

감소된 복호 지연을 갖는 SOVA기반 터보 부호에 관한 연구

강 경 우, 박 노 진, 강 철 호
 광운대학교 전자통신공학과

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

Kyung Woo Kang, No Jin Park, Chul Ho Kang
 Dept. of Electronic Communication Eng. Kwangwoon Univ.
 kwkang@explore.kwangwoon.ac.kr

요 약

Turbo Code는 반복 복호알고리즘을 사용함으로써 백색 가우시안 잡음(AWGN)채널 환경에서 Shannon의 한계에 가까운 성능을 보이는 오류정정 방식으로 제안되었다. 그러나 Turbo code는 반복복호로 인해 매복호시마다 큰 인터리버와 복호기를 거쳐야 하기 때문에 수신과정에서 커다란 지연을 요구하게 된다. 따라서 차세대 무선 멀티미디어 통신에서 실시간으로 음성서비스나 화상서비스를 제공하는데 어려움이 많다. 본 논문에서는 기존의 터보 복호기를 변형하여 매 복호시 각각의 복호기에서 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 줄일수 있었다.

1. 서 론

1993년 Berrou 등에 의해 처음으로 발표된 터보코드(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[1]. Berrou는 AWGN 채널에서 인터리버의 크기가 65532이고 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

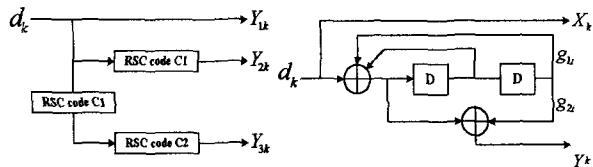
복호화 횟수 등이 있다. 본 논문에서는 복호화에 하나의 출력 값을 발생시키는 기존의 직렬 모드에 3개의 출력값을 내도록 변형하여 지연에 커다란 영향을 미치는 반복 복호 횟수를 가변적으로 줄이는 방법을 제안한다.

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. Turbo code의 기본 구조

2.1 터보 부호기(Turbo Encoder)

일반적인 터보코드에 대한 부호기의 구조가 [그림 1]에 나타나 있다. 부호기는 두 개의 RSC(Recursive Systematic Convolutional)부호와 이 두 RSC 부호를 연결시켜주는 인터리버로 구성되어 있다. 터보코드가 우수한 성능을 내기 위해서는 [그림 1]의 인터리버가 비균일(non-uniform) 인터리버여야 한다고 알려져 있다.[2]



[그림 1] 터보 부호기의 구조 [그림 2] RSC 부호(R=1/2, K=3)

[그림 2]에는 구속장의 길이(constraint length)가 3, 부호화율이 1/2인 부호어를 생성시키는 부호기가 나타나 있다. [그림 1,2]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 출력 사이의 관계를 살펴보면 다음과 같다.

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

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

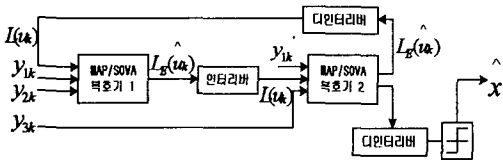
$$Y_{3k} = d_k + \sum_{i=0}^{K-1} r_i a_{2k-i} \pmod{2} \quad (3)$$

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 $\bar{}$ 는 원래 정보 d_k 의 인터리빙된 정보를 나타낸다.

2.2 터보 복호기(Turbo Decoder)

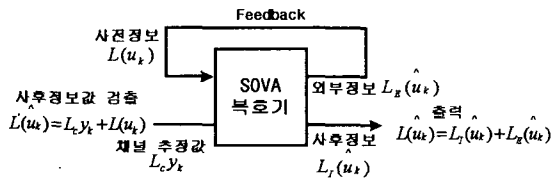
터보 코드의 복호화 알고리즘으로는 MAP(Maximum A Posteriori)알고리즘과 SOVA(Soft Output Viterbi Algorithm)알고리즘을 이용하는 방법이 있다. MAP 알고리즘을 사용함으로써 BER(Bit Error Rate)을 최소화시키고 SOVA보다 더 좋은 성능을 얻을 수 있다. 이에 반하여 SOVA의 BER 성능은 MAP보다 다소 떨어지는 단점이 있지만, 시퀀스 오류 확률을 최소화시킬 수 있고, 상대적으로 복호기 구현이 용이하다는 장점을 가지고 있다.

일반적인 터보 복호기의 구조는 [그림 3]과 같다.



[그림 3] 터보 복호기 구조

2.3 SOVA(Soft Output Viterbi Algorithm)



[그림 4] SOVA 복호기

[그림 4]는 SOVA를 이용한 반복 복호 구조를 나타내었다. 맨처음 연판정(soft-decision) 비터비 알고리즘

이 수행된다. 시간 k 에서 경로 m 일 확률은 다음과 같이 정의된다.

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

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.

$$L_c = 4 \frac{E_s}{N_0} \quad (6)$$

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 메트릭이며 P_k^m 은 현재 경로에 대한 확률을 표시한다. 그리고 경로 판정에 대한 LLR(log-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 LLR값을 구하기 위해 경로신뢰도(path reliability) Δ_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 Δ_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 갱신 규칙(update rule)은 G. Battail에 의한 방법(BR)[3]과 G. Hagenauer이 제안한 방법(HR)[4]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 LLR값 L_j^s 는 L_j^s 와 Δ_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 LLR값 L_j^s 는 Δ_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.

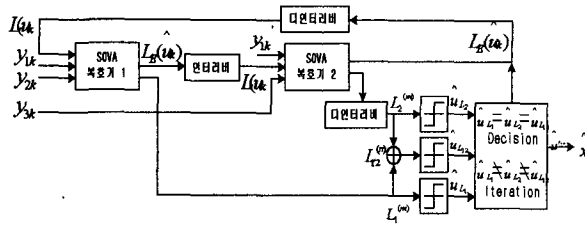
$L(\hat{u}_k)$ 는 채널 추정값 L_{ch} 로 얻어지는 내부정보 (intrinsic information) $L_I(\hat{u}_k)$ 와 사전정보 $L(\hat{u}_k)$ 로 구성된다. 사전정보는 외부정보 $L_E(\hat{u}_k)$ 와 동일한 값으로서 새로운 사전 추정을 위해 다음 복호기의 입력이 된다. 이 외부정보는 다음과 같이 구해질 수 있다.

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

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 매복호시 터보 복호기와 인터리버를 통과하게 된다. 이렇게 함으로써 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 5] 제안된 터보 복호기

변형 제안된 터보 복호기의 구조는 [그림 5]와 같다. [그림 5]에서 $L_1^{(m)}$ 과 $L_2^{(m)}$ 는 m 번째 복호화가 진행된 후 복호기1과 복호기2의 LLR(log-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 LLR값은 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 LLR값이 서로 다르다는 분포 특성을 이용하여 복호 횟수를 상당량 줄일 수 있는 방법을 본 논문에서 제안한다. 제안한 터보 복호 방식은 다음과 같다. $\hat{u}_{L_2}^{(m)}$ 와 $\hat{u}_{L_1}^{(m)}$,

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

$$\begin{cases} \hat{u}_{L_1}^{(m)} = \hat{u}_{L_2}^{(m)} = \hat{u}_{L_{12}}^{(m)} = \hat{u}_{L_2}^{(m)} \\ \hat{u}_{L_1}^{(m)} \neq \hat{u}_{L_2}^{(m)} \neq \hat{u}_{L_{12}}^{(m)} = \text{반복복호} \end{cases} \quad (13)$$

여기서 LLR 시퀀스의 sign을 고려해 보면 다음과 같다. 시퀀스 $L_1^{(m)}$ 과 $L_2^{(m)}$ 의 sign을 비트 단위로 서로 비교해 보면 두 시퀀스는 sign이 서로 같은 비트들과 sign이 서로 반대인 비트들로 구성되어 있다는 것을 알 수 있다. 따라서 두 시퀀스를 더한 $L_{12}^{(m)}$ 의 신뢰도 값은 sign이 같을 경우 그 비트에 대한 신뢰도가 더 커지고 sign이 다를 경우에는 신뢰도 값이 큰 쪽에 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.

IV. 실험 결과 및 고찰

본 논문의 채널 모델은 AWGN 채널이며 BPSK 변조를 가정하였다. 복호화 알고리즘은 SOVA를 사용하였고, 반복 복호 횟수는 5회, 정보 시퀀스의 프레임 크기 N 이 128, 256, 512 비트인 경우에 대하여 실험하였다. 터보 부호화기는 동일한 2개의 RSC 부호화기를 사용하였고 구성 부호화기의 생성 다항식은 (7,5)₈, 부호율은 1/2과 1/3이 사용되었다. 인터리버는 m 시퀀스 발생기를 이용한 랜덤 인터리버를 사용하였다.

[그림 6]와 [그림 7]은 각각의 프레임에 대해 부호율을 1/2과 1/3로 했을 때 제안한 방식의 복호기와 기존의 터보 복호기의 오차성능(BER)을 비교한 것인데, 성

능에 커다란 변화가 없음을 알 수 있다. 약간 성능에 변화를 보인 것은 복호기1과 복호기2의 LLR값을 판정한 값이 에러가 줄어드는 방향으로 수렴하지 못하고 동일한 위치의 비트에서 에러가 생겨 복호를 중단함으로써 생긴 성능의 변화이다. [그림 8]과 [그림 9]에서는 프레임 길이 256비트와 512비트에 대한 프레임 에러 성능(FER)을 보였다. [그림 10]과 [그림 11]은 제안한 방식으로 복호했을 때 감소된 평균 복호 횟수를 나타낸 것이다. 그림에서 보면 부호율이 1/3일때보다는 1/2일때 프레임 사이즈가 적을수록 복호 횟수 감소 정도가 큼을 알 수 있다. 프레임 길이에 따라 다소 차이가 있지만 부호율이 1/2일때 2.5~3.0dB에서 반복 복호 횟수가 평균 2.3 ~ 3.1회, 1/3일 때는 2.0~3.0dB에서 평균 2.2 ~ 3.2회 정도 복호횟수로도 기존의 터보 복호기의 성능을 유지할 수 있음을 알 수 있었다. 그리고 2.5dB 부터 서서히 평균 복호 횟수 감소 정도가 줄어 일정한 값으로 수렴함을 확인하였다.

V. 결론

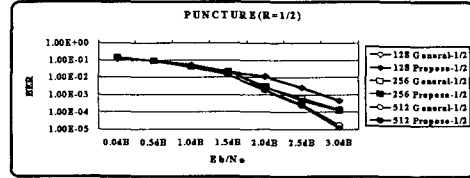
본 논문에서는 기존의 터보 복호기의 LLR 출력값 분포 특성을 이용하여 강판정된 값을 간단히 비교하는 방법을 제안함으로써 반복 복호 횟수를 상당량 줄일 수 있었다. 제안된 방법을 적용함으로써 비트 에너지 대 잡음의 전력비가 클수록 프레임 길이가 적을수록 반복 복호 횟수가 크게 감소됨을 알 수 있었다. 따라서 터보 부호에서 가장 커다란 단점으로 지적되어온 복호 지연을 줄임으로써 실시간 통신의 요구 사항에 어느 정도 부응할 것으로 본다.

향후 연구 과제는 복호기의 출력 시퀀스 분포에 대한 연구와 함께 다양한 채널 환경과 복호 알고리즘을 적용한 성능 평가가 이루어져야 하겠다.

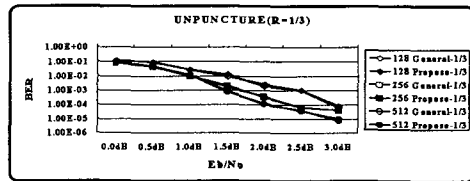
참고 문헌

- [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] G. Battail, "Ponderation des symboles decodes paralgorithem de Viterbi (in French)," Ann. Telecommun., Fr., 42, N 1-2, pp. 31-38, Jan. 1987
- [4] J. Hagenauer and P. Robertson, "Iterative

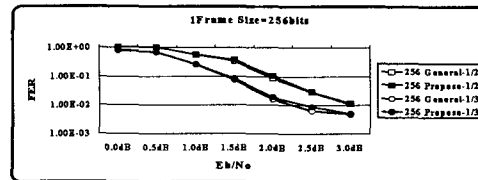
("TURBO") Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms," ITG-Fachberichte, V. 130, pp.21-29, 1995



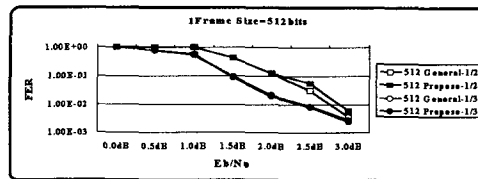
[그림 6] 부호율 1/2인 경우 BER 성능 비교



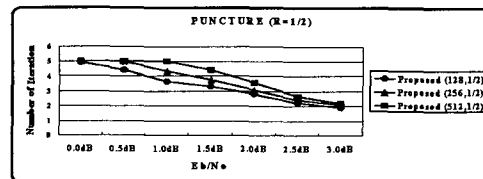
[그림 7] 부호율 1/3인 경우 BER 성능 비교



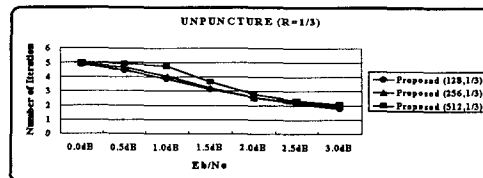
[그림 8] 프레임 길이 256 비트 FER 성능 비교



[그림 9] 프레임 길이 512 비트 FER 성능 비교



[그림 10] 제안된 복호기의 평균 복호 횟수(1/2)



[그림 11] 제안된 복호기의 평균 복호 횟수(1/3)