FILTER DESIGN ASSIGNMENT

Digital Signal Processing

MARCH 31, 2022

Badal Varshney

19D070015

EE 338: Filter Design Assignment

Abstract

This report is made as a part of the Evaluation Component 1 of the Digital Signal Processing happening in Spring Semester: Jan-Apr 2022 at IIT Bombay. This report summarizes the filter design techniques for designing a bandpass and a bandstop filter of Butterworth and Chebyshev nature.

Author

Badal Varshney

Group-16, Filter Number-27

Roll Number: 19D070015

Department of Electrical Engineering

Indian Institute of Technology Bombay

Instructor

Prof. Vikram.M.Gadre

Department of Electrical Engineering

Indian Institute of Technology Bombay

Peer Reviewer

Yeshwanth sunnapu

Roll no: 19D070071

Department of Electrical Engineering

Indian Institute of Technology Bombay

Contents

Filter-1 (Bandpass) details

1.Unnormalized Discrete time specifications

Filter number: 27

Since filter number is < 80, m = 27.

q(m) = greatest integer less than 0.1m = floor(0.1m) = 2

r(m) = m – 10q(m) = 7

BL(m) = 10 + 5 q(m) + 13 r(m) = 111

BH(m) = BL(m) + 45 = 156

The first filter is given to be a Band-Pass filter with passband from BL(m) to BH(m) kHz. Therefore, the specifications are:

- Passband: 111 kHz to 156 kHz
- Transition band: 3 kHz on either side of passband
- Stopband: 0 to 108 kHz and 159 kHz to 270 kHz (Sampling rate is 540 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Monotonic
- Stopband Nature: Monotonic

2.Normalized Digital Filter Specifications

Sampling rate: 540 kHz

In the normalized frequency axis, sampling rate corresponds to 2π. Thus, any frequency (Ω), up to 270 kHz can be represented on the normalized axis ω as:

$$\omega = \frac{\Omega * 2\pi}{\Omega_s(Sampling\ rate)}$$

Therefore, the corresponding normalized discrete filter specifications are:

- Passband: 0.4111π to 0.5778π
- Transition band: 0.0111π on either side of passband
- Stopband: 0 to 0.40π and 0.5889π to π (Sampling rate is 540 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband

- Passband Nature: Monotonic
- Stopband Nature: Monotonic

3.Analog filter specifications for Band-pass filter using Bilinear transformation

We use the following transformation to move from the *z* domain to the *s* domain.

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}$$

If we put *z* = *jω* and *s* = *j*Ω, we get

$$\Omega = \tan\left(\frac{\omega}{2}\right)$$

Applying the bilinear transformation to the frequencies at the band edges, we get:

ω	Ω
0.4111π	0.7536
0.5778π	1.2799
0.40π	0.7265
0.5889π	1.3270
0	0
π	∞

Therefore, the corresponding analog filter specifications for the same type of analog filter using the bilinear transformation are:

- Passband: 0.7536(Ω_{P1}) to 1.2799(Ω_{P2})
- Stopband: 0 to 0.7265(Ω_{S1}) and 1.3270(Ω_{S2}) to ∞
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Monotonic
- Stopband Nature: Monotonic

4.Analog Frequency transformation and relevant parameters

We need to transform a Bandpass analog filter to a Lowpass analog filter. We require two parameters in such a case. We can make use of the Bandpass transformation which is given as:

$$\Omega_L = \frac{\Omega^2 - \Omega_0^2}{B\Omega}$$

The two parameters in the above equation are B and Ω_0. They can be determined using the specifications of bandpass analog filter using the following relations:

$$\Omega_0 = \sqrt{\Omega_{P1}\Omega_{P2}} = \sqrt{0.7536 * 1.2799} = 0.9821$$

$$B = \Omega_{P1} - \Omega_{P2} = 1.2799 - 0.7536 = 0.5264$$

Ω	Ω_L
0^+	$-\infty$
0.7265	-1.1417
0.7536	-1.00
0.9821	0
1.2799	1.00
1.2892	1.1403
∞	∞

5.Frequency transformed lowpass analog filter specifications

- Passband edge: 1 (Ω_{LP})
- Stopband edge: min ($-\Omega_{LS}$, Ω_{LS}) = 1.1403 (Ω_{LS})
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Monotonic
- Stopband Nature: Monotonic

6.Analog Lowpass Transfer function

We need an Analog Filter which has a monotonic passband and a monotonic stopband. Therefore, we need to design using the Butterworth approximation. Since the tolerance in both passband and stopband is 0.15, we define two new quantities in the following way:

$$D_1 = \frac{1}{(1-\delta)^2} - 1 = 0.384$$

$$D2 = \frac{1}{\delta^2} - 1 = 43.44$$

Now using the inequality on the order N of the filter for the Butterworth Approximation we get:

$$N_{min} = \left\lceil \frac{log\sqrt{\frac{D_2}{D_1}}}{\log\frac{\Omega_{Ls}}{\Omega_{LP}}} \right\rceil$$

$$N_{min} = 19$$

The cut off frequency (Ω_c) of the Analog LPF should satisfy the following constraint:

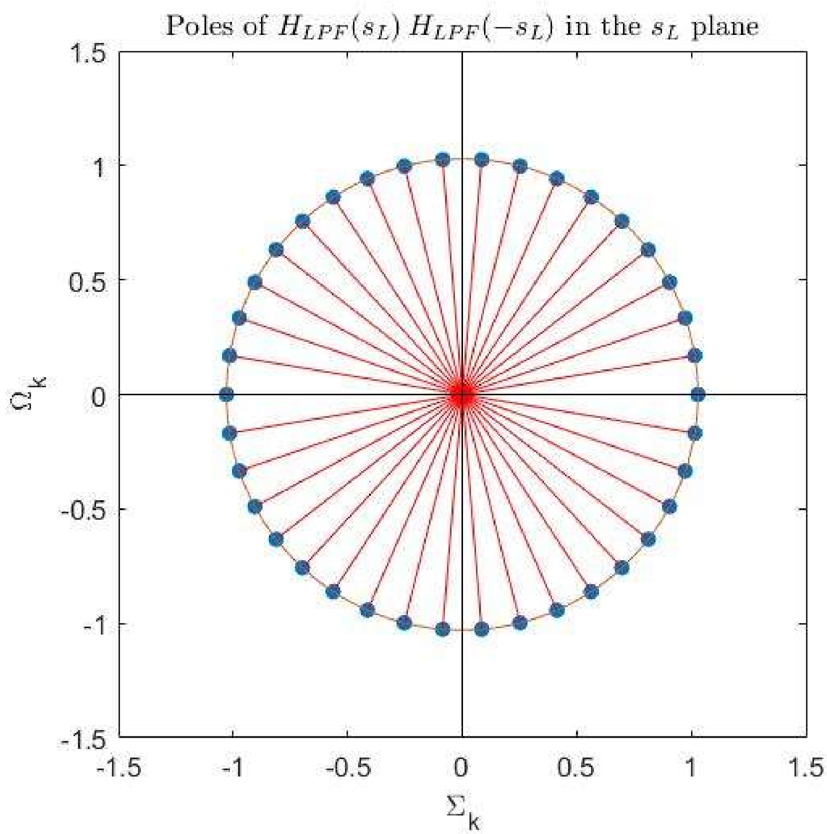
$$\frac{\Omega_{LP}}{D_1^{\frac{1}{2N}}} \leq \Omega_c \leq \frac{\Omega_{Ls}}{D_2^{\frac{1}{2N}}}$$

$$1.0255 \leq \Omega_c \leq 1.0326$$

Thus, we can choose the value of Ω_c as 1.0290. Now, the poles of the transfer function can be obtained by solving the equation:

$$1 + \left(\frac{sL}{j\Omega_c}\right)^{2N} = 1 + \left(\frac{sL}{j1.0290}\right)^{38} = 0$$

Solving for the roots (using Wolfram) we get:



Poles of the equivalent lowpass filter

Note that the above figure shows the poles of the Magnitude Plot of the Transfer Function. To get a stable Analog LPF, we must include the poles lying in the Left Half Plane in the Transfer Function.

```
p1 = -0.2526 + 0.9975i;
p2 = -0.4133 + 0.9423i
p3 = -0.5628 + 0.8614i;
p4 = -0.6969 + 0.7571i;
p5 = -0.8120 + 0.6320i;
p6 = -0.9050 + 0.4897i;
p7 = -0.9732 + 0.3341i;
p8 = -1.0150 + 0.1694i;
p9 = -1.0290 + 0.0000i;
p10 = -1.0150 - 0.1694i;
p11 = -0.9732 - 0.3341i;
p12 = -0.9050 - 0.4897i;
p13 = -0.8120 - 0.6320i;
p14 = -0.6969 - 0.7571i;
p15 = -0.5628 - 0.8614i;
p16 = -0.4133 - 0.9423i;
p17 = -0.2526 - 0.9975i;
p18 = -0.0850 - 1.0255i;
p19 = -0.0850 + 1.0255i;
```

Using the above poles which are in the left half plane we can write the Analog Lowpass Transfer Function as:

$$H_{analog,LPF} = \frac{\Omega_c^N}{\prod_{k=0}^{19}(s_L - pk)} = \frac{1.7224}{\sum_{k=0}^{19} a_k S_L^k}$$

H_analog_lp =

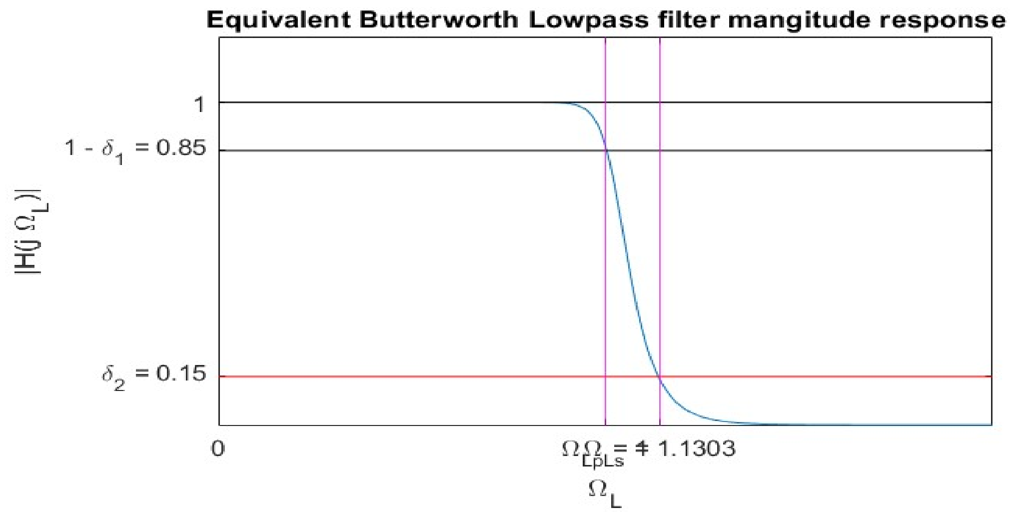
1.7224

(s+1.029) (s^2 + 2.03s + 1.059) (s^2 + 1.947s + 1.059) (s^2 + 1.81s + 1.059) (s^2 + 1.624s + 1.059)

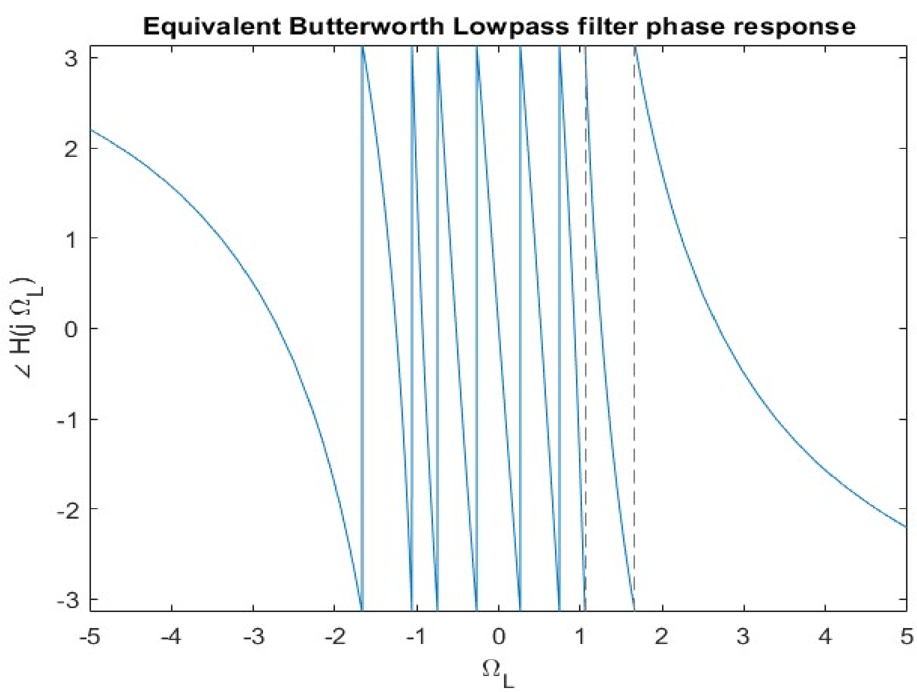
(s^2 + 1.394s + 1.059) (s^2 + 1.126s + 1.059) (s^2 + 0.8267s + 1.059) (s^2 + 0.5052s + 1.059)

(s^2 + 0.17s + 1.059)

The magnitude and the phase responses of this equivalent Butterworth LPF are as follows:



Magnitude Response



Phase Response

7.Analog Bandpass Transfer function

The transformation equation is given by:

$$s_L = \frac{s^2 + \Omega_0^2}{Bs}$$

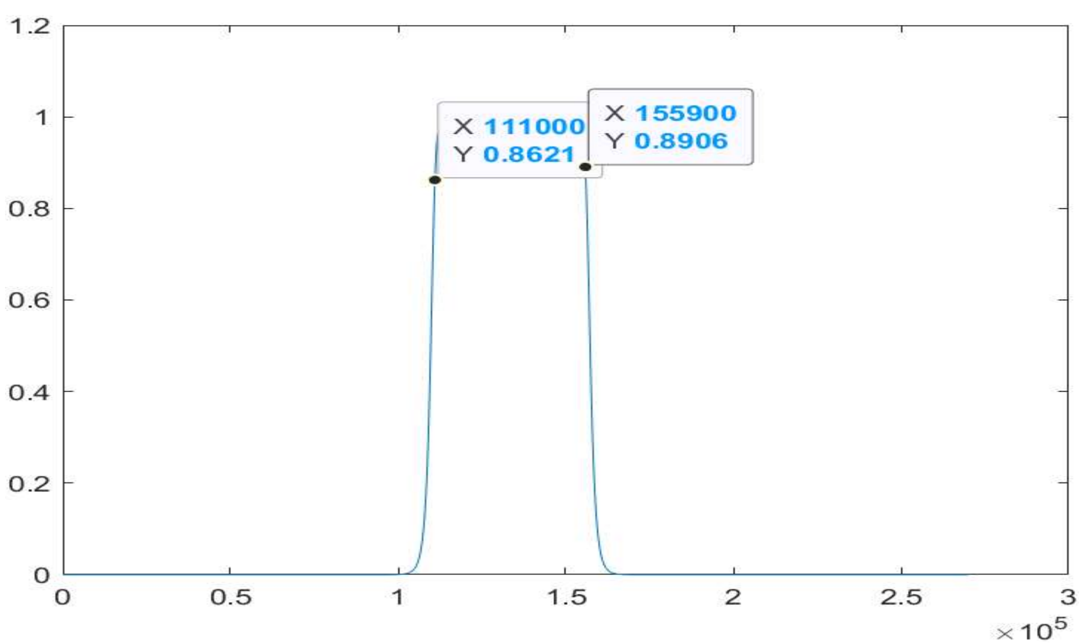
Substituting the values of B (0.5264) and Ω_0 (0.9821), we get:

$$s_L = \frac{s^2 + 0.9645}{0.5264s}$$

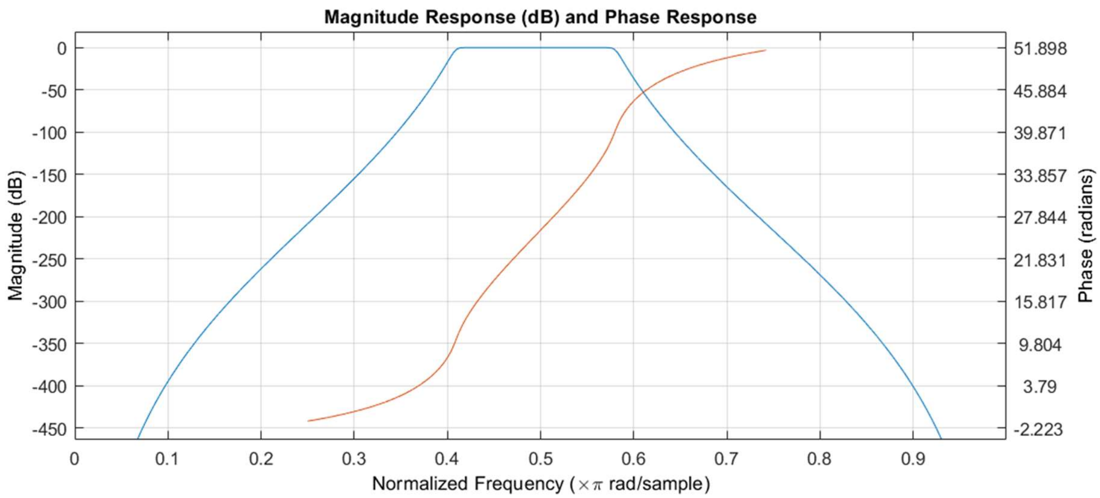
Substituting this value in $H_{analog,LPF}(s_L)$ we get $H_{analog,BSF}(s)$.

Hbp_analog =

$$\frac{1.0\, s^{19}}{\begin{array}{l} 1.0\, s^{38} + 6.8007597860152204496473065123028\, s^{37} + 42.824386685922617515943347995916\, s^{36} + 179.10101534093999373693166113418\, s^{35} \\ +678.90611218195705745494225433062\, s^{34} + 2099.8484101964450978106339232108\, s^{33} + 5946.4417754182684131794774297243\, s^{32} \\ +14708.401534050401910807238477768\, s^{31} + 33651.581997205676707733071497709\, s^{30} + 69394.048050053675792956750 2589\, s^{29} \\ +133396.86698367230182464527640851\, s^{28} + 235099.16259630056826158456873707\, s^{27} + 388406.44514261909215954873 73865\, s^{26} \\ +594264.52838443816946795404532446\, s^{25} + 855628.10324045891980397371804947\, s^{24} + 1147898.3685785020272825457473372\, s^{23} \\ +1452896.4316027302310039372406266\, s^{22} + 1719558.6368013469747605166061876\, s^{21} + 1922804.9867500225239599049991659\, s^{20} \\ +2013789.7748238216612905515919201\, s^{19} + 1993566.2193286656493677439709282\, s^{18} + 1848450.5732009880623761447420973\, s^{17} \\ +1619276.2112287595318361851471496\, s^{16} + 1326432.4667330658640156574218788\, s^{15} + 1025090.4123599670761882062074347\, s^{14} \\ +738163.15733081136784906798763032\, s^{13} + 500212.35142143183434546311385801\, s^{12} + 313916.75169495544486832689600046\, s^{11} \\ +184673.45121742606029048683055364\, s^{10} + 99603.932304633732726727053736787\, s^{9} + 50078.945315747230072108709116321\, s^{8} \\ +22693.976030124156834552737760228\, s^{7} + 9512.5666223020604754409817473725\, s^{6} + 3482.7629084139444315904878 61028\, s^{5} \\ +1167.4575862680200800156871407394\, s^{4} + 319.31907833655174019084678242276\, s^{3} + 79.161387419735933966043542184654\, s^{2} \\ +13.033921669974769338725074861 59\, s + 1.9870696850476324947487035376206 \end{array}}$$



Magnitude response of the BPF



Magnitude and Phase response of the BPF

8.Discrete Time Filter Transfer Function

To transform the analog domain transfer function into the discrete domain, we need to make use of the Bilinear Transformation which is given as:

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}$$

Using this equation, we get $H_{discrete,BSF}(z)$ from $H_{analog,BSF}(s)$.

The discrete time filter transfer function is

$$H_{bpf}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) = H(z) = \frac{\sum_{k=0}^{38} b_k z^k}{\sum_{k=0}^{38} a_k z^k}$$

with coefficients

Coefficient	Value	Coefficient	Value	Coefficient	Value	Coefficient	Value
		a19	- 928.3716			b19	0
a38	0.0013	a18	3132.8126	b38	-1	b18	-92378
a37	- 0.0010	a17	- 1182.545	b37	0	b17	0
a36	0.0343	a16	3634.0974	b36	19	b16	75582
a35	- 0.0260	a15	-1225.093	b35	0	b15	0
a34	0.4214	a14	3454.649	b34	-171	b14	-50388
a33	- 0.3002	a13	- 1024.93	b33	0	b13	0
a32	3.2673	a12	2668.704	b32	969	b12	27132
a31	- 2.1834	a11	- 683.351	b31	0	b11	0
a30	17.9454	a10	1651.509	b30	-3876	b10	-11628
a29	- 11.2154	a9	- 355.25	b29	0	b9	0
a28	74.2066	a8	800.43	b28	11628	b8	3876
a27	- 43.2240	a7	- 139.030	b27	0	b7	0

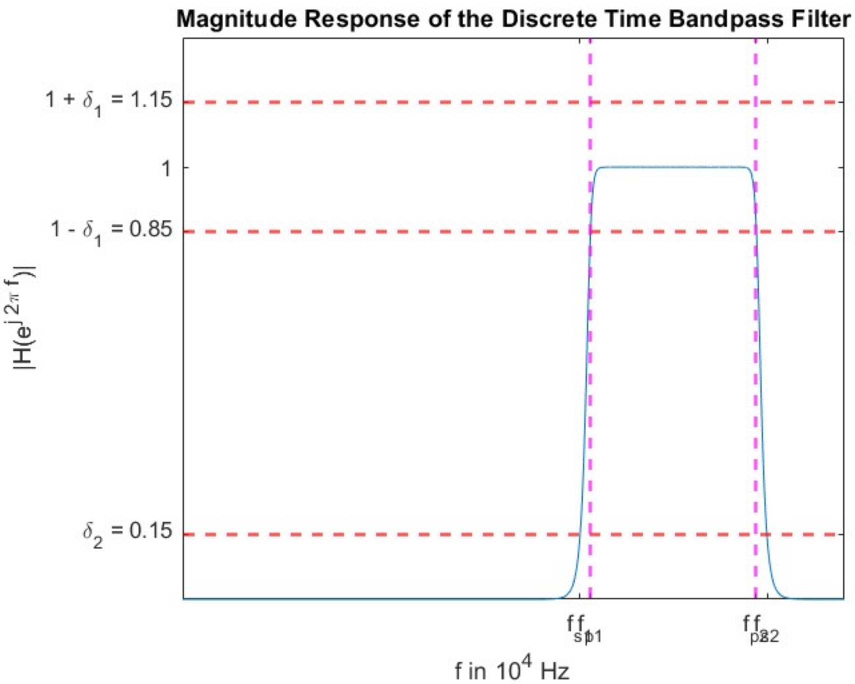
a26	239.6353	*a6*	293.12	*b26*	-27132	*b6*	-969
a25	- 129.5592	*a5*	- 38.5906	*b25*	0	*b5*	0
a24	618.5768	*a4*	76.371	*b24*	50388	*b4*	171
a23	- 308.8901	*a3*	- 6.784	*b23*	0	*b3*	0
a22	1295.55	*a2*	12.638	*b22*	-75582	*b2*	-19
a21	- 593.9782	*a1*	- 0.568	*b21*	0	*b1*	0
a20	2221.5952	*a0*	*1*	*b20*	92378	*b0*	1

Denominator(ak) Numerator(bk)

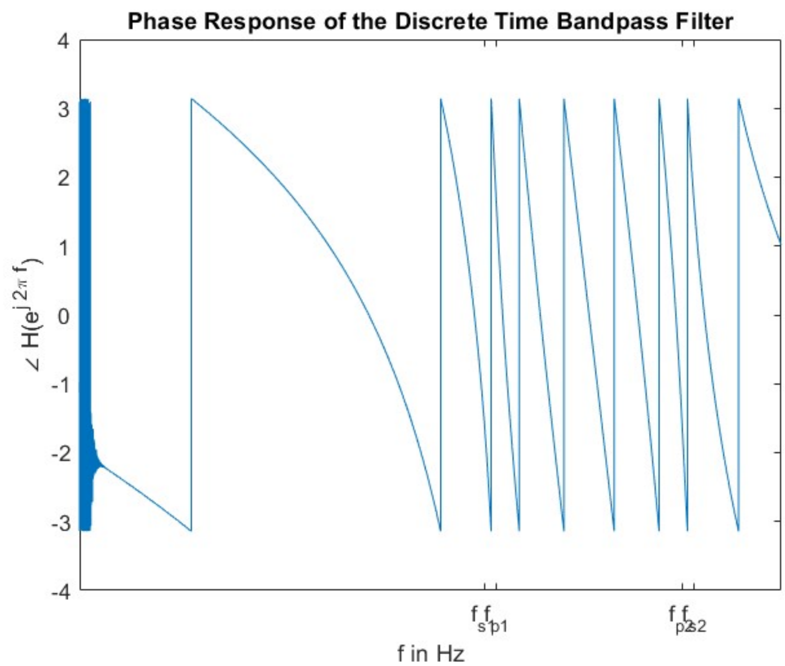
Hbp_discrete =

-(z^{38}- 19*z^{36} + 171.0*z^{34} - 969.0*z^{32}+ 3876.0*z^{30} - 11628.0*z^{28} + 27132.0*z^{26} - 50388.0*z^{24} + 75582.0*z^{22} - 92378.0*z^{20} + 92378.0*z^{18} - 75582.0*z^{16} + 50388.0*z^{14}- 27132.0*z^{12} + 11628.0*z^{10} - 3876.0*z^{8} + 969.0*z^{6} - 171.0*z^{4} + 19.0*z^{2} - 1.0) /

(0.0013*z^{38}- 0.0010*z^{37} + 0.0343*z^{36} - 0.0260*z^{35} + 0.4214*z^{34} - 0.3002*z^{33} + 3.2673*z^{32} - 2.1834*z^{31} + 17.9454*z^{30} - 11.2154*z^{29} + 74.2066*z^{28} - 43.2240*z^{27} + 239.6353*z^{26} - 129.5592*z^{25} + 618.5768*z^{24} - 308.8901*z^{23}+ 1295.55*z^{22} - 593.9782*z^{21} + 2221.5952*z^{20} - 928.3716*z^{19} + 3132.8126*z^{18} - 1182.545*z^{17} + 3634.0974*z^{16} - 1225.093*z^{15} + 3454.649*z^{14} - 1024.93*z^{13} + 2668.704*z^{12} - 683.351*z^{11} + 1651.509*z^{10} - 355.25*z^{9} + 800.43*z^{8} - 139.030*z^{7} + 293.12*z^{6} - 38.5906*z^{5} + 76.371*z^{4} - 6.784*z^{3} + 12.638*z^{2} - 0.568*z^{1} + 1.0)



Magnitude Response of Discrete Time Bandpass Filter



Phase Response of Discrete Time Bandpass Filter

9.FIR Filter Transfer Function using Kaiser Window

9.1. Kaiser Window Parameters

The tolerance in both the stopband and passband is given to be 0.15. Therefore δ = 0.15 and we get the minimum stopband attenuation to be:

$$A = -20\log(0.15) = 16.4782dB$$

Since A < 21, we get β to be 0 where β is the shape parameter of Kaiser window. Now to estimate the window length required, we use the empirical formula for the lower bound on the window length.

$$M \geq \frac{A - 7.95}{2.285 * \Delta\omega_T}$$

Here $\Delta\omega_T$ is the minimum transition bandwidth. In our case, the transition width is the same on either side of the passband.

$$\Delta\omega_T = 0.0111\pi$$

This gives us M = 108. On successive trials in MATLAB, it was found that a window length of 124 is required to satisfy the required constraints. Also, since β is 0, the window is a rectangular window.

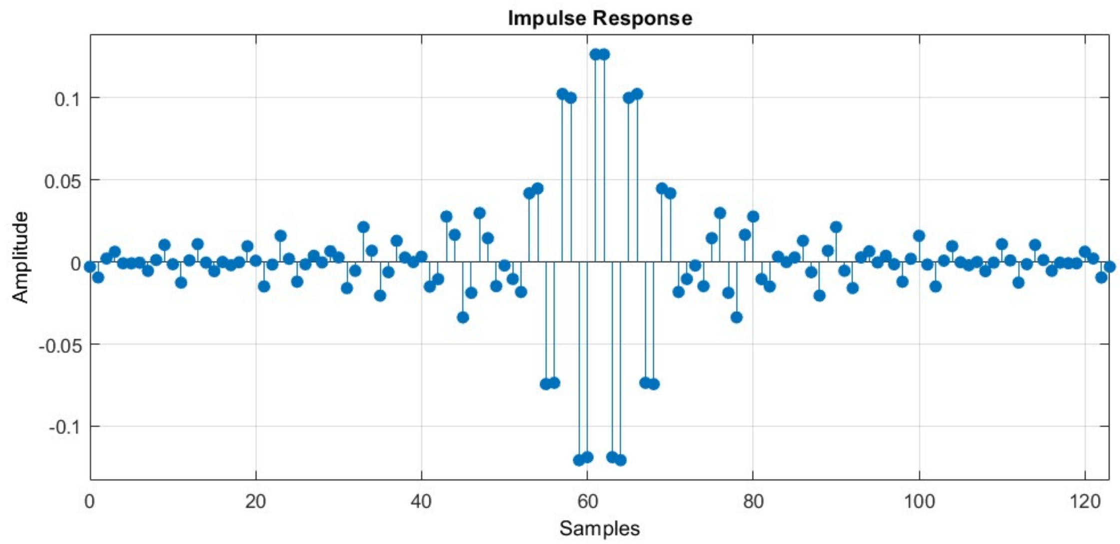
9.2. Discrete Time FIR Filter Transfer Function

The time domain coefficients were obtained by first generating the ideal impulse response samples for the same length as that of the window. The Kaiser Window was generated using the MATLAB function and applied on the ideal impulse response samples. For generating the ideal impulse response, a separate function was made to generate the impulse response of Low-Pass filter. It took the cut off value and the number of samples as input argument. The band-pass impulse response samples were generated as the difference between two low-pass filters as done in class.

```
Columns 1 through 10

   -0.0029   -0.0094    0.0021    0.0062   -0.0008   -0.0008   -0.0004   -0.0054
0.0012    0.0104

  Columns 11 through 20

   -0.0013   -0.0125    0.0009    0.0109   -0.0004   -0.0056    0.0000   -0.0020
-0.0002    0.0096

  Columns 21 through 30

    0.0008   -0.0149   -0.0016    0.0159    0.0019   -0.0120   -0.0014    0.0038
-0.0003    0.0065

  Columns 31 through 40

    0.0028   -0.0159   -0.0054    0.0213    0.0069   -0.0205   -0.0062    0.0129
0.0027   -0.0000

  Columns 41 through 50

    0.0032   -0.0150   -0.0104    0.0277    0.0166   -0.0337   -0.0188    0.0298
0.0145   -0.0147


  Columns 51 through 60

   -0.0021   -0.0104   -0.0183    0.0418    0.0448   -0.0744   -0.0736    0.1023
0.0999   -0.1207

  Columns 61 through 70

   -0.1188    0.1264    0.1264   -0.1188   -0.1207    0.0999    0.1023   -0.0736
-0.0744    0.0448

  Columns 71 through 80

    0.0418   -0.0183   -0.0104   -0.0021   -0.0147    0.0145    0.0298   -0.0188
-0.0337    0.0166

  Columns 81 through 90

    0.0277   -0.0104   -0.0150    0.0032   -0.0000    0.0027    0.0129   -0.0062
-0.0205    0.0069

  Columns 91 through 100

    0.0213   -0.0054   -0.0159    0.0028    0.0065   -0.0003    0.0038   -0.0014
-0.0120    0.0019

  Columns 101 through 110

    0.0159   -0.0016   -0.0149    0.0008    0.0096   -0.0002   -0.0020    0.0000
-0.0056   -0.0004

  Columns 111 through 120

    0.0109    0.0009   -0.0125   -0.0013    0.0104    0.0012   -0.0054   -0.0004
-0.0008   -0.0008

  Columns 121 through 124

    0.0062    0.0021   -0.0094   -0.0029
```

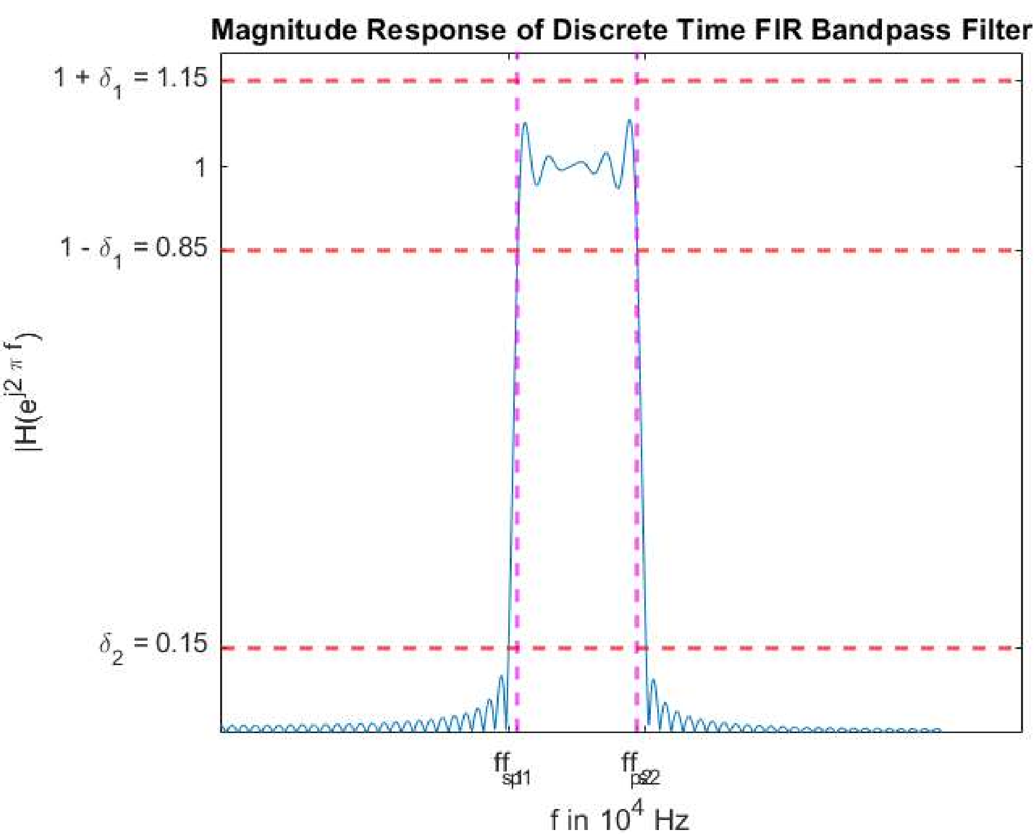
Bandpass FIR Transfer Function

The z-transform can be formed from the coefficients of this finite sequence.

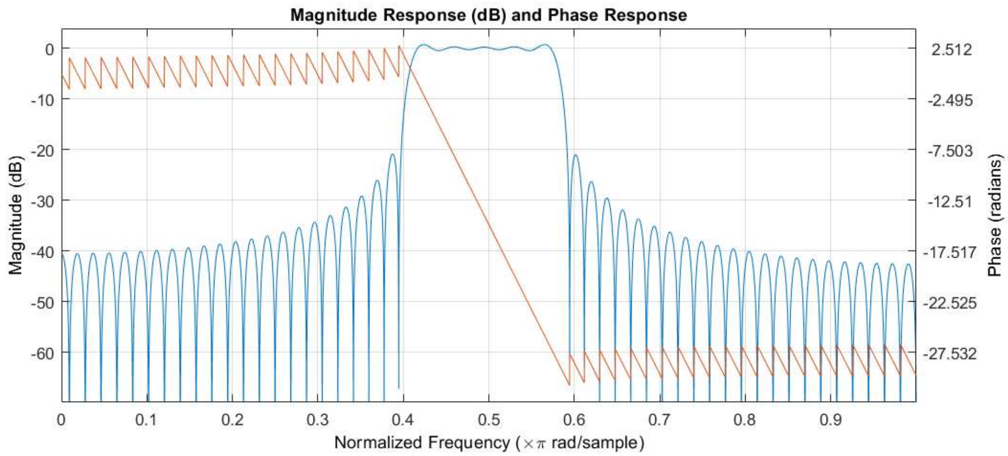


Bandpass FIR Impulse Response

9.3. Magnitude and Phase Response



Magnitude Response of the FIR filter



Phase Response of the FIR filter

9.4. Comparison with their Filter

we can see that the FIR filter gives a linear phase response for the frequencies in the passband range i.e from 111kHz to 156kHz.

Also, we can observe that the order of the IIR filter is 19 and the order of the FIR filter is 124 which is much higher. This once again proves the fact that *there is no free lunch*, i.e., we are getting a linear phase response at the expense of resources.

Filter-2(Bandstop) Details

1.Un-normalized Discrete Time Filter Specifications

Filter number: 27

Since filter number < 80, m = 27.

q(m) = greatest integer less than 0.1m = floor(0.1m) = 2

r(m) = m – 10q(m) = 7

BL(m) = 5 + 3 q(m) + 11 r(m) = 88

BH(m) = BL(m) + 25 = 113

The second filter is given to be a Band-Stop filter with stopband from BL(m) kHz to BH(m) kHz. Therefore, the specifications are:

- Stopband: 88 kHz to 113 kHz
- Transition band: 3 kHz on either side of passband
- Passband: 0 to 85 kHz and 116 kHz to 200 kHz (Sampling rate is 400 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple
- Stopband Nature: Monotonic

2.Normalized Digital Filter Specifications

Sampling rate: 400 kHz

In the normalized frequency axis, sampling rate corresponds to 2π.

This any frequency (Ω), up to 200 kHz can be represented on the normalized axis ω as:

$$\omega = \frac{\Omega * 2\pi}{\Omega_s(Sampling\ rate)}$$

Therefore, the corresponding normalized discrete filter specifications are:

- Stopband: 0.44π to 0.565π
- Transition band: 0.015π on either side of passband
- Passband: 0 to 0.425π and 0.58π to π (Sampling rate is 400 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple
- Stopband Nature: Monotonic

3.Analog filter specifications for Band-pass filter using Bilinear transformation

We use the following transformation to move from the *z* domain to the *s* domain.

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}$$

If we put $z = j\omega$ and $s = j\Omega$, we get

$$\Omega = \tan\left(\frac{\omega}{2}\right)$$

Applying the bilinear transformation to the frequencies at the band edges, we get:

ω	Ω
0.44π	0.8273
0.565π	1.2283
0.425π	0.7883
0.58π	1.2892
0	0
π	∞

Therefore, the corresponding analog filter specifications for the same type of analog filter using the bilinear transformation are:

- Stopband: 0.8273(Ω_{S1}) to 1.2283(Ω_{S2})
- Passband: 0 to 0.7883(Ω_{P1}) and 1.2892(Ω_{P2}) to ∞
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple
- Stopband Nature: Monotonic

4.Frequency Transformation & Relevant Parameters

We need to transform a Band-Stop analog filter to a Lowpass analog filter. We require two parameters in such a case. We can make use of the Bandstop transformation which is given as:

$$\Omega_L = \frac{B\Omega}{\Omega_0^2 - \Omega^2}$$

The two parameters in the above equation are B and Ω_0. They can be determined using the specifications of the bandpass analog filter using the following relations:

$$\Omega_0 = \sqrt{\Omega_{P1}\Omega_{P2}} = \sqrt{0.7883 * 1.2892} = 1.0081$$

$$B = \Omega_{P2} - \Omega_{P1} = 1.2892 - 0.7883 = 0.5009$$

Ω	Ω_L
0^+	0^+
0.7883	+1
0.8273	+1.2488
1.0081 (Ω_0^-)	∞
1.0081 (Ω_0^+)	$-\infty$
1.2283	-1.2494
1.2892	-1
∞	0^-

5.Frequency Transformed Lowpass Analog Filter Specifications

- Passband edge: 1 (Ω_{LP})

- Stopband edge: min (Ω_{LS} , $-\Omega_{LS}$) = 1.2488 (Ω_{LS})
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple
- Stopband Nature: Monotonic

6.Analog Lowpass Transfer Function

We need an Analog Filter which has an equiripple passband and a monotonic stopband. Therefore, we need to design using the Chebyshev approximation. Since the tolerance (δ) in both passband and stopband is 0.15, we define two new quantities in the following way:

$$D_1 = \frac{1}{(1-\delta)^2} - 1 = 0.384$$

$$D2 = \frac{1}{\delta^2} - 1 = 43.44$$

Now choosing the parameter ϵ of the Chebyshev filter to be $\sqrt{D_1}$, we get the minimum value of N as:

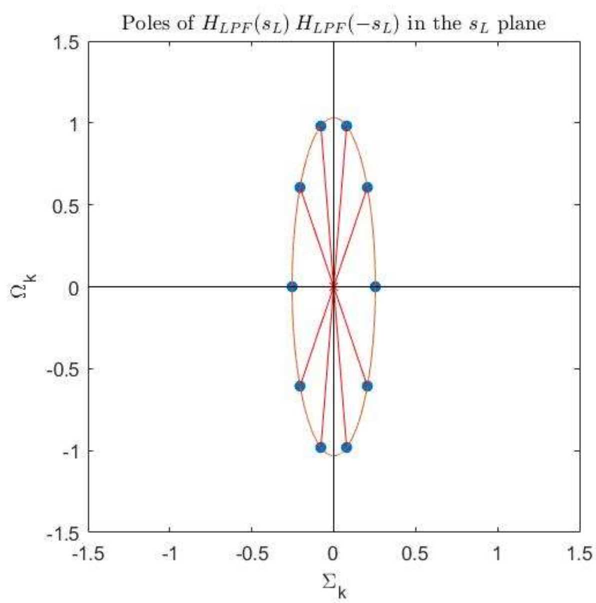
$$N_{min} = \left\lceil \frac{\cosh^{-1}\left(\frac{\sqrt{D_1}}{\sqrt{D_2}}\right)}{\cosh^{-1}(\frac{\Omega_{LS}}{\Omega_{LP}})} \right\rceil$$

$$N_{min} = 5$$

Now, the poles of the transfer function can be obtained by solving the equation:

$$1 + D_1 \cosh^2(N_{min} \cosh^{-1}(\frac{s}{j})) = 1 + 0.384 \cosh^2(5 \cosh^{-1}(\frac{s}{j})) = 0$$

Solving for the roots (using Wolfram) we get:



Poles of the equivalent lowpass filter

Note that the above figure shows the poles of the Magnitude Plot of the Transfer Function. To get a stable Analog LPF, we must include the poles lying in the Left Half Plane in the Transfer Function.

```
p1 = -0.0785 + 0.9812i;
p2 = -0.2054 + 0.6064i;
p3 = -0.2539 + 0.0000i;
p4 = -0.2054 - 0.6064i;

p5 = -0.0785 - 0.9812i;
```

Using the above poles which are in the left half plane and the fact that N is odd we can write the Analog Lowpass Transfer Function as:

$$H_{analog,LPF}(s_L) = \frac{(-1)^4 p_1 p_2 p_3 p_4 p_5}{\sqrt{1+D_1}(s_L - p_1)(s_L - p_2)(s_L - p_3)(s_L - p_4)(s_L - p_5)}$$

Note that since it is even order we take the DC Gain to be $\frac{1}{\sqrt{1+\epsilon^2}}$

H_analog_lp =

0.10085

(s+0.2539) (s^2 + 0.4108s + 0.4099) (s^2 + 0.1569s + 0.969)

7.Analog Bandstop Transfer Function

The transformation equation is given by:

$$s_L = \frac{Bs}{\Omega_0^2 + s^2}$$

Substituting the values of the parameters B(0.5009) and $\Omega_0(1.0081)$, we get,

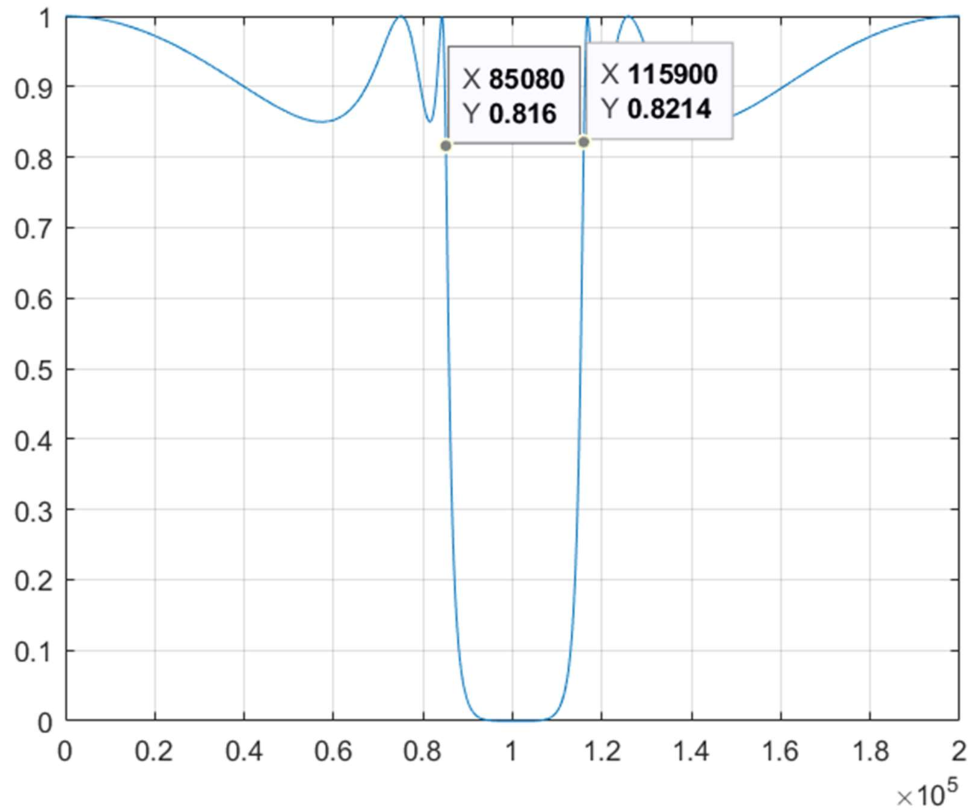
$$s_L = \frac{0.5009s}{1.0081^2 + s^2}$$

Substituting this value in $H_{analog,LPF}(s_L)$ we get $H_{analog,BSF}(s)$.

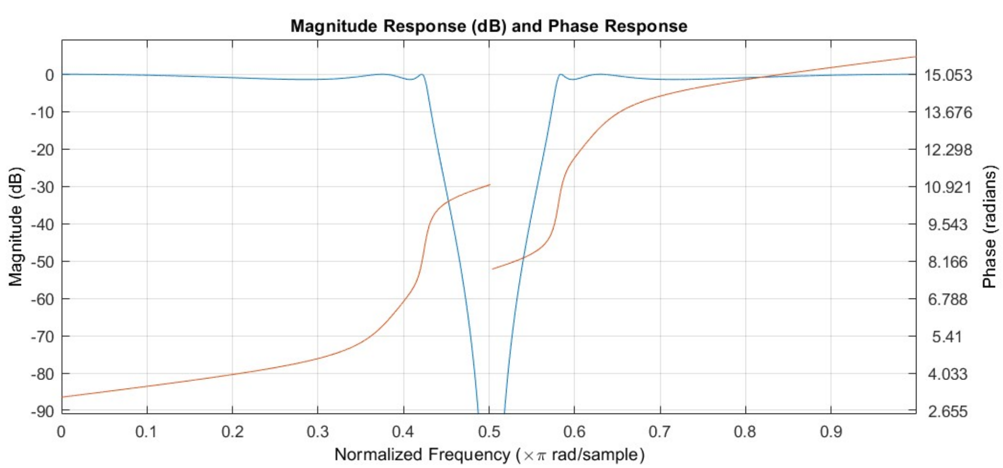
Hstp_analog =

(1.0*s^10 + 5.08*s^8 + 10.32*s^6 + 10.49*s^4 + 5.33*s^2 + 1.08) /

(1.0*s^10 + 2.55*s^9 + 7.14*s^8 + 12.36*s^7 + 17.12*s^6 + 20.17*s^5 + 17.40*s^4 + 12.77*s^3 + 7.49*s^2 + 2.72*s + 1.084)



Magnitude Response of Chebyshev Bandstop filter



Magnitude and Phase Response of Chebyshev Bandstop filter

8.Discrete Time Filter Transfer Function

To transform the analog domain transfer function into the discrete domain, we need to make use of the Bilinear Transformation which is given as:

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}$$

Using this equation, we get $H_{discrete,BSF}(z)$ from $H_{analog,BSF}(s)$.

Hstp_discrete =

(1.0*z^{-10} + 0.08*z^{-9} + 5.002*z^{-8} + 0.32*z^{-7} + 10.007*z^{-6} + 0.48*z^{-5} + 10.007*z^{-4} + 0.32*z^{-3} + 5.002*z^{-2} + 0.08*z^{-1} + 1.0)/

(0.006*z^{-10} + 0.005*z^{-9} + 0.62*z^{-8} + 0.05*z^{-7} + 2.26*z^{-6} + 0.14*z^{-5}+ 3.66*z^{-4} + 0.15*z^{-3}+ 2.92*z^{-2} + 0.06*z^{-1} + 1.0)

Coefficient	Value	Coefficient	Value
$a10$	1	b10	0.006
$a9$	0.08	$b9$	0.005
$a8$	5.002	$b8$	0.62
$a7$	0.32	$b7$	0.05
$a6$	10.007	$b6$	2.26
$a5$	0.48	$b5$	0.14
$a4$	10.007	$b4$	3.66
$a3$	0.32	$b3$	0.15
$a2$	5.002	$b2$	2.92
$a1$	0.08	$b1$	-0.06
$a0$	*1*	$b0$	*1*

Denominator(ak) Numerator(bk)

9.FIR Filter Transfer Function using Kaiser Window

9.1. Kaiser Window Parameters

The tolerance in both the stopband and passband is given to be 0.15. Therefore δ = 0.15 and we get the minimum stopband attenuation to be:

$$A = -20\log(0.15) = 16.482dB$$

Since A < 21, we get β to be 0 where β is the shape parameter of Kaiser window. Now to estimate the window length required, we use the empirical formula for the lower bound on the window length.

$$M \geq \frac{A - 7.95}{2.285 * \Delta\omega_T}$$

Here $\Delta\omega_T$ is the minimum transition bandwidth. In our case, the transition width is the same on either side of the passband.

$$\Delta\omega_T = 0.015\pi$$

This gives us M = 80. The above equation gives a loose bound on the window length when the tolerance is not very stringent. On successive trials in MATLAB, it was found that a window length of 93 is required to satisfy the required constraints. Also, since β is 0, the window is a rectangular window.

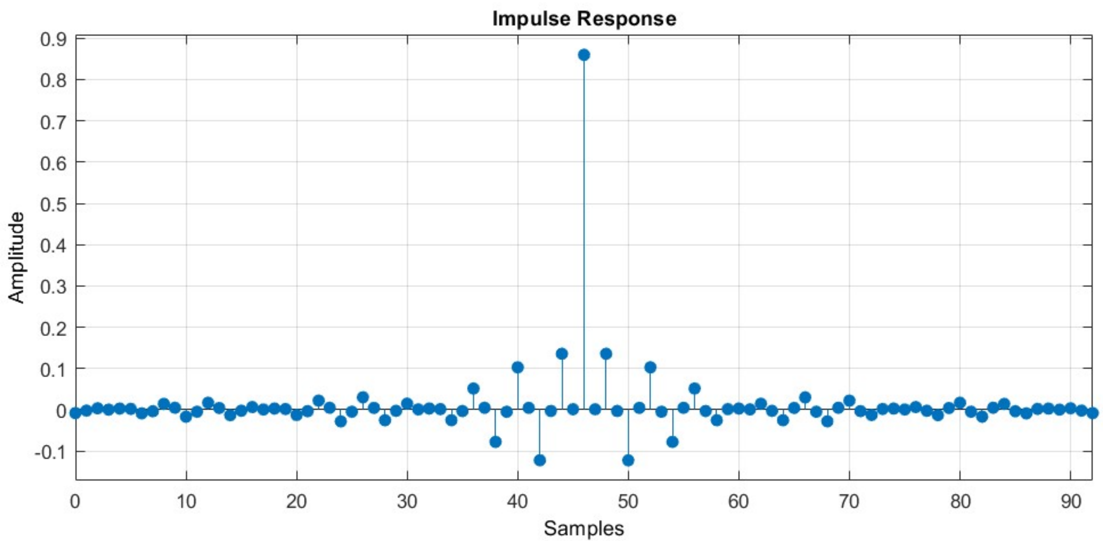
9.2. Discrete Time FIR Filter Transfer Function

The time domain coefficients were obtained by first generating the ideal impulse response samples for the same length as that of the window. The Kaiser Window was generated using the MATLAB function and applied on the ideal impulse response samples. For generating the ideal impulse response, a separate function was made to generate the impulse response of Low-Pass filter. It took the cut off value and the number of samples as input argument. The band-stop impulse response samples were generated as the difference between three low-pass filters (all-pass - bandpass) as done in class.

```
  Columns 1 through 10

   -0.0083   -0.0022    0.0034    0.0002    0.0027    0.0020   -0.0089   -0.0037
0.0140    0.0047

  Columns 11 through 20

   -0.0169   -0.0049    0.0168    0.0041   -0.0132   -0.0025    0.0064    0.0005
0.0028    0.0017

  Columns 21 through 30

   -0.0128   -0.0035    0.0220    0.0047   -0.0283   -0.0050    0.0299    0.0043
-0.0255   -0.0028

  Columns 31 through 40

    0.0145    0.0008    0.0028    0.0014   -0.0254   -0.0033    0.0513    0.0046
-0.0780   -0.0050



  Columns 41 through 50

    0.1027    0.0045   -0.1226   -0.0031    0.1355    0.0011    0.8600    0.0011
0.1355   -0.0031

  Columns 51 through 60

   -0.1226    0.0045    0.1027   -0.0050   -0.0780    0.0046    0.0513   -0.0033
-0.0254    0.0014

  Columns 61 through 70

    0.0028    0.0008    0.0145   -0.0028   -0.0255    0.0043    0.0299   -0.0050
-0.0283    0.0047

  Columns 71 through 80

    0.0220   -0.0035   -0.0128    0.0017    0.0028    0.0005    0.0064   -0.0025
-0.0132    0.0041

  Columns 81 through 90

    0.0168   -0.0049   -0.0169    0.0047    0.0140   -0.0037   -0.0089    0.0020
0.0027    0.0002
```

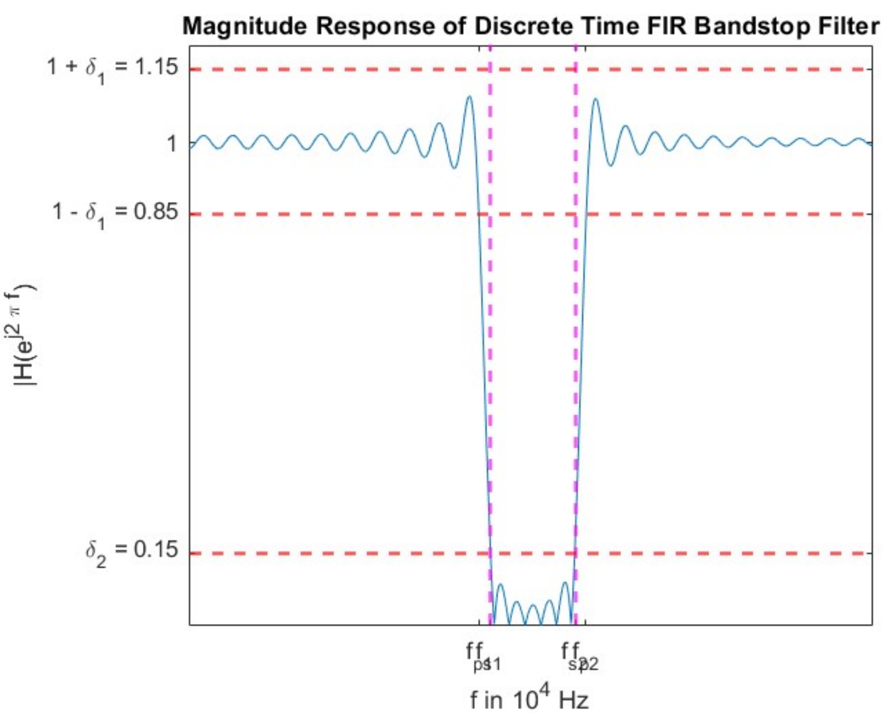
```
 Columns 91 through 93

   0.0034   -0.0022   -0.0083
```

Bandpass FIR Transfer Function

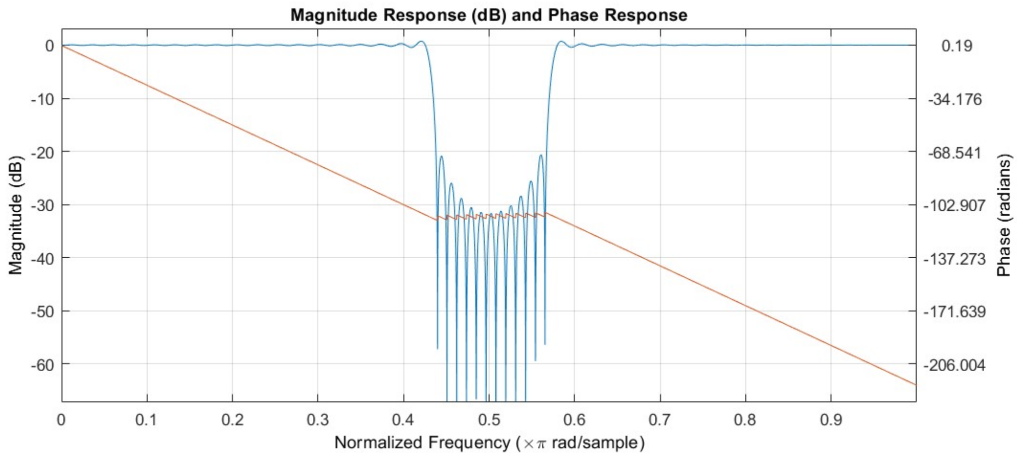


Bandstop FIR Impulse Response

9.3. Magnitude and Phase Response



Magnitude Response of the FIR filter



Phase Response of the FIR filter

9.4. Comparison with their Filter

we can see that the FIR filter gives a linear phase response for the frequencies in the passband range.

Also, we can observe that the order of the IIR filter is 5 and the order of the FIR filter is 93 which is much higher. This once again proves the fact that *there is no free lunch*, i.e., we are getting a linear phase response at the expense of resources.

Filter-3 (Elliptical Bandpass) details

1.Unnormalized Discrete time specifications

Filter number: 27

Since filter number is < 80, m = 27.

q(m) = greatest integer less than 0.1m = floor(0.1m) = 2

r(m) = m – 10q(m) = 7

BL(m) = 10 + 5 q(m) + 13 r(m) = 111

BH(m) = BL(m) + 45 = 156

The first filter is given to be a Band-Pass filter with passband from BL(m) to BH(m) kHz. Therefore, the specifications are:

- Passband: 111 kHz to 156 kHz
- Transition band: 3 kHz on either side of passband
- Stopband: 0 to 108 kHz and 159 kHz to 270 kHz (Sampling rate is 540 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: equiripple

- Stopband Nature: equiripple

2.Normalized Digital Filter Specifications

Sampling rate: 540 kHz

In the normalized frequency axis, sampling rate corresponds to 2π. Thus, any frequency (Ω), up to 270 kHz can be represented on the normalized axis ω as:

$$\omega = \frac{\Omega * 2\pi}{\Omega_s(Sampling\ rate)}$$

Therefore, the corresponding normalized discrete filter specifications are:

- Passband: 0.4111π to 0.5778π
- Transition band: 0.0111π on either side of passband
- Stopband: 0 to 0.40π and 0.5889π to π (Sampling rate is 540 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: equiripple
- Stopband Nature: equiripple

3.Analog filter specifications for Band-pass filter using Bilinear transformation

We use the following transformation to move from the *z* domain to the *s* domain.

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}$$

If we put $z = j\omega$ and $s = j\Omega$, we get

$$\Omega = \tan\left(\frac{\omega}{2}\right)$$

Applying the bilinear transformation to the frequencies at the band edges, we get:

ω	Ω
0.4111π	0.7536
0.5778π	1.2799
0.40π	0.7265

0.5889π	1.3270
0	0
π	∞

Therefore, the corresponding analog filter specifications for the same type of analog filter using the bilinear transformation are:

- Passband: 0.7536(Ω_{P1}) to 1.2799(Ω_{P2})
- Stopband: 0 to 0.7265(Ω_{S1}) and 1.3270(Ω_{S2}) to ∞
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: equiripple
- Stopband Nature: equiripple

4.Analog Frequency transformation and relevant parameters

We need to transform a Bandpass analog filter to a Lowpass analog filter. We require two parameters in such a case. We can make use of the Bandpass transformation which is given as:

$$\Omega_L = \frac{\Omega^2 - \Omega_0^2}{B\Omega}$$

The two parameters in the above equation are B and Ω_0. They can be determined using the specifications of bandpass analog filter using the following relations:

$$\Omega_0 = \sqrt{\Omega_{P1}\Omega_{P2}} = \sqrt{0.7536 * 1.2799} = 0.9821$$

$$B = \Omega_{P1} - \Omega_{P2} = 1.2799 - 0.7536 = 0.5264$$

Ω	Ω_L
0^+	$-\infty$
0.7265	-1.1417
0.7536	-1.00
0.9821	0
1.2799	1.00
1.2892	1.1403

∞	∞

5.Frequency transformed lowpass analog filter specifications

- Passband edge: 1 (Ω_{LP})
- Stopband edge: min ($-\Omega_{LS1}, \Omega_{LS2}$) = 1.1403 ($\Omega_{LS}$)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: equiripple
- Stopband Nature: equiripple

6.Analog Lowpass Transfer function

The general transfer function for an **Elliptic** filter is given by

$$H_{LPF}(s_L)\, H_{LPF}(-s_L) = \frac{1}{1+\epsilon^2\, F_N^2\left(\frac{s_L}{j\Omega_{Lp}}\right)}$$

with

$$F_N(w) = \mathrm{cd}(N\, uK_1, k_1)$$

and

$$w = \mathrm{cd}(uK, k))$$

where cd(x, k) denotes the Jacobian elliptic function cd with modulus k and real quarter-period K.

6.1.Jacobian Elliptic Functions

The elliptic function w = sn(z, k) is defined as follows:

$$z = \int_0^{\phi} \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} = \int_0^{w} \frac{dt}{\sqrt{(1-t^2)\,(1-k^2\,t^2)}} \quad , w = \sin\left(\phi(z,k)\right)$$

There are three more relevant elliptic functions, cn, dn, and cd defined as:

$$w = \mathrm{cn}(z, k) = \cos(\phi(z, k))$$

$$w = dn(z,k) = \sqrt{1-k^2 sn^2(z,k)}$$

$$w = cd(z,k) = \frac{cn(z,k)}{dn(z,k)}$$

We define the *complete elliptic integral of first kind* $K(k)$ or simply K as the value of z when $\mathrm{sn}(z, k) = 1$ i.e. when $\phi = \pi/2$

$$K(k) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - k^2 \sin^2 \theta}}$$

Clearly $\mathrm{cd}(K, k) = 0$. Associated with an elliptic modulus k, there is a complementary modulus $k' = \sqrt{1 - k^2}$ and its associated complete elliptic integral $K(k')$ denoted by $K'(k)$ or simply K'

$$K'(k) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - k'^2 \sin^2 \theta}} = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - (1 - k^2) \sin^2 \theta}}$$

The significance of K and K' is that sn and cd are *doubly-periodic functions* in the z plane with a real period $4K$ and a complex period $2jK'$. Some of the useful properties are as follows:

$$\begin{aligned}
\mathrm{cd}(z + (2n-1)K, k) &= (-1)^n \,\mathrm{sn}(z, k) \\
\mathrm{cd}(z + 2nK, k) &= (-1)^n \,\mathrm{cd}(z, k) \\
\mathrm{cd}(z + j\,K', k) &= \frac{1}{k\,\mathrm{cd}(z, k)} \\
\mathrm{cd}(j\,z, k) &= \frac{1}{\mathrm{dn}(z, k')} \\
\mathrm{cd}(j\,K', k) &= \frac{1}{k}
\end{aligned}$$

$\mathrm{cd}(z, k)$ has zeroes at $(2m+1)K + j\,2nK'$ and poles at $(2m+1)K + j\,(2n+1)K'$ for any integers m and n in the z plane.

6.2. Elliptic Filter Parameters

The order of the filter N is determining by putting a constraint that the magnitude response at the passband edge is not less than $1-\delta 1$ and at the stopband edge not greater than $\delta 2$.

$$N = \left\lceil \frac{KK1'}{K'K1} \right\rceil$$

where K, K', $K1$, $K1'$ are defined for the complete elliptic integrals of modulus k and $k1$ given by

$$k = \frac{\Omega_{LP}}{\Omega_{LS}} = \frac{1}{1.1403} = 0.87696$$

$$k1 = \sqrt{\frac{D_1}{D_2}} = 0.0940$$

We then have

Parameter	Value
K	2.19213
K’	1.67557
K1	1.5743
K1’	3.7566
N	4

We now have to find the poles of the the transfer function. The poles of the transfer function are given by the zeros of the denominator and the zeros of the transfer function are given by the poles of $F_N\left(\frac{s_L}{j\Omega L_p}\right)$. The zeros and poles of the transfer function are given by

$$z_i = j\,\Omega_{Lp}\,(k\,\zeta_i)^{-1}\,,\ i = 1, 2, \ldots, \lfloor N/2 \rfloor$$

$$p_i = j\,\Omega_{Lp}\,\mathrm{cd}((u_i - jv_0)K, k)\ ,\ i = 1, 2, \ldots, \lfloor N/2 \rfloor$$
$$p_0 = j\,\Omega_{Lp}\,\mathrm{sn}(jv_0K, k)$$

where

$$\zeta_i = \mathrm{cd}(u_iK, k)$$

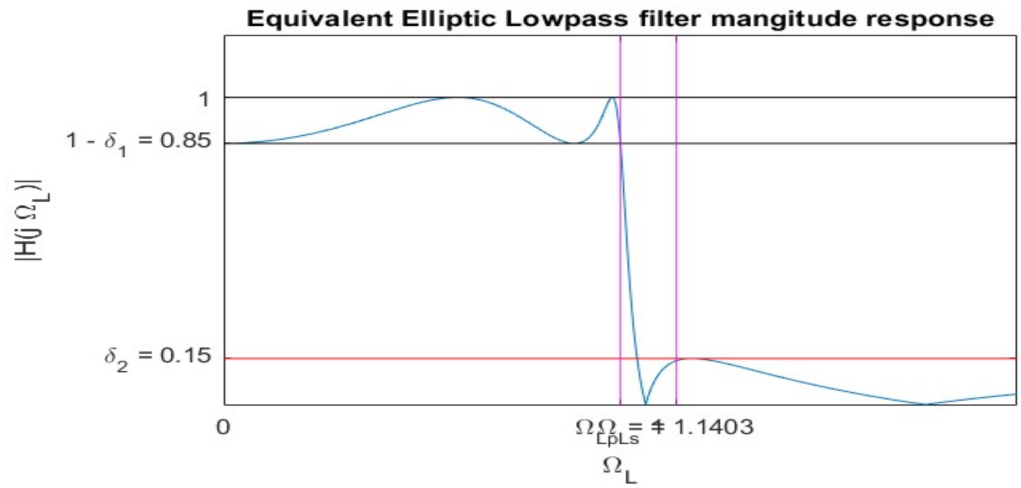
$$u_i = \frac{2i-1}{N},\ \ i = 1, 2, \ldots, \lfloor N/2 \rfloor$$
$$v_0 = -\frac{j}{NK1}\,Sn^{-1}\left(\frac{j}{e}, k1\right)$$

As N is even, the zeros and poles are

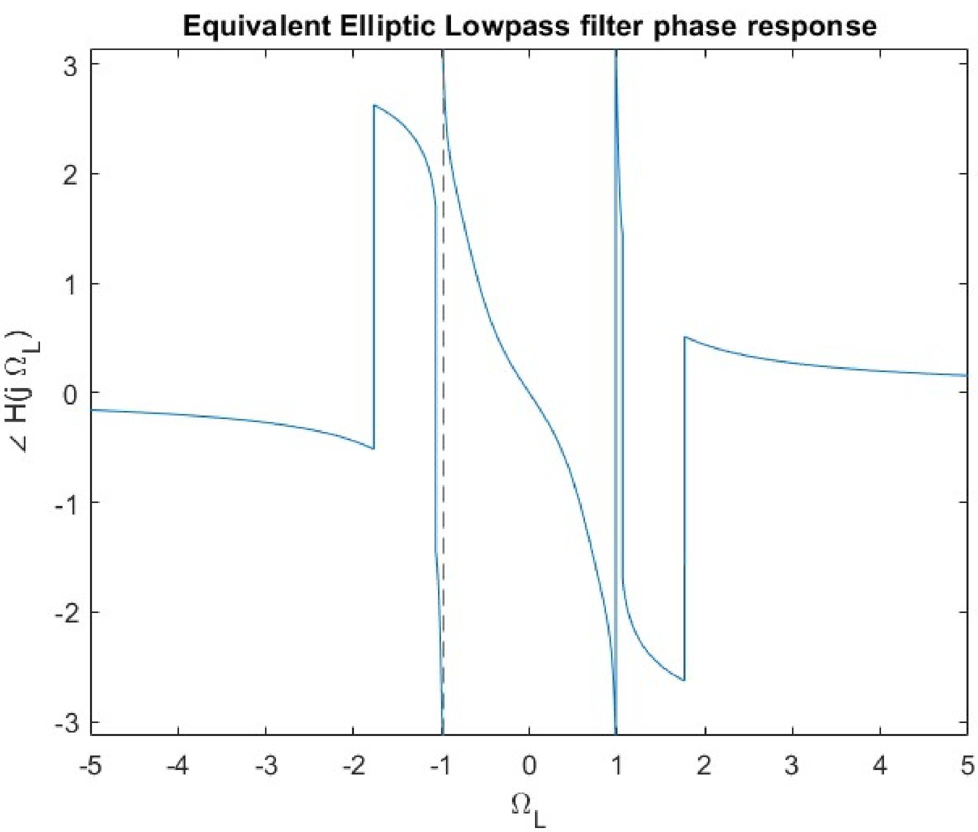
z1	-j1.06394
z2	j1.06394
z3	-j1.7692
z4	j1.7692
p1	-0.353652 - j0.707099
p2	-0.353652 +j0.707099
p3	-0.0309733 - j0.99954
p4	-0.0309733 +j0.99954

We then have

$$H_{LPf}(s_L) = \frac{0.8766s_L^4 + 3.7361s_L^2 + 3.1059}{5.8453s_L^4 + 4.4965s_L^3 + 9.7553s_L^2 + 4.3609s_L\ + 3.6538}$$



Magnitude Response of the equivalent Elliptic Lowpass Filter



Phase Response of the equivalent Elliptic Lowpass Filter

7.Analog Bandpass Transfer function

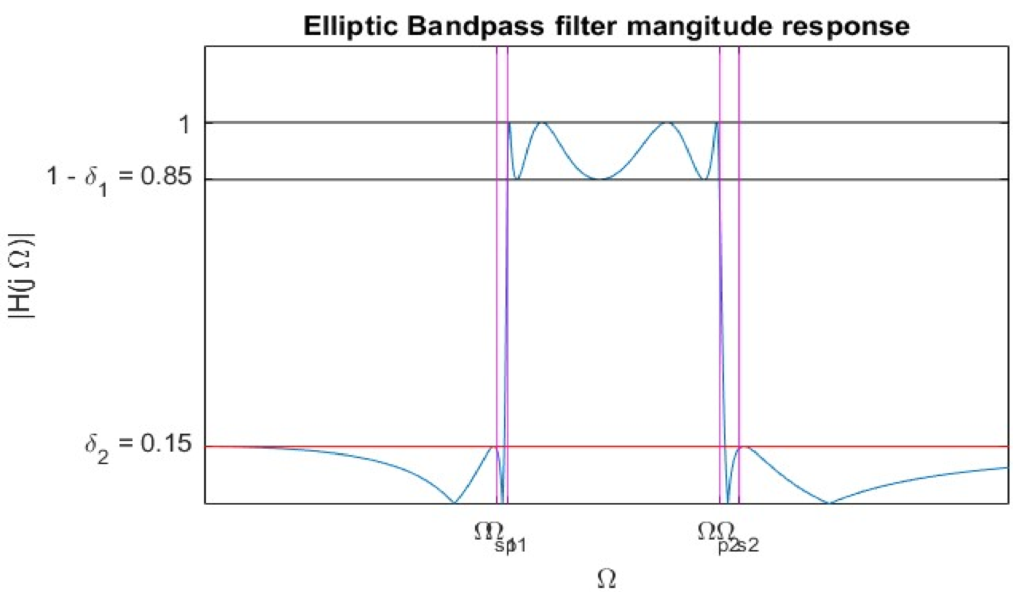
We now use the inverse transformation to convert the equivalent Elliptic lowpass transfer function to get the Bandpass transfer function using the relation:

$$s_L <- \frac{s^2 + \Omega_0^2}{B\Omega}$$

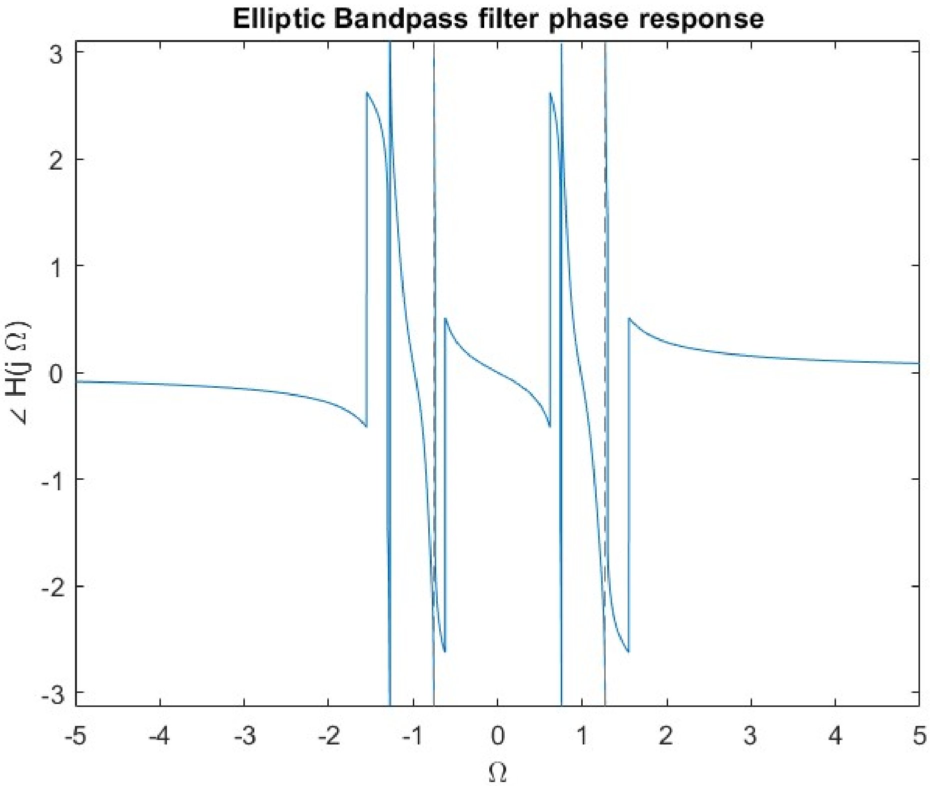
where *B* and Ω0 have the values found above

$$H_{lpf\left(\frac{s^2+\Omega_0^2}{B\Omega}\right)} = H_{bpf}(s)$$

$$H_{bsf}(s) = \frac{0.15s^8 + 0.7557s^6 + 1.2195s^4 + 0.7030s^2 + 0.1298}{s^8 + 0.4049s^7 + 4.3205s^6 + 1.2805s^5 + 6.5219s^4 + 1.2351s^3 + 4.0194s^2 + 0.3633s + 0.8655}$$



Magnitude Response of the Elliptic BPF



Phase Response of the Elliptic BPF

8.DiscreteTime Bandpass Transfer Function

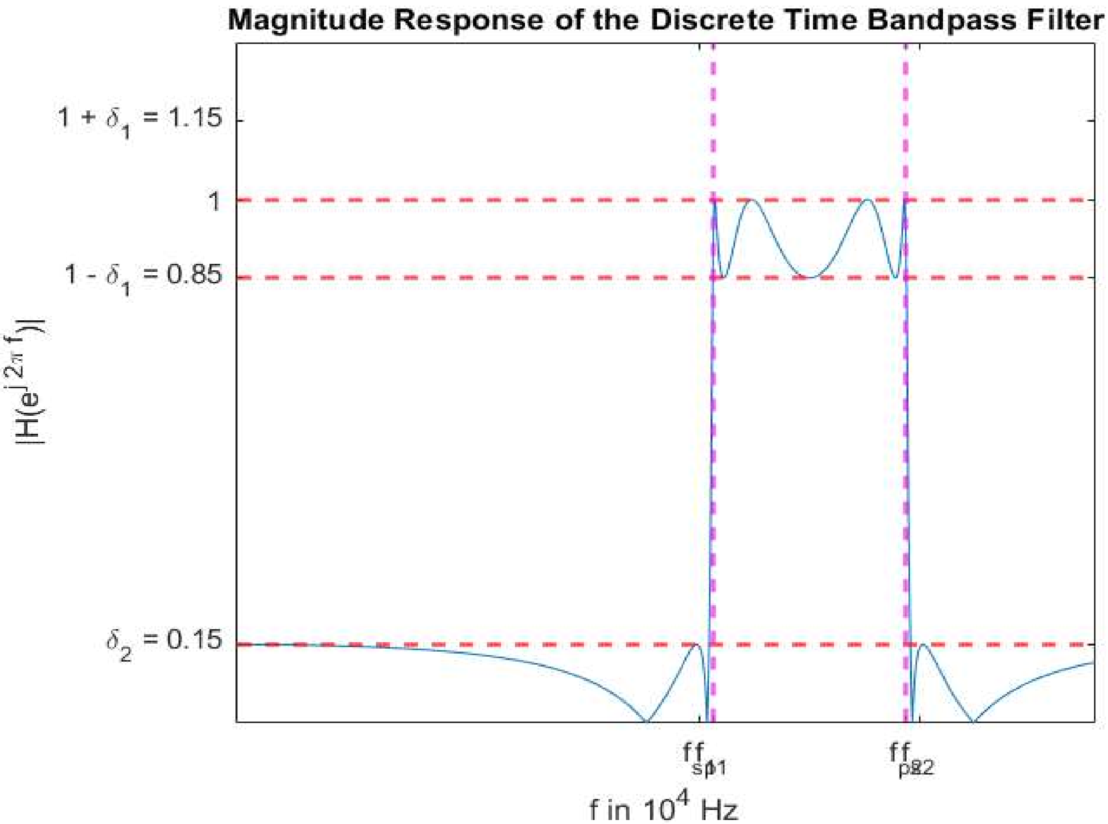
We now make use of the Bilinear Transformation to convert the analog bandpass filter into a discrete bandpass filter in the normalized angular frequency domain. The transform is as follows:

$$s \quad <- \quad \frac{1-z^{-1}}{1+z^{-1}}$$

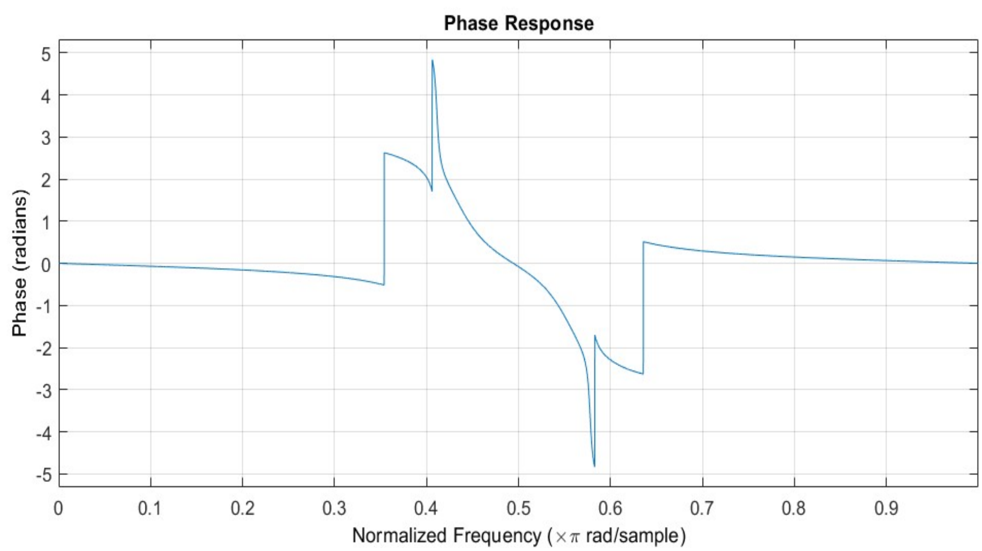
The discrete time filter transfer function is

$$H_{bpf}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) = H(z)$$

$$H(z) = \frac{0.1478z^8 - 0.0186z^7 + 0.4393z^6 - 0.0459z^5 + 0.6153z^4 - 0.0459z^3 + 0.4393z^2 - 0.0186z + 0.1478}{z^8 - 0.1310z^7 + 3.2597z^6 - 0.3318z^5 + 4.3133z^4 - 0.3008z^3 + 2.6875z^2 - 0.0970z + 0.6718}$$



Magnitude Response of Discrete Time Bandpass Filter



Phase Response of Discrete Time Bandpass Filter

Filter-4(Elliptic Bandstop) Details

1.Un-normalized Discrete Time Filter Specifications

Filter number: 27

Since filter number < 80, m = 27.

q(m) = greatest integer less than 0.1m = floor(0.1m) = 2

r(m) = m – 10q(m) = 7

BL(m) = 5 + 3 q(m) + 11 r(m) = 88

BH(m) = BL(m) + 25 = 113

The second filter is given to be a Band-Stop filter with stopband from BL(m) kHz to BH(m) kHz. Therefore, the specifications are:

- Stopband: 88 kHz to 113 kHz
- Transition band: 3 kHz on either side of passband
- Passband: 0 to 85 kHz and 116 kHz to 200 kHz (Sampling rate is 400 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple
- Stopband Nature: Equiripple

2. Normalized Digital Filter Specifications

Sampling rate: 400 kHz

In the normalized frequency axis, sampling rate corresponds to 2π.

This any frequency (Ω), up to 200 kHz can be represented on the normalized axis ω as:

$$\omega = \frac{\Omega * 2\pi}{\Omega_s(Sampling\ rate)}$$

Therefore, the corresponding normalized discrete filter specifications are:

- Stopband: 0.44π to 0.565π
- Transition band: 0.015π on either side of passband
- Passband: 0 to 0.425π and 0.58π to π (Sampling rate is 400 kHz)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple

- Stopband Nature: Equiripple

3.Analog filter specifications for Band-pass filter using Bilinear transformation

We use the following transformation to move from the *z* domain to the *s* domain.

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}$$

If we put *z* = *jω* and *s* = *j*Ω, we get

$$\Omega = \tan\left(\frac{\omega}{2}\right)$$

Applying the bilinear transformation to the frequencies at the band edges, we get:

ω	Ω
0.44π	0.8273
0.565π	1.2283
0.425π	0.7883
0.58π	1.2892
0	0
π	∞

Therefore, the corresponding analog filter specifications for the same type of analog filter using the bilinear transformation are:

- Stopband: 0.8273(Ω_{S1}) to 1.2283(Ω_{S2})
- Passband: 0 to 0.7883(Ω_{P1}) and 1.2892(Ω_{P2}) to ∞
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple
- Stopband Nature: Equiripple

4.Frequency Transformation & Relevant Parameters

We need to transform a Band-Stop analog filter to a Lowpass analog filter. We require two parameters in such a case. We can make use of the Bandstop transformation which is given as:

$$\Omega_L = \frac{B\Omega}{\Omega_0^2 - \Omega^2}$$

The two parameters in the above equation are B and Ω_0. They can be determined using the specifications of the bandpass analog filter using the following relations:

$$\Omega_0 = \sqrt{\Omega_{P1}\Omega_{P2}} = \sqrt{0.7883 * 1.2892} = 1.0081$$

$$B = \Omega_{P2} - \Omega_{P1} = 1.2892 - 0.7883 = 0.5009$$

Ω	Ω_L
0^+	0^+
0.7883	+1
0.8273	+1.2488
$1.0081\ (\Omega_0^-)$	∞
$1.0081\ (\Omega_0^+)$	$-\infty$
1.2283	-1.2494
1.2892	-1
∞	0^-

5. Frequency Transformed Lowpass Analog Filter Specifications

- Passband edge: 1 (Ω_{LP})
- Stopband edge: min ($\Omega_{LS1}, -\Omega_{LS2}$) = 1.2488 ($\Omega_{LS}$)
- Tolerance: 0.15 in magnitude for both Passband and Stopband
- Passband Nature: Equiripple
- Stopband Nature: Equiripple

6. Analog Lowpass Transfer function

The general transfer function for an **Elliptic** filter is given by

$$H_{LPF}(s_L)\, H_{LPF}(-s_L) = \frac{1}{1 + \epsilon^2\, F_N^2(\frac{s_L}{j\Omega_{Lp}})}$$

with

$$F_N(w) = \mathrm{cd}(N\,uK_1, k_1)$$

and

$$w = \mathrm{cd}(uK, k))$$

where cd(x, k) denotes the Jacobian elliptic function cd with modulus k and real quarter-period K.

6.1. Jacobian Elliptic Functions

The elliptic function w = sn(z, k) is defined as follows:

$$z = \int_0^{\phi} \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} = \int_0^{w} \frac{dt}{\sqrt{(1-t^2)(1-k^2t^2)}} \quad, w = \sin\left(\phi(z,k)\right)$$

There are three more relevant elliptic functions, cn, dn, and cd defined as:

$$w = \mathrm{cn}(z, k) = \cos(\phi(z, k))$$

$$w = dn(z,k) = \sqrt{1-k^2 sn^2(z,k)}$$

$$w = cd(z,k) = \frac{cn(z,k)}{dn(z,k)}$$

We define the *complete elliptic integral of first kind* $K(k)$ or simply K as the value of z when sn(z, k) = 1 i.e. when $\phi = \pi/2$

$$K(k) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}}$$

Clearly cd(K, k) = 0. Associated with an elliptic modulus k, there is a complementary modulus $k' = \sqrt{1-k^2}$ and its associated complete elliptic integral $K(k')$ denoted by $K'(k)$ or simply K'

$$K'(k) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-k'^2\sin^2\theta}} = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-(1-k^2)\sin^2\theta}}$$

The significance of K and K' is that sn and cd are *doubly-periodic functions* in the z plane with a real period $4K$ and a complex period $2jK'$. Some of the useful properties are as follows:

$$
\begin{aligned}
\mathrm{cd}(z+(2n-1)K,k) &= (-1)^n\,\mathrm{sn}(z,k)\\
\mathrm{cd}(z+2nK,k) &= (-1)^n\,\mathrm{cd}(z,k)\\
\mathrm{cd}(z+j\,K',k) &= \frac{1}{k\,\mathrm{cd}(z,k)}\\
\mathrm{cd}(j\,z,k) &= \frac{1}{\mathrm{dn}(z,k')}\\
\mathrm{cd}(j\,K',k) &= \frac{1}{k}
\end{aligned}
$$

cd(z, k) has zeroes at $(2m+1)K + j\,2nK'$ and poles at $(2m+1)K + j\,(2n+1)K'$ for any integers m and n in the z plane.

6.2. Elliptic Filter Parameters

The order of the filter N is determining by putting a constraint that the magnitude response at the passband edge is not less than $1-\delta 1$ and at the stopband edge not greater than $\delta 2$.

$$N = \left\lceil \frac{KK1'}{K'K1} \right\rceil$$

where K, K', $K1$, $K1'$ are defined for the complete elliptic integrals of modulus k and $k1$ given by

$$k = \frac{\Omega_{LP}}{\Omega_{LS}} = \frac{1}{1.2488} = 0.8007$$

$$k1 = \sqrt{\frac{D_1}{D_2}} = 0.0940$$

We then have

Parameter	Value
K	1.9975
K’	1.7496
K1	1.5743
K1’	3.7566
N	3

We now have to find the poles of the the transfer function. The poles of the transfer function are given by the zeros of the denominator and the zeros of the transfer function are given by the poles of $F_N\left(\frac{S_L}{j\Omega L_p}\right)$. The zeros and poles of the transfer function are given by

$z_i = j\,\Omega L_p\,(k\,\zeta_i)^{-1}$, $i = 1, 2, \ldots, \lfloor N/2 \rfloor$

$p_i = j\,\Omega_{Lp}\,\mathrm{cd}((u_i - jv_0)K, k)$, $i = 1, 2, \ldots, \lfloor N/2 \rfloor$

$p_0 = j\,\Omega_{Lp}\,\mathrm{sn}(jv_0K, k)$

where

$\zeta_i = \mathrm{cd}(u_i K, k)$

$u_i = \frac{2i-1}{N}$, $i = 1, 2, \ldots, \lfloor N/2 \rfloor$

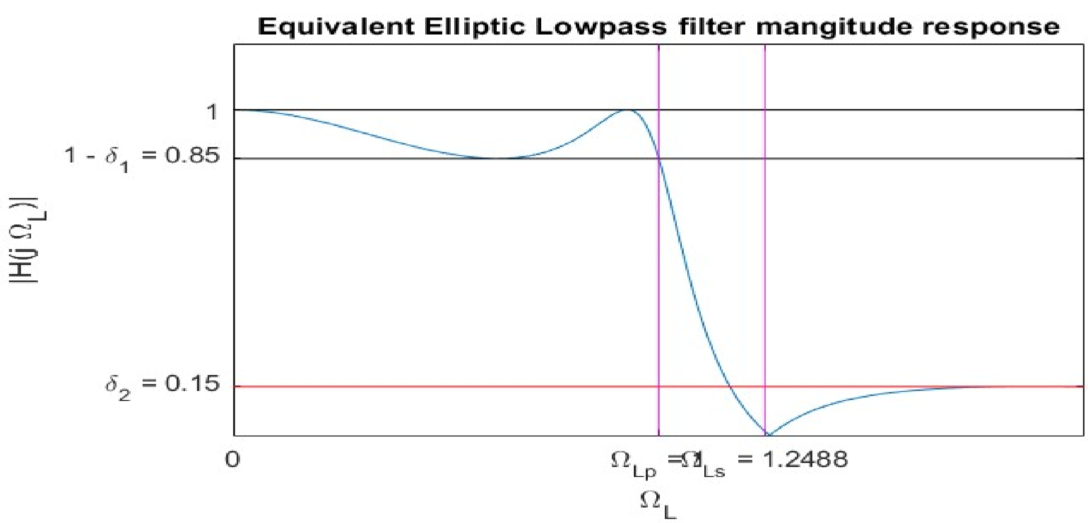
$v_0 = -\frac{j}{NK1} Sn^{-1}\left(\frac{j}{e}, k1\right)$

As N is odd, the zeros and poles are

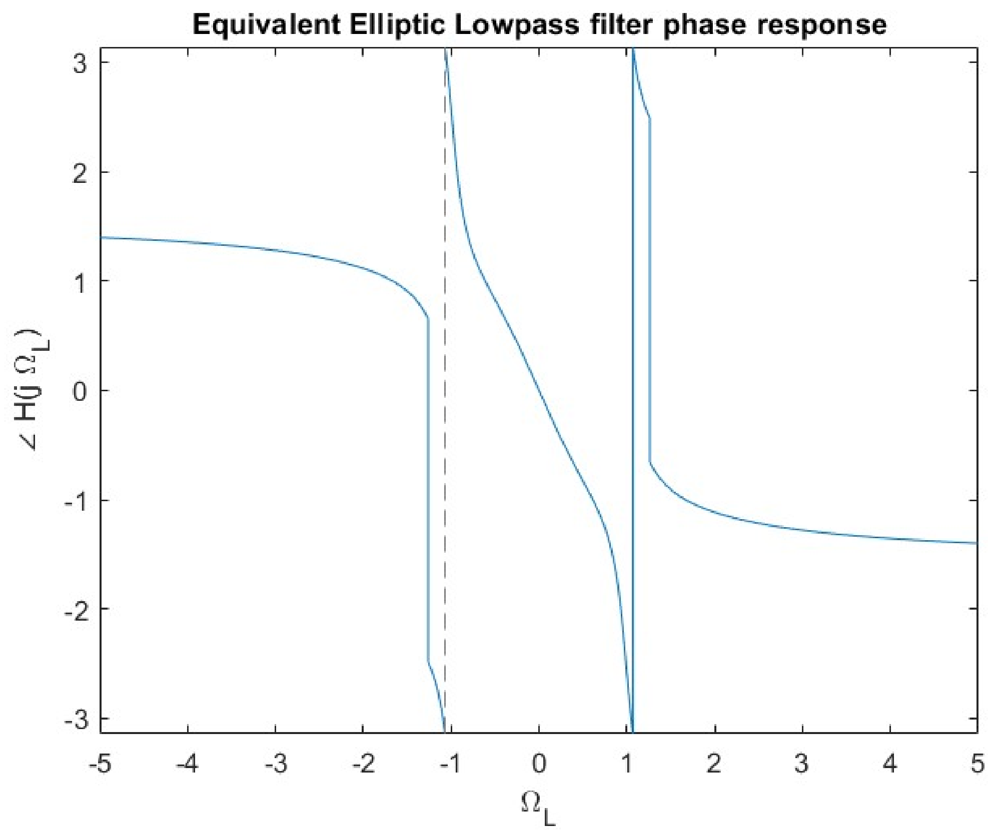
z1	-j1.26052
z2	j1.26052
p1	-0.623209
p2	-0.115375 - j0.993635
p3	-0.115375 +j0.993635

We then have

$$H_{LPf}(s_L) = \frac{1.0212 s_L^2 + 1.6226}{2.6020 s_L^3 + 2.2220 s_L^2 + 2.9778 s_L + 1.6226}$$



Magnitude Response of the equivalent Elliptic Lowpass Filter



Phase Response of the equivalent Elliptic Lowpass Filter

7.Analog Bandstop Transfer function

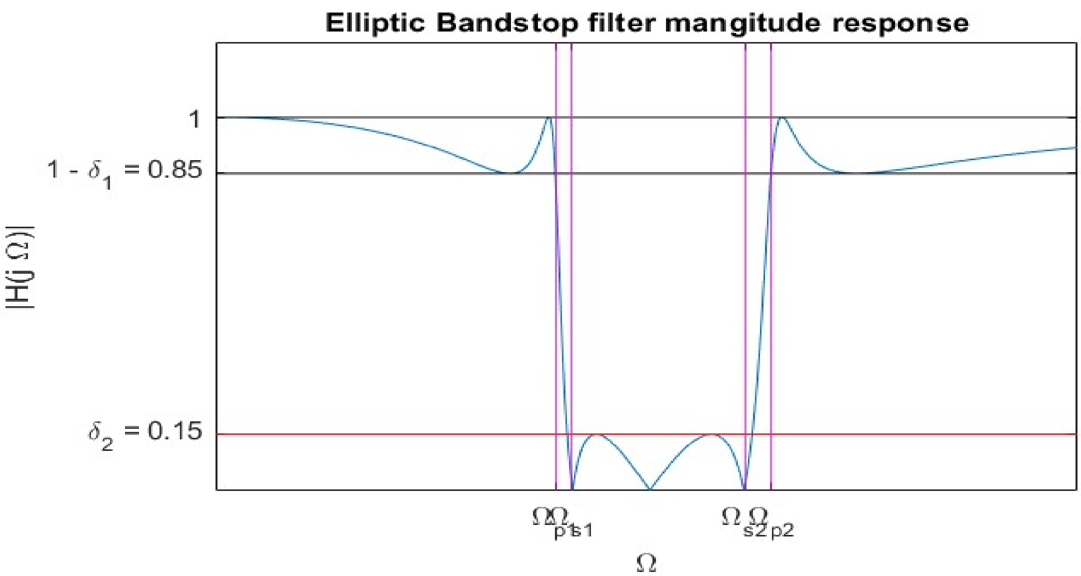
We now use the inverse transformation to convert the equivalent Elliptic lowpass transfer function to get the Bandstop transfer function using the relation:

$$s_L <- \quad \frac{Bs}{s^2 + \Omega_0^2}$$

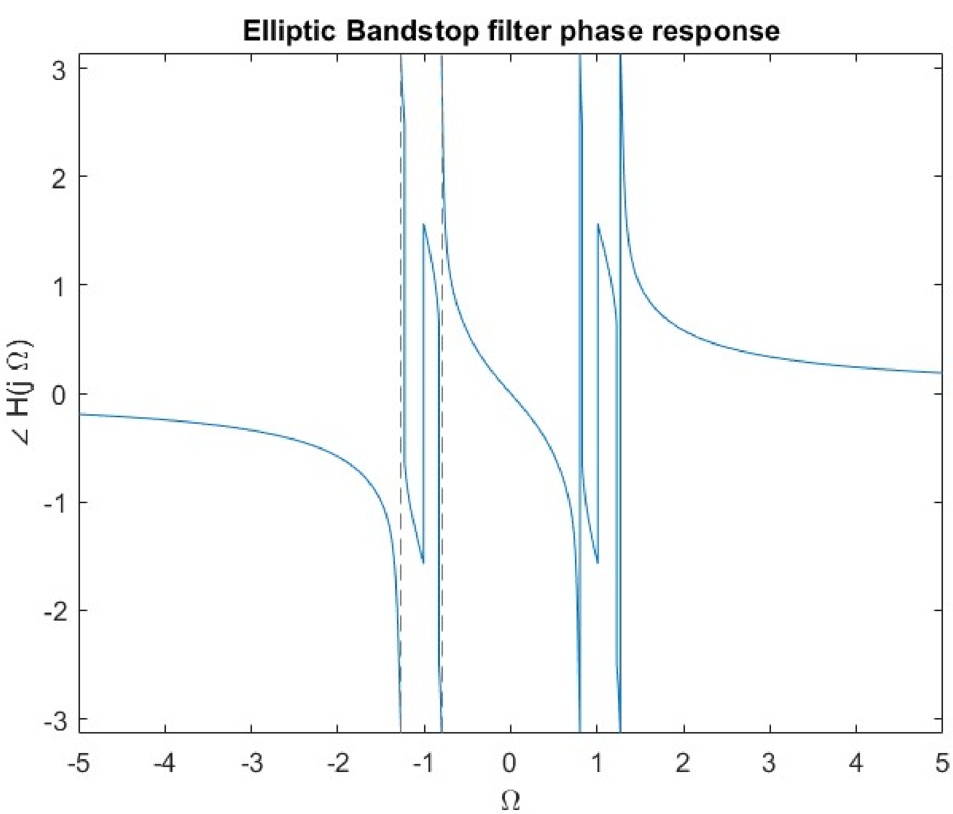
where *B* and Ω0 have the values found above

$$H_{lpf\left(\frac{Bs}{s^2+\Omega_0^2}\right)} = H_{bsf}(s)$$

$$H_{bsf}(s) = \frac{s^6 + 3.2067s^4 + 3.2589s^2 + 1.0496}{s^6 + 0.9192s^5 + 3.3924s^4 + 2.0699s^3 + 3.4476s^2 + 0.9494s + 1.0496}$$



Magnitude Response of Elliptic Bandstop filter



Phase Response of Elliptic Bandstop filter

8.DiscreteTime Bandpass Transfer Function

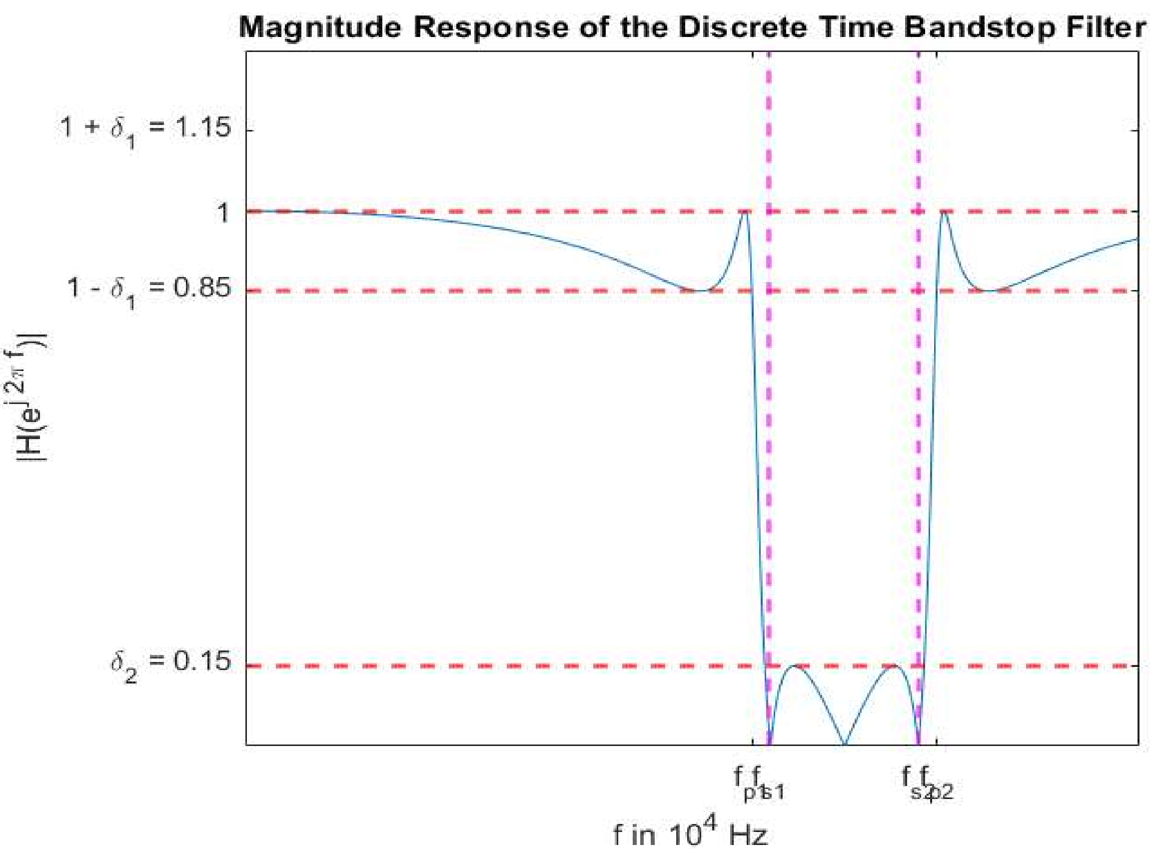
We now make use of the Bilinear Transformation to convert the analog bandstop filter into a discrete bandpass filter in the normalized angular frequency domain. The transform is as follows:

$$s \quad <- \quad \frac{1-z^{-1}}{1+z^{-1}}$$

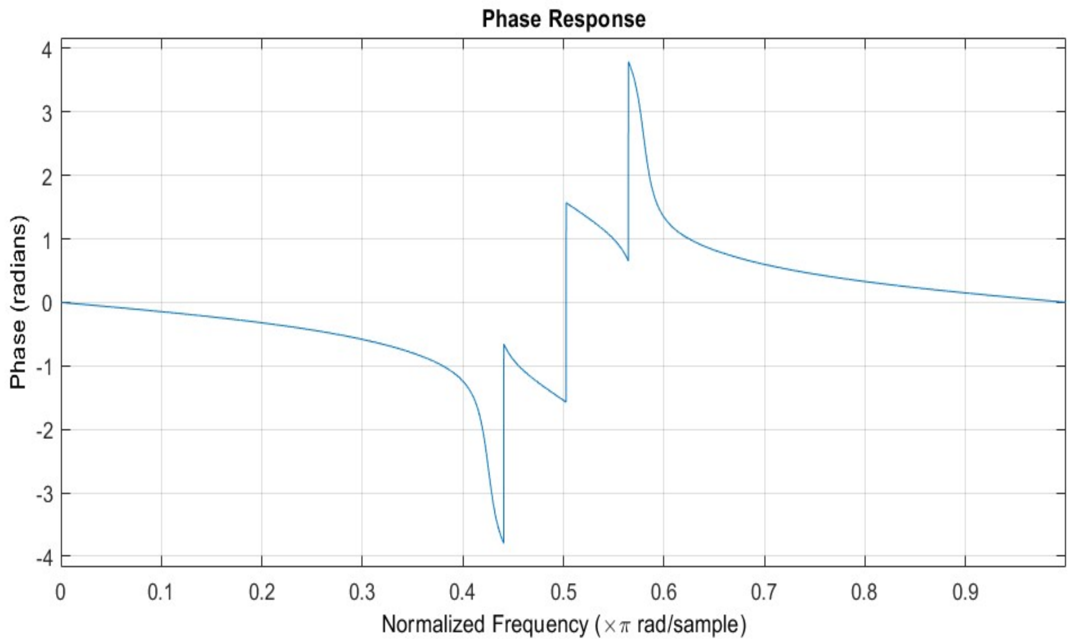
The discrete time filter transfer function is

$$H_{bsf}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) = H(z)$$

$$H(z) = \frac{0.6638z^6 + 0.0313z^5 + 1.8926z^4 + 0.0611z^3 + 1.8926z^2 + 0.0313z + 0.6638}{z^6 + 0.0412z^5 + 2.1077z^4 + 0.0601z^3 + 1.6191z^2 + 0.0224z + 0.3859}$$



Magnitude Response of Discrete Time Bandstop Filter



Phase Response of Discrete Time Bandstop Filter

Conclusions

After realising the bandpass and bandstop filters for a given set of specifications in various forms, i was able to make the following conclusions –

- Realisation using an elliptic approximation gives the least order, i.e, it requires mininum resources. The order of resources required is as follows (for a given set of specifications): Elliptic (4 or 3) < Chebyshev (5) < Butterworth (19) < Kaiser FIR (124 or 93)

- The elliptic filter has the sharpest transition from passband to stopband or vice versa for the same specifications. In particular, the decreasing order of transition sharpness is Elliptic > Chebyshev > Butterworth > Kaiser FIR

- The FIR realisation gives a linear phase response in the passband region. The nonlinearity of the phase response decreases in the following order- Elliptic > Chebyshev > Butterworth > Kaiser FIR

Peer Review

1. Review Received

Review by : Yeshwanth Sunnapu, 19D070071, Group 16

I have reviewed this report made by Badal Varshney, I certify that he has completed all the steps for the design of a bandpass and a bandstop filter using IIR, FIR and elliptic filter design methods.

2. Review Given

Review given to: Vangala porus sai sujith, 19D070065, Group 16

I have reviewed the report made by Vangala porus sai sujith and I certified that he has

completed all the steps for the design of a bandpass and a bandstop filter using IIR, FIR filter design methods.

Appendix

A. IIR Bandpass Filter (Matlab Code)

```
clc; clear all? close all?
 tic;
 %% Poles of H_{analog }(s_L) H_{analog}(?s_L)
 N = 19;
 Omega_c = 1.0290;
k = 1:2*N;
poles = Omega_c .* exp(1i .* (pi/2) .* (1 + (2.*k + 1)./ N));
figure();
plot(poles, '.', 'MarkerSize', 20);
xlim([-1.5 1.5]);
ylim([-1.5 1.5]);
daspect([1 1 1]);
hold on;
x = linspace(-pi, pi, 10000);
a = Omega_c .* cos(x);
b = Omega_c .* sin(x);
plot(a,b);
hold on;
plot(0,0,'r*');
left_plane_poles = [];
for i=1:size(poles,2)
    hold on;
    plot([0, real(poles(1,i))], [0, imag(poles(1,i))], 'r-');
    if real(poles(1,i)) < 0
         disp(poles(1,i));
        left_plane_poles = [left_plane_poles, poles(1,i)];
    end
end
plot([0, 0],[-1.5, 1.5], 'k-');
plot([-1.5, 1.5],[0, 0], 'k-');
title('Poles of $H_{LPF}(s_L) \, H_{LPF}(-s_L)$ in the $s_L$ plane',
'Interpreter', 'latex');
xlabel('\Sigma_k');
ylabel('\Omega_k');
%% Finding poles of H_{analog}(s_L)
syms s x;
f = 1;
for i=1:size(left_plane_poles,2)
    f = f * (s - left_plane_poles(1,i));
end
%disp(1.7214/f);
[num,den]=numden(f);
```

```
disp(num);
disp(den);
%% Magnitude and Phase response of H_{analog}(s_L)

g = subs(f, s, 1i * x);
figure();
fplot(abs(1.7214/g)); %(omega_c)^N

hold on;
fplot(s - s - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(x-x+1, 'k-', 'Markersize', 10);
hold on;
fplot(x - x + 0.15, 'r-', 'Markersize', 10);
xline(1, 'magenta-');
hold on;
xline(1.1403,'magenta-');
axis([0 2 0 1.2]);
set(gca, 'XTick', [0, 1, 1.1403], 'xticklabel', {'0', '\Omega_{Lp} =
1','\Omega_{Ls} = 1.1403'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
daspect([1 1 1]);
title('Equivalent Butterworth Lowpass filter mangitude response');
xlabel('\Omega_L');
ylabel('|H(j \Omega_L)|');

figure();
fplot(angle(1.7214/g));
xlabel('\Omega_L');
ylabel('\angle H(j \Omega_L)');
title('Equivalent Butterworth Lowpass filter phase response');
%% Bandpass analog frequency transformation
syms s1;
Omega_0 = 0.9821;
B = 0.5264;
s1 = (s^2 + Omega_0^2) ./ (B * s);
h = subs(f, s, s1);

%disp(1.7214/h); % transformed i.e., H_{analog}(s)

a = vpa(1.7214/h, 5); % bandpass system function

b = subs(a, s, 1i * x); % to get the frequency response
figure();
fplot(abs(b));
set(gca, 'XTick', [0.7265, 0.7536, 1.2799, 1.3270], 'xticklabel',
{'\Omega_{s1}', '\Omega_{p1}', '\Omega_{p2}', '\Omega_{s2}'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
hold on;
xline(0.7265, 'magenta-');hold on;
xline(0.7536, 'magenta-');hold on;
xline(1.2799, 'magenta-');hold on;
xline(1.3270, 'magenta-');
hold on;
```

```
fplot(s - s - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(x-x+1, 'k-', 'Markersize', 10);
hold on;
fplot(x - x + 0.15, 'r-', 'Markersize', 10);
daspect([1 1 1]);
title('Butterworth Bandpass filter mangitude response');
xlabel('\Omega');
ylabel('|H(j \Omega)|');
axis([0 2 0 1.2]);

figure();
fplot(angle(b));
daspect([1 1 1]);
title('Butterworth Bandpass filter phase response');
xlabel('\Omega');
ylabel('\angle H(j \Omega)');
%% bilinear transformation
syms w;
c = subs(a, s, (1-1/w) / (1+1/w));
[Nz, Dz] = numden(1/c);
Nz = sym2poly(Nz);
Dz = sym2poly(Dz);
% disp('######################################');
 %disp(Nz);
% disp(Dz);
%%
[H,f] = freqz(Dz,Nz,1024*1024, 540e3);
plot(f,abs(H));
axis([0 180e3 0 1.3]);
hold on;
yline(1.15, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.85, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.15, 'red--', 'LineWidth', 1.5);
hold on;
xline(111e3, 'magenta--', 'LineWidth', 1.5);
hold on;
xline(156e3, 'magenta--', 'LineWidth', 1.5);
xlabel('f in 10^4 Hz');
ylabel('|H(e^{j 2\pi f})|');
title('Magnitude Response of the Discrete Time Bandpass Filter');
set(gca, 'XTick', [108e3, 111e3, 156e3, 159e3], 'xticklabel',
{'f_{s1}', 'f_{p1}', 'f_{p2}', 'f_{s2}'});
set(gca, 'YTick', [0.15, 0.85, 1, 1.15], 'yticklabel', {'\delta_2 =
0.15', '1 - \delta_1 = 0.85', '1', '1 + \delta_1 = 1.15'});
%%
figure();
plot(f, angle(H));
 set(gca, 'XTick', [108e3, 111e3, 156e3, 159e3], 'xticklabel',
{'f_{s1}', 'f_{p1}', 'f_{p2}', 'f_{s2}'});
xlim([1e4 180e3]);
ylabel('\angle H(e^{j 2\pi f})');
xlabel('f in Hz');
title('Phase Response of the Discrete Time Bandpass Filter');
```

B. FIR Bandpass Filter (Matlab Code)

```
%%
clc; clear all; close all;
%% BP Filter Specs
omega_s1 = 0.4 * pi;
omega_s2 = 0.5889 * pi;
omega_p1 = 0.4111 * pi;
omega_p2 = 0.5778 * pi;
transition_bw = 0.0111 * pi;
delta = 0.15;
omega_c1 = 0.5 * (omega_p1 + omega_s1);
omega_c2 = 0.5 * (omega_p2 + omega_s2);
%% Kaiser window parameters
A = -20 * log10(delta);
fprintf('A = %f\n', A);

width_min = ceil((A-7.95) / (2.285*0.0111*pi));
fprintf('Minimum width = %d\n', width_min);

alpha = -1;
if A < 21
    alpha = 0;
elseif A >=21 && A <= 50
    alpha = 0.5842 * (A - 21) ^ 0.4 + 0.07886 * (A - 21);
elseif A > 50
    alpha = 0.1102 * (A - 8.7);
else

end
beta = alpha /width_min;
fprintf('beta = %f\n', beta);
%%
width = width_min + 16;
w = kaiser(width,beta);
ideal_bpf = ideal_lpf(omega_c2, width) - ideal_lpf(omega_c1, width);
fir_bpf = ideal_bpf .* w';
[H,f] = freqz(fir_bpf,1,1024, 540e3);
figure();
plot(f, abs(H));
hold on;
xline(111e3, 'magenta--', 'LineWidth', 1.5);
hold on;
xline(156e3, 'magenta--', 'LineWidth', 1.5);
hold on;
yline(1.15, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.85, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.15, 'red--', 'LineWidth', 1.5);
xlabel('f in 10^4 Hz');
ylabel('|H(e^{j2 \pi f})');
title('Magnitude Response of Discrete Time FIR Bandpass Filter');
```

```
set(gca, 'XTick', [108e3, 111e3, 156e3, 159e3], 'xticklabel',
{'f_{s1}', 'f_{p1}', 'f_{p2}', 'f_{s2}'});
set(gca, 'YTick', [0.15, 0.85, 1, 1.15], 'yticklabel', {'\delta_2 =
0.15', '1 - \delta_1 = 0.85', '1', '1 + \delta_1 = 1.15'});

%%
disp(fir_bpf);
%%
fvtool(fir_bpf, 'Analysis', 'Phase');
fvtool(fir_bpf, 'Analysis', 'Impulse');
```

C. IIR Bandstop Filter (Matlab Code)

```
clc; clear all; close all;
% transformed normalized filter specs
syms x B Omega_0 f(x);
B = 0.5009;
Omega_0 = 1.0081;
f(x) = (B * x) / (-x^2 + Omega_0^2);
%disp(vpa(f(0.8273), 5));
%disp(vpa(f(1.2283), 5));
%disp(vpa(f(0.7883), 5));
%disp(vpa(f(1.2892), 5));
%% equilavent lowpass filter specs
D2 = 43.44;
D1 = 0.384;
Omega_ls = 1.2488;
Omega_lp = 1;
epsilon = sqrt(D1);
N = ceil((acosh(sqrt(D2/D1))) / (acosh(Omega_ls / Omega_lp)));
fprintf('N = %d\n', N);
%% finding out the left half poles
k = 0:2 * N - 1;
Ak = (2 * k + 1) .* (pi /(2 * N));
Bk = (1 / N) * asinh(1 / epsilon);
poles = (-1 * sin(Ak) * sinh(Bk)) + 1i * (cos(Ak) * cosh(Bk));
figure();
plot(poles, '.', 'MarkerSize', 20);
daspect([1 1 1]);
hold on;
x = linspace(-pi, pi, 10000);
a = sinh(Bk) .* cos(x);
b = cosh(Bk) .* sin(x);
plot(a,b);
hold on;
plot(0,0,'r*');
xlim([-1.5 1.5]);
ylim([-1.5 1.5]);
left_plane_poles = [];
for i=1:size(poles,2)
    hold on;
    plot([0, real(poles(1,i))], [0, imag(poles(1,i))], 'r-');
    if real(poles(1,i)) < 0
        disp(poles(1,i));
```

```
        left_plane_poles = [left_plane_poles, poles(1,i)];
    end
end
plot([0, 0],[-1.5, 1.5], 'k-');
plot([-1.5, 1.5],[0, 0], 'k-');
title('Poles of $H_{LPF}(s_L) \, H_{LPF}(-s_L)$ in the $s_L$ plane',
'Interpreter', 'latex');
xlabel('\Sigma_k');
ylabel('\Omega_k');
%% Analog Lowpass transfer function in s_L domain
num = 1;
den = 1;
syms sl;
for i=1:size(left_plane_poles, 2)
    num = num * left_plane_poles(1,i);
    den = den * (sl - left_plane_poles(1,i));
end
H_LPF = num / (den);
disp(expand(vpa(den , 5)));
disp(num);
disp(den);
syms Omega_L;
H_LPF_freq = subs(H_LPF, sl, 1i * Omega_L);
figure();
fplot(abs(H_LPF_freq));
hold on;
fplot(sl - sl - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(sl-sl+1, 'k-', 'Markersize', 10);
hold on;
fplot(sl - sl + 0.15, 'r-', 'Markersize', 10);
xline(1, 'magenta-');
hold on;
xline(1.2488,'magenta-');
axis([0 2 0 1.2]);
set(gca, 'XTick', [0, 1, 1.2488], 'xticklabel', {'0', '\Omega_{Lp} =
1','\Omega_{Ls} = 1.2488'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
daspect([1 1 1]);
title('Equivalent Chebyshev Lowpass filter mangitude response');
xlabel('\Omega_L');
ylabel('|H(j \Omega_L)|');

figure();
fplot(angle(H_LPF_freq));
xlabel('\Omega_L');
ylabel('\angle H(j \Omega_L)');
title('Equivalent Chebyshev Lowpass filter phase response');
%% Chebyshev lowpass magnitude response
syms s;
H_BSF = subs(H_LPF, sl, (B * s)/(s^2 + Omega_0^2));
[N, D] = numden(H_BSF);

disp(N);
disp(expand(vpa(N, 2)));
```

```
disp(D);
disp(expand(vpa(D, 2)));
%% Chebyshev bandstop magnitude response
syms Omega;
N1 = subs(N* 1e-138, s, 1i * Omega);
D1 = subs(D * 1e-138, s, 1i * Omega);
H_BSF_freq_resp = (N1) / (D1);
H_BSF_freq_resp = vpa(H_BSF_freq_resp, 2);

figure();
fplot(abs(H_BSF_freq_resp));
set(gca, 'XTick', [0.7883, 0.8273, 1.2283, 1.2892], 'xticklabel',
{'\Omega_{p1}', '\Omega_{s1}', '\Omega_{s2}', '\Omega_{p2}'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
hold on;
xline(0.7883, 'magenta-');hold on;
xline(0.8273, 'magenta-');hold on;
xline(1.2283, 'magenta-');hold on;
xline(1.2892, 'magenta-');
hold on;
fplot(s - s - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(s-s+1, 'k-', 'Markersize', 10);
hold on;
fplot(s -s + 0.15, 'r-', 'Markersize', 10);
daspect([1 1 1]);
title('Chebyshev Bandstop filter mangitude response');
xlabel('\Omega');
ylabel('|H(j \Omega)|');
axis([0 2 0 1.2]);
%% Chebyshev bandstop phase response
figure();
fplot(angle(H_BSF_freq_resp));
daspect([1 1 1]);
title('Chebyshev Bandstop filter phase response');
xlabel('\Omega');
ylabel('\angle H(j \Omega)');
%% Bilinear transformation from analog to digital
syms z;
Hz = subs(H_BSF, s, (z - 1) / (z + 1));
[Nz, Dz] = numden(Hz);
disp(expand(vpa(Nz, 2)));
disp(expand(vpa(Dz, 2)));
Nzz = sym2poly(Nz);
Dzz = sym2poly(Dz);
[H,f] = freqz(Nzz,Dzz,1024*1024, 400e3);
plot(f,abs(H));
axis([0 150e3 0 1.3]);
hold on;
yline(1.15, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.85, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.15, 'red--', 'LineWidth', 1.5);
```

```
hold on;
xline(88e3, 'magenta--', 'LineWidth', 1.5);
hold on;
xline(113e3, 'magenta--', 'LineWidth', 1.5);
xlabel('f in 10^4 Hz');
ylabel('|H(e^{j 2\pi f})|');
title('Magnitude Response of the Discrete Time Bandstop Filter');
set(gca, 'XTick', [85e3, 88e3, 113e3, 116e3], 'xticklabel',
{'f_{p1}', 'f_{s1}', 'f_{s2}', 'f_{p2}'});
set(gca, 'YTick', [0.15, 0.85, 1, 1.15], 'yticklabel', {'\delta_2 =
0.15', '1 - \delta_1 = 0.85', '1', '1 + \delta_1 = 1.15'});
%% Phase response of DT BSF
figure();
plot(f, angle(H));
xlim([1e4 150e3]);
ylabel('\angle H(e^{j 2\pi f})');
xlabel('f in Hz');
title('Phase Response of the Discrete Time Bandstop Filter');
hold on;
xline(88e3, 'magenta--', 'LineWidth', 1);
hold on;
xline(113e3, 'magenta--', 'LineWidth', 1);
set(gca, 'XTick', [1e4, 85e3, 88e3, 113e3, 116e3, 150e3],
'xticklabel', {1e4 ,'f_{p1}', 'f_{s1}', 'f_{s2}', 'f_{p2}', 150e3});
%%
% fvtool(Nzz, Dzz);
```

D. FIR Bandstop Filter (Matlab Code)

```
%%
clc; clear all; close all;
%% BS Filter Specs
omega_s1 = 0.44 * pi;
omega_s2 = 0.565 * pi;
omega_p1 = 0.425 * pi;
omega_p2 = 0.58 * pi;
transition_bw = 0.015 * pi;
delta = 0.15;
omega_c1 = 0.5 * (omega_p1 + omega_s1);
omega_c2 = 0.5 * (omega_p2 + omega_s2);
%% Kaiser window parameters
A = -20 * log10(delta);
fprintf('A = %f\n', A);

width_min = ceil((A-7.95) / (2.285*0.015*pi));
fprintf('Minimum width = %d\n', width_min);

alpha = -1;
if A < 21
    alpha = 0;
elseif A >=21 && A <= 50
    alpha = 0.5842 * (A - 21) ^ 0.4 + 0.07886 * (A - 21);
elseif A > 50
    alpha = 0.1102 * (A - 8.7);
```

```
else

end
beta = alpha /width_min;
fprintf('beta = %f\n', beta);
%% Magnitude Response
width = width_min + 13;
w = kaiser(width,beta);
ideal_bsf = ideal_lpf(pi, width) - ideal_lpf(omega_c2, width) +
ideal_lpf(omega_c1, width);
fir_bsf = ideal_bsf .* w';
[H,f] = freqz(fir_bsf,1,1024, 400e3);
figure();
plot(f, abs(H));
hold on;
xline(88e3, 'magenta--', 'LineWidth', 1.5);
hold on;
xline(113e3, 'magenta--', 'LineWidth', 1.5);
hold on;
yline(1.15, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.85, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.15, 'red--', 'LineWidth', 1.5);
xlabel('f in 10^4 Hz');
ylabel('|H(e^{j2 \pi f})');
title('Magnitude Response of Discrete Time FIR Bandstop Filter');
set(gca, 'XTick', [85e3, 88e3, 113e3, 116e3], 'xticklabel',
{'f_{p1}', 'f_{s1}', 'f_{s2}', 'f_{p2}'});
set(gca, 'YTick', [0.15, 0.85, 1, 1.15], 'yticklabel', {'\delta_2 =
0.15', '1 - \delta_1 = 0.85', '1', '1 + \delta_1 = 1.15'});
%%
disp(fir_bsf);
%%
fvtool(fir_bsf, 'Analysis', 'Phase');
fvtool(fir_bsf, 'Analysis', 'Impulse');
```

E. Helper Low-pass Filter for Bandpass and Bandstop Filter Design (Matlab Code)

```
function hd = ideal_lpf(wc,M);

alpha = (M-1)/2;
n = [0:1:(M-1)];
m = n - alpha + eps;
hd = sin(wc*m) ./ (pi*m);
end
```

F. Elliptic Bandpass Filter (Matlab Code)

```
% Please execute this code section wise
% so that the values of the variables
% settle and are stable
```

```
%

clc; clear ; close all;
%% Elliptic Lowpass filter transfer function and plots
[z, p, H0, B, A] = ellipap2(4, 1.411, 16.48);
syms sl Omega_L;
H_LPF =
(B(1,1)+B(1,2)*sl+sl^2*B(1,3))*(B(2,1)+B(2,2)*sl+sl^2*B(2,3))*(B(3,1
)+B(3,2)*sl+sl^2*B(3,3))/((A(1,1)+A(1,2)*sl+sl^2*A(1,3))*(A(2,1)+A(2
,2)*sl+sl^2*A(2,3))*(A(3,1)+A(3,2)*sl+sl^2*A(3,3)));
H_LPF_freq = subs(H_LPF, sl, 1i * Omega_L);
[ns, ds] = numden(H_LPF);
nsl = sym2poly(ns);
dsl = sym2poly(ds);
k = ds(1);
ns = nsl / k;
ds = dsl / k;
disp(nsl);
disp(dsl);
figure();
fplot(abs(H_LPF_freq));
hold on;
fplot(sl - sl - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(sl-sl+1, 'k-', 'Markersize', 10);
hold on;
fplot(sl - sl + 0.15, 'r-', 'Markersize', 10);
xline(1, 'magenta-');
hold on;
xline(1.1403,'magenta-');
axis([0 2 0 1.2]);
set(gca, 'XTick', [0, 1, 1.1403], 'xticklabel', {'0', '\Omega_{Lp} =
1','\Omega_{Ls} = 1.1403'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
daspect([1 1 1]);
title('Equivalent Elliptic Lowpass filter mangitude response');
xlabel('\Omega_L');
ylabel('|H(j \Omega_L)|');
 figure();
 fplot(angle(H_LPF_freq));
 xlabel('\Omega_L');
 ylabel('\angle H(j \Omega_L)');
 title('Equivalent Elliptic Lowpass filter phase response');
%% Analog lowpass to bandpass frequency transformation
syms s Omega;
Omega_0 = 0.9821;
B = 0.5264;
H_BPF = subs(H_LPF, sl, (s^2 + Omega_0^2) / (B * s));
[ns, ds] = numden(H_BPF);
ns = sym2poly(ns);
ds = sym2poly(ds);
k = ds(1);
ns = ns / k;
ds = ds / k;
disp(ns);
```

```
disp(ds);
H_BPF_freq = subs(H_BPF, s, 1i * Omega);
figure();
fplot(abs(H_BPF_freq));
set(gca, 'XTick', [0.7265, 0.7536, 1.2799, 1.3270], 'xticklabel',
{'\Omega_{s1}', '\Omega_{p1}', '\Omega_{p2}', '\Omega_{s2}'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
hold on;
xline(0.7265, 'magenta-');hold on;
xline(0.7536, 'magenta-');hold on;
xline(1.2799, 'magenta-');hold on;
xline(1.3270, 'magenta-');
hold on;
fplot(s - s - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(s-s+1, 'k-', 'Markersize', 10);
hold on;
fplot(s - s + 0.15, 'r-', 'Markersize', 10);
daspect([1 1 1]);
title('Elliptic Bandpass filter mangitude response');
xlabel('\Omega');
ylabel('|H(j \Omega)|');
axis([0 2 0 1.2]);
 figure();
 fplot(angle(H_BPF_freq));
 xlabel('\Omega');
 ylabel('\angle H(j \Omega)');
 title('Elliptic Bandpass filter phase response');
%% Analog to z bilinear transformation
syms z;
Hz = subs(H_BPF, s, (z-1)/(z+1));
[Nz, Dz] = numden(Hz);
Nz = sym2poly(Nz);
Dz = sym2poly(Dz);
k = Dz(1);
Nz = Nz / k;
Dz = Dz / k;
disp(Nz);
disp(Dz);
[H,f] = freqz(Nz,Dz,1024*1024, 540e3);
plot(f,abs(H));
axis([0 200e3 0 1.3]);
hold on;
yline(1.00, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.85, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.15, 'red--', 'LineWidth', 1.5);
hold on;
xline(111e3, 'magenta--', 'LineWidth', 1.5);
hold on;
xline(156e3, 'magenta--', 'LineWidth', 1.5);
xlabel('f in 10^4 Hz');
ylabel('|H(e^{j 2\pi f})|');
title('Magnitude Response of the Discrete Time Bandpass Filter');
```

```
set(gca, 'XTick', [108e3, 111e3, 156e3, 159e3], 'xticklabel',
{'f_{s1}', 'f_{p1}', 'f_{p2}', 'f_{s2}'});
set(gca, 'YTick', [0.15, 0.85, 1, 1.15], 'yticklabel', {'\delta_2 =
0.15', '1 - \delta_1 = 0.85', '1', '1 + \delta_1 = 1.15'});
fvtool(Nz, Dz, 'Analysis', 'Phase');
```

G. Elliptic Bandstop Filter (Matlab Code)

```
% Please execute this code section wise
% so that the values of the variables
% settle and are stable
%

clc; clear ; close all;
%
%% Elliptic Lowpass filter transfer function and plots
[z, p, H0, B, A] = ellipap2(3, 1.411, 16.48);
syms sl Omega_L;
H_LPF =
(B(1,1)+B(1,2)*sl+sl^2*B(1,3))*(B(2,1)+B(2,2)*sl+sl^2*B(2,3))/((A(1,
1)+A(1,2)*sl+sl^2*A(1,3))*(A(2,1)+A(2,2)*sl+sl^2*A(2,3)));
H_LPF_freq = subs(H_LPF, sl, 1i * Omega_L);
[ns, ds] = numden(H_LPF);
nsl = sym2poly(ns);
dsl = sym2poly(ds);
k = ds(1);
ns = ns / k;
ds = ds / k;
disp(nsl);
disp(dsl);
figure();
fplot(abs(H_LPF_freq));
hold on;
fplot(sl - sl - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(sl-sl+1, 'k-', 'Markersize', 10);
hold on;
fplot(sl - sl + 0.15, 'r-', 'Markersize', 10);
xline(1, 'magenta-');
hold on;
xline(1.2488,'magenta-');
axis([0 2 0 1.2]);
set(gca, 'XTick', [0, 1, 1.2488], 'xticklabel', {'0', '\Omega_{Lp} =
1','\Omega_{Ls} = 1.2488'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
daspect([1 1 1]);
title('Equivalent Elliptic Lowpass filter mangitude response');
xlabel('\Omega_L');
ylabel('|H(j \Omega_L)|');
 figure();
 fplot(angle(H_LPF_freq));
 xlabel('\Omega_L');
 ylabel('\angle H(j \Omega_L)');
```

```
 title('Equivalent Elliptic Lowpass filter phase response');
%% Analog lowpass to bandstop frequency transformation
syms s Omega;
B = 0.5009;
Omega_0 = 1.0081;
H_BSF = subs(H_LPF, sl, (B * s) / (s^2 + Omega_0^2) );
[ns, ds] = numden(H_BSF);
ns = sym2poly(ns);
ds = sym2poly(ds);
k = ds(1);
ns = ns / k;
ds = ds / k;
disp(ns);
disp(ds);
H_BSF_freq = subs(H_BSF, s, 1i * Omega);
figure();
fplot(abs(H_BSF_freq));
set(gca, 'XTick', [0.7883, 0.8273, 1.2283, 1.2892], 'xticklabel',
{'\Omega_{p1}', '\Omega_{s1}', '\Omega_{s2}', '\Omega_{p2}'});
set(gca, 'YTick', [0.15, 0.85, 1], 'yticklabel', {'\delta_2 = 0.15',
'1 - \delta_1 = 0.85', '1'});
hold on;
xline(0.7883, 'magenta-');hold on;
xline(0.8273, 'magenta-');hold on;
xline(1.2283, 'magenta-');hold on;
xline(1.2892, 'magenta-');
hold on;
fplot(s - s - 0.15 + 1, 'k-', 'Markersize', 10);
hold on;
fplot(s-s+1, 'k-', 'Markersize', 10);
hold on;
fplot(s -s + 0.15, 'r-', 'Markersize', 10);
daspect([1 1 1]);
title('Elliptic Bandstop filter mangitude response');
xlabel('\Omega');
ylabel('|H(j \Omega)|');
axis([0 2 0 1.2]);
figure();
 fplot(angle(H_BSF_freq));
 xlabel('\Omega');
 ylabel('\angle H(j \Omega)');
 title('Elliptic Bandstop filter phase response');
%% Analog to z bilinear transformation
syms z;
Hz = subs(H_BSF, s, (z-1)/(z+1));
[Nz, Dz] = numden(Hz);
Nz = sym2poly(Nz);
Dz = sym2poly(Dz);
k = Dz(1);
Nz = Nz / k;
Dz = Dz / k;
disp(Nz);
disp(Dz);
[H,f] = freqz(Nz,Dz,1024*1024, 400e3);
plot(f,abs(H));
axis([0 150e3 0 1.3]);
```

```
hold on;
yline(1, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.85, 'red--', 'LineWidth', 1.5);
hold on;
yline(0.15, 'red--', 'LineWidth', 1.5);
hold on;
xline(88e3, 'magenta--', 'LineWidth', 1.5);
hold on;
xline(113e3, 'magenta--', 'LineWidth', 1.5);
xlabel('f in 10^4 Hz');
ylabel('|H(e^{j 2\pi f})|');
title('Magnitude Response of the Discrete Time Bandstop Filter');
set(gca, 'XTick', [85e3, 88e3, 113e3, 116e3], 'xticklabel',
{'f_{p1}', 'f_{s1}', 'f_{s2}', 'f_{p2}'});
set(gca, 'YTick', [0.15, 0.85, 1, 1.15], 'yticklabel', {'\delta_2 =
0.15', '1 - \delta_1 = 0.85', '1', '1 + \delta_1 = 1.15'});
fvtool(Nz, Dz, 'Analysis', 'Phase');
```

H. Helper Function for Elliptic Filter Design (Matlab Code)

```
%Helper Function for Elliptic Filter Design
function [z,p,H0,B,A] = ellipap2(N,Ap,As)
if nargin==0, help ellipap2; return; end
Gp = 10^(-Ap/20); % passband gain
ep = sqrt(10^(Ap/10) - 1); % ripple factors
es = sqrt(10^(As/10) - 1);
k1 = ep/es;
k = ellipdeg(N,k1); % solve degree equation
L = floor(N/2); r = mod(N,2); % L is the number of second-order
sections
i = (1:L)'; ui = (2*i-1)/N; zeta_i = cde(ui,k); % zeros of elliptic
rational function
z = j./(k*zeta_i); % filter zeros = poles of elliptic rational
function
v0 = -j*asne(j/ep, k1)/N; % solution of sn(jv0NK1, k1) = j/?p
p = j*cde(ui-j*v0, k); % filter poles
p0 = j*sne(j*v0, k); % first-order pole, needed when N is odd
B = [ones(L,1), -2*real(1./z), abs(1./z).^2]; % second-order
numerator sections
A = [ones(L,1), -2*real(1./p), abs(1./p).^2]; % second-order
denominator sections
if r==0, % prepend first-order sections
B = [Gp, 0, 0; B]; A = [1, 0, 0; A];
else
B = [1, 0, 0; B]; A = [1, -real(1/p0), 0; A];
end
z = cplxpair([z; conj(z)]); % append conjugate zeros
p = cplxpair([p; conj(p)]); % append conjugate poles
if r==1, p = [p; p0]; end % append first-order pole when N is odd
H0 = Gp^(1-r); % dc gain
end
```

I. References

1. Prof. V.M.Gadre's NPTEL Video Lectures
2. Lecture Notes on Elliptic Filter Design by Sophocles J. Orfanidis