

초기 임계값 설정에 의한 효율적인 터보 복호기 설계

정회원 김 동 한*, 황 선 영**

Design of an Efficient Turbo Decoder by Initial Threshold Setting

Dong-Han Kim*, Sun-Young Hwang** *Regular Members*

요 약

터보 부호는 반복적인 복호 알고리즘을 사용함으로써 가산성 백색 가우시안 잡음(AWGN) 채널 환경에서 Shannon 한계에 가까운 성능을 보이는 오류정정 방식으로 제안되었으나, 반복 연산량에 따른 복호 지연과 인터리버에 따른 지연에 의해 실시간 처리의 어려움이라는 문제점을 안고 있다. 본 논문에서는 터보 부호의 성능을 저하시키지 않는 범위에서 적절한 초기 임계값 설정에 따라 불필요한 반복 복호 횟수를 줄일 수 있는 터보 복호기 구조를 제안한다. 적절한 초기 임계값 설정은 LLR(Log-Likelihood Ratio)값의 평균값과 분산, 복호기의 출력에 대한 BER에 근거하여 여러 번의 모의 실험을 통해서 최적의 값으로 결정된다. 제안한 방식은 초기 임계값을 적절히 선택하면 손실이 없는 범위내에서 반복횟수를 감소시키므로써 기존의 정해진 반복횟수로 인한 큰 복호 지연을 미연에 방지하고, 이에 따른 계산량 감소는 저전력의 효과도 가져온다. 성능 평가를 위해 BER = 10^{-6} 이내이고, 전송 속도가 32kbps이상인 IMT2000의 고속 데이터 전송 환경에서 모의 실험을 하였다. 실험 결과로 기존의 정해진 반복횟수를 갖는 터보 복호기에 비해 SNR 변동(0~3dB)에서 평균적으로 55 ~ 90% 정도의 감소된 반복횟수를 검증하였다.

ABSTRACT

Turbo Codes provide reliable, bandwidth-efficient communication over an AWGN (Additive White Gaussian Noise) channel with power efficiencies close to the Shannon limits. Despite the significant coding gains, the delays due to recursive coding structure and the interleaver adversely impact real-time performances. This paper describes the design of an efficient turbo decoder adopting a fast decoding structure. Since the performance of the turbo code is sensitive to its code structure and the number of decoding iterations, the proposed decoder has been implemented such that the number of decoding iterations can be reduced. The initial iteration threshold has been decided via simulations by varying decoder parameters (means/normalized-variances of LLR (Log-Likelihood Ratio), and the BER of the decoder outputs). Decoding iterations have been cut down to the point where an acceptable tolerance is achieved without significant performance degradation. Performance validations for the proposed decoder were performed under the IMT-2000 high-speed data transmission test-bed with the channel characteristic set to BER = 10^{-6} . Simulation results of the proposed decoder show 55% ~ 90% reductions in the number of decoding iterations when compared with the conventional turbo decoder with a fixed number of iterations.

I. 서 론

1993년 Berrou등에 의해서 제안된 터보 코드는 비교적 간단한 부호기와 인터리버를 이용하여 Shannon 한계에 가까운 오류 정정 성능을 얻는다고

보고되었다^[1]. 초기에는 하드웨어 복잡성, 복호시 시간지연으로 인한 실시간 처리의 어려움 등으로 인해 실시간 처리가 필요 없는 원거리 우주 통신용으로 연구되었으나 성능의 우수성으로 인해 다양한 분야의 응용을 위해 많은 연구가 활발히 이루어지

* 삼성전자(주) 시스템LSI 사업부 근무중
논문번호: 00283-0724, 접수일자: 2000년 7월 24일

** 서강대학교 전자공학과 교수.

고 있다. 최근에는 IMT-2000을 위한 비동기식과 동기식 규격안 양측에서 전송 속도가 낮은 음성쪽에는 길쌈코드(convolutional codes)를 사용하고, 전송 속도가 32kbps 이상이 필요한 데이터와 영상등의 고속 데이터 전송인 경우 채널 코드 방식으로 터보 코드를 채택하고 있다^{[2][3]}.

터보 코드의 복호기는 복호 알고리즘에 따라 SOVA(Soft Output Viterbi Algorithm)방식^[4]의 복호기와 MAP(Maximum A Posteriori)방식^[4]의 복호기로 분류된다. 이 복호기들은 모두 연판정값(soft-decision)을 출력하는 SISO(Soft Input / Soft Output) 방식을 사용하며, 터보 코드의 구조상 두 개가 한쌍으로 구성된 SOVA 방식이나 MAP 방식의 복호기가 번갈아 가면서 동작을 수행한다. 첫 번째 복호기는 두 번째 복호기의 연판정 출력 값으로부터 부가 정보(extrinsic information)를 구해서 이 값을 사전 확률값(APP : A Priori Probability)으로 이용하여 복호한다. 이들간에 부가정보를 반복적으로 교환함으로써 반복횟수에 비례하여 BER 성능 개선이 이루어진다^[4]. 이러한 터보 코드의 반복적인 복호 과정은 한계 횟수에 도달하기 전에는 빠르게 오류 정정 성능이 좋아지지만, 그 이상의 횟수에서는 성능이 향상되지 않는 것으로 보고되었다^[1]. 또한 채널환경이 좋은 곳에서는 목표로 하는 BER에 도달하기 위해 여러 번의 반복 복호는 요구되지 않고, 단지 몇 번의 반복으로 연산을 중지할 수 있다. 기존의 정해진 반복 복호를 갖는 터보 코드를 사용할 경우, 다양한 채널환경에서 SNR이 증가할 경우 무의미한 반복이 야기되어 복호 지연과 이에 따른 계산량 증가로 전력소모의 최적화가 필요한 이동통신기동에는 적합하지 않으므로, 무의미한 반복을 피하기 위해서 적절한 반복횟수의 선택이 필요하다.

본 논문에서, 우리는 다양한 채널환경에 따라서 반복복호 횟수를 결정할 수 있는 터보 복호기를 제안한다. 이는 기존의 터보 복호기의 무의미한 반복을 피할 수 있다. 그림 8에 결정 모듈(decision module)이 포함되어진 제안된 터보 복호기 구조를 보인다. 제안된 복호기는 결정 모듈을 사용하여 무의미한 반복복호를 피하여 복호지연을 줄인다. 결정 모듈은 두 개의 MAP 복호기의 출력 중 하나라도 임계값을 초과하면 반복복호를 종료하도록 구성이 되어있다. 이러한 과정 중, 만약 임계값이 최적의 값 이상으로 설정되면, 무의미한 반복이 야기되어 복호지연과 전력소모를 가져온다. 또한 임계값이 최적의 값 이하로 설정되면, BER 성능은 저하된다.

제안된 복호기는 다양한 채널환경에 최적의 임계값을 갖도록 구성하였다. 이 임계값은 복호기의 파라미터인 LLR(Log-Likelihood Ratio)값에 따른 평균값(means), 평균화된 분산값(normalized-variance)과 복호기의 출력에 대한 BER을 고려하여, 여러 번의 모의 실험을 거쳐서 얻어진 결과이다.

II. 터보 코드의 구조와 MAP 알고리즘

1. 터보 코드의 기본 구조

일반적인 터보 부호기는 부호율(code rate) 1/2인 두 개 이상의 동일한 생성 다항식(generate polynomial)을 갖는 순환 길쌈 부호기(Recursive Convolutional Coder)가 병렬로 연결되고, 이 사이에 인터리버가 위치한다^[1]. 그림 1은 전형적인 터보 부호기의 구조를 나타낸다. 터보 부호기가 구조적(systematic)인 형태이므로 한 프레임의 입력인 정보 비트 d_k 는 부호화 과정을 수행하지 않고 정보비트 X_k 를 출력한다. 첫 번째 부호기는 정보 비트의 원래 입력 순서에 따라 패리티 정보 Y_{1k} 를 출력하고, 두 번째 부호기는 인터리버에 의하여 순서가 재 배열된 정보 비트의 순서에 따라 이중의 패리티 정보 Y_{2k} 를 출력하게 된다. 이러한 부호화 과정은 프레임 단위로 이루어지며 한번에 부호화되는 정보 프레임의 길이는 부호기의 내부에서 사용되는 인터리버의 크기와 같다.

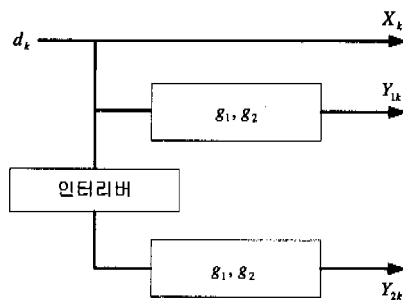


그림 1. 터보 부호기 구조

기본적으로 복호기 구조는 그림 2와 같이 각각의 구성 부호기에 대응되는 SISO 복호기가 인터리버와 디인터리버를 사이에 두고 연결된 형태로 구현되며, 부호기에서와 같이 프레임 단위로 복호가 이루어진다^[1].

AWGN(Additive White Gaussian Noise) 채널을 통과한 입력은 정보 비트 x_k 와 패리티 비트 Y_{1k} ,

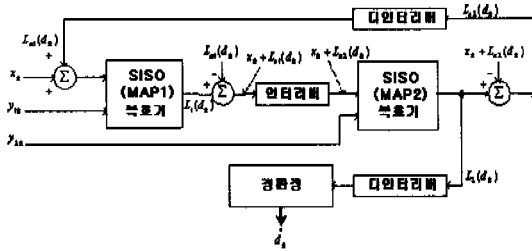


그림 2. MAP 복호기 구조

Y_{2k} 로 구성된다. 이때 첫번째 복호기(MAP1)는 입력으로 정보 비트 x_k 와 이전 정보(a priori information)인 $L_{a1}(d_k)$ 그리고 복호기 출력인 Y_{1k} 에 대응하는 패리티 비트 y_{1k} 를 받아들여 복호를 수행한다. $L_{a1}(d_k)$ 의 초기값은 '0'이다. 복호기(MAP1)의 출력은 $L_1(d_k)$ 이고, $L_{e1}(d_k)$ 는 정보 비트 d_k 의 부가 정보로서, 채널 에러를 정정하기 위해서 첫 번째 복호기에서 생성된 정정 항(correction term)이다.

첫 번째 복호기(MAP1)의 출력은 인터리버를 거치게 되며, 이는 채널에 의한 연접 에러를 랜덤하게 분산하기 위해서 사용이 되고, 인터리버의 출력인 $x_k + L_{e1}(d_k)$ 와 패리티 비트 y_{2k} 는 두 번째 복호기(MAP2)의 입력으로 사용된다. 이때 첫 번째 복호기의 부가 정보는 두 번째 복호기에 대한 이전 정보가 되고 출력은 $L_2(d_k)$ 가 되며, $L_{e2}(d_k)$ 은 복호기(MAP2)에 의해 생성된 새로운 부가 정보이며, 반복복호를 위해 첫 번째 복호기의 이전정보로 사용된다. 이제 복호기(MAP2)의 출력인 연관정값인 $L_2(d_k)$ 는 디인터리버 과정을 거친 후 경판정값(hard decision)으로 변환되어 복호된 데이터를 출력하게 되어, 이러한 과정이 터보 코드의 한 번의 복호 과정이 된다.

2. 재귀 MAP 복호 알고리즘

반복 복호를 위한 핵심 구조는 그림 3과 같은 연관정 복호기에서 각각의 복호된 비트에 대한 LLR 값을 출력할 수 있는 SISO 복호기이다. SISO 복호기의 입력은 이전 정보 $L_a(d_k)$ 와 수신된 신호 x_k 의 채널 신뢰도에 대한 LLR값인 $L_c(d_k)$ 이다. SISO 복호기의 출력은 다음 반복의 입력값이 될 부가 정보 $L_e(d_k)$ 와 이후 값 $L(d_k)$ 가 출력되어 나온다^{[5][6]}.

정보 비트 d_k 의 수신된 심볼 x_k 에 대한 조건부 확률 LLR은 식 (1)이다.

$$L(d_k | x_k) = \log \frac{P_r(d_k = +1 | x_k)}{P_r(d_k = -1 | x_k)} \tag{1}$$

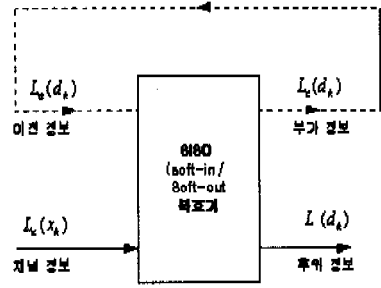


그림 3. SISO(Soft Input/Soft Output) 복호기 구조

식 (1)을 다시 표현하면, 식 (2)처럼 간단히 표현할 수 있다.

$$L(d_k | x_k) = L_c(x_k) + L_a(d_k) \tag{2}$$

여기서 $L_c(x_k)$ 는 수신된 신호 x_k 의 채널 신뢰도에 대한 LLR값으로 정의되고, 식 (3)의 수식을 얻는다. σ^2 은 잡음신호의 분산을 의미한다.

$$L_c(x_k) = \frac{2}{\sigma^2} x_k \tag{3}$$

또한 식 (2)의 $L_a(d_k)$ 는 정보 비트 d_k 에 대한 이전 정보이다. 따라서, 복호기의 출력 LLR 값은 식 (4)가 된다.

$$L(d_k) = L_c(x_k) + L_a(d_k) + L_e(d_k) \tag{4}$$

여기서 $L_e(d_k)$ 는 모든 y_{1k}, y_{2k} 와 시간 변수 k 가 아닌 모든 x_k 로 이끌어낸 부가 정보(extrinsic information)이다. 그림 2의 터보 복호기에 채널 개념을 대입하면, 첫번째 복호기(MAP1)에서는 이전 정보가 $L_a(d_k)$ 되어 복호기의 출력은 식 (5)가 된다.

$$L_1(d_k) = L_c(x_k) + L_{a1}(d_k) \tag{5}$$

그리고, 두 번째 복호기(MAP2)에서는 첫 번째 복호기(MAP1)의 이전 정보인 $L_{e1}(d_k) = L_{a2}(d_k)$ 가지므로 식 (6)의 출력을 갖게 된다.

$$L_2(d_k) = L_c(x_k) + L_{a2}(d_k) + L_{e2}(d_k) \tag{6}$$

여기서 $L_{e2}(d_k)$ 는 다음 단계의 반복 복호에서 이전 정보 값 $L_{a1}(d_k)$ 가 되어, 다시 첫 번째 복호기(MAP1)의 출력은 식 (7)과 같이 된다.

$$L_1(d_k) = L_c(x_k) + L_{a1}(d_k) + L_{e1}(d_k) \tag{7}$$

이러한 과정의 반복으로 인해 LLR값의 신뢰도는 높아져서 만족할 만한 BER성능을 얻을 수 있다.

III. 반복횟수 제어 터보 복호기

일반적인 터보 복호기는 반복적인 복호가 어떤 한계 횟수에 도달하기 전에는 빠르게 성능이 좋아지나, 그 이상의 횟수에서는 성능이 향상되지 않는 것을 알 수 있다^[1]. 그림 4는 기존의 고정된 반복횟수를 갖는 터보 복호기의 결과로서 이러한 사실을 입증한다. SNR(0~3dB)에 따라 각각 11회를 반복 복호하였으나, SNR이 10⁻⁷ 정도에 도달한 후의 반복은 사실상 무의미함을 보인다.

인터리버 크기 = 65,536

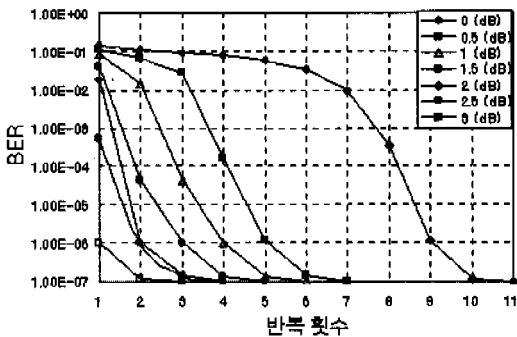


그림 4. SNR의 증가에 따른 반복횟수 비교

제안된 터보 복호기는 이러한 무의미한 반복을 피하기 위해서 간단한 초기 감지(early-detection) 부분인 결정 모듈을 추가하였다. 초기 감지에 필요한 신뢰성 있는 초기 임계값 설정을 위해서 MAP 복호의 출력인 LLR값의 평균과 분산값을 이용한 평균화된 분산값이 0.45 ~ 0.6 이내, 그리고 복호기의 출력에 대한 BER이 10⁻⁶에 가깝게 도달하는 범위 내에서 안정적인 값을 설정하였다. 또한 좀 더 효율을 높이기 위해 제안된 터보 복호기는 결정 모듈에 따라 두 복호기 중 하나라도 임계값의 범위를 넘게 되면 반복 복호를 중단하고, 연관정값을 경관정으로 바뀌어서 원하는 정보비트를 얻게 된다.

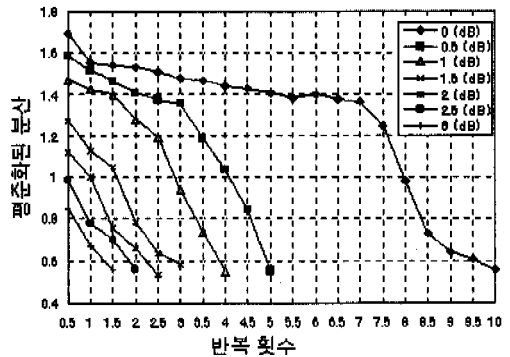
1. 평균화된 분산(Normalised Variance)

Pietrobon등은 생성 다항식 $g_1/g_2 = 23/25$ ^[7]인 터보 부호기를 사용했을 때 터보 복호기의 LLR의 평균값은 $u_L \approx L_c^2$, LLR의 분산값은 $\sigma_L^2 \approx 2u_L$

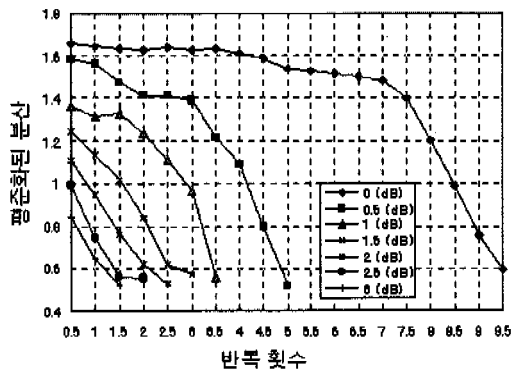
$\approx 2L_c^2$ 이 됨을 입증하였다^[8]. 또한 평균값 u_L 과 분산 값 σ_L^2 을 이용하여, LLR의 평균화된 분산값을 식(8)과 같이 정의하였다^[8].

$$\sigma_L^2 = \frac{\sigma_L^2}{u_L^2} \approx \frac{2L_c^2}{L_c^4} = \frac{2}{L_c^2} \quad (8)$$

MAP 복호기의 출력값을 평균화된 분산값인 식(8)을 사용한 결과는 SNR이 대략 3dB 정도로 증가할 때 0.5 ~ 0.6 정도의 값으로 감소하였고, 이때 BER이 대략 10⁻⁶이 되는 것을 보인다^[8]. 이와 같은 증명을 토대로 제안된 터보 복호기를 사용할 경우 인터리버 크기와 반복횟수에 따라 실험해 본 결과 참고문헌 [8]의 실험치와 거의 일치하는 LLR의 평균화된 분산값이 BER = 10⁻⁶에서 0.45 ~ 0.6 정도의 범위에 도달하는 것을 그림 5와 같이 확인하였다. 인터리버의 크기에 따라 이 값에 도달하기 위해선 반복횟수의 차이가 있었으나, 평균화된 분산값은 거의 비슷함을 보인다. 그림 5는 인터리버 크기가 다른 경우 SNR이 증가함에 따라 변화하는 반복횟수 차이와 평균화된 분산값의 변화를 보인다.



(a)



(b)

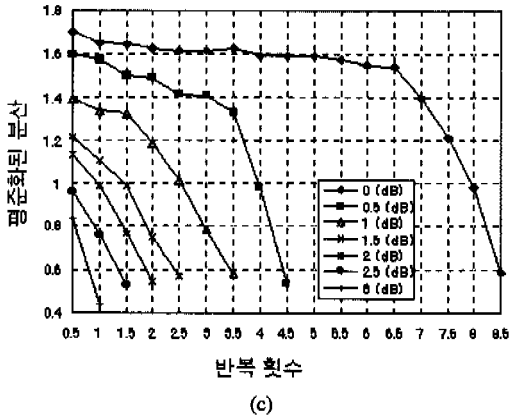
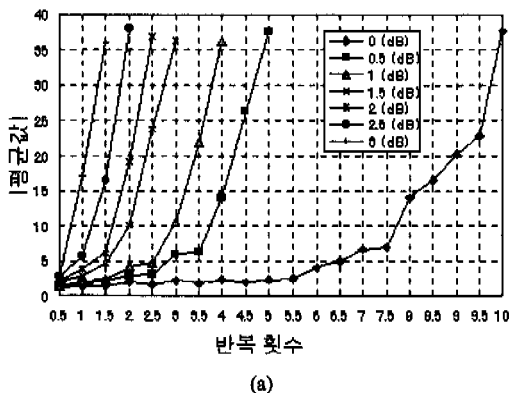


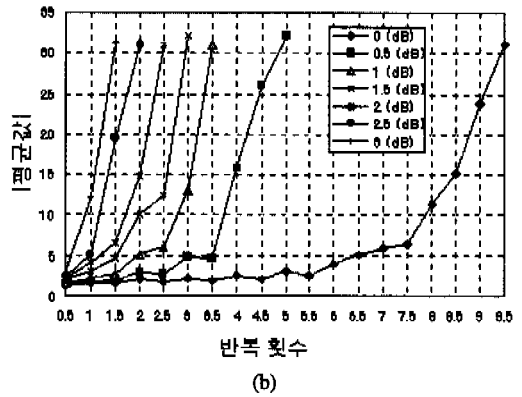
그림 5. BER = 10⁻⁶ 이내에서 반복횟수 증가에 따른 평균화 ($\sigma_i^2 = \sigma_L^2/u_i^2 \approx 2L_i^2$). (a) 인터리버 크기 = 4,096, (b) 8,192, (c) 65,536.

그림 6은 제안된 터보 복호기를 사용할 경우 BER = 10⁻⁶ 범위에서 인터리버 크기와 반복횟수에 따른 LLR의 평균값에 대한 분포를 보인다. LLR값이 양의 값과 음의 값으로 나오기 때문에 실험상 편의를 위해서 LLR값에 절대값을 적용한 평균값의 경우이다. 그리고, SNR에 따라 각각의 반복횟수를 나타내었다. 그림 6(a)는 인터리버 크기가 4,096일 경우 평균값이 36.154 ~ 37.962 범위내에서 일정하게 분포되었음을 보인다. 그림 6(b)는 인터리버 크기가 8,192일 경우 평균값이 31.016 ~ 33.076 범위내에서 일정하게 분포되어 있다. 그림 6(c)는 인터리버 크기가 65,536일 경우 평균값이 28.000 ~ 29.079 범위내에서 일정하게 분포되어 있다.

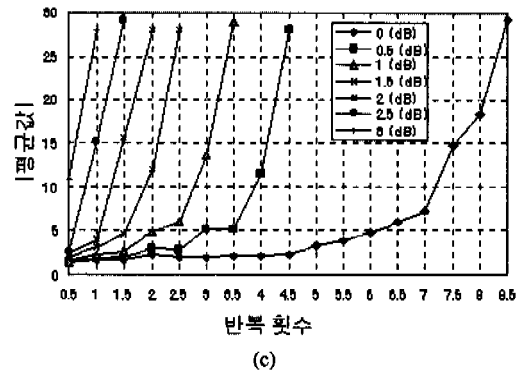
그림 4와 그림 5의 결과는 BER = 10⁻⁶ 이내가 되는 범위에서 평균화된 분산값이 0.45 ~ 0.6 범위에 도달하였고, 이때 평균값의 분포도 안정적이 되어 적절한 임계값을 설정할 수 있는 기준이 됨을 보인다.



(a)



(b)



(c)

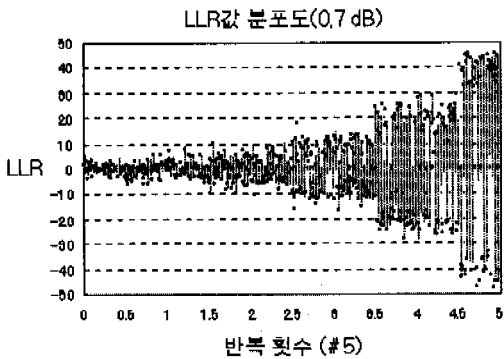
그림 6. BER < 10⁻⁶ 이내에서 반복횟수와 SNR에 따른 LLR값의 평균값. (a) 인터리버 크기 = 4,096, (b) 8,192, (c) 65,536.

2. LLR값 분포도

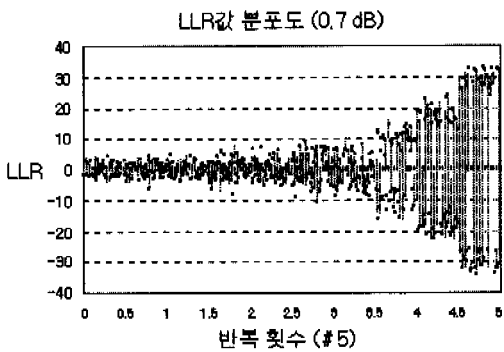
그림 7은 제안된 터보 복호기의 사용으로 각 인터리버 크기에 따라 반복 복호에 의해 LLR값이 초기 임계값에 도달함을 보인다. SNR이 0.7dB일 때, 그리고 반복횟수 증가로 인해 10⁶ 정도로 감소했을 때, 무작위로 100개 정도 추출된 LLR값의 증가와 감소된 분포를 나타내고 있다. 증가와 감소된 LLR값은 각각 경관정에 의해 '1'과 '0'의 값으로 출력된다.

그림 7(a)는 인터리버 크기가 4,096일 경우 5번의 반복 복호를 수행하게 되고, LLR값은 34 ~ 42 범위와 -34 ~ -42의 범위내에서 다소 안정된 분포를 보인다. 그림 7(b)는 인터리버 크기가 8,192일 경우 5번의 반복 복호를 수행하고 LLR값이 28 ~ 33 범위와 -28 ~ -33의 범위내에서 다소 안정된 분포를 보인다. 또한 제안된 터보 복호기는 좀 더 효율을 높이기 위해 MAP1에서도 반복을 끝낼 수 있게 하였으므로, 그림 7(c)는 인터리버 크기가 65,536일 경우 4.5번의 반복 복호으로 LLR값이 26

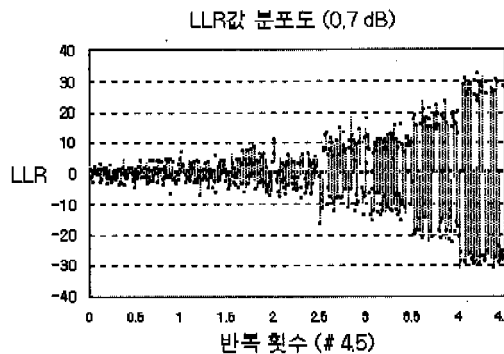
~ 30, -26 ~ -30의 범위에서 다소 안정된 분포를 보인다. 이러한 결과는 인터리버의 크기가 증가함에 따라 이득의 증가로 인해 반복횟수도 줄어드는 것을 볼 수 있다.



(a)



(b)



(c)

그림 7. E_b/N_0 가 0.7dB일 때, 반복복호로 인해 BER이 10^{-6} 에 도달한 경우의 LLR값의 분포도 (a) 인터리버 = 4,096, (b) 8,192, (c) 65,536.

그림 7은 BER이 10^{-6} 으로 감소하기까지 인터리버 크기에 따라 LLR값이 일정한 범위내에서 분포가 안정적으로 나타나는 범위가 다소 차이가 있음을 보인다. 반복 복호에 따른 LLR값의 증가와 감소가 의미하는 바는 수신된 정보비트의 신뢰도가 점차 커져서 잡음으로 인한 에러정정을 쉽게할 수 있음을 의미한다. 앞 절의 결과인 평균화된 분산과 평균값, 그리고 BER을 기준으로 LLR의 분포가 안정성을 갖는 범위를 찾아서 적정 초기 임계값을 설정한다.

3. 초기 임계값 설정

앞 절에서 확인한 결과를 토대로 신뢰적인 초기 임계값 설정을 위해서 평균화된 분산이 0.45 ~ 0.6 이내, BER = 10^{-6} 이내의 범위에서 첫 번째 복호기 MAP1과 두 번째 복호기 MAP2의 출력인 LLR값을 이용하여 임계값으로 설정하였다. 표 1은 여러 번의 실험을 거쳐서 MAP 복호기의 출력인 LLR값에 따른 평균값과 평균화된 분산, 그리고 복호기의 출력에 대한 BER = 10^{-6} 을 고려해서 산출한 초기 임계값이다.

인터리버 크기 즉, 정보 비트에 따라 각각의 임계값에는 차이가 있다. 표 1에서 보듯이 인터리버의 크기가 작을수록 성능이 떨어져서 임계값이 3 ~ 5의 차이를 보인다. 이는 성능이 좋아짐에 따라 LLR값의 신뢰도가 더 빠르게 증가함을 보여준다.

4. 반복횟수 제어 터보 복호기 구조

기존의 정해진 반복횟수를 갖는 터보 코드를 사용할 경우, 다양한 페이딩 채널을 갖는 환경에서 불필요한 반복으로 인한 복호 지연과 전력소모등을 야기할 수 있다. 제안한 터보 복호기는 기존의 터보 복호기에 간단한 결정 모듈을 추가함으로써 다양한 채널환경에서 반복횟수를 효율적으로 제어할 수 있도록 하였다. 그림 8에 제안된 터보 복호기의 구조를 보인다. 결정 모듈은 두개의 MAP 복호기의 출력값과 임계값을 비교하여 반복복호를 종료한다. 두개의 MAP 복호기의 출력($L_1(d_k)$, $L_2(d_k)$)이 임계값을 초과하게 되면 반복복호는 종료된다. MAP1 복호기의 출력이 임계값을 초과하게 되면, 복호기의 출력은 경관정값으로 변환하여 터보 복호기의 출력을 산출한다. MAP2 복호기의 출력이 임계값을 초과하게 되면, MAP2에서의 입력은 인터리버를 거쳐서 온 값이므로 출력은 다시 디인터리버를 거쳐서 경관정값으로 변환하여 터보 복호기의 출력을 산출한다. 제안된 터보 복호 알고리즘은 다음과 같다.

표 1. 인터리버 크기에 의한 초기 임계값.

Interleaver sizes (Information bits)	SNR(E_b/N_0)[dB]	Mean of LLR	Normalized-variance	Threshold
4,096	0.1 ~ 1.0	36.165 ~ 37.956	0.533 ~ 0.557	36
	1.1 ~ 2.0	35.025 ~ 37.352		
	2.1 ~ 3.0	35.121 ~ 36.230		
8,192	0.1 ~ 1.0	31.016 ~ 33.076	0.599 ~ 0.517	32
	1.1 ~ 2.0	31.298 ~ 33.015		
	2.1 ~ 3.0	31.198 ~ 32.021		
65,536	0.1 ~ 1.0	28.006 ~ 29.279	0.590 ~ 0.433	28
	1.1 ~ 2.0	27.978 ~ 28.783		
	2.1 ~ 3.0	27.275 ~ 29.001		

STEP 1: 이전 정보 $L_{ai}(d_k) = 0$

STEP 2: 첫 번째 복호기 (MAP1)는 LLR값 $L_1(d_k)$
 $= x_k + L_{a1}(d_k) + L_{e1}(d_k)$ 을 산출한다. 여기서 $L_{e1}(d_k)$ 는 $x_k + L_{a1}(d_k)$ 과 y_{1k} 에서 산출된 오류 정정 값이다.

STEP 3: 만약 $L_1(d_k) >$ 임계값, 다음과 같은 경판 정값을 산출한다:

$$\hat{d}_k = 1 \text{ when } L_1(d_k) \geq 0$$

$$\hat{d}_k = 0 \text{ otherwise.}$$

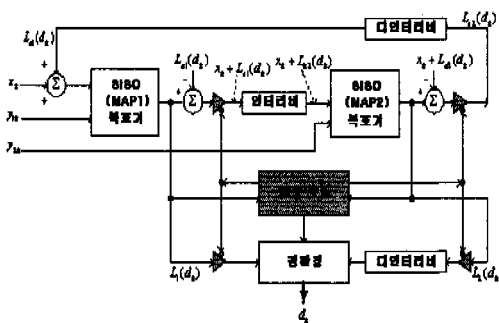
STEP 4: 두 번째 복호기 (MAP2)는 LLR값 $L_2(d_k)$
 $= x_k + L_{a2}(d_k) + L_{e2}(d_k)$ 을 산출한다. 여기서 $L_{e2}(d_k)$ 는 $x_k + L_{a2}(d_k)$ 과 y_{2k} 에서 산출된 오류 정정 값이다.

STEP 5: 만약 $L_2(d_k) >$ 임계값, 다음과 같은 경판 정값을 산출한다:

$$\hat{d}_k = 1 \text{ when } L_2(d_k) \geq 0$$

$$\hat{d}_k = 0 \text{ otherwise.}$$

STEP 6: 다시 step 2로



· $L_1(d_k)$ & $L_2(d_k)$ MAP1 복호기에서 도출된 오류 정정값.
 $L_1(d_k) = x_k + L_{a1}(d_k) + L_{e1}(d_k)$, $L_2(d_k) = x_k + L_{a2}(d_k) + L_{e2}(d_k)$
 * 정보 비트 d_k 는 외부기입 입력.

그림 8. 제안된 터보 복호기 구조

VI. 실험 결과

1. 모의 실험

모의 실험은 UNIX 환경의 C언어를 사용하여 구현하였다. 입력은 SNR과 최대 에러율, 생성다항식 $g_1/g_2 = 23/35$, 그리고 원하는 만큼의 반복횟수를 사용자가 입력할 수 있도록 하였다. 신뢰성 있는 적정 임계값을 찾아내기 위해서, 출력으로 LLR값의 평균값(mean), 분산(variance), 표준화된 분산(normalised variance), 그리고 터보 복호기의 출력에 대한 BER을 산출하였다. 이 값들을 이용해서 각 인터리버 크기에 따라 표 1에서 보인바와 같이 적정 임계값을 찾아낸 후, 제안된 터보 복호기를 구성하여 채널 환경에 따라 반복횟수의 변화를 확인할 수 있다. 모의 실험에서 사용된 파라미터들은 표 2에 제시하였다.

IMT-2000의 데이터 서비스는 138 kbps, 144 kbps, 2,048 kbps의 고속 전송율을 사용하고 지연은 200ms가 허용되며 BER < 10^{-6} 을 요구한다^{[2][3]}. 이에 따라 고속 데이터가 전송되고 복호의 처리속도가 실시간으로 된다는 가정하에서 인터리버의 크기(=정보 비트)를 지연이 200ms가 넘지 않는 범위에서 사용하였다. 인터리버 크기 65,536은 제안된 복호기와 기존의 복호기의 성능을 비교하기 위해서 사용하였다.

2. 반복횟수에 따른 결과

제안된 터보 복호기는 좀 더 효율을 높이기 위해 MAP1에서도 반복을 끝낼 수 있도록 하였으므로, MAP1에서 복호가 끝나게 되면 0.5회의 복호를 수행한 것으로 나타내었다. 그림 9는 각 인터리버 크기와 SNR의 증가에 따른 반복횟수와 BER을 비교하였다.

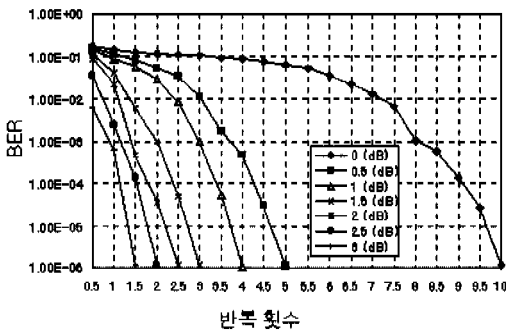
표 2. 실험에서 사용된 파라미터.

파라미터	값
구속장 (K)	5
메모리 길이 (v)	4
생성다항식(g_1/g_2)	23/35
인터리버	랜덤 인터리버
정보 비트 (N) [bits]	4,096, 8,192, 16,384
부호율	1/3
채널, 변조방식	AWGN, BPSK
복호기 알고리즘	MAP 알고리즘
E_b/N_0 [dB]	0 ~ 3 (0.5 단위)
BER	10^{-6} 이내

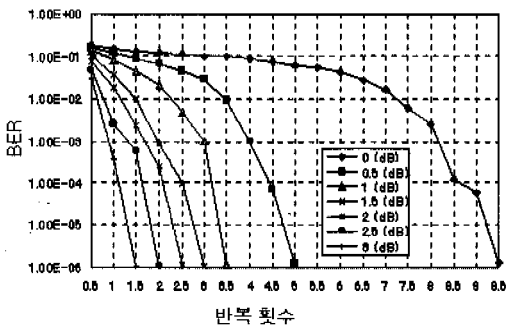
표 3과 4는 제안된 터보 복호기에서 얻어진 실험 결과를 보인다. 실험 결과는 Berrou^[1]에 의해 제안된 기존의 터보 코드의 결과와 비교되었다. 표 3은 인터리버 크기가 65,536일 때, BER = 10^{-6} 이내에 도달하기 위한 반복횟수를 보인다. 채널 환경이 비교적 나쁜 0dB에서의 비교 결과는 대략 1/2 정도의 반복횟수가 감소된 결과를 확인할 수 있다. 채널 환경이 비교적 좋은 3dB인 경우, 기존의 복호기 구조로는 18번의 반복이 요구되는 것에 비해 제안된 복호기는 단지 1번의 복호로 원하는 BER에 도달할 수 있다.

표 3. 인터리버 크기가 65,536 비트일 때, BER = 10^{-6} 에 도달하기 위한 반복복호 횟수.

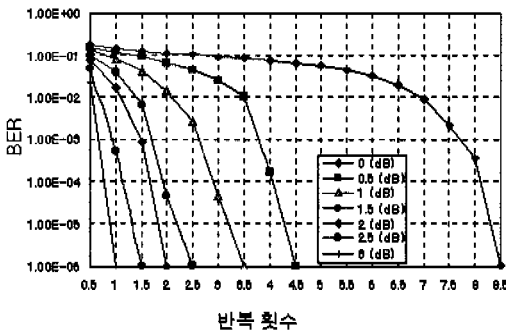
SNR [dB]	Previous Work [1]	제안된 방식	
	반복횟수	반복횟수	감소된 반복횟수(%)
0	18	9.0	50.0
0.1	18	8.5	52.8
0.5	18	4.5	75.0
1.0	18	3.5	80.6
1.5	18	2.5	86.1
2.0	18	2.0	88.9
2.5	18	1.5	91.7
3.0	18	1.0	94.5



(a)



(b)



(c)

그림 9. 인터리버 크기의 변화에 따른 모의 실험 결과
(a) 인터리버 = 4,096, (b) 8,192, (c) 65,536.

표 4는 인터리버 크기가 감소했을 때 제안된 복호기에서 요구된 반복복호 횟수를 보인다. 이 결과는 기존의 복호기가 인터리버 크기 65,536일 때와 비교된 수치이다. 제안된 복호기가 인터리버 크기 8,192일 경우, 채널환경에 따라 반복횟수가 42.2% ~ 91.7%로 감소된 결과를 보인다. 인터리버 크기가 4,096으로 감소될 경우에도 반복횟수는 거의 비슷하게 변화하는 결과를 보인다. 인터리버 크기의 감소는 면적을 줄일 수 있지만, BER의 같은 결과를 유지하기 위해서는 더 많은 반복복호가 요구되는 것을 볼 수 있다.

V. 결론 및 추후 과제

터보 코드의 강력한 오류 정정 특성은 부호기에 내재하는 인터리버와 복호기에서 수행되는 반복 복호에 기인하며, 복호 성능 향상이 복호 지연 시간에 비례하므로 실시간 처리가 필요한 이동 통신 환경

표 4. SNR(0 ~ 3dB) 대해서, BER = 10⁻⁶ 범위에 도달하기 위한 반복횟수.

SNR [dB]	Previous Work [1] (N = 65,536)	제안된 방식			
		N = 8,192	감소된 반복횟수(%)	N = 4,096	감소된 반복횟수(%)
0	18	10.0	42.2	11.0	40.0
0.1	18	9.5	47.2	10.0	45.0
0.5	18	5.0	77.2	5.0	72.2
1.0	18	3.5	80.6	4.0	77.8
1.5	18	3.0	83.3	3.0	83.3
2.0	18	2.5	86.1	2.5	86.1
2.5	18	2.0	88.9	2.0	88.9
3.0	18	1.5	91.7	1.5	91.7

에서는 보완이 필요하다.

본 논문에서 제안된 터보 복호기는 간단한 초기 감지 모듈인 결정 모듈을 추가하여 기존의 고정된 반복횟수로 인한 무의미한 반복을 피할 수 있도록 구성하였다. 반복복호의 종료를 결정할 수 있는 결정 모듈로 인해서, 제안된 터보 복호기는 채널환경에 따라 다른 반복복호를 수행한다. 실험 결과는 기존의 정해진 반복횟수를 수행하는 터보 복호기와 비교했을 때 복호시간과 전력소모 측면에서 큰 성능 개선을 보인다. 복호 종료율을 위한 최적의 임계값은 복호기의 파라미터인 LLR(Log-Likelihood Ratio)값에 따른 평균값(means), 평균화된 분산값(normalized-variance), 그리고 복호기의 출력에 대한 BER을 고려하여, 여러 번의 모의 실험을 거쳐서 얻어진 결과이다. 제안된 복호기는 반복복호의 감소로 인해 복호지연과 전력소모가 상당히 줄어들 수 있다.

추후 연구 과제로는 반복 연산량에 따른 큰 복호 지연, 이에 따른 연산량의 증가로 인한 전력소모의 해결방안이 필요하다. 또한 제안된 터보 복호기를 하드웨어 구현 시 채널환경을 고려하여 복호를 수행하는 MAP 복호와 정확한 반복횟수 결정을 위해서 수신단의 첫 부분에서 신뢰할만한 채널환경을 평가하여 성능열화를 방지할 수 있는 부분에 대한 연구가 필요하다.

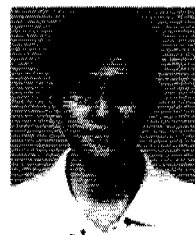
감사의말

본 논문은 서강대학교 산업기술연구소에 의해 지원되었음

참고 문헌

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, 'Near Shannon Limit Error-correcting Coding and Decoding : Turbo-Codes(1),' in Proc. ICC'93, Geneva, Switzerland, pp. 1064-1070, May 1993.
- [2] 김수영, 이수인. "터보코드(Turbo Codes) 개발 동향," 전자저널, 주간 기술동향 3, Mar. 1999.
- [3] 이문호, "IMT-2000을 위한 채널 코딩 기술," 한국통신학회지 Vol. 14, no. 9, Sep. 1996.
- [4] S. Pietrobon, 'Implementation and Performance of a Turbo/MAP Decoder,' Int. J. Satellite Commun., Vol. 16, pp. 23-46, Jan.-Feb. 1998.
- [5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, 'Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,' in Proc. IEEE Int. Symp. Inform. Theory, Asilomar, CA, p. 90, May 1972.
- [6] B. Sklar, 'A Primer on Turbo Code Concepts,' IEEE Communications Magazine, Dec. 1997.
- [7] J. Odenwalder, 'Optimal Decoding of Convolutional Codes,' Ph.D. Dissertation, Univ. of California, Los Angeles, CA, USA, 1970.
- [8] S. Pietrobon and F. Malardel, 'MAP Decoder Output Statistics and Its Effect on Turbo Decoders,' in Proc. IEEE Int. Symp. Inform. Theory, pp. 6, Sep. 1996.

김 동 한(Dong-Han Kim)



1998년 2월 : 홍익대학교 전자
전기제어공학과 졸업
2000년 8월 : 서강대학교 전자
공학과 대학원
공학석사 취득
2000년 8월 : 현재 삼성전자(주)
시스템LSI 사업부
근무중

<주관심 분야> 통신용 chip-set, 알고리즘 개발

황 선 영(Sun-Young Hwang)



1976년 2월 : 서울대학교 전자
공학과 졸업.

1978년 2월 : 한국과학기술원 전기
및 전자공학과
공학석사 취득.

1986년 10월 : 미국 Stanford
대학 전자전산

공학박사 학위 취득.

1976년~1981년 : 삼성 반도체 주식회사 연구원, 팀
장.

1986년~1989년 : Stanford 대학 Center for
Integrated Systems 연구소 책임연구원.
Fairchild Semiconductor Palo Alto
Research Center 기술자문.

1989년~1992년 : 삼성전자(주) 반도체 기술자문.

1989년 3월~현재 : 서강대학교 전자공학과 교수.

<주관심 분야> CAD 시스템, Computer Architecture
및 DSP Systems Design, VLSI 설계 등임.