

Log-MAP을 사용한 3GPP용 터보 복호기의 설계

강형구^{*} · 전홍우^{*}

^{*}국립금오공과대학교

A Design of Turbo Decoder for 3GPP using Log-MAP Algorithm

Heyng-goo Kang^{*} · Heung-woo Jeon^{**}

^{*}Kumoh National Institute of Technology

E-mail : safebest@kumoh.ac.kr

요 약

MAP 알고리즘은 터보코드의 최적 복호알고리즘으로 알려져 있지만, 연산의 복잡도가 매우 크다. 이러한 MAP 방식의 단점을 개선하기 위하여 일반적으로 log-MAP 알고리즘을 이용하게 된다.

본 논문에서는 log-MAP 복호기에서 비교적 연산량이 크고 동작속도에 큰 영향을 미치는 상태 메트릭 계산 블록의 연산 속도를 향상시킨 개선된 구조를 제안하고, LUT(Lookup Table)을 이용하지 않고 간단한 연산에 의해 보상하는 linear 보상을 제안하여 동작속도를 개선하였다.

ABSTRACT

MAP algorithm is known for optimal decoding algorithm of Turbo codes, but it has very large computational complexity and delay. Generally log-MAP algorithm is used in order to overcome the defect.

In this paper we propose modified scheme of the state metric calculation block which can improve the computation speed in log-MAP decoder and simple linear offset unit without using LUT. The simulation results show that the operation speed of the proposed scheme is improved as compared with that of the past scheme.

키워드

터보 복호기, log-MAP Algorithm, ACS, 보상방법,

I. 서 론

1993년 Berrou 등에 의해 제시된 터보코드는 AWGN환경에서 비교적 간단한 구조로 Shannon의 한계에 근접하는 매우 우수한 오류정정 성능을 제공하므로 지난 10여 년간 많은 관심의 대상이 되어왔다.[1]

터보 코드는 반복횟수가 증가할수록 우수한 성능을 보인다. 그러나 그에 따르는 복호지연이 발생한다.

터보 코드의 성능에 크게 영향을 미치는 것이 인터리버이다. 본 논문에서는 3GPP표준으로 제안된 Mother 인터리버를 사용한다. 복호 알고리즘으로는 Log-MAP 알고리즘을 사용하였다.

본 논문에서는 Log-MAP 복호기에서 가장 연산빈도가 많은 ACS구조를 변형하여 빠른 연산을 수행할 수 있는 구조를 제안한다.

II. 터보 부호기

그림1은 3GPP 표준의 터보 부호기의 구조이다. 인터리버를 사이에 두고 두 개의 RSC 부호기가 병렬로 연결한 형태를 갖고, 데이터 전송률에 따라 천공(puncture)하여 채널로 전송한다.

각각의 RSC 부호기는 $g_1/g_2 = (13)_8/(15)_8$ 인 생성다항식에 의해 부호화된다.

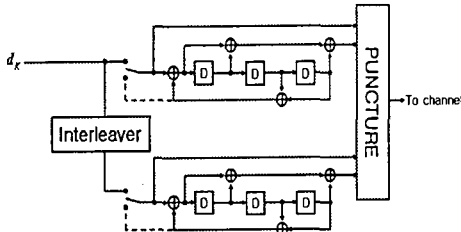


그림1 3GPP표준의 부호기 구조

3GPP에서는 부호율 1/3에 데이터 전송률을 32Kbps 이상으로 정하고 있다. 그에 따라 그림2와 같이 상태 천이에 대한 다이어그램으로 나타난다. 각 시간마다 전체 상태수는 8상태이다.

두 개의 RSC 부호기 사이에 입력 데이터의 순서 열을 재배열하는 인터리버는 해밍거리를 가장 크게 하는 Mother 인터리버를 사용한다.

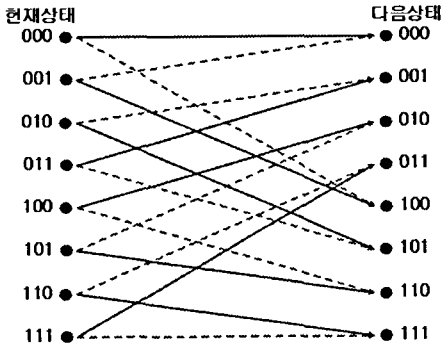


그림2 상태 천이 다이어그램

부호화된 부호어는 천공기를 거쳐 BPSK로 변조하여 AWGN 채널로 보내지게 된다. 복호기의 수신부에서는 식(1)과 같은 변조된 신호를 받아서 복호하게 된다.

$$R_1^N = (R_1, \dots, R_k, \dots, R_N) \quad (1)$$

여기서 $R_k = (x_k, y_k)$ 는 시간 k에 수신된 심벌이고 x_k, y_k 는 식(2), 식(3)과 같이 정의된다.

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

$$y_k = (2Y_k - 1) + q_k \quad (3)$$

여기서 p_k 와 q_k 는 분산이 σ^2 인 서로 독립적으로 정규화 된 Gaussian Noise이다.

III. Log-MAP 복호기

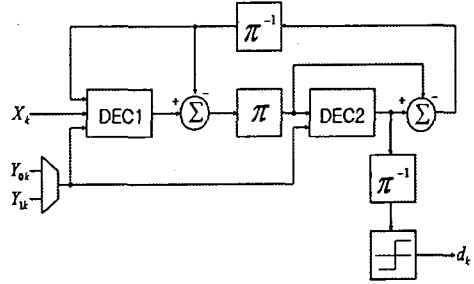


그림3 복호기의 구조

MAP 알고리즘은 각 정보에 대한 LLR(Log Likelihood Ratio) $L(d_k)$ 를 계산하고 반복 복호 과정을 통해 그 값을 갱신시키게 된다. 복호에 필요한 파라미터들은 다음과 같이 가지 매트릭 δ , 순방향 매트릭 α , 역방향 매트릭 β 를 정의한다.

$$\delta_i(R_k, m) = \exp\left(\frac{2}{2\sigma^2}(x_k i + y_k Y_k^i(m))\right) \quad (4)$$

$$\alpha_k^i(m) = \delta_i(R_k, m) \sum_{j=0}^1 \alpha_{k-1}^j(S_j^i(m)) \quad (5)$$

$$\beta_k^i(m) = \sum_{j=0}^1 \beta_{k+1}^j(S_j^i(m)) \delta_j(R_{k+1}, S_j^i(m)) \quad (6)$$

$$L(d_k) = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \quad (7)$$

이 값들은 곱셈 연산으로 이루어져 있어 구현에 어려움이 있다. 그래서 식 (4)에 log를 취하는 log-MAP 알고리즘을 이용한다. 이 log-MAP 알고리즘은 Jacobian log 식을 이용하여 복잡한 곱셈 연산을 비교적 간단한 max연산과 보상값 계산으로 쉽게 구현한다.

$$\begin{aligned} \ln(e^{\delta_1} + e^{\delta_2}) &= \max^*(\delta_1, \delta_2) \\ &= \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}) \\ &= \max(\delta_1, \delta_2) + f_c(|\delta_1 - \delta_2|) \end{aligned} \quad (8)$$

식(5)을 이용하여 복호기에 필요한 파라미터들을 다음과 같이 다시 정의할 수 있다.

$$D_k^{i,m} = \frac{1}{2}(L_a(d_k)u_k^i + L_{c_k}x_k u_k^i + L_{y_k}y_k v_k^i(m)) \quad (9)$$

$$A_k^m = \max^*(A_{k-1}^{j,m} + D_k^{j,m}) \quad (10)$$

$$B_k^m = \max^*(B_{k+1}^{j,m} + D_k^{j,m}) \quad (11)$$

$$\begin{aligned} L(d_k) &= \max^*(A_k^{1,m} + B_{k+1}^{f(1,m)}) \\ &\quad - \max^*(A_k^{0,m} + B_{k+1}^{f(0,m)}) \end{aligned} \quad (12)$$

식(9)에서 u_k^i 는 현재 상태와 상관없는 조직적인 정보 비트이고 $v_k^i(m)$ 은 $S_k = m$ 인 상태에서

$d_k = i$ 일 때의 정상적인 패리티 비트이다. 그림 4는 $u_k^i, v_k^i(m)$ 의 모든 경우에 따른 가지 메트릭을 계산하는 방법을 보이고 있다. 여기서 $L_a(d_k)$ 는 사전 확률 값으로 이전 복호기의 LLR값이다. $L_c x_k$ 와 $L_c y_k$ 는 AWGN 채널을 거쳐 수신된 신호이다.[3]

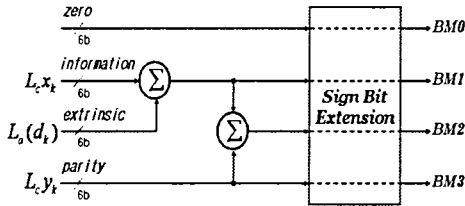


그림4 가지 메트릭 계산 구조

식(10)과 식(11)은 순방향 상태 메트릭과 역방향 상태 메트릭 값으로 식(8)을 이용하면 그림5와 같은 상태 메트릭 계산 유닛이 된다. d_k 의 값이 '1'인 경우의 BM과 SM의 합과 d_k 의 값이 '0'인 경우의 BM과 SM의 합을 비교하여 큰 값을 선택하는 MAX연산과 보상값 $f_c(\delta_1 - \delta_2)$ 을 계산하기 위한 LUT과 두 값을 더해서 MAX*를 구성하게 된다.

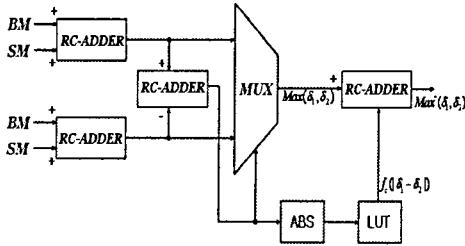


그림5 상태 메트릭 계산 구조

메트릭 값을 MAX*연산을 통해 연산하여 나가면 그 값은 점점 증가하여 일정한 비트로 제한된 메트릭 값은 결국 오버 플로우가 발생하게 된다. 오버 플로우가 발생하는 것을 방지하기 위해서는 매 복호 시간마다의 메트릭 값을 정규화 할 필요가 있다. log-MAP 방식에서 메트릭 값은 각각 상대적인 값을 비교하므로 일정한 방법으로 정규화를 거처도 복호기의 성능에는 큰 영향이 없다. 정규화 방법으로는 최소값을 빼주는 방법과 평균값을 빼는 방법 그리고 모듈로 정규화 방법 등이 있다. 최소값을 빼주는 방법과 평균값을 빼는 방법은 추가적으로 값을 계산하는 블록이 필요하고 모듈로 정규화 방법 역시 추가적인 비트를 필요로 한다. 본 논문에서는 메트릭 값 중 하나를 고정적으로 선택하여 각 메트릭 값에 빼주는 방법을 이용한다.[4]

IV. 본 논문에서 제안한 구조

1. 제안된 상태 메트릭 계산 구조

이전 질에서 언급한 상태 메트릭 계산 구조는 BM과 SM을 각각 더한 후 그 결과 값을 다시 빼주는 방식을 사용하였다. 이 방법은 결국 carry의 전파가 RC-ADDER(Ripple Carry adder) 두 개만큼의 지연을 기다려야한다. 즉 BM값과 SM값의 비트수가 클수록 carry 지연은 더욱 길어지게 될 것이다. 본 논문에서는 carry 지연을 줄이기 위하여 두 번째 RC-ADDER를 첫 번째 RC-ADDER와 병렬로 처리하여 메트릭 계산의 속도를 향상시키는 구조를 제안한다. 그림6은 본 논문에서 제안하는 개선된 상태 메트릭 계산 구조를 보여주고 있다.

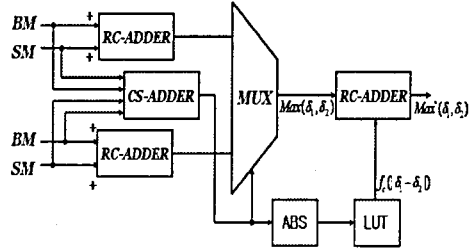


그림6. 개선된 상태 메트릭 계산 구조

제안된 구조는 차이값을 계산하기 위해 입력값을 모두 받아서 CS-ADDER(Carry Save adder)를 이용하여 차이 값을 계산한다. 이 구조는 BM값과 SM값의 비트수가 크면 클수록 기존의 방식에 비해 더욱 빠른 연산을 수행하게 된다.

2 제안된 보상값 계산 구조

log-MAP 방식과 MAX-log-MAP 방식의 차이는 상태 메트릭 계산에서 보상값 $f_c(\delta_1 - \delta_2)$ 의 고려 여부에 달려있다. 보상값을 계산하는 방법으로는 상수값 보상방법[5], 선형값 보상방법[6], LUT 참조방법이 있다. LUT을 이용하는 방법은 입력비트수에 따른 메모리의 증가를 감수하여야 한다. 이러한 부담을 줄이기 위하여 본 논문에서는 세 단계의 선형값을 보상하는 방법을 제안한다.

$$f(x) = \begin{cases} 0.75 - x/2 & , x < 1 \\ 0.25 - (x+1)/8, & 1 \leq x < 3 \\ 0 & , x \geq 3 \end{cases} \quad (13)$$

식(13)은 선형값 보상방법을 수식으로 표현하였다.

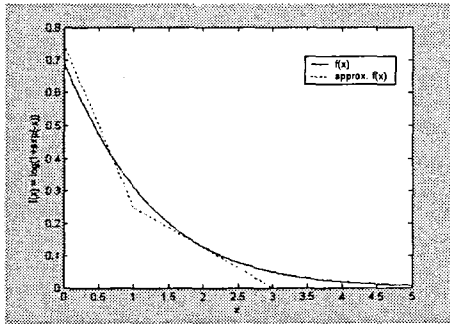


그림7 보상함수와 선형값 보상함수

그림7은 보상함수 $f(x) = \log(1 + \exp(-x))$ 의 값과 제안한 선형값 보상함수의 값을 비교한 그래프이다. 제안한 값이 보상함수의 값에 근접함을 확인할 수 있다. 상수값 보상방법은 복호기의 성능을 열화시키는 것으로 알려져 있다. 하지만 제안된 선형값 보상방법은 복호기의 성능 열화가 하드웨어의 면적을 줄일 수 있다.

V. 실험 및 결과

터보 복호기의 시뮬레이션은 Matlab과 c언어를 사용하여 구현되었으며, verilog-HDL로 구현된 모델을 Synopsys사의 Design Analyzer를 사용해 합성하여 동작속도를 측정하였다. 사용된 라이브러리는 UMC 0.25 μ m 공정라이브러리를 이용하였다. 기존의 단위 매트릭 계산 블록과 동작 속도를 비교하였으며 설계된 복호기는 3GPP의 표준[2]을 따르며 설계되었다.

표 1 설계 파라미터

파라미터	조건
구속장(K)	4
생성 다항식(g_1/g_2)	$(13)_8/(15)_8$
부호율(R)	1/3
인터리버	Mother 인터리버
블록 크기	1024
채널	AWGN
변조 방식	BPSK

BM과 SM의 비트수를 8로 하였을 때 기존의 방식과 제안한 구조의 동작속도는 16.14ns와 14.85ns로 7.99%의 속도 이득을 확인하였다. 또한 비트수를 10으로 하였을 때는 18.34ns와 16.50ns로 10.03%의 속도 이득을 확인할 수 있었다.

VI. 결 론

기존의 매트릭 계산 구조와 비교해서 동작속도를 높이면서 BER의 성능에는 큰 차이 없는

방법을 제안하였다. RC-ADDER를 직렬로 늘여 놓았던 기존의 방식에 비해 제안된 구조는 Carry의 전파로 인한 매트릭 계산 구조의 지연을 막을 수 있어 동작속도를 향상시킬 수 있다.

참고문헌

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : Turbo-Codes." Proc. ICC93, Geneva, Switzerland, pp. 1064 - 1070, MAY 1993.
- [2] TSG RAN#5, "TS25.212 v5.0.0", 3GPP(3rd Generation Partnership Project) March. 2002.
- [3] G. Lee and S. Park, "turbo Decoder Design for IS-2000 System," in Proc. VTC, pp. 412 - 415, Sept. 2000.
- [4] J. W. Woo, S. Kim, S. Y. Hwang, "저전력 및 면적 효율적인 터보 복호기를 위한 유닛 설계" 한국통신학회논문지 Vol. 28 Nov.2003.
- [5] W. J Gross and P. G Gulak, "Simplified MAP algorithm suitable for implementation of turbo decoders", Electronics Letters Volume 34, Issue 16, Page(s):1577 - 1578, 6 Aug. 1998 .
- [6] J. Cheng and T. Ottosson"Linearly Approximated Log-MAP Algorithms for Turbo Decoding" in IEEE Vehic. Tech. Conf. Tokyo, 3:2252-2256, May 2000.

※ 본 논문에 사용된 장비는 반도체 설계교육 센터(IDE)의 CAD Tool을 지원받아 사용함