ELEC 470 Communication and Information Systems Fall 2014

Project #2: PCM Coding and Compression

Due in BRKI 368 and online by 11:59 pm on Tuesday, Dec. 9, 2014

(revised at 6:45 pm Dec. 7, 2014)

In this project you will use Matlab to generate a pulse code modulated (PCM) binary encoded signal from a sampled audio signal. Uniform sampling intervals will be employed first and then mu-law compression will be applied to the signal before it is encoded. The signal-to-noise ratio of each case will be compared.

***Assignment*:**

1. Using Matlab, load the audio sample file SpeechSample.mat (Matlab data file format) that contains a modified sampled audio signal from the Keele University speech database. The file is available in Professor Kelley's public space in the ELEC 470 folder. The signal was sampled at a rate of 20 kHz using 16 bits per sample. A segment of Matlab code that can be used to load the speech file is given below:

```
clear           % clear variables from any previous program executions
close all       % delete all figures

% Load speech sample stored in *.mat file. Two vectors, 'tsamples' and
% 'message', are stored in the file. 'tsamples' contains the sample times
% in seconds, and 'message' contains the sampled speech data at those
% times:

load SpeechSample;
```

   Note that using 16 bits per sample leads to $L = 2^{16} = 65{,}536$ quantization levels. There are 70,500 samples in all in the speech file, which is a large amount of data. Therefore, to keep the project somewhat manageable, use the Matlab colon operator to select samples 3000 through 8000 of the signal. These correspond to the word "north" in the reading passage stored in the speech file.

2. Within the same Matlab m-file, normalize the message signal data to a maximum value of $m_p = 1$ (i.e., so that $|m(t)|_{\text{pk}} = 1$), and then quantize each sample using $L = 2^8 = 256$ uniform levels. Of course, the message signal is already quantized to 65,536 levels, but we are going to assume that the original sampled message is so finely quantized that it is essentially continuous (an unquantized analog signal) compared to the relatively coarse quantization of 256 levels we will use. If a sample value falls exactly on a boundary between two quantization levels, round the sample value toward zero (i.e., round down for positive values and up for negative values). **[Added 12/7/14:]** If a sample value is exactly zero, its quantized value should be the first quantization level above zero (i.e., round up).

(*continued on next page*)

3. Save the sample values represented in hexadecimal format in an ASCII file, with one line per sample. There are 5001 samples to be encoded, so please do not print the file! A submission link will be provided in Moodle for this file and one other file you will generate. You do not have to include the sample times in the ASCII file. Eight bits are used to encode each sample, so each sample can be represented using a two-digit hexadecimal value. For example, 01100101 would be represented by 65 in hex, and 11010001 would be represented by D1 in hex. (Recall that A = 10, B = 11, C = 12, D = 13, E = 14, and F = 15.)

4. Calculate the mean squared amplitude of the message signal, and then use that information to calculate the signal-to-quantization noise ratio of the sampled data (for $L = 256$) in decibels. The calculations should be included in your m-file, but you should write the numerical values of the mean squared amplitude and the SNR on the hard copy of the m-file you submit.

5. Now apply mu-law compression to the normalized sampled amplitude data (for samples 3000 through 8000) using $\mu = 255$. Again encode the data into an 8-bit binary word (corresponding to $L = 256$), and write the encoded samples in hexadecimal format to a second ASCII file, with one line per sample.

6. Calculate the mean squared amplitude of the compressed signal. Also calculate the signal-to-quantization noise ratio of the compressed sampled data (again for $L = 256$) in decibels. The calculations should be included in your m-file, but write both numerical results on the hard copy of the m-file you submit.

7. Submit the two ASCII files you have generated to the link provided at the Moodle site. The file names should have the formats:

   Lastname_uniformPCM.txt   and   Lastname_compressedPCM.txt,

   where of course “Lastname” is your last name. The rest of the file name text should be self-explanatory.

   Also submit a hard copy of the m-file you used to make all calculations with the SNRs and mean squared message amplitudes for both cases written on the hard copy and clearly identified.