

CHAPTER 5

DESIGN FOR AWGN CHANNELS

This chapter is concerned with the design of trellis codes for AWGN channels, with specific reference to the application of classical and TCM forward error correction techniques to Q^2 PSK. The first section of this chapter considers the classical techniques. The final sections deal with the design and application of TCM techniques for Q^2 PSK.

5.1 INTRODUCTION TO ERROR CORRECTION

The task facing the communications engineer is that of providing a cost-effective system for transmitting information (data or voice) from a sender to a user at a specific rate and an acceptable level of reliability. Practical considerations usually place a limit on the value that can be assigned to E_b/N_o . In other words, in practice, we may find that a specific modulation scheme may not provide acceptable performance. For a fixed E_b/N_o , the only practical option available for improving data quality to an acceptable level is to resort to coding techniques. The use of Forward Error Correcting (FEC) codes is well suited to channels that can be modeled as Additive White Gaussian Noise (AWGN) channels [22, 75, 76].

In a general FEC scheme the *channel encoder* accepts message bits and adds controlled *redundancy* according to a prescribed encoding rule, thereby producing encoded data at a higher bit rate. The *channel decoder* exploits the redundancy to decide which message bit was actually transmitted. In FEC systems, the motivation is to achieve a *coding gain*, defined as the difference in E_b/N_o required for coded and uncoded systems to provide a specified BER when operating on an AWGN channel. Moreover, the use of coding adds complexity to the system, especially for the implementation of the decoding operation at the receiver. Thus, the design trade-offs in the use of error-control coding are considerations of bandwidth and system complexity.

5.2 CLASSICAL ERROR CORRECTION

In the following sections low-complexity sub-optimal convolutional encoding schemes are proposed for Q²PSK. The first scheme employs a single rate-3/4 convolutional encoder, yielding an effective data rate equal to that of 3/4, the rate of uncoded Q²PSK, operating at R_b . A Maximum-Likelihood (ML) soft-decision Viterbi decoder is utilised at the receiver. The second encoding scheme, with so called *dual-use* of conventional 2D signals, employs two rate-1/2 convolutional encoders. At the receiver two soft-decision Viterbi decoders are utilised in parallel. The resulting bit rate is reduced by a factor of two, compared to that achieved with the uncoded transmission, to yield an effective bit rate equal to that of conventional 2D-QPSK.

Since the envelope of the Q²PSK signal in the absence of any additional constraint is non-constant, a third encoding scheme is proposed, employing a hybrid convolutional-block coding scheme. The encoder consists of a rate-2/3 convolutional encoder, followed by a simple block encoding scheme to facilitate a constant envelope. At the receiver a single soft-decision Viterbi decoder, together with a block decoding scheme, is employed.

5.2.1 Convolutional Channel Coding

Convolutional codes differ from block codes in that the encoder contains m memory elements, and the n encoder outputs at any given time unit depend not only on the k input bits at that time, but also on m previous input bits. A convolutional code is generated by passing the information sequence to be transmitted through a linear finite-state shift register. In general, the shift register consists of L_c (k -bit) stages and n linear algebraic function generators, called *generator sequences*. The parameter L_c is called the *constraint length* of the convolutional code, and can be interpreted as the maximum number of encoder outputs that can be affected by a single information bit. The number of states N_s , of the convolutional encoder which generates n encoded bits is a function of L_c and k input bits, given by $N_s = 2^{k(L_c-1)}$.

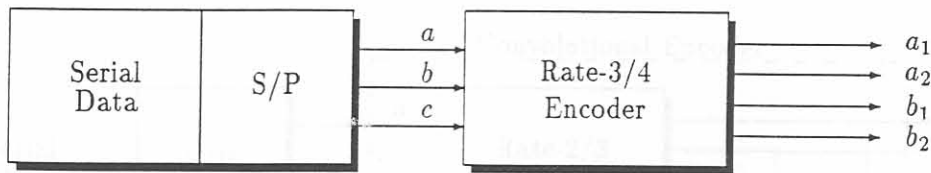
An (n, k, L_c) convolutional code can be implemented with a k -input, n -output linear shift register with input memory, $m = kL_c$, and at most knL_c modulo-2 adders. For each of the n outputs, a generator sequence of length kL_c is needed, describing the connections of the kL_c shift register stages to the modulo-2 adder of that output stage.

For our application, rate- $\frac{k}{k+1}$ trellis encoders, introduced by Ungerboeck [28, 29], are employed to select 2^n ($n = k + 1$) equally likely channel symbols.

5.2.1.1 Rate-3/4 encoder for Q²PSK

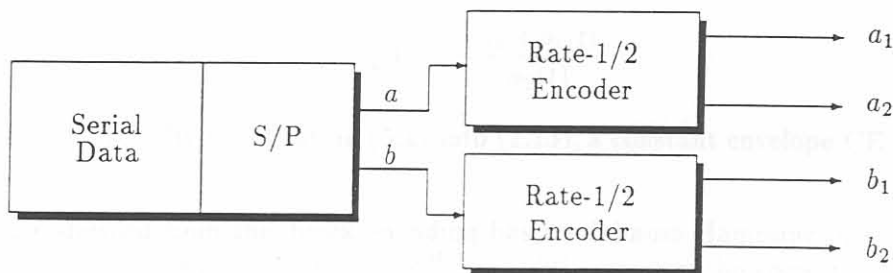
The first encoding strategy proposed for Q²PSK consists of a single rate-3/4 convolutional encoder, as illustrated in Figure 5.1. The serial data bit stream is Serial-to-Parallel (S/P) converted, producing three parallel bit streams, a , b and c . These uncoded bit streams are then fed to the rate-3/4 encoder, producing a coded signal set A consisting of four coded bit streams $\{a_1, a_2, b_1, b_2\}$. Signal set A therefore consists of 2^4 equally likely signal symbols.

Since the ratio between the number of input bits to the number of output bits is 3/4, the effective code rate is reduced by 1/4 compared to that of uncoded transmission.

Figure 5.1: Rate-3/4 encoding scheme for Q²PSK.

5.2.1.2 Rate-1/2 Encoder for Q²PSK

In the coding scheme proposed for Q²PSK, two rate-1/2 ($k = 1$) convolutional encoders, i.e., *dual-use* of 2D encoders, are employed as depicted in Figure 5.2. The serial data bit stream is S/P converted, producing two parallel bit streams, a and b at a rate $R'_b = R_b/2$. Recall that R_b is the original data rate. The uncoded bit streams, a and b , are then fed to the two rate-1/2 convolutional encoders, each producing two sets, A and B , consisting of two encoded data streams $\{a_1, a_2\}$ and $\{b_1, b_2\}$, respectively. The two sets of encoded data streams each produce a signal set consisting of $2^2 = 4$ symbols, resulting in a total signal set of $M = 16$ symbols given by the Cartesian products of the sets A and B .

Figure 5.2: Half rate encoding scheme for Q²PSK.

Since the ratio between the number of input bits to the number of output bits is one half, this is a true *half rate* convolutional encoding scheme.

5.2.1.3 Encoder for CE-Q²PSK

The coding scheme proposed for constant envelope CE-Q²PSK employs a hybrid convolutional-block coding scheme. The block encoding scheme is incorporated to force a constant envelope signal.

The front-end of this encoder is formed by a single rate-2/3 ($k = 2$) convolutional encoder as depicted in Figure 5.3. As for the Q²PSK encoder, the serial data bit stream is S/P converted,

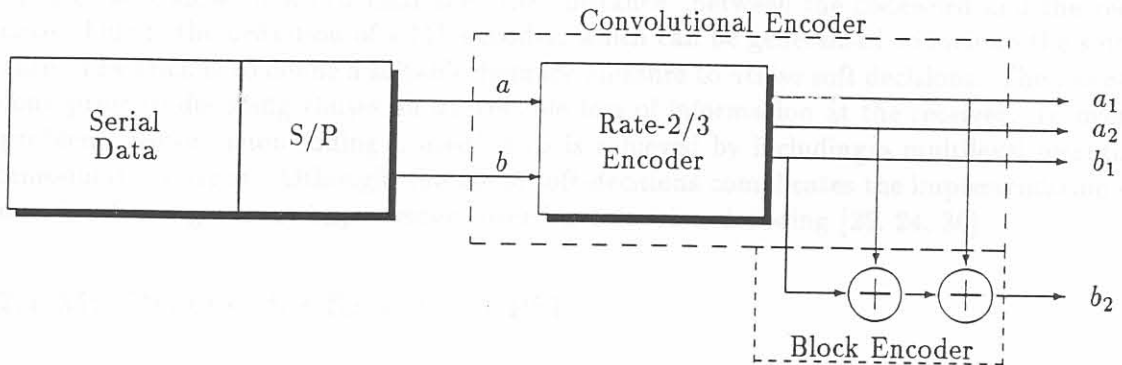


Figure 5.3: Half rate encoding scheme for CE-Q²PSK.

producing two parallel bit streams, a and b . These uncoded bit streams are then fed to the rate-2/3 convolutional encoder, which produces a single symbol set, C , consisting of the three encoded data streams, a_1, a_2 and b_1 , respectively. Note, that this set consists of $2^3 = 8$ equally likely symbols.

The block coding scheme used to achieve a constant envelope Q²PSK signal can be described as follows: the coder accepts serial input data and for every three information bits $\{a_1, a_2, b_1\}$, it generates a codeword $\{a_1, a_2, b_1, b_2\}$ such that the first three bits in the codeword are the encoded information and the fourth one is an odd parity check bit. The parity check bit $b_2(t)$ is given by

$$b_2(t) = -\frac{a_1(t)b_1(t)}{a_2(t)} \quad (5.1)$$

in agreeing with (2.15). By substituting (5.1) into (2.13), a constant envelope CE-Q²PSK signal is obtained.

The resulting codeword from this block encoding has a minimum Hamming distance of $d_{min}^H = 2$. Recall from block coding theory that $t = \lfloor (d_{min}^H - 1)/2 \rfloor$ errors can be corrected, where $\lfloor x \rfloor$ is the largest integer less than or equal to x , implying that the added extra data bit, $b_2(t)$, can not be used for error correction. Instead, the redundancy in the resulting signal set can be used to improve the BER performance of the modulation scheme.

5.2.2 Maximum Likelihood Decoding

In Appendix A (section A.4) it is stated that Maximum Likelihood (ML) decoding implies finding the path with largest metric through the trellis by comparing the metrics of all branch paths entering each state with the corresponding received sequence in an iterative manner. The general Viterbi algorithm can be described as follows: In the decoding process, if at some level it is found that a path cannot possibly yield the largest metric, then the path is discarded by the decoder [77, 24]. In this manner, a decoder that compares the metrics of all paths entering a state and keeps only the survivor at that state, will yield a most likely path if the operation is repeated for all distinct states at each level.

In a coded system one would normally wish to structure the decoder such that for a given code the probability of error is minimised. In the hard decision case the optimum decoding procedure is to pick the codeword which differs from the received sequence in the smallest number of positions, i.e., choose the codeword which minimises the "distance" between the codeword and the received sequence. This is the definition of a ML decoder, which can be generalised to include the soft decision case. The trick is to define a suitable distance measure to utilise soft decisions. The use of hard decisions prior to decoding causes an irreversible loss of information at the receiver. To overcome this problem, *soft-decision* coding is used. This is achieved by including a multilevel quantiser at the demodulator output. Although, the use of soft-decisions complicates the implementation of the decoder, it offers significant improvement over hard-decision decoding [25, 24, 30].

5.2.2.1 ML Decoder for Rate-3/4 Q^2 PSK

Figure 5.4 depicts the decoding procedure for rate-3/4 Q^2 PSK, consisting of a Q -bit multi-level soft-quantiser and a Viterbi decoder. The Q -bit quantiser accepts serial input data and produces serial output data quantised into 2^Q levels. These quantised outputs are then fed into the Viterbi decoder, where the data is decoded and the output data stream formed.

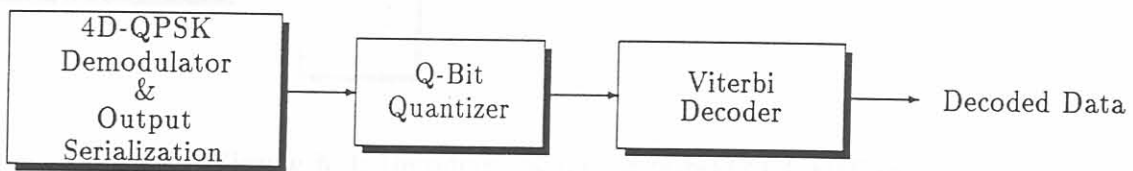


Figure 5.4: Decoding scheme for rate-3/4 Q^2 PSK.

5.2.2.2 ML Decoder for Rate-1/2 Q^2 PSK

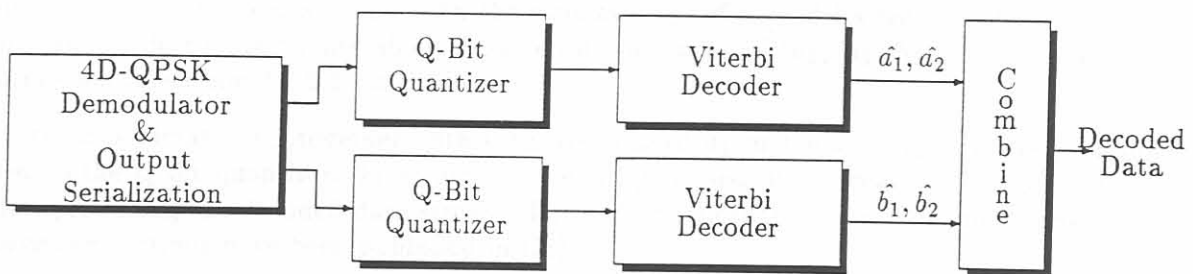


Figure 5.5: Decoding scheme for rate-1/2 Q^2 PSK.

Figure 5.5 depicts the decoding procedure for Q^2 PSK, consisting of two Q -bit multi-level soft-quantisers and two ML decoders, employing the Viterbi algorithm. Each of the two Q -bit quantisers accepts serial input data and produces serial output data quantised into 2^Q levels. These quantised outputs are then fed into the Viterbi decoders, where the data is decoded. From the decoders the output data streams are combined to form the decoded output data stream.

5.2.2.3 ML Decoder for Constant Envelope CE-Q²PSK

Figure 5.6 depicts the decoding scheme for CE-Q²PSK, consisting firstly of a block decoding procedure, and secondly, of a ML Viterbi decoder.

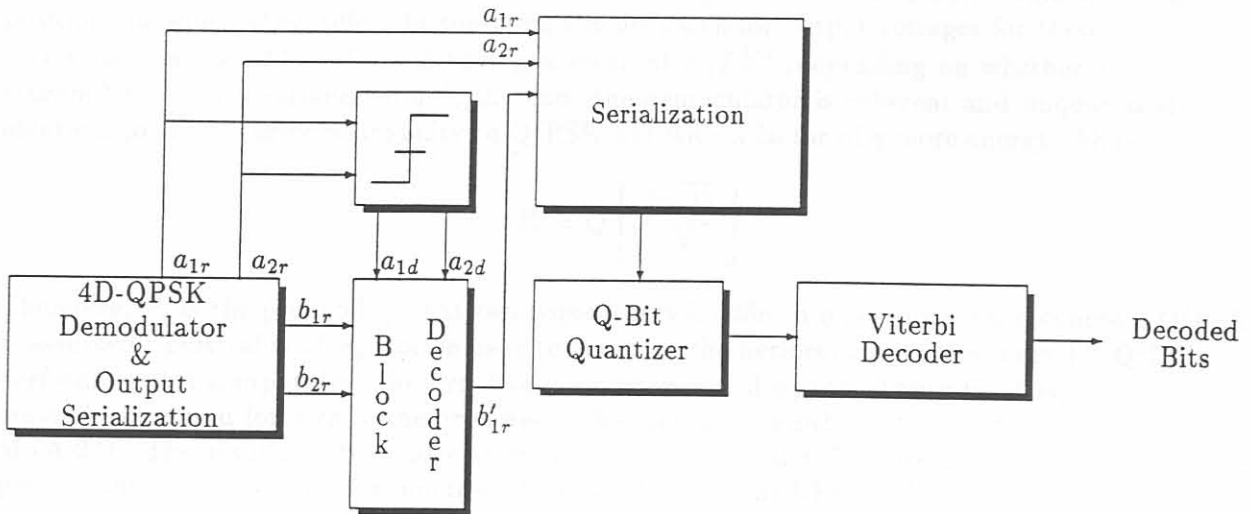


Figure 5.6: Decoding scheme for rate-1/2 CE-Q²PSK.

Decisions about demodulator outputs a_{1r} and a_{2r} associated with pulse trains $a_1(t)$ and $a_2(t)$ respectively, are made from the other Q²PSK demodulator outputs to produce the decoded information streams, a_{1d} and a_{2d} . These two streams, together with the redundant information associated with $b_2(t)$, are used only in making the decision about the information bit in $b_1(t)$. To make a decision about $b_1(t)$, a simplifying assumption that a_{1d} and a_{2d} are decoded correctly has to be made. These decoded information streams, along with the estimates b_{1r} of b_1 and its redundant version b_{2r} of b_2 ; observing a decision estimate about b_1 , given by: $b'_{1r} = b_{1r} - (a_{1d}/a_{2d})b_{2r}$ (in agreement with the discussion in section 4.1.2.1), is made.

The decision estimate, b'_{1r} together with estimates a_{1r} of a_1 and a_{2r} of a_2 , are then serialised and fed to the Q-bit quantiser. From here the serial quantised data stream is fed to the Viterbi decoder, producing the decoded data stream. The code classical trellis code designs carried out in the foregoing sections have been published in [78].

5.2.3 Performance Estimates

The most useful techniques of estimating the performance of convolutional coding are the union bound technique and computer simulation. This section will try to quantify the expected performance of the Forward Error Correction (FEC) coders presented in the foregoing sections. Most of the performance evaluation will, however be based on extensive computer simulations. The usefulness of computer simulation is limited by the long execution times required to obtain a good statistical sample (it may take several hours to yield a single point on the error graph). The union

bound approach for convolutional codes is virtually identical to that of block codes, and it provides performance estimates accurate within a small fraction of a decibel for all SNR large enough to give an error rate of 10^{-3} or less.

Assume that coherent Q²PSK is used with a hard quantised demodulator output. Then to apply the union bound (presented in Appendix A, section A.3.3) we must calculate the probability that the distance between the received sequence and a weight- q codeword is less than the distance to the transmitted all-zero word. Call this probability, P_q . P_q is a function of E_s/N_o and the number of positions in which they differ. In summing the demodulator output voltages for these q positions, a new random variable is formed having a mean of $\pm qE_s^{1/2}$ (depending on whether a 1 or 0 was transmitted) and a variance of $qN_o/2$. Since the demodulator is coherent and unquantised, P_b is identical to the bit error probability of Q²PSK but with a factor of q more energy. Thus,

$$P_q = Q \left[\sqrt{\frac{2qE_b}{N_o}} \right] \quad (5.2)$$

Therefore, P_q is the probability that two sequences will differ in q positions when coherent Q²PSK is assumed. This value of P_q can be used to compute the performance of the codes for Q²PSK. In performing this computation the first five nonzero values of w_q (see Appendix A equation (A.24)) must be computed for each of the proposed codes and then be substituted in the truncated version of (A.24). The results of this calculation are shown in Figures 5.7 through 5.9, where the BER performance P_b is plotted as a function of E_b/N_o (in dB). The BER performance curves are for the code rates of 1/2, 2/3 and 3/4, respectively and include the constraint lengths L_c . In Chapter 7 these performance curves will be benchmarked against graphs obtained by means of simulation.

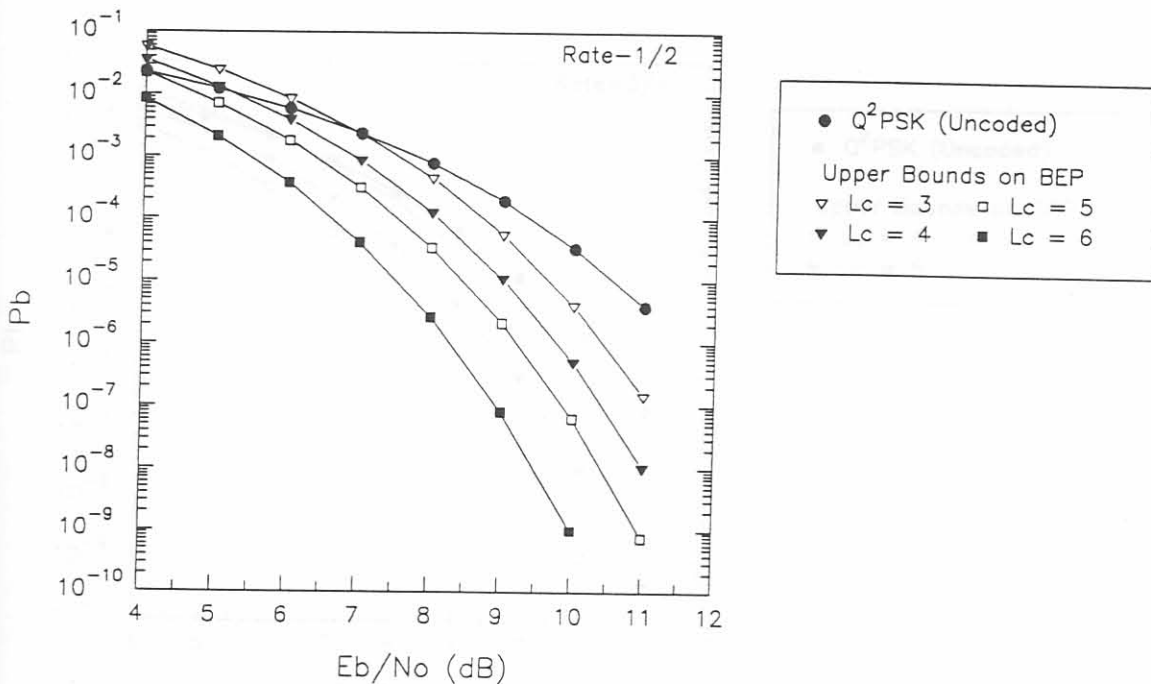


Figure 5.7: Bit error probability, P_b for Rate-1/2 codes with Viterbi decoding (Hard quantisation) and Q²PSK modulation.

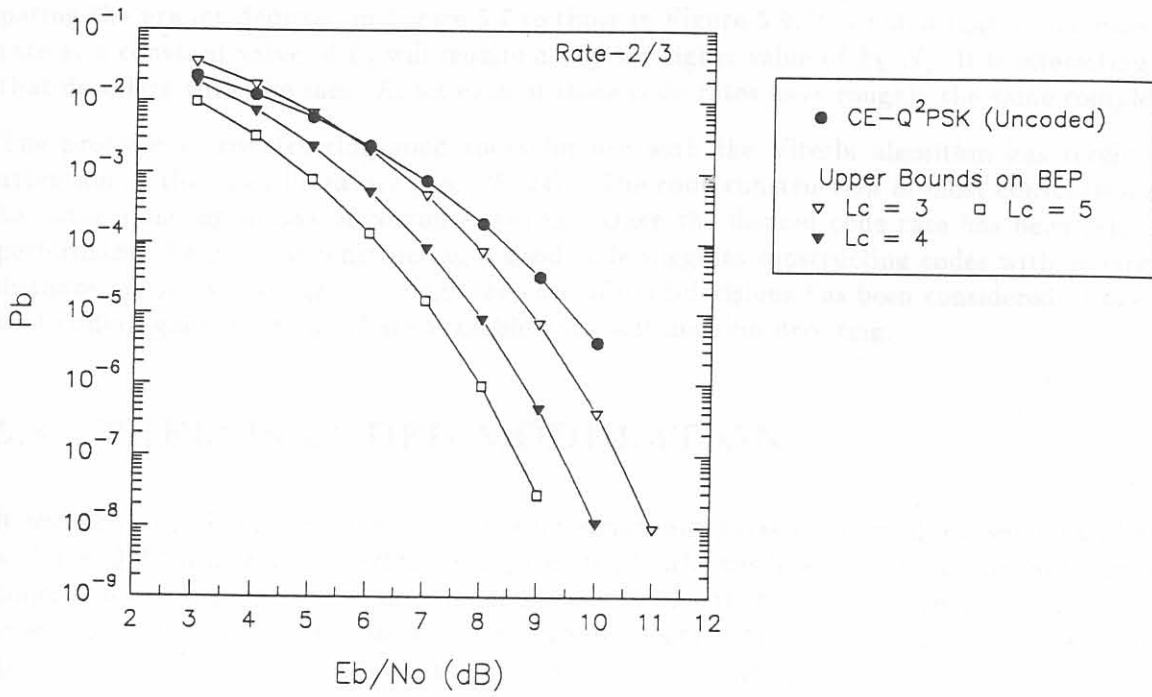


Figure 5.8: Bit error probability for Rate-2/3 codes with Viterbi decoding (Hard quantisation) and Q^2 PSK modulation.

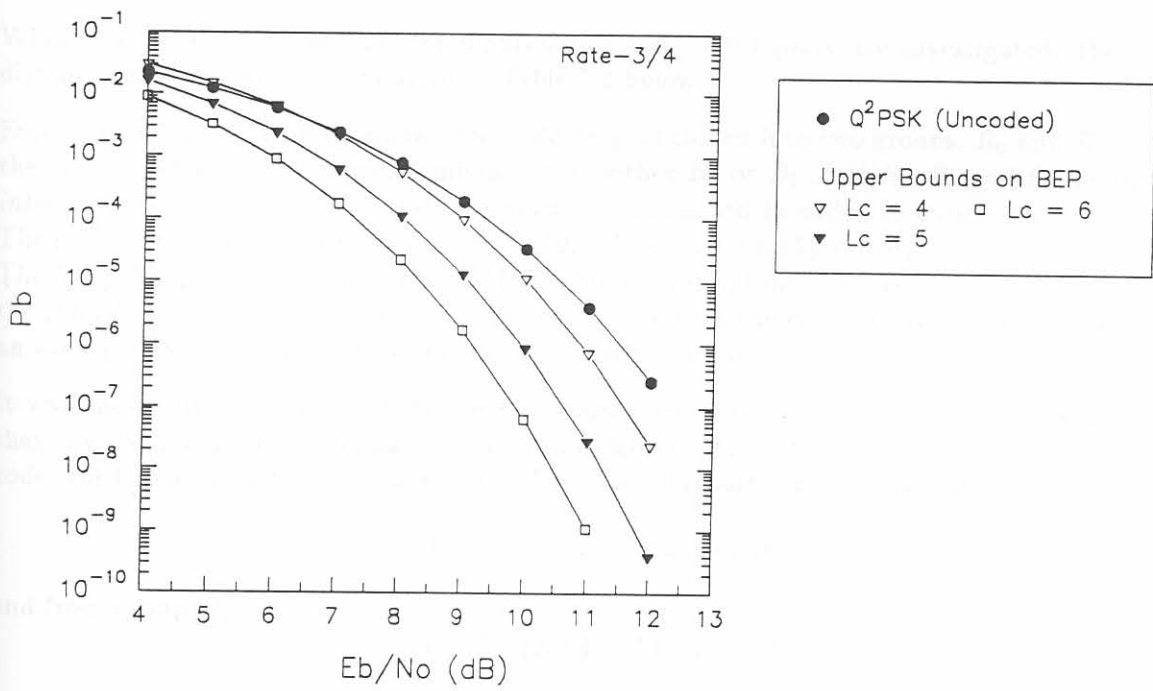


Figure 5.9: Bit error probability for Rate-3/4 codes with Viterbi decoding (Hard quantisation) and Q^2 PSK modulation.

The BER performance are for a BSC (i.e., hard-decision decoding) in an AWGN channel. Comparing the graphs depicted in Figure 5.7 to those in Figure 5.9, it is noted that an increase in code rate at a constant value of L_c will require a slightly higher value of E_b/N_o . It is interesting to note that decoders with the same L_c for each of these code rates have roughly the same complexity.

The problem of constructing good codes for use with the Viterbi algorithm has received much attention in the open literature (see [25, 24]). The code construction of most convolutional codes has been done by means of computer search. Once the desired code rate has been selected, the performance bounds for constructing a good code suggests constructing codes with as large a free distance, d_{free} as possible. Although the case of hard-decisions has been considered, it is expected that coding gains up to 3 dB are available with soft-decision decoding.

5.3 TRELIS CODED MODULATION

It was shown in Chapter 2 (section 2.3) that significant gains in spectral efficiency can be gained with the Q²PSK modulation scheme compared to 2D schemes by exploiting all available signal space dimensions within the given transmission bandwidth. Furthermore, utilisation of all available signal space dimensions may bring about further improvements in the coding gain. Therefore, high coding gains are expected when TCM techniques are combined with Q²PSK.

The Minimum Squared Euclidean Distance (MSED), d_{free}^2 between any pair of 4-tuples or code words $(a_1 a_2 a_3 a_4)$, say s_x and s_y , can be easily evaluated and is given by

$$d_{free}^2 = \sum_{i=1}^4 (a_i^{(x)} - a_i^{(y)})^2 \quad (5.3)$$

When the MSED between all combinations of code word pairs are investigated, the following distances are obtained (summarised in Table 5.1 below.)

From Table 5.1, the Q²PSK signal space can be partitioned into two groups, B_0 and B_1 , such that the MSED between any pair of symbols within either B_0 or B_1 is $d^2(B_0, B_1) = 8E_b$, known as the *intradistance* of a code. This distance must be maximised in order to ensure an optimum code. The resulting subgroups are given by $B_0 = \{0, 3, 5, 6, 9, 10, 12, 15\}$ and $B_1 = \{1, 2, 4, 7, 8, 11, 13, 14\}$. The foregoing partitioning results in a MSED between subgroups, B_0, B_1 of $4E_b$, which is known as the *interdistance* of the code. It is interesting to note that the code words from group B_0 maintain an even parity, while the code words from B_1 have odd parity.

It was shown by Saha in [12] that the minimum correlation among code words is achieved when they are antipodal. Considering the partitioned groups B_0 and B_1 , we identify four pairs of such code words out of each of the subgroups. These symbol pairs are, from group B_0 :

$$\{0, 15\}, \{3, 12\}, \{5, 10\}, \{6, 9\} \quad (5.4)$$

and from group B_1 :

$$\{1, 14\}, \{2, 13\}, \{4, 11\}, \{7, 8\} \quad (5.5)$$

Note that for these antipodal code word pairs the MSED is maximised. Therefore, we can partition subgroups B_0 and B_1 even further, such that the MSED between the antipodal symbol pair in any

Reference	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
s_0	4	4	8	4	8	8	12	4	8	8	12	8	12	12	16
s_1	—	8	4	8	4	12	8	8	4	12	8	12	8	16	12
s_2	—	—	4	8	12	4	8	8	12	4	8	12	16	8	12
s_3	—	—	—	12	8	8	4	12	8	8	4	16	12	12	8
s_4	—	—	—	—	4	4	8	8	12	12	16	4	8	8	12
s_5	—	—	—	—	—	8	4	12	8	16	12	8	4	12	8
s_6	—	—	—	—	—	—	4	12	16	8	12	8	12	4	8
s_7	—	—	—	—	—	—	—	16	12	12	8	12	8	8	4
s_8	—	—	—	—	—	—	—	—	4	4	8	4	8	8	12
s_9	—	—	—	—	—	—	—	—	—	8	4	8	4	12	8
s_{10}	—	—	—	—	—	—	—	—	—	—	4	8	12	4	8
s_{11}	—	—	—	—	—	—	—	—	—	—	—	12	8	8	4
s_{12}	—	—	—	—	—	—	—	—	—	—	—	—	4	4	8
s_{13}	—	—	—	—	—	—	—	—	—	—	—	—	—	8	4
s_{14}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4

Table 5.1: Minimum squared Euclidean distances between all pairs of Q²PSK code words.

group is $16E_b$, forming subgroups C_0, \dots, C_7 , each containing an antipodal code word pair. These subgroups are defined by

$$\begin{aligned} B_0 &= \{C_0, C_1, C_2, C_3\} \\ B_1 &= \{C_4, C_5, C_6, C_7\} \end{aligned} \quad (5.6)$$

where

$$\begin{aligned} C_0 &= \{0, 15\}, C_1 = \{3, 12\}, C_2 = \{5, 10\}, C_3 = \{9, 6\} \\ C_4 &= \{1, 14\}, C_5 = \{2, 13\}, C_6 = \{4, 11\}, C_7 = \{8, 7\} \end{aligned} \quad (5.7)$$

as illustrated in Figures 5.10 and 5.11.

Symbol	a_1 a_2 a_3 a_4	a_c b_c a_s b_s	4D Signal Components	4D Signal Space	Coder Output
C_0 0	-1 -1 -1 -1	-1 -1 -1 +1	$s_{13} + s_{22}$		1
15	+1 +1 +1 +1	+1 -1 +1 +1	$s_{11} + s_{24}$		-15
C_1 3	-1 -1 +1 +1	-1 +1 +1 -1	$s_{14} + s_{23}$		5
12	+1 +1 -1 -1	+1 +1 -1 -1	$s_{12} + s_{21}$		-11
C_2 5	-1 +1 -1 +1	-1 +1 -1 -1	$s_{12} + s_{23}$		9
10	-1 -1 +1 -1	+1 +1 +1 -1	$s_{14} + s_{21}$		-7
C_3 9	+1 -1 -1 +1	+1 -1 -1 +1	$s_{11} + s_{22}$		-3
6	-1 +1 +1 -1	-1 -1 +1 +1	$s_{13} + s_{24}$		13

 Figure 5.10: Group $B_0 = \{C_0, C_1, C_2, C_3\}$: Even Parity

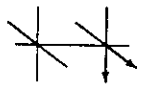
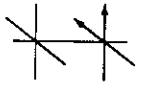
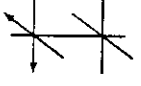
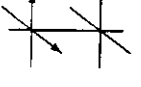
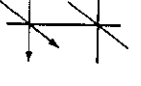
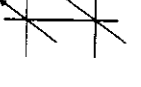
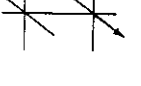
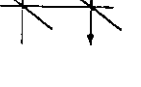
Symbol	a_1 a_2 a_3 a_4	a_c b_c a_s b_s	4D Signal Components	4D Signal Space	Coder Output	
C_4	1	-1 -1 -1 +1	-1 +1 -1 +1	$s_{22} + s_{23}$		3
	14	+1 +1 +1 -1	+1 +1 +1 +1	$s_{21} + s_{24}$		-13
C_5	2	-1 -1 +1 -1	-1 -1 +1 -1	$s_{13} + s_{14}$		7
	13	+1 +1 -1 +1	+1 -1 -1 -1	$s_{11} + s_{12}$		-9
C_6	4	-1 +1 -1 -1	-1 -1 -1 -1	$s_{12} + s_{13}$		11
	11	+1 -1 +1 +1	+1 -1 +1 -1	$s_{11} + s_{14}$		-5
C_7	8	+1 -1 -1 -1	+1 +1 -1 +1	$s_{21} + s_{22}$		-1
	7	-1 +1 +1 +1	-1 +1 +1 +1	$s_{23} + s_{24}$		15

Figure 5.11: Group $B_1 = \{C_4, C_5, C_6, C_7\}$: Odd Parity

The foregoing set partitioning of the Q^2 PSK signal space is graphical displayed in Figure 5.12.

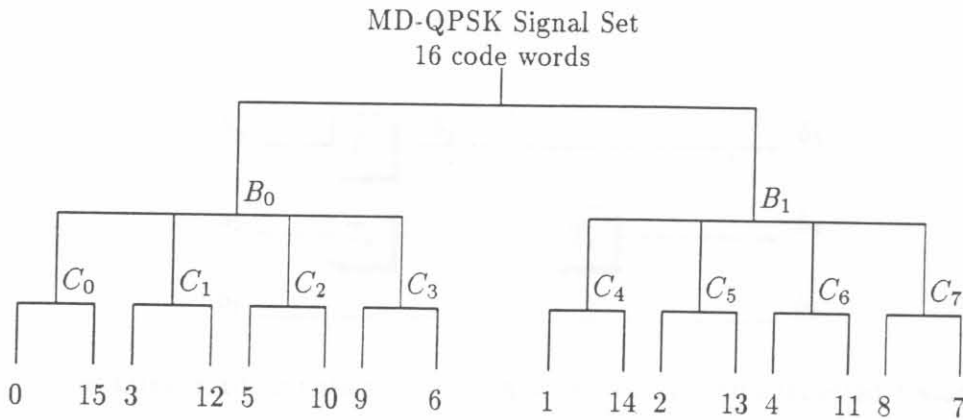


Figure 5.12: Q^2 PSK signal space partition tree.

In the design of the rate $3/4$ and rate $2/4$ Q^2 PSK trellis codes, the original 4D Q^2 PSK constellation (i.e. unexpanded) is utilised. Therefore, coding gains are obtained by sacrificing the effective data rate of the system.

For small code memory codes, the design can be produced "by hand", following the set partitioning concepts of Ungerboeck and then applying the analytical code design approach, presented in Appendix A. For more complex codes, say with more than 32 trellis states, a computer search for maximal free distance codes is definitely warranted. However, it will be shown that surprisingly large coding gains can be achieved with simple rate $3/4$ and $2/4$ TCM "hand" code designs.

5.3.1 Rate $3/4$ Q^2 PSK/TCM

In the following section, the design of codes for rate $3/4$ TCM are carried out. The $3/4$ code rate implies that the information transmission rate is reduced by a factor of $3/4$, resulting in a bandwidth efficiency of 1.5 bits/s/Hz , compared to the 2.0 bits/s/Hz of the uncoded case.

5.3.1.1 8-State Code design

The code structure for the half-connected code is presented in Figure 5.13, defining the sliding block of source variables $(b_1, b_2, b_3, b_4, b_5, b_6)$. The trellis diagram illustrated in Figure 5.14, has been labeled with the Q^2 PSK code words (presented in Figures 5.10 and 5.11), and the variables of the sliding block, (b_1, b_2, \dots, b_6) taking on the values 0 or 1. For the rate $3/4$ code, the 3 input bits (b_1, b_2, b_3) are input to the encoder, producing the 4 output bits (b_1, b_4, b_5, b_6) . The values of the output bits are not only determined by the current input bits, but also the encoder state formed by bits (b_4, b_5, b_6) . The 4-tuple code word (b_1, b_4, b_5, b_6) produced, must then be converted to values of ± 1 instead of 0 and 1, producing a Q^2 PSK symbol being output as presented in Figures 5.10 and 5.11.

Substituting the coder output code words for the Q^2 PSK code words, and by using the Calderbank—Mazo algorithm (presented in Appendices A and B), the following solutions for the D matrix is

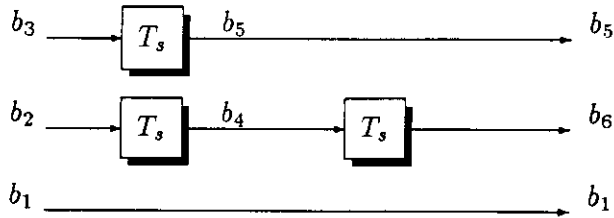


Figure 5.13: Inputs and state variables for the half-connected 8-state code.

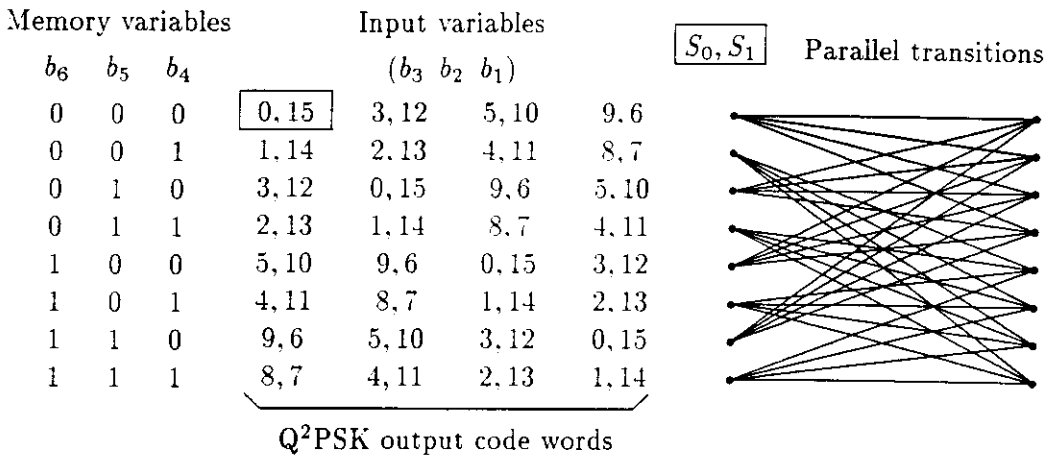


Figure 5.14: Half-connected $R = 3/4$ 8-state trellis code.

obtained:

$$d_1 = 8, d_4 = -1, d_{25} = -2, d_{36} = -4 \tag{5.8}$$

The analytical description of the encoder is:

$$X = 8b_1 - b_4 - 2b_2b_5 - 4b_3b_6 \tag{5.9}$$

where X denotes the coder output code word.

The achievable coding gain of the 8-state code is limited by parallel transitions which occur when two different symbols can cause a transition along the same trellis branch. The minimum free distance, d_{free}^2 of the code is equal to $d_{free,c}^2 = 8E_b$, where E_b is the average energy per bit in the coded signals. (For uncoded Q²PSK the minimum free distance, $d_{free,u}^2 = 4E_b$). Therefore an asymptotic coding gain (defined in (A.4)), $\gamma_c = 10 \log_{10} d_{free,c}^2/d_{free,u}^2 = 10 \log_{10} 8E_b/4E_b = 3.01$ dB is obtained for the 8 state trellis codes, given by a length $L = 1$ Error Event Path (EEP).

The code structure for the fully-connected code is presented in Figure 5.15, again defining the sliding block of source variables ($b_1, b_2, b_3, b_4, b_5, b_6$). No parallel transition is included in the code structure. The corresponding trellis diagram is illustrated in Figure 5.16.

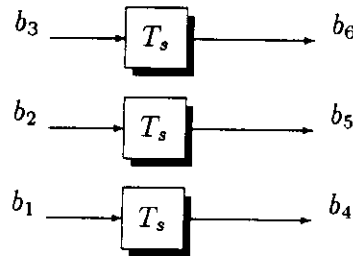


Figure 5.15: Inputs and state variables for fully-connected 8-state code.

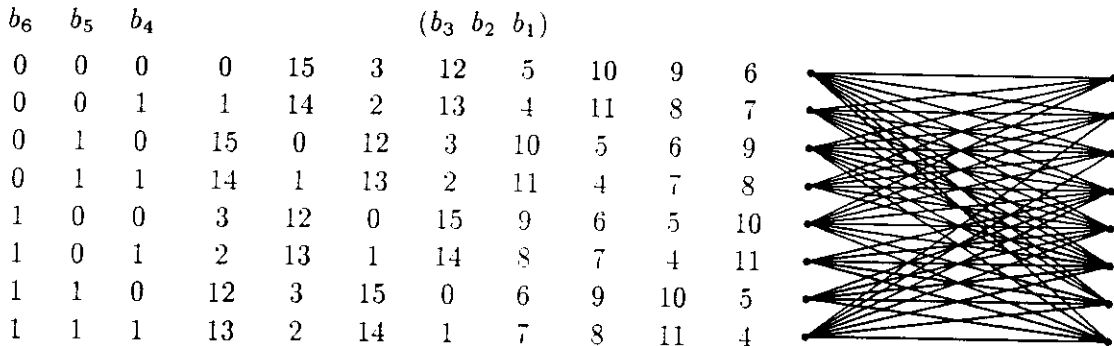


Figure 5.16: Fully-connected $R = 3/4$ 8-state trellis code.

The d_{free}^2 of the code is equal to $12E_b$, representing an asymptotic coding gain of $10 \log_{10} 12/4 = 4.77$ dB achievable with this 8 state trellis codes. The EEP is $L = 3$, since the code performance is not limited by parallel transitions.

5.3.2 Rate 2/4 Q²PSK/TCM

In the following discussion, the design of codes for rate 2/4 TCM are carried out. The information transmission rate is reduced by a factor of 2/4, resulting in a reduction in bandwidth efficiency to

1.0 bits/s/Hz. In the design of the new rate 2/4 Q²PSK trellis codes, selecting code words from the Q²PSK code words that maintain odd parity, a constant envelope Q²PSK signal is obtained which improves performance on non-linear channels.

Learning from the experience gained from the Rate-3/4 code designs, we consider in the following only coders excluding parallel transitions in their trellis structures. The code structures for the half-rate coders are presented in Figure 5.17, defining the sliding block of source variables (b_1, \dots, b_6).

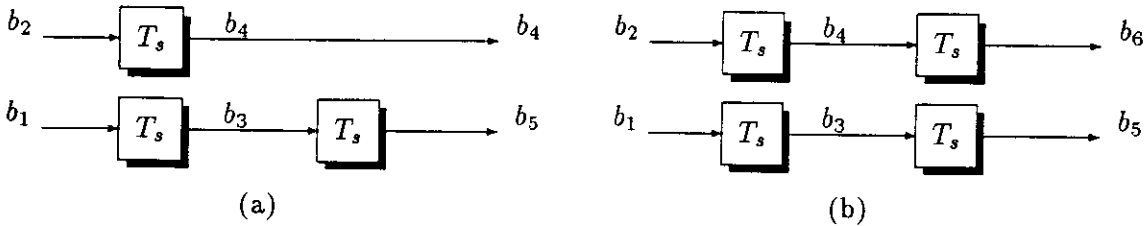


Figure 5.17: Code structures for half-rate trellis coders.

5.3.2.1 Non-Constant Envelope

b_5	b_4	b_3		$(b_2 \ b_1)$		
0	0	0	0	15	3	12
0	0	1	1	14	2	13
0	1	0	5	10	9	6
0	1	1	4	11	8	7
1	0	0	15	0	12	3
1	0	1	14	1	13	2
1	1	0	10	5	6	9
1	1	1	11	4	7	8

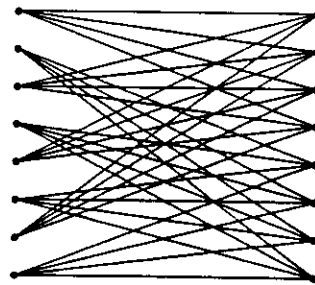


Figure 5.18: Half-connected $R = 2/4$ 8-state trellis code.

The trellis diagram of the 8-state code is illustrated in Figure 5.18, leading to the analytical description of the encoder, given by:

$$X = 8b_1b_5 - 4b_4 - 2b_2 - b_3 \tag{5.10}$$

The analytical description of the encoder, depicted in Figure 5.19 is given by:

$$X = 7b_1b_5 - 3b_4 - b_2b_6 - b_3 \tag{5.11}$$

Half rate trellis codes for Q²PSK signals have been reported by Saha and Acha [47, 23, 48], different 2, 8 and 16 state trellis codes were designed, and asymptotic coding gains up to 6.0 dB were calculated. However, a closer examination of the 2/4 rate codes proposed by Saha revealed a few

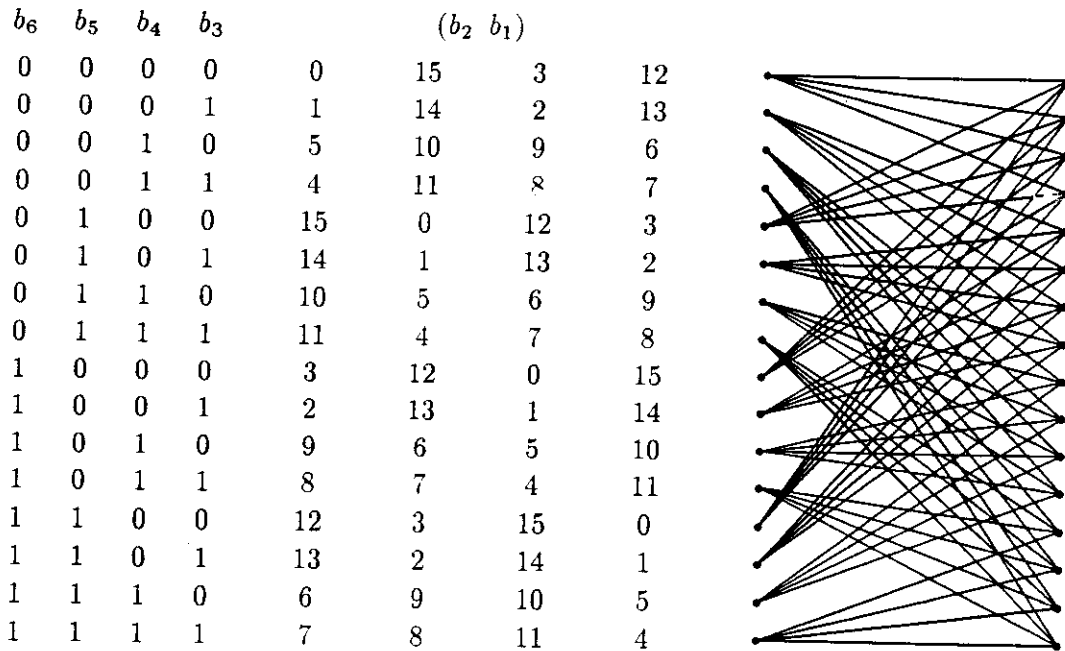


Figure 5.19: Quarter-connected $R = 2/4$ 16-state trellis code.

catastrophic deficiencies. Firstly, the trellis labeling proposed by Saha for the 2 state $2/4$ rate code leads to catastrophic error propagation. This can be attributed to the fact that Saha incorporated no redundant information into the symbol sequences, resulting in infinite error propagation.

5.3.2.2 Constant Envelope

The analytical description of the encoder, depicted in Figure 5.20, is given as:

$$X = 7b_1b_5 - 3b_4 - b_2b_6 - b_3 \quad (5.12)$$

The squared minimum distance, d_{free}^2 of this code is equal to $24E_b$ resulting in an asymptotic coding gain of $10 \log_{10} 16/4 = 6.02$ dB. Note, that the coding gain of this 8-state code is not limited by parallel transitions and that the number of nearest neighbors at a distance of d_{free} , is reduced from 4 to 2.

5.3.3 Q^2 PSK/TCM code performance

In this section the performance evaluation of TCM systems when utilised on AWGN channels are derived. The classical performance measure for TCM, is the asymptotic coding gain, which is directly determined by the MSED of the code (see Appendix A). While this is an indication of the limit of the code performance, it can be an unreliable measure to use under practical operational conditions. For this reason, performance evaluation of trellis coded systems are usually accomplished by derivation of the upper and lower bounds of the code if the convolutional coders are linear.

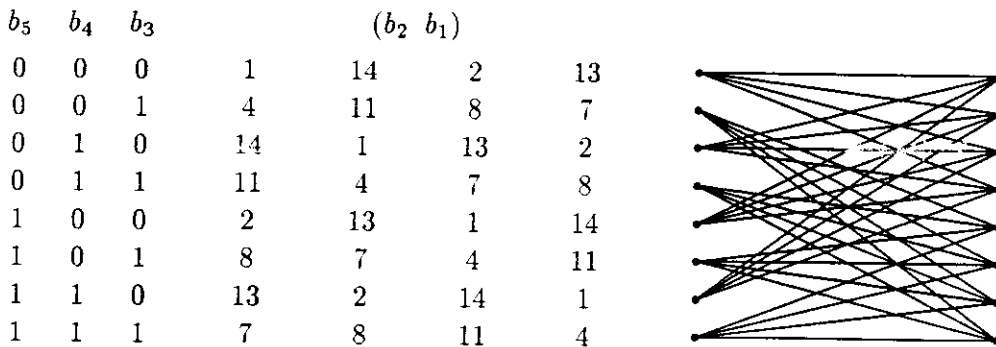


Figure 5.20: Half-connected $R = 2/4$ 8-state trellis code for Constant Envelope (CE) Q^2 PSK.

If the code is non-linear theoretical performance calculation becomes extremely cumbersome, and computer simulations are commonly used to evaluate code performance.

In the following section upper bounds for TCM schemes are derived when employed on the AWGN channel based on the so-called transfer function bound [30]. Furthermore, we will discuss the application of a new algebraic algorithm for generating the transfer function of a trellis encoder, proposed by Chan and Norton [79] (presented in detail in Appendix C).

There are basically two ways to generate the transfer function. The first one is by graphical means — the state diagram is first drawn and properly labeled. The transfer function can then be found by a graph-reduction technique, or Mason's rule. The second approach is algebraic, i.e., solving state equations, or performing matrix inversion. All of these methods have a common difficulty: The complexity of calculation increases exponentially as the number of encoder states increases.

5.3.4 Analytical upper performance bounds for Q^2 PSK/TCM

As an example we consider the derivation of the tight upper bound on BEP for the 8-state fully-connected rate-3/4 trellis code presented earlier in this chapter (see section 5.3.1).

In order to calculate the transfer function one has to utilise the error weight profiles, and the trellis diagram showed in Figure 5.15. The error weights are easily calculated as defined according to (A.18) for the error vector $\bar{e} = (e_1, e_2, e_3, e_4)$, and are given by

$$\begin{aligned}
 W(0000) &= 1 & W(0001) &= D^4 & W(0010) &= D^4 & W(0011) &= D^8 \\
 W(0100) &= D^4 & W(0101) &= D^8 & W(0110) &= D^8 & W(0111) &= D^{12} \\
 W(1000) &= D^4 & W(1001) &= D^8 & W(1010) &= D^8 & W(1011) &= D^{12} \\
 W(1100) &= D^8 & W(1101) &= D^{12} & W(1110) &= D^{12} & W(1111) &= D^{16}
 \end{aligned} \tag{5.13}$$

Following the algebraic approach described in Appendix C, the transfer function of this encoder (after some intermediate algebra) is given by

$$T(D) = \frac{D^{20} + 24D^{36} - 16D^{18} + 48D^8 + 24D^{24}}{2 + 18D^4 - 8D^8 + D^{12}} \tag{5.14}$$

By substituting $D = \exp(-E_b/N_o)$ in equation (5.14), one obtains the upper bound (UB1) to the BEP. The derivation of the tighter upper bound (UB2) requires knowledge of the minimum free distance, d_{free} of the code, which was computed by the algorithm proposed by Mulligan and Wilson [80], and discussed in Appendix B. The calculation gave $d_{free} = 3.464$. Substituting the values for D and d_{free} into equation (A.21), the tighter upper bound (UB2) is obtained.

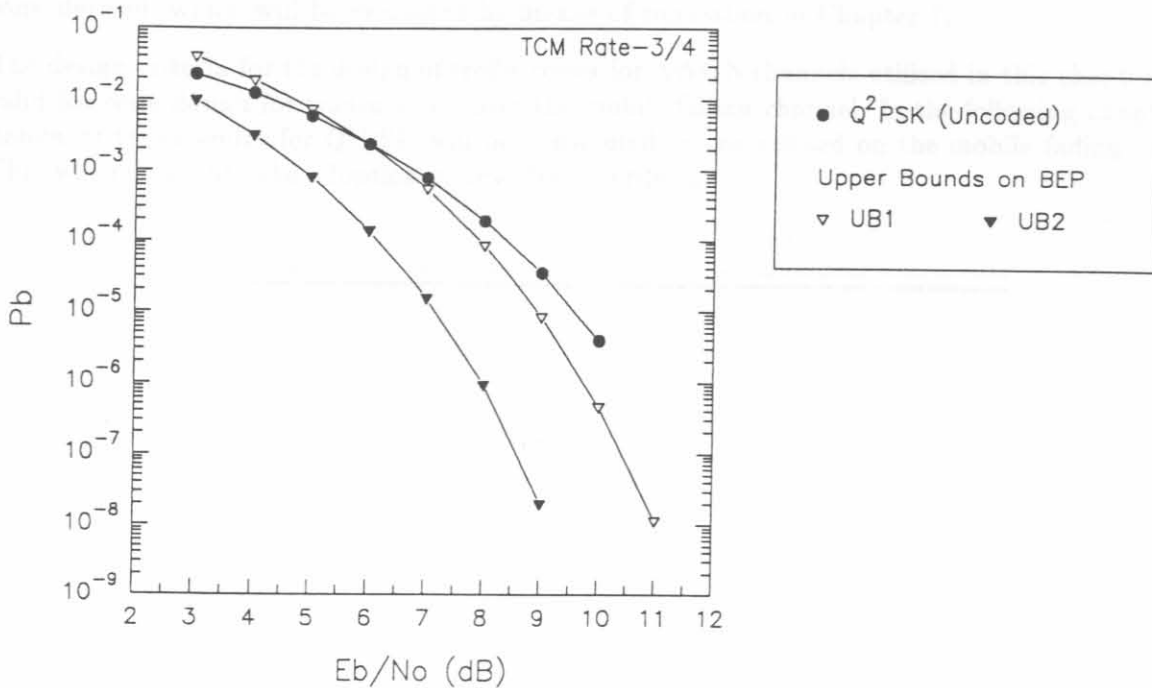


Figure 5.21: Upper bounds to bit error probabilities for 8-state rate-3/4 Q²PSK/TCM.

The calculated upper bounds to the BEP have been calculated for this trellis coder and are shown in Figure 5.21. The curve defined as UB1 denotes the upper bound calculated by equation (A.20), and that defined as UB2 corresponds to the bound calculated with (A.21), i.e., the tighter bound on the BEP.

5.4 CONCLUDING REMARKS: CHAPTER 5

In this chapter the designs of classical and TCM codes were carried out, performed for a Q²PSK system when utilised on the AWGN channel.

For the classical code designs, low-complexity sub-optimal convolutional encoding schemes were proposed for Q²PSK. Three coding schemes were presented: The first scheme employs a single rate-3/4 convolutional encoder, yielding a effective data rate of 3/4 uncoded Q²PSK. The second encoding scheme, with so called *dual-use* of conventional 2D signals, employs two rate-1/2 convolutional encoders. At the receiver two soft-decision Viterbi decoders are utilised in parallel. Finally, a third encoding scheme was proposed, employing a hybrid convolutional-block coding scheme. The encoder consists of a rate-2/3 convolutional encoder, followed by a simple block encoding scheme to produce a constant envelope. At the receiver a single soft-decision Viterbi decoder, together with a block decoding scheme, is employed. Performance estimates of these classical coders were

carried out based on the union bound technique. Upper bounds to BEP were derived. In Chapter 7 these bounds will be benchmarked against actual simulation curves.

The final sections of this chapter dealt with the design of TCM codes for Q^2 PSK when utilised on the AWGN channel. For these channels, the parameter to be maximised, is the Euclidean distance of the TCM code. The designs and realisations for rate 3/4 and 2/4 "hand designed" trellis codes were presented. In addition, upper bounds on the BEP for a specific 8-state rate 3/4 trellis code were derived, which will be evaluated by means of simulation in Chapter 7.

The design criteria for the design of trellis codes for AWGN channels utilised in this chapter, is not valid for code design for transmission over the mobile fading channel. In the following chapter, the design of trellis codes for Q^2 PSK will be considered, when utilised on the mobile fading channel. This will necessitate the adoption of new design criteria.

DESIGN OF TCM AND MTCM FOR FADING CHANNELS