23-10-2015

Digital Filters

Practice 3

Mariano Orozco García
PROCESAMIENTO DE SEÑALES BIOLÓGICAS, 3BM4
Tovar Corona Blanca

Procesamiento de señales biológicas
3BM4
Tovar Corona Blanca
Orozco García Mariano
23th October, 2015

Practice 3 – Digital Filters

INTRODUCTION:

A digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. It is a computation which takes the input signal and produces the filtered output signal. Digital filters may be more expensive than an equivalent analog filter due to their increased complexity, but they make practical many designs that are impractical or impossible as analog filters. When used in the context of real-time analog systems, digital filters sometimes have problematic latency (the difference in time between the input and the response) due to the associated analog-to-digital and digital-to-analog conversions and anti-aliasing filters, or due to other delays in their implementation.

Bilinear transformation:

The bilinear transformation is a mathematical mapping of variables. In digital filtering, it is a standard method of mapping the "s" or analog plane into the "z" or digital plane. It transforms analog filters, designed using classical filter design techniques, into their discrete equivalents. Mathematically the bilinear transformation is expressed:

$$s = c\frac{1 - z^{-1}}{1 + z^{-1}}, \quad c > 0,\ c = 2/T$$

FIR & IIR filters:

The Finite Impulse Result (FIR) filters are named like this because when you use an impulse (Dirac's Delta) as an input the output will stop being non-zero when the delay is over. In contrast, the Infinite Impulse Result (IIR) filters will be infinitely non-zero (at least theoretically).

IIR filters can achieve a given filtering characteristic using less memory and calculations than a similar FIR filter. IIR Filters are more susceptible to problems of finite-length arithmetic, such as noise generated by calculations, and limit cycles. They are slower to implement using fixed-point arithmetic.

FIR filters are usually linear-phase, this means that this filters delay the input signal but don't distort its phase. They are simple to implement mostly by looping a single instruction. They are suited for either "decimation" (reducing the sampling rate), "interpolation" (increasing the sampling rate), or both.

GENERAL PROPUSE:

To understand, differentiate and apply the different kind of digital filters knowing the advantages and disadvantages of each type and recognizing differences and similarities in comparison with analog filters.

PROCEDURE:

The current practice is divided into two main sections. In the first part we deal with a band pass filter that is based in an analog filter. This band pass filter is designed with a low-pass and a high-pass filter in cascade to generate a 1 Hz to 150 Hz bandwidth. From the analog design we use the bilinear transformation to reach the transfer function in the z domain that we can see in the attached sheet with the calculus. Then we analyze the signals with a Bode diagram as we can see in the results section. Finally this filter is used in a ECG signal from PhysioNet and a sum of sine waves. This is also discussed later in the results section.

The second part of the practice compares the IIR and FIR filters. Fist we design four different configurations of low-pass filters with a cut-off frequency of 60 Hz and third order. Then we choose five different FIR filters and compare the behavior between all of them. Finally we choose, design and apply a filter to observe a specific characteristic of a bio-signal recovered from PhysioNet. This can be seen in the results section.

To implement this practice we use the following code:

```
function practice3() % Digital filters

%% Part I: Design of a band-pass filter from an analog filter
% Obtain the bode diagrams:
    Fs = 1500;
    n = 4096; % Number of points to be ploted
    figure
        freqz([0.3249,0.3249],[1.3249,-0.6751],n,Fs)
        title('Low pass 150Hz filter')
    figure
        freqz([1,-1],[1.0021,-0.9979],n,Fs)
        title('High pass 1Hz filter')
    B = [0.3249,0,-0.3249];
    A = [1.3277,-1.9986,0.6737];
    figure
        freqz(B,A,n,Fs)
        title('Band pass 1-150Hz filter')
% Select an ECG from PhysioNet and apply your filter to it:
    [time,signal,~,~] = readPhysionet('ecg1');
    fsignal = filter(B,A,signal);
    figure
        subplot(2,1,1)
            plot(time,signal)
            title('Original ECG signal')
            xlabel('Time (s)')
            ylabel('Amplitude')
            xlim([0,2])
            grid on
        subplot(2,1,2)
            plot(time,fsignal)
            title('Filtered ECG signal')
            xlabel('Time (s)')
```

```
            ylabel('Amplitude')
            xlim([0,2])
            grid on
% Test your filter with a sum of sine-waves to observe its behavior:
    F = [1,10,50,100,150,200];
    [t,signal2] = sumSines(F,Fs);
    fsignal2 = filter(B,A,signal2);
    [f2,S2] = fourierVector(signal2,Fs);
    [ff2,FS2] = fourierVector(fsignal2,Fs);
    figure
        subplot(2,2,1)
            plot(t,signal2)
            title('Original sum of sines signal')
            xlabel('Time (s)')
            ylabel('Amplitude')
            xlim([0,0.2])
            grid on
        subplot(2,2,2)
            plot(t,fsignal2)
            title('Filtered sum of sines signal')
            xlabel('Time (s)')
            ylabel('Amplitude')
            xlim([0,0.02])
            grid on
        subplot(2,2,3)
            plot(f2,S2)
            title('Frequency domain original signal')
            xlabel('Frequency (Hz)')
            ylabel('Amplitude')
            xlim([0,250])
            grid on
        subplot(2,2,4)
            plot(ff2,FS2)
            title('Frequency domain filtered signal')
            xlabel('Frequency (Hz)')
            ylabel('Amplitude')
            xlim([0,250])
            grid on

%% Part II: IIR and FIR filters
% Design a low pass filter in the 4 diferent configurations (Butterwooth,
Cheby I, Cheby II, Elliptic) fc = 60 Hz, order 3.
    fc = 60;
    fs = 600;
    [b,a] = butter(3,fc/(fs/2));
    figure
        freqz(b,a,n,fs)
        title('Butter 60 Hz low pass filter')
    [b,a] = cheby1(3,3,fc/(fs/2));
    figure
        freqz(b,a,n,fs)
        title('Cheby1 60 Hz low pass filter 3 dB Bandpass Ripple')
    [b,a] = cheby2(3,30,fc/(fs/2));
    figure
        freqz(b,a,n,fs)
```

```
        title('Cheby2 60 Hz low pass filter 30 dB StopBand Ripple')
    [b,a] = ellip(3,3,30,fc/(fs/2));
    figure
        freqz(b,a,n,fs)
        title('Elliptic 60 Hz low pass 3db Bandpass and 30dB Stopband Ripple')
% Choose 5 different FIR filters and compare their behaviour.
    [b,a] = fir1(6,fc/(fs/2),hamming(7));
    figure
        freqz(b,a,n,fs)
        title('FIR filter Hamming window')
    [b,a] = fir1(6,fc/(fs/2),hann(7));
    figure
        freqz(b,a,n,fs)
        title('FIR filter Hann window')
    [b,a] = fir1(6,fc/(fs/2),gausswin(7));
    figure
        freqz(b,a,n,fs)
        title('FIR filter Gauss window')
    [b,a] = fir1(6,fc/(fs/2),kaiser(7));
    figure
        freqz(b,a,n,fs)
        title('FIR filter Kaiser window')
    [b,a] = fir1(8,fc/(fs/2),blackmanharris(9));
    figure
        freqz(b,a,n,fs)
        title('FIR filter Blackman-Harris window')
% Select a bio-signal from PhysioNet and design a digital filter to observe a
specific characteristic of your signal. Explain why did you choose that filter
    [b,a] = fir1(15,30/(1000/2),blackmanharris(16));
        figure
        freqz(b,a,4095,1000)
    title('FIR filter Blackman-Harris window for abdominal (fetus) ECG')
    [time,signals,~,~] = readPhysionet('f_a_ecg');
    signal3 = signals(:,2);
    fsignal3 = filter(b,a,signal3);
    figure
    subplot(2,1,1)
        plot(time,signal3)
        title('Original abdominal (fetus) ECG signal')
        xlabel('Time (s)')
        ylabel('Amplitude')
        xlim([0,1])
        ylim([-70,30])
        grid on
    subplot(2,1,2)
        plot(time,fsignal3)
        title('Filtered abdominal (fetus) ECG signal')
        xlabel('Time (s)')
        ylabel('Amplitude')
        xlim([0,1])
        ylim([-70,30])
        grid on
end
```

In the code we include the following complementary custom functions:

```
function [time,val,labels] = readPhysionet(Name)
% This function reads a pair of files (Name.mat and Name.info) from a
% PhysioBank record, baseline-corrects and scales the time series contained
% in the .mat file.  The baseline-corrected and scaled time series are the
% rows of matrix 'val', and each column contains simultaneous samples of
% each time series.

% Read mat File:
    matName = strcat(Name, '.mat');
    load(matName);
    n = size(val,1);
% Read info File:
    infoName = strcat(Name, '.info');
    fid = fopen(infoName, 'rt');
    fgetl(fid);
    fgetl(fid);
    fgetl(fid);
    freqint = sscanf(fgetl(fid), 'Sampling frequency: %f Hz  Sampling
interval: %f sec');
    interval = freqint(2);
    fgetl(fid);
% Read data of each signal
    signal = cell(1,n);
    gain = zeros(1,n);
    base = zeros(1,n);
    units = cell(1,n);
    for i = 1:n
      [~, signal(i), gain(i), base(i), units(i)] =
strread(fgetl(fid),'%d%s%f%f%s','delimiter','\t');
    end
    fclose(fid);

% Baseline-corrects and scales the time series:
    val(val==-32768) = NaN;
    for i = 1:n
        val(i, :) = (val(i, :) - base(i)) / gain(i);
    end
    time = (1:size(val, 2)) * interval;
    val = val';

% Gives information of each signal:
    labels = cell(1,length(signal));
    for i = 1:length(signal)
        labels{i} = strcat(signal{i}, ' (', units{i}, ')');
    end

end
```

```
function [f,X] = fourierVector(x,Fs)
% To obtain absolute positive values of the fourier transform of x
% it also obtains the corresponding frequency vector
    X = fft(x);
    Ie = floor(length(X)/2);
    X = abs(X(:,1:Ie))/length(X);
    f = (0:Ie-1)*((Fs/2)/(Ie-1));
end

function [t,x] = sumSines(F,Fs)
    t = 0:1/Fs:1-1/Fs;
    x = 0*t;
    for i = 1:length(F)
        x = x + sin(2*pi*F(i)*t);
    end
end
```

RESULTS:

In the first section we design the following analog filters that are simulated and show the cut-off frequencies at 1 Hz and 150 Hz.

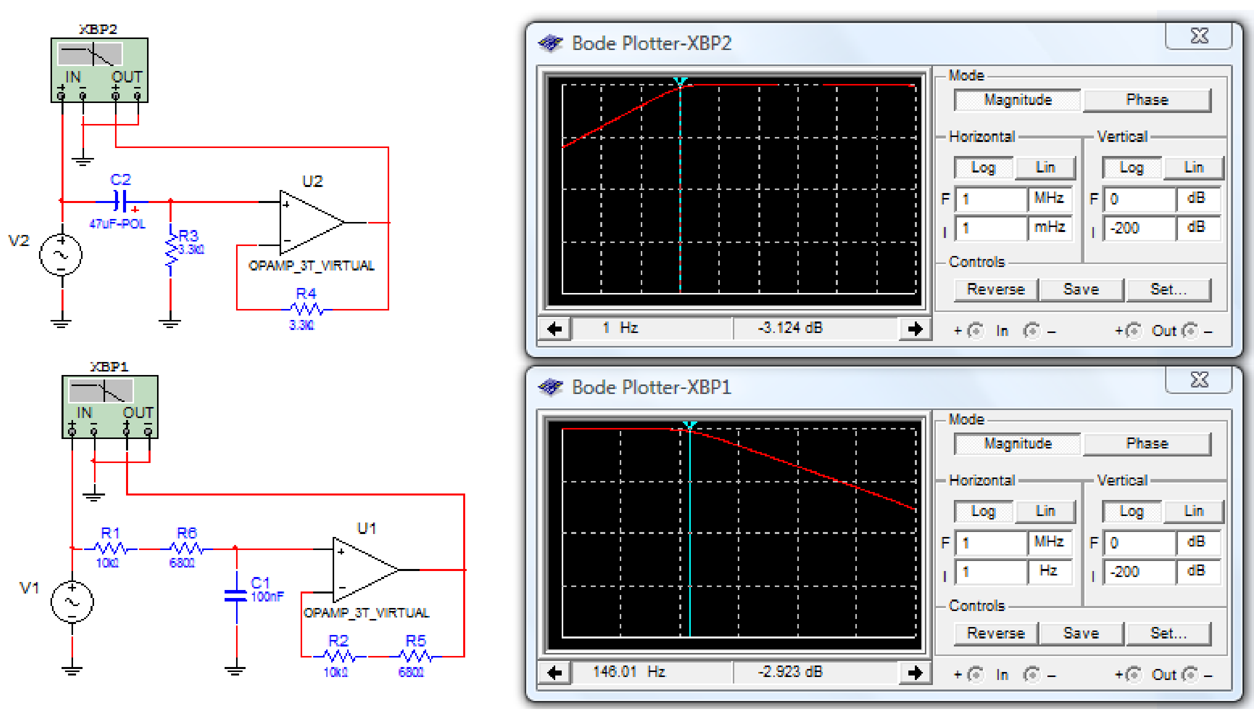


Figure1. *Electric diagrams and cut-off frequencies of Low pass (down) and High pass (up) filters.*

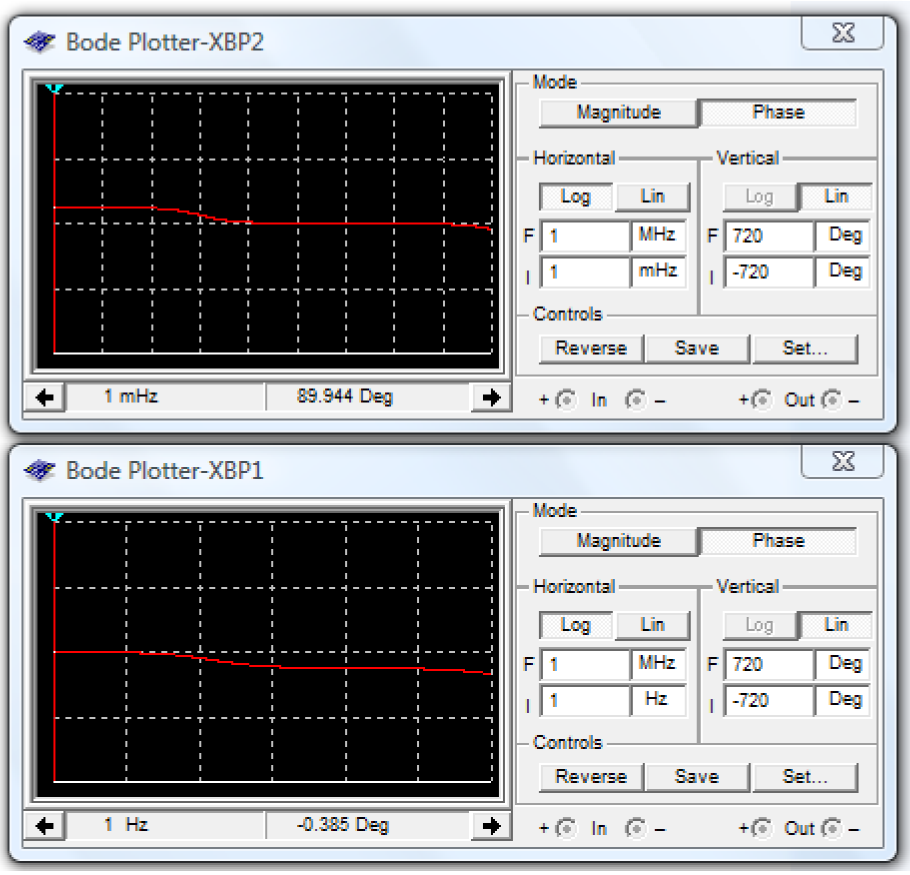


Figure2. *Bode plot in phase modes of Low pas (down) s and High pass (up) filters.*

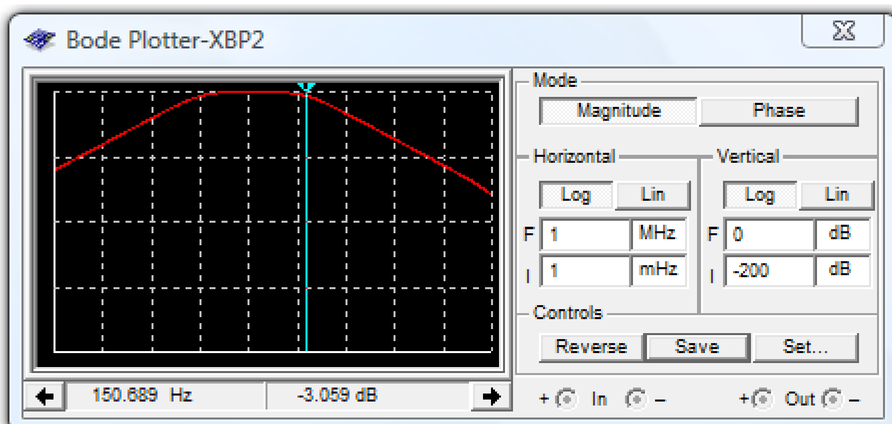


Figure3. *Bode plot in magnitude mode showing low pass cut off frequency.*

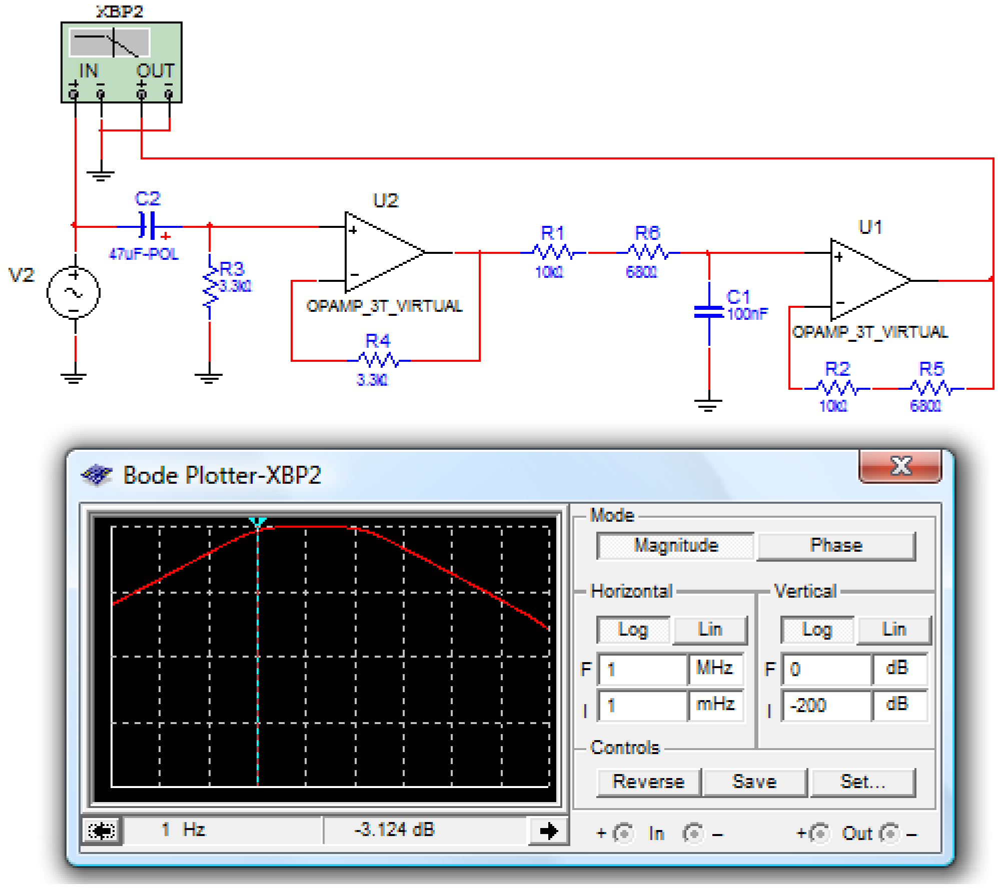


***Figure4. Diagram and B**ode plot in magnitude mode showing high pass cut-off frequency.*

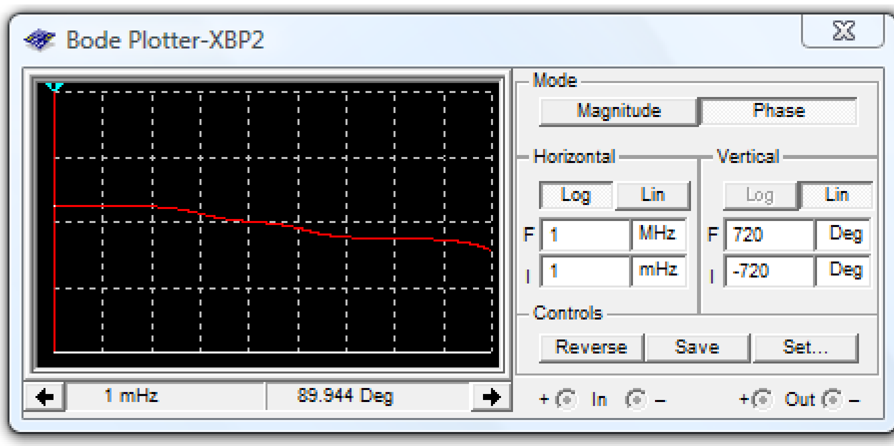
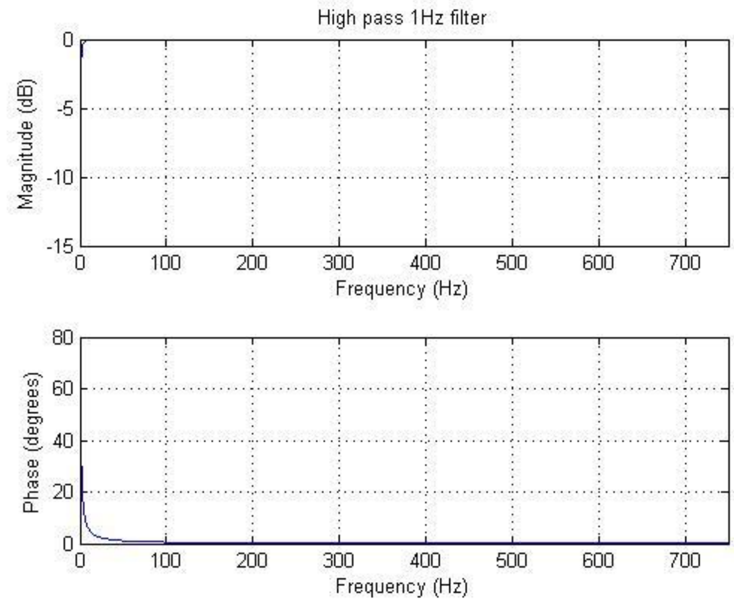
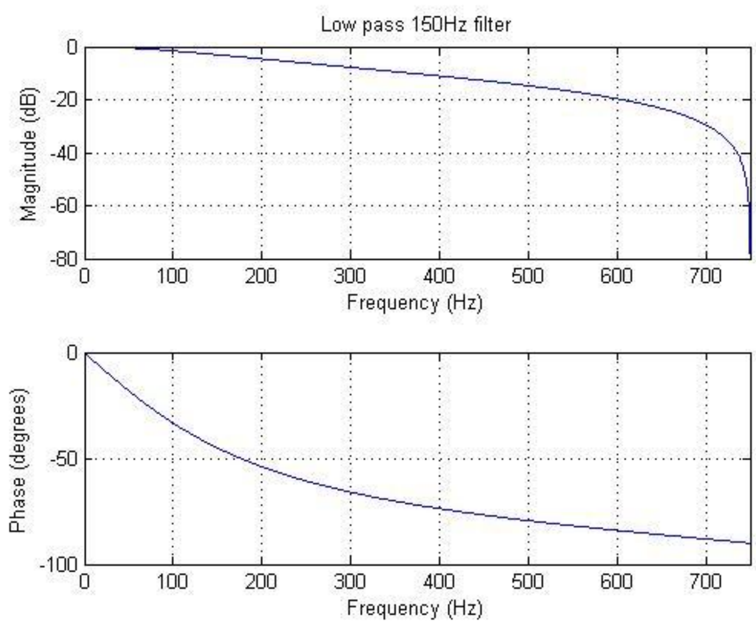


***Figure5.** Bode plot in phase mode of Band pass filter.*

Then we analyze the signals with a Bode diagram of the digital filter.



Figur6. *Bode plot of digital High pass filter.*



Figur6. *Bode plot of digital Low pass filter.*

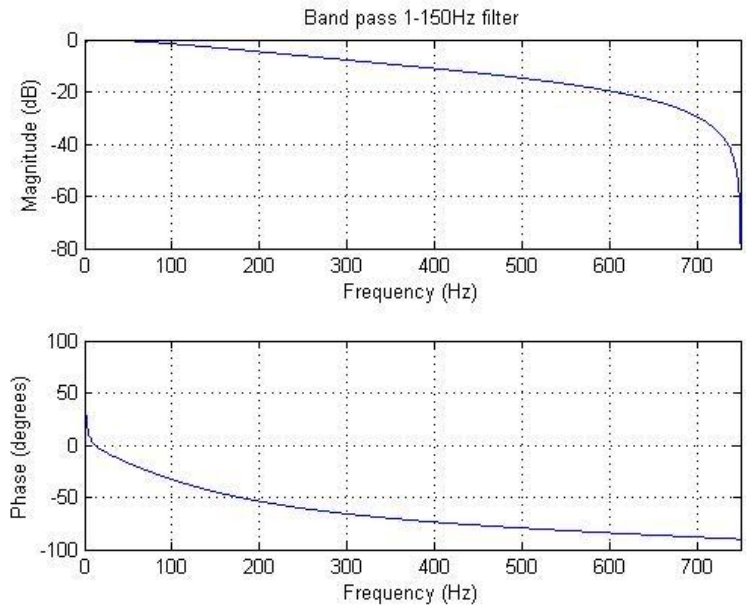


Figure7. *Bode plot in o digital Band pass filter.*

Finally this are the results of the filter applied in a ECG signal from PhysioNet and a sum of sine waves.

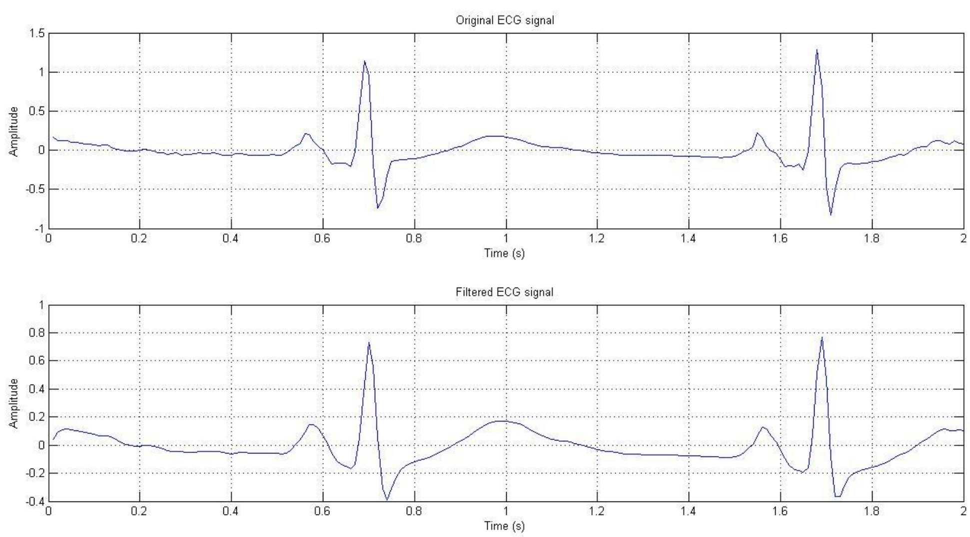


Figure8. *ECG signal before and after filter.*

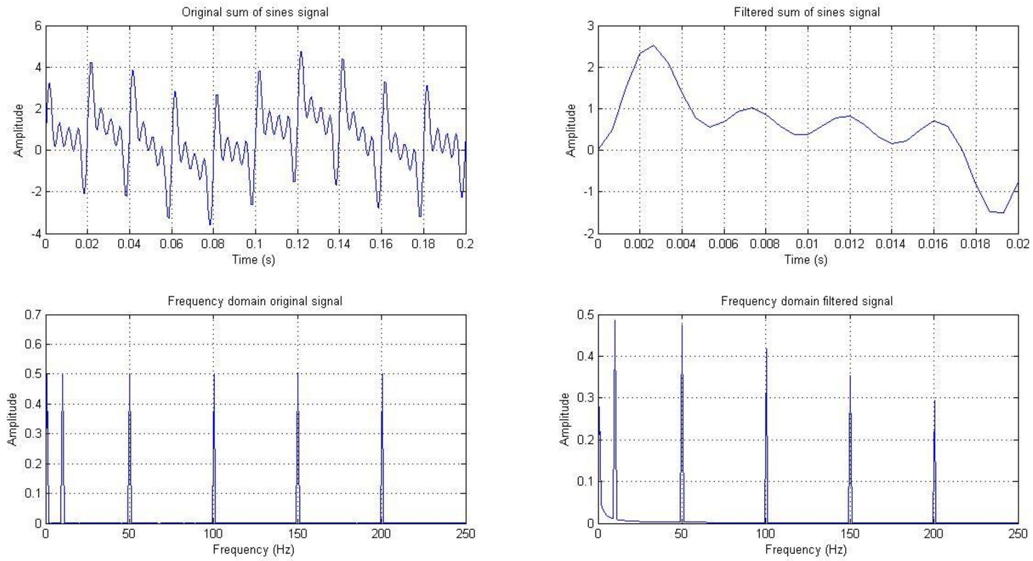


Figure9. *Sum of sine wave before and after filter in time and frequency domain.*

The results of the second part of the practice begin with the Bode plots of low-pass filters with a cut-off frequency of 60 Hz and third order.

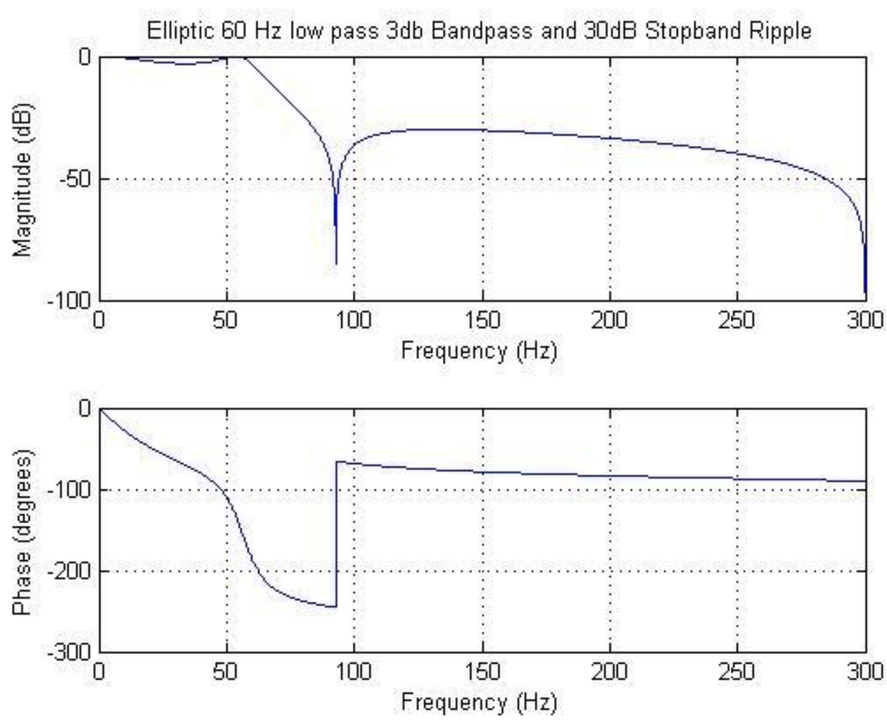


Figure10. *Bode plot of Elliptic filter.*

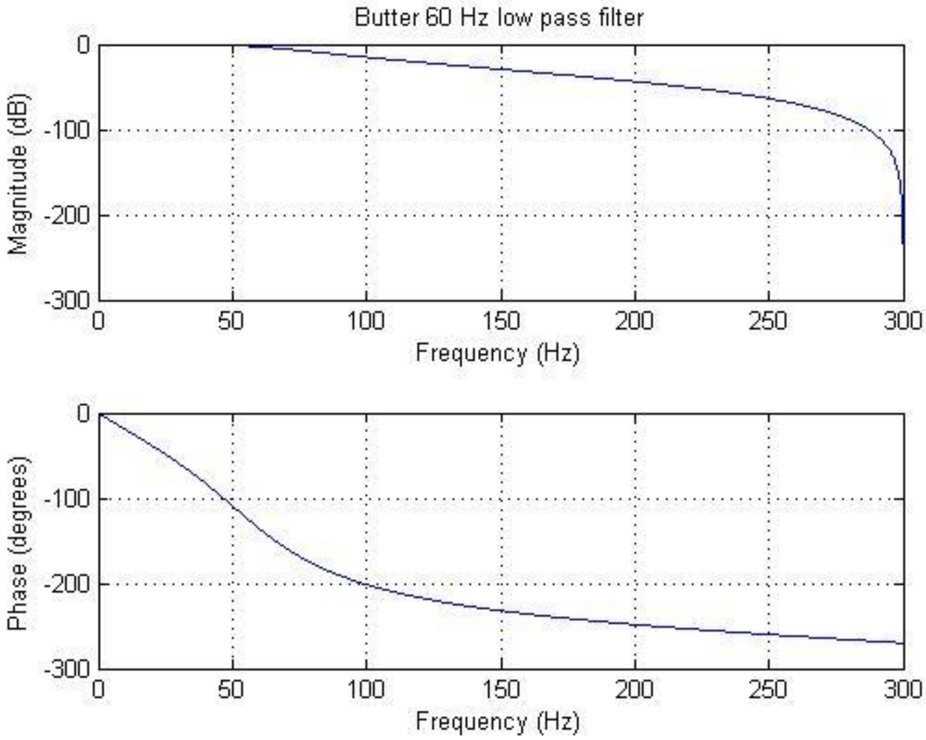


Figure11. *Bode plot of Butterwood filter.*

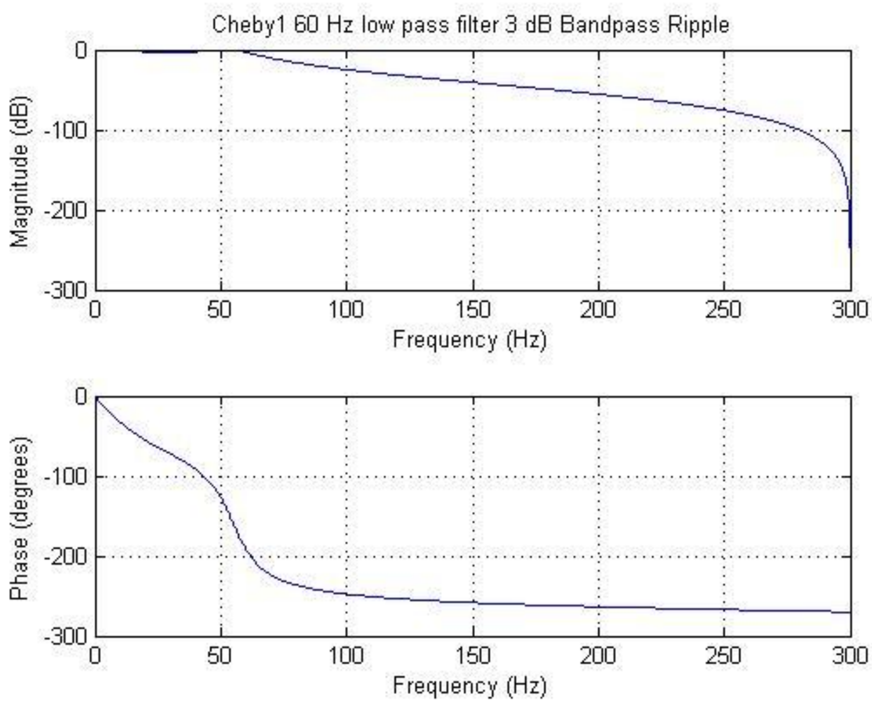


Figure12. *Bode plot of Cheby1 filter.*

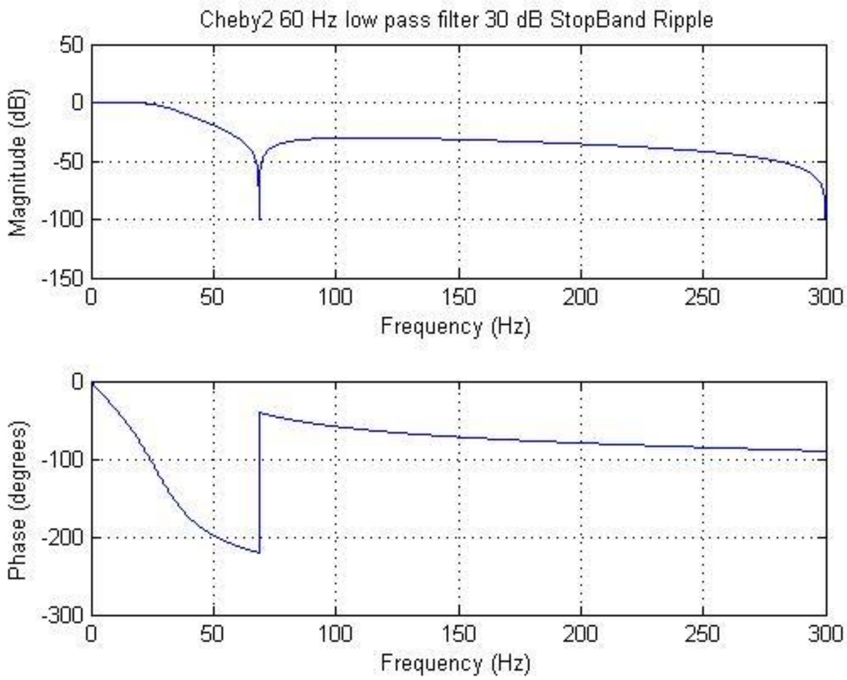


Figure13. *Bode plot of Cheby2 filter.*

Then we show the bode plots of five different FIR filters.

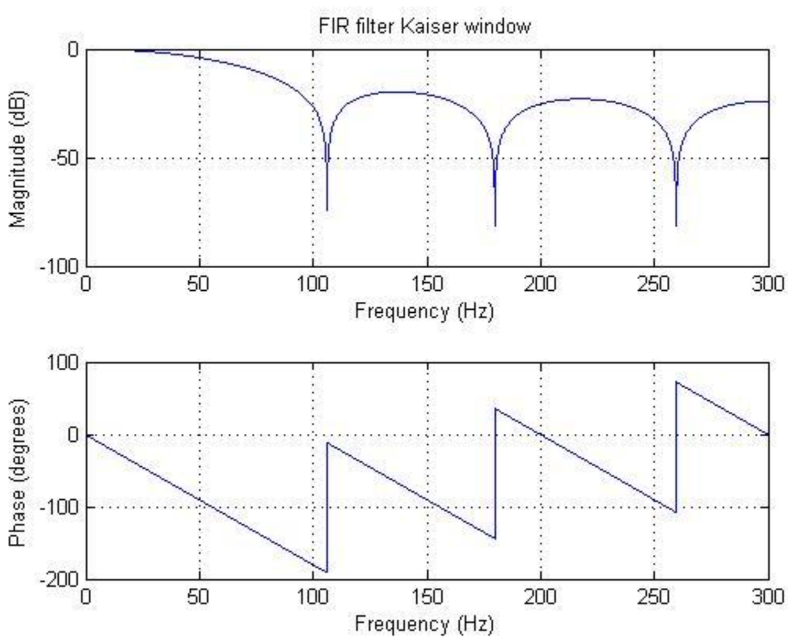


Figure14. *Bode plot of Kaiser filter.*

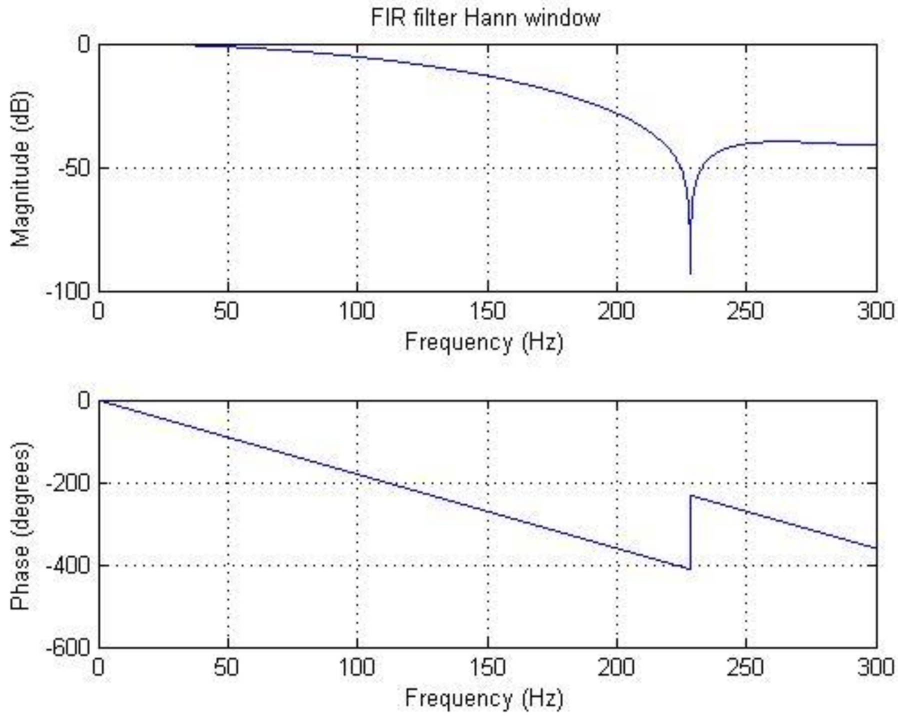


Figure15. *Bode plot of Hann filter.*

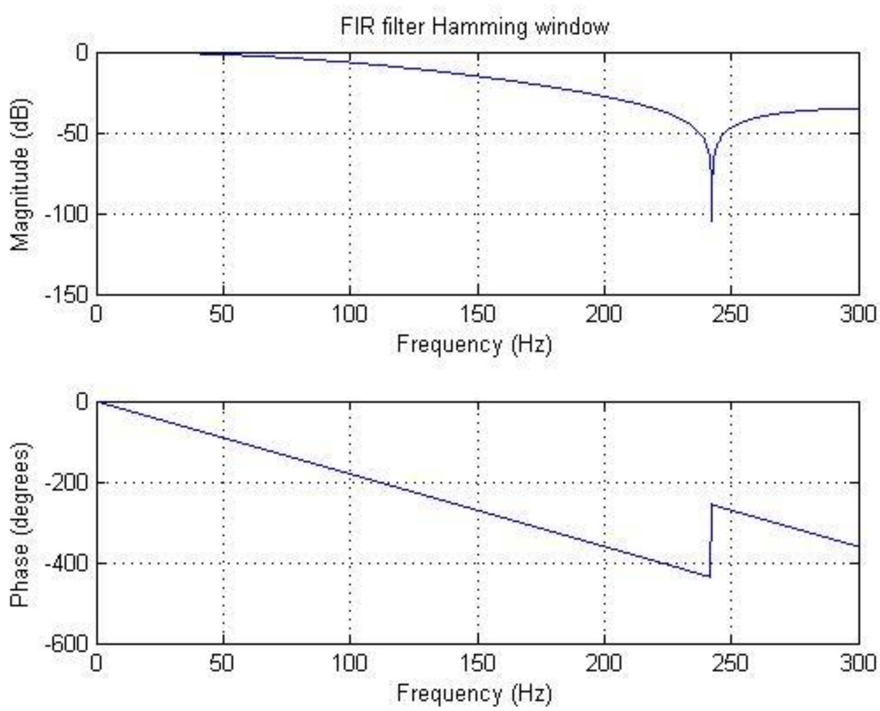


Figure16. *Bode plot of Hamming filter.*

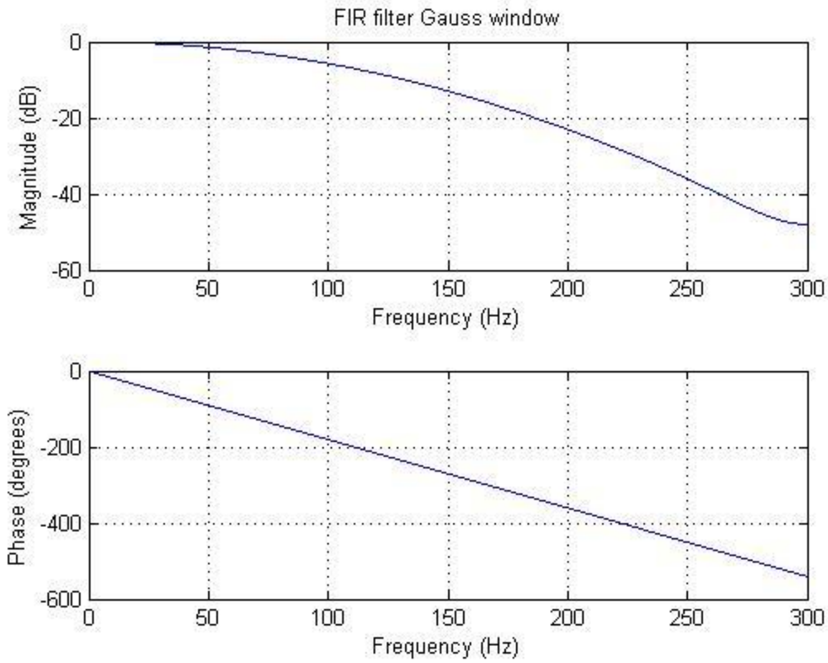


Figure16. *Bode plot of Gauss filter.*

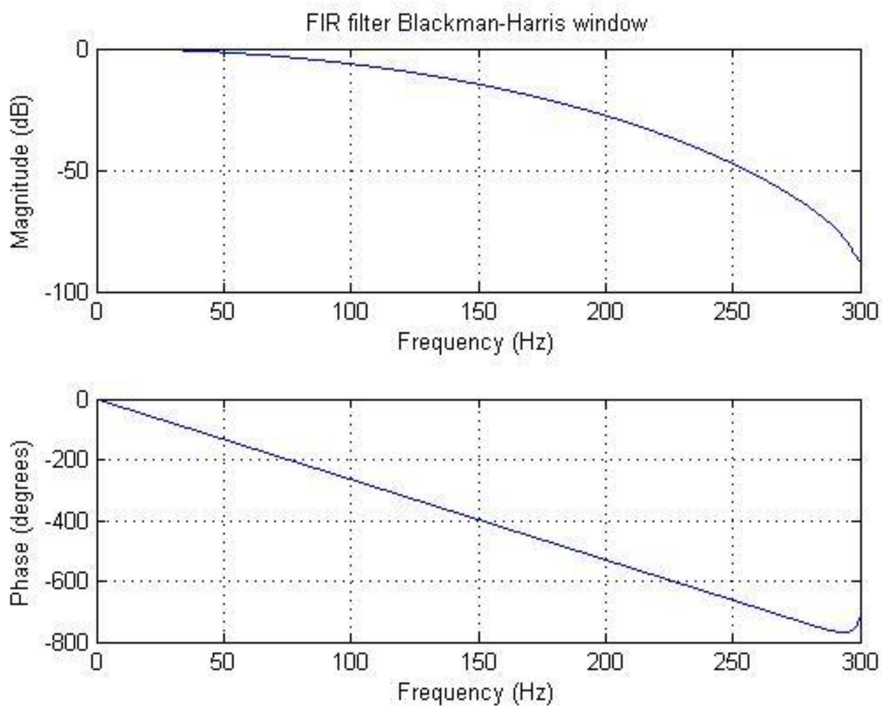


Figure18. *Bode plot of Balckman-Harris filter.*

Finally we show the filter that we use in the bio-signal recovered from PhysioNet and the results of the filtered signal.

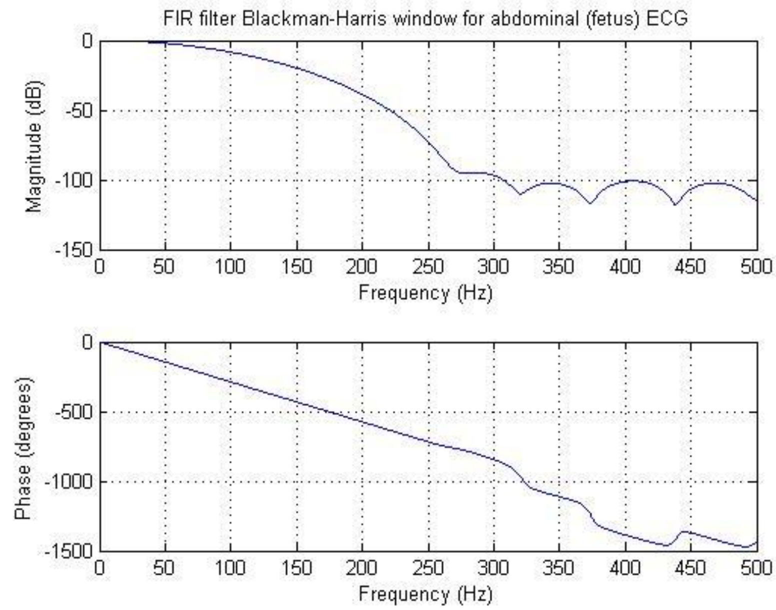


Figure19. *Bode plot of Balckman-Harris filter for abdominal (fetus) ECG.*

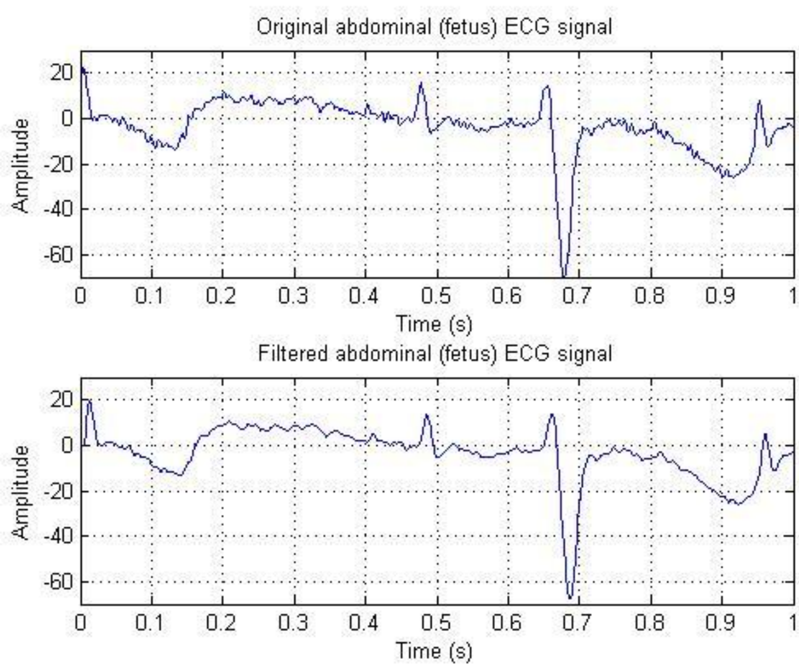


Figure20. *Abdominal ECG before and after filter.*

CONCLUSION:

In this practice we could see that we have a wide variety of options to filter a digital signal. Its crucial to understand the characteristics of each type to make an adequate filtering process.

We saw that the Blackman-Harris window have a Stop Band Ripple that is minimum in the case of the signal that we had to process as well as a linear phase in the range of our interest. That was the main reason of choosing this filter for the last signal. We also notice that the slope was adecuate and the cut-off Frequencies are not in the exact place.

We saw the differences of having the Stop Band Ripple and Band Pass Ripple that we don't have in the analog filters and how they affect the magnitude and the phase of the signal.

It's also important to notice that a "good filter" not depends in the filter itself, so there is no superior filter in the general sense but it depends in the results we want to obtain and the characteristics we want to extract form a signal.

REFERENCES:

[1] Julius O. Smith III, Introduction to digital filters with audio applications, Center for Computer Research in Music and Acoustics [CCRMA], Standford University, 2007. Recovered from: https://ccrma.stanford.edu/~jos/filters/ [24/10/15].

[2] Julius O. Smith III, Physical audio signal processing for virtual musical instruments and audio effects, Center for Computer Research in Music and Acoustics [CCRMA], Standford University, 2010. https://ccrma.stanford.edu/~jos/pasp/Bilinear_Transformation.html [24/10/15].

[3] FIR Filter Basics, IIR Filter Basics, dspGuru, lorwegian International. Recovered from: http://dspguru.com/dsp/faqs/fir/basics [24/10/15].

[4] IIR Filter Basics, IIR Filter Basics, dspGuru, lorwegian International. Recovered from: http://dspguru.com/dsp/faqs/iir/basics [24/10/15].

[5] Practical Introduction to Digital Filtering, Mathworks. Recovered from: http://www.mathworks.com/help/signal/examples/practical-introduction-to-digital-filtering.html [24/10/15].

[6] Practical Introduction to Digital Filter Design, Matchworks. Receovered from: http://www.mathworks.com/help/signal/examples/practical-introduction-to-digital-filter-design.html [24/10/15].

[7] Bilinear, Mathworks. Recovered from: http://www.mathworks.com/help/signal/ref/bilinear.html [24/10/15].