

병렬 구조의 복호기에서 LLR의 합을 사용한
터보 부호의 성능 분석
계환원, 구정우, 송홍엽
연세대학교 전기컴퓨터공학과

Performance analysis of turbo codes
for Parallel decoder structure using sum of LLR

Hwan Won Kye, Jung Woo Ku, Hong-Yeop Song
Dept. of Electrical and Computer Engineering, Yonsei Univ.

※ 본 논문은 한국 과학재단 특정기초연구(과제번호 97-0100-0501-3)지원사업에 의한 결과입니다.

요 약

터보 부호의 복호기는 복호화 알고리즘으로 MAP 알고리즘이나 SOVA 알고리즘을 이용하는 2개의 SISO(soft input soft output)복호기가 직렬 모드로 연결되어져서 반복적으로 정보들을 서로 교환하는 구조이다. 그런데 직렬 모드 복호기가 하나의 출력 시퀀스 생성하는 반면 병렬 모드 복호기는 동시에 2개의 출력 시퀀스를 제공하는 형태이다. 본 논문에서는 이러한 병렬 복호기의 2개의 출력 정보들은 변형함으로써 성능의 향상을 얻을 수 있는 방법을 제안한다. 바로 두 복호기의 LLR 출력값을 더하는 방법인데 제안한 방법의 병렬 복호기는 직렬 모드 복호기에 비해 구축장의 길이가 4이고 프레임 길이가 1024일 때 AWGN 채널에서 MAP의 경우는 0.15dB, SOVA의 경우에는 0.4dB의 성능 향상을 얻을 수 있었다. 또한 본 논문에서는 복호화 횟수가 증가함에 따라 제안한 방법의 복호기의 LLR 출력값 분포 특성과 성능 변화를 직렬 모드 복호기와 비교, 분석한다.

I. 서 론

무선통신 채널에서는 다중경로, 간섭, 잡음 등으로 인해 수신단에서 신호는 왜곡된다. 디지털 통신 시스템에서 이러한 신호의 왜곡으로부터 전송하는 정보를 보호하기 위해서 오류 정정 부호는 요소 기술이다. 1993년 Berrou등에 의해서 발표된 터보 부호(Turbo Code)는 병렬 연결 길쌈 부호화와 SISO(soft input soft output)복호화 알고리즘을 적절히 결합한 새로운 부호로서 그 성능의 우수함으로 인해 많은 연구의 대상이 되고 있다. 비교적 간단한 구성 부호들을 이용하지만 사소한계에 근접하며 AWGN 채널에서 큰 길이의 프레임(65,532)과 인터리버를 이용할 때 10^{-5} 의 비트 오류율을 얻는 E_b/N_0 는 0.7dB이다[1][2].

터보 부호를 설계할 때 고려할 사항으로는 ①구성 부호화의 선택의 문제, ②인터리버 설계와 프레임 길이 결정, ③프레임 끝에서 부호화의 종료문제 ④반복 복호화 횟수와 본 논문의 연구 주제인 복호기의 구조에 관한 문제등이 있다

일반적인 터보 복호기의 구조는 2개의 복호기가 직렬 모드로 연결되어져서 반복적으로 정보를 서로 교환하는 구조이나 다중 터보 부호일 경우에는 MAP 알고리즘을 사용하는 병렬 모드 복호기로 구성된다[3]. 본 논문에서는 복호화에 하나의 출력 값을 발생시키는 직렬 모드에 비해 2개의 출력값을 동시에 발생시키는 병렬 모드를 변형해 성능의 향상을 가져오는 방법을 제안한다.

2장에서는 터보 복호기의 복호화 진행 형태를 다루게 되고 3장에서는 본 논문에서 제안한 방법의 병렬 복호기를 소개한다. 4장에서는 모의 실험을 통해 기존의 터보 부호의 복호기에 대한 성능과 제안한 방법의 복호기와의 성능을 비교, 분석한다. 마지막으로 5장에서는 4장까지의 결과를 정리하고 앞으로의 연구 방향을 제시한다.

II. 터보 복호화의 진행 구조 [3][4]

2.1 직렬 모드(serial mode) 복호화

[2]에서 소개된 2개의 구성 부호를 사용하는 터보 부호에 대한 복호화 개념은 그림 1로 설명할 수 있다. 이것은 직렬 모드로 동작한다. 즉 Dec 2가 동작하기 전에 Dec 1이 정보들을 처리해서 출력값을 넘겨주고 Dec 2는 다시 출력값을 Dec 1에 넘겨주는 과정을 반복한다. 이러한 개념은 구성 부호가 3개 이상인 다중 터보 부호에 대해서도 적용된다.

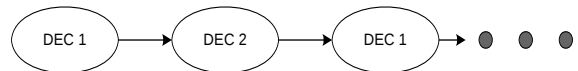


그림 1. 2개의 구성 부호에 대한 직렬 모드

2.2 병렬 모드(parallel mode) 복호화

또 다른 복호화 개념으로는 그림 2과 같이 병렬 모드인데 어떠한 시간에서도 모든 복호기가 동시에 병렬로 동작한다. 병렬 모드에서는 이전 시간의 자기 자신으로부터 오는 정보는 받지 않는데 그 이유는 복호화 과정이 진행되면서 positive 피드백이 되어 오히려 성능이 감소하기 때문이다. 병렬 모드를 구성 부호가 3인 경우에 대해서 적용해 보자. 이진 정보 u_k 는 $u_k = \{0, 1\}$ 이고 부호화된 후에 BPSK로 변조되어 AWGN 채널을 통과한다. 정보 시퀀스가 $u = \{u_1, \dots, u_N\}$ 이고 수신 시퀀스가 y 이며 복호기에서 MAP 알고리즘을 사용할 때 정보 비트 u_k 에 대한 LLR 출력값 $L(u_k)$ 는 식 1과 같다.

여기서 y_0 는 정보 시퀀스 u 의 채널 출력 시퀀스이며 y_i 는 3개의 잉여 시퀀스에 대한 채널 출력 시퀀스들이다

($i = 1, 2, 3$). 그리고 $\mathcal{L}_0$ 는 정보 시퀀스에 관계된 값

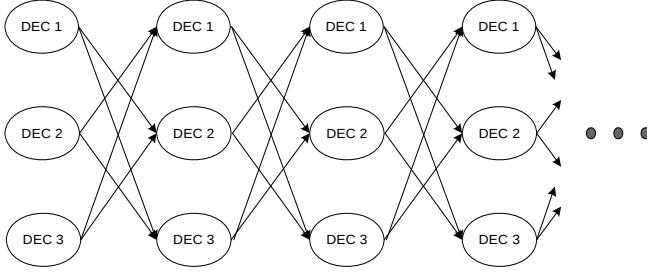


그림 2. 병렬 모드

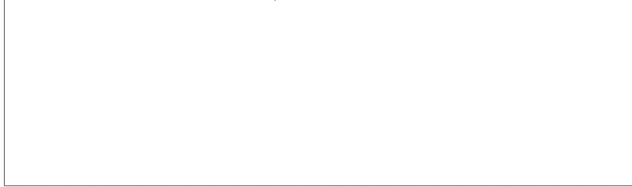


그림 3. 2개의 구성 부호에 대한 병렬 모드

$\mathcal{L}_0 = 4y_0E_s/N_0$ 이고 $\mathcal{L}_i$ 는 i 번째 복호기의 출력값에서 모든 입력 정보를 뺀 외부 정보 시퀀스이다. 다중 부호의 복호화 알고리즘으로 쓰이는 병렬 모드는 구성 부호가 2개일 때도 적용될 수 있다. 그림 3은 2개의 구성 부호에 대한 병렬 모드 구조이다. 2개의 구성 부호에 대한 병렬 모드는 2개의 패스가 서로 분리되어 있는 형태이므로 그림 1처럼 직렬 모드로 간소화될 수 있다. 그림 4는 2개의 구성 부호에 대한 병렬 모드 복호기이다[4].

III. 제안한 방법의 병렬 복호기

2개의 구성 부호의 경우 병렬 모드의 개념은 그림 3과 같다. 그런데 이것은 2개의 직렬 모드의 패스가 서로 결합된 형태이므로 각각의 패스를 따로 분리해 보면 그림 5처럼 2개의 직렬 모드 A와 B로 생각할 수 있다. 직렬 모드 복호기는 패스 A 나 B 중에서 한 패스만을 선택해서 복호화를 하지만 병렬 모드는 두 가지의 직렬 모드 패스에 대한 복호화가 동시에 진행된다. 따라서 일정한 횟수의 반복적인 복호화 과정이 진행된 후에 복호기의 출력값을 강 판정(hard decision)해서 정보 비트를 복호화 할 때 직렬 모드는 A나 B중 한 패스의 출력값 만을 사용한다. 그러나 병렬 모드는 A와 B 패스에서 각각 발생된 서로 다른 값의 분포를 갖는 2 개의 출력값 모두를 사용할 수 있으므로 한 개의

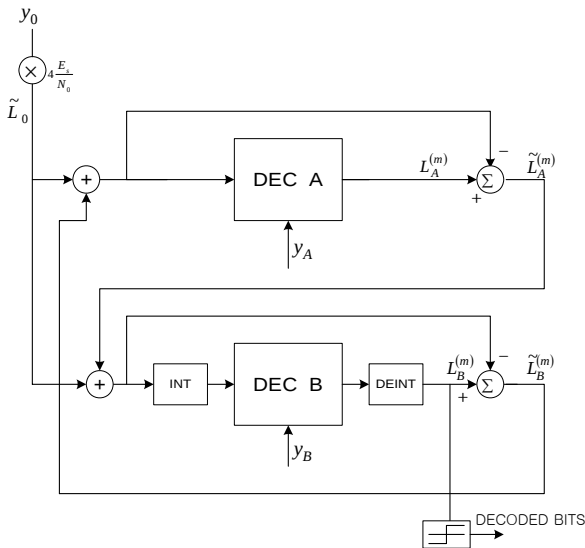


그림 4. 2개의 구성 부호에 대한 일반적인 병렬 복호기

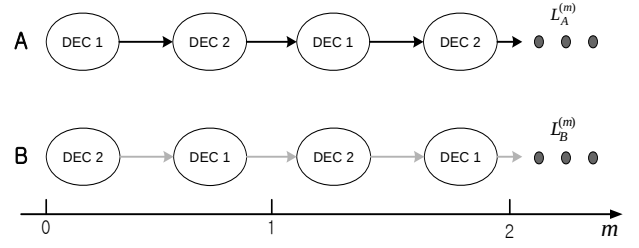


그림 5. 2개의 구성 부호에 대한 병렬 모드의 분리 형태

결과값 만을 이용할 수 있는 직렬 모드에 비해 정보 비트에 대해 더 많은 정보를 가지고 있게된다. 3장에서는 이러한 병렬 모드 복호기를 변형함으로써 성능의 향상을 줄 수 있는 복호화 방법을 제안한다. [3],[4]에서는 복호화 알고리즘으로 MAP을 사용하는 병렬 모드를 소개되었으나 본 논문에서는 SOVA를 사용하는 2개의 구성 부호에 대한 제안한 방법의 병렬 모드를 주로 다룬다.

3.1 LLR의 합을 이용하는 병렬 복호 방법

그림 5에서 $L_A^{(m)}$ 는 m 번의 반복 복호화가 진행된 패스 A의 LLR 출력값이다. $L_B^{(m)}$ 는 패스 B의 LLR 출력값이다. 두 패스의 출력 시퀀스 $L_A^{(m)}$ 와 $L_B^{(m)}$ 는 서로 다른 값들의 분포이다. 따라서 위의 출력 시퀀스들을 강 판정한 각각의 시퀀스 $\hat{u}_A^{(m)}$, $\hat{u}_B^{(m)}$ 도 서로 다른 분포이며 각각의 오류를 계산해 보면 두 패스의 오류값들이 서로 다르다. 일반적으로 $\hat{u}_A^{(m)}$ 와 $\hat{u}_B^{(m)}$ 의 분포들은 서로 다르지만 반복 횟수 m 이 증가할수록 두 값은 신뢰도가 높아지는 분포로 새롭게 갱신되고 오류가 줄어드는 분포로 수렴하게 된다. 기존의 직렬 복호기는 $\hat{u}_A^{(m)}$ 와 $\hat{u}_B^{(m)}$ 중 하나만을 이용해서 복호화 하였지만 두 가지 모두를 이용하기 위해 LLR 시퀀스의 sign을 고려해보자.

시퀀스 $L_A^{(m)}$ 와 $L_B^{(m)}$ 의 sign을 비트 단위로 서로 비교해 보면 두 시퀀스는 sign이 서로 같은 비트들과 sign이 서로 반대인 비트들로 구성되어 있다. 두 시퀀스의 sign이 서로 같은 비트들은 그 sign으로 강 판정해서 복호화 하면 정보 비트와 값이 같을 확률이 높지만 두 시퀀스의 sign이 반대인 비트들은 상대적으로 신뢰도가 많지 않다. 따라서 두 시퀀스의 sign이 서로 반대인 비트들의 경우 두 비트 중에 값이 큰 것에 신뢰도를 많이 두고 작은 것에 신뢰도를 덜 주는 방법으로서 두 비트들을 더해준다. 이렇게 두 시퀀스들을 더해주면 새로운 시퀀스 $L_{sum}^{(m)}$ 가 만들어지는데 $L_{sum}^{(m)}$ 의 각각의 비트 sign은 $L_A^{(m)}$ 와 $L_B^{(m)}$ 의 비트값 중에서 값이 큰 것의 sign과 같다. 여기서 두 LLR 시퀀스의 합 $L_{sum}^{(m)}$ 은 다음과 같이 계산된다.

$$L_{sum}^{(m)} = L_A^{(m)} + L_B^{(m)} \quad (2)$$

이제 m 번의 반복 복호화 후에 시간 k 비트의 두 LLR 출력값의 합을 정보 시퀀스의 채널 출력값과 사전 정보값과 외부 정보값으로 표현하면 다음과 같다.

여기서 $\mathcal{L}_0(u_k)$ 는 정보 비트에 관계된 값이고 $L_{P_i}(u_k)$ 는 m 번째 복호화 때 복호기 i 의 사전 정보, $\mathcal{L}_i(u_k)$ 는 m 번째 복호화 때 발생된 복호기 i 의 외부 정보값이다 ($i = A, B$). 채널의 상태가 좋은 경우에는 복호화의 결과는 $\mathcal{L}_0(u_k)$ 에 의존하지만 채널의 상태가 열악하면 복호화의 값은 사전 정보와 외부 정보의 값에 의존한다[5].

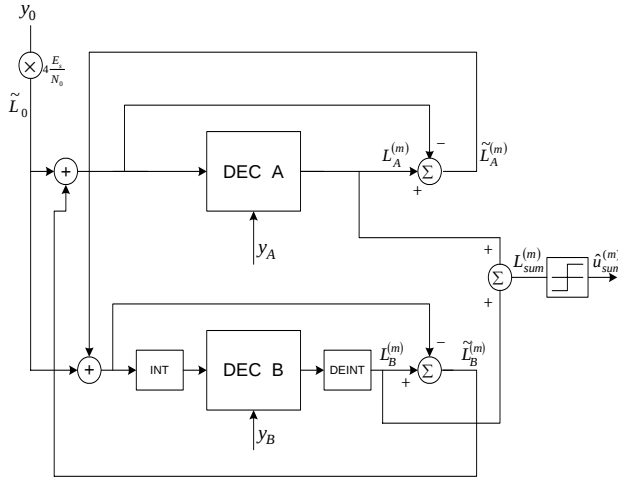


그림 6. 제안한 LLR의 합을 이용하는 병렬 복호기

병렬 모드 복호기의 2개의 LLR 출력값의 합 $L_{sum}^{(m)}$ 를 강판정한 $\hat{u}_{sum}^{(m)}$ 는 직렬 모드의 $\hat{u}_A^{(m)}$ 와 $\hat{u}_B^{(m)}$ 에 비해서 더 작은 오류값을 갖는 분포임이 본 논문의 실험을 통해 확인되었다. LLR의 합을 사용하도록 병렬 복호기를 변형하면 그림 6처럼 나타낼 수 있다. 제안한 구조는 그림 4의 구조보다 두 LLR 값을 더하는 계산량이 추가적으로 소요되지만 성능의 향상을 가져온다.

3.2 두 LLR 합의 분포 특성

그림 5에서 두 직렬 페스의 출력 시퀀스 $L_A^{(m)}$ 와 $L_B^{(m)}$ 는 서로 다른 값들의 분포이며 반복 횟수 m 이 증가할수록 두 값은 신뢰도가 높아지는 분포로 새롭게 갱신되지만 복호기에서 발생된 외부정보들이 이전 복호기의 출력값들과 서로 연관성을 가지게 되면서 성능 향상의 폭이 작아지고 결국 더 이상 성능 향상이 없는 오류값으로 수렴하게 된다. 그러나 $L_{sum}^{(m)}$ 의 오류값은 직렬 모드의 LLR 출력값 $L_A^{(m)}$ 와 $L_B^{(m)}$ 가 수렴하게 되는 오류값보다 더 작은 값을 갖는다. 즉 m 이 증가하면 $L_{sum}^{(m)}$ 의 오류값도 결국 수렴하게 되지만 $L_A^{(m)}$ 와 $L_B^{(m)}$ 가 수렴하는 오류값보다 더 작은 값으로 수렴하게 되는 분포를 갖는다. 그림 7은 AWGN 채널에서 SOVA의 경우에 $L_A^{(m)}$ 을 복호화하는 직렬 모드와 두 LLR의 합 $L_{sum}^{(m)}$ 을 복호화 하는 제안한 방법에 대해 복호화 횟수에 따른 성능을 비교하고 있다. 그림에서 점선은 LLR의 합을 이용하는 방법의 결과를 나타내고 있다. 직렬 모드의 경우 반복 횟수 m 이 15이상일 때는 성능 향상이 거의 없는 오류값으로 수렴한다. m 이 5이고 E_b/N_0 가 2dB 이상일 때 제안한 방법의 오류값은 m 이 15이상인 직렬 모드 복호기의 오류값보다 작으며 m 이 10이상일 때 제안한 방법이 모든 E_b/N_0 에 대해서 직렬 모드가 수렴하는 오류값보다 더 작은 오류값을 갖는 것을 알 수 있다. 따라서 $L_{sum}^{(m)}$ 은 직렬 모드의 복호화 횟수 m 보다 작은 반복 횟수 m 에서 $L_A^{(m)}$ 와 $L_B^{(m)}$ 가 수렴하게 되는 오류율보다 더 작은 오류율을 갖게된다.

IV. 실험 결과 및 고찰

본 논문의 채널 모델은 AWGN 채널이며 BPSK 변조를 가정하였다. 복호화 알고리즘은 MAP 과 SOVA를 사용하였고 반복 횟수는 5, 정보 시퀀스의 프레임 크기 N 이 256, 512, 1024, 4096 비트인 경우에 대해 실험하였다. 터보 부호화기는 동일한 2개의 RSC 부호화기를 사용하였고 주로 부호화기의 피드백 생

그림 7. 반복 횟수에 따른 직렬 모드와 LLR의 합을 이용하는 병렬 모드와의 성능 비교
(구속장의 길이 4, 프레임 길이 1024, 부호율 1/3)

성 다항식은 $G_0 = (15)_8$, 잉여 비트의 생성 다항식은 $G_0 = (17)_8$ 이며 8진수 형태로 표현하였다. 구속장의 길이 $K = 3, 4, 5$ 에 대해서 각각 실험하였고 인터리버는 m 시퀀스 발생기를 이용한 랜덤 인터리버, 부호율은 1/3과 1/2 이 사용되었다.

그림 8의 (8-a)와 (8-b)에서 사용된 터보 부호의 프레임 길이는 1024, 구속장의 길이는 4, 부호율이 1/3과 1/2일 때 MAP과 SOVA알고리즘에 대해서 제안한 LLR의 합을 이용하는 복호기와 직렬 복호기의 성능을 비교한다. 그림들에서 점선들은 LLR의 합을 이용해 복호화한 결과들이며 “LLR sum”으로 표기했다. SOVA가 MAP보다 제안된 방법으로 인한 이득이 많이 나타남을 알 수 있다. 제안한 방법은 SOVA 경우 $p_b = 10^{-4}$ 에서 약 0.4dB의 이득이 나타났으며 MAP은 약 0.15dB의 성능 향상이 있었다. 그림 9의 (9-a)는 SOVA를 이용할 때 프레임 길이에 따른 제안한 복호기의 성능 결과를 비교하고 있으며 부호율 1/3이고 구속장의 길이가 4인 부호화기가 사용되었다. 프레임 길이가 $N = 256$ 일때 제안한 방법은 $p_b = 10^{-4}$ 에서 약 0.6dB의 이득이 있으며 $N = 512$ 일 때는 0.45dB, $N = 4096$ 일 때는 약 0.3dB의 이득이 있었다. 따라서 프레임 길이가 적을수록 큰 이득이 있음을 알 수 있다. 그림 9의 (9-b)는 복호화 알고리즘으로 SOVA를 이용하고 부호율은 1/3, 프레임 길이가 1024이며 표 1의 구속장을 쓰는 터보 부호이다. 제안한 방법은 $p_b = 10^{-4}$ 에서 구속장의 길이가 $K = 3, 4$ 일 때는 0.4dB의 이득이 있었고 $K = 5$ 일 때는 약 0.6dB의 성능 향상이 생겼다.

V. 결론

본 논문에서는 병렬 모드의 두 복호기들의 최종적인 LLR 출력 시퀀스들을 더하는 방법을 제안함으로써 성능 향상을 얻을 수 있었다. 또한 SOVA의 경우 반복 복호화 횟수에 따라서 제안한 방법의 LLR 출력값의 분포 특성, 오류값의 수렴, 성능 변화를 직렬 모드와 비교, 분석하였다. 제안된 방법은 직렬 모드에 비해 SOVA의 경우에는 프레임 길이가 작을수록 성능의 이득이 크게 나타났다. 또한 제안한 방법이 직렬 모드가 수렴하는 오류율보다 더 작은 오류율로 수렴하는 사실이 확인되었다. 본 논문에서는 두 병렬 복호기의 최종 출력 시퀀스가 합해지면 더욱 신뢰성 있는 분포의 출력값으로 갱신된다는 사실이 실험을 통해 확인되었지만 수학적으로 명확하게 규명되지 않았다. 따라서 앞으로 좋은 성능의 향상을 줄 수 있는 복호기의 출력 시퀀스의 분포에 대한 연구가 이루어져야 하겠다.

제안된 방법의 병렬 복호기는 성능의 향상을 주는 장점이 있지만 동시에 2개의 직렬 모드 복호기를 복호화하는 것과 같은 복잡도가 소요되는 단점이 있으므로 복호기를 구성할 때 복잡도와 성능을 고려한 절충이 필요하다. 터보 부호는 현재 IMT-2000에서 길쌈 부호, RS 부호와 길쌈 부호의 직렬 연결 부호와 함께 오류 정정 부호의 표준안 중의 하나로 제안되고 있는데 이를 위해서는 보다 간단한 복잡도와 시간 지연이 적은 복호화 방법에 대한 연구가 이루어 져야 한다.

참고 문헌

- [1] C.Berrou, A.Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding:turbo-codes," in *Proc. ICC'93*, pp. 1064-1070
- [2] C.Berrou, A.Glavieux, and P.Thitimajshima, "Near optimum error correcting:coding and decoding:'Turbo-Codes," *IEEE Trans. Commun.*, vol.44, no.10, pp.1261-1271, Oct. 1996
- [3] S.Benedetto, D.Divsalar and F.Pollara, "Soft-Output Dcoding Algorithm in Iterative Decoding of Turbo Codes", TDA progress rep. 42-124, Jet Propulsion Lab., Pasadena, CA, Feb. 15, 1995, pp. 63-86
- [4] D.Divsalar and F.Pollara, "Multiple Turbo Codes for Deep-Space Communications", TDA progress rep. 42-121, Jet Propulsion Lab., Pasadena, CA, May 15, 1995, pp. 66-67
- [5] J. Hagenauer, Iterative decoding of binary block and convolutional cods, *IEEE Trans.Info. Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996

(8-b) 부호율이 1/2 인 경우
그림 8. LLR 합을 이용하는 복호기의 성능

표 1. 각 구속장에서의 생성 다항식

구속장의 길이 (K)	생성 다항식	
	G_0	G_1
3	$(7)_8$	$(5)_8$
4	$(15)_8$	$(17)_8$
5	$(23)_8$	$(35)_8$

(9-a) 프레임 길이에 따른 성능

(8-a) 부호율이 1/3인 경우

(9-b) 구속장의 길이에 따른 성능
그림 9. LLR의 합을 이용하는 복호기의 성능