**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member: Mr. Kalimullah** | **Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |  |
| **Course/Section: BEE8** | **Semester: Spring 2019** |
|  |  |

**EE-330 Digital Signal Processing**

**Lab7: Sampling and Quantization of audio signal in Matlab**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4-CLO4** | | **PLO5-CLO5** | **PLO8-CLO6** | **PLO9-CLO7** |
| **Name** | **Reg. No** | **Viva / Quiz / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Team Work** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
| **Muhammad Haris Rehman** | **182797** |  |  |  |  |  |
| **Muhammad Anus Siddiqui** | **174308** |  |  |  |  |  |
| **Muhammad Hasnain Afridi** | **192414** |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Lab7: Sampling and Quantization in Matlab**

**Objectives**

The objective in this lab is down sampled audio signal and its analysis in frequency domain.

* Familiarization with sampling
* Analysis of down sampled signal in frequency domain
* Quantization

**Lab Instructions**

* The students should perform and demonstrate each lab task separately for step-wise evaluation (please ensure that course instructor/lab engineer has signed each step after ascertaining its functional verification)
* Each group shall submit one lab report on LMS within 6 days after lab is conducted. Lab report submitted via email will not be graded.

. Students are however encouraged to practice on their own in spare time for enhancing their

**Lab Report Instructions**

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

* Lab objectives
* MATLAB codes
* Results (graphs/tables) duly commented and discussed
* Conclusion

**Sampling and Quantization**

In this lab you will gain some practical knowledge about how to handle real world signals. In any modern signal processing system (e.g. in a telecommunication system), signal acquisition, its processing and efficient storage/transmission are the critical steps. In the class lectures, you have gained the knowledge about sampling of a continuous-time signal, change of sampling rates and their hierarchical criteria. Similarly you are familiar with the significance of quantization of a discrete signal. This lab covers the above mentioned tasks i.e., change of sampling rate and quantization.

**Change of Sampling Rate:**



You are familiar with the above figure for downsampling. For a discrete-time signal to be sampled by the factor of *M*, you need to first pass the signal from the low pass filter with a certain cutoff frequency to avoid the frequency aliasing in the downsampled signal.

**LAB TASK-1:**

You are given a speech signal. Consider it a discrete-time signal with the sampling frequency 𝑓𝑠=16 𝑘𝐻𝑧.

1. Load the signal in Matlab using the function *audioread.* Listen to the signal using *sound*.

2. Design a 6th order low-pass butterworth filter. Hint: see Matlab help for *butter* and *filter.* The butter command takes the normalized cutoff frequency (in the range 0-1) as an input argument where the maximum 1 means 𝑓𝑆/2

3. Consider the maximum frequency of the speech signal 𝑓𝑁 = 𝑓𝑆/2. Apply the filter.

4. Now downsample the filtered signal by the factor of 2 i.e., *M*=2. Do this manually by picking up every alternative sample and storing it in a different array.

5. See the Matlab help for the function *downsample*. Apply this function for downsampling the signal by the factors *M* = 3,5,10. Listen to the output signal in every case and prepare your conclusions. Also plot the spectrum of the input and output signal in a subplots for original and three cases for different *M*.

6. For *M* = 10, avoid the anti-aliasing filter and directly downsample the speech to listen if there is any difference. Also plot the spectrum using code given below for input and output signal .

clear all, close all, clc

% INPUT SIGNAL

[x,Fs] = audioread('sample.wav');

subplot(421), plot(x)

L=length(x);

NFFT = 2^nextpow2(L);% Next power of 2 from length of y

Y = fft(x,NFFT)/L;

f = Fs/2\*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.

subplot(422), plot(f,2\*abs(Y(1:NFFT/2+1))),grid on

title('Single-Sided Amplitude Spectrum of filtered signal')

xlabel('Frequency (Hz)')

% FILTERED

Fc = 1000;

[b, a] = butter(6,Fc/(Fs/2), 'low');

y = filter(b, a, x)

subplot(423), plot(y)

% OUTPUT FILTERED SIGNAL

L2=length(y);

NFFT2 = 2^nextpow2(L2);% Next power of 2 from length of y

Y = fft(y,NFFT2)/L2;

f = Fs/2\*linspace(0,1,NFFT2/2+1);

% Plot single-sided amplitude spectrum.

subplot(424), plot(f,2\*abs(Y(1:NFFT2/2+1))),grid on

title('Single-Sided Amplitude Spectrum of filtered signal')

xlabel('Frequency (Hz)')

% Down Sampling

z = y(1:10:end)

subplot(425), plot(z)

L2=length(z);

NFFT2 = 2^nextpow2(L2);% Next power of 2 from length of y

Y = fft(y,NFFT2)/L2;

f = Fs/2\*linspace(0,1,NFFT2/2+1);

% Plot single-sided amplitude spectrum.

subplot(426), plot(f,2\*abs(Y(1:NFFT2/2+1))),grid on %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

title('Single-Sided Amplitude Spectrum of filtered signal')

xlabel('Frequency (Hz)')

sound(z, Fs/10)

**Code to plot spectrum**

% x is your input signal Fs is sampling frequencey

[x,Fs] = audioread('sample.wav');

L=length(x);

NFFT = 2^nextpow2(L);% Next power of 2 from length of y

Y = fft(x,NFFT)/L;

f = Fs/2\*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.

subplot(2,1,2)

plot(f,2\*abs(Y(1:NFFT/2+1))),grid on

title('Single-Sided Amplitude Spectrum of filtered signal')

xlabel('Frequency (Hz)')

**Quantization**



The above mentioned figure a quantizer function you have learned . Given below is the quantization function in Matlab (algorithm)

X=(1:99)\*(8/100)-4;

N = 8; %number of quantization levels.

% find the highest value point in the signal, round it to the upper limit.

% find the lowest value point in the signal, round it to the lower limit. qstep = (high-low)/N;

Q = floor((X-low)/qstep);

low = low + qstep/2;

Y = low + qstep\*Q;

TASK-2

1. Redo the above code in Matlab, plot the original and quantized signal in a subplot for N = 8, 16, 32, 64.

2. Explain each and every step in the code.

**N=8**

clear all, close all, clc

% INPUT SIGNAL

[x,Fs] = audioread('sample.wav');

subplot(311), plot(x)

L=length(x);

NFFT = 2^nextpow2(L);% Next power of 2 from length of y

Y = fft(x,NFFT)/L;

f = Fs/2\*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.

subplot(312), plot(f,2\*abs(Y(1:NFFT/2+1))),grid on

title('Single-Sided Amplitude Spectrum of filtered signal')

xlabel('Frequency (Hz)')

low=0,qstep=1;

X=(1:99)\*(8/100)-4;

N = 8; %number of quantization levels.

% find the highest value point in the signal, round it to the upper limit.

% find the lowest value point in the signal, round it to the lower limit. qstep = (high-low)/N;

Q = floor((X-low)/qstep);

low = low + qstep/2;

Y = low + qstep\*Q;

subplot(313), plot(Y,Q)

**N=16**

clear all, close all, clc

% INPUT SIGNAL

[x,Fs] = audioread('sample.wav');

subplot(311), plot(x)

L=length(x);

NFFT = 2^nextpow2(L);% Next power of 2 from length of y

Y = fft(x,NFFT)/L;

f = Fs/2\*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.

subplot(312), plot(f,2\*abs(Y(1:NFFT/2+1))),grid on

title('Single-Sided Amplitude Spectrum of filtered signal')

xlabel('Frequency (Hz)')

low=0,qstep=1;

X=(1:99)\*(8/100)-4;

N = 16; %number of quantization levels.

% find the highest value point in the signal, round it to the upper limit.

% find the lowest value point in the signal, round it to the lower limit. qstep = (high-low)/N;

Q = floor((X-low)/qstep);

low = low + qstep/2;

Y = low + qstep\*Q;

subplot(313), plot(Y,Q)

**N=32**

clear all, close all, clc

% INPUT SIGNAL

[x,Fs] = audioread('sample.wav');

subplot(311), plot(x)

L=length(x);

NFFT = 2^nextpow2(L);% Next power of 2 from length of y

Y = fft(x,NFFT)/L;

f = Fs/2\*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.

subplot(312), plot(f,2\*abs(Y(1:NFFT/2+1))),grid on

title('Single-Sided Amplitude Spectrum of filtered signal')

xlabel('Frequency (Hz)')

low=0,qstep=1;

X=(1:99)\*(8/100)-4;

N = 32; %number of quantization levels.

% find the highest value point in the signal, round it to the upper limit.

% find the lowest value point in the signal, round it to the lower limit. qstep = (high-low)/N;

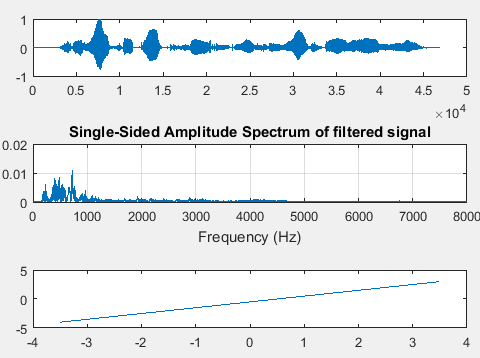
Q = floor((X-low)/qstep);

low = low + qstep/2;

Y = low + qstep\*Q;

subplot(313), plot(Y,Q)

**Snapshot**



**Explaination:**

In this task, the signal was quantized into the given number i.e. 8, 16, 32 etc. the signal was plotted with its frequency response floor command was used to round off the signal to its lower quantized value.

**Conclusion:**

In this lab, we learnt about the sampling and quantization in the Matlab. We took an audio signal passed it from the low pass filter and re-sampled it. As a result, high frequincy components were removed from the sound and the sound was not as much clear. In ths second part of the lab, we understood the quantization and implemented it in the Matlab.