

Using cardinal sinus signal to generate FIR filters

SAMUEL DUPONT

Samuel.dupont.etu@univ-lemans.fr

November 06, 2016

Last update: November 25, 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. The matlab program used to implement the different FIR filter can be found on the following link: [Link to program](#)

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, as shown on Figure 1.

Then, 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 it gives the equation 1.

$$\begin{aligned} h(t) = F^{-1}\{H(f)\} &= \frac{\sin(\pi f_c t)}{\pi t} \\ &= f_c \operatorname{sinc}(\pi f_c t), \end{aligned} \quad (1)$$

with,

$$H(f) = \operatorname{rect}\left(\frac{f}{f_c}\right), \quad (2)$$

the normalised cut off frequency denoted as:

$$f_c = \frac{2f_c}{f_s}, \quad (3)$$

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 2.

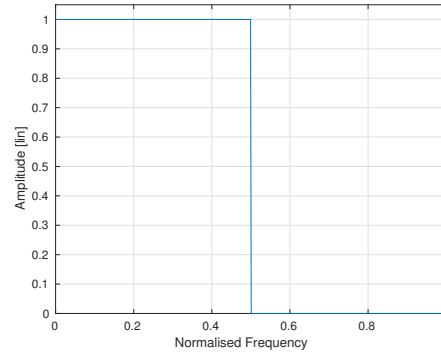


Figure 1: Normalised step, $f_c = 0.5$

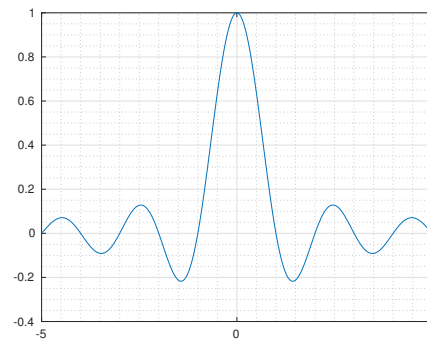


Figure 2: Normalised cardinal sinus, $f_c = 1$

II. METHODS

It's not possible to have an infinite number of coefficients, thus in order to implement FIR

filter it's required to sample and window it. 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 3 shows the weight 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 by windowing the signal.

The filters are shown from $-\frac{N}{2}$ to $\frac{N}{2}$ in order to show the filter in a centred way, however when filtering the first coefficient correspond to $-\frac{N}{2}$. When implementing n goes from 1 to $N + 1$ or 0.5 to $N + 0.5$, if even and odd number of N respectively.

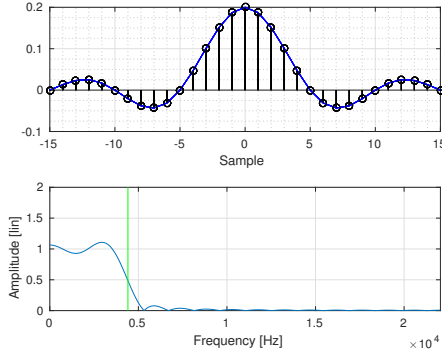


Figure 3: Low-pass, 31 coefficients, obtained from a cardinal sinus and its frequency response, $f_c = 0.2 = \frac{4410 \times 2}{44100}$. The green line shows the cut off frequency.

ii. Windowing

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

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

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

Table 1: Different common windows

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 the windowed version, the roll-off is however more fast.

There are several different types of win-

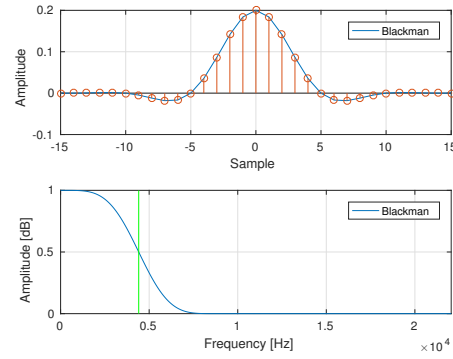


Figure 4: Low-pass 31 coefficients obtained from a cardinal sinus and its frequency response, $f_c = 0.2$, with Blackman window.

dows, the most common are introduced in table 1, the Figure 5 is presenting their different shape, and the Figure 6 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.

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

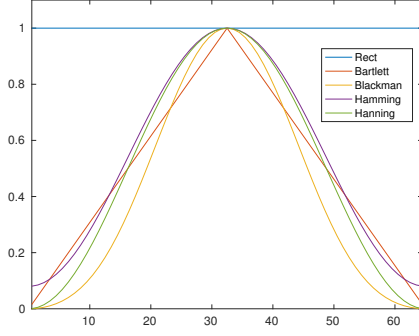


Figure 5: Different windows in time domain.

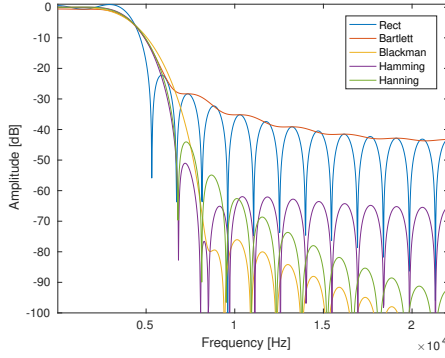


Figure 6: Different windows applied to a low-pass filter $f_c = 0.2$.

low-pass and stop the ones that it lets pass. For the sake of this result it can be used the spectral inversion explained on the following equations, departing from the frequency domain.

$$H_{hp}[f] = 1 - H_{lp}[f], \quad (5)$$

then, using the inverse Fourier transform:

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

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 $h_{hp}[n_c]$ to $1 - h_{lp}[n_c]$.

The equation is written as follow (even number N):

$$h_{hp}[n] = \begin{cases} -f_c \operatorname{sinc}(\pi f_c n) & \text{if } n \neq \frac{N}{2} \\ 1 - f_c & \text{if } n = \frac{N}{2} \end{cases} \quad (7)$$

The figure 7 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.

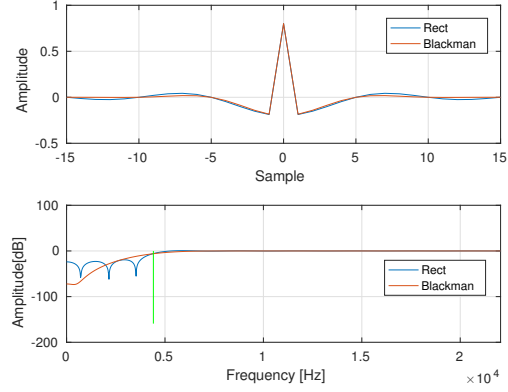


Figure 7: High-pass filter 31 coefficients $f_c = 0.2$, with and without Blackman windows.

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:

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

where f_{cl} is the low cut off frequency and f_{ch} is the high cut off frequency. The Figure 8 show a bandpass produced.

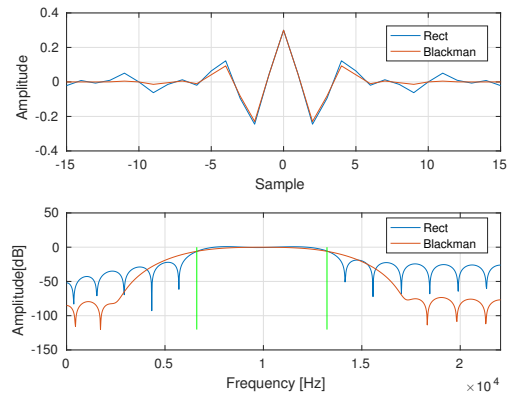


Figure 8: Band pass filter 31 coefficients $f_{c1} = 0.3$, $f_{c2} = 0.6$, with and without Blackman windows.

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 9, $f_c = 0.2$ so 4410 Hz for a sampling frequency of 44100 Hz and 31 coefficients.

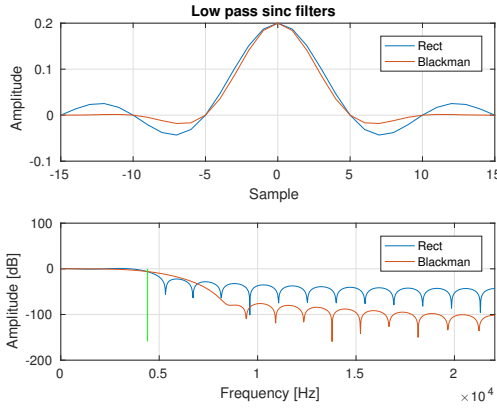


Figure 9: Low-pass filter 31 coefficients $f_c = 0.2$, with and without Blackman windows.

i. Spectral shifting

The spectral shifting is obtained by multiplying the filter by the window :

$$w_s[n] = \cos(n\pi shift), \quad (9)$$

where the *shift* is the normalised frequency (ex: 4000/Nyquist frequency).

The shift filter is presented Figure10. This effect shift the negative frequencies in the positive part of the spectrum, which mean that if the shift is too important the negative cut off will appear in the positive part at $shift - f_c$, as shown in Figure 11.

ii. Spectral mirroring

The spectral mirroring is obtained by using a filter consisting of alternated positive and negative impulse as shown on Figure 12. The filter

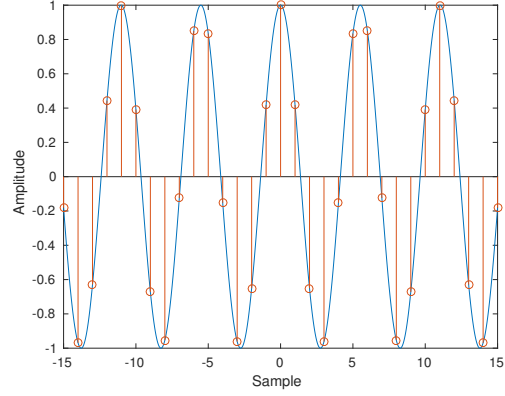


Figure 10: Shift filter used for 31 coefficients

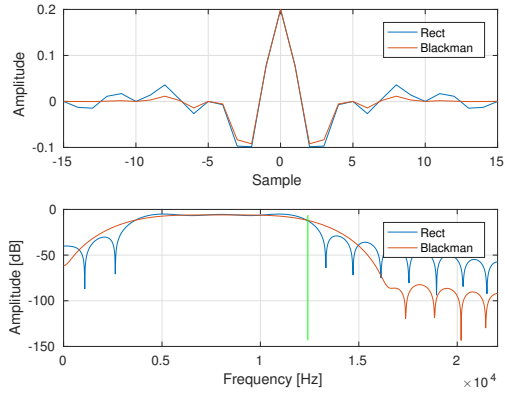


Figure 11: Low-pass shifted filter 31 coefficients $f_c = 0.2$, with and without Blackman windows, green line at $f_c + 8000 * 2 / f_s$.

applied on the low pass will give a high-pass with a new $f_{c_{mirror}} = 1 - f_c$ as shown on Figure 13.

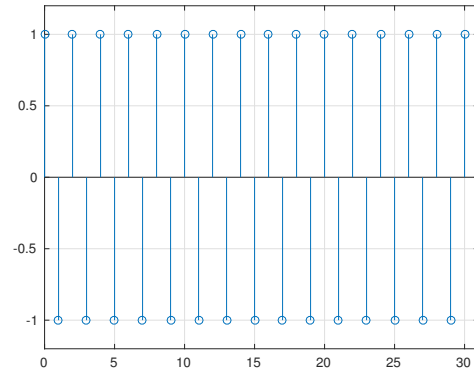


Figure 12: Mirror filter used for 31 coefficients

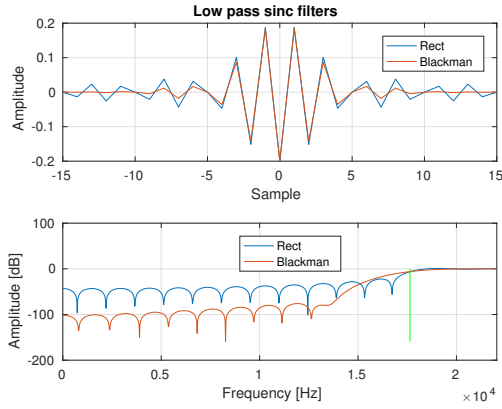


Figure 13: Low-pass 31 coeffs, mirrored with and without Blackman windows. Green line at Nyquist frequency minus $f_c = 0.2$

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

i. Influence of points numbers

In time domain it's obvious that the filter will just be longer if there are more points, the cardinal sinus being truncated with a bigger window...

The influence on the frequency is illustrated by the Figure 14, 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 long and slow filter implementation as numerous operation will be needed to apply this filter to the signal, this is the so called duality time/frequency in signal processing.

ii. 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.

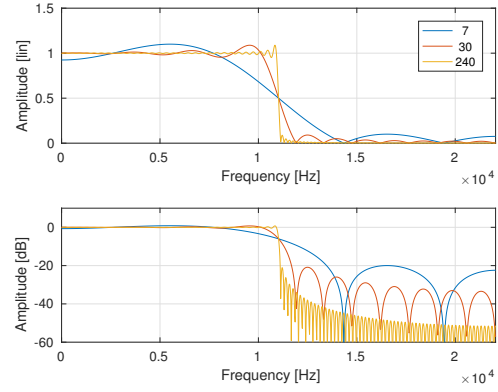


Figure 14: Linear and dB frequency response of low-pass sinc filters $f_c = 0.5$ with 7, 30 and 240 points

For odd number it's quite simple, the center is defined they will be even number on each side of the center.

For even number, I choose by analogy to the fft to 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 are based on a ratio of n over N , the trick is just that n at the center of the filter is equal to $\frac{N}{2}$ so that the ratio at the center is 0.5:

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

VI. CONCLUSION

This article shows a recap to realise different FIR filter based on cardinal sinus filter, due to the limitation induced by the truncation a set of windows are proposed to reduce the ripple. It been seen that it needed an important number of points in order to achieve the desired step frequency however this imply a slow fil-

ter at the implementation. If speed is an issue
IIR filter should be considered.

REFERENCES

[<http://www.labbookpages.co.uk/audio/firWindowing.html>]

[<http://www.phon.ucl.ac.uk/courses/spsci/dsp/filter.html>]

[http://www.eetimes.com/document.asp?doc_id=1275863]
