ECE 695C, Homework #4, due date: 2/23/2012

https://engineering.purdue.edu/∼chihw/12ECE695C/12ECE695C.html

Question 1: Consider the convolutional code described in the end of this homework. [Lin Constello, "Error Control Coding, 2nd Ed." FIGURE 11.7(b)].

1. What is the rate of the convolutional code?
2. If an user wants to transmit 4 information bits 0110 using this code, find out the final codeword.
3. Draw the trellis representation of the convolutional code, say the trellis structure from the t-th to the $(t+1)$-th stage. (Draw your trellis structure for general t values even though in this example, we only have $t = 1$ or 2.)

Question 2: Continue from the previous question. Suppose we are transmitting over an i.i.d. Laplacian channel

$$P_{Y_i|X_i}(y|x) \sim \frac{1}{2}e^{-|y-(-1)^x|}$$

and the observation $\overrightarrow{y}_{\text{obs}} = (0.3, 0.2, -0.5, 0.4, -0.7, 0.3)$.

- Find out the most likely codeword by the Viterbi algorithm.
- Find out the minimum-mean-square-error (MMSE) codeword by the Viterbi algorithm. That is, we are interested in a codeword $x_1x_2\cdots x_6$ that minimizes $\sum_{i=1}^{6}(y_{\text{obs},i} - (-1)^{x_i})^2$.
- Find out the ML value of the 4-th information bit by the BCJR decoder.

Question 3: [Optional but highly recommended] Implement the Viterbi and BCJR decoders by MATLAB that take arbitrary observation $\overrightarrow{y}_{\text{obs}}$ as input. Use Monte-Carlo simulation to plot the curve of the frame-error rate and bit-error rate versus different values of γ, where the channel is

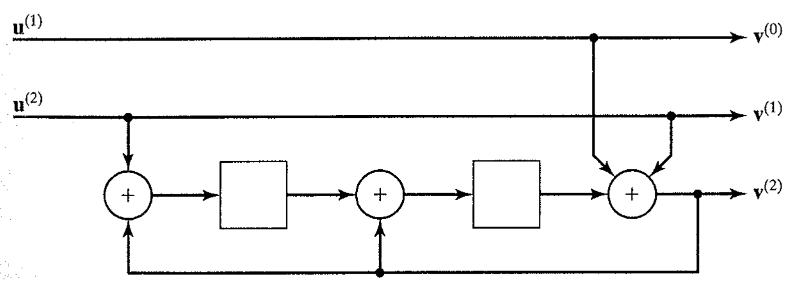
$$P_{Y_i|X_i}(y|x) \sim \frac{1}{\sqrt{2\pi}}e^{-\frac{(y-\sqrt{\gamma}(-1)^x)^2}{2}}.$$

That is, you first choose randomly a 4-bit string u_1 to u_4 (out of totally 16 different choices). Then generate the corresponding codeword x_1 to x_6. After that, generate a random realization of y_1 to y_6 based on the conditional distribution. Finally, run the VA/BCJR decoder and record the output of the decoder. Repeat this process for 10^4 and compute the average frame error rates and the bit error rates. Then use a different γ value and restart the entire process.

Compare the performance of the FER and BER when we use the same code in Questions 1 and 2 but now encode 80 information bits instead of 4. Is the performance of the longer convolutional code better than that of the shorter code?

What is the γ^* for which the Shannon capacity is $\frac{2}{3}$ bits per symbol usage. How far away from the capacity is the performance of this convolutional code (when focusing on bit-error-rate being 10^{-3})?

(a)



(b)

FIGURE 11.7: (a) A (3, 2, 4) systematic feedback encoder in controller canonical form and (b) an equivalent (3, 2, 2) systematic feedback encoder in observer canonical form.

require 64 states.) To convert $\mathbf{G}(D)$ to an equivalent systematic feedback encoder, we apply the following sequence of elementary row operations:

Step 1. Row 2 $\Rightarrow [1/(1 + D + D^2 + D^3)]$ [Row 2].
Step 2. Row 1 $\Rightarrow$ Row 1 + [D][Row 2].

The result is the modified generator matrix (see Problem 11.7)

$$\mathbf{G}'(D) = \begin{bmatrix} 1 & 0 & (1 + D + D^2)/(1 + D + D^2 + D^3) \\ 0 & 1 & (1 + D + D^3)/(1 + D + D^2 + D^3) \end{bmatrix} \tag{11.71}$$