

Digital Signal Processing with FIR Filter Design and Fast Fourier Transform

Xuerong Wang

*Faculty of Science and Engineering, The University of Manchester, Manchester, UK
Xuerong.Wang.28@outlook.com*

Abstract. Digital Signal Processing (DSP) is a very significant research area in communication. Finite impulse response (FIR) filters are one of the most efficient and commonly used practical filters for digital signals, which can remain stable all the time and tend to implement different frequency responses ideally with greater flexibility. In this research, the windowing method is used to design the digital filters due to its simplicity, which scales each sample in impulse response. By using this method, the discrete-time system produces the continuous spectra. In addition, Discrete Fourier Transform (DFT) is used to provide discrete spectra instead. However, the DFT can be efficiently computed with the Fast Fourier Transform (FFT), which is a very widely used and practical algorithm. Two samples are analyzed by using the FIR digital filter design with the window method and FFT in this paper, which are the audio signal and satellite transmission signal respectively. The audio signal is from a 5-second recorded voice, and the satellite transmission signal. The property and effect of the FIR filter can then be finally found by using MATLAB to process the signal in this research.

Keywords: Digital Signal Processing, FIR filter design, Window Method, DFT&FFT, MATLAB

1. Introduction

Digital filters play a significant role in digital signal processing, which are useful tools in Linear Shift Invariant (LSI) digital system design to modify the characteristics of the input [1]. Although most practical physical phenomena can be represented as continuous-time analogue signals, most modern signal processing is done in the digital domain. The analogue and digital signals can be converted from one to the other by using an analogue-to-digital converter (ADC). If we want to have this conversion, sampling is needed to record the amplitude of the analogue signal at specific times, for which the sampling frequency is fixed when recording each sample and this is needed for analyzing the frequency response for each digital system. The frequency response shows the output in response to a sinusoid input of unit magnitude and some specified frequency. After the sampling and conversion, a digital signal is produced for further processing and analysis. If people want to change the digitalized signals back to analogue, the Digital-to Analogue Converter (DAC) needs to be used to achieve this. With the development of software and hardware, there are many applications

of digital signal processing that are mainly implemented and analyzed for their algorithms either in software like MATLAB or a processor [2].

The digital filters can be divided into Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) according to their different algorithms [3]. FIR filters are very useful to analyze the applications because we want the system to be linear-phase and easily controlled, which the IIR filter cannot achieve [4]. The window method is one of the most common methods used in FIR filter design, which can be theoretically analyzed by using MATLAB. Some specifications are given to use in designing FIR filter by using the window method, which is the requirement for designing the filter [1]. In addition, the DFT has become the core in digital signal processing, which can be computed more efficiently with the Fast Fourier Transform (FFT). Many signal processing applications use FFT to perform filtering tasks to analyze the processing data.

In this research study, two examples are used to analyze the effect and property of the FIR filter by designing the FIR digital filter with the window method. In the first sample, a 5-second audio signal is studied, which chooses the most suitable window type and calculates the relevant parameter to design a low-pass filter by using the given specifications. In the second sample, the satellite transmission signal is researched for its FFT property with the given FIR digital filter window method. The two examples are representative applications of study in the modern world, which can investigate the digital signal processing more effectively.

2. Methodology

The two examples of signal are going to use FIR filter and FFT to process and then find the effect of them. The FIR Digital filter plays the most significant role in this research, and the FFT is usable in the second sample.

2.1. FIR digital filter

The FIR digital filter is a very useful tool for signal processing to get the desired output signal with the given specifications [5]. Its non-recursive property is due to the utilization of the current and prior inputs, which can guarantee the system to be more stable [6]. There is no feedback in the FIR filter because the output is independent of the prior values. Also, they have the exact linear phase, which can be designed more easily [2]. Nowadays, the basic structure of the FIR digital filter is shown in Figure 1 below.

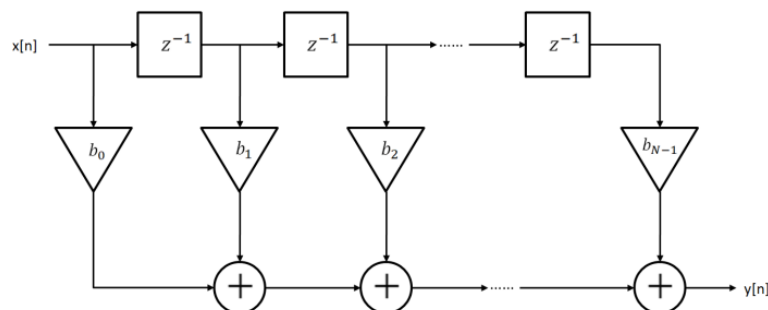


Figure 1. Basic FIR digital filter architecture [2]

There are exist implementation methods with high computational efficiency for the FIR digital filter. The output signal produced by the FIR digital filter is the convolution of the impulse response of the system and the input signal, which is represented as:

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n-k] = \sum_{k=0}^{N-1} b_k x[n-k] \quad (1)$$

where $x[n]$ is the input signal, $h[n]$ is the impulse response of the filter, k is the order of the filter, $y[n]$ is the output signal [7].

The transfer function is defined as the output of filter divided by the input of filter, which is obtained from the z-transform of its impulse response:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} h(n)z^{-n} \quad (2)$$

When designing the FIR digital filter, there are various excellent method that can be used, including the windowing-based method and optimization-based method, which are mainly used nowadays [8]. Filter specifications are needed before designing the filter, including pass band ripple, stop band ripple, pass band edge frequency, stop band edge frequency and filter length [1]. Sampling frequency is also given, which is fixed in designing the filter. The normalized frequency in rad/sample is used instead of the frequency instead in Hz in digital filter design. The formula for calculating the normalized frequency is given as:

$$\omega = \frac{2\pi f}{f_s} \quad (3)$$

where f_s is sampling frequency in Hz.

These four window types have fixed window functions, which are rectangular, Hann, Hamming and Blackman.

They have only one independent parameter, the filter length, which can be varied to determine the fixed main lobe width and transition band functions. When there are two or more independent parameters, such as filter length, and one or more other parameters, the characteristics of the other window can be changed; these types of windows are called adjustable windows [5]. The fixed window functions are the main research direction that is focused on in this study.

Here are the characteristics for the 4 fixed Windows type, where L is the window length (filter order).

Table 1. Properties of 4 common FIR window functions [9]

Window Type	Main Lobe Width	Peak side Lobe Ripple / dB	Transition Band (Ω_{Δ})	Stop band Attenuation (G_s) / dB	Pass Band Ripple(r_p) / dB
Rectangular	$\frac{4\pi}{L}$	-13.3	$\frac{1.84\pi}{L-1}$	20.9	1.57
Hann	$\frac{8\pi}{L}$	-31.5	$\frac{6.22\pi}{L-1}$	43.9	0.11
Hamming	$\frac{8\pi}{L}$	-42.7	$\frac{6.64\pi}{L-1}$	54.5	0.033

Table 1. (continued)

Blackman	$\frac{12\pi}{L}$	-52.1	$\frac{11.13\pi}{L-1}$	75.3	0.0030
----------	-------------------	-------	------------------------	------	--------

When designing a FIR digital filter with these 4 fixed window types, using the Table 1 and the given specifications to achieve the desired target.

2.2. DFT and FFT

As the signal is only processed in the sampled form in digital signal processing [10], the discrete Fourier transform is a necessary and fundamental algorithm and is used in many practical applications to perform Fourier analysis [11]. The discrete data is converted from one domain to another [12]. The DFT can be used to represent signals that are comprised of some frequency components very efficiently [13]. The more efficient computation of the DFT algorithm is called Fast Fourier Transform (FFT), which is easier to perform in MATLAB. The FFT is a form that is decomposed recursively from the DFT into smaller and smaller DFTs [12]. Generally, the DFT is expressed as the sum of the sine and cosine functions.

The Discrete Fourier Transform $X[k]$ of $x[n]$ is defined as following:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jnk \frac{2\pi}{N}} \quad (4)$$

where $x[n]$ is the input sequence with length N , $X[k]$ is the DFT of $x[n]$ with length N .

The transform also can be reversed, which is expressed as Inverse Discrete Fourier Transform (IDFT) [11].

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{jnk \frac{2\pi}{N}} \quad (5)$$

3. Results & discussion

3.1. 5-second audio signal

The first sample data is related to a five-second audio signal, which is recorded in MATLAB and the voice needs to make sure that it is clear. To capture enough detail without using too much memory, the audio should be sampled at 22050 Hz and with 8 bits (1 byte) per sample. After recording, the audio is converted to a format that can be readily used for signal processing, which has 110,250 elements in this vector.

To process this 110250-element audio signal vector called "audioOrig" that has already been created above, an FIR low-pass digital with the windowing method should be designed. By using the window information tables introduced above, the shortest window type and length need to be determined first which will give a pass band ripple below r_p at frequencies less than f_{p1} , and that has a stop band defined by frequencies above f_{stop1} . The sampling rate being used is $f_{s1} = 22.05$ kHz as written above. (At this time, trialing different parameters to meet the specification is not needed; just assume that the tables are accurate.) In this research, use $r_p = 0.135$ dB, $f_{p1} = 3384$ Hz, $f_{stop1} = 3789$ Hz. Then the transition band frequency f_{t1} in Hz can be calculated as $f_{stop1} - f_{p1}$, which is 405 Hz. To calculate the window length more conveniently, the $ft1$ should be sampled as $\Omega\Delta$, which is

calculated as $(2 \times \pi \times f_{c1})/f_{s1} = 0.1154$ rad/sample. As $r_p = 0.135$ dB and the shortest window length needs to be determined and smaller than this value, "Hann" is the most suitable window type to meet this standard. By using the table shown above, the window length L can be calculated as $1 + (6.22 \times \pi)/\Omega\Delta = 170.33$, which rounds to the integer 171. After that, assume that the cut-off frequency f_{c1} is exactly halfway between f_{p1} and f_{stop1} , the normalized cut-off frequency w_{c1} should be calculated as $(2 \times f_{c1})/f_{s1} = 0.325$ rad/sample.

The specifications of designing this FIR low pass filter are shown in Table 2.

Table 2. FIR low pass filter specifications of the audio signal

Filter Performance Parameter	Value
Pass Band Frequency (f_{p1})	3384 Hz
Stop Band Frequency (f_{stop1})	3789 Hz
Pass Band Ripple (r_p)	0.135 dB
Transition Band Frequency (f_t)	405 Hz
Sampling Frequency (f_s)	22.05 kHz
Window Length (L)	171

By using "fir1" function in MATLAB, a FIR low pass digital filter can be designed to determine the 1×171 row scale impulse response vector "hb", which the function is written as "hb=fir1(170, wc1, 'low', hann(171))" (170 is the filter order). Then the frequency response of this system can be determined use the MATLAB function "freqz", which is written as "freqz(hb,22.05)". The plot of it can be seen in Figure 2.

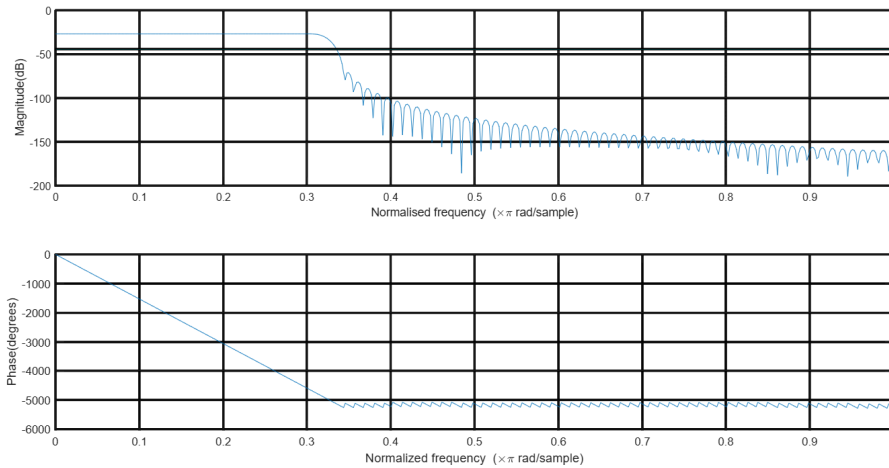


Figure 2. Frequency response of the designed "hann" low pass filter

The final step is to filter the audio signal, which is to display the original and lowpass-filtered signal. The low pass filter output can be determined by using the MATLAB function "filter", which can be written as "y1 = filter(hb,1, audioOrig)". The plots of the original and lowpass-filtered signal are shown in Figure 3.

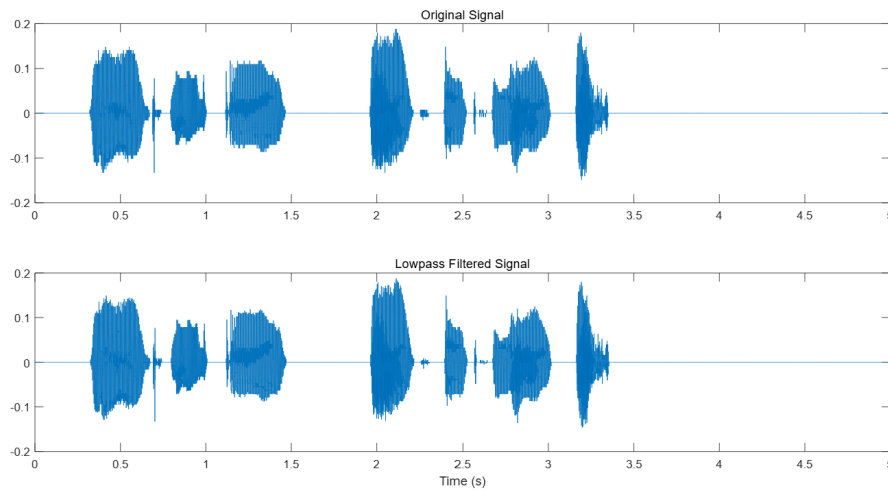


Figure 3. Comparison plot of original and lowpass-filtered signal

Using the same specification, the plots of the original and high pass-filtered signal are shown in Figure 4.

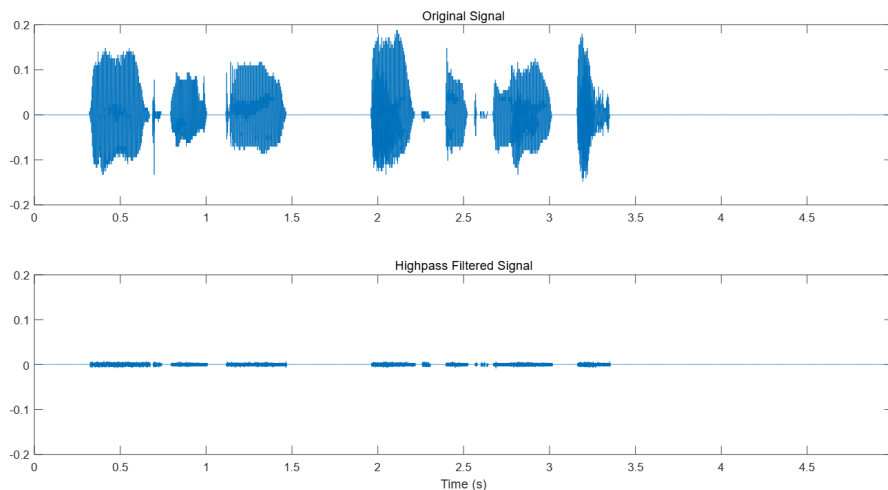


Figure 4. Comparison plot of original and highpass-filtered signal

By comparing these two plots, the FIR low-pass digital filter is more suitable for this 5-second audio signal, and this is the reason to design a low-pass filter for the audio signal at the beginning of the study, which is explained in the two figures above. The low-pass filtered signal shows almost the same shape and unapparent change as there are too many elements, but the high-pass filter largely attenuates the original audio signal, which is not appropriate to use.

By looking at this example, the FIR digital filter is very useful in using a continuous audio signal and processing it, and the "Hann" window is more suitable to use in this sample to meet the frequency specifications. However, there will be different window methods to be used with different filter design specifications.

3.2. Transmission signal from satellite

The data in the second example is related to satellite transmission and mainly focuses on the analysis of the FFT of the signal. Assuming there is a new deep space satellite deployed by SpaceZ, the signal is received from an engineer at SpaceY. The terrestrial hardware is being tuned using transmissions from the satellite and a sequence of time-series data with 3900 elements called "satellite_data" was sampled at a frequency f_{s2} of 3900 Hz. A signal frequency of 1160 Hz from the satellite is expected to be observed. Assumption conditions are used in this study, and the data is generated by using a complex sine function plus a noise function in MATLAB. It is only used as the database of this theoretical research, which is not practical data and not for practical use.

The double-sided FFT is calculated at first by using the MATLAB function "fft" and the plot of the magnitude of this FFT output as a function of Hz frequency with the range of 0 to $f_{s2} - 1$ then can be determined in Figure 5.

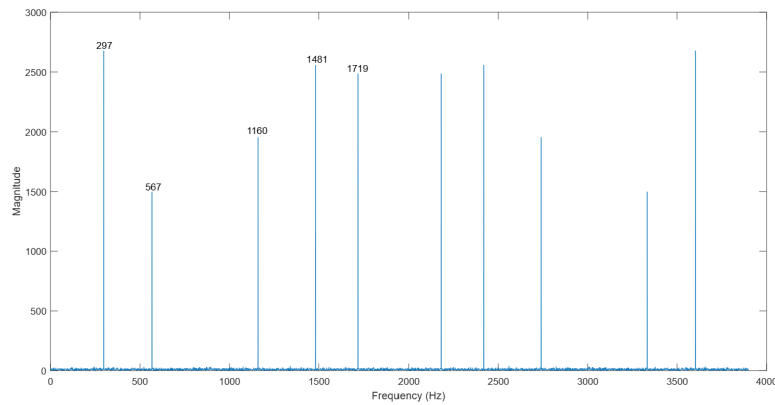


Figure 5. Magnitude of FFT output as a function of frequency in Hz

It is very apparent from this FFT output plot that there are unexpected signal components at other frequencies, which assumes that there is no aliasing. As the signal with frequency 1160 Hz is expected to be detected, the frequency at the other 4 frequencies needs to be removed. To clean up these signals, a scaled FIR bandpass digital filter with the order "Nband" of 200, cut-off frequencies f_{c1} of 1104 Hz and f_{c2} of 1216 Hz, and the "blackman" window type is designed.

The specifications of this bandpass FIR digital filter are shown in Table 3.

Table 3. FIR pass band filter specifications of the transmission signal

Filter Performance Parameter	Value
Cut Off Frequency 1 (f_{c1})	1104 Hz
Cut Off Frequency 2 (f_{c2})	1216 Hz
Sampling Frequency (f_s)	22.05 kHz
Window Length (L)	200

Before determining the discrete-time impulse response of this band pass filter, the normalized cut-off frequencies w_{c1} and w_{c2} need to be calculated first as $(2 \times f_{c1})/f_{s2}$ and $(2 \times f_{c2})/f_{s2}$ respectively, which are 0.566 rad/sample and 0.624 rad/sample. Then the MATLAB function "fir1" is used to design the bandpass filter and determine the impulse response "hband", which can be written as "hband=fir1(Nband, [wband1 wband2], 'band', blackman(Nband+1))" and is a 1×201 row vector. At this time, the output "y2" of this bandpass filter can be determined by using the function of "filter",

which is expressed as "y = filter (hband,1, satellite_data)". Finally, the plot of magnitude of FFT with output y2 can be determined, which can be seen in Figure 6.

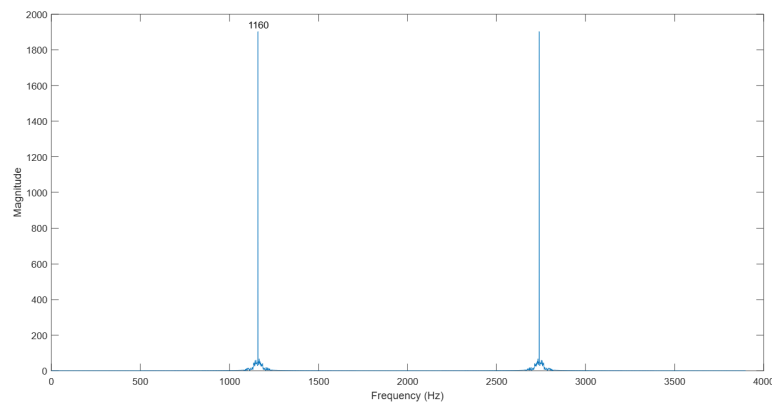


Figure 6. Magnitude of FFT output as a function of frequency in Hz after passing the bandpass filter

By comparing the plots before and after designing the FIR bandpass filter, the unexpected signals have already been removed, which demonstrates that the FIR digital filter is used to filter out the unwanted signal and leave the wanted signal. Although the magnitude of the signal is slightly lower than before, it almost does not influence the overall performance. This is very useful in signal processing in modern society.

4. Conclusion

The FIR digital filter is a very useful tool to process signals in digital signal processing nowadays. In this research, two samples are mainly focused on to analyze the properties and effects of the FIR digital filter: one is the audio signal, and the other is the satellite transmission signal. There are 4 fixed window methods that are mainly considered to design a FIR digital filter, which include rectangular, hann, hamming and blackman. For the audio signal, the optimized filter with the appropriate window length can be designed to meet the given specifications, which is the "Hann window type lowpass filter", and the frequency response is given to show the property of this designed filter. Also, it is proved that the audio signal is suitable for processing it by using low-pass filter. For the transmission signal, the FFT is mainly used to analyze the signal, and the digital filter can be used to filter out the unwanted signal by using the "Blackman" window type to design a bandpass filter, which is also very useful in processing data.

With different examples and specifications, the properties of the filter shown in the frequency response are also different. However, these two simple examples are introduced to design methodology for the optimized filter by only using software in MATLAB to theoretically show some fundamental effects; the hardware implementation also can be used to analyze more experimentally and precisely. In addition, many other uses and properties of digital filters are also very useful to be studied in the future.

References

- [1] Gawande, Gopal S., K. B. Khanchandani, and T. P. Marode. (2012) Performance Analysis of FIR digital Filter Design Techniques. *International Journal of Computing and Corporate Research*, 2.1.
- [2] Zahoor, Sumbal, and Shahzad Naseem. (2017) Design and implementation of an efficient FIR digital filter. *Cogent Engineering*, 4.1: 1323373.

- [3] Yanjie Xu. (2021) Design of FIR filter with several window functions. 2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC), IEEE.
- [4] Gupta, Sonika, and Aman Panghal. (2012) Performance analysis of fir filter design by using rectangular, hanning and hamming windows methods. International Journal of Advanced Research in Computer Science and Software Engineering, 2.6.
- [5] Rajput, Saurabh Singh, and Dr SS Bhadauria. (2012) Implementation of fir filter using efficient window function and its application in filtering a speech signal. International Journal of Electrical, Electronics and Mechanical Controls, 1.1.
- [6] Islam, S., et al. (2014) Design of FIR filter using hamming window. Int. J. Emerg. Res. Manag. Technol, 3.2: 13-16.
- [7] Chonghua Li. (2010) Design and realization of FIR digital filters based on MATLAB. 2010 International Conference on Anti-Counterfeiting, Security and Identification, IEEE.
- [8] Kamal Hossain, et al. (2021) A review of Digital FIR Filter Design in communication Systems. International Journal of Science and Research (IJSR), 10.2.
- [9] Saramäki, Tapio, S. K. Mitra, and J. F. Kaiser. (1993) Finite impulse response filter design. Handbook for digital signal processing, 4: 183.
- [10] Smith, Julius O. (2007) Mathematics of the discrete Fourier transform (DFT): with audio applications. Julius Smith.
- [11] Islam, Saiful, and Md Islam S. (2019) A comparative study on discrete Fourier transformation for digital signal analysis. International Research Journal of Nature Science and Technology.
- [12] Sattari, Omar. (2004) Fast fourier transforms on a distributed digital signal processor. Diss. University of California, Davis.
- [13] Selesnick, Ivan W., and Gerald Schuller. (2001) The discrete Fourier transform. The transform and data compression handbook, pp. 37-74.