

# 감소된 복호 지연을 갖는 SOVA 기반 터보 부호에 관한 연구

정회원 강 경 우\*, 박 노 진\*, 강 철 호\*

## A Study on SOVA-Based Turbo Code with Reduced Decoding Delay

Kyung-Woo Kang\*, No-jin Park\*, Chul-ho Kang\* *Regular Members*

### 요 약

Turbo Code는 반복 복호 알고리즘을 사용함으로써 백색 가우시안 잡음(AWGN) 채널 환경하에서 Shannon의 한계에 가까운 성능을 보이는 오류정정 방식으로 제안되었다. 그러나 Turbo code는 반복복호로 인해 매 복호시마다 큰 인터리버와 복호기를 거쳐야 하기 때문에 수신과정에서 커다란 지연을 요구하게 된다. 따라서 차세대 무선 멀티미디어 통신에서 실시간 음성서비스나 화상서비스를 제공하는데 어려움이 많다. 본 논문에서는 기존의 터보 복호기를 변형하여 매 복호시 각각의 복호기에서 LLR 출력시퀀스를 발생시킴으로서 반복 복호 횟수를 줄이는 방법을 제안하였다. 이렇게함으로써 기존의 Turbo code가 갖는 성능은 크게 변화시키지 않으면서 각각의 정보프레임을 가변적으로 복호함으로써 반복 복호로 인한 시간 지연을 줄였다.

### ABSTRACT

Turbo code was proposed as the error correction method which has a nearly shannon-limit performance in AWGN(Additive White Gaussian Noise) channel environments using iterative decoding algorithm. But it needs the large delay in the receiver because each iterative decoding process is required to pass by the interleaver and decoder. Thus, it has much difficulty in real-time voice and image services for the future wireless multimedia communication. In this paper, the method which can reduce the number of iterative decoding is proposed by generating the LLR output sequence of decoder per each iteration with modified conventional turbo code. In the result, the time delay of iterative decoding can be reduced by variably decoding each information frame without much changing the conventional turbo code performance.

### I. 서 론

디지털 통신 시스템에서 오류정정 부호는 무선통신 채널상에서 발생할 수 있는 페이딩, 간섭, 잡음 등으로 왜곡된 신호로부터 전송된 정보를 보호하기 위한 기술이다. 1993년 Berrou 등에 의해 처음으로 발표된 터보부호(turbo code)는 낮은  $E_b/N_0$ 에서 강력한 오류정정 능력을 가진 코딩기법으로 반복적인 복호 알고리즘을 사용하여 샤논(shannon)의 한계에 거의 근접하는 코드로 알려져 왔다<sup>[1]</sup>. Berrou

는 AWGN 채널에서 인터리버의 크기가 65532이고 반복 복호를 18회 반복했을 때 BER이  $10^{-5}$ 이 되는 지점에서 비트 에너지 대 잡음의 전력비가 0.7dB임을 보였다. 터보부호에서 성능에 영향을 미치는 주요 요소로는 ①프레임 사이즈, ②부호기 메모리 사이즈, ③평처링의 유무, ④노이즈 레벨과 본 논문에서 연구 주제인 반복 복호 횟수 등이 있다. 일반적인 터보 복호기의 구조는 2개의 복호기가 직렬로 연결되어 반복적으로 정보를 교환하는 구조로 구성된다. 본 논문에서는 복호시에 하나의 LLR 출

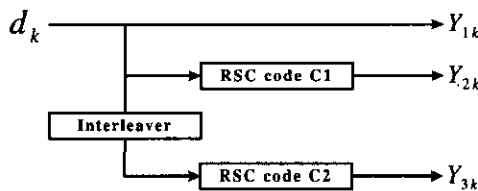
\* 광운대학교 전자통신공학과 신호처리연구실(kwkang@explore.kwangwoon.ac.kr)  
논문번호 : 00326-0816, 접수일자 : 2000년 8월 16일

력 값을 발생시키는 기존의 직렬 모드에 3개의 LLR 출력값을 내도록 변형하여 지연에 커다란 영향을 미치는 반복 복호 횟수를 가변적으로 줄이는 방법을 제안한다.

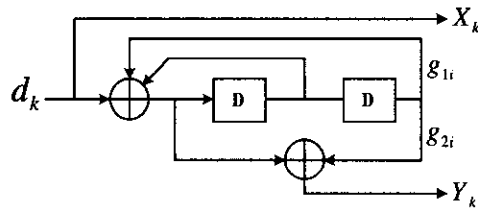
먼저 2장에서는 터보 부호의 부호기/복호기 및 본 논문에서 사용된 SOVA 알고리즘을 살펴보고 3장에서는 본 논문에서 변형 제안된 터보 복호기를 소개한다. 4장에서는 모의 실험을 통해 기존의 터보 복호기에 대한 성능과 제안한 터보 복호기와의 성능을 비교, 분석한다. 마지막으로 5장에서는 4장까지의 결과를 정리하고 앞으로의 연구 방향을 제시한다.

## II. 터보 코드의 기본구조

### 2.1. 터보 부호기(Turbo Encoder)



(a) 터보 부호기의 구조



(b) RSC 부호(R=1/2, K=3)

그림 1. 터보 부호기의 구조

일반적인 터보코드에 대한 부호기의 구조가 [그림 1-a]에 나타나 있다. 부호기는 두 개의 RSC(Recursive Systematic Convolutional) 구성 부호와 이 두 RSC 구성 부호를 연결시켜주는 인터리버로 구성되어 있다. 여기서 RSC 부호란 길쌈 부호의 일종으로 피드백이 존재하는 systematic 부호이다. [그림 1-b]에는 구속장의 길이(constraint length)가 3, 부호화율이 1/2인 부호어를 생성시키는 구성 부호기가 나타나 있다. 즉 하나의 입력 비트에 대해 두 개의 부호 비트를 생성하는 부호이다. 여기서

$d_k$ 는 입력 데이터 비트이고,  $X_k$ 는 첫 번째 부호 비트,  $Y_k$ 는 두 번째 부호 비트를 각각 나타낸다. RSC 부호는 일반적으로 낮은 부호화율을 갖는 부호에서 non-systematic 부호보다 우수한 비트오류율(BER) 성능을 낼 수 있다고 알려져 있다<sup>[2]</sup>.

[그림 1-a]에서 인터리버는 입력되어 들어오는 데이터를 분산시켜 두 번째 RSC 부호기로 보낼 수 있도록 해주는 기능을 수행한다. 따라서 터보 코드의 부호화 과정을 총체적으로 살펴보면 다음과 같다. 입력 데이터 비트  $d_k$ 가 들어오면 첫 번째 부호기의 출력  $Y_{1k}$ 는  $d_k$ 가 되고, 두 번째 출력비트  $Y_{2k}$ 는 첫 번째 RSC 구성 부호의 부호 비트, 세 번째 출력 비트  $Y_{3k}$ 는 입력 비트가 인터리버를 거친 후 두 번째 RSC 구성 부호기로 입력되어 생성된 부호 비트가 된다. 따라서 [그림 1-a]에 나타난 터보 부호기로부터 생성되는 부호어의 부호화율은 1/3이 된다.

터보코드에서는 비교적 낮은 부호율의 부호어가 생성되는데 이는 곧 주파수 대역폭이 많이 필요하다는 의미로 볼 수 있다. 이와같은 단점을 보완하기 위해서 생성된 부호어의 일부를 삭제하고 전송하는 평처링(puncturing)기법을 사용한다<sup>[1]</sup>. 평처링은 정보비트에 대해서는 적용되지 않고 패리티 비트에 대해서만 적용된다. 부호율 1/3의 터보코드로부터 부호화율 1/2의 터보코드를 생성하는 평처링기법은 다음과 같다. [그림 1-a]에 나타난 부호기에서 생성되는 출력을  $\{Y_{1k}^1, Y_{2k}^1, Y_{3k}^1, Y_{1k}^2, Y_{2k}^2, Y_{3k}^2, \dots\}$ 라 한다면 평처링된 부호기의 출력 부호어는  $\{Y_{1k}^1, Y_{2k}^1, Y_{1k}^2, Y_{3k}^2, \dots\}$ 가 된다.

터보코드가 우수한 성능을 내기 위해서는 [그림 1-a]의 인터리버가 비균일(non-uniform) 인터리버여야 한다고 알려져 있는데, 이것은 RSC 부호의 특성상 최대한 거리 특성이 우수한 부호어를 생성하기 때문이다<sup>[2]</sup>.

본 논문에서 부호기는 2개의 RSC 코드가 동일한 구성 부호라고 간주하였으며, 2개의 터보 부호에 사용된 구성부호로는 생성 다항식이  $(7,5)_8$ 인 4 상태 코드가 사용되었다. 이 코드는 메모리가 2개일 때 가장 효과적인 거리를 제공하는 것으로 알려져 있다<sup>[3]</sup>. [그림 1]에서 입력정보 열  $\{d_k\}$ 의 경우, 임의의 시간  $k$ 에서의 부호기 출력  $\{Y_k\}$ 는 각각  $Y_{1k}$ 와 첫 번째 부호화기의 출력  $Y_{2k}$  또는 두 번째 부호화기의 출력  $Y_{3k}$ 를 나타내며 부호화기의 입력과 출

력 사이의 관계를 살펴보면 다음과 같다.

$$d_k = Y_{1k} \quad (1)$$

$$Y_{2k} = d_k + \sum_{i=0}^{K-1} r_i a_{1K-i} \pmod{2} \quad (2)$$

$$Y_{3k} = \bar{d}_k + \sum_{i=0}^{K-1} r_i a_{2K-i} \pmod{2} \quad (3)$$

여기서,  $r_i$ 는  $Y_{1k} = d_k$ 일때는  $g_{1i}$ 이고  $Y_{2k} = d_k$  또는  $Y_{3k} = d_k$ 일때는  $g_{2i}$ 가 된다. 그리고  $a_k$ 는 정보  $d_k$ 와 피드백값의 모듈로(modulo-2) 연산으로 표현되며  $\bar{d}_k$ 는 원래 정보  $d_k$ 의 인터리빙된 정보를 나타낸다.

### 2.2 터보 복호기(Turbo Decoder)

터보 코드의 복호 알고리즘으로는 MAP(Maximum A Posteriori) 알고리즘과 SOVA(Soft-Output Viterbi Algorithm) 알고리즘을 이용하는 방법이 있다. MAP 알고리즘을 사용함으로써 BER(Bit Error Rate)을 최소화시키고 SOVA보다 더 좋은 성능을 얻을 수 있다. 보통 BER  $10^{-4}$ 에서 MAP이 SOVA보다 약 0.7[dB]정도 우수한 것으로 알려져 있다. 그러나 MAP 알고리즘은 계산량이 많고, 매우 복잡하기 때문에 복호기를 실제적으로 구현하는데 어려움이 많다. 이에 반하여 SOVA의 BER 성능은 MAP보다 다소 떨어지는 단점이 있지만, 시퀀스 오류 확률을 최소화 시킬 수 있고, 상대적으로 복호기 구현이 용이하다는 장점을 가지고 있다.

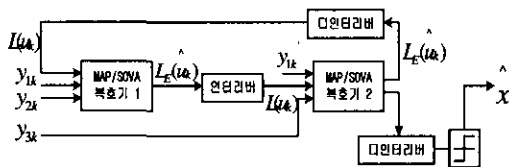


그림 2. 터보 복호기 구조

일반적인 터보 복호기의 구조는 [그림 2]와 같다. 여기서  $y_k = (y_{1k}, y_{2k}, y_{3k})$ 는  $k$ 번째 수신된 정보 벡터를 나타내고,  $y_{1k}$ 는 인터리버를 통과한  $y_{1k}$ 을 표시한다. 맨 처음 채널을 통과한 정보비트  $y_{1k}$ 와 패리티 비트에 대한 연판정(soft-decision)정보  $y_{2k}$ , 그리고 외부정보(extrinsic information)인 사전정보(a priori information)  $L(u_k)$ 가 첫 번째 복호기의 입

력으로 들어간다. 복호기1의 출력인  $L_e(u_k)$ 가 인터리버를 거친 후 사전정보로서 복호기2의 입력으로 들어간다. 동시에 채널을 통과한 패리티비트  $y_{3k}$ 와 인터리버를 통과한 정보비트인  $y_{1k}$ 이 복호기2로 입력된다. 복호기2의 출력인 외부정보  $L_e(u_k)$ 는 사전정보로서 다시 복호기1의 입력으로 들어가는 데 이러한 과정을 원하는 성능을 얻기 위해 필요한 만큼 반복복호하게 된다. 반복복호의 과정이 끝나면 복호기2의 LLR(log-likelihood ratio)출력을 디인터리빙한 후 강판정(hard-decision)해서 정보를 추정한다.

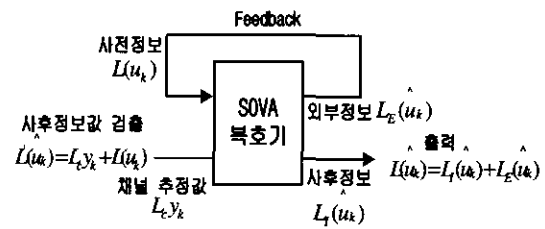


그림 3. SOVA 복호기

[그림 3]은 SOVA를 이용한 반복 복호 구조를 나타내었다. 맨처음 연판정(soft-decision) 비터비 알고리즘이 수행된다. 시간  $k$ 에서 경로  $m$ 일 확률은 다음과 같이 정의된다.

$$P_k^m = C e^{M_k^{-1/2}} \quad (4)$$

여기서,  $C$ 는 상수이고  $M_k^m$ 은 경로 메트릭(path metric)이다. AWGN 채널에서 부호율  $1/N$  인 축적 경로 메트릭(accumulative path metric)은

$$M_k^m = M_{k-1}^m + \sum_{n=1}^N x_{k,n}^m L_c y_{k,n} + x_{k,1}^m L(u_k) \quad (5)$$

에 의해서 갱신(updating)된다. 여기서  $x$ 는 부호어 시퀀스,  $y$ 는 수신된 시퀀스이며  $L_c$ 는 수신기에서 부호 시퀀스 채널 추정값인데 다음 식으로 정의된다.

$$L_c = 4 \frac{E_s}{N_o} \quad (6)$$

따라서 잘못된 경로를 선택할 확률은

$$\Phi_k^m = \frac{P_k^m}{P_k^m + P_k^m} = \frac{1}{1 + e^{-(M_k^m - M_k^m)/2}} \quad (7)$$

이다. 여기서  $M_k^m$ 은 현재 경로에서의 경로 매트릭이며  $P_k^m$ 은 현재 경로에 대한 확률을 표시한다. 그리고 경로 판정에 대한  $LLR(\log - \text{likelihood ratio})$ 은

$$\Delta_k^m = \log \frac{1 - \phi_k^m}{\phi_k^m} = (M_k^m - M_k^m) / 2 \quad (8)$$

이 된다. 각각의 출력  $L(\hat{u}_k)$ 에 대해서 신뢰성있는  $LLR$ 값을 구하기 위해 경로신뢰도(path reliability)  $\Delta_k^m$ 로부터 비트신뢰도  $L(\hat{u}_k)$ 을 갱신해야 한다.  $u_k^m$ 에 대한 중간 신뢰값을  $L_k^m$ 이라 놓으면,  $L_k^m$ 의 초기값은  $\Delta_k^m$ 이 되고 복호가 진행되는 동안 갱신규칙에 의해 수정되게 된다. 즉,

$$L(\hat{u}_k) = \hat{u}_k L_k^m \quad (9)$$

이 되고, 여기서  $\hat{u}_k$ 는 단 한 개의 생존 시퀀스이다. SOVA 알고리즘에서 비트 신뢰도를 얻기 위한 갱신규칙(updating rule)은 G.Battail에 의한 방법(BR)<sup>[4]</sup>과 J.Hagenauer가 제안한 방법(HR)<sup>[5]</sup>이 있다.

BR-SOVA에서 만약 어떤 시간  $k$ 에서 생존 시퀀스  $u_j^s$ 의  $j$ 번째( $j < k$ )비트가 경쟁 시퀀스  $u_j^c$ 의  $j$ 번째 비트와 같지 않다면 새로운  $LLR$ 값  $L_j^s$ 는  $L_j^s$ 와  $\Delta_k^m$ 사이의 최소값이 되고,

$$u_j^s \neq u_j^c \Rightarrow \min(L_j^s, \Delta_k^m) \quad (10)$$

반대의 경우라면 새로운  $LLR$ 값  $L_j^s$ 는  $\Delta_k^m$ 과  $L_j^c$ 를 합한 값과  $L_j^s$ 사이의 최소값이 된다.

$$u_j^s \neq u_j^c \Rightarrow \min(L_j^s, \Delta_k^m + L_j^c) \quad (11)$$

HR-SOVA에서는 간단히  $u_j^s \neq u_j^c$ 일때만 갱신된다. 따라서 계속해서  $L_j^c$ 를 갱신할 필요가 없어 SOVA를 사용한 하드웨어 구현시 보다 낮은 복잡도를 갖는 복호기 구현에 용이하다.  $L(\hat{u}_k)$ 는 채널 추정값  $L_y$ 로 얻어지는 내부정보(intrinsic information)  $L_I(\hat{u}_k)$ 와 사전정보(a priori information)  $L(\hat{u}_k)$ 로 구성된다. 사전정보는 외부정보  $L_E(\hat{u}_k)$ 와 동일한 값으로서 새로운 사전 추정을 위해 다음 복호기의 입력이 된다. 이 외부정보는 다음과 같이 구해질 수 있다.

$$L_E(\hat{u}_k) = L(\hat{u}_k) - L_I(\hat{u}_k) \quad (12)$$

따라서 위의 과정을 필요한 만큼 반복함으로써 보다 신뢰성 있는 터보 복호기 구현이 가능하다.

### III. 제안한 방법의 터보 복호기

어떤 시점에서 임의의 정보 프레임이 복호되기 위해서는 필요한 만큼의 반복복호 횟수에 따라 매 복호시 터보 복호기와 인터리버를 통과하게 된다. 이렇게함으로써 원하는 성능은 얻을 수 있지만, 그에 대한 보상으로 반복복호를 할 때마다 실시간 통신에서 커다란 장애라 할 수 있는 큰 지연을 수반하게 된다. 이 장에서는 이러한 단점을 보완하기 위해 기존의 터보 복호기를 변형 제안함으로써 지연을 줄일 수 있는 복호화 방법을 제안한다.

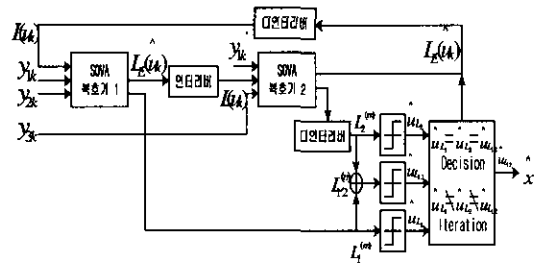


그림 4. 제안된 터보 복호기

변형 제안된 터보 복호기의 구조는 [그림 4]와 같다. [그림 4]에서  $L_1^{(m)}$ 과  $L_2^{(m)}$ 는  $m$  번째 복호화가 진행된 후 복호기1과 복호기2의  $LLR(\log - \text{likelihood ratio})$ 출력값이다. 그리고  $L_{12}^{(m)}$ 는  $L_1^{(m)}$ 과  $L_2^{(m)}$ 값을 합한  $LLR$ 값이다. 매 복호시마다 복호기1과 복호기2에서 각각 출력되는  $LLR$ 값은 서로 다른 값들의 분포이다. 따라서 각각의  $LLR$  출력값을 강판정(hard-decision)한 시퀀스  $\hat{u}_{L_1}^{(m)}$ ,  $\hat{u}_{L_2}^{(m)}$ 도 서로 다른 분포값을 가지며, 각각의 오류를 계산해 보면 두 오류값이 서로 다르다. 강판정된  $\hat{u}_{L_1}^{(m)}$ 와  $\hat{u}_{L_2}^{(m)}$ 의 분포는 서로 다르지만, 복호 횟수  $m$ 이 증가할수록 두 값의 신뢰도가 높아지는 분포로 새롭게 갱신되어 더 이상 성능 향상이 없는 오류값으로 수렴하게 된다. 기존의 터보 복호기에서는 요구되는 반복 복호횟수  $m$ 번의 복호를 진행한 후  $\hat{u}_{L_2}^{(m)}$ 를 추정한다. 이럴 경우

오류가 줄어드는 방향으로 수렴이 빨리 이루어지지 않으면 정해진 복호 횟수만큼 복호를 해야하겠지만, 그렇지 않고 정해진 복호횟수 이전에 수렴이 빨리 이루어질때는 그만큼의 불필요한 복호가 이루어져 바람직한 현상이라 할 수 없다. 특히 비트 에너지 대 잡음의 전력비가 클수록 오류분포 수렴 속도가 빨라져 불필요한 복호로 인한 지연은 더 크다고 할 수 있다. 그러므로 위에서 언급한 각각의  $LLR$ 값이 서로 다르다는 분포 특성을 이용하여 복호 횟수를 상당량 줄일 수 있는 방법을 본 논문에서 제안한다. 제안한 터보 복호 방식은 다음과 같다.  $\hat{u}_{L_u}^{(m)}$ 와

$\hat{u}_{L_1}^{(m)}$ ,  $\hat{u}_{L_2}^{(m)}$ 를 매 복호시마다 서로 비교함으로써 오류 분포가 수렴하여 강판정한 3개의 출력값이 같으면 더 이상 복호화를 진행하지 않고

$\hat{u}_{L_u}^{(m)}$ 를 판정하고, 그렇지 않으면 정해진 복호 횟수만큼 반복복호를 진행하게 된다. 이렇게 함으로써 기존의 터보부호의 성능은 크게 변화시키지 않고 가변적으로 복호를 진행함으로써 지연을 상당량 줄일 수 있음을 본 논문의 모의 실험을 통해 확인하였다. 이것을 정리하면 다음과 같다.

$m$  번째 복호  $\Rightarrow$

$$\hat{u}_{L_1}^{(m)} = \hat{u}_{L_2}^{(m)} = \hat{u}_{L_u}^{(m)} \Rightarrow \hat{u}_{L_u}^{(m)}$$

$$\hat{u}_{L_1}^{(m)} \neq \hat{u}_{L_2}^{(m)} \neq \hat{u}_{L_u}^{(m)} \Rightarrow \text{반복복호} \quad (13)$$

여기서  $LLR$  시퀀스의  $sign$ 을 고려해 보면 다음과 같다. 시퀀스  $L_1^{(m)}$ 과  $L_2^{(m)}$ 의  $sign$ 을 비트 단위로 서로 비교해 보면 두 시퀀스는  $sign$ 이 서로 같은 비트들과  $sign$ 이 서로 반대인 비트들로 구성되어 있다는 것을 알 수 있다. 따라서 두 시퀀스를 더한  $L_{12}^{(m)}$ 의 신뢰도 값은  $sign$ 이 같을 경우 그 비트에 대한 신뢰도가 더 커지고  $sign$ 이 다를 경우에는 신뢰도 값이 큰쪽에 가까워져 전체적으로 오류 없는 값으로 수렴하지 않으면 세 값이 틀린 분포를 가지게 된다는 것을 알 수 있다. 물론 복호가 진행될수록 이 세 값은 오류가 없는 값으로 수렴하게 된다. 제안한 구조는 기존의 터보 복호기에 비해 약간의 계산과정이 추가되지만 그에 대한 보상으로 지연에 커다란 영향을 미치는 복호 횟수를 가변적으로 줄일 수 있어 성능의 향상을 가져온다.

#### IV. 실험 결과 및 고찰

본 논문의 채널 모델은  $AWGN$  채널이며  $BPSK$  변조를 가정하였다. 복호화 알고리즘은  $SOVA$ 를 사용하였고, 반복 복호 횟수는 5회, 정보 시퀀스의 프레임 크기  $N$ 이 128, 256, 512비트인 경우에 대하여 실험하였다. 터보 부호화기는 동일한 2개의  $RSC$  부호화기를 사용하였고 구성 부호화기의 생성 다항식은  $(7,5)_8$ , 부호율은  $1/2$ 과  $1/3$ 이 사용되었다. 인터리버는  $m$ 시퀀스 발생기를 이용한 랜덤 인터리버를 사용하였다.

[그림 5]와 [그림 6]는 각각의 프레임에 대해 부호율을  $1/2$ 과  $1/3$ 로 했을 때 제안한 방식의 복호기와 기존의 터보 복호기의 오차성능 ( $BER$ )을 비교한 것인데, 성능에 커다란 변화가 없음을 알 수 있다. 약간 성능에 변화를 보인 것은 복호기1과 복호기2의  $LLR$ 값을 판정한 값들이 오류가 줄어드는 방향으로 수렴하지 못하고 동일한 위치의 비트에서 에러가 생겨 복호를 중단함으로써 생긴 성능의 변화이다.

[그림 7]과 [그림 8]에서는 프레임 길이 256비트와 512비트에 대한 프레임 에러 성능 ( $FER$ )을 보였다. [그림 9]와 [그림 10]은 제안한 방식으로 복호했을 때 감소된 평균 복호 횟수를 나타낸 것이다. 그림에서 보면 부호율이  $1/3$ 일때보다는  $1/2$ 일 때 프레임 사이즈가 적을수록 복호 횟수 감소 정도가 큼을 알 수 있다. 프레임 길이에 따라 다소 차이가 있지만 부호율이  $1/2$ 일때 2.5~3.0dB에서 반복 복호 횟수가 평균 2.63 ~ 1.87회,  $1/3$ 일 때는 2.0~3.0dB에서 평균 2.77 ~ 1.78회 정도 복호함으로써 기존의 터보 복호기의 성능을 유지할 수 있음을 알 수 있었다. 그리고 2.5dB부터 서서히 평균 복호 횟수 감소 정도가 줄어 일정한 값으로 수렴함을 확인하였다.

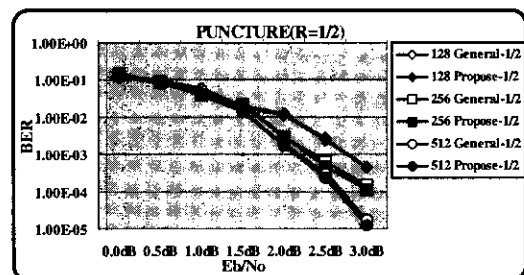


그림 5. 부호율 1/2인 경우 BER 성능 비교

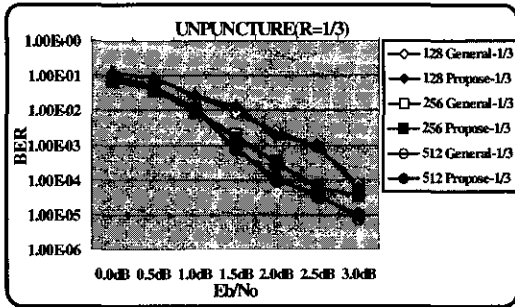


그림 6. 부호율 1/3인 경우 BER 성능 비교

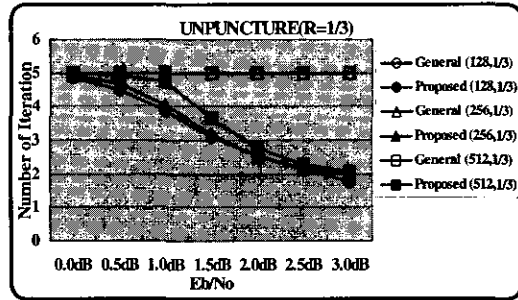


그림 10. 제안된 복호기의 평균 복호 횟수(1/3)

### V. 결론

본 논문에서는 기존의 터보 복호기의 LLR 출력 값 분포 특성을 이용하여 강판정된 값을 간단히 비교하는 방법을 제안함으로써 반복 복호 횟수를 상당량 줄일 수 있었다. 제안된 방법을 적용함으로써 비트 에너지 대 잡음의 전력비가 클수록 프레임 길이가 적을수록 반복 복호 횟수가 크게 감소됨을 알 수 있었다. 따라서 터보 부호에서 가장 커다란 단점으로 지적되어온 복호 지연을 줄임으로써 실시간 통신의 요구 사항에 어느 정도 부응할 것으로 본다.

향후 연구 과제는 복호기의 출력 시퀀스 분포에 대한 연구와 함께 다양한 채널 환경과 복호 알고리즘을 적용한 성능 평가가 이루어져야 하겠다. 또한 채널 환경을 변화시켜 다양한 프레임과 부호율에 대한 실험을 통해 제안한 방법을 일반화시키고 분포 특성을 수학적으로 규명하는 것이다.

### 참고 문헌

- [1] Claude Berrou, Alain Glavieux and Punya Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes(1)", in Proc. IEEE ICC, pp.1064-1070 May 1993
- [2] Berrou. C., Glavieux. A., "Near Optimum Error Correcting Coding and Decoding : Turbo Codes", IEEE Transaction on Communications, Vol. 44, No. 10, pp. 1261-1271, 1996
- [3] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," IEEE Trans. Commun., vol. 44, iss. 5, pp. 591-600, June 1996
- [4] G. Battail, "Ponderation des symboles decodes

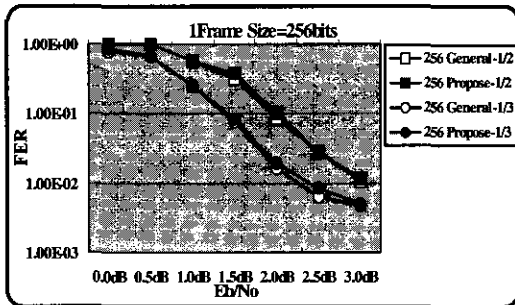


그림 7. 프레임길이 256비트 FER 성능 비교

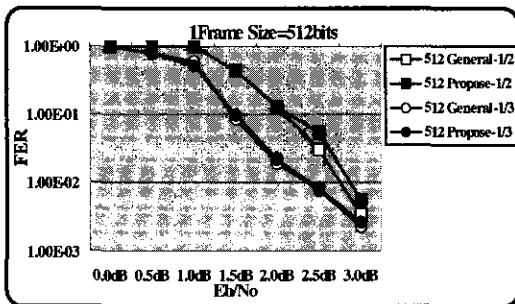


그림 8. 프레임길이 512비트 FER 성능 비교

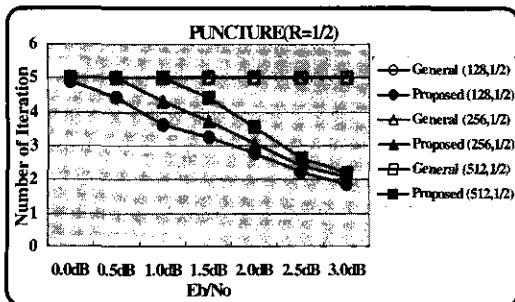


그림 9. 제안된 복호기의 평균 복호 횟수(1/2)

parlagorithem de Viterbi (in French),” Ann. Telecommun., Fr., 42, N 1-2, pp. 31-38, Jan. 1987

- [5] J. Hagenauer and P. Robertson, “Iterative (“TURBO”) Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms,” ITG-Fachberichte, V. 130, pp.21-29, 1995

강 경 우(Kyung-Woo Kang)

정회원



1999년 2월 : 수원대학교

정보통신공학과 공학사

1999년 3월~현재 : 광운대학교

전자통신공학과 석사과정

<주관심 분야> 통신신호처리,

채널 코딩

박 노 진(No-jin Park)

정회원

한국통신학회 논문지 2000년 제25권 제10호 참조

현재 : 광운대학교 대학원 전자통신공학과 박사수료

<주관심 분야> 통신신호처리, W-CDMA, 채널코딩

강 철 호(Chul-ho Kang)

정회원

한국통신학회 논문지 2000년 제25권 제10호 참조

현재 : 광운대학교 전자공학부 교수

<주관심 분야> 통신 및 음성신호처리, 이동통신