

# MAP복호기를 이용한 Turbo code 설계

박태운, 조원경

경희대학교 전자공학과

bigcloud@csvlsi.kyunghee.ac.kr

## Design Turbo code with MAP decoder

taewoon park, wonkyung cho

Dept. of Electronic Engineering, Kyunghee Univ

bigcloud@csvlsi.kyunghee.ac.kr

### Abstract

Turbo decoder were shown to achieve performance within 0.7dB of the Shannon capacity limit. This constituted a significant gain in power efficiency over other coding techniques known at the time. In this paper, Turbo code with constraint length  $K=4$ , code rate  $1/3$ , frame size 196bits(6 tail bits), 20ms frame and 6bit MAP decoder is implemented using VHDL.

The designed Turbo code is used for voice service. Interactions of the system are used to attain large performance improvements.

### 1. 서론

1993년 Berrou에 의해 발표된 Turbo code는 BER  $10^{-5}$  과 전송율  $1/2$ 에서  $E_b/n_0$ 가 0.7dB가 된다는 것을 증명해 보였다[1]. 이는 Shannon limit에 거의 근접한 성능으로 지금까지의 오류 정정 부호 중 가장 우수한 것으로 많은 연구가 진행되고 있다.

Turbo code는 사용하는 복호기의 종류에 따라 SOVA(Soft-Output Viterbi Algorithm)와 MAP(Masimum A Posteriori) 복호기로 나누어진다. SOVA는 Viterbi 복호기를 확장한 것으로 구현은 간단하나 MAP 복호기에 비해 BER 성능이 떨어진다[2]. 이러한 이유로 하드웨어 구현은 복잡하지만 BER 성능이 좋은 MAP 알고리즘이 사용되고 있는 추세이다[3]. 본 논문에서는 MAP 알고리즘의 하드웨어 구현을 위해서 곱셈을 특정한(E)함수로 바꾸는 log-MAP 알고리즘을 사용한다. 이와 같은 과정을 통해서 실제 하드웨어로 구현이 가능하다[4][5].

IMT2000의 음성서비스는 8kbps, 32kbps와 같은 저속

전송율과 지연이 40ms이내가 되어야 하고 BER $<10^{-4}$ 을 만족해야 한다[6][7]. 본 논문은 프레임 크기가 192bits(6 tail bits), 20ms로 전송율이 9.6kbps인 규격을 제시한다. 부호기는 구속장  $K=4$ , 생성다항식이 (13,15)이며  $1/3$ -rate인 구조이다. 인터리버 크기는 tail bits가 6bits이기에 186이다.

인터리버는 2차원 인터리버를 이용했다. 이때 2차원 인터리버는 프레임의 크기가 제한되는 단점이 있기에 본 논문에서는 IAG(interleaver address generator)를 하나의 블록으로 만들어 이것과 제어신호의 조작으로 인터리버 크기를 바꿀 수 있도록 설계했다.

본 논문은 MAP복호기를 이용해 IMT2000의 음성서비스에 이용될 수 있는 Turbo code를 설계함을 목적으로 한다.

### 2. MAP 복호기 알고리즘

정보 비트를  $d_k$ , 패리티 비트를  $Y_k$ 라 하고 AWGN 채널을 통해 변조되어 수신된 시퀀스를  $R_1^N = (R_1, \dots, R_N)$ 라 하자. 여기서  $R_k = (x_k, y_k)$ 이다. 심볼  $x_k, y_k$ 는

$$x_k = (2d_k - 1) + p_k$$

$$y_k = (2Y_k - 1) + q_k$$

$p_k, q_k$ 는 평균이 zero이고 동일한 분산  $\sigma^2$ 를 갖는 독립된 잡음이다.

각 복호 비트  $d_k$ 에 대한 Log Likelihood Ratio  $L(d_k)$

$$L(d_k) = \log \frac{P_r(d_k = 0|R_1^N)}{P_r(d_k = 1|R_1^N)} \quad (1)$$

여기서  $P_r(d_k = i | R_k^N)$ 는 데이터 비트  $d_k$ 의 APP(a posteriori probability)가 된다. 복호기는 LLR함수  $L(d_k)$ 를 판별하여 정보 비트 값을 결정한다.

$$\bar{d}_k = \begin{pmatrix} 1 ; L(d_k) \geq 0 \\ 0 ; L(d_k) < 0 \end{pmatrix}$$

식(1)을 Bayes' Rule의 결합확률식을 이용해  $\alpha_k^i(m)$ 와  $\beta_k^i(m)$ 을 정의하면

$$\alpha_k^i(m) = P_r(d_k=i, S_k=m, R_k^k)$$

$$\beta_k^i(m) = P_r(R_{k+1}^N | d_k=i, S_k=m)$$

LLR은 다음과 같다.

$$L(d_k) = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \quad (2)$$

식(2)를 Modified Bahl Decoding Algorithm[4]에 의해 바꾸면 식(3)(4)(5)와 같다.

순방향 메트릭  $\alpha$

$$\alpha_k^i(m) = \delta_i(R_k, m) \sum_{j=0}^1 \alpha_{k-1}^j(S_b^j(m)) \quad (3)$$

역방향 메트릭  $\beta$

$$\beta_k^i(m) = \sum_{j=0}^1 \beta_{k-1}^j(S_b^j(m)) \delta_j(R_{k+1}, S_b^j(m)) \quad (4)$$

가지 메트릭  $\delta$

$$\delta_i(R_k, m) = \exp\left(-\frac{2}{\sigma^2}(x_k i + y_k Y_k^i(m))\right) \quad (5)$$

위  $\alpha, \beta, \delta$ 는 하드웨어로 구현하기에는 적합하지 않기에 식(2)에 로그를 취하는 log-MAP 알고리즘을 구한다. 이 과정에서 E 함수를 정의해야 한다[4].

E 함수는

$$x \text{ E } y = \min(x, y) - \ln(1 + e^{-|x-y|})$$

위 식을 통해  $x \text{ E } y$ 는  $\min(x, y)$ 와  $f(|x-y|)$ 로 나타낼 수 있다.

$$f(z) = \log_e(1 + e^{-z}) = c \ln(1 + e^{-z/c})$$

여기서  $c = \sigma^2 A / 2$ 가 되며  $f(z)$ 는 lookup table로 구성된다.

E함수와 log-MAP 알고리즘을 이용해서 식(2)를 바꾸면 LLR은 식(6)(7)(8)(9)와 같다.

$$L(d_k) = E_{m=0}^{2^r-1} [A_k^1(m) + B_k^1(m)] - E_{m=0}^{2^r-1} [A_k^0(m) + B_k^0(m)] \quad (6)$$

순방향 메트릭 A

$$A_k^i(m) = D_i(R_k, m) + E_{j=0}^1 [A_{k-1}^j(S_b^j(m))] \quad (7)$$

역방향 메트릭 B

$$B_k^i(m) = E_{j=0}^1 [B_{k+1}^j(S_b^j(m))] + D_j(R_{k+1}, S_b^j(m)) \quad (8)$$

가지 메트릭 D

$$D_i(R_k, m) = \frac{2}{\sigma^2} (x_k i + y_k Y_k^i(m)) \quad (9)$$

식(6)을 계산하는 과정은 먼저 각 입력신호에 대해 가지 메트릭을 구해 MN의 배열에 저장한다. 여기서 M은 상태의 개수이며 N은 데이터의 크기이다. 역방향 메트릭을 구해서 MN에 저장한다. 여기서 대칭성을 이용해 역방향 메트릭의 저장 공간을 MN/2로 줄일 수 있다. 순방향 메트릭의 값을 구해 이 값을 해당 역방향 메트릭의 값과 함께 LLR연산부에 입력하여 결과 값을 구한다.

### 3. Turbo 부호기 구조

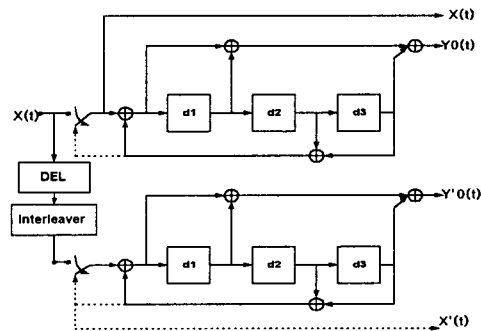


그림1 Turbo 부호기  
Fig1. Turbo encoder

Turbo 부호기는 2개의 RSC(recursive systematic convolutional)부호기를 이용하며 생성다항식은 구성장  $K=4$ 에서 가장 좋은 code로 알려진 (13,15)를 이용했다. DEL의 MAP 복호기의 지연을 고려해주는 부분으로  $3N + \alpha$ 의 시간지연이 필요하다.(N은 데이터의 크기,  $\alpha$ 는 연산과정에 필요한 단위 지연이다) 부호율은 1/3이다. 점선 부분은 tail bits를 의미하는 것으로 2개의 RSC에 각각 3bits이므로 tail bits는 6bits가 쓰였다.

### 4. Turbo 복호기 구조

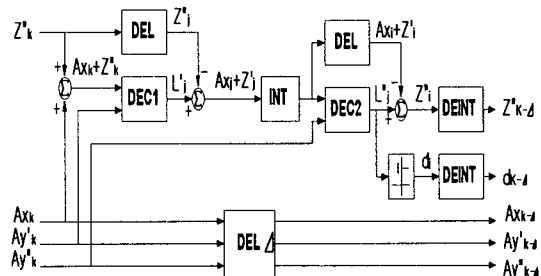


그림2 Turbo 복호기  
Fig2. Turbo decoder

그림2와 같이 Turbo 복호기는 인터리버 1개, 디인터리버 2개, MAP 복호기 2개와 지연 부분들로 구성되어 있다.

DEC1은 입력으로  $Ax_k + z'_k$ 와  $Ay'_k$ 를 이용한 첫 번째 MAP 복호기이다. 여기서  $z'_k$ 는 사전확률이며 DEC2(두 번째 MAP 복호기)에서 출력된 외부 정보(extrinsic information)이며 첫 번째 반복 복호시에는 zero가 된다. DEL은 DEC1의 지연으로서  $3N + \alpha$ 가 된다. 인터리버와 디인터리버의 지연은  $N-6$ 이다.(부호기의 구조장이 4이고 tail bits가 6이기 때문이다) 따라서 복호기 전체 연산에 필요한 지연은  $\Delta = 2(3N + \alpha) + 2(N-6) + \beta = 8N - 12 + 2\alpha + \beta$ 가 된다. ( $\alpha$ 는 DEC연산에 필요한 단위 지연,  $\beta$ 는 블록사이의 덧셈연산에 필요한 단위 지연이다)

반복 복호를 하는 동안에 필요한 지연은  $\Delta * I$ 가 된다.(표4) 여기서  $I$ 는 반복복호의 횟수이다. 따라서 20ms이내에서 동작될 수 있는 범위 내에서  $I$ 를 증가시킬 수 있다. 그러나 메모리 크기에 대한 고려도 해야 한다.

#### 4.1 인터리버 구조

2차원 인터리버는 블록, 대각, 랜덤인터리버등이 있다. 인터리버의 크기가 작을 때는 랜덤인터리버 보다 구조화된 인터리버를 사용하는 것이 유리하다[7]. 본 논문에서는 여러 형태의 인터리버를 비교하여 가장 적당한 구조화된 인터리버를 찾아서 사용했다. 하지만 실험결과 인터리버의 크기가 186처럼 작은 경우는 여러 구조화된 인터리버가  $m*(m+1)$ 블록인터리버와 거의 비슷한 성능을 보였다. 표1은 186 인터리버의 경우에서 여러 구조화된 인터리버 중 6\*31 블록인터리버 후 각 열의 주소를 bit reverse해서 보낸 경우와  $m*(m+1)$ 블록인터리버의 성능을 비교해 보았다.

또한 각 인터리버 크기마다 가장 최적화된 인터리버를 찾아서 그 값을 메모리에 저장하는 방식을 취했다. 이런 방법을 통해 인터리버의 크기를 확장하는 경우에도 IAG(interleaver address generator)와 간단한 제어신호 조작만으로 적용 가능하도록 만들었다.

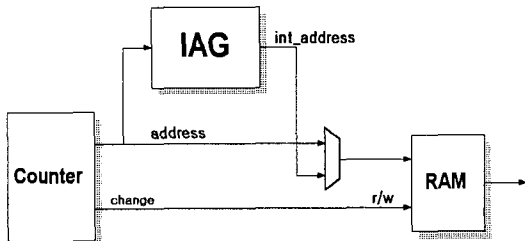


그림3 인터리버의 구조  
Fig3. interleaver structure

#### 4.2 MAP 복호기 구조

MAP 복호기는 크게 BMC(branch metric calculator), RSMC(reverse state metric calculator), FSMC(forward state metric calculator), LLRC(log likelihood ratio calculator)로 구성된다. 여기에 E 함수 테이블과 상태 테이블이 추가되며 BMC의 결과를 저장하는 공간과 RSMC의 결과를 저장하는 부분이 필요하다. 본 논문에서 MAP 복호기 입력은 6bit로 양자화 시켰으며 구현하고자 하는 프레임의 크기가 192이고 20ms에서 동작하기에 복호기는 10Mhz이상에서 동작하도록 설계를 하였다. 인터리버의 크기가 확장되어도 음성서비스를 위한 복호기에는 10Mhz이면 서비스를 할 수 있다.

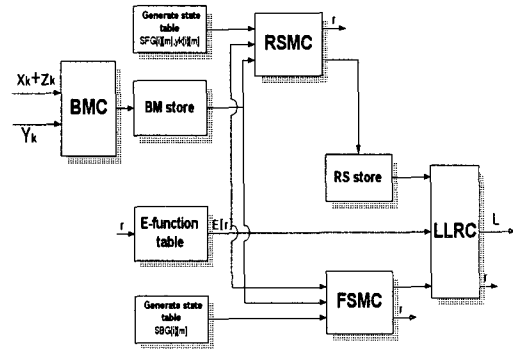


그림4 MAP 복호기  
Fig4. MAP decoder

BM store는 BMC의 출력결과를 저장하는 곳으로 BM(branch metric)을 8bits로 처리했기에  $M*N*8$ bits가 필요하다. 프레임에서  $N=192$ 로 선택했고  $M=8$  상태이므로 여기서는  $8*192*8=12288$ bits의 저장 공간이 필요하다. RS store는 RSMC의 결과를 저장하는 곳으로  $M*N*8/2$ 의 저장 공간이 필요하기에 6144bits의 저장공간이 필요하다. E-function table과 Generator state table은 테이블 형태로 메모리에 저장하는 방법으로 구현을 했으며 연산을 간단히 하기 위해 사용된다.

#### 5. 실험 및 고찰

표1에서 보듯이  $N=192$ 로 인터리버의 크기가 작은 경우 구조화된 인터리버가  $m*(m+1)$ 블록인터리버에 비해 현저한 BER 성능향상을 보이지 않는다는 것을 보여준다.

표2에서는  $N=192$ 에서 반복횟수에 따른 성능 평가를 보여준다. 반복횟수에 따라 성능은 향상되지만 향상되는 폭은 점점 줄어들며 또한 표3은  $N=192$ 와  $N=516$ 의 인터리버 크기에 따른 성능비교를 나타낸 것으로 인터리버 크기가 커질수록 BER 성능은 좋아진다.

표1. 인터리버의 종류에 따른 성능 비교

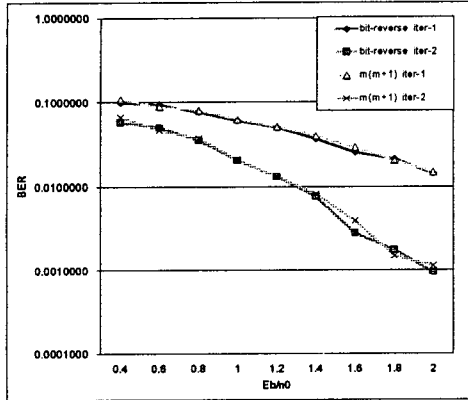


표2. 반복횟수에 따른 성능 비교

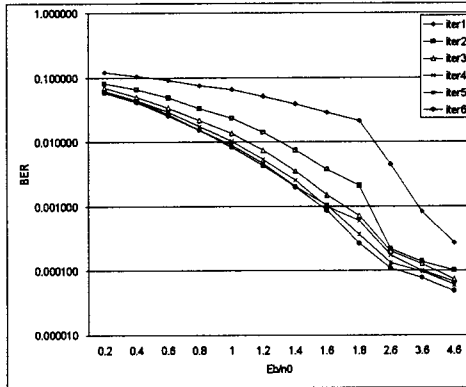


표3. 인터리버 크기에 따른 성능 비교

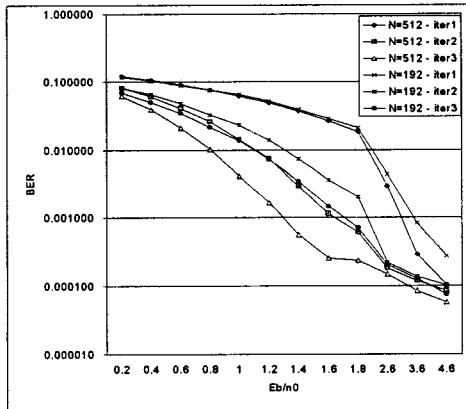


표4. 지연시간 비교

- N : 데이터의 크기, - I : 반복복호 수
- $\alpha$  : MAP복호기 1개 연산 단위지연
- $\beta$  : Turbo 부호기 블록사이의 덧셈 단위지연

	총지연
MAP복호기 1개	$T = 3N + \alpha$
인터리버 1개	$T = N - 6$
Turbo code 1번	$\Delta = 2T + 2I + \beta$
I 번 반복복호	$\Delta * I$

## 6. 결론

본 논문에서 설계한 Turbo code는 음성서비스를 위해 프레임을 192bits, 20ms, 전송율 9.6kbps로 구성했고 구조상 K=4, 1/3-rate, (13,15)인 부호기를 이용했으며 인터리버 크기는 확장이 가능하고 반복횟수도 1~6까지 반복 가능한 구조로 설계를 했다. 이와 같은 구조를 통해 인터리버 크기와 반복횟수를 조정함으로써 음성서비스에서 원하는 BER( $10^{-4}$ )를 만족하는 Turbo code를 설계했다.

## 참고문헌

- [1] C.Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes," ICC'93, Geneva, Switzerland, pp. 1064-1070, May 1993.
- [2] David Kwan and Samir Kallel. "A Rate-k/n Heuristic Soft-Output Viterbi Algorithm(SOVA) that is Postdetector-Compatible
- [3] Bahl, L., Cocke, J., Jelinek, F., and Raviv, J., "Optimal decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inform. Theory, vol. IT-20, pp. 284-287, Mar. 1974
- [4] S. S. Pietrobon and A. S. Barbulesscu, "A simplification of the modified Bahl decoding algorithm for systematic convolutional codes," Int. Symp. Inform. Theory & its Applic., Sydney, Australia, pp.1073-1077, Nov. 1994.
- [5] Steven S. Pietrobon "Implementation and Performance of a Serial MAP Decoder for use in an Iterative Turbo Decoder" 1995 IEEE International Symposium on Information Theory Whistler, B.C, Canada, 17-22 September 1995
- [6] FPLMTS/IMT-2000, Report of the Tenth Meeting of ITU-R Task Group 8/1, Mainz, April 1996.
- [7] 김 용희 4인 "FPLMTS에서 음성 전송을 위한 Turbo code 개발" 포항공과대학교 전자전기공학과, 삼성전자 정보통신연구소. 1998