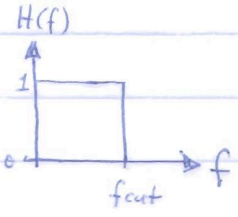
(Tapers / Windows) closely related to filters

multiplication of data and "weights" → convolution in frequency

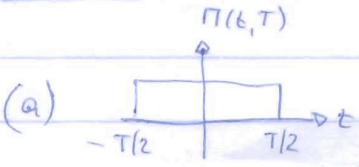
convolution of data and "weights" → multiplication in frequency

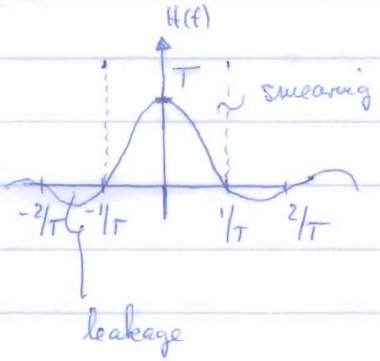
ideal ~~taper~~ window: $\lim_{T \to \infty} \Pi(t,T)$

ideal filter



Windows





$\mathcal{F}(t,T) = T \operatorname{sinc} fT$

leakage: part of the signal at one frequency is leaked to other frequencies

one measure is to take the highest side lobe level

$10 \cdot \log\left(\frac{H^{lobe}}{H_0}\right) = dB$

-13 dB

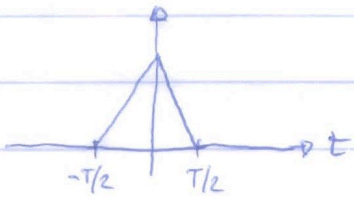
smearing: the signal at a given frequency is smeared over a frequency interval

location of the first zero crossing

$1/T$

GOAL of windowing / tapering : reduce the amplitude of the sidelobes by tapering off the sharp edges of the gate fctn.

(5)



$$h(t,T) = \begin{cases} 1-2|t|/T & |t| \le T/2 \\ 0 & |t| > T/2 \end{cases}$$

Find Fourier transform, noting that

$$h(t,T) \cdot T/2 = \Pi(t,T/2) * \Pi(t,T/2)$$

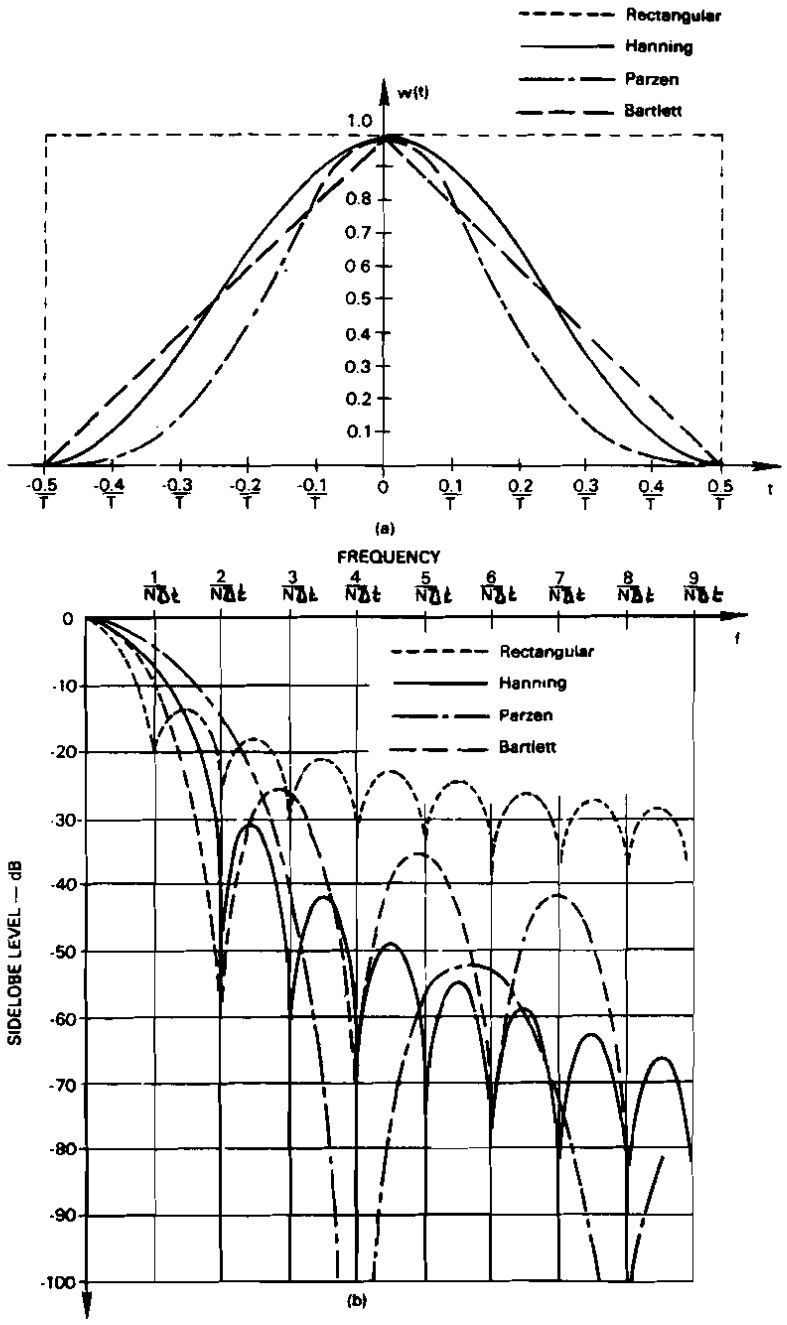
$$\mathcal{F}(h(t,T)) = \frac{2}{T} \mathcal{F}(\Pi(t,T/2)) \cdot \mathcal{F}(\Pi(t,T/2))$$

$$= \frac{2}{T} \frac{T}{2} \operatorname{sinc} \frac{fT}{2} \cdot \frac{T}{2} \operatorname{sinc} \frac{fT}{2}$$

$$= \frac{T}{2} \operatorname{sinc}^2(fT/2)$$

highest sidelobe level -26 dB
first zero crossing $2/T$

→ reduce leakage by a factor of 100
increase smearing (frequency resolution) by factor of 2



$T = N \cdot \Delta t$

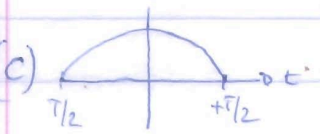
Figure 9.8 FFT weighting or window functions.

Modified from Brigham, O.E., 1988: The fast Fourier transform and its application. Englewood Cliffs, NJ, Prentice Hall, 448pp.

other windows

Hanning: $h(t,T) = \begin{cases} \frac{1}{2}\,1+\cos(2\pi t/T) & |t| \le T/2 \\ 0 & |t| > T/2 \end{cases}$

(c)



highest sidelobe level -32 dB

first zero crossing $2/T$

leakage reduced by a factor of 1000 as compared to $\Pi(t,T)$

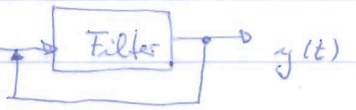
(scale factor $\sqrt{8/3}$)

FILTERS

1. sharp cut-off
2. good transient (oscillation in data)
3. minimal phase shift
4. leave low frequency undisturbed
5. efficient, i.e., little CPU time little filter loss

non-recursive

recursive



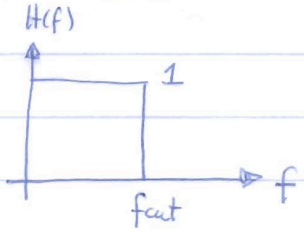
general form

$$y_n = \sum_{k=-N}^{N} c_k\, x_{n-k}$$ non-recursive part

c_k filter weights

$$+ \sum_{k=1}^{N} d_k\, y_{n-k}$$ recursive part

Ideal filter



(1) compute FFT
(2) set coefficients above fcut to 0
(3) compute inverse FFT
(4) presto !