

論文 92-29A-3-3

임펄스잡음 환경하에서의 TCM 방식 최적화에 관한 연구

(A Study on the Optimization of TCM System in the Impulsive Noise Environment)

高 晟 燦*, 宋 在 哲**, 崔 炯 辰*

(Sung Chan Ko, Jae Chul Song, and Hyung Jin Choi)

要 約

본 논문에서는 Monte-Carlo 기법에 의하여 개발된 시뮬레이터를 이용하여 Coded 8PSK TCM 시스템의 성능을 평가하였다. Gauss 잡음 채널과 Gauss 및 Impulse 잡음이 혼재하는 채널을 시스템의 채널모델로 사용하였다. TCM 시스템에 대한 S/W 시뮬레이터 개발에 필요한 기본 사항을 언급하였고 시뮬레이션 결과를 통하여 최적이라 사료되는 복호기억장을 결정하였다.

Abstract

In this paper, the performance evaluation of coded 8PSK TCM system is presented which is evaluated on the basis of the software simulator developed by Monte-Carlo simulation method. The Gaussian noise as well as Gaussian/Impulsive noise are used for the system channel. The required basic concepts for developing a S/W simulator applicable to TCM system are presented and the decoding memory length considered as optimal is determined.

I. 서 론

현대의 통신 시스템에서는 급격히 증가한 통신량으로 인하여 오률향상의 측면 만큼이나 통신채널의 대역제한성(Bandlimit)도 중요해졌다 하겠다. 시스템의 오률향상을 위한 목적으로 Coding 기법의 적용을 생각할 수 있는데, Coding 기법에서는 에러사건(Error Event)의 검출 및 정정을 위하여 항장비트(Redundancy Bit)의 추가가 불가피하다. 비록 추가된 항장비트는 오률향상을 위한 중요한 정보가 되지만,

이로 인하여 전송대역당의 유효 데이터 전송률 뿐만 아니라 SNR에 손실이 발생한다는 것은 Coding계에 있어서 하나의 자명한 원리로서 인식되어 왔다.¹⁾ 유효 데이터 전송률의 손실은 전송대역의 확장을 의미하므로 대역제한된 채널에서는 Coding 기법의 적용이 사실상 어려운 실정이다. 그러나 1982년, Ungerboeck에 의하여 대역의 확장없이 Coding기법에 기초하여 오률을 정정함으로써 대역제한된 채널에도 쉽게 적용할 수 있는 TCM(trellis coded modulation, 격자부호변조)이 제안되었다.²⁾ 이후 현재까지 많은 연구 결과들이 발표되었으며 연구 결과들에 의하면, 기본적으로는 구속장(Constraint Length)의 함수로 주어지는 Euclidean Distance를 결정함으로써 그 시스템의 개략적인 성능예견이 가능하다.³⁾ 복호과정의 매우 복잡한 메카니즘 때문에 아직까지는 TCM 시스템에 대하여 제안된 정확한 오률수식은 없고, 단지

*正會員, 成均館大學校 電子工學科
(Dept. of Elec. Eng., Sungkyunkwan Univ.)

**正會員, 仁德工業專門大學 事務自動學科
(Dept. of O. A., Induk Technical College)

接受日字: 1992年 1月 14日

오류의 상한·하한수식 정도만 제안되어 있는 실정이다. 따라서 채널환경(Gauss, Impulse, Fading, Satellite Transponder 등) 및 필터·증폭기 따위의 주요소자를 포함하는 실제의 시스템 설계에 있어서, 현재까지의 연구 결과에만 의존하여 구속장(Constraint Length)이나 복호 기억장(Memory Length) 등의 파라메타를 결정하기에는 미흡한 점이 많다 하겠다.

본 논문에서는 S/W 시뮬레이터에 의한 성능평가가 예견되는 시점에서, TCM 시스템의 성능 평가를 위한 프로그램 개발에 필요한 기본적인 이론을 논하고 시뮬레이션 결과를 토대로 하여 TCM 시스템의 성능을 분석하였다. 단 본 논문에서는 채널 환경만을 고려하여 서술하였으므로 필터링이나 비선형 증폭에 따른 성능 검토는 제외되었다.

II. 이론적 배경

여러 시뮬레이션 방법들 중에서 Coding 시스템에 Monte-Carlo [MC] 기법이 가장 적합하므로 본 논문에서는 MC기법에 의한 프로그램 개발을 전제로 한다.^[5] 2장에서는 그림1에 나타난 TCM 시스템의 구성도를 기초로 하여 각 블록에서 고려해야 할 이론적 배경을 논한다.

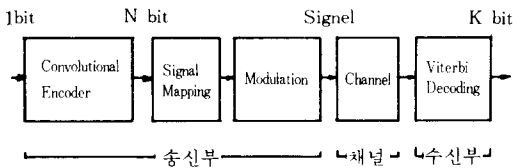


그림 1. TCM 시스템의 구성도
Fig. 1. Block diagram of TCM system.

1. 부호기

Euclidean Distance[ED]는 구속장 K의 함수이고 또 구속장은 부호기에서 사용하는 레지스터의 수에 의하여 결정되므로 레지스터의 수를 결정하는 문제는 TCM 시스템의 성능을 좌우하는 중요한 문제이다. K가 크면 클수록 오류가 향상되는 것은 자명한 사실이지만 Maximum Likelihood Path를 결정하는 복호회로가 K에 따라 지수함수적으로 복잡해지기 때문에 이들간의 신중한 절충(Tradeoff)이 요구된다. K(=1, 3, 4, 5, 6)에 따른 Coded 8PSK 시스템의 오류를 3장에 제시하였다. 그리고 결선 벡터(Connection Vector) 즉 생성함수에 따라 시스템의

성능에 차이가 있을 수 있는데, 이 문제는 두 가지의 측면-병렬천이, 비병렬천이-에서 해석할 수 있다. 생성함수의 결정에 있어서 가장 중요한 것은 전송로 상에서 발생한 유한 심벌에러로 인하여 무한에러를 유발시키는 이른바 catastrophic error propagation [CEP]을 방지해야 한다는 것이다. 생성함수들간에 공통인자가 없도록 결선함으로써 CEP 문제를 해결할 수 있다.^[6] CEP 문제를 해결한 부호기를 사용할 때 병렬천이 시스템의 ED보다 비병렬천이 시스템의 ED를 더 크게할 수 있다. 3장에서 병렬 및 비병렬 시스템의 성능을 비교 검토한다. 한편 주어진 입력 비트수 I에 대하여 출력비트수 N이 I+1이 되도록 하여 채널신호집합을 두배로할 때 채널용량이 최대가 된다는 것이 입증되어 있으므로 Rate=I/(I+1)인 시스템의 설계가 바람직하다 하겠다.^[2]

2. 신호 매핑 (Mapping)

부호기 출력을 채널신호에 적절히 매핑시킴으로써 주어진 시스템에 대한 ED를 최대화 할 수 있다. ED를 최대화하기 위한 매핑규칙이 바로 Ungerboeck에 의해 제안된 Set Partition 규칙이다. Set Partition 규칙을 정리하면 다음과 같다. 동일한 상태노드에서 출력되는 2개의 출력 Branch상에 있거나 동일한 상태노드로 입력되는 2개의 입력 Branch상에 있는 비트열의 집합(집합원소는 2개임)을 원래의 신호집합에서 처음으로 파생된 두개의 부집합 중 하나에 매핑시킨다. 병렬천이한 입·출력 Branch상에 있는 비트열의 집합(집합원소는 2개임)은 2개의 최소 부집합들 중 하나에 매핑시킨다. 단 ED를 증가시키는 방향으로 집합을 세분화 한다. 그림2를 기초로 하여 매핑규칙을 살펴본다. 동일한 상태노드에서 입·출력되는 4개의 Branch상에 있는 부호열은 다음의 두

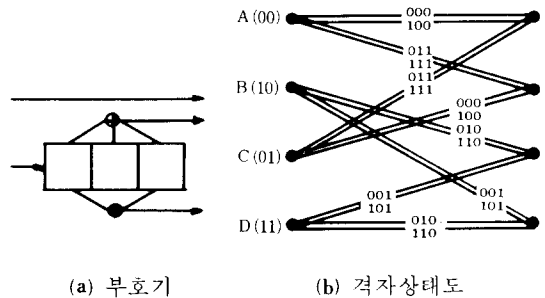


그림 2. Coded 8PSK 시스템의 부호기 및 상태도
Fig. 2. Encoder & state diagram of coded 8PSK system.

부집합 중 하나에 속하게 된다.

$$BO = \{000, 100, 011, 111\}, B1 = \{010, 110, 001, 101\}$$

병렬천이 Branch상에 있는 부호열은 다음의 네 부집합 중 하나에 속하게 된다.

$$\begin{aligned} C0 &= \{000, 100\}, & C1 &= \{011, 111\}, \\ C2 &= \{010, 110\}, & C3 &= \{001, 101\} \end{aligned}$$

따라서 ED를 증가시키는 방향으로 신호집합을 세분화하면 각 부호열에 대하여 채널신호가 일대일로 매핑된다(그림3 참조).

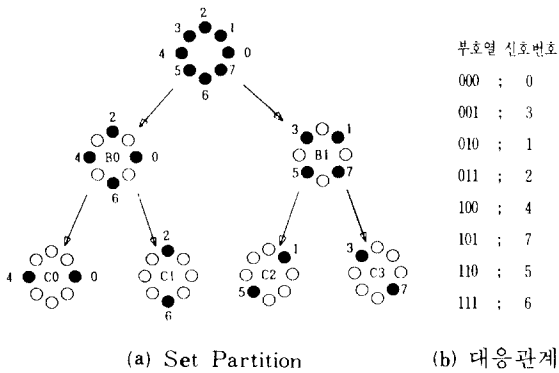


그림 3. 신호 매핑
Fig. 3. Channel signal mapping.

3. 채널

MC 기법에 의한 시뮬레이터에서는 잡음의 처리가 시스템의 오물산정에 중요한 영향을 끼치기 때문에 매우 신중한 잡음처리가 요구된다. 먼저 Gauss 잡음에 대한 처리과정을 논하면 다음과 같다. 0과 1사이의 균등난수 $h_1, h_2, \dots, h_n, \dots$ 에 대한 평균 및 분산은 다음과 같다.

$$m = \int_0^1 h \, dh = 1/2 \tag{1}$$

$$v^2 = \int_0^1 (h - 1/2)^2 \, dh = 1/12 \tag{2}$$

그리고 식(3)과 같이 N개의 균등난수의 합으로 결정되는 난수 $g_1, g_2, \dots, g_n, \dots$ 들은

$$g_i = \frac{1}{N} (h_1 + \dots + h_{i+N-1}) \tag{3}$$

중심극한 정리에 의하여 N이 클 때, 평균 $m=1/2$ 이고 분산 $\sigma^2 = v^2/N = 1/(12*N)$ 인 정규분포로 볼 수

있다. 따라서 $N=12$ 로 했을 때 식(4)에 의하여 결정되는 난수 x_i ,

$$x_i = \sqrt{(12*N)} \cdot (g_i - 1/2) = (h_1 + \dots + h_{i+N-1}) - 6 \tag{4}$$

x_2, \dots, x_n, \dots 들은 평균이 0이고 분산이 1인 표준정규분포를 가진다. Gauss 잡음은 잡음전력을 P_n 라할 때 평균이 0이고 표준편차가 $\sqrt{P_n}$ 인 표준정규분포를 가지므로 식(4)에 $\sqrt{P_n}$ 을 곱하여 Gauss 잡음 $GN(t)$ 을 얻어낼 수 있다.

$$GN(t) = \sqrt{P_n} \cdot [\sum_{i=1}^{12} h_i(t) - 6] \tag{5}$$

I·Q채널에 대한 각각의 잡음전력은 신호전력을 1로 정규화 했을 때 입력된 SNR[dB]로부터 다음과 같이 계산된다.

$$P_n = 0.5 \cdot 10^{-(SNR/10)} \tag{6}$$

따라서 표준편차를 $\sqrt{P_n}$ 로 해서 I·Q 채널에 대하여 각각 독립적인 Gauss 잡음생성 루틴을 Call함으로써 두 잡음성분을 얻어낼 수 있다.

한편 Impulse 잡음은 문헌 [7]에 주어진 확률분포를 이용하여 모델링할 수 있으며 Impulse 잡음을 I 채널과 Q채널에 대한 Joint Probability Density로 나타내면 식(7)과 같다.

$$P_1(N) = \frac{Exp(-A)}{W} \sum_{j=0}^{\infty} [A^j \cdot N] / [j! \cdot \sigma_j^2] \tag{7}$$

- W : Gauss 잡음전력 + Impulse 잡음전력
- A : 단위시간당 존재하는 Impulse 잡음의 시간비
- σ_j^2 : $(j/A + \Gamma) / (1 + \Gamma)$
- Γ : Impulse 잡음에 대한 Gauss 잡음의 전력비

신호전력을 1로 정규화 했을 때 Gauss 잡음에 대한 신호대 잡음비가 5dB인 Gauss 잡음과 $A=0.0025$ 이고 $\Gamma=0.01$ 인 Impulse 잡음이 혼재하는 경우의 파형을 그림 4에 나타내었다. 그리고 그림5에는 Gauss 잡음에 대한 신호대 잡음비가 5dB일 때 $A=0.1, \Gamma=0.01$ 인 Impulse 잡음과 $A=0.01, \Gamma=0.1$ 인 Impulse 잡음이 혼재하는 경우에 대한 확률분포를 Log Scale로 나타내었다.

3장에서는 Impulse 잡음 ($A=0.1, \Gamma=0.01$ & $A=0.01, \Gamma=0.1$)과 Gauss 잡음이 혼재하는 채널하에서의 시뮬레이션 결과를 제시하였다.

4. 복호기

복소수로 신호를 처리할 때 수신된 신호 RS는 식(8)과 같다.

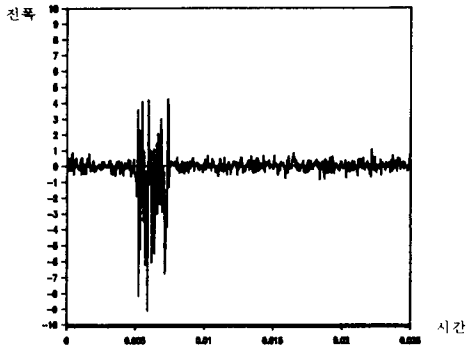


그림 4. Gauss 및 Impuse 잡음에 대한 파형
Fig. 4. Waveform of Gaussian/Impulsive noise.

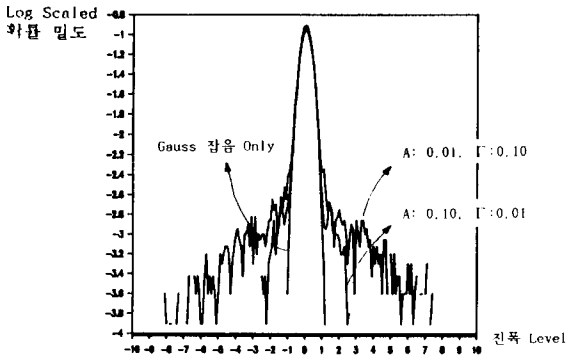


그림 5. Log Scale로 나타낸 잡음의 확률분포
Fig. 5. Log scaled probability density of noise.

$$RS = \text{COMPLEX}(SR + GNR, SI + GNI) \quad (8)$$

여기서 SR, SI는 각각 매핑된 신호의 I채널, Q채널 성분을 의미하며 GNR, GNI는 Gauss 잡음 생성루틴의 출력값으로 각각 I채널, Q채널의 잡음성분을 의미한다. 한편 Backward 방식의 Viterbi 복호 알고리즘을 기초로 할 때,^[8] 복호과정에서 필요한 정보는 각 상태노드에서 Survivor가 되는 Branch의 Branch Number와 각 상태노드의 Metric 들 중 최소의 Metric을 가지는 상태노드에 대한 상태노드 Number이다. 이러한 정보를 산출하는 방법을 그림2에 나타낸 시스템에 대하여 요약하면 다음과 같다. 시간 $t=k$ 일 때 $s(=0, 1, 2, 3)$ 상태노드에 입력되는 4개의 입력 Branch에 대한 Branch Metric $BM_{s,k}(i)$ 는 식(9)로 부터 얻을 수 있다.

$$BM_{s,k}(i) = [\text{Real}(REF(i)) - \text{Real}(RS)]^2 + [\text{Imag}(REF(i)) - \text{Imag}(RS)]^2 \quad (9)$$

여기서 $i(=0, 1, 2, 3)$ 는 Branch Number이고 $REF(i)$ 와 RS 는 각각 i Branch에 대한 복소 기준신호와 수신된 복소신호를 의미한다. 그리고 상태노드 s 에서의 Path Metric $PS_{s,k}(i)$ 는 식(10)으로 부터 얻을 수 있다.

$$PS_{s,k}(i) = BM_{s,k}(i) + PS_{s',k-1} \quad (10)$$

여기서 $s'(=0, 1, 2, 3)$ 는 $t=k-1$ 일 때의 상태노드 Number이며 i 와 s 에 의해 식(11)과 같이 규정된다.

$$\begin{cases} s'=0 & \text{for } i=0, 1 \text{ and } s=0, 1 \\ & 2 \text{ for } i=2, 3 \text{ and } s=0, 1 \\ s'=1 & \text{for } i=0, 1 \text{ and } s=2, 3 \\ & 3 \text{ for } i=2, 3 \text{ and } s=2, 3 \end{cases} \quad (11)$$

또한 Survivor에 대한 누적 Metric $PS_{s',k-1}$ 는 식(12)로 부터 얻을 수 있다.

$$PS_{s',k-1} = \min[PS_{s',k-1}(i)] \quad (12)$$

이러한 절차에 의해서 복호에 필요한 정보를 얻은 후, 얻은 정보를 복호메모리에 저장한다. 따라서 기억장을 M 이라할 때 $t=k+M$ 에서 복호메모리에 저장된 내용은 표1과 같다.

표 1. 복호메모리에 저장된 정보

Table 1. Content of decoding memory.

$PS_{0,k}$	$PS_{0,k+1}$	·	$PS_{0,k+M-1}$	$PS_{0,k+M}$
$PS_{1,k}$	$PS_{1,k+1}$	·	$PS_{1,k+M-1}$	$PS_{1,k+M}$
$PS_{2,k}$	$PS_{2,k+1}$	·	$PS_{2,k+M-1}$	$PS_{2,k+M}$
$PS_{3,k}$	$PS_{3,k+1}$	·	$PS_{3,k+M-1}$	$PS_{3,k+M}$

(a) survivor에 대한 누적 metric

$B_0(k)$	$B_0(k+1)$	·	$B_0(k+M-1)$	$B_0(k+M)$
$B_1(k)$	$B_1(k+1)$	·	$B_1(k+M-1)$	$B_1(k+M)$
$B_2(k)$	$B_2(k+1)$	·	$B_2(k+M-1)$	$B_2(k+M)$
$B_3(k)$	$B_3(k+1)$	·	$B_3(k+M-1)$	$B_3(k+M)$

(b) survivor에 대한 branch number

여기서 $B_s(t)$ 는 시간 t 에서 상태노드 s 의 Path Metric들 중 최소의 값이 되는 Path에 있는 Branch Number를 의미한다. 이러한 정보를 기초로 하여 Backward Viterbi 복호하는 방식을 언급하면 다음과 같다. $t=k+M$ 에서 4개의 $PS_{s,k+M}$ 들 중 그 값이 최

소가 되는 상태노드 $s_{\min}(k+M)$ 에 기억된 Branch Number $B_{s,k+M}$ 를 이용하여 식(13)의 규칙에 의해 $t=k+M-1$ 에서의 Survivor 상태노드를 찾는다.

$$s_{\min}(k+M-1) = 2 * \{ B_{s,k+M} / 2 \} + \{ s_{\min}(k+M) / 2 \} \quad (13)$$

이 과정을 기억장만큼 반복하여 $s_{\min}(k)$ 를 찾고 상태노드 $s_{\min}(k)$ 에 기억된 $B_{s,k}$ 의 값에 따라 식(14)의 규칙으로 전송된 데이터를 복호한다.

$$\begin{aligned} [B_{s,k}=0 \text{ 또는 } 2] \ \& \ [s_{\min}(k)=0 \text{ 또는 } 2] \text{일때 } 00 \text{을 복호} \\ [B_{s,k}=0 \text{ 또는 } 2] \ \& \ [s_{\min}(k)=1 \text{ 또는 } 3] \text{일때 } 01 \text{을 복호} \\ [B_{s,k}=1 \text{ 또는 } 3] \ \& \ [s_{\min}(k)=0 \text{ 또는 } 2] \text{일때 } 10 \text{을 복호} \\ [B_{s,k}=1 \text{ 또는 } 3] \ \& \ [s_{\min}(k)=1 \text{ 또는 } 3] \text{일때 } 11 \text{을 복호} \end{aligned} \quad (14)$$

이상의 과정을 기억장이 5인 시스템에 대하여 설명한다. 표2에 시간에 따른 부호기 입·출력값 및 매핑된 채널 신호를 나타내었고 표3에 Branch Metric 과 Path Metric을 나타내었다. 그리고 복호 메모리에 저장된 내용을 표4에 나타내었다.

여기서 이중사각은 상태노드 s_{\min} 에 기억된 Branch Number를 나타내고 점선사각은 식(13)에 의하여 Backward 방식으로 결정되는 $s_{\min}(k+M-i)$ 에서의 Branch Number를 나타낸다. $k=5$ 에서 $PS_{s,k+5}$ 들 중 최소가 되는 것은 $PS_{1,k+5}$ 이므로 $s_{\min}(5)$ 는 1이 된다. $B_{s,5}=2$ 이고 $s_{\min}(5)=1$ 이므로 $s_{\min}(4)=2$ 가 되며 $B_{s,4}=1$ 이고 $s_{\min}(4)=2$ 이므로 $s_{\min}(3)=1$ 이 된다. 이렇게 반복하여 $s_{\min}(1)=1, B_{1,1}=0$ 을 구한 후 식(14)에 규정된 규칙을 적용하면 01을 복호할 수 있다. 다음은 $k=6$ 일때 $s_{\min}(6)$ 은 3이고 $B_{s,6}=0$ 이므로 $s_{\min}(5)$ 는 1이 되

표 2. 부호기 입·출력 및 채널신호의 관계
Table 2. Encoder input, output & channel signal.

시 간	1	2	3	4	5	6	...
입력 1	0	0	1	1	0	0	...
입력 2	1	0	1	0	1	1	...
부호기출력	011	010	100	110	000	001	...
채널신호	2	1	4	5	0	3	...
상 태	B	C	B	C	B	D	...

표 3. Branch metric과 path metric
Table 3. Branch metric & path metric.

s	Branch Metric	k=1	k=2	k=3	k=4	k=5	k=6
0	$BM_{0k}(0)$	A2	A1	A4	A3	A0	A3
	$BM_{0k}(1)$	A2	A3	A0	A0	A4	A1
	$BM_{0k}(2)$	A0	A0	A2	A3	A2	A1
	$BM_{0k}(3)$	A4	A3	A2	A0	A2	A3
1	$BM_{1k}(0)$	A0	A1	A2	A3	A2	A1
	$BM_{1k}(1)$	A4	A3	A2	A0	A2	A3
	$BM_{1k}(2)$	A2	A1	A4	A3	A0	A3
	$BM_{1k}(3)$	A2	A3	A0	A0	A4	A1
2	$BM_{2k}(0)$	A1	A0	A3	A4	A1	A2
	$BM_{2k}(1)$	A3	A4	A1	A0	A3	A2
	$BM_{2k}(2)$	A1	A2	A1	A2	A3	A0
	$BM_{2k}(3)$	A3	A2	A3	A2	A1	A4
3	$BM_{3k}(0)$	A1	A2	A1	A2	A3	A0
	$BM_{3k}(1)$	A3	A2	A3	A2	A1	A4
	$BM_{3k}(2)$	A1	A0	A3	A4	A1	A2
	$BM_{3k}(3)$	A3	A4	A1	A0	A3	A2

(a) branch metric

s	Path Metric	k=1	k=2	k=3	k=4	k=5	k=6
0	$P_{0k}(0)$	A2	B0	B5	B3	B0	B3
	$P_{0k}(1)$	A2	B3	B0	B0	B5	B0
	$P_{0k}(2)$	∞	∞	A2	B4	A2	B1
	$P_{0k}(3)$	∞	∞	A2	B1	A2	B4
1	$P_{1k}(0)$	A0	B0	B2	B3	B2	B0
	$P_{1k}(1)$	A4	B3	B2	B0	B2	B3
	$P_{1k}(2)$	∞	∞	A4	B4	A0	B4
	$P_{1k}(3)$	∞	∞	A0	B1	A4	B1
2	$P_{2k}(0)$	∞	A0	B4	A4	B1	A2
	$P_{2k}(1)$	∞	A4	B1	A0	B4	A2
	$P_{2k}(2)$	∞	∞	B0	B2	B3	B0
	$P_{2k}(3)$	∞	∞	B3	B2	B0	B5
3	$P_{3k}(0)$	∞	A2	B1	A2	B4	A0
	$P_{3k}(1)$	∞	A2	B4	A2	B1	A4
	$P_{3k}(2)$	∞	∞	B3	B5	B0	B2
	$P_{3k}(3)$	∞	∞	B0	B0	B3	B2

(b) path metric

여기서

$A0=0.000, A1=0.586, A2=2.000, A3=3.621, A4=4.000$

$B0=2.586, B1=3.172, B2=4.586, B3=5.621, B4=6.207, B5=6.586$ 이다.

표 4. 복호메모리의 내용
Table 4. Content of decoding memory.

	0	1	2	3	4	5	6
PS _{sk}	A0	A2	B0	A2	B0	A2	B0
PS _{1k}	∞	A0	B0	A0	B0	A0	B0
PS _{2k}	∞	∞	A0	B0	A0	B0	A2
PS _{3k}	∞	∞	A2	B0	A2	B0	A0

(a) survivor에 대한 누적 metric

	k=1	k=2	k=3	k=4	k=5	k=6
B _{0,k}	0	0	2	1	2	1
B _{1,k}	0	0	3	1	2	0
B _{2,k}	0	0	2	1	3	0
B _{3,k}	0	0	3	0	2	0

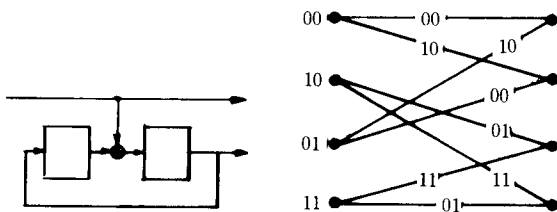
(b) survivor에 대한 branch number

이 복호과정이 앞서서의 과정과 같으므로 5번 Backward해서 $s_{\min}(2)=2, B_{2,2}=0$ 을 구한 후 식(14)의 규칙을 적용하면 00을 복호할 수 있다. 복호기에서의 복호기억장의 결정은 시스템성능에 중요한 영향을 끼칠 수 있으므로 복호기억장에 따른 시뮬레이션 결과를 3장에 제시하였고 결과를 검토하여 최적이라 사료되는 기억장의 범위를 논하였다.

III. 결과 및 검토

먼저 시뮬레이터의 정확성을 검토하기 위하여 문헌[9]에서와 같이 생성함수를 이용하여 상한치(Upper Bound)를 측정하는 방법으로 그림6에 주어진 Coded 4ASK 시스템의 이론치 BER을 구하였다.

주어진 신호집합에 대한 Error Weight와 Weight Profile은 표5와 같다.



(a) 부호기

(b) 격자 상태도

그림 6. Coded 4ASK 시스템
Fig. 6. Coded 4ASK system.

표 5. Error weight 및 error profile
Table 5. Error weight & error profile.

신호 Set	Error Weight				Weight Profile
	$E_p=00$	$E_p=01$	$E_p=10$	$E_p=11$	
S _A	0	0.8	3.2	7.8	2
S _B	0	0.8	3.2	0.8	2W ^{0.8}
S _C	0	0.8	3.2	0.8	2W ^{3.2}
S _D	0	0.8	3.2	7.2	W ^{7.2} +W ^{0.8}
S _A	0	0.8	3.2	7.2	2
S _B	0	0.8	3.2	0.8	2W ^{0.8}
S _C	0	0.8	3.2	0.8	2W ^{3.2}
S _D	0	0.8	3.2	7.2	W ^{7.2} +W ^{0.8}

두 신호집합(S_i, S_j)에 대한 Weight Profile은 서로 같고 문헌[9]의 정리1이 만족되므로 생성함수를 이용한 오률상한치 측정이 가능하다. Weight Profile을 포함한 상태도는 그림7과 같다.

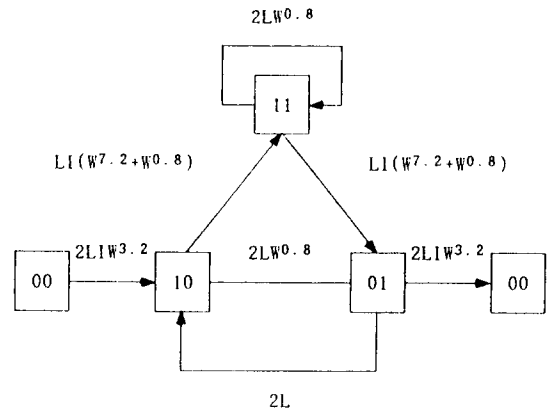


그림 7. Weight profile을 포함한 상태도
Fig. 7. State diagram with weight profile.

따라서 본 시스템의 오률 상한식은 식(15)와 같다.

$$P_b \leq \frac{1}{2} \text{Erfc}[\sqrt{d_f^2 E_s / 4N_0}] \cdot \text{Exp}[d_f^2 E_s / 4N_0] \cdot \frac{\partial T(W, L, I)}{\partial I} \Big|_{L=0.5, W=\text{Exp}[-E_s / 4N_0] \cdot I=1} \quad (15)$$

여기서 d_f^2 와 L은 각각 최소 ED와 부호기 입력부호의 사전확률을 의미하고 함수 T(W, L, I)는 그림7로 부터 식(16)으로 주어진다.

$$T(W, L, I) = \frac{8L^3 I^2 W^{2*3.2+0.8}(1-2LW^{0.8}) + 4L^4 I^2 W^{2*3.2}(W^{0.8} + W^{7.2})^2}{1-2LW^{0.8}(1+2L) - 2L[IL(W^{0.8} + W^{7.2})]^2 + 8L^3 W^{2*0.8}} \quad (16)$$

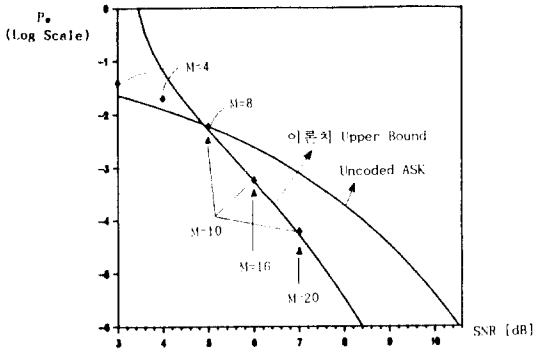


그림 8. 이론치 BER과 시뮬레이션 결과
 Fig. 8. Theoretical BER & simulation result.

이렇게 하여 구한 이론치 BER곡선과 본 시뮬레이터에서 산출한 BER 곡선을 그림8에 나타내었다.

그림8을 검토하면 생성함수를 이용한 BER계산은 Low SNR에서 그 한계가 드러나고 또 기억장에 대한 고려가 없음을 알 수 있다. 그러나 기억장을 무한대로 생각할 때 High SNR에서는 오를상한치 계산에 적용 가능하다. 그림8로 부터 본 시뮬레이터는 충분한 정확성을 가짐을 알 수 있다. 따라서 본 시뮬레이터를 통하여 산출한 결과들을 기초로하여 전장에서 논의한 사항들에 대한 근거를 제시한다. 먼저 그림9에 주어진 부호기들을 이용하여 구속장에 따른 성능차이를 구하였고 그 결과를 그림10에 나타내었다.

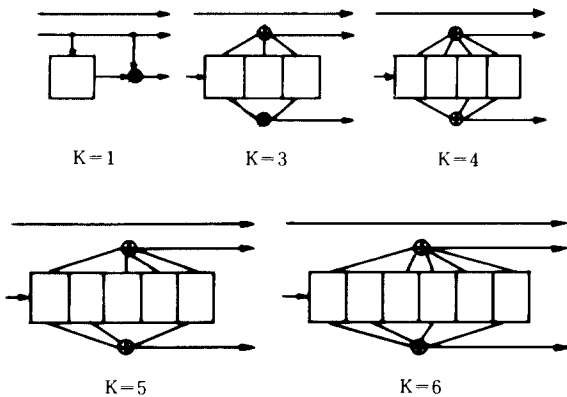


그림 9. 구속장에 따른 Coded 8PSK 시스템의 부호기
 Fig. 9. Encoder of coded 8PSK system with different constraint length.

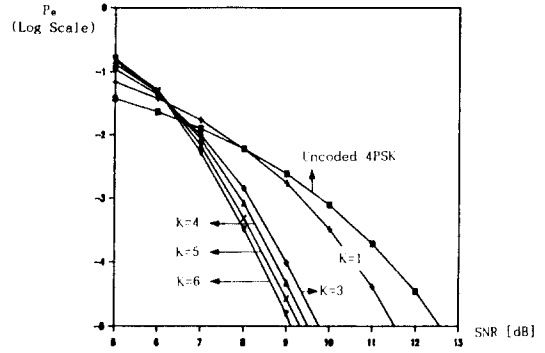


그림10. 구속장에 따른 오률
 Fig. 10. BER due to different constraint length.

BER=10⁻⁵를 기준으로할 때 K=1에서 약1dB, K=3에서 약 2.9dB, K=4에서 약 3.1dB, K=5에서 약 3.2dB 그리고 K=6에서 약 3.3dB 가량의 Coding Gain을 얻을 수 있다. 단 사용된 복호기억장은 구속장의 10배이다. K=4 이상인 시스템에 대하여 복호회로의 복잡성은 지수함수적으로 증가하는데에 반하여 Coding Gain의 차이는 두드러지지 않음을 알 수 있다. 그림11에는 Gauss 및 Impulse 잡음이 혼재하는 채널에서 구속장에 따른 오률을 나타내었다.

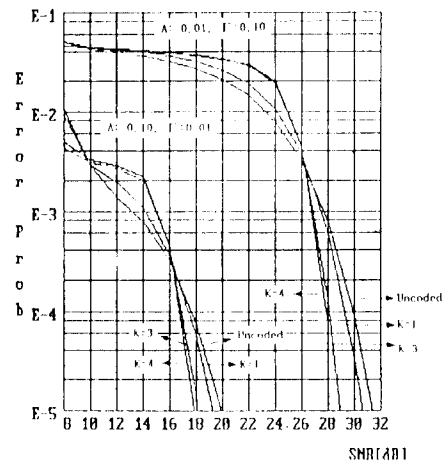


그림11. Gauss & Impulse 잡음하에서의 구속장에 따른 오률
 Fig. 11. BER due to different constraint length over Gaussian/Impulsive noise.

Impulse 잡음하에서도 구속장이 클 때 성능이 좋을 수 있으나 BER = 10^{-5} 를 기준으로 할 때 K = 3에서 약 2dB 가량의 Coding Gain을 보이므로 채널 환경이 나빠질 때 Coding에 의한 오률향상의 효과가 다소 약해짐을 알 수 있다.

다음은 그림12에 나타난 비병렬천이 시스템과의 성능차이를 검토하기 위하여 Gauss 잡음의 채널환경과 Gauss 및 Impulse 잡음이 혼재하는 채널환경에서의 시뮬레이션 결과를 각각 그림13, 그림14에 나타내었다. 그림13을 검토하면 Gauss 잡음하에서 High SNR로 가면서 비병렬천이 시스템의 성능이 병렬천이 시스템보다 조금 우수하지만 그림14를 보면 채널 환경이 나쁜 상태에서는 비병렬천이 시스템의 설계가 바람직한 것만을 알 수 있다.

다음은 복호기억장을 12로 고정시키고 그림9에 주어진 시스템들에 대하여 시뮬레이션한 결과를 그림15에 나타내었다. 그림15를 검토하면 복호기억장을 12로 고정한 경우에는 구속장이 클수록 오히려 성능이 저하됨을 알 수 있다. 따라서 각 구속장에 대하여 적절한 복호기억장을 결정할 필요가 있다. 그림

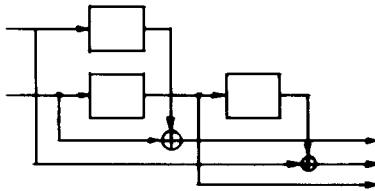


그림12. 비병렬천이 시스템의 부호기
Fig. 12. Encoder of nonparallel transition system.

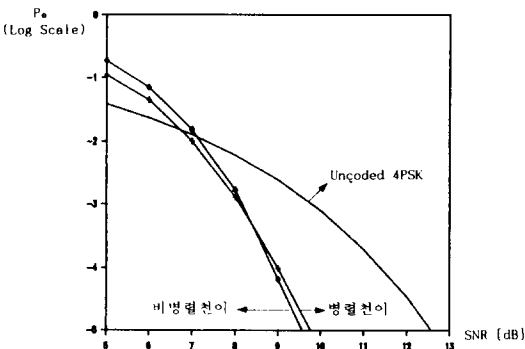


그림13. Gauss 잡음하에서의 병렬 및 비병렬천이 시스템의 오률
Fig. 13. BER of parallel & nonparallel transition system over Gaussian noise.

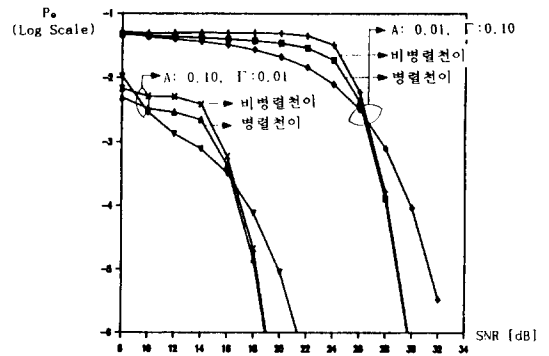


그림14. Gauss & Impulse 잡음하에서의 병렬 및 비병렬천이 시스템의 오률
Fig. 14. BER of parallel & nonparallel transition system.

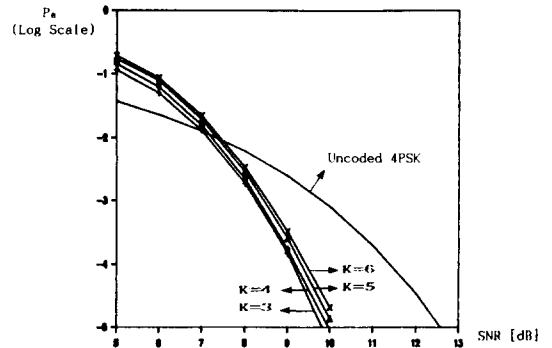


그림15. Gauss 잡음하에서 복호기억장이 12일때의 시스템 오률
Fig. 15. System BER when decoding memory length M = 12.

16은 SNR을 8dB로 고정시키고 복호기억장을 변화시키면서 시뮬레이션한 결과를 나타낸 것이고 이를 정규화시켜 그림17에 나타내었다.

그림17에서 알 수 있듯이 복호기억장을 구속장의 8배 이상으로 했을 때는 더 이상의 성능 향상이 없고 또 복호기억장이 구속장의 5배 이상이 되면 복호기억장을 무한대로 했을 때의 성능과 별 차이가 없으므로 최적의 복호기억장은 5K에서 8K사이라 사료된다.

그리고 MC 기법으로 10^{-6} 정도의 BER을 산출하기 위하여는 적어도 10^6 개 가량의 데이터를 처리하여야 비교적 정확한 평균 BER을 얻을 수 있다. 본 논문에 제시된 결과들은 Low SNR에서는 $2 * 10^6$ 개의 데이터를, Middle SNR에서는 $2 * 10^6$ 개의

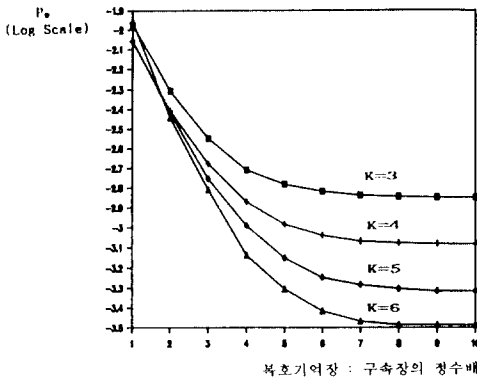


그림 16. Gauss 잡음하에서 복호기억장에 따른 오률변화
 Fig. 16. Deviation of BER due to decoding memory length.

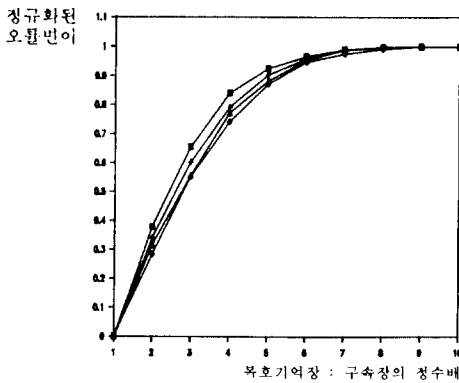


그림 17. Gauss 잡음하에서 복호기억장에 따른 정규화된 오률변화
 Fig. 17. Normalized deviation of BER due to decoding memory length.

데이터를 그리고 High SNR에서는 2×10^7 개 혹은 10^8 개의 데이터를 처리하여 얻은 결과들이다. IBM 486PC(33MHz Main Clock, with Coprocessor)를 이용하여 10^8 개의 데이터를 처리하는데 약 89시간이 소요되었다. 따라서 10^{-7} 이하의 BER은 MC 기법의 시간적 제한성 때문에 산출하지 못하였다.

IV. 결 론

본 논문에서는 Monte-Carlo 기법으로 개발된 시뮬레이터를 이용하여 Coded 8PSK TCM 시스템의 성능을 분석했다. Gauss 잡음 채널에 대하여 생성

함수를 이용해서 구한 오률상한치와 시뮬레이션을 통하여 얻은 결과를 비교 분석하여 시뮬레이터의 정확성을 확인하였다. 구속장이 서로 다른 Coded 8 PSK TCM 시스템의 오률을 검토하여 $K=3$ 이상일 때는 복호기의 복잡성에 비해 증가된 이득이 미약함을 알 수 있었다. 또 TCM 시스템의 Coding Gain은 채널특성에 따라 다르게 주어짐을 알 수 있었다. 그리고 병렬 및 비병렬천이 시스템의 성능을 비교 검토하였고 복호기억장이 시스템의 성능에 큰 영향을 끼침을 보였다. 최적의 복호기억장을 결정하기 위하여 복호기억장을 변수로 하여 시뮬레이션을 행한 결과를 제시하였으며, 구속장의 5배에서 8배사이로 복호기억장을 결정하는 것이 최적임을 알 수 있었다. 본 논문에서는 주로 Coded 8PSK TCM 시스템에 대한 성능을 평가하였지만 2장에 제시한 이론적 배경을 바탕으로 하면 임의의 TCM 시스템에 대한 시뮬레이션 프로그램도 작성할 수 있을 것으로 사료된다. 차후 보다 다양한 채널을 모델하고 비트당의 샘플수를 늘려 동기회로, 필터대역, 비선형 증폭에 따른 시스템의 민감도(System Sensitivity)를 살펴볼 계획이다.

參 考 文 獻

- [1] Shu Lin, "Error control coding : fundamentals and applications," Prentice Hall, Inc., Englewood Cliffs, N.J. 07632, 1983.
- [2] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 1, Jan. 1982.
- [3] S.G. Wilson, "Rate 5/6 trellis-coded 8-PSK," *IEEE Trans. Commun.*, vol. COM-34, no. 10, Oct. 1986.
- [4] D. Divsalar, "Trellis coding with asymmetric modulations," *IEEE Trans. Commun.*, vol. COM-35, no. 2, Feb. 1987.
- [5] 고성찬, 최형진, "분석적 기법에 의한 디지털통신 시뮬레이터에 관한 연구." 대한전자공학회 6월호, 1991.
- [6] J.L. Massey, "Inverse of linear sequential circuits," *IEEE Trans. Comput.*, vol. C17, Apr. 1968.
- [7] D. Middleton, "Statistical models of electromagnetic interference," *IEEE Trans. Electromag. Compat.*, vol. EMC-19, no. 3, Aug. 1977.
- [8] G.D.Forney, "The viterbi algorithm," *Proc. IEEE*, vol. 61, Mar. 1973.
- [9] E. Zehavi, "On the performance evaluation of trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 2, Mar. 1987.

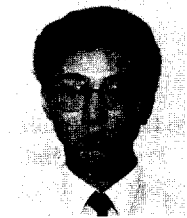
著 者 紹 介



高 晟 燦 (正會員)

1964年 7月 26日. 1987年 2月
성균관대학교 전자공학과 졸업
(학사). 1989年 2月 성균관대학
교 대학원 전자공학과 졸업(석사).
1992年 2月 성균관대학교 전자
공학과 박사과정 수료. 주관심분
야는 디지털 통신, 무선통신, 이동통신, 위성통신 등임.

●



宋 在 哲 (正會員)

1961年 9月 9日生. 1985年 2月
성균관대학교 전자공학과 졸업
(학사). 1989年 2月 성균관대학
교대학원 전자공학과 졸업(석사).
1992年 2月 성균관대학교 전자
공학과 박사과정 수료. 1991年 9
月~현재 인덕전문대학 OA과 전임강사. 주관심분야
는 디지털 통신, 위성통신, 디지털 동기 등임.



崔 炯 辰 (正會員)

1952年 8月 30日生. 1974年 2月
서울대학교 전자공학과 졸업(학
사). 1976年 2月 한국과학기술
원 전기전자공학과 졸업(석사).
1976年 3月~1979年 7月 주식회
사 금성사 중앙연구소 근무(연구
원). 1979年 9月~1982年 12月 미국 Universty of
Southern California 전기공학과 박사(Ph. D). 1982年
10月~1989年 2月 미국 LinCom Corp. 연구원으로
근무. 1989年 3月~현재 성균관대학교 전자공학과
근무(조교수). 주관심분야는 디지털통신, 무선통신,
이동통신, 위성통신 및 동기화이론을 포함한 Modem
기술 등임.