

Radix-4 방식의 터보 MAP 복호 알고리즘

정희원 정 지원*, 성진숙**, 김명섭**, 오덕길**, 고성찬***

Turbo MAP Decoding Algorithm based on Radix-4 Method

Ji-Won Jung*, Jin-Suk Sung**, Myung-Sup Kim**, Duk-Gil Oh**, Sung-Chan Ko***

Regular Members

요 약

터보부호의 복호기는 두개이상의 연판정 입출력이 가능한 복호기로 구성되며, 이러한 복호기는 일정길이의 비트 열에서 최적의 사후확률(a posteriori probability)을 이용한 MAP(maximum-a-posteriori) symbol estimator를 이용한다. 기존의 radix-2 MAP복호기는 아주 큰 인터리버 블록 크기로 인해 고속 통신시스템의 적용에는 문제점이 있다. 따라서 본 논문에서는 인터리버 블록크기를 줄일 수 있는 radix-2 MAP복호기를 기반으로 하는 새로운 radix-4 MAP복호기를 제안하였다. Radix-4 MAP 복호기 구조에 적용하기 위해 순방향, 역방향 state metric과 채널 metric을 제안하였으며, 가우시안 채널에서 기존의 radix-2 기반의 MAP 복호기와 성능을 비교하였다.

ABSTRACT

The decoding of Turbo-Code relies on the application of a soft input/soft output decoders which can be realized using maximum-a-posteriori(MAP) symbol estimator^[1]. Radix-2 MAP decoder can not be used for high speed communications because of a large number of interleaver block size N. This paper proposed a new simple method for radix-4 MAP decoder based on radix-2 MAP decoder in order to reduce the interleave block size. A branch metrics, forward and backward recursive functions are proposed for applying to radix-4 MAP structure with symbol interleaver. Radix-4 MAP decoder shall be illustratively described and its error performance capability shall be compared to conventional radix-2 MAP decoder in AWGN channel

I. 서 론

무선통신시스템은 무선채널의 특성으로 비트오류가 발생하기 쉬우며 이를 정정하기위해 사용되는 채널부호는 무선통신시스템에서 매우 중요한 요소기술이다. 위성통신 및 이동통신등에서 사용되는 채널부호는 일반적으로 연판정이 가능한 길쌈부호와 연집오류특성에 강한 RS(Reed Solomon)부호를 결합한 연집부호(Concatenate Code)를 사용한다. 연집부호를 이용한 오류제어방식 또한 shannons limit에

다소 큰 격차를 보이고 있으며 이에 근접한 성능을 나타내는 터보부호에 대한 연구가 집중되고 있다. IMT-2000 시스템의 무선전송기술규격(RSPS series)이 1999년 말에 완료될 예정이므로 2000년전까지 IMT-2000 시스템에 터보부호의 적용방안이 구체화될 것으로 판단되고 또한, IMT-2000 뿐만 아니라 위성통신 및 타 이동 무선통신에서의 적용이 확대되고, 음성, 데이터, 영상등 무선 멀티미디어 서비스 개발과 함께 터보부호의적용 범위가 점차 확대될 전망이다.

1993년 Berrou 등에 의해 제안된 터보부호는

* 한국해양대학교 전파공학과 위성통신연구실(jwjung@hanara.kmaritime.ac.kr)

** 한국전자통신연구원 무선방송기술연구소 초고속위성통신연구팀

*** 안동대학교 전자정보산업학부 디지털통신시스템 연구실(kschan@anu.andong.ac.kr)

논문번호 : 99363-0908, 접수일자 : 1999년 9월 8일

※ 이 논문은 1999년도 정보통신부 대학기초과제 (C1-1999-2070-00)에 의해서 연구되었음.

Eb/No 0.7dB, 부호율 1/2에서 비트오류확률 10^{-5} 의 성능을 보였다^[1]. 터보부호는 연판정 입/출력(soft-in/soft-out)이 가능하고, 정보신호에 대해서 서로다른 인터리버에 의해 분리된 2개이상의 구성코드(component code)들이 병렬연접(parallel concatenation)된 구성을 하고 있다. 터보부호의 기본 개념은 선행하는 구성코드의 복호기 soft decision output을 다시 나머지 복호기에 입력하고 이러한 과정을 반복함으로써 향상된 decision을 가능하게 하는 것이다. 터보부호의 복호기로는 SOVA(Soft Output Viterbi Algorithm), MAP, Sub-MAP복호기등이 있는데 채널의 잡음분산평가가 필요하다라는 단점이 있지만 일반적으로 성능이 우수한 MAP을 사용한다^{[2][3]}. 이러한 MAP기반의 터보부호는 MAP복호기의 복잡성과 많은 연산량, 아주 큰 인터리버 블록크기로 인해 음성, 데이터, 동영상을 포함한 무선 멀티미디어 서비스를 요구하는 고속 무선 통신시스템의 채널부호로는 문제점을 가지고 있다. 따라서, 본 논문에서는 기존 radix-2방식을 확장해 한번의 연산으로 2비트를 동시에 복호 가능한 radix-4 방식의 MAP복호기를 제안하였다. 기존의 트렐리스 구조에서 2개의 시점을 하나의 시점으로 간주하는 Radix-4방식을 적용하기위해 branch metrics, forward recursive metrics, backward recursive metric 공식을 제안하였으며, 이를 토대로 블록크기가 300이하일 때, 이를 절반으로 줄여도 성능차이가 나지 않음을 시뮬레이션을 통해 검증하였다.

본 논문의 구성은 II장에서 기본적인 radix-2 MAP 알고리즘의 기본이론을 살펴보고 III장에서는 제안된 radix-4방식의 MAP 복호기의 trellis구조를 분석하고 기본 알고리즘에서의 각 metric 함수들을 radix-4방식의 함수들로 유도하였다. IV장에서 제안한 방법을 적용하여 기존의 방식과 성능을 컴퓨터 시뮬레이션을 이용하여 비교 분석하였으며, V장에서 결론을 맺도록 한다

II. MAP 알고리즘의 기본 이론

입력 정보비트열 $\{d_k\}$ 이 그림2.1과 같은 RSC(recursive systematic code)부호기에 의해 생성된 d_k 와 Y_k 를 QPSK 변조시키면, +1/-1값을 가지는 전송신호 a_k 와 b_k 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} a_k &= 2 \times d_k - 1 \\ b_k &= 2 \times Y_k - 1 \end{aligned} \quad (1)$$

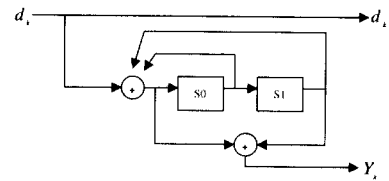


그림 1. R=1/2, v=2 인 RSC(Recursive Systematic Convolutional)부호기

시간 k 에서 전송된 신호 (a_k, b_k) 들을 심볼 C_k 라 정의하면, 가우시안 채널에 입력되는 전송신호열은 다음과 같이 주어진다.

$$C_1^N = (C_1, C_2 \dots C_k \dots C_N) \quad (2)$$

시간 k 에서 분산 σ^2 을 가지는 채널잡음 p_k, q_k 과 더해진 수신신호 x_k, y_k 는

$$\begin{aligned} x_k &= a_k + p_k \\ y_k &= b_k + q_k \end{aligned} \quad (3)$$

으로 나타내지며, 수신 심볼 $R_k = (x_k, y_k)$ 는 다음과 같다.

$$R_1^N = (R_1, R_2 \dots R_k \dots R_N) \quad (4)$$

MAP 복호기는 다음 식(5)와 같이 정의되는 LLR(Log-Likelihood Ratio)를 이용하여 복호한다. 여기서, m 은 부호기의 상태값이다.

$$\begin{aligned} L(d_k) &= \log \left(\frac{\Pr(d_k=1|observation)}{\Pr(d_k=0|observation)} \right) \\ &= \frac{\Pr(d_k=1, S_k=m| R_1^N)}{\Pr(d_k=0, S_k=m| R_1^N)} \\ &= \log \frac{\sum_m \lambda_k^1(m)}{\sum_m \lambda_k^0(m)} \end{aligned} \quad (5)$$

이것은 MAP 복호기의 softoutput 이며, 마지막엔 아래와 같이 $L(d_k)$ 를 임계값 0와 비교함으로써 복호기는 hard-decision할 수 있다.

식(5)에서 정의된 결합확률을 계산하기 위해 다음과 같이 확률함수를 정의한다.

$$\begin{aligned} \alpha_k^i(m) &= \Pr(d_k=i, S_k=m, R_k^\dagger) \\ &= \sum_{j=0}^1 \alpha_{k-1}^{b(j,m)} \delta_{k-1}^{i,b(j,m)} \end{aligned} \quad (6)$$

$$\begin{aligned} \beta_k^i(m) &= \Pr(R_{k+1}^N | d_k=i, S_k=m, R_k^\dagger) \\ &= \sum_{j=0}^1 \beta_{k+1}^{a(j,m)} \delta_k^{i,a(j,m)} \end{aligned} \quad (7)$$

α_k^i 는 순방향 recursive function 이고 β_k^i 는 역방향 recursive function 이다. 그리고 $b(i, m)$ 은 격자도 상에서 입력이 i 이고 m 으로 천이 시킨 이전의 상태값을, $f(i, m)$ 은 상태 m 에서 bit i 가 입력되었을 때 다음에 천이될 상태 값을 나타낸다.

식(6), 식(7)을 식(5)에 대입한 $L(d_k)$ 는 다음과 같다.

$$L(d_k) = \frac{\sum_m \alpha_k^m \beta_{k+1}^{f(0,m)} \delta_k^{0,m}}{\sum_m \alpha_k^m \beta_{k+1}^{f(1,m)} \delta_k^{1,m}} \quad (8)$$

$\delta_k^{i,m}$ 은 branch metrics로서 다음과 같이 표현된다.

$$\begin{aligned} \delta_k^m &= \Pr(d_k=i, S_k=m, R_k) \\ &= \exp\left(-\frac{2}{\sigma^2} \times (X_k i + Y_k C_k^{i,m})\right) \end{aligned} \quad (9)$$

여기서, $C_k^{i,m}$ 은 부호기의 상태가 m 이고 입력이 i 일 때 부호기 출력값이다.

III. Radix-4 MAP 알고리즘 제안

MAP1과 MAP2가 직렬로 구성된 radix-4 기반의 MAP 터보 복호기 구조는 그림 2와 같다.

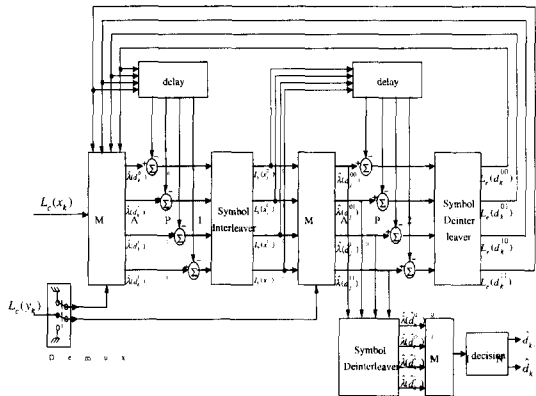


그림 2. Radix-4 기반의 MAP 터보 복호기 구조

기존의 radix-2 구조와는 달리 각 상태에서 두 비트에 대한 LLR을 동시에 계산하므로 두 비트 심볼 인터리버가 필요하며 네 개의 LLR, $\lambda(d_k^{00}), \lambda(d_k^{01}), \lambda(d_k^{10}), \lambda(d_k^{11})$ 이 각 MAP에서 구해지며, $\lambda(d_k^{00})$ 은 입력비트 "00"에 대한 LLR이다.

MAP기반의 터보부호 복호시, radix-2 방식은 임의의 k 시점에서 복호할 때 $k-1$ 시점에서의 순방향

state metric $\alpha_{k-1}^m, k+1$ 시점에서의 역방향 state metric β_{k+1}^m , 그리고 k 시점에서 branch metric δ_k^m 를 이용하여 k 시점에서 "0"과 "1"에 대한 Log Likelihood Ratio 인 LLR을 구하여 1 비트를 복호한다. 이에 반해 radix-4방식은 k 시점에서 $k-2$ 시점에서의 $\alpha_{k-2}^m, k+2$ 시점에서의 β_{k+2}^m 를 구하여 k 시점에서 2비트를 동시에 복호하기 때문에 radix-2방식보다 속도가 2배 빠르며, 저장되는 인터리버 블록 크기도 절반으로 감소할 수 있다. 과거 2단의 수신비트를 입력 받아 한꺼번에 처리하는 radix-4방식의 trellis 구조는 radix-2방식의 2개 시점을 하나의 시점으로 간주하여 처리하며 radix-2방식에서 radix-4방식으로 trellis구조 변경은 그림 3과 같다.

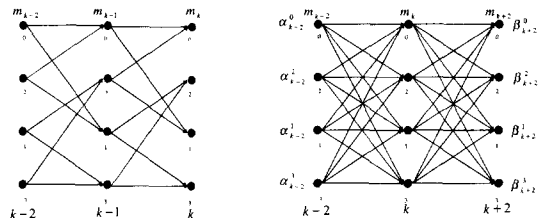


그림 3. (a) 4-state radix-2 trellis (b) 4-state radix-4 trellis.

Radix-4 trellis 상에 있어서 시간 $k-2$ 에서 k 까지의 α_{k-2}^m 연산과 k 에서 $k+2$ 까지의 β_{k+2}^m 을 구하기 위해 $k-2$ 시점에서 k 시점으로 들어오는 상태 m_{k-2} , $k+2$ 시점에서 k 시점으로 들어오는 상태 m_{k+2} 는 다음 식(10)과 같이 표현된다.

$$m_{k-2}(d_{k-1}, d_k, m_k) = (d_{k-1} \oplus d_k \oplus m_{0,k}) \parallel (d_k \oplus m_{0,k} \oplus m_{1,k})$$

$$m_{k+2}(d_{k+1}, d_{k+2}, m_k) = (d_{k+1} \oplus d_{k+2} \oplus m_{1,k}) \parallel (d_{k+1} \oplus m_{0,k} \oplus m_{1,k})$$

(10)

d_k 는 k 시점에서의 uncoded data bit 이고, m_k 는 k 시점에서의 state 번호 그리고 $(a \parallel b)$ 는 a 와 b 의 연결(concatenation)을 뜻한다.

3.1 $\delta_k^m, \alpha_k^m, \beta_k^m, \lambda_k^m$ 의 계산

평균이 0이고 분산 σ^2 인 AWGN 채널에서의 radix-4 방식의 branch metric δ_k^m 은 식(11)과 같이 유도 할 수 있다.

$$\begin{aligned}
 \delta_k^{p,m} &= P_r(D_k=p, S_k=m_1, R_k) \\
 &= P_r(D_k=i_{k-1} \| i_k, S_k=m_k, R_k) \\
 &= P_r(d_{k-1}=i_{k-1}, S_k=m_{k-1}, R_{k-1}) \\
 &\quad \times P_r(d_k=i_k, S_k=m_k, R_k) \\
 &= P_r(R_{k-1} | d_{k-1}=i_{k-1}, S_{k-1}=m_{k-1}) \\
 &\quad \times P_r(d_{k-1}=i_{k-1}, S_{k-1}=m_{k-1}) \\
 &\quad \times P_r(R_k | d_k=i_k, S_k=m_k) \\
 &\quad \times P_r(d_k=i_k, S_k=m_k) \\
 &= P_r(R_{k-1} | d_{k-1}=i_{k-1}, S_{k-1}=m_{k-1}) \\
 &\quad \times P_r(S_{k-1}=m_{k-1} | d_{k-1}=i_{k-1}) \\
 &\quad \times P_r(d_{k-1}=i_{k-1}) \\
 &= P_r(R_k | d_k=i_k, S_k=m_k) \\
 &\quad \times P_r(S_k=m_k | d_k=i_k) \\
 &\quad \times P_r(d_k=i_k) \\
 &= K_k \exp\left(-\frac{2}{\sigma^2}(x_{k-1} d_{k-1} \right. \\
 &\quad \left. + y_{k-1} Y_{k-1} + x_k d_k + y_k Y_k)\right) \quad (11)
 \end{aligned}$$

MAP 복호기의 branch metric $\delta_k^{p,m}$ 은 식(11)와 같이 전개되며 i_k 는 k시점에서의 정보비트를 의미하며, $p=i_{k-1} \| i_k$ 이다. 따라서, 정보비트 $i=\{0,1\}$ 에 대한 복호비트열 $p=\{00,01,10,11\}$ 을 얻게 된다. K_k 는 상수, Y_k 는 부호기의 state 와 입력비트 d_k 에 대한 함수이다. k시점에서의 순방향 state metric α_k^m 은 다음과 같다.

$$\begin{aligned}
 \alpha_k^m &= P_r(R_1^{k-2} | D_k=p, S_k=m_k, R_k^N) \\
 &= P_r(R_1^{k-2} | S_k=m_k) \\
 &= \sum_m \sum_{p=0}^3 p_r(D_{k-2}=p, S_{k-2}=m', \\
 &\quad R_1^{k-2} | S_k=m_k) \\
 &= \sum_m \sum_{p=0}^3 p_r(D_{k-1}=p, S_{k-2}=m', \\
 &\quad (R_1^{k-4}, R_{k-2}) | S_k=m_k) \\
 &= \sum_{p=0}^3 a_{k-2}^{(p,m_k)} \delta_{k-2}^{(p,m_k)} \quad (12)
 \end{aligned}$$

$b(p, m_k)$ 는 시점 k이고 m state에서 입력이 p일 때 시점 k-2만큼 뒤쪽으로 향하는 state의 번호를 뜻하며, 식(10)에 의해 식(13)과 같이 표현된다.

$$b(p, m_k) = m_{k-2}(d_{k-1}, d_k, m_k) \quad (13)$$

유사한 방법으로 역방향 state metric β_k^m 역시

$$\begin{aligned}
 \beta_k^m &= P_r(R_N^k | S_k=m_k) \\
 &= \sum_m \sum_{p=0}^3 P_r(D_k=p, S_{k+2}=m' \\
 &\quad \times R_N^k | S_k=m_k) \\
 &= \sum_m \sum_{p=0}^3 P_r(D_k=p, S_{k+2}=m' \\
 &\quad \times R_{k+2}^N, R_k | S_k=m_k)
 \end{aligned}$$

와 같이 나타낼 수 있으며, $f(p, m_k)$ 는 입력이 p이고 state m_k 일 때 k+2 시점만큼 앞으로 향하는 state의 번호를 뜻한다. 마찬가지로 식(10)에 의해서 다음과 같이 나타낼 수 있다.

$$f(p, m_k) = m_{k+2}(d_{k+1}, d_{k+2}, m_k) \quad (15)$$

식(12)와 (14)에 의해 사후확률(APP)는 식(16)와 같다.

$$\begin{aligned}
 \lambda_k^{p,m} &= P_r(D_k=p, S_k=m | R_1^N) \\
 &= P_r(D_k=p, S_k=m, R_1^N) / P_r(R_1^N) \\
 &= P_r(D_k=p, S_k=m, R_1^{k-2}, R_k^N) / P_r(R_1^N) \\
 &= P_r(R_1^{k-2} | D_k=p, S_k=m, R_k^N) \\
 &\quad \times P_r(D_k=p, S_k=m, R_k, R_{k+2}^N) \\
 &\quad / p_r(R_1^N) \\
 &= a_k^m \beta_{k+2}^{(p,m)} \delta_k^{p,m} \quad (16)
 \end{aligned}$$

시점 k이후의 사건은 시점 k까지 관찰한 부분에 영향을 주지 않으므로 R_1^k 와 R_{k+2}^N 은 서로 독립적이다. 최종적으로 log-MAP일 경우 radix-4방식의 LLR은 다음과 같다.

$$\begin{aligned}
 LLR(D_k) &= \text{MIN}\left\{\sum_m \lambda_k^{00}(m), \sum_m \lambda_k^{01}(m), \right. \\
 &\quad \left. \sum_m \lambda_k^{10}(m), \sum_m \lambda_k^{11}(m)\right\} \quad (17)
 \end{aligned}$$

$D_k=\{00,01,10,11\}$ 을 나타내므로 만약, $LLR(D_k) == \lambda_k^{01}$ 이면, $d_{k-1}=0, d_k=1$ 로 복호하여, 한 시점에서 두비트를 동시에 복호할 수 있다. 그림 2에서 turbo iteration 시 extrinsic정보 $L_e(d_k)$, $p=\{00,01,10,11\}$ 을 첫 번째 MAP 에서의 branch metric $\delta_k^{p,m}$ 은 식 (18)과 같이 나타낼 수 있다.

$$\begin{aligned}
 \delta_k^{p,m} &= K_k \exp\left(-\frac{2}{\sigma^2}(x_{k-1} d_{k-1} + y_{k-1} Y_{k-1} \right. \\
 &\quad \left. + x_k d_k + y_k Y_k)\right) \times L_e(d_k) \quad (18)
 \end{aligned}$$

IV. Simulation 결과 및 복잡도 분석

4.1 Simulation 결과

그림 4.에 블록크기 N=50인 radix-2 MAP과 Ns=25인 radix-4 방식의 MAP복호기를 AWGN채

널에서 성능을 나타내었다. Radix-2 방식에서 radix-4 방식으로 확장해도 성능의 열화가 없다는 것을 알 수 있다.

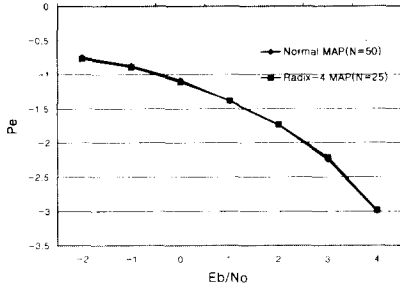


그림 4. N=50일때의 radix-2 MAP과 N=25일 때 radix-4 MAP의 성능

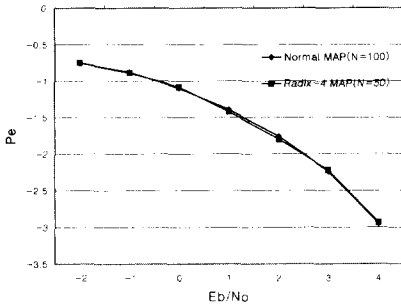


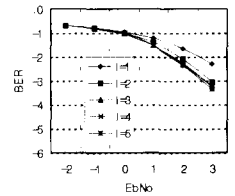
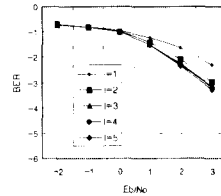
그림 5. N=100일때의 radix-2 MAP과 N=50일 때 radix-4 MAP의 성능

그림 5는 같은 조건에서 블록의 크기 N=100, Ns=50일 때의 두 복호기의 성능이다. 그림 4.에서의 결과와 마찬가지로 두 복호기의 성능이 거의 일치함을 알 수 있다.

그림 4와 그림 5는 터보 복호기중 MAP 복호기 하나일 때의 성능이며, 그림 6에서 그림 10은 N=100에서 500까지 변화시키면서 iteration 수 I를 변화하였을 때의 성능이다. Radix-2 방식일 때 N=100 이면 radix-4 방식일 때 심볼 인터리버 크기 Ns = 50을 의미하며, N이 작을 때는 iteration을 하여도 성능 열화가 없음을 알 수 있으며, N이 약 300을 넘으면 심볼 단위로 인터리버를 하기 때문에 인터리버 효과가 radix-2 방식보다 없기 때문에 성능이 약간 열화가 됨을 알 수 있다.

그림 6에서 그림 10까지 성능 분석 결과, 인터리버 크기가 300이상 일 때 반복횟수의 증가에 따라 약 2.5 dB까지는 radix-2 방식과 radix-4 방식의 성능이 거의 같으나 3dB일 때 약 0.5 dB의 성능 열화가 있음을 알 수 있다. 그러나 IMT-2000 등의 무

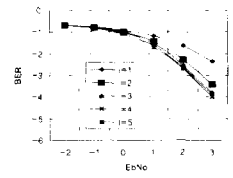
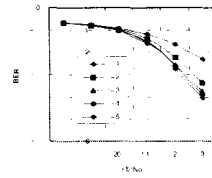
선통신에서 보통 터보부호의 적용은 블록크기가 300이하이며, 또한 무선 ATM과 같이 패킷길이가 보통 53byte이므로 블록크기가 약 500 이하이다. 따라서 터보부호의 적용은 무엇보다도 처리 속도의 문제로 인해 무선 통신의 적용에 어려움이 있기 때문에 0.5dB의 성능 차이를 극복하면서 속도를 빠르게 할 수 있는 radix-4 알고리즘의 적용은 지연 및 처리 속도를 빠르게 하여 무선 통신에의 적용이 가능하리라 사료된다.



(a) Radix-2 방식(N=100)

(b) Radix-4 방식(Ns=50)

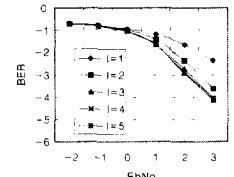
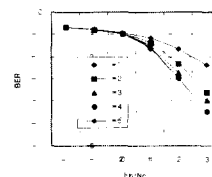
그림 6. N=100, Ns=50 일 때 radix-2 방식과 radix-4 방식 성능 비교



(a) Radix-2 방식(N=200)

(b) Radix-4 방식(Ns=100)

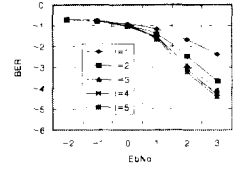
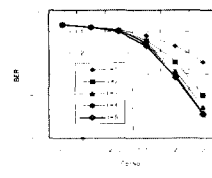
그림 7. N = 200, Ns = 100 일때 radix-2 방식과 radix-4 방식 성능 비교



(a) Radix-2 방식(N=300)

(b) Radix-4 방식(Ns=150)

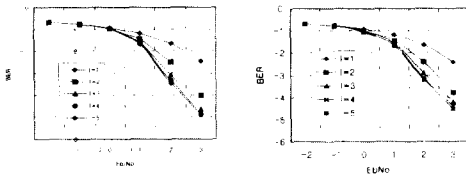
그림 8. N = 300, Ns = 150 일때 radix-2 방식과 radix-4 방식 성능 비교



(a) Radix-2 방식(N=400)

(b) Radix-4 방식(Ns = 200)

그림 9. N=400, Ns=200 일때 radix-2 방식과 radix-4 방식 성능 비교



(a) Radix-2 방식(N=500) (b) Radix-4 방식(Ns=250)

그림 10. N = 500, Ns = 250 일때 radix-2 방식과 radix-4 방식 성능 비교

4.2 복잡도 분석

표 1. Radix-2 와 Radix-4 MAP복호기의 계산량

방식 metric	Radix-2	Radix-4
δ	$N \times (2 \times 2^v) \times ((1/R)\text{exponentials and } 2^{(1-R)-1} \text{ multiplications})$	$\frac{N}{2} \times (2 \times 2^v) \times ((1/R)\text{exponentials and } 2 \times 2^{(1-R)-1} \text{ multiplications})$
α	$N \times 2^l \times (\text{ladditions and } 2 \text{ multiplications})$	$\frac{N}{2} \times 2^l \times (\text{ladditions and } 2 \text{ multiplications})$
β	$N \times 2^l \times (\text{ladditions and } 2 \text{ multiplications})$	$\frac{N}{2} \times 2^l \times (\text{ladditions and } 2 \text{ multiplications})$
LLR	$N \times (l\text{ogarithm and divisions and } 2 \times (2^{v-1})\text{additions and } 2 \times (2^v) \text{ multiplications})$	$\frac{N}{2} \times (l\text{ogarithm and divisions and } 2 \times (2^{v-1})\text{additions and } 2 \times (2^v) \text{ multiplications})$

두 복호기의 복잡도를 분석하기 위해서 표 1.에 블록의 크기 N일 때 각 metric 함수연산에 필요한 계산량을 나타내었다. 여기서, R=1/2로 부호화율이며 v는 부호기의 메모리 수를 의미한다. Branch metric δ 의 경우 radix-4 방식에서 곱셈연산이 2배로 증가하지만 전체블록의 수가 반으로 감소함으로써 전체 계산량은 radix-2방식의 결과와 같아짐을 알 수 있다. 나머지 metric연산의 경우 radix-4 방식이 radix-2 방식에 비해 계산량이 절반으로 줄어드는 것을 알 수 있다. 메모리 요구량에 있어서 radix-4 방식의 LLR의 경우 각 시점마다 p={00,01,10,11}에 해당하는 4개의 값이 필요하므로 메모리 요구량이 증가하지만 나머지 metric값에 필요한 메모리 요구량이 반으로 줄어들므로 전체적으로 메모리 요구량에 있어서도 radix-4방식이 radix-2방식에 비해 적은 메모리 요구량을 가진다. 그러나 실제 구

현시 N이 절반으로 줄어진다고 해서 계산량이 반드시 반으로 절감한다고는 볼 수 없으나 만약 radix-2 방식에서 순방향, 역방향 metric 계산 시 n 비트가 필요하다면, radix-4 방식에서는 2n 비트가 필요한 것이 아니라, n+1 비트만 필요하므로 계산량이 거의 절반가량 줄어들음을 알 수 있다.

V. 결론

MAP기반의 터보부호는 MAP복호기의 복잡성과 많은 연산량, 아주 큰 인터리버 블록크기로 인해 음성, 데이터, 동영상상을 포함한 무선 멀티미디어 서비스를 요구하는 고속 무선 통신시스템의 채널부호로는 문제점을 가지고 있다. 따라서, 본 논문에서는 기존 radix-2방식을 확장해 한번의 연산으로 2비트를 동시에 복호 가능한 radix-4 방식의 MAP복호기를 제안하였다. 기존의 트렐리스 구조에서 2개의 시점을 하나의 시점으로 간주하는 Radix-4방식을 적용하기 위해 branch metrics, forward recursive metrics, backward recursive metric 공식을 제안하였으며, 시뮬레이션 결과 메모리 크기가 작을 경우에는 성능이 일치 함을 확인하였다. 실제 IMT-2000 등에서 채택된 메모리 크기는 크지 않기 때문에 속도면에서 radix-2방식보다 빠르며, 메모리는 다소 커지리라 사료되지만, 계산량에 있어서는 radix-4방식이 radix-2방식에 비해 약 절반가량 줄일 수 있음을 알 수 있었다. 이를 토대로 radix-4 뿐만 아니라 radix-n(n=8,16,..)으로 확장 가능하리라 사료되며 고속 무선 통신에서 터보부호를 적용하기 위한 유용한 자료가 되리라 사료된다.

참고 문헌

[1] C.Berrou, A.Glavieus, and P.Thitimajshima, "Near Shanon Limit Error-Correcting Coding and Decoding: Turbo-Codes," in *proc. ICC93*, pp. 1064 - 1070.

[2] D.Divsalar and F.Pollara, "Serial and Hybrid Concatenated Codes with Applications," *Proceedings of the International Symposium on Turbo Codes & Related Topics*, pp.80-87 Sep. 1997..

[3] S.S.Pietrobo, "Implementation and Performance of a Turbo/MAP Decoder," *to be appear in International Journal of Satellite Communica-*

tions.

[4] P.Robertson, E.Villebrun, and P.Hoher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," *ICC95*, pp.1009-1013.

[5] P.Hoher, "New Iterative ("Turbo") Decoding Algorithms," *Proceedings of the International Symposium on Turbo Codes & Related Topics*, pp.63-70 Sep. 1997..

[6] S.Benedetto and G.Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schmes," *IEEE Transcation on Information Theory*, vol.42, no. 2, pp.409-429, March 1996.

정 지 원(Ji-Won Jung)



정회원
 1989년 2월: 성균관대학교 전자공학과 학사
 1991년 2월: 성균관대학교 전자공학과 석사
 1995년 2월: 성균관대학교 전자공학과 박사

1991년 1월~1992년 2월: LG 정보통신연구소 연구원

1995년 9월~1996년 8월: 한국통신 위성통신연구실 선임연구원

1998년 9월~현재: 한국해양대학교 전파공학과 조교수

<주관심 분야> 위성통신, 이동통신, 변.복조기술, 채널코딩, FPGA 기술 등

성 진 숙(Jin-Suk Sung)

정회원
 1990년: 한국과학기술원 전기 및 전자공학과 학사
 1991년~현재: 한국전자통신연구원 초고속위성통신연구팀 연구원

<주관심 분야> 고속무선 전송 모델

김 명 섭(Myung-Suk Kim)

정회원
 1986년: 한양대학교 전자공학과 학사
 1991년: 충남대학교 전자공학과 석사
 1999년: 한국과학기술원 정보및통신공학과 공학박사
 1986년 1월~현재: 한국전자통신연구원 초고속위성통신연구팀 팀장

<주관심 분야> 위성통신, 이동통신, 통신 프로토콜

오 덕 길(Duk-Gil Oh)

정회원
 1980년: 서울대학교 공과대학 전자공학과 학사
 1984년: 서울대학교 공과대학 전자공학과 석사
 1996년: 서울대학교 공과대학 전자공학과 박사
 1982년~현재: 한국전자통신연구원 지상시스템연구부장

<주관심 분야> 무선멀티미디어 전송 기술 및 서비스 시스템

고 성 찬(Sung-Chan Ko)

정회원
 1987년 2월: 성균관대학교 전자공학과 학사
 1989년 2월: 성균관대학교 전자공학과 석사
 1993년 2월: 성균관대학교 전자공학과 박사
 1993년 3월~1996년 2월: 한국전자통신연구소 위성통신연구단 선임연구원

1996년 3월~현재: 안동대학교 전자정보산업학부 조교수

1997년 2월~1998년 12월: 한국전자통신연구원 초빙 연구원

<주관심 분야> 위성통신, 이동통신, 변.복조기술, 채널코딩, VHDL 등