EE6133: Experiment 1
Upsampling and Downsampling
Date: October 14, 2021

1 Part 1

Block Diagram of Operation:

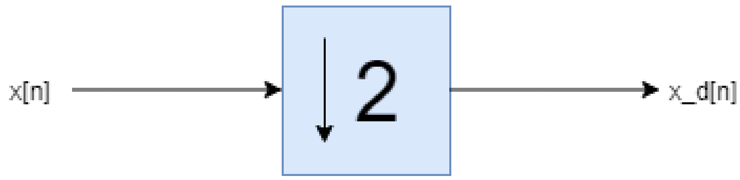

Figure 1: Block Diagram for the Downsampling Operation

Matlab Code:

```
%question 1
% Downsampling without AA filtering

path = fullfile('Audio_Files','speech8khz.wav') %path where the audio files are hosted

[x fs]=audioread(path); %speech data or music data depending on file

%Original magnitude spectrum
n_dtft=2^(ceil(log2(length(x)))); %number of point in DTFT
X = fftshift(fft(x,n_dtft));
f_dtft = linspace(-1,1,n_dtft); % freq
figure();
q1_plt1 = plot(f_dtft,abs(X)); %plotting magnitude spectrum
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the speech signal');
x1 = [0.8 0.6];
y1 = [0.5 0.8];
annotation('textarrow',x1,y1,'String','Peak')
saveas(q1_plt1,'q1_plt1.png','png');


%Downsampling signal by 2
x_d = downsample(x,2);
```

```
n_dtft = 2^(ceil(log2(length(x_d))));
f_dtft = linspace(-1,1,n_dtft);
X_d = fftshift(fft(x_d,n_dtft));

figure();
q1_plt2 = plot(f_dtft,abs(X_d));  %downsampled plot
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the DS-by-2 speech signal');
x1 = [0.8 0.65];
y1 = [0.5 0.7];
annotation('textarrow',x1,y1,'String','Peak')
saveas(q1_plt2,'q1_plt2.png','png');


%generating audio file
audiowrite('speech_ds_by2.wav',x_d,fs/2)
```

Plots for the Speech Signal:

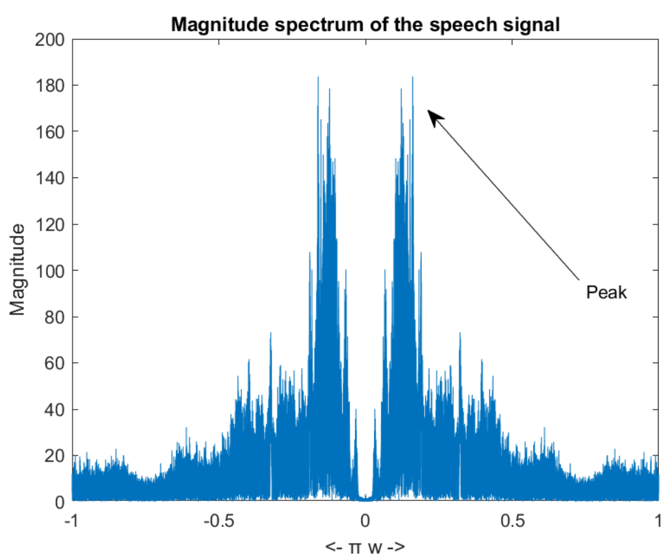

Figure 2: Magnitude Spectrum of the Speech Signal

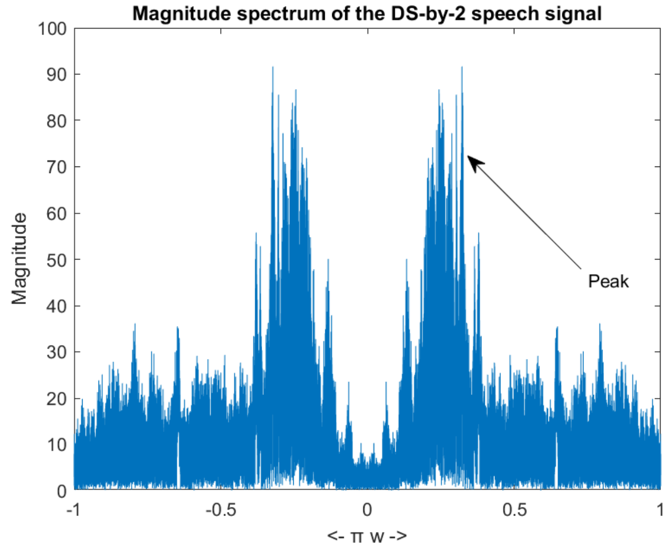

Figure 3: Magnitude Spectrum of the DS-by-2 Speech Signal

Plots for the Music Signal:

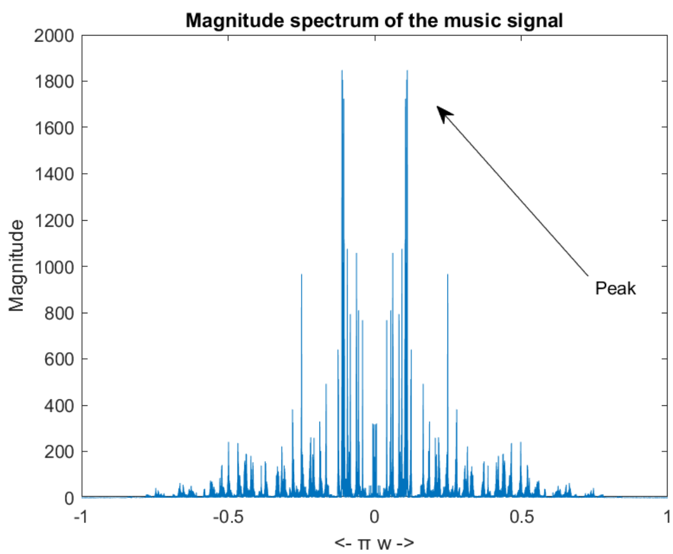

Figure 4: Magnitude Spectrum of the Music Signal

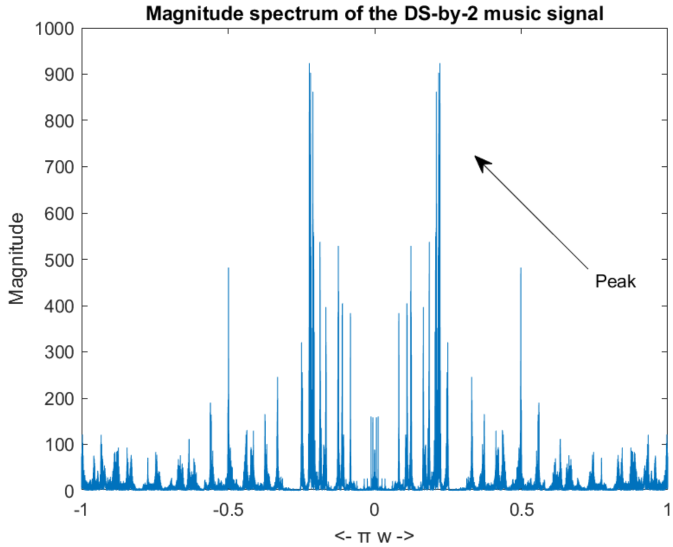

Figure 5: Magnitude Spectrum of the DS-by-2 Music Signal

Observations
If we downsample a signal x by M, we know that,

$$\underline{X_d}(w) = \frac{1}{M}\sum_{k=0}^{M-1}\underline{X}\left(w - \frac{2\pi k}{M}\right) \tag{1}$$

We observe this in practice here. As the downsampled spectrum is an average of several images, we see that the portions in the spectrum which were originally not busy are now relatively busier.

2 Part 2

Block Diagram of Operation:

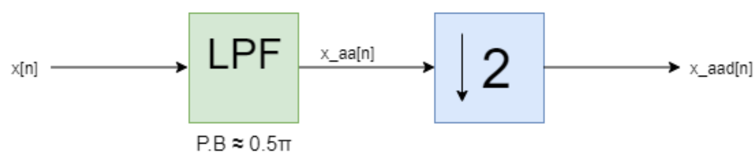

Figure 6: Block Diagram for the Downsampling Operation after anti-aliasing filter

Matlab Code:

```
%question 2
% Downsampling by 2 using an AA filter

%creating the required LPF/AA filter

rp = 0.01;          % Passband ripple in dB
rs = 20;            % Stopband ripple in dB
fs = 8000;          % Sampling frequency, 16kHz for music
f = [1800 2200];     % LPF cutoff for question 2
```

```
%f = [880,1120]; % use this and comment the above for question 3
a = [1 0];        % Desired amplitudes

dev = [(10^(rp/20)-1)/(10^(rp/20)+1) 10^(-rs/20)];
[n,fo,ao,w] = firpmord(f,a,dev,fs);
b = firpm(n,fo,ao,w);
n_dtft = 1024
f_dtft = linspace(-1,1,n_dtft);
B = fftshift(fft(b,n_dtft));


%plot for the filter magnitude

figure();
q1_plt_3 = plot(f_dtft,abs(B));  %filter magnitude response
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the equiripple LPF');
dim = [.2 .3 .3 .3];
str = {'wp = 0.45pi', 'ws = 0.55pi'};
annotation('textbox',dim,'String',str,'FitBoxToText','on');
saveas(q1_plt_3,'q1_plt_3.png','png');


%Passing the input signal through the AA Filter above,

x_aa  = filter(b,1,x)
x_aad = downsample(x_aa,4) %downsample by 2 for question 2, downsample by 4 for question 5
n_dtft = 2^(ceil(log2(length(x_aad))));
f_dtft = linspace(-1,1,n_dtft);
X_aad = fftshift(fft(x_aad,n_dtft));

%magnitude spectrum of signal after AA and DS by 2
figure();
q1_plt3 = plot(f_dtft,abs(X_aad));  %downsampled plot
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the DS-by-2 speech signal with AA filter');
x1 = [0.8 0.65];
y1 = [0.5 0.7];
annotation('textarrow',x1,y1,'String','Peak')
```

```
saveas(q1_plt3,'q1_plt3.png','png');

%generating audio file
audiowrite('speech_ds_by2_aa.wav',x_aad,fs/2)
```

Plots for the AA Filter:

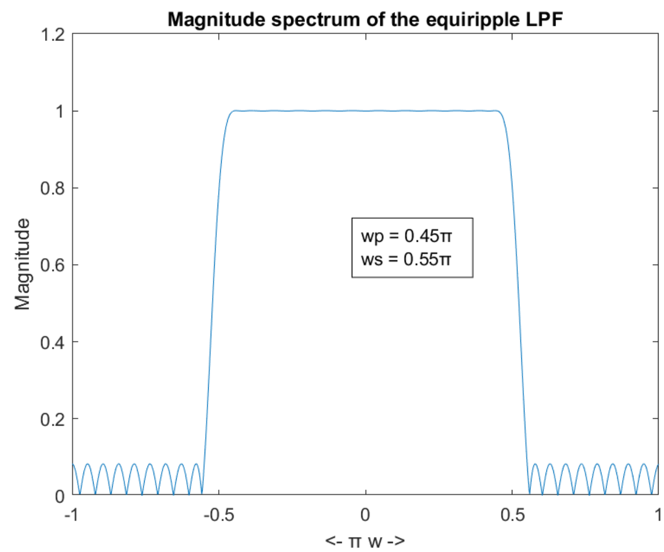

Figure 7: Filter Response of the Anti Aliasing Filter

Plots for the Speech Signal:

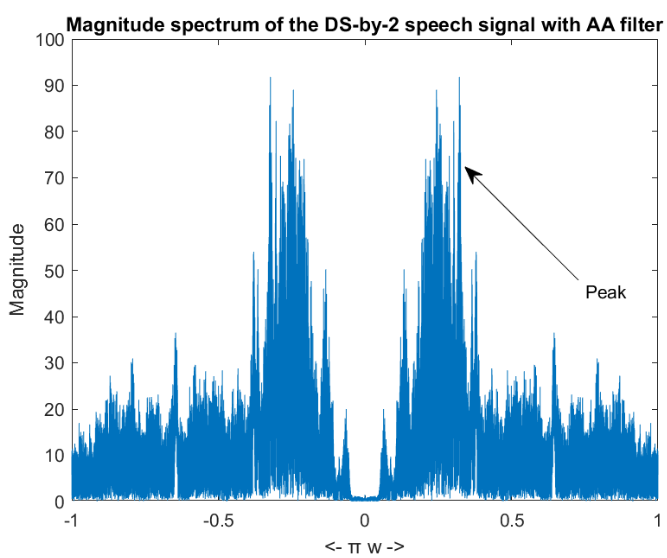

Figure 8: Magnitude Spectrum of the downsampled Speech Signal with anti aliasing

Plots for the Music Signal:

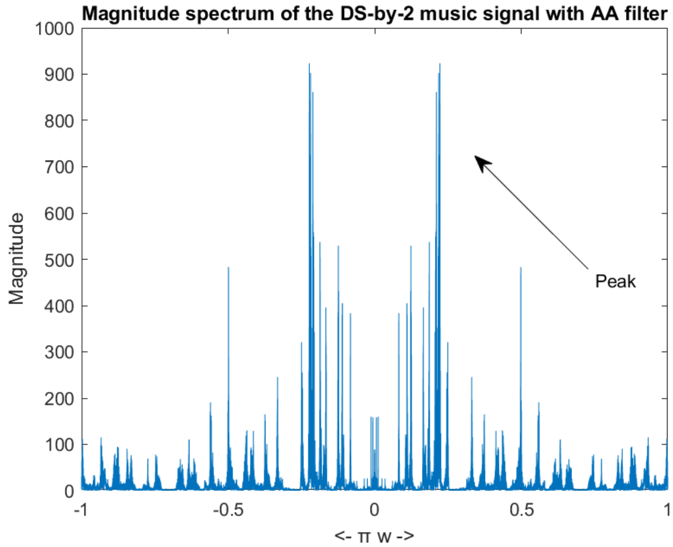

Figure 9: Magnitude Spectrum of the downsampled Music Signal with anti aliasing

2.1 Audio Comparison

Comparing the audio obtained in the first two subdivisions, we see that,

- Both the audios aren't as good as the original one for both music and speech, which is expected, as downsampling reduces the sampling frequency.
- For the audio signal which has been downsampled without anti aliasing, we see that there are some artifacts present in the audio, which distort the downsampled audio signal.This is especially problematic and perceptible when the volume or intensity of the audio is very less.This confirms what we'd seen in theory, as the downsampling operation results in aliasing. As a consequence, the audio signal isn't very *sharp*.
- For the audio signal which has been downsampled with an anti aliasing filter, we see that the artifacts which were present earlier are no longer there resulting in a much clearer audio.Thus, anti aliasing filters help prevent aliasing and make the audio sharper.

3 Part 3

Matlab Code:

```
%question 3

%plot for the filter magnitude

figure();
q1_plt_4 = plot(f_dtft,abs(B));  %filter magnitude response
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the equiripple LPF');
dim = [.2 .3 .3 .3];
str = {'wp = 0.22pi', 'ws = 0.28pi'};
```

```
annotation('textbox',dim,'String',str,'FitBoxToText','on');
saveas(q1_plt_3,'q1_plt_4.png','png');
```

Plots for the AA Filter:

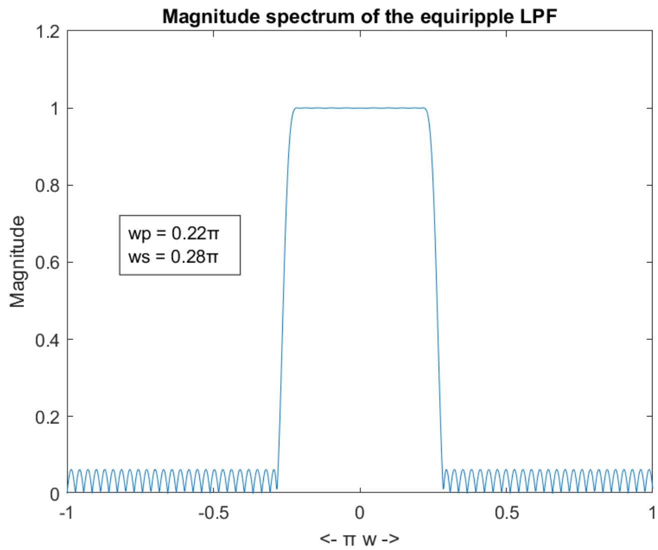

Figure 10: Filter Response of the Anti Aliasing Filter

4 Part 4

Block Diagram of Operation:

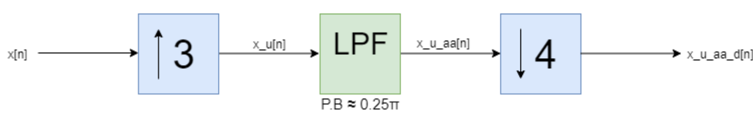

Figure 11: Block Diagram for the upsampling operation followed by downsampling

Matlab Code:

```
%question 4
x_u = upsample(x,3)
n_dtft = 2^(ceil(log2(length(x_u))));
f_dtft = linspace(-1,1,n_dtft);
X_u = fftshift(fft(x_u,n_dtft));

%plot of upsampled input
figure();
q1_plt4 = plot(f_dtft,abs(X_u));  %upsampled plot
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the US-by-3 speech signal');
```

```
x1 = [0.85 0.8];
y1 = [0.4 0.6];
annotation('textarrow',x1,y1,'String','Peak-3')
x1 = [0.2 0.25];
y1 = [0.4 0.6];
annotation('textarrow',x1,y1,'String','Peak-1')
x1 = [0.6 0.55];
y1 = [0.4 0.6];
annotation('textarrow',x1,y1,'String','Peak-2')
saveas(q1_plt4,'q1_plt4.png','png');

x_u_aa   = filter(b,1,x_u) %passing the upsampled output into AA filter
x_u_aa_d = downsample(x_u_aa,4) %DS by 4
n_dtft = 2^(ceil(log2(length(x_u_aa_d))));
f_dtft = linspace(-1,1,n_dtft);
X_u_aa_d = fftshift(fft(x_u_aa_d,n_dtft));
 %plot of final output
figure();
q1_plt5 = plot(f_dtft,abs(X_u_aa_d));  % plot
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the output speech signal');
x1 = [0.8 0.65];
y1 = [0.5 0.7];
annotation('textarrow',x1,y1,'String','Peak')
saveas(q1_plt5,'q1_plt5.png','png');

%generating audio file
audiowrite('speech_u_d.wav',x_u_aa_d,3*fs/4)
```

Plots for the Speech Signal:

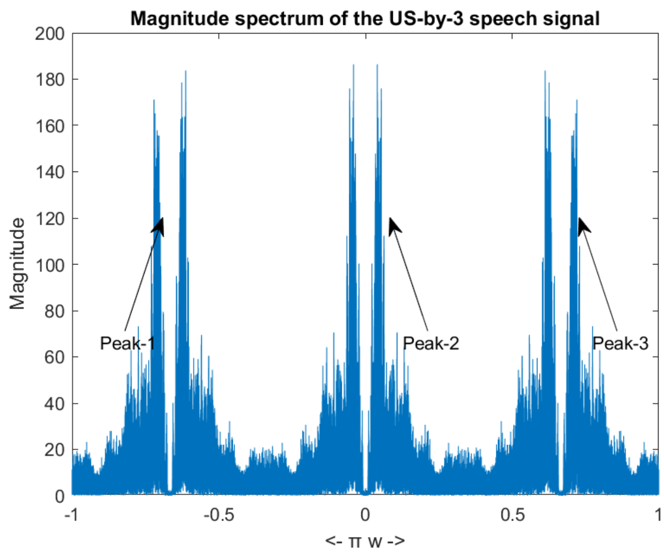

Figure 12: Magnitude Spectrum of the upsampled speech signal

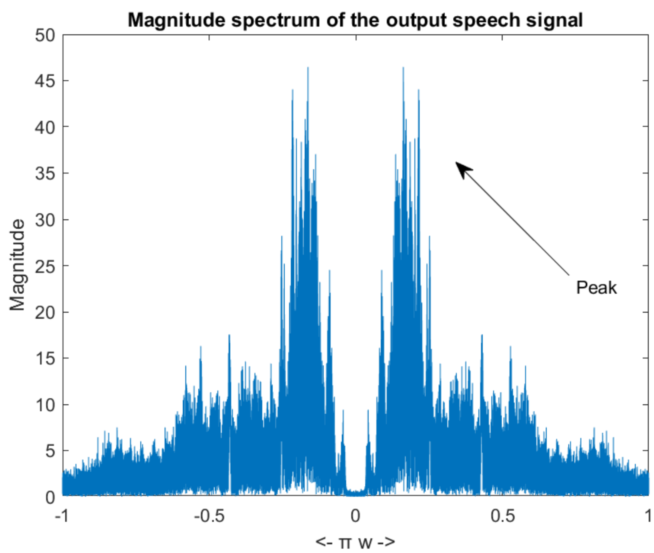

Figure 13: Magnitude Spectrum of the downsampled speech signal after upsampling

Plots for the Music Signal:

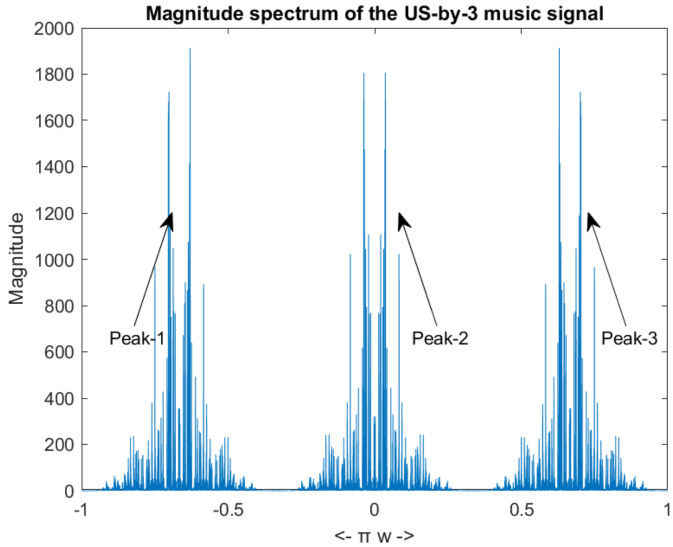

Figure 14: Magnitude Spectrum of the upsampled music signal

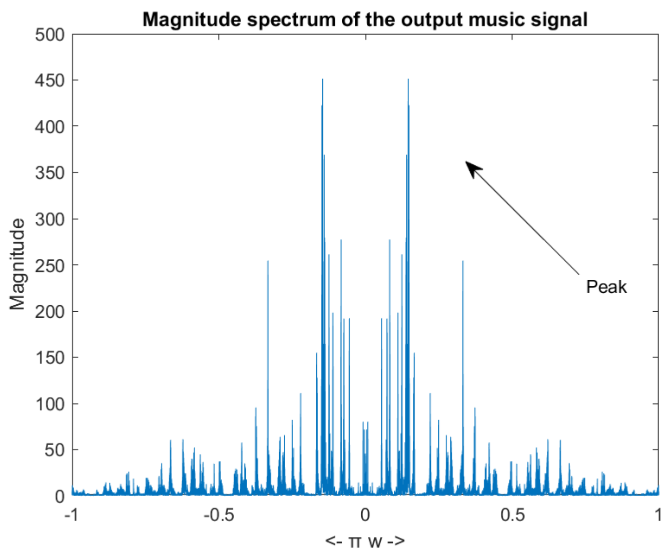

Figure 15: Magnitude Spectrum of the downsampled music signal after upsampling

5 Part 5

Block Diagram of Operation:

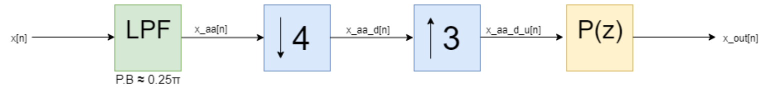

Figure 16: Block Diagram for the downsampling operation followed by upsampling

Matlab Code:

```
%Question 5
% Go to question 2 make the appropriate changes
```

```
x_aad_u = upsample(x_aad,3)
n_dtft = 2^(ceil(log2(length(x_aad_u))));
f_dtft = linspace(-1,1,n_dtft);
X_aad_u = fftshift(fft(x_aad_u,n_dtft));
%plot of the upsampled output after DS by 4

figure();
q1_plt6 = plot(f_dtft,abs(X_aad_u));  %upsampled plot
xlabel('<- pi w ->');
ylabel('Magnitude');
title('Magnitude spectrum of the US after DS speech signal');
saveas(q1_plt6,'q1_plt6.png','png');


h = intfilt(3,3,1); %interpolation filter after upsampling
n = linspace(-8,8,length(h));
figure();
q1_plt9 = plot(n,h);  %impulse response plot
xlabel('<- n ->');
ylabel('h[n]');
title('Impulse response of the interpolation filter');
saveas(q1_plt9,'q1_plt9.png','png');

H = fftshift(fft(h, 17));
f_dtft = linspace(-8,8,length(H));
figure();
q1_plt_9 = plot(f_dtft, abs(H));
xlabel("pi w");
ylabel("Magnitude");
title("Magnitude response of interpolation filter");
saveas(q1_plt_9,'q1_plt_9.png','png');

x_out = filter(h,1,x_aad_u) %interpolation
n_dtft = 2^(ceil(log2(length(x_out))));
f_dtft = linspace(-1,1,n_dtft);
X_out = fftshift(fft(x_out,n_dtft));
 %plot of the final output after interpolation

q1_plt7 = plot(f_dtft,abs(X_out));  %magnitude response plot
xlabel('<- pi w ->');
```

```
ylabel('Magnitude');
title('Magnitude spectrum of the output speech signal after interpolation');
dim = [.2 .3 .3 .3];
str = {'Image',' Rejection'};
annotation('textbox',dim,'String',str,'FitBoxToText','on');
saveas(q1_plt7,'q1_plt7.png','png');

%generating audio file
audiowrite('speech_d_u.wav',x_out,3*fs/4)
```

Plots for the Interpolation Filter:

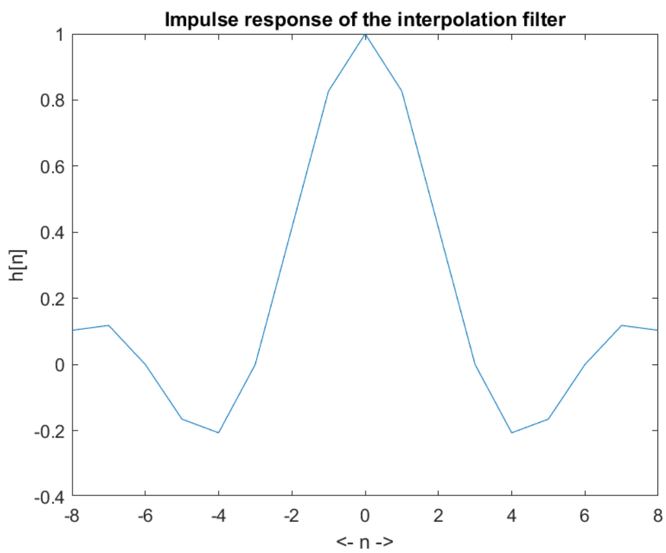

Figure 17: Impulse Response of the Interpolation Filter

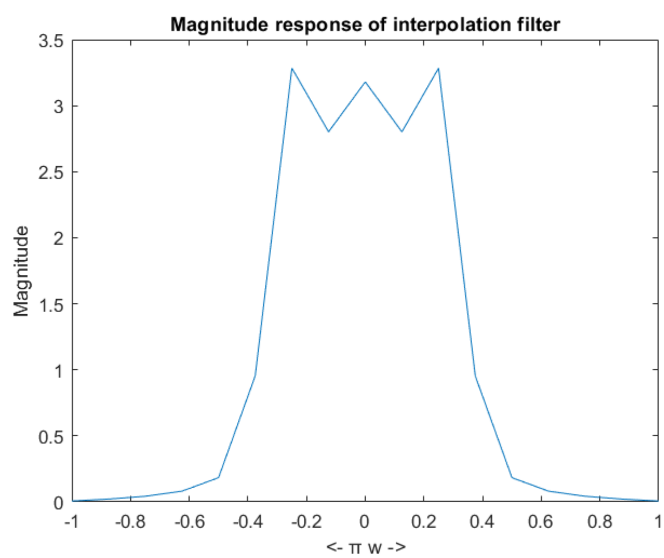

Figure 18: Magnitude Response of the Interpolation Filter

Plots for the Speech Signal:

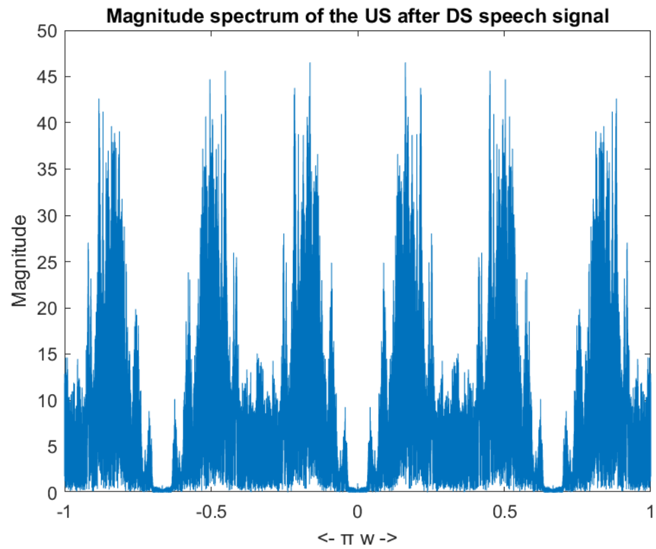

Figure 19: Magnitude Spectrum of the upsampled speech signal after downsampling

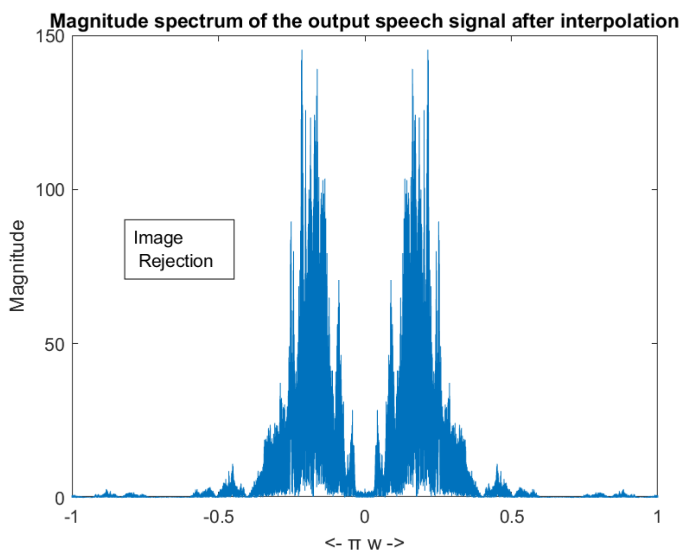

Figure 20: Magnitude Spectrum of the interpolated speech signal

Plots for the Music Signal:

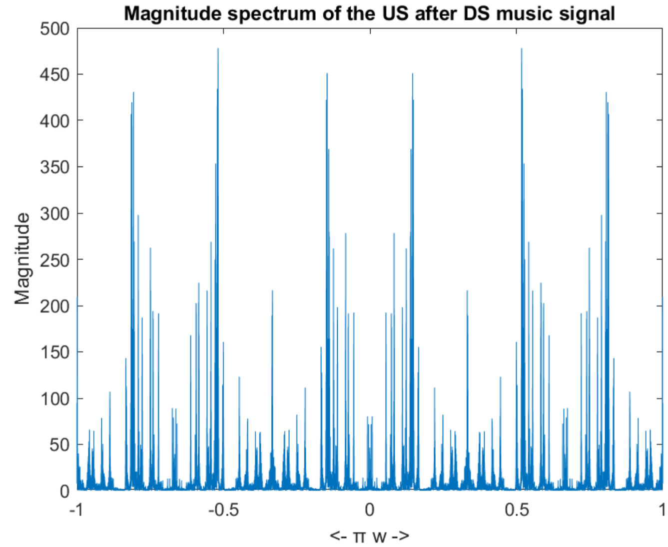

Figure 21: Magnitude Spectrum of the upsampled music signal after downsampling

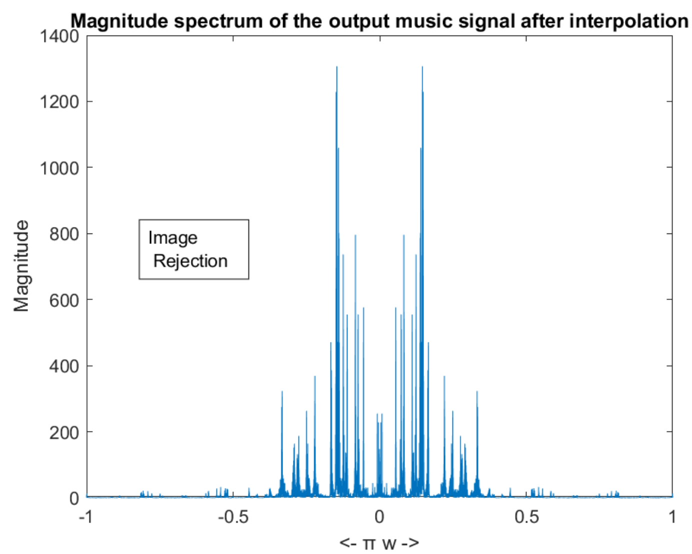

Figure 22: Magnitude Spectrum of the interpolated music signal

5.1 Audio Comparison:

Comparing the audios obtained in the last two sections, we see that,

- Although not as good as the original audio signals, we see that these audios are much better than their earlier counterparts obtained in parts 1 and 2. This is expected, since the resulting sampling frequency ($\frac{3f_s}{4}$) here is higher than that in parts 1 and 2 ($\frac{f_s}{2}$) .
- In the fourth part, due to our operational set up as per the block diagram, we see that, the frequency content present in the signals in $\left[\frac{-\pi}{3}, \frac{-\pi}{4}\right]$ and $\left[\frac{\pi}{4}, \frac{\pi}{3}\right]$ is lost, as we upsample first and then downsample. However, this doesn't affect the perception of the audio a lot,as the intensity of the audio at these frequencies aren't that high, which is evident from the magnitude spectrum.
- In the fifth part, we see that the anti aliasing filter is the first block in the set up. As a result, we lose frequency content beyond $\|\frac{\pi}{4}\|$ which is a costlier loss than in the above case, as a significant portion of the original spectrum is cut off. As a result, we see that the output audio is very muffled and isn't as well as that of the previous case. We also see that the additional images generated by the upsampling process are removed by the interpolation filter.

6 Audio Files:

Click on the below link to access the audio files.
Link to Audio Files