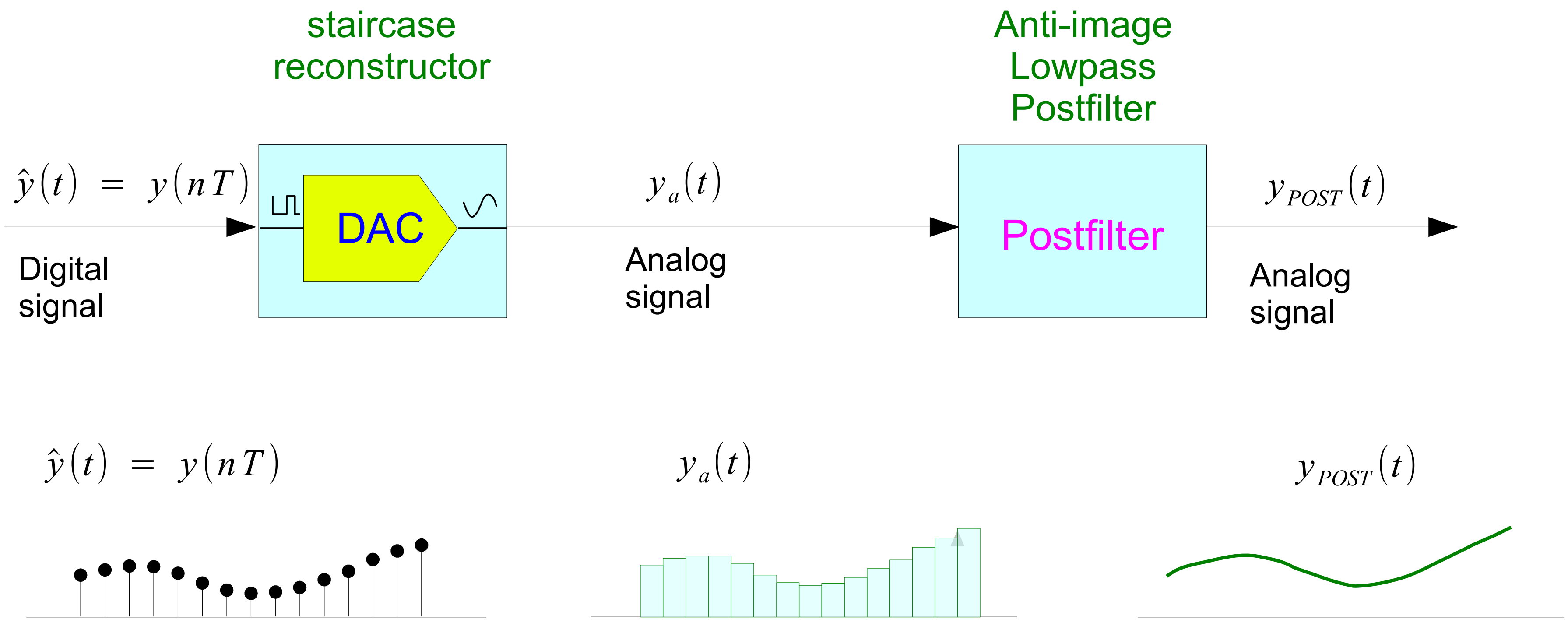
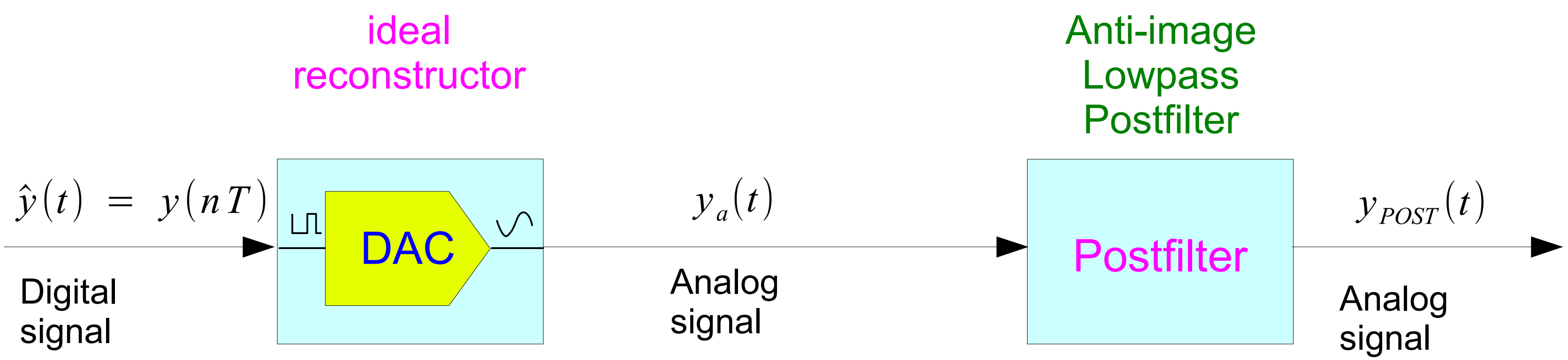
Anti-Image Postfilter (7B)

-
-

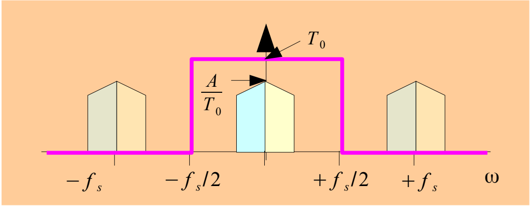
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

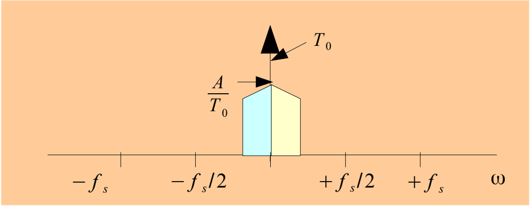

staircase
reconstructor
Anti-image
Lowpass
Postfilter
$\hat{y}(t) = y(nT)$
DAC
$y_a(t)$
Postfilter
$y_{POST}(t)$
Digital
signal
Analog
signal
Analog
signal
$\hat{y}(t) = y(nT)$
$y_a(t)$
$y_{POST}(t)$



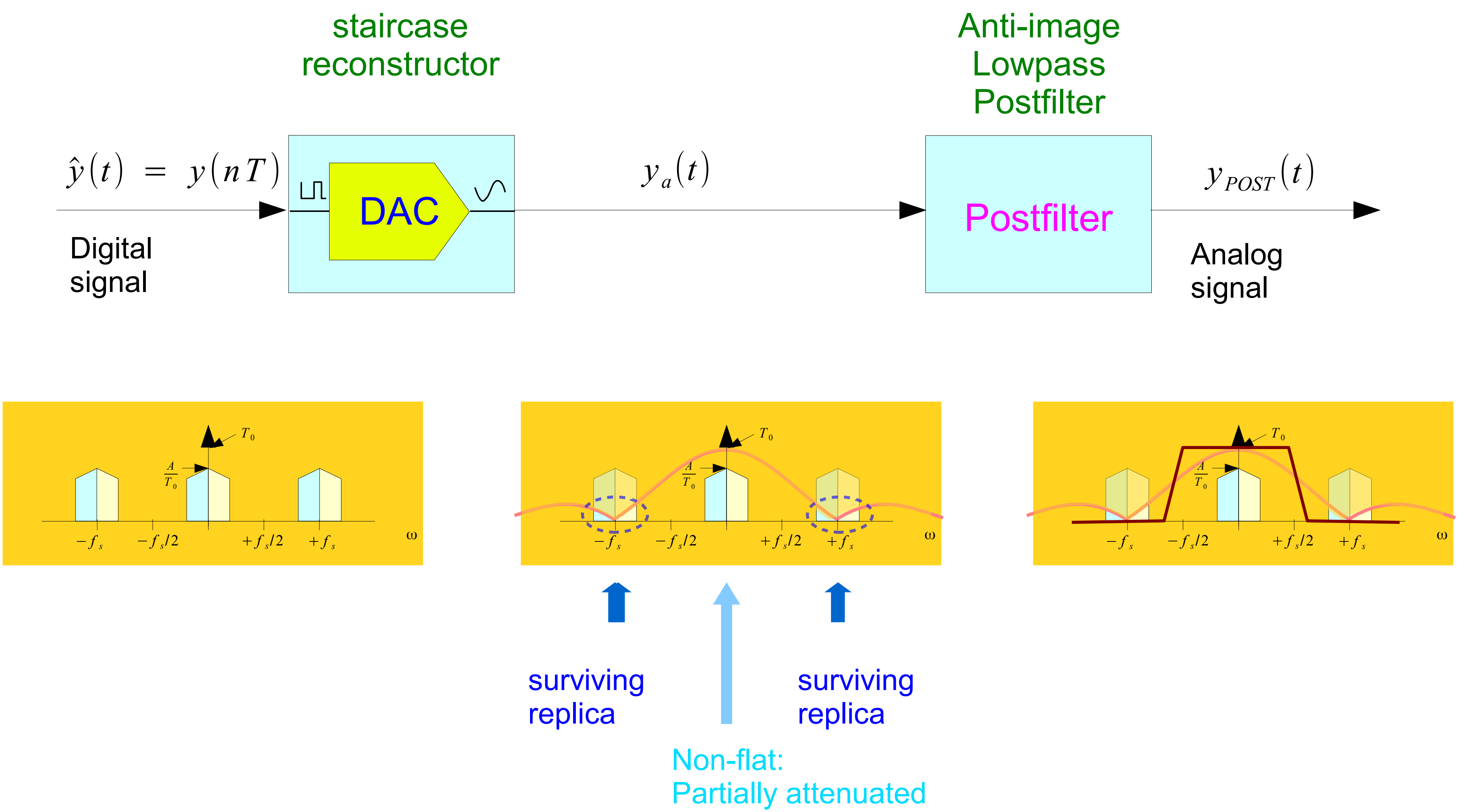
ideal
reconstructor
Anti-image
Lowpass
Postfilter
$\hat{y}(t) = y(nT)$
DAC
$y_a(t)$
Postfilter
$y_{POST}(t)$
Digital
signal
Analog
signal
Analog
signal



T_0
$\frac{A}{T_0}$
$-f_s$
$-f_s/2$
$+f_s/2$
$+f_s$
ω

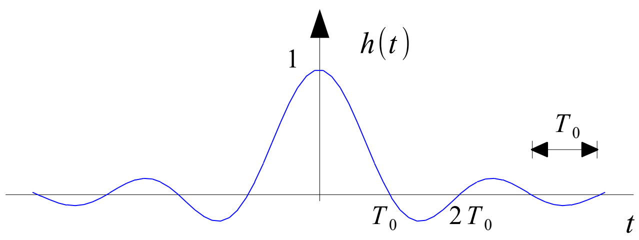


T_0
$\frac{A}{T_0}$
$-f_s$
$-f_s/2$
$+f_s/2$
$+f_s$
ω

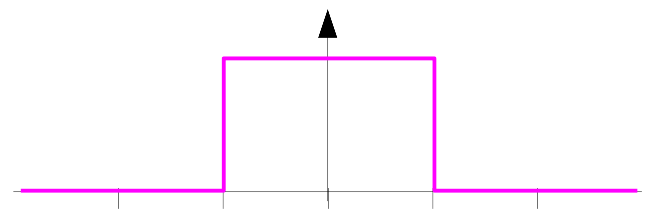

staircase reconstructor
Anti-image Lowpass Postfilter
$\hat{y}(t) = y(nT)$
DAC
$y_a(t)$
Postfilter
$y_{POST}(t)$
Digital signal
Analog signal
surviving replica
surviving replica
Non-flat:
Partially attenuated

CTFT of Reconstructors (1)

$t = \pm T_0,\ \pm 2T_0,\ \pm 3T_0,\ \cdots \Rightarrow h(t) = 0$



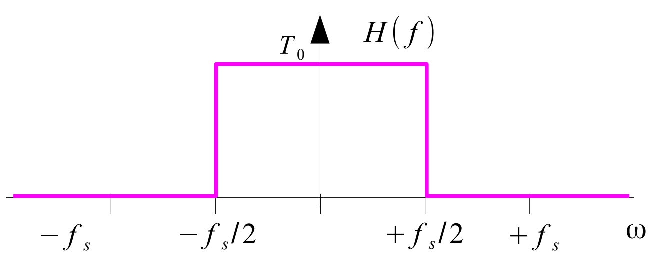
$$\frac{1}{T_0} \equiv f_s$$

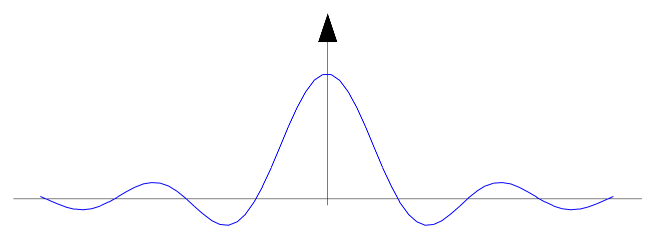


$$h(t) = \frac{\sin(\pi t/T_0)}{\pi t/T_0} = \frac{\sin(\pi f_s t)}{\pi f_s t}$$

CTFT

CTFT



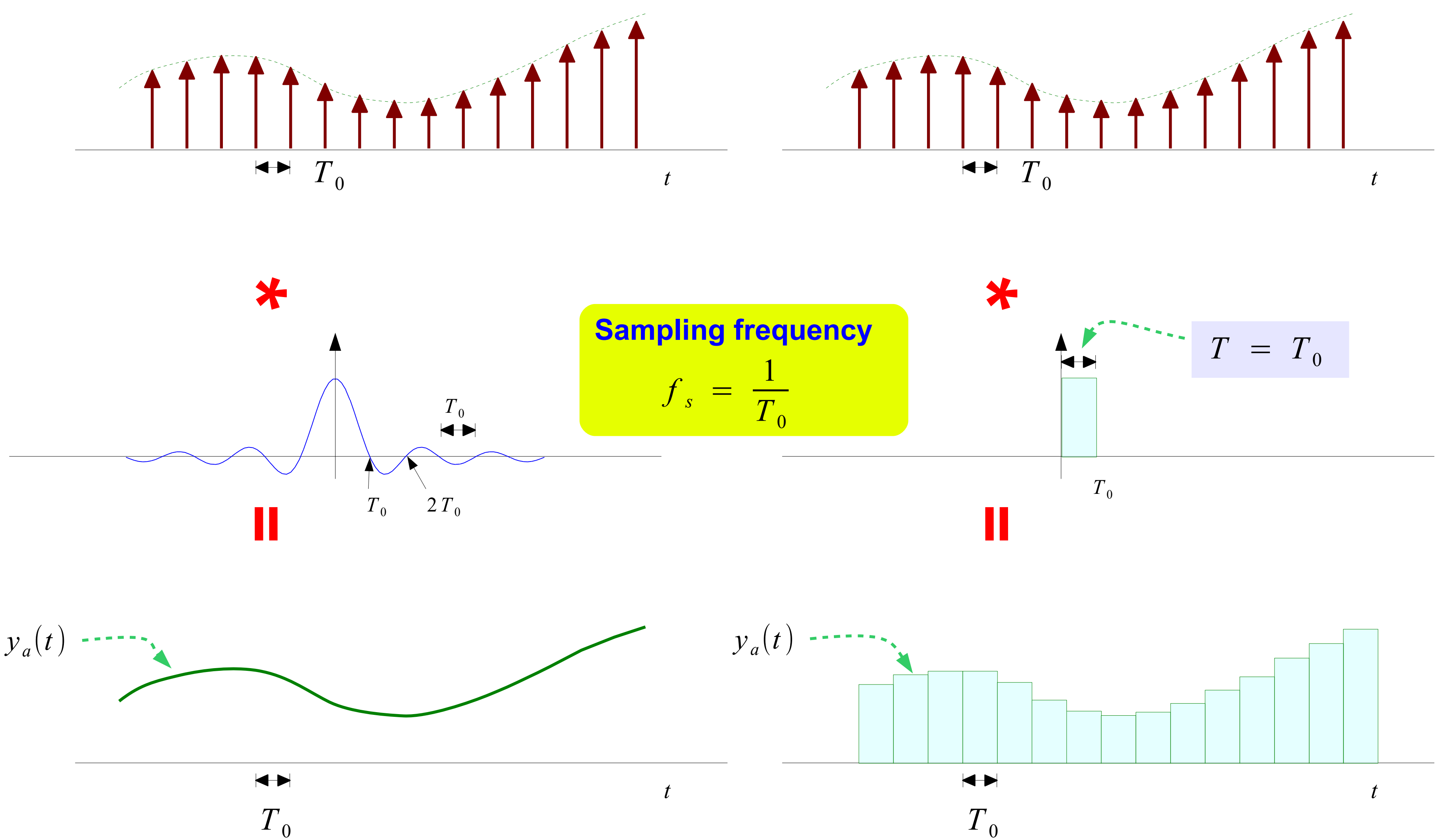


$$H(f) = \begin{cases} T_0, & |f| \le f_s/2 \\ 0, & otherwise \end{cases}$$

Reconstruct via Convolution

Ideal Reconstructor

Practical Reconstructor

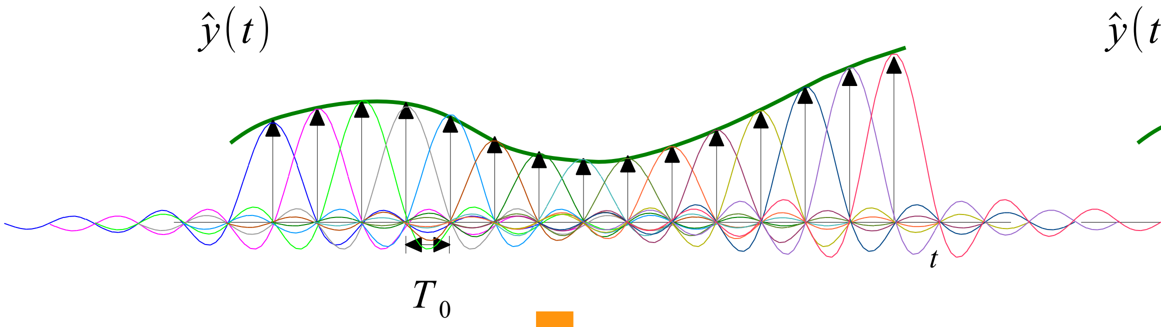


Sampling frequency

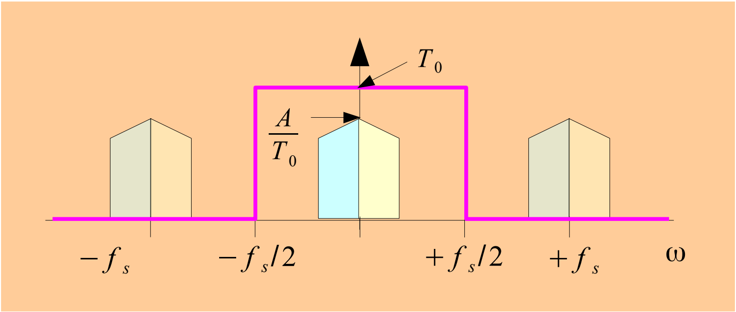
$$f_s = \frac{1}{T_0}$$

Reconstructors in Frequency Domain

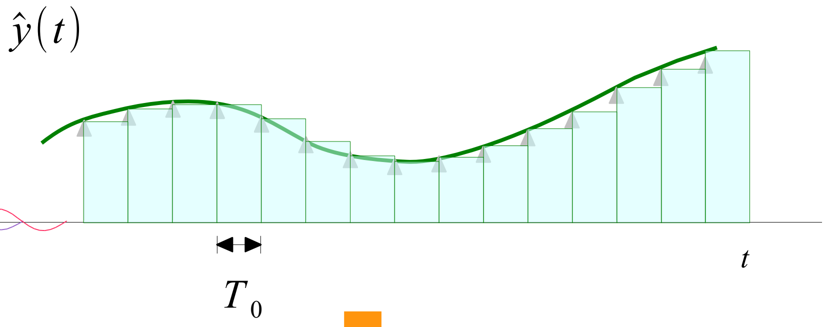
Ideal Reconstructor



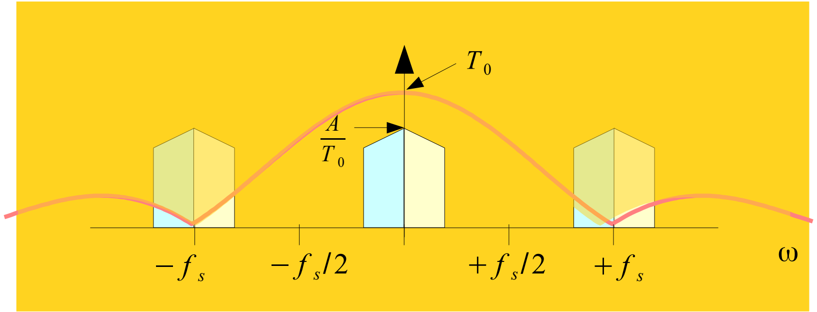
CTFT

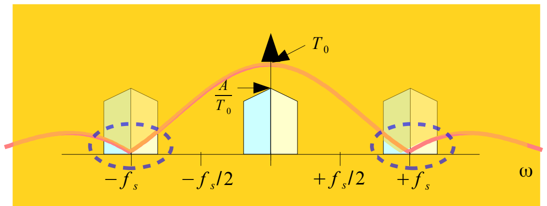


Practical Reconstructor



CTFT





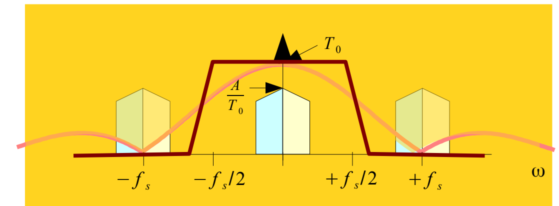
surviving
replica

Non-flat:
Partially attenuated

surviving
replica

Surviving spectral replicas
can be removed by
an additional lowpass filter

Anit-image Postfilter



Cutoff Frequency $f_{max} \le \dfrac{f_s}{2}$

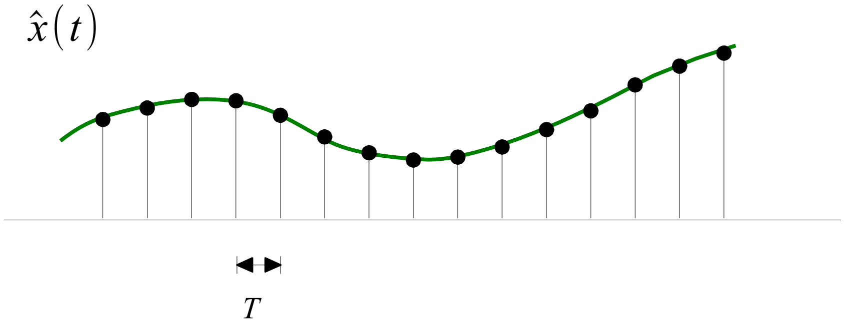
Freq domain (reconstructor + postfilter)
to remove the spectral replicas
as much as possible

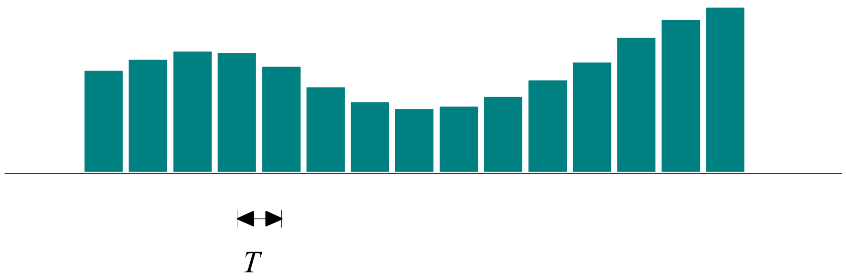
Time domain Effect of rounding off the corners
of staircase output making smoother

Two stage (Staircase Reconstructor + Postfilter) →
simplicity of implementation of reconstructor
: DAC – generating an analog output that remains constant during T

Emulate the ideal reconstructor

Analog Reconstructor





$$\hat{y}(t) = \sum_{n=-\infty}^{+\infty} y(nT)\,\delta(t-nT)$$

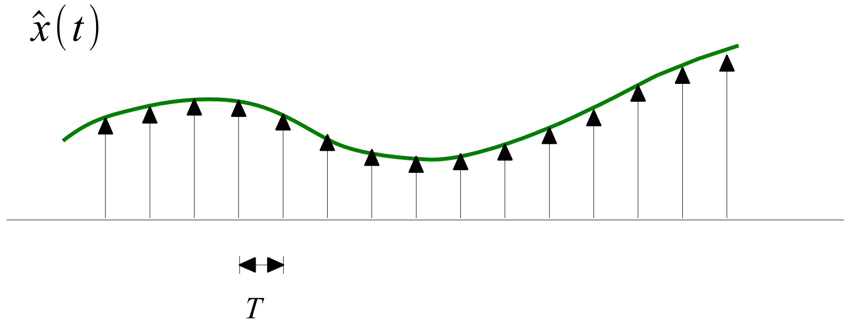
$$Y_a(f) = H(f)\,\hat{Y}(f)$$

$$y_a(t) = \int_{-\infty}^{+\infty} h(t-t')\,\hat{y}(t')\,dt'$$

$$\hat{Y}_a(f) = \frac{1}{T}\sum_{m=-\infty}^{+\infty} Y(f-mf_s)$$

$$y_a(t) = \sum_{n=-\infty}^{+\infty} y(nT)\,h(t-nT)$$

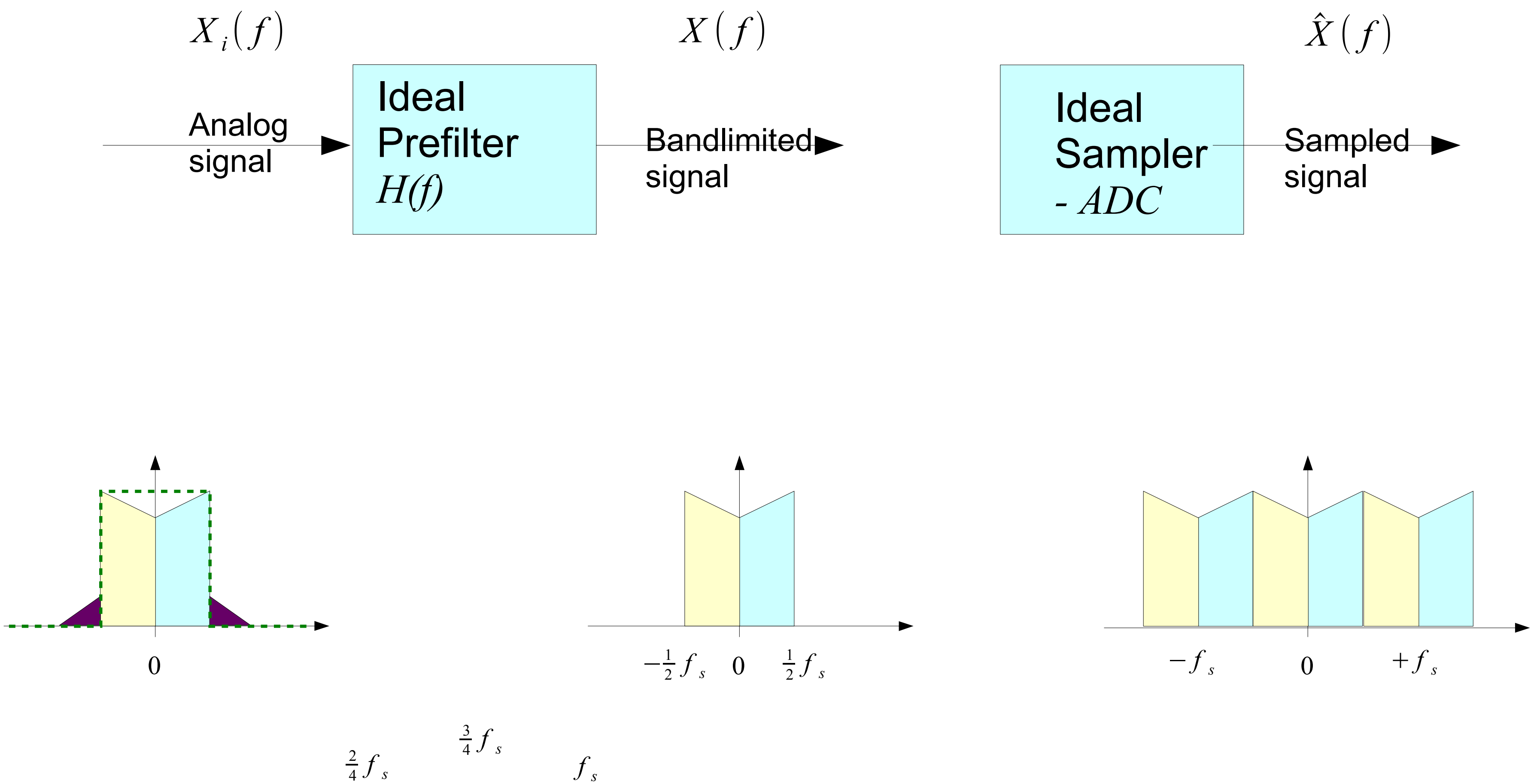
Impulse Response of Ideal Reconstructor



$$\hat{Y}(f) = \frac{1}{T} Y(f) \qquad -\frac{f_s}{2} \le f \le +\frac{f_s}{2}$$

$$y(t) = \sum_{n=-\infty}^{+\infty} y(nT) h(t-nT)$$

$$h(t) = \frac{\sin(\pi t/T)}{\pi t/T} = \frac{\sin(\pi f_s t)}{\pi f_s t}$$



$X_i(f)$
Analog signal
Ideal Prefilter $H(f)$
$X(f)$
Bandlimited signal
Ideal Sampler - ADC
$\hat{X}(f)$
Sampled signal
0
$-\frac{1}{2}f_s$
0
$\frac{1}{2}f_s$
$-f_s$
0
$+f_s$
$\frac{2}{4}f_s$
$\frac{3}{4}f_s$
f_s

References

[1] http://en.wikipedia.org/
[2] J.H. McClellan, et al., Signal Processing First, Pearson Prentice Hall, 2003
[3] A "graphical interpretation" of the DFT and FFT, by Steve Mann
[4] R. G. Lyons, Understanding Digital Signal Processing, 1997
[5] AVR121: Enhancing ADC resolution by oversampling
[6] S.J. Orfanidis, Introduction to Signal Processing
www.ece.rutgers.edu/~orfanidi/intro2sp