

DSP 칩을 이용한 구속장이 6인 (2,1) Viterbi복호기의 구현에 관한 연구

*최상호, 노종선
건국대학교 전자공학과

A Paper on the DSP Chip Implementation of (2,1) Viterbi Decoder with Constraint Length 6

*Sang-ho Choi, Jong-seon No

Dept. of Electronic Engineering, Kon-Kuk University

ABSTRACT

In this paper, the Viterbi decoder of punctured convolutional codes which are derived from (2,1) convolutional code with constraint length 6 and generator polynomial (65,57) are simulated by computer. As a decoder of convolutional code with the constraint length 6 and generator polynomial (65,57), Viterbi decoder which is widely being used in the area of satellite communications and digital cellular system is implemented on SDE1600, the emulator board of Samsung SSP1600 fixed point digital signal processing chip. In this hardware, the number of soft decision bits and traceback depth are programmable. In order to test Viterbi decoder implemented on emulator board by using IBM-PC, test tool is developed by Borland C on IBM-PC 386 and the test results of implemented Viterbi Decoder are shown by using the developed test tool on IBM-PC.

I. 서론

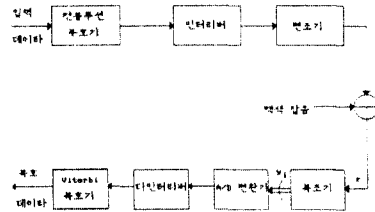
(n, k) 길쌈부호기(convolutional encoder)와 연성판정 Viterbi복호기(soft decision Viterbi decoder)는 우수한 오류정정 능력 때문에 위성통신과 디지털 셀룰라 시스템(digital cellular system)등에서 널리 사용되고 있다.

이 부호의 특징은 백색가우시안잡음 채널(additive white Gaussian noise channel)에서 오류정정능력이 특히 우수하고, 잡음이 연집특성을 갖는 채널에서도 interleaver / deinterleaver를 이용하여 쉽게 연집에러를 정정할 수 있다. 또한 연성판정 복호는 경성판정 복호(hard decision decoding)보다 약 2dB의 부호이득(coding gain)을 추가로 얻을 수 있고, 낮은 전송율을 극복하기 위해서 puncturing에 의한 높은 부호율(high rate code)을 갖는 부호부호기를 구현하기가 용이하다. 길쌈부호의 복호는 여러가지 방법이 있는데, 구현시 하드웨어는 가장 복잡하지만 복호의 성능이 최적이라고 알려진 Viterbi복호기가 있다. 이는 연성판정 복호가 용이하여 높은 부호이득을 얻을 수 있으나 하드웨어 복잡성때문에 사용되지 않고 있다가 최근에서야 VLSI기술의 발달에 의해 개발되어 활용되고 있다. 본 논문에서 IBM-PC의 Borland C를 이용하여 Viterbi복호기와 punctured Viterbi복호기의 구현하여 Viterbi복호기의 성능을 분석하였고, 또한 삼성전자의 fixed point digital signal processing(DSP)칩인 SSP1600의

emulator board, SDE1600상에서 DSP코드를 이용하여 Viterbi 복호기를 구현하였다. 여기서 사용한 길쌈부호기는 구속장(constraint length)이 6이고 생성다항식(generator polynomial)이 (65,57)이다.

II. 길쌈부호기와 Viterbi복호기의 이론적 고찰

2.1 디지털 통신시스템의 블럭도와 채널모형



그림(1) 디지털 통신시스템의 블럭도
Fig. (1) Block diagram of digital communication system.

본 논문에서는 부호화된 데이터가 PSK변조되어 가산성 백색 가우시안잡음 채널을 통하여 전송되었다고 가정하였다. 그리고 수신단에서는 잡음이 포함된 PSK신호가 동기검파되어 얻어진 기저대역 신호로 매 심볼시간마다 표본화된다고 가정하였다. 따라서 표본화된 기저대역 신호 r은 다음과 같이 표현될 수 있다.

$$r = \pm \sqrt{E_s} + n \quad \text{식(1)}$$

여기서 E_s 는 부호화된 심볼당 신호에너지이고, n은 백색가우시안 잡음이다. 이러한 백색가우시안잡음 채널의 경우 수신측에서 신호대 잡음비는 E_s/N_0 로 표현되며 길쌈부호기의 부호율이 $R=(k/n)$ 일때 부호화하기전에 정보데이터의 에너지, 즉 E_b 와 부호화된 심볼당 신호에너지, 즉 E_s 와의 관계는 아래의 수식(2)과 같이 표현될 수 있다.

$$E_s = R \cdot E_b \quad \text{식(2)}$$

여기서 $E_b = \int_0^T s^2(t) dt$ 는 비트당 에너지이고, s(t)는 부호화된 변조신호이다. 따라서 신호대 잡음비의 관계는 다음과 같이 표현된다.

$$\frac{E_s}{N_0} = R \cdot \frac{E_b}{N_0} \quad \text{식(3)}$$

여기서 N_0 는 단측파대 잡음전력 스펙트럼밀도 (single-sided noise power spectral density)이며, 채널부호화 및 복호기를 사용하는 경우 성능평가의 척도로서 비트 오류율(bit error rate, BER)이 있으며, 이는 E_b/N_0 값에 의해서 결정된다.

2.2 길쌈부호기

구속장이 $K(=m+1)$ 인 (n, k) 길쌈부호기는 m 개의 이진 시프트 레지스터(shift register)와 n 개의 이진 가산기(mod-2 adder)가 시프트레지스터에 연결된 선형 시퀀셜회로(linear sequential circuit)이다. 즉 정보비트가 한번에 부호기 시프트레지스터에 k 비트가 천이되고, 이후에 이진 가산기들은 코드심볼들을 순서있게 연속적으로 부호 출력을 하게 된다. 여기서 부호율(R)은 k/n 가 되고, 한개의 정보 비트가 부호기 출력에 영향을 주는 범위를 구속장이라 하는데 이를 K 로 표현한다. 이는 메모리 소자의 갯수에 하나를 더한 값인데 Viterbi복호기의 하드웨어의 복잡성을 결정해 주는 것으로 K 값이 1 증가하면 하드웨어의 크기는 2배로 증가한다. 또한 일반적으로 메모리 소자에서 이진 가산기에 어떻게 연결하는가에 따라 generator polynomial이 달라지게 되어 길쌈부호기가 역시 달라지게 되고, 이에 따라서 부호기의 성능도 달라지게 된다. 실제로 관심의 대상이 되는 길쌈부호기는 k 값이 1이고, n 값이 2인데 가장 좋은 성능을 갖는 생성다항식이 구속장 K 값에 대해 표 2-1에 제시되어 있다[1].

2.3 양자화, BER성능과 부호이득(Coding Gain)

수신기에서 수신된 신호가 복조기에서 복조되었을 때 복조된 데이터를 이진으로 하는 경성판정 대신에 A/D변환기를 이용한 Q레벨 연성판정으로 데이터를 복호하면 복호기의 성능을 향상시킬 수 있다. 복조된 아날로그 기저대역 신호(baseband signal) r 은 A/D변환기에 의해 디지털 데이터 $y_i (i=1, 2, 3, 4, 5, 6, \dots, Q)$ 인 Q레벨로 양자화 되고, 여기서 A/D변환기는 미리 결정된 연성판정 임계값 b_1, b_2, \dots, b_{Q-1} 에 의해 복조된 신호를 양자화한다. 이렇게 얻어진 Q레벨 연성판정된 데이터 y_i 가 Viterbi복호기에 보내진다.

(n, k) 길쌈부호의 복호기로서 Viterbi복호기가 사용될 때 비트 오류 확률(P_b)의 상위 경계값은 경성판정($Q=2$)인 BSC(binary symmetric channel)채널과 이상적인 연성판정($Q = \infty$)인 경우만 주어지고, 그 이상은 계산의 어려움 때문에 많은 시도가 있었지만 정확하게 계산하기는 어려운 것으로 알려져 있다.

이 비트 오류율에 커다란 영향을 줄 수 있는 것으로 minimum free distance(d_{free})라는 것이 있는데, 이에 대한 정의는 아래의 수식과 같이 표현될 수 있다.

$$d_{free} = \min_{u_1, u_2} \{ d(v_1, v_2) : u_1 \neq u_2 \} \quad \text{식(4)}$$

여기서 v_1 과 v_2 는 0이 아닌 모든 가능한 정보시퀀스 u_1 과 u_2 , 각각에 대응되는 codeword들을 표시하는데 이들 codeword 사이의 가장 작은 거리(distance) d 를 갖는 값이다. 또한 BER 성능은 연성판정 threshold spacing $T(=|b_i - b_{i-1}|)$ 에 따라 크게 좌우되는데, T 는 4레벨과 8레벨 연성판정의 경우에 대해 아래의 식과 같이 정규값 α 가 b_i 대신 판정 임계값의 변수로서 사용된다.

$$T = \alpha \sqrt{N_0/2} \quad Q = 8$$

$$T = 2\alpha \sqrt{N_0/2} \quad Q = 4 \quad \text{식(5)}$$

여기서 보통 α 는 최적인 경우 0.5~0.6으로 알려져 있고, 본 논문에서의 α 값은 실험한 데이터에 따라 설정하였다.

채널부호를 사용하는 경우에 채널부호의 성능을 평가하는 척도로 부호이득(coding gain)이라고 하는 것이 있다. 즉 채널부호를 사용하는 경우에 채널부호를 사용하지 않는 경우보다 송신기에서 출력전력을 얼마나 절감할 수 있는가를 나타내는 것으로 아래의 수식으로 표현될 수 있다. 주어진 BER에서 부호이득 G 는 [2]

$$G = (E_b/N_0)_{\text{without FEC}} - (E_b/N_0)_{\text{with FEC}} \text{ [dB]} \quad \text{식(6)}$$

여기서 $(E_b/N_0)_{\text{without FEC}}$ [dB]는 주어진 BER을 얻기 위하여 채널부호를 사용하지 않았을 경우에 요구되는 E_b/N_0 이며 $(E_b/N_0)_{\text{with FEC}}$ [dB]는 주어진 BER을 얻기 위하여 채널부호를 사용하였을 경우에 요구되는 E_b/N_0 이다.

2.4 Viterbi복호

Viterbi복호는 복호성능이 최적이거나 짧은 구속장을 갖는 부호에 적합한 복호기로서 Viterbi복호 알고리즘은 ML(maximum likelihood)복호로서 알려져 있다. Viterbi복호 알고리즘은 수신된 부호심볼 시퀀스를 trellis상에서의 가능한 여러 경로에 대해 확률을 계산하여 가장 큰 확률을 갖는 경로를 선택하는 알고리즘인데, 선택된 경로에 해당되는 정보비트들은 복호기의 출력이 된다. Trellis도에서 하나의 상태(state)에서 나가는 가지(branch)수는 2개가 되고, 또한 하나의 상태로 들어오는 가지수 역시 2개가 되는데 수신된 부호심볼과 비교하여 각 가지수의 확률을 계산한 후 그중 가장 큰 확률을 갖는 가지가 남고 나머지는 제거된다.

주어진 trellis도의 시간 t 에서 $2^k (k-1)$ 상태들 각각에 대해 반복되고, 각 상태로 들어오는 가지수는 단지 한개의 경로만 남게 되어 trellis에서 시간 t 가 진행됨에 따라 위의 과정들을 반복하게 된다. Trellis상에서 이렇게 남은 가지들은 역으로 traceback해서 데이터를 복호하게 된다.

2.5 Punctured 길쌈부호기

높은 부호율을 갖는 길쌈부호의 구현은 Viterbi복호에서 k 값이 1보다 큰, 즉 예를 들어 3/4코드의 경우 trellis의 어떤 상태에서 다음 상태로 들어가는 경우 2^k 의 경로가 존재하게 되는데 이는 복호기의 하드웨어적인 복잡성을 유발하게 된다. 이런 문제를 피하기 위해서 부호율이 $1/n$ 길쌈부호를 변형시킴으로써, 부호율이 큰 $(n-1)/n$ 부호를 실현할 수 있다.

III. 컴퓨터 시뮬레이션 및 하드웨어 실험

3.1 컴퓨터 시뮬레이션 및 하드웨어 실험에 관한 개요

본 논문에서는 구속장이 6이고, 생성다항식이 (65, 57)인 길쌈부호기와 백색가우시안 잡음 채널하에서, Viterbi복호기를 IBM-PC상에서 시뮬레이션을 하였고, 또한 DSP코드에 필요한 데이터를 IBM-PC에서 삼선전자의 SSP1600 fixed point DSP칩의 emulator board인 SDEI600상으로 입출력 포트(port)를 통해 전송하는 프로그램을 작성하였다. 또한 Viterbi복호기를 SSP1600 DSP칩의 DSP코드를 통해 SDEI600상에서 구현하였고,

연성판정 비트의 수와 traceback depth는 가변할 수 있도록 설계하였으며 BER성능은 emulator board에서 IBM-PC로 복호된 데이터를 전송하여 계산하였다.

Punctured길쌈부호는 IBM-PC에서 Borland C를 이용 시뮬레이션 하였다.

3.2 IBM-PC와 Emulator Board의 입출력 방법

Emulator 제어 프로그램과 어셈블러에 연결된 emulator board는 PC의 외부 bus에 연결되어 있는데 여기서 emulator 제어 프로그램은 emulator board와 IBM-PC사이의 모든 가능한 신호들을 제공하고 사용자의 응용 프로그램으로부터 만들어지는 어셈블러, 즉 기계어 코드는 emulator 제어 프로그램에 의해서 down-load되고 실행된다. 특히 emulator board의 내부 ROM의 역할을 하는 SRAM, IBM-PC에서 SRAM으로 명령코드를 전달하기 위한 프로그램 데이터 buffer/latch, 전달되는 기계어 코드가 위치하게 될 SRAM의 위치를 지정하기 위한 어드레스 latch, emulator board의 동작을 제어하기 위한 제어 latch와 클럭(clock)을 제공하기 위한 몇가지 부분으로 구성된다.

표(1) Emulator Board에서 사용된 IBM-PC의 I/O포트(port)
Table(1) Input/output of IBM-PC used in emulator board.

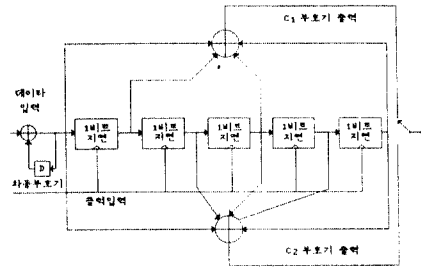
I/O Port	기능
280/2A0H 281/2A1H	IBM-PC와 emulator board 사이에서 어셈블러를 전달하기 위해 사용되는 데이터 포트 Up-load buffer, down-load latch
282/2A2H 283/2A3H	IBM-PC와 emulator board 사이에서 데이터 전달시 SRAM의 위치지정을 위한 어드레스 포트 Address-buffer, address-latch
284/2A4H	Emulator board 동작을 제어하기 위한 제어 port Control-latch
285/2A5H	SRAM의 writing을 제어하기 위한 제어 port
286/2A6H 287/2A7H	Emulator board와 외부의 word단위 데이터 전송시 사용되는 외부핀에 할당된 데이터 포트

3.3 길쌈부호기

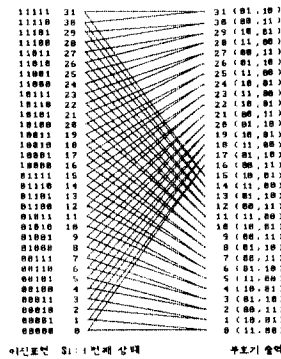
본 논문에서는 생성다항식이 (65, 57)인 (2,1)길쌈부호를 선택하였는데 이 부호의 minimum free distance는 8이고 generator polynomial은 2진으로 다음과 같이 표현된다[1].

$$\begin{aligned} 65_g &\Rightarrow (110101)_2 \\ 57_g &\Rightarrow (101111)_2 \end{aligned} \quad \text{식(7)}$$

이 길쌈부호기의 블럭도가 그림(2)에 그려져 있으며, 길쌈 부호의 상태수는 $2^{K-1} = 2^{6-1} = 32$ 가 되고, 시프트레지스터의 갯수는 5이며 구속장은 6이다. c_1 과 c_2 는 한개의 입력 데이터에 대한 부호기의 출력이며, 00, 01, 10, 11과 같은 비트의 쌍을 출력한다. 그림(3)은 trellis도를 나타내는데 여기서 알 수 있듯이 한 상태에서 나가는 가지수는 2개인데 부호기에서 입력 데이터가 0이면 아래의 가지에 해당되고, 1이면 위의 가지에 해당된다. 특히 좌측에서는 이전 32개의 상태에 대한 비트표현과 상태를 나타내고, 각각의 상태에서 나가는 가지들을 보이고 있다. 또한 우측에서는 다음의 상태와 부호기의 출력을 나타낸다.

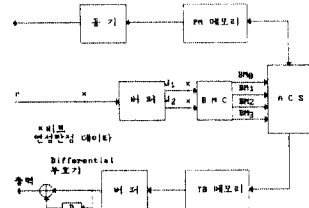


그림(2) 구속장이 6인 (2,1)길쌈부호기 블럭도
Fig. (2) Block diagram of (2,1) convolutional encoder with constraint length 6.



그림(3) 구속장이 6인 (2,1)길쌈부호기의 trellis도
Fig. (3) Trellis diagram of (2,1) convolutional encoder with constraint length 6.

3.4 Viterbi복호기



그림(4) Viterbi복호기의 블럭도
Fig. (4) Block diagram of Viterbi decoder.

3.4.1 입력 인터페이스(Input Interface)

3비트 또는 8비트로 연성판정된 복조기의 출력을 병렬로 생성하는 것인데, 이때의 입력 비율은 정보율의 1/R배에 해당한다. 본 논문에서는 3비트 또는 8비트 연성판정 데이터를 생성시키고, 이것에 대한 각각의 BER를 계산하였고, 연성판정 데이터를 위해 sign-magnitude표현을 사용하였다.

3.4.2 가지Metric 계산

연성판정 데이터 J_1 과 J_2 가 가지metric 계산소자에 입력된다. 그림(3)에서 볼 수 있듯이 상태수는 32 이고 과거 상태에서 다음 상태로의 가지는 2개씩 존재하며, 부호기에서 입력 데이터가 0이면 아래의 가지에 해당되고, 1이면 위의 가지에 해당한다. 이때 한개의 데이터에 의해 출력되는 2개의 심볼 c_1 과 c_2 가 존재하며 전송되는 2개의 심볼 c_1 과 c_2 가 그림(3)

우측면에서 볼 수 있다. 그러므로 복호기에서의 수신된 부호 비트 쌍은 3비트 양자화된 2개의 데이터 J_1 과 J_2 가 되는데 J_1 은 x_2, x_1, x_0 으로 J_2 는 y_2, y_1, y_0 로 표시할 수 있다.

가지metric의 계산은 아래 수식에서 알 수 있는데 본 논문에서는 연성판정의 데이터를 가변 가능하도록 설계하였기 때문에 x비트로 양자화 되었다고 가정하고, J_1 과 J_2 을 이용하여 아래의 네가지 가지metric 값을 계산하여야 한다. 여기서 부호화된 비트의 쌍이 00, 01, 10, 11인 각 경우의 가지metric인 $BM_0(t)$, $BM_1(t)$, $BM_2(t)$, $BM_3(t)$ 는 아래와 같다.

$$\begin{aligned} 00 &\Rightarrow BM_0(t) = (2^{x-1}-J_2) + (2^{x-1}-J_1) \\ 01 &\Rightarrow BM_1(t) = (2^{x-1}-J_2) + J_1 \\ 10 &\Rightarrow BM_2(t) = J_2 + (2^{x-1}-J_1) \\ 11 &\Rightarrow BM_3(t) = J_2 + J_1 \end{aligned} \quad \text{식(8)}$$

3.4.3 ACS (Add-Compare-Select)와 경로Metric 메모리

ACS는 가지metric계산단에서 계산된 네개의 가지 metric과 현재의 경로metric을 이용하여 새로운 경로metric을 계산한다. 그런데 경로metric의 갯수는 구속장 또는 길쌈부호의 메모리수에 의해서 $2^{K-1} = 2^m$ 으로 결정된다.

본 논문에서 구현하려는 Viterbi복호기는 구속장이 6 즉 메모리의 갯수가 5이므로 경로metric의 갯수가 32개가 된다. 따라서 시간 t에서 경로metric은 $PM_0(t), PM_1(t), PM_2(t), \dots, PM_{31}(t)$ 와 같이 표현될 수 있다. 현재의 경로metric 값 및 네개의 가지metric 값들로부터 32개의 새로운 경로metric은 다음과 같은 수식에 의해서 구한다.

$$\begin{aligned} PM_{i3}(t) &= \text{Min} \{ PM_{i1}(t-1) + BM_1(t), PM_{i2}(t-1) + BM_2(t) \} \\ PM_{i4}(t) &= \text{Min} \{ PM_{i1}(t-1) + BM_2(t), PM_{i2}(t-1) + BM_1(t) \} \\ PM_{j3}(t) &= \text{Min} \{ PM_{j1}(t-1) + BM_0(t), PM_{j2}(t-1) + BM_3(t) \} \\ PM_{j4}(t) &= \text{Min} \{ PM_{j1}(t-1) + BM_3(t), PM_{j2}(t-1) + BM_0(t) \} \end{aligned} \quad \text{식(9)}$$

여기서 i_1, i_2, i_3, i_4 와 j_1, j_2, j_3, j_4 는 다음과 같은 관계를 갖고 있는 것으로 생각할 수 있다. 즉 (i_1, i_2) 와 (j_1, j_2) 는 각각 $2P, 2P+1$ 이고, (i_3, i_4) 와 (j_3, j_4) 는 각각 $P, P+1$ 인데 여기서 P는 상태의 위치에 해당된다. 이렇게 계산되는 경로metric은 항상 가지metric값이 더해지기 때문에 계속 증가하는 값이 된다. 따라서 한차례의 계산이 끝나면 경로metric의 값을 정규화 해 주어야 한다. 즉 32개의 경로metric 값 중에서 가장 작은 값을 찾아내어 모든 32개의 경로metric 값에서 빼주어서 항상 가장 작은 값이 0이 되도록 한다. 즉 다음과 같이 표현할 수 있다.

$$PM_{\text{min}} = \text{Min}_i PM_i(t) \quad \text{식(10)}$$

윗식과 같이 경로metric 값중 가장 작은 state i 를 찾아서 아래의 식과 같이 경로metric을 정규화 한다.

$$PM_i(t) \leftarrow PM_i(t) - PM_{\text{min}} \quad \text{식(11)}$$

3.4.4 Traceback 메모리

앞에서 언급했던 것처럼 ACS단에서 생존자 가지에 대한 포인터(pointer)가 traceback 메모리에 저장되어 있으며, 그 가지에 대한 포인터는 그 가지가 $(2P, 2P+1)$ 중 어디에서 도래되었는가를 나타낸다. 앞에서 언급했던 traceback 길이는 하드웨어로 32인 경우로 잡았을때 생존자(survivor) 경로가 32개 존재하므로 32×32 RAM이 필요하다. 이 traceback길이 만큼

상태가 진행됨으로써 그 이전 상태에 대한 생존자 가지에 대한 포인터(pointer)가 traceback 메모리에 제거되면서 계속 진행하게 된다.

3.4.5 RAM Bank의 할당과 Viterbi복호기의 초기화

1) RAM bank 할당

Viterbi복호기를 DSP칩을 이용하여 구현하는 경우에 필요한 데이터의 메모리 및 RAM bank의 할당은 다음과 같다. 가장 큰 양의 메모리를 차지하는 것은 traceback depth의 데이터인 포인터를 저장하는 메모리이다. RAM0는 256개의 16비트 word가 있는데, 메모리 포인터를 여기에 저장하였다. 이로인해 traceback depth는 최대 128까지 가능하다. 역시 다음으로 메모리를 필요로 하는 것은 경로metric 메모리인데 $(2, 1, 5)$ 코드는 상태의 수가 32이기 때문에 32 word가 필요하게 된다. 또한 ACS전 임의의 경로metric 메모리를 저장하기 위해 32 word가 필요하게 된다. 위에서 언급한 경로metric 메모리는 RAM에 저장하였고, 양자화된 데이터, 가지metric 메모리, 8개의 복호된 데이터, 입력 데이터 모두 역시 RAM에 저장하였다.

2) Viterbi복호기의 초기화

Viterbi복호기를 구현하기 위해 필요한 데이터로 RAM bank를 초기화하는 것인데 이 과정에서 RAM bank1에 저장되는 것은 traceback depth값, 경로 metric값, 최대 양자화 값과 상태수 값등이다.

3.5 Punctured 길쌈부호

Punctured 길쌈부호의 Viterbi복호는 가짜비트(dummy bit)를 부호화된 비트가 전송측에서 제거된 위치에 삽입하고, 가짜비트에 대한 가지metric계산을 금지하게 하여 앞장에서 언급한 Viterbi 복호과정용 그대로 실현한다. 본 논문에서는 컴퓨터 모의실험을 통해 여러가지의 punctured 길쌈부호를 구현 실험 하였다.

IV. 실험 결과

4.1 컴퓨터 시뮬레이션에 의한 결과

본 논문에서는 punctured 길쌈부호에 대해서는 부호기, 채널과 Viterbi복호기등을 컴퓨터 시뮬레이션에 위해서 구현하였는데, 이때 사용한 punctured 길쌈부호의 발생기와 이들 각각에 대한 BER가 10^{-5} 에서의 부호이득이 표(2)에 제시되어 있다.

4.2 DSP칩을 이용한 Viterbi복호기의 하드웨어 실험 결과

본 논문에서는 삼성전자의 DSP칩인 SSP1600의 DSP코드를 이용하여 구속장이 6인 $(2, 1)$ 길쌈부호기의 연성판정 Viterbi복호기를 emulator board SDE1600상에서 구현하여 3비트와 8비트 연성판정데이터를 이용하여 BER를 계산하였다. 또한 traceback depth에 따른 BER를 나타내고, threshold spacing대 BER를 구하여 α 값을 최적화시켰다. 이에 대한 결과는 그림(5), (6)과 (7)에 나타내었다.

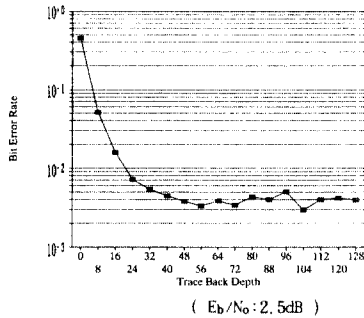
실험 결과 3비트 및 8비트 연성판정시 부호이득은 각각 4.9와 5.1dB를 얻었다.

표(2) 2/3~7/8 Punctured 길쌈부호의 제거비트 매트릭스, 최소자유거리, 및 부호이득

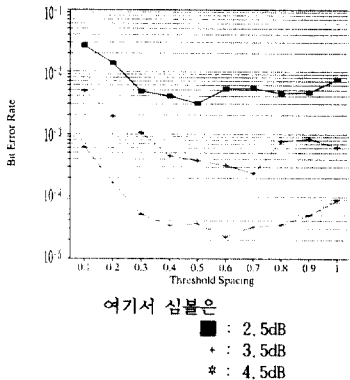
Table 4-1 The deleting bit matrix, minimum free distance, and coding gain of 2/3~7/8 punctured convolutional code.

코드	제거비트	d_{free}	부호이득
2/3	10 11	6	3.1
3/4	100 111	4	2.45
4/5	1000 1111	4	1.8
5/6	10000 11111	4	1.5
6/7	110110 101001	3	1.2
7/8	1011101 1100010	3	1.1

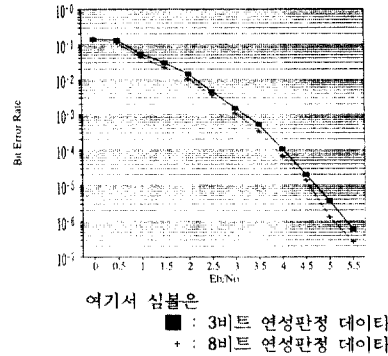
제거비트: 구속장 6인 경우 최고의 BER을 갖는 제거비트 매트릭스
 d_{free} : 최소자유거리



그림(5) DSP칩을 이용하여 구현한 구속장이 6인 1/2 길쌈부호에 대한 Viterbi복호기의 Traceback Depth에 따른 BER
 Fig. (5) The BER performance vs. traceback depth of Viterbi decoder implemented by DSP chip for 1/2 convolutional code with constraint length 6.



그림(6) DSP칩을 이용하여 구현한 구속장이 6인 1/2 길쌈부호에 대한 Viterbi복호기의 임계값 α 대 BER
 Fig. (6) The BER performance vs. threshold spacing α of Viterbi decoder implemented by DSP chip for 1/2 convolutional code with constraint length 6.



여기서 삼봉은
 ■ : 3비트 연성판정 데이터
 + : 8비트 연성판정 데이터

그림(7) DSP칩을 이용하여 구현한 구속장이 6인 1/2 길쌈부호에 대한 Viterbi복호기의 3비트 및 8비트 연성판정에 따른 E_b/N_0 대 BER
 Fig. (7) The BER performance vs. E_b/N_0 according to the 3bit and 8bit soft decision of Viterbi decoder implemented by DSP chip for 1/2 convolutional code with constraint length 6.

V. 결론

본 논문에서는 길쌈부호기와 Viterbi복호기의 알고리즘을 논하였고 3비트와 8비트 연성판정 양자화 및 양자화 임계값에 대해 α 를 변수로 언급하였다. DSP칩을 이용하여 구속장이 6, 부호율이 1/2이고 생성다항식이 (65, 57)인 길쌈부호의 복호기인 Viterbi복호기를 구현하였고, 역시 컴퓨터 시뮬레이션을 통해 각 punctured 길쌈부호에 대한 Viterbi복호기의 BER를 구하였다. 부호이득은 부호화를 실현하지 않은 PSK의 BER가 10^{-5} 인 경우, E_b/N_0 [dB]가 9.6이 되는데 이를 기준잡아 각 부호의 부호이득을 얻었으며 1/2길쌈부호에 대해서는 약 5.1dB의 부호이득을 얻었다.

< 참고 문헌 >

- [1] S. Lin and D. J. Costello Jr., *Error control coding, fundamentals and applications*, ch.10.11.12, Prentice-Hall.
- [2] J. A. Heller and I. M. Jacobs, "Viterbi decoding for satellite and space communication," *IEEE Trans.*, vol.COM-19, pp.835-848, Oct. 1971.
- [3] A. J. Viterbi, "Convolutional codes and their performance in communication system," *IEEE Trans.*, vol.COM-19, pp.751-772, Oct. 1971.
- [4] Y. Yasuda, Y. Hirata and A. Ogawa, "Theoretical bit error rate performance of soft decision Viterbi decoding," Paper of Technical Group on Communication Systems IECE of Japan, CS80-126, pp. 31-36, Sept. 1980.
- [5] Y. Yasuda, Y. Hirata and A. Ogawa, "Optimum soft decision for Viterbi decoding," *5th International Conference on Digital Satellite Communications*, pp.251-258, Genoa, Italy, March 1981.
- [6] Y. Yasuda, K. Kashiki and Y. Hirata, "High-rate punctured convolutional codes soft decision Viterbi decoding", *IEEE Trans. Commun.*
- [7] G. D. Forney Jr., "The Viterbi algorithm," *Proc. IEEE*, vol.61, pp.268-278, Mar. 1973.