# Signals final project

# **MATLAB**

| ID | الاسم |
|---|---|
| 19015534 | بيشوي إبراهيم منير عجيب |
| 19016193 | كيرلس نبيل وهيب غطاس |
| 19016212 | مارفين أيوب عوض أيوب |
| 19015991 | عبدالقادر مسعد عبدالقادر |

# **First stage (**transmitter**)**

- First we take the number of seconds that he wants to perform in from the user.

```
ns=input('please enter the number of the secounds needed  ');
```

- Second we will enter the sound from our files and use the function aduioread for inserting the data file.

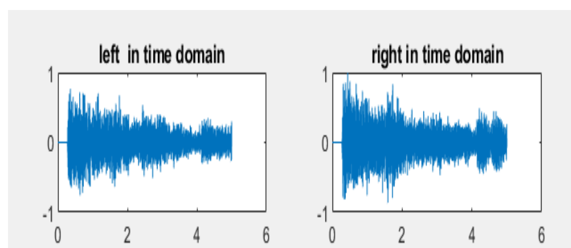
```
fprintf('here we have loooded our music .\n');
[Y,Fs]=audioread('moonnight.mpeg');
Y=Y(1:ns*Fs,:);
```

- Third we will play the seconds we have chosen and pause the program to hear the music.

```
fprintf('the music is playing\n');
sound(Y,Fs);
pause(ns);
fprintf('the music is finished\n ');
```

- Fourth
    1. we will plot the right and the lift column of the sound wave in time domain .

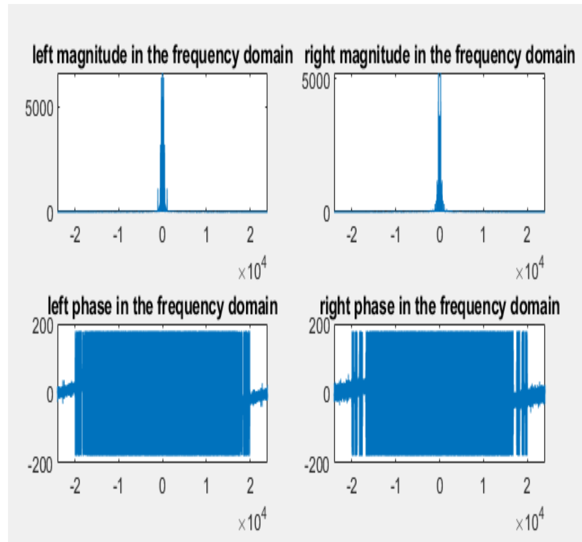
    ```
    t= linspace(0,ns,ns*Fs);
    figure;
    subplot(3,2,1);
    plot(t,Y(:,1));
    title ('left  in time domain');
    subplot (3,2,2);
    plot(t,Y(:,2));
    title('right in time domain');
    ```

    2. we will change the signal by fourier transform to frequency domain then plot the right and the left column of the magnitude and phase.

    ```
    Fvec=linspace (-Fs/2,Fs/2,ns*Fs);
    Ys=fftshift(fft (Y));
    ymag=abs(Ys);
    yphase=angle(Ys)*(180/pi);
    ```

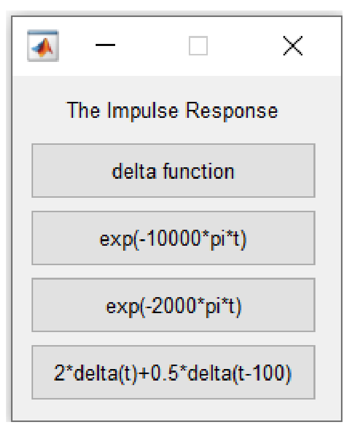
```
subplot(3,2,3);
plot(Fvec,ymag(:,1));
 title('left magnitude in the frequency domain');
subplot(3,2,4);
plot(Fvec,ymag(:,2));
title('right magnitude in the frequency domain');
subplot(3,2,5);
plot(Fvec,yphase(:,1));
title('left phase in the frequency domain');
subplot(3,2,6);
plot(Fvec,yphase(:,2));
title('right phase in the
frequency domain');
```




## second stage (channel)

- First a menu will pop up for the user to make him / her chooses the impulse response needed for the channel

```
F=menu('The Impulse Response','deltafunction','exp(-
10000*pi*t)','exp(-2000*pi*t)','2*delta(t)+0.5*delta(t-100)');
switch F
    case 1
        h_1=zeros (ns*Fs,1);
        h_1(1)=1;
        Yt(:,1)=conv(Y(:,1),h_1);
        Yt(:,2)=conv(Y(:,2),h_1);
    case 2
        h_2=exp(-10000*pi*t);
        Yt(:,1)=conv(Y(:,1),h_2);
        Yt(:,2)=conv(Y(:,2),h_2);
    case 3
       h_3=exp(-2000*pi*t);
        Yt(:,1)=conv(Y(:,1),h_3);
```



```
        Yt(:,2)=conv(Y(:,2),h_3);
    case 4
        h_4=zeros (ns*Fs,1);
        h_4(1)=2; h_4(Fs+1)=0.5;
        Yt(:,1)=conv(Y(:,1),h_4);
        Yt(:,2)=conv(Y(:,2),h_4);
end
```

1. The first case of the delta function the sound doesn't change because the convolution with the delta results the same signal.

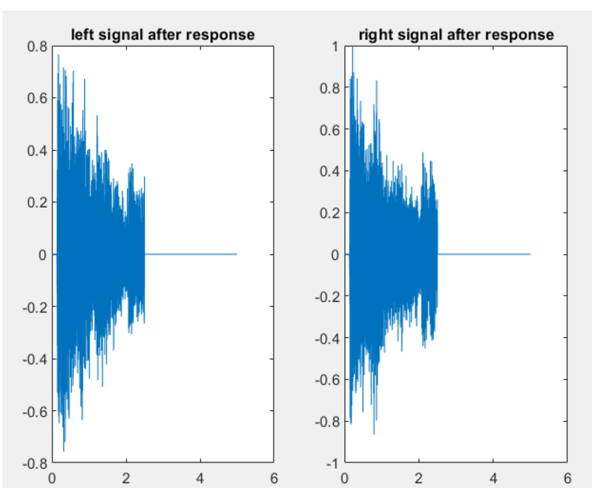
2. The second one exp(-10000*pi*t) it made the sound ramp from low to high.

3. The third one exp(-2000*pi*t) it made the sound higher but no ramp noticed.

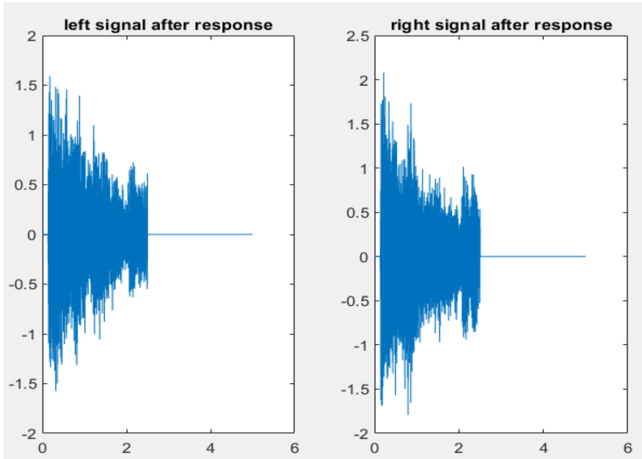
4. The last 2*delta(t)+0.5*delta(t-100) it made an echo of lower sound runs after the original.

- Second we will plot the right and the lift column of the sound wave in time domain after convolution.

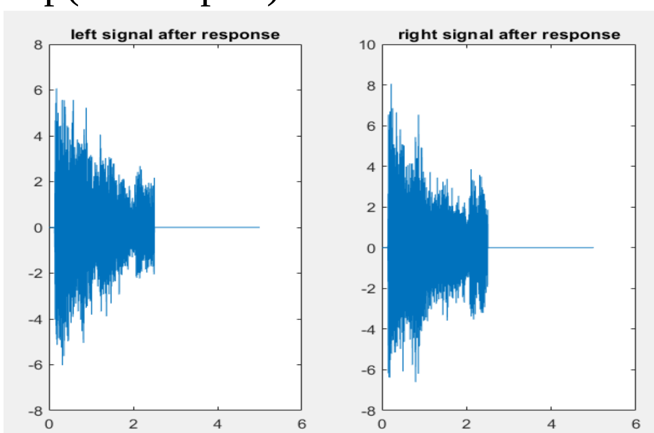
1. delta function



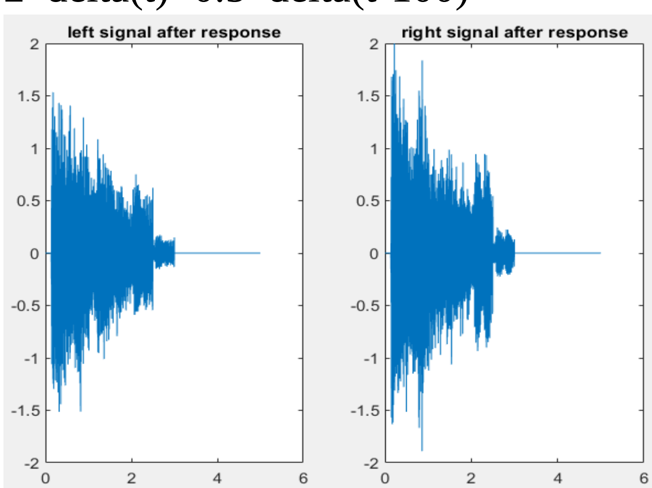
2. exp(-10000*pi*t)




3. exp(-2000*pi*t)




4. 2*delta(t)+0.5*delta(t-100)



- Third we will play the music after the convolution and tries to notice the deference between the all 4 convoluted signals for the same sound track.

## Third stage (Noise)

- First we will make the user enter the value of the segma which is better to be less than 0.1 to be able to hear the noise with out hurting your ears and then we will make our noise and add it to the original signal.

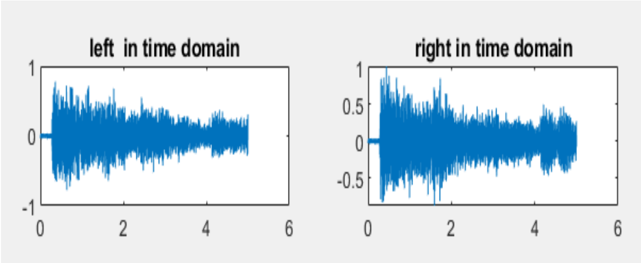
```
segma=input('please enter the value of the segma in range < 0.1
');
z=segma*randn(length(D(:,1)),1);
D(:,1)=D(:,1)+z;
D(:,2)=D(:,2)+z;
```

- Second we will hear the music signal after adding the noise.

```
fprintf('the music is playing aftetr adding noise\n');
sound(D,Fs);
pause(ns);
fprintf('the music is finished\n ');
```

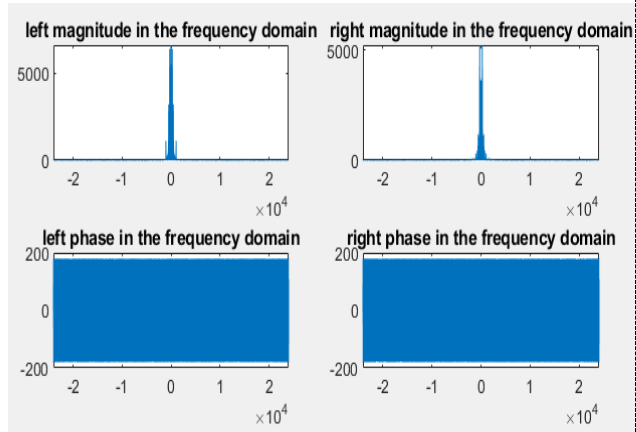
- Third

1. we will plot the right and the lift column of the sound wave in time domain after adding the nose.




```
t= linspace(0,ns,ns*Fs);
figure;
subplot(3,2,1);
plot(t,D(:,1));
title ('left  in time domain');
subplot (3,2,2);
plot(t,D(:,2));
title('right in time domain');
```

2. we will change the signal by fourier transform to frequency domain then plot the right and the left column of the magnitude and phase after adding the nose.




```
Fvec=linspace (-
Fs/2,Fs/2,ns*Fs);
Ds=fftshift(fft (D));
Dmag=abs(Ds);
Dphase=angle(Ds)*(180/pi);
subplot(3,2,3);
plot(Fvec,Dmag(:,1));
title('left magnitude in the frequency domain');
subplot(3,2,4);
plot(Fvec,Dmag(:,2));
title('right magnitude in the frequency domain');
subplot(3,2,5);
plot(Fvec,Dphase(:,1));
title('left phase in the frequency domain');
subplot(3,2,6);
plot(Fvec,Dphase(:,2));
title('right phase in the frequency domain');
```

# Fourth stage (receiver)

- first to avoid removing from the original signal we generate a zero signal have the same length of the original one then at the frequency domain we calculate the no. of the samples by knew how many sample in one hertz the calculate the start and the end of the filter with the given values from (-3.4KHz) to (3.4KHz) then we multiply it to a copy from the original to filter it then we return it from frequency domain to time domain.

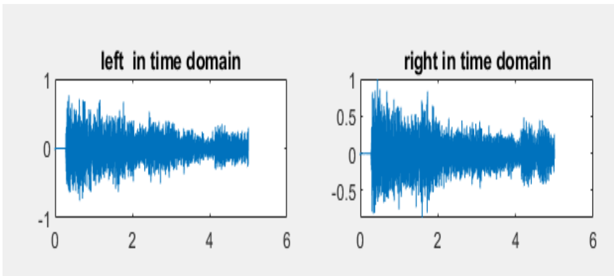
```
l_filter=zeros(ns*Fs,1);
X1=ns*((-3400)-(-Fs/2));
X2=ns*((3400)-(-Fs/2));
l_filter(X1:X2)=1 ;
```

```
s=Ds;
s(:,1)=Ys(:,1).*l_filter;
s(:,2)=Ys(:,2).*l_filter;
Sf=real(ifft(ifftshift(s)));
```

- Second we will hear the music signal after trying to remove the noise by filtering it.

```
fprintf('the music is playing aftetr trying to remove the noise\n');
sound(Sf,Fs);
pause(ns);
fprintf('the music is finished\n ');
```

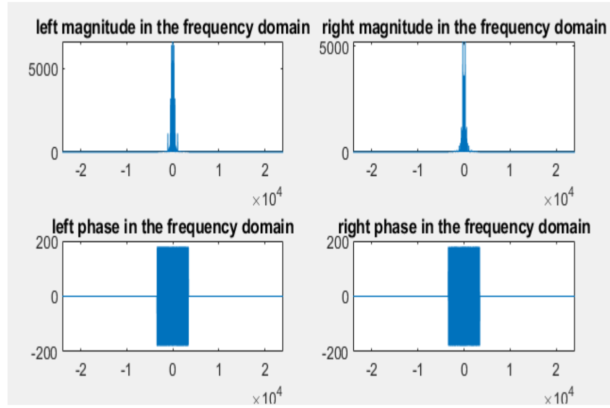
- Third




1. we will plot the right and the lift column of the sound wave in time domain after adding trying to remove the noise by filtering it.

```
t= linspace(0,ns,ns*Fs);
figure;
subplot(3,2,1);
plot(t,Sf(:,1));
title ('left  in time domain');
subplot (3,2,2);
plot(t,Sf(:,2));
title('right in time domain');
```

2. we will change the signal by fourier transform to frequency domain then plot the right and the left column of the magnitude and phase after trying to remove the noise by filtering it.




```
SFmag=abs(s);
SFphase=angle(s)*(180/pi);
subplot(3,2,3);
plot(Fvec,SFmag(:,1));
title('left magnitude in the frequency domain');
subplot(3,2,4);
plot(Fvec,SFmag(:,2));
title('right magnitude in the frequency domain');
subplot(3,2,5);
plot(Fvec,SFphase(:,1));
title('left phase in the frequency domain');
subplot(3,2,6);
plot(Fvec,SFphase(:,2));
title('right phase in the frequency domain');
```