

프레임 크기가 작은 터보 코드의 복잡도에 대한 성능 분석

정회원 김 연구*, 고 영 훈**, 김 남*

Performance analysis on the complexity of turbo code with short frame sizes

Yeun-Goo Kim*, Young-Hoon Ko**, Nam Kim* *Regular Members*

요 약

PCCC(Parallel Concatenated Convolutional Codes: 터보 코드)는 블록 크기가 크면 클수록 성능이 우수한 것으로 잘 알려져 있다. 본 논문은 차세대 이동 통신 시스템에서 프레임 크기가 작은 음성/제어 프레임을 이용하여 터보 코드의 성능을 검증하여 보고, 비슷한 디코딩 복잡도에서 터보 코드와 컨벌루션 코드의 성능과 시스템에 적용하기 위한 조건들을 고려하였다. 그 결과 터보 코드는 프레임 크기가 작은 조건에서 반복 횟수 3회만으로도 차세대 통신 시스템에 적합한 10^{-3} 이상의 좋은 성능을 나타내었다. 그러나, 10^{-3} 의 BER을 기준으로 비슷한 복잡도를 고려하였을 때는 부호화율이 1/2인 터보 코드(K = 5)의 성능은 낮은 E_b/N_0 에서 컨벌루션 코드(K = 9)보다 우수하고, K = 3인 터보 코드의 성능은 K = 7인 컨벌루션 코드보다 우수하였다. 또한 부호화율이 1/3인 경우 K = 3 또는 5인 터보 코드가 부호화율이 1/2인 터보 코드와 비슷한 성능을 나타내었다.

ABSTRACT

It is well known that Parallel Concatenated Convolutional Codes(turbo codes) has a good performance for long block sizes. This thesis has analyzed the performance of turbo code which is based on voice or control frames with short frame sizes in the future mobile communication system. Also, at the similar decoding complexity, the performance of turbo code and convolutional codes in the speech/control frames, and the applicability of this system are considered.

As a result, turbo code in short frame sizes present the performance of a BER of 10^{-3} or more over 3 iterations in the future mobile communication system. However, at a BER of 10^{-3} , if the same complexity is considered, the performance of rate 1/2 turbo code with K = 5 is better than that of convolutional code with K = 9 at low E_b/N_0 , and the performance of turbo code with K = 3 is superior to that of convolutional code with K = 7. Rate 1/3 turbo code with K = 3 and 5 have similar to performance of rate 1/2 turbo code.

I. 서 론

오류 정정 부호는 페이딩, 간섭, 잡음 등에 대항하기 위해 셀룰라, 이동위성 시스템 같은 무선 통신 시스템에서 사용되고 있다. 이러한 시스템들은 일반적으로 짧은 프레임(300비트이하)에서 정보를 전송

하는데, 전송프레임의 크기는 오류 정정 부호의 선택에 따라 제한된다. 일반적으로 무선통신에서는 컨벌루션 코드가 주로 사용되는데, 성능면에서 컨벌루션 코드는 정보 프레임 크기가 코드의 구속장(constraint length, K)보다 커도 프레임 크기에 영향을 주지 않기 때문이다.

* 충북대학교 전기전자공학부

논문번호 : 98477-1103, 접수일자 : 1998년 11월 3일

** 협성대학교 컴퓨터공학과

최근, RSC(recursive systematic convolutional codes)를 이용한 터보 코드가 새로 제안되었다^[1]. 이 코드의 인코더는 원래 정보의 인터리빙된 데이터를 입력으로 갖는 다중 RSC 인코더들로 구성되어 있으며 블록 크기가 큰 경우에 채널용량에 근접하는 성능을 갖는다. 사실 이러한 종류의 코드 성능은 인코딩된 프레임 크기가 증가할수록 향상되는데, 이것은 터보 코드에 포함된 인터리빙 이득이 증가하기 때문이다^[2]. 그러나 입력 비트에 대한 디코딩 복잡도는 프레임 크기와 관계없이 같다.

프레임 크기가 작은 인코더에 대한 터보 코드의 성능에 대한 연구는 계속되고 있고 셀룰라나 이동 위성 통신에서 사용하는 프레임 크기에 대해서도 좋은 성능을 나타내는 것으로 알려졌다^[3]. 그러나, 터보 코드에서 사용되는 디코더는 컨벌루션 코드의 ML(maximum likelihood) 디코딩보다 복잡도가 훨씬 큰 반복 디코더이다. 예를 들면, P. Jung에 의해 사용된 터보 디코딩 방법은 각각의 구성코드(component code)가 ML 디코딩보다 복잡도가 약 54배에 달하고, 복잡도가 약 32배에 해당하는 긴 구속장을 갖는 컨벌루션 코드는 BER이 10^{-3} 에서 0.25dB 정도 손실이 갖는다^[3].

본 논문에서는 AWGN 채널상에서 부호화율과 구속장 변화에 따른 프레임 크기가 작은 터보 코드의 성능을 검증하여 보고, 비슷한 복잡도를 갖을 때의 터보 코드와 컨벌루션 코드의 성능과 이러한 시스템 적용하기 위한 조건들을 고려하여 보았다.

본 논문의 구성은 II장에서는 터보 코드의 인코더와 디코더를 중심으로 개념적으로 설명하고 III장에서는 반복 디코딩에 따른 터보 코드와 컨벌루션 코드의 복잡도 관점에서 부호화율과 구속장의 변화에 따라 비교한다. IV장에서는 이러한 이론을 바탕으로 한 모의실험을 통해 프레임 크기가 작은 터보 코드의 성능과 반복 디코딩에 따른 성능, 그리고 복잡도에 따른 컨벌루션 코드와 터보 코드를 비교 분석하고자 한다.

II. 터보 코드의 구조

터보 코드의 기본 구조는 두 개의 RSC 코드와 인코더를 분리해주기 위한 인터리버로 구성되어 있으며 이는 두 번째 인코더의 입력에 들어갈 정보를 재배열하는 역할을 한다. 또한, RSC 코드를 사용하는 이유는 낮은 신호대 잡음비에서 기존의 컨벌루션 코드보다 좋은 성능을 보이기 때문이다.

터보 코드는 두 인코더가 같은 입력을 사용하기는 하지만, 시간 지연과 인터리버가 첨가됨으로서 실제적인 입력 열의 순서가 바뀌게 된다. 인코더의 입력단이 하나이고 두 개의 RSC를 사용하게 되면 출력단에서 비트들을 모두 전송하게 되면 터보 코드의 코드율은 1/3이 된다. 하지만 인코더의 출력단에서 펄칭(puncturing)을 사용하면 코드율은 1/2로 변형되며, 펄칭은 정보 비트에 대해 펄칭하지 않고 잉여비트(redundant bits)에 대해서만 적용된다. 그림 1은 터보 코드의 인코더 구조를 나타낸 블록도이다.

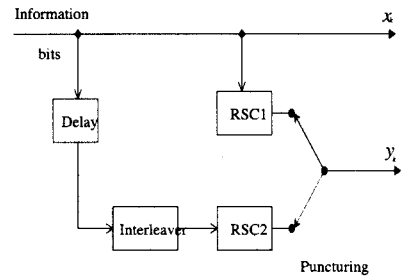


그림 1. 코드율이 1/2 인 터보 코드 인코더

AWGN 채널 환경하에서 부호화율이 1/2일때 수신측에서 수신하는 데이터는 x_k 와 펄칭에 의한 y_{1k}, y_{2k} 가 반복적으로 수신되고 부호화율이 1/3일 경우에는 x_k, y_k 모두 수신된다. 수신된 데이터는 데이터와 반복 디코딩 알고리즘에 의한 부가 정보(extrinsic information) L_{21}^e 가 첫 번째 MAP 디코더에 입력되고, L_{21}^e 의 초기값은 0이다. 첫 번째 MAP 디코더에 의해 처리된 L_{12}^e 는 두 번째 MAP 디코더의 입력으로 수신된 데이터와 함께 입력된다. 이렇게 해서 두 번째 MAP 디코더에서 출력된 L_{21}^e 값은 다시 첫 번째 MAP 디코더의 입력값이 되고 나머지 출력은 원래의 정보를 산출하는데 사용된다.

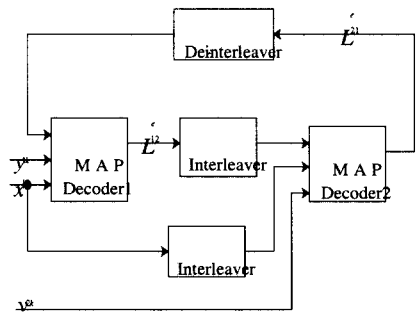


그림 2. MAP을 이용한 터보 코드의 디코더

이러한 과정을 반복 횟수가 만큼 반복함으로써 에러 성능을 향상시키고 원래의 정보에 가까운 값을 얻게 된다. 그러나, 복잡도는 반복을 거듭할수록 증가하게 된다. 그림 2는 이러한 터보 코드의 디코더 구조를 나타내었다.

III. 터보 코드와 컨벌루션 코드의 복잡도

부호화율(code rate)을 $1/n$, 트래리스 상태(state)를 S , 정보 블록 길이를 N 이라 하면, 컨벌루션 코드의 디코딩 복잡도는 다음과 같은 두가지 가정에 의해 계산할 수 있다.

첫째로 경로 메트릭의 업데이트에서 발생하는 상태에 대한 연산(operation per state)이다. 컨벌루션 코드에는 각 상태마다 덧셈(add), 비교(compare), 그리고 선택(select)이라는 중요한 세 가지 연산이 있다. 두번째는 메트릭 계산에서 stage에 대한 연산(operation per stage)이다. 각각의 가지 메트릭을 계산하기 위해서는 $n-1$ 개의 덧셈과 2^{n-1} 의 가지 메트릭이 필요하기 때문에 $2^{n-1}(n-1)$ 연산이 된다. 그러므로 컨벌루션 코드의 디코딩 알고리즘인 Viterbi 알고리즘의 stage에 대한 복잡도는 상태에 대한 각각의 세 가지 연산(ACS)과 가지 메트릭 계산에서의 연산이 존재하므로 식(1)과 같이 나타낼 수 있다.

$$C_{conv} = [3S + 2^{n-1}(n-1)] \quad (1)$$

MAP 알고리즘을 사용한 터보 코드에 대해 stage에 대한 복잡도를 살펴보면 가지 메트릭 계산, 상태에 대한 α, β, γ 계산, 그리고 LLR(Log-Likelihood Ratio) 계산으로 크게 나눌 수 있다.

가지 메트릭(branch metric) 계산은 컨벌루션 코드와 마찬가지로 $2^{n-1}(n-1)$ 연산이 필요하고, 상태에 대한 α 의 계산은 $\log[\exp(a) + \exp(b)]$ 을 계산하기 위해 두 개의 덧셈과 하나의 연산이 요구된다. 상태에 대한 β 의 계산은 α 와 같은 연산이 필요하고, γ 의 계산은 S 의 수와 같게된다. 마지막으로 MAP 값(Log-Likelihood Ratio values)은 $\log[\exp(a_1) + \exp(a_2) + \dots + \exp(a_S)]$ 을 계산하기 위한 정보 비트 0, 1에 대한 $S-1$ 연산과 하나의 나눗셈 연산을 가지므로 $2S-1$ 이 요구된다. 여기서 나눗셈 계산은 단일 연산으로 처리한다.

두개의 MAP 디코더에서는 이러한 연산을 1번 반복하기 때문에 디코더의 stage에 대한 복잡도는

식(2)와 같이 나타낼 수 있다.

$$C_{turbo} = 2I[9S + 2^{n-1}(n-1) - 1] \quad (2)$$

IV. 모의 실험 및 결과 분석

본 장에서는 구축장이 3일때와 5일때 부호화율이 1/2, 1/3인 터보 코드를 모의 실험하였다. 블록 크기는 무선 통신에서 일반적으로 사용되는 음성/제어 프레임 크기를 고려하였고 인코더의 구조는 그림 1과 같다. 본 논문에서는 구축장 K 는 3과 5만을 고려하였고, 사용된 생성 벡터는 가장 좋은 성능을 나타내는 7/5와 33/31을 선택하였다⁴⁾. 채널 조건은 AWGN을 적용하였고, 변조 방식은 BPSK, 인터리버는 랜덤 인터리버를 사용하였다. 터보 디코더의 반복 횟수는 4 번으로 하였다. 다양한 블록 크기를 비교하기 위해 같은 인터리버를 사용하였고, 프레임 크기에 따라 인터리버 크기가 결정되도록 모의 실험하였다. 컨벌루션 코드도 같은 AWGN 채널 조건 상에서 상태를 변화시켜가며 모의 실험하였다.

그림 3과 4는 AWGN 채널 환경에서 블록 크기가 192비트이고 생성다항식이 7/5일때 부호화율의 변화에 따른 터보 코드의 에러 성능을 나타내었다.

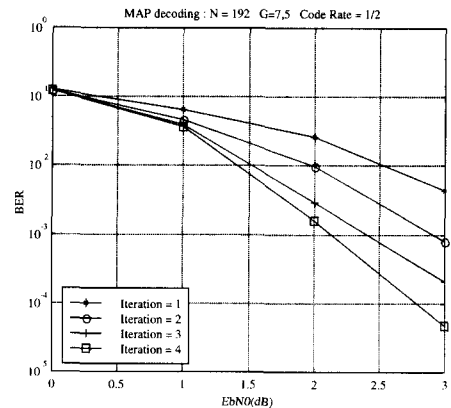


그림 3. 터보 코드의 반복횟수에 따른 성능(부호화율 1/2)

그림 3은 BER이 10^{-3} 에서 반복 2와 4사이에 0.8dB의 성능 차이를 볼 수 있고, 그림 4에서는 반복 2와 4사이에 0.6dB의 차이를 볼 수 있다. 이것은 부호화율이 작을수록 오류 정정 능력이 향상되기 때문이다.

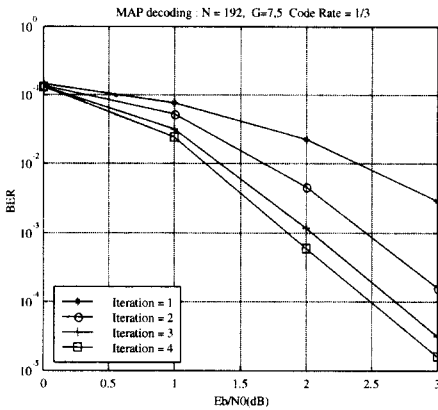


그림 4. 터보 코드의 반복횟수에 따른 성능(부호화율 1/3)

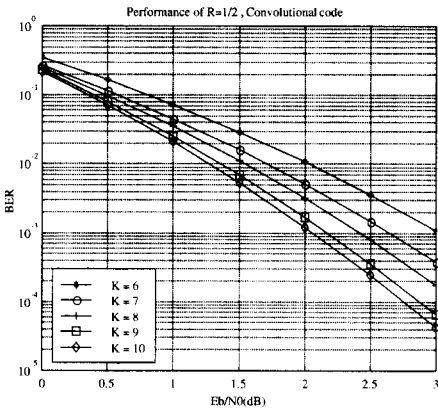


그림 5. 구속장에 따른 컨벌루션 코드의 성능(부호화율 1/2)

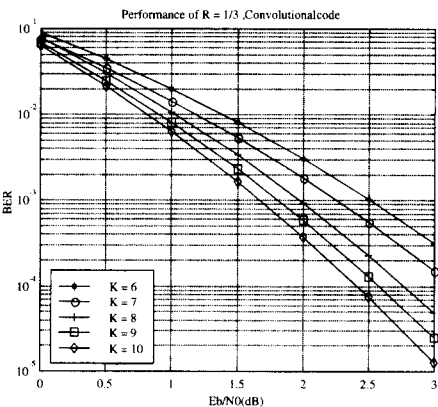


그림 6. 구속장에 따른 컨벌루션 코드의 성능(부호화율 1/3)

그림 5와 6은 부호화율이 1/2, 1/3일 때 구속장이

6 ~ 10까지 변하는 컨벌루션 코드의 성능을 나타내었다. BER이 10^{-3} 에서 부호화율 1/2과 1/3사이에 0.35dB 정도의 성능 차이를 보이고 있다.

표 1. 컨벌루션 코드와 터보 코드의 디코딩 복잡도 비교

구속장	부호화율 = 1/2				
	반복횟수				
	1	2	3	4	Conv.
3	74	148	222	296	14
5	290	580	870	1160	50
6	578	1156	1734	2312	98
7	1154	2308	3462	4616	194
8	2306	4612	6918	9224	386
9	4610	9220	13830	18440	770
10	9218	18436	27654	36872	1538

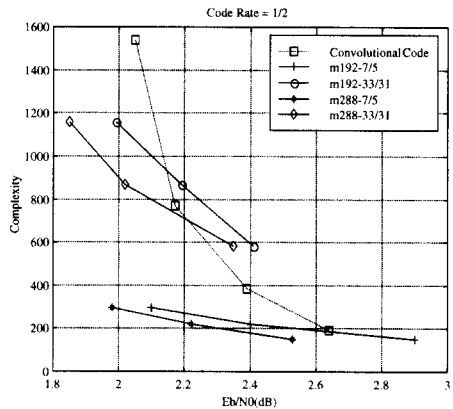


그림 7. 10^{-3} BER에서 컨벌루션 코드와 터보 코드의 디코딩 복잡도 비교 (부호화율 1/2)

그림 7과 그림 8은 BER이 10^{-3} 에서 구속장이 3, 5 일 때 터보 코드와 구속장이 6 ~ 10까지 변하는 컨벌루션 코드의 복잡도를 비교했다. 그림 7은 부호화율이 1/2 일 때 구속장과 블록 크기의 증가에 따른 모의 실험결과이다. 같은 복잡도 관점 (complexity = 800)에서 블록크기가 192비트일 때 K = 9인 컨벌루션 코드와 반복 횟수가 4인 터보 코드를 비교해 볼 때 BER이 10^{-3} 에서 터보 코드가 컨벌루션 코드보다 0.1 dB 정도의 성능이 우수하고 K = 3 인 터보 코드인 경우에는 K = 7 (complexity = 200)의 컨벌루션 코드 성능보다 0.44 dB 정도 성능이 향상된 것을 알 수 있다. 또한,

E_b/N_0 가 2 ~ 2.2dB사이에서 $K = 9$ 인 컨벌루션 코드의 복잡도(complexity = 770)보다 $K = 3$ 인 터보코드가 작은 복잡도(complexity = 296)로 같은 성능을 얻는 것을 볼 수 있다. 그림 8은 부호화율이 1/3 일때 컨벌루션 코드와 터보 코드의 복잡도 결과이다. 그림 7과 같은 조건에서 $K = 5$ 인 터보 코드가 1.8dB에서 $K = 9$ 인 컨벌루션 코드와 비슷한 성능을 갖고 $K = 3$ 인 터보 코드는 2.27 dB에서 $K = 7$ 과 같은 성능을 나타내었다.

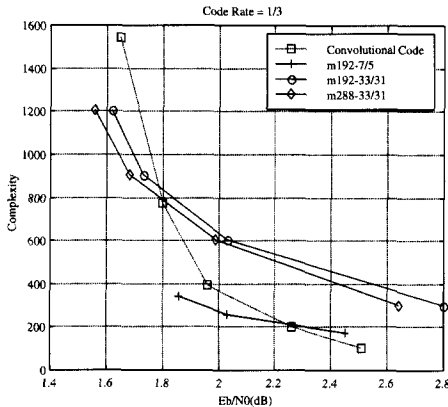


그림 8. 10^{-3} BER에서 터보 코드와 컨벌루션 코드의 디코딩 복잡도 비교 (부호화율 1/3)

이것으로 시스템 조건과 채널환경에 맞추어 적절한 구속장과 채널 코드, 부호화율의 선택이 필요함을 알 수 있고, 복잡도 관점에서 볼 때 낮은 E_b/N_0 에서 터보 코드가 우수함을 알 수 있다.

V. 결론

본 논문에서는 차세대 이동 통신 시스템에서 사용될 음성/제어 프레임 크기를 이용하여 복잡도 관점에서 터보 코드와 컨벌루션 코드를 비교해 보았다. 실험은 AWGN 환경하에서 부호화율, 구속장의 크기, 반복횟수를 고려하였고, 인터리버는 정보를 랜덤하게 분포시키는 랜덤 인터리버를 사용하였다. 디코딩은 반복에 의한 MAP알고리즘의 사용하여 성능을 검증해 보았으며 컨벌루션 코드 또한 다양한 구속장을 가지고 연판정(soft decision) 디코딩 방법을 택하여 실험하였다. 그 결과 같은 구속장에서 터보 코드는 컨벌루션 코드에 비해 반복 횟수와 인터리버 크기를 늘림으로써 성능의 향상을 볼 수 있고, 복잡도 관점에서는 부호화율이 1/2인 경우 2.2dB이

하에서, 부호화율이 1/3에서는 1.8dB이하에서 터보 코드의 성능이 우수하였다. 즉, 낮은 E_b/N_0 에서 터보 코드가 성능이 우수함을 알 수 있다. 또한 같은 복잡도 관점에서 블록 크기가 192비트, 반복횟수가 4, 부호화율 1/2인 경우 $K = 5(33/31)$ 인 터보 코드가 $K = 9$ 인 컨벌루션 코드보다 0.1dB정도의 성능 향상이 있었다. $K = 3(7/5)$ 인 터보 코드는 $K = 7$ 인 컨벌루션 코드보다 0.44dB정도의 성능향상을 보였다. 같은 조건에서 $K = 3$ 또는 5인 터보 코드는 부호화율이 1/3에서 컨벌루션 코드와 비슷한 성능을 나타내었다. 결론적으로 프레임 크기가 작아도 반복 횟수가 증가할수록 터보 코드의 성능향상을 기대할 수 있고 복잡도 관점에서 볼 때 음성/제어 프레임을 사용하는 무선 통신 시스템에서는 낮은 E_b/N_0 일수록 터보 코드가 우수하고 높은 E_b/N_0 에서는 컨벌루션 코드가 우수함을 알 수 있다.

향후 과제는 다양한 채널 환경에서의 복잡도를 고려한 성능 분석과 현재 계속 연구되고 있는 다른 복호 알고리즘을 적용한 성능 평가가 이루어져야 한다. 또한, 본 논문에서 복잡도 연산이 소프트웨어적으로 이루어 졌기 때문에 앞으로는 하드웨어 복잡도를 고려한 성능 평가가 적용되어야 한다.

참고 문헌

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes(1)", in *Proc. IEEE ICC*, pp. 1064-1070, May. 1993.
- [2] S. Benedetto, G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated schemes," *IEEE Trans., Theory*, vol. IT-42, no. 2, pp. 409-428, 1996.
- [3] P. Jung, M. nasshan, "Comprehensive comparison of turbo-code decoders," in *Proc. IEEE Veh. Tech. Conf.*, pp. 624-628, 1995.
- [4] S. Benedetto, G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Comm.*, vol. 44, no. 5, pp. 591-600, 1996.
- [5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287, Mar.

