**SAMPLING:**

%Input Signals(**code** generated a digital signal (**x**) of 2800 Hz with sampling frequency of 8 kHz. Then frequency component of that signal is drawn in the first subplot.)

clc

clear all

t=1:4000;

x=sin(2\*pi\*2900\*t/8000); %Input Signal Frequency = 2900Hz % Sampling Frequency = 8000Hz

figure(1)

subplot(4,1,1);

t1=-4000:2:4000-2;

plot(t1,abs(fftshift(fft(x))));

title('Original Signal in frequency domain');

%2-factor linear down-sampling and reconstruction or sampling(down-sampled signal is reconstructed using interpolation. The frequency components of the reconstructed signal is shown in the second subplot.)

l=1;

for i=1:length(x)

if (mod(i-1,2)~=1)

xt(l)=x(i);

xin(l)=i;%index

l=l+1;

end

end

k=1:length(x);

pp = interp1(xin,xt,'spline','pp');

yi = ppval(pp,k);

subplot(4,1,2);

plot(t1,abs(fftshift(fft(yi))));

title('Folded signal, after 2-factor linear down-sampling and reconstruction');

%1.5-factor non-linear down-sampling(Two noise signal is introduced. One is of 2533 Hz frequency (folding noise), another one is of 133 Hz (low-frequency noise) frequency.)

l=1;

for i=1:length(x)

if (mod(i-1,3)~=1)

xt(l)=x(i);

xin(l)=i;%index

l=l+1;

end

end

pp = interp1(xin,xt,'spline','pp');

yi = ppval(pp,k);

subplot(4,1,3);

plot(t1,abs(fftshift(fft(yi))));

title('Signal, after 1.5-factor non-linear down-sampling and interpolation reconstruction');

%Getting Low freq Noise(Two noise signal is introduced. One is of 2533 Hz frequency (folding noise), another one is of 133 Hz (low-frequency noise) frequency.)

b=[1 -2 1];%Highpass filter fc=400Hz//266.6Hz

a=[1 -1.5610180758007182 .64135153805756318];

yi1= filter(b,a,yi);

% sound(yi1,8000)

yi1=yi-yi1;%LOWPASS

%Signal of Folding Frequency(**code** generates a signal **x2**of 2666.67 Hz, as folding frequency after 1.5 factor down-sampling **Ffl = 1.5\* Fs/2**)

t=1:length(yi);

x2=sin(2\*pi\*2666.666666666667\*t/8000);

%Noise Elimination(**code** shifts **yil** in the frequency domain by **Ffl** and adds shifted signal **x2.\*yi1** with reconstructed signal **yi** to eliminate high-frequency noise component. Finally, low-frequency components are eliminated from **yi**by high pass filter.)

yi=yi+x2.\*yi1;

yi= filter(b,a,yi);

yi= filter(b,a,yi);

subplot(4,1,4);

plot(t1,abs(fftshift(fft(yi))));

title('After noise Elimination of Previous signal');

**QUANTIZATION:**

t = [0:.1:2\*pi]; % Times at which to sample the sine function

x=sin(2\*pi\*2900\*t/8000)

L = length(x)

sig = sin(t); % Original signal, a sine wave

partition= [-.5 0 .5]; % The vector of endpoints of the partition intervals

codebook=[-0.75 -0.25 0.25 0.75];%The vector of output values assigned to each partition

[index,quants] = quantiz(sig,partition,codebook); % Quantize.

plot(t,sig,'x',t,quants,'.')

legend('Original signal','Quantized signal');

axis([-.2 7 -1.2 1.2])