

# 면적 효율적인 구조의 블록 MAP 터보 복호기 설계

정회원 강문준\*, 김식\*, 황선영\*

## Design of an Area-Efficient Architecture for Block-wise MAP Turbo Decoder

Moon-jun Kang\*, Sik Kim\*, Sun-young Hwang\* *Regular Members*

### 요약

터보 부호의 복호에 사용되는 블록 MAP (Maximum A Posteriori) 복호 알고리즘은 Log-MAP 복호 알고리즘으로부터 메모리 사용량을 감소시킨 알고리즘이다. 기존 블록 MAP 복호기의 BER (Bit Error Rate) 성능은 블록 크기와 트레이닝 크기에 의해 결정되며, 하드웨어의 활용도를 최대로 하고 연속적인 복호를 위해 트레이닝 크기와 같은 블록 크기를 사용한다. 블록 크기와 트레이닝 크기가 복호기의 BER 성능에 미치는 영향을 실험한 결과, 트레이닝 크기가 충분하면 블록 크기가 작아도 BER 성능이 유지됨을 보인다.

본 논문에서는 면적 효율적인 블록 MAP 복호기의 구조를 제안한다. 제안된 복호 방식은 블록 크기의 정수 배인 트레이닝 크기를 사용함으로써 메모리 사용량을 감소시키는 복호 순서를 사용한다. 제안된 복호 순서를 효과적으로 수행하기 위해 파이프라인 구조를 제안하였다. 실험 결과 제안된 복호 방식은 BER 성능을 유지하면서 메모리 사용량이 30~45% 감소하는 것을 확인하였다.

### ABSTRACT

Block-wise MAP (Maximum A Posteriori) decoding algorithm for turbo-codes requires less memory than Log-MAP decoding algorithm. The BER (Bit Error Rate) performance of previous block-wise MAP decoding algorithms depend on the block length and training length. To maximize hardware utilization and perform successive decoding, the block length is set to be equal to the training length in previous MAP decoding algorithms. Simulation result on the BER performance shows that the BER performance can be maintained with shorter blocks when training-length is sufficient.

This paper proposes an architecture for area efficient block-wise MAP decoder. The proposed architecture employs the decoding scheme for reducing memory by using the training length, which is N times larger than block length. To efficiently handle the proposed scheme, a pipelined architecture is proposed. Simulation results show that memory usage can be reduced by 30%~45% in the proposed architecture without degrading the BER performance.

### I. 서론

1948년 Shannon이 통신 채널의 전송률 한계에 관한 연구 결과를 발표한 이래<sup>[1]</sup>, Hamming, BCH, Reed Solomon 코드 등이 개발되었으나, Shannon의 이론에 근접하는 부호는 개발되지 못했다. 1993년

Berrou 등에 의해 제안된 터보 코드는 비교적 간단한 부호화 구조로 Shannon의 한계 이론에 근접하는 오류 정정 성능을 갖는다고 보고되었다<sup>[2]</sup>. 초기 터보 부호는 하드웨어의 복잡성, 복호 시간 지연으로 인해 실시간 처리가 필요하지 않은 장거리 우주 통신용으로 사용되었으나, 실시간 처리를 위한 연구

\* 서강대학교 전자공학과 CAD & Computer Systems 연구실(hwang@ccs.sogang.ac.kr),

논문번호: 020034-0124, 접수일자: 2002년 1월 24일

\* 본 논문 연구는 서강대학교 산업기술연구소의 지원에 의해 수행되었습니다.

개발 결과 차세대 이동 통신 시스템인 IMT-2000 (International Mobile Telecommunication 2000)의 고속 데이터 전송용 채널 부호화 방식으로 채택되었다<sup>[3]</sup>.

터보 부호기는 부호기 입력과 인터리버에 의해 재배열된 입력을 각각 구성 부호기(constituent encoder)를 통해 부호화하여, 입력 정보와 다중 패리티 정보를 출력하도록 구성된다. 독립적인 다중 패리티는 반복 복호를 가능하게 하고 인터리버는 연접오류를 효과적으로 분산시키는 역할을 한다. 터보 코드 복호기는 각 구성 부호기에 대응하는 SISO (Soft Input/Soft Output) 복호기와 경판정 블록으로 구성된다. SISO 복호기의 연판정 출력으로부터 복호기에 의해 추가된 부가 정보(Extrinsic Information)를 계산하여 다른 SISO 복호기의 사전 확률(APP: A Priori Probability) 입력으로 사용하는 과정의 반복으로 복호기 출력의 신뢰도를 증가시킨다<sup>[4]</sup>.

SISO 복호기는 복호 알고리즘에 따라 SOVA (Soft Output Viterbi Algorithm) 방식과 MAP (Maximum A Posteriori) 방식으로 분류된다. MAP 복호기는 SOVA 복호기에 비해 4배 가량 복잡한 구조를 가지고 있으나, 성능은 2배 정도 좋다고 알려져 있다<sup>[5]</sup>. 반도체 공정의 발달과 회로 기술의 발달로 터보 코드의 구현에 있어서 하드웨어 면적보다는 그 BER 성능에 치중하면서 MAP 복호 알고리즘을 사용하는 추세이다. MAP 복호기의 구현에는 연산 복잡도를 줄이기 위해 복호 과정의 데이터를 로그 차원에서 계산하는 Log-MAP 복호 알고리즘이 사용되며, 프레임 크기에 비례하는 메모리 요구량을 감소시키기 위해 입력 프레임을 여러 개의 블록으로 나눠서 복호하는 블록 MAP 알고리즘을 사용한다<sup>[6][7]</sup>.

블록 MAP 알고리즘은 MAP 알고리즘보다 적은 메모리를 요구하나, 역방향 상태 매트릭의 불연속에 따른 BER 성능 감소를 막기 위해 트레이닝 과정(training process)이 필요하다<sup>[6]</sup>. 트레이닝 과정은 각 블록의 역방향 상태 매트릭 초기값을 결정하는 과정으로 블록의 크기와 함께 블록 MAP 알고리즘의 BER 성능에 영향을 준다. 기존의 블록 MAP 복호 알고리즘에서는 하드웨어의 활용도를 높이고 연속적인 복호를 위해 블록 크기와 트레이닝 크기를 같게 하여 사용하고 있다<sup>[7]</sup>. 블록 크기와 트레이닝 크기가 BER 성능에 미치는 영향에 대한 실험 결과, 트레이닝 크기가 충분하면 블록 크기가 BER 성능에 미치는 영향이 미약함을 확인하였다. 본 논

문에서는 실험 결과를 바탕으로 블록 크기가 트레이닝 크기의 1/n이 되는 복호 방법을 제안하고 제안된 방법에 효과적인 복호 구조를 제안한다.

## II. 터보 복호기-MAP복호기

### 1. MAP 복호 알고리즘

AWGN 채널을 통해 시간  $k$  에 MAP 복호기에 입력된 정보 신호를  $x_k$ , 패리티 신호를  $y_k$  라 하면, 송신측 정보 비트  $d_k$  의 LLR(Log Likelihood Ratio)은 Bahl의 복호 알고리즘을 이용해 식 (4)와 같이 정리된다<sup>[9]</sup>.

$$\alpha_k^m = \sum_{j=0}^1 \alpha_{k-1}^{b(j,m)} \delta_{k-1}^{j,b(j,m)} \quad (1)$$

$$\beta_k^m = \sum_{j=0}^1 \beta_{k+1}^{b(j,m)} \delta_k^{j,m} \quad (2)$$

$$\delta_k^{i,m} = \pi_k^i \exp\left(\frac{x_k u_k^i + y_k v_k^i}{\sigma^2}\right) \quad (3)$$

$$L(d_k) = \log \frac{\sum_m \alpha_k^m \delta_k^{0,m} \beta_{k+1}^{1,m}}{\sum_m \alpha_k^m \delta_k^{0,m} \beta_{k+1}^{0,m}} \quad (4)$$

입력 프레임의 각각의 정보로부터 가지 매트릭( $\delta_k^{i,m}$ )을 계산한 후, 순방향 상태 매트릭( $\alpha_k^m$ )과 역방향 상태 매트릭( $\beta_k^m$ )을 계산한다. 마지막으로 이미 계산된 가지 매트릭과 순방향 상태 매트릭, 역방향 상태 매트릭으로부터 MAP 복호기의 출력인  $d_k$  의 LLR 값을 계산한다.

### 2. MAP 복호기의 구현

Log-MAP 복호 알고리즘은 복호에 필요한 데이터들을 로그 차원으로 변환하여 연산 복잡도를 줄였지만<sup>[10]</sup>, LLR 계산에 사용되는 가지 매트릭과 상태 매트릭 계산 결과를 저장하기 위해서는 다량의 메모리를 요구한다<sup>[4][8][11]</sup>. 블록(Block-wise) MAP 복호 알고리즘은 Log-MAP 복호 과정에서 필요한 메모리를 감소시키는 방법으로 MAP복호기를 채용한 터보복호기의 현실적인 SISO 복호 방법으로 부각된 알고리즘이다. 그림 1은 블록 MAP 복호 방법을 나타낸다. 블록 MAP 복호는 길이  $N$ 인 프레임을 길이  $L$ 인  $M$ 개의 블록으로 나누어 블록 단위로 MAP 복호 알고리즘을 적용함으로써 메모리 사용량을 감소시킨다<sup>[6][7]</sup>.

블록 단위로 MAP 복호 알고리즘을 적용하려면,

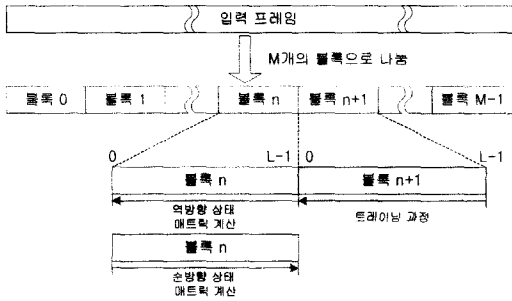


그림 1. 블록 단위 프로세스

순방향 및 역방향 상태 매트릭스를 블록 단위로 계산해야 한다. 역방향 상태 매트릭스의 계산 방향은 블록 진행 방향과 반대 방향이기 때문에 순방향 상태 매트릭스처럼 이전 블록의 최종 상태 매트릭스 값을 이용한 역방향 상태 매트릭스 초기값 계산이 불가능하며, 이는 BER 성능을 저하시키는 요인이 된다. 이 문제를 해결하기 위해 블록 MAP 복호 알고리즘에서는 각 블록의 역방향 상태 매트릭스 초기값을 결정해주는 트레이닝 과정(training process)이 도입되었다. 트레이닝은 역방향 상태 매트릭스 계산을 위해 한 블록 앞서 역방향 상태 매트릭스를 계산하여 신뢰성 높은 역방향 상태 매트릭스 초기값을 결정하는 과정이다. 트레이닝 과정을 포함한 블록 MAP 복호를 연속적으로 수행하기 위해 두 개의 역방향 상태 매트릭스 프로세서를 사용하여 역방향 상태 매트릭스 생성 과정과 트레이닝 과정을 병행하는 구조가 필요하다<sup>[6][7]</sup>.

그림 2는 블록 MAP 복호기의 복호 순서를 나타낸다. 오른쪽 방향의 화살표는 블록 내에서 각 매트릭스 계산 프로세스의 순방향 진행을 나타내며 왼쪽 방향의 화살표는 역방향 진행을 나타낸다. 화살표 안의 수는 처리 과정에서의 블록 번호를 의미한다. 이 그림은 각 블록 시간에 따른 각 상태 매트릭스 프로세서의 동작을 명료하게 나타낸다.

복호가 시작되면 입력 정보로부터 가지 매트릭스를 계산한다. 진행 시간 2에서 첫 번째 블록의 순방향

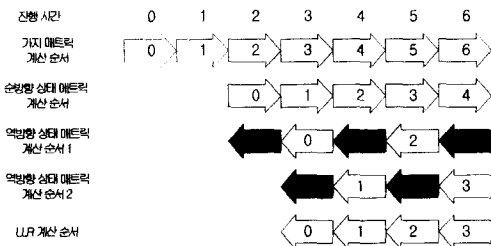


그림 2. 블록 MAP 복호순서

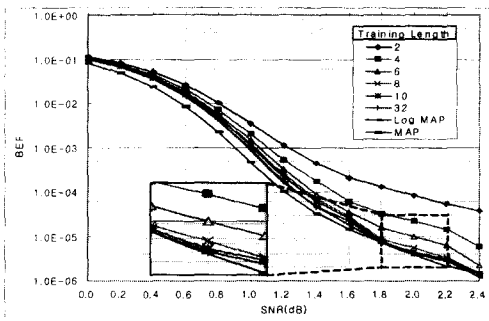
상태 매트릭스를 계산하여 결과를 메모리에 저장하고 같은 시간에 두 번째 블록에서 회색 음영의 화살표가 의미하는 트레이닝 과정을 수행한다. 진행 시간 3에서 트레이닝 과정의 결과를 초기값으로 사용하여 블록 0에 대한 역방향 상태 매트릭스를 계산하고 시간 2에서 저장된 순방향 상태 매트릭스 값과 함께 LLR 값 계산에 사용된다. 이 때 다음 시간에 블록 1에서 계산될 LLR 값 계산에 필요한 순방향 상태 매트릭스를 계산하여 메모리에 저장하고, 블록 2에서는 블록 1의 역방향 매트릭스 초기값을 결정하는 트레이닝 과정을 수행한다. 이 순서로 복호를 할 경우 연속적인 복호를 위해 4개 블록 분량의 가지 매트릭스를 저장할 수 있는 메모리가 필요하다.

블록 MAP 복호는 메모리 사용량이 감소된다는 장점이 있지만, 여러 개의 상태 매트릭스 프로세스가 동시에 수행되어야 하기 때문에 여러 개의 상태 매트릭스 프로세서가 필요하다는 단점이 있다. 이를 해결하는 방법으로 파이프라인 구조의 상태 매트릭스 프로세서가 제안되었다<sup>[7]</sup>.

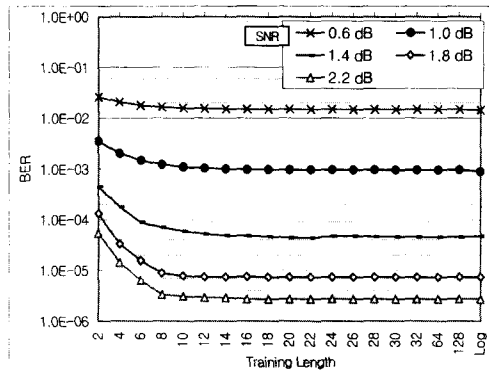
### 3. 블록 MAP 복호기 파라미터

블록 MAP 복호기의 구현에서 블록 크기와 트레이닝 크기는 복호에 필요한 메모리의 양과 BER 성능을 결정한다. 그림 3(a)는 각 트레이닝 크기에서 SNR에 대한 BER 성능의 변화를 나타낸 그래프이다. Log-MAP 복호기는 트레이닝 크기가 프레임의 크기와 같은 경우의 블록 MAP 복호기에 해당하기 때문에, 트레이닝의 크기가 커질수록 Log-MAP의 성능에 근접한다. 반대로 트레이닝 크기가 작아지면 신뢰성이 높은 역방향 상태 매트릭스 초기값을 결정하지 못하기 때문에 BER 성능이 저하된다. 그림 3(b)는 SNR이 각각 0.6, 0.8, 1.4, 1.8, 2.2 dB일 때 트레이닝 크기에 대한 BER 성능의 변화를 보인다. SNR 2.2 dB의 경우 트레이닝 크기가 16 이상, 1.0 dB의 경우 트레이닝 크기가 10 이상이면 BER 성능에 큰 차이를 보이지 않는다. 즉, 트레이닝의 크기가 일정 이상이면 BER의 성능은 큰 차이가 없는 것으로 나타난다. 우측의 Log 항목은 Log-MAP 복호 알고리즘을 사용했을 때의 BER 값들로 블록 MAP 알고리즘의 BER 한계를 나타낸다. 일반적으로 트레이닝 크기는 구속장 길이의 5배면 충분한 것으로 알려져 있다.

트레이닝 과정은 역방향 상태 매트릭스의 초기값을 결정하며, 트레이닝 크기가 작으면 초기값 설정이 불안정하게 된다. 블록이 크기가 클 때는 초기값에



(a)



(b)

그림 3. 블록 단위 트레이닝 크기에 따른 BER 성능 변화  
(a) 각 SNR 별 BER 성능, (b) 각 트레이닝 크기별 BER 성능

오류가 있어도 계산이 진행될수록 올바른 역방향 상태 매트릭 값을 찾아가지만, 그렇지 못할 경우는 매트릭 계산이 불완전한 값에서 종료되기 때문에 BER 성능이 저하된다.

그림 4는 두 트레이닝 크기에서 블록의 크기에 따른 BER 성능의 변화를 보인다. 트레이닝 크기가 4일 때는 블록 크기에 따른 BER 성능의 편차가 크나, 트레이닝 크기가 16일 때는 블록 크기에 따른

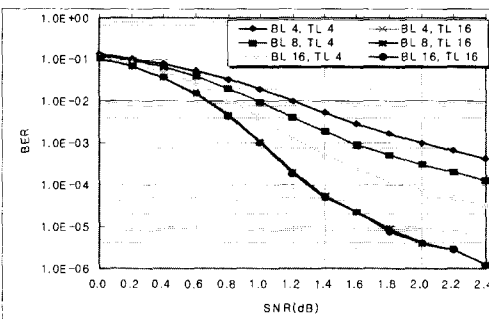


그림 4. 트레이닝 크기 4와 16일 때의 블록 크기에 따른 BER 성능 차이

BER 성능의 편차가 거의 없는 것으로 나타난다. 즉, 트레이닝 크기가 클 때에는 블록 크기의 영향이 거의 없으나, 작을 때는 블록 크기의 영향이 BER 성능에 큰 영향을 준다.

그림 5는 각 트레이닝 크기에 대해서 블록 크기가 4일 때와 블록 크기가 16일 때의 BER 성능 차이를 나타낸다. 트레이닝 크기 32와 16은 두 배의 크기 차이에도 불구하고 두 블록 크기에 대한 BER 성능 차이는 거의 같은 것으로 나타나며, 트레이닝 크기 16 이하에서는 두 블록 크기에 따른 BER 성능 차이가 두드러지게 나타난다. 즉, 트레이닝 크기가 16 이상일 때는 블록 크기가 달라도 BER 성능은 비슷하지만, 16 이하일 때는 큰 BER 성능 차이를 보인다. 또한, SNR이 클수록 블록 크기에 따른 BER 성능 차이가 작은 것으로 나타난다.

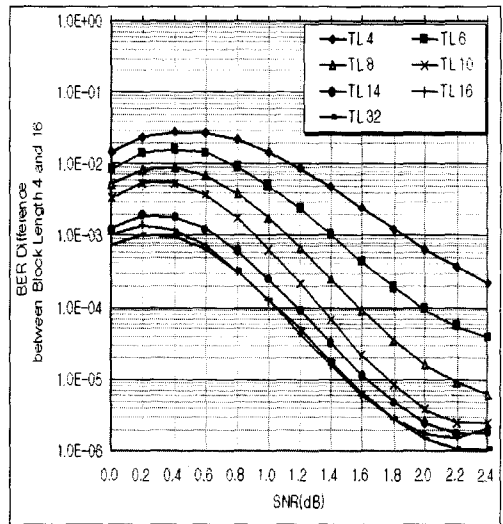


그림 5. 각 트레이닝 크기에 대한 블록 크기 4와 16의 BER 성능 차이

### III. 제안된 블록 MAP 복호기

#### 1. 제안된 블록 MAP 복호 알고리즘

일반적인 블록 MAP 복호기는 블록의 크기와 동일한 트레이닝 크기를 가지고 있다. 그러나, 트레이닝 크기가 일정 이상이면 블록 크기가 작더라도 BER 성능에 큰 영향 없이 복호에 필요한 메모리 양이 감소한다. 이 점에 착안하여 본 연구에서는 블록 크기가 트레이닝 크기의  $1/k$  배인 구조를 제안한다. 그림 6은  $k=3$  인 경우의 제안된 블록 MAP 복호 방법을 보인다. 제안된 복호 방법은 트레이닝 블

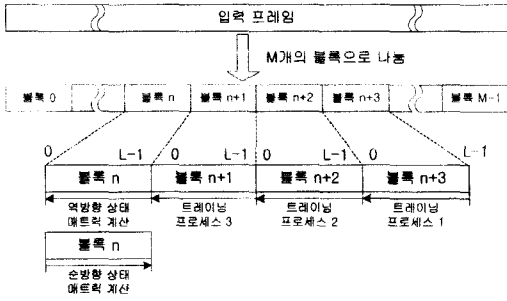


그림 6. 제안된 블록 MAP 복호구조 ( $K = 3$ )

록을  $k$ 개의 블록으로 나누어  $k$ 개의 역방향 상태 매트릭 프로세서를 사용하여 순차적으로 트레이닝 과정을 수행한다. 이 경우, 상태 매트릭 프로세서는 역방향 상태 매트릭 프로세서와 순방향 상태 매트릭 프로세서를 포함하여  $k+2$  개가 필요하다.

그림 7은 기존 블록 MAP 복호기와 제안된 블록 MAP 복호기의 블록 및 트레이닝 크기의 차이를 보인다.  $k=3$  인 제안된 구조는 블록 크기( $L_2$ )가 기존 구조에서의 블록 크기( $L_1$ )에 비해  $1/3$ 이 되나, 3개 블록으로 구성된 트레이닝 과정의 크기를 모두 합하면( $3L_2$ ) 기존 구조의 트레이닝 크기( $L_1$ )와 같다.

그림 8은  $k=3$  인 제안된 블록 MAP 복호기의 각 매트릭과 LLR 출력의 계산 순서를 나타낸다. 블록 MAP 복호기의 상태 매트릭 프로세서는 한 개의 순방향 상태 매트릭 프로세서와 네 개의 역방향 상태 매트릭 프로세서로 구성된다. 역방향 상태 매트릭 값을 계산할 때 세 개의 역방향 상태 매트릭 프로세서는 차후에 계산될 역방향 상태 매트릭의 초기값 계산을 위해 트레이닝 과정을 수행한다. 즉, 세 블록

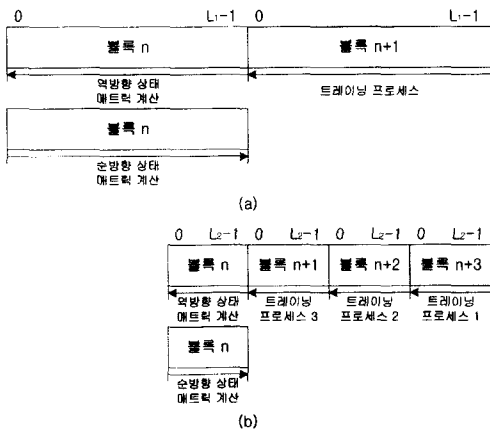


그림 7. 제안된 구조와 기존 구조의 블록 및 트레이닝 크기 차이 ( $K = 3$ ) (a)기존복호기, (b)제안된 복호기

만큼 트레이닝 과정을 거쳐 역방향 상태 매트릭을 계산한다. 제안된 구조의 트레이닝 프로세스 1, 2, 3과 역방향 상태 매트릭 프로세스는 각 프로세스의 최종 상태 매트릭 값을 다음 프로세스의 초기값으로 사용하는 구조로 연속적인 복호를 가능하게 한다.

기존 블록 MAP 복호 방식과 같이 제안된 구조는 순방향 상태 매트릭과 가지 매트릭을 저장하기 위한 메모리를 필요로 한다. 이 때 사용되는 메모리의 양은 블록의 크기에 비례하므로 기존 블록 MAP 방식에 비해 필요한 메모리의 양이 감소한다.

## 2. 제안된 상태 매트릭 프로세서의 파이프라인 구조

제안된 블록 MAP 복호 순서를 효율적으로 수행하려면  $k+2$ 개의 상태 매트릭 프로세스가 병렬 처리되는 구조가 필요하다. 블록 MAP 복호에서 사용되는 상태 매트릭 프로세서의 파이프라인 구조는 하드웨어 공유를 통해 병렬 처리와 같은 효과를 얻을 수 있다<sup>[11]</sup>. 여기서는 기존의 파이프라인 구조를 응용한 상태 매트릭 프로세서를 제안한다.

그림 9는 제안된 상태 매트릭 프로세서의 파이프라인 구조를 나타낸다. 기존 블록 MAP 복호 구조의 주 연산 블록인 ACS 블록이 Trellis MUX 블록이나 정규화 블록에 비해 복잡도가 높은 것을 고려하여,  $k=3$  인 제안된 복호 구조에서는 ACS 블록을 3단으로 나눠 총 5단 파이프라인 구조를 갖는다. 하나의 5단 파이프라인 상태 매트릭 프로세서는 다섯 개의 상태 매트릭 프로세서를 대신하여 각 상태 매트릭을 계산한다.

## IV. 실험 결과

시뮬레이션 시스템은 UNIX 환경의 C언어를 사

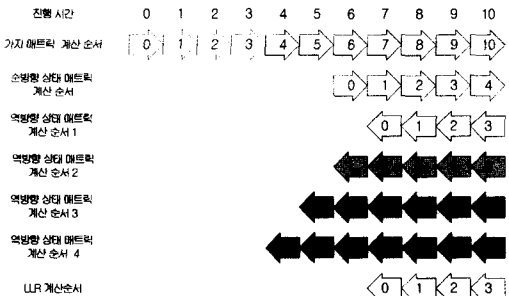


그림 8. 제안된 구조의 복호 순서

용하여 구현하였다. 사용자가 각종 파라미터 및 알고리즘을 선택할 수 있도록 구현하였다. 실험에 사용된 공통된 파라미터는 표 1과 같다. 제안된 블록 MAP 알고리즘의 트레이닝 크기는 24로 했으며, 블록 크기가 6, 8, 12, 24(기존 구조)에 대하여 각각 실험하였다.

1. 메모리 사용량

표 2는 각 터보 복호기의 메모리 사용량을 보인다. 입력과 부가 정보의 저장에 외부 메모리를 사용하고, 각 복호기가 동일한 메모리 양을 필요로 하기 때문에 여기서는 논의하지 않는다. 상태 매트릭과 가지 매트릭은 8비트의 분해능을 갖는 것으로 가정한다. 비교 항목은 블록 MAP 복호기와 제안된 블록 MAP 복호기를 비교한 것이다. 트레이닝 크기가 24이기 때문에 블록 크기 12는  $k=2$ 에 해당하며, 이 경우 상태 매트릭 메모리에서 50%, 가지 매트릭 메모리에서 25%의 사용량 감소로 내부 메모리의 총 사용량은 30% 감소하는 결과를 보인다. 블록 크기 8인 경우( $k=3$ )는 내부 메모리 사용량이 40% 감소하며, 블록 크기 6인 경우( $k=4$ )는 45% 감소하는 결과를 보인다.

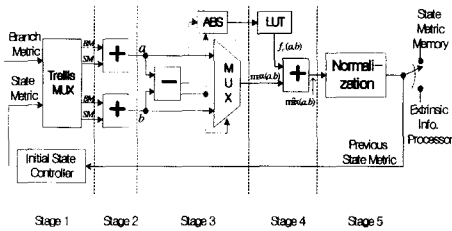


그림 9. 제안된 상태 매트릭 프로세서의 파이프라인 구조

표 1. 실험에 사용된 파라미터

파라미터	값
구속장 ( $k$ )	3
메모리길이 ( $\nu$ )	2
생성다항식 ( $g_1/g_2$ )	7/3
프레임 크기 ( $N$ )	1024
부호율 ( $R$ )	1/3
채널, 변조방식	AWGN, BPSK

2. BER 성능

그림 11은 제안된 블록 MAP 복호기의 BER 성능을 보인다. 블록의 크기가 6, 8, 12인 그래프는

표 2. 터보 복호기의 메모리 사용량

	상태 매트릭 메모리	가지 매트릭 메모리	합계
MAP 복호기	1024x4x8 =32,768	1024x8x8 =65,536	98,304
기존 블록 MAP 복호기	2x4x24x8 =1,536	4x8x24x8 =6,144	7,680
제안된 블록 MAP 복호기 ( $k=2$ )	2x4x12x8 =768	6x8x12x8 =4,608	5,376
	-50%	-25%	-30%
제안된 블록 MAP 복호기 ( $k=3$ )	2x4x8x8 =512	8x8x8x8 =4,096	4,608
	-66.7%	-33.3%	-40%
제안된 블록 MAP 복호기 ( $k=4$ )	2x4x6x8 =384	10x8x6x8 =3,840	4,224
	-75%	-37.5%	-45%

제안된 블록 MAP 복호기의 BER 성능을 나타내며, 24인 그래프는 기존의 블록 MAP 복호기의 BER 성능을 나타낸다. 실험 결과 제안된 방식은 SNR 1.6dB 이하에서는 같은 BER 성능을 보이며, 1.6dB 이상에서도 큰 차이를 보이지 않는다. SNR 1.9dB 부근에서는 기존의 MAP 알고리즘과 같은 BER 성능을 갖기 위한 SNR이 0.01~0.03dB 정도인 것으로 나타난다.

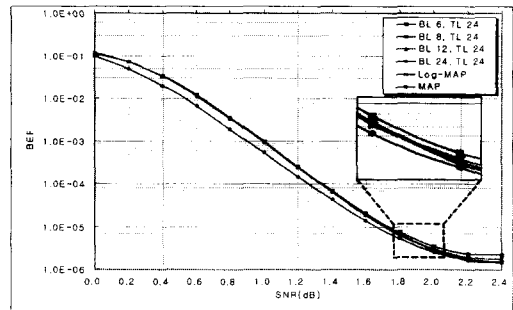


그림 10. 제안된 블록 MAP 복호기의 BER 성능

V. 결론

블록 크기와 트레이닝 크기에 따른 블록 MAP 복호기의 BER 성능을 분석한 결과, 트레이닝 크기가 충분하면 블록 크기가 BER 성능에 미치는 영향은 미미한 것으로 확인되었다. 본 논문에서는 이와 같은 결과를 이용하여 트레이닝 크기가 블록 크기의 정수배가 되는 블록 MAP 복호 알고리즘을 제안하였다. 제안된 블록 MAP 복호 구조는 순방향 상태 매트릭과 역방향 상태 매트릭을 효과적으로



1976 - 1981 : 삼성 반도체 주식회사 연구원, 팀장.

1986 - 1989 : Stanford 대학 Center for Integrated Systems 연구소 책임연구원.

Fairchild Semiconductor Palo Alto Research Center 기술자문.

1989 - 1992 : 삼성전자(주) 반도체 기술자문.

1989년 3월 - 현재 : 서강대학교 전자공학과 교수.

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