Çankaya University – ECE Department – ECE 587

Student Name : Duration : 2 hours
Student Number : Open book exam

Questions

(70 Points) The convolutional encoder shown below is intended to convert 4 PSK into 8 PSK TCM. Find the output by hand and using Matlab m file Conv12_Exp2.m, if the input message sequence is $\mathbf{x} = [1,\ 0,\ 1, 1,\ 0,\ 0,\ 1,\ 1]$. Draw the related state trellis diagram. Organize the output (for all possible inputs) in the form of Tables similar to Table 5.1 of ECE 587_Notes on Codes. From there and the state trellis diagram, make an assignment of symbols according to partition rules of Ungerboeck, then draw the constellation diagram of 8 PSK TCM. Using the trellis diagram, place the signal vectors correctly in TCM_Pe8PSK.m which compares the probability of error performance of 8 PSK TCM against uncoded 4 PSK for the encoder configuration in Fig. 1.1 and constellation that you have derived. Use the model files TCM_Dec.mdl and TCM_Enc.mdl during these runs. All Matlab files are available on course webpage.

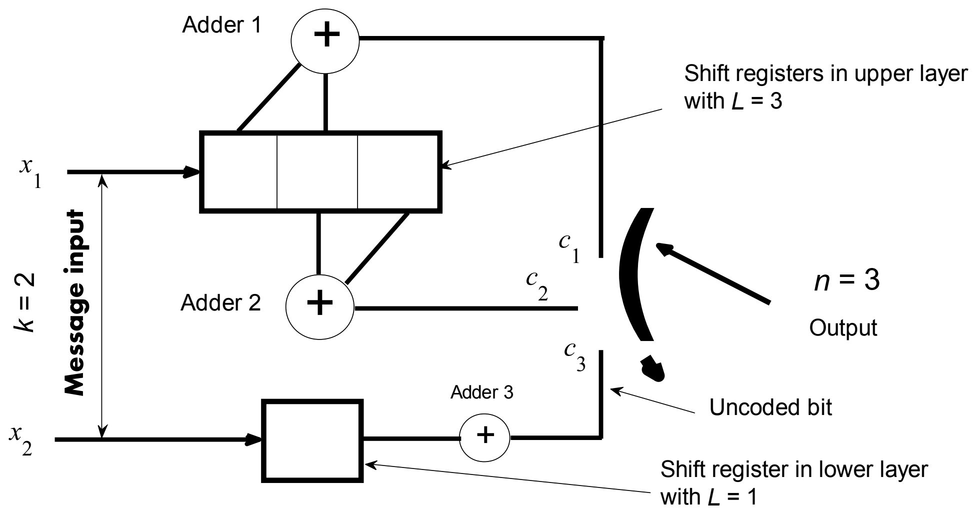

Fig. 1.1 Convolution encoder for Q1.

Solution : We assume that the shift registers are initially loaded with zeros, then by hand tracing we get the following

Input	Contents of shift registers in upper layer	Adder 1	Adder 2	Adder 3	**c** - Output $c_1\ c_2\ c_3$	
	0 0 0	0	0	0	0 0 0	(Prior to any input)
1 0	1 0 0	1	0	0	1 0 0	
1 1	1 1 0	0	1	1	0 1 1	
0 0	0 1 1	1	0	0	1 0 0	
1 1	1 0 1	1	1	1	1 1 1	

$$\mathbf{x}^t = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \qquad (1.1)$$

So the output of the convolutional encoder shown in Fig. 1.1 is

$$\begin{aligned} \mathbf{x} &= [1,\ 0, \quad 1, 1, \quad 0, 0, \quad 1, 1] \\ & \qquad \downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow \\ \mathbf{c} &= [\overbrace{1\,0\,0}^{10},\ \overbrace{0\,1\,1}^{11},\ \overbrace{1\,0\,0}^{00},\ \overbrace{1\,1\,1}^{11}] \end{aligned} \qquad (1.2)$$

By considering Fig. 1.1 and using (4.5) and (4.6) of ECE 587_Notes on Codes

$$\text{Adder 1} = \begin{bmatrix} 1, & 1, & 0 \\ 0, & 0, & 0 \end{bmatrix} \quad \text{Adder 2} = \begin{bmatrix} 0, & 1, & 1 \\ 0, & 0, & 0 \end{bmatrix} \quad \text{Adder 3} = \begin{bmatrix} 0, & 0, & 0 \\ 0, & 0, & 1 \end{bmatrix} \qquad (1.3)$$

We write for the poly2trellis function of Matlab as follows

Constraint length (number of shift registers in upper and lower layers) - L → [3, 1]

Connection of Adder 1 to shift registers in upper layer → 6

Connection of Adder 3 to shift registers in upper layer → 0

t = poly2trellis([3, 1], [6, 3, 0;0 0 1])

Connection of Adder 2 to shift registers in upper layer → 3

Connection of Adder 3 to shift register in lower layer → 1 (1.4)

Subsequently running Conv12_Exp2.m with the input in (1.2) and the ploys2trellis function of Matlab given in (1.4), we get the same output as written on the final line of (1.2).

Either by hand or by using Conv12_Exp2.m, we get the state trellis diagram drawn in Fig. 1.2. From there we derive Table 1.1 for originating states and Table 1.2 for terminating states.

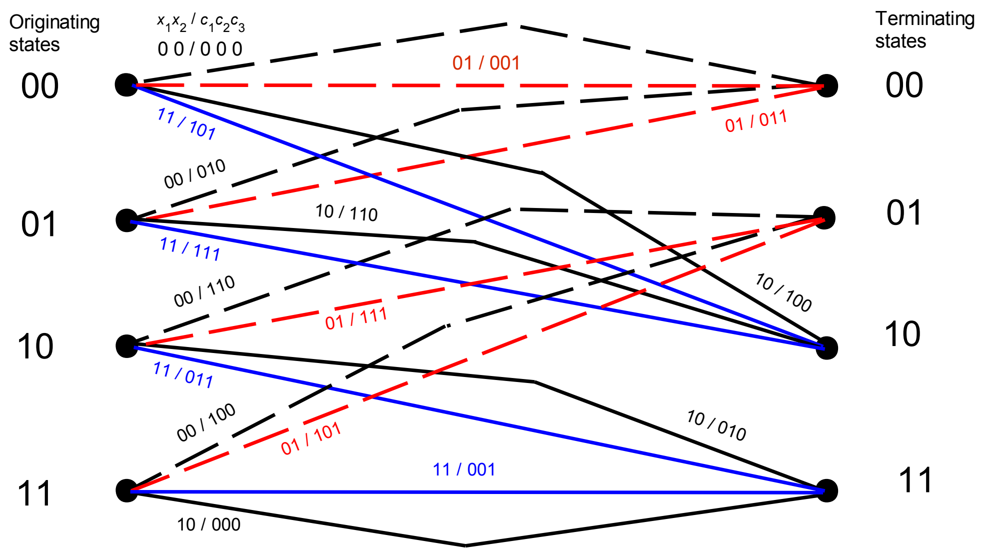

Fig. 1.2 State trellis diagram of the TCM encoder in Fig. 1.1.

Originating state	Input / Output symbol	**s** designation of output sysmbol	**B** designation	Parallel paths
00	00/000	$\mathbf{s}_1$	$\mathbf{B}_{1o}$	Parallel
00	01/001	$\mathbf{s}_2$	$\mathbf{B}_{1o}$	
00	10/100	$\mathbf{s}_5$	$\mathbf{B}_{1o}$	Parallel
00	11/101	$\mathbf{s}_6$	$\mathbf{B}_{1o}$	
01	00/010	$\mathbf{s}_3$	$\mathbf{B}_{2o}$	Parallel
01	01/011	$\mathbf{s}_4$	$\mathbf{B}_{2o}$	
01	10/110	$\mathbf{s}_7$	$\mathbf{B}_{2o}$	Parallel
01	11/111	$\mathbf{s}_8$	$\mathbf{B}_{2o}$	
10	00/110	$\mathbf{s}_7$	$\mathbf{B}_{2o}$	Parallel
10	01/111	$\mathbf{s}_8$	$\mathbf{B}_{2o}$	
10	10/010	$\mathbf{s}_3$	$\mathbf{B}_{2o}$	Parallel
10	11/011	$\mathbf{s}_4$	$\mathbf{B}_{2o}$	
11	00/100	$\mathbf{s}_5$	$\mathbf{B}_{1o}$	Parallel
11	01/101	$\mathbf{s}_6$	$\mathbf{B}_{1o}$	
11	10/000	$\mathbf{s}_1$	$\mathbf{B}_{1o}$	Parallel
11	11/001	$\mathbf{s}_2$	$\mathbf{B}_{1o}$	

Table 1.1 Organization of the originating state information in Fig. 1.2 in the form of a table.

Terminating state	Output symbol	$\mathbf{s}$ designation	B designation	Parallel paths
00	00/000	$\mathbf{s}_1$	$\mathbf{B}_{1t}$	Parallel
00	01/001	$\mathbf{s}_2$	$\mathbf{B}_{1t}$	
00	00/010	$\mathbf{s}_3$	$\mathbf{B}_{1t}$	Parallel
00	01/011	$\mathbf{s}_4$	$\mathbf{B}_{1t}$	
01	00/110	$\mathbf{s}_7$	$\mathbf{B}_{2t}$	Parallel
01	01/111	$\mathbf{s}_8$	$\mathbf{B}_{2t}$	
01	00/100	$\mathbf{s}_5$	$\mathbf{B}_{2t}$	Parallel
01	01/101	$\mathbf{s}_6$	$\mathbf{B}_{2t}$	
10	10/100	$\mathbf{s}_5$	$\mathbf{B}_{2t}$	Parallel
10	11/101	$\mathbf{s}_6$	$\mathbf{B}_{2t}$	
10	10/110	$\mathbf{s}_7$	$\mathbf{B}_{2t}$	Parallel
10	11/111	$\mathbf{s}_8$	$\mathbf{B}_{2t}$	
11	10/010	$\mathbf{s}_3$	$\mathbf{B}_{1t}$	Parallel
11	11/011	$\mathbf{s}_4$	$\mathbf{B}_{1t}$	
11	10/000	$\mathbf{s}_1$	$\mathbf{B}_{1t}$	Parallel
11	11/001	$\mathbf{s}_2$	$\mathbf{B}_{1t}$	

Table 1.2 Organization of the terminating state information in Fig. 1.2 in the form of a table.

From Table 1.1, we have the following $\mathbf{B}$ square designations.

$$\mathbf{B}_{1o} = (000,\ 001,\ 100,\ 101) \quad , \quad \mathbf{B}_{2o} = (010,\ 011,\ 110,\ 111)$$
$$\mathbf{B}_{1o} = (\ \mathbf{s}_1,\ \ \mathbf{s}_2,\ \ \mathbf{s}_5,\ \ \mathbf{s}_6) \quad , \quad \mathbf{B}_{2o} = (\ \mathbf{s}_3,\ \ \mathbf{s}_4,\ \ \mathbf{s}_7,\ \ \mathbf{s}_8) \tag{1.5}$$

Whereas, the square designations of Table 1.2, are the followings.

$$\mathbf{B}_{1t} = (000,\ 001,\ 010,\ 011) \quad , \quad \mathbf{B}_{2t} = (100,\ 101,\ 110,\ 111)$$
$$\mathbf{B}_{1t} = (\ \mathbf{s}_1,\ \ \mathbf{s}_2,\ \ \mathbf{s}_3,\ \ \mathbf{s}_4) \quad , \quad \mathbf{B}_{2t} = (\ \mathbf{s}_5,\ \ \mathbf{s}_6,\ \ \mathbf{s}_7,\ \ \mathbf{s}_8) \tag{1.6}$$

Comparing (1.5) to (1.6), we do not find the same $\mathbf{B}$ designations for originating and terminating cases. Recall that in Table 5.1 of ECE 587_Notes on Codes, the same $\mathbf{B}$ designations for originating and terminating cases cloud be made. From these observations, we deduce that it is not possible to apply partitioning rules of Ungerboeck to the encoder of Fig. 1.1 and obtain desirable probability of error performance in 8 PSK TCM, compared with uncoded 4 PSK. To illustrate this, we use Table 1.1, or (1.5) and assign the symbols of 8 PSK TCM to two $\mathbf{B}$ squares as shown in Fig. 1.3. Running TCM_Pe8PSK.m with the constellation of Fig. 1.3, we get the curves in Fig. 1.4. As seen from Fig. 1.4, the performance of 8 PSK TCM based on the encoder configuration of Fig. 1.1 is unacceptable.

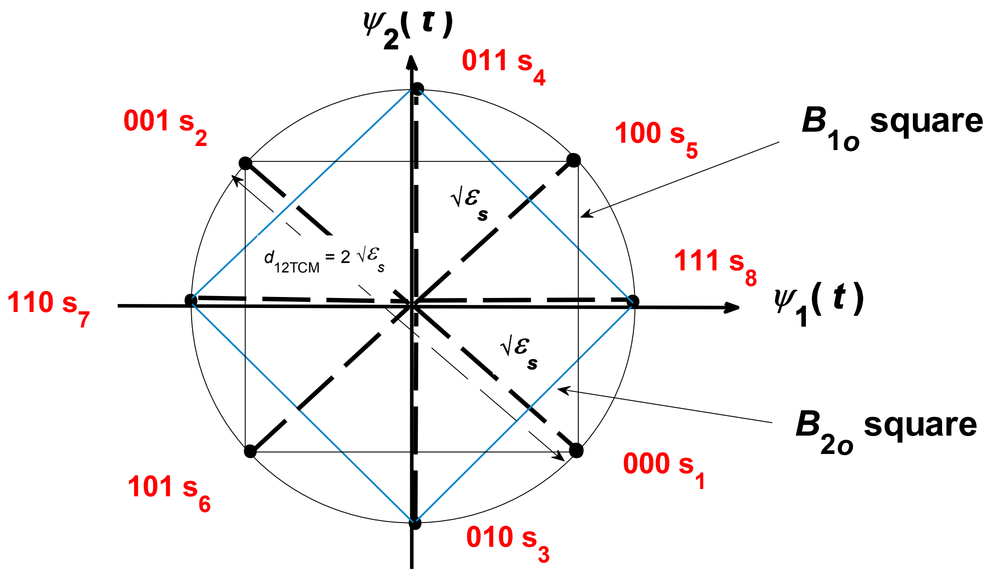

Fig. 1.3 8 PSK TCM constellation using Table 1.1.

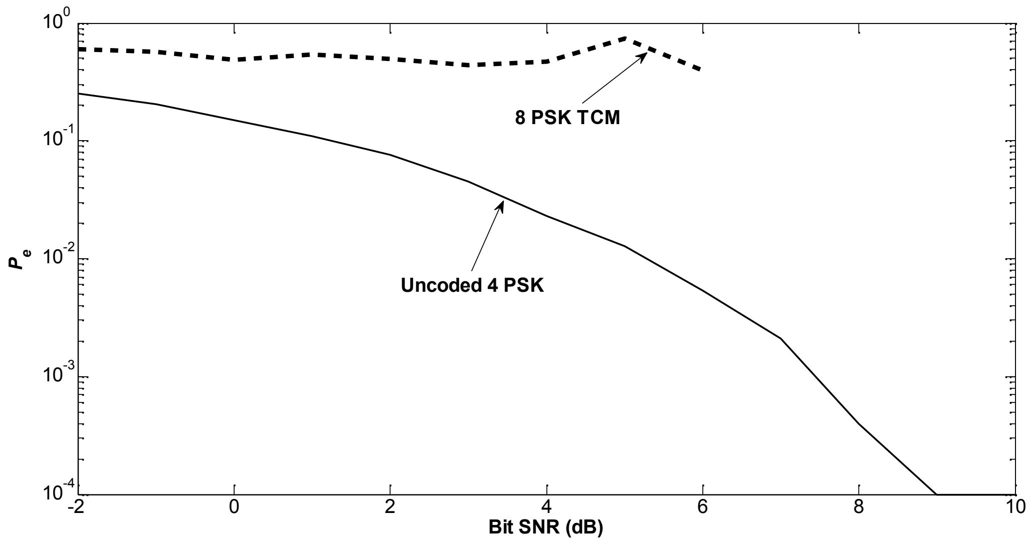

Fig. 1.4 Probability of error curves for uncoded 4 PSK and for 8 PSK TCM constellation of Fig. 1.3.

2. (30 Points) Answer the following questions as **True** or **False.** For the **False** ones give the correct answer or the reason. For the **True** ones justify your answer.

a) Coding increases the linearity property : False, coding increases the dimensionality of the signal, hence the bandwidth requirement.

b) There is memory, i.e. dependence on future in TCM coding : The first part is true, but the second part should say "past", instead of "future".

c) We have only systematic way of generating cyclic codes : False, on page 12 of ECE 587_Notes on Codes, there is systematic and non-systematic ways of generating cyclic codes.

d) In TCM the symbols on parallel paths are placed in different B squares : False, parallel paths are placed in the diagonal corners of the same B square.

e) In TCM, the minimum distance between signal vector ends is more than the uncoded case : True, as proven in (5.3) of ECE 587_Notes on Codes.

f) Parity matrix is obtained from parity bits : True as shown in (2.17) and (2.18) of of ECE 587_Notes on Codes.