Using cardinal sinus signal to generate FIR filters

Samuel Dupont
Samuel.dupont.etu@univ-lemans.fr

November 4, 2016

Abstract

A recap to generate FIR filter coefficients using cardinal sinus. It encompasses different filters : low-pass, high pass and bandpass. A such method is useful to generate the coefficients in embedded system in a versatile way, the program calculating the coefficients automatically. However this method as the draw back of necessitating a great number of points which lead to a slow paced filter.

I. Introduction

It's quite easy to understand that the ideal filter in frequency term is a step such that from the drop of the step there is no more signal.
It's well known in signal processing that the Fourier transform of a step or rectangular window is a cardinal sinus (sinc), in the other way around a *sinc* in the time domain gives a rectangular shape in the Fourier domain, if written in equation it gives the equation 1.

$$h(t) = F^{-1}\{H(f)\} = \frac{sin(\pi f_c t)}{\pi t} = f_c \; sinc(\pi f_c t), \tag{1}$$

with,

$$H(f) = rect(\frac{f}{f_c}), \tag{2}$$

the normalised cut off frequency denoted as:

$$f_c = \frac{2f_c}{f_s}, \tag{3}$$

f_s being the sampling frequency, f the frequency, t the index of the abscissa.
An illustration of the cardinal sinus can be seen on figure 1.

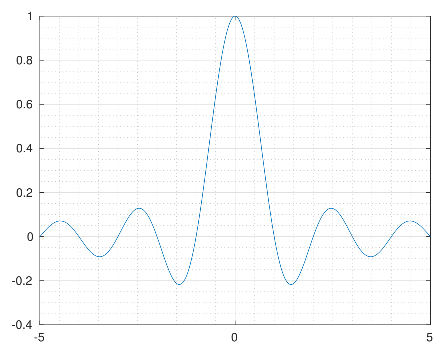

Figure 1: *Normalised cardinal sinus,* $f_c = 1$

II. Methods

In order to implement FIR filter it's required to sample and window it as it's not possible to have an infinite number of coefficients. This lead in the discrete world with all the problems of aliasing, maximum frequency...

The number of coefficients in the filter is often critical as the more there is the slower will be the response time, but if there is not enough the frequency response will be badly impacted (further explanation in section V.i) .

i. FIR filter implementation

The figure 2 shows the weights coefficients of a 31 points low-pass filter obtained from equation 1.
It can be seen that -6dB is attained (0.5) for the desired cut of frequency. The linear frequency response highlight the ripples which are generated by the truncation of the infinite *sinc* to a finite number.
It's possible to reduce this ripple using by windowing the signal.

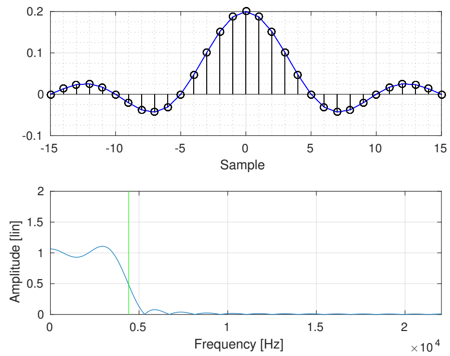

Figure 2: *Low-pass 31 coefficients for 4410 Hz obtained cardinal sinus and its frequency response,* $f_c = 0.2 = \frac{4410*2}{44100}$. *The green line shows the cut off frequency.*

ii. Windowing

In order to prevent the ripple induced by a low number of coefficient it's possible to window it. The Blackman window is took for example defined by the equation:

$$0.452 - 0.5 * cos(\frac{2\pi n}{N}) + 0.08cos(\frac{4\pi n}{N}), \quad (4)$$

where n is the index value of the filter coefficients and N the number of coefficients.

The figure 4 shows the previous filter and its Blackman windowed version. It can be seen that the ripple are no more present in th windowed version, the roll-off is however more fast.

There are several different types of windows, the most common are introduced in table 1, the Figure 4 is presenting their different shape, and the Figure 5 their impact on the low-pass frequency response. It can be observed from the last Figure that each windows reduce the ripple but have different roll off.

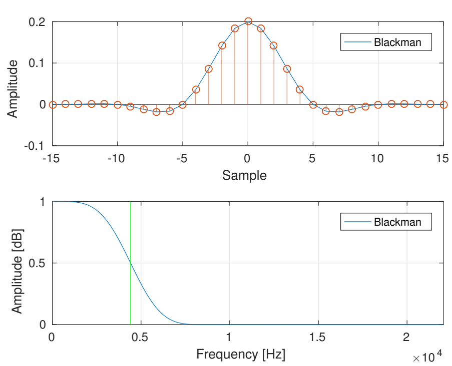

Figure 3: *Low-pass 31 coefficients for 4410 Hz obtained cardinal sinus and its frequency response,* $f_c = 0.2$, *with Blackman window.*

Bartlett	$w(n) = 1 - \frac{2\lvert n-\frac{N}{2}\rvert}{N}$
Blackman	$w(n) = 0.452 - 0.5cos(\frac{2\pi n}{N})$ $+0.08cos(\frac{4\pi n}{N})$
Hamming	$w(n) = 0.54 - 0.456cos(\frac{2\pi n}{N})$
Hanning	$w(n) = 0.5 - 0.5cos(\frac{2\pi n}{N})$
Rectangular	$w(n) = 1$

Table 1: *Different common windows*

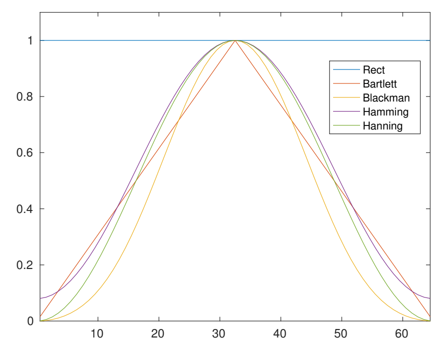

Figure 4: *Different windows in time domain.*

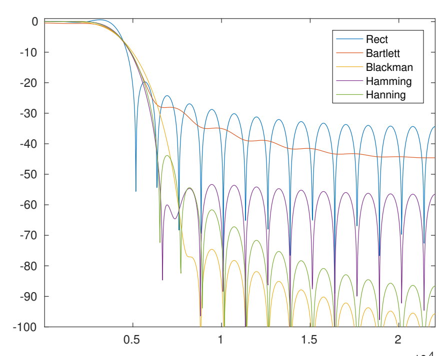

Figure 5: *Different windows applied to a low-pass filter* $f_c = 0.2$.

III. Other kind of Filters

i. High-pass filter

The high-pass filter can be constructed using the low-pass filter. Indeed it just needed to let pass all the frequencies that are stopped by the low-pass and stop the ones that it let pass. In order to achieve this result it's used the spectral inversion, in equation :

$$H_{hp}[f] = 1 - H_{lp}[f], \tag{5}$$

now using the inverse Fourier transform:

$$F^{-1}\{H_{hp}[f]\} = \sigma[n] - H_{lp}[n], \tag{6}$$

where $\sigma[n]$ is the Dirac impulse.

In the end in order to implement the high-pass filter it needed to inverse the sign of the low-pass and to set the center of the filter to $1 - h_{lp}[n_c]$.
The equation is written as follow:

$$h_{hp}[n] = \begin{cases} -f_c \; sinc(\pi f_c n) \text{ if } n \neq \dfrac{N}{2} \\ 1 - f_c \text{ if } n = \dfrac{N}{2} \end{cases} \tag{7}$$

The figure 6 shows the time and frequency domain of a high pass filter. In the time domain the spectral inversion is easily observable by it's peak at 0.

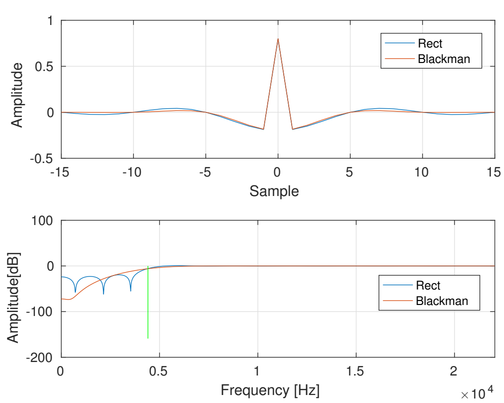

Figure 6: *High-pass filter 31 coefficients $f_c = 0.2$, with and without Blackman windows.*

ii. Bandpass filter

The bandpass filter is constructed from a high-pass filter and a low-pass filter. The equation can be written as follow:

$$h_{bp}[n] = f_{ch} \; sinc(\pi f_{ch} n) - f_{cl} \; sinc(\pi f_{cl} n), \tag{8}$$

where f_{cl} is the low cut off frequency ans f_{ch} is the high cut off frequency. The figure 7 show a bandpass produced.

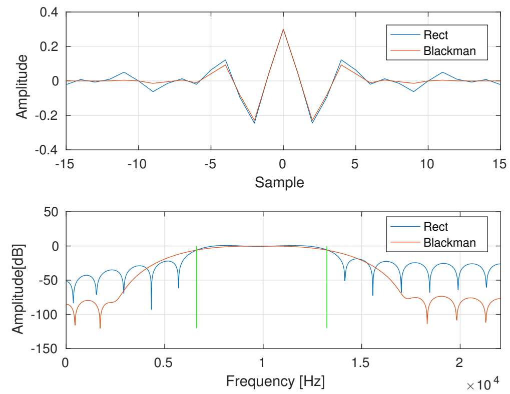

Figure 7: *Band pass filter 31 coefficients $f_{c1} = 0.3$, $f_{c2} = 0.6$, with and without Blackman windows.*

IV. Other kind of transformations

We have seen before that the spectral inversion consist in applying a Dirac in the middle of the filter and inverting the sign of the filter. In the same way, it exists other kind of simple transformations which can applied to the filter. It will be seen the spectral shifting and the spectral mirroring.

In this part the transformation will be applied on the low pass showed on figure 8, $f_c = 0.2$ 4410 Hz for a sampling frequency of 44100 Hz and 31 coefficients.

i. Spectral shifting

The spectral shifting is obtained by multiplying the filter :

$$w[n] = cos(n\pi shift), \tag{9}$$

where the $shift$ is the normalised frequency (ex: 4000/Nyquist frequency).
The shift filter is presented figure9. This effect shift the negative frequencies in the positive part of the spectrum, which mean that if

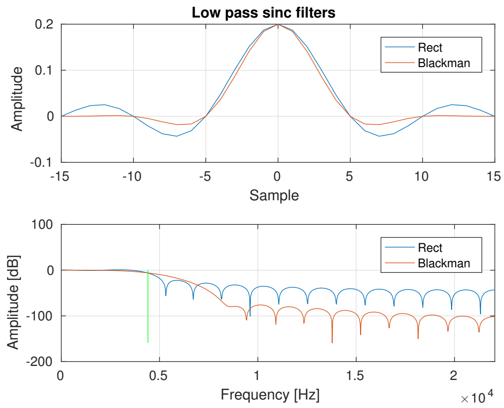

Figure 8: *Low-pass filter 31 coefficients $f_c = 02$, with and without Blackman windows.*

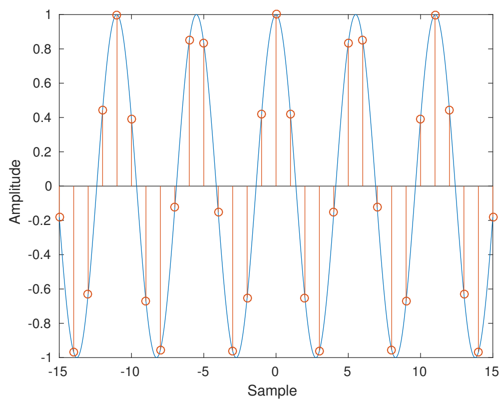

Figure 9: *Shift filter used for 31 coefficients*

the shift is to important the negative cut off will appear in the positive part at $shift - f_c$, as shown in figure 10.

ii. Spectral mirroring

The spectral mirroring is obtained by using a filter consisting of alternated positive and negative impulse as shown on figure 11. The filter applied on the low pass will give a high-pass with a new $f_{cmirror} = 1 - f_c$ as shown on figure 12.

V. Code implementation of the filters

This part will not discuss of code in itself but rather in the thing that it need to be paid attention during the implementation, especially about the number of coefficients influence and

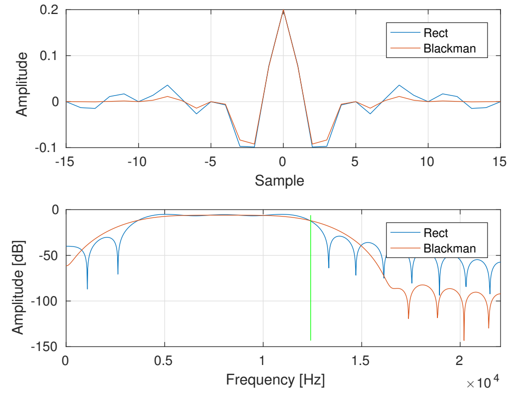

Figure 10: *Low-pass shifted filter 31 coefficients $f_c = 0.2$, with and without Blackman windows,green line at $f_c + 8000 * 2/f_s$.*

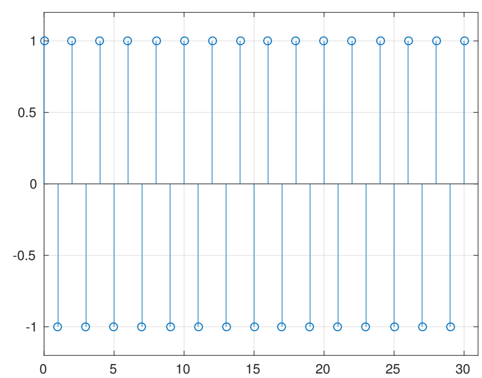

Figure 11: *Mirror filter used for 31 coefficients*

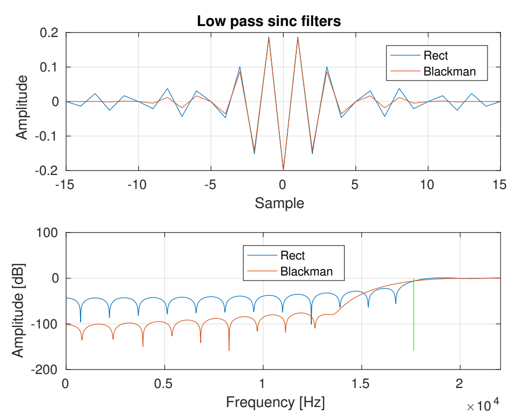

Figure 12: *Low-pass 31 coefs, mirrored with and without Blackman windows. Green line at Nyquist frequency minus $f_c = 0.2$*

how to implement in case of odd or even numbers, what does it imply.

i. Influence of points numbers

The influence of points number is directly linked to the duality time/frequency in signal frequency. In time domain it's obvious that the filter will just be longer if there are ;ore points, the cardinal sinus being truncated with a bigger window...

The influence on the frequency is illustrated by the figure 13, where it can be seen that the more points is it the more accurate will be the rectangle reconstruction in frequency domain. This accuracy in frequency imply a slow filter implementation as numerous operation will be needed to apply this filter to the signal.

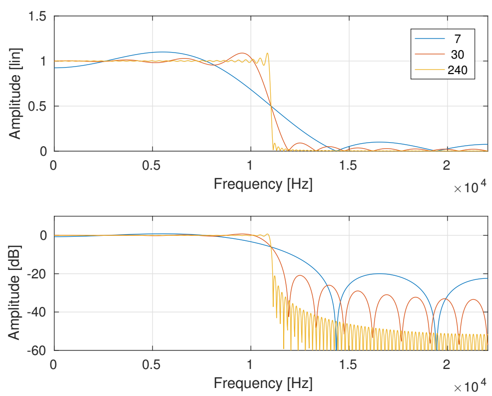

Figure 13: *Linear and dB frequency response of low-pass sinc filters $f_c = 0.5$ with 7,30 and 240 points*

VI. The trap of odd and even points

It's totally possible to implement a odd or an even tap number for the filter, it's just needed to define how the filter will be organised.

For odd number it's quite simple, the center is defined they will be even number on each side. For even number, I choose by analogy to the fft set the center in the left part (as if it's the positive part), thus there will be one more coefficient in the first half as there is the center in the second half.

Ok, nothing difficult until there. But now with the windows to avoid ripple? They need to be aligned to the center of the filter (center=peak of the sinc), else it will slightly damage the filter.
Then how to achieve this? Actually all the previous windows on a ratio of n over N, the trick is just that n at the center is equal to $\frac{N}{2}$ so that the ratio at the center is 0.5:

- for odd number, n must go begin at 0.5 ,
- for even number, n begin a 1 so that the center is well situated at $\frac{N}{2} + 1$.