Hand-in assignement for chapter 8

May 7, 2014

1 Overview and description of provided software

This problem is matlab based and investigates the performance of combining convolutional codes with PAM alphabets. The convolutional code to be studied is the (7,5) code which has the rate $R = 1/2$. Matlab's communication toolbox contains a package for encoding and decoding of convolutional codes, but we will make use of my own old software. The reason for this is that in order to make use of soft information in Matlab, one must understand the theory of convolutional codes a bit deeper than what was taught in class.

We first discuss the software I provide. There are two routines:

- **v = encoder_75(msg)**. This routine takes a row vector of information bits 'msg', and encode them by a (7,5) convolutional code. The routine also pads two 0's at the end of 'msg' in order to terminate the codeword to the all-zero state. Thus, with N bits in 'msg', the row vector 'v' contains 2N+4 bits. (The 'bits' are of course 0/1, not ± 1).

- **dec_bits = decoder_75(Lext)**. This is where it gets a bit tricky.... This routine is a decoder for the received codeword and attempts to re-create the sequence 'msg'. (For those of you who have a bit more experience, this decoder is a symbol-by-symbol decoder, and not a sequnce decoder. It is implemented through the BCJR algorithm.) The simplest possible setting would be if the input to the routine where just a bit sequence 'r', where 'r' was the coderword 'v', but where some of the bits were flipped (i.e., some errors were introduced). *This is not what the routine is assuming, as it is more advanced than that.* Instead, the sequence 'Lext', which is of length 2N+4, contains the probabilities

$$\log \frac{\Pr(v_k = 1)}{\Pr(v_k = 0)},$$

  where k is the kth element in the sequence 'Lext'. These are the so-called log-likelihood-ratios (LLR). Once the LLRs are obtained, one can just plug them into the routine, and out comes the optimal decision of the information bits. The output format is 0/1.

I will get back shortly with a discussion of how to generate the LLRs.

2 Problem formulation

Consider a discrete time model

$$y_k = x_k + n_k$$

where x_k is the channel input and n_k is white Gaussian noise stemming from a passband noise process with spectral density $N_0/2$ W/Hz. We shall consider two options for the modulation: 2PAM and 4PAM. Choose bit-mappings as you like. Now, the bits carried by the symbols $\{x_k\}$ are not information bits, but are encoded by the (7,5) convolutional code, followed by an interleaver. The interleaver is in matlab implemented by the command

scrambled_data = randintrlv(data,seed).m

and scrambles the bits prior to mapping to the PAM constellation. The inverse of this operation, the de-interleaver, is

data = randdeintrlv(scrambled_data,the-same-seed).m

We shall consider two types of detection: hard detection and soft detection.

2.1 Hard detection

Let us discuss 4PAM as 2PAM is simpler. Each observed sample y_k contains 2 code bits, let us denote these as $v_k[1]$ and $v_k[2]$, respectively. With hard detection, we look at y_k and do an ML detection of the two bits,

$$(\hat{v}_k[1], \hat{v}_k[2]) = \arg\max \Pr(y_k | v_k[1], v_k[2]).$$

This maximization is simple to do as every pair of two code-bits corresponds to one 4PAM symbol, and then the probability is essentially the Euclidean distance from that symbol to y_k. These bit-decisions should now be plugged into the routine 'decoder_75', but this routine does not accept bit decisions as it is expecting LLRs. However, to say that the bit $v_k[1] = 1$ is really the same thing as saying that the probability $\Pr(v_k[1] = 1) = 1$, and therefore

$$\log \frac{\Pr(v_k = 1)}{\Pr(v_k = 0)} = \infty.$$

Hence, if we have decoded a particular bit to be 1, we just set the LLR to be infinity, and if the decoded bit is 0, the LLR is set to minus infinity. For the routine I provided, the LLRs must be finite, otherwise it will crash, therefore the LLRs can be set to some large, but finite value; 10 is a good choice.

2.2 Soft detection

In this case, we aim at producing actual LLRs, and not only truncated values to ± 10.

The probability that the bit $v_k[1]$ is 0 given the observed sample y_k is

$$\begin{aligned}
\Pr(v_k[1]=0|y_k) &= \sum_{x_k:\text{bit 1 of x}_\text{k}\text{ is 0}} \Pr(x_k|y_k) \\
&= \sum_{x_k:\text{bit 1 of x}_\text{k}\text{ is 0}} \Pr(y_k|x_k)\frac{\Pr(x_k)}{\Pr(y_k)} \\
&= \sum_{x_k:\text{bit 1 of x}_\text{k}\text{ is 0}} \kappa \exp\left(-\frac{|y_k-x_k|^2}{2\sigma^2}\right) \qquad (1)
\end{aligned}$$

where κ is a constant which will show up to be irrelevant, and σ^2 is the variance of the real-part of the noise sample n_k. For the probability that the bit $v_k[1]$ is 1, we of course have a similar equation, and the important thing to note is that the constant κ will be the same. Therefore, we can form the LLRs as

$$\log\left(\frac{\Pr(v_k[\ell]=1|y_k)}{\Pr(v_k[\ell]=0|y_k)}\right) = \log\left(\frac{\sum_{x_k:\text{bit }\ell\text{ of x}_\text{k}\text{ is 1}} \exp\left(-\frac{|y_k-x_k|^2}{2\sigma^2}\right)}{\sum_{x_k:\text{bit }\ell\text{ of x}_\text{k}\text{ is 0}} \exp\left(-\frac{|y_k-x_k|^2}{2\sigma^2}\right)}\right).$$

These values can now be given to the routine 'decoder_75'.

2.3 Tasks

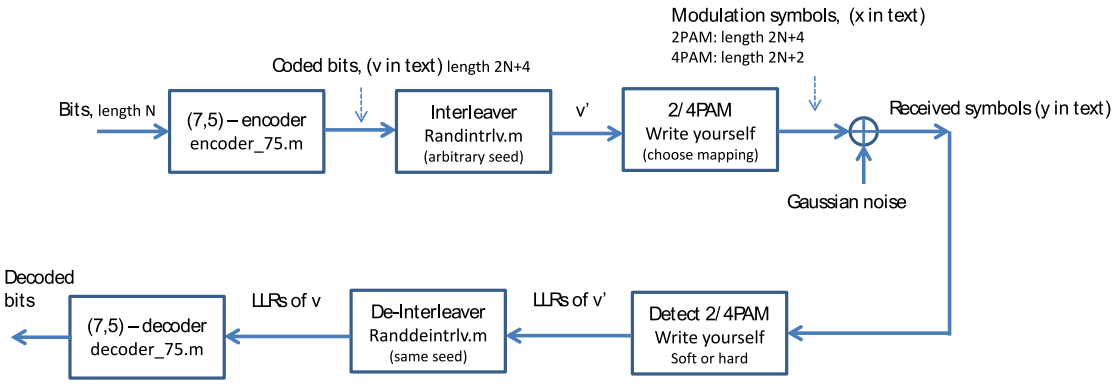
You should simulate and compare four systems;

1. 2PAM with hard detection
2. 2PAM with soft detection
3. 4PAM with hard detection
4. 4PAM with soft detection

Plots of bit error rates versus E_b/N_0 should be given, and conclusions should be made. Particular questions that should be answered are:

1. What is the gain of soft detection over hard detection?
2. Which system is best, uncoded 2PAM or (7,5)-encoded 4PAM?
3. At an error rate of 10^{-3}, plot the obtained results in Figure 4.6-1 at page 229 in the course book.
4. Extra: What is the role of the interleaver?
5. Extra: With soft detection, are we performing optimal detection or not?

A system model is shown at next page. A good number for the block length N is 10000, which means that you must average over a number of blocks. This is because my routine is badly written.


Modulation symbols, (x in text)
2PAM: length 2N+4
4PAM: length 2N+2
Coded bits, (v in text) length 2N+4
Bits, length N
(7,5) – encoder
encoder_75.m
Interleaver
Randintrlv.m
(arbitrary seed)
v'
2/4PAM
Write yourself
(choose mapping)
Received symbols (y in text)
Gaussian noise
Decoded
bits
(7,5) – decoder
decoder_75.m
LLRs of v
De-Interleaver
Randdeintrlv.m
(same seed)
LLRs of v'
Detect 2/4PAM
Write yourself
Soft or hard