University of California, San Diego

Electrical & Computer Engineering Department
ECE 259BN - Winter Quarter 2008
Trellis-Coded Modulation

Solutions to Problem Set 4, Part 2

Please send comments to psiegel@ucsd.edu.

Problem 1

(a)

Diagram	**Encoder**
X	$G_2(D)$
Y	$G_0(D)$
Z	$G_1(D)$

(b)

(i)

$$S_1(I, D) = T_1(ID, D)$$

(ii)

$$S_1(I, D) = \frac{D^5 I^2 - D^6 I^2 + D^6 I^4}{1 - 2D + D^2 - D^2 I^2}$$

(c)

(i)

$$\begin{aligned} T_2(I, D_p) &= T_1(D_p, I) \\ S_2(I, D) &= T_1(D, ID) \end{aligned}$$

Proof: $G_1(D)$ generates the same code, C, as $G_2(D)$. $G_2(D)$ generates the code obtained from the code C by switching the order of digits in the length-2 codewords of C. Since $G_1(D)$ and $G_2(D)$ are both systematic encoders, this means that the information bit in words produced by $G_1(D)$ is the parity bit in words produced by $G_2(D)$, and vice-versa. So, I and D_p reverse roles. ∎

(ii)

$$T_2(I, D_p) = T_1(D_p, I)$$
$$S_2(I, D) = T_2(ID, D) = T_1(D, ID)$$

Using the identity:

$$S_1(UV^{-1}, V) = T_1(U, V),$$

we get

$$S_2(I, D) = S_1(I^{-1}, ID).$$

(d)

(i) For $G_1(D)$, the input $x(D) = 1 + D^2$ produces the codeword $c(D) = [1 + D^2, 1 + D + D^2]$. This codeword has the minimum Hamming weight of any codeword produced by a weight-2 input, so this encoder has effective free distance $d_{free}^{eff} = 5$.
(i) For $G_2(D)$, the input $x(D) = 1 + D^3$ produces the codeword $c(D) = [1 + D^3, 1 + D + D^2 + D^3]$. This codeword has the minimum Hamming weight of any codeword produced by a weight-2 input, so this encoder has effective free distance $d_{free}^{eff} = 6$.

(e)

$G_2(D)$ is preferred because of its larger effective free distance, $d_{free}^{eff} = 6$.

Problem 2

(a)

$$A_{w,j} = \begin{cases} \binom{n-1}{w} & , \quad \text{for } w \text{ even}, \ j = 0 \\ \binom{n-1}{w} & , \quad \text{for } w \text{ odd}, \ j = 1 \\ 0 & , \quad \text{otherwise.} \end{cases}$$

So, the IRWEF $A(W, Z_p)$ is:

$$A(W, Z_p) = \sum_{w=0}^{n-1} \binom{n-1}{w} W^w Z_p^{(w \bmod 2)}.$$

(b)

$$B_{w,z} = \begin{cases} \binom{n-1}{w} & , \quad \text{for } w \text{ even}, \ z = w \\ \binom{n-1}{w} & , \quad \text{for } w \text{ odd}, \ z = w + 1. \\ 0 & , \quad \text{otherwise.} \end{cases}$$

The IOWEF $B(W, Z)$ can be determined from the expression above, or by noting that

$$B(W, Z) = A(WZ, Z).$$

However you choose to derive it, $B(W,Z)$ is:

$$B(W,Z) = \sum_{w=0}^{n-1} \binom{n-1}{w} W^w Z^{w+(w \bmod 2)}.$$

(c)

Note that the parity bits produced by the two encoders are equal, regardless of what sort of interleaver is used. So, the output parity weight is 0 if the input weight w is even, and it is 2 if the input weight w is odd.

Now, recall the expression for the average CWEF when a uniform interleaver is used with systematic encoders in a parallel turbo architecture:

$$A_w^C(Z_p) = \frac{A_w^{C_1}(Z_p) A_w^{C_2}(Z_p)}{\binom{n-1}{w}}.$$

Therefore, assuming a uniform interleaver, the average IRWEF of the resulting turbo encoder is given by:

$$A_{w,j}^C = \begin{cases} \frac{\binom{n-1}{w}\binom{n-1}{w}}{\binom{n-1}{w}} & , \quad \text{for } w \text{ even}, \ j=0 \\ \frac{\binom{n-1}{w}\binom{n-1}{w}}{\binom{n-1}{w}} & , \quad \text{for } w \text{ odd}, \ j=2 \\ 0 & , \quad \text{otherwise} \end{cases}$$

which simplifies to

$$A_{w,j}^C = \begin{cases} \binom{n-1}{w} & , \quad \text{for } w \text{ even}, \ j=0 \\ \binom{n-1}{w} & , \quad \text{for } w \text{ odd}, \ j=2 \\ 0 & , \quad \text{otherwise.} \end{cases}$$

The average IRWEF is therefore given by:

$$A^C(W,Z_p) = \sum_{w=0}^{n-1} \binom{n-1}{w} W^w Z_p^{2(w \bmod 2)}.$$

(d)

From part (c), the IOWEF $B_{w,z}^C$ is given by:

$$B_{w,z}^C = \begin{cases} \binom{n-1}{w} & , \quad \text{for } w \text{ even}, \ z=w \\ \binom{n-1}{w} & , \quad \text{for } w \text{ odd}, \ z=w+2. \\ 0 & , \quad \text{otherwise.} \end{cases}$$

So,

$$B^C(W,Z) = \sum_{w=0}^{n-1} \binom{n-1}{w} W^w Z^{w+2(w \ mod \ 2)}.$$

So, the smallest non-zero codeword weight is $z = 2$.

(e)

This is not a good turbo code architecture. The IOWEF of the resulting turbo code is independent of the interleaver, the constituent encoders are not recursive, and they have very low minimum codeword weight.