Q4 (explanation) :

Question 1.4: Use Matlab (or any programming language) for generating a one second discrete-time signal with three frequencies; 500Hz, 1.5KHz and 3.5KHz and sampling frequency Fs=8KHz. Display your signal using stem or plot functions.
Design and implement a moving average DSP system with a configurable window size. Tune filter window size so the cutoff frequency is equivalent to 2KH. Pass your signal (generated above) through this filter to get output signal. Display output signal and compare it with your input signal? Explain differences?

Moving average technique is one of method to filter the input signal and it is work as low pass filter .

What is the moving average formula ?

$$y(n) = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} x[n-k]$$

This is the formula of moving average .

Example :

let m1 = 0 , m2 = 2 what is Y(5) .

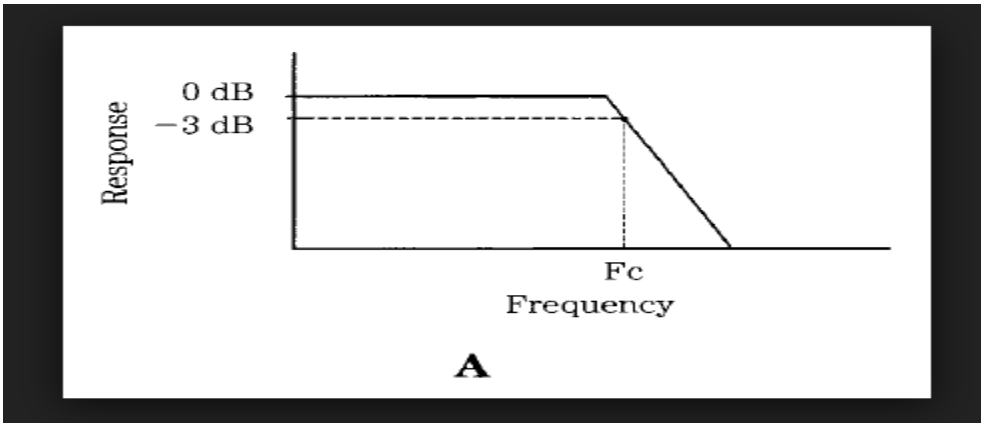
Solution :

Y(5) = (1/3)(x(5) + x(4)+x(3))

Which is the average of the last 3 inputs , that's means that the output signal will prevent the deviation the input signal which actually how the low pass filter works .

Now we know that it is low pass filter , so what it's cutoff frequency ?

The cutoff frequency is approximately means up to which frequency the filter pass the signal , and calculated as shown :



How to get the previous graph ?

The command : freqz(h,1) is give us this graph where h is the impulse response .

Example : freqz([1 1 1],1)

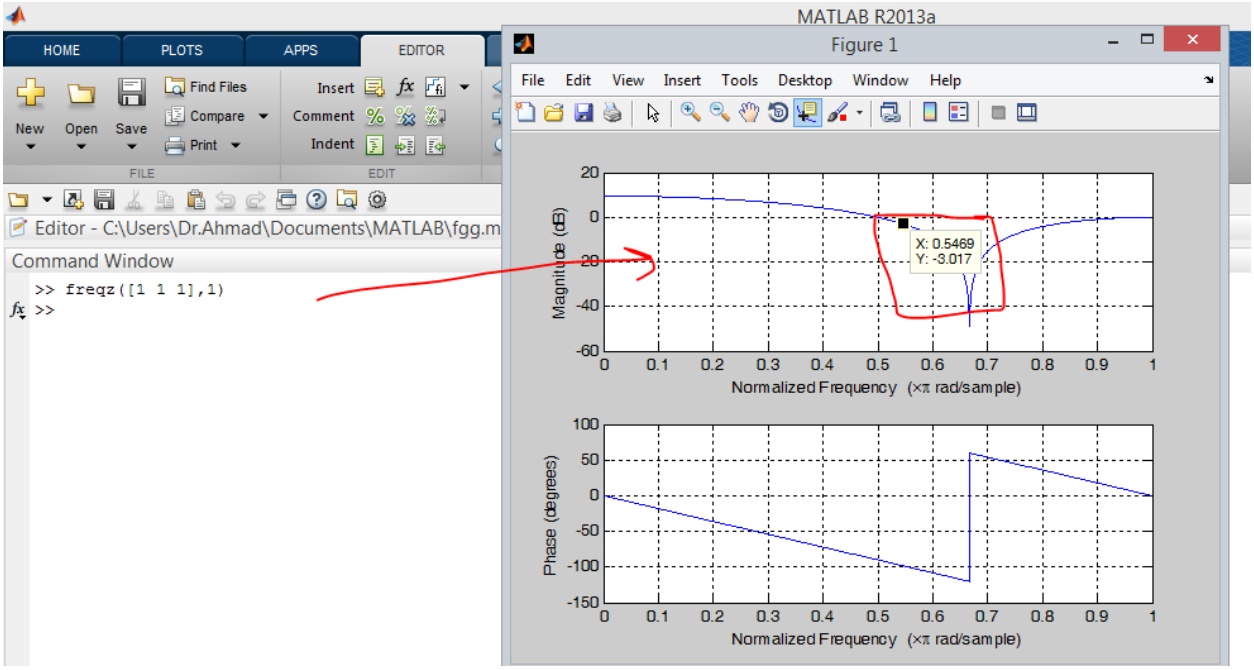
How to make the cutoff frequency = 2 k approximately ?

W = 2*pi*f/fs fs : sampling frequency .

So W = 2*pi*2k/8k = pi/2 .

This means that we want H such that the X-axis of freqz graph = 0.5 approximately when the Y axis = -3 ;

The correct one is when h = [1 1 1]



This will provide me filter with cutoff frequency = 2.2k approximately = 2k .

How can I implement it ?

Simple now we know that M1 = 0 and M2 = 2 (any window with length = 3).

Apply the moving average algorithm with this paramenters .

How I generate input with fs = 8k , with 500hz , 1.5hz , 3.5hz?

n = 0:1/(8000):1-1/(8000);

x = 2*cos(2*pi*500*n) + 7*cos(2*pi*1500*n) - 4*cos(2*pi*3500*n);

This will provide me 8000 sample with these 3 frequencies simple you can use the indices

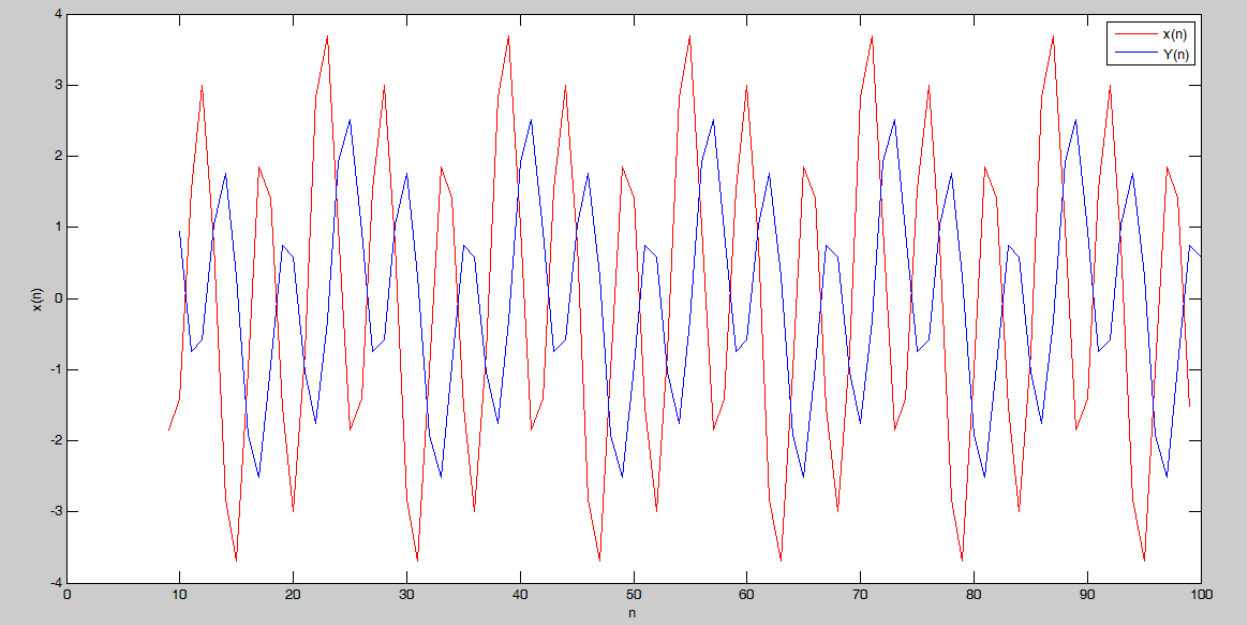
N1 = 0:7999 ; to denote the indices .

Shortly , now I have signal [x , N1]

Final step ?

Just implement moving average like Q1,E , and pass this signal to it , you can note that the output signal will reject the high frequencies .

Like this :



Red : input .

Blue : output .