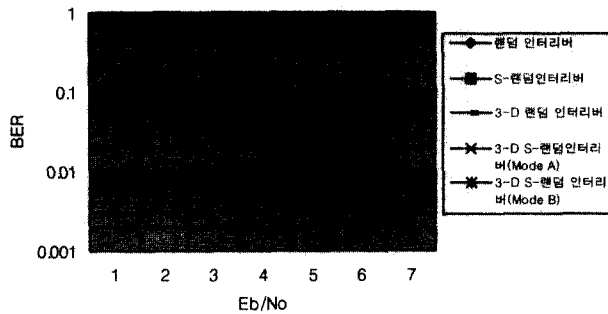
(그림 6)은 터보 코드 복호기로 SOVA 방식을 이용하여 데이터의 길이가 27000개 일때 3-D s-랜덤 인터리버(Mode A, Mode B), 3-D 랜덤 인터리버, 기존의 랜덤 인터리버, 기존의 s-랜덤 인터리버의 BER 성능을 각각 비교하여 나타낸 그래프이다. 기존의 랜덤 인터리버의 성능이 제일 낮고 3-D 랜덤 인터리버와 3-D s-랜덤 인터리버(Mode A)를 비교했을때 평균 0.1dB 정도의 차이로 Mode A 인터리버가 좋은 성능을 나타낸다. 이 중 3-D s-랜덤 인터리버(Mode B)가 가장 나은 성능을 보인다. 기존의 s-랜덤 인터리버와 3-D s-랜덤 인터리버(Mode A)를 비교했을때는 평균 0.2 dB로 s-랜덤 인터리버가 나은 성능을 보인다. 3-D s-랜덤 인터리버(Mode A)는 알고리즘 상으로 s랜덤 인터리버와 비슷하고 비트거리에 제한을 두었다고 할지라도 랜덤으로 데이



(그림 6) N = 27000(30 × 30 × 30) 일때의 결과

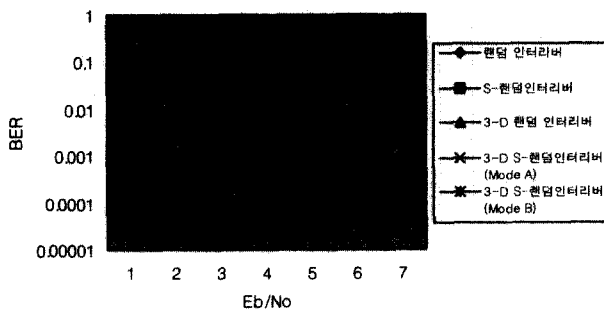
터블 추출하는 것에 바탕을 두므로 성능이 나빠질 수도 있다.

(그림 7)은 터보 코드 복호기로 SOVA 방식을 이용하여 입력 데이터가 64000개 일때의 그래프이다. 기존의 랜덤 인터리버와 3-D 랜덤 인터리버는 평균 0.1dB의 차이를 보이고 기존의 s-랜덤 인터리버와 3-D s-랜덤 인터리버(Mode A)는 평균 1.5dB의 차이를 보인다. s-랜덤 계열의 인터리버들이 랜덤 계열보다 나은 성능을 나타낸다.



(그림 7) N = 64000(40 × 40 × 40) 일때의 결과

위의 (그림 6)~(그림 8)에서 기존의 랜덤 인터리버나 3차원 랜덤 인터리버는 평균적으로 비슷한 BER 성능을 나타내고 랜덤 인터리버의 나쁜 특성이 나올 가능성을 줄이기 위해 비트거리의 일정 거리이상으로 조절한 s-랜덤 계열의 인터리버는 랜덤 인터리버 계열의 인터리버에 비해 좋은 성능을 보인다. 3차원 s-랜덤 인터리버(Mode A)는 기존의 s-랜덤 인터리버에 비해 평균적으로 0.1dB 정도 좋은 성능을 보인다. 또한, 3차원 s-랜덤 인터리버(Mode B)는 기존의 s-랜덤 인터리버에 비해 평균 0.3dB의 부호화 이득을 보임을 알 수 있다. 3-D s-랜덤 인터리버(Mode A)는 기존의 s-랜덤 인터리버와 비교해 그리 향상된 성능을 나타낸다고 말할 수는 없지만 3-D s-랜덤 인터리버(Mode B)의 경우 인접 데이터간의 거리를 제한하는 알고리즘의 변화에 따라 s-랜덤 인터리버의 성능이 개선될 수 있음을 잘 말해 주고 있다.



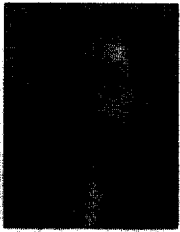
(그림 8) N = 125000(50 × 50 × 50) 일때의 결과

### 5. 결 론

터보 코드에서 부호화한 비트들의 상관성이 증가할수록 복호화 동작을 위한 많은 정보를 얻을 수는 있겠지만 에러들에 대한 상관성이 증가하게 된다. 따라서 에러들 사이에 상관성을 제거하기 위해 인터리버를 사용하며 인터리버의 크기나 구조에 따라 복잡성이나 지연의 문제도 수반하게 된다. 본 논문에서는 터보 코드의 성능 평가에 가장 많이 쓰이는 랜덤 인터리버, s-랜덤 인터리버에 대한 성능과 비교해 3차원 랜덤 인터리버와 3차원 s-랜덤 인터리버에 대한 특징을 살펴보고 컴퓨터 시뮬레이션을 통해 성능을 알아보았다. 3차원 랜덤 인터리버는 랜덤 인터리버에 비해 나은 성능을 나타내지는 못했지만 3차원 s-랜덤 인터리버인 Mode A와 Mode B는 각각 0.1dB와 0.4 dB로 기존의 s-랜덤 인터리버에 비해 나은 성능을 나타낸다. 따라서 3차원 s-랜덤 인터리버를 IMT-2000의 음성채널이나 멀티미디어 서비스 시스템에 적용한다면 보다 우수한 채널 부호화의 효과를 얻게 될 것이며, s-랜덤 인터리버가 쓰이는 많은 다른 시스템에 적용이 가능 할 것이다. 향후 과제로는 최적의 3차원 s-랜덤 인터리버 알고리즘을 위해 다양한 비트거리 제한 방법의 연구와 이러한 인터리버를 무선 랜 시스템이나 인터리버를 사용하는 여러가지 시스템에 적용하여 성능을 검증하는 것이다.

### 참 고 문 헌

- [1] Bernard Sklar, "DIGITAL COMMUNICATIONS," Second Ed. Prentice Hall, pp.461-477, 2001.
- [2] John G. Proakis, "DIGITAL CUMMUNICATIONS," Third Ed. McGraw-Hill, 1995.
- [3] Mustafa Erooz, A. Roger Hammons Jr., "On The Design of Prunable Interleavers for Turbo Codes," Vehicular Technology Conference, IEEE 49th, Vol.2, pp.1669-1673, 1999.
- [4] Charles C. Wangs, "On the Performance of Turbo codes," Military Communications Conference, 1998. MILCOM 98, Proceedings., IEEE, Vol.3, pp.987-992, 1998.
- [5] Johan Hokfelt, Ove Edfors and Torleiv Maseng, "Interleaver design for turbo codes based on the performance of iterative decoding," ICC'99, IEEE International Conference on Communications, Vol.1, pp.93-97, 1999.
- [6] 공형윤, 이창희, "새로운 블록 인터리버를 이용한 터보코드의 성능 분석", 2000년도 한국통신학회 하계종합학술대회, pp. 645-648, 2000.



### 공형운

e-mail : hkong@uou.ulsan.ac.kr

1989년 New York Institute of Technolgy  
(학사)

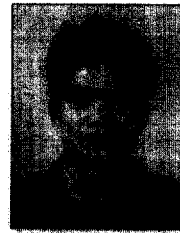
1991년 Polytechnic University(석사)

1996년 Polytechnic University(박사)

1996년~1998년 LG 멀티미디어연구실 PCS  
팀장 및 LG 그룹 회장실 전략사  
업개발단 과장

1998년~현재 울산대학교 전자공학과 교수

관심분야 : 변조방식(OQDM, QAM, ...), 코딩(LDPC, Turbo-  
code), SDR, Wireless Internet



### 최지웅

e-mail : tmwstw@orgio.net

2001년 울산대학교 전자공학과 졸업  
(학사)

2001년~현재 울산대학교 전자공학과  
재학(석사)

관심분야 : 무선통신, 변복조, 채널코딩