Filtration with IIR for Equalization & Replacing

Group Members Information:

Roll No	Name of Each Member	Email
2k18/TCT/23	Hafiz Muhammad Attaullah	attaullahshafiq10@gmail.com
2k18/TCT/99	Umm-e-Farwa	ummyfarwa110@gmail.com
2k18/TCT/09	Aiza	aizaabbasi911@gmail.com
2k18/TCT/12	Alishba Butt	alishbab673@gmail.com

DEPARTMENT: Telecommunications Engg. (Faculty of Engg: & Technology)

CLASS: BS Telecommunications (3rd year)

SUBJECT: Digital Signal Prosessing

TEACHER: Dr. Sundar Ali Khuwaja[1]

GitHub Repo[2]: https://github.com/attaullahshafiq10/dsp-mini

---

[1] Assistant Professor @ faculty of Engineering & Technology, University of Sindh

[2] This Code is also available at our GitHub repo, Link [https://github.com/attaullahshafiq10/dsp-mini]

Contents

Filtration Process with IIR

(Analysis of IIR Filtration in terms of Replacing and minimum order)

1 Abstract

IIR (infinite impulse response) filters are usually used for applications where the linear process is not too large and memory is constrained. They have been widely deployed in audio equalization, sensor signal processing, IoT/IIoT smart sensors and high-speed telecommunication or RF applications and form a crucial building block in algorithmic design. This code shows how easy it is to replace a Butterworth design with either a Chebyshev or an elliptical filter of the same order and to obtain a steeper roll-off at the expense of some ripple in the passband and/or stopband of the filter. The minimum order designs are explored after this.

2 Introduction & Background of the Project

The IIR filter is categorized by its theoretically infinite impulse response:

$$y(n)=\sum_{k=0}^{\infty}h(k)x(n-k)$$

Practically speaking, it is not possible to calculate the output of the IIR using this equation. The equation may therefore be re-written in terms of the finite number of poles p and zeros q, as defined by the linear constant coefficient difference equation given by:

$$y(n)=\sum_{k=0}^{q}b(k)x(n-k)-\sum_{k=1}^{p}a(k)y(n-k)$$

Where a(k) and b(k) are the denominator and numerator polynomial coefficients of the filter, the roots of which are equal to the poles and zeros of the filter. The relationship between the difference equation and the z-transform (transfer function) can therefore be defined by using the z-transform delay property.

2.1.1 Butterworth

Butterworth filters have a magnitude response that is maximum flat in the passenger band and overall monotonous, making them an excellent choice for DC and load cells. However, this highly desirable 'smoothness' comes at the price of reduced roll-off steepness. As a result, the Butterworth method has the slowest roll-off characteristics of all the methods.

2.1.2 Elliptic

Elliptic filters are steeper than Butterworth and Chebyshev filters, but they are equipment for both the passband and stopband. Elliptic filters generally comply with the design requirements in the lowest order of all the methods discussed.

2.1.3 Chebyshev Type I

Chebyshev Type I filters are equiripple in the passband and monotone in the stopband. As such, filters of type I roll out quicker than filters of type II and Butterworth Chebyshev, but to the detriment of wider ribbon ribbons.

2.1.4 Chebyshev Type II

Chebyshev Type II filters are monotonous in the passband and equiripple in the stopband, making them a good choice for bridge sensor applications. Although filters designed using the

Type II method are slower to roll-off than those designed using the Chebyshev Type I method, roll-off is faster than those designed using the Butterworth method.

3 Similar Projects and Literature Review

We have arranged and studied these all projects. And generate this code accordingly. This code is basically cited from a code used in python[3] and we have generated this as MATLAB code.

3.1.1 IIR Filter Design Using the Helper Modules with Python

The Scipy package signal assists in the design of many digital filter types. In this Project IIR filter designed with jupyter notebook. Here is the link for complete details: https://scikit-dsp-comm.readthedocs.io/en/latest/nb_examples/FIR_and_IIR_Filter_Design.html

3.1.2 Compensate for the Delay Introduced by an IIR Filter

In this Project, Filtering a signal will cause a delay. This means that the output signal is shifted to the input in time. So Infinite impulse response filters delay some of the frequency components more than others. They are effectively distorting the input signal. The filter function compensates for the delays introduced by such filters and thus corrects the distortion of the filter. This "zero-phase filtering" is the result of filtering the signal in the forward and backward directions.

Here is the link to find this Project: https://www.mathworks.com/help/signal/ug/compensate-for-the-delay-introduced-by-an-iir-filter.html

3.1.3 Digital IIR Filter Design Using Differential Evolution Algorithm (Springer-2005)

In this work, the DE algorithm was applied to the design of digital IIR filters and its performance was compared to that of a genetic algorithm. Here is the link to access[4] this paper: https://link.springer.com/article/10.1155/ASP.2005.1269

3.1.4 Electronic Filter Design

In this book they keep up with major developments in Electronic Filter Design, including the latest developments in both analogue and digital filters. McGraw-classic Hill's Electronic Filter Design Handbook, long established as the "Bible" of practical electronic filter design, has now been completely revised and updated for a new generation of design engineers. Link:[5] https://www.accessengineeringlibrary.com/content/book/9780071471718

---

[3] https://scikit-dsp-comm.readthedocs.io/en/latest/nb_examples/FIR_and_IIR_Filter_Design.html

[4] Citation: Karaboga, Nurhan. "Digital IIR filter design using differential evolution algorithm." *EURASIP Journal on Advances in Signal Processing* 2005.8 (2005): 856824.

[5] Williams, Arthur B., and Fred J. Taylor. *Electronic filter design handbook*. McGraw-Hill Education, 2006.

4 Aims and Objectives of Project

How to design classical IIR filters. The initial focus is on the situation in which the critical design parameter is the cut-off frequency at which the filter power decays to half (-3 dB) of the nominal passband value.

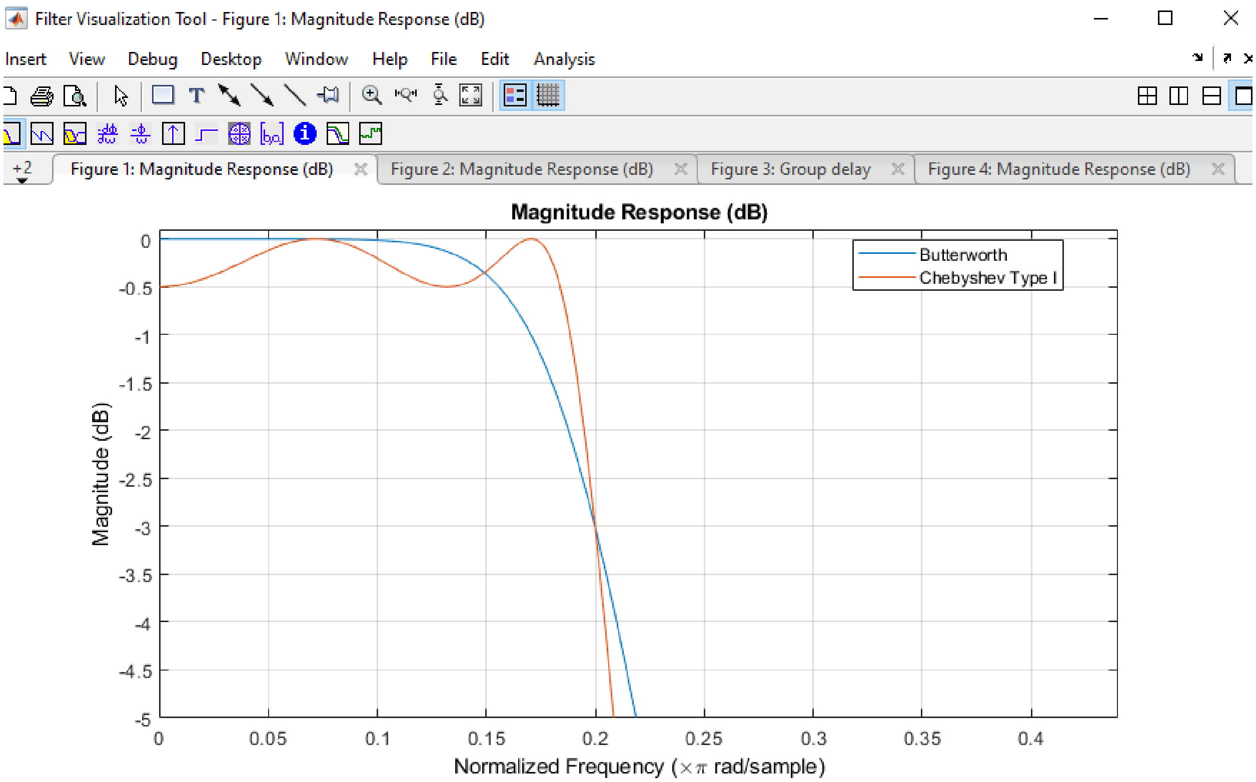
How easy it is to replace the Butterworth design with either a Chebyshev or an elliptical filter of the same order and to obtain a steeper roll-off at the expense of some ripple in the passband and/or stop band of the filter. The minimum order designs are explored after this.

5 Tools

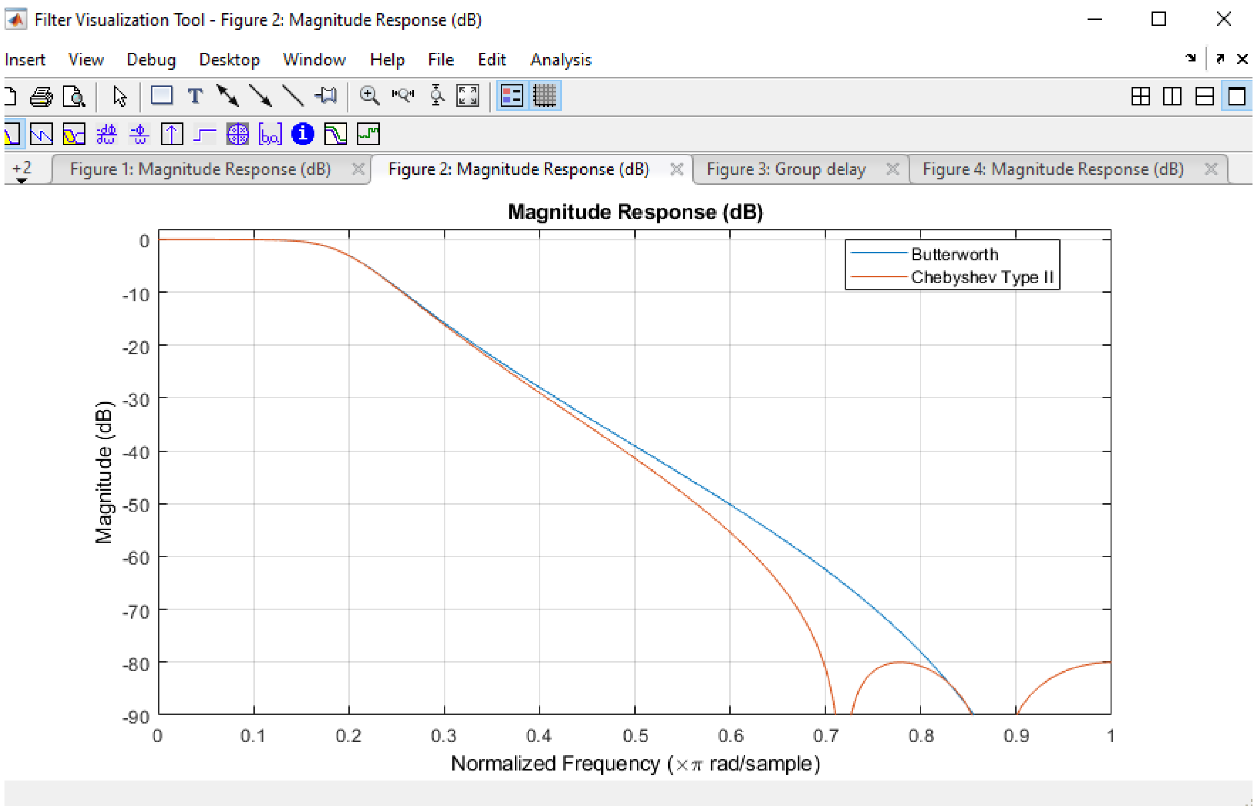
- MATLAB (R2020a)
- DSP Systems Toolbox
- Filter Design Toolbox

6 Result and Discussions

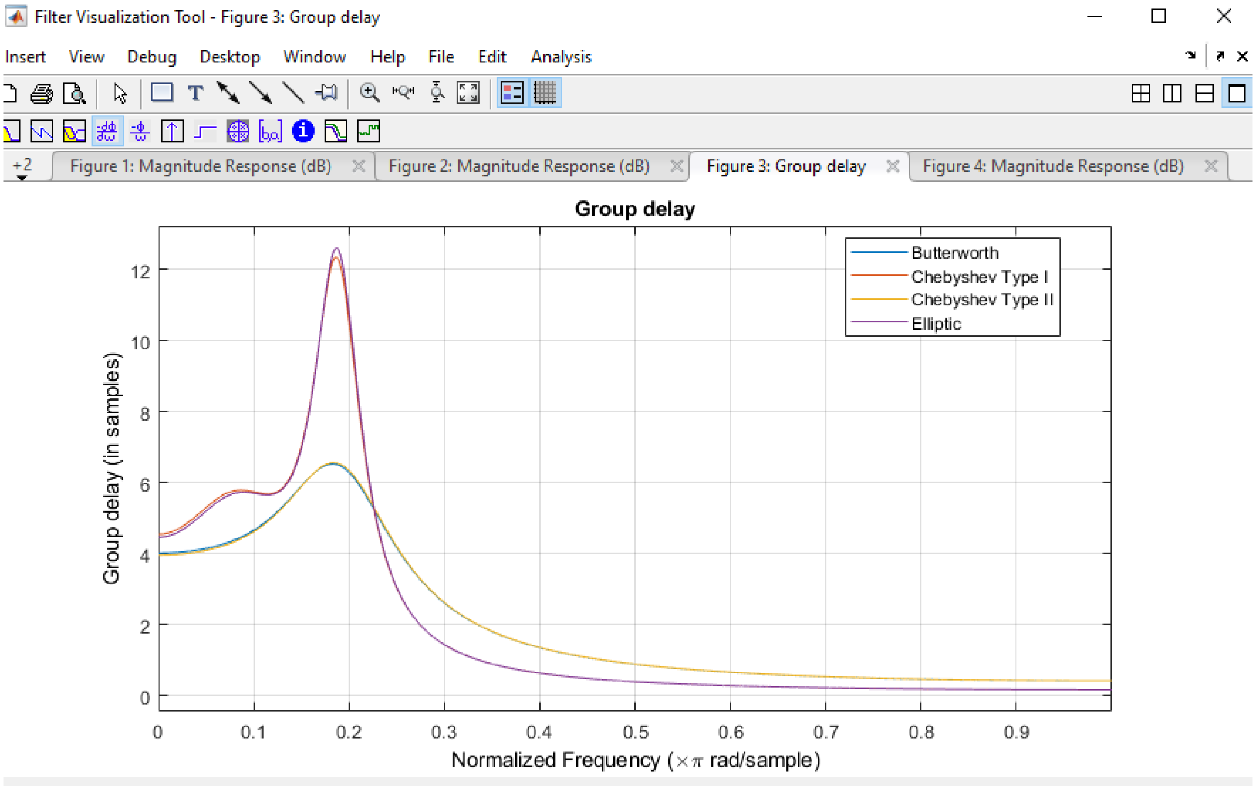
As we mentioned earlier, it is easy to replace the Butterworth design with either a Chebyshev or an elliptical filter of the same order and to obtain a steeper roll-off at the expense of some ripple in the passband and/or the filter stop band. And also mentioned the minimum order designs are explored. So first of all, it explored 8th order filter with a normalized cut-off frequency of 0.4pi. Then we design a Butterworth filter that is as flat as possible (no ripple in the passband or in the stopband). After this the Chebyshev Type-I design makes it possible to control the ripples in the passband. There's still no ripples in the stop band. Larger ripples allow a steeper roll-off. So, we specify a peak-to-peak ripple of 0.5dB.



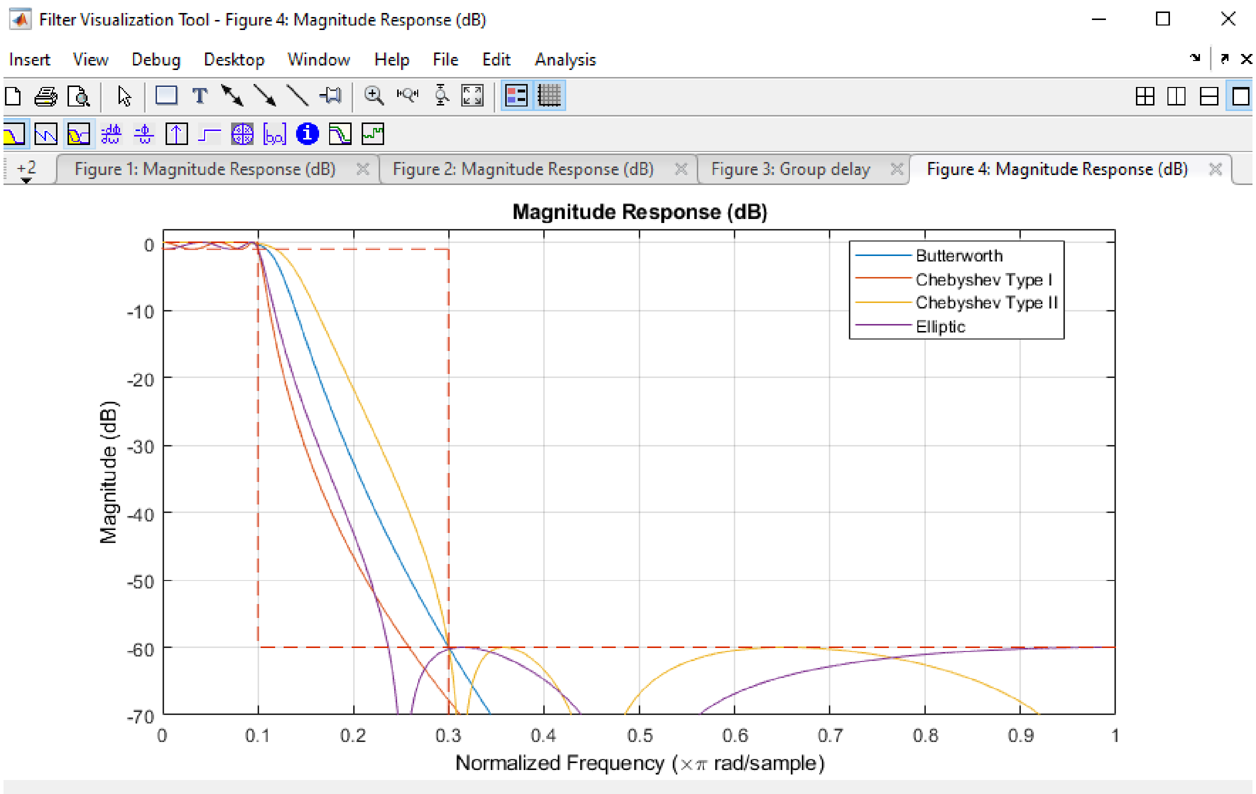
Now the Chebyshev Type II design allows the stopband attenuation to be controlled. In the passband, there are no ripples. A smaller stopband attenuation enables a steeper roll-off. In this example, a stopband attenuation of 80 dB is specified:



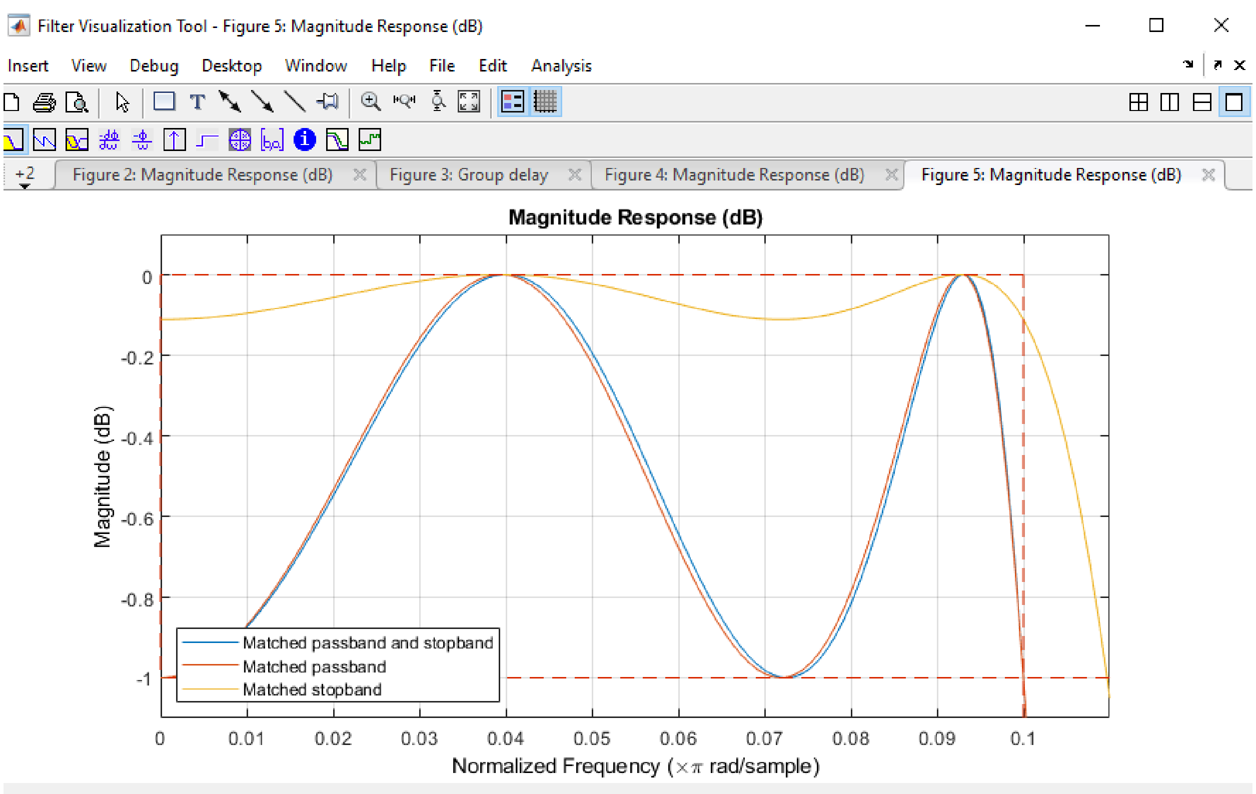
Finally, an elliptical filter can provide a steeper roll-off compared to previous designs by allowing both the stopband and the passband to ripple. To illustrate this, we reuse the same type of passband and stopband as above



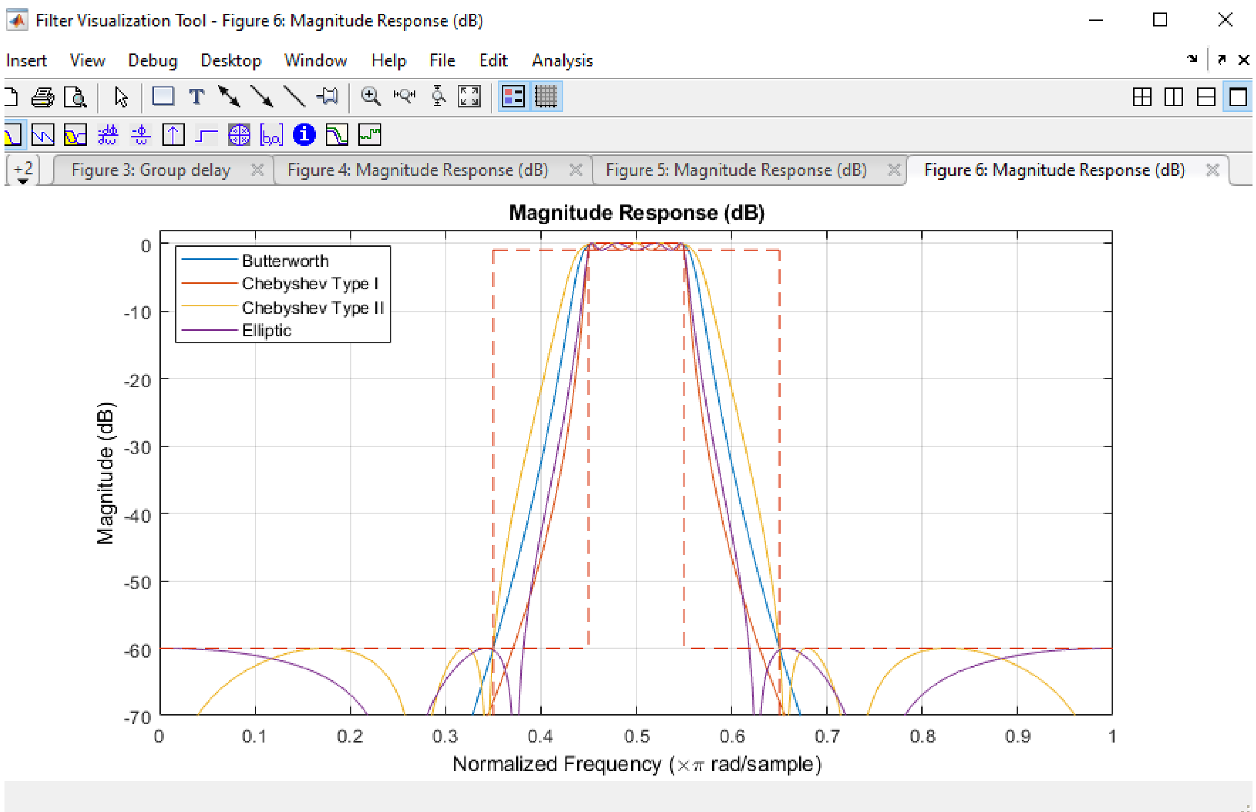
Now we check that all filters have the same-3dB frequency point and that only Butterworth and Chebyshev Type II designs have a perfectly flat passband, for this we used zooming in the passband.



Now we are using Group delay for phase consideration.



Then applied to minimum Order designs in terms of frequencies and the amount of tolerable ripples. After this it extended to highpass, bandpass and bandstop response types. So its minimum order bandpass filters.



7 Complete Program with some specific comments[6]:

```
% Filtration with IIR for Equalization
% Analysis of IIR Filtration in terms of Replacing and minimum order
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Authors:
%   Hafiz Muhammad Attaullah - attaullahshafiq10@gmail.com
%   Alishba Butt - alishbab673@gmail.com
%   Aiza Abbasi - aizaabbasi911@gmail.com
%   Umm-e-Farwa - ummyfarwa110@gmail.com
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lowpass Filters
% First, we design a Butterworth filter
fdB = .2;
X = 4;
d = fdesign.lowpass('N,F3dB',X,fdB);
```

[6] https://github.com/attaullahshafiq10/dsp-mini

```
Hbutter = design(d,'butter','SystemObject',true);
% Now Chebyshev, to allow ripples in the passband to be controlled. No waves
on the stopband are yet.
Ap = .5;
setspecs(d,'N,F3dB,Ap',X,fdB,Ap);
Hcheby1 = design(d,'cheby1','SystemObject',true);
hfvt = fvtool(Hbutter,Hcheby1,'Color','white');
axis([0 .44 -5 .1])
legend(hfvt,'Butterworth','Chebyshev Type I');
%Chebyshev TypeII design It allows the stopband attenuation to be managed.
Ast = 80;
setspecs(d,'N,F3dB,Ast',X,fdB,Ast);
Hcheby2 = design(d,'cheby2','SystemObject',true);
hfvt = fvtool(Hbutter,Hcheby2,'Color','white');
axis([0 1 -90 2])
legend(hfvt,'Butterworth','Chebyshev Type II');
% Finally, An elliptical filter can provide a steeper roll-off compared to
previous designs by allowing both the stopband and the passband to ripple.
setspecs(d,'N,F3dB,Ap,Ast',X,fdB,Ap,Ast);
Hellip = design(d,'ellip','SystemObject',true);
hfvt = fvtool(Hbutter,Hcheby1,Hcheby2,Hellip,'Color','white');
axis([0 1 -90 2])
legend(hfvt, ...
    'Butterworth','Chebyshev Type I','Chebyshev Type II','Elliptic');
% Now set the access
axis([0 .44 -5 .1])
% Group Delay checking
hfvt.Analysis = 'grpdelay';

% Minimum Order Designs
Ap = 1;
Ast = 60;
Fp = .1;
Fst = .3;

setspecs(d,'Fp,Fst,Ap,Ast',Fp,Fst,Ap,Ast);
Hbutter = design(d,'butter','SystemObject',true);
Hcheby1 = design(d,'cheby1','SystemObject',true);
Hcheby2 = design(d,'cheby2','SystemObject',true);
Hellip = design(d,'ellip','SystemObject',true);
hfvt = fvtool(Hbutter,Hcheby1,Hcheby2,Hellip, 'DesignMask', 'on',...
    'Color','white');
axis([0 1 -70 2])
legend(hfvt, ...
    'Butterworth','Chebyshev Type I','Chebyshev Type II','Elliptic');

% 7th, 6th, 5th order (Chebyshev techniques)
order(Hbutter)
order(Hcheby1)
order(Hcheby2)
order(Hellip)

% Now Matching the Passband or Stopband Specifications
Hellipmin1    = design(d, 'ellip', 'MatchExactly', 'passband',...
    'SystemObject',true);
Hellipmin2 = design(d, 'ellip', 'MatchExactly', 'stopband',...
```

```
    'SystemObject',true);
hfvt = fvtool(Hellip, Hellipmin1, Hellipmin2, 'DesignMask', 'on',...
    'Color','white');
axis([0 1 -80 2]);
legend(hfvt, 'Matched passband and stopband', ...
    'Matched passband', 'Matched stopband', ...
    'Location', 'Northeast')

% compare passband edges
axis([0 .11 -1.1 0.1]);
legend(hfvt, 'Location', 'Southwest')

% verifying that, resulting order of filters are not changed
order(Hellip)
order(Hellipmin1)
order(Hellipmin2)

% Now finally. Highpass, Bandpass and Bandstop Filters
d = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2', ...
    .35,.45,.55,.65,60,1,60);
Hbutter = design(d,'butter','SystemObject',true);
Hcheby1 = design(d,'cheby1','SystemObject',true);
Hcheby2 = design(d,'cheby2','SystemObject',true);
Hellip = design(d,'ellip','SystemObject',true);
hfvt = fvtool(Hbutter,Hcheby1,Hcheby2,Hellip, 'DesignMask', 'on',...
    'Color','white');
axis([0 1 -70 2])
legend(hfvt, ...
    'Butterworth','Chebyshev Type I','Chebyshev Type II','Elliptic',...
    'Location', 'Northwest')
% end
```

8 References

Classical IIR filter design: a practical guide, https://www.advsolned.com , accessed on 11 jan 2021

Karaboga, Nurhan. "Digital IIR filter design using differential evolution algorithm." *EURASIP Journal on Advances in Signal Processing* 2005.8 (2005): 856824.

Williams, Arthur B., and Fred J. Taylor. *Electronic filter design handbook*. McGraw-Hill Education, 2006.

Classic IIR Filter Design, https://www.mathworks.com , accessed on 12,13,14,15,16 January 2021

Compensate for the Delay Introduced by an IIR Filter, https://www.mathworks.com , accessed on,14 and 15 January 2021

https://github.com/mwickert/scikit-dsp-comm , accessed on 16 January 2021

Minimum order for elliptic filters, https://www.mathworks.com , accessed on 16 January 2021

https://www.mathworks.com/help/signal/ref/cheby1.html , accessed on 15 January 2021