

논문 2014-51-7-27

고속 데이터 전송을 위한 변형 해밍망 설계

(Design of modified HN for High Data Transmission)

권 용 광*

(Yong-kwang Kwon[©])

요 약

비터비는 상관성이 없는 노이즈 환경 하에서 이산 유한 상태 기계(FSM)의 상태 변이를 추정하는데 사용되는 알고리즘이다. 본 논문에서는 FSM의 상태를 추정하기 위해 해밍 네트워크를 변형하여 상태 병렬 및 블록 병렬 처리 비터비 복호기를 제안한다. 제안된 mHN(modified Hamming Network)는 길쌈 부호를 복호함으로써 기존의 비터비 복호기와 같은 동작을 수행한다. 제안된 비터비 복호기는 기존의 비터비 복호기보다 약 10% 낮은 복잡도를 제공하며, 40%의 수행시간을 단축시킬 수 있다.

Abstract

The Viterbi algorithm(VA) is used to estimate the state transition of discrete-time finite state machine(FSM) that is in an uncorrelated noisy environment. This paper modified the Hamming Network to estimate the state transitions in the finite state machines, and proposed state-parallel and block-parallel Viterbi decoder. The modified Hamming Network(mHN) can perform the decoding of convolutional codes correctly as conventional Viterbi decoder. Furthermore, the complexities of the proposed Viterbi decoder are reduced approximately 10% less than conventional Viterbi decoder, and the processing times are improved approximately 40% more than conventional Viterbi decoder.

Keywords : Viterbi, FSM, Hamming Network

I. 서 론

휴대용 모바일 기기가 대중화 되고 통신 환경이 보다 유연해지면서 초고속 인터넷 이용자는 매분기 급격하게 증가하고 있으며, 이러한 현상은 사용자들의 욕구가 증가하면서 고속 대용량 전송에 대한 요구를 급격하게 상승 시키고 있다. 따라서 대용량화되면서 효율적이고 신뢰성 있는 디지털 데이터 전송 시스템의 필요성은 계속적으로 대두되고 있다. 따라서 고속 디지털 전송 시스템 설계 시 전송 채널에서 발생하는 오류의 제어와 복

원에 대한 관심은 지속될 수밖에 없다^[1~2]. 이러한 요구에 따라 고속 디지털 전송 시스템에서 요구되는 신뢰도를 실현하고자 하는 방향으로 개발이 이루어지고 있으며 오류 제어를 위한 오류 정정 부호의 이용은 필수가 되었다.

비터비 알고리즘(Viterbi algorithm)은 잡음 환경 하에서 채널의 상태 변이 과정을 추정하는 방법으로^[3], 오류 정정 부호 기법으로써 널리 사용된다. 비터비 알고리즘에서는 변이 가능 상태 열은 트렐리스 도의 경로에 의해 표현되므로, 선택되는 최단 경로의 복잡도는 절점이나 경로의 수에 비례한다. 결과적으로 비터비를 이용한 복호기의 동작 속도는 복잡도가 증가함에 따라 근본적으로 제한되며, 또한 길쌈 부호의 구속장(constraint length)이 증가하면 기억 용량 및 계산량이 증가하므로 구현이 용이하지 못하다. 따라서, 길쌈 부호의 구속장이

* 정회원, 신안산대학교 컴퓨터정보과
(Department of Computer Information, Shin-Ansan University)

© Corresponding Author(E-mail: fifthave@sau.ac.kr)

접수일자: 2014년05월09일, 수정일자: 2014년05월30일
수정완료: 2014년06월25일

증가하여도 구현에 용이하도록 복잡도를 감소시키면서 최적 성능에 근접하는 복호기의 구현이 필요하다.

일반적으로 비터비 디코더에서 잡음 채널을 통해 데이터를 전송하기 위한 방법에는 하드 결정(Hard Decision), 양자화 결정(Quantization Decision), 소프트 결정의 방법(Soft Decision)이 사용되는데^[4], 하드웨어의 처리 속도를 향상시키거나, 기억 용량 및 계산량과 같은 하드웨어의 복잡도를 감소시킴으로써 복호기 성능을 향상시키는 방법이 사용된다^[5]. 전자에서는 병렬 처리를 이용하여 모든 상태나 여러 단을 동시에 처리하는 방법을 사용하며 계산 구조의 변환 방법이 중요하다. 그러나 속도를 개선하려면 상태 병렬 비터비 복호기가 트렐리스도의 여러 단을 동시에 처리할 수 있어야 하는데, 여러 단에서 경로 선택이 동시에 가능한 알고리즘의 하드웨어 복잡도는 복호 속도와 상태 수에 비례하여 커지므로 실현의 한계가 나타난다.

본 연구에서는 대규모 병렬성을 지니고 있는 신경회로망(neural network)을 이용한 방법에 관심을 갖는다. 신경회로망 소자는 디지털 소자보다 처리 속도는 다소 느리나 구조가 단순하며 소자의 병렬 연결은 계산의 효율성, 강인성 및 학습 능력을 지니고 있으므로 비선형 시스템에 응용이 용이하다^[6]. 따라서 대량의 데이터를 병렬로 처리할 수 있는 신경회로망을 이용하여 복호기를 구성한다면 설계를 단순화하면서 속도가 개선되는 결과를 얻을 수 있다.

신경회로망의 하나인 해밍 네트워크(Hamming network, HN)는 해밍 거리(Hamming Distance, HD)를 계산하여 기준 패턴에 가장 가까운 입력을 분류한다^[7]. 유한 상태 기계의 상태 변이 추정은 트렐리스 도에서 최소 해밍 거리와 같은 의미이므로 HN를 응용하면 유한 상태 기계의 상태 추정이 가능하다. 길쌈 부호는 유한 상태 기계와 같은 개념이므로, 오류로 인해 전송 시스템의 부호어가 달라질 때, HN를 이용하여 유효 상태 변이를 추정하여 송신 데이터 복원에 응용할 수 있다.

본 논문은 디지털 전송 시스템에 적용하는 길쌈 부호의 복호 성능을 개선하기 위하여 HN에 기초한 길쌈 부호의 병렬 비터비 복호기를 설계한다. 제안한 방법의 유용성은 길쌈 부호화된 정보를 AWGN를 갖는 채널로 통과시킨 후 복호하는 시뮬레이션을 이용하여 기존 비터비 복호기와 비교한다.

논문 구성은 다음과 같다. II장은 길쌈 부호와 비터

비에 대하여 설명하고, III장에서 신경회로망과 HN를 소개한다. IV장은 변형 해밍 네트워크를 제안하고, 구현된 복호기의 성능을 검증한다. VI장에서 결론을 맺는다.

II. 본 론

1. 길쌈 부호 (Convolutional Code)

오류 정정 부호는 메모리의 존재 유무에 따라 블록 부호(block code)와 길쌈 부호로 분류되는데, 길쌈 부호의 출력은 과거 입력의 영향도 받게 되어 블록 부호에 비해 오류 정정 능력이 우수하다^[1~2].

길쌈 부호는 출력 단자 수 n , 입력 단자 수 k 그리고 기억소자 단수 m 이 사용되며, 일반적으로 정수 k 와 n 은 $n > k$ 인 관계가 있다. 길쌈 부호는 블록 부호에 비해 다소 복잡한 구조를 가지나 오류 정정 능력은 우수하며, 한 개의 정보 블록이 직접 영향을 줄 수 있는 블록 수는 $m+1$ 개이며 길쌈 부호의 구속장은 $n_A = (m+1)n$ 으로 정의된다. 부호율은 $R = k/n$ 로 표시된다. 구속장이 n_A 인 길쌈 부호는 부호기가 m 단의 지연 소자로 구성되면 정보 계열을 k 비트씩 블록으로 분리하여 각 블록의 k 비트는 직병렬 변환기에 의하여 병렬화되어 임의 시점 t 에서 m 단의 지연 소자에 입력된다. 다음 지연 소자의 출력 및 직병렬 변환기의 출력이 modulo-2 가산기에 입력되며 n 개의 출력은 병직렬 변환기에 의하여 직렬화되어 부호 계열로서 전송된다. 이 부호 계열을 (n, k, m) 길쌈 부호라 한다. 그림 1은 $k=1, n=2$ 및 $m=2$ 인 2진 전방향 (2,1,2) 길쌈 부호기를 도식한다.

길쌈 부호는 메모리를 포함하고 있어서 한 시점의 부호기 출력은 그 시점의 입력 뿐 만 아니라 이전 시점의 입력에 영향을 받는다. 메모리 크기 m 인 $R = k/n$ 율의 길쌈 부호기는 입력 메모리 m 인 k -입력, n -출력 선형 순차 회로로 구현할 수 있다. 길쌈 부호는 블록 부호와 달리 k 및 n 을 증가시키지 않고 메모리 크기 m 을 증가시킴으로써 최소 거리를 증가시키고 오류 확률을 감

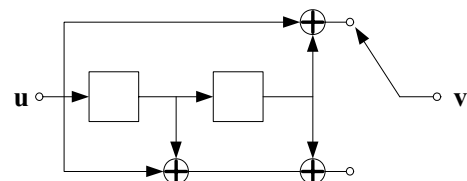


그림 1. 전방향 (2,1,2) 길쌈 부호기
Fig. 1. FeedForward (2,1,2) Convolutional Encoder.

소시킬 수 있다.

2. 비터비 알고리즘 (Viterbi Algorithm)

비터비 알고리즘은 트렐리스도 상에서 최단 경로를 반복적으로 탐색하는 방법으로 잡음 환경에서 이산 시간 유한 상태 기계(finite state machine)의 상태 변이 과정을 반복적으로 추정하는 최적 방법이다. 이 알고리즘은 최단 경로 결정을 위해 최소 HD를 탐색하는 복호법이기 때문에 가능성이 희박한 경로는 포기하면서 탐색을 될 수 있는 대로 간단하게 이행하는 방법이다. 그러나 오류 확률은 구속장 n_A 가 증가함에 따라 지수 함수적으로 증대하므로 비터비 복호는 $n_A > 8$ 인 긴 부호에 대해서는 비실용적이다. 즉 부호의 구속장이 증가될수록 계산량과 지연 시간이 증가하기 때문에 실용적으로는 구속장이 짧은 부호에만 적용되는 단점이 있다.

III. 해밍 네트워크

고정 길이의 2진 부호어가 무기억 2진 대칭 채널(memoryless binary symmetric channel)을 통해 전송되는 경우 수신단에서 이를 복호하는 문제는, 신경회로망에서 기준 패턴을 선택한 후 주어진 확률로 불규칙하고 독립적으로 비트를 반전된 수신 신호에 대하여 그 수신 신호에 가장 가까운 기준 패턴을 찾는 문제와 같다고 볼 수 있다. 이 경우 최적 최소 오류 분류기(optimum minimum error classifier)는 각 클래스에 대한 기준 패턴과의 해밍 거리를 계산한 후 해밍 거리가 최소인 클래스를 선택하는 것이다. 이 알고리즘을 신경

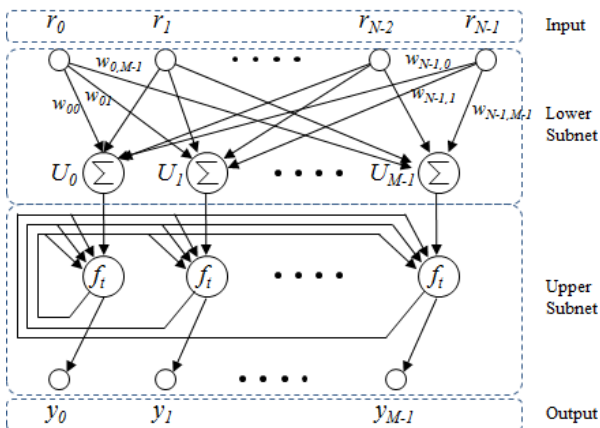


그림 2. 해밍 네트워크
Fig. 2. Hamming Network.

회로망 구성으로 실현한 것이 그림 2의 HN이다^[5].

HN 하부망은 퍼셉트론(perceptron)으로 하중(weight)과 임계치(threshold)는 중간 절점(node)들의 출력으로 나타나는 매칭 점수(matching score, 이하 MS)가 입력 요소의 개수 N 에서 기준 패턴과의 해밍 거리를 뺀 값과 같도록 조정한다(식 1). MS는 $0 \sim N$ 범위로 입력에 가장 일치되는 기준 패턴의 클래스에 해당하는 절점이 가장 큰 값을 갖는다.

$$U_j = N - N_{dj} = N_{aj} \quad (1)$$

식에서 N_{dj} 는 \mathbf{r} 와 \mathbf{c}^j 사이에 일치하지 않는 비트수, N_{aj} 는 \mathbf{r} 와 \mathbf{c}^j 사이에 일치하는 비트수이다. 기준 패턴 수를 M 이라 할 때, 연결 하중 w_{ij} 와 임계치 θ_j 의 학습은 식(2)와 같다.

$$w_{ij} = \frac{c_i^j}{2}, \theta_j = \frac{N}{2} \quad \begin{cases} i=0, \dots, N-1 \\ j=0, \dots, M-1 \end{cases} \quad (2)$$

그림 2로부터 식(2)를 대입한 U_j 는 식(3)과 같다.

$$\begin{aligned} U_j &= \sum_{i=0}^{N-1} (w_{ij} r_i) + \theta_j \\ &= \sum_{i=0}^{N-1} \left(\frac{c_i^j}{2} r_i \right) + \frac{N}{2} = (N - N_{dj}) = N_{aj} \end{aligned} \quad (3)$$

상부망의 임계치와 하중은 고정되어있다. 모든 임계치는 0으로 설정되고 각 절점 자체로의 하중은 1이며, 그 외 서로 다른 절점 간 하중은 $-e$ 로 금지성(inhibitory)이다. 여기에서 $e < 1/M$ 이다.

하중과 임계치가 설정된 후 N 개 요소로 구성된 2진 패턴이 해밍 네트워크의 하단에 입력된다. 이 입력은 하부망의 MS 출력이 안정되고 상부망의 출력을 초기화할 수 있을 만큼 충분히 긴 시간 동안 유지되어야 한다. 다음 입력이 제거되고 상부망은 한 노드의 출력만이 양(positive)으로 될 때까지 순환(iterate)한다. 이와 같이 하여 분류(classification)가 완료되고 양(+)의 출력의 노드가 선택된 클래스가 된다.

IV. 제안 방법

비터비 복호기에 HN를 적용하려면 HN의 하부망은 (N-HD) 대신 HD 자체로 MS를 계산하고, 상부망은 최

대치가 아니라 최소치를 갖는 노드만이 승자가 되도록 순환해야 한다. 이 모델을 변형 해밍 네트워크(Modified Hamming network: 이하 mHN)라 한다.

1. 하부망과 상부망

하부망에서 HD 자체로 MS를 계산하고 상부망에서 최소치를 갖는 절점이 승자가 되도록 HN을 변형하면 트렐리스도에서 경로 탐색하는 비터비 복호기 실현이 가능하다. MS를 HD로 계산하기 위한 하중과 임계치의 학습은 다음과 같다.

$$w_{ij} = -\frac{c_i^j}{2}, \theta_j = \frac{N}{2} \tag{4}$$

식 (4)을 식(3)에 대입하여 다시 정리하면, c_i^j 와 r_i 은 -1 또는 1 이므로 식(5)과 같다.

$$U_j = 0.5 \left[\sum_{i=0}^{N-1} (-c_i^j r_i) + N \right] = N_{dj} \tag{5}$$

따라서, 식(4)의 하중과 임계치를 적용한 하부망은 HD를 MS로 계산한다. HN의 상부망의 동작은 표 1과 같으며 순환이 종료되면 초기 상태가 가장 큰 절점만이 능동 상태로 수렴하고 그 절점을 승자라 한다. mHN의 하부망은 식(5)와 같이 HD를 MS로 계산하므로 상부망은 초기 상태(MS)가 가장 작은 절점만이 0이 되도록 순환해야 한다. 이와 같이 가장 먼저 0으로 비활성화된 절점이 승자가 되어 입력 벡터는 그 절점에 해당하는 클래스로 분류된다.

표 1. 해밍 네트워크의 상부망 동작
Table 1. Upper subnet operation of HM.

구분	동작
전달 함수	$f_t(U_j) = \begin{cases} U_j & U_j > 0 \\ 0 & U_j \leq 0 \end{cases}$
하중	$t_{kl} = \begin{cases} 1.0 & k=l \\ -e & k \neq l \end{cases}$
초기화	$U_j(0) = f_t \left(\sum_{0 \leq i \leq N_i} w_{ij} r_i + \theta_j \right)$
순환 후 j번째 절점 출력	$U_j(t) = f_t(U_j(t-1) - e \sum_{k \neq j,k} U_k(t-1))$

2. mHN을 이용한 비터비 복호기 설계

그림 3은 그림 1의 길쌈 부호기를 신경회로망 구조로

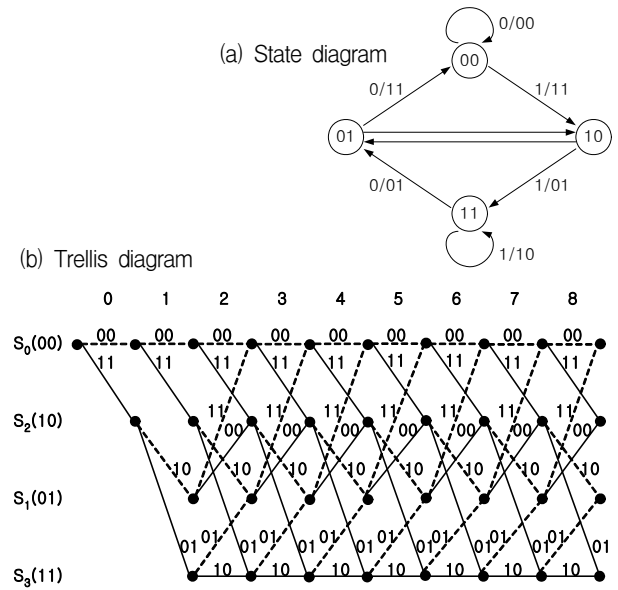


그림 3. 그림 1에 대한 상태도와 트렐리스도
Fig. 3. State diagram and Trellis diagram for Fig. 1.

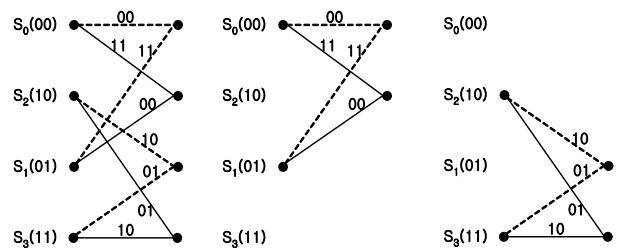


그림 4. 트렐리스도의 한 단 표현
Fig. 4. 1-stage of trellis diagram.

설계하기 위한 길쌈 부호기의 상태도 및 트렐리스도이다. 그림의 트렐리스도는 한단의 상태 수가 4개, 각 절점의 가능한 변이 경로 수가 2개씩이다.

mHN 구성의 기본 개념은 그림 3(b)에서 각 단의 상태 변이를 추정할 수 있도록 하는 것이다. 그림에서 각 절점은 최소 HD를 갖는 가치를 선택한다. 그러므로 어느 한 단에서는 s^m 개 절점으로의 경로와 길이를 기억해야 한다. 여기서, s 는 신호 레벨 수, s^m 은 상태 수 그리고 m 은 부호기의 메모리 개수이다.

그림 4(a)의 트렐리스도에서 각 절점은 상태 $s_0(00)$, $s_2(10)$, $s_1(01)$, $s_3(11)$ 에 해당하고 각 절점으로 변이 가능한 가치는 2개씩이다. 그림 4(b)에서 s_0 과 s_2 절점으로 변이 가능한 경로는 00와 01에서 시작하고 그림 4(c)에서 s_1 과 s_3 절점으로 변이 가능한 경로도 10과 11에서 시작한다. 그러므로 두 종류의 mHN가 필요하고 이를 net_02 및 net_13으로 표시하면 각 하중과 임

표 2. net_02와 net_13의 하중 및 임계치
Table 2. Weight and threshold for net_02 and net_13.

구분		Value
하부	하중	$w_{00} = -\frac{c_0^0}{2} = \frac{1}{2}$, $w_{01} = -\frac{c_0^1}{2} = -\frac{1}{2}$ $w_{10} = -\frac{c_1^0}{2} = \frac{1}{2}$, $w_{11} = -\frac{c_1^1}{2} = -\frac{1}{2}$ 만일 net_02이면 $c^0 = (-1, -1)$, $c^1 = (1, 1)$, net_13이면 $c^0 = (-1, 1)$, $c^1 = (1, -1)$
	임계치	$\theta_0 = \frac{N}{2} = \frac{2}{2} = 1$, $\theta_1 = \frac{N}{2} = \frac{2}{2} = 1$
상부	하중	$t_{00} = t_{11} = 1$, $t_{01} = t_{10} = -e = -\frac{1}{4}$

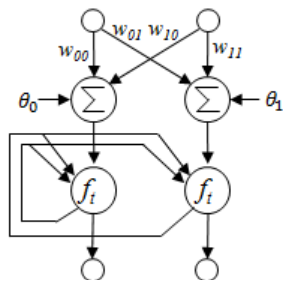


그림 5. 변형 해밍 네트워크 (mHN)
Fig. 5. Modified Hamming network.

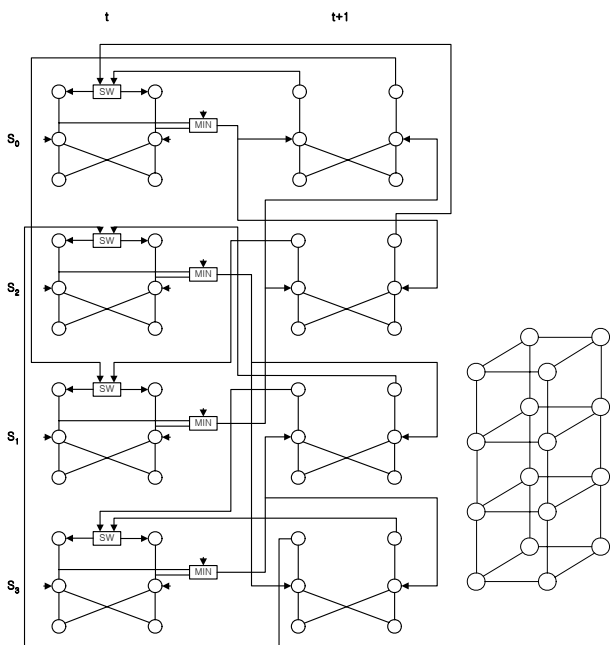


그림 6. mHN를 이용한 상태 병렬 비터비 복호기
Fig. 6. State parallel Viterbi decoder using Modified Hamming Network.

계치는 표 2와 같으며, 그림 5에 그 동작을 도식하였다. 상태 병렬 비터비 복호기는 트렐리스도의 한 단의 모

든 상태를 병렬로 처리하는 복호기로서 한 단의 계산을 할 수 있도록 구성한다. 상태 변이 추정은 가능 변이 경로 중 가장 작은 U_j 의 선택이므로 경로 탐색을 위한 트렐리스도의 절점마다 mHN가 필요하며, 이전 단의 경로 길이를 누산하므로, 선택 경로 저장을 위한 메모리가 필요하다.

그림 5의 회로가 인접 단 간 변이 경로를 선택하는 기능을 하고 각 단 사이에 삽입된다. 다음 단으로 진행 시 비터비 복호기는 그 전 경로를 기억해야 하므로 하부망의 MS에 이전 경로 값을 누산하도록 단 간을 연결하고 비터비 복호기에 의해 이전 경로가 제거되는 경우 그 경로에 해당하는 노드를 종료하도록 귀환연결한다.

그림 6은 그림 5의 mHN를 트렐리스도의 4개 절점에 해당하도록 연결한 구조로서, MIN 블록은 이전 경로의 최소치를 가산하기 위한 부분이며, SW 블록은 최소 경로로 선택된 절점만을 동작시킨다.

3 블록 병렬 비터비 복호기

그림 7은 상태 병렬 복호기를 기본으로 하는 창길이 10인 블록 병렬 비터비 복호기를 하였다. 설계된 복호기는 윈도우 길이만큼 상태 병렬 비터비 복호기를 병렬 접속한 구조로 생존 경로를 위한 기억 장치가 필요 없으므로 역추적하지 않는다. 이 복호기는 트렐리스도에서 한 단 씩 이동하면서 한 정보를 출력한다. 복호는 비터비 복호기 내에서 독립적으로 이루어지므로 완전한 로컬 처리가 가능하다. 복잡도는 증가하지만 구성이 규칙적이므로 VLSI 실현에 적합하다.

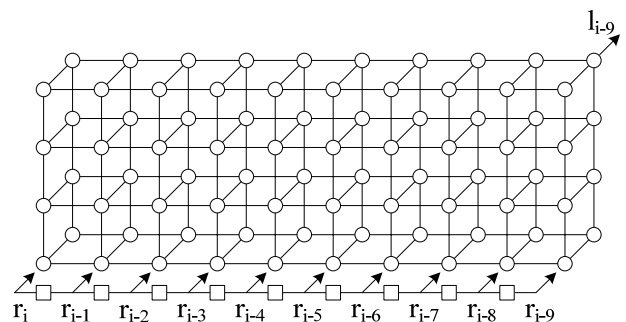


그림 7. 창길이 10인 블록 병렬 비터비 복호기
Fig. 7. Block parallel Viterbi decoder with window length 10.

V. 실험

제안 방법의 검토를 위해, 먼저 그림 1에 생성 다항식 $G(x) = (x^2 + x + 1, x^2 + 1)$ 인 부호기를 실험한다. 트렐리스도의 하중과 임계치는 표 2와 같다.

mHN가 단 간에 삽입된 그림 7의 블록 병렬 비터비 복호기에 입력 i 과 출력 c 인 전송 계열에 전송 채널의 오류로 인하여 수신 r 이 입력된다고 할 때, 복호를 트렐리스 도로 표현하면 그림 8과 같다.

$$i = (0,0,1,0,1,0,0,0,1,1,0,1,0,0,1,0,0,1,1,0)$$

$$c = (00,00,11,10,00,10,11,00,11,01,01,00,10,11,11,10,11,11,01,01)$$

$$r = (00,01,11,10,10,10,11,00,11,11,01,00,10,10,11,10,11,11,01,01)$$

제안된 mHN 복호기로 r 을 복호하면 정확한 복호기능을 수행하여 부호기 입력과 동일한 정보열을 얻을 수 있었다.

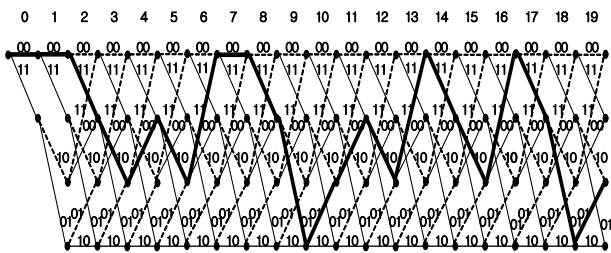


그림 8. 트렐리스도 상의 복호 과정
Fig. 8. Decoding process on trellis diagram.

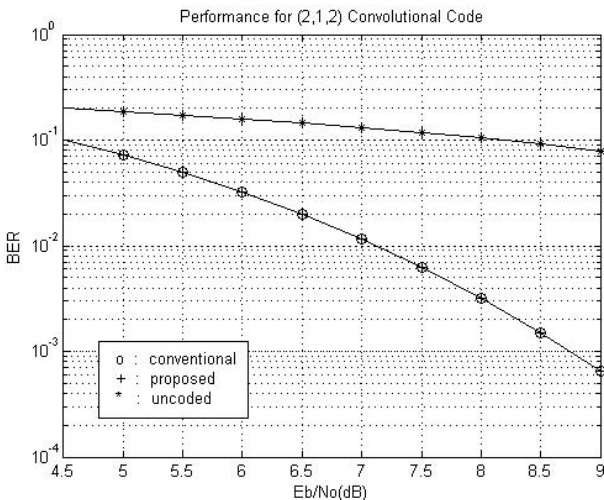


그림 9. AWGN 채널에 대한 시뮬레이션 결과
Fig. 9. Simulation result for AWGN channel.

표 3. mHN 복호기의 복잡도 비교

Table 3. Comparison of Complexity of mHN Decoder.

기존 비터비 복호기			제안하는 복호기		
구성	논리소자	# gate	구성	회로소자	개수
BMU	전가산기	10	상부망	저항	4
	전가산기	10		Op-amp	2
ACS	비교기	3	하부망	저항	4
	다중화기	2		Op-amp	2
SMU	기억소자	4			
계		29	계		12

표 4. 복호기 입출력 지연 시간 비교

Table 4. Comparison of input-to-output delay times.

방식	내용	delay(nsec)
기존	전가산기+비교기+다중화기+기억소자	50
제안	하부망+상부망	30

AWGN 채널에 대하여 부호화하지 않은 경우와 그림 7의 블록 병렬 비터비 복호기로 복호한 경우를 시뮬레이션 한 결과는 그림 9와 같다. 그림에서 윈도우 길이가 같은 경우 제안한 블록 병렬 비터비 복호기는 기존 복호기에 근접한 성능을 얻었다.

표 3은 기존 비터비 복호기와 제안하는 비터비 복호기의 복잡도를 비교하였다. 기존 비터비 복호기의 워드 길이가 4bit이면 $29 \times 4 = 116$ 개의 논리 게이트가 필요하지만, mHN 복호기는 기존 디지털 방식의 BMU, ACS 및 SMU가 mHN로 통합된 구조로 상부망은 BMU와 ACS의 가산 기능을 구현하고 하부망은 ACS의 비교/선택 기능과 SMU 기능을 구현하여 복잡도가 기존에 비하여 약 10% 감소하였다.

표 4는 기존 복호기와 mHN 복호기의 입출력 지연 시간을 계산하여 속도를 비교하였다. 이때 반도체 제조업체의 데이터 시트를 참조하면 상기 논리 소자 및 회로 소자의 입출력 전달 시간은 약 10[nsec]로 상이하다 [7]. 즉, mHN 복호기는 기존에 비해 약 60%의 처리 시간만이 필요하며, 역추적이 필요하지 않으므로 처리시간이 40%이상 개선될 수 있다.

VI. 결론

기존의 비터비 알고리즘은 구속장이 증가할수록 구현의 복잡도가 크게 증가되어 현장에 적용하기에 많은 어려움이 발생한다. 제안된 복호기는 HN를 수정하여

트렐리스도의 각 절점에 해당하는 기본 모듈로 이용하였다. 복호 속도 개선을 위하여 트렐리스도의 연속한 p 개 지로를 한 지로로 변환한 후 p 개 지로의 연산을 한 번으로 수행하여 복호 시간을 $1/p$ 로 감소시킬 수 있도록 변형하였다. 제안된 복호기는 이상적으로 동작하며 기존 비터비 복호기와 동등한 성능을 얻을 수 있다. 또한 기존 방법과 비교할 때, 복잡도는 10% 감소되고 지연 및 처리 시간은 40%이상 개선될 수 있다.

REFERENCES

- [1] Shu Lin, Daniel J. Costello, Jr, *Error Control Coding*, Prentice Hall, 2004.
- [2] Man Y. Rhee, *Error Correcting Coding Thoery*, McGraw Hill, 1988.
- [3] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260-269, Apr. 1967.
- [4] Q.N. Charei, H.F. Agfamalek, "An Improved soft decision method in Viterbi decoder using artificial neural networks," *IEEE PRIA*, pp. 1-4, March 2013.
- [5] G. David Forney, Jr., "The Viterbi Algorithm," *IEEE Proceedings*, vol. 61, No 3, pp. 268-278, March 1973.
- [6] *Neural Works Professional II and Neural Works Explorer*, Volume 1, "Neural Computing," 1989 Neural Ware Inc., pp. 165-177.
- [7] Richard P. Lippmann, "An Introduction to computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-22, April 1987.

저 자 소 개



권 용 광(정회원)

1998년 관동대학교 전자공학과
공학사.

2002년 동국대학교 전자공학과
공학석사(디지털 통신).

2008년 동국대학교 전자공학과
공학박사(영상처리).

<주관심분야 : 디지털 영상처리, 디지털통신>