Laboratory work : Audio signal filtering

Alban-Félix Barreteau
Master 1 of Control and Robotics - Signal and Image Processing
Ecole Centrale de Nantes
alban-felix.barreteau@eleves.ec-nantes.fr
Phone : +33 6 51 68 15 98

December 11, 2020

Professor : Saïd Moussaoui

Abstract

This laboratory work gives an example of filter synthesis and its application to an audio signal processing (elimination or extraction of notes) using the available functions in Matlab. This report contains the results, answers to the questions and the Matlab script is provided in an appendix.

Contents

1 Signal generation and analysis

We simulate a sequence of the eight following notes :

Note	Do	Re	Mi	Fa	Sol	La	Si	Do
Frequency (Hz)	262	294	330	349	392	440	494	523

The duration is one second for each note and the sampling frequency is $f_s = 8192$ Hz.

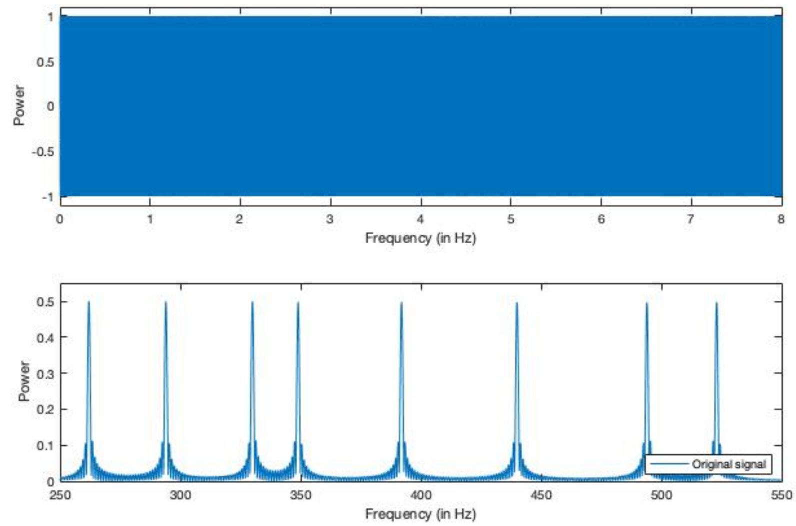


Figure 1: The temporal signal and its spectrum

2 Filtering of an audio signal

2.1 Analog low-pass filtering

In this first part, the retained specifications for the analog filters are $R_p = 3$ dB, $R_s = 40$ dB, $f_c = 420$ Hz and $\Delta f = 100$ Hz.

Question 1 :

	Required order
Butterworth filter	20
Chebyshev type 1 filter	8
Chebyshev type 2 filter	8
Cauer filter	5

The filter the most advantageous is the one which have the lowest order. With this preliminary filter specifications, the most advantageous analog filter is the Cauer filter is a required order of 5.
We can notice that the required order of the Butterworth filter is high in despite of low constraints on the filter specifications.

Question 2 : The four filters meet the specifications but we can notice that :
- The butterworth and the Chebyshev type 2 filters have no ripples in the passband ;
- The Cauer and Chebyshev type 1 filters attenuate a lot the frequencies in the transition band.

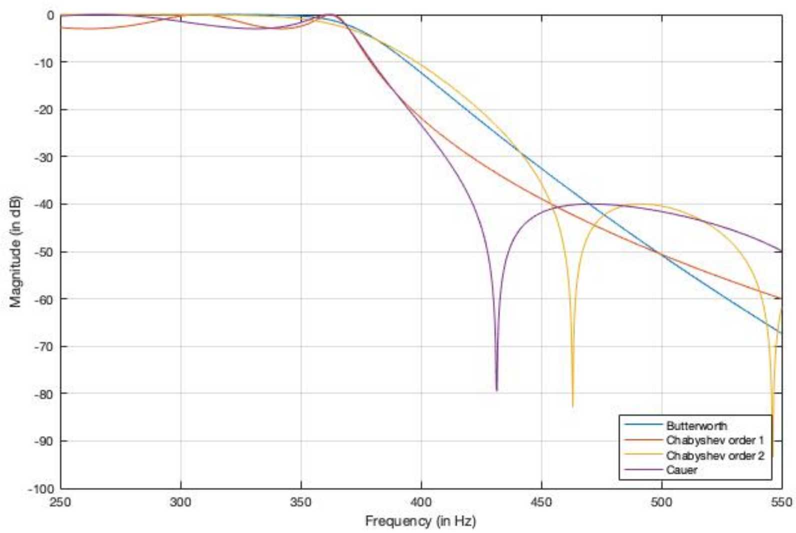
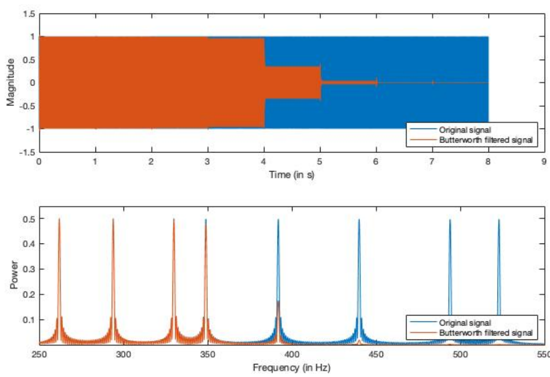


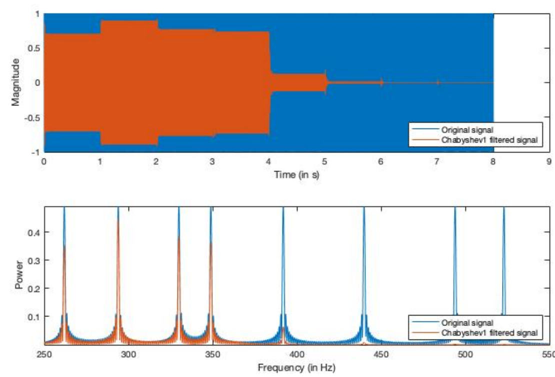
Figure 2: Magnitude frequency responses of the 4 filters.

Question 3 : By applying the four filters on the gam signal, we hear that the last three notes are not totally suppressed but very attenuated. But the Sol, which is the fourth penultimate note (i.e. the last note that we want to keep without filtering it), is also attenuated a bit.

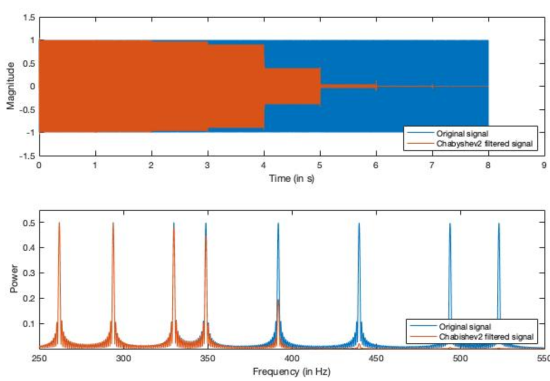
Question 4 : By looking at the figure (filtered signals and spectral analysis), we see that the Sol is partially filtered, with the 4 filters. We deduce that the constraints for the filter specifications are too small, in particular the bandwidth is too large. Furthermore, in this case, the Chebyshev type 1 and the Cauer filters filtered a bit the four first gam's notes, which is an unwanted effect. So with this low specifications, the best results are obtained with the Butterworth and the Chebyshev type 2 filters. But as the required order of the Butterworth filter is high (≈ 20), we should prefer to use the Chebyshev type 2 filter (required order equals to 8).



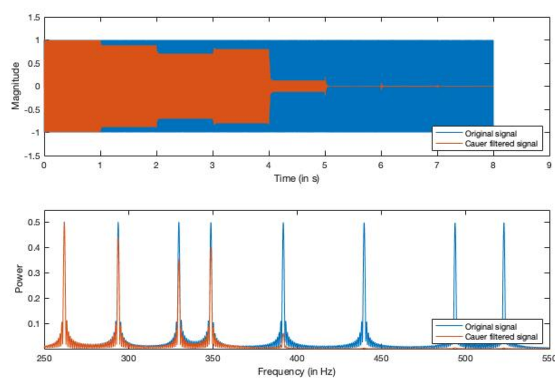
(a) Butterworth filter



(b) Chebyshev type 1 filter



(c) Chebyshev type 2 filter



(d) Cauer of order 1 filter

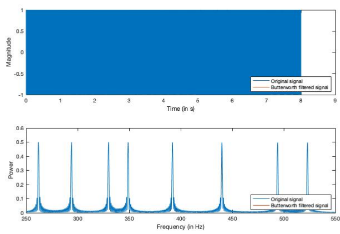
Figure 3: Filtered signals and spectral analysis

Question 5 :

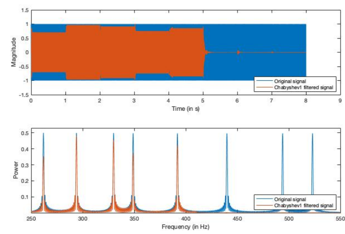
a) We decide to change the filter specifications. In fact, we reduce the transition bandwidth. The specifications are now $R_p = 3$ dB, $R_s = 40$ dB, $f_c = 420$ Hz and $\Delta f = 20$ Hz.

The required orders are now :

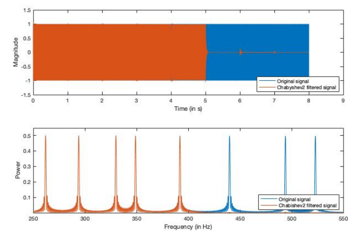
	Required order
Butterworth filter	97
Chebyshev type 1 filter	18
Chebyshev type 2 filter	18
Cauer filter	7



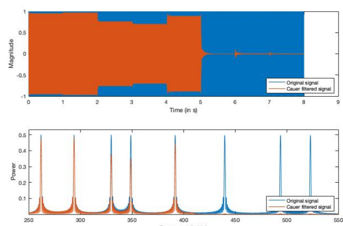
(a) Butterworth filter



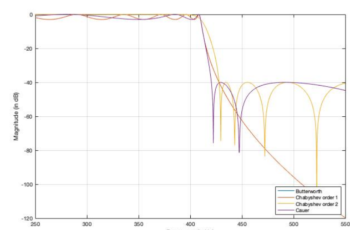
(b) Chebyshev type 1 filter



(c) Chebyshev type 2 filter



(d) Cauer filter



(e) Frequency responses of the filters

Figure 4: Reduction of the transition bandwidth : $R_p = 3$ dB and $\Delta f = 20$ Hz

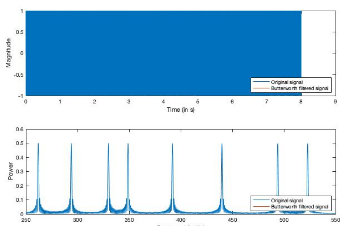
By reducing the bandwidth of the filter specifications, then the signal is very well filtered by the analog Chebyshev type 2 filter. This result is very convincing : the last three notes are well filtered and the five others are not so degraged (we can see little transient effects but they are very small).
Note that as the bandwidth is tight, the required order of the analog Butterworth filter is too high and there is an error during the numerical computation (we are not able to calculate it as the order is too high).
The Chebyshev type 1 and the Cauer filters filtered well the last three notes of the signal but attenuate a bit the five others. That's why we should prefer the Chebyshev type 2 filter in this case, in despite of his required order which is a bit high (equals to 18).

b) Then we also reduce the ripples in the passband : $R_p = 1$ dB, $R_s = 40$ dB, $f_c = 420$ Hz and $\Delta f = 20$ Hz.

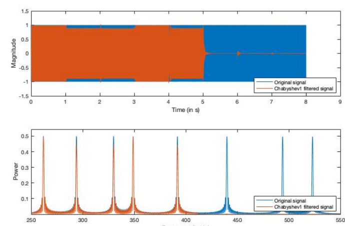
The required orders are :

	Required order
Butterworth filter	111
Chebyshev type 1 filter	20
Chebyshev type 2 filter	20
Cauer filter	7

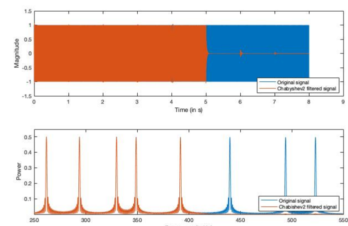
Once the ripples in the passband reduced to 1 dB, the required orders of the filters are not significatively increased (except for the Butterworth filter but we don't consider him anymore since his order is too high). The conclusion in this case is the same than for the question 5a : we should prefer the Chebyshev type 2 filter for the same reasons than above. But the ripples in the passband are even smaller. This means that the signal in the passband is less degraded by the filter.
This result is coherent with the idea that the more restrictive the specifications of the filter are, the better the signal will be filtered, but this filter will require an higher order and will be more expensive.



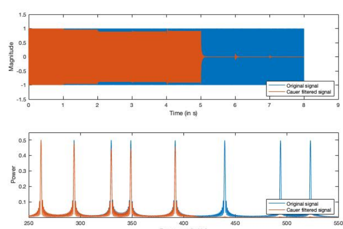
(a) Butterworth filter



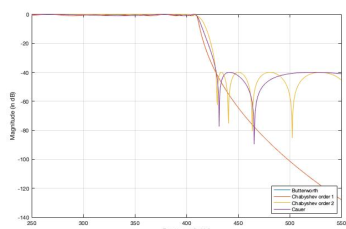
(b) Chebyshev type 1 filter



(c) Chebyshev type 2 filter



(d) Cauer of order 1 filter



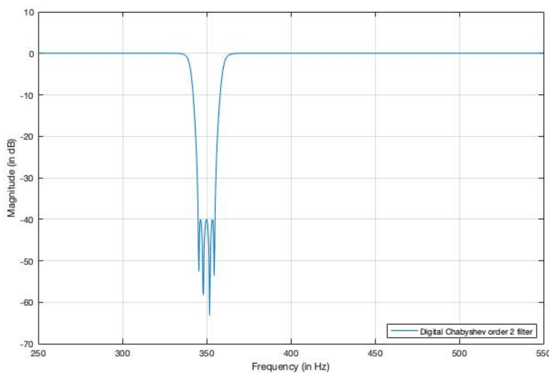
(e) Frequency responses of the filters

Figure 5: Reduction of the transition bandwidth : $R_p = 1$ dB and $\Delta f = 20$ Hz

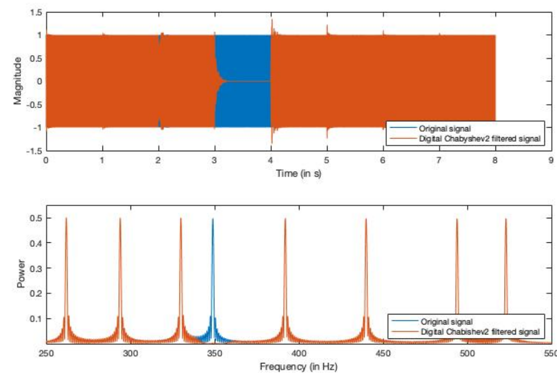
2.2 Rejection of one note by digital filtering

The retained specifications for the digital filters are $f_c^l = 340$ Hz, $f_c^h = 360$ Hz, $\Delta f_l = \Delta f_h = 10$ Hz, $R_p = 1$ dB and $R_s = 40$ dB. We first design a IIR filter of type Chebyshev 2.

Question 1 and 2 : The required order for the digital filter of type Chebyshev 2 that allows to eliminate the Fa note from the signal is 4 (calculation made by the Matlab script provided in appendix).
We can observe some transient effects in the filtered signal magnitude at the note discontinuities because of the brutal change of frequency input. In fact, it is like an overshoot in the step response. We can't avoid this effect.



(a) Magnitude frequency response of the digital Chebyshev type 2 stopband filter.



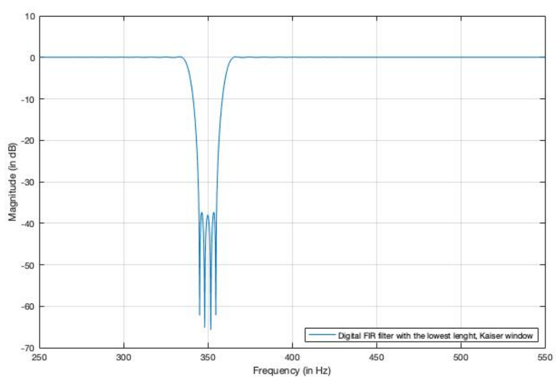
(b) Original and digital Chebyshev type 2 filtered signals, and their magnitude spectrum.

Figure 6: IIR filter (digital Chebyshev type 2)

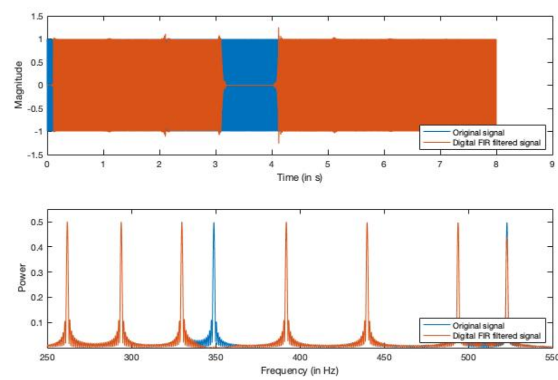
Now we want to design a FIR filter.

Question 3 : We used a Kaiser window for the design of this FIR filter. This window is defined by the Bessel function and allows to adapt the filter to the desired attenuation. β is the shape parameter of the window and its value is determined according to this desired attenuation. The length of the filter N is determined according to the ripples in the attenuated band R_s and the transition bandwidth Δf such as : $N \geq \frac{(R_s-8)}{4{,}57\pi} \cdot \frac{f_s}{\Delta f}$.

The *fir1* function calculates the coefficients of the FIR filter by truncating the impulse response of the ideal filter. In this case, the required length for the FIR digital filter is $N = 1830$. Note that the attenuation band is not -40 dB everywhere because the numerical computation of the required order is an estimation (see the *Caution* in the Matlab documentation of the *kaiserord* function : it specifies that the order N is just an estimation and we should use an higher order if the filter doesn't meet the original specifications). In order to strictly respect the specifications of -40 dB, we can either increase a bit the contraints on the attenuation band, or increase the order of the filter.
On the temporal filtered signal figure, one can observe the linear phase : at time $t = 0$ s, the signal isn't filtered, the Fa note isn't filtered at time $t = 3$ s but at time $t = 3 + \delta t$ s, and so on. This lag effects are due to the phase shift.
We also have transient effects at the note discontinuities like we had with the IIR filter.



(a) Magnitude frequency response of the digital FIR stopband filter, having the lowest lenght (Kaiser window).



(b) Original and digital Chebyshev type 2 filtered signals, and their magnitude spectrum.

Figure 7: FIR filter with the lowest lenght, Kaiser window

Question 4 : As explained above, the FIR filter doesn't respect exactly the $R_s = -40$ dB specification as the IIR filter does. We also can see that the FIR filter has a transition band a bit wider and ripples in the attenuation band bigger than the IIR filter.

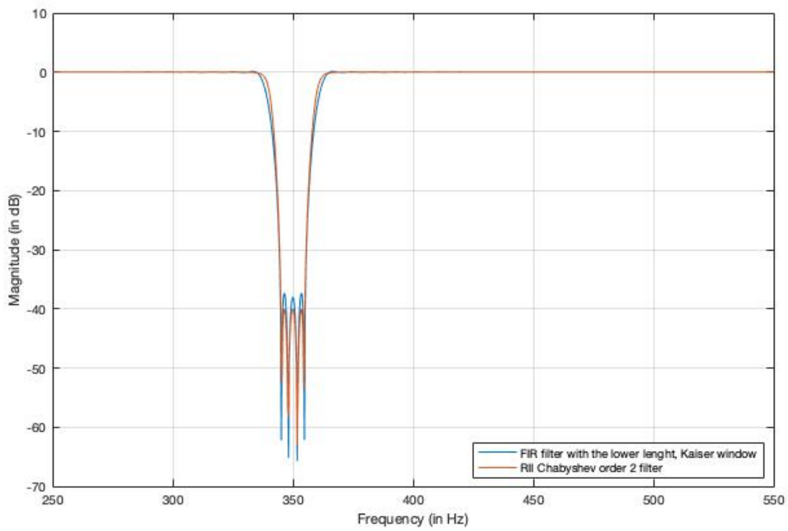


Figure 8: Magnitude frequency responses of the RII and FIR filters.

2.3 Analog and digital filters : Conclusion

2.3.1 Analog filters

An analog filter is a filter which is composed of an agacement of material components, such as resistors, inductors, capacitors, and so on.
They are easy to apply, as there is no need for a microprocessor or write an algorithm and simple RC filters have very few components. Analog filters used to be faster than digital ones.

2.3.2 Digital filters

A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. They are programmable, which means they can be changed without modify the hardware. Also, they are very stable along the time and temperature, while analog filters could depend of these parameters.

IIR filters : they can have narrow transition band and have a greater digital sensitivity than a FIR filter but they are potential instable due to the poles of $H(z)$ located outside the unit circle. They also are less complex than a FIR filter with equivalent specifications.

To synthetize IIR filters, we start from analog filters. In fact, the frequency behavior of the analog filter is completely determined by its continuous transfer function $H(s)$, and we want to obtain the same type of frequency response with the digital filter described by its disctrete transfer function $\widetilde{H}(z)$. A common method of synthesis is based on the correspondence rule between the s Laplace variable and the z variable such as we obtain a numerical approximation of the temporal and frequency responses of the analog filter. The usual transformations are : the associated transformation, the transformation by approximation of the derivative, the Tustin transformation, or the impulse/step/ramp invariance methods.

FIR filters : their transition band is wider than an IIR filter having the same number of coefficients. Its synthesis methods allow to derivate any frequency response and it is always stable (they have a bound response for a bound input). The phase can be exactly linear, so it avoid harmonic distortion in the signal. Finally, FIR filters are easier to implemant in a digital processing system.

In order to synthetize FIR filters : the impulse responses of the ideal filters are infinitely long and non-causal, so we must first truncate the response and then shift it so that the filter is causal. This truncation modifies the frequency response : it leads to ripples in the pass and attenuation bands, and introduces a transition band. Besides, the shift in the impulse response creates a linear phase. In order to obtain the best compromise between weak ripples and narrow transition band, we have to choose a window whose the frequency behavior is matching the filter to be produced.

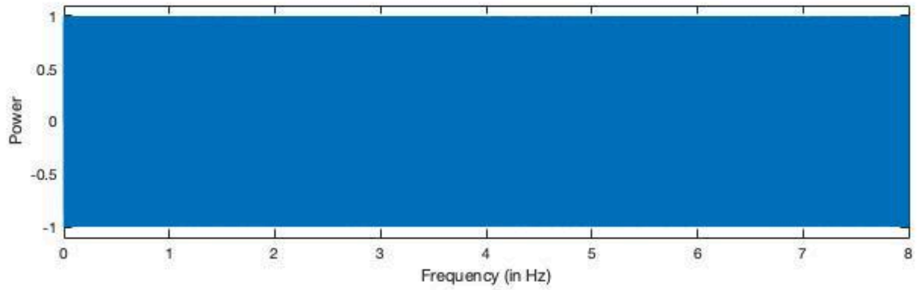
Table of Contents

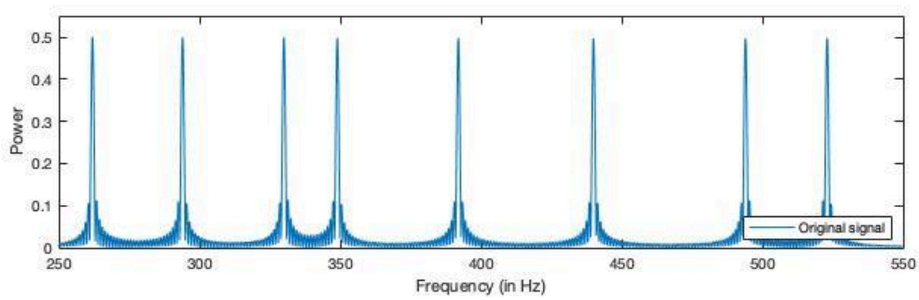

```
% Alban-Félix Barreteau, M1 CORO SIP
% Mail : alban-felix.barreteau@eleves.ec-nantes.fr
% Matlab R2020a Update 5 (9.8.0.1451342), Student license
% Signal Filtering and System Identification (SISIF)
% Labwork 1 : Audio signal filtering

% Professor : said.moussaoui@ec-nantes.fr

%-----------------------------------------------------------------%

clc

% Definition of your screen :
hpixels=1440;
vpixels=900;
```

Signal generation and analysis

```
% Signal simulation
duration=1; %duration of each note
fs=8192; %sampling frequency
fmin=250; fmax=550; f = linspace(fmin,fmax,1000);
[sig,t]=gamme_lab1(duration, fs); %simulate the signal
%soundsc(sig, fs); %play the signal

%Spectral analysis of the signal
fsig = [0:length(sig)-1]*(fs/length(sig));
S = fft(sig)/fs;

%Plotting the temporal signal and its spectrum
figure(10), hold off
set(figure(10),'name','Temporal gam signal and its
 spectrum','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])
subplot(2,1,1)
plot(t,sig)
ymin=-1.1*max(abs(sig)); ymax=1.1*max(abs(sig)); axis([0 8 ymin ymax])
```

```
xlabel('Frequency (in Hz)'), ylabel('Power')

subplot(2,1,2)
plot(fsig, abs(S));
xmin=fmin; xmax=fmax; ymin=0; ymax=1.1*max(abs(S)); axis([xmin xmax
 ymin ymax])
xlabel('Frequency (in Hz)'), ylabel('Power')
legend('Original signal',"location",'southeast')
```





The retained specifications of the analog low-pass filter

```
fc=420; %The cutoff frequency
deltaf=20; %The transition band's width frequency
fpb=fc-deltaf/2; %Passband frequency
fsb=fc+deltaf/2; %Stopband frequency
Wp=2*pi*fpb; %Normalized passband edge frequency
Ws=2*pi*fsb; %Normalized stopband edge frequency
Rp=1; %Ripple in the passband
Rs=40; %Ripple in the stopband
```

Required order of the 4 filters

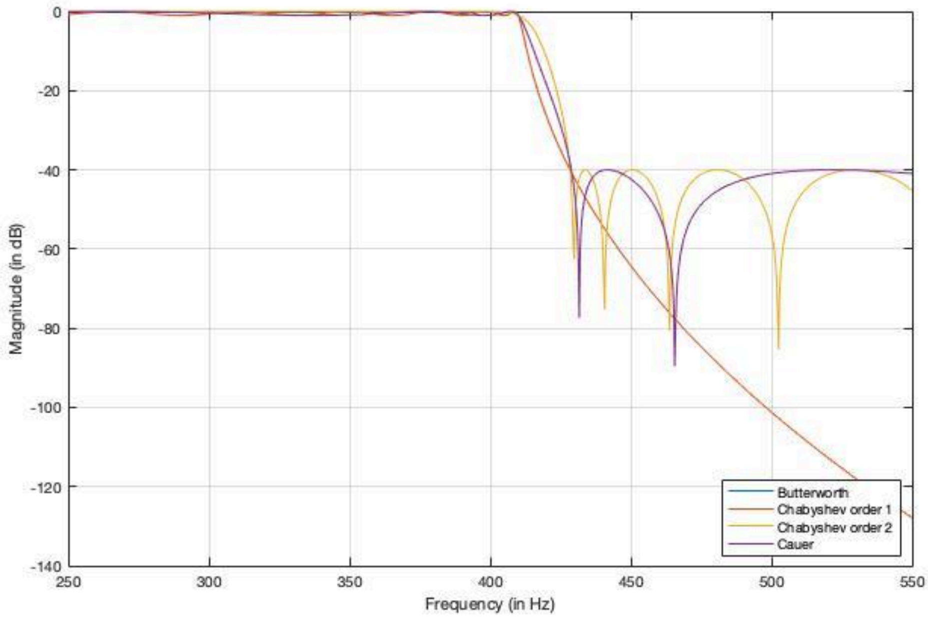
```
% Butterworth filter, N the lowest order and Wn the Butterworth
 natural frequency
[NButterworth, WnButterworth]=buttord(Wp,Ws,Rp,Rs,'s'); %for an analog
 filter, Wp and Ws in rad/s
[numButterworth,denButterworth]=butter(NButterworth,WnButterworth,'s');
```

```
HcButterworth=freqs(numButterworth,denButterworth,2*pi*f);
disp(['The required order for the Butterworth filter is
 ',num2str(NButterworth)])

% Chabyshev filter order 1, N the lowest order and Wp the Chebyshev
 natural frequency
[NChabyshev1, WpChabyshev1]=cheb1ord(Wp, Ws, Rp, Rs, 's'); %for an
 analog filter, Wp and Ws in rad/s
[numChabyshev1,denChabyshev1]=cheby1(NChabyshev1,Rp,WpChabyshev1,'s');
HcChabyshev1=freqs(numChabyshev1,denChabyshev1,2*pi*f);
disp(['The required order for the Chabyshev order 1 filter is
 ',num2str(NChabyshev1)])


% Chabyshev filter order 2, N the lowest order and Wp the Chebyshev
 natural frequency
[NChabyshev2, WpChabyshev2]=cheb2ord(Wp, Ws, Rp, Rs, 's'); %for an
 analog filter, Wp and Ws in rad/s
[numChabyshev2,denChabyshev2]=cheby2(NChabyshev2,Rs,WpChabyshev2,'s');
HcChabyshev2=freqs(numChabyshev2,denChabyshev2,2*pi*f);
disp(['The required order for the Chabyshev order 2 filter is
 ',num2str(NChabyshev2)])


% Cauer filter, N the lowest order and Wp the elliptic natural
 frequency
[NCauer, WpCauer] = ellipord(Wp, Ws, Rp, Rs, 's');
[numCauer, denCauer]=ellip(NCauer,Rp,Rs,Wp,'s');
HcCauer=freqs(numCauer,denCauer,2*pi*f);
disp(['The required order for the Cauer filter is ',num2str(NCauer)])
```

```
The required order for the Butterworth filter is 111
The required order for the Chabyshev order 1 filter is 20
The required order for the Chabyshev order 2 filter is 20
The required order for the Cauer filter is 7
```

Magnitude frequency response of the 4 filters

```
figure(20), hold off
set(figure(20),'name','Magnitude frequency responses of the 4
 filters','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])
plot(f,20*log10(abs(HcButterworth))); hold on
plot(f,20*log10(abs(HcChabyshev1)));
plot(f,20*log10(abs(HcChabyshev2)));
plot(f,20*log10(abs(HcCauer)));
xlabel('Frequency (in Hz)'), ylabel('Magnitude (in dB)'), grid on
legend('Butterworth','Chabyshev order 1','Chabyshev order
 2','Cauer',"location",'southeast')
% The most advantageous filter is the cheaper one wich means the one
 with the lowest order
```



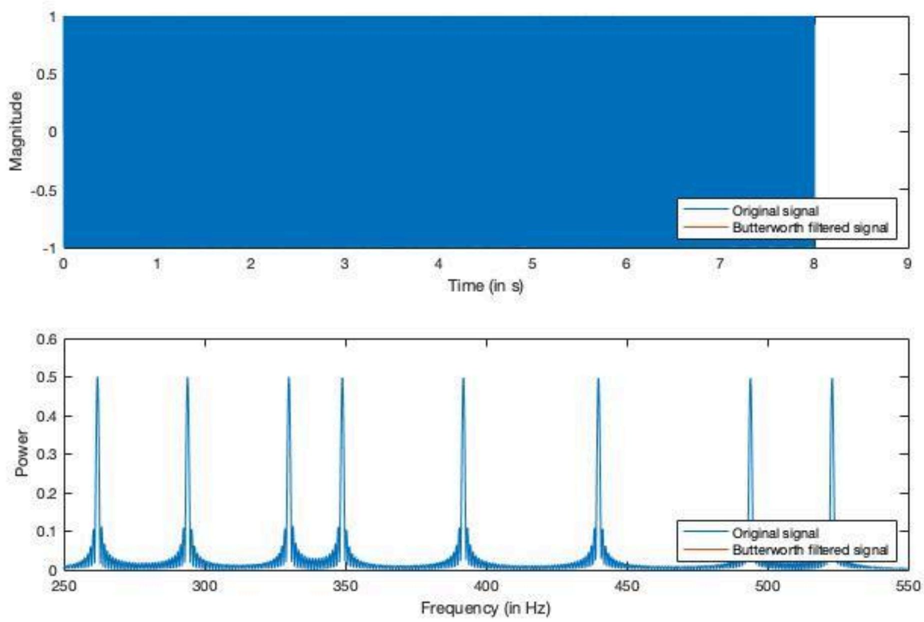
Signal filtering and spectral analysis

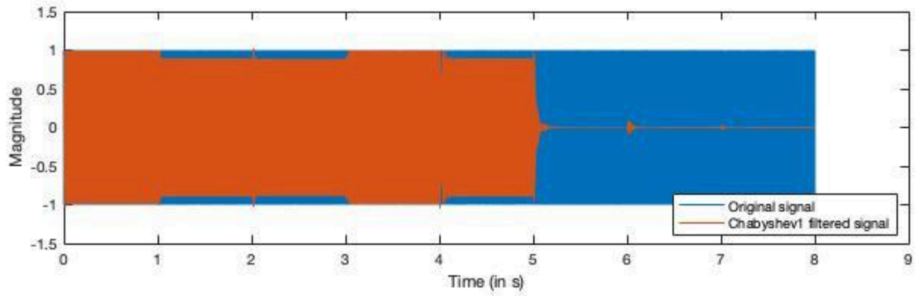
```
% Butterworth filter
sigfButterworth =
 lsim(tf(numButterworth,denButterworth),sig,t); %soundsc(sigfButterworth,
 fs);
fsigButterworth=[0:length(sigfButterworth)-1]*(fs/
length(sigfButterworth));
SButterworth=fft(sigfButterworth)/fs;

figure(31), hold off
set(figure(31),'name','Original signal and Butterworth filtered
 signal','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])

subplot(2,1,1)
plot(t,sig); hold on; plot(t,sigfButterworth);
xlabel('Time (in s)'), ylabel('Magnitude')
legend('Original signal','Butterworth filtered
 signal',"location",'southeast')

subplot(2,1,2)
plot(fsig, abs(S)), hold on; plot(fsigButterworth,abs(SButterworth))
xlabel('Frequency (in Hz)'), ylabel('Power')
legend('Original signal','Butterworth filtered
 signal',"location",'southeast')
try
axis([fmin fmax min(abs(SButterworth)) 1.1*max(abs(SButterworth))])
catch disp('Error during the numerical computation')
end
```

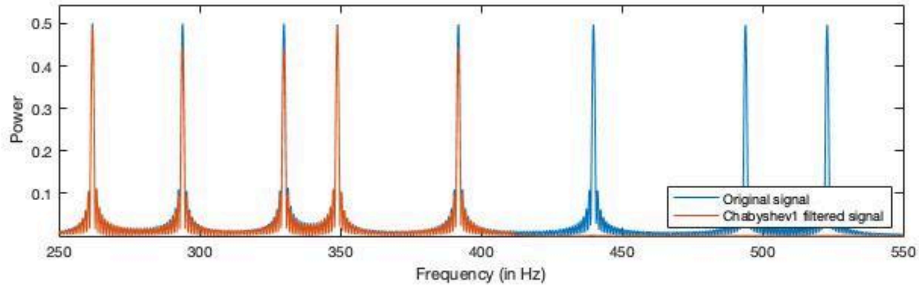
```
% Chabyshev filter of order 1
sigfChabyshev1 =
 lsim(tf(numChabyshev1,denChabyshev1),sig,t); %soundsc(sigfChabyshev1,
 fs);
fsigChabyshev1=[0:length(sigfChabyshev1)-1]*(fs/
length(sigfChabyshev1));
SChabyshev1=fft(sigfChabyshev1)/fs;

figure(32), hold off
set(figure(32),'name','Original signal and Chabyshev1 filtered
 signal','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])

subplot(2,1,1)
plot(t,sig); hold on; plot(t,sigfChabyshev1);
xlabel('Time (in s)'), ylabel('Magnitude')
legend('Original signal','Chabyshev1 filtered
 signal',"location",'southeast')

subplot(2,1,2)
plot(fsig, abs(S)), hold on; plot(fsigChabyshev1,abs(SChabyshev1))
xlabel('Frequency (in Hz)'), ylabel('Power')
legend('Original signal','Chabyshev1 filtered
 signal',"location",'southeast')
axis([fmin fmax min(abs(SChabyshev1)) 1.1*max(abs(SChabyshev1))])



% Chabyshev filter of order 2
sigfChabyshev2 =
 lsim(tf(numChabyshev2,denChabyshev2),sig,t); %soundsc(sigfChabyshev2,
 fs);
fsigChabyshev2=[0:length(sigfChabyshev2)-1]*(fs/
length(sigfChabyshev2));
SChabyshev2=fft(sigfChabyshev2)/fs;

figure(33)
set(figure(33),'name','Original signal and Chabyshev2 filtered
 signal','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])

subplot(2,1,1)
plot(t,sig); hold on; plot(t,sigfChabyshev2);
xlabel('Time (in s)'), ylabel('Magnitude')
legend('Original signal','Chabyshev2 filtered
 signal',"location",'southeast')

subplot(2,1,2)
plot(fsig, abs(S)), hold on; plot(fsigChabyshev2,abs(SChabyshev2))
xlabel('Frequency (in Hz)'), ylabel('Power')
legend('Original signal','Chabishev2 filtered
 signal',"location",'southeast')
axis([fmin fmax min(abs(SChabyshev2)) 1.1*max(abs(SChabyshev2))])
```

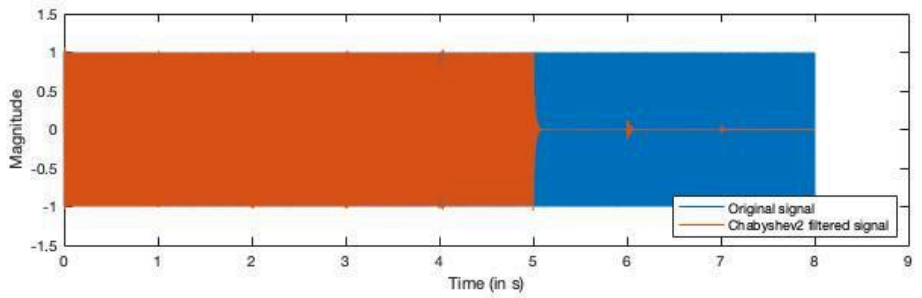
```
% Cauer filter
sigfCauer = lsim(tf(numCauer,denCauer),sig,t); %soundsc(sigfCauer,
 fs);
fsigCauer=[0:length(sigfCauer)-1]*(fs/length(sigfCauer));
SCauer=fft(sigfCauer)/fs;

figure(34)
set(figure(34),'name','Original signal and Cauer filtered
 signal','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])

subplot(2,1,1)
plot(t,sig); hold on; plot(t,sigfCauer);
xlabel('Time (in s)'), ylabel('Magnitude')
legend('Original signal','Cauer filtered
 signal',"location",'southeast')

subplot(2,1,2)
plot(fsig, abs(S)), hold on; plot(fsigCauer,abs(SCauer))
xlabel('Frequency (in Hz)'), ylabel('Power')
legend('Original signal','Cauer filtered
 signal',"location",'southeast')
axis([fmin fmax min(abs(SCauer)) 1.1*max(abs(SCauer))])
```

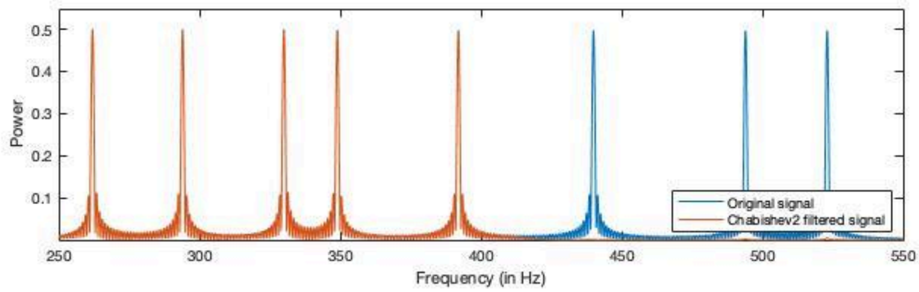
Error during the numerical computation

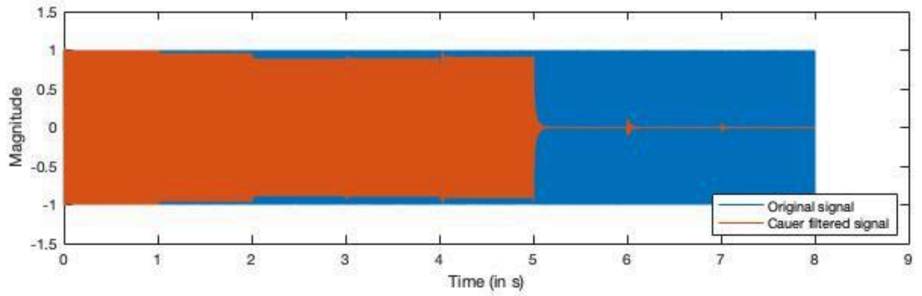


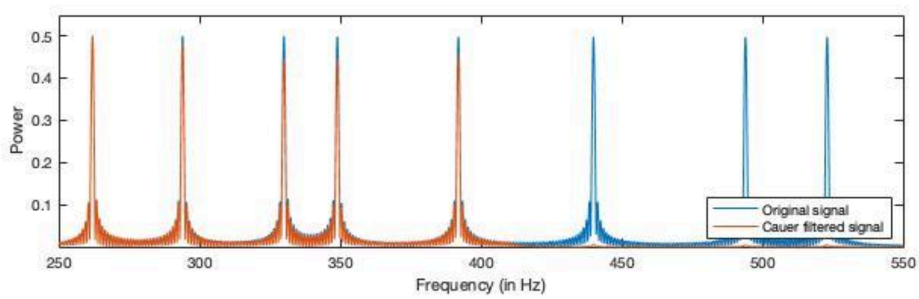

Magnitude
Time (in s)
Original signal
Chabyshev1 filtered signal


Power
Frequency (in Hz)
Original signal
Chabyshev1 filtered signal


Magnitude
Time (in s)
Original signal
Chabyshev2 filtered signal


Power
Frequency (in Hz)
Original signal
Chabishev2 filtered signal





The retained specification for the digital filter

```
fcl=340; %The low cutoff frequency
fch=360; %The high cutoff frequency
deltafl=10; %The low transition band's width frequency
deltafh=10; %The high transition band's width frequency
Rp=1; %Ripple in the passband
Rs=40; %Ripple in the stopband
```

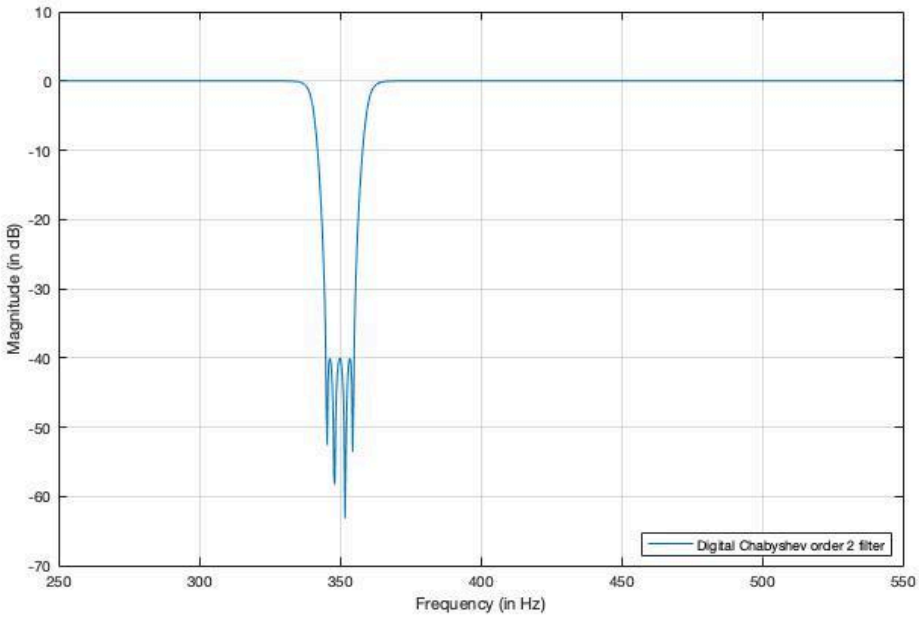
Digital RII filter Chabyshef of order 2

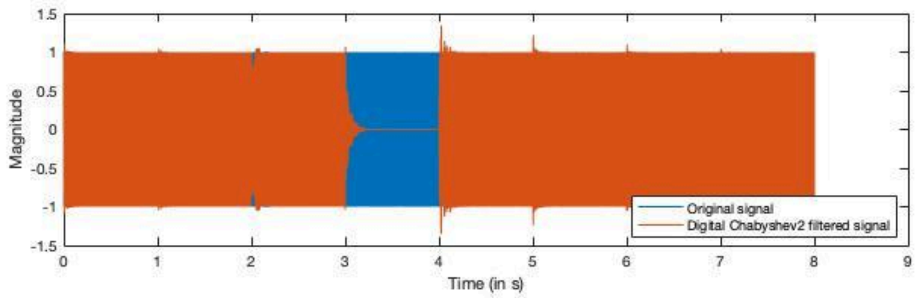
fpb and fsb should have 2 components because we design a stopband filter

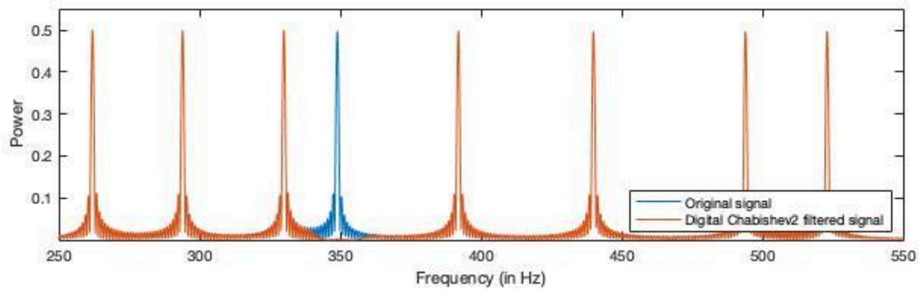
```
fpl=fcl-deltafl/2; %Low passband edge frequency
fph=fch+deltafh/2; %High passband edge frequency
fpb=[fpl fph]; %Passband edges frequency
Wp=2*fpb/fs; %Normalized passband edge frequency

fsl=fcl+deltafl/2; %Low stopband edge frequency
fsh=fch-deltafh/2; %High stopband edge frequency
fsb=[fsl fsh]; %Stopband edges frequency
Ws=2*fsb/fs; %Normalized stopband edge frequency

% Chabyshev filter order 2, N the lowest order and Wp the Chebyshev
 natural frequency
[NChabyshev2, WpChabyshev2]=cheb2ord(Wp, Ws, Rp, Rs); %for a digital
 filter
[numChabyshev2,denChabyshev2]=cheby2(NChabyshev2,Rs,WpChabyshev2,'stop');
HdChabyshev2=freqz(numChabyshev2,denChabyshev2,f,fs);
```

```
disp(['The required order for the Chabyshev order 2 digital filter is
 ',num2str(NChabyshev2)])


figure (40), hold off
set(figure(40),'name','Magnitude frequency response of the Chabyshev2
 digital stopband filter','position',[hpixels/2 vpixels/2 hpixels/2
 vpixels/2])
plot(f,20*log10(abs(HdChabyshev2))) %Frequency response of the RII
 filter
xlabel('Frequency (in Hz)'), ylabel('Magnitude (in dB)'), grid on
legend('Digital Chabyshev order 2 filter',"location",'southeast')



% Signal filtering and spectral analysis
sigfChabyshev2=filter(numChabyshev2,denChabyshev2,sig); %soundsc(sigfChabyshev2,
 fs);
fsigChabyshev2=[0:length(sigfChabyshev2)-1]*(fs/
length(sigfChabyshev2));
SChabyshev2=fft(sigfChabyshev2)/fs;

figure(50)
set(figure(50),'name','Original signal and digital Chabyshev2 filtered
 signal','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])

subplot(2,1,1)
plot(t,sig); hold on; plot(t,sigfChabyshev2);
xlabel('Time (in s)'), ylabel('Magnitude')
legend('Original signal','Digital Chabyshev2 filtered
 signal',"location",'southeast')

subplot(2,1,2)
plot(fsig, abs(S)), hold on; plot(fsigChabyshev2,abs(SChabyshev2))
xlabel('Frequency (in Hz)'), ylabel('Power')
legend('Original signal','Digital Chabishev2 filtered
 signal',"location",'southeast')
axis([fmin fmax min(abs(SChabyshev2)) 1.1*max(abs(SChabyshev2))])
```

The required order for the Chabyshev order 2 digital filter is 4





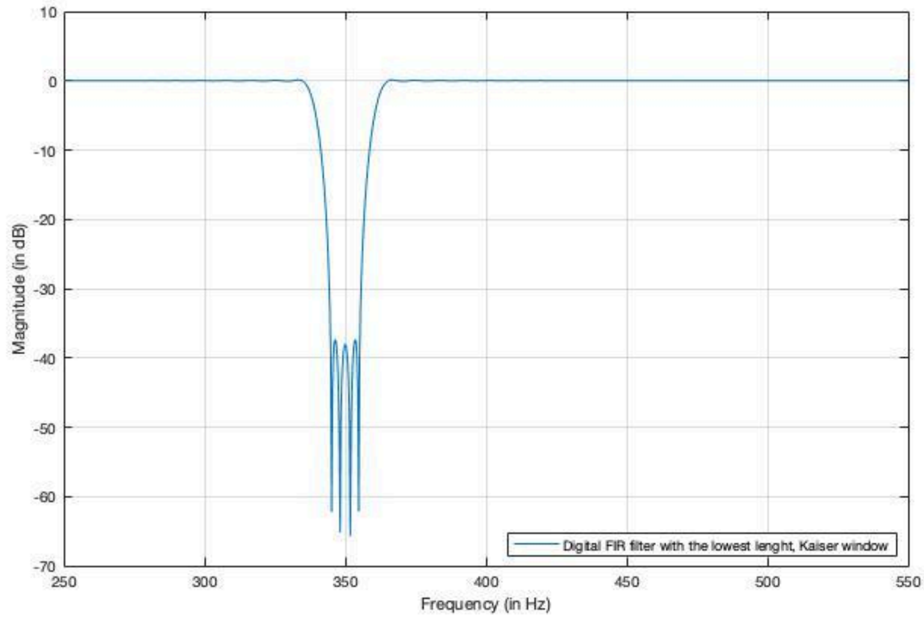


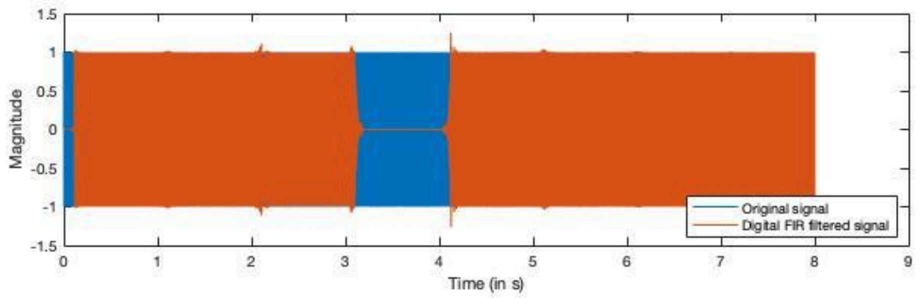
FIR filter having the lowest lenght

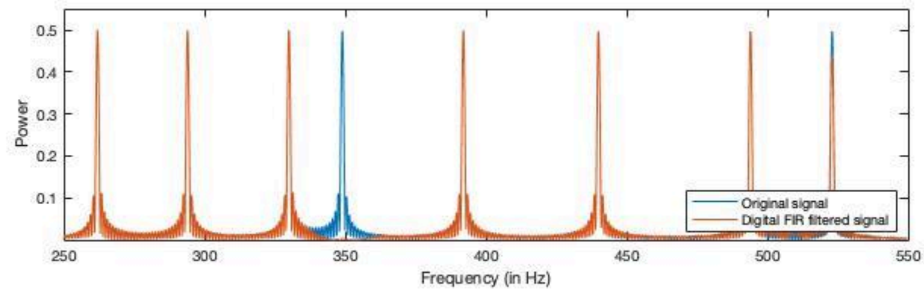
```
% Calculation of the filter length
fcuts=[fpl fsl fsh fph]; %Specifications of characteristics
 frequencies sorted in increasing values
mags=[1 0 1]; %Filter magnitude in each band (linear scale)
epsilonp=1-10^(-Rp/20);
```

```
deltaa=10^(-Rs/20);
devs=[epsilonp deltaa epsilonp]; %Magnitude of admitted ripples in the
 bands for a stopband filter
[N,Wn,beta,ftype] = kaiserord(fcuts,mags,devs,fs);
disp(['The required order for the FIR digital filter is ',num2str(N)])


% Calculation of the FIR filter by the window method
window='kaiser';
numFIR=fir1(N,Wn,ftype,kaiser(N+1,beta)); %Window based FIR filter
 design
HdFIR=freqz(numFIR,1,f,fs);

figure(60), hold off
plot(f,20*log10(abs(HdFIR))); %Frequency response of the filter
xlabel('Frequency (in Hz)'), ylabel('Magnitude (in dB)'), grid on
legend('Digital FIR filter with the lowest lenght, Kaiser
 window',"location",'southeast')
set(figure(60),'name','Frequency response of the FIR
 filter','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])


% Signal filtering and spectral analysis
sigfFIR=filter(numFIR,1,sig); %soundsc(sigfFIR,fs)
fsigFIR=[0:length(sigfFIR)-1]*(fs/length(sigfFIR));
SFIR=fft(sigfFIR)/fs;

figure(70)
subplot(2,1,1)
plot(t,sig); hold on; plot(t,sigfFIR);
xlabel('Time (in s)'), ylabel('Magnitude')
legend('Original signal','Digital FIR filtered
 signal',"location",'southeast')
subplot(2,1,2)
plot(fsig, abs(S)), hold on; plot(fsigFIR,abs(SFIR))
xlabel('Frequency (in Hz)'), ylabel('Power')
legend('Original signal','Digital FIR filtered
 signal',"location",'southeast')
axis([fmin fmax min(abs(SFIR)) 1.1*max(abs(SFIR))])
set(figure(70),'name','Original signal and digital FIR filtered
 signal','position',[hpixels/2 vpixels/2 hpixels/2 vpixels/2])
```

The required order for the FIR digital filter is 1830



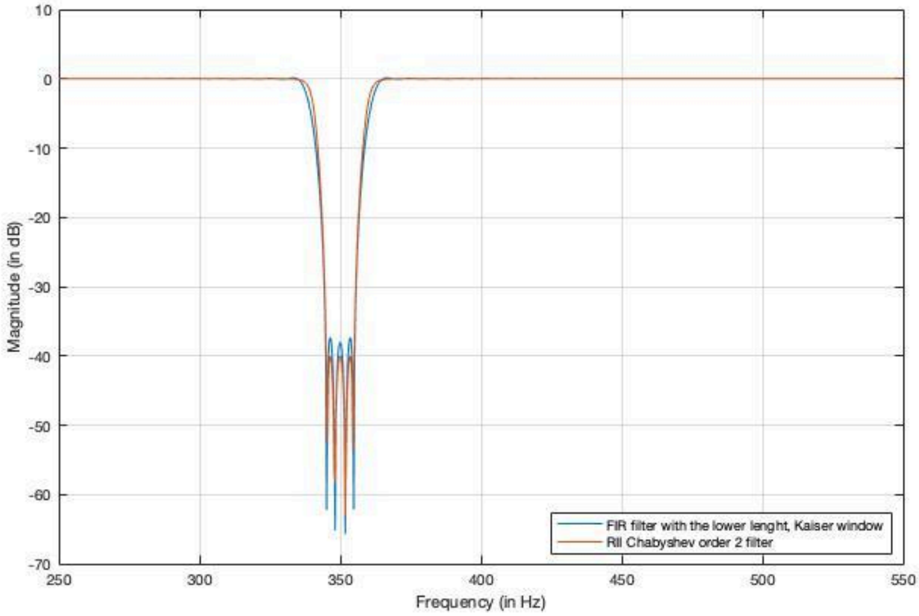




FIR and RII filters comparison

```
figure(80), hold off
plot(f,20*log10(abs(HdFIR))), hold on %Frequency response of the FIR
 filter
plot(f,20*log10(abs(HdChabyshev2))) %Frequency response of the RII
 filter
```

```
legend('FIR filter with the lower lenght, Kaiser window','RII
 Chabyshev order 2 filter','location','southeast'), grid on
xlabel('Frequency (in Hz)'); ylabel('Magnitude (in dB)');
set(figure(80),'name','FIR and RII filter magnitude frequency
 responses comparison','position',[hpixels/2 vpixels/2 hpixels/2
 vpixels/2])
```



Published with MATLAB® R2020a