

대수적 구조를 가진 2단 연판정 출력 비터비 알고리즘

준회원 김우태*, 정회원 배상재*, 종신회원 주언경*

A Two-Step Soft Output Viterbi Algorithm with Algebraic Structure

Woo-tae Kim* Associate Member, Sang-jae Bae*, Eon-kyeong Joo* Regular Members

요약

본 논문에서는 터보복호기 설계를 위하여 2단 연판정 출력 비터비 알고리즘에 대수적 구조를 적용한 대수적 (algebraic) 2단 연판정 출력 비터비 알고리즘이 제시된다. 제시된 알고리즘은 대수적 구조를 이용함으로써 행렬화된 가지(branch) 및 상태(state) 메트릭의 병렬연산이 가능하다. 따라서 기존의 방식에 비해 곱의 연산량이 감소되며 전체 메모리가 줄어든다. 그러므로 제시된 대수적 2단 연판정 출력 비터비 알고리즘은 적은 계산량과 단순한 하드웨어가 요구되는 터보부호의 복호기에 적합할 것으로 사료된다.

ABSTRACT

A new two-step soft output Viterbi algorithm (SOVA) for turbo decoder is proposed and analyzed in this paper. Due to the algebraic structure of the proposed algorithm, state and branch metrics can be obtained with parallel processing using matrix arithmetic. As a result, the number of multiplications to calculate state metrics of each stage and total memory size can be decreased tremendously. Therefore, it can be expected that the proposed algebraic two-step SOVA is suitable for applications in which low computational complexity and memory size are essential.

I. 서론

최근 오류 성능이 우수하면서 구조가 간단한 통신 시스템에 대한 수요가 증가함에 따라 터보부호 (Turbo code)의 연산량 및 메모리 수 감소 방식에 대한 연구가 활발히 이루어지고 있다. 터보부호는 지금까지의 오류정정부호 가운데 성능이 가장 우수한 것으로 알려져 있다^[1]. 일반적으로 터보부호의 복호 알고리즘은 MAP(maximum a posteriori) 알고리즘과 연판정 출력 비터비 알고리즘(soft output Viterbi algorithm; SOVA)의 크게 두 가지로 나누어진다^{[2],[5]}. MAP 알고리즘은 격자도상의 메트릭 (metric)들을 구하기 위해서 순방향과 역방향 반복 연산을 행하기 때문에 성능 면에서는 우수하지만 구조는 복잡해진다. MAP 알고리즘에 비해서 약간의 성능저하가 있지만 비교적 간단한 구조로 구현

이 가능한 SOVA가 제안되었다. 기존의 Viterbi 알고리즘에서 유도된 SOVA는 순방향 반복연산만이 필요하므로 비교적 구조가 간단하다^{[6],[7]}.

이러한 SOVA와 시스템의 오류성능은 비슷하게 유지하면서 복잡도를 더욱 줄여주는 방식으로 전체 격자도(trellis)를 크게 두 개의 부분으로 나누어서 복호를 행하는 방식인 2단(two-step) SOVA가 제안되었다^{[8]-[11]}. 2단 SOVA는 전체 격자도 중에서 첫 번째 격자도에서 하나의 생존상태를 찾고, 그 찾은 생존상태에 대해서만 나머지 격자도에서 신뢰도값을 갱신시켜 주는 방식이다. 신뢰도값이 모든 상태가 아닌 정해진 하나의 상태에 대해서만 갱신이 되기 때문에 복호 복잡도가 기존의 SOVA에 비해서 감소되는 것으로 알려져 있다.

본 논문에서는 기존의 알고리즘에 비해 복호 복잡도를 더욱 줄여줄 수 있는 새로운 2단 SOVA를

* 경북대학교 전자전기공학부 (ekjoo@ec.knu.ac.kr)

논문번호 : K01174-0801, 접수일자 : 2001년 8월 1일

* 이 논문은 2000년도 학술진흥재단의 지원에 의하여 연구되었음. (KRF-2000-E00208)

제안하고 그 성능을 분석한다. 제안된 알고리즘에는 격자도 상에서의 메트릭들을 행렬의 형태로 바꾸어서 연산하는 대수적 구조(algebraic structure)^{[12],[13]}를 적용한다. 대수적 구조는 행렬의 특징인 결합법칙을 메트릭의 연산에 이용함으로써 기존의 방식에 비해 연산량을 줄여 줄뿐만 아니라 사용되는 메모리의 수도 줄여줄 수 있다. 이러한 대수적 구조를 2단 SOVA의 핵심 부분인 생존메모리블록(survivor memory unit; SMU)에 적용함으로써 기존의 알고리즘과 오류성능은 동일하게 유지하면서 곱셈 연산량과 시스템에 필요한 메모리의 수를 줄여 줄 수 있다. 제안된 대수적 2단 SOVA와 기존의 알고리즘에 대한 곱셈 연산량 그리고 메모리 사용량의 감소를 컴퓨터 모의실험을 통하여 확인하였다.

II. 2단 SOVA

터보복호기에서 복잡도를 줄여주기 위하여 사용되는 SOVA는 Hagenauer^[6]에 의해 처음 제안되었다. 이는 Viterbi 알고리즘을 적용하여 얻는 경관정값(hard decision value)과 이에 대한 신뢰도값을 같이 계산하여 복호하는 방식이다. 이 신뢰도값을 이용해서 복호기에서 반복복호가 가능하다.

2단 SOVA는 SOVA와 오류성능은 거의 비슷하게 유지하면서 복잡도를 많이 줄일 수 있는 방식이다. Berrou^[8]에 의해서 제안되어진 2단 SOVA 복호기는 그림 1에서 보는 바와 같이 크게 격자도 T(D)와 T(U)의 두 개의 블록으로 구성된다. 여기에서 D는 생존길이(survivor depth)이고 U는 신뢰도값 갱신길이(updating depth)이다.

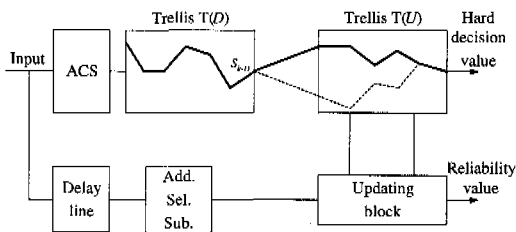


그림 1. 2단 SOVA 복호기 블록도

T(D)블록에서는 덧셈-비교-선택(add-compare-selector; ACS) 블록을 거쳐 계산된 상태메트릭(state metric)값을 이용하여 시간의 생존상태인 S_{k-D} 를 찾는다. 그리고 T(U)블록에서는 결정된 S_{k-D} 에 대해서만 생존경로와 비생존(concurrent)경

로의 경관정값들을 출력한다. 이 값들의 부호를 비교하여 부호가 같은 경우에만 신뢰도값 갱신블록(updating block)에서의 연산을 통하여 갱신된 신뢰도값을 출력한다. 여기에서 사용되는 신뢰도값은 지연블록(delay line)을 거친 메트릭들을 덧셈(Addition), 선택(Selection), 뺄셈(Subtraction) 연산들을 거쳐서 구한다. 격자도상의 모든 상태에 대하여 신뢰도값이 갱신되는 기존의 SOVA에 비해서 연산횟수가 줄어들어 시스템의 복잡도가 줄어든다.

III. 대수적 구조

시스템에 대수적 구조를 적용한다는 것은 격자도상의 상태메트릭과 가지메트릭(branch metric)을 행렬로 나타냄으로서 가능하다. 기존의 방식에서 상태메트릭과 가지메트릭은 이전 시간에서의 각각의 메트릭들을 알아야만 구할 수 있다. 그러나 대수적 구조가 적용된 방식에서는 메트릭들이 행렬로 표현되므로 행렬의 결합법칙을 이용하여 병렬처리가 가능하다. 즉, 메트릭이 모든 시간마다 계산되는 기존의 방식과는 다르게 연산을 한번에 할 수 있다.

$\lambda_{i,k}$ 가 k시간의 j번째 상태에서 k+1시간의 i번째 상태로 천이가 될 때의 가지메트릭이고 $\gamma_{i,k}$ 가 k시간의 i번째 상태의 상태메트릭이면 k+1시간에서의 ACS연산을 통한 상태메트릭은 아래의 식 (1), (2)와 같이 구할 수 있다.

$$\gamma_{1,k+1} = (\lambda_{11,k} \oplus \gamma_{1,k}) \otimes (\lambda_{12,k} \oplus \gamma_{2,k}) \quad (1)$$

$$\gamma_{2,k+1} = (\lambda_{21,k} \oplus \gamma_{1,k}) \otimes (\lambda_{22,k} \oplus \gamma_{2,k}) \quad (2)$$

여기서 \oplus 은 덧셈연산 그리고 \otimes 은 최대선택연산을 나타낸다. 즉, k시간에서의 상태메트릭과 해당 경로의 가지메트릭을 합한 값들 중 큰 값을 k+1시간에서의 상태메트릭으로 정한다.

일반적으로 식 (1), (2)를 식 (3)과 같이 행렬의 형태로 나타낼 수 있다.

$$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}_{k+1} = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix}_k \otimes \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}_k \quad (3)$$

식 (3)을 일반화해서 벡터 형태로 나타내면 식 (4)와 같다.

$$\Gamma_{k+1} = A_k \otimes \Gamma_k \quad (4)$$

그리고 k+2시간에서의 상태메트릭을 식 (5)와 같

이 구할 수 있다.

$$\begin{aligned} \Gamma_{k+2} &= \Lambda_{k+1} \otimes \Gamma_{k+1} = \Lambda_{k+1} \otimes (\Lambda_k \otimes \Gamma_k) \\ &= (\Lambda_{k+1} \otimes \Lambda_k) \otimes \Gamma_k \end{aligned} \quad (5)$$

식 (5)를 보면 $k+2$ 시간의 상태메트릭을 바로 이전 시간인 $k+1$ 의 상태메트릭을 구하지 않고, k 시간의 상태메트릭과 이전 시간의 가지메트릭들만을 이용해서 구할 수 있음을 알 수 있다.

일반적으로 M -스텝의 ACS 연산을 아래 식 (6)과 같이 나타낸다.

$$\Gamma_{k+M} = M \Delta_k \otimes \Gamma_k \quad (6)$$

여기서 $M \Delta_k$ 은 $\Delta_{k+M-1} \otimes \dots \otimes \Delta_k \otimes \Delta_{k+1}$ 연산을 의미한다. 즉, 대수적 구조를 이용함으로써 이전의 모든 시간에 대한 상태메트릭을 구하지 않고도 원하는 시간의 상태메트릭을 구할 수 있다.

IV. 대수적 2단 SOVA

기존의 2단 SOVA 복호기의 구조는 가지메트릭 블록(branch metric unit; BMU), ACS블록(ACS unit; ACSU), SMU, 그리고 신뢰도값 갱신블록들로 나누어진다. 본 논문에서는 대수적 구조를 2단 SOVA의 핵심 부분인 SMU에 적용하였다.

그림 2는 SMU에 대수적 구조를 적용한 대수적 2단 SOVA 복호기의 블록도이다. 직·병렬 변환기(S/P)를 거친 입력신호를 이용하여 BMU에서 가지메트릭이 구해진다. ACSU에서는 생존상태를 행렬로 나타낸 값과 신뢰도값, 그리고 k 시간의 생존상태인 S_k 를 구한다. 이런 값들을 RAM에 저장한다. SMU에서는 ACSU에서 계산된 각 상태메트릭의 행렬값(Δ)을 이용하여 생존경로 비트와 비생존경로 비트값을 구하여 신뢰도값 갱신블록으로 넘겨준다. 신뢰도값 갱신블록에서 최종 갱신이 된 값을 출력한다.

그림 3은 2단 SOVA-SMU에 적용이 되는 대수적 구조를 설명하기 위한 상태의 수가 4개인 격자도의 예이다.

그림의 화살표는 이전 시간에서의 천이가 어떤 상태에서 이루어졌는지를 가리킨다. S_k^i 는 k 시간에서 i 번째 생존상태를 나타낸다. k 시간에서 이전 시간 $k-1$ 의 생존상태를 나타내는 값은 식 (7)과 같다.

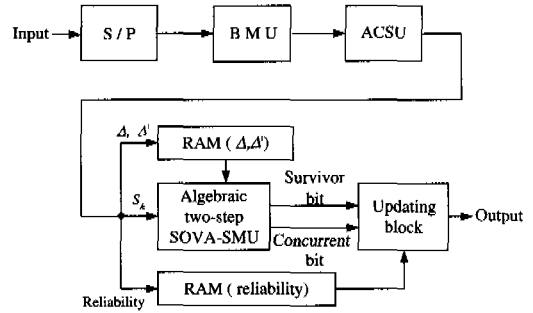


그림 2. 대수적 2단 SOVA 시스템 블록도

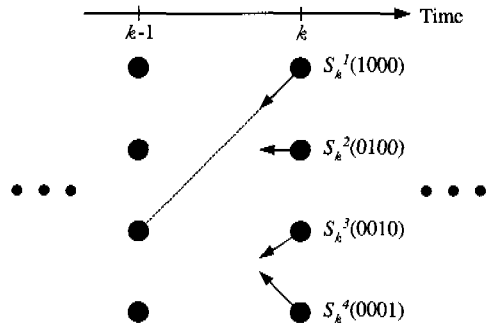


그림 3. 대수적 구조의 예제 격자도

$$\delta_k(1)=3, \delta_k(2)=2, \delta_k(3)=4, \delta_k(4)=2 \quad (7)$$

여기서 $\delta_k(i)=j$ 는 k 시간의 i 번째 상태인 S_k^i 가 $k-1$ 시간에서는 j 번째 상태인 S_{k-1}^j 로부터 천이 되었음을 나타낸다. 이와 같은 값들을 행렬의 형태로 표현하면 j 번째 상태만을 1로 나타내고 나머지 상태값들은 모두 0으로 표현한다. 이와 같이 식 (7)을 행렬로 나타내면 식 (8)와 같다.

$$\Delta_k = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (8)$$

만일 k 시간에서의 생존상태인 S_k^1 이 1, 즉 (1000)이라면 이전시간 $k-1$ 에서의 생존상태인 S_{k-1} 은 다음과 같이 구해진다.

$$\begin{aligned} S_{k-1} &= S_k^1 \cdot \Delta_k \\ &= (1000) \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = (0010) \end{aligned} \quad (9)$$

따라서 k 시간에서 생존상태 $S_k^{(1000)}$ 에 대한 $k-1$ 시간에서의 생존상태는 세 번째 상태인 $S_{k-1}^{(0010)}$, 즉 (0010)임을 알 수 있다. 이와 같은 과정을 되풀이함으로써 일정시간 D 이전의 생존상태인 S_{k-D} 를 구할 수 있다. S_{k-D} 는 아래의 식과 같이 정리된다.

$$S_{k-D} = S_k \cdot \Delta_k \cdots \Delta_{k-D+1} \tag{10}$$

식 (10)에 나타난 바와 같이 이전시간 $k-D$ 의 생존상태인 S_{k-D} 는 행렬 $\Delta_i (i = k, k-1, \dots, k-D+1)$ 와 k 시간의 생존상태인 S_k 의 곱으로 구할 수 있다.

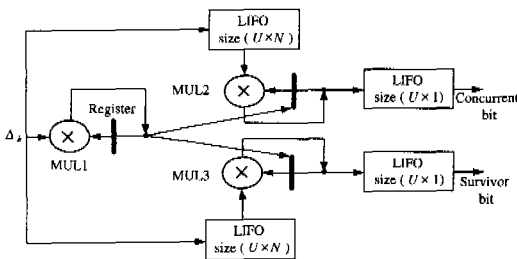


그림 4. 대수적 2단 SOVA-SMU 블록도

이와 같은 대수적 구조가 적용된 2단 SOVA-SMU를 좀 더 자세히 나타내면 그림 4와 같다. 여기서 사용된 RAM의 구조는 LIFO(last in first out) 구조를 사용하였다. 그리고 U 는 신뢰도값 갱신길이, N 은 상태의 수를 나타낸다.

첫 번째 궤환곱셈블록(MUL1)에서는 대수적 구조의 특징인 결합법칙을 이용하여 생존상태 S_{k-D} 를 찾는다. 즉 행렬의 형태로 바뀌어진 매트릭들(Δ_k)을 먼저 구한 후 기준 시간에서의 벡터값을 곱해주어서 원하는 시간에서의 생존상태를 찾는다.

두 번째와 세 번째 곱셈블록(MUL2, MUL3) 역시 RAM에 저장된 행렬 값을 이용하여 U 시간의 생존 비트와 비생존 비트를 찾는다. 신뢰도값 갱신블록에서는 SMU에서 찾은 각 생존 비트와 비생존 비트들과 ACS블록에서 계산되어 램에 저장되어 있는 신뢰도값을 이용하여 갱신된 새로운 신뢰도값을 출력한다. 이러한 연산을 수행하는 과정을 그림 5의 신뢰도값 갱신블록에 나타내었다.

2단 SOVA-SMU에서 계산된 생존 비트와 비생존 비트 값들의 부호를 비교하여 부호가 다른 경우에만 신뢰도값을 갱신한다. 즉 신뢰도값과 기준이 되는 신뢰도값 중 큰 값을 부호가 다른 경우에만 갱

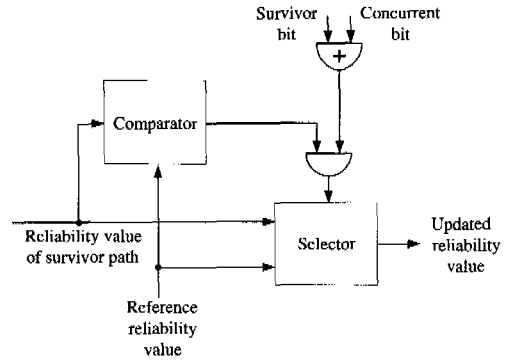


그림 5. 신뢰도값 갱신블록도

신을 한다. 부호가 같은 경우에는 기준이 되는 신뢰도값과는 관계없이 그 상태가 가진 신뢰도값을 그대로 출력하면 된다. 결국 2단 SOVA 복호기의 최종 출력값은 생존 비트와 최종 출력된 신뢰도값을 곱함으로써 구할 수 있다.

V. 성능 및 연산량 분석

1. 성능 분석

제안된 2단 SOVA의 성능을 분석하기 위해 터보 복호기의 구성복호기로 대수적 2단 SOVA 복호기를 사용하였다. 본 논문에서 제안된 대수적 2단 SOVA는 기존의 방식을 알고리즘의 변화가 아닌 연산 방식의 변화를 이용하기 때문에 두 가지 방식의 오류성능은 동일하다. 일반적인 터보복호기와 같이 부가정보(extrinsic information)를 이용하여 반복 복호를 수행하여 원하는 성능을 얻을 수 있다.

그림 6은 반복복호 횟수에 따른 대수적 2단 SOVA 복호기의 성능을 AWGN 채널에서 모의실험을 행한 결과이다. 여기서 사용된 터보부호의 구성 부호기로는 가장 일반적으로 사용되는 3개의 메모리와 15₍₈₎, 13₍₈₎인 생성다항식을 이용했다. 인터리버는 성능이 가장 우수하다고 알려진 s-랜덤 인터리버^[14]가 사용되었다. 그리고 한 프레임의 길이는 200인 것을 사용하였고 부호율은 1/3로 하였다. 사용되는 인터리버의 크기와 종류 그리고 반복복호 횟수에 따라서 성능이 달라지므로 원하는 성능에 적합한 인터리버와 반복복호를 사용할 수 있다.

그러나 그림 6에 나타난 듯이 반복복호 횟수가 4에서 5회 이상이면 성능 향상이 거의 없음을 알 수 있다. 따라서 실제 시스템의 구현에는 반복복호를 4회 또는 5회 정도로 하는 것이 적절하리라 사료된다.

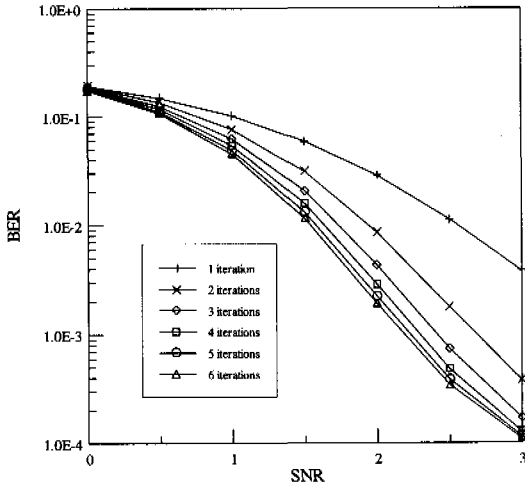


그림 6. 2단 SOVA의 성능

2. 연산량 분석

표 1은 상태의 수가 N , 생존경로의 길이가 D , 신뢰도값의 갱신 길이 U 에 대한 기존의 2단 SOVA와 대수적 구조가 적용된 2단 SOVA의 SMU구조를 메모리의 수와 곱셈 연산량의 관점에서 비교한 것이다.

표 1. 2단 SOVA-SMU의 복잡도 비교

Structure	Number of Multiplications	Total memory size	Register	RAM
Conventional 2-step SOVA-SMU	$DN+2UN$	$DN+2UN$	$DN+2UN$	-
Algebraic 2-step SOVA-SMU	$2(N+1)$	$2UN+2U$	-	$2UN+2U$

기존의 방식은 길이가 $D+U$ 인 생존경로와 길이가 U 인 비생존 경로의 모든 상태와 시간마다 곱셈 연산을 수행해야 하기 때문에 $DN+2UN$ 의 곱셈연산수가 필요하다. 그러나 제안된 알고리즘에서는 생존경로와 비생존 경로에 대해서 일정시간 만큼의 연산을 한번에 할 수 있으므로 상태 수만 고려해주는 $2(N+1)$ 번의 곱셈연산만이 필요하다. 예를 들면 부호기 메모리가 3일 때 N 은 8이 되고 일반적으로 사용되는 D 가 20 그리고 U 가 10인 경우를 보면 기존의 방식은 곱셈연산수가 320인 반면 제안된 구조는 18로 약 5% 정도로 줄어든다. 그리고 부호기 메모리의 개수가 증가함에 따라 곱셈 연산량의 감

소폭이 더욱 더 커짐을 알 수 있다. 즉, 부호기 메모리의 개수가 4인 경우, 기존의 방식은 816으로 3인 경우에 비해 약 500이 증가한 반면 제안된 알고리즘에서는 34로 16밖에 증가하지 않았다.

또한 $DN+2UN$ 의 메모리 수가 필요한 기존의 방식과는 다르게 제안된 알고리즘의 전체 메모리의 수는 그림 4에 나타난 바와 같이 $2UN+2U$ 이던 된다. 예를 들면 실제로 부호기 메모리의 수가 3인 경우에 기존의 방식은 320인 반면 제안된 방식은 180으로 감소됨을 알 수 있다. 그리고 부호기 메모리의 수가 4인 경우는 기존의 방식은 816 그리고 제안된 방식은 442임을 알 수 있다. 따라서 기존의 방식에 비하여 부호기 메모리의 수에 관계없이 약 50%정도만으로 충분함을 알 수 있다.

또한 제안된 알고리즘의 또 다른 특징으로 기존의 2단 SOVA는 메모리로 레지스터를 사용하는 반면 제안된 대수적 2단 SOVA는 모두 RAM을 사용한다. 따라서 행렬의 계산 결과가 모두 RAM에 저장되므로 레지스터를 사용할 때 발생하는 레지스터 사이의 천이로 인한 전력소모를 줄일 수 있을 것으로 사료된다.

따라서 예를 들어 IMT-2000^[15]에서 사용되는 메모리 수가 3이고 부호율이 1/3인 터보부호의 복호 알고리즘으로 위에서 제안된 대수적 2단 SOVA를 사용하면 곱셈 연산량은 약 95%, 그리고 메모리는 약 50%의 감소 효과를 얻을 수 있을 것으로 사료된다.

VI. 결론

2단 SOVA는 기존의 알고리즘과는 다르게 격자도를 두 부분으로 나누어서 첫 번째 부분에서는 하나의 생존상태만을 찾고 두 번째 부분에서는 이렇게 정해진 하나의 생존상태에 대해서만 신뢰도값을 갱신해 주는 방식이다. 따라서 기존의 SOVA에 비해 복호 복잡도가 감소되는 것으로 알려져 왔다.

본 논문에서는 기존의 2단 SOVA의 복잡도를 더욱 감소시키기 위해 기존의 2단 SOVA에 대수적 구조를 적용한 대수적 2단 SOVA를 제안하였다. 제안된 대수적 2단 SOVA는 매트릭을 대수적 구조를 사용함으로써 곱셈의 결합 법칙을 이용할 수 있다. 따라서 최종 매트릭을 얻기 위한 곱셈을 순차적으로 해야만 하는 기존의 방식에 비해 행렬의 곱을 먼저 수행한 후 원하는 시간의 매트릭을 찾을 수 있으므로 시스템의 복잡도를 줄여줄 수 있다.

모의 실험결과 메모리의 수가 3일 때 곱셈의 연산량은 약 95%정도가 줄어들고 메모리의 수도 기존의 방식에 비해 약 50%정도 감소함을 확인할 수 있었다. 따라서 제안된 대수적 2단 SOVA는 단순하고 연산량이 적게 요구되는 통신 시스템에 적합할 것으로 사료된다.

References

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261-1271, Oct. 1996.

[2] P. Jung, "Comparison of turbo-code decoders applied to short frame transmission systems," *IEEE J. Select. Areas Commun.*, vol. 14, no. 3, pp. 530-537, Apr. 1996.

[3] J. Hagenauer and P. Robertson, "Iterative ("Turbo") decoding of systematic convolutional codes with the MAP and SOVA algorithms," *ITG-Fachberichte*, Frankfurt, German, vol. 130, pp. 21-29, 1995.

[4] M. P. C. Fossorier, F. Burkert, S. Lin, and J. Hagenauer, "On the equivalence between SOVA and Max-Log-MAP decodings," *IEEE Commun. Lett.*, vol. 2, no. 5, pp. 137-139, May 1998.

[5] P. Robertson, E. Villebrun, and P. Hoher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," *Proc. IEEE ICC'95*, Seattle, WA, pp. 1009-1013, June 1995.

[6] J. Hagenauer and P. Hoher, "A Viterbi algorithm with soft-decision outputs and its applications," *Proc. IEEE Globecom'89*, Dallas, TX, pp. 1680-1686, Nov. 1989.

[7] G. Battail, "Ponderation des symboles decodes par l'algorithm de Viterbi," *Amm. Telecommun.*, France, pp. 31-38, Jan. 1987. (in French)

[8] C. Berrou, P. Adde, E. Angui, and S. Faudeil, "Low complexity soft-output Viterbi decoder architecture," *Proc. IEEE ICC'93*, Geneva, Switzerland, pp. 737-740, May 1993.

[9] O. J. Joeressen, M. Vaupel, and H. Meyr, "High speed VLSI architectures for soft output

Viterbi decoding," *Proc. IEEE ICASAP'92*, Oakland, CA, pp. 373-384, Aug. 1992.

[10] L. Lin and R. S. Cheng, "Improvements in SOVA-based decoding for turbo codes," *Proc. IEEE ICC'97*, Montreal, Canada, pp. 1473-1478, June 1997.

[11] O. J. Joeressen, M. Vaupel, and H. Meyr, "Soft-output Viterbi decoding: VLSI implementation issues," *Proc. IEEE VTC'93*, New York, NY, pp. 941-944, May 1993.

[12] G. Fettweis and H. Meyr, "High-speed parallel Viterbi decoding," *IEEE Commun. Mag.*, vol. 29, no. 5, pp. 46-55, May 1991.

[13] G. Fettweis, "Algebraic survivor memory management design for Viterbi detectors," *IEEE Trans. Commun.*, vol. 43, no. 9, pp. 2458-2463, Sep. 1995.

[14] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," TDA Progress Report 42-122, JPL, Pasadena, CA, pp. 56-65, Aug. 1995.

[15] Telecommunications Technology Association, The cdma2000 ITU-R Candidate Submission, USA, 1998.

김 우 태(Woo-tae Kim)

준회원



1998년 2월 : 경북대학교
전자공학과 졸업
2000년 2월 : 경북대학교
전자공학과 석사
2000년 3월~현재 : 경북대학교
전자공학과 박사과정

<주관심 분야> 오류정정부호, 시공간부호, 이동통신

배 상 재(Sang-jae Bae)

정회원



1993년 2월 : 경북대학교
전자공학과 졸업
1995년 2월 : 경북대학교
전자공학과 석사
1995년 3월~현재 : 경북대학교
전자공학과 박사과정

<주관심 분야> 터보부호, 다중레벨 부호변조, 이동통신, WCDMA

주 언 경(Eon-kyeong Joo) 종신회원



1976년 : 서울대학교 공과대학
전자공학과 졸업(공학사)

1984년 : 미국 Ohio State
University 전기공학과
졸업(M.S.)

1987년 : 미국 Ohio State
University 전기공학과
졸업(Ph.D.)

1976년~1979년 : 해군통신전자 기술장교

1979년~1982년 : 한국과학기술원 연구원

1987년~현재 : 경북대학교 공과대학 전자전기공학부
재직(교수)

<주관심 분야> : Digital Communication Systems,
Coding and Decoding, Modulation and
Demodulation, Digital Signal Processing
for Communications