UCSD ECE154C Handout #18
Prof. Young-Han Kim Thursday, May 30, 2017

Programming Assignment (Extra Credit)
Due: Tuesday, June 8, 2017, by 11:59p

You can use any programming language you prefer (MATLAB, Python, C/C++, or Julia, for example). Write down your code as clearly as possible and add suitable comments. For the submission, please follow the instruction below.

- Summarize the answers concisely in a document of any extension (e.g. hw6-ans.doc, hw6-ans.pdf). If you cannot get an answer because your code does not run, please comment your progress in the answer file.
- Please zip your code and the answer file in one file with the exact name "hw6-Last name-First name.zip".
- Submit your zip file to `ece154ucsd@gmail.com` with the exact subject `ECE 154C (HW6)`.

1. In this homework, we examine the properties of binary convolutional codes and implement Viterbi decoders. Note: in the following, we always assume the input is a single sequence of bits.

   (a) Write a program for a function `convEncode`(`connections`, `inputbits`) that takes an array of octal tap connections `connections` and a sequence of bits `inputbits` as inputs and outputs the encoded bit sequence. For example, for `connections` = [4 5] and `inputbits` = 1001010001110, your function should output the bit sequence

   $$11000111001000010011111001.$$

   Use this function to encode the bit sequence 1111111111111111111111111111 using

   i. the convolutional code with generator matrix $G(D) = [1 + D^2 \quad 1 + D + D^2]$, i.e., the code with octal tap connections [5 7].

   ii. the convolutional code with generator matrix $G(D) = [1 \quad 1 + D]$, i.e., the code with octal tap connections [1 3].

   (b) Write a program for a function `freeDistance`(`connections`) that takes an array of octal tap connections `connections` as input and outputs the free distance d_{free} of the code. For example, for `connections` = [4 5], your function should output

   $$d_{\text{free}} = 3.$$

   Use this function to find d_{free} for the two codes in part (a), as well as for the code with octal tap connections [155 117].

(c) Write a program for a function `ViterbiBSC(connections, receivedsequence)` that takes an array of octal tap connections `connections` of a convolutional code, as well as the received sequence when the code is used over a BSC with $p < 1/2$, and outputs the most likely input sequence using Viterbi decoding. Verify that this function works as expected for the code with connections $[5 \quad 7]$. You should initialize your decoder at the all-zero state.

(d) Use the functions in parts (a) and (c) to plot the bit error rate (BER) performance of the codes $[1 \quad 3]$ and $[5 \quad 7]$ over a BSC(p) with $p \in [0\colon 0.05\colon 0.5]$. For each value of p, you should encode and decode a block of at least 2,000 symbols and calculate the average bit error rate. Please make sure that the encoder always start from the all-zero state. (No termination is required.)

(e) Write a program for a function `ViterbiAWGN(connections, receivedsequence)` that takes an array of octal tap connections `connections` of a convolutional code, as well as the received sequence when the code is used over an AWGN, and outputs the most likely input sequence using Viterbi decoding. Here, we assume that BPSK with unit power is used for transmission, i.e., the symbol 0 is mapped to -1 and 1 is mapped to 1 while transmitting over the channel. Verify that this function works as expected for the code with connections $[5 \quad 7]$. You should initialize your decoder at the all-zero state.

(f) Use the functions in parts (a) and (e) to plot the bit error rate (BER) performance of the codes $[1 \quad 3]$ and $[5 \quad 7]$ over an AWGN with SNR ranging from -1 dB to 10 dB. For each value of the SNR, you should encode and decode a block of at least 2,000 symbols and calculate the average bit error rate. Please make sure that the encoder always start from the all-zero state. (No termination is required.)