# Finite Impulse Response (FIR) Filter Design

D

## Objectives

1. To design low-pass, high-pass, band-pass and band-stop FIR filters using different windowing techniques.
2. To compare the frequency responses of filters designed with Rectangular, Blackman, Hamming, Hann, and Triangular windows.
3. To understand the impact of the window type on the filter's performance.

## Apparatus

Software: MATLAB R2022b

Hardware: Personal Computer

## Theory

Filters play an essential role in signal processing by altering the frequency content of signals, either by amplifying or attenuating specific frequency components. In digital signal processing, filters are classified into two types: Infinite Impulse Response (IIR) filters and Finite Impulse Response (FIR) filters.

FIR filters are advantageous because they rely solely on present and past input samples, eliminating feedback and ensuring stability. They are also less prone to numerical errors during implementation, which is crucial in high-precision applications.

FIR filters are widely utilized in various applications because of their inherent stability and ability to preserve a linear phase. This linear phase property is important for ensuring accurate signal processing in areas like data transmission, telecommunications, and image processing.

## FIR Filter Design

The FIR filters are designed by modulating the impulse response h(n), to approximate a desired frequency response $H(e^{j\Omega})$. The general mathematical expression for an ideal filter's (e.g., low-pass, high-pass, band pass and band stop) impulse response is given by the following integral,

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\Omega})e^{j\Omega n}\, d\Omega \quad \text{for } -\infty < n < \infty$$

Where $\Omega = 2\pi$ radians.

However, this ideal impulse response has an infinite duration, which is impractical for real-world applications. Therefore, to create a finite-duration filter, a window function is used to truncate the ideal response. This method allows the designer to customize the filter's frequency characteristics while making it implementable in digital systems.

An impulse function is often used to test filters as it contains all frequencies simultaneously. When an impulse signal is applied to a filter, the output is the impulse response of the filter, which fully characterizes its behavior. Since the impulse function in the time domain corresponds to a flat spectrum in the frequency domain, all frequencies are present with equal amplitude. This allows us to directly observe how the filter attenuates or passes different frequencies, providing a clear understanding of the filter's frequency response.

For example, when an impulse function is applied to a low-pass filter, the impulse response smooths out as higher frequencies are attenuated. In contrast, a high-pass filter's impulse response exhibits rapid oscillations, reflecting its attenuation of lower frequencies. A band-pass filter isolates specific frequencies, showing oscillations at those frequencies, while a band-stop filter suppresses certain frequencies, with its impulse response lacking oscillations in the stopband.

- Low-pass filter

The ideal impulse response for a low-pass filter is given by the following equation.

$$h(n) = \begin{cases} \dfrac{\Omega_c}{\pi} & for\ n = 0 \\ \dfrac{\sin(\Omega_c n)}{n\pi} & for\ n \neq 0 \end{cases} \quad -M \le n \le M$$

Where $\Omega_c$ is the cutoff frequency in normalized units, and M represents the order of the filter.

- High-pass filter

A high-pass filter allows frequencies higher than a specified cutoff frequency ($\Omega_c$) to pass while attenuating lower frequencies. The ideal impulse response for a high-pass filter is derived by subtracting the response of a low-pass filter from a delta function, effectively removing the low-frequency components.

The mathematical expression is given as:

$$h(n) = \begin{cases} \dfrac{\pi - \Omega_c}{\pi} & for\ n = 0 \\ \dfrac{\sin(\Omega_c n)}{n\pi} & for\ n \neq 0 \end{cases} \quad -M \le n \le M$$

This filter is commonly used in applications such as signal differentiation, edge detection in images, and removing low-frequency noise.

- Band-pass filter

A band-pass filter allows frequencies within a specified range (passband) between a lower cutoff frequency ($\Omega_L$) and an upper cutoff frequency ($\Omega_H$) to pass through while attenuating frequencies outside this range. The ideal impulse response is derived by subtracting the response of a low-pass filter with the lower cutoff frequency ($\Omega_L$) from another low-pass filter with the upper cutoff frequency ($\Omega_H$).

The impulse response is given as:

$$h(n) = \begin{cases} \dfrac{\Omega_H - \Omega_L}{\pi} & for\ n = 0 \\ \dfrac{\sin(\Omega_H n)}{n\pi} - \dfrac{\sin(\Omega_L n)}{n\pi} & for\ n \neq 0 \end{cases} \quad -M \leq n \leq M$$

This filter is widely used in applications such as audio processing, communication systems, and biomedical signal analysis, where isolating signals in a specific frequency range is essential.

- Band-stop filter

A band-stop filter attenuates frequencies within a specified range (stopband) between a lower cutoff frequency ($\Omega_L$) and an upper cutoff frequency ($\Omega_H$) while allowing frequencies outside this range to pass. It is the complement of a band-pass filter, isolating the unwanted frequency range.

The ideal impulse response is given as:

$$h(n) = \begin{cases} \dfrac{\pi - \Omega_H + \Omega_L}{\pi} & for\ n = 0 \\ \dfrac{\sin(\Omega_H n)}{n\pi} + \dfrac{\sin(\Omega_L n)}{n\pi} & for\ n \neq 0 \end{cases} \quad -M \leq n \leq M$$

Alternatively, the band-stop filter can be constructed by subtracting the impulse response of a band-pass filter ($h_{band-pass}[n]$) from a delta function (δ[n]).

This type of filter is commonly used in applications such as noise reduction, where specific frequency bands (e.g., powerline interference at 50/60 Hz) need to be attenuated while preserving the rest of the signal.

**Frequency Response and Lobes**

The frequency response of a filter describes its ability to modify the amplitude and phase of input signal frequencies. In FIR filters, the frequency response depends on the impulse response and the specific window function used during the filter design process.

For ideal filters, the transition between passband and stopband would be instantaneous, achieving perfect attenuation of undesired frequencies. However, practical filters exhibit gradual transitions due to the truncation of the ideal impulse response, leading to spectral leakage where some unwanted frequencies are partially transmitted.

The magnitude response of an FIR filter typically shows a prominent central peak, referred to as the main lobe, which defines the frequency range allowed by the filter. Surrounding this main lobe are smaller peaks known as side lobes, which correspond to leakage into frequencies outside the desired range.

The main lobe's width determines how sharply the filter can differentiate between the passband and stopband. Narrower main lobes provide sharper transitions, improving frequency selectivity. Conversely, the height of the side lobes reflects the extent of spectral leakage, with smaller side lobes indicating better attenuation of unwanted frequencies. The choice of window function is crucial in balancing these characteristics. Different windows (Rectangular, Blackman, Hamming, Hann, and Triangular) produce varying levels of main lobe sharpness and side lobe attenuation.

**MATLAB Code Overview**

- User Input for Filter Design

  The user inputs, such as the normalized cutoff frequency and the filter order, are essential for designing the FIR filters. The normalized cutoff frequency specifies the threshold beyond which frequencies are attenuated, while the filter order determines the steepness of the filter's transition band, with higher orders yielding sharper transitions.

  - MATLAB code for obtaining the user inputs is

```
f = input('Enter cut off frequency (normalized,0<f<1): ');
N = input('Enter filter order (N > 1): ');
```

- Frequency Range for Visualization

  A vector w0 is defined to represent the range of normalized angular frequencies for analyzing the frequency response of the filter.

  - MATLAB code to define the frequency range for visualization is,

```
w0 = 0:0.1:pi;
```

- Window Functions for Designing the FIR Filters

  1. Rectangular Window:

     The Rectangular window applies a uniform truncation of the impulse response. It results in a narrow main lobe but higher side lobe levels, which can lead to significant spectral leakage. Mathematically, the Rectangular window can be expressed as follows.

$$W_R(n) = 1 \quad for\ 0 \leq n \leq N-1$$

     where $N$ is the length of the window.

- MATLAB code to generate the Rectangular window and compute the corresponding frequency response for the Low-Pass Filter design is,

```
% FIR Filter using Rectangular Window (Low-Pass)
b = fir1(N, f/pi, 'low', rectwin(N+1));

% Calculate frequency response
h = freqz(b, 1, w0);
```

2. Blackman Window:

The Blackman window offers better stopband attenuation by tapering the edges of the impulse response. It has a wider main lobe compared to the rectangular window, but its side lobes are significantly lower, making it more effective in suppressing leakage. This is especially useful in applications where minimizing side lobes is critical, such as in spectral analysis and filter design. While the Blackman window has a wider main lobe, its lower side lobes make it ideal for applications where spectral leakage needs to be minimized.
Mathematically, the Blackman Window can be expressed as:

$$W_{Blc}(n) = 0.42 - 0.5\ \cos\left(\frac{2\pi n}{N-1}\right) + 0.08\ \cos\left(\frac{4\pi n}{N-1}\right)\ for\ \ 0 \le n\ \le N-1$$

- MATLAB code to generate the Blackman Window and compute its corresponding frequency response for the Low-Pass Filter design is,

```
% FIR Filter using Blackman Window (Low-Pass)
b = fir1(N, f/pi, 'low', blackman(N+1));

% Calculate frequency response
h = freqz(b, 1, w0);
```

3. Hamming Window:

The Hamming window provides a good compromise between main lobe width and side lobe attenuation. This window function is designed to reduce spectral leakage, balancing both the width of the main lobe and the level of side lobes. It is widely used in signal processing and filter design to reduce the impact of spectral leakage while maintaining reasonable frequency resolution.

$$W_{hm}(n) = 0.54 - 0.46\ \cos\left(\frac{2\pi n}{N-1}\right)\ for\ \ 0 \le n\ \le N-1$$

- MATLAB code to generate the Hamming Window and compute its corresponding frequency response for the Low-Pass Filter design is,

```
% FIR Filter using Hamming Window (Low-Pass)
b = fir1(N, f/pi, 'low', hamming(N+1));

% Calculate frequency response
h = freqz(b, 1, w0);
```

4. Hanning Window:

The Hanning window (also known as the Hann window) is similar to the Hamming window but with slightly different coefficients. The Hanning window provides a smoother transition between the main lobe and side lobes, leading to reduced side lobe levels and less spectral leakage compared to the rectangular window. Mathematically, the Hanning window can be expressed as:

$$W_{hn}(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) \; for \;\; 0 \leq n \; \leq N-1$$

- MATLAB code to generate the Hanning window and compute its corresponding frequency response for the Low-Pass Filter design is,

```
% FIR Filter using Hann Window (Low-Pass)
b = fir1(N, f/pi, 'low', hann(N+1));

% Calculate frequency response
h = freqz(b, 1, w0);
```

5. Triangular Window:

The Triangular window is a simple window function where the weights follow a linear rise to the center and then decrease symmetrically. It is used in signal processing to reduce spectral leakage while maintaining a straightforward shape. The Triangular window has a shape similar to a triangle, which gives it its name. Mathematically, the Triangular Window can be expressed as:

$$W_{Tri}(n) = 1 - \frac{|2n - (N-1)|}{N-1} \; for \;\; 0 \leq n \; \leq N-1$$

- MATLAB code to generate the Triangular window and compute its corresponding frequency response for the Low-Pass Filter design is,

```
% FIR Filter using Triangular Window (Low-Pass)
b = fir1(N, f/pi, 'low', triang(N+1));

% Calculate frequency response
h = freqz(b, 1, w0);
```

In the mathematical derivations for each window functions, n is the sample index, ranging from 0 to N−1, and N is the length of the window (or filter order). The parameter n represents the position within the window sequence.

**Procedure**

1. Open MATLAB and create a new script by navigating to **Home → New → Script**.

2. Begin your script by prompting the user to input the cutoff frequency (f) and the filter order (N). Ensure that the cutoff frequency is normalized, i.e., between 0 and 1, where 1 corresponds to the Nyquist frequency.

3. Define the frequency range over which the filter's frequency response will be visualized.

4. Design the low pass, FIR Filters to visualize their magnitude responses of using various window Functions.

5. Save the script with an appropriate name, such as low_pass_fir_filters.m.

6. Run the script to execute the FIR filter design process, plot the magnitude responses as shown in Figure 1 (Hint: Consider the cutoff frequency fc = 0.5 and the filter order N = 7 for the design).

7. Save the generated figure by selecting File → Save As in the Figure window and choose an appropriate location and file name for future reference.

8. To generate the high pass responses in Figure 2, open a new script, write the corresponding MATLAB program, and repeat the process used for the low-pass filter design. Ensure that the parameter '`high`' is used in the `fir1` function for each window type. (Hint: Consider the cutoff frequency fc = 0.5 and the filter order N = 20 for the design).

## Results

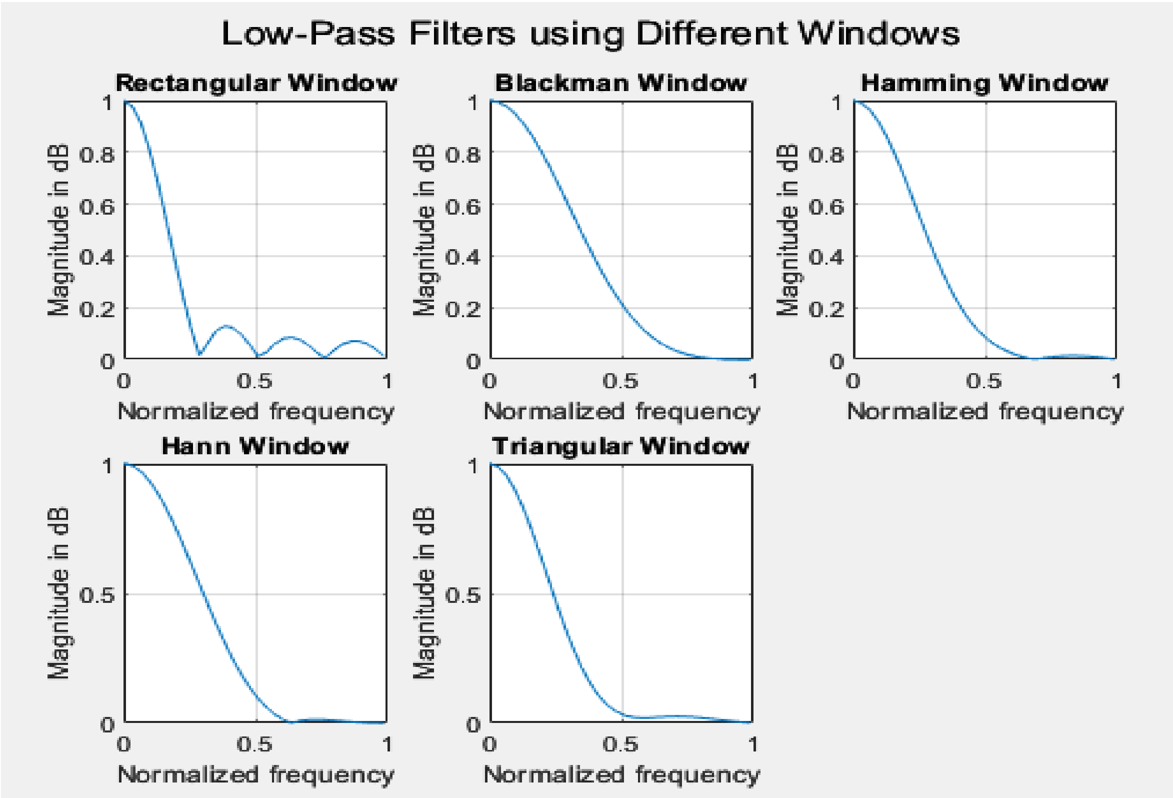


Figure 1. The magnitude responses for different window functions with a low-pass filter.

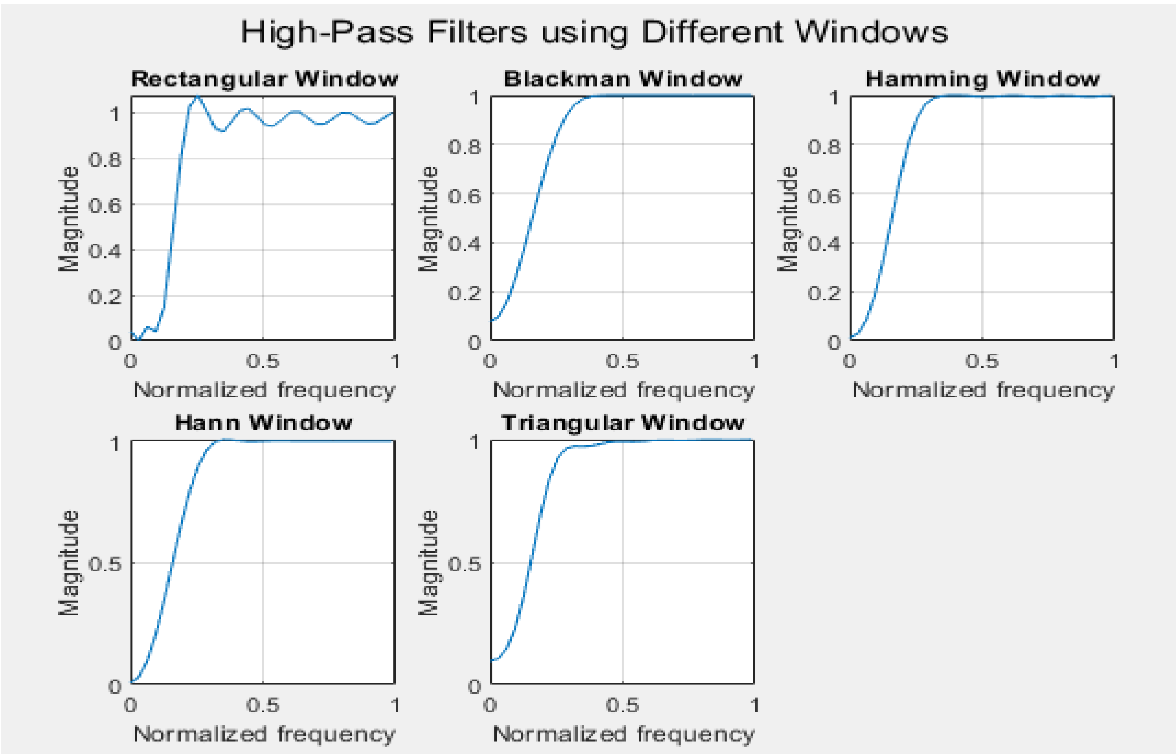


Figure 2. The magnitude responses for different window functions with a high-pass filter.

**Exercise:**

1. (a) Write a MATLAB program to generate the following multi-tone discrete-time signal in green.

$$x[n] = \cos(0.3\pi n) + \cos(0.6\pi n) + \cos(0.9\pi n)$$

   (b) Design a band-stop filter to attenuate the $0.6\pi$ frequency component using the Hamming window in blue.
   (c) Simulate the effect of filtering in MATLAB by plotting the spectrum of the input and output signals in magenta. Discuss the effectiveness of the filter based on the plots.

2. (a) Write a MATLAB program to design bandpass filters using the Rectangular, Blackman, Hamming, Hann, and Triangular window functions. Use normalized passband edge frequencies fp = [0.15, 0.45] (normalized values) and a filter order of N = 30. Plot the magnitude responses (in decibels) for all five filters on four subplots within the same figure to facilitate comparison.

   (b) Using the Hamming window, design band-pass filters with the same passband edge frequencies as in part (a) but with varying filter orders of N = 13, N = 14, and N = 15. Plot the magnitude responses (in decibels) for all three filters on the same graph to compare the effects of filter order on the frequency response.

3. Design band-stop FIR filters using the Rectangular, Blackman, Hamming, Hann, and Triangular window functions with stopband edge at $f_{s1} = 0.1$ and $f_{s2} = 0.8$ (normalized values) and filter order is N =20.

   (a) Plot the magnitude and phase responses (in decibels and radians, respectively) of these filters on two subplots for comparison.

   (b) Analyze the results. Compare the filters based on main lobe width, side lobe attenuation, and computational complexity. Discuss the trade-offs between sharp transitions and side lobe suppression and recommend the best window for applications requiring sharp transitions or low side lobes.

4. Design a low-pass filter with a cutoff frequency fc = 0.2 and a high-pass filter with a cutoff frequency fc = 0.8 (normalized) using the Triangular window in red.

   (a) Combine the two filters design above to create a bandpass filter that passes frequencies between 0.2 and 0.8 (normalized). Use the `conv` function to combine the filters.

(c) Plot the magnitude response (in decibels) of the resulting band-pass filter. Explain how the low-pass and high-pass filters together form the band-pass filter, focusing on their contribution to the passband and stopband.