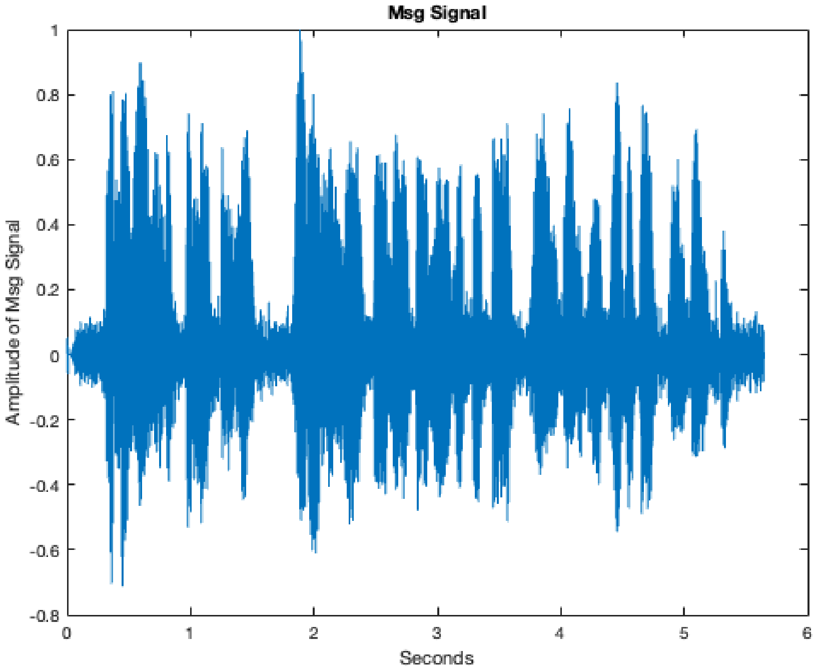
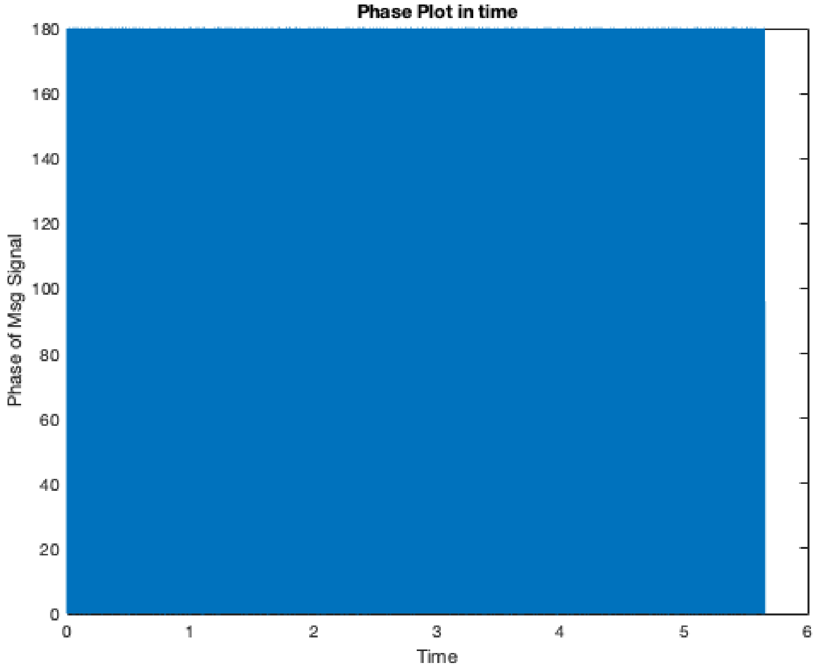
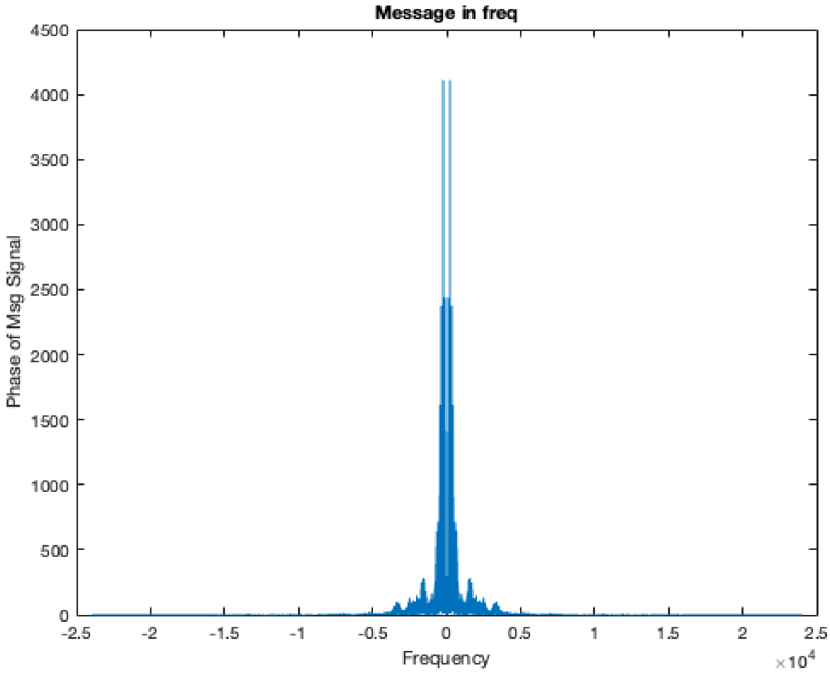
Contents

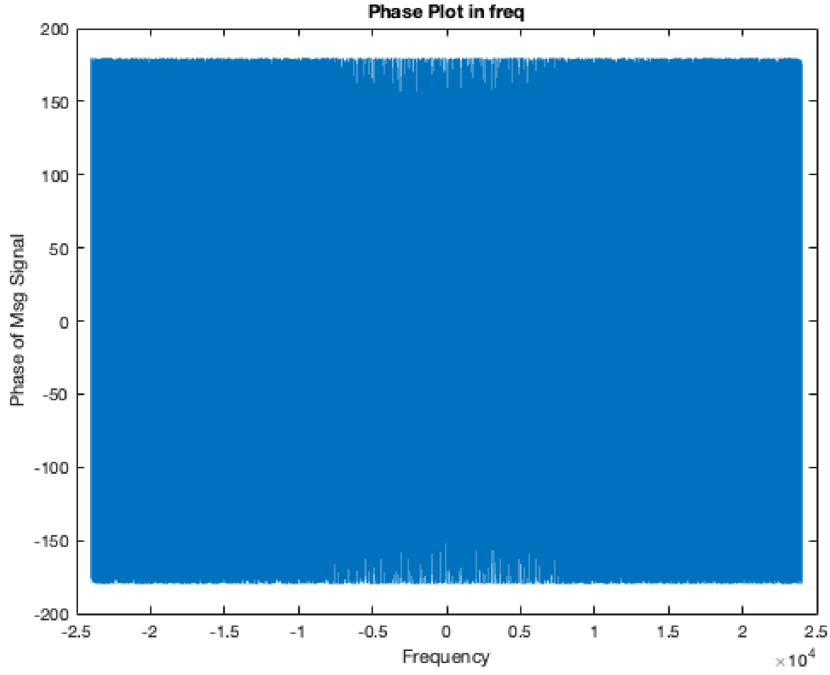
```
% Assignment 3
% Pallav Singla
% 2020225


[y, fs] = input_2020225();

% sound(y,Fs)
                           %
t = 0:(1/fs):(length(y)*(1/fs))-(1/fs);             % Time for ploting purpose
figure(1)
plot(t,y);
title('Msg Signal');                    % Plottingh the message signal
xlabel('Seconds');
ylabel('Amplitude of Msg Signal');

% figure
% plot(t,abs(y));
% title('Amplitude Plot');
% xlabel('Seconds');

figure(2)
plot(t,angle(y)*180/pi);
title('Phase Plot in time');        % plot of phase in time domain
xlabel('Time');
ylabel('Phase of Msg Signal');

len = length(y);
Y = abs(fftshift(fft(y)));              % frequency representation of message signal
F = (-(1-1/len)/2:1/len:(1-1/len)/2)*fs;
figure(3);
plot(F,Y);
title('Message in freq');
xlabel('Frequency');
ylabel('Phase of Msg Signal');

figure(4)
plot(F,angle(fft(y))*180/pi);           %phase in Frequency domain
title('Phase Plot in freq');
xlabel('Frequency');
ylabel('Phase of Msg Signal');

% used for displaying the frequency which is end point in the frequency
% plot

display_2020225();                   % Display Function for displaying the max freq

info = audioinfo('rec2.m4a');

f = 5000;     % max frequency of msg signal


% [k,n] = size(y);
```

```
        5000
```


Msg Signal
Amplitude of Msg Signal
Seconds
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6

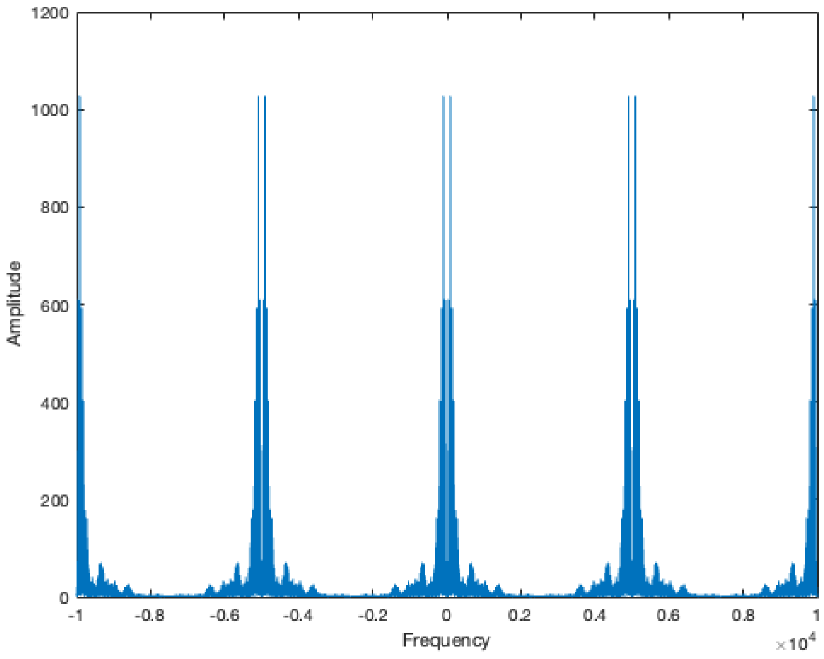

Phase Plot in time
Phase of Msg Signal
Time
180
160
140
120
100
80
60
40
20
0
0
1
2
3
4
5
6





for FS greater then 2FM

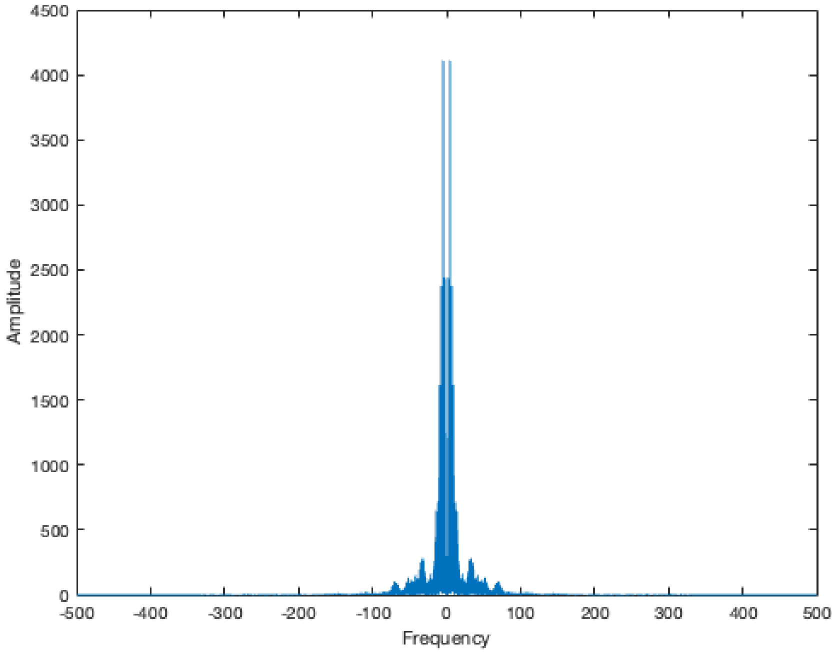
```
fs1 = 2*f*2;    % Sampling frequency
fac = 2;
% %
tr = zeros(size(t));      % Making Matrix and Assigning 0 to each value
tr(1:fs1/f:end) = 1;           % impulse train

trr = transpose(tr);        % taking transpose of the impulse train
Z = y.*trr;
ZZ = fftshift(fft(Z));       % sampled output

freq1 = (-(1-1/len)/2:1/len:(1-1/len)/2)*fs1;

figure(5);

plot(freq1,abs(ZZ));        % plot of sampled output wrt to freq  %https://in.mathworks.com/matlabcentral/answers
xlabel('Frequency');
ylabel('Amplitude');
```

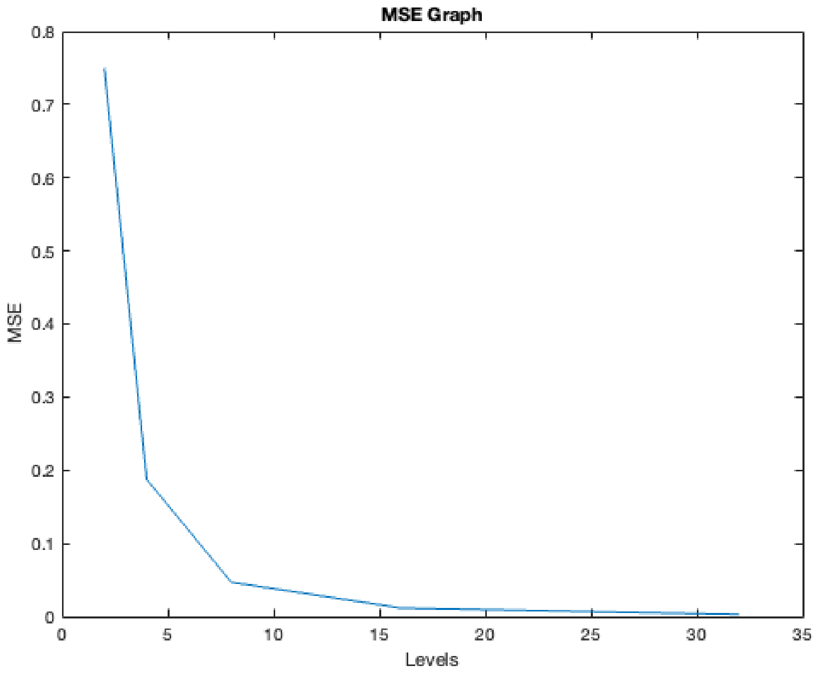


For FS less then 2FM

```
fs2  =2*f*(1/10);       % Samplin g Freq

tr2 = zeros(size(t));       % Making Matrix and Assigning 0 to each Value
tr2(1:fs2/f:end) = 1;       % impulse train

tr2r = transpose(tr2);      % taking transpose of the impulse train
Z1 = y.*tr2r;

freq = (-(1-1/len)/2:1/len:(1-1/len)/2)*fs2;    % Frequency same as above

Z2 = fftshift(fft(Z1));             % Sampled Output

figure(6);

plot(freq,abs(Z2));
xlabel('Frequency');
ylabel('Amplitude');


%%part4 quantisation

quantise_2020225();     %% quantise fuction made in function


L =32;  %% Because MSE is tending to zero( almost) at the end i have seen it from graph of MSE given
          % code in quantise_2020225()
          % mse  = 3*1/32*32  it is very very saml1 almost to 0

mse_32 = 3*1/(32*32);       % for showing the value at 32
disp(mse_32);
```

```
Warning: Integer operands are
required for colon operator when
used as index.
    0.0029
```

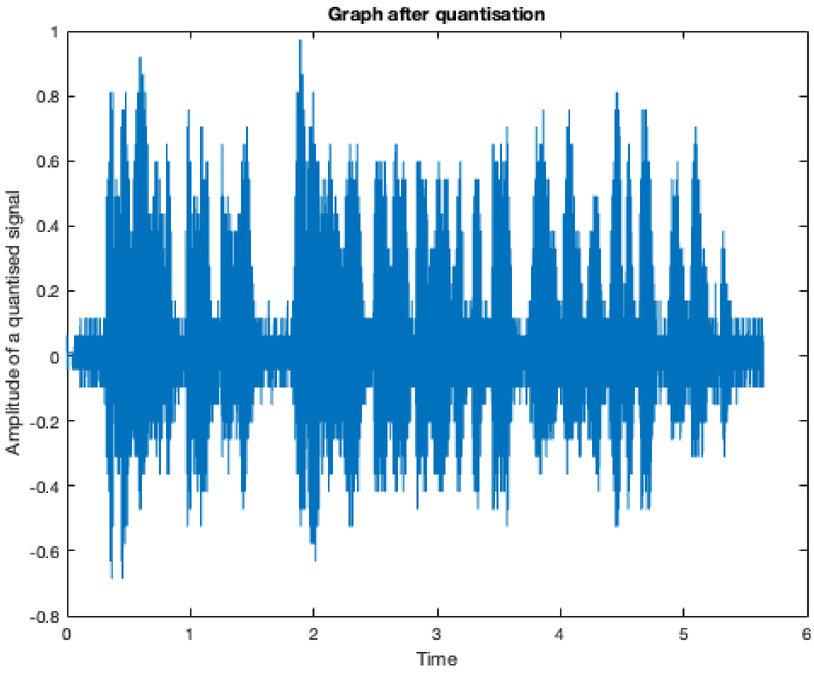


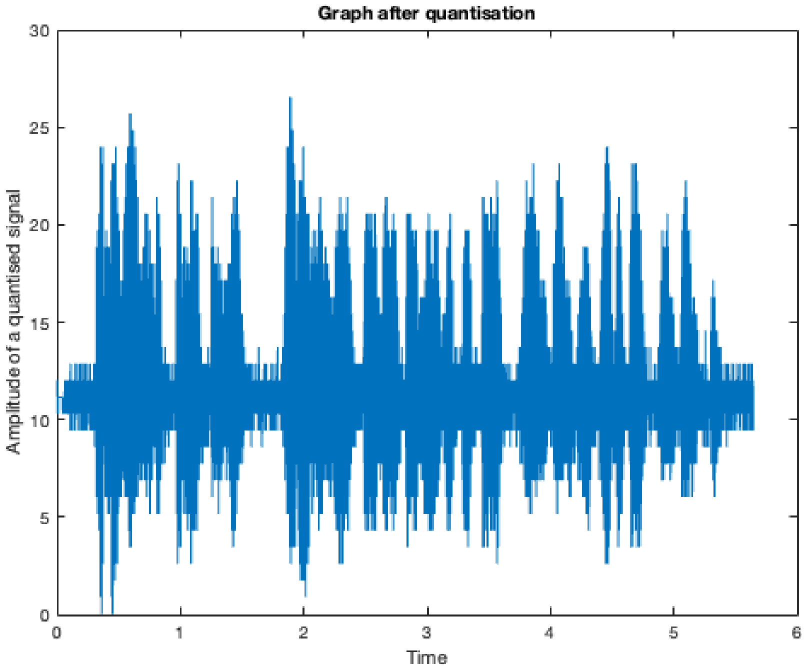


Quantisation

```
% % s = quantisation_2020225(Z1);
% % plot(t,s*32/1.8);
% %
% % xlabel('Time');
% %
% % ylabel('Amplitude of a quantised signal');
% % title('Graph after quantisation');


L=32;                               % Number of Levels
[a,b] = quanta_2020225(Z1,L);   % using the function made in and output quantised signal is a

figure(8)                       % plotting the quantised signal
plot(t,a)
xlabel('Time');
ylabel('Amplitude of a quantised signal');
title('Graph after quantisation');
```

```
figure(9);
plot(t,(a*32/2)+11);             % Plotting the number of level vs time graph
xlabel('Time');
ylabel('Amplitude of a quantised signal');
title('Graph after quantisation');
```





part 5

```
enc = dec2bin((a*32/2)+11);    % COnverting decimal to binary using inbuild function

%Here as I have to encode it so for this i have to use matlab inbuilt
%funcition de2bi
% n=5;
% m = dec2bin(s);
% so = 1;
%
% for kl=1:31                         % for iterating in the rows
%     for jp=1:n
%
%         mi(so) = m(kl,jp);              % Here converting into row vector
%
%         so=so+1;
%         jp=jp+1;
%
%     end
%     kl=kl+1;
% end
% Li =0:1:L-1;



%
% Array = ['000','001','010','011','100','101','110','111'];
%
% mp = 1;
%
% delv = 1/4;
%
% v1 = -1;
% v2 = v1+1/4;
% v3 = v2+1/4;
% v4 = v3+1/4;
% v5 = v4+1/4;
% v6 = v5+1/4;
% v7 = v6+1/4;
% v8 = v7+1/4;
%
% qy = zeros(size(y));
%
% for i=1:length(y)
%     if v1<=y(i) || y(i)<v2
%         qy(i) = v1;
%     elseif v2<=y(i) || y(i)<v3
%         qy(i) = v2;
%     elseif v3<=y(i) || y(i)<v4
%         qy(i) = v3;
%     elseif v4<=y(i) || y(i)<v5
%         qy(i) = v4;
%     elseif v5<=y(i) || y(i)<v6
%         qy(i) = v5;
%     elseif v6<=y(i) || y(i)<v7
%         qy(i) = v6;
%     elseif v7<=y(i) || y(i)<v8
%         qy(i) = v7;
%     else
%         qy(i) = v8;
%     end
%
% end
%
% plot(qy);
```

Published with MATLAB® R2021a