



New Mexico State University
Klipsch School of Electrical Engineering

EE589 Digital Speech Processing
Spring 2011 – Project #3
Due: 8.55am Thu. Mar. 10

Name: _____

Grade: _____

Project

The purpose of this project is to study waveform-based speech coding. We will analyze uniform and non-uniform quantizers and μ -law companding. Some of the software will be developed by students in MATLAB based on the *DSP Software Toolkit* which can be found at

http://www.ece.nmsu.edu/~pdeleon/Teaching/EE589/DSP_Toolkit.pdf

In addition, we will utilize several popular MATLAB toolboxes such as the Signal Processing Toolbox and Statistics Toolbox as well as third-party toolboxes such as Voicebox

<http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>

You are free (unless otherwise noted) to use other toolboxes or develop your own functions.

Report

Please submit a printed report with your results including commentary and plots. Please `mailto:pdeleon@nmsu.edu` a zip file containing your report and all MATLAB code to recreate your results.

Notes

Unless otherwise noted, all speech signals should be from the student's own recordings. Students are encouraged to discuss detailed, technical aspects with each other and Prof. De Leon. Students are encouraged to utilize existing MATLAB functions in this project. However, students must write all other required codes on an *individual* basis.

Some problems ask for comments. As is always the case, your analysis, comparison to expected theory or model, and resulting comments are far more important than the code or plot. Therefore, please pay special attention to your comments.

For some of these experiments, we want to ensure the signal has a zero mean:

```
x = x - mean(x); % force x to have zero mean
```

For some of these experiments, we want to ensure the signal has a unit variance:

```
x = x ./ sqrt(cov(x)); % force x to have unit variance
```

For some of these experiments, we want to ensure the signal amplitude is normalized:

```
x = x ./ max(abs(x)); % for -1 <= x <= 1
```

1 Software Tools

Please write the following function from the *DSP Software Toolkit*: **requantize.m**. Be sure to use the appropriate function header described on p. 5 and available at

<http://www.ece.nmsu.edu/~pdeleon/Teaching/EE589/funchead.m>

In addition, you may wish to use tools from the MATLAB Signal Processing Toolbox and the Voicebox Toolbox in this project.

2 Problems

1. Generate a 10 s ($f_s = 8000$), uniformly-distributed (-1.0 to $+1.0$) random signal, x (rand in MATLAB). Requantize (see requantize function below) the signal with 3, 5, and 8 bits/sample (three separate signals), to generate the quantized signal \hat{x} . Plot the quantization error signals $e = x - \hat{x}$ and compute the SNRs:

```
sigma_x2 = cov(x);
sigma_e2 = cov(e);
SNR_dB = 10*log10(sigma_x2 / sigma_e2)
```

How do your SNR values compare with the “6dB-per-bit rule”?

2. Repeat Prob. 1 using 60s of speech.wav instead of the random signal. How does your 3 bit quantizer’s SNR compare with SNR data in Lecture 11?
3. Using speech.wav, recreate the speech pdf plot in Lecture 10. Be sure your speech signal is zero mean, unit variance before computing the pdf.
4. Comband ($\mu = 100$) speech.wav and save the resulting signal, $T(x[n]) = g[n]$ as comband_speech.mat (floating-point). Plot the pdf of the comband speech. How does it compare with the plot in 3)?
5. The inverse transform for the μ -law compander is given by

$$x[n] = \left[\exp\left(\frac{\log(1 + \mu)}{x_{\max}} |g[n]|\right) - 1 \right] \frac{x_{\max}}{\mu} \text{sign}(g[n]) \quad (1)$$

where $g[n]$ are comband samples. Using the inverse transform, uncomband comband_speech.mat. Plot the difference between this signal and your original SPEECH.WAV. It should be very nearly zero.

6. Consider Figure 12.10. Quantize floating-point samples in comband_speech.mat with 3, 5, and 8 bits/sample (three separate signals) and then uncomband each quantized signal to get \hat{x} .
 - (a) Plot (plotcsig2) the three quantization error (noise) signals, $e = x - \hat{x}$. Comment
 - (b) Compute the SNRs for each signal. Compare these to Lecture 11 and comment.
 - (c) Using your results in (b) and those in Prob. 2, quantify the benefit of the μ -law quantizer.
7. We can use MATLAB’s kmeans function (Lloyd’s algorithm), to design a non-uniform quantizer for speech. The non-uniform quantizer can then be applied to signal:

```
[x,fs,bits] = wavread('Speech.wav',60*fs);
K = 8; % K levels of quantization
[idx,r] = kmeans(x,K);
x_hat = zeros(length(x),1);
for k = 1:K
    tmp = find(idx == k);
    x_hat(tmp) = r(k);
end;
```

(a) Read MATLAB's help file on `kmeans` and then use the above code to construct a 3-bit non-uniform quantizer for speech and quantize the first 20s of `speech.wav`. List your eight reconstruction levels (cluster centroids).

(b) Determine the resulting SNR of your non-uniform quantizer. How does it compare to Lecture 8 data (non-uniform quantizer designed specifically for speech input pdf)?