Linear Filtering in Frequency-Domain

1 Introduction

Linear filtering can be easily implemented in time-domain resorting to the use of finite impulse response (FIR) digital filters and convolution property as,

$$y(n) = \sum_{k=0}^{N-1} x(n-k)h\left(k\right),\tag{1}$$

where $x(n)$ is the input signal, $h(k)$ is a length-N sequence filter coefficients, $y(n)$ represents the filtered output signal and N is the number of filter coefficients. Analysing this equation we can note that, for a block signal of length N, the required number of operations for the direct calculation of equation (1) evolves with N^2, $\mathcal{O}(N^2)$. This limitation imposed the emergence of algorithm, where the linear convolution is calculated faster than directly implementing (1). In this sense, fast convolution can be implemented by sectioning or block the signal and use the fast Fourier transform (FFT)/ inverse fast Fourier transform (IFFT) on these blocks to take advantage of the efficiency of the FFT. However, the implementation of non-cyclic convolution with the finite-length of cyclic convolution that the FFT provides needs to be fixed. A solution to this problem was achieved through the application of overlap-save and overlap-add method, where the complexity evolves in log scale $\mathcal{O}(N \log_2 N)$. The general method is to split the input signal into manageable blocks, then apply the FFT to to perform the linear convolution and at the end recombine the output blocks such that it is avoided the wrap-around errors.

2 Overlap-Save Method

In this method we start by adding R null samples at the beginning of the sequence, which is commonly equal to the length of the impulse response of the filter. Then, the data is divided into blocks of length N samples, but with successive blocks overlapping by R samples. Then, the an N-point FFT is computed for each data block. Then, the multiplication of N-point FFT and the transfer function is performed, followed by IFFT operations to obtain the time-domain block signal. These initial samples are then discarded, and only the error-free $N-R$ samples are saved in the output record. In the figure below is illustrated an example of overlap-add method with $R = \frac{N}{2}$ and $L = N - R = R$.

2.1 Frequency Response of Filter

An example of FIR filter (*raised-cosine filter*) is presented in Fig. 1, including its impulse response, $h(k)$, and frequency response, $H(f)$. The frequency response of filter can be obtained from the FFT of impulse response of the filter or it can be directly defined using the frequency-domain formula. When it is used the FFT we must zero-pad the impulse response to the length N before performing the FFT operation or computing the N-point FFT of impulse response. When directly defined in frequency-domain, we directly define $H(f)$ over N frequency bins. In all the cases, the frequency response of the filter will be limited to the frequency interval $fwindowTF$, $[-\frac{fwindowTF}{2}, \frac{fwindowTF}{2}]$, and this range show us the N frequency components obtained from FFT. The minimum frequency bin is $-\frac{fwindowTF}{2}$ and the maximum bin is $\frac{fwindowTF}{2}$, in which F_s corresponds to the sampling frequency. We can note that the spectral width of $H(f)$ is $fwindowTF$, which is the inverse of sampling period, dt. It is also important to note that, for a
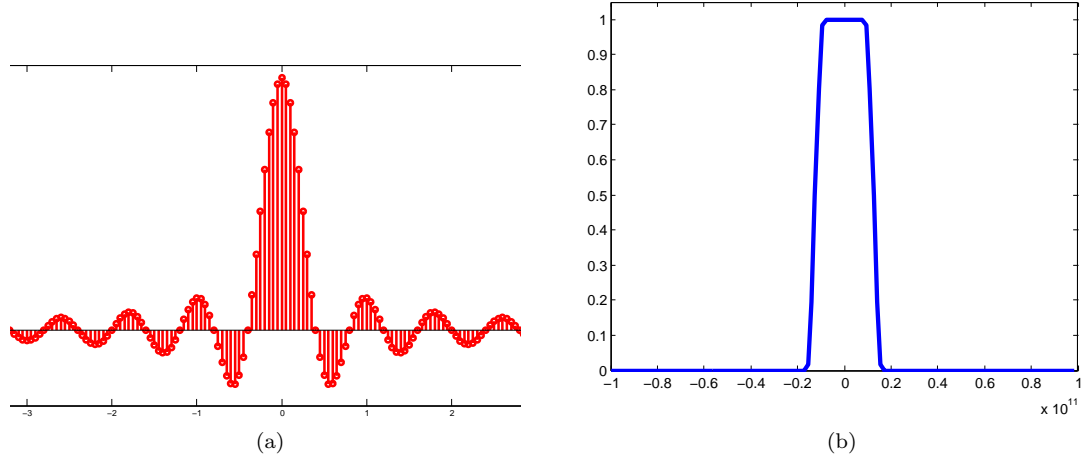
Figure 1: a) Impulse response of raised cosine filter; b) Frequency response of raised cosine filter.

given sampling frequency, the frequency resolution, ΔR, of $H(f)$ depends on the parameter N and it increases with N, $\Delta f = \frac{fwindowTF}{N}$.

t

L

| x1 | x2 | x3 | x4 | x5 | x6 |

N

| Added zeros | x1 |

R

| x1 | x2 |

| x2 | x3 |

| x3 | x4 |

| x4 | x5 |

| x5 | x6 |

FFT

*

TF

IFFT

y1

discard

y2

discard

y3

discard

y4

discard

y5

discard

y6

discard

| y1 | y2 | y3 | y4 | y5 | y6 |

- - -