

LVQ Network를 적용한 순방향 비터비 복호기

준회원 박 지 응

Forward Viterbi Decoder applied LVQ Network

Ji-woong Park Associate Member

요 약

IS-95와 IMT-2000 시스템에서 사용되고 있는 여러 종류의 길쌈 부호기를 부호율 1/2, 구속장 3인 길쌈 부호기로 한정하여, neural network의 LVQ(Learning Vector Quantization)과 PVSL(Prototype Vector Selecting Logic)을 적용하여 비터비 복호기에서 사용되는 PM(Path Metric)과 BM(Branch Metric) 메모리 수와 산술·비교 연산량을 줄임으로써 시스템의 단순화와 순방향 복호를 가능하게 한다

구속장의 확장성 여부와 관계없이 간단한 응용으로 기존의 비터비 복호기에 적용할 수 있는 새로운 비터비 복호기의 구조와 적용 알고리즘을 제시하고, 제시된 비터비 복호기의 합리성을 VHDL 시뮬레이션으로 검증 후, 기존의 복호기와의 성능을 비교 분석한다

Key Words LVQ(Learning Vector Quantization), PVSL(Prototype Vector Selecting Logic), 4-bit soft-decision

ABSTRACT

In IS-95 and IMT-2000 systems using variable code rates and constraint lengths, this paper limits code rate 1/2 and constraint length 3 and states the effective reduction of PM(Path Metric) and BM(Branch Metric) memories and arithmetic comparative calculations with applying PVSL(Prototype Vector Selecting Logic) and LVQ(Learning Vector Quantization) in neural network to simplify systems and to decode forwardly

Regardless of extension of constraint length, this paper presents the new Vierbi decoder and the appied algorithm because new structure and algorithm can apply to the existing Viterbi decoder using only uncomplicated application and verifies the rationality of the proposed Viterbi decoder through VHDL simulation and compares the performance between the proposed Viterbi decoder and the existing.

1. 서 론

비터비 복호기의 PM메모리 사용 용량의 증가에 대한 문제에 대하여 많은 연구가 지속되어 왔으며, 또한 BM연산에서의 동물의 문제점을 개선하기 위하여 연판정법도 도입되었음은 익히 알고 있을 것이다

기존의 비터비 복호기의 문제점들을 개선하기 위하여 Neural Network의 LVQ 이론과 본 논문에서 제안된 PVSL(Prototype Vectors Selecting Logic)을 응용·적용하여, 하드웨어의 복잡도와 복호 시간을 감소

시키는 비터비 복호기를 설계한다

본 논문에서는 부호율 1/2, 구속장 3인 길쌈 부호기를 이용하여, 본 논문에서 제시한 PVSL알고리즘의 합리성과 LVQ 알고리즘의 적합성을 VHDL을 사용한 시뮬레이션을 통하여 검증한다

본 논문의 구성은 제2장 비터비 순방향 복호기에서는 기존의 비터비 복호방식을 간략히 설명하고 제3장 LVQ Network를 적용한 비터비 복호기에서는 LVQ 이론과 적용된 PVSL 알고리즘에 대하여 자세히 설명한다 그리고 제4장 VHDL 시뮬레이션에서는 본 논

문에서 제시한 비터비 복호기의 알고리즘의 합리성을 VHDL 시뮬레이션을 통하여 검증한다 제5장 성능 비교에서는 기존의 비터비 복호기와 본 논문에서 제시한 비터비 복호기와의 성능차이를 분석한 후, 제6장에서 결론을 맺는다

II. 비터비 복호기

비터비 복호기는 부호어들의 열, 즉 길썬 부호기의 trellis에서 발생하는 모든 부호어들과 채널 수신된 부호어에 대하여 MAP 알고리즘을 이용한 ML(Maximum Likelihood)결정을 통하여 모든 경로에 대한 벡터적 거리를 계산하고 그 중 가장 작은 거리를 선택하여 현재 상태의 $PM(t)$ 에 저장하고, 다음의 상태의 채널 수신된 부호어와 부호어의 벡터적 거리를 $M(t+1)$ 으로 계산하여 이전의 $PM(t)$ 과 더함으로써 현재의 $PM(t+1)$ 을 결정한다 이러한 반복 과정으로 복호 trellis가 완성되면 전체 PM값을 통한 유사한 부호어들의 열을 역추적하는 과정을 통하여 복호한다

위의 복호 과정은 PM의 값이 계속 증가하게 되므로 PM의 메모리 용량을 크게 잡아야 하는 문제점이 발생된다

요약에서 언급하였듯이 이러한 문제점에 대한 여러 가지 개선 방향들이 연구되어 왔다^{[1][2][3]}

III. LVQ Network를 적용한 순방향 비터비 복호기

본 논문에서 제시된 비터비 복호기는 PVSL Block과 LVQ Network Block으로 구성되어 있으며, 4-bit 연판정 방식을 사용한다

1 1 PVSL Block

PVSL Block에서는 길썬 부호기의 trellis에서 발생 가능한 모든 부호어들의 현재 상태와 다음 상태로의 규칙적 변화를 적용하여 LVQ Network Block에서 데이터 복호에 사용되는 원형 벡터 UPV(Upper Prototype Vector)와 LPV(Lower Prototype Vector)를 LVQ Network Block으로 전송한다

$G_1=(111)$, $G_2=(101)$ 인 길썬 부호기의 trellis를 발생 부호어들의 규칙적 변화에 따라서 분해하여 그리면 그림 1과 같다

그림 1의 규칙성을 look-up table로 나타내면 표 1

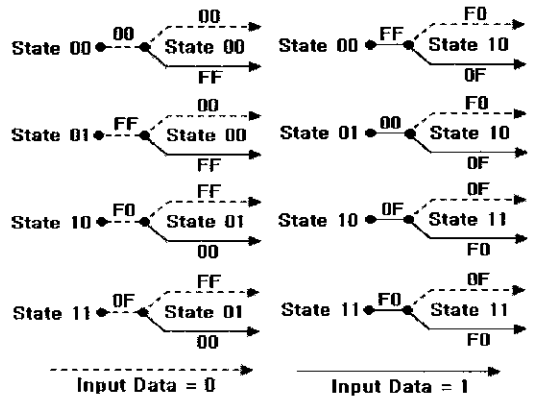


그림 1 발생 부호어들의 규칙적 변화에 의한 tree diagram 분해도

표 1 PVSL구성과 4-비트 A/D변환

PVSL₁ = Decision Data Bit Win Prototype Converting Bits
 Bit[2] Bit[1] Bit[0] Bit[2] Bit[1] Bit[2]
 MSB LSB

PVSL₂ = Invert Decision Data Bit Win Prototype Converting Bits
 Bit[2] Bit[1] Bit[0] Bit[2] Bit[1] Bit[2]
 MSB LSB

PVSL	UPV	LPV
000	00	FF
001	FF	00
010	FF	00
011	00	FF
100	F0	0F
101	0F	F0
110	0F	F0
111	F0	0F

A/D Converter Specifications

1111	-1V
1110	-0.87V
1101	-0.74V
1100	-0.61V
1011	-0.48V
1010	-0.35V
1001	-0.22V
1000	-0.09V
0111	+0.09V
0110	+0.22V
0101	+0.35V
0100	+0.48V
0011	+0.61V
0010	+0.74V
0001	+0.87V
0000	+1V

Win Prototype Converting Bit => 00 => 00 F0 => 10
 0F => 01 FF => 11

과 같다 표 1에서는 PVSL의 상세한 내부 구성과 4-bit A/D변환을 나타내고 있다

Win Prototype Converting Bits는 LVQ Network Block의 복호 시점에서 Decision Data에 상응되는 UPV 또는 LPV를 2비트 형식으로 변환한 것이다

부연 설명을 하자면, UPV나 LPV는 모두 그림 1에서 부호어들의 규칙적인 변화를 따르는 발생 부호어들 중의 하나이며 PVSL은 그 발생 부호어들을 규칙적인 변화에 맞게 선택하여 LVQ Network Block으로 전송한다

기존의 비터비 복호기가 8개의 발생 부호어들을 전부 사용하는 반면, 본 논문에서 제시된 비터비 복호기는 규칙적 변화에 의하여 선택된 4개의 발생 부호어들만을

사용하여 각각을 UPV_1과 LPV_1를 class_1, UPV_2와 LPV_2를 class_2로 분류하여 전송된 부호어와 벡터적 거리를 비교하여 가장 가까이 있는 UPV나 LPV가 있는 class를 LVQ Network로 전송한다

위에서 설명한 규칙적 변화를 다시 간략하게 설명하자면 현재 상태에서 다음의 상태로 가는 모든 경우의 수는 2가지이며 그 2가지 상태에 상응하는 발생 부호어들은 4가지라는 것이다

본 논문의 A/D변환에서는 아날로그적 같은 거리인 0V가 제외되어 있으나, 비터비 복호기의 기본 PM선택 알고리즘인, 같은 벡터적 최소 거리에 있는 여러가지의 경로 중 임의의 하나만을 선택한다는 것을 생각하여 볼 때, 그다지 심각한 오류를 발생시키지는 않을 것이다

위의 문제에 대한 해결 방안으로 채널 수신된 부호어와 벡터적 같은 거리에 위치하고 있는 두 부호어들 사이의 복호는 UPV가 우선하며, 다음의 채널 수신된 부호어와 두 개의 classes내의 벡터적 최소 거리에 있는 UPV와 LPV가 속한 class를 선택하여 이전의 복호된 데이터의 결정 값을 수정한다

1 2 LVQ Network Block

LVQ Network Block에서는 LVQ 알고리즘을 이용하여 PVSL에서 전송된 classes를 LVQ Network의 원형 벡터로 정의하고 채널수신된 순차적 부호어들을 weight 벡터로 입력하여, 반복된 학습에 의하여 weight 벡터를 update 한다

기존의 LVQ network와는 입력부분에서 차이가 있으나 이 입력 부분을 약간의 응용된 적용을 통하여, 기존의 원형 벡터 입력을 추종하는 weight 벡터 update 시스템 형식에서, 입력을 weight 벡터로 하여 원형 벡터들을 추정하는 시스템으로 변형하였다

그림 2는 변형된 LVQ Network의 내부 구성도를 나타낸다

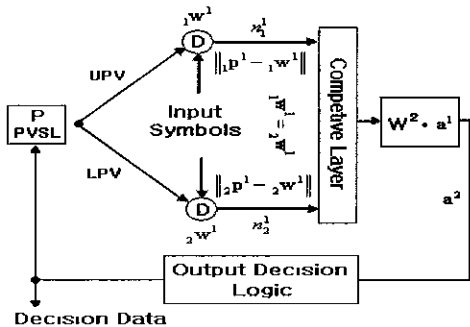


그림 2 변형된 LVQ Network의 내부 구성도

그림 2에서 Input Symbols로 사용되는 수신 순차적 부호어들은 같은 메모리를 사용하고 있는 weight로 입력된다 그림 2에서는 이해의 편의를 위하여 마치 2개의 메모리를 사용하는 것처럼 묘사하였다

첫 번째 layer에서는 PVSL Block에서 전송한 class를 UPV와 LPV로 정의하고 AWGN채널 수신된 부호어들은 weight로 입력된다 Competitive Layer에서는 UPV, LPV와 weight사이의 벡터적 최소 거리를 비교하여 Win prototype vector를 선택한다 Weight는 두 번째 layer의 출력을 이용하여 2개의 UPV와 LPV중 승리한 원형 벡터로 $\alpha = 1$ 인 Kohonen rule로 update 된다

두 번째 Layer는 supervised learning을 사용하여 두 종류의 classes로 분류한다 하나는 upper 방향이고 다른 하나는 lower 방향이다 즉, upper 방향의 class가 Decision Data = 0이고, lower 방향의 class가 Decision Data = 1이 된다

결국 두 번째 Layer의 출력을 통하여 Decision Data를 결정하게 된다

다시 말하면 채널 수신된 부호어들을 첫 번째 layer에서 가장 유사한 부호어로 경쟁 · update 하고, 두 번째 Layer에서 길잡 부호기의 입력 데이터를 복호하게 된다

미리 결론 부분을 잠시 언급하자면 기존의 비터비

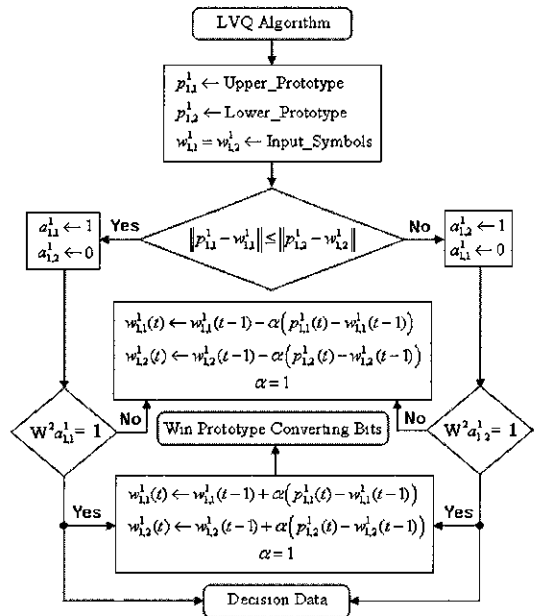


그림 3 변형된 LVQ Network의 알고리즘의 순서도

복호기는 전송된 데이터 개수의 길이를 갖는 trellis를 형성한 후에 역추적으로 채널 전송된 데이터를 결정하는데 반해, 본 논문에서 제시한 비터비 복호기에서는 채널 전송된 부호어들을 복원과 거의 동시에 입력 데이터를 순방향 복호한다.

그림 3은 변형된 LVQ Network의 알고리즘의 순서도이다.

IV. VHDL 시뮬레이션

3장에서 설명한 PVSL Block과 LVQ Network Block을 VHDL로 프로그래밍하여 시뮬레이션을 한다.

본 논문에서의 VHDL 시뮬레이션은 AWGN채널을 단순하게 모델링하여 전체 알고리즘의 합리성을 검증하는데 편리하게 한다.

시뮬레이션 파라미터는 클럭 주기 2.0 μ s, reset시 입력 부호어들의 값은 모두 00₍₁₆₎, 각 입력 데이터의 전송 시간은 0.1ms로 한다. 이는 IS-95의 순방향 링크의 데이터 전송율과 근접하게 잡았다.

1.1 PVSL Block

PVSL Block의 VHDL시뮬레이션은 그림 4와 그림 5이다.

그림 4는 1.0ms까지의 시뮬레이션 결과를 나타내며, 그림 5는 1.9ms까지의 시뮬레이션 결과를 나타낸다.

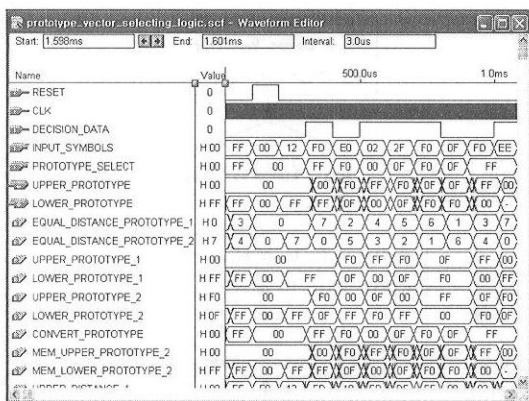


그림 4. 1.0ms까지의 PVSL Block의 VHDL 시뮬레이션 결과

그림 4와 그림 5를 다시 정리하여 표현한 그림 6을 보면,

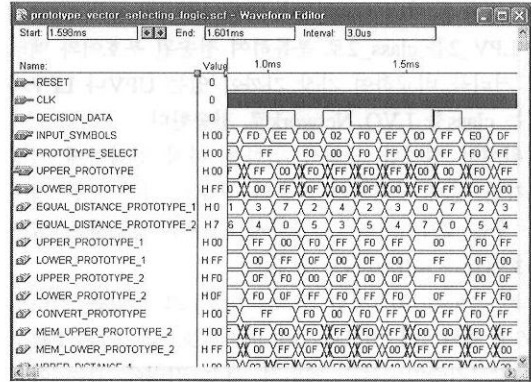


그림 5. 1.9ms까지의 PVSL Block의 VHDL 시뮬레이션 결과

```

INPUT_SYMBOLS
=[12 FD E0 02 2F F0 0F FD EE D0 02 F0 EF 00 FF E0 DF]
DECISION_DATA
=[0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0]
PROTOTYPE_SELECT
=[00 FF F0 00 0F F0 0F FF FF F0 00 F0 FF 00 FF F0 FF]
PROTOTYPE_VECTOR_SELECTING_LOGIC (PVSL)
=[000 111 010 100 101 110 001 011 111 010 100 010 011 000 111 010 011]
UPPER_PROTOTYPE (UPV)
=[00 00 0F FF F0 0F F0 FF 00 F0 FF F0 FF 00 00 F0 FF]
LOWER_PROTOTYPE (LPV)
=[FF FF F0 00 0F F0 0F 00 FF 0F 00 0F 00 FF FF 0F 00]
    
```

그림 6. PVSL Block의 입·출력 데이터

DECISION_DATA와 AWGN 채널을 통과한 INPUT_SYMBOLS 그리고 LVQ Network Block의 경쟁에서 승리한 원형 벡터로 update된 weight 벡터, 즉 VHDL 변수명 PROTOTYPE_SELECT를 변수로 사용하는 PVSL 알고리즘으로 PVSL Block은 정확한 UPV와 LPV를 LVQ Network Block으로 전송하고 있음을 알 수 있다.

그림 6을 살펴보면, INPUT_SYMBOLS와 UPV, LPV의 벡터 최소 거리(최소 해밍거리)에 상응하는 DECISION_DATA를 쉽게 결정할 수 있을 것이다. 그러므로 PVSL Block은 길쌈 부호기의 입력 데이터에 해당되는 2가지 경로의 발생 부호어들을 정확히 출력하고 있음을 알 수 있다.

VHDL의 시간 지연으로 인한 타이밍 지터 현상으로 길쌈 부호기의 입력 데이터 전송율에 정확하게 맞지는 않지만, 이것을 하드웨어 설계상의 문제점으로 생각한다면, PVSL의 알고리즘은 정확히 검증되었음을 알 수 있다.

1.2 LVQ Network Block

LVQ Network Block을 검증하기에 앞서, 한 가지 제한 조건을 두어 적용 알고리즘의 이론 검증을 편리하게 하고자 한다. 제한 조건은 VHDL 시뮬레이션에 한하여, PVSL Block에서 전송되는 UPV와 LPV 데이터 전송 시간과 길쌈 부호기의 입력 데이터 전송 시간이 정확히 동기화되었다고 가정하는 것이다.

부연 설명을 하자면, 4.1절에서 언급하였듯이 VHDL 코딩에 있어서 고유의 시간 지연 현상과 알고리즘 처리에 관계되는 시간 지연 현상으로 인한 타이밍 지터 현상이 발생되므로, PVSL Block에서 발생하는 타이밍 지터의 왜곡 영향을 줄이기 위하여 LVQ Network Block에서 UPPER_PROTOTYPE(UPV)와 LOWER_PROTOTYPE(LPV) 데이터 전송 시간을 길쌈 부호기의 입력 데이터 전송 시간과 동기화한 것이다.

LVQ Network Block의 VHDL 시뮬레이션은 그림 7과 그림 8이다.

그림 7은 1.0ms까지의 VHDL 시뮬레이션 결과를 나타낸다.

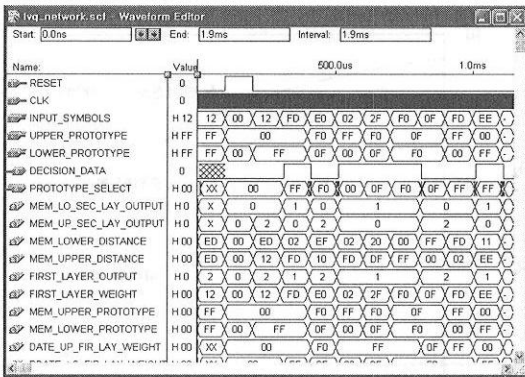


그림 7. 1.0ms까지의 LVQ Network Block의 VHDL 시뮬레이션 결과

그림 8은 1.9ms까지의 VHDL 시뮬레이션 결과를 나타낸다.

그림 7과 그림 8을 다시 정리하고 표현한, 그림 9를 보면 AWGN 채널을 통과한 INPUT_SYMBOLS와 PVSL Block에서 전송된 UPV와 LPV를 사용하여, 첫 번째 layer는 ML결정 방식을 이용하여 두 원형 벡터와 INPUT_SYMBOLS간의 경쟁을 시키고, 두 번째 layer에서는 데이터를 결정하는 두 개의 classes중 하나의 class로 수렴시킨다. 수렴된 class의 weight 벡터는 다시 경쟁에서 승리한 원형 벡터로

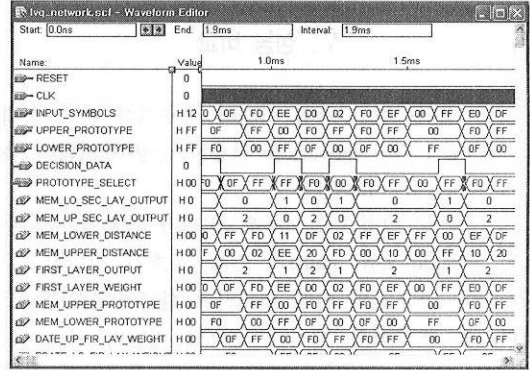


그림 8. 1.9ms까지의 LVQ Network Block의 VHDL 시뮬레이션 결과

update되어 PVSL Block으로 전송되는 것을 알 수

```

INPUT_DATA
=[0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0]
INPUT_SYMBOLS
=[12 FD E0 02 2F F0 0F FD EE D0 02 F0 EF 00 FF E0 DF]
DECISION_DATA
=[0 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0]
PROTOTYPE_SELECT
=[00 FF F0 00 0F F0 0F FF FF F0 00 F0 FF 00 FF F0 FF]
UPPER_PROTOTYPE (UPV)
=[00 00 F0 FF F0 0F 0F FF 00 F0 FF F0 FF 00 00 F0 FF]
LOWER_PROTOTYPE (LPV)
=[FF FF 0F 00 0F F0 F0 00 FF 0F 00 0F 00 FF FF 0F 00]
    
```

그림 9. LVQ Network Block의 입·출력 데이터

있다.

그림 9를 통한 검증으로도 INPUT_SYMBOLS와 UPV, LPV의 벡터 최소 거리(최소 해밍거리)에 상응하는 DECISION_DATA를 쉽게 결정할 수 있다.

결정된 DECISION_DATA와 길쌈 부호기의 입력 데이터 INPUT_DATA를 비교하여 보면 서로 일치함을 알 수 있다. 결국 LVQ Network Block의 알고리즘이 검증되었음을 알 수 있다.

PVSL Block에서와 마찬가지로 시뮬레이션상의 타이밍 지터 현상은 하드웨어 설계의 동작적 검증의 문제로 간주한다.

위의 두 가지 Block에 대한 간단한 결론을 미리 내리자면, PVSL Block에서는 PM과 BM의 사용 개수를 1/2로 줄이고, 또한 메모리의 용량도 최소화하는 알고리즘을 검증한 것이며, LVQ Network Block에서는 기존의 비터비 복호기의 데이터 결정 방식인 역추적 방식을 개선하여 전송된 부호어들을 복원하는 거의 동시에 전송 데이터를 순방향 복호하는 알고리즘을 검증한 것이다.

V. 성능 비교

이번 장에서는 기존의 비터비 복호기와 본 논문에서 제시된 LVQ Network를 적용한 순방향 비터비 복호기의 성능을 비교한다.

성능 비교 방법으로 기존의 비터비 복호기는 모두 영인 경로(all-zero path)의 길쌈 부호기 전달함수를 이용하여 10 bits의 입력 데이터에 한하여 BER을 구하며, 본 논문에서 제시된 순방향 비터비 복호기는 10 bits 입력 데이터에 대한 제한된 오류정정 능력 BER을 구하여 비교한다.

순방향 비터비 복호기의 제한된 오류정정 능력이란 비터비 복호기의 복호 성능의 기준인 자유거리보다 작은 자유거리를 가짐으로 인한 복호 성능의 저하를 막기 위해 부가된 소프트웨어적인 복호능력을 말한다.

G₁=(111), G₂=(101), 코드율 1/2, 구속장 3인 길쌈 부호기에 대한 비터비 복호기의 4-bit soft-decision BER를 구하면

$$P_{B,AVGN}(e) \leq \frac{1}{k} Q\left(\sqrt{\frac{2drE_b}{N_0}}\right) e^{\frac{drE_s/N_0}{\partial N}} \frac{\partial T(D,N)}{\partial N} \quad N=1, D=e^{-rE_s/N_0} \quad (1)$$

$$\frac{\partial T(D,N)}{\partial N} \quad N=1, D=e^{-rE_s/N_0} = \frac{\partial}{\partial N} \left(\frac{D^{20}N}{1-2D^4N} \right) \quad N=1, D=e^{-rE_s/N_0}$$

$$= D^{20}(1+2 \cdot 2D^4+3 \cdot 2^2D^8+4 \cdot 2^3D^{12}+75D^{16}+96D^{20}+49D^{24}+8D^{28}) \quad D=e^{-rE_s/N_0} \quad (2)$$

와 같다. 위의 식(1)과 식(2)에 k=1, d=20, r=1/2를 대입하여 식(1)을 간단히 정리하면

$$P_{B,AVGN}(e) \leq Q\left(\sqrt{\frac{10E_b}{N_0}}\right) \cdot (1+4e^{-2E_s/N_0}+12e^{-4E_s/N_0}+32e^{-6E_s/N_0}+75e^{-8E_s/N_0}+96e^{-10E_s/N_0}+49e^{-12E_s/N_0}+8e^{-14E_s/N_0}) \quad (3)$$

이 된다.

기존의 비터비 복호기와 동일한 길쌈 부호기를 사용하여, 본 논문에서 제시된 순방향 비터비 복호기의 4-bit soft-decision BER를 구하면

$$P_{B,AVGN}(e) \leq 5 \cdot 2 \left(Q\left(\sqrt{\frac{2rE_b}{N_0} \cdot 2d}\right) \cdot Q\left(\sqrt{\frac{2rE_b}{N_0} \cdot d}\right) \right) \quad (4)$$

와 같이 비터비 전달함수만의 BER을 구할 수 있다. 위의 식(4)에 d=4, r=1/2를 대입하여 간단히 하여 정리하면 식(5)와 같다.

$$P_{B,AVGN}(e) \leq 5 \cdot 2 \left(Q\left(\sqrt{\frac{E_b}{N_0} \cdot 8}\right) \cdot Q\left(\sqrt{\frac{E_b}{N_0} \cdot 4}\right) \right) \quad (5)$$

식 (4)는 제한된 오류정정 능력을 표현한 식이다. 부연 설명을 하면, 순방향 복호를 수행하면서 첫 번째 천이 에러가 발생했을 때, 혹은 두 번째 천이 과정 중에 에러가 발생하였다고 하더라도, 2개의 연속적인 천이과정에서 2개의 천이 과정 모두 에러가 아니라면 이전 천이 과정 중에 발생된 하나의 비트 에러를 정정할 수 있는 소프트웨어적 정정능력을 말한다.

최소 자유거리 d 는 해밍 거리를 이용한 비터비 전달함수에서 중요한 역할을 한다. 전달함수에서 최소 자유 거리의 표현은 Q-Function Inequality를 이용하여 간단히 나타내어진다. Soft-decision에서의 최소 자유거리에 대한 정확한 관점은 해밍 거리가 아닌 Euclidean 거리로 계산되어야 한다는 것이다.

식 (3)과 식 (4)를 최대 에러 상한 값을 이용하여 비교한 그래프인 그림 10을 보면 본 논문에서 제안된 순방향 비터비의 BER 성능이 비터비 복호기보다 뒤지고 있음을 알 수 있다. 하지만 같은 비트 soft-decision의 코딩되지 않은 기저대역 전송의 2진 반극신호와 비교해볼 때 성능이 우수하다는 것을 알 수 있다.

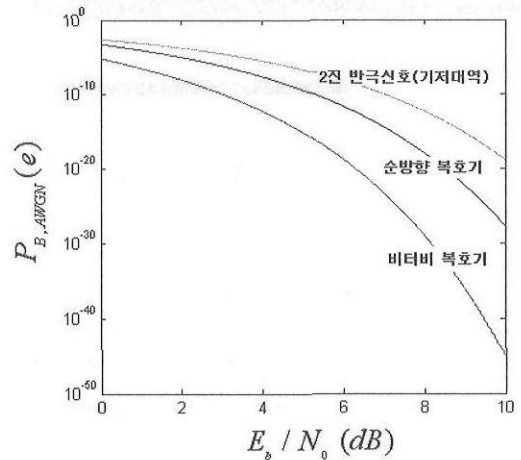


그림 10. 본 논문에서 제시된 순방향 비터비 복호기와의 BER 성능 비교

VI. 결론

본 연구의 VHDL 시뮬레이션 결과로부터 LVQ 알고리즘과 PVSL 알고리즘의 합리성을 확인하였으며, 제안된 순방향 비터비 복호기가 비록 비터비 복호기 보다는 BER 성능이 뒤지지만 코딩되지 않은 2진 반

극 기저대역 신호보다는 성능이 우수하다는 것을 알 수 있었다.

본 연구의 결과로서 제시된 순방향 비터비 복호기와 기존의 비터비 복호기를 비교하여 보면 표 2와 같다.

표 2. 본 논문에서 제시한 순방향 비터비 복호기와 비터비 복호기와의 비교

	비터비 복호기	순방향 비터비 복호기
사용 알고리즘	BM : Hamming Distance PM : Storage and Comparison	BM : $\ P - W^1\ $ or Hamming Distance PM : Two Classes ($W^2 a^1 = a^2$)
복호 방법	4개의 PM과 8개의 BM을 사용하여 모든 현이 상태의 trellis를 구성한 후, trace-back 방법을 사용하여 데이터 복호. Between PM(t-1) and PM(t) Upper Direction='0' Lower Direction='1'	각 단계마다 4개의 BM을 사용하여 데이터 순방향 복호와 제한된 오류 정정 BM : Two Classes PM : Two Classes Upper Class='0' Lower Class='1'
개선 방향	1. 각 단계에서 증가되는 PM값을 threshold함. 2. 각 단계에서 증가되는 PM값을 현재 단계의 최소 PM값으로 치등시킴. 3. 각 단계에서 증가되는 PM값을 circular counter를 사용하여 threshold함.	4개의 BM을 사용 2개의 PM을 사용 (PM값이 증가하지 않음)

본 논문에서 제시된 순방향 비터비 복호기는 하드웨어(PM, BM, trace-back 메모리)의 사용량을 최소화하여 통신 시스템을 단순화하며, 산술·비교 연산량 감소로 기존의 기저대역 전송보다 고속의 정확한 데이터통신을 가능하게 할 수 있을 것이다.

본 연구결과인 순방향 복호와 단순한 하드웨어의 구성의 장점은 보다 효율적인 복호 시스템을 구축하려는, 혹은 순방향 실시간 복호기를 개발하려는 연구원들에게 새로운 관점의 설계방향을 제시할 것이다.

참 고 문 헌

[1] Han, Y.S, Poning Chen, Hongbin Wn, "A maximum-likelihood Soft- decision sequential decoding algorithm for binary convolutional codes," IEEE Transactions On Communications, vol. 50, No. 2, pp. 173-178, 2002.

[2] Anderson, J.B, Hladik, S.M, "An optimal circular Viterbi decoder for the bounded distance criterion," IEEE Transactions On Communications, vol. 50, No. 11, pp. 1736-1742, 2002.

[3] Boutillon, E, Gross, W.J, Gulak, P.G, "VLSI architectures for the MAP algorithm," IEEE Transactions On Communications, vol. 51, No. 2, pp. 175-185, 2003.

[4] Martin T. Hangan, Howard B. Demuth, Mark H. Beale, *NEURAL NETWORK DESIGN*, PWS PUBLISHING COMPANY, pp. 14.16-14.21, 1999.

박 지 웅 (Ji-wong Park)

준회원



1998년 8월 : 광운대학교 제어계측공학과 졸업
2004년 3월 : 광운대학교 제어계측공학과 석사

<관심분야> 통신시스템제어, 전력제어공학