

PR4 channel에서의 High Rate Turbo Code의 설계

변 남 균, 김 종 태
성균관 대학교 전기전자 및 컴퓨터 공학부
전화 : 031-290-7173 / 핸드폰 : 011-895-7229

Design of High Rate Turbo Code for PR4 Channel

NamKyun Beon, Jong Tae Kim
School of Electrical and Computer Engineering, SungKyunKwan University
E-mail : gaius@ece.skku.ac.kr

Abstract

Turbo code shows the great performance near Shannon limit on AWGN channel. Mainly, turbo code has been studied and designed for wireless digital communications. There are recent studies that applies turbo decoder on magnetic recording. Because of the limited capacity of magnetic storages, high rate turbo code is used for magnetic storages. This paper presents some issues on implementing high rate turbo code and structures for designing turbo decoder.

I. 서론

Turbo code는 Berrou등[1]에 의 1993년에 처음 소개 되었다. Turbo code는 Shannon limit에 접근하는 우수한 성능을 보이며 이러한 turbo code의 우수한 성능은 반복 복호, 연입력-연출력, 병렬연접, RSC(Recursive Systematic Convolutional) code 그리고 interleaver에 기인한다. Turbo code는 이미 디지털 통신분야에서 많이 응용되고 있으며 최근에는 magnetic recording 분야, 특히 hard disk driver 에 응용하고자하는 연구들이 이루어

지고 있다.

본 논문에서는 주로 알고리즘 레벨에서 이루어져 왔던 turbo code의 시뮬레이션 결과를 토대로 magnetic recording 분야에 사용되어지는 turbo decoder를 설계 및 검증하고자 한다.

II. Turbo decoding algorithm

2.1 MAP algorithm

정보 비트를 $u=(u_1, u_2, \dots, u_N)$ 이라고 하면 systematic encoder의 출력은 정보 비트와 같은 $x_s=(x^s_1, x^s_2, \dots, x^s_N)$ 와 parity bit sequence $x_p=(x^p_1, x^p_2, \dots, x^p_N)$ 로 표현된다. 인코딩된 비트들이 NRZ 형태로 전송되어지고 noise spectrum density가 N_0 인 AWGN(Additive White Gaussian Noise) channel를 통과한 후에는 $y=(y_s, y_p)$, $y_s=y(y^s_1, y^s_2, \dots, y^s_N)$, $y_p=(y^p_1, y^p_2, \dots, y^p_N)$ 로 표현된다.

MAP decoder에서는 log-likelihood ratio(LLR)

$$L_R(u_k) = \log \frac{P(u_k = +1/y)}{P(u_k = -1/y)} \quad (1)$$

의 부호에 따라 $u_k=+1$ 혹은 $u_k=-1$ 을 결정하게 된다. S_k 를 시간 k에서의 encoder state라고 하면 S_k 는

2^M-1 (M : encoder에서의 메모리 수)의 state를 가지게 된다. LLR를 다시 쓰면

$$L_R(u_k) = \log \frac{\sum_{S_k} \sum_{S_{k-1}} \gamma_1(y_k, S_{k-1}, S_k) \cdot \alpha_{k-1}(S_{k-1}) \cdot \beta_k(S_k)}{\sum_{S_k} \sum_{S_{k-1}} \gamma_0(y_k, S_{k-1}, S_k) \cdot \alpha_{k-1}(S_{k-1}) \cdot \beta_k(S_k)} \quad (2)$$

이다. 여기서 α 는 forward state metric, β 는 backward state metric, γ 는 branch metric 이며 다음과 같은 식으로 나타낼 수 있다[1].

$$\alpha_k(S_k) = \sum_{S_{k-1}} \gamma_i(y_k, S_{k-1}, S_k) \cdot \alpha_{k-1}(S_{k-1}) \quad (3)$$

$$\beta_k(S_k) = \sum_{S_{k+1}} \gamma_i(y_{k+1}, S_k, S_{k+1}) \cdot \beta_{k+1}(S_{k+1}) \quad (4)$$

$$\gamma_i(y_k^i, y_k^p, S_{k-1}, S_k) = q(u_k = i/S_{k-1}, S_{k-1}) \cdot p(y_k^i/u_k = i) \cdot p(y_k^p/u_k = i, S_k, S_{k-1}) \cdot \Pr(S_k/S_{k-1}) \quad (5)$$

2.2 Log-MAP algorithm

식 (3),(4),(5)에서 곱셈 연산을 피하기 위해서 각 각의 식에 log를 취하게 되면 곱셈 연산을 덧셈연산으로 바꾸어줄 수가 있다[2][3].

$$\bar{\alpha}_k(S_k) = \log(\alpha_k(S_k)), \quad \bar{\beta}_k(S_k) = \log(\beta_k(S_k)), \quad \bar{\gamma}_k(S_k) = \log(\gamma_k(S_k)) \quad (6)$$

$$\log(e^a - e^b) = \max(a, b) + \log(1 + e^{-|b-a|}) = \max(a, b) + f_c(|b-a|) = \max(a, b) + f(a, b) \quad (7)$$

식 (7)에서의 f 함수는 작은 ROM table로 구현이 가능하다[3].

식 (7)를 이용하여 식 (6)을 다음과 같이 다시 쓸 수 있다[3][4].

$$\bar{\alpha}_k(S_k) = \max(\bar{\gamma}_i(y_k, S_{k-1}, S_k) + \bar{\alpha}_{i, k-1}(S_{k-1})) + f(\bar{\gamma}_i(y_k, S_{k-1}, S_k) + \bar{\alpha}_{i, k-1}(S_{k-1})) \quad (8)$$

$$\bar{\beta}_k(S_k) = \max(\bar{\gamma}_i(y_{k+1}, S_k, S_{k+1}) + \bar{\beta}_{i, k+1}(S_{k+1})) + f(\bar{\gamma}_i(y_{k+1}, S_k, S_{k+1}) + \bar{\beta}_{i, k+1}(S_{k+1})) \quad (9)$$

Branch metric 부분은 AWGN channel에서 다음과 같이 표현된다[3].

$$\bar{\gamma}_k = (y_k, S_{k-1}, S_k) = \frac{1}{2} (L_{in}^e(u_k)u_k + L_c y_k^i u_k + L_c y_k^p x_k^p) \quad (10)$$

$L_{in}^e(u_k)$ a priori information으로 다른 디코더로부터 받은 입력 값이고 L_c 는 channel reliability 값이다.

$$L_c = \frac{4E_c}{N_0} \quad (E_c : \text{energy per coded bit}) \quad (11)$$

$$L_R(u_k) = \max(\bar{\gamma}_1(y_k, S_{k-1}, S_k) + \bar{\alpha}_{i, k-1}(S_{k-1}) + \bar{\beta}_k(S_k)) + f(\bar{\gamma}_1(y_k, S_{k-1}, S_k) + \bar{\alpha}_{i, k-1}(S_{k-1}) + \bar{\beta}_k(S_k)) - \max(\bar{\gamma}_0(y_k, S_{k-1}, S_k) + \bar{\alpha}_{i, k-1}(S_{k-1}) + \bar{\beta}_k(S_k)) - f(\bar{\gamma}_0(y_k, S_{k-1}, S_k) + \bar{\alpha}_{i, k-1}(S_{k-1}) + \bar{\beta}_k(S_k)) \quad (12)$$

$$L_R(u_k) = Lcy_k^i + L_{in}^e(u_k) + L_{out}^e(u_k) \quad (13)$$

$L_{out}^e(u_k)$ 은 다음 decoder의 extrinsic information 이고 turbo decoder의 iteration 수행이 끝난 후에 $L_{out}^e(u_k)$ 의 부호로 -1 혹은 1을 판정한다.

III. Components

3.1 RSC encoder

Rrate 1/2 RSC encoder 2개를 사용하며 두 개의 encoder는 동일한 구조를 하고 있다. 사용되어진 encoder는 $(g_1, g_2) = (31, 33)_8$ (그림1)이며 MUX에서 puncturing 함으로써 high rate code를 얻을 수 있으며 본 논문에서는 rate-8/9 로 puncturing 한다. turbo encoder의 경우에는 다른 일반적인 convolutional encoder와는 다른 격자 종결(Trellis Termination)이 필요하다. 격자 종결 후에는 8비트의 tail 비트와 tail 비트에 대한 parity 비트가 추가된다.

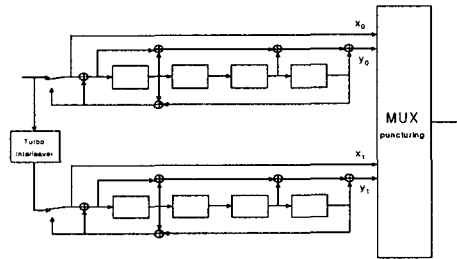


그림 1 turbo encoder

3.2 Interleaver

Interleaver는 burst error 발생 시 error들을 분산시키며 decoder의 성능에도 영향을 미친다.

Interleaver의 크기 N 은 일반적으로 프레임 크기와 같

PR4 channel에서의 High Rate Turbo Code의 설계

다. Hard disk driver의 경우 디스크 표면에 읽고 쓸 때에 한 sector씩 수행하므로 프레임 크기는 한 섹터 크기(512byte)로 하여 $N=512*8=4096$ 이다.

Interleaver는 block interleaver, pseudo random interleaver, random interleaver 등이 있으며 본 논문에서는 일반적으로 프레임의 크기가 클 때 우수한 성능을 보이는 random interleaver를 이용하여 설계하였다.

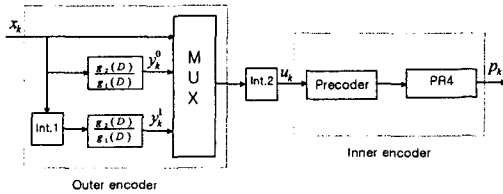


그림 2 Turbo encoder 와 Precoder

3.3 Partial Response Channel

Magnetic channel 에서의 PR(Partial Response) channel은 다음과 같이 다항식의 형태로 표현되어진다.

$$F(D)=(1-D)(1+D)^n \quad n=0,1,2,\dots \quad (14)$$

$n=1$ 일 경우를 class-4 partial response(PR4) channel 이라고 하고 $n=2$ 인 경우를 extended class-4 partial response (EPR4) channel 이라고 부른다. n 값이 커질 수록 ISI(intersymbol interference)가 커지고 또한 recording density가 커짐을 의미한다[4]. 본 논문에서는 PR4 channel에서의 turbo decoder를 설계하였다.

3.4 Precoder

PR channel 에서의 turbo decoding은 precoding함으로써 성능향상을 얻을 수 있다. PR4에서의 precoding은 $1/(1 \oplus D^2)$ 이 사용되어진다[5]. Precoder와 PR4 channel은 inner encoder로서 동작하며 inner decoder에서 decoding되며 inner decoder는 SISO(Soft-Input Soft-Output)이며 log-MAP 알고리즘을 이용한다[6].

3.5 AWGN channel

본 논문에서는 digital magnetic recording channel에서 가장 간단한 모델인 PR4 channel에 AWGN이 더해지는 모델을 사용하였다.

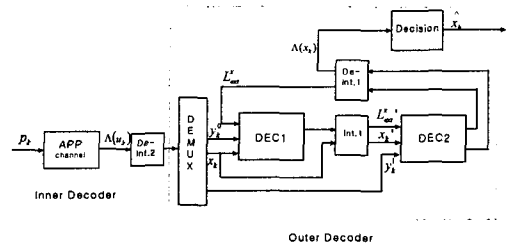


그림 3 Turbo decoder

IV. Turbo Decoder Design

4.1 Simulation 결과

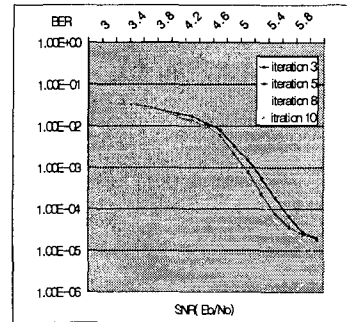


그림 4 iteration 횟수에 따른 8/9 turbo code with precoder의 성능

Rate 8/9 turbo code의 성능은 그림4 에서 보이는 바와 같다. 설계하고자 하는 터보코드의 성능은 $BER=1.0e-5$ 이다. 목표한 성능은 10번 반복 decoding 시는 약 5.8 db에서 8번 반복 decoding 시에는 6.1db에서 BER 이 $1.0e-5$ 이 된다. 본 논문에서는 8번 반복 decoding를 하며 uncoded 시스템에 비해 약 4db 정도의 gain을 얻을 수 있다.

4.2 Quantization

대부분의 논문들의 시뮬레이션 결과들은 부동소수점 연산을 한 결과들이다. 하지만 실제 turbo decoder의 구현 시에는 고정 소수점 연산을 수행하게 된다. 따라서 입력 값의 quantization level에 따라서 내부 데이터 비트의 크기가 결정되어진다. MAP algorithm의 경우에는 branch metric과 forward state metric 연산결과가 backward state metric과 LLR 계산에 이용되어지기

위해서 각 state에 대한 모든 값들을 메모리에 저장해야 하므로 많은 메모리를 필요로 하게 된다. 따라서 내부 비트 수와 quantization level를 decoder의 성능 저하를 줄이면서 요구되어지는 메모리의 양을 적절하게 조절하는 것이 필요하다.

본 논문에서는 입력 quantization level를 [5 : 2] ([전체 비트 수 : 소수점 자릿수]), outer decoder의 입력은 [7 : 3], branch metric 은 [11 : 3], forward 와 backward state metric, LLR 은 [12 : 3], extrinsic information 은 [8 : 3]으로 하여 설계하였다.

4.3 Branch, forward state, backward state metric 연산

Branch metric 부분은 식(10)을 가지고 계산한다. 식(10)은 식(15),(16) 과 같이 줄일 수 있으며 모든 state에 대한 연산이 필요 없고 단지 두 개의 값만 연산해 주면 된다. Extrinsic information 과 L_c 값이 2로 나누어진 값이 입력으로 들어옴으로써 나누기 연산을 없앨 수 있다.

$$\overline{\gamma}_{11} = (y_k, S_{k-1}, S_k) = (L_{in}^e(u_k) + L_{2c}y_k^i + L_{2c}y_k^p) \quad (15)$$

$$\overline{\gamma}_{10} = (y_k, S_{k-1}, S_k) = (L_{in}^e(u_k) + L_{2c}y_k^i - L_{2c}y_k^p) \quad (16)$$

$$L_{out}^e(u_k) = \frac{1}{2}(L_R(u_k) - L_{in}^e(u_k)) - L_{2c}y_k^i \quad (17)$$

$(L_{2c} = L_c / 2)$

그림5는 forward state metric 연산부의 block diagram이다. backward state metric 연산부도 기본적으로 같은 구조로 되어있으며 다만 state metric memory가 필요 없다.

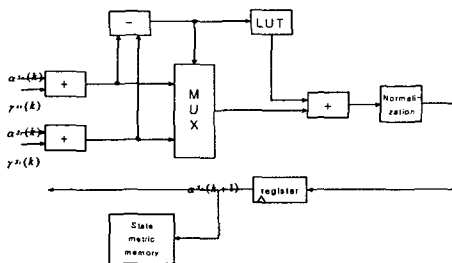


그림5 forward state metric 연산부 block diagram

V. 결론

본 논문에서는 log-MAP 알고리즘을 이용하여 magnetic recording, 특히 hard disk driver에 사용되는 turbo decoder를 설계하였다. 설계 시 C 언어를 이용하여 알고리즘을 검증하였고 VHDL을 이용하여 RTL 레벨에서 설계 및 functional simulation을 하여 이상 없이 동작함을 확인하였고, SYNOPSIS 사의 Design Compiler를 이용하여 논리 합성을 하였다.

참고문헌(또는 Reference)

- [1] C. Berrou, A. GalVieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : turbo-codes," in *proc. IEEE int. Conf. on Communications*(Geneva, Switzerland, May 1993), pp.1064-1070.
- [2] P. Robertson, E. Vilebrun, P. Hoeher,"A comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," in *Proc. ICC'95*, Seattle, WA, 1995, pp. 1009-1013.
- [3] Zhongfeng Wang; Suzuki, H.; Parhi, K.K., "VLSI Implementation Issues of Turbo Decoder Design for Wireless Applications," *Signal Processing Systems, 1999. SiPS 99. 1999 IEEE Workshop on*, 1999 Page(s): 503 -512
- [4] Souvignier, T.V., Oberg, M.; Siegel, P.H.; Swanson, R.E.; Wolf, J.K., "Turbo Decoding for Partial Response Channels," *Communications, IEEE Transactions on*, Volume: 48 Issue: 8, Aug. 2000 Page(s): 1297 -1308
- [5] Souvignier, T.; Friedmann, A.; Oberg, M.; Siegel, P.H.; Swanson, R.E.; Wolf, J.K., "Turbo Decoding for PR4 : Parallel Versus Serial Concatenation," in *Proc. IEEE Int. Conf. Communications*, Vancouver, BC, Canada, June 1999, pp. 1638-1642
- [6] McPheters, L.L., McLaughlin, S.W., Hirsch, E.C., "Turbo codes for PR4 and EPR4 magnetic recording," *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, Volume: 2, 1998 Page(s): 1778 -1782 vol.2