Digital Filter Design and Signal Analysis

Bhavanisankar G S

April 5, 2025

Contents

1 Introduction

This report documents the process of designing a digital filter using MATLAB's filterDesigner tool and its application to a mixture of sine waves. The objective was to design an appropriate filter, apply it to a generated signal containing multiple frequency components, and analyze the results in both time and frequency domains.

2 Filter Design Methodology

2.1 Filter Design Specifications

For this assignment, I designed a bandpass filter with the following specifications:

- Filter Type: Bandpass
- Passband: 1000 Hz to 2000 Hz
- Filter Order: Minimum
- Density Factor: 20
- Sampling Frequency: 48,000 Hz

2.2 Filter Design Process

The filter design was carried out using MATLAB's filterDesigner tool following these steps:

1. Opened MATLAB and executed the command `filterDesigner` in the command window
2. Selected "Bandpass" as the filter type
3. Set the filter order to "Minimum"
4. Set the density factor to 20
5. Set the sampling frequency to 48,000 Hz
6. Specified the passband frequencies as 1000 Hz to 2000 Hz
7. Designed the filter and observed the frequency response

8. Exported the filter coefficients as a MATLAB file for further use

Figure 1: Screenshot of the filterDesigner tool showing the designed bandpass filter

3 Signal Generation and Analysis

3.1 Signal Generation

A test signal was created by mixing ten sine waves with different frequencies and amplitudes. The frequencies chosen were: 200 Hz, 500 Hz, 1000 Hz, 1500 Hz, 1800 Hz, 2500 Hz, 3000 Hz, 5000 Hz, 7000 Hz, and 10000 Hz. This range of frequencies allowed for a comprehensive evaluation of the filter's performance, as it included components both within and outside the designed passband.

3.2 MATLAB Implementation

Below is the MATLAB code used for signal generation, filtering, and analysis:

```matlab
% Signal Generation and Filtering
clear all;
close all;

% Parameters
fs = 48000;                % Sampling frequency (Hz)
T = 1;                     % Duration (seconds)
t = 0:1/fs:T-1/fs;         % Time vector
N = length(t);             % Number of samples

% Generate a mixture of 10 sine waves with different frequencies
f = [200, 500, 1000, 1500, 1800, 2500, 3000, 5000, 7000, 10000]; % Frequencies in Hz
amplitudes = [1, 0.8, 1.2, 0.7, 0.9, 1.1, 0.6, 0.5, 0.4, 0.3];    % Varying amplitudes

% Generate the mixed signal
mixed_signal = zeros(size(t));
for i = 1:length(f)
    mixed_signal = mixed_signal + amplitudes(i) * sin(2*pi*f(i)*t);
end

```

```
% Compute FFT of the original signal
Y = fft(mixed_signal);
P2 = abs(Y/N);
P1 = P2(1:N/2+1);
P1(2:end-1) = 2*P1(2:end-1);
freq = fs*(0:(N/2))/N;

% Plot the original signal in time domain
figure;
subplot(2,2,1);
plot(t(1:1000), mixed_signal(1:1000));
title('Original Signal (Time Domain - First 1000 samples)');
xlabel('Time (s)');
ylabel('Amplitude');

% Plot the frequency spectrum of the original signal
subplot(2,2,2);
plot(freq, P1);
title('Original Signal (Frequency Domain)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
xlim([0 12000]);
grid on;

% Load the filter coefficients
% Note: Replace 'myFilter.mat' with the name of your exported
    filter file
load('myFilter.mat');

% Apply the filter to the mixed signal
filtered_signal = filter(b, a, mixed_signal);

% Compute FFT of the filtered signal
Y_filtered = fft(filtered_signal);
P2_filtered = abs(Y_filtered/N);
P1_filtered = P2_filtered(1:N/2+1);
P1_filtered(2:end-1) = 2*P1_filtered(2:end-1);

% Plot the filtered signal in time domain
subplot(2,2,3);
plot(t(1:1000), filtered_signal(1:1000));
title('Filtered Signal (Time Domain - First 1000 samples)');
xlabel('Time (s)');
ylabel('Amplitude');

% Plot the frequency spectrum of the filtered signal
subplot(2,2,4);
plot(freq, P1_filtered);
title('Filtered Signal (Frequency Domain)');
```

```
xlabel('Frequency (Hz)');
ylabel('Magnitude');
xlim([0 12000]);
grid on;

% Compare original and filtered signals in a separate figure
figure;
plot(freq, P1, 'b', freq, P1_filtered, 'r');
title('Comparison of Original and Filtered Signals');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
xlim([0 12000]);
legend('Original Signal', 'Filtered Signal');
grid on;

% Zoom in to see the bandpass region more clearly
figure;
plot(freq, P1, 'b', freq, P1_filtered, 'r');
title('Zoomed Comparison (500-3000 Hz)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
xlim([500 3000]);
legend('Original Signal', 'Filtered Signal');
grid on;
```

Listing 1: MATLAB code for signal generation and filtering

4 Results and Analysis

4.1 Time Domain Analysis

Figure 2 shows the original and filtered signals in the time domain. As expected, the filtered signal has a different waveform compared to the original signal. This is because the filter has removed frequency components outside the 1000-2000 Hz passband, resulting in a signal that primarily consists of the frequencies within that range.

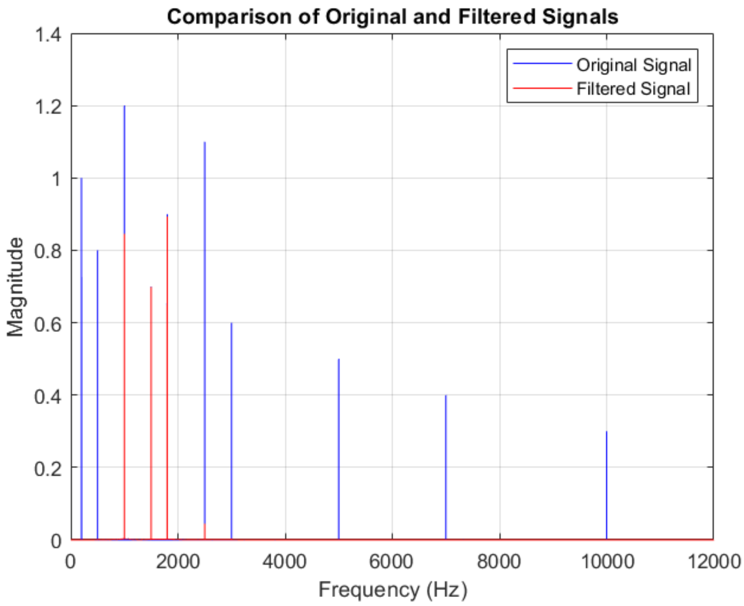

Figure 2: Time domain representation of the original and filtered signals

4.2 Frequency Domain Analysis

Figure 3 illustrates the frequency content of both the original and filtered signals. The original signal's spectrum clearly shows peaks at all ten frequencies that were used to generate the signal. In contrast, the filtered signal's spectrum shows significant attenuation of frequencies outside the 1000-2000 Hz range, with peaks only for the frequencies 1000 Hz, 1500 Hz, and 1800 Hz being preserved.

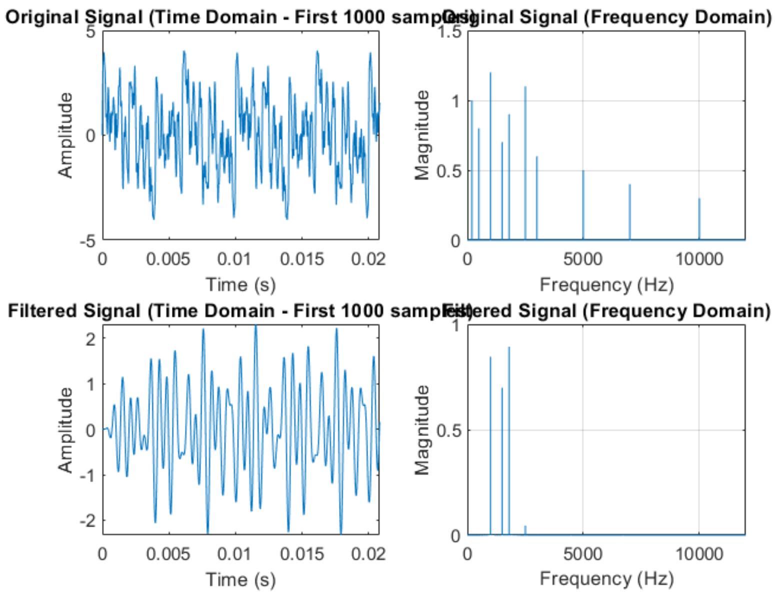

Figure 3: Frequency domain representation of the original and filtered signals

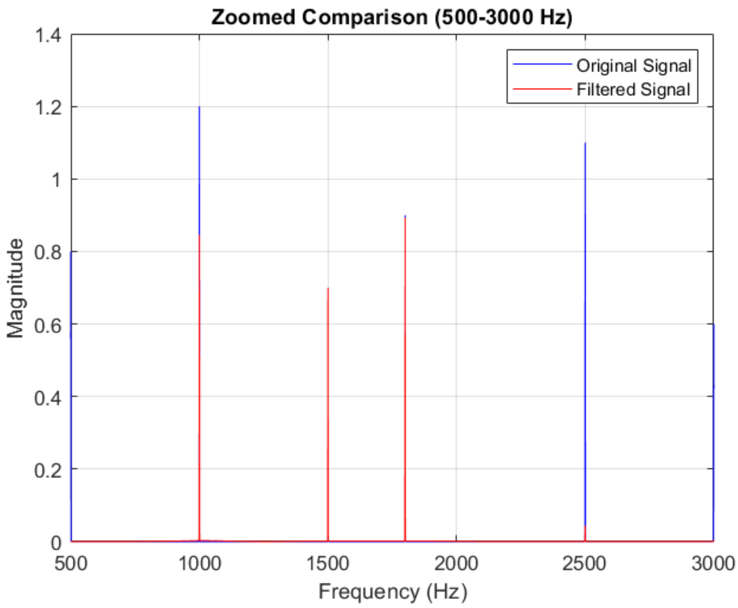

Figure 4: Zoomed view of the frequency response in the 500-3000 Hz range

5 Observations and Conclusions

Based on the results of this experiment, the following observations and conclusions can be made:

1. **Filter Performance**: The bandpass filter designed using MATLAB's filterDesigner tool effectively attenuated frequencies outside the 1000-2000 Hz range while preserving frequencies within this range.

2. **Passband Analysis**: The three frequency components at 1000 Hz, 1500 Hz, and 1800 Hz were well preserved in the filtered signal, as they fall within the filter's passband.

3. **Stopband Analysis**: Frequencies below 1000 Hz (200 Hz and 500 Hz) and above 2000 Hz (2500 Hz, 3000 Hz, 5000 Hz, 7000 Hz, and 10000 Hz) were significantly attenuated, demonstrating the filter's ability to reject frequencies outside the desired range.

4. **Time Domain Effects**: The filtering process altered the waveform of the signal in the time domain, as expected. The filtered signal appears to have a more regular pattern, which is consistent with the removal of multiple frequency components.

5. **Filter Design Trade-offs**: While designing the filter, I noticed that increasing the filter order would provide steeper roll-offs at the band edges, but at the cost of increased computational complexity. For this application, a minimum order filter provided adequate performance.

6 Appendix

6.1 Filter Specification Details

Parameter	**Value**
Filter Type	Bandpass
Passband	1000-2000 Hz
Filter Order	Minimum
Density Factor	20
Sampling Frequency	48000 Hz

Table 1: Detailed filter specifications