Digital Signal Processing (DSP)

Designing lowpass and bandpass filters (Parks-McCellan and Kaiser window method)

by Zavareh Bozorgasl

Report of the results of homework 4 of DSP Course

This homework is about designing digital filters with finite-duration impulse response (all-zero, or FIR filters). In the report, we mostly use LPF and BPF for lowpass filter and bandpass filter, respectively.

In the first part, we are supposed to design lowpass and bandpass filters with the given specifications in the problem statement. As it is mentioned in the statement, there is need to do the design for each filter by the Parks-McCellan Algorithm, which is usually called, the optimal method and also the Kaiser window method. These two were carefully chosen, because the Kaiser method is the best candidate of the window method and, for a given specification, it is nearly optimal and gives us the minimum mainlobe width among different window methods, and also this method is simple and can be used generally.

Table 1, shows different window methods [1].

Table 1. comparison of different window method (and equivalence of the Kaiser for each one)

Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log_{10}\delta$ (dB)	Equivalent Kaiser Window, β	Transition Width of Equivalent Kaiser Window
Rectangular	−13	$4\pi/(M+1)$	−21	0	$1.81\pi/M$
Bartlett	−25	$8\pi/M$	−25	1.33	$2.37\pi/M$
Hanning	−31	$8\pi/M$	−44	3.86	$5.01\pi/M$
Hamming	−41	$8\pi/M$	−53	4.86	$6.27\pi/M$
Blackman	−57	$12\pi/M$	−74	7.04	$9.19\pi/M$

M is the window length, and σ is for the amount of allowed ripple, β is a parameter of the Kaiser window method.

Moreover, the Parks-McCellan is known as an optimal method of the equiripple filter design. To shorten the report, I do not add the process of their design and their formulas. The comments in the piece of Matlab code are self-explanatory. Alongside presenting some results, I will give my observations.

Figure 1 shows the result of the magnitude and phase of the filter designed by the Kaiser method.

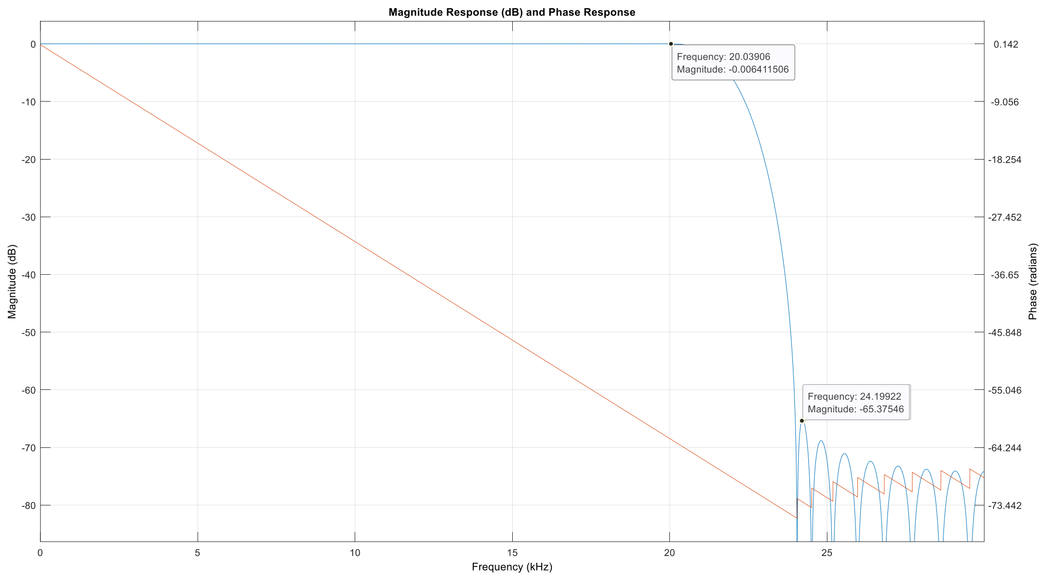


Fig. 1 LPF designed by Kaiser window

As we expected for FIR filters, the phase response is linear in the passband, and this is the most important region for us. Outside of this region, as it has a very low magnitude (more than 65 dB), we can ignore the nonlinearity of that region. In other words, it is typically sufficient to have linear phase only in the pass band of a filter, and it is common to refer to such filters as having linear phase, even if their entire phase response is technically non-linear. The lack of phase/delay distortion can be a critical advantage of FIR filters over IIR.

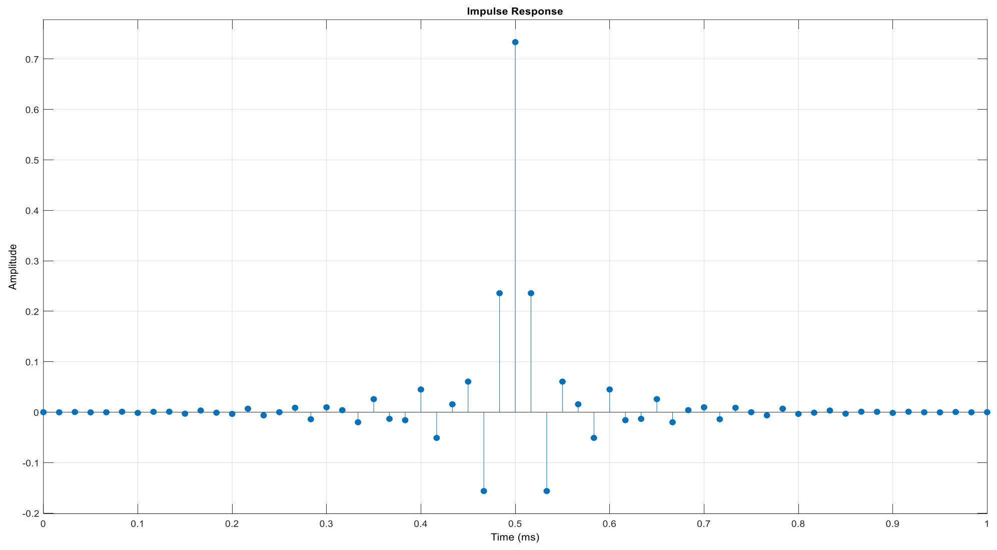
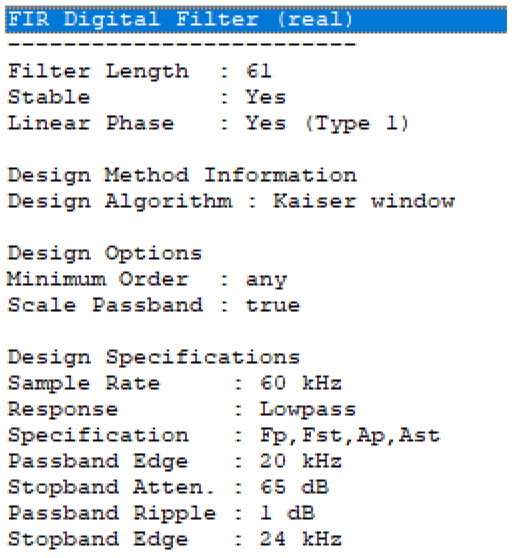


Fig. 2 Impulse response of the LPF designed by the Kaiser window

FIR filters are usually designed to be linear-phase (but they don’t have to be). A FIR filter is linear-phase if (and only if) its coefficients are symmetrical around the center coefficient, that is, the first coefficient is the same as the last; the second is the same as the next-to-last, etc. (A linear-phase FIR filter having an odd number of coefficients will have a single coefficient in the center which has no mate). A lowpass FIR filter has its largest-magnitude coefficients in the center of the impulse response. We see this in the impulse response of the LPF filter, in Figure 2, designed by the Kaiser window method. Figure 3 presents the information of the filter.



```
FIR Digital Filter (real)
-------------------------
Filter Length  : 61
Stable         : Yes
Linear Phase   : Yes (Type 1)

Design Method Information
Design Algorithm : Kaiser window

Design Options
Minimum Order  : any
Scale Passband : true

Design Specifications
Sample Rate      : 60 kHz
Response         : Lowpass
Specification    : Fp,Fst,Ap,Ast
Passband Edge    : 20 kHz
Stopband Atten.  : 65 dB
Passband Ripple  : 1 dB
Stopband Edge    : 24 kHz
```

Fig. 3 Information of the LPF designed by the Kaiser window

We can see all the poles and zeros both visually (by fvtool) and also by the function below in Matlab

[vz,vp,vk] = zplane(d) returns the zeros (vector vz), poles (vector vp), and gain (scalar vk) corresponding to the digital filter d.

It has 59 zeros near the origin of the real-imaginary plane and 60 poles in the origin, and 1 zero in the infinity. As we expected, the FIR filter does not have any pole in the plan (there might be several ones in the origin or at the infinity).

If we design a filter with Parks-McCellan Algorithm, then we have Fig. 4. For better illustration, we compare that we the filter we earlier designed by the Kaiser window method. The estimated order for Parks-McCellan was 54 (i.e., **n = 54**) , and it does not satisfy our requirements in the stopband. Hence, we increase that to 55 , i.e., (**n+1**); which results in Fig. 5.

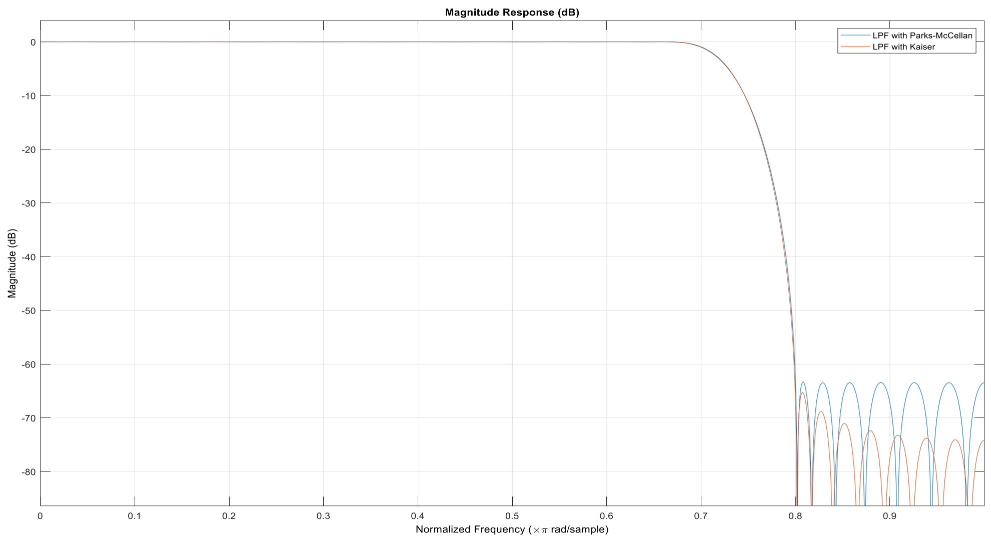


Fig. 4. Comparison of the LPFs designed by the Parks-McCellan and the Kaiser window (n = 54).

We see that this filter does not give us the desired response in the passband; hence, we increase the number, **n**, to **n+1** (the filter order), and plot it again:

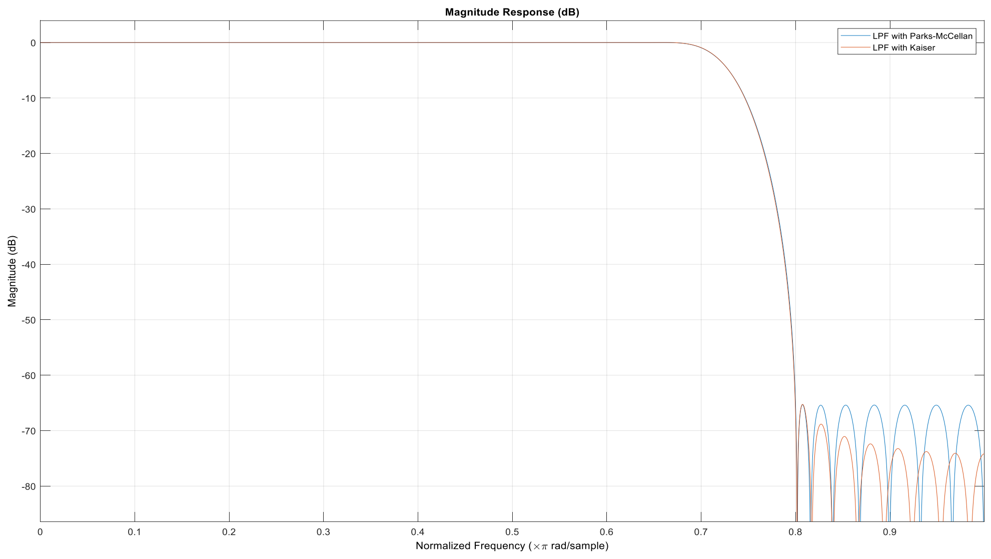
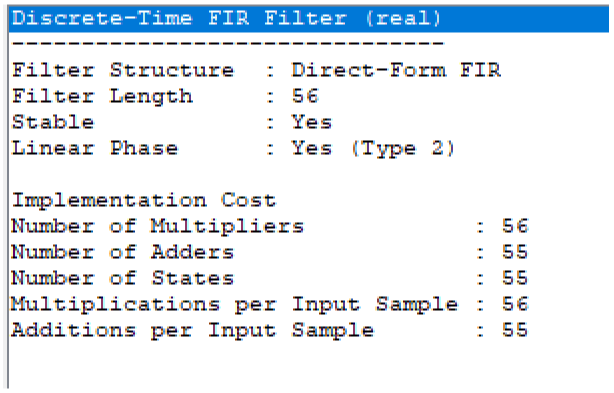


Fig. 5. Comparison of the LPFs designed by the Parks-McCellan and the Kaiser window (n = 54 + 1).

Indeed, the LPF designed with the Parks-McCellan has the information given in Figure 6.



```
Discrete-Time FIR Filter (real)
-------------------------------
Filter Structure    : Direct-Form FIR
Filter Length       : 56
Stable              : Yes
Linear Phase        : Yes (Type 2)

Implementation Cost
Number of Multipliers              : 56
Number of Adders                   : 55
Number of States                   : 55
Multiplications per Input Sample : 56
Additions per Input Sample         : 55
```

Fig. 5. Information of the LPFs designed by the Parks-McCellan.

And the LPF designed by the Kaiser method has the following impulse response as in Figure 6.

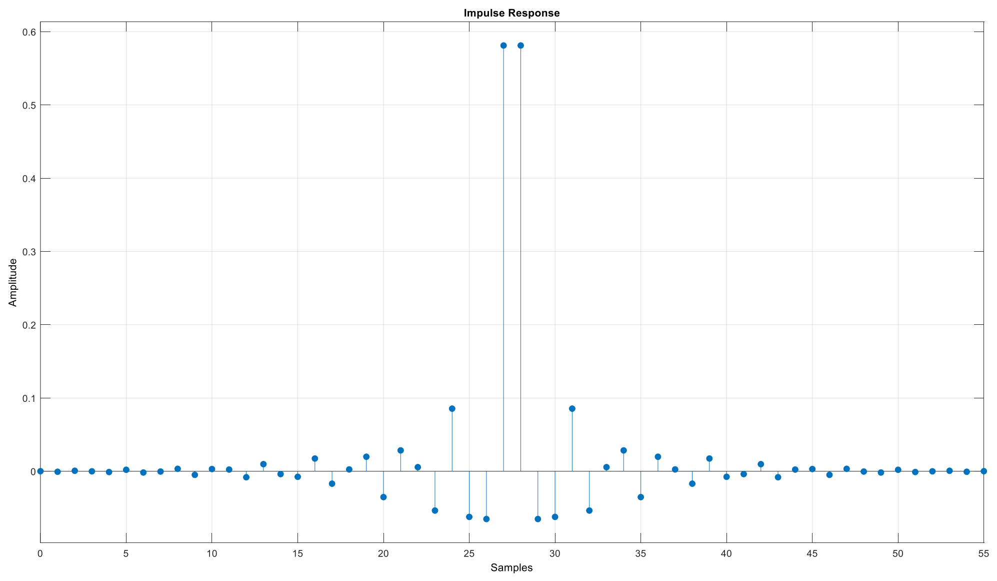


Fig. 6. Impulse response of the LPFs designed by the Parks-McCellan.

The Filter designed by Kaiser has the passband ripples as in Figure 7.

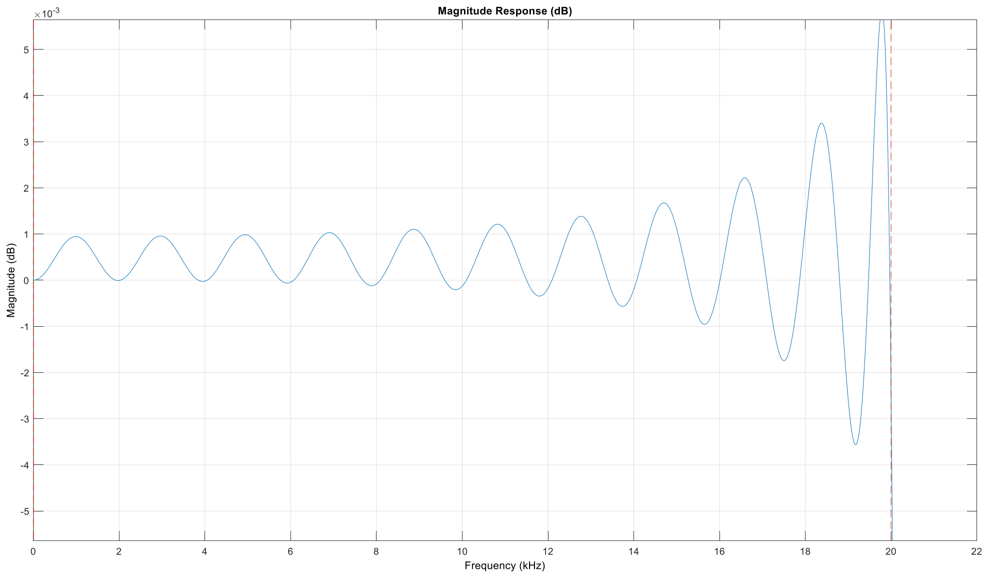


Fig. 7. Passband of the LPFs designed by the Parks-McCellan.

Which we can see it by the function zoom (in the code). And Figure 8 is the zooming on the passband of the filter which is designed with Parks-McCellan method. Comparison of Figure 7 and Figure 8 shows a disadvantage of the Kaiser window method. Indeed, the Kaiser window method is not equiripple in both the passband and the stopband. In other words, the Kaiser window method cannot efficiently use all the degree of freedom for the design. We see that the filter which is designed by the Parks-McCellan method is equiripple in both the passband and the stopband.

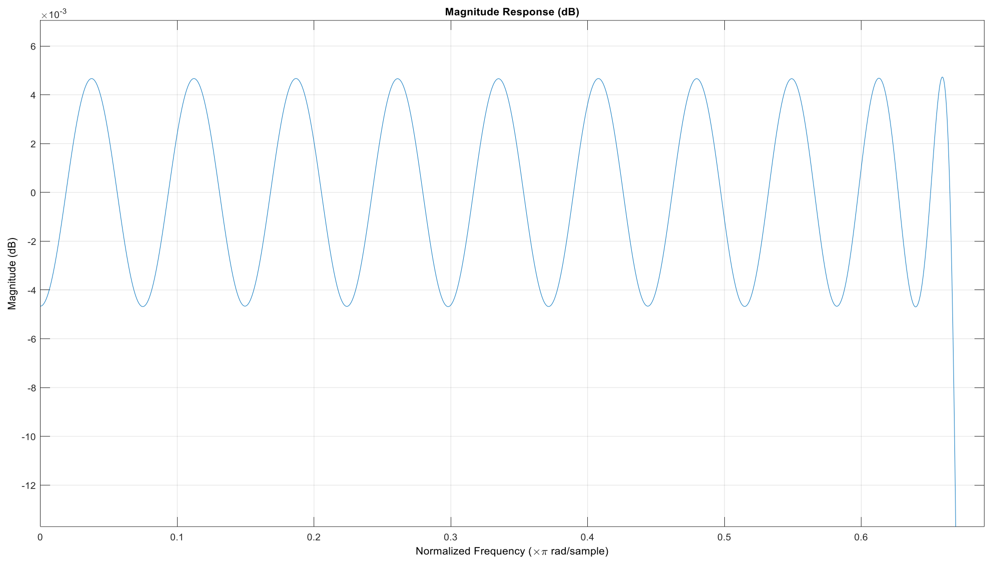


Fig. 8. Passband of the LPFs designed by the Parks-McCellan.

Figure 9 shows the pole-zero plot of the LPF designed by the Parks-McCellan method. As all the coefficients are real, all the zeros must be real or otherwise must be conjugate symmetric.

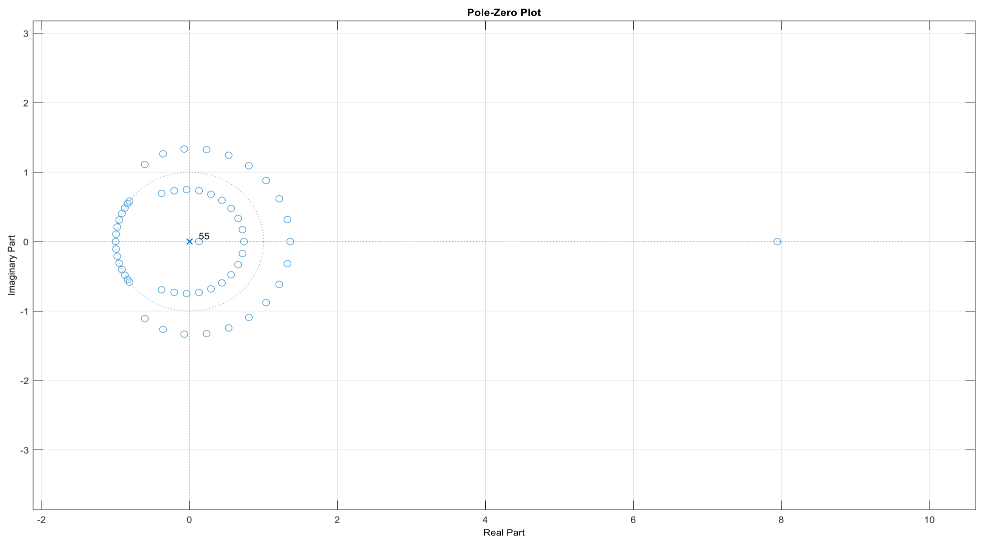


Figure 9. Pole-zero plot of the LPF designed by the Parks-McCellan method.

The zeros are complex-valued roots. The region of convergence (ROC) z ≠ 0 includes the unit circle, the system is Bounded Input Bounded Output (BIBO) stable. Indeed, FIR filters are always BIBO stable.

The filter designed by the Kaiser method is type 1 and the one designed by the Parks-McCellan is type 2. Type 1 filters are symmetric sequence of odd length (here, length of 61) and type 2 filters are symmetric sequence of even length (here, length of 56). There are several points about these filters which we don't repeat them here. For example, type 2 filter cannot be used as highpass filters.

As both systems almost satisfy the specification of the given LPF, but the LPF designed by the Parks-McCellan needs less filter taps (number of coefficients), which in turn leads to less cost of implementation and also less delay (we study group delay later), we choose the LPF designed by this optimal method.

Bandpass Filter design

As much of the explanation are similar to LPF design, we shorten the discussion to some extent. Figure 10 shows the comparison of the LPFs designed by the Parks-McCellan and the Kaiser window (**n + 4**).

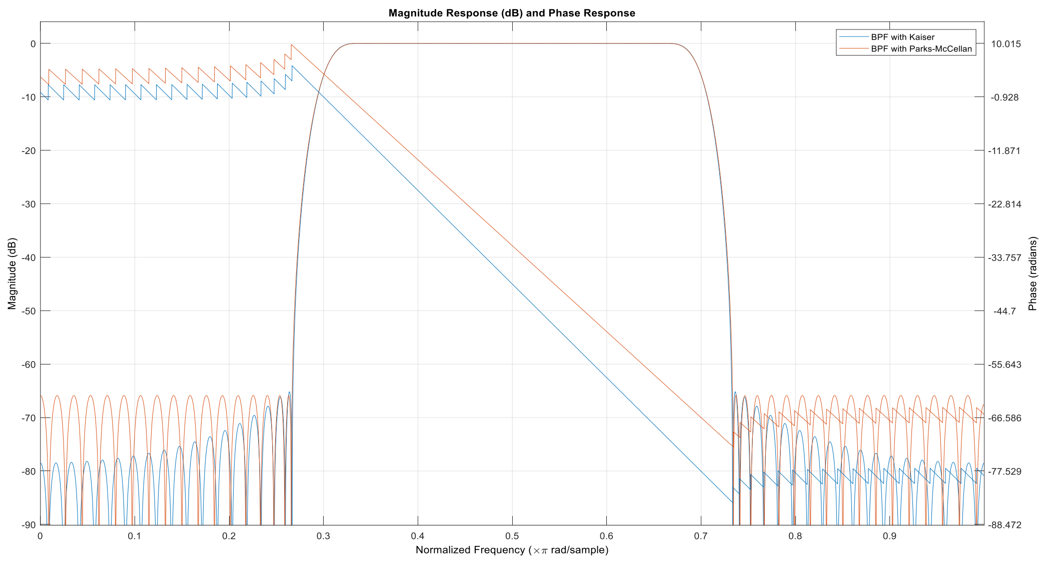
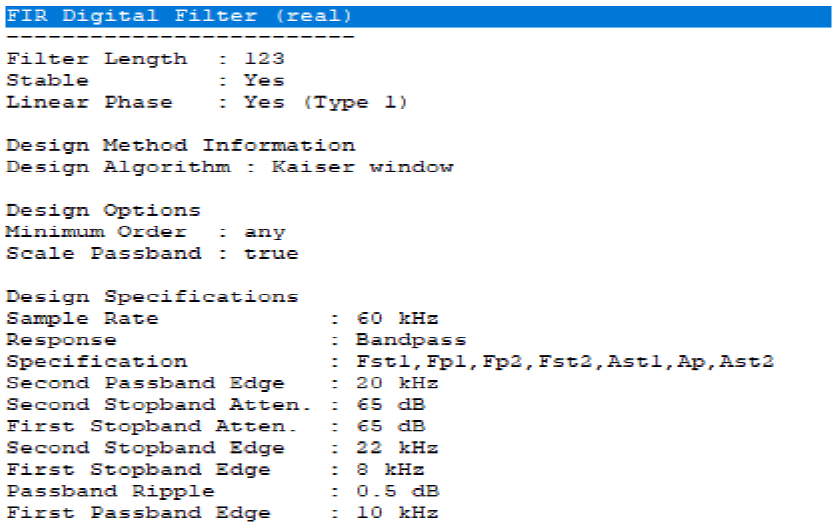


Fig. 4. Comparison of the LPFs designed by the Parks-McCellan and the Kaiser window (n + 4).

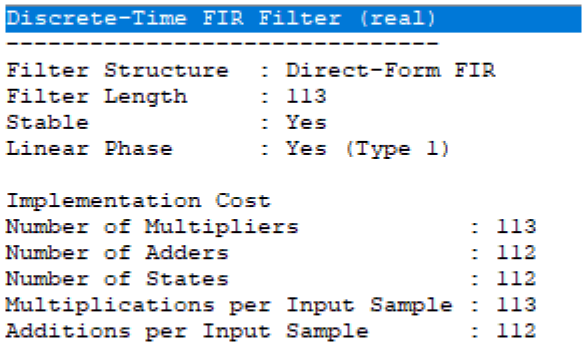
By increasing **n** in firpm to **n+4** we achieve the similar response as the design with the Kaiser method as well as with the fewer number of taps. In Figures 11 and 12, we see the information of both the Kaiser and Park-McCellan design methods.



```
FIR Digital Filter (real)
-------------------------
Filter Length  : 123
Stable         : Yes
Linear Phase   : Yes (Type 1)

Design Method Information
Design Algorithm : Kaiser window

Design Options
Minimum Order  : any
Scale Passband : true

Design Specifications
Sample Rate              : 60 kHz
Response                 : Bandpass
Specification            : Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2
Second Passband Edge     : 20 kHz
Second Stopband Atten.   : 65 dB
First Stopband Atten.    : 65 dB
Second Stopband Edge     : 22 kHz
First Stopband Edge      : 8 kHz
Passband Ripple          : 0.5 dB
First Passband Edge      : 10 kHz
```

Fig. 11 Information of the BPF designed by the Kaiser window

Again, we plotted pole-zeros and we had similar observations as in the LPF. And for the Parks-McCellan filter, we have



```
Discrete-Time FIR Filter (real)
-------------------------------
Filter Structure  : Direct-Form FIR
Filter Length     : 113
Stable            : Yes
Linear Phase      : Yes (Type 1)

Implementation Cost
Number of Multipliers            : 113
Number of Adders                 : 112
Number of States                 : 112
Multiplications per Input Sample : 113
Additions per Input Sample       : 112
```

Fig. 12 Information of the BPF designed by the Parks-McCellan Algorithm

Both are type 1 filters. In Figure 13, we see the impulse response of the BPF designed by the Parks-McCellan Algorithm.

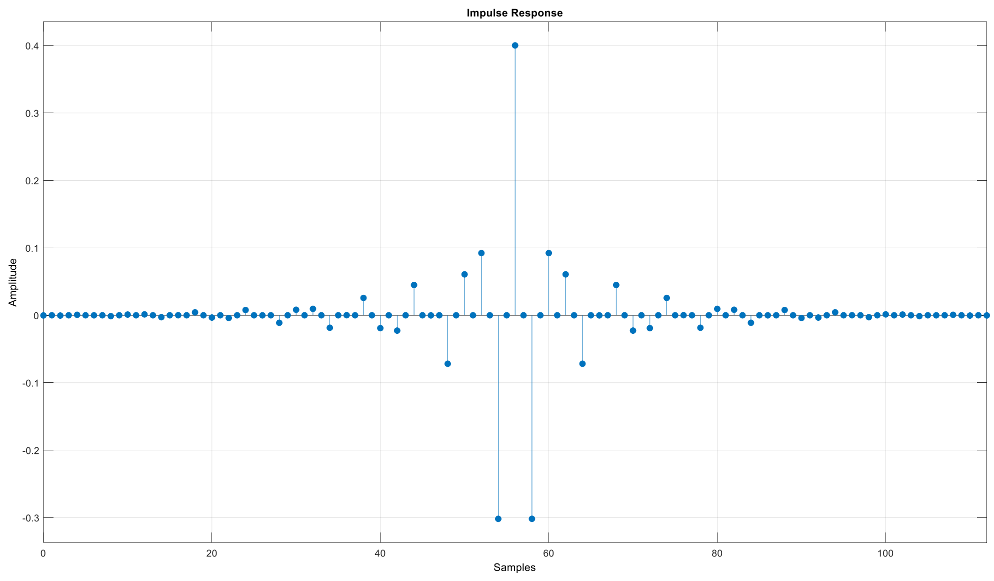


Fig. 13 Impulse response of the BPF designed by the Parks-McCellan algorithm

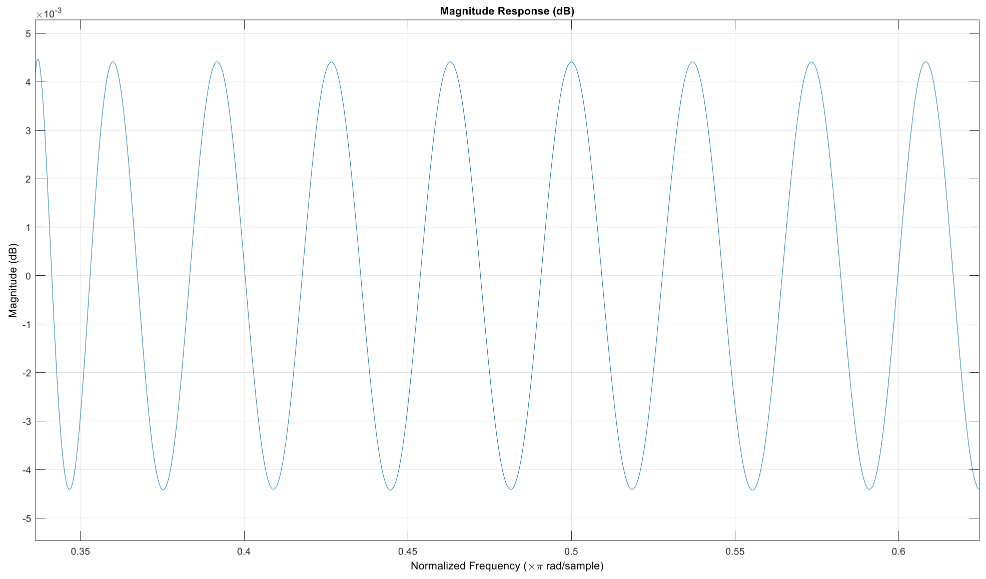


Fig. 14 Ripples in the passband of the BPF designed by the Parks-McCellan algorithm

Figure 14 presents the behavior of the ripples which is similar to the stopband ripples. By checking the values on the both stopbands and passband, we see that amount of ripple is more than 65 dB. And again, the number of filter coefficients are lesser than the filter designed by the Kaiser algorithm.

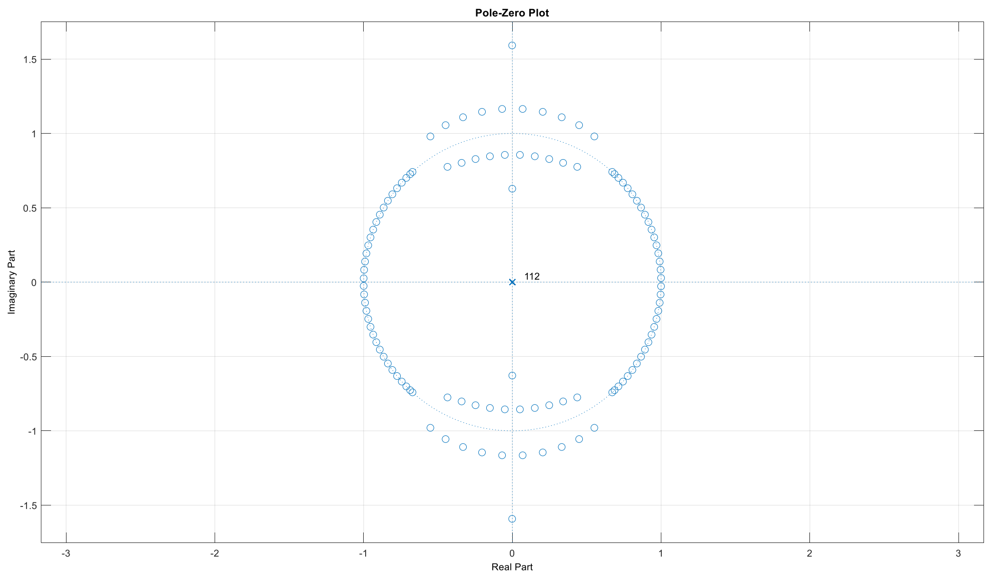


Fig. 15 Pole-zero plots of the BPF designed by the Parks-McCellan algorithm

All of the coefficients of polynomials are real, therefore the poles and zeros must be either purely real, or appear in complex conjugate pairs (Figure 15).

As we the cost (number of taps of the filter) is very important for us, we choose the optimum design which were done by Parks-McCellan algorithm.

The relation between filter length, group delay, and phase response

Let’s examine the group delay. For LPF with Kaiser window,

(frequency,phase) pairs of (0 kHz,0 rad), and (24kHz, -75.54088 rad):

$$\frac{d\emptyset}{df} = \frac{-75.54088\ \text{rad} - 0\ \text{rad}}{(24-0)\times 10^3 Hz} = 0.0031\ \frac{rad}{Hz}$$

$$\tau_g = -\frac{d\emptyset}{d\omega} = -\frac{-75.54088\ \text{rad} - 0\ \text{rad}}{2\pi \times (24-0)\times 10^3 Hz} = 0.00049338\frac{rad}{Hz} \approx 0.5\ \text{ms}$$

$$f_s = 60\ \text{kSamples/second}$$

$$\tau_g = \frac{30\ samples}{60\ \text{kSamples/second}} = 0.5\ \text{ms}$$

Let's try it for the normalized frequency case for the filter designed by the Parks-McCellan algorithm (f_n is the normalized frequency). Let's consider two points, $(0.7324219, -72.30572)$ and $(0.3009033, -3.610991)$

$$-\frac{d\emptyset}{df_n} = -\frac{-72.30572 \text{ rad} - 3.610991 \text{ rad}}{(0.7324219 - 0.3009033) \times \pi \frac{rad}{sample}} = 56 \text{ samples}$$

Which is equal with the delay (number of samples that we see from the fvtool phase menu).

$$f_s = 60 \text{ kSamples/second}$$

$$\tau_g = \frac{56\, samples}{60 \text{ kSamples/second}} \approx 1 \text{ ms}$$

By making some tables, we do these calculations and also changing some other parameters to have a better understanding of the possible effects of some of these changings. As I have tried several different cases the group delay predicts the slope of the phase response, as above.

Table 2. Lowpass filter design with different setting of parameters

Filter type	**Design method**	**Number of taps**	**Delay (samples)**	**Delay (mSecond)**	**Passband frequency (kHz)**	**Stopband frequency (kHz)**	**Fs kSamples/ Second**	**Ripple (atten.) dB**
LPF	Kaiser	61	30	0.5	20	24	60	65
LPF	Kaiser	122	60.5	0.5	20	24	120	65
LPF	Parks-McCellan	56	27.5	0.46	20	24	60	65
LPF	Parks-McCellan	110	54.5	0.45	20	24	120	65
LPF	Kaiser	121	60	1	20	22	60	65
LPF	Kaiser	241	120	2	20	21	60	65
LPF	Kaiser	82	40.5	0.7	20	24	60	85
LPF	Kaiser	30	14.5	0.24	20	24	60	35
LPF	Kaiser	42	20.5	0.34	20	26	60	65
LPF	Kaiser	52	25.5	0.5	20	24	50	65
LPF	Kaiser	478	238.5	4	20	20.5	60	65

Moreover, as I have tried a wide variety of the parameter settings for both of the LPFs designed by the Kaiser window method and also the Parks-McCellan algorithm, their behavior to the changes in the parameters are somewhat similar. Hence, in most cases in Table 2, we see the LPF designed by the Kaiser window method.

As we see from Table 1, by increasing the sampling frequency, for example, when fs = 120 kSamples/second, the filter length would be twice, and the group delay approximately will not change. Decreasing the sampling frequency (to 50 kHz) is shown in the last row of the Table 2; it leads to the smaller number of taps, but the group delay approximately stays the same.

Suppose we fix the sampling frequency and decrease the stopband frequency to 22 kHz, or 21 kHz (any number smaller than 24 kHz), the filter length will increase (as we see, 121 taps for the first case and 241 taps for the second case). This in turn, will increase the group delay (since we did not change the sampling frequency in this case). In the opposite side, if we increase the stopband frequency to 26 kHz, we have the smaller number of taps (here, 42 for the LPF designed by the Kaiser window method); in turn, it leads to the smaller cost of implementation and also smaller amount of the delay of the filter. Indeed, as we decrease the transition band, we need a window with a greater length (a greater number of taps), which in the frequency domain is a sort of *sinc* function with a narrower mainlobe width, hence, it results in a sharper transition in the transient band of the filter.

If we increase the attenuation to 85 dB (along with the fixed parameters as our original case), the number of taps will increase. That is what we expected (more limitation on the specification, leads to some costs). And, if we decrease the attenuation to 35 dB, the number of taps will decrease (here, 30 for the LPF designed by the Kaiser method).

Moreover, by halving the transition band (in the window method) and fixing the other parameters, we have twice number of filter taps (with respect the earlier case). We can see from the window methods, that the length of window, has an inverse relationship with the mainlobe width (or equivalently the transition band width).

Table 3. Bandpass filter design with different setting of parameters

Filter type	Design method	Number of taps	Delay (samples)	Delay (mSecond) ≈	Passband frequency (kHz)	Stopband frequency (kHz) Low, High	Fs kSamples/ Second	Ripple (atten.) dB
BPF	Kaiser	123	61	1	10 - 20	8 , 22	60	65
BPF	Kaiser	246	122.5	1	10-20	8 , 22	120	65
BPF	Parks-McCellan	113	56	0.93	10-20	8 , 22	60	65
BPF	Parks-McCellan	221	110	0.92	10-20	8 , 22	120	65
BPF	Parks-McCellan	221	110	1.83	10-20	9 , 21	60	65
BPF	Parks-McCellan	221	110	2	10-20	8 , 21	60	65
BPF	Parks-McCellan	154	76.5	1.28	10-20	8 , 22	60	85
BPF	Parks-McCellan	54	26.5	0.44	10-20	8 , 22	60	35
BPF	Parks-McCellan	59	29	0.48	10-20	6 , 24	60	65
BPF	Parks-McCellan	80	39.5	0.66	10-20	8 , 22		35 and 65 (different ripples for stop 1 and stop 2)

For the Table 3, we have similar inferences as for Table 2. We can see that how changing different parameter affects the others (especially, their effects on the filter length). To shorten the report, I wouldn't repeat the similar discussions. One odd behavior that I have seen is illustrated in the following.

For the specifications of the BPF which is given in the problem, if we change the Fstop2 to 21 kHz (other parameters are the same), we see a phenomenon which is one of the problems with the optimal Parks-McCellan algorithm. Figure 16 shows the magnitude and phase of the filter

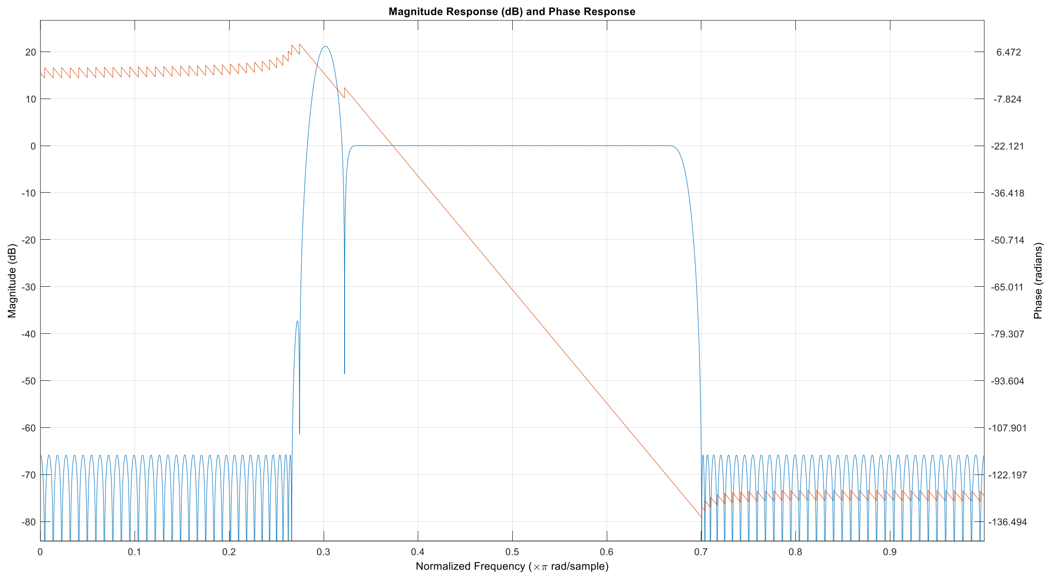


Fig. 16 Changing the Fstop2 to 21 kHz (other parameters are the same BPF in the problem)

In the normalized frequency axis, for the first stopband we have $\frac{f_{stop1}}{f_s/2} = \frac{8\ kHz}{30\ kHz} \approx 0.27$ and for the first passband we have $\frac{f_{pass1}}{f_s/2} = \frac{10\ kHz}{30\ kHz} \approx 0.33$; in between these two numbers we expected a smooth transition, but, there is a blow up. Indeed, the filters designed by the Parks-McCellan may have the problem of the blow up in the transition band (as mentioned in [2]). The row corresponding to this figure is shown with red in the Table 3. This is more surprising as the amount of space between 8 kHz to 10 kHz is more than the space between 20 kHz and 21 kHz (second stopband frequency). We can alleviate this by increasing (this is indeed surprising) 8 kHz to 9 kHz, which leads to a narrower transition band. The row above the red row in the Table 3 and also the Figure 17 show the result

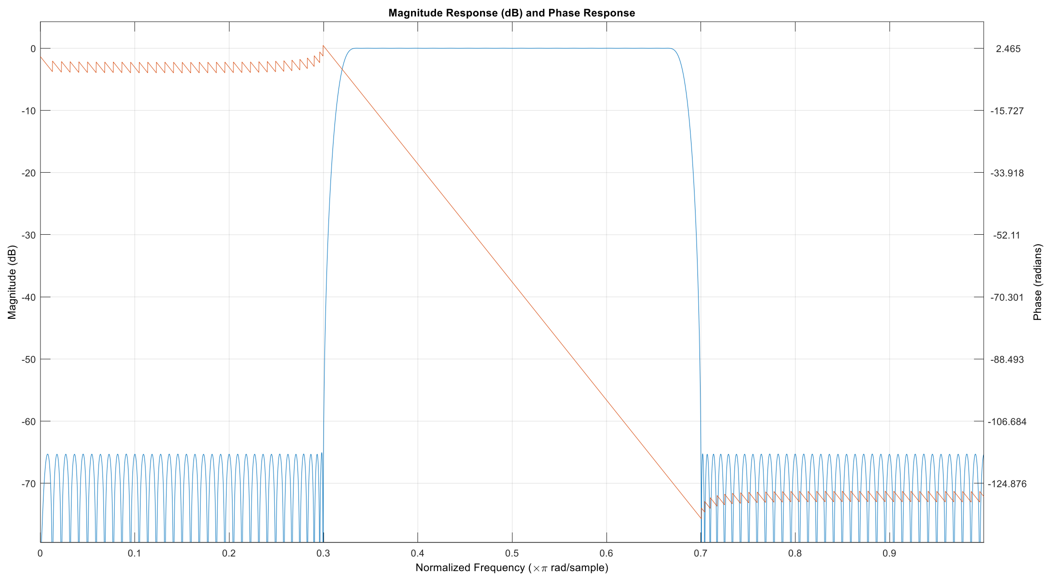
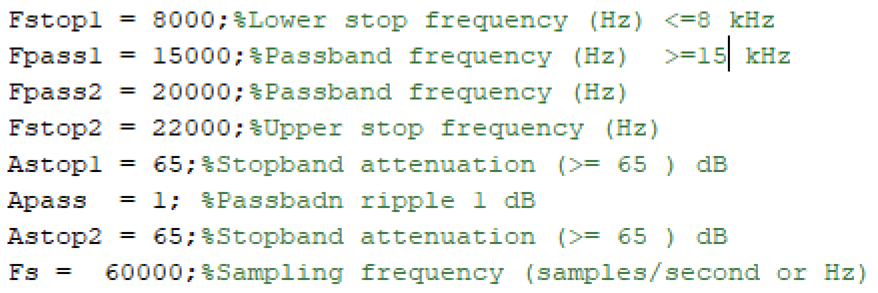


Fig. 16 Changing the Fstop1 to 9 kHz Fstop2 to 21 kHz (other parameters are the same BPF in the problem)

This is in clear contrast with the mindset that increasing transition band leads to a smoother transition of the filter in that region.

With the following parameters



```
Fstop1 = 8000;%Lower stop frequency (Hz) <=8 kHz
Fpass1 = 15000;%Passband frequency (Hz)  >=15 kHz
Fpass2 = 20000;%Passband frequency (Hz)
Fstop2 = 22000;%Upper stop frequency (Hz)
Astop1 = 65;%Stopband attenuation (>= 65 ) dB
Apass  = 1; %Passbadn ripple 1 dB
Astop2 = 65;%Stopband attenuation (>= 65 ) dB
Fs =  60000;%Sampling frequency (samples/second or Hz)
```

Figure 17 presents the filter designed by this parameter setting

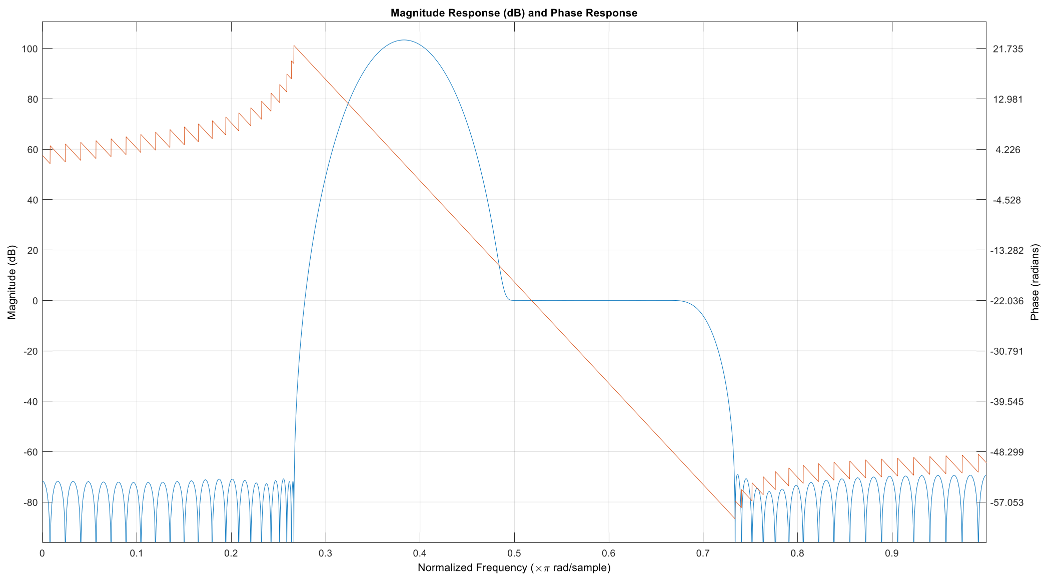


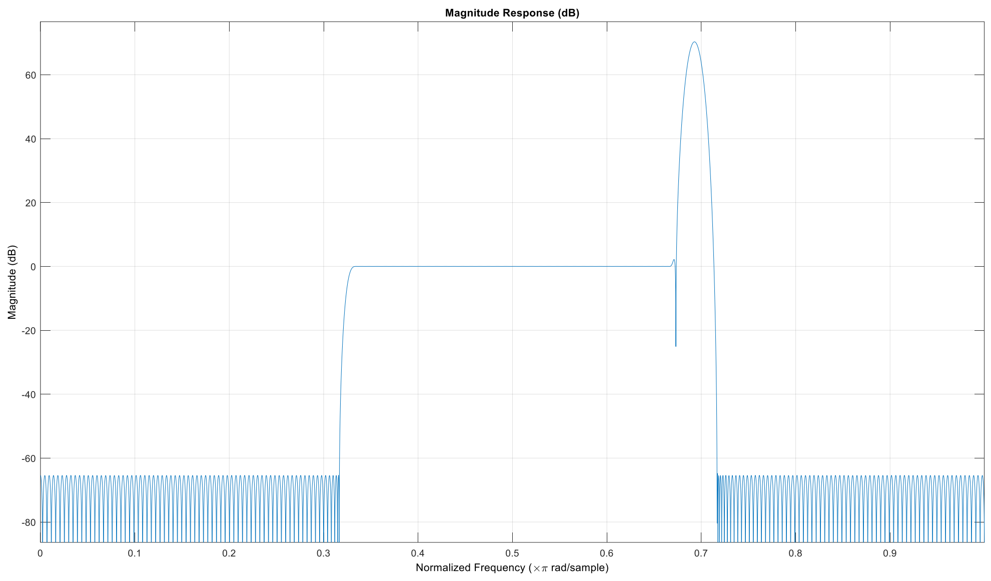
Fig. 17 Blow up phenomenon in a BPF designed by the Parks-McCellan Algorithm

Indeed, we expected that it passes the frequencies between 15000 and 2000, but it blows up in the left transition band. That's a quite surprise for me which I have not been familiar with that earlier. Moreover, the ripples are not exactly equal.

In case we decrease the transition band to a small region, as in

```
Fstop1 = 9500;%Lower stop frequency (Hz) <=9.5 kHz
Fpass1 = 10000;%Passband frequency (Hz)  >=10 kHz
Fpass2 = 20000;%Passband frequency (Hz)
Fstop2 = 21500;%Upper stop frequency (Hz)
Astop1 = 65;%Stopband attenuation (>= 65 ) dB
Apass  = 1; %Passbadn ripple 1 dB
Astop2 = 65;%Stopband attenuation (>= 65 ) dB
Fs =  60000;%Sampling frequency (samples/second or Hz)
```

We see the similar phenomenon



As I showed earlier, we can fix the problem by making some changes to the intervals. In summary, this problem occurs for the BPFs designed by the Parks-McCellan (it does not happen for the Kaiser window method). Another nice property which attracted me, is the small coefficients of the filter taps, and I would like to get into that deeper.

To sum up, the primary advantages FIR filters are:

- They can have exactly linear phase.
- They are always stable.
- They can be realized efficiently in hardware.
- The filter startup transients have finite duration.

The primary disadvantage of FIR filters is that they are slow, i.e., they often require a much higher filter order (then, a lot of computations) than IIR filters to achieve a given level of performance. In accordance, the delay of these filters is frequently substantially higher than for an IIR filter with equivalent performance.

References

[1] Discrete-Time Signal Processing. A. Oppenheim, R. Schafer, and J. Buck. Prentice-hall Englewood Cliffs, Second edition, (1999).

[2] https://www.dsprelated.com/freebooks/sasp/Bandpass_Filter_Design_Example.html