

# 동영상 전송을 위한 RCPT 코드의 설계

정회원 이창우\*

## RCPT Code Design for Video Transmission

Chang Woo Lee\* *Regular Member*

### 요약

무선 채널 상으로 동영상 신호를 전송하는 경우 채널 상의 잡음과 간섭 신호 등이 압축된 동영상 신호를 크게 왜곡시키기 때문에 무선 전송 상의 오류에 강인한 영상 전송 기법 개발이 필수적이다. 본 논문에서는 오류 정정 부호로 우수한 성능을 보이는 turbo 코드를 압축된 동영상 신호의 안정적인 전송에 적용하기 위해서 동영상 신호 전송에 맞는 RCPT(rate compatible punctured turbo) 코드를 설계하고 이의 성능을 해석하였다. 특히 RCPT 코드는 동영상 신호의 UEP(unequal error protection) 적용에 유용하다. 모의 실험 결과 turbo 코드의 인터리버(interleaver)의 크기가 크지 않을 때는 helical 인터리버를 사용한 turbo 코드가 우수한 성능을 보이는 것을 확인하였다.

### ABSTRACT

It is essential to develop the robust video transmission system over wireless channel, since the effect of the noise and the interference to compressed video may be fatal. In this paper, we have designed turbo code, which is adequate for the transmission of video and analyzed the performance of that. Especially, we have focused the rate compatible punctured turbo(RCPT) code, which has rate compatible(RC) property for unequal error protection(UEP). It has been shown through computer simulation that the turbo code with helical interleaver yields better performance than those with other interleavers when the size of the interleaver is not large.

### 1. 서론

최근 표준화 논의가 활발히 진행되고 있는 차세대 이동통신 시스템인 IMT-2000(International Mobile Telecommunication-2000) 시스템에서는 본격적인 멀티미디어 통신 서비스를 제공하기 때문에 기존의 개인 휴대통신 시스템에 비해서 전송해야할 정보량이 크게 늘어나게 된다<sup>[1]</sup>. 그러나 전송률이 커지면 무선 채널 상에서 발생하는 왜곡이나 잡음 등이 전송되는 정보에 크게 영향을 미치게 된다. 특히 동영상 신호는 시간축 상의 상관도를 이용하여 압축되기 때문에 전송 중에 오류가 발생하면 해당 프레임 뿐만 아니라 연속되는 프레임에 계속 영향을 미치

게 되어 수신측에서 재생되는 화질이 크게 저하되게 된다<sup>[2,3]</sup>. 이에 대한 대처 방법으로는 오류에 강인한 영상 부호화 기법을 개발하는 방법이나 오류 은폐(error concealment) 기법을 사용하는 방법 또는 압축된 신호에 오류정정 부호를 적용하여 수신측에서 오류를 보정할 수 있도록 하는 방법이 있다. 오류 정정 부호화 기법을 사용하는 경우에는 오류 정정 부호로 인해서 부가 정보가 증가하는 단점이 있는 반면에 무선 채널 상에서 발생하는 오류의 발생을 크게 줄일 수 있는 장점이 있다.

오류 정정 부호 중에서 Berrou 등이 1993년에 제안한 turbo 코드는 반복적 복호(iterative decoding) 기법을 사용하여 Shannon이 제시한 이론적 한계치에 근접하는 성능을 보이는 오류 정정 부호 기법이

\* 가톨릭대학교 컴퓨터·전자공학부 (lcw@www.cuk.ac.kr)  
논문번호: 00133-0422, 접수일자: 2000년 4월 22일

다<sup>[4-9]</sup>. 일반적으로 많이 사용되는 RSC (recursive systematic convolutional) turbo 코드는 두 개의 RSC 부호기가 인터리버(interleaver)를 사이에 두고 병렬로 연결된 형태를 갖는다. 이 때 turbo 코드의 성능은 사용되는 부호기와 인터리버에 따라서 결정된다. 부호기로는 RSC 부호기의 성능을 결정하는 최소 자유거리(minimum free distance)를 크게 하는 부호기가 좋은 성능을 보인다<sup>[6,7]</sup>. 또한 인터리버로는 random 인터리버가 많이 사용되는데 일반적으로 인터리버의 크기를 크게 할수록 오류 정정 능력이 좋아진다<sup>[4-9]</sup>. 그런데 압축된 동영상 신호에 turbo 코드를 적용하는 경우에는 복호시의 시간 지연 등으로 인터리버의 크기를 일정한 크기 이하로 제한해야 할 필요가 있다<sup>[8]</sup>. 또한 기본적인 turbo 코드의 형태는 원래의 정보에 두 개의 부호화된 parity 비트를 첨가하여 부호화율을 1/3로 하거나 두 parity 비트 중에서 1 비트를 교대로 펀처링(puncturing)하여 부호화율을 1/2로 하는 turbo 코드가 많이 연구되었다<sup>[4-9]</sup>. 그러나 이러한 부호화율을 갖는 turbo 코드는 위성 통신 등에서는 같이 충분한 대역폭을 확보할 수 있는 경우에는 문제가 없지만 이동 통신 환경과 같이 대역폭이 제한되어 있을 경우에는 문제가 된다. 따라서 turbo 코드의 부호화율을 낮추는 것이 필수적인데 turbo 코드의 부호화율을 낮추는 방법으로 펀처링 기법이 많이 사용된다<sup>[10]</sup>. 한편 압축된 영상 신호에 오류정정 부호를 효과적으로 적용하기 위해서 많이 연구되고 있는 기법 중에서 중요한 정보와 중요하지 않은 정보의 오류정정 부호화율에 차등을 두어서 오류 정정 부호를 적용하는 UEP(unequal error protection) 기법이 효율적이다. 이를 위해서 RSC 부호기의 부호화율을 효율적으로 조절할 수 있는 RCPC(rate compatible punctured convolutional) 기법<sup>[10]</sup> 등이 제안되었고 turbo 코드에도 이를 적용하려는 연구가 있었다<sup>[8]</sup>.

본 논문에서는 압축된 동영상 신호의 안정적인 전송을 위하여 오류 정정 부호로 우수한 성능을 보이는 turbo 코드를 적용하기 위해서 동영상 신호 전송에 맞는 turbo 코드를 설계하고 이의 성능을 해석하였다. 이를 위하여 turbo 코드는 정보의 중요도를 고려한 오류 정정 부호의 적용을 위한 UEP의 적용을 위해서 RC(rate compatible) 특성을 갖도록 하였다. 이러한 RCPT 코드에 대하여 turbo 코드의 성능에 큰 영향을 미치는 인터리버의 선택을 위하여 random 인터리버, S-random 인터리버, helical 인터리버 등의 성능을 해석하였고 동영상 신호의 복호

시 시간 지연을 최소화하기 위해서 인터리버의 크기를 500 이하로 제한하였다. 또한 turbo 코드의 성능에 큰 영향을 미치는 RSC 부호기를 입력 비트열 중에서 1이 각각 2 개와 3 개 포함된 weight 2, weight 3 입력 시퀀스에 대한 최소 자유거리가 최대화되도록 설계하였다. 다음에 이러한 방법으로 설계된 turbo 코드에 대해서 RC 조건에 맞는 펀처링 행렬을 사용하여 동영상 전송에 맞도록 다양한 부호화율을 구현하고 Monte Carlo 시뮬레이션을 행하여 RCPT 코드의 성능을 해석하였다.

## II. RCPT (rate compatible punctured turbo) 코드

오류 정정 부호 중에서 Berrou 등이 1993년에 제안한 turbo 코드는 반복적 복호(iterative decoding) 기법을 사용하여 Shannon이 제시한 이론적 한계치에 근접하는 성능을 보이는 오류 정정 부호 기법이다<sup>[4,5]</sup>. Turbo 코드는 그 성능의 우수함으로 인해서 많은 주목을 받고 있는데 turbo 코드를 구현하기 위한 부호기와 복호기의 블록도를 그림 1과 그림 2에 각각 제시하였다.

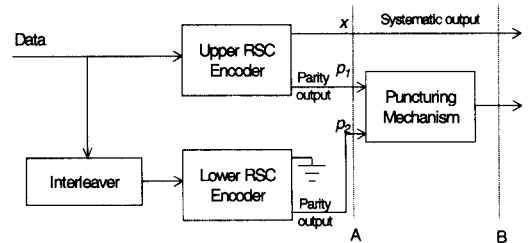


그림 1. Turbo 코드의 부호기

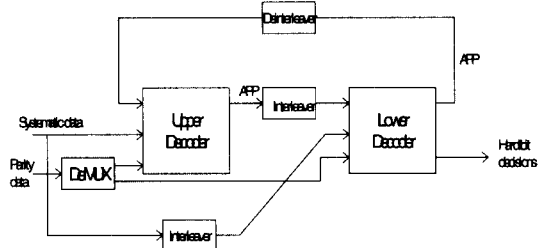


그림 2. Turbo 코드의 복호기

Turbo 코드가 이론적 한계치에 근접하는 성능을 보이는 것은 그림 2에서 제시하는 복호기에서 upper 복호기(decoder)와 lower 복호기의 반복적 복호를 통해서 오류 정정 능력을 계속 높이는데 있다.

Upper 복호기와 lower 복호기는 각각 복호의 결과인 APP(a posteriori probability)를 soft decision 형태로 전달하고 채널에서 각각에 입력되는 systematic data와 APP는 인터리버를 거쳐서 각 복호기에 전달된다. 따라서 각 복호기에 입력되는 정보는 인터리버에 의해서 재배치된 정보이므로 채널에 대한 새로운 정보라고 해석할 수 있기 때문에 반복적 복호를 통해서 성능을 계속 높일 수 있다. Turbo 코드는 오류 정정 능력이 Shannon의 이론적인 한계치에 근접할 정도로 뛰어나기 때문에 압축된 영상 정보를 무선 채널을 통해 전송할 때 오류 정정 기법으로 적용하면 전체 시스템의 성능을 높이는데 크게 기여할 수 있을 것으로 판단된다.

그림 1에 제시한 turbo 부호기는 RSC(recursive systematic convolutional) 부호기를 사용한 turbo 코드인데 전송하려는 data  $x$ 와 두 개의 RSC 부호기에서 생성하는 parity 비트인  $p_1, p_2$ 로 인해서 부호화율은 A 지점에서 1/3이다. 그런데 부호화율이 1/3인 경우 전송률이 원 신호만을 전송하는 경우에 비해 정보량이 3배가 되기 때문에 parity 비트 중에서 일부만을 전송하는 기법이 많이 사용된다. 이를 parity 비트를 평처링한다고 하는데 특히  $p_1, p_2$  2비트 중에서 1비트만을 교대로 전송하는 방법이 많이 사용된다. 이 경우에는 B 지점에서 부호화율이 1/2로 되어서 전송률을 줄일 수 있지만 영상부호화에 적용하기에는 전송률도 크고 임의의 전송률을 구현하기 어려운 단점이 있다. 따라서 turbo 코드가 영상 정보의 오류 정정 부호로 활용되기 위해서는 전송률을 일반화하기 위한 일반적인 평처링 기법이 연구되어야 한다. 이를 위해서는 그림 1의 평처링 방법에 대한 연구가 필요한데 특히 평처링 기법 중에서 정보의 중요도에 따라서 차등을 두어서 오류 정정 부호화를 하는 UEP(unequal error protection)을 하기 위해서는 낮은 부호화율을 사용할 때의 parity 비트가 높은 부호화율을 사용할 때의 parity 비트를 포함하는 RC(rate compatible) 특성을 만족하여야 한다. 이를 위하여 그림 1에서 정보  $x$ 와 잉여 정보  $p_1, p_2$ 가 전송되어 부호화율이 1/3이고 평처링 주기가  $T$ 일 때  $p_1, p_2$ 를 평처링하기 위한  $3 \times T$  평처링 행렬  $P$ 를 다음 식 (1)과 같이 정의하자<sup>[10]</sup>.

$$P = (p_{ij}(l)), \quad p_{ij}(l) \in (0, 1) \quad (1)$$

식 (1)에서  $p_{ij}(l)$ 은 평처링 행렬의 각 원을 나타

내는 것으로서 0일 때가 해당 parity 비트가 평처링되는 것을 나타낸다. 또한  $l$ 이 커질수록 평처링되는 비트가 줄어들어서 부호화율이 작아짐을 의미한다. 이때 RC 조건을 만족하려면  $p_{ij}(l)$ 이 다음의 조건을 만족하여야 한다<sup>[10]</sup>.

$$\text{If } p_{ij}(l_0) = 1, \text{ then } p_{ij}(l) = 1, \text{ for all } l \geq l_0 \geq 1. \quad (2)$$

식 (1), (2)의 RC 조건을 만족하는 평처링 행렬 중에서 본 논문의 성능 분석에서 사용한 주기 8의 평처링 행렬  $P$ 를 표 1에 제시하였다.

표 1. RC 조건을 만족하는 평처링 행렬  $P$

| $l$ , 부호화율 | 평처링 행렬  |
|------------|---|
| 1, 4/5     | $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ |
| 2, 2/3     | $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ |
| 3, 4/7     | $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ |
| 4, 1/2     | $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$ |
| 5, 1/3     | $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ |

이러한 turbo 코드의 성능을 결정하는 것은 그림 1에서 도시한 turbo 코드의 부호기에서 사용되는 두 개의 RSC 부호기와 인터리버이다. 부호기와 인터리버에 따르는 turbo 코드의 성능 변화에 대해서는 지금까지 많은 연구 결과들이 발표되었다<sup>[6,9,11-13]</sup>. 본 논문에서는 동영상 전송 시스템에서 복호 시간에 따르는 시간 지연을 최소화하기 위해서 인터리버의 크기를 제한하였을 때 RSC 부호기를 설계하고 여러 가지 인터리버를 사용하였을 때의 성능을 3장에서 해석한다.

Turbo 코드의 복호를 위해서는 일반적으로 MAP(maximum a posteriori)에 근거한 BCJR(Bahl, Cocke, Jelinek and Rajiv) 알고리즘<sup>[14]</sup>이나 Viterbi 알고리즘의 출력을 soft output 형태로 고친 SOVA(soft output Viterbi algorithm)<sup>[15]</sup> 등이 많이 사용된다. 그런데 BCJR 알고리즘의 계산량은 SOVA에 비해서 많은 반면에 성능은 가장 좋은 것으로 알려져 있기 때문에 본 논문에서는 turbo 코드

를 복호하는데 BCJR 알고리즘을 사용하였다. BCJR 알고리즘을 사용한 turbo 코드의 복호는 인터리버의 크기에 해당하는 block에 대해서 그림 2의 upper 복호기와 lower 복호기의 반복적인 복호에 의해서 비트를 검출해내는 방법을 사용하기 때문에 기본적으로 block 코드라고 볼 수 있다.

### III. RCPT 코드의 설계

RCPT 코드의 성능은 그림 1의 turbo 부호기에서 병렬로 연결된 두 개의 RSC 부호기, 인터리버와 parity 비트를 펼쳐링하기 위한 펼쳐링 방법이 결정한다. 만일 RCPT 코드의 부호화율이 정해져 있어서 부호화율에 맞는 펼쳐링 방법을 사용해야 한다면 그 성능은 사용되는 RSC 부호기와 인터리버에 전적으로 의존하게 된다. 먼저 동일한 2 개의 RSC 부호기를 사용한다고 가정하면 부호기 trellis의 최소 자유거리, 특히 입력 비트열 중에서 1이 2 개 포함된 weight 2 시퀀스와 1이 3 개 포함된 weight 3 시퀀스에 대한 최소 자유거리를 최대화하는 부호기가 최적으로 알려져 있다<sup>[11]</sup>. 한편 인터리버에 대해서는 일반적으로 인터리버의 크기가 커질수록 turbo 코드의 오류 정정 능력이 높아지게 되는데 인터리버를 크게 할수록 turbo 복호기가 복잡해지고 복호 시간이 오래 걸리는 단점이 있다. 특히 영상 신호의 경우 복호 지연으로 인한 실시간 처리에 영향을 받을 수 있다.

Turbo 코드의 인터리버로는 보통 입력 비트열을 불규칙하게 재배치하여 두 RSC 복호기에 입력되는 비트열의 상관 관계를 최소화하는 random 인터리버가 많이 사용되고 비교적 좋은 성능을 보이는 것으로 알려져 있다. 한편 random 인터리버의 불규칙한 특성을 유지하면서 두 RSC 부호기를 모두 고려한 전체 부호기 trellis의 최소 자유거리를 최대로 하기 위해서 random 인터리버에 제한조건을 가하는 S-random 인터리버가 SNR(signal to noise ratio)이 클 때 random 인터리버에 비해서 좋은 성능을 보인다는 사실이 알려져 있다<sup>[6,16]</sup>. S-random 인터리버는 다음과 같이 설계할 수 있다. 즉, random number를 발생시킬 때마다 S 개의 이전에 선택된 정수와 비교한다. 만일 현재 선택된 정수와 이전에 선택된 S 개의 정수와의 차이가 S 이내라면 현재 선택된 정수를 버리고 다른 random number를 발생시켜서 이전의 S 개의 정수와의 차이가 S 이상이 되도록 한다. 이 과정을 인터리버의 설계가 완료될

때까지 반복하여 설계를 완료한다. 만일 인터리버의 크기를 N이라 하고  $S < \sqrt{N/2}$ 의 조건을 만족하게 하면 이러한 S-random 인터리버는 제한된 시간 안에 설계 가능하다는 사실이 알려져 있다<sup>[6]</sup>.

그런데 turbo 코드는 부호기 trellis 종료 문제를 고려해야 한다. 만일 RSC 부호기에 v 비트의 기억 장소가 사용되었다면 trellis 종료를 위해서 v 비트를 추가로 부가하여야 trellis를 종료할 수 있다. 그런데 하나의 RSC 부호기의 trellis가 종료되었다고 하더라도 인터리버에 의해서 비트열이 재배치되게 되면 다른 부호기에서는 trellis 종료를 보장할 수 없다. 이 문제를 해결하기 위해서 helical 인터리버가 제안되었다<sup>[8]</sup>. 두 RSC 부호기의 trellis를 동시에 종료시키려면 부호기의 최종 state가 두 비트열에 대해서 동일해야 하고, 이를 위하여 turbo 부호화기의 인터리버와 RSC 부호기에 모두 제한 조건이 가해진다. 보기로서 depth가 6이고 크기가 30인 helical 인터리버의 입,출력 비트열의 배치를 표 2와 표3에 나타내었다<sup>[8]</sup>. 또한 helical 인터리버와 함께 사용되는 RSC 부호기의 제한 조건으로는 모든 feedback 비트가 모두 부호기 입력에 더해져야 한다. 그런데 turbo 코드의 성능은 1이 2 개 포함된 weight 2 시퀀스와 1이 3 개 포함된 weight 3 시퀀스에 대한 최소 자유거리 d2, d3를 최대화하는 RSC 부호기가 최적으로 알려져 있다. 앞에서 서술한 것과 같이 RSC 부호기 중에서 feedback 비트가 모두 부호기 입력에 더해지는 제한 조건을 만족하면서 d2, d3가 최대화하는 부호기를 설계하여 helical 인터리버와 같이 사용하였다. 설계한 부호기를 그림 3에 도시하였고 이러한 부호기의 weight 2 시퀀스에 대한 최소의 parity 수를 표 4에 제시하였다.

표 2. Helical 인터리버의 입력의 보기

|     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|
| x1  | x2  | x3  | x4  | x5  | x6  |
| x7  | x8  | x9  | x10 | x11 | x12 |
| x13 | x14 | x15 | x16 | x17 | x18 |
| x19 | x20 | x21 | x22 | x23 | x24 |
| x25 | x26 | x27 | x28 | x29 | x30 |

표 3. Helical 인터리버의 출력

|    |    |    |    |   |    |    |    |   |   |    |    |    |   |   |    |    |    |   |   |     |
|----|----|----|----|---|----|----|----|---|---|----|----|----|---|---|----|----|----|---|---|-----|
| x  | x  | x  | x  | x | x  | x  | x  | x | x | x  | x  | x  | x | x | x  | x  | x  | x | x | ... |
| 25 | 20 | 15 | 10 | 5 | 30 | 19 | 14 | 9 | 4 | 29 | 24 | 13 | 8 | 3 | 28 | 23 | 18 | 7 | 2 |     |

Helical 인터리버를 사용하면 두 부호기 trellis를 모두 종료시킬 수 있기 때문에 turbo 코드의 복호에

사용되는 BCJR 알고리즘을 적용하는데 있어서 최종 state를 모르는 경우보다 부호화 이득이 있고 trellis 종료에 따르는 최소 자유거리가 늘어나기 때문에 parity 비트의 평처링으로 인한 영향을 덜 받는다. 다음 4장에서 제시하는 모의 실험 결과에서 helical 인터리버에 대해서는 본 논문에서 설계한 그림 3의 부호기를 사용하였고 random 인터리버와 S-random 인터리버에 대한 부호기로는 [9]에서 제시하는 최적의 부호기를 사용하였다. Random 인터리버에서 사용하는 부호기를 그림 4에 도시하였다. Helical 인터리버에 대한 부호기와 random 인터리버에 대한 부호기를 다르게 한 것은 helical 인터리버에서 종료 조건을 만족하려면 앞에서 서술한 제한 조건이 부호기에 가해져야 하기 때문이다. 다음에 turbo 코드의 parity 비트의 평처링을 위해서는 2장의 표 1에서 제시한 RC 조건을 만족하는 평처링 행렬을 사용하였다.

#### IV. 모의 실험

제 3장에서 살펴본 것과 같이 turbo 코드의 성능은 인터리버의 크기와 종류, 사용된 부호기와 평처링 행렬에 의해서 결정된다. 그런데 RCPT 코드를 구현하기 위해서는 표 1에서 도시한 것과 같은 RC 조건을 만족하는 평처링 행렬을 사용해야 한다. 또한 동영상 복호시의 복잡도와 복호 지연 등을 고려하면 인터리버의 크기는 500 이하가 적당하다. 이러한 사항들을 고려하여 turbo 코드의 성능을 측정하기 위한 모의 실험에서는 인터리버의 크기를 360으로 제한하였고 표 1에서 제시한 평처링 행렬을 사용하여 부호화율 1/3~4/5의 turbo 코드에 대한 성능을 측정하였다.

부호화율과 인터리버의 크기가 정해지면 사용되는 인터리버의 종류와 RSC 부호기가 turbo 코드의 성능을 결정하게 된다. 먼저 인터리버에는 3장에서 설명한 것과 같이 일반적으로 많이 사용되는 random 인터리버, S가 10인 S-random 인터리버와 두 부호기 trellis를 모두 종료할 수 있는 helical 인터리버를 성능 평가 대상으로 하였다. 또한 각각의 인터리버에 최적화된 RSC 부호기, 즉 그림 3과 그림 4에 제시된 부호기를 각각의 인터리버와 함께 사용하였다. 각각의 turbo 부호기로 부호화된 코드가 AWGN (additive white Gaussian) 채널을 통과할 때 비트 오류 발생 확률  $10^{-2} \sim 10^{-5}$ 가 되는  $E_b/N_0$  구간을 측정하였다. 이때 turbo 코드의 복호에는 BCJR 알고리즘<sup>[14]</sup>을 사용하였고 복호기에서 15회의 반복적 복호를 통하여 비트를 검출한다고 가정하였다.

모의 실험 결과를 그림 5에 도시하였는데, 그림에서 알 수 있듯이 부호화율이 1/3일 때는 각 인터리버 간의 성능 차이는 크지 않다. 그러나 부호화율이 증가할수록, 즉 turbo 코드에서 평처링되는 parity 비트들이 많아질수록 helical 인터리버를 사용한 turbo 코드의 성능이 random 혹은 S-random 인터리버를 사용하였을 때보다 좋아지는 것을 알 수 있다. 특히 부호화율이 4/5일 때는 helical 인터리버를 사용하였을 때의 비트 오류 발생 확률이 S-random 인터리버 혹은 random 인터리버를 사용하였을 때의 약 절반 이하인 것을 알 수 있다. 이는 helical 인터리버를 사용하면 두 부호기 trellis를 모두 종료시킬 수 있으므로 BCJR 알고리즘에서 부호화 이득이 발생하고 최소 자유거리가 늘어남으로써 parity 비트의 평처링으로 인한 영향을 덜 받기 때문인 것으로

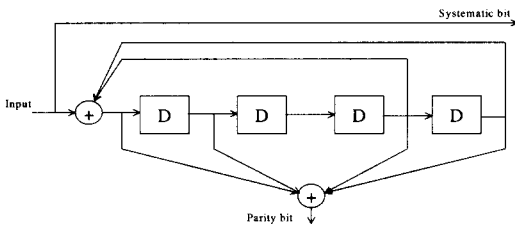


그림 3. Helical 인터리버에 최적인 RSC 부호기  $((D4+D2+1)/(D4+D3+D2+D+1))$

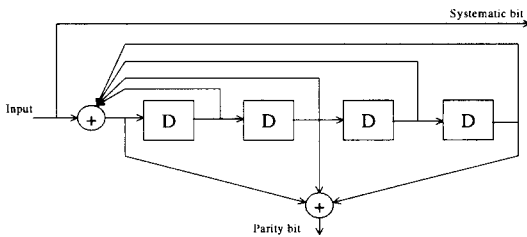
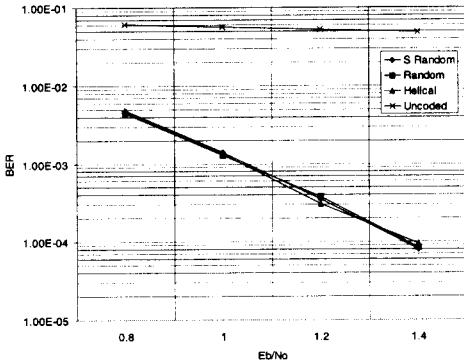


그림 4. Random 인터리버에 최적인 RSC 부호기  $((D4+D3+D+1)/(D4+D3+1))$

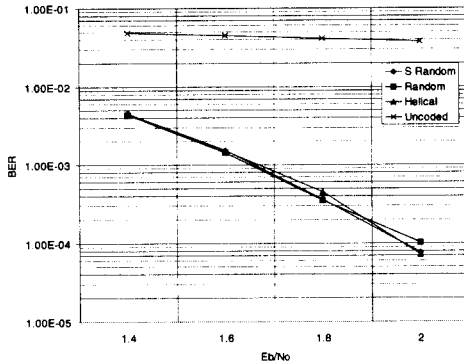
표 4.  $(D4+D2+1)/(D4+D3+D2+D+1)$  RSC 부호기의 weight 2 사이클스의 최소의 parity 수

| 부호화율   | $(D4+D2+1)/(D4+D3+D2+D+1)$ RSC 부호기의 weight 2 사이클스의 최소의 parity 수 |
|--------|---|
| 5, 1/3 | 6   |
| 4, 1/2 | 3   |
| 2, 2/3 | 1   |
| 1, 4/5 | 0   |

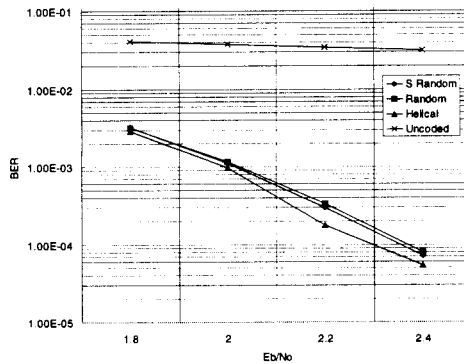
이해할 수 있다. 특히 본 논문에서와 같이 인터리버의 크기를 제한해야 하는 경우라면 인터리버의 크기가 큰 경우보다 trellis 종류에 따르는 성능 향상이 크다. 또한 random 인터리버와 S-random 인터리버를 사용한 경우를 비교해 보면 S-random 인터리버를 사용한 경우의 성능이 우수함을 알 수 있는데 그 차이는 helical 인터리버를 사용했을 때와 S-random 인터리버를 사용했을 때의 성능 차이보다 크지 않음을 알 수 있다.



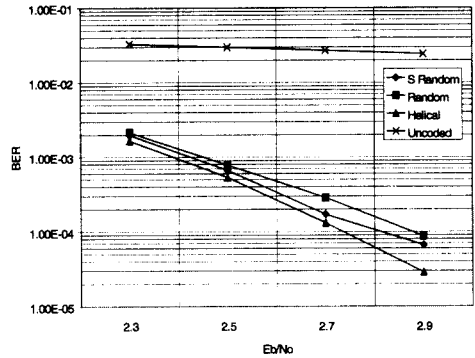
(a)



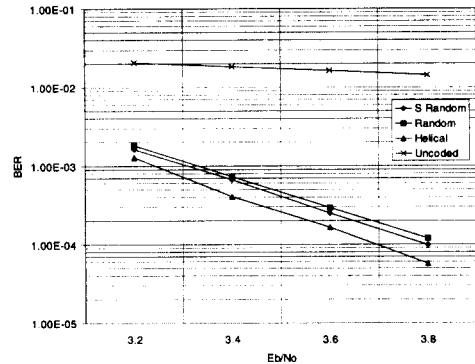
(b)



(c)



(d)



(e)

그림 5. Turbo 코드의 BER(bit error rate) 성능  
 (a)  $l = 5$  (부호화율: 1/3) (b)  $l = 4$  (부호화율: 1/2)  
 (c)  $l = 3$  (부호화율: 4/7) (d)  $l = 2$  (부호화율: 2/3)  
 (e)  $l = 1$  (부호화율: 4/5)

## V. 결론

최근에 동영상 정보를 효율적으로 전송하기 위한 다양한 구현 방법이 제시되고 있는데 영상 부호화 측면에서 전송 오류에 강인한 부호화 기법을 개발하거나 오류 정정 기법을 동영상 전송에 효율적으로 적용하는 방법 등이 주로 연구되고 있다. 또한 최근에는 이 두 방법을 동시에 최적화하려는 시도도 활발히 연구되고 있다. 이러한 방법 중에서 오류 정정 부호를 사용하는 기법은 부가 정보가 늘어나는 단점은 있지만 비트 오류를 크게 줄일 수 있는 장점이 있기 때문에 이동 통신 환경과 같이 전송되는 채널의 상태가 좋지 않은 경우에는 필수적으로 사용된다.

본 논문에서는 동영상 전송에 적용하기 위해서 오류 정정 부호 기법 중에서 우수한 성능으로 최근에 크게 주목을 받고 있는 turbo 코드의 성능을 분

