

# 새로운 블럭 인터리빙 방법을 적용한 터보 부호의 Flattening Effect의 성능개선에 관한 연구

정희원 정 대 호\*, 김 환 용\*

## Performance Improvement of the Flattening Effect for Turbo Codes applying the New Block Interleaving Methods

Dae-ho Jeong\*, Hwan-yong Kim\* *Regular Members*

### 요 약

터보 부호는 종료 알고리즘(termination algorithm)과 반복복호 알고리즘(iterative decoding algorithm)을 사용하므로써 AWGN 채널환경에서 BER 성능이 우수한 것으로 알려져 있다. 그러나 터보 부호는 복호과정에서 큰 복호지연을 요구하며 음성 신호를 전송할 때 프레임의 크기가 작아서 성능의 열화가 생긴다. 또한, 높은 SNR에서는 BER 성능의 개선이 없는 Error Floor 현상이 나타나기 때문에 오류정정 능력의 한계에 도달하는 큰 단점을 가진다. Flattening Effect는 종료 알고리즘을 사용하는 터보 부호(terminated conditions : TC)보다는 종료 알고리즘을 사용하지 않는 터보 부호(non terminated conditions : NTC)에서 상대적으로 많이 나타난다. 따라서 본 논문에서는 NTC하에서 터보 부호의 Flattening Effect를 개선시키기 위한 새로운 블럭 인터리빙 방법을 제시한다. 모의실험 결과, 새로운 블럭 인터리빙 방법을 적용하므로써 효과적으로 Flattening Effect를 제거할 수 있음을 알 수 있었다.

### ABSTRACT

Turbo codes, decoded through a termination algorithm and an iterative decoding algorithm, have recently been shown to yield remarkable coding gains close to theoretical limits in the AWGN channel environment. But, in spite of the superior BER performance of turbo codes, it is required large decoding delay and caused the "Error Floor phenomenon". because of that reason, conventional turbo codes is not enough to voice transmission systems. The flattening effect is observed in turbo codes under non-terminated conditions. This effect can be effectively eliminated by changing the reading principle of the turbo interleavers. In this paper, new block interleaving methods, can be effectively eliminated flattening effect, is present. The simulation results are shown that the coding performance to be effectively improved, compared with conventional block interleaving, by removing a flattening effect.

### 1. 서 론

데이터의 신뢰도를 높이기 위한 방법으로 사용되는 오류정정 부호는 디지털 통신시스템에서 아주 중요한 요소로 자리잡고 있다. 이동통신등의 무선통신 분야에서는 전파지연 및 다중경로 전파수신에 의한 페이딩으로 인한 채널잡음 때문에 많은 전송 오류가

발생하게 된다. 따라서 이동통신처럼 대역과 전력이 제한된 상황에서는 대역폭이 효율적이면서 오류정정 능력이 우수한 채널부호가 요구되고 있다.

1993년에 C. Berrou등에 의해서 처음으로 제안된 터보 부호는 길쌈부호를 병렬 연결(parallel concatenation) 형태로 배열한 비교적 간단한 두 개의 구성 부호와 프레임 크기가 큰 인터리버를 가지고 있어서

\* 원광대학교 전자공학과(hykim@wonkwang.ac.kr, jdh@gaebiyok.wonkwang.ac.kr)

논문번호 : K01026-0118, 접수일자 : 2001년 1월 18일

※ 이 논문은 2000년도 원광대학교의 교비지원에 의해서 연구됨

사논의 한계에 근접하는 아주 우수한 오류정정 능력을 가지는 것으로 알려지고 있다.<sup>[1]</sup> 터보 부호가 발표된 당시에는 긴 프레임 길이에 따른 복호 지연으로 인한 실시간 처리의 어려움으로 우주 통신용으로만 이용되었으나 최근에 셀룰라 통신이나 이동통신 시스템과 같은 짧은 프레임(일반적으로 300비트 이하) 단위로 정보를 송·수신하는 무선통신 시스템에 터보 부호를 적용하려는 연구가 진행되고 있다.<sup>[2]</sup>

Berrou 등의 논문에서는 변형된 MAP 알고리즘을 제시하여 터보 부호의 복호동작으로 사용하였으나 여전히 복잡한 구조는 크게 개선되지 않았다. 1994년에 S. S. Pietrobon 등에 의해서 복잡도를 줄이고 하드웨어 구현이 가능한 구조로 MAP 알고리즘을 단순화한 논문이 제시되었다.<sup>[3]</sup> 그러나 여전히 우수한 BER 성능에도 불구하고 Berrou의 터보 부호는 많은 연산량으로 인한 복잡도의 증가, 하드웨어 구현상의 어려움 등의 문제점을 안고있으며 특히, 높은 SNR에서 BER 성능곡선이 평평한 특성(Flattening Effect)을 나타내기 때문에 이러한 문제점을 개선시키고자 많은 연구가 진행되고 있다.<sup>[4]</sup>

터보 부호의 성능을 개선시키는데 있어서 종료 알고리즘을 적용한 시도는 많이 이루어지고 있으나 종료 알고리즘을 적용하지 않은 NTC하에서의 연구는 미비한 실정이다. Flattening Effect는 TC하에서의 터보 부호보다는 NTC하에서 상대적으로 많이 나타난다.

본 논문에서는 기존의 인터리빙 방법을 확장시킨 새로운 블럭 인터리빙 방법을 적용하여 NTC하에서 터보 부호의 단점인 Flattening Effect 즉, Error Floor 현상을 감소시킬 수 있는 방법을 제시하였다. 그리고 제시된 방법을 이용하여 기존의 인터리빙 방법과 비교하였고 MAP 알고리즘을 적용하여 터보 부호의 성능을 분석하였다.

## II. 터보 부호의 부호화 및 복호화

그림 1은 인터리버와 두 개의 RSC(Recursive Systematic Convolutional) 부호기로 구성된 부호율이 1/2이고 구속장 K=3인 터보 부호의 부호기를 나타낸다.

그림 1의 터보 부호기는 1비트 입력( $d_k$ )에 2비트씩 출력( $X_k, Y_k$ )되는데  $X_k$ 는  $d_k$ 와 같은 값으로서 systematic 정보를 나타내며,  $Y_k$ 는  $y_{1k}$ 와  $y_{2k}$ 를 교대로 평처멍(puncturing)한 값으로서 parity 정보를 나타낸다.

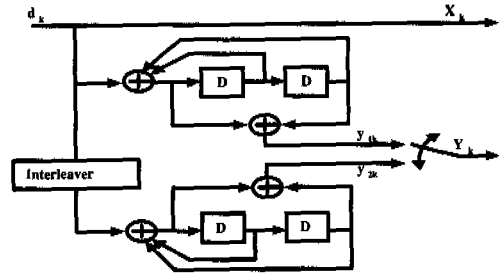


그림 1. 터보 부호기(부호율 1/2, 구속장 3인 경우)

터보 부호기의 입력은 같은 비트열 ( $d_k$ )을 사용하지만 그 사이에 인터리버가 있기 때문에 서로 다른 순서를 갖는 비트열이 채널을 통해서 전송된다. 각각의 RSC 부호기에 입력되는 비트열은 서로 상관관계가 적을수록 성능이 우수해지므로 적은 상관관계를 갖는 인터리버의 설계가 필요하다.

그림 2는 2개의 MAP 복호기와 인터리버, 디인터리버로 구성된 터보 부호의 MAP 복호기를 나타낸다.

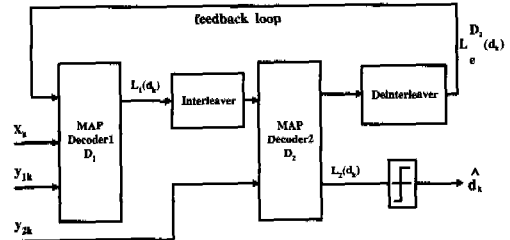


그림 2. 터보 부호의 MAP 복호기

MAP 복호기에서 오류를 정정하기 위해 사용하는 복호 알고리즘으로는 MAP (Maximum a Posteriori) 알고리즘과 SOVA(Soft Output Viterbi Algorithm)가 사용되고 있는데 MAP 복호기가 BER 성능면에서 우수한 것으로 알려져 있다.

MAP 알고리즘은 이산 무기억 채널을 통해서 전송된 Markov source의 상태와 천이에 대한 APP(a Priori Probability)를 계산하여 복호동작을 수행하는 것으로 격자복호(trellis decoding)에 대하여 심볼의 에러율을 최소화하는 알고리즘이다.

시간  $k$ 에서 MAP 복호기의 입력이 다음과 같이 BPSK 변조된 수신신호  $R_k^N = (R_1, R_2, \dots, R_N)$ 이라 하자. 여기서  $R_k = (X_k, Y_k)$ 는 시간  $k$ 에서 수신된 심볼을 나타내며  $X_k$ 와  $Y_k$ 는 각각 다음과 같다.

$$X_k = (2d_k - 1) + n_k \tag{1}$$

$$Y_k = (2p_k - 1) + m_k \tag{2}$$

여기서  $d_k, p_k$ 는 각각 부호기의 출력값이며  $n_k, m_k$ 는 각각 백색 가우시안 잡음(a White Gaussian Noise)으로서 평균(mean)이 0(zero)이고 분산(variance)이  $\sigma^2$ 인 값이다. 본 논문에서 사용한 신호대 잡음비의 정의는 다음과 같다.<sup>[5]</sup>

$$\frac{S}{N_0} = \frac{E_b}{N_0} R \text{ [dB]} \quad (3)$$

MAP 알고리즘의 목적은  $R_1^N$ 값이 수신된 후 수신된 심볼에 대하여 유사도(likelihood ratio)를 구하여 정보 비트  $d_k$ 를 결정하는 것이다. 이러한 과정은 다음과 같이 정의되는 각각의  $d_k$ 에 대한 LLR (Log-Likelihood Ratio)  $\Lambda_k$ 값을 계산하므로써 얻어 질 수 있다.

$$\Lambda_k(d_k) = \log \left( \frac{P_r(d_k=1 | R_1^N)}{P_r(d_k=0 | R_1^N)} \right) \quad (4)$$

복호된 비트  $d_k$ 의 APP는 결합 확률(Joint Probability)의 형태로 나타나며 다음과 같은  $\lambda_k^i(m)$ 으로 정의된다.

$$\lambda_k^i(m) = P_r(d_k=i, S_k=m | R_1^N) \quad (5)$$

여기서  $i$ 는 0과 1이고,  $m$ 은 0, 1, 2, 3,  $2^{\nu-1}$ 로서 상태지수(state index)이고  $\nu$ 는 각각의 길쌈부호기의 메모리 갯수이다. 식 (5)에서  $S_k$ 는 시간  $k$ 에서의 부호기 상태이며 이를 이용하면 다음과 같은 결과를 얻을 수 있다.

$$\begin{aligned} P_r(d_k=i | R_1^N) &= \sum_{m=0}^{2^{\nu}-1} \lambda_k^i(m) \\ &= \sum_{m=0}^{2^{\nu}-1} P_r(d_k=i, S_k=m | R_1^N) \end{aligned} \quad (6)$$

여기서  $i$ 는 0과 1이며 식 (6)의 결과를 이용하여 복호된 비트  $d_k$ 와 관련되는  $\Lambda_k$ 는 다음과 같이 변형된다.

$$\Lambda_k(d_k) = \log \left( \frac{\sum_{m=0}^{2^{\nu}-1} \lambda_k^1(m)}{\sum_{m=0}^{2^{\nu}-1} \lambda_k^0(m)} \right) \quad (7)$$

여기서  $\lambda_k^i(m)$  값을 구하기 위하여 다음과 같은 파라미터들을 정의한다.

$$\alpha_k^i(m) = P_r(d_k=i, S_k=m, R_1^k) \quad (8)$$

$$\beta_k^i(m) = P_r(R_{k+1}^N | d_k=i, S_k=m) \quad (9)$$

위의 두 파라미터  $\alpha_k^i(m), \beta_k^i(m)$ 는 두 값 모두 정보 비트  $d_k$ 값에 의존하게 되며 Bayes' rule을 적용하여 식 (5)의  $\lambda_k^i(m)$ 를 다시 나타내면 다음과 같다.

$$\lambda_k^i(m) = \frac{\alpha_k^i(m) \beta_k^i(m)}{P_r(R_1^N)} \quad (10)$$

이를 다시 식 (7)에 적용하면 다음과 같다.

$$\Lambda_k(d_k) = \log \left( \frac{\sum_{m=0}^{2^{\nu}-1} \alpha_k^1(m) \beta_k^1(m)}{\sum_{m=0}^{2^{\nu}-1} \alpha_k^0(m) \beta_k^0(m)} \right) \quad (11)$$

앞에서 언급한  $\alpha_k^i(m), \beta_k^i(m)$ 를 Bayes' rule을 적용하여 다시 정리하면 식 (12), 식 (14)와 같으며 그림 3과 그림 4는 이들 파라미터에 대한 도식적인 표현을 나타낸다.

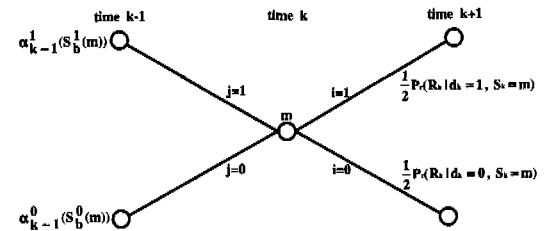


그림 3.  $\alpha_k^i(m)$ 값에 대한 도식적인 표현

$\alpha_k^i(m)$ 값은 순방향 상태 매트릭(forward state metric)이라 하며 이것은 정보 비트  $i$ 의 천이 비트를 가지고 시간  $k$ 와 그 때의 상태  $S_k(m)$ 에서 다음 상태로의 천이를 위한 상태 매트릭 값을 나타낸다.

$$\begin{aligned} \alpha_k^i(m) &= P_r(d_k=i, S_k=m, R_1^k) \\ &= \delta_i(R_k, m) \sum_{j=0}^{2^{\nu}-1} \alpha_{k-1}^j(S_{k-1}^j(m)) \end{aligned} \quad (12)$$

여기서  $\delta_i(R_k, m)$ 은 가지 매트릭(branch metric)값으로 다음과 같으며  $S_{k-1}^j(m)$ 은 이전 상태를 나타낸다.

$$\delta_i(R_k, m) = P_r(R_k | d_k=i, S_k=m)/2 \quad (13)$$

$\beta_k^i(m)$ 값의 계산은  $\alpha_k^i(m)$ 값의 계산 과정과 유사하게 나타나며 단지 계산의 진행 과정이  $\alpha_k^i(m)$ 과는 반대인 역방향 상태 매트릭(backward state metric) 형태로 나타난다.

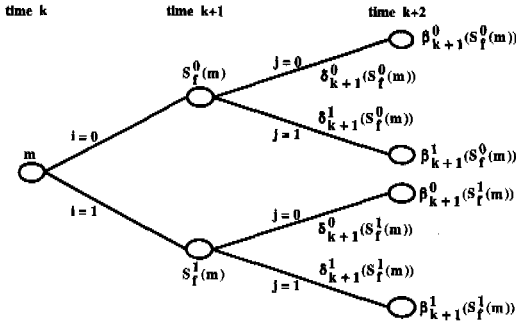


그림 4.  $\beta_k^i(m)$  값에 대한 도식적인 표현

$$\beta_k^i(m) = P_r(R_{k+1}^N | d_k = i, S_k = m) = \frac{1}{2} \sum_{j=0}^1 \beta_{k+1}^j(S_k^i(m)) \delta_j(R_{k+1}, m) \quad (14)$$

여기서  $\delta_j(R_{k+1}, m)$ 는 가지 매트릭(branch metric) 값으로 다음과 같으며  $S_k^i(m)$ 은 다음 상태를 나타낸다.

$$\delta_j(R_{k+1}, m) = P_r(R_{k+1} | d_{k+1} = j, S_{k+1} = m) / 2 \quad (15)$$

MAP 알고리즘을 터보 부호의 반복부호 알고리즘에 적용하면 임의의 시간  $k$ 에서의 입력 심볼에 대한 LLR은 위에서 정의한 식 (4)와 같다. 식 (4)에서  $P_r(d_k = i | R_k^N)$ 는 수신 심볼  $R_k^N$ 에 대한  $d_k = i$ 의 APP이며, 복호기는  $\Lambda_k(d_k)$ 에 대한 임계값을 통하여 다음과 같은 정보 비트  $\hat{d}_k$ 를 얻게된다.

$$\hat{d}_k = \begin{cases} 1, & \Lambda_k(d_k) \geq 0 \\ 0, & \Lambda_k(d_k) < 0 \end{cases} \quad (16)$$

위의 결과식을 터보 부호의 복호기에 적용하면 출력  $\Lambda_k(d_k)$ 는 다음과 같은 표현식으로 나타낼 수 있다.

$$\Lambda_k(d_k) = \Lambda_c x_k + \Lambda_a(d_k) + \Lambda_e(d_k) \quad (17)$$

식(17)에서  $\Lambda_c x_k$ 는 시간  $k$ 에서 수신된 systematic 심볼에 대한 것만을 나타내며  $\Lambda_a(d_k)$ 는 입력 비트  $d_k$ 에 대한 APP를 나타낸다. 그리고  $\Lambda_e(d_k)$ 는 입력 비트  $d_k$ 와 관계하여 나타나는 새로운 정보인 외부 정보(extrinsic information)로서 시간  $k$ 에서 systematic 데이터를 제외한 모든 데이터와 패리티 비트들에 대한 정보를 가지게 된다. 이러한  $\Lambda_e(d_k)$ 에 대한 정보를 이용하여 터보 부호의 MAP 복호기에서는 반복부호 동작을 수행하며 이러한 과정을 나타낸 MAP 복호기의 전체 블럭도는 그림 2와 같다.

### III. Non-Terminated Conditions (NTC)과 Flattening Effect

각각의 MAP 복호기는 수신된 비트에 대한 연관정 출력값(soft-output value)을 생성하기 위해서 상태 매트릭 값  $\beta_N(m)$ 을 역방향으로 계산하게 된다.

NTC하에서  $\beta_N(m)$ 는 두 개의 MAP 복호기에서 매 상태마다 똑같은 확률로 초기화되는데 다음과 같이 나타낼 수 있다.

$$\beta_N(m) = 1/2^\nu \quad (18)$$

여기서,  $m$ 은  $0, 1, 2, 3, \dots, 2^\nu - 1$ 로서 상태지수(state index)를 나타내고  $\nu$ 는 각각의 길쌈 부호기의 메모리 갯수이다. 이 식에서 알 수 있는 바와같이 초기화가 정확하지 않다면 터보 부호의 전체적인 성능이 떨어지는 결과를 가져올 것이다. 또한, 프레임의 끝부분에 있는 데이터는 복호시에 오류가 발생하기 쉬워서 이러한 데이터의 처리가 터보 부호에 있어서 중요한 문제가 된다. 이러한 문제를 종료(termination)문제라고 하는데 프레임의 크기가 큰 경우는 종료 알고리즘(termination algorithm)에 의한 성능에 미치는 영향은 작지만 이동통신 시스템 등과 같이 작은 프레임을 갖는 음성을 전송하는 경우에는 음성의 성능이 종료 알고리즘에 따라 크게 영향을 받게된다. 일반적으로 터보 부호에서는 이러한 초기화의 부정확성으로 인한 성능의 감소를 개선해 주기 위한 한 가지 방법으로 터보 부호의 부호기를 종료시키는 종료 알고리즘을 사용한다. TC하에서는 터보 부호의 첫 번째 부호기를 종료 시키기 위해서 메모리 갯수 만큼인  $\nu$ 개의 꼬리비트(tail bits)를 부호기의 끝에 첨가하고 두 번째 부호기의 격자(trellis)는 종료시키지 않고 놓아둔다. TC하에서 첫 번째 복호기에 대한 상태 매트릭 값  $\beta_N(m)$ 의 초기화를 나타내면 다음과 같으며 두 번째 복호기에 대한 초기화는 식 (18)을 그대로 사용한다.

$$\beta_N(0) = 1, \quad \beta_N(m) = 0 \quad (19)$$

여기서,  $m$ 과  $\nu$ 의 정의는 식 (18)과 동일하고 단지  $m$ 의 구간이  $1, 2, 3, \dots, 2^\nu - 1$ 이다. NTC와 TC를 비교할 때 TC의 장점은 첫 번째 복호기의 상태 매트릭  $\beta_N(m)$ 의 정확한 초기화에 있다. 기존의 블럭 인터리머가 사용된다 할지라도 TC를 적용해서 성능을 분석해 보면 마지막 비트에

대해서 추가적인 복호 에러가 발생되지 않는다는 것을 알 수가 있지만 NTC와 비교해서 TC하에서의 성능 개선은 BER이  $10^3$ 에서 단지 0.7dB~1.2dB 정도임을 알 수 있다. 터보 복호기에서 모든 비트에 대한 외부 정보는 두 개의 복호기 사이에서 교차한다. 기존의 터보 부호의 우수한 성능은 반복복호로부터 달성될 수 있다.

기존의 블럭 인터리버 동작원리를 사용할 때 두 개의 구성 부호기의 입력 시퀀스는 똑같은 시작비트와 마지막 비트를 가지게 된다. 이러한 부정확한 상태 매트릭은 각 복호기에 대해서 마지막 비트에 대한 추정값의 신뢰도를 떨어뜨리고 다른 복호기에 영향을 주어서 마지막 비트의 연판정 출력값을 개선시키는데 도움을 주지 못하며 수많은 복호 에러를 발생시킨다. 다른 비트들의 추정값은 큰 SNR에서 반복 후에 상당히 정확한 반면 마지막 비트의 추정값은 복호 에러로 인해서 전체적인 BER 성능에 악영향을 끼친다. 이러한 이유로 큰 SNR에서 종료 알고리즘을 사용하지 않는 NTC하에서의 터보 부호의 성능은 평평(flat)하게 되고 더이상 성능곡선이 아래로 떨어 지지 않아서 성능 개선이 없게된다. 따라서, 본 논문에서는 종료 알고리즘을 사용하지 않는 NTC하에서의 터보 부호의 Flattening Effect를 개선시킬 수 있는 새로운 블럭 인터리빙 방법을 제시하여 기존의 블럭 인터리빙 방법과 비교하였다.

#### IV. 제안된 블럭 인터리빙 방법

##### 1. 기존의 블럭 인터리빙 방법

가장 많이 사용하는 블럭 인터리버로는 기존의 블럭 인터리버와 역 블럭 인터리버가 있다. 기존의 블럭 인터리버는 데이터를 행방향으로 쓰고 열방향으로 읽어내는 구조로서 4x4 행렬 형태로 읽고 쓰는 동작을 나타내면 다음과 같다.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

(a) 쓰기(write) 동작 (b) 읽기(read) 동작

그림 5. 기존의 블럭 인터리버 동작

역 블럭 인터리버는 기존의 블럭 인터리버와 반대로 데이터를 열방향으로 쓰고 행방향으로 읽어내는 구조로서 읽고 쓰는 동작을 나타내면 다음과 같다.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

16	12	8	4
15	11	7	3
14	10	6	2
13	9	5	1

(a) 쓰기(write) 동작 (b) 읽기(read) 동작

그림 6. 역 블럭 인터리버 동작

##### 2. 제안된 블럭 인터리빙 방법

제안된 블럭 인터리빙 방법에 대한 동작 원리를 4x4 행렬 형태로 그림 7에 나타내었다. 그림에 나타나 있는 표기법은 인터리버 내에 저장되어 있는 데이터를 읽어내는 방법에 따라서 기존의 인터리버의 읽는 동작을 down-right, 역 인터리버의 읽는 동작을 up-left, 본 논문에서 제시한 인터리버의 읽는 동작을 down-left, up-right 이라고 나타내기로 한다.

먼저, 입력 시퀀스는 4x4 행렬 형태를 가정하였으므로 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}으로 나타낼 수 있다. 입력 시퀀스를 인터리버 내에 저장하는 동작은 기존의 블럭 인터리버의 방법과 동일하게 row-by-row, left-to-right, top-to-bottom 방식으로 저장할 수 있다. 출력 시퀀스를 생성하기 위해 기존의 블럭 인터리빙 방식인 down-right이 사용되는 경우에 입력과 출력시퀀스는 프레임의 시작과 끝에서 같은 비트를 가지게 된다.

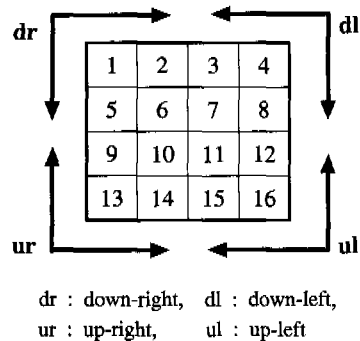


그림 7. 읽기(read) 동작을 다르게 한 블럭 인터리빙 방법

이러한 방식은 상대적으로 낮은 자유거리를 가지는



새로운 블럭 인터리빙 방법과 기존의 블럭 인터리빙 방법에 대하여 모의실험을 수행하였고 그들의 BER 성능을 분석하였다.

성능분석을 위한 파라미터는 표 1과 같이 설정하였고 모의실험에서 사용된 터보 부호의 각 구성장에 따른 최적의 tap connections은 표 2와 같이 설정하였다.

표 1. 성능분석을 위한 파라미터

채널	AWGN		
구속장(K)	K=3, 4, 5		
생성다항식(G)	K=3	K=4	K=5
	$[111;101] = (7, 5)_8$	$[1011;1111] = (13, 17)_8$	$[10011;11101] = (23, 35)_8$
부호율(R)	1/2, 1/3		
인터리버	블럭 인터리버(dr, ur, ul, dl)		
프레임 크기	128(8×16),		1024(32×32)
종료 형태	NTC, TC		
변조 형태	BPSK		
Tools	MATLAB ver. 5.3		

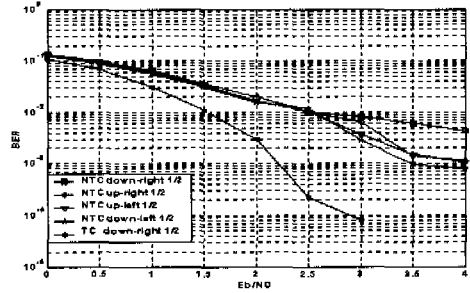
표 2. 최적의 Tap Connections

K=3	K=4	K=5
(7, 5) 부호	(13, 17) 부호	(23, 35) 부호

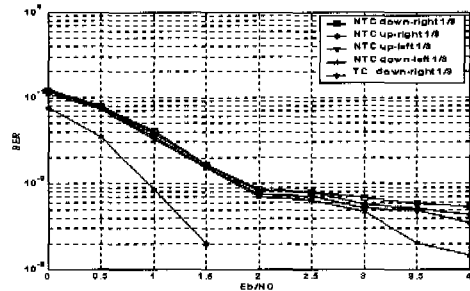
AWGN 채널을 통한 터보 부호의 성능분석 결과를 보면 다른 논문들에서 제시된 바와 마찬가지로 구속장이 클수록, 프레임 크기가 클수록 그리고 평처링을 사용하지 않았을 때에 성능의 향상이 상대적으로 증가하였다. 그 이유는 순방향과 역방향으로 프로세서를 수행하는 동안 trellis diagram에서 올바른 경로를 찾아낼 기회를 많이 가지게 되기 때문으로 파악된다. 본 논문에서는 K=3일 때에 대체적으로 0.5dB 이상의 성능 향상을 보였고 K=5일 때는 2.0dB 이상의 성능 향상을 보였다.

그림 9는 K=4인 경우에 프레임 크기를 128로 설정하고 부호율을 서로 다르게 하여 각 블럭 인터리빙 방법에 대한 모의실험 결과를 나타낸 것이다. 부호율이 1/2인 경우, 새로운 블럭 인터리빙 방법이 기존의 블럭 인터리빙 방법에 비하여 10<sup>-3</sup>의 BER에서 0.7dB 이상의 성능 향상을 나타냄을 확인할 수 있었다.

부호율이 1/3인 경우, 새로운 블럭 인터리빙 방법이



(a) 부호율이 1/2인 경우

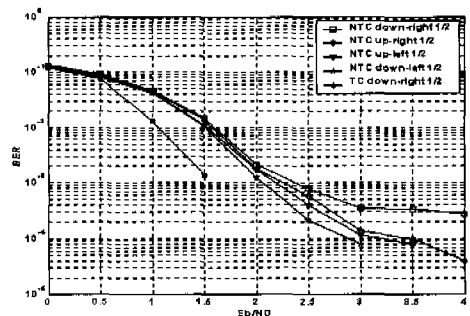


(b) 부호율이 1/3인 경우

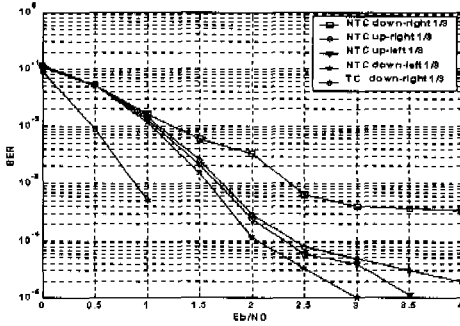
그림 9. 프레임 크기가 128일 때 각 인터리빙 방법에 대한 성능분석 곡선(k=4)

10<sup>-3</sup>의 BER에서 1.0dB 이상의 성능 향상을 나타냄을 확인할 수 있었다.

그림 10은 K=4인 경우에 프레임 크기를 1024로 설정하고 부호율을 서로 다르게 하여 각 블럭 인터리빙 방법에 대한 모의실험 결과를 나타낸 것이다. 부호율이 1/2인 경우, 새로운 블럭 인터리빙 방법이 기존의 블럭 인터리빙 방법에 비하여 10<sup>-3</sup>의 BER에서 1.3dB 이상의 성능 향상을 나타냄을 확인할 수 있었다. 부호율이 1/3인 경우, 새로운 블럭 인터리빙 방법이 10<sup>-3</sup>의 BER에서 1.5dB 이상의 성능 향상을 나타냄을 확인할 수 있었다.



(a) 부호율이 1/2인 경우



(b) 부호율이 1/3인 경우

그림 10. 프레임 크기가 1024일 때 각 인터리빙 방법에 대한 성능분석 곡선(k=4)

또한, 기존의 블록 인터리빙 방법은 프레임 크기가 128비트인 경우에, 그림 9(a)와 그림 9(b)에서 나타난 바와같이 Eb/No가 2.0dB와 2.5dB 이상에서 error floor 현상이 나타나고 있음을 확인할 수 있었고 프레임 크기가 1024비트인 경우에, 그림 10(a)와 그림 10(b)에서 나타난 바와같이 3.0dB 이상에서 error floor 현상이 나타나고 있음을 확인할 수 있었다.

### VI. 결론

터보 부호의 인터리버는 인접 비트간의 상관성에 의해서 BER 성능의 영향을 많이 받는다. 따라서 터보 부호의 구성요소인 인터리버의 상관성을 최소화 하기 위해서 본 논문에서는 블록 인터리버의 읽는 방식을 다르게 변화시킨 새로운 블록 인터리빙 방법을 제안하였다. 제안한 블록 인터리빙 방법과 기존의 블록 인터리빙 방법의 성능 분석을 위한 모의실험 결과, 구속장과 인터리버의 크기가 커질수록 성능 향상의 폭이 더욱 커짐을 알 수가 있었다.

flattening effect 관점에서는 NTC하에서 각 인터리빙 방법의 성능을 보면 전체적으로 down-left(=dl) 방법의 성능이 가장 우수 하였고 up-right(=ur) 방법과 역 블록 인터리빙 방법인 up-left(=ul) 방법이 그 다음으로 우수한 성능을 보였다. 그리고 기존의 인터리빙 방법인 down-right(=dr) 방법은 성능이 가장 좋지 않은 것으로 나타났다. 따라서 각각의 경우에 새로운 블록 인터리빙 방법은 error floor 현상 즉, flattening effect를 상당히 개선시킬 수 있음을 확인할 수 있었다. 결론적으로, NTC 하에서 블록 인터리버를 읽어내는 원리를 변화시켜 주므로서 인터리버의 상관성을 더욱 최소화시켜줄 수 있었고 효과적으로

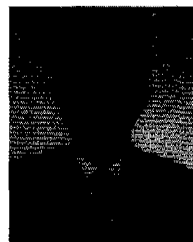
flattening effect가 제거될 수 있음을 확인할 수 있었다.

### 참고 문헌

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding : Turbo Codes," Proc. of the ICC, pp. 1064-1070, 1993.
- [2] P. Jung, M. Nasshan, "Performance Evaluation of Turbo-Codes for Short Frame Transmission Systems," Electronics Letters, vol 30, No 2, pp. 111-113, Jan. 1994.
- [3] P. Jung, "Comparison of Turbo-Code Decoders Applied to Short Frame Transmission System," IEEE Journal on Selected Areas in Comunciations, vol 14, No 3, pp. 530 -537, Apr. 1996.
- [4] Lance C. Perez, Jan Seghers, and Daniel J. Costello, "A Distance Spectrum Interpolation of Turbo Codes," IEEE Trans. on Information Theory, vol. 42, No 6, Nov. 1996.
- [5] Bernard Sklar, "Definning, Designing, and Evaluating Digital Communication Systems" IEEE Communications Magazine, pp. 92-101, Nov. 1993.

정 대 호(Dae-ho Jeong)

정희원



1995년 2월: 원광대학교  
전자공학과 공학사  
1997년 2월: 원광대학교  
전자공학과 공학석사  
1997년 3월~현재: 원광대학교  
전자공학과 박사과정

<주관심 분야> 오류정정부호, 채널코딩, VLSI 구현, CAD & ASIC Design, 회로 및 시스템 설계

김 환 용(Hwan-yong Kim)

정희원

한국통신학회 논문지 제 21권 11호 참조

현재: 원광대학교 전자공학과 교수

<주관심 분야> 채널코딩, CAD & ASIC Design, VLSI 구현, 디지털 신호처리