Project I: Filter Design

6.341 Discrete-Time Signal Processing

Jason Chang
Fall 2007

Part A. Lowpass Filter Design

A.1. Understanding Filter Specifications

Definitions, Variables, and Abbreviations
PG – Passband Gain
γ – Minimum Passband Gain
δ – Ripple Magnitude
Δ – Desired Minimum Passband Gain

(a) If you normalize this filter so that the maximum passband gain is unity, what is the new minimum passband gain, γ?

$$\max(PG') = \frac{PG}{\max(PG)} = \frac{1+\delta}{1+\delta} = 1$$

$$\gamma = \min(PG') = \frac{\min(PG)}{\max(PG)} = \frac{1-\delta}{1+\delta}$$

$$\boxed{\gamma = \frac{1-\delta}{1+\delta}}$$

(b) Given a minimum passband gain specification of −Δ dB, express γ in terms of Δ so that the two are equal.

$$\gamma[\text{dB}] = 20\log_{10}(\gamma)\ \text{dB} = -\Delta\ \text{dB}$$

$$\boxed{\gamma = 10^{-\frac{\Delta}{20}}}$$

(c) Express δ in terms of Δ. Use this relation in your design of FIR filters.

$$\gamma = 10^{-\frac{\Delta}{20}}$$

$$\frac{1-\delta}{1+\delta} = 10^{-\frac{\Delta}{20}}$$

$$1-\delta = 10^{-\frac{\Delta}{20}} + 10^{-\frac{\Delta}{20}}\delta$$

$$\delta\left(1+10^{-\frac{\Delta}{20}}\right) = 1-10^{-\frac{\Delta}{20}}$$

$$\boxed{\delta = \frac{1-10^{-\frac{\Delta}{20}}}{1+10^{-\frac{\Delta}{20}}}}$$

A.2. Design and Compare Filters

Design Specifications

1. Passband edge = 0.15π
2. Stopband edge = 0.21π
3. Maximum gain in the passband = 0 dB
4. Minimum gain in the passband = –1 dB
5. Maximum gain in the stopband = –40 dB

Questions per Filter

(a) The order of the filter.

(b) A plot of the magnitude response (in dB) from $\omega = 0$ to $\omega = \pi$. Plot a detail of the magnitude response, focusing on the passband ripple (with linear scale). Plot the group delay (in samples) using grpdelay. (Use subplot to fit the three plots onto the same page for each filter.)

(c) A pole-zero diagram produced with zplane.

NOTE: I used the [Z, P, K] values returned from the filter creation functions from Matlab to avoid roundoff error. As seen in class, a filter implemented in cascade is much less prone to errors, and thus the pole-zero plot will be much more accurate.

(d) A plot of the impulse response using filter and stem for 50 samples. (Use subplot to fit the pole-zero diagram and the impulse response onto the same page.)

(e) A paragraph of your observations and comments about this section.

A.2.1 – Butterworth

The following code was used in Matlab to design the Butterworth Filter.

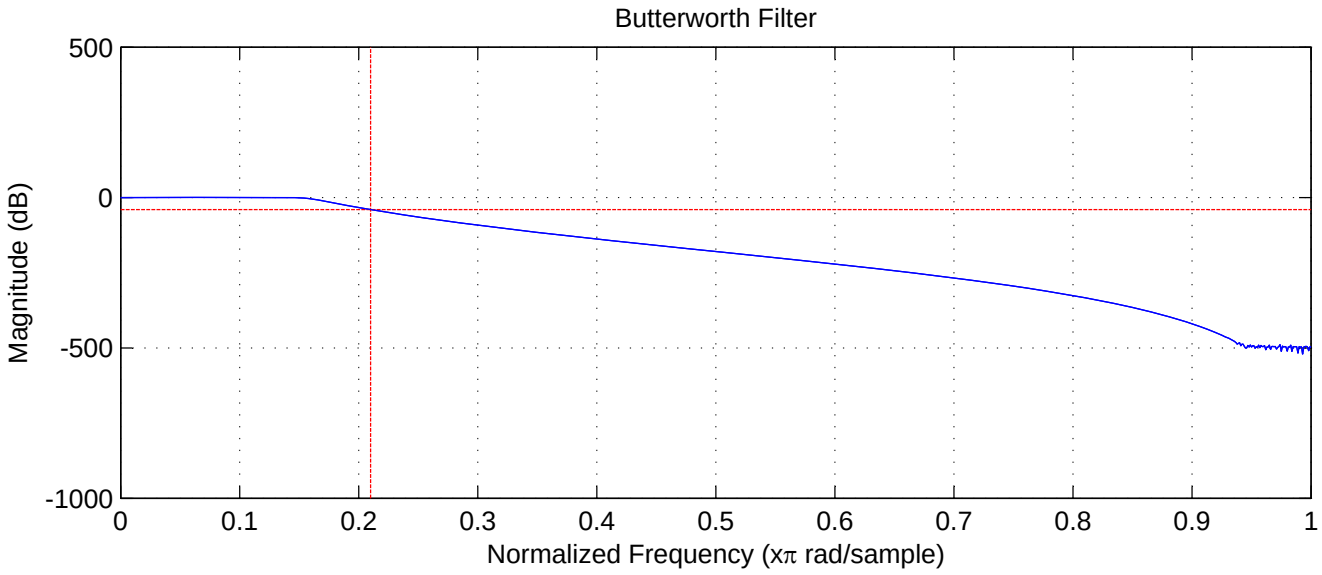
```
% Define the filter parameters
Wp = 0.15;
Ws = 0.21;
Rp = 1;
Rs = 40;

% Estimate the IIR filter order perfectly
[N, Wn] = buttord(Wp, Ws, Rp, Rs);

% Find the filter with the perfectly estimated order
[B, A] = butter(N, Wn);
[Z, P, K] = butter(N, Wn);
```

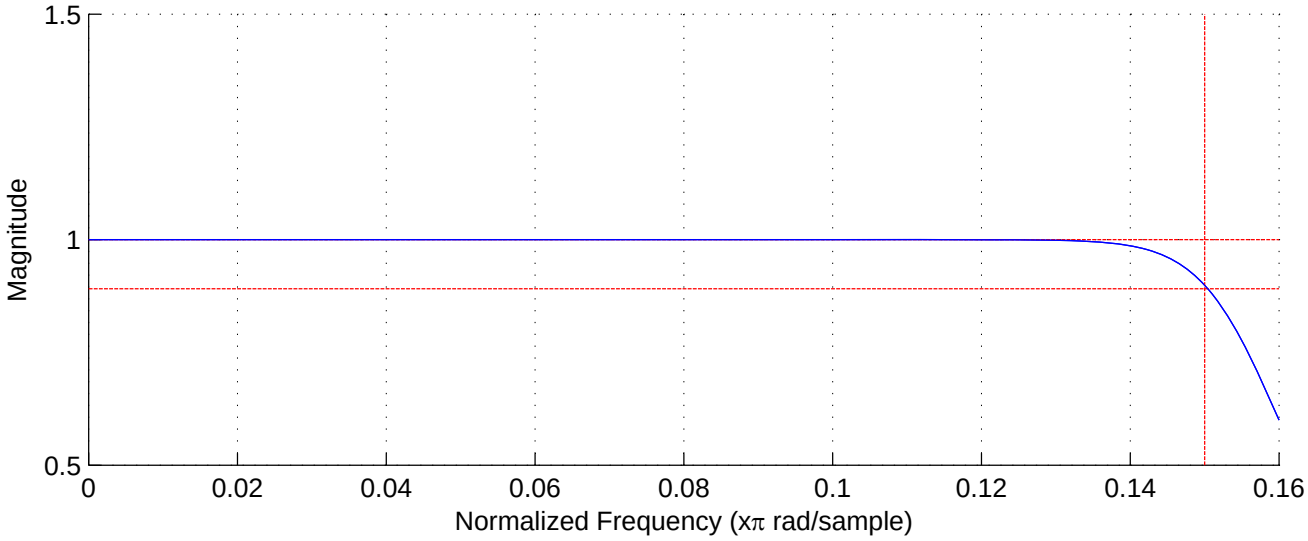
(a) The order of the filter is 15

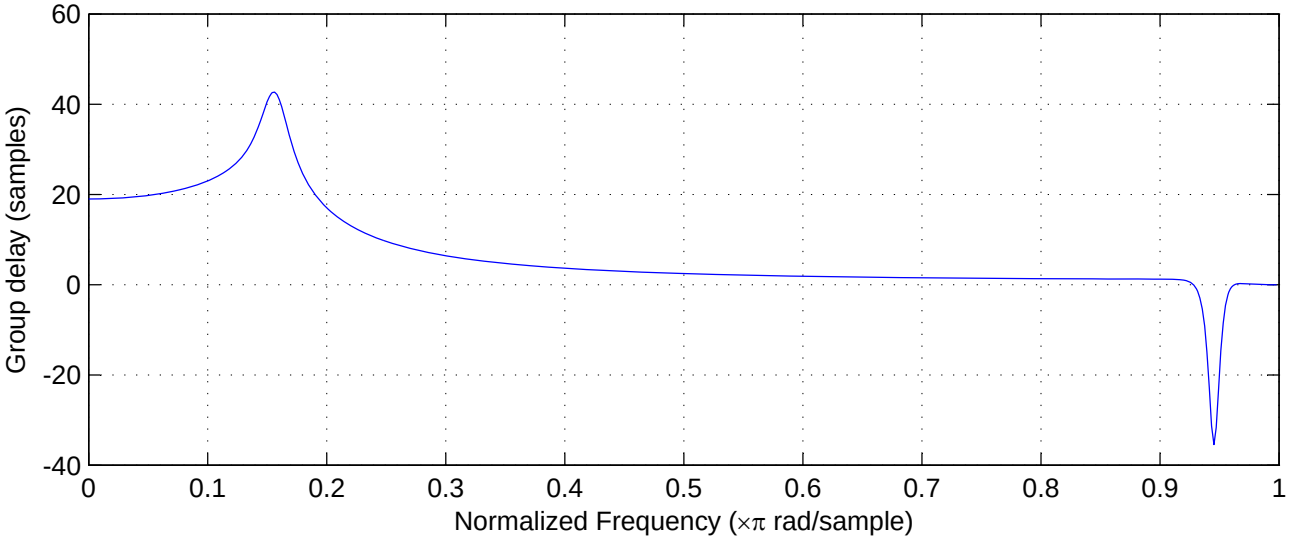
(b-d) Plots Attached

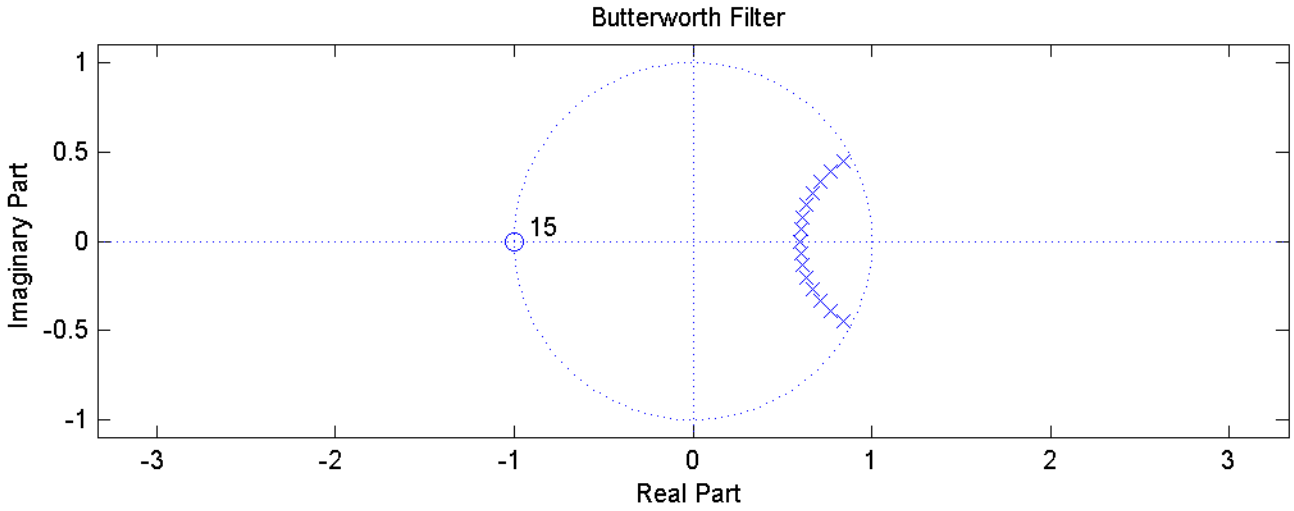
(e) The Butterworth filter has the highest order amongst the IIR filters to achieve the same specifications. The main advantage of using a Butterworth filter is that the passband is very flat. The magnitude plot shows this clearly. However, one undesirable feature of the Butterworth filter is that its transition band does not drop off very quickly compared to the other IIR filters. The main reason for using a Butterworth filter is because of its flat passband and because they are the easiest IIR filters to design. However, as with all IIR filters, the non-linear phase (and thus varying group delay) of the filter is very undesirable. Although the group delay for the Butterworth filter does not change as significantly as some other IIR filters, it is still not very good.

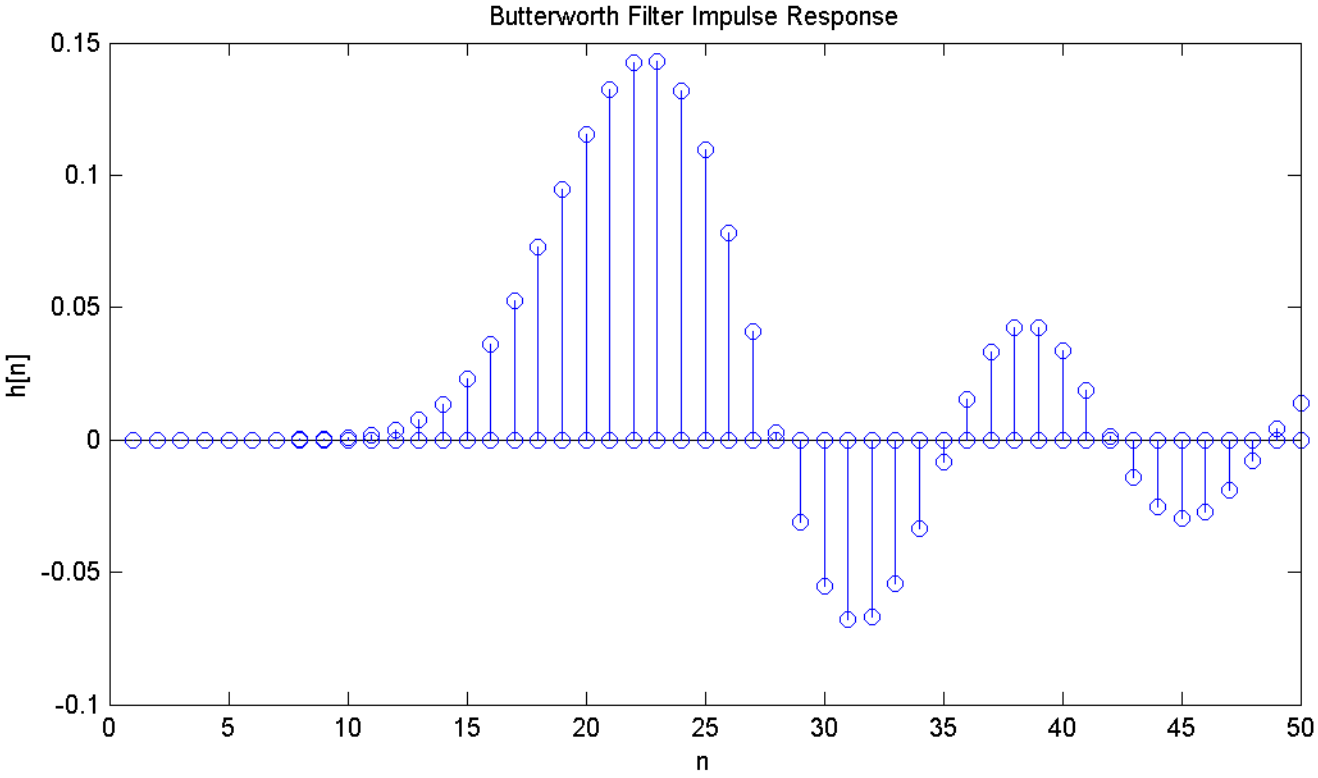
On another note, the pole-zero plot does show that the filter created is indeed a Butterworth filter. Clearly, all the zeroes are located at -1, which is the criterion used when designing a Butterworth filter.


Butterworth Filter
Magnitude (dB)
500
0
-500
-1000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (xπ rad/sample)


Magnitude
1.5
1
0.5
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
Normalized Frequency (xπ rad/sample)


Group delay (samples)
60
40
20
0
-20
-40
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (×π rad/sample)


Butterworth Filter
Imaginary Part
Real Part
15
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3


Butterworth Filter Impulse Response
h[n]
n
0.15
0.1
0.05
0
-0.05
-0.1
0
5
10
15
20
25
30
35
40
45
50

A.2.2 – Chebyshev Type I

The following code was used in Matlab to design the Chebyshev Type I Filter.

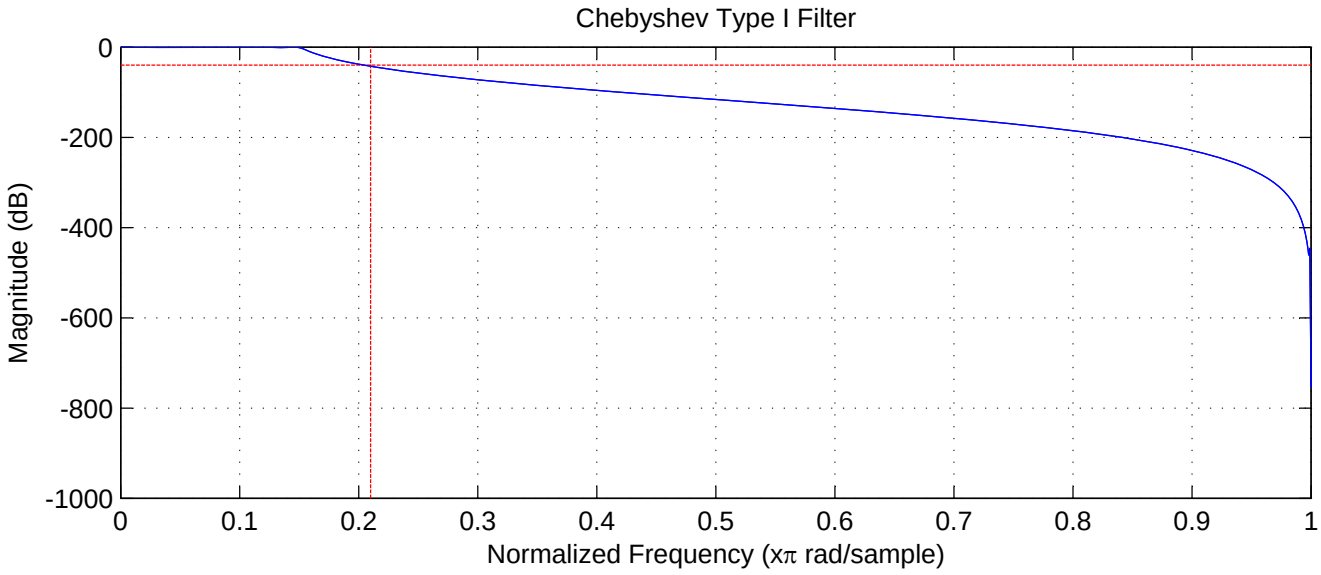
```
% Define the filter parameters
Wp = 0.15;
Ws = 0.21;
Rp = 1;
Rs = 40;

% Estimate the IIR filter order perfectly
[N, Wn] = cheb1ord(Wp, Ws, Rp, Rs);

% Find the filter with the perfectly estimated order
[B, A] = cheby1(N, Rp, Wn);
[Z, P, K] = cheby1(N, Rp, Wn);
```

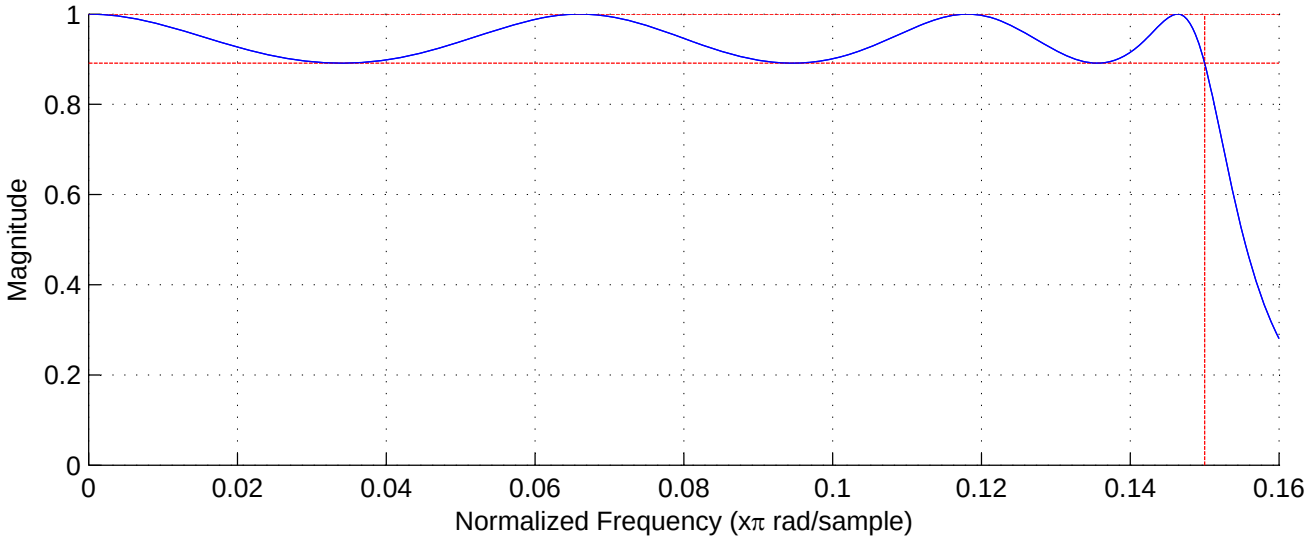
(a) The order of the filter is 7

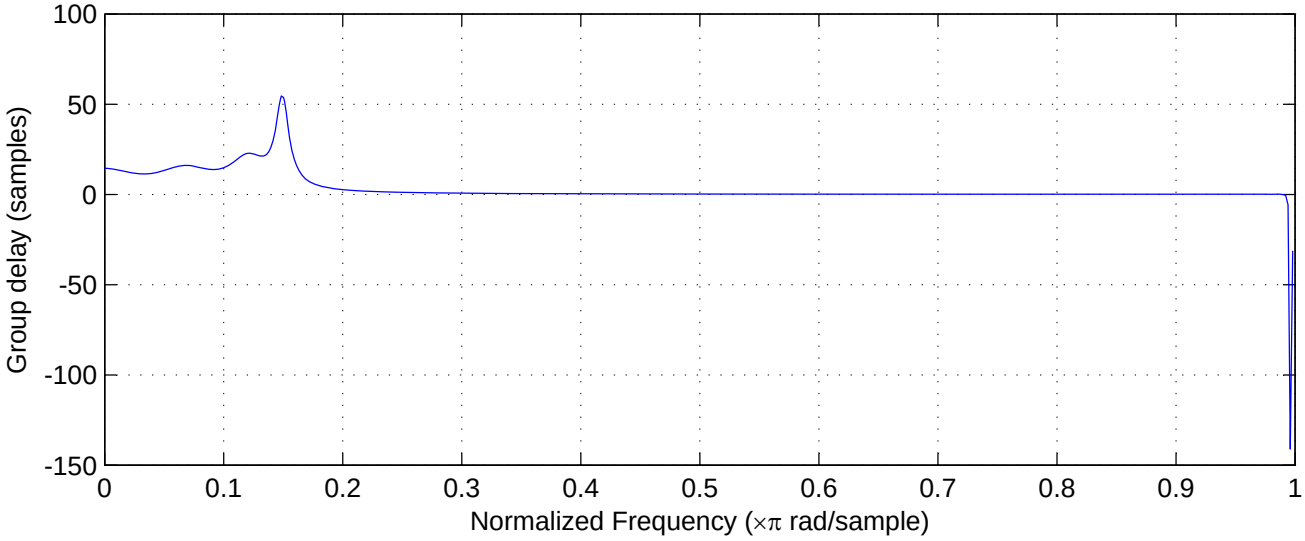
(b-d) Plots Attached

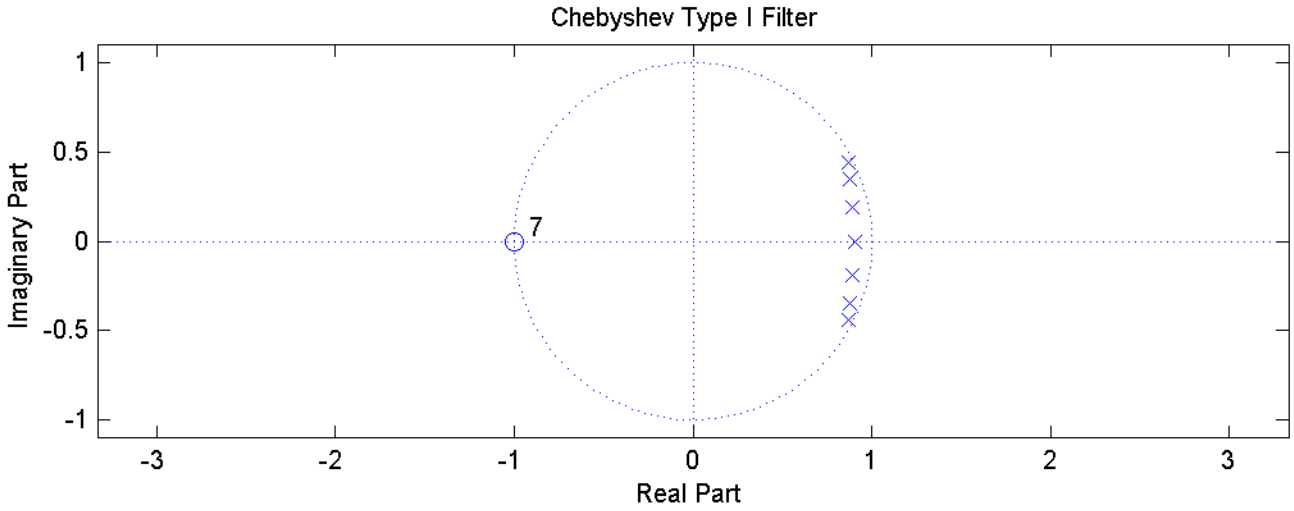
(e) The Chebyshev Type I filter, unlike the Butterworth filter, has ripples in its passband. However, similar to the Butterworth filter, this Chebyshev Type I filter does not have ripples in the stopband. There are a few things that make this filter less desirable then the Butterworth. Obviously, ripples are an undesirable affect if they can be avoided. In addition to the passband ripples, the group delay of this Chebyshev Type I filter in the passband is also much greater (reaching more than a 50 sample delay). By compromising a little in the passband ripples and allowing a bit of group delay, the Chebyshev Type I filter is able to reduce the order of the filter by approximately ½ and still meet the design specifications.

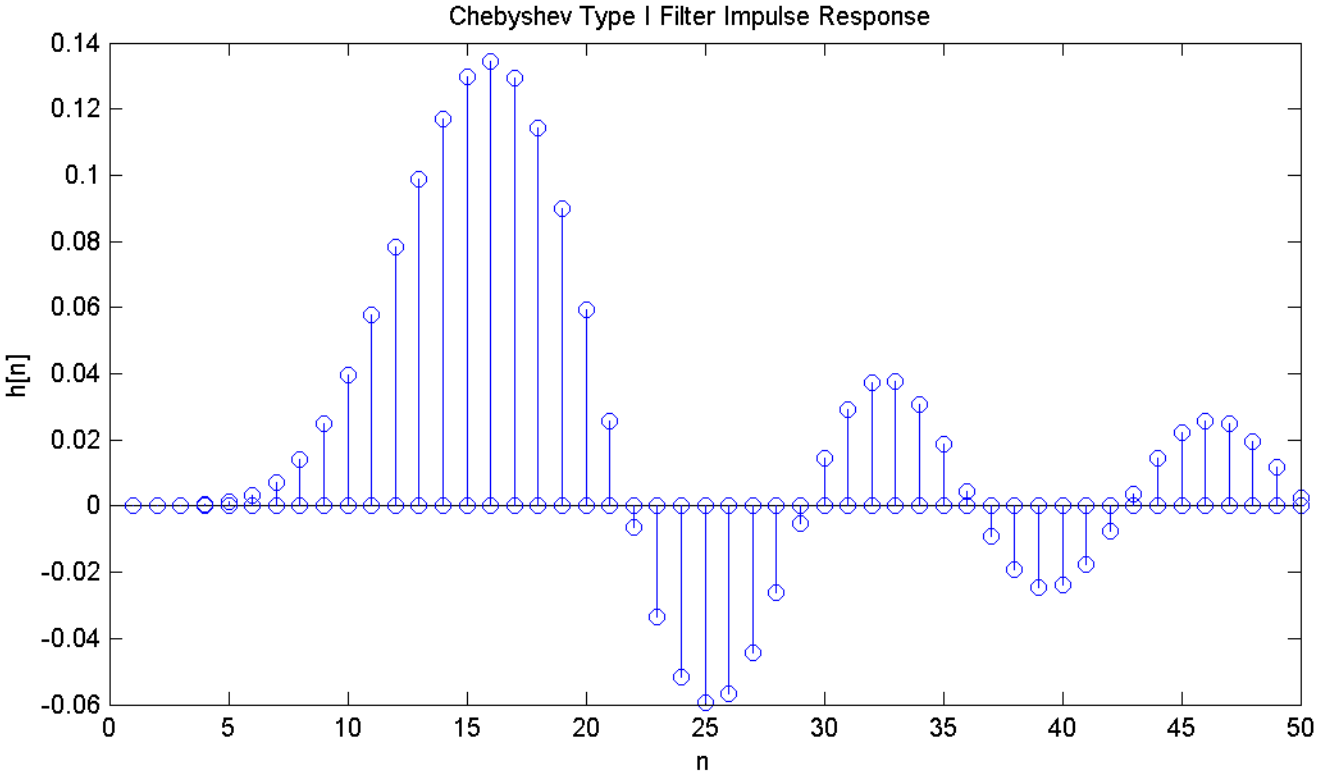
Similar to the Butterworth filter, this filter also places all zeroes at z=-1. However, the poles are chosen more carefully here to reduce the order of the filter as much as possible while still maintaining the desired gains and ripples.


Chebyshev Type I Filter
Magnitude (dB)
0
-200
-400
-600
-800
-1000
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (xπ rad/sample)


Magnitude
1
0.8
0.6
0.4
0.2
0
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16
Normalized Frequency (xπ rad/sample)


Group delay (samples)
100
50
0
-50
-100
-150
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)


Chebyshev Type I Filter
Imaginary Part
Real Part
7
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3


Chebyshev Type I Filter Impulse Response
h[n]
n
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
0
5
10
15
20
25
30
35
40
45
50

A.2.3 – Chebyshev Type II

The following code was used in Matlab to design the Chebyshev Type II Filter.

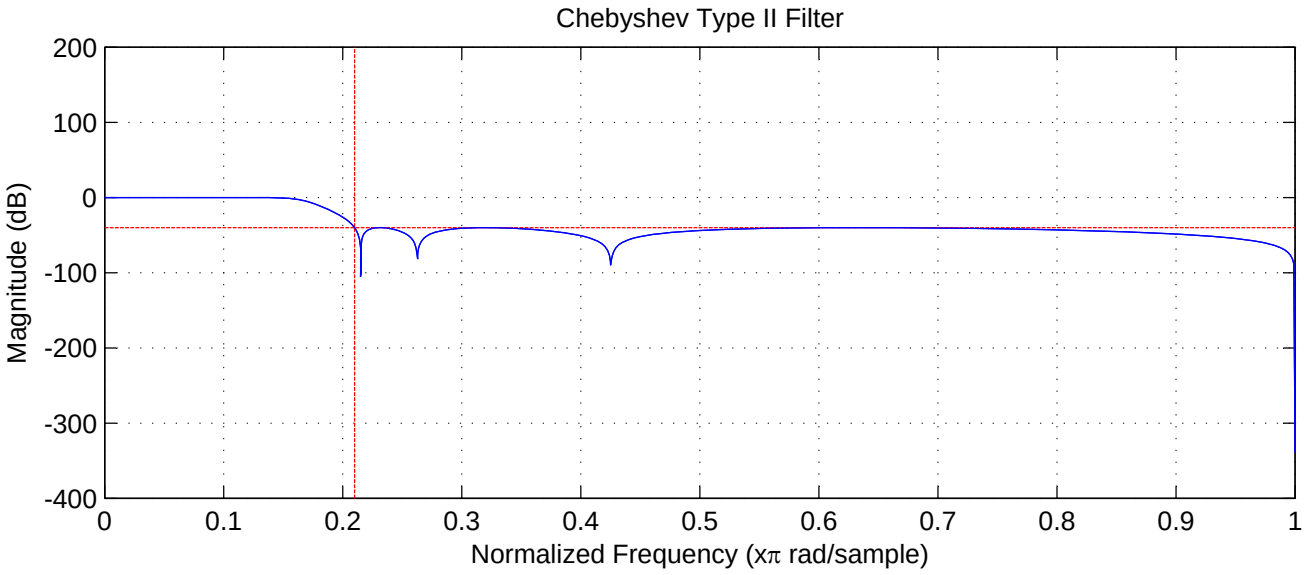
```
% Define the filter parameters
Wp = 0.15;
Ws = 0.21;
Rp = 1;
Rs = 40;

% Estimate the IIR filter order perfectly
[N, Wn] = cheb2ord(Wp, Ws, Rp, Rs);

% Find the filter with the perfectly estimated order
[B, A] = cheby2(N, Rs, Wn);
[Z, P, K] = cheby2(N, Rp, Wn);
```

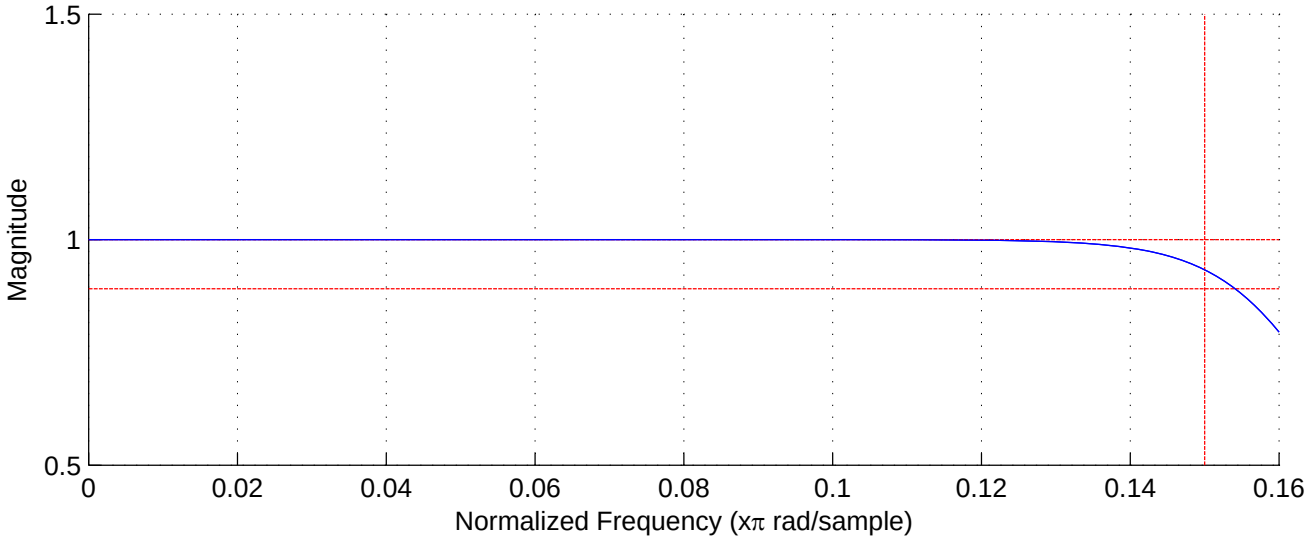
(a) The order of the filter is 7

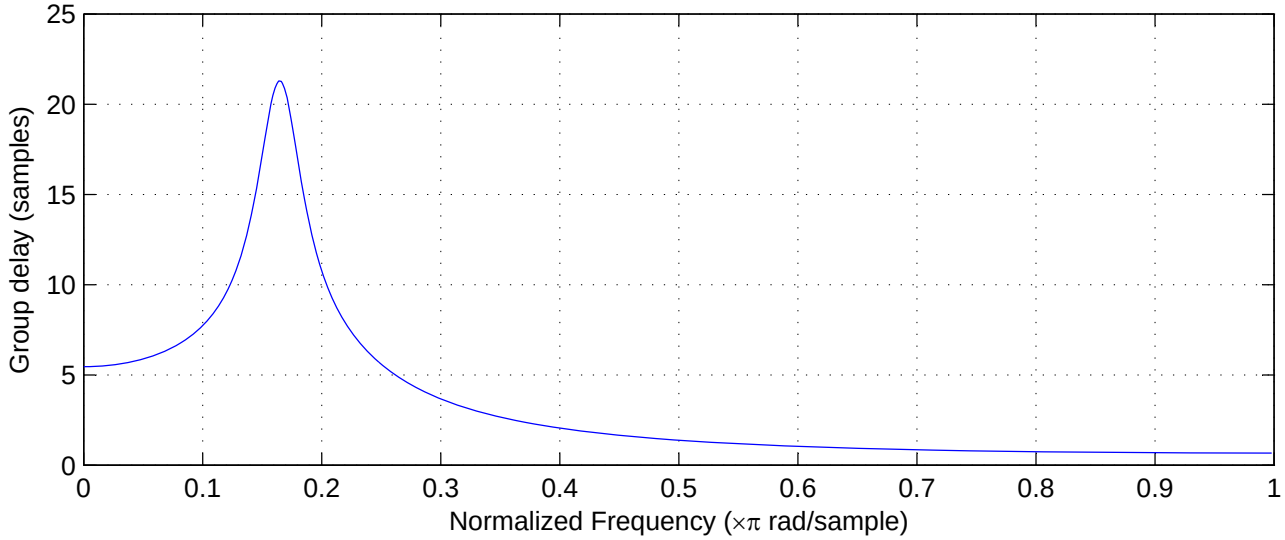
(b-d) Plots Attached

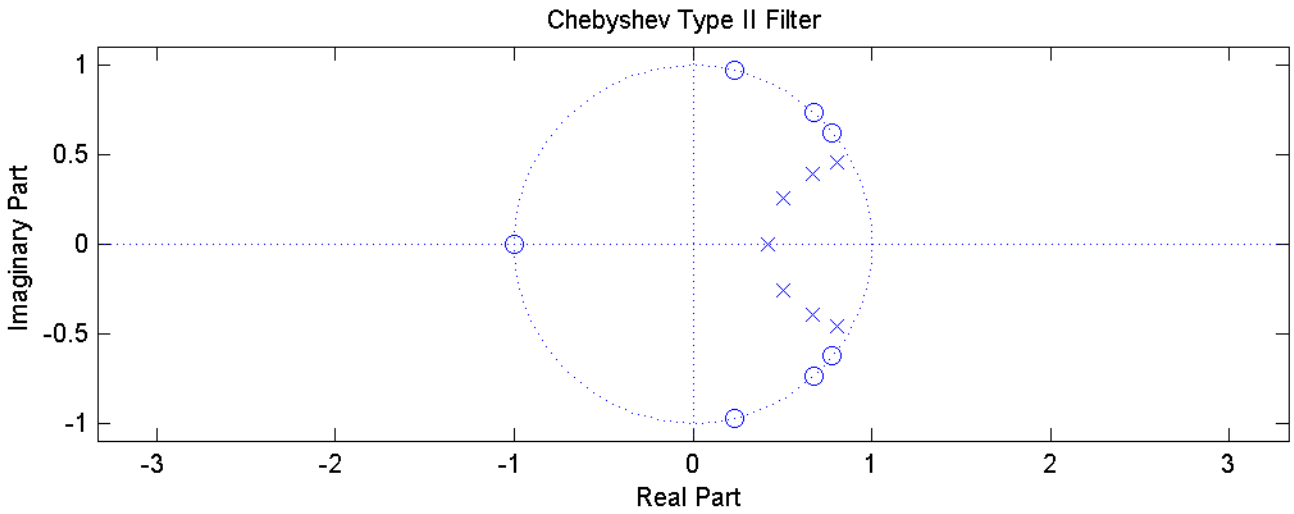
(e) The Chebyshev Type II filter is basically the opposite of the Chebyshev Type I filter. Instead of having ripples in the passband, the Chebyshev Type II filter has ripples in the stopband and a fairly flat magnitude in the passband. It achieves the filter design specifications with the same order filter as the Chebyshev Type I, but with more desirable features. We usually care more about what happens in the passband then the stopband because the signals that we want are in the stopband. Therefore, any filter with a flat passband is very desirable. The only filter that we have designed that is flatter then the Chebyshev Type II was the Butterworth, but it was also twice the order. In addition to this trait, the Chebyshev Type II filter also has the smallest group delay. This is desirable because the less change in group delay a filter has, the less distorted the input signal well get.

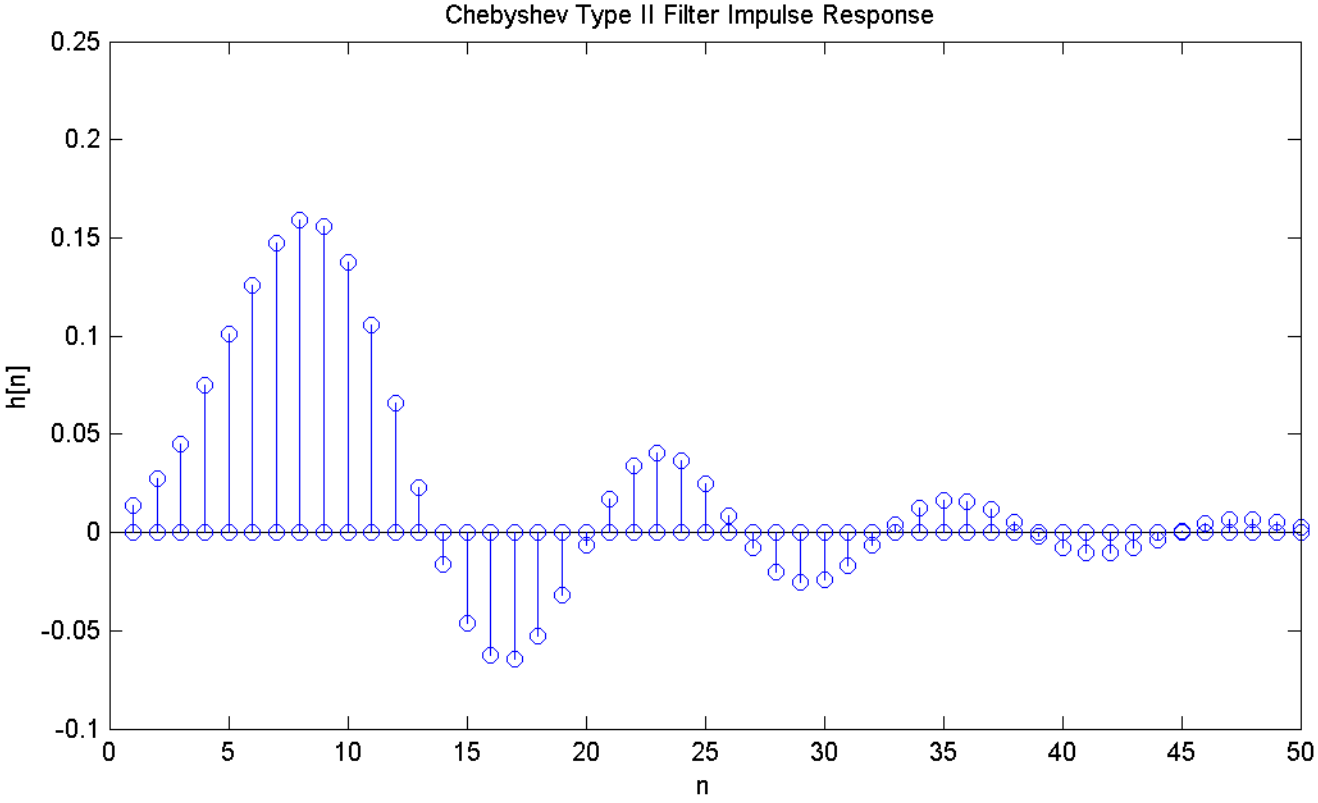
We can see a lot of why the Chebyshev Type II filter does much better then the Butterworth and the Chebyshev Type I filters from the pole-zero plot. As you can see, the zeroes of this filter are not all constrained to be at -1. Therefore, the design method can place zeroes near the unit circle to pull down the magnitude of the filter near the stopband. These zeroes near the unit circle cause the magnitude to ripple near their locations, and thus providing us with a Chebyshev Type II filter.


Chebyshev Type II Filter
Magnitude (dB)
200
100
0
-100
-200
-300
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (xπ rad/sample)


Magnitude
1.5
1
0.5
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
Normalized Frequency (xπ rad/sample)


Group delay (samples)
25
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Frequency (×π rad/sample)


Chebyshev Type II Filter
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3


Chebyshev Type II Filter Impulse Response
h[n]
n
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
5
10
15
20
25
30
35
40
45
50

A.2.3 – Elliptic

The following code was used in Matlab to design the Elliptic Filter.

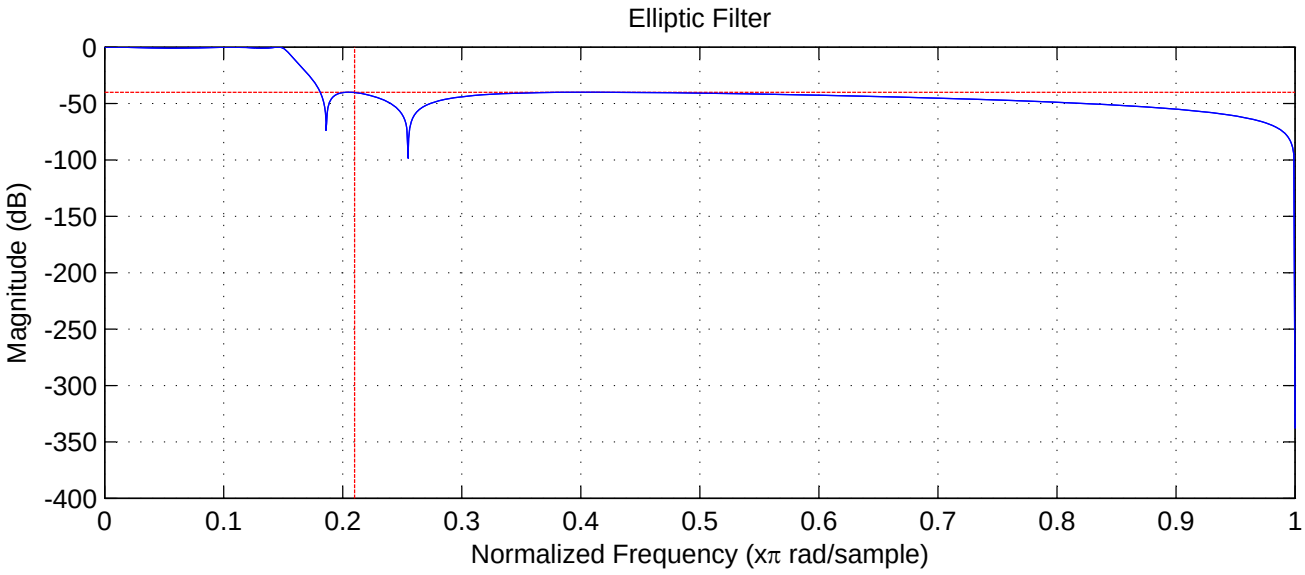
```
% Define the filter parameters
Wp = 0.15;
Ws = 0.21;
Rp = 1;
Rs = 40;

% Estimate the IIR filter order perfectly
[N, Wn] = ellipord(Wp, Ws, Rp, Rs);

% Find the filter with the perfectly estimated order
[B, A] = ellip(N, Rp, Rs, Wn);
[Z, P, K] = ellip(N, Rp, Rs, Wn);
```

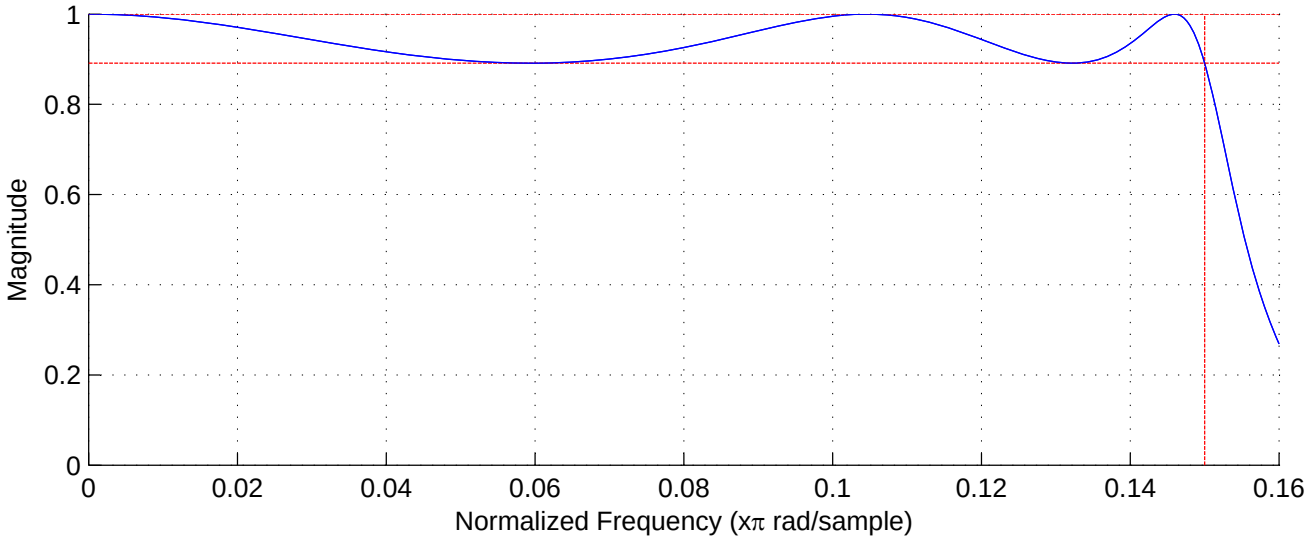
(a) The order of the filter is 5

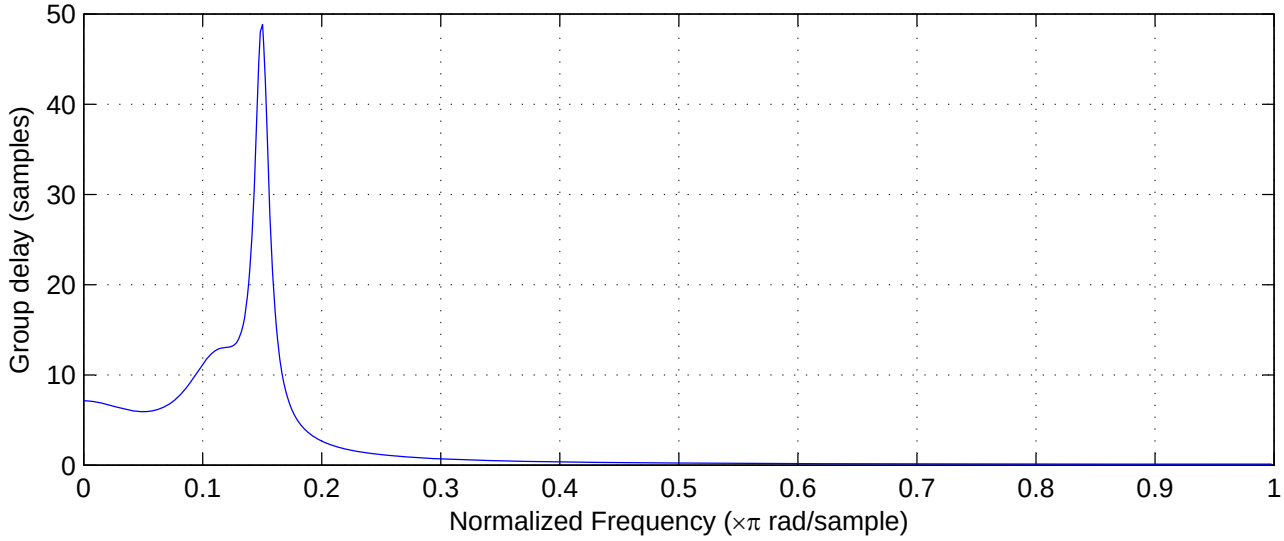
(b-d) Plots Attached

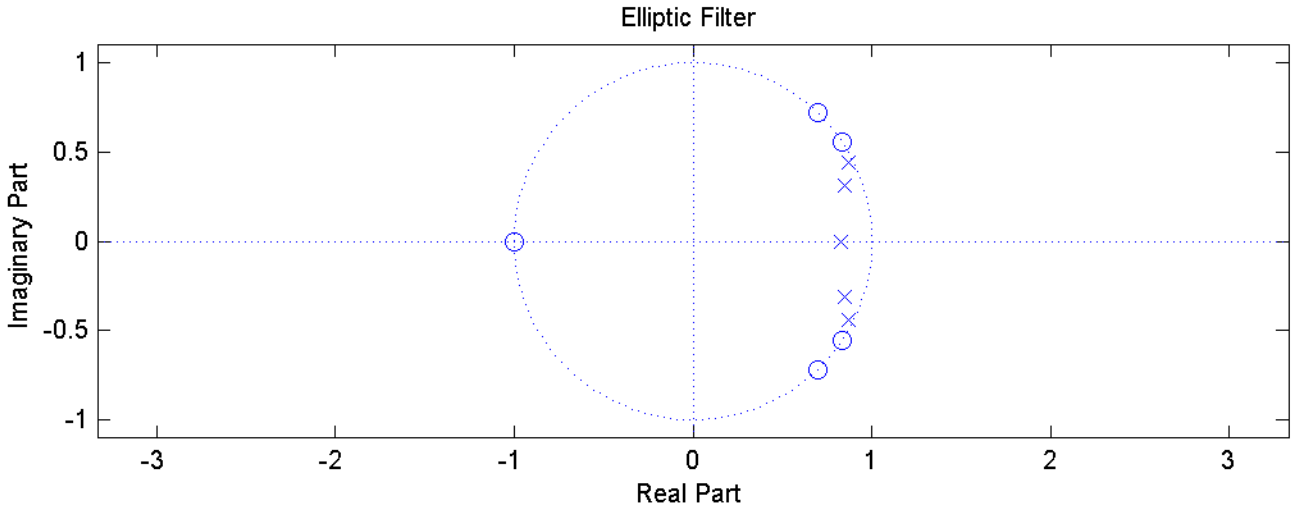
(e) The Elliptic filter is the lowest order IIR filter that we designed that was still able to meet the specifications. The in itself is a great advantage because lower order filters are easier, faster, and cheaper to implement. Unlike the previously designed IIR filters, the Elliptic filter has ripples in both the passband and stopband. In the passband, it uses poles near the unit circle to pull the magnitude up. In the stopband, it uses zeroes on the unit circle to pull the magnitude down at specific points.

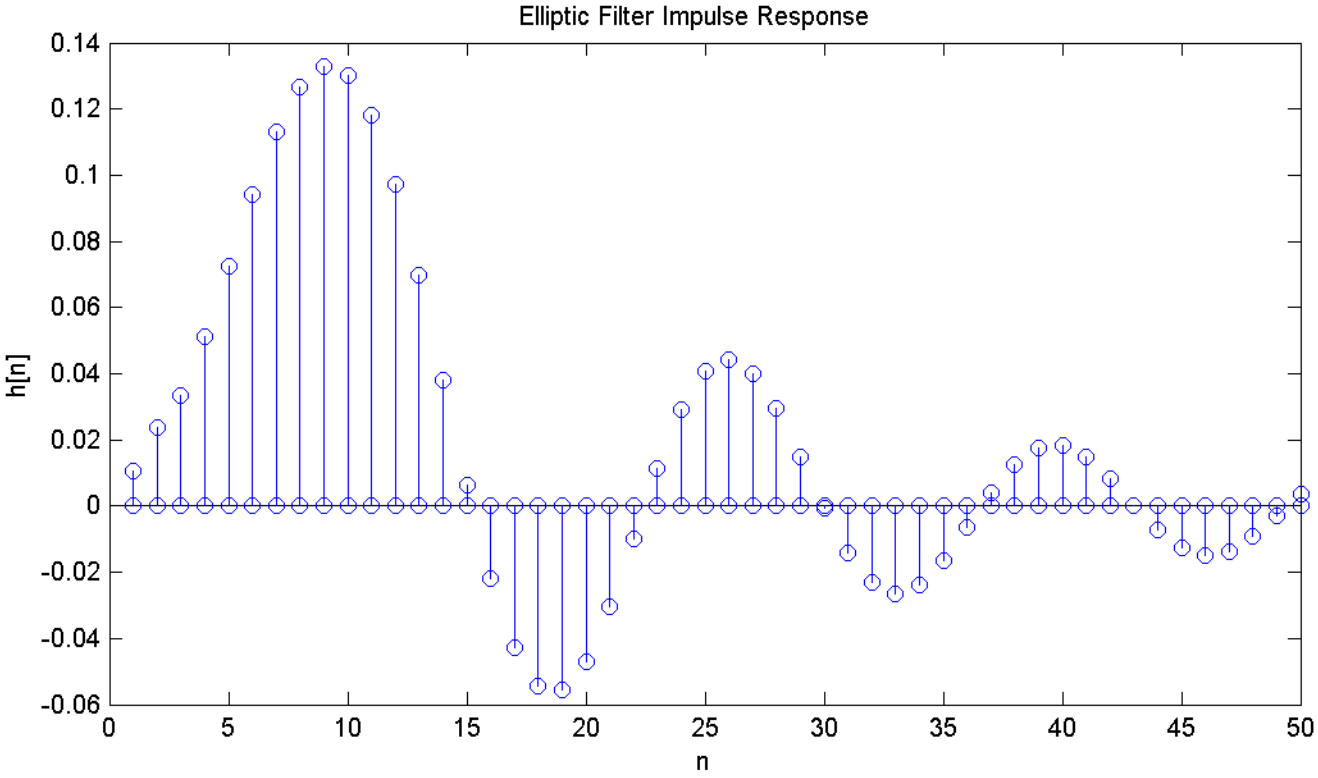
The Elliptic filter is actually the lowest ordered filter that can achieve a set of design specifications. Much like the Parks-Mclellan filters that will be discussed next, the Elliptic filter has equiripple in both the passband and stopband. When a filter has equiripple, it is usually the optimal solution. In the three previous IIR filters, flat passbands or stopbands meant that more optimization could have been done to lower the order and still stay within the design constraints. The Elliptic filter does just that and achieves the design with the lowest order.


Elliptic Filter
Magnitude (dB)
0
-50
-100
-150
-200
-250
-300
-350
-400
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (xπ rad/sample)


Magnitude
1
0.8
0.6
0.4
0.2
0
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16
Normalized Frequency (xπ rad/sample)


Group delay (samples)
50
40
30
20
10
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)


Elliptic Filter
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3


Elliptic Filter Impulse Response
h[n]
n
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
0
5
10
15
20
25
30
35
40
45
50

A.2.4 – Parks-Mclellan Filter

The –1 dB ripple in the passband and the -40 dB attenuation do not produce an equally weighted FIR filter in the two bands. Therefore, we need to calculate the ripple weight constant. First, we need to calculate the ripple magnitude, δ, given that the desired minimum passband gain, Δ, is -1 dB for a normalized filter.

$$\delta_p = \frac{1-10^{-\frac{1}{20}}}{1+10^{-\frac{1}{20}}} = 5.750113\times10^{-2}$$

Next, we calculate the ripple specification for the stopband after normalization.

$$-40\text{ dB} = 10^{-\frac{40}{20}} = 0.01 = \frac{\delta_s}{1+\delta_p}$$

$$\delta_s = 0.01\left(1+\delta_p\right) = 1.057501\times10^{-2}$$

The following code was used in Matlab to design the Parks-Mclellan Filter.

```
% Define the filter parameters
Wp = 0.15;
Ws = 0.21;
Rp = (1-10^(-1/20)) / (1+10^(-1/20));
Rs = (10^-(40/20)) * (1+Rp);

% Estimate the FIR filter order (usually underestimated)
N = firpmord([Wp, Ws], [1, 0], [Rp, Rs]);

% Find the filter with the 'fixed' filter order
B = firpm(N+5, [0, Wp, Ws, 1], [1, 1, 0, 0], [Rs, Rp]);
A = 1+Rp;
```

(a) The order of the filter is 52

(b-d) Plots Attached

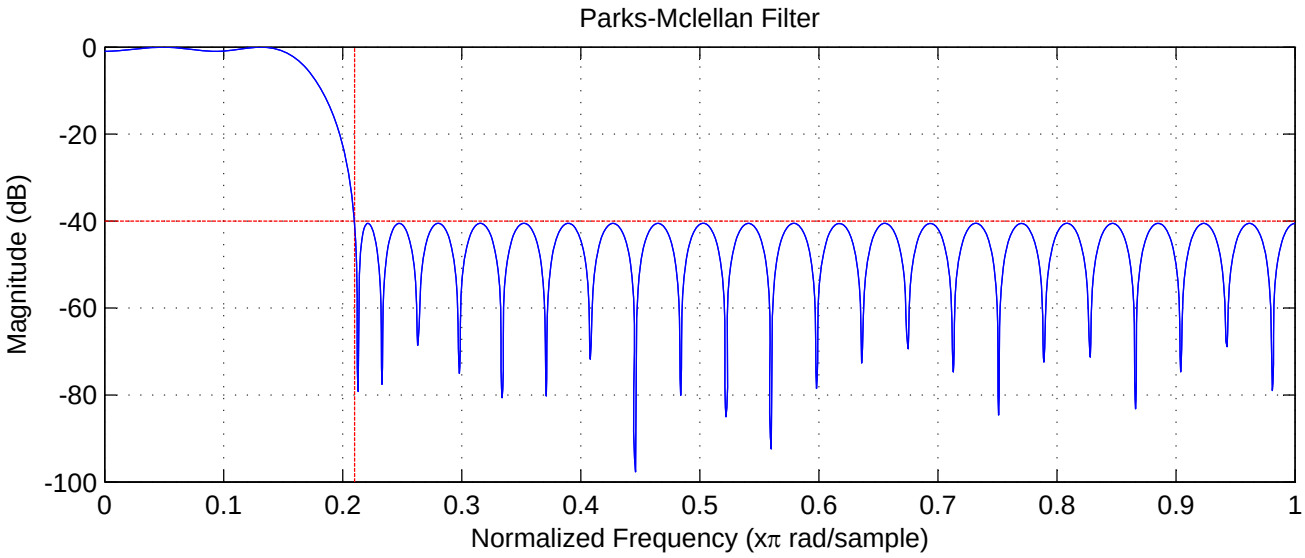
(e) The Parks-Mclellan filter is the most optimal FIR filter possible. In this case, optimal means the lowest order FIR filter that meets all design specifications. Much like the Elliptic filter, the equiripple in the passband and stopband of the Parks-Mclellan filters should indicate to us that it is indeed an optimal solution.

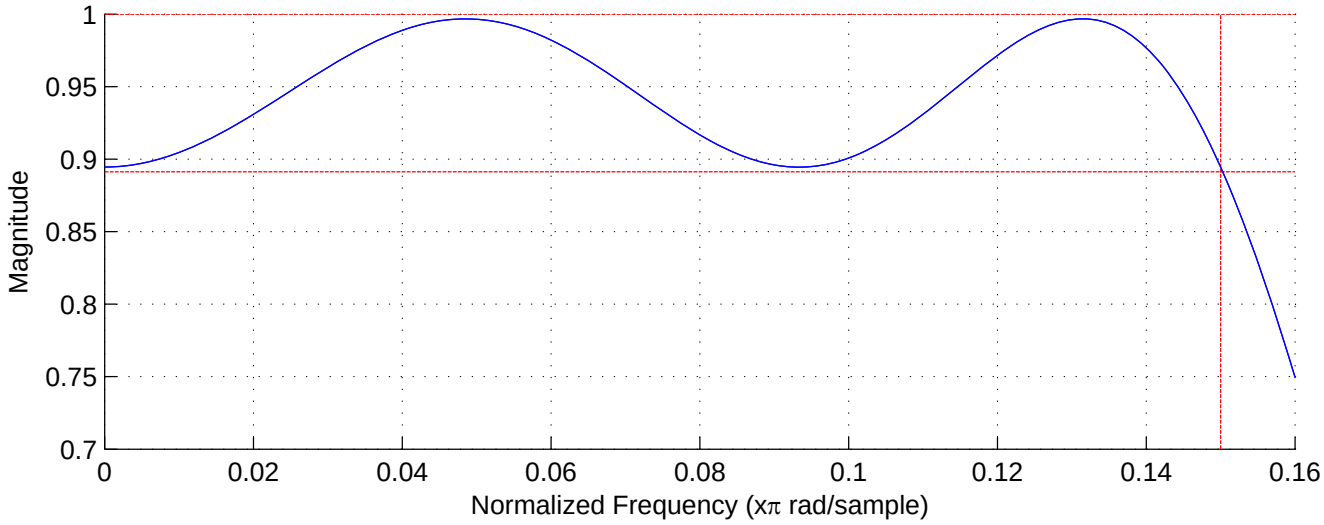
It should be noted here that Matlab's *firpmord* function, which estimates the order needed to achieve filter specifications, underestimated the necessary filter order. In all the previous IIR cases, the filter order estimation functions worked perfectly. This is because Matlab uses the analog equivalent version of these IIR filters and does a bilinear transformation. In the case of FIR filters, no analog equivalent exists, so Matlab must use a less accurate approximation. When desgning Parks-Mclellan filters, the engineer must manually adjust the order until specifications are met.

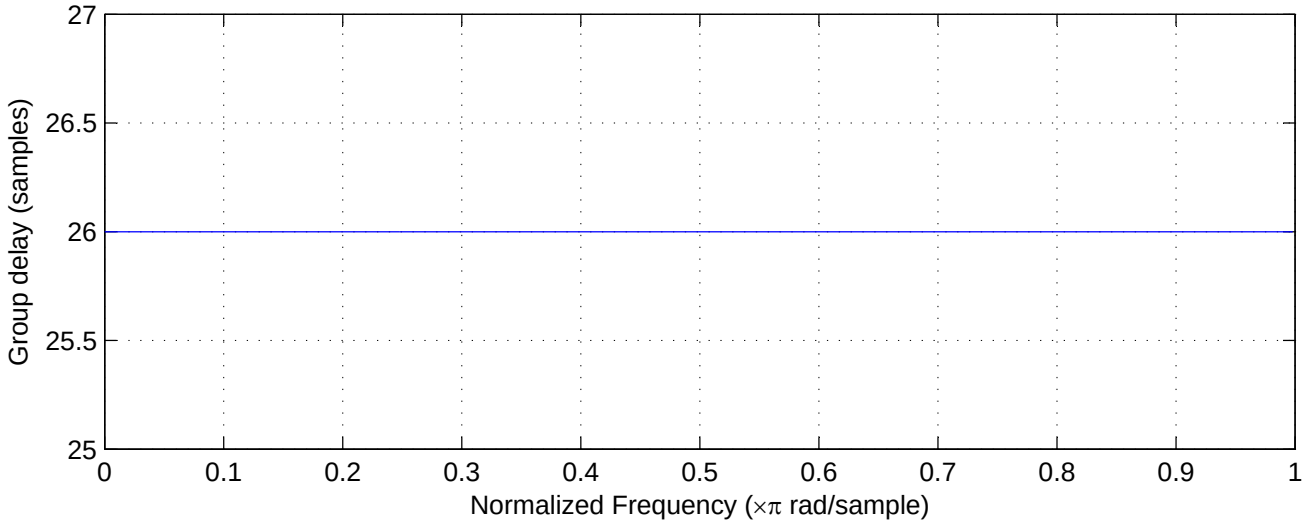
Though this seems like an inconvenience, there are still many advantages to having an optimal FIR filter. FIR filters can have generalized linear phase, which implies that it will also have

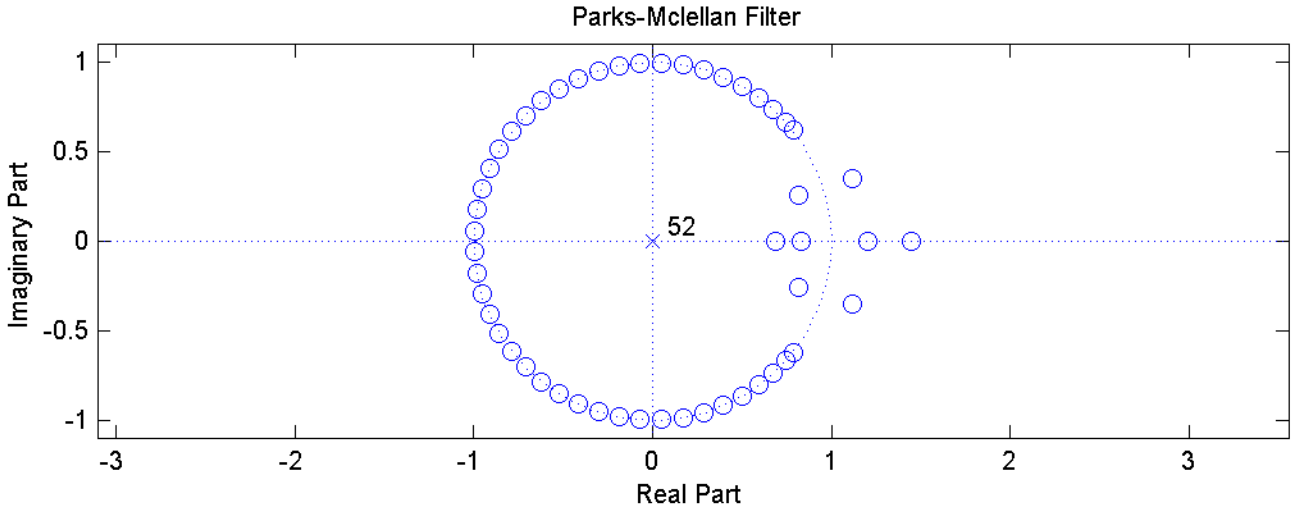
constant group delay. The only real draw back to using an IIR filter is that it does not have this trait. With non-constant group delay, your input signal could be very distorted and shifted even though the magnitude does not affect it. In most cases, we only care about constant group delay in the passband (and sometimes the transition band) because the amplitude of the distortion caused by signals in the stopband are very small anyway.

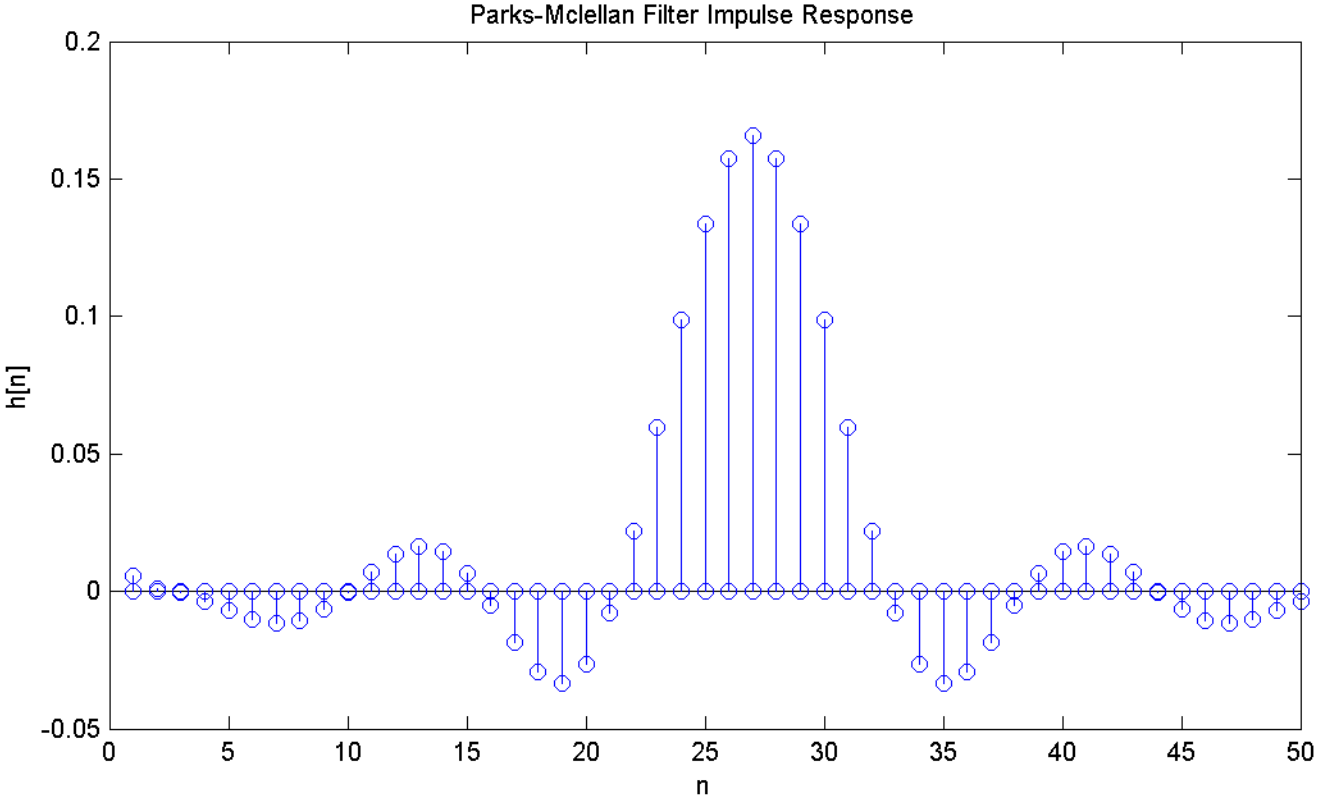
While the Parks-Mclellan filter does have generalized linear phase and meets the specifications, it does so by a great cost. The minimum order needed to achieve all of these things was 52. If we look at the minimum order of an IIR filter, the Elliptic filter achieved the same specifications with only an order of 5. This enormous factor is one reason why engineers still use IIR filters. An increased order means increased costs, computation time, and complexity. In the end, it's really a tradeoff for the design engineer to weight how important constant group delay is. Both the IIR filters and the FIR filters have their own advantages and disadvantages.


Parks-Mclellan Filter
Magnitude (dB)
0
-20
-40
-60
-80
-100
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (xπ rad/sample)


Magnitude
1
0.95
0.9
0.85
0.8
0.75
0.7
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16
Normalized Frequency (xπ rad/sample)


Group delay (samples)
27
26.5
26
25.5
25
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Normalized Frequency (×π rad/sample)


Parks-Mclellan Filter
Imaginary Part
Real Part
52
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3


Parks-Mclellan Filter Impulse Response
h[n]
n
0.2
0.15
0.1
0.05
0
-0.05
0
5
10
15
20
25
30
35
40
45
50

Part B. All-pass Filters and Fractional Delay

Given a particular transfer function, $H(e^{j\omega})$, the group delay of the transfer function is just:

$$\tau(\omega) = -\frac{d}{d\omega}\left[\angle H\left(e^{j\omega}\right)\right]$$

First, we will consider the case where the group delay is a non-integer constant.

$$\tau(\omega) = \tau_0, \ \tau_0 \notin \mathbb{Z}$$

Rewriting the transfer function, we see the following:

$$H\left(e^{j\omega}\right) = \left|H\left(e^{j\omega}\right)\right| e^{j\angle H\left(e^{j\omega}\right)} = \left|H\left(e^{j\omega}\right)\right| e^{-j\int \tau(\omega) d\omega} = \left|H\left(e^{j\omega}\right)\right| e^{-j(\tau_0\omega + C)}$$

Here, we will consider the specific case where the transfer function, $H(e^{j\omega})$, is an all-pass filter, meaning that it has a constant magnitude anywhere. This assumption can be made with no loss of generality because we are only considering the phase in this discussion. We can always perform filtering by $H(e^{j\omega})$ in two steps.

1) Multiply the input magnitude, $\left|X\left(e^{j\omega}\right)\right|$, by the transfer function magnitude, $\left|H\left(e^{j\omega}\right)\right|$
2) Add the input phase, $\angle X\left(e^{j\omega}\right)$, with the transfer function phase, $\angle H\left(e^{j\omega}\right)$

Therefore, we can just ignore the magnitude when we are considering the phase of the transfer function. For our case, we will just assume the magnitude is constant.

Step 2 in the filtering process described above is equivalent to the following:

$$X\left(e^{j\omega}\right) e^{j\angle H\left(e^{j\omega}\right)} = X\left(e^{j\omega}\right) e^{-j(\tau_0\omega + C)}$$

We can think of this multiplication in the frequency domain as a convolution in the time domain. In other words, we have that the output is just:

$$y[n] = e^{-jC} x[n] * DTFT^{-1}\left\{e^{-j\tau_0\omega}\right\} = \begin{cases} e^{-jC} x[n] * \delta[n - \tau_0] & \tau_0 \in \mathbb{Z} \\ e^{-jC} x[n] * \frac{\sin(\pi(n-\tau_0))}{\pi(n-\tau_0)} & \tau_0 \notin \mathbb{Z} \end{cases}$$

Clearly, when τ_0 is an integer, the system is rational (actually it's just a shifted version of the sampled input). However, when τ_0 is not an integer, the transfer function is equivalent to interpolating the input with *sinc* functions, shifting by a non-integer factor of the sampling period, and then resampling and the same sampling period. This obviously can not be viewed as a rational system because it can not be written as a fraction of polynomials. Therefore, in the following section, we will only be considering rational transfer functions.

B.1. Exactly All-pass Filters

As stated in the problem, we are interested in finding "…*stable*, *causal* systems with *real impulse responses* and *rational system functions*." These conditions will simplify the placement of poles in the all-pass filters a great deal. However, the simplifications will partially depend on what order filter we are designing. In each section, restrictions to pole locations will be stated according to these design conditions.

In general, the following restrictions are needed:

1) *Stability* – ROC must include unit circle
2) *Causality* – All poles must be inside unit circle
3) *Real Impulse Response* – All complex conjugates of poles must also be a pole, and all complex conjugates of zeroes must also be a zero.

The first two conditions combine to imply that all poles must lie within the unit circle ($|r| < 1$). The third condition implies that a pole or zero must lie on the real axis, or exist in a pair with its complex conjugate. Both of these constraints will be kept in mind when designing the rest of the filters.

B.1.A – Exactly All-pass Filters (Minimizing Error Energy)

Design all-pass filters of order 1 and 2 with Property 1 above and the following minimized:

$$\varepsilon = \int_{\pi/4}^{\pi/2} \left| \tau(\omega) - 0.5 \right|^2 d\omega$$

B.1.A – Order 1 All-pass Filter

The most general 1st order all-pass filter with a constant magnitude response of 1 has the form:

$$H_{AP}(z) = \frac{z^{-1} - a^*}{1 - az^{-1}} = \frac{z^{-1} - r_a e^{-j\theta_a}}{1 - r_a e^{j\theta_a} z^{-1}}$$

From Eqn. 5.95 in OSB, we know that the group delay for this 1st order all-pass filter is given by:

$$\tau(\omega) = \frac{1 - r_a^2}{1 + r_a^2 - 2r_a \cos(\omega - \theta_a)}$$

A 1st order filter means that there is only one pole and one zero. With this in mind, and remembering that we are designing a filter that must have a real impulse response, it is obvious that the pole and zero in the all-pass filter must lie on the real axis. In other words, we are restricted to the following pole location:

$$a(r_a, \theta_a) = (r_a, 0) \text{ or } (r_a, \pi) = (\pm r_a, 0)$$

This special 1st order restriction simplifies the impulse response and group delay to be:

$$H_{AP}(z) = \frac{z^{-1} - r_a}{1 - r_a z^{-1}} \quad \text{and} \quad \tau(\omega) = \frac{1 - r_a^2}{1 + r_a^2 - 2r_a \cos(\omega)}$$

Therefore, all we need to do is run *fmincon* on the optimization parameter to find the pole. We are using a constrained minimization because we know that the radius has to exist in the interval [0, 1) to maintain causality and stability.

The following code was used in Matlab to minimize the parameter.

```
% Designing an all-pass filter of order 1
% Pole located at z = a = r*e^(-j*theta)
% x(1) = r

% Define the Group Delay
grd = @(x, w) ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w)));

% Define the parameter to minimize
e = @(x) (quad( @(w) (abs(grd(x,w)-0.5)).^2 , pi/4, pi/2));

% Find the minimum of the minimization parameter
%   Search with r=[-1,1]
for i=1:7
    % get r to jump by 0.25 from [-0.75, 0.75]
    r = (mod((i-1),7)-3)*0.25;
    a(i,:) = fmincon(e, [r], [],[],[],[],[-1],[1]);
end
```

Matlab's *fmincon* returns two different possible pole locations:

$$a_1(r, \theta) = (-0.4771, 0) \quad \text{or} \quad a_2(r, \theta) = (0.7647, 0)$$

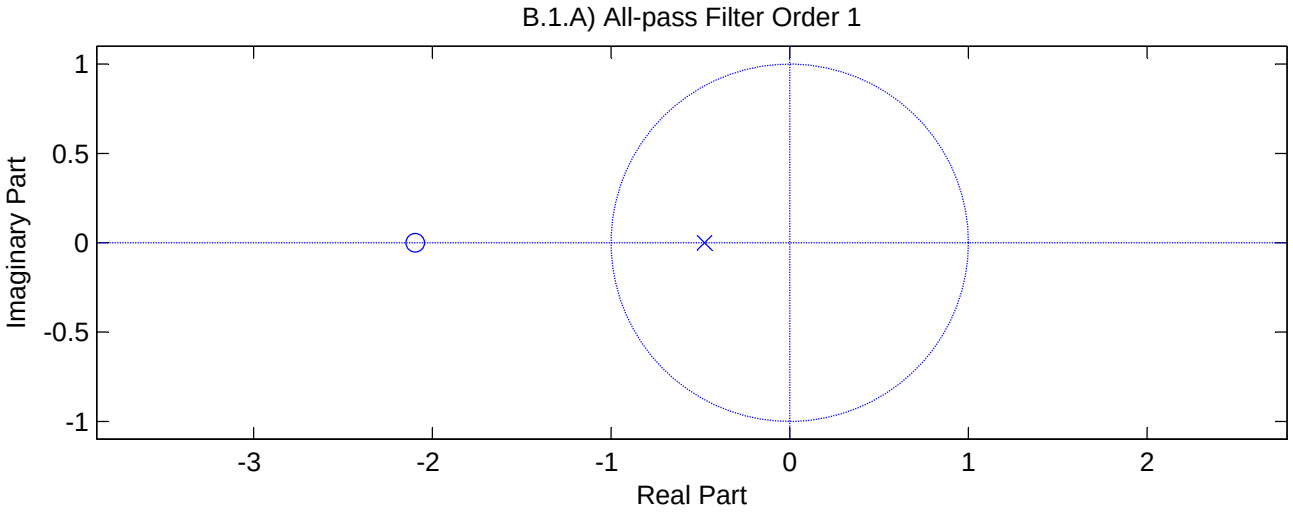
Because *fmincon* finds a local minimum, we need to actually evaluate the optimization parameter at these different pole locations to find the global minimum.

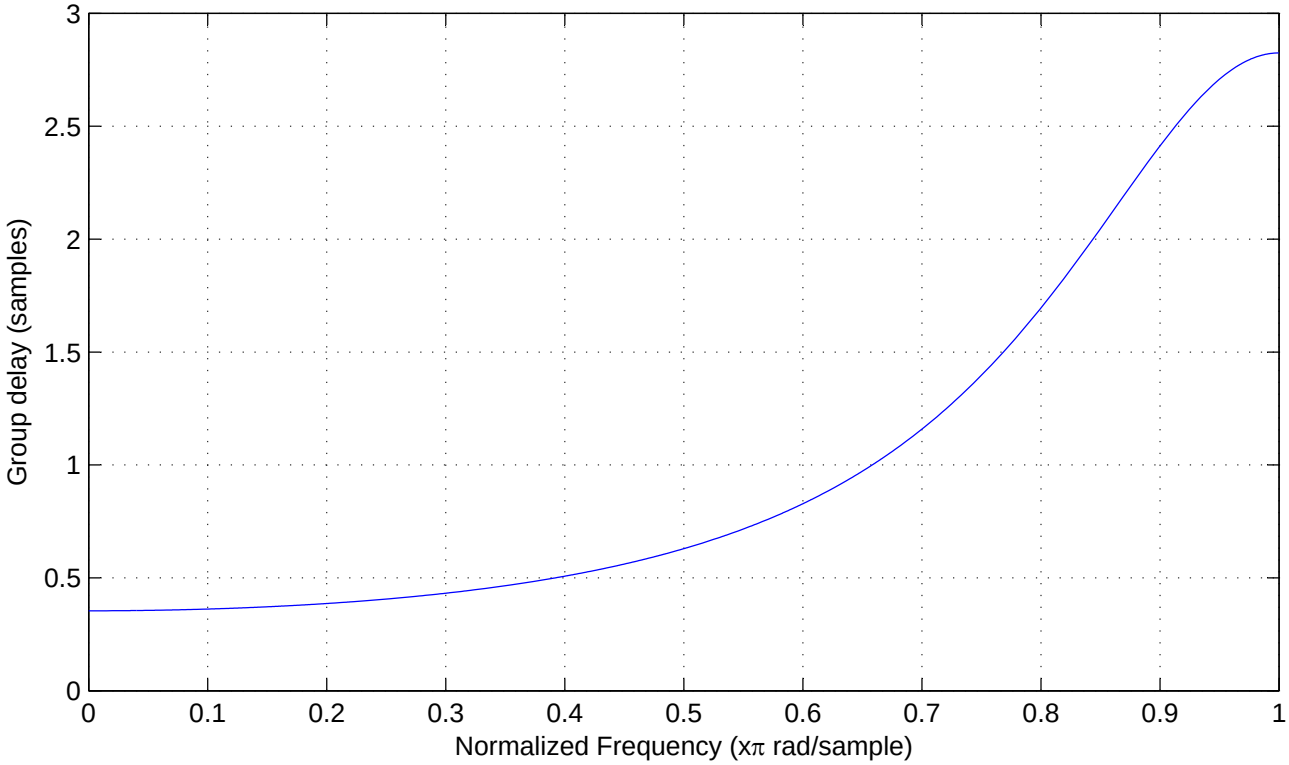
$$\boxed{\varepsilon\left(a_1\right) = 3.204\times10^{-3}} \quad \text{or} \quad \varepsilon\left(a_2\right) = 2.389\times10^{-2}$$

Therefore, the optimal solution for pole placement under the given restrictions leads to the pole being at $a_1\left(r,\theta\right) = \left(\text{-}0.4771, 0\right) = \left(0.4771, \pi\right)$

The following page contains a pole-zero plot and a plot of the group delay for the optimal filter.

$$\boxed{H_{AP}\left(z\right) = \frac{z^{-1}+0.4771}{1+0.4771z^{-1}} = 0.4771\left[\frac{1+2.0962z^{-1}}{1+0.4771z^{-1}}\right]}$$


B.1.A) All-pass Filter Order 1
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2


Group delay (samples)
Normalized Frequency (xπ rad/sample)
3
2.5
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

B.1.A – Order 2 All-pass Filter

The most general 2nd order all-pass filter with a constant magnitude response of 1 has the form:

$$H_{AP}(z) = H_a(z) \times H_b(z) = \left(\frac{z^{-1} - a^*}{1 - az^{-1}}\right)\left(\frac{z^{-1} - b^*}{1 - bz^{-1}}\right)$$

We know group delay is given by just:

$$\tau(\omega) = -\frac{d}{d\omega}\left[\angle H_{AP}\left(e^{j\omega}\right)\right] = -\frac{d}{d\omega}\left[\angle H_a\left(e^{j\omega}\right) + \angle H_b\left(e^{j\omega}\right)\right]$$

$$\tau(\omega) = -\frac{d}{d\omega}\left[\angle H_a\left(e^{j\omega}\right)\right] - \frac{d}{d\omega}\left[\angle H_b\left(e^{j\omega}\right)\right] = \tau_a(\omega) + \tau_b(\omega)$$

In other words, the group delay of the cascaded 2nd order all-pass filter is just the sum of the group delays of each 1st order all-pass system. Therefore, we have the following expression for the group delay of our 2nd order all-pass filter:

$$\tau(\omega) = \left[\frac{1 - r_a^2}{1 + r_a^2 - 2r_a\cos(\omega - \theta_a)}\right] + \left[\frac{1 - r_b^2}{1 + r_b^2 - 2r_b\cos(\omega - \theta_b)}\right]$$

Again, we can make simplifications from the restrictions given in the problem. In addition to knowing the poles have to lie inside the unit circle again, we can make a further restriction knowing that there are only two poles in a 2nd order system. The two poles must satisfy **one** of the following conditions:

1) The poles must both lie on the real axis, thus both being real

   $\theta_a = \theta_b = 0$ (assuming *r* can take on negative values)

2) The poles must be complex conjugates of each other

   $r_a = r_b$ and $\theta_a = -\theta_b$

Therefore, the impulse response and group delay can be simplified to be **either** one of the following:

$$H_1(z) = \left(\frac{z^{-1} - r_a}{1 - r_a z^{-1}}\right)\left(\frac{z^{-1} - r_b}{1 - r_b z^{-1}}\right), \quad \tau_1(\omega) = \left[\frac{1 - r_a^2}{1 + r_a^2 - 2r_a\cos(\omega)}\right] + \left[\frac{1 - r_b^2}{1 + r_b^2 - 2r_b\cos(\omega)}\right]$$

or

$$H_2(z) = \left(\frac{z^{-1} - r_a e^{-j\theta_a}}{1 - r_a e^{j\theta_a} z^{-1}}\right)\left(\frac{z^{-1} - r_a e^{j\theta_a}}{1 - r_a e^{-j\theta_a} z^{-1}}\right), \quad \tau_2(\omega) = \left[\frac{1 - r_a^2}{1 + r_a^2 - 2r_a\cos(\omega - \theta_a)}\right] + \left[\frac{1 - r_a^2}{1 + r_a^2 - 2r_a\cos(\omega + \theta_a)}\right]$$

The optimization must now be done on both of these group delays.

The following code was used in Matlab to minimize the parameter for both cases. Again, we use *fmincon* because the radius still has to be in the interval [0, 1) for causality and stability.

```
% Designing an all-pass filter of order 2
% Pole located at z = r*e^(-j*theta)
% x corresponds to the case when poles are both real
%   x(1) = r_a
%   x(2) = r_b
% y corresponds to the case when poles are complex conjugate pairs
%   y(1) = r_a
%   y(2) = theta_a

% Define the Group Delay
grd1 = @(x, w) ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w))) + ...
               ( (1-x(2).^2) ./ (1+x(2).^2-2.*x(2).*cos(w)));

grd2 = @(x, w) ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w-x(2)))) + ...
               ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w+x(2))));

% Define the parameter to minimize
e1 = @(x) (quad( @(w) (abs(grd1(x,w)-0.5)).^2 , pi/4, pi/2));
e2 = @(x) (quad( @(w) (abs(grd2(x,w)-0.5)).^2 , pi/4, pi/2));

% Find the minimum of the minimization parameter
%   Search with r1=[-1,1] and r2=[-1,1]
for i=1:49
    % get r1 to jump by 0.25 from [-0.75, 0.75]
    r1 = (mod((i-1),7)-3)*0.25;
    % get r2 to jump by 0.25 from [-0.75, 0.75]
    r2 = (floor((i-1)/7)-3)*0.25;
    p1(i,:) = fmincon(e1, [r1 r2],[],[],[],[],[-1;-1],[1;1]);
end

% Find the minimum of the minimization parameter
%   Search with r=[-1,1] and theta=[0,2pi]
for i=1:28
    % get r to jump by 0.25 from [-0.75, 0.75]
    r = (mod((i-1),7)-3)*0.25;
    % get r to jump by pi/2 from [0, 3pi/2]
    theta = floor((i-1)/7)*pi/2;
    p2(i,:) = fmincon(e2, [r theta],[],[],[],[],[-1;0],[1;2*pi]);
end
```

The following pole locations were found by Matlab as local minimums:

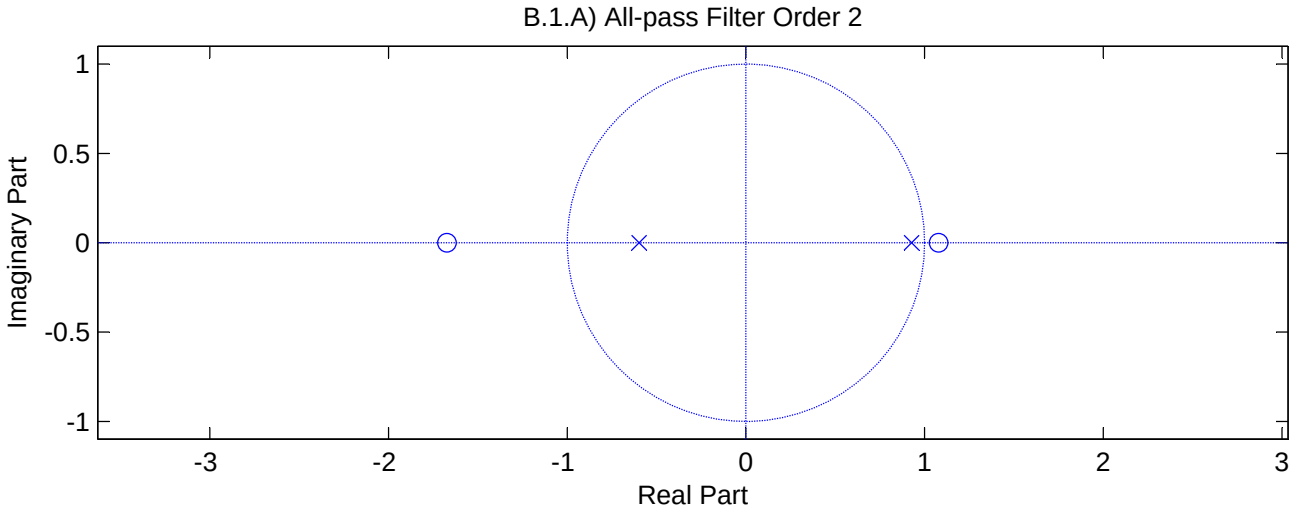
	Pole 1 (r,θ)	Pole 2 (r,θ)	$\varepsilon = \int_{\pi/4}^{\pi/2} \left\lvert \tau(\omega) - 0.5 \right\rvert^2 d\omega$
1	(-0.7144, 0)	(-0.7144, 0)	4.353×10^{-3}
2	(-0.6017, 0)	(-0.8365, 0)	4.102×10^{-3}
3	(-0.4770, 0)	(-1.0000, 0)	3.204×10^{-3}
4	(0.9270, 0)	(-0.5977, 0)	3.630×10^{-4}
5	(0.7647, 0)	(1.0000, 0)	2.039×10^{-2}
6	(0.8824, 0)	(0.8824, 0)	2.244×10^{-2}

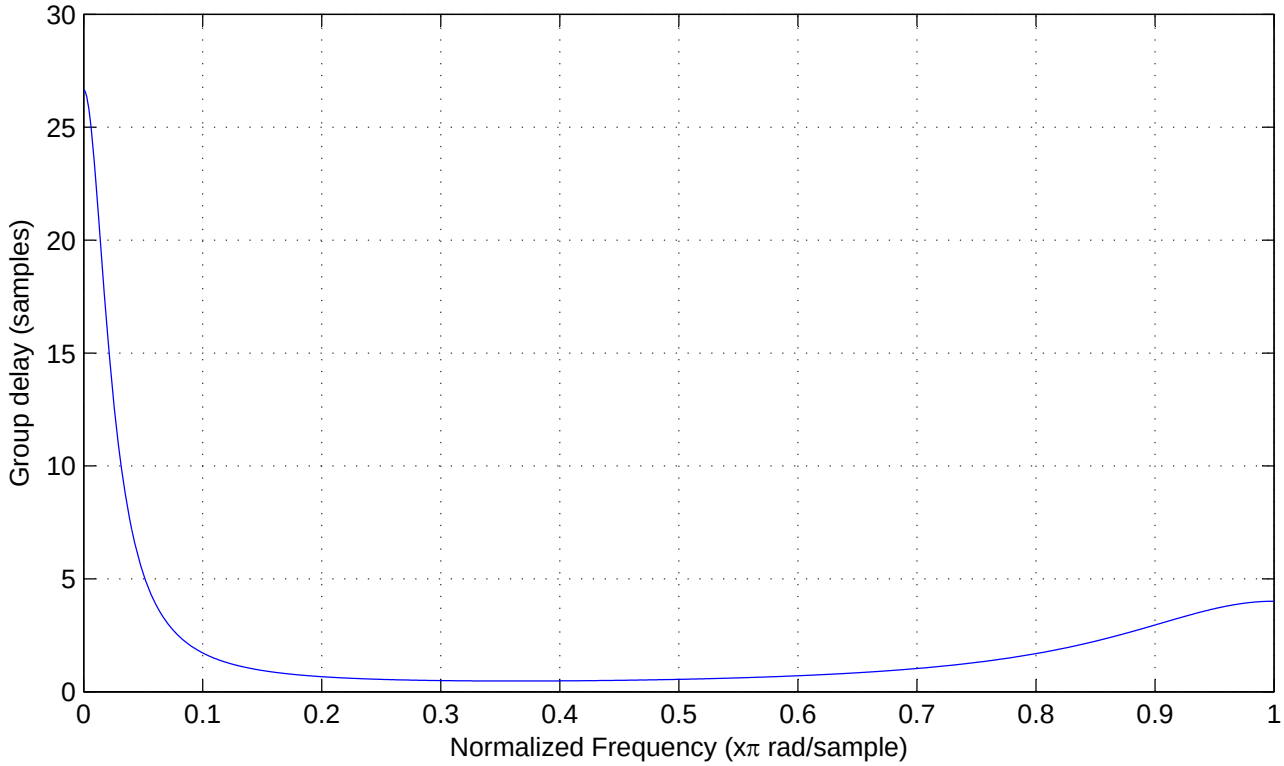
Clearly, the minimum optimization parameter occurs in case 4, where the poles exist at:

$$a(r,\theta) = (0.9270, 0) \quad \text{and} \quad b(r,\theta) = (0.5977, \pi)$$

The following page contains a pole-zero plot and a plot of the group delay for the optimal filter.

$$H_{AP}(z) = \left(\frac{z^{-1} - 0.9270}{1 - 0.9270 z^{-1}}\right)\left(\frac{z^{-1} + 0.5977}{1 + 0.5977 z^{-1}}\right) = -0.5541 \frac{\left(1 - 1.0787 z^{-1}\right)\left(1 + 1.6731 z^{-1}\right)}{\left(1 - 0.9270 z^{-1}\right)\left(1 + 0.5977 z^{-1}\right)}$$


B.1.A) All-pass Filter Order 2
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3


Group delay (samples)
Normalized Frequency (xπ rad/sample)
30
25
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

B.1.B – Exactly All-pass Filters (Minimizing Maximum Error)

Design all-pass filters of order 1 and 2 with Property 1 above and the following minimized:

$$\varepsilon = \max_{\omega \in [\pi/4, \pi/2]} \left| \tau(\omega) - 0.5 \right|$$

B.1.B – Order 1 All-pass Filter

For the same reason as the 1st order all-pass filter designed in B.1.A, we know that this all-pass filter must have the following transfer function and group delay:

$$H_{AP}(z) = \frac{z^{-1} - r_a}{1 - r_a z^{-1}} \quad \text{and} \quad \tau(\omega) = \frac{1 - r_a^2}{1 + r_a^2 - 2r_a \cos(\omega)}$$

Therefore, all we need to do is run *fmincon* on the optimization parameter to find the pole.

The following code was used in Matlab to minimize the parameter.

```
% Designing an all-pass filter of order 1
% Pole located at z = a = r*e^(-j*theta)
% x(1) = r

% Define the Group Delay
grd = @(x, w) ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w)));

% Define the parameter to minimize
% Create a range of w=[pi/4,pi/2];
wrange = pi/4 : 0.0001 : pi/2;
e = @(x) (max( abs(grd(x,wrange)-0.5)));

% Find the minimum of the minimization parameter
%   Search with r=[-1,1]
for i=1:7
    % get r to jump by 0.25 from [-0.75, 0.75]
    r = (mod((i-1),7)-3)*0.25;
    a(i,:) = fmincon(e, [r], [],[],[],[],[-1],[1]);
end
```

Matlab's *fmincon* returns two different possible pole locations:

$$a_1(r,\theta) = (-0.4926, 0) \quad \text{or} \quad a_2(r,\theta) = (0.7841, 0)$$

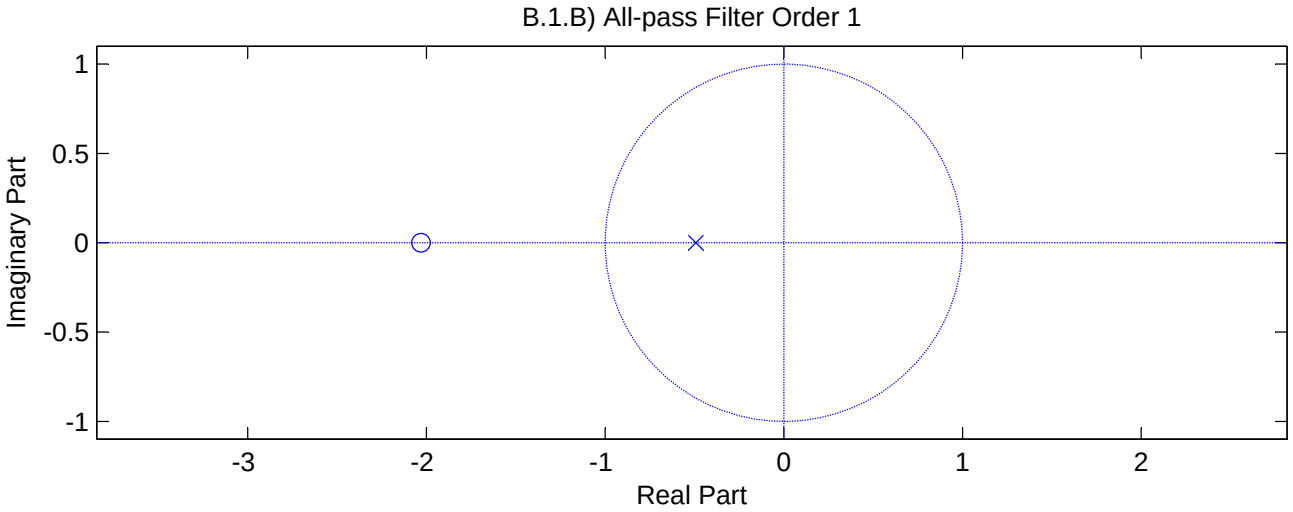
Because *fmincon* finds a local minimum, we need to actually evaluate the optimization parameter at these different pole locations to find the global minimum.

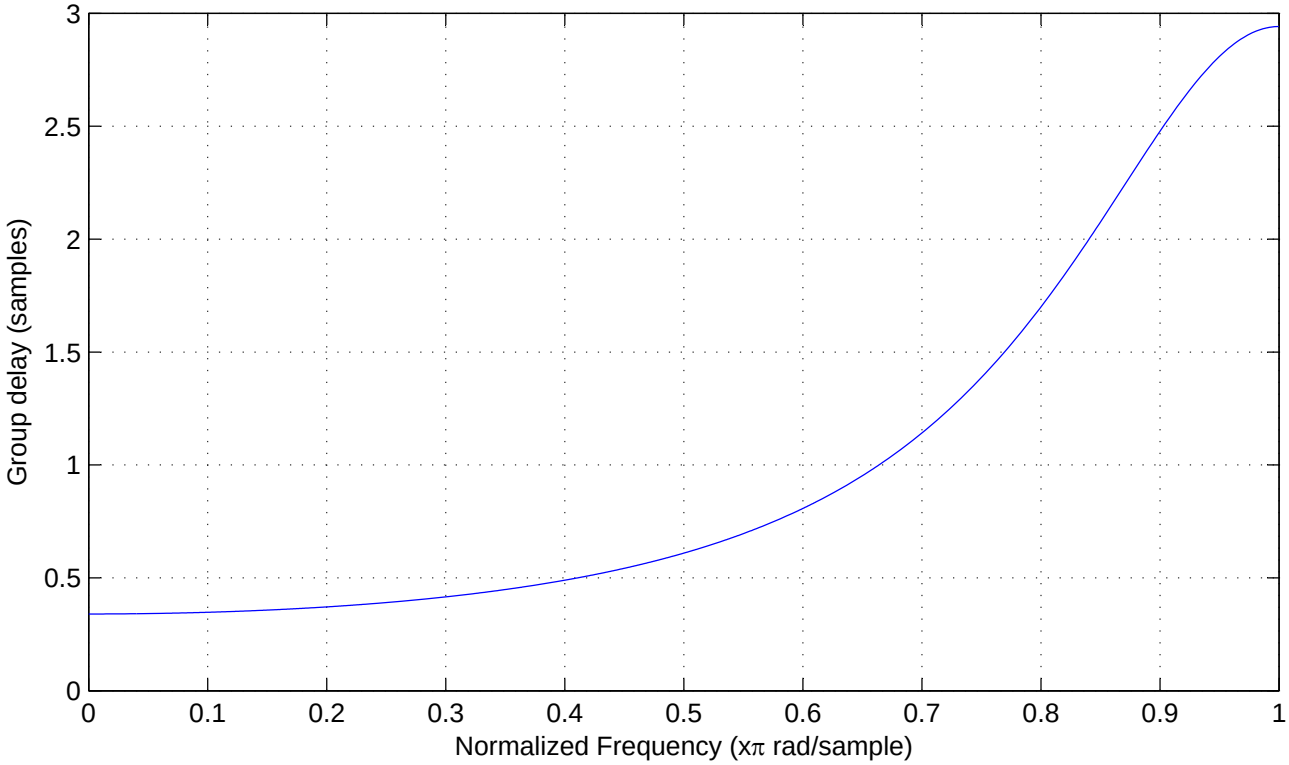
$$\boxed{\varepsilon(a_1) = 1.094 \times 10^{-1}} \quad \text{or} \quad \varepsilon(a_2) = 2.614 \times 10^{-1}$$

Therefore, the optimal solution for pole placement under the given restrictions leads to the pole being at $a_1(r,\theta) = (-0.4926, 0) = (0.4926, \pi)$

The following page contains a pole-zero plot and a plot of the group delay for the optimal filter.

$$\boxed{H_{AP}(z) = \frac{z^{-1} + 0.4926}{1 + 0.4926 z^{-1}} = 0.4926 \left[\frac{1 + 2.0302 z^{-1}}{1 + 0.4926 z^{-1}} \right]}$$


B.1.B) All-pass Filter Order 1
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2


Group delay (samples)
Normalized Frequency (xπ rad/sample)
3
2.5
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

B.1.B – Order 2 All-pass Filter

For the same reason as the 2nd order all-pass filter designed in B.1.A, we know that this all-pass filter must have one of the following transfer functions and group delays:

$$H_1(z)=\left(\frac{z^{-1}-r_a}{1-r_a z^{-1}}\right)\left(\frac{z^{-1}-r_b}{1-r_b z^{-1}}\right),\quad \tau_1(\omega)=\left[\frac{1-r_a^2}{1+r_a^2-2r_a\cos(\omega)}\right]+\left[\frac{1-r_b^2}{1+r_b^2-2r_b\cos(\omega)}\right]$$

or

$$H_2(z)=\left(\frac{z^{-1}-r_a e^{-j\theta_a}}{1-r_a e^{j\theta_a} z^{-1}}\right)\left(\frac{z^{-1}-r_a e^{j\theta_a}}{1-r_a e^{-j\theta_a} z^{-1}}\right),\quad \tau_2(\omega)=\left[\frac{1-r_a^2}{1+r_a^2-2r_a\cos(\omega-\theta_a)}\right]+\left[\frac{1-r_a^2}{1+r_a^2-2r_a\cos(\omega+\theta_a)}\right]$$

Therefore, all we need to do is run *fmincon* on the optimization parameter to find the pole.

The following code was used in Matlab to minimize the parameter.

```
% Designing an all-pass filter of order 2
% Pole located at z = r*e^(-j*theta)
% x corresponds to the case when poles are both real
%   x(1) = r_a
%   x(2) = r_b
% y corresponds to the case when poles are complex conjugate pairs
%   y(1) = r_a
%   y(2) = theta_a

% Define the Group Delay
grd1 = @(x, w) ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w))) + ...
               ( (1-x(2).^2) ./ (1+x(2).^2-2.*x(2).*cos(w)));
grd2 = @(x, w) ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w-x(2)))) + ...
               ( (1-x(1).^2) ./ (1+x(1).^2-2.*x(1).*cos(w+x(2))));

% Define the parameter to minimize
% Create a range of w=[pi/4,pi/2];
wrange = pi/4 : 0.0001 : pi/2;
e1 = @(x) (max( abs(grd1(x,wrange)-0.5)));
e2 = @(x) (max( abs(grd2(x,wrange)-0.5)));

% Find the minimum of the minimization parameter
%   Search with r1=[-1,1] and r2=[-1,1]
for i=1:49
    % get r1 to jump by 0.25 from [-0.75, 0.75]
    r1 = (mod((i-1),7)-3)*0.25;
    % get r2 to jump by 0.25 from [-0.75, 0.75]
    r2 = (floor((i-1)/7)-3)*0.25;
    p1(i,:) = fmincon(e1, [r1 r2],[],[],[],[],[-1;-1],[1;1]);
end

% Find the minimum of the minimization parameter
%   Search with r=[-1,1] and theta=[0,2pi]
for i=1:28
    % get r to jump by 0.25 from [-0.75, 0.75]
    r = (mod((i-1),7)-3)*0.25;
    % get r to jump by pi/2 from [0, 3pi/2]
    theta = floor((i-1)/7)*pi/2;
    p2(i,:) = fmincon(e2, [r theta],[],[],[],[],[-1;0],[1;2*pi]);
end
```

The following pole locations were found by Matlab as local minimums:

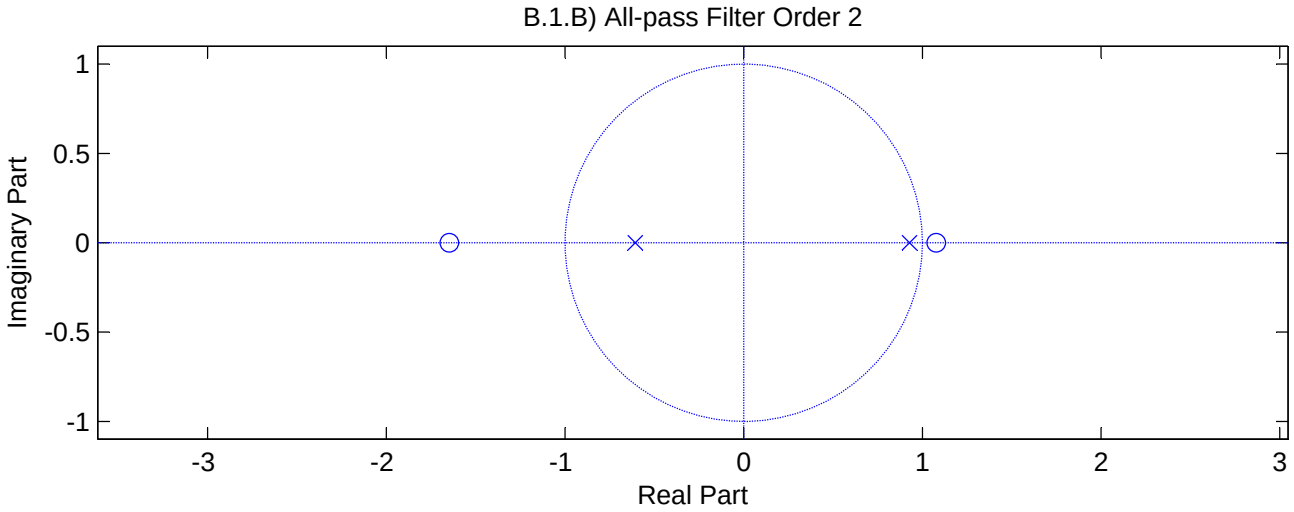
	Pole 1 (r,θ)	Pole 2 (r,θ)	$\varepsilon = \int_{\pi/4}^{\pi/2} \left\lvert \tau(\omega) - 0.5 \right\rvert^2 d\omega$
1	(-0.7235, 0)	(-0.7235, 0)	1.257 x 10^{-1}
2	(-0.4926, 0)	(-1.0000, 0)	1.094 x 10^{-1}
3	(0.9282, 0)	(-0.6070, 0)	3.582 x 10^{-2}
4	(0.8912, 0)	(0.8912, 0)	2.7066 x 10^{-1}
5	(0.9999, 0)	(0.7841, 0)	2.6143 x 10^{-1}
6	(1.0000, 1.5543)	(1.0000, -1.5543)	5.000 x 10^{-1}
7	(0.0004, 3.1416)	(0.0004, -3.1416)	1.500 x 10^{0}

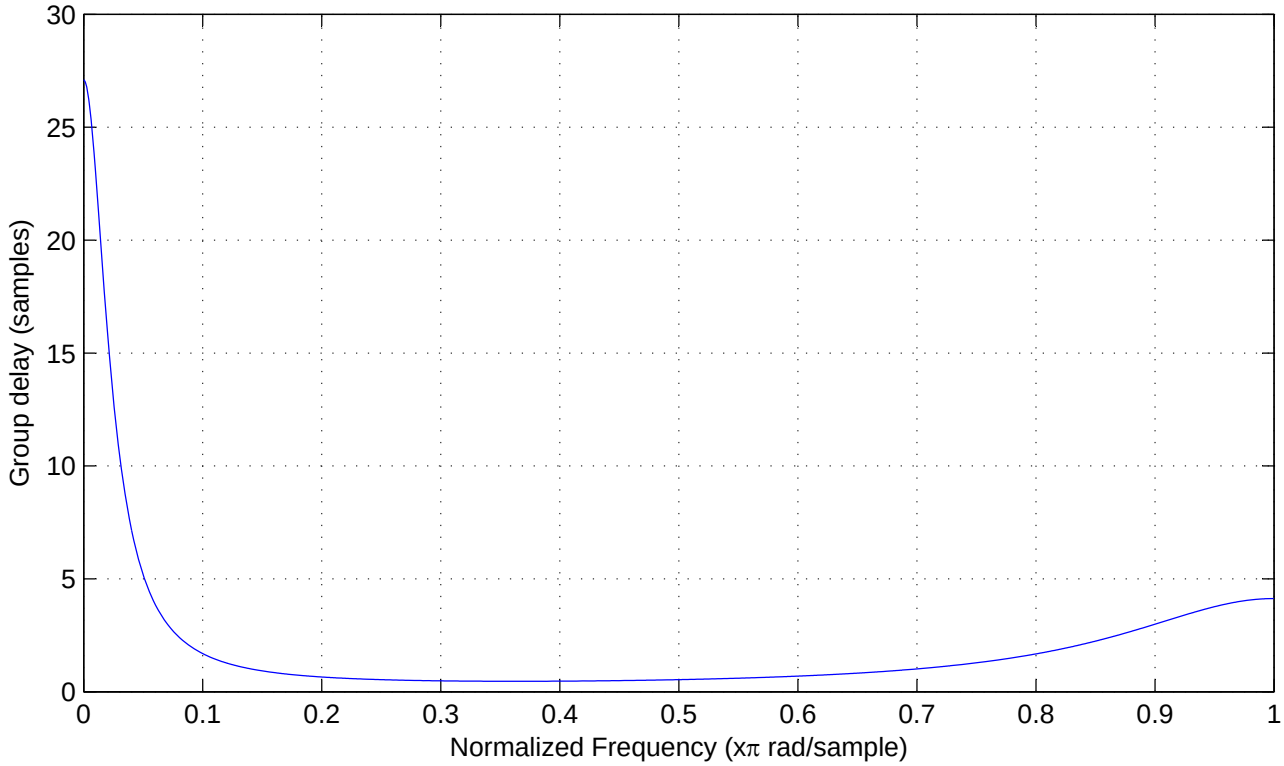
Clearly, the minimum optimization parameter occurs in case 3, where the poles exist at:

$$a(r,\theta) = (0.9282, 0) \quad \text{and} \quad b(r,\theta) = (0.6070, \pi)$$

The following page contains a pole-zero plot and a plot of the group delay for the optimal filter.

$$H_{AP}(z) = \left(\frac{z^{-1} - 0.9282}{1 - 0.9282 z^{-1}}\right)\left(\frac{z^{-1} + 0.6070}{1 + 0.6070 z^{-1}}\right) = -0.5634 \frac{\left(1 - 1.0774 z^{-1}\right)\left(1 + 1.6474 z^{-1}\right)}{\left(1 - 0.9282 z^{-1}\right)\left(1 + 0.6070 z^{-1}\right)}$$


B.1.B) All-pass Filter Order 2
Imaginary Part
Real Part
1
0.5
0
-0.5
-1
-3
-2
-1
0
1
2
3


Group delay (samples)
Normalized Frequency (xπ rad/sample)
30
25
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

B.1 – Observations and Comments

The following are the optimal solutions for section B.1:

1st Order All-pass Filter (Part A)

$$H_{AP}(z) = \frac{z^{-1} + 0.4771}{1 + 0.4771 z^{-1}}$$

2nd Order All-pass Filter (Part A)

$$H_{AP}(z) = \left(\frac{z^{-1} - 0.9270}{1 - 0.9270 z^{-1}}\right)\left(\frac{z^{-1} + 0.5977}{1 + 0.5977 z^{-1}}\right)$$

$$\varepsilon_A = \int_{\pi/4}^{\pi/2} \left|\tau(\omega) - 0.5\right|^2 d\omega$$

1st Order All-pass Filter (Part B)

$$H_{AP}(z) = \frac{z^{-1} + 0.4926}{1 + 0.4926 z^{-1}}$$

2nd Order All-pass Filter (Part B)

$$H_{AP}(z) = \left(\frac{z^{-1} - 0.9282}{1 - 0.9282 z^{-1}}\right)\left(\frac{z^{-1} + 0.6070}{1 + 0.6070 z^{-1}}\right)$$

$$\varepsilon_B = \max_{\omega \in [\pi/4, \pi/2]} \left|\tau(\omega) - 0.5\right|$$

It is interesting that under the two separate optimization parameters, the optimal results were still fairly close. The 2nd order systems are actually even closer than the 1st order systems. In general, if you minimize the maximum value of something, it does not mean that you are also minimizing the total energy of the same thing. However, in this case, it seems to be doing just that. It seems like the optimal solutions for both error criterion converge to the same thing.

If we look at the group delay for any of the optimal 1st and 2nd order filters, and concentrate on the range of $\omega \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right]$, we see that the group delay is nearly centered around 0.5. In this interval, the group delay is smooth, continuous, and already fairly flat. Therefore, minimizing the maximum error makes the entire group delay within this interval closer to 0.5. In other words, minimizing the maximum error also minimizes the total energy of the error.

It should also be noted that these optimization criteria are still not the best. We are looking over a very specific range of frequencies (specifically, $\omega \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right]$), which means the group delay could be potentially very far from 0.5 outside of this interval. Actually, this is exactly what happens with higher orders. Looking at the plots for the 1st order versus the 2nd order, you can see that the group delay around $\omega \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right]$ gets closer to 0.5 with the extra pole-zero pair. However, the group delay near $\omega = 0$ is much larger (almost a factor of 100). This is clearly due to the pole-zero pair very close to z=1. However, it illustrates that adding a pole-zero pair may not actually make the entire transfer function have more linear phase. If this is indeed what we are after, I think the best error criterion to evaluate over would be the total energy in the error between the group delay and some constant over the entire interval.

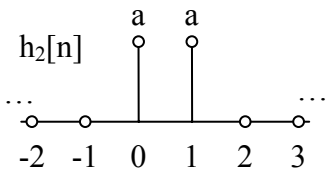
$$\varepsilon = \int_0^{\pi} \left|\tau(\omega) - \tau_0\right|^2 d\omega$$

B.2. FIR Approximations to All-pass Filters

The following calculations find expressions for the magnitude of the transfer function given a general FIR filter of Type II and IV, and of lengths 2 and 4.

FIR Filter (M=2)

Type II FIR Filter (Even Symmetry)



$$h_2[n] = a\delta[n] + a\delta[n-1]$$

$$\tau(\omega) = \tfrac{1}{2},\ \angle H_2\left(e^{j\omega}\right) = -\tfrac{\omega}{2}$$

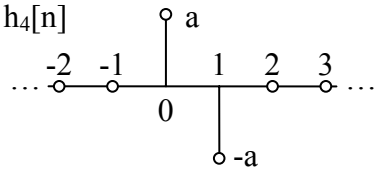
$$H_2\left(e^{j\omega}\right) = \left|H_2\left(e^{j\omega}\right)\right| e^{-j\frac{\omega}{2}}$$

$$H_2\left(e^{j\omega}\right) = a + ae^{-j\omega} = ae^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} + e^{-j\frac{\omega}{2}}\right]$$

$$H_2\left(e^{j\omega}\right) = 2a\cos\left(\tfrac{\omega}{2}\right)e^{-j\frac{\omega}{2}}$$

$$\boxed{\left|H_2\left(e^{j\omega}\right)\right| = \left|2a\cos\left(\tfrac{\omega}{2}\right)\right|}$$

Type IV FIR Filter (Odd Symmetry)



$$h_4[n] = a\delta[n] - a\delta[n-1]$$

$$\tau(\omega) = \tfrac{1}{2},\ \angle H_4\left(e^{j\omega}\right) = -\tfrac{\omega}{2}$$

$$H_4\left(e^{j\omega}\right) = \left|H_4\left(e^{j\omega}\right)\right| e^{-j\frac{\omega}{2}}$$

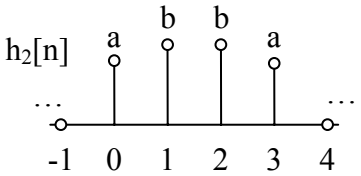
$$H_4\left(e^{j\omega}\right) = a - ae^{-j\omega} = ae^{-j\frac{\omega}{2}}\left[e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right]$$

$$H_4\left(e^{j\omega}\right) = j2a\sin\left(\tfrac{\omega}{2}\right)e^{-j\frac{\omega}{2}} = 2a\sin\left(\tfrac{\omega}{2}\right)e^{-j\frac{\omega}{2}-\frac{\pi}{2}}$$

$$\boxed{\left|H_4\left(e^{j\omega}\right)\right| = \left|2a\sin\left(\tfrac{\omega}{2}\right)\right|}$$

FIR Filter (M=4)

Type II FIR Filter (Even Symmetry)



$$h_2[n] = a\delta[n] + b\delta[n-1]$$
$$+b\delta[n-2] + a\delta[n-3]$$

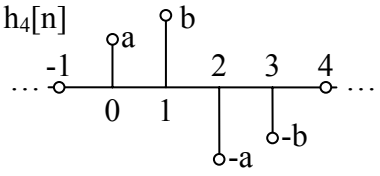
$$H_2\left(e^{j\omega}\right) = \left|H_1\left(e^{j\omega}\right)\right| e^{-j\frac{\omega}{2}}$$

$$H_2\left(e^{j\omega}\right) = a + be^{-j\omega} + be^{-2j\omega} + ae^{-3j\omega}$$

$$H_2\left(e^{j\omega}\right) = e^{-j\frac{3\omega}{2}}\left[2a\cos\left(\tfrac{3\omega}{2}\right) + 2b\cos\left(\tfrac{\omega}{2}\right)\right]$$

$$\boxed{\left|H_2\left(e^{j\omega}\right)\right| = \left|2a\cos\left(\tfrac{3\omega}{2}\right) + 2b\cos\left(\tfrac{\omega}{2}\right)\right|}$$

Type IV FIR Filter (Odd Symmetry)



$$h_4[n] = a\delta[n] + b\delta[n-1]$$
$$-b\delta[n-2] - a\delta[n-3]$$

$$H_4\left(e^{j\omega}\right) = \left|H_4\left(e^{j\omega}\right)\right| e^{-j\frac{\omega}{2}}$$

$$H_4\left(e^{j\omega}\right) = a + be^{-j\omega} - be^{-2j\omega} - ae^{-3j\omega}$$

$$H_4\left(e^{j\omega}\right) = e^{-j\frac{3\omega}{2}}j\left[2a\sin\left(\tfrac{3\omega}{2}\right) + 2b\sin\left(\tfrac{\omega}{2}\right)\right]$$

$$\boxed{\left|H_4\left(e^{j\omega}\right)\right| = \left|2a\sin\left(\tfrac{3\omega}{2}\right) + 2b\sin\left(\tfrac{\omega}{2}\right)\right|}$$

B.2.A – FIR Approximation to All-pass Filters (Minimizing Error Energy)

Design 2 and 4 tap FIR filters with constant group delay (M-1)/2 and the following minimized:

$$\varepsilon = \int_0^{\pi} \left(\left| H\left(e^{j\omega}\right) - 1 \right| \right)^2 d\omega$$

B.2.A – 2-Tap FIR Filter

The magnitude of a 2-tap FIR filter was shown previously to be one of the following (where the subscript denotes the type of the FIR filter):

$$\left|H_2\left(e^{j\omega}\right)\right| = \left|2a\cos\left(\tfrac{\omega}{2}\right)\right| \quad \text{or} \quad \left|H_4\left(e^{j\omega}\right)\right| = \left|2a\sin\left(\tfrac{\omega}{2}\right)\right|$$

Therefore, all we need to do is run *fminsearch* on the optimization parameter to find the taps.

The following code was used in Matlab to minimize the parameter.

```
% Designing a 2 tap FIR Filter
% x(1) = a

% Define the magnitude of the transfer function
mag2 = @(x, w) (abs(2.*x(1).*cos(w/2)));
mag4 = @(x, w) (abs(2.*x(1).*sin(w/2)));

% Define the parameter to minimize
e2 = @(x) (quad( @(w) (mag2 - 1).^2 , 0, pi));
e4 = @(x) (quad( @(w) (mag4 - 1).^2 , 0, pi));

% Find the minimum of the minimization parameter
%   Search with r=[-1,1]
for i=1:7
    % get a to jump by 0.5 from [-1.5, 1.5]
    a = (mod((i-1),7)-3)*0.5;
    a2(i,:) = fminsearch(e2, [a]);
    a4(i,:) = fminsearch(e4, [a]);
end
```

For both the Type II and the Type IV filters, Matlab returned the same optimal answers. This intuitively should be true because the period of the absolute value of sine and cosine are both π. Thus, integrating over a period of π should produce the same answer for the Type II and the Type IV filters.

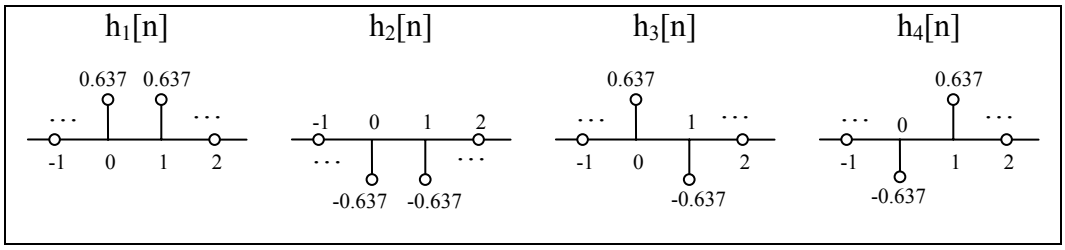
Matlab returns two locally optimal solutions for *a*:

$$a_1 = 0.6366 \quad \text{and} \quad a_2 = -0.6366$$

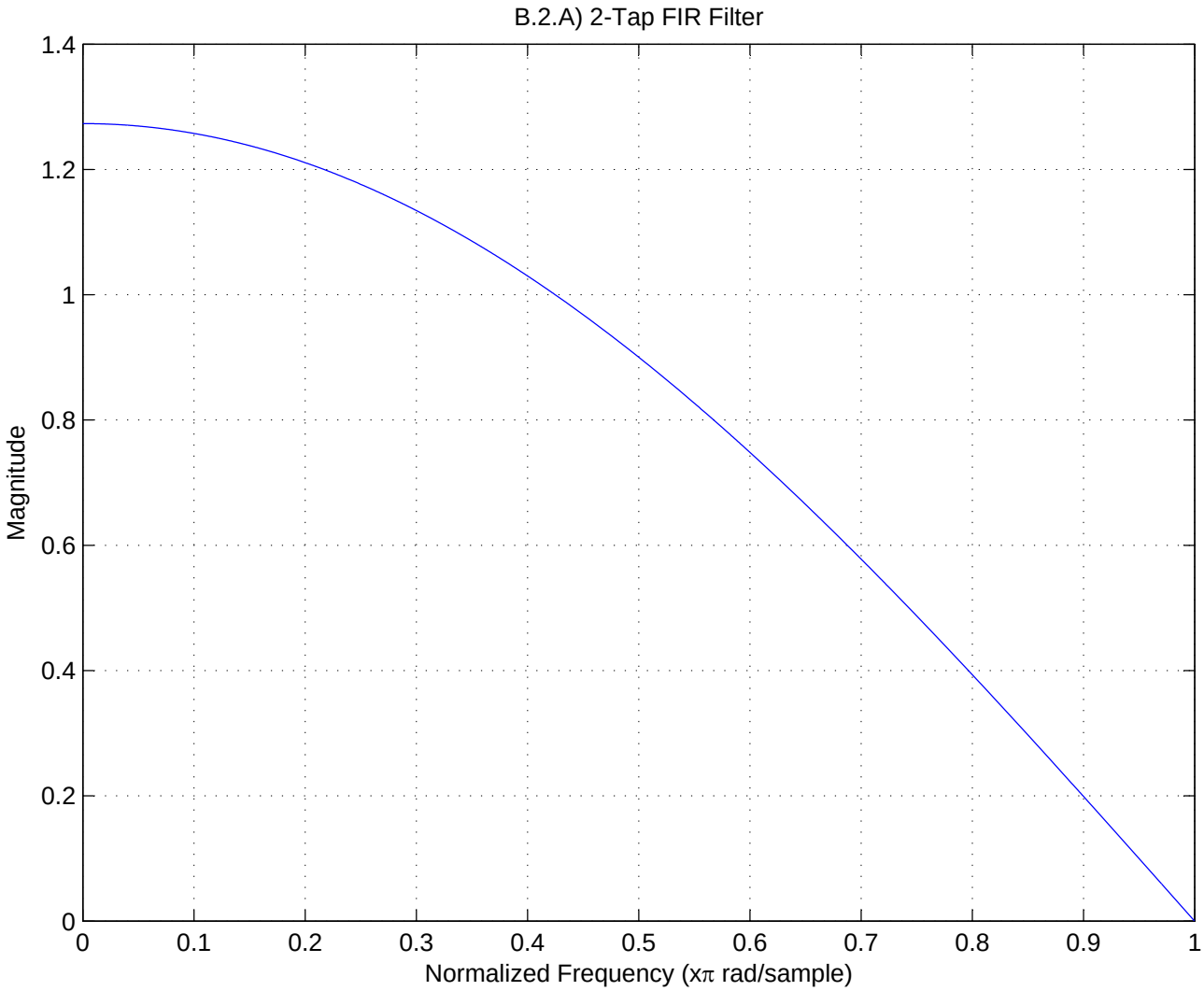
Clearly, the two different values of *a* are just the negative of each other. This negation is equivalent to a phase shift of π without changing the magnitude response. Thus, both solutions are equally optimal for the design specifications. The actual value of the parameter we are minimizing over is just:

$$\boxed{\varepsilon = \int_0^{\pi} \left(\left| H\left(e^{j\omega}\right) - 1 \right| \right)^2 d\omega = 5.951 \times 10^{-1}}$$

Our optimal filter can be any one of the following:



For any of the above four filters, the magnitude response of the transfer function is the same. The following page contains a plot of this magnitude response.


B.2.A) 2-Tap FIR Filter
Magnitude
Normalized Frequency (xπ rad/sample)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0.1
0.3
0.5
0.7
0.9

B.2.A – 4-Tap FIR Filter

The magnitude of a 4-tap FIR filter was shown previously to be one of the following (where the subscript denotes the type of the FIR filter):

$$\left|H_2\left(e^{j\omega}\right)\right| = \left|2a\cos\left(\tfrac{3\omega}{2}\right) + 2b\cos\left(\tfrac{\omega}{2}\right)\right| \quad \text{or} \quad \left|H_4\left(e^{j\omega}\right)\right| = \left|2a\sin\left(\tfrac{3\omega}{2}\right) + 2b\sin\left(\tfrac{\omega}{2}\right)\right|$$

Therefore, all we need to do is run *fminsearch* on the optimization parameter to find the taps.

The following code was used in Matlab to minimize the parameter.

```
% Designing a 4 tap FIR Filter
% x(1) = a
% x(2) = b

% Define the magnitude of the transfer function
mag2 = @(x, w) (abs(2.*x(1).*cos(3.*w/2) + 2.*x(2).*cos(w/2)));
mag4 = @(x, w) (abs(2.*x(1).*sin(3.*w/2) + 2.*x(2).*sin(w/2)));

% Define the parameter to minimize
e2 = @(x) (quad( @(w) ((mag2(x,w) - 1).^2) , 0, pi));
e4 = @(x) (quad( @(w) ((mag4(x,w) - 1).^2) , 0, pi));

% Find the minimum of the minimization parameter
%   Search with r=[-1,1]
for i=1:7
    % get a to jump by 0.5 from [-1.5, 1.5]
    a = (mod((i-1),7)-3)*0.5;
    b = (floor((i-1)/7)-3)*0.5;
    a2(i,:) = fminsearch(e2, [a b]);
    a4(i,:) = fminsearch(e4, [a b]);
end
```

Again, the solutions for the Type II and the Type IV filters are very similar, and both produce the same error in the optimization.

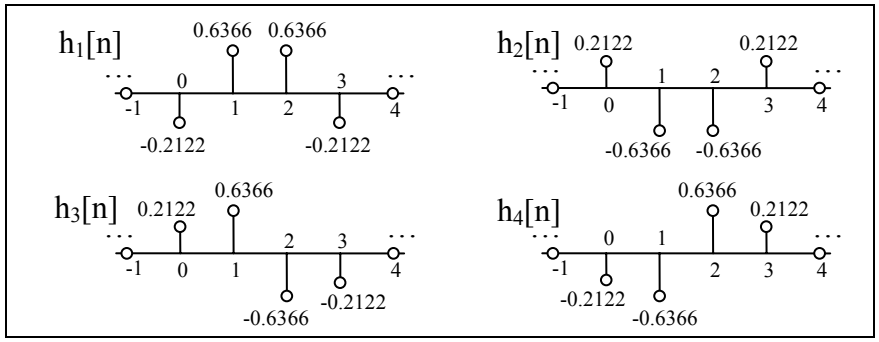
$$\varepsilon = \int_0^{\pi} \left(\left|H\left(e^{j\omega}\right) - 1\right|\right)^2 d\omega = 3.122\times10^{-1}$$

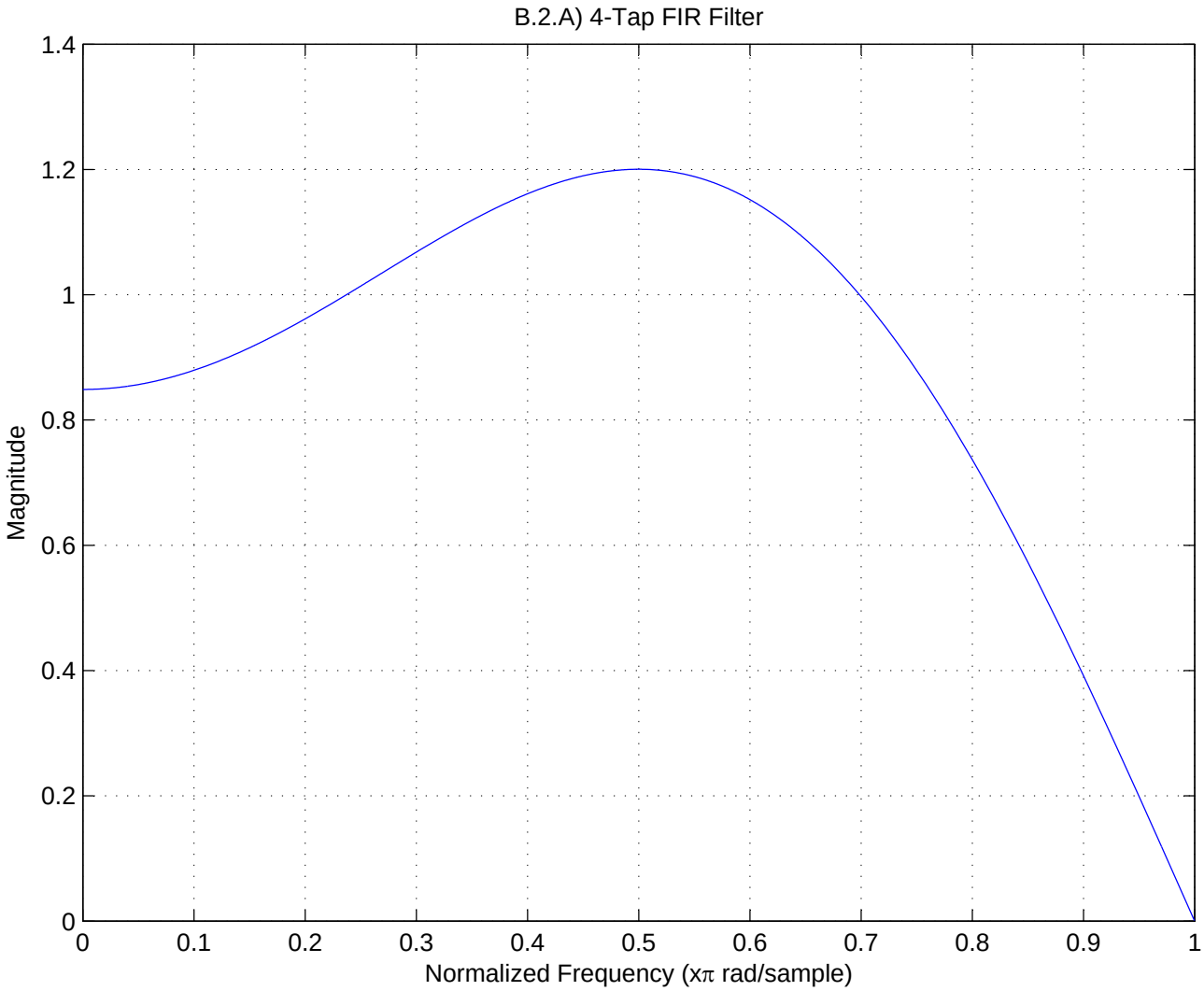
Matlab returns the following solutions for the optimal filter:

Type II: $(a,b) = (-0.2122, 0.6366)$ and $(0.2122, -0.6366)$

Type IV: $(a,b) = (0.2122, 0.6366)$ and $(-0.2122, -0.6366)$

These values for the taps correspond to four different filters. Each of these filters have the same magnitude response (plotted on next page), thus they all equally optimize the design criteria.




B.2.A) 4-Tap FIR Filter
Magnitude
Normalized Frequency (xπ rad/sample)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0.1
0.3
0.5
0.7
0.9

B.2.B – FIR Approximation to All-pass Filters (Minimizing Maximum Error)

Design all-pass filters of order 1 and 2 with Property 1 above and the following minimized:

$$\varepsilon = \max_{\omega\in[0,\pi]} \left| H\left(e^{j\omega}\right) - 1 \right|$$

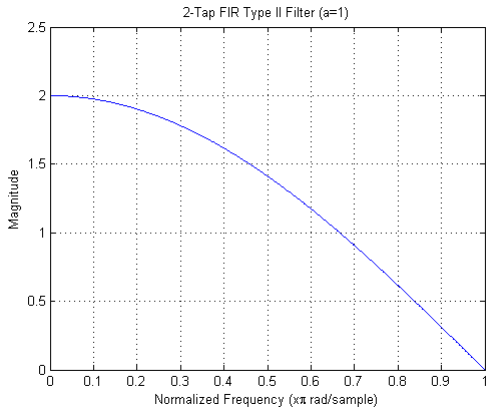
B.2.B – 2-Tap FIR Filter

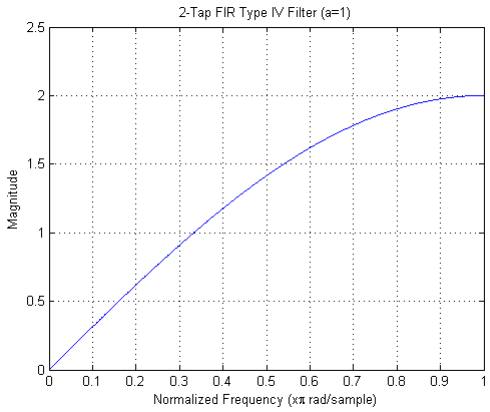
The magnitude of a 2-tap FIR filter was shown previously to be one of the following (where the subscript denotes the type of the FIR filter):

$$\left|H_2\left(e^{j\omega}\right)\right| = \left|2a\cos\left(\tfrac{\omega}{2}\right)\right| \quad \text{or} \quad \left|H_4\left(e^{j\omega}\right)\right| = \left|2a\sin\left(\tfrac{\omega}{2}\right)\right|$$

Blindly running a Matlab script to optimize the error criterion leads to a rather odd solution: any starting point that you choose for a seems to be like a minimum as long as $a<1$. Instead of just running an optimization script, let's examine why this odd phenomenon is happening.

The following plots show the magnitude of these 2-tap FIR filters over the range of $[0, \pi]$.





As you can see, the only parameter we can change here is a, which will stretch or shrink the magnitude in the vertical direction. No matter what the value of a is, the magnitude will always have a zero in both the Type II and the Type IV filters. In both filters, we also note that the maximum value achieved for the magnitude is just $2a$. Therefore, we know that:

$$\left|H_2\left(e^{j\omega}\right)\right|, \left|H_4\left(e^{j\omega}\right)\right| \in \left[0, 2|a|\right]$$

The error criterion is just a mathematical way of saying "minimize the maximum distance to 1". The maximum distance must exist at the lower or upper bounds of the magnitude which are 0 and $2a$. Therefore, the following is rather obvious:

$$\varepsilon = \max_{\omega\in[0,\pi]} \left|H_2\left(e^{j\omega}\right) - 1\right| = \max_{\omega\in[0,\pi]} \left|H_4\left(e^{j\omega}\right) - 1\right| = \max\left(1, \left|\left(2|a| - 1\right)\right|\right)$$

$$\varepsilon = \begin{cases} 1 & , |a| \le 1 \\ 2|a| & , |a| > 1 \end{cases}$$

In other words, to minimize the error criterion, we can choose any a for which $|a| \le 1$. For any of these values of a, the error criterion will always be constant and equal to 1. Therefore, the optimal solutions for the FIR filters are just:

$$\begin{aligned} h_2[n] &= a(\delta[n] + \delta[n-1]) \\ h_4[n] &= a(\delta[n] - \delta[n-1]) \end{aligned} \qquad \forall a \in [-1,1]$$

There are an infinite number of optimal solutions, and thus this error criterion is not a very good parameter to optimize over. This result explains why Matlab chooses the starting search point as the optimal solution to minimize the error criterion. When the optimal solution is chosen, we have:

$$\varepsilon = \max_{\omega \in [0,\pi]} \left| H\left(e^{j\omega}\right) - 1 \right| = 1$$

B.2.B – 4-Tap FIR Filter

The magnitude of a 2-tap FIR filter was shown previously to be one of the following (where the subscript denotes the type of the FIR filter):

$$\left|H_2\left(e^{j\omega}\right)\right| = \left|2a\cos\left(\tfrac{3\omega}{2}\right) + 2b\cos\left(\tfrac{\omega}{2}\right)\right| \quad \text{or} \quad \left|H_4\left(e^{j\omega}\right)\right| = \left|2a\sin\left(\tfrac{3\omega}{2}\right) + 2b\sin\left(\tfrac{\omega}{2}\right)\right|$$

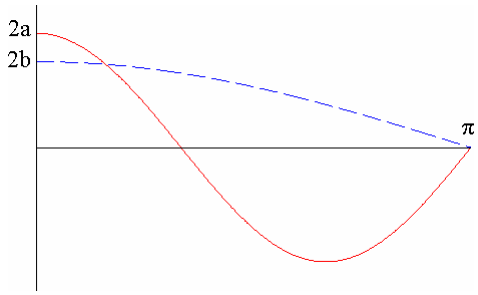
Running a Matlab script here again produced very random answers with no clear optimal solution. Analyzing these magnitude responses will explain these results.

Both of these magnitudes are much harder to analyze than the 2-tap filters because they are no longer monotonically increasing or decreasing. This means that we will need to search for local minima and maxima when optimizing the error criterion.

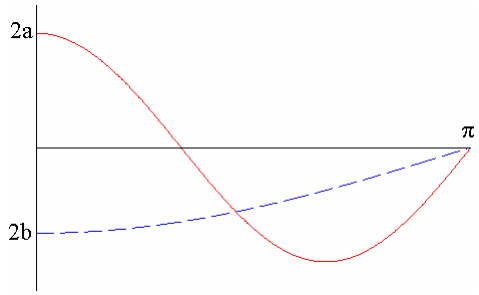
The problem of optimizing the Type II FIR Filter is actually exactly the same as the problem of optimizing the Type IV FIR Filter. Let's first explain why this is true.

The following plots are of each sine or cosine inside the absolute value of each filter.

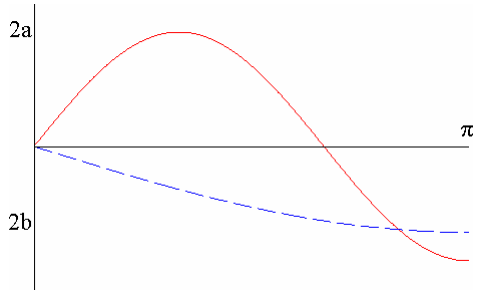
Type II: $sign(a) = sign(b)$



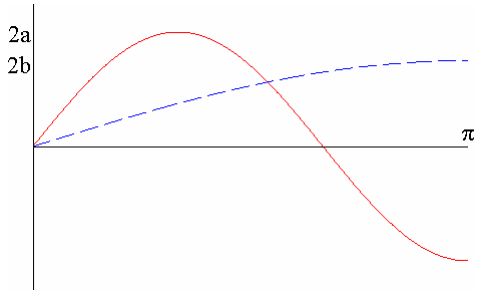
Type II: $sign(a) = -sign(b)$



Type IV: $sign(a) = -sign(b)$



Type IV: $sign(a) = sign(b)$



As you can see, the Type II filter where *a* and *b* are of the same sign and the Type IV filter where *a* and *b* have opposite sign (the two plots on the left) are symmetric about the lines $y = 0$ and $x = \frac{\pi}{2}$ over the interval we are interested in. The same symmetry exists in the two plots on the right. Because of this symmetry, optimizing over the magnitude of the sum of cosines or sines minus some constant is the same in both cases. Thus, we only need to consider one type of filter because the optimization problem is exactly the same for the other. Let us consider the Type II filter for a lack of a better choice.

To find local extrema, we just differentiate this function and set it equal to 0.

$$\frac{d}{d\omega}\left|H_2\left(e^{j\omega}\right)\right| = \left[\frac{d}{d\omega}\left(2a\cos\left(\tfrac{3\omega}{2}\right)+2b\cos\left(\tfrac{\omega}{2}\right)\right)\right] sign\left(2a\cos\left(\tfrac{3\omega}{2}\right)+2b\cos\left(\tfrac{\omega}{2}\right)\right) = 0$$

$$\left[-3a\sin\left(\tfrac{3\omega}{2}\right)-b\sin\left(\tfrac{\omega}{2}\right)\right]\times\left[sign\left(2a\cos\left(\tfrac{3\omega}{2}\right)+2b\cos\left(\tfrac{\omega}{2}\right)\right)\right] = 0$$

$$-3a\sin\left(\tfrac{3\omega}{2}\right)-b\sin\left(\tfrac{\omega}{2}\right) = 0 \quad \text{or} \quad 2a\cos\left(\tfrac{3\omega}{2}\right)+2b\cos\left(\tfrac{\omega}{2}\right) = 0$$

Here, we will apply the lesser known triple-angle trigonometric formulas on both parts:

$$\sin(3\theta) = 3\sin(\theta)-4\sin^3(\theta) \qquad \cos(3\theta) = 4\cos^3(\theta)-3\cos(\theta)$$

(Keep in mind that we are only concerned about the region $\omega\in[0,\pi]$)

$$-3a\sin\left(\tfrac{3\omega}{2}\right)-b\sin\left(\tfrac{\omega}{2}\right) = 0 \qquad 2a\cos\left(\tfrac{3\omega}{2}\right)+2b\cos\left(\tfrac{\omega}{2}\right) = 0$$

$$-3a\left[3\sin\left(\tfrac{\omega}{2}\right)-4\sin^3\left(\tfrac{\omega}{2}\right)\right]-b\sin\left(\tfrac{\omega}{2}\right) = 0 \qquad 2a\left[4\cos^3\left(\tfrac{\omega}{2}\right)-3\cos\left(\tfrac{\omega}{2}\right)\right]+2b\cos\left(\tfrac{\omega}{2}\right) = 0$$

$$\sin\left(\tfrac{\omega}{2}\right)\left[12a\sin^2\left(\tfrac{\omega}{2}\right)-9a-b\right] = 0 \qquad \cos\left(\tfrac{\omega}{2}\right)\left[8a\cos^2\left(\tfrac{\omega}{2}\right)-6a+2b\right] = 0$$

$$\sin\left(\tfrac{\omega}{2}\right) = 0 \quad \text{or} \quad \sin\left(\tfrac{\omega}{2}\right) = \pm\sqrt{\left|\frac{9a+b}{12a}\right|} \qquad \cos\left(\tfrac{\omega}{2}\right) = 0 \quad \text{or} \quad \cos\left(\tfrac{\omega}{2}\right) = \pm\sqrt{\left|\frac{3a-b}{4a}\right|}$$

$$\frac{\omega}{2} = 0, \sin^{-1}\left(\pm\sqrt{\left|\frac{9a+b}{12a}\right|}\right) \qquad \frac{\omega}{2} = \frac{\pi}{2}, \cos^{-1}\left(\pm\sqrt{\left|\frac{3a-b}{4a}\right|}\right)$$

With a little algebra and trigonometry, the magnitude can be simplified to:

$$\left|H_2\left(e^{j\omega}\right)\right| = \left|2a\cos\left(\tfrac{3\omega}{2}\right)+2b\cos\left(\tfrac{\omega}{2}\right)\right| = \left|2a\left[4\cos^3\left(\tfrac{\omega}{2}\right)-3\cos\left(\tfrac{\omega}{2}\right)\right]+2b\cos\left(\tfrac{\omega}{2}\right)\right|$$

$$\left|H_2\left(e^{j\omega}\right)\right| = \left|8a\cos^3\left(\tfrac{\omega}{2}\right)-6a\cos\left(\tfrac{\omega}{2}\right)+2b\cos\left(\tfrac{\omega}{2}\right)\right|$$

Now we will evaluate the magnitude at each of these local extrema to find the global extrema.

$$f_1(a,b) = \left|H_2\left(e^{j\omega}\right)\right|_{\omega/2=0} = \left|8a-6a+2b\right| = \boxed{\left|2a+2b\right|}$$

$$f_2(a,b) = \left|H_2\left(e^{j\omega}\right)\right|_{\omega/2=\pi/2} = \left|8a(0)-6a(0)+2b(0)\right| = \boxed{0}$$

$$f_3(a,b) = \left|H_2\left(e^{j\omega}\right)\right|_{\omega/2=\sin^{-1}\left(\pm\sqrt{\left|\frac{9a+b}{12a}\right|}\right)} = \left|H_2\left(e^{j\omega}\right)\right|_{\omega/2=\cos^{-1}\left(\pm\sqrt{\left|\frac{3a-b}{4a}\right|}\right)} =$$

$$= \left|8a\left[\sqrt{\left|\frac{3a-b}{12a}\right|}\right]^3-6a\sqrt{\left|\frac{3a-b}{12a}\right|}+2b\sqrt{\left|\frac{3a-b}{12a}\right|}\right| = \left|8a\left[\frac{3a-b}{12a}\right]^2-6a+2b\right|\times\sqrt{\left|\frac{3a-b}{12a}\right|}$$

$$= \left|8a\left[\frac{9a^2-6ab+b^2}{144a^2}\right]-6a+2b\right|\times\sqrt{\left|\frac{3a-b}{12a}\right|} = \left|\frac{9a^2-6ab+b^2}{18a}-6a+2b\right|\times\sqrt{\left|\frac{3a-b}{12a}\right|}$$

$$f_3(a,b) == \boxed{\left|\frac{-99a^2+30ab+b^2}{18a}\right|\times\sqrt{\left|\frac{3a-b}{12a}\right|}}$$

Therefore, we can find the global minimum and maximum of the magnitude to be:

$$\min\left(\left|H_2\left(e^{j\omega}\right)\right|\right) = \min\left(f_1, f_2, f_3\right) = 0$$

$$\max\left(\left|H_2\left(e^{j\omega}\right)\right|\right) = \max\left(f_1, f_2, f_3\right) = \max\left(\left|2a+2b\right|, \left|\frac{-99a^2+30ab+b^2}{18a}\right| \times \sqrt{\left|\frac{3a-b}{12a}\right|}\right)$$

It can be shown (but is not done here), that the following is true:

$$\max\left(\left|H_2\left(e^{j\omega}\right)\right|\right) = \begin{cases} \left|2a+2b\right| & , sign(a) = sign(b) \\ \left|\frac{-99a^2+30ab+b^2}{18a}\right| \times \sqrt{\left|\frac{3a-b}{12a}\right|} & , sign(a) = -sign(b) \end{cases}$$

Thus,

$$\left|H_2\left(e^{j\omega}\right)\right| \in \left[0, \max\left(\left|H_2\left(e^{j\omega}\right)\right|\right)\right]$$

Therefore, for a similar argument as the 2-tap filter, the $\max\left(\left|H_2\left(e^{j\omega}\right)\right|\right)$ has to be chosen such that it is less then or equal to 2 to belong to the set of optimal solutions. In other words, *a* and *b*, must be chosen such that they satisfy:

1) If [(Type II FIR) and (*sign(a)=sign(b)*)] or [(Type IV FIR) and (*sign(a)=–sign(b)*)] then
$$\left|a+b\right| \le 1$$

2) If [(Type II FIR) and (*sign(a)=–sign(b)*)] or [(Type IV FIR) and (*sign(a)=sign(b)*)] then
$$\left|\frac{-99a^2+30ab+b^2}{18a}\right| \times \sqrt{\left|\frac{3a-b}{12a}\right|} \le 2$$

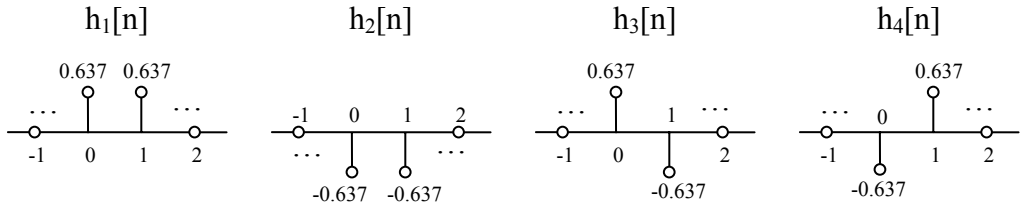
Any *a* and *b* that satisfy either of the above conditions will be an optimal solution. Similar to the 2-tap case, there are an infinite number of optimal solutions to the problem of minimizing:

$$\varepsilon = \max_{\omega \in [0,\pi]} \left|H\left(e^{j\omega}\right) - 1\right| = 1$$

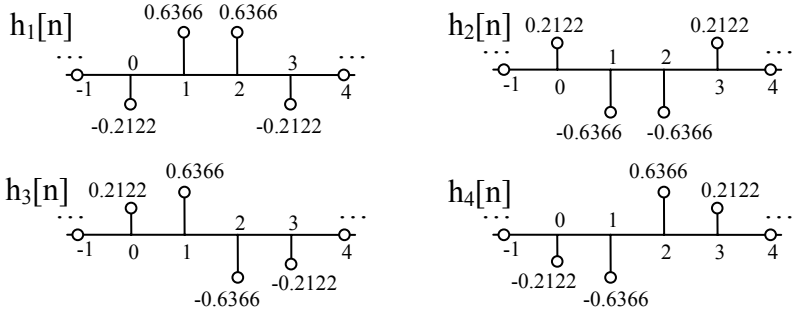
B.2 – Observations and Comments

The optimal solutions from section B.2.A are as follows:

2-Tap FIR Optimal Filters



4-Tap FIR Optimal Filters

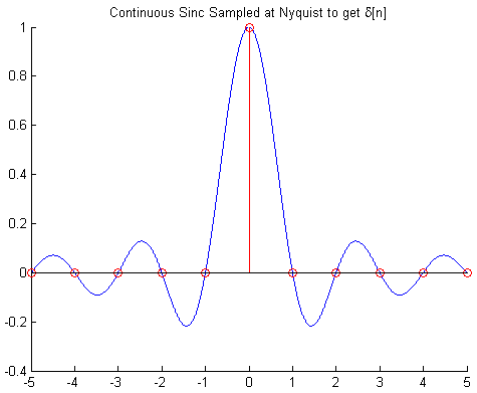


If you notice, the only two taps in the 2-tap filters correspond perfectly with the middle two taps in the 4-tap filters. Let's consider $h_1[n]$, and explore this a little further, because we are very close to a great finding!

If we think about the best possible magnitude response to minimize $\varepsilon = \max_{\omega\in[0,\pi]} \left| H\left(e^{j\omega}\right) - 1 \right|$, then we can think of the transfer function in two different ways. The first thought for the most optimal solution (ignoring how many taps there were) would be the following:

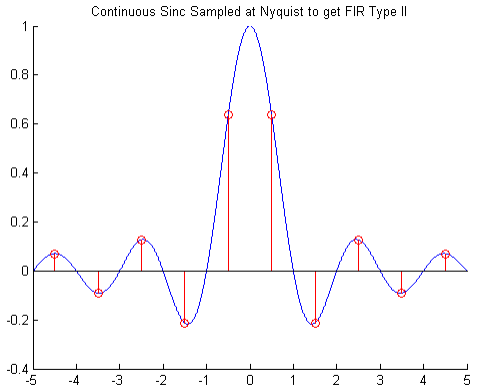
$$h[n] = \delta[n] \Leftrightarrow H\left(e^{j\omega}\right) = 1$$

However, looking a little deeper, $\delta[n]$ is just a *sinc* function sampled at the right places (sampling at exactly the Nyquist rate picks off the peak and the zeroes). The figure on the next page shows this clearly. This $\delta[n]$ and sampled *sinc* parallelism exists in the frequency domain as well. The transform of $\delta[n]$ is a constant 1, which is basically the same as a rectangle with width 2π (because the DTFT is periodic with period 2π). We know that the inverse DTFT of the rectangle is just our *sinc*. Therefore, we can think of the ideal h[n] to just be a sampled sinc instead of the simple $\delta[n]$.



In the problem statement, we are restricted to even length filters. If we could use odd length filters, the optimal solution would have been just a delta function because the transform would be constant.

However, the delta function would have been optimal for a much deeper reason: it is actually an infinitely long sequence of the sampled *sinc* (just that all samples besides the middle one are zero). When we start using even length FIR filters, we no longer can have a perfectly sampled *sinc* because we only have a finite number of taps. The even length samples do not lie on the zeroes, and thus, we have to cut off the samples at a certain length (in our case, 2 and 4).



Therefore, our optimal even length FIR filter, is formed in the same way as the optimal odd length FIR filter: just sample the *sinc* for as many samples as our length. This hypothesis matches exactly with our optimal even length FIR filters!

$$\operatorname{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$$

$$\operatorname{sinc}(0.5) = \operatorname{sinc}(-0.5) = \frac{2}{\pi} \approx 0.6366$$

$$\operatorname{sinc}(1.5) = \operatorname{sinc}(-1.5) = -\frac{2}{3\pi} \approx -0.2122$$

It is no coincidence that our "optimal" even length FIR filters have exactly the same values as sampling the *sinc* function right in between the integer samples. From this argument, it is trivial to construct any even length FIR filter because all we do is pick off values from the *sinc*. The length of the filter just describes how many values to sample the *sinc* at.

Unlike section B.2.A, there is not much to say about section B.2.B. From the optimal designs in section B.2.B, we saw that there were always an infinite number of solutions. This is based on a bad optimization criterion. Section B.2.A was a better way to optimize the filters because it worried about the average magnitude response, and tried to get that as close to unity as possible. Worrying about the maximum error is sometimes a good way to optimize, but was a poor design criterion in this case. According to this criterion, a filter that has zero magnitude response everywhere is as good as an ideal notch filter that has zero magnitude at one point. Obviously, the ideal notch filter is a much better approximation to the ideal all-pass filter. However, because this criterion only worries about the maximum error, the better choice is no more optimal then the worse choice. In this specific case, the optimization criterion was not good because the maximum distance included a zero of the function. If the interval was instead chosen to be over the range of $[\pi/4, 3\pi/2]$, the results may have been more interesting.