

# Filter Design

PALAVELLI SRIJA  
EE23BTECH11045

## 1 Introduction

We are supposed to design the equivalent FIR and IIR filter realizations for given filter number. This is a bandpass filter whose specifications are available below.

## 2 Filter Specifications

### 2.1 The Digital Filter

1. Passband: The passband is from  $\{4 + 0.6(j)\}$  kHz to  $\{4 + 0.6(j+2)\}$  kHz.  
where

$$j = (r - 11000) \mod \sigma \quad (1)$$

where  $\sigma$  is sum of digits of roll number and  $r$  is roll number.

$$r = 11045 \quad (2)$$

$$\sigma = 11 \quad (3)$$

$$j = 1 \quad (4)$$

substituting  $j = 1$  gives the passband range for our bandpass filter as 4.6 kHz - 5.8 kHz. Hence, the un-normalized discrete time filter passband frequencies are  $F_{p1} = 4.6$  kHz and  $F_{p2} = 5.8$  kHz.

The corresponding normalized digital filter passband frequencies are

$$\omega_{p1} = 2\pi \frac{F_{p1}}{F_s} = 0.192\pi \quad (5)$$

$$\omega_{p2} = 2\pi \frac{F_{p2}}{F_s} = 0.242\pi \quad (6)$$

2. Tolerances: The passband ( $\delta_1$ ) and stopband ( $\delta_2$ ) tolerances are given to be equal, so we let  $\delta_1 = \delta_2 = \delta = 0.15$ .
3. Stopband: The transition band for bandpass filters is  $\Delta F = 0.3$  kHz on either side of the passband.

$$F_{s1} = 4.6 - 0.3 = 4.3\text{KHz} \quad (7)$$

$$F_{s2} = 5.8 + 0.3 = 6.1\text{KHz} \quad (8)$$

$$\omega_{s1} = 2\pi \frac{F_{s1}}{F_s} = 0.179\pi \quad (9)$$

$$\omega_{s2} = 2\pi \frac{F_{s2}}{F_s} = 0.254\pi \quad (10)$$

$$(11)$$

## 2.2 The Analog filter

In the bilinear transform, the analog filter frequency ( $\Omega$ ) is related to the corresponding digital filter frequency( $\omega$ ) :

$$\Omega = \tan \frac{\omega}{2} \quad (12)$$

Using this relation, we obtain the analog passband and stopband frequencies as:  $\Omega_{p1} = 0.3111$ ,  $\Omega_{p2} = 0.3995$  and  $\Omega_{s1} = 0.2888$ ,  $\Omega_{s2} = 0.4215$  respectively.

## 3 The IIR Filter Design

We are supposed to design filters whose stopband is monotonic and passband equiripple. Hence, we use the Chebyshev approximation to design our bandpass IIR filter.

### 3.1 The Analog Filter

1. Low Pass Filter Specifications: Let  $H_{a,BP}(j\Omega)$  be the desired analog bandpass filter, with the specifications provided in Section 2.2, and  $H_{a,LP}(j\Omega_L)$  be the equivalent low pass filter, then

$$\Omega_L = \frac{\Omega^2 - \Omega_0^2}{B\Omega} \quad (13)$$

where  $\Omega_0 = \sqrt{\Omega_{p1}\Omega_{p2}} = 0.352$  and  $B = \Omega_{p2} - \Omega_{p1} = 0.0884$ .

Substituting  $\Omega_{s1}$  and  $\Omega_{s2}$  in (13) we obtain the stopband edges of lowpass filter

$$\Omega_{Ls1} = \frac{\Omega_{s1}^2 - \Omega_0^2}{B\Omega_{s1}} = -1.586 \quad (14)$$

$$\Omega_{Ls2} = \frac{\Omega_{s2}^2 - \Omega_0^2}{B\Omega_{s2}} = 1.442 \quad (15)$$

And we choose the minimum of these two stopband edges

$$\Omega_{Ls} = \min(|\Omega_{Ls1}|, |\Omega_{Ls2}|) = 1.442. \quad (16)$$

2. The Low Pass Chebyshev Filter Paramters: The magnitude of frequency response of the low pass filter is given by

$$|H_{a,LP}(j\Omega_L)|^2 = \frac{1}{1 + \epsilon^2 c_N^2(\Omega_L/\Omega_{Lp})} \quad (17)$$

The passband edge of the low pass filter is chosen as  $\Omega_{Lp} = 1$ . Therefore ,

$$|H_{a,LP}(j\Omega_L)|^2 = \frac{1}{1 + \epsilon^2 c_N^2(\Omega_L)} \quad (18)$$

Here  $c_N$  denote the chebyshev polynomials for a particular order  $N$  of the filter.

$$c_N(x) = \cosh(N \cosh^{-1} x), x = \Omega_L \quad (19)$$

$$c_0(x) = 1 \quad (20)$$

$$c_1(x) = x \quad (21)$$

There exists a recurssive relation from which all the polynomials can be found out.

$$c_{N+2} = 2xc_{N+1} - c_N \quad (22)$$

Imposing the band restrictions on (17)

$$|H_{a,LP}(j\Omega_L)|^2 < \delta_2 \text{ for } \Omega_L = \Omega_{Ls} \quad (23)$$

$$1 - \delta_1 < |H_{a,LP}(j\Omega_L)|^2 < 1 \text{ for } \Omega_L = \Omega_{Lp} \quad (24)$$

$$(25)$$

we obtain :

$$\frac{\sqrt{D_2}}{c_N(\Omega_{Ls})} \leq \epsilon \leq \sqrt{D_1},$$

$$N \geq \left\lceil \frac{\cosh^{-1} \sqrt{D_2/D_1}}{\cosh^{-1} \Omega_{Ls}} \right\rceil, \quad (26)$$

where  $D_1 = \frac{1}{(1-\delta)^2} - 1$  and  $D_2 = \frac{1}{\delta^2} - 1$  and  $\lceil . \rceil$  is known as the ceiling operator .

Parameter	Value
$D_1$	0.384
$D_2$	43.44
$N$	4
$c_4(x)$	$8x^4 - 8x^2 + 1$

Table 1: Parameter Table

we get  $N \geq 4$  and  $0.337 \leq \epsilon \leq 0.61$

The below code plots (17) for different values of  $\epsilon$ .

<https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/a.py>

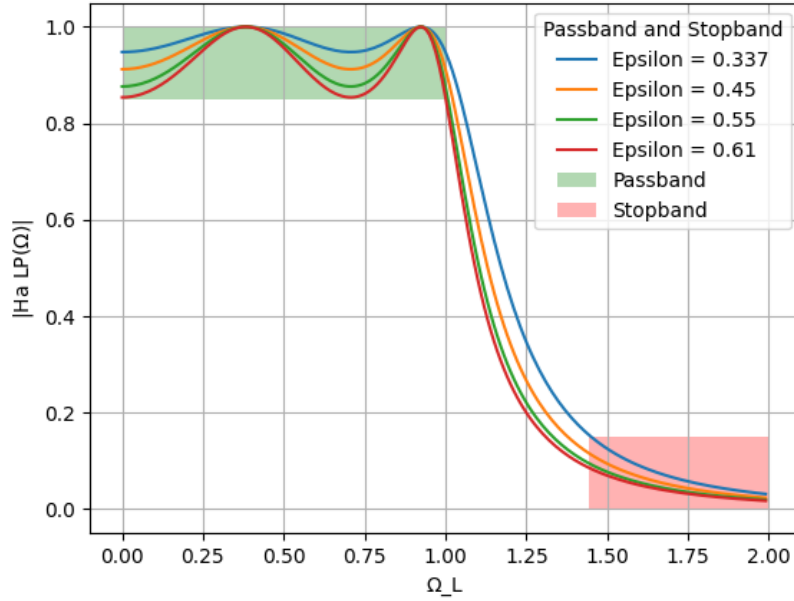


Figure 1: The Analog Low-Pass Frequency Response for  $0.337 \leq \epsilon \leq 0.61$

In Fig. 1 we can observe the equiripple behaviour in passband and monotonic behaviour in stopband. As the value of  $\epsilon$  increases the value of  $|H_{a,LP}(j\Omega_L)|$  decreases.

3. The Low Pass Chebyshev Filter: The next step in design is to find an expression for magnitude response in  $s$  domain.

Using  $s = j\Omega$  or in this case  $s_L = j\Omega_L$  we obtain:

$$|H_{a,LP}(j\Omega_L)|^2 = \frac{1}{1 + \epsilon^2 c_N^2\left(\frac{s_L}{j}\right)} \quad (27)$$

To find poles equate the denominator to zero:

$$1 + \epsilon^2 c_N^2\left(\frac{s_L}{j}\right) = 0 \quad \text{where} \quad c_N(x) = \cos\left(N \cos^{-1}(x)\right) \quad (28)$$

On solving (28) we obtain poles :

$$s_k = -\Omega_{Lp} \sin(A_k) \sinh(B_k) - j\Omega_{Lp} \cos(A_k) \cosh(B_k) \quad (29)$$

where  $k$  is the index of the pole and

$$A_k = (2k + 1) \frac{\pi}{2N} \quad (30)$$

$$B_k = \frac{1}{N} \sinh^{-1} \left( \frac{1}{\epsilon} \right) \quad (31)$$

The below code computes the values of  $s_k$  and stores it in a text file.

[https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/sk\\_gen.c](https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/sk_gen.c)

The poles obtained are formulated in the table below.

<i>Pole</i>	<i>Value</i>
$s_1$	$0.1621 + j1.0033$
$s_2$	$0.3913 + j0.4156$
$s_3$	$0.3913 - j0.4156$
$s_4$	$0.1621 - j1.0033$
$s_5$	$-0.1621 + j1.0033$
$s_6$	$-0.3913 + j0.4156$
$s_7$	$-0.3916 - j0.4156$
$s_8$	$-0.1621 - j1.0033$

Table 2: Values of  $s_k$

The below code plots the pole-zero plot.

[https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/pole\\_zero\\_plt.py](https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/pole_zero_plt.py)



Figure 2: The Pole zero plot and all the poles lie on an ellipse. The left and right poles have been identified as shown.

The poles in the left half of the plane are considered in the design as we intend to design a stable system.

Therefore the magnitude response is written as :-

$$H_{a,LP}(s_L) = \frac{G_{LP}}{(s_L - s_5)(s_L - s_6)(s_L - s_7)(s_L - s_8)} \quad (32)$$

where  $G_{LP}$  is the gain of the Low pass filter. Refer to Table 2 for  $s_k$  values.

We know that from (17):-

$$|H_{a,LP}(s_L)| = \frac{1}{\sqrt{1 + \epsilon^2}} \text{ at } \Omega_L = 1 \implies s_L = j \quad (33)$$

The value of  $G_{LP}$  can be found using the following code after appropriate substitution

```
https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/
codes/gain_LP.py
```

Substituting respective values in (33) we get  $G_{LP} = 0.3125$  Simplifications for equation (35) and (43) are done using the following code

[https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/exp\\_calc.py](https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/exp_calc.py)

$$H_{a,LP}(s_L) = \frac{0.3125}{(s_L - s_5)(s_L - s_6)(s_L - s_7)(s_L - s_8)} \quad (34)$$

$$= \frac{0.3125}{s_L^4 + 1.1068s_L^3 + 1.612445s_L^2 + 0.91397s_L + 0.336555} \quad (35)$$

The figure to compare design and specification is produced using the following code

<https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/b.py>

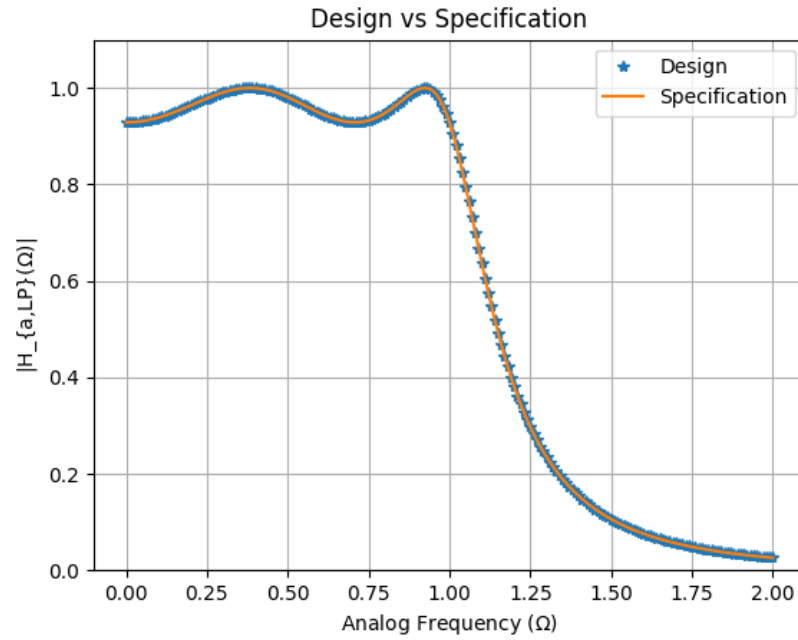


Figure 3: Design vs Specification corresponding to (35) and (18)

4. The Band Pass Chebyshev Filter: After verifying design with the required specifications the next step in design is to jump to required type of filter using fre-

quency transformation.

$$s_L = \frac{s^2 + \Omega_0^2}{Bs} \quad (36)$$

$$H_{a,BP}(s) = G_{BP} H_{a,LP}(s_L) \Big|_{s_L = \frac{s^2 + \Omega_0^2}{Bs}}, \quad (37)$$

As there is one to one correspondence between the filters so  $\Omega = \Omega_{p1}$  should correspond to  $\Omega_{LP}$

$$s = j\Omega_{p1} \quad (38)$$

$$s_L = \frac{(j\Omega_{p1})^2 + \Omega_0^2}{B(j\Omega_{p1})} \quad (39)$$

$$|H_{a,BP}(j\Omega_{p1})| = 1 \quad (40)$$

$$G_{BP} |H_{a,LP}(s_L)| = 1 \quad (41)$$

Substituting (39) in (41) we obtain Gain of required bass pass filter:

$$G_{BP} = 1.048 \quad (42)$$

Thus the response in s domain

$$H_{a,BP}(s) = \frac{1.908 \times 10^{-5} s^4}{s^8 + 0.0978s^7 + 0.5082s^6 + 0.0375s^5 + 0.0952s^4 + 0.00458s^3 + 0.0078s^2 + 0.000186s + 0.00024} \quad (43)$$

The expressions in the s-domain and gain factors are computed by writing the following Python code.

<https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/g.py>

In Figure 3, we plot  $|H_{a,BP}(j\Omega)|$  as a function of  $\Omega$  for both positive as well as negative frequencies. We find that the passband and stopband frequencies in the figure match well with those obtained analytically through the bilinear transformation.

The below code plots the Band Pass Filter plot

<https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/c.py>



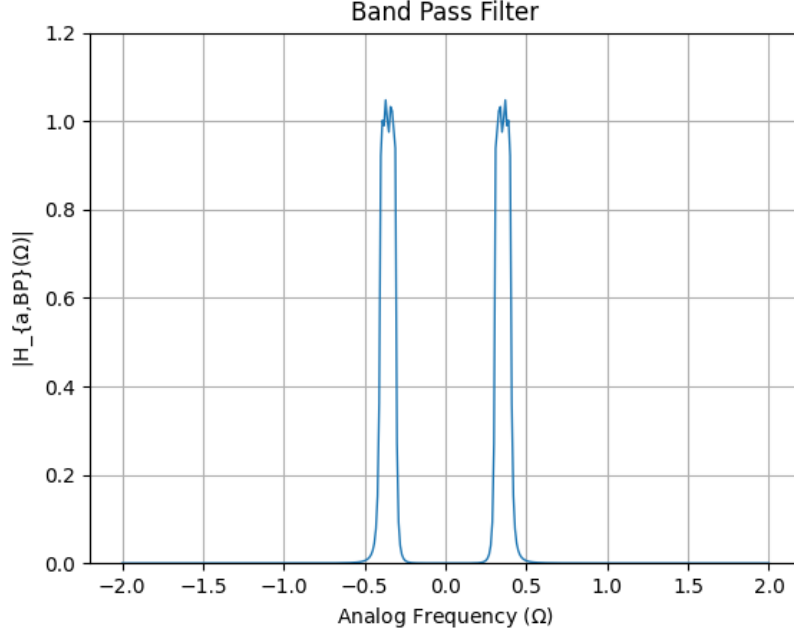


Figure 4: The Analog Bandpass Magnitude Response from (43). The filter design specifications are satisfied

### 3.2 The Digital Filter

From the bilinear transformation, we obtain the digital bandpass filter from the corresponding analog filter as

$$H_{d,BP}(z) = GH_{a,BP}(s) \Big|_{s=\frac{1-z^{-1}}{1+z^{-1}}} \quad (44)$$

Substituting  $s = \frac{1-z^{-1}}{1+z^{-1}}$  in (43) and calculating expression using a python code we get :

$$H_{d,BP}(z) = \frac{G(1 - 4z^{-2} + 6z^{-4} - 4z^{-6} + z^{-8})}{1.7508 - 10.6494 * z^{-1} + 30.978z^{-2} - 55.158z^{-3} + 65.43z^{-4} - 52.814z^{-5} + 28.4z^{-6} - 9.349z^{-7} + 1.472z^{-8}} \quad (45)$$

where  $G = 1.908 \times 10^{-5}$

[https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/filter\\_in\\_z.py](https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/filter_in_z.py)

The below code plots the Digital Band Pass Filter plot

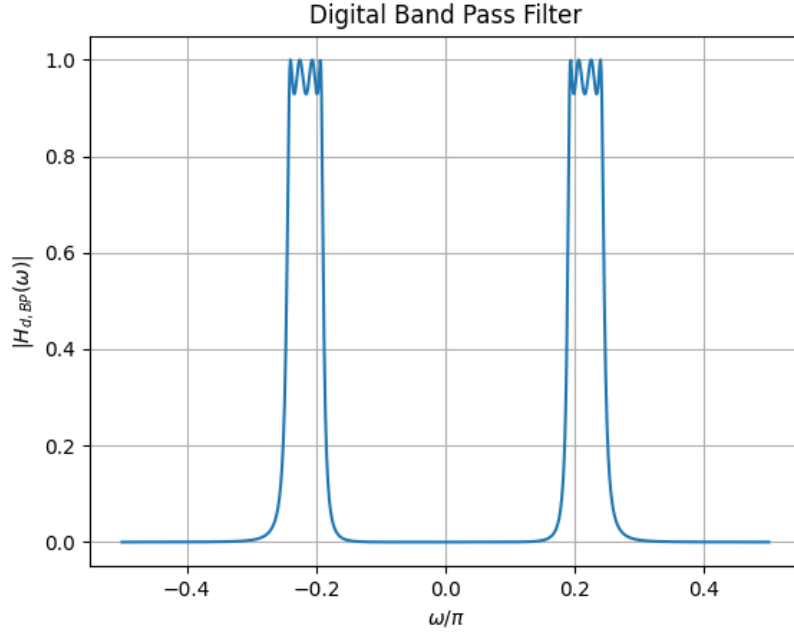


Figure 5: Digital Specifications are met. Passband and stopband frequencies are same

## 4 The FIR Filter

We design the FIR filter by first obtaining the (non-causal) lowpass equivalent using the Kaiser window and then converting it to a causal bandpass filter.

### 4.1 The Equivalent Lowpass Filter

The lowpass filter has a passband frequency  $\omega_l$  and transition band  $\Delta\omega = 2\pi\frac{\Delta F}{F_s} = 0.0125\pi$ . The stopband tolerance is  $\delta = 0.15$ . The cutoff-frequency is given by :

$$\omega_l = \frac{B}{2} \quad (46)$$

$$= 0.025\pi \quad (47)$$

The following python code is used to plot the frequency and the impulse response of an ideal Low Pass Filter.

[https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/FIR\\_1.py](https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/FIR_1.py)

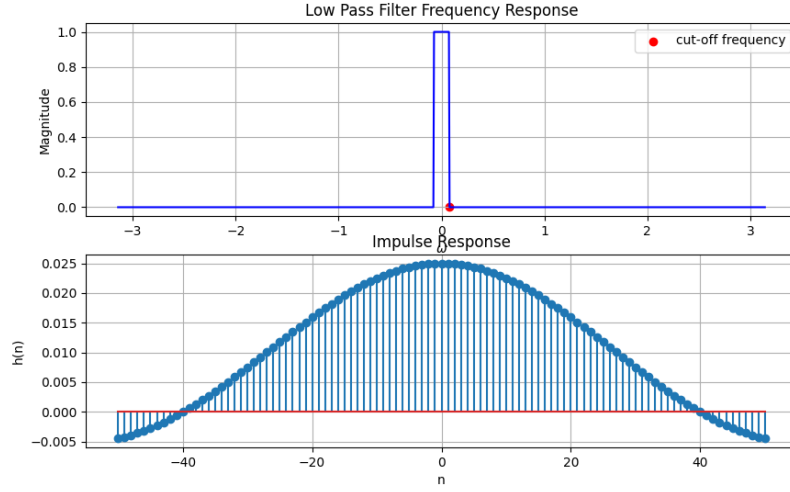


Figure 6: Frequency response and impulse response of an ideal Low Pass Filter

The impulse response of ideal Low Pass Filter is given by :

$$h(n) = \begin{cases} \frac{w_l}{\pi}, & \text{if } n = 0 \\ \frac{\sin(w_l n)}{n\pi}, & \text{if } n \neq 0 \end{cases} \quad (48)$$

From (48) we conclude that  $h(n)$  for an ideal Low Pass Filter is not causal and can neither be made causal by introducing a finite delay. And  $h(n)$  do not converge and hence the system is unstable.

## 4.2 The Kaiser Window

Therefore we move on windowing the impulse response. A window function is chosen and multiplied. The Kaiser window is defined as

$$w(n) = \begin{cases} \frac{I_0 \left[ \beta N \sqrt{1 - \left( \frac{n}{N} \right)^2} \right]}{I_0(\beta N)}, & -N \leq n \leq N, \quad \beta > 0 \\ 0, & \text{otherwise,} \end{cases}$$

1. N is chosen according to

$$N \geq \frac{A - 8}{4.57 \Delta \omega}, \quad (49)$$

where  $A = -20 \log_{10} \delta$ . Substituting the appropriate values from the design specifications, we obtain  $A = 16.4782$  and  $N \geq 48$ .

2.  $\beta$  is chosen according to

$$\beta N = \begin{cases} 0.1102(A - 8.7) & A > 50 \\ 0.5849(A - 21)^{0.4} + 0.07886(A - 21) & 21 \leq A \leq 50 \\ 0 & A < 21 \end{cases} \quad (50)$$

The window function is defined as :

$$w(n) = \begin{cases} 1, & \text{for } -48 \leq n \leq 48 \\ 0, & \text{otherwise} \end{cases} \quad (51)$$

Therefore the desired impulse response is :

$$h_{lp} = h_n w_n \quad (52)$$

$$h(n) = \begin{cases} \frac{\sin(\omega_p n)}{n\pi}, & \text{for } -48 \leq n \leq 48 \\ 0 & \text{otherwise} \end{cases} \quad (53)$$

The code to plot the magnitude response of the FIR Low Pass Filter is as follows

```
https://github.com/Srija04092006/EE1205/blob/main/Filter%20Design/codes/d.py
```

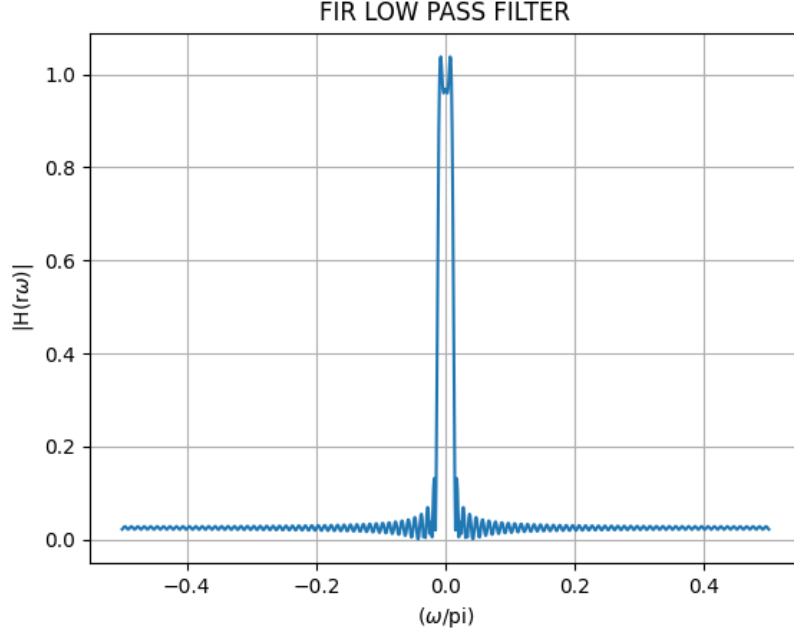


Figure 7: Magnitude Response of Low Pass Filter after using Kaiser Window

### 4.3 The Equivalent Band Pass Filter

A Band-Pass Filter (BPF) can be obtained by subtracting the magnitude response of a Low-Pass Filter (LPF) with cutoff frequency  $\omega_{p1}$  from another LPF magnitude response with cutoff frequency  $\omega_{p2}$ .

$$h_{BP}(n) = \begin{cases} \frac{\sin(\omega_{p2}n)}{n\pi} - \frac{\sin(\omega_{p1}n)}{n\pi}, & \text{for } n \neq 0 \\ \frac{\omega_{p2} - \omega_{p1}}{\pi}, & \text{for } n = 0 \end{cases} \quad (54)$$

$$\frac{\sin(\omega_{p2}n)}{n\pi} - \frac{\sin(\omega_{p1}n)}{n\pi} = 2 \cos\left(\frac{\omega_{p2}n + \omega_{p1}n}{2}\right) \sin\left(\frac{\omega_{p2}n - \omega_{p1}n}{2}\right) \quad (55)$$

$$= \frac{2 \cos(0.217n\pi) \sin(0.025n\pi)}{n\pi} \quad (56)$$

Multiplying by window function we get :

$$h_{BP}(n) = \begin{cases} \frac{2 \cos(0.217n\pi) \sin(0.025n\pi)}{n\pi}, & \text{for } -48 \leq n \leq 48 \\ 0 & \text{otherwise} \end{cases} \quad (57)$$

The code to plot the same is as follows

[https://github.com/HarryNyquist/EE1205/blob/main/Filter\\_Design/codes/e.py](https://github.com/HarryNyquist/EE1205/blob/main/Filter_Design/codes/e.py)

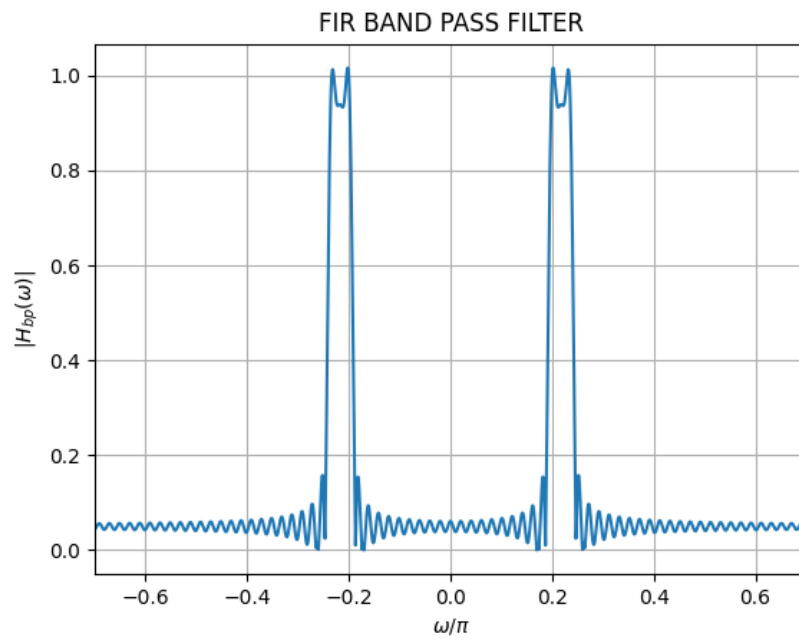


Figure 8: Magnitude Response of Band Pass Filter after using Kaiser Window