5-min knowledge sharing/discussion

Week 10	**12010615**	**廖子涵**
	12010402	**成松泽**
	12010520	**伍泽星**
	11910526	**杨家升**
Week 11	12010102	陈思佑
	12012927	聂晶磊
	12012709	杨恒宇
	12012926	雷其融

TencentMeeting: 614-588-8061

Lab X
Digital Coding of Speech Signals
– part 2

DONG Yunyang

dongyy@mail.sustech.edu.cn

North Tower 316, College of Engineering

TencentMeeting: 614-588-8061

2023-04-18

Purpose of this lab...

1. Test and understand the process of u-law quantization
2. Compare uniform and u-law quantizers on their SNRs
3. Test and understand the process of adaptive quantization

Problem 1

11.27. (MATLAB Exercise) The goal of this MATLAB exercise is to experiment with the process of μ-law compression of speech. Using the MATLAB function `y=mulaw(x,mu)`, where:

- `x` is the input signal
- `mu` is the compression parameter
- `y` is the μ-law compressed signal

$$y[n] = \frac{\ln[1 + \mu|x[n]|]}{\ln[1 + \mu]} \cdot sign[x[n]]$$

Do the following:

(a) Create a linearly increasing input vector $(-1{:}0.001{:}1)$ and use it, with the above function `mulaw( )` to plot the μ-law characteristic for $\mu = 1, 20, 50, 100, 255, 500$, all on the same plot. (Note that the value $\mu = 255$ is a standard value used in landline telephony.)

(b) Using the segment of speech, from sample 1300 to sample 18,800, from the file `s5.wav`, and a value of $\mu = 255$, plot the output waveform $y[n]$ of the μ-law compressor. Observe how the low amplitude samples are increased in magnitude. Plot a histogram of the output samples

(c) A block diagram of a full μ-law quantization method is shown in Figure P11.27. To implement this system, you are asked to write an m-file for the inverse of the μ-law compressor. This m-file should have the following calling sequence and parameters:

```
function x=mulawinv(y,mu)
% function for inverse mulaw for Xmax=1
% x=mulawinv(y,mu)
% y=input column vector
% mu=mulaw compression parameter
% x=expanded output vector
```

Use the technique used in `mulaw( )` to set the signs of the samples. Test the inverse system by applying it directly to the output of `mulaw( )` without quantization; e.g., try

```
v=mulawinv(mulaw(2,255),255);
```

In this case, v and x should be essentially identical.

(d) The MATLAB statement

```
yh=fxquant(mulaw(x,255),6,'round','sat');
```

implements a 6-bit μ-law quantizer. Thus the μ-law compressed samples are coded by a uniform quantizer with 6 bits. When the coded samples are used in a signal

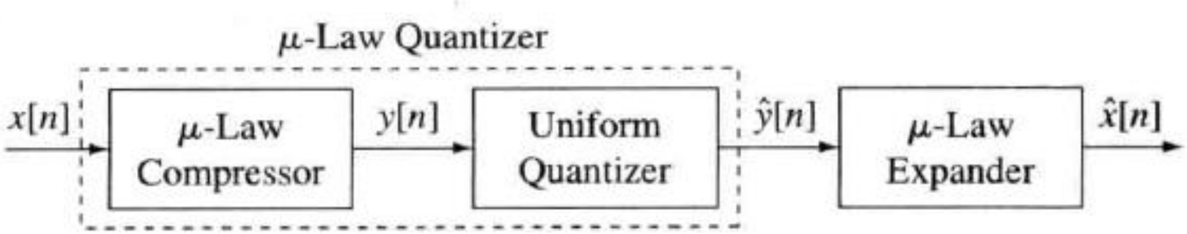

FIGURE P11.27
Representation of μ-law quantization.

processing computation, or when a continuous-time signal needs to be reconstructed, the uniformly coded samples of the μ-law quantizer must be expanded. Hence, the quantization error is likewise expanded, so that to determine the quantization error, it is necessary to compare the output of the inverse system to the original samples. Thus the quantization error is ~~e=mulaw(yh, 255) - x~~; . Using uniform quantizers with 10, 8, and 4 bits, compute and plot the first 8000 samples of the resulting quantization error, plot a histogram of the quantization error amplitudes, and plot the power spectrums of the resulting quantization errors, all on common plots.

e = mulawinv(yh, 255) – x;

Problem 2

11.28. (MATLAB Exercise) The goal of this MATLAB exercise is to compare uniform and μ-law quantizers on the basis of their SNRs.

A convenient definition of a waveform (with duration L samples) SNR is:

$$\text{SNR} = 10\log\left(\frac{\sum_{n=0}^{L-1}(x[n])^2}{\sum_{n=0}^{L-1}(\hat{x}[n]-x[n])^2}\right).$$

Note that the division by L, as required for averaging, cancels in the numerator and denominator.

It was shown in this chapter that the SNR for a uniform B-bit quantizer was of the form

$$\text{SNR} = 6B + 4.77 - 20\log_{10}\left(\frac{X_{\max}}{\sigma_x}\right),$$

where $X_{\max}$ is the clipping level of the quantizer (1 for the files in these exercises), and σ_x is the rms value of the input signal amplitude. We see that the SNR increases 6 dB per bit added to the quantizer word length. Furthermore we see that if the signal level is decreased by a factor of 2, the SNR decreases by 6 dB.

(a) Write an m-file to compute the SNR, given the unquantized and quantized versions of the signal. The calling sequence and parameters of this m-file should be:

```
function [s_n_r,e]=SNR(xh,x);
% xh=quantized signal
% x=unquantized signal
% e=quantization error signal (optional)
% s_n_r=snr in dB
```

s5.wav . (1300:18800)

Use your SNR function to compute the SNRs for uniform quantization with 8 and 9 bits. Do the results differ by the expected amount?

(b) An important consideration in quantizing speech is that signal levels can vary with speakers and with transmission/recording conditions. Write a program to plot the measured SNR for uniform and μ-law quantization as a function of $1/\sigma_x$. Vary the signal level by multiplying the input speech signal, `s5.wav`, by the factors `[2^(0:-1:-12)]` and measure the SNR for all 13 cases. Plot the resulting SNR

semilogx(X,Y)

data on a semi-log plot. Repeat the calculation for 10, 9, 8, 7, and 6 bit uniform quantizers and for $\mu = 100,255,500$ over the same range.

Your plots should show that the μ-law quantizer maintains a constant SNR over an input amplitude range of about 64:1. How many bits are required for a uniform quantizer to maintain at least the same SNR as does a 6-bit, μ-law quantizer over the same range?

Problem 3

11.29. (MATLAB Exercise) The goal of this MATLAB exercise is to demonstrate the process of adaptive quantization using either an IIR filter or an FIR filter. As discussed in this chapter, there are two ways of adapting the quantizer, namely gain adaptation or step size adaptation. Both adaptive methods require the estimation of the signal standard deviation, $\sigma[n]$. Assuming that the signal has a mean of zero (hence you need to subtract any DC component of the signal prior to variance computation), the signal variance, $\sigma^2[n]$, can be computed using an IIR filter of the form:

$$H(z) = \frac{(1-\alpha)z^{-1}}{(1-\alpha z^{-1})}$$

as:

$$\sigma^2[n] = \alpha\sigma^2[n-1] + (1-\alpha)\cdot x^2[n-1],$$

or, using a rectangular window FIR filter of length M samples, as:

$$\sigma^2[n] = \frac{1}{M}\sum_{m=n-M+1}^{n} x^2[m].$$

Using an IIR filter, with values of α of 0.9 and 0.99, and using the speech file `s5.wav`, calculate the standard deviation, $\sigma[n]$, of the speech signal and plot the following:

1. both the speech samples, $x[n]$, and the superimposed standard deviation samples, $\sigma[n]$, for the sample range of $2700 \le n < 6700$ on the upper plot of a figure with two plots

2. the gain equalized speech samples, $x[n]/\sigma[n]$, for the sample range of $2700 \leq n < 6700$ on the lower plot of the same figure

(a) How do the two plots (i.e., the plot using $\alpha = 0.9$ and the plot using $\alpha = 0.99$) compare, especially in terms of rate of adapting to the changing speech level and ability to equalize the local dynamic range of the speech signal?

(b) Repeat the above exercise using an FIR filter (rectangular window) of duration $M = 10$ and $M = 100$ samples, and replot the same waveforms. How do the results using the FIR window compare to the results using the IIR filter?