영상통신에서 잡음 제거를 위한 새로운 세미 랜덤 인터리버 알고리즘

홍 성 원† · 박 진 수††

요 약

본 논문에서는 영상 통신 채널 상에서 발생하는 잡음을 효과적으로 제거하기 위해 터보코드를 사용하였다. 터보코드는 복호 성능이 우수하지만 시스템의 복잡도와 복호 과정의 시간지연 때문에 실시간 통신에는 부적합하다는 단점이 있다. 이 문제를 극복하기 위해, 본 논문에서는 터보코드의 부·복호기에 사용되는 인터리버의 크기를 감소시켜 영상 데이터를 전송할 때 소요되는 시간지연을 줄이는 새로운 세미 랜덤(Semi-Random) 인터리버 알고리즘을 제안하였다. 세미 랜덤 인터리버 알고리즘은 입력 프레임의 길이를 1/2 크기만큼 인터리버를 구성하고, 인터리버 내에 데이터를 입력할 때는 블록 인터리버처럼 행으로 입력하며, 데이터를 읽을 때는 랜덤하게 읽음과 동시에 다음 데이터가 그 주소 번지에 위치하게 된다. 그러므로, 기존의 블록, 대각, 랜덤 인터리버와 알고리즘의 복잡도를 비교할 시 그 복잡도가 1/2로 감소되어 세미 랜덤 인터리버를 터보코드에 적용할 때 영상 데이터를 실시간 처리할 수 있다.

A New Semi-Random Interleaver Algorithm for the Noise Removal in Image Communication

Sung-Won Hong† · Jin-Soo Park††

ABSTRACT

In this paper, The turbo code is used to effectively remove noise which is generated on the image communication channel. Turbo code had excellent decoding performance. However, it had limitations for real time communication because of the system complexity and time delay in decoding procedure. To overcome this problem, this paper proposed a new SRI(Semi Random Interleaver) algorithm, which decrease the time delay, when the image data, which reduced the interleaver size of turbo code encoder and decoder, transmitted. The SRI algorithm was composed of 0.5 interleaver size from input frame sequence. When the data inputs in interleaver, the data recorded by row such as block interleaver. But, When the data read in interleaver, the data was read by randomly and the next data located by the just address simultaneously. Therefore, the SRI reduced half-complexity when it was compared with pre-existing method such as block, helical, random interleaver. The image data could be the real time processing when the SRI applied to turbo code.

1. 서 론

기존의 통신이 음성 서비스를 주축으로 영상과 데이터 서비스가 부가적인 것이라면 앞으로의 통신은 영상 서비스를 주축으로 발전해 나갈 전망이다. 예로서 IMT-2000은 영상 서비스를 중심으로 전세계 어느 곳에서나 하나의 단말기로 통신이 가능하도록 하고자하는 시스템이다[1]. 즉, IMT-2000은 전 세계 어느 지역에서나 하나의 단말기로 통신 가능한 단말 이동성(Terminal Mobility)과 개인이 어느 상황에 있든 자신

※ 본 연구는 과학기술부·한국과학재단 지정 청주대학교 정보통신연구센터의 지원에 의한 것입니다.
† 정 회 원 : 모아통신(주) 전임연구원
†† 정 회 원 : 청주대학교 첨단공학부 교수
논문접수 : 2000년 5월 9일, 심사완료 : 2000년 8월 11일

의 고유번호로 통신 가능한 개인 이동성(Personal Mobility)을 제공하여야 하고 가능한 모든 서비스를 모든 환경에서 제공하자는 것이며 그 주된 서비스가 바로 영상이다. 그러나 이같이 차세대 통신 시스템에서 가장 중히 여기는 영상 서비스는 그 데이터 양의 방대함으로 인해 실현에 많은 난제가 있었다. 이를 위해 데이터의 양을 줄이는 영상 압축에 대한 연구가 지속적으로 수행되어 왔다[2-4]. 그러나 효과적으로 영상 압축이 수행되었다 하더라도 채널 상에서 발생하는 잡음은 제거되지 않는다. 따라서 고화질의 영상 통신이 행해지기 위해서는 채널 상에서 발생하는 잡음을 효과적으로 제거시켜 주어야만 한다. 이를 위해 본 논문에서는 채널 상에서 발생하는 잡음을 효과적으로 제거할 수 있는 방법을 제안하고자 하며 이를 터보코드를 통해 구현하고자 한다.

터보코드는 1993년 C. Berrou가 ICC(International Conference on Communications)에서 "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo Codes"라는 논문을 발표하면서 소개되었다[5].

터보코드는 소개되면서부터 우수한 복호 성능으로 많은 분야에서 연구되고 있다. 특히 위성통신은 지구상공 35,860[㎞]에서 지구국과 통신을 하기 때문에 채널상의 대기 잡음과 강우 감쇠로 인한 신호의 왜곡이 발생한다. 이를 극복하기 위해 복호 성능이 우수한 터보코드를 오류정정코드로 사용할 수 있는 기법의 연구가 활발히 진행되고 있다[6]. 또한 이동통신은 채널상의 오류를 정정하기 위해 부호화율이 1/2과 1/3인 길쌈부호를 사용하고 있다. 그러나 길쌈부호는 오류 확률에 따른 구속장이 증가하면 시스템의 복잡성이 지수적으로 증가하기 때문에 $K=9$이상의 경우 구현이 어렵고 비실용적이다[7, 8]. 따라서 구속장이 짧으면서도 복호성능이 우수한 터보코드를 PCS와 CDMA 시스템에 적용하기 위해 많은 연구가 이루어지고 있다[9-11].

터보코드의 구조를 살펴보면, 터보코드 부호기는 두 개의 RSC(Recursive Systematic Convolutional) 부호와 인터리버(interleaver)로 구성되어 있다. RSC 부호는 조직 길쌈부호(systematic convolutional code)에 궤환이 더해진 형태이다. 터보코드 복호기는 부호기의 두 RSC 부호에 대한 복호기가 직렬로 연결되어 있으며, 각 복호기에서는 MAP 복호 알고리즘[12-14] 또는 SOVA(Soft-Output Viterbi Algorithm)[15-17]를 이용하여 복호를 수행한다. 터보코드가 우수한 성능을 내는 가장 큰 이유는 복호기에 내재하는 인터리버와 복호기의 반복 복호 때문이다. 그러나, 인터리버의 크기가 클수록, 반복 복호 횟수가 많을수록 터보코드의 복호성능은 우수하지만 시스템이 복잡해져 한 개의 정보비트를 복호할 때 많은 시간지연을 발생시켜 실시간 통신에는 부적합하다.

그러므로 터보코드의 복잡도를 줄이기 위해 Franz V.는 트렐리스 경로에 임의의 임계값을 주어 각 경로의 메트릭 값이 임계값 이상이면 생존경로로 선택하고 임계값 이하면 경로를 삭제하여 터보코드의 복잡도를 감소시킬 수 있는 알고리즘을 제시하였다[18]. 또한 S. Hong은 터보코드를 VLSI로 설계시 복잡도를 감소시키는 방법을 연구하는 등[19], 복잡도를 감소시키기 위한 많은 연구가 진행 중에 있다[20, 21].

터보코드의 중요한 성능 파라메타인 인터리버는 페이딩 채널에서 발생하는 연집 오류(burst error)에 대비하기 위하여 사용하는 시간 다이버시티의 형태로 부호어를 분산시켜서 비트와 비트가 서로 독립적으로 페이딩이 되도록 하는 것이다[22, 23]. 이 경우 연집 오류는 다수의 부호어에 속하는 다수의 비트군에 영향을 미친다. 부호화된 메시지를 전송하기 전 인터리빙을 하고, 메시지를 수신한 후에 역 인터리빙을 행하는데 채널에 오류가 연집으로 발생할 경우 시간적으로 연집 오류를 확산시켜 복호기에서는 랜덤 오류처럼 되어 오류정정이 가능하다. 인터리버는 메모리를 사용하여 구현할 수 있으며, 무선 페이딩 채널하에서 디지털 신호를 전송할 때 그 품질을 향상시킬 수 있다. 그러나 부가적으로 시간지연과 메모리의 공간을 더 많이 요구하며, 시스템의 복잡도가 높아지는 단점이 있다.

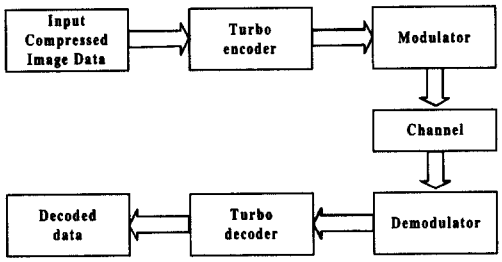
본 논문에서는 한 개의 정보비트를 복호할 때 소요되는 시간지연을 줄이기 위한 새로운 세미 랜덤(Semi-Random) 인터리버 알고리즘을 제안하였다. 세미 랜덤 인터리버 알고리즘은 입력 데이터 길이의 1/2 크기 만큼 인터리버를 구성하고, 인터리버 내에 쓸때는 블록 인터리버와 동일하게 행으로 쓰고, 읽을 때는 랜덤하게 읽음과 동시에 다음 데이터가 그 주소 번지에 위치하게 된다. 따라서 기존의 블록, 대각, 랜덤 인터리버 알고리즘의 복잡도를 비교할 때 세미 랜덤 인터리버의 복잡도가 1/2로 감소됨을 입증하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 본 논문에서 구현하고자 하는 시스템의 구성에 대하여, 제3장에서는 영상 압축기법에 대하여 기술하였다. 그리고 제4장에서는 터보코드 부호기, 복호기 구조와 기존의

인터리버 및 본 논문에서 제안한 새로운 세미 랜덤 인터리버 알고리즘에 대하여 기술하였다. 제5장에서는 기존의 블록, 대각, 랜덤 인터리버와 세미 랜덤 인터리버의 특성을 비교하고, 제6장에서는 시뮬레이션 결과 및 고찰에 대하여 기술하였다. 마지막으로 제7장에서는 결론을 맺었다.

2. 시스템의 구성

(그림 1)은 본 논문에서 구축하고자 하는 시스템의 구성도를 나타내었다. 우선 터보코드 부호기의 입력은 영상데이터를 이용하게 되는데 이 영상 데이터는 입력되기 전에 영상 압축 단계를 거쳐 고효율로 압축된 영상 데이터 계열로 출력된 것이다. 이후 부호기를 거쳐 부호화된 데이터 즉, 부호어는 변조기에 입력되어 변조된 후 채널을 통하여 전송되게 된다.



(그림 4) 본 시스템의 구성

최종적으로 채널을 통하여 수신된 수신계열은 복호기에 입력되어 복조된 후 터보코드 복호기에 의하여 오류정정이 되어 복호 데이터가 출력된다. 구축하고자 하는 시스템은 복잡도의 최소화를 위하여 간단한 부호기를 이용하며 복호는 MAP알고리즘을 이용한다.

3. 영상 압축

영상은 512 × 512 크기의 흑백 영상의 경우만 하더라도 초당 약 8M 바이트 정도의 정보를 처리해야 하기 때문에 데이터의 저장 및 전송이라는 측면에 있어서 많은 어려움이 따른다. 따라서 영상 통신에 있어 압축에 대한 기술 연구는 매우 중요한 위치를 차지하고 있으며 이를 위해 이진 영상 부호화, DPCM, Huffman 코딩, DCT를 이용한 압축 방법들이 발전되어 왔다. 그러나 80년대 이후 시작된 차세대 압축 부호화 기법은 기존의 DCT 기반 부호화 기법의 한계를 극복하고 인간 시각계에 더욱 적합한 압축 기법에 관한 연구를 수행하는 것으로 모델 기반 부호화 기법, 객체 지향 부호화 기법, 세그먼트 기반 부호화 기법, Wavelet 코딩기법, Fractal 부호화 기법등이 제안되었다. 본 시스템에서는 이중 Wavelet 코딩 기법을 이용하여 기존의 DCT 기법의 경우보다 훨씬 압축율이 높은 영상 코딩 기법을 연구중이며, 이의 최종 목표는 터보코드의 적용이 가능할 정도의 압축율 실현에 있다.

4. 터보코드를 이용한 채널 코딩

터보코드는 오류정정 능력이 우수하여 통신채널 상에서 일어나는 오류를 복호하는 오류정정 부호기로서 많은 연구가 진행되어지고 있으며 그 구성은 다음과 같이 3부분으로 구분 지을 수 있다. 첫째 연집오류를 산발오류로 분포시킬 수 있는 인터리빙단이며, 둘째 정보비트를 부호화하는 부호기단, 마지막으로 부호화된 정보비트를 복호화하는 복호기단으로 구성된다.

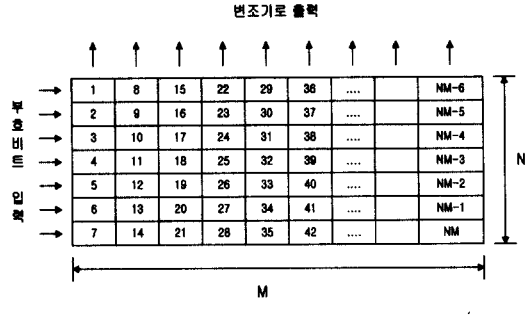
4.1 인터리빙

인터리버의 사용은 잡음으로 인한 신호의 훼손을 통계적 독립으로 만드는 효과적인 방법으로 기존의 블록, 대각, 랜덤 인터리버과 본 논문에서 제안한 새로운 세미 랜덤 인터리버를 기술하였다.

4.1.1 블록 인터리빙

잘 알려진 인터리빙 방법으로 블록 인터리빙은 매우 간단한 인터리빙 방법이며 <표 1>에 나타내었다. 입력 비트의 계열을 $N \times M$ 행렬로 본다면, 정보 비트의 배열을 열로 데이터를 쓰고, 행으로 데이터를 읽어서 변

〈표 1〉 블록 인터리버

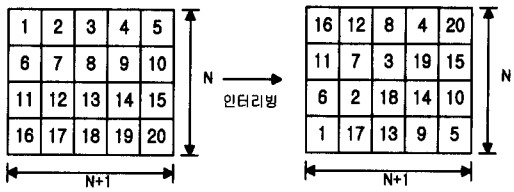


조기로 출력하는 방식이다. 이 인터리빙 방법의 가장 큰 장점은 구조가 간단하다는 점이다. 그러나 전체 프레임에 걸쳐서 발생하는 연집 오류에 대하여 여전히 상관이 존재하게 되는 단점이 있다. 블록 인터리빙을 사용하면 채널에서의 연집 오류는 M만큼 분산되어 나타나게 된다.

4.1.2 대각 인터리빙

<표 2>와 같이 대각 인터리빙 역시 간단한 인터리빙 방법이며 입력 정보를 $N\times(N+1)$ 행렬로 보고 행으로 데이터를 읽고 대각으로 데이터를 쓰는 방식이다. 대각 인터리빙을 사용할 경우 연집 오류는 $N+1$만큼 분산되어 나타나게 된다.

〈표 2〉 대각 인터리버



4.1.3 랜덤 인터리빙

터보코드에서 우수한 성능을 나타내는 인터리빙 방법으로 알려진 인터리빙 방식으로서 입력된 정보 비트를 랜덤하게 재배열하며, 출력으로 내보내기 위한 정보 또한 규정된 난수체계를 이용하여 난수를 발생시킴으로써 랜덤하게 출력한다.

랜덤 인터리버의 lookup-table을 만드는 알고리즘은 다음과 같다.

단계1. 크기가 2^M인 배열 A를 0으로 초기화하고, $m=0$로 둔다.

단계2. $m\le 2^{M-1}$에 대하여 0과 2^{M-1}사이의 난수 p를 발생한다.

단계3. 만약 $A[p]=0$이면 $A[p]$를 1로 두고 배열 B의 m위치에 p를 저장한다. 즉 $B[m]=p, m=m+1$, 단계2로 간다.

배열 B는 소스와 목적지의 관계를 결정하는데, 만약 $B[m]=p$이면 주소 m에 위치한 비트는 주소 p로 보내진다.

4.1.4 새로운 Semi-Random 인터리버 알고리즘 제안

기존의 블록, 대각 인터리버는 입력 비트의 수가 S라고 하면, 인터리버의 크기는 $S=N\times M$으로 구성되어 데이터를 일정한 규칙에 의해 읽고 쓴다. 그리고 데이터를 랜덤하게 읽고 쓰는 랜덤 인터리버의 크기 또한 $S=N\times M$이다.

본 논문에서 제안한 세미 랜덤 인터리버 알고리즘은 블록, 대각 인터리버처럼 데이터를 인터리버 내에 쓸 때는 규칙성을 갖고 데이터를 쓰고, 데이터를 읽을 때는 랜덤 인터리버처럼 데이터를 랜덤하게 읽는다. 제안한 세미 랜덤 인터리버의 알고리즘은 다음과 같다. 입력 비트의 수가 S라고 하면 인터리버의 크기는 $S/2$로서 인터리버를 구성하고, 순차적으로 데이터를 메모리 내에 행으로 저장한다. 메모리 내에 데이터가 모두 저장되면 랜덤하게 난수를 발생하여 데이터를 읽는다. 이때 메모리 내의 데이터를 읽음과 동시에 $S/2+1$번째 데이터가 그 위치에 입력된다. 이렇게 해서 메모리 내의 모든 데이터를 읽는다. 단, 메모리 내의 모든 데이터를 읽을 때 동일한 데이터의 번지 값은 모든 메모리의 번지 값이 한번씩 읽기 전에는 두 번 중복해서 읽지 않는다.

본 논문에서 제안하는 알고리즘을 예를 들어서 설명하도록 해보자. 입력 비트의 값 $S=16$이라고 하면, 1, 2, 3, 4, 5, ..., 14, 15, 16 로서 인터리버의 크기는 2×4로 구성되어 메모리의 수는 8개가된다. 입력되는 비트는 행으로 순차적으로 아래 <표 3> 처럼 쓰여진다.

〈표 3〉 인터리버 내에 데이터를 입력할 때

1	2	3	4
5	6	7	8

첫번째에서 $S/2$번째인 8번째 데이터까지는 가로로 먼저 메모리 내에 쓰여지며, 읽을 때는 랜덤하게 난수를 발생하여 읽는다. 그 과정을 살펴보면 <표 4>와 같다. 만약 5번째 주소번지의 데이터를 읽고 전송되면 $S/2+1$번째인 9번째 데이터가 그 빈 주소 번지에 위치하게 되며, 다음으로 2번째 주소번지의 데이터를 읽고 전송되면 $S/2+2$번째인 10번째 데이터가 그 빈 주소 번지에 위치하게 된다. 그리고 7번째 주소번지의 데이터를 읽고 전송되면 $S/2+3$번째인 11번째 데이터가 빈 주소 번지에 위치하게 된다. 다음으로 6번째 주소번지를 읽고 전송되면 $S/2+4$번째인 12번째 데이터

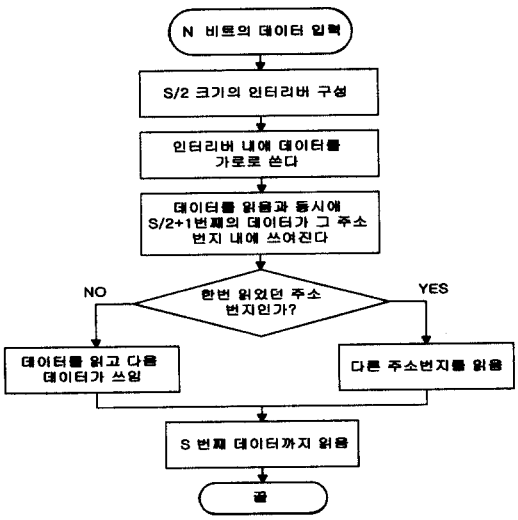
가 빈 주소 번지에 위치하며, 1번째 주소번지를 읽고 전송되면 $S/2+5$번째인 13번째 데이터가 빈 주소 번지에 위치 위치하고, 4번째 주소번지를 읽고 전송되면 $S/2+6$번째인 14번째 데이터가 그 빈 주소 번지에 위치하게 된다. 그리고 3번째 주소번지의 데이터를 읽고 전송되면 $S/2+7$번째인 15번째 데이터가 그 빈 주소 번지에 위치하고, 8번째 데이터를 읽고 전송되면 $S/2+8$번째인 16번째 데이터가 그 빈 주소 번지에 위치함으로 데이터를 인터리버 내에 쓰고 읽는 과정을 반복한다.

〈표 4〉 인터리버 내에서 데이터를 출력하는 과정

1(13)	2(10)	3(15)	4(14)
5(9)	6(12)	7(11)	8(16)

이런 순서로 1에서 16번째 비트를 한번씩 메모리 내에 있는 모든 데이터를 읽는다. 위의 방법으로 읽은 데이터는 변조기로 전송되며 이때 전송되는 비트의 순서는 5, 2, 7, 6, 1, 4, 3, 8이고 두 번째에도 랜덤하게 데이터를 읽어서 전송된다. 이때의 조건은 메모리 내의 주소 번지 값을 한번씩 모두 읽기 전에는 반복해서 읽지 않는다.

변조기에 의해 변조된 신호는 채널을 통하여 전송되고 송신기에서는 위의 신호를 수신하여 복조기를 통하여 복조한 후 역 인터리버를 통하여 원래의 데이터의 순서인 1, 2, 3,.....15, 16으로 복원한다.

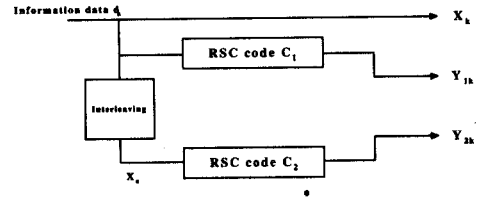


(그림 2) 제안한 알고리즘 흐름도

역 인터리버를 시행할 때 인터리버에서 인터리빙된 데이터는 메모리 내의 주소 번지 값을 기억하고 있어야 한다. 이상과 같은 제안한 세미 랜덤 인터리버 알고리즘을 흐름도로 나타내면 (그림 2)와 같다.

4.2 터보코드 부호기

터보코드의 부호기는 (그림 3)과 같이 두 개의 RSC (Recursive Systematic Convolutional Code)와 인터리버의 결합으로 구성되어있다.



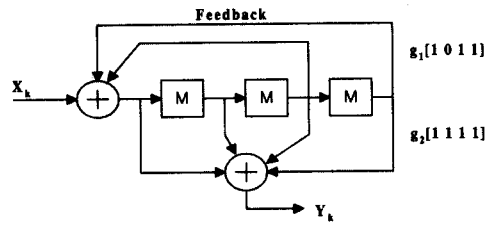
(그림 3) 터보코드 부호기

위의 그림에서 부호화 율이 1/3, 구속장의 길이가 K인 길쌈 부호기의 k번째 부호기 입력이 d_k 비트 일 때, 출력 (X_k, Y_k)는 다음과 같이 나타낼 수 있다.

$$X_k = \sum_{i=0}^{v} g_{1i} d_{k-1} \qquad d_{1i}=0,1$$
$$Y_k = \sum_{i=0}^{v} g_{2i} d_{k-1} \qquad d_{2i}=0,1 \tag{1}$$

여기서 $G_1 : \{g_{1i}\}$, $G_2 : \{g_{2i}\}$는 두 개의 부호기의 생성 계열이다. 터보코드 부호기의 구성부호는 전송률 및 오류정정 능력을 고려하여 원하는 만큼 구성할 수 있다.

두 개의 RSC 코드는 조직 길쌈코드에 궤환이 이루어진 형태라고 할 수 있는데 그 실제적인 예로 (그림 4)와 같은 형태 외에 다양하게 구성할 수 있다. RSC 코드의 구조가 터보코드의 성능 전체에 미치는 영향이



(그림 4) RSC 부호기

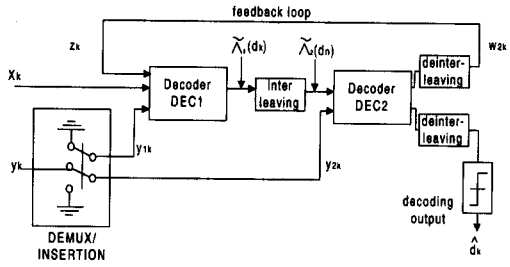
상당한 비중을 차지하며 다양한 구성부호에 대하여도 터보코드의 성능은 차이를 나타낸다.

이제 부호화 과정을 살펴봐야 하는데 이의 과정은 다음과 같다.

k번째 정보비트 d_k는 채널을 통하여 출력으로 직접 전송되는 동시에 첫 번째 RSC 부호기에 입력이 되어 출력 Y_{1k}를 생성하고, 또 두 개의 RSC 부호기 사이에 있는 인터리버에 의해 인터리빙된 후 두 번째 RSC 부호기에 입력이 되어 출력 Y_{2k}가 생성되어 다음 통신블록으로 전송되어 진다.

4.3 터보코드 복호기

(그림 5)는 일반적인 터보코드의 복호기를 나타내는데, 부호기에서의 두 RSC 부호에 대한 복호기, DEC1과 DEC2가 직렬로 연결되어 있다. DEC1은 정보비트 x_k와 첫번째 RSC 부호의 출력 y_{1k}을 입력받아 복호된 신호를 출력한다. DEC1에서 복호된 신호는 인터리버를 거친 후 두번째 RSC의 출력 y_{2k}와 DEC2로 입력되어 복호된다. DEC2에서 복호된 신호는 역 인터리버를 거친후 DEC1으로 궤환되어 반복 복호를 실행함으로 성능을 개선시킨다.



(그림 5) 터보코드 복호기

DEC1과 DEC2에서 사용되는 복호 알고리즘을 간단히 설명하면 다음과 같다. 정보비트는 부호기의 인터리버 크기가 N인 프레임 단위로 채널을 통하여 전송되고, 복호기는 MAP 알고리즘을 이용하는데, MAP 알고리즘이란 전체 수신된 신호 프레임을 관찰한 후 임의의 시점에서 정보비트가 0일 확률과 1일 확률을 비교하여 확률이 더 큰 값을 복호 값으로 선택하는 알고리즘이다. 즉, 다음과 같이 LLR(logarithm of likelihood ratio)을 계산하고

$$\Lambda(d_k)=\log\frac{\Pr\{d_k=1|\ \text{observation}\}}{\Pr\{d_k=0|\ \text{observation}\}} \tag{2}$$

다음과 같은 기준으로 복호하게 된다.

$$\begin{aligned}&\Lambda(d_k)\geq 0 \quad \text{이면} \quad \tilde{d_k}=1\\ &\Lambda(d_k)<0 \quad \text{이면} \quad \tilde{d_k}=0\end{aligned} \tag{3}$$

식 (2)의 LLR은 상태메트릭 $\alpha_k(m)$과 $\beta_k(m)$을 이용하여 다음과 같이 정의될 수 있다.

$$\Lambda(d_k)=\log\frac{\sum_m \alpha_k^1(m)\beta_k^1(m)}{\sum_m \alpha_k^0(m)\beta_k^0(m)} \tag{4}$$

식 (4)의 $\alpha_k(m)$은 순방향 메트릭으로 정의되는데, 시점 0에서부터 시점 k에 이르기까지 k시점에서 m 상태에 이를 수 있는 모든 경로의 메트릭 값의 합으로써 시점 0에서부터 k까지 순 방향으로 순환적으로 계산된다. 반면에 $\beta_k(m)$은 트렐리스의 마지막 시점에서 k 시점에 역으로 이르기까지 k시점에서 m 상태에 이를 수 있는 모든 경로의 메트릭 값의 합으로써 트렐리스의 마지막 시점에서부터 k까지 역 방향으로 순환적으로 계산된다.

5. 제안한 인터리버와 기존의 인터리버와의 알고리즘 복잡도 비교

현재 디지털통신 시스템에서 사용되는 블록, 대각 인터리버는 구조가 간단하여 많이 사용되고 있으나 성능이 랜덤 인터리버와 비교하여 떨어진다. 그러나 우수한 복호 성능 때문에 활발히 연구되고 있는 랜덤 인터리버는 성능이 다른 블록과 대각 인터리버와 비교하여 우수하지만 데이터를 랜덤하게 읽고 쓰기 때문에 구현상의 어려움이 있다. 만약 터보코드를 영상 통신 시스템에 적용할 경우 한 개의 정보비트를 복호하기 위해 MAP 복호 알고리즘은 트렐리스 상에서 순방향과 역방향 메트릭 값을 계산 해야하며, 터보코드의 고유한 특성을 살리기 위해 반복 복호를 실행함으로 심각한 시간 지연이 발생한다. 뿐만 아니라 인터리버 2개와 역 인터리버 2개를 사용함으로 한 개의 정보비트를 복호하기 위한 메트릭의 계산횟수 즉, 복잡도가 높아 심각한 시간 지연이 발생하여 실시간을 요하는 영상 통신 시스템에 적용하기는 매우 어려운 단점이 있다.

〈표 5〉 제안한 인터리버와 기존의 인터리버의 특성 비교

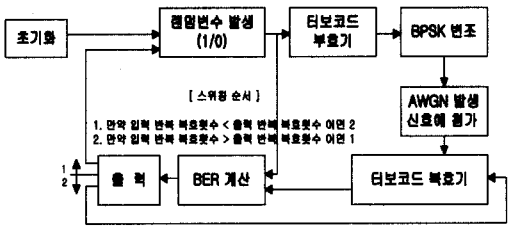
비교항목 \ 방법론	블록 인터리버	대각 인터리버	랜덤 인터리버	제안한 세미 랜덤 인터리버
방 법	데이터를 인터리버에 가로로 쓰고 세로로 읽음	데이터를 인터리버에 가로로 쓰고 대각으로 읽음	데이터를 인터리버에 랜덤하게 쓰고 랜덤하게 읽음	데이터를 인터리버에 가로로 쓰고, 랜덤하게 읽음과 동시에 다음 데이터가 입력됨
특 성	알고리즘이 간단	시스템에 적용시 블록 인터리버보다 성능이 우수	구현은 복잡하지만 시스템에 적용시 블록, 대각 인터리버보다 우수함	인터리버의 크기를 1/2로 감소시켜 데이터를 인터리버에서 읽음과 동시에 쓰기 때문에 인터리버의 크기가 1/2로 감소됨
연산횟수	O(S)	O(S)	O(S)	O(S/2)

s : 인터리버의 크기

이와 같은 문제점을 해결하기 위해 세미 랜덤 인터리버를 제안하였다. 제안한 세미 랜덤 인터리버와 기존의 블록, 대각, 랜덤 인터리버 알고리즘과 비교하면 <표 5>와 같다. <표 5>에서 알 수 있듯이 제안한 세미 랜덤 인터리버 알고리즘이 그 복잡도가 기존의 인터리버 알고리즘과 비교하여 1/2로 감소함을 알 수 있다. 이로 인해 시스템이 간단해지며, 데이터를 인터리버 내에 읽음과 동시에 다음 데이터가 그 주소 번지에 쓰여지기 때문에 시간 지연을 감소시킬 수 있음을 알 수 있었다.

6. 컴퓨터 시뮬레이션 및 성능분석

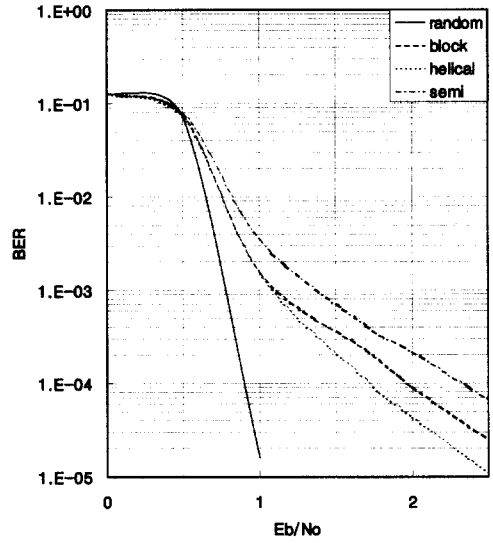
터보코드의 성능을 분석하기 위하여 (그림 6)에 나타난 시뮬레이션 모델을 사용하였다. 반복 복호횟수, E_b/N_o값, 생성다항식을 초기화를 하고, 랜덤신호 발생기에서 2진 신호(0/1)을 발생하여 터보 부호기에 입력된다. 정보비트인 2진 신호를 입력받아 부호화 율 1/2로 천공된 신호는 1을 1로, 0를 -1로 BPSK변조되어 AWGN 채널로 전송된다. 잡음이 첨가된 신호는 복호기에서 MAP 복호 알고리즘을 이용하여 신호를 판정하고 복호한 후에 부호기의 입력과 비교하여 BER을 계산한다. 이때 궤환은 (그림 6)의 스위칭 순서에 의해 이루어진다.



(그림 6) 시뮬레이션 모델

6.1 인터리버의 종류에 따른 성능분석

(그림 7)과 <표 6>은 인터리버의 크기가 65536이고 생성다항식 g=37/21, 반복 복호횟수가 6회일 때 BPSK 변조와 AWGN 채널을 이용하여 인터리버 종류에 따른 각각의 비트 오류 확률을 나타냈다. 그 결과 $BER = 10^{-4}$일 때, 랜덤 인터리버는 E_b/N_o의 값이 0.96[dB]이고, 대각 인터리버는 1.81[dB], 블록 인터리버는 1.92



(그림 7) 인터리버 종류에 따른 성능분석
(인터리버의 크기가 65536, 생성다항식 g=37/21, 반복복호횟수 6회)

〈표 6〉 인터리버 종류에 따른 성능 분석

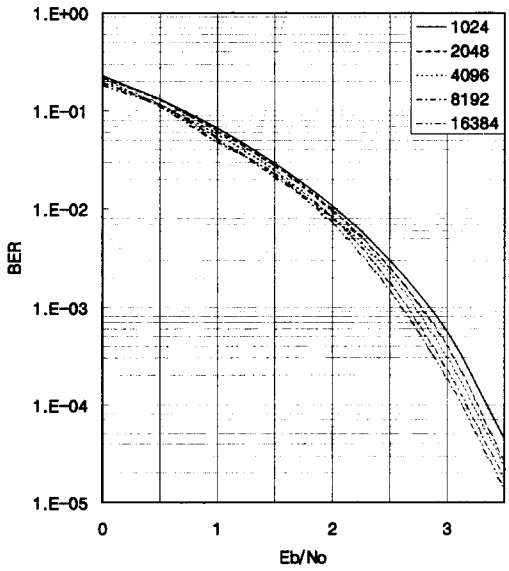
인터리버의 크기	반복횟수	BER	인터리버 종류	Eb/No[dB]
65536	6	10^{-4}	랜 덤	0.96
			대 각	1.81
			블 록	1.92
			세미랜덤	2.45

[dB], 그리고 세미 랜덤 인터리버는 2.45[dB]이다. 세미 랜덤 인터리버와 기존의 블록 인터리버와 비교할 때 그 성능이 0.53[dB] 감소되었다.

6.2 Semi-Random 인터리버의 크기와 구속장에 따른 성능분석

(그림 8)에서 (그림 11)은 한 개의 정보비트를 복호하기 위한 시스템의 복잡도를 고려하여 반복 복호횟수를 3회, 세미 랜덤 인터리버의 크기를 1024에서 16384까지 2배씩 증가시키고, 생성다항식을 37/21, 17/15, 7/5, 3/1로 각각 변화시키며 BPSK로 변조하여 AWGN 채널로 전송 후 복호기에서 MAP 알고리즘과 세미 랜덤 인터리버를 이용한 터보코드의 성능을 분석하였다.

(그림 8)과 <표 7>은 생성다항식이 g=37/21, 반복 복호횟수가 3회, $BER=10^{-4}$일 때, 세미 랜덤 인터리버의 크기를 1024에서 16384까지 2배씩 증가시키며 각각에 대한 E_b/N_o 값을 나타냈다. 세미 랜덤 인터리버의 크기가 1024일 때 E_b/N_o 값은 3.40[dB], 2048일 때는 3.25[dB], 4096일 때는 3.15[dB]이다. 그리고 8192일 때는 3.05[dB]이고, 16384일 때는 E_b/N_o값이 2.95[dB]이다. 그 결과 세미 랜덤 인터리버의 크기가 1024에서 4096까지는 E_b/N_o=0.35[dB] 향상되고, 세미 랜덤 인터리버의 크기가 4096에서 16384까지는 E_b/N_o=0.20[dB] 향상되었다. 그러므로 세미 랜덤 인터리버의 크기가 4096이상일 때 성능은 향상되지만 크게 개선되지는 않았다.

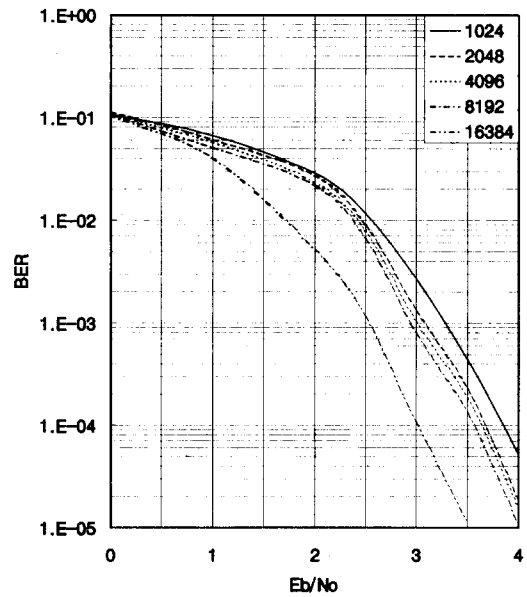


(그림 8) 세미 랜덤 인터리버 크기에 따른 성능분석
(생성다항식 g=37/21, 반복복호횟수 3회)

〈표 7〉 세미 랜덤 인터리버 크기에 따른 성능 분석
(생성다항식 g=37/21)

생성다항식	복호 횟수	BER	인터리버 크기	Eb/No[dB]
37 / 21	3	10^{-4}	1024	3.40
			2048	3.25
			4096	3.15
			8192	3.05
			16384	2.95

(그림 9)와 <표 8>은 생성다항식이 g=17/15, 반복 복호횟수는 3회, $BER=10^{-4}$로 일정할 때, 세미 랜덤 인터리버의 크기를 1024에서 16384까지 2배씩 증가시키며 세미 랜덤 인터리버 크기에 따른 성능을 분석하였다. 세미 랜덤 인터리버의 크기가 1024일 때 E_b/N_o 값은 3.80[dB], 2048일 때는 3.60[dB], 4096일 때는 3.55[dB]이다. 그리고 8192일 때는 3.48[dB]이고, 16384일 때는 E_b/N_o값이 2.98[dB]이다.



(그림 9) 세미 랜덤 인터리버 인터리버 크기에 따른 성능분석
(생성다항식 g=17/15, 반복복호횟수 3회)

〈표 8〉 세미 랜덤 인터리버 크기에 따른 성능 분석
(생성다항식 g=17/15)

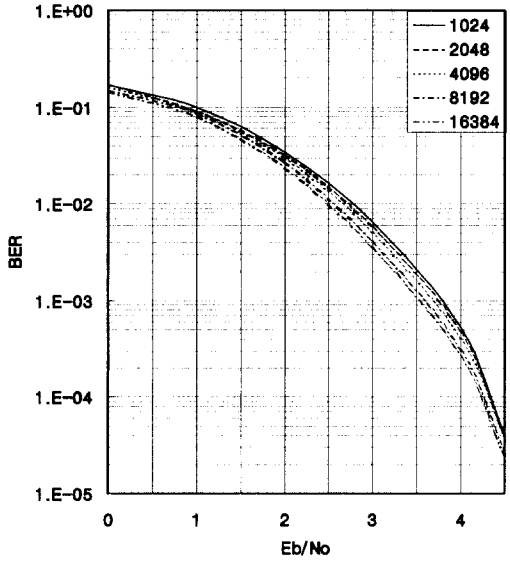
생성다항식	복호 횟수	BER	인터리버 크기	Eb/No[dB]
17 / 15	3	10^{-4}	1024	3.80
			2048	3.60
			4096	3.55
			8192	3.48
			16384	2.98

〈표 9〉 세미 랜덤 인터리버 크기에 따른 성능 분석
(생성다항식 g=7/5)

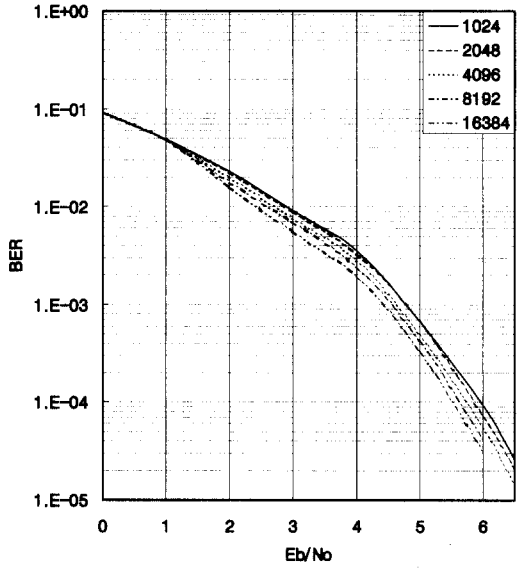
생성다항식	복호 횟수	BER	인터리버 크기	Eb/No[dB]
7 / 5	3	10^{-4}	1024	4.30
			2048	4.25
			4096	4.20
			8192	4.17
			16384	4.15

(그림 10)과 <표 9>는 생성다항식이 g=7/5, 반복 복호횟수가 3회, 그리고 세미 랜덤 인터리버의 크기를 1024에서 16384까지 2배씩 증가시키며 $BER=10^{-4}$일 때, E_b/N_o값을 구하였다. 세미 랜덤 인터리버의 크기가 1024일 때 E_b/N_o 값은 4.30[dB], 2048일 때는 4.25[dB], 4096일 때는 4.20[dB]이다. 그리고 8192일 때는 4.17[dB]이고, 16384일 때는 E_b/N_o 값이 4.15[dB]이다. 그 결과 세미 랜덤 인터리버의 크기가 1024에서 4096까지는 E_b/N_o=0.10[dB] 향상되고, 세미 랜덤 인터리버의 크기가 4096에서 16384까지는 E_b/N_o=0.05[dB] 향상된다. 그러므로 세미 랜덤 인터리버의 크기가 4096이상일 때 BER에 대한 E_b/N_o의 값은 향상되지만 세미 랜덤 인터리버의 크기가 4096이하 일 때보다 크게 개선되지는 않는다.

(그림 11)과 <표 10>은 생성다항식이 g=3/1, 반복 복호횟수가 3회, 그리고 세미 랜덤 인터리버의 크기를 1024에서 16384까지 2배씩 증가시키며 $BER=10^{-4}$로 일정할 때 세미 랜덤 인터리버 크기에 따른 성능을 분석하였다. 세미 랜덤 인터리버의 크기가 1024일 때 E_b/N_o 값은 5.95[dB], 2048일 때는 5.80[dB], 4096일 때는 5.70[dB]이다. 그리고 8192일 때는 5.60[dB]이고, 16384일 때는 E_b/N_o 값이 5.50[dB]이다. 그 결과 세미 랜덤 인터리버의 크기가 1024에서 4096까지는 E_b/N_o =0.25[dB] 향상되고, 세미 랜덤 인터리버의 크기가 4096에서 16384까지는 E_b/N_o=0.20[dB] 향상되었다. 그러므로 세미 랜덤 인터리버의 크기가 4096이상일 때 E_b/N_o 값은 조금씩 향상되지만 세미 랜덤 인터리버의 크기가 4096이하 일 때보다 성능은 크게 향상되지 않았다.



(그림 10) 세미 랜덤 인터리버 크기에 따른 성능분석
(생성다항식 g=7/5, 반복복호횟수 3회)



(그림 11) 세미 랜덤 인터리버 크기에 따른 성능분석
(생성다항식 g=3/1, 반복복호횟수 3회)

〈표 10〉 인터리버 크기에 따른 성능분석
(생성다항식 g=3/1)

생성다항식	복호 횟수	BER	인터리버 크기	Eb/No[dB]
3/1	3	10^{-4}	1024	5.95
			2048	5.80
			4096	5.70
			8192	5.60
			16384	5.50

컴퓨터 시뮬레이션 결과 동일한 조건하에서 기존의 블록 인터리버와 E_b/N_o대 BER을 비교할 때 블록 인터리버가 본 논문에서 제안한 세미랜덤 인터리버보다 성능은 0.53[dB]정도 향상되지만 복잡도가 증가하여 한 개의 정보비트를 복호할 때 시간지연이 발생하여 실시간을 요하는 통신에는 적합하지 않다. 그리고 세미 랜덤 인터리버를 이용한 터보코드의 성능 시뮬레이션 결과 BPSK 시스템을 이용한 AWGN 채널 상에서 복호기의 복잡도와 시간 지연을 고려할 때 반복 복호 횟수는 3회, 인터리버의 크기는 4096, 생성다항식은 17/15에서 우수한 복호 성능을 나타냈다.

7. 결 론

영상 데이터를 압축하여 무선 통신 채널 상으로 전송할 때 대기 중의 잡음으로 데이터에 오류가 발생하게 되며, 이때 수신 단에서 압축된 영상 데이터를 복호하더라도 채널 상에서 발생한 잡음으로 인해 원 영상을 복호할 수 없다. 따라서, 본 논문에서는 영상통신에 있어 채널 상에서 발생하는 잡음을 효과적으로 제거할 수 있는 방법으로 복호 성능이 우수한 터보코드를 적용하였으며, 실시간 영상 데이터를 처리하기 위해 기존의 인터리버보다 시스템의 복잡도를 1/2로 감소시키는 새로운 세미 랜덤 인터리버의 알고리즘을 제안하였다. 세미 랜덤 인터리버 알고리즘은 입력한 프레임의 길이를 1/2 크기만큼 인터리버를 구성하고, 인터리버 내에 데이터를 입력할 때는 블록 인터리버처럼 행으로 입력하고, 데이터를 읽을 때는 랜덤하게 읽음과 동시에 다음 데이터가 그 주소 번지에 위치하게 된다. 따라서 기존의 블록, 대각, 랜덤 인터리버와 알고리즘의 복잡도를 비교할 시 그 복잡도가 1/2로 감소되므로, 터보코드에 세미 랜덤 인터리버를 적용하여 영상 데이터를 전송할 때 채널 상에서 발생하는 잡음을 효과적으로 제거할 수 있으며, 영상 데이터를 복호할 때 시간지연을 감소시킴으로 실시간 영상처리가 가능하다.

참 고 문 헌

[1] FPLMTS/IMT-2000, Report of the Tenth Meeting of ITU-R Task Group8/1, Mainz, April 1996.

[2] Y. Fisher(Ed.), "Fractal Compression : Theory and Application to Digital Images," Springer Verlag, New York. 1994.

[3] A. Averbuch, D. Lazer and M. Israeli, "Images Compression Using Wavelet Transform and Multi-resolution Decomposition," IEEE Trans. Image Processing, Vol.5, No.1, pp.4-15, Jan., 1996.

[4] 공성곤, "부영상의 퍼지 분류에 의한 영상 데이터 압축", 한국통신학회지, Vol.14, No.9, 1997.

[5] Berrou.C, Glavieux.A, "Near shannon limit error coding and decoding : turbo codes," Proc. ICC'93, pp.1064-1070.

[6] D. Divsalar and F. Pollara, "Turbo codes for deep space communIcations," TDA progress rep. 42-120. Jet propulsion lab., pasadena, CA, pp.66-77, Feb. 15 1995.

[7] Heller, J. A., Jacobs, I. W., "Viterbi decoding for satellite and space communications," IEEE Trans. Commun.Technol., Vol.COM19, No.5, pp.835-848, Oct. 1971.

[8] Kohlenberg, A., Forney, G. D., "Convolutional coding for channels with memory," IEEE Trans. Inf. Theory, Vol.IT2, pp.618-626, 1968.

[9] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," Proc. ICC95, seattle, WA, pp.18-22, June 1995.

[10] Wang XD, Poor HV, "Iterative(Turbo) soft interference cancellatin and decoding for coded CDMA," IEEE Trans. on Communication Vol.47, No.7, pp. 1046-1061, July. 1999.

[11] Alexander PD, Reed MC, Asenstorfer JA, Schlegel CB, "Iterative multiuser interference reduction Turbo CDMA," IEEE Trans. on Communication Vol.47, No.7, pp.1008-1014, July. 1999.

[12] L. R. Bahl, J. Cocke, F. Jelinek, and Jraviv, "Opti-

mal Decoding of Linear Codes for Minimizing Symbol Error Rate," IEEE Trans. Inform. Theory., Vol.IT-20, pp. 284-287, Mar. 1974.

[13] Hagenauer. J, Robertson. P, and Papke. L, "Iterative(turbo) decoding of systematic convolutional codes with the MAP and SOVA," Submitted to ITG 1994 Conf., October 1994.

[14] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in Proc., IEEE Int. conf, on Commun. (seattle, 1995), pp. 1009-1013.

[15] L. Lin and R. Cheng, "Improvements In SOVA-Based Decoding For Turbo Codes," Proc of ICC, pp.1473-8, June 1997.

[16] L. Papke, p. Robertson, and E. Villebrun, "Improve decoding with the SOVA in a parallel concated (Turbo-code) schem," in Proc., IEEE Int. Conf. on Commun. pp.102-106, 1996.

[17] Minowa T, Ogiwara H, "Application of soft-in/soft-out Viterbi algorithm to turbo trellis coded," IEICE Transactions on Fund. of Elect. Communi. & Computer Vol.E81-A, No.10, pp.2047-2054.

[18] Franz V, Anderson JB, "Concatenated decoding with a reduced search BCJR algorithm," IEEE Journal on Sel. Areas in Comm.,Vol.16, No.2, Feb. 1998.

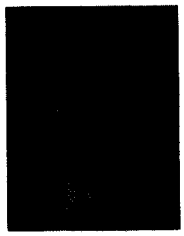
[19] S. Hong, W. E.Stark, "VLSI Circuit Complexity and Decoding Performance Analysis for Low-Power RSC," Proceedings of the Military Comm. Conf. Vol.3, pp.708-712, Oct. 1998.

[20] Ping L, "Modified turbo codes with low decoding complexity," Electronics Letters, Vol.34 No.23, pp. 2228-2229, Dec. 1998.

[21] F. Berens, T. Bing, H. Michel, A. Worm, P. W. Baier, "Performance of Low Complexity Turbo-Codes in the UTRA-TDD-Mode," Proceedings of the IEEE VTS 50th Vehicular Tech. Conf. - Vol. 5, pp.2621-2625, September 1999.

[22] Ramsey, J. L., "Realization of optimum interleavers," IEEE Trans. Inf. theory. Vol IT16, No.3, pp.338-345, May 1970.

[23] A. S. Barbulescu and S. S. Pietrobon, "Interleaver design for turbo codes," Electronics Letters 8th Vol.30 No.25, Dec. 1994.



홍 성 원

e-mail : hongsw@wslab.chongju.ac.kr

1993년 청주대학교 전자공학과 졸업(학사)

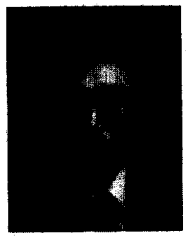
1995년 청주대학교 대학원 전자공학과(공학석사)

2000년 청주대학교 대학원 전자공학과(공학박사)

1995년~현재 모아통신(주) 부설통신연구소 전임연구원

1999년~현재 남서울대학교 정보통신공학과 겸임전임강사

관심분야 : 디지탈 이동통신, 부호이론, Spread Spectrum 통신, Multimedia 통신, 영상통신



박 진 수

e-mail : parkjs@chongju.ac.kr

1975년 한양대학교 전자공학과 졸업(학사)

1977년 한양대학교 대학원 전자통신공학과(공학석사)

1975년 한양대학교 대학원 전자통신공학과(공학박사)

1987년~1988년 Univ. Colorado at Colorado Spring (Post Doc.)

1978년~현재 청주대학교 첨단공학부 교수

1999년~현재 과학기술부 · 한국과학재단 지정 정보통신연구센터장

2000년~현재 한국정보처리학회 충북지부 지부장

관심분야 : 디지탈 이동통신, 부호이론, Spread Spectrum 통신, Multimedia 통신, 영상통신