

유니온 상한을 이용한 터보코드의 인터리버 설계

안 홍 영

홍익대학교 과학기술대학 전자전기컴퓨터공학부

전화: (041) 860 - 2583

Design of Turbo Code Interleaver Using Union Bound

Ahn, Hong-Young

School of Electronic, Electrical, and Computer Engineering, Hongik University

hyahn@wow.hongik.ac.kr

Abstract

Given the frame size and constituent encoders, interleaver plays a central role in a turbo code performance. Interleaver should be designed to generate a heavy weight codeword in the second encoder if the first encoder generates a low weight codeword. In this paper we present a search procedure for high performance interleaver whose performance is measured by computing union bound using the weight spectrum of all the weight 2, 3 input sequences.

I. 서론

길쌈부호를 이용한 채널코딩을 통하여 통신시스템의 비트오류율을 줄이는 연구에는 두 가지 접근방법이 있다. 전통적인 접근방법은 부호의 최소 해밍거리를 늘이는 방법들에 대한 연구이다. 두 번째 접근 방법은 낮은 해밍 무게를 가지는 부호어의 수를 줄이고 반복 부호를 사용하는 방법이다. Berrou 등[1,2]에 의해 제안되어 최근에 큰 관심을 끌고 있는 Turbo Code 는 두 번째 접근 방법에 기초해서 발견되었다.

그림 1은 전형적인 터보코드의 부호기이다. 터보코드 부호기는 인터리버들에 의해 분리된 두 개 이상의 재귀 길쌈부호기들이 패리티 비트들을 생성한 후, 정보 비트들과 체계적으로 결합시켜 부호어를 생성한다. 따라서 터보코드의 출력은 구성 길쌈부호기 수 만큼의 다중 패리티 정보를 지닐 수 있으며 요구되는 부호율에 맞추어 puncturing 할 수 있다. 인터리버를 통하여 터보코드의 부호어는 무작위성을 가지게되며 터보코드

의 성능을 결정하는 중요한 인자중의 하나이다.

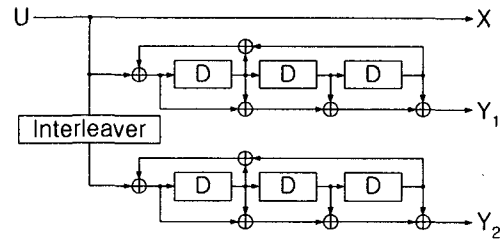


그림 1. 터보코드의 구조

복호기는 부호기에 있는 재귀 길쌈부호기와 같은 수의 디코더가 직렬로 연쇄된다. 디코더는 데이터와 대응되는 부호기 패리티를 이용하여 각 비트에 대한 Maximal a posteriori probability(MAP)를 계산하고 이로부터 외부 정보를 추출한다. 외부정보는 디인터리빙되어 다음단 디코더의 a priori 정보가 되어, 데이터 비트, 패리티 비트들과 함께 다음단의 디코더의 입력이 되어 MAP을 다시 계산한다. 마지막 디코더의 외부 정보는 첫단의 a priori 정보가 되어 복호의 새로운 반복이 시작된다. 복호의 이러한 과정이 터보엔진의 동작과 유사하여 부호의 이름이 지어졌고, 전 단에서 계산된 외부정보가 다음단의 a priori 정보가 되어 복호의 신뢰성을 높여주며, 이것이 반복되면 각 비트 결정에 대한 신뢰도를 높여주게 되고 신뢰도 값으로 연판정이나 경판정을 할 수 있게 되는 것이다.

MAP 알고리즘은 계산량의 방대함으로 비터비 알고리즘(VA)을 개선한 SOVA(Soft Output Viterbi Algorithm)가 Hagenauer등[6]에 의해 제안되었으며 터보코드 복호에도 적합하다. SOVA는 생존경로와 경쟁경로의 경로 메트릭의 차를 이용하여 생존경로가 잘못될 확률을 구하고, 생존경로와 경쟁경로의 출력비

트가 다른 노드에서는 연판정 값을 수정해 나간다.

터보코드의 성능은 생성다항식, 인터리빙 방법, 프레임 크기 등에 따라서 달라진다. 본 논문에서는 프레임 크기와 생성다항식이 주어졌을 때 좋은 성능을 낼 수 있는 인터리버를 찾는 절차를 제시한다. 인터리버의 성능을 평가하는 척도로서 무게 2와 3 입력에 대한 부호어 무게 스펙트럼을 구하고 20항까지의 유니온 상한 값을 이용하였다.

II. Turbo Code 의 Union Bound

입력열의 길이가 N인 터보코드를 가정하자. 부호율 1/2, 메모리 ν 인 $(2, 1, \nu)$ 터보코드는 길이가 $2(\nu + N)$ 인 2^N 개의 부호어로 구성되었다고 볼 수 있다. 터보코드는 선형이므로 오류분석을 위해 모두 0인 부호어가 전송되었다고 가정하자. AWGN 채널에서 최우복호기가 0 아닌 부호어로 잘못 판단했을 때 BER은 유니온 한계에 의해 다음식으로 상한된다[3].

$$P_b \leq \sum_{i=1}^{\nu} \frac{w_i}{N} Q\left(\sqrt{d_i \frac{2RE_b}{N_o}}\right) \quad (2.1)$$

여기서 w_i 는 i 번째 정보열의 무게이고 d_i 는 이에 대응하는 부호어의 무게이다.

같은 무게를 가지는 부호어를 모으기 위해 W_d 를 무게 d인 모든 부호어들에 대응되는 정보열들의 총 무게라 하고 N_d 를 무게 d인 부호어의 개수라 하면 평균 입력무게는 $\tilde{w}_d = \frac{W_d}{N_d}$ 이다. 그러면 (2.1)식은 다음과 같이 표현 할 수 있다.

$$P_b \leq \sum_{d=d_{free}}^{2(\nu+N)} \frac{N_d \tilde{w}_d}{N} Q\left(\sqrt{d \frac{2RE_b}{N_o}}\right) \quad (2.2)$$

여기서 d_{free} 는 부호의 자유거리이고 0 아닌 부호어중 무게가 가장 낮은 부호어의 무게이다.

$\frac{N_d}{N}$ 를 무게 d 인 부호어의 유효계수(effective multiplicity)라 하며 터보코드의 성능을 결정하는 중요한 인자중의 하나이다.

(2.2)식을 보면 터보코드의 성능을 결정하는 2가지 요소가 있다. 첫째는 d_{free} 를 증가시켜 (2.2)의 Q 항의 효과를 감소시키는 것이고 길쌈부호 설계에서 전통적으로 취하는 접근법이다.

둘째는 d_{free} 는 작더라도 $\frac{N_d \tilde{w}_d}{N}$ 의 값을 작게 하는 접근법이고 터보코드에서 긴 프레임 길이를 사용하고 (N 증가) 유사랜덤 인터리버를 통하여 낮은 무게의 부호어가 작게 발생하도록 ($N_d \tilde{w}_d$ 감소) 부호어를 설계하는 것이다. 따라서 최적 터보코드의 설계는 d_{free} 를 크

게 하면서 낮은 무게의 부호어들이 작게 나타나도록 구성부호기와 인터리버를 찾는 것이다.

중간크기와 큰 SNR에서 BER에 대한 유니온 상한은 d_{free} 항에 의해 거의 결정된다. 따라서 이런 SNR에서의 터보코드의 점근적 성능은 (2.2)식의 첫째항에 의해 결정되고 다음과 같이 나타낼 수 있다.

$$P_b \approx \frac{N_{free} \tilde{w}_{free}}{N} Q\left(\sqrt{d_{free} \frac{2RE_b}{N_o}}\right) \quad (2.3)$$

여기서 N_{free} 는 d_{free} 무게를 가지는 부호어의 수고 \tilde{w}_{free} 는 d_{free} 부호어를 야기하는 정보열의 평균 무게이다. (2.3)식을 터보코드의 자유거리 점근식 P_{free} 라 한다.

터보코드는 SNR이 어느 정도 증가하면 복호의 반복 회수를 증가시키더라도 BER의 개선이 낮은 SNR에서의 급격한 개선에 비해 현저히 감소하는 “오류바닥” 현상이 나타난다. (2.3)식은 터보코드 성능의 “오류바닥”을 제어하는 두 가지 방법을 제공해 준다.

첫째는 자유거리와 오류계수가 고정될 경우 인터리버의 길이를 증가시킴으로써 유효계수를 줄이고 결과적

으로 $\frac{N_{free} \tilde{w}_{free}}{N}$ 가 감소하여 자유거리 점근식의 기울기가 변하지는 않더라도 점근식을 낮춤으로써 더 큰 SNR에서 오류바닥이 나타나도록 할 수 있다. 역으로 자유거리와 오류계수가 고정되었다면 인터리버의 길이가 줄어들면 점근선은 올라가게 되고 오류바닥은 낮은 SNR에서 나타나게 된다.

둘째는 인터리버의 길이와 오류계수가 고정될 경우 부호의 자유거리를 증가시킴으로써 오류바닥을 제어하는 것이다. 이 경우 자유거리 점근식의 기울기가 변하게 되며 d_{free} 가 클수록 더 커지게 되고 작을수록 작아지게 된다. 따라서 오류바닥은 더 큰 SNR에서 일어나게 되고 d_{free} 가 작으면 낮은 SNR에서 일어나게 된다.

유니온 한계에 사용되는 N_d 와 \tilde{w}_d 를 구하기는 대단히 어려우며, 무게2 입력에 대한 최소부호어 무게(유효자유거리, $d_{free, eff}$)가 부호의 성능을 결정하는 척도로 자주 사용되고 있다. 또한 Frame 크기가 작고 낮은 SNR에서는 (2.2)식의 큰 무게 항들의 영향도 무시할 수 없다. 본 논문에서는 무게 2와 3인 입력열들에 의해 생성되는 부호어에 대한 무게 분포를 구하고 (2.2) 식의 20항까지 더하여 ($P_{b, \omega 2 \omega 3}$) 인터리버 성능을 결정하는 척도로 사용하였으며 실험결과 효과적인 성능척도임을 알 수 있었다.

III. Interleaver 설계

터보코드의 성능은 부호어의 무게분포에 의해 결정됨을 앞 절에서 살펴보았다. 인터리버는 부호어의 무게 분포를 결정하게 되므로 터보코드의 설계에 중요한 역

할을 하게 된다. 터보코드의 기술은 한 부호기에서 작은 무게의 구성부호어가 나오더라도 인터리버에 의해 재배열된 입력에 대해 다른 부호기가 큰 무게의 구성부호어가 나오도록 인터리버를 설계하는 것이다.

작은 무게의 구성부호어는 작은 무게 입력열에 대해 작은 무게의 패리티를 만드는 경우이다. 입력열 $u(D)$ 가 피드백 다항식 $g_1(D)$ 로 나누어질 때이며 이를 자체 종료열 이라한다. 따라서 인터리버는 자체 종료 입력열이 다른 자체 종료 입력열로 재배열되지 않도록 해야 한다. 낮은 무게 입력열들은 다음 특성을 지닌다.

무게 1인 입력열 : 부호기의 재귀적 특성상 이전의 정보 비트가 메모리로 부터 계속해서 부호기 입력으로 제한 되므로 무한길이 응답(IIR)을 가지게 되며 낮은 무게의 부호어는 생성되지 않는다.

무게 2인 입력열 : 피드백 다항식 $g_1(D)$ 에 의해 나누어지는 $D^i(1+D^q)$ 형태의 무게 2 자체 종료 입력열이 반드시 존재하게 된다. $g_1(D)$ 가 m 차 원시 다항식 이면 $q=2^m-1$ 이고 최대 길이열이 된다. 또한 $D^i(1+D^{mq})(n: 양정수)$ 도 자체 종료 입력열이다. 따라서 인터리버는 q 의 양수배 만큼 떨어진 위치에 재배열 되는 수는 차이가 q 로 나누어지지 않도록 배치하여 2번째 부호기의 입력열은 $g_1(D)$ 로 나누어지지 않도록 한다.

무게 3인 입력열 : 입력무게 3인 $1+D^r+D^s$ 가 $g_1(D)$ 로 나누어지면 $1+D^r+D^{s+mq} = (1+D^r+D^s) + (D^s+D^{s+mq}) = g_1(D)Q(D) + g_1(D)Q'(D)$ 이므로 $g_1(D)$ 로 나누어진다. 인터리버는 q 의 양수배 위치들끼리 재배열 되지 않도록 한다.

무게 4이상의 입력열 : 랜덤 인터리버를 사용하면 낮은 무게의 부호어를 생성할 확률을 현저히 줄일 수 있다.

터보코드 성능개선과 관련하여 다음과 같은 사실들을 관찰 할 수 있다.

1. 구성부호기에서 제한 생성기는 원시다항식을 사용하여 d_{free} 를 크게한다[4].
2. i 번째 정보 비트 d_i 에 대한 외부정보는 MAP 복호 알고리즘의 경우 95% 정도가 M단계 전방제귀와 M단계 후방제귀에 의해 결정된다. 여기서 $M=10 \times$ 구속장 이다. 따라서 두 복호기에서 외부정보를 무상관으로 계산하기 위해서는 인접한 비트들은 2M 보다 멀리 떨어져게 인터리브할 필요가 있다.
3. 한 복호기에서 일어나는 연립오류를 분산시키기 위해서는 랜덤 인터리버가 가장 좋다.
4. 부호율을 높이기 위해 천공을 할 경우 천공과 맞추어 1 패리티들이 생성되지 않도록 천공패턴을 바꾸거나 인터리빙된 1 패리티 비트들은 가능한 한 천공을 피하도록 한다. Barbulesa는 홀수 위치는 홀수 위치로 짝수 위치는 짝수 위치로 인터리빙하여 모든 위치에서 균등한 오류보호를 함으로써 더 나은 성능을 얻을 수 있음을 보였다.
5. 유사 랜덤 인터리버의 경우 양 구성부호기가 동시

에 0 상태로 종료하기는 대단히 어렵다. 마지막 m 비트를 추가하여 첫째 부호기만 최종 0 상태로 만들면 두 번째 부호기의 상태는 인터리버 크기가 클 경우 부호의 성능에 미치는 영향은 무시할수 있다.

6. 입력열의 1들이 프레임의 뒤쪽에서만 존재하게 되면 낮은 무게 부호어를만들 수 있으므로 프레임 뒤쪽의 비트는 앞쪽이나 중간으로 인터리빙 하는게 좋다.

이러한 사실들을 고려하여 다음과 같은 설계절차를 제안한다.

step 1. 시스템 요구조건에 맞는 메모리 차수 ν 와 인터리버의 크기(N , 프레임 크기)를 결정한다.

step 2. 주어진 메모리 차수에 맞는 원시다항식을 table에서 찾는다. 입력무게 2의 유효 자유거리를 최대로 하는 낮은 차수의 구성부호기는[4]에서 찾을 수 있다.

step 3. 인터리버를 0으로 초기화한다.

탐색창을 $w = \sqrt{N/2}$ 로 설정한다.

For I=1 : 인터리버의 크기(N) - L (L은 w 로설정)

- ① 1에서 N 사이의 랜덤 숫자를 하나 생성한다.
- ② 랜덤숫자가 짝수 위치에는 짝수 홀수 위치에는 홀수가 아니면 ①로 되돌아간다.
- ③ 지금까지 생성된 랜덤 숫자가 이미 있는지 조사한다. 있으면 ①로 되돌아간다.
- ④ 탐색창 w 이내에 이미 생성된 랜덤 숫자들과의 차이가 w 이내면 ①로 되돌아간다.
- ⑤ 원시 다항식 주기 $q=2^m-1$ 정수배에 생성된 랜덤 숫자와의 차이가 q 로 나누어 떨어지는지 조사한다. 나누어 떨어지면 ①로 되돌아간다.
- ⑥ ①, ②, ③, ④, ⑤ 조건을 만족하면 현재위치에 저장한다. 인터리버의 나머지 L항들은 1에서 N까지 생성되지 않은 숫자를 역으로 삽입한다.

step 4. 생성된 인터리버를 사용하여 입력 무게 2와 3의 가능한 모든 입력패턴에 대해 부호어의 무게 분포를 구한다. (2.2) 식의 유니온 상한을 20항까지 계산하여 $(P_{b, \omega 2 \omega 3})$ 초기에 설정한 값 이하이면 인터리버를 저장한다. step 3으로 되돌아 간다.

IV. 실험 및 고찰

위의 알고리즘을 사용하여 그림 1의 터보부호기에 대해 프레임 크기 256, 400에 대해 1.5 dB에서 최소거리 11, $P_{b, \omega 2 \omega 3} < 2 \times 10^{-6}$ 인터리버들을 찾았다. 부호는 패리티열을 교대로 천공하여 부호율 $\frac{1}{2}$ 이다.

<표> 1, 2, 3은 인터리버들의 부호어 무게 23까지의 무게 2, 3 입력에서 생성된 부호어 수들이다. g_1 은 제한 다항식, g_2 는 순방향 다항식이다.

<표 1> Interleaver1 부계분포 (Frame=256, $g_1=15_8$, $g_2=17_8$)

부호부계	11	12	13	14	15	16	17	18	19	20	21	22	23
W_2 입력	0	2	3	4	2	17	6	12	5	19	7	15	13
W_3 입력	0	4	10	9	29	18	40	20	66	30	121	74	165
총수	0	6	13	13	31	35	46	32	71	49	128	89	178

<표 2> Interleaver2 부계분포 (Frame=256, $g_1=15_8$, $g_2=17_8$)

부호부계	11	12	13	14	15	16	17	18	19	20	21	22	23
W_2 입력	1	1	1	3	4	17	2	12	4	26	6	15	12
W_3 입력	3	2	11	8	18	20	41	22	68	49	117	48	148
총수	4	3	12	11	22	51	43	34	72	75	123	63	160

<표 3> Interleaver3 부계분포 (Frame=400, $g_1=15_8$, $g_2=17_8$)

부호부계	11	12	13	14	15	16	17	18	19	20	21	22	23
W_2 입력	0	2	3	6	1	20	2	9	1	33	1	12	2
W_3 입력	0	0	4	4	13	5	21	10	39	28	59	25	88
총수	0	2	7	10	14	25	23	19	40	61	60	37	90

그림 3은 인터리버 1, 2의 Log-Max 알고리즘을 사용한 6회 반복복호한 모의 실험 결과이다. 1-2.5dB 까지는 10000 Frame을 3dB 는 15000 프레임을 실험하였다.

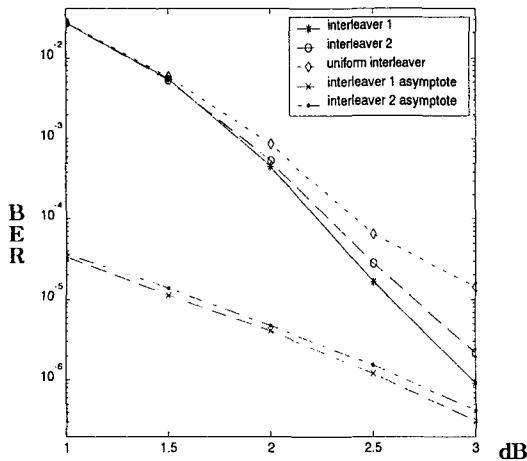


그림 6. Framesize:256, $g_1=15_8$, $g_2=17_8$, 6 iteration

다. 유니폼 인터리버는 [3]에서 제시한 가상적 모델로서 인터리버 양상불의 평균성능을 나타내는데 본 실험에서는 매 프레임마다 새로운 랜덤인터리버를 생성하여 사용하였다. 점근선은 각 인터리버의 $P_{b, \omega 2\omega 3}$ 값들이다. 인터리버들의 성능은 점근선들이 예측했던 성능과 유사하게 나왔으며 인터리버 1의 경우 BER 10^{-5} 에서 유니폼인터리버에 0.5dB 이상의 부호이득을 보였다. 그림 4는 프레임 크기 400에 대한 실험 결과이다. 1-2.5 dB 까지는 8000 프레임 3 dB는 10000 프레임을 실험한 결과이다. 1-2.5 dB 까지는 당연히 256 비트 프레임 보다 성능이 좋았으나 3.0 dB에서는 성능이 다소 떨어졌다. 실험회수가 부족이라 예측된다.

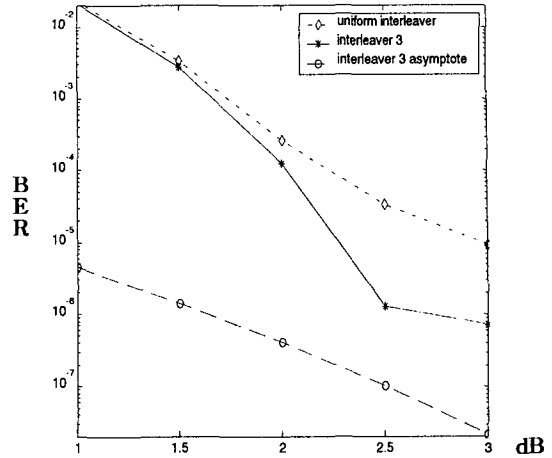


그림 4. Framesize:400, $g_1=15_8$, $g_2=17_8$, 6 iteration

V. 결론

인터리버는 터보코드의 성능을 결정하는 중요한 요소 중 하나이다. 본 논문에서는 좋은 성능을 낼 수 있는 인터리버에 국한하여 탐색하는 절차를 제시하고 인터리버 성능의 척도로서 유니온 상한 $P_{b, \omega 2\omega 3}$ 가 유용한 도구가 될 수 있음을 실험을 통하여 확인하였다.

<참고 문헌>

- [1] C. Berrou, A. Glavieux and P.Thitimajshima, "Near Shannon limit Error-Correcting Coding and Decoding: Turbo Codes", Proceedings of ICC'93, pp. 1064-1070, Geneva, Switzerland, May 1993
- [2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261-1271, Oct. 1996.
- [3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409-428, Mar. 1996.
- [4] S. Benedetto and G. Montorsi, "Design of Parallel Concatenated Convolutional Codes," *IEEE Trans. Commun.*, vol. 44, pp. 591-600, May 1996
- [5] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in *Proc., IEEE Int. Conf. on Commun.*, pp. 54-59, May 1995.
- [6] J. Hagenauer and L. Papke, "Decoding turbo codes with the soft-output Viterbi algorithm (SOVA)," in *Proc., IEEE Int. Symp. on Inform. Theory*, p. 164, June 1994.