



§4. Maximum-Likelihood Decoding (Viterbi Decoding)

The difficulty of the maximum-likelihood (ML) decoding is the complexity that grows exponentially in terms of decoding tree depth for a fixed constraint length, i.e., a fixed number of states of the trellis diagram (or tree) (signal length).

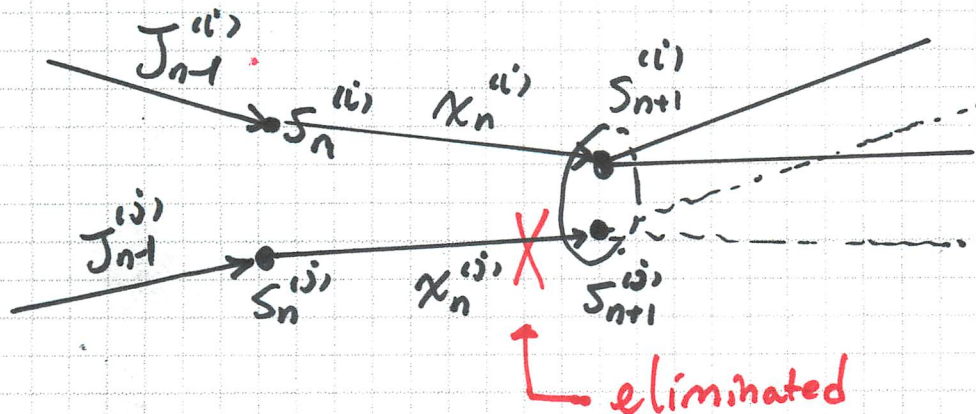
The Viterbi decoding algorithm basically says that this tree can be reduced by merging nodes such that the tree only grows to a maximum size of 2^S nodes, where S is the number of states, i.e., the complexity is only exponential in terms of constraint length but not signal length.

To see this, let $J_{n-1}^{(i)}$ and $J_{n-1}^{(j)}$ be the metrics of two nodes corresponding to the partial sequences $\underline{\hat{x}}^{(i)}$ and $\underline{\hat{x}}^{(j)}$ of length $n-1$, respectively.

Let the encoder states corresponding $\underline{\hat{x}}^{(i)}$ and $\underline{\hat{x}}^{(j)}$ after time $n-1$ be $S_n^{(i)}$ and $S_n^{(j)}$ and $S_n^{(i)} \neq S_n^{(j)}$.



Assume that the next extension of $\underline{\tilde{x}}^{(i)} \rightarrow (\underline{\tilde{x}}^{(i)}, x_n^{(i)})$
 $\underline{\tilde{x}}^{(j)} \rightarrow (\underline{\tilde{x}}^{(j)}, x_n^{(j)})$
is such that $S_{n+1}^{(i)} = S_{n+1}^{(j)}$, i.e., the encoder
states at time n are identical.



We propose to merge nodes $(\underline{\tilde{x}}^{(i)}, x_n^{(i)})$ and
 $(\underline{\tilde{x}}^{(j)}, x_n^{(j)})$ into one node which we call a state.

We retain the partial sequence with the larger
metric J_n at time n and discard the partial sequence
with the smaller metric.

The following theorem guarantees that the above
merging nodes processing keeps the ML decoding.



Thm (Theorem of non optimality): The procedure of merging nodes that corresponding to identical encoder states and discarding the path with the smaller metric never eliminates the maximum-likelihood path.

Proof: The metric at time $n+k$ can be written as

$$J_{n+k}^{(i)} = J_n^{(i)} + \sum_{h=1}^k \beta_{n+h}^{(i)}$$

for every future metric $J_{n+k}^{(i)}$, $0 \leq k \leq l-n$, where $\beta_{n+h}^{(i)} = 2 \operatorname{Re} \{x_{n+h}^{(i)} y_{n+h}^*\} - |x_{n+h}^{(i)}|^2$ is the metric increment (also called the branch metric) at time $n+h$.

If the nodes of paths i and j correspond to the same encoder state at time n , there exists for every possible extension of the i th path

$(x_n^{(i)}, \dots, x_{n+k}^{(i)})$ a corresponding identical extension $(x_n^{(j)}, \dots, x_{n+k}^{(j)})$ of the j th path, and vice versa.

Let $\mathcal{P}^{(i)}$ denote the set of all possible i th paths

Let $\mathcal{P}^{(j)}$ passing $s_{n+1}^{(i)}$ after the time n until time $n+l$.
 $\text{---} \text{---} \text{---} s_{n+1}^{(j)} \text{---} \text{---} \text{---} j \text{th} \text{---} \text{---} \text{---}$



$$\Rightarrow \mathcal{P}^{(i)} = \mathcal{P}^{(j)} = \{\beta_{n+1}, \dots, \beta_{n+k}\}$$

Assume without loss of generality that the i th path accumulates the largest metric at time $n+k$, i.e., $J_{n+k}^{(i)} \geq J_{n+k}^{(j)}$.

Therefore

$$J_n^{(i)} + \sum_{h=1}^k \beta_{n+h}^{(i)} \geq J_n^{(j)} + \sum_{h=1}^k \beta_{n+h}^{(j)},$$

and $\sum_{h=1}^k \beta_{n+h}^{(i)}$ is the maximum metric sum for the extensions of the i th path.

Since $\mathcal{P}^{(i)} = \mathcal{P}^{(j)}$, we have

$$\sum_{h=1}^k \beta_{n+h}^{(i)} = \sum_{h=1}^k \beta_{n+h}^{(j)} \Rightarrow J_n^{(i)} \geq J_n^{(j)}$$

(since the extensions from $S_n^{(i)}$ and $S_n^{(j)}$ are identical).

This proves that path j can therefore never accumulate a larger metric than path i . $\square$

There are 2^k paths merging in a single state at each step and there are $S \cdot 2^k$ total partial sequences (paths) at each stage (time).



After the merging/elimination, there are at most S retained partial sequences $\underline{\hat{x}}^{(i)}$ called the Survivors.

The most convenient labeling convention is that each state is labeled by the corresponding encoder state plus the survivor that leads to it. The task of the decoder is to retrace the path that the encoder traced through this trellis. The theorem guarantees that this procedure is optimal.

This complexity is the exponential of the constraint length ν rather than the length of the signal sequence.

The Viterbi Algorithm

Step 1: Initialize the S states of the ML decoder with a metric $J_{-1}^{(i)} = -\infty$ and survivors $\underline{\hat{x}}^{(i)} = \{ \}$. Initialize the starting state of the encoder, usually state $i'=0$, with the metric $J_{-1}^{(0)} = 0$. Let $n=-1$.



Step 2: Calculate the branch metric

$$\beta_n = 2 \operatorname{Re} \{ x_n y_n^* \} - |x_n|^2$$

for each state $S_n^{(i)}$ and each extension $K_n^{(i)}$.

Step 3: Shift the state $S_n^{(i)} \rightarrow S_{n+1}^{(i)}$ according to the trellis transitions determined by the encoder FSM and, from the 2^k merging paths, retain the survivor $\underline{X}_n^{(i)}$ for which $J_n^{(i)}$ is maximized.

Step 4: If $n < L$, let $n = n+1$ and go to Step 2.

Step 5: Output the survivor $\underline{X}_L^{(i)}$ that maximizes $J_L^{(i)}$ as the ML estimate of the transmitted sequence.

One does not have to wait until the entire sequence is decoded before starting to output the estimated symbols $\hat{x}_n^{(i)}$ or the corresponding data.

The probability that the symbols in all survivors $\underline{X}_n^{(i)}$ are identical for $m < n - n_t$, where n is the current active decoding time and n_t , called the truncation



length or decision depth, is $\approx 5v$, where v is the constraint length of the code, is very close to 1.

Modify

Step 4: If $n \geq n_t$, output $\hat{x}_{n-n_t}^{(i)}$ from the survivor $\hat{x}_n^{(i)}$ with the largest metric $J_n^{(i)}$ as the estimated symbol at time $n-n_t$.

If $n < l-1$, let $n = n+1$ and go to Step 2.

Step 5: Output the remaining estimated symbols $\hat{x}_n^{(i)}$, $l-n_t \leq n \leq l$, from the survivor $\hat{x}_l^{(i)}$ that maximizes $J_l^{(i)}$.

The complexity of the Viterbi algorithm is $O(52^k)$,

where 5 is the # of states.

HWK: Make the general Viterbi algorithm for a general Convolutional code with BPSK signaling. Test it for a $1/2$ rate CC.



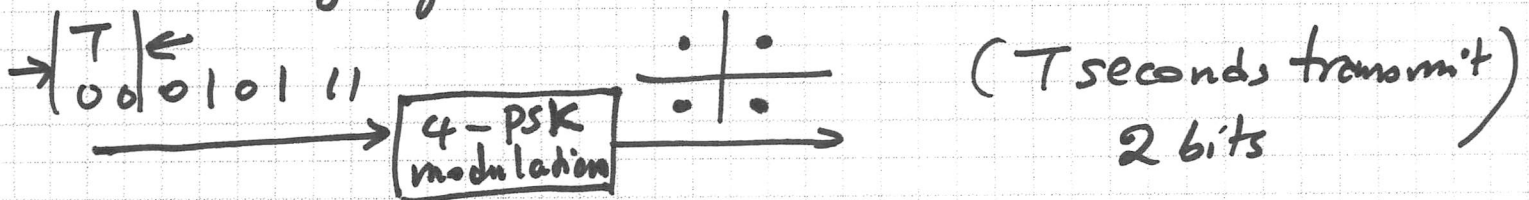
Chapter 7. Trellis Coded Modulation (TCM): Fundamentals.

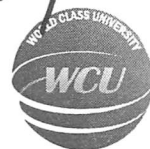
TCM is a combined coding and modulation scheme for improving the reliability of a digital transmission system without increasing the transmitted signal power or the required bandwidth.

§7.1. Introduction

For a coding and transmission system, two things:
throughput (bandwidth)
power (distance property)

(a). Use no coding and 4-PSK modulation, with one signal every T seconds. In this case, every signal carries two information bits.

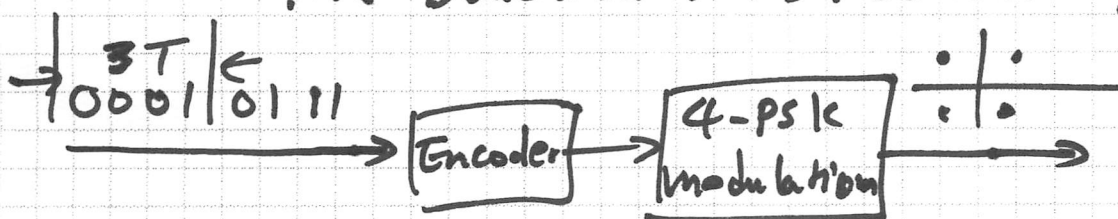




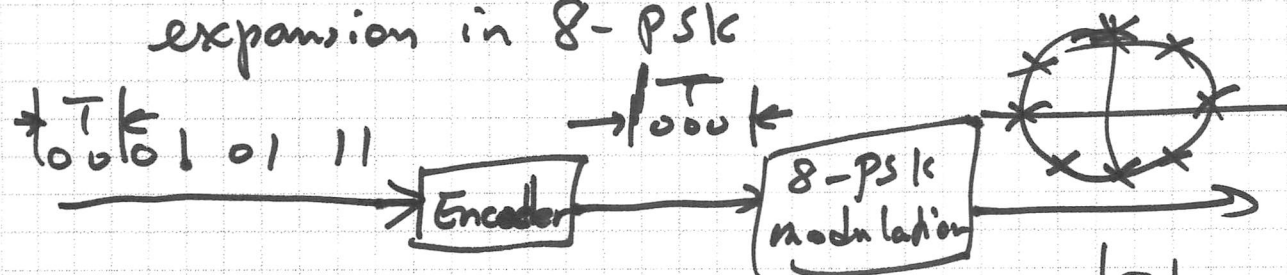
(b). Use a convolutional code with rate $2/3$ and 4-PSK modulation. Every signal now carries $4/3$ information bits. T seconds transmit $4/3$ bits.

4 bits $\rightarrow$ 6 bits $\rightarrow$ 3 symbols $\rightarrow 3T$

The bandwidth is reduced by $3/2$

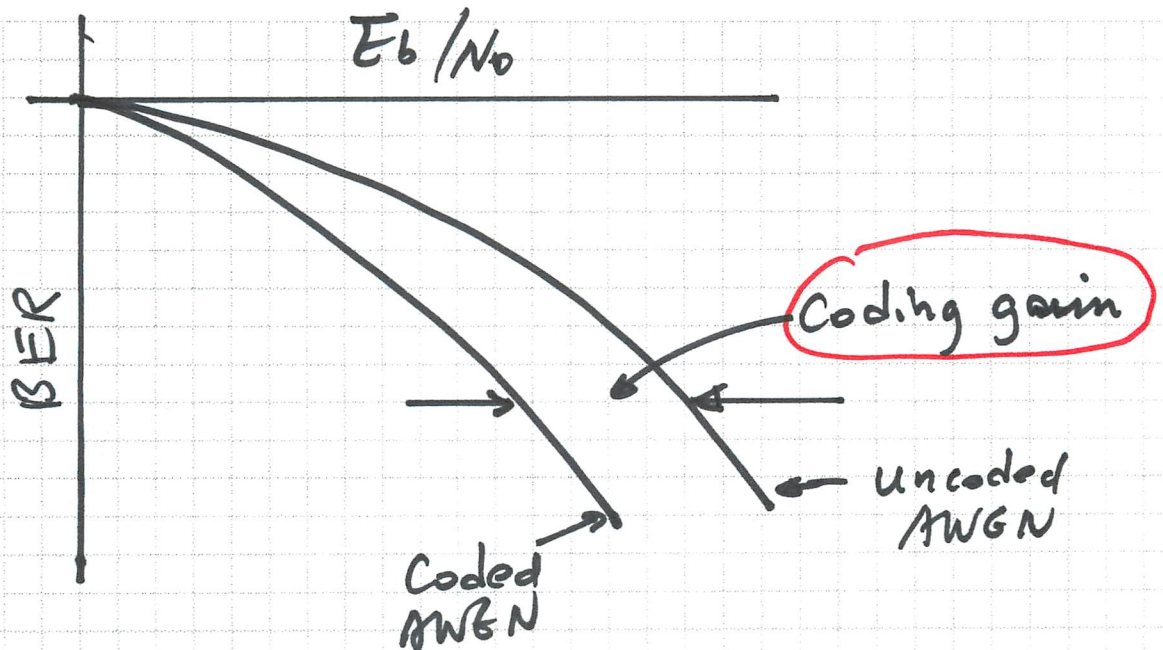


(c). Use a convolutional code with rate $2/3$ and 8-PSK modulation to avoid reducing the signal duration. Each signal carries two information bits and hence no bandwidth expansion in 8-PSK.



distance is reduced

TCM: treat convolutional encoding & modulation as a single step
decoding & demodulation
to achieve (better) coding gain.



- * The free Hamming distance of a convolutional code becomes the free Euclidean distance between transmitted signal sequences.
- * The optimization of the TCM design will be based on Euclidean distance rather than on Hamming distance.
- * The detection process will involve soft rather than hard decision, i.e., received signals are processed before making decision as to which transmitted symbol they correspond.



What is TCM?

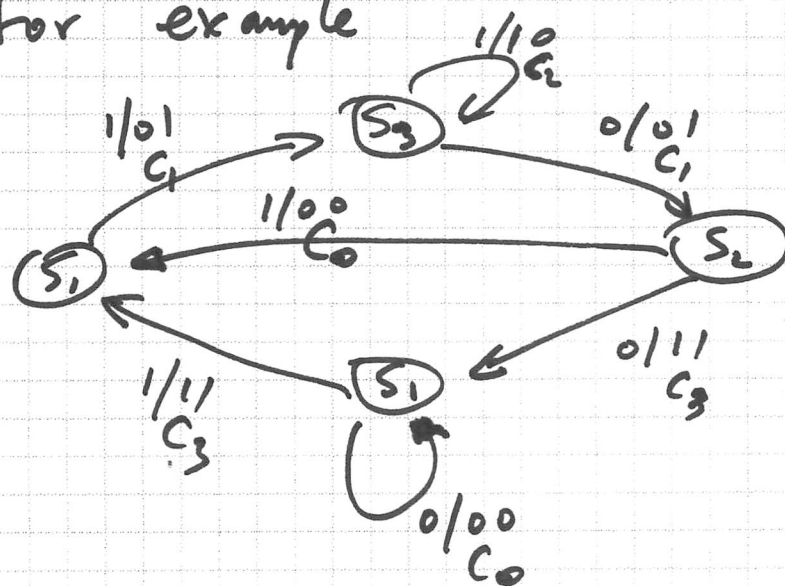
Consider a (n, k, v) convolutional code

k bits in $\rightarrow$ n bits out

$\downarrow$ map n bits to a modulated symbol

k bits in $\rightarrow$ a complex symbol out

For example



0 1 1 1 0 0 1
 $\downarrow$
 $C_0 C_3 C_1 C_2 C_1 C_3 C_2$

§7.2. What is good about TCM?

Coding Gain Concept

* Communication Model: discrete-time & continuous-amplitude & additive white Gaussian noise

Messages to be delivered to the user are represented by points or vectors, in an N -dimensional Euclidean space $\mathbb{R}^N$, called signal space.



For example in QPSK, $N=2$, $\mathcal{R}^2 = \mathbb{C}$.

* When $\underline{x}$ is transmitted, the received signal

$$\underline{z} = \underline{x} + \underline{v},$$

where $\underline{v}$ is a noise vector whose components are independent Gaussian random variables with mean zero and the same variance $N_0/2$.

* Vector $\underline{x}$ is chosen from a set Ω' consisting of M' signal vectors, the signal constellation. For example, in QPSK, $M'=4$, $\Omega' = \left\{ \begin{smallmatrix} - \\ \vdots \\ \vdots \\ \vdots \end{smallmatrix} \right\}$

The average square length (for vector)

$$E' = \frac{1}{M'} \sum_{\underline{x} \in \Omega'} \|\underline{x}\|^2,$$

is referred to as the average signal energy.

* Consider a sequence $\{\underline{x}_i\}_{i=0}^{K-1}$ of transmission and $\{\underline{y}_i\}_{i=0}^{K-1}$ is the received sequence. The

receiver chooses "valid" $\underline{z}_0, \dots, \underline{z}_{K-1}$ from the set Ω' such that this distance

$$d = \sqrt{\sum_{i=0}^{K-1} \|\underline{y}_i - \underline{z}_i\|^2} \quad \text{is minimized.}$$



* The error probability is upper bounded at least for high SNR, by a decreasing function of $\frac{d_{\min}^2}{N_0}$, where $d_{\min}$ is the minimum squared Euclidean distance between two allowable signal vector sequences in space $\Omega' \times \Omega' \times \dots \times \Omega' = (\Omega')^K$

* Uncoded transmission

The "valid" sequence $\mathcal{E}_0, \dots, \mathcal{E}_{K-1}$ runs all possible sequences in the product set $(\Omega')^K$. For example, in QPSK modulation, $\mathcal{E}_k$ may take any one of the four symbols for any time index k , $0 \leq k \leq K-1$. All $\mathcal{E}_0, \dots, \mathcal{E}_{K-1}$ are independent.

In this case, $d_{\min}^2 = \min_{\substack{\underline{x}' \neq \underline{x}'' \\ \underline{x}', \underline{x}'' \in \Omega'}} \|\underline{x}' - \underline{x}''\|^2$,

i.e., the minimum distance between all "valid" sequences in $(\Omega')^K$ is the same as the minimum distance of the constellation Ω' . This is because you may have $K-1$ equal $\mathcal{E}_k$ but only one different $\underline{x}'$ and $\underline{x}''$ in two valid sequences:

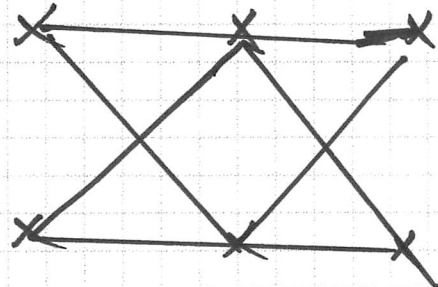
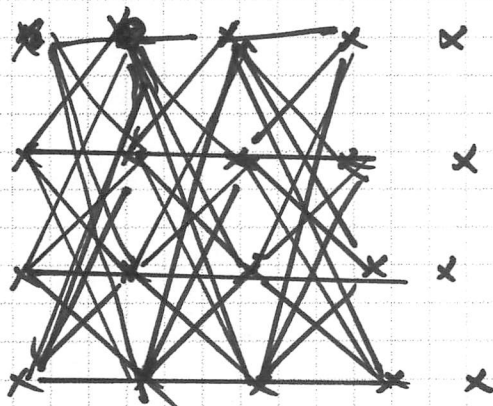
$(\mathcal{E}_0, \dots, \mathcal{E}_{K-2}, \underline{x}')$

$(\mathcal{E}_0, \dots, \mathcal{E}_{K-2}, \underline{x}'')$



8-14

$$E_b = \frac{E'}{\log_2 M'}, \text{ where } E' = \frac{1}{|\Omega'|} \sum_{\mathbf{z} \in \Omega'} \|\mathbf{z}'\|^2$$



* Coded transmission

One way to improve the system BER vs E_b/N_0 performance is to remove the assumption that the "valid" signal sequences are independent. We introduce the dependence for $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{k-1}$. Then the "valid" sequences are not the whole space $(\Omega')^K$ but a subset of $(\Omega')^K$. This, however, will reduce the transmission rate (use less sequences)

Basically, how many "valid" sequences 16
 $\Leftrightarrow$ how many information bits $2^4 \Rightarrow 4 \text{ bits}$

To achieve the goal without reducing the information rate, we may increase the signal constellation size M' , i.e., choose $\Omega \supset \Omega'$ with $|\Omega| = M$ and



$M > M'$. For example:

$$\Omega' = \left\{ \begin{array}{c} \cdot \\ \hline \cdot \end{array} \right\}, \quad \Omega = \left\{ \begin{array}{c} \times \times \times \\ \times \times \times \\ \times \times \times \end{array} \right\}$$

4-PSK $\uparrow$ $d_{\min}$ 8-PSK $\uparrow$ $d_{\min}$

$d_{\min} > d_{\min}$

Although the minimum distance between the points in Ω may be reduced from the one in Ω' , the minimum distance between the "valid" sequences in $(\Omega)^K$ may be increased from the minimum distance of all possible sequences in $(\Omega')^K$.

$\uparrow$ This is where the coding gain comes from.

Note: We only need to choose $(M')^K$ many valid sequences from the set $(\Omega)^K$ for keeping the same transmission rate
(total M^K many)

The minimum distance between all different "valid" sequences in $(\Omega)^K$ is called d_{free} . Let E denote the signal mean energy for the constellation Ω . Then, the asymptotic coding gain of a TCM scheme is



Coding gain $\gamma \triangleq \frac{d_{\text{free}}^2 / E}{d_{\text{min}}^2 / E'}$ $\leftarrow$ Coded $\leftarrow$ uncoded

where d_{min} is the minimum distance between the points in Ω' and E' is the corresponding mean energy in Ω' .

The total # of codewords of TCM is $(M')^K$ rather than M^K , which is equal to the total # of the uncoded codewords/sequences in $(\Omega')^K$.

* Two basic ideas for TCM

- ① the introduction of interdependencies among the signals and expansion of the signal set
- ② set partition (symbol assignments).

* General Model for TCM

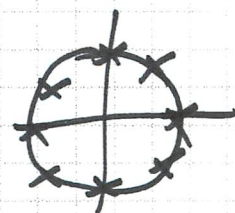
not only on the source symbol an



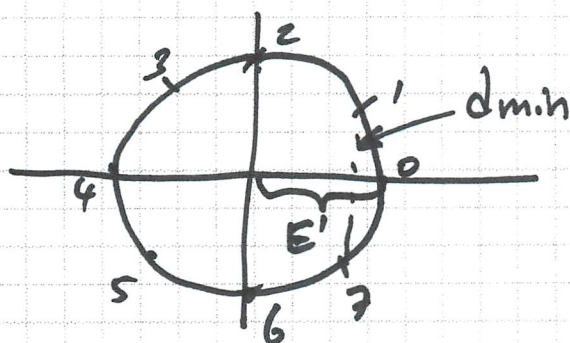
§ 7.3. Examples

Consider $M' = 4$ and $\Omega' = \left\{ \begin{array}{c} \times \\ \times \\ \times \\ \times \end{array} \right\}$ QPSK

Consider TCM schemes with $M = 2M' = 8$
 $\Omega = 8\text{-PSK}$



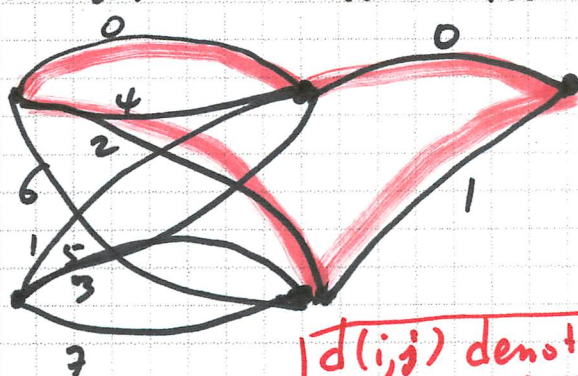
with rate $2/3$ TCM schemes



$$\frac{d_{\min}^2}{E'} = 2$$

* Example 1

Consider a two state trellis



state

S_1 associate with alphabet $\{0, 2, 4, 6\}$

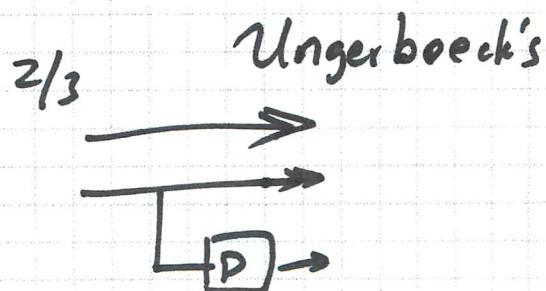
S_2 associate with alphabet $\{1, 3, 5, 7\}$

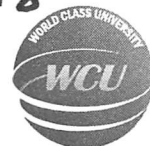
$d(i, j)$ denotes the Euclidean distance between signals i & j

$$E = E' = 1$$

$$\begin{aligned} \frac{d_{\text{free}}^2}{E} &= \frac{1}{E} [d^2(0, 2) + d^2(0, 1)] \\ &= 2 + 4 \sin^2 \frac{\pi}{8} = 2.586 \end{aligned}$$

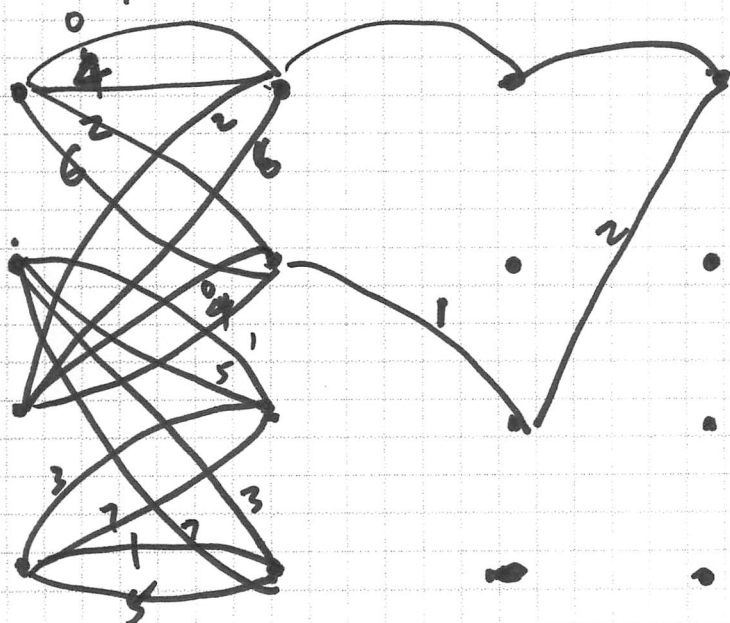
$$\gamma = \frac{2.586}{2} = 1.293 = 1.1 \text{ (dB)} \quad (\text{coding gain})$$





To increase the coding gain, we may increase the number of states

Example II. Consider 4 state trellis



Assign $\{0, 2, 4, 6\}$ to states S_1 and S_3
 $\{1, 3, 5, 7\}$ to states S_0 and S_2

$$\frac{d_{\text{free}}^2}{E} = d^2(0, 4) = 4$$

Coding gain $\gamma = \frac{4}{2} = 2 = 3 \text{ (dB)}$



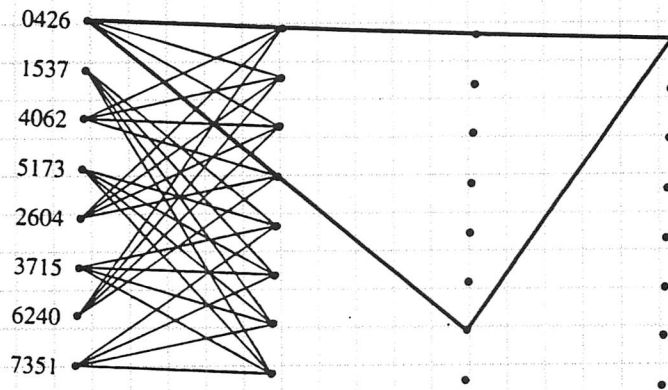
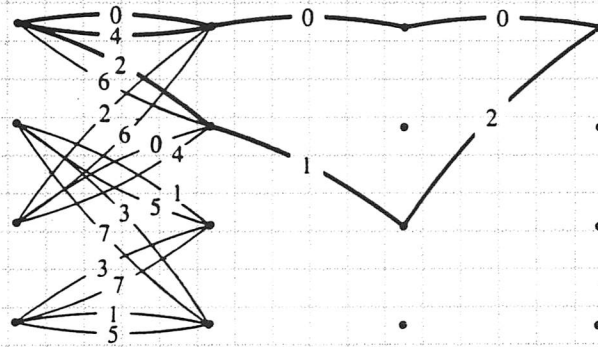
全北大學校 WCU 情報通信 事業團

WCU Center of Information & Communication, Chonbuk National University

561-756 664-1Ga Deokjin-dong, Jeonju, Korea

TEL: (063) 270-4165, FAX: (063) 270-4166, HOMEPAGE: wcu.chonbuk.ac.kr

8-19



$$\gamma = 3.6 \text{ dB}$$



general, its elements are sums of or the computation of d_{free}^2 it is not nces, since only the smallest among matrices include *only one term*, the side of (4.7) that has the smallest

ritten in the form

$$d_{\text{next}})D^{d_{\text{next}}^2} + \dots \quad (4.51)$$

distance. In [6, p. 127] it is shown ecreases monotonically to the limit $\phi_1(D)$ will provide an upper bound ring the function

$$\frac{D}{1}.$$

51). We have

$$+ \frac{N(d_{\text{free}})}{N(d_{\text{next}})} D^{d_{\text{next}}^2 - d_{\text{free}}^2} + \dots$$

$$\epsilon(D) \quad (4.52)$$

ls to zero monotonically as $D \rightarrow 0$. es of D , from $\phi_1(D)$ and $\phi_2(D)$, r to each other, and hence to d_{free}^2 . $V(d_{\text{free}}^2)$ [3].

a good deal of simplification arises w the use of *scalar* branch labels ated by using scalar branch labels is case the labels are the sums of As before, since only the smallest *one term* of the summation—the

correspond to the channel symbols trellis, we must pick the minimum-

$$\in \mathcal{C}_0$$

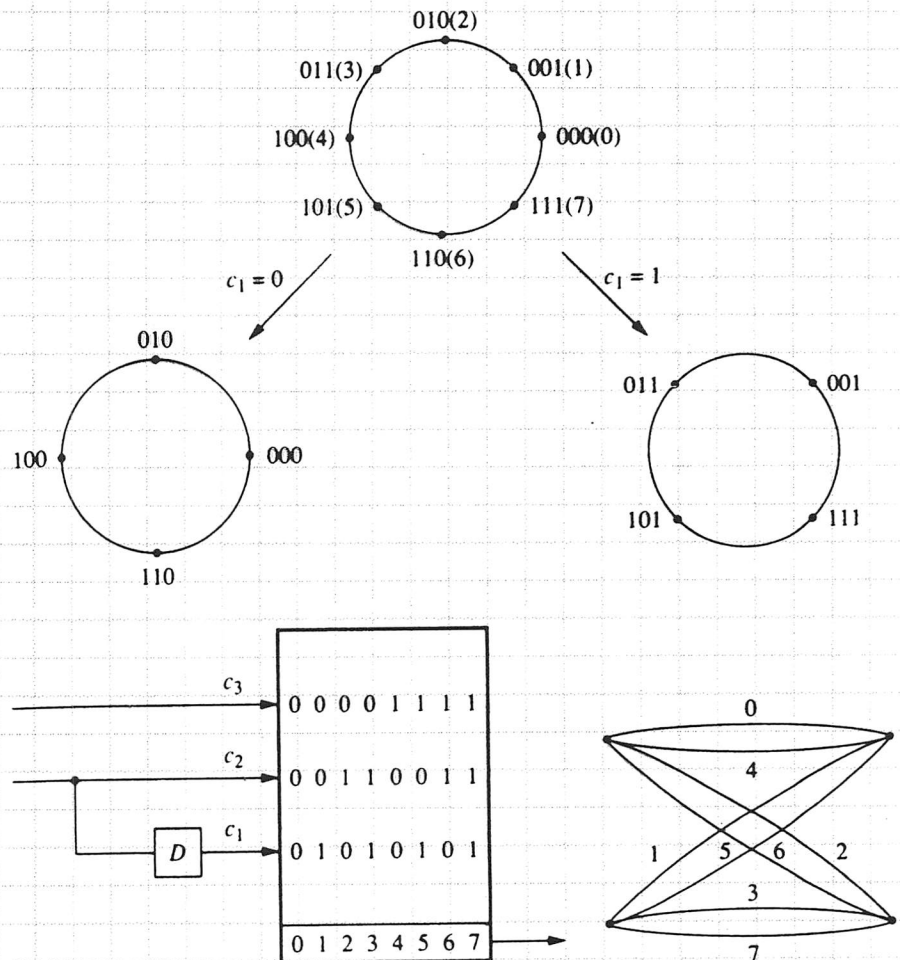


FIGURE 4.20 TCM scheme with $m = 2$, two states, and based on 8-PSK.

where as before $\mathcal{C}_0$ is the set of 2^m vectors $\mathbf{c}$ corresponding to the all-zero state of the convolutional code. (Of course, if we choose any other state, the result will not change since we assume that the uniformity conditions are satisfied.) Thus we can define the function, depending only on the error vector $\mathbf{e}$:

$$w(\mathbf{e}) = \min_{\mathbf{c} \in \mathcal{C}_0} \|f(\mathbf{c}) - f(\mathbf{c} \oplus \mathbf{e})\|^2$$

(this is called the *Euclidean weight* of $\mathbf{e}$), and label the nodes of the error-state diagram with the values $D^{w^2(\mathbf{e})}$.

EXAMPLE 4.2

Consider the TCM scheme of Fig. 4.20. First, observe that the uniformity conditions are satisfied by this scheme. The parallel transitions give a squared Euclidean



8-21

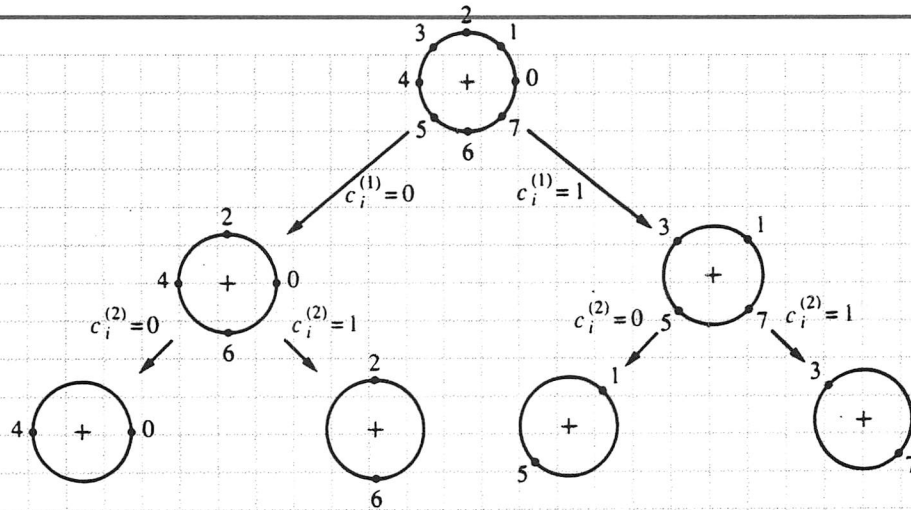


FIGURE 3.11 Set partition of an 8-PSK constellation.

One dimensional

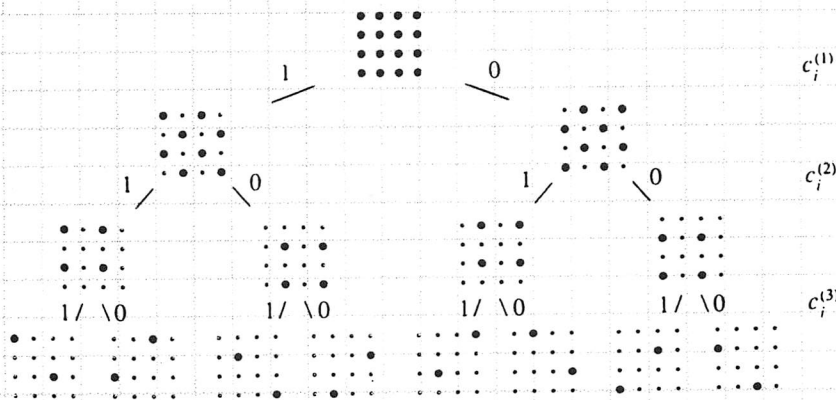


FIGURE 3.12 Set partition of a 16-QAM constellation.

two dimensional

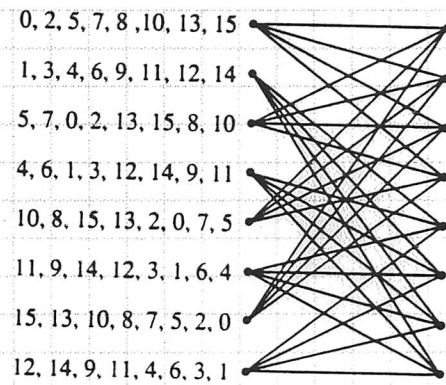


FIGURE 3.9 TCM scheme based on an eight-state trellis, $M' = 8$, and $M = 16$.

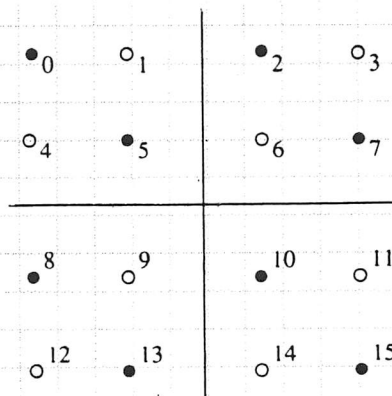
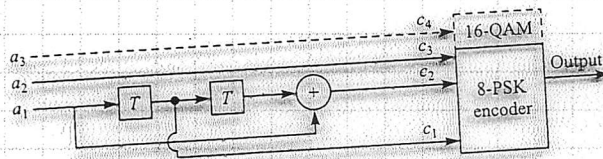
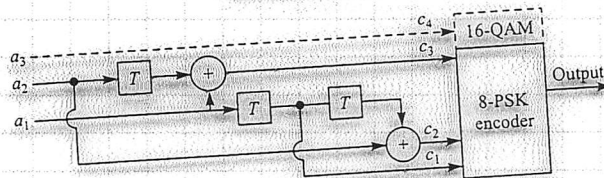


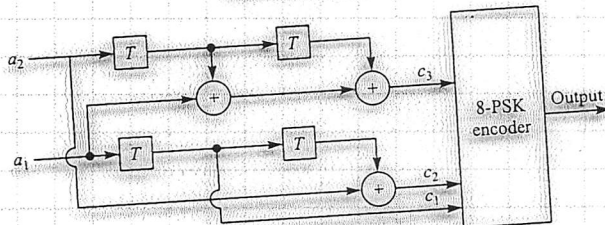
FIGURE 3.8 Octonary AM-PM alphabet.



(a) 4-state encoder

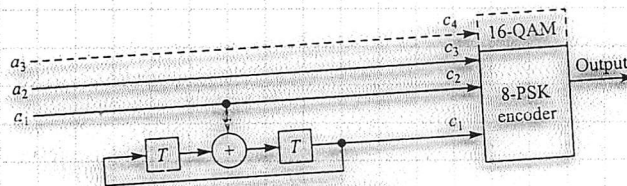


(b) 8-state encoder

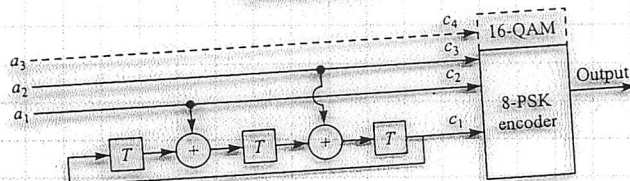


(c) 16-state encoder

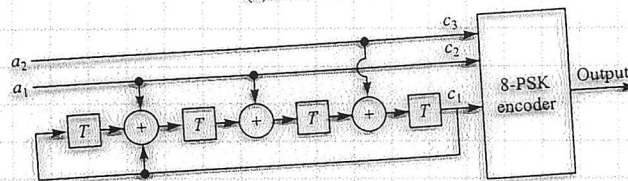
FIGURE 8.3-11
Minimal feedback-free
convolutional encoders for
8-PSK and 16-QAM signals.
[From Ungerboeck (1982).
© 1982 IEEE.]



(a) 4-state encoder



(b) 8-state encoder



(c) 16-state encoder

FIGURE 8.3-12
Equivalent realizations of
systematic convolutional
encoders with feedback for
8-PSK and 16-QAM.
[From Ungerboeck (1982).
© 1982 IEEE.]



8-24

TABLE 8.3-1
Coding gains for trellis-coded PAM signals

Number of states	k_1	Code rate	$m = 1$	$m = 2$	$m \rightarrow \infty$	N_{fed}
		$\frac{k_1}{k_1 + 1}$	coding gain (dB) of 4-PAM versus uncoded 2-PAM	coding gain (dB) of 8-PAM versus uncoded 4-PAM	asymptotic coding gain (dB)	
4	1	1/2	2.55	3.31	3.52	4
8	1	1/2	3.01	3.77	3.97	4
16	1	1/2	3.42	4.18	4.39	8
32	1	1/2	4.15	4.91	5.11	12
64	1	1/2	4.47	5.23	5.44	36
128	1	1/2	5.05	5.81	6.02	66

Source: Ungerboeck (1987).

TABLE 8.3-2
Coding gains for trellis-coded 16-PSK modulation

Number of states	k_1	Code rate	$m = 3$	N_{fed}
		$\frac{k_1}{k_1 + 1}$	coding gain (dB) of 16-PSK versus uncoded 8-PSK	
4	1	1/2	3.54	4
8	1	1/2	4.01	4
16	1	1/2	4.44	8
32	1	1/2	5.13	8
64	1	1/2	5.33	2
128	1	1/2	5.33	2
256	2	2/3	5.51	8

Source: Ungerboeck (1987).

TABLE 8.3-3
Coding gains for trellis-coded QAM modulation

Number of states	k_1	Code rate	$m = 3$	$m = 4$	$m = 5$	$m = \infty$	N_{fed}
		$\frac{k_1}{k_1 + 1}$	gain (dB) of 16-QAM versus uncoded 8-QAM	gain (dB) of 32-QAM versus uncoded 16-QAM	gain (dB) of 64-QAM versus uncoded 32-QAM	asymptotic coding gain (dB)	
4	1	1/2	3.01	3.01	2.80	3.01	4
8	2	2/3	3.98	3.98	3.77	3.98	16
16	2	2/3	4.77	4.77	4.56	4.77	56
32	2	2/3	4.77	4.77	4.56	4.77	16
64	2	2/3	5.44	5.44	5.23	5.44	56
128	2	2/3	6.02	6.02	5.81	6.02	344
256	2	2/3	6.02	6.02	5.81	6.02	44