ECEG 470/670 Communication and Information Systems Fall 2025

Project #2: PCM Coding and Compression

In this project you will write a Matlab script (m-file) to generate a pulse code modulated (PCM) binary encoded signal from a sampled audio signal. Uniform quantization intervals without compression 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 at the course Moodle site in the "Recitation" section. 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 (about 3.5 seconds of recording time) 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. If you are interested, you can use the following Matlab command to listen to the full speech file:

   ```
   sound(message, fsamp, samp_bits);
   ```

   where `fsamp` is the sampling frequency (20,000 Hz) and `samp_bits` is the number of bits per sample (16). This command should play the audio clip through your computer speakers.

2. Within the Matlab m-file, normalize the message signal data to a maximum value of 1 to make $|m(t)|_{pk} = 1$. The peak value of $|m(t)|$ should be determined from the full message signal, not just samples 3000 through 8000. Then quantize each sample using $L = 2^8 = 256$ uniform levels (8 bits). 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 that 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). If a sample value is exactly zero, round up to the first quantization level above zero.

3. Save the sample values represented in hexadecimal format (i.e., with levels 0 through 255) in an ASCII file, with one line per sample. Use the Matlab `dec2hex` command. There are 5001 samples to be encoded, so please do not print the file! I will ask you to e-mail this file and one other file to me after you have completed the assignment. 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.) Name this ASCII file using the following format, where the text "Lastname" is replaced with your last name:

   Lastname_uniformPCM_fa25.txt

4. Calculate the mean squared amplitude of the message signal, and use that information to calculate the signal-to-quantization noise ratio (SQNR) of the sampled data (for $L = 256$) in decibels. Add the results of the two calculations to the end of the ASCII file containing the uncompressed sample values generated in the previous step. The code to make the two calculations and to write the results to the ASCII file should be included in your Matlab script (m-file).

5. Now apply mu-law compression to the normalized sampled amplitude data (for samples 3000 through 8000) with $\mu = 255$. As with the uncompressed data, encode the compressed data using 8 bits, corresponding to $L = 256$, and write the encoded samples in hexadecimal format to a second ASCII file, with one line per sample. Name this ASCII file using the following format, where the text "Lastname" is replaced with your last name:

   Lastname_compressedPCM_fa25.txt

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. Add the results of the two calculations to the end of the ASCII file containing the compressed sample values generated in the previous step. The code to make the two calculations and to write the results to the ASCII file should be included in your Matlab script (m-file).

7. Add detailed comments to your Matlab script (m-file) so that it is easy to see what is being calculated in each step and so that the meanings of the variables that you chose are clear. Add header comments that include your name, ECEG 470 (or 670), the semester (Fall 2025), the project number ("Project #2"), and a brief description of the file.

8. I will post ASCII files with partial solutions soon so that you can check your results.

9. E-mail to me the two ASCII files that you generated and the Matlab script (m-file) that you used to make all calculations by **11:59 pm on Friday, October 31, 2025**.