

# 호핑방식을 적용한 블록 인터리버를 이용한 터보코드의 성능분석

공 형 윤<sup>†</sup>

요 약

현대 디지털 통신시스템에서 없어서는 안되는 중요한 요소 중의 하나가 오류정정 부호화 기법이다. 본 논문에서는 디지털 통신 시스템에서 놀라운 성능으로 인하여 주목을 받고있는 터보코드에 비균일 인터리버의 일종인 호핑방식의 알고리즘을 적용한 인터리버를 터보코드에 적용하여 성능을 비교 분석하였다. 기존의 블록 인터리버는 메모리의 입력과 출력이 행과 열에 대해서 차례대로 이루어지는 반면, 호핑방식을 적용한 인터리버는 행과 열을 건너뛰면서 입출력이 이루어지므로 이중으로 인접 데이터간의 최소거리와 평균거리를 증가시킨다. 제안된 인터리버의 성능분석과 비교를 위해 가우시안 잡음 환경 하에서 컴퓨터 모의 실험을 하였다.

## Performance Analysis of Turbo Code with Block Interleaver using Hopping Method

Hyung-Yun Kong<sup>†</sup>

ABSTRACT

Channel coding is one of the most important things to improve digital communications. In this paper, we analyze the performance of turbo code with block interleaver using hopping algorithm (i. e., non-linear interleaver) for high speed multi-media service. The input and output of conventional block interleaver is achieved by the order of column and row, but hopping algorithm is achieved by hopping the column and row that increase the minimum distance and average distance between the nearest data dually. To verify and compare the performance of an proposed method the computer simulation have been performed using turbo code in gaussian channel environment.

키워드 : 부호화(coding), 호핑방식(hopping method), 비균일 인터리버(non-linear interleaver)

### 1. 서 론

차세대 멀티미디어 통신은 많은 양의 데이터와 영상정보를 고속으로 전송할 수 있어야 한다. 터보코드에서 부호화한 비트들의 상관성(correlation)이 증가할수록 부호화 동작을 위한 많은 정보를 얻을 수는 있겠지만 에러들에 대한 상관성 또한 증가하게 된다. 따라서 에러들 사이의 상관성을 제거하기 위해 인터리버를 사용하며 인터리버 크기에 따라 상관성을 어느정도 제거해 줄 것인지도 어려운 문제이며 복잡성이나 지연의 문제도 수반하게 된다. 인터리버의 역할은 일반적인 채널 환경에서는 연접에러를 랜덤에러로 전환하여 주는 역할을 하게되며 터보코드에서 인터리버의 역할은 첫 번째 구성부호기에서 낮은 무게(low-weight)를 출력하는 입력열의 특징한

패턴이 인터리버를 통해서도 그대로 나타나게 되면, 두 번째 구성 부호기에서도 낮은 무게의 패리티 열이 출력되며, 전체적으로 낮은 무게의 부호어가 생성되어 터보 부호의 성능이 열화되므로 상관관계가 있는 정보를 효과적으로 상관관계가 없는 정보로 전환하기 위해 정보 비트의 입력순서를 재배열하여 에러 패턴을 제거하는 역할을 한다. 즉, 하나의 RSC(recursive systematic convolutional code) 부호기의 입력 패턴이 작은 거리 특성을 주는 경우, 다른 하나의 RSC 부호기로의 입력 패턴이 큰거리 특성을 줄 수 있도록 입력 패턴에 대한 치환(permutation)을 수행하는 것이다. 터보코드의 성능을 좌우하는 요소에는 여러가지가 있다. 우선 부호기의 측면에서 바라보면, RSC 부호의 생성함수(generator)와 RSC 부호기 내의 메모리 개수이다. 이것은 터보코드를 구성하고 있는 것이 길쌈부호의 일종인 RSC 부호이므로 길쌈부호에서와 마찬가지로

<sup>†</sup> 정 회 원 : 울산대학교 전기 전자 정보시스템공학부 교수  
논문접수 : 2002년 6월 5일, 심사완료 : 2002년 9월 17일

가지로 최적의 성능을 나타낼 수 있는 생성기를 구성해야 하며, RSC 부호기의 개수가 증가할수록 성능은 향상된다. 부호기 측면에서 터보코드의 성능을 좌우하는 또 하나의 요소는 두 RSC 부호를 연결하는 인터리버이다. 인터리버의 사이즈가 클수록 성능은 좋아지며, 인터리버의 종류도 일반적인 균일 인터리버보다는 비균일 인터리버가 성능을 더욱 향상시킬 수 있다고 알려져 있다. 이것은 터보코드의 거리특성과 깊이 연관이 되어있는데, 터보코드의 거리특성과 생성함수와의 관계 그리고 비균일 인터리버의 필요성 등에 대해서는 [1]에 자세히 언급되어 있다. 인터리버의 사이즈가 클수록 성능은 향상되지만 그만큼 지연이 생긴다는 의미이므로 이 또한 trade-off 분석이 요구된다고 할 것이다. 앞서 언급했듯이 터보코드의 놀라운 성능은 부호기에 내재하는 인터리버와 복호기에서 수행되는 반복 복호에 기인하는 것이므로 그만큼 놀라운 성능을 얻기 위해서는 엄청난 지연시간이 요구된다고 할 수 있다. 수많은 연구가 수행되면서, 터보코드의 거리특성과 성능과의 관계라든가, 이에 따라 효율적인 인터리버 설계에 대한 연구 결과[2]가 쏟아져 나왔다. 뿐만 아니라, 독일의 University of Kaiserslautern을 중심으로 하여 통신시스템에서의 실시간 통신서비스를 위한 터보코드 부호화방식[3] 개발에 많은 노력이 투자되었다. 위에서 언급한 것 외에도 터보코드에서의 출력에서 회귀되는 정보를 이용하여 반복적으로 복호를 수행하여 성능을 향상시키는 기법은 여러가지 응용분야에 적용될 수가 있는데 이러한 원리를 응용할 수 있는 몇가지를 다음과 같이 요약할 수 있다. 블록코드를 이용한 터보코드[4], TCM(trellis coded modulation)에서의 터보코드 방식[5], 등화기[6] 그리고 Multi-user detection in CDMA system[7].

또한, 터보코드의 성능은 자유거리(free distance)로 설명할 수 있다. 자유거리란 0 부호어를 제외한 가능한 모든 부호어 무게 중 최소 해밍무게를 말한다. 좋은 성능을 갖는 터보부호는 자유거리가 크고, 거리가 낮은 부분의 부호어 수가 적은 부호일 것이다. 따라서 부호어의 거리 스펙트럼에 큰 영향을 미치는 인터리버는 터보코드에서 매우 중요한 역할을 한다 [8-11]. 본 논문에서는 기존의 블록 인터리버의 성능을 개선시킨 호핑방식의 블록 인터리버를 터보코드에 적용하여 성능을 분석하였다. 본 논문에서 제시한 인터리버에 대하여 간단히 설명하면 다음과 같다. 기존의 블록 인터리버는 메모리에 입/출력이 차례대로 이루어지지만, 제한하는 방식은 메모리에 입/출력이 차례대로 이루어지지 않고 행과 열을 건너뛰는 방식으로 이루어진다. 따라서, 입력되면서 인접데이터간 거리를

증가시키고, 출력되면서 또 한번 인접데이터간 거리를 증가시키게 된다. 다양한 호핑방법 중에서 본 논문에서는 mode-3 방식과 mode-5 방식을 이용하였는데, 이는 각각 3칸씩, 5칸씩 건너뛰면서 입출력하는 방법이다. 호핑방식을 이용한 블록 인터리버의 성능분석을 위하여 이를 SOVA(soft-input/soft-output) 디코딩방식을 이용한 터보코드에 적용하여 성능을 분석한 것이다. 본 논문의 구성은 아래와 같다. 2장에서는 이 논문에 적용된 시스템에 대한 기본지식을 소개하고, 3장에서는 호핑 알고리즘을 적용한 블록 인터리버의 구조와 동작에 대해 설명하였다. 4장에서는 컴퓨터를 이용해 성능분석을 위한 모의실험을 SOVA 복호기를 이용한 터보코드에 호핑 알고리즘을 적용하여 하였으며, 5장에서는 본 논문의 결론을 맺도록 하겠다.

## 2. 적용된 시스템의 소개

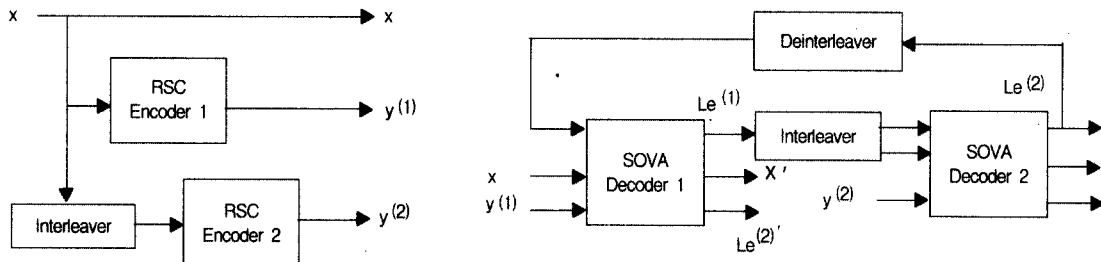
이번 장에서는 호핑방식을 이용한 블록 인터리버의 성능을 알아보기 위해 적용된 SOVA 복호기를 이용한 터보코드 시스템과 기존의 블록 인터리버에 대한 기본적인 설명을 하기로 하겠다.

### 2.1 SOVA 복호화 방식을 이용한 터보코드의 구성

터보코드라는 명칭은 복호 동작원리에서 유래된 말로써 터보엔진의 동작원리와 복호기의 출력을 다시 회귀시켜서 반복적으로 복호를 수행함으로써 그 성능을 향상시킨다. 터보코드는 병렬연접부호방식(Parallel Concatenated Code)으로 2개의 부호기와 이를 구분하는 인터리버로 구성되어지며, 2개의 RSC를 병렬로 연결해 부호화 과정을 수행한다. 또한, 터보코드는 인터리버에 따라 성능이 달라질 수 있기때문에 본 논문에서는 호핑 알고리즘을 적용한 블록 인터리버를 soft-decision 방법을 이용한 SOVA 복호기로 구성된 터보코드에 적용시켜 그 성능을 알아보았다. 그 구성을 그림으로 나타내면 다음과 같다.

### 2.2 블록 인터리버의 특징

인터리버는 입력된 데이터의 거리분포를 개선시켜 전송채널에 의한 연접오류(burst error)를 최소화시키는 방법이다. 블록 인터리버는 입력 데이터를 일정한 저장공간(행의 크기는  $x$ , 열의 크기는  $y$ 인  $x \times y$ 행렬)에 행(x축)방향으로 입력하고 출력할때는 열(y축)방향으로 출력시켜 입력 데이터의 인접 데이터간의 거리를 크게 만드는 방법이다. 여기서  $x$ 와  $y$ 의



(그림 1) 터보코드의 부호기와 복호기

선택에 있어서는 연결에러의 길이가  $x$  또는  $y$ 보다 작다는 가정 하에서  $x$ 와  $y$ 를 선택하여야 한다. 블록 인터리버는 인터리버 중에서 가장 구성하기가 간단한 인터리버이다. 이러한 구조적인 인터리버는 입력 데이터가 작을 경우 유용하고 입력 데이터의 길이가 짧을 경우에는 랜덤 인터리버나  $s$ -랜덤 인터리버가 많이 쓰이고 있으며[12], 또한 헤리컬 인터리버 방법도 복잡한 구조에도 불구하고 유용하게 이용되고 있다.

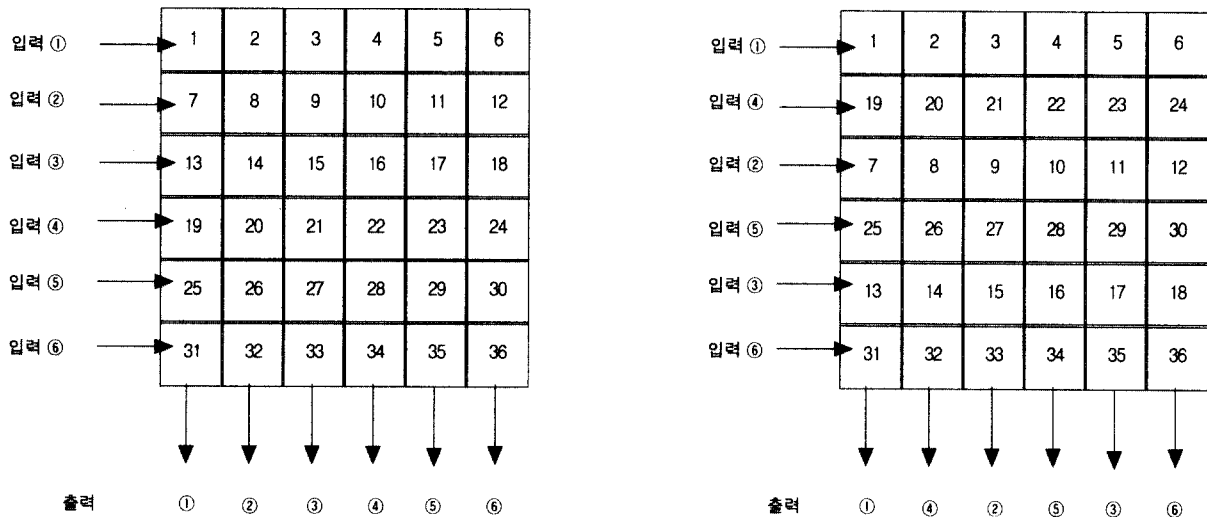
### 3. 호핑 알고리즘

이번 장에서는 호핑방식의 블록 인터리버의 구성과 동작에 대해서 알아보도록 하겠다. 인터리버란 전송하고자 하는 데이터를 적당한 알고리즘에 의해 전송순서를 바꿈으로써 메모리를 가지는 전송채널에 의한 연결오류를 최소화시키는 방식이다. 인터리버의 성능은 인접데이터간 최소거리와 평균거리에 의해 결정되는데, 제안하는 블록 인터리버는 기존의 블록 인터리버보다 인접데이터간 최소거리와 평균거리를 증가시킨다는 장점을 가지고 있다. 호핑방식의 블록 인터리버가 인

접데이터간 최소거리와 평균거리를 증가시키는 이유를 설명하면 다음과 같다. 기존의 블록 인터리버는 메모리 블록에의 입력과 출력이 행과 열에 대해 차례대로 진행되는데 반해 호핑방식의 인터리버는 행과 열을 건너뛰면서 동작을 하게된다. 이를 자세히 설명하면 메모리 블록에 입력되어질때, 1행씩 건너뛰는 방식으로 이루어지고, 출력 역시 1열씩 건너뛰는 방식으로 이루어진다. 따라서, 입력 시 인접 데이터간 거리를 증가시키고, 마찬가지로 출력시에도 인접 데이터간 거리를 증가시켜 준다. 즉, 2번에 걸친 거리의 증가로 기존의 블록 인터리버보다 더 큰 인접데이터간 최소거리와 평균거리를 가지게 된다. 따라서 인터리버 성능에 많은 영향을 준다.

#### 3.1 호핑 알고리즘을 적용한 블록 인터리버의 설계

호핑방식을 이용한 가장 기본적인 mode-2에 대하여 알아보면((그림 2) 참조) mode-2는 메모리 블록에 입력과 출력이 2칸마다 이루어지게 되는데 즉 첫 번째 행의 입력이 끝나면 다음에는 세 번째 행, 다섯 번째 행의 순서로 흡수행에 대해 먼저 입력이 이루어지게되고 다음으로 짝수행에 차례대로 입력



기존의 블록 인터리버

호핑 알고리즘을 적용한 Mode-2 인터리버

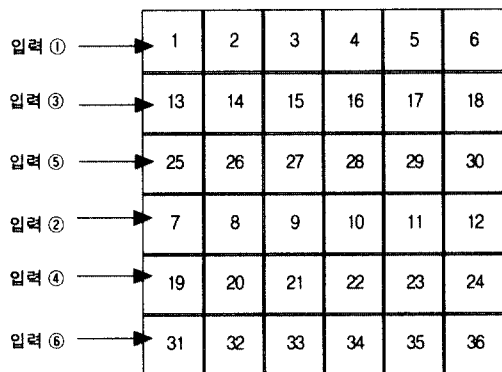
(그림 2) 블록 인터리버와 호핑 알고리즘을 적용 한 블록 인터리버의 비교

이 이루어진다. 출력시에도 입력의 동작과 마찬가지로 홀수행의 데이터를 먼저 출력하고 난후 짝수행의 데이터를 출력하게 된다.

3.2 호핑 알고리즘을 적용한 mode-3 인터리버

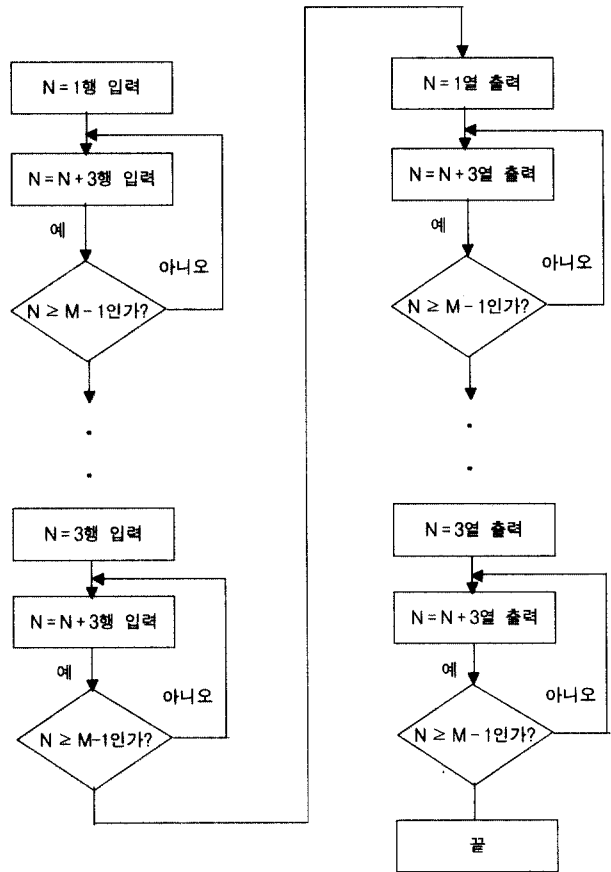
Mode-3 방법은 기본적인 호핑 알고리즘을 적용한 인터리버의 건너뛰는 간격을 조절하는 방식이다. 즉 첫 행을 입력하고 그 다음에 3칸을 건너뛰어서 입력을 하는 방식이다. 출력역시 첫째열을 출력하고 다음으로 3칸 건너뛰어서 출력하는 방식이다. 호핑 알고리즘은 이와같이 블록 인터리버와 같은 구조를 가지면서 다른 알고리즘을 적용하여 데이터간의 거리를 크게하여 여러가지 에러에 대해 효율이 좋도록 구성한 알고리즘이다. 또한 알고리즘이 기존의 블록 인터리버에 비해 복잡해지기 때문에 실제로 하드웨어를 구현할 때는 각각의 행과 열에 대해서 서로다른 지연시간을 할당해야하는 어려움이 따르게 된다. 하지만 이와같은 문제는 랜덤 인터리버나 s-랜덤 인터리버와 같은 구조적으로도 복잡하고 알고리즘도 복잡한 인터리버를 만드는 것에 비해서는 훨씬 효율적이라고 할 수 있다. (그림 3)은 mode-3 인터리버의 그림이다. 입력데이터가 6x6 형태를 가지는 메모리 저장공간에 입력시 3칸의 건너뛰는 동작을 수행하면서 입력되어지고 출력시에는 3칸을 건너뛰면서 출력하는 동작을 한다.

Mode-3 인터리버는 첫 번째 행의 입력이 끝나면 다음에는 네 번째 행의 순으로 입력이 이루어지고, 출력 역시 첫 번째, 네번째 열의 순으로 이루어지게 된다. 이러한 호핑 알고리즘은 데이터의 크기가 커짐에 따라서 호핑간격을 크게해서 멀티미디어 시스템과 같이 고용량의 데이터를 가지는 시스템에 적용하면 효율적인 동작을 수행할 수 있을 것이다. 즉 mode-n으로 동작하는 방식의 호핑방식의 블록 인터리버는 메모리 블록에서 건너뛰는 칸 수가 n행, n열 단위로 이루어지는 것



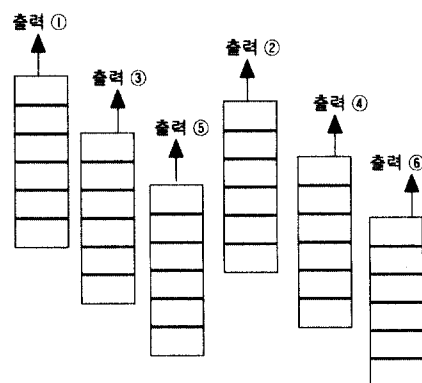
(그림 3) Mode-3 인터리버(6x6)

이다[13].

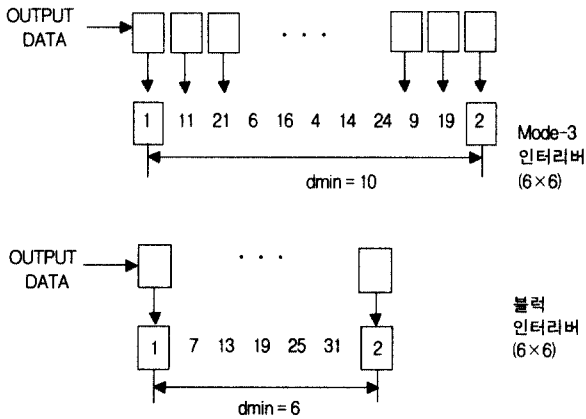


(그림 4) Mode-3의 순서도

(그림 4)는 mode-3의 동작에 대한 순서도를 나타내었는데, 호핑방식이 mod-4 또는 mod-5등으로 바뀌게 되면 입/출력 때의 초기값 N = 1, 2, 3 뿐만아니라 N = 1, 2, 3, 4 또는 N = 1, 2, 3, 4, 5로 설정되고, 증가값 N = N + 3가 아니고 N = N + 4 또는 N = N + 5 등으로 바뀌어야 한다. 즉, mode의 변화가 있을 때 마다 순서도에 변화를 가져오게 되지만 단순한 과정을 거



처 각각의 mode에 적당한 순서도 작성이 가능하게 된다.



(그림 5) 인접 데이터간의 거리

(그림 5)는 블록 인터리버와 mode-3 인터리버의 인접데이터간의 거리를 나타내고 있다. 이 때의 인접데이터간의 거리는 블록 인터리버와 비교해 더 커짐을 알 수 있다. 같은 크기의 메모리 블록이라고 하더라도 호핑 mode에 따라서 서로 다른 출력을 시퀀스를 가지며 인접데이터간의 최소거리 또한 서로 다르게 되므로, 다른 성능을 가지는 인터리버의 구성이 가능하게 된다.

### 3.3 호핑 알고리즘을 적용한 mode-5 인터리버

호핑 알고리즘을 적용한 블록 인터리버 mode-5는 mode-3과 같은 형태의 인터리버이지만 건너뛰는 간격이 3칸이 아닌 5칸으로 늘어나게 된다. 즉 입력 시퀀스를 메모리 블록에 입력할때 첫째행에 입력하고 다음에 입력할 때는 5칸 떨어진 여섯 번째행에 입력한다. 출력도 역시 1번째 열을 먼저 출력하고 다음으로 여섯칸씩 떨어진 열을 출력하는 방식이다. 그리고 이때 주의해야 할 점은 인터리버를 구성할 때 메모리의 저장공간이 최소한 건너뛰는 간격 이상이 되어야 효율적으로 인터리빙 할 수 있다. 만약 mode-5의 인터리버에 메모리 저장 공간의 구성이 4x4라고 한다면 효율적인 인터리빙을 기대하기는 어렵다. 멀티미디어 데이터와 같이 데이터량이 많은 경우에는 문제가 되지않지만 데이터량이 작을때는 주의하여야 한다. 호핑 알고리즘을 적용한 블록 인터리버는 콘벌루션 인터리버나 랜덤, s-랜덤 인터리버에 비해 비교적 간단한 구조를 가지면서도 기존의 블록 인터리버보다 큰 인접데이터간의 최소거리와 평균거리를 가진다. 그리고 기본적인 호핑 방식인 mode-2을 바탕으로 하여 mode-3, mode-4, mode-5 ... 등과 같은 다양한 호핑 알고리즘의 적용이 가능하다. 블록

의 크기가 증가할수록 기존의 블록 인터리버와 비교하여 인접 데이터간의 최소거리와 평균거리의 차이가 증가한다. 하지만 메모리 블록에서 행과 열에 대해 서로다른 지연시간을 두어야 하므로 기존의 블록 인터리버에 비해 복잡한 알고리즘을 필요로하며 실제로 하드웨어를 만들때 어려움이 따르게 된다. 다음 장에서는 알고리즘을 바탕으로 한 컴퓨터 시뮬레이션을 통해 이 장의 설명을 하기로 하겠다.

## 4. 모의실험

이번 장에서는 기존의 블록 인터리버와 호핑 알고리즘을 적용한 mode-3 인터리버, mode-5 인터리버를 SOVA 디코더를 이용한 터보코드에 접목하여 모의실험을 하였다. 이 때 데이터의 길이  $N = 10000$ ,  $N = 40000$ ,  $N = 90000$ 개로 나누어 실험하였고 채널 환경은 가우시안 노이즈 환경을 이용하였다.

(그림 6) 콘벌루션 부호에 적용한 Mode-2, Mode-3

(그림 6)은 이전의 논문인 호핑방식을 이용한 블록 인터리버의 성능분석에서 호핑 알고리즘을 적용한 블록 인터리버 mode-2, mode-3을 채널부호화율이 1/2인 콘벌루션 부호기에 적용하였을때의 결과를 나타낸 것이다. 기존의 블록 인터리버와 호핑방식의 블록 인터리버의 성능을 비교해보면 호핑방식을 이용한 mode-2, mode-3 인터리버가 SNR(signal to noise ratio)이 7dB가 넘어서면서 기존의 블록 인터리버에 비해 좋은 성능을 나타내는데 BER(bit error rate)이  $10^{-5}$  정도에서 약 1dB 정도의 부호화 이득을 얻는다. 그리고 SNR이 8dB 이상에서는 mode-3가 mode-2에 비해 우수한 성능을 나타낸다. 이는 호핑방식의 인터리버의 mode가 커질수록 인접 데이터간의 최소거리가 크게 나타나기 때문이다. (그림 7), (그

림 8), (그림 9)는 각각 입력데이터의 길이에따른 블록 인터리버와 호핑 알고리즘을 적용한 mode-3 인터리버, mode-5 인터리버의 BER 성능을 SOVA 디코딩방식을 이용한 터보코드에 적용하여 나타낸 것이다.

(그림 7) N=10000 (100×100) 일때의 결과

(그림 7)은 호핑 알고리즘을 SOVA 방식을 이용한 터보코드에 적용했을때 입력데이터의 길이가 10000개일때의 BER 성능그래프이다. 블록 인터리버의 성능이 제일 낮고 호핑 알고리즘을 적용한 mode-3 인터리버와 mode-5 인터리버는 7dB 전에는 거의 비슷한 성능을 나타내고 있다. 블록 인터리버와 호핑방식의 블록 인터리버의 성능을 비교해보면 호핑방식을 사용한 인터리버의 성능이 신호 대 잡음비가 7dB를 넘어 서면서 우수한 결과를 나타내었고 비트 오차율이  $10^{-4}$  정도에서 1dB 정도의 이득을 얻을 수 있다. 그리고 mode-3과 mode-5는 7dB이상에서 mode-5가 0.2dB 정도 우수한 성능을 보이고 있다. 이는 mode-5 인터리버의 인접데이터 간의

(그림 8) N=40000 (200×200) 일때의 결과

거리가 mode-3 인터리버 보다 크게 나타나기 때문이다.

(그림 8)은 입력데이터의 길이가 40000개 일때의 BER 성능 그래프이다. 블록 인터리버의 성능이 역시 제일 낮고 호핑 알고리즘을 적용한 mode-3 인터리버와 mode-5 인터리버는 신호 대 잡음비가 6dB 이전에는 거의 비슷한 성능을 가진다. 신호 대 잡음비가 6dB를 넘어서면서 호핑 알고리즘을 적용한 블록 인터리버가 기존의 블록 인터리버에 비해 개선된 성능을 보이는데 비트 오차율이  $10^{-5}$  정도에서 0.6dB 정도의 이득을 얻는다. 그리고 mode-3과 mode-5는 7dB이상에서 mode-5가 0.3dB 정도 우수한 성능을 보이고 있다.

(그림 9) N=90000 (300×300) 일때의 결과

(그림 9)는 입력데이터의 길이가 90000개 일때의 BER 성능 그래프이다. 호핑방식을 사용한 인터리버의 성능이 신호 대 잡음비가 6dB를 넘어서면서 기존의 블록 인터리버에 비해 우수한 결과를 나타내고 비트 오차율이  $10^{-5}$  정도에서 0.7dB 정도의 이득을 얻을 수 있다. 그리고 mode-5는 mode-3에 비해 6dB를 넘어서면서 점점 향상된 성능을 나타내다가 BER이  $10^{-6}$  정도에서 0.2dB정도 우수한 성능을 보이고 있다.

위의 시뮬레이션 결과에서 SOVA 방식을 이용한 터보코드에서 기존의 블록 인터리버에 비해서 호핑 알고리즘을 적용한 mode-3 인터리버와 mode-5 인터리버의 성능이 SNR이 약 6dB를 넘어서면서 향상된 성능을 나타내기 시작하였고 BER이  $10^{-5}$  정도에서 0.7dB~1dB 정도의 이득을 나타낸다. mode-5 인터리버는 mode-3 인터리버에 비해 약간의 우수한 성능을 보여주고 있다. 이는 호핑방식의 모드에 따라서 인접데이터간의 최소거리가 달라지므로 mode-3과 mode-5의 차이가 나타나게 되는데 mode-5의 비트거리가 mode-3보다 더 크기 때문이다. 터보코드에서도 블록 인터리버에 비해 호핑 알고리즘을 이용한 블록 인터리버의 성능이 나아지는 것을 알 수

가 있다. 그리고 mode의 간격을 크게한다고 좋은 성능을 나타내는 것이 아니라 디코더의 성능이나 메모리 블록의 길이에 따른 간격조절이 필요하다.

버와 mode-5 인터리버에 대한 특징을 살펴보고 컴퓨터 시뮬레이션을 통해 성능을 알아보았다. 블록 인터리버에 비해서 호핑 알고리즘을 적용한 블록 인터리버의 성능이 신호 대 잡음비 6dB 이상에서 우수한 성능을 보였고, 비트 오차율이  $10^{-5}$ 를 넘어서면서 1dB 정도의 개선된 부호화 이득을 나타낸다. 따라서 호핑 알고리즘을 적용한 인터리버를 IMT-2000의 음성채널이나 멀티미디어 서비스 시스템에 적용한다면 기존의 블록 인터리버에 비해 우수한 채널 부호화 효과를 얻을 수 있을 것이며 인터리버를 이용하는 많은 다른 시스템에 적용이 가능할 것이다. 향후 과제로는 최적의 호핑방식을 적용한 블록 인터리버 알고리즘을 위해 다양한 방식의 모드에 대한 연구와 메모리 크기에 따른 최적화 방식의 연구에 이러한 인터리버를 무선 랜 시스템이나 인터리버를 사용하는 여러 가지 시스템에 적용하여 성능을 검증하는 것이다.

## 참 고 문 헌

- [1] Perez, L. C., Seghers, J. and Costello, D. L., "A Distance Spectrum Interpretation of Turbo Codes," IEEE Transactions on Information Theory, Vol.IT-42, No.6, pp.1698-1709, 1996.
- [2] Andersen, J. D. and Zyablov, V. V., "Interleaver Design for Turbo Coding," Proceedings of International Symposium on Turbo Codes, pp.154-156, 1997
- [3] Jung, P., "Comparison of Turbo-Code Decoders Applied to Short Frame Transmission Systems," IEEE Journal on Selected Areas in Communication, Vol.14, No.3, pp.530-537, 1996.
- [4] Pyndiah, R., Picart, A., and Glavieux, A., "Performance of Block Turbo Coded 16-QAM and 64-QAM Modulation," Globecom, 95, pp.1039-1043, 1995.
- [5] Benedetto, S. and Pollara, F., "Parallel Concatenated Trellis Coded Modulation," Proceedings of International Conference on Communication 1996, pp.974-978, 1996.
- [6] Raphaeli, D. and Zarai, Y., "Combined Turbo Equalization and Turbo Decoding," Proceedings of International Symposium on Turbo Codes, pp.180-183, 1997.
- [7] Reed, M. C. and Asenstrofer, J., "Iterative Multi-User Detection for DS-CDMA with FEC," Proceedings of International Symposium on Turbo Codes, pp.162-165, 1997.
- [8] Bernard Sklar, "DIGITAL COMMUNICATIONS," Second Ed. Prentice Hall, pp475-510, 2001.
- [9] John G. Proakis, "DIGITAL COMMUNICATIONS," Third

(그림 10) 입력데이터의 길이(N)에 따른 Mode-3, Mode-5 결과 비교

## 5. 결 론

터보코드에서 부호화한 비트들의 상관성이 증가할수록 복호화 동작을 위한 많은 정보를 얻을 수는 있겠지만 예러들에 대한 상관성이 증가하게 된다. 따라서 예러들 사이에 상관성을 제거하기 위해서 인터리버를 사용하며 인터리버의 크기나 구조에 따라 복잡성이나 지연의 문제도 수반하게 된다. 본 논문에서는 터보코드에서 가장 기본이 되는 블록 인터리버에 대한 성능과 비교해 호핑 알고리즘을 적용한 mode-3 인터리

Ed. McGraw-Hill, 1995.

- [10] Raymond Steele, "MOBILE RADIO COMMUNICATIONS," IEEE Press, 1994.
- [11] Danny T. Chi, "A NEW HELICAL INTERLEAVER," MILCOM'92, Vol.2, pp.799-804.
- [12] STEPHENG. WILSON, "DIGITAL MODULATION AND CODING," Prentice Hall, 1996.
- [13] Hyung-Yun Kong, Il-Seung Woo, "Performance Analysis of Block Interleaver by Using Hopping Method," The Institute of Electronics Engineers of Korea, May, 2001.

## 공 헝 궤

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

1989년 NYIT 전자공학과(학사)

1991년 Polytechnic University 전자공학과  
(석사)

1996년 Polytechnic University 전자공학과  
무선통신 전공(박사)

1998년~현재 울산대학교 전기 전자 정보시스템공학부 교수

2001년~현재 ETRI 초빙 연구원

관심분야 : SDR, STTC, STDC, STBC, 변복조 및 코딩 기술