Problem 8.1

(a) The encoder for the \((3, 1)\) convolutional code is depicted in the next figure.

(b) The state transition diagram for this code is depicted in the next figure.

(c) In the next figure we draw two frames of the trellis associated with the code. Solid lines indicate an input equal to 0, whereas dotted lines correspond to an input equal to 1.

(d) The diagram used to find the transfer function is shown in the next figure.
Using the flow graph results, we obtain the system

\[
\begin{align*}
X_c &= D^3NJX_{a'} + NJX_b \\
X_b &= D^2JX_c + DJX_d \\
X_d &= DJX_c + D^2NJX_d \\
X_{a''} &= D^3JX_b 
\end{align*}
\]

Eliminating \(X_b\), \(X_c\) and \(X_d\) results in

\[
T(D, N, J) = \frac{X_{a''}}{X_{a'}} = \frac{D^8NJ^3(1 + NJ - D^2NJ)}{1 - D^2NJ(1 + NJ^2 + J - D^2J^2)}
\]

To find the free distance of the code we set \(N = J = 1\) in the transfer function, so that

\[
T_1(D) = T(D, N, J)|_{N=J=1} = \frac{D^8(1 - 2D^2)}{1 - D^2(3 - D^2)} = D^8 + 2D^{10} + \ldots
\]

Hence, \(d_{\text{free}} = 8\)

(e) Since there is no self loop corresponding to an input equal to 1 such that the output is the all zero sequence, the code is not catastrophic.

Problem 8.2

The code of Problem 8-1 is a \((3, 1)\) convolutional code with \(K = 3\). The length of the received sequence \(y\) is 15. This means that 5 symbols have been transmitted, and since we assume that the information sequence has been padded by two 0’s, the actual length of the information sequence is 3. The following figure depicts 5 frames of the trellis used by the Viterbi decoder. The numbers on the nodes denote the metric (Hamming distance) of the survivor paths (the non-survivor paths are shown with an X). In the case of a tie of two merging paths at a node, we have purged the upper path.
The decoded sequence is \{111,100,011,100,111\} (i.e the path with the minimum final metric - heavy line) and corresponds to the information sequence \{1,1,1\} followed by two zeros.

Problem 8.3

(a) The encoder for the (3, 1) convolutional code is depicted in the next figure.

(b) The state transition diagram for this code is shown below
(c) In the next figure we draw two frames of the trellis associated with the code. Solid lines indicate an input equal to 0, whereas dotted lines correspond to an input equal to 1.

(d) The diagram used to find the transfer function is shown in the next figure.

Using the flow graph results, we obtain the system

\[
\begin{align*}
X_c &= D^3 N J X_{a'} + DNJ X_b \\
X_b &= D^2 J X_c + D^2 J X_d \\
X_d &= DN J X_c + DN J X_d \\
X_{a''} &= D^2 J X_b
\end{align*}
\]

Eliminating \(X_b\), \(X_c\), and \(X_d\) results in

\[
T(D, N, J) = \frac{X_{a''}}{X_{a'}} = \frac{D^7 N J^3}{1 - DNJ - D^3 N J^2}
\]

To find the free distance of the code we set \(N = J = 1\) in the transfer function, so that

\[
T_1(D) = T(D, N, J)|_{N=J=1} = \frac{D^7}{1 - D - D^3} = D^7 + D^8 + D^9 + \cdots
\]

Hence, \(d_{\text{free}} = 7\)

(e) Since there is no self loop corresponding to an input equal to 1 such that the output is the all zero sequence, the code is not catastrophic.
Problem 8.4

(a) The state transition diagram for this code is depicted in the next figure.

(b) The diagram used to find the transfer function is shown in the next figure.

Using the flow graph results, we obtain the system

\[
\begin{align*}
X_c &= D^3NJX_{a'} + DJNX_b \\
X_b &= DJX_c + DJX_d \\
X_d &= D^2NJX_c + D^2NJX_d \\
X_{a''} &= D^2JX_b
\end{align*}
\]

Eliminating \(X_b, \quad X_c\) and \(X_d\) results in

\[
T(D, N, J) = \frac{X_{a''}}{X_{a'}} = \frac{D^6NJ^3}{1 - D^2NJ - D^2NJ^2}
\]

(c) To find the free distance of the code we set \(N = J = 1\) in the transfer function, so that

\[
T_1(D) = T(D, N, J)|_{N=J=1} = \frac{D^6}{1 - 2D^2} = D^6 + 2D^8 + 4D^{10} + \cdots
\]

Hence, \(d_{\text{free}} = 6\)

(d) The following figure shows 7 frames of the trellis diagram used by the Viterbi decoder. It is assumed that the input sequence is padded by two zeros, so that the actual length of the information
sequence is 5. The numbers on the nodes indicate the Hamming distance of the survivor paths. The deleted branches have been marked with an X. In the case of a tie we deleted the upper branch. The survivor path at the end of the decoding is denoted by a thick line.

The information sequence is 11110 and the corresponding codeword 111 110 101 101 010 011 000...

(e) An upper to the bit error probability of the code is given by

\[ P_b \leq \left. \frac{dT(D, N, J = 1)}{dN} \right|_{N=1, D = \sqrt{4p(1-p)}} \]

But

\[ \frac{dT(D, N, 1)}{dN} = \frac{d}{dN} \left[ \frac{D^6 N}{1 - 2D^2 N} \right] = \frac{D^6 - 2D^8(1 - N)}{(1 - 2D^2 N)^2} \]

and since \( p = 10^{-5} \), we obtain

\[ P_b \leq \left. \frac{D^6}{(1 - 2D^2)^2} \right|_{D = \sqrt{4p(1-p)}} \approx 6.14 \cdot 10^{-14} \]

Problem 8.5

(a) The state transition diagram for this code is shown below
(b) The diagram used to find the transfer function is shown in the next figure.

Using the flow graph results, we obtain the system

\[
\begin{align*}
X_c &= D^3N J X_{a'} + DN J X_b \\
X_b &= D^2 J X_c + D^2 J X_d \\
X_d &= DN J X_c + DN J X_d \\
X_{a''} &= D^2 J X_b 
\end{align*}
\]

Eliminating \(X_b, X_c\) and \(X_d\) results in

\[
T(D, N, J) = \frac{X_{a''}}{X_{a'}} = \frac{D^7 N J^3}{1 - DN J - D^3 N J^2}
\]

(c) To find the free distance of the code we set \(N = J = 1\) in the transfer function, so that

\[
T_1(D) = T(D, N, J)|_{N=J=1} = \frac{D^7}{1 - D - D^3} = D^7 + D^8 + D^9 + \cdots
\]

Hence, \(d_{\text{free}} = 7\). The path, which is at a distance \(d_{\text{free}}\) from the all zero path, is the path \(X_a \rightarrow X_c \rightarrow X_b \rightarrow X_a\).

(d) The following figure shows 6 frames of the trellis diagram used by the Viterbi algorithm to decode the sequence \(\{111, 111, 111, 111, 111, 111\}\). The numbers on the nodes indicate the Hamming distance of the survivor paths from the received sequence. The branches that are dropped by the
The decoded sequence is \{111, 101, 011, 111, 101, 011\} which corresponds to the information sequence \{x_1, x_2, x_3, x_4\} = \{1, 0, 0, 1\} followed by two zeros.

Problem 8.6

(a) The state transition diagram and the flow diagram used to find the transfer function for this code are depicted in the next figure.

Thus,

\[
X_c = DNJ X_{d'} + D^2 NJ X_b \\
X_b = DJX_c + D^2 JX_d \\
X_d = NJX_c + DNX_{d'} \\
X_{a''} = DJX_b
\]
and by eliminating $X_b$, $X_c$ and $X_d$, we obtain

$$T(D, N, J) = \frac{X_{a''}}{X_{a'}} = \frac{D^3N J^3}{1 - DNJ - D^3NJ^2}$$

To find the transfer function of the code in the form $T(D, N)$, we set $J = 1$ in $T(D, N, J)$. Hence,

$$T(D, N) = \frac{D^3N}{1 - DN - D^3N}$$

(b) To find the free distance of the code we set $N = 1$ in the transfer function $T(D, N)$, so that

$$T_1(D) = T(D, N)|_{N=1} = \frac{D^3}{1 - D - D^3} = D^3 + D^4 + D^5 + 2D^6 + \cdots$$

Hence, $d_{\text{free}} = 3$

(c) An upper bound on the bit error probability, when hard decision decoding is used, is given by (see (8-2-34))

$$P_b \leq \frac{1}{k} \left. \frac{dT(D, N)}{dN} \right|_{N=1, D = \sqrt{4p(1-p)}}$$

Since

$$\left. \frac{dT(D, N)}{dN} \right|_{N=1} = \frac{d}{dN} \frac{D^3N}{1 - (D + D^3)N} \bigg|_{N=1} = \frac{D^3}{(1 - (D + D^3))^2}$$


with $k = 1$, $p = 10^{-6}$ we obtain

$$P_b \leq \frac{D^3}{(1 - (D + D^3))^2} \bigg|_{D = \sqrt{4p(1-p)}} = 8.0321 \times 10^{-9}$$

Problem 8.7

(a)

$$g_1 = [10], \quad g_2 = [11], \quad \text{states : (a) = [0], (b) = [1]}$$

The tree diagram, trellis diagram and state diagram are given in the following figures:
(b) Redrawing the state diagram:
\[ X_b = JN^2D^2X_a + JNDX_b \Rightarrow X_b = \frac{JN^2D^2}{1 - JND}X_a \]
\[ X_c = JDX_b \Rightarrow X_c = T(D, N, J) = \frac{J^2ND^3}{1 - JND} = J^2ND^3 + J^3N^2D^4 + ... \]

Hence :
\[ d_{\text{min}} = 3 \]

Problem 8.8

(a) \[ g_1 = [111], \ g_2 = [101], \text{ states } : (a) = [00], \ (b) = [01], \ (c) = [10], \ (d) = [11] \]

The tree diagram, trellis diagram and state diagram are given in the following figures :
(b) Redrawing the state diagram:
\[
\begin{align*}
X_c &= JND^2X_a + JNX_b \\
X_b &= JDX_c + JDX_d \\
X_d &= JNDX_d + JNDX_c = NX_b
\end{align*}
\]

\[ X_e = JD^2X_b \Rightarrow \frac{X_e}{X_a} = \frac{J^3ND^5}{1 - JND(1 + J)} = J^3ND^5 + J^4N^2D^6(1 + J) + \ldots \]

Hence:

\[ d_{\text{min}} = 5 \]

Problem 8.9

1. The state transition diagram is shown below

2. The transition diagram is shown below

and the Equations are

\[
\begin{align*}
X_c &= YZX_a + YZX_b \\
X_b &= X_c + Z^2X_d \\
X_d &= YZX_c + YZX_d \\
X_e &= Z^2X_b
\end{align*}
\]
Eliminating $X_b$, $X_c$, and $X_d$, we obtain

$$T(Y, Z) = \frac{X_e}{X_a} = \frac{YZ^3 - Y^2Z^4 + YZ^6}{1 - 2YZ + Y^2Z^2 - Y^2Z^4}$$

3. The code is not catastrophic since in the state transition diagram there are no loops of weight zero corresponding to inputs of nonzero weight.

4. From expansion of $T(Y, Z)$ we observe that the lowest power of $Z$ is 3, hence $d_{\text{free}} = 3$. This is also observed from the state transition diagram.

5. We use Equations 8.2-16–8.2-19 to find a bound on the bit error probability. First we note that $T(Y, Z) = YZ^3 + Y^2Z^4 + Y^3Z^5 + \ldots$, hence $a_3 = a_4 = a_5 = 1$ and

$$P_2(3) = \sum_{k=2}^{3} \binom{3}{k} p^k (1 - p)^{3-k} \approx 3p^2$$

$$P_2(4) = \sum_{k=3}^{4} \binom{4}{k} p^k (1 - p)^{4-k} + \frac{1}{2} \binom{4}{2} p^2 (1 - p)^2 \approx 3p^2$$

Hence

$$P_e \lesssim 6p^2 = 6 \times 10^{-6}$$

Problem 8.10

1. The state transition diagram is shown below

![State transition diagram](image-url)

2. Non catastrophic since in the state transition diagram there are no loops of weight zero corresponding to inputs of nonzero weight.

3. The transition diagram is shown below
and the Equations are
\[ X_c = YZ X_a + YX_b \]
\[ X_b = Z^2 X_c + ZX_d \]
\[ X_d = YZX_c + YZ^2 X_d \]
\[ X_e = ZX_b \]

Eliminating \(X_b, X_c,\) and \(X_d,\) we obtain
\[ T(Y, Z) = \frac{X_e}{X_a} = \frac{YZ^4 + Y^2 Z^4 - Y^2 Z^6}{1 - 2YZ^2 - Y^2 Z^2 + Y^2 Z^4} \]

4. Expanding \(T(Y, Z),\) we have
\[ T(Y, Z) = YZ^4 + Y^2 Z^4 + 3Y^3 Z^6 + Y^4 Z^6 + \ldots \]

Hence, \(d_{\text{free}} = 4.\)

5. We use Equations 8.2-16–8.2-19 to find a bound on the bit error probability. First we note that \(T(Z) = 2Z^4 + 4Z^6 + \ldots,\) hence \(a_4 = 2, a_6 = 3\) and
\[ P_2(4) = \sum_{k=3}^{4} \binom{4}{k} p^k (1-p)^{4-k} + \frac{1}{2} \binom{4}{2} p^2 (1-p)^2 \approx 3p^2 \]

Hence
\[ P_e \leq a_4 \times P_2(4) = 6 \times 10^{-6} \]

**Problem 8.11**

1. We have
\[ p(r|s) = p(n = r - s) \propto e^{-\sum_j |r_j - c_j|} \]
where \(\propto\) denotes proportionality. This shows that \(p(r|s)\) is maximized when \(\sum_j |r_j - c_j|\) is minimized, thus giving the optimal decoding rule.
2. Since soft decision decoding is employed, we use the bound given by Equation 8.2.15 with $\Delta$ given by Equation 8.2.10. From 8.2.10 we have

$$\Delta = \int_{-\infty}^{\infty} \sqrt{p(r|c = \sqrt{E_c})p(r|c = -\sqrt{E_c})} \, dr$$

$$= \int_{-\infty}^{\infty} \frac{1}{4} e^{-|r-\sqrt{E_c}|} e^{-|r+\sqrt{E_c}|} \, dr$$

$$= \int_{-\sqrt{E_c}}^{\sqrt{E_c}} \frac{1}{4} e^{-\sqrt{E_c}e^{r}} \, dr$$

$$\quad + \int_{-\sqrt{E_c}}^{\sqrt{E_c}} \frac{1}{4} e^{\sqrt{E_c}e^{-r}} \, dr$$

$$\quad + \int_{\sqrt{E_c}}^{\infty} \frac{1}{4} e^{-r+\sqrt{E_c}e^{-r}} \, dr$$

$$\quad + \int_{-\infty}^{-\sqrt{E_c}} \frac{1}{4} e^{r-\sqrt{E_c}e^{r}} \, dr$$

$$= e^{-\sqrt{E_c}} \left( 1 + \sqrt{E_c} \right)$$

The transfer function for this convolutional code was derived in the solution to Problem 8.10 and is given by

$$T(Y, Z) = T(Y, Z) = YZ^4 + Y^2Z^4 + 3Y^3Z^6 + Y^4Z^6 + \ldots$$

Therefore,

$$\frac{1}{k} \frac{\partial}{\partial Y} T(Y, Z) \bigg|_{Y=1} = 3Z^4 + 13Z^6 + \ldots$$

Substituting $Z = \Delta$, we obtain

$$P_b \leq 3\Delta^4 + 13\Delta^6 + \ldots$$

$$\approx 2.9415$$

when we put $E_c = 1$. This bound is obviously useless since at such a low SNR, the Bhattacharyya union bound is very loose.

3. Since the length of the received sequence is 12 and $n = 2$, we need a trellis of depth 6 which is shown below
Note that on the trellis each 0 corresponds to $-\sqrt{E_c} = -1$ and each 1 corresponds to $\sqrt{E_c} = 1$. The accumulated metrics are shown in red on each node. The # denotes a path that is not a survivor. From the trellis it is clear that the optimal path corresponds to the information sequence 100000. Since the last two zeros indicate the two zeros padded to the original information sequence we conclude that the transmitted sequence is 1000.

4. After hard decision decoding we have

$$y = \left(0, 0, 1, 1, 0, 0, 0, 1, 0, 1\right)$$

The trellis and the metrics for hard decision decoding are shown below

and the decoded sequence is either 110100 or 010100, thus the information sequence can be either 1101 or 0101.
5. For hard decision decoding we have

\[ p = P(r > 0 | c = -1) \]
\[ = \int_0^\infty \frac{1}{2} e^{-|r+1|} \]
\[ = \frac{1}{2} e^{-1} \approx 0.184 \]

and from Equation 8.2-14 we have \( \Delta = \sqrt{4p(1-p)} = 0.775 \). If we use this new \( \Delta \) in the error bound expression of part 3 we obtain

\[ P_b \leq 3\Delta^4 + 13\Delta^6 \approx 3.9 \]

This bound is also useless similar to the bound in part 3.

**Problem 8.12**

1. Here \( k = 2 \), \( n = 3 \), and \( K = 2 \), hence the number of states is \( 2^{k(K-1)} = 2^2 = 4 \).

2. This is similar to Example 8.1-5 and the resulting state diagram is shown below
and we have the following equations

\[ X_b = YZ^2 X_a + YZ^2 X_b + YZX_c + YZX_2 \]
\[ X_c = YZ^2 X_a + YZ^2 X_b + YZX_c + YZX_d \]
\[ X_d = Y^2 Z^2 X_a + Y^2 Z^2 X_b + Y^2 Z^3 X_c + Y^2 Z^3 X_d \]
\[ X_e = X_b + ZX_c + ZX_d \]

from which, after eliminating, we obtain

\[ T(Y,Z) = \frac{YZ^2 + YZ^3 + Y^2 Z^3 + Y^3 Z^3 - Y^3 Z^5}{1 - YZ - YZ^2 - Y^2 Z^3 - Y^3 Z^3 + Y^3 Z^5} \]

which can be expanded as

\[ T(Y,Z) = YZ^2 + YZ^3 + 2Y^2 Z^3 + Y^3 Z^3 + 2Y^2 Z^4 + 2Y^3 Z^4 + Y^4 Z^4 + Y^2 Z^5 + \ldots \]

The free distance of the code is the lowest power of \( Z \), i.e., 2.

3. There is only one path at free distance 2, this corresponds to the input sequence 0100, with output 011000.
4. The code is not catastrophic since there exists no loop of nonzero weight corresponding to an input sequence of weight zero.

5. For \( p = 10^{-4} \) we have \( \Delta = \sqrt{4p(1 - p)} \approx 0.02 \), and

\[
\left. \frac{1}{k} \frac{\partial}{\partial Y} T(Y, Z) \right|_{Y=1} = \frac{1}{2} Z^2 + 4Z^3 + 7Z^4 + \ldots
\]

resulting in

\( P_b \lessapprox 0.000233 \)

**Problem 8.13**

1. For this code we have

\[ g_1 = [1 \ 0 \ 0 \ 0] \quad g_2 = [0 \ 1 \ 0 \ 0] \quad g_3 = [1 \ 1 \ 1 \ 0] \]

Therefore,

\[
g_1^{(1)} = [1 \ 0] \quad g_2^{(1)} = [0 \ 0] \\
g_1^{(2)} = [0 \ 0] \quad g_2^{(2)} = [1 \ 0] \\
g_1^{(3)} = [1 \ 1] \quad g_2^{(3)} = [1 \ 0]
\]

resulting in

\[ G(D) = \begin{bmatrix} 1 & 0 & 1 + D \\ 0 & 1 & 1 \end{bmatrix} \]

2. From \( u \) we have \( u^{(1)} = (10110) \) and \( u^{(2)} = (01101) \), hence

\[
u^{(1)}(D) = 1 + D^2 + D^3 \\
u^{(2)}(D) = D + D^2 + D^4
\]

and

\[ u(D) = \begin{bmatrix} 1 + D^2 + D^3 & D + D^2 + D^4 \end{bmatrix} \]

and

\[ c(D) = u(D)G(D) = \begin{bmatrix} 1 + D^2 + D^3 & D + D^2 + D^4 & 1 \end{bmatrix} \]

Hence,

\[ c(D) = c^{(1)}(D^3) + Dc^{(2)}(D^3) + D^2c^{(3)}(D^3) = 1 + D^2 + D^4 + D^6 + D^7 + D^9 + D^{13} \]

and \( c = (101010110100010) \).
3. Working directly on the finite-state machine describing the code, we obtain the same sequence as in the previous part.

4. We need to use Equation 8.1-38, to obtain

\[ \text{GCD} \{1, 1, 1 + D\} = 1 = D^0 \]

Thus the code is not catastrophic.

**Problem 8.14**

1. The state transition diagram is shown below

![State Transition Diagram](attachment:state_diagram.png)

and the state diagram is

![State Diagram](attachment:state_diagram.png)

Solving the following equations

\[
\begin{align*}
X_b &= ZX_c + Z^2X_d \\
X_c &= YZX_a + YZ^2X_b \\
X_d &= YX_c + YZX_d \\
X_e &= ZX_b
\end{align*}
\]
results in
\[ T(Y, Z) = \frac{X_b}{X_a} = \frac{YZ^3}{1 - YZ - YZ^3} \]
or \[ T(Y, Z) = YZ^3 + Y^2Z^4 + Y^3Z^5 + (Y^2 + Y^4)Z^6 + (2Y^3 + Y^5)Z^7 + (3Y^4 + Y^6)Z^8 + \ldots \]
2. The code is not catastrophic since there are no loops of output weight equal to zero corresponding to non-zero input weights.

3. From the expression for \( T(Y, Z) \) we see that \( d_{\text{free}} = 3 \)

4. The crossover probability of the BSC is given by \( p = Q(\sqrt{2RC}) \) where \( RC = \frac{1}{2} \) and \( \gamma_b = E_b/N_0 = 10^{12.6/10} \approx 18.2 \). Hence \( p = Q(\sqrt{18.2}) \approx Q(4.27) \approx 10^{-5} \). From Equations 8.2-14 and 8.2-15 we obtain \( \Delta = \frac{1}{4} \times 10^{-5}(1 - 10^{-5}) \approx 0.00632 \) and
\[
\frac{\partial}{\partial Y} T(Y, Z) = \frac{YZ^3(Z + Z^3)}{(1 - YZ - YZ^3)^2} + \frac{Z^3}{1 - YZ - YZ^3}
\]
resulting in \( P_b \leq 2.55657 \times 10^{-7} \).

Problem 8.15

(a)
\[ g_1 = [23] = [10011], \ g_2 = [35] = [11101] \]

(b)
\[ g_1 = [25] = [10101], \ g_2 = [33] = [11011], \ g_3 = [37] = [11111] \]
Problem 8.16

For the encoder of Probl. 8.15(c), the state diagram is as follows:

\( g_1 = [17] = [1111], \ g_2 = [06] = [0110], \ g_3 = [15] = [1101] \)
The 2-bit input that forces the transition from one state to another is the 2-bits that characterize the terminal state.

**Problem 8.17**

The encoder is shown in Probl. 8.8. The channel is binary symmetric and the metric for Viterbi decoding is the Hamming distance. The trellis and the surviving paths are illustrated in the following figure:
Problem 8.18

In Probl. 8.8 we found :

\[ T(D, N, J) = \frac{J^3 ND^5}{1 - JND(1 + J)} \]

Setting \( J = 1 \):

\[ T(D, N) = \frac{ND^5}{1 - 2ND} \Rightarrow \frac{dT(D, N)}{dN} = \frac{D^5}{(1 - 2ND)^2} \]

For soft-decision decoding the bit-error probability can be upper-bounded by :

\[ P_{bs} \leq \frac{1}{2} \frac{dT(D, N)}{dN} \bigg|_{N=1, D=\exp(-\gamma_b R_c)} = \frac{1}{2} \frac{D^5}{(1 - 2ND)^2} \bigg|_{N=1, D=\exp(-\gamma_b/2)} = \frac{1}{2} \frac{\exp(-5\gamma_b/2)}{\left(1 - \exp(-\gamma_b/2)\right)^2} \]

For hard-decision decoding, the Chernoff bound is :

\[ P_{bh} \leq \frac{dT(D, N)}{dN} \bigg|_{N=1, D=\sqrt{4p(1-p)}} = \frac{\left[ \sqrt{4p(1-p)} \right]^{5/2}}{\left[ 1 - 2\sqrt{4p(1-p)} \right]^2} \]

where \( p = Q(\sqrt{\gamma_b R_c}) = Q(\sqrt{\gamma_b/2}) \) (assuming binary PSK). A comparative plot of the bit-error probabilities is given in the following figure :

![Bit-error probability plot](image)

Problem 8.19

\[ g_1 = [110], \ g_2 = [011], \text{ states:} \ (a) = [00], \ (b) = [01], \ (c) = [10], \ (d) = [11] \]
The state diagram is given in the following figure:

We note that this is a catastrophic code, since there is a zero-distance path from a non-zero state back to itself, and this path corresponds to input 1.

A simple example of an $K = 4$, rate $1/2$ encoder that exhibits error propagation is the following:

The state diagram for this code has a self-loop in the state $111$ with input $1$, and output $00$. A more subtle example of a catastrophic code is the following:

In this case there is a zero-distance path generated by the sequence $0110110110...$, which encompasses the states $011, 101$, and $110$. That is, if the encoder is in state $011$ and the input is $1$,
the output is 00 and the new state is 101. If the next bit is 1, the output is again 00 and the new state is 110. Then if the next bit is a zero, the output is again 00 and the new state is 011, which is the same state that we started with. Hence, we have a closed path in the state diagram which yields an output that is identical to the output of the all-zero path, but which results from the input sequence 110110110...

For an alternative method for identifying rate 1/n catastrophic codes based on observation of the code generators, please refer to the paper by Massey and Sain (1968).

**Problem 8.20**

There are 4 subsets corresponding to the four possible outputs from the rate 1/2 convolutional encoder. Each subset has eight signal points, one for each of the 3-tuples from the uncoded bits. If we denote the sets as A,B,C,D, the set partitioning is as follows:

```
G G G   G G G
G G G   G G G
G G G   G G G
G G G   G G G
```

The minimum distance between adjacent points in the same subset is doubled.

**Problem 8.21**
Over $P$ frames, the number of information bits that are being encoded is

$$k_P = P \sum_{j=1}^{J} N_j$$

The number of bits that are being transmitted is determined as follows: For a particular group of bits $j$, $j = 1, ..., J$, we may delete, with the corresponding puncturing matrix, $x_j$ out of $nP$ bits, on the average, where $x$ may take the values $x = 0, 1, ..., (n - 1)P - 1$. Remembering that each frame contains $N_j$ bits of the particular group, we arrive at the total average number of bits for each group

$$n(j) = N_j(nP - x_j) \Rightarrow n(j) = N_j(P + M_j), \quad M_j = 1, 2, ..., (n - 1)P$$

In the last group $j = J$ we should also add the $K - 1$ overhead information bits, that will add up another $(K - 1)(P + M_j)$ transmitted bits to the total average number of bits for the $J^{th}$ group.

Hence, the total number of bits transmitted over $P$ frames be

$$n_P = (K - 1)(P + M_j) + \sum_{j=1}^{J} JN_j(P + M_j)$$

and the average effective rate of this scheme will be

$$R_{av} = \frac{k_P}{n_P} = \frac{\sum_{j=1}^{J} N_j P}{\sum_{j=1}^{J} JN_j(P + M_j) + (K - 1)(P + M_j)}$$

**Problem 8.22**

By definition, from Equation 8.8-26, we have

$$\max^* \{x, y\} = \ln (e^x + e^y)$$

$$= \begin{cases} 
\ln [e^x (1 + e^{y-x})], & x > y \\
\ln [e^y (1 + e^{x-y})], & y > x 
\end{cases}$$

$$= \begin{cases} 
x + \ln (1 + e^{y-x}), & x > y \\
y + \ln (1 + e^{x-y}), & y > x 
\end{cases}$$

$$= \max \{x, y\} + \ln \left(1 + e^{|x-y|}\right)$$

To prove the second relation we note that $\max^* \{x, y\} = \ln (x^x + e^y)$, or $e^{\max^*\{x,y\}} = e^x + e^y$. Therefore

$$\max^* \{x, y, z\} = \ln (e^x + e^y + e^z)$$

$$= \ln \left(e^{\max^*\{x,y\}} + e^z\right)$$

$$= \max^* \{\max^* \{x, y\}, z\}$$
Problem 8.23

The encoder and the state transition diagram are shown below

The trellis diagram for the received sequence (i.e., a terminated trellis of depth 4) is shown below
where $0/-1,-1$ on a trellis branch indicates that this branch corresponds to an information bit $u_i = 0$ and encoded bits $0,0$ that are modulated to $-\sqrt{E_c}, -\sqrt{E_c} = -1,-1$.

1. To use the BCJR algorithm we first use Equation 8.8-20 to compute $\gamma$’s, then use the recursive relations 8.8-11 and 8.8-14 to compute $\alpha$’s and $\beta$’s and finally use Equation 8.8-17 to find the likelihood values. In using these equations we assume equal probability for input sequence, i.e., $P(u_i = 0) = P(u_i = 1) = 1/2$, and put $N_0 = 4$ and $E_c = 1$. Using these relations we obtain the following values (note that states a, b, c, and d are represented by numbers 1 to 4)

<table>
<thead>
<tr>
<th>$\gamma_1(1,1)$</th>
<th>$\alpha_1(1)$</th>
<th>$\beta_1(1)$</th>
<th>$\beta_3(1)$</th>
<th>$\beta_3(2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0181939</td>
<td>0.0181939</td>
<td>0.0250553</td>
<td>0.0250553</td>
<td>0.0338211</td>
</tr>
<tr>
<td>$\gamma_1(1,3)$</td>
<td>$\alpha_1(3)$</td>
<td>$\beta_1(3)$</td>
<td>$\beta_2(2)$</td>
<td>$\beta_3(2)$</td>
</tr>
<tr>
<td>0.0299966</td>
<td>0.0299966</td>
<td>0.000480464</td>
<td>0.000480464</td>
<td>0.0338211</td>
</tr>
<tr>
<td>$\gamma_2(1,1)$</td>
<td>$\alpha_2(1)$</td>
<td>$\beta_2(1)$</td>
<td>$\beta_2(2)$</td>
<td>$\beta_3(2)$</td>
</tr>
<tr>
<td>0.0144918</td>
<td>0.000263662</td>
<td>0.000434705</td>
<td>0.00015952</td>
<td>0.0338211</td>
</tr>
<tr>
<td>$\gamma_2(1,3)$</td>
<td>$\alpha_2(2)$</td>
<td>$\beta_2(3)$</td>
<td>$\beta_2(2)$</td>
<td>$\beta_3(2)$</td>
</tr>
<tr>
<td>0.0118649</td>
<td>0.0002415869</td>
<td>0.000215869</td>
<td>0.000215869</td>
<td>0.0338211</td>
</tr>
<tr>
<td>$\gamma_2(3,2)$</td>
<td>$\alpha_2(3)$</td>
<td>$\beta_2(4)$</td>
<td>$\beta_2(2)$</td>
<td>$\beta_3(2)$</td>
</tr>
<tr>
<td>0.0144918</td>
<td>0.000215869</td>
<td>0.000215869</td>
<td>0.000215869</td>
<td>0.0338211</td>
</tr>
<tr>
<td>$\gamma_2(3,4)$</td>
<td>$\alpha_2(4)$</td>
<td>$\beta_2(4)$</td>
<td>$\beta_2(2)$</td>
<td>$\beta_3(2)$</td>
</tr>
<tr>
<td>0.0118649</td>
<td>0.000355907</td>
<td>0.000355907</td>
<td>0.000355907</td>
<td>0.0338211</td>
</tr>
<tr>
<td>$\gamma_3(1,1)$</td>
<td>$\alpha_3(1)$</td>
<td>$\beta_1(1)$</td>
<td>$\beta_1(1)$</td>
<td>$\beta_1(1)$</td>
</tr>
<tr>
<td>0.00191762</td>
<td>3.28818 $\times 10^{-6}$</td>
<td>1.46579 $\times 10^{-6}$</td>
<td>1.46579 $\times 10^{-6}$</td>
<td>1.46579 $\times 10^{-6}$</td>
</tr>
<tr>
<td>$\gamma_3(2,1)$</td>
<td>$\alpha_3(2)$</td>
<td>$\beta_1(3)$</td>
<td>$\beta_1(3)$</td>
<td>$\beta_1(3)$</td>
</tr>
<tr>
<td>0.00636672</td>
<td>2.6799 $\times 10^{-6}$</td>
<td>4.86659 $\times 10^{-6}$</td>
<td>4.86659 $\times 10^{-6}$</td>
<td>4.86659 $\times 10^{-6}$</td>
</tr>
<tr>
<td>$\gamma_3(3,2)$</td>
<td>$\alpha_4(1)$</td>
<td>$\beta_0(1)$</td>
<td>$\beta_0(1)$</td>
<td>$\beta_0(1)$</td>
</tr>
<tr>
<td>0.00191762</td>
<td>1.73024 $\times 10^{-7}$</td>
<td>1.72649 $\times 10^{-7}$</td>
<td>1.72649 $\times 10^{-7}$</td>
<td>1.72649 $\times 10^{-7}$</td>
</tr>
<tr>
<td>$\gamma_3(4,2)$</td>
<td>$\alpha_4(2)$</td>
<td>$\beta_0(1)$</td>
<td>$\beta_0(1)$</td>
<td>$\beta_0(1)$</td>
</tr>
<tr>
<td>0.00636672</td>
<td>1.73024 $\times 10^{-7}$</td>
<td>1.72649 $\times 10^{-7}$</td>
<td>1.72649 $\times 10^{-7}$</td>
<td>1.72649 $\times 10^{-7}$</td>
</tr>
<tr>
<td>$\gamma_4(1,1)$</td>
<td>$\alpha_4(1)$</td>
<td>$\beta_4(1)$</td>
<td>$\beta_4(1)$</td>
<td>$\beta_4(1)$</td>
</tr>
<tr>
<td>0.0250553</td>
<td>0.0250553</td>
<td>0.0250553</td>
<td>0.0250553</td>
<td>0.0250553</td>
</tr>
<tr>
<td>$\gamma_4(2,1)$</td>
<td>$\alpha_4(2)$</td>
<td>$\beta_4(1)$</td>
<td>$\beta_4(1)$</td>
<td>$\beta_4(1)$</td>
</tr>
<tr>
<td>0.0338211</td>
<td>0.0338211</td>
<td>0.0338211</td>
<td>0.0338211</td>
<td>0.0338211</td>
</tr>
</tbody>
</table>
Using these values in Equation 8.8-17, we obtain

\[ L(u_1) \approx 1.7 > 0 \Rightarrow \hat{u}_1 = 1 \]
\[ L(u_2) \approx 0.1 > 0 \Rightarrow \hat{u}_2 = 1 \]
\[ L(u_3) \approx 1.7 > 0 \Rightarrow \hat{u}_3 = 1 \]
\[ L(u_4) \approx 0.095 > 0 \Rightarrow \hat{u}_4 = 1 \]

2. For Viterbi algorithm we have to minimize the Euclidean distance, the trellis with accumulated metrics (in red) is shown below

![Trellis Diagram](image)

and hence the Viterbi algorithm detects \( \hat{u} = (1, 1, 1, 1) \).

**Problem 8.24**

The encoder and the state transition diagram are shown below
The trellis diagram for the received sequence (i.e., a terminated trellis of depth 4) is shown below

where 0/−1, −1 on a trellis branch indicates that this branch corresponds to an information bit \( u_i = 0 \) and encoded bits 0,0 that are modulated to \(-\sqrt{E_c}, -\sqrt{E_c} = -1, -1\).

To use the Max-Log-APP algorithm we first use Equation 8.8-19 to compute \( \tilde{\gamma} \)'s, in computing
these values we drop the term corresponding to the constant coefficient and define

\[ \tilde{\gamma}_i = -\frac{||y_i - c_i||^2}{N_0} \]

We then use the recursive relations in 8.8-27 to compute \( \tilde{\alpha} \)'s and \( \tilde{\beta} \)'s and finally use Equation 8.8-28 to find the likelihood values. In using these equations we assume equal probability for input sequence, i.e., \( P(u_i = 0) = P(u_i = 1) = 1/2 \), and put \( N_0 = 4 \) and \( \mathcal{E}_c = 1 \). Using these relations we obtain the following values (note that states a, b, c, and d are represented by numbers 1 to 4)

| \( \tilde{\gamma}_1(1, 1) = -0.7825 \) | \( \tilde{\alpha}_1(1) = -0.7825 \) | \( \tilde{\beta}_3(1) = -0.4625 \) |
| \( \tilde{\gamma}_1(1, 3) = -0.2825 \) | \( \tilde{\alpha}_1(3) = -0.2825 \) | \( \tilde{\beta}_3(2) = -0.1625 \) |
| \( \tilde{\gamma}_2(1, 1) = -1.01 \) | \( \tilde{\alpha}_2(1) = -1.7925 \) | \( \tilde{\beta}_2(1) = -3.495 \) |
| \( \tilde{\gamma}_2(1, 3) = -1.21 \) | \( \tilde{\alpha}_2(2) = -1.4925 \) | \( \tilde{\beta}_2(2) = -2.295 \) |
| \( \tilde{\gamma}_2(3, 2) = -1.21 \) | \( \tilde{\alpha}_2(3) = -1.9925 \) | \( \tilde{\beta}_2(3) = -3.195 \) |
| \( \tilde{\gamma}_2(3, 4) = -1.01 \) | \( \tilde{\alpha}_2(4) = -1.2925 \) | \( \tilde{\beta}_2(4) = -1.995 \) |
| \( \tilde{\gamma}_3(1, 1) = -3.0325 \) | \( \tilde{\alpha}_3(1) = -3.325 \times 10^{-6} \) | \( \tilde{\beta}_1(1) = -3.405 \) |
| \( \tilde{\gamma}_3(2, 1) = -1.8325 \) | \( \tilde{\alpha}_3(2) = -3.125 \times 10^{-6} \) | \( \tilde{\beta}_1(3) = -3.005 \) |
| \( \tilde{\gamma}_3(3, 2) = -3.0325 \) | \( \tilde{\alpha}_3(4) = -3.2875 \times 10^{-7} \) | \( \tilde{\beta}_0(1) = -3.383 \) |
| \( \tilde{\gamma}_3(4, 2) = -1.8325 \) | \( \tilde{\alpha}_4(1) = -3.2875 \times 10^{-7} \) | \( \tilde{\beta}_0(3) = -3.283 \) |
| \( \tilde{\gamma}_4(1, 1) = -0.4625 \) | \( \tilde{\alpha}_4(2) = -1.8325 \) | \( \tilde{\beta}_0(4) = -0.4625 \) |
| \( \tilde{\gamma}_4(2, 1) = -0.1625 \) | \( \tilde{\alpha}_4(3) = -0.1625 \) | \( \tilde{\beta}_0(5) = -0.1625 \) |

Using these values in Equation 8.8-17, we obtain

\[
L(u_1) \approx 0.9 > 0 \Rightarrow \hat{u}_1 = 1 \\
L(u_2) \approx 0 \Rightarrow \hat{u}_2 = 0 \text{ or } 1 \\
L(u_3) \approx 1.9 > 0 \Rightarrow \hat{u}_3 = 1 \\
L(u_4) \approx 0.5 > 0 \Rightarrow \hat{u}_4 = 1
\]

**Problem 8.25**

1. Let us define new random variables \( W_i = 1 - 2X_i \), obviously \( W_i \)'s are independent and each \( W_i \) takes values of +1 and -1 with probabilities \( p(0) \) and \( p(1) \), respectively. This means that \( \mathbb{E}[W_i] = p_i(0) - p_i(1) \). We also define

\[
T = \prod_{i=1}^{n} W_i
\]
$T$ is equal to $+1$ or $-1$ if an even or odd number of $W_i$’s are equal to $-1$, or equivalently if an even or odd number of $X_i$’s are equal to $1$. Similarly $Y$ is equal to $0$ or $1$ if an even or odd number of $X_i$’s are equal to $1$. This shows that $T = 1$ if and only if $Y = 0$ and $T = -1$ if and only if $Y = 1$; i.e., $P[T = 1] = P[Y = 0] = p(0)$ and $P[T = -1] = P[Y = 1] = p(1)$. Therefore, $E[T] = P(T = 1) - P(T = -1) = p(0) - p(1)$ and by independence of $W_i$’s we have

$$E[T] = \prod_{i=1}^{n} E[W_i] = \prod_{i=1}^{n} (p_i(0) - p_i(1)) = p(0) - p(1)$$

2. From part 1 and $p(0) + p(1) = 1$ we obtain

$$p(0) = \frac{1}{2} + \frac{1}{2} \prod_{i=1}^{n} (p_i(0) - p_i(1))$$

$$p(1) = \frac{1}{2} - \frac{1}{2} \prod_{i=1}^{n} (p_i(0) - p_i(1))$$

3. Note that

$$p_j(0) - p_j(1) = 1 - 2p_j(1) \quad (*)$$

at a check node the messages are binary and the check node function is true if the binary sum of messages is equal to zero. Therefore from part 2 and relation (*) we obtain Equation 8.10-27.

**Problem 8.26**

For an equality constraint node (cloning node) that generates replicas of $x_i$ we have

$$g(x_1, x_2, \ldots, x_n) = \prod_{j \neq i} \delta[x_j = x_i]$$

Using Equation 8.10-22 we have

$$\mu_{g \rightarrow x_i}(x_i) = \sum_{\sim x_i} \prod_{j \neq i} \delta[x_j = x_i] \prod_{j \neq i} \mu_{x_j \rightarrow g}(x_j)$$

$$= \prod_{j \neq i} \mu_{x_j \rightarrow g}(x_i)$$

where we have used the relation

$$\sum_{x_j} \delta[x_j = x_i] \mu_{x_j \rightarrow g}(x_j) = \mu_{x_j \rightarrow g}(x_i)$$
Problem 8.27

From $H$ we have the following parity check equations

\begin{align*}
  c_3 + c_6 + c_7 + c_8 &= 0 \\
  c_1 + c_2 + c_5 + c_{12} &= 0 \\
  c_4 + c_9 + c_{10} + c_{11} &= 0 \\
  c_2 + c_6 + c_7 + c_{10} &= 0 \\
  c_1 + c_3 + c_8 + c_{11} &= 0 \\
  c_4 + c_5 + c_9 + c_{12} &= 0 \\
  c_1 + c_4 + c_5 + c_7 &= 0 \\
  c_6 + c_8 + c_{11} + c_{12} &= 0 \\
  c_2 + c_3 + c_9 + c_{10} &= 0
\end{align*}

resulting in the Tanner graph shown below
Problem 8.28
For a repetition code $G = [1 1 \ldots 1]$ and hence

$$H = \begin{bmatrix}
1 & 1 & 0 & 0 & \cdots & 0 \\
1 & 0 & 1 & 0 & \cdots & 0 \\
1 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & & & & \ddots & 0 \\
1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

which clearly corresponds to an irregular LDPC matrix.

**Problem 8.29**

We have

$$H = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

and the parity check equations are

$$c_1 + c_2 = 0$$
$$c_1 + c_3 = 0$$
$$c_1 + c_4 = 0$$
$$c_1 + c_5 = 0$$
$$c_1 + c_6 = 0$$

Two Tanner graphs for this code are shown below.