MATLAB exercises report:

Software used : GNU Octave

The purpose of those exercises is to help students, to deal with the IIR and FIR filtering procedure and parametrization. Be able understand the determination type of filter for the different operations. The exercises have to be completed in MATLAB or OCTAVE where functions referring of those filters have already been implemented .

We went through the usage of the different filters and setting during the course and now we should filtered out signals : Digital Filters & Noise Removal; and also work with Floating Point Numbers concept.

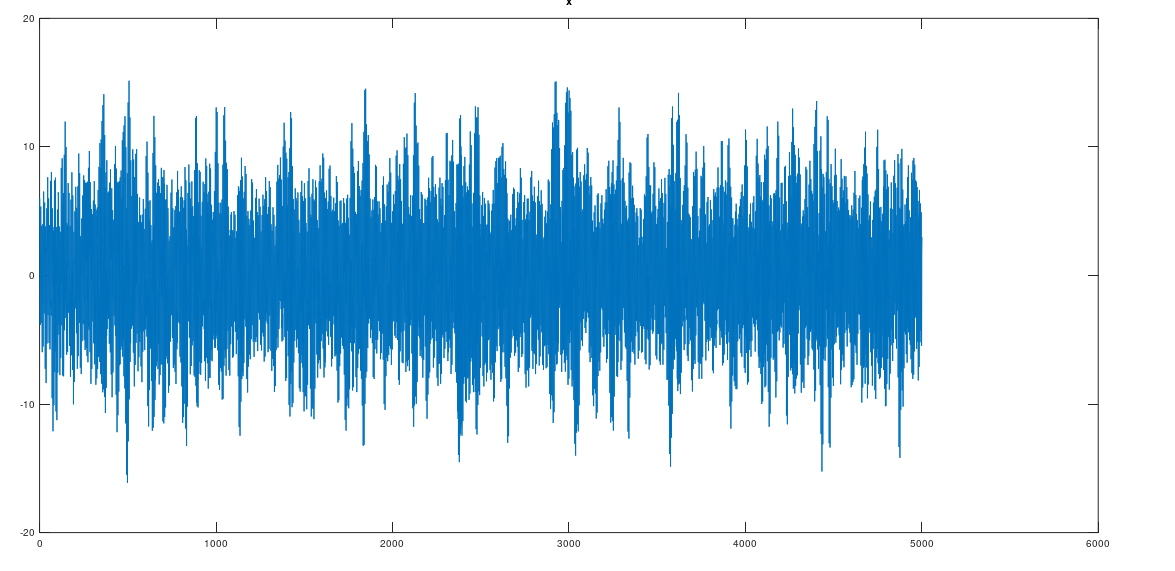
Exercise 2: Digital Filters & Noise Removal

1. **Load the “confondre.sig” signal in MATLAB: Sampling frequency = 1kHz**

>> pkg load signal

>> x0 =load('confondre.sig');

>> >> figure (); plot (x0); title ("x");



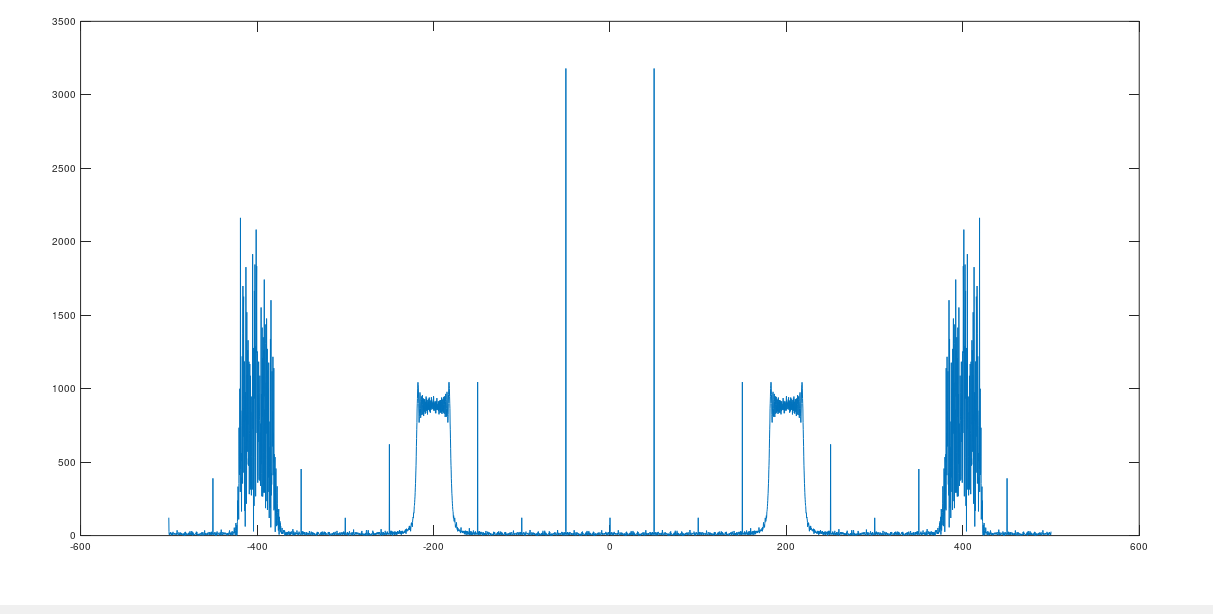
1. **To identify how many and what kind of signals are mixed together we plot the FFT of our signal.**

>>N0=5000

>>y0 = abs((fft(x0,N0)))

>>t0 = linspace(-500,500,N0)

>>plot(t0,fftshift(y0))



From what we obtained above, we observe that 'confondre.sig' is mixed with two types high noises frequencies.

We note that those noises are dissimulating the Original signal which is a square wave. In the following we use FIR and IIR filter to those impurities on the signal.

1. **We use filters to separate each of the signals:**

• First, we use multiple FIR filters: with the command ***fir1*** and ***freqz***

>>fs=1000

>>N=200

>>fk=100

>>Wn= fk/(fs/2)

We choose arbitrarily the cutoff frequency to be fk =100 in concordance with the aquation: fk <(fs/2)

Low pass filter:

>>b = fir1(200,Wn)

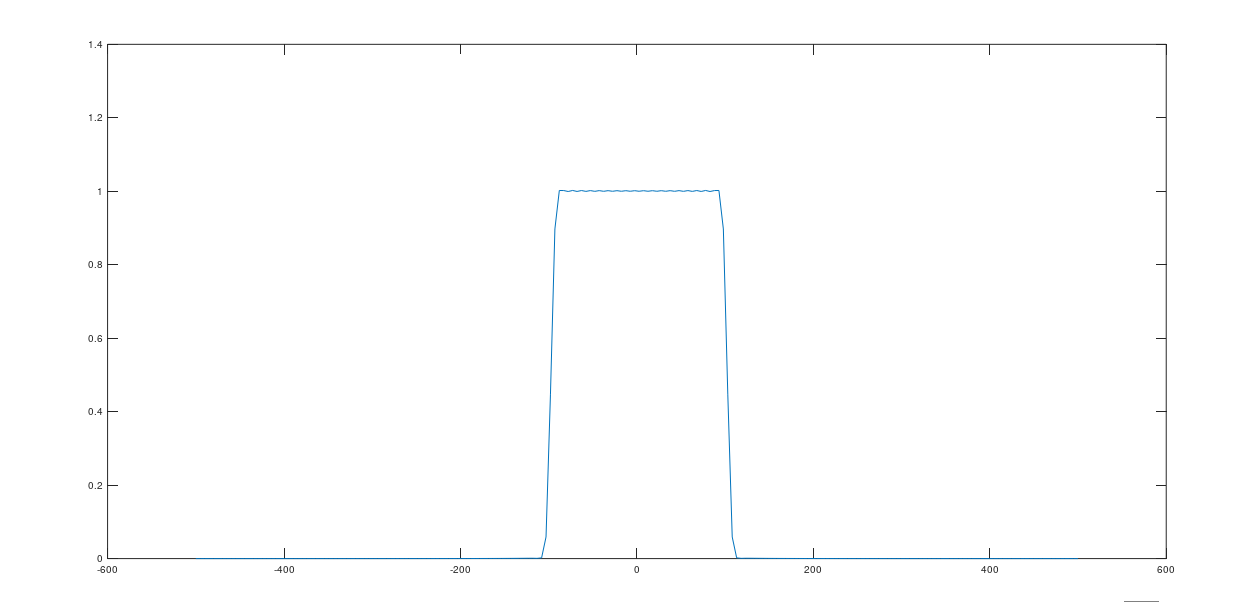
>>[A,~] = freqz(b, 1, 200, "whole");

>>f = linspace(-500,500, 200)

>>figure;

>>plot(f, abs(fftshift(A)));

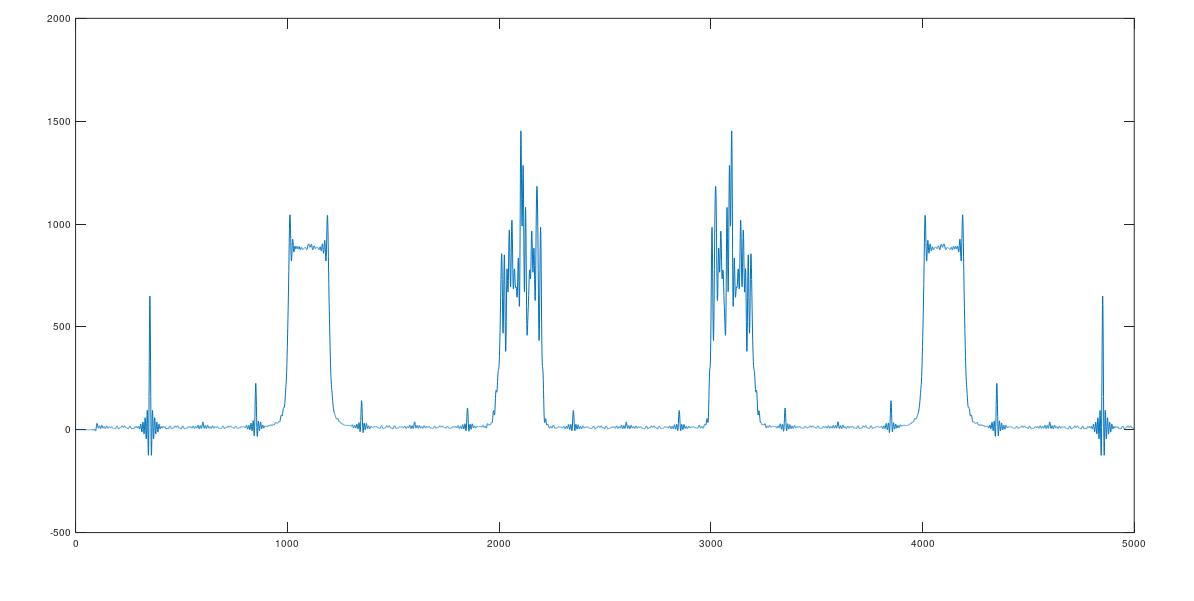
We visualize the filter response of our design while 200 frequency points .



>>y=filter(b,1,y0)

>>plot(y)

We obtained an attenuation of low frequencies after filtering the original signal.

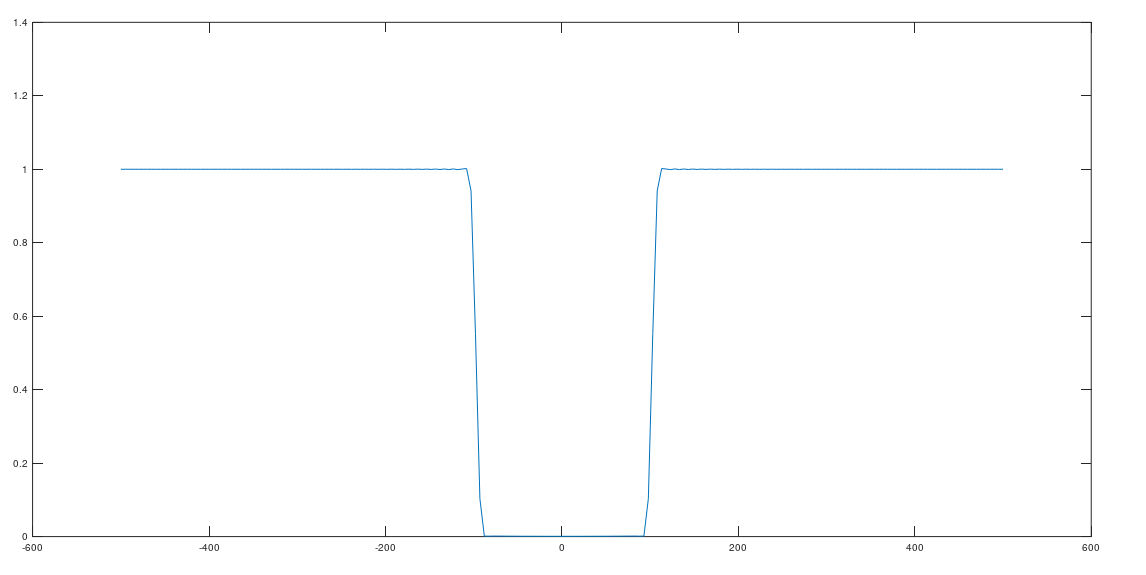


High pass filter:

b2 = fir1(200,Wn,'high')

>> [A2,~] = freqz(b2, 1, 200, "whole");

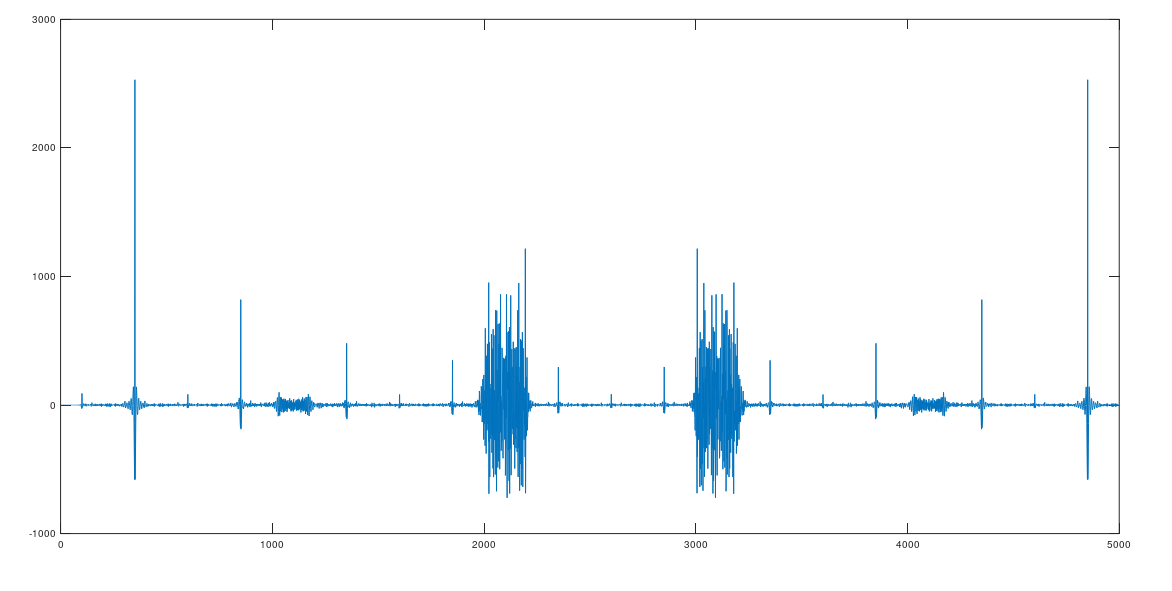
>> plot(f, abs(fftshift(A2)));

Then we designed high pas filter and plot the response similarly to the previous step;

Applied high pass to initial signal:

>> y2=filter(b2,1,y0)

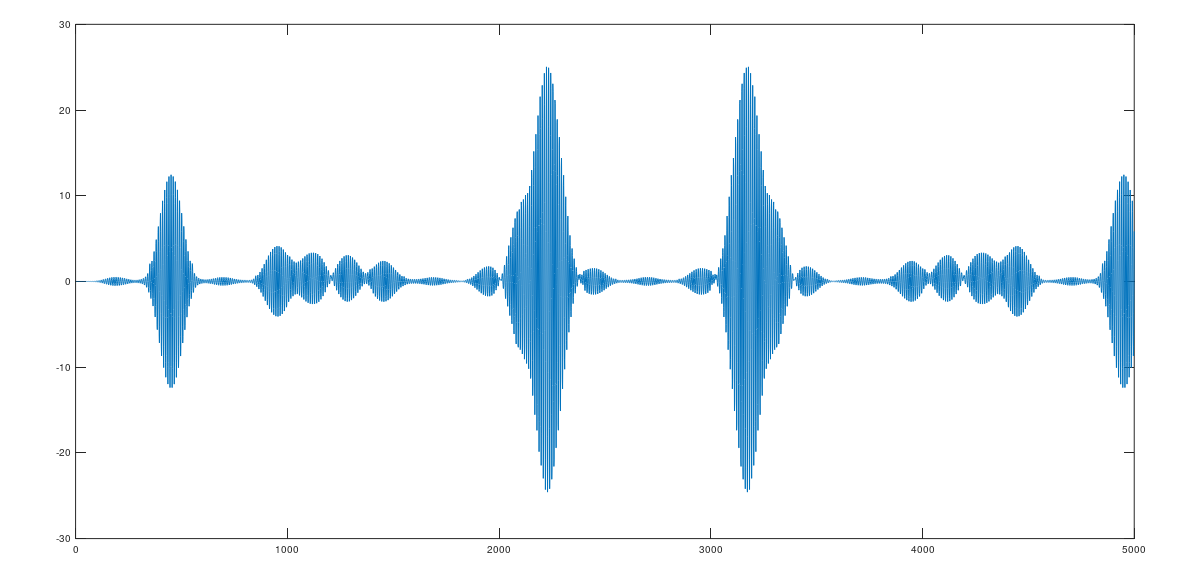
>> plot(y2)



Then we Applied both low pass and then high pass to our signal:

>>s=filter(b2,1,y)

>> plot(s)



• After, use multiple IIR filters: Where we use commands ***butter*** or ***ellip;***

***Firstly we used the butter command***

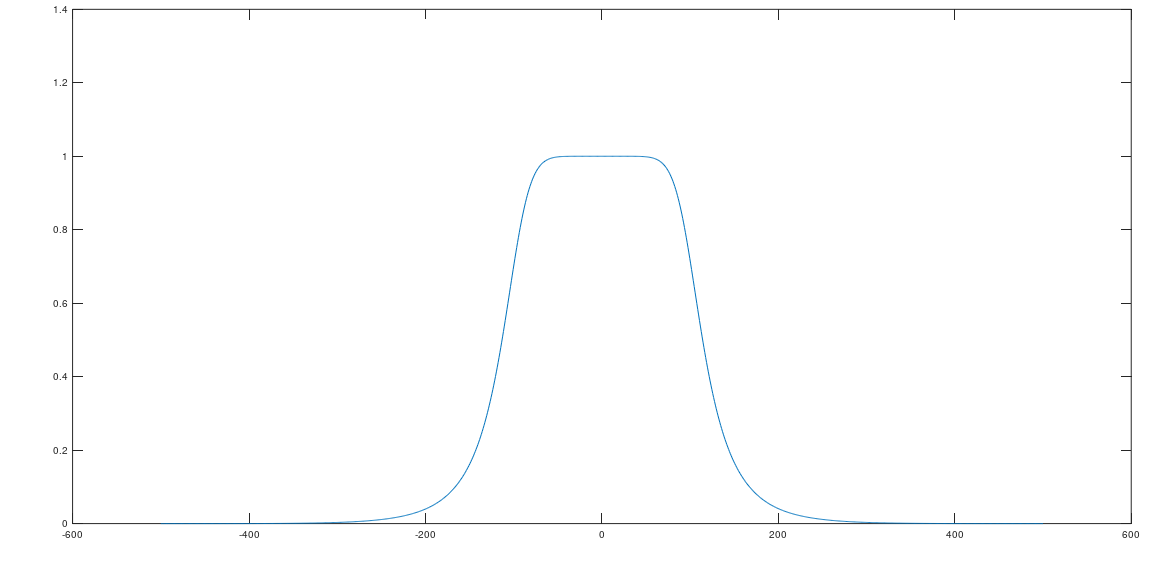
>> [C,D]= butter(4,100/(1000/2))

>> fqc = linspace(-500, 500, 1000);

>> [H,~] = freqz(C, D, 1000, "whole");

>> plot(fqc,fftshift(abs(H)));

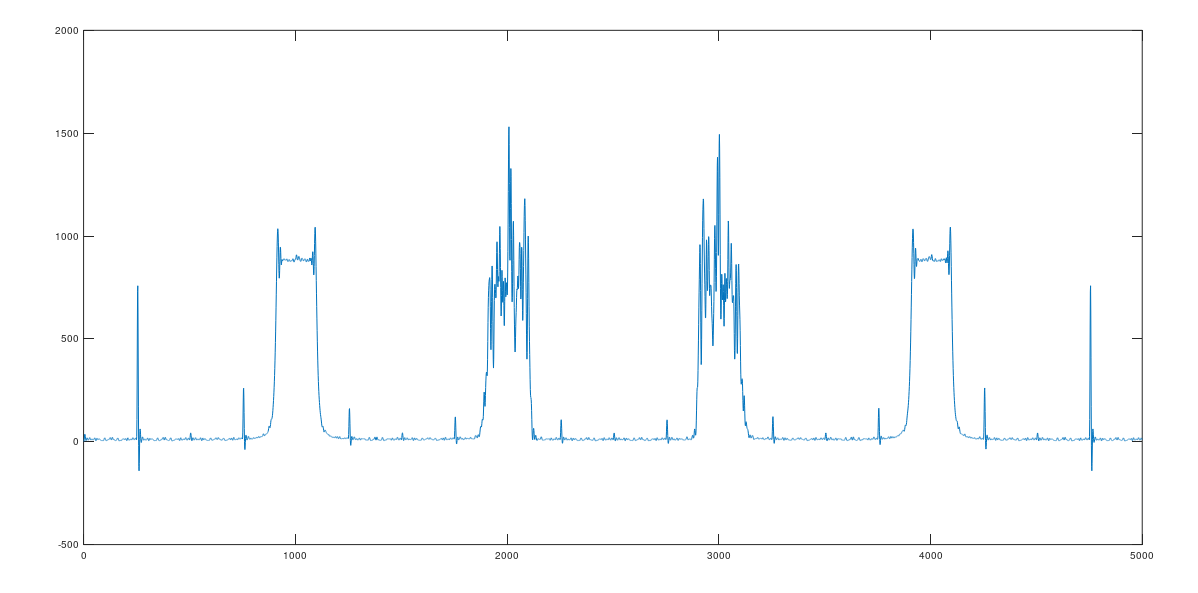
We generate the parameters of our butter filter with the same formula for ***Wn*** and visualize the filter response of our design with 1000 frequency points



>>z=filter(C,D,y0)

>>plot(z)

Here too we observe small an attenuation of low frequencies after filtering the original signal.



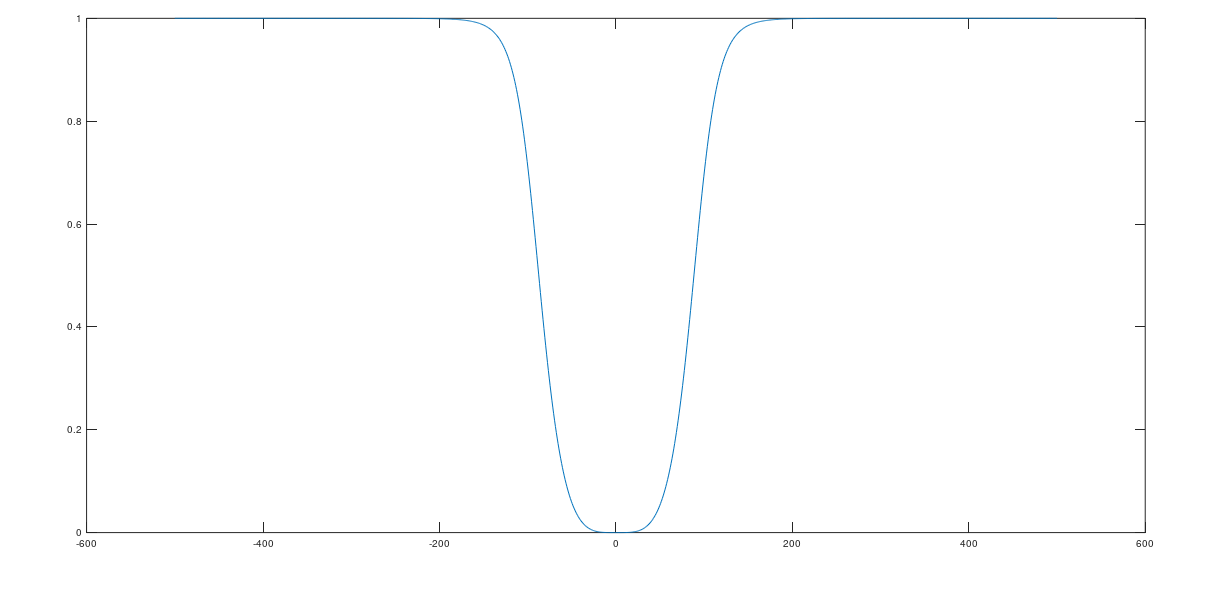
High pass butter

>> [C2,D2]= butter(4,100/(1000/2),"high")

>> [H2,~] = freqz(C2, D2, 1000, "whole")

>> plot(fqc,fftshift(abs(H2)));

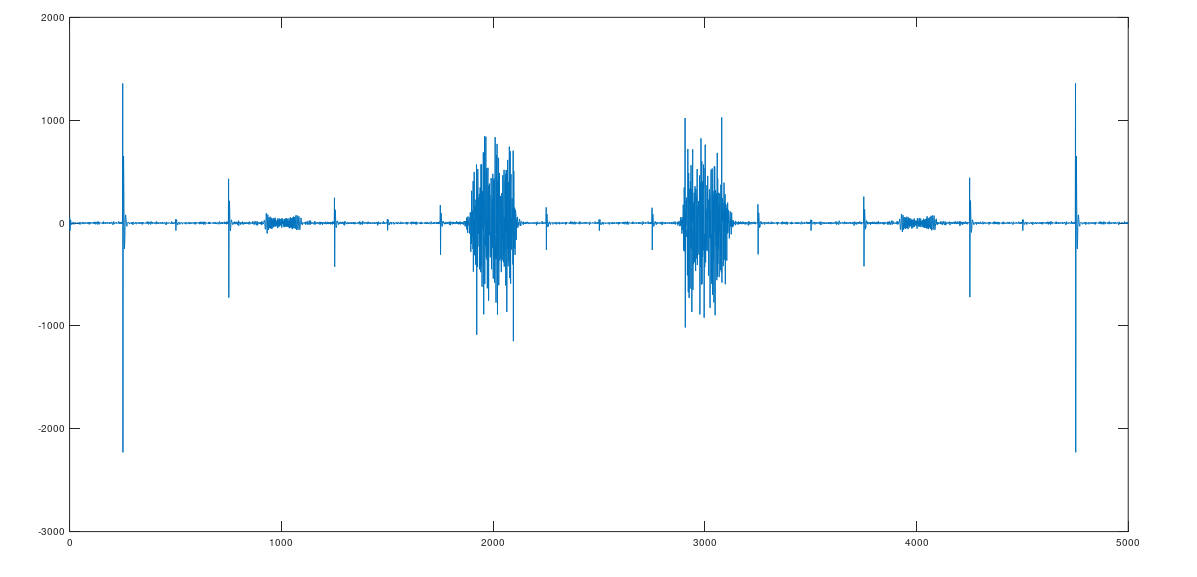
High pass butter filter response plot :



Applied high pass butter filter on the initial signal:

>>z2=filter(C2,D2,y0)

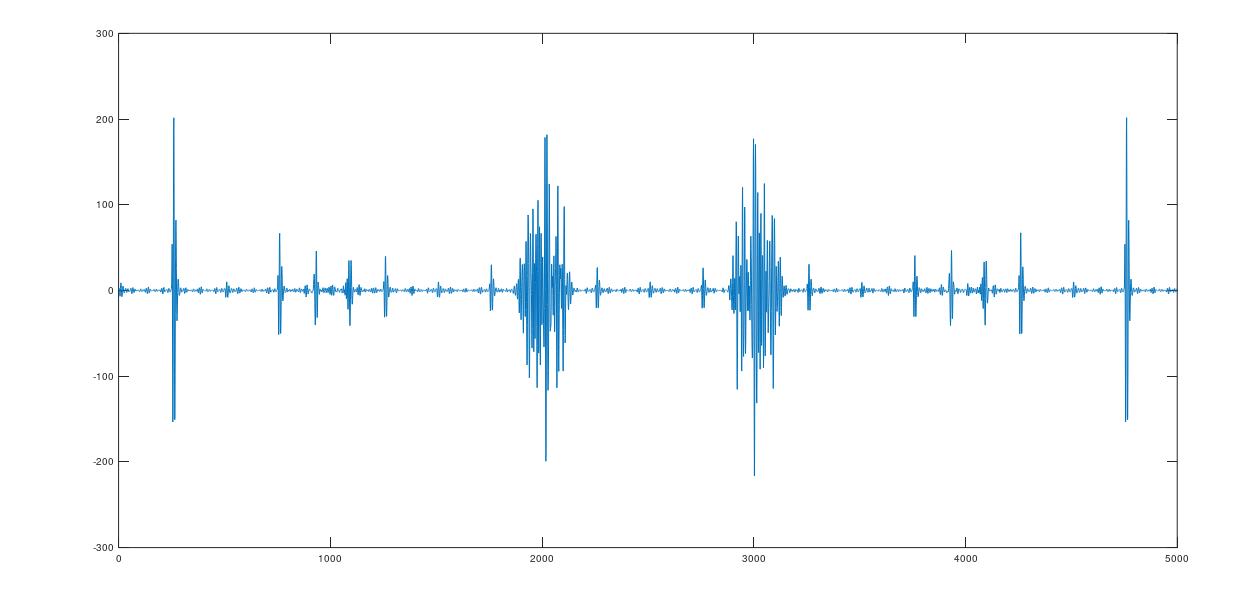
>>plot(z2)



Applied both filters to initial signal:

>>z3=filter(C2,D2,z)

>>plot(z3)



1. For each case, we show all the signals and the filters response in a same and unique plot by using the command : subplot

>> subplot (431)

>> plot(x0);title("raw signal");

>> plot(y0);title("raw signal sampled");

>> subplot(432);

>> plot(y0);title("raw signal sampled");

>> subplot(433);

>> plot(f, abs(fftshift(A))); title("FIR low pass filter");

>> subplot(434);

>> plot(y);title("low pass filter applied ");

>> subplot(435);

>> plot(f, abs(fftshift(A2))); title ("high pass filter");

>> subplot(436)

>> plot(s);title("final signal with low pass and high pass filters applied");

>> subplot(437);

>> plot(fqc,fftshift(abs(H)));title("IIR low pass filter");

>> subplot(438);

>> plot(z);title("IIR low pass applied to initial signal");

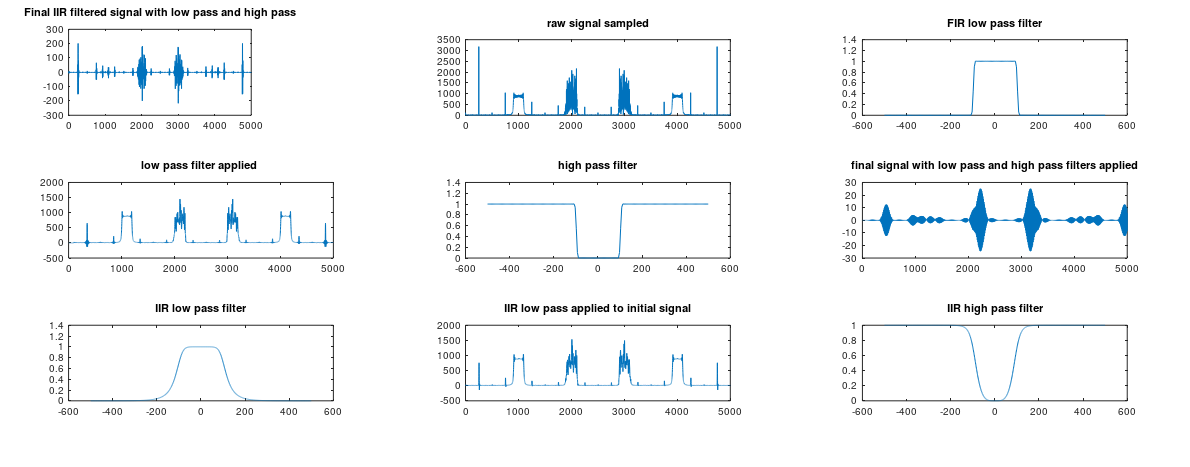
>> subplot(439);

>> plot(fqc,fftshift(abs(H2)));

>> title("IIR high pass filter");

>> subplot(441);

>> plot(z3);title("Final IIR filtered signal with low pass and high pass");



Exercise 3:

a. We create a signal with two frequency components: 1kHz and 8kHz with Sampling frequency of 48kHz;

>> fs=48000

>> f1=1000

>>f2=8000

>> N=2^12;

>>f1 = 1000;  
>>t1 = 1/f1;  
>>n1 = fs\*t1

>>t= linspace(0,t1,n1)

**We generated the frequencies in two different *sin* signal that we sum after .**

>>x=sin(2\*pi\*f\*t)

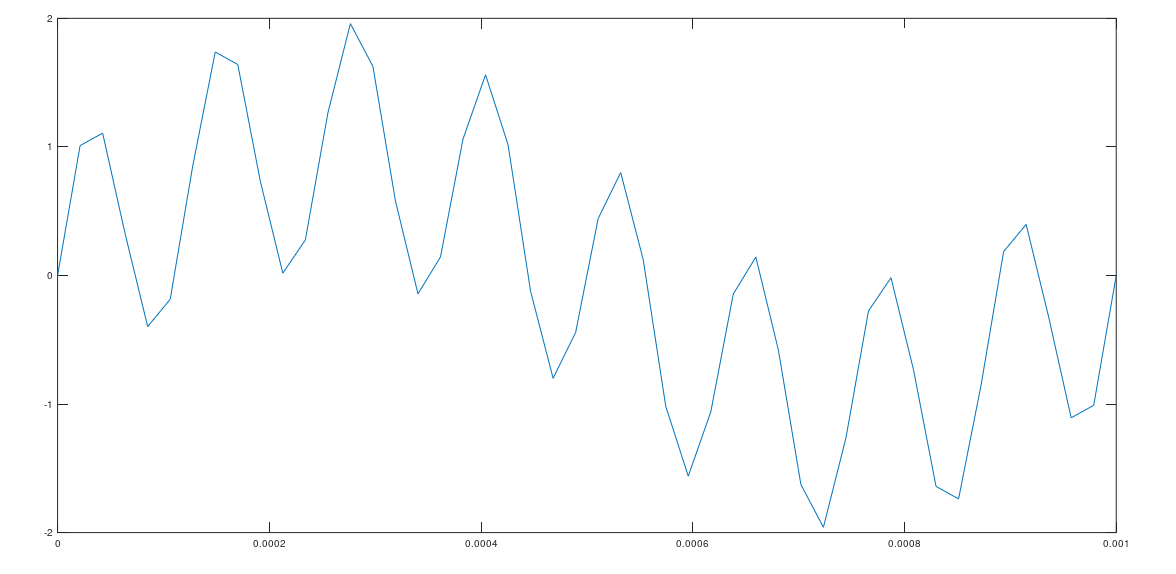
>>x = sin(2\*pi \* t \*f1)

>>y= sin(2\*pi \* t \*f2)

>>z= x + y

>>plot(t,z)

The final summation signal :



b. The design of IIR digital filter to remove the high frequency component: (using butter filter )

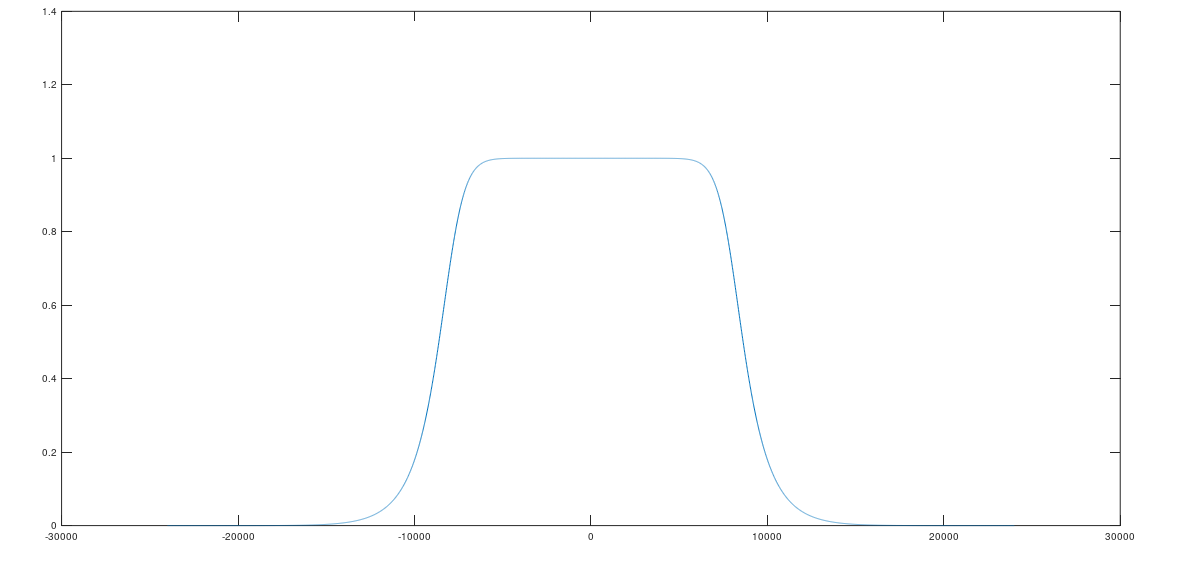
>> [A,B]= butter(5,8000/(fs/2))

>> [H2,~] = freqz(A, B, N, "whole")

>>fqc = linspace(-fs/2, fs/2, N);

>>figure;

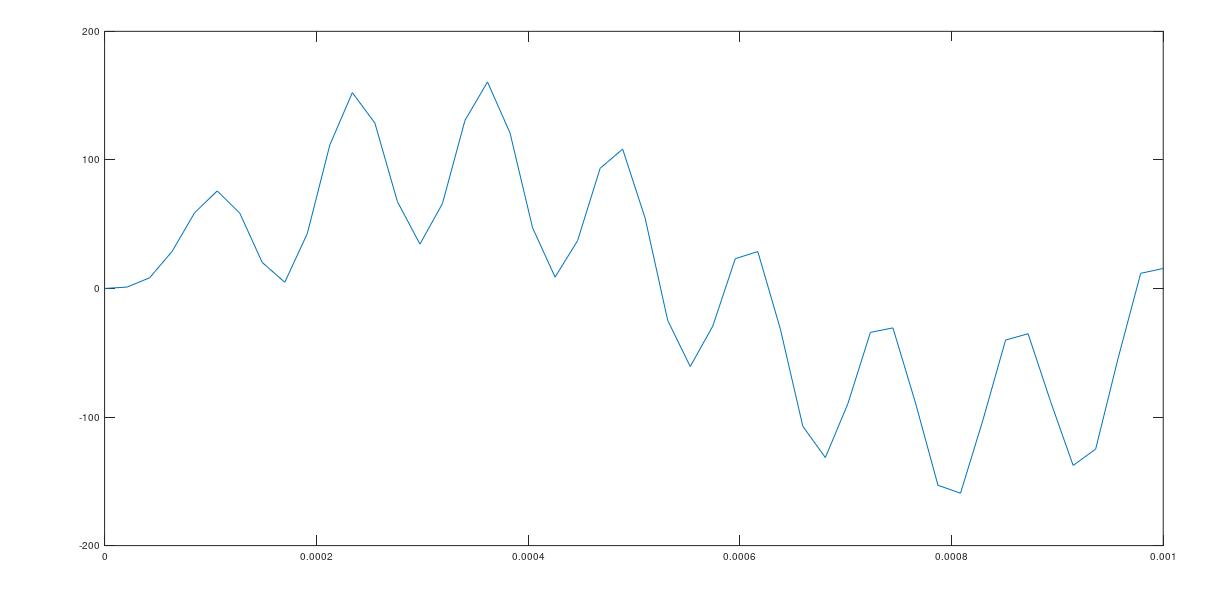
>>plot(fqc,fftshift(abs(H2)));



**c. We filtered the signal in double precision; we use the command *filter;***

F=filter(A,B,z)

plot(t,F)



**d. We Change the original signal and the filter coefficients to single precision;**

singleX = single(x);

singleB = single(B);

singleA = single(A);

**e. Re-filter the original signal:**

• Use functions myfilter1 and myfilter2;

F1 = myfilter1(singleB,singleA,singleX);

F2 = myfilter2(singleB,singleA,F1);

**f. Calculation of error for the new outputs**:

Root\_MSE = sqrt(mean((F2-F).^2));

= 0.70872