****

**UNIVERSITI TUNKU ABDUL RAHMAN**

**UCCE2073 : Introduction to Digital Signal Processing**

**Session 2025-02**

**Assignment – FIR Filter Design**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | *Remark (If applicable)* |
| Cheah Lik Ding | 2206416 |  |
| Chong Bing Hong | 2305444 |  |
| Jayson Chen Min Xiang | 2305373 |  |
| Shu Mee Ang | 2303988 |  |

\* Highlight the assigned signal row and circle the assigned passband frequency.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal** | **(Hz)** | **(Hz)** | **(Hz)** | **(Hz)** |
| Signal A | 500 | 1000 | 1500 | 2000 |
| Signal B | 600 | 1200 | 1800 | 2400 |
| Signal C | 700 | 1400 | 2100 | 2800 |

-------------- The following section is to be completed by the lecturer --------------

|  |  |
| --- | --- |
| **Assessment Details** | **Awarded Marks** |
| Time Graphs (Pre-filtering) (1%) |  |
| Time Graphs (Post-filtering) (2%) |  |
| Frequency Graph (Pre-filtering) (2%) |  |
| Frequency Graph (Post-filtering) (4%) |  |
| Description of Assignment (1%) |  |
| Description of Filter Design (3%) |  |
| Comparison of Windowing Techniques (3%) |  |
| Source Codes and Documentation (2%) |  |
| Overall Report Structure (2%) |  |
| *\* Penalty (If applicable)* |  |
| **Total** **(20%)** |  |

**Introduction**

This assignment involves designing a Finite Impulse Response (FIR) filter for Signal C, which is a sum of four equal-amplitude frequency components at 700 Hz, 1400 Hz, 2100 Hz, and 2800 Hz. The objective is to design a bandpass filter that passes only the 2100 Hz component while suppressing the other frequencies (700 Hz, 1400 Hz, and 2800 Hz). The filters are designed using Hanning and Blackman windowing techniques with window lengths N = 200, 400, and 800. MATLAB is used to implement the filter design, apply the filters, and visualize the results in time and frequency domains.

**Methods and Parameters Used in FIR Filter Design**

1. Pre-Filtering Signal Generation and Spectral Analysis

* Sampling and Time Vector

Sampling frequency: Fs = 48000 Hz

Duration: t = 0:1/Fs:1 s

= 1/48000 s

* Signal Composition (Signal C)

Component frequencies: 700, 1400, 2100, 2800 Hz

freqs1 = sin(2\*pi\*700\*t);  
freqs2 = sin(2\*pi\*1400\*t);  
freqs3 = sin(2\*pi\*2100\*t);  
freqs4 = sin(2\*pi\*2800\*t);

Composite signal: signal = sum of all freqs

* Initial Frequency-Domain Analysis

FFT length: L = 48000

Frequency axis: f = Fs \* (0:(L/2)) / L, range from 0–24,000 Hz

One-sided spectrum computed via FFT and magnitude scaling as in MATLAB

Plot range: 0–3000 Hz with X-ticks at 700,1400,2100 and 2800Hz

2. FIR Filter Design

* Filter Specifications

Center frequency: fc = 2100 Hz

Half-bandwidth: b = 50 Hz

Passband edges: f\_low = 2050 Hz, f\_high = 2150 Hz

Normalized edges: WnL = f\_low/(Fs/2) ≈ 0.0854, WnH = f\_low/(Fs/2) ≈ 0.0896

* Filter Orders and Window Functions

Filter lengths tested: N = {200, 400, 800} (order = N-1)

Windows: Hanning (hann(N)), Blackman (blackman(N))

* Filter Coefficient Computation

b = fir1(N-1, [WnL, WnH], “bandpass”, wdw)

wdw = hann(N) or blackman(N)

Apply filter: filtered\_signal = filter(b, 1, signal)

3. Post-Filter Analysis

* Time-Domain Inspection

Stable oscillation offset: T = 1/700 s, plot window [10T, 12T]

Examine two periods of the filtered waveform in steady state

3.2 Frequency-Domain Inspection of Filtered Outputs

FFT length = 48,000

Compute one-sided spectrum and Plot range: 0–3000 Hz with X-ticks at 700,1400,2100 and 2800Hz

4. Plotting Parameters

* Time-domain plots per sinusoid:

use subplot(5,1,i) for i=1 to 4, plus the composite on row 5.

* Frequency-domain plots:

plot(f, 20\*log10(P1));

Grid, axis labels, and appropriate titles.

* Consistent styling:

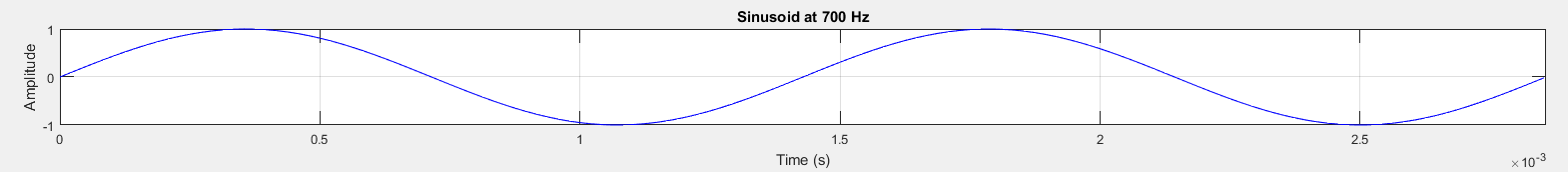
xlabel('Time (s)'), ylabel('Amplitude') for time plots;

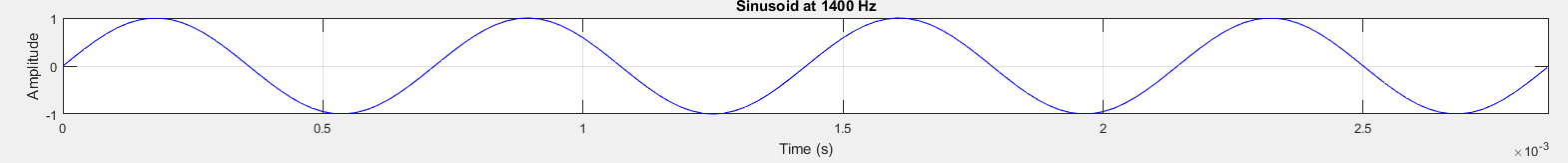
xlabel('Frequency (Hz)'), ylabel('Magnitude (dB)') for spectra.

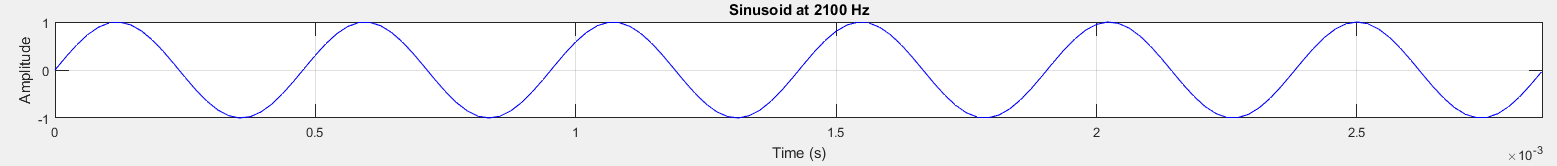
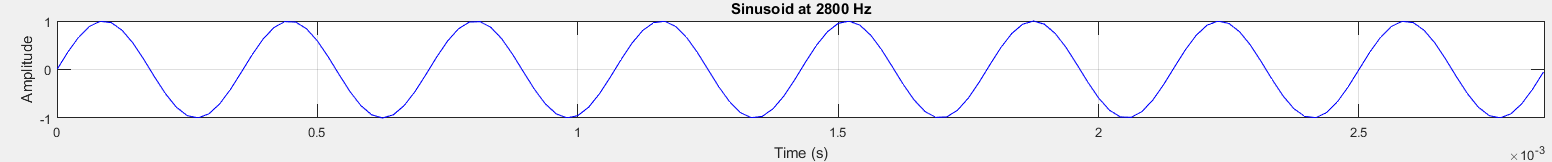
**Graph Plots for Signal C Before Filtering**

Time-domain x-axis tick labels are normalized to multiples of the period of the lowest frequency (1/700 Hz) for readability.

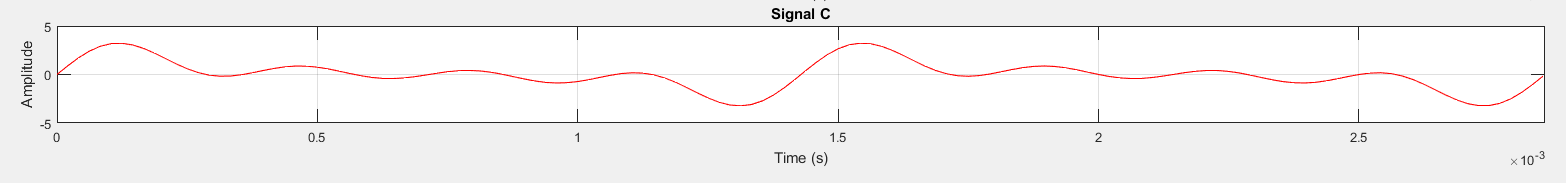
**Unfiltered Signal C Frequency Components**

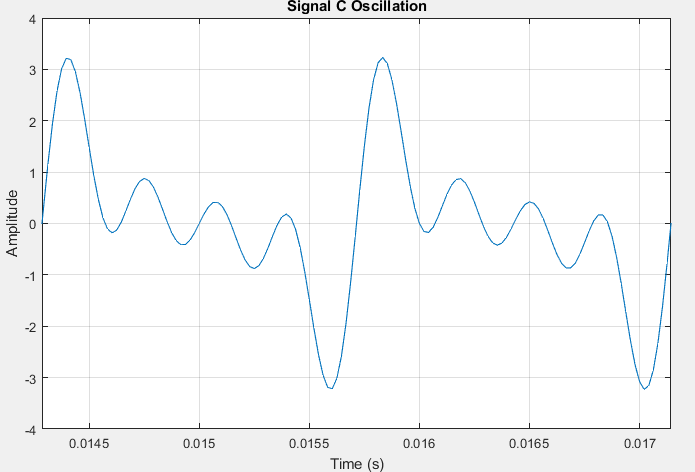
Figure 1: 700 Hz sinusoid time-domain graph.

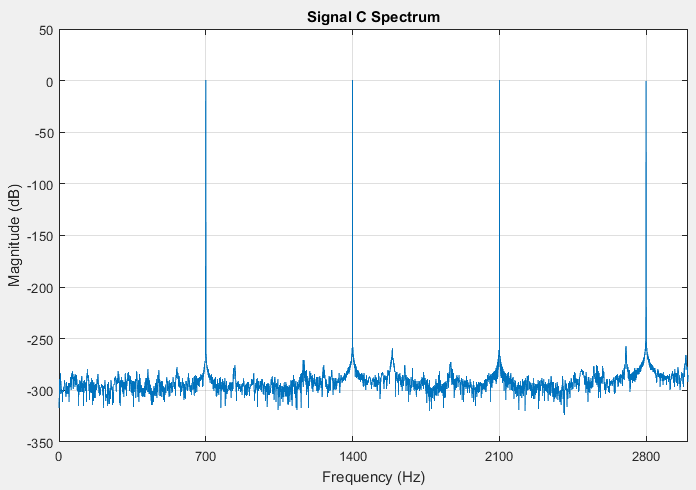
Figure 2: 1400 Hz sinusoid time-domain graph.

Figure 3: 2100 Hz sinusoid time-domain graph.  
Figure 4: 2800 Hz sinusoid time-domain graph.

**Unfiltered Signal C**

Figure 5: Unfiltered Signal C (sum of four components).

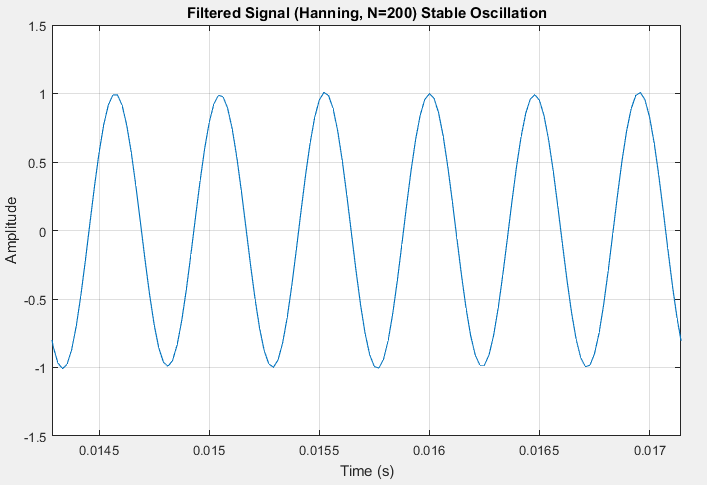
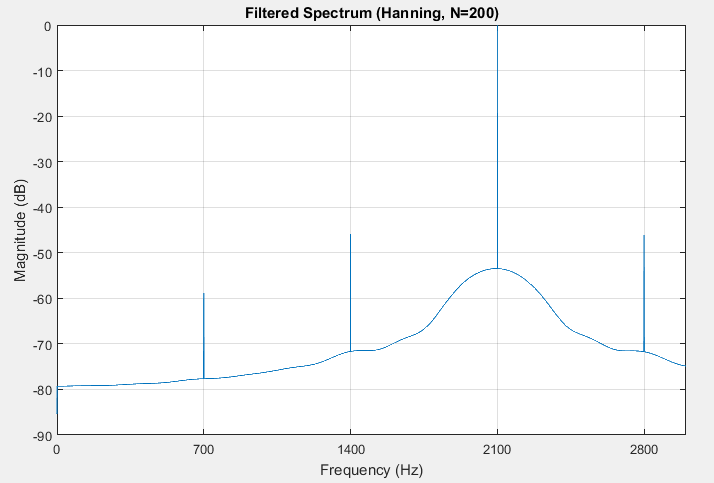
Figure 6: Signal C Oscillation graph.

Figure 7: Unfiltered Signal C frequency-domain graph.

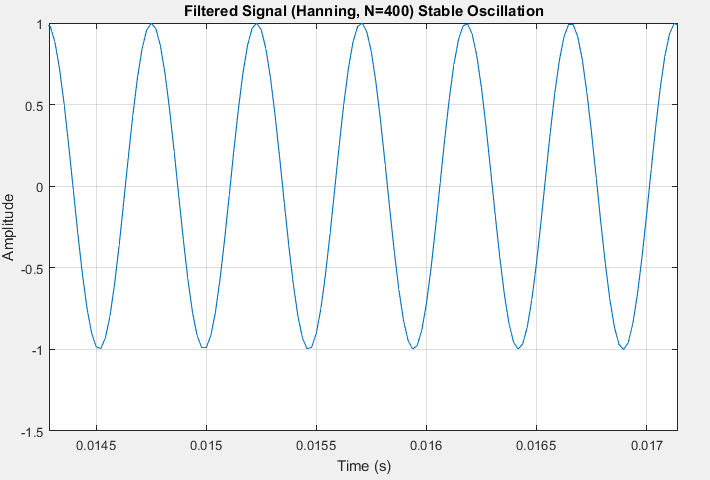
**Graph Plots for Signal C After Filtering**

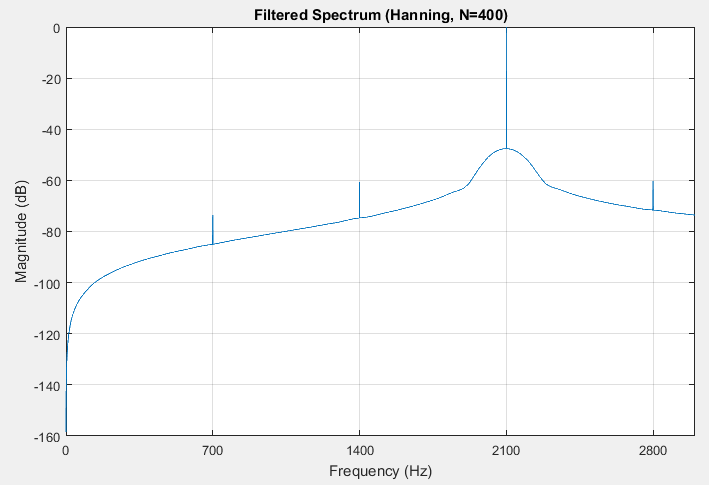
Plots are provided for Hanning and Blackman windows with N = 200, 400, and 800, showing time-domain during stable oscillation and frequency-domain results.

**Filtered Signal (Hanning Window; N = 200)**

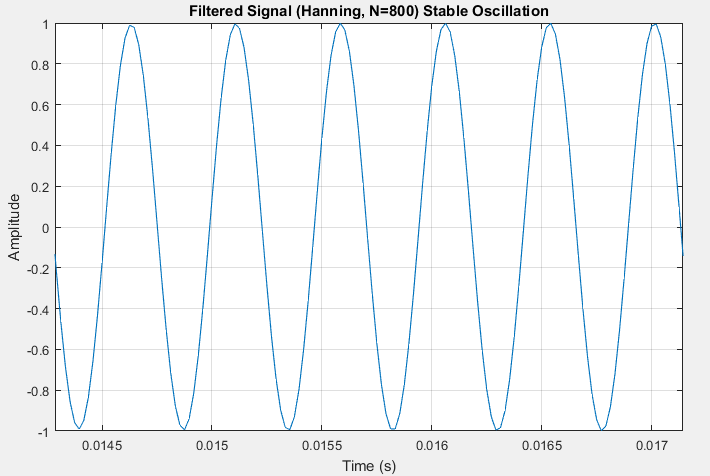
Figure 8: Time-domain graph during stable oscillation (Hanning Window N = 200)Figure 9: Frequency-domain graph (Hanning Window N = 200)

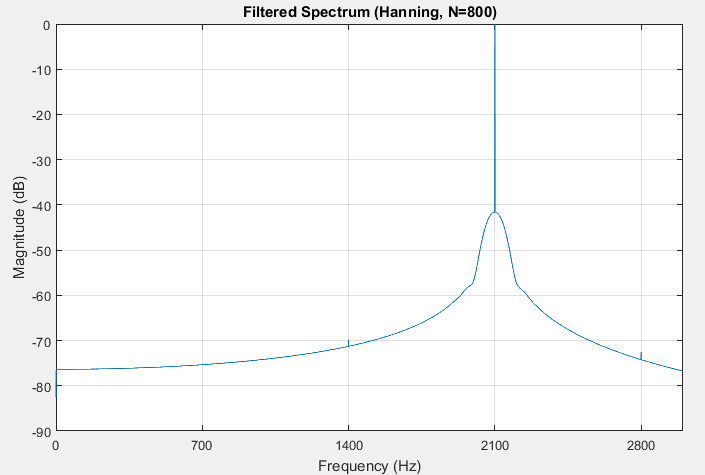
**Filtered Signal (Hanning Window; N = 400)**

  
Figure 10: Time-domain graph during stable oscillation (Hanning Window N = 400)

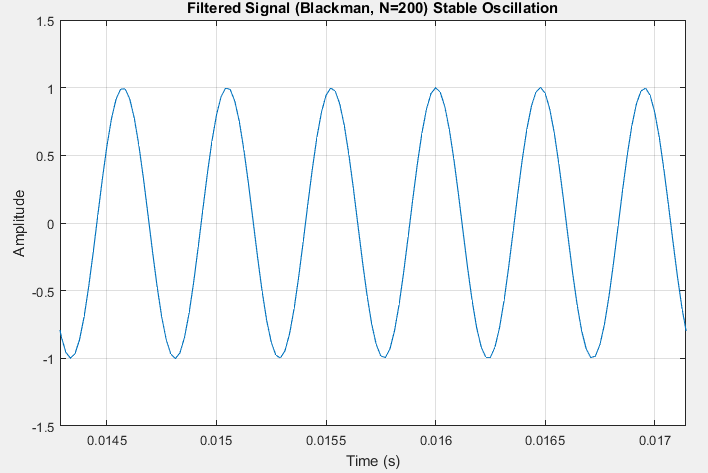
  
Figure 11: Frequency-domain graph (Hanning Window N = 400)

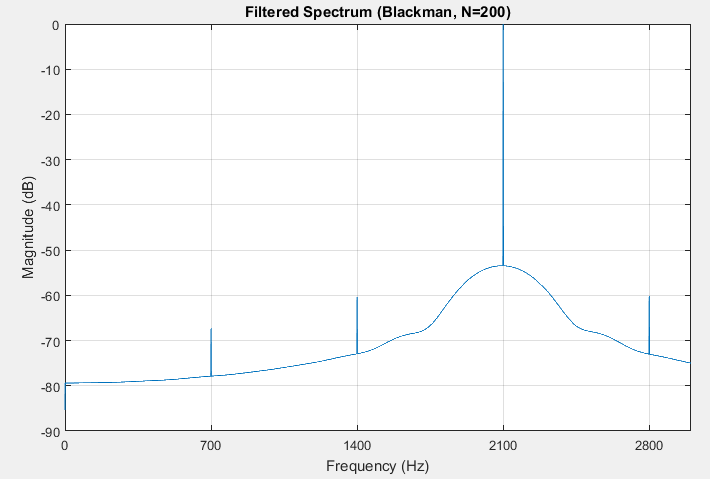
**Filtered Signal (Hanning Window; N = 800)**

  
Figure 12: Time-domain graph during stable oscillation (Hanning Window N = 800)

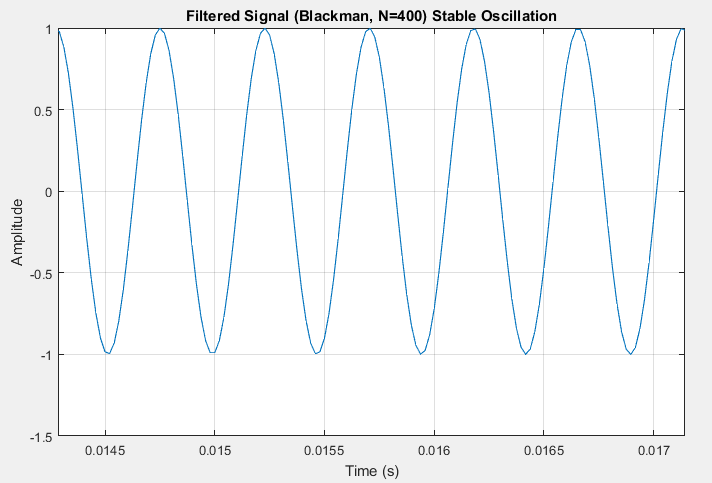
  
Figure 13: Frequency-domain graph (Hanning Window N = 800)

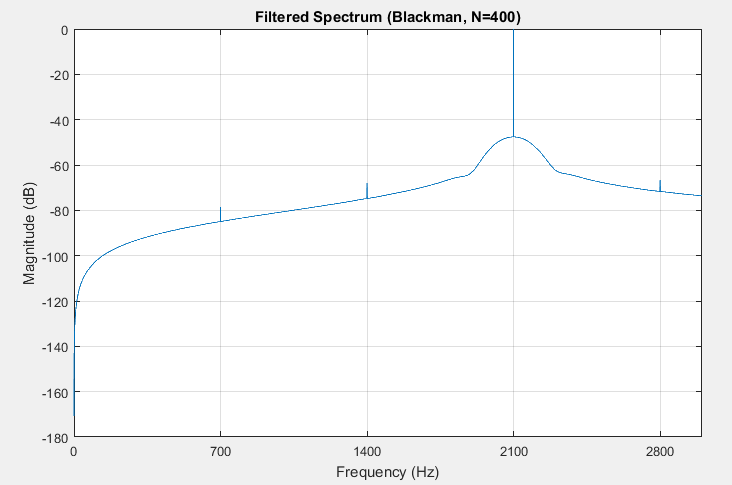
**Filtered Signal (Blackman Window; N = 200)**

  
Figure 14: Time-domain graph during stable oscillation (Blackman Window; N = 200)

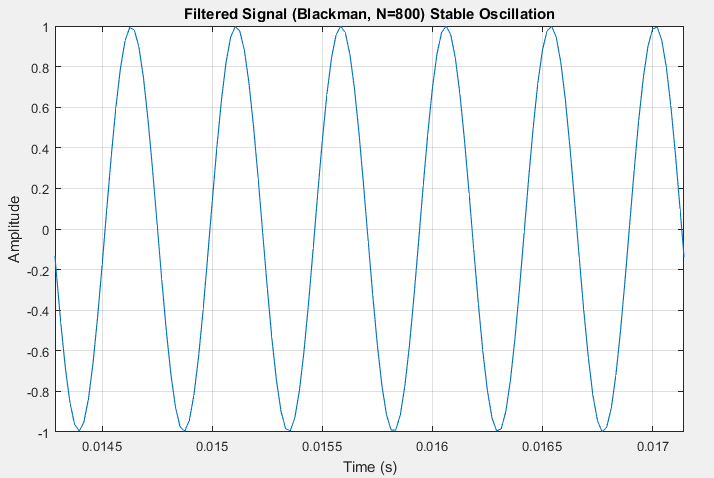
  
Figure 15: Frequency-domain graph (Blackman Window; N = 200)

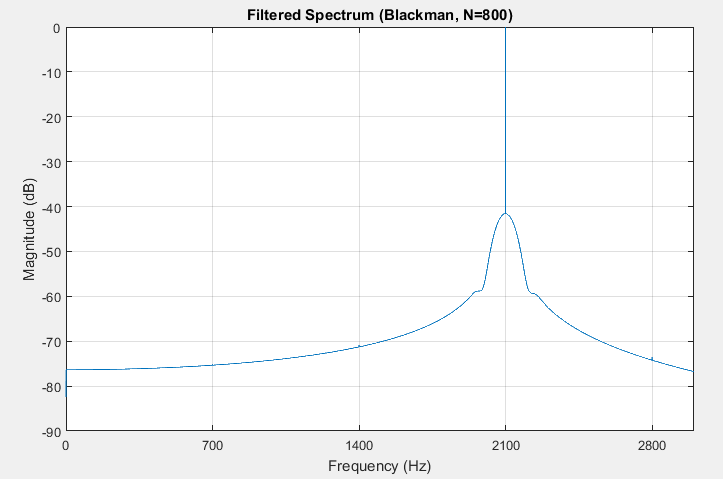
**Filtered Signal (Blackman Window; N = 400)**

  
Figure 16: Time-domain graph during stable oscillation (Blackman Window; N = 400)

  
Figure 17: Frequency-domain graph (Blackman Window; N = 400)

**Filtered Signal (Blackman Window; N = 800)**

Figure 18: Time-domain graph during stable oscillation (Blackman Window; N = 800)

Figure 19: Frequency-domain graph (Blackman Window; N = 800)

**Comparison of Different Windowing Techniques: Hanning vs. Blackman**

Frequency-Domain Performance:

N = 200  
 At a shorter window length (N=200), both Hanning and Blackman show similar filter attenuation performance. However, Blackman achieves a noticeably higher attenuation for the 600 Hz component (farthest from the passband center frequency), attenuating it to approximately -84 dB, which is around 8 dB better than Hanning’s attenuation of -72 dB.  
 This indicates that at smaller window lengths, Blackman has a slight advantage in suppressing unwanted frequencies at greater distances from the passband. This superior suppression is due to Blackman’s more aggressive sidelobe tapering. From a practical standpoint, this difference can be significant in applications requiring stringent noise rejection. However, this improved attenuation comes with a trade-off of a slightly broader transition region due to Blackman's wider main lobe.

N = 400  
 The differences between the two windowing techniques become more apparent. For instance, Hanning attenuates the 1200 Hz and 2400 Hz components to about -60 dB, while Blackman achieves approximately -70 dB of attenuation at the same frequencies—roughly a 10 dB improvement in stopband attenuation. This 10 dB improvement in stopband rejection demonstrates Blackman's greater ability to minimize spectral leakage, especially for frequencies further away from the passband. This stronger suppression makes Blackman the better choice when improved stopband performance is critical, particularly at moderate window lengths.

N = 800  
 At this larger window length, both Hanning and Blackman show nearly identical performance. The unwanted frequency components are attenuated to levels below detectable thresholds, making the frequency-domain graphs look similar for both techniques. In this case, the difference between the two methods is negligible. This indicates that for high N values, the window choice becomes less critical, as the sheer length of the filter dominates performance. However, the higher computational cost and latency associated with N = 800 must be considered in real-time applications.

Mathematical Context:

The performance difference arises due to the inherent properties of each window. The Blackman window uses a three-term cosine series:

This results in better side lobe suppression but with a wider main lobe, meaning broader transition bands.

In contrast, the Hanning window:

has moderate side lobe attenuation and narrower main lobes, resulting in sharper frequency transitions but more leakage. This theoretical explanation aligns with the measured values in the frequency-domain plots provided in the MATLAB section of the report.

Conclusion:  
In terms of stopband decay rate, Blackman windowing outperforms Hanning windowing, particularly at moderate window lengths (N = 400). The Blackman window provides a steeper fall-off in the stopband, leading to stronger suppression of unwanted frequencies. The Hanning window may be more favorable for applications requiring faster transitions and slightly lower computational load due to its narrower main lobe. At larger window lengths (N = 800), both techniques show comparable performance, effectively eliminating unwanted frequencies.

**Trade-Off: Low N vs. Moderate N vs. High N**

The window length (N) plays a significant role in determining the balance between filter latency and stopband attenuation.

Low N (200):  
 At low N values, the filter latency is the shortest, as shown in the time-domain graphs. However, the stopband attenuation is poor, with components like 1200 Hz and 2400 Hz only attenuated to about -35 dB. This results in visible distortions in the time-domain, making the filter unsuitable for applications that require higher precision and attenuation.

Moderate N (400):  
 Using a moderate N length offers a balanced compromise between latency and performance. The filter manages to attenuate the unwanted components (1200 Hz and 2400 Hz) to levels below -60 dB, making the unwanted frequencies almost negligible. Distortions in the time-domain are minimal or absent, making this an ideal choice for many applications that require both reasonable latency and strong stopband attenuation.

High N (800):  
 At higher N values, the FIR filter virtually eliminates unwanted frequencies, as seen in the frequency-domain graphs. However, this comes at the cost of significantly increased latency. The time-domain graphs show noticeable delays in the signal, making this setting less ideal for applications where low latency is critical.

Conclusion:  
 The choice of N depends on the desired trade-off between stopband attenuation and latency. For most applications, a moderate N (400) offers a practical balance between the two, providing sufficient attenuation of unwanted frequencies with minimal distortion in the time-domain. If a higher level of attenuation is required and latency is not a concern, a higher N (800) can be used, although this introduces noticeable delays.

**Appendix: MATLAB Source Code**  
  
% Parameters

Fs = 48000; % Sampling frequency (Hz)

freqs = [700, 1400, 2100, 2800]; % Signal C frequencies

t = 0:1/Fs:1; % 1-second time vector

% Generate sinusoids

freqs1 = sin(2\*pi\*700\*t);

freqs2 = sin(2\*pi\*1400\*t);

freqs3 = sin(2\*pi\*2100\*t);

freqs4 = sin(2\*pi\*2800\*t);

signal = freqs1 + freqs2 + freqs3 + freqs4; % Signal C

% Filter parameters

fc = 2100; % Center frequency (Hz)

BW = 50; % Half bandwidth (Hz)

f\_low = fc - BW; % Low cutoff

f\_high = fc + BW; % High cutoff

WnL = f\_low/(Fs/2); % Scaled low cutoff

WnH = f\_high/(Fs/2); % Scaled high cutoff

% Plotting parameters

T = 1/700; % Period of 700 Hz

num\_periods = 2; % Plot 2 periods

time\_limit = num\_periods \* T;

offset\_count = 10; % Offset for stable oscillation

ot = offset\_count \* T;

t\_plot = 0:1/Fs:time\_limit;

% Time-domain plots before filtering

figure;

for i = 1:4

subplot(5,1,i);

plot(t\_plot, sin(2\*pi\*freqs(i)\*t\_plot));

title(['Sinusoid at ', num2str(freqs(i)), ' Hz']);

xlabel('Time (s)'); ylabel('Amplitude');

xlim([0 time\_limit]); grid on;

end

subplot(5,1,5);

plot(t\_plot, sum(sin(2\*pi\*freqs'\*t\_plot)));

title('Signal C'); xlabel('Time (s)'); ylabel('Amplitude');

xlim([0 time\_limit]); grid on;

% Frequency-domain plot before filtering

L = 48000;

f = Fs\*(0:(L/2))/L;

Y = fft(signal, L);

P2 = abs(Y/L);

P1 = P2(1:(L/2)+1);

P1(2:end-1) = 2\*P1(2:end-1);

figure;

plot(f, 20\*log10(P1));

title('Signal C Spectrum'); xlabel('Frequency (Hz)'); ylabel('Magnitude (dB)');

xlim([0 3000]); set(gca, 'XTick', 0:700:2800); grid on;

% Time-domain plot signal before filtering

figure;

plot(t, signal);

title(['Signal C Oscillation']);

xlabel('Time (s)'); ylabel('Amplitude');

xlim([ot time\_limit+ot]); grid on;

% Filter design and plotting

window\_types = {'hanning', 'blackman'};

N\_values = [200, 400, 800];

for win = 1:length(window\_types)

window\_type = window\_types{win};

if strcmp(window\_type, 'hanning')

plot\_lbl = 'Hanning';

else

plot\_lbl = 'Blackman';

end

for N = N\_values

% Generate window

if strcmp(window\_type, 'hanning')

wdw = hann(N);

else

wdw = blackman(N);

end

% Design and apply filter

b = fir1(N-1, [WnL, WnH], 'bandpass', wdw);

filtered\_signal = filter(b, 1, signal);

% Time-domain plot

figure;

plot(t, filtered\_signal);

title(['Filtered Signal (', plot\_lbl, ', N=', num2str(N), ') Stable Oscillation']);

xlabel('Time (s)'); ylabel('Amplitude');

xlim([ot time\_limit+ot]); grid on;

% Frequency-domain plot

Y\_filt = fft(filtered\_signal, Fs);

P2\_filt = abs(Y\_filt/Fs);

P1\_filt = P2\_filt(1:floor(Fs/2)+1);

P1\_filt(2:end-1) = 2\*P1\_filt(2:end-1);

figure;

plot(f, 20\*log10(P1\_filt));

title(['Filtered Spectrum (', plot\_lbl, ', N=', num2str(N), ')']);

xlabel('Frequency (Hz)'); ylabel('Magnitude (dB)');

xlim([0 3000]); set(gca, 'XTick', 0:700:2800); grid on;

end

end