Name: SalahDin Ahmed Salh Rezk
ID: 202201079
Date: April 23, 2024

Signals and Systems (CIE 227)

Assignment 3

Contents

List of Figures

1 Introduction to FIR Filters

1.1 Concept

An FIR filter is a type of digital filter used in signal processing. It operates on a discrete-time signal i.e. it processes data that is sampled at specific intervals. The core idea behind FIR filters is to perform convolution between the input signal and a set of filter coefficients (also known as taps). These coefficients determine how the filter processes the input signal to achieve desired filtering characteristics.

1.2 Characteristics

1. **Finite Impulse Response:** The term "finite" indicates that the filter's output response to an impulse (a single sample of unity magnitude followed by zeros) is of finite duration. This makes FIR filters inherently stable and easy to implement in digital systems.

2. **Linear Phase Response:** FIR filters can achieve linear phase response, meaning they introduce constant delay across all frequencies. This characteristic is desirable in applications where phase distortion must be minimized, such as in audio processing.

3. **Finite Impulse Response:** As the name suggests, FIR filters have a finite impulse response, which means that their output response to an impulse input eventually decays to zero.

4. **No Feedback:** Unlike IIR (Infinite Impulse Response) filters, FIR filters do not have feedback loops in their structure. This absence of feedback simplifies their implementation and eliminates stability concerns related to feedback loops.

1.3 Applications

1. **Digital Signal Processing:** FIR filters are widely used in digital signal processing applications such as audio processing, image processing, communication systems, biomedical signal processing, and more.

2. **Audio Processing:** FIR filters are used in audio equalization, noise reduction, and echo cancellation. [1]

3. **Image Processing:** FIR filters are used in image enhancement, edge detection, and image restoration. [2]

4. **Communication Systems:** FIR filters are used in channel equalization, pulse shaping, and digital modulation. [3]

5. **Biomedical Signal Processing:** FIR filters are used in ECG signal processing, EEG signal analysis, and medical imaging. [4]

1.4 Structure

The basic structure of an FIR filter consists of input samples, filter coefficients (taps), and a summing junction. It operates by convolving the input signal with the filter coefficients to produce the filtered output signal.

1. **Input Signal ($x[n]$):** The input to an FIR filter is a sequence of discrete-time samples, where n represents the time index.

2. **Filter Coefficients ($h[n]$):** The filter coefficients are the weights applied to the input samples during convolution. These coefficients determine the filter's frequency response and filtering characteristics.

3. **Convolution Operation ($*$):** The convolution operation involves multiplying the input samples by the filter coefficients and summing the results to produce the output signal.

4. **Summing Junction ($\sum$):** The summing junction weighted input samples to generate the output signal.

5. **Output Signal ($y[n]$):** The output of an FIR filter is the result of convolving the input signal with the filter coefficients. It represents the filtered version of the input signal.

$$y[n] = \sum_{k=0}^{N-1} h[k] * x[n-k] \tag{1}$$

The sum runs over the range of filter coefficients from $k = 0$ to $N - 1$, where N is the number of taps or coefficients in the FIR filter.

The filter coefficients $h[k]$ are typically designed based on the desired frequency response of the filter, such as low-pass, high-pass, band-pass, or notch filters. The order of the FIR filter (determined by N) affects its frequency selectivity and performance characteristics.

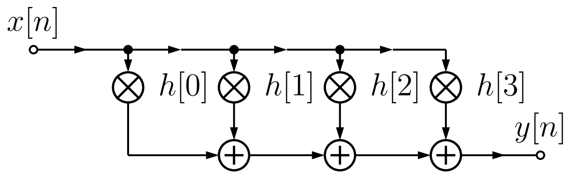

Figure 1: Basic Structure of an FIR Filter

2 Designing a Low-pass FIR Filter

2.1 Concept

A low-pass FIR filter is designed to pass signals with frequencies lower than a specified cutoff frequency while attenuating signals with higher frequencies. This type of filter is commonly used in applications where high-frequency noise or interference must be removed while preserving the low-frequency components of the signal.

2.2 Design Steps:

1. **Determine Specifications:** Specify the desired cutoff frequency, passband ripple, stopband attenuation, and filter order.

2. **Design Filter Coefficients:** Use windowing, frequency sampling, or optimization techniques to design the filter coefficients based on the desired frequency response.

3. **Implement Filter:** Implement the FIR filter using the designed coefficients and convolve it with the input signal to obtain the filtered output.

2.3 MATLAB Code

```
Fs = 8000;        % Sampling frequency in Hz
Fc = 1000;        % Cutoff frequency in Hz
N = 50;           % Filter order
ftype = "low";    % Filter type

Fc_norm = Fc / (Fs/2);         % Normalize the cutoff frequency
h = fir1(N, Fc_norm, ftype);   % Design the filter

freqz(h, 1, 1024, Fs); % Plot the frequency response of the filter
```

In this code snippet, the `fir1` function is used to design an FIR filter with a low-pass response. The filter order `N`, cutoff frequency `Fc`, and sampling frequency `Fs` are specified to design the filter coefficients `h`. The frequency response of the filter is then plotted using the `freqz` function.

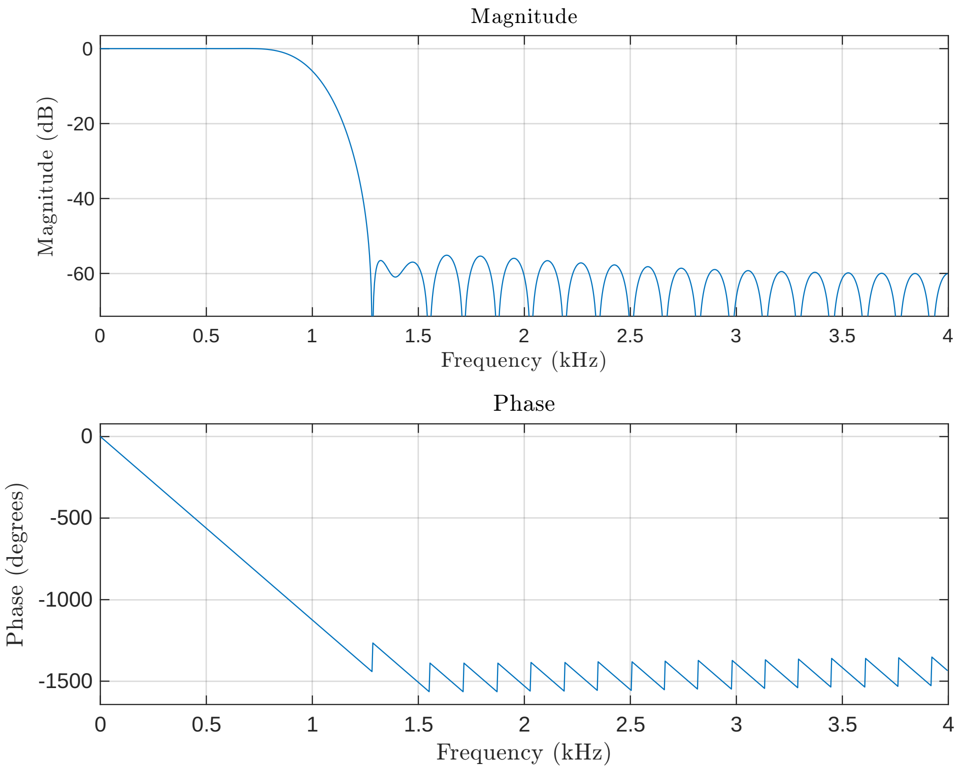

Figure 2: Frequency Response of FIR Low-Pass Filter

2.4 Frequency Response:

The frequency response of a low-pass FIR filter shows the attenuation of high-frequency components beyond the cutoff frequency. The passband region contains the low-frequency components that are passed by the filter, while the stopband region shows the attenuation of high-frequency components.

3 Designing a High-pass FIR Filter

3.1 Concept

A high-pass FIR filter is designed to pass signals with frequencies higher than a specified cutoff frequency while attenuating signals with lower frequencies. This type of filter is commonly used in applications where low-frequency noise or interference must be removed while preserving the high-frequency components of the signal.

3.2 Design Steps

1. **Determine Specifications:** Specify the desired cutoff frequency, passband ripple, stopband attenuation, and filter order.

2. **Design Filter Coefficients:** Use windowing, frequency sampling, or optimization techniques to design the filter coefficients based on the desired frequency response.

3. **Implement Filter:** Implement the FIR filter using the designed coefficients and convolve it with the input signal to obtain the filtered output.

3.3 MATLAB Code

```
Fs = 8000;       % Sampling frequency in Hz
Fp = 1000;       % Passband frequency in Hz
N = 50;          % Filter order
ftype = "high"; % Filter type

Fp_norm = Fp / (Fs/2);        % Normalize the passband frequency
h = fir1(N, Fp_norm, ftype);  % Design the filter

freqz(h, 1, 1024, Fs);  % Plot the frequency response of the filter
```

3.4 Frequency Response

The frequency response of a high-pass FIR filter shows the attenuation of low-frequency components below the cutoff frequency. The passband region contains the high-frequency components that are passed by the filter, while the stopband region shows the attenuation of low-frequency components.

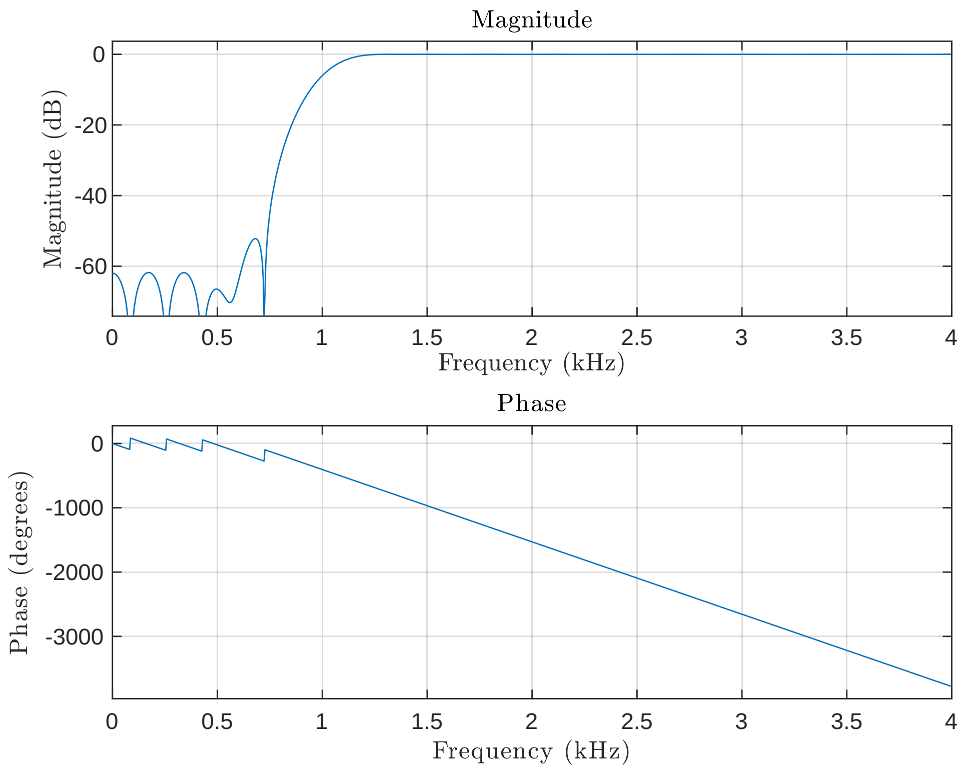

Figure 3: Frequency Response of FIR High-Pass Filter

4 Implementation and Analysis

4.1 Low-Pass FIR Filter on a Sinusoidal Signal with Noise

```
% Design a low-pass filter with a cutoff frequency of 1000 Hz
% Apply the filter to a sinusoidal signals with added noise
% Plot the original signal and the filtered signal
Fs = 48000;      % Sampling frequency in Hz
Fc = 1000;       % Cutoff frequency in Hz
N = 100;         % Filter order
ftype = "low";   % Filter type

Fc_norm = Fc / (Fs/2);         % Normalize the cutoff frequency
h = fir1(N, Fc_norm, ftype);   % Design the filter

% Generate a sinusoidal signal with added noise
t = 0:1/Fs:1;
x = sin(2*pi*1000*t) + 0.5*randn(size(t));

% Apply the filter to the signal
y = filter(h, 1, x);

% Plot the original signal and the filtered signal
figure;
subplot(2,1,1);
plot(t, x);
```

```
title('Original Signal');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

subplot(2,1,2);
plot(t, y);
title('Filtered Signal');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

zoom xon;
zoom(100);
```

In this code snippet, the cutoff frequency is set to 1000 Hz so that it matches our desired frequency of the sinusoidal signal $\sin(2\pi \cdot 1\text{k} \cdot t)$. The filter is then applied to the sinusoidal signal with added noise to demonstrate the filtering effect.

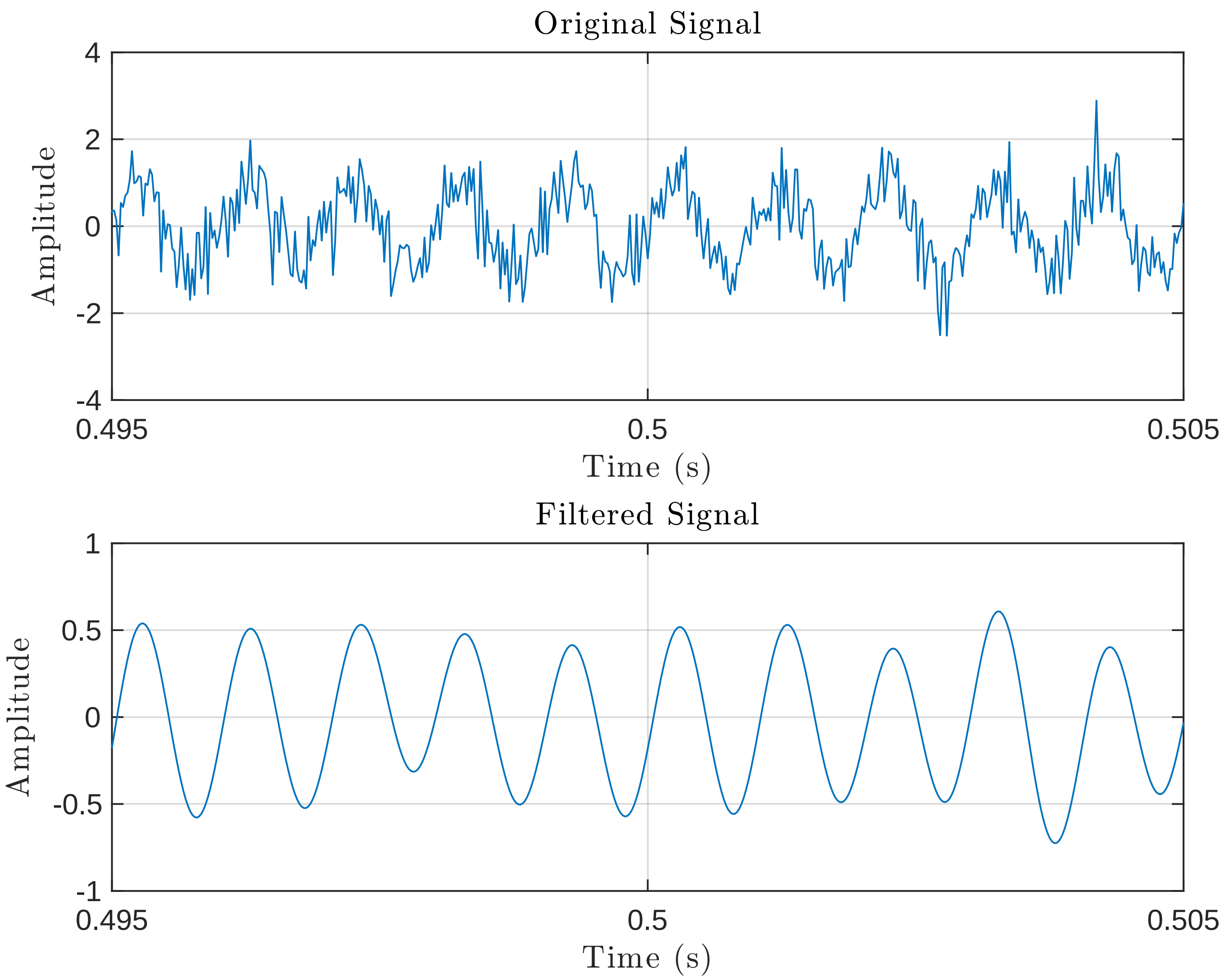

Figure 4: Low-Pass FIR Filter Applied to a Sinusoidal Signal with Noise

4.2 High-Pass FIR Filter on a Composite Sinusoidal Signal

```
% Design a high-pass filter with a passband frequency of 1500 Hz
% Apply the filter to a composite sinusoidal signal
% Plot the original signal and the filtered signal
Fs = 48000;     % Sampling frequency in Hz
Fp = 1500;      % Passband frequency in Hz
N = 100;        % Filter order
ftype = "high"; % Filter type

Fp_norm = Fp / (Fs/2);        % Normalize the passband frequency
h = fir1(N, Fp_norm, ftype);  % Design the filter

% Generate a composite sinusoidal signal
t = 0:1/Fs:1;
x = sin(2*pi*1000*t) + sin(2*pi*2000*t);

% Apply the filter to the signal
y = filter(h, 1, x);

% Plot the original signal and the filtered signal
figure;
subplot(2,1,1);
plot(t, x);
title('Original Signal');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

subplot(2,1,2);
plot(t, y);
title('Filtered Signal');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;

zoom xon;
zoom(100);
```

In this code snippet, the passband frequency is set to 1500 Hz so that the composite sinusoidal wave of $\sin(2\pi \cdot 1\text{k} \cdot t) + \sin(2\pi \cdot 2\text{k} \cdot t)$ get filtered into the higher frequency component of 2 kHz.

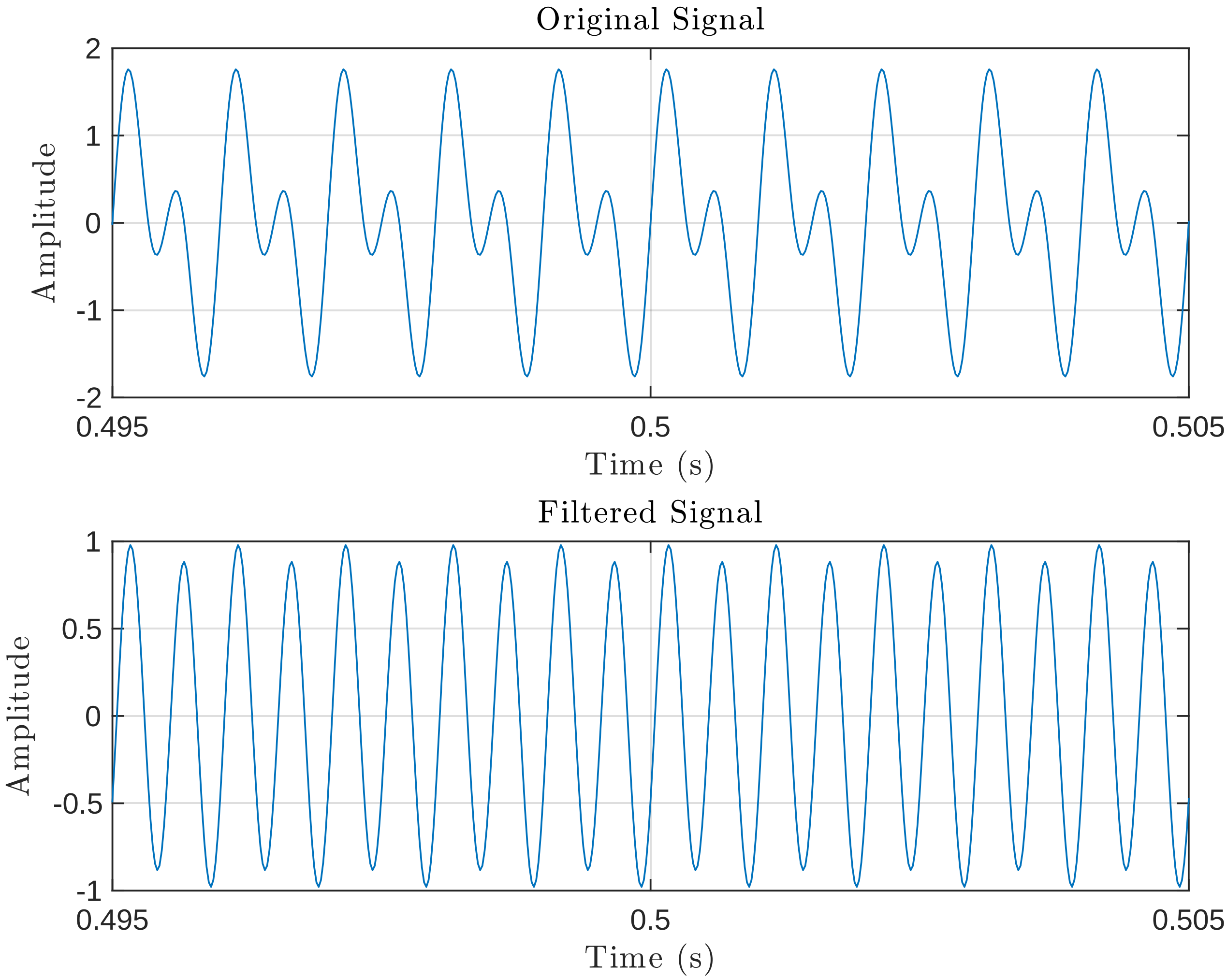

Figure 5: High-Pass FIR Filter Applied to a Composite Sinusoidal Signal

5 Conceptual Understanding

5.1 Low-pass Filter

For a simple first-order low-pass filter

$$H(s) = \frac{1}{1 + s/\omega_c}, \tag{2}$$

where s is the complex frequency variable, and ω_c is the cutoff frequency. The transfer function $H(s)$ describes the relationship between the input and output signals of the filter.

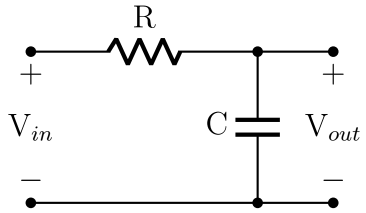

Figure 6: RC Low-Pass Filter

The input signal V_{in} is applied across the series combination of the resistor R and the capacitor C. Since the resistor R and the capacitor C are in series, they form a voltage divider network. The voltage across the capacitor (V_{out}) is the output of the low-pass filter.

At low frequencies, the capacitive reactance X_c of the capacitor is high, meaning it acts like an open circuit and allows most of the input signal to pass through to the output V_{out}. The impedance Z of a capacitor is given by the formula $Z = \frac{1}{j\omega C}$, where is the imaginary unit,   is the angular frequency $2\pi f$, and C is the capacitance.

As the frequency of the input signal increases, the capacitive reactance (X_c) decreases inversely proportional to frequency $Xc = \frac{1}{2fC}$. When the frequency reaches a certain point known as the cutoff frequency f_C, the capacitive reactance becomes comparable to the resistance in the circuit.

At frequencies below the cutoff frequency ($f < f_C$), the capacitor effectively blocks the higher-frequency components of the input signal, allowing primarily the lower-frequency components to pass through to the output. This filtering action is due to the high impedance of the capacitor at lower frequencies.

As the frequency of the input signal exceeds the cutoff frequency ($f > f_C$), the capacitive reactance (X_c) decreases, and more signal is diverted through the capacitor to ground rather than passing through to the output. This results in attenuation of higher frequencies, effectively filtering them out from the output.

5.2 High-pass Filter:

For a simple first-order high-pass filter

$$H(s) = \frac{s}{s + \omega_c}, \tag{3}$$

The input signal V_{in} is applied across the series combination of the capacitor C and the resistor R. Since the capacitor C and the resistor R are in series, they form a voltage divider network. The voltage across the resistor (V_{out}) is the output of the high-pass filter.

At low frequencies, the capacitive reactance X_c of the capacitor is high, meaning it acts like an open circuit and allows most of the input signal to pass through to the output V_{out}. The

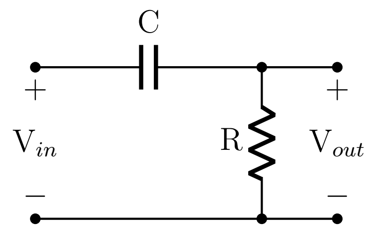

Figure 7: RC High-Pass Filter

impedance Z of a capacitor is given by the formula $Z = \frac{1}{j\omega C}$, where j is the imaginary unit, is the angular frequency $2\pi f$, and C is the capacitance.

As the frequency of the input signal increases, the capacitive reactance (X_c) decreases inversely proportional to frequency $X_c = \frac{1}{2\pi f C}$. When the frequency reaches a certain point known as the cutoff frequency f_C, the capacitive reactance becomes comparable to the resistance in the circuit.

At frequencies below the cutoff frequency ($f < f_C$), the capacitor effectively blocks the lower-frequency components of the input signal, allowing primarily the higher-frequency components to pass through to the output. This filtering action is due to the high impedance of the capacitor at lower frequencies.

As the frequency of the input signal exceeds the cutoff frequency ($f > f_C$), the capacitive reactance (X_c) decreases, and more signal is diverted through the capacitor to ground rather than passing through to the output. This results in attenuation of lower frequencies, effectively filtering them out from the output.

5.3 Trade-offs in Filter Design

1. **Filter Order:** The order of a filter refers to the number of reactive components (capacitors or inductors) in the filter design. Higher-order filters have steeper roll-off characteristics, meaning they can achieve sharper transitions between passbands and stopbands. However, increasing the filter order also increases complexity and introduces more components, leading to higher cost and potential signal distortion.

2. **Frequency Selectivity:** Frequency selectivity refers to how well a filter can discriminate between desired signal frequencies and unwanted frequencies. Higher-order filters generally offer better frequency selectivity by providing narrower passbands and deeper stopbands. However, this increased selectivity often comes at the expense of wider transition bands and higher latency.

3. **Passband Ripple:** Passband ripple refers to the variation in gain within the passband of a filter. Lower passband ripple indicates a more consistent gain across the passband, resulting in less distortion of the desired signal. However, achieving lower passband ripple may require more complex filter designs or higher-order filters.

4. **Attenuation:** Attenuation denotes how effectively a filter suppresses frequencies outside its passband. Higher attenuation levels are crucial for applications where noise or interference must be minimized. However, achieving higher attenuation may require sacrificing passband flatness or introducing more components.

5. **Phase Response:** Filters can introduce phase shifts to the signals passing through them, affecting the timing relationships within the signal. Linear phase filters maintain constant group delay across all frequencies, preserving signal integrity but often at the cost of increased complexity.

6. **Group Delay:** Group delay measures the time delay experienced by different frequency components of a signal. For real-time applications, minimizing group delay is essential to prevent signal distortion and maintain fidelity.

6 Nosiy ECG Signal (Bonus)

Used data from the PhysioNet MIT-BIH Noise Stress Test Database [5], the ECG signal was recorded at a sampling frequency of 360 Hz. The signal was corrupted with a signal-to-noise ratio (SNR) of 12 dB. The cutoff and passband frequencies were set to 5 Hz and 20 Hz, respectively, with a filter order of 1001 taps [4].

```
Fs = 360; % Sampling frequency in Hz
Fp = 5; % Passband frequency in Hz
Fc = 20; % Cutoff frequency in Hz
N = 1001;    % Filter order

Fp_norm = Fp / (Fs/2);        % Normalize the passband frequency
Fc_norm = Fc / (Fs/2);        % Normalize the cutoff frequency
h = fir1(N, [Fp_norm Fc_norm]); % Generate the filter coefficients

% Read the ECG signal
% The ECG signal is stored in a CSV file with one column
% The sampling frequency is 360 Hz
ecg = csvread('118e12.csv');

% Apply the filter to the ECG signal
filtered_ecg = filter(h, 1, ecg);

% Plot the original and filtered ECG signals
figure;
subplot(2,1,1);
plot(ecg);
title('Original ECG Signal');
xlim([1000 2000]);
ylim([-8 -4]);
grid on;

subplot(2,1,2);
plot(filtered_ecg);
title('Filtered ECG Signal');
xlim([1000 2000]);
ylim([-1 1]);
grid on;
```

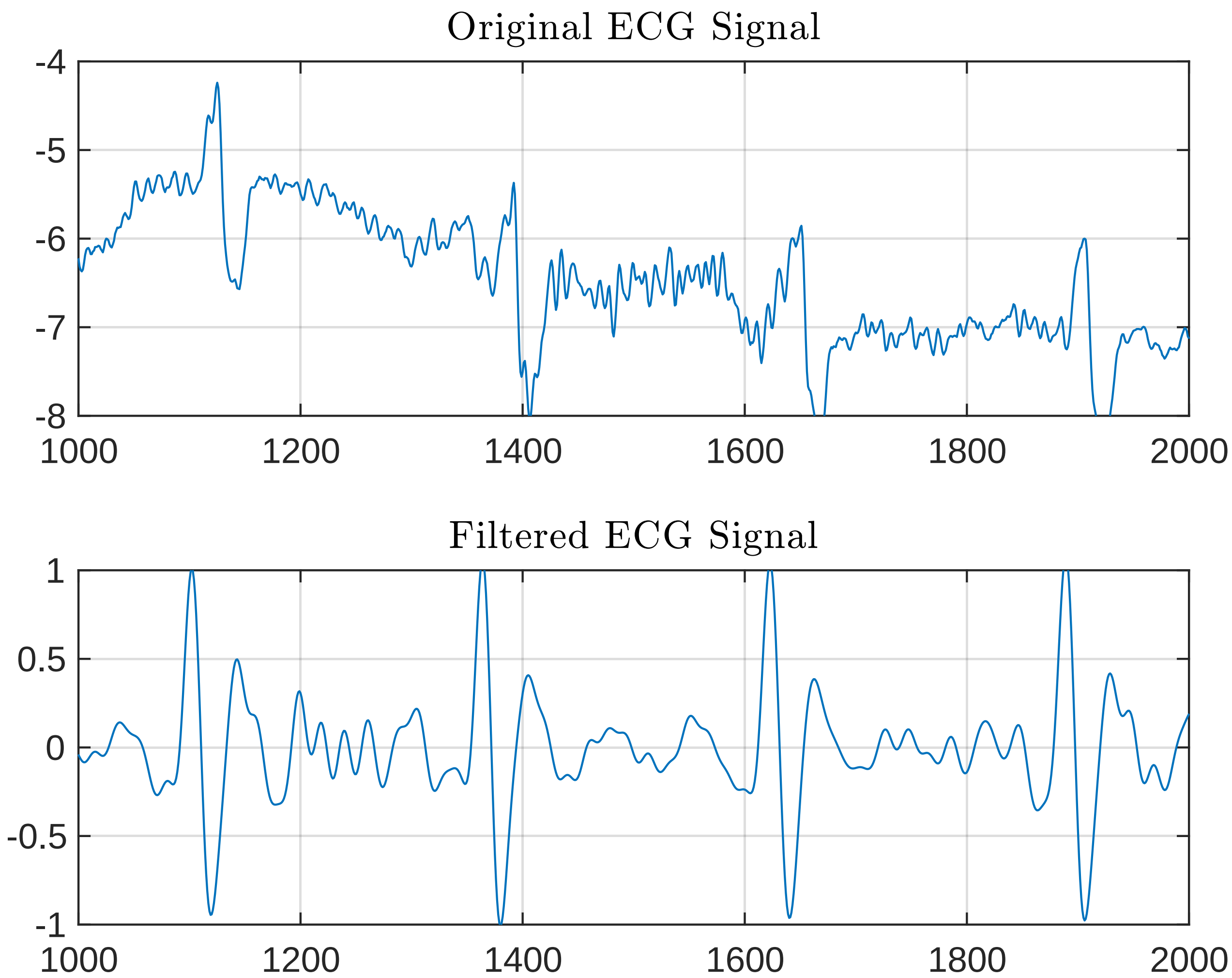

Figure 8: ECG Signals of Heartbeat

7 References

[1] R. Pawar and D. R. Mehra, "Design and performance analysis of fir filter for audio applications," *International Journal of Scientific Research and Technology*, pp. 122–126, Aug. 2014.

[2] U. of Michigan. "Lab6 manual." (2002), [Online]. Available: `https://www.eecs.umich.edu/courses/eecs206/archive/spring02/lab.dir/Lab6/lab6_v3_0_release.pdf`.

[3] K. Hossain, R. Ahmed, A. Haque, and M. Rahman, "A review of digital fir filter design in communication systems," *International Journal of Science and Research (IJSR)*, vol. 10, p. 09, Mar. 2021. DOI: `10.21275/SR21221133619`.

[4] N. Marchon, "Efficient fir filters for biomedical signals," pp. 1947–1951, 2019. DOI: `10.1109/TENCON.2019.8929397`.

[5] A. Goldberger, L. Amaral, L. Glass, *et al.* "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals." (2000), [Online]. Available: `https://physionet.org/content/physiobank/1.0.0/`.