Student Name: Emirhan Uçar (25265), Emre Eren (25139)

Date of Submission: **31.03.2020**

EE312 LAB REPORT#2

SPRING 2020

PROBLEM 1

LCCDE is given with the formula, $$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$$

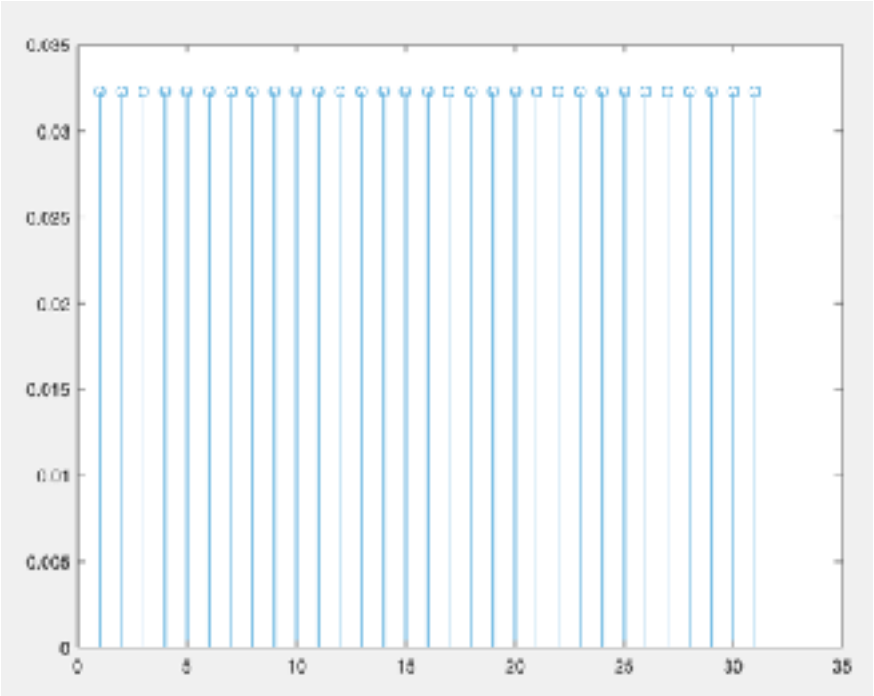
System 1

(i) $b = \underbrace{\left[\frac{1}{M}, \ldots, \frac{1}{M}\right]}_{M}$, where $M = 31$; and $a = [1]$.

a)

Firstly we read the sound file, after that we assigned the value 31 into the variable m to form vector that has length 31 and each of the elements equals to the 1/31 and we assigned 1 to the variable a. After that, by using impz function we obtained impulse response of the system. Then we used stem function to sketch impulse response.

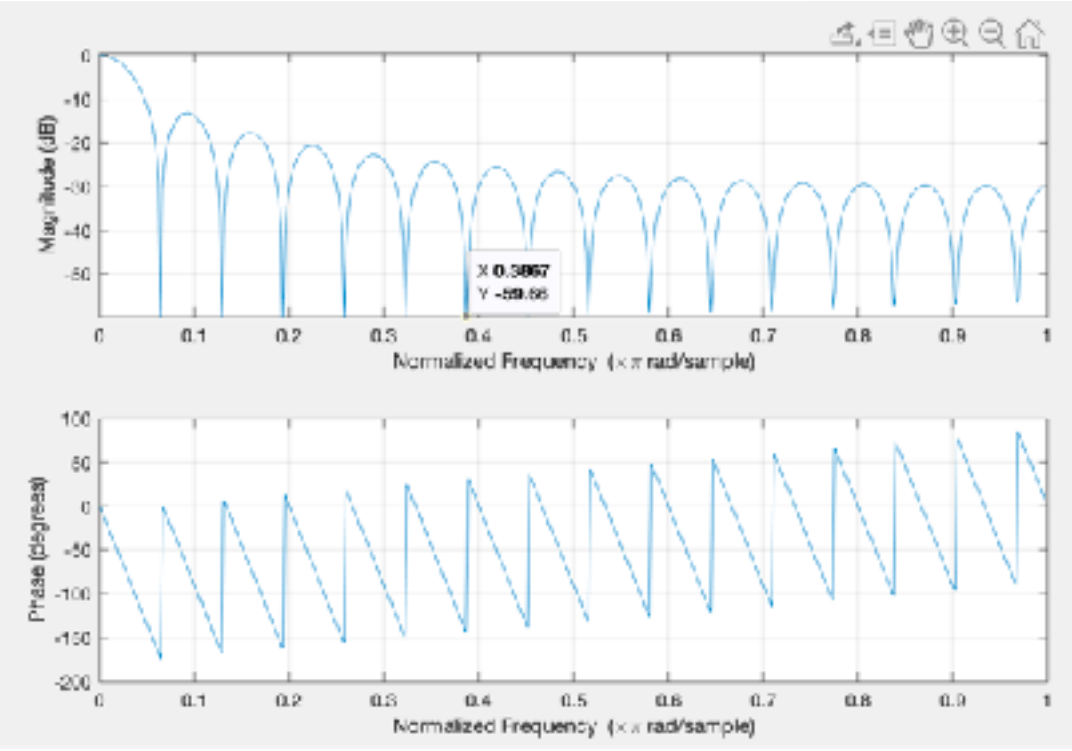
```
[x,fs] = audioread('music1.wav');
m = 31;
b(1:m) = 1/m;
a = 1;

h = impz(b,a);
stem(h);
```



*This system represents the 31 point averaging FIR

b) We did same process to calculate impulse response. After that we used freqz function to obtain the plot of magnitude response in dB and phase response in degrees.

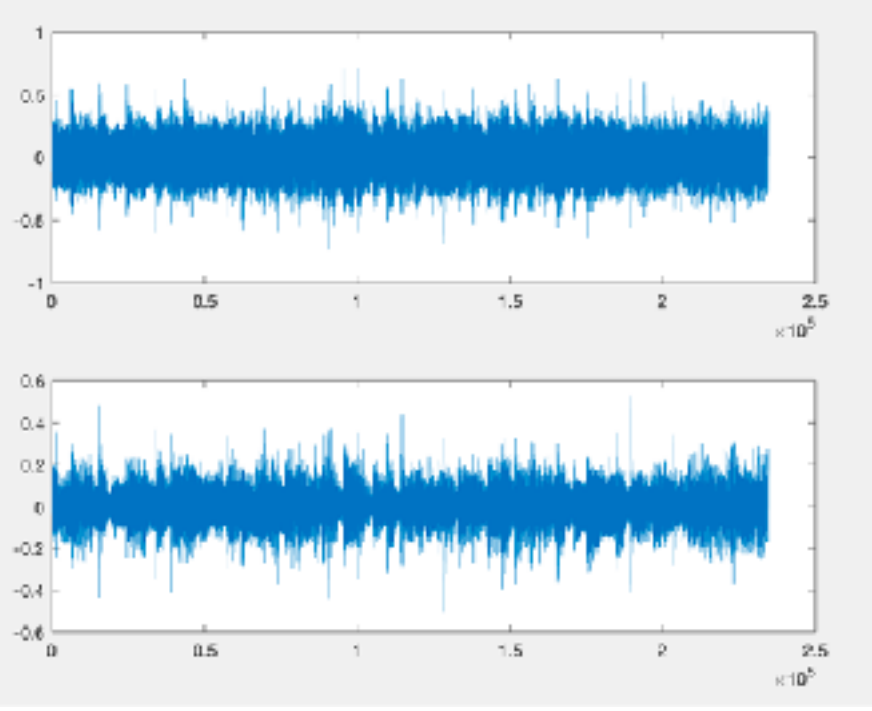
```
[x,fs] = audioread('music1.wav');
m = 31;
b(1:m) = 1/m;
a = 1;

h = impz(b,a);
freqz(h);
```



c) This filter is low-pass filter because it passes components at the low frequencies in greater magnitude when compared with high frequency components and it is finite impulse response filter (FIR) because behavior of phase response is linear.

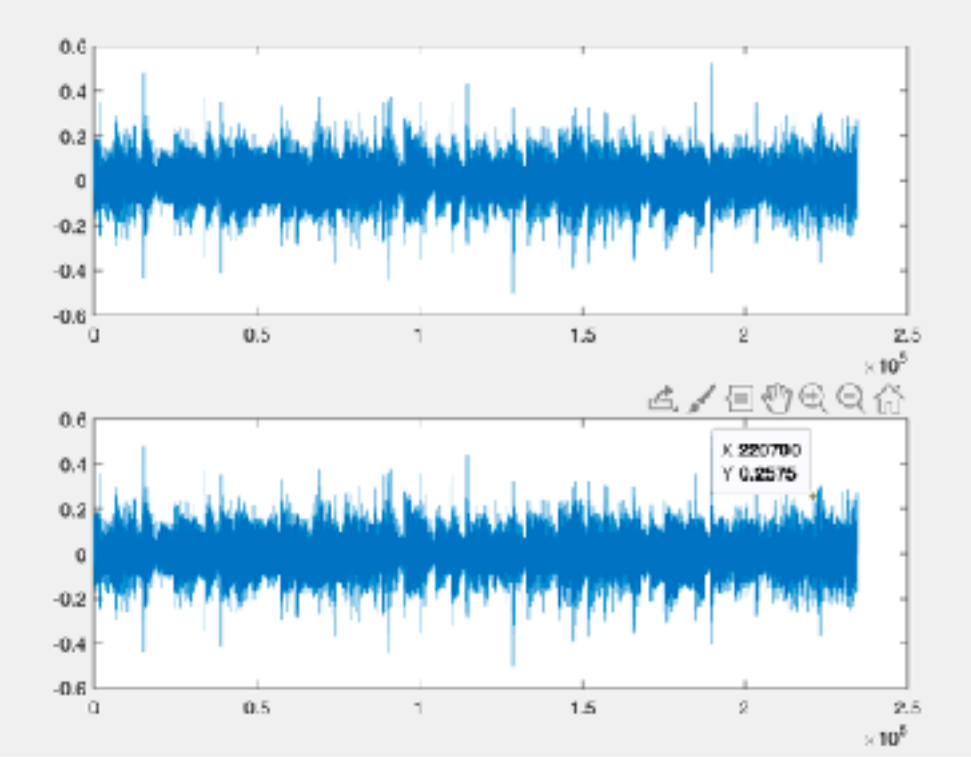
d) Lengths of the input signal and filtered output signal are same. So it can be said that filter function does not affect the length of the signal. Its because of filter behaves as low pass filter. It can be inferred when listen to both of the sounds. High frequency components are suppressed when listen to output signal.

```
[x,fs] = audioread('music1.wav');

m = 31;
b(1:m) = 1/m;
a = 1;

y= filter(b,a,x);
sound(y,fs);

subplot(211);
plot(x);
subplot(212);
plot(y);
```



e) When we convolve input signal with impulse response we hear almost same sound as filtered version, however there is a difference in terms of their lengths. Because when we convolve something we also shift it.

```
[x,fs] = audioread('music1.wav');

m = 31;
b(1:m) = 1/m;
a = 1;
h = impz(b,a);

y1 = filter(b,a,x);
y2 = conv(x,h);

subplot(211);
plot(y1);
subplot(212);
plot(y2);
```



f) The length of the convolution output signal is M+N-1. Filter function reduces the length of the output signal as M-1. So, we added M-1 zero pad to our input to obtain same length signals.

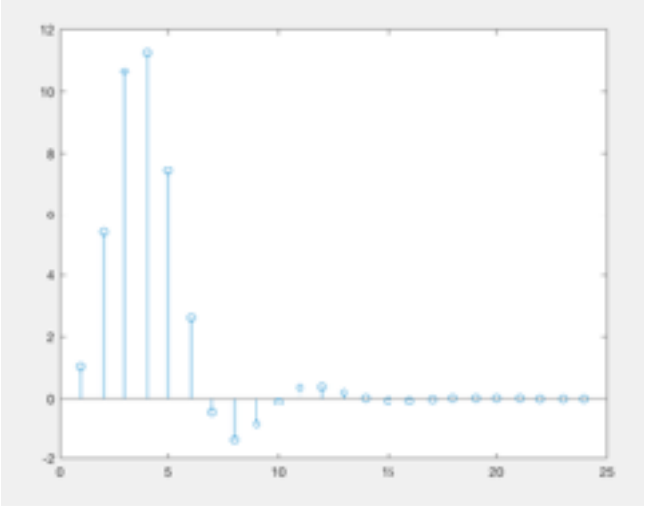
x	234531x1 double
x1	234561x1 double
y1	234561x1 double
y2	234561x1 double

```
[x,fs] = audioread('music1.wav');

m = 31;
b(1:m) = 1/m;
a = 1;
h = impz(b,a);

y1 = conv(x,h);

len = m-1;
x1 = [x;zeros(len,1)];
y2 = filter(b,a,x1);

subplot(211);
plot(y1);
subplot(212);
plot(y2);
```

System 2

(ii) $b = [1,\ 4,\ 4,\ 1]$; and $a = [1,\ -1.4,\ 0.9025,\ -0.2263]$.

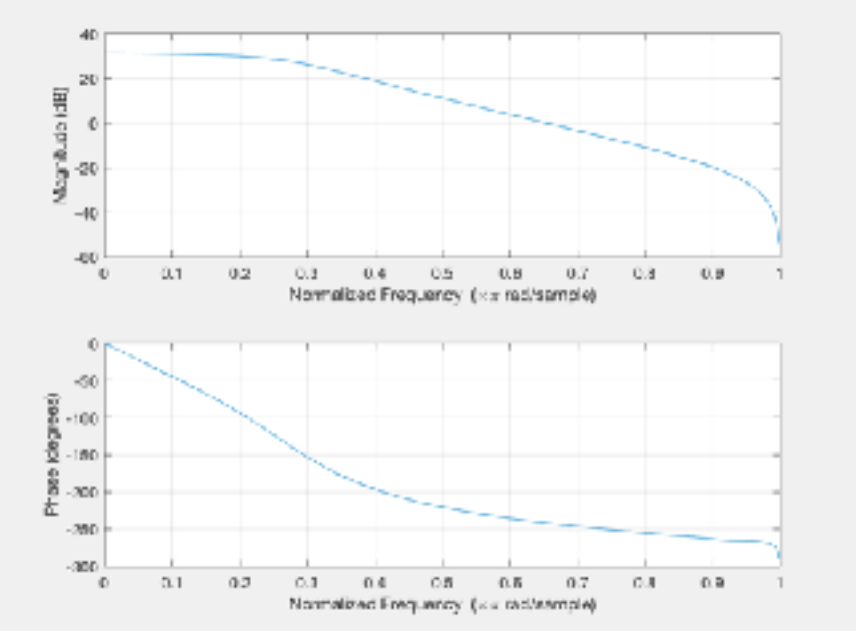
a) Firstly we read the sound file, after that we assigned b as [1, 4, 4 ,1] and a as [1, -1.4, 0.9025, -0.2263]. impz function returns the impulse response of the digital filter with numerator coefficients b and denominator coefficients a. Therefore, by using impz function we obtained impulse response of the system. Then we used stem function to sketch impulse response.



```
[x,fs] = audioread('scales.wav');
b = [1, 4, 4, 1];
a = [1, -1.4, 0.9025, -0.2263];

h = impz(b,a);
stem(h);
```

b) We did same process to calculate impulse response. After that we used freqz function to obtain the plot of magnitude response in dB and phase response in degree.

```
[x,fs] = audioread('scales.wav');
b = [1, 4, 4, 1];
a = [1, -1.4, 0.9025, -0.2263];

h = impz(b,a);
stem(h);
freqz(h);
```

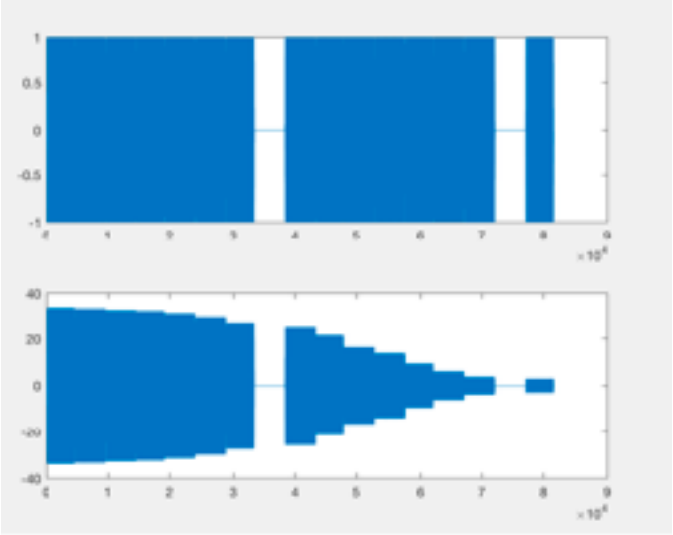


c) This filter is low-pass filter because it passes low frequencies in Magnitude Response. It is infinite impulse response filter (IIR) because behavior of Phase Response is not linear.

d) Lengths of the real input signal and filtered output signal are same. So it can be said that filter function does not affect the length of the signal. Its

because of filter behaves as low pass filter. It can be inferred when listen to both of the sounds. High frequency components are suppressed when listen to output signal.

```
[x,fs] = audioread('scales.wav');
b = [1, 4, 4, 1];
a = [1, -1.4, 0.9025, -0.2263];

h = impz(b,a);
stem(h);
freqz(h);

y = filter(b,a,x);
sound(y,fs);

subplot(211);
plot(x);

subplot(212);
plot(y);
```

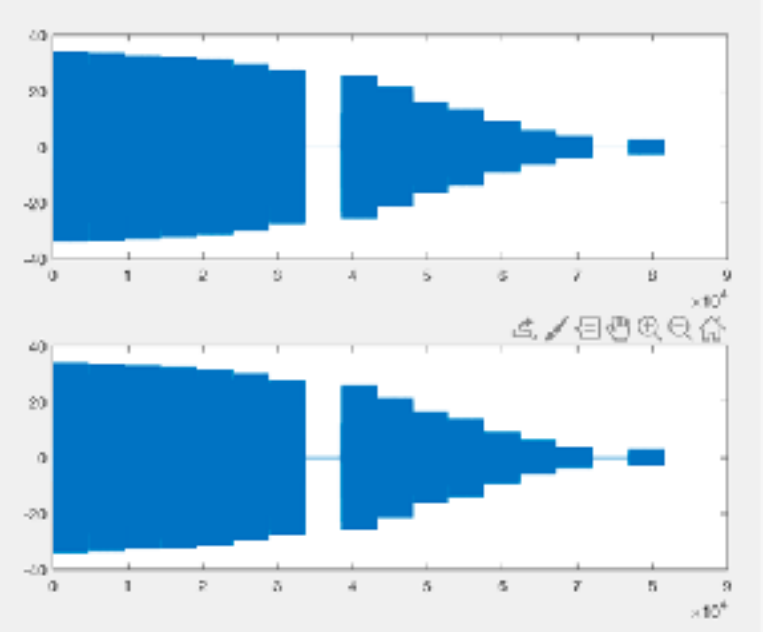


e) When we convolve input signal with impulse response we hear almost same sound as filtered version, however there is a difference in terms of their lengths (not huge difference (related with length of impulse response)). Because when we convolve something we also shift

```
[x,fs] = audioread('scales.wav');
b = [1, 4, 4, 1];
a = [1, -1.4, 0.9025, -0.2263];

h = impz(b,a);

y1 = filter(b,a,x);
y2 = conv(x,h);

subplot(211);
plot(y1);
%scund(y1,fs)

subplot(212);
plot(y2);
sound(y2,fs)
```

it. In this case our filter is IIR, which means impulse response has infinite length which requires infinite memory. Therefore, to be able to convolve signal with IIR filter conv command approximate impulse response and makes it finite length.

y1	81617x1 double
y2	81640x1 double



f) The length of the convolution output signal is 24. Convolution increase the length of the input signal in output by 24 - 1 = 23. However, filtering input signal does not change the length of the signal. To be able to obtain same length with filtering signal, we should do zero padding to

```
[x,fs] = audioread('scales.wav');
b = [1, 4, 4, 1];
a = [1, -1.4, 0.9025, -0.2263];
h = impz(b,a);

y = conv(x,h);

len = 23; %length of impulse response minus
x1 = [x;zeros(len,1)];
y1 = filter(b,a,x1);

subplot(312);
plot(y);
subplot(313);
plot(y1);
```

input signal. So, we add 23 zero pad to our input to obtain same length signals.

Name	Value
a	[1,-1.4000,0.902...
b	[1,4,4,1]
fs	16000
h	*24x1 double*
len	23
x	*81617x1 double*
x1	*81640x1 double*
y	*81640x1 double*
y1	*81640x1 double*

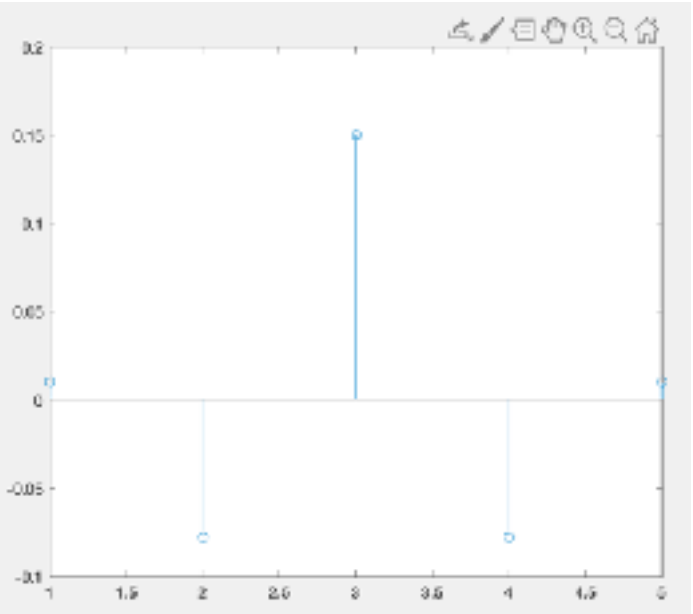
System 3

(iii) $b = [0.0102,\ -0.0779,\ 0.1504,\ -0.0779,\ 0.0102]$; and $a = [1]$.

a)

Firstly we read the sound file, after that we assigned the value 31 into the variable m to form vector that has length 31 and each of the elements equals to the 1/31 and we assigned 1 to the variable a. After that, by using impz function we obtained impulse response of the system. Then we used stem function to sketch impulse response.

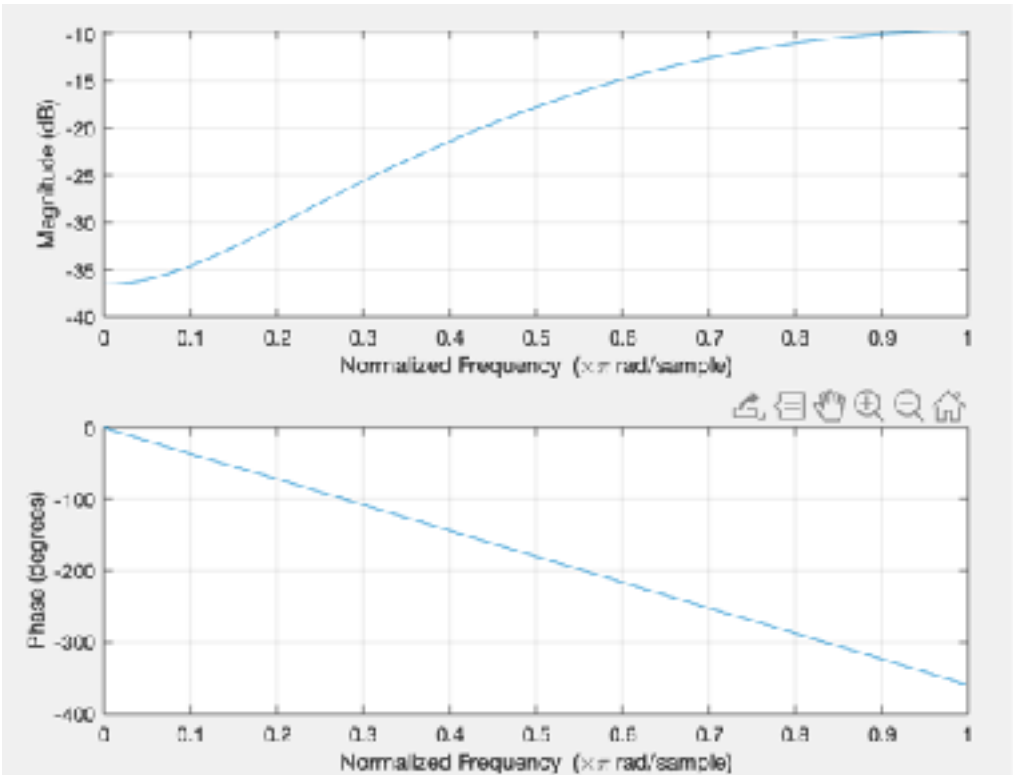
```
[x,fs] = audioread('music1.wav');

b = [0.0102, -0.0779, 0.1504, -0.0779, 0.0102];
a = 1;

h = impz(b,a);
stem(h);
```



b)

We did same process to calculate impulse response. After that we used freqz function to obtain the plot of magnitude response in dB and phase response in degrees.

```
[x,fs] = audioread('music1.wav');

b = [0.0102, -0.0779, 0.1504, -0.0779, 0.0102];
a = 1;

h = impz(b,a);
freqz(h);
```

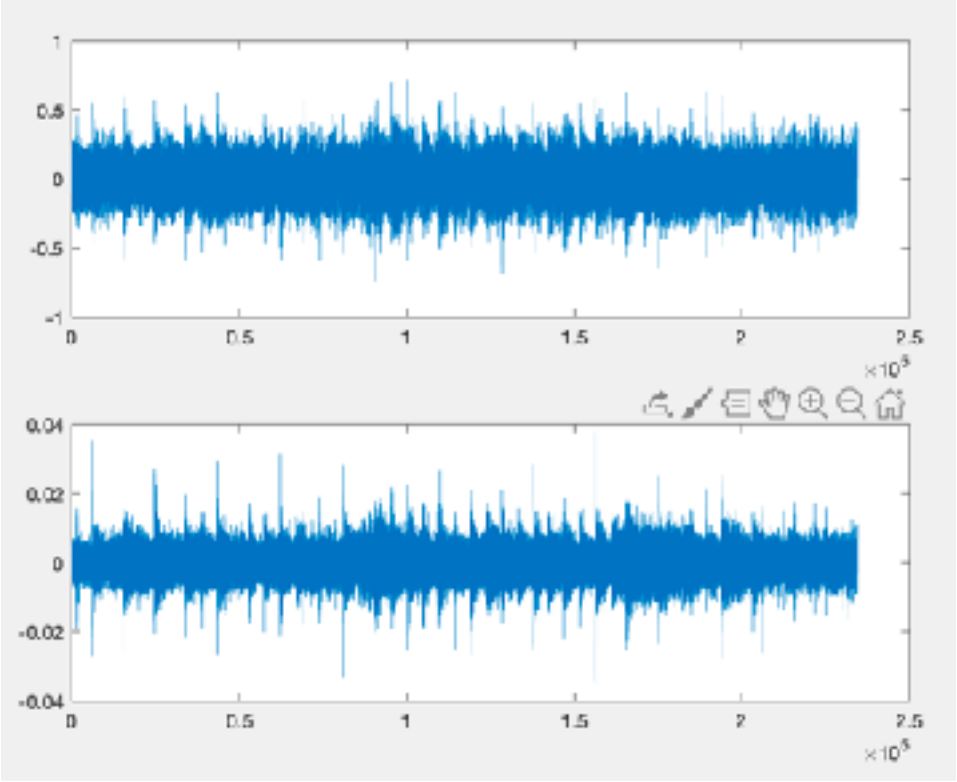


c) This filter is high-pass filter because it passes components at the high frequencies in greater magnitude when compared with low frequency components and it is finite impulse response filter (FIR) because behavior of phase response is linear.

d) Lengths of the read input signal and filtered output signal are same. So it can be said that filter function does not affect the length of the signal. Its because of filter behaves as low pass filter. It can be inferred when listen to both of the sounds. High frequency components are suppressed when listen to output signal.

```
[x,fs] = audioread('music1.wav');

b = [0.0102, -0.0779, 0.1504, -0.0779, 0.0102];
a = 1;

y= filter(b,a,x);

sound(y,fs);

subplot(211);
plot(x);
subplot(212);
plot(y);
```

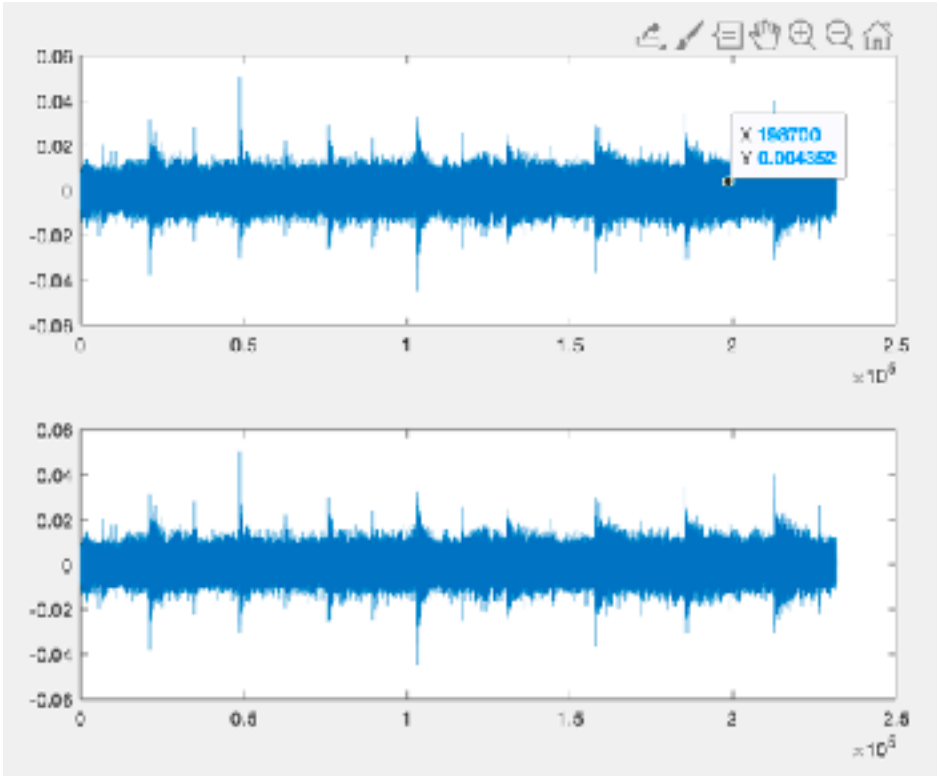
x	234531x1 double
y	234531x1 double



e) When we convolve input signal with impulse response we hear almost same sound as filtered version, however there is a difference in terms of their lengths. Because when we convolve something we also shift it.

y1	231521x1 double
y2	231517x1 double

```
[x,fs] = audioread('music2.wav');

b = [0.0102, -0.0779, 0.1504, -0.0779, 0.0102];
a = 1;
h = impz(b,a);

y1 = conv(x,h);
y2= filter(b,a,x);

sound(y1,fs);

subplot(211);
plot(x);
subplot(212);
plot(y1);
```



f) The length of the convolution output signal is 5. Convolution increase the length of the input signal in output by 5 - 1 = 4. However, filtering input signal does not change To be able to obtain same length with filtering signal we should do zero padding to input signal. So, we added 4 zero pad to our input to obtain same length signals. As a consequence we prohibited the shifts by convolution.

```
[x,fs] = audioread('music2.wav');

b = [0.0102, -0.0779, 0.1504, -0.0779, 0
a = 1;
h = impz(b,a);

y1 = conv(x,h);

len = length(h) - 1;
x1 = [x;zeros(len,1)];
y2 = filter(b,a,x1);
```

y1	231521x1 double
y2	231521x1 double

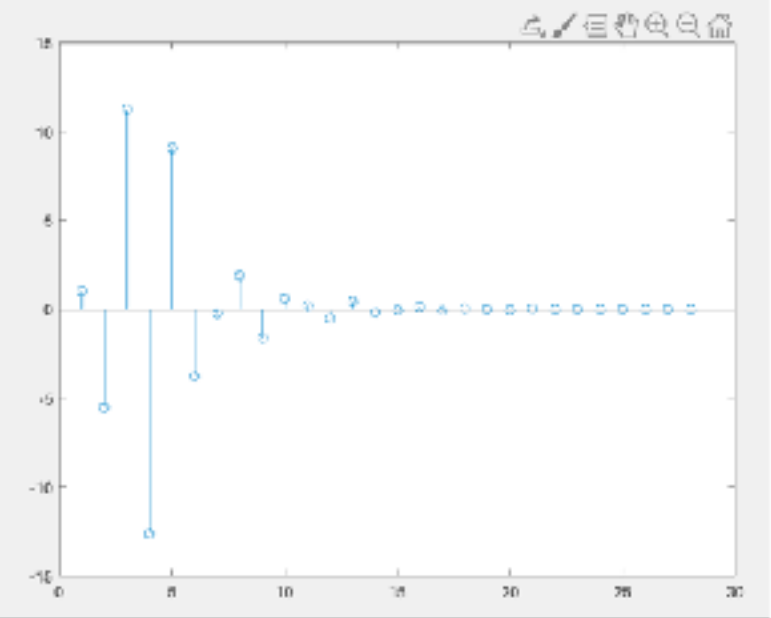
System 4

(iv) $b = [1, -4, 4, -1]$; and $a = [1, 1.5, 1, 0.25]$.

a) Firstly we read the sound file, after that we assigned b as [1, -4, 4 ,-1] and a as [1, -1.5, 1, 0.25]. impz function returns the impulse

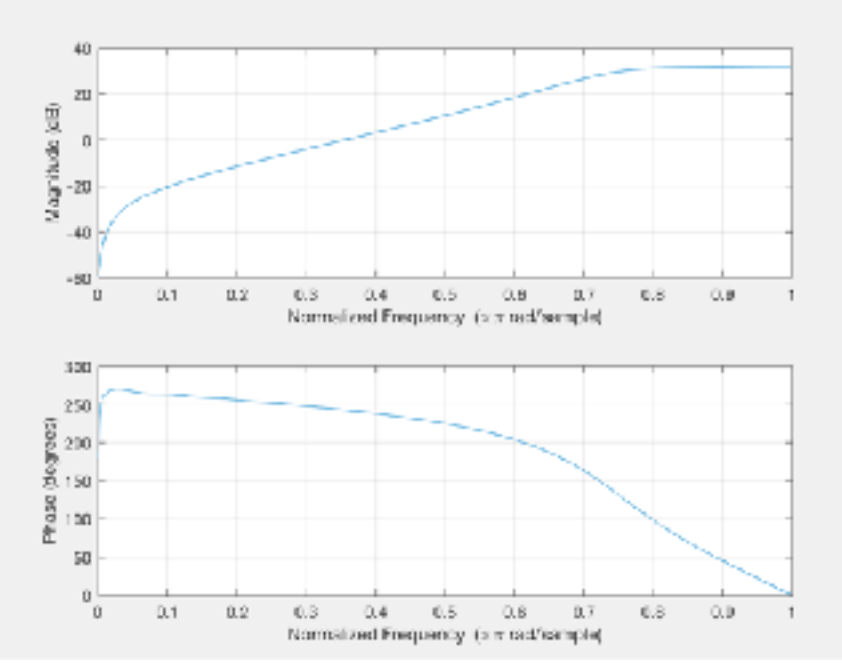
```
[x,fs] = audioread('scales.wav');
b = [1, -4, 4, -1];
a = [1, 1.5, 1, 0.25];

h = impz(b,a);
stem(h);
```

response of the digital filter with numerator coefficients b and denominator coefficients a. Therefore, by using impz function we obtained impulse response of the system. Then we used stem function to sketch impulse response.



b) We did same process to calculate impulse response. After that we used freqz function to obtain the plot of magnitude response in dB and phase response in degree.

```
[x,fs] = audioread('scales.wav');
b = [1, -4, 4, -1];
a = [1, 1.5, 1, 0.25];

h = impz(b,a);
stem(h);
freqz(h);
```

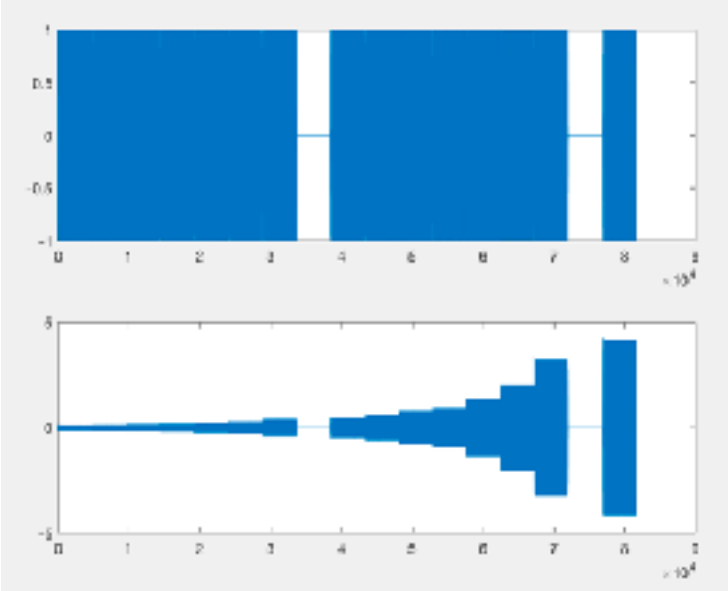


c) This filter is high-pass filter because it passes high frequencies in Magnitude Response as can be seen above. It is infinite impulse response filter (IIR) because behavior of Phase Response is not linear.

d) Lengths of the real input signal and filtered output signal are same as can be seen in variable image. So it can be said that filter command does not affect the length of the signal because filter does not manipulate length of input signal. “scales.wav” is going to low note to high note as we know which means low frequency to high frequency. As can be seen in plot, in filtered signal low note which refers low frequency components are suppressed and high frequency components are allowed to pass.

```
[x,fs] = audioread('scales.wav');
b = [1, -4, 4, -1];
a = [1, 1.5, 1, 0.25];

h = impz(b,a);
stem(h);
freqz(h);

y = filter(b,a,x);
sound(y,fs);

subplot(211); plot(x);
subplot(212); plot(y);
```

h	28x1 double
x	81617x1 double
y	81617x1 double

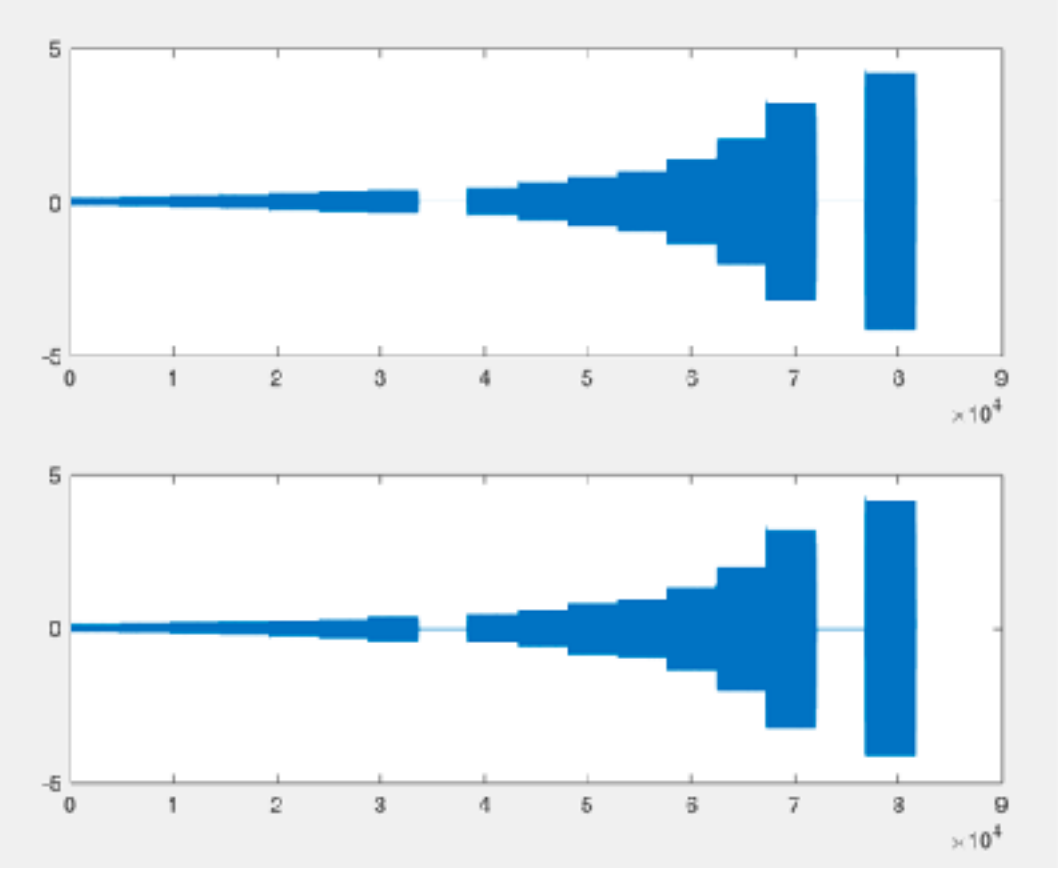


e) When we convolve input signal with impulse response we hear almost same sound as filtered version, however there is a difference in terms of their lengths (not huge difference (related with length of impulse response)). Because when we convolve something we also shift it. In this case our filter is IIR, which means impulse response has infinite length which requires infinite memory.

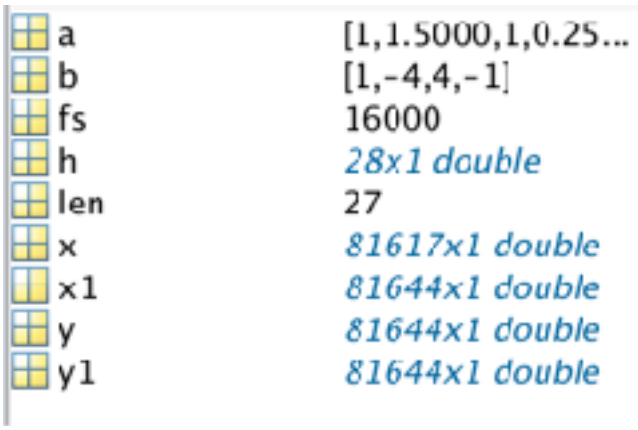
```
[x,fs] = audioread('scales.wav');
b = [1, -4, 4, -1];
a = [1, 1.5, 1, 0.25];

h = impz(b,a);

y1 = filter(b,a,x);
y2 = conv(x,h);

subplot(211);
plot(y1);
%sound(y1,fs)

subplot(212);
plot(y2);
sound(y2,fs)
```

h	28x1 double
x	81617x1 double
y1	81617x1 double
y2	81644x1 double

Therefore, to be able to convolve signal with IIR filter conv command approximate impulse response and makes it finite length.



f) The length of the impulse response is 28. Convolution increases the length of the signal by 28 - 1 = 27. However, filtering input signal does not change the length of the signal. To be able to obtain same length with filtering, we should do zero padding to input signal. So, we add 27 zero pad to our input to obtain same length signals.




```
[x,fs] = audioread('scales.wav');
b = [1, -4, 4, -1];
a = [1, 1.5, 1, 0.25];
h = impz(b,a);

y = conv(x,h);

len = 27; %length of impulse response minus one
x1 = [x;zeros(len,1)];
y1 = filter(b,a,x1);

subplot(211);
plot(y);
subplot(212);
plot(y1);
```

PROBLEM 2

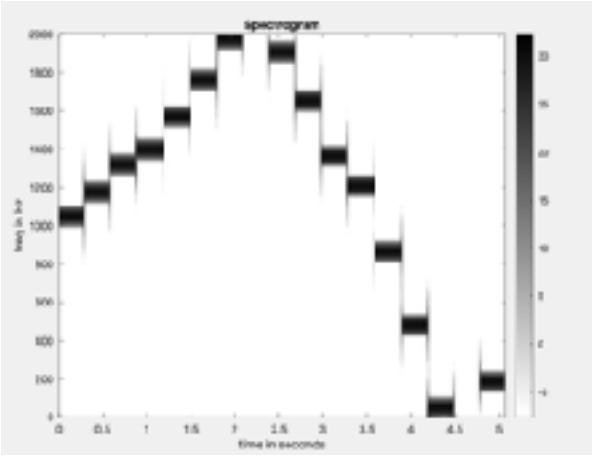
a)

i) Firstly we read the sound file. After that we listened both of them, then we plotted their spectrogram. When we compare the original version and downsampled version. There are differences in terms of melody which is because of aliasing. We think it is more obvious in 'music2.wav'. There is really obvious noise because of aliasing and also there is coming voice from the behind which is belong to the first seconds of the sound.

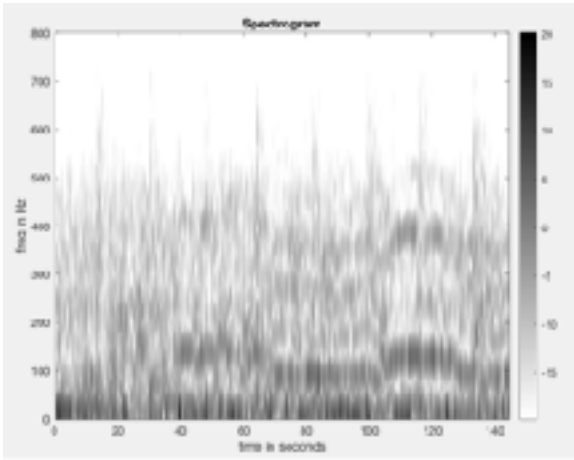
```
[x,fs] = audioread('scales.wav');
y = x(1:4:end);
soundsc(x,16000);
figure(1);
myspecgram(x,16000);

pause(2);

soundsc(y,4000);
figure(2);
myspecgram(y,4000);
```

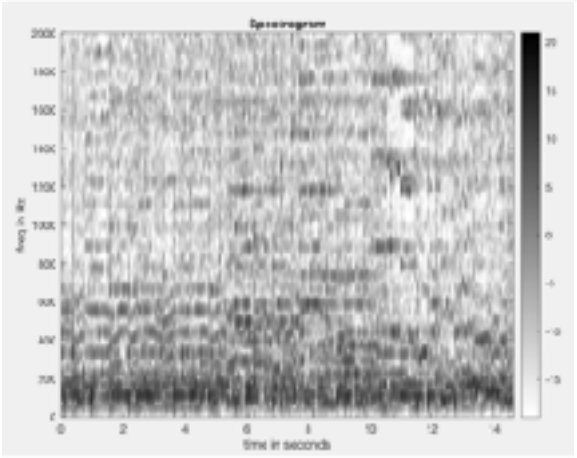
ii)

-When we read 'scales.wav' which originally sampled 16kHz and if we resample this signal with 4kHz we obtain aliasing because as we go further which will come higher frequencies in this case, we hear something that not normal. According to our ears, loudness of signal is going to go lower in some part and notes that, we hear, combined with different notes. We can perceive that there is an overlap. Spectrogram of the input and output signal is in the near part. As it can be seen, their

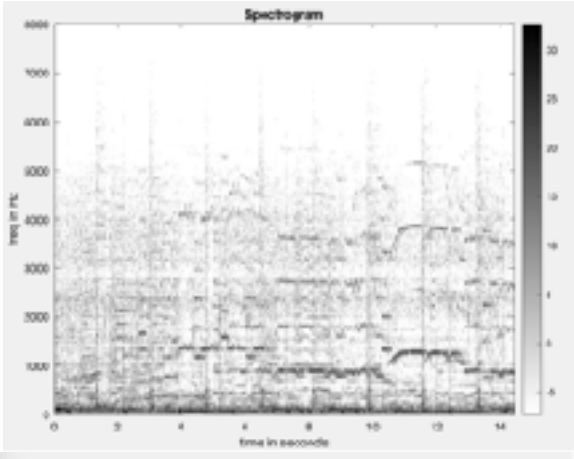


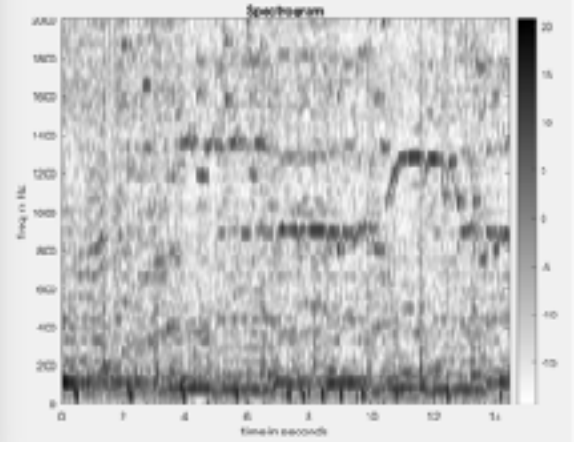
spectrogram graph are different from each other which is because of distortion by aliasing.





- When we examine the 'music1.wav' , we easily see that their spectrogram really different from each other. We determine that, sound became more hoarse, we thought that when we resample, we lost some of the frequency components and some of them suppressed because of downsampling.

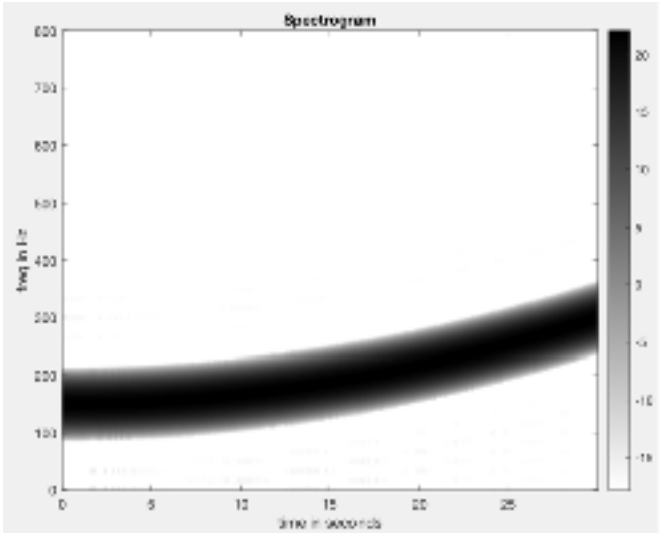


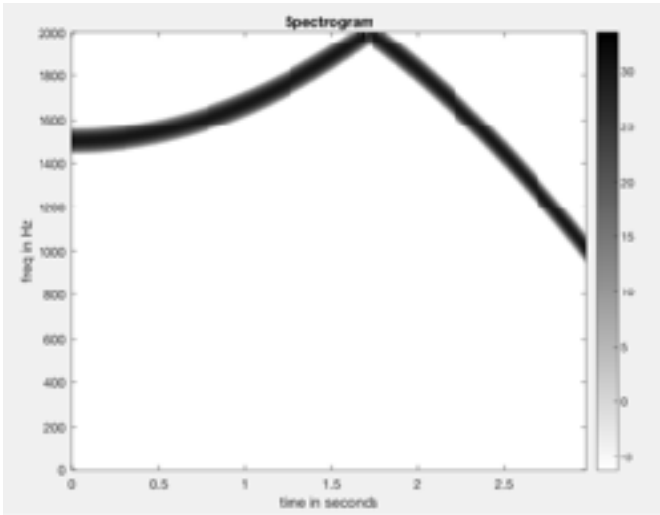


- When we examine the 'music2.wav' , we can easily see differences between these two. Firstly, especially lower frequencies and some other frequencies (~900, ~1300...) are emphasized. Secondly, according to our ears, voice has been scattered which means that, as can be seen in spectrogram 4kHz sampled version spread all around the spectrogram compared to 16kHz version, so this can be proof our opinion. According to our ears, we detect an aliasing in

background. Especially between middle and end of the sample it can be easily detected.

- When we read chirp.wav which originally sampled 16kHz and if we resample this signal with 4kHz we obtain aliasing. As we go further, magnitude of the signal is going to go lower as can be seen in plot. According to our ears, loudness of the signal is going to go lower however, story is not like that, because 4kHz version goes to very high frequency which is not detected easily, our ability to hear is going to low. Moreover, frequency of original signal goes higher as go further, however in resampled signal we hear that firstly frequency goes to higher then it goes back to lower frequency. Also original signal starts to lower frequency compared to resampled signal. Both of these cases, spectrogram can be seen as a proof of our opinion.

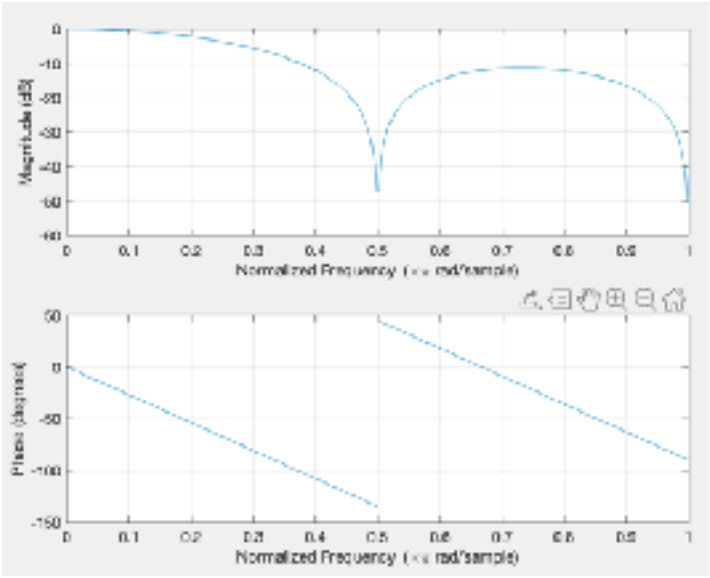




b) We examined all of the parts i) ii) iii) in each of examination of signal.

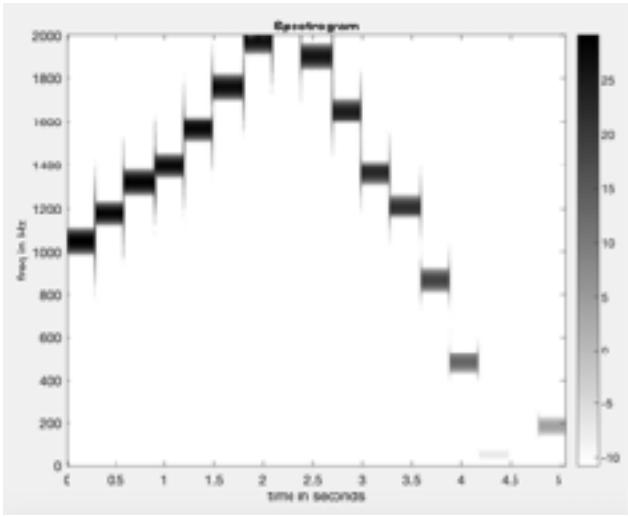
FILTER 1

```
[x,fs] = audioread('scales.wav');
b=1/4*ones(1,4);
a=[1];
xaa = filter(b,a,x);
yaa = xaa(1:4:end);

soundsc(yaa,4000);
figure(1);
myspecgram(yaa,4000);
figure(2);
freqz(b,a);
```

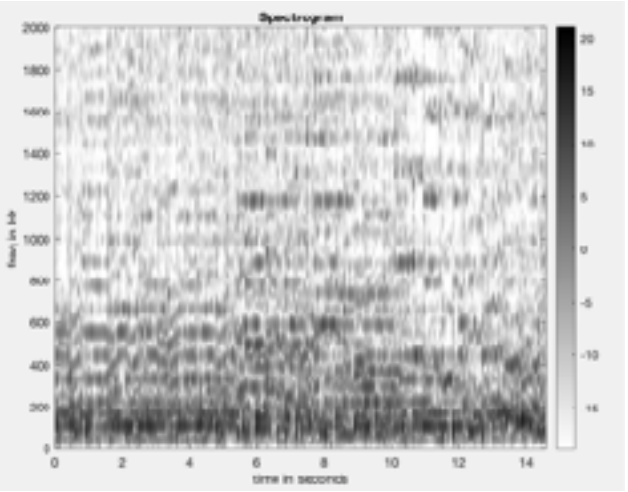


Spectrograms

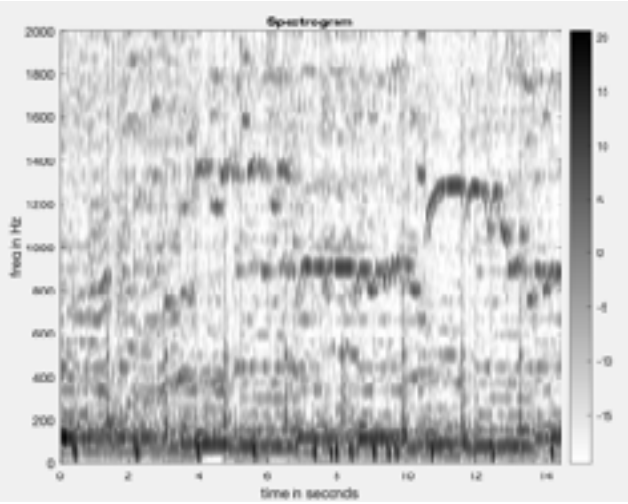
* It is spectrogram of 'scales.wav'. Distortion because of aliasing is slightly fixed by this filter. It seems in the right below part of the spectrogram.



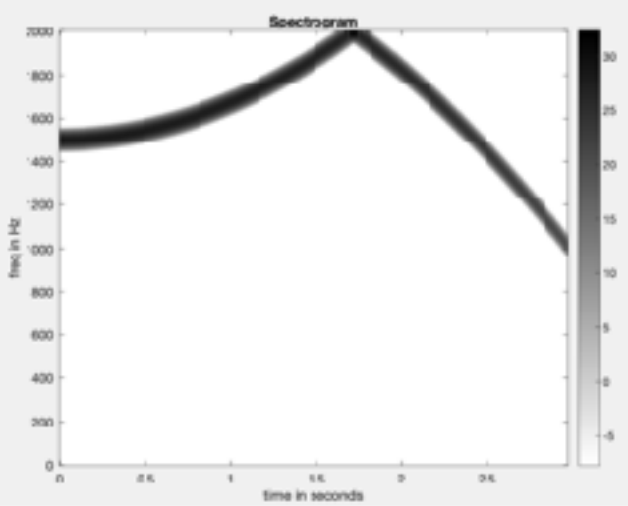
* It is spectrogram of 'music1.wav'. Distortion because of aliasing is slightly fixed by this filter.



* It is spectrogram of 'music2.wav'. Distortion because of aliasing is slightly fixed by this filter.

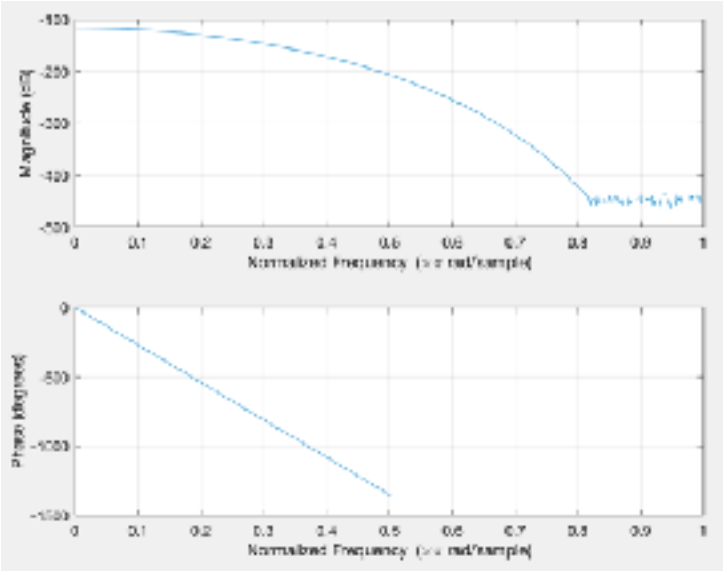


* It is spectrogram of 'chirp1.wav'. Distortion because of aliasing is slightly fixed by this filter.



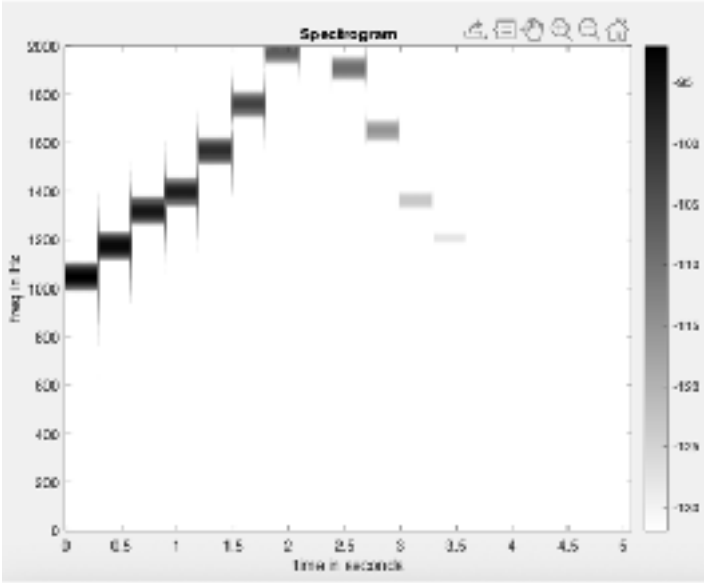
FILTER 2

```
[x,fs] = audioread('scales.wav');

[b,a]= butter(30,1/4);
a=[1];

xaa2=filter(b,a,x);
yaa2=xaa2(1:4:end);

soundsc(yaa2,4000);
figure(1);
myspecgram(yaa2,4000);
figure(2);
freqz(b,a);
```

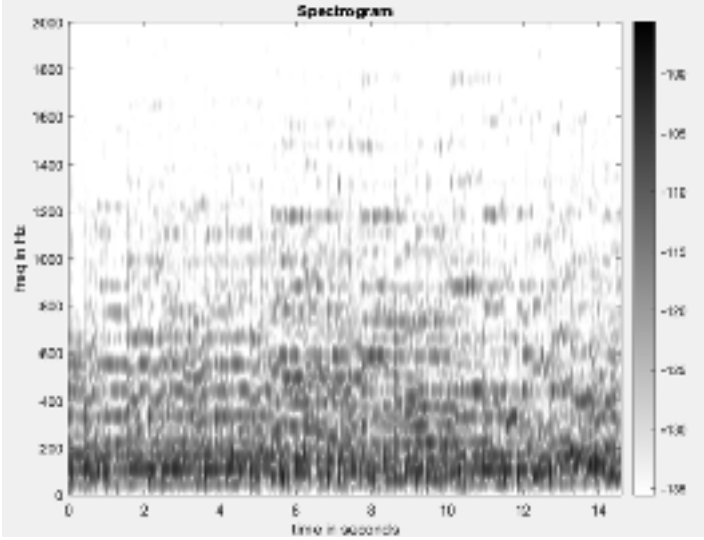


Spectograms

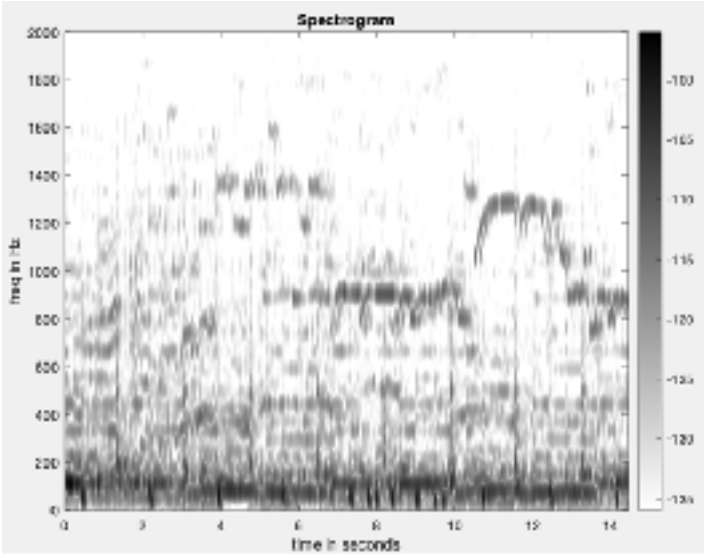
* It is spectrogram of 'scales.wav'. Distortion because of aliasing is more fixed by this filter when compared with filter 1. It seems in the right below part of the spectrogram.



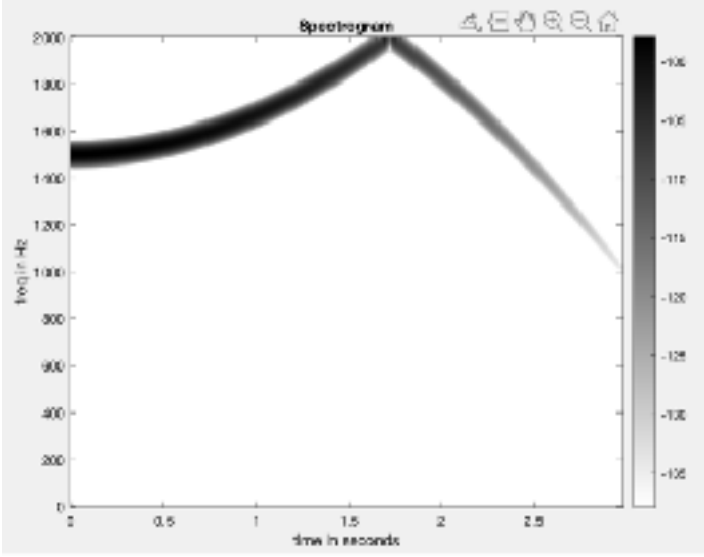
* It is spectrogram of 'music1.wav'. Distortion because of aliasing is more fixed by this filter when compared with filter 1.



* It is spectrogram of 'music2.wav'. Distortion because of aliasing is more fixed by this filter when compared with filter 1.

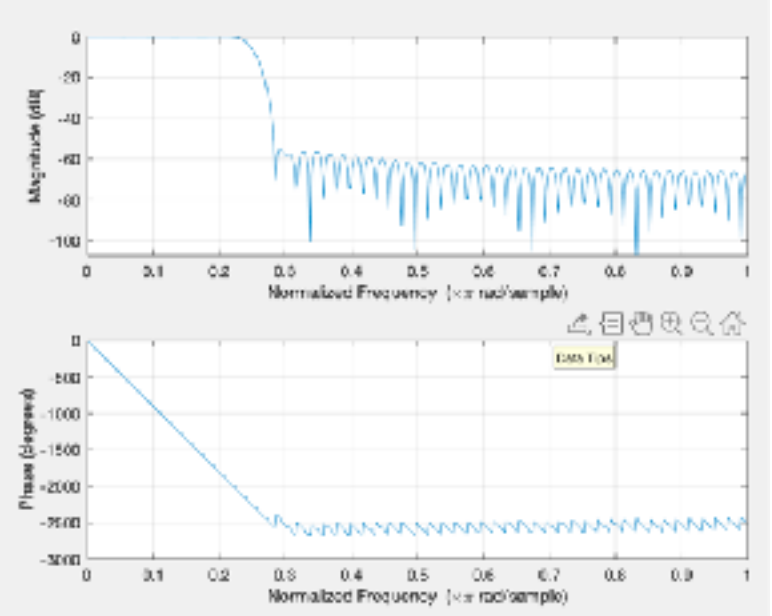


* It is spectrogram of 'chirp1.wav'. Deviation because of aliasing is more fixed by this filter when compared with filter 1.



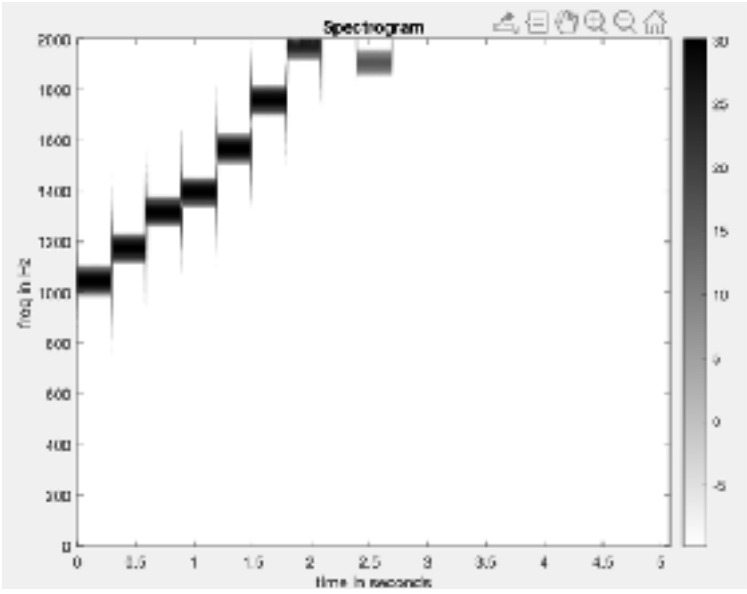
FILTER 3

```
[x,fs] = audioread('scales.wav');
b=fir1(100,1/4);
a=[1];
xaa3=filter(b,a,x);
yaa3=xaa3(1:4:end);
soundsc(yaa3,4000);
figure(1);
myspecgram(yaa3,4000);
figure(2) ;
freqz(b,a) ;
```

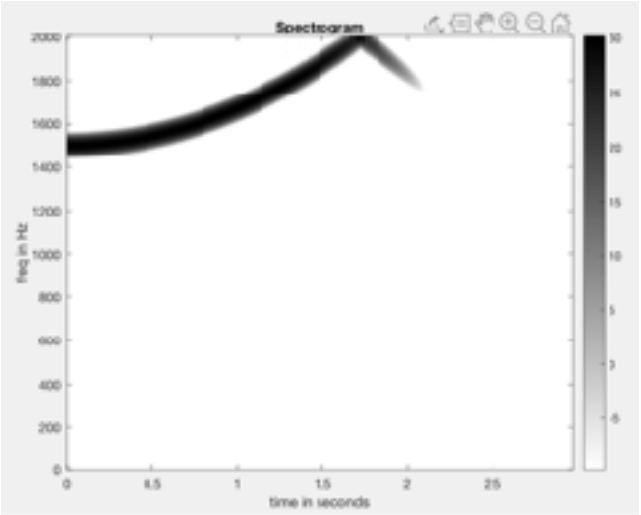


Spectrograms

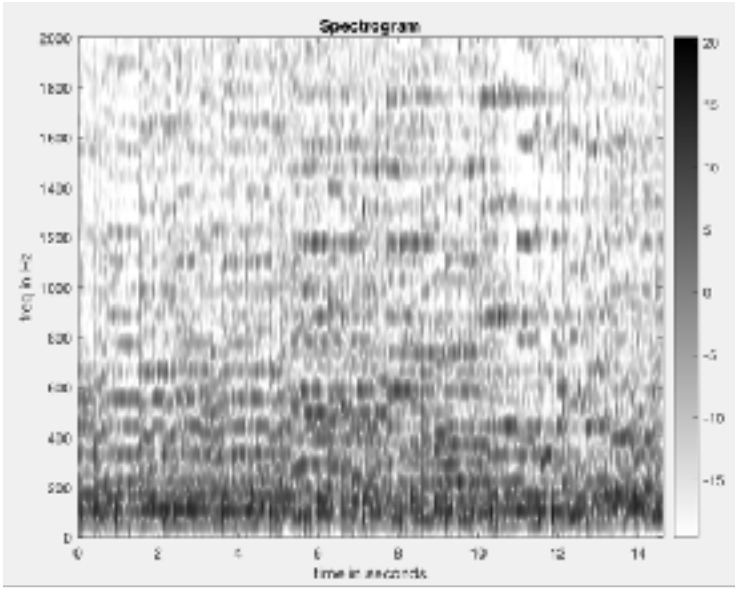
* It is spectrogram of 'scales.wav'. Deviation because of aliasing much more fixed by this filter when compared with other filters.It seems in the right below part of the spectogram. Distortion is minimized with this filter.



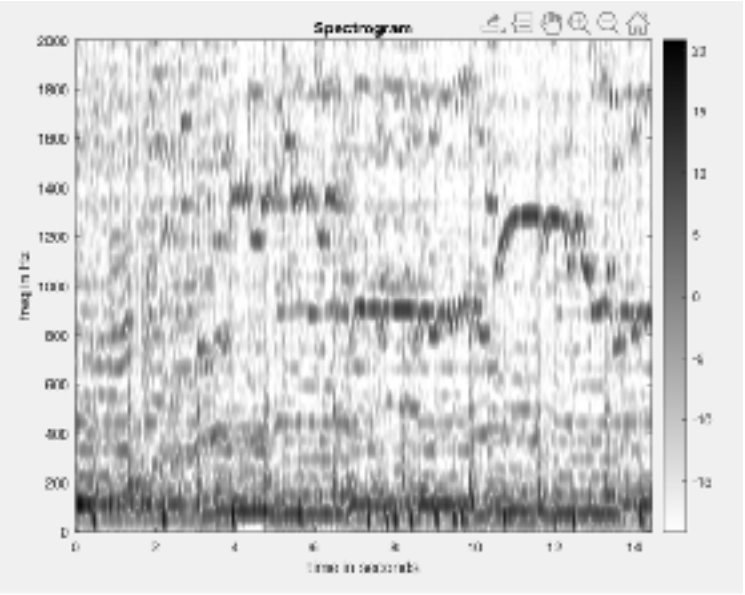
* It is spectrogram of 'chirp1.wav'. According to our observations, if we compare it only resampled signal, we can easily see improvements. Also if we compared to other filters we can easily see the most minimized deviation in this filter. Deviation is minimized because aliasing is much more fixed by this filter. It seems in the right below part of the spectrogram. Distortion is minimized with this filter.

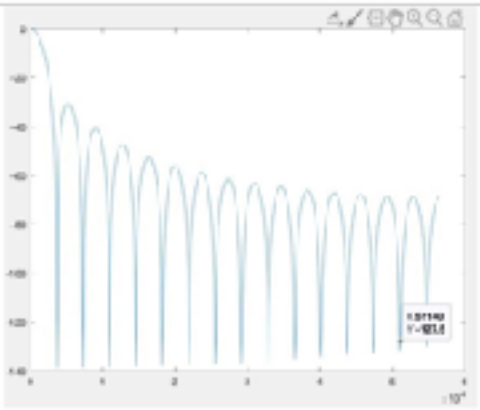


* It is spectrogram of 'msuic1.wav'. Deviation because of aliasing is much more fixed by this filter when compared with other filters. It is the best filter for reduce aliasing.



*It is spectrogram of 'music2.wav'. Deviation because of aliasing is much more fixed by this filter when compared with other filters. Distortion is minimized with this filter. It is the best filter for reduce aliasing.





```
[x,fs] = audioread('music1.wav');

m = 31;
b(1:m) = 1/m;
a = 1;


h = impz(b,a);
[h,w]=freqz(h);

plot(w*18000,20*log(abs(h)));
```

There is not any difference in magnitude and shape of the signal. However, it is x axis converted into Hz from normalized radian frequency (rad/ sample).

(This is part of problem 1 section b. We do it after we finish the homework and when we try to add it where t should be in, everything was tangle each other, therefore we add it here)

Thanks…