Communication Lab II

(ELC3940)

Experiment No.: 03

Object:

Plot the quantization errors of the signal samples obtained in Exp. 1.

Reconstruct the signal and observe the quality degradation in the reconstructed signal.

G. No: GL3136

S. No: A3EL-02

F. No: 19ELB056

Name: Maha Zakir Khan

Date of performing experiment: 01|07|2022

Date of report submission: 07|02|2022

Software Used:

MATLAB®, Release 2021a (R2021a), a programming platform designed specifically to analyze and design systems and products. The heart of MATLAB is the MATLAB language, a matrix-based language, it provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

Procedure:

(a). Quantization Error

Let the actual sample value be $x[nT]$ and the quantized sample value be $x_q[nT]$, then the quantization error $e[nT]$ will be given by:

$$\mathbf{e[nT] = x[nT] - x_q[nT]}$$

Plot the following signals (use appropriate time scale for clear visibility):

(a). Sampled signal

(b). Quantized signal

(c). Quantization error signal

Plot of different signals of the recorded speech in Experiment 1 (first 20 samples only).

(b). Signal Reconstruction and quality assessment

Reconstruct the speech signal from the quantized signal using a 7^{th} order FIR reconstruction filter. Specify the Filter type and cutoff frequency.

Now, listen the recorded speech signal and the reconstructed signal using an audio player and rate the quality of the reconstructed signal with reference to the recorded speech signal on the following scale of 5.

1	2	3	4	5
Bad	Poor	Fair	Good	Excellent

Program:

Quantizer.m : Function that quantizes input signal amplitude to 2^k Levels and returns quantized signal with its sampled time

```
function [Quantized,n] = Quantizer(k, Signal,info)

dur = info.Duration;            % Signal duration in seconds
y = Signal(:,1);                % Channel 1 of the signal sig

% Sampling Signal Waveform
fs = info.SampleRate;           % Sampling frequency
k ;                             % bit-rate for 6-bit quantizer
l = 2^k;                        % 64-levels
n = 0:1/fs:dur-1/fs;

 ymax = max(y);
 ymin = min(y);


%Quantizing Signal Waveform using quantiz function
partition = linspace(ymin,ymax,l-1);
codebook = linspace(ymin,ymax,l);

[~,quants] = quantiz(y,partition,codebook); %quantizer

   Quantized = quants;
end
```

SignalReconstruction.m : Script to perform the reconstruction of speech signal using FIR Filter

For quantization, I used 6-bit, 64-level quantizer

```
% Loading and Quantizing Voice-Signal using 6-bit Quantizer

info = audioinfo('C:\Users\Maha Khan\Downloads\Hello4.wav');
[signal , fs] = audioread('C:\Users\Maha Khan\Downloads\Hello4.wav'); % Voice signal sig with dual channel
                                                                       % and fs as the sampling frequency
k = 6;
[Quantized,n] = Quantizer(k, signal, info);       % Calling Function to psm encode the signal
```

Instead of 20 samples, I extracted 500 samples for plotting the signal, as the first 20 samples hold zero values and don't provide much information about the signal

```
% PART A : QUANTIZATION ERROR

T = 1/fs;
nT = n(500:1000);
sampled = signal(500:1000,1)';  % Extracting 500 samples from the original signal for processing
quantized = Quantized(500:1000);% Quantizing the Extracted signal

figure('NumberTitle', 'off', 'Name', 'Quantization Error');
subplot(3,1,1)                          % Sampled Signal plot of 500 samples
stem(nT,sampled,'Marker','.');
xlabel('Discrete Time [nT] in seconds');
ylabel('Analog level');
title('Sampled Signal x[nT]');
axis([0.01 0.022 -0.05 0.02]) ;

subplot(3,1,2)                          % Quantized Signal plot of same 500 samples
stairs(nT,quantized,'LineWidth',1.2);
yline(0,'--');
xlabel('Discrete Time [nT] in seconds');
ylabel('Quantized level ');
title('Quantized Signal x_{q}[nT]');
axis([0.01 0.022 -0.05 0.02]) ;

subplot(3,1,3)
err = sampled-quantized;                % Error Signal calculation
plot(nT,err);                           % Error Signal plot of same 500 samples
yline(0,'--');
xlabel('Discrete Time [nT] in seconds');
ylabel('Error level');
title('Error Signal err[nT] = x[nT]-x_{q}[nT]');
axis([0.01 0.022 -0.01 0.01]) ;

txt = ['Time period T: ' num2str(T) ' seconds'];
text(nT(end),-0.015,txt,'FontSize',10,'FontWeight','bold');
```

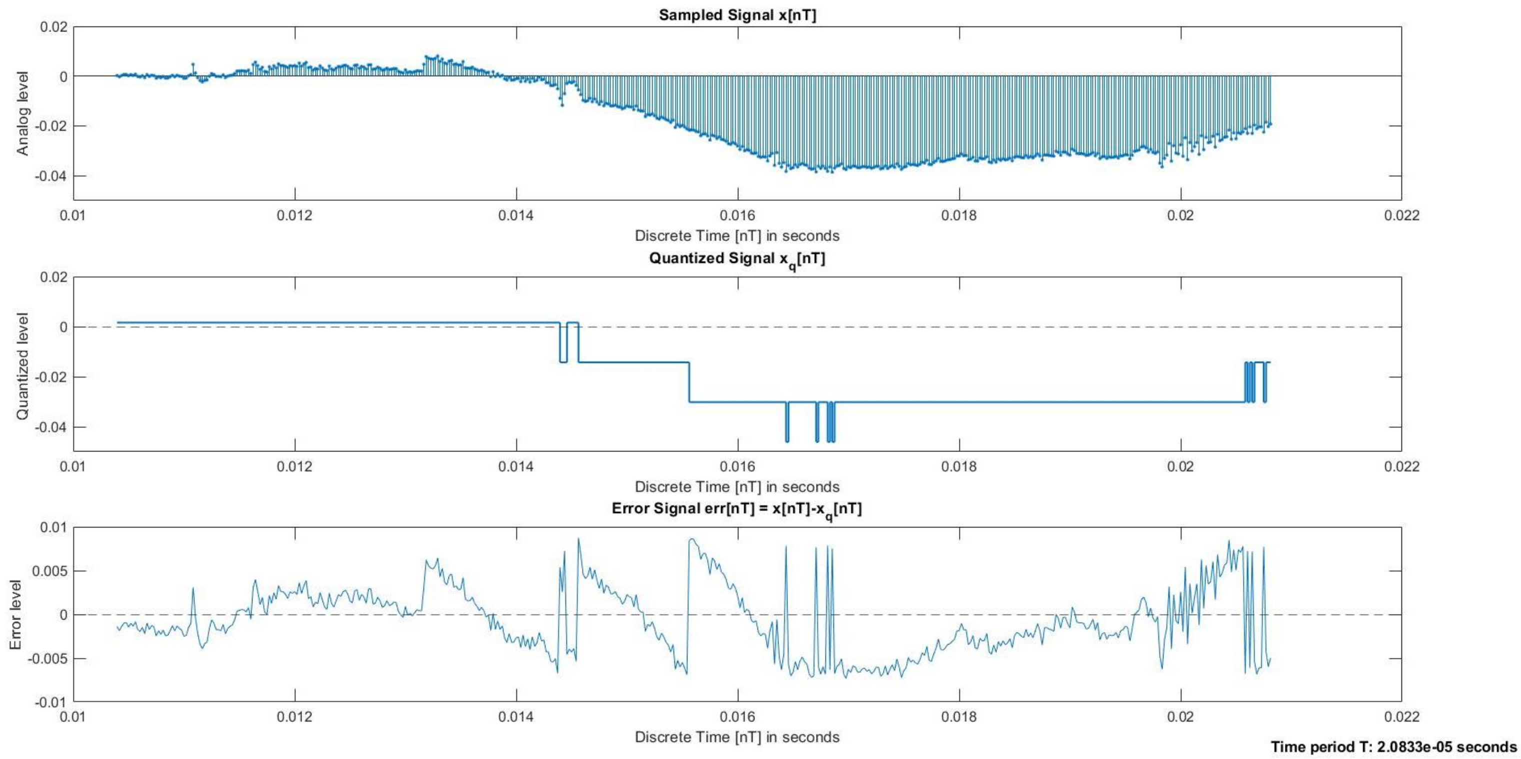
```
% PART B : SIGNAL RECONSTRUCTION


% Filter Design
order = 7;              % Order of the FIR Low-pass filter used for reconstruction of speech signal from
                        % Quantized signal
cutoff = 5000/(fs/2);   % Cut-off frequency given is 5khz
firLow = fir1(order, cutoff, kaiser(order+1,1)); % FIR Lowpass filter with kaiser-windowed function having
outlow = filter(firLow,1,Quantized);             % Filtered / Reconstructed Signal from Quantized signal


% Signal Plot
figure('NumberTitle', 'off', 'Name', 'Signal Reconstruction using 7th Order FIR Filter');
subplot(4,1,1);                         % Quantized signal plot of 500 samples
stairs(nT,quantized,'Linewidth',1.5);
yline(0,'--');
xlabel('Discrete Time [nT]');
ylabel('Quantized level');
title('Quantized Signal x_{q}[nT]');
axis([0.01 0.022 -0.05 0.02]) ;

subplot(4,1,2);                         % Reconstructed signal plot of same 500 samples
plot(nT,outlow(500:1000),'Linewidth',1.5,'Color','r');
yline(0,'--');
xlabel('Discrete Time [nT]');
ylabel('Reconstructed level');
title('Reconstructed Signal r_{q}[nT] = h[nT] \oplus x_{q}[nT] ');
axis([0.01 0.022 -0.05 0.02]) ;
txt = ['Time period T: ' num2str(T) ' seconds'];
text(nT(end),-0.07,txt,'FontSize',8,'FontWeight','bold');


subplot(4,1,3);                         % Power Spectrum of the Quantized Speech Signal
                                        % (over entire speech duration)
p = pspectrum(Quantized);
normalised = linspace(0,1,length(p));
plot(normalised,pow2db(p),'LineWidth',1.5);
xlabel('Normalised Frequency \omega (x \pi radians/sample)')
ylabel('Power Spectrum (in dB)');
title('Power Spectrum of Quantized Signal X_{q}[\omega]')

subplot(4,1,4);                         % Power Spectrum of the Reconstructed Speech Signal
                                        % (over entire speech duration)
p = pspectrum(outlow);
normalised = linspace(0,1,length(p));
plot(normalised,pow2db(p),'LineWidth',1.5,'Color','r');
xlabel('Normalised Frequency \omega (x \pi radians/sample)')
ylabel('Power Spectrum (in dB)');
title('Power Spectrum of Reconstructed Signal R_{q}[\omega] = H[\omega] \times X_{q}[\omega] ')

txt = ['Sampling Frequency fs: ' num2str(fs) ' Herts'];
text(0.9,-180,txt,'FontSize',8,'FontWeight','bold');
```

```matlab
% Filter-Response using freqz() function
fig = figure('NumberTitle', 'off', 'Name', '7th Order FIR Filter');

% Impulse-Response of Filter
subplot(3,2,1);
[h, w] = freqz(firLow);
magnitude = pow2db(abs(h));
plot(w/pi,magnitude,'LineWidth',1.2);            % Magnitude Response
title('7^{th}order FIR Lowpass Filter H[\omega]');

subplot(3,2,2);
phase = unwrap(angle(h));
plot(w/pi,rad2deg(phase),'LineWidth',1.2);      % Phase Response
title('7^{th}order FIR Lowpass Filter H[\omega]');

% Quantized Signal Frequency-Response
subplot(3,2,3);
[h, w] = freqz(Quantized);
magnitude = pow2db(abs(h));
plot(w/pi,magnitude,'LineWidth',1.2,'Color','m');           % Magnitude Response
title('Quantized Signal X_{q}[\omega]');
txt = ['Magnitude (db) \rightarrow'];
th = text(-0.1,-20,txt,'FontSize',14,'FontWeight','bold');
set(th,'Rotation',90);

subplot(3,2,4);
phase = unwrap(angle(h));
plot(w/pi,rad2deg(phase),'Color','m','LineWidth',1.2);      % Phase Response
title('Quantized Signal X_{q}[\omega]');
txt = ['Phase (degrees) \rightarrow'];
th = text(-0.1,-40,txt,'FontSize',14,'FontWeight','bold');
set(th,'Rotation',90);

% Filtered Signal Frequency-Response
subplot(3,2,5);
[h, w] = freqz(outlow);
magnitude = pow2db(abs(h));
plot(w/pi,magnitude,'LineWidth',1.2,'Color','r');           % Magnitude Response
title('Filtered/Reconstructed Signal H[\omega]\times X_{q}[\omega]');

subplot(3,2,6);
phase = unwrap(angle(h));
plot(w/pi,rad2deg(phase),'Color','r','LineWidth',1.2);      % Phase Response
title('Filtered/Reconstructed Signal H[\omega]\times X_{q}[\omega]');

txt = ['Normalised Frequency \omega (\times \pi radians/sample) \rightarrow'];
text(-0.5,-1000,txt,'FontSize',14,'FontWeight','bold');

txt = ['Sampling Frequency fs: ' num2str(fs) ' Herts'];
text(0.9,-900,txt,'FontSize',8,'FontWeight','bold');

sgtitle('Frequency Response','FontWeight','bold');
```

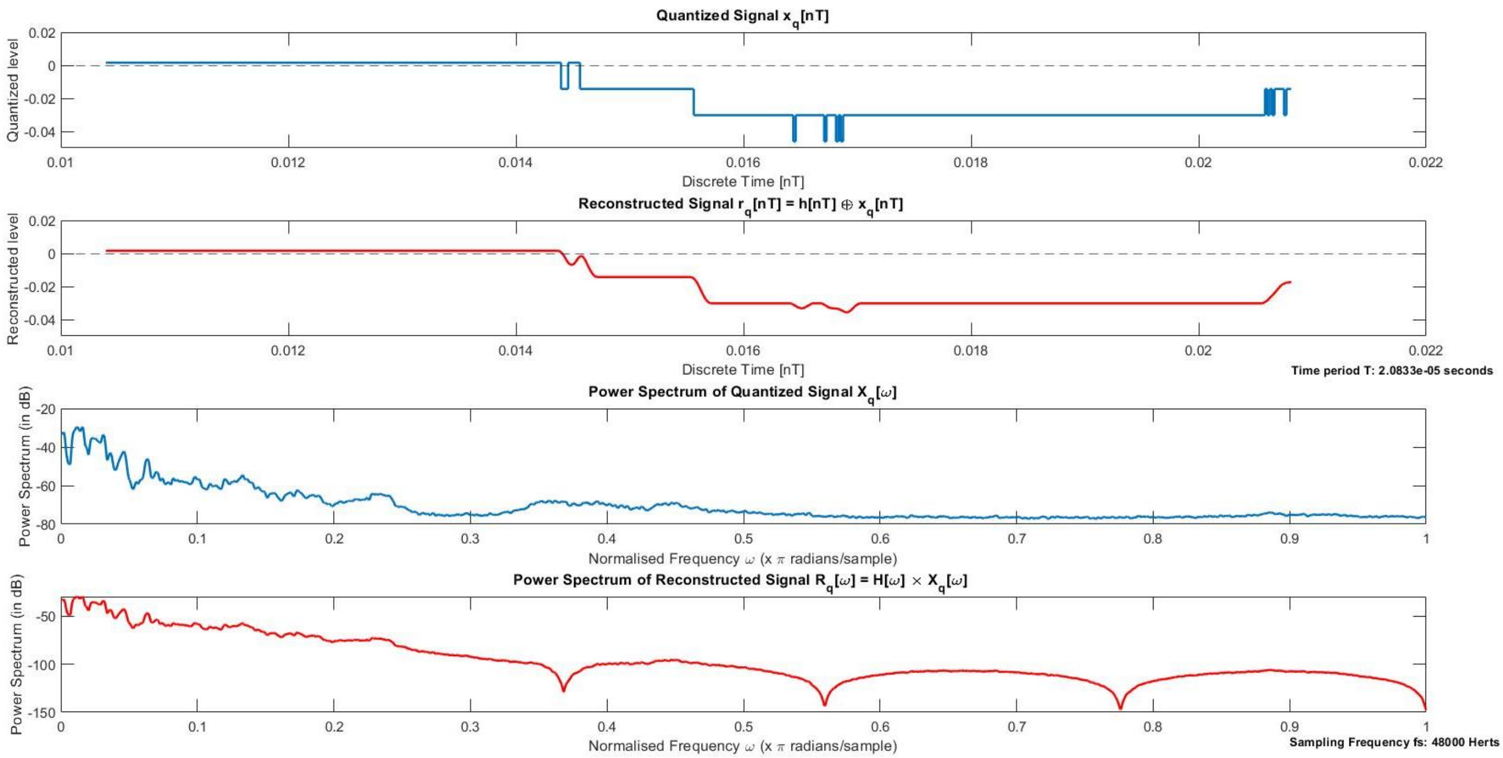
```
% QUALITY-ASSESSMENT

 sound(signal,fs);           % Original Speech Signal -- Excellent
 sound(Quantized,fs);        % Quantized Signal       -- Fair
 sound(outlow,fs);           % Reconstructed Signal   -- Good
```

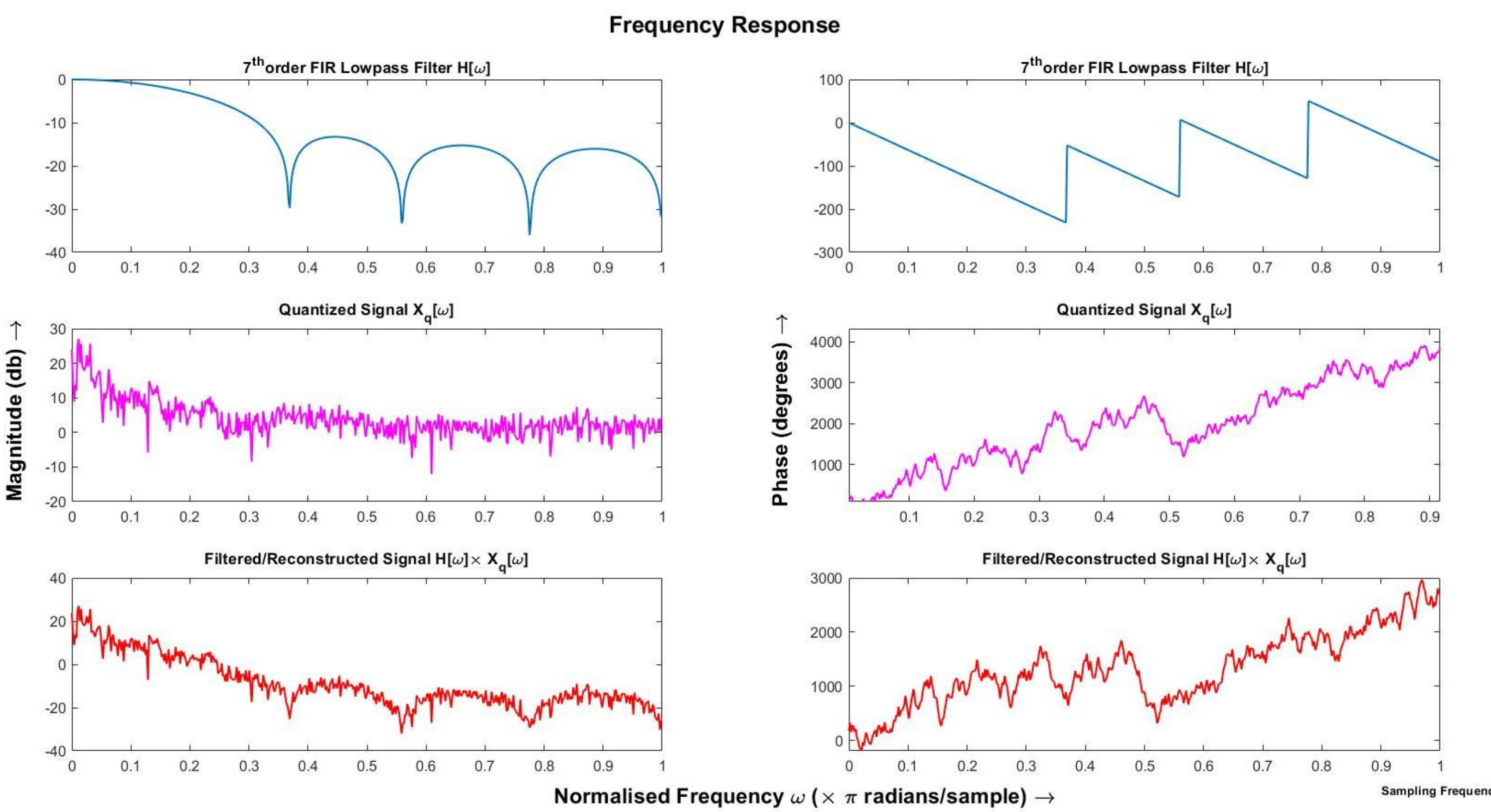
Result:

(a). Part a: Quantization Error



(b). Part b: Signal Reconstruction





Discussion:

While performing this Experiment, I made slight changes with the procedure, by instead of taking 20 samples I extracted 500 samples which gave a much better idea of the speech signal, since first 20 samples had all zero values.

I quantized the speech signal with a 6-bit Quantizer (in previous experiments I used a 4-bit Quantizer) for better sound quality.

For reconstruction of speech signal from quantized signal, I used 7^{th} order FIR lowpass filter for smoothing out the quantized signal and further remove high-frequency components. To implement the filter in MATLAB I used fir1() function, The cut-off frequency for the filter I designed is 5khz. Since the Sinc function (Fourier-transform of a rectangular pulse) extends to infinity in both sides, it results in an infinite number of taps of the FIR filters. Then such filters are not practical and the number of taps must be truncated. This is accomplished by using windows. In my case, I experimented with hamming, gaussian, kaiser windows, and observed kaiser window to give the best normalized impulse response within the given constraints, so for my final output, I used kaiser window with parameter alpha = 1 and window size = order (7) +1 as my windowing function.

By plotting the frequency response of the filter, I also observed in magnitude response, the transition from the passband to stopband is steeper as the order of the filter is increased i.e., for every increment by one in order, the transition width was reduced by half and the stopband attenuation came out to be around 12db for order of 7. The phase response is linear, as expected of the FIR filter

For checking the quality of the reconstructed signal, I played the original, quantized, reconstructed signal one by one to compare the sounds and noticed a slight improvement of the reconstructed signal over quantized one. I consider the quality of the signal to be as follows:

1	2	3	4	5
Bad	Poor	Fair	Good	Excellent
		Quantized	Reconstructed	Original