가변 크기 인터리버를 사용한 turbo 부호의 설계와 성능 해석

정회원 이 창 우*

Design and performance analysis of turbo codes employing the variable-sized interleaver

Chang-Woo Lee* *Regular member*

요 약

이동 통신에 대한 수요가 크게 늘어나면서 멀티미디어 정보를 포함한 대용량의 정보를 이동 통신 채널 상으로 전송하기 위해서는 전송 오류에 대한 대책이 필수적이다. 전송 오류에 대한 대책 중에서 오류 정정 부호화 기법이 많이 사용되는데 turbo 부호는 이론적 한계치에 근접하는 우수한 성능을 보이는 오류 정정 부호화 기법이다. 본 논문에서는 멀티미디어 정보의 특성을 고려하여 입력되는 데이터의 크기가 가변인 경우 turbo 부호를 효율적으로 적용하기 위해서 3GPP 표준에서 사용되는 turbo 부호의 인터리버를 포함한 여러 가지 가변 크기 인터리버의 특성을 해석하고 AWGN 환경에서 그 성능을 분석하였다. 특히 인터리버의 불규칙도와 s-parameter가 성능에 미치는 영향을 분석하고 블록 크기와 오류 정정 부호화율이 변할 때 turbo 부호의 성능 변화를 해석함으로써 가변 크기 인터리버를 사용한 turbo 부호의 성능을 분석하고 멀티미디어 정보의 오류 정정 부호 기법으로 turbo 부호가 효율적으로 사용될 수 있음을 입증하였다.

ABSTRACT

With the advent of future mobile communication systems, the wireless transmission of the huge amount of multimedia data over the error-prone multipath fading channel has to overcome the inherent sensitivity to channel errors. To alleviate the effect of the channel errors, hosts of techniques based on the forward error correction(FEC) has been proposed at the cost of overhead rate. Among the FEC techniques, turbo code, whose performance has been shown to be very close to the Shannon limit, can be classified as a block-based error correction code. In this paper, considering the variable packet size of the multimedia data, we analyzed turbo codes employing the variable-sized interleaver. The effect of the various parameters on the BER performance is analyzed. We show that the turbo codes can be used as efficient error correction codes of multimedia data.

Ⅰ. 서 론

차세대 이동통신 시스템에서는 본격적인 멀티미디어 통신 서비스를 제공하기 때문에 전송해야 할 정보량이 크게 늘어나게 된다[1]. 특히 압축된 동영상 신호나 음성 신호는 전송 오류에 민감하기 때문에 이를 안정적으로 전송하기 위해서는 전송 오류에 강인한 무선 통신 시스템을 구현하는 것이 필수

* 가톨릭대학교 컴퓨터 · 전자공학부(changwoo@catholic.ac.kr)

논문번호: 020388-0907, 접수일자: 2002년 9월 7일

※ 본 연구는 2001년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

적인데 전송 오류에 강인하게 하기 위한 한가지 방법으로 오류 정정 부호화 기법이 많이 사용된다. 오류 정정 부호화 기법은 오류 정정 부호의 부가로 인해서 전체 전송률이 늘어나는 단점은 있지만 전송 오류를 크게 줄일 수 있게 된다[2-5].

오류 정정 부호 중에서 Berrou 등이 1993년에 제안한 turbo 부호[6-8]는 그 성능이 Shannon이 제시한 이론적 한계치에 근접하는 오류 정정 부호 기법으로서 그 성능의 우수함으로 인해서 많은 주목을 받고 있다. 특히 turbo 부호는 최근 상용화 연구가 활발히 진행되고 있는 3GPP(3rd Generation Partnership Project) 표준과 3GPP2(3rd Generation Partnership Project 2) 표준의 제3세대 이동통신 시스템의 오류 정정 부호화 기법 중의 하나로 채택되었는데 보통 32kbps 이상의 데이터 전송에 사용된다[9,10]. Turbo 부호는 두 개의 부호화기와 인터리버를 사용하여 부호화되는데 입력되는 비트열을 일정한 크기의 블록으로 나누어서 부호화하는 블록 부호의 일종이라고 볼 수 있다. 그런데 오류 정정 능력이 뛰어난 turbo 부호를 멀티미디어 정보에 사용하는 경우는 멀티미디어 정보의 특성을 고려해야 한다. 일반적으로 멀티미디어 정보는 정보량이 많고 계층별로 부호화되는데 각 계층에서 일정 단위마다 크기가 변한다. 이를 고려하여 정보량이 많을 때 오류 정정 부호화율을 조절하여 전체 정보량을 원하는 수준으로 유지해야 하고 크기가 변하는 정보에 turbo 부호를 효율적으로 적용할 수 있어야 한다. 크기가 변하는 정보에 turbo 부호를 적용하기 위해서는 turbo 부호의 블록 크기를 가변으로 해야 하는데 turbo 부호의 블록 크기는 사용되는 인터리버에 의해서 결정된다. 따라서 turbo 부호의 블록 크기를 가변으로 하는 문제는 가변 크기 인터리버를 효율적으로 구현하는 문제로 귀착된다.

본 논문에서는 앞에서 지적한 문제를 고려하여 블록 크기를 가변으로 하는 turbo 부호에 대해서 연구하였다. 먼저 3GPP와 3GPP2에서 사용되는 인터리버를 포함하여 블록 인터리버와 JPL 인터리버 등 블록 크기를 가변으로 할 수 있는 여러 가지 인터리버의 특성과 구현 방법을 고찰하고 AWGN(additive white Gaussian noise) 환경에서 성능을 분석하였다. 특히 압축된 동영상이나 음성 신호 등의 정보량을 고려하여 전체 정보량을 적정 수준으로 유지할 수 있도록 turbo 부호에서 오류 정정 부호화율을 변화시키는 경우와 블록의 크기가 변할 때의 성능을 분석하였다. 이를 바탕으로 turbo 부호가 멀티미디어 정보의 효율적인 오류 정정 부호 기법으로 사용될 수 있음을 입증하였다.

Ⅱ. Turbo 부호

2.1 Turbo 부호화기

Turbo 부호를 구현하기 위한 부호화기와 복호화기의 블록도를 그림 1과 그림 2에 각각 제시하였다[6-8]. 그림 1에 제시한 부호화기는 RSC(recursive systematic convolutional) 부호를 사용한 turbo 부호인데 인터리버에 의해서 재배치된 입력 비트열이 RSC 부호화기 1과 RSC 부호화기 2에 의해서 각각 부호화된 후에 전송되게 된다. 이때 전송하려는 데이터 비트 x와 두 개의 RSC 부호기에서 생성하는 패리티 비트인 p1, p2로 인해서 부호화율은 A 지점에서 1/3이다. 그런데 부호화율이 1/3인 경우 전송률이 원 신호만을 전송하는 경우에 비해 정보량이 3 배가 되기 때문에 패리티 비트를 펑쳐링하게 되는데 turbo 부호가 동영상 정보 등 멀티미디어 정보의 오류 정정 부호로 활용되기 위해서는 전송률을 일반화하기 위한 일반적인 펑처링 기법이 필요하다. 또한 펑처링 기법 중에서 낮은 부호화율에서 전송되는 패리티 비트가 이보다 높은 부호화율에서 전송되는 패리티 비트를 반드시 포함하는 RC(rate compatible)의 특성을 만족하는 것이 UEP(unequal error protection) 등의 적용에 적합하다[3]. RC 특성을 만족하는 펑처링 방법을 표 1에 정리하였는데 8/8부터 8/24까지의 8 단계 부호화율에 대해서 전송/비전송을 각각 1/0으로 하였을 때 16 진수를 사용하여 표현하였다. 3GPP 표준에서는 8 개의 스테이트를 갖는 동일한 RSC 부호화기를 그림 1의 RSC 부호화기1과 RSC 부호화기2로 사용한다. 3GPP에서 사용하는 RSC 부호화기를 그림 2에 도시하였다[9]. 그림 3의 복호기에서는 수신된 신호를 RSC 복호화기 1과 RSC 복호화기 2의 반복적 복호에 의해서 최종 신호를 검출하게 된다. Turbo 부호의 블록 크기는 그림 1과 그림2의 인터리버의 크기에 의해서 결정되기 때문에 turbo 부호의 블록의 크기가 가변이려면 그림 1과 그림 2에서 사용되는 인터리버(interleaver)의 크기가 가변이어야 한다.

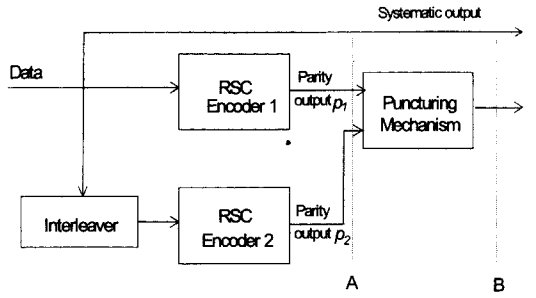


그림 1. Turbo 코드의 부호화기

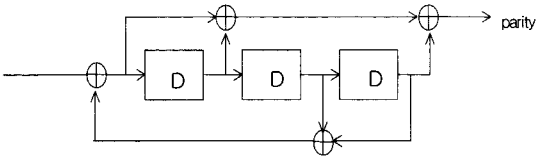


그림 2. 3GPP 표준의 RSC 부호화기

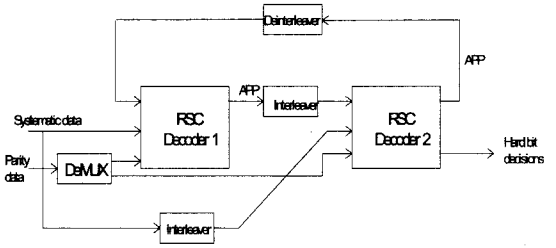


그림 3. Turbo 복호화기

표 1. Turbo 부호를 위한 펑처링

부호화율	8/8	8/10	8/11	8/12	8/14	8/16	8/20	8/24
x	FF	FF	FF	FF	FF	FF	FF	FF
p_1	00	80	88	88	AA	AA	FF	FF
p_2	00	40	40	44	44	55	55	FF

2.2 Turbo 부호의 복호 알고리즘

Turbo 부호를 복호하기 위해서는 그림 3의 각 복호기에서 MAP(maximum-a-posteriori) 복호 방법이 많이 사용된다. MAP 복호는 최적의 symbol-by-symbol decision 알고리즘으로서 soft decision 값을 출력하는데 이를 구현하기 위해서 BCJR 알고리즘이 많이 사용된다[11]. BCJR 알고리즘의 계산을 위해서는 곱하기와 exponential 계산이 필요한데 이러한 계산을 간단히 하기 위해서 Log-BCJR 알고리즘이 제안되었다. Log-BCJR 알고리즘을 요약하면 다음과 같다[12].

먼저 +1 혹은 -1 값을 갖는 N 비트의 정보 비트열 $\boldsymbol{u}$를 가정한다. 이러한 블록을 부호화하여 심볼 블록인 $\boldsymbol{c}$를 구성한다고 가정한다. 이러한 심볼열이 전송되어 수신측에서는 수신 블록 $\boldsymbol{y}$를 수신하였다고 가정하자. 만일 이진 trellis의 시간 k-1에서 시작 스테이트가 s'이고 시간 k에서 끝나는 스테이트가 s인 transition을 가정하면 Log-BCJR 알고리즘의 출력은 다음과 같은 APP(a posteriori probabilities)의 log-likelihood ratio이다.

$$\begin{aligned} L_k &= \log \frac{P(u_k=+1|y)}{P(u_k=-1|y)} \\ &= MAX*_{(s',s)}^{u_k=+1}(A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)) \\ &\quad - MAX*_{(s',s)}^{u_k=-1}(A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)) \end{aligned} \tag{1}$$

식 (1)에서 $A_k(s')$, $\Gamma_k(s',s)$, $B_k(s)$는 각각 forward state metric, backward state metric, branch metric에 $\log$를 취한 값이다. 이때 $MAX*(X, Y)$는 다음과 같은 Jacobi logarithm을 사용하여 계산할 수 있다.

$$\begin{aligned} MAX*(X, Y) &\equiv \log(e^X + e^Y) \\ &= MAX(X, Y) + \log(1+e^{-|X-Y|}) \end{aligned} \tag{2}$$

Trellis가 all-zero state에서 시작하고 끝난다면 forward state metric은 다음 식(3)과 같은 forward recursion에 의해서 계산할 수 있고 초기 조건은 식 (4)와 같이 주어진다.

$$A_k(s) = MAX*_{s'}(\Gamma_k(s',s) + A_{k-1}(s')), \quad k=1,\cdots,N-1 \tag{3}$$

$$A_0(s) = \begin{cases} 0, & \text{if } s=0 \\ -\infty, & otherwise \end{cases} \tag{4}$$

또한 backward state metric은 다음 식 (5)와 같은 backward recursion에 의해서 계산할 수 있고 초기 조건은 식 (6)과 같다.

$$B_{k-1}(s) = MAX*_{s}(\Gamma_k(s',s) + B_k(s)), \quad k=N,\cdots,2 \tag{5}$$

$$B_N(s) = \begin{cases} 0, & \text{if } s=0 \\ -\infty, & otherwise \end{cases} \tag{6}$$

위에서 설명한 Log-BCJR 알고리즘에서 식 (2)의 *MAX**의 값 중에서 *log* 값을 계산하기 위해서 다음과 같은 근사식이 사용될 수 있다[13].

$$\log(1+e^{-x}) \approx -ax+b, \quad 0<x<b/a \tag{7}$$

$$\log(1+e^{-x}) \approx \begin{cases} 0, & \text{if } x>\eta \\ c, & \text{if } x<\eta \end{cases} \tag{8}$$

식 (7)은 선형 보정(linear correction)으로 $a=0.3$, $b=\log(2)$의 값을 사용하고 식 (8)은 임계치 보정(threshold correction)으로 $c=\log(2)$, $\eta=1.0$ 혹은 $c=\frac{3}{8}$, $\eta=2.0$ 등의 값이 사용될 수 있다[12,13].

Ⅲ. 가변 크기 인터리버

Turbo 부호가 우수한 오류 정정 부호화 성능을 갖는 이유는 그림 1의 turbo 부호화기의 RSC encoder 1에서 부호화된 신호 중에서 낮은 부호화 가중치(encoded weight)를 갖는 비트열을 인터리버의 비트열 재배치를 통해서 RSC encoder2에서 부호화할 때는 부호화 가중치를 크게 할 수 있기 때문이다[14]. 따라서 turbo 부호의 인터리버는 출력의 비트열 배치가 입력과 다른 불규칙한 형태를 갖는 것이 입, 출력의 상관성을 제거하여 전체 성능을 높일 수 있게 된다. 이러한 특성을 만족시키는 랜덤(random) 인터리버가 많이 사용되고 성능도 좋은 것으로 알려져 있다. 그런데 랜덤 인터리버는 불규칙한 특성상 크기를 가변으로 하기 위해서는 모든 크기에 대한 인터리버가 부호화기와 복호화기에서 정의되어 있어야 하므로 구현이 매우 힘들다.

Turbo 부호에서 사용되는 인터리버 중에서 크기를 8의 정수 배로 쉽게 변화시킬 수 있는 인터리버로는 블록 인터리버(block interleaver)를 생각할 수 있다. 만일 인터리버의 i번째 출력이 $\pi(i)$번째 입력 값이라면 행 수가 M이고 열 수가 N인 블록 인터리버는 다음과 같은 입, 출력 관계를 갖는다[8].

$$\pi[i*M+j]=N*j+i, \quad (0\le i\le N-1, 0\le j\le M-1) \tag{9}$$

만일 입력이 byte-align되어 있어서 블록 크기가 8의 배수인 turbo 부호를 구현한다면 식 (9)에서 N을 8로 하면 된다. 그런데 블록 인터리버는 그 구조가 매우 규칙적이어서 입, 출력의 상관성을 제거하는 능력이 떨어지는 단점이 있기 때문에 랜덤 인터리버에 비해서 그 성능이 크게 낮아진다. 따라서 랜덤 인터리버와 같이 입, 출력의 상관성을 제거하는 특성을 갖고 있으면서 어느 정도의 규칙성을 갖고 있어서 인터리버의 크기가 변해도 쉽게 구현 가능한 인터리버가 가변 크기 turbo 부호에 적합하다고 볼 수 있다. 이러한 특성을 만족시키는 turbo 부호를 위한 인터리버 중에서 JPL 인터리버가 있는데 그 구현 방법을 부록에 정리하였다[8]. 또한 최근 실용화가 활발히 진행되고 있는 3GPP와 3GPP2 표준에서 사용되는 turbo 부호를 위한 인터리버 역시 그 크기를 가변으로 할 수 있는데 그 자세한 구현 방법은 3G 관련 문헌에 기술되어 있다[9,10].

부록에서 보이는 것과 같이 JPL 인터리버는 인터리버의 크기가 2의 배수이어야 하고 N과 M의 크기에 따라서 인터리버의 구조가 달라진다. 반면에 3GPP에서 사용하는 인터리버는 크기 40~5,114 사이의 임의의 값을 가질 수 있다. 만일 입력되는 블록이 byte-align되어 있어서 입력 블록의 크기가 8의 배수라면 JPL, 3GPP, 3GPP2 인터리버가 모두 가변으로 구현 가능하다. 앞으로 본 논문에서 제시하는 블록 인터리버와 JPL 인터리버에서는 N을 8로 가정하였다. 한편 구현상의 복잡도는 블록, JPL, 3GPP2, 3GPP 순으로 복잡하다. 특히 3GPP에서 사용하는 인터리버는 그 구현을 위해서는 비트 삽입 과정을 포함하여 행렬을 구성해야 하고 행렬의 행내(intra-row)와 행간(inter-row)의 각 비트의 순서를 정해진 규칙에 의해서 재배치 한 후에 그 결과 행렬을 이용하여 비트 제거 과정을 거친 후에 출력 비트를 생성해야 한다. 특히 입력되는 비트열의 크기에 따라서 행렬의 구조가 달라지고 행내와 행간의 비트 재배치를 위해서도 여러 표를 참조하는 것을 포함한 복잡한 과정을 거친다. 3GPP 인터리버를 구현하기 위한 가장 간단한 방법은 가능한 인터리버의 크기에 따르는 인터리버의 출력 값을 미리 계산하여 저장하는 것이다. 그러나 이 방법은 랜덤 인터리버의 구현 방법과 같이 메모리가 많이 소요되고 모든 크기의 인터리버에 대한 출력 값이 정의되어 있어야 하므로 인터리버의 크기를 임의로 변화시키는 가변 크기 인터리버의 구현에는 적합하지 않다. 3GPP 인터리버의 구현을 좀 더 간단하게 하기 위해서 전체 인터리버의 출력 값을 저장하지 않고 행내와 행간의 비트 재배치 패턴만을 저장하고 나머지 과정은 인터리버의 크기에 따라 계산하는

방법을 사용할 수 있다[15].

다음에 각 인터리버의 특성을 살펴보기 위해서 그림 4에 블록 크기가 4,096인 경우에 각 인터리버의 입, 출력 관계를 도시하였다. 그림에서 도시한 것과 같이 가변 크기로 구현 가능한 인터리버 중에서 3GPP 표준에서 사용하는 인터리버의 출력이 랜덤 인터리버와 가장 가까운 불규칙적인 특성을 보인다. 반면에 JPL 인터리버와 3GPP2 인터리버는 출력에 약간의 상관성을 보이는 것을 알 수 있다. 인터리버의 불규칙성을 정량적으로 측정하기 위한 지표로서 정규화된 산포도(dispersion)를 사용할 수 있다[8]. 먼저 인터리버의 변위 벡터(displacement vector)를 다음과 같이 정의한다.

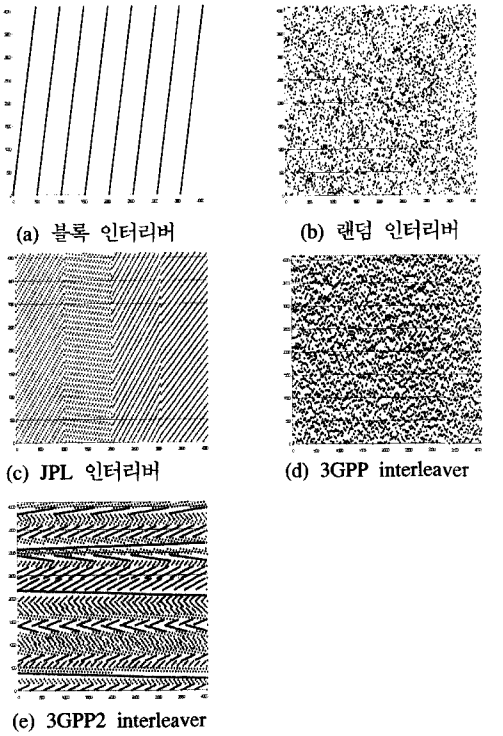
$$D_I=\{(\Delta x,\Delta y)|\Delta x=j-i,\Delta y=\pi(j)-\pi(i),0\le i<j<N\} \quad (10)$$

위 식에서 $\pi(i)$는 인터리버의 i 번째 출력을 의미하고 N은 인터리버의 크기이다. 변위 벡터가 가질 수 있는 원소의 수 $|D_I|$를 인터리버의 산포도 Γ_I라 정의한다. 이때 최대 산포도는 $\frac{N(N-1)}{2}$가 되고 다음 식 (11)과 같이 정규화된 산포도 γ가 인터리버의 불규칙성을 나타낼 수 있는 지표로 사용될 수 있다[8].

$$\gamma=\frac{2\Gamma_I}{N(N-1)} \quad (11)$$

그림 5에 각 인터리버에 대해서 정규화된 산포도를 블록 크기의 함수로 표시하였다. 그림에서 알 수 있듯이 불규칙성을 나타내는 정규화된 산포도는 랜덤 인터리버가 가장 크고 블록 인터리버가 가장 작다. 3GPP 인터리버의 경우 랜덤 인터리버에 근접하는 값을 보이고 JPL, 3GPP2 인터리버는 랜덤 인터리버와 블록 인터리버의 중간적인 특성을 보인다. 인터리버의 본래의 역할인 비트열 재배치를 통한 상관성 제거라는 측면에서 볼 때는 랜덤 인터리버와 3GPP 인터리버가 가장 좋은 특성을 보이는 것을 알 수 있다.

Turbo 부호의 성능을 저하시킬 수 있는 요소 중의 하나가 입력 비트열 중에서 1이 2 개 포함된 weight-2 자기 종료(self-terminating) 비트열이다[14]. 예를 들면 그림 2의 3GPP turbo 부호화기의 경우 (…0010010…)의 형태의 비트열이 weight-2 비트열 중에서 가장 작은 부호화 weight를 갖는 자기 종료 비트열인데 이러한 비트열이 인터리버를 통과한 후에도 다시 (…0010010…)의 형태를 갖는다면 RSC encoder1과 RSC encoder2를 고려한 전체 최소 자유 거리(free distance)가 작아져서 전체 성능을 떨어뜨릴 수 있다. 1이 3개 포함된 weight-3 이상의 비트열의 경우 인터리버가 불규칙한 특성을 갖는다면 같은 패턴의 비트열로 재배치될 확률은 크게 작아진다. 따라서 turbo 부호에서 사용되는 인터리버는 불규칙한 특성을 갖고 있으면서 인터리버의 입력으로 가까운 곳에 위치한 두 비트를 일정 거리 이상으로 분리하는 특성을 갖는 것이 좋은데 이를 측정하기 위한 도구로 s-parameter가 사용된다[8,14]. 만일 인터리버의 두 입력간의 거리가 s보다 작을 때 ($|i-j|<s$), 인터리버의 출력간의 거리가 항상 $|\pi(i)-\pi(j)|\ge t$를 만족하면 인터리버는 spreading factor (s,t)를 갖는다고 하고 s_m보다 작은 모든 s가 $s\le t$를 만족할 때 s_m을 인터리버의 s-parameter라 부른다. 예를 들면 s-parameter가 5라는 것은 인터리버의 두 입력 비트의 거리가 5 미만일 때 출력의 최소 거리는 입력의 비트 거리보다 1이상 큰 것을 의미한다. 그림 6에 turbo 부호의 블록 크기가 변할 때의 각 인터리버의 s-parameter



(a) 블록 인터리버 (b) 랜덤 인터리버

(c) JPL 인터리버 (d) 3GPP interleaver

(e) 3GPP2 interleaver

그림 4. 각 인터리버의 입, 출력 관계 (블록 크기 4,096)

를 도시하였다. 그림 6에서 알 수 있는 것과 같이

블록 인터리버는 모든 블록 크기에 대해서 s-parameter는 8로 일정한 값을 갖고 3GPP, 3GPP2, JPL 인터리버의 s-parameter 값은 블록 인터리버와 랜덤 인터리버의 중간 정도의 크기를 갖는다. 본 논문에서 사용한 랜덤 인터리버의 경우 s-parameter를 고려하지 않은 인터리버인데 [14]에서는 s-parameter 값을 일정한 값 이상으로 유지할 수 있는 s-랜덤 인터리버의 설계 방법을 제안하였다.

Ⅳ. 가변 크기 인터리버를 사용한 turbo 부호의 성능 해석

앞에서 설명한 가변 크기 인터리버를 사용한 turbo 부호를 Log-BCJR 알고리즘을 이용하여 복호하였을 때의 성능을 AWGN(additive white Gaussian noise) 채널에서 분석하였다. Log-BCJR 알고리즘에는 근사식을 사용할 수 있지만 본 논문에서는 각 인터리버를 사용하였을 때의 성능 해석이 목적이기 때문에 근사식을 사용하지 않았다. 만일 BPSK(binary phase shift keying)에 의해서 신호가 전송될 때 동기식 수신기가 사용되어 위상을 수신측에서 완전히 복원할 수 있다면 수신측에서 샘플링되는 신호는 다음 식 (12)와 같은 관계를 갖는다.

$$y = a \cdot x + n \tag{12}$$

식 (12)에서 x, y는 각각 전송 신호와 수신 신호의 샘플링 값으로 수신된 신호의 심볼당 에너지가 E_s일 때 x는 $\pm\sqrt{E_s}$의 값을 갖는다. a는 진폭값을 나타내고 n은 평균이 0이고 분산 값이 $N_0/2$인 가산 백색 잡음이다. 식 (12)의 a 값은 다경로 전송으로 인해서 발생하는 페이딩의 영향을 받을 때는 a의 확률 밀도 함수를 알아야 하지만 AWGN 채널에서는 1의 고정된 값을 갖는다. 가산 가우시안 잡음의 확률 밀도 함수를 고려하면 식 (1)의 $\Gamma_k(s', s)$ 값을 계산하기 위한 transition probability $P(y_k | x_k = x)$는 다음 식으로 계산할 수 있다.

$$P(y_k | x_k = x) = \frac{1}{\sqrt{\pi N_0 / E_s}} \exp\left\{ \frac{-E_s}{N_0} \sum_{q=0}^{n} \left[y_i^q - (2x_i^q - 1) \right]^2 \right\} \tag{13}$$

식 (13)에서 n은 패리티 비트의 개수에 따라서 0~2의 값을 갖는다. 한편 오류 정정 부호화를 고려하여 실제 비트당 에너지를 E_b라 하면 오류 정정 부호화율을 γ라 할 때 심볼당 에너지는 $E_s = \gamma E_b$로 나타낼 수 있다.

각 인터리버의 성능을 분석하기 위해서 여러 가지 모의 실험 환경에서 가변 크기 인터리버의 성능을 해석하였다. Turbo 부호의 부호화기로는 그림 2에서 제시한 3GPP 표준에서 사용되는 RSC 부호화기를 사용하였고 비교 목적을 위하여 랜덤 인터리버를 사용하였을 때의 성능을 함께 제시하였다. 각각의 성능 분석에 사용한 turbo 블록의 수는 20,000개이다. 먼저 그림 7에는 turbo 복호기의 iteration 수가 변할 때의 BER 변화를 제시하였다. Iteration 수가 10 이상이면 대체로 BER은 안정화되는 경향을 보이는데 3GPP 인터리버나 랜덤 인터리버가 iteration 수가 증가할수록 BER 감소가 더욱 두드러진다. 이는 3GPP 인터리버와 랜덤 인터리버의 불규칙성이 크므로 비트 재배치로 인한 성능 증가가 다른 인터리버에 비해서 좋기 때문이다. 랜덤 인터리버와 3GPP 인터리버는 거의 비슷한 성능을 보이는데 그림 7에서 랜덤 인터리버의 성능이 3GPP 인터리버의 성능에 비해서 다소 떨어지는 이유는 본 논문에서 사용한 랜덤 인터리버의 경우 인터리버의 불규칙도는 좋으나 s-parameter를 고려하지 않았기 때문이다. Iteration 수가 15 이상이면 성능 증가가 거의 없기 때문에 앞으로 제시하는 결과는 iteration 수를 15로 하였다. 다음에 turbo 부호의 블록의 크기가 512이고 E_b/N_0가 2.0dB라고 가정했을 때 오류 정정 부호화율이 변할 때의 BER(bit error rate) 성능을 그림 8에 나타내었다. 그림에서 알 수 있는 것과 같이 오류 정정 부호화율이 클 때는 각 인터리버를 사용했을 때의 성능 차이가 크지 않지만 오류 정정 부호화율이 작아질수록 3GPP 인터리버와 랜덤 인터리버에 비해서 JPL 인터리버와 블록 인터리버의 성능이 떨어진다. 이는 3장에서 분석한 것과 같이 인터리버의 불규칙성에 따라서 재배치되는 비트열의 상관도가 달라지기 때문에 오류 정정 부호화율이 작아져서 패리티 비트가 많이 포함될수록 인터리버의 불규칙성에 BER 성능이 더 많이 영향을 받는다고 해석할 수 있다. 만일 멀티미디어 정보와 같이 정보량이 크고 어느 정도의 오류를 용인할 수 있는 경우에 전체 정보량의 제한으로 오류 정정 부호화율을 높여야 한다면 랜덤 인터리버와 3GPP 인터리버를 사용한 경우와 비교해서 JPL 인

터리버를 사용한 경우의 성능 차이가 별로 없기 때문에 구현이 간단한 JPL 인터리버가 유력한 대안이 될 수 있다. 특히 오류 정정 부호화율이 1/2보다 큰 경우에는 JPL 인터리버의 성능이 3GPP 인터리버나 랜덤 인터리버의 성능과 거의 비슷한 경향을 보인다. 그림 9에는 Eb/No의 변화에 따른 성능 변화를 제시하였고 그림 10~그림 13에는 블록 크기가 변할 때의 결과를 제시하였다. 비교 목적을 위해서 멀티미디어 정보의 전송을 위한 오류 정정 부호화로 많이 사용되는 RCPC 부호를 사용한 결과를 함께 제시하였는데 RCPC 부호와 turbo 부호의 성능 차이는 크다. 특히 블록의 크기가 커질수록 그 차이는 더욱 커지는 것을 알 수 있다. 그림 12와 그림 13에서 보는 것과 같이 turbo 부호의 1 블록을 1 패킷으로 생각한 경우 RCPC 부호의 경우에는 패킷 오류가 패킷의 크기가 증가할수록 커지는데 비해서 turbo 부호는 패킷 크기가 증가할수록 패킷 오류도 낮출 수 있다. BER과 패킷 오류 확률이 블록 크기 증가에 따라서 안정적으로 감소하지 않는 경우가 있는데 이는 3장에서 분석한 것과 같이 인터리버의 불규칙도와 s-parameter가 블록 크기에 따라서 일정한 값을 보이지 않기 때문이다. 특히 불규칙도가 낮은 JPL 인터리버를 사용한 경우 패킷 오류의 감소율이 일정하지 않은데 비해서 3GPP 인터리버를 사용한 경우에는 패킷 오류도 블록 크기가 커질수록 비교적 안정적으로 감소한다. 특히 동영상 신호의 경우 오류 전파 등을 고려할 때 intra 프레임의 오류를 줄이는 것이 매우 중요한 문제인데 압축된 동영상 신호의 특성상 intra 프레임의 정보량이 inter 프레임에 비해서 크기 때문에 turbo 부호를 오류 정정 부호로 사용하는 경우에 intra 프레임의 오류를 크게 줄일 수 있게 된다.

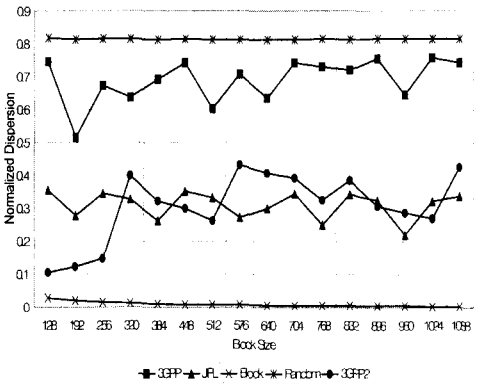


그림 5. 각 인터리버에 대한 정규화된 산포도

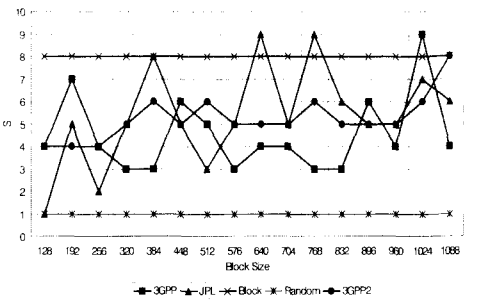


그림 6. 각 인터리버에 대한 s-parameter

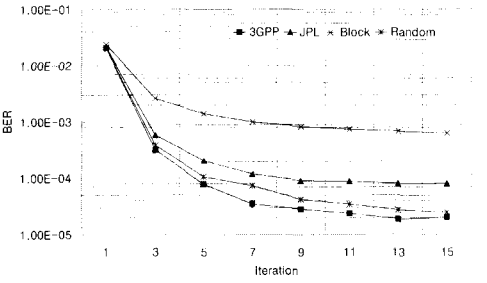


그림 7. Iteration 수를 변화시킬 때 각 인터리버의 성능 비교 (블럭 크기: 512, 채널 부호화율: 1/2, Eb/No: 2.0dB)

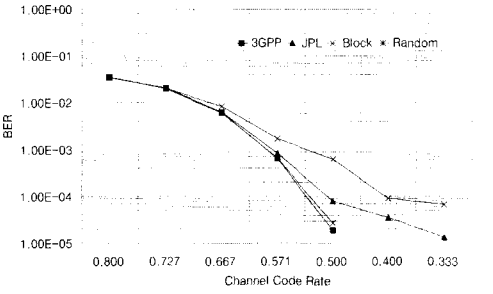


그림 8. 오류정정 부호화율을 변화시킬 때 각 인터리버의 성능 비교 (블럭 크기: 512, Eb/No: 2.0dB)

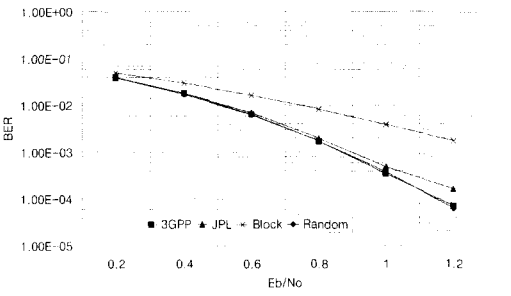


그림 9. Eb/No을 변화시킬 때 각 인터리버의 성능 비교 (블럭 크기: 512, 채널 부호화율: 1/3)

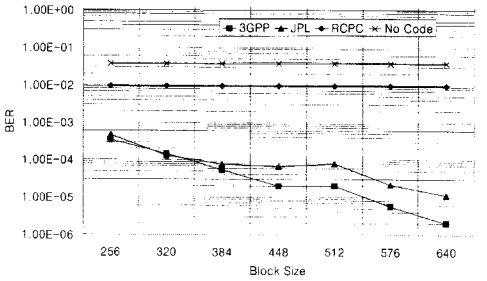


그림 10. 블록크기를 변화시킬 때 오류정정 기법의 BER 성능 비교 (채널 부호화율: 1/2, Eb/No: 2.0dB)

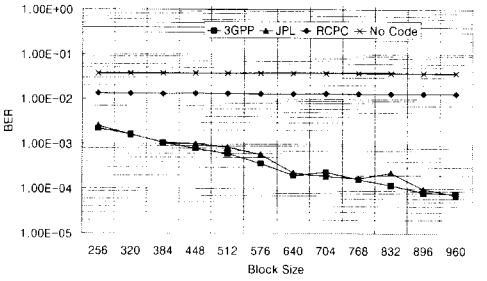


그림 11. 블록크기를 변화시킬 때 오류정정 기법의 BER 성능 비교 (채널 부호화율: 4/7, Eb/No: 2.0dB)

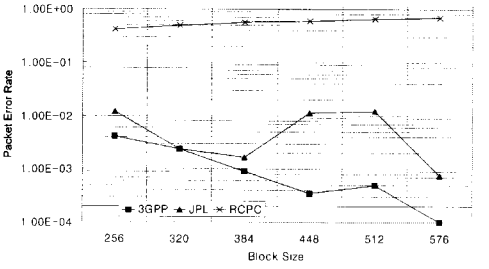


그림 12. 블록크기를 변화시킬 때 오류정정 기법의 packet error rate 성능 비교 (채널 부호화율: 1/2, Eb/No: 2.0dB)

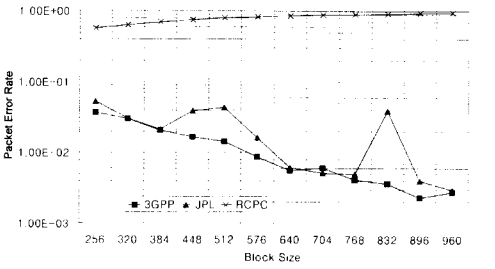


그림 13. 블록크기를 변화시킬 때 오류정정 기법의 packet error rate 성능 비교 (채널 부호화율: 4/7, Eb/No: 2.0dB)

V. 결 론

본 논문에서는 멀티미디어 정보를 안정적으로 전송하기 위해서 오류 정정 부호로 우수한 특성을 보이는 turbo 부호를 효율적으로 적용하기 위한 방법을 연구하였다. 멀티미디어 정보의 크기가 정보의 특성에 따라서 변하는 것을 고려하여 가변 크기 turbo 부호의 사용을 제안하고 그 특성과 성능을 해석하였다. 가변 크기 turbo 부호를 구현하기 위해서는 인터리버의 크기를 가변으로 해야 하는데 가변 크기 인터리버로 사용될 수 있는 3GPP 표준의 turbo 부호에서 사용되는 인터리버와 JPL 인터리버, 블록 인터리버를 중요한 비교 대상으로 하였다. 본 논문의 실험 결과 인터리버의 불규칙성이 우수한 3GPP 인터리버가 모든 오류 정정 부호화율과 블록 크기에 대해서 우수한 성능을 보인다. 그런데 원 정보량이 커서 총 정보량을 고려할 때 오류 정정 부호화율을 작게 할 수 없는 경우에는 그 구현 방법이 비교적 간단한 JPL 인터리버도 3GPP 인터리버와 비슷한 성능을 보이는 것을 알 수 있다. 특히 가변 크기 turbo 부호의 경우 블록 크기가 커지면 블록 오류 발생 확률이 오히려 감소하는데 이는 압축된 동영상 신호의 특성을 고려하면 오류 정정 부호로는 매우 우수한 특성이라고 판단된다. 앞으로 turbo 부호가 압축된 동영상을 포함한 멀티미디어 정보의 오류 정정 부호 기법으로 효율적으로 사용될 수 있도록 구현 방법을 단순화하기 위한 연구가 필요하다고 생각된다. 특히 3GPP 인터리버를 효율적으로 구현하는 방안과 구현 방법이 간단한 JPL 인터리버의 구조를 변형하여 불규칙도를 높이는 방법 등이 연구되어야 한다.

참고문헌

[1] E. Dahlman, B. Gudmundson, M Nilsson and J. Skold, "UMTS/IMT-2000 based on wideband CDMA," IEEE Commun. Mag., vol. 36, pp. 70-80, Sep. 1998.

[2] N. Farber, B. Girod and J. Villasenor, "Extension of ITU-T recommendation H.324 for error-resilient video transmission," IEEE Commun. Mag.,

vol. 36, pp. 120-128, June 1998.

[3] J. Hagenauer, "Rate-compatible punc tured convolutional codes and their applications," IEEE Trans. Commun., vol. 36, pp. 389-400, Apr. 1988.

[4] ITU-T, "Multiplexing protocol for lo w bitrate multimedia communication over highly error-prone channels, dr aft ITU-T recommendation H.223-A nnex C," Dec. 1998.

[5] ITU-T, "Optional multiplexing protoc ol for low bitrate multimedia comm unication over highly error-prone ch annel, draft ITU-T recommendation H.223-Annex D," Dec. 1998.

[6] C. Berrou, A. Glavieux and P. Thiti majshima, "Near Shannon limit error -correcting coding and decoding: Tu rbo codes," IEEE Proc. ICC '93, Ge neva, Switzerland, pp.1064-1070, Ma y 1993.

[7] C. Berrou and A. Glavieux, "Near op timum error correcting coding and d ecoding: turbo-codes," IEEE Trans. Commun., vol. 44, pp.1261-1271, Oct. 1996.

[8] C. Heegard and S. B. Wicker, Turbo coding, Boston: Kluwer Academic P ublisher, 1999.

[9] 3rd Generation Partnership Project, " Multiplexing and channel coding(FD D)," 3GPP Technical Specification, TS 25.212 v4.2.0, Sep. 2001.

[10] 3rd Generation Partnership Project 2, "Physical layer standard for cdma 2000 spread spectrum system," Rele ase A, July 2001.

[11] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linea r codes for minimizing symbol error rate," IEEE Trans. on Information T heory, pp. 284-287, March 1974.

[12] Warren J. Gross and P. Glenn Gula k, "Simplified MAP algorithm suitab le for implementation of turbo decod ers," Electronics Letters, vol. 34. no. 16, pp. 1577-1578, Aug. 1998.

[13] S. Benedetto, D. Divsalar, G. Monto rsi and F. Pollara, "Soft-output deco ding algorithms in iterative decoding of turbo codes," TDA Progress Rep. 42-124, JPL, pp. 63-87, Feb. 1996.

[14] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using r andom and nonrandom permutations, " TDA Progress Rep. 42-122, JPL, Aug. 1995.

[15] D. Garett, B. Xu and C. Nicol, "Ene rgy efficient turbo decoding for 3G mobile," International Symposium on Low Power Electronics and Design, pp. 328-333, 2001.

부 록

블록의 크기가 $N \cdot M$ (N은 짝수) 인 JPL 인터리버의 입력 i와 출력 $\pi(i)$ 간의 관계는 다음 식에 의해서 나타낼 수 있다[8].

$$\pi(i) = 2 \cdot r(i) + N \cdot c(i) - m(i) + 1 \qquad \text{(a.1)}$$

식 (a.1)에서 $m(i)$, $r(i)$, $c(i)$는 각각 다음 식 (a.2)~(a.4)와 같은 관계를 갖는다.

$$m(i) = \mathrm{mod}(i, 2) \qquad \text{(a.2)}$$

$$r(i) = \mathrm{mod}((19 \cdot r_0 + 1), N/2) \qquad \text{(a.3)}$$

$$c(i) = \mathrm{mod}(p(l+1) \cdot c_0 + 21 \cdot m, M) \qquad \text{(a.4)}$$

이때 $c_0 = \mathrm{mod}((i - m(i))/2, M)$,

$r_0 = ((i - m(i))/2 - c_0)/M$,

$l = \mathrm{mod}(r(i), 8)$이고 $p(1) \sim p(8)$은
$p(1) = 31$, $p(2) = 37$, $p(3) = 43$,
$p(4) = 47$, $p(5) = 53$, $p(6) = 59$,
$p(7) = 61$, $p(8) = 67$인 소수이다. N을 8로 하고 M의 크기를 변화시키면 크기가 8인 가변 블록 크기 turbo 부호를 구현할 수 있다. 그런데 만일 행의 수인 M이 $p(1) \sim p(8)$ 의 배수가 된다면 c_0와 $c(i)$가 정확히 일대일 대응 관계를 만족하지 않으므로 본 논문에서는 이러한 $p(l)$을 미리 제외하여 모든 크기에 대해서 JPL 인터리버가 잘 동작되도록 설계하였다.

이 창 우 (Chang-Woo Lee) 정회원
한국통신학회 논문지 제25권 제7호 참조
현 재 : 가톨릭대학교 컴퓨터 · 전자공학부 부교수

<주관심분야> 영상 압축, 영상 전송, turbo 부호