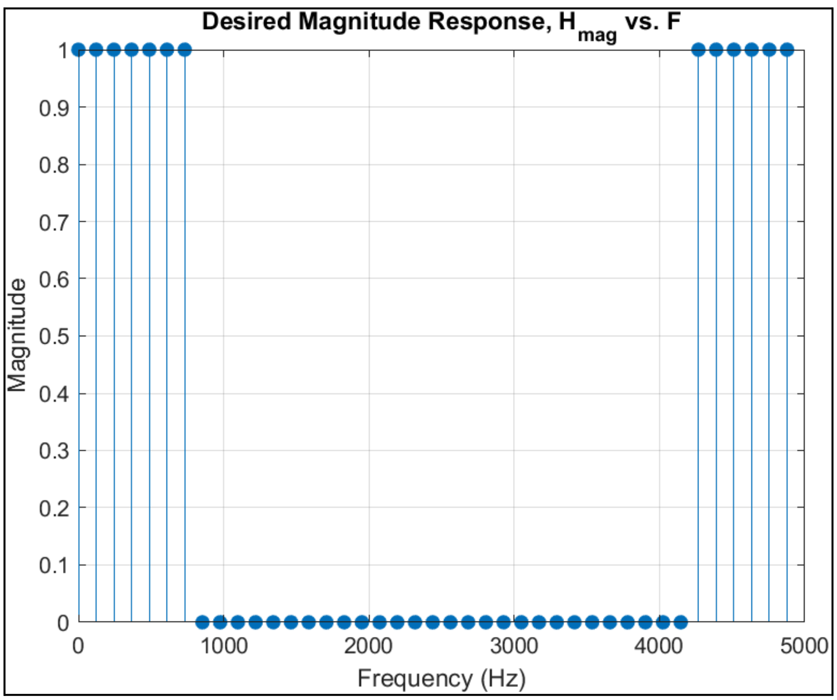
Remy Ren
Professor Rosenthal
EE125 - Digital Signal Processing
Dec. 10, 2024

Project 6: FIR Filter Design

Part 1: Filter Design by Frequency Sampling

Desired Magnitude Response

For our desired magnitude response, we want a length 41 filter, where the lowpass response is at 750 Hz. However, we want to leverage the conjugate-symmetry property of the Fourier transform. Therefore, we set up our desired response to respond to both the positive and negative frequencies.



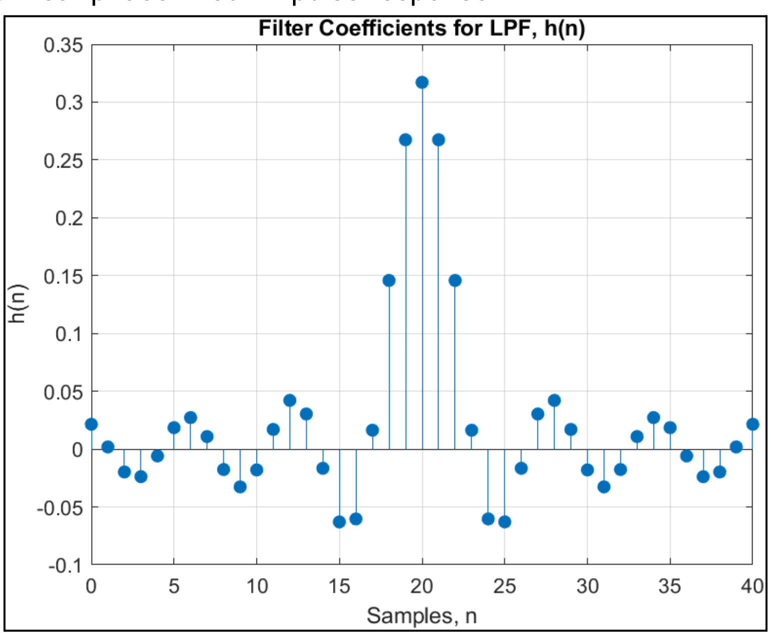
As a result, the left-side response contains the frequencies and the right-side response contains the negative frequencies. Each side has 6 points. The left-side contains an additional point at F=0Hz, as we require it to keep the DC component of a signal. Without it, the filter would no longer be a low pass, and instead be a DC-filtering bandpass.

```
% Parameters
Fs = 5000; % in Hz
L  = 41;
Fc = 750; % Cut off freq. in Hz
```

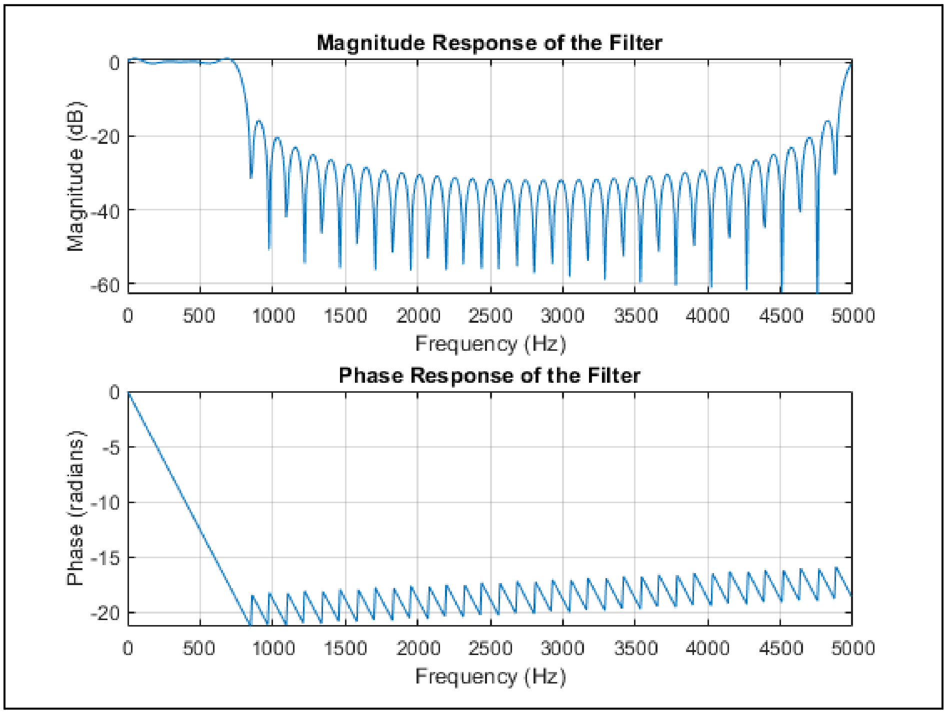
```
% Evenly spaced frequencies for F
F = (0:L-1) * (Fs / L);

disp(F);
length(F)

% Magnitude response
Hmag = F < Fc; % Set Hmag to 1 where F < 750 Hz, otherwise 0
% Conjugate-symmetric frequency response
% Filter is low pass, need to include negative, so F > -750 Hz
% 6 points on the right side, would be shifted to neg.
Hmag = Hmag + (F > (Fs - Fc));

% Plot the magnitude response
figure;
stem(F, Hmag, 'filled');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Desired Magnitude Response, H_{mag} vs. F');
grid on;
```

Checking Sinc Behavior of Impulse Response h(n)

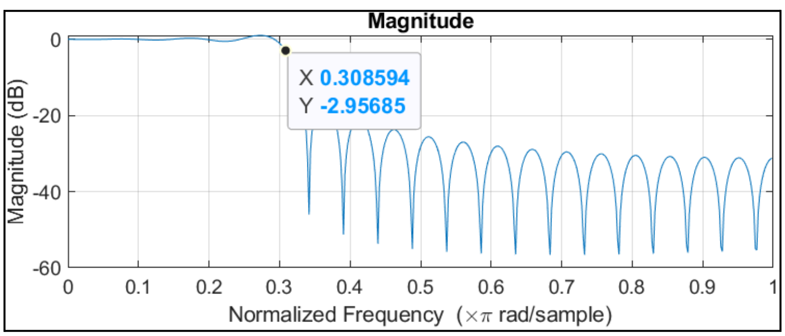
As our filter is an ideal low pass in frequency, it should resemble a sinc in samples. The impulse response is derived by taking the IFFT (same as FFT in MATLAB) of H(f). Importantly, we multiply H(f) by Hphase (I used the second method with dw, in discrete-time frequencies) to ensure that we get linear phase in our impulse response.



We do indeed get our expected sinc behavior in the impulse response of the filter.To further verify that our result is correct, we can see that the length in samples is 41, matching L=41. We also take the freqz, the frequency response of the filter, on h(n) to analyze it further.



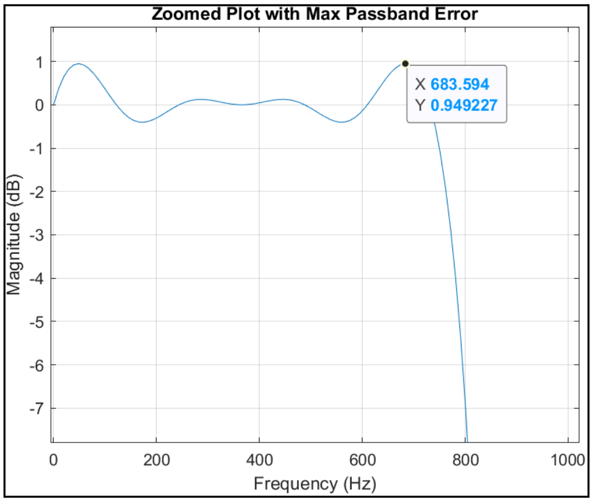
In the phase response of the filter, we obtained a linear phase up to the mainlobe/cutoff frequency, which is what we wanted. Once at the sidelobes, the phase is no longer linear, which makes sense as in the frequency magnitude response there's a lot of stopband ripple.

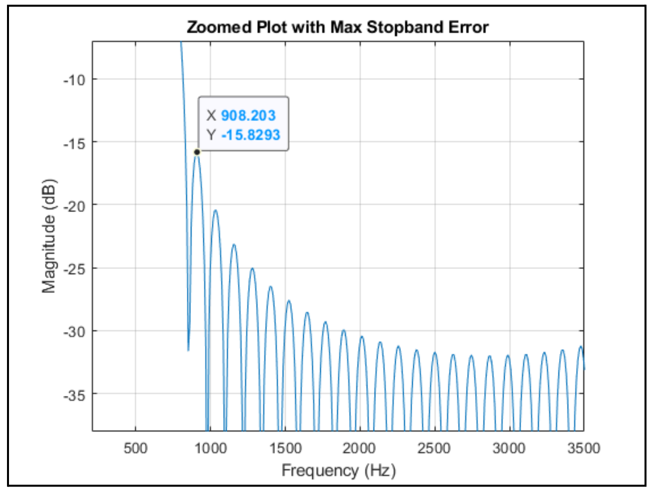


We can further verify that the operations we did are correct, by checking the cutoff frequency of our frequency magnitude response of h(n). -3dB typically marks the cutoff frequency, and the annotation shows that although there is not an exact sample point at -3dB, the frequency this occurs at is (unnormalized): 0.308594 * 2500Hz = 771.485 Hz. This is very close to our desired cutoff frequency of 750Hz, which means the filter is likely correct.

Characterize the Filter Performance

We designed our filter's desired magnitude response to immediately go from 1 to 0 as soon as the cutoff frequency is reached. This has negative effects on the ripple in our passband and stopband in the actual frequency magnitude response. We can see the effects of the max passband and stopband error below.



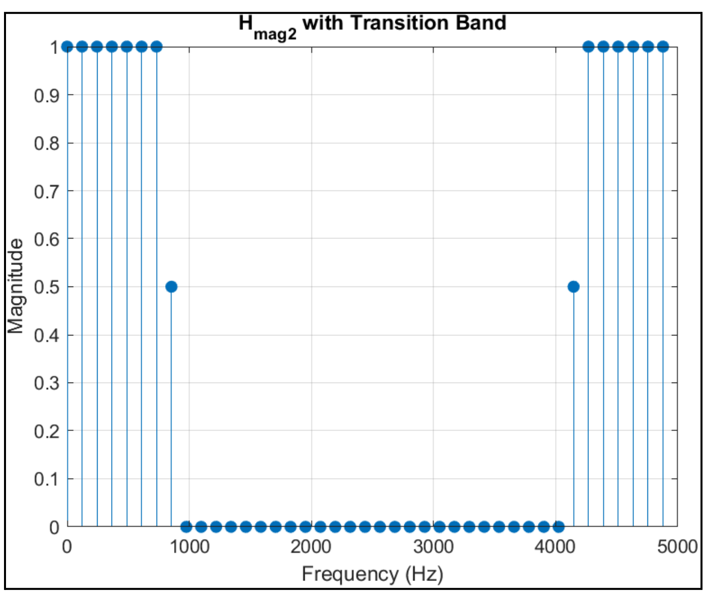


The Y values are in dB, which can make comparison a bit difficult without a linear scale. The calculated max error in linear is (passband is difference from 1, stopband is difference from 0):

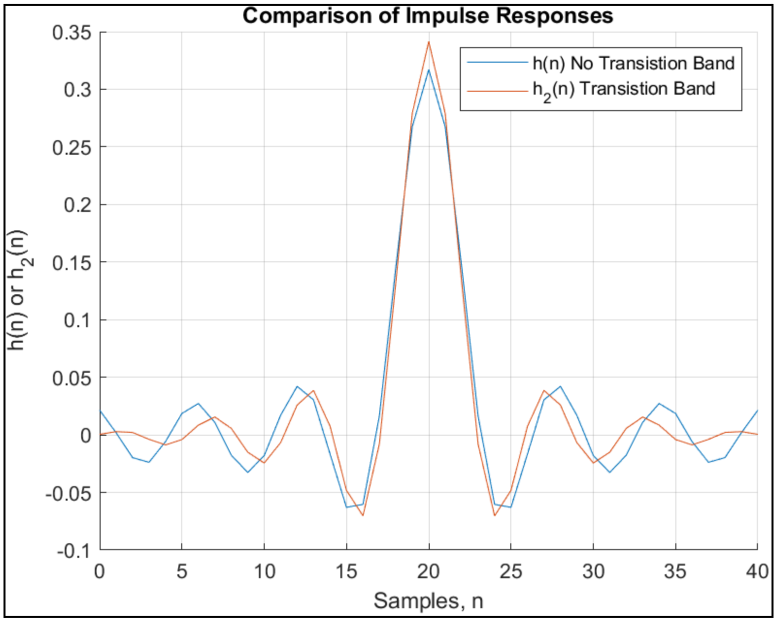
- H_max_passband_error = 1.1265 - 1 = 0.12665
- H_max_stopband_error = db2mag(-15.8293) = 0.1616

Comparing Current Filter to a Filter with Slower Transition

For a slower transition, we add an additional point after our last 1 (and before our first 1 on the right side) of 0.5 in our desired magnitude response.



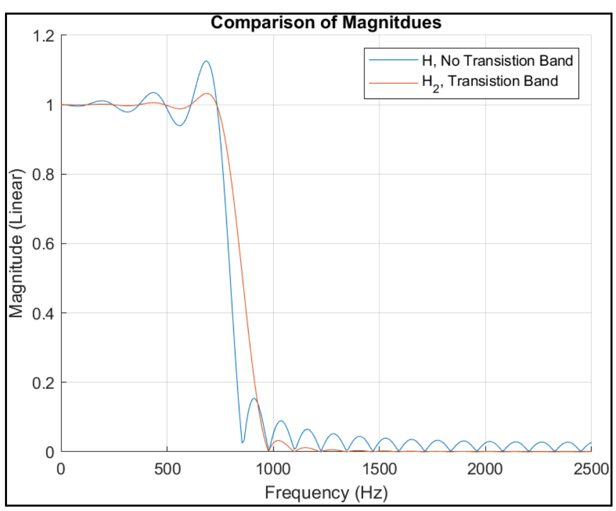
The overall number of 1's remain the same, but the two 0.5's mean that the filter becomes wider by two points, and the transition is less sharp. The transform of this in samples means that the impulse response will become slightly sharper, and reduce the amplitudes of the ripples of the sinc behavior.

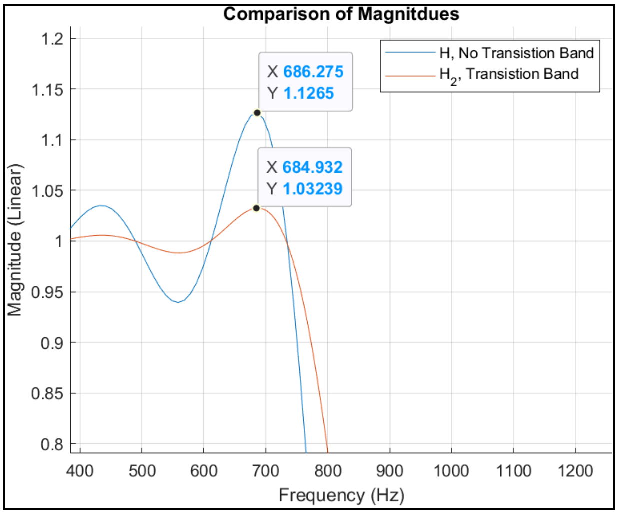


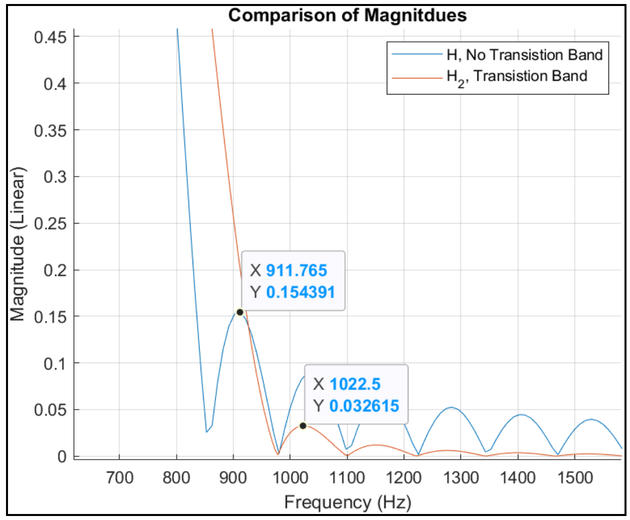
We see that by comparing the impulse response after taking the FFT, that the predictions above occurred! The response with the transition band is sharper and has a higher peak amplitude, and the ripple of the sinc behavior has lower amplitudes. Interestingly, the ripple becomes closer to the main peak.

Comparing Max Passband and Stopband Errors

With the new transition band, we want to compare the max passband and stopband error to our previous magnitude frequency response. We use linear scale for easier comparison.







We see that the max error in both the passband and stopband in our filter with transition band is significantly reduced, which is beneficial to obtain a more ideal low pass filter. In addition, the passband overall has values closer to 1 and with less ripple, meaning it is flatter to 1. The sidelobes have significantly lower amplitudes, meaning they are flatter to 0. Is it appropriate to say that one of the sidelobes in H_2 got squeezed into the transition band (Part 3 hint)?

Optimizing the Transition Sample Value

Currently, the transition value is set at 0.5. Changing the transition value from 0.1 to 0.9, should cause a change in the max error. The table below shows transition value vs. max error at or beyond 1000Hz, with the highlighted value giving the lowest error.

Transition Band Value	Largest Error for F ≥ 1000Hz
0.1	0.0655571
0.2	0.0415017
0.3	0.0190492
0.4	0.00961277
0.5	0.032615
0.6	0.0566924
0.7	0.0809333
0.8	0.105174
0.9	0.129415

Part 2: Equiripple Filter Design by Parks-McClellan

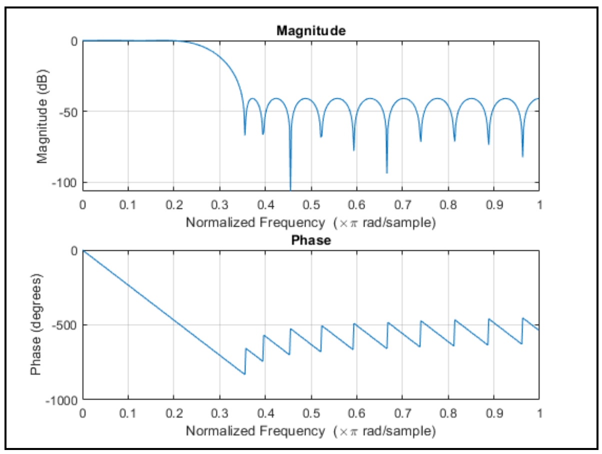
Filter Specification and Inputs:

- Sampling Frequency = 1000 Hz
- Low pass response, cutoff 100 Hz
- Allowable passband distortion (ripple) is ±2%, i.e. $\delta 1 = 0.02$
- Above 175 Hz, attenuation of at least 40 dB, i.e. $20 \log10(\delta 2) = -40$

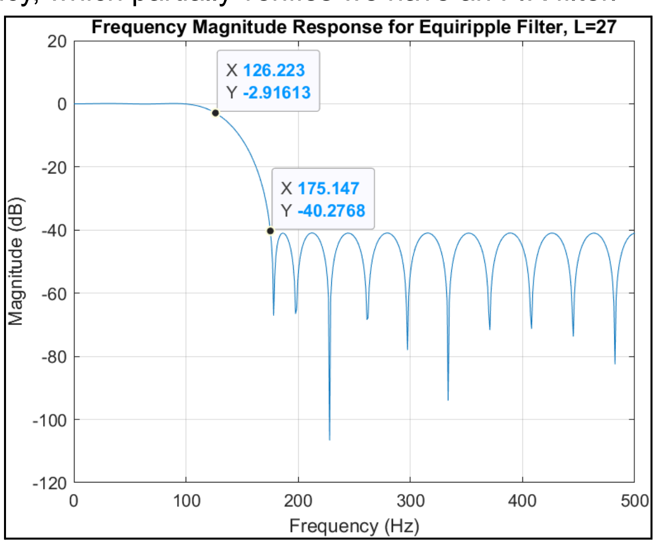
Determine Filter Length

Using the following estimate from the project page, I got a length of 24.2424. However, as filter lengths must be integers and we want a lowpass response (need a point at F=0Hz that's 1), we let the filter length be 25.

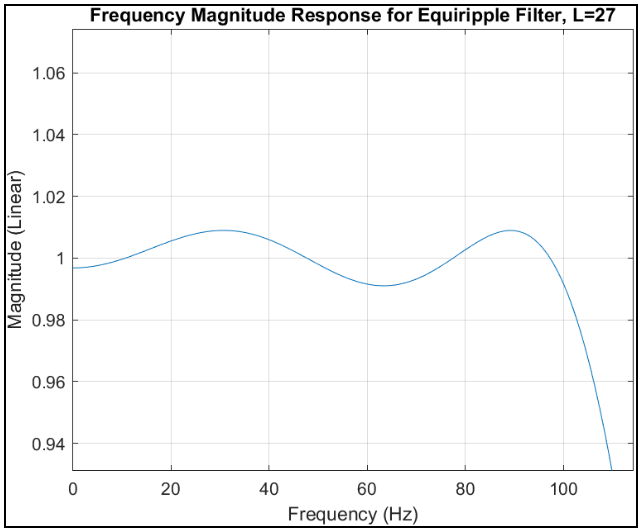
Unfortunately, when the length is 25, the attenuation above 175 Hz is below the 40dB requirement. Therefore, we try the next length 27 (can't be 26 as it's even, DC values will not be sampled).



We see that when running freqz, the frequency response of our filter, that we have linear phase up to the cutoff frequency, which partially verifies we have an FIR filter.



Upon closer inspection of the magnitudes from our frequency response, we see that we closely meet our cutoff frequency of 100 Hz, as at -3dB, it is 126.223 Hz. The attenuation above 175 Hz is also correct as it is -40.2768 dB, which is at least -40dB.



To also verify that the passband distortion is within limits, we can see that when the magnitude is in linear scale, that it never goes above 1.02 or below 0.98. Therefore, as we have met our lowpass response, cutoff frequency, passband distortion, and attenuation requirements, our equiripple filter is complete! Hurray!

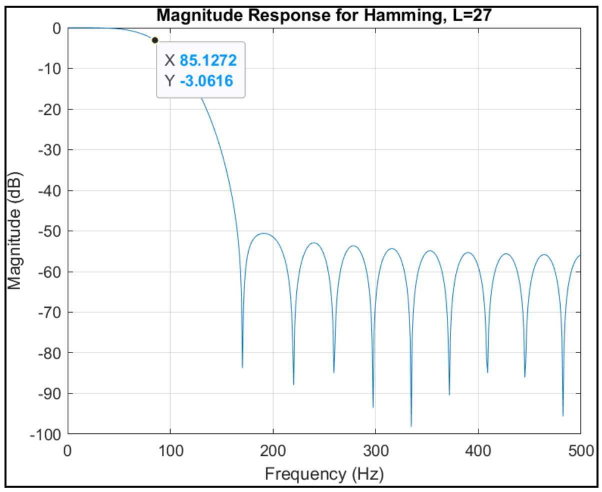
Part 3: Filter Design by Windowing

We want to design a filter by windowing technique, with the same specifications as before.

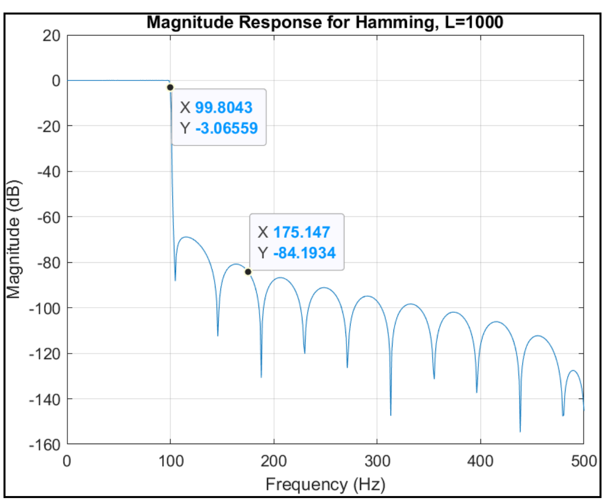
Filter Specification and Inputs:

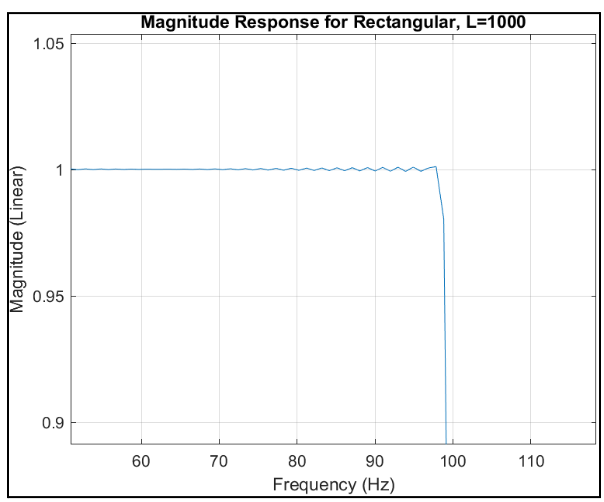
- Sampling Frequency = 1000 Hz
- Low pass response, cutoff 100 Hz
- Allowable passband distortion (ripple) is ±2%, i.e. $\delta 1 = 0.02$
- Above 175 Hz, attenuation of at least 40 dB, i.e. $20 \log 10(\delta 2) = -40$

Hamming FIR Window



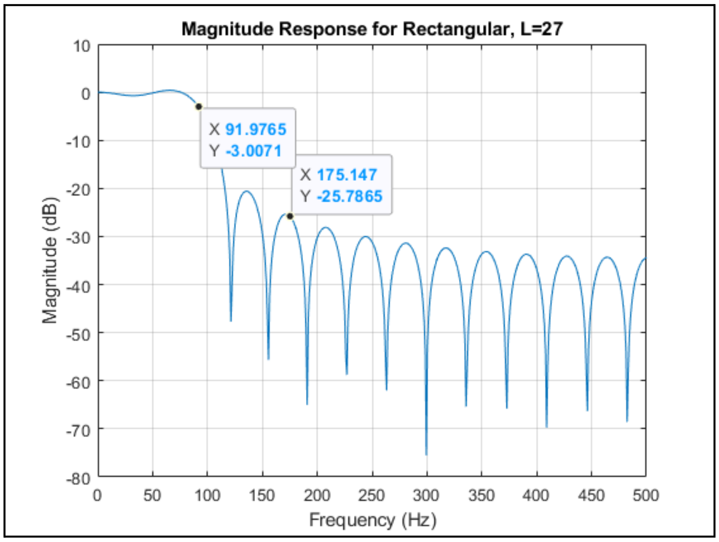
At L = 27 like in the previous part, the frequency cutoff is not met as at -3dB, it is 85Hz, not 100Hz. To find the filter length, I essentially kept adjusting the length until the filter matched the spec.



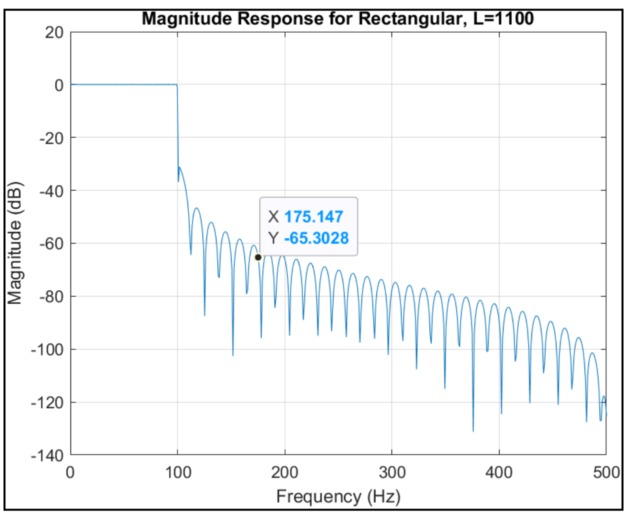


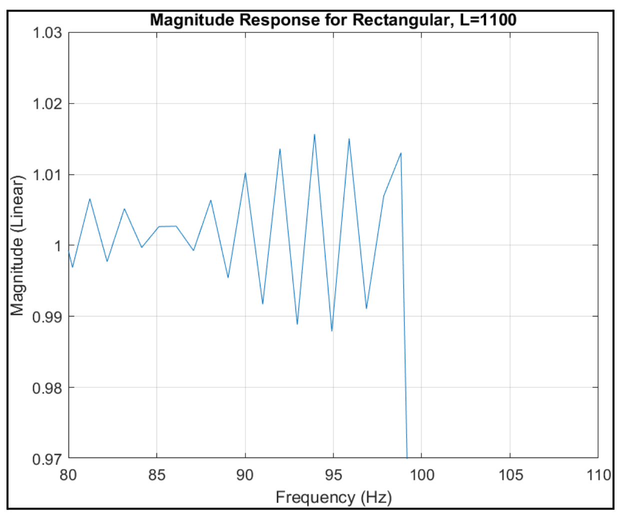
We see that for L=1000, that the cutoff frequency is at 99.8 Hz which for me is close enough to 100Hz. The passband distortion is far below 0.02, and the attenuation above 175 Hz is also far below -40dB. Therefore, the filter designed with Hamming window is up to spec. However, this is likely far from the minimum length required as this was done by hand and via loops and conditionals.

Rectangular FIR Window
We repeat the same process but use a rectangular window.



At L = 27, hardly any of our spec requirements are met. We must now adjust L to meet spec.





For L = 1100, the cutoff frequency is also at 99.8 Hz (not labeled, sorry):) and the attenuation at 175 Hz is far below -40dB, now at -65dB. The passband distortion is also within 0.02 of 1, which is acceptable. This took a lot of attempts to get the distortion down, as increasing the length beyond a certain point increases the distortion, and too low also increases the distortion.

Finally, the filter designed with a rectangular window is up to spec! Likewise, this filter length is likely not the minimum required to meet spec.

Estimate Percent Savings in Runtime of Equiripple vs. Windowed
In the equiripple-designed filter, the length was 27, whereas the lowest length for the window-designed filter was 1000.

The window-designed filter is about 1000/27 = 37 times the length of the equiripple-designed! This means that in terms of samples alone, the window-designed is 37 times or 3700% slower in runtime! However, it is clear in their tradeoffs why one would choose one method over the other.

Equiripple will take more work to set up, but requires less samples and thus is faster in runtime. Windowed requires far less work to create but then requires far more samples and is then slower in runtime.

Appendix

part1.mlx

```
% Parameters
Fs = 5000; % in Hz
L  = 41;
Fc = 750; % Cut off freq. in Hz

% Evenly spaced frequencies for F
F = (0:L-1) * (Fs / L);

disp(F);
length(F)

% Magnitude response
Hmag = F < Fc; % Set Hmag to 1 where F < 750 Hz, otherwise 0
% Conjugate-symmetric frequency response
% Filter is low pass, need to include negative!
% 6 points on right side, would be shifted to neg.
Hmag = Hmag + (F > (Fs - Fc));

% Plot the magnitude response
figure;
stem(F, Hmag, 'filled');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Desired Magnitude Response, H_{mag} vs. F');
grid on;
```

```
% Shift in time to let Hphase be causal
M = (L-1)/2;

% Discrete frequency step
dw = 2*pi / L;

% Phase vector using continuous-time frequency (used method 2)
% F is from above, F = (0:L-1) * (Fs / L);
Hphase = exp(-1j * M * 2 * pi * F / Fs);

% Plot the phase response
figure;
plot(F, angle(Hphase));
xlabel('Frequency (Hz)');
ylabel('Phase (rad)');
title('Desired Phase Response');
grid on;
```

```
H = Hmag .* Hphase
h=ifft(H)
hreal = real(h);
```

```
n = 0:1:L-1;
figure;
stem(n, hreal, "filled");
xlabel('Samples, n');
ylabel('h(n)');
title('Filter Coefficients for LPF, h(n)');
grid on;
```

```
% FFT to get samples in time to frequency!
Hplot = fft(h, 512);
Fplot = (0:512-1) * (Fs / 512);

% Magnitude Response
Hplot_mag = abs(Hplot);

% Phase Response
Hplot_phase = angle(Hplot);

% % Plot magnitude response of filter.
% figure;
% subplot(2, 1, 1);
% plot(Fplot, mag2db(Hplot_mag));
% xlabel('Frequency (Hz)');
% ylabel('Magnitude (dB)');
% title('Magnitude Response of the Filter');
% grid on;
%
% % Plot phase response.
% subplot(2, 1, 2);
% plot(Fplot, unwrap(Hplot_phase)); % Unwrap phase for continuity
% xlabel('Frequency (Hz)');
% ylabel('Phase (radians)');
% title('Phase Response of the Filter');
% grid on;
% hold off;

% Better to just use built in functions...
figure;
freqz(h);
ax = gca;
```

```
freqzchart = ax.Children(1);
% Can't quite get to -3db, but this is the closest
% [0.308593750000000,-2.956845442494000]
datatip(freqzchart,0.308593750000000,-2.956845442494000);
```

```
% Plot Filter Mag in dB.
figure;
plot(Fplot, mag2db(Hplot_mag));

xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
title('Zoomed Plot with Max Passband Error');
grid on;

% Find max passband error and annotate
[H_max_passband_error, Hmpbe_error_index] = max(Hplot_mag)

ax = gca;
chart = ax.Children(1);
% datatip(chart,Hmpbe_error_index,H_max_passband_error);
datatip(chart, 683.593750000000,0.949226504919851);

xlim([-537 5463])
ylim([-41 23])

xlim([-6 1022])
ylim([-7.8 1.8])
```

```
% Plot Filter Mag in dB
figure;
plot(Fplot, mag2db(Hplot_mag));

xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
title('Zoomed Plot with Max Stopband Error');
grid on;

% Find max stopband error and annotate (clicked on).
ax2 = gca;
chart2 = ax2.Children(1);
datatip(chart2,908.2,-15.83);

xlim([204 3500])
ylim([-38.0 -7.0])
```

```
% Find the index for the transistion band.
% index_lastone_Hmag = find(Hmag == 1, 1, 'last');
index_transition_pos = 8
index_transition_neg = L - 6

Hmag2 = double(Hmag);
Hmag2(1, index_transition_pos) = 0.5;
Hmag2(1, index_transition_neg) = 0.5;


% check Hmag2 - its proper
figure;
stem(F, Hmag2, 'filled');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('H_{mag2} with Transition Band');
grid on;
```

```
% No change to Hphase. Therefore, Hphase is same.
H2 = Hmag2 .* Hphase;
h2=ifft(H2);
h2real = real(h2);
```

```
% Coefficient in denominator is 1 as signal is real, no poles.
H2plot = freqz(h2real, 1, 512, 5000);

% Show plot
freqz(h2real, 1, 512, 5000)

% Find max passband and stopband errors
H2plot_mag = abs(H2plot);

[H2_max_passband_error, H2mpbe_index] = max(H2plot_mag)

half_length = 512/2;
Fplot_halved = linspace(0, Fs/2, length(Hplot_mag(1:half_length)));
Fplot2 = linspace(0, Fs/2, 512);

figure;
hold on;
plot(Fplot_halved, Hplot_mag(1:half_length));


plot(Fplot2, H2plot_mag);
xlabel('Frequency (Hz)');
```

```
ylabel('Magnitude (Linear)');
title('Comparison of Magnitdue Responses');
grid on;
ax3 = gca;
chart3 = ax3.Children(1);
hold off;
legend('H, No Transistion Band', 'H_{2}, Transistion Band');
```

```
Hmag3 = double(Hmag2);
transition_value = 0.9;
Hmag3(1, index_transition_pos) = transition_value;
Hmag3(1, index_transition_neg) = transition_value;
H3 = Hmag3 .* Hphase;
h3=ifft(H3);
H3plot = freqz(real(h3), 1, 512, 5000);
plot(Fplot2, abs(H3plot));
xline(1000, '--r')
grid on

xlim([877 1271])
ylim([0.000 0.133])

% Transition Band Value | Largest Error for >= 1000Hz
% 0.1                   | 0.0655571
% 0.2                   | 0.0415017
% 0.3                   | 0.0190492
% 0.4                   | 0.00961277
% 0.5                   | 0.032615
% 0.6                   | 0.0566924
% 0.7                   | 0.0809333
% 0.8                   | 0.105174
% 0.9                   | 0.129415
```

part2.m

```
% EE125 Project 6 - FIR Filter Design
% December 7, 2024
%% Part 2: Equirriple Filter Design by Parks-McClellan
% Design a filter with this spec:
% Fs = 1000Hz, Fc = 100Hz (Lowpass), Ripple is max +/- 2%
% F > 175Hz, attenuation should be at least 40dB.
Fs = 1000;  % Hz
Fc = 100;   % Hz
Fstopband = 175; % Hz
max_passband_distortion = 0.02;
transitionband_width = 175 - Fc; % Hz
```

```
stopband_attenuation = 40; % dB
%% Determine Filter Length
% Given equation for estimation (too low, attenuation too low)
L_est = (stopband_attenuation * Fs) / (22 * transitionband_width);
% Equation 10.2.97? Not necessary, just increment our picked L.
% Needs to be odd, whole for low pass (one point at F = 0Hz)
% L = 25 doesn't work. Stopband Attenuation is not 40dB.
L_pick = 27;
order = L_pick - 1;
% firpm uses normalized frequency points. Need to normalize Fstopband, Fc.
Fc_norm = Fc * (2 / Fs);
Fstopband_norm = Fstopband * (2 / Fs);
% From range [0, 1], where 1 is the Nyquist Frequency (so why we normalize)
freq_pts = [0, Fc_norm, Fstopband_norm, 1];
% Desired amplitudes at frequencies between pairs of points
% So from F = 0 to F = Fc_norm, desired amplitude is 1 for LPF.
desired_amp = [1, 1, 0, 0];
%% Run the filter designer :D
h = firpm(order, freq_pts, desired_amp);
H = freqz(h);
% Plot with the normalized frequencies
f_norm = linspace(0, Fs/2, length(H));
% Error
plot(f_norm, abs(H));
% For dB
% plot(f_norm, mag2db(abs(H)));
xlabel("Frequency (Hz)");
% ylabel("Magnitude (dB)");
ylabel("Magnitude (Linear)");
title("Frequency Magnitude Response for Equiripple Filter, L=27");
grid on
```

part3.m

```
% EE125 Project 6 - FIR Filter Design
% December 7, 2024
%% Part 3: Filter Design by Windowing
% Design a filter with this spec:
% Fs = 1000Hz, Fc = 100Hz (Lowpass), Ripple is max +/- 2%
% F > 175Hz, attenuation should be at least 40dB.
Fs = 1000;  % Hz
Fc = 100;   % Hz
Fstopband = 175; % Hz
max_passband_distortion = 0.02;
stopband_attenuation = 40; % dB
%% Filter Design with Hamming Window, then Rectangular Window
% Need to increase filter L until filter meets desired characteristics.
```

```
L = 1000;
order = L - 1;
Fc_norm = Fc * (2 / Fs);
f_norm = linspace(0, Fs/2, length(H));
% h = fir1(order, Fc_norm, hamming(L));
h = fir1(order, Fc_norm, rectwin(L));
H = freqz(h);
% Cutoff Frequency and Passband Distortion
plot(f_norm, mag2db(abs(H)))
% plot(f_norm, abs(H))
xlabel("Frequency (Hz)");
ylabel("Magnitude (dB)");
% ylabel("Magnitude (Linear)");
% title(['Magnitude Response for Hamming, L=' num2str(L)]);
title(['Magnitude Response for Rectangular, L=' num2str(L)]);
grid on
% xlim([80 110])
% ylim([0.97 1.03])
```