

Comping

This lab will demonstrate the effect of quantization while giving you practice in capturing audio waveforms and processing them with Matlab.

Introduction

In this lab you will study the effect of companding on speech waveforms.

You will record a short sample of your own speech and then use Matlab to analyze the quantization noise with and without the use of μ -law companding.

Quantized Speech Waveforms

Much of the information transmitted over data communications networks is speech. A common method of digitizing a speech signal is known as Pulse Code Modulation (PCM).

The speech waveform is sampled at a fixed rate, typically 8 kHz, and each sample is quantized using an analog-to-digital converter, typically with a resolution of 8 bits. The resulting 64 kb/s data stream is then transmitted over a network. The receiver uses a D/A converter and low-pass “reconstruction” filter to reconstruct an analog waveform that is continuous in both level and time.

Quantization Noise

Since each sample of the speech waveform is quantized to one of 256 levels (2^8), the reconstructed waveform is not the same as the original waveform. Quantization has the same effect as adding a noise signal whose level is equal to the quantization error. This distortion of the signal is called quantization noise.

The quantizer step size, q , is the signal voltage range divided by the number of steps. For example, for an 8-bit A/D converter with input levels limited to $\pm V$ then $q = 2V/256$.

When the probability distribution of the quantization noise is uniform, the quantization noise power is $q^2/12$ where q is the quantization step size¹.

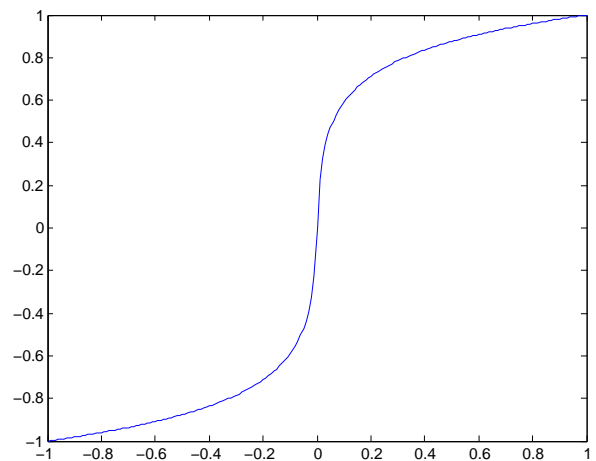
¹This is the variance of a uniform distribution with range q .

Comping

Comping is a *portmanteau* word derived from compressing and expanding. A signal can be compressed before quantization and the expanded after un-quantization.

A compressor applies a non-linear function (e.g. logarithmic) to the audio waveform.

The μ -law compression function² for $\mu = 256$ is shown below. The input level is along the x-axis and the corresponding output level along the y-axis.



Comping provides more gain to lower-level inputs. This increases the signal-to-quantization noise ratio at lower signal levels and reduces it for higher signal levels. However, since the quantization noise is more noticeable at low signal levels, the use of companding can improve the perceived quality of the speech signal.

Other signal processing techniques can be used with speech signals. It is possible to dynamically vary the gain of the system according to short-term measurements of the signal power. This type of dynamic gain control is the basis of various noise-reduction techniques that were popular with analog recording technologies.

²A similar function, “A-law,” is commonly used outside North America.

Some communication systems use compression by itself – without a corresponding expander at the receiver. Although this distorts the waveform, the increase in average signal power can improve intelligibility for systems operating at low SNRs. This technique can also be [ab]used to increase the apparent volume of advertising messages.

Software

In this lab you will use the free audio editing program Audacity to record and select a segment of your speech. You will use Matlab (or one of the free compatible programs) to apply the compression, expansion and quantization functions to the waveform and to compute the signal and quantization noise powers, probability distributions and SNRs.

Comanding and Quantization Functions

The μ -law compression formula is:

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

which can be implement as the following Matlab function:

```
function y = ulaw(x)
mu = 255 ;
y = sign(x) .* log(1+mu*abs(x)) ./ log(1+mu) ;
```

You will write the inverse (expansion) function as part of your Pre-Lab report.

The following Matlab function quantizes a signal with a range of ± 1 to (approximately) 8-bit resolution:

```
function y = quant(x)
y = round(128*x) ./ 128 ;
```

To make these functions available to your script you need to save the code above to a file with the same name as the function and a file type suffix of `.m`. For example, save the first section of code above to the file `ulaw.m`. You can also download these two functions from the course web site. The `.m` files should be placed in your working directory.

Pre-Lab

Using the Matlab companding function above as a model, write the corresponding expansion function. The expansion equation is:

$$F^{-1}(y) = \frac{\text{sgn}(y)}{\mu} \left((1 + \mu)^{|y|} - 1 \right)$$

where $\text{sgn}(x)$ is 1 if x is positive, -1 if x is negative and 0 if x is zero. The Matlab function `sign` implements this function. The Matlab function `abs` implements the absolute value function.

As above, use the element-wise Matlab division, multiplication and exponentiation operators (`./`, `.*` and `.^`) so that your function can handle vector inputs. You may want to test your function against the supplied compression function to make sure they are inverses of each other.

Also answer the following question: If the quantization range is ± 1 V and there are 256 levels, what is the expected quantization noise power?

To test companding you will make up a unique nonsense test sentence. The goal is to create a sound segment that is a couple of seconds long that includes both loud and soft sounds. See the Wikipedia article [Help:IPA for English](#) for a list of English phonemes. Choose about a half-dozen words from the IPA pronunciation key given above. Alternate consonant and vowel sample words. For example, “buy palm sang boy zoo coin.” Include your test sentence in the pre-lab report.

Submit your pre-lab report, *in PDF format*, to the appropriate dropbox on the course web site before the start of your lab.

Procedure

Recording

Use one the computers with a microphone to record a speech segment as described below.

Set the recording format: Select `Edit` \rightarrow `Preferences`. Under `Devices` \rightarrow `Recording` set the channels to 1 (mono) and under `Quality` \rightarrow `Sampling` set the sample rate to 8 kHz and 16-bit samples.

Press the record button and hold the microphone near your mouth while you say the test sentence.

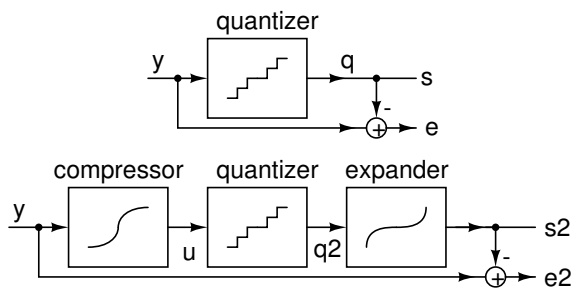
Adjust the microphone gain control while talking so that the recorded signal level is as high as possible without clipping the signal peaks.

Use the cursor to select only the portion of audio you want to analyze. Then remove the rest of the audio as follows: Edit -> Remove Audio or Labels -> Trim Audio, then Tracks -> Align Tracks -> Start to Zero.

Save the audio samples to a file as follows: File -> Export and select WAV(Microsoft) signed 16-bit PCM.

Signal Processing

The diagram below shows systems without and with companding:



where y is the input speech signal, q is the quantized signal, s is the output signal and e is the quantization noise signal. With companding added the compressed signal is u and q_2 , s_2 and e_2 are the corresponding quantized, output and quantization noise signals with companding.

Read and Normalize Waveform

Use the Matlab function `wavread()` to read the samples of the recorded waveform:

```
y = wavread('lab2.wav') ;
```

Normalize the samples so that the maximum absolute value is 1:

```
y = y ./ max(abs(y)) ;
```

Plot the result (and save it for your report):

```
plot(y) ;
```

Quantization Noise without Companding

Create a quantized version of the signal and an error (difference) signal:

```
q = quant(y) ;
s = q ;
e = y - s ;
```

Compute the signal power and the quantization noise power (variance):

```
var(q)
var(e)
```

Is the quantization noise power what you expected?

Compute the instantaneous ratio of signal to quantization noise power in dB³ as shown below, plot the histogram over the range 20 dB to 70 dB and find the median⁴ SNR:

```
snr=10.*log10((eps+s.^2) ./e.^2) ;
hist(snr, [20:70])
median(snr)
```

Record the results and save the plot for your lab report.

Quantization Noise with Companding

Compress the signal using the mu-law function:

```
u=ulaw(y) ;
```

Plot the compressed signal (and save the plot for your report):

```
plot(u)
```

Does this signal appear different than the uncompressed signal? How?

Repeat the above quantization steps but uncompress the quantized signal before computing the error:

```
q2 = quant(u) ;
s2 = ulawex(q) ;
e2 = y - s2 ;
```

³We add `eps` (epsilon, the smallest value) to avoid computing 0/0 which would give an undefined (NaN) result.

⁴When data contains outliers such as very high SNRs (e.g. when the quantization error is very low), the median is a more reliable estimate of the typical value.

Compute the signal power, quantization noise power, per-sample SNR, the median SNR and plot the histogram of the quantization SNR as before (substitute s_2 for s , etc.).

Did the signal or quantization noise powers change significantly? Why or why not?

Report

Prepare a report including the following:

- the plots of the uncompressed (y) and compressed signal (u)
- the plots of the distribution of the instantaneous SNR (in dB)
- the answers to the questions above
- the summary table described below
- a *brief* explanation, with reference to your results, of how companding might improve (or not) the perceived quality of the speech signal.

Summarize your results in a table. The table should have four columns: type of measurement, the value without companding, the value with companding and difference (ratio) in dB. Include the following measurements: signal power, quantization noise power, median SNR (dB).

Submit your report *in PDF format* and the `.wav` file containing your digitized waveform to the dropbox on the course web site.