

論 文

Path Back 방식을 이용한 TCM의 복호 알고리즘에 관한 연구

정희원 정 지원\* 정희원 장 청 룡\*\* 정희원 이 인 숙\*\* 정희원 원 동 호\*

A Study on Decoding Algorithm of TCM by Path Back Method

Ji Won Jung\*, Chung Ryong Jang\*\*, In Sook Lee\*\*, Dong Ho Won\* *Regular Members*

요 약

TCM(Trellis Coded Modulation)은 위성통신이나 음성통신과 같이 대역폭이 제한된 채널에서 코딩과 변조가 결합되어 대역폭의 증가없이 코딩 이득을 가져올 수 있는 하나의 통신 방식이다. 본 논문에서는 PAM, PSK, QAM 변조를 TCM에 적용하고, Viterbi decoding의 단점을 개선할 수 있는 확장된 path back method 복호 알고리즘을 제안하여 TCM의 복호 알고리즘에 적용하였다. 또한, Monte Carlo simulation을 이용하여 각 변조기법에 따른 성능과 복호 효율성에 대하여 분석하였다.

ABSTRACT

On band limited channels like satellite communication and voice communication, TCM(Trellis Coded Modulation) is a communication method that has coding gain which combines modulation with channel coding without bandwidth expansion.

In this paper, we apply PAM, PSK, QAM to TCM, and propose the extended path back method decoding algorithm which improved drawback of viterbi decoding algorithm and apply to TCM this decoding algorithm.

Using Monte Carlo simulation, we analyze performance of each modulation technique and efficiency of decoding algorithm.

I. 서 론

통신 방식에서 안전하고 오류율(error rate)이 적은 통신을 원할 때 주로 코딩 기법을 사용해 왔다. 이러

한 코딩 기법의 사용은 삽입되는 용장 비트(redundancy bit)로 인하여 대역폭이 증가하게 된다. 따라서 코딩 기법은 사용자의 증가로 인하여 대역폭이 제한되는 위성통신이나 mobile 통신, 음성통신에서 크게 이용되지 못하고 있다. 그러므로 대역폭이 제한된 채널에서도 대역폭의 확장없이 코딩 이득을 얻을 수 있는 방법이 꾸준히 연구되어 왔다. 이것이 TCM이 나

\*成均館大學校 情報工學科  
Dept. of Infor. Eng., Sungkyunkwan Univ.

\*\*韓國通信 研究開發團

KT

論文番號 : 92 - 139 (接受1992. 5. 28)

오게 된 동기이다.<sup>(1)</sup>

TCM은 채널 코딩과 변조가 결합된 redundant nonbinary modulation 방식으로 1982년 Ungerboeck이 처음 제안한 이후 각종 통신 방식에 널리 이용되고 있다.<sup>(2) (3)</sup>

본 본문에서는 PAM, PSK, QAM을 TCM에 적용시키고, 채널모델로 AWGN(Additive White Gaussian Noise)이 첨가된 수신 신호를 제안한 TCM 복호기를 통하여 효과적으로 복호하는 과정과 성능을 Monte Carlo simulation을 이용하여 분석하였다.<sup>(4)</sup>

중전의 Viterbi decoding algorithm을 TCM에 적용한 경우 구현시 문제가 되는 algorithm의 복잡성과 많은 연산 과정에서 야기되는 회로의 복잡성, 복호기의 크기에 따른 복호 속도 문제가 대두되는데, 이러한 문제의 해결을 위하여 본 논문에서는 convolutional coding에서 적용하는 path back method 복호 알고리즘을 변형하여 TCM에 적용하였으며, 시뮬레이션을 통하여 효율성을 확인하였다.

## II. TCM의 구성 개요

### II.1 System 구성

본 논문에서 제안한 TCM 전체 블록도는 그림 1과 같으며, 중전의 Viterbi 복호 알고리즘 대신 path back method 복호 알고리즘을 사용하였다.

그 내용을 간단히 요약하면 아래와 같다.

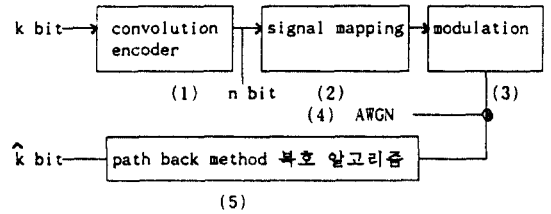
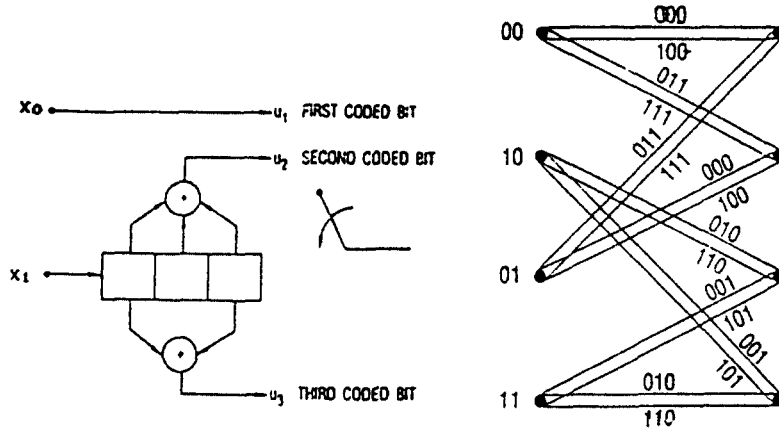


그림 1. TCM의 전체블럭도  
Fig. 1. Block diagram of TCM

- (1) convolution encoder : rate  $k/n$ 인 길쌈 부호기.
- (2) signal mapping : TCM의 성능을 좌우하는 부분으로 각 signal set에 대해 set partition과 그에 따른 code word를 할당한다.
- (3) modulation : 변조 기법으로는 PAM, PSK, QAM을 TCM에 적용한다.
- (4) channel model : channel model로서는  $mean=0$ ,  $variance=\sigma^2$ 인 AWGN을 사용.
- (5) 복호 알고리즘 : path back method 복호 알고리즘 적용.
- (6) simulation 방식 :  $10^6$ 개의  $k$  bits를 부호기의 입력 data로 주어 복호기에서 출력된  $\hat{k}$  bits와 비교하여 error를 counting하는 Monte Carlo simulation 방식을 사용.

### II.2 Signal mapping rule

그림 2에서 보는 바와 같이 부호기는 rate 1/2의



(a) 부호기

(b) Trellis diagram

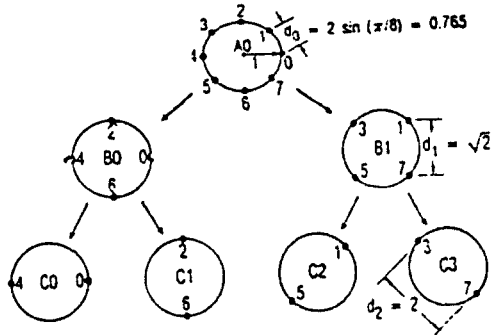
그림 2. 부호기와 격자구조(rate=2/3)

Fig. 2. Encoder and trellis diagram(rate=2/3)

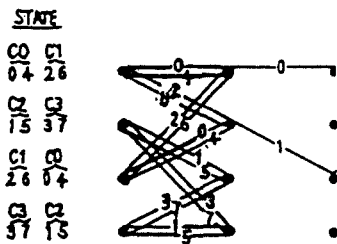
부호기에 uncoded 1 bit를 첨가시킨 형태이다. trellis diagram의 위의 두 branch는 입력 비트  $x_0x_1$ 이 00 또는 10이며 아래의 두 branch는  $x_0x_1$ 이 01 또는 11을 나타낸다. 일반적으로 신호 공간에서 coded 8 PSK는 uncoded 4 PSK보다 각 신호들 사이의 최소 거리가 좁아진다. 즉, MPSK(MPAM)에서 M이 증가할수록 오류율이 더욱 증가하는 것을 알 수 있으며 이는 전송 속도의 증가에 따라 성능이 감소되는 것을 의미한다. 이때 trellis coding 개념을 적용시킬 때 TCM 이론에 의하면 신호들 사이의 거리가 좁혀지더라도 수신된 신호를 복호할 때 trellis에 존재하는 신호의 거리만 넓혀 주면 절대 거리의 감소를 보상할 수 있다.

Trellis상에서 branch code word를 재구성하는 규칙이 Ungerboeck이 제안한 set partition에 관한 규칙이며 다음과 같다.

1) 같은 state에서 발생하는 천이는 subset B0, B1의 waveform에서 할당된다.



(a) Set partition



(b) Trellis diagram

그림 3. Set partition에 의한 trellis diagram

Fig. 3. Trellis diagram for set partition

2) 한 state에서 결합되는 천이는 subset B0, B1의 waveform에서 할당된다.

3) Parallel transition은 subset C0, C1, C2, C4로부터 할당된다.

위의 1) 2) 3)의 규칙에 따라 그림 2의 trellis diagram을 재구성하면, 즉, 격자 상태의 구조를 보고 재부호화하면 그림 3과 같이 된다.

한가지 주목해야 할 TCM의 특징은 branch에 할당된 code word들이 binary가 아닌 그에 대한 phase (PSK) 혹은 level(PAM)을 나타내는 nonbinary라는 점에서 종전의 convolution coding과의 차이점을 보인다. convolution coding과 TCM의 차이점은 표 1과 같다.

표 1. Convolution coding와 TCM의 비교

Table 1. Comparison of convolution coding and TCM

|                  | convolution coding | TCM                |
|------------------|--------------------|--------------------|
| performance 척도   | hamming distance   | euclidean distance |
| branch code word | binary             | nonbinary          |
| node             | 90° 안됨             | 90°                |
| state            | 90° 안됨             | 90°                |
| branch           | 180°               | 180°               |

### III. TCM에 적용된 변조와 성능 분석

본 논문에서는 II장에서 언급된 Ungerboeck 규칙에 따라 각 변조 기법을 TCM에 적용시켜 Monte-Carlo simulation을 이용하여 성능을 분석하였다.

표 2는 각 변조 기법에서 convolutional encoder의 속도에 따라 reference signal과 그에 따른 coded signal을 나타내었다.

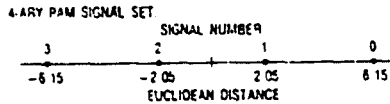
표 2. 부호기 속도에 따른 기준 신호와 부호화된 신호

Table 2. Reference signal and coded signal for encoder rate

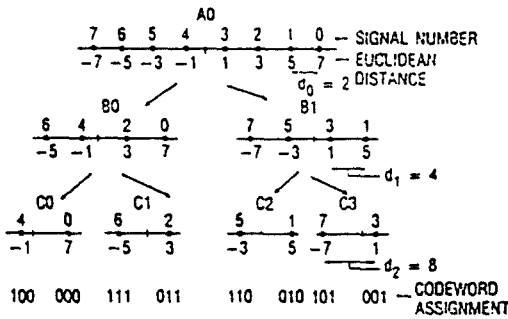
| 변조 기법 | reference signal (uncoded signal) | encoder rate | coded signal |
|-------|-----------------------------------|--------------|--------------|
| PAM   | 4 PAM                             | 2/3          | 8 PAM        |
| PSK   | 4 PSK                             | 2/3          | 8 PSK        |
| QAM   | 8 PSK                             | 3/4          | 16 QAM       |

### III.1. PAM

부호기와 trellis diagram은 그림 2, 그림 3과 동일하며, reference signal과 그의 set partition에 대한 code word 할당은 그림 4와 같다. 성능은 그림 7에 나타내었다.



(a) Reference 4 PAM signal



(b) Coded 8 PAM set partition

그림 4. Set partition에 의한 8 PAM의 code word 할당  
Fig 4. Code word assignment of 8 PAM for set partition

### III.2 PSK

PSK의 부호기는 PAM의 경우와 마찬가지로 set partition에 의한 trellis diagram은 II장의 그림 3에서 살펴보았으며, 성능은 그림 7과 같다.

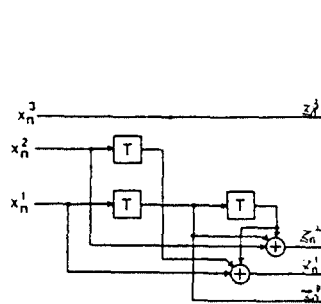
### III.3 QAM

QAM은 reference 8 PSK와는 다른 constellation을 갖고있으며 부호기와 그에 따른 signal constellation은 그림 5와 같다.

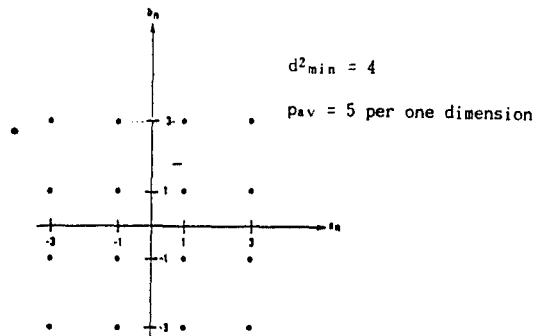
Set partition에 의한 code word 할당과 trellis diagram은 그림 6과 같으며 성능은 그림 7과 같다.

$E_s/N_0$ 와  $E_b/N_0$ 가 동일한 BPSK와는 달리 4 PSK, 8 PSK, 16 QAM에서는 2bit, 3bit, 4bit를 한 symbol로 할당하였기 때문에 본 논문에서는 x축을  $E_s/N_0$ 로써 시블레이션을 하였다.

그림 7(a)에서 보는 바와 같이 convolutional coding 보다 TCM이 약 1.1dB, uncoded system보다 약 3dB정도 코딩 이득을 얻을 수 있음을 알 수 있으며, 그림 7(c)에서 state수의 증가에 따라 코딩 이득을 더 많이 얻을 수 있음을 알 수 있다.



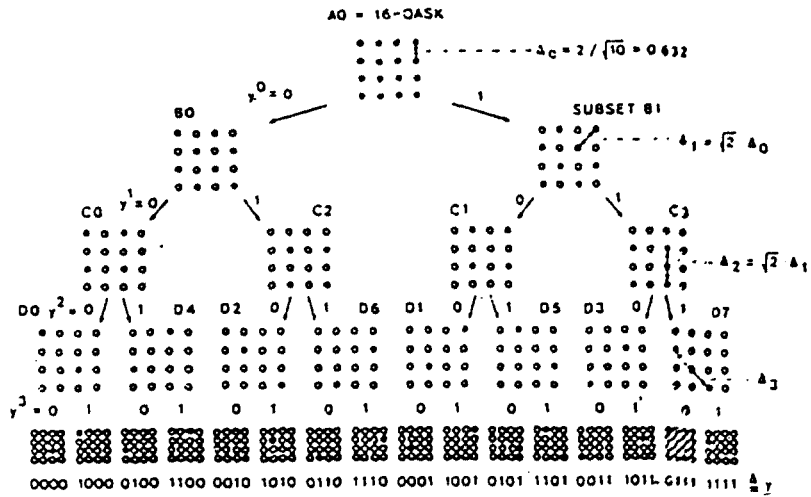
(a) 부호기



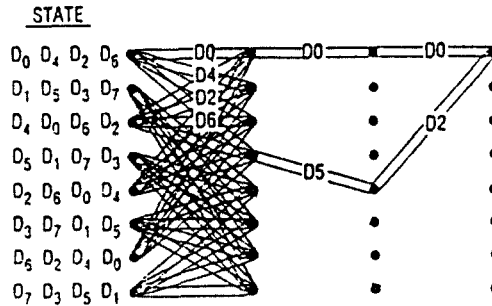
(b) Signal constellation

그림 5. 16 QAM의 부호기와 signal constellation  
(rate=3/4)

Fig 5. Encoder and signal constellation of 16 QAM  
(rate=3/4)



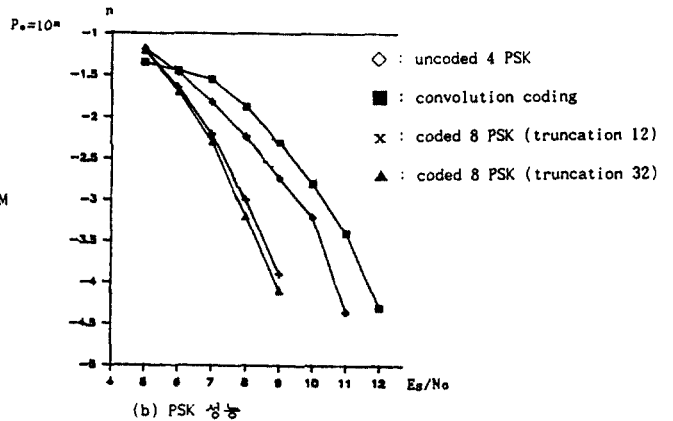
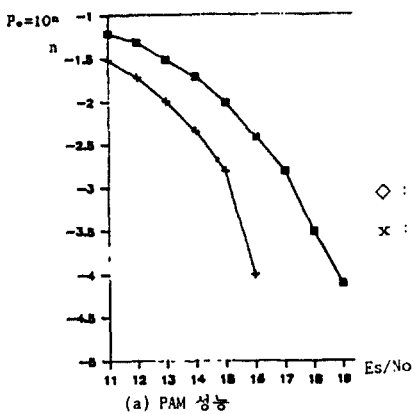
(a) QAM code word assignment

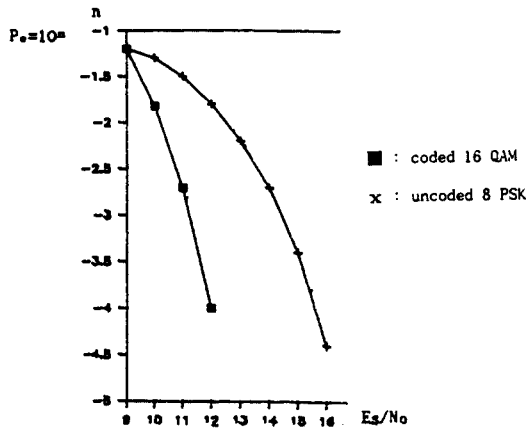


(b) Trellis diagram

그림 6. Set partition에 의한 QAM 부호할당과 격자구조

Fig 6. QAM code word assignment and trellis diagram for set partition





(c) QAM 성능

그림 7. PAM, PSK, QAM, 성능 분석

Fig 7. PAM, PSK, QAM performance

각 절에서 살펴본 각 변조 기법에 대한 전체적인 결과는 표 3과 같다.

표 3. PAM, PSK, QAM의 코딩 이득

Table 3. Coding gain of PAM, PSK, QAM

| 변조기법 | State | Coding gain(dB) | Sav |
|------|-------|-----------------|-----|
| PAM  | 4     | 2.1             | 21  |
| PSK  | 4     | 2.9             | 1   |
| QAM  | 8     | 3.8             | 2.1 |

평균신호전력(Sav) : 신호 공간에서 각 신호와 중심과의 거리의 평균값

$Sav = (d_1^2 + d_2^2 + d_3^2 + M^2) / M$ , 여기서  $d_i$  : euclidean distance

#### IV. Convolution coding에서 Path-back method 복호알고리즘

Viterbi decoding은 forward error correction 방식으로 state와 branch수의 증가로 인하여 많은 연산작용이 요구된다.

복호과정에서 이러한 회로의 복잡성, 복호기 크기의 증가로 인하여 메모리, 복호 속도 문제가 대두된다. 이러한 단점을 극복하기 위한 연구가 계속되고 있으며, 1988년 path-back method 복호 알고리즘이 처음으로 도입되었다. 이 방식의 경우 복호기 크기는 종전에 비해 급격히 줄일 수 있었으나, 한 bit를 복호하기 위한 복호 속도에는 별 영향을 미치지 못하므로 개선된 path back method 복호 알고리즘이 도입되게 되었다. 종전의 viterbi decoding 방식과 path back method 복호 알고리즘의 차이점은 표 4와 같다. 표 4에서 T는 path memory에서 한 bit를 복호하기 위해 저장되어야 하는 path로 구축장 수의 4배 혹은 5배 정도 해당하면 최적의 복호기 성능에 관점함을 알 수 있다<sup>7)</sup>

#### IV.1 Convolution coding에 있어서 path back method 복호 알고리즘

먼저 TCM의 path back 복호 알고리즘의 이해를 돕기 위하여 그림 8의 부호기와 격자구조를 예로 들어 기존의 convolution coding에 대한 path back method 복호 알고리즘에 대해 논하겠다.

표 4. Viterbi decoding 방식과 path back method 복호 알고리즘의 비교

Table 4. Comparison of Viterbi decoding and path back method decoding algorithm

|   | Viterbi decoding 방식                  | path back method   |
|---|--------------------------------------|--|
| error correction 방식                           | forward                              | back   |
| k에 따른 performance                             | 항상                                   | 항상   |
| k에 따른 hardware                                | 연산작용이 많아져 기하급수적으로 decoder size가 커진다. | time-shared multiple use of circuit (ACS작용)로 이전의 방식에 비해 줄인다  |
| 행렬 [ $2^{k-1} \times T$ ]<br>path memory cell | selector D F/F (소자의 수가 많음)           | path selector memory, node number calculator, 에 대해 LSI RAM만 사용 (속도에는 별 영향이 없다-개선된 path back method에서 보완) |

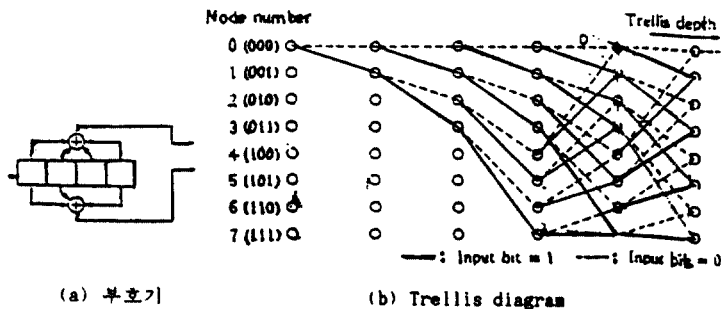


그림 8. 부호기와 격자구조(R=1/2, K=4)  
Fig 8. Encoder and trellis diagram

- (1) Branch matrices를 구한다. (branch signal과 received signal과의 거리(Hamming distance)를 구한다.
- (2) ACS(Adder-Compare-Selector) : 한 node로 들어오는 것 중에서 branch matrices 이 가장 작은 path를 선택함으로써 path select signal 결정한다. 한 node로 들어오는 것 중 위로부터 들어오는 path를 "0"으로 보고 아래서 들어오는 path를 "1"로 본다.
- (1), (2)의 과정을 그림 9의 branch matrices calculator, ACS-0, ..., ACS-2<sup>n-1</sup>-1로 나타내었다.

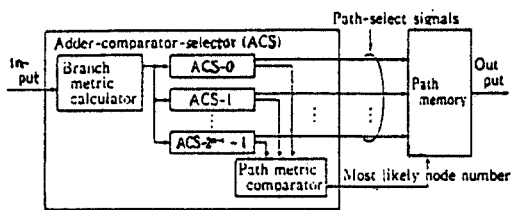


그림 9. Viterbi decoder 블럭도  
Fig 9. Viterbi decoder block diagram

- (3) 행렬 [2<sup>k-1</sup> X T]의 원소인 path select signal을 path memory에 저장.

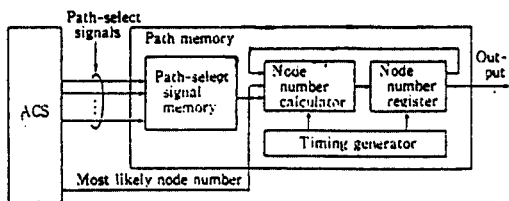


그림 10. Path memory  
Fig 10. Path memory block diagram

- (4) Node number calculator에 의해 T만큼 trace back 한 뒤에 한 bit를 복호시킨 다음 shift시킨다. path memory에서 복호되는 과정은 그림 11과 같다. Next node number calculator 공식은 다음과 같다.

$$N_i = 2^{k-2} \times PS_i + \lceil N_{i-1} / 2 \rceil \dots\dots\dots (1)$$

「X」=X보다 크지않는 최대의 정수  
N<sub>i</sub>=node number  
PS<sub>i</sub>=path select signal  
k=constranin length

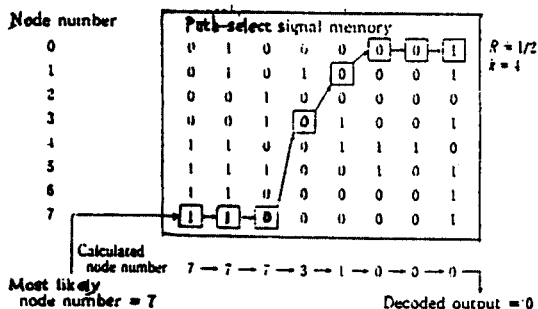


그림 11. Path memory 작용  
Fig 11. Operation of path memory

4.2. 개선된 path back method 복호 알고리즘  
이상 위에서 살펴본 바와 같이 부호기 크기는 줄일 수 있으나 복호 속도에는 영향을 미치지 못한다는 것

을 알 수 있다. 그림 12, 13에서와 같이 복호 속도를 향상시킬 수 있는 새로운 개선된 방법이 제안되었다.

Trace memory에 저장되어 있던 node number와 node number calculator에서 출력되는 현재의 most likely node number를 comparator에서 비교하여 일치하면 더 이상 trace back 하지않고 바로 복호한다.

Path memory의 블록도는 그림 12과 같으며 그에 따른 복호과정은 그림 13에 나타내었다.

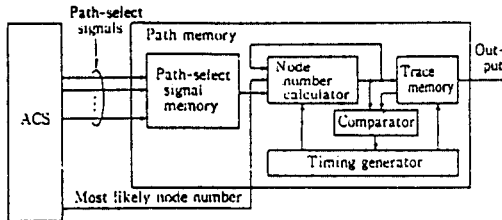


그림 12. Path memory의 블록도  
Fig 12. Block diagram of path memory

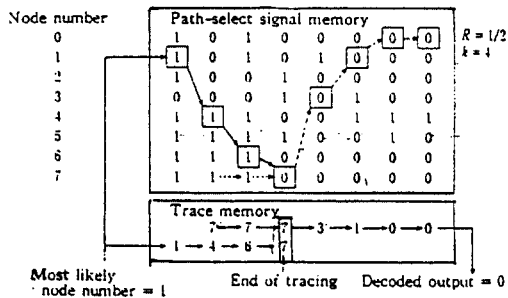
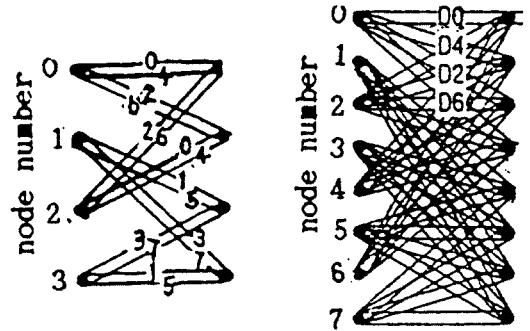


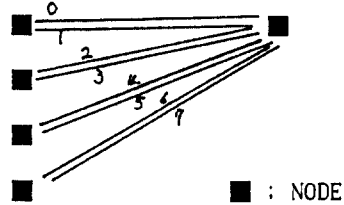
그림 13. Path memory 연산작용  
Fig 13. Path memory operation

### V. 제안한 TCM에서의 path back method 복호 알고리즘

본 논문에서 제안한 TCM에서의 path back method 복호 알고리즘의 ACS, comparator 작용은 기존의 path back method 복호 알고리즘과 동일하나 path select signal과 node number calculator의 변화를 살펴보고 Monte Carlo simulation을 이용하여 각 SNR에서 평균 추적 횟수를 구함으로써 효율성이 향상됨을 확인하였다.



(a) Node number



(b) Path select signal

그림 14. 4 state와 8 state의 node number와 path select signal

Fig 14. Node number and path select signal for 4 state and 8 state

위의 그림 14에서 node number는 기존의 convolution coding과 동일하나 node로 들어 오는 path select signal을 본 논문에서는 0,1,...,2<sup>k</sup>-1을 할당하여 그에 따른 node number calculator의 공식을 제안한다.

TCM에서 node number calculator는 다음과 같은 공식에 의하여 다음 node를 결정한다. (공식 (1)과 (2)의 차이점은 표 4에 나타내었음)

$$N_i = 2 \times \lceil PS_i / 2 + N_{i-1} / 2^{k-1} \rceil \dots \dots \dots (2)$$

$$PS_i = 0, 1, \dots, 2^k - 1$$

$$N_i = 0, 1, \dots, 2^k - 1$$

「X」= X보다 크지않는 최대의 정수

k = constrain length

위에서 제안한 공식을 적용하면 TCM에서 구속장수가 증가하여도 path select signal과 종전의 node



number를 이용하여 다음 node number를 추적할 수 있다. Path memory operation을 통하여 4 state, 8 state에서 복호하는 과정은 그림 15와 같다.

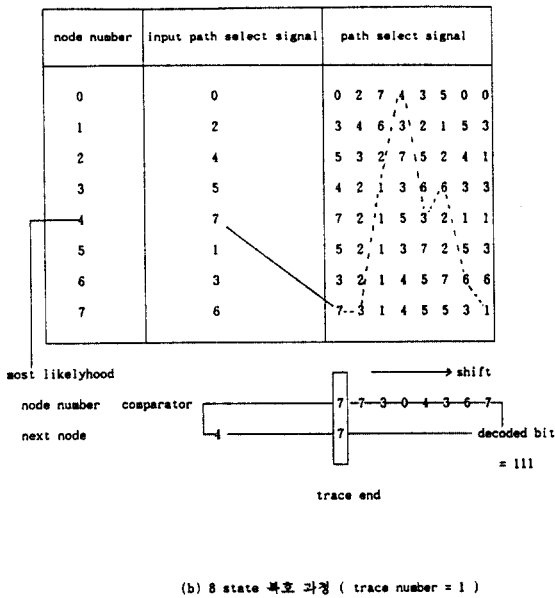
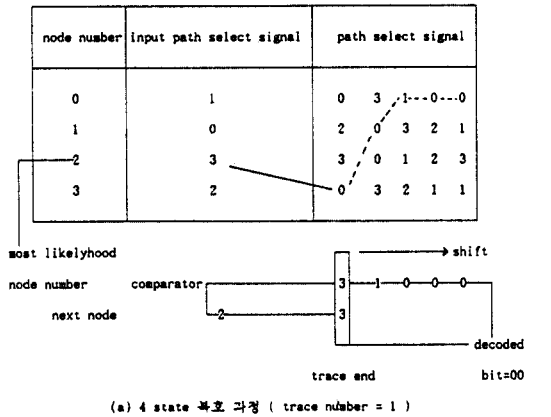


그림 15. 4 state와 8 state에서 path memory operation  
Fig 15. Path memory operation of TCM for 4 state and 8 state

그림 15의 복호 과정의 순서는 다음과 같다.

- 1) 이전의 path select signal과 node number를 path memory(행렬  $[2^{k-1} \times T]$ )에 기억시킨다.
- 2) 현재 입력되는 node number와 path select signal을 node number calculator에 입력 시킨다.(그림 11)

- 3) Node number calculator의 출력 node number와 path memory에 저장된 node number를 comparator에서 비교한다.
- 4) 비교한 결과 먼저 저장되어 있던 node number와 일치하는 경우 더 이상 trace back하지 않고 그림 16과 같이 복호 균일성에 의해서 2bit, 3bit의 data를 복호한다.

| node number | decoded bit | path select signal |
|-------------|-------------|--------------------|
| 0           | 00          | 0, 2               |
|             | 10          | 1, 3               |
| 1           | 10          | 0, 2               |
|             | 11          | 1, 3               |
| 2           | 00          | 0, 2               |
|             | 10          | 1, 3               |
| 3           | 01          | 0, 2               |
|             | 11          | 1, 3               |

(a) 4 state

| node number | decoded bit | path select signal |
|-------------|-------------|--------------------|
| 0           | 000         | 0, 2, 4, 6         |
|             | 100         | 1, 3, 5, 7         |
| 1           | 001         | 0, 2, 4, 6         |
|             | 101         | 1, 5, 3, 7         |
| 2           | 010         | 0, 2, 4, 6         |
|             | 110         | 1, 3, 5, 7         |
| 3           | 011         | 0, 2, 4, 6         |
|             | 111         | 1, 3, 5, 7         |
| .           | .           | .                  |
| .           | .           | .                  |
| 7           | 011         | 0, 2, 4, 6         |
|             | 111         | 1, 3, 5, 7         |

(b) 8 state

그림 16. TCM의 복호 균일성

Fig 16. Uniform property of decoding for TCM

- 5) 다음 path memory에 있는 node number와 path select signal을 한 bit shift시킨다.
- 6) Next node number와 path select signal을 path memory에 입력한다.
- 7) 1).....6)의 복호 과정을 전송된 비트수 만큼 반복하여 수신 신호와 송신 신호를 비교한 후 오류 비트수를 측정하여 변조 기법에 대한 성능을 분석한다.

다.

위의 step을 통하여 T만큼 trace back하지 않고도 2 bit, 3 bit를 복호할 수 있기 때문에 T-trace number 만큼의 복호 효율성이 향상됨을 알 수 있다.

Convolution coding과 제안한 TCM의 path back method 복호 알고리즘의 차이점은 표 4와 같다.

표 4. Path back method를 적용할 경우 TCM과 convolution coding의 복호과정의 차이점.

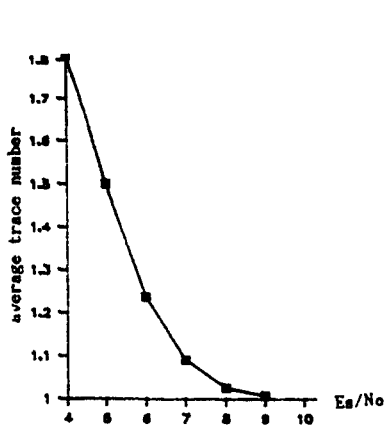
Table 4. Comparison of TCM and convolution coding for path back method decoding algorithm

|                        | convolutional coding                                    | TCM (제안한 복호알고리즘)  |
|------------------------|---|---|
| node number calculator | $N_i = 2^{k-2} \times PS_i + \lceil N_{i-1} / 2 \rceil$ | $N_i = 2 \times PS_i / 2 + \lceil N_{i-1} / 2^{k-1} \rceil$ |
| path select signal     | $PS_i = 0, 1$   | $PS_i = 0, 1, \dots, 2^{k-1} - 1$                           |
| node number            | $N_i = 0, 1, \dots, 2^{k-1} - 1$                        | $N_i = 0, 1, \dots, 2^{k-1} - 1$                            |

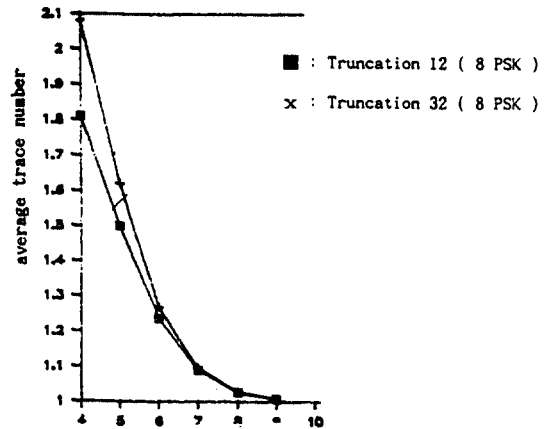
제안한 TCM 복호 알고리즘과 기존의 복호 알고리즘과 비교하면 path select signal과 node number calculator에서 표 4와 같은 차이점을 갖게 된다. 또한 제안한 TCM 복호 알고리즘을 실행한 후 알고리즘의 효율성을 알 수 있는 SNR에서 평균 추적횟수는 그림 17과 같다.

그림 17(a)에서 low SNR보다 high SNR일때 평균

추적 횟수가 1에 근접하기 위해서 더 많은 SNR이 필요하게 됨을 알 수 있다. (b)에서는 같은 구축장 수에서 truncation수가 증가하면 7dB에서 평균 추적 횟수가 같게 됨을 알 수 있으며 (c)에서는 구축장 수가 커짐에 따라 평균 추적 횟수가 1에 근접하기 위해서 더 많은 SNR이 필요하게됨을 알 수 있다.

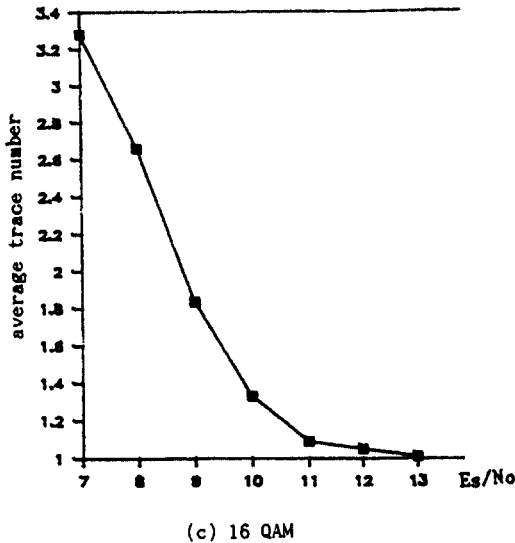


(a) 8 PSK



(b) Truncation (T)에 따른 평균추적횟수

참 고 문 헌



(c) 16 QAM  
 그림 16. Decoding cycle에서의 평균 추적횟수  
 Fig 16. Average trace number at decoding cycle

VI. 결 론

TCM은 위성통신이나 음성통신과 같이 대역폭이 제한된 채널에서 코딩과 변조가 결합되어 대역폭의 증가없이 코딩 이득을 향상시킬 수 있는 통신 방식으로 널리 이용되고 있다. 기존의 TCM 복호 방식으로는 Viterbi 복호 알고리즘을 사용하고 있으나 본 논문에서는 path back method 복호 알고리즘을 변형시켜 TCM에 적용시킴으로써 Viterbi 복호 알고리즘의 단점인 복호 회로의 복잡성과 복호 속도를 개선하였다.

Path back method 복호 알고리즘을 TCM에 적용시키기 위해서 path select signal과 다음 node number를 결정하는 node number calculator의 공식을 제안하였으며, 이를 TCM에 실제로 적용시켜 복호 cycle에서 평균 추적 횟수가 감소됨을 Monte Carlo simulation을 통하여 확인하였다.

본 논문에서 path back method 복호 알고리즘을 적용시켜 구성한 TCM 복호 방식은 대역폭이 제한된 통신 방식이 유용하리라 사료된다.

1. Sklar, B. "Digital Communications : Fundamentals and Applications" Prentice Hall, pp.333-428, 1988
2. Sklar, "Trellis Coded Modulation", pp.35.1.1-35.1.9, 1988
3. Ungerboeck, G., "Channel Coding with Multilevel/Phase Signals" IEEE Trans. Inform. Theory, Vol. IT-28, pp. 55-67, January, 1982
4. Shu Lin, "Error Control Coding," Prentice Hall, pp. 287-349, 1983
5. 이만영, "符號理論", pp.277-353, 회중당, 1984
6. Hemant K. "Real Time Application of Trellis Coding to High Speed Voice Band Data Transmission", IEEE Journal Select Area in Commun., Vol. 2, pp. 648-658, Sept, 1984
7. Atsushi Yamashita, "A New Path Memory for Viterbi decoders" FUJITSU Sci. Tech.J., pp.37-42, March, 1989
8. J.A.Heller, 'Viterbi Decoding for Satellite and Space Communication' IEEE Trans. Commu. Technol., Vol. COM-19, pp.835-848, 1971
9. G.D.Forney, Jr., 'The Viterbi Algorithm', Proc. IEEE, Vol.61, pp.268-278 Mar, 1973
10. Rader, C.M "Memory Management in a Viterbi Decoder" IEEE Trans. Commun. Vol.9 pp.1399-1401, 1981
11. Coates, 'Monte Carlo Simulation' IEEE Journal on Select Vol. 6, pp.58-66, 1988



정 지원(Ji-Won Jung) 正會員  
1989년 : 성균관대학교 전자공학과  
졸업(공학사)  
1991년 : 성균관대학교 대학원 전  
자공학과 졸업(공학석사)  
1992년~현재 : 성균관대학교 대학  
원 정보공학과 박사과정  
1990년 11월~1992년 1월 : 금성정  
보통신연구소 연구원



장 청 룡(Chung Ryong Jang) 正會員  
1980년 : 성균관대학교 전자공학과  
졸업(공학사)  
1986년 : 연세대학교 대학원 전자  
공학과 졸업(석사)  
1979년~1983년 : 한국전자통신연  
구소 연구원  
1984년~현재 : 한국통신 연구개발  
단 선임연구원



이 인 숙(In-Sook Lee) 正會員  
1979년 : 이화여자대학교 수학과 졸  
업(이학사)  
1985년 : 이화여자대학교 대학원 수  
학과 졸업(이학석사)  
1979년~1984년 : 한국 전자 통신 연  
구소 연구원  
1984년~현재 : 한국통신 연구개발  
단 선임연구원



元 東 豪(Dong-Ho Won) 正회원  
1976년 : 성균관대학교 전자공학과  
졸업(공학사)  
1978년 : 성균관대학교 대학원 전  
자공학과 졸업(공학석사)  
1988년 : 성균관대학교 대학원 전  
자공학과 졸업(공학박사)  
1978년~1980년 : 한국전자통신연  
구소 전임연구원  
1985년~1986년 : 일본 동경공대 객원연구원  
1982년~현재 : 성균관대학교 정보공학과 교수