# ECE 569 Resubmission of Project 1

Bandpass FIR/IIR Filters
Professor: Geoffrey Williamson

Nader Alnatsheh
A20452265
Due Date: 04/22/2022

# Objective:

The objective of the project is to design a Linear-phase Bandpass FIR filter using three different designs "Windowing *method with the Hamming Window, Frequency Sampling, and the Equiripple via the Remez Algorithm*", also, designing an IIR Bandpass filter using three different designs "Butterworth, *Chebyshev type I, and Elliptic design"*, so all six different designs meet an exact specific requirement. The project was Implemented using the MATLAB software. A total of nine codes were written to get the figures appears in the appendices section. The following sections are divided as follows*: I. Introduction to filter design and bandpass filter, II. Results of the FIR/IIR filter designs and a detail description of each design procedure in each case, III. Comparison between the six different designs and the advantages, disadvantage of each one of them, IV. Conclusion, V. References, VI. Appendices.*

# I. Introduction:

Filter design is a deep topic in DSP, and this comes because it forms the backbone of 90% of what we do in audio. There are two types of filters: *Finite Impulse Response (FIR) and Infinite Impulse Response (IIR)* and there are four main shapes of filters with Bandpass filter being the most frequently used filter among the other. A bandpass filter which may also be called a band-select filter because it selects a specific frequency range to pass a signal unattenuated and eliminate the other frequencies. One of its many applications is that they are used in a communication system for choosing the signals with a particular bandwidth.

# II. Results of the FIR/IIR filters designs and a detail description of each design procedure:

All the filter designs meet a specific Requirements as below:

- passband of $\Omega p = [0.15\pi, 0.4\pi]$;
- stopband of $\Omega s = [0, 0.1\pi] \cup [0.45\pi, \pi]$;
- passband tolerance $\delta p = 0.06$;
- stopband tolerance $\delta s = 0.02$.

## [1]: Linear-Phase FIR Filters:

**A) Windowing method with the Hamming Window:**

**Results:**

Figure 1 and 2 in the appendices section shows the frequency response and the phase response respectively of the filter that was designed using the Windowing method with the Hamming window.

**Detail Description:**

1. I started the design by first setting the Sampling frequency to **Fs =pi**, and the lower and upper cutoff frequencies respectively to **Wn = [0.125*pi 0.425*pi]/Fs**.

2. After that I used the command: **s = fir1(n,Wn).**

3. The command **fir1()**; uses a least-squares approximation to compute the filter coefficients and then smooths the impulse response with window, but it does need an integer number to define the number of coefficients it is going to compute, hence, I set the filter order **N.** The n in the fir1

command tell MATLAB software that I want to compute the coefficients of the nth order FIR filter, which in turn will calculate nth+1 coefficients.

4. After that I calculated and then plotted the frequency response of the filter designed by using the command **freqz**, which determines the transfer function evaluated at $z = e^{j\omega}$. and returns the n-point complex frequency response $H(e^{j\omega})$ for the designed digital filter.

5. Finally, the linear-phase response was plotted using the command **angle** which returns the phase angle and the **unwrap** command was used because it is difficult to distinguish the 360° jumps from the 180° jumps so the unwrap command unwraps the phase to make it continuous across 360° phase discontinuities by adding multiples of ±360°.

6. Applying this procedure with these band edges and with N = 90 results in a filter that meets the specifications. The filter order N was gradually reduced, and the smallest N for which I could adjust the band edges to successfully meet the design criteria was **N = 88**. The cutoff band edges frequencies that used were **f = 0.125 and f = 0.425**.

**B) Frequency-sampling method:**

**Results:**
Figure 3 and 4 in the appendices section shows the frequency response and the phase response respectively of the filter that was designed using the Frequency-sampling method.

**Detail Description:**
1. First, I started the design by setting the vector **f;** which is a vector of frequency points in the range from 0 to 1, where 1 corresponds to the Nyquist frequency. The first point in f must be 0 and the last point 1 and the frequency points must be in increasing order. So, I **had f = [ 0 0.1 0.15 0.40 0.45 1],** where the between frequencies is the passband and stopband edges.

2. Second, I set the vector **m** which is a vector containing the desired magnitude response at the points specified in **f**. Note here that s and m must be the same length . So, I set **m = [ 0 0 1 1 0 0].**

3. Third, I used the command **fir2(),** which designs frequency sampling-based digital FIR filters with arbitrarily shaped frequency response. By using this command, MATLAB software will return a new row vector containing the nth+1 coefficients of an order n FIR filter. The frequency-magnitude characteristics of this filter match those given by vectors f and m.

4. Fourth, I calculated the frequency response of the filter using the command **freqz** and plot it, the phase response was plotted too using the commands **angle** and **unwrap**.

5. Applying this procedure with these band edges and with N = 150 results in a filter that meets the specifications. The filter order N was gradually reduced, the smallest N for which I could adjust the band edges to successfully meet the design criteria was **N = 130.** The band edges frequencies that used were **f = [ 0 0.1 0.15 0.4 0.45 1]** & **m = [0 01 1 1 0 0]**.

**C) Optimum Equiripple design via the Remez Algorithm:**

**Results:**
Figure 5 & 6 in the appendices section shows the frequency response and the phase response respectively of the filter that was designed using the optimum Equiripple design via the Remez Algorithm

**Detail Description:**
**The first two steps are exactly the same as the Frequency Sampling Method procedure**

- Third, I used the command **remez;** which designs a linear-phase FIR filter using the Parks-McClellan algorithm, the Parks-McClellan algorithm uses the Remez exchange algorithm and Chebyshev approximation theory to design filters with an optimal fit between the desired and actual frequency responses. The filters are optimal in the sense that the maximum error between the desired frequency response and the actual frequency response is minimized. Filters designed this way exhibit an equiripple behavior in their frequency responses and are sometimes called *equiripple* filters, remez exhibits discontinuities at the head and tail of its impulse response due to this equiripple nature.

- Fourth, I calculated the frequency response of the filter using the command **freqz** and plot it and the phase response was plotted too using the commands **angle** and **unwrap**.

- Applying this procedure with these band edges and with N = 90 results in a filter that meets the specifications. The filter order N was gradually reduced, the smallest N for which I could adjust the band edges to successfully meet the design criteria was **N = 70.** The band edges frequencies that used were **f = [ 0 0.1 0.15 0.4 0.45 1]** & **m = [0 01 1 1 0 0]**.

## [2]: Nonlinear-Phase IIR Filters:

**D) Butterworth filter:**

**Results:**
Figure 7 and 8 in the appendices section shows the frequency response and the phase response respectively of the filter that was designed using the Butterworth method.

**Detail Description:**
1. First, I started by calculating/setting the passband ripple and stopband attenuation respectively to dB values as **Rp=0.5374** and **Rs=33.79**, setting the frequency sampling **Fs= pi**, setting **Wp**; the passband edges to **[0.15*pi 0.40*pi]**, and setting **Ws** the stopband edges to **[0.1*pi 0.45*pi].**

2. Second, I used the command **buttord,** which will return the lowest order, n, of the digital Butterworth filter with no more than Rp dB of passband ripple and the least Rs dB of attenuation in the stopband.

3. Thirdly, I used the command **[A, B, C, D] = butter();** which that will design a Butterworth digital filter and returns the matrices that specify its state-space representation. A note here is that Butterworth filters have a magnitude response that is maximally flat in the passband and monotonic overall. This smoothness comes at the price of decreased rolloff steepness.

4. Fourth, I calculated the frequency response of the filter using the command **freqz** and plot it and the phase response was plotted too using the commands **angle.**

5. Applying this procedure steps above with the band edges requirements, I got the result from **buttord** command a value of **N = 9,** which is the minimum filter order that meets the specifications.

**E) Chebyshev type I filter:**

**Results:**
Figure 9 and 10 in the appendices section shows the frequency response and the phase response respectively of the filter that was designed using the Chebyshev Type I method.

**Detail Description:**
****The first two steps are exactly the same as the Butterworth filter, but instead I used the command **cheb1ord** to calculate the minimum filter order that will achieve the required specifications**

- Thirdly, I used the command **[A, B, C, D] = cheyb1();** which will design a Cheybshev Type I digital filter and returns the matrices that specify its state-space representation. A note here is that Chebyshev Type I filters are equiripple in the passband and monotonic in the stopband.

- Fourth, I calculated the frequency response of the filter using the command **freqz** and plot it and the phase response was plotted too using the commands **angle.**

- Applying this procedure steps above with the band edges requirements, I got the result from **cheyb1ord** command a value of **N = 5,** which is the minimum filter order that meets the specifications.

**F) Elliptic Filter:**

**Results:**
Figure 11 and 12 in the appendices section shows the frequency response and the phase response respectively of the filter that was designed using the Elleptic method.

**Detail Description:**
****The first two steps are exactly the same as the Butterworth filter, but instead I used the command **ellipord** to calculate the minimum filter order that will achieve the required specifications**

- Thirdly, I used the command **[A, B, C, D] = ellip();** which will design an elliptic digital filter and returns the matrices that specify its state-space representation. A note here is that Elliptic filters offer steeper roll off characteristics than Butterworth or Chebyshev filters but are equiripple in both the passband and the stopband. In general, elliptic filters meet given performance specifications with the lowest order of any filter type.

- Fourth, I calculated the frequency response of the filter using the command **freqz** and plot it and the phase response was plotted too using the commands **angle.**

- Applying this procedure steps above with the band edges requirements, I got the result from **ellipord** command a value of **N = 4,** which is the minimum filter order that meets the specifications.

**Q: Why IIR filters have a Non-linear phase response?**
**A:** The IIR Filters have a phase response that vary wildly from each other, we can see that the Butterworth filter has a phase response that is almost linear in the passband, but this line then does not behave linearly in the transition band, on the other hand the Chebyshev & the Elliptic filter don't have a flat passband or stopband, and hence the phase response is nowhere near linear in either of these areas. This is because IIR filters are not BIBO stable, and that is mean that their poles are not in the unit circuit,

and so in order to get a linear phase IIR filter, we need conjugate reciprocal poles outside of the unit circle, making it BIBO unstable. And hence it is impossible to design a Linear-phase IIR filters.

# III. Comparison between the six different designs and the advantages/disadvantage of each one of them.

To better compare between the three FIR methods presented, I had to plot them all in one figure, and so Figure 13 shows the three methods of designing an FIR bandpass filters plotted all in one figure.

- **Windowing with Hamming Window:** The major disadvantage of the windowing method is the lack of precise control of the critical frequencies, such as wp and ws, in the design of a lowpass FIR.

- **Frequency Sampling:** very simple but not very effective, because much longer filter lengths are required over optimized methods such as least squared and equiripple (Remez) filter design. The coefficients of the FIR filter are the inverse DFT of the sampled frequency response desired. A note here is that the solution will be exact at the frequency sample locations, with ripple in between. Hence, this type of filter methods is used when an application requires an exact solution at discrete frequency locations, and we don't care what happens in between.

- **Optimum Equirriple via Remez Algorithm:** The Remez/Parks McLellan method produces a filter which just meets the specification without overperforming. The advantage is that it produces a linear-phase FIR filters that minimize the weighted Chebyshev error, also it provides explicit control of band edges and relative ripple sizes, and hence it's an efficient algorithm that always converges.

Similarly, to better compare between the three IIR methods presented, I had to plot them all in one figure, Figure 14 shows the three methods of designing an IIR bandpass filters plotted all in one figure.

- **The Butterworth Filter** is commonly referred to as the "maximally flat" option because the passband response offers the steepest roll-off without inducing a passband ripple. In addition to the flat passband response, the selectivity of the Butterworth filter is better than many other filter types. The disadvantage of this improved selectivity is greater delay and poorer phase linearity. Butterworth filters offer solid performance considering the number of components needed to implement the filter.

- **Chebyshev Type I Filter:** is known for its ripple response, which can be designed to be present in the passband region and hence it's a Chebyshev Type I. The amplitude of this ripple is directly proportional to steepness of the roll off, so if we want a steeper response, we will see a larger ripple response. These filters offer performance between that of Elliptic function filters and Butterworth filters.

- **Elliptic Filter:** they have the sharpest roll-off compared to all other filters, which comes with the unintended consequence of ripples in both the passband and stopband. The passband ripple of the elliptic filter is similar to the Chebyshev filter; however, the selectivity is greatly improved. The disadvantage to this improved selectivity is a more complex filter network that requires more components.

**For the sake of curiosity Figure 15 shows the six different designs methods of Linear Phase FIR and IIR bandpass filters plotted all in one figure.**

# IV. Conclusion:

In conclusion, a linear phase digital FIR filter was designed and implemented via MATLAB software using three different methods; "Windowing with Hamming Window, Frequency Sampling and Equiripple via the Remez Algorithm", similarly, digital IIR filters was designed and implemented using three different methods; "Butterworth, Chebyshev Type I, and Elliptic" so they all meet a specific requirement. two comparisons were described: one between the three FIR methods, and the second comparison between three IIR methods, the advantages and disadvantages of each method was discussed too.

# V. References:


[1]: https://www.mathworks.com/help/signal/ug/fir-filter-design.html

[2]: https://www.mathworks.com/help/signal/ug/iir-filter-design.html


# VI. Appendices:

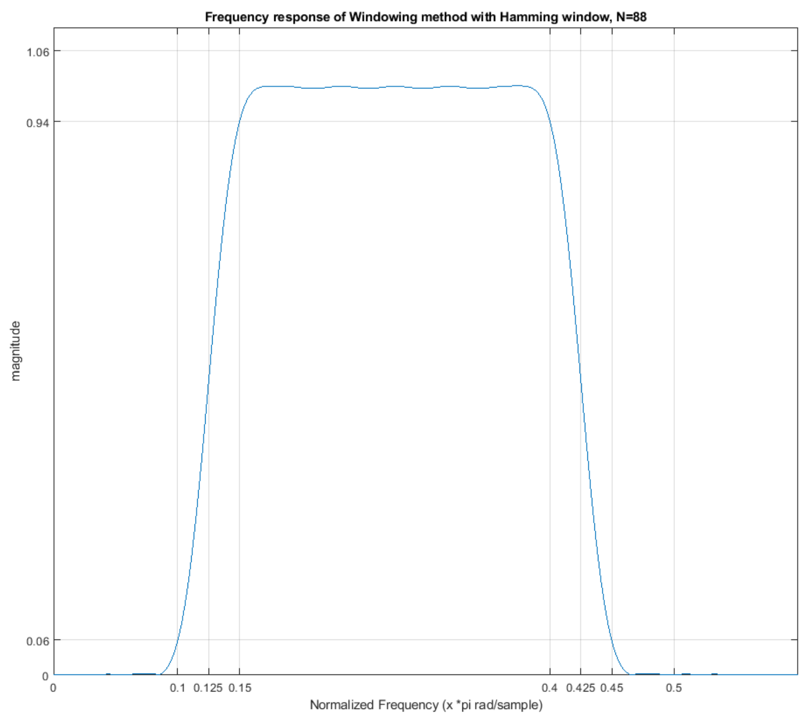


Figure 1: Frequency Response of Hamming Window Method

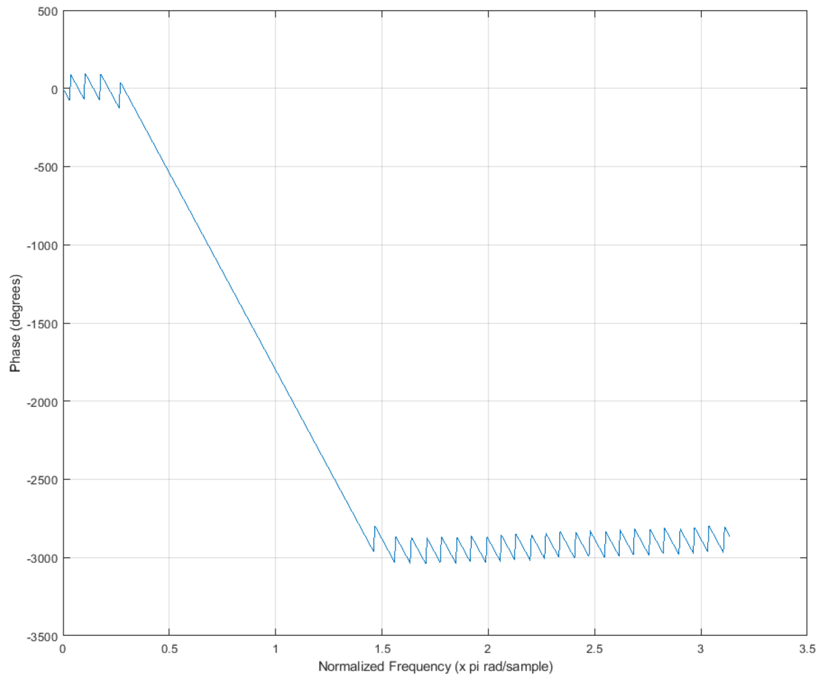


Figure 2: Magnitude Response of Hamming Window Method

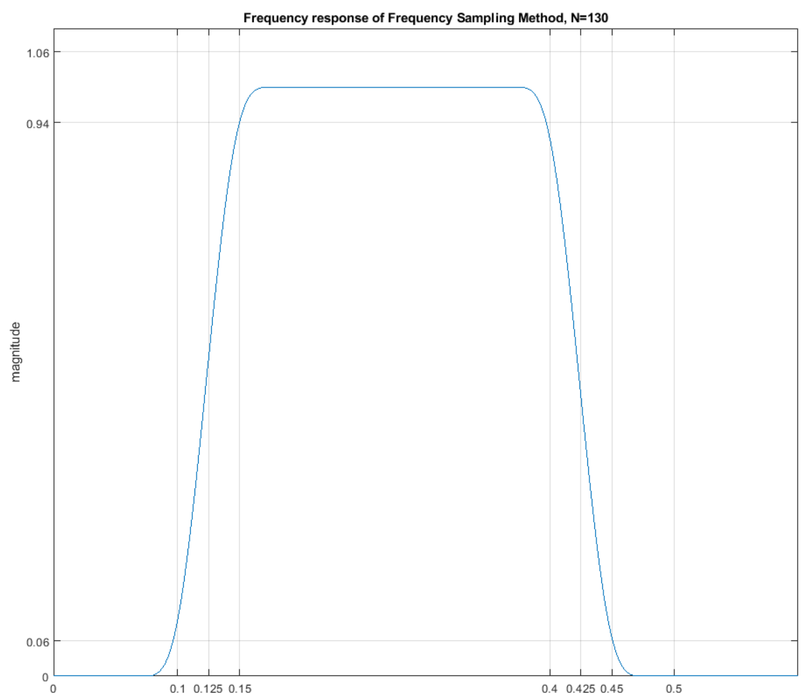


Figure 3: Frequency Response of Frequency Sampling Method

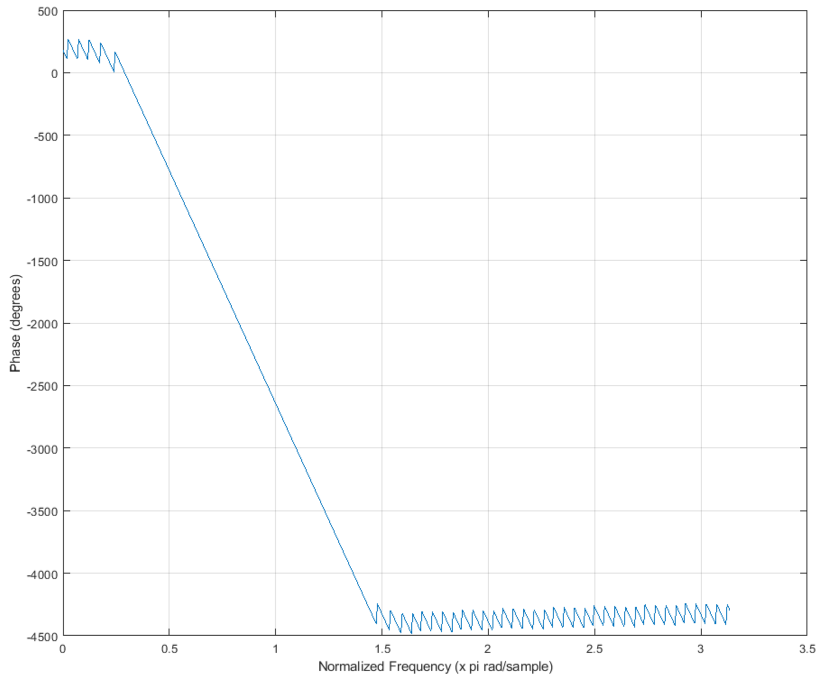


Figure 4: Magnitude Response of Frequency Sampling Method

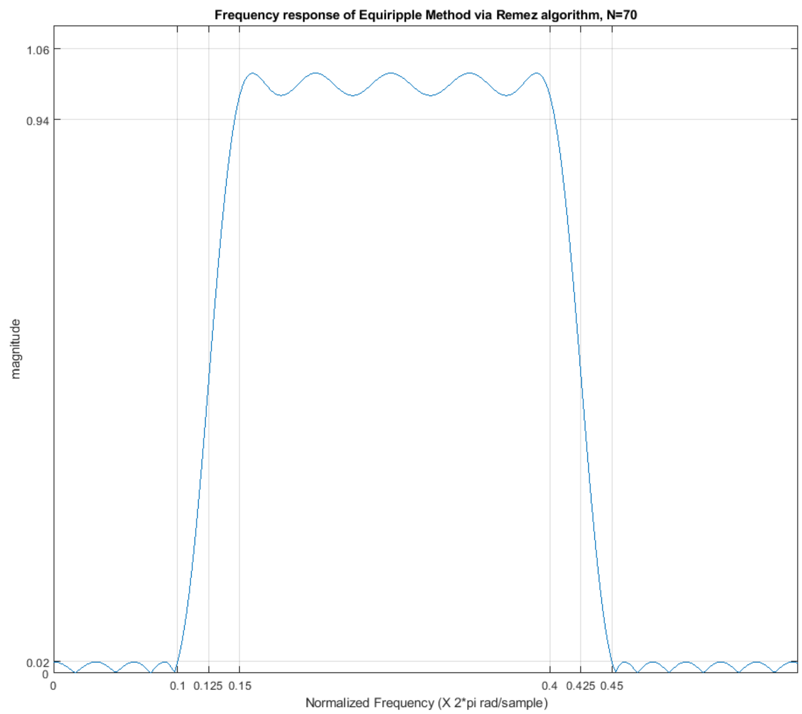


Figure 5: Frequency Response of Equiripple Method via Remez Algorithm

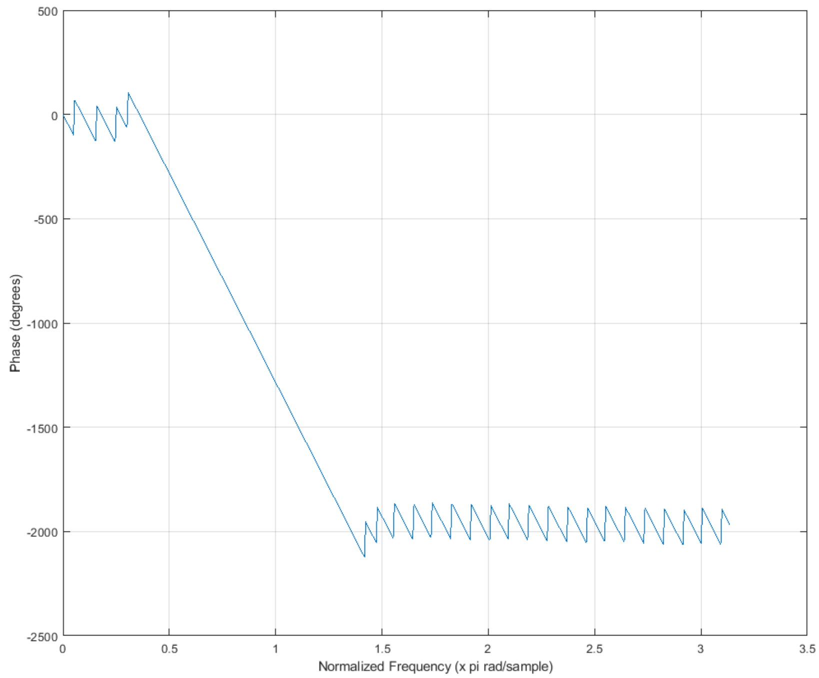


Figure 6: Magnitude Response of Equiripple Method via Remez Algorithm

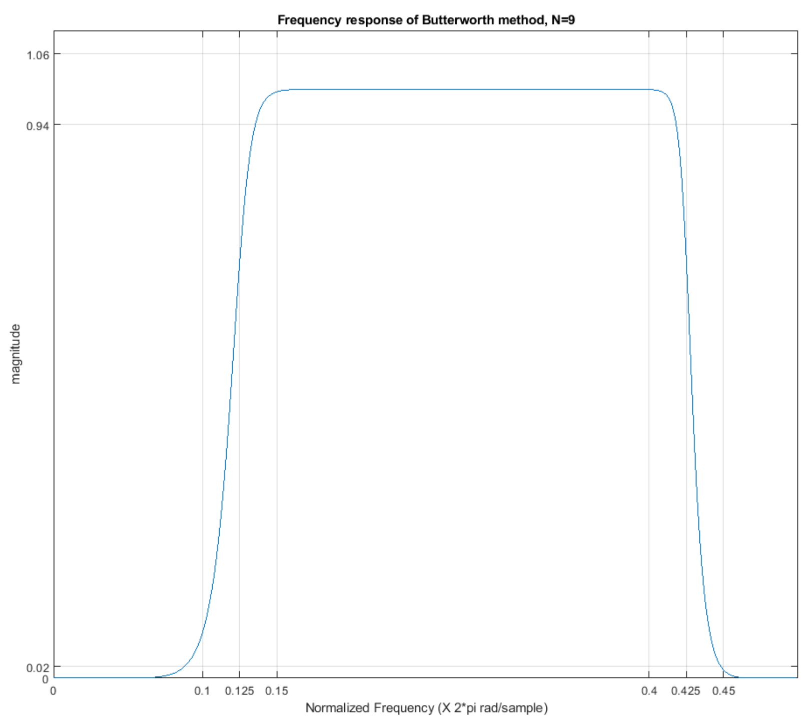


Figure 7: Frequency Response of Butterworth Method

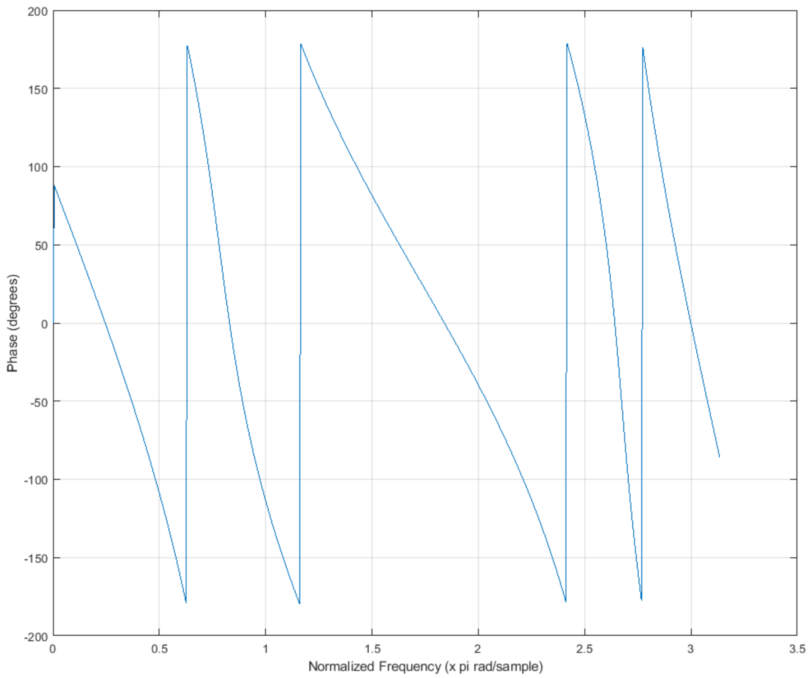


Figure 8: Magnitude Response of Butterworth Method

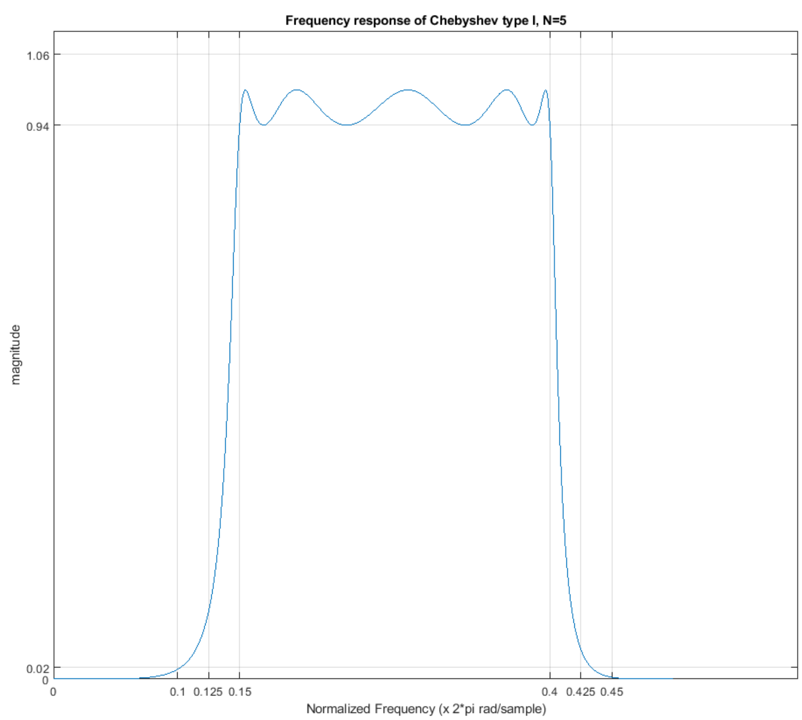


Figure 9: Frequency Response of Chebyshev type I Method

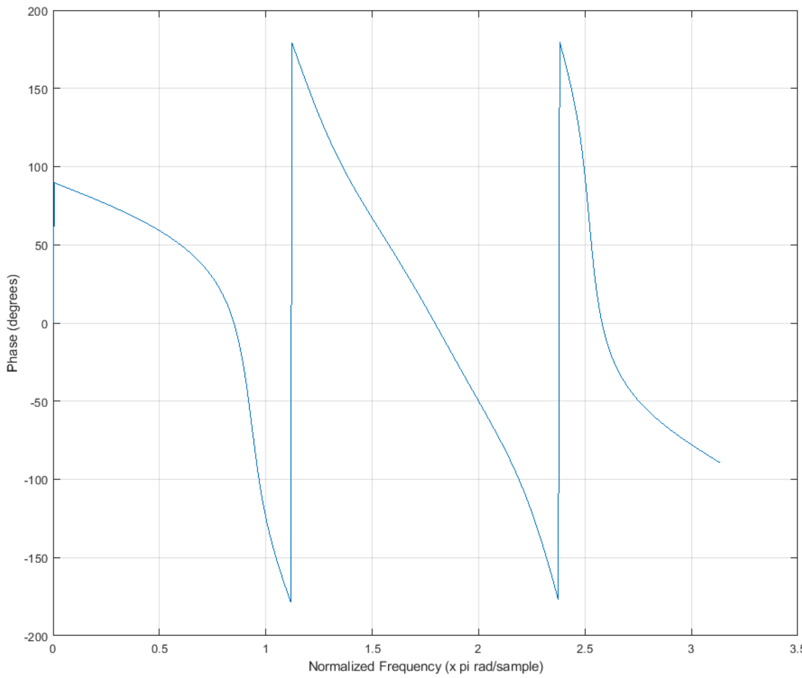


Figure 10: Magnitude Response of Chebyshev type I Method

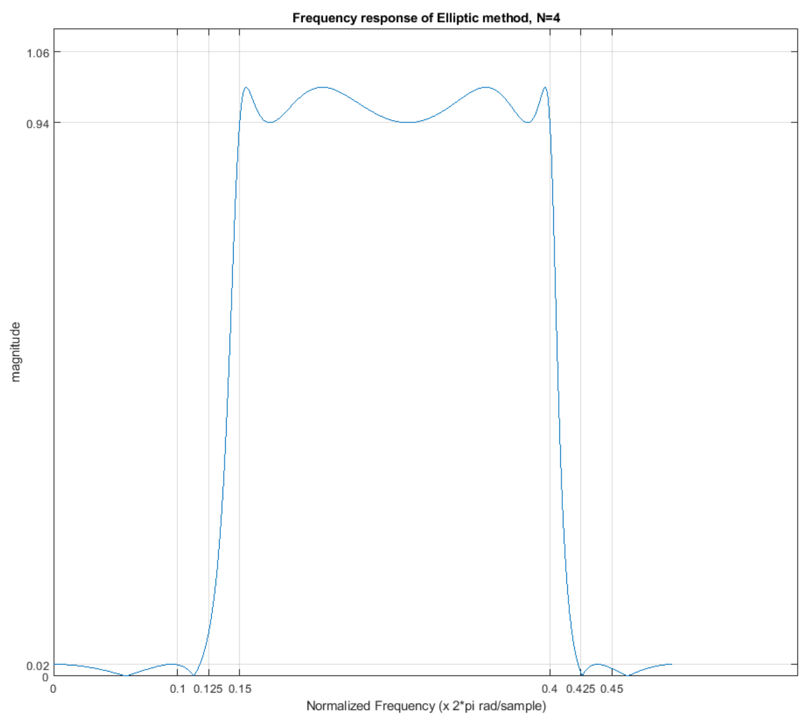


Figure 11: Frequency Response of Elliptic Method

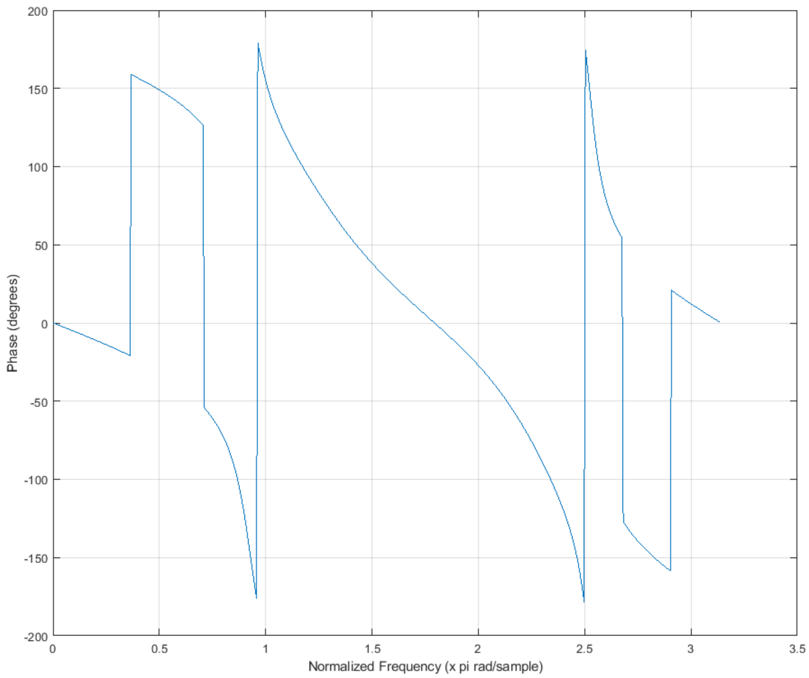


Figure 12: Magnitude Response of Elliptic Method

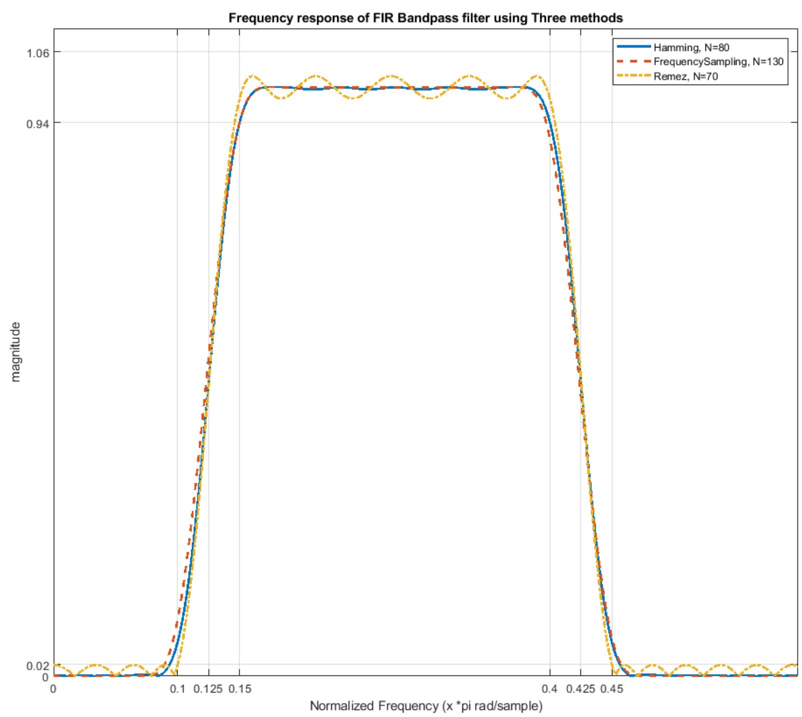


Figure 13: Frequency Response of all FIR Methods

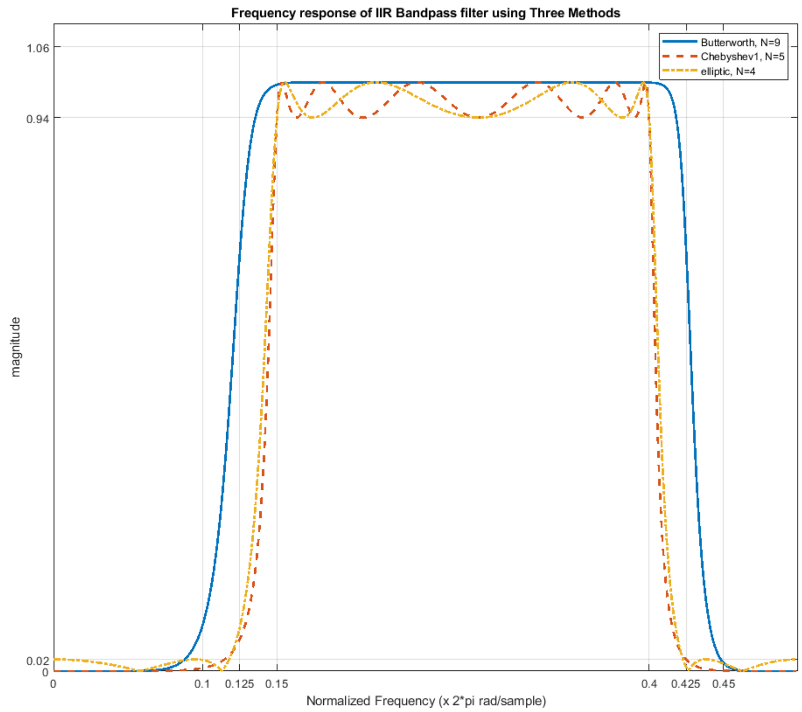


Figure 14 Frequency Response of all IIR Methods

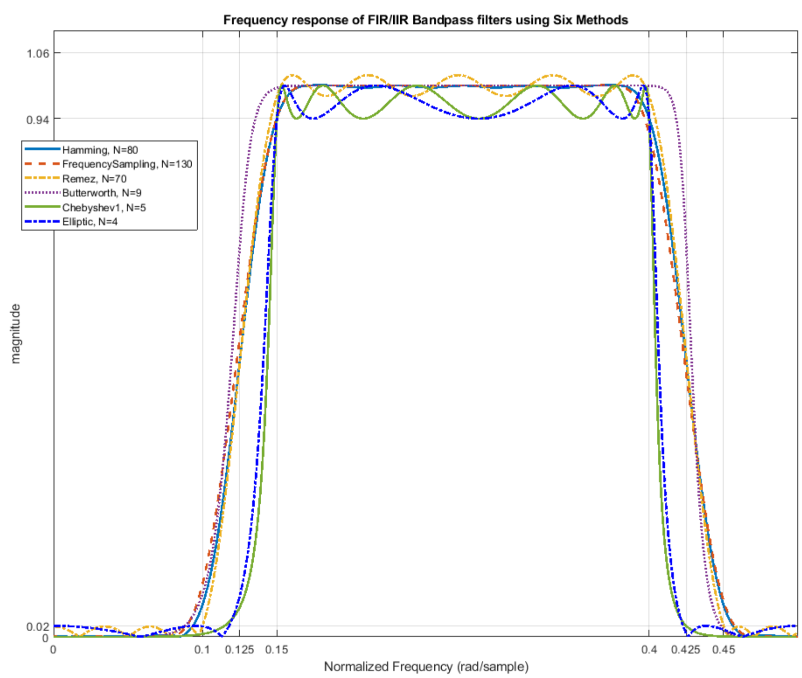


Figure 15: Frequency Response of the FIR & IIR Methods