EE352 Digital Signal Processing

Project 4

Name: Aibek Bekbergen

ID: 20202012

1. 6 Band Equalizer:

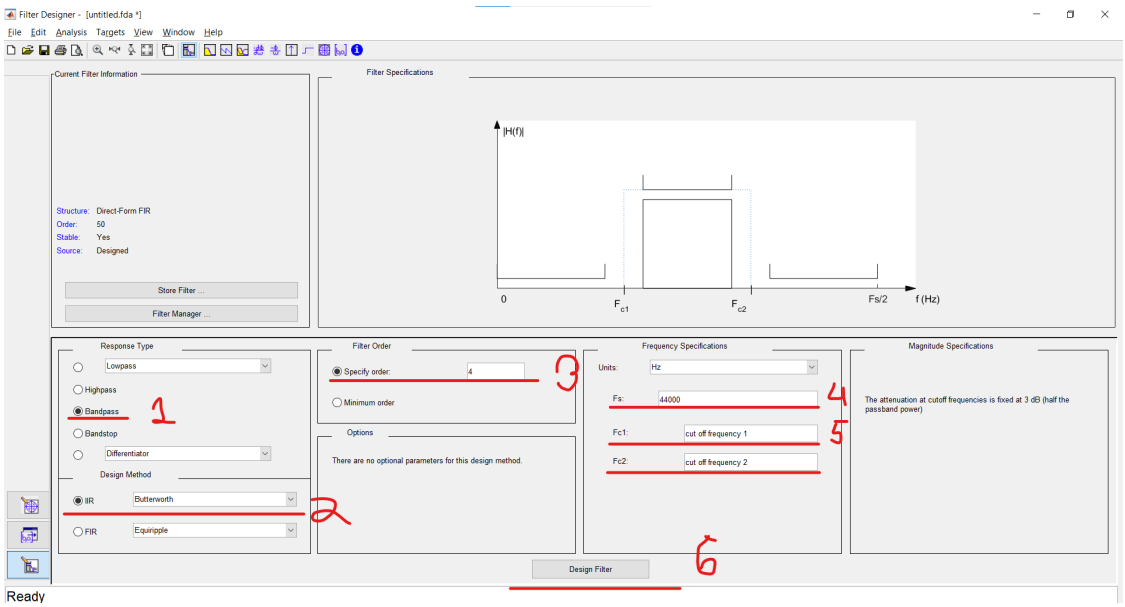
Problem 1.1)

In this Part of the project, we designed 6 bands using 4th order Butterworth filter.

To do that, we need to run the following command in command window.

```
>> fdatool
```

This will open filter designer problem. After that we have to create 6 bands as follows:



The following window will pop up. And we choose required parameters in points 1-5. And then press Design filter, then we need to export the following filter to our workspace in MATLAB. Then, by typing:

```
>> [Bi, Ai] = sos2tf(SOSi, Gi)
```

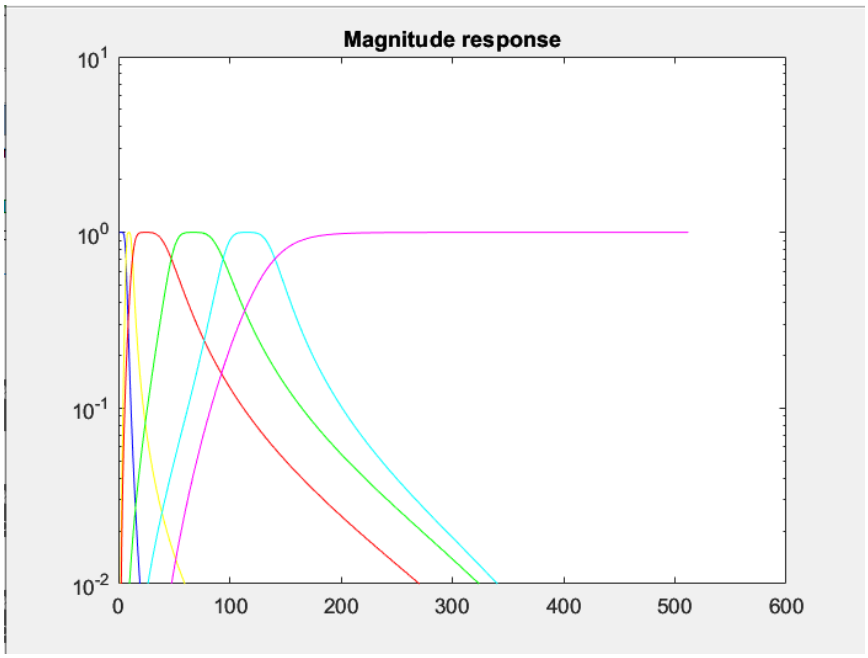
Instead of I, we put ith filter index. I chose from 1 to 6. Now we need to save all the parameters by using the following command:

>>save m20202012_4.mat B6 B5 B4 B3 B2 B1 A6 A5 A4 A3 A2 A1

Then, by using the following code, we can plot the magnitude response of our 6 filters:

```
%% Problem 1.1
load m20202012_4.mat

H1 = freqz(B1,A1);
% figure; plot(abs(h1)); % this is to obtain linear scale, not necessary
figure; semilogy(abs(H1), 'b'); hold on % dB scale

H2 = freqz(B2,A2);
% figure; plot(abs(h2)); % this is to obtain linear scale, not necessary
semilogy(abs(H2), 'y'); hold on % dB scale

H3 = freqz(B3,A3);
% figure; plot(abs(h3)); % this is to obtain linear scale, not necessary
semilogy(abs(H3), 'r'); hold on % dB scale

H4 = freqz(B4,A4);
% figure; plot(abs(h4)); % this is to obtain linear scale, not necessary
semilogy(abs(H4), 'g'); hold on % dB scale

H5 = freqz(B5,A5);
% figure; plot(abs(h5)); % this is to obtain linear scale, not necessary
semilogy(abs(H5), 'c'); hold on % dB scale

H6 = freqz(B6,A6);
% figure; plot(abs(h6)); % linear scale
semilogy(abs(H6), 'm'); title('Magnitude response'); hold off
ylim([0.01 10]);
```

As the result we obtain the following plot of Magnitude Responses:

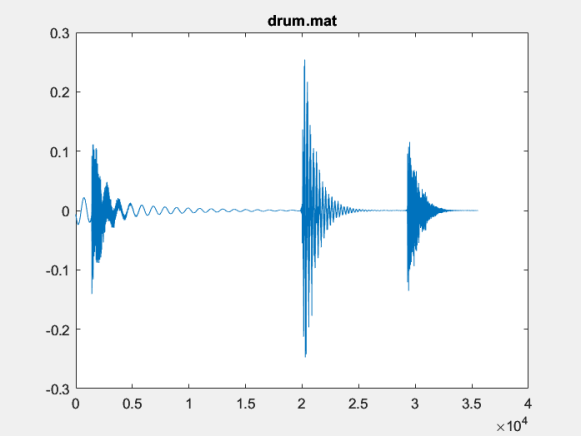


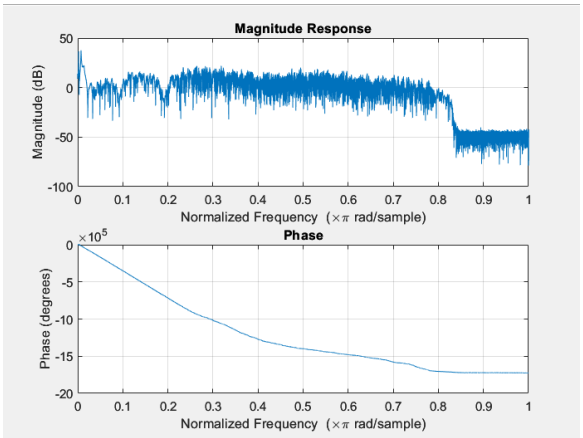
Problem 1.2)

In this part of the project, by using the following code

```
%% Problem 1.2
clear all; close all; clc;

load drum.mat
load m20202012_4.mat

y1 = filter(B1, A1, a);
y2 = filter(B2, A2, a);
y3 = filter(B3, A3, a);
y4 = filter(B4, A4, a);
y5 = filter(B5, A5, a);
y6 = filter(B6, A6, a);
y = y1+y2+y3+y4+y5+y6;

sound(y, Fs);
y = y(100000 : 100000+35536);
figure; plot(y); title("drum.mat");
figure; freqz(y,1,35536); title("Magnitude Response");
```

We obtain the following results:





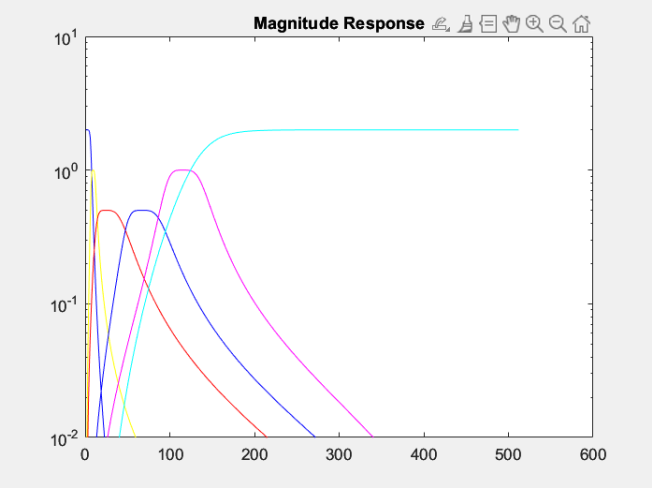
In the first plot we can see our input signal, and in second plot we can see input signal's magnitude and phase response.

Problem 1.3)

In this part of the project, by using the following code:

```
%%  Problem 1.3
clear all; close all; clc;

load drum.mat
load m20202012_4.mat

H1 = freqz(B1*2,A1);
% figure; plot(abs(H1)); % this is to obtain linear scale, not necessary
figure; semilogy(abs(H1), 'b'); hold on % dB scale

H2 = freqz(B2,A2);
% figure; plot(abs(H2)); % this is to obtain linear scale, not necessary
semilogy(abs(H2), 'y'); hold on % dB scale

H3 = freqz(B3*0.5,A3);
% figure; plot(abs(H3)); % this is to obtain linear scale, not necessary
semilogy(abs(H3), 'r'); hold on % dB scale

H4 = freqz(B4*0.5,A4);
% figure; plot(abs(H4)); % this is to obtain linear scale, not necessary
semilogy(abs(H4), 'b'); hold on % dB scale

H5 = freqz(B5,A5);
% figure; plot(abs(H5)); % this is to obtain linear scale, not necessary
semilogy(abs(H5), 'm'); hold on % dB scale

H6 = freqz(B6*2,A6);
% figure; plot(abs(H6)); % this is to obtain linear scale, not necessary
semilogy(abs(H6), 'c'); title('Magnitude Response'); hold off % dB scale
ylim([0.01 10]);
```

This almost the same to the one in Problem 1.1, however we multiplied signals by their gains. And as the result, we obtain the following plot:



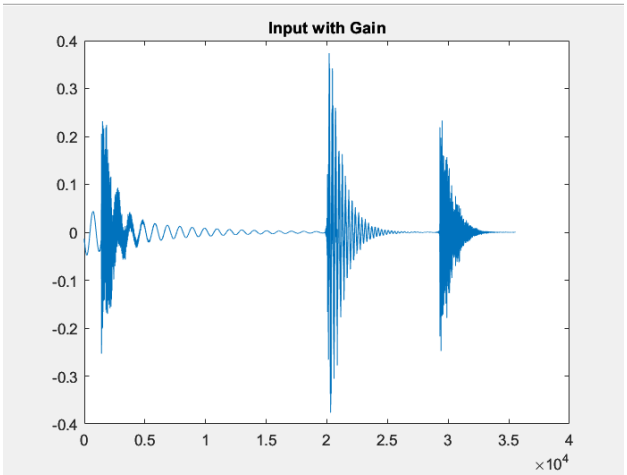
As we can see the magnitude responses of the signals multiplied by their gains is different from the regular magnitude responses.

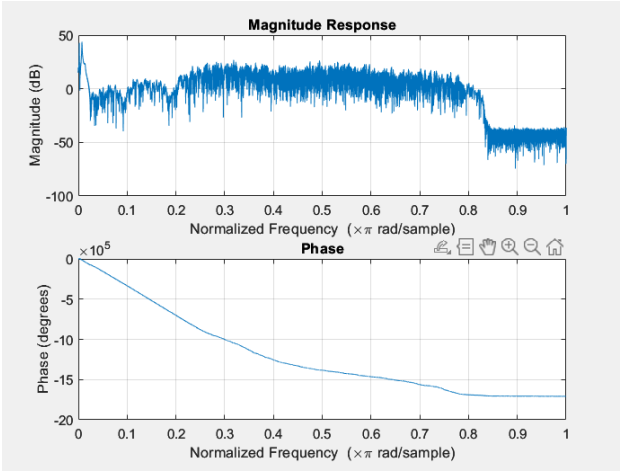
Problem 1.4)

In this part of the project, by using the following code:

```
%% Problem 1.4
y1 = filter(B1*2, A1, a);
y2 = filter(B2, A2, a);
y3 = filter(B3*0.5, A3, a);
y4 = filter(B4*0.5, A4, a);
y5 = filter(B5, A5, a);
y6 = filter(B6*2, A6, a);

y = y1+y2+y3+y4+y5+y6;
sound(y, Fs);

y = y(100000 : 100000+35536);
figure; plot(y); title("Input with Gain");
figure; freqz(y,1,35536); title("Magnitude Response");
```

As the result, we obtain the following plots:





As we can see from these plots, the input signal amplitude is higher in some parts and lower in some parts comparing to the result we obtained in Problem 1.2. That is because of the gains that we multiplied to our input signals. Magnitude and phase responses are the same.

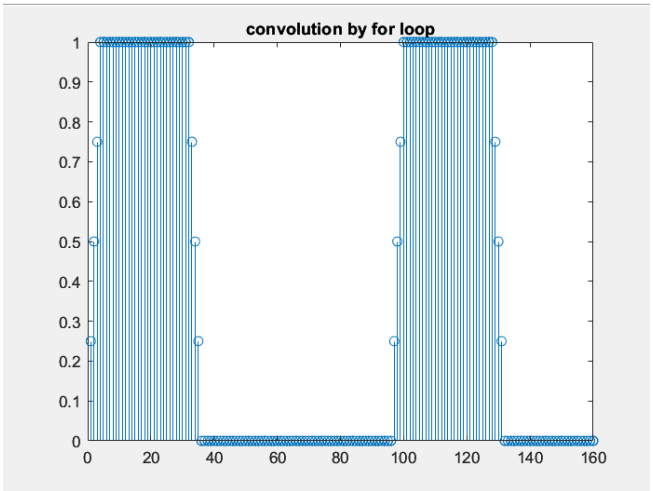
2. Filter Implementation

Problem 2.1)

In this part of the project, by using the following code:

```
%% Problem 2.1

clear all; close all; clc;

h = [1,1,1,1]/4; % our filter
x = [ones(1,32),zeros(1,32),zeros(1,32),ones(1,32)]; % input signal

% create zeros of length 32 * 4 = 128
q = zeros(1, 128);

% we create boundaries  conditions by adding q to the left
% and right of our input signal
w = [q,x,q];

y = zeros(1, 160);

% now convolve our signal and filter
for n = 1 : 160
    sum = 0;
    for k = 1 : 4
        sum = sum + h(k)*w(n-k+128+1);
    end
    y(n) = sum;
end
figure; stem(y); title("convolution by for loop")
```

As the result we obtain the following discrete time signal:



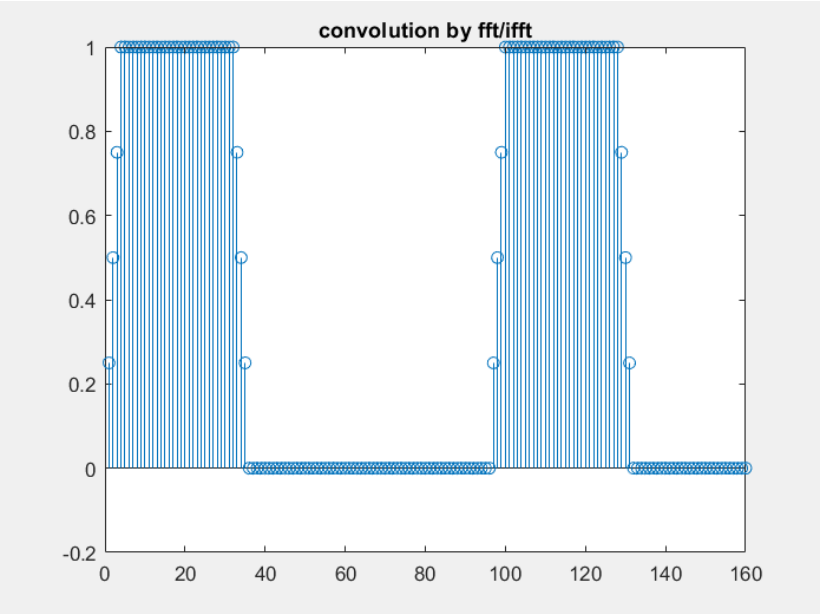
Problem 2.2)

In this part of the project, by using the following code:

```
%% Problem 2.2
Hw = fft(h, 160);
Xw = fft(x, 160);

Yw = Hw.*Xw;
y = real(ifft(Yw));
figure; stem(y); title("convolution by fft/ifft");
```

As you can see, we took the FFT of our filter and input signal. We know that convolution in time domain is multiplication in frequency domain, so we multiplied the results from FFT's and took inverse FFT to get the signal in time domain again.

As the result we obtain:



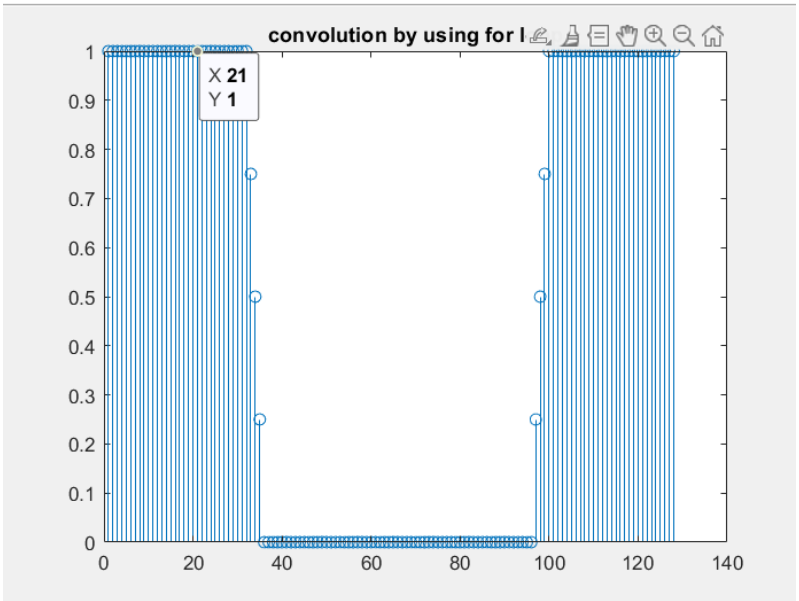
As you can see, we obtained the same result as in Problem 2.1.

Problem 2.3)

In this part of the project, by using the following code:

```
%% Problem 2.3
w2  = [x, x, x]; % since input is now periodic

y3 = zeros(1, 128);

for n = 1 : 128
    sum = 0;
    for k = 1 : 4
        sum = sum + h(k)*w2(n-k+128+1);
    end
    y3(n) = sum;
end
figure; stem(y3(1:128)); title("convolution by using for loop");
```

As the result we obtain:

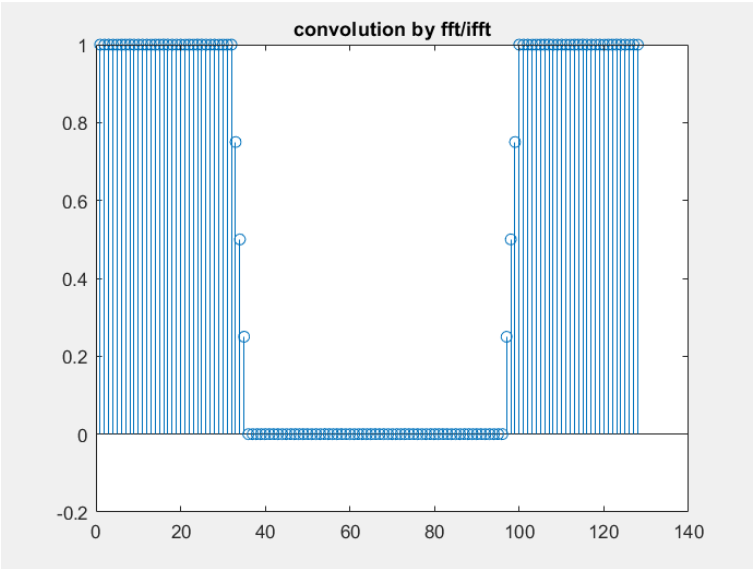


Problem 2.4)

Analogically, following the same process as in Problem 2.2, using the following code:

```
%% Problem 2.4
Hw2 = fft(h, 128);
Xw2 = fft(w2, 128);
result = ifft(Hw2.*Xw2);
figure; stem(result); title("convolution by fft/ifft");
```

As the result we obtain:



As you can see, we obtain the same result as in Problem 2.3.

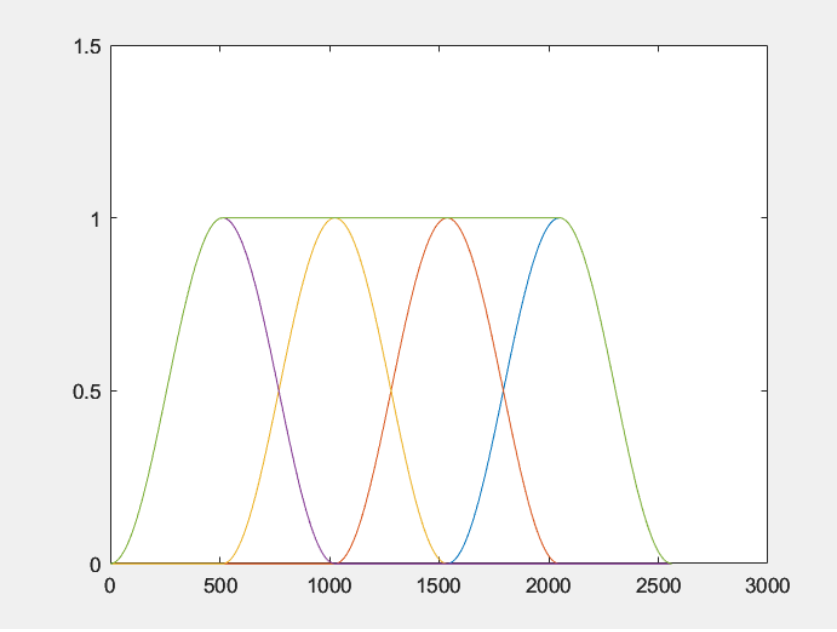
Problem 3.1)

In this part of the project, by using the following code:

```
%% Problem 3.1

clear all; close all; clc;

L = 1025;
M = 512;

P = M*3 + L;
w = hann(L);
w1 = zeros(P, 1);
w2 = zeros(P, 1);
w3 = zeros(P, 1);
w4 = zeros(P, 1);

w1((1:L)+3*M) = w;
w2((1:L)+2*M) = w;
w3((1:L)+M) = w;
w4((1:L)) = w;

x = w1 + w2 + w3 + w4;
t = 1:P;

figure; plot(t, w1, t, w2, t, w3, t, w4, t, x)

V = axis;
V(4) = 1.5;
axis(V)
```

MATLAB does not have negative indices in arrays, so we shifted the input signal by 3M to the right, and then computed other windows w_p.

As the result, we obtain:



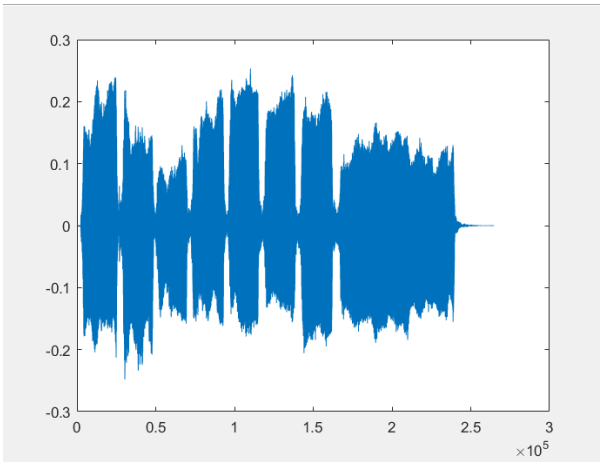
As you can see, we obtained 4 aliasing hanning windows. We also can see that the sum of two overlapping hanning windows is one long hanning window with the same amplitude, but longer range. From these plots that we can get long hanning window (labeled with green color), which can be used for long input signals.

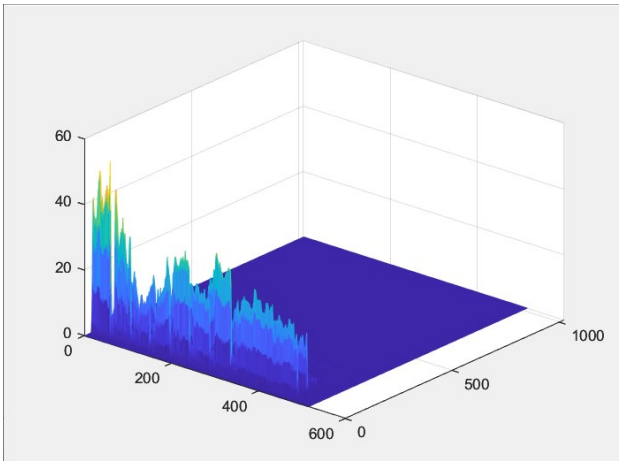
Problem 3.2)

In this part of the project, by using the following code:

```
%% Problem 3.2
clear all; close all; clc;

load flute.mat
figure; plot(a)

N = length(a);
L = 1025;
P = 512;
w = hann(L);

K = length(1:P:N)-2;
S = zeros(2048, K);

for p=1:K
    x = a((1:L) + (p-1)*P);
    x = w.*x;
    X = fft(x, 2048);
    S(:, p) = X;
end

figure; mesh(abs(S(1:1024,:))); view([0,0,1]);
```

As the result we obtain:





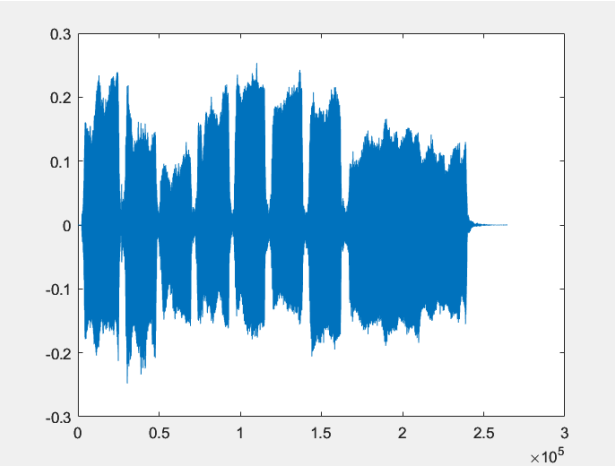
We obtained our input signal and it's spectrogram.

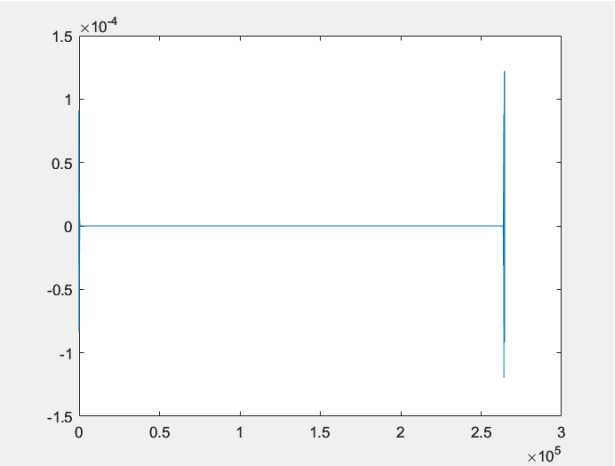
Problem 3.3)

In this part of the project, by using the following code:

```
%% Problem 3.3:

y = zeros(N, 1);

for p = 1:K
    X = S(:, p);
    x = real(ifft(X));
    y((1:L) + (p-1)*P) = y((1:L) + (p-1)*P) + x(1:L);
end

figure; plot(y)

figure; plot(y - a);
```

As the result, we obtain:





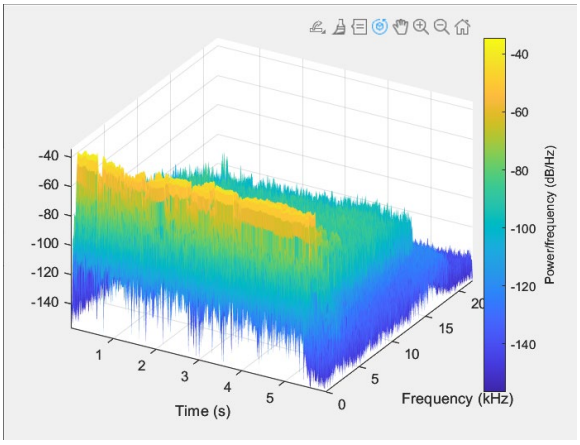
The plot on the right is the signal $y[n]$ that we obtained manually. The plot on the right is the $y[n] - a[n]$. As we got zero signal, we can conclude that we got our signal back.

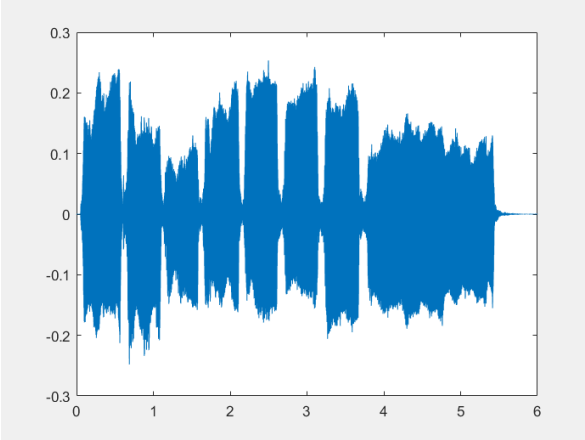
Problem 3.4)

In this part of the project, by using the following code:

```
%% Problem 3.4
clear all; close all; clc;

load flute.mat
figure; plot(a);

N = length(a);
x = (1:N)/Fs;
[s,w,t] = spectrogram(a, hann(1025), 512, 2048, Fs, 'yaxis');

figure(1); plot(x, a)
figure(2); plot(t)

figure(3); spectrogram(a, hann(1025), 512, 2048, Fs, 'yaxis');
```

We obtain the plot of our flute input signal and its spectrogram:





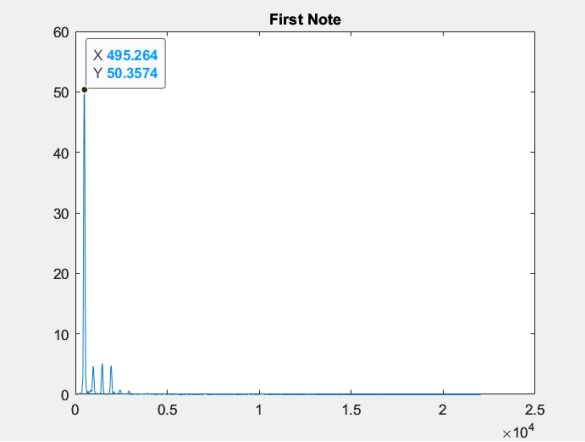
Now to find the first and fifth signal magnitude responses from the spectrogram, we implemented following for loops:

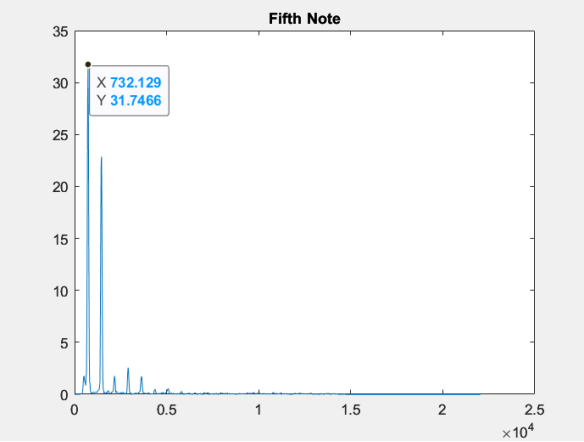
```
% to find the sample of the first note with maximum peak
for i = 0:1:50
    time = i;
    S = s(:, time);
    figure(2); stem(w, abs(S));
    title(i)
    pause(1);
end
% the 25th sample has the maximum peak

% to find the sample of the fifth note with maximum peak
for i = 210:1:250
    time = i;
    S = s(:, time);
    figure(2); stem(w, abs(S));
    title(i)
    pause(1);
end
% the 210th sample has the maximum peak
```

Now, we can plot the magnitude response at the sample we found:

```
time1 = 25;
time2 = 210;

S1 = s(:, time1);
S2 = s(:, time2);

figure(3); spectrogram(a, hann(1025), 512, 2048, Fs, 'yaxis');
figure(4); plot(w, abs(S1)); title("First Note")
figure(5); plot(w, abs(S2)); title("Fifth Note")
figure(6); plot(w, abs(S1), w, abs(S2))

t(time1)
t(time2)
```

As the result we obtain:





As we can see the first peak of the first note is at ~495 Hz, and the first peak of the fifth note is at ~732 Hz.

By looking at the fundamental frequencies of notes, we can conclude that first not is B_4, and the fifth note is $F^{\#}{}_5/G^{b}{}_4$.