Astronomy&Astrophysics(Caucasus) 9,(2025)

IMPLEMENTATION AND EVALUATION OF A 10TH-ORDER LOW-PASS FINITE IMPULSE RESPONSE FILTER

Julieta Tabeshadze
PhD in Informatics,
Samtskhe-Javakheti State University,
Rustaveli St. 124, Akhaltsikhe, Georgia
julietatabeshadze@gmail.com
https://orcid.org/0009-0008-7463-7801

Abstract

This paper presents the process of designing and analyzing a low-pass FIR (Finite Impulse Response) filter using window functions in the MATLAB environment. The theoretical foundation is discussed, including impulse response, transfer function, and stability considerations. The study is based on an experiment aimed at constructing a practically realizable version of an ideal filter, calculating its coefficients, and comparing its frequency characteristics. The results demonstrate that the resulting 10th-order filter closely approximates an ideal low-pass filter and exhibits both stability and a linear phase response.

Keywords: FIR filter, low-pass filter, window function, MATLAB, freqz, impulse response, phase response

Introduction

Low-pass filters are one of the fundamental components in digital signal processing, enabling the suppression of unwanted high-frequency components from signals. While the realization of an ideal filter is not feasible in practice, it is possible to approximate its behavior using window functions. This paper aims to generate a low-pass FIR filter using the window method in MATLAB, supported by theoretical analysis and comparison with practical results.

Theoretical Background

An FIR filter (Finite Impulse Response filter) is a non-recursive digital filter. FIR filters operate by processing the input signal using a set of filter coefficients and previously received input samples.

The operation of a finite impulse response filter can be described mathematically as follows [1]:

$$y(k) = b_0 \cdot x(k) + \sum_{i=1}^{N} b_i \cdot x(k-i)$$

- The coefficients $\boldsymbol{b_0}, \boldsymbol{b_1}, \dots, \boldsymbol{b_{\mathrm{N}}}$ define the weighting factors of the filter. There are N+1 coefficients in total, where N is the order of the FIR filter. This means that the number of coefficients is always one greater than the filter's order.
- Designing an FIR filter requires specifying its order and identifying a set of coefficients that satisfy the desired frequency response or filtering criteria.
- The output signal is obtained by computing the weighted sum of the current and preceding N input samples, where each input is multiplied by its corresponding coefficient.
- To implement the FIR filter, one must allocate memory to store the coefficients, the current input, and a record of the previous N input samples. Since the filter output is not influenced by past outputs, this structure is classified as non-recursive.

Impulse Response Characteristics of an FIR Filter

The behavior of discrete-time linear time-invariant (LTI) systems can be characterized through their impulse response—often referred to as the impulse transfer function. This response illustrates how the system reacts when subjected to a unit impulse input (commonly called a delta function). It describes the output generated by the system when it begins from a resting state, with no initial internal energy present.

The discrete-time impulse signal δ(m) is defined as:

$$\text{Unit Impulse } \delta(m) = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases}$$

If a filter’s output relies solely on the present and past input samples—without referencing any prior output values—then the system is considered non-recursive. In such cases, In FIR filters, the impulse response matches the coefficients that appear in the system’s difference equation [1].

$$\sum_{n=0}^{N} a_n \cdot y(k-n) = \sum_{m=0}^{L} b_m \cdot x(k-m)]$$

$$h(m) = [b_0 \quad b_1 \quad b_2 \quad \dots \quad b_N]$$

FIR Filter Transfer Function

In the context of discrete-time signal processing, applying the Z-transform to the filter's difference equation yields its transfer function. This procedure yields a general expression of the system's frequency-domain behavior. [5]

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{1} = \frac{b_0 z^N + b_1 z^{N-1} + b_2 z^{N-2} + \cdots + b_N}{z^N}$$

FIR Filter Stability

The inherent stability of FIR filters stems from the fact that their poles are located at the origin in the Z-plane, making them more robust than many recursive filter designs.

Output Values of an FIR Filter

The output values of an FIR filter can be calculated using various methods, such as:

- Input-output difference equations
- Convolution: $y(m) = x(m) * h(m)$
- Using the Inverse Z-Transform $Y(z) = X(z)H(z)$
- By executing the MATLAB function `y = filter(num_tf, den_tf, x)`, which applies a digital filter to the input signal.
- Through the application of the Fast Fourier Transform (FFT) to perform frequency-domain filtering.

Window-Based Design of FIR Filters via Fourier Series Approach

The windowing method for FIR filter design starts by defining an ideal filter response, which is subsequently modified to create a realizable implementation. The amplitude and phase characteristics of an ideal low-pass digital filter are shown below.

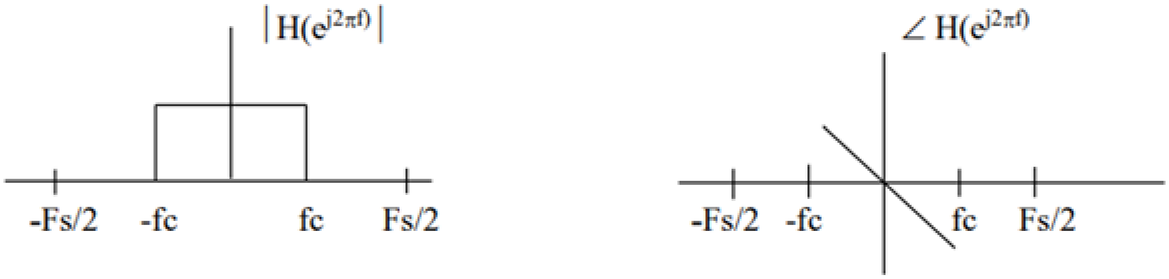

Fig. 1. Visualization of the Frequency Response (Amplitude and Phase) for an Ideal FIR Low-Pass Filter

Where fc = is the desired analog cutoff frequency, which defines the frequency range that the filter should pass or attenuate.
Fs = represents the sampling frequency, indicating how many samples are taken per second by the system.

The impulse response corresponding to the ideal filter is determined through the inverse Fourier series representation of its frequency response. [2].

$$h(m) = \int H\left(e^{j2\pi f m}\right)e^{j2\pi f m}df$$

Evaluating the integral yields the following expression:

$$h(m) = \frac{sin\left(2\pi \frac{f_c}{F_s} m\right)}{\pi m} = 2\left(\frac{f_c}{F_s}\right) sinc\left(2 \cdot \frac{f_c}{F_s} m\right) \qquad m \in \mathbb{Z}$$

$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

The conceptual model of an ideal low-pass filter helps us understand its characteristics; however, it cannot be implemented in practical hardware due to the infinite number of coefficients and limitations of time-domain systems. The following equation expresses the combined effect of the impulse response and the input signal, which represents the output of the ideal low-pass filter. In practice, however, the infinite number of coefficients and the constraints of real-time systems make it impossible to realize the filter exactly as defined.

$$h(m) = \sum_{k=-\infty}^{+\infty} h(k) \cdot x(m-k)$$

Clearly, computing the output of an ideal filter involves an infinite number of multiplications and additions, which is not feasible in practice. Furthermore, it would require future values of the input signal—something that is not accessible in real-time systems, where only present and past values are available.

To overcome these limitations, the impulse response $h(m)$ can be truncated to a finite length and shifted to the right, making the system time-causal. Naturally, this modification results in a deviation from the behavior of an ideal low-pass filter. By restricting the impulse response coefficients to a finite range of values for m, the infinite-length ideal response is approximated by a realizable version, m∈ $\left[-\frac{N}{2}, \frac{N}{2}\right]$ and shifting them right by N/2, one obtains the coefficients of a realizable low-pass FIR filter of order N.

$$h(m) = \frac{sin\left(2\pi \left(\frac{f_c}{\mathrm{F}_s}\right)\left(\mathrm{m} - \frac{\mathrm{N}}{2}\right)\right)}{\pi(m - N/2)} = 2\left(\frac{\mathrm{f_c}}{\mathrm{F}_s}\right) sinc\left(2\left(\frac{\mathrm{f_c}}{\mathrm{F}_s}\right)\left(\mathrm{m} - \frac{\mathrm{N}}{2}\right)\right) \quad \mathrm{m} \in \mathbb{Z}$$

Where $h(N/2) = 2(f_c/F_s)$

The process of truncating the filter coefficients and transforming them into a causal system is illustrated in the figures below.

In the first figure, the coefficients related to the ideal filter design are shown, extended up to $m = +20$. In reality, these coefficients should extend to infinity $(+\infty)$ but for practical reasons, the plot is limited to $m = +20$.

In the second figure, the filter has been truncated to a finite length, with coefficients shown only up to $m = +10$. This means that an infinite number of coefficients is no longer required, making the filter practically implementable.

To achieve causality, the coefficients shown in the third figure are delayed by 10 units. As a result, we obtain a physically realizable 20th-order low-pass FIR filter.

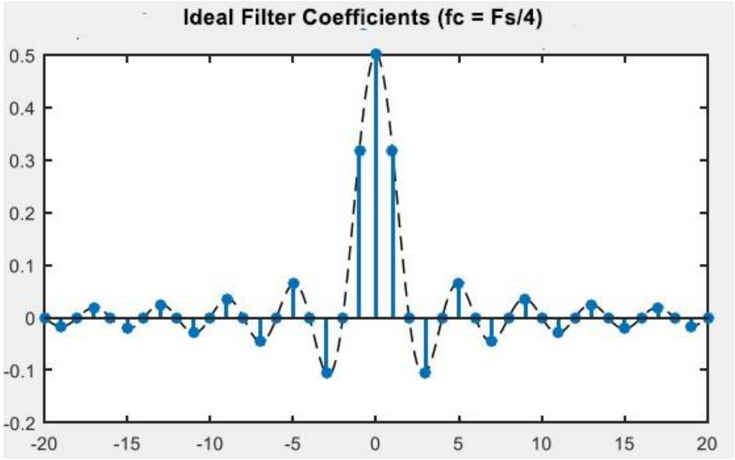

Fig. 2. Time-Domain Coefficient Values of an Ideal FIR Low-Pass Filter$(fc = \mathrm{F}s/4)$

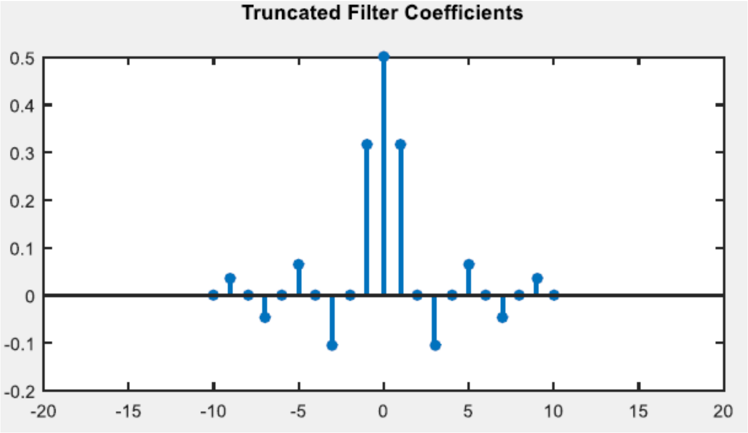

Fig. 3. Truncated filter coefficients

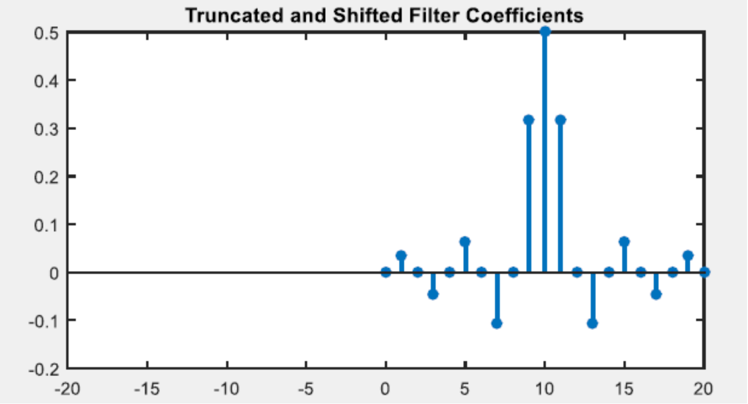

Fig. 4. Truncated filter coefficients shifted right by 10 units

Numerical Experiment:

a) Using 44,1 kHz as the sampling rate and 20 kHz as the cutoff point, calculate the filter taps for a 10 th-order FIR low-pass filter.

b) Formulate the input-output difference equation used to calculate the output signal y(m)y(m)y(m) of the filter.

c) Obtain the Z-domain transfer function corresponding to the designed FIR filter.

d) Visualize the frequency response of the 10th-order FIR filter and compare its behavior against that of an ideal low-pass design to evaluate performance.

a) The coefficients of the filter are obtained by truncating the ideal impulse response and shifting it to achieve causality, as expressed in the following equation:

$$h(m) = \frac{sin\left(2\pi\left(\frac{f_c}{F_S}\right)\left(m-\frac{N}{2}\right)\right)}{\pi\left(m-\frac{N}{2}\right)} = 2\left(\frac{f_c}{F_S}\right) sinc\left(2\left(\frac{f_c}{F_S}\right)\left(m-\frac{N}{2}\right)\right) \quad m = 0,1,2,\dots,N \qquad [4]$$

$$h[m] = A \cdot \mathrm{sinc}\big(B \cdot (m - M)\big), \text{where } A = 2 \cdot \frac{f_c}{F_s}, B = 2 \cdot \frac{f_c}{F_s}, M = \frac{N}{2}$$

$$f_c = 20\mathrm{kHz}, F_s = 44.1\mathrm{kHz}, N = 10$$

The coefficients can be obtained more easily using MATLAB:

```
m = 0:1:10; %  A vector from 0 to 10
h = 2 * (20 / 44.1) * sinc(2 * (20 / 44.1) * (m - 5)); % Calculation of Filter Coefficients
disp(h);
```

A total of 11 coefficients were obtained, exhibiting symmetry around the central peak $h[5] = 0{,}9070$ which is typical for linear-phase FIR filters

b) The coefficients found in the difference equation of an FIR filter correspond exactly to its impulse response samples.

$$y(m) = \sum_{k=0}^{10} h(k) \cdot x(m-k)$$

Where

$$h = [0.0633, -0.0732, 0.0815, -0.0878, 0.0917, 0.9070, 0.0917, -0.0878, 0.0815, -0.0732, 0.0633]$$

The filter's output can be calculated iteratively by utilizing the current input along with the ten preceding samples, requiring a total of eleven multiplications and ten additions.

c) The Z-transform applied to the input-output difference equation yields the transfer function of the filter, which can be expressed as follows:

$$H(z) = [0.0633z^{10} - 0.0732z^{9} + 0.0815z^{8} - 0.0878z^{7} + 0.0917z^{6} + 0.9070z^{5} + 0.0917z^{4} - 0.0878z^{3} + 0.0815z^{2} - 0.0732z + 0.0633]/z^{10}$$

d) The frequency characteristics of the filter can be evaluated in MATLAB through the use of the `freqz` function:

```
% Define coefficients
k = 0:10;
Fc = 20e3;          % Cutoff frequency (Hz)
Fs = 44.1e3;        % Sampling frequency (Hz)
h = 2 * (Fc/Fs) * sinc(2*(Fc/Fs)*(k - 5));

% Define numerator and denominator for the transfer function
numerator = h;
denominator = zeros(1, length(h));
denominator(1) = 1;

% Calculate frequency response
[H, f] = freqz(numerator, denominator, 200, Fs);

% Plot the magnitude and phase response
figure;
subplot(2,1,1);
plot(f / 1000, 20 * log10(abs(H)), 'LineWidth', 1.5);
title('Magnitude Response');
xlabel('Frequency (kHz)');
ylabel('Gain (dB)');
grid on;

subplot(2,1,2);
plot(f / 1000, angle(H), 'LineWidth', 1.5);
title('Phase Response');
xlabel('Frequency (kHz)');
ylabel('Phase (radians)');
grid on;
```

The frequency response of the 10th-order FIR filter is depicted in the figure. To assess its effectiveness, it may be compared to that of an ideal low-pass filter with a cutoff frequency of 20 kHz. Ideally, such a filter would maintain a constant magnitude of 0 dB (or a gain of 1) across the entire passband, up to 20 kHz, with a sharp transition to $-\infty$ dB (or zero gain) for all frequencies beyond that point. Unlike the ideal response, the 10th-order FIR filter exhibits minor ripples within the passband, and the attenuation begins gradually at approximately 17,5 kHz. The 3 dB cutoff occurs near 18 kHz. Importantly, the phase response is linear, closely matching the behavior of an ideal filter. This linear phase characteristic, combined with the filter’s inherent stability, is one of the primary advantages of FIR filters in digital signal processing tasks.

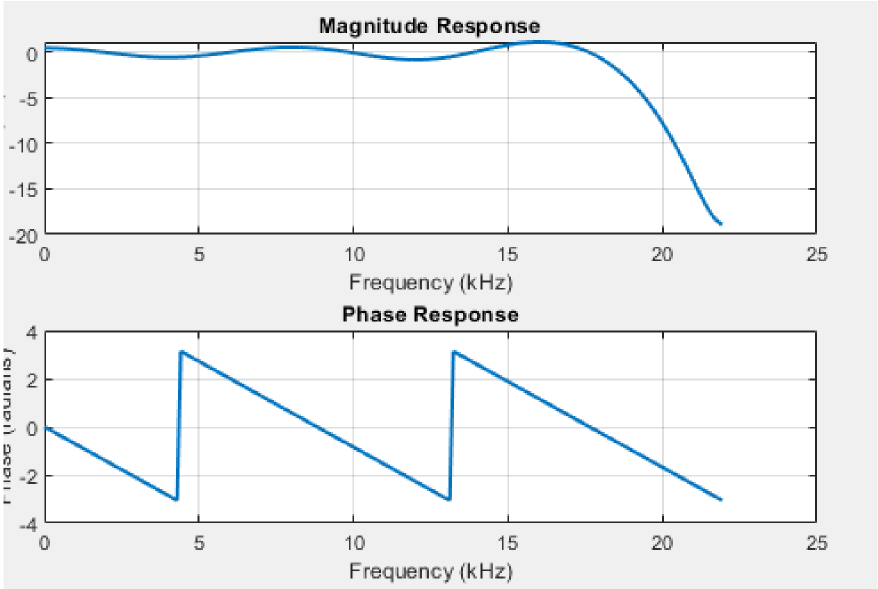

Fig. 5. Frequency response of the FIR filter and its phase response.
A linear phase relationship with respect to frequency is observed, which has a positive effect on time-domain signals.

Results:

- The coefficients of the 10th-order low-pass FIR filter were calculated using MATLAB. The resulting values were: h= 0.0633 -0.0732 0.0815 -0.0878 0.0917 0.9070 0.0917 -0.0878 0.0815 -0.0732 0.0633,

- The frequency response exhibits ripples in the passband.

- The magnitude begins to decrease gradually around 17,5 kHz.

- signal drop of 3 dB is observed near the 18 kHz frequency mark, indicating the filter's effective cutoff.

- Phase behavior of the system is linear, which is beneficial for signal processing applications.

- Unlike an ideal low-pass filter, the designed filter features a transition band in which the magnitude gradually decreases. This is due to the limited filter order and restricts sharp cutoff capability.

- The filter is stable and characterized by a linear phase response.

4. Analysis and Discussion of Results

The frequency response obtained using the coefficients of the designed 10th-order low-pass FIR filter demonstrates a reasonably high-quality filtering effect. Ripples are observed in the magnitude response within the passband, which is typical for filters limited by window functions. However, these ripples are minimal and do not significantly impact the filter's usability.

he cutoff frequency at –3 dB occurs at approximately 18 kHz , which is close to the target value of 20 kHz. At the same time, the phase response is nearly perfectly linear, ensuring minimal phase distortion—an important advantage for processing time-domain signals.

Compared to an ideal filter, the windowed FIR filter exhibits a transition band instead of an abrupt cutoff. This is due to the limited number of coefficients. Nevertheless, the filter's linear phase response, inherent stability, and simple implementation make it a practically successful approximation of the ideal low-pass filter.

5. Conclusion

The design of the low-pass FIR filter discussed in this paper was successfully implemented using MATLAB. The resulting filter is practically realizable and meets the essential functional requirements, including an appropriate cutoff frequency, stability, and a linear phase response. It is particularly effective in reducing noise and preserving the structure of the original signal.

For future work, it is recommended to explore higher-order filters or alternative window functions to further minimize passband ripples and bring the response even closer to that of an ideal filter.

References:

[1] Phagava, A. (2020). *Fundamentals of Biosignal Processing.* https://gtu.ge/book/DSP_Analog.pdf

[2] Qubaneishvili, E. (2018). *Digital Processing of Biosignals.* https://gtu.ge/book/ims/cifruli_damushaveba.pdf

[3] **Oppenheim, Alan V., schafer, Ronald W.,** Discrete-Time signal Processing, 3rd edition, Prentice Hall, 2009;

[4] **John G. Proakis**, **Dimitris G. Manolakis,** "Digital signal Processing: Principles, Algorithms, and Applications" (4th edition, Pearson, 2006);

[5] Michael Parker, Chapter 5 - Finite Impulse Response (სირ) Filters, Editor(s): Michael Parker, Digital Signal Processing 101 (Second Edition), Newnes, 2017, Pages 41-57,

ISBN 9780128114537, https://doi.org/10.1016/B978-0-12-811453-7.00005-6. (https://www.sciencedirect.com/science/article/pii/B9780128114537000056)

[6] Robert Lacoste, Chapter 7 - No Fear with ხიძ: Put a Finite Impulse Response Filter to Work, Editor(s): Robert Lacoste, Robert Lacoste's The Darker Side, Newnes, 2010, Pages 93-110, ISBN 9781856177627, https://doi.org/10.1016/B978-1-85617-762-7.00007-1. (https://www.sciencedirect.com/science/article/pii/B9781856177627000071)

[7] Youssef N. Raffoul, Chapter 3 - Z-transform, Editor(s): Youssef N. Raffoul, Difference Equations and Applications, Academic Press, 2025, Pages 83-136, ISBN 9780443314926, https://doi.org/10.1016/B978-0-44-331492-6.00009-1. (https://www.sciencedirect.com/science/article/pii/B9780443314926000091)