Digital Filter Designing
(FIR & IIR)

Falak Sher	(15-EE-96)
Muhammad Idrees	(15-EE-88)
Hamza Niaz	(15-EE-136)

July 2018

Department of Electrical Engineering

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

TAXILA– PAKISTAN

Abstract

In digital control system, interference, which is mixed in the input signal, has a great influence on the performance of the system. Therefore, processing of input signal has to be done to get useful signal. Finite impulse response (FIR) and Infinite impulse response (IIR) filter plays an important role in the processing of digital signal. Designing the FIR and IIR filter by Matlab can simplify the complicated computation in simulation and improve the performance. This project is undertaken with the objective of demonstrating the design process of a digital filters that is FIR and IIR. In this project, we have to design both types of filters to remove noise from a given corrupted signal by using frequency analysis technique. Firstly, we computed the DFT of the given corrupted signal such that to avoid aliasing. Frequency domain analysis was carried out to identify the frequency or frequencies pertaining to the noise in order to filter them out. After identifying the frequencies, we designed a **bandstop filter** to filter the noise as per our best knowledge. The experimental results are given in this paper showing that the FIR filters are more effective because of its property of linear phase than that of IIR filters.

Table of Contents

Abstract 2
1. Objective 4
2. Introduction to Digital Filters 4
2.1 FIR Filter 4
2.2 IIR Filter 5
2.3 Comparison of FIR and IIR 5
2.4 Advantages & Disadvantages of FIR & IIR 5
3. Description 5
4. MATLAB Plots 6
5. MATLAB Code 9
6.Conclusion .... **Error! Bookmark not defined.**

1. Objective

Our objective in this project is to design a FIR and IIR digital filter to remove the noise from the given noisy or corrupted signal.

2. Introduction to Digital Filters

The digital filter is a discrete system, and it can do a series of mathematic processing to the input signal, and therefore obtain the desired information from the input signal. The transfer function for a linear, time-invariant, digital filter is usually expressed as;

$$H(z) = \frac{\sum_{j=0}^{M} b_j z^{-j}}{1+\sum_{i=1}^{N} a_i z^{-i}} \ ,$$

Where a_i and b_i are coefficients of the filter in Z-transform.

There are many kinds of digital filters, and also many different ways to classify them. According their function, the FIR filters can be classified into four categories, which are lowpass filter, highpass filter, bandpass filter, and bandstop filter.

According to the impulse response, there are usually two types of digital filters, which are finite impulse response (FIR) filters and infinite impulse response (IIR) filters. According to the formula above, if a_i is always zero, then it is a FIR filter, otherwise, if there is at least one none-zero a_i, then it is an IIR filter. Usually we need three basic arithmetic units to design a digital filter, which are the adder, the delay, and the multiplier.

2.1 FIR Filter

The finite impulse response (FIR) filter is one of the most basic elements in a digital signal processing system, and it can guarantee a strict linear phase frequency characteristic with any kind of amplitude frequency characteristic. Besides, the unit impulse response is finite; therefore, FIR filters are stable system. The FIR filter has a broad application in many fields, such as telecommunication, image processing, and so on. The system function of FIR filter is

$$H(z) = \sum_{n=0}^{L-1} h[n]\, z^{-n},$$

Where L is the length of the filter, and h[n] is the impulse response

2.2 IIR Filter

The infinite impulse response (IIR) filter is recursive structure, and it has a feedback loop. The precision of amplitude frequency characteristic is very high, and IIR filters are not linear phase

2.3 Comparison of FIR and IIR

- Under the same conditions as in the technical indicators, output of the IIR filter has feedback to input, so it can meet the requirements better than FIR. The storage units are less than those of IIR, the number of calculations is also less, and it's more economical.
- The phase of FIR filter is strictly linear, while the IIR filter is not. The better the selectivity of IIR filter is, the more serious the nonlinearity of the phase will be.
- The FIR filter is non-recursive structure, finite precision arithmetic error is very small. While IIR filter is recursive structure, and parasitic oscillation may occur in the operation of IIR filter.
- Fast Fourier Transformation can be used in FIR filter, while IIR cannot.
- The IIR filter can make use of the formulas, data and tables of the analog filter, and only a small amount of calculation. While FIR filter design may always make use of the computer to calculate, and the order of FIR filter could be large to meet the design specifications.

2.4 Advantages & Disadvantages of FIR & IIR

The advantage of IIR filters over FIR filters is that IIR filters usually require fewer coefficients to execute similar filtering operations, that IIR filters work faster, and require less memory space.

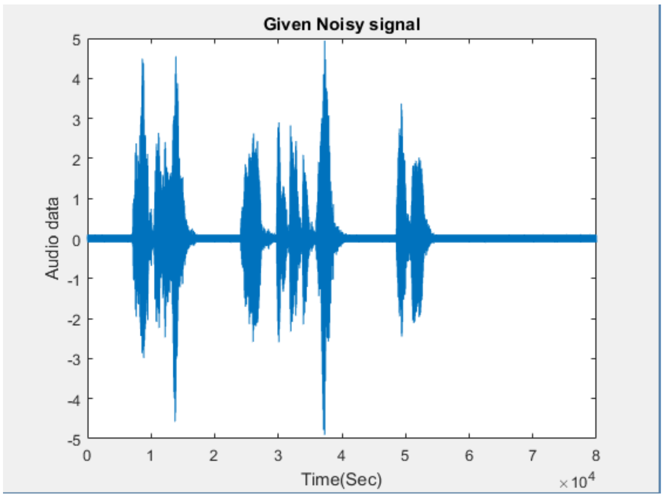
The disadvantage of IIR filters is the nonlinear phase response. IIR filters are well suited for applications that require no phase information, for example, for monitoring the signal amplitudes. FIR filters are better suited for applications that require a linear phase response.

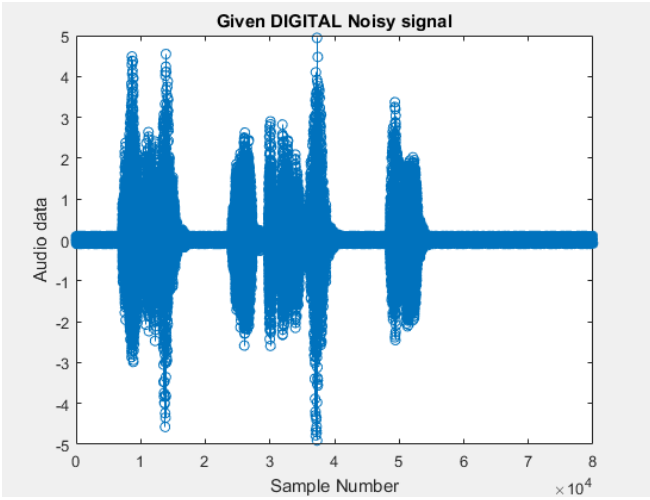
3. Description

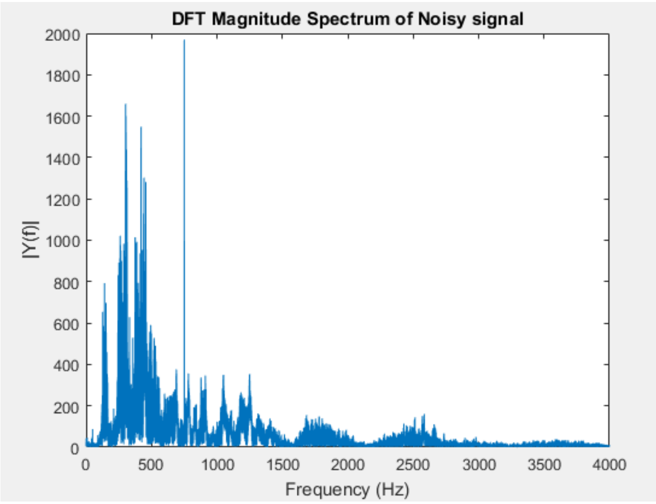
As the given noisy signal has a total number of 80000 samples we find it by using the command in matlab.After finding the length of noisy signal, we have to compute the DFT of the signal using FFT algorithm which is an effective way of computing DFT.To avoid aliasing we must have to keep N>=L (Length of signal).Therefore, we selected N=131072 to compute DFT of the signal. If this does not happen so, aliasing will occur. A NYQUIST criteria must be met for ADC to avoid aliasing which is defined by $f_{n}=2f_{max}$ where f_{max} maximum frequency in the given signal. .We took the last 10000~15000 samples and took the FFT of these samples. After doing this we were able to find the frequencies of noise which was at 420Hz and 750Hz.To remove these frequencies, we decided to design BANDSTOP filter by our best knowledge. To design this type filter we used the 'fdatool'command of MATLAB and design the bandstop filters at two frequencies (420Hz & 750Hz) which were then cascaded and used the 'filter' command to remove the noise. Order of the filter was automatically generated by fdatool command for better results.

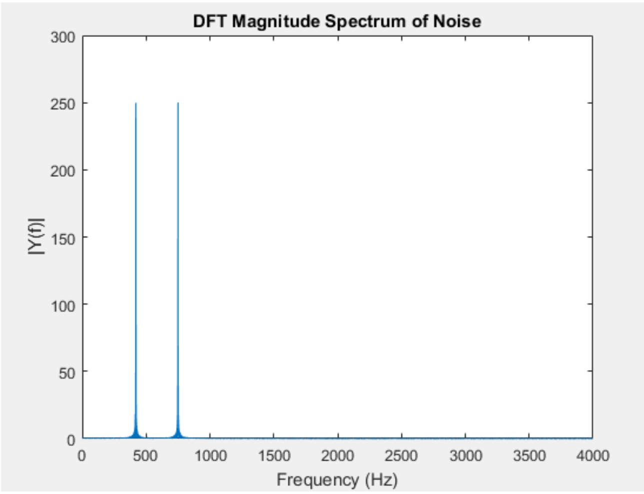
4. MATLAB Plots

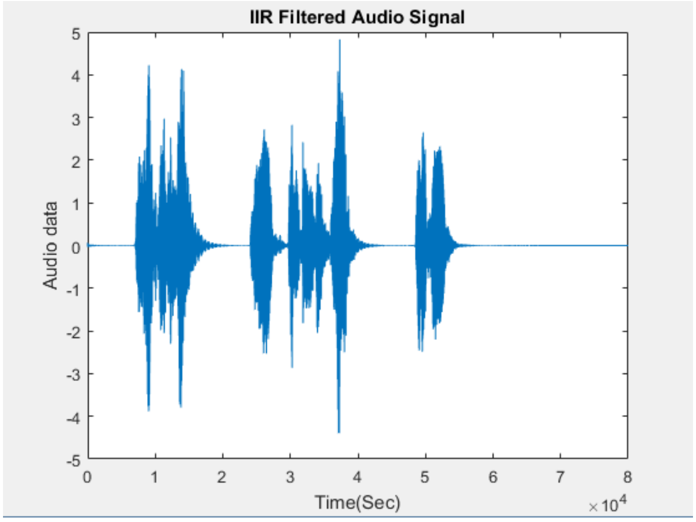
Following are the plots.

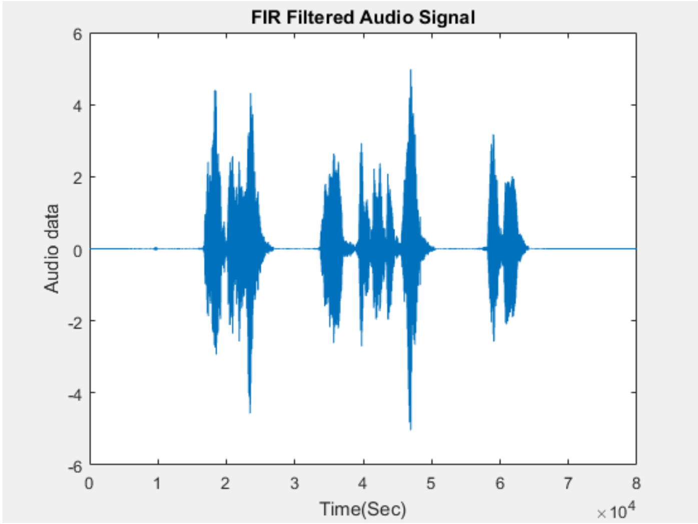


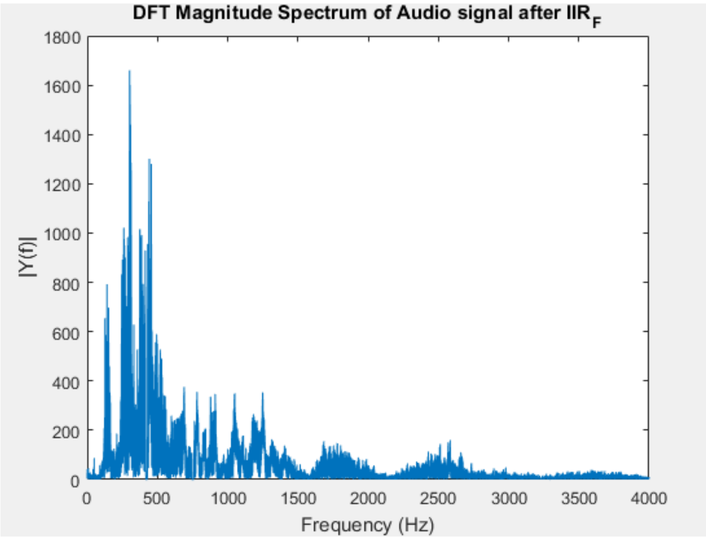


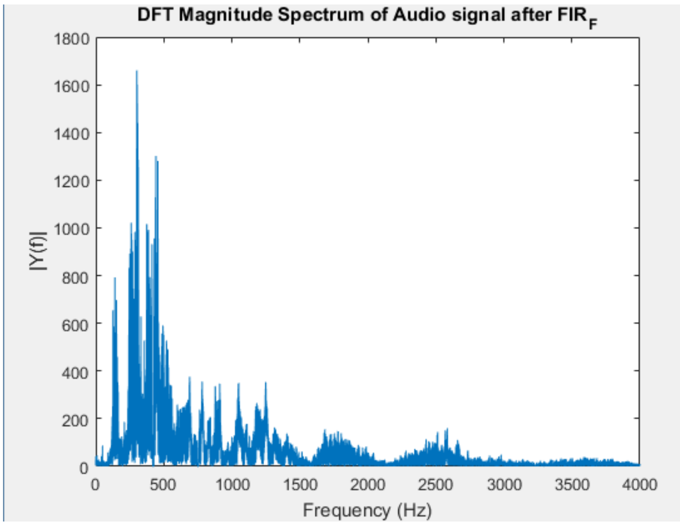




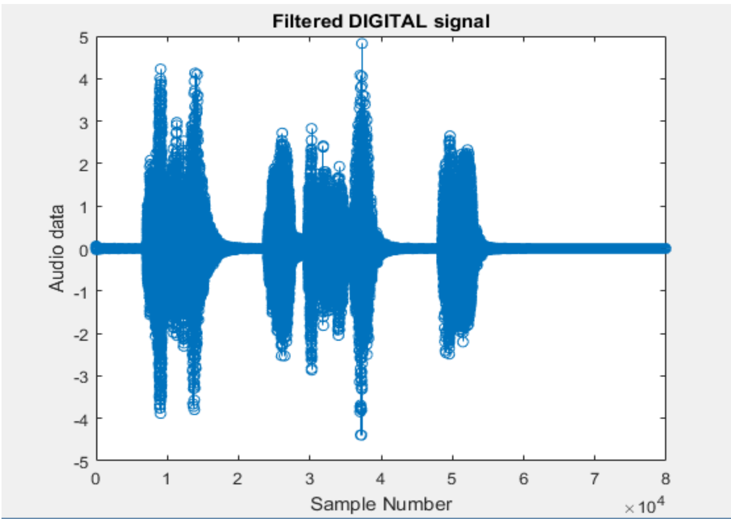

IIR Filtered Audio Signal
Audio data
Time(Sec)
$\times 10^4$


FIR Filtered Audio Signal
Audio data
Time(Sec)
$\times 10^4$

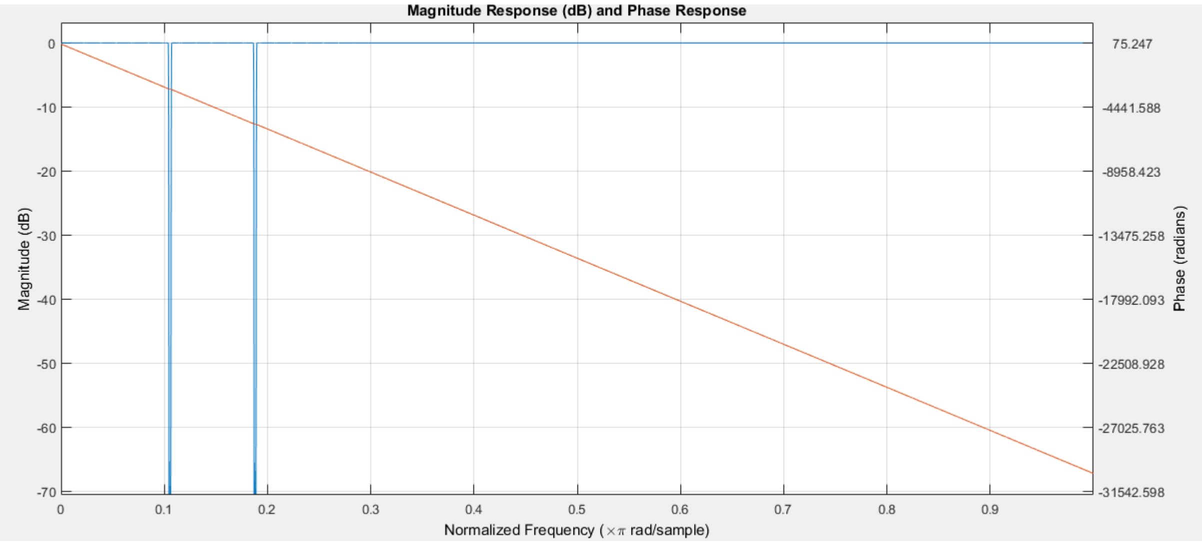

DFT Magnitude Spectrum of Audio signal after IIR_F
|Y(f)|
Frequency (Hz)


DFT Magnitude Spectrum of Audio signal after FIR_F
|Y(f)|
Frequency (Hz)

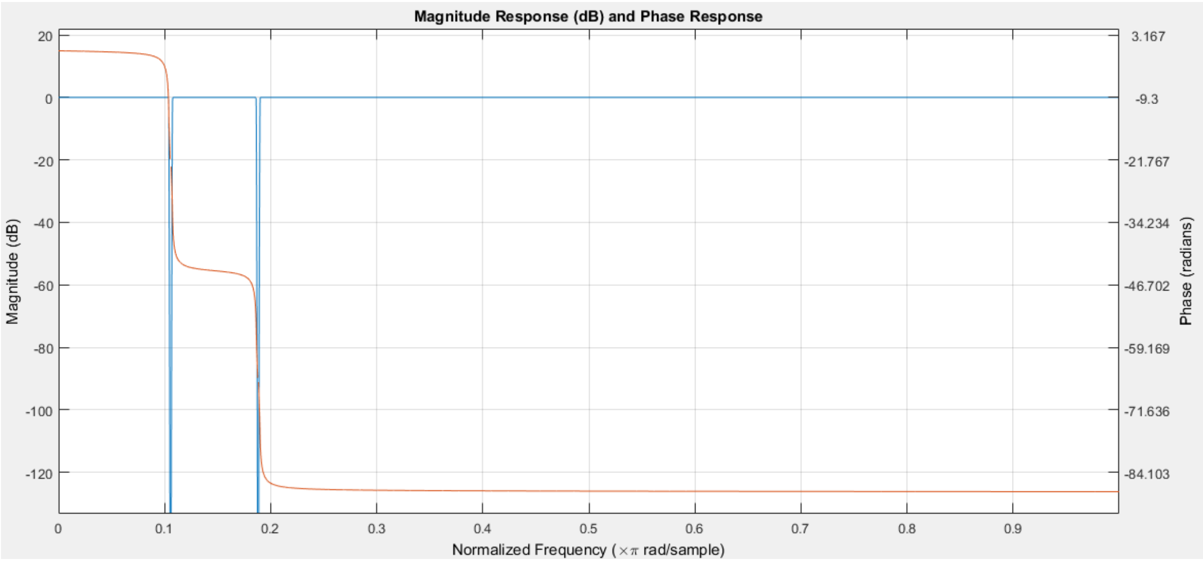
.


Filtered DIGITAL signal
Audio data
Sample Number
$\times 10^4$

FIR filter's magnitude and phase response (Linear)



IIR filter's magnitude and phase response (Non-Linear)



5. MATLAB Code

```
%loading the file noisy_signal
close all
load noisy_signal.mat;
p=audioplayer(noisy_signal,8000); %sampling frequency fs= 8000Hz by audiorecorder command
play(p);
audiowrite('Noisy-Signal.wav',noisy_signal,8000);
figure;
plot(noisy_signal);
title('Given Noisy signal');
xlabel('Time(Sec)');
ylabel('Audio data');

%DT noisy signal
figure;
stem(noisy_signal);
title('Given DIGITAL Noisy signal');
xlabel('Sample Number');
ylabel('Audio data');

%spectral analysis of the signal
L=length(noisy_signal);
NFFT=2^nextpow2(L);
fft_ns=abs(fft(noisy_signal,NFFT));

%creating frequecny axix to find the freq comp
freq_axis=8000/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
figure;
plot(freq_axis,fft_ns(1:NFFT/2+1));
title('DFT Magnitude Spectrum of Noisy signal');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

%To identify noise frequencies
lastsamples=noisy_signal(70000:80000);
fft_lastsamples=abs(fft(lastsamples,NFFT));
figure;
plot(freq_axis,fft_lastsamples(1:NFFT/2+1));
title('DFT Magnitude Spectrum of Noise');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

%IIR filter designing using fdatool
iir_f_sg=filter(IIR,noisy_signal)
pl=audioplayer(iir_f_sg,8000);
play(pl);
audiowrite('IIR Filtered Audio Signal.wav',iir_f_sg,8000);
```

```
fvtool(IIR)

%FIR filter designing using fdatool
fir_f_sg=filter(FIR,noisy_signal)
player=audioplayer(fir_f_sg,8000);
play(player);
audiowrite('FIR Filtered Audio Signal.wav',fir_f_sg,8000);
fvtool(IIR);

audiowrite('FIR_filtered.wav',ns_f_fir,8000);

figure;
plot(fir_f_sg);
title('FIR Filtered Audio Signal');
xlabel('Time(Sec)');
ylabel('Audio data');

figure;
plot(iir_f_sg);
title('IIR Filtered Audio Signal');
xlabel('Time(Sec)');
ylabel('Audio data');

%DFT Magnitude spectrum of filtered signal FIR
z=abs(fft(fir_f_sg,NFFT));
figure;
plot(freq_axis,z(1:NFFT/2+1));
title('DFT Magnitude Spectrum of Audio signal after FIR_F');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
%DT filtered signal
figure;
stem(iir_f_sg);
title('Filtered DIGITAL signal');
xlabel('Sample Number');
ylabel('Audio data');

%DFT Magnitude spectrum of filtered signal IIR
z1=abs(fft(iir_f_sg,NFFT));
figure;
plot(freq_axis,z1(1:NFFT/2+1));
title('DFT Magnitude Spectrum of Audio signal after IIR_F');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
```

6. Conclusion

The use of digital filters is now paramount to see the results correctly and clearly. Digital filters are easily designed, tested and implemented on a general-purpose computer or workstation. Fast DSP processors can handle complex combinations of filters in parallel or cascade (series), making the hardware requirements relatively simple and compact in comparison with the equivalent analog circuitry. The given signal was present with noise but after the design of filters the noise has been attenuated and the audio signal can be heard clearly.