

論文2002-39TC-4-1

CRC를 사용한 터보부호에서 edge-effect를 감소시키기 위한 인터리버

(An interleaver to reduce the edge-effect in turbo codes with CRC)

李秉吉*, 裴商宰*, 鄭建鉉**, 朱彦熈*

(Byung Gil Lee, Sang Jae Bae, Keon Hyun Jung, and Eon Kyeong Joo)

요약

차세대 이동통신 시스템에서 고품질의 멀티미디어 서비스를 제공하기 위해서는 강력한 오류정정부호의 사용이 필수적이다. 터보부호에서는 반복복호를 수행함으로써 우수한 오류성능을 얻을 수 있지만 반복복호 회수가 증가될수록 연산량과 지연시간도 증가된다. 따라서 우수한 오류성능을 유지하면서 반복복호 회수를 줄이기 위한 방법이 요구된다. CRC를 사용하는 터보부호는 이러한 반복복호 회수를 줄일 수 있는 가장 효과적인 방법으로 알려져 있다. 그러나 CRC를 사용하지 않은 기존의 터보부호에서와 같이 CRC를 사용하는 터보부호에서도 edge-effect에 의해 오류성능이 크게 저하될 수 있다. 따라서 본 논문에서는 기존의 s-랜덤 인터리버에 D-파라미터를 적용하여 edge-effect를 제거하기 위한 방법이 제시된다. 모의실험 결과 CRC를 사용한 터보부호에서 본 논문에서 제시된 D-파라미터를 적용한 인터리버를 사용하는 경우 edge-effect가 완전히 제거됨을 확인할 수 있었다.

Abstract

In the next generation mobile communications, powerful channel coding is essential in order to obtain high quality multimedia services. Turbo code can achieve good error performance by iterative decoding, but more iterations result in additional computational complexity and delay. Thus, a method to reduce the number of iterations without additional performance degradation is needed. Turbo code with CRC is known to be the most efficient method to reduce the number of iterations. In this scheme, the performance may be degraded by the edge-effect like the conventional turbo code without CRC. In this paper, a method to eliminate the edge-effect is proposed by adopting D-parameter to the conventional s-random interleaver. As results of simulation, the edge-effect of the turbo code with CRC is shown to be successfully eliminated by using the new interleaver designed with D-parameter.

Key words : turbo codes, interleaver, edge-effect, CRC.

* 正會員, 慶北大學校 電子電氣工學部

(School of Electronic and Electrical Engineering,
Kyungpook National University)

** 正會員, 三星電子(株) 無線事業部

(Mobile Communication Division, Samsung Electronics, Co. Ltd.)

接受日字:2001年7月11日, 수정완료일:2002年3月13日

I. 서론

차세대 이동통신 시스템은 기존의 음성 서비스뿐만 아니라 고속 데이터와 영상 등이 복합된 멀티미디어 서비스 제공을 목표로 하고 있다. 그러나 이동통신 시스템처럼 매우 열악한 무선채널 환경에서 고신뢰성 및 고속의 멀티미디어 서비스가 제공되기 위해서는 매우 우수

한 오류정정 능력을 가진 채널 부호의 사용이 필수적이다.

1993년 Berrou 등에 의해서 발표된 터보부호^[1~4]는 비교적 간단한 구조를 가지면서도 매우 우수한 성능을 나타낸다. 터보부호의 우수한 성능은 인터리버와 반복복호 때문이며, 인터리버의 크기가 크고 반복복호 회수가 증가될수록 오류정정 능력이 더 우수해진다. 그러나 반복복호 회수가 증가할수록 복호기에서의 복잡도와 지연시간이 증가하게 된다. 이것은 실시간 및 고속의 데이터 통신에서 큰 단점이 된다.

이러한 단점을 보완하기 위하여 반복복호 회수를 줄이는 방법들이 제안되었다. 그 종류에는 복호기 출력인 LLR(log likelihood ratio)값을 문턱 값과 비교하는 방법^[5]과 LLR 값의 분산을 이용하는 방법^[6], CRC(cyclic redundancy check)를 이용하는 방법^[7] 등이 있다. 이러한 방법들 중에서 CRC를 사용하는 방법이 반복 회수를 가장 많이 줄여 줄 수 있으며 성능 또한 가장 우수한 것으로 알려져 있다^[5~7].

또한 터보부호에서는 사용되어지는 인터리버의 종류에 따라 성능의 차이가 난다. 그 중 일반적으로 s-랜덤 인터리버가 사용되는 경우 성능이 가장 우수한 것으로 알려져 있다^[8,9]. 그러나 이러한 s-랜덤 인터리버에서는 프레임의 끝 부분에 비트 오류가 집중되는 경우 edge-effect로 인해 오류정정 능력이 크게 저하될 수 있다^[6].

본 논문에서는 터보부호에서 발생할 수 있는 edge-effect의 영향을 분석하고 이 edge-effect를 제거할 수 있는 인터리버의 설계 방법을 제시한다. 제시된 인터리버에서는 인터리버의 끝 부분에 위치한 비트들이 다시 끝 부분으로 인터리빙 되는 것을 막아주기 위하여 적절한 보호구간의 길이를 나타내는 D-파라미터를 도입하였다. 또한 기존의 터보부호에서 제안된 인터리버를 사용하여 edge-effect에 의한 성능저하가 방지될 수 있음을 모의 실험을 통하여 확인한다.

II. 터보부호 및 CRC를 이용한 터보부호

1. CRC를 사용한 터보부호

CRC를 사용한 터보부호의 부호기는 그림 1과 같이 나타낼 수 있다. 터보 부호기 앞에 CRC 부호기를 붙여 주는 것을 제외하면 동작원리는 기존의 터보부호의 부호기와 동일하다. 즉 입력된 정보열은 CRC 부호화되어서 기존의 터보부호의 부호기에 입력되고, 입력된 비트

열들이 다시 부호화되어 체계적 정보(systematic information)인 d_k 와 패리티 성분인 p_{1k}, p_{2k} 가 만들어진다.

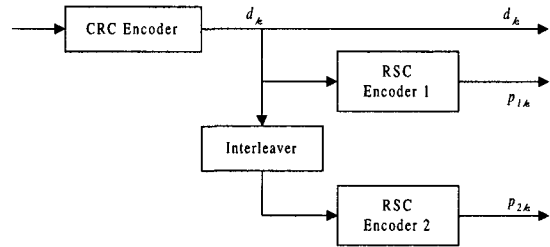


그림 1. CRC를 사용한 터보부호의 부호기
Fig. 1. Encoder of turbo code with CRC.

복호기의 구조는 그림 2와 같이 나타낼 수 있다. 여기서 x_k 는 체계적 정보인 d_k 가 채널을 통하여 복호기에서 수신된 값이며 y_{1k} 와 y_{2k} 는 각각 p_{1k}, p_{2k} 이 수신된 값이다. 그리고 사전정보를 나타내는 z_k 는 잉여정보인 $L_e(d_k)$ 가 역인터리빙된 값이다. 기존의 터보 복호기는 일정한 반복복호 회수만큼 반복복호한 후에 출력되는 LLR값을 경판정하여 출력 정보열을 만들어 내지만, CRC를 이용한 경우는 매 반복복호 때마다 출력되는 LLR값을 경판정하여 그 값으로 CRC 복호기에서 오류 여부를 확인하는 과정이 추가된다. 이때 오류가 없다면 반복을 멈추고, 오류가 있다면 계속해서 반복복호를 한다. 또 반복복호를 하더라도 오류가 수정되지 않는 프레임임을 위해 일정한 반복 회수가 되면 반복복호가 멈추어질 수 있도록 최대 반복복호 회수를 정해준다.

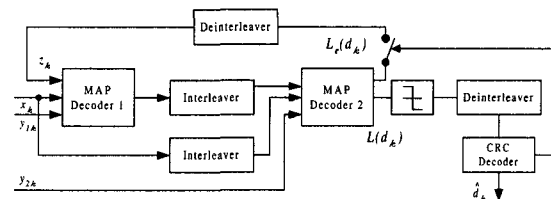


그림 2. CRC를 사용한 터보부호의 복호기
Fig. 2. Decoder of turbo code with CRC.

2. CRC를 사용한 터보부호의 오류 성능 및 edge-effect의 영향

CRC를 사용한 터보부호의 성능을 살펴보면 그림 3과 4와 같다. 터보부호의 부호율은 1/3이며 구성부호의 생성 다항식은 (7,5)₍₈₎이다. 변조방식으로는 BPSK를 사용하였고 가산성 백색 가우시안 잡음(additive white

Gaussian noise ; AWGN) 채널 환경에서 모의실험을 행하였다. 터보부호의 내부 인터리버로는 s-랜덤 인터리버를 사용하였으며 그 크기는 1296이다. 그리고 프레임 오류를 검출하기 위한 CRC로는 생성다항식이 $x^{16} + x^{15} + x^2 + 1$ 인 16-CRC를 사용하였다.

s-랜덤 인터리버는 랜덤변수생성기에 의해 설계되기 때문에 동일한 파라메타를 가지는 경우라도 설계된 인터리버는 달라질 수 있다. 따라서 실제 설계된 인터리버에 따라 매번 성능이 다르게 나타날 수 있다. 그 중 성능이 가장 나쁜 것과 가장 좋은 것의 예를 인터리버 1 과 인터리버 2로 나타내었다. 그림으로부터 3dB의 E_b/N_0 에서 오류성능차이가 BER의 경우 약 20배, FER의 경우 약 50배 정도의 차이가 남을 알 수 있다.

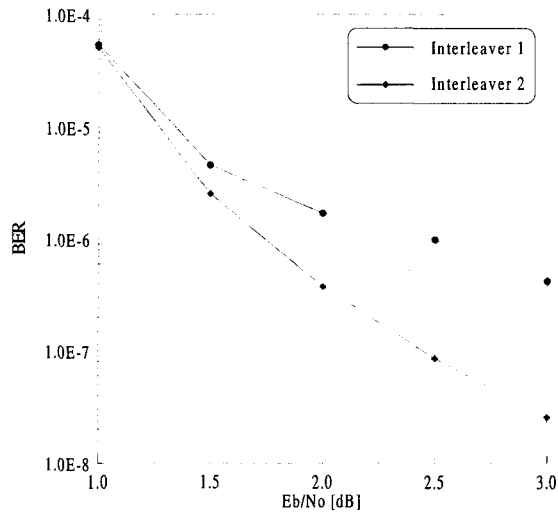


그림 3. 비트 오류의 비교
Fig. 3. Comparison of bit error rate.

표 1은 오류가 발생하는 위치를 정보 비트와 CRC 비트로 나누어 오류 분포를 나타내었다. 만약 한 프레임 내의 모든 비트에서 오류가 일어날 확률이 동일하다면 CRC 비트에 오류가 일어날 확률은 $16/1296 = 1.23\%$ 가 된다. 즉 대부분의 오류는 정보 비트에 발생하고 CRC 비트에는 적게 발생하여야 한다. 그러나 인터리버 1의 경우 낮은 E_b/N_0 에서는 인터리버 2와 비슷한 분포를 보이지만 E_b/N_0 가 높아질수록 CRC 비트에 일어난 오류의 비가 증가한다. 이와는 달리 인터리버 2는 모든 E_b/N_0 에서 거의 비슷한 비를 가짐을 알 수 있다. 이로부터 인터리버 1과 같은 특정한 인터리빙 패턴에서는 인터리버 2에 비해 정보 비트의 비트오율은 유사하지만

프레임의 끝 부분에 위치한 CRC 비트에서는 훨씬 많은 비트 오류가 발생함을 알 수 있다.

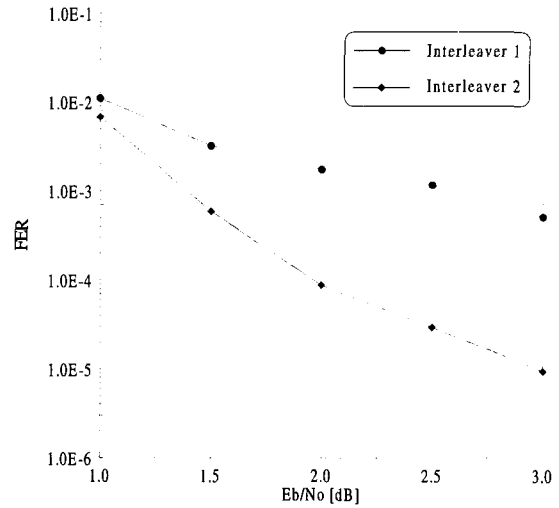


그림 4. 프레임 오류의 비교
Fig. 4. Comparison of frame error rate.

표 1. 오류 분포

Table 1. Error distribution.

| E_b/N_0 [dB] | Interleaver 1 | | Interleaver 2 | |
|----------------|---------------|--------------|---------------|--------------|
| | ϵ_i | ϵ_c | ϵ_i | ϵ_c |
| 1.0 | 92 | 8 | 98 | 2 |
| 1.5 | 53 | 47 | 98 | 2 |
| 2.0 | 27 | 73 | 98 | 2 |
| 2.5 | 11 | 89 | 96 | 4 |
| 3.0 | 8 | 92 | 96 | 4 |

ϵ_i : 정보 비트 내의 오류 [%]

ϵ_c : CRC내의 오류 [%]

이러한 CRC 비트에 오류가 집중되는 패턴을 살펴보면 그림 5와 같이 나타낼 수 있다. $\pi(\cdot)$ 는 인터리빙을 의미하며, $\pi(i)$ 는 인터리빙 후 i 번째 위치하는 비트를 의미한다. 프레임의 마지막 부분에 위치한 비트가 인터리빙 후에도 프레임의 마지막 부분에 위치하게 될 경우 이러한 프레임의 끝 부분에 위치한 비트들에서는 다른 부분의 비트들에 비해 훨씬 많은 비트 오류가 발생한다. 이러한 현상을 edge-effect라고 한다^[6]. 그림 3과 4에서 인터리버 1은 edge-effect가 발생한 대표적인 경우이며 인터리버 2는 그렇지 않은 경우이다.

이 edge-effect가 발생하는 원인은 프레임의 마지막에 위치한 비트들의 경우 대체로 낮은 무게(weight)를 축적시키기 때문이다. 따라서 프레임의 마지막 부분에 위치한 비트가 역시 프레임의 끝 부분으로 인터리빙되는 경우에는 터보부호의 부호기에 의해 발생하는 부호어의 무게도 작아지며 이러한 작은 무게를 가지는 부호어에 의해 오류성능도 저하된다.

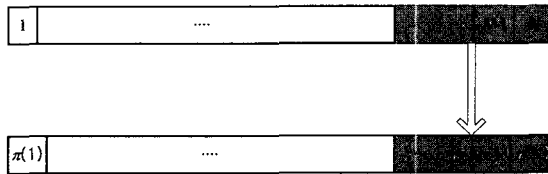


그림 5. edge-effect가 발생하는 인터리빙 형태
Fig. 5. Interleaving pattern with edge-effect.

표 2는 edge-effect가 발생한 인터리버 1과 그렇지 않은 인터리버 2에 대한 평균 반복복호 회수를 나타내었다. 오류성능과는 달리 반복복호 회수의 차이는 비교적 미미함을 알 수 있다.

표 2. 평균 반복복호 회수
Table 2. Average number of iterations.

| Interleaver type | E_b/N_0 [dB] | | | | |
|------------------|----------------|------|------|------|------|
| | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 |
| Interleaver 1 | 3.65 | 2.64 | 2.12 | 1.95 | 1.61 |
| Interleaver 2 | 3.64 | 2.63 | 2.11 | 1.94 | 1.60 |

III. edge-effect를 제거하기 위한 인터리버의 설계

인터리버의 종류에는 블록 인터리버, 대각 인터리버, 랜덤 인터리버, s-랜덤 인터리버 등이 있다. 이 중에서 일반적으로 s-랜덤 인터리버의 성능이 가장 우수하다고 알려져 있다^[8,9]. s-랜덤 인터리버는 랜덤 인터리버의 단점인 자기 종료(self terminating)를 최소화하고, 부호의 무게(weight)를 크게 하기 위해 인접한 위치에 있는 비트들을 일정한 거리 S 이상으로 분산시킴으로서 만들 수 있다. 이때 S 의 값은 N 을 인터리버 크기로 정의할 때 일반적으로 $S < \sqrt{N/2}$ 인 값을 사용한다^[8,9].

s-랜덤 인터리버의 구현 알고리즘은 첫 번째로 임의의 수를 생성하고, 두 번째로 생성된 수와 이전 S 번째

이내의 값과 비교하여 차가 $\pm S$ 내의 값이면 이 값을 버리고, 아니면 선택한다. 동일한 과정을 인터리버의 첫 번째 위치부터 마지막 N 번째 위치까지 반복한다. 이 알고리즘의 흐름도를 나타내면 그림 6과 같이 나타낼 수 있다. 그러나 s-랜덤 인터리버에서 프레임의 끝 부분 비트가 인터리빙 후 다시 프레임의 끝 부분에 위치하게 되면 edge-effect가 나타난다.

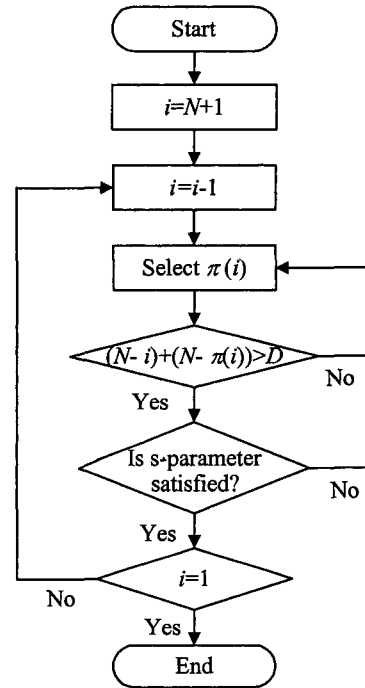


그림 6. s-랜덤 인터리버의 흐름도
Fig. 6. Flow chart of the s-random interleaver.

이때 edge-effect가 나타나는 인터리빙전후의 프레임의 끝 부분으로부터의 거리의 합을 D 라고 하면, 프레임의 길이 N 과 거리 D 와의 관계는

$$(N-i) + (N-\pi(i)) = D \tag{1}$$

로 나타낼 수 있다. 여기서 $\pi(i)$ 는 i 번째 인터리빙 테이블의 값을 나타낸다. 생성다항식이 (7,5)₍₈₎인 RSC 부호기를 사용하였을 경우 무게가 1인 입력 정보열이 (...100000000000...)이면 출력되는 패리티열은 (...1110110110110...)이 되며 입력되는 프레임 중 '1'의 위치에 따라 패리티열에서 축적되는 '1'의 개수가 달라진다. 즉 입력 정보열에서 비트 '1'과 프레임 끝과의 거리가 멀어질수록 더 큰 무게의 패리티를 발생시킨다. 따라서 edge-effect가 나타날 수 있는 프레임의 끝에서 D 이

내에 있는 비트들에 대하여는 프레임의 끝에 가까이 있을수록 인터리빙 후에는 프레임의 끝에서 더 멀어지도록 인터리버가 설계되어야한다.

식 (1)의 D 값을 이용하여 s -랜덤 인터리버의 구현 알고리즘을 수정한 흐름도를 나타내면 그림 7과 같이 나타낼 수 있다.

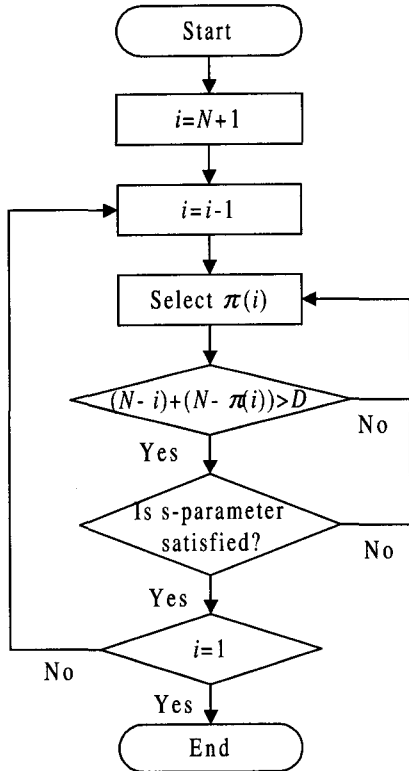


그림 7. 제안된 인터리버의 설계 흐름도
Fig. 7. Flow chart of the proposed interleaver.

기존의 s -랜덤 인터리버의 구현 알고리즘과의 차이점은 크게 두 가지로 나타낼 수 있다.

첫째, D -파라메타의 도입이다. D -파라메타의 값을 나타내는 D 는 프레임 끝에서 선택된 정보 비트까지의 거리와 인터리빙 된 후 이 정보 비트와 프레임의 끝까지의 거리를 합한 값이다. 이 D 값을 적당히 정하고, 이 값과 s 값이 만족되도록 인터리버를 구성한다. 이렇게 함으로써 프레임의 끝 부분에 위치한 D 개의 비트가 edge-effect가 나타날 수 있는 프레임의 범위 내로 인터리빙 되는 것을 막아준다.

둘째, 인터리버를 구현할 때 인터리빙 테이블의 마지막 값부터 구한다. D -parameter는 인터리버의 끝에서

D 이내의 거리에 있는 인터리버 값들을 구할 때만 적용된다. 즉, s -랜덤 인터리버에 D -파라메타를 도입하는 경우에는 인터리버의 끝 부분에 위치한 D 개의 인터리버 값에 대하여는 s -파라메타와 D -파라메타가 모두 만족되어야 하며 나머지 값들에 대하여는 s -파라메타만 만족시키면 된다. 만약 인터리버의 처음 부분부터 값을 구하게 되면 끝 부분에서는 선택 가능한 인터리버 값의 수가 아주 작아지기 때문에 이러한 값들을 가지고 D -파라메타를 만족시키는 인터리버를 구하기가 불가능해질 수 있다. 이 경우에는 인터리버의 처음 값부터 다시 구성하여야 하며 따라서 수많은 시행착오 과정을 반복하여야 한다. 이로 인해 인터리버를 구성하기 위한 조건이 더 까다로운 끝 부분의 인터리버 값들부터 선택하는 것이 처음에 위치한 값들부터 선택하는 것보다 쉽고 빠르게 구현할 수 있다.

이렇게 보완된 알고리즘으로 만들어진 인터리버는 기존의 s -랜덤 인터리버에서 나타날 수 있는 edge-effect를 제거시켜 터보부호의 성능을 향상시켜 줄 수 있다.

IV. 실험 결과 및 분석

모의 실험에는 구속장(constraint length)이 3이고 부호율이 1/3인 터보부호를 사용하였다. 구성 부호기인 RSC부호기의 생성다항식은 (7,5)₈이고, 인터리버의 크기 N 은 1296이며, AWGN 채널을 사용하였다. 복호 알고리즘은 성능이 우수한 MAP 알고리즘을 사용하였다^[10~12]. CRC를 이용한 반복복호 감소 알고리즘을 사용하였으며 최대 반복복호 회수는 8로 두었다. CRC는 생성다항식이 $x^{16} + x^{15} + x^2 + 1$ 인 16-CRC를 사용하였다.

그림 8은 D 의 값에 따른 비트 오류율을 나타내었다. 그림에서와 같이 D 가 0일때 오류성능이 가장 나쁘고 D 가 증가함에 따라 오류성능도 향상되며 D 가 15이상인 경우에는 더 이상의 오류성능의 향상이 거의 미미함을 알 수 있다. 따라서 제시된 모의실험 환경에서 s -랜덤 인터리버에서의 edge-effect를 완전히 제거하기 위해서는 D 의 값이 15이상이어야 함을 알 수 있다. 여기서 D 가 0인 것은 기존의 edge-effect가 발생된 s -랜덤 인터리버인 그림 3과 4의 인터리버 1에 해당한다. 또한 E_b/N_0 가 증가될수록 D 값에 따른 오류성능의 차이가 더욱 증가됨을 알 수 있다.

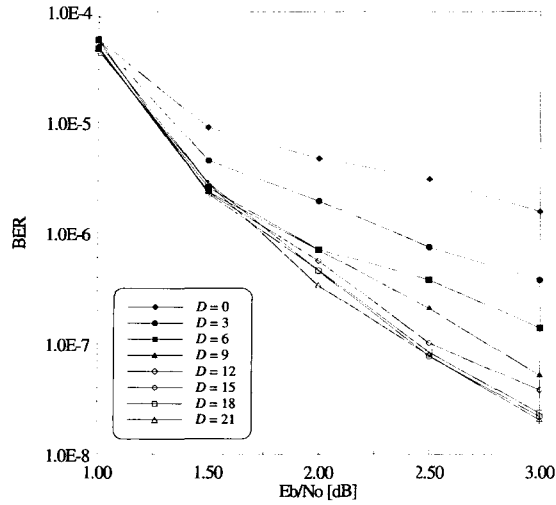


그림 8. D 값에 따른 비트 오류
Fig. 8. Bit error rate according to D .

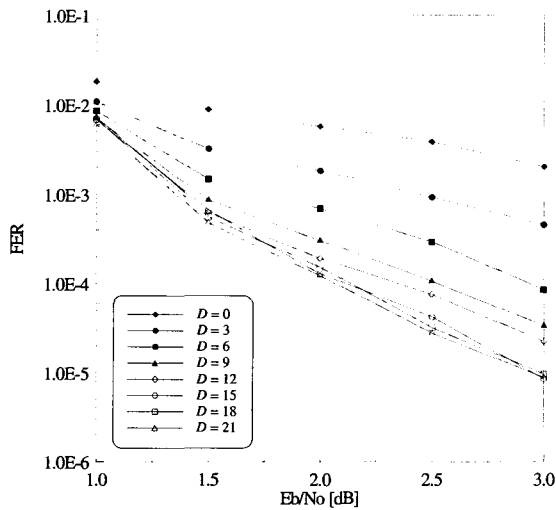


그림 9. D 값에 따른 프레임 오류
Fig. 9. Frame error rate according to D .

그림 9는 D 의 값에 따른 프레임 오류를 나타내었다. 프레임 오류도 비트 오류와 동일한 양상을 보임을 알 수 있다. 즉 인터리버의 크기가 1296이고 16-CRC를 사용하는 경우에 15이상의 D 값이 요구되는 것은 인터리빙 전 프레임의 끝 부분에 위치한 CRC비트들이 인터리빙 후에 다시 인접한 CRC비트 위치에 재정되지 않게 함으로써 edge-effect를 제거할 수 있음을 보여준다.

표 3은 D 값에 따른 오류 분포를 나타내었다. D 가 0일 경우 edge-effect에 의해 CRC 비트의 오류 확률이 특히 증가됨을 알 수 있다. 또한 높은 E_b/N_0 영역일수록 D 값이 큰 인터리버에 비해 CRC 비트에서의 오류가 더욱 증가된다. 이에 비해 D 가 15이상인 경우에는 CRC에서의 비트 오류가 아주 적게 발생함을 알 수 있다.

표 4. D 값에 대한 평균 반복복호 회수
Table 4. Average number of iterations according to D .

| $D \backslash E_b/N_0$ [dB] | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 |
|--------------------------------|------|------|------|------|------|
| 0 | 3.69 | 2.67 | 2.15 | 1.96 | 1.62 |
| 3 | 3.66 | 2.64 | 2.12 | 1.94 | 1.61 |
| 6 | 3.64 | 2.63 | 2.12 | 1.94 | 1.61 |
| 9 | 3.64 | 2.63 | 2.11 | 1.94 | 1.61 |
| 12 | 3.64 | 2.63 | 2.11 | 1.94 | 1.61 |
| 15 | 3.64 | 2.63 | 2.11 | 1.94 | 1.61 |
| 18 | 3.64 | 2.63 | 2.11 | 1.94 | 1.61 |
| 21 | 3.63 | 2.63 | 2.11 | 1.94 | 1.61 |
| Conventional MAP (without CRC) | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 |

표 3. D 값에 따른 오류분포
Table 3. Error distribution according to D .

| E_b/N_0 [dB] \ D | 0 | | 3 | | 6 | | 9 | | 12 | | 15 | | 18 | | 21 | |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ϵ_i | ϵ_c | ϵ_i | ϵ_c | ϵ_i | ϵ_c | ϵ_i | ϵ_c | ϵ_i | ϵ_c | ϵ_i | ϵ_c | ϵ_i | ϵ_c | ϵ_i | ϵ_c |
| 1.0 | 82 | 18 | 93 | 7 | 96 | 4 | 97 | 3 | 97 | 3 | 98 | 2 | 98 | 2 | 99 | 1 |
| 1.5 | 28 | 72 | 53 | 47 | 73 | 27 | 86 | 14 | 96 | 4 | 96 | 4 | 98 | 2 | 97 | 3 |
| 2.0 | 8 | 92 | 32 | 68 | 58 | 42 | 60 | 40 | 90 | 10 | 97 | 3 | 96 | 4 | 99 | 1 |
| 2.5 | 4 | 96 | 6 | 94 | 41 | 59 | 71 | 29 | 81 | 19 | 93 | 7 | 88 | 12 | 97 | 3 |
| 3.0 | 1 | 99 | 8 | 92 | 35 | 65 | 59 | 41 | 56 | 44 | 91 | 9 | 100 | 0 | 97 | 3 |

ϵ_i : 정보 비트 내의 오류 [%]
 ϵ_c : CRC내의 오류 [%]

표 4는 D 값에 따른 평균 반복복호 회수를 나타내었다. CRC를 사용하지 않고 반복복호를 8번으로 고정시켰을 경우에 비해 CRC를 사용한 경우에는 평균 반복복호 회수가 1dB에서는 3.63~3.69회로 약 54~55% 감소하고 3dB에서는 1.61~1.62회로 약 80% 감소되었다. 또한 아주 낮은 E_b/N_0 영역 외에는 D -파라메타 값에 따른 반복복호 회수의 감소 효과는 미미하였다. 이처럼 오류성능의 향상에 비해 반복복호의 감소효과가 미미한 이유는 프레임 오류이 10^{-2} 이하로 비교적 작은 값을 가지고 최대 반복복호 회수가 8로 고정되어 있기 때문으로 사료된다.

V. 결 론

차세대 이동통신 시스템에서는 음성 서비스뿐만 아니라 고 신뢰성 및 고속의 멀티미디어 서비스가 요구되어진다. 이러한 서비스를 만족시키기 위해서는 우수한 오류정정 능력을 가진 채널 부호의 사용이 필수적이다. 터보부호는 우수한 오류성능을 가지며 이러한 뛰어난 오류성능에 비해 비교적 간단한 구조를 가지므로 다양한 차세대 이동통신 시스템에서 표준으로 채택되고 있다. 터보부호는 반복복호를 함으로써 우수한 성능을 얻을 수 있지만 반복복호 회수가 많아지면 지연시간과 연산 수가 증가하게 된다. 그러므로 반복복호 회수를 줄여주는 알고리즘이 요구되어지며, 그 방법 중에서 CRC를 이용하는 것이 가장 우수한 성능을 가지는 것으로 알려져 있다.

본 논문에서는 반복복호 회수를 줄이기 위해 CRC를 사용하여 터보부호를 구성할 경우 나타날 수 있는 edge-effect에 대해 분석하고 그 대책을 제시하였다. edge-effect는 프레임의 끝 부분에 위치한 비트가 인터리빙 된 후에도 프레임 끝 부분에 위치할 때 나타난다. 그러므로 기존의 s-랜덤 인터리버 설계 알고리즘에 D -파라메타를 도입하여 프레임 끝 부분의 비트가 다시 프레임 끝 부분으로 가는 것을 막아주도록 하였다. 모의 실험 결과 제안된 방법으로 구성된 인터리버는 D -파라메타 값을 적절히 선택하면 프레임의 끝 부분에 집중적으로 일어나던 오류를 감소시켜 edge-effect를 완전히 제거할 수 있음을 확인하였다. 또한 본 논문에서 제시된 D -파라메타는 CRC를 사용하지 않는 일반적인 터보부호에도 적용되어질 수 있으며 오류성능의 향상 효과는 동일하게 나타날 것으로 사료된다.

이러한 D -파라메타를 적용하여 인터리버를 구성하는 경우 D 값의 크기가 클수록 edge-effect의 영향이 감소하지만, 인터리버의 랜덤한 성질이 사라지게 되므로 가급적이면 작은 값으로 정하는 것이 바람직하다. 따라서 인터리버를 최적의 조건으로 만들기 위한 D 값의 정확한 경계값에 대한 연구가 더 필요할 것으로 사료된다.

참 고 문 헌

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, Vol. 20, pp. 1064~1070, May 1993.
- [2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, Vol. 44, No. 10, pp. 1261~1271, Oct. 1996.
- [3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, Vol. 42, pp. 409~428, Mar. 1996.
- [4] J. Hagenauer, "The turbo principle: tutorial introduction and state of the art," *Proc. Int. Symp. Turbo Codes and Related Topics*, Brest, France, pp. 1~11, Sep. 1997.
- [5] 배성일, 김재석, "효율적 저전력 터보 복호기," 대한전자공학회 춘계학술대회 논문집, 제22권, 제1호, pp. 73~76, 1999년 6월
- [6] P. Robertson, "Illuminating the structure of parallel concatenated recursive systematic (turbo) codes," *Proc. IEEE GLOBECOM'94*, San Francisco, USA, pp. 1298~1303, Dec. 1994.
- [7] A. Shitbutani, H. Suda, and F. Adachi, "Reducing average number of turbo decoding iterations," *Electron. Lett.*, Vol. 35, No. 9, pp. 701~702, Apr. 1999.
- [8] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," *TDA Progress Report*, JPL, Pasadena, USA, Vol. 42, pp. 56~65, Aug. 1995.

- [9] B.G Lee, S.J. Bae, S.G. Kang, and E.K. Joo
"Design of swap interleaver for turbo codes,"
Electron. Lett., Vol. 35, No. 22, pp. 1939~1940,
Oct. 1999.
- [10] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv,
"Optimal decoding of linear code for minizing
symbol error rate," *IEEE Trans. Inform.
Theory*, Vol. 20, No. 2, pp. 284~287, Mar.
1974.
- [11] S.S. Pietrobon and A.S. Barbuлесcu, "A
simplification of the modified Bahl decoding
algorithm for systematic convolutional codes,"
Proc. IEEE ISITA'94, Sydney, Australia, pp.
1073~1077, Nov. 1994.
- [12] P. Robertson, E. Villerun, and P. Hoeher, "A
comparison of optimal and sub-optimal MAP
decoding algorithms operating in log domain,"
Proc. IEEE Int. Conf. Commun., Seattle, USA,
pp. 1009~1013, June 1995.

저 자 소 개



李秉吉(正會員)

1997년 2월 : 경북대학교 전자공학과
졸업(공학사). 1999년 2월 : 경북대학
교 전자공학과 석사(M.S.). 1999년 3
월~현재 : 경북대학교 전자공학과
박사과정. <주관심분야 : 터보부호,
이동통신, IMT-2000>



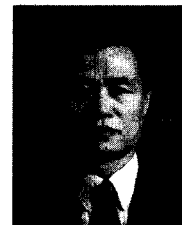
鄭建鉉(正會員)

1998년 2월 : 경북대학교 전자공학과
졸업(공학사). 2000년 2월 : 경북대학
교 전자공학과 석사(M.S.). 2000년 2
월~현재 : 삼성전자 무선사업부 선
임연구원. <주관심분야 : 디지털통신,
통신시스템>



裴商宰(正會員)

1993년 2월 : 경북대학교 전자공학과
졸업(공학사). 1995년 2월 : 경북대학
교 전자공학과 석사(M.S.). 1995년 3
월~현재 : 경북대학교 전자공학과
박사과정. <주관심분야 : 터보부호, 다
중레벨 부호변조, 이동통신, WCDMA>



朱彦暎(終身會員)

1976년 : 서울대학교 공과대학 전자
공학과 졸업(공학사). 1984년 : 미국
Ohio State University 전기공학과
졸업(M.S.). 1987년 : 미국 Ohio
State University 전기공학과 졸업
(Ph.D.). 1976년~1979년 : 해군통신
전자 기술장교. 1979년~1982년 : 한국과학기술원 연구
원. 1987년~현재 : 경북대학교 공과대학 전자전기공학
부 재직(교수). <주관심분야 : Digital Communication
Systems, Coding and Decoding, Modulation and
Demodulation, Digital Signal Processing for
Communications>