Filter Design and Analysis with (FDA) Tool

Objectives

1. To be familiarized with the FDA Tool in MATLAB for designing digital filters.
2. To understand the process of designing filters and specifying parameters such as filter order, cutoff frequencies, and attenuation requirements.
3. To analyze the performance of the designed filter through their magnitude, phase, impulse, and step responses.

Apparatus

Software: MATLAB R2022b
Hardware: Personal Computer

Theory

Filters are essential components in signal processing, designed to enhance, suppress, or isolate specific parts of a signal. They play a critical role in applications such as noise reduction, signal enhancement, and feature extraction. Filters are broadly classified into two major categories:

- Finite Impulse Response (FIR) Filters**:** The filters have a finite duration impulse response and are known for their inherent stability and linear phase characteristics. FIR filters are especially suitable for applications requiring minimal phase distortion. The design and analysis of FIR filters were explored in laboratory manual 7.
- Infinite Impulse Response (IIR) Filters: The IIR filters employ feedback, resulting in an infinite duration impulse response. They are computationally efficient and commonly used in real-time systems, though they require careful design to ensure stability. The principles of IIR filters were addressed in laboratory manual 8.

The FDA Tool in MATLAB is a graphical user interface (GUI) that simplifies the design, analysis, and implementation of digital filters. It eliminates the need for complex mathematical derivations, enabling users to focus on filter specifications and their real-world applications. The FDA Tool supports the design of both FIR and IIR filters, visualizes the frequency and phase responses of filters, and exports filter coefficients to the MATLAB workspace for further analysis. This tool can be opened by simply typing '`fdatool`' in the command window.

By default, the interface of the FDA Tool is set to design a low-pass filter. It provides users with controls for specifying filter parameters and visualizing their frequency and phase responses, as shown in Figure 1.

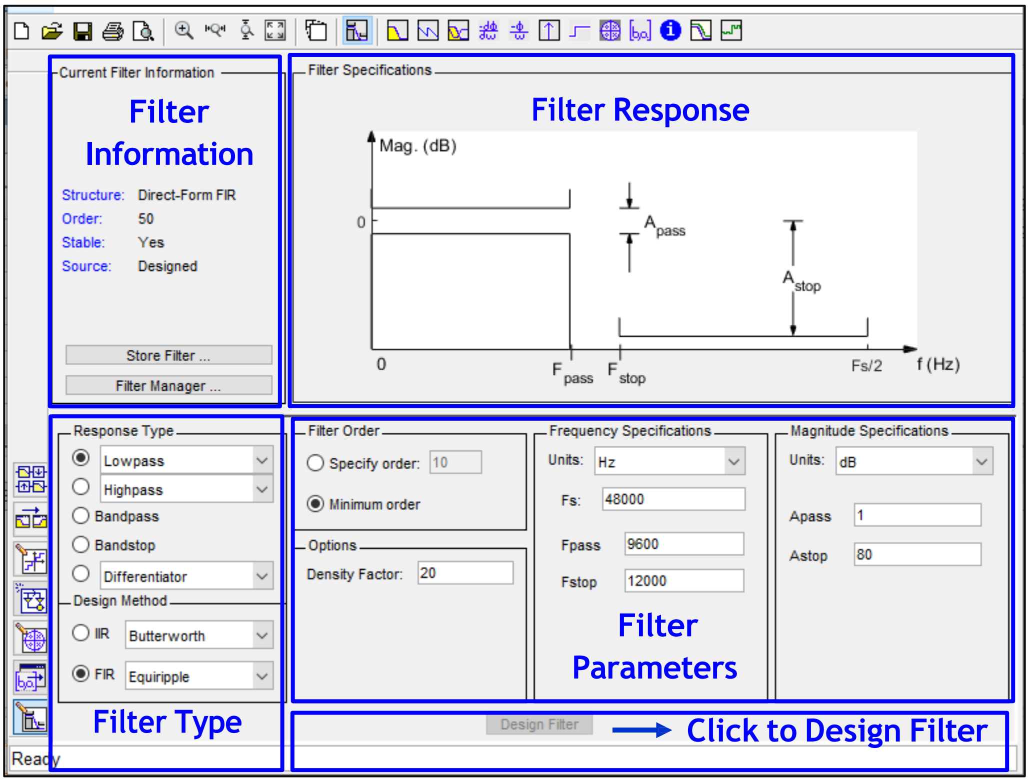


Figure 1. Interface of the FDA Tool.

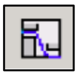
FDA Tool Design Parameters

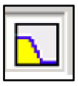
The FDA Tool allows users to define various parameters when designing digital filters. Some of the important parameters include:

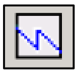
- Filter Type: Users can select from different filter types such as low-pass, high-pass, band-pass, and band-stop. The default filter type is typically a low-pass filter.
- Cutoff Frequency: This defines the frequency at which the filter transitions from passband to stopband. For low-pass and high-pass filters, this is a single frequency, while for band-pass and band-stop filters, two cutoff frequencies are specified.
- Filter Order: The order of the filter determines its complexity and the steepness of the frequency response transition between the passband and stopband. Higher-order filters provide sharper transitions but may introduce more delay or computational complexity.
- Sampling Frequency: This parameter specifies the rate at which the signal is sampled and is used to normalize the frequency parameters (such as the cutoff frequency) in relation to the Nyquist frequency.
- Ripple: For certain filter designs (such as Chebyshev or elliptic filters), users can specify the ripple allowed in the passband or stopband, controlling the filter's amplitude response.

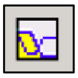
- Stopband Attenuation: This defines the amount of attenuation required in the stopband. It ensures that unwanted frequencies are sufficiently suppressed.
- Passband Edge: This is the frequency at which the filter begins to transition out of the passband into the stopband, crucial for defining the filter's performance in its passband.
- Transition Band: This is the range of frequencies where the filter transitions from the passband to the stopband, and it plays a role in the sharpness of the filter's frequency response.

The toolbar in the FDA Tool provides quick access to various functionalities for filter design and analysis as follows.

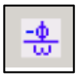

Frequency Response: Displays the filter's frequency response, including both magnitude and phase plots.

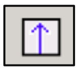

Magnitude Response: Shows only the magnitude response, useful for analyzing the gain or attenuation across frequencies.

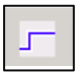

Phase Response: Focuses exclusively on the phase response of the filter, aiding in the analysis of phase distortion.

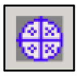

Magnitude & Phase Response: Combines the magnitude and phase plots into one visualization for comprehensive analysis.


Group Delay Response: Shows the group delay of the filter, which indicates the variation of phase delay with frequency and is important for phase-sensitive applications.

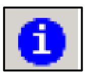

Phase Delay Response: Displays the phase delay of the filter, representing the time shift of sinusoidal components at each frequency.


Impulse Response: Illustrates the time-domain representation of the filter's response to a discrete impulse input.

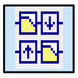

Step Response: Depicts the time-domain representation of the filter's response to a step input.


Pole-Zero Plot: Displays the poles and zeros of the filter in the z-plane, essential for assessing stability and frequency response characteristics.

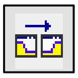

Filter Coefficients: Provides access to the filter's coefficients (numerator and denominator). These coefficients can be exported to the MATLAB workspace for further use in simulations or implementations.


Help: Opens the MATLAB documentation for the FDA Tool, providing detailed guidance on its features and usage.

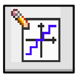
Beyond offering convenient access to fundamental functionalities, the FDA tool incorporates a range of advanced features accessible via its graphical user interface These advanced tools capabilities enhance the tool's flexibility and facilitate the development of complex and sophisticated filter designs.



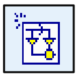
Multilayer Filters: These allow users to combine multiple filters, such as low-pass, high-pass, and band-pass filters, to create complex frequency responses tailored to specific applications. Multilayer filters can be designed by cascading or connecting individual filters within the interface.



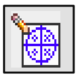
Filter Transformations: The FDA Tool enables the transformation of filters into different types or configurations, such as converting a low-pass filter into a high-pass or band-stop filter. Additionally, analog filters can be transformed into digital filters using methods like Bilinear Transformation or Impulse Invariant techniques.



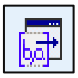
Set Quantization Parameters: Provides control over filter coefficient quantization, allowing users to specify word length and fractional length. This feature is particularly useful when implementing filters on hardware where precision and memory constraints are critical.



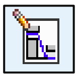
Realization Models: These determine the structural implementation of a filter, such as direct, cascade, or parallel forms. Users can explore these models using the Pole-Zero Plot or other analysis tools available in the GUI.



Pole-Zero Editor: This feature allows users to directly edit the poles and zeros of the filter in the z-plane. By manipulating poles and zeros, users can fine-tune filter stability and frequency response characteristics.



Import from Workspace: The FDA Tool supports importing existing filter coefficients or data directly from the MATLAB workspace. This functionality is useful for refining pre-designed filters or analyzing custom filter designs.



Design Filter: The central functionality of the FDA Tool enables users to design filters by specifying parameters such as filter type, cutoff frequency, filter order, and response characteristics. This process is streamlined through an intuitive interface for both FIR and IIR filters.

These advanced features, combined with the core functionalities of the FDA Tool, provide a comprehensive environment for designing, analyzing, and implementing digital filters in a user-friendly manner.

Exporting a Filter Design

The FDA Tool in MATLAB provides multiple options for exporting filter designs, ensuring flexibility in implementing and sharing designs across various platforms. These export features allow users to leverage their filters effectively in different applications, such as simulation, hardware implementation, or integration into larger systems.

Exporting Coefficients or Objects to the Workspace

Users can export filter coefficients (numerator and denominator) or a filter object directly to the MATLAB workspace. This enables seamless integration into simulations or further manipulation within the MATLAB environment. The exported coefficients can be used with MATLAB functions like ‘`filter`’ or ‘`freqz`’ to analyze or apply the filter to signals.

To export a filter design to the workspace using the FDA Tool, follow these straightforward steps:

1. Select the Export Option: Start by navigating to File → Export. This action opens the Export dialog box, which provides various export settings.
2. From the ‘Export To’ menu, select Workspace as the destination for the exported filter.
3. Specify Export Type: From the ‘Export As’ menu, choose whether to export the filter as Coefficients or as an Object.
4. Assign Variable Names:
    - For coefficients: Use the Numerator (for FIR filters) or Numerator and Denominator fields (for IIR filters), or the SOS Matrix and Scale Values fields (for IIR filters in second-order section form) within the Variable Names region to assign appropriate variable names.
    - For objects: Assign a variable name in the Discrete Filter (or Quantized Filter) text box.
    - If variables with the same names already exist in your workspace and need to be replaced, check the Overwrite Variables option.
5. Click the Export button to complete the process

Exporting to a Simulink Model

The FDA Tool allows users to generate a Simulink block representing the designed filter. This feature simplifies incorporating the filter into Simulink-based models, making it easy to simulate signal processing systems that utilize the filter. The exported block retains all filter characteristics, including coefficients and type, ensuring accurate representation.

To export a filter design as a Simulink block using the FDA Tool, follow these steps:

1. Open the Export Panel: After designing your filter, click the Realize Model sidebar button or select File → Export to Simulink Model. This action displays the Realize Model panel.

2. Specify Block Name: In the panel, assign a name to your block in the Block name field. This name will represent your filter in the Simulink model.
3. Select the Destination:
   - To insert the block into the current (most recently selected) Simulink model, set Destination to Current.
   - To insert the block into a new model, select New.
   - To insert the block into a user-defined subsystem, choose User defined.
4. Overwrite Option: If you want to overwrite an existing block previously created from this panel, check the Overwrite generated 'Filter' block option.
5. Basic Elements Option: If you select the Build model using basic elements check box, your filter will be created as a subsystem block using separate sub-elements for its components.

Alternating Methods for Exporting

The FDA Tool offers additional methods for exporting filter designs, providing flexibility for automation, documentation, and integration into different workflows. The tool can generate MATLAB scripts that recreate the filter design programmatically. This is ideal for documenting the design process or automating filter creation for similar specifications in future projects.

To generate MATLAB code for your filter design using the FDA Tool, follow these steps:

1. Access the Generate Code Option: Navigate to File → Generate MATLAB Code → Filter Design Function. This opens the Generate MATLAB Code dialog box.
2. Specify the Filename: Enter the desired filename for the generated MATLAB code in the dialog box.
3. Click the save button to generate and save the code file.

Saving and Opening Filter Design Sessions

Users can save their filter design sessions as MAT-files to return to them later. To save a session,

1. Click the Save session button.
2. If this is the first time saving, the Save Filter Design File browser opens, prompting you to provide a session name.
3. Save the session as a MAT-file for future access.

Procedure

1. Open MATLAB and type ‘`fdatool`’ in the Command Window to launch the FDA Tool.

2. In the FDA Tool interface, First select ‘FIR’ as the filter type, and then choose ‘Equiripple’ as the design method. Next set the Response Type to ‘Lowpass’ from the drop-down menu.

3. Specify the filter Order as 20 in the filterorder section.

4. Enter a desired Sampling Frequency, such as 1000 Hz, in the Sampling Frequency field of the tool.

5. Set the Passband Frequency (Fpass) to 200 Hz and Specify the Stopband Frequency (Fstop) as 300 Hz in the Filter Specifications panel.

6. Define the normalized digital frequencies for the passband (Wpass) and stopband (Wstop) edges as 0.4 and 0.6, respectively.

7. Click the Design Filter button to generate the FIR filter and the tool will display the Magnitude Response (Figure 1) and Phase Response (Figure 2) of the filter.

8. Use the toolbar to view additional filter responses, including the impulse response (Figure 3) and the step response (Figure 4).

9. Export the filter coefficients and assign appropriate variable names (e.g., b for numerator coefficients).

10. Write a MATLAB code to generate a discrete composite signal (x) consisting of a low-frequency component of 125 Hz and a high-frequency component of 350 Hz, using a sampling rate of 1000 Hz. Visualize the composite signal in the time domain as shown in Figure 5

11. Use the ‘`filter`’ function to apply the designed FIR filter and remove the 350 Hz frequency component from the composite signal.

    Hint: The general form of the filter function is, `y = filter (b, a, x)`.
    Here, b: The numerator coefficients of the filter (FIR/IIR).
    a: The denominator coefficients of the filter (only for IIR filters).
    x: The input signal.
    y: The filtered output signal.

12. Plot the FFT of both the original and filtered signals to visualize the frequency content before and after filtering, under the same sampling rate of 1000 Hz as shown in Figure 6.

13. Save all generated figures in JPEG format with appropriate and descriptive filenames.

Results

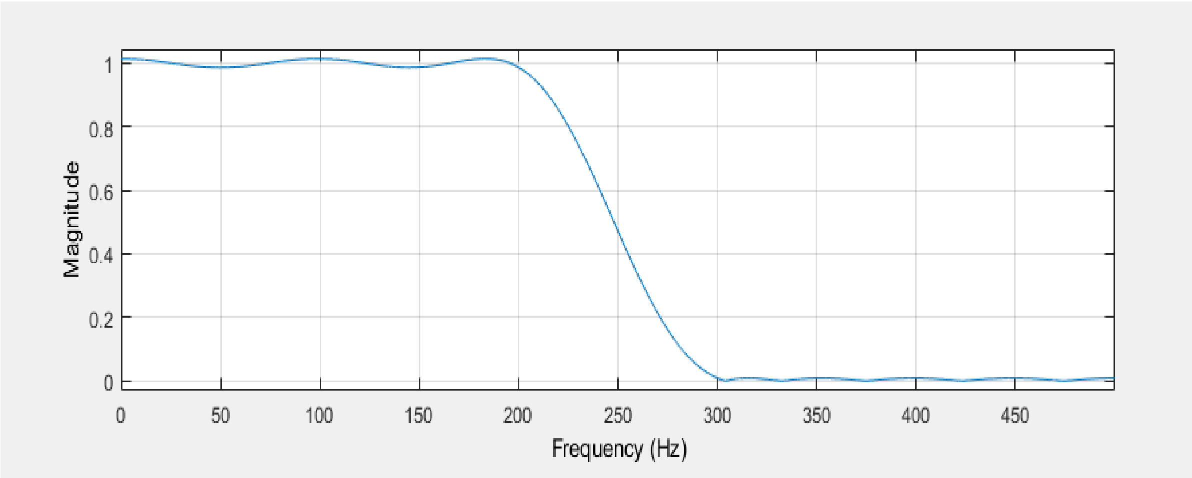


Figure 1. Magnitude Response of the Designed FIR Filter.

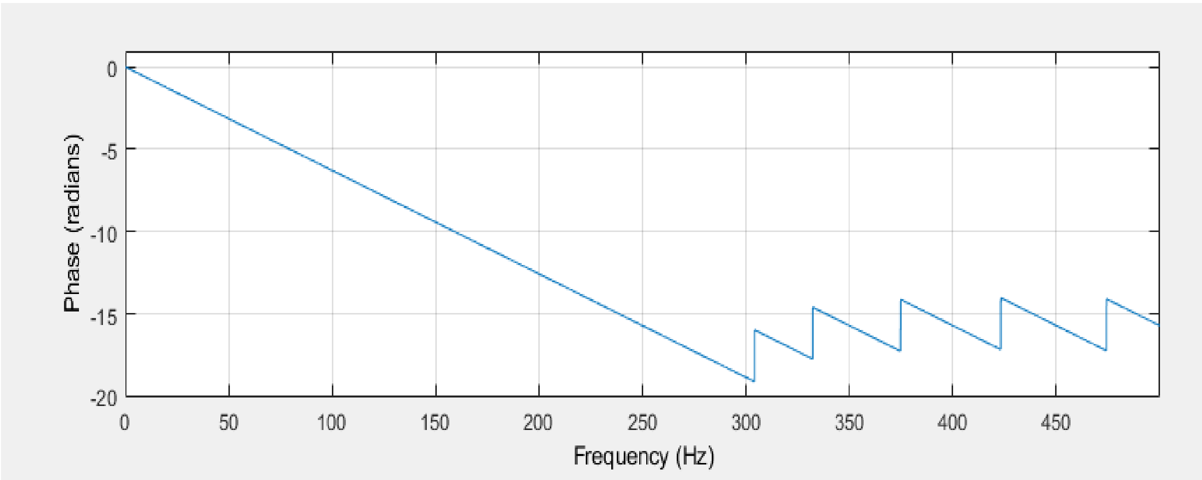


Figure 2: Phase Response of the Designed FIR Filter.

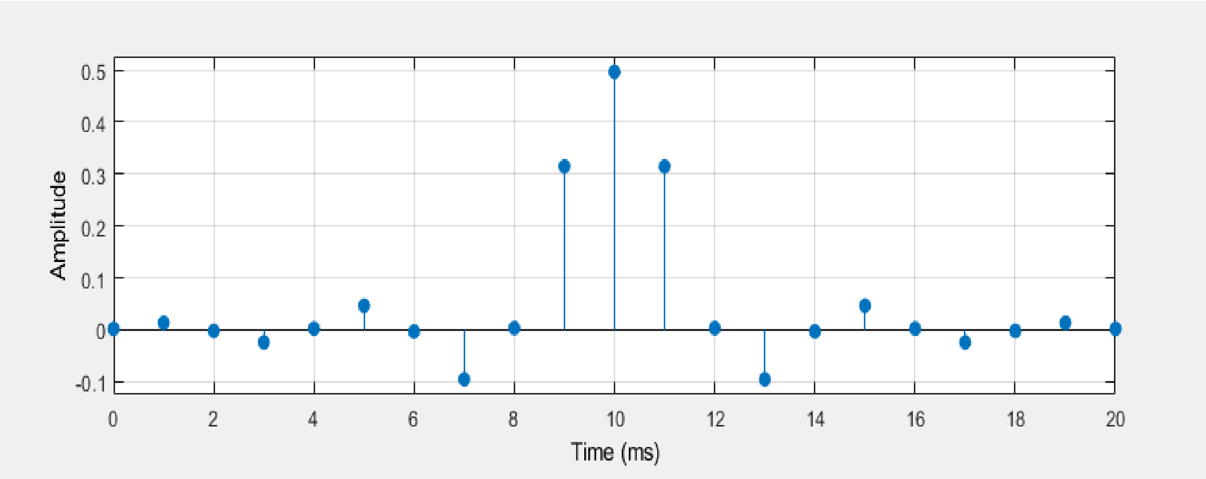


Figure 3. Impulse Response of the FIR Filter.

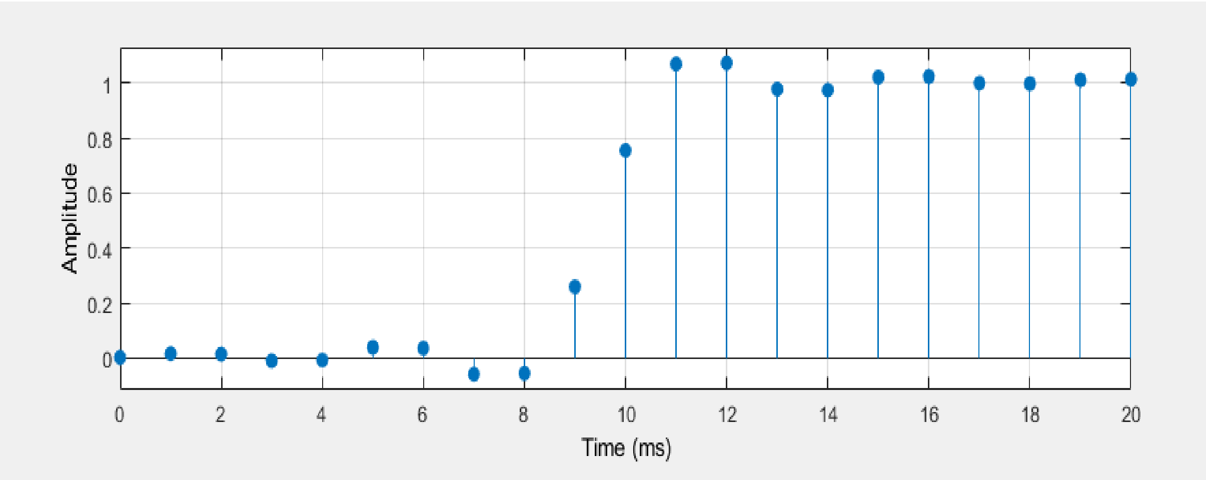


Figure 4. Step Response of the FIR Filter.

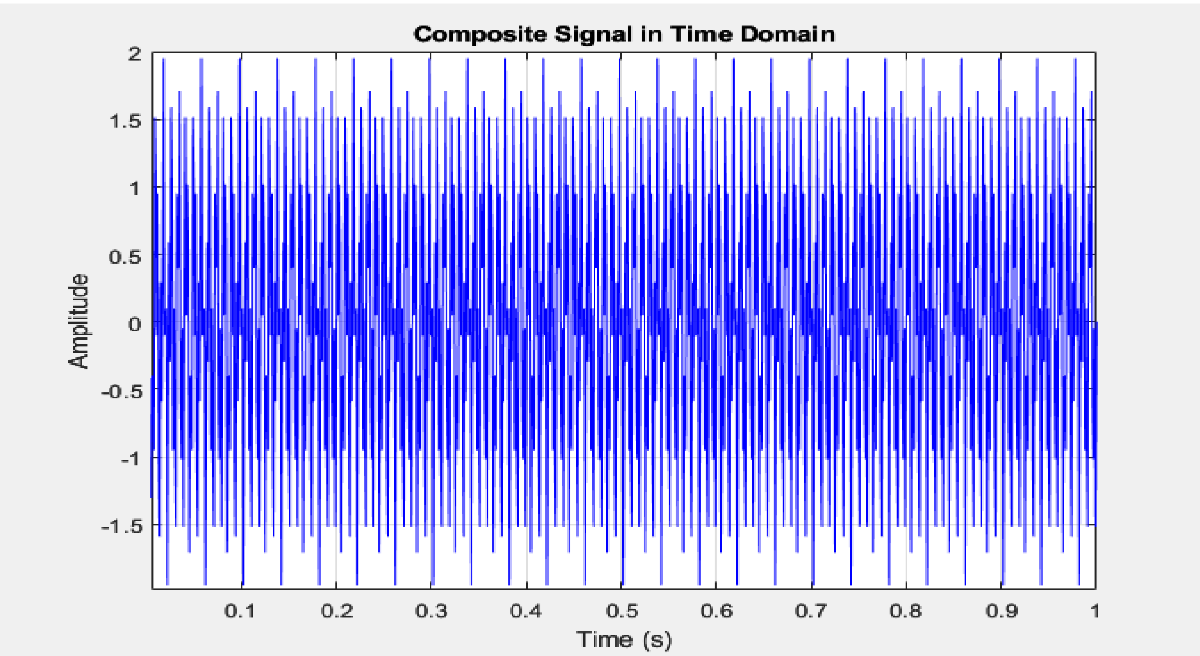


Figure 5. Time-Domain Representation of the Composite Input Signal

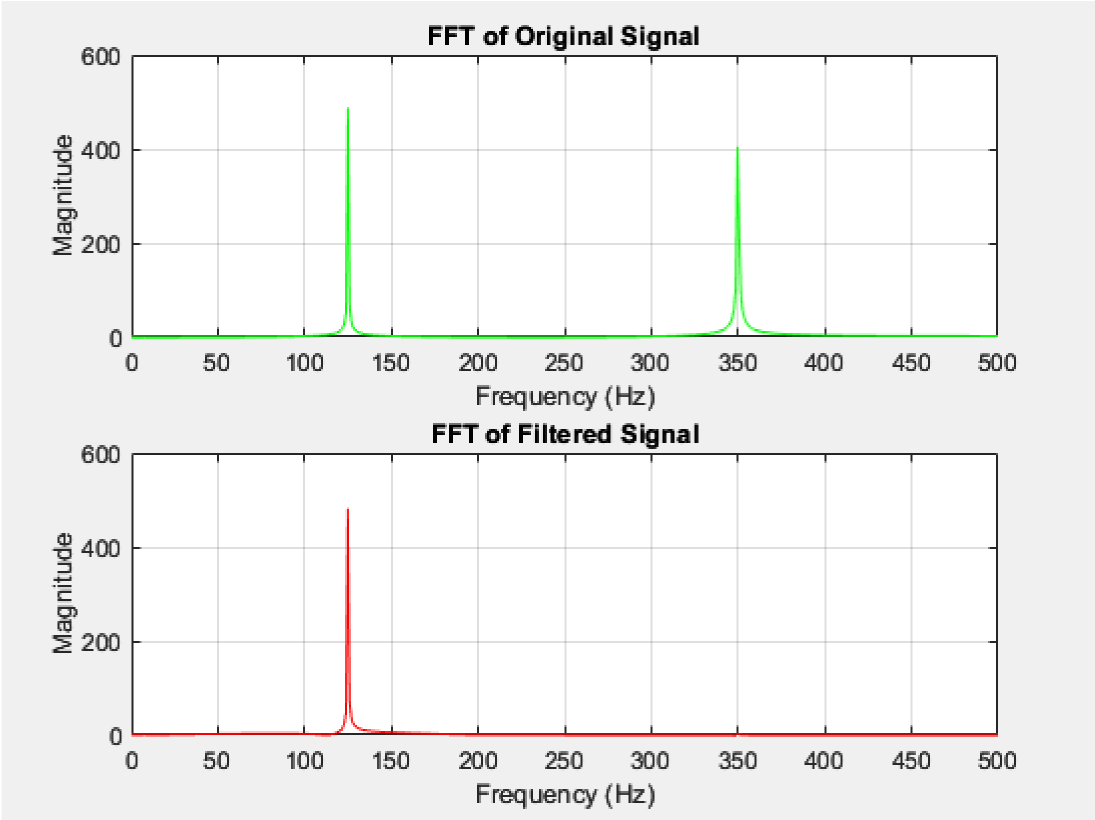


Figure 6. Frequency Domain Analysis Before and After Filtering.

Exercise:

1. Design a high-pass FIR filter using the FDA Tool in MATLAB with the following specifications:
   - Filter Type: High-Pass
   - Design Method: FIR Equiripple
   - Filter Order: 30
   - Sampling Frequency: 2000 Hz
   - Stopband Frequency (Fstop): 400 Hz
   - Passband Frequency (Fpass): 500 Hz
   - passband (Wpass) and stopband (Wstop) edges as 0.4 and 0.6.

   (a) Generate and analyze the Magnitude Response, Phase Response, Impulse response, Step response, and Pole-Zero Plot using the FDA Tool.
   (b) Export the filter coefficients to the MATLAB workspace. Assign appropriate variable names to the coefficients.
   (c) Use the '`freqz`' function in MATLAB to plot the frequency and magnitude responses of the filter. Compare these plots with those generated in the FDA Tool.
   (d) Display the Transfer Function in the MATLAB Command Window using the exported filter coefficients. Use the '`tf`' function.
   (e) Export the filter design to Simulink and implement it as a block.

2. Design a 3rd order high-pass FIR filter using the Kaiser window method. The filter should have the following specification:

   - Cutoff frequency: 1 kHz
   - Sampling frequency: 10 kHz
   - Filter order: 3
   - Stopband and Passband attenuation 40 dB
   - Passband ripple: Less than 1 dB

   (a) Export the corresponding filter coefficients to the MATLAB workspace as a (numerator) and b (denominator) using the FDA Tool.
   (b) Use the FDA Tool to visualize the pole-zero plot of the filter and determine whether the filter is stable or unstable. Provide a brief explanation of your observations.
   (c) Add a pole at z = -1.5 + 0i and a zero at z = 2 + 0i. After modifying the filter, obtain and plot both the magnitude response and phase response of the modified filter.
      (i) Export the coefficients of the modified filter to the MATLAB workspace as c (numerator) and d (denominator).
      (ii) Using the coefficients c and d, analyze the system's response to a unit impulse for the modified filters. Comment on the stability of the modified filter based on the impulse response.
   (d) Plot the magnitude responses of both the original and modified filters on the same graph using the '`freqz`' function. Ensure the graph includes axis labels for frequency and magnitude, legends to distinguish between the original and modified filters and a frequency range from 0 to 5 kHz.

   Hint: Use the same sampling frequency of 10 kHz to compute and plot the responses

3. Design a low-pass Butterworth filter with the following specifications:

   - Sampling Frequency: 8 kHz
   - Filter Order: 3

   Cutoff Frequency: 1.5 kHz
   (a) Export the MATLAB code by selecting File → Generate MATLAB Code and save it.

   Note - The generated code will define a function that creates the filter object, typically referred to as Hd.
   (b) Modify the exported MATLAB code to include a plot of the filter's frequency response using the '`fvtool`' function to visualize the cutoff frequency and analyze the overall frequency characteristics of the filter.
   (c) Plot the impulse response of the filter using the '`impz`' function and display it with a stem plot to examine the filter's behavior in the time domain.
   (d) visualize the step response of the filter using the '`stepz`' function and display it with a stem plot to further analyze its time-domain behavior.