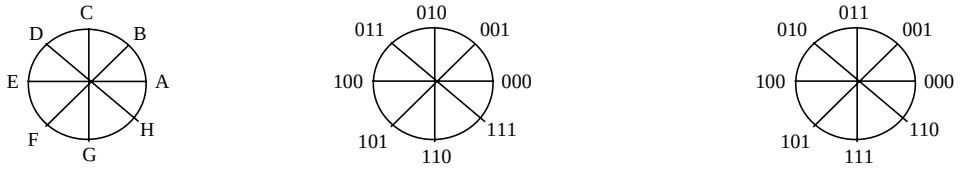
Trellis Coded Modulation

Trellis coded modulation (TCM) is a marriage between codes that live on trellises and signal designs. We have already seen that trellises are the preferred way to view convolutional codes and that even block codes can be thought of as having a trellis structure. What is new in TCM is the incorporation of some properties of the signal design for a modulation waveform along with the error correction capability of trellis codes.

We illustrate the idea by an example. Consider an 8-PSK modulation system with the 8 phases being denoted A, B, C, D, E, F, G and H as shown in Figure 44(a) below. Two different methods of assigning binary digits to the 8 phases are also shown in Figure 44(b) and 44(c).



(a) (b) (c)

Figure 44. 8-PSK Modulation.

For the moment, assume that signal phase A (0°) was transmitted and that no coding is used. In a high signal-to-noise ratio AWGN channel the receiver might make an error but if it did it would probably choose phases B or H. Next most likely for the choice of the detector are the phases C and G and after that comes the pair D and F. The most unlikely choice for the decision by the detector (given that phase A was transmitted) would be phase E. The probability of mistaking one phase decreases exponentially with the squared Euclidean distance between the two phases. Assuming the points are on a circle of radius 1, the squared Euclidean distances between all pairs of phases is listed in the table below. The key point to be observed is how the spacing of these 8 points in itself protects against certain types of errors.

	A	B	C	D	E	F	G	H
A	0	.586	2	3.414	4	3.414	2	.586
B	.586*	0	.586	2	3.414	4	3.414	2
C	2	.586	0	.586	2	3.414	4	3.414
D	3.414	2	.586	0	586	2	3.414	4
E	4	3.414	2	.586	0	.586	2	3.414
F	3.414	4	3.414	2	.586	0	.586	2
G	2	3.414	4	3.414	2	.586	0	.586
H	.586	2	3.414	4	3.414	2	.586	0

*$(2\sin 22.5^0)^2 = .586$

Ungerboeck in a now classic paper ("Channel Coding with Multilevel Phase Signaling," *IEEE Transactions on Magnetics*, Vol. 25, pp. 55-67, January 1982) explained how to marry trellis based error correcting codes and signal constellations such as 8-PSK modulation. He used a construction now called "set partitioning" based upon the bit assignment for the phases shown in Figure 44(b). A simpler to understand construction (at least for me) is known as the "pragmatic" approach and is based upon the binary assignment in Figure 44(c). What differentiates Figure 44(c) from Figure 44(b) is how the two least significant bits are assigned to the phases. In Figure 44(c), for the most probable error where a phase is mistaken for its nearest neighbor, only one of the two least significant bits are changed. It takes a phase error of 90° to change both of the two least significant bit.

The most important characteristic of the pragmatic approach is that an already existing Viterbi decoder matched to a convolutional code can be modified slightly to decode the trellis coded 8-PSK modulation. The details of the decoder will be discussed after we explain the pragmatic approach.

The example we will pursue is that of a rate R = 2/3 trellis code combined with 8-PSK modulation. In this example, two information bits will be passed through a convolutional encoder resulting in 3 coded bits which then will be mapped to one of the 8 phases of an 8-PSK modulation scheme using the mapping of Figure 44(c). Since 2 information bits are transmitted for every 8-PSK transmitted signal, the bandwidth efficiency of the system is 2 bits/sec/Hz. We will compare the performance of this system with uncoded QPSK since both have the same bandwidth efficiency. An AWGN channel will be assumed .

The basic idea behind the pragmatic approach is to protect the two least significant bits by using an ordinary rate 1/2 convolutional code with soft inputs. We note that if our decoder produces the correct values for these least significant bits, then it will be highly unlikely that the most significant bit is in error. This is the case since the two possibilities for the most significant bit correspond to phases that are 180° apart and thus are highly unlikely to be confused with each other.

The encoder for the pragmatic TCM code for this example is the "seemingly stupid" rate 2/3 convolutional encoder shown in Figure 34. The uncoded bit ($V^{(1)}(D) = U^{(1)}(D)$) is the most significant bit in the mapping and the coded outputs, ($V^{(2)}(D)$ and $V^{(3)}(D)$) become the two least significant bits. The trellis diagram for a 4-state version of the code is shown in Figure 35. Here we are interested in the case where the rate 1/2 code is the 64-state convolutional code with $G_1(D) = 1+D^2+D^3+D^5+D^6$ and $G_2(D) = 1+D+D^2+D^3+D^6$ with free Hamming distance 10. The trellis diagram for this rate 2/3 code is too busy to draw but is similar to that of the 64-state rate 1/2 convolutional code except that it has two parallel branches every place that the rate 1/2 code has a single branch.

The paths in the trellis specify the allowable sequences of transmitted symbols. In the case of 8-PSK modulation, the transmitted sequence is a sequence of phases. Thus, the noiseless signal associated with each branch of the trellis is one of these 8 phases. Thus, one could either think of the branches of the trellis as being labeled by the 3 binary digits produced by the encoder or by one of the 8 phases that correspond to those 3 binary digits. Note that since a pair

of parallel transitions differ in only the most significant bit, the phases associated with this pair of parallel transitions differ from each other by 180°.

So much for encoding. We now consider the decoding operation. When one transmits one of these 8 phases over an AWGN channel, the noisy received signal is first mapped by the front end of the decoder into a point in the plane. We call this point $\underline{R} = (R_x, R_y)$. If there were no noise, the point would be the point corresponding to the transmitted phase. With noise, the point will be displaced from the transmitted phase by a noise vector. This is shown in Figure 45(a). The proper branch matrix to be used by the Viterbi decoder is the squared Euclidean distance between the phase corresponding to that branch and the received point $\underline{R}$. An example of these distances is shown in Figure 45(b)

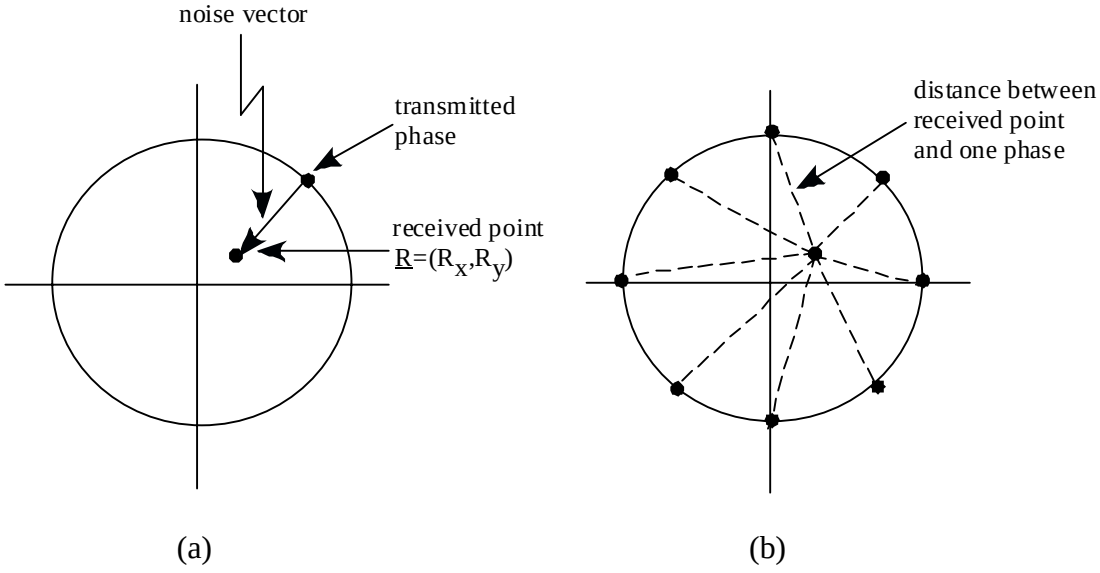


Figure 45. Effects of Noise on 8-PSK and Euclidean Distances.

Figure 46 shows the squared Euclidean distance between the received point $\underline{R}$ and the phases associated with a pair of phases on parallel branches.

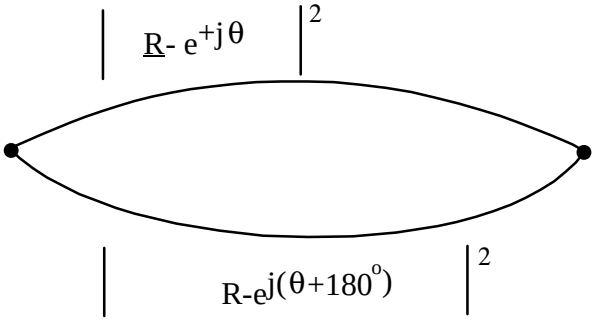


Figure 46. Squared Euclidean Distances for a Pair of Parallel Branches.

It should be noted that when Viterbi decoding is applied for every pair of parallel branches, only the branch with the smaller squared Euclidean is used. However, we must remember which of these two branches is used at every step in the winning path. This is how the trellis code protects the most significant bit (which we referred to as being uncoded).

Let us now seek the closest two paths on the trellis that start in a common state and end in a common (but perhaps different) state. By closest we mean the sum of the squared Euclidean distances between the points on the corresponding paths on the two paths is a minimum. One possible choice for this pair of points are the parallel paths. Since the points corresponding to a pair of parallel points differ by 180°, they are separated by a diameter of the circle. Since the radius of the circle is equal to 1, the diameter is equal to 2 and the square of this diameter is equal to 4. Let us now try to find the sum of the squares of the Euclidean distances for a pair of paths that start and end at a common state but are not parallel paths. We first note that when two paths diverge from a state in the trellis for the 64-state rate 1/2 convolutional code, the two bits on one branch are the complements of the two bits on the other branch. The same is true for two paths that come together at a state. This is shown in Figure 47. The former is true since if the uncoded bit entering the encoder is complemented, both of the coded output bits are complemented. The latter is true since if the least significant bit of the state of the encoder is complemented, both of the coded output bits are complemented. This translates to the fact that a pair of signal points on diverging branches differ by 90°. The same is true for the pair of signal points on converging branches. The squared Euclidean distance between any two signal points that differ by 90° on a circle of radius 1 is equal to 2. Thus, the sum of the squares of the Euclidean distances between any of the non-parallel paths will be more than 4 since the Hamming distance between the corresponding two code words of the 64-state, rate 1/2 code is at least 10 and we have only accounted for a Hamming distance of 4. The remaining Hamming distance of 6 will produce at least 6 (.586) = 3.515 additional squared Euclidean distance. We need not concern ourselves with this since we have now found that the parallel branches which give squared Euclidean distance of 4 yield the closest code sequences.

Figure 47. Diverging and Converging Branches for Trellis Branches in 64-State, Rate 1/2, Convolutional Code.

We now compare the performance of this system with uncoded QPSK. As stated previously, both systems transmit two information bits per transmitted phase. The two closest code symbols in uncoded QPSK differ by 90° so have a squared Euclidean distance of 2. In our rate 2/3 trellis coded 8-PSK, the squared Euclidean distance between any pair of transmitted phases is equal to 4. This is a 3 dB gain in the required E_b/N_o. This gain is termed the asymptotic coding gain (ACG) in that it only applies at very low error probabilities. In actuality, at a bit error rate of 10^{-5}, this TCM scheme actually has a 3.2 dB coding gain.

It should be realized that using a 64-state 1/2 convolutional code was not at all necessary to achieve the 3 db ACG. Any rate 1/2 convolutional code with free distance at least equal to 4 would suffice. Thus, the 4-state code shown in Figure 14 with free distance 5 could have been used. The reason that the 64-state code was used in this example was that a VLSI decoder already existed for this system. By a slight modification to this decoder a VLSI decoder (Q 1875) for the TCM system was made available by Qualcomm. This chip was incorporated in many systems by other communication companies. Although pragmatic coding was explained in terms of 8-PSK modulation, the scheme applies to other modulation formats. In particular, the Q 1875 decoder has been used in conjunction with 16-PSK modulation, M-level amplitude shift keying (MASK) and M-point quadrature amplitude modulation (M-QAM). A detail left out in the above discussion of 8-PSK was how phase synchronization was achieved. The details of this are given in a patent (J. K. Wolf, Patent 5,233,630 – "Method and Apparatus for Resolving Phase Ambiguities in Trellis Coded Modulation Data", August 3, 1993).

As stated previously, better trellis coded modulation codes are obtained by a technique called set partitioning which was introduced by Ungerboeck. Indeed the best 64-state rate 2/3 TCM code for 8-PSK given by Ungerboeck achieves an ACG of 5 dB and a coding gain of 3.6 dB at a bit error rate of 10^{-5}. This scheme uses the binary mapping to the 8-PSK phases shown in Figure 44(b). The TCM scheme employs binary convolutional codes that do not necessarily have good Hamming distance but lead to TCM codes with good squared Euclidean distance. The basic difference between the two design strategies can be seen by looking at the difference between the two binary mappings given in Figure 44. In the mapping used for the pragmatic approach there is a relationship between the squared Euclidean distance between two signal points and the Hamming distance between the two least significant bits. That is, the two closest neighbors to a signal point had the two least significant bits differing in just one position while the next two closest neighbors had the two least significant bits differing in both positions. This is not the case for the mapping used by Ungerboeck. Thus, the TCM codes found by Ungerboeck make use of strange convolutional codes; that do not have good Hamming distance.

The basic idea of set partitioning is very easy to understand. We begin with a signal constellation (such as 8-PSK) and successively divide it in half into subsets with increasing minimum Euclidean distance. If Δ_0 is the minimum Euclidean distance of the original signal constellation, then Δ_1 is the minimum Euclidean distance after the first partition, Δ_2 the minimum Euclidean distance after the second partition, etc. where $\Delta_0 < \Delta_1 < \Delta_2 <\ldots$. The concept is illustrated in Figure 48 for 8-PSK modulation. Note that this set partitioning yields a labeling for the 8 points ($y^{(2)}\, y^{(1)}\, y^{0)}$), which is the same as in Figure 44(b).

Uncoded QPSK modulation can be considered as a 1-state trellis code with parallel transitions where the signals are chosen either from subset B0 or subset B1. The trellis for this 1-state code, assuming the signals are chosen from subset B0, is shown in Figure 49(a). The minimum-squared Euclidean distance between any pair of output sequences is $\Delta_1^2 = 2$, the squared distance between parallel transitions. A rate 2/3 2-state TCM code for 8-PSK is shown in Figure 49(b). Now the squared Euclidean distance between parallel transitions is $\Delta_2^2 = 4$ and the squared Euclidean distance between a pair of non-parallel paths that start in a common state and end in a common state is $\left(\sqrt{2}\right)^2 + (.7654)^2 = 2.585$. An example of a pair of paths that have squared Euclidean distance 2.585 is shown in Figure 50.

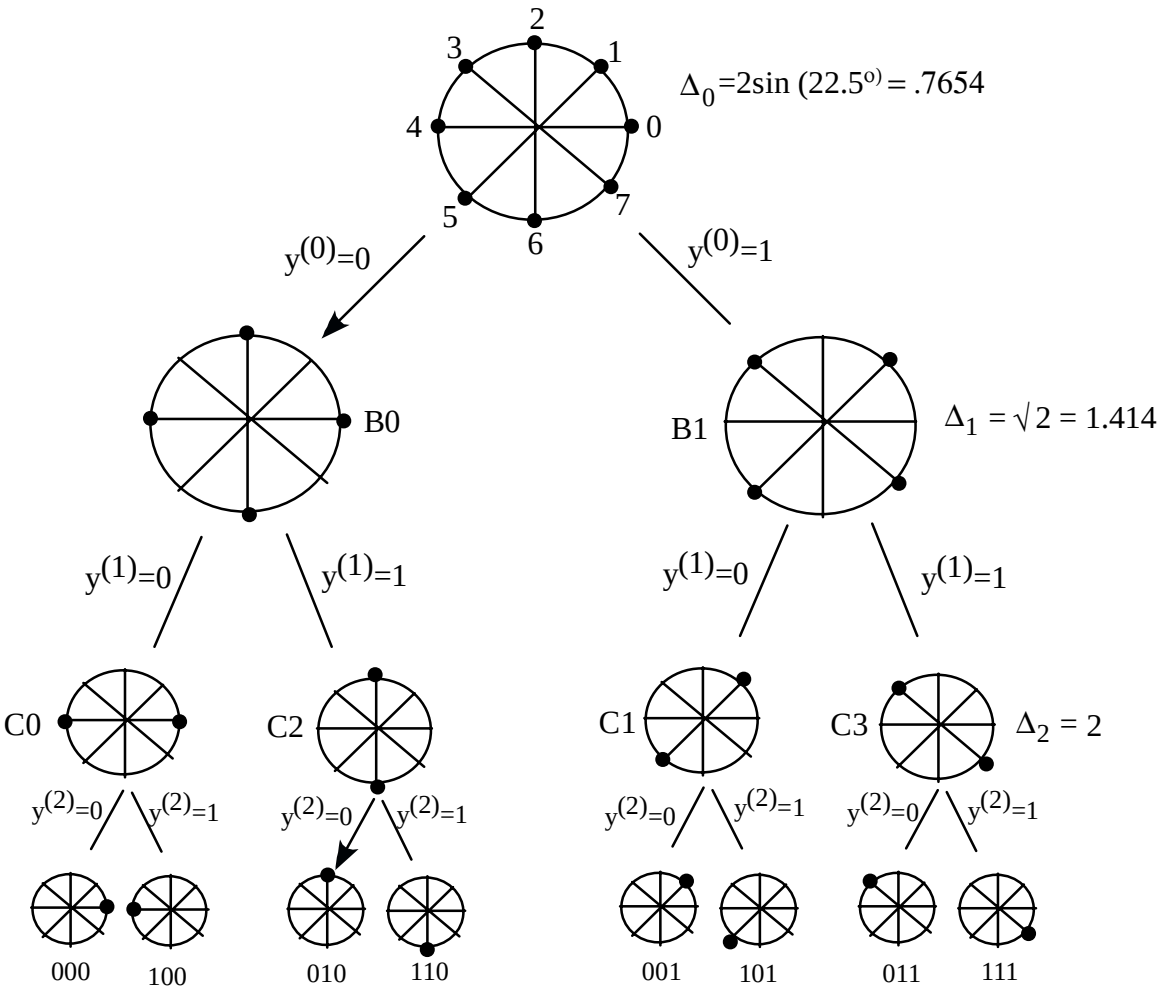
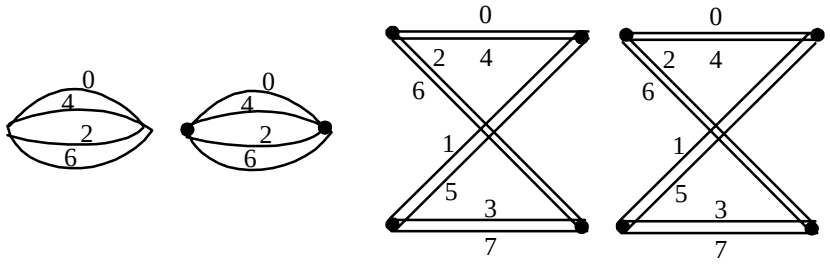


Figure 48. Set Partitioning for 8-PSK Modulation.



(a) Trellis for QPSK (b) Trellis for 2-State TCM 8-PSK.

Figure 49. Some Trellises for TCM Codes.

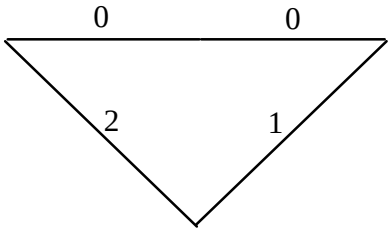


Figure 50. A Pair of Minimum Distance Paths from Trellis of Figure 49(b).

The ACG for this code is then $10 \log_{10} \frac{2.585}{2.000} = 1.12$ dB. The four state rate 2/3 code given in Figure 51 has an ACG of 3 dB since the minimum squared Euclidean distance for any pair of paths starting in a common state and ending in a common state is $\Delta_2^2 = 4$.

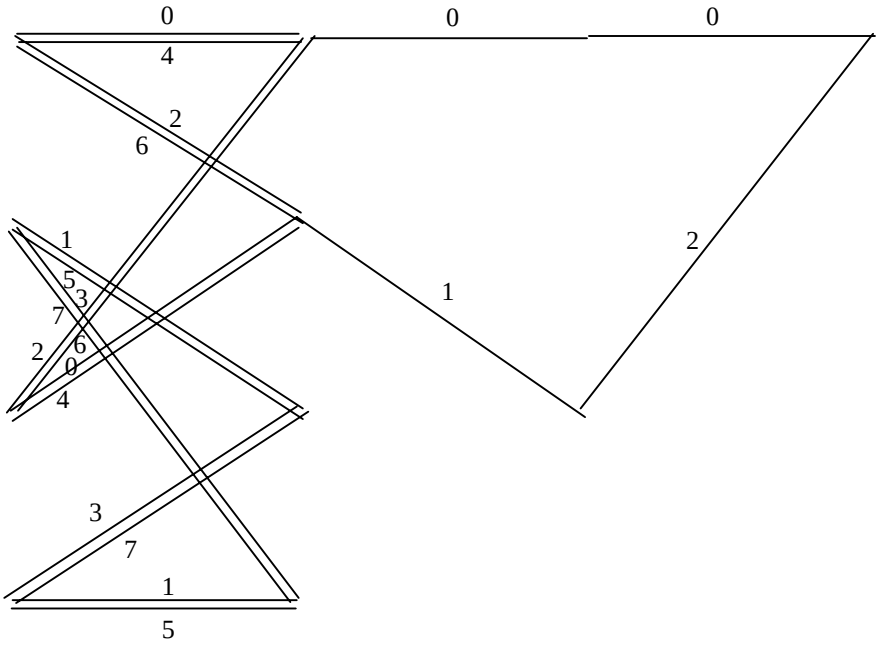


Figure 51. 4-State Rate 2/3 TCM Code for 8-PSK with $d_{free}^2 = 4.0$.

With an 8-state code, the best rate 2/3 TCM trellis has $d^2_{free} = 2\Delta_1^{\,2} + \Delta_0^{\,2} = 4.5839$ which gives an ACG of 3.6 dB. A 16-state rate 2/3 code was found with $d^2_{free} = 2\Delta_1^{\,2} + 2\Delta_0^{\,2} = 5.171$ and ACG of 4.126 dB. The 8- and 16-state codes do not have any parallel transitions. Codes with parallel transitions have at most a 3 dB ACG. (Why?) The following are a set of rules given by Ungerboeck that seem to apply to the best codes.

" 1) All 8-PSK signals occur with equal frequency and with a fair amount of regularity and symmetry,
2) transitions originating from the same state receive signals either from subset B_0 or B_1,
3) transitions joining in the same state receive signals either from subset B_0 or B_1,
4) parallel transitions receive signals either from subset C_0 or C_1 or C_2 or C_3."

We note that with a 2-state trellis, rules 2 and 3 could not be satisfied simultaneously.

An encoder for the 4-state trellis shown in Figure 51 is given in Figure 52. A slightly different labeling of the outputs is adopted to agree with Ungerboeck's notation.

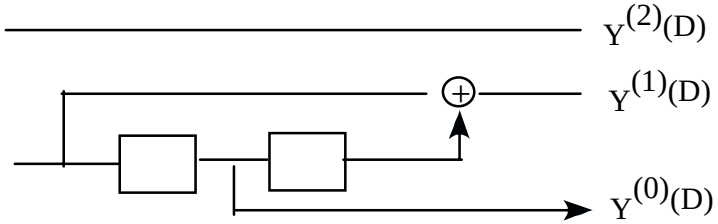


Figure 52. Encoder for 4-state TCM Trellis of Figure 51.

The 4-state rate 1/2 encoder which produces the pair $(Y^{(1)}, Y^{(0)})$ has free Hamming distance only equal to 3 and illustrates the fact that the TCM encoder does not utilize an encoder with good Hamming distance. A systematic form for this TCM encoder is shown in Figure 53.

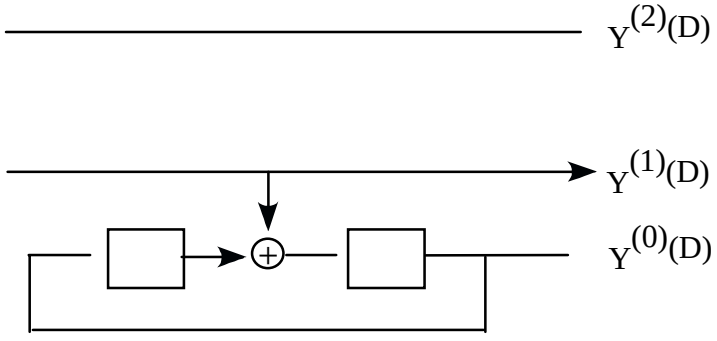


Figure 53. Systematic Form for Encoder for Figure 52.

Ungerboeck describes the encoders for TCM in terms of a parity check polynomial matrix. We first point out that a k_0 input by n_0 output encoder for TCM is a convolutional encoder defined in the usual way by the generator polynomial matrix $\underset{\sim}{G}$ given as:

$$\underset{\sim}{G} = \begin{bmatrix} G_{11}(D) & \dots & G_{1n_0}(D) \\ G_{k_0 1}(D) & \dots & G_{k_0 n_0}(D) \end{bmatrix}.$$

For example, the generator polynomial describing the encoder shown in Figure 52 is given as:

$$\underset{\sim}{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1+D^2 & D \end{bmatrix}.$$

Now one can define for any encoder for a convolutional code with k_0-inputs by n_0-outputs a parity check polynomial, H, with $(n_0\text{-}k_0)$ rows and n_0 columns defined as:

$$\underset{\sim}{G}\ \underset{\sim}{H}^t = \underset{\sim}{O}\ .$$

Thus, for the encoder shown in Figure 52, $\underset{\sim}{H}$ is given as:

$$\underset{\sim}{H} = \begin{bmatrix} 0 & D & 1+D^2 \end{bmatrix}.$$

The systematic form for this encoder, shown in Figure 53, has $\underset{\sim}{G}$ satisfying the equation:

$$\underset{\sim}{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \dfrac{D}{1+D^2} \end{bmatrix}.$$

The encoder for the TCM codes described by Ungerboeck have one more output than input. Ungerboeck denotes k_0 as m so the codes are of rate $R = \dfrac{m}{m+1}$. In general $m - \tilde{m}$ of the input bits are uncoded and pass directly to the output of the encoder. Ungerboeck denotes the polynomial components of $\underset{\sim}{H}$ in accordance with the equation:

$$\underset{\sim}{H} = \begin{bmatrix} H^{(m)}(D) & \dots & H^{(1)}(D) & H^{(0)}(D) \end{bmatrix}.$$

It can be shown that $H^{(j)}(D) = 0, \qquad \tilde{m} < j \le m$. Ungerboeck only considered codes for which

$$H^{(j)}(D) = 0 + h^{j}_{v-1}D^{v-1} + \dots + h^{(j)}_{1}D + 0, \qquad 1 \le j \le \tilde{m}$$
$$H^{(0)}(D) = D^{v} + h^{(0)}_{v-1}D^{v-1} + \dots + h^{(0)}_{1}D + 1.$$

The reason for this will be explained later.

For any encoder with $\underset{\sim}{H} = \left[H^{(m)}(D) \ \ldots \ H^{(1)}(D) \ \ H^{(0)}(D) \right]$ we now show that the systematic form for the encoder will have a generator polynomial matrix $\underset{\sim}{G}$ given by:

$$\underset{\sim}{G} = \begin{bmatrix} & H^{(m)}(D)/H^{(0)}(D) \\ & \vdots \\ \underset{\sim}{1}_{m\times m} & \vdots \\ & \vdots \\ & \vdots \\ & H^{(1)}(D)/H^{(0)}(D) \end{bmatrix} .$$

This follows since:

$$\begin{bmatrix} & H^{(m)}(D)/H^{(0)}(D) \\ & \vdots \\ \underset{\sim}{1}_{m\times m} & \vdots \\ & \vdots \\ & H^{(1)}(D)/H^{(0)}(D) \end{bmatrix} \begin{bmatrix} H^{(m)}(D)/H^{(0)}(D) \\ \vdots \\ \vdots \\ \vdots \\ H^{(1)}(D)/H^{(0)}(D) \\ 1 \end{bmatrix} = \underset{\sim}{0}$$

or

$$\begin{bmatrix} & H^{(m)}(D)/H^{(0)}(D) \\ & \vdots \\ \underset{\sim}{1}_{m\times m} & \vdots \\ & \vdots \\ & H^{(1)}(D)/H^{(0)}(D) \end{bmatrix} \begin{bmatrix} H^{(m)}(D) \\ \vdots \\ \vdots \\ \vdots \\ H^{(1)}(D) \\ H^{(0)}(D) \end{bmatrix} = \underset{\sim}{0} .$$

A general form for the systematic encoder is shown in Figure 54.

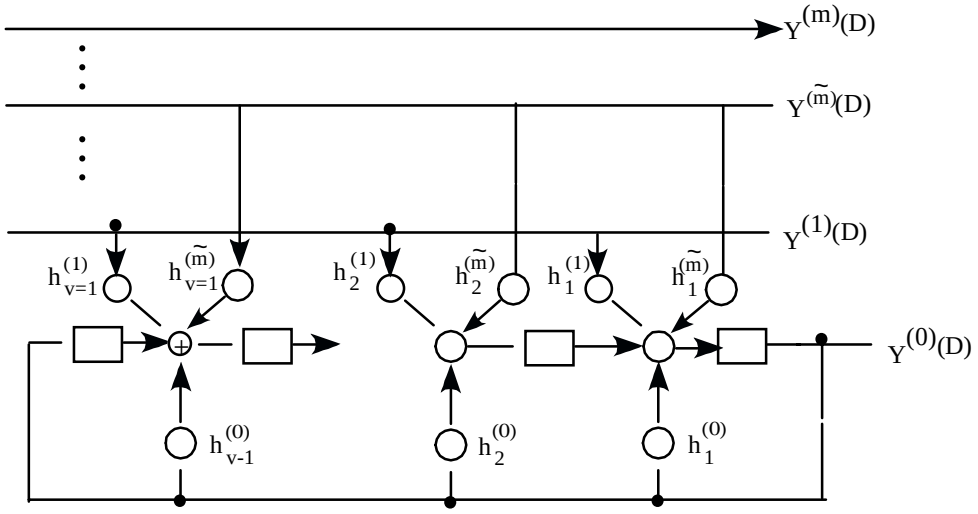


Figure 54. Systematic Form of TCM Encoder.

From Figure 54, we can now see why Ungerboeck only considered codes for which $h_0^{(j)} = h_v^{(j)} = \begin{cases} 0 & j \neq 0 \\ 1, & j = 0 \end{cases}$. Assume that the encoder is in state $\underline{S} = (S_{v-1}, S_{v-2}, \ldots, S_1, S_0)$ where S_0 is the content of the right-most stage of the shift register. The (parity) bit on level $Y^{(0)}(D)$ will then be independent of the information bits to be encoded and equal to S_0. This says that the value of $y^{(0)}$ will be the same on all branches diverging from any state in the trellis. Referring to Figure 48 where set partitioning is defined, we then see that diverging paths leaving any state have signals only from subset B0 or B1. Thus, any two paths that diverge in any state picks up a square Euclidean distance of at least Δ_1^2 from this divergence.

Now notice that because of the feedback path, the next state $\underline{S}^1 = \left(S^1_{v-1}, S^1_{v-2}, \ldots, S^1_1, S^1_0\right)$ will be such that $S^1_{v-1}=S_0$. The other values of the state variables will depend upon the information bits, but every path that ends up in a particular state $\underline{S}^1 = \left(S^1_{v-1}, \text{a b c} \ldots\right)$ will have to contain the parity bit S^1_{v-1}. This then says that the signal points corresponding to all paths converging on a given node will all be from the subset B0 or B1 and thus contribute at least another Δ_1^2 to the squared Euclidean distance.

The advantage of dealing with the parity check polynomials is that there are only $(\tilde{m}+1)\,(v-1)$ coefficients in the parity check polynomials to be determined. A computer search was used to find the best codes. Ungerboeck gave a set of rules to use in making this computer search. Among other simplifications, Ungerboeck showed that one only consider the cases $\tilde{m}=1$ or $\tilde{m}=2$. The result of his search for 8-PSK is shown in the table below and are taken directly from his paper. An octal representation is used to simplify the parity checks polynomials. For example, $H^{(0)}(D)=D^5+D^2+1 \Leftrightarrow (100\ 101) \Leftrightarrow 4\ 5$. The squared free distance of

the code is first given in the normalized form $d_{free}^2 / \Delta_1^2(8-\text{PSK})$. But $\Delta_1^2(8-\text{PSK}) = \Delta_0^2(4-\text{PSK})$ so that this ratio is the ACG shown in the last column of the following table.

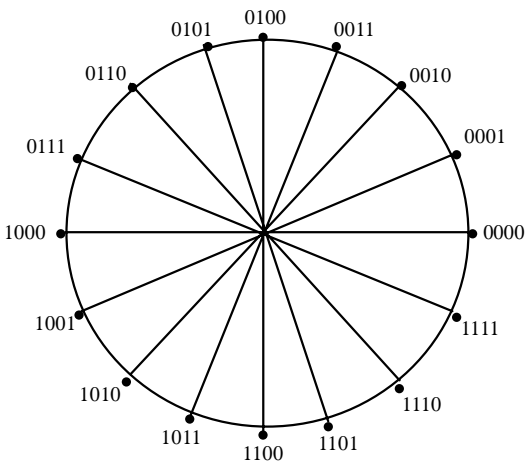
R = 2/3 CODES FOR 8-PSK MODULATION						
ν	$\tilde{m}$	$H^0(D)$	$H^1(D)$	$H^2(D)$	d_{free}^2 / Δ_1^2	$C_{8PSK/4PSK}^{m=2}$
2	1	5_8	2_8	-	2.000	~ 3.0
3	2	11	02	04	2.293	3.6
4	2	23	04	16	2.586	4.1
5	2	45	16	34	2.879	4.6
6	2	105	036	074	3.000	4.8
7	2	203	014	016	3.172	5.0
8	2	405	250	176	3.465	5.4
9	2	1007	0164	0260	3.758	5.7
$\overset{*}{\Delta}$ 10	2	2003	0164	0770	3.758	5.7
$E\{\lvert a_n^2 \rvert\} = 1$: $\Delta_0(4PSK) = \sqrt{2},\ \Delta_0(8PSK) = 2\sin(\pi/8),$ $\Delta_1(8PSK) = \sqrt{2},\ \Delta_2(8PSK) = 2$					$\Delta_1^2(8PSK)/\Delta_0^2(4PSK) = 1$ (0 dB)	

* Search not completed.

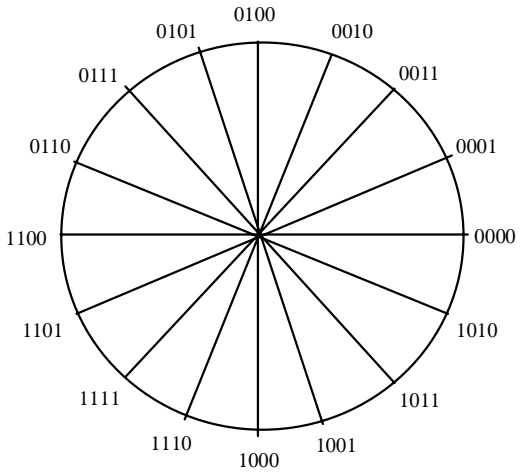
Δ No improvement obtained.

d_{free}^1 / Δ_1^2

TCM for 16-PSK is a natural extension of TCM for 8-PSK. Again, two different binary encodings are used, one for coding by set partitioning and the other for the pragmatic approach. These are shown in Figures 55(a) and (b), respectively.



(a) For Set Partitioning.



(b) For Pragmatic Approach.

Figure 55. Binary Mapping to Phases of 16-PSK.

In the pragmatic approach, the encoder accepts 3 information bits and produces 4 coded bits which are mapped to a 16-PSK signal in accordance with the mapping of Figure 55(b). Two of the 3 input information bits are uncoded and become 2 of the coded bits. The third information bit is passed through a rate 1/2 convolutional encoder with good Hamming distance. If the encoder for the rate 1/2 convolutional code has 2^s states, the trellis for the TCM code will also have 2^s states. It will look exactly like the trellis of the rate 1/2 convolutional code but now there will be 4 parallel branches wherever the trellis for the rate 1/2 encoder had a single branch.

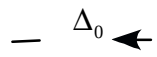
If the 64-state *de facto* standard rate 1/2 convolutional code is used with $G_1(D) = 1 + D^2 + D^3 + D^5 + D^6$ and $G_2(D) = 1 + D + D^2 + D^3 + D^6$, the squared Euclidean free distance of the TCM code will be the minimum squared Euclidean distance between parallel transitions. This follows from a careful examination of how Hamming distances on the branches relate to Euclidean distances. In Ungerboeck's notation, this is $\Delta_2^2(16-\mathrm{PSK}) = 2$. Since this TCM system transmits 3 information bits for each phase transmitted it should be compared with uncoded 8-PSK transmission. In Ungerboeck's notation the squared Euclidean distance for uncoded 8-PSK, $\Delta_0^2(8-\mathrm{PSK}) = \Delta_1^2(16-\mathrm{PSK}) = .5858$ The ACG of this TCM system over uncoded 8-PSK is then $10 \log_{10}(2/.5858) = 5.333$ dB. The actual measured coding gain at a bit error rate of 10^{-5} is 3.5 dB. The best Ungerboeck codes with 64-states have comparable performance.

Both the pragmatic approach and set partitioning can be used to construct TCM codes for Quadrature Amplitude Modulation (QAM).* However, in this case the two approaches lead to different codes since a straightforward application of the pragmatic approach yields codes of rate $R = \frac{n-2}{n}$ while set partitioning yields codes of rate $R = \frac{n-1}{n}$. The pragmatic approach has been further modified to yield codes of rate $R = \frac{n-1}{n}$ but these will not be discussed here.

The most basic signal constellation for QAM puts $M = 2^a$ ($a \geq 4$) signal points on a grid with the constellation in the shape of a square or a cross. Signal constellations for M = 4, 16, 32, 64 and 128 are shown in Figure 56. Using the same notation as Ungerboeck we denote by Δ_0, the minimum Euclidean distance between the points in each constellation.

One can choose the scales on the axes in such a manner that the squared Euclidean distance between a signal point and the origin is equal to the energy of that particular signal. The average energy associated with all of the signals can then be computed for any signal constellation. For example, for the case of 16-QAM where the origin is chosen to be in the center of the constellation, the squared Euclidean distances are shown in Figure 57.

* Ungerboeck calls this QASK (Quadrative Amplitude Shift Keying).



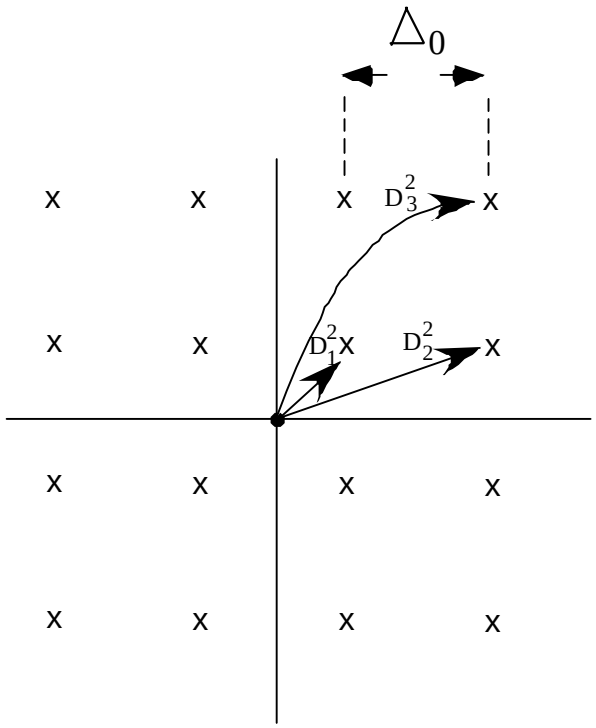
(a) M = 16

(b) M = 32

(c) M = 64

(d) M = 128

Figure 56. Signals Constellations for QAM.



$$D_1^2 = \left(\frac{\Delta_0}{2}\right)^2 + \left(\frac{\Delta_0}{2}\right)^2 = 2\frac{\Delta_0^2}{4}$$

$$D_2^2 = \left(\frac{3\Delta_0}{2}\right)^2 + \left(\frac{\Delta_0}{2}\right)^2 = \frac{10\Delta_0^2}{4}$$

$$D_3^2 = \left(\frac{3\Delta_0}{2}\right)^2 + \left(\frac{3\Delta_0}{2}\right)^2 = \frac{18\Delta_0^2}{4}$$

Figure 57. Squared Euclidean Distances from Origin for 16-QAM.

The average signal energy for the 16 signals, E_{AV}, assuming they are transmitted with equal probability would then be given as:

$$E_{AV} = \frac{1}{16}\left[4D_1^2 + 8D_2^2 + 4D_3^2\right] = \frac{5}{2}\Delta_0^2.$$

Often, Δ_0 is chosen so that $E_{AV} = 1$. If this is done for 16-QAM, then $\Delta_0 = \sqrt{\frac{2}{5}}$. (Note that the value of Δ_0 needed to make $E_{AV} = 1$ depends upon the number of signal points in the constellation.) Ungerboeck, however, fixed Δ_0 as a constant independent of the number of signal points.

Before leaving the example of uncoded QAM, it should be noted that the 4 points at the corners of the constellation for 64-QAM have more energy than the other 60 points. If one moved these points to the axes in such a way to preserve the minimum distance as shown in Figure 58, one could reduce the average energy of the signals. How much this average energy is reduced is left as an exercise. This approach is called signal shaping. Another approach to reduce the average signal energy is to use the high energy signal points less frequently than the low energy signal points. This approach, however, impinges on the information rate while moving the signal points does not.

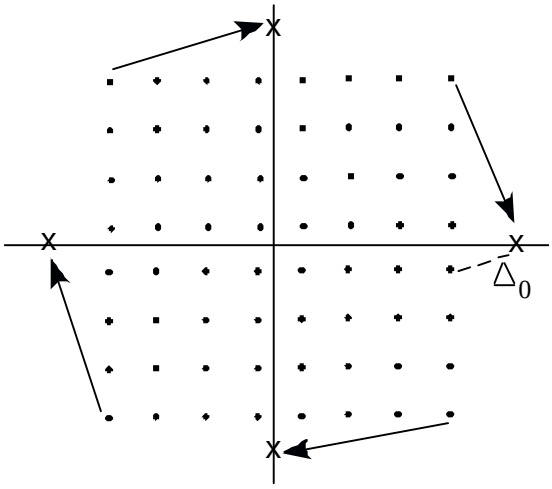


Figure 58. Reducing the Signal Energy by Moving High Energy Points.

The basic pragmatic approach for TCM QAM can be used only for square constellations. The idea is to decompose the constellation into two one-dimensional Pulse Amplitude Modulation (PAM) constellations in the x and y directions and code each of these signals separately. Thus, for example for 64-QAM we would decompose the 64 points into two 8-PAM constellations, one of which is shown in Figure 59.

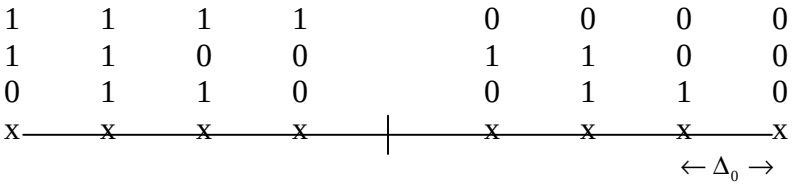


Figure 59. An 8-PAM Constellation.

Also shown in Figure 59 is the assignment of 3 binary digits to each of these points. It should be apparent that a rate 2/3 TCM code designed for 8-PSK can then be used for this 8-PAM constellation. The same scheme could be used for the 8-PAM signal in quadrature. The output of the two encoders would then be used to modulate two sinusoids that are 90° out of phase to form the QAM transmitted signal QAM. This is shown in Figure 60. Note that the 64-QAM signal carries only 4 bits of information so that the rate, R, of the code is $R = \frac{4}{6}$. Thus, one should compare this system with an uncoded 16-QAM system to determine the coding gain.

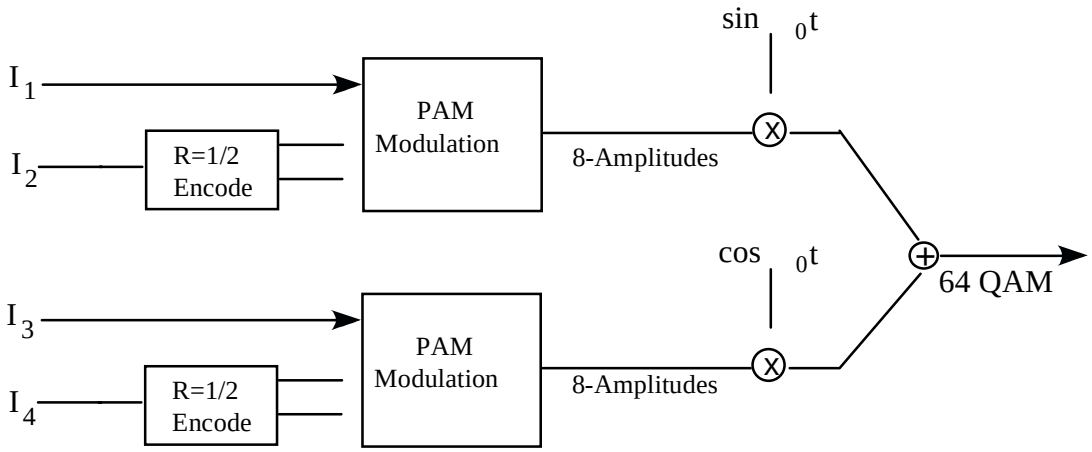


Figure 60. Block Diagram of Encoder for Pragmatic TCM 64-QAM System.

The pragmatic approach for QAM used four times the number of signal points as would be used for the uncoded system. Unbergoeck's set partitioning approach uses only twice the number of signal points and is preferable. Thus, we will omit further discussion of the pragmatic approach and go on to explain how to use set partitioning for QAM.

Set partitioning for QAM constellations works in a manner very similar to PSK. Consider the case of 16-QAM signals placed on a rectangular grid as shown as A0 in Figure 61. The nearest neighbor Euclidean distance, Δ_0, was chosen equal to $\frac{2}{\sqrt{10}}$ so that the average energy (i.e., the average squared Euclidean distance from the origin) for the 16 signal points is equal to 1. Set partitioning leads to two subsets, B0 and B1, of minimum Euclidean distance $\Delta_1 = \sqrt{2}\ \Delta_0 = \frac{2}{\sqrt{5}}$, four subsets , C0, C1, C2 and C3, of minimum Euclidean distance $\Delta_2 = \sqrt{2}\ \Delta_1 = \frac{2\sqrt{2}}{\sqrt{5}}$, eight subsets, D0, D1, …, D7, of minimum Euclidean distance $\Delta_3 = \sqrt{2}\ \Delta_0 = \frac{4}{\sqrt{5}}$ and finally 16 points. The assignment of 4 binary digits to each of the 16 points is in accordance with the values of $y^{(3)}$, $y^{(2)}$, $y^{(1)}$ and $y^{(0)}$.

An 8-state trellis for rate 3/4 TCM code for 16-QAM is shown in Figure 62. There are 8 branches leaving each state, the 8 branches being comprised of 4 pairs of parallel branches. Each pair of parallel branches is one of the 8 subsets D0, D1, …, D7. Thus, the Euclidean distance between parallel branches is $\Delta_3 = \frac{4}{\sqrt{5}} = 1.789$. A pair of paths having the smallest free Euclidean distance, $\sqrt{\Delta_1^2 + \Delta_0^2 + \Delta_1^2} = \sqrt{2} = 1.414$ is also shown in Figure 57.

There is really not a good QAM constellation with 8 points so it is difficult to compare this code with uncoded QAM. Instead we will compare it with 8-PSK since both systems have a spectral efficiency of 3 bits/Hz. Since the minimum squared Euclidean distance of 8-PSK is $(2\sin 22.5°)^2 = .5858$, the ACG of this rate 3/4 TCM coded 16-QAM system over 8-PSK is $10 \log_{10} \frac{2}{.5858} = 5.33$ dB.

Some results of a computer search by Ungerboeck for good rate 3/4 TCM codes for 16-QAM are given in the table below:

# of States	$\tilde{m}$	$H^{(0)}(D)$	$H^{(1)}(D)$	$H^{(2)}(D)$	ACG Over Uncoded 8-PSK
4	1	5_8	2_2	-	4.4 dB
8	2	11	02	04	5.3 dB
16	2	23	04	16	6.1 dB
32	2	41	06	10	6.1 dB
64	2	101	016	064	6.8 dB
128	2	401	056	354	7.4 dB

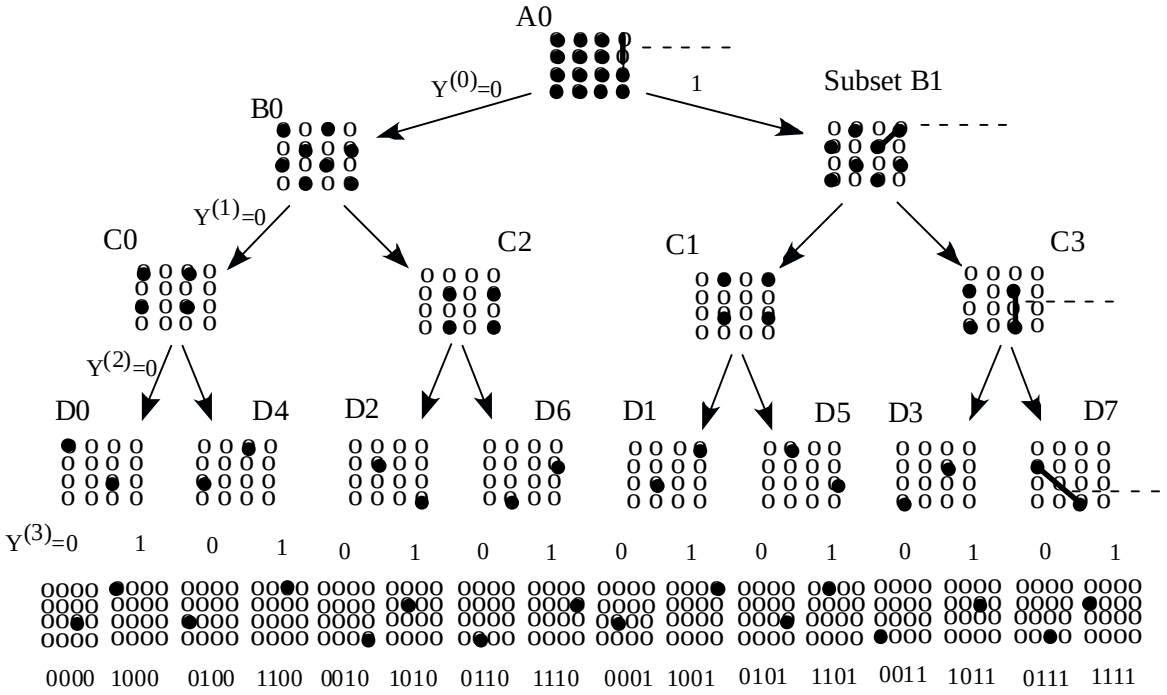


Figure 61. Set Partitioning for 16-QAM.

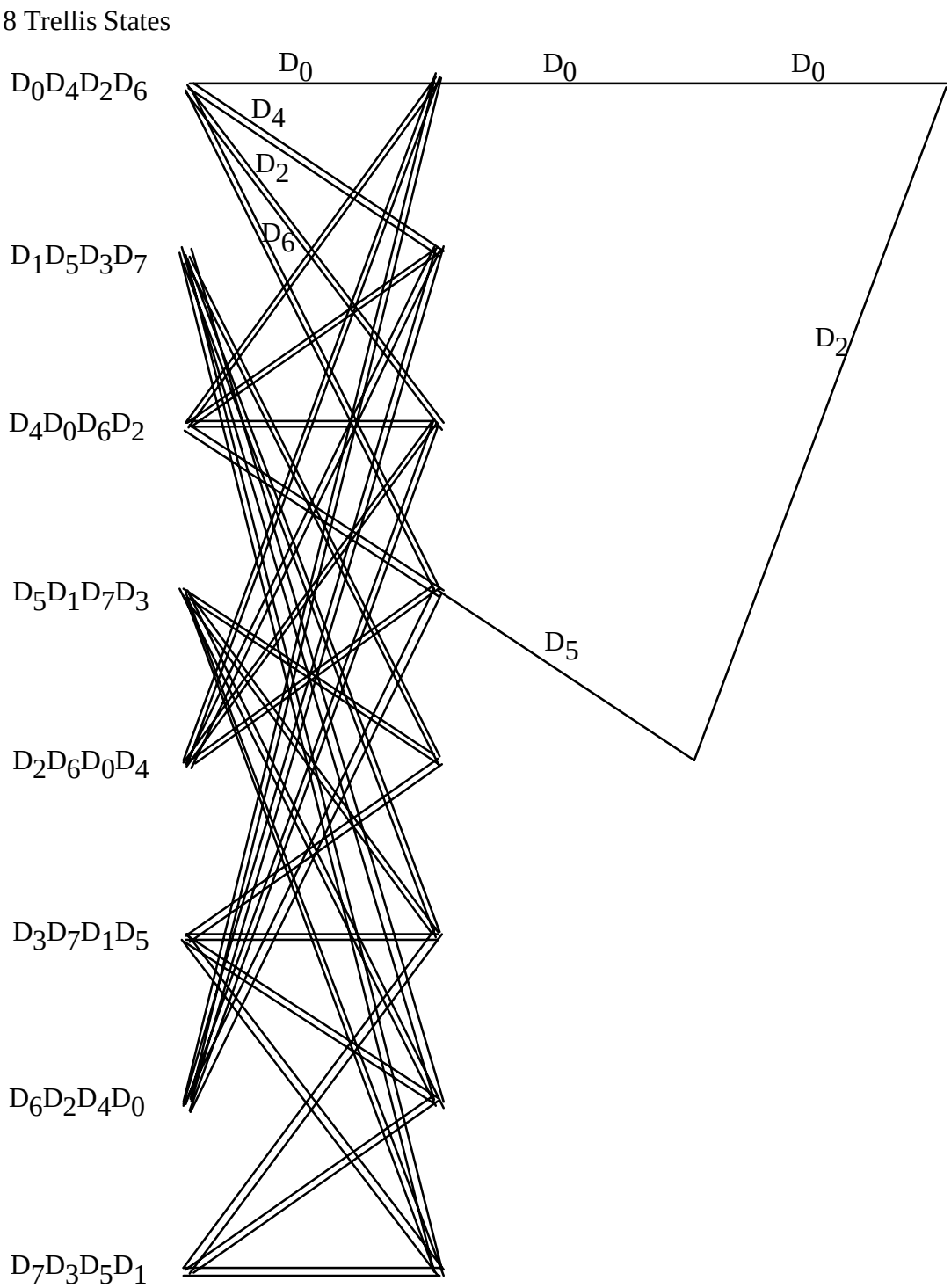


Figure 62. Trellis for Rate 3/4 TCM Code for 16-QAM.

To find the free, squared Euclidean distance for a TCM code is more difficult than finding the free Hamming distance for a convolutional code. That is the case since even though TCM codes use convolutional encoders as part of the TCM encoder, the codes are nonlinear because of the nonlinear mapping in assigning binary sequences to signal points.

Before examining the problem for TCM codes, let us review the situation for convolutional codes. Let $\underline{v}$ and $\underline{v}^1$ be any two code words in a convolutional code that start and end in a common state. (For simplicity we consider an output code word as a single stream of digits that can be represented by a vector.) The free Hamming distance of the code is then given as:

$$d_{free}^{(H)}(\text{code}) = \min_{\substack{\underline{v},\ \underline{v}^1 \\ \underline{v} \neq \underline{v}^1}} d^{(H)}\left(\underline{v}, \underline{v}^1\right)$$

where $d^{(H)}(\underline{v}, \underline{v}^1)$ is the Hamming distance between. But since the code is linear, one of these code words (say $\underline{v}$) can be taken as the all zero code word so that:

$$d_{free}^{(H)}(\text{code}) = \min_{\underline{v} \neq 0} w^{(H)}\left(\underline{v}\right)$$

where $w^{(H)}(\underline{v})$ is the Hamming weight of $\underline{v}$, that is the number of 1's in $\underline{v}$. By writing $\underline{v}^1 = \underline{v} + \underline{e}$, where $\underline{e}$ is a code word, this also could be written as:

$$d_{free}^{(H)}(\text{code}) = \min_{\underline{e} \neq 0} d_{free}^{(H)}\left(\underline{v}, \underline{v} + \underline{e}\right) = \min_{\underline{e} \neq 0} w^{(H)}(\underline{e}).$$

As stated previously, one can assign to each branch in the trellis, the Hamming weight of the code symbols assigned to that branch and then use a Viterbi decoder to find the minimum weight of a branch that diverges from the all-zero state and ends in the all-zero state.

If one did not have a linear code, the simplification of always allowing one of the code words to be the all-zero code word could not be done. In that case, one could build a trellis that simultaneously tracks two code words. One way of doing this would be to use a new trellis where the number of states in the new trellis was equal to the number of states in the old trellis. For example, if the original trellis had four states, a, b, c and d, the new trellis would have 16 states aa, ab, ac, ad, ab, …, dc, dd. Then if we had segments of two code words in the original trellis, one of which took a branch from state a to state b (with label A), and the other took a branch from state d to state c (with label B) these two code words would be represented by a branch on the new trellis that went from branch ad to state bc. The Hamming distance between these two word segments would be the label assigned to that branch. This is shown in Figure 63. The Viterbi algorithm would then be used on the new trellis to find the pair of paths with minimum free Hamming distance.

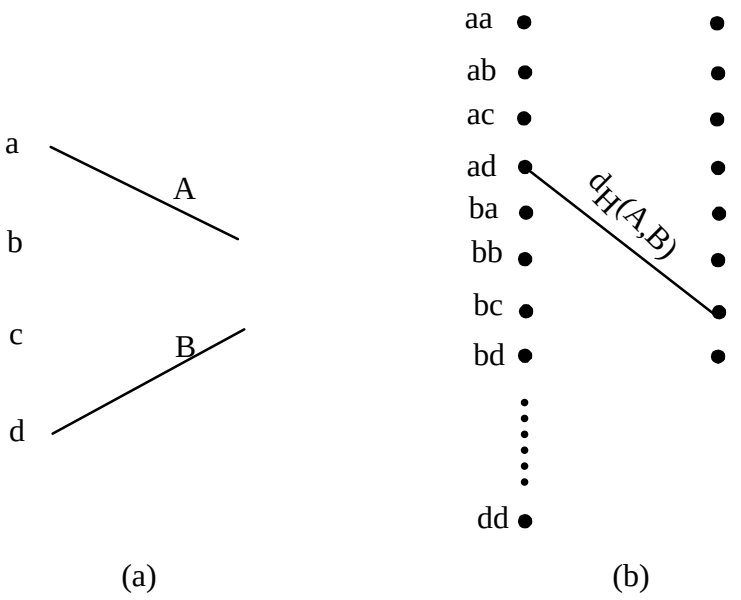


Figure 63. New Trellis for Tracing Two Paths.

Let us now consider the case of TCM codes. The code words of the code are now sequences of symbols from some signal constellation and can be represented by paths in the trellis. Now we are interested in the free squared Euclidean distance between code words in the code.

Referring to Figure 64, an encoder for a TCM code employs a rate k_0/n_0 binary convolutional encoder with 2^s states and a non-linear mapper that maps the n_0 encoded binary output digits to a point (or a symbol) in a 2^{n_0} signal constellation.

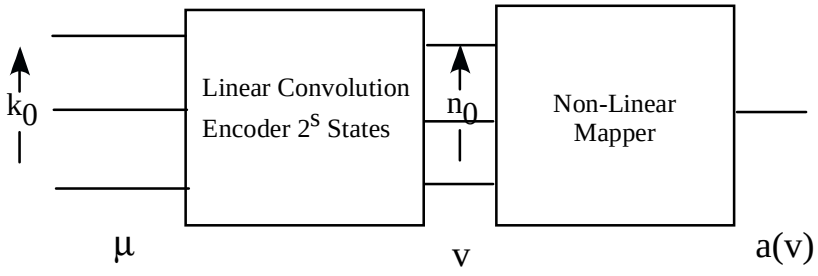


Figure 64. General Encoder for TCM.

It should be noted that we now refer to the input k_0-tuple to the convolutional encoder by the letter μ (without an underbar) and refer to the output of the convolutional encoder by the letter v. We reserve the use of vectors to refer to sequences. In particular we now denote by $\underline{v}$, the sequence $v_1, v_2, \ldots$ where v_i is the n_0-tuple output by the convolutional encoder at time i. Finally

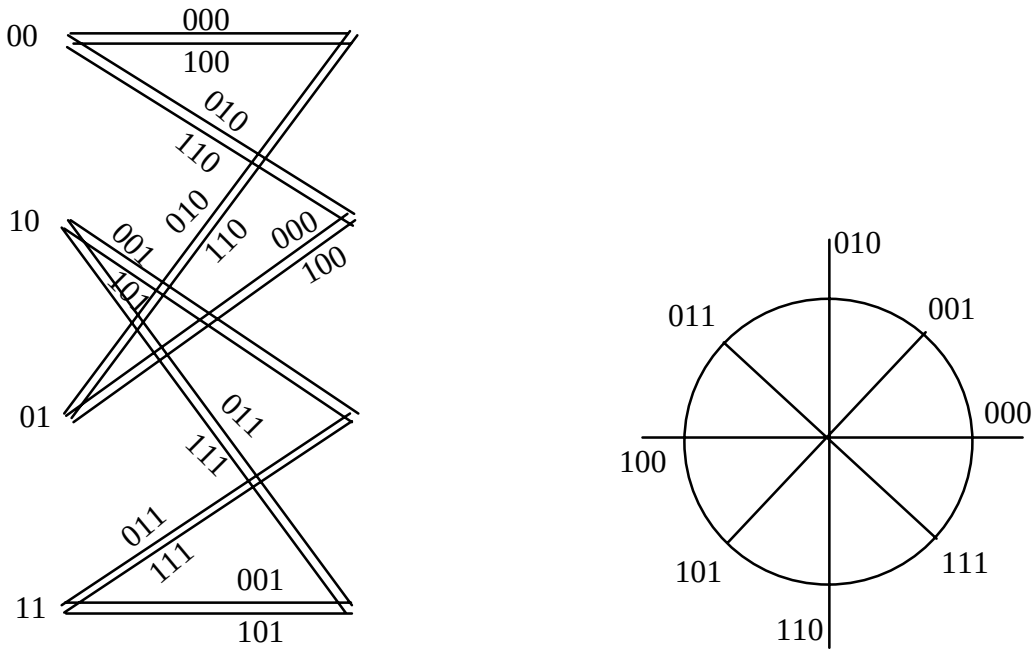
we denote by $a(v_i)$ the symbol in the signal constellation that v_i maps to and to $a(\underline{v})$, the sequence $a(v_1)$, $a(v_2)$, … . Since the code is linear, of v_1 and v_2 are two code n_0-tuples that can be produced by the encoder v_2 can be written as $v_2 = v_1 + e$ where e is an n_0-tuple that can be produced by the encoder.

Let us now associate with each non-zero n_0-tuple, e, that can be produced by the encoder, the quantity $w^2(e)$ defined as:

$$w^2(e) = \min_{\substack{y \\ y \neq 0}} \left[d^{(E)}\left(a(y),\ a(y+e)\right)\right]^2$$

where the minimum is taken over all non-zero n_0-tuples that can be produced by the encoder. Then let us use the ordinary trellis for tracking a single code sequence. Each branch in this trellis corresponds to the n_0-tuple which is output from the convolutional encoder. For every $e \neq 0$, let us place the label $w^2(e)$ on every branch corresponding to e. Let us label branches corresponding to e = 0 by the label 0. Then a Viterbi decoder can find the path or paths that diverge from the all-zero state at time t = 0 and eventually remerge to the all-zero state. The result will be a lower bound on the squared free Euclidean distance of the TCM code.

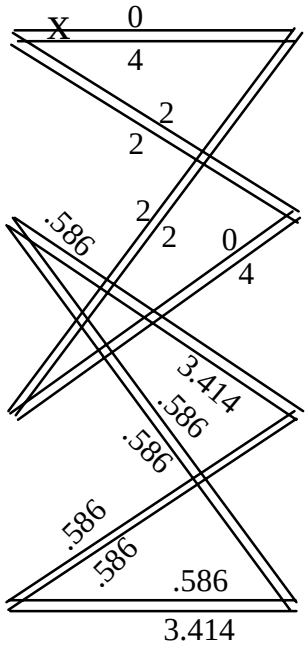
Let us show how this is done for the 4-state TCM code for 8-PSK modulation using the convolutional encoder shown in Figure 2. The trellis for this convolutional code labeled with the output 3-tuples are shown in Figure 65(a).



(a) Encoder Trellis (b) Nonlinear Mapping

Figure 65. 4-State TCM Code for 8-PSK

The trellis with labels $w^2(e)$ is shown in Figure 66.



$w^2(000) = 0$

$w^2(100) = 4$

$w^2(010) = 2$

$w^2(110) = 2$

$w^2(001) = .586$

$w^2(101) = 3.414$

$w^2(011) = .586$

$w^2(111) = .586$

Figure 66. Trellis for Calculating Lower Bound to Squared Free Euclidean Distance.

If one deletes the path that goes from the all-zero state to the all-zero state with $w^2(e) = 0$, we see that the lower bound to the squared Euclidean distance is equal to 4 as shown by the parallel path on top.

Although it would appear that this procedure only gives a lower bound to the squared free Euclidean distance, Ungerboeck shows that for his construction of TCM codes the lower bound is achieved with equality. For details, see Ungerboeck. Thus we can compute an exact expression for $\left(d_{\text{free}}^{(E)}\right)^2$ for any TCM code using the Ungerboeck construction by using the above procedure. However, we would have to make a different calculation for every different signal constellation.

A lower bound that may not be achievable was described by Ungerboeck that holds for any signal constellation. We note that if e has b=b(e) > 0 trailing 0's (i.e., $e = e^{(n-1)}e^{(n-2)} \ldots e^{(1)}e^{(0)}$ where $e^{(0)} = e^{(1)} = \ldots = e^{(b-1)} = 0$ and $e^{(b)} \neq 0$) then b is the last level in the set partitioning where y and y + e lie in the same subset. Thus,

$$w^2(e) \geq \Delta_b^2 \ .$$

For example, in the example described in Figures 65 and 66,

$$w^2(e^{(2)}e^{(1)}1) \geq .586 = \Delta_0^2$$

and

$$w^2(e^{(2)}\ 1\ \ 0) \geq 2 = \Delta_1^2$$
$$w^2(1\ \ 0\ \ 0) \geq 4 = \Delta_2^2 \quad .$$

Note that although sometimes the inequality holds with equality (for example, $w^2(1\ 0\ 0) = 4$ and $w^2(0\ 1\ 0) = w^2(1\ 1\ 0) = 2$) there are cases where the inequality is strict (for example, $w^2(1\ 0\ 1) = 3.414 > .586$).

Ungerboeck's search procedure used this bound. If we let $\underline{e} = e_1\ e_2\ e_3\ \ldots$be a sequence of e's corresponding to a path, and if we define

$$\Delta^2(\underline{e}) = \sum_{i=1} \Delta^2_{b(e_1)}$$

then

$$\left(d^{(E)}_{free}\right)^2 \geq \min_{\substack{\underline{e} \\ \underline{e} \neq \underline{0}}} \Delta^2\ (e) \square\ \Delta^2_{free} \quad .$$

By searching for the best codes while maximizing Δ^2_{free}, Ungerboeck had a procedure that depended only on the Δ_i^2 and not on the specific constellation. That allowed a search for TCM codes for QAM constellations that was independent on the number of points. It is interesting to note that Ungerboeck states that he never found a case where $\left(d^{(E)}_{free}\right)^2 > \Delta^2_{free}$.

Note that in the partitioning for the 16-QAM constellation shown in Figure 61, the minimum Euclidean distance at step i in the partitioning, Δ_i, is equal to the minimum Euclidean distance at the previous step, Δ_{i-1}, multiplied by $\sqrt{2}$. That is,

$$\Delta_i = \sqrt{2}\ \Delta_{i-1} \quad \text{for}\ \ i=1, 2, \ldots \quad .$$

The same will be true for any QAM constellation based upon the square lattice. This equation was the essential ingredient in the search for the best codes for 16-QAM. Thus, the same codes that were found for 16-QAM also are the best codes for 32-QAM, 64-QAM, 12-QAM, etc. This is an extraordinary result. Specifically, this result states that the search for good rate R=m/(m+1) codes for any QAM constellation where the points lie on a square lattice is over – Ungerboeck found them. Since there are either square or cross constellations for 16- , 32- and 64-QAM one

can give the ACG for Ungerboeck's codes comparing these constellations. The results are given in the table below for the same codes that were previously described in the table on page 63[*].

# of States	$\tilde{m}$	$H^{(0)}(D)$	$H^{(1)}(D)$	$H^{(2)}(D)$	ACG 32-QAM / 16-QAM	64-QAM/ 32-QAM
4	1	5_8	2_2	-	2.8 dB	3.0 dB
8	2	11	02	04	3.8 dB	4.0 dB
16	2	23	04	16	4.6 dB	4.8 dB
32	2	41	06	10	4.6 dB	4.8 dB
64	2	101	106	064	5.2 dB	5.4 dB
128	2	401	056	354	5.8 dB	6.0 dB

It should be remembered that of the total of m bits that enter the encoder, $(m-\tilde{m})$ of them are uncoded and are passed directly to the non-linear mapper. This results in $2^{(m-\tilde{m})}$ parallel branches in the trellis between each pair of states that have any branch connecting them at all. The truly surprising result is that, for the best codes, $\tilde{m}$ was found to be either equal to 1 or 2. Not only does this result simplify the search but it also greatly simplifies both the encoder and decoder implementation.

A little thought will see why a very small value of $\tilde{m}$ is best. The squared free Euclidean distance of the code is equal to the minimum of two quantities: (1) the minimum Euclidean distance of the parallel transitions, $\Delta^2_{\tilde{m}+1}$, and (2) the minimum squared Euclidean distance of two paths in the trellis that diverge in some state and remerge later in some state. The larger the value of $\tilde{m}$, the larger the value of $\Delta^2_{\tilde{m}+1}$. However, the larger the value of $\tilde{m}$, the more states that are reached by branches from any state. (This number is $2^{\tilde{m}}$.) But the more states that are reached from any state, the shorter the length before two paths remerge. Short paths in general result in a small value for the minimum squared Euclidean distance between non-parallel paths. For example, if the number of states in the trellis were equal to $2^{\tilde{m}}$, we would have a fully connected trellis and the minimum length non-parallel path would be equal to 2. Such a short path would very likely have small squared Euclidean distance. Thus, $\tilde{m}$ must be chosen as to balance the minimum squared Euclidean distance of parallel and non-parallel paths. The result found by Ungerboeck was that $\tilde{m}$ was equal to 1 or 2 for the best codes.

One might also wonder if better TCM codes could be found at lower rates than R=m/(m+1). One disadvantage of a lower rate code would be the excessive number of signal points required. A rate R=m/(m+1) already requires twice the number of signal points as in the

[*] The information for this table was taken from Table III of Ungerboeck, "Channel Coding with Multi-Level/Phase Signals.", *IEEE Transactions on Magnetics*, pp. 55-67. Vol. IT-28, January 1982. The last two columns were reversed in a later paper by Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets. Part II, State of the Art." *IEEE Communications Magazine*, Vol. 25, pp. 12-20, February, 1987. I believe the above table is correct.

uncoded case. A rate R=m/(m+2) code would require four times the number of signal points. Ungerboeck found that allowing for lower rate codes did not lead to higher coding gain. Thus, the subject of lower rate TCM codes will not be discussed further here.

Many important points regarding TCM codes were not discussed to this point. One of these is phase invariance. In M-PSK, the phase lock loop could lock onto any one of the M phases. In QAM, phase ambiguity of 90°, 180°, or 270° could occur. The CCITT V:32 standard uses a non-linear 8-state encoder with a 32-QAM (cross) constellation and differential encoding to cope with this phase ambiguity. Details are given in the references (G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets. Part II, State of the Art." *IEEE Communications Magazine*, Vol. 25, pp. 12-20, February, 1987).

Another aspect of TCM codes that was also introduced by Ungerboeck is the notion of multi-dimensional trellis codes. The easiest way of thinking of 2K-dimensional (2K-D) codes is to think of the code symbols as a K-tuple of signals from a 2-D constellation. We will illustrate the idea for K=2 using a 16-QAM constellation. A pair of uncoded 16-QAM signals transmitted serially can be thought of as one of 256 points in 4-D space. Since 256 signal points can carry 8 bits of information, Ungerboeck considered rate R=7/8 TCM codes. That is, the 7 information bits are first converted to 8 coded bits by an encoder. In analogy to what was done before $(7-\tilde{m})$ of the 7 information bits will be uncoded resulting in a trellis with $2^{(7-\tilde{m})}$ transitions. The $\tilde{m}$ coded information bits are passed through a rate $R=\tilde{m}/(\tilde{m}+1)$ convolutional code resulting in $(\tilde{m}+1) = 8$ coded bits. The 8 coded bits are then mapped to a pair of 16-QAM signals. In the case of 4-D codes, the optimal value of $\tilde{m}$ is larger than that found for 2-D codes. Ungerboeck's best 4-D codes used values of $\tilde{m}$ of 3 or 4.

Set partitioning for this pair of 16-QAM signals is slightly more complicated than in the 2-D case. The following description pertains to the set partitioning shown in Figure 67. We also refer to the 16-QAM constellation shown in Figure 61. At the highest level of the tree in Figure 67 is the set A0 x A0 where A0 refers to the set of 16 points shown at the highest level of the tree in Figure 61. This corresponds to choosing one of 16 points from the set A0 and another one of 16 points from a second set A0. The next level of the tree contains the sets $B_4^0 = (B0\times B0)\cup(B1\times B1)$ and $B_4^1 = (B0\times B1)\cup(B1\times B0)$. The set B_4^0 then splits into two subsets: subset B0×B0 and subset B1×B1. Each of these subsets contain 64 choices. The minimum distance between points in each subset are shown in the right of Figure 67. Figure 67 does not show all of the partitioning since it would take 8 levels of splitting to get to the subsets with single points. These codes are better than 2-D codes in that, for a given size constellation, say 16-QAM, they have a higher spectral efficiency than 2-D codes. They also have better signal-to-noise performance.

TCM has been applied to channels other than the AWGN channel. One important class of channel models is that of fading channels. There are many such models, one of which is slow fading where the S/N varies slowly over time. This variation in S/N ratio can be thought of as if the signal points in the constellation had a scale that contracted or expanded with time. For example, with PSK modulation, one could consider that the radius of the circle was described by a random variable G (G for gain) which varied slowly with time. The result would be that if the gain G remained constant over the amount of time corresponding to the path length of the free

Euclidean distance error event, then this free Euclidean distance would be multiplied by a factor G (which is a random variable). It would be much better if one could average over the fading values by multiplying the free Euclidean distance by the average of G. Bit interleaving is one way of accomplishing this averaging. (See: E. Zehavi and J. K. Wolf, Patent 5,633,881, "Trellis Encoder and Decoder Based Upon Punctured Rate 1/2 Convolutional Codes", May 27, 1997.) In this approach, a rate 1/2 convolutional code is punctured to obtain a high rate code, the output of the punctured encoder is put through an interleaver and then to a signal mapper. The resulting trellis does not have parallel branches. Parallel branches are bad in TCM codes for fading channels if the mapping is such that the parallel branches can fade with little or no diversity protection. An encoder for TCM coding of 8-PSK in slow fading is shown in Figure 68.

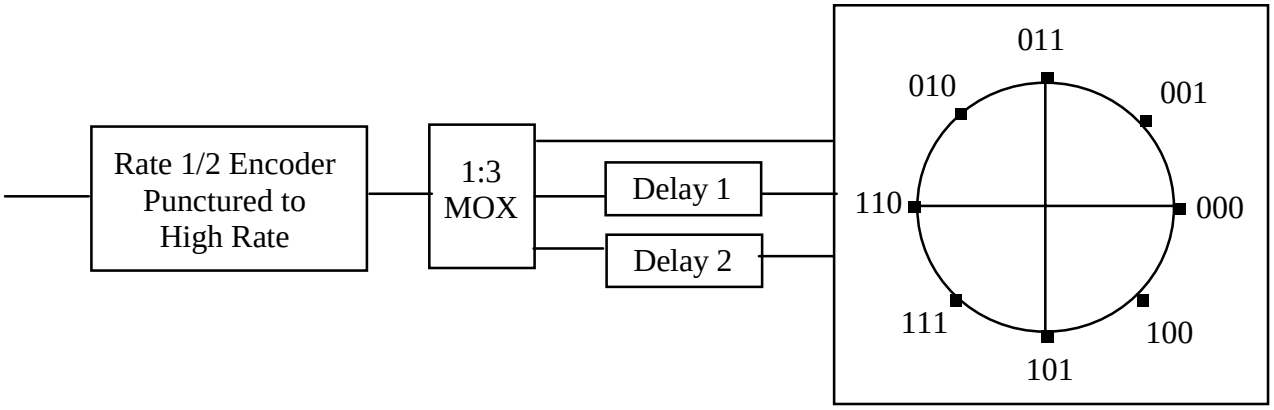


Figure 68. Encoder for 8-PSK in Fading.

Note the special assignment for 3-tuples to the 8-PSK modulation. This is done to give good soft decisions for the 3 bits.