Multi-rate Digital Signal Processing

Experiment: 1

Haricharan B

Theory

Part 1

We are required to downsample the given signal by two.

- In the time domain, downsampling by M can be represented as:

$$x_d(n) = x(Mn) \forall n$$

- In the frequency domain, we have

$$X_{du}(w) = \frac{1}{M} \sum_{i=0}^{M-1} X\left(w - \frac{2\pi i}{M}\right)$$
$$X_d(w) = X_{du}\left(\frac{w}{M}\right)$$

To carry out the downsampling, we use MATLAB `downsample` function. After this, the magnitude spectrum of the downsample output is plotted using the `plot` function.

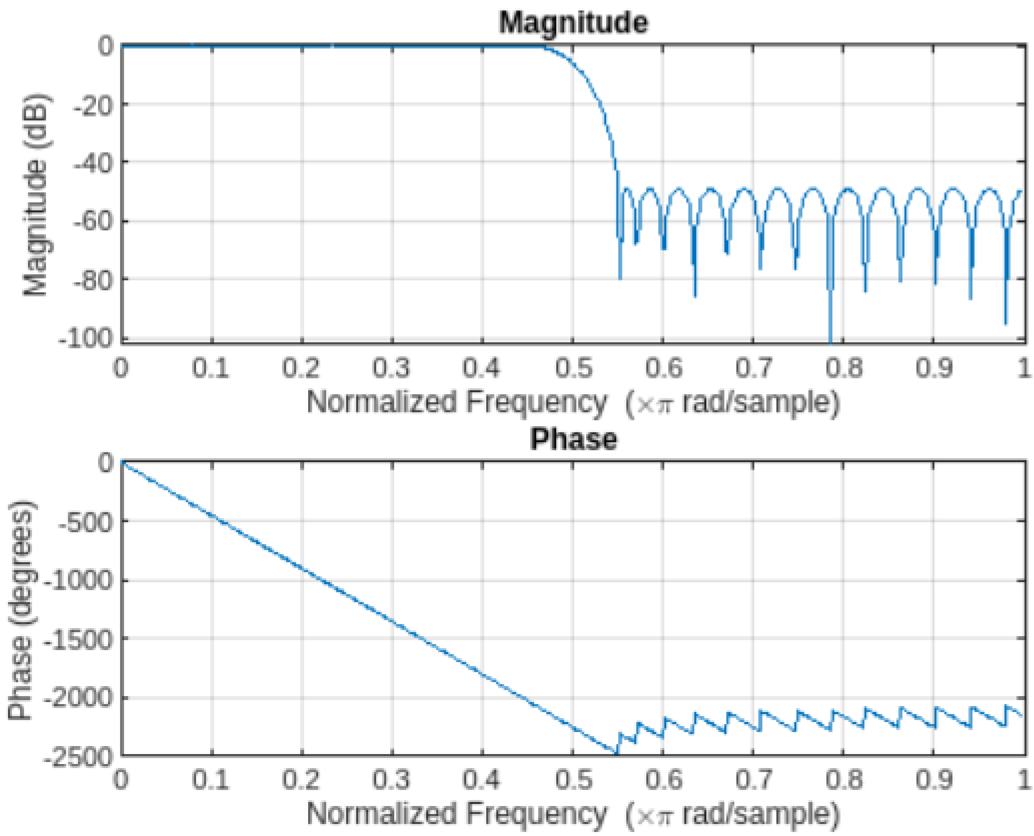
Part 2

An equiripple filter $F_p = 0.45, F_{st} = 0.55, N = 50$ is designed using the `fdesign.lowpass` function, and the corresponding filter response is plotted.

After this, the input signal is passed through the above lowpass anti-aliasing filter and then downsampled by 2. The final signal is plotted using the `plot` function.

The outputs are compared by playing the `sound` function, and qualitatively, the anti-aliased output is clearly better.

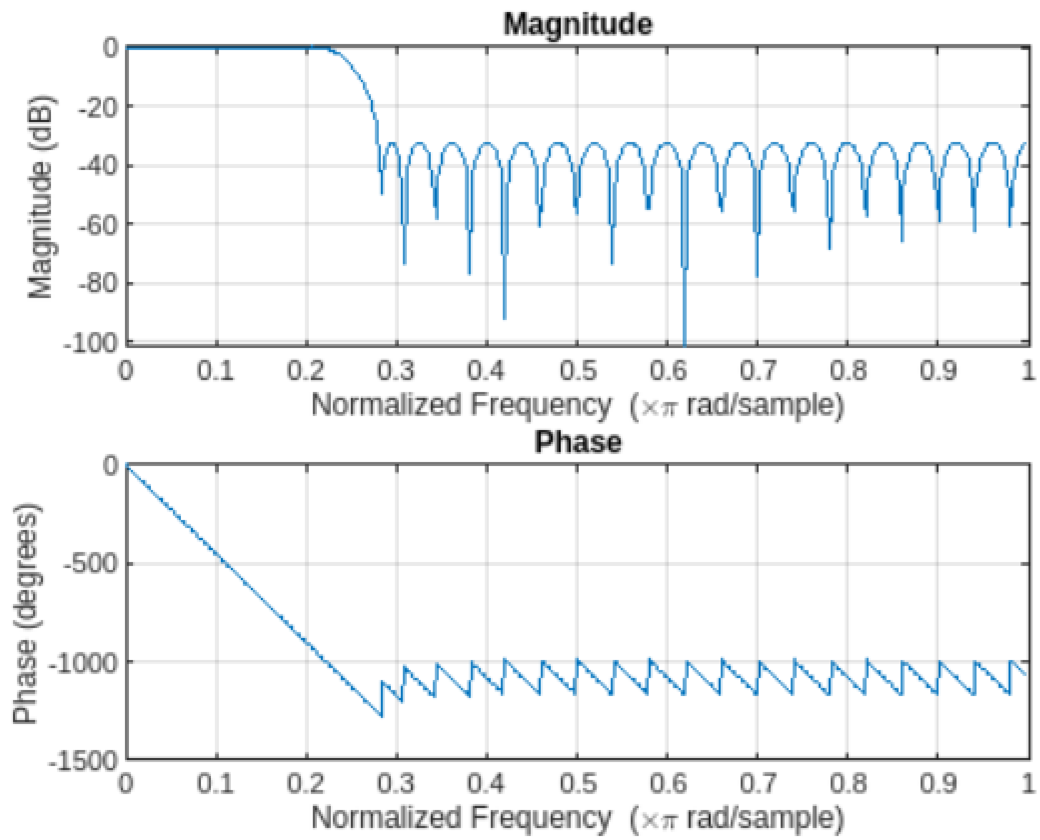
```
Part 2
Plotting the filter response:
```



Part 3

An equiripple filter $F_p = 0.22, F_{st} = 0.28, N = 50$ is designed using the `fdesign.lowpass` function, and the corresponding filter response is plotted.

```
Plotting the filter response:
```



Part 4

We are required to Upsample the given signal by 3.

- In time domain, upsampling by L is represented as:

$$x(n) = \begin{cases} 0, & \text{if } n \bmod L \neq 0 \\ x\left(\frac{n}{L}\right), & \text{if } n \bmod L = 0 \end{cases}$$

- In the frequency domain, upsampling by L is:

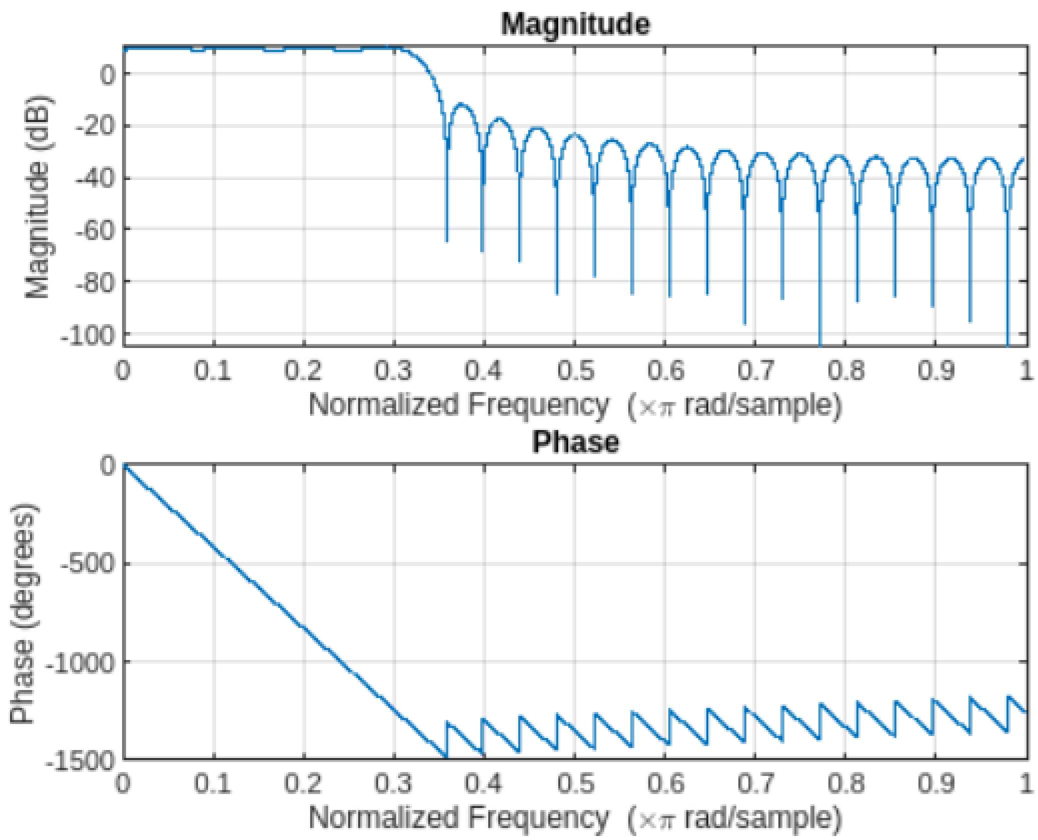
$$X(w) = X(Lw)$$

Upsampling in MATLAB can be achieved by the `upsample` command. Plotting this can be achieves by the `plot` function in MATLAB.

After upsampling, an LPF with cutoff $\frac{\pi}{4}$ is used to filter the upsampled signal and get rid of the copies. Finally, the filtered signal is downsampled to receive output signal. Note that this signal has a sampling frequency of $\boxed{\text{Original Frequency} \times \frac{3}{4}}$

Part 5

The interpolation filter with support $[-8, 8]$ is generated using the MATLAB `intfilt` command, and is also plotted.



The original signal is passed through the LPF with cutoff $\frac{\pi}{4}$. It is then downsampled by 4, then upsampled by 3. Finally, the output is passed through the interpolation filter. Clearly, after interpolating, image-rejection can be observed!

Code

The MATLAB code for all the parts is attached.

```
function sound(audio, Fs)
% Sound function, that plays all sound files one-by-one.
y = audioplayer(audio,Fs);
playblocking(y);
end

function analyze(filename)
% Analyze function analyzes the sound, as required in the experiment
disp(filename)

% Reading the input audio .wav file
[audio, SamplingRate] = audioread(filename);
disp (["Sampling Rate: ", num2str(SamplingRate)])

% Playing the Audio file
% sound(audio, SamplingRate);

%% Part 1
disp("Part 1");

% Downsampling the audio file
audioDS = downsample(audio, 2);
% sound(audioDS, SamplingRate/2);

% Plotting figure
figure;
[X, w] = freqz(audio, 1, SamplingRate);
subplot(2, 1, 1);
plot(w, abs(X));
title("Original Signal in Frequency Domain");
ylabel("$|X(s)|$", 'Interpreter','latex');
xlabel("Frequency (rad)");

[XDS, w] = freqz(audioDS, 1, SamplingRate);
subplot(2, 1, 2);
plot(w, abs(XDS));
title("Downsampled Signal in Frequency Domain");
ylabel("$|X_{d}(s)|$", 'Interpreter','latex');
xlabel("Frequency (rad)");

% Part 2
disp("Part 2");

% Designing the low-pass filter
h1 = fdesign.lowpass('N,Fp,Fst', 50, 0.45, 0.55);
filter1 = design(h1, 'equiripple');

% Plotting the filter response. The plots have been commented out for
```

```
% clarity

%{
figure
disp("Plotting the filter response:")
freqz(filter1.Numerator, 1, SamplingRate)
%}


% Antialiasing, and then downsampling
out = filter(filter1, audio);
outDS = downsample(out, 2);
[YDS, w] = freqz(outDS, 1, SamplingRate);

% Plotting the magnitude spectrum
figure;
hold on
plot(w, abs(YDS));
title("Output after AA filter and downsampling")
ylabel("$|Y_{d}(s)|$", 'Interpreter','latex');
xlabel("Frequency (rad)");

% sound(outDS, SamplingRate/2);
hold off

% Part 3
disp("Part 3");
h2 = fdesign.lowpass('N,Fp,Fst', 50, 0.22, 0.28);
filter2 = design(h2, 'equiripple');
% Plotting filter response
%{
figure
disp("Plotting the filter response:")
freqz(filter2.Numerator, 1, SamplingRate)
%}

% Part 4
disp("Part 4");

% Upsampling and passing through filter and downsampling the audio signal
xup = upsample(audio, 3);
yup = filter(filter2, xup);
ydown = downsample(yup, 4);

% Plotting the figure
figure;
subplot(2, 1, 1);
[XUP, w] = freqz(xup, 1, SamplingRate);
plot(w, abs(XUP));
title("Upsampled Signal in Frequency Domain");
ylabel("$|X_{u}(s)|$", 'Interpreter','latex');
xlabel("Frequency (rad)");
```

```
subplot(2, 1, 2);
[YD, w] = freqz(ydown, 1, SamplingRate);
plot(w, abs(YD));
title("Final Signal in Frequency Domain");
ylabel("$|Y_{d}(s)|$", 'Interpreter','latex');
xlabel("Frequency (rad)");

% sound(ydown, SamplingRate * 3/4);

% Part 5
disp("Part 5");

% Plotting interpolation filter
figure
interpolationFilter = intfilt(3, 8, 1);
% freqz(interpolationFilter, 1, SamplingRate)

% Passing through filter, downsampling, and then upsampling
y = filter(filter2, audio);
yd = downsample(y, 4);
yu = upsample(yd, 3);

% Plotting the above signal
[YU, w] = freqz(yu, 1, SamplingRate);
figure
plot(w, abs(YU));
title("Output after Upsampling by 3 in Frequency Domain");
ylabel("$|Y_{d}(s)|$", 'Interpreter','latex');
xlabel("Frequency (rad)");
% sound(yu, SamplingRate * 3/4);

yy = filter(interpolationFilter, 1, yu);
figure
[YY, w] = freqz(yy, 1, SamplingRate);
plot(w, abs(YY));
title("Output after interpolation in Frequency Domain, with image rejection");
ylabel("$|Y_{interpol}(s)|$", 'Interpreter','latex');
xlabel("Frequency (rad)");

% sound(yy, SamplingRate * 3/4);

end
```

Deliverables

Attached below

Conclusions

Speech File

Part 2

The downsampled signal with the anti-aliasing filter has a greater quality than the downsampled signal without the anti-aliasing filter, for both the speech and the audio file.

Part 4

The signal with the anti-aliasing filter has a greater quality than the signal without the anti-aliasing filter, for both the speech and the audio file.

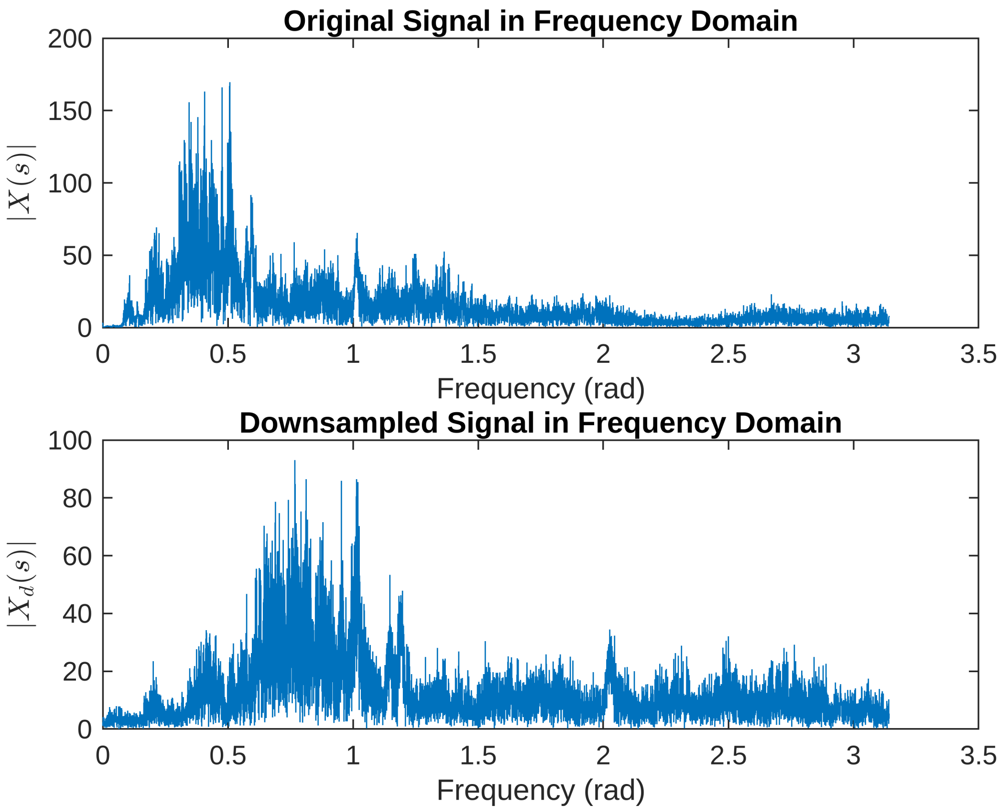
Part 5

The signal has a greater quality after passing through interpolation filter, for both the speech and audio file.

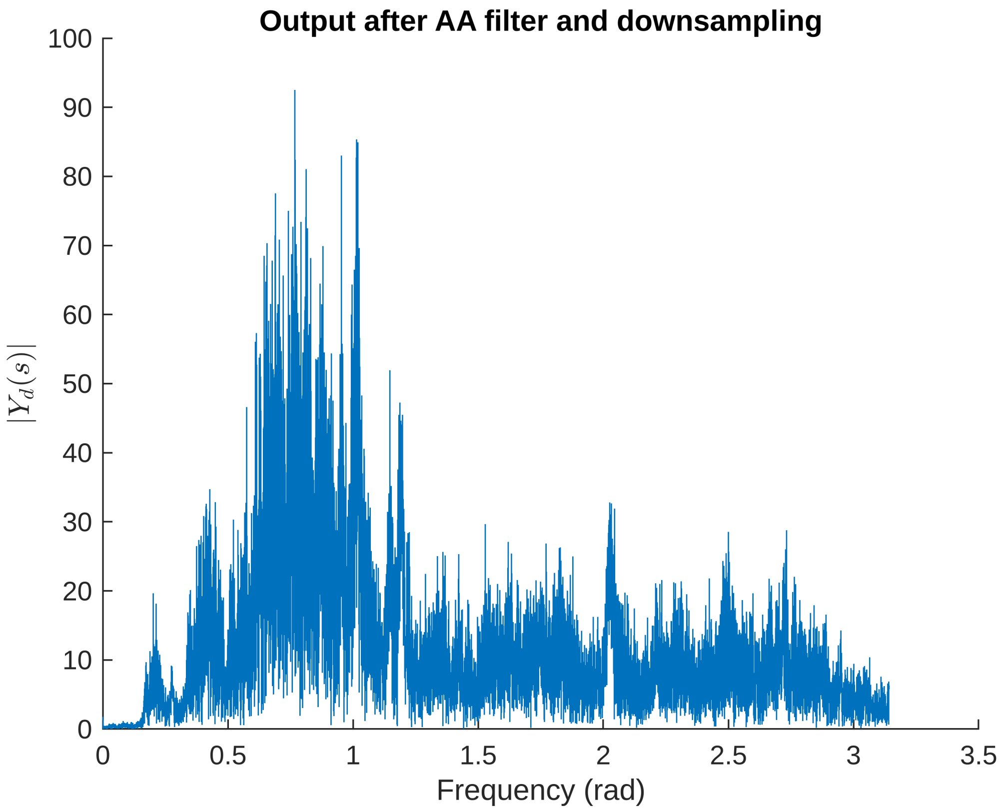
Experiment-1

```
speech8khz.wav
    "Sampling Rate: "    "8000"
```

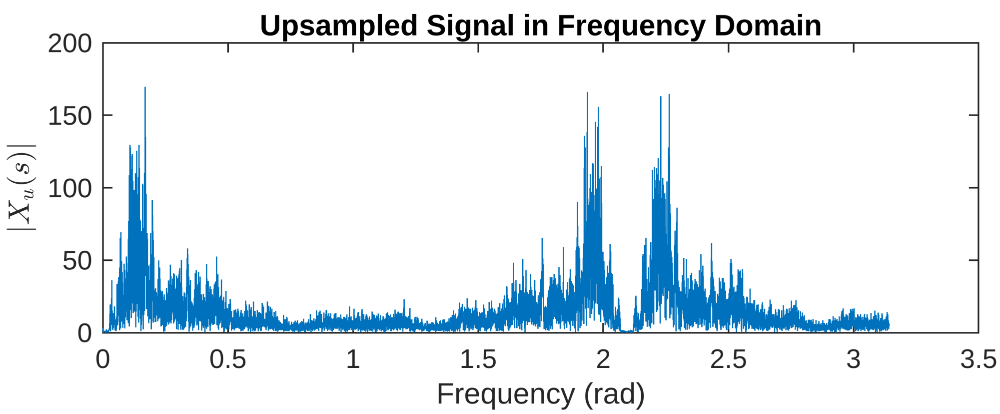
```
Part 1
```

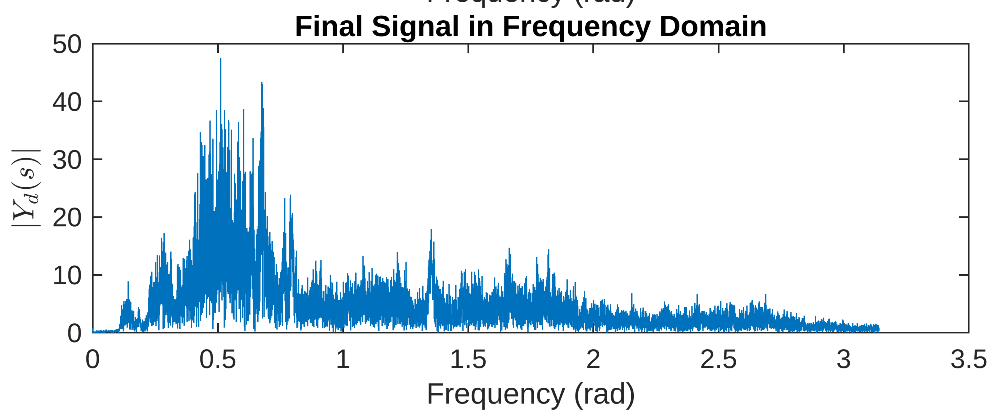



```
Part 2
```



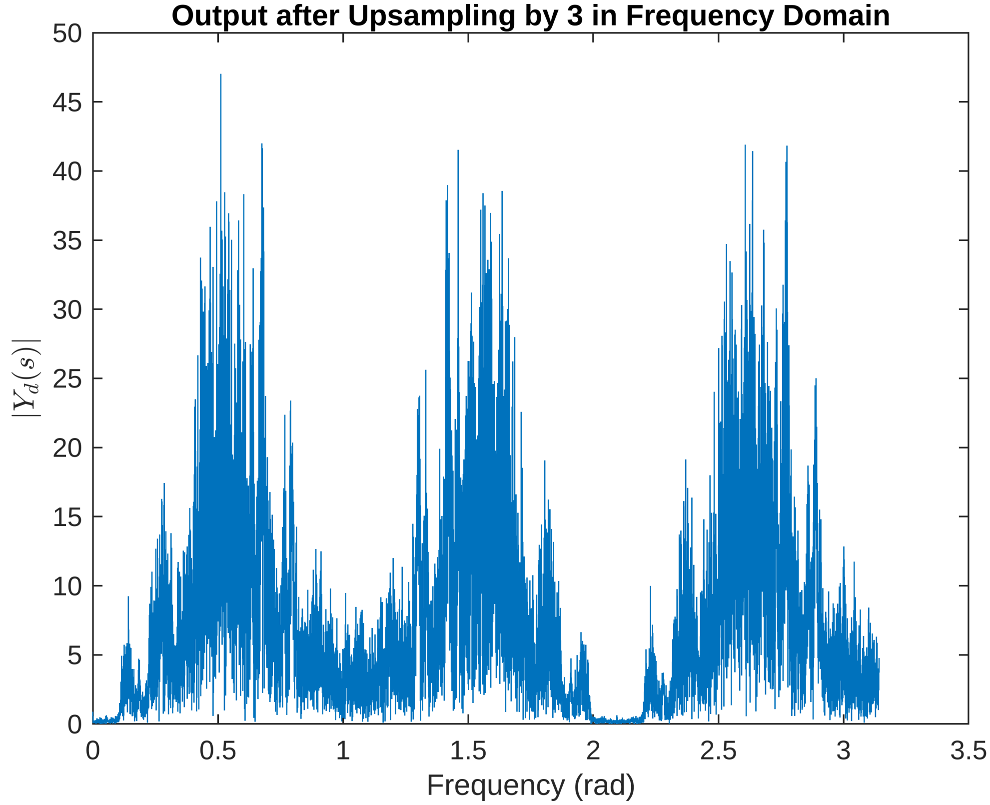

```
Part 3
Part 4
```



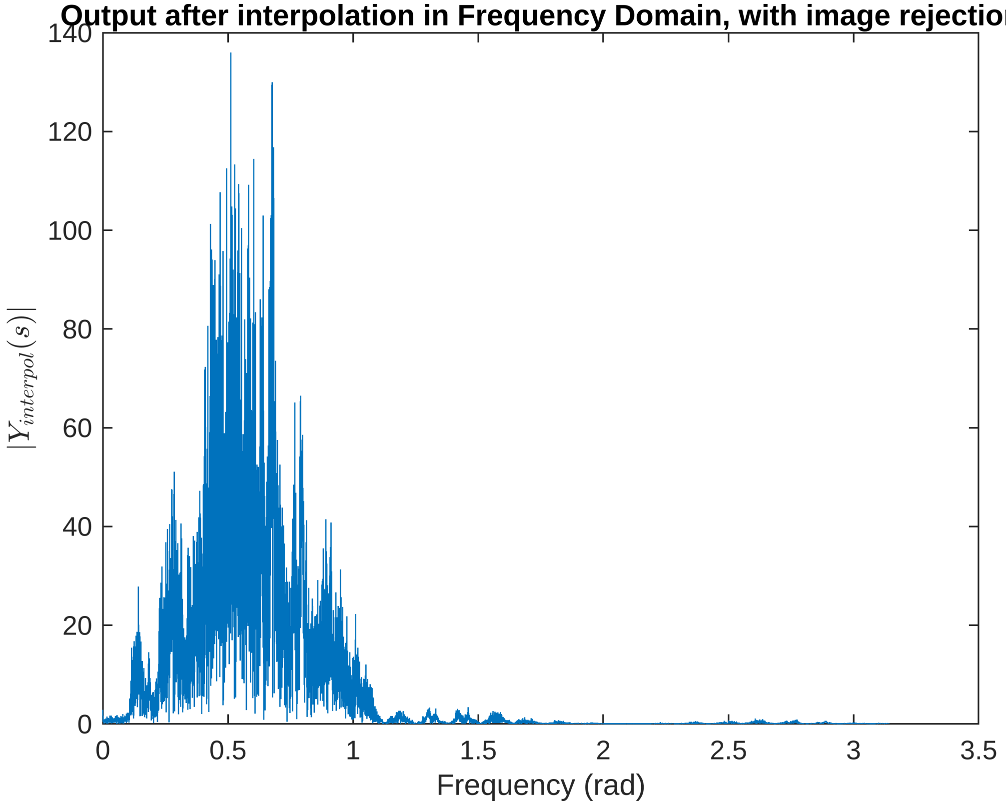



```
Part 5
```

Output after Upsampling by 3 in Frequency Domain

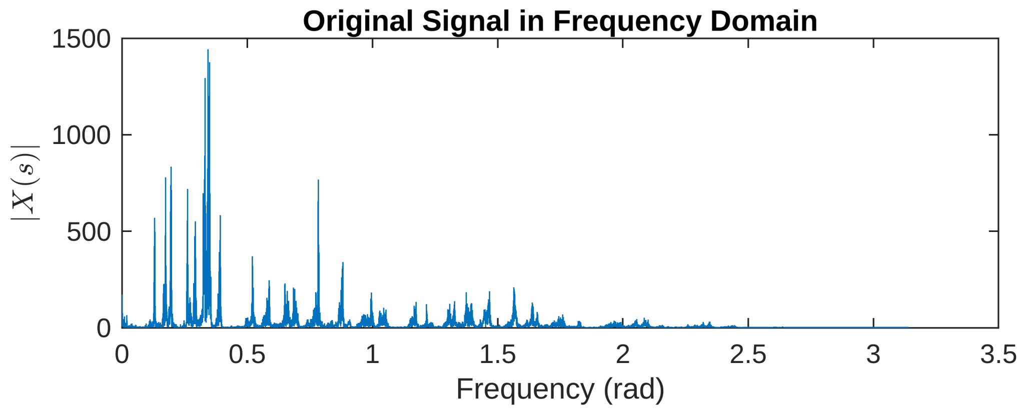


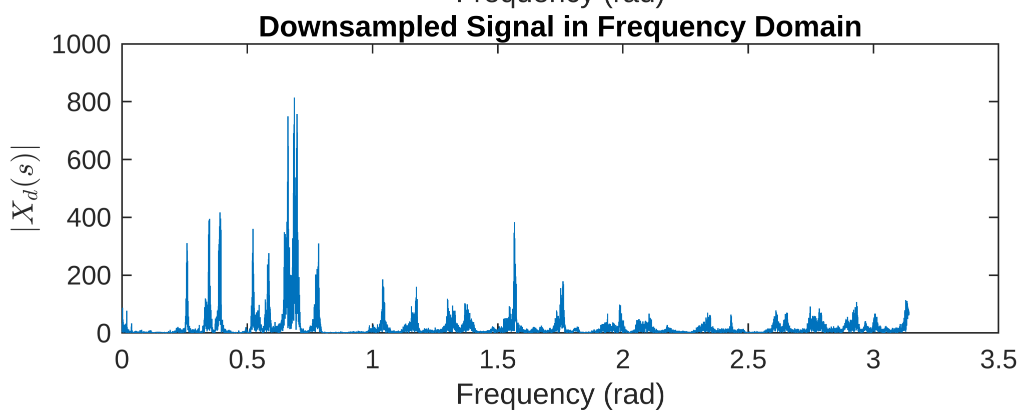
Output after interpolation in Frequency Domain, with image rejection



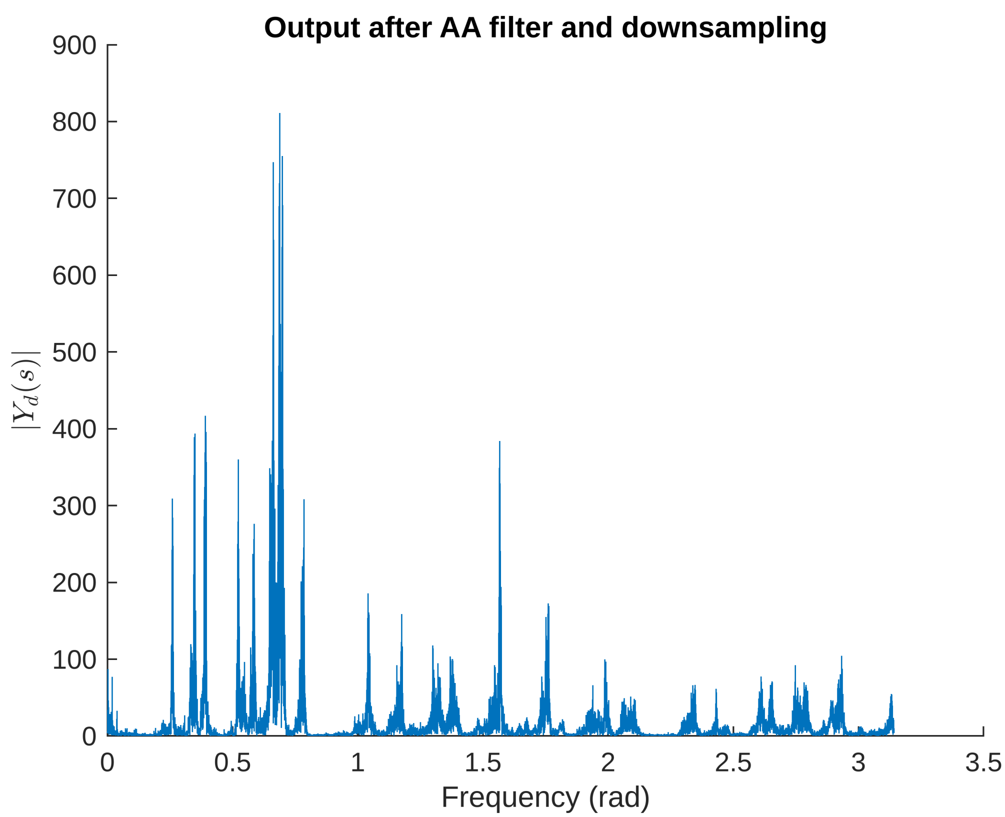

```
music16khz.wav
    "Sampling Rate: "    "16000"
```

Part 1

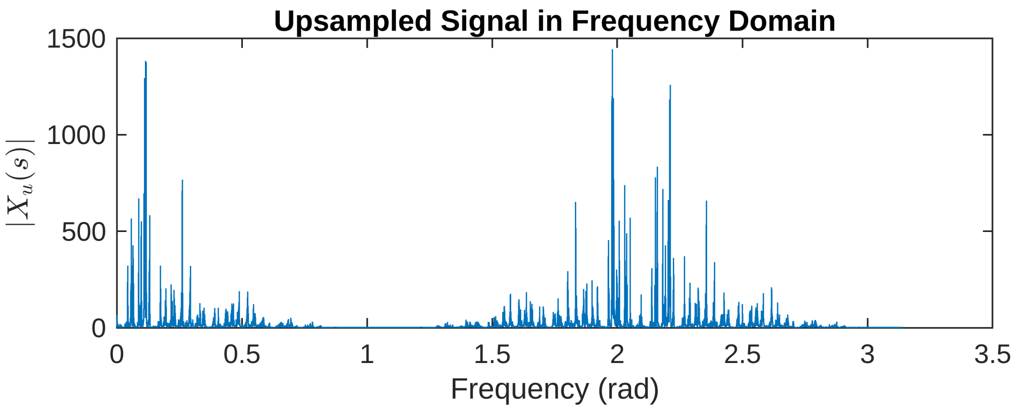

Original Signal in Frequency Domain
1500
1000
500
0
$|X(s)|$
0
0.5
1
1.5
2
2.5
3
3.5
Frequency (rad)

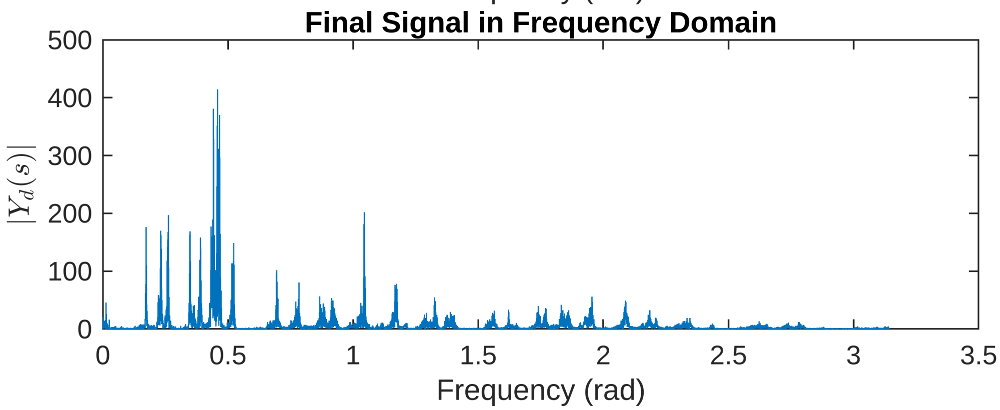

Downsampled Signal in Frequency Domain
1000
800
600
400
200
0
$|X_d(s)|$
0
0.5
1
1.5
2
2.5
3
3.5
Frequency (rad)

Part 2

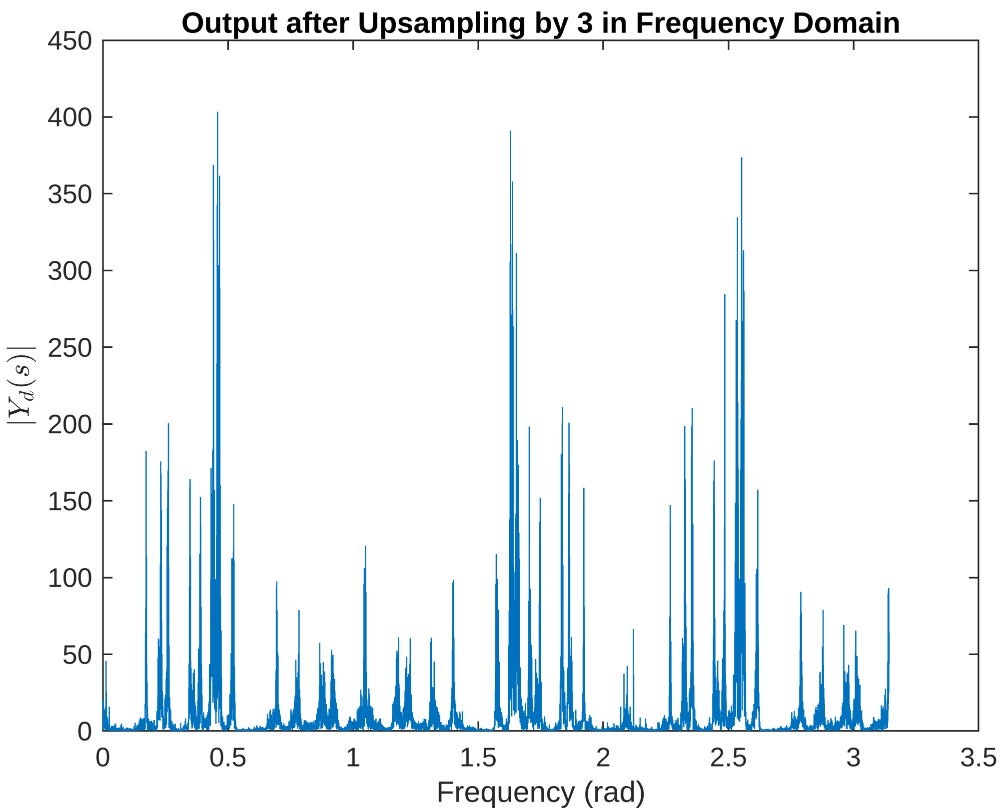

Output after AA filter and downsampling
900
800
700
600
500
400
300
200
100
0
$|Y_d(s)|$
0
0.5
1
1.5
2
2.5
3
3.5
Frequency (rad)

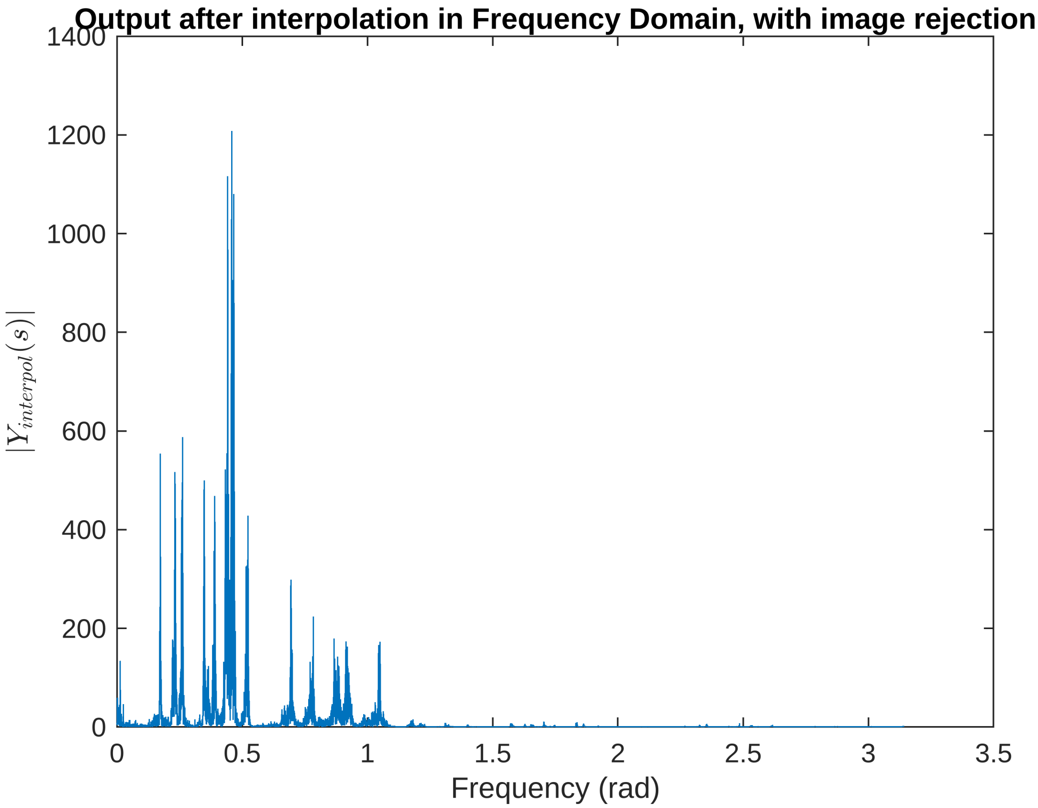
Part 3
Part 4





Part 5




Output after interpolation in Frequency Domain, with image rejection
1400
1200
1000
800
600
400
200
0
$|Y_{interpol}(s)|$
0
0.5
1
1.5
2
2.5
3
3.5
Frequency (rad)