

University of southern queensland

Ele3107-signal processing

Assignment 1

Submitted by

?????????

?????????

Contents

Question1	3
Question2-Part a	4
Question2-Part b	6
Question3	7
Question4	8

Question 1

Part a

Quantizing is done by dividing by $2^{|k|}$ where k is number of bits to be removed from the sample.

This can be done by bitshift function in Matlab every time by one bit to the right, and doing this for 15 times.

Part b

sample	Subjective assessment	SNR
15 bits	no perceptual difference	0.0032
14 bits	no perceptual difference	0.0121
13 bits	minor noise present	0.0456
12 bits	quite noisy	0.1696
11 bits	quite noisy	0.5757
10 bits	poor quality	0.9114
9 bits	poor quality	0.9802
8 bits	poor quality	1
7 bits	poor quality	1
6 bits	poor quality	1
5 bits	poor quality	1
4 bits	poor quality	1
3 bits	poor quality	1
2 bits	poor quality	1
1 bit	poor quality	1

SNR is calculated as follow

We calculate the RMS of original audio and for each quantized version of it, and SNR is the division of RMS of original audio by the RMS of noisy audio.

It is clear that SNR values are matched with subjective assessment

Matlab code

```
%% Q1 part a
closeall;
clear;clc;

fs = 8000;
tmax = 2;
nbits = 16;
nchan = 1;
Recorder = audiorecorder(fs, nbits, nchan);
record(Recorder);
pause(tmax);
stop(Recorder);
```

```

yi = getaudiodata(Recorder, 'int16');
outData(:,1) = yi; % original one before quantizing
%playSound(yi,fs);

for i = 2 : nbits
outData(:,i) = bitshift(yi, -1);
yi = outData(:,i);
%playSound(yi,fs); % you should put break point at this line to listen for
every quantized record
end

% **** part b **** %
P_signal = mean(outData(:,i).^2);
for i = 2 : nbits
P_noise = mean(outData(:,i).^2);
SNR(i-1,1) = P_signal/P_noise;
end

```

Q2

Part a

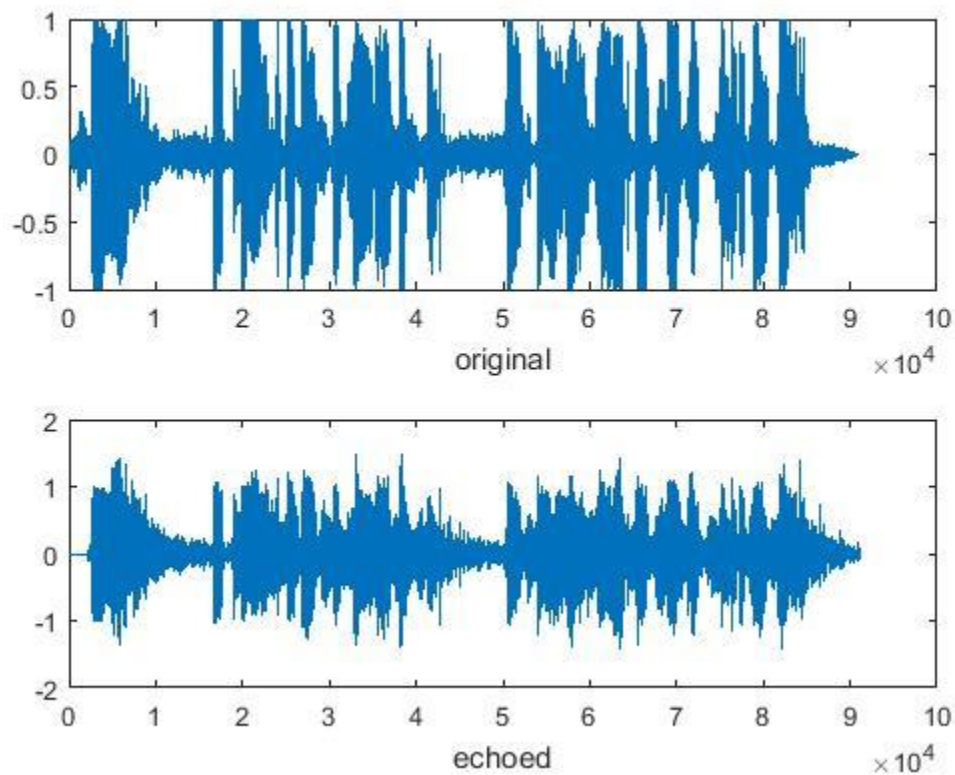


Figure 1 $D = 0.2$ and $\text{Alpha} = 0.5$

Varying delay and attenuation factor results in:

- Sometimes it causes specific part of the audio file to be repeated continuously until audio ends while the audio file is played normally
- Sometimes it causes echoed voices to interact with each other because delay frequency is less than nyquist rate
- Sometimes voice is clearer than others depending on chosen values

Matlab code

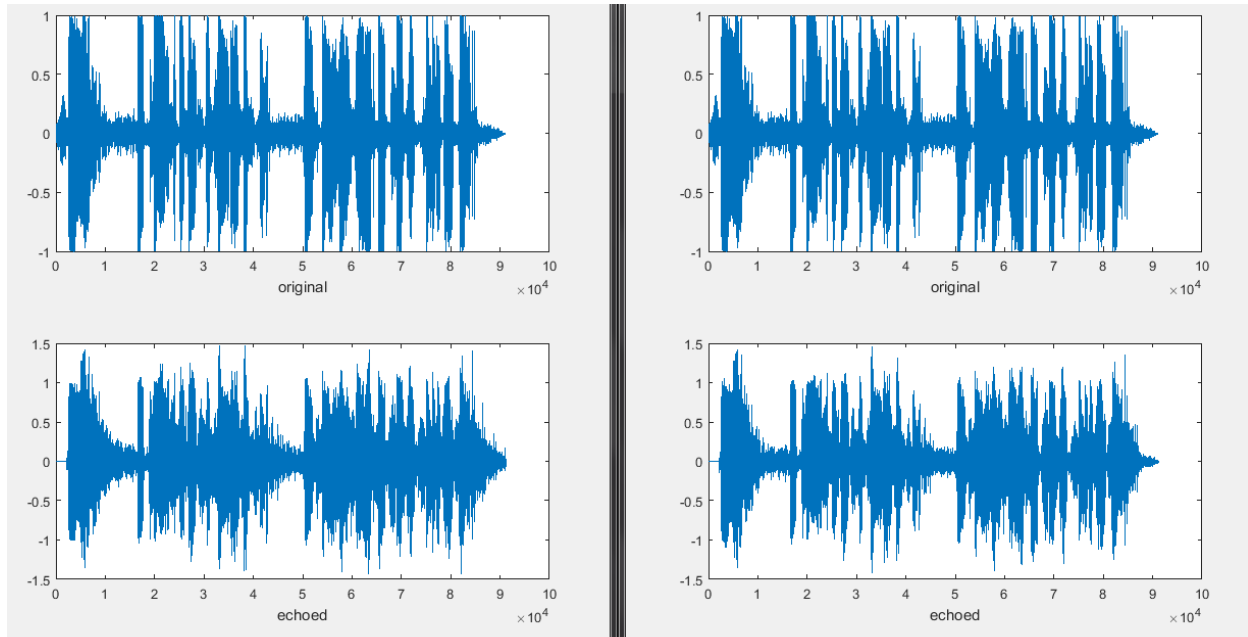
```
%% Q2 part a
closeall;
clear;clc;

[x,fs] = audioread('test.wav');%read your waed.wav
sound(x, fs);
pause(8);
subplot(2,1,1);
plot(x);
xlabel('original')
% Calculate the number of samples in the file
xlen = length(x); %Initialize all constants
a = 0.5;
delay = 0.2;
D = delay*fs;
% create a matrix , the size of x for efficient and speedier operation
y = zeros(size(x)); % filter the signal
for i = D + 1 : 1 : xlen;
y(i) = x(i) + a*y(i-D);
end;
sound(y, fs); %play the echo signal
subplot(2,1,2);
plot(y)
xlabel('echoed')
```

Part b

The difference between both lies in the period of delay in which the output y is zero so nothing is actually added to the output in this period

However when using x with the delay, then by default it will affect the output because their values won't be all zeros.



using y with D

using x with D

Matlab code

```
%% Q2 part b
%close all;
figure
clear;clc;

[x,fs] = audioread('test.wav');%read your waed.wav
sound(x,fs);
pause(8);
subplot(2,1,1);
plot(x);
xlabel('original')
% Calculate the number of samples in the file
xlen = length(x); %Initialize all constants
a = 0.5;
delay = 0.2;
D = delay*fs;
% create a matrix , the size of x for efficient and speedier operation
y = zeros(size(x)); % filter the signal
```

```

for i = D + 1 : 1 : xlen;
y(i) = x(i) + a*x(i-D);
end;
sound(y, fs); %play the echo signal
subplot(2,1,2);
plot(y)
xlabel('echoed')

```

Q3

Part a

Omega: how frequent the wave oscillates and come back to specific point again

t: whole period of oscillation

A: magnitude of the wave

fs: sampling frequency, represents steps in frequency domain at which function value is calculated

$$\text{True frequency} = \frac{\omega_0}{2*\pi} = \frac{\frac{\pi}{10}}{2*\pi} = 0.05 \text{ Hz}$$

Part b

Upon increasing alpha noise increases

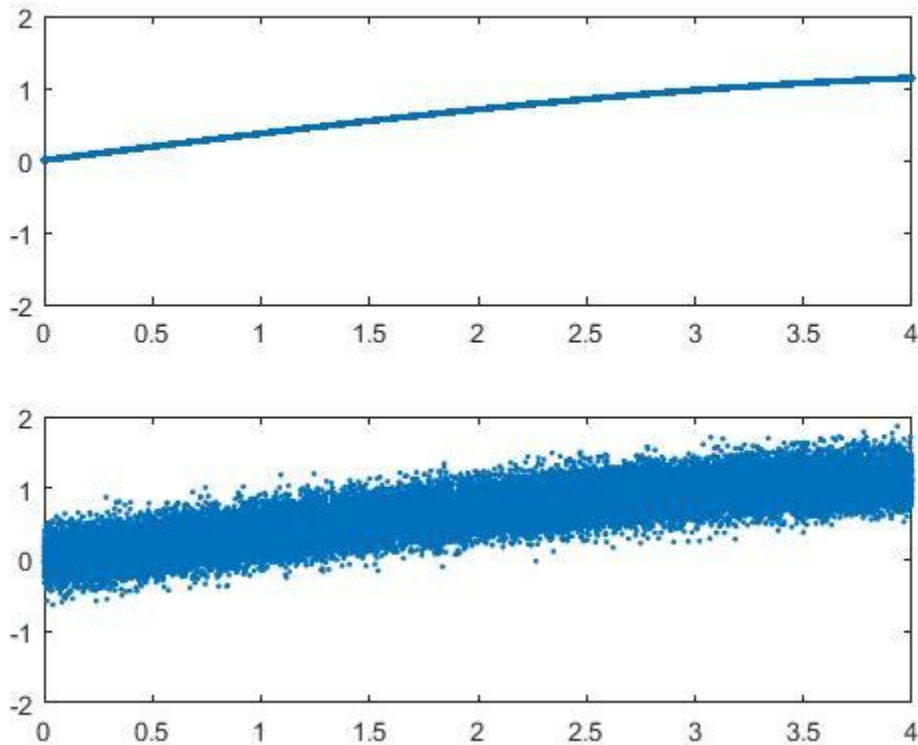


Figure 2 alpha = 0.2

Matlab code

```
%% Q3
closeall;
clear;clc;

Fs = 8000;
n = 0:1/Fs:4;
x = 1.2*sin(pi*n/10);
subplot(2,1,1);
plot (n,x, '.');
ylim([-2 2])

v = wgn(length(n),1,0);
y = x' + (0.2 * v);

subplot(2,1,2);
plot (n,y, '.');
ylim([-2 2])
```

Q4

We replace z by $e^{j\omega}$ to calculate the phase and magnitude

$$\frac{1}{1 - (p + p^*) * e^{-j\omega} + p * p^* e^{-\omega^2}} = \frac{1}{c - b * e^{-j\omega}} * \frac{c + b * e^{j\omega}}{c + b * e^{j\omega}}$$
$$= \frac{c + b(\cos(\omega) + jbsin(\omega))}{c^2 - b^2}$$

$$A = \text{Re}(H(\omega)) = \frac{c + b\cos(\omega)}{c^2 - b^2}$$

$$B = \text{Im}(H(\omega)) = \frac{jbsin(\omega)}{c^2 - b^2}$$

Where

$$c = 1 + p * p^* e^{-\omega^2}$$

&&

$$b = p + p^*$$

$$|H(\omega)| = \sqrt{A^2 + B^2}$$

$$\angle H(\omega) = \tan^{-1} \frac{B}{A}$$

Matlab code

```
%% Q4 part a
closeall;
clear; clc;

r = 0.95;
w = pi/10;
p = r * exp(1i*w);
pp = r * exp(-1i*w);

B = 1;
A = [1, -(p+pp), p*pp]; % transfer function numerator
                        % transfer function denominator

Fs = 8000;
n = 0:1/Fs:4;
x = 1.2*sin(pi*n/10);
y = filter(B,A,x);

subplot(2,1,1);
plot (n,x, '.');

subplot(2,1,2);
plot (n,y, '.');

%% Q4 part b
closeall;
clear; clc;

r = 0.95;
w = pi/10;
p = r * exp(1i*w);
pp = r * exp(-1i*w);

B = 1;
A = [1, -(p+pp), p*pp]; % transfer function numerator
                        % transfer function denominator

Fs = 8000;
n = 0:1/Fs:4;
x = 1.2*sin(pi*n/10);
y = filter(B,A,x);

freqz(B,A);
figure

subplot(2,1,1);
plot (n,x, '.');

subplot(2,1,2);
plot (n,y, '.');
```

