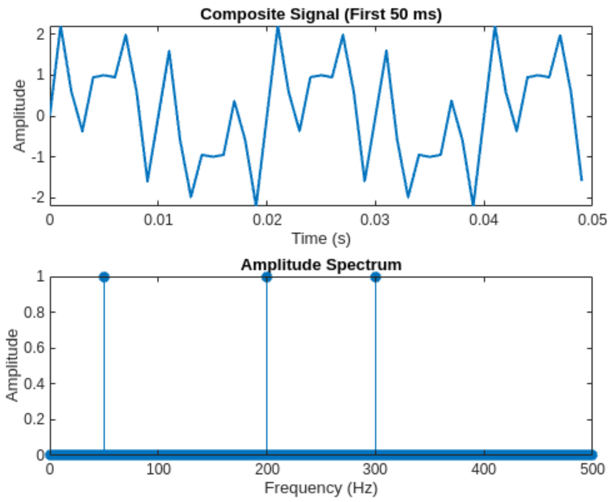
DSP lab 10
Name: Basil khowaja bk08432

Task 1)

code:

```
clear; clc; close all;

fs = 1000;
tmax = 3;
dt = 1/fs;
t = 0:dt:(tmax - dt);
N = length(t);

flow  = 50;
fmid  = 200;
fhigh = 300;

signal_low  = sin(2*pi*flow*t);
signal_mid  = sin(2*pi*fmid*t);
signal_high = sin(2*pi*fhigh*t);
composite_signal = signal_low + signal_mid + signal_high;
X     = fft(composite_signal);
X_mag = abs(X) / N;
f     = (0:N-1) * (fs/N);
halfN = floor(N/2) + 1;
X_mag_single = X_mag(1:halfN);
f_single     = f(1:halfN);

if mod(N, 2) == 0
    X_mag_single(2:end-1) = 2 * X_mag_single(2:end-1);
else
    X_mag_single(2:end) = 2 * X_mag_single(2:end);
end
samples_to_plot = 1:50;

figure('Name','Composite Signal & Single-Sided FFT','NumberTitle','off');
subplot(2,1,1);
plot(t(samples_to_plot), composite_signal(samples_to_plot), 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Composite Signal (First 50 ms)');

subplot(2,1,2);
stem(f_single, X_mag_single, 'filled');
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Amplitude Spectrum');
xlim([0 500]);
```

output:



The output confirms that the composite signal consists of three sinusoidal components at 50 Hz, 200 Hz, and 300 Hz. The time-domain plot shows variations in amplitude due to constructive and destructive interference between these sinusoids, resulting in peaks reaching around ±2. The frequency-domain plot correctly identifies the three dominant frequency components, each with an amplitude of approximately 1, verifying that the FFT accurately represents the signal's spectral content. This demonstrates that the composite signal is correctly generated and its frequency components align with expectations.

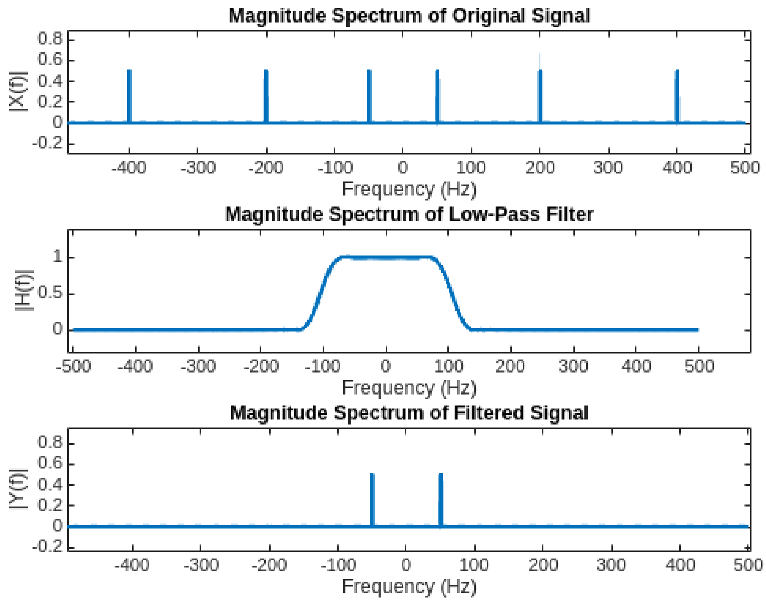
Task 2)

code:

```
%task 2
clc; clear; close all;
fs = 1000;
tmax = 3;
dt = 1/fs;
t = dt:dt:tmax;
N = length(t);

fc = 100;
Omega_p2 = 2*pi*fc/fs;
Omega_s2 = 1.1 * Omega_p2;
Omega_t2 = Omega_s2 - Omega_p2;
Omega_c = (Omega_p2 + Omega_s2) / 2;

f_low = 50;
f_mid = 200;
f_high = 400;

signal_low = sin(2*pi*f_low*t);
signal_mid = sin(2*pi*f_mid*t);
signal_high = sin(2*pi*f_high*t);
xt = signal_low + signal_mid + signal_high;
```

```
Xs = fft(xt, N);
X = fftshift(Xs) / N;
freq = linspace(-fs/2, fs/2, N);

figure;
subplot(3,1,1);
plot(freq, abs(X), 'linewidth', 2);
ylim([0 1.2]);
xlabel('Frequency (Hz)');
ylabel('|X(f)|');
title('Magnitude Spectrum of Original Signal');

L = 51;
M = (L-1)/2;
n = 0:(L-1);

h_ideal = sin(Omega_c*(n - M)) ./ (pi*(n - M));
h_ideal(M+1) = Omega_c / pi;

%hanning: a = 0.5, b = 0.5, c = 0 so W[n] = 0.5 - 0.5*cos(2*pi*n/(L-1))
w = 0.5 - 0.5*cos(2*pi*n/(L-1));

h = h_ideal .* w;
H = fft(h, N);
H = fftshift(H);
H_mag = abs(H);

subplot(3,1,2);
plot(freq, H_mag, 'linewidth', 2);
xlabel('Frequency (Hz)');
ylabel('|H(f)|');
title('Magnitude Spectrum of Low-Pass Filter');
xlim([-fs/2 fs/2]);

y = conv(xt, h, 'same');
Ys = fft(y, N);
Y = fftshift(Ys) / N;

subplot(3,1,3);
plot(freq, abs(Y), 'linewidth', 2);
ylim([0 1.2]);
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
title('Magnitude Spectrum of Filtered Signal');
```

Output :



Observation:

In the top plot, the original signal's FFT shows three clear peaks at ±50 Hz, ±200 Hz, and ±400 Hz, each with amplitude around 0.5 because it's a two-sided spectrum. The middle plot is the magnitude response of the low-pass filter designed with a manually computed Hanning window. It has a pass band around ±100 Hz, gradually rolling off beyond that range. Finally, the bottom plot confirms that after convolution with this filter, only the ±50 Hz component remains strong in the filtered signal, while the ±200 Hz and ±400 Hz components are suppressed. This verifies that the low-pass filter effectively allows frequencies below 100 Hz and attenuates higher frequencies.

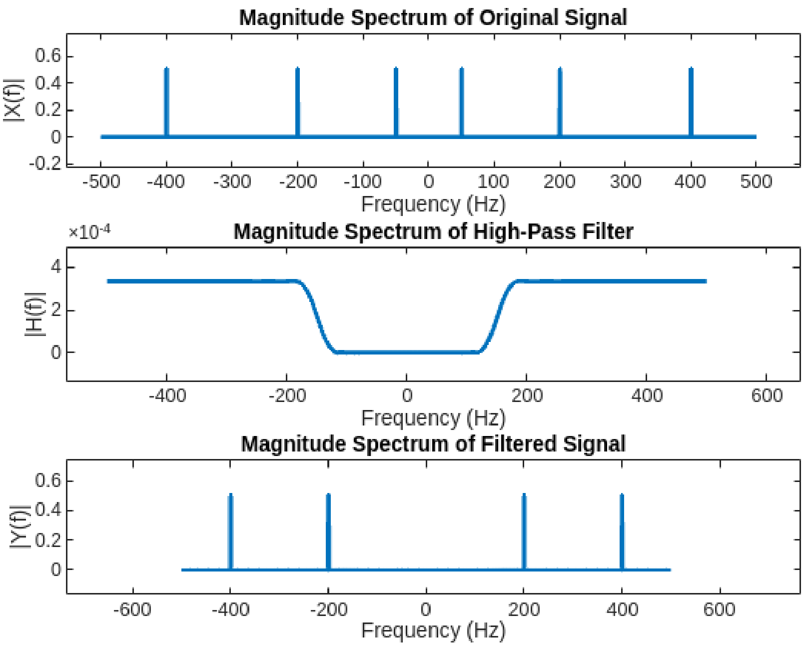
Task 3)

code:

```
%task 3
clc; clear; close all;

fs   = 1000;
tmax = 3;
dt   = 1/fs;
t    = 0:dt:(tmax-dt);
N    = length(t);

f_low  = 50;
f_mid  = 200;
f_high = 400;
x_low  = sin(2*pi*f_low*t);
x_mid  = sin(2*pi*f_mid*t);
x_high = sin(2*pi*f_high*t);

xt = x_low + x_mid + x_high;

Xs   = fft(xt, N);
X    = fftshift(Xs) / N;
freq = linspace(-fs/2, fs/2, N);

figure('Name','High-Pass Filter (Task 3)','NumberTitle','off');
subplot(3,1,1);
plot(freq, abs(X), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|X(f)|');
title('Magnitude Spectrum of Original Signal');

fc = 150;
L  = 51;
M  = (L - 1) / 2;
n  = 0:(L-1);

omega_c = 2*pi*fc/fs;
h_ideal_lp = sin(omega_c * (n - M)) ./ (pi * (n - M));
h_ideal_lp(M+1) = omega_c / pi;
ap = sin(pi * (n - M)) ./ (pi * (n - M));
ap(M+1) = 1;
h_ideal_hp = ap - h_ideal_lp;
% Hanning formula: w[n] = 0.5 - 0.5*cos(2*pi*n/(L-1))
w = 0.5 - 0.5*cos(2*pi*n/(L-1));
h_hp = h_ideal_hp .* w;

y = conv(xt, h_hp, 'same');

H = fftshift(fft(h_hp, N));
Hmag = abs(H) / N;

subplot(3,1,2);
plot(freq, Hmag, 'LineWidth', 2);
xlabel('Frequency (Hz)');
```

```
ylabel('|H(f)|');
title('Magnitude Spectrum of High-Pass Filter');

Ys = fft(y, N);
Y  = fftshift(Ys) / N;

subplot(3,1,3);
plot(freq, abs(Y), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
title('Magnitude Spectrum of Filtered Signal');
```

Output:



Observation:

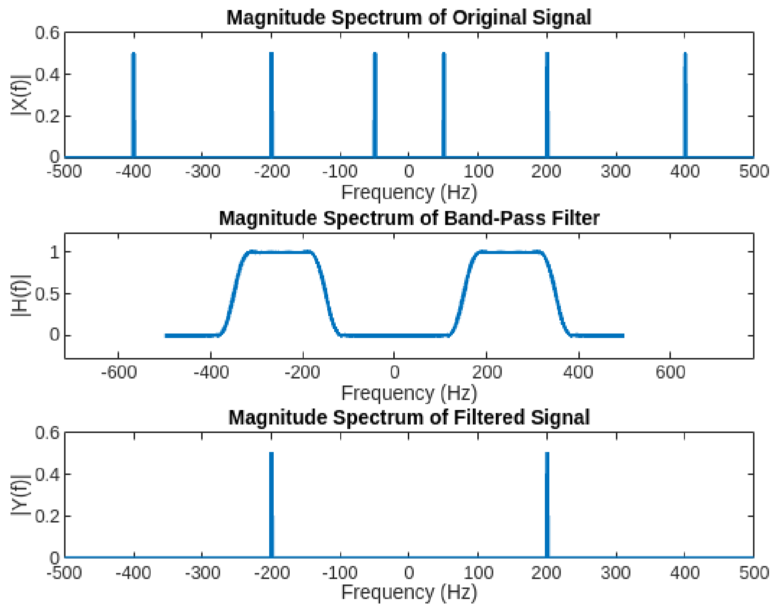
In the top plot, the original signal's spectrum shows three distinct peaks at ±50 Hz, ±200 Hz, and ±400 Hz. The middle plot displays the high-pass filter's magnitude response, which strongly attenuates low frequencies below about ±150 Hz and allows higher frequencies to pass. Although its passband amplitude appears small (on the order of 10^{-4}), that simply reflects the filter not being scaled for unity gain. Finally, the bottom plot confirms that after filtering, the 50 Hz component is removed, while the 200 Hz and 400 Hz components remain, demonstrating that the filter effectively suppresses low frequencies while preserving higher ones.

Task 4)

```matlab
%task4
clc; clear; close all;
fs   = 1000;
tmax = 3;
dt   = 1/fs;
t    = dt:dt:tmax;
N    = length(t);
f_low  = 50;
f_mid  = 200;
f_high = 400;

x_low  = sin(2*pi*f_low*t);
x_mid  = sin(2*pi*f_mid*t);
x_high = sin(2*pi*f_high*t);

xt = x_low + x_mid + x_high;

Xs   = fft(xt, N);
X    = fftshift(Xs) / N;
freq = linspace(-fs/2, fs/2, N);

figure('Name','Band-Pass Filter (Task 4)','NumberTitle','off');
subplot(3,1,1);
plot(freq, abs(X), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|X(f)|');
title('Magnitude Spectrum of Original Signal');

f1 = 150;
f2 = 350;

L  = 51;
M  = (L-1)/2;
n  = 0:(L-1);
omega_c2    = 2*pi*f2/fs;
h_low_f2    = sin(omega_c2*(n - M)) ./ (pi*(n - M));
h_low_f2(M+1) = omega_c2/pi;

omega_c1    = 2*pi*f1/fs;
h_low_f1    = sin(omega_c1*(n - M)) ./ (pi*(n - M));
h_low_f1(M+1) = omega_c1/pi;

h_bp_ideal  = h_low_f2 - h_low_f1;

% w[n] = 0.5 - 0.5*cos(2*pi*n/(L-1))
w = 0.5 - 0.5*cos(2*pi*n/(L-1));
h_bp = h_bp_ideal .* w;
y = conv(xt, h_bp, 'same');

H = fft(h_bp, N);
H = fftshift(H);
H_mag = abs(H);
```

```
subplot(3,1,2);
plot(freq, H_mag, 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|H(f)|');
title('Magnitude Spectrum of Band-Pass Filter');
xlim([-fs/2 fs/2]);
Ys = fft(y, N);
Y  = fftshift(Ys) / N;

subplot(3,1,3);
plot(freq, abs(Y), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
title('Magnitude Spectrum of Filtered Signal');
```

output:



Observation:

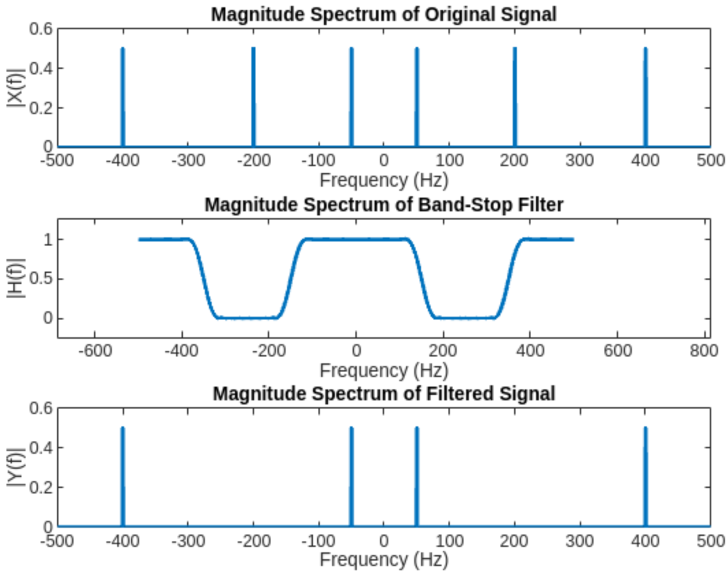
In the top plot, the original signal's spectrum shows peaks at ±50 Hz, ±200 Hz, and ±400 Hz, indicating three distinct frequency components. The middle plot shows the band-pass filter's magnitude response, with a clear passband from about 150 Hz to 350 Hz and attenuation outside that range. Finally, the bottom plot reveals that after filtering, only the ±200 Hz component remains strong, while the ±50 Hz and ±400 Hz components have been removed. This confirms that the band-pass filter passes frequencies within 150–350 Hz and rejects those below 150 Hz or above 350 Hz, as intended.

task 5)

code:

```
%task 5
clc; clear; close all;

fs= 1000;
tmax= 3;
dt= 1/fs;
t= dt:dt:tmax;
N= length(t);
f_low = 50;
f_mid = 200;
f_high= 400;
x_low = sin(2*pi*f_low*t);
x_mid = sin(2*pi*f_mid*t);
x_high= sin(2*pi*f_high*t);
xt = x_low + x_mid + x_high;

Xs   = fft(xt, N);
X    = fftshift(Xs) / N;
freq = linspace(-fs/2, fs/2, N);

figure('Name','Band-Stop Filter (Task 5)','NumberTitle','off');
subplot(3,1,1);
plot(freq, abs(X), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|X(f)|');
title('Magnitude Spectrum of Original Signal');

f1 = 150;
f2 = 350;

L = 51;
M = (L-1)/2;
n = 0:(L-1);

omega_f2   = 2*pi*f2/fs;
ImpLoWide  = sin(omega_f2*(n - M)) ./ (pi*(n - M));
ImpLoWide(M+1) = omega_f2 / pi;

omega_f1   = 2*pi*f1/fs;
ImpLoNarrow = sin(omega_f1*(n - M)) ./ (pi*(n - M));
ImpLoNarrow(M+1) = omega_f1 / pi;
ImpBand = ImpLoWide - ImpLoNarrow;
ap = sin(pi*(n - M)) ./ (pi*(n - M));
ap(M+1) = 1;
h_bs_ideal = ap - ImpBand;
% w[n] = 0.5 - 0.5*cos(2*pi*n/(L-1))
w = 0.5 - 0.5*cos(2*pi*n/(L-1));
h_bs = h_bs_ideal .* w;

y = conv(xt, h_bs, 'same');
H    = fft(h_bs, N);
H    = fftshift(H);
Hmag = abs(H);
```

```
subplot(3,1,2);
plot(freq, Hmag, 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|H(f)|');
title('Magnitude Spectrum of Band-Stop Filter');

Ys = fft(y, N);
Y  = fftshift(Ys) / N;

subplot(3,1,3);
plot(freq, abs(Y), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
title('Magnitude Spectrum of Filtered Signal');
```



Observation:

In the top plot, the original signal's spectrum reveals three frequency components at ±50 Hz, ±200 Hz, and ±400 Hz. The middle plot shows the band-stop filter's frequency response, which exhibits a deep notch between about ±150 Hz and ±350 Hz. Finally, the bottom plot confirms that after filtering, the ±200 Hz component is heavily attenuated, while the ±50 Hz and ±400 Hz components remain. This demonstrates that the filter effectively rejects the frequencies in the 150–350 Hz range and preserves those outside that band.

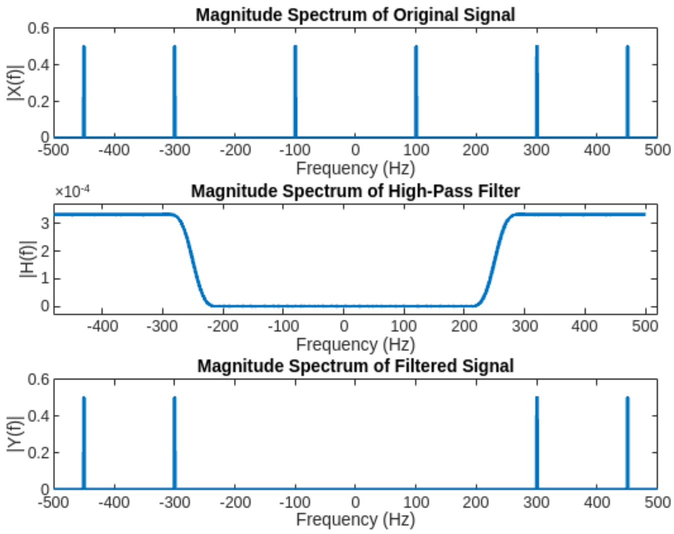
Post lab

Task 1

code:

```
clc; clear; close all;

fs = 1000;
tmax = 3;
dt = 1/fs;
t = dt:dt:tmax;
N = length(t);

f1 = 100;
f2 = 300;
f3 = 450;

x1 = sin(2*pi*f1*t);
x2 = sin(2*pi*f2*t);
x3 = sin(2*pi*f3*t);
xt = x1 + x2 + x3;

X = fftshift(fft(xt))/N;
freq = linspace(-fs/2, fs/2, N);

figure('Name','Post Lab Task 1: High-Pass Filter with Manual Hamming
Window','NumberTitle','off');
subplot(3,1,1);
plot(freq, abs(X), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|X(f)|');
title('Magnitude Spectrum of Original Signal');
xlim([-fs/2 fs/2]);

fc = 250;
L = 51;
M = (L-1)/2;
n = 0:(L-1);
omega_c = 2*pi*fc/fs;

h_lp = sin(omega_c*(n-M))./(pi*(n-M));
h_lp(M+1) = omega_c/pi;

ap = sin(pi*(n-M))./(pi*(n-M));
ap(M+1) = 1;
h_hp_ideal = ap - h_lp;

w = 0.54 - 0.46*cos(2*pi*n/(L-1));

h_hp = h_hp_ideal .* w;

H = fftshift(fft(h_hp, N))/N;
subplot(3,1,2);
plot(freq, abs(H), 'LineWidth', 2);
xlabel('Frequency (Hz)');
```

```
ylabel('|H(f)|');
title('Magnitude Spectrum of High-Pass Filter');
xlim([-fs/2 fs/2]);

y = conv(xt, h_hp, 'same');
Y = fftshift(fft(y, N))/N;
subplot(3,1,3);
plot(freq, abs(Y), 'LineWidth', 2);
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
title('Magnitude Spectrum of Filtered Signal');
xlim([-fs/2 fs/2]);
```



Observation:

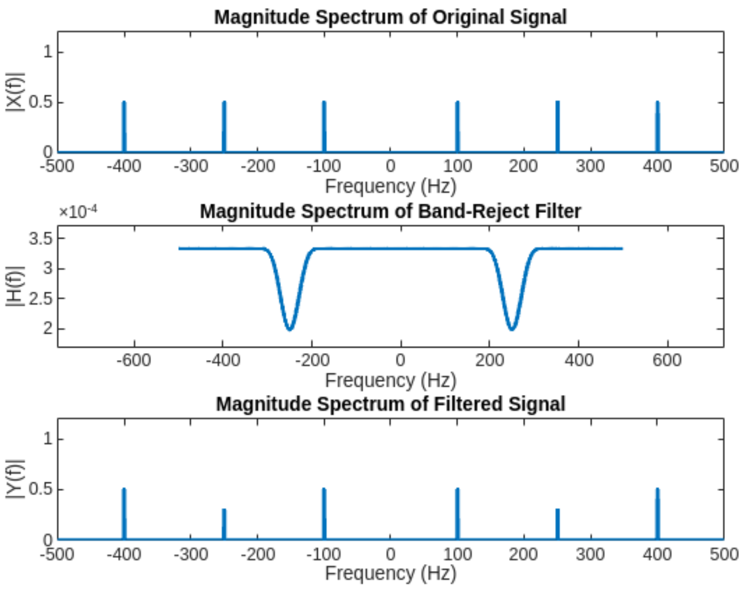
In the top plot, the original signal's spectrum shows three peaks at ±100 Hz, ±300 Hz, and ±450 Hz. The middle plot illustrates the high-pass filter's frequency response, created by subtracting a low-pass sinc from an all-pass response and then applying a manually computed Hamming window. The filter's gain is low for frequencies below about 250 Hz and higher for frequencies above that point. Finally, the bottom plot confirms that after filtering, the ±100 Hz component is significantly attenuated, while the ±300 Hz and ±450 Hz components remain, verifying that the filter effectively passes higher frequencies and rejects the lower ones around 100 Hz.

Task 2)

code:

```
%post lab task 2
clc; clear; close all;
fs = 1000;
tmax = 3;
dt = 1/fs;
t = dt:dt:tmax;
N = length(t);

f1 = 100;
f2 = 250;
f3 = 400;

x1 = sin(2*pi*f1*t);
x2 = sin(2*pi*f2*t);
x3 = sin(2*pi*f3*t);
xt = x1 + x2 + x3;

X = fftshift(fft(xt))/N;
freq = linspace(-fs/2, fs/2, N);

figure('Name','Post Lab Task 2: Band-Reject Filter using Blackman
Window','NumberTitle','off');
subplot(3,1,1);
plot(freq, abs(X), 'LineWidth',2);
xlabel('Frequency (Hz)');
ylabel('|X(f)|');
title('Magnitude Spectrum of Original Signal');
xlim([-fs/2 fs/2]);
ylim([0 1.2]);

L = 51;
M = (L-1)/2;
n = 0:(L-1);

f_low_stop  = 240;
f_high_stop = 260;
omega1 = 2*pi*f_low_stop/fs;
omega2 = 2*pi*f_high_stop/fs;

h_lp_wide = sin(omega2*(n - M)) ./ (pi*(n - M));
h_lp_wide(M+1) = omega2/pi;

h_lp_narrow = sin(omega1*(n - M)) ./ (pi*(n - M));
h_lp_narrow(M+1) = omega1/pi;

h_bp = h_lp_wide - h_lp_narrow;

ap = sin(pi*(n - M)) ./ (pi*(n - M));
ap(M+1) = 1;
h_bs_ideal = ap - h_bp;

w = 0.42 - 0.5*cos(2*pi*n/(L-1)) + 0.08*cos(4*pi*n/(L-1));

h_bs = h_bs_ideal .* w;
```

```
H = fftshift(fft(h_bs, N))/N;
subplot(3,1,2);
plot(freq, abs(H), 'LineWidth',2);
xlabel('Frequency (Hz)');
ylabel('|H(f)|');
title('Magnitude Spectrum of Band-Reject Filter');
xlim([-fs/2 fs/2]);

y = conv(xt, h_bs, 'same');
Y = fftshift(fft(y, N))/N;
subplot(3,1,3);
plot(freq, abs(Y), 'LineWidth',2);
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
title('Magnitude Spectrum of Filtered Signal');
xlim([-fs/2 fs/2]);
ylim([0 1.2]);
```



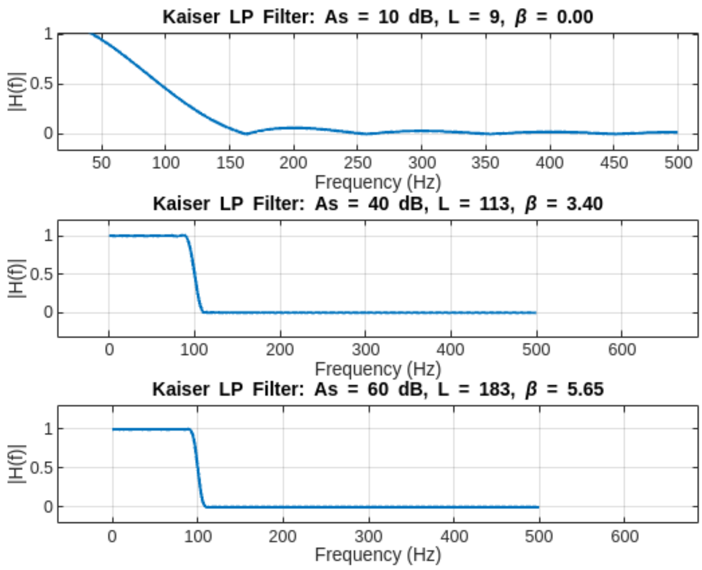
Observation:

In the top plot, the original signal's spectrum shows three main peaks at ±100 Hz, ±250 Hz, and ±400 Hz. The middle plot illustrates the band-reject filter's magnitude response, which features a narrow notch around ±250 Hz (from about 240 to 260 Hz) and retains higher gain at frequencies below 240 Hz and above 260 Hz. Finally, the bottom plot confirms that after filtering, the ±250 Hz component is greatly diminished, while the ±100 Hz and ±400 Hz components remain, demonstrating that the filter successfully removes the band around 250 Hz while preserving the other frequencies.

task 3)

code:

```
%post lab task 3:
clc; clear; close all;
fs = 1000;
fp = 100;
fs_stop = 120;
omega_p = 2*pi*fp/fs;
omega_s = 2*pi*fs_stop/fs;
delta_omega = omega_s - omega_p;

As_values = [10, 40, 60];

figure('Name','Kaiser Window Low-Pass Filter Designs','NumberTitle','off');

for i = 1:length(As_values)
    As = As_values(i);
    L_est = (2*pi*(As - 7.95))/(14.36*delta_omega) + 1;
    L = ceil(L_est);
    if mod(L,2)==0, L = L + 1; end
    if As < 21
        beta = 0;
    elseif As <= 50
        beta = 0.5842*(As-21)^0.4 + 0.07886*(As-21);
    else
        beta = 0.1102*(As-8.7);
    end

    M = (L-1)/2;
    n = 0:(L-1);
    omega_c = 2*pi*fp/fs;
    h_ideal = sin(omega_c*(n - M))./(pi*(n - M));
    h_ideal(M+1) = omega_c/pi;
    w = besseli(0, beta*sqrt(1 - ((n - M)/M).^2)) ./ besseli(0, beta);
    h = h_ideal .* w;
    [H, f_plot] = freqz(h, 1, 1024, fs);
    subplot(length(As_values),1,i);
    plot(f_plot, abs(H), 'LineWidth',1.5);
    xlabel('Frequency (Hz)');
    ylabel('|H(f)|');
    title(sprintf('Kaiser LP Filter: As = %d dB, L = %d, \\beta = %.2f', As, L, beta));
    grid on;
    xlim([0 fs/2]);
end
```



In these plots, each subplot corresponds to a Kaiser-windowed low-pass filter designed for a different stopband attenuation (As = 10 dB, 40 dB, and 60 dB). As As increases, the code calculates a larger filter length L and a higher β value. This results in a sharper transition at 100 Hz and deeper stopband attenuation. Concretely, the top filter (As=10 dB) has a very short length and a gentle slope, while the middle (As =40 dB) and bottom (As=60 dB) filters become progressively longer and exhibit a more abrupt cutoff with greater attenuation beyond 100 Hz. The key takeaway is that higher desired attenuation demands more filter taps (longer L) and a larger β, producing a narrower transition band and improved stopband suppression, as illustrated by these frequency responses.

1) How does the attenuation affect the filter length?

each higher desired attenuation (As) results in a larger filter length L. For example, at 10 dB the code computes a very short filter (only 9 taps), whereas at 60 dB the filter length jumps to 183 taps. This increase in L is necessary because achieving deeper stopband attenuation requires a longer impulse response (more taps). Higher attenuation demands a bigger filter so it can have a steeper transition and better suppression of unwanted frequencies.

2) What impact does increasing attenuation have on the transition width?

As the desired attenuation increases, the filter becomes longer and its transition from passband to stopband happens over a smaller range of frequencies. In our output, the filter for 10 dB attenuation shows a gradual change around the cutoff frequency, while the filters for 40 dB and 60 dB have a very steep change. This means that with higher attenuation, the filter more quickly switches from passing to blocking frequencies, resulting in a narrower transition width.

3) How does the Kaiser window compare to the Hanning window in terms of filter performance?

The Kaiser window is more flexible than the Hanning window. With the Kaiser window we can adjust a parameter (β) to control the trade-off between how sharp the filter's cutoff is and how well it suppresses unwanted frequencies. This means we can achieve deeper stopband attenuation and a narrower transition band if needed but for the Hanning window it has a fixed shape that generally results in a smoother, wider transition and less stopband suppression. so the Hanning window is simpler, but the Kaiser window can provide better performance when we need a more aggressive filter response.

4)What happens to the stopband attenuation as the attenuation increases?

When we increase the attenuation to 40 dB or 60 dB, the stopband level in the plot drops much lower, showing that the filter is much better at reducing the amplitude of frequencies in the stopband. Actually higher attenuation makes the stopband deeper, which means the filter rejects unwanted frequencies more effectively.