Signal acquisition and processing

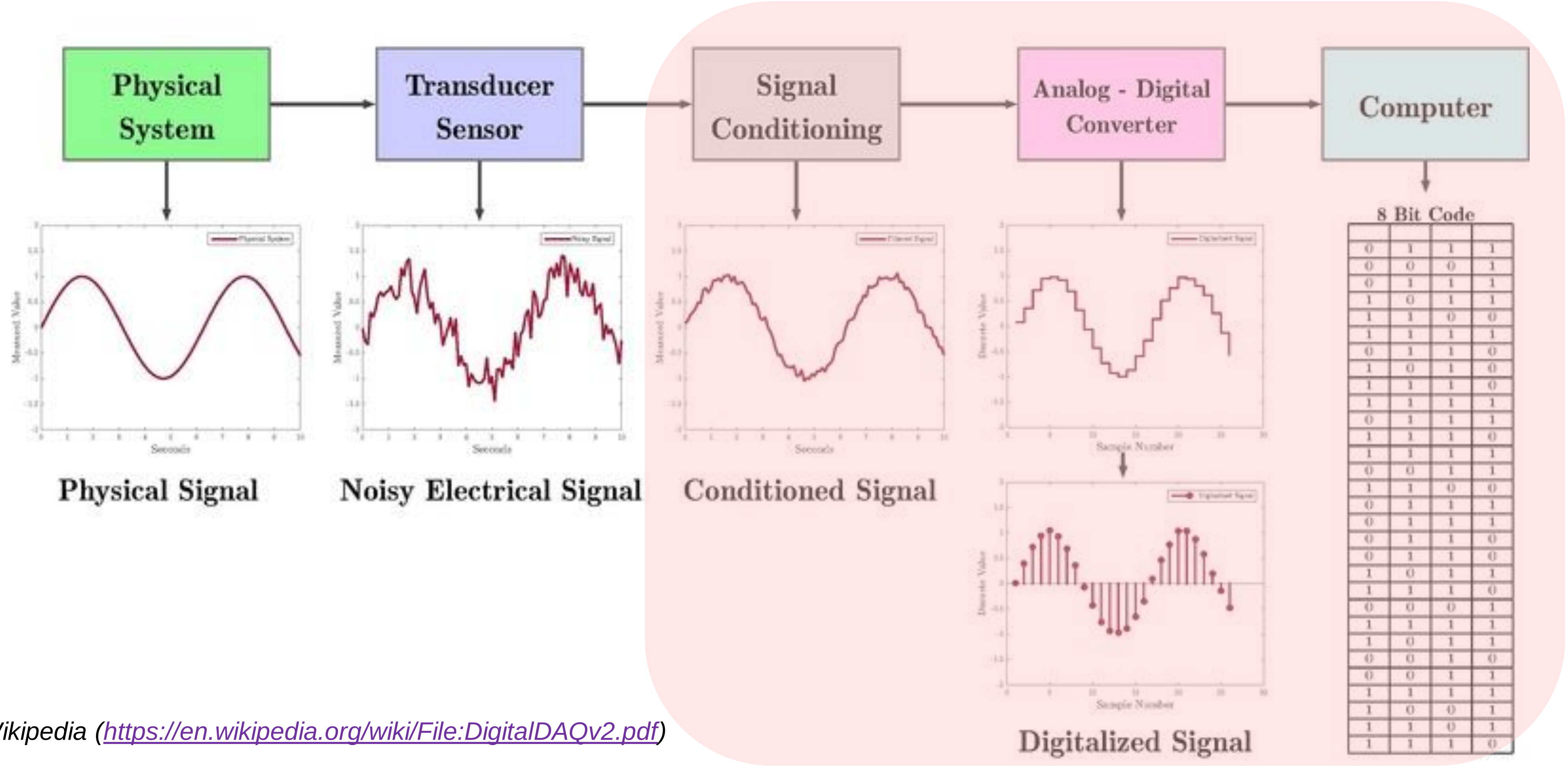
— ISEP, IG 2407, Acquisition et traitement du signal

Hmaied SHAIEK

E-mail : hmaied.shaiek@cnam.fr

Tel : 01 40 27 24 99

Digital Data Acquisition System



Wikipedia (https://en.wikipedia.org/wiki/File:DigitalDAQv2.pdf)

Outline

1. **Data acquisition and analysis (2 lectures)**

2. **Digital data filtering (2 lectures)**

3. **Random signal processing (1 lecture)**

Outline

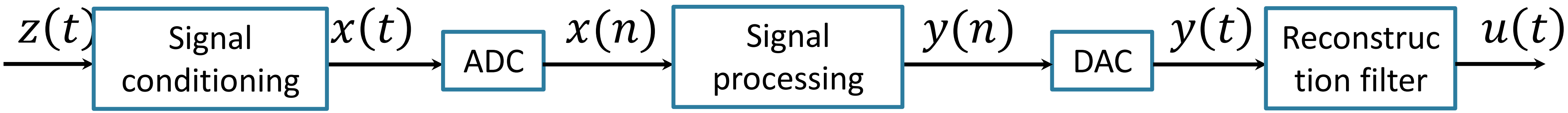
1. **Data acquisition and analysis**
   - **Digital data acquisition system**
   - **Discrete Fourier Transform**
   - **Fast Fourier transform**
   - **Z Transform and transfer function**

Outline

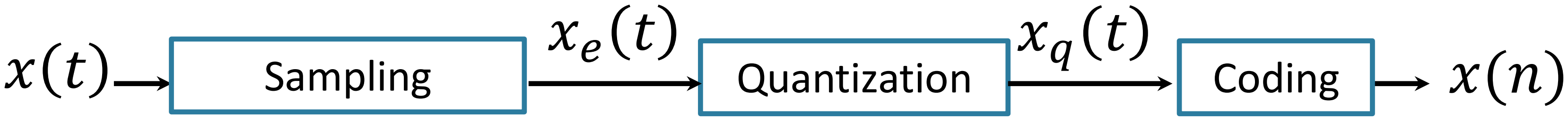
1. **Data acquisition and analysis**
   - **Digital data acquisition system**
   - Discrete Fourier Transform
   - Fast Fourier transform
   - Z Transform and transfer function

Digital data processing scheme

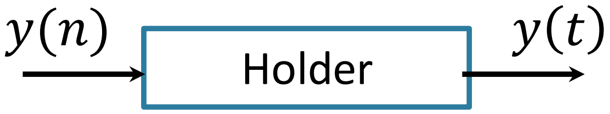
Block diagram



ADC: Analog to Digital Converter



DAC : Digital to Analog Converter

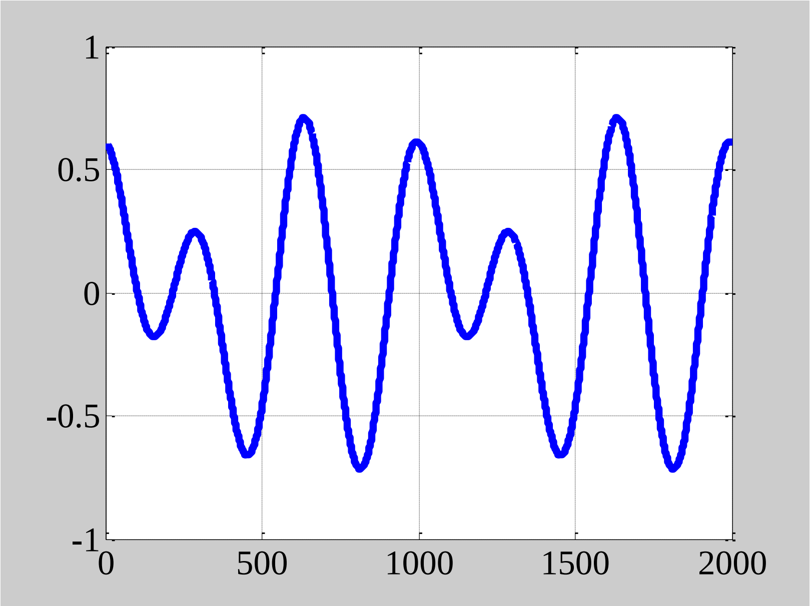


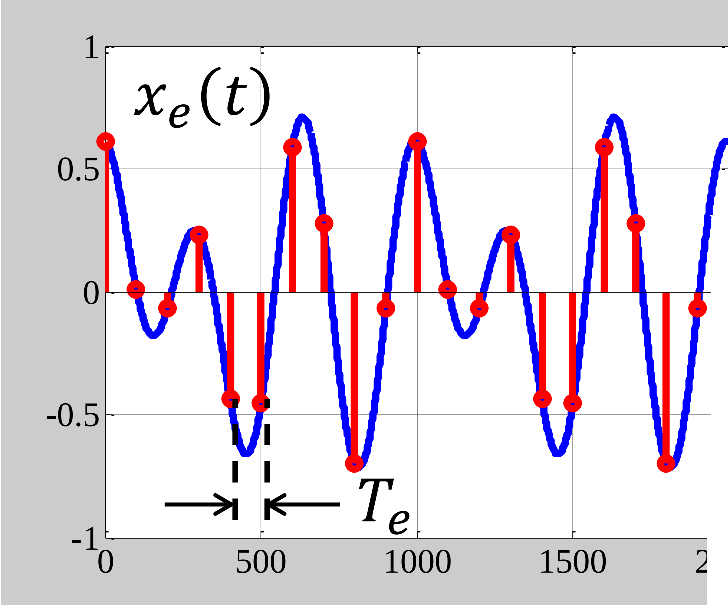
ADC/Sampling

Principle

$x(t) \rightarrow$ Sampling $\xrightarrow{x_e(t)}$ Quantization $\xrightarrow{x_q(t)}$ Coding $\rightarrow x(n)$

An analog signal $x(t)$:





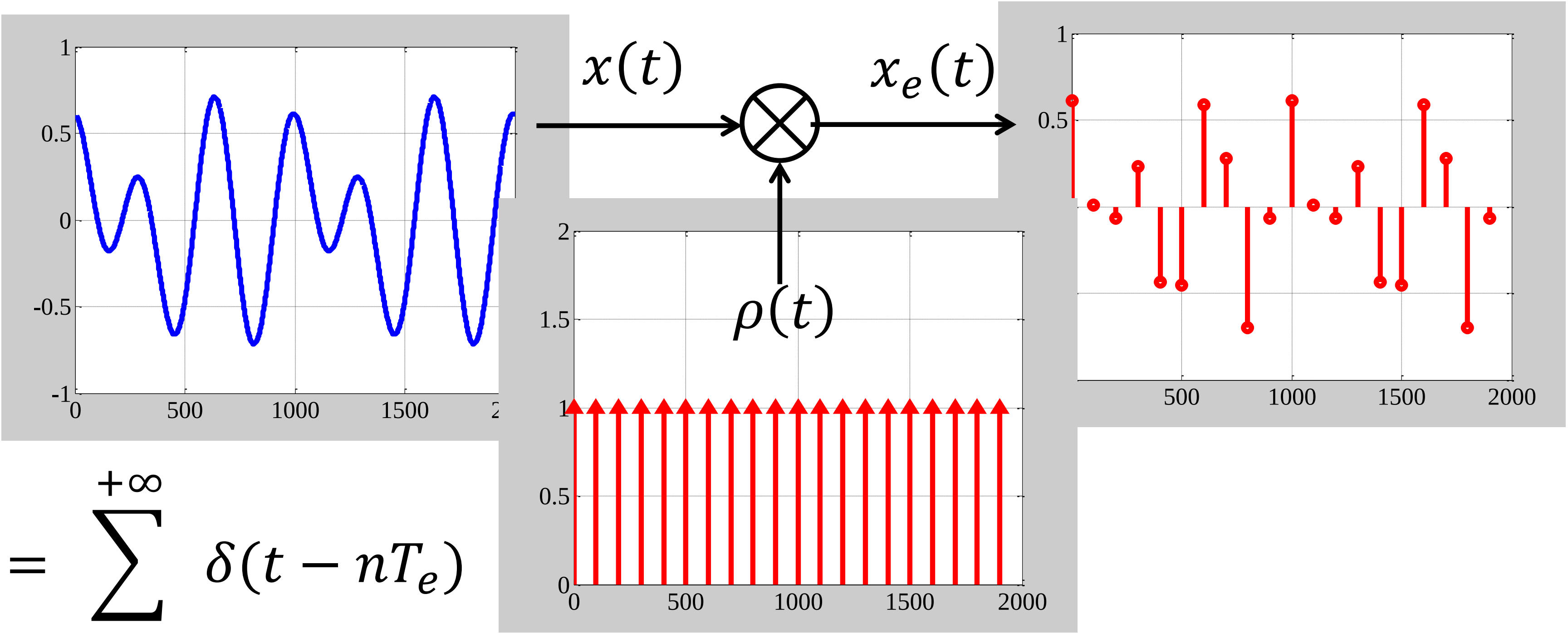
T_e : Sampling time

$F_e = 1/T_e$: sampling frequency

Sampled signal $x_e(t) = \{x(nT_e)\}_{n \in Z}$ $x_e(t) = \{x_n\}_{n \in Z} = x(n)$

ADC/Sampling

Principle



$$\rho(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT_e)$$

Sampled signal : $x_e(t) = x(t)\rho(t) = x(t)\sum_{n=-\infty}^{+\infty} \delta(t - nT_e)$

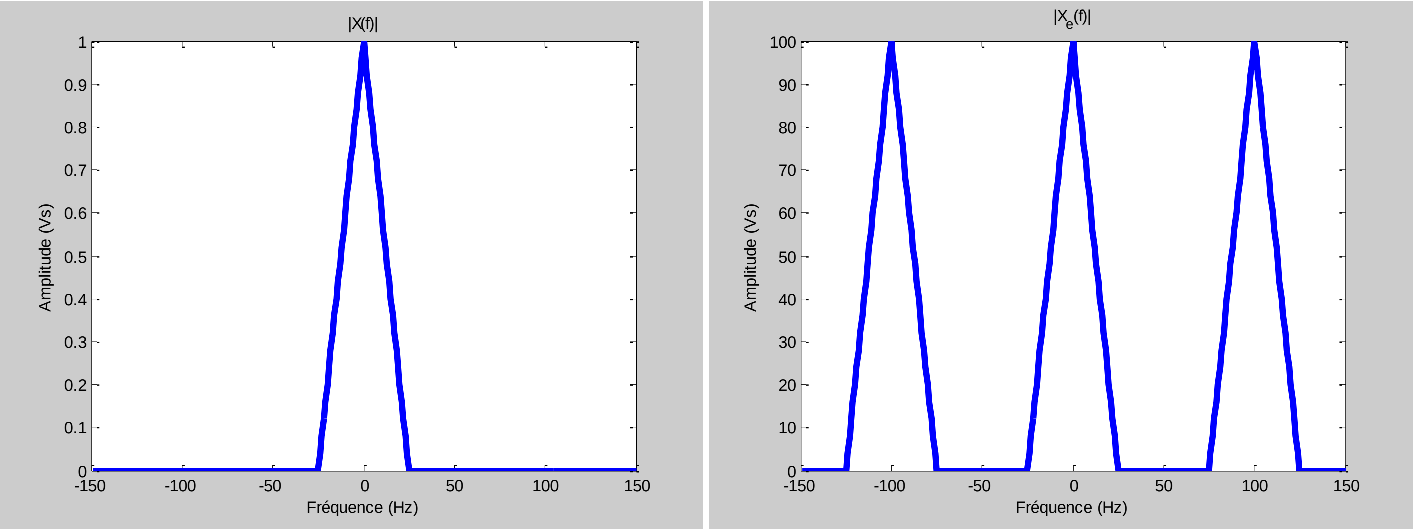
ADC/Sampling

Spectrum periodization

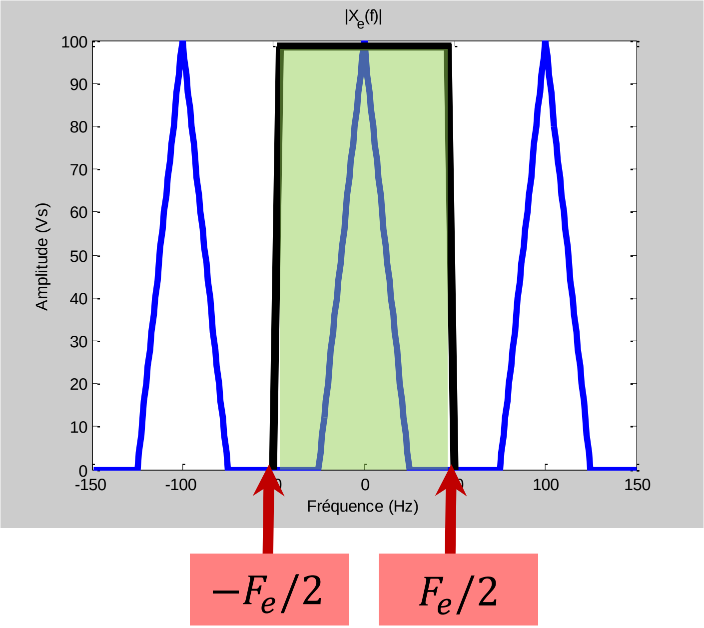
$$X_e(f) = \mathrm{FT}[x_e(t)] = \mathrm{FT}[x(t)\rho(t)] = \mathrm{FT}[x(t)] * \mathrm{FT}[\rho(t)]$$

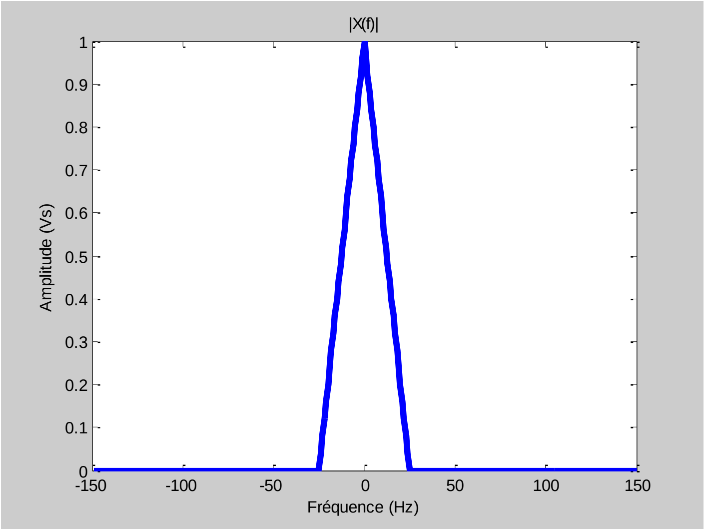
$$= X(f) \; * \left[\frac{1}{T_e}\sum_{k=-\infty}^{+\infty} \delta\left(f - k\frac{1}{T_e}\right)\right] = \frac{1}{T_e}\sum_{k=-\infty}^{+\infty} X\left(f - k\frac{1}{T_e}\right)$$

$X_e(f)$: repetition of $\mathrm{X}(f)$ around des frequencies kF_e, with $k \in Z$



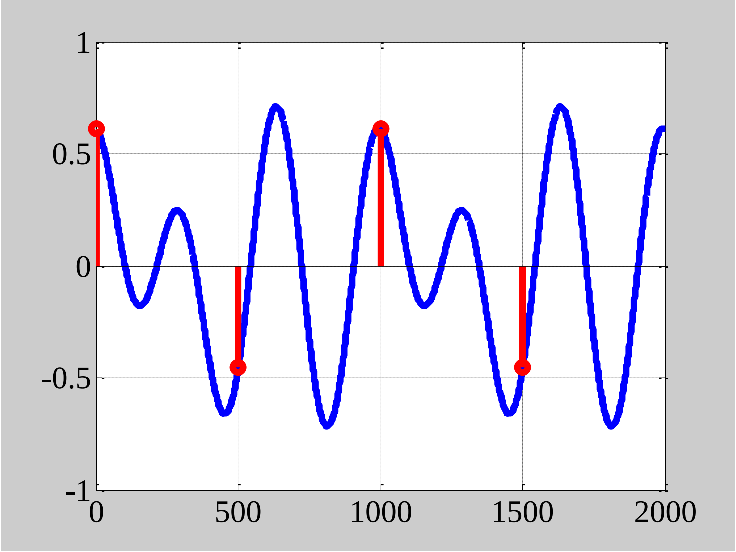
ADC/Sampling
Perfect reconstruction condition





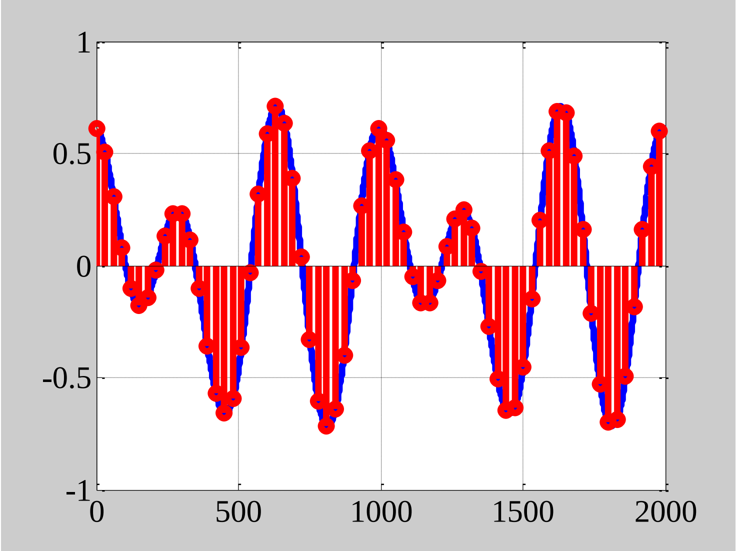
ADC/Sampling

Choice of F_e



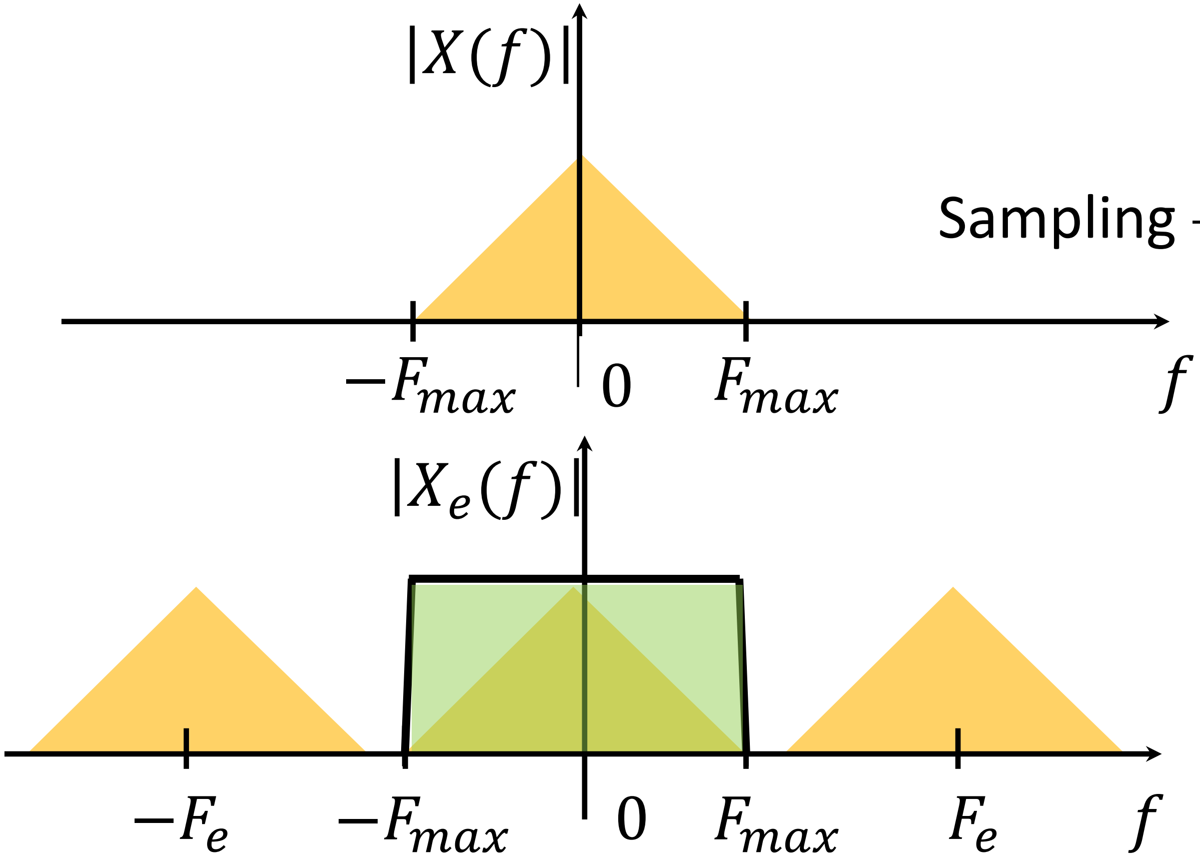
It's difficult to reconstruct the signal

Solution : increase the number of samples $\rightarrow$ increase F_e.



ADC/Sampling

Shannon theorem



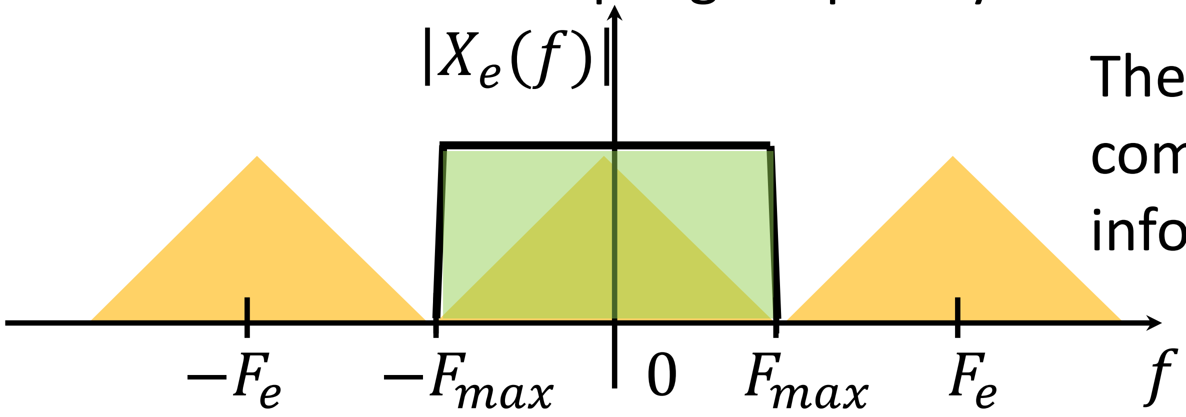
Sampling $\rightarrow$ spectrum periodisation

Spectra are disjoint

ADC/Sampling

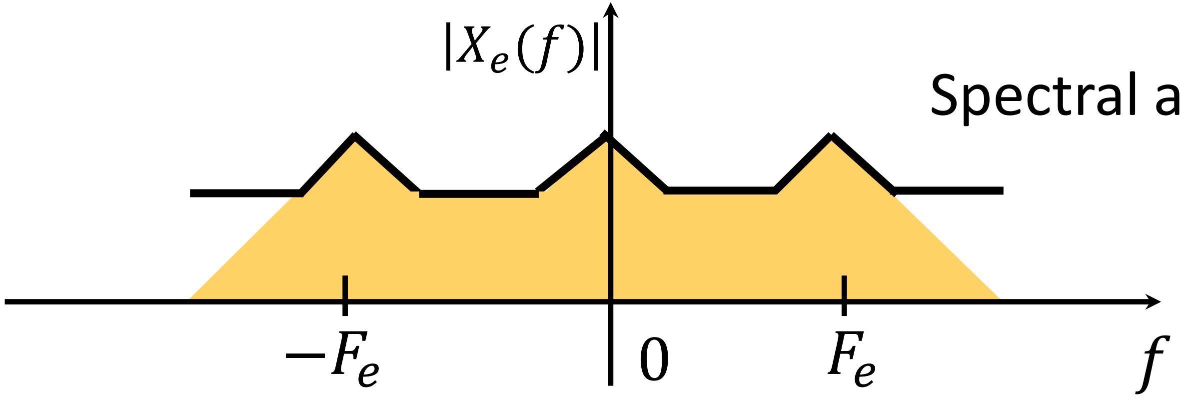
Shannon theorem

If we decrease the sampling frequency



The spectrum components come together but the information is "reconstructible"

If we further decrease the sampling frequency

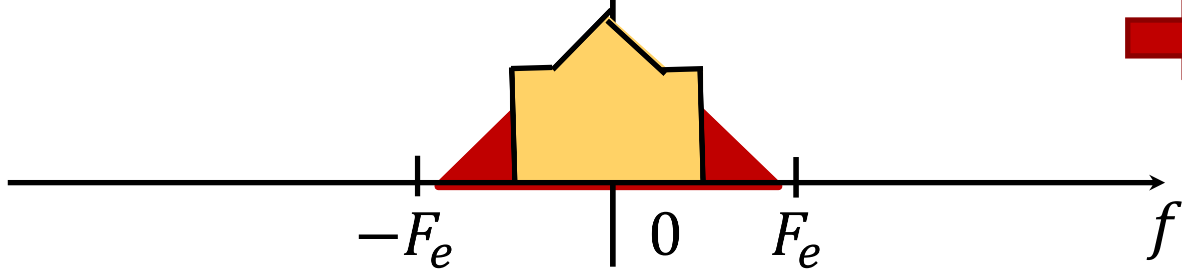


Spectral aliasing

ADC/Sampling

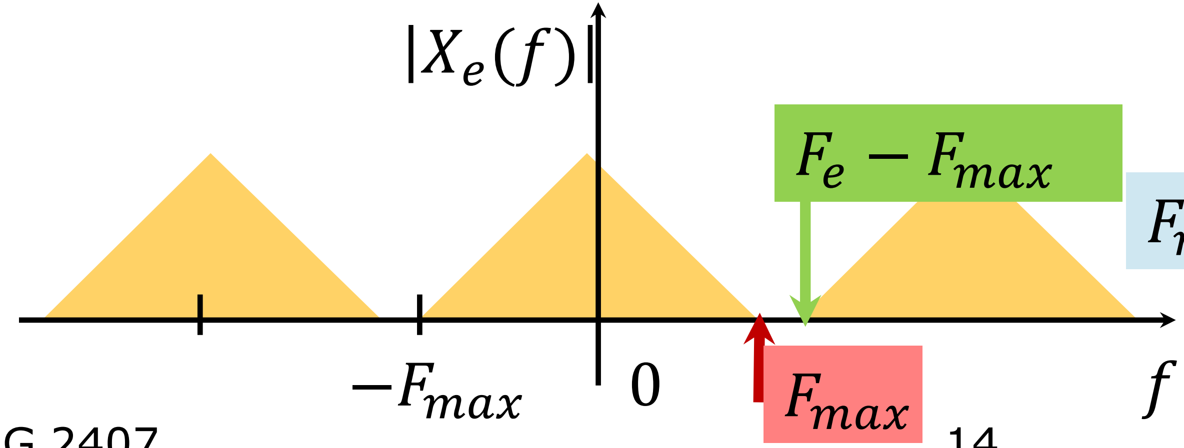
Shannon theorem

Signal spectrum after reconstruction (Low-pass filtering)



Loss of information

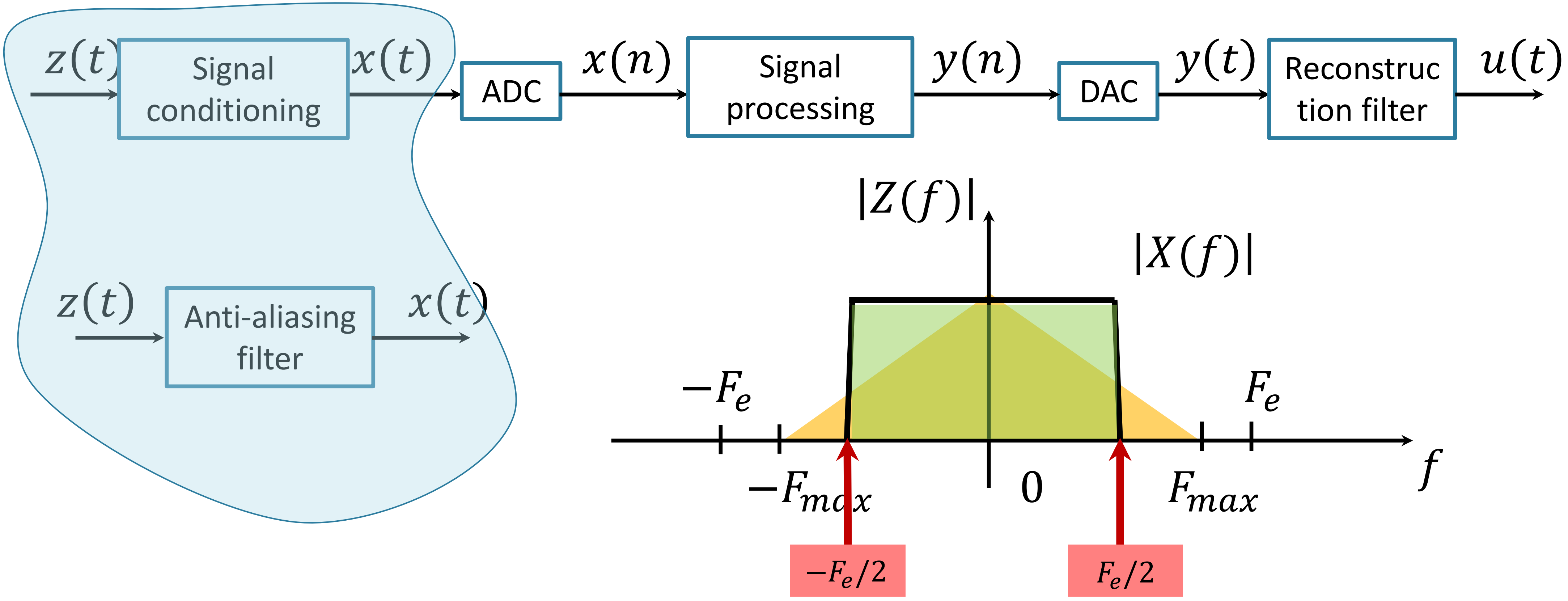
Condition to avoid spectral aliasing : Shannon theorem



$$F_{max} < F_e - F_{max} \rightarrow F_e > 2F_{max}$$

Signal conditioning

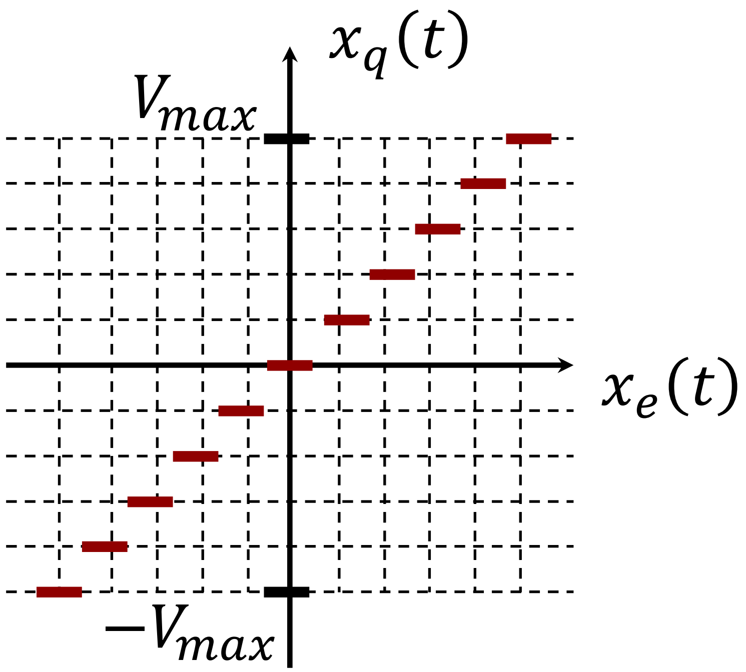
Anti-aliasing filter



ADC/Quantization

Quantization step size

$$x(t) \rightarrow \boxed{\text{Sampling}} \xrightarrow{x_e(t)} \boxed{\text{Quantization}} \xrightarrow{x_q(t)} \boxed{\text{Coding}} \rightarrow x(n)$$



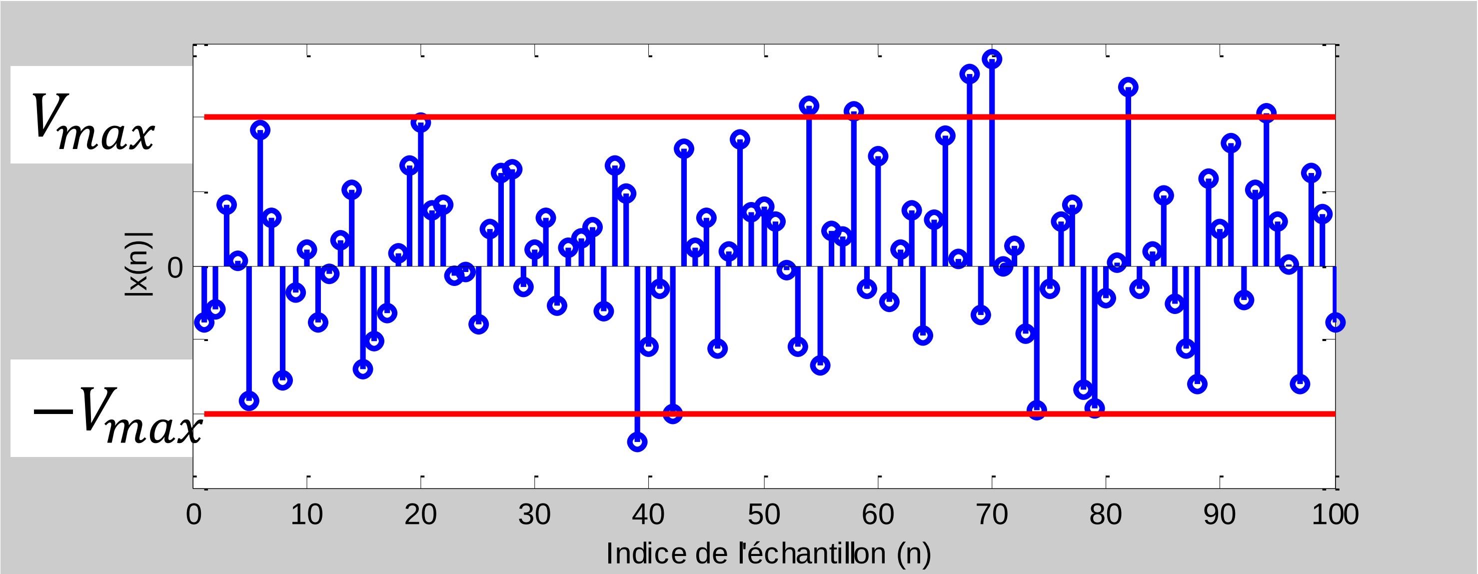
A sysmetric ADC provides a binary number of N bits over the range $2V_{max}$

Quantization step size $q = \frac{2V_{max}}{2^N}$

ADC/Quantization

Crest factor

Quantization imposes limits on large amplitudes.



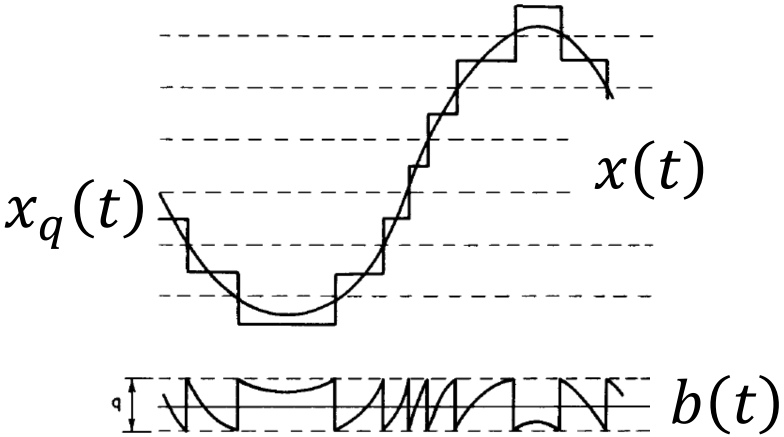
Crest factor : $F_c = 20log10(\frac{V_{max}}{V_{rms}})$

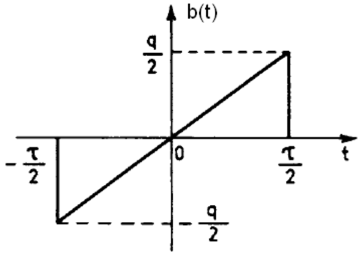
ADC/Quantization

Quantization error

$$x(t) \longrightarrow \boxed{\text{Sampling}} \xrightarrow{x_e(t)} \boxed{\text{Quantization}} \xrightarrow{x_q(t)} \boxed{\text{Coding}} \longrightarrow x(n)$$

Quantization : $b(t) \; = x_q(t) - x_e(t)$





Power of $b(t)$: $P_b = \frac{1}{\tau}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} b^2\,(t)dt = \frac{q^2}{12}$

ADC/Quantization

Signal to noise ratio (SNR)

For a signal quantized over N bits, the quantization step size is given by : $q = \frac{2V_{max}}{2^N}$

$$P_b = \frac{q^2}{12} = \frac{1}{12}\left(\frac{2V_{max}}{2^N}\right)^2 = \frac{1}{3}\frac{{V_{max}}^2}{2^{2N}} \rightarrow \frac{{V_{max}}^2}{P_b} = 3 \times 2^{2N}$$

$$\mathrm{SNR} = \frac{\text{signal power}}{\text{quantization noise power}}$$

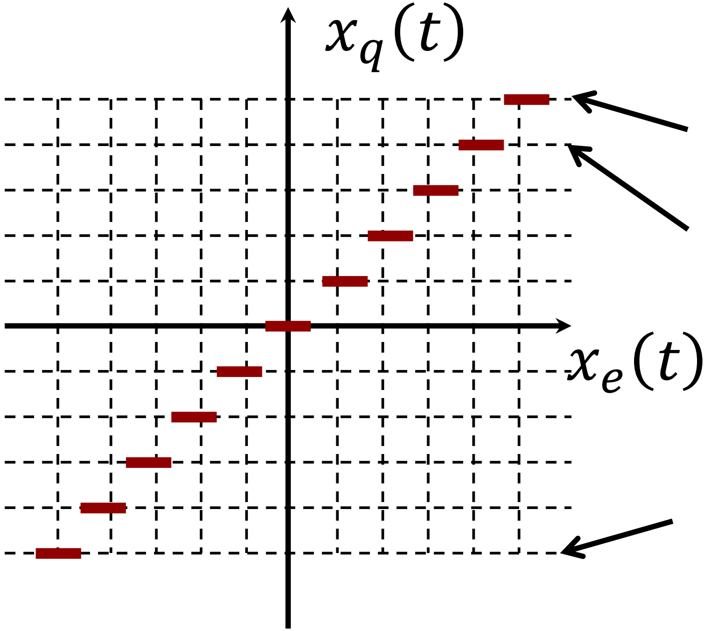
$$SNR = 10log10\left(\frac{P_x}{P_b}\right) = 10log10\left(\frac{{V_{max}}^2}{P_b}\frac{{V_{rms}}^2}{{V_{max}}^2}\right)$$
$$= 20Nlog10(2) + 10log10(3) - F_c$$
$$= 6{,}02N + 4{,}77 - F_c$$

ADC/Coding

Principle

$x(t) \rightarrow$ Sampling $\xrightarrow{x_e(t)}$ Quantization $\xrightarrow{x_q(t)}$ Coding $\rightarrow x(n)$

2^N levels: we make correspond to each level a binary code, written with only 0s and 1s



code $011\cdots111$

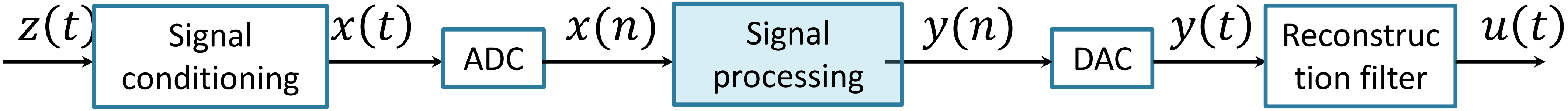
code $011\cdots110$

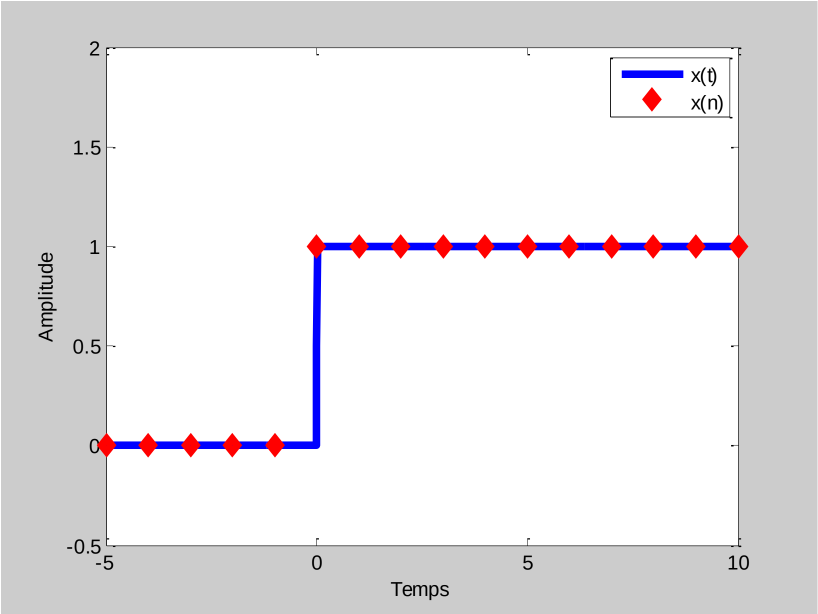
$\vdots \qquad \vdots$

$\vdots \qquad \vdots$

code $\underbrace{100\cdots000}_{N\ bits}$

Digital signal processing





Digital signal processing

Linear Time Invariant system : LTI system

$$x(n) \longrightarrow \boxed{\;H\;} \longrightarrow y(n)$$

A linear system processing a discrete sequence $\mathrm{x}(n)$, produces a discrete sequence $\mathrm{y}(n)$: $\mathrm{y}(n) = H\{x\ (n)\}$

<u>Linear system :</u>

$$H(\alpha x_1(n)+\beta x_2(n)) = \alpha H(x_1(n)) + \beta H(x_2(n))$$

<u>Time invariant system :</u>

$$H(x(n)) = y(n) \quad \Rightarrow \quad H(x(n-n_0)) = y(n-n_0)$$

Digital signal processin

LTI system, impulse response

Given a LTI system : $y(n) = H\{x\,(n)\}$

The input signal can be written as

$x(n) = \sum_{k=-\infty}^{+\infty} x_k \delta(n-k)$

$$y(n) = H(x(n)) = \sum_{k=-\infty}^{+\infty} x_k H(\delta(n-k)) = \sum_{k=-\infty}^{+\infty} x_k h(n-k) = \sum_{k=-\infty}^{+\infty} h_k x(n-k)$$

$$h(n-k) = H\{\delta\,(n-k)\}$$

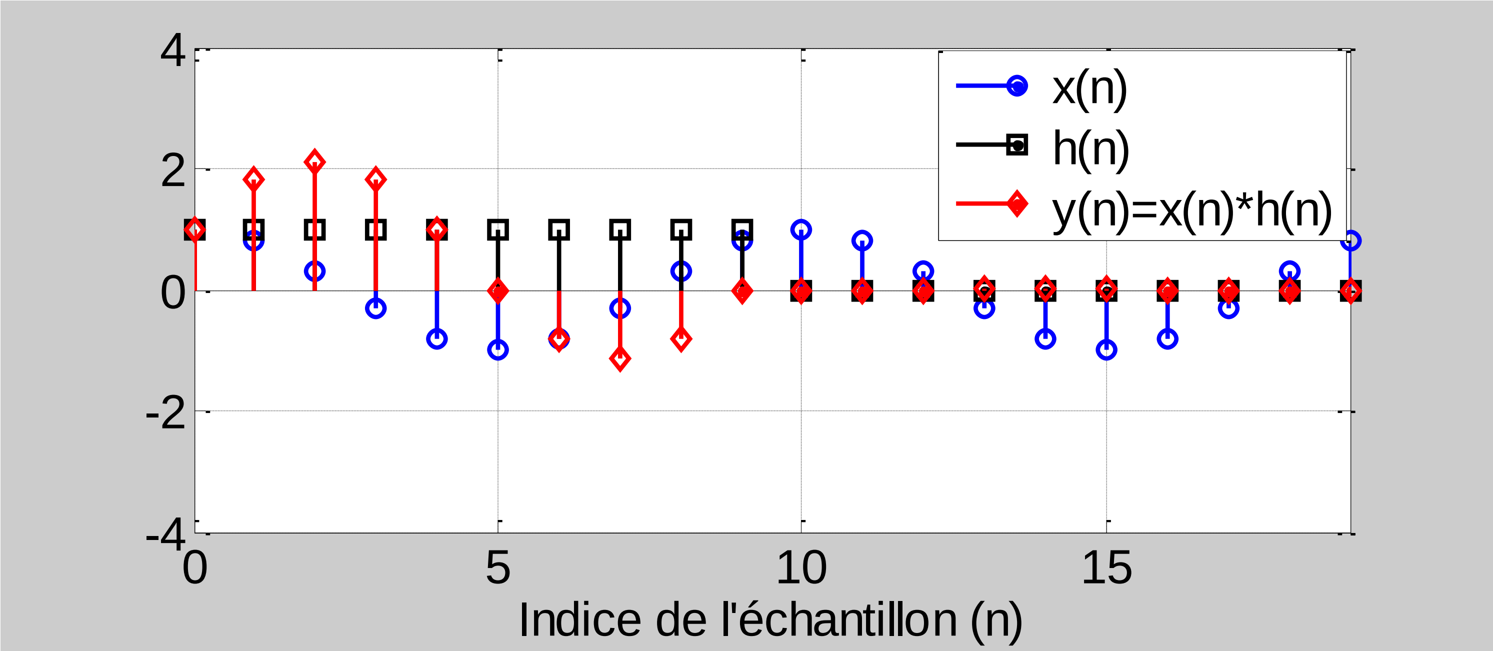
For the ${n_0}^{th}$ samples, we can write $y_{n_0} = (x * h)\,(n_0)$

$$y_{n_0} = \sum_{k=-\infty}^{+\infty} x_k h_{n_0-k} = \sum_{k=-\infty}^{+\infty} h_k x_{n_0-k}$$

Digital signal processing

LTI system, example

$$y(n) = x(n) * h(n) = h(n) * x(n) = \sum_{k=-\infty}^{+\infty} x_k h(n-k) = \sum_{k=-\infty}^{+\infty} h_k x(n-k)$$



Digital signal processing

FT applied to convolution

The Fourier transform applied to the convolution of two discrete signals :

$$y(n) = x(n) * h\,(n)$$

$$Y_e(f) = FT\{y(n)\} = \sum_{n=-\infty}^{+\infty} y_n\, e^{-j2\pi f n T_e} = \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} x_k h_{n-k}\, e^{-j2\pi f n T_e}$$

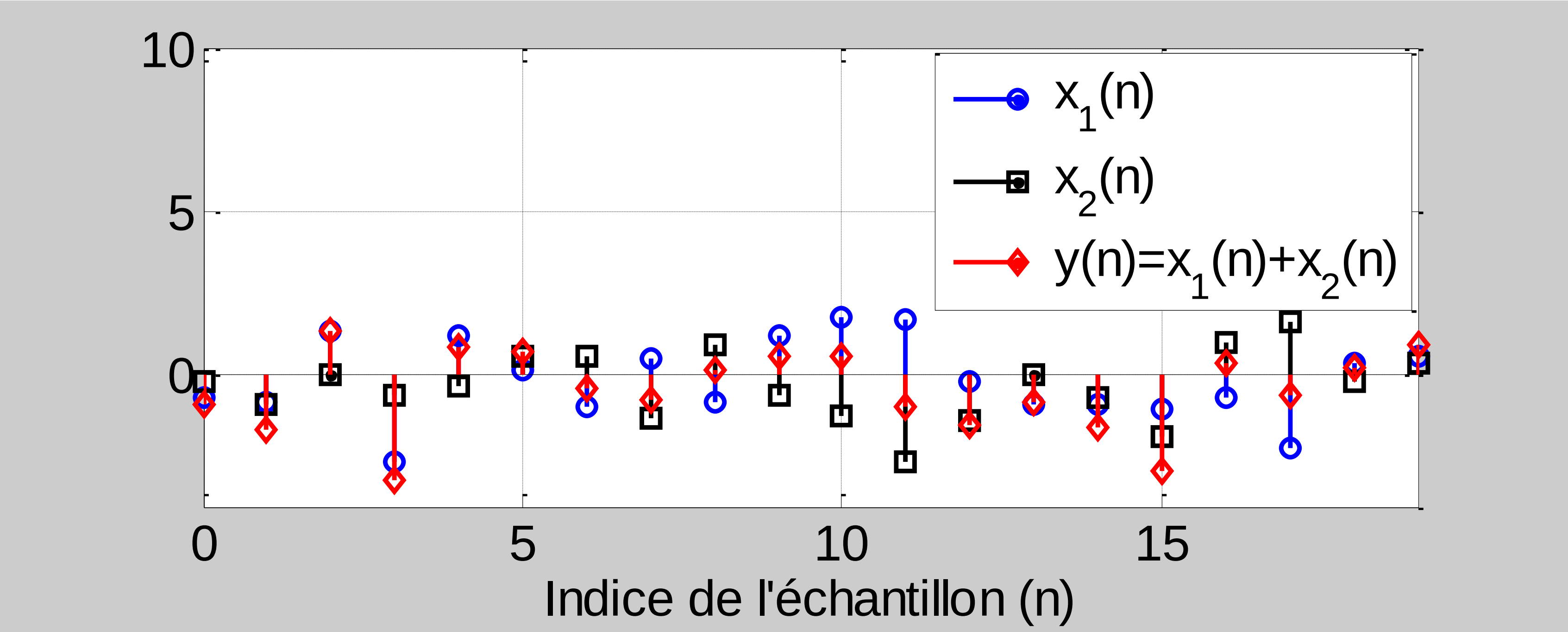
$$= \sum_{k=-\infty}^{+\infty} x_k e^{-j2\pi f k T_e} \sum_{n=-\infty}^{+\infty} h_{n-k}\, e^{-j2\pi f (n-k) T_e}$$

$$= X_e(f) H_e(f)$$

Where $X_e(f) = FT\{x(n)\}\; and\;\; H_e(f) = FT\{h(n)\}$

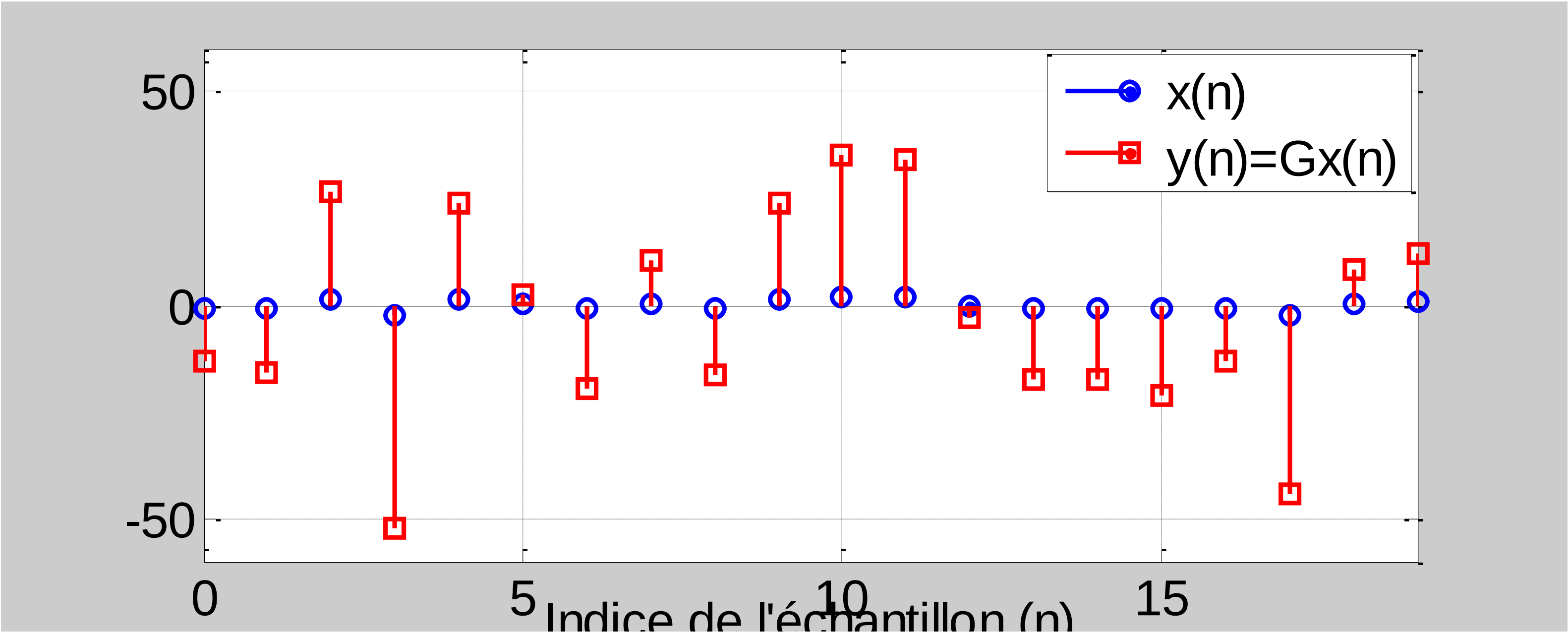
Digital signal processing

Sum of discrete signals



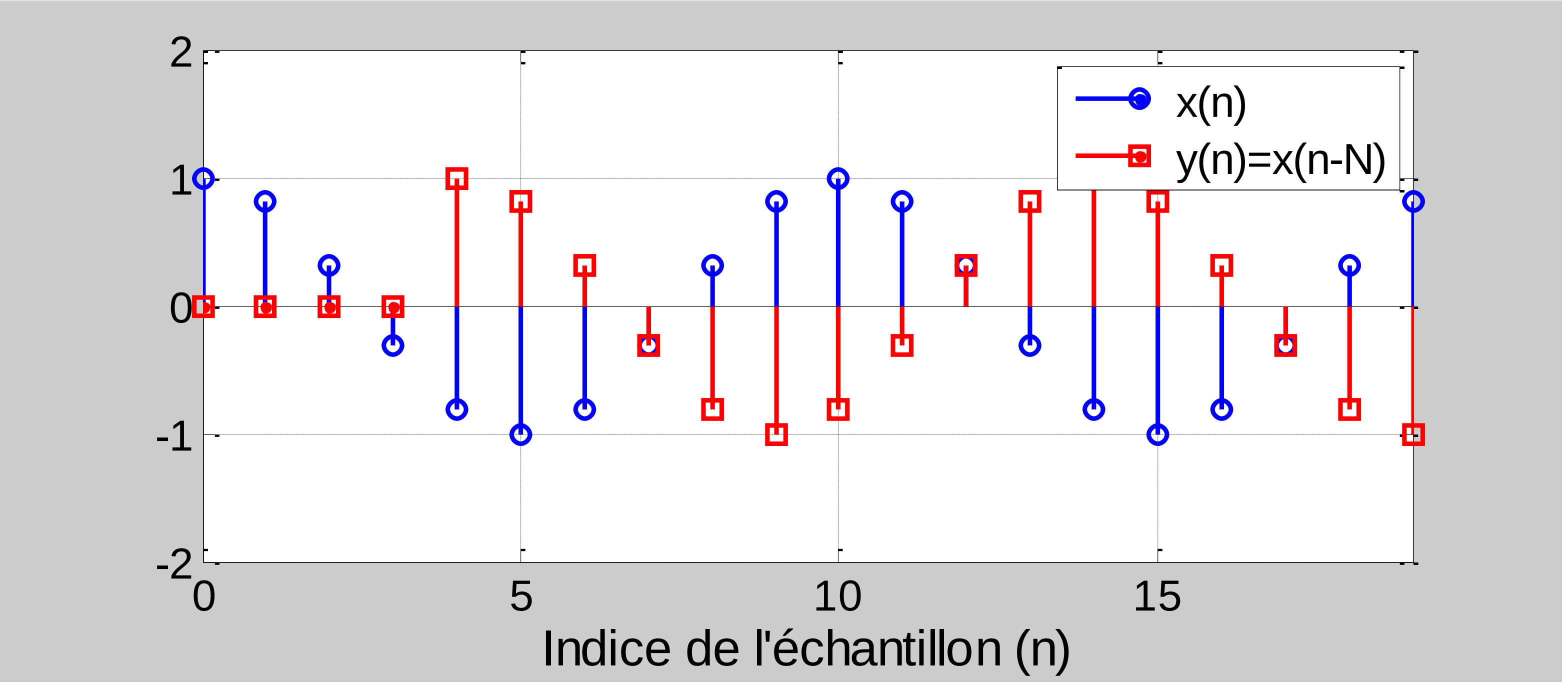
Digital signal processing

Multiplication by a scalar



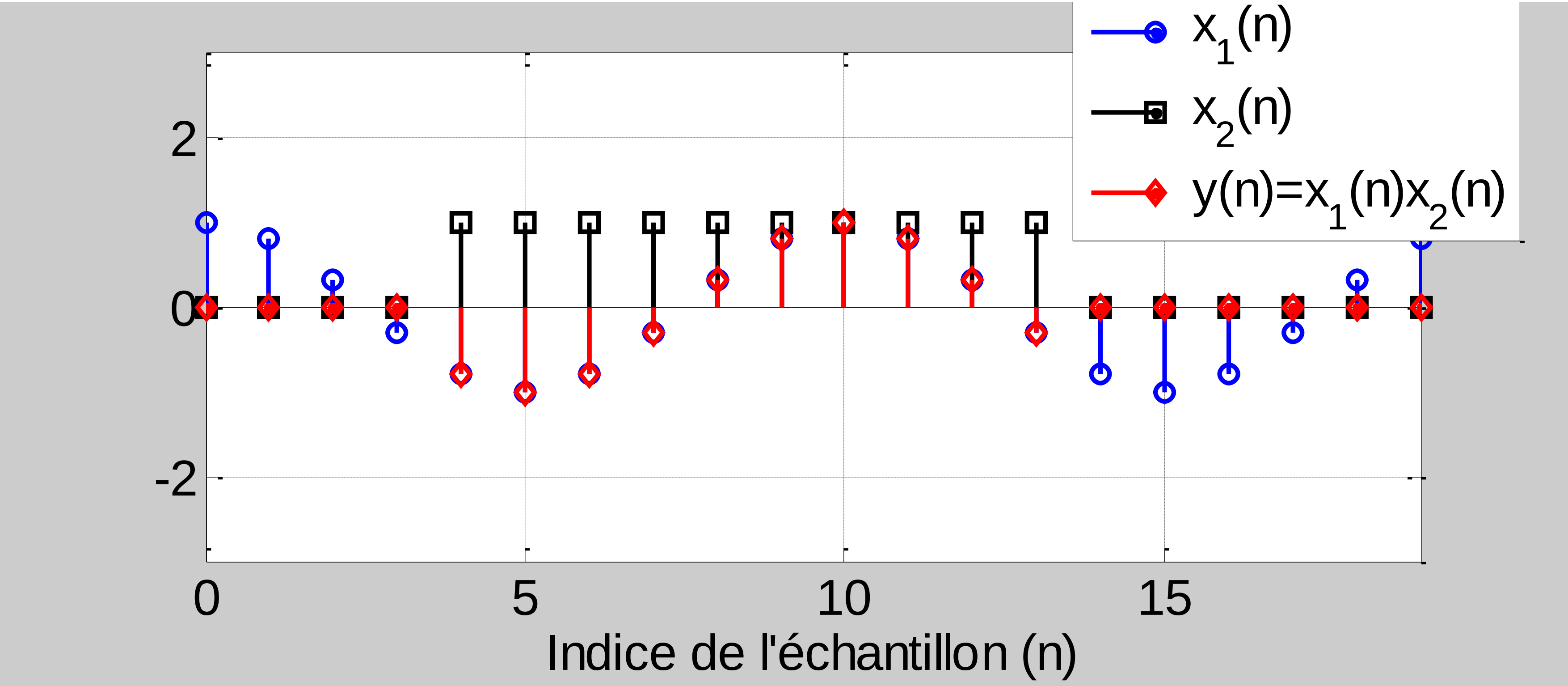
Digital signal processing

Delay



Digital signal processing

Multiplication of two signals

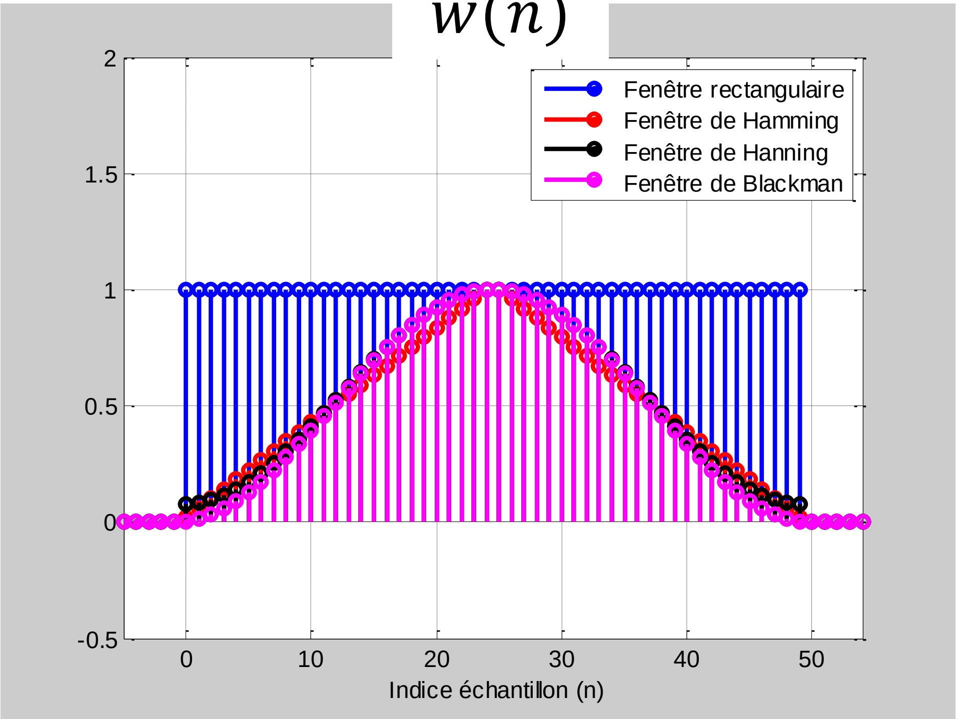


Digital signal processing

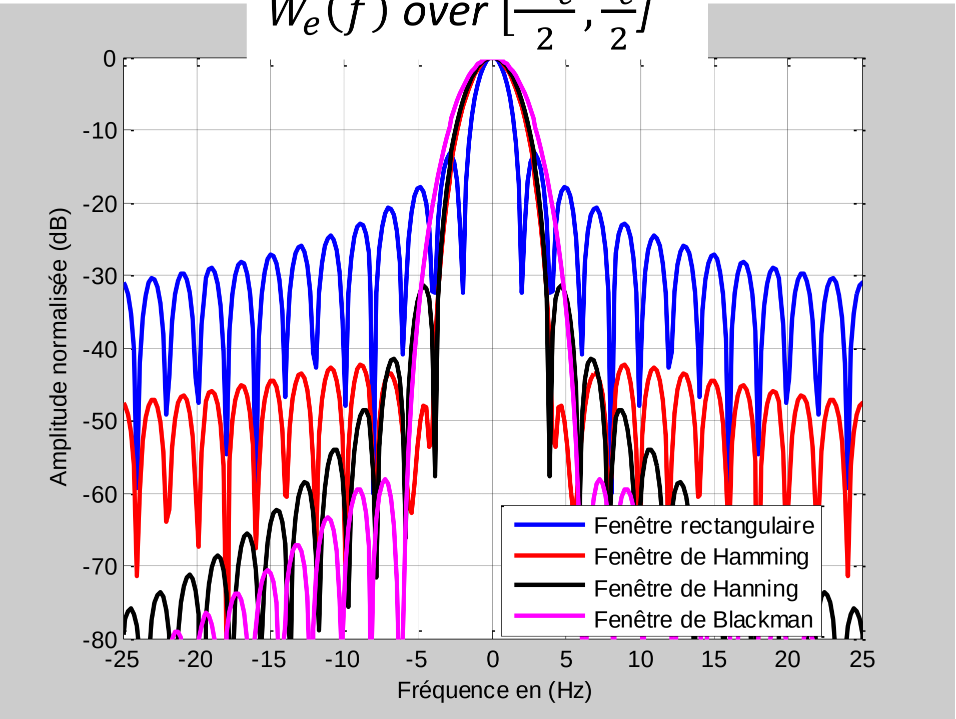
FT applied to the product of two discrte signals

$$FT\big(x(n)w(n)\big) = \sum_{n=0}^{N-1} x_n w_n\, e^{-j2\pi f n T_e} = X_e(f) * W_e(f)$$

$w(n)$



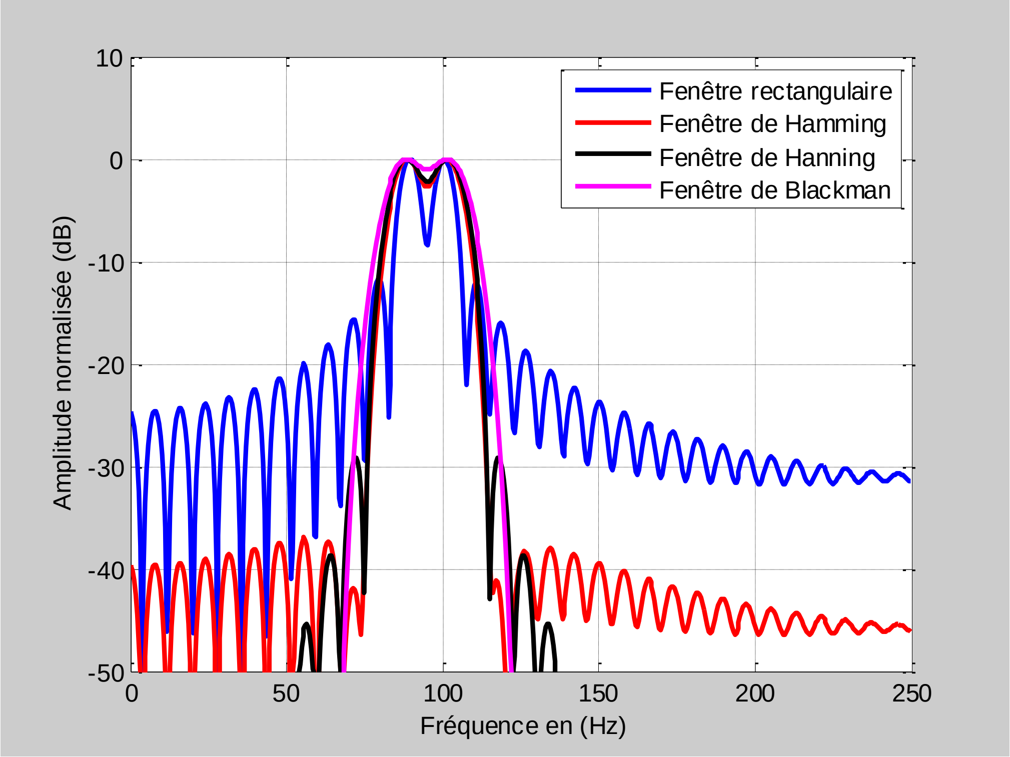
$W_e(f)$ over $[\frac{-F_e}{2}, \frac{F_e}{2}]$



Digital signal processing

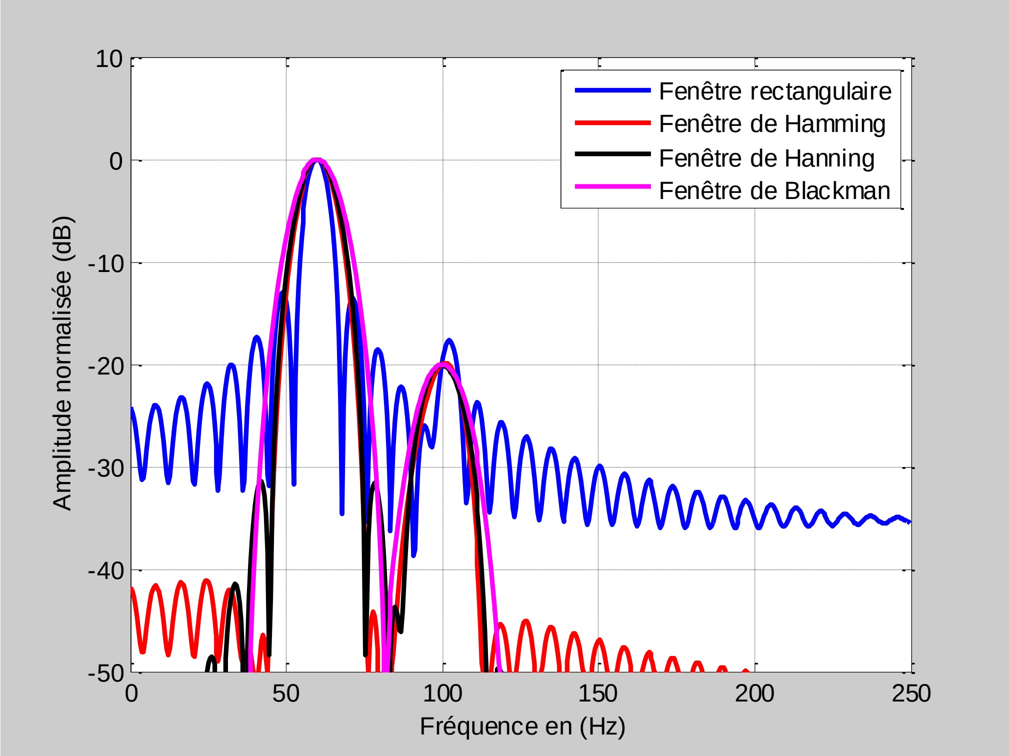
FT applied to the product of two discerte signals

$$x(n) = \cos(2\pi n \frac{90}{500}) + \cos(2\pi n \frac{100}{500})$$



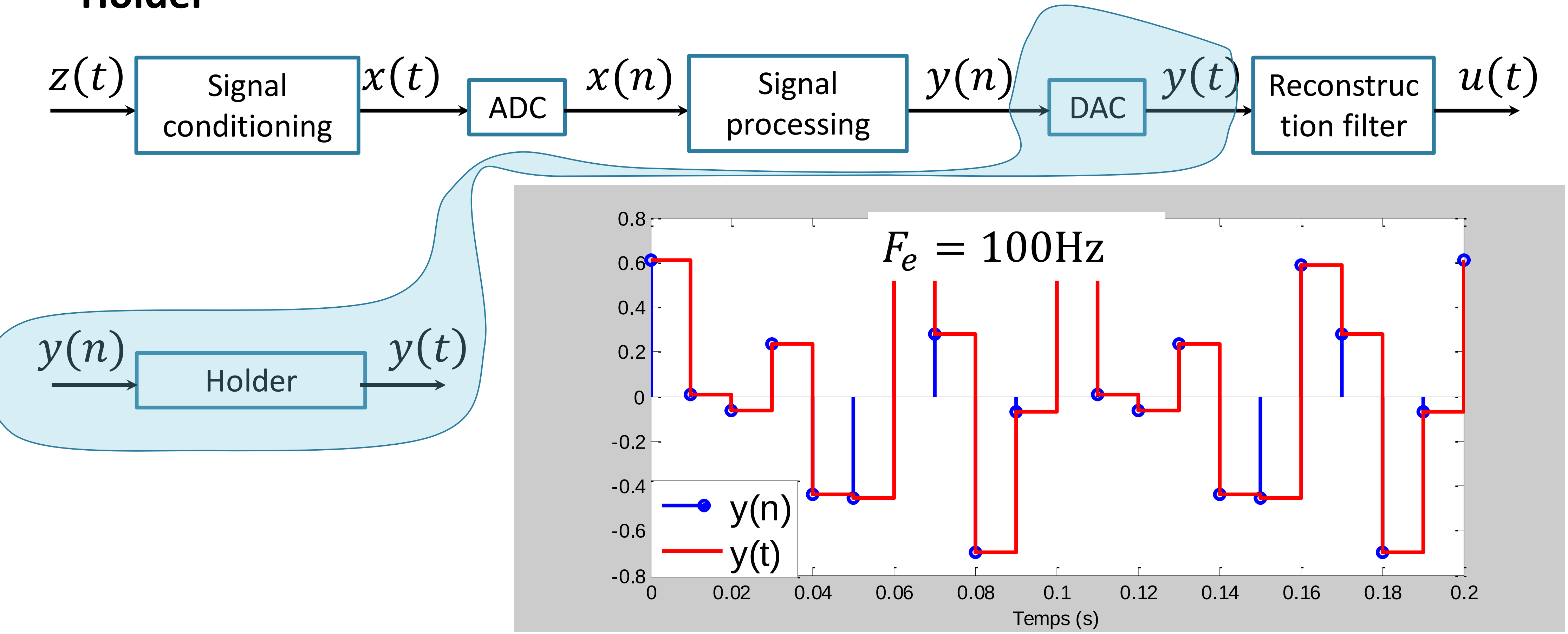
Digital signal processing
FT applied to the product of two discrte signals

$$x(n) = \cos(2\pi n\frac{90}{500}) + 0.1\cos(2\pi n\frac{100}{500})$$



DAC

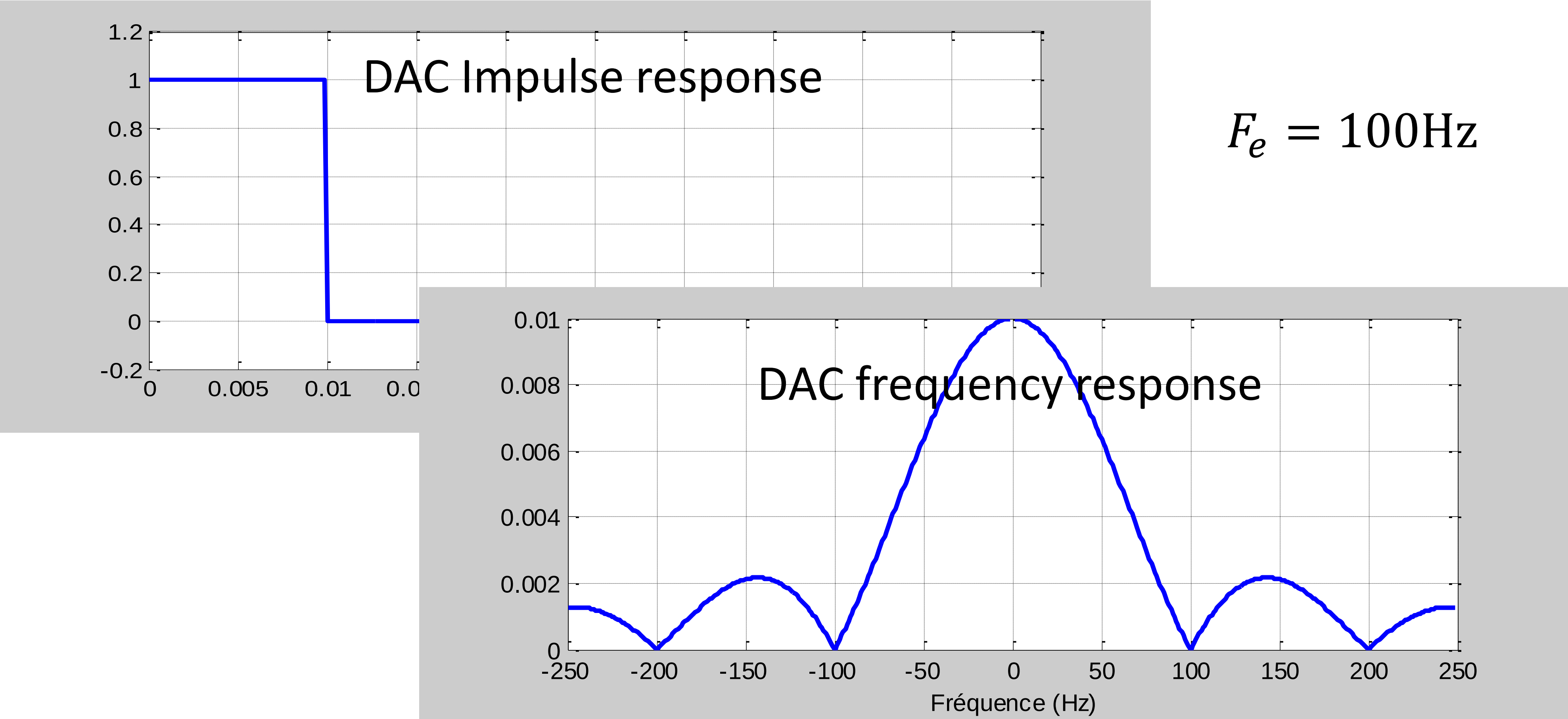
Holder



$z(t)$
Signal conditioning
$x(t)$
ADC
$x(n)$
Signal processing
$y(n)$
DAC
$y(t)$
Reconstruction filter
$u(t)$
$y(n)$
Holder
$y(t)$
$F_e = 100\text{Hz}$
y(n)
y(t)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
Temps (s)

DAC

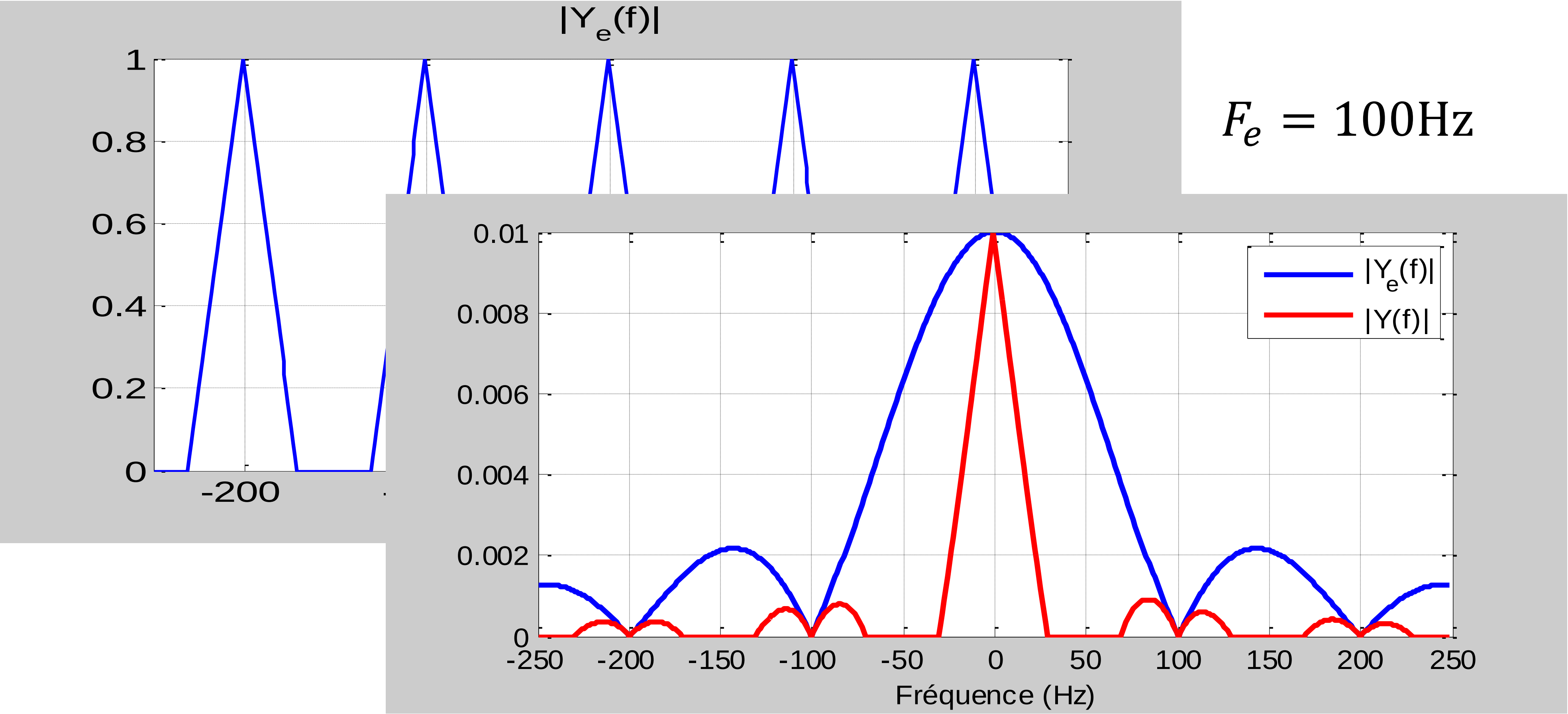
Holder, impulse and frequency responses



$$F_e = 100\text{Hz}$$

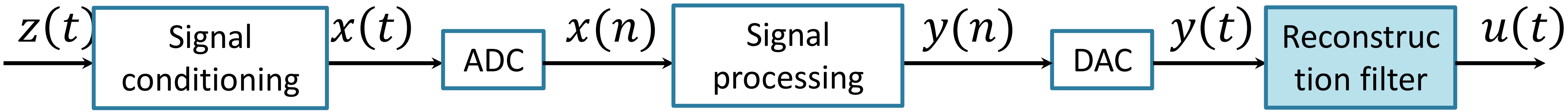
DAC

Input and output signals spectra

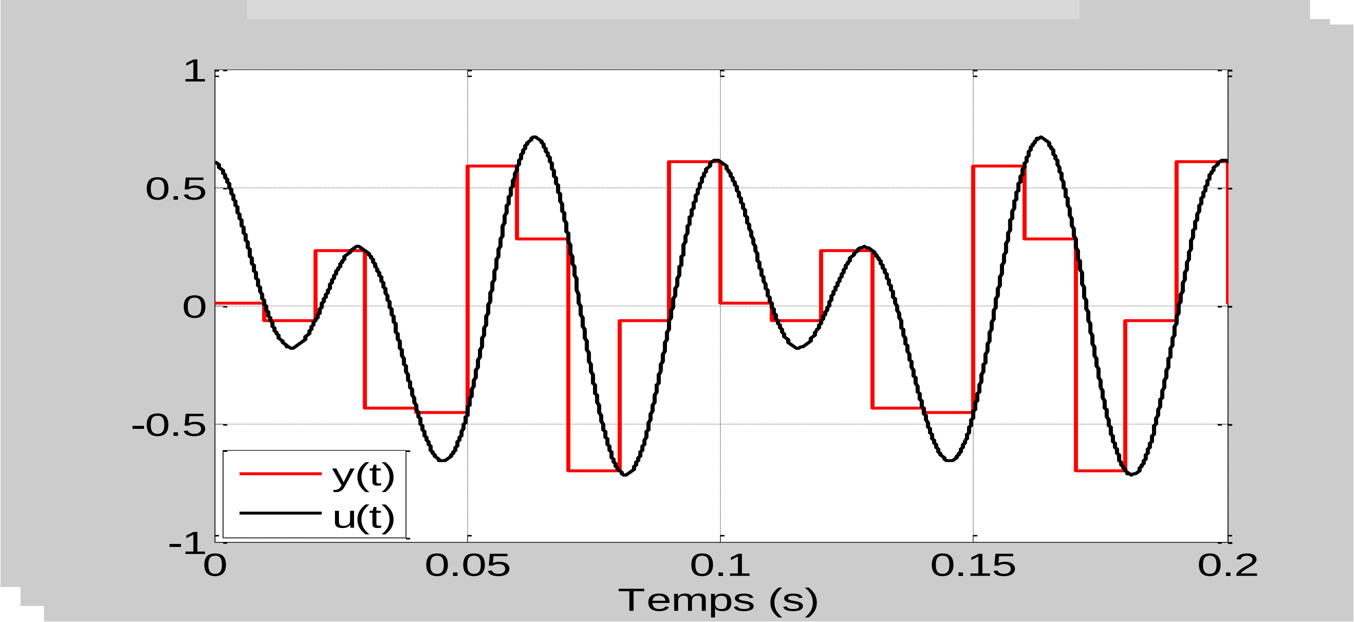


$F_e = 100\text{Hz}$

Reconstruction filter



The DAC provides a signal, which spectrum ranges from -∞ to +∞. A reconstruction filter (ideal low-pass filter) is needed to cut-off the frequencies outside $[-\frac{F_e}{2}, \frac{F_e}{2}]$



Outline

1. Data acquisition and analysis

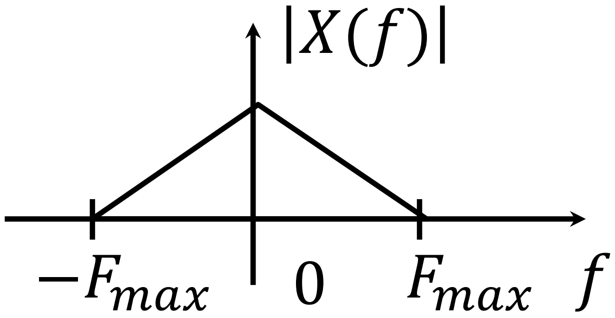
- Digital data acquisition system
- **Discrete Fourier Transform**
- Fast Fourier transform
- Z Transform and transfer function

Discrete Fourier transform (DFT)

Definition

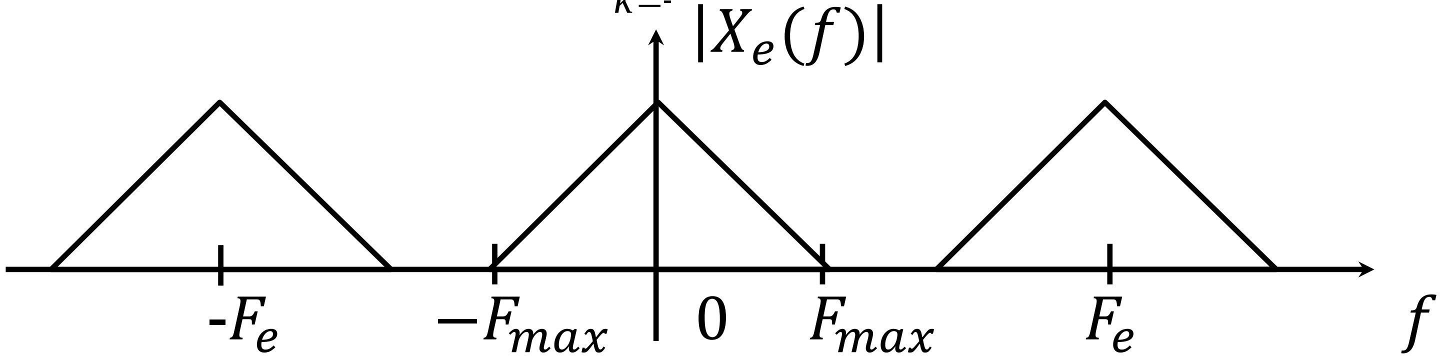
The FT of an analog signal $x(t)$:

$$X(f) = \int_{-\infty}^{-\infty} x(t) e^{-j2\pi ft} dt$$



The FT of the discrete séquence $x_e(t)$

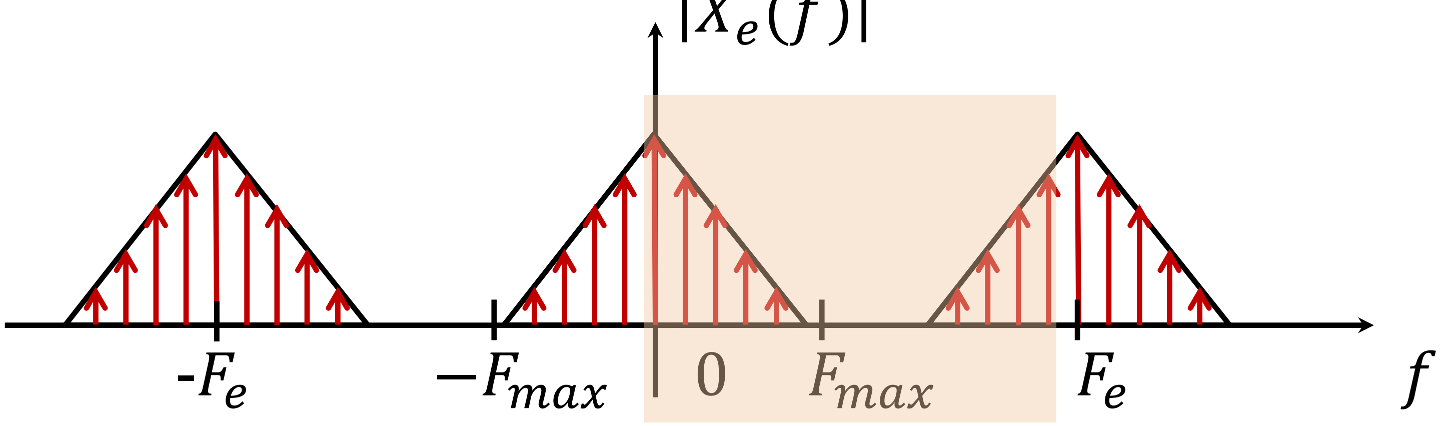
$$X_e(f) = F_e \sum_{k=-}^{+\infty} X(f - kF_e)$$



Discrete Fourier transform (DFT)

Definition

To compute the spectrum of a sampled signal, it is necessary to calculate an infinity of values → Important calculation resources. The idea of the DFT is to compute the Fourier transform only for a few frequency values: sample the TF in the frequency domain.



Discrete Fourier transform (DFT)

Definition

For a causal discrete sequence $\mathrm{x(n)} = \{x_0, x_1, x_2, \ldots x_{N-1}\}$, the Fourier transform is given by :

$$X(f) = \sum_{n=0}^{N-1} x_n e^{-j2\pi f n T_e}$$

Compute $\mathrm{X(f)}$ for finite number (N) of frequencies f_k

$$f_k = \frac{kF_e}{N} = \frac{k}{NT_e}$$

$$X_k = X(f_k) = \sum_{n=0}^{N-1} x_n e^{-j2\pi n f_k} = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}$$

Sampled spectrum $\mathrm{X}(k) = \{X_k = X(f_k)\}_{k\in[0,N-1]}$

Discrete Fourier transform (DFT)

Properties

The DFT

$$X(k) = DFT(x(n)) = \sum_{n=0}^{N-1} x_n e^{-j2\pi\frac{nk}{N}}$$

$$X(k) = \{X_0, X_1, \ldots, X_{N-1}\}$$

The Inverse DFT (IDFT)

$$x(n) = IDFT(X(k)) = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j2\pi\frac{nk}{N}}$$

$$x(n) = \{x_0, x_1, \ldots, x_{N-1}\}$$

<u>Linearity</u> : $DFT(\alpha x(n) + \beta y(n)) = \alpha X(k) + \beta \mathrm{Y(k)}$

<u>Periodicity</u> : $X(k+N) = X(k)$

Discrete Fourier transform (DFT)

Properties

Tilme shift : $Y(k) = DFT\left(x(n-n_0)\right) = DFT\left(x(n)\right)e^{-j2\pi\frac{n_0 k}{N}} = X(k)e^{-j2\pi\frac{n_0 k}{N}}$

Symetry : if $x(n)$ is real $X^*(k) = X(-k) = X(N-k)$

Parseval's identity :

$$P_x = \frac{1}{N}\sum_{n=0}^{N-1}\left|x_n\right|^2 = \frac{1}{N}\sum_{n=0}^{N-1}x_n x_n^* = \frac{1}{N}\sum_{n=0}^{N-1}x_n\left(\frac{1}{N}\sum_{k=0}^{N-1}X_k^* e^{-j2\pi\frac{kn}{N}}\right)$$

$$= \frac{1}{N^2}\sum_{k=0}^{N-1}X_k^*\sum_{n=0}^{N-1}x_n e^{-j2\pi\frac{kn}{N}} = \frac{1}{N^2}\sum_{k=0}^{N-1}\left|X_k\right|^2$$

Discrete Fourier transform (DFT)

Properties

<u>Convolution</u> : $x(n) \longrightarrow \boxed{h(n)} \longrightarrow y(n) = x(n) * h(n)$

$$Y(k) = DFT\left(y(n)\right) = \sum_{n=0}^{N-1} y_n e^{-j2\pi\frac{kn}{N}} = \sum_{n=0}^{N-1}\sum_{i=0}^{N-1} x_i h_{n-i} e^{-j2\pi\frac{kn}{N}}$$

$$= \sum_{n=0}^{N-1} h_{n-i} e^{-j2\pi\frac{k(n-i)}{N}} \sum_{i=0}^{N-1} x_i e^{-j2\pi\frac{ki}{N}} = X(k)H(k)$$

$$\xrightarrow{x(n)} \boxed{h(n)} \xrightarrow{x(n)*h(n)} \boxed{g(n)} \xrightarrow{x(n)*h(n)*g(n)}$$

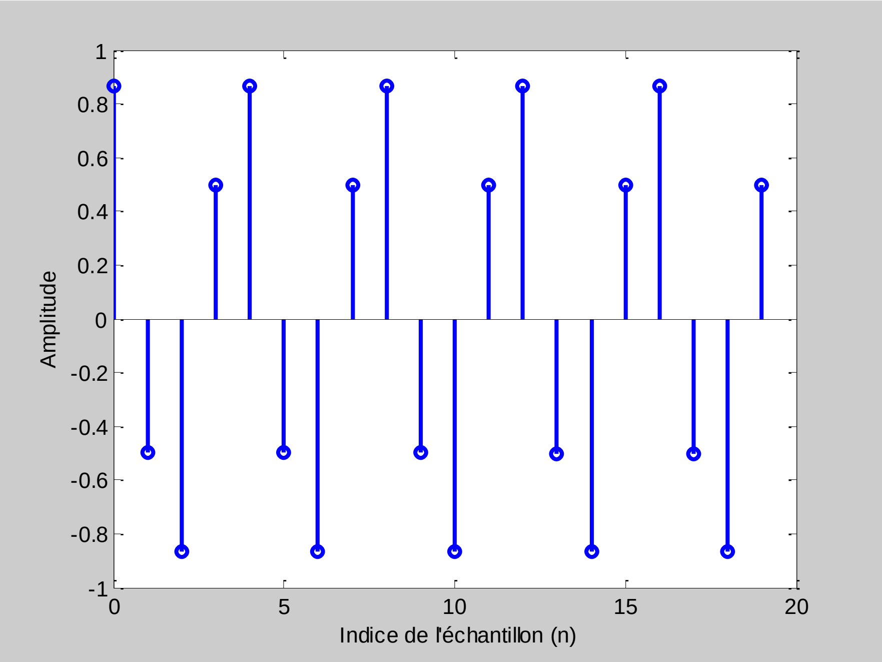
$$\xrightarrow{X(k)} \boxed{H(k)} \xrightarrow{X(k)H(k)} \boxed{G(k)} \xrightarrow{X(k)H(k)\,G(k)}$$

Discrete Fourier transform (DFT)

Example 1

We consider $N = 20$ samples of the signal $x(\mathrm{n}) = cos\left(\frac{\pi n}{2} + \frac{\pi}{6}\right)$

This signal is samples at $F_e = 20Hz$. The frequency of this sinusoidal signal is $f_1 = {}^{F_e}/_4 = 5\mathrm{Hz}$.

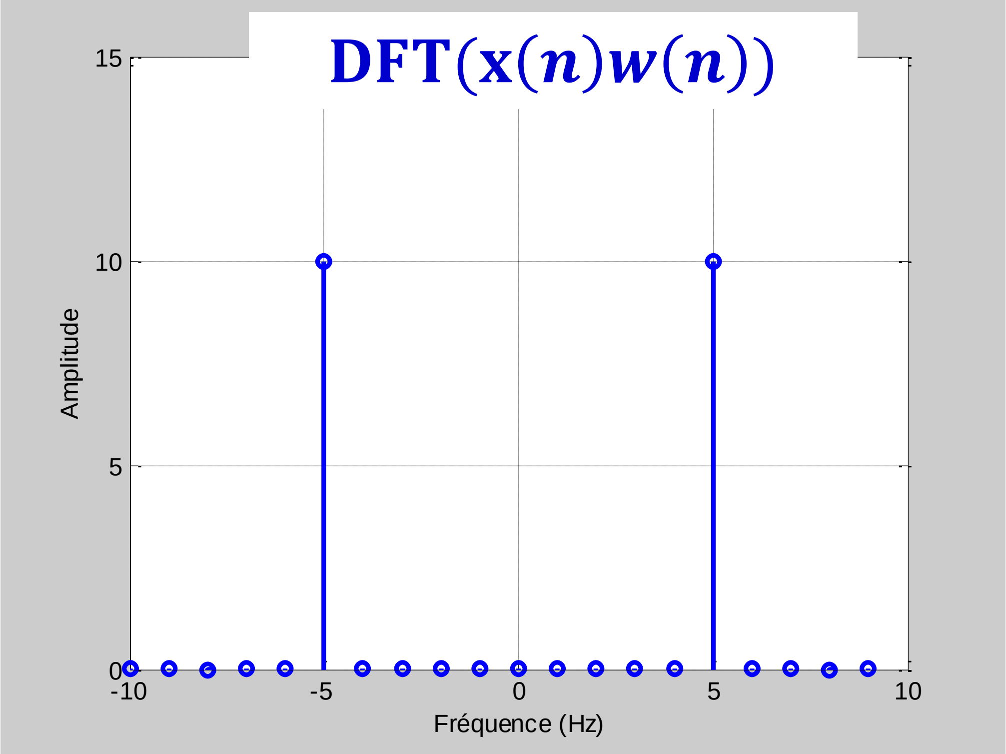


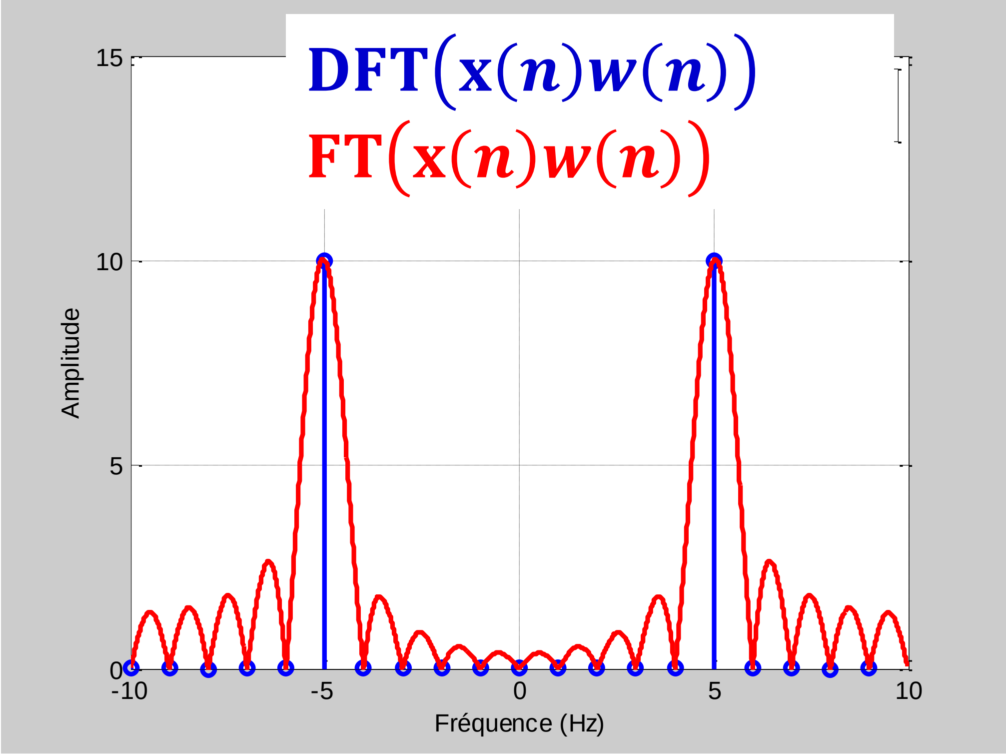
Discrete Fourier transform (DFT)

Example 1

DFT over $N = 20$ points

$$Spectra\ over\ [\frac{-F_e}{2}, \frac{F_e}{2}]$$



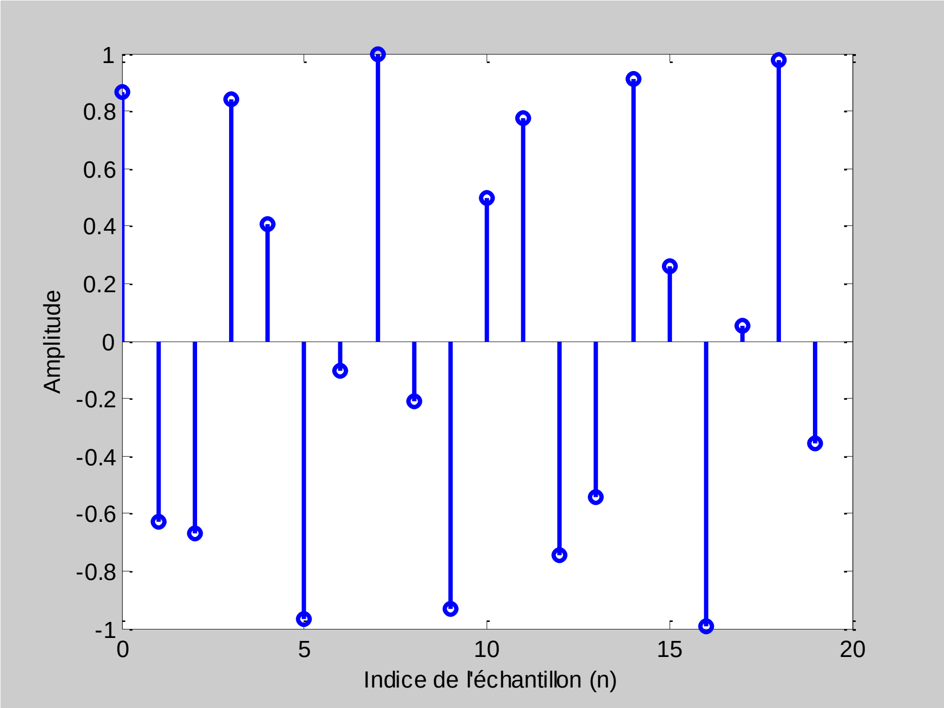


Discrete Fourier transform (DFT)

Example 2

We keep the same sampling frequency as chosen in example 1 : $F_e = 20Hz$ and change the frequency of the sinusoidal signal to $f_1 = 5{,}5\text{Hz}$.

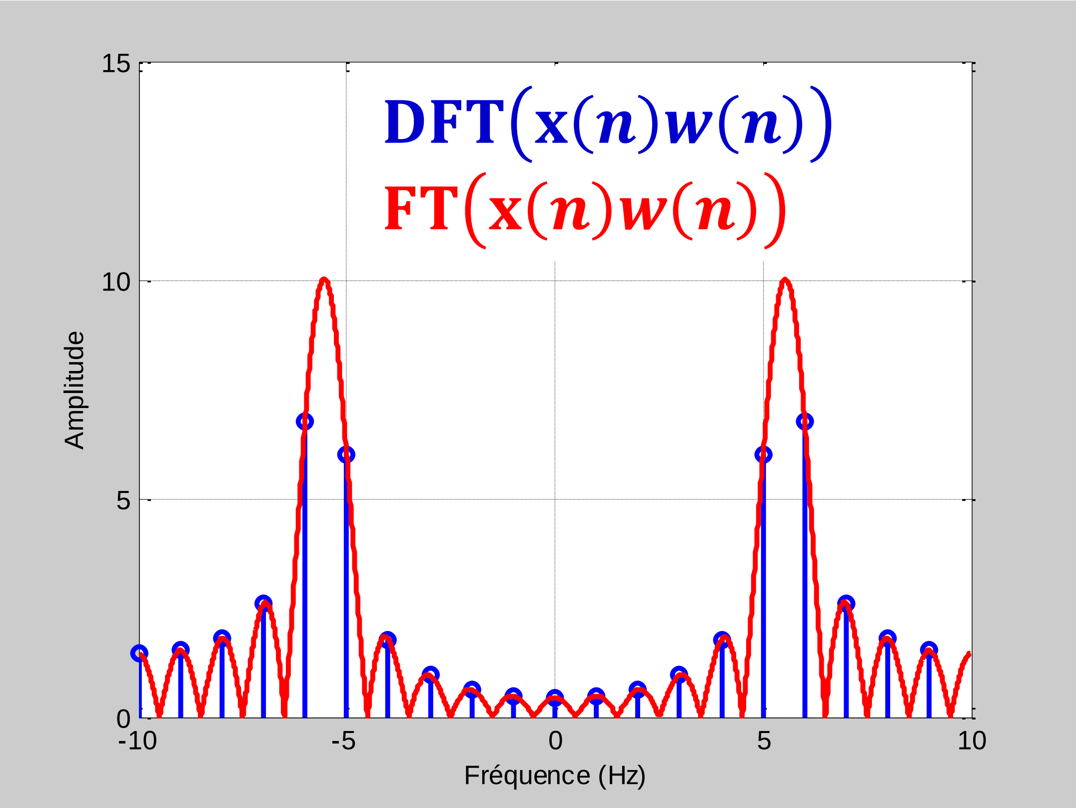
$$x(\text{n}) = cos\left(\frac{11\pi n}{20} + \frac{\pi}{6}\right)$$



Discrete Fourier transform (DFT)

Example 2

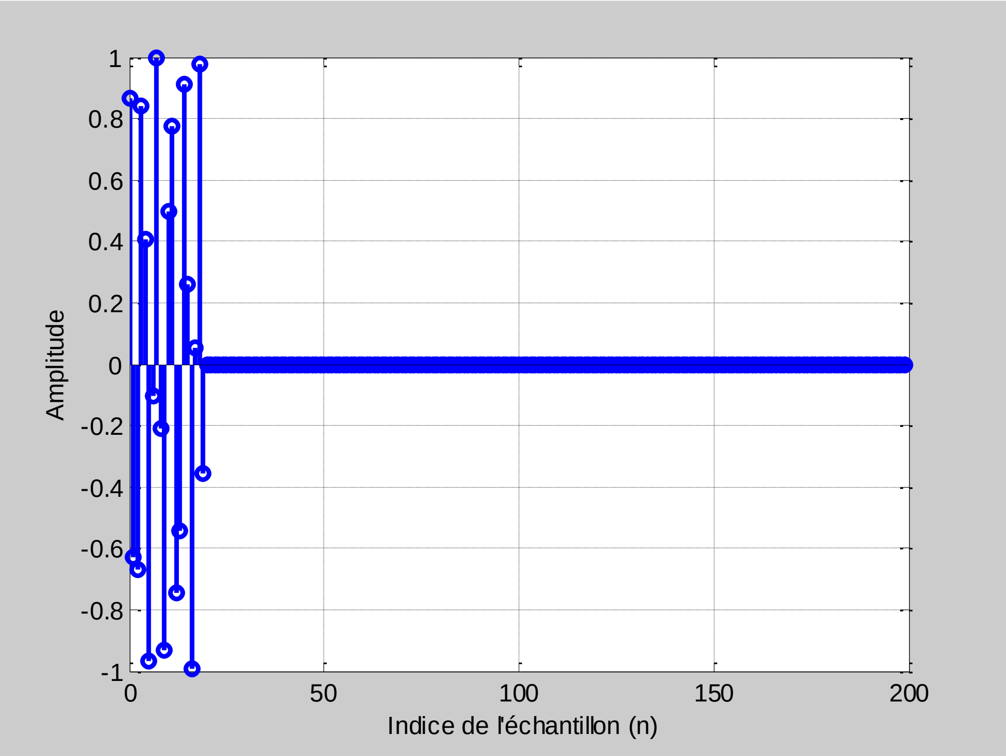
The spectrum samples no longer correspond to maxima and zeros.

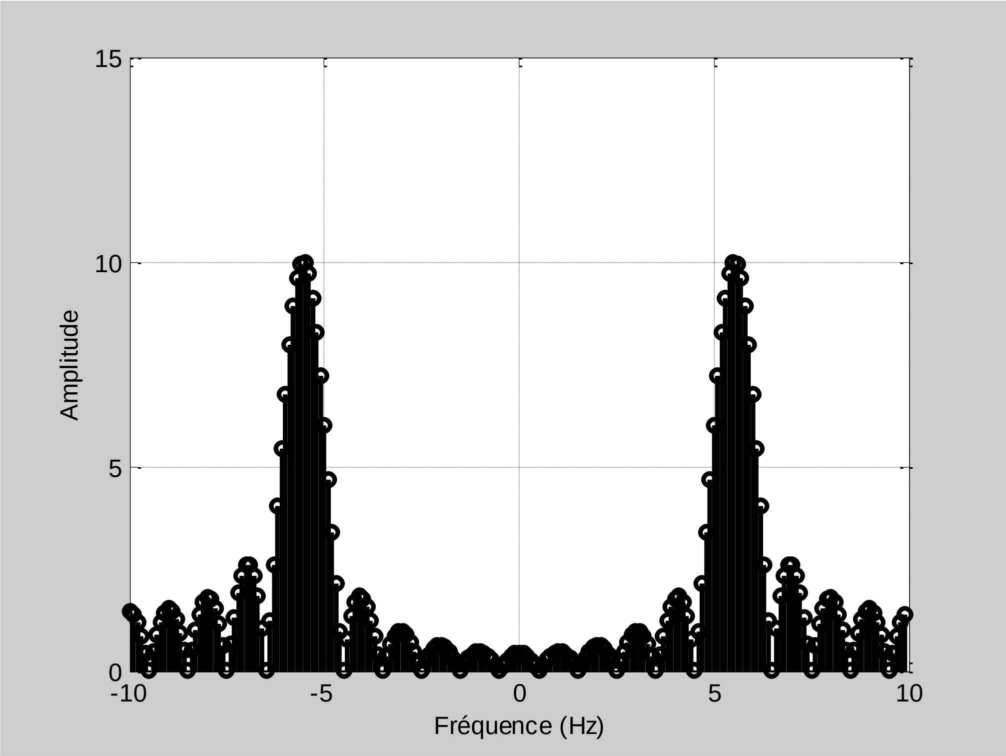


Discrete Fourier transform (DFT)

Example 2

Increase the DFT resolution by : « zero padding »





Outline

1. **Data acquisition and analysis**
   - Digital data acquisition system
   - Discrete Fourier Transform
   - **Fast Fourier transform**
   - Z Transform and transfer function

From DFT to FFT

$$W_N = e^{-j\frac{2\pi}{N}}$$

The DFT of x(n) : $$X(k) = \sum_{n=0}^{N-1} x_n W_N^{nk}$$

The IDFT of X(k) : $$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_k W_N^{-nk}$$

The problem of DFT is the high computation complexity.
Example : A DFT over N points requires N^2 Multiplication and $N(N-1)$ addition, $\sim N^2$ Multiplication and Accumulation (MAC).
For $N = 1024$ and $F_e = 48\ kHz$, a processor should carry $50\ 10^9$ MAC/s.

The Fast Fourier Transform (FFT)

The Fast Fourier Transform, is proposed to reduce the computation complexity of the DFT. It's based on some propriities of W_N:

- Symetry with respect to the real axis :

$$W_N^{-k} = (W_N^k)^*$$

- Symetry with respect to the origin :

$$W_N^k = -W_N^{N/2+k}$$

The Fast Fourier Transform (FFT)

We suppose that N is a power of 2.

$$X(k) = \sum_{n=0}^{N-1} x_n W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{(2n+1)k}$$

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{2nk}$$

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_{N/2}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_{N/2}^{nk}$$

$$X(k) = X_P(k) + W_N^k X_I(k)$$

The Fast Fourier Transform (FFT)

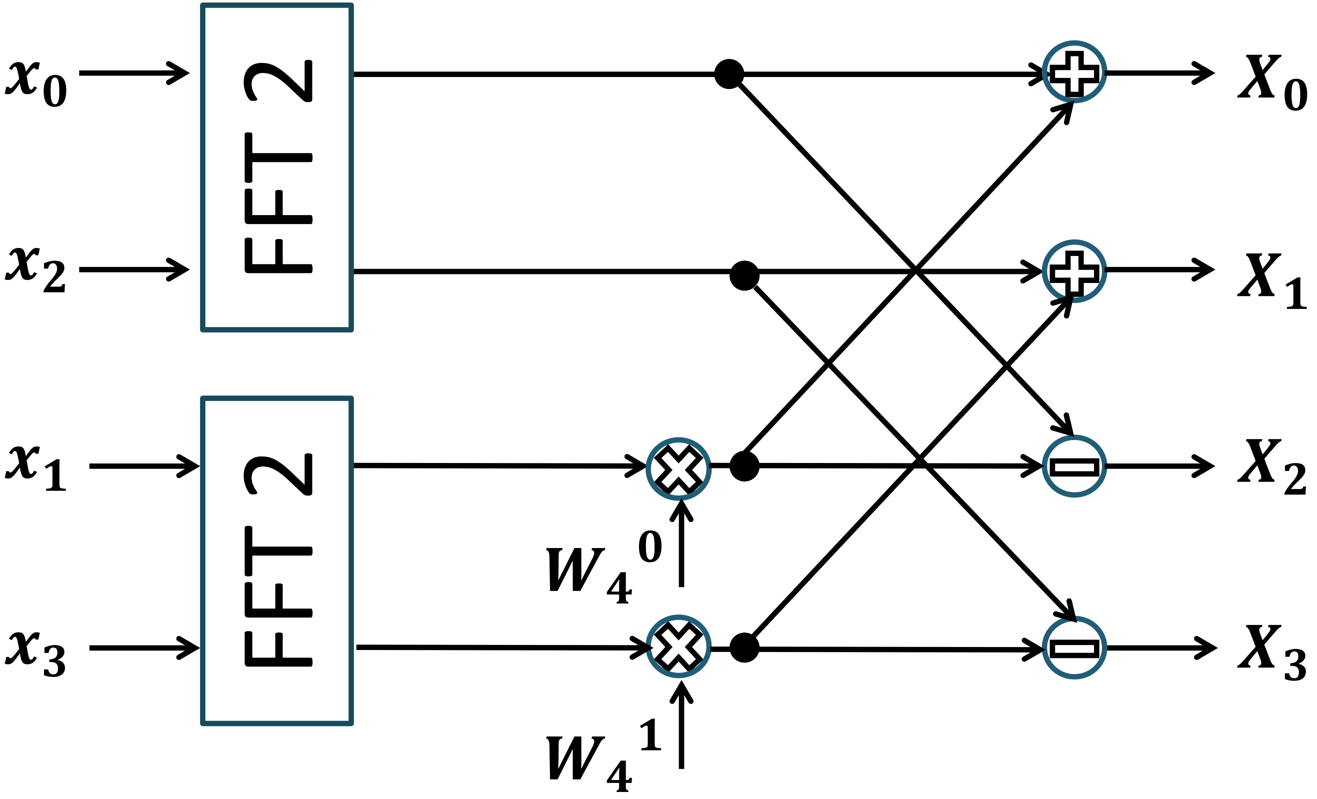
$$\begin{cases} W_N^{k+N/2} = -W_N^k \\ X_P(k+N/2) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_{N/2}^{nk} = X_P(k) \\ X_I(k+N/2) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_{N/2}^{nk} = X_I(k) \end{cases}$$

$$X(k) = \underbrace{X_P(k)}_{\text{FFT N/2}} + W_N^k \underbrace{X_I(k)}_{\text{FFT N/2}}$$

$$For\ 0 \le k < \frac{N}{2} - 1 \qquad \begin{cases} X(k) = X_P(k) + W_N^k X_I(k) \\ X(k+N/2) = X_P(k) - W_N^k X_I(k) \end{cases}$$

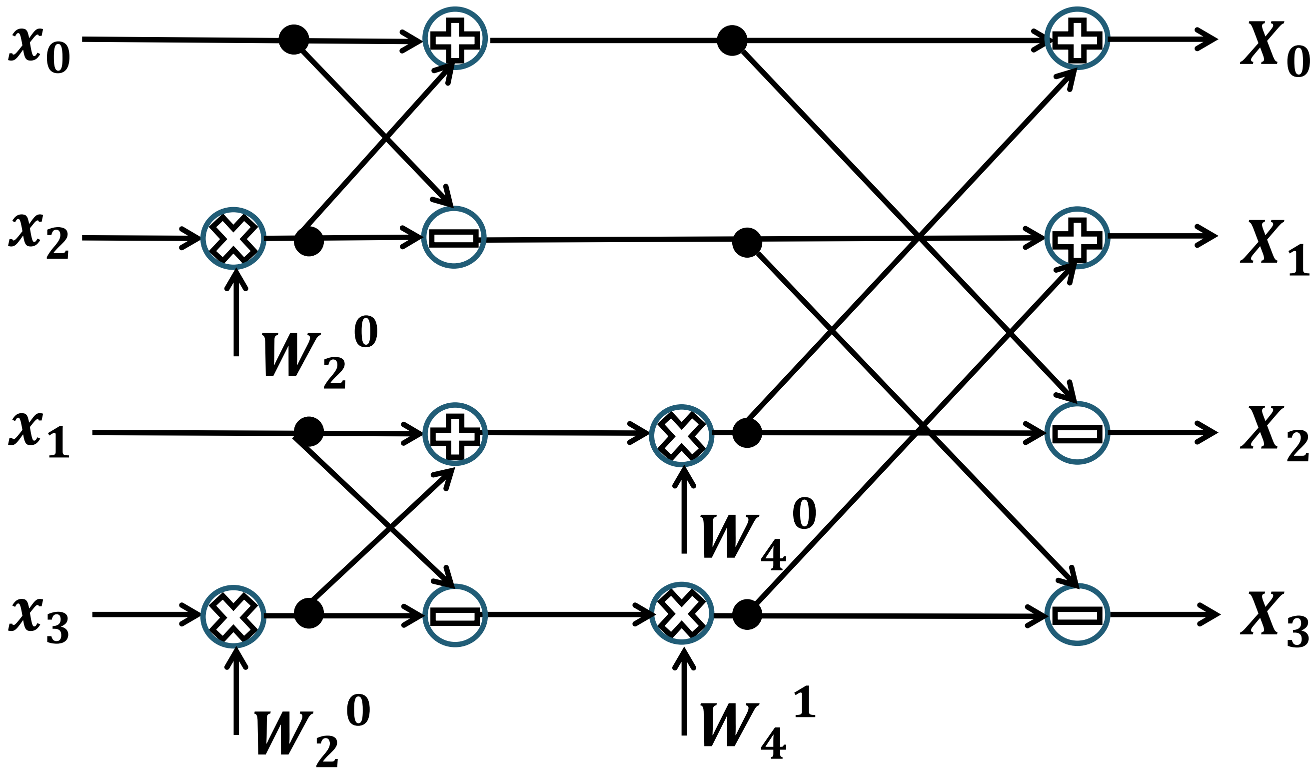
FFT

Example with $N = 4$



FFT

Example with $N = 4$



FFT

Example with $N = 1024 = 2^{10}$

- N is a power of 2. The computation of the N FFT values is achieved through $log_2(N)$ steps. In each step the computation requires N/2 multiplication et N addition.
- For an N points FFT, we need MN/2 multiplication et MN addition.

	DFT	**FFT**
Number of multiplication	N^2	$Nlog_2(N)/2$
Number of addition	$N^2 - N$	$Nlog_2(N)$

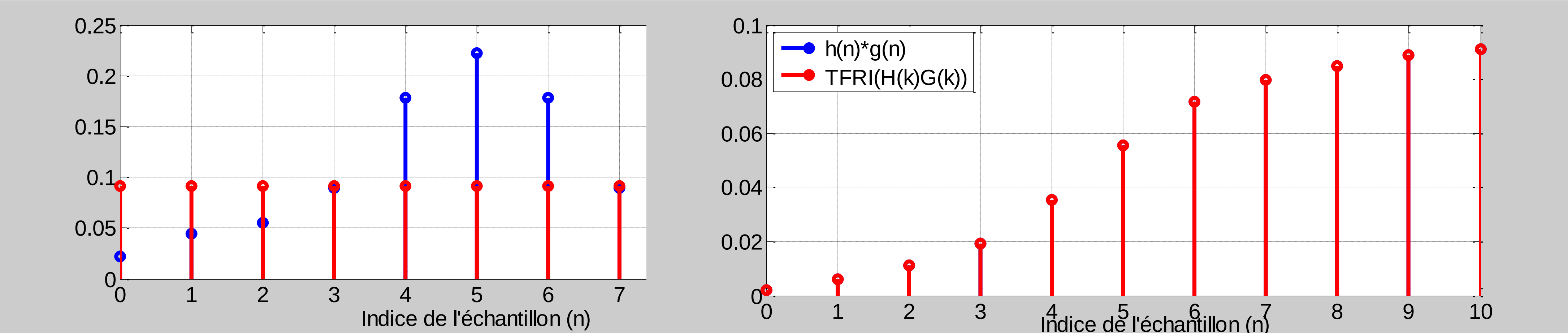
N**=1024**	**DFT**	**FFT**
Number of multiplication	$\sim 10^6$	5120
Number of addition	$\sim 10^6$	10240

FFT

Impulse response of two cascaded systems

$$\rightarrow \boxed{h(n)} \rightarrow \boxed{g(n)} \rightarrow \quad \xrightarrow{FFT} \quad \rightarrow \boxed{H(k)} \rightarrow \boxed{G(k)} \rightarrow$$

$$\leftarrow \boxed{\mathrm{h}(n) * g(n)} \leftarrow \quad \xleftarrow{IFFT} \quad \leftarrow \boxed{H(k)\,G(k)} \leftarrow$$



Outline

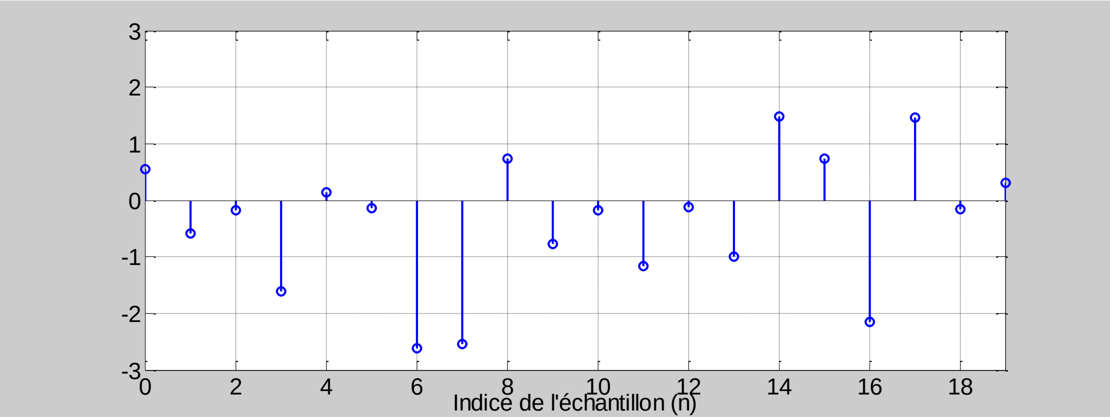
1. Data acquisition and analysis

- Digital data acquisition system
- Discrete Fourier Transform
- Fast Fourier transform
- **Z Transform and transfer function**

Z transform (ZT)

Definition

A discrete signal (sampled) can be represented by the following samples $\mathrm{x}(n) = \{x_0, x_1, x_2, x_3, x_4, x_5, \ldots\}$



This representation $\{x_n\}$ is difficult to handle

To have a better (simplified) representation, we transform the signal $\mathrm{x}(n)$ into another mathematical tool, with easier handling.

$$X(z) = ZT(x(n)) = \sum_{n=-\infty}^{+\infty} x_n \, z^{-n}$$

Delay operator z^{-1}

Z transform

Properties

<u>Linearity</u> :

$$ZT(\alpha x(n) + \beta y(n)) = \alpha X(z) + \beta Y(z)$$

<u>Time shift</u> :

$$ZT(x(n - n_0)) = z^{-n_0} X(z)$$

To delay a signal with $n_0 T_e \rightarrow$ multiply the ZT by z^{-n_0}

<u>Initial value theorem (for causal signal):</u>

$$\lim_{z \to +\infty} X(z) = x_0$$

<u>Final value theorem (for causal signal):</u>

$$\lim_{n \to +\infty} x(n) = \lim_{z \to 1} (z - 1) X(z)$$

Z transform
Properties

Convolution :

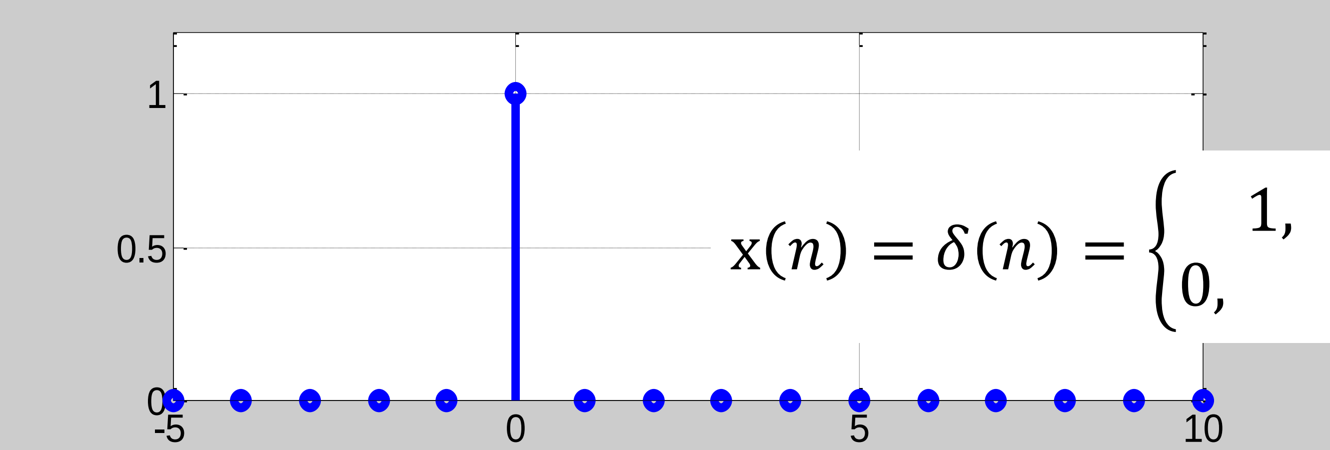
$$ZT(x(n)*y(n)) = ZT\left(\sum_{k=-\infty}^{+\infty} x_k y_{n-k}\right) = \sum_{n=-\infty}^{+\infty}\left(\sum_{k=-\infty}^{+\infty} x_k y_{n-k}\right) z^{-n}$$

$$= \sum_{k=-\infty}^{+\infty} x_k z^{-k} \sum_{n=-\infty}^{+\infty} y_{n-k} z^{-(n-k)} = X(z)Y(z)$$

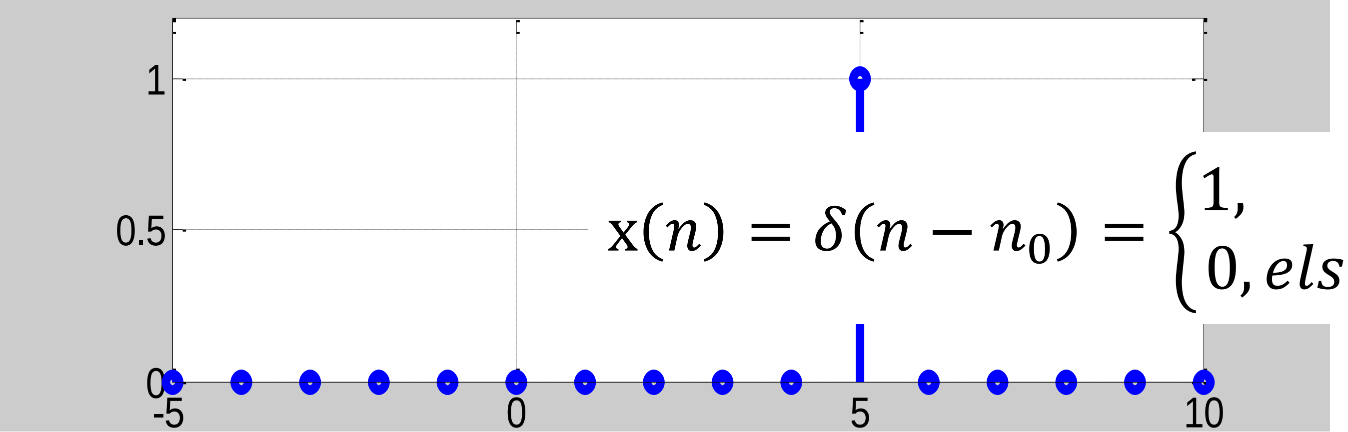
As the FT, the ZT transforms a convolution into a simple product.

Z transform

Examples



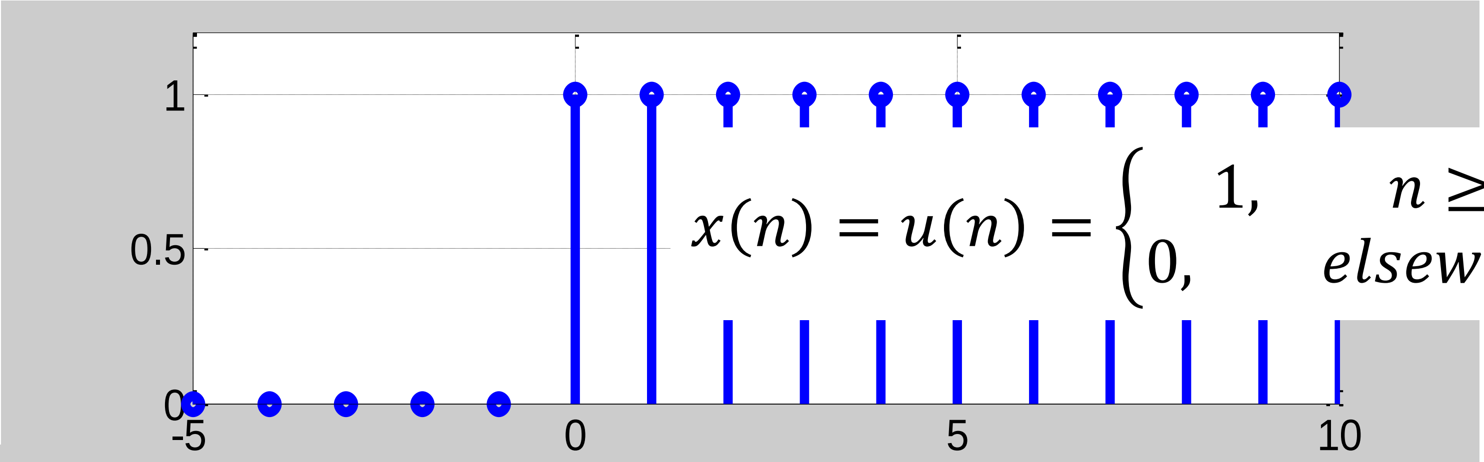
$$x(n) = \delta(n) = \begin{cases} 1, & n = 0 \\ 0, & elsewhere \end{cases} \rightarrow X(z) = 1$$



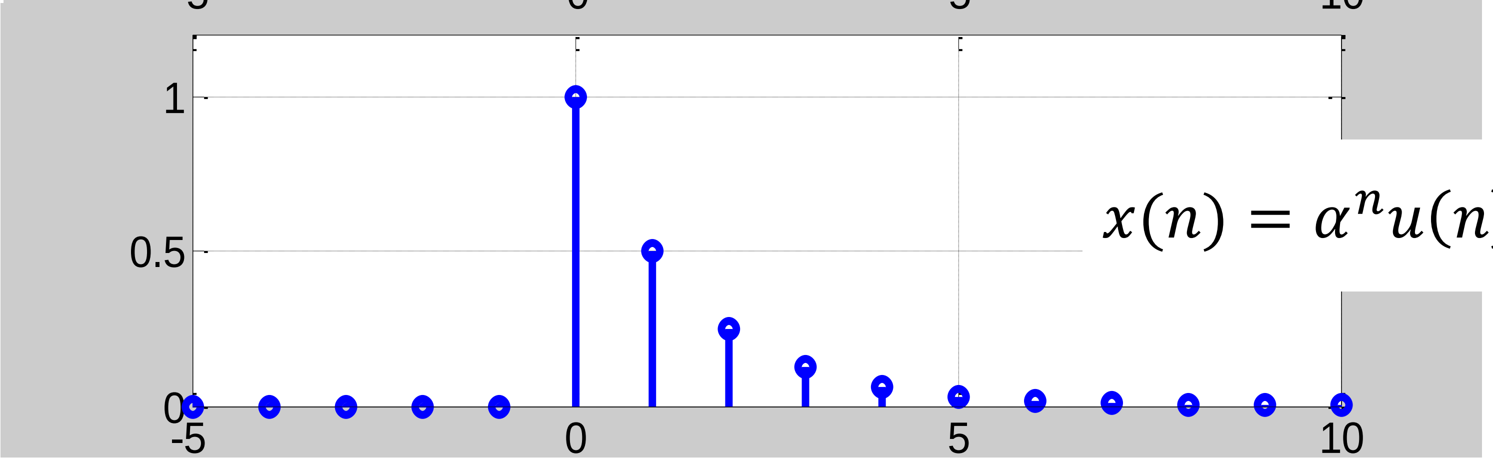
$$x(n) = \delta(n - n_0) = \begin{cases} 1, & n = n_0 \\ 0, & elsewhere \end{cases} \rightarrow X(z) = z^{-n_0}$$

Z transform

Examples



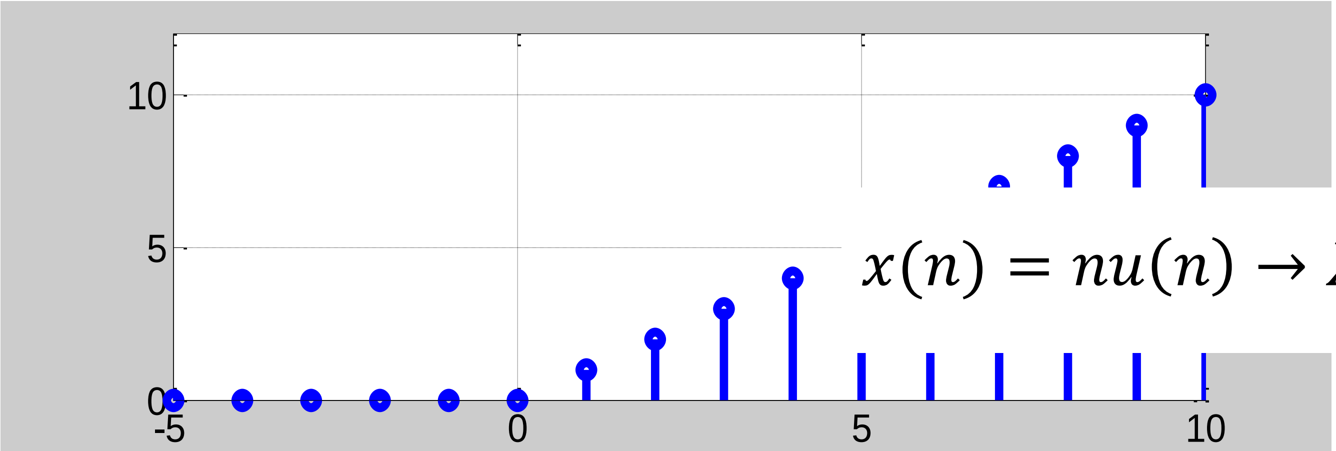
$$x(n) = u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & elsewhere \end{cases} \rightarrow X(z) = \frac{z}{z-1}$$



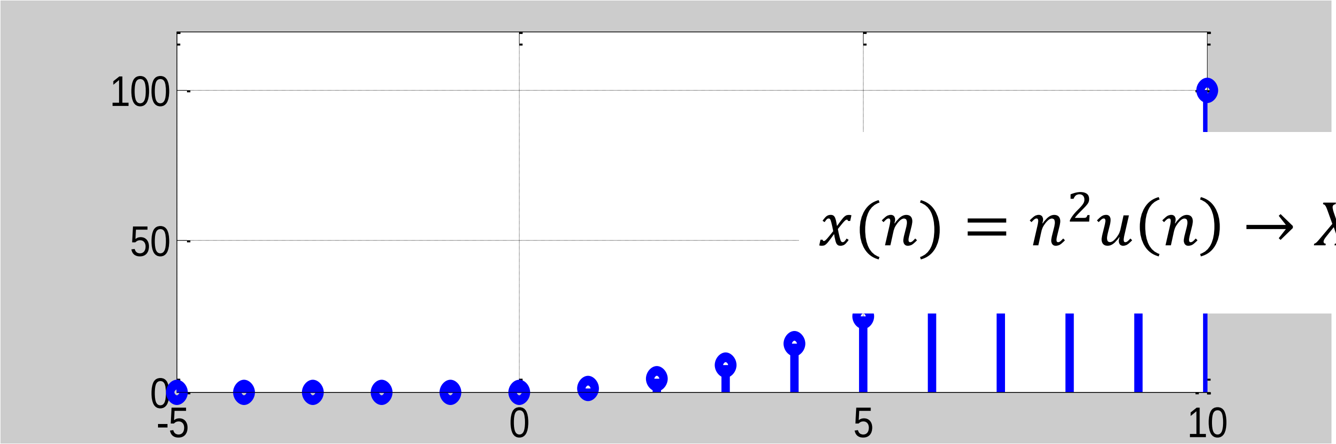
$$x(n) = \alpha^n u(n) \rightarrow X(z) = \frac{z}{z-\alpha}$$

Z transform

Examples



$$x(n) = nu(n) \rightarrow X(z) = \frac{z}{(z-1)^2}$$



$$x(n) = n^2u(n) \rightarrow X(z) = \frac{z(z+1)}{(z-1)^3}$$

Link between FT and ZT

A sequence $\{x_n\}$ represents the samples of a signal $x(t)$ taken each T_e.

$$X(z)_{|z=e^{j2\pi fT_e}} = \sum_{n=-\infty}^{+\infty} x_n e^{-j2\pi kfT_e} = X(f)$$

For $z = e^{j2\pi fT_e}$, the z-transform of the discrete sequence $\{x_n\}$ coincides with its Fourier transform.

The inverse Z transform

Definition

Fond the sequence $x(n)$ from its ZT : $\mathrm{X(z)}$.

<u>Compute the inverse ZT by integration</u>

$$x(n) = ZT^{-1}\{X(z)\} = \frac{1}{2\pi j}\oint_{\gamma} z^{n-1}X(z)dz$$

<u>Decomposition into partial fractions</u>

- Decomposition into simple elements,
- Reverse function search based on ZT tables.

The inverse Z transform

Example

Find the signal $x(n)$ from its Z transform.

$$X(z) = \frac{z^2}{(z-1)(z-\alpha)}$$

We decompose into simple elements.

$$X(z) = \frac{z^2}{(z-1)(z-\alpha)} = \frac{z}{1-\alpha}\left(\frac{z}{z-1} - \frac{z}{z-\alpha}\right)$$

The signal $x(n)$ is given by the inverse Z transform of $\mathrm{X}(z)$.

$$x(n) = ZT^{-1}\left\{\frac{z}{1-\alpha}\left(\frac{z}{z-1} - \frac{z}{z-\alpha}\right)\right\}$$

The inverse Z transform

Example

The inverse Z transform is linear

$$x(n) = ZT^{-1}\left\{\frac{z}{1-\alpha}\left(\frac{z}{z-1} - \frac{z}{z-\alpha}\right)\right\}$$

$$= \frac{1}{1-\alpha}\left[ZT^{-1}\left\{z\frac{z}{z-1}\right\} - ZT^{-1}\left\{z\frac{z}{z-\alpha})\right\}\right]$$

Multiplying by z correspond to one sample time advance

$$x(n) = \frac{1}{1-\alpha}\delta(n+1) * \left[ZT^{-1}\left\{\frac{z}{z-1}\right\} - ZT^{-1}\left\{\frac{z}{z-\alpha})\right\}\right]$$

$$= \frac{1}{1-\alpha}\delta(n+1) * [u(n) - \alpha^n u(n)]$$

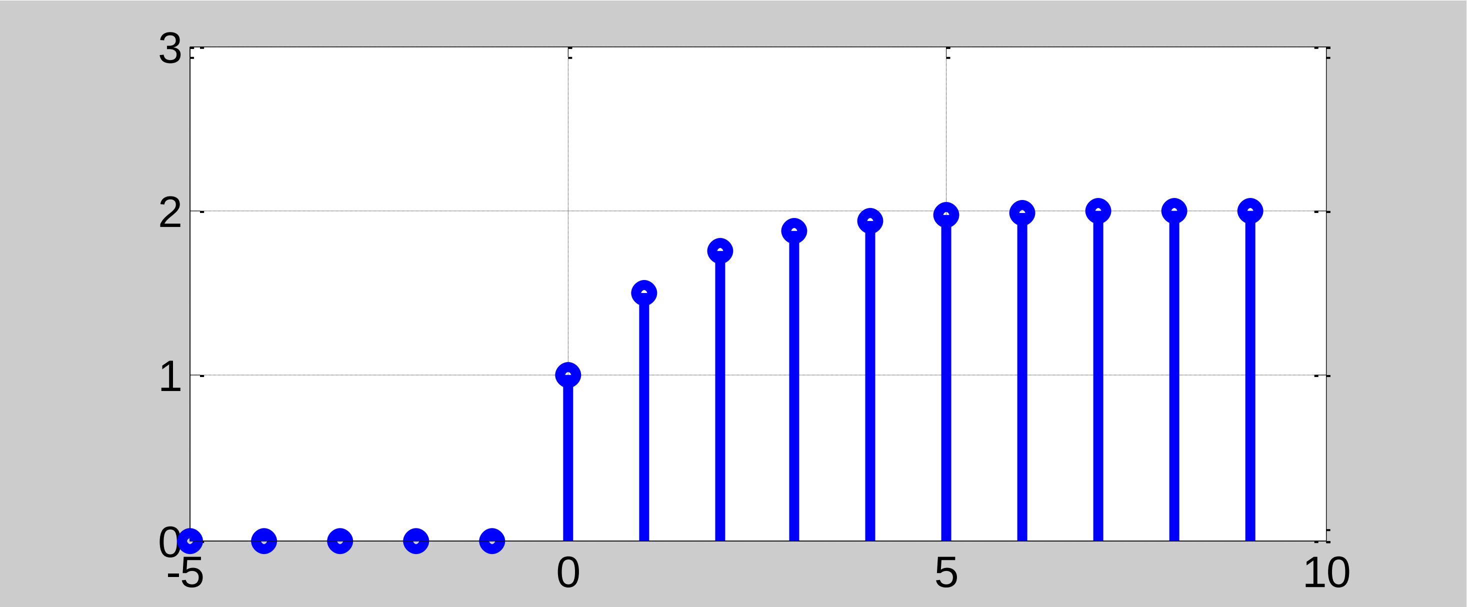
$$= \frac{1}{1-\alpha}\delta(n+1) * [(1-\alpha^n)u(n)]$$

The inverse Z transform

Example

The signal $x(n)$ is given by :

$$x(n) == \frac{1}{1-\alpha}(1-\alpha^{n+1})u(n+1)$$



Transfer function

Definition

$$x(n) \longrightarrow \boxed{\; h(n) \;} \longrightarrow y(n) = x(n) * h(n)$$

$$x(n) \xrightarrow[TZ]{} X(z) \qquad\qquad y(n) \xrightarrow[TZ]{} Y(z)$$

$$y(n) = x(n) * h(n) \xrightarrow[\mathrm{TZ}]{} Y(z) = X(z)H(z)$$

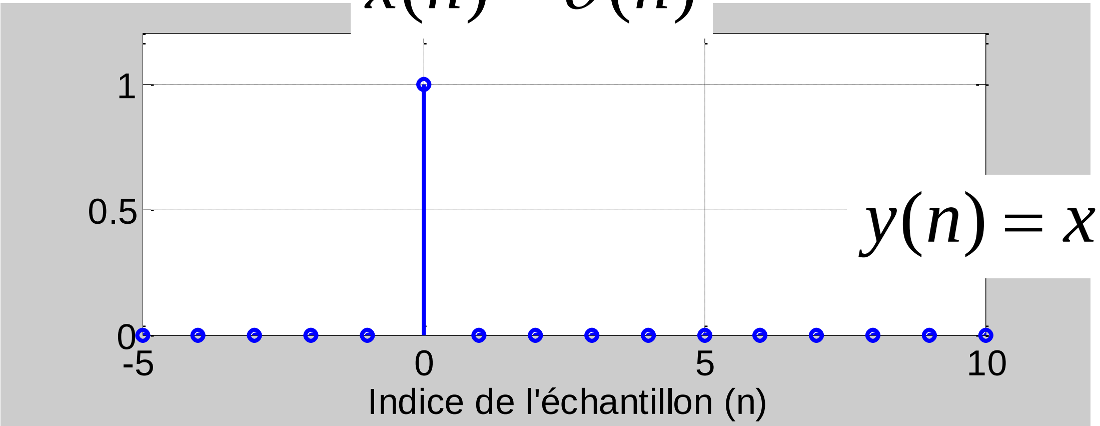
The system transfer function is given by :

$$H(z) = \frac{Y(z)}{X(z)}$$

Transfer function

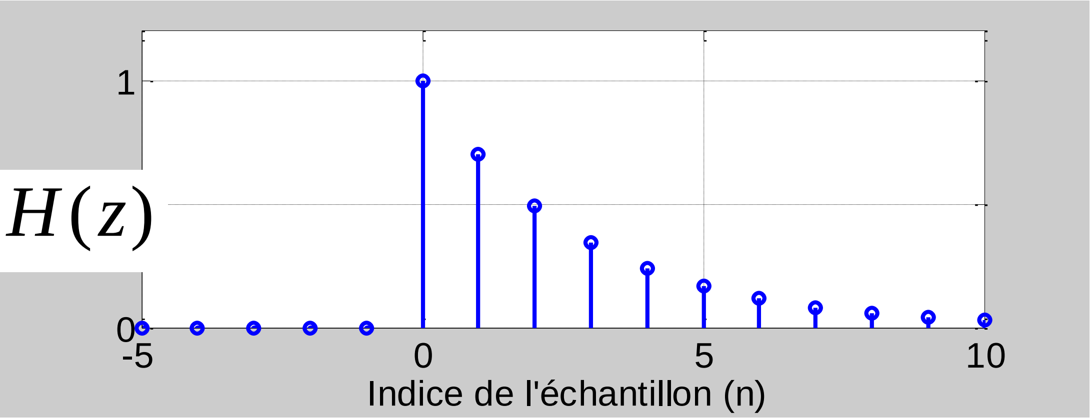
Link with the impulse response

$$x(n) = \delta(n)$$



$$y(n) = x(n) * h(n) \xrightarrow[TZ]{} Y(z) = X(z)H(z)$$

$$y(n) = h(n) \xrightarrow[TZ]{} Y(z) = H(z)$$



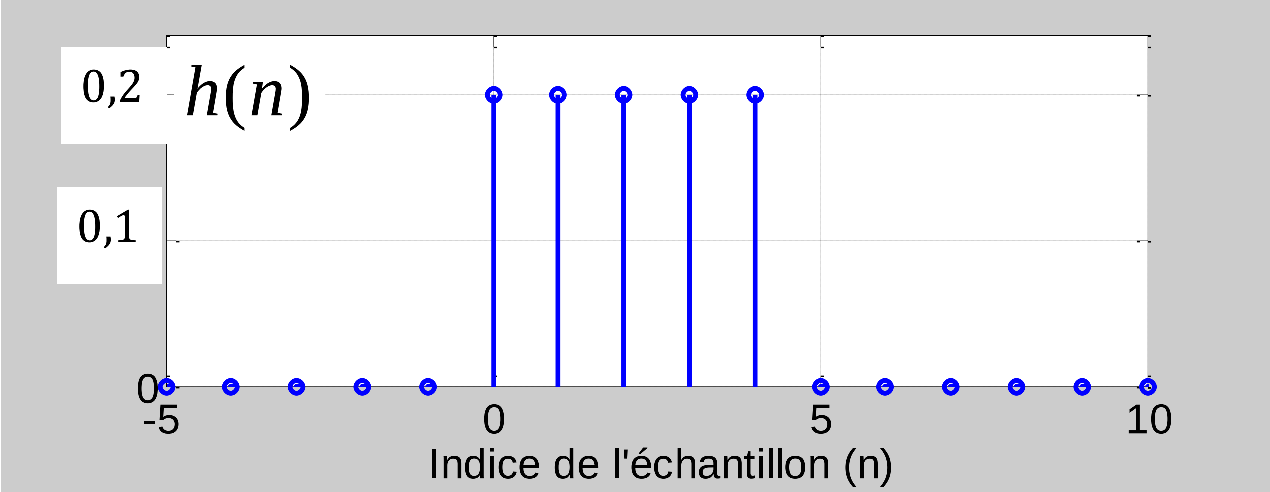
Transfer function

Example 1

Moving average over 5 samples

$$h(n)=\frac{1}{5}(\delta(n)+\delta(n-1)+\delta(n-2)+\delta(n-3)+\delta(n-4))$$

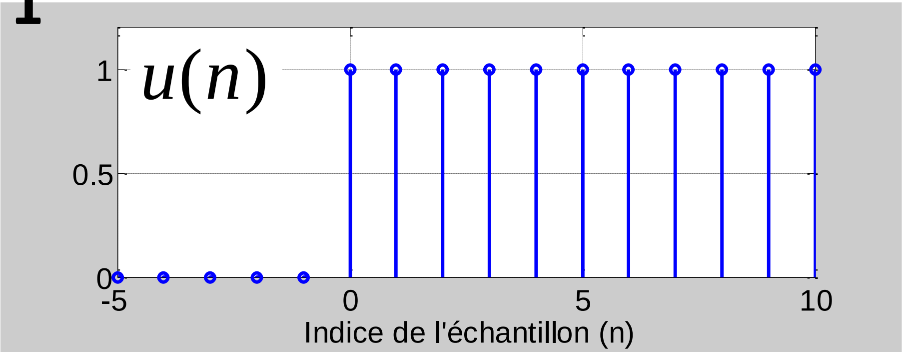
$$H(z)=\frac{1}{5}(1+z^{-1}+z^{-2}+z^{-3}+z^{-4})=\frac{z^4+z^3+z^2+z+1}{5z^4}$$



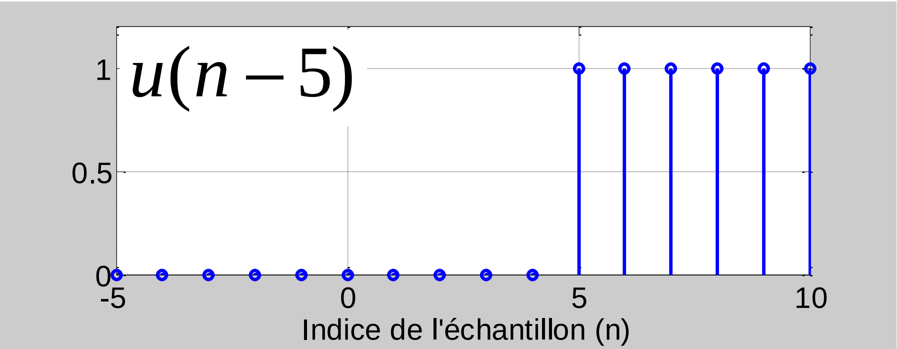
Transfer function

Example 1

$$h(n)=\frac{1}{5}($$



$$-$$



$$)$$

The impulse response can be seen as a weighted sum of two delayed step functions.

$$h(n)=\frac{1}{5}(u(n)-u(n-5))$$

$$H(z)=\frac{1}{5}(U(z)-z^{-5}U(z))=\frac{1}{5}\frac{1-z^{-5}}{1-z^{-1}}$$

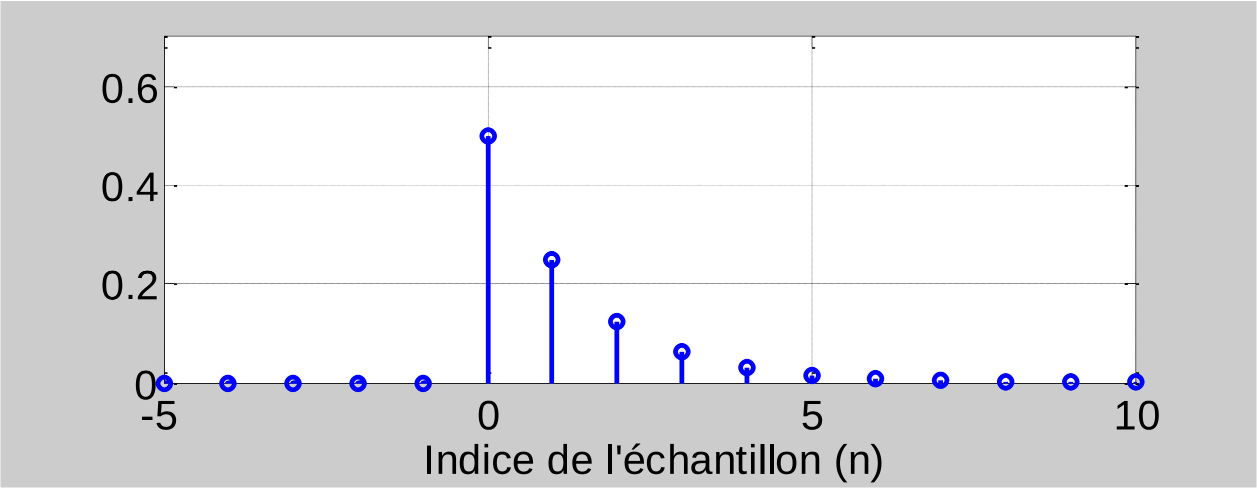
$$=\frac{1}{5}\frac{(1-z^{-1})(1+z^{-1}+z^{-2}+z^{-3}+z^{-4})}{1-z^{-1}}$$

Transfer function

Example 2

$$y(n) = 0.5x(n) + 0.5y(n-1) \quad Y(z) = 0.5X(z) + 0.5z^{-1}Y(z)$$

$$H(z) = \frac{0.5}{1-0.5z^{-1}} = \frac{0.5z}{z-0.5} = 0.5\frac{(\frac{z}{0.5})}{(\frac{z}{0.5})-1} = 0.5U(\frac{z}{0.5})$$



Transfer function

Link to the difference equation

$$\sum_{k=0}^{K-1} a_k y(n-k) = \sum_{m=0}^{M-1} b_m x(n-m)$$

$$ZT\left\{\sum_{k=0}^{K-1} a_k y(n-k)\right\} = ZT\left\{\sum_{m=0}^{M-1} b_m x(n-m)\right\}$$

$$\sum_{k=0}^{K-1} a_k z^{-k} Y(z) = \sum_{m=0}^{M-1} b_m z^{-m} X(z)$$

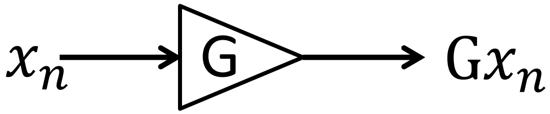
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{\sum_{k=0}^{K-1} a_k z^{-k}} = \frac{N(z)}{D(z)}$$

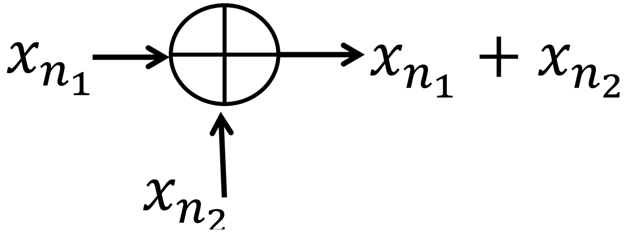
Transfer function

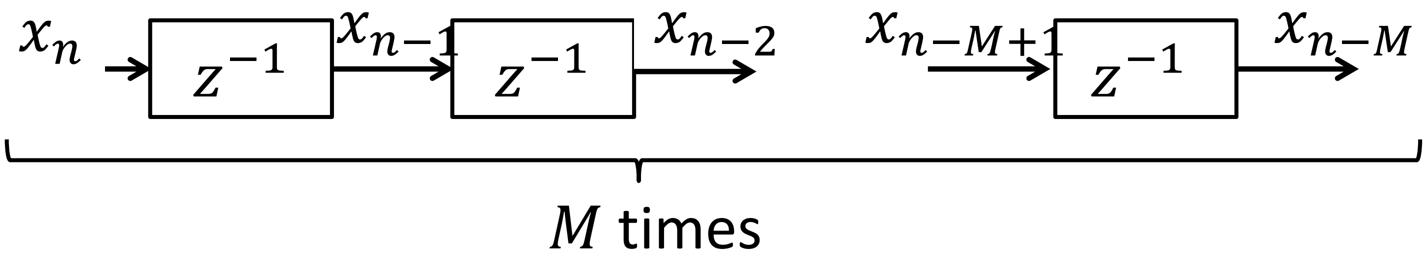
Implementation scheme

$$\sum_{k=0}^{K-1} a_k y(n-k) = \sum_{m=0}^{M-1} b_m x(n-m)$$

The linear equation differences can be represented by a diagram in which gain, delay and sum are represented by functional blocks



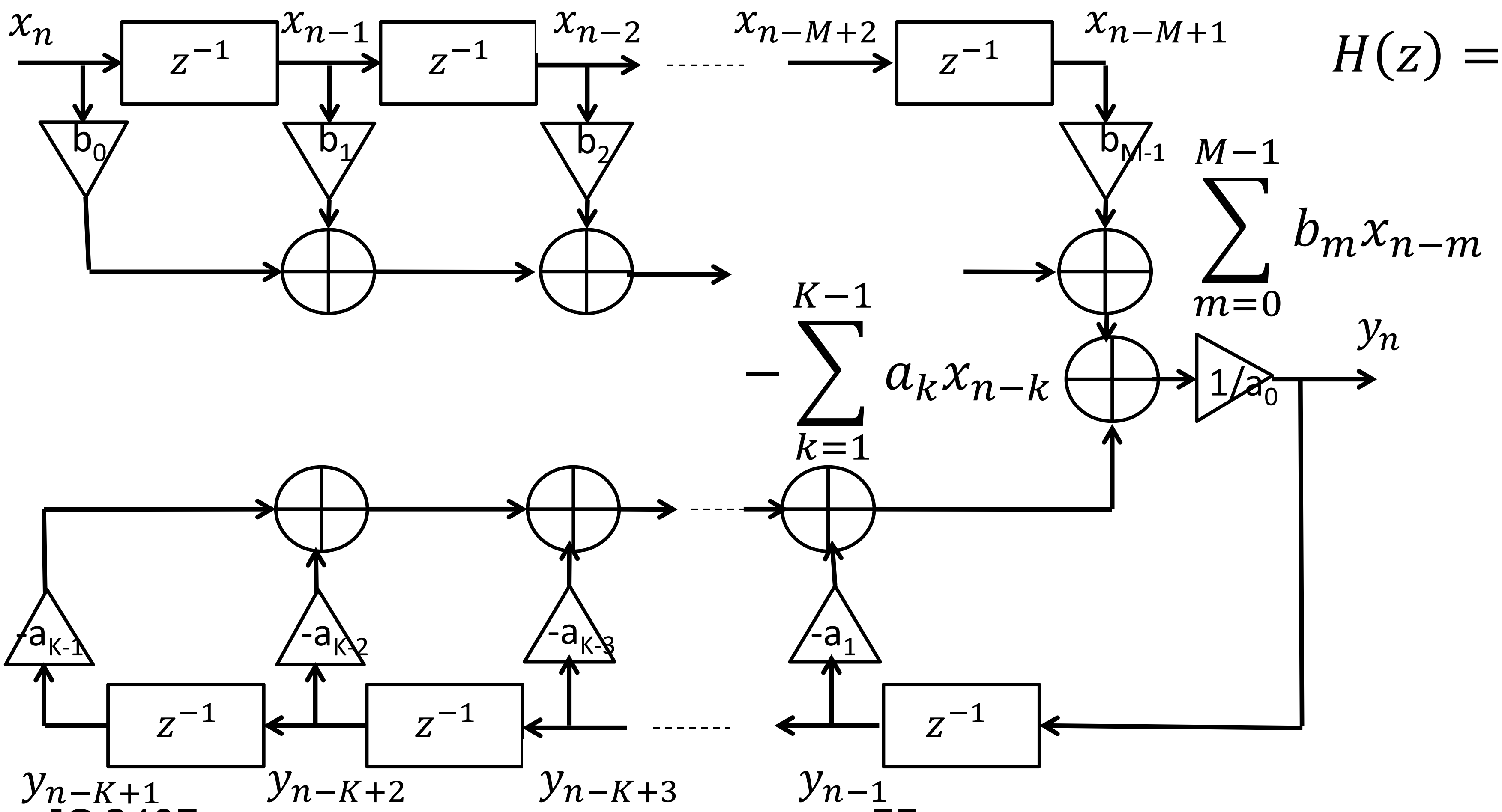




Transfer function

Implementation scheme

$$H(z) = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{\sum_{k=0}^{K-1} a_k z^{-k}}$$

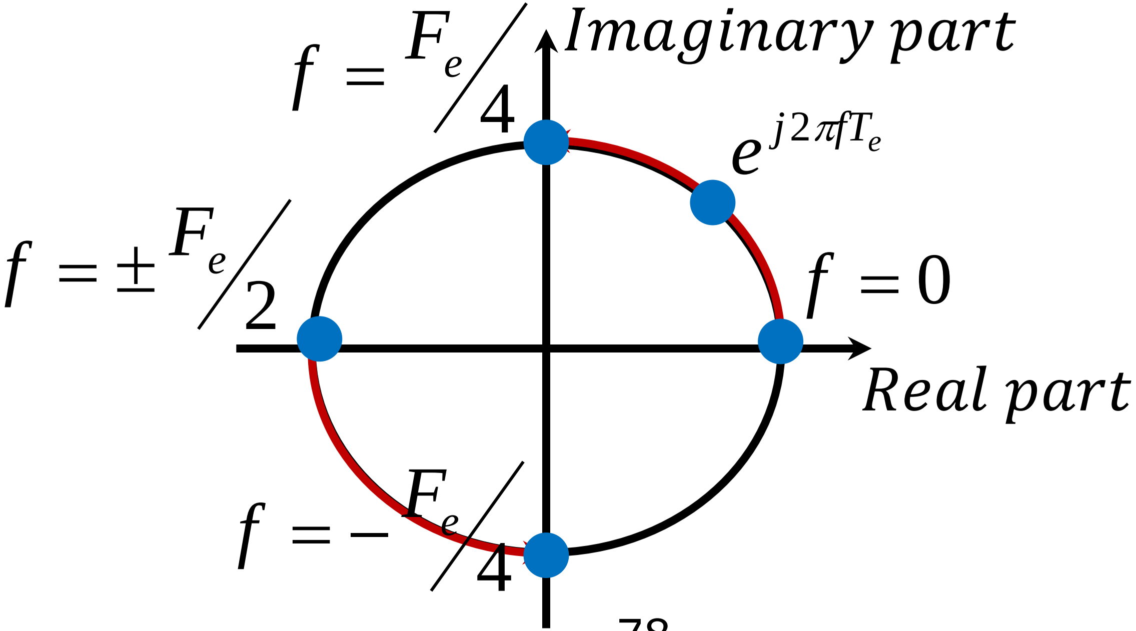


Transfer function

Link with the frequency response

$$H(z) = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{\sum_{k=0}^{K-1} a_k z^{-k}} \quad \underset{z=e^{j2\pi f T_e}}{\Longleftrightarrow} \quad H(f) = \frac{\sum_{m=0}^{M-1} b_m e^{-j2\pi m f T_e}}{\sum_{k=0}^{K-1} a_k e^{-j2\pi k f T_e}}$$

$H(f)$ is the particular value of $H(z)$, when z is located on the unit circle



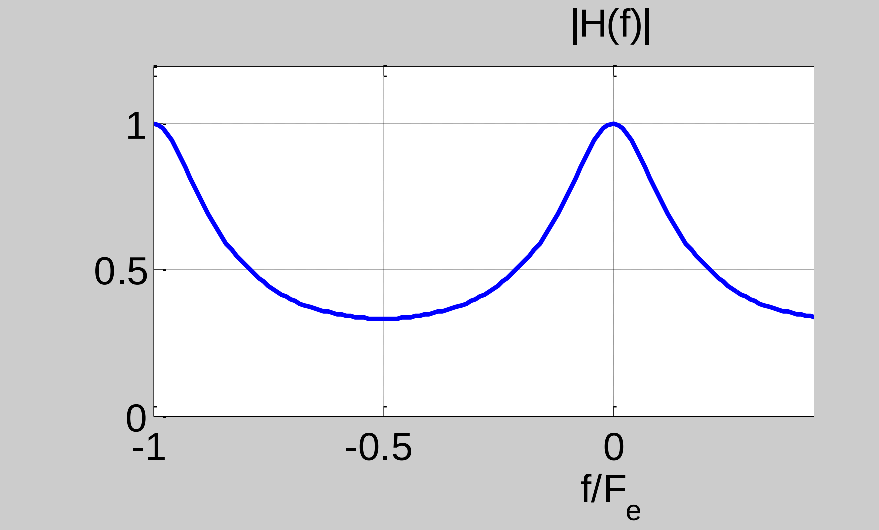
Transfer function

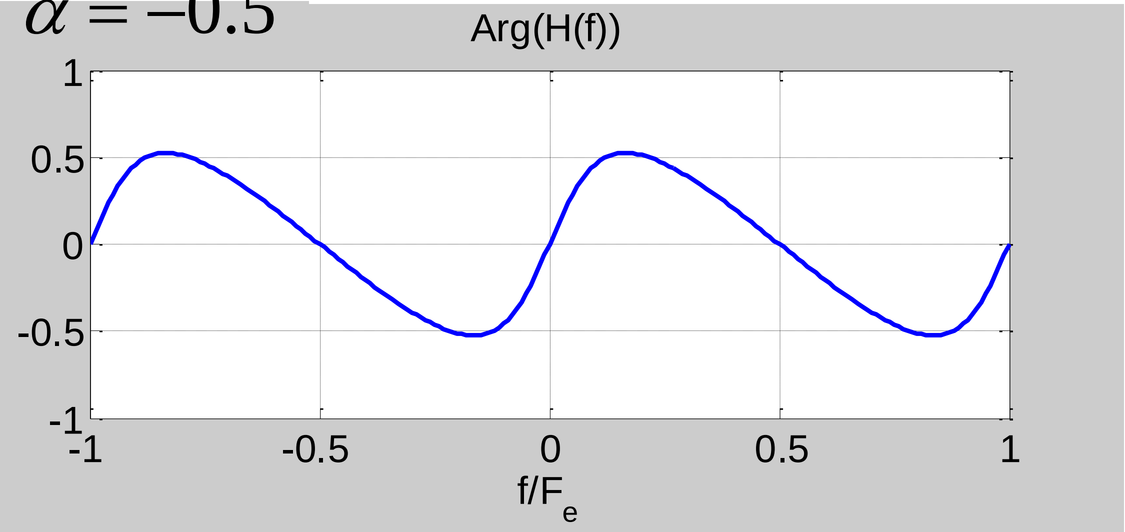
Link with the frequency response, example

$$H(z)=\frac{1+\alpha}{1+\alpha z^{-1}}\xrightarrow[z=e^{j2\pi f/f_e}]{}H(f)=\frac{1+\alpha}{1+\alpha e^{-j2\pi f/F_e}}=\frac{1+\alpha}{1+\alpha\cos(2\pi f/F_e)+j\alpha\sin(2\pi f/F_e)}$$

$$|H(f)|=\frac{|1+\alpha|}{\sqrt{1+2\alpha\cos(2\pi f/F_e)+\alpha^2}}\qquad Arg(H(f))=-\arctan\left(\frac{\alpha\sin(2\pi f/F_e)}{1+\alpha\cos(2\pi f/F_e)}\right)$$

$$\alpha=-0.5$$





Factorization of the transfer function

The transfer function, $H(z)$ can be written

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{\sum_{k=0}^{K-1} a_k z^{-k}} = z^{K-M}\,\frac{b_0 z^{M-1} + b_1 z^{M-2} + \cdots + b_{M-1}}{a_0 z^{K-1} + a_1 z^{K-2} + \cdots + a_{K-1}}$$

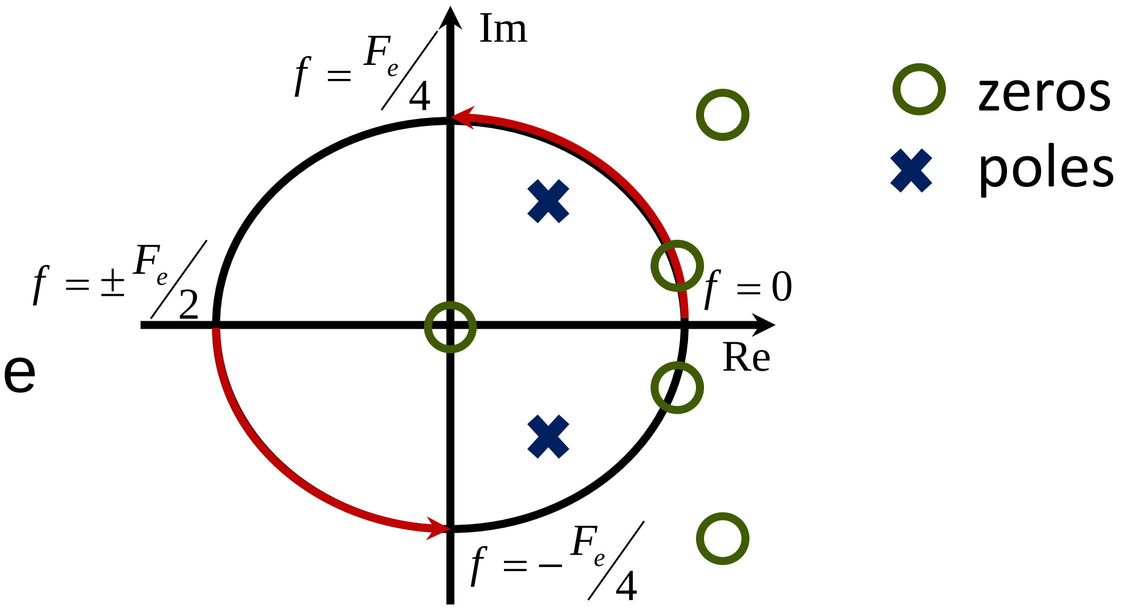
$$= \frac{b_0}{a_0} z^{K-M}\,\frac{(z - z_1)(z - z_2)\ldots(z - z_{M-1})}{(z - p_1)(z - p_2)\ldots(z - p_{K-1})}$$

The zeros of $H(z)$ are solutions of the numerator $\mathrm{N}(z)(\{z_1, z_2, \ldots z_{M-1}\})$, and the poles are solutions of the denominator $\mathrm{D}(z)$ $(\{p_1, p_2, \ldots p_{K-1}\})$.

The zeros are the values of z annulating the transfer function, and the poles are the values which render it infinite.

Zeros and poles in the complex plane

Zeros and poles can be plotted in the complex plane



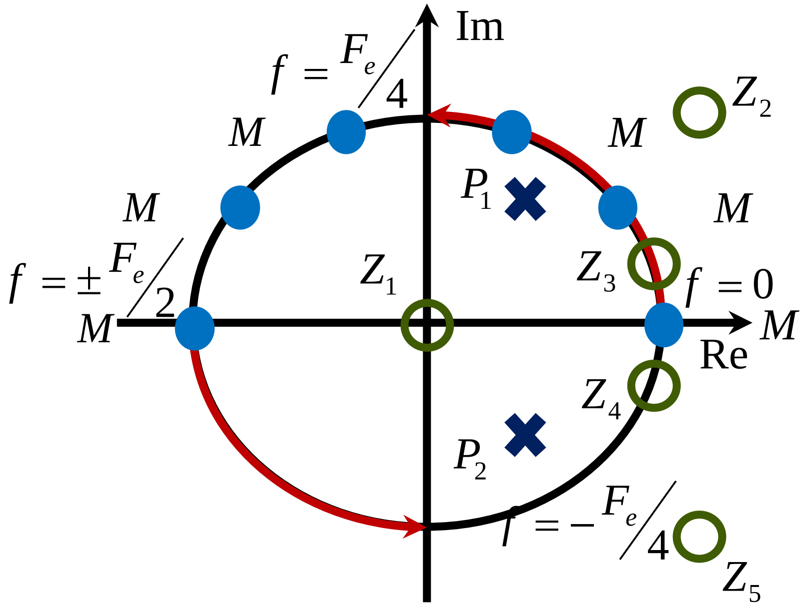
The diagram of zeros and poles helps in :

- understanding the frequency responses,
- studying the stability of the system.

Interpretation of the frequency reponses (1/3)

$$H(z) = \frac{b_0}{a_0} z^{K-M} \frac{\prod_{m=1}^{M-1}(z - z_m)}{\prod_{k=1}^{K-1}(z - p_k)} \xrightarrow[z=e^{j2\pi f T_e}]{} H(f)$$

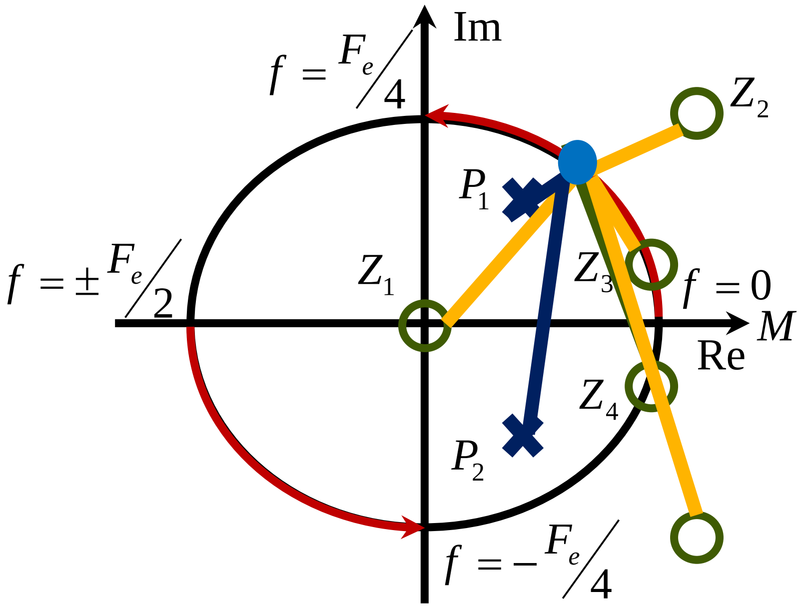
- The affixes of zeros and poles are Z_m and P_k.
- Let M, be the affixe of z for the curent value of f on the unit circle.



Interpretation of the frequency reponses (2/3)

- Magnitude (amplitude) of the frequency reponse

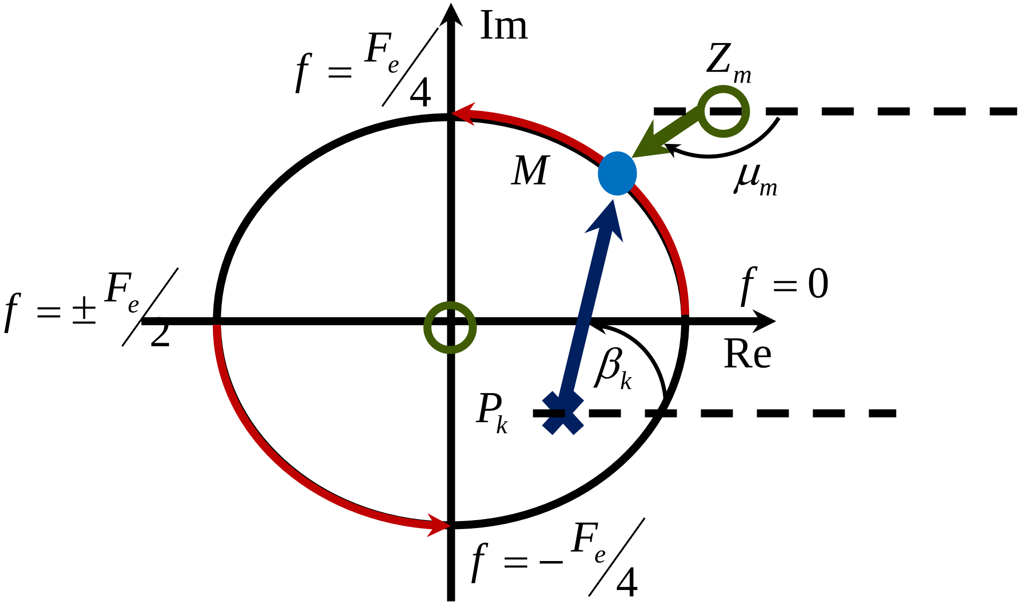
$$|H(f)| = \left|\frac{b_0}{a_0}\right| \frac{\prod_{m=1}^{M-1} MZ_m}{\prod_{k=1}^{K-1} MP_k} = \left|\frac{b_0}{a_0}\right| \frac{MZ_1 MZ_2 \dots MZ_{M-1}}{MP_1 MP_2 \dots MP_{K-1}}$$



Interpretation of the frequency reponses (3/3)

- Phase of the frequency reponse

$$Arg\big(H(f)\big) = 2\pi(K-M)fT_e + \sum_{m=1}^{M-1}\mu_m - \sum_{k=1}^{K-1}\beta_k$$



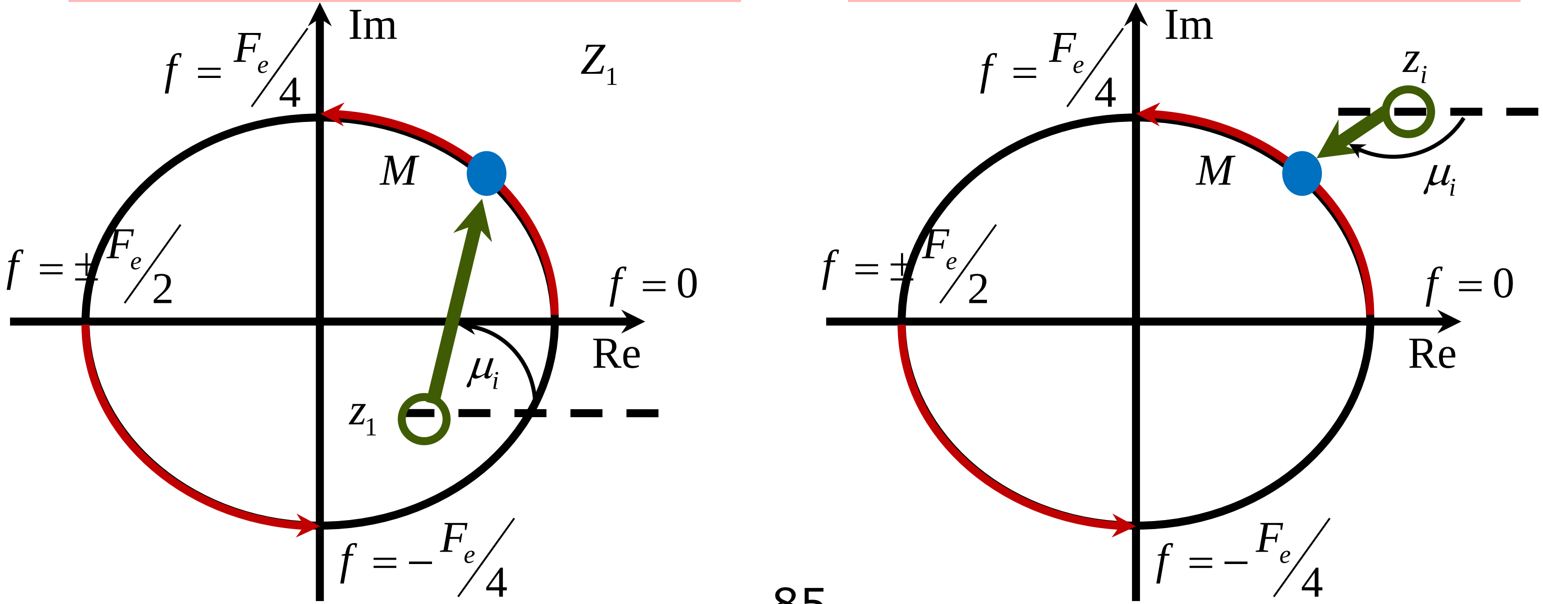
μ_m, β_k are the angles between $\overrightarrow{Z_m M}$, $\overrightarrow{P_k M}$ and the real axis.

Minimum phase / maximal phase system

- Non recursive system $(K = 1)$: $\quad Arg\big(H(f)\big) = 2\pi(1-M)fT_e + \sum_{m=1}^{M-1} \mu_m$

Zeros inside the unit circle : minimum phase

Zeros outside the unit circle : maximum phase

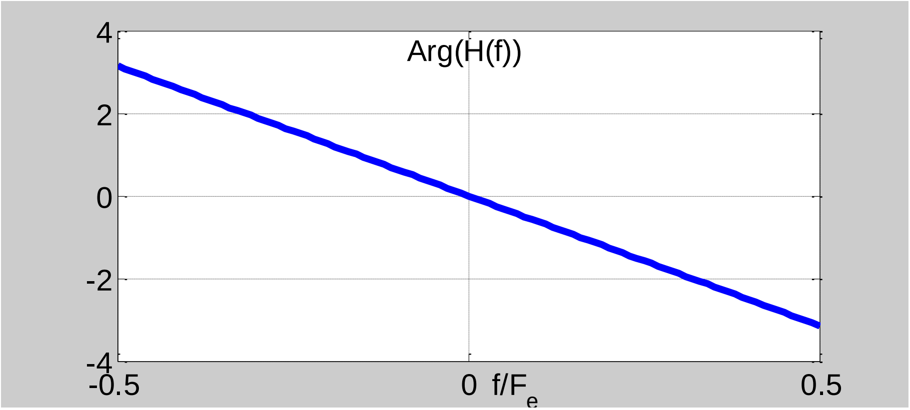


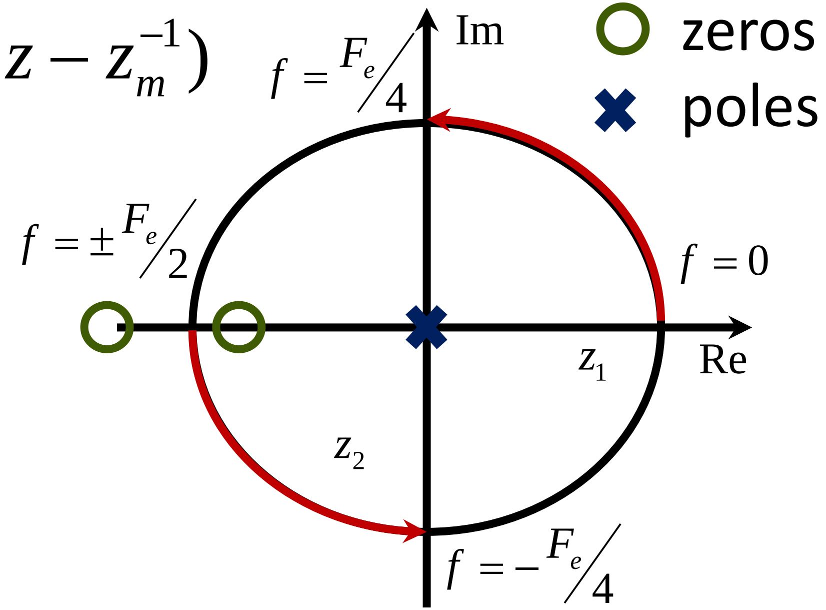
Linear phase system

$$H(z) = \sum_{m=0}^{M-1} b_m z^{-m} = z^{1-M}\left(b_0 z^{M-1} + b_1 z^{M-2} + \cdots + b_{M-1}\right)$$

$$= b_0 z^{1-M}\left((z-z_1)(z-z_2)\ldots(z-z_{M-1})\right)$$

$$H(z) = b_0 z^{1-M} \prod_{m=1}^{\frac{M-1}{2}} (z - z_m)(z - z_m^{-1})$$





Stability (1/3)

$$x(n) \longrightarrow \boxed{H(z)} \longrightarrow y(n) = x(n) * h(n)$$

$$y_n = \sum_{i=0}^{+\infty} h_i x_{n-i}$$

Stability condition

finite input → finite output

if $|x_n| < A \; \forall \, n, |y_n| < A \sum_{i=0}^{+\infty} |h_i|$

Necessary and sufficient condition

$$\sum_{i=0}^{+\infty} |h_i| < +\infty$$

Stability (2/3)

$$H(z) = \frac{b_0}{a_0} z^{K-M} \frac{(z - z_1)(z - z_2) \ldots (z - z_{M-1})}{(z - p_1)(z - p_2) \ldots (z - p_{K-1})}$$

$$= \frac{b_0}{a_0} z^{K-M} \frac{a_1}{(z-p_1)} + \frac{a_2}{(z-p_2)} + \cdots + \frac{a_{K-1}}{(z-p_{K-1})}$$

$$h(n) = TZ^{-1}\big(H(z)\big) = \left(a_1 {p_1}^n + \cdots + a_{K-1} {p_{K-1}}^n\right) u(n)$$

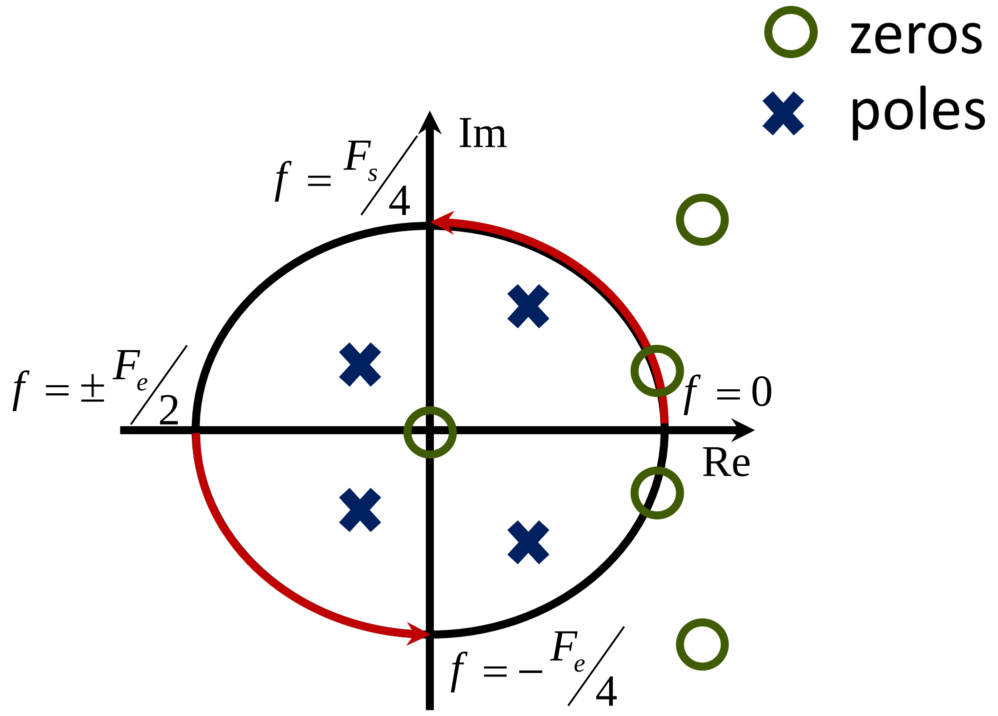
Stability condition

$$\sum_{i=0}^{+\infty} |h_i| < +\infty \rightarrow |p_1| < 1\, et\, |p_2| < 1\, et \ldots |p_N| < 1$$

All the poles are within the unit circle

Stability (3/3)

A LTI system is stable when all its poles are located inside the unit circle.



Outline

1. Data acquisition and analysis (2 lectures)

2. Digital data filtering (2 lectures)

3. Random signal processing (1 lecture)

Outline

2. Digital data filtering

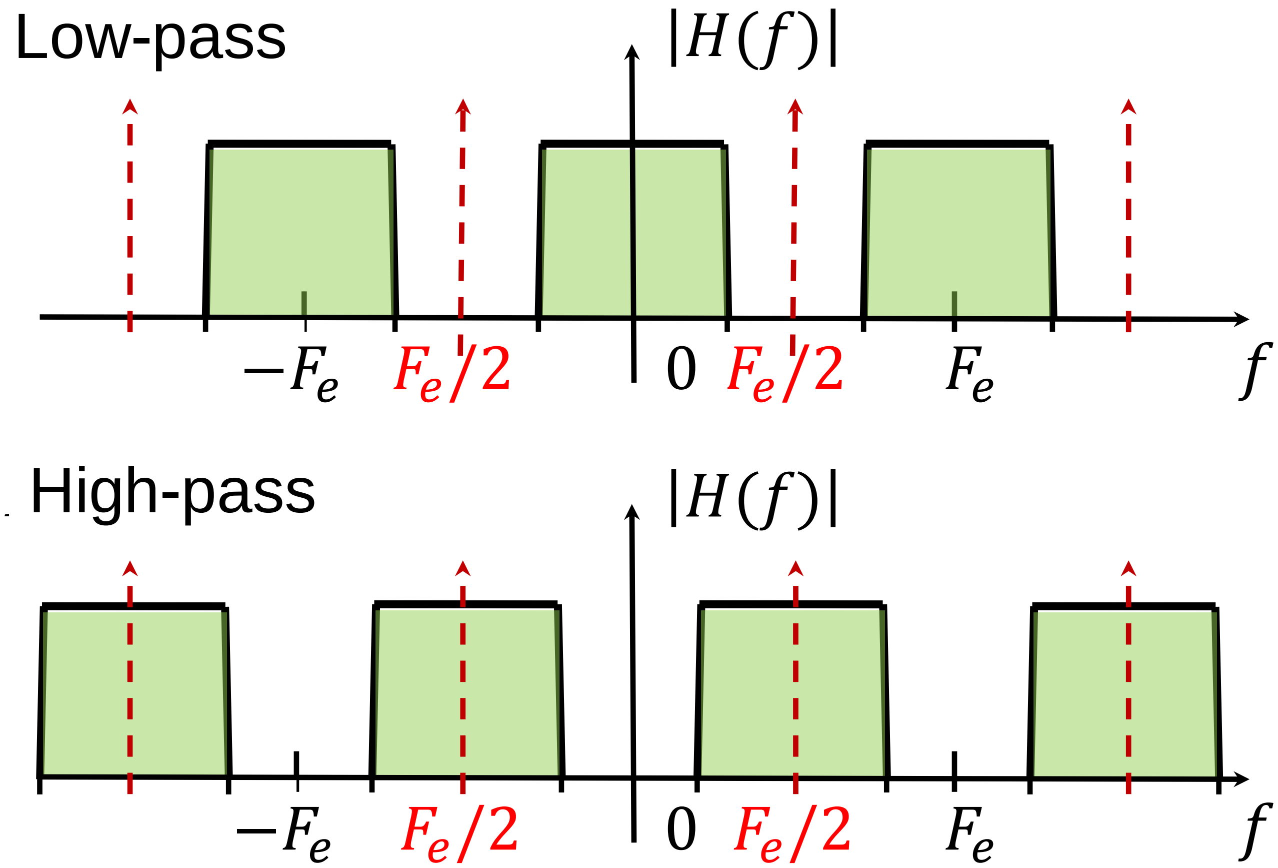
– Finite Impulse Response (FIR) filters
– Infinite Impulse Response (IIR) filters

Generalities

As in analog, a digital filter makes it possible to separate, different components of a signal:

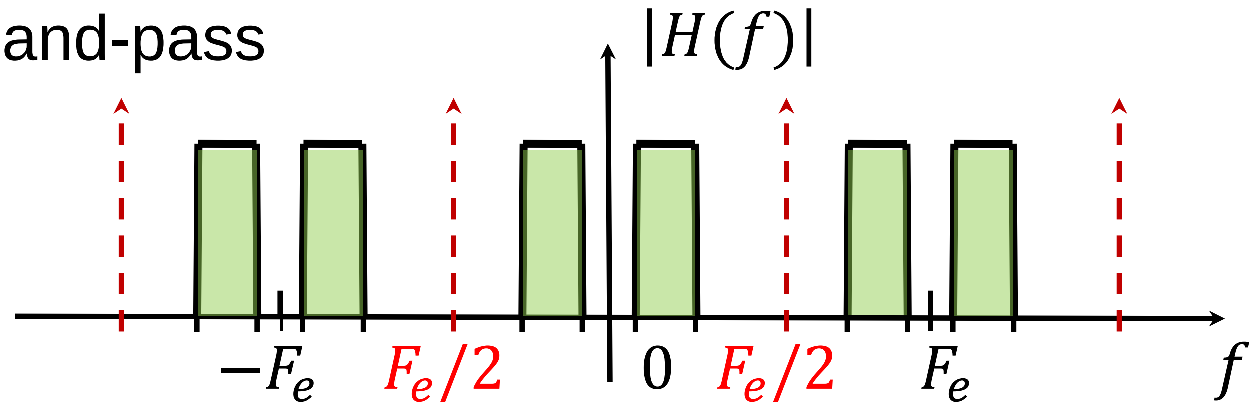
- Retrieve the part of the signal whose spectrum is located in a given frequency band: low-pass, band-pass, high-pass,
- Remove an unwanted signal from the wanted signal: noise, echo, …
- Correct locally the spectrum of a signal,
- Retrieve information transmitted via a propagation channel, which is time varying.
- …

Classification (1/3)

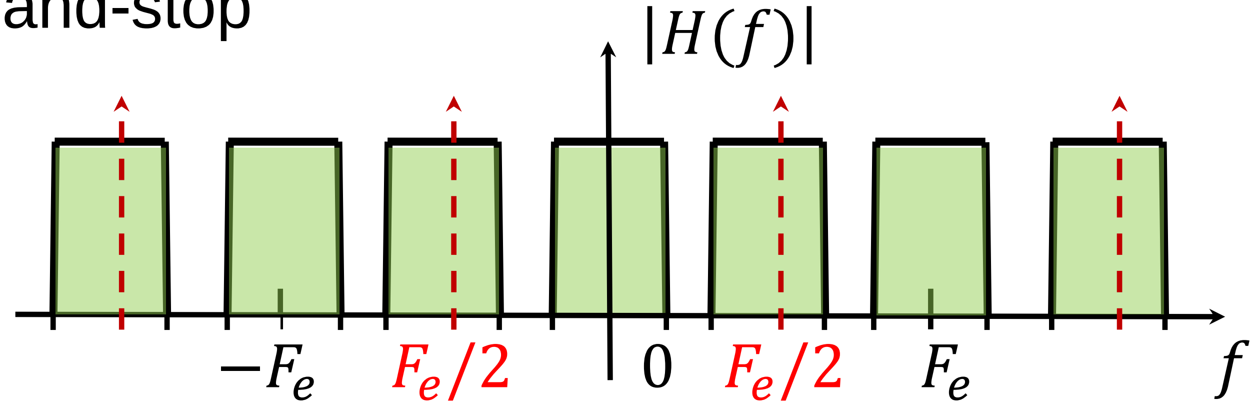


Classification (2/3)

Band-pass



Band-stop



Classification (3/3)

$$x(n) \longrightarrow \boxed{H} \longrightarrow y(n)$$

The samples at the input and at the output of a digital filter are given by the difference equation :

$$y_n = \sum_{m=0}^{M-1} \frac{b_m}{a_0} x_{n-m} - \sum_{k=1}^{K-1} \frac{a_k}{a_0} y_{n-k}$$

Another classification of digital filters derives from the values taken by a_k , $\mathrm{k} \in [1, K-1]$.

If $\begin{cases} a_k = 0, \forall\, \mathrm{k} \in [1, K-1] & \rightarrow FIR\ filter \\ \exists \mathrm{k} \in [1, K-1]\ such\ as\ a_k \neq 0 & \rightarrow IIR\ filter \end{cases}$

Outline

2. Digital data filtering

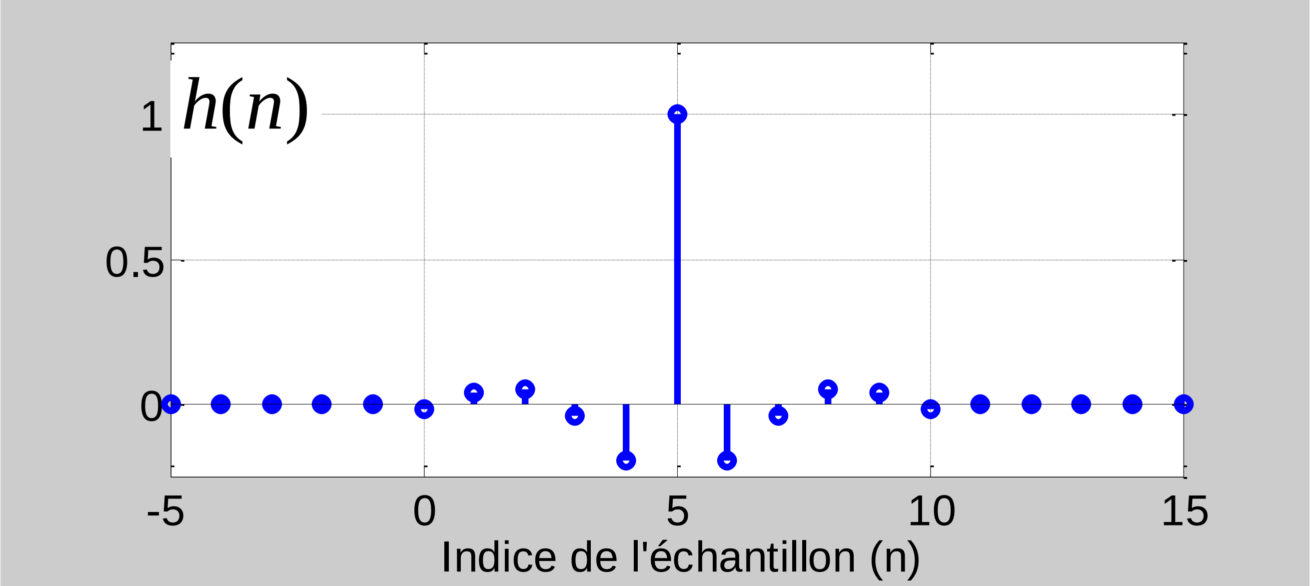
- **Finite Impulse Response (FIR) filters**
- **Infinite Impulse Response (IIR) filters**

Impulse response

$$y_n = \sum_{m=0}^{M-1} \frac{b_m}{a_0} x_{n-m} = \sum_{m=0}^{M-1} h_m x_{n-m}$$

We identify the coefficients $\frac{b_m}{a_0}$, to the coefficients h_m, of the impulse response.

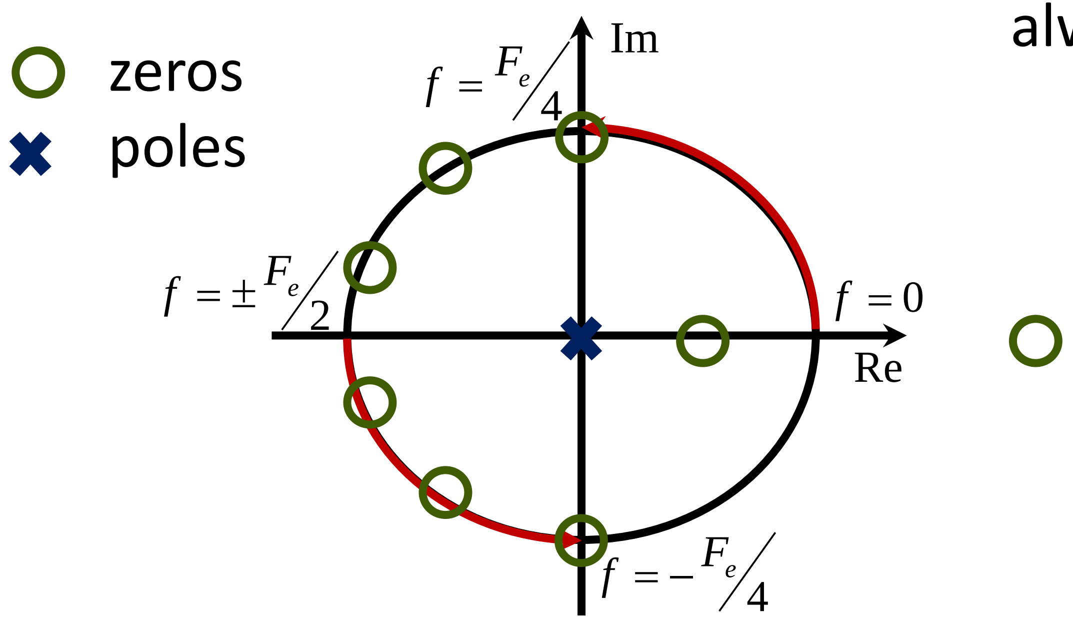
n	0	1	2	3	4	5	6	7	8	9	10
h_n	-0.017	0.039	0.053	-0,043	-0,197	1	-0,197	-0,043	0.053	0.039	-0.017



Transfer function

$$H(z)=\sum_{m=0}^{M-1} h_m z^{-m} = \frac{\sum_{m=0}^{M-1} h_m z^{M-m-1}}{z^{M-1}} = \frac{N(z)}{z^{M-1}}$$

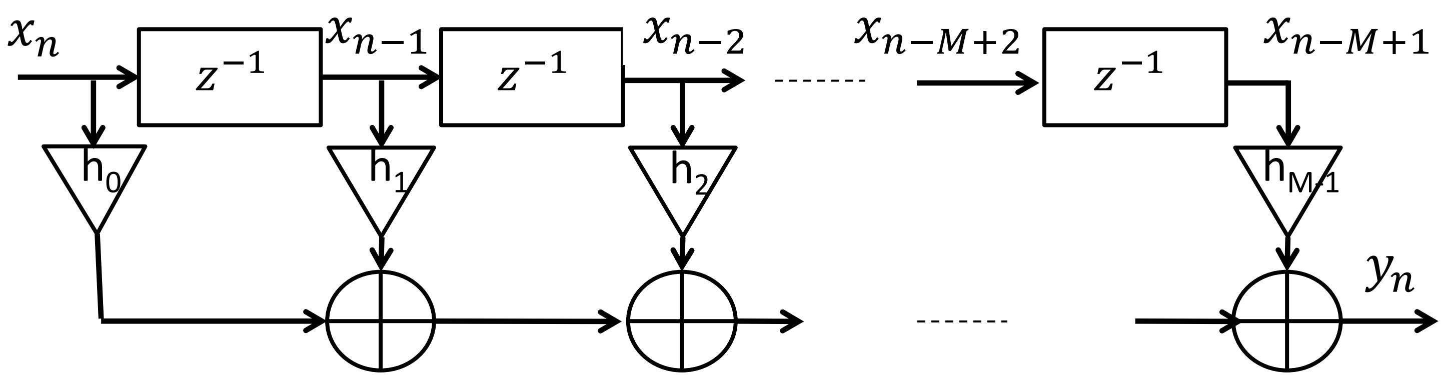
The poles of the FIR filter are located at the origin of the unit circle. This filter is so always stable.



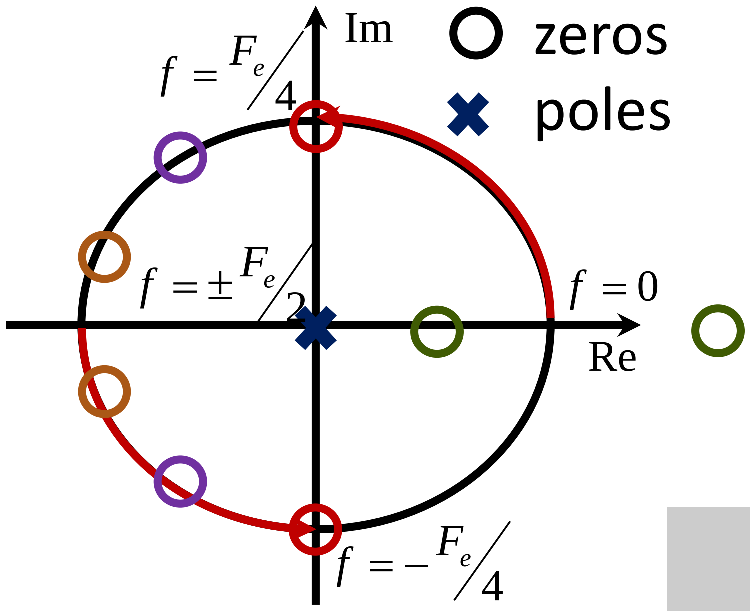
Implementation scheme

$$y(n) = \sum_{m=0}^{M-1} h_m x(n-m)$$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{m=0}^{M-1} h_m z^{-m} \rightarrow Y(z) = \sum_{m=0}^{M-1} h_m z^{-m} X(z)$$

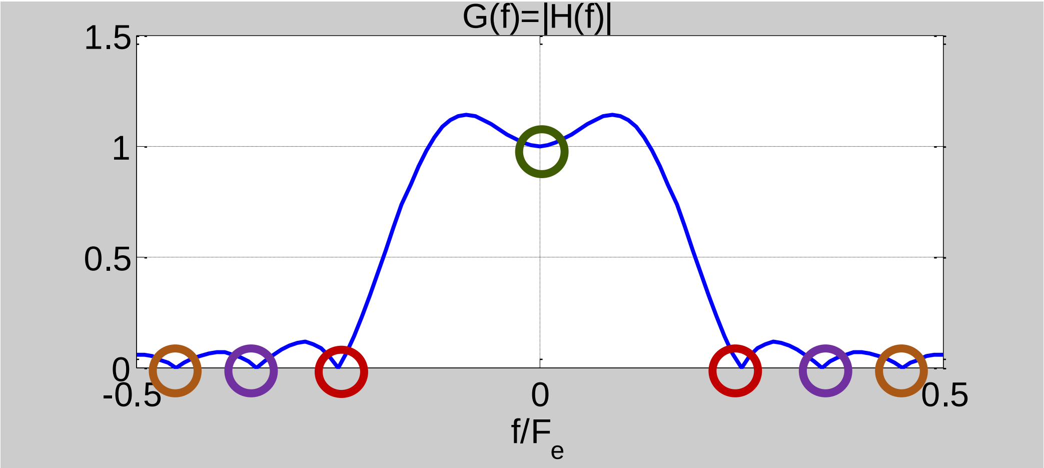


Frequency responses (1/2)

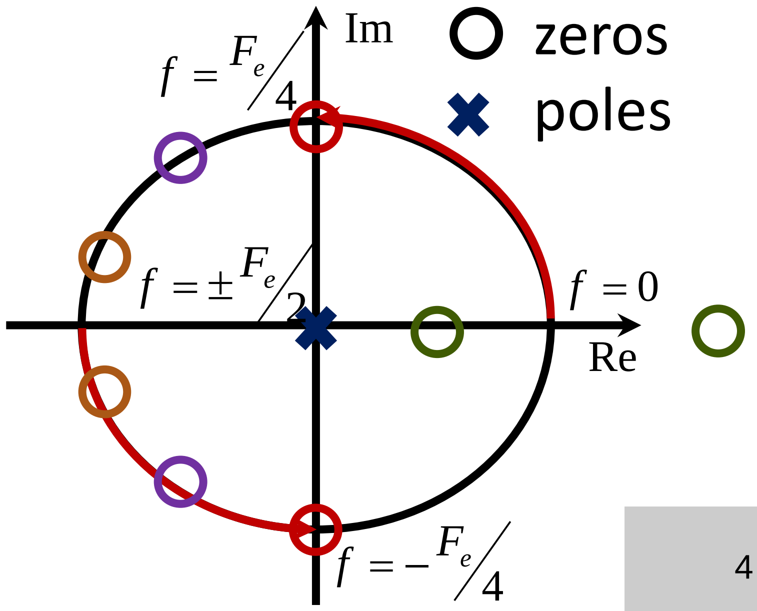


$$H(f) = \left. \frac{\sum_{m=0}^{M-1} h_m z^{M-m-1}}{z^{M-1}} \right|_{z=e^{j2\pi f T_e}}$$

$$= \sum_{m=0}^{M-1} h_m e^{-j2\pi m f / F_e} = G(f) e^{j\phi(f)}$$



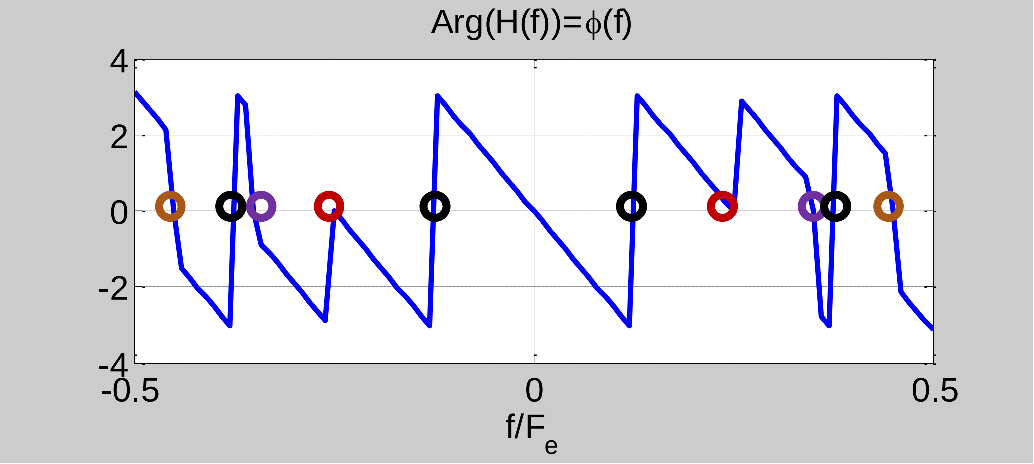
Frequency responses (2/2)



$$H(f) = \frac{\sum_{m=0}^{M-1} h_m z^{M-m-1}}{z^{M-1}}\Bigg|_{z=e^{j2\pi f T_e}} = \sum_{m=0}^{M-1} h_m e^{-j2\pi m f/F_e}$$

$$= G(f) e^{j\phi(f)}$$

Fictitious discontinuity

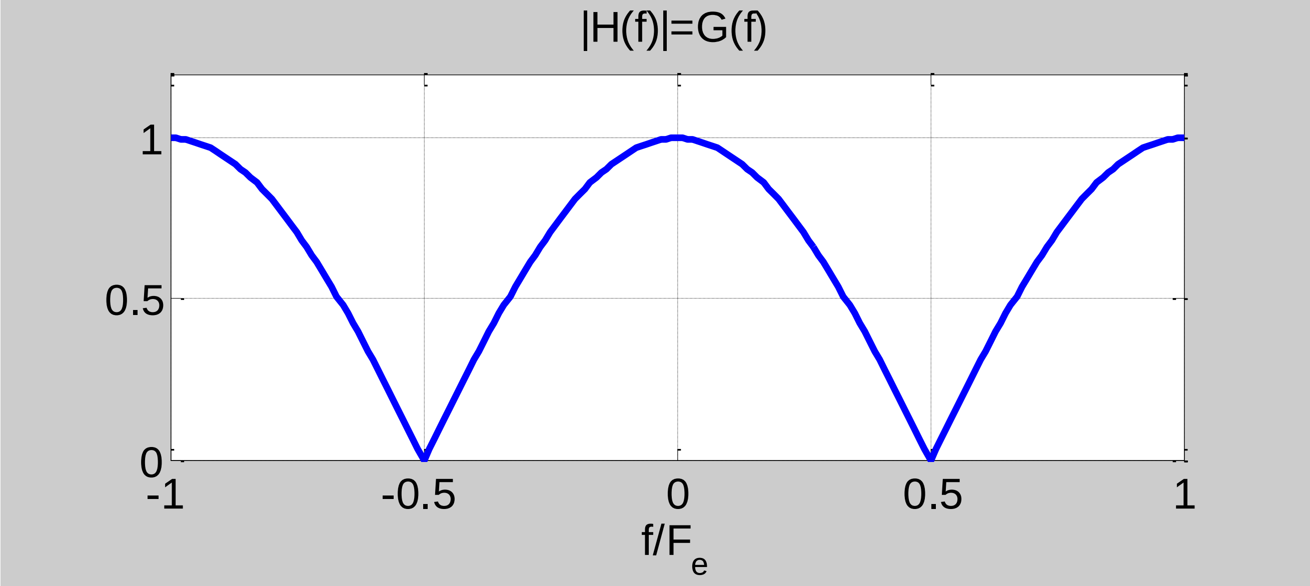


Example (1/2)

Averaging over 2 samples : $y_n = \frac{1}{2}(x_n + x_{n-1})$

$$H(z) = \frac{1}{2}(1+z^{-1}) \xrightarrow[z=e^{j2\pi f/F_e}]{} H(f) = \frac{1}{2}(1+e^{-j2\pi f/F_e}) = \cos(\pi f/F_e)e^{-j\pi f/F_e}$$

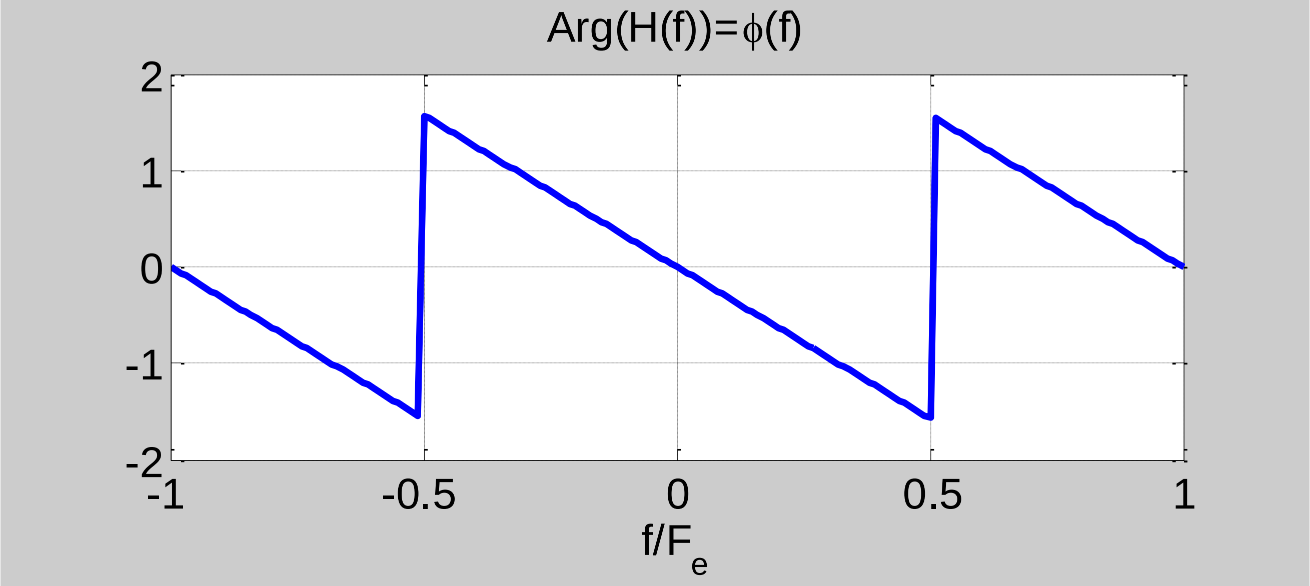
$$G(f) = \left|H(f)\right| = \left|\cos(\pi f/F_e)\right|$$



Example (2/2)

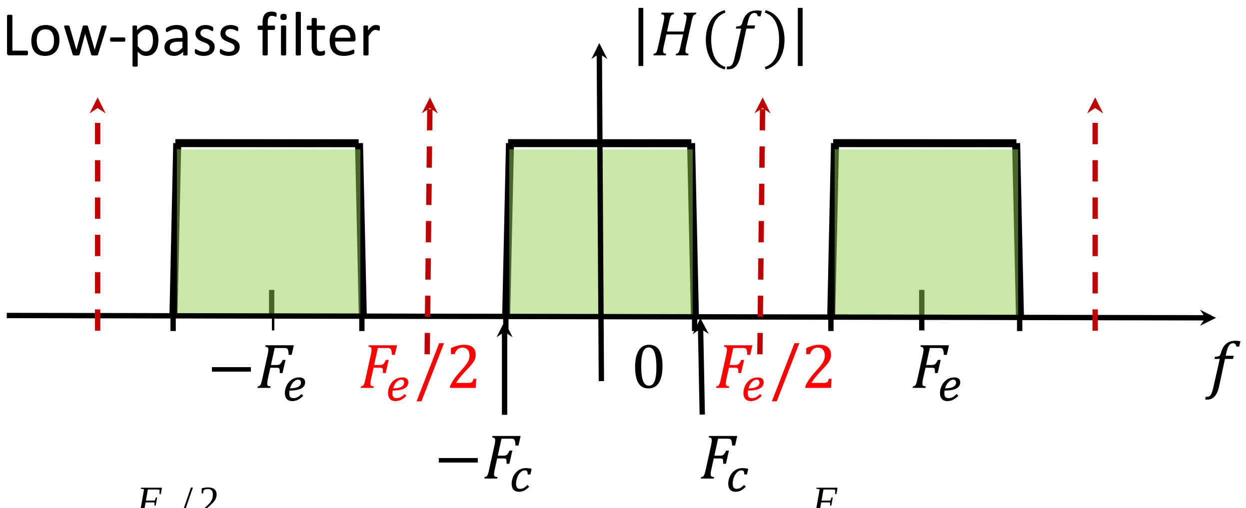
Averaging over 2 samples : $y_n = \frac{1}{2}(x_n + x_{n-1})$

$$\phi(f) = Arg(H(f)) = \begin{cases} -\pi f/F_e & \text{if} \quad \cos(\pi f/F_e) \succ 0 \\ -\pi f/F_e \pm \pi & \text{if} \quad \cos(\pi f/F_e) \prec 0 \end{cases}$$



FIR filter design

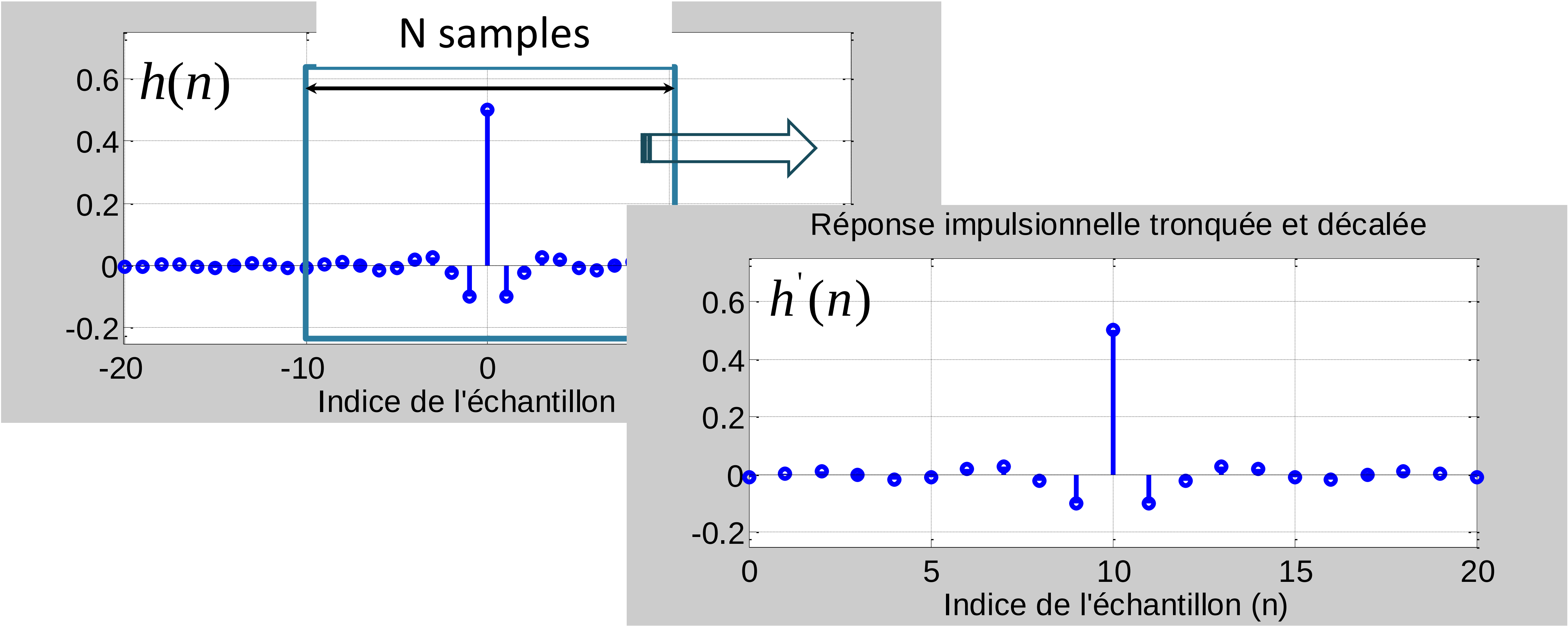
Window method, target response



$$h_n = \frac{1}{F_e} \int_{-F_e/2}^{F_e/2} H(f) e^{j2\pi nf/F_e} df = \frac{1}{F_e} \int_{-F_c}^{F_c} H(f) e^{j2\pi nf/F_e} df$$

$$= \frac{\sin(2\pi n F_c / F_e)}{2\pi n} = 2F_c / F_e \sin c(2\pi n F_c / F_e)$$

FIR filter design
Window method, impulse reponse



FIR filter design
Window method, impulse response

$$h_n^{'} = \begin{cases} 2(F_c / F_e)\,\mathrm{sinc}(2\pi(n - \frac{N-1}{2})F_c / F_e) & \text{if } 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases}$$

$$h^{'}(n) = h(n).w(n) * \delta(n - (\frac{N-1}{2}))$$

$$H^{'}(f) = TF(h(n)w(n) * \delta(n - (\frac{N-1}{2})))$$

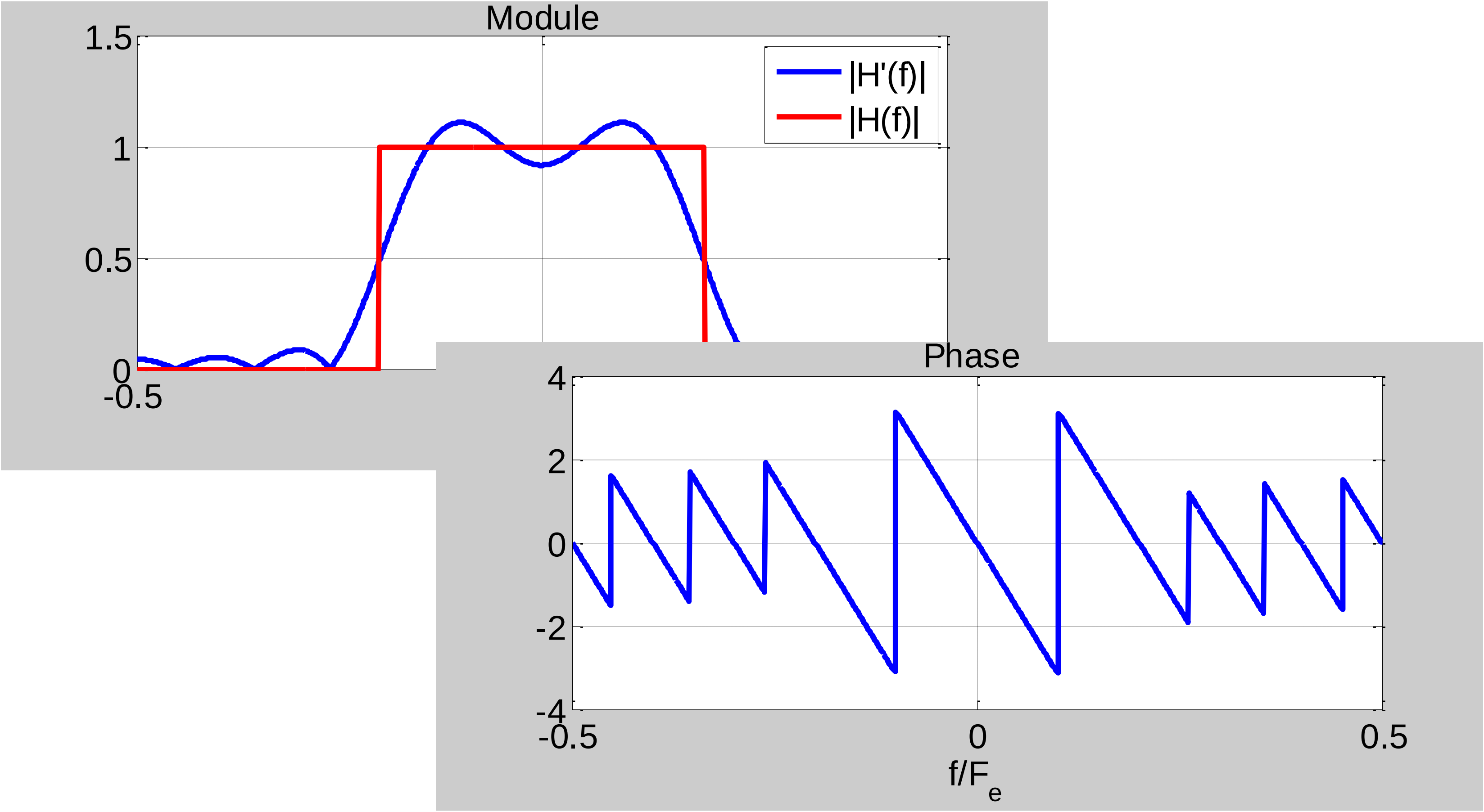
$$= (H(f) * W(f))e^{-j2\pi f(\frac{N-1}{2F_e})}$$

FIR filter design

Window method, frequency responses

$$F_c = 0.2F_e$$

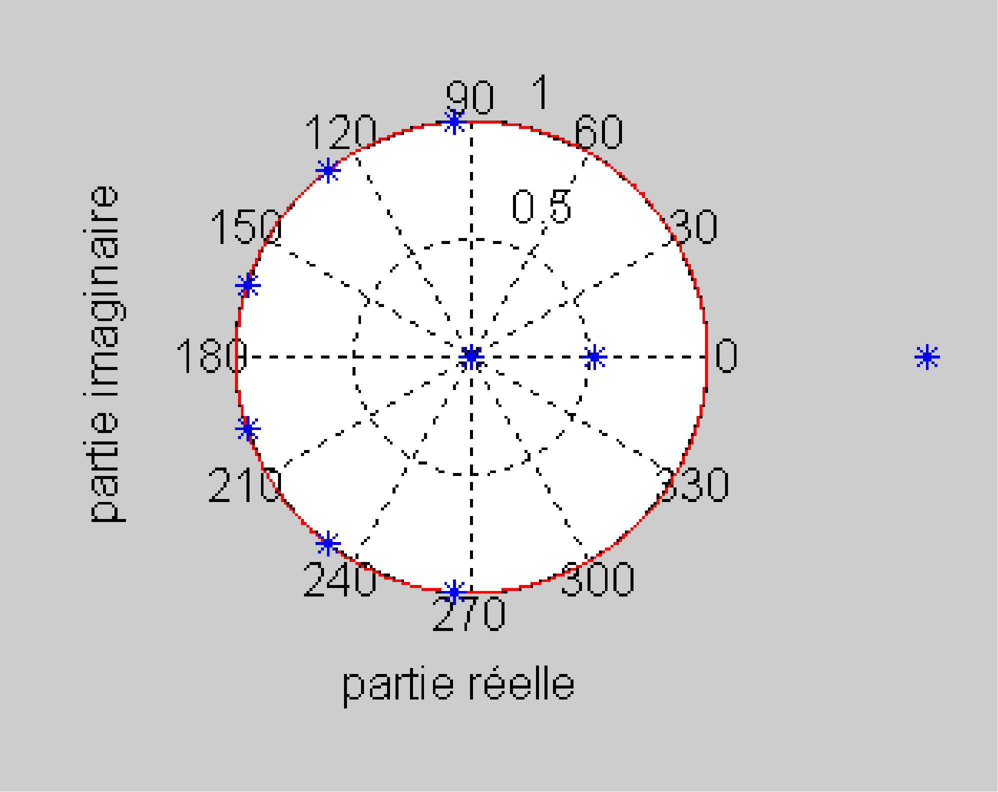
$$N = 11$$



FIR filter design
Window method, zeros

$$F_c = 0.2F_e$$

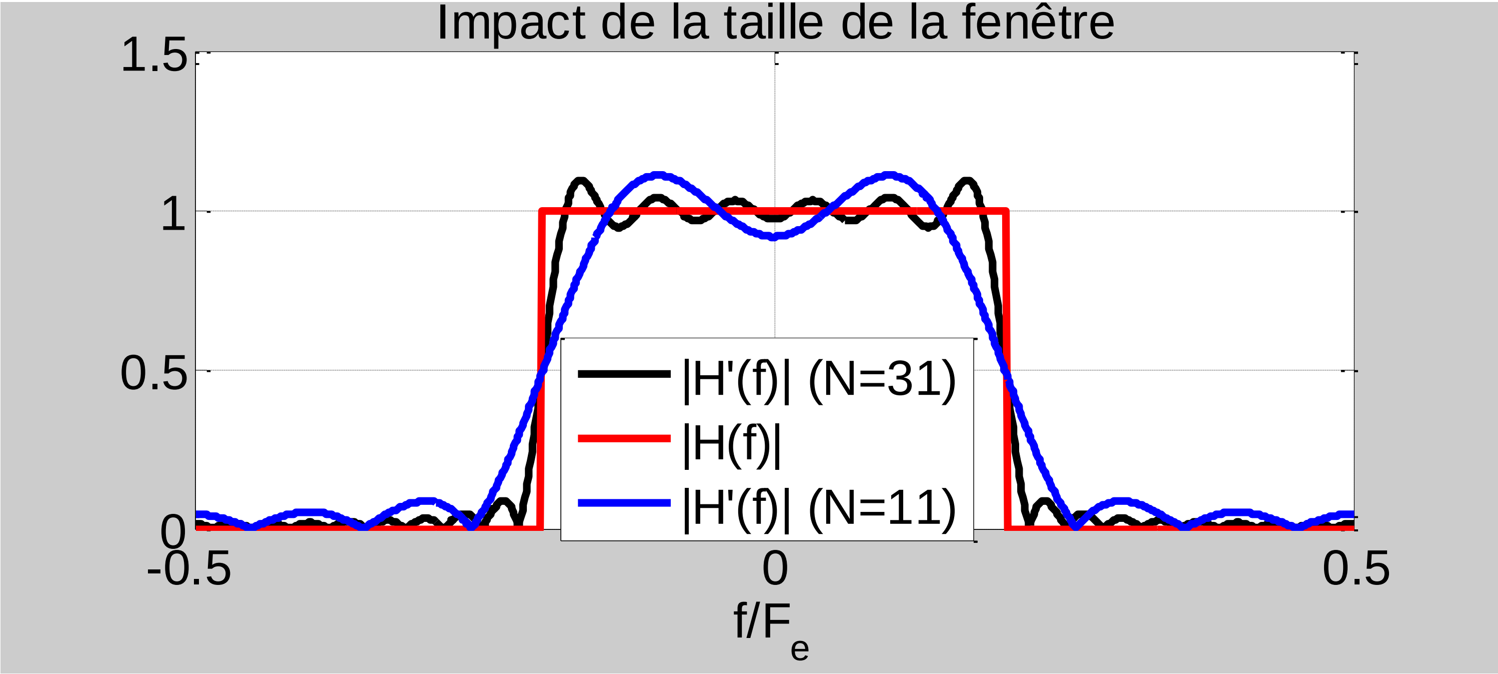
$$N = 11$$



FIR filter design

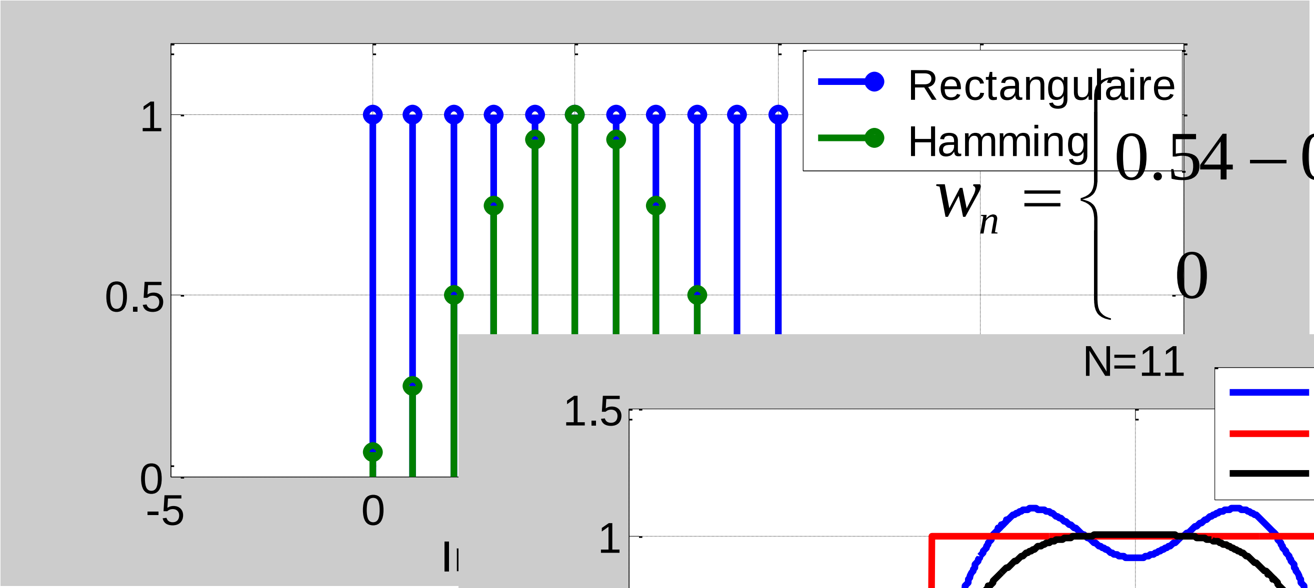
Window method, impact of the window size

$$x_c = F_c / F_e = 0.2$$

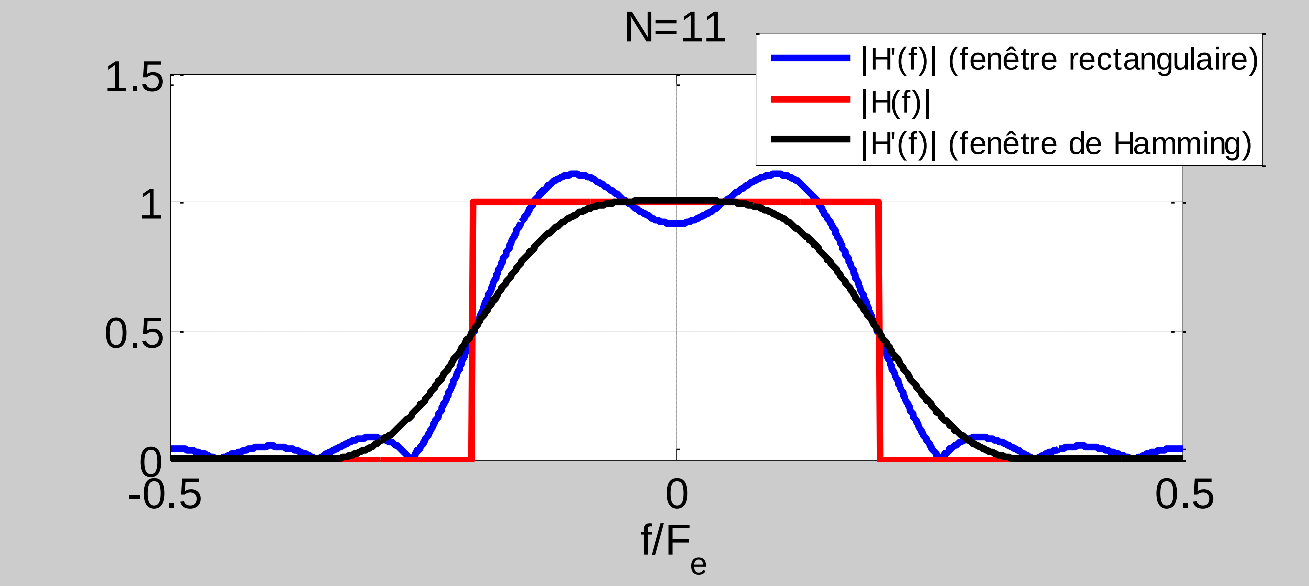


FIR filter design

Window method, impact of the window type

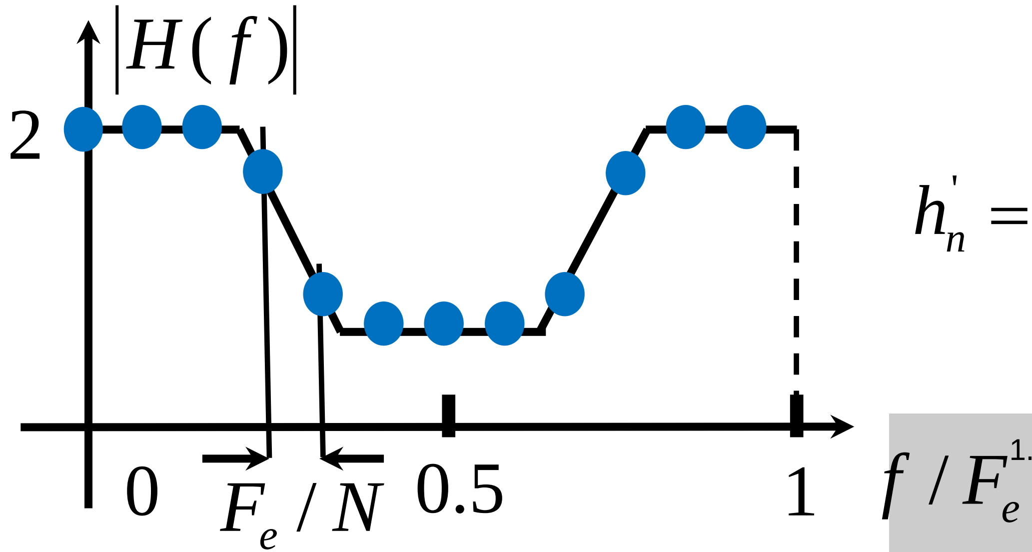


$$w_n = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right) & \text{if } 0 \leq n \leq N-1 \\ 0 & \text{elsewhere} \end{cases}$$

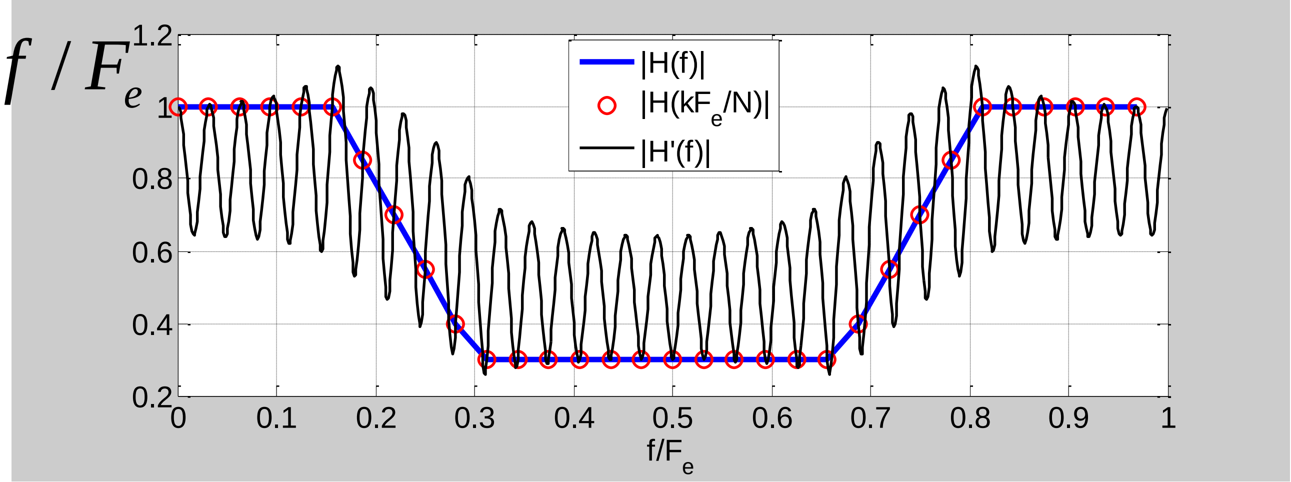


FIR filter design

Frequency sampling method

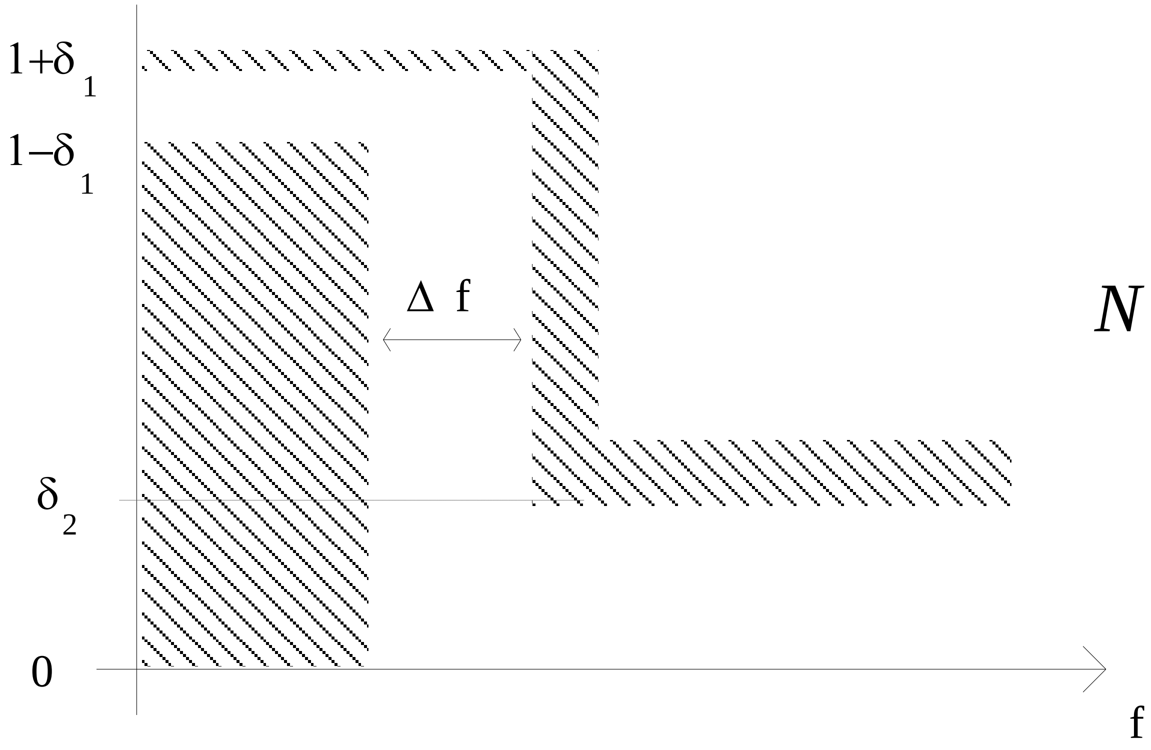


$$h'_n = \sum_{k=0}^{N-1} H\left(\frac{k}{NT_e}\right) e^{j2\pi \frac{k}{N} n} \quad \text{for} \quad n = 0 \text{ à } N-1$$



FIR filter design
Equiripple method

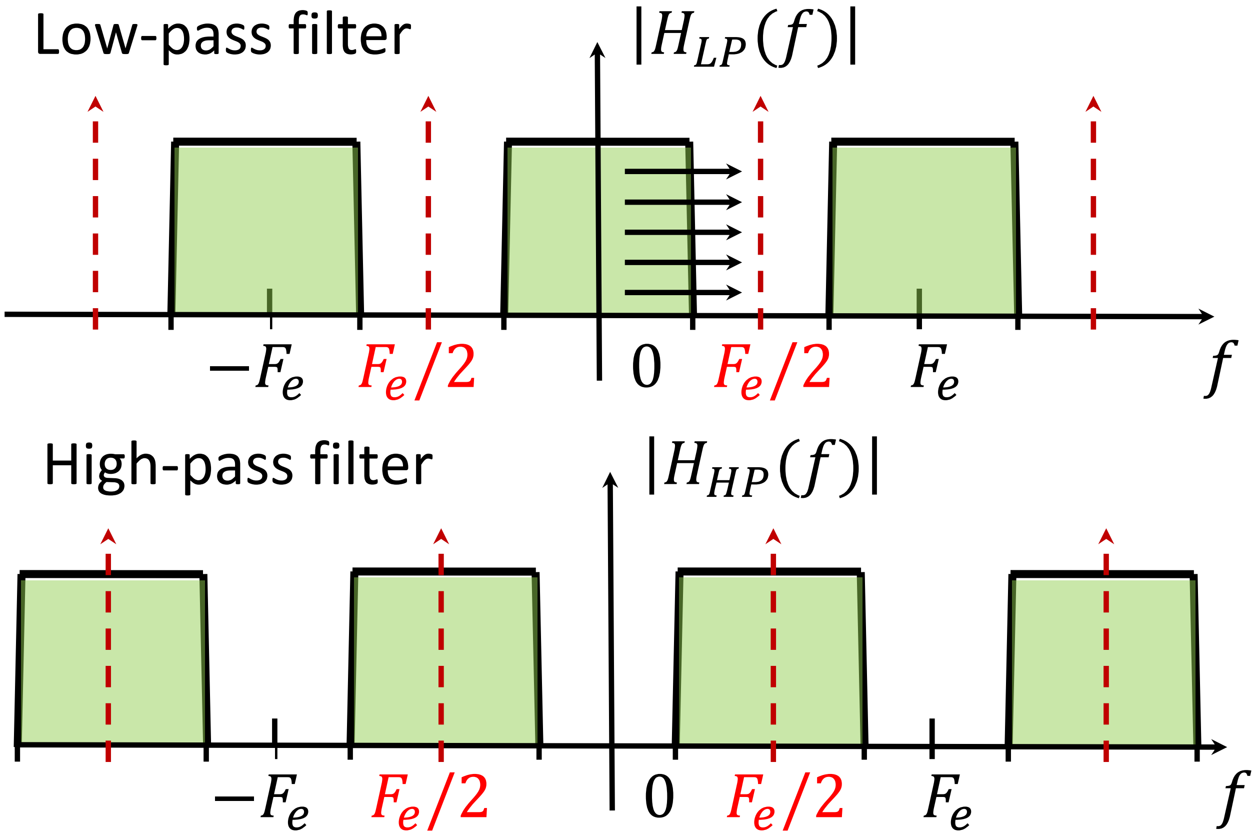
Control the modulus of the ripples in the bandwidth and in the rejected band.



$$N \approx \frac{2}{3}\log_{10}\left[\frac{1}{10\delta_1\delta_2}\right]\frac{F_e}{\Delta f}$$

FIR filter design

Low-pass filter → high-pass filter (1/2)



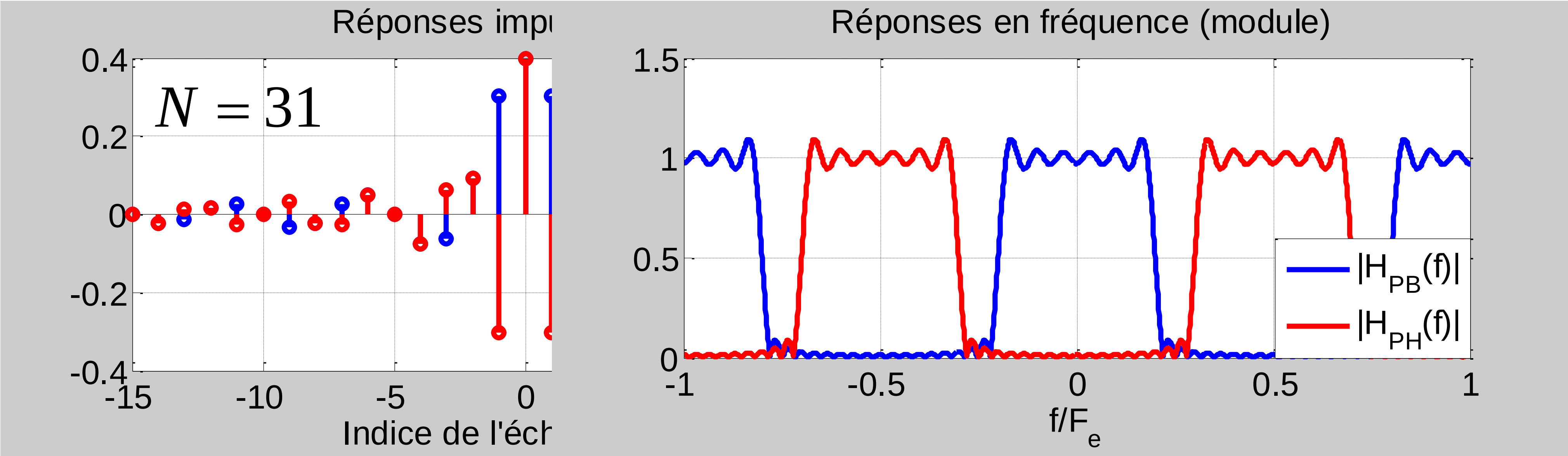
$$H_{HP}(f) = H_{LP}(f - \frac{F_e}{2})$$

FIR filter design

Low-pass filter → high-pass filter (2/2)

$$\sum_{m=0}^{M-1} h_m^{PH} e^{-j2\pi mf/F_e} = \sum_{m=0}^{M-1} h_m^{PB} e^{-j2\pi m(f-F_e/2)/F_e} = \sum_{m=0}^{M-1} (-1)^m h_m^{PB} e^{-j2\pi mf/F_e}$$

$$h_m^{PH} = (-1)^m h_m^{PB}$$



Example : moving average filter
Difference equation and transfer function

Difference equation

$$y_n = \frac{1}{N}\sum_{k=0}^{N-1} x_{n-k}$$

Transfer function

$$H(z) = \frac{1}{N}\sum_{k=0}^{N-1} z^{-k} = \frac{1}{N}\sum_{k=0}^{N-1} \frac{z^{N-k}}{z^{N}}$$

N poles at the origin plus N zeros.

Example : moving average filter

Frequency responses (1/2)

$$H(f)=\frac{1}{N}\sum_{k=0}^{N-1}e^{-j2\pi kf/F_s}$$

$H(f)$ is the sum of N components of a geometric series with a constant term $e^{-j2\pi f/F_e}$

$$H(f)=\frac{1}{N}\frac{1-e^{-j2\pi Nf/F_s}}{1-e^{-j2\pi f/F_s}}=\frac{e^{-j\pi Nf/F_s}}{Ne^{-j\pi f/F_s}}\frac{e^{j\pi Nf/F_s}-e^{-j\pi Nf/F_s}}{e^{j\pi f/F_s}-e^{-j\pi f/F_s}}$$

$$=\frac{e^{-j\pi(N-1)f/F_s}}{N}\frac{\sin(\pi Nf/F_s)}{\sin(\pi f/F_s)}=e^{-j\pi(N-1)f/F_e}\frac{\dfrac{\sin(\pi Nf/F_s)}{\pi Nf/F_s}}{\dfrac{\sin(\pi f/F_s)}{\pi f/F_s}}$$

$$=e^{-j\pi(N-1)f/F_s}\frac{\mathrm{sinc}(\pi Nf/F_s)}{\mathrm{sinc}(\pi f/F_s)}$$

Example : moving average filter

Frequency responses (2/2)

$$H(f)=\frac{\mathrm{sinc}(\pi Nf/F_s)}{\mathrm{sinc}(\pi f/F_s)}e^{-j\pi(N-1)f/F_s}=G(f)e^{j\phi(f)}$$

$$G(f)=\left|\frac{\mathrm{sinc}(\pi Nf/F_s)}{\mathrm{sinc}(\pi f/F_s)}\right| \qquad \phi(f)=-2\pi\frac{N-1}{2F_s}f$$

$$G(f)=\left|\frac{\mathrm{sinc}(\pi Nf/F_s)}{\mathrm{sinc}(\pi f/F_s)}\right|=0 \text{ for } f=k\frac{F_s}{N}$$

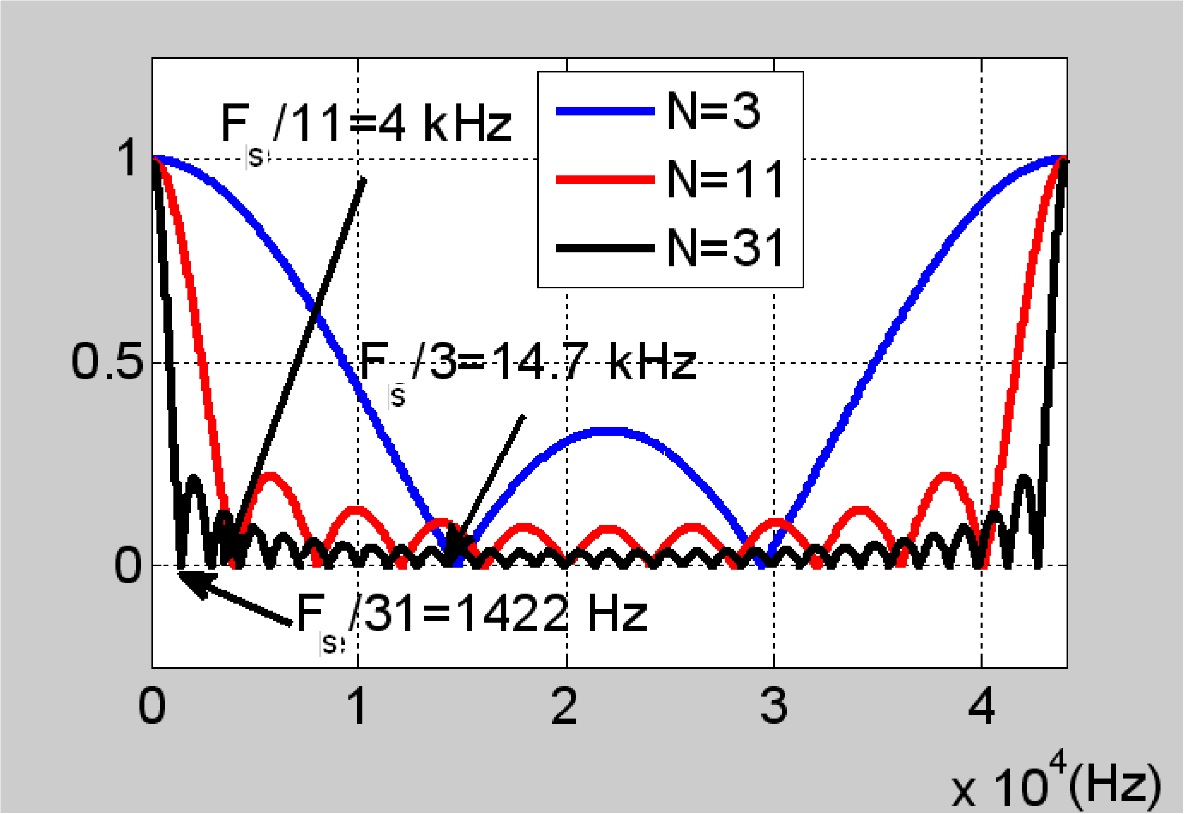
$$\phi(f)=-2\pi\frac{N-1}{2F_s}f \Rightarrow \text{linear phase with } f$$

This filter introduces a delay $\tau=\frac{N-1}{2F_s}$

Example : moving average filter

Amplitude responses

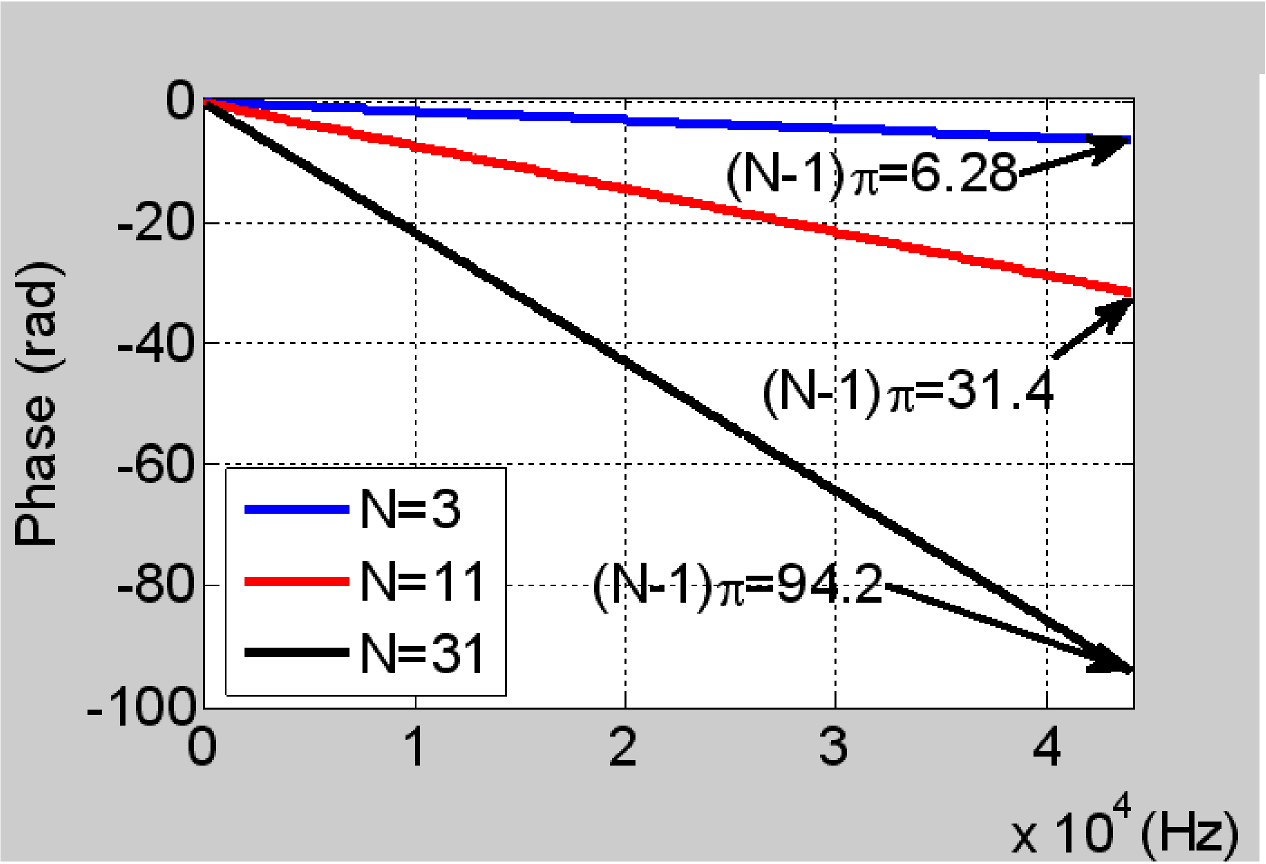
$$F_s = F_e = 44.1\ kHz$$



Example : moving average filter

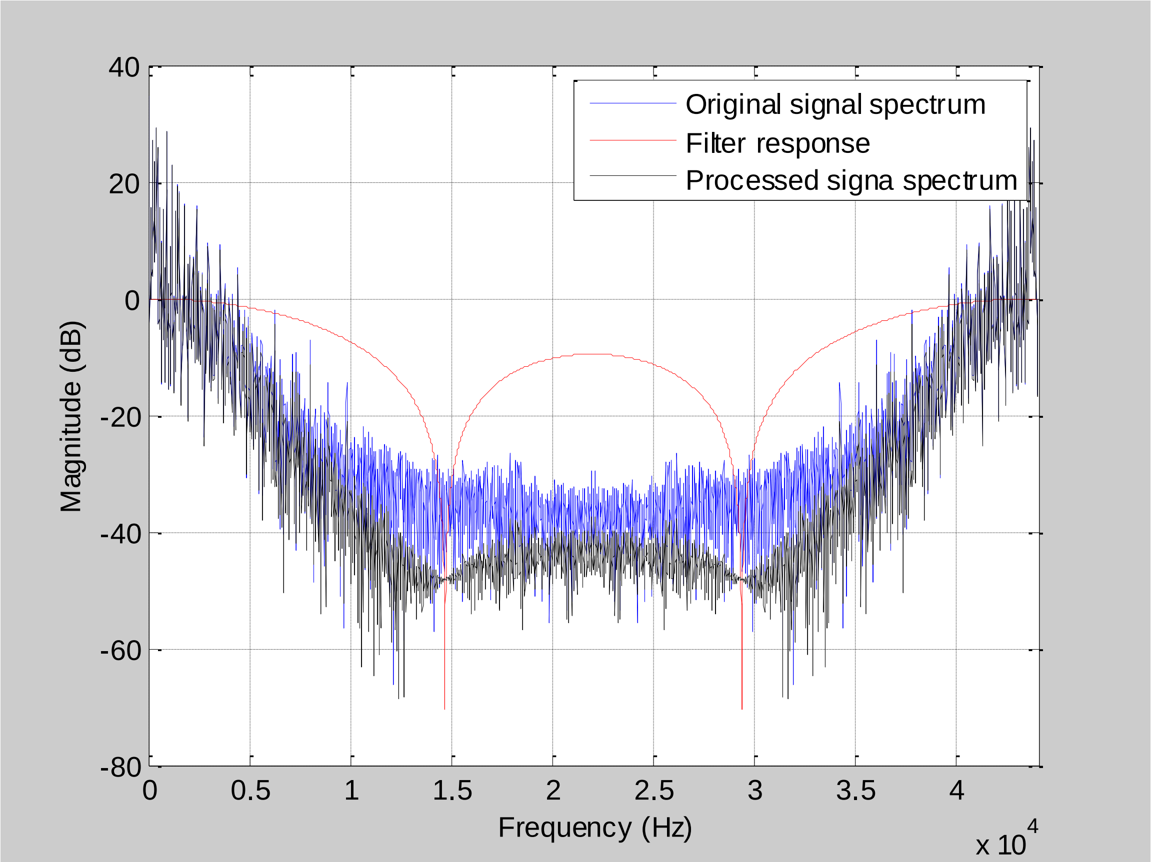
Phase responses

$F_s = F_e = 44.1\ kHz$



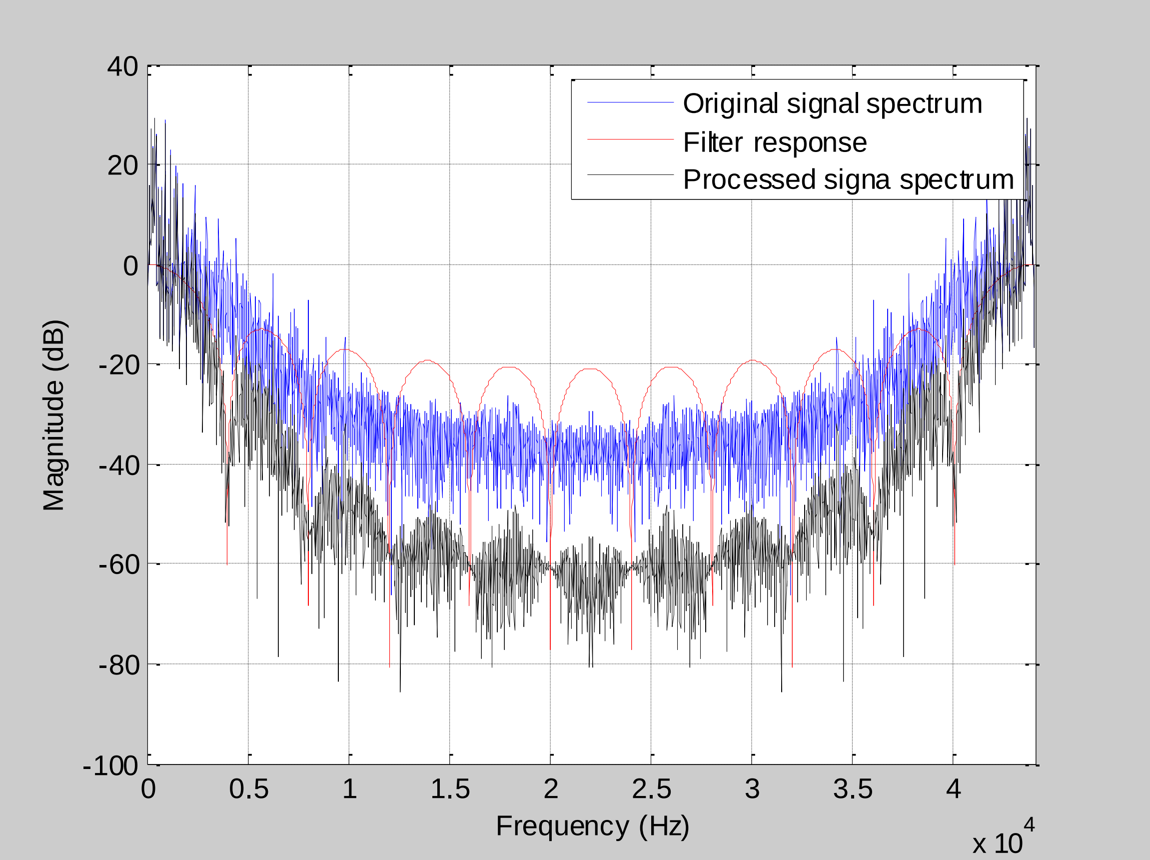
Example : moving average filter

N=3

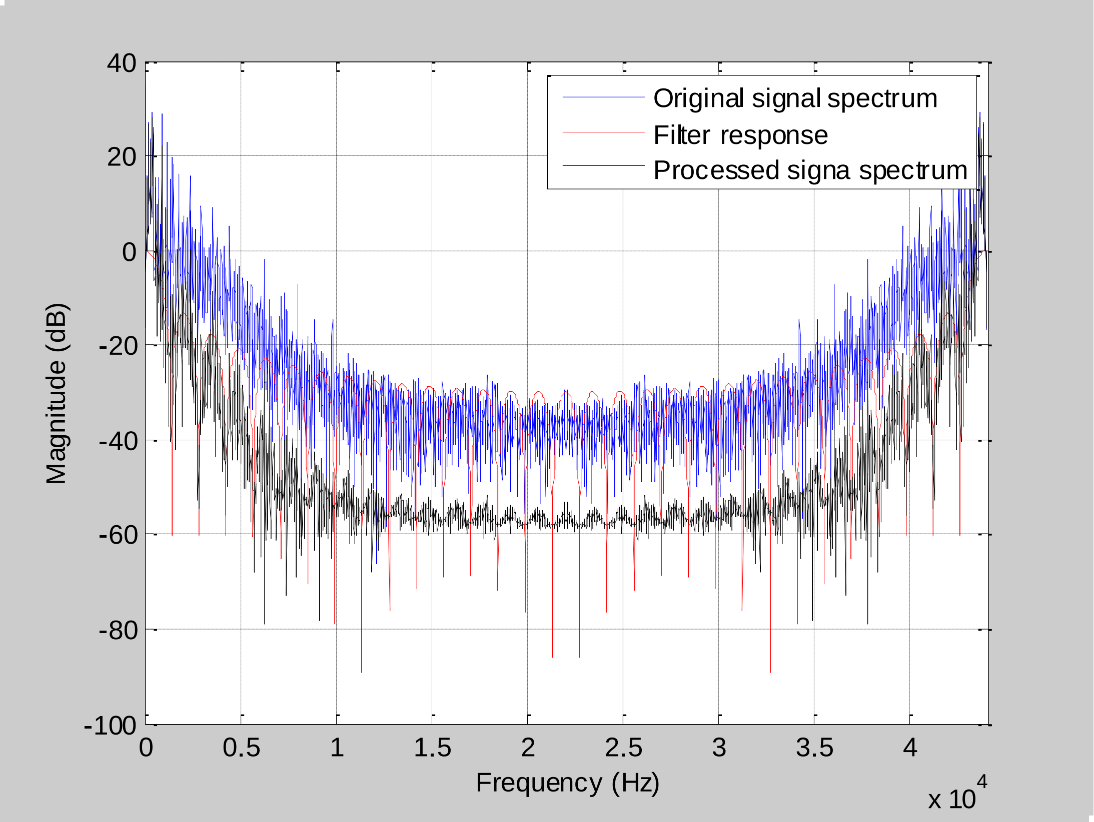


Example : moving average filter

N=11



Example : moving average filter
N=31



Outline

2. Digital data filtering

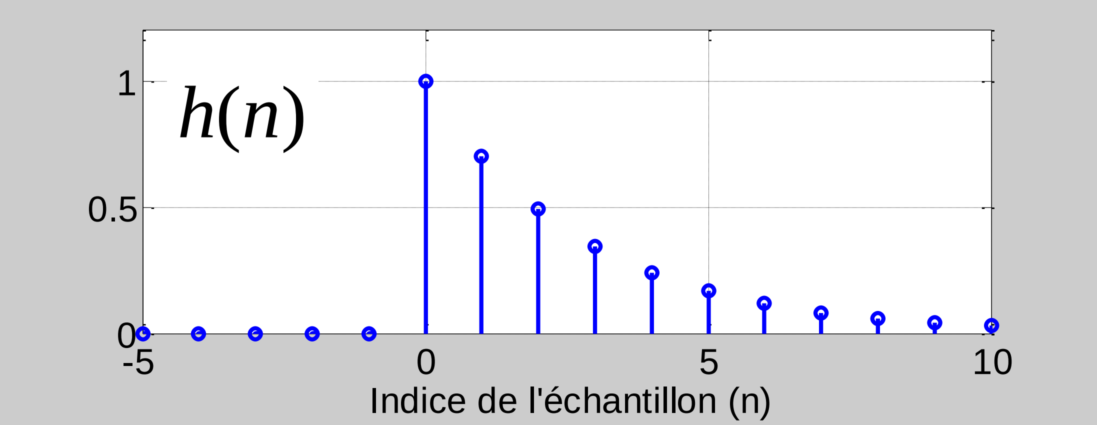
- Finite Impulse Response (FIR) filters
- Infinite Impulse Response (IIR) filters

Impulse response

$$x(n) \longrightarrow \boxed{H} \longrightarrow y(n)$$

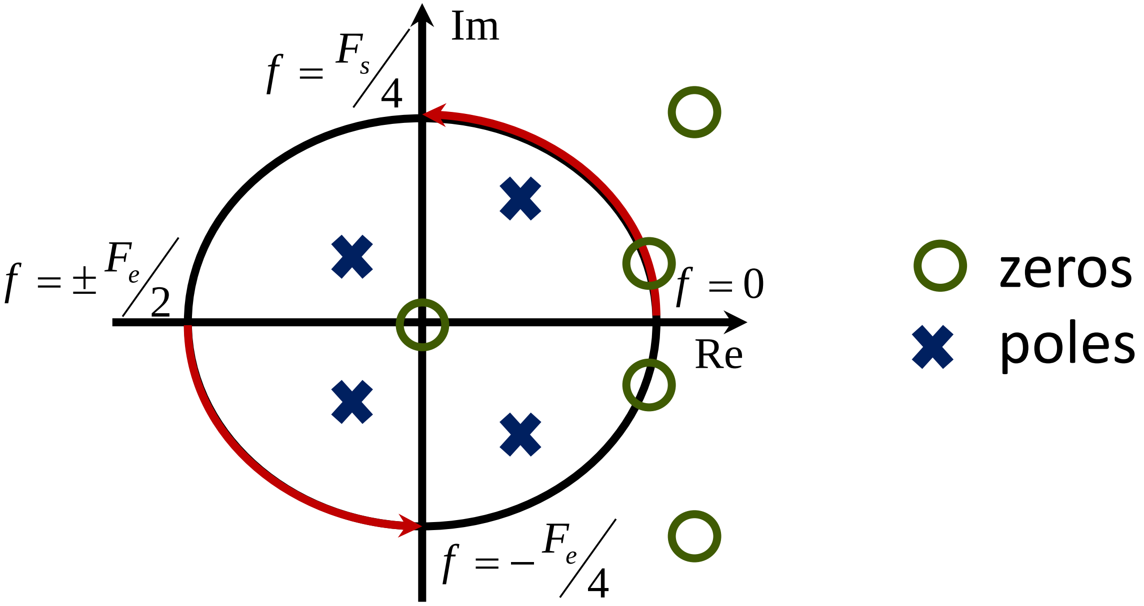
The output of this filter at the instant n depends on the input samples and also outputs to the preceding instants

$$y_n = \sum_{m=0}^{M-1} \frac{b_m}{a_0} x_{n-m} - \sum_{k=1}^{K-1} \frac{a_k}{a_0} y_{n-k}$$



Stability

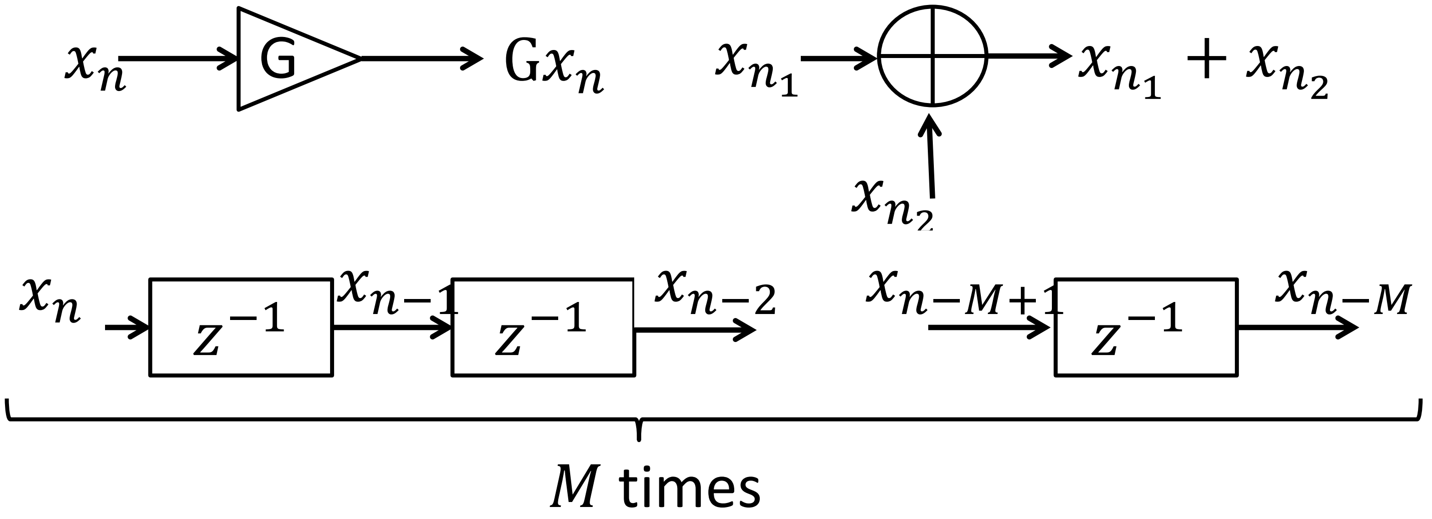
An IIR filter is stable when all its poles are located inside the unit circle.



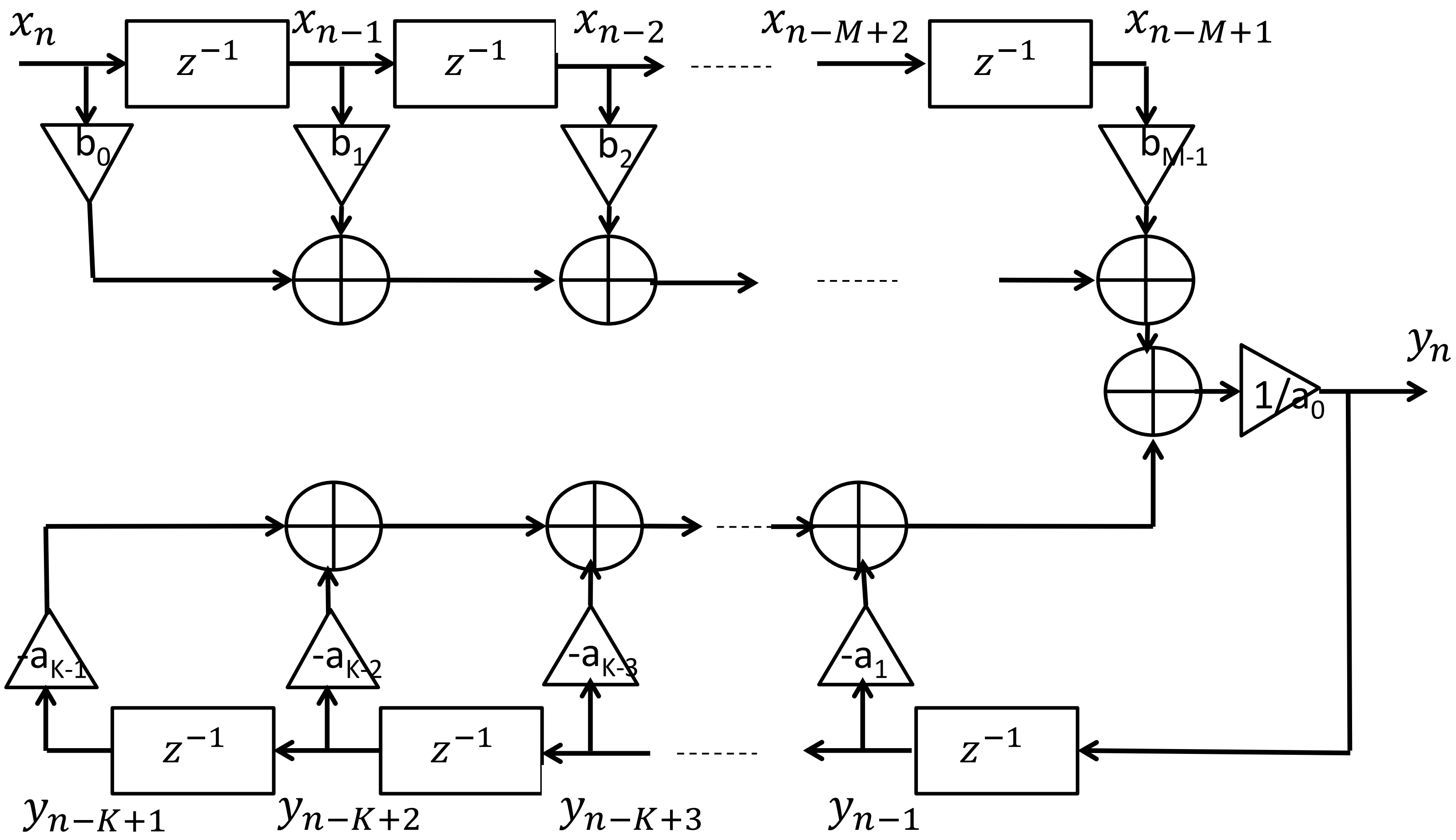
Implementation schemes (1/3)

$$y_n = {}^{1}\!/_{a_0}\left(\sum_{m=0}^{M-1} b_m x_{n-m} - \sum_{k=1}^{K-1} a_k y_{n-k}\right)$$

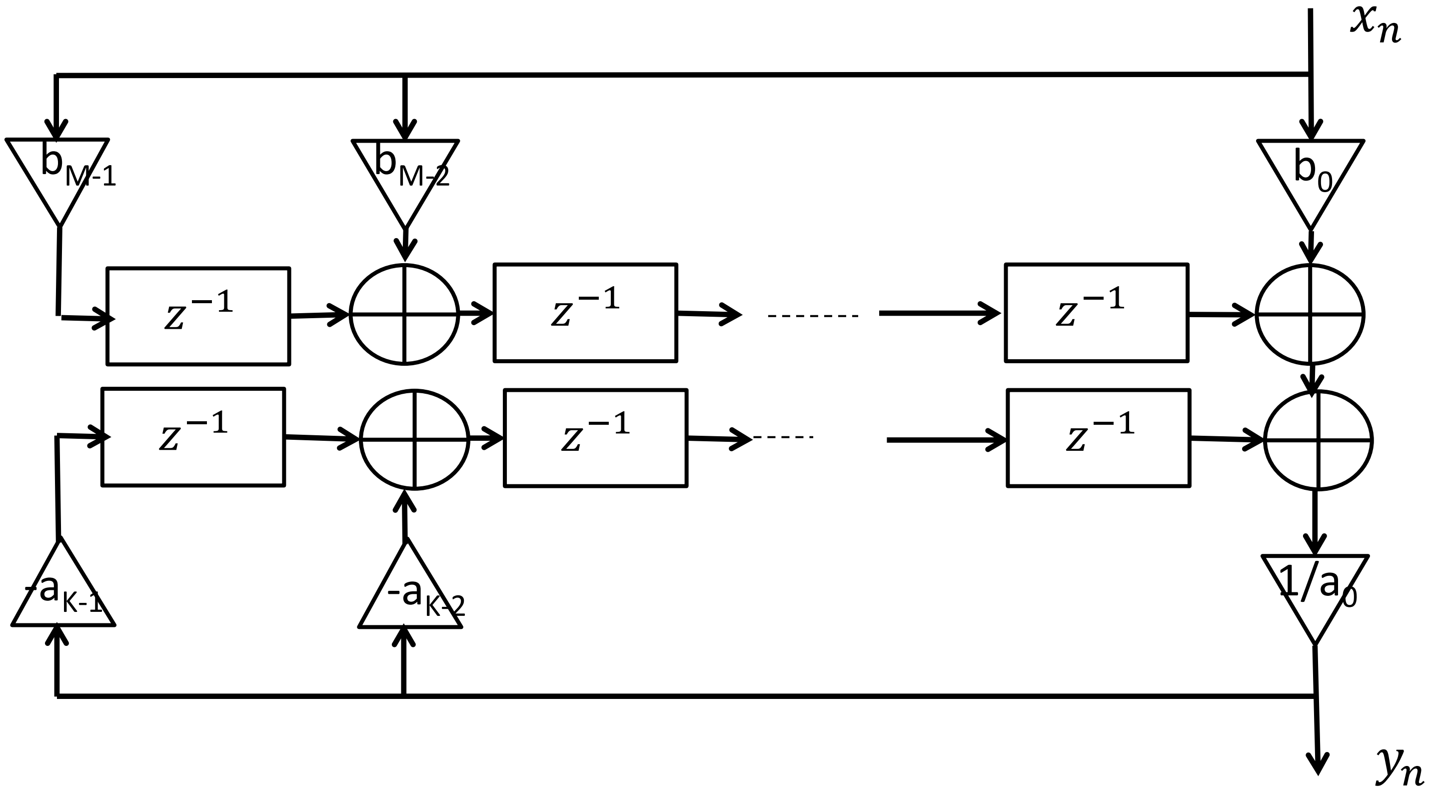
The linear equation differences can be represented by a diagram in which gain, delay and sum are represented by functional blocks



Implementation schemes (2/3)

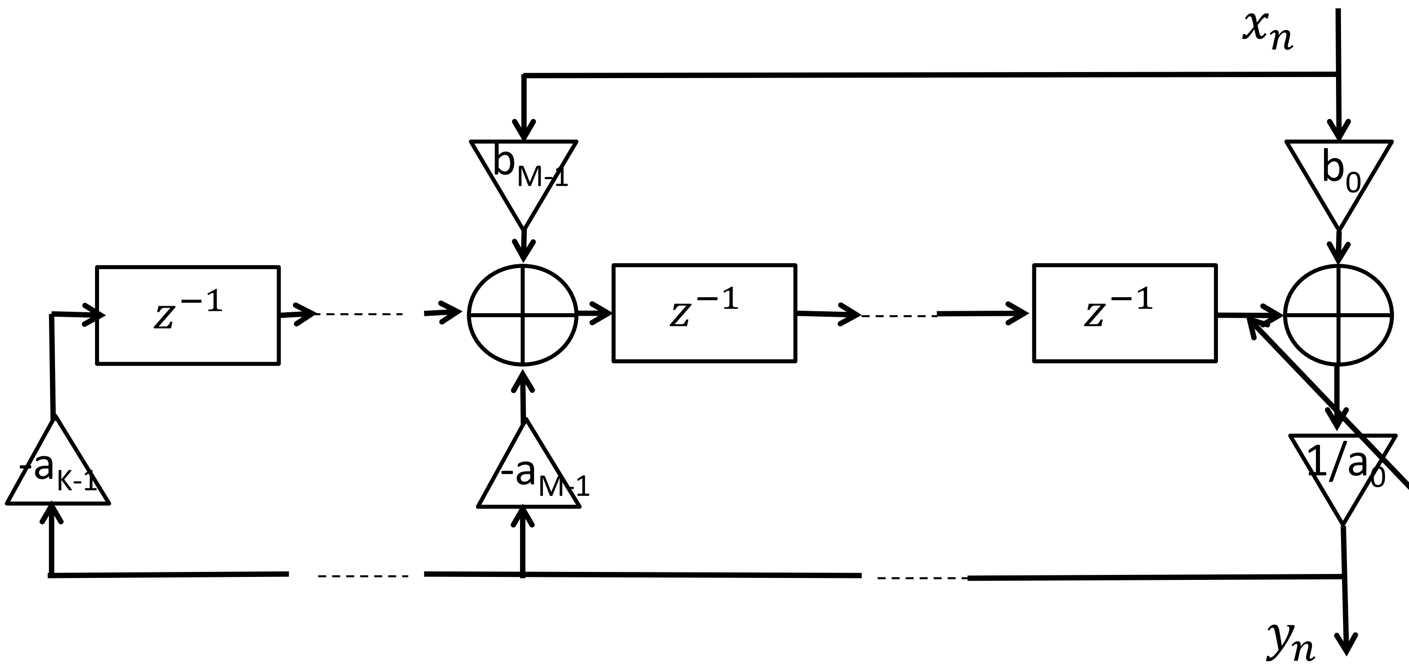


Implementation schemes (3/3)



$$\frac{Y(z)}{X(z)} = \frac{N(z)}{D(z)}$$

Implementation scheme ND

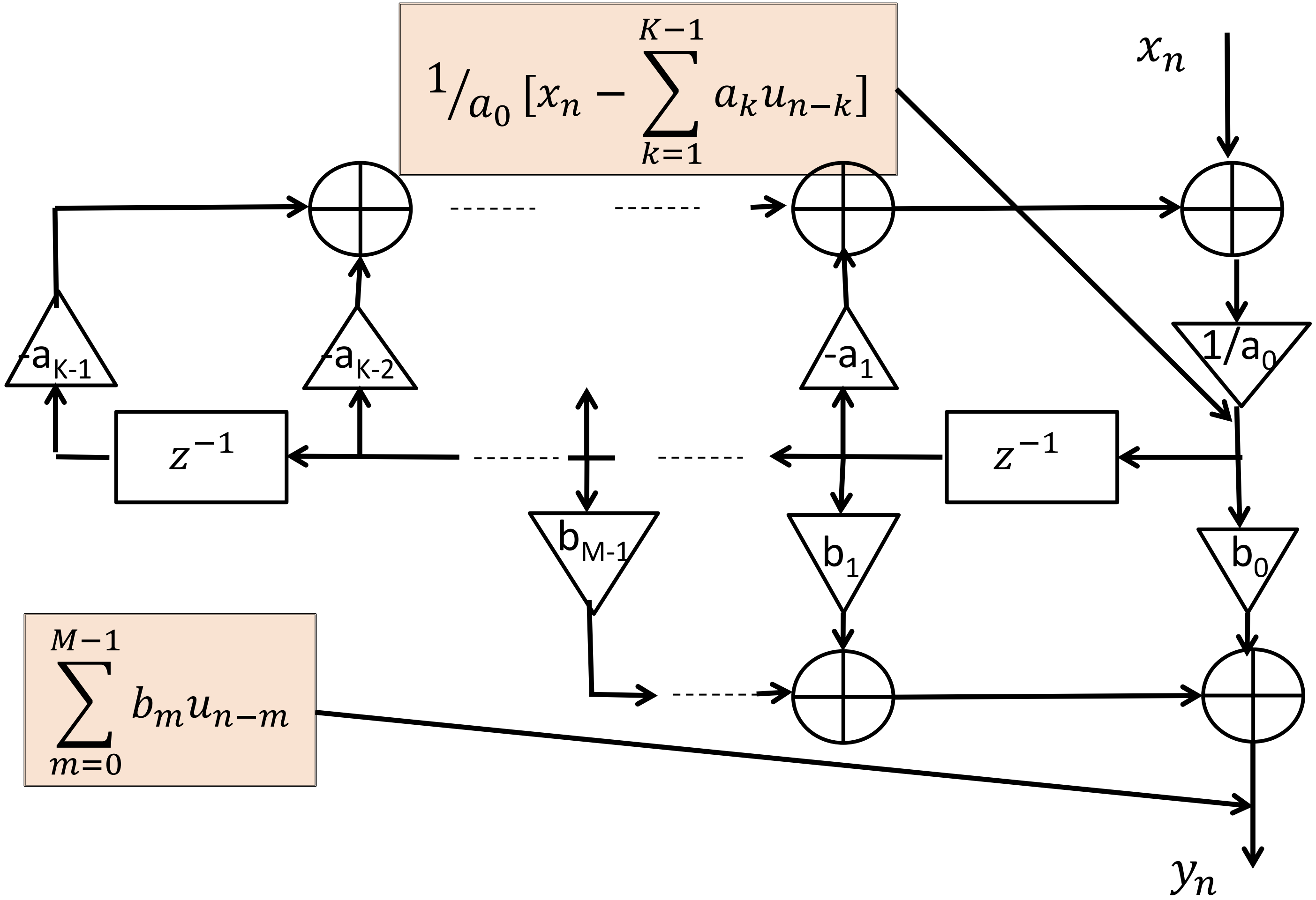


$$\frac{Y(z)}{X(z)} = \frac{N(z)}{D(z)}$$

$$Y(z) = N(z)X(z)\frac{1}{D(z)}$$

$$\sum_{m=0}^{M-1} b_m x_{n-m} - \sum_{k=0}^{K-1} a_k y_{n-k}$$

Implementation scheme DN



$$\frac{Y(z)}{X(z)} = \frac{N(z)}{D(z)}$$

$$Y(z) = \underbrace{\frac{1}{D(z)} X(z)}_{U(z)} N(z)$$

where

$$\begin{cases} U(z) = \dfrac{1}{\sum_{k=0}^{K-1} a_k z^{-k}} X(z) \\ Y(z) = \sum_{k=0}^{M-1} b_m z^{-m} U(z) \end{cases}$$

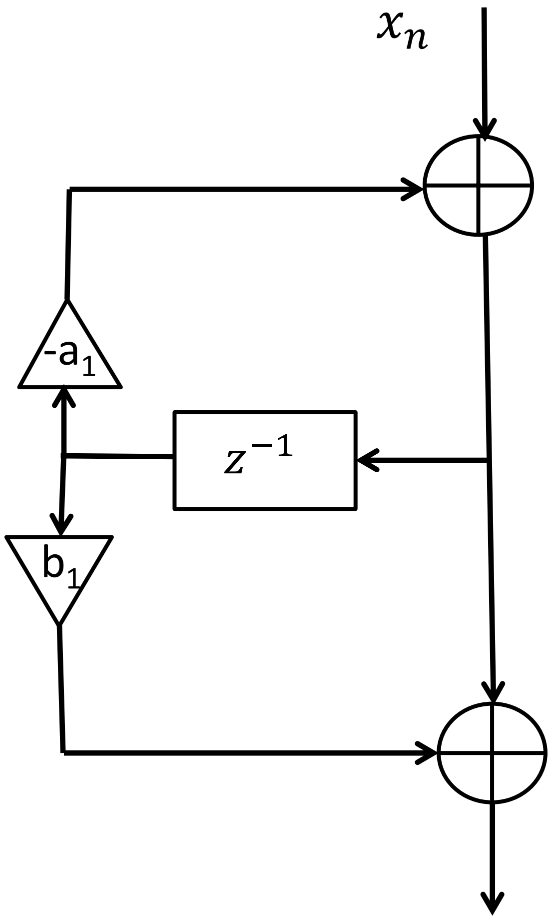
Example 1 : 1st order filter (1/4)

$$x_n + b_1 x_{n-1} = y_n + a_1 y_{n-1}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$

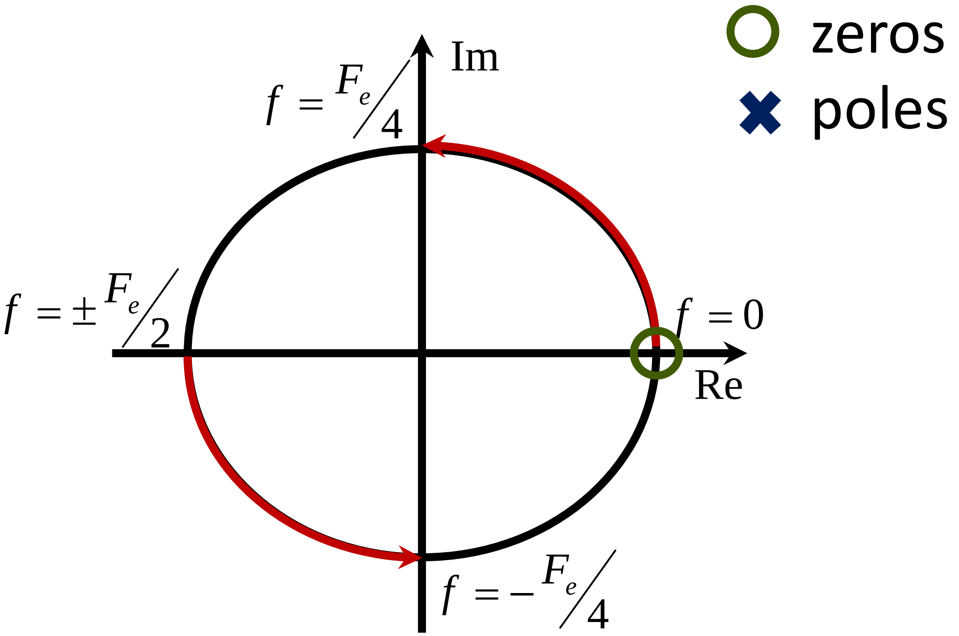
$$= \frac{z + b_1}{z + a_1}$$

$$= \frac{z - z_1}{z - p_1} \quad \rightarrow \quad \begin{cases} b_1 = -z_1 \\ a_1 = -p_1 \end{cases}$$



Example 1 : 1st order filter (2/4)

A filter as $|H(0)| = 0$, $\rightarrow$ $b_1 = -z_1 = -1$



Example 1 : 1st order filter (3/4)

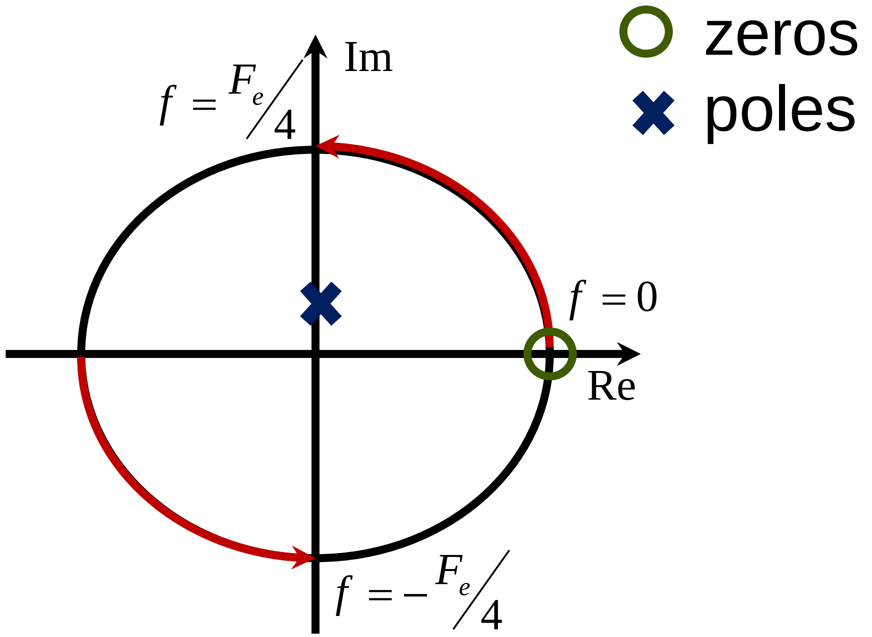
A pole placed at $F_e/4$ such that $|H(F_e/4)| = 2, \rightarrow \quad \mathrm{p}_1 = |\mathrm{p}_1| e^{j2\pi F_e/4F_e} = j|\mathrm{p}_1|$

$$|H(F_e/4)| = \left| \frac{e^{j2\pi F_e/4F_e} - 1}{e^{j2\pi F_e/4F_e} - j|p_1|} \right|$$

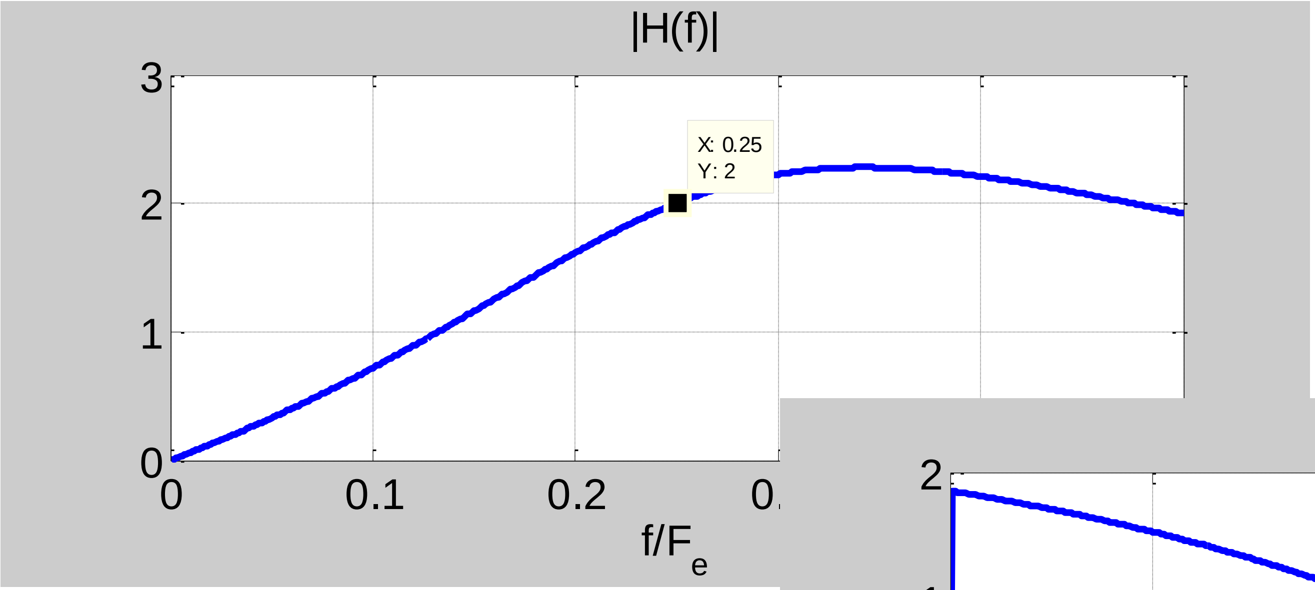
$$= \left| \frac{j-1}{j - j|p_1|} \right| = 2 \qquad f = \pm F_e/2$$

$$\rightarrow \quad |p_1| = 1 - \sqrt{2}/2$$

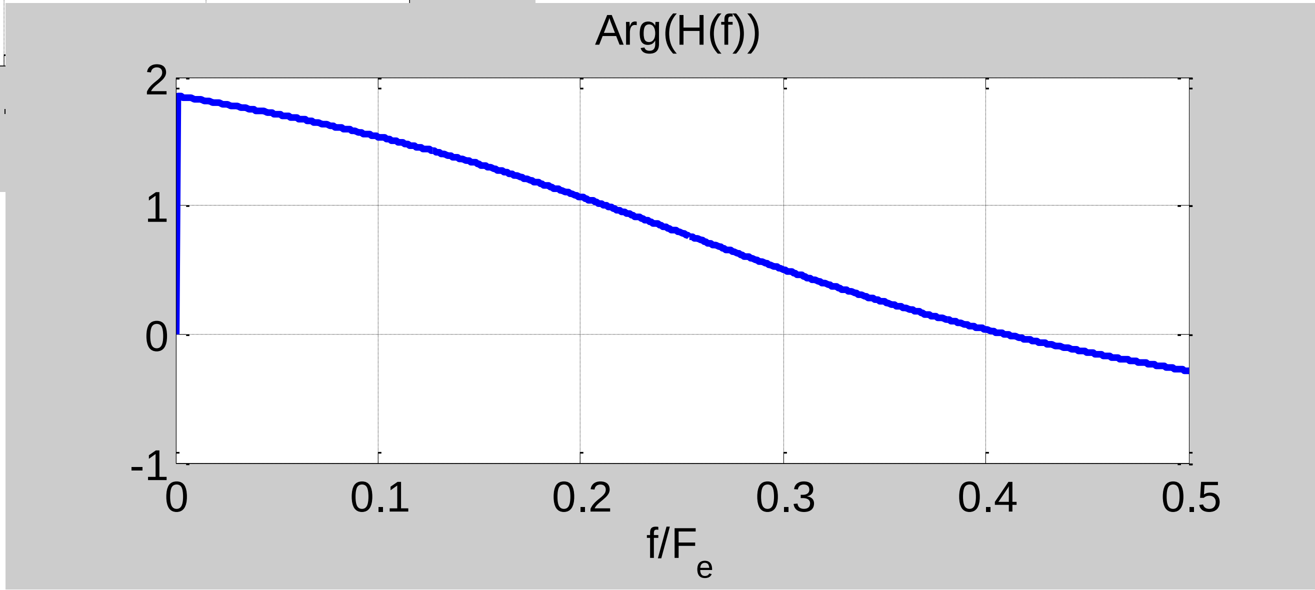
$$\rightarrow \quad a_1 = -p_1 = -(1 - \sqrt{2}/2)\,j$$



Example 1 : 1st order filter (4/4)



Frequency responses

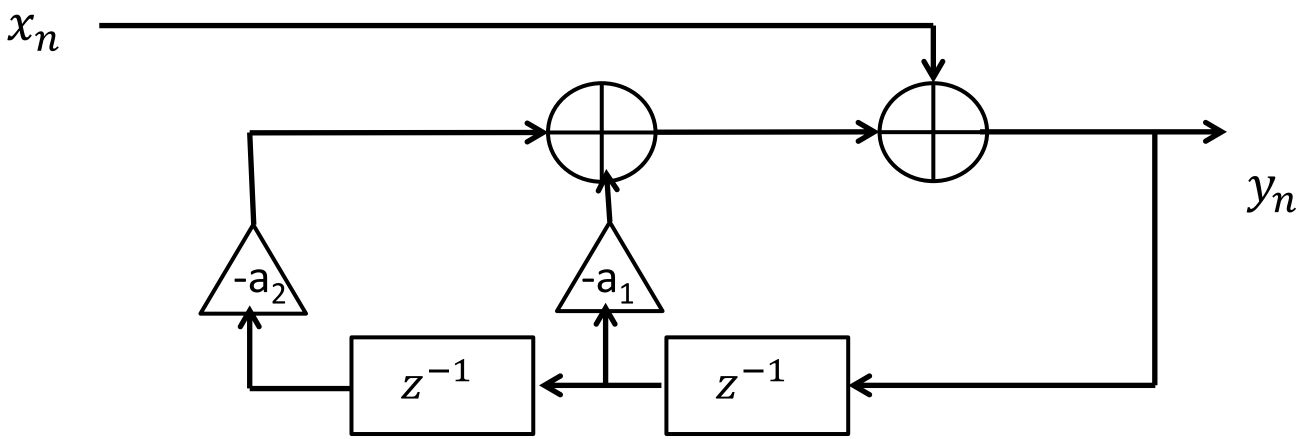


Example 2 : pure recursive 2nd order filter (1/7)

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{\sum_{k=0}^{2} a_k z^{-k}} = \frac{1}{a_0 + a_1 z^{-1} + a_2 z^{-2}} = \frac{1}{D(z)}$$

Without loss of generality, we assume that $a_0 = 1$

Difference equation $y_n = x_n - a_1 y_{n-1} - a_2 y_{n-2}$



Example 2 : pure recursive 2nd order filter (2/7)

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{z^2}{z^2 + a_1 z + a_2} = \frac{z^2}{(z - p_1)(z - p_2)}$$

$$a_1 = -(p_1 + p_2) \text{ and } a_2 = p_1 p_2$$

This filter has two zeros at the origin and two poles p_1 and p_2.

$p_1 = \frac{-a_1 + \sqrt{\Delta}}{2}$ and $p_2 = \frac{-a_1 - \sqrt{\Delta}}{2}$ with $\Delta = {a_1}^2 - 4a_2$

Two cases are distinguished by the sign of Δ

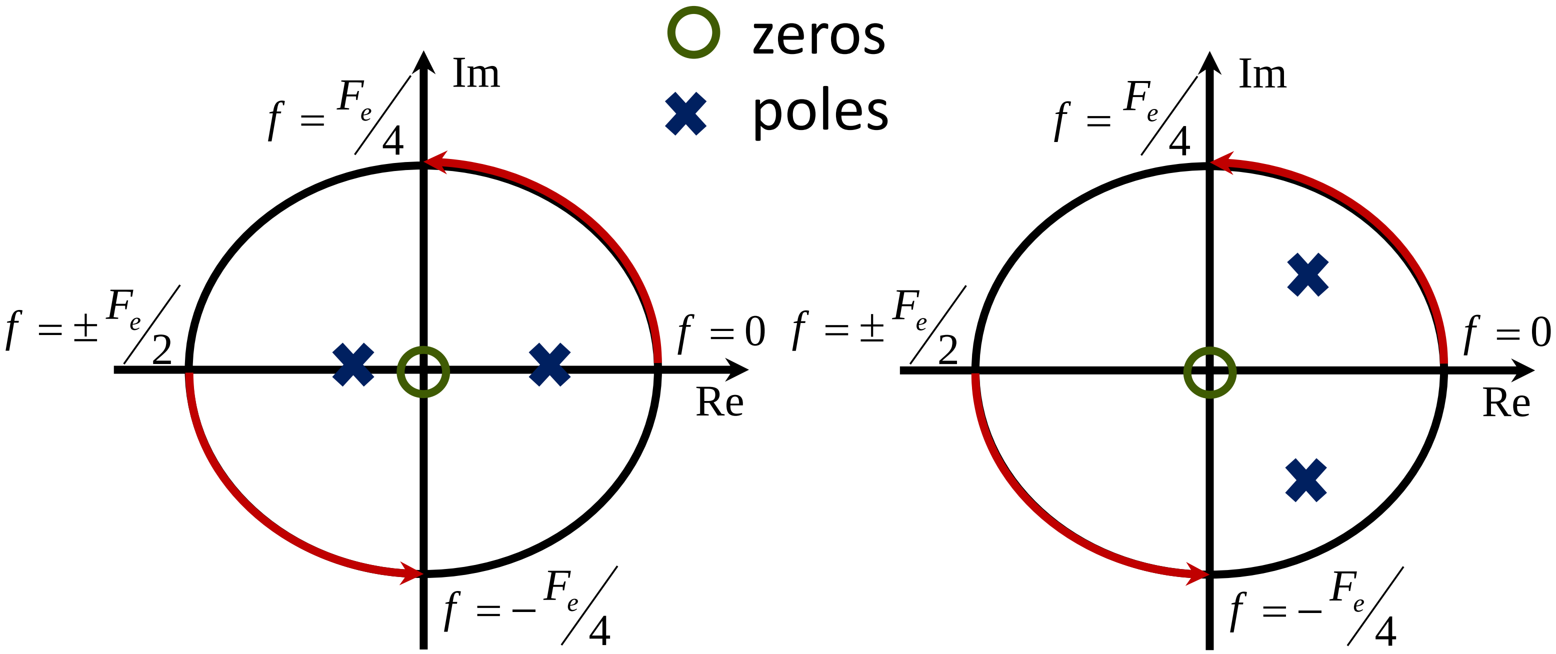
Example 2 : pure recursive 2nd order filter (3/7)

1st case : $\Delta \geq 0$

2nd case : $\Delta < 0$

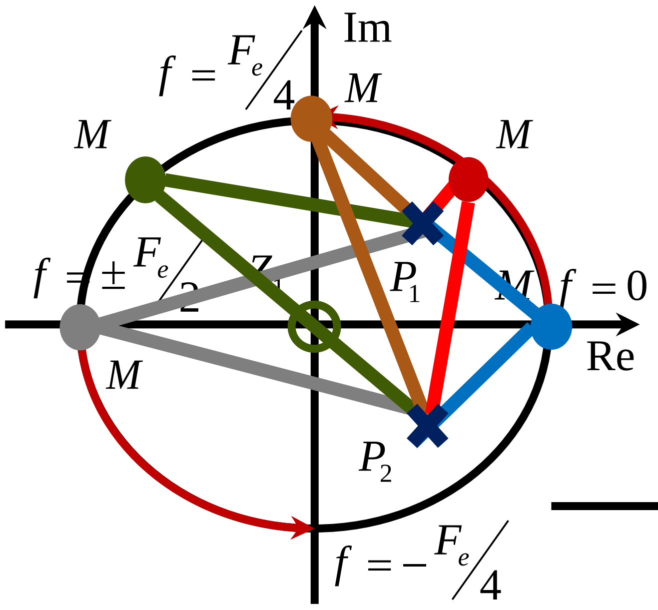
Real Poles

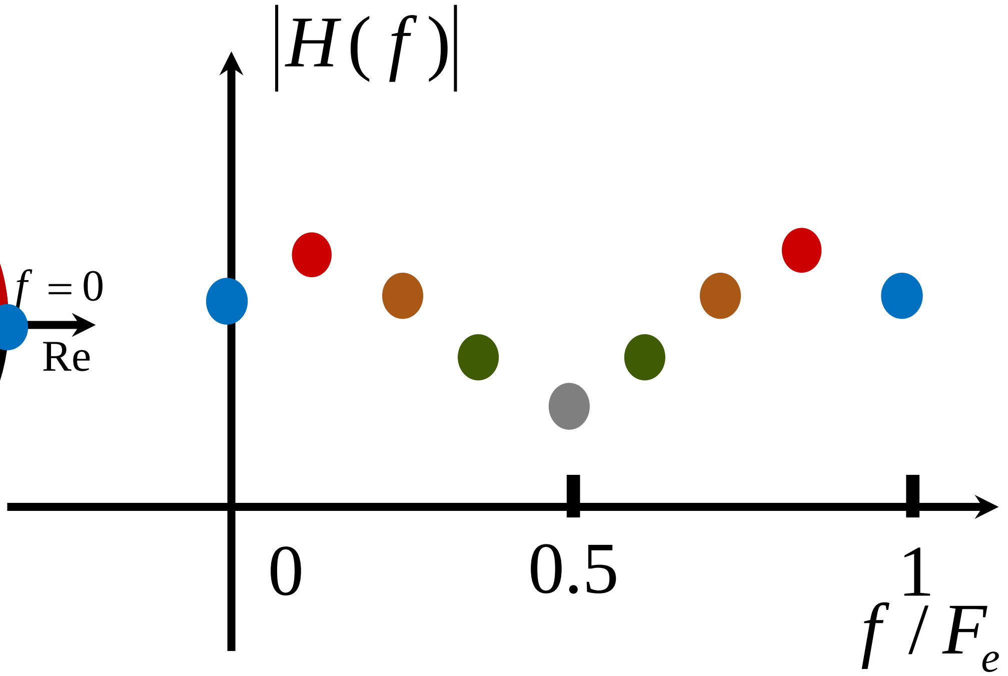
Complex conjugate Poles



Example 2 : pure recursive 2nd order filter (4/7)

$$H(z)=\frac{z^2}{(z-p_1)(z-p_2)} \qquad \left|H(f)\right|=\frac{1}{MP_1MP_2}$$





Example 2 : pure recursive 2nd order filter (5/7)

$$\left|H(f)\right| = \frac{1}{\sqrt{1+a_1^2+a_2^2+2a_1(1+a_2)\cos(2\pi f/F_e)+2a_2\cos(4\pi f/F_e)}}$$

$$Arg(H(f)) = -Arctg\left[\frac{a_1\sin(2\pi f/F_e)+a_2\sin(4\pi f/F_e)}{1+a_1\cos(2\pi f/F_e)+a_2\cos(4\pi f/F_e)}\right]$$

The resonance frequency is solution of ➔ $\dfrac{\partial\left|H(f)\right|}{\partial f} = 0$

$$\sin(2\pi f/F_e)\left[2a_1(1+a_2)+8a_2\cos(2\pi f/F_e)\right] = 0$$

$$F_r = \frac{F_e}{2\pi}\text{Arcos}\left(-\frac{a_1(1+a_2)}{4a_2}\right)$$

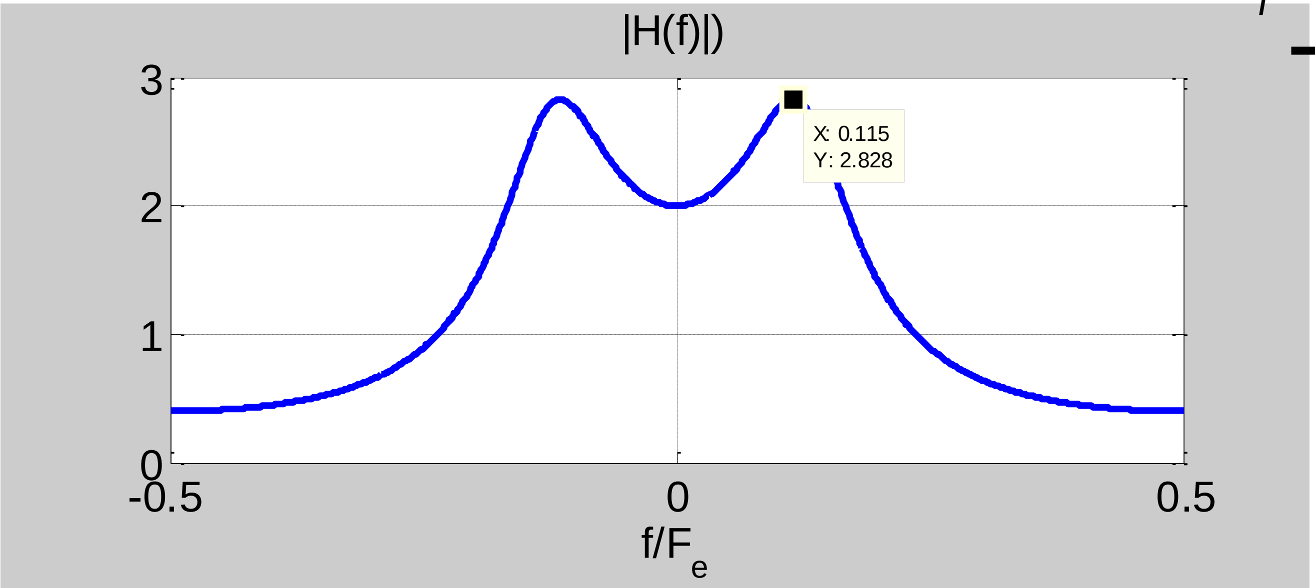
Example 2 : pure recursive 2nd order filter (6/7)

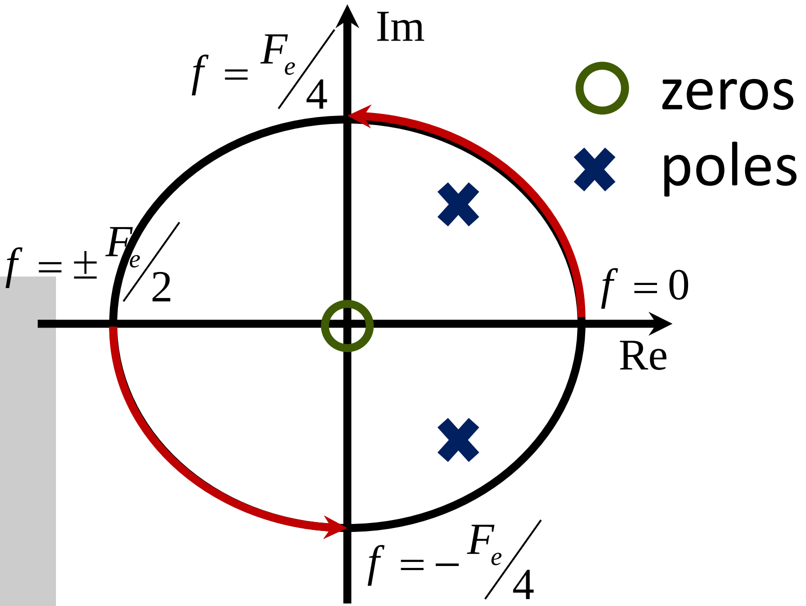
Two zeros at the origin and two complex conjugate poles

$$p_1 = \sqrt{2}/_2\, e^{j\pi/4} \text{ et } p_2 = \sqrt{2}/_2\, e^{-j\pi/4}$$

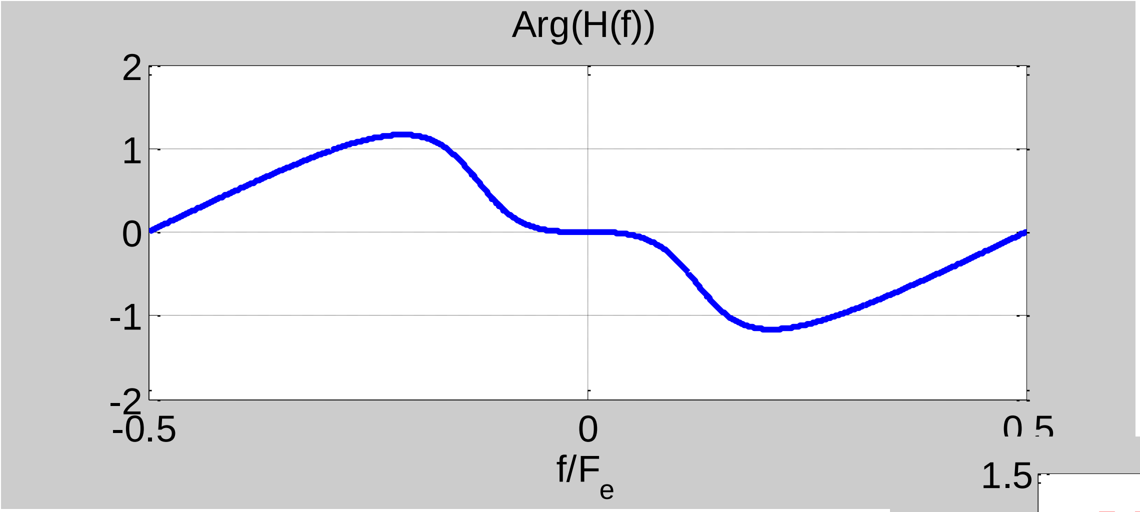
$$a_1 = -(p_1+p_2) = -1 \text{ et } a_2 = p_1 p_2 = 0{,}5$$

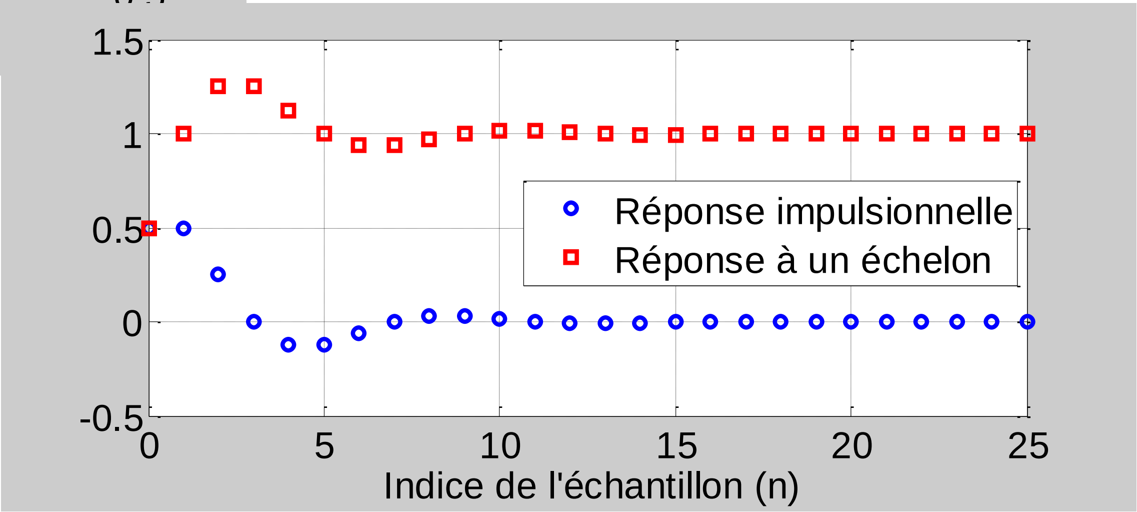
$$F_r = 0{,}115 F_e$$





Example 2 : pure recursive 2nd order filter (7/7)



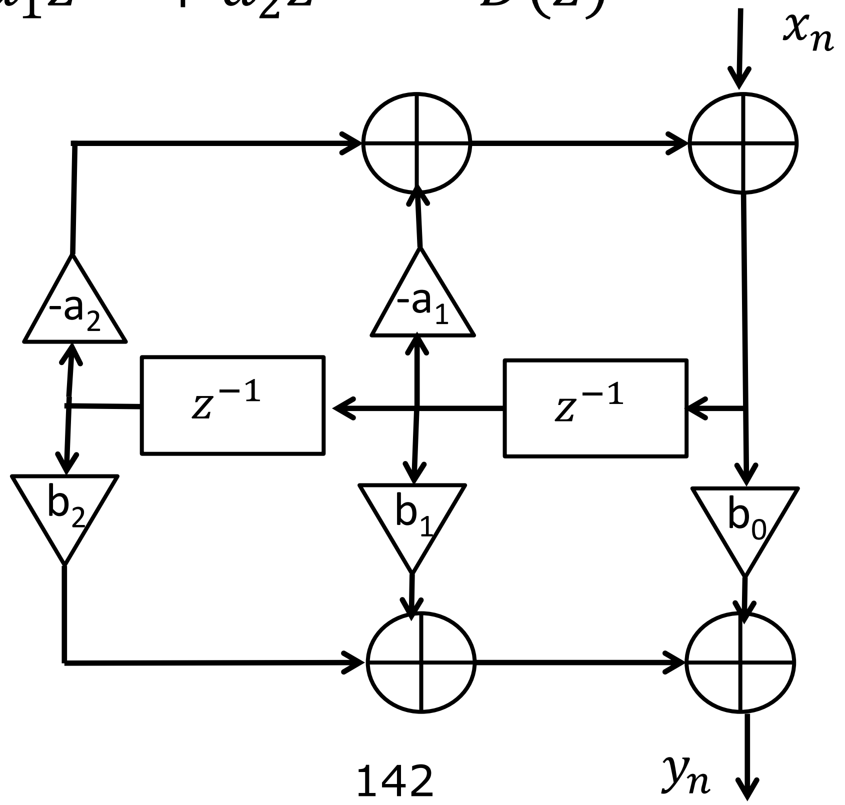


Example 3 : generalized recursive 2nd order filter (1/4)

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{N(z)}{D(z)}$$

Difference equation

$$y_n = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} - a_1 y_{n-1} - a_2 y_{n-2}$$



Example 3 : generalized recursive 2nd order filter (2/4)

$$H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2} = b_0 \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)}$$

$a_1 = -(p_1 + p_2)$ and $a_2 = p_1 p_2$ $\quad$ $b_1/b_0 = -(z_1 + z_2)$ and $b_2/b_0 = z_1 z_2$

This filter has two zeros z_1 and z_2, and two poles p_1 and p_2.

$z_1 = \frac{-b_1 + \sqrt{\Delta_\mathrm{z}}}{2b_0}$ and $z_2 = \frac{-b_1 - \sqrt{\Delta_\mathrm{z}}}{2b_0}$ with $\Delta_z = {b_1}^2 - 4b_0 b_2$

$p_1 = \frac{-a_1 + \sqrt{\Delta_p}}{2}$ and $p_2 = \frac{-a_1 - \sqrt{\Delta_p}}{2}$ with $\Delta_p = {a_1}^2 - 4a_2$

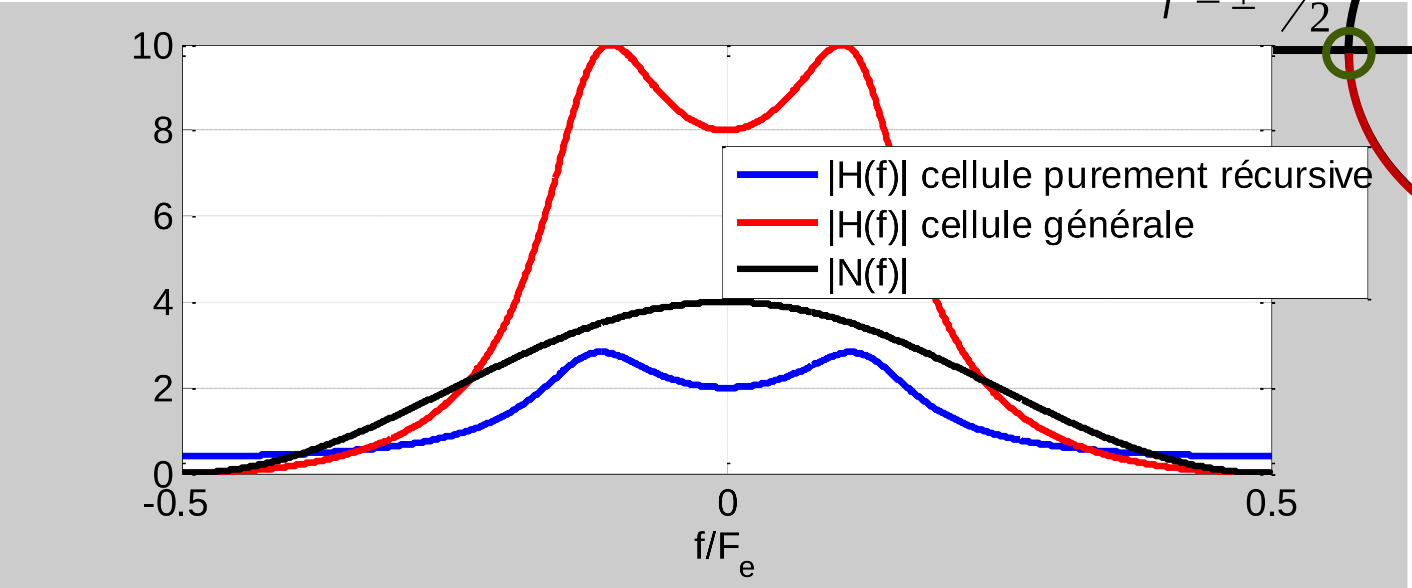
Four possible cases are distinguished by the signs of Δ_z and Δ_p.

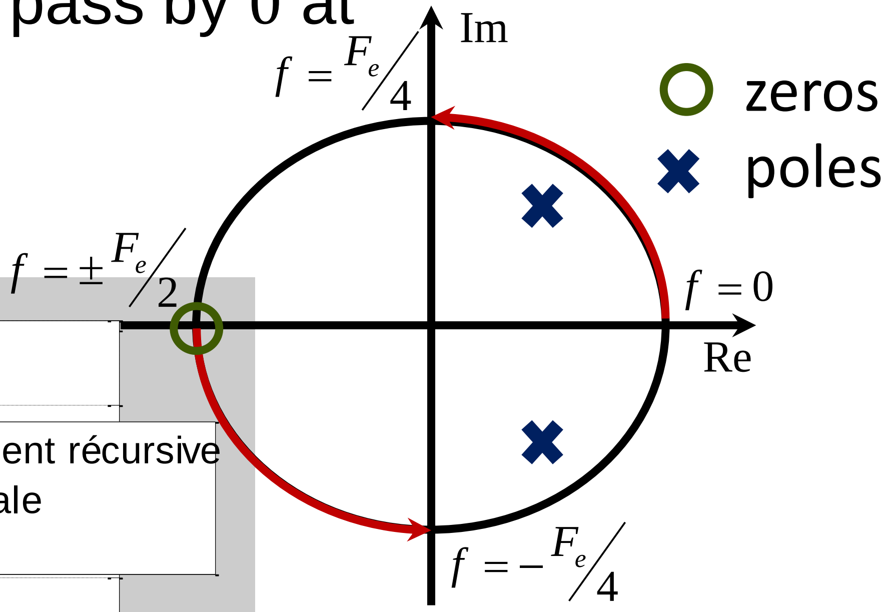
Example 3 : generalized recursive 2nd order filter (3/4)

Keeping the same poles, we force H(f) to pass by 0 at $f = \pm F_e/2$

$$z_1 = z_2 = -1$$

$$b_1/b_0 = 2 \quad \text{and} \quad b_2/b_0 = 1$$



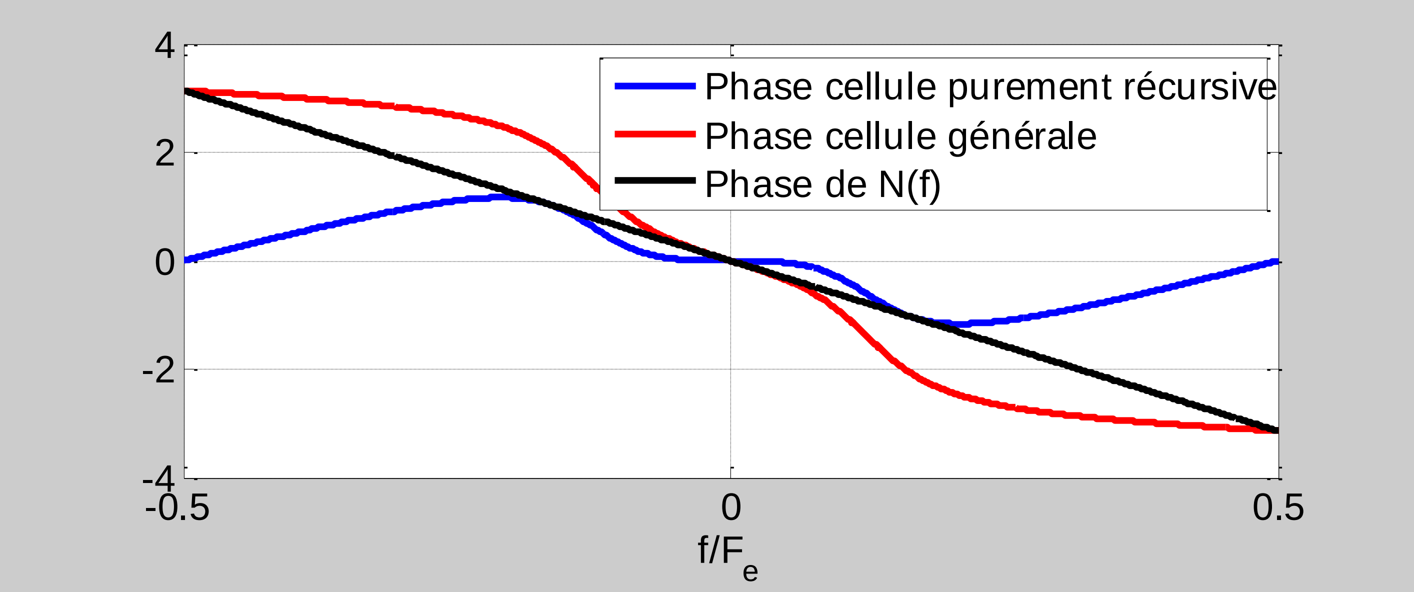


Example 3 : generalized recursive 2nd order filter (4/4)

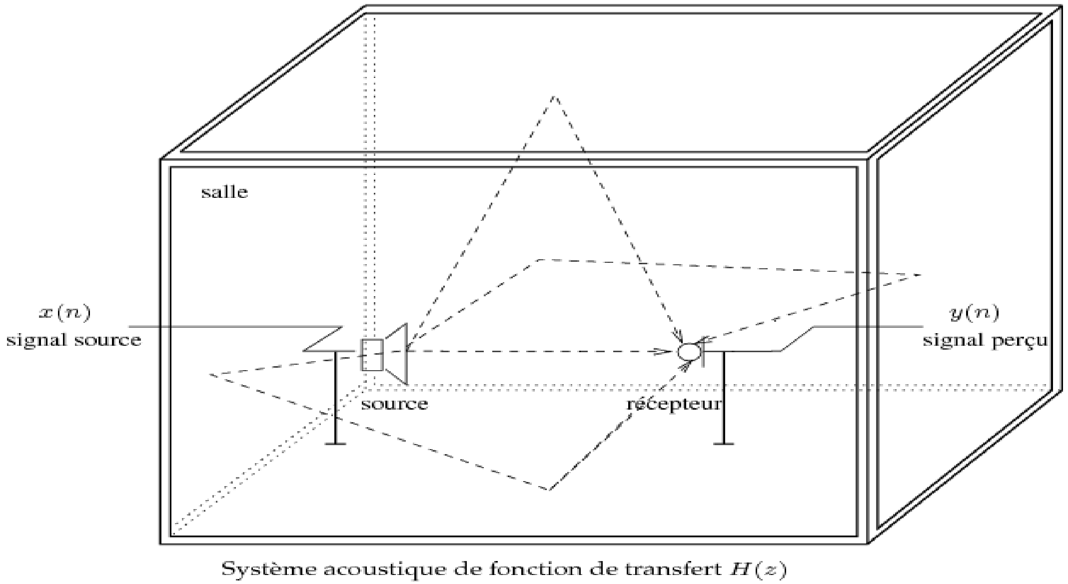
$$H(z) = \frac{N(z)}{D(z)} = \frac{(1+z^{-1})^2}{(1-p_1z^{-1})(1-p_2z^{-1})}$$

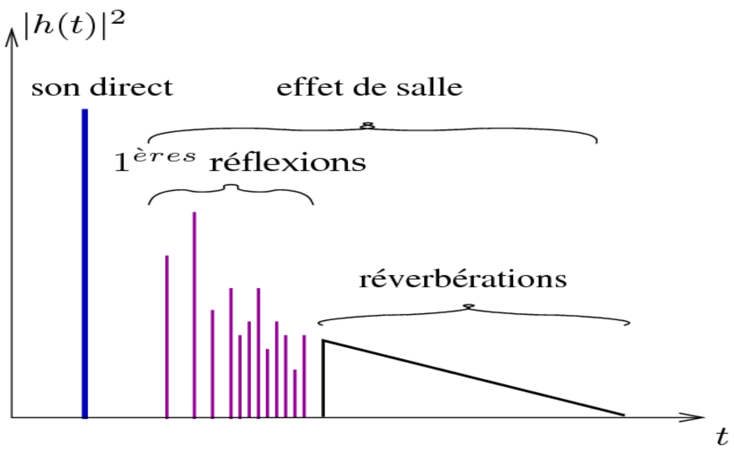
$$Arg(H(f)) = Arg(N(f)) - Arg(D(f))$$

$$Arg(N(f)) = -2\pi f / F_e$$



Application : Notch filter
Context





Reflexions

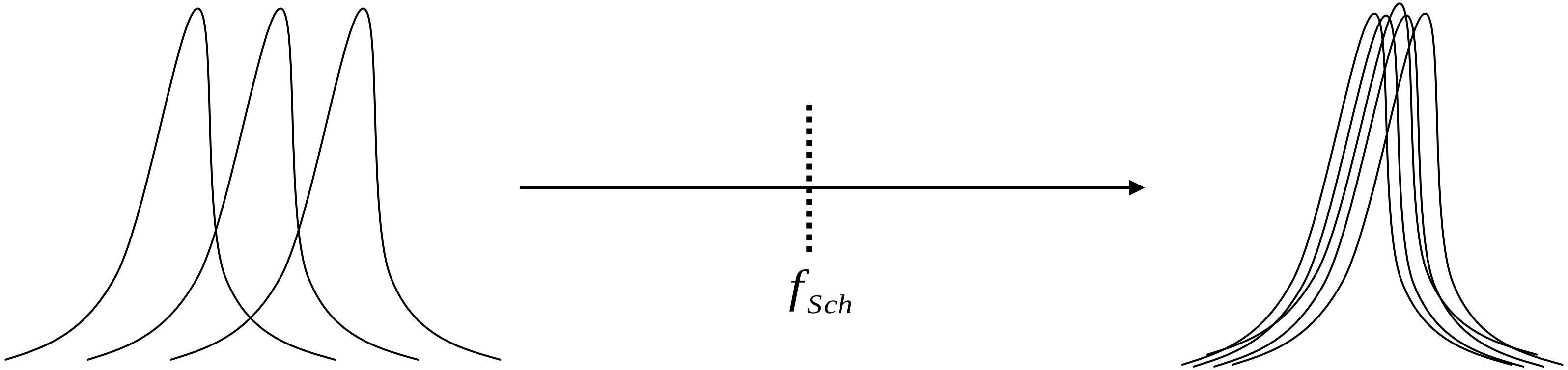
⇨ Standing waves

Perception problems

Application : Notch filter
Modal density

Nb of modes over *[0,f]* : $N_f \approx \frac{8\pi}{3} V (\frac{f}{c})^3$

Schroder frequency : $f_{Sch} \approx 2000 \sqrt{\frac{T_r}{V}}$

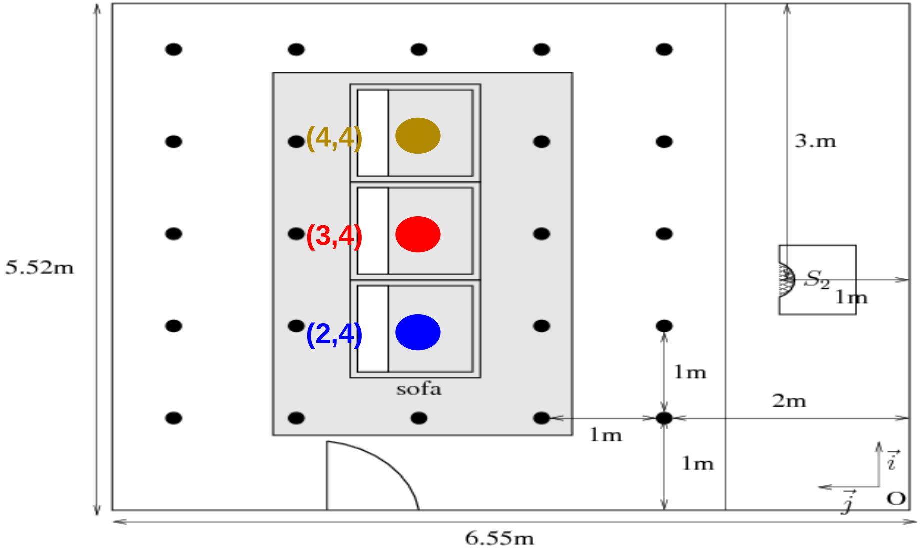


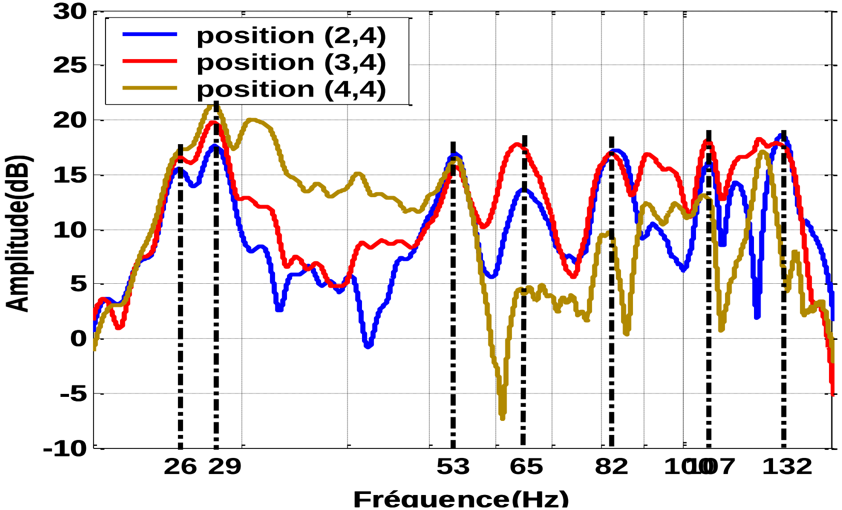
LF → localized modes HF → Diffuse wave field

$\Rightarrow$ Equalization for low frequencies: $f < f_{Sch}$

Application : Notch filter
Experimental protocol

The limitations of conventional inversion techniques, plus a specific room acoustics : acoustic channel variability with the position for source - receiver in the room





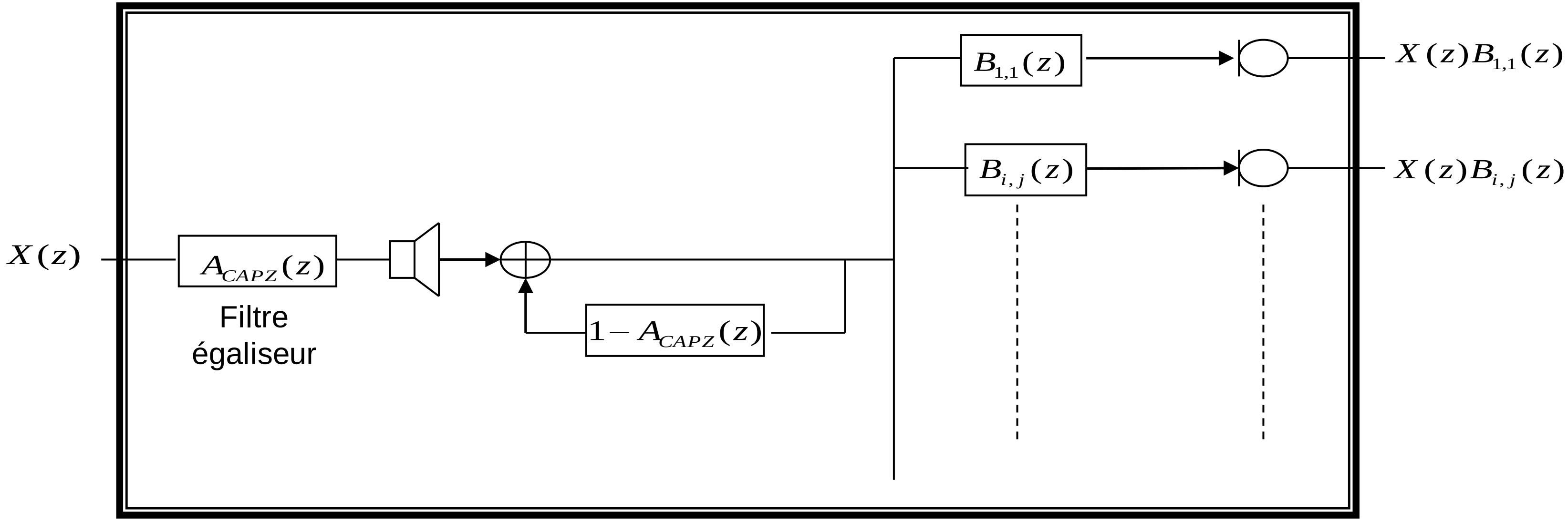
Application : Notch filter

Room Transfer Function modelling

Recursive modelling of Room Transfer Functions (RTF)

$$\hat{H}(r_{i,j},z)=\frac{Cz^{-Q_1}\prod_{m=1}^{Q_2}(1-q_m(r_{i,j})z^{-1})}{\prod_{n=1}^{P}(1-p_n(r_{i,j})z^{-1})}=\frac{\sum_{m=0}^{Q}b_m(r_{i,j})z^{-i}}{1+\sum_{n=1}^{P}a_n z^{-i}}=\frac{B_{i,j}(z)}{A_{CAPZ}(z)}$$

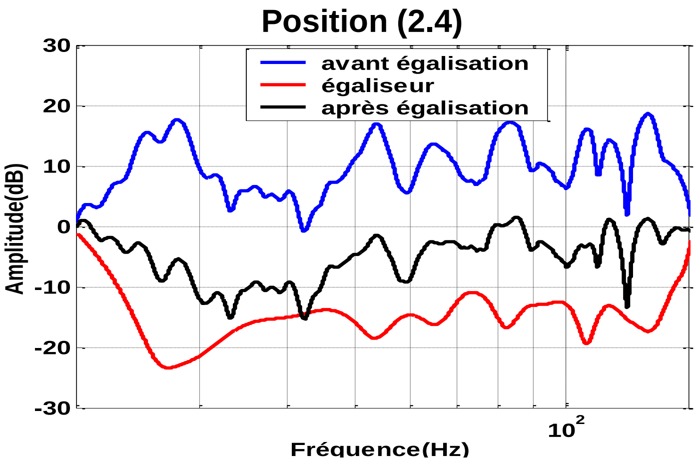
Poles model the room resonances and are common to all positions

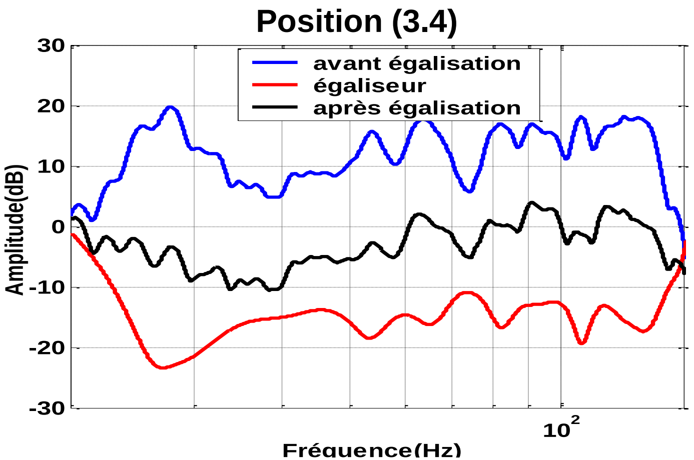


Application : Notch filter

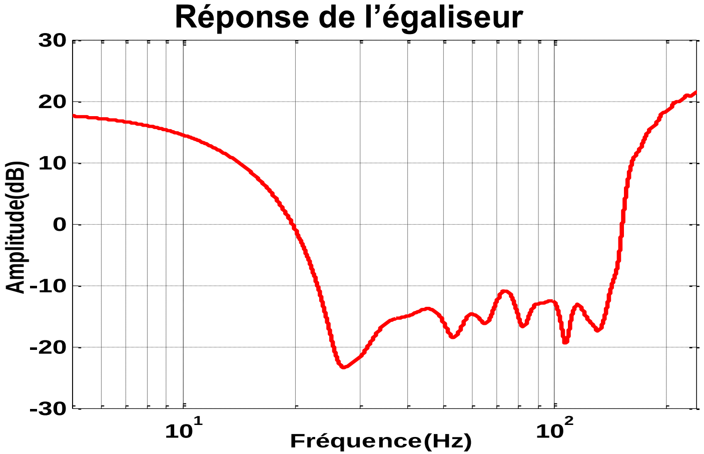
FIR equalizer implementation

P = 36





Out of band gain



→ NL distortion risk

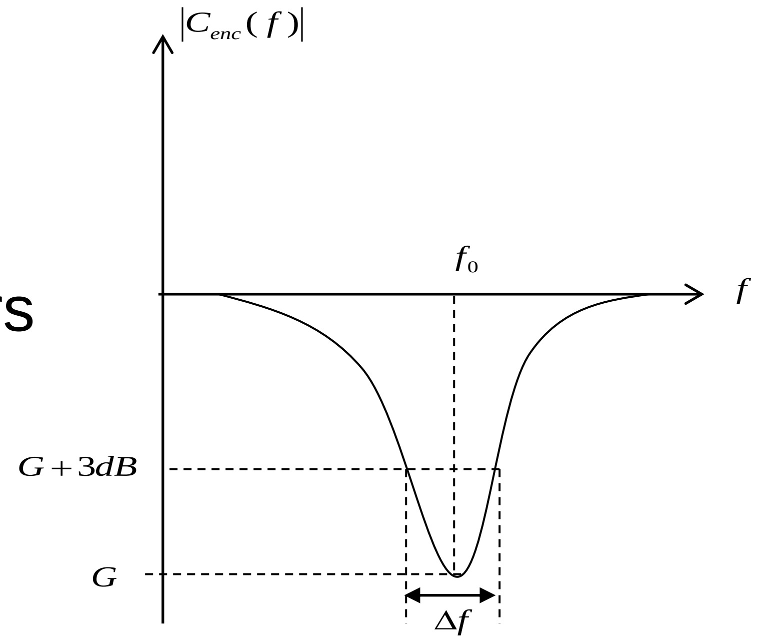
Application : Notch filter

IIR notch filter design (1/3)

Approach:

1. Compute average RTF
2. Select the dominant modes
3. Design cascaded second order IIR filters

$$C_{enc}(z) = \frac{b_0(G,\Delta f) - b_1(G,\omega_0)z^{-1} + b_2(G,\Delta f)z^{-2}}{1 - a_1(G,\Delta f)z^{-1} + a_2(G,\Delta f)z^{-2}}$$

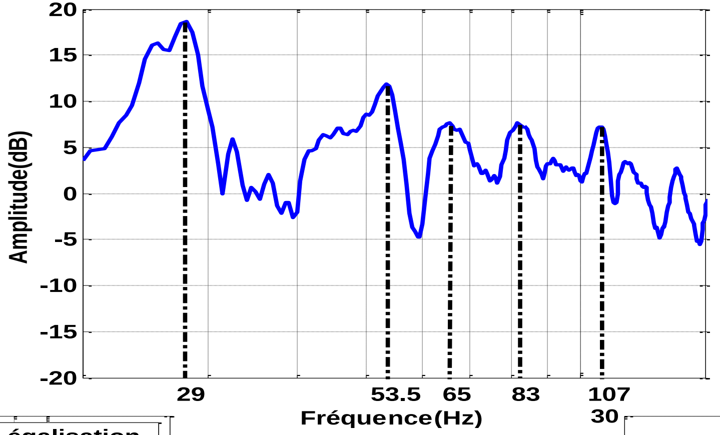


Advantages:

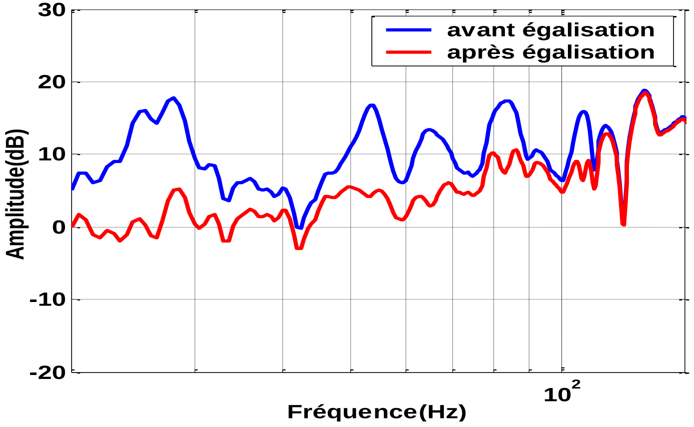
- Simple (reduced complexity)
- Selective correction of most problematic resonances

Application : Notch filter
IIR notch filter design (2/3)

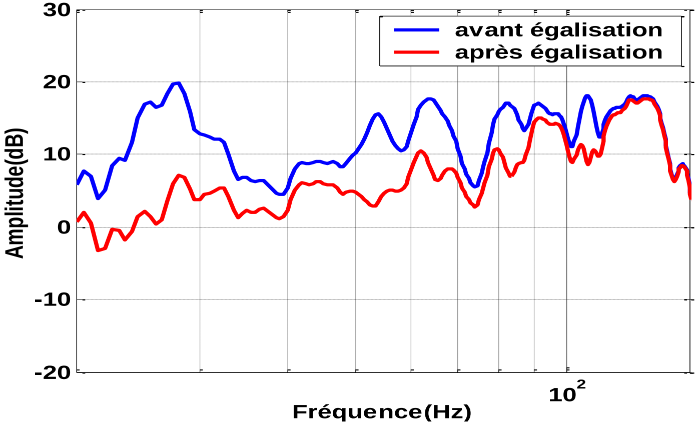
5 peaks



Position (2.4)



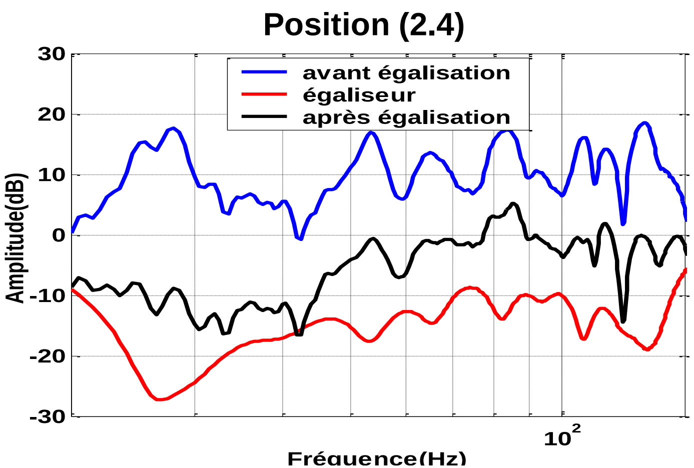
Position (3.4)

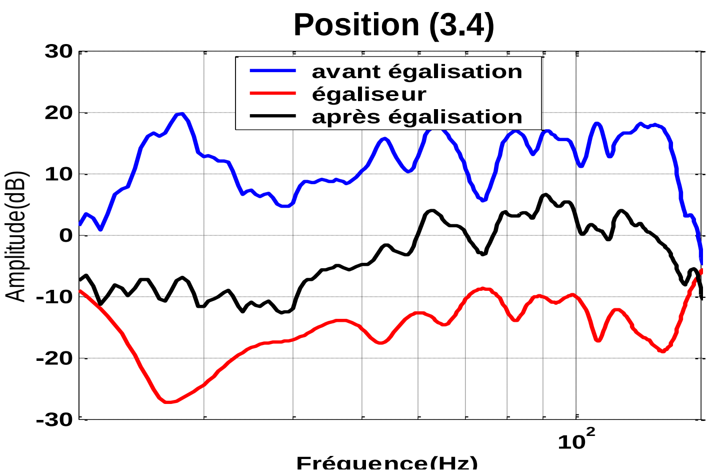


Application : Notch filter

IIR notch filter design (3/3)

11 peaks





Limitation :

- When peaks are close $\Rightarrow$ filters overlap

$\Rightarrow$ Need to ptimize parameters $(G,, f_{0,} \Delta f)$

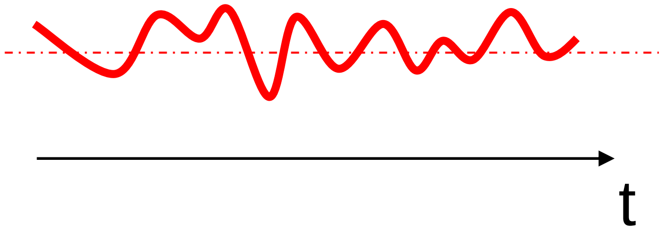
Outline

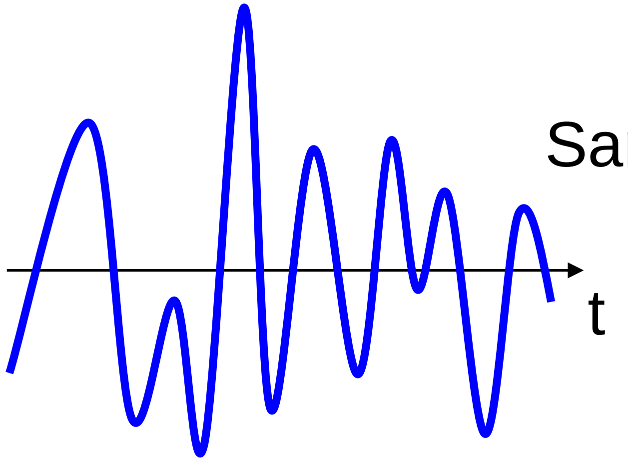
1. Data acquisition and analysis (2 lectures)

2. Digital data filtering (2 lectures)

3. **Random signal processing (1 lecture)**

Random signal and amplitude (1/2)





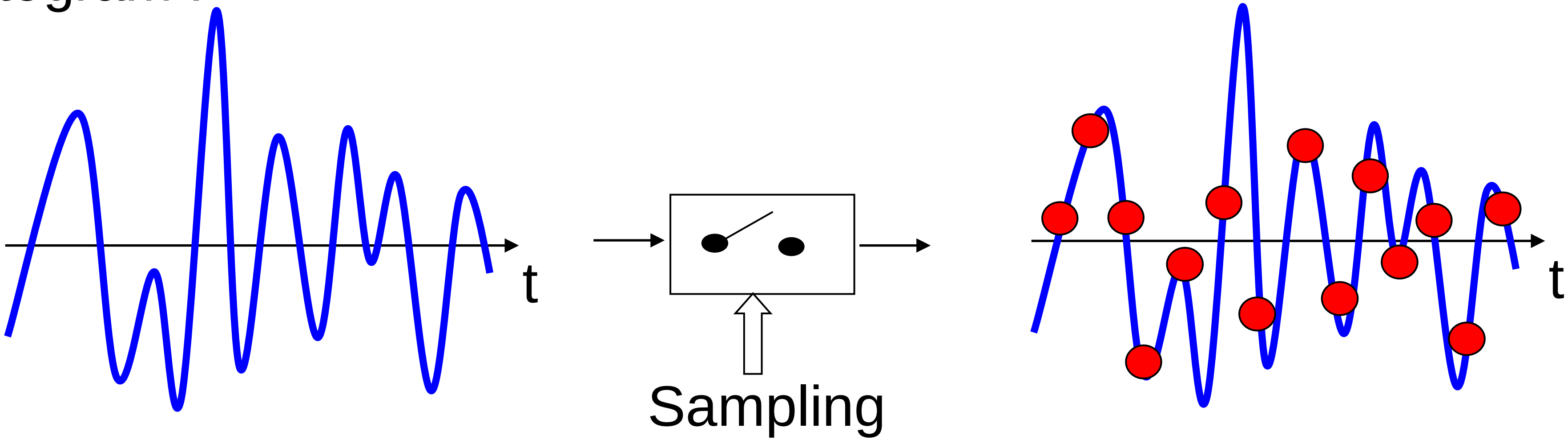
Same speed variation

The shapes of the signals are quite diffrent :

- for signal 1, the mean value is not equal to 0. The amplitudes are fluctuationg near to that mean value.
- for signal 2 the mean value is equal to 0. The amplitudes are taking important values arround that mean value.

Random signal and amplitude (2/2)

In practice the amplitude of a random signal Is characterized with an histogram :



Analog random signal
$X=x(t)$

Discrete random signal
$X=\{x_1, x_2\ x_3\ \ldots\ x_N\}$

Expectation of a Random Variable: E[X]

The expectation (mean or average) of a continuous random variable ***X*** is given by

$$m_X = E(X) = \int_{-\infty}^{+\infty} x f_X(x)dx$$

The expectation (mean or average) of a discrete random variable ***X*** is given by

$$m_X = E(X) = \sum_{x=-\infty}^{x=+\infty} x \Pr(X = x)$$

Discrete random process

For a discrete random process $\{x_k\}_{k\in Z}$

Resulting for example, from the sampling of a realization $x(t)$ of a continuous time random signal

1st order stationnarity : $m_k = E\{x_k\} = \text{constant} = m$

2nd order stationnarity : $E\{x_k x_l\}$ is function of $k-l$

$$r_{xx}(l) = E\{x_k x_{k-l}\} \ \ is\ even$$

Power of the signal : $\forall k : \mathrm{E}\{\mathrm{x}_\mathrm{k}^2\} \int_{x=-\infty}^{+\infty} x^2 f(x,k)dx = r_{xx}[0]$

Intercorrelation of two processess : $r_{xy}[n] = E\{x_k y_{k+n}\}$

White noise

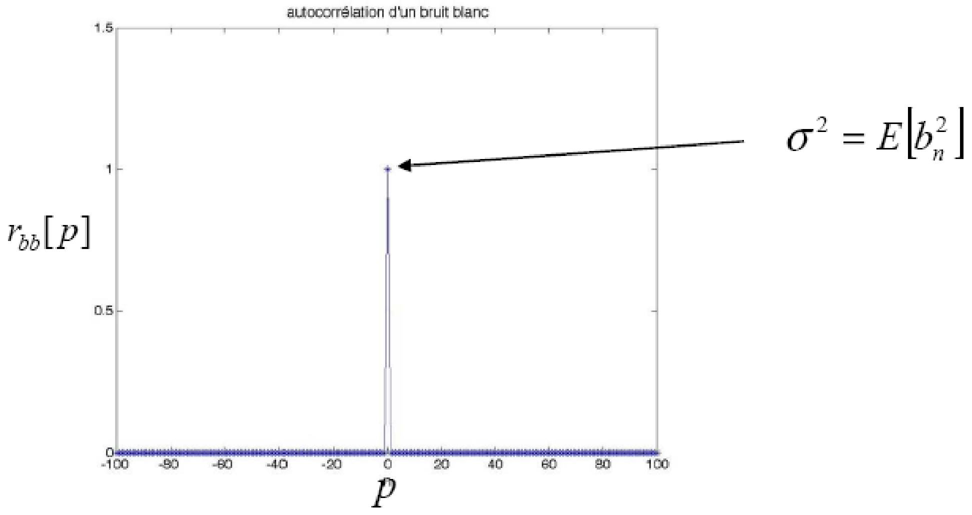
Modelling the measurment error $b_n = b(nT_e)$

Stationnary noise.

Zero mean : $E\{b_k\} = 0$

Uncorrelated samples :
$E\{b_n b_{n-p}\} = E\{b_n\}E\{b_{n-p}\} = 0$

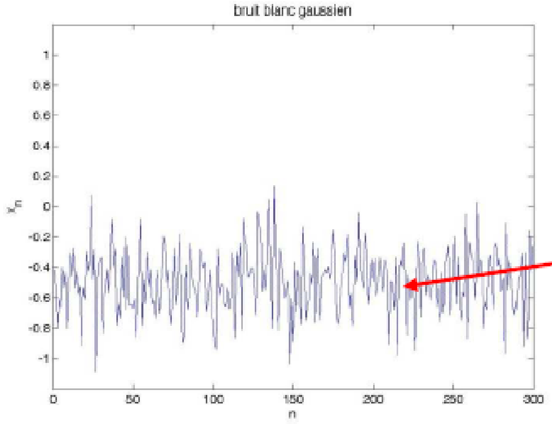
Autocorrelation : $r_{xx}(p) = E\{b_n b_{n-p}\} = \sigma^2\delta(p)$



Ergodism

A stationary random signal is ergodic if the statistical averages are equal to the temporal averages.

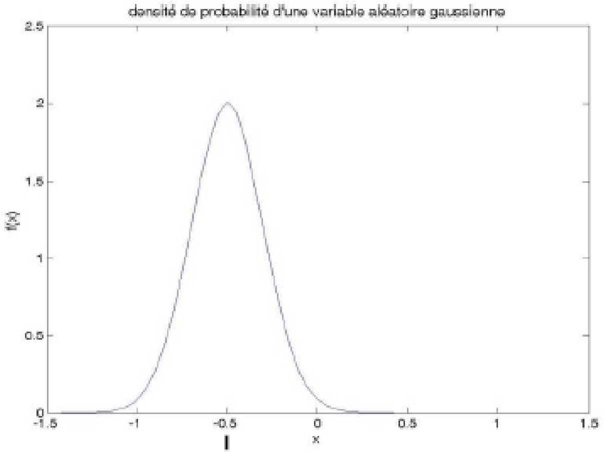
Let: $\{x_k\}$ be a random stationnary gaussian process



temporal average

$$\frac{1}{N+1}\sum_{n=0}^{N} x_n$$

$$f(x,t) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)$$



$m = -0.5$

statistical average $\int_{x=-\infty}^{+\infty} x f(x,t)\,dx$

$$\boxed{\lim_{N\to+\infty}\frac{1}{N+1}\sum_{n=0}^{N} x_n = \int_{x=-\infty}^{+\infty} x f(x,t) = m}$$

Estimators for a stationnary discret random process

Let: $\{x_k\}$ be a random stationnary gaussian process

Temporal average: $$\frac{1}{N+1}\sum_{n=0}^{N} x_n \approx E\{x_n\}$$

Power: $$\frac{1}{N+1}\sum_{n=0}^{N} x_n^2 \approx E\{x_n^2\}$$

Autocorrelation: $$\frac{1}{N+1}\sum_{n=p}^{N} x_n x_{n-p} \approx E\{x_n x_{n-p}\}$$

intercorrelation: $$\frac{1}{N+1}\sum_{n=p}^{N} x_n y_{n-p} \approx E\{x_n y_{n-p}\}$$

Temporal averages

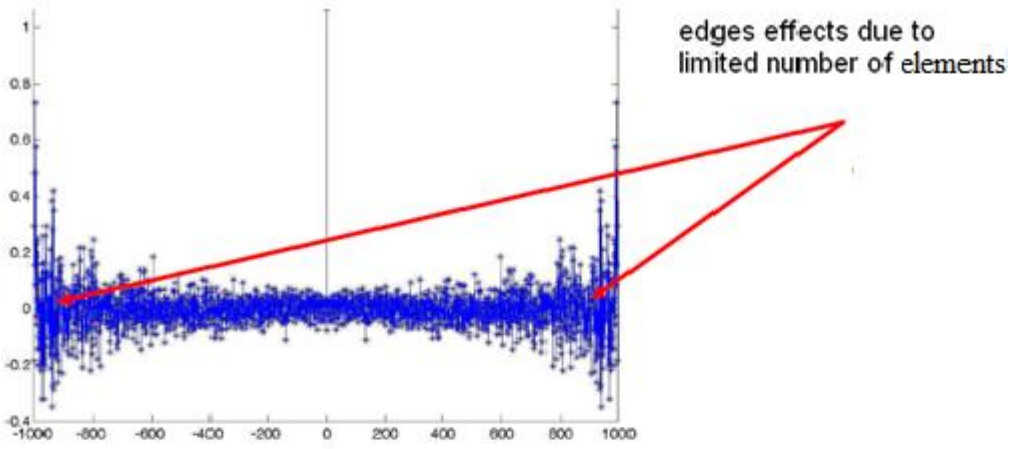
Statistical averages

Estimation of autocorrelation (1/3)

Unbiased estimator

$$p \geq 0: \ E\{x_n x_{n-p}\} \approx \frac{1}{\text{number of terms}} \sum_{n=p}^{N} x_n x_{n-p} = \frac{1}{\mathrm{N} - \mathrm{p} + 1} \sum_{n=p}^{N} x_n x_{n-p}$$

For a white noise



Estimation of autocorrelation (2/3)

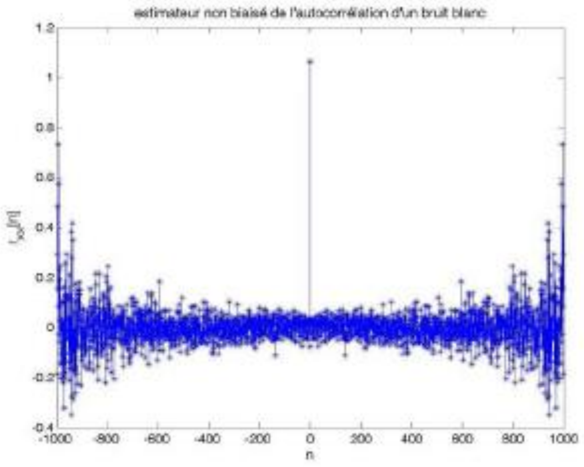
Better estimator (biased)

$$p \geq 0: \; E\{x_n x_{n-p}\} \approx \frac{1}{\mathrm{N} - p + 1} \sum_{n=p}^{N} x_n x_{n-p} \times \frac{N+1-p}{\mathrm{N}+1}$$

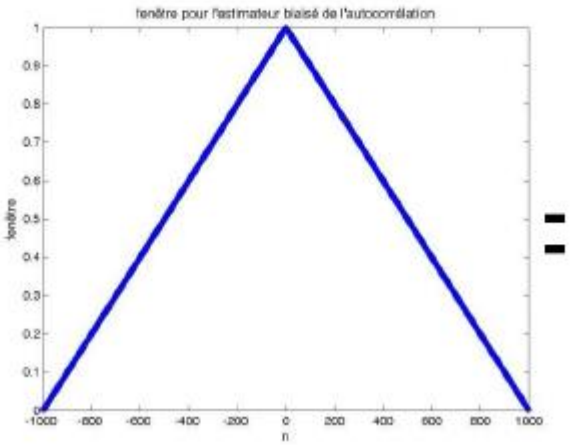
Weighting window

$$p \geq 0: \; E\{x_n x_{n-p}\} \approx \frac{1}{\mathrm{N}+1} \sum_{n=p}^{N} x_n x_{n-p}$$

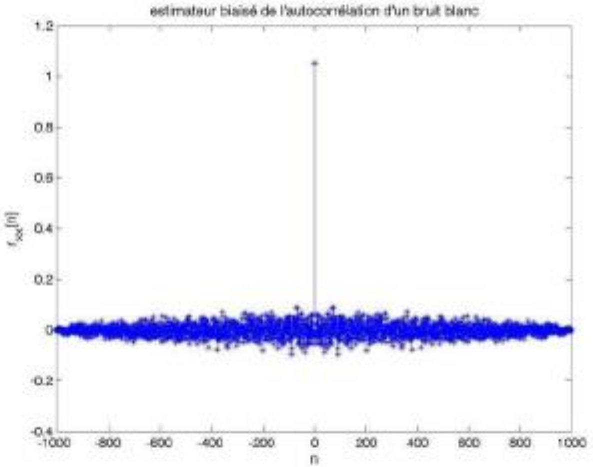
For a white noise



$\times$



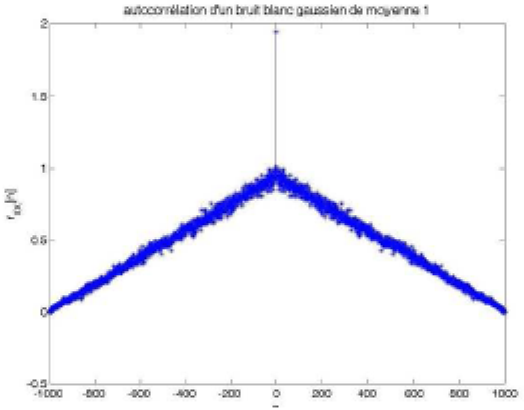
$=$



Estimation of autocorrelation (3/3)

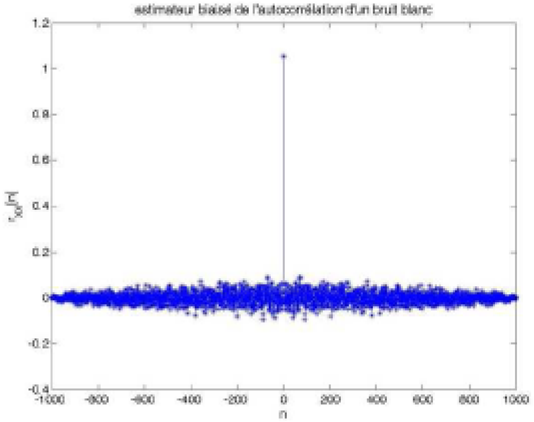
Let: $\{x_k\}$ be an uncorrelated random process with average m

$$p \neq 0:$$
$$E\{x_n x_{n-p}\} = E\{x_n\}E\{x_{n-p}\}$$
$$E\{x_n x_{n-p}\} = m^2 \neq 0$$



If we want a zero mean process, we study $\{x_k - m\}$

$$p \neq 0$$
$$E\{(x_n - m)(x_{n-p} - m)\} = E\{x_n\}E\{x_{n-p}\} - m.E\{x_n\} - mE\{x_{n-p}\} + m^2$$
$$E\{(x_n - m)(x_{n-p} - m)\} = 0$$

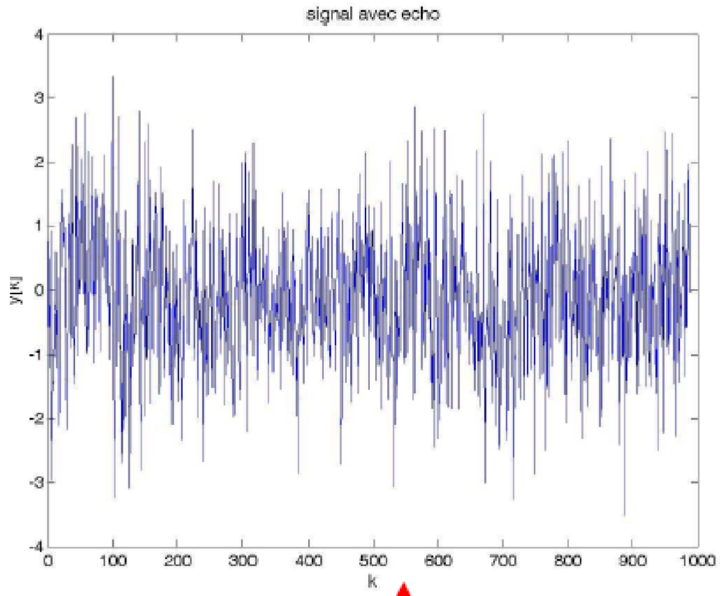


Example 1 : signal with echo

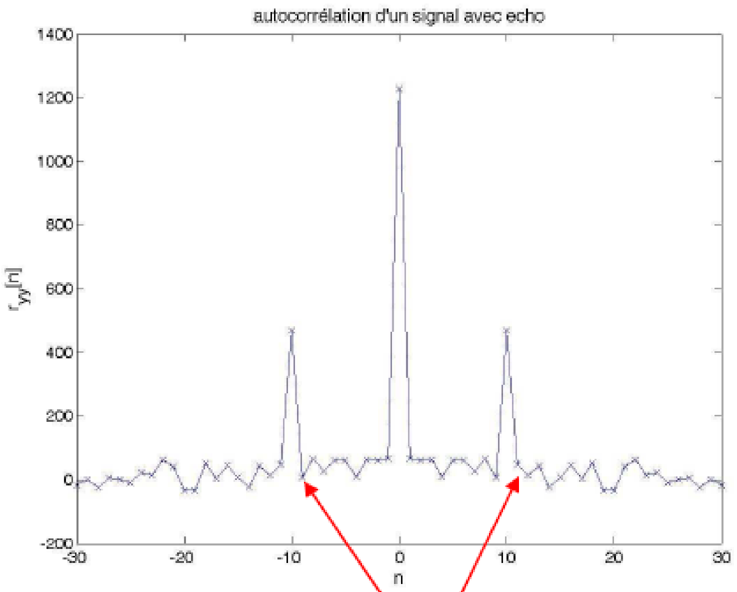
Let: $\{x_k\}$ be a white Gaussian noise

$$y_n = x_n + \alpha x_{n-p}$$

y_k



$r_{yy}[n]$

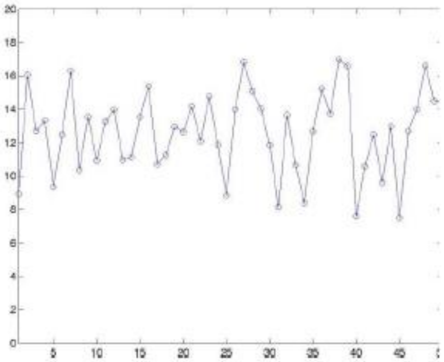


We can't deduce any information from the time domain signal

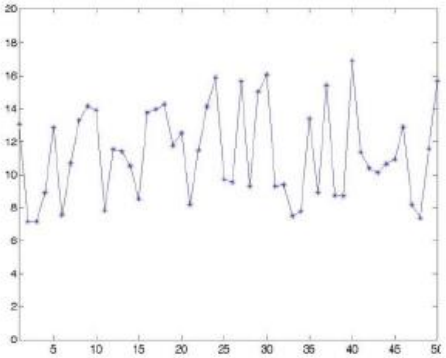
The secondary peaks indicate the presence of echos

Example 2 : students marks

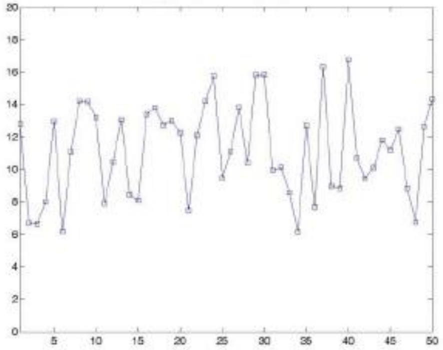
Durand:



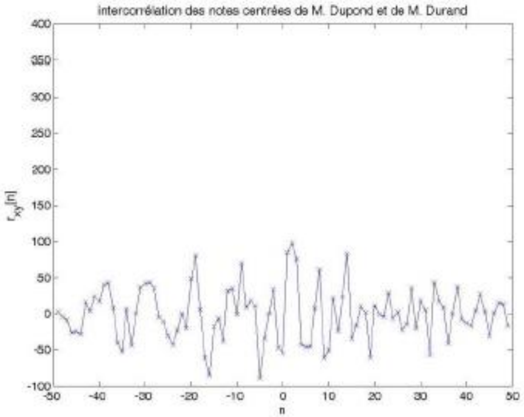
Dupond:



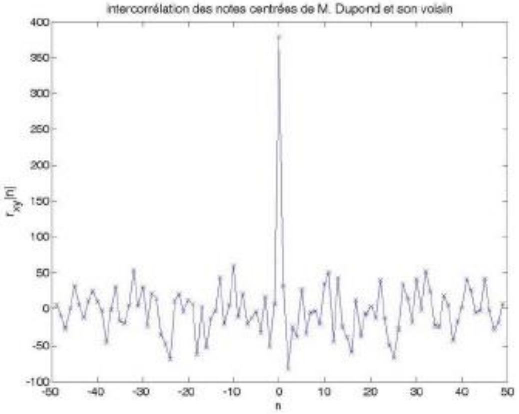
Neighbour of Dupond



intercorrelation Dupond/Durand:



intercorrelation Dupond/ Neighbour Dupond



Spectral representation

Case of deterministic signal (1)

Impulse response $\{h_k\}$

Frequency response $= \mathrm{TF}\{\, h_k \,\}\,(f\,)$

$$e^{j2\pi fkT_e} \longrightarrow \boxed{h_k} \longrightarrow H(f)e^{j2\pi fkT_e}$$

$$X(f)e^{j2\pi fkT_e} \longrightarrow \boxed{h_k} \longrightarrow X(f)H(f)e^{j2\pi fkT_e}$$

$$x_k = \int_{f=0}^{f_e} X(f)e^{j2\pi fkT_e} \longrightarrow \boxed{h_k} \longrightarrow y_k = \int_{f=0}^{f_e} X(f)H(f)e^{j2\pi fkT_e}$$

Spectra of the input signal

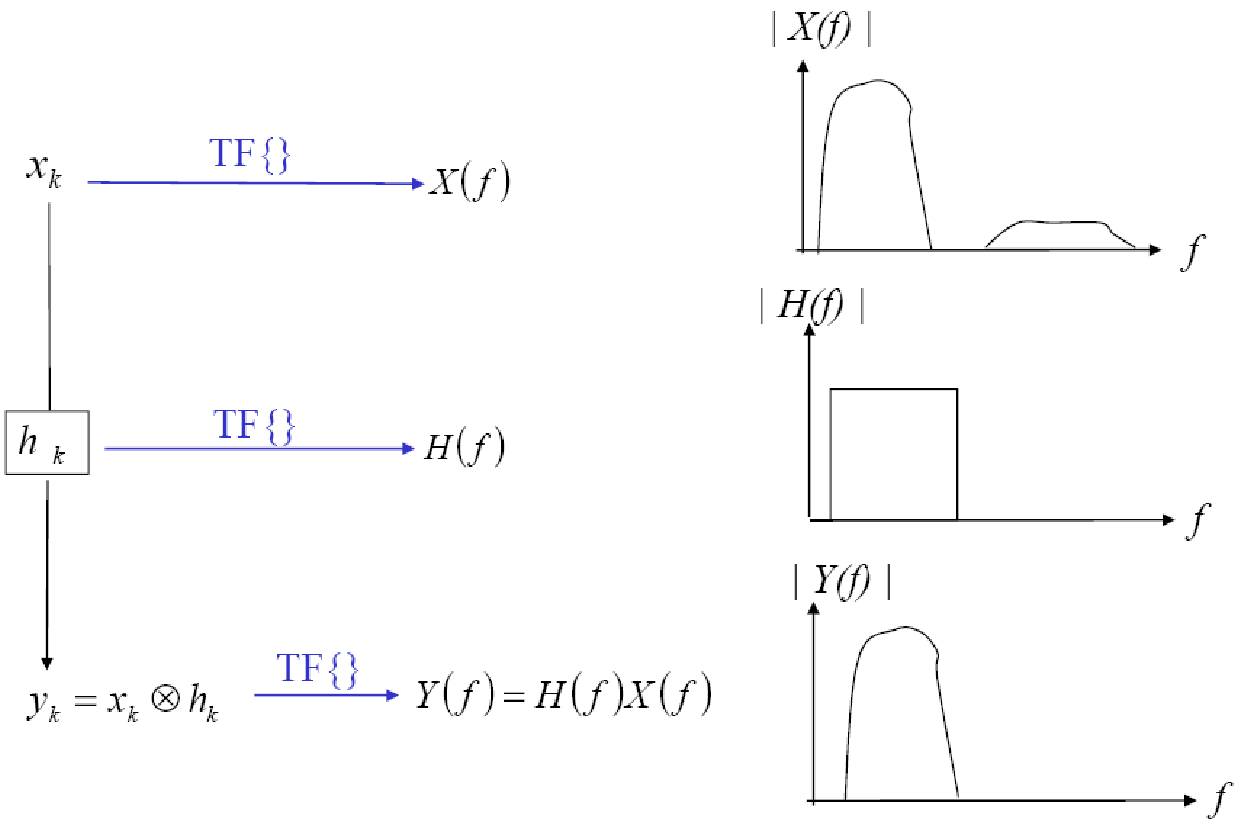
X(f)

Spectra of the output signal

Y(f)=H(f).X(f)

Spectral representation

Case of deterministic signal (2)



Spectral representation

Case of discret random process (1)

$$X(f)=\sum_k x_k e^{-j2\pi f k T_e}$$

Random spectra

Random process

Power Spectral Density (PSD) $P(f)=E\left\{\|X(f)\|^2\right\}$

We can demonstrate that

$$r_{xx}[n]=\int_{f=0}^{f_e} \underbrace{E\left\{\|X(f)\|^2\right\}}_{P(f)} e^{j2\pi f n T_e}\,df$$

Autocorrelation $\underset{FT}{\overset{FT^{-1}}{\rightleftarrows}}$ $P(f)$

Spectral representation

Case of discret random process (2)

$$P(f) = T_e \sum_{n=-\infty}^{+\infty} r_{xx}[n] e^{-j2\pi f n T_e}$$

$$r_{xx}[n] = \int_{f=0}^{f_e} P(f)\, e^{j2\pi f n T_e} df$$

For n=0

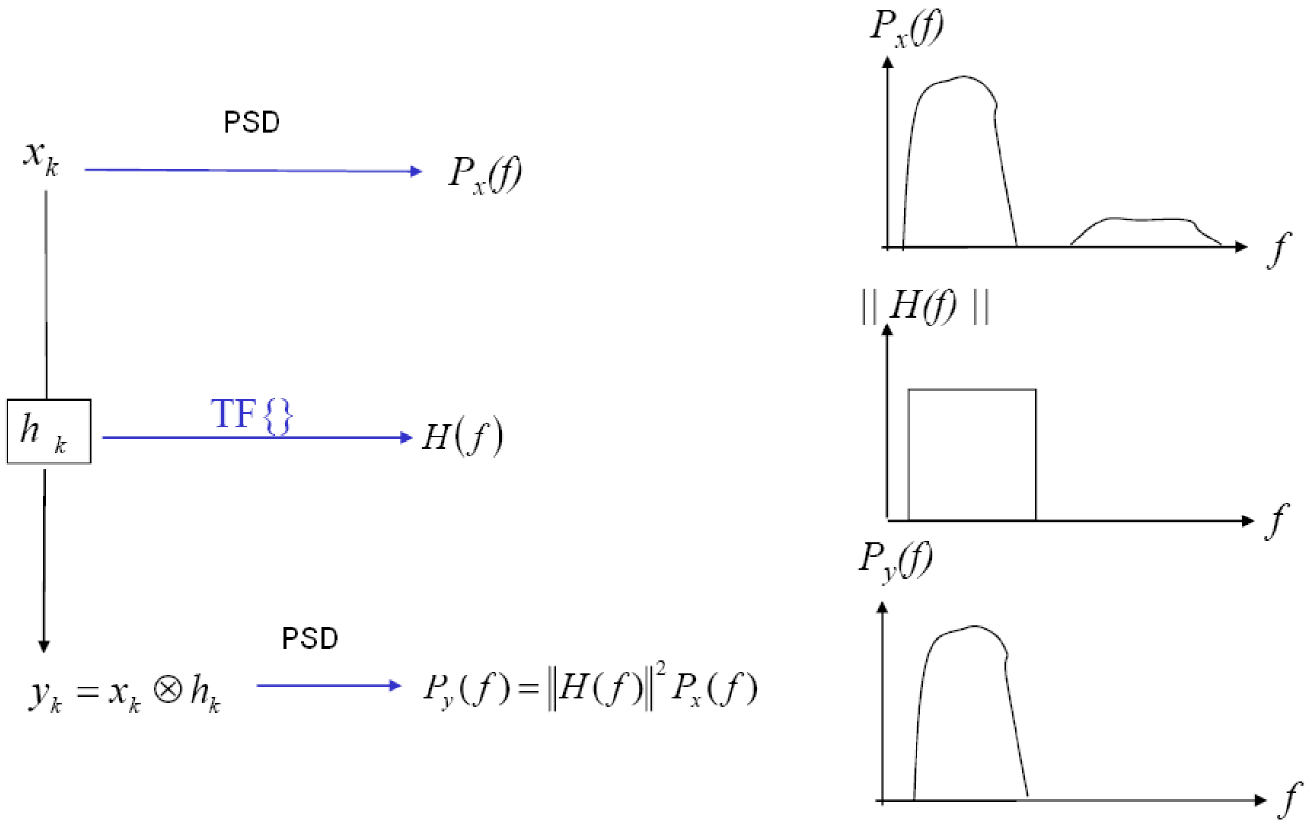
$$\underbrace{r_{xx}[0] = E\{x_k^2\}}_{\text{Power}} = \int_{f=0}^{f_e} P(f) df$$

Power PSD

Spectral representation

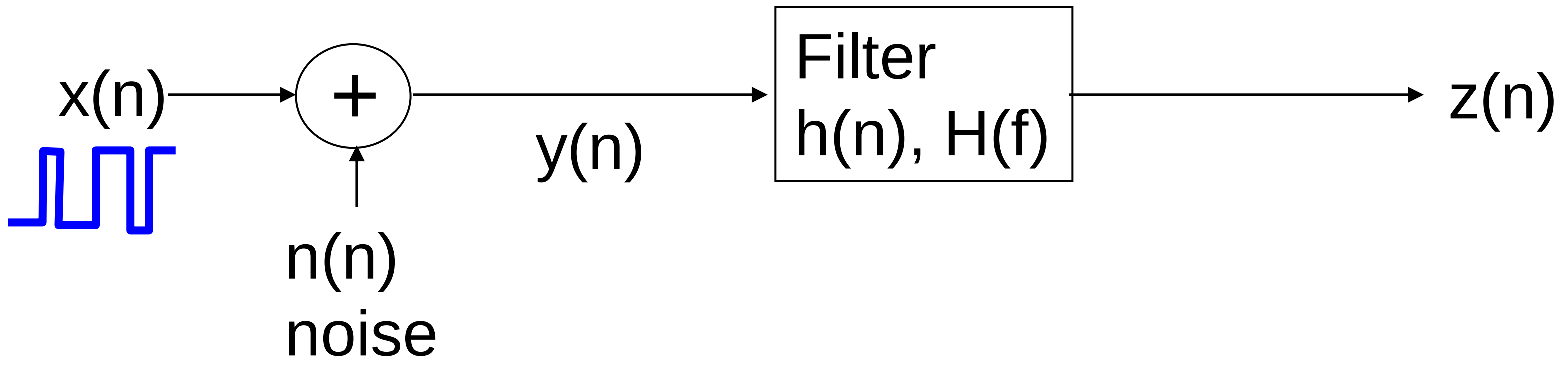
Case of discret random process (3)

The PSD at the output of an LTI system



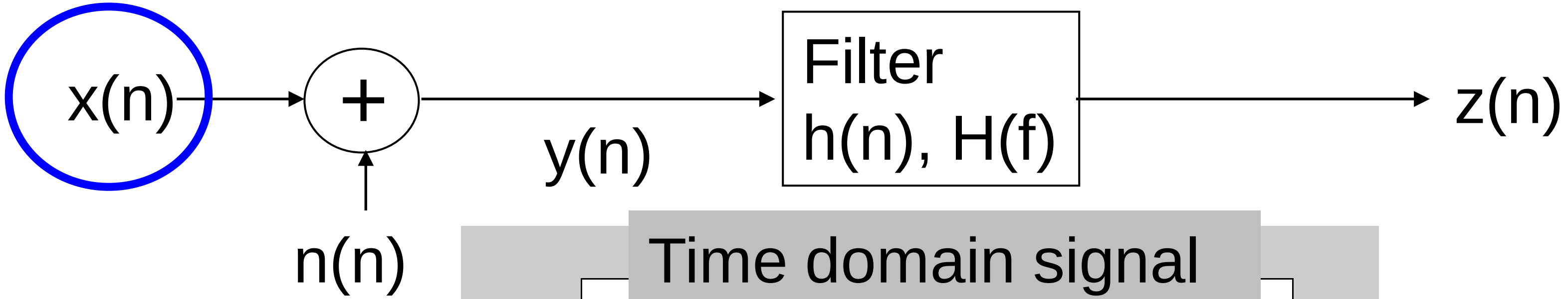
$$P_y(f) = E\{Y(f)Y^*(f)\} = E\{H(f)X(f)H^*(f)X^*(f)\} = \|H(f)\|^2 E\{X(f)X^*(f)\}$$

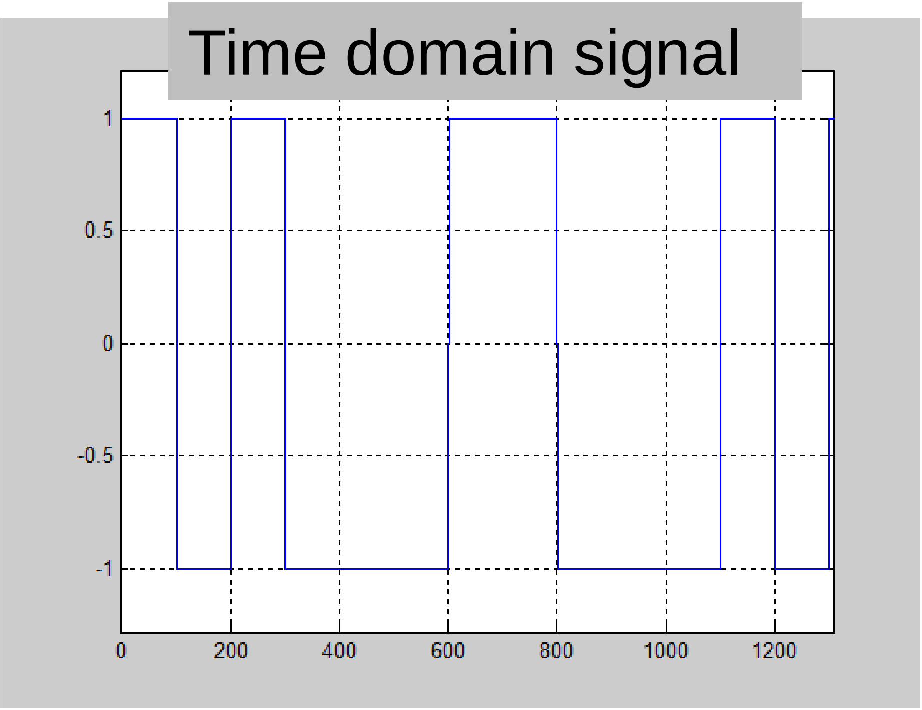
Spectral analysis with Matlab



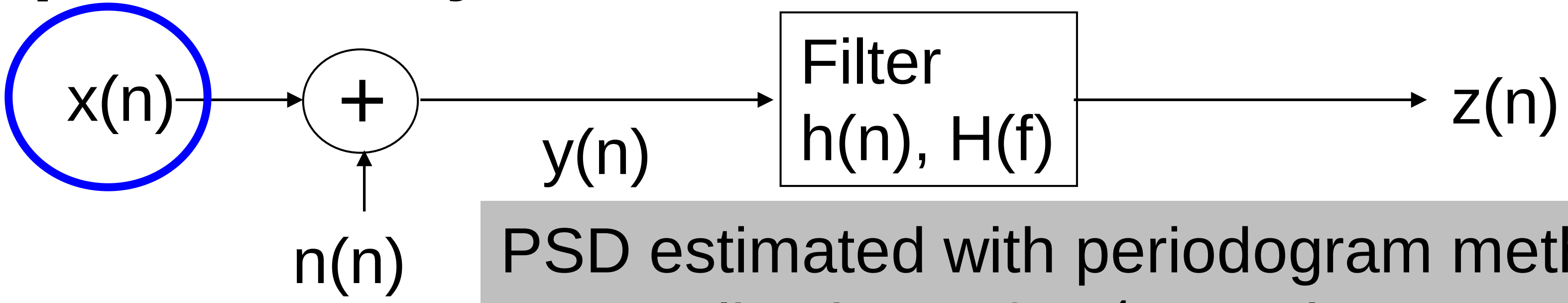
- ✓ 1 sample = 25µs (Fe=40 kHz)
- ✓ 10000 transmitted samples
- ✓ 1 sample = 100 bits
- ✓ Signal with 10^6 transmitted bits
- ✓ $Px=1V^2$
- ✓ $Pn=4V^2$

Spectral analysis with Matlab

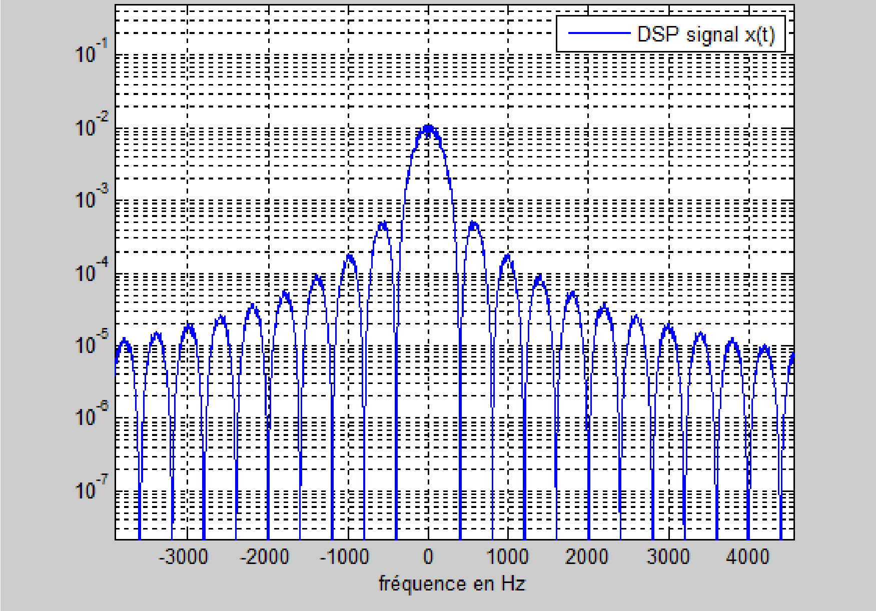




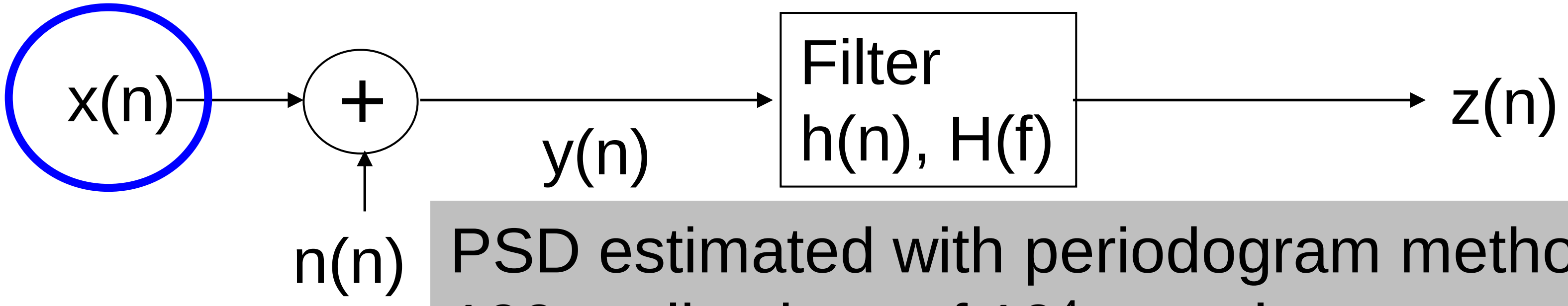
Spectral analysis with Matlab



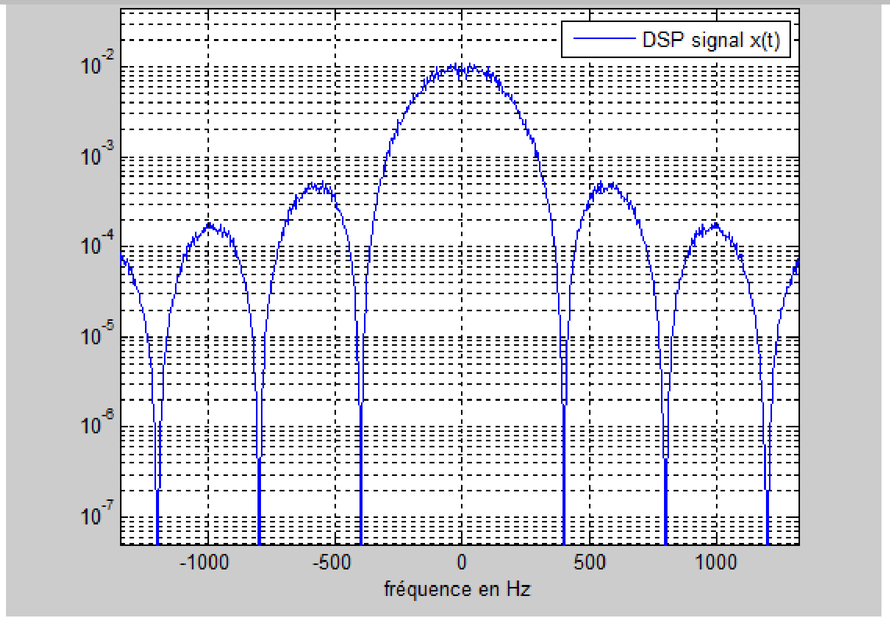
PSD estimated with periodogram method
100 realizations of 10^4 samples



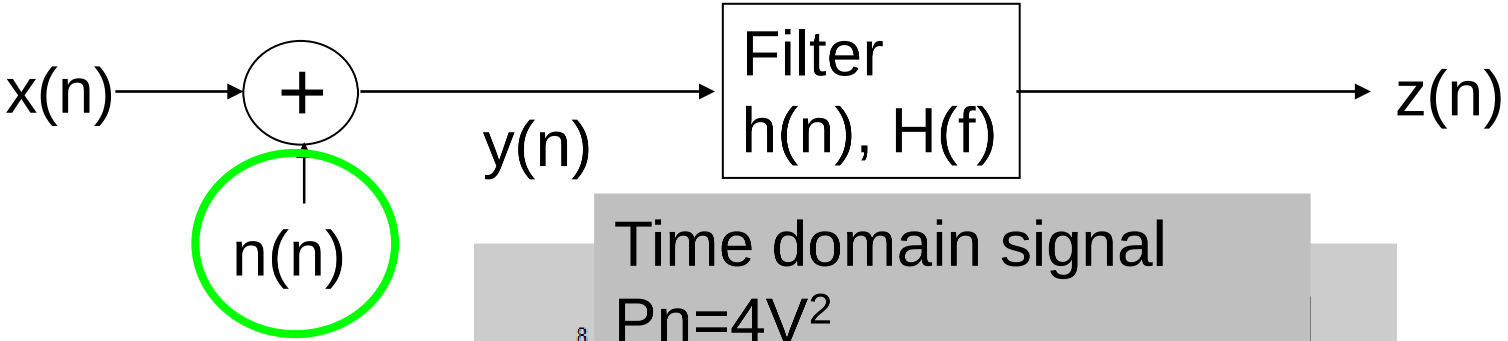
Spectral analysis with Matlab

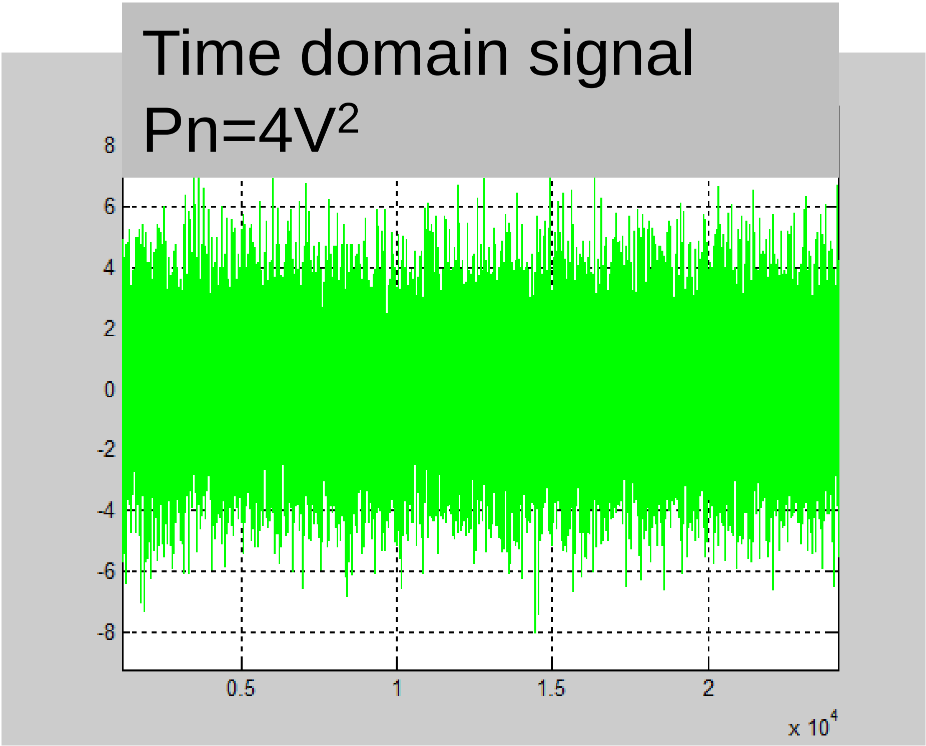


PSD estimated with periodogram method
100 realizations of 10^4 samples

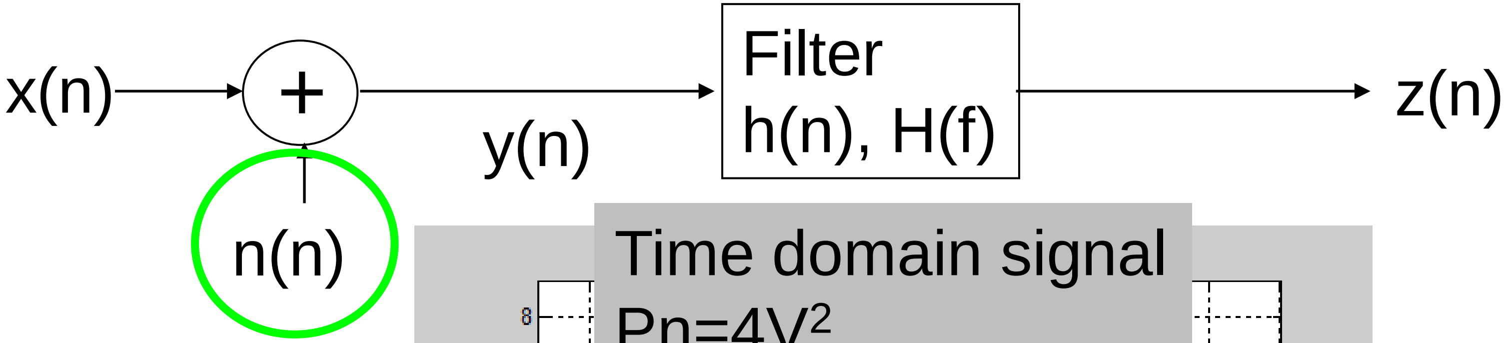


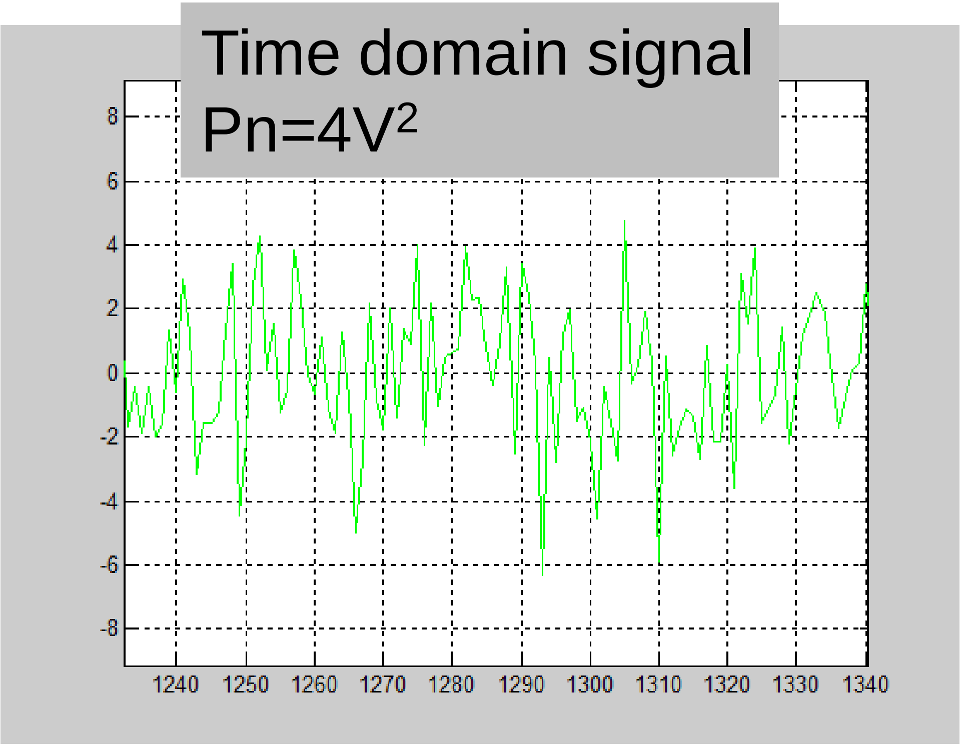
Spectral analysis with Matlab



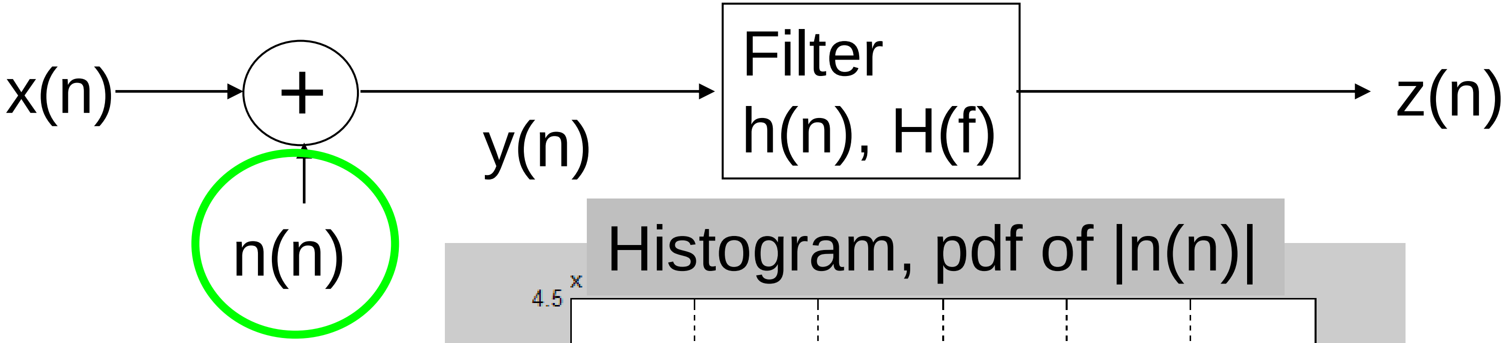


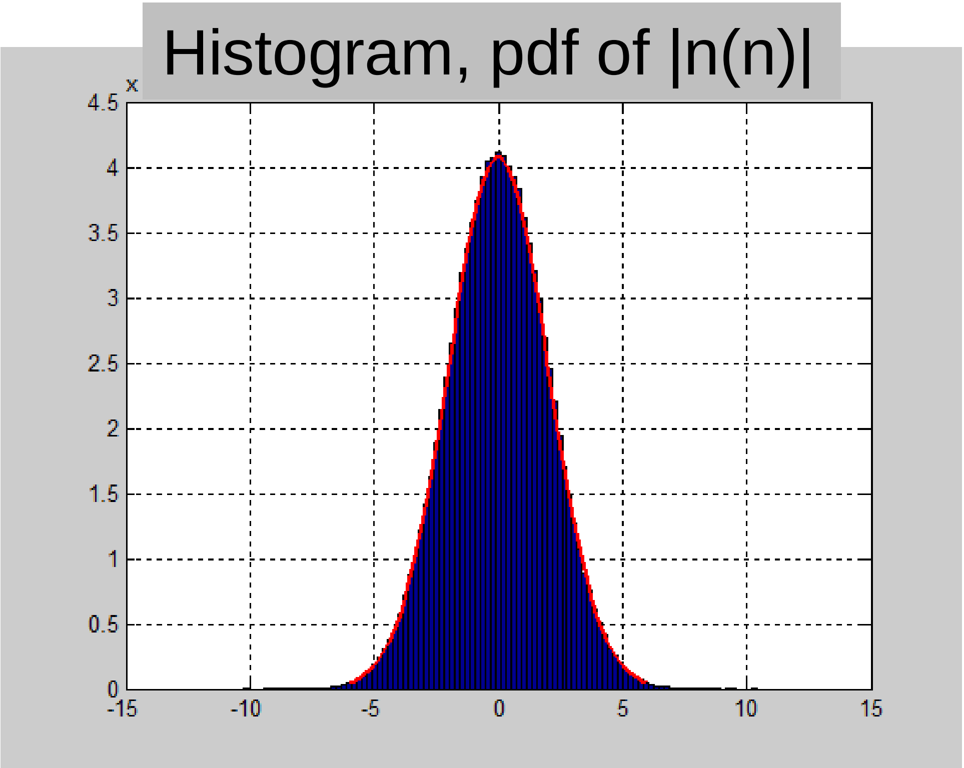
Spectral analysis with Matlab



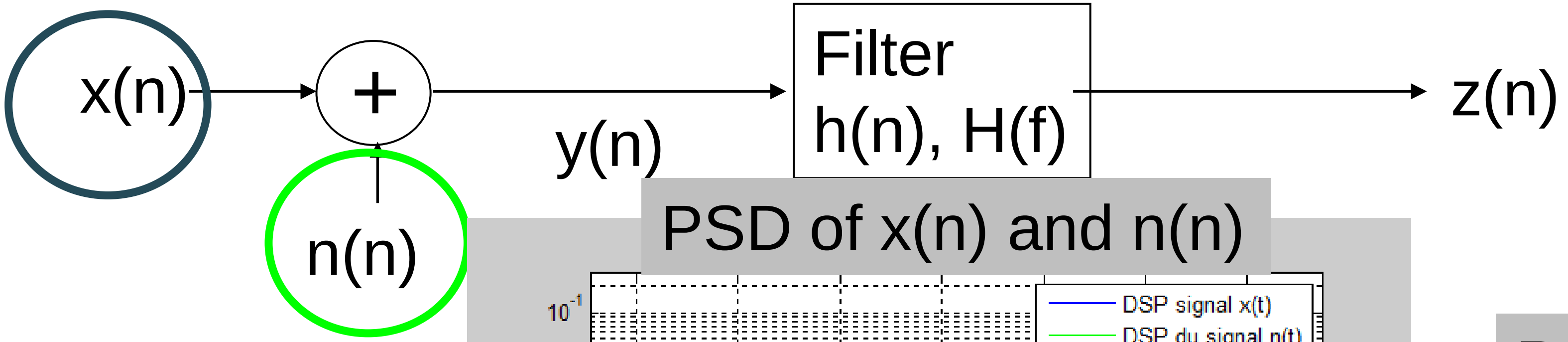


Spectral analysis with Matlab

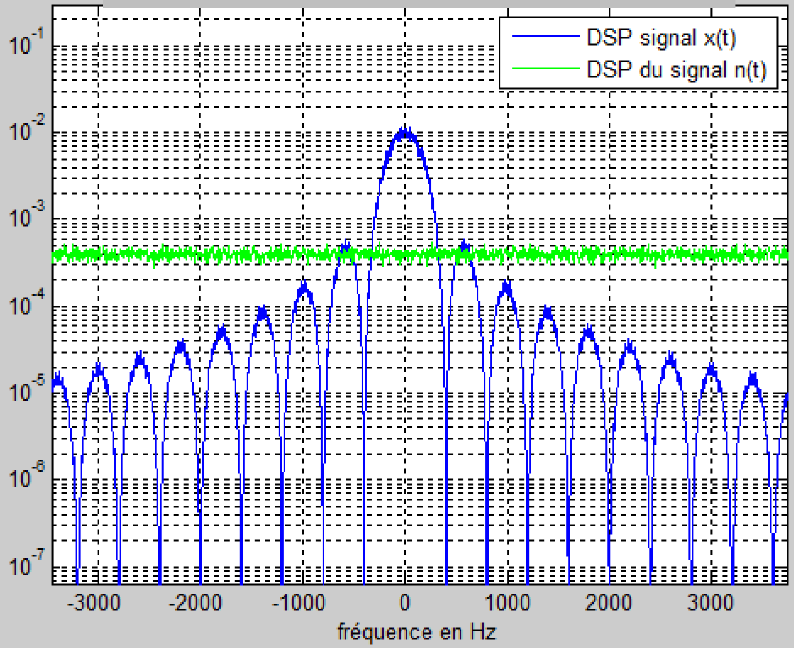




Spectral analysis with Matlab



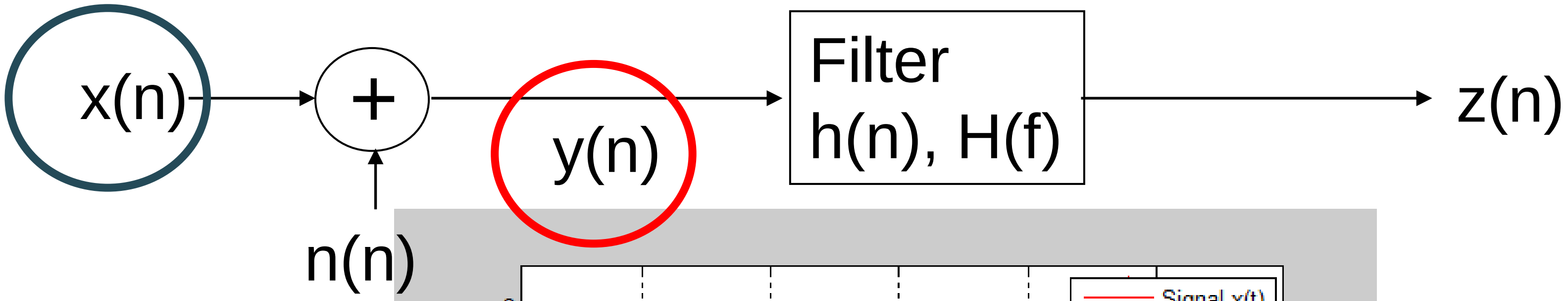
PSD of x(n) and n(n)

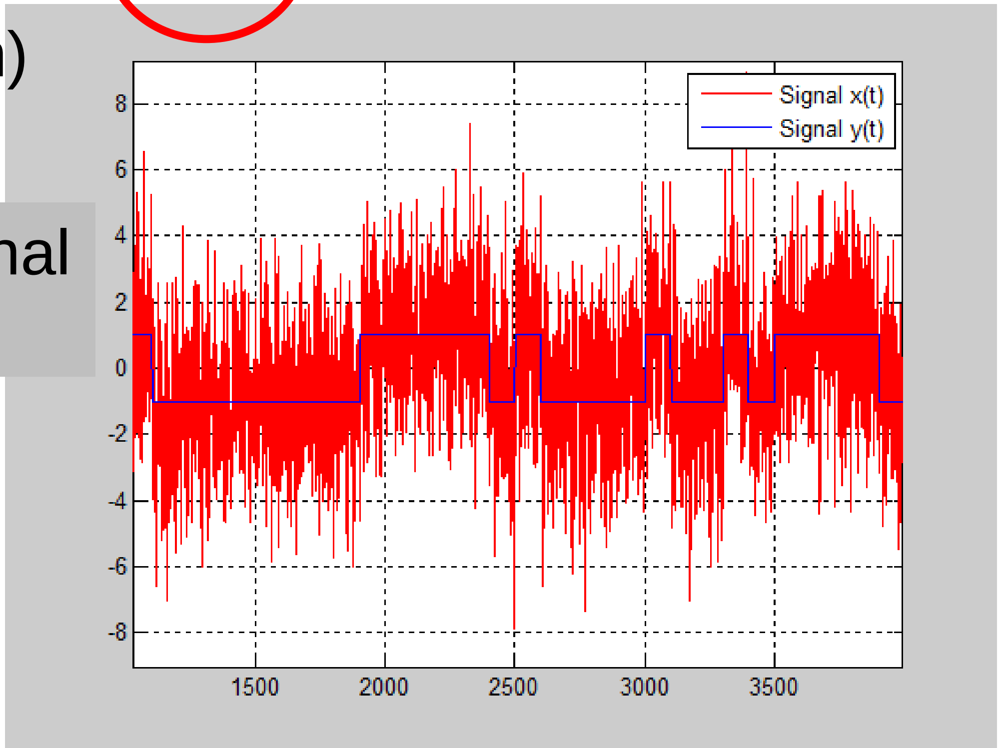


$P_x = 1V^2$
$P_n = 4V^2$

PSD estimated with periodogram method 100 realizations of 10^4 samples

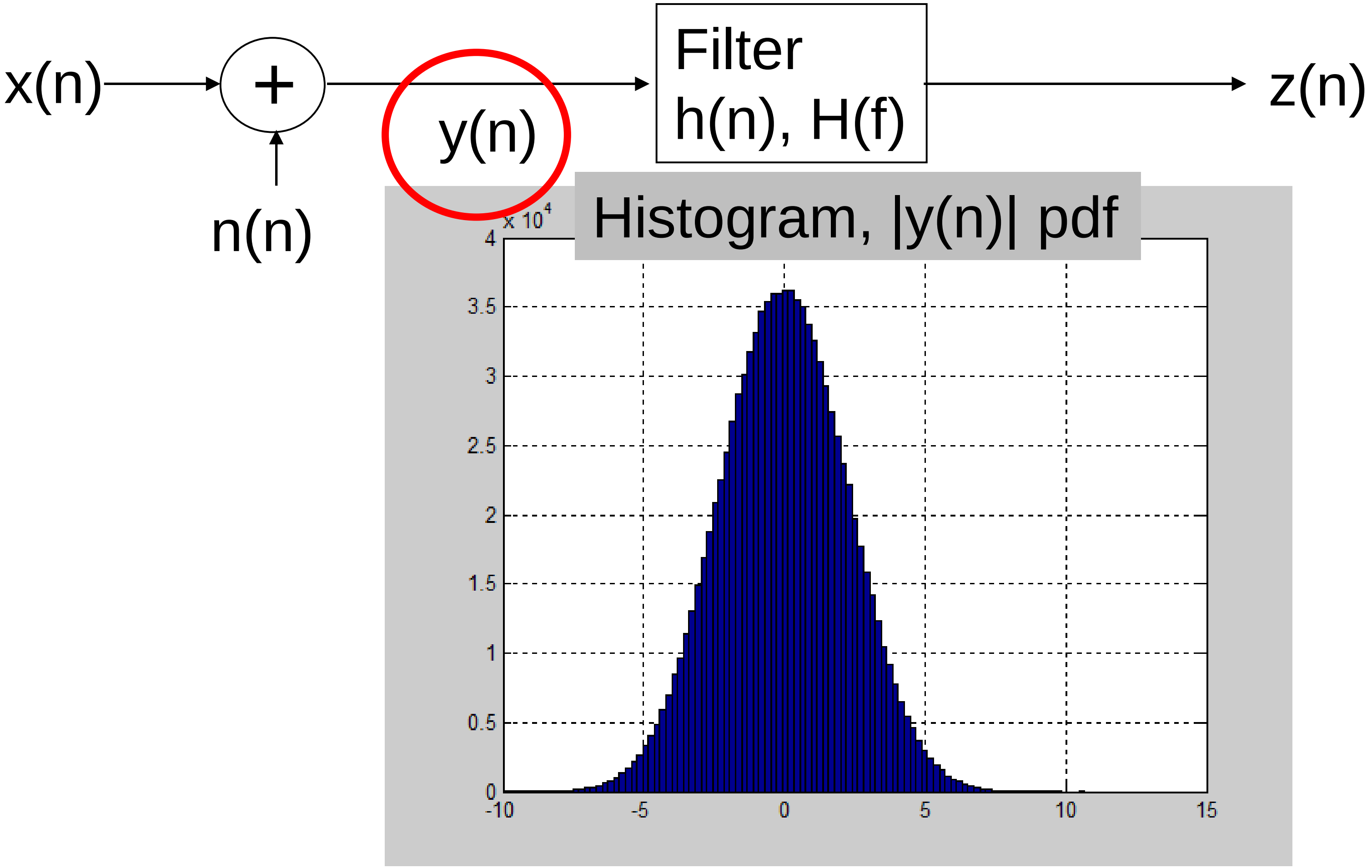
Spectral analysis with Matlab



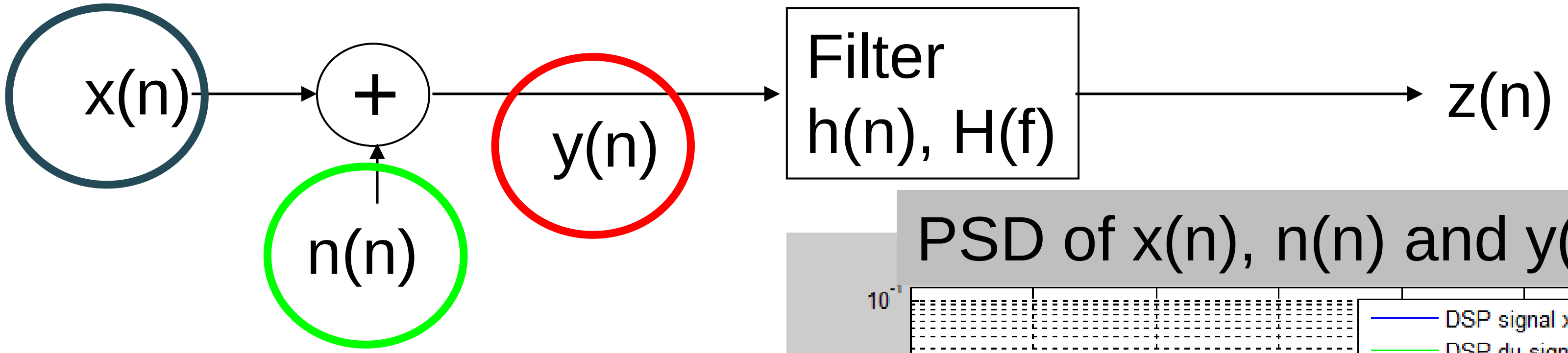


Time domain signal
$P_y = 5V^2$

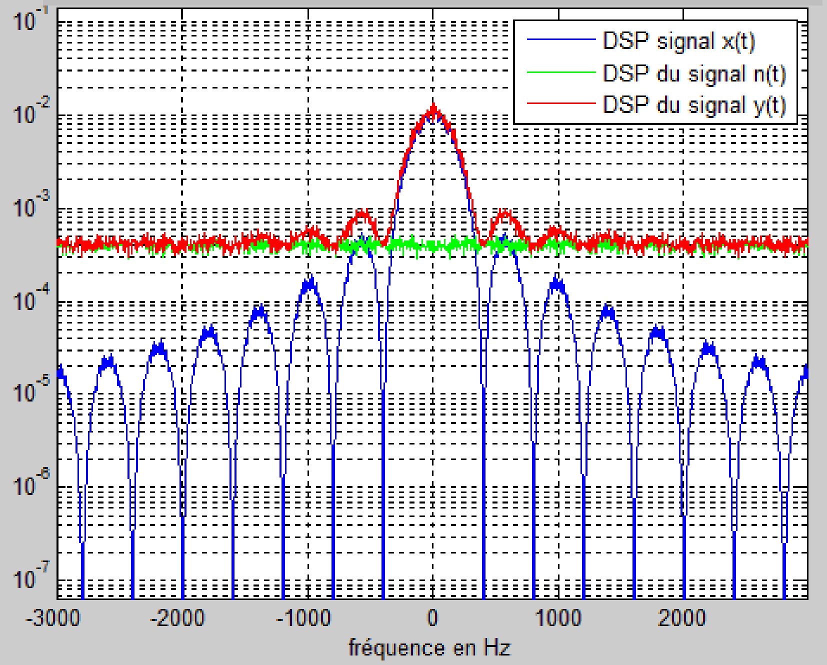
Spectral analysis with Matlab



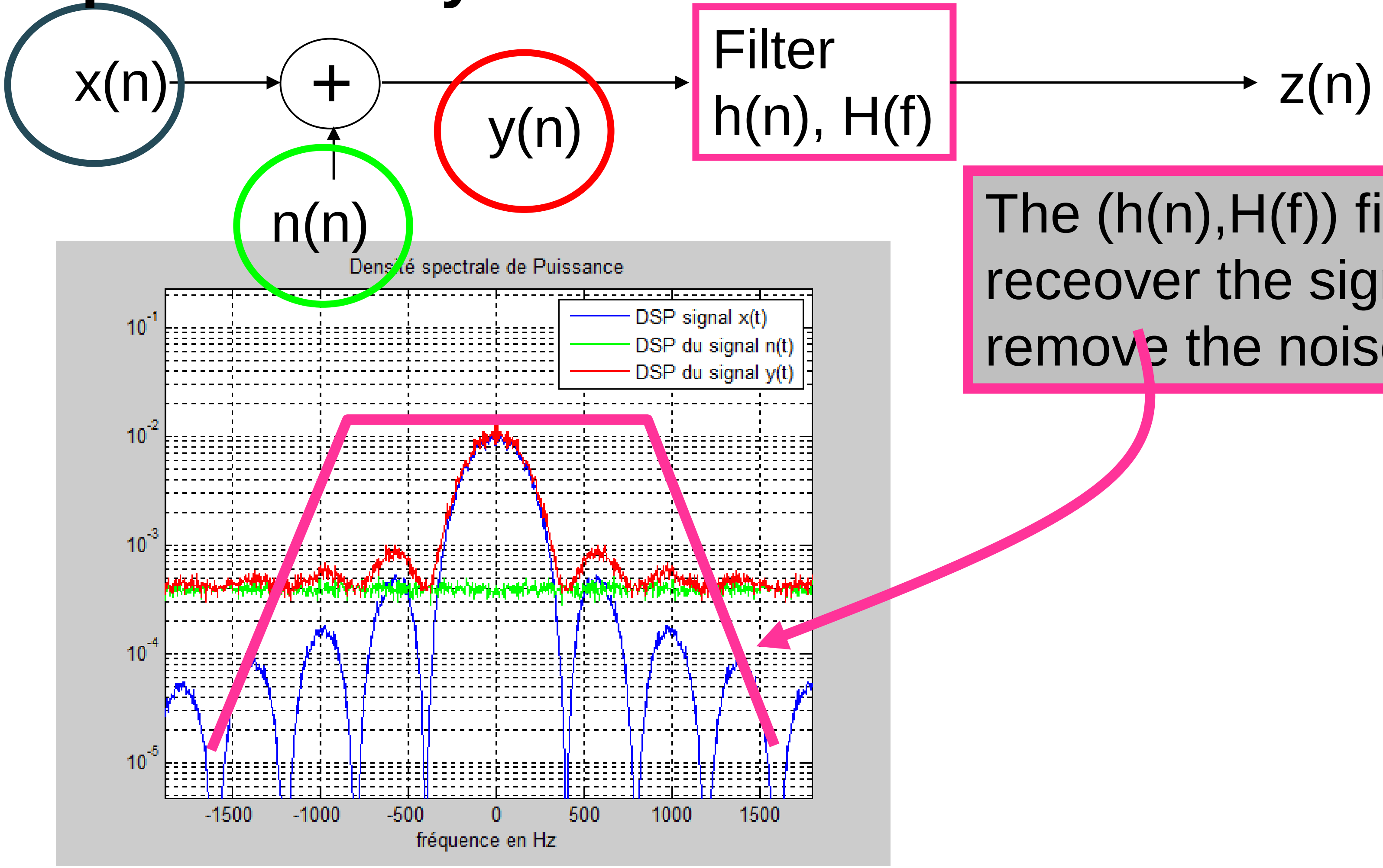
Spectral analysis with Matlab



PSD of x(n), n(n) and y(n)

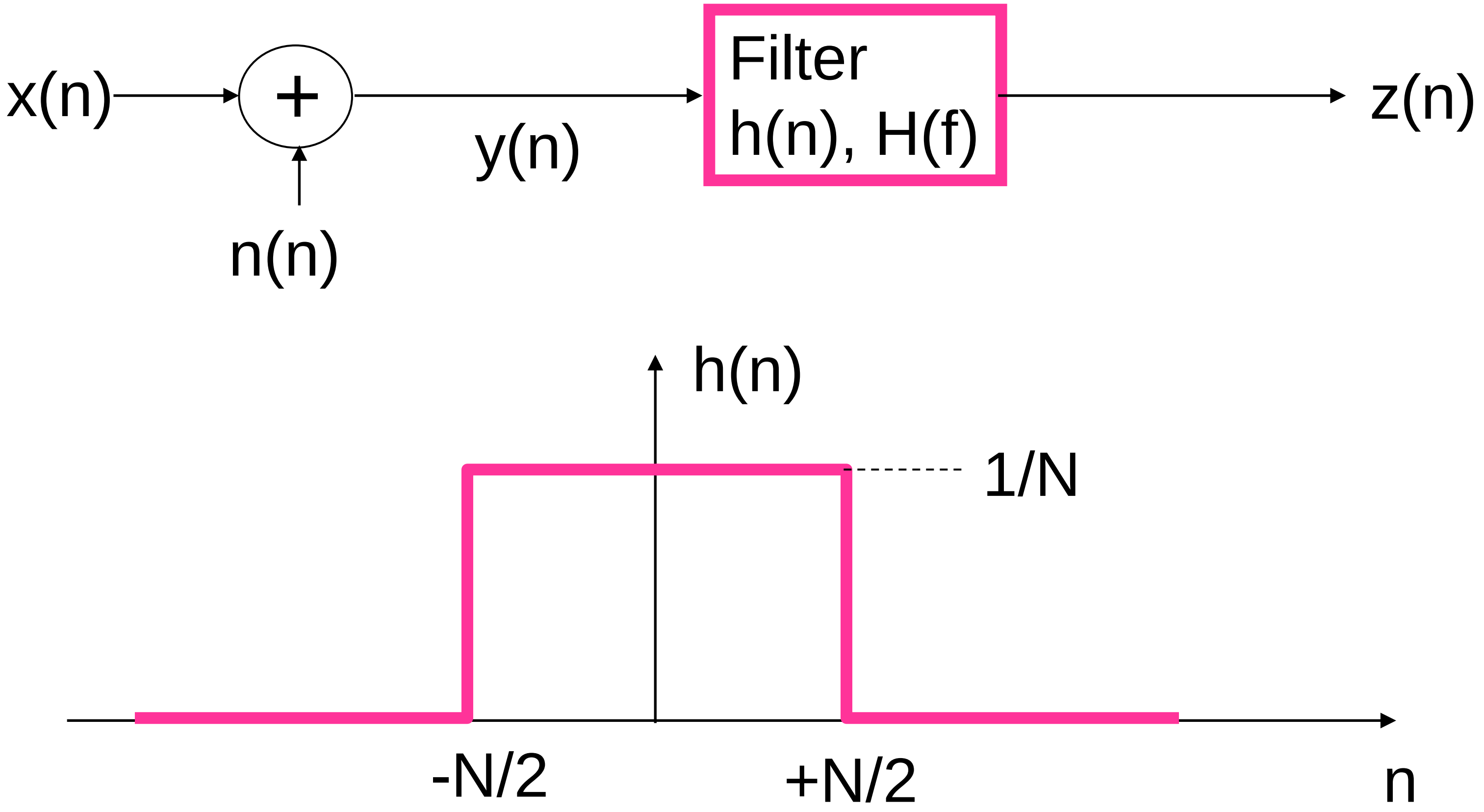


Spectral analysis with Matlab

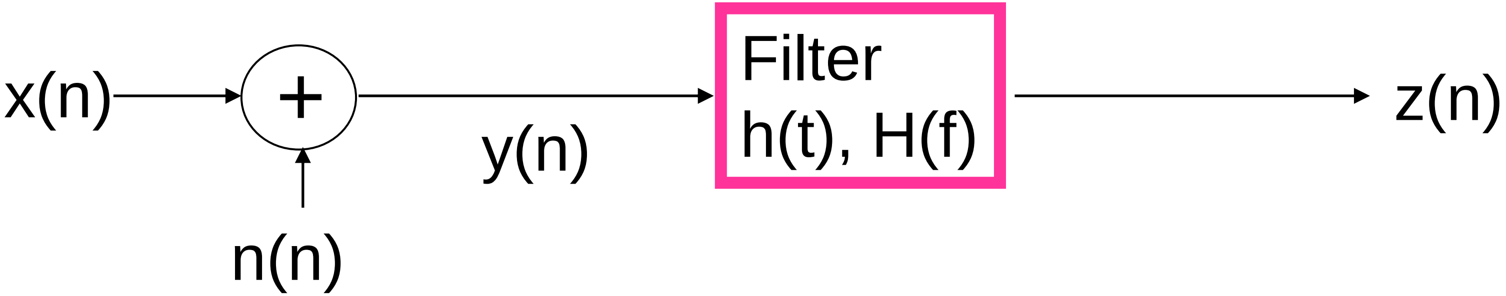


The (h(n),H(f)) filter : should receover the signal x(n) and remove the noise n(n)

Spectral analysis with Matlab

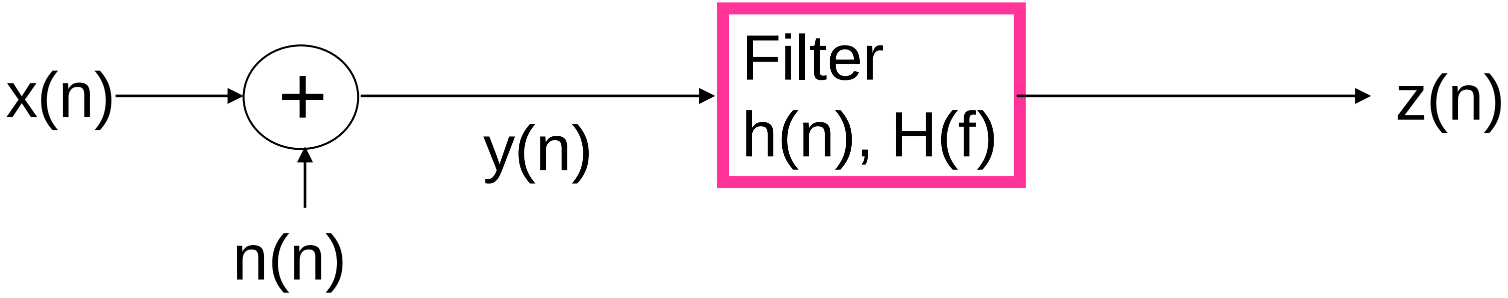


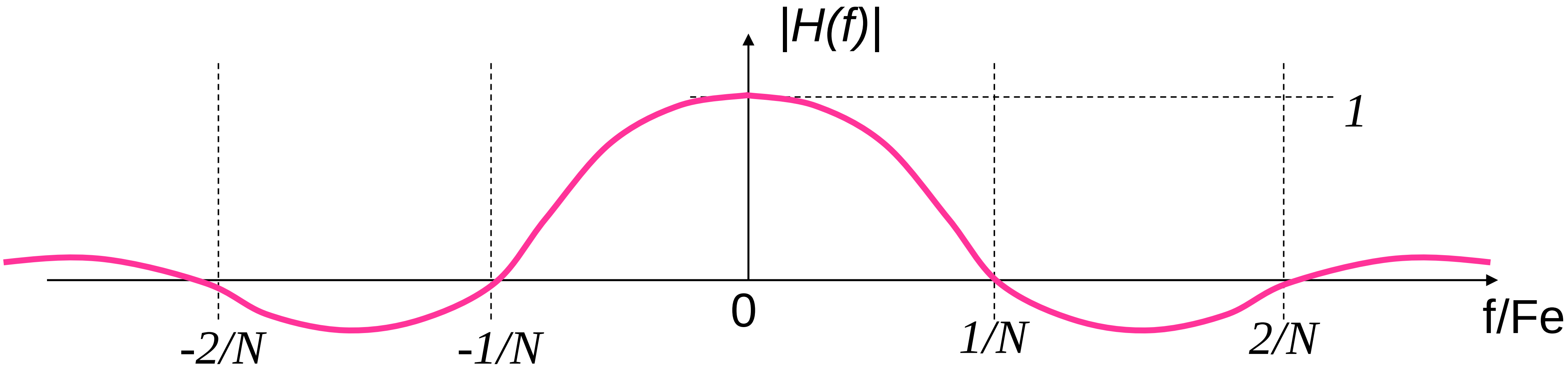
Spectral analysis with Matlab



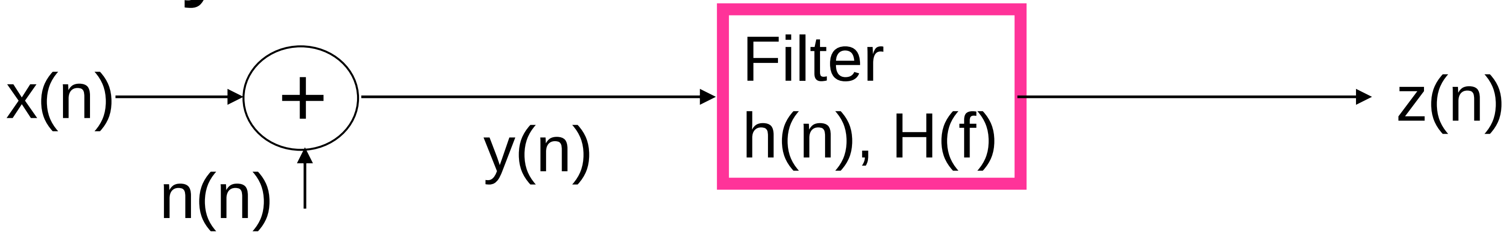
$$h(n) = \begin{bmatrix} \text{Rectangle of width N} \\ \text{and amplitude equal to} \\ \text{1//N centered arround 0} \end{bmatrix} \leftrightarrow H(f) = \frac{\sin(\pi f N)}{\pi f N}$$

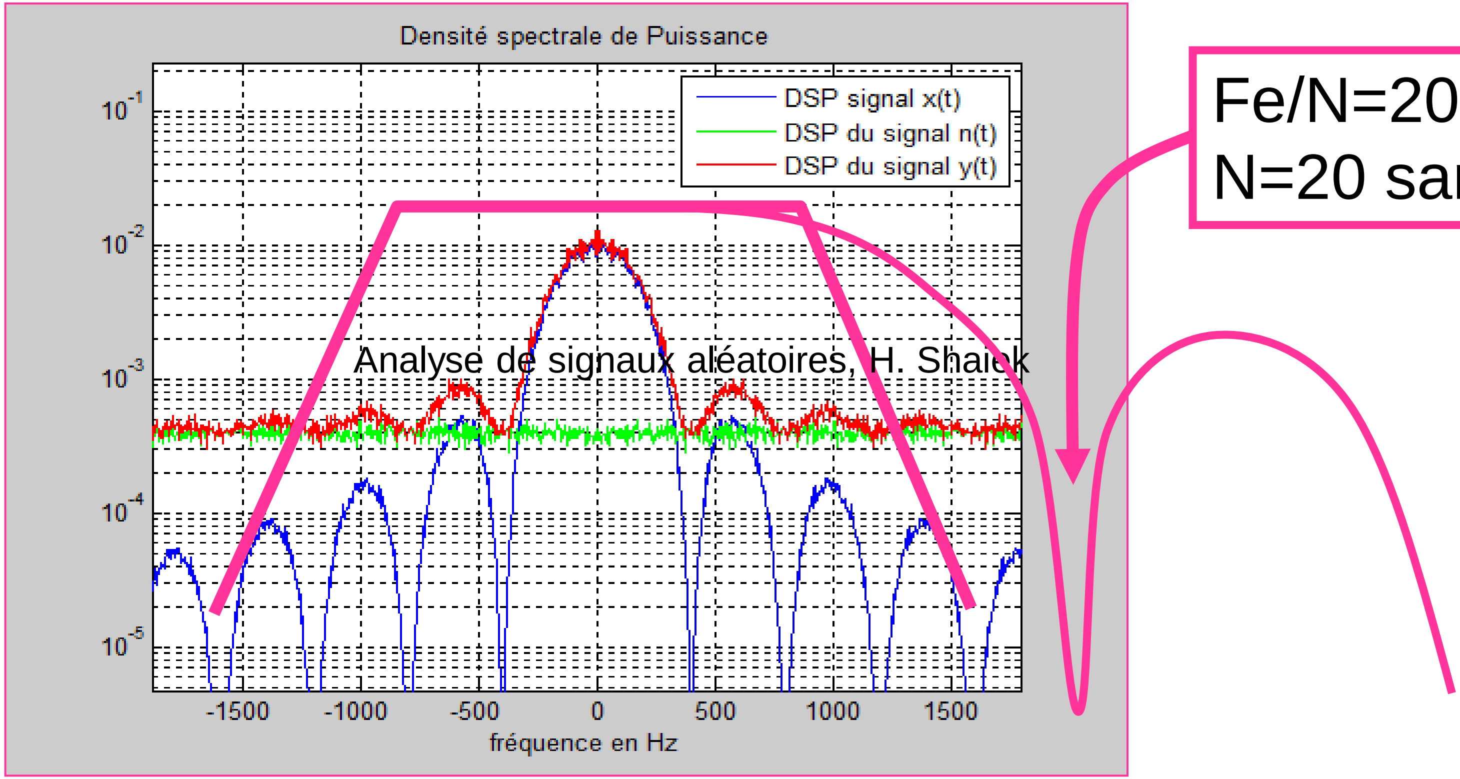
Spectral analysis with Matlab





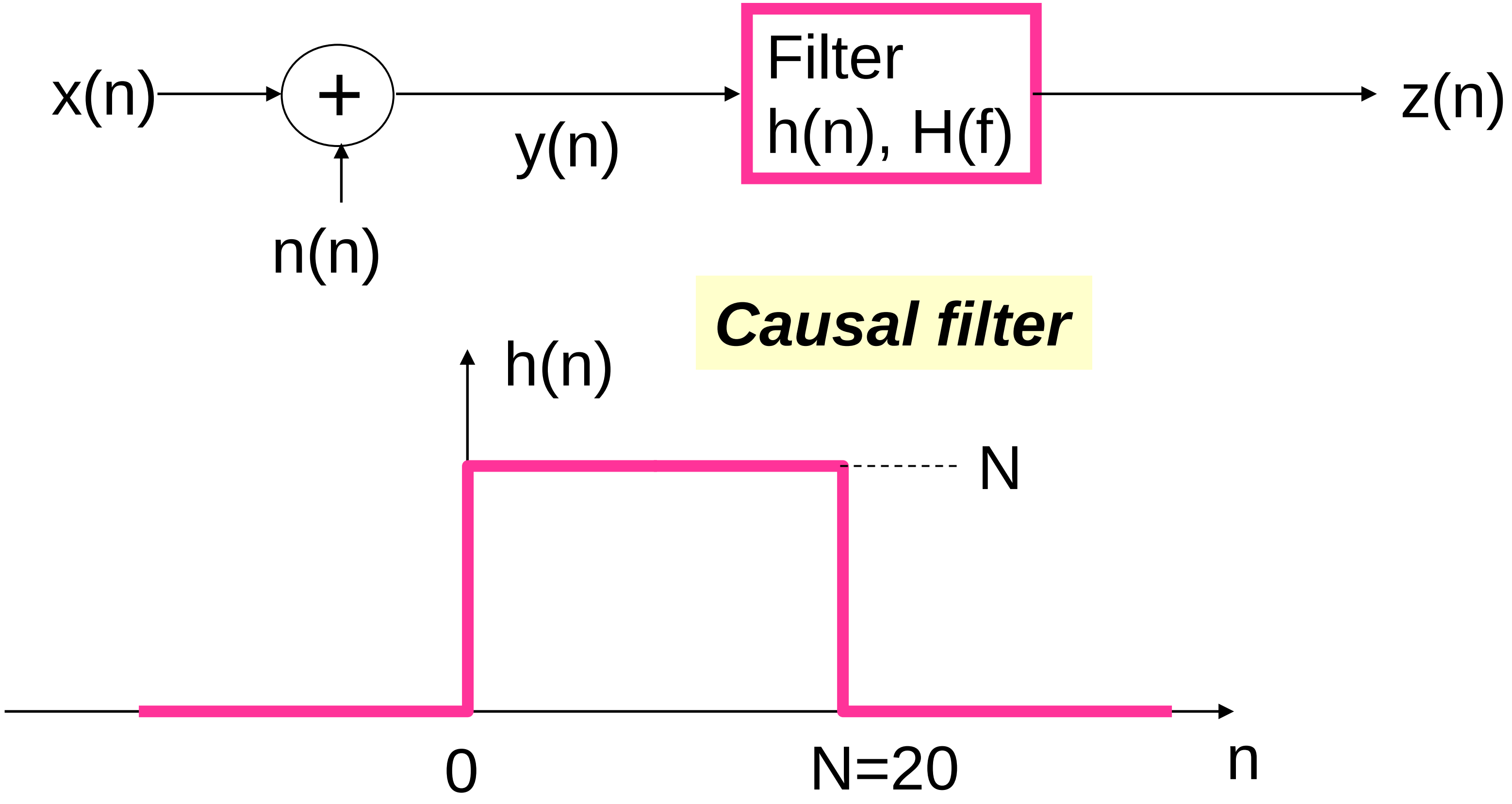
Spectral analysis with Matlab



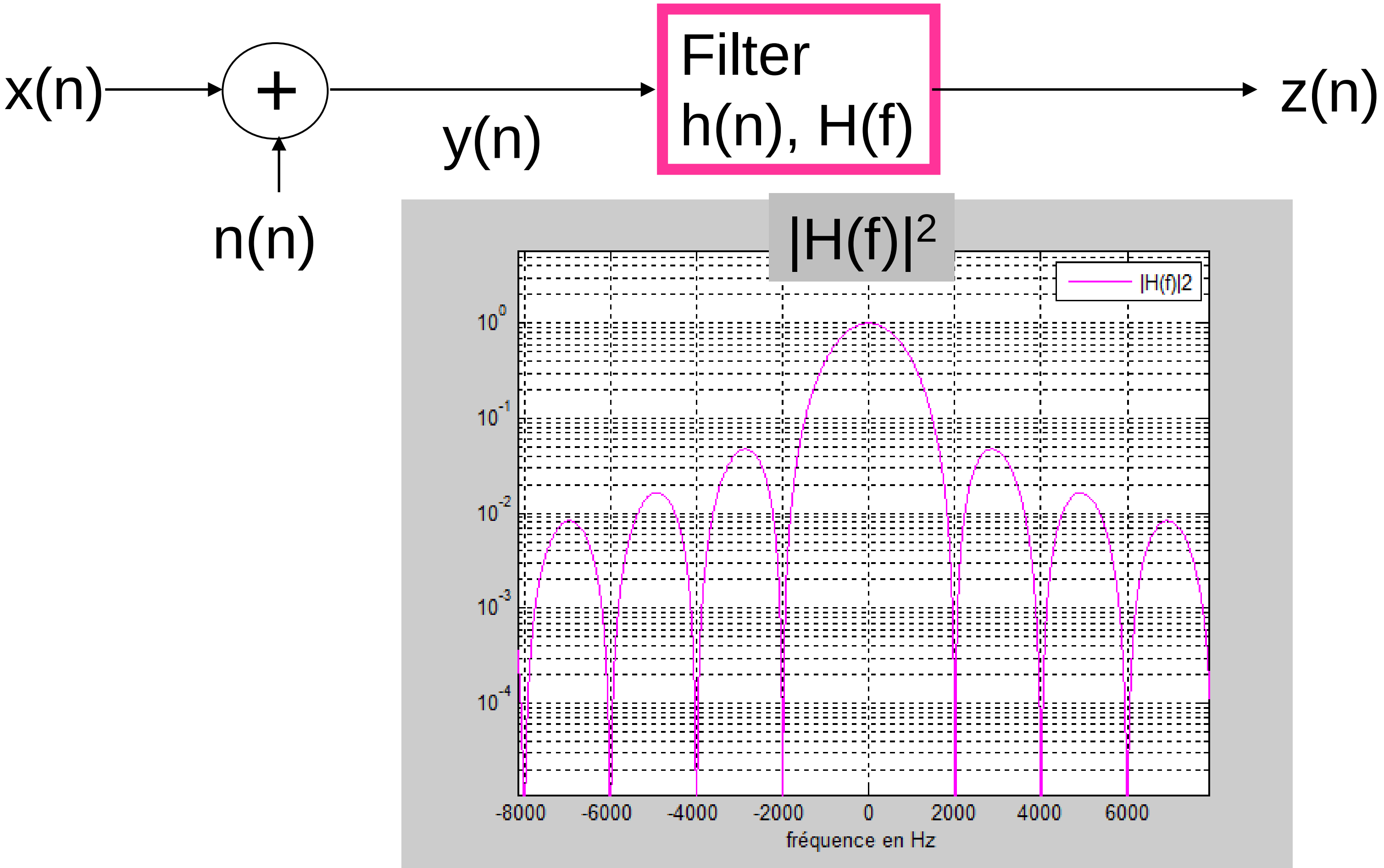


Fe/N=2000Hz
N=20 samples

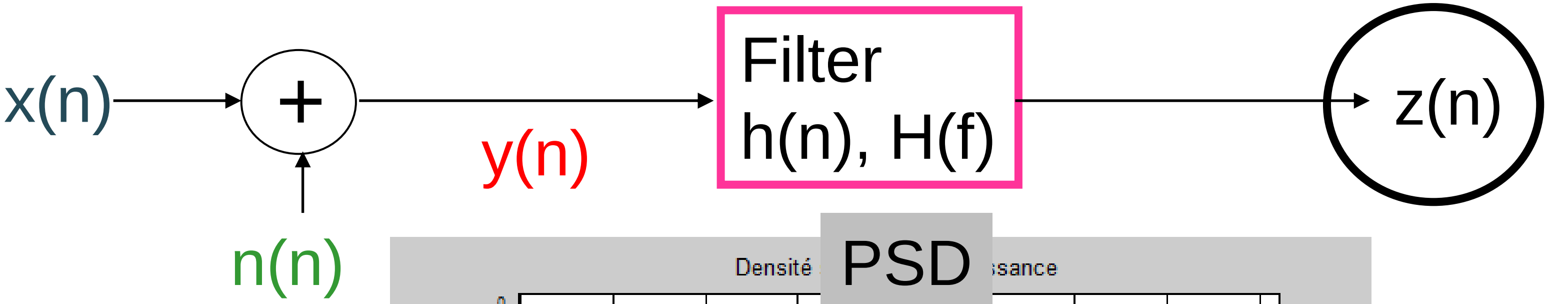
Spectral analysis with Matlab

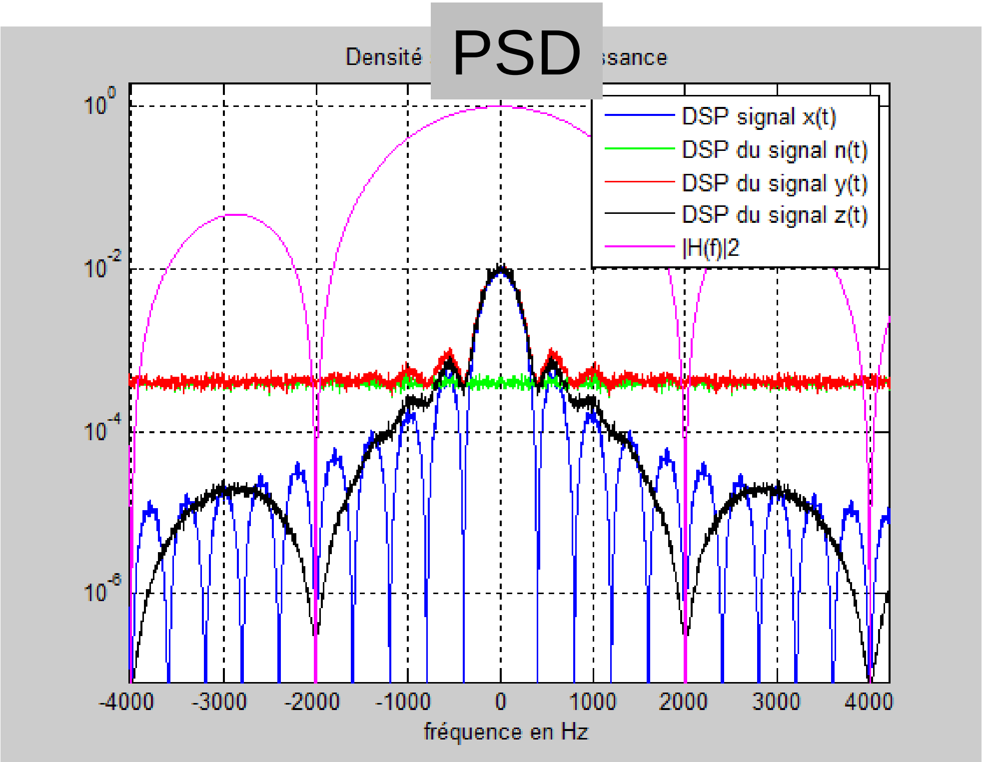


Spectral analysis with Matlab



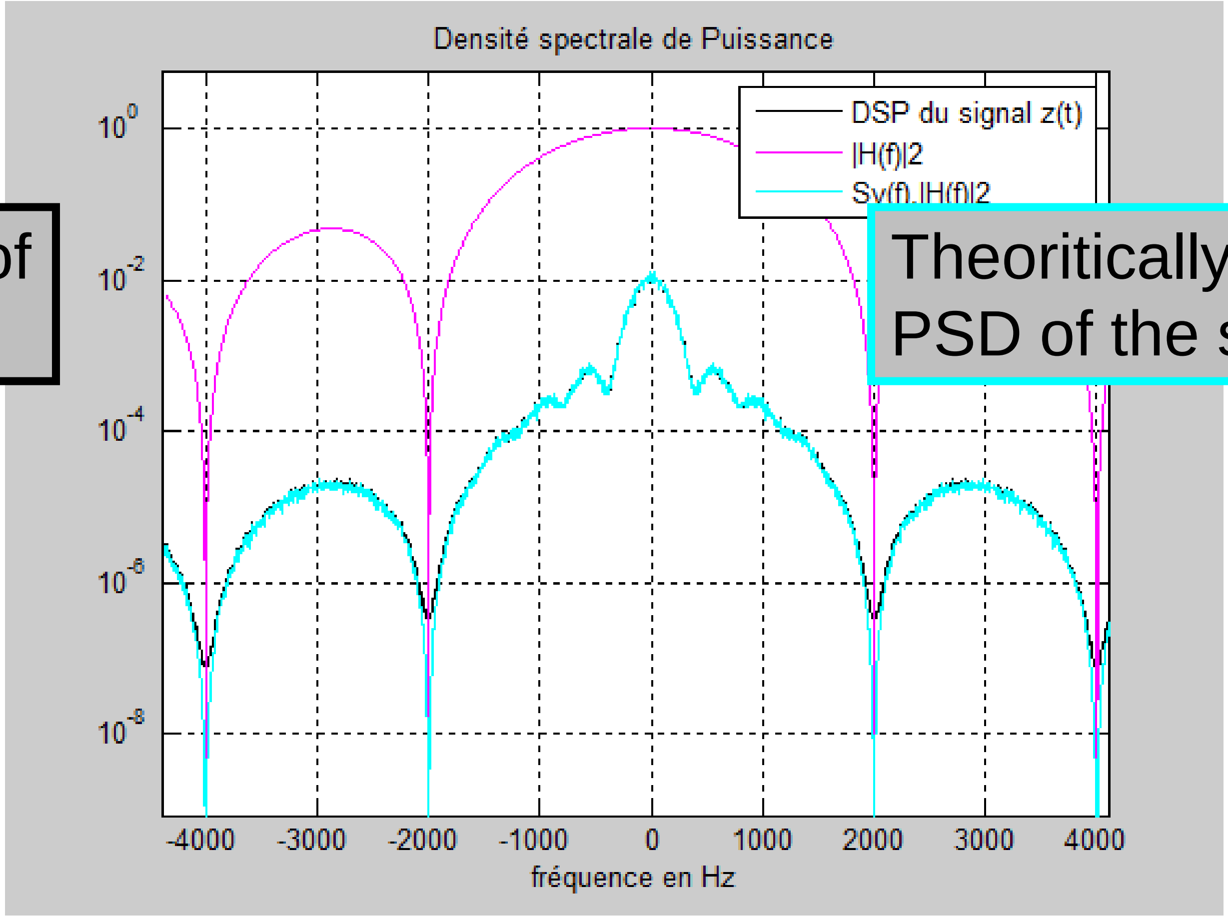
Spectral analysis with Matlab





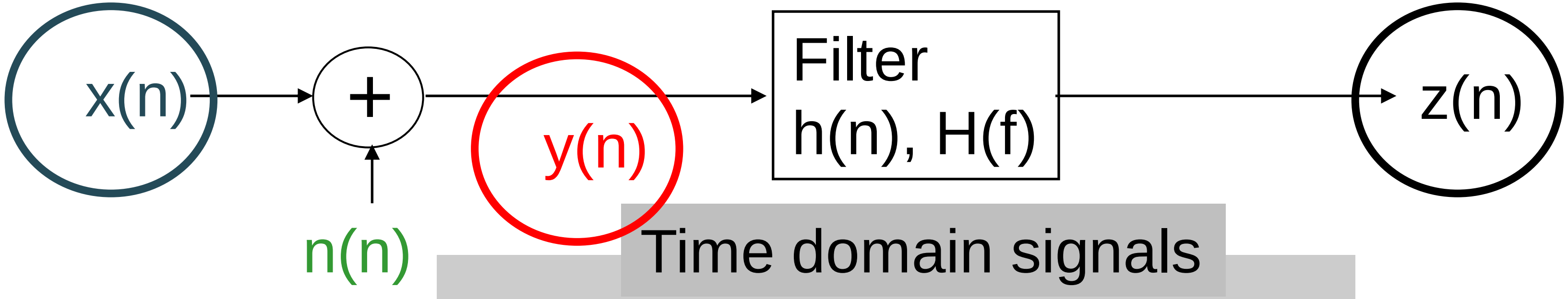
Spectral analysis with Matlab

Simulated PSD of the signal z(n)

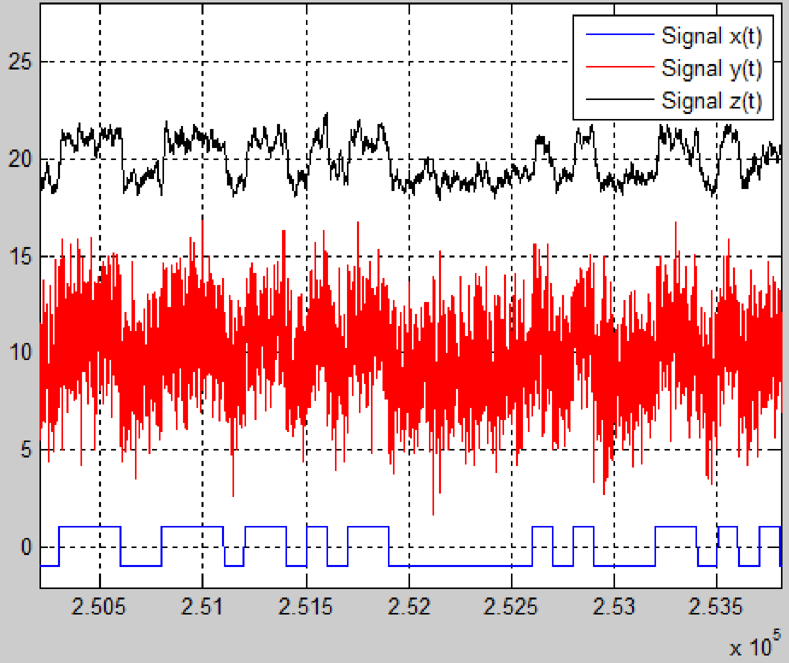


Theoritically computed PSD of the signal z(n)

Spectral analysis with Matlab



Time domain signals



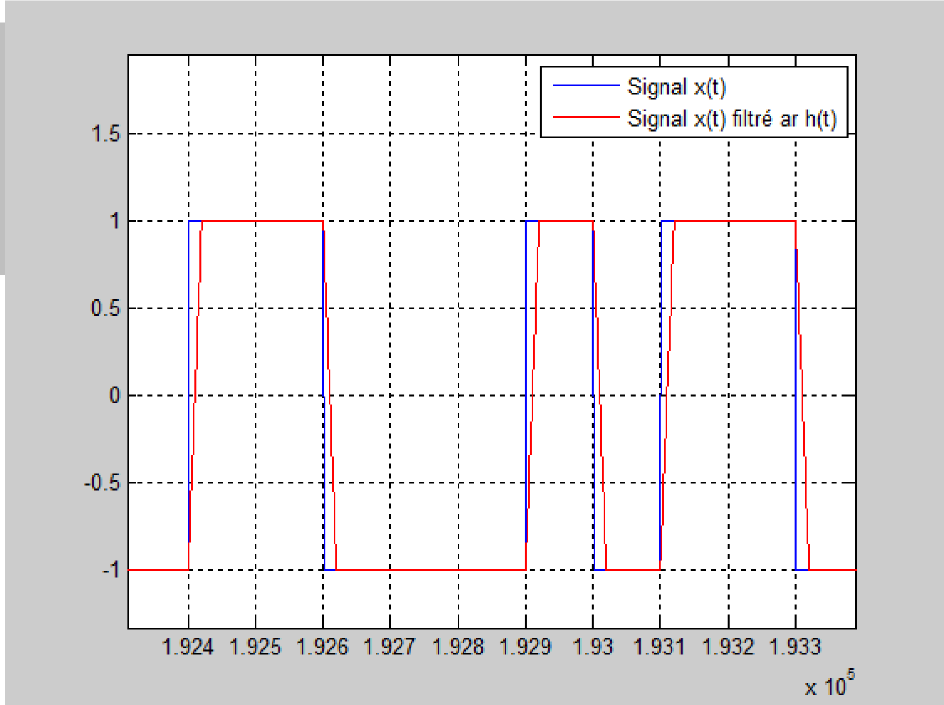
Spectral analysis with Matlab

x(n) → Filter h(n), H(f) → x1(n)

Time domain signal

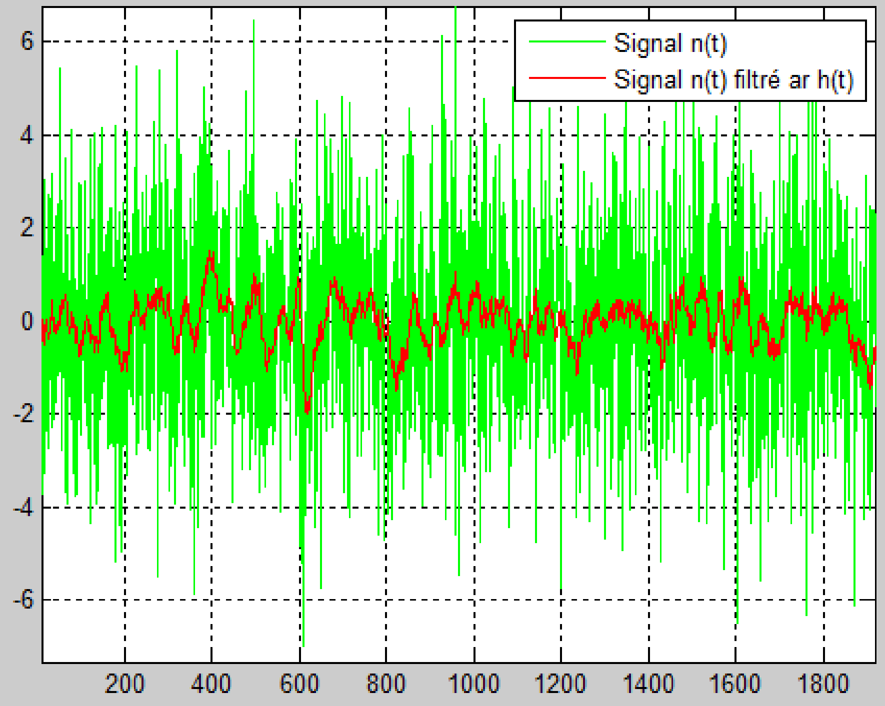
$P_x = 1V^2$

$P_{x1} = 0.93V^2$



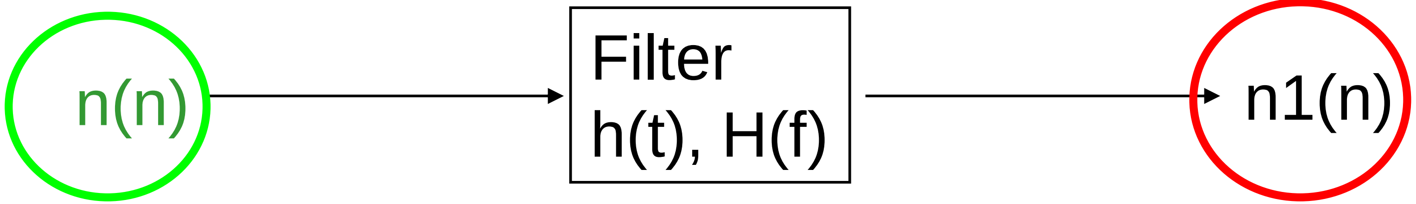
Spectral analysis with Matlab

n(n) → Filter h(n), H(f) → n1(n)

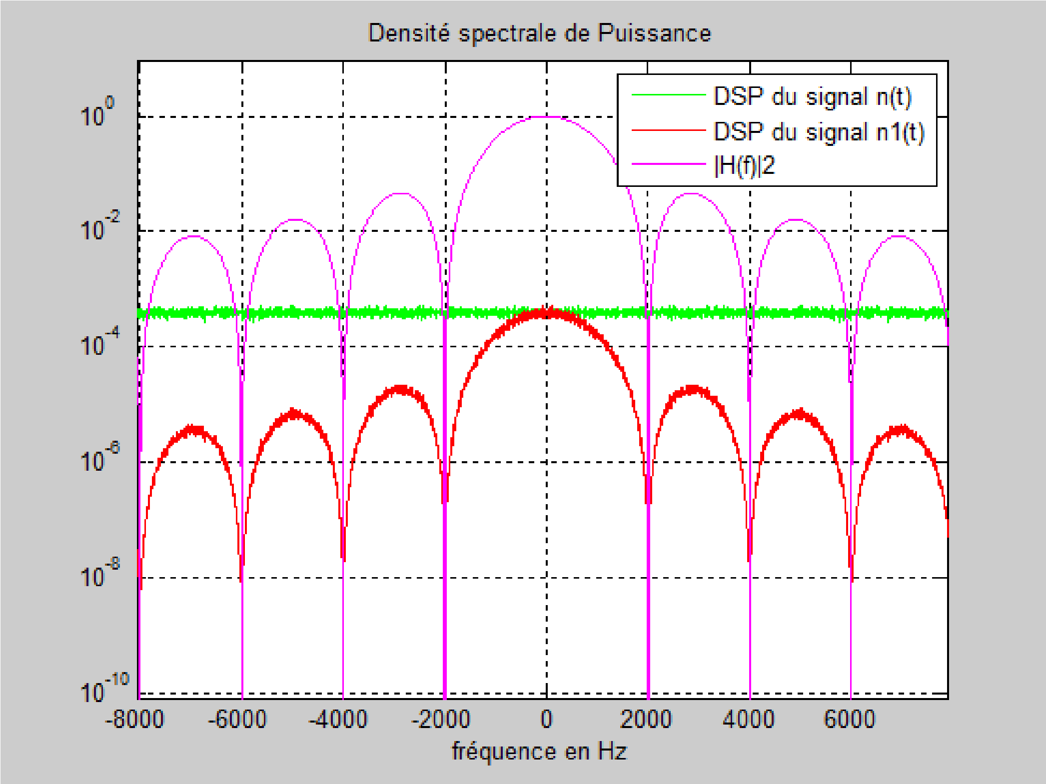


Time domain signal
$P_n = 4V^2$
$P_{n1} = 0.2V^2$

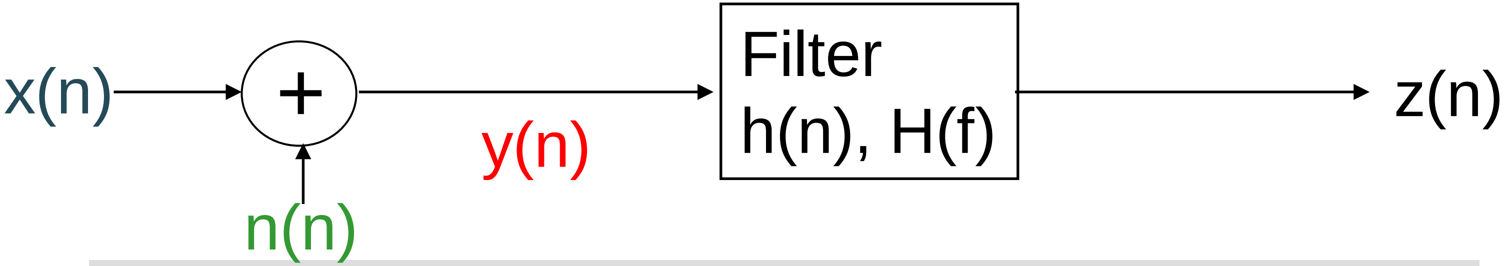
Spectral analysis with Matlab



PSD
Pn=4V²
Pn1=0.2V²



Spectral analysis with Matlab



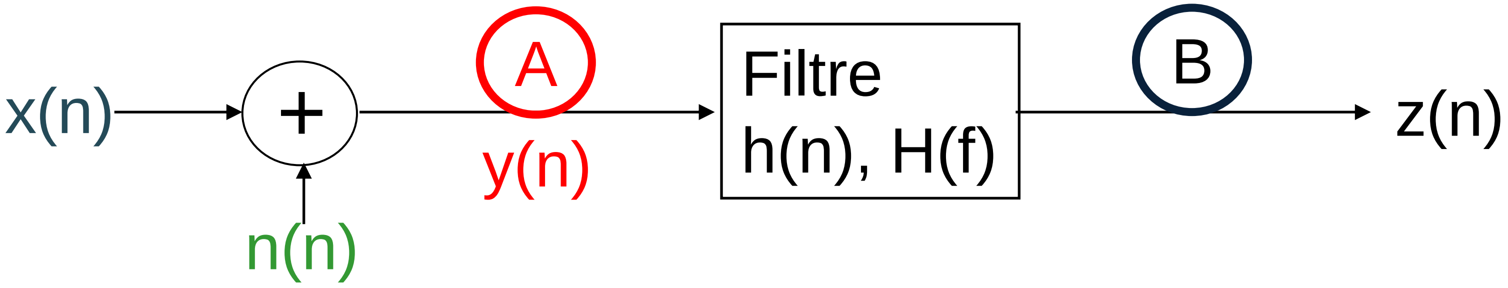
$$Px = \overline{x(n)^2} = 1$$

$$y(n) = x(n) + n(n), \qquad z(n) = x1(n) + n1(n)$$

$$Px1 = \int_{-\infty}^{+\infty} Sx1(f)df = \int_{-\infty}^{+\infty} Sx(f).|H(f)|^2 df = 0.93$$

$$Pn1 = \int_{-\infty}^{+\infty} Sn1(f)df = \int_{-\infty}^{+\infty} Sn(f).|H(f)|^2 df = 0.2$$

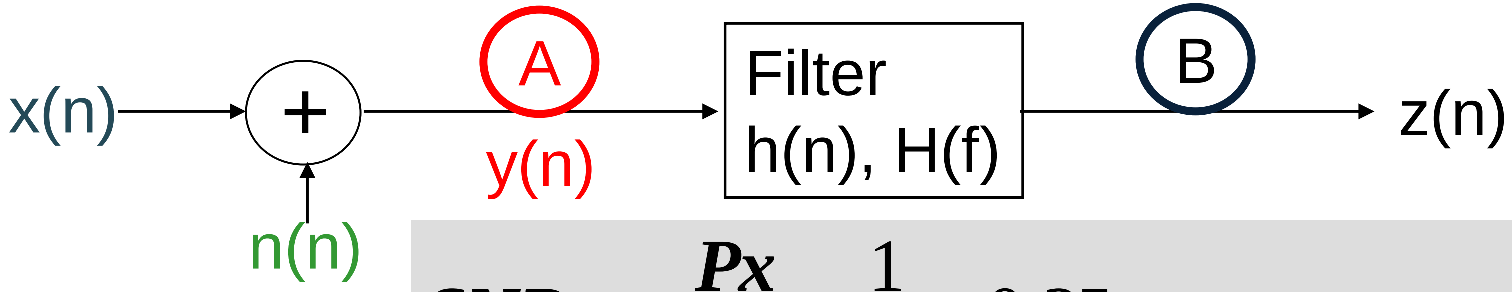
Spectral analysis with Matlab



Signal to **N**oise **R**atio (en anglais) = SNR

SNR=Power of the useful signal / Noise power

Spectral analysis with Matlab



$$SNR_A = \frac{Px}{Pn} = \frac{1}{4} = 0.25$$

$$SNR_B = \frac{Px1}{Pn1} = \frac{0.93}{0.2} = 4.61$$

$$SNR_A dB = 10.\log 10(SNR_A) = -6dB$$

$$SNR_B dB = 10.\log 10(SNR_B) = +6.63dB$$

Refrences

For further reading

[1] Boaz Porat: A Course in Digital Signal Processing, Wiley, ISBN 0-471-14961-6.

[2] John G. Proakis, Dimitris Manolakis: Digital Signal Processing: Principles, Algorithms and Applications, 4th ed, Pearson, April 2006, ISBN 978-0131873742.

[3] Oppenheim, Alan V.; Schafer, Ronald W. (2001). Discrete-Time Signal Processing. Pearson. ISBN 1-292-02572-7.