Homework3_2

Programmers

Mohammad MAhdi Elyasi - 9823007

Moein Nasiri - 9823093

Clear Workspace

```
close all;
clear;
clc;
```

Homework1

Here we declare some essential variables

```
fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
f2 = 8;
f3 = 12;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];
b = [0.969531 -1.923772 0.969531];
a = [1 -1.923772 0.939063];
filtered_signal = filter(b, a, x);
```

Now we plot filtered signals

```
figure('Name', 'filtered signal delta_f=4 Hz');
plot(t, filtered_signal);
grid on;
title('Filtered Signal with delta_f =4 Hz');
xlabel('n');
ylabel('amplitude');
```

```
x_0 = [1 zeros(1, 49)];
filtered_signal = filter(b, a, x_0);
figure('Name', 'impulse response delta_f=4 Hz');
stem(filtered_signal);
grid on;
title('impulse response with delta_f =4 Hz');
xlabel('n');
ylabel('amplitude');
```

```
b1 = [0.996088 -1.976468 0.996088];
a1 = [1 -1.976468 0.992177];
filtered_signal1 = filter(b1, a1, x);
figure('Name', 'filtered signal delta-f=0.5 Hz');
plot(t, filtered_signal1);
grid on;
title('Filtered Signal with delta-f =0.5 Hz');
xlabel('n');
ylabel('amplitude');
```

```
x_0 = [1 zeros(1, 49)];
filtered_signal = filter(b1, a1, x_0);
figure('Name', 'impulse response delta_f=0.5 Hz');
stem(filtered_signal);
grid on;
title('impulse response with delta_f =0.5 Hz');
xlabel('n');
ylabel('amplitude');
```

Homework2

Here we add some essential variables

```
fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
f2 = 8;
f3 = 12;
N = 2400;
n = 0:N - 1;
f0 = 8;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
w0 = 2 * pi * f0 / fs;

delta_f = 4;
delta_w = 2 * pi * delta_f / fs;
beta = tan(delta_w / 2);
x_delta = zeros(1, N + 1);
x_delta(1) = 1;

x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];
b1 = [1 -2 * cos(w0) 1] / (beta + 1);
a1 = [-2 * cos(w0) / (1 + beta) (1 - beta) / (1 + beta)];
w2 = 0;
w1 = 0;
x2 = 0;
```

```
x1 = 0;
y = zeros(1, N);
```

Here we do the diffrential equation

```
for i = 1:N
    y(i) = -a1(1) * w1 -a1(2) * w2 + b1(1) * x(i) + b1(2) * x1 + b1(3) * x2;
    w2 = w1;
    w1 = y(i);
    x2 = x1;
    x1 = x(i);
end
```

Now we plot the impulse responses

```
figure('Name', 'Impulse Responses of the Two Notch Filters');
plot(n, y, "LineWidth", 1.5);
title("Impuulse response of Filter H1 in Time Domain (\Deltaf = 4Hz)");
xlabel("n");
ylabel("Amplitude");
xlim([0 2400]);
grid on;
```

Homework3

Now we declare some essential variables

```
N = 1e4;
n = 0:N;
x_step = ones(1, N + 1);
w2 = 0;
w1 = 0;
x2 = 0;
x1 = 0;
y_step = zeros(1, N + 1);
```

Here we add differential equation

```
for i = 1:N + 1
    y_step(i) = -a1(1) * w1 -a1(2) * w2 + b1(1) * x_step(i) + b1(2) * x1 + b1(3) * x2;
    w2 = w1;
    w1 = y_step(i);
    x2 = x1;
    x1 = x_step(i);
end
```

Now we plot settling time value and step response of filters

```
settling_time_value = find(abs(y_step - y_step(end)) >= 0.01, 1, 'last') + 1;
figure('Name', 'Step Responses of the Two Notch Filters');
stem(n, y_step, "LineWidth", 1.5);
```

```
title("Step Response of Filter H1 in Time Domain (\Deltaf = 4Hz)");
xlabel("n");
ylabel("Amplitude");
xlim([0 200]);
ylim([1 - 0.1, 1 + 0.1]);
grid on;
hold on;
stem(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "LineWidth", 1.5, "Marker", "o");
text(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "Settling Time = sample " + (settling_time_value -1), 'position', ...
    [settling_time_value - 1, y_step(settling_time_value) + 0.04]);
```

```
fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
f2 = 8;
f3 = 12;
N = 2400;
n = 0:N;
f0 = 8;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
w0 = 2 * pi * f0 / fs;

delta_f = 0.5;
delta_w = 2 * pi * delta_f / fs;
beta = tan(delta_w / 2);
x_delta = zeros(1, N + 1);
x_delta(1) = 1;
x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];

b1 = [1 -2 * cos(w0) 1] / (beta + 1);
a1 = [-2 * cos(w0) / (1 + beta) (1 - beta) / (1 + beta)];
w2 = 0;
w1 = 0;
x2 = 0;
x1 = 0;
y_step = zeros(1, N + 1);

for i = 1:N + 1
    y_step(i) = -a1(1) * w1 -a1(2) * w2 + b1(1) * x_step(i) + b1(2) * x1 + b1(3) * x2;
    w2 = w1;
    w1 = y_step(i);
    x2 = x1;
    x1 = x_step(i);
end
```

```
settling_time_value = find(abs(y_step - y_step(end)) >= 0.01, 1, 'last') + 1;
figure('Name', 'Step Responses of the Two Notch Filters');
stem(n, y_step, "LineWidth", 1.5);
title("Step Response of Filter H1 in Time Domain (\Deltaf = 4Hz)");
xlabel("n");
ylabel("Amplitude");
xlim([0 600]);
ylim([1 - 0.1, 1 + 0.1]);
grid on;
hold on;
stem(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "LineWidth", 1.5, "Marker", "o");
text(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "Settling Time = sample " + (settling_time_value -1), 'position', ...
    [settling_time_value - 1, y_step(settling_time_value) + 0.04]);
```

part3

```
fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
f2 = 8;
f3 = 12;
N = 2400;
n = 0:N - 1;
f0 = 8;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
w0 = 2 * pi * f0 / fs;

delta_f = 4;
delta_w = 2 * pi * delta_f / fs;
beta = tan(delta_w / 2);
x_delta = zeros(1, N + 1);
x_delta(1) = 1;

x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];

b = [0.969531 -1.923772 0.969531];
a = [1 -1.923772 0.939063];
filtered_signal = filter(b, a, x);
figure('Name', 'compare x(t) and after filter H1');
subplot(211)
plot(t, filtered_signal);
grid on;
title('Filtered Signal with delta_f =4 Hz');
xlabel('n');
ylabel('amplitude');
```

```
subplot(212)
plot(t, x)
grid on;
title('input');
xlabel('n');
ylabel('amplitude');
```

Part4

```
Nf = 4096;
b1 = [0.969531 -1.923772 0.969531];
a1 = [1 -1.923772 0.939063];
[h1, w1] = freqz(b1, a1, Nf);

w1 = w1 ./ pi .* fs ./ 2;
```

H1

```
i4 = find(abs(w1 - 4) < 0.02);
h1g4 = abs(h1(i4(1))) % 4Hz gain of H1
```

```
h1g4 = 0.9483
```

```
b_4 = max(filtered_signal(1:800))
```

```
b_4 = 0.9695
```

```
i8 = find(abs(w1 - 8) < 0.02);
h1g8 = abs(h1(i8(1))) % 8Hz gain of H1
```

```
h1g8 = 0.0039
```

```
b_8 = max(filtered_signal(1000:1200))
```

```
b_8 = 0.0014
```

```
i12 = find(abs(w1 - 12) < 0.02);
h1g12 = abs(h1(i12(1))) % 12Hz gain of H1
```

```
h1g12 = 0.8584
```

```
b_12 = max(filtered_signal(1600:2000))
```

```
b_12 = 1.0239
```

Plotting Settling Times

```
[peaksH1_value, peaksH1_index] = findpeaks(filtered_signal);
```

H1 Settling Times

```
figure('name', 'Settling Times for H1');
plot(filtered_signal);
```

```
i = 1;
sections = [0, 2, 4] .* fs; % Start of signal sections
freqs = [4, 8, 12];
current_section = 1;

for delta_g = peaksH1_value(2:end) - peaksH1_value(1: end - 1) % Calculating changes

if (peaksH1_index(i) < sections(current_section)) % Skip other samples if settled one is alread
    i = i + 1;

    if (i > length(peaksH1_index))
        break
    end

    continue;
end

if (abs(delta_g) < 0.01) % Assumming that when the peaks value dont fluctuate as much, the outp
    xline(peaksH1_index(i), '--', {'Settling Time', peaksH1_index(i) * (1 / fs) - sections(curr
    fprintf("H1, Settling amplitude for f=%.0f: %.2f\n", freqs(current_section), filtered_signa
    current_section = current_section + 1; % Move on to next section when first settling point

    if (current_section > length(sections))
        break
    end

end

i = i + 1;
end
```

```
H1, Settling amplitude for f=4: 0.95
H1, Settling amplitude for f=8: 0.01

H1, Settling amplitude for f=12: 0.87
```

part5

```
fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
f2 = 8;
f3 = 12;
N = 2400;
n = 0:N - 1;
f0 = 8;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
w0 = 2 * pi * f0 / fs;

delta_f = 4;
delta_w = 2 * pi * delta_f / fs;
beta = tan(delta_w / 2);
```

```
x_delta = zeros(1, N + 1);
x_delta(1) = 1;

x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];
figure('Name', 'compare x(t) and after filter H1');
b1 = [0.996088 -1.976468 0.996088];
a1 = [1 -1.976468 0.992177];
filtered_signal = filter(b1, a1, x);
subplot(211)
plot(t, filtered_signal);
grid on;
title('Filtered Signal with delta_f =0.5 Hz');
xlabel('n');
ylabel('amplitude');
subplot(212)
plot(t, x)
grid on;
title('input');
xlabel('n');
ylabel('amplitude');
```

```
b2 = [0.996088 -1.976468 0.996088];
a2 = [1 -1.976468 0.992177];
[h2, w2] = freqz(b2, a2, Nf);
w2 = w2 ./ pi .* fs ./ 2;
```

H2

```
i4 = find(abs(w2 - 4) < 0.02);
h2g4 = abs(h2(i4(1))) % 4Hz gain of H2
```

```
h2g4 = 0.9990
```

```
b2_4 = max(filtered_signal(1:800))
```

```
b2_4 = 0.9996
```

```
i8 = find(abs(w2 - 8) < 0.02);
h2g8 = abs(h2(i8(1))) % 8Hz gain of H2
```

```
h2g8 = 0.0321
```

```
b2_8 = max(filtered_signal(1000:1200))
```

```
b2_8 = 0.4548
```

```
i12 = find(abs(w2 - 12) < 0.02);
h2g12 = abs(h2(i12(1))) % 12Hz gain of H2
```

```
h2g12 = 0.9973
```

```
b2_12 = max(filtered_signal(1600:2000))
```

```
b2_12 = 1.4959
```

Plotting Settling Times

```
[peaksH2_value, peaksH2_index] = findpeaks(filtered_signal);
```

H1 Settling Times

```
figure('name', 'Settling Times for H2');
plot(filtered_signal);
i = 1;
sections = [0, 2, 4] .* fs; % Start of signal sections
freqs = [4, 8, 12];
current_section = 1;

for delta_g = peaksH2_value(2:end) - peaksH2_value(1: end - 1) % Calculating changes

if (peaksH2_index(i) < sections(current_section)) % Skip other samples if settled one is alread
    i = i + 1;

    if (i > length(peaksH2_index))
        break
    end

    continue;
end

if (abs(delta_g) < 0.01) % Assumming that when the peaks value dont fluctuate as much, the outp
    xline(peaksH2_index(i), '--', {'Settling Time', peaksH2_index(i) * (1 / fs) - sections(curr
    fprintf("H1, Settling amplitude for f=%.0f: %.2f\n", freqs(current_section), filtered_signa
    current_section = current_section + 1; % Move on to next section when first settling point

    if (current_section > length(sections))
        break
    end

end

i = i + 1;
end
```

```
H1, Settling amplitude for f=4: 1.00
H1, Settling amplitude for f=8: 0.05

H1, Settling amplitude for f=12: 1.08
```

Part6

```
fs = 400;
Nf = 4096;
b1 = [0.969531 -1.923772 0.969531];
a1 = [1 -1.923772 0.939063];
[h, w1] = freqz(b1, a1, Nf);
```

Plotting frequency response of H1

```
figure('name', "Filter Responses H1")
subplot(211)
plot(w1 / pi * fs / 2, abs(h), 'LineWidth', 1.5);
xlim([0, 20]);
grid on;
hold on
xlabel('freq');
ylabel('|H(f)|');
title('Frequency Responses of H1');
```

Finding section of array which resides in the bandwidth of response

```
bw1 = find(abs(abs(h) - .5 ^ .5 * max(abs(h))) < 0.02);

fli1 = bw1(1); % F_l_1 index
fhi1 = bw1(end); % F_h_1 index

plot(w1(fli1) / pi * fs / 2, abs(h(fli1)), 'ro', 'LineWidth', 1.5);
plot(w1(fhi1) / pi * fs / 2, abs(h(fhi1)), 'ro', 'LineWidth', 1.5);

w1 = w1 ./ pi .* fs ./ 2;
```

H1

```
i4 = find(abs(w1 - 4) < 0.02);
h1g4 = abs(h(i4(1))); % 4Hz gain of H1
h1ag4 = angle(h(i4(1)));
i12 = find(abs(w1 - 12) < 0.02);
h1g12 = abs(h(i12(1))); % 12Hz gain of H1
h1ag12 = angle(h(i12(1)));
plot(4, h1g4, 'ko', 'LineWidth', 1.5);
plot(12, h1g12, 'ko', 'LineWidth', 1.5);

subplot(212)
plot(w1, angle(h), 'LineWidth', 1.5);
xlim([0, 20]);
grid on;
hold on
xlabel('freq');
ylabel('phase H1');
title('Frequency Responses of H1');
```

Finding section of array which resides in the bandwidth of response

```
bw1 = find(abs(abs(h) - 0.5 ^ .5 * max(abs(h))) < 0.02);

fli1 = bw1(1); % F_l_1 index
fhi1 = bw1(end); % F_h_1 index
fprintf('H1:%f\n', (fhi1 - fli1) * (fs / Nf / 2))
```

```
H1:4.150391
```

```
plot(w1(fli1), angle(h(fli1)), 'ro', 'LineWidth', 1.5);
plot(w1(fhi1), angle(h(fhi1)), 'ro', 'LineWidth', 1.5);
plot(4, h1ag4, 'ko', 'LineWidth', 1.5);
plot(12, h1ag12, 'ko', 'LineWidth', 1.5);
```

```
figure('name', "Filter Responses H2")
b1 = [0.996088 -1.976468 0.996088];
a1 = [1 -1.976468 0.992177];
[h1, w1] = freqz(b1, a1, Nf);
subplot(211)
plot(w1 / pi * fs / 2, abs(h1), 'LineWidth', 1.5);
xlim([0, 20]);
grid on;
hold on
xlabel('freq');
ylabel('|H(f)|');
title('Frequency Responses of H2');
```

Finding section of array which resides in the bandwidth of response

```
bw1 = find(abs(abs(h1) - .5 ^ .5 * max(abs(h1))) < 0.02);

fli1 = bw1(1); % F_l_1 index
fhi1 = bw1(end); % F_h_1 index

plot(w1(fli1) / pi * fs / 2, abs(h1(fli1)), 'ro', 'LineWidth', 1.5);
plot(w1(fhi1) / pi * fs / 2, abs(h1(fhi1)), 'ro', 'LineWidth', 1.5);
plot(4, abs(h1(4)), 'ko', 'LineWidth', 1.5);
plot(12, abs(h1(12)), 'ko', 'LineWidth', 1.5);
subplot(212)
plot(w1 / pi * fs / 2, angle(h1), 'LineWidth', 1.5);
xlim([0, 20]);
grid on;
hold on
xlabel('freq');
ylabel('phase H2');
title('Frequency Responses of H2');
```

Finding section of array which resides in the bandwidth of response

```
bw1 = find(abs(abs(h1) - 0.5 ^ .5 * max(abs(h1))) < 0.02);

fli1 = bw1(1); % F_l_1 index
fhi1 = bw1(end); % F_h_1 index

plot(w1(fli1) / pi * fs / 2, angle(h1(fli1)), 'ro', 'LineWidth', 1.5);
plot(w1(fhi1) / pi * fs / 2, angle(h1(fhi1)), 'ro', 'LineWidth', 1.5);
plot(4, angle(h1(4)), 'ko', 'LineWidth', 1.5);
```

```
plot(12, angle(h1(12)), 'ko', 'LineWidth', 1.5);
```

```
fprintf('H2:%f\n', (fhi1 - fli1) * (fs / Nf / 2))
```

```
H2:0.488281
```

last Part ;)

H3

```
b3 = [0.030469 0 -0.030469];
a3 = [1 -1.923772 0.939063];
```

H4:

```
b4 = [0.003912 0 -0.003912];
a4 = [1 -1.976468 0.992177];

fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
f2 = 8;
f3 = 12;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];

filtered_signal = filter(b3, a3, x);
figure('Name', 'filtered signal delta_f=4 Hz');
plot(t, filtered_signal);
grid on;
title('Filtered Signal with delta_f =4 Hz');
xlabel('n');
ylabel('amplitude');
```

```
x_0 = [1 zeros(1, 49)];
filtered_signal = filter(b3, a3, x_0);
figure('Name', 'impulse response delta_f=4 Hz');
stem(filtered_signal);
grid on;
title('impulse response with delta_f =4 Hz');
xlabel('n');
ylabel('amplitude');
```

```

filtered_signal1 = filter(b4, a4, x);
figure('Name', 'filtered signal delta-f=0.5 Hz');
```

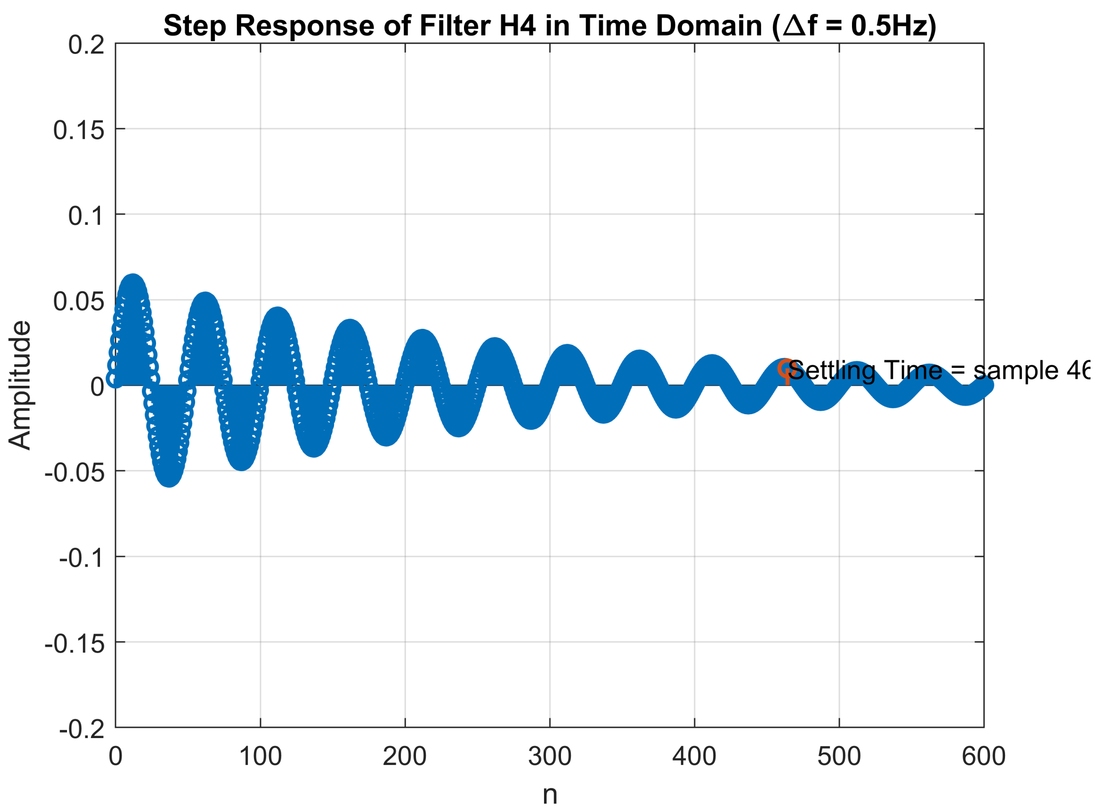
```
plot(t, filtered_signal1);
grid on;
title('Filtered Signal with delta-f =0.5 Hz');
xlabel('n');
ylabel('amplitude');
```

```
x_0 = [1 zeros(1, 49)];
filtered_signal = filter(b4, a4, x_0);
figure('Name', 'impulse response delta_f=0.5 Hz');
stem(filtered_signal);
grid on;
title('impulse response with delta_f =0.5 Hz');
xlabel('n');
ylabel('amplitude');
```

```
N = 1e4;
n = 0:N;
x_step = ones(1, N + 1);
w2 = 0;
w1 = 0;
x2 = 0;
x1 = 0;
y_step = filter(b3, a3, x_step);

settling_time_value = find(abs(y_step - y_step(end)) >= 0.01, 1, 'last') + 1;
figure('Name', 'Step Responses of the Two Notch Filters');
stem(n, y_step, "LineWidth", 1.5);
title("Step Response of Filter H3 in Time Domain (\Deltaf = 4Hz)");
xlabel("n");
ylabel("Amplitude");
xlim([0 200]);
ylim([- 0.2, + 0.2]);
grid on;
hold on;
stem(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "LineWidth", 1.5, "Marker", "o");
text(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "Settling Time = sample " + (settling_time_value -1), 'position', ...
    [settling_time_value - 1, y_step(settling_time_value) + 0.04]);
```

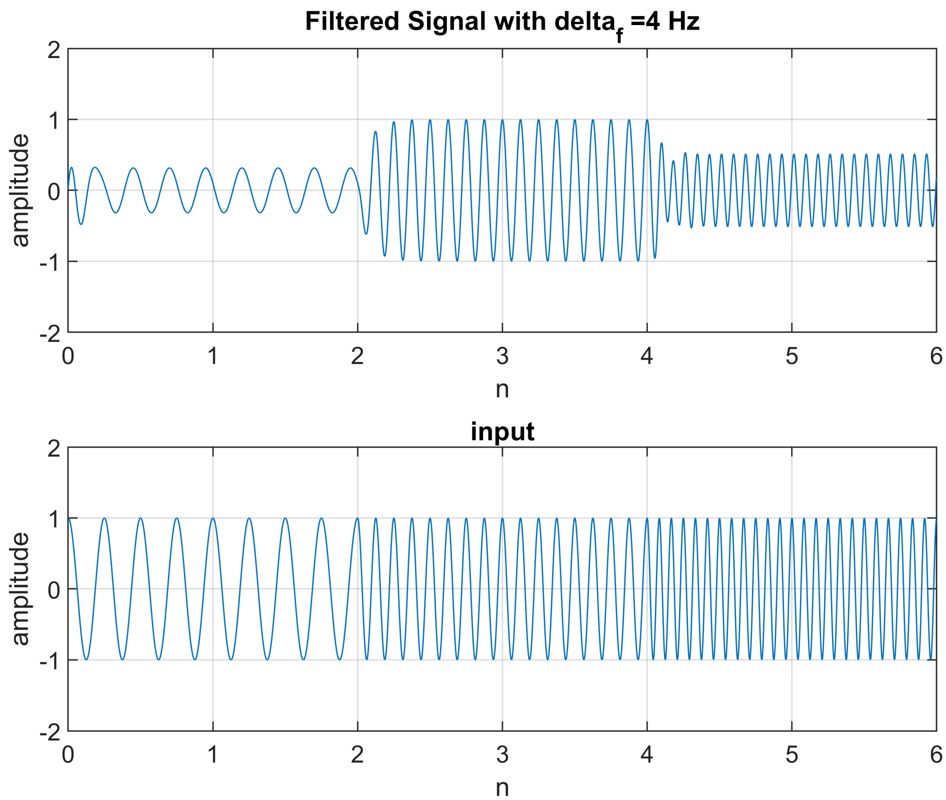
```
N = 1e4;
n = 0:N;
x_step = ones(1, N + 1);
```

```
w2 = 0;
w1 = 0;
x2 = 0;
x1 = 0;
y_step = filter(b4, a4, x_step);

settling_time_value = find(abs(y_step - y_step(end)) >= 0.01, 1, 'last') + 1;
figure('Name', 'Step Responses of the Two Notch Filters');
stem(n, y_step, "LineWidth", 1.5);
title("Step Response of Filter H4 in Time Domain (\Deltaf = 0.5Hz)");
xlabel("n");
ylabel("Amplitude");
xlim([0 600]);
ylim([- 0.2, + 0.2]);
grid on;
hold on;
stem(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "LineWidth", 1.5, "Marker", "o");
text(settling_time_value - 1, ...
    y_step(settling_time_value), ...
    "Settling Time = sample " + (settling_time_value -1), 'position', ...
    [settling_time_value - 1, y_step(settling_time_value) + 0.04]);
```




```
fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
```

```
f2 = 8;
f3 = 12;
N = 2400;
n = 0:N - 1;
f0 = 8;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
w0 = 2 * pi * f0 / fs;

delta_f = 4;
delta_w = 2 * pi * delta_f / fs;
beta = tan(delta_w / 2);
x_delta = zeros(1, N + 1);
x_delta(1) = 1;

x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];
figure('Name', 'compare x(t) and after filter H3');

filtered_signal = filter(b3, a3, x);
subplot(211)
plot(t, filtered_signal);
ylim([-2, + 2]);

grid on;
title('Filtered Signal with delta_f =4 Hz');
xlabel('n');
ylabel('amplitude');
subplot(212)
plot(t, x)
grid on;
title('input');
xlabel('n');
ylabel('amplitude');
ylim([-2, + 2]);
```




```
a2 = a3;
b2 = b3;
[h2, w2] = freqz(b2, a2, Nf);
w2 = w2 ./ pi .* fs ./ 2;
```

H2

```
i4 = find(abs(w2 - 4) < 0.02);
h2g4 = abs(h2(i4(1))) % 4Hz gain of H2
```

```
h2g4 = 0.3170
```

```
b2_4 = max(filtered_signal(1:800))
```

```
b2_4 = 0.3265
```

```
i8 = find(abs(w2 - 8) < 0.02);
h2g8 = abs(h2(i8(1))) % 8Hz gain of H2
```

```
h2g8 = 1.0000
```

```
b2_8 = max(filtered_signal(1000:1200))
```

```
b2_8 = 1.0000
```

```
i12 = find(abs(w2 - 12) < 0.02);
h2g12 = abs(h2(i12(1))) % 12Hz gain of H2
```

```
h2g12 = 0.5131
```

```
b2_12 = max(filtered_signal(1600:2000))
```

```
b2_12 = 1.0000
```

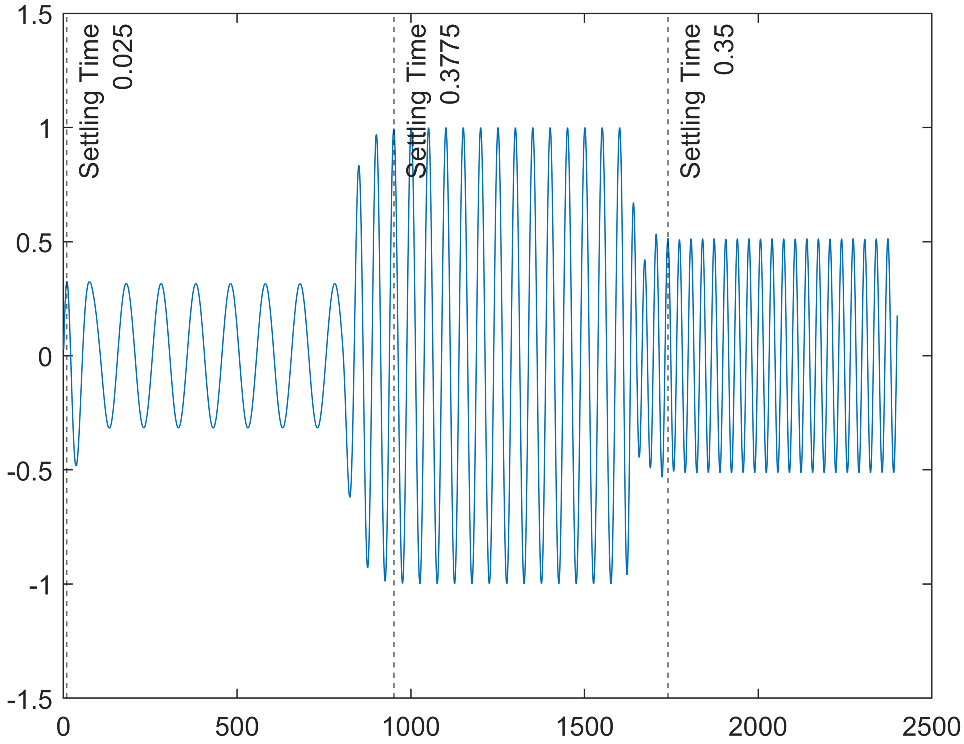
Plotting Settling Times

```
[peaksH2_value, peaksH2_index] = findpeaks(filtered_signal);
```

H3 Settling Times

```
figure('name', 'Settling Times for H3');
plot(filtered_signal);
i = 1;
sections = [0, 2, 4] .* fs; % Start of signal sections
freqs = [4, 8, 12];
current_section = 1;

for delta_g = peaksH2_value(2:end) - peaksH2_value(1: end - 1) % Calculating changes

if (peaksH2_index(i) < sections(current_section)) % Skip other samples if settled one is alreac
    i = i + 1;

    if (i > length(peaksH2_index))
        break
    end

    continue;
end

if (abs(delta_g) < 0.01) % Assumming that when the peaks value dont fluctuate as much, the outp
    xline(peaksH2_index(i), '--', {'Settling Time', peaksH2_index(i) * (1 / fs) - sections(curr
    fprintf("H1, Settling amplitude for f=%.0f: %.2f\n", freqs(current_section), filtered_signa
    current_section = current_section + 1; % Move on to next section when first settling point

    if (current_section > length(sections))
        break
    end

end

i = i + 1;
end
```

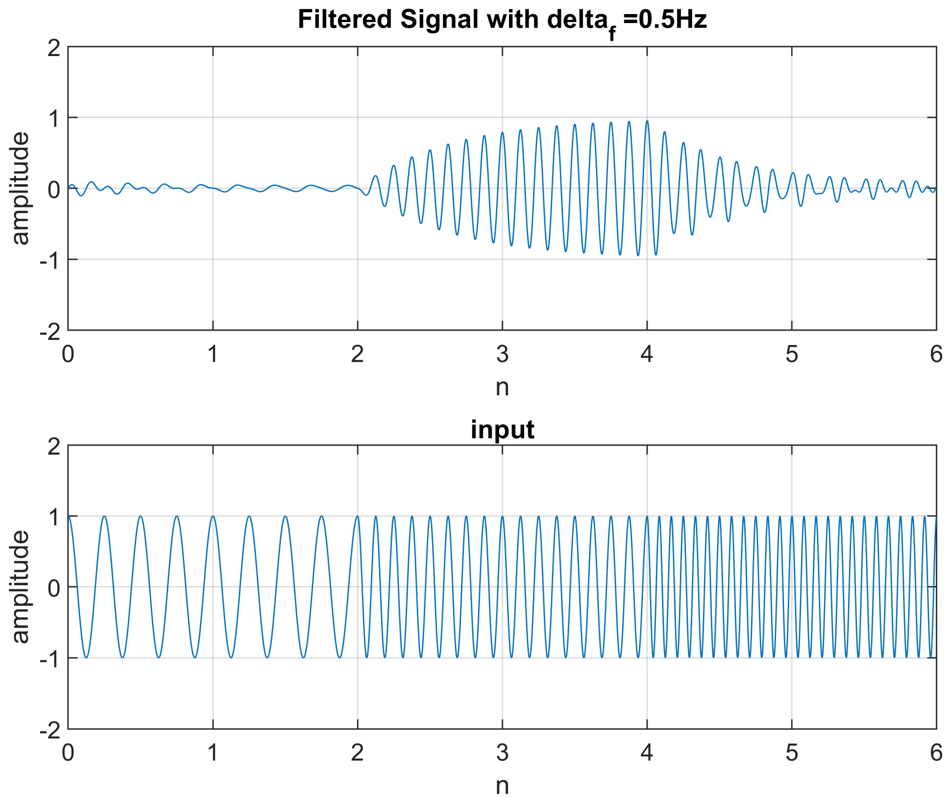
```
H1, Settling amplitude for f=4: 0.33
H1, Settling amplitude for f=8: 1.00
```




```
H1, Settling amplitude for f=12: 0.52
```

```
fs = 400;
t = 0:1 / fs:6 - (1 / fs);
f1 = 4;
f2 = 8;
f3 = 12;
N = 2400;
n = 0:N - 1;
f0 = 8;
t1 = 0:1 / fs:2 - (1 / fs);
t2 = 2:1 / fs:4 - (1 / fs);
t3 = 4:1 / fs:6 - (1 / fs);
% fs = 2000;
w0 = 2 * pi * f0 / fs;
delta_f = 4;
delta_w = 2 * pi * delta_f / fs;
beta = tan(delta_w / 2);
x_delta = zeros(1, N + 1);
x_delta(1) = 1;
x = [cos(2 * pi * f1 * t1) cos(2 * pi * f2 * t2) cos(2 * pi * f3 * t3)];
figure('Name', 'compare x(t) and after filter H3');

filtered_signal = filter(b4, a4, x);
subplot(211)
plot(t, filtered_signal);
ylim([-2, + 2]);

grid on;
```

```
title('Filtered Signal with delta_f =0.5Hz');
xlabel('n');
ylabel('amplitude');
subplot(212)
plot(t, x)
grid on;
title('input');
xlabel('n');
ylabel('amplitude');
ylim([-2, + 2]);
```




```
a2 = a4;
b2 = b4;
[h2, w2] = freqz(b2, a2, Nf);
w2 = w2 ./ pi .* fs ./ 2;
```

H2

```
i4 = find(abs(w2 - 4) < 0.02);
h2g4 = abs(h2(i4(1))) % 4Hz gain of H2
```

```
h2g4 = 0.0417
```

```
b2_4 = max(filtered_signal(1:800))
```

```
b2_4 = 0.0942
```

```
i8 = find(abs(w2 - 8) < 0.02);
```

```
h2g8 = abs(h2(i8(1))) % 8Hz gain of H2
```

```
h2g8 = 0.9996
```

```
b2_8 = max(filtered_signal(1000:1200))
```

```
b2_8 = 0.7877
```

```
i12 = find(abs(w2 - 12) < 0.02);
h2g12 = abs(h2(i12(1))) % 12Hz gain of H2
```

```
h2g12 = 0.0745
```

```
b2_12 = max(filtered_signal(1600:2000))
```

```
b2_12 = 0.9571
```

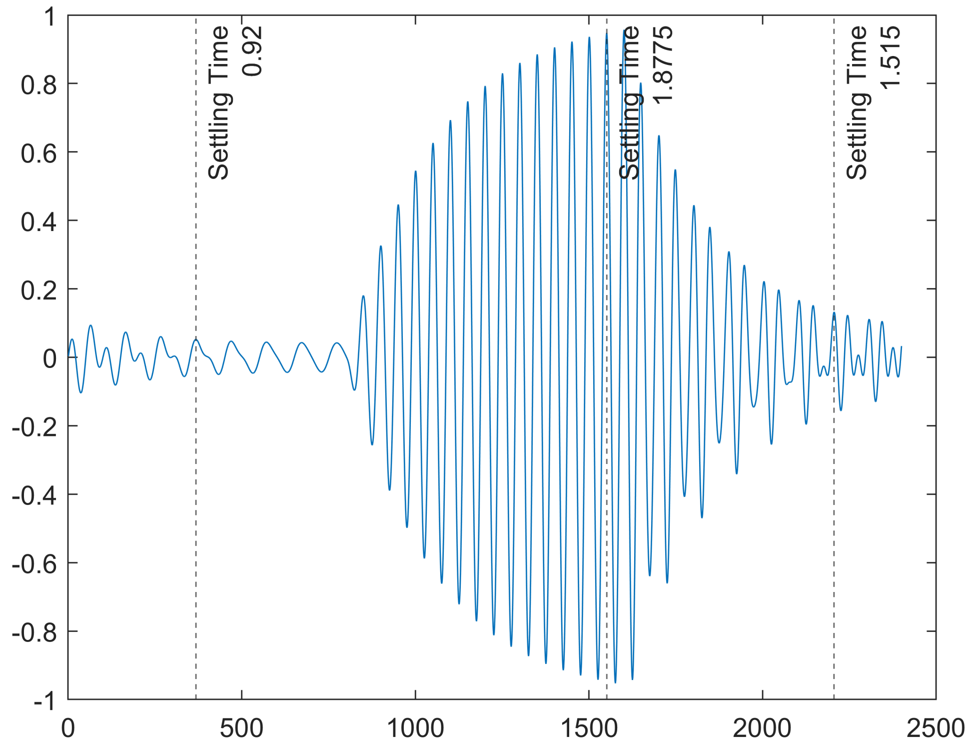
Plotting Settling Times

```
[peaksH2_value, peaksH2_index] = findpeaks(filtered_signal);
```

H4 Settling Times

```
figure('name', 'Settling Times for H3');
plot(filtered_signal);
i = 1;
sections = [0, 2, 4] .* fs; % Start of signal sections
freqs = [4, 8, 12];
current_section = 1;

for delta_g = peaksH2_value(2:end) - peaksH2_value(1: end - 1) % Calculating changes

if (peaksH2_index(i) < sections(current_section)) % Skip other samples if settled one is alread
    i = i + 1;

    if (i > length(peaksH2_index))
        break
    end

    continue;
end

if (abs(delta_g) < 0.01) % Assumming that when the peaks value dont fluctuate as much, the outp
    xline(peaksH2_index(i), '--', {'Settling Time', peaksH2_index(i) * (1 / fs) - sections(curr
    fprintf("H1, Settling amplitude for f=%.0f: %.2f\n", freqs(current_section), filtered_signa
    current_section = current_section + 1; % Move on to next section when first settling point

    if (current_section > length(sections))
        break
    end

end

i = i + 1;
end
```

```
H1, Settling amplitude for f=4: 0.05
H1, Settling amplitude for f=8: 0.95
```




```
H1, Settling amplitude for f=12: 0.13
```

```
fs = 400;
Nf = 4096;
[h1, w1] = freqz(b3, a3, Nf);
[h2, w2] = freqz(b4, a4, Nf);
```

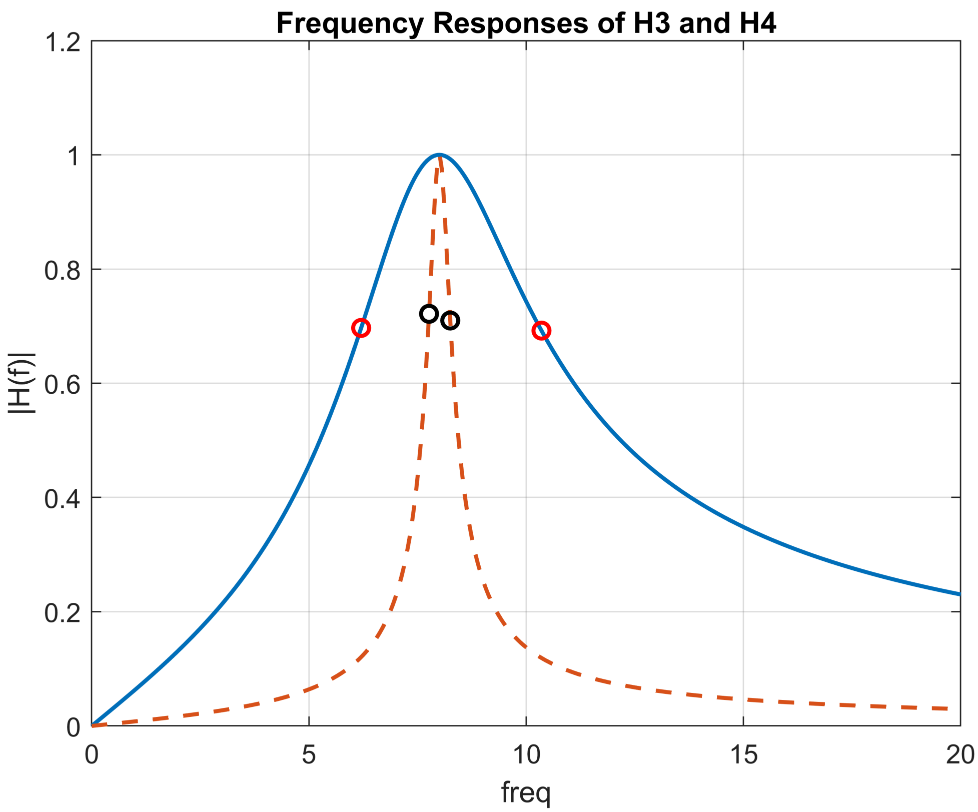
Plotting frequency response of H3 and H4

```
figure('name', "Peaking Filter Responses")
plot(w1 / pi * fs / 2, abs(h1), 'LineWidth', 1.5);
xlim([0, 20]);
grid on;
xlabel('freq');
ylabel('|H(f)|');
hold on;
title('Frequency Responses of H3 and H4');
plot(w2 / pi * fs / 2, abs(h2), '--', 'LineWidth', 1.5);
```

Finding section of array which resides in the bandwidth of response

```
bw1 = find(abs(abs(h1) - .5 ^ .5 * max(abs(h1))) < 0.02);
bw2 = find(abs(abs(h2) - .5 ^ .5 * max(abs(h2))) < 0.02);
fli1 = bw1(1);
fhi1 = bw1(end);
fli2 = bw2(1);
```

```
fhi2 = bw2(end);
plot(w1(fli1) / pi * fs / 2, abs(h1(fli1)), 'ro', 'LineWidth', 1.5);
plot(w1(fhi1) / pi * fs / 2, abs(h1(fhi1)), 'ro', 'LineWidth', 1.5);
plot(w2(fli2) / pi * fs / 2, abs(h2(fli2)), 'ko', 'LineWidth', 1.5);
plot(w2(fhi2) / pi * fs / 2, abs(h2(fhi2)), 'ko', 'LineWidth', 1.5);
```




```
fprintf("H3:\nFh-Fl=%f\n\nH4:\nFh-Fl=%f\n", (fhi1 - fli1) * (fs / Nf / 2), (fhi2 - fli2) * (fs
```

```
H3:
Fh-Fl=4.150391

H4:
Fh-Fl=0.488281
```