# ECEN 455 - Digital Communications

# Lab 3 - Analog to Digital Conversion

## Fall 2015

### Introduction

Many digital communication systems are used to transmit messages that are inherently analog. For example, your cell phone transmits your analog voice in a digital fashion. In order to do this, it is first necessary to convert the analog signal into a digital form. We have discussed several methods for doing this in class and will investigate a few of them further in this lab. In particular, we will be digitizing a few different speech waveforms.

In order to represent an analog signal in a digital fashion, we often use some form of quantization. This process corrupts the signal in a manner that is not reversible and so it is important to keep this corruption to a tolerable level. The amount of error induced by quantization is often measured qualitatively in terms of a signal-to-quantization-noise ratio (SQNR). Let $\{x_1, x_2, x_3, \ldots\}$ be a sequence of voice samples. The average power in the voice signal is then

$$P_x = \frac{1}{N} \sum_{n=1}^{N} |x_n|^2 .$$

If the signal is quantized to $y_n = Q(x_n)$, then the quantization noise (or error) is $e_n = x_n - Q(x_n)$. The average power in the quantization noise is

$$P_e = \frac{1}{N} \sum_{n=1}^{N} |x_e|^2 .$$

The SQNR for the quantized signal is then defined as

$$SQNR = 10 \log_{10} \left( \frac{P_x}{P_e} \right) .$$

SQNR can often be a misleading measure of speech quality, so it is common to use subjective (human) hearing tests to measure the amount of distortion in a speech signal. The mean opinion score (MOS) rating scale is most commonly used and the rubric is as shown in Table 1 whereby speech quality is rated on a scale of 1-5. In this lab, you will use both SQNR and MOS to measure the quality of digitized speech.

| Rating | Speech Quality | Level of Distortion |
|---|---|---|
| 5 | Excellent | Imperceptible |
| 4 | Good | Just perceptible, but not annoying |
| 3 | Fair | Perceptible and slightly annoying |
| 2 | Poor | Annoying, but not objectionable |
| 1 | Bad | Very annoying and objectionable |

Table 1 - MOS speech quality rubric

## Task 1 - Uniform Quantization

On the course web site there are 4 different speech files named "speech0" "speech1", etc. They are sampled at 8kHz with 16bits per sample. In this first task, we are going to look at what happens to the quality of speech as we reduce the number of bits in the quantization. Since $2^{16}$ is a fairly large number, we will treat the 16 bit quantization as if there is no quantization at all.

- First, construct a MATLAB function `[xq, SQNR]=UniformQuantizer(x,n)` which performs uniform quantization on the input signal `x` using `n` bits of quantization ($M = 2^n$ quantization levels). The outputs should be the quantized signal `xq` and the SQNR in dB.

- Choose one of the speech files and run your quantizer on it using $n = 8$ bits. Note the SQNR and also listen to the quantized signal and give it an MOS score.

- Repeat the quantization of the speech signal using $n = 7, 6, 5, \ldots, 1$ bits of quantization. Record SQNR and an MOS score for each case.

- Provide a plot of SQNR vs. $n$, and another plot of MOS vs. $n$. Do you see evidence of the 6dB rule on your SQNR plot?

## Task 2 - Non-uniform Quantization with $\mu$-law Companding

In this task, you will repeat exactly what you did in Task 1, except this time you should use a $\mu$-law quantizer. Choose the standard value of $\mu = 255$. Recall that $\mu$-law quantization can be performed by first compressing the signal using the function

$$g(x) = \frac{\ln(1 + \mu|x|)}{\ln(1 + \mu)} \text{sgn}(x),$$

then use a uniform quantizer on the compressed signal, and finally expanding the compressed and quantized signal using the function

$$g^{-1}(x) = \frac{1}{\mu}\left[(1 + \mu)^{|x|} - 1\right] \text{sgn}(x).$$

- First, construct a MATLAB function `[xq, SQNR]=MuLawQuantizer(x,n,mu)` which performs $\mu$-law quantization on the input signal `x` using `n` bits of quantization ($M = 2^n$ quantization levels). The outputs should be the quantized signal `xq` and the SQNR in dB.
- Choose one of the speech files and run your quantizer on it using $n = 8$ bits. Note the SQNR and also listen to the quantized signal and give it an MOS score.
- Repeat the quantization of the speech signal using $n = 7, 6, 5, \ldots, 1$ bits of quantization. Record SQNR and an MOS score for each case.
- Provide a plot of SQNR vs. $n$, and another plot of MOS vs. $n$. Comment on whether or not you notice any improvement relative to the uniform quantizer in Task 1.

## Task 3 - Delta Modulation

In order to quantize the signal using delta modulation, we will need to upsample the signal to a higher sampling rate. You can accomplish this in one of several manners:

- Use the reconstruction formula in the sampling theorem to find the value of the signal at any points in between the given sample values. In an ideal world, this should give you the the reconstructed signal perfectly, but it requires an infinite number of samples (which you do not have). However, the approximation you get with a finite number of samples should be very good. It also requires the original signal to be perfectly bandlimited which it is not, but again this should introduce minimal error in our case.

- You can upsample the signal followed by a lowpass filter. If you carefully choose the filter, this should also work quite nicely.

- You can use some kind of practical interpolation. Linear interpolation might be good enough or maybe you could get fancy and use a cubic spline or something similar. There are many interpolation functions in MATLAB that you could probably use here.

Using one of the techniques described above (or maybe something you devised on your own), write a program `xu=UpDownSample(x,dx,du)`, where `x` is a vector containing the original samples, `xu` is a vector of samples at the new sampling rate, `dx` specifies the sampling interval for `x`, and `du` specifies the sampling interval for `xu`. The new and old sampling rates should not need to be related by an integer factor and the program should allow for the new sampling rate to be either higher (upsampling) or lower (downsampling) than the original signal. In your lab report, describe how your function works and how you tested it to make sure that it functions correctly.

Next, create a program `xq=DeltaMod(x,delta)` to implement delta modulation, where `x` is the vector of input samples, `xq` is the quantized signal, and `delta` is the step size. As always, test your program on a simple example that you can work out by hand and report the results of your test. Once you have this program running correctly, complete the following measurements:

- Run your delta modulation program on the same 8kHz audio file used in Tasks 1 and 2. You will have to experiment with the step size to see what works best. Be sure to mention in your lab report what step size you used and why. Note that since delta modulation uses 1 bit per sample, this corresponds to the same bit rate as using the quantizers in Tasks 1 and 2 with $n = 1$. Note the SQNR and listen to the quantized signal and give it an MOS score.

- Next, upsample the audio signal to 16kHz and apply the delta modulator. Again, you will have to adjust the step size. Note the SQNR and listen to the quantized signal and give it an MOS score.

- Repeat this process with the audio signal upsampled to 24kHz and 32kHz. In each case, note the SQNR and listen to the quantized signal and give it an MOS score.

- Compare your SQNR numbers and also your MOS scores with what was obtained using the quantizers in Tasks 1 and 2. Comment on whether you find delta modulation to be better or worse than just using a quantizer.