Traitement Numérique du Signal

Nathalie Thomas

IRIT/ENSEEIHT
Nathalie.Thomas@enseeiht.fr

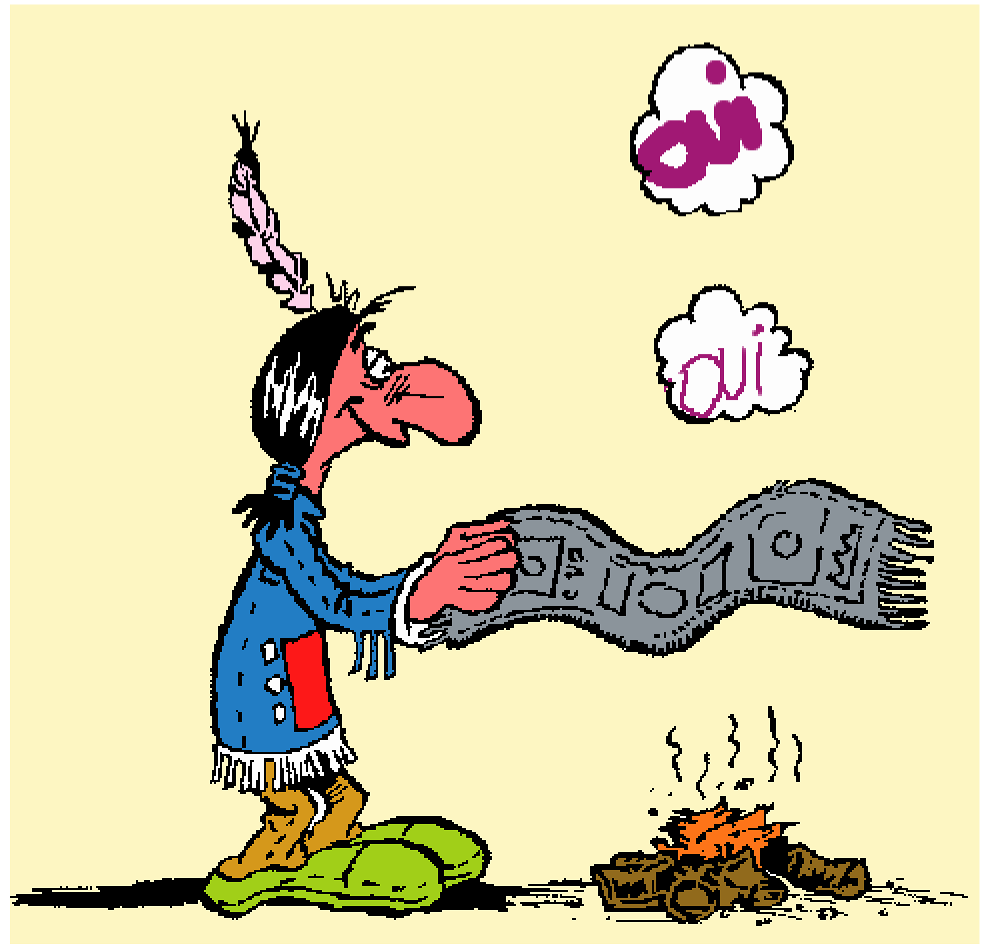
Traitement Numérique du Signal

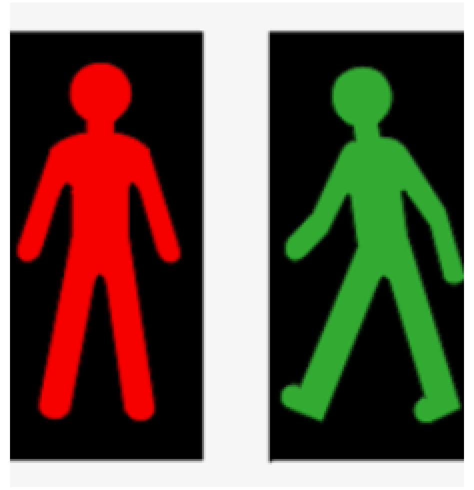
INTRODUCTION

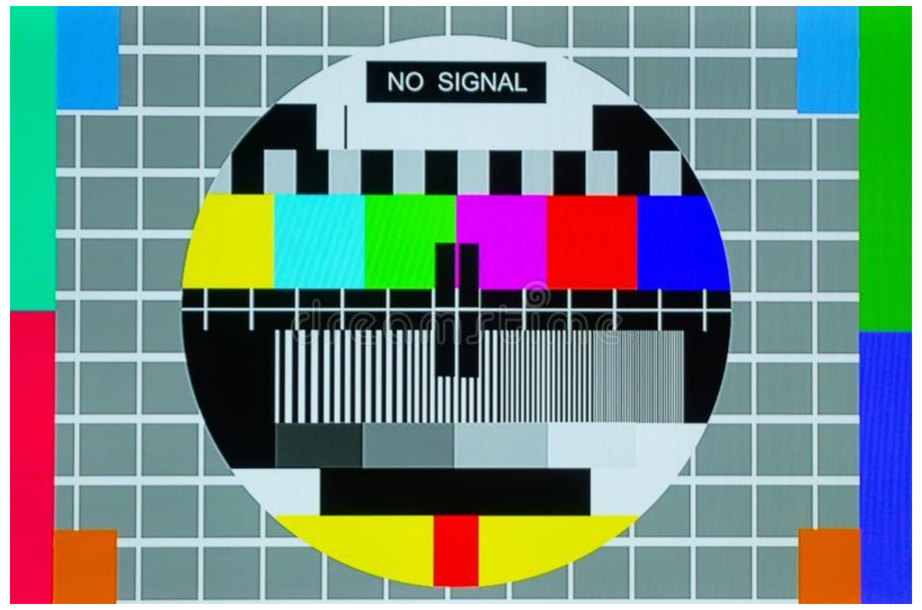
Qu’est-ce qu’un signal ?

Qu’est-ce qu’un signal ?

→ Formes multiples et variées de signaux

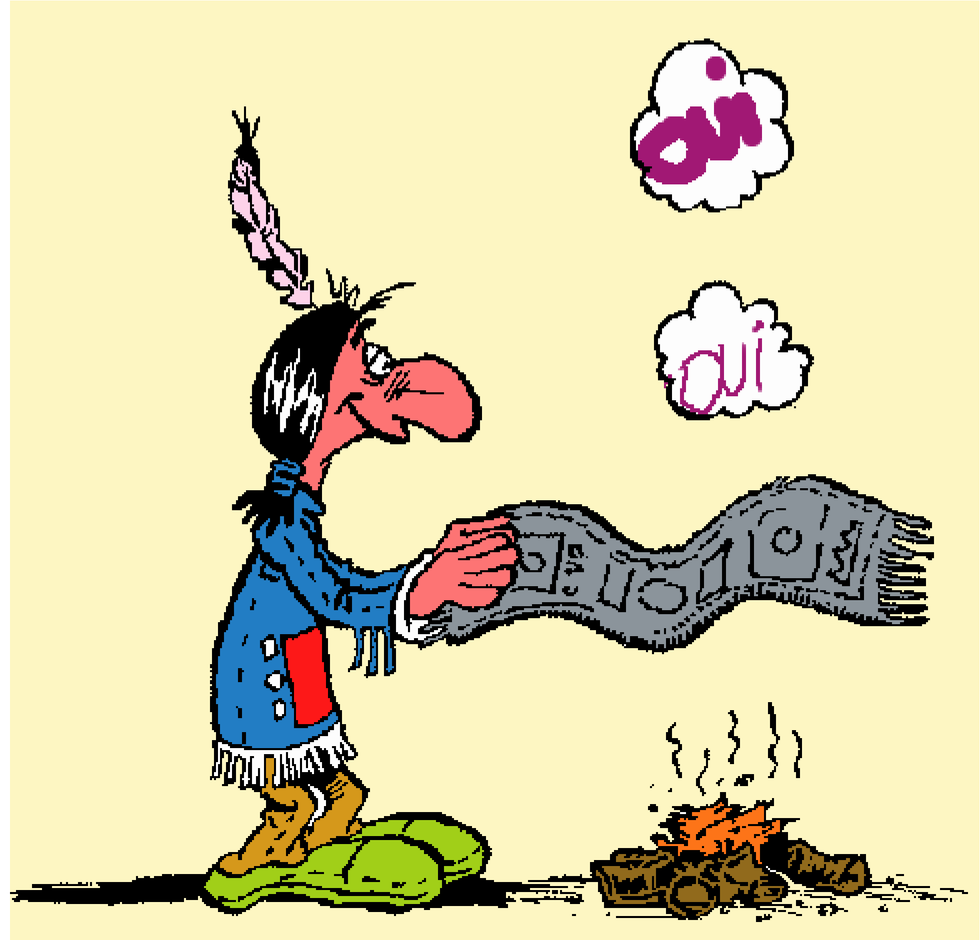


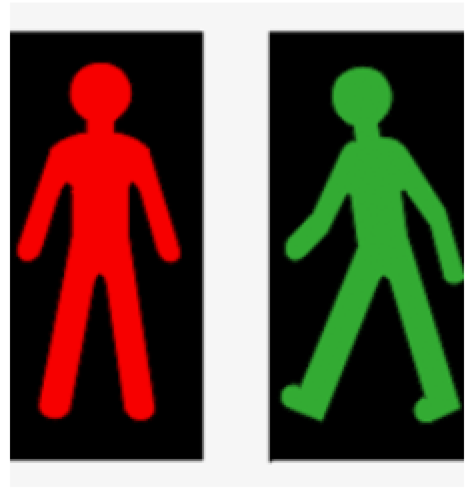


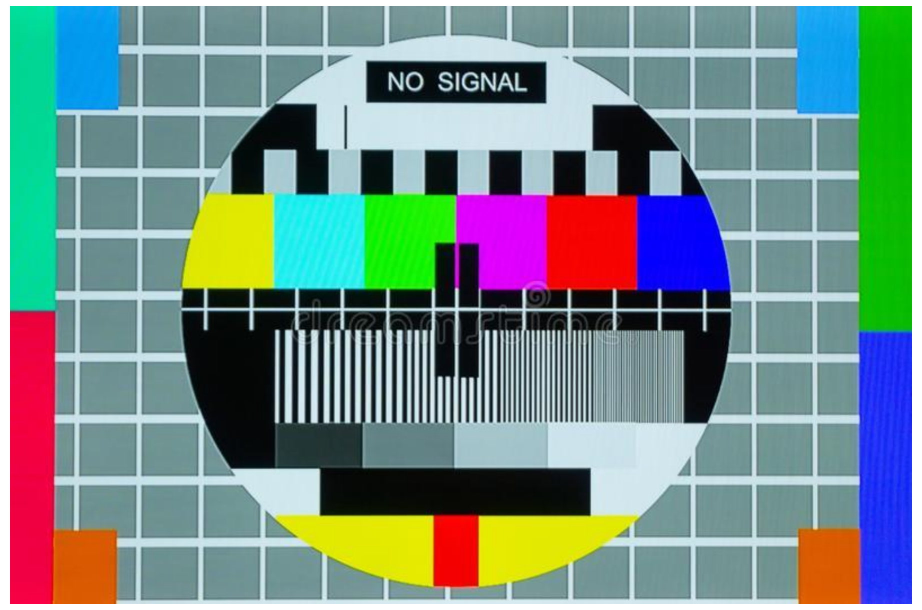


Qu’est-ce qu’un signal ?

→ <u>Formes multiples et variées de signaux</u>



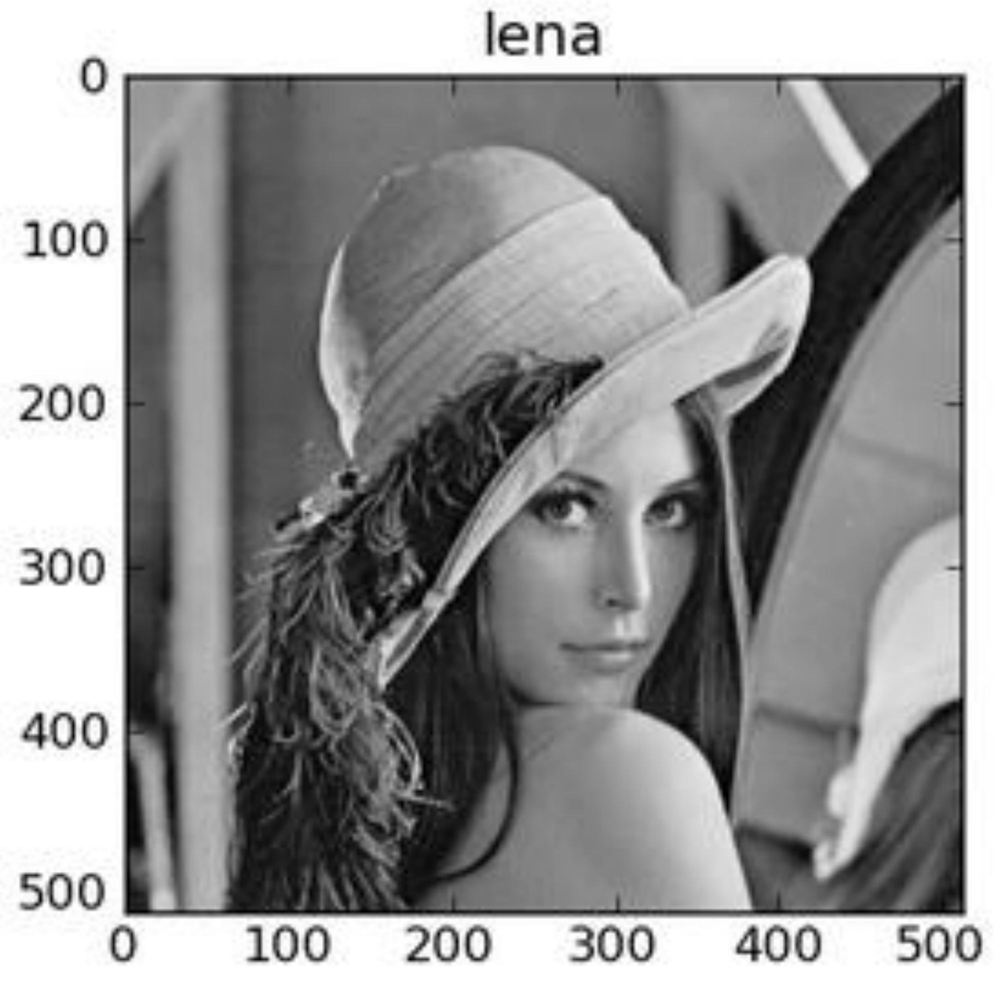




<u>Point commun :</u> représentent un message, contiennent une information.

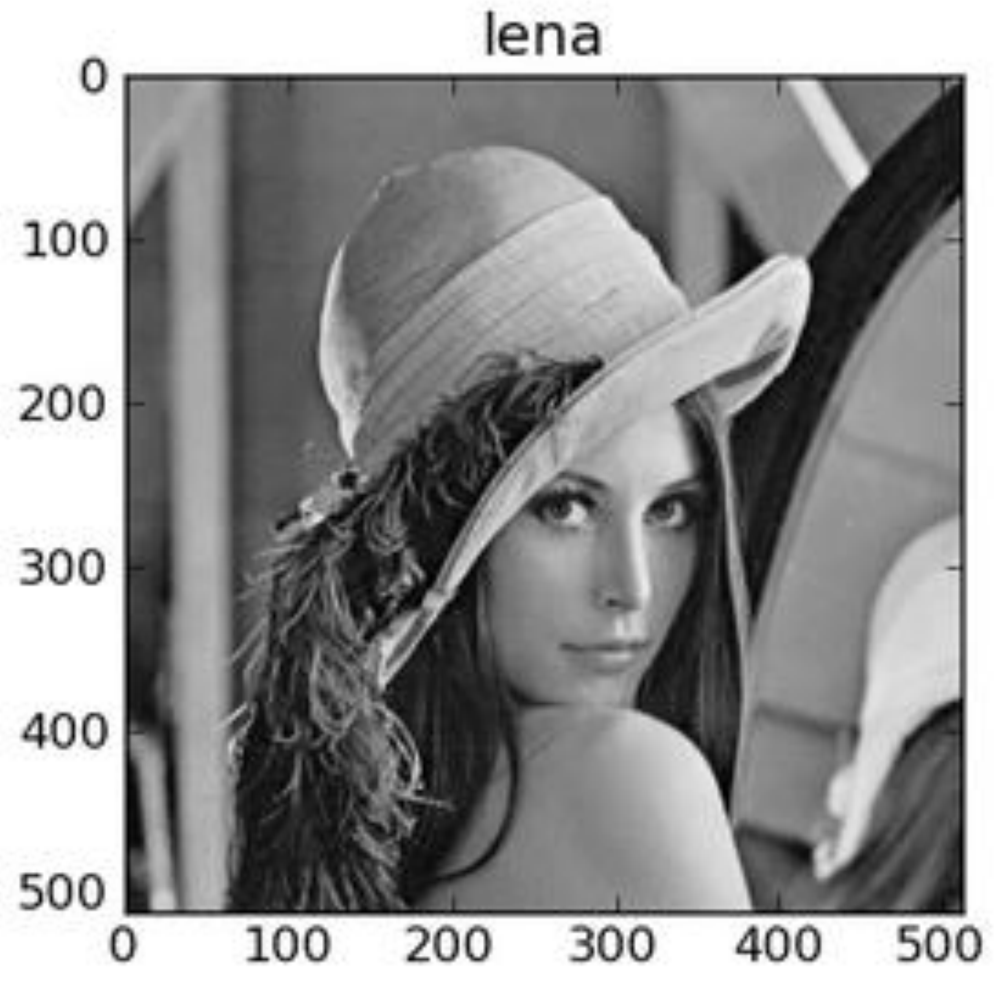
Qu'est-ce qu'un signal ?

→ <u>Représentation théorique :</u> x(t), I(x,y)…

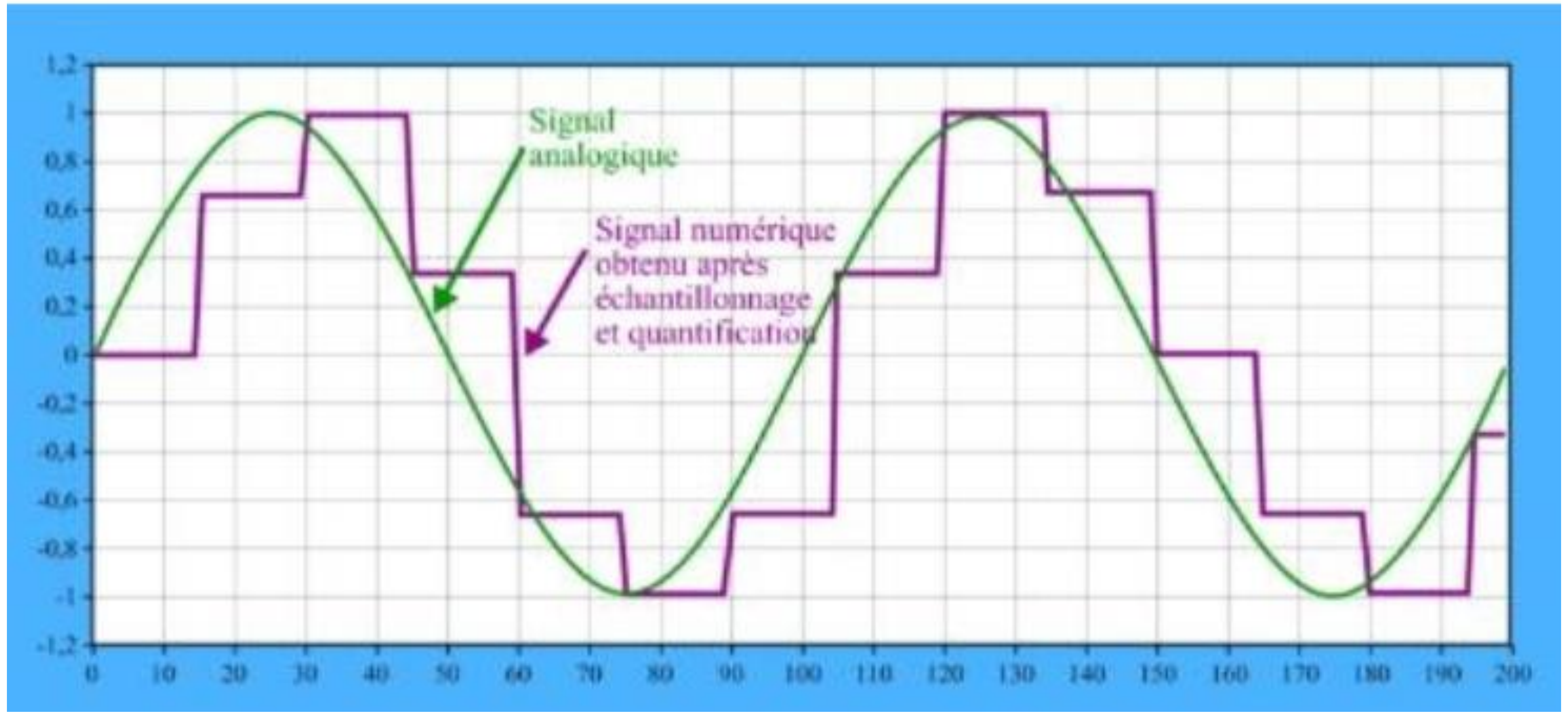


Qu’est-ce qu’un signal ?

→ Représentation théorique : x(t), I(x,y)…



→ Signaux analogiques, signaux numériques (échantillonnage, quantification)



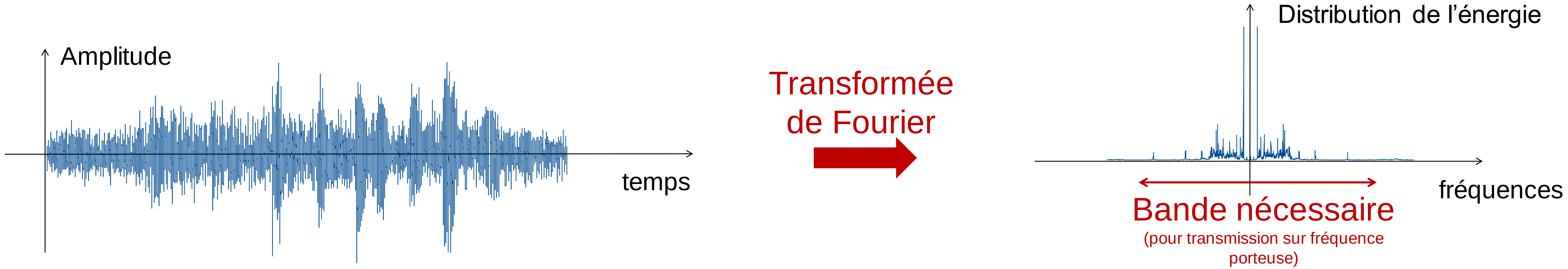
Le traitement du signal : pourquoi ?

Le traitement du signal : pourquoi ?

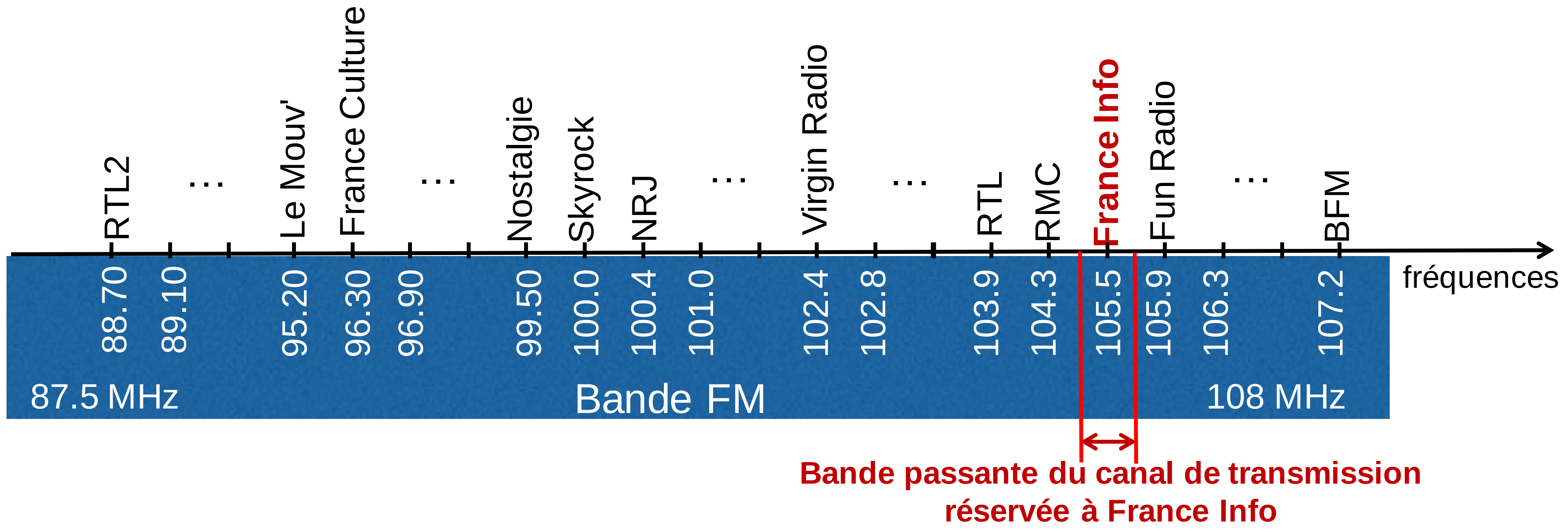
→ Pour extraire de l'information des signaux :

Exemple 1

- Identifier la bande de fréquence nécessaire à la transmission d'un signal,
- Détecter des anomalies, des défauts (ECG, Arcs électriques sur les câbles d'alimentation d'un avion, dent cassée dans un engrenage…)
- En éliminant des composantes indésirables : le bruit, certaines fréquences…



Exemple de partage de bande : bande FM à Toulouse



Le traitement du signal : pourquoi ?

→ Pour extraire de l'information des signaux :

Exemple 2

- Identifier la bande de fréquence nécessaire à la transmission d'un signal,
- Détecter des anomalies, des défauts (ECG, Arcs électriques sur les câbles d'alimentation d'un avion, dent cassée dans un engrenage…)
- En éliminant des composantes indésirables : le bruit, certaines fréquences…

SNR = 0 dB

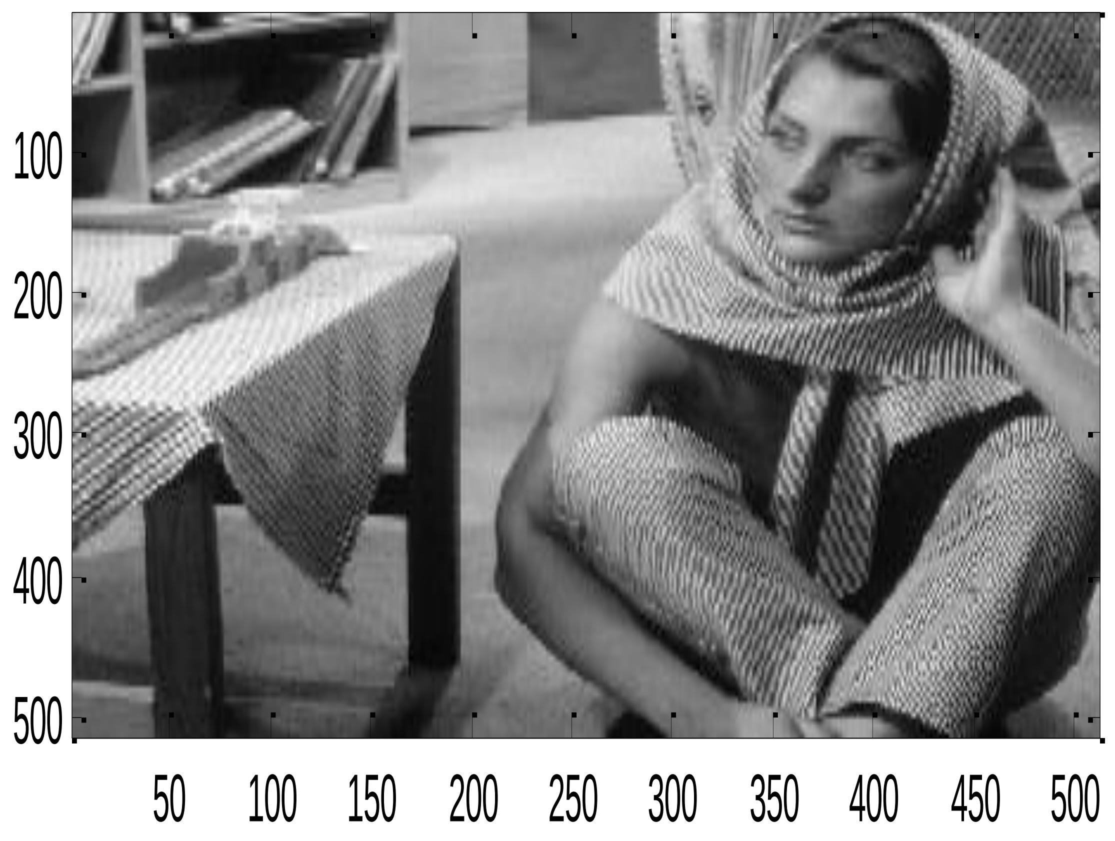


Image émise

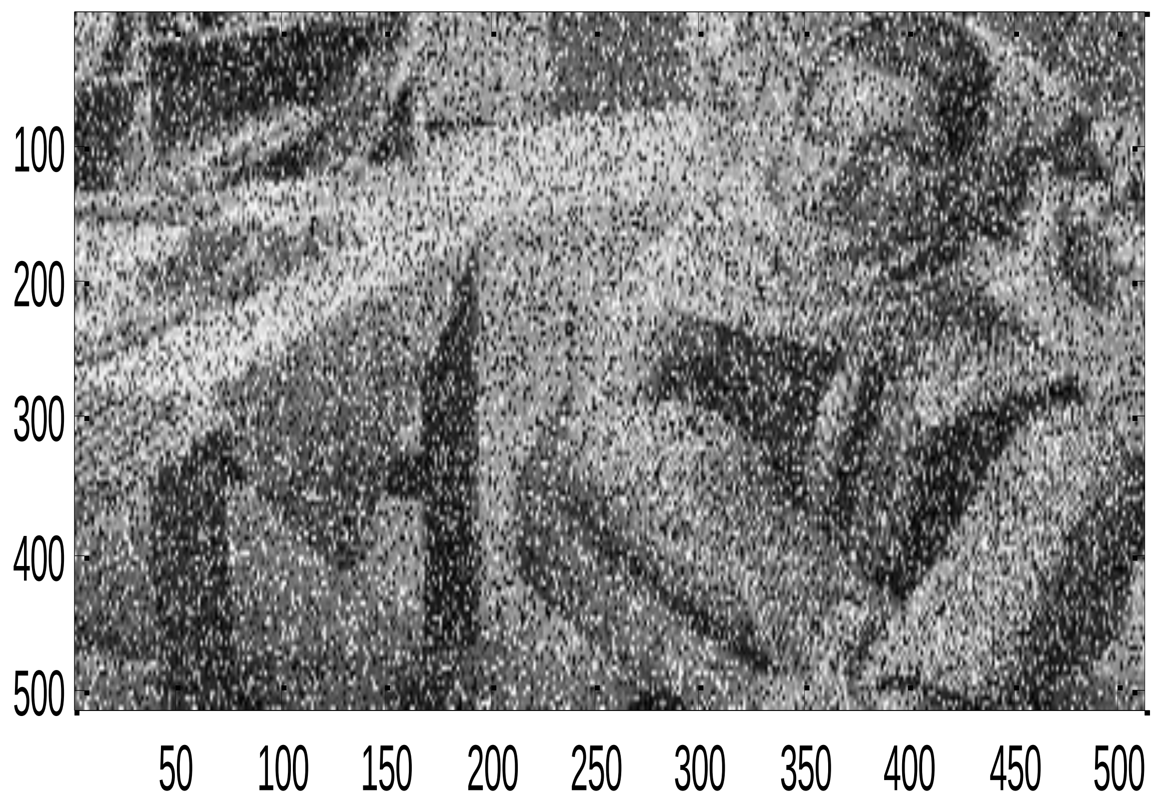


Image reçue
Avant filtrage de réception

TEB = 0.1581

Filtrage

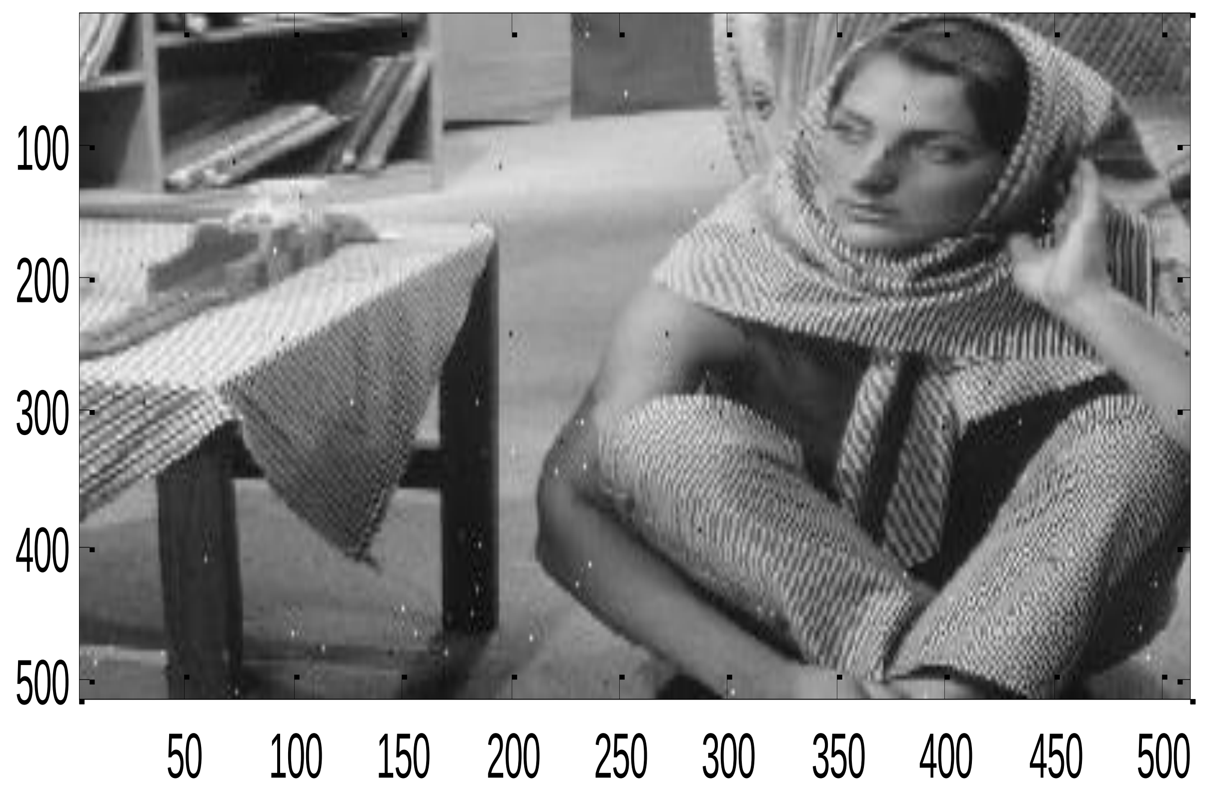


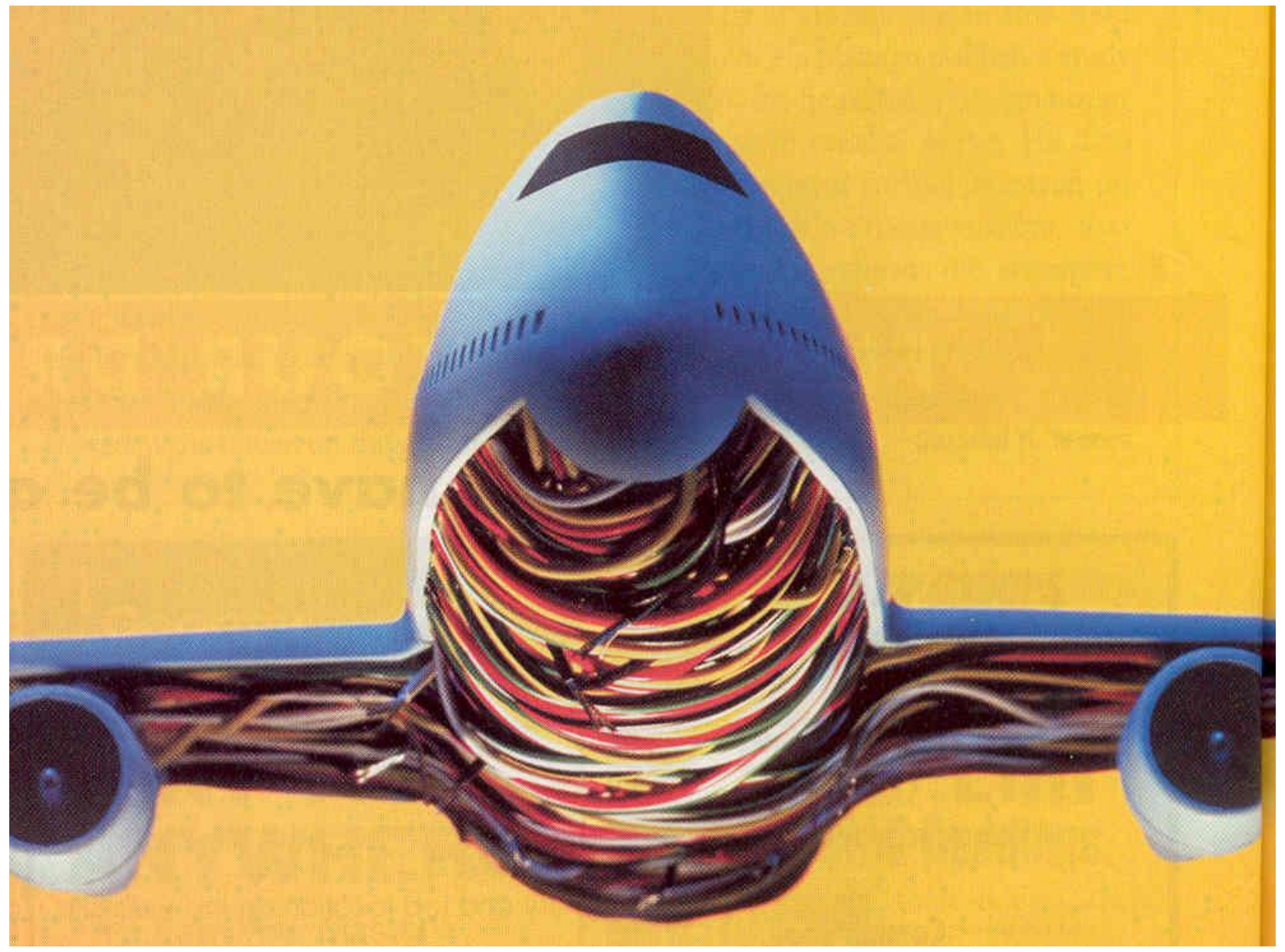
Image reçue
après filtrage de réception

TEB = 7.5483e-04

Le traitement du signal : pourquoi ?

→ Pour extraire de l'information des signaux :   Exemple 3

- Identifier la bande de fréquence nécessaire à la transmission d'un signal,
- Détecter des anomalies, des défauts (ECG, Arcs électriques sur les câbles d'alimentation d'un avion, dent cassée dans un engrenage…)
- En éliminant des composantes indésirables : le bruit, certaines fréquences…



Usure des gaines d'isolation

⇒ Arcs électriques

⇒ Possible destruction d'une partie du réseau d'alimentation de l'avion.

Le traitement du signal : pourquoi ?

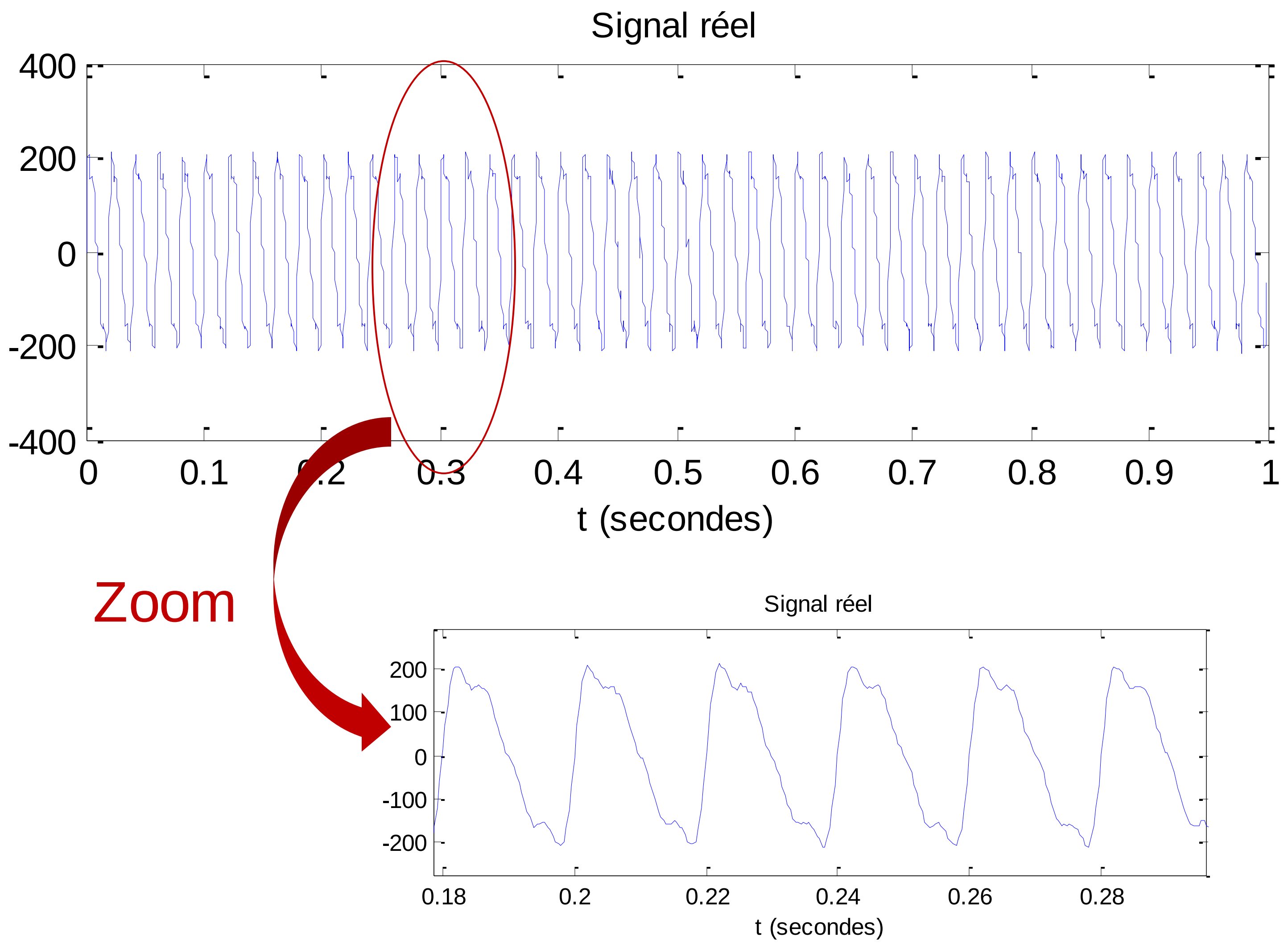
→ <u>Pour extraire de l'information des signaux :</u>

<u>Exemple 3</u>

- Identifier la bande de fréquence nécessaire à la transmission d'un signal,
- Détecter des anomalies, des défauts (ECG, Arcs électriques sur les câbles d'alimentation d'un avion, dent cassée dans un engrenage…)
- En éliminant des composantes indésirables : le bruit, certaines fréquences…

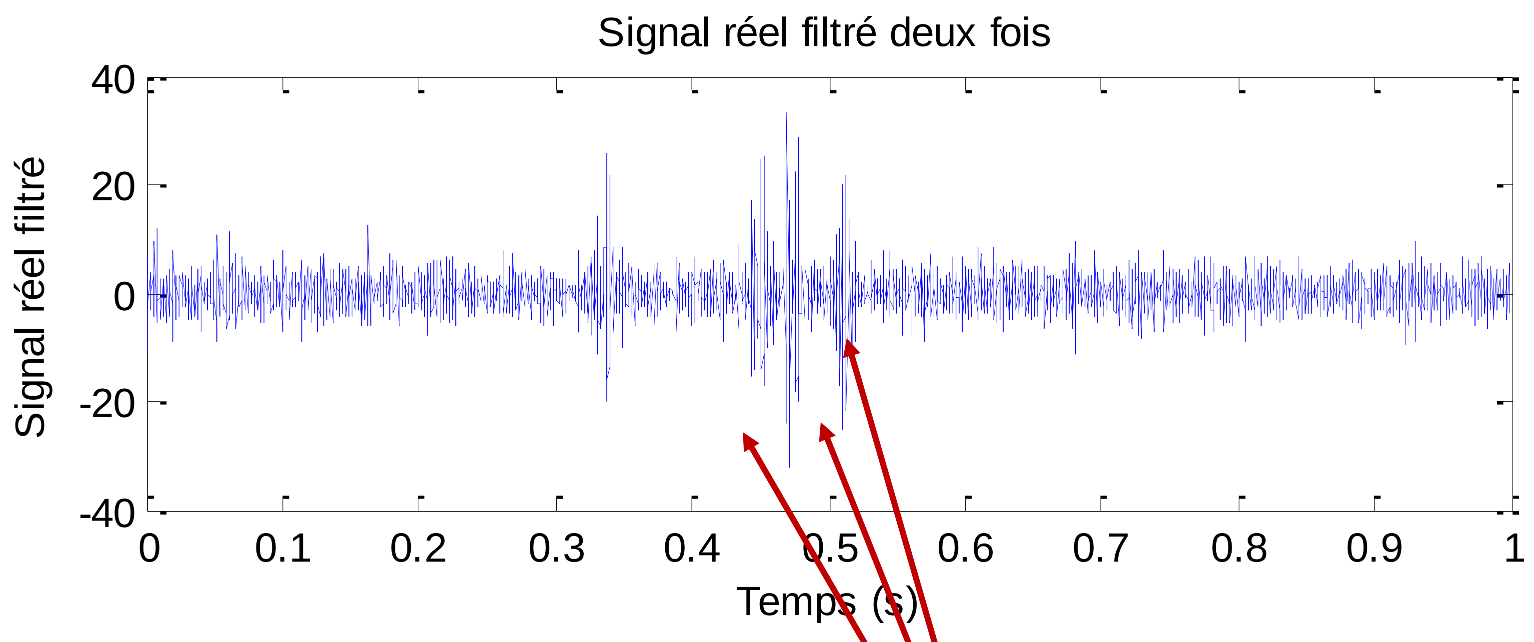
Détection de perturbations « annonciatrices »
(de fréquences > 500 Hz noyées dans le 50 Hz)

Signal d'alimentation contenant la perturbation



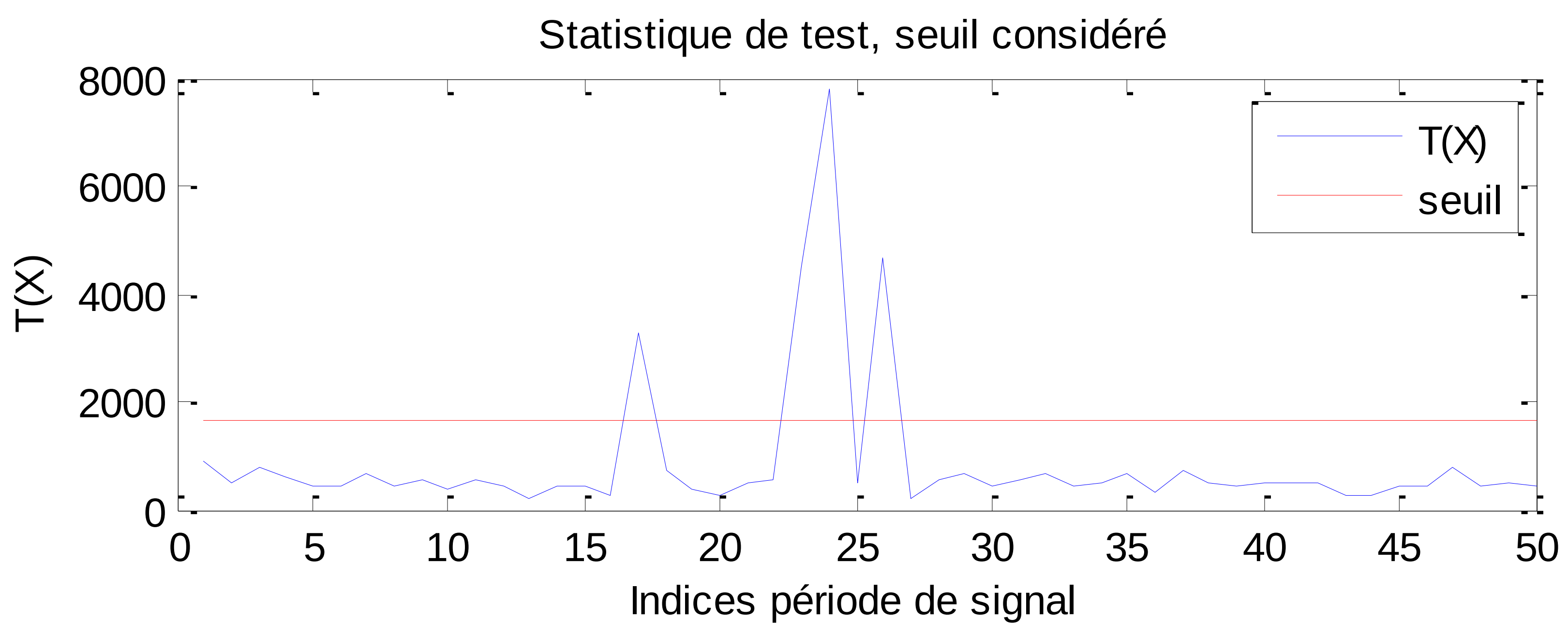
Filtrage

Signal après filtrage passe-haut (élimination du 50 Hz)



Transitoires à détecter

Détecteur d'énergie



Le traitement du signal a besoin d’outils…

Partie 1 : Signaux et Systèmes à temps continu

Cours 1 à 4

- Modèles de signaux,
- Outils pour la représentation et l’analyse de signaux :
  + Représentation fréquentielle ou « spectre » (Transformée de Fourier, Densité Spectrale de Puissance : DSP),
  + Fonctions d’inter et d’autocorrélation.
  + Filtrage (linéaire, non linéaire) des signaux à temps continu.

TD1

Etude de différentes modélisations d’un signal, calcul de fonctions d’autocorrélation et de spectres (TF, DSP)

TD2

Exercices sur le filtrage linéaire et non linéaire.

…qui doivent être implantés en numérique

Partie 2 : Signaux et Systèmes à temps discret

Cours 5 à 7

- Numérisation des signaux : échantillonnage, quantification.
- Numérisation des outils pour la représentation et l'analyse de signaux (Transformée de Fourier Discrète, DSP et fonctions d'inter et d'autocorrélation numériques).
- Définition et implantation de filtres numériques.

TD3

Etude de l'échantillonnage (impact, échantillonnage non idéal)

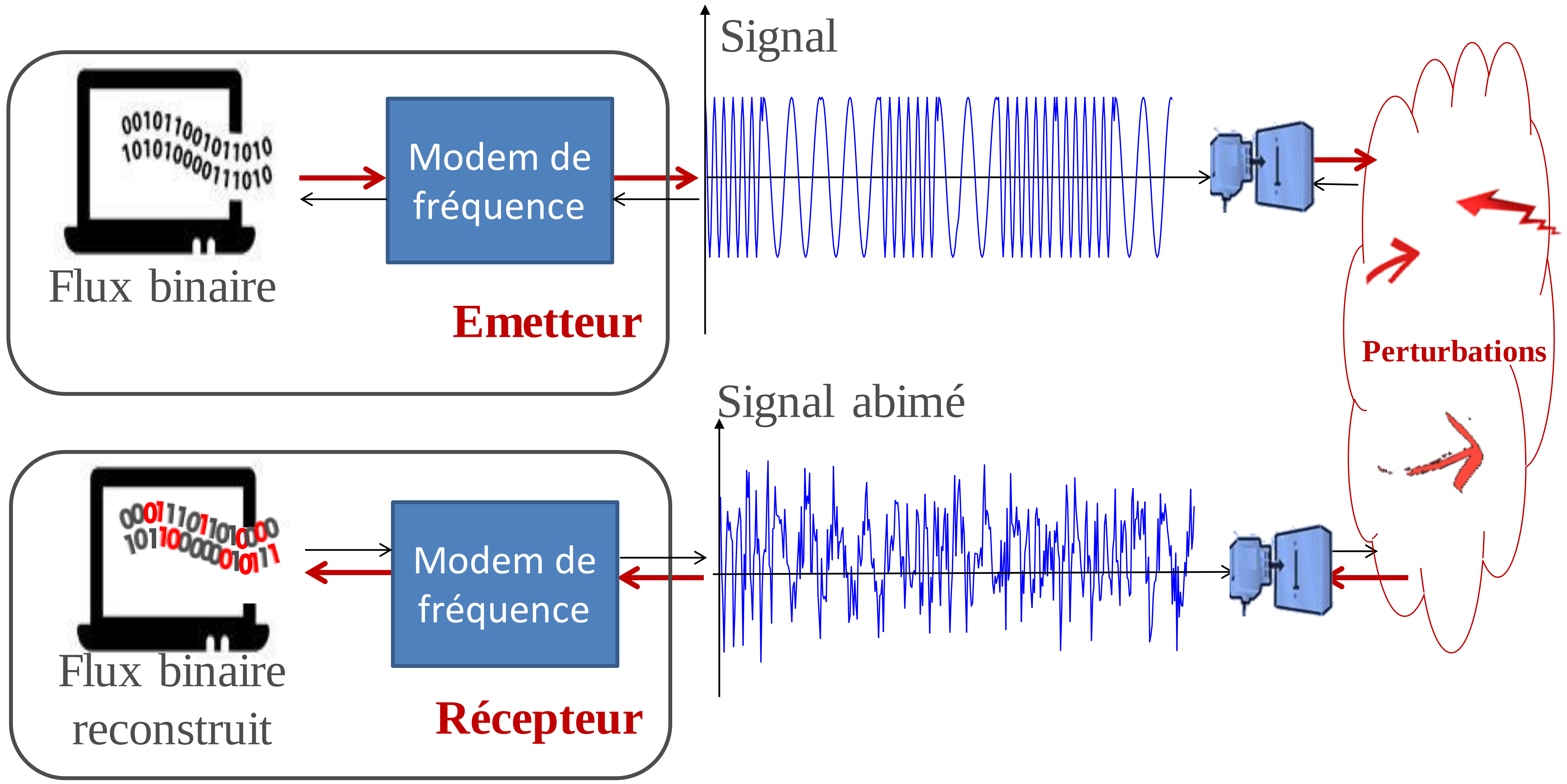
Mise en pratique (TPs et Projet)

Mise en place d'un modem de fréquence répondant à la recommandation V21 de l'Union Internationale des Télécommunications

Projet 2022 - 2023

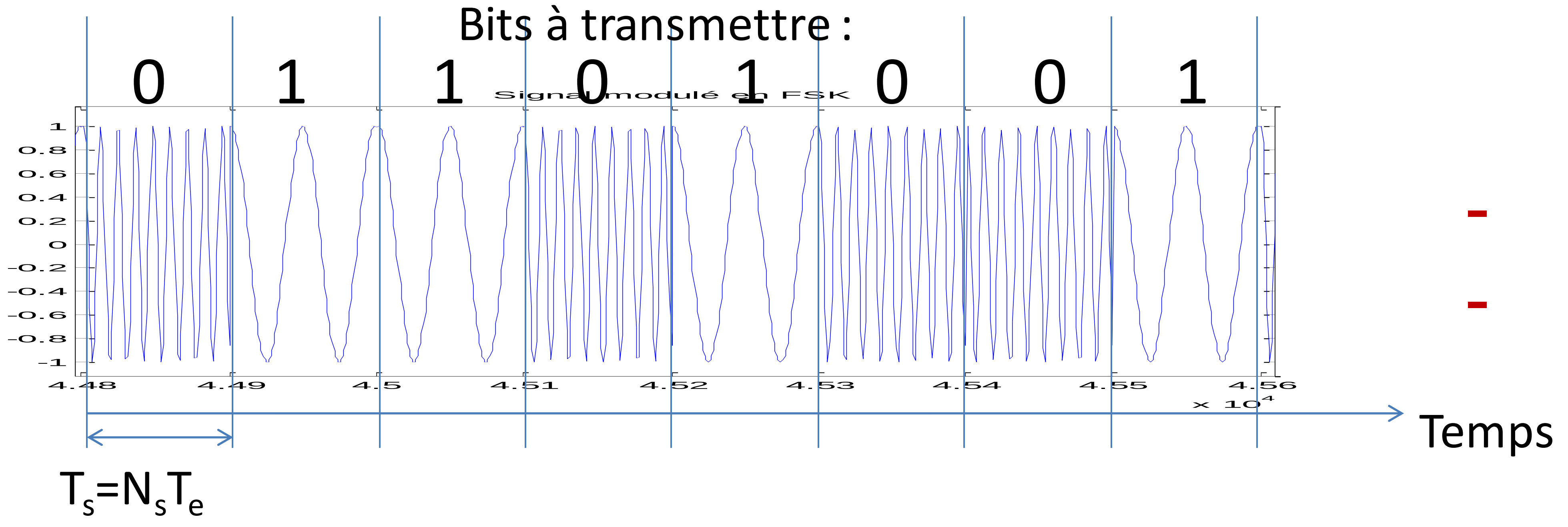
Réalisation d'un modem selon la recommandation V21 de l'ITU (Union Internationale des Télécommunication)

Modem de fréquence



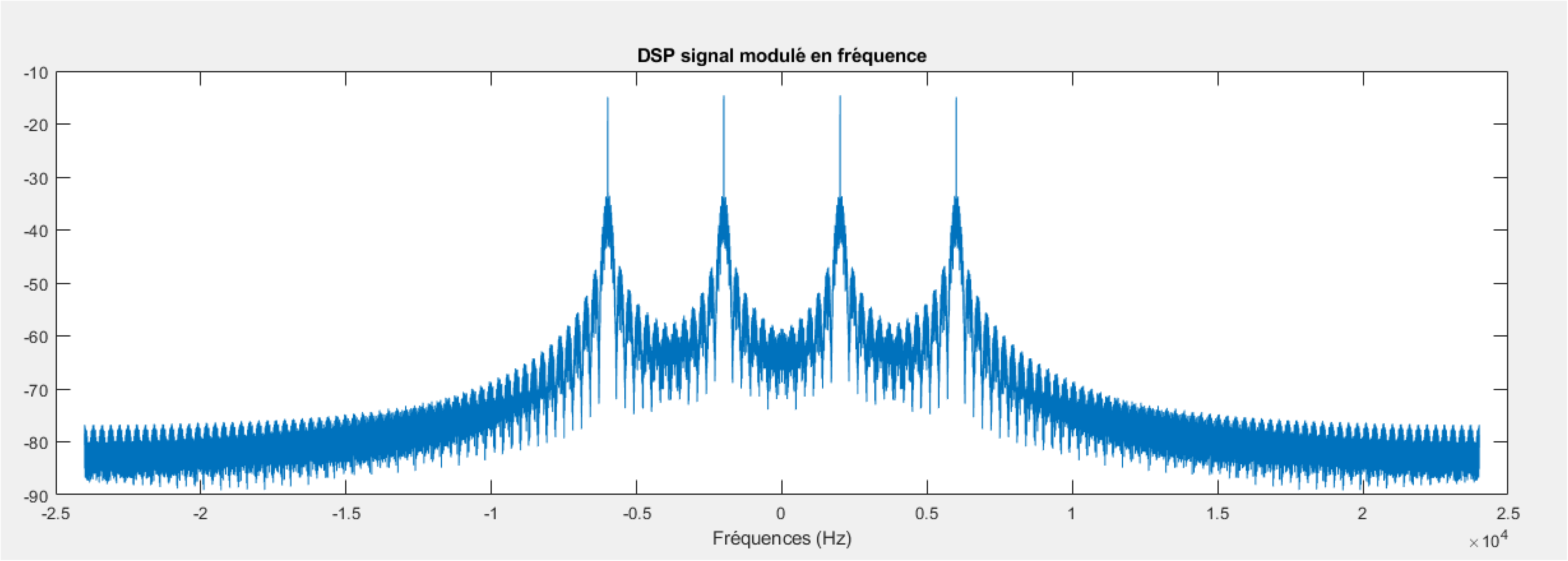
Projet 2022-2023

1- Construction du signal modulé en fréquence



- Génération d'un signal numérique
- Echelle temporelle en secondes

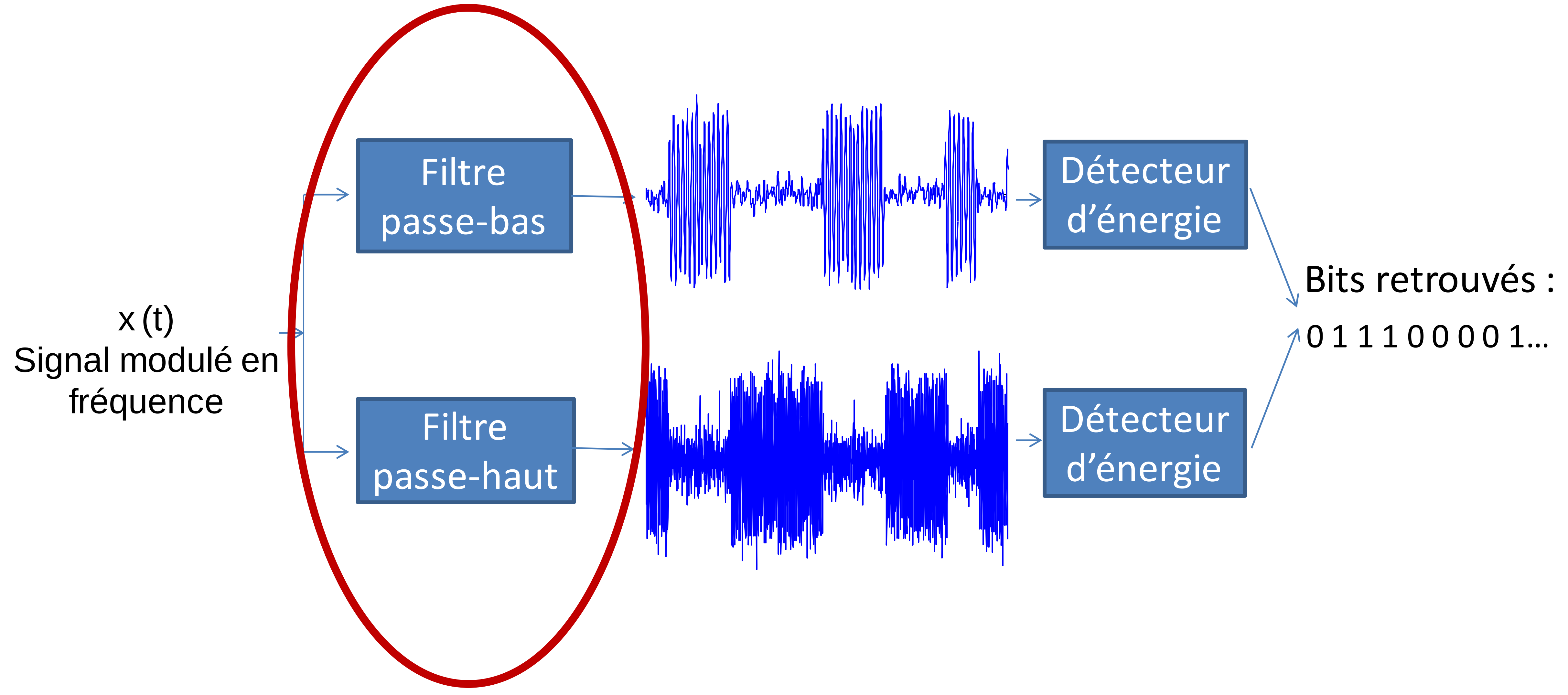
TFD ou DSP



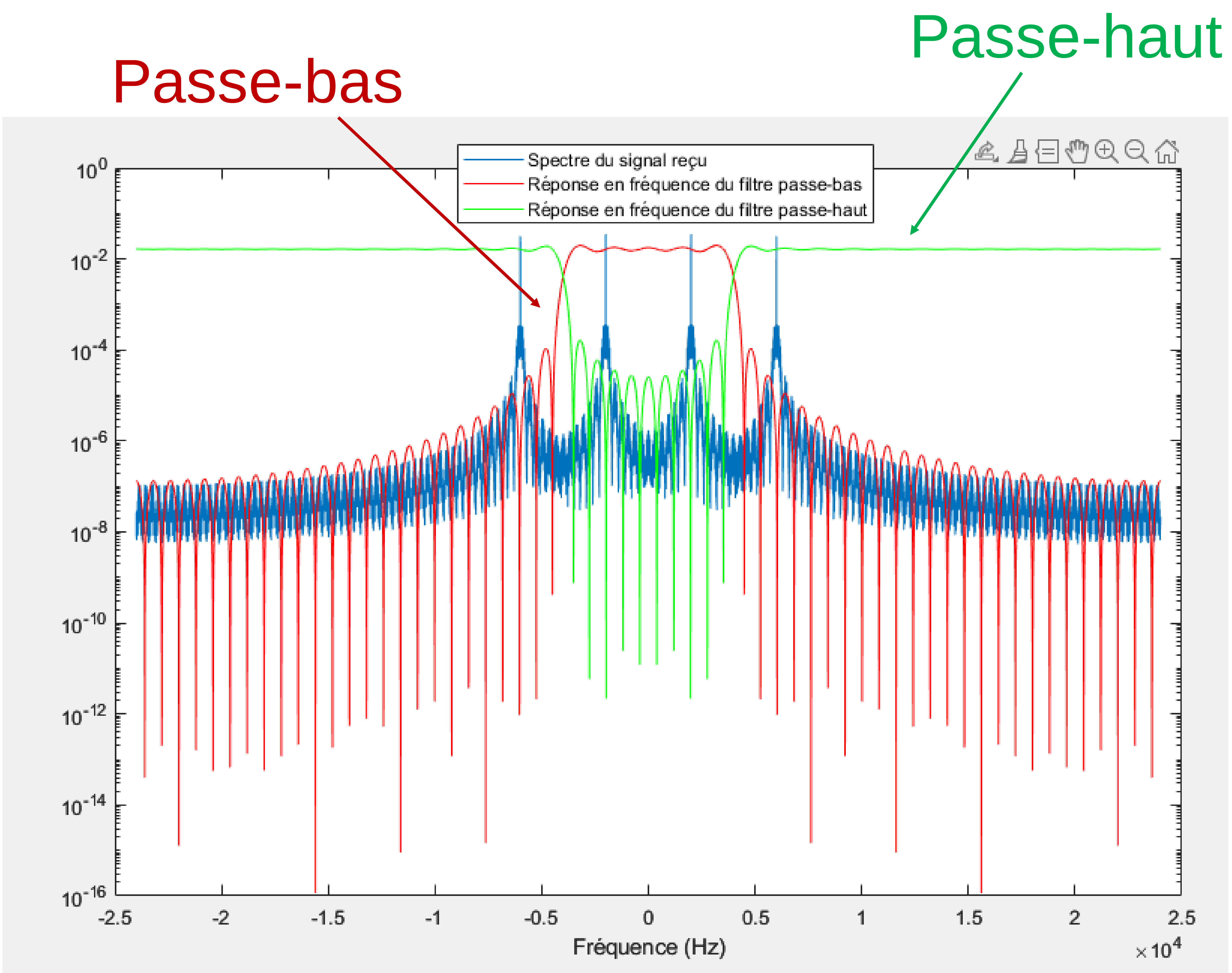
- TFD ou estimation de la DSP
- Echelle fréquentielle en Hz

Projet 2022-2023

2- Démodulation par filtrage

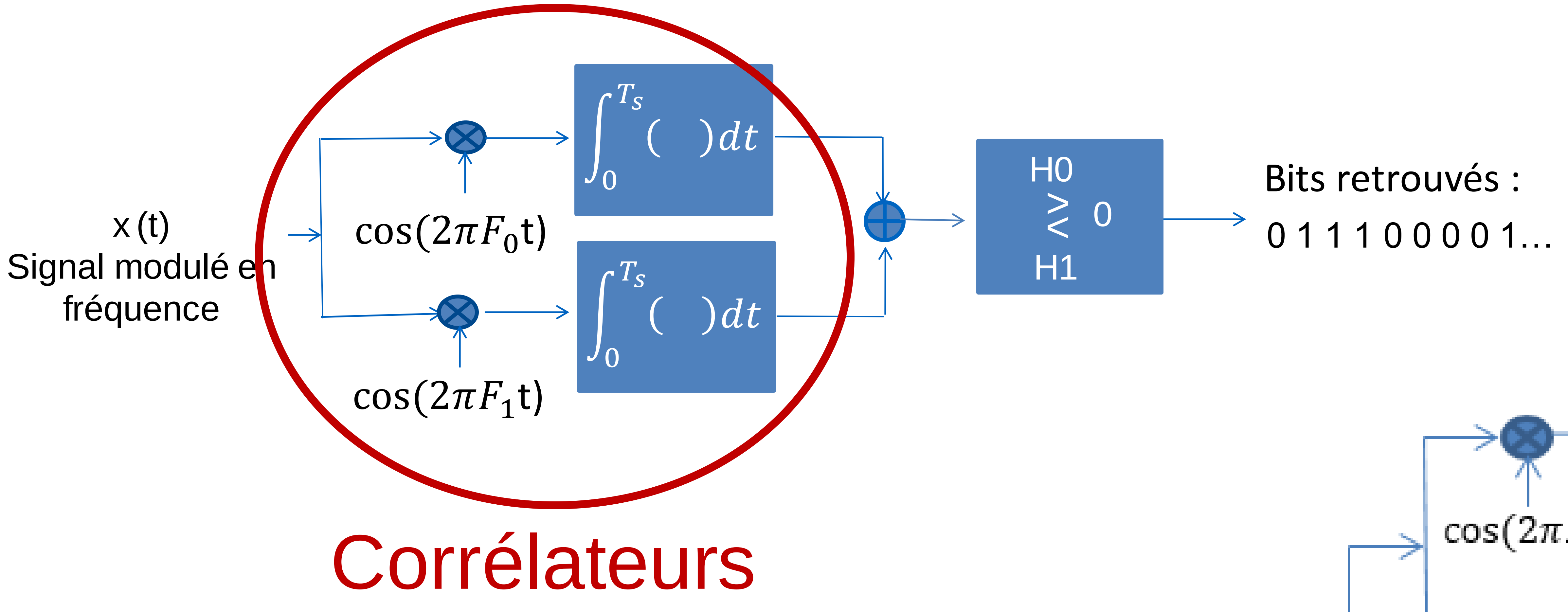


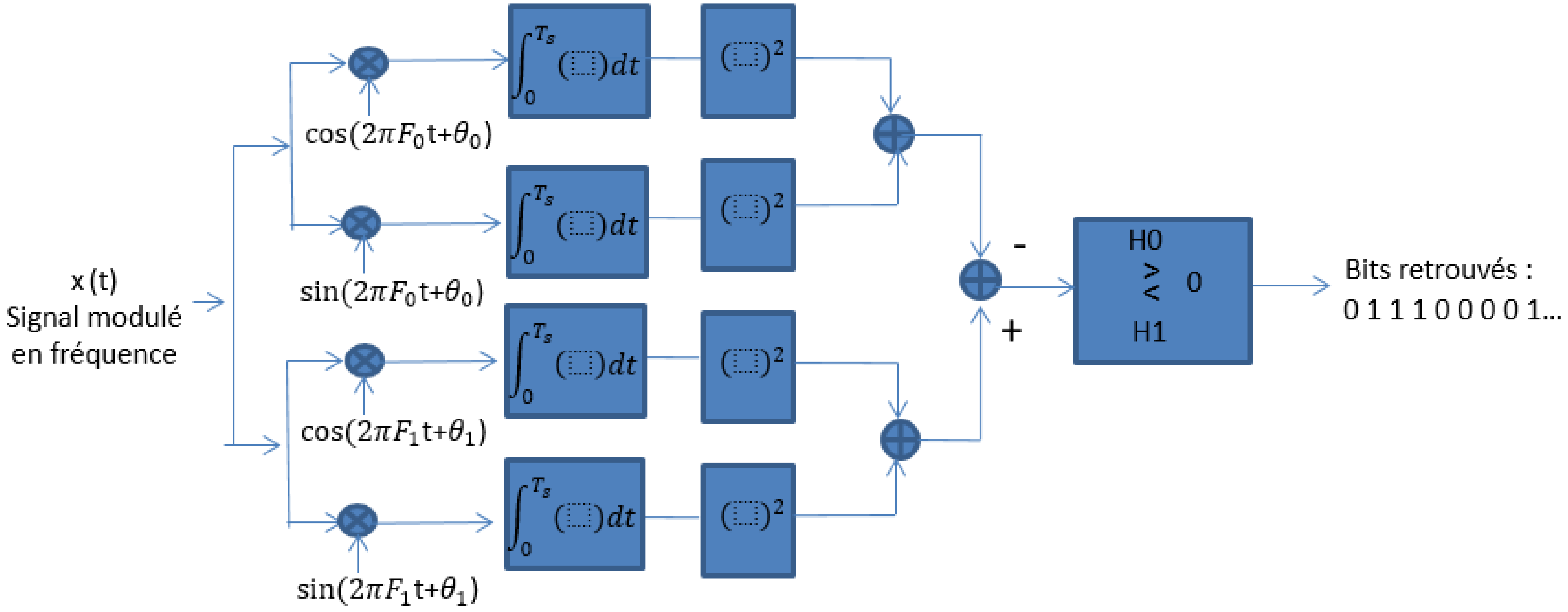
Synthèse de
Filtres Numériques



Projet 2022-2023

3- Démodulateur FSK, rec. V21, sans et avec gestion d'une erreur de synchronisation





- Découverte des images cachées, calcul du TEB -

Des traitements en temps réel ?

→ Temps de calcul en traitement numérique du signal :

Nombre d'opérations d'addition/multiplication

(MAC = Multiplication Accumulation)

→ Temps réel

y(n) est calculé avant que x(n+1) ne se présente

(T_e secondes entre deux x(n) et x(n+1))

$x(n) \longrightarrow$ Traitement numérique $\longrightarrow y(n)$

Exemples :

→ Estimation biaisée de la fonction d'autocorrélation de x:

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k) \times x^*(k-n), \;\; n = 0, ..., N-1$$

→ Transformée de Fourier Discrète (TFD) de x:

$$y(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi\frac{kn}{N}}, \;\; n = 0, ..., N-1$$

→ Filtrage numérique à réponse impulsionnelle finie de x :

$$y(n) = \sum_{k=0}^{N-1} b_k \times x(n-k), \;\; n = 0, ..., N-1$$

Rappels de traitement du signal

1- Signaux
2- Fonctions d'inter et d'auto corrélation
3- Transformée de Fourier
4- Densité spectrale de puissance
5- Filtres linéaires invariants dans le temps

Rappels de traitement du signal

1- Signaux
2- Fonctions d’inter et d’auto corrélation
3- Transformée de Fourier
4- Densité spectrale de puissance
5- Filtres linéaires invariants dans le temps

Classes de signaux

Deterministes

Classe 1 : Deterministes, à énergie finie

$$\mathsf{E} = \int_{\mathbb{R}} |x(t)|^2 dt = \int_{\mathbb{R}} |X(f)|^2 df < \infty$$

Classe 2 : Deterministes, à puissance moyenne finie périodiques

$$\mathsf{P} = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} |x(t)|^2 dt < \infty$$

Classe 3 : Deterministes, à puissance moyenne finie non périodiques

$$\mathsf{P} = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 dt < \infty$$

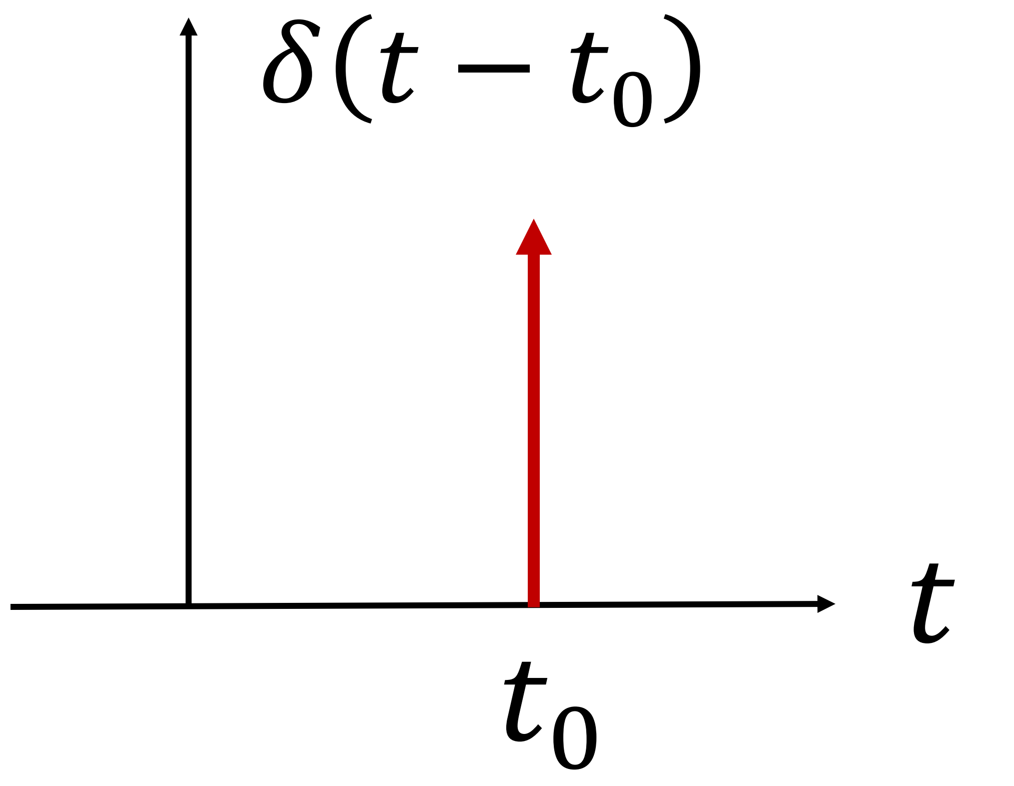
Aléatoires

Classe 4 : Aléatoires et stationnaires

Définis par des propriétés statistiques indépendantes du temps

Distributions

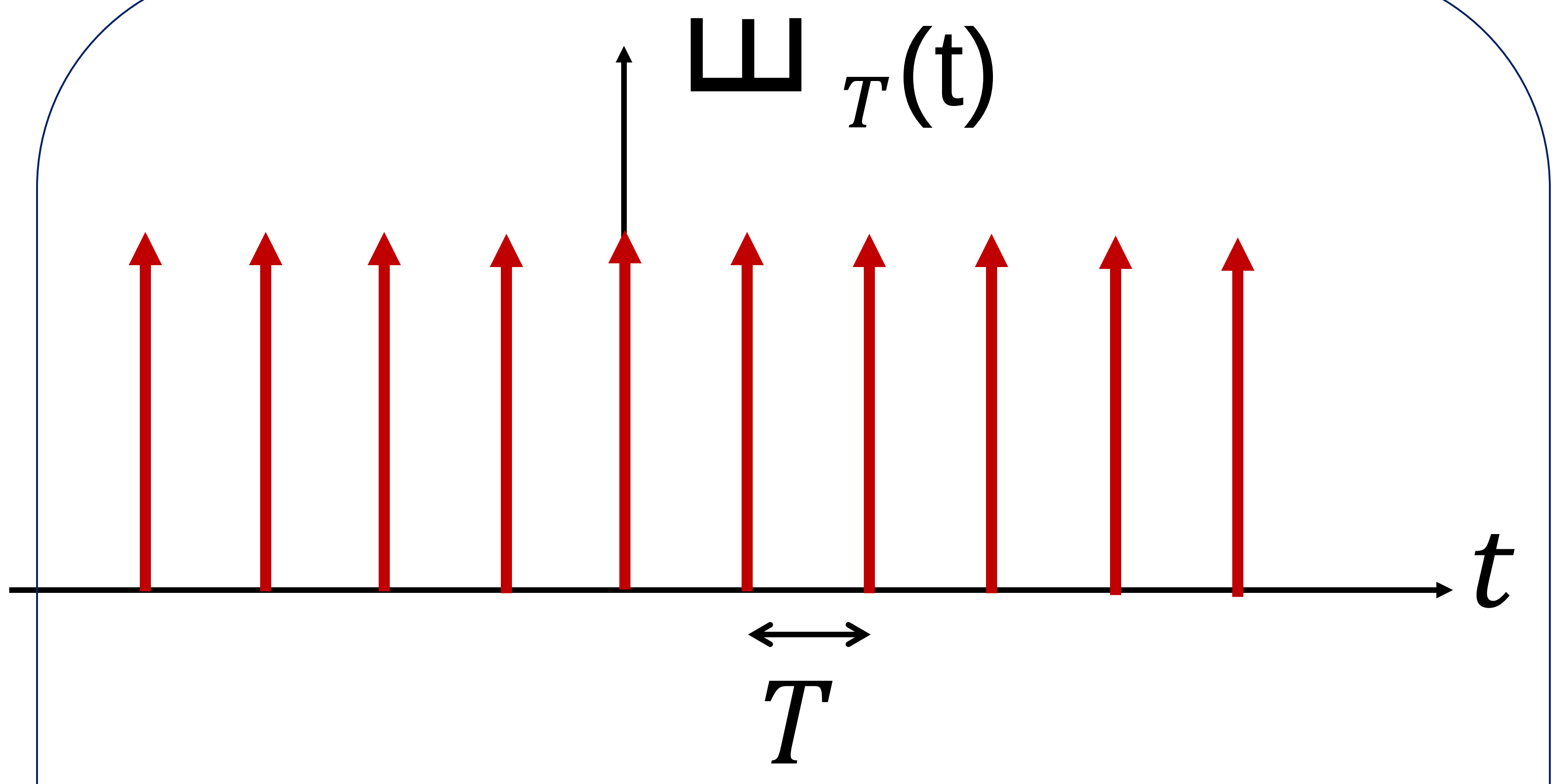
Dirac



$$\delta(t-t_0)=\begin{cases}0 \; si \; t \neq t_0 \\ \infty \; si \; t = t_0\end{cases}$$

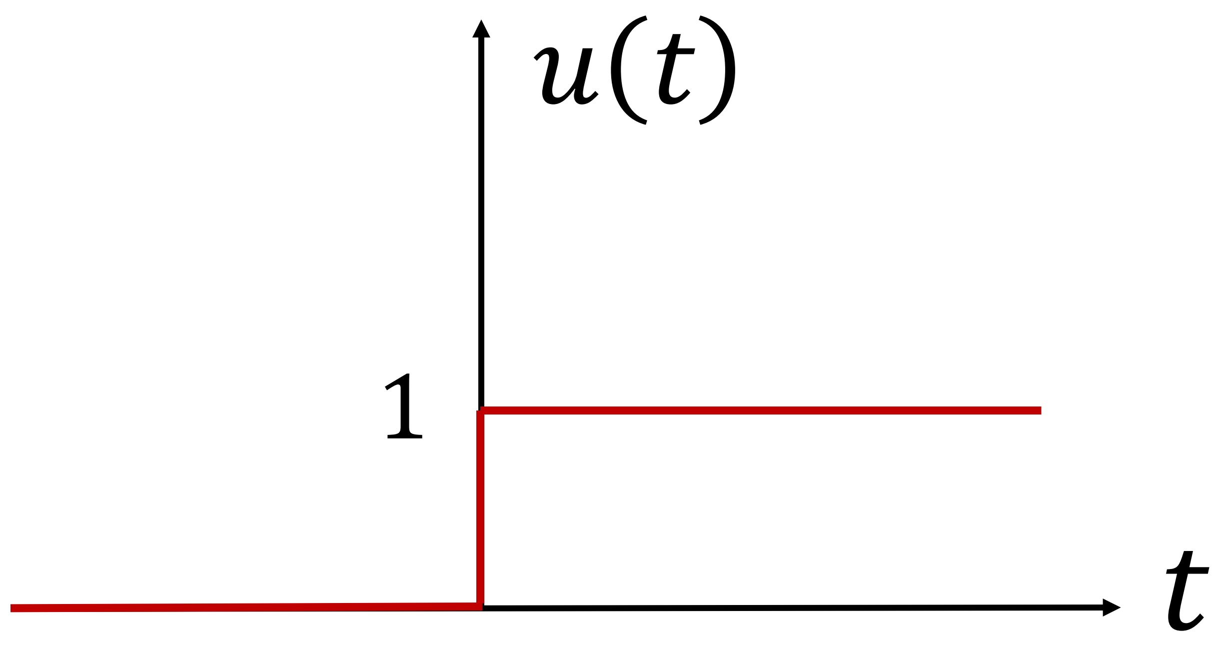
$$\int_{-\infty}^{+\infty} \delta(t-t_0)dt = 1$$

Peigne de Dirac



$$\text{Ш}_T(t)=\sum_{k=-\infty}^{+\infty} \delta(t-kT)$$

Fonction de Heaviside (ou échelon unité)



$$u(t)=\begin{cases}1 \; pour \; t>0 \\ 0 \; pour \; t<0\end{cases}$$

Rque : $\begin{cases} x(t)\delta(t-t_0) = x(t_0)\delta(t-t_0) \\ x(t) * \delta(t-t_0) = x(t-t_0)\end{cases}$

Rappels de traitement du signal

1- Signaux
2- Fonctions d’inter et d’auto corrélation
3- Transformée de Fourier
4- Densité spectrale de puissance
5- Filtres linéaires invariants dans le temps

Définitions

Fonctions d'intercorrélation | **Fonctions d'autocorrélation**

Classe 1 : Signaux déterministes à énergie finie

$$R_{xy}(\tau) = \int_{\mathbb{R}} x(t)y^*(t-\tau)dt \qquad R_x(\tau) = \int_{\mathbb{R}} x(t)x^*(t-\tau)dt$$

Classe 2 : Signaux déterministes à puissance moyenne finie périodiques

$$R_{xy}(\tau) = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} x(t)y^*(t-\tau)dt \qquad R_x(\tau) = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} x(t)x^*(t-\tau)dt$$

Classe 3 : Signaux déterministes à puissance moyenne finie non périodiques

$$R_{xy}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t)y^*(t-\tau)dt \qquad R_x(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t)x^*(t-\tau)dt$$

Class 4 : Signaux aléatoires stationnaires

$$R_{xy}(\tau) = E[x(t)y^*(t-\tau)] \qquad R_x(\tau) = E[x(t)x^*(t-\tau)]$$

Quelques propriétés

Symétrie Hermitienne

$$R_x^*(-\tau) = R_x(\tau)$$

Valeur max

$$|R_x(\tau)| \leq R_x(0)$$

Distance antre x(t) et x(t-τ) quand x est réel

$$d^2\left[x(t), x(t-\tau)\right] = 2\left[R_x(0) - R_x(\tau)\right]$$

Rappels de traitement du signal

1- Signaux
2- Fonctions d’inter et d’auto corrélation
3- Transformée de Fourier
4- Densité spectrale de puissance
5- Filtres linéaires invariants dans le temps

Transformée de Fourier

Définition

$$X(f) = \int_{\mathbb{R}} x(t) \exp\left(-j2\pi f t\right) dt$$

$$x(t) = \int_{\mathbb{R}} X(f) \exp\left(j2\pi f t\right) df$$

Propriétés

Linéarité $\text{TF}\left[ax(t) + by(t)\right] = aX(f) + bY(f)$

Translation $\text{TF}\left[x(t-t_0)\right] = \exp(-j2\pi f t_0)X(f)$

Modulation $\text{TF}\left[x(t)\exp(j2\pi f_0 t)\right] = X(f-f_0)$

La TF transforme un produit en produit de convolution $\begin{cases} \text{TF}\left[x(t) * y(t)\right] = X(f)Y(f) \\ \text{TF}\left[x(t)y(t)\right] = X(f) * Y(f) \end{cases}$

Egalité de Parseval $\int_{\mathbb{R}} x(t)y^*(t)dt = \int_{\mathbb{R}} X(f)Y^*(f)df$

Tables de Transformée de Fourier

$x(t)$	$\xrightarrow{TF}$ / $\xleftarrow{TF^{-1}}$	$X(f)$
$ax(t)+by(t)$	$\rightleftharpoons$	$aX(f)+bY(f)$
$x(t-t_0)$	$\rightleftharpoons$	$X(f)e^{-i2\pi f t_0}$
$x(t)e^{+i2\pi f_0 t}$	$\rightleftharpoons$	$X(f-f_0)$
$x^*(t)$	$\rightleftharpoons$	$X^*(-f)$
$x(t)\ .\ y(t)$	$\rightleftharpoons$	$X(f) * Y(f)$
$x(t) * y(t)$	$\rightleftharpoons$	$X(f)\ .\ Y(f)$
$x(at+b)$	$\rightleftharpoons$	$\frac{1}{\lvert a\rvert}X\left(\frac{f}{a}\right)e^{i2\pi\frac{b}{a}f}$
$\frac{dx^{(n)}(t)}{dt^n}$	$\rightleftharpoons$	$(i2\pi f)^n X(f)$
$(-i2\pi t)^n x(t)$	$\rightleftharpoons$	$\frac{dX^{(n)}(f)}{df^n}$

TF ; TF^{-1}

$x(t)$	$\xrightarrow{TF}$ / $\xleftarrow{TF^{-1}}$	$X(f)$
1	$\rightleftharpoons$	$\delta(f)$
$\delta(t)$	$\rightleftharpoons$	1
$e^{+i2\pi f_0 t}$	$\rightleftharpoons$	$\delta(f-f_0)$
$\delta(t-t_0)$	$\rightleftharpoons$	$e^{-i2\pi f t_0}$
$Ш_T(t)=\sum_{k\in\mathbb{Z}}\delta(t-kT)$	$\rightleftharpoons$	$\frac{1}{T}Ш_{1/T}(f)$
$\cos(2\pi f_0 t)$	$\rightleftharpoons$	$\frac{1}{2}[\delta(f-f_0)+\delta(f+f_0)]$
$\sin(2\pi f_0 t)$	$\rightleftharpoons$	$\frac{1}{2i}[\delta(f-f_0)-\delta(f+f_0)]$
$e^{-a\lvert t\rvert}$	$\rightleftharpoons$	$\frac{2a}{a^2+4\pi^2 f^2}$
$e^{-\pi t^2}$	$\rightleftharpoons$	$e^{-\pi f^2}$
$\Pi_T(t)$	$\rightleftharpoons$	$T\frac{\sin(\pi T f)}{\pi T f}=T\sin c(\pi T f)$
$\Lambda_T(t)$	$\rightleftharpoons$	$T\sin c^2(\pi T f)$
$B\sin c(\pi B t)$	$\rightleftharpoons$	$\Pi_B(f)$
$B\sin c^2(\pi B t)$	$\rightleftharpoons$	$\Lambda_B(f)$

Rappels de traitement du signal

1- Signaux
2- Fonctions d’inter et d’auto corrélation
3- Transformée de Fourier
4- Densité spectrale de puissance
5- Filtres linéaires invariants dans le temps

Densité Spectrale de Puissance (DSP)

Définition

$$s_x(f) = \mathsf{TF}\left[R_x(\tau)\right]$$

Propriétés

→ $S_x(f)$ est réelle
→ Si x est réel alors $S_x(f)$ est réelle et paire
→ $S_x(f)$ est positive

Densité Spectrale de Puissance (DSP)

Classe 1 : Signaux déterministes à énergie finie

$$s_x(f) = |X(f)|^2$$

Classe 2 : Signaux déterministes à puissance moyenne finie périodiques

$$s_x(f) = \sum_{k \in \mathbb{Z}} |c_k|^2 \delta(f - k f_0) \quad \text{avec} \quad x(t) = \sum_{k \in \mathbb{Z}} c_k \exp(j 2\pi k f_0 t)$$

Classe 3 : Signaux déterministes à puissance moyenne finie non périodiques

$$s_x(f) = \lim_{T \to \infty} \frac{1}{T} |X_T(f)|^2 \quad \text{avec} \quad X_T(f) = \int_{-T/2}^{T/2} x(t) \exp(-j 2\pi f t) dt$$

Class 4 : Signaux aléatoires stationnaires

$$s_x(f) = \text{TF}\left[R_x(\tau)\right]$$

Puissance/Energie d'un signal

$$P \text{ ou } E = R_x(0) = \int_{\mathbb{R}} s_x(f)df$$

Classe 1 : x déterministe à énergie finie

$$\int_{-\infty}^{+\infty} |x(t)|^2 dt = \int_{-\infty}^{+\infty} |X(f)|^2 df$$

Classe 2 : x déterministe à puissance finie périodique

$$\frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} |x(t)|^2 dt$$

Classe 3 : x déterministe à puissance finie non périodique

$$\lim_{t \to \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |x(t)|^2 dt$$

Classe 4 : x aléatoire stationnaire

$$E[|x(t)|^2]$$

Rappels de traitement du signal

1- Signaux
2- Fonctions d’inter et d’auto corrélation
3- Transformée de Fourier
4- Densité spectrale de puissance
5- Filtres linéaires invariants dans le temps

Filtres linéaires invariants dans le temps

$$x(t) \longrightarrow \boxed{T}$$

$$x(t) \;\xrightarrow{TF}\; X(f)$$

$$y(t) = h(t) * x(t)$$

$$TF \downarrow \; \uparrow TF^{-1}$$

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

Pondération de la TF du signal d'entrée

$$y(t) = h(t) * x(t) = \int_{\mathbb{R}} h(u)x(t-u)du = \int_{\mathbb{R}} x(u)h(t-u)du = x(t) * h(t)$$

Linéarité

$$T\left[a_1 x_1(t) + a_2 x_2(t)\right] = a_1 T\left[x_1(t)\right] + a_2 T\left[x_2(t)\right]$$

Invariance dans le temps

Si $y(t) = T\left[x(t)\right]$ alors $T\left[x(t-t_0)\right] = y(t-t_0)$

Filtres linéaires invariants dans le temps

$$x(t) \longrightarrow \boxed{h(t) \underset{FT^{-1}}{\overset{FT}{\rightleftarrows}} H(f)} \longrightarrow y(t) = h(t) * x(t)$$

$$x(t) \xrightarrow{TF} X(f) \qquad y(t) \underset{TF^{-1}}{\overset{TF}{\rightleftarrows}} Y(f) = H(f).X(f)$$

!! Signaux déterministes !!

Définis par

$$\begin{cases} h(t) : réponse\ impulsionnelle \\ H(f) = TF[h(t)] : réponse\ en\ frequence \end{cases}$$

Conditions de réalisabilité

→ Causalité : $h(t) = 0\ pour\ t < 0$

→ Stabilité : $\int_{\mathbb{R}} |h(t)|\, dt < +\infty$

→ Réponse impulsionnelle réelle : $h(t) \in \mathbb{R}$

Mais aussi

$$R_{xy}(\tau) = R_x(\tau) * h(\tau)$$

$$R_y(\tau) = R_x(\tau) * h(\tau) * h^*(-\tau)$$

$$S_y(f) = S_x(f)\,|H(f)|^2$$

Relations de Wiener

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d’inter et d’auto corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d'inter et d'auto corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

Numérisation du signal

Signal analogique :
signal défini à tout instant par des valeurs réelles

Signal numérique :
signal défini à des instants discrets par un nombre fini de valeurs

Numérisation du signal

Signal analogique :
signal défini à tout instant par des valeurs réelles

↓

Numérisation :
Échantillonnage + quantification

↓

Signal numérique :
signal défini à des instants discrets par un nombre fini de valeurs

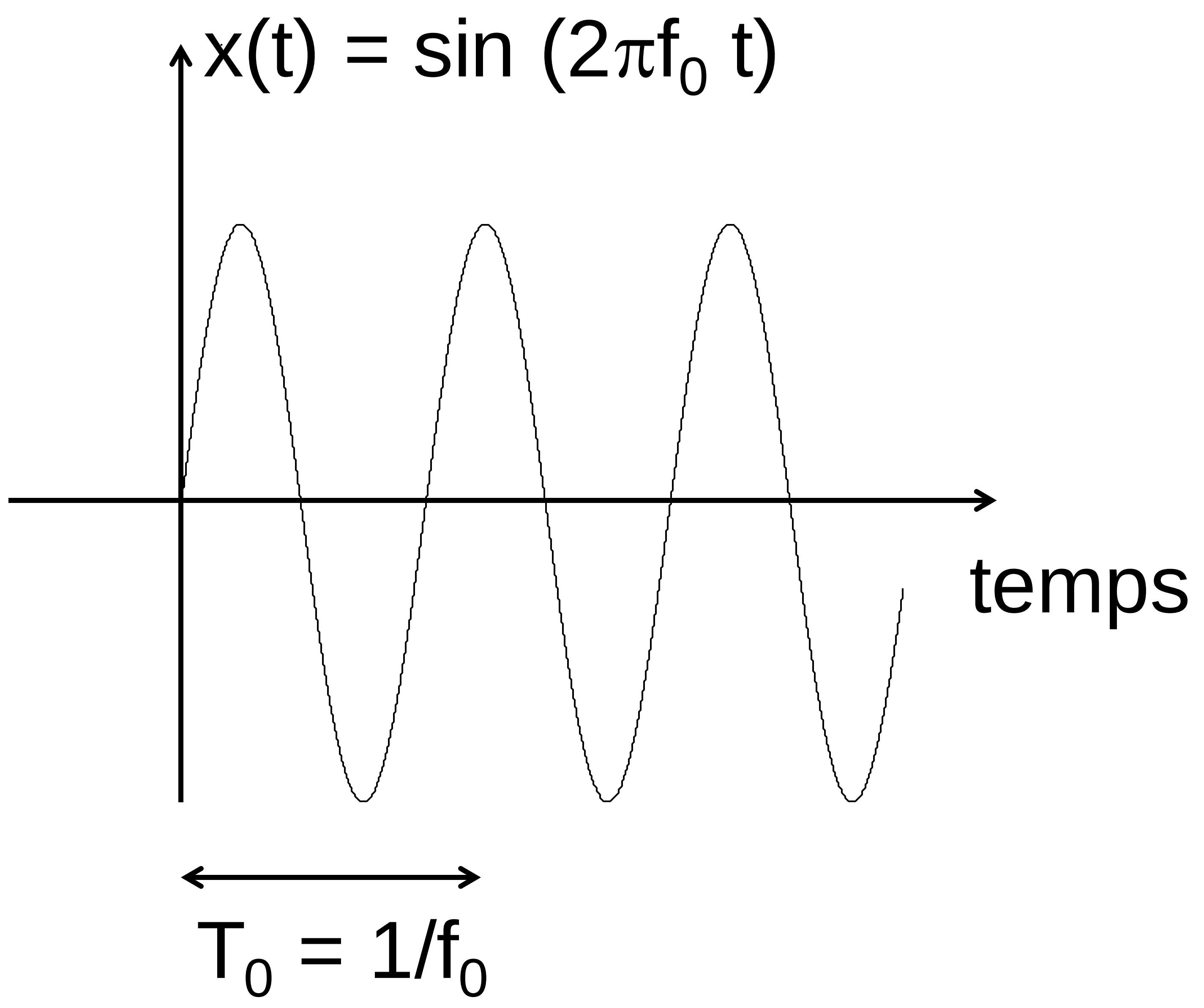
Numérisation du signal : Echantillonnage

Signal échantillonné :
signal défini à des instants discrets par des valeurs réelles

Numérisation du signal : Echantillonnage

Signal échantillonné :
signal défini à des instants discrets par des valeurs réelles

Exemple :



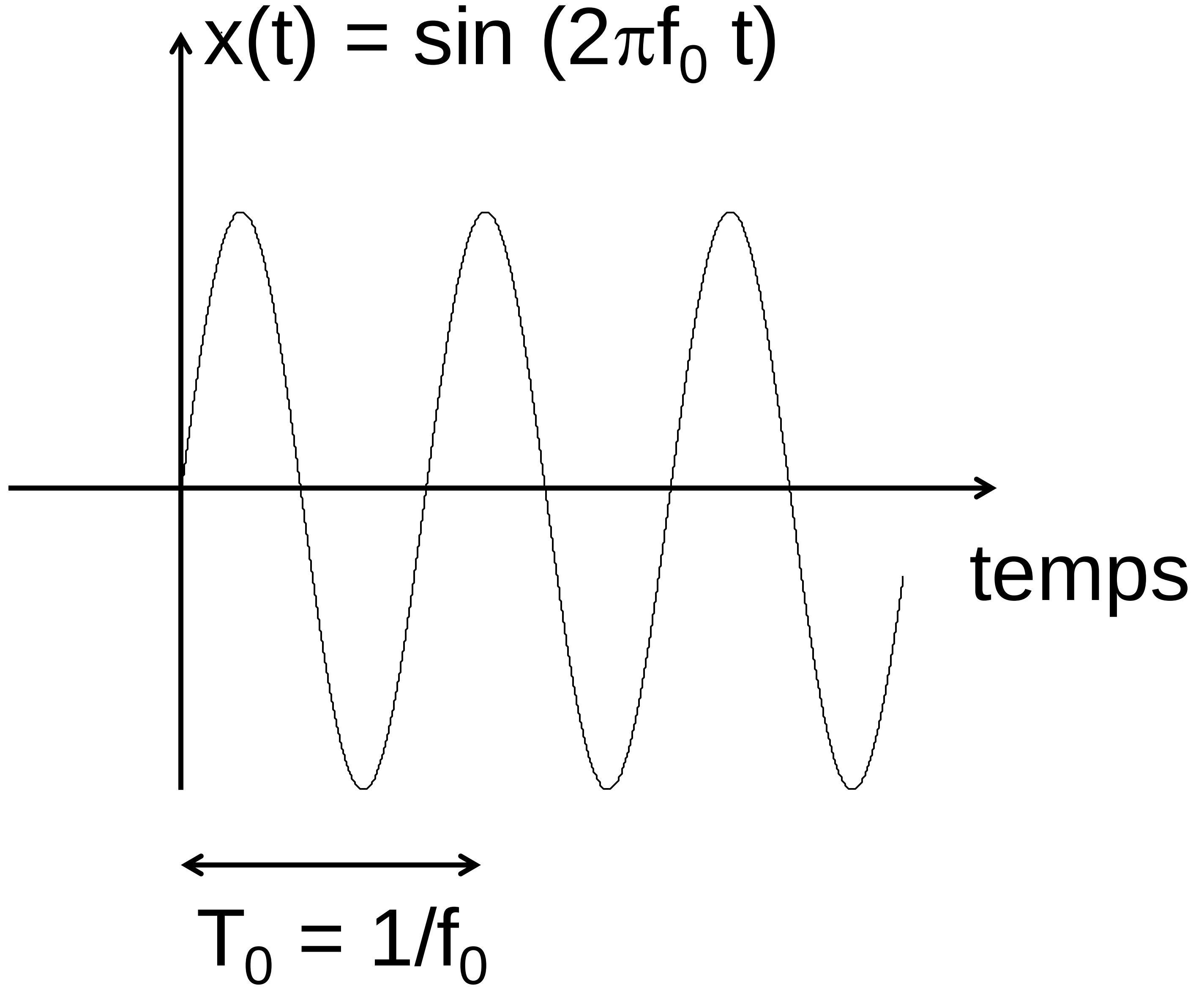
Numérisation du signal : Echantillonnage

Signal échantillonné :
signal défini à des instants discrets par des valeurs réelles

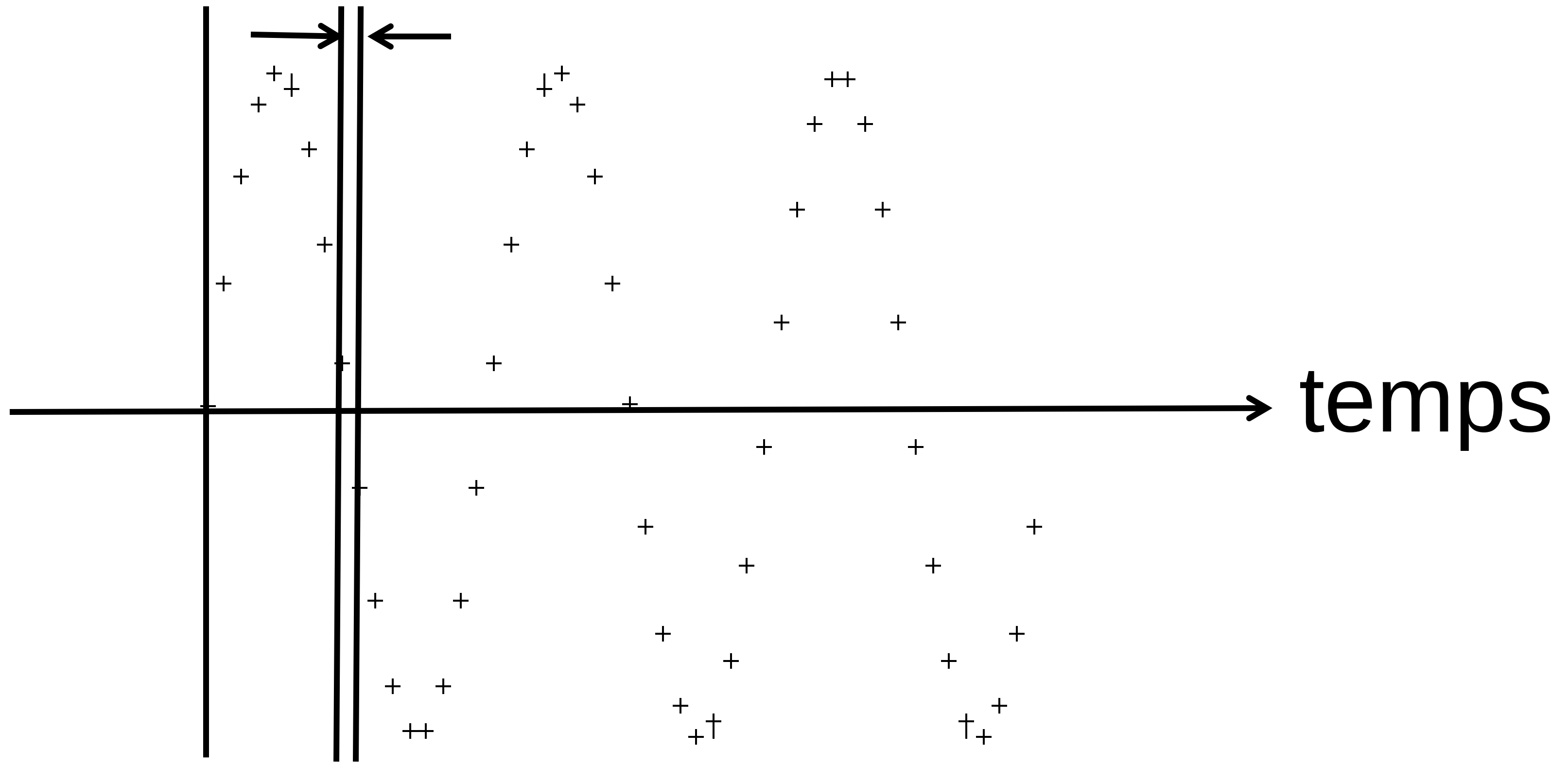
<u>Exemple :</u>

k = indice de l'échantillon

$$x(kT_e) = \sin(2\pi f_0\, kT_e)$$



T_e : période d'échantillonnage



Numérisation du signal : Echantillonnage

Est-il possible de conserver toute l’information dans la suite d’échantillons prélevés ?

Numérisation du signal : Echantillonnage

Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\,\text{Ш}_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * \text{Ш}_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Numérisation du signal : Echantillonnage

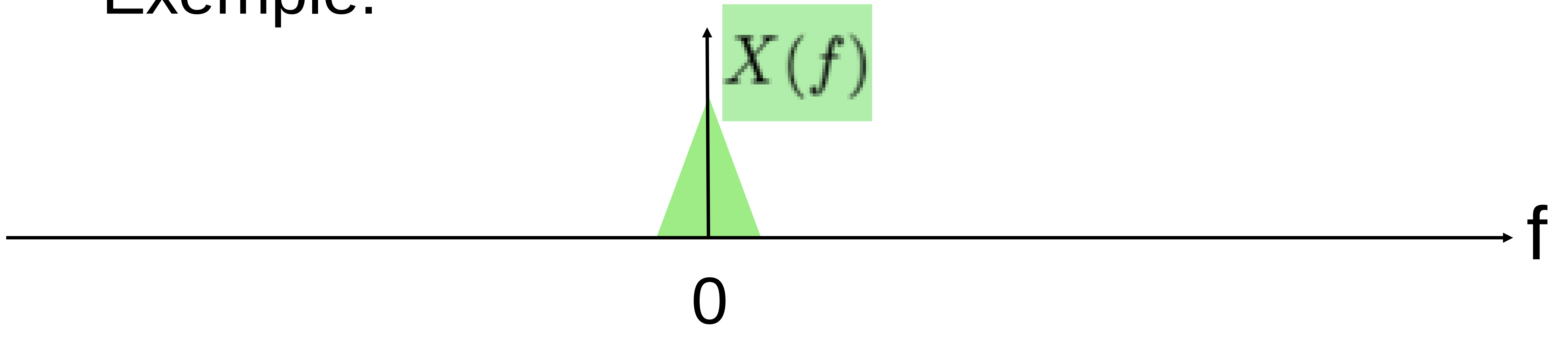
Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\,\text{Ш}_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * \text{Ш}_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Exemple:



Numérisation du signal : Echantillonnage

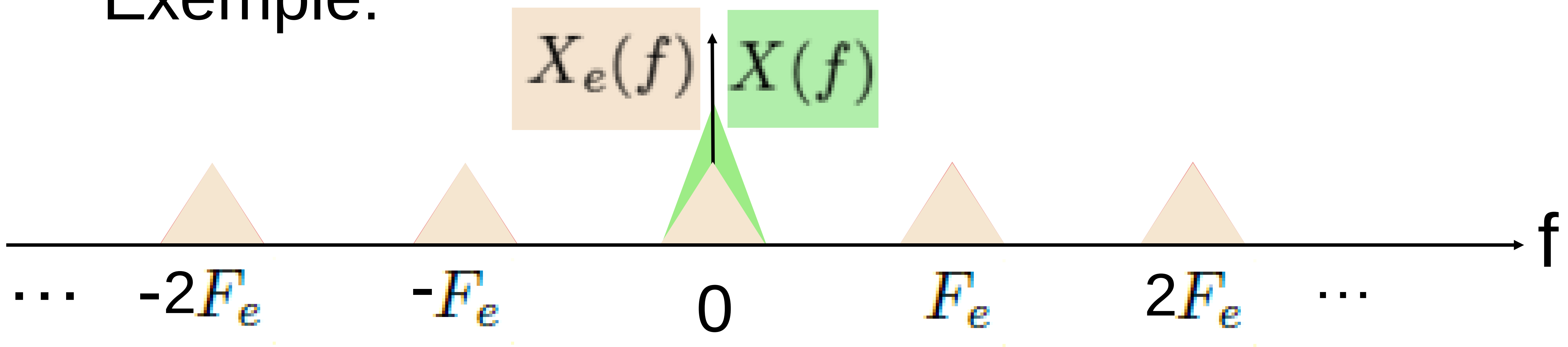
Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\,\text{Ш}_{T_e}(t)$$

$$X_e(\text{f})=\frac{1}{T_e}X(f) * \text{Ш}_{\frac{1}{T_e}}(f) = F_e \sum_n X(f-kF_e)$$

Exemple:



Numérisation du signal : Echantillonnage

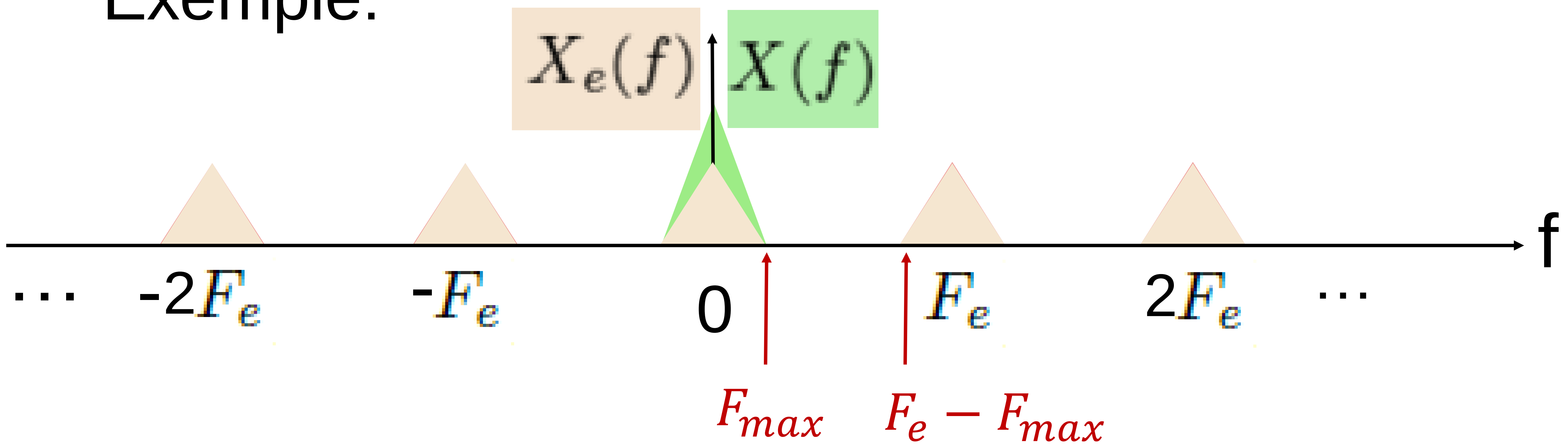
Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\,\text{Ш}_{T_e}(t)$$

$$X_e(\text{f}) = \frac{1}{T_e} X(f) * \text{Ш}_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Exemple:



Numérisation du signal : Echantillonnage

Est-il possible de conserver toute l'information dans la suite d'échantillons prélevés ?

Modélisation de l'échantillonnage :

$$x_e(t) = x(t)\, \text{III}_{T_e}(t)$$

$$X_e(f) = \frac{1}{T_e} X(f) * \text{III}_{\frac{1}{T_e}}(f) = F_e \sum_n X(f - kF_e)$$

Oui ça l'est !

A une condition :

$$F_e = \frac{1}{T_e} > 2F_{max}$$

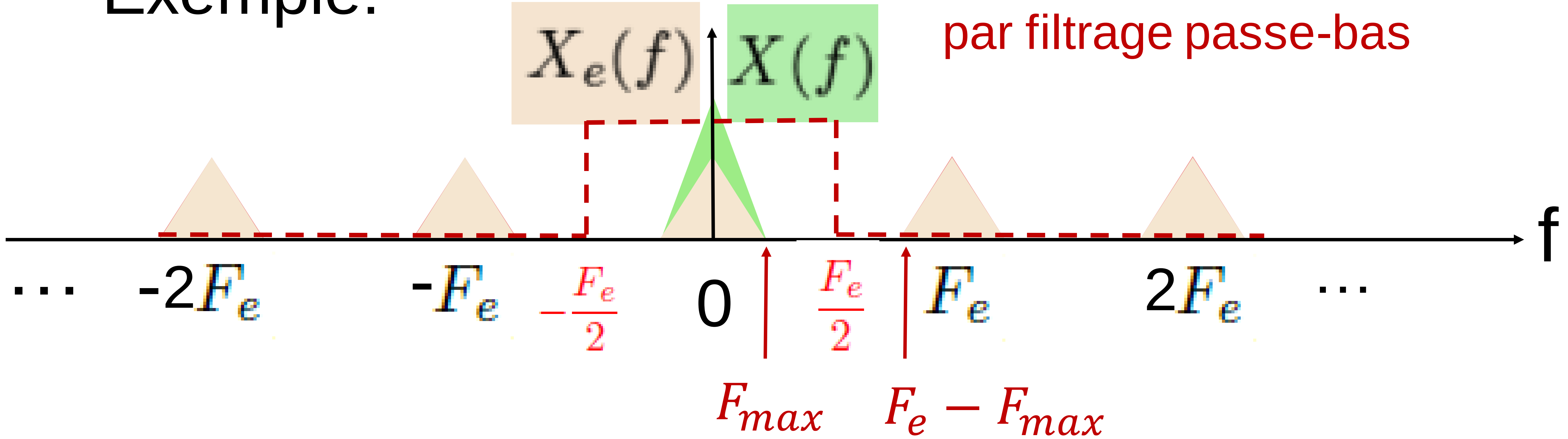
Condition de Shannon

Oberwolfach Photo Collection

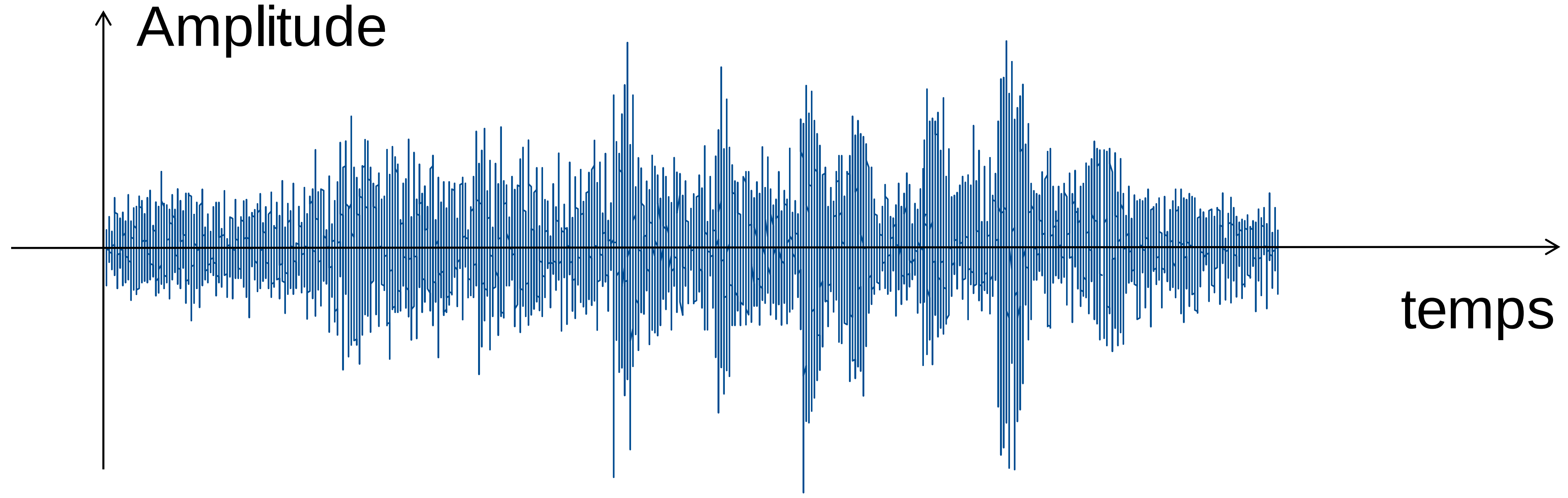
Claude Elwood Shannon (1916 -2001)

Exemple:

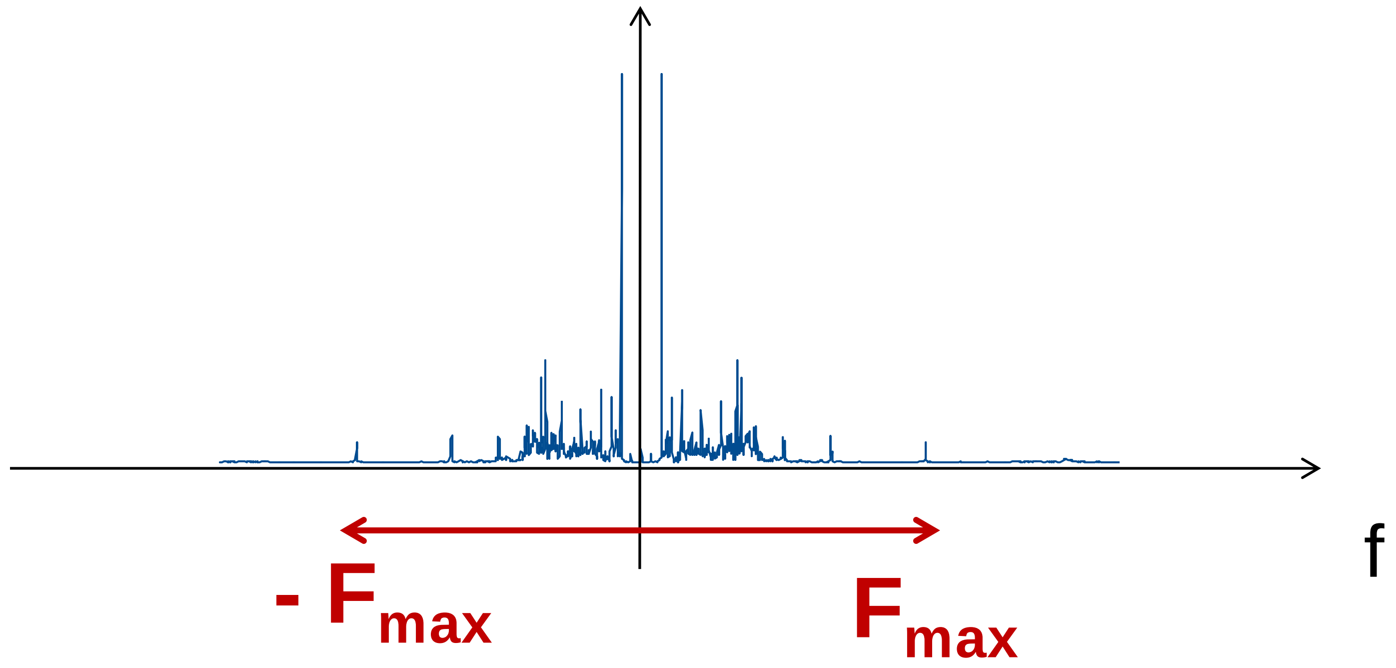
Possibilité d'isoler le spectre d'ordre 0 par filtrage passe-bas



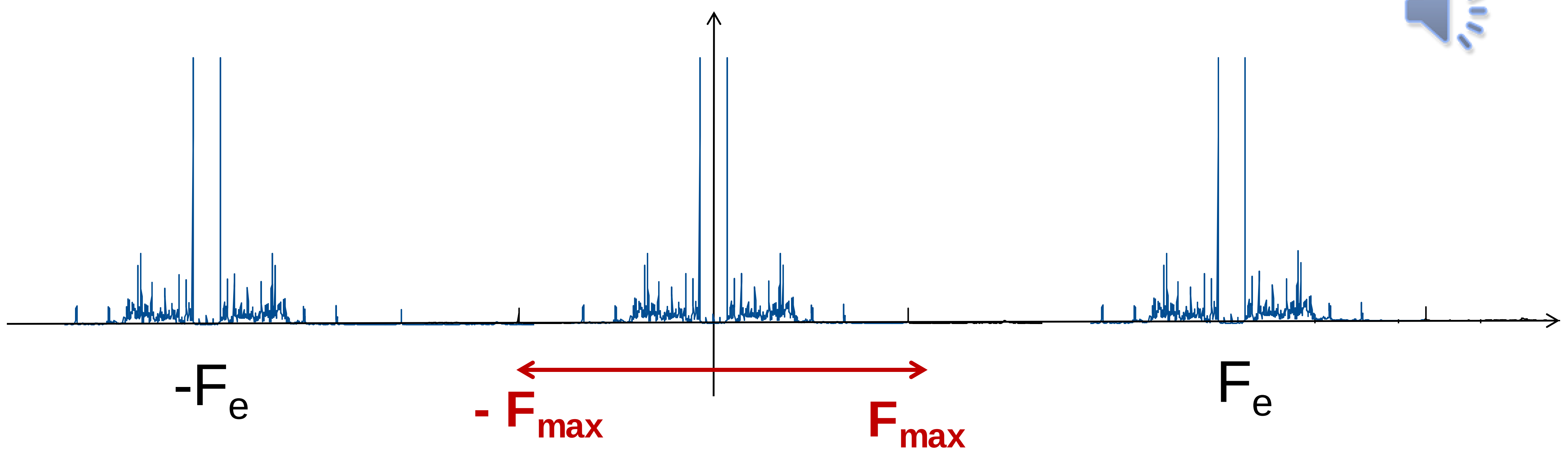
Exemple 1



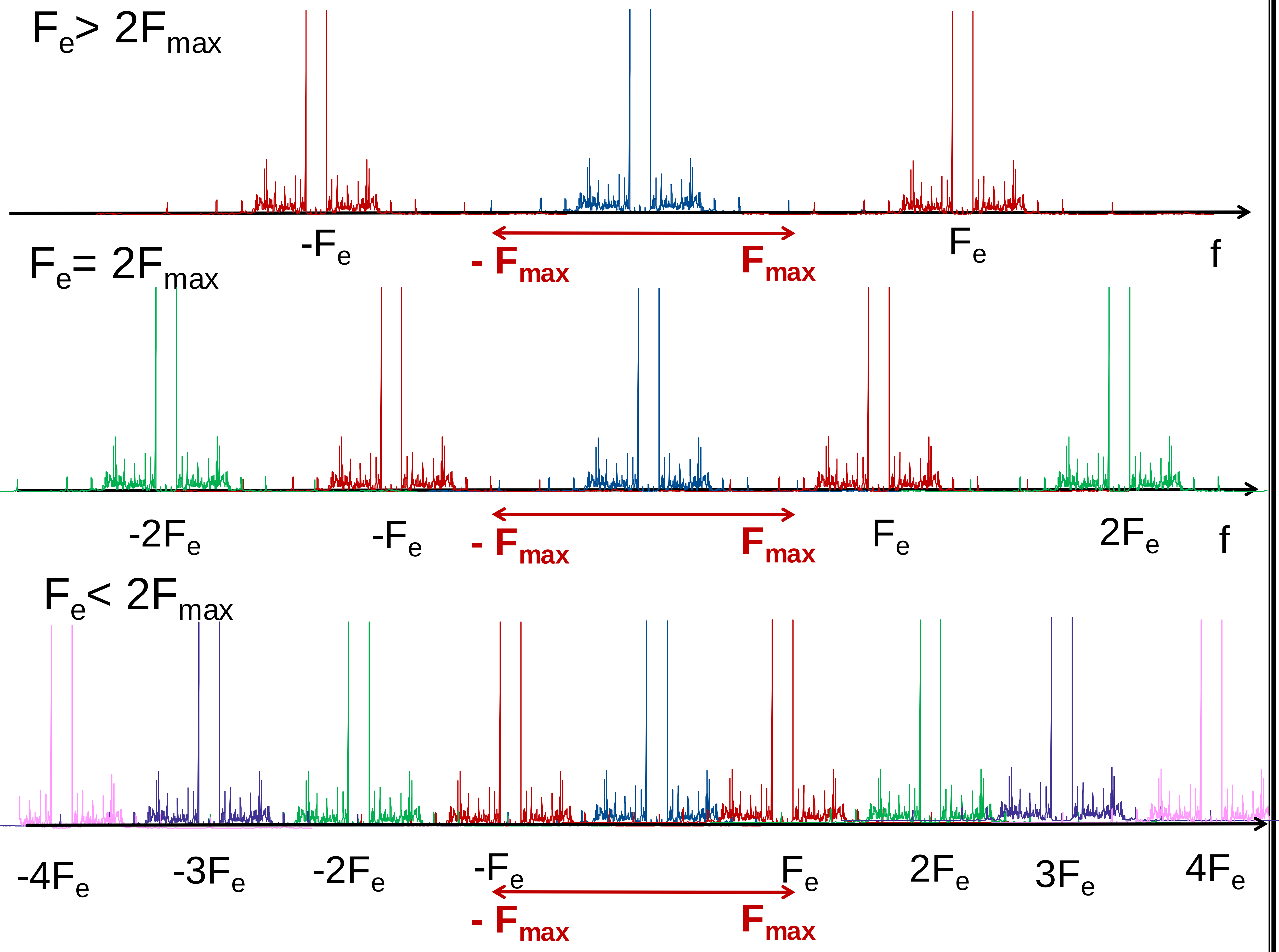
Transformée de Fourier :



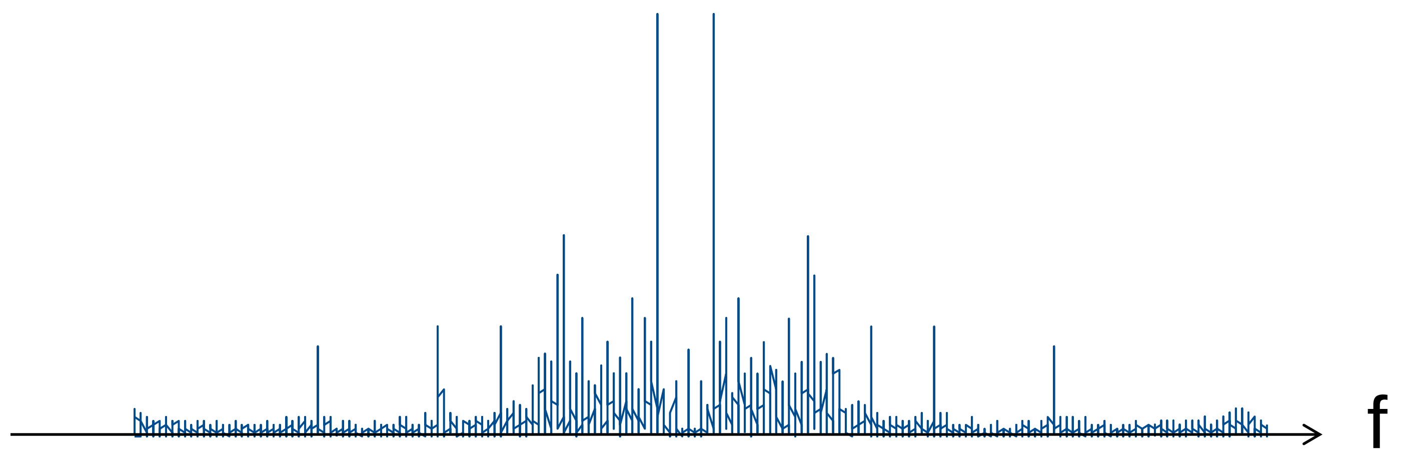
Transformée de Fourier du signal correctement échantillonné :



Quand on diminue la fréquence d'échantillonnage, les périodisations se rapprochent et finissent par apparaitre dans la bande occupée par le signal générant du recouvrement :



Transformée de Fourier du signal échantillonné à $F_e = F_{max} / 12$:



Exemple 2

Image d’origine :
512*512 pixels, quantifiée sur 8 bits

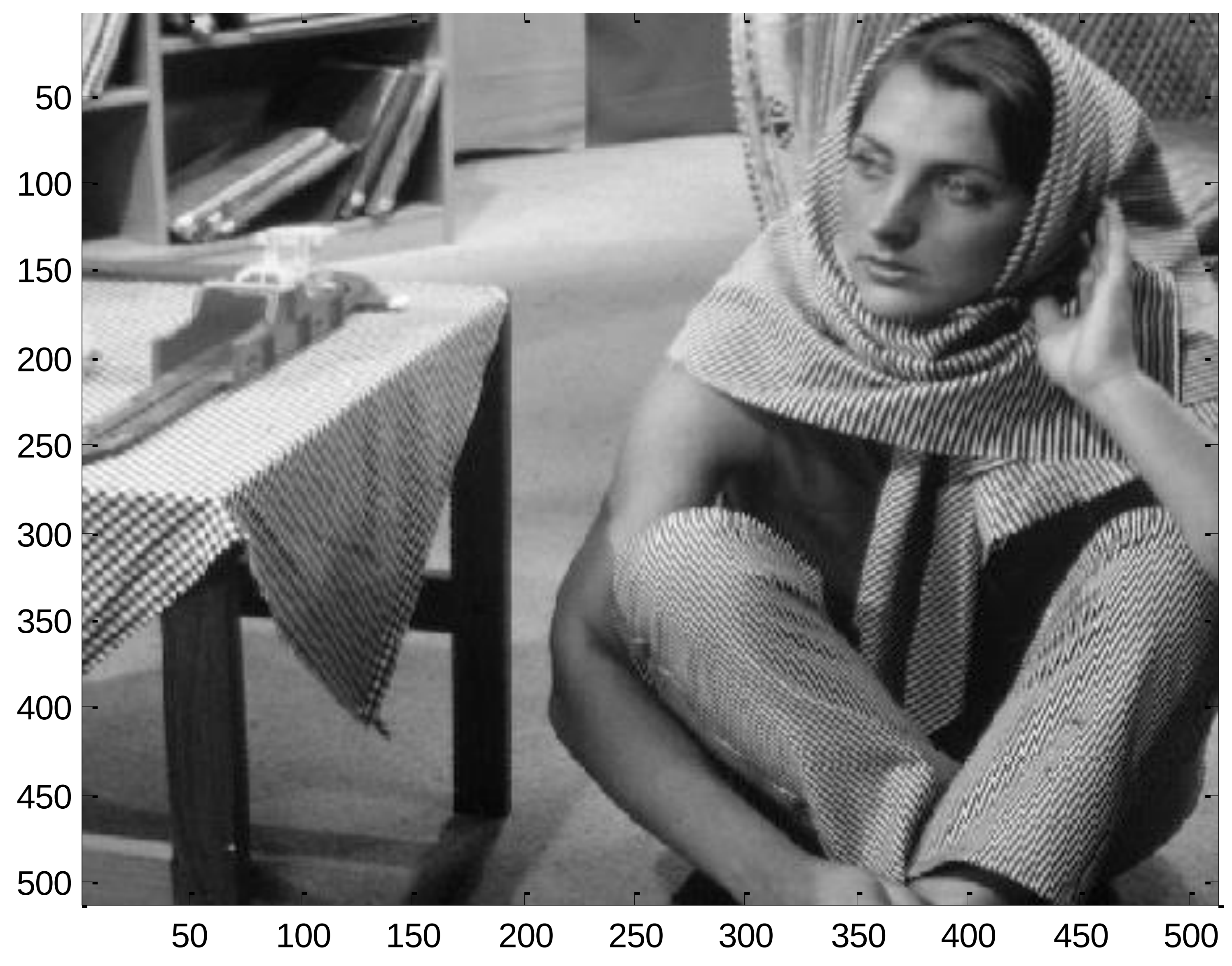


Image sous échantillonnée d’un facteur 2 :
256*256 pixels, quantifiée sur 8 bits

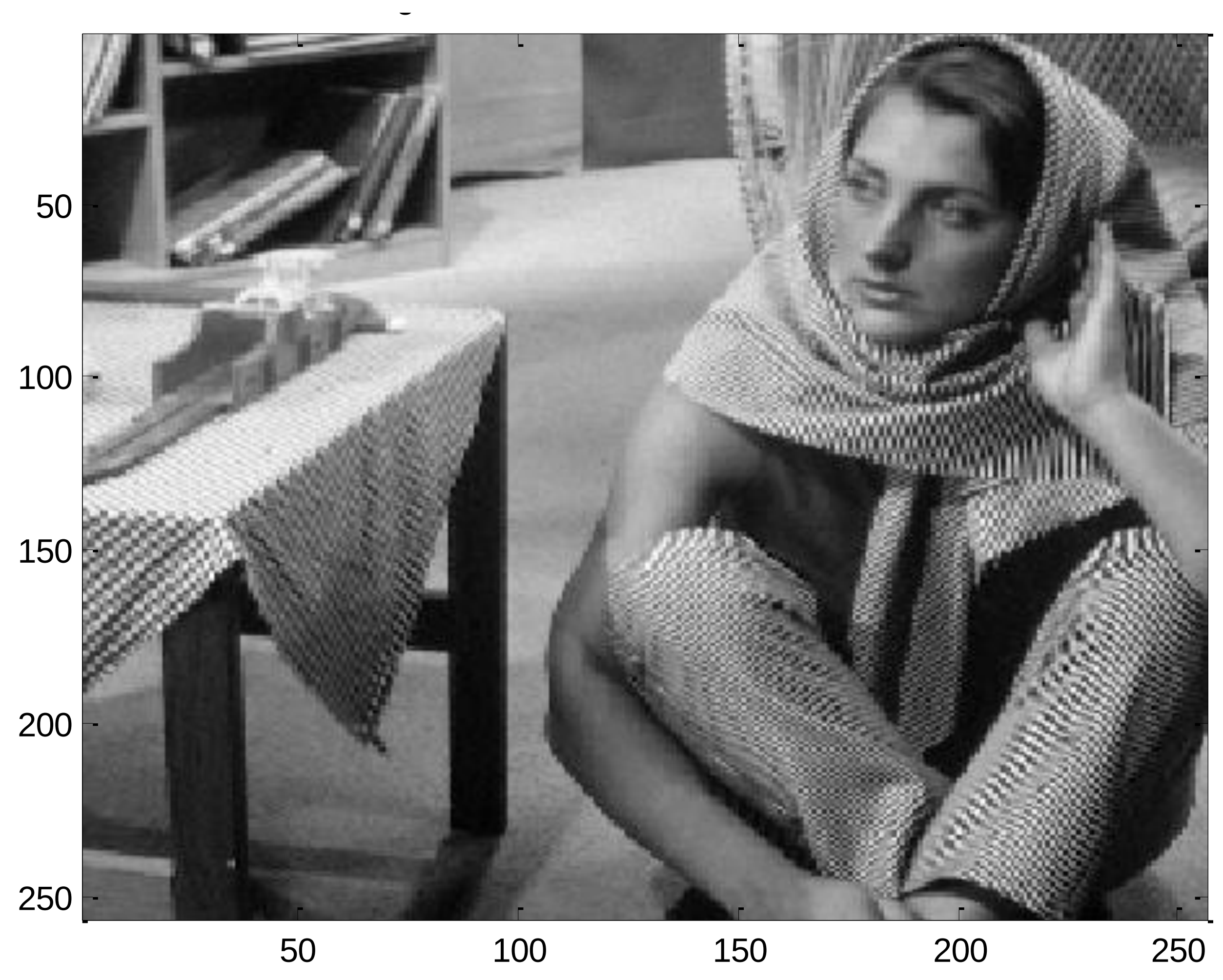
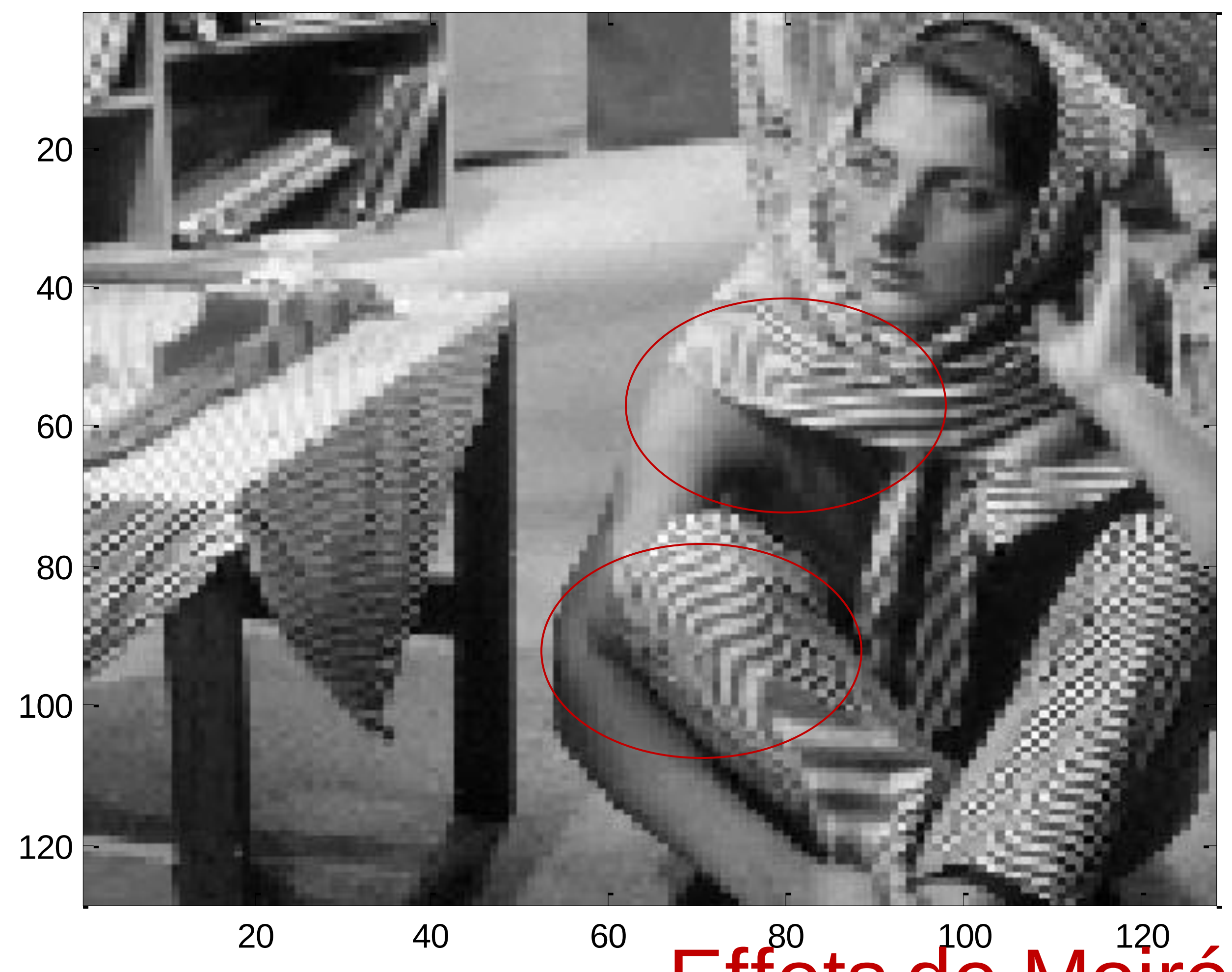


Image sous échantillonnée d’un facteur 4 :
128*128 pixels, quantifiée sur 8 bits



Effets de Moiré

Numérisation du signal : Quantification

Signal quantifié :

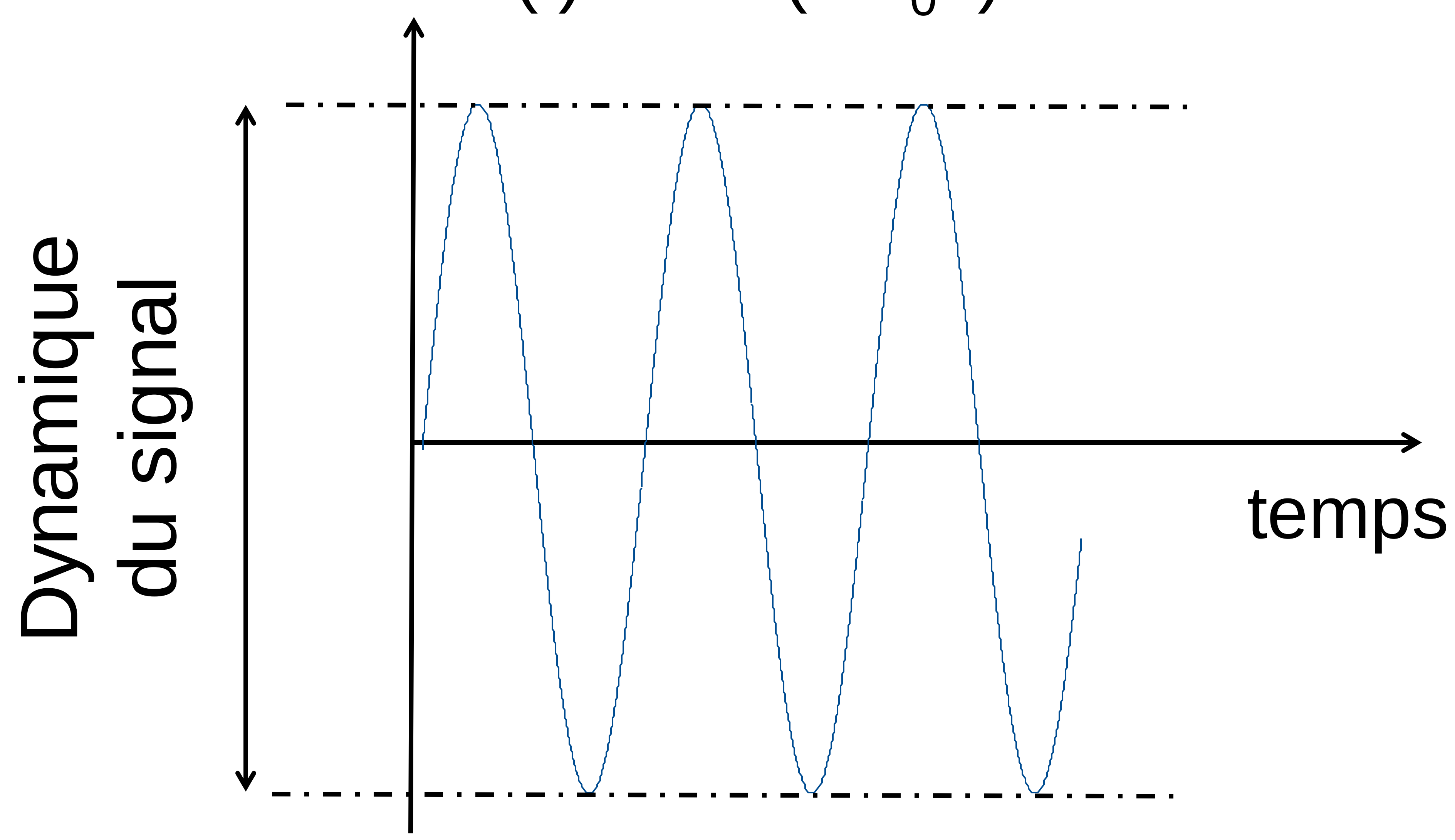
signal défini à chaque instant par un nombre fini de valeurs

Numérisation du signal : Quantification

Signal quantifié :
signal défini à chaque instant par un nombre fini de valeurs

<u>Exemple :</u>

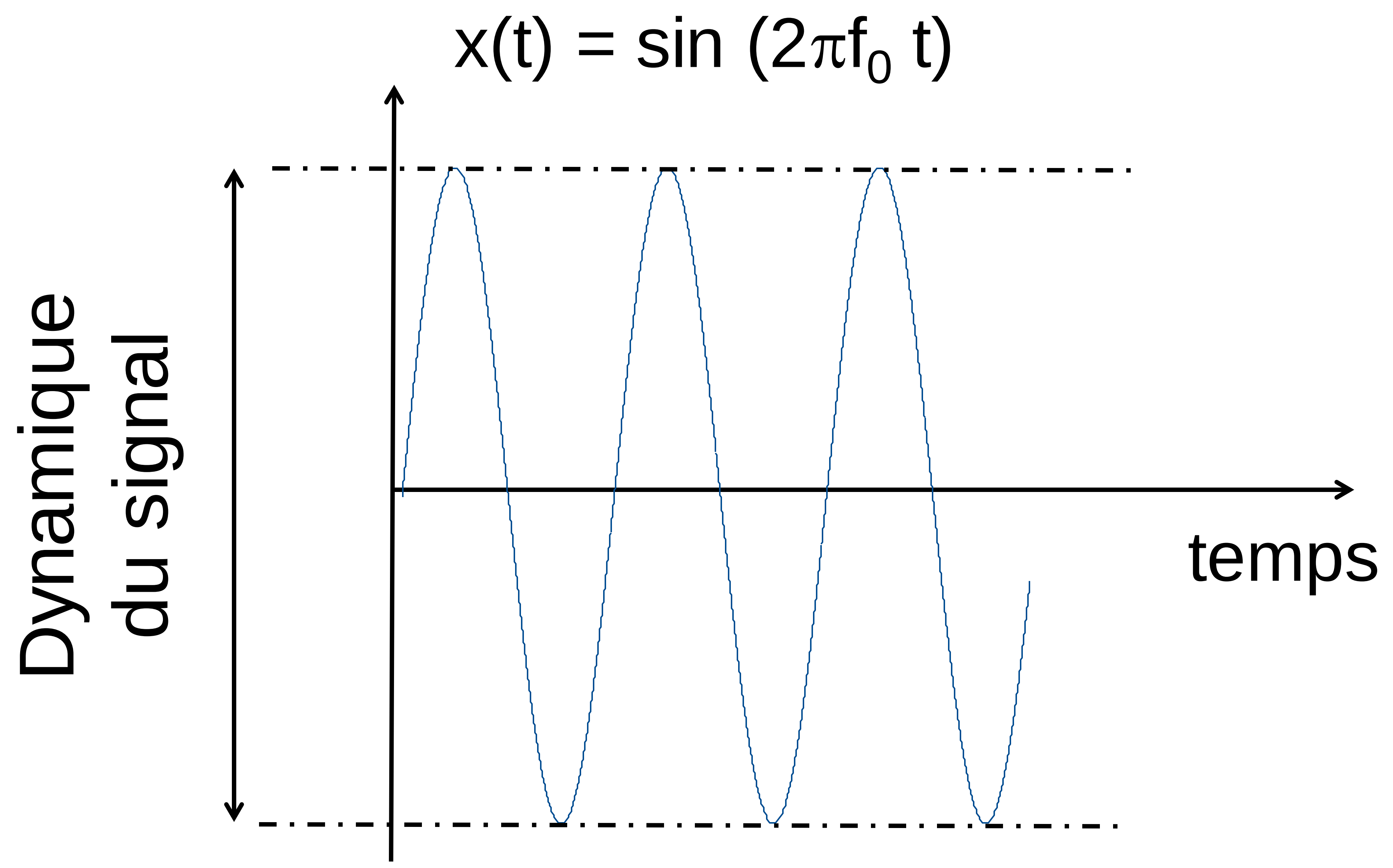
$x(t) = \sin(2\pi f_0\, t)$



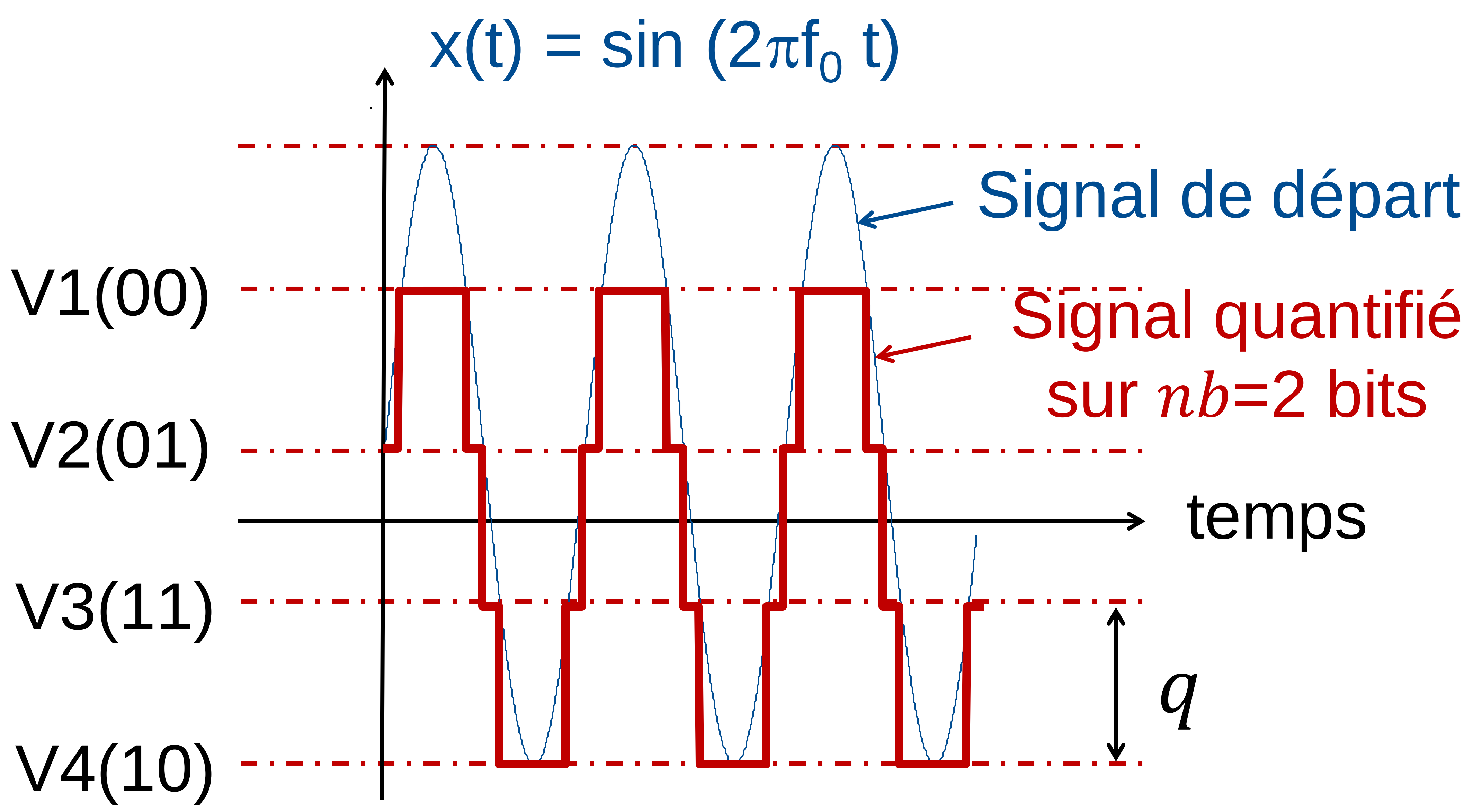
Numérisation du signal : Quantification

Signal quantifié :
signal défini à chaque instant par un nombre fini de valeurs

Exemple :



$x(t) = \sin(2\pi f_0\, t) \in \{V1, V2, V3, V4\}$



Pas de quantification : $q = \dfrac{Dynamique\ du\ signal}{2^{nb}}$

Numérisation du signal : Quantification

Peut-on conserver la même information dans le signal quantifié ?

Numérisation du signal : Quantification

Peut-on conserver la même information dans le signal quantifié ?

Non,
L'opération de quantification est une opération irréversible.

Numérisation du signal : Quantification

Peut-on conserver la même information dans le signal quantifié ?

Mais le signal quantifié et le signal non quantifié peuvent cependant être très proches…

Non,
L'opération de quantification est une opération irréversible.

Une opération de quantification correctement réalisée va ajouter un bruit, dit *bruit de quantification*, au signal de départ (non quantifié).

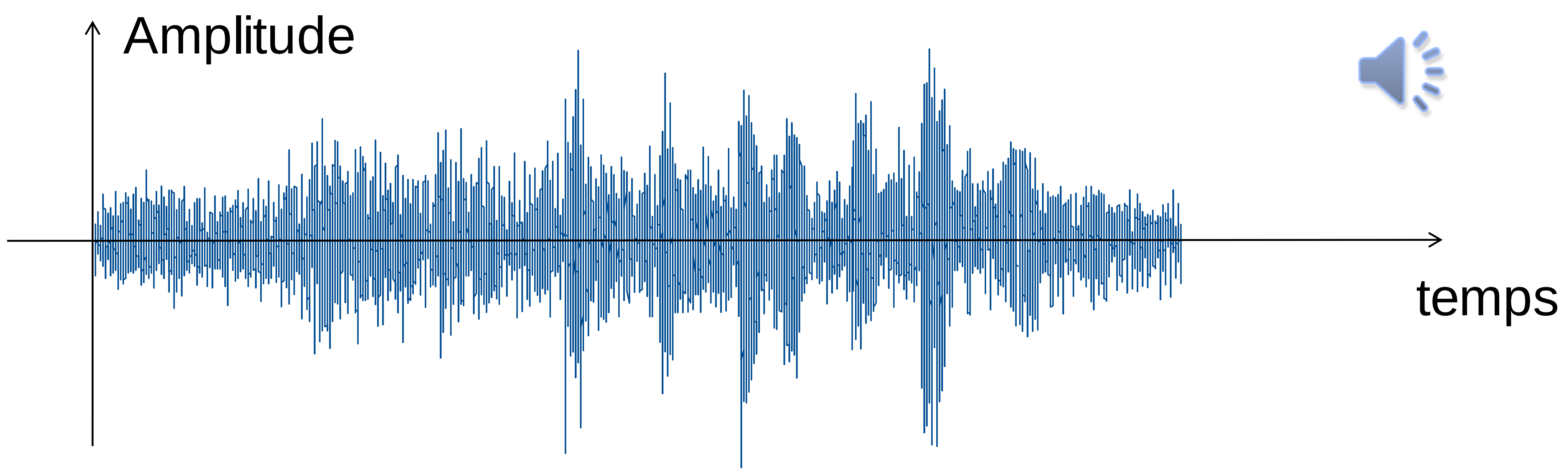
On montre que le rapport signal sur bruit (ou SNR) de quantification est donné par :

$$SNR_Q(dB) = 10\ log\frac{P_{signal\ non\ quantifié}}{P_{bruit\ de\ quantification}} = 6nb + constante$$

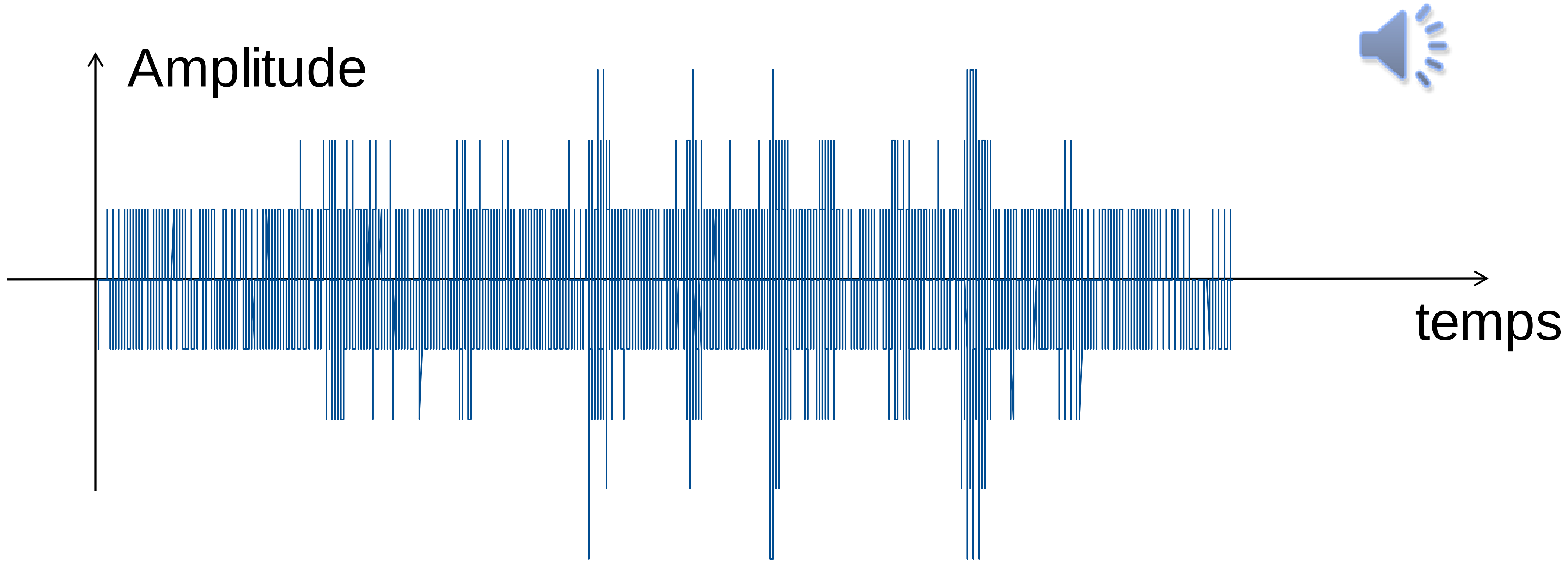
La constante dépendant du signal considéré

Exemple 1

→ Signal quantifié sur nb=16 bits => 2^{16}=65536 niveaux :



→ Signal quantifié sur nb = 4 bits => 2^4 = 16 niveaux :



Exemple 2

Image d'origine
512*512,
Quantifiée sur 8 bits

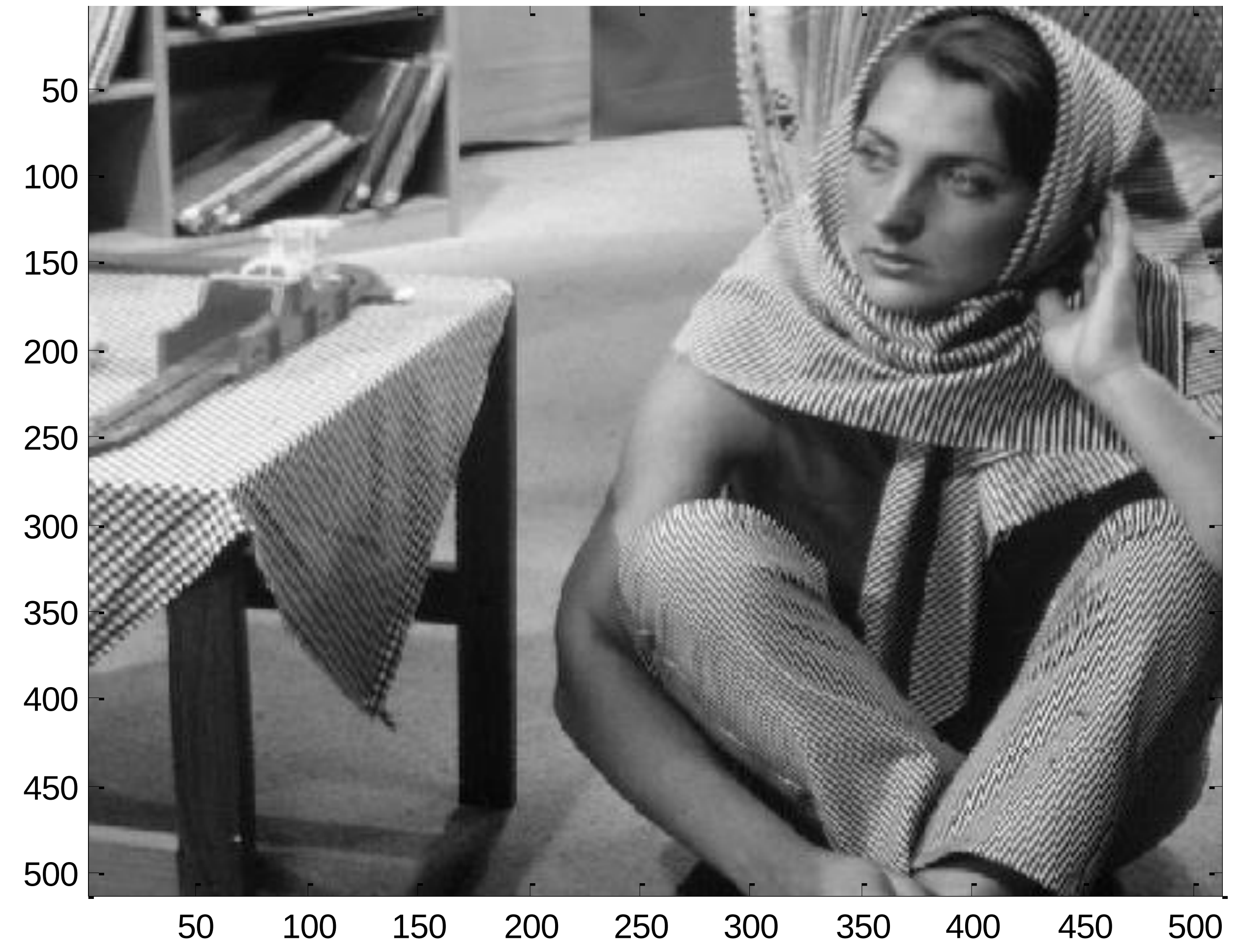
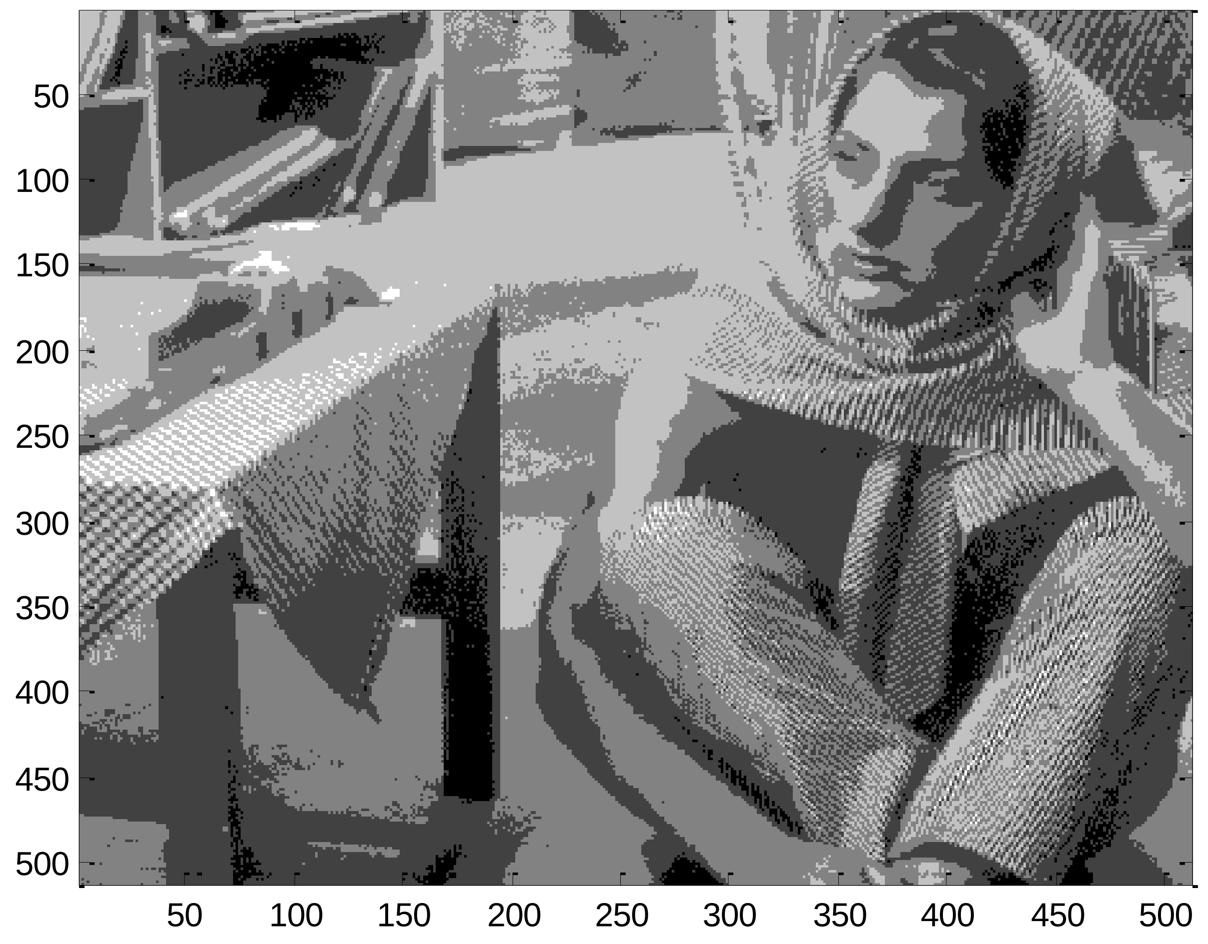


Image quantifiée **sur 4 bits**

Image quantifiée **sur 2 bits**



Numérisation du signal : Echantillonnage + Quantification

Signal numérique :

signal défini à des instants discrets par un nombre fini de valeurs

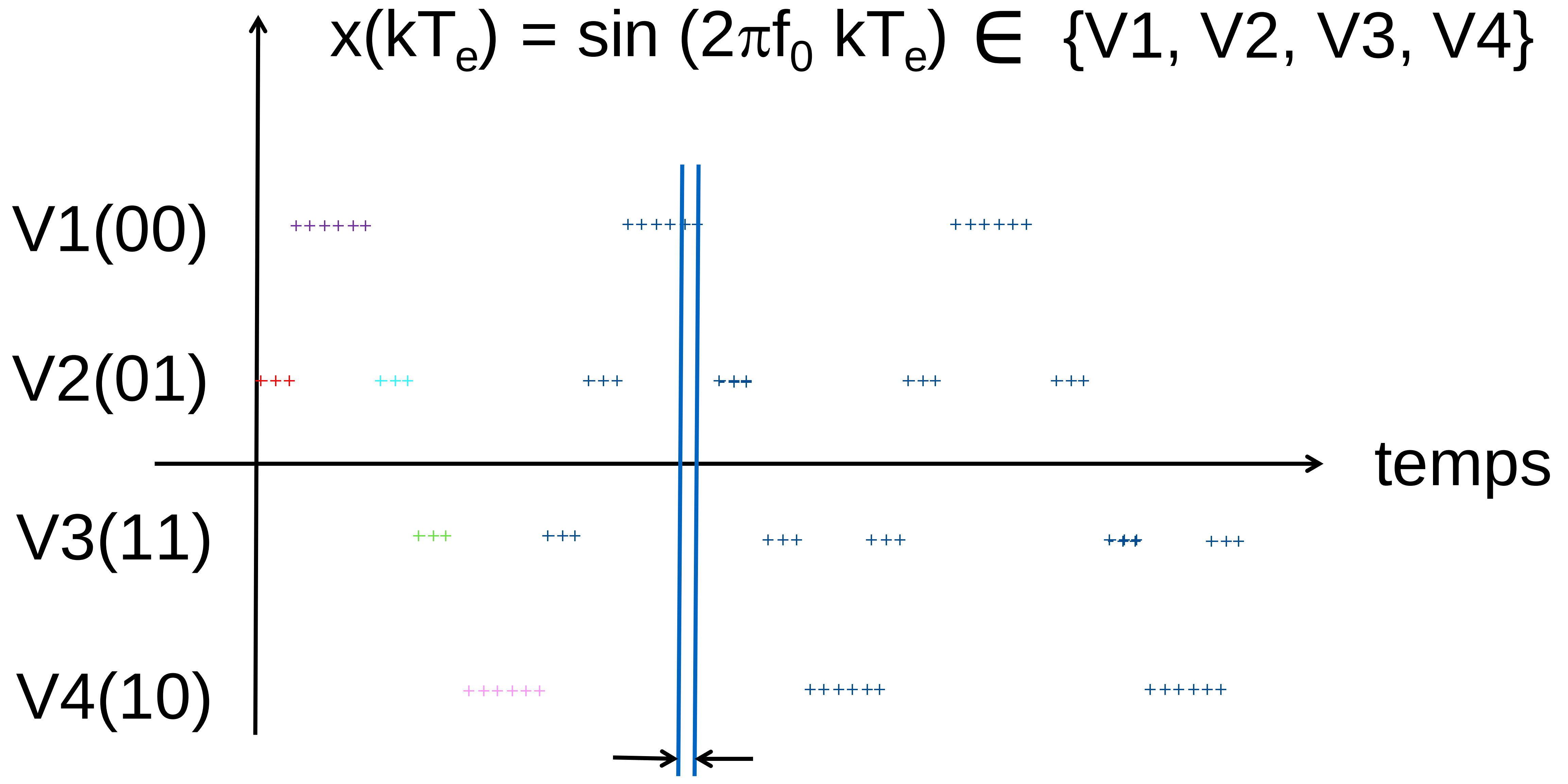
= signal échantillonné et quantifié

Numérisation du signal : Echantillonnage + Quantification

Signal numérique :
signal défini à des instants discrets par un nombre fini de valeurs
= signal échantillonné et quantifié

<u>Exemple 1 :</u>

$x(kT_e) = \sin(2\pi f_0 kT_e) \in \{V1, V2, V3, V4\}$



T_e : période d'échantillonnage

Finalement l'information binaire associée à cette sinusoïde, ou sinusoïde numérisée, va être donnée par :

010101000000000000010101111111101010101010

Numérisation du signal : Echantillonnage + Quantification

Signal numérique :

signal défini à des instants discrets par un nombre fini de valeurs

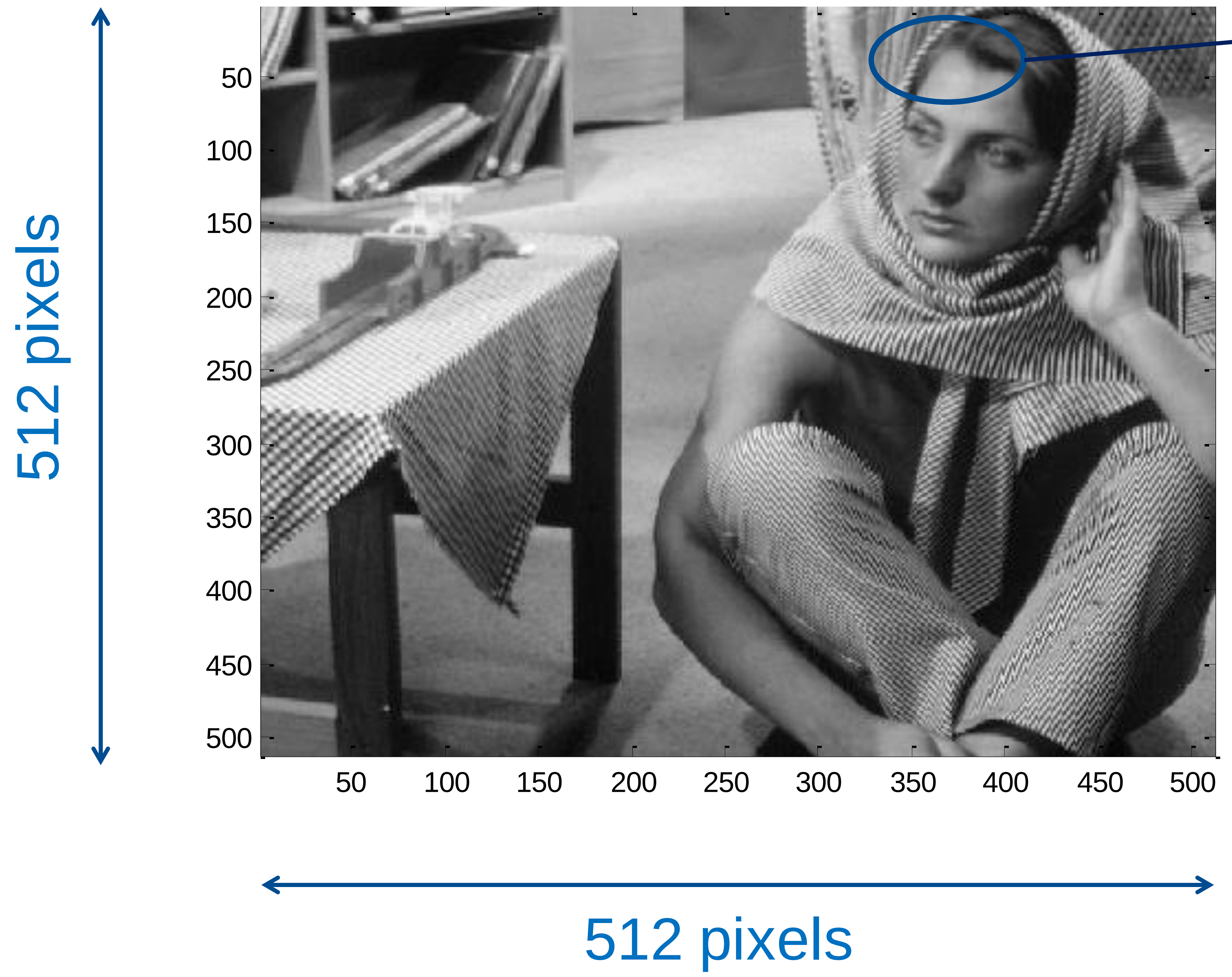
= signal échantillonné et quantifié

Exemple 2 :

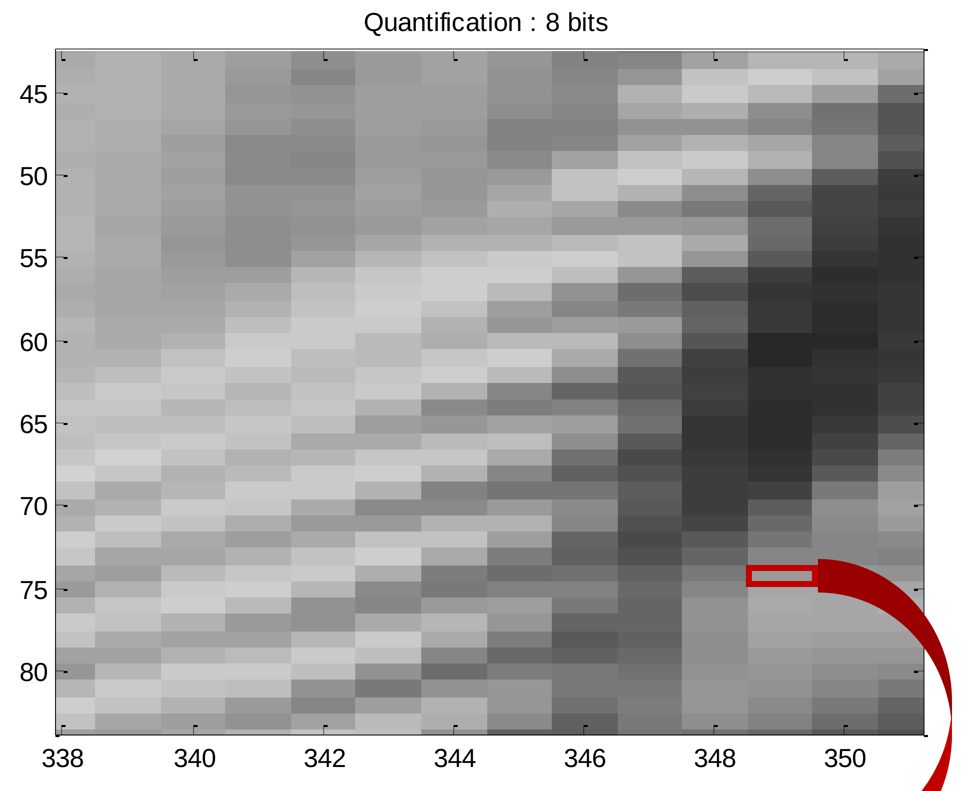
Original image:

512*512,

Quantized on 8 bits



Zoom



1 gray level among 256 (2^8)

Génération d'un signal numérique

Exemple

$$x(t) = \cos\left(2\pi f_0 t\right)\,; \;\; f_0 = 100\ Hz$$

Génération du signal :

$$t \rightarrow [0 : T_e : (N-1)\,T_e]$$

T_e ?

1- $T_e = 100$ Hz
2- $T_e > 5$ ms
3- $T_e < 5$ ms

Génération d'un signal numérique

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Génération du signal :

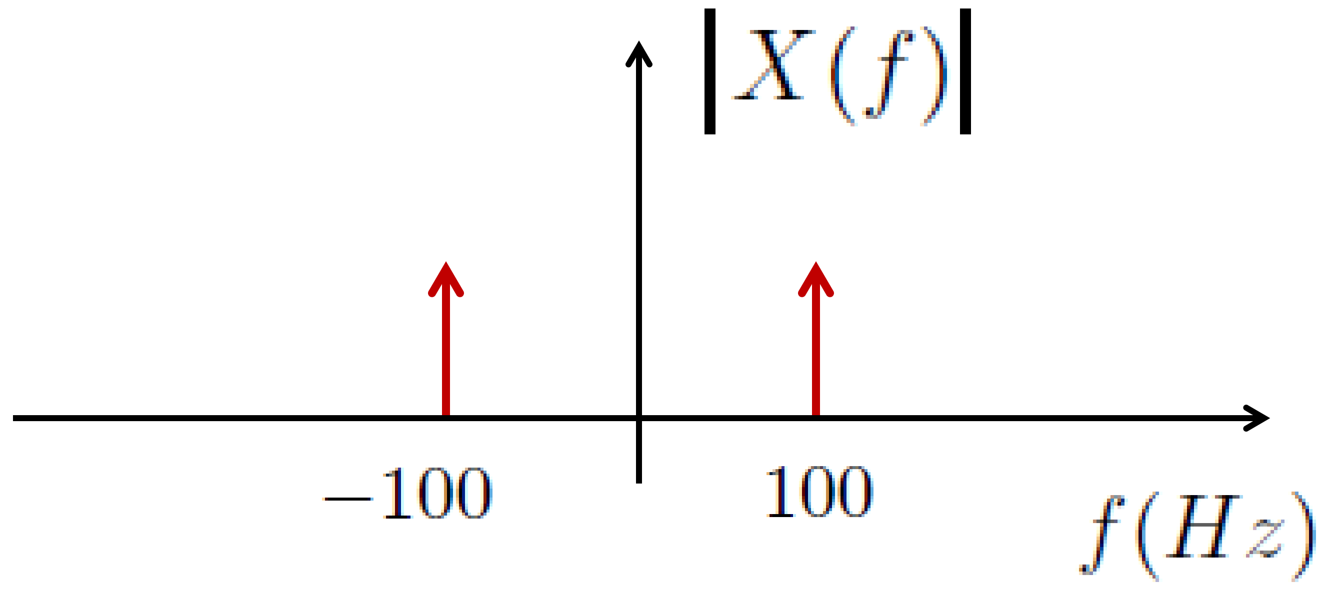
$$t \rightarrow [0 : T_e : (N-1)\,T_e]$$

T_e ?

1- T_e = 100 Hz
2- T_e > 5 ms
3- T_e < 5 ms

Transformée de Fourier :

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$



Génération d'un signal numérique

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Génération du signal :

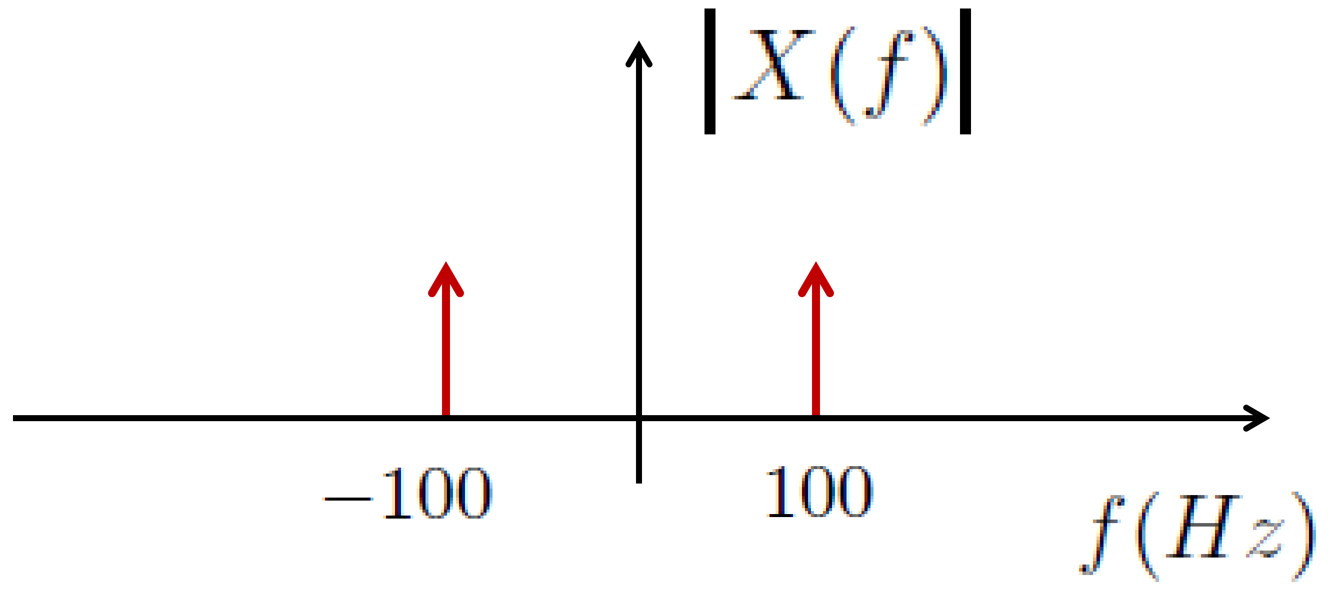
$$t \rightarrow [0 : T_e : (N-1)\,T_e]$$

T_e ?

1- T_e = 100 Hz
2- T_e > 5 ms
3- T_e < 5 ms

Transformée de Fourier :

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$



F_{max}=100 Hz

Génération d'un signal numérique

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Génération du signal :

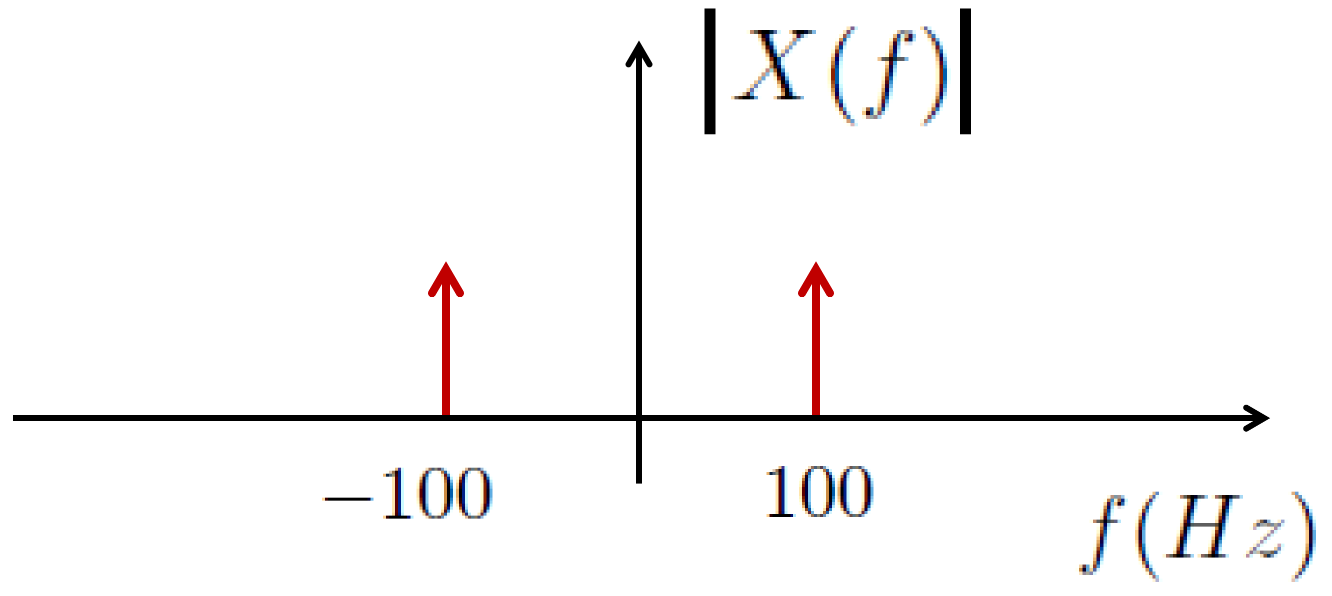
$$t \to [0 : T_e : (N-1)\,T_e]$$

T_e ?

1- T_e = 100 Hz
2- T_e > 5 ms
3- T_e < 5 ms

Transformée de Fourier :

$$X(f) = \frac{1}{2}\left(\delta(f + f_0) + \delta(f - f_0)\right)$$



F_{max}=100 Hz

$$F_e = \frac{1}{T_e} > 2F_{max} \Rightarrow T_e < 5\ ms$$

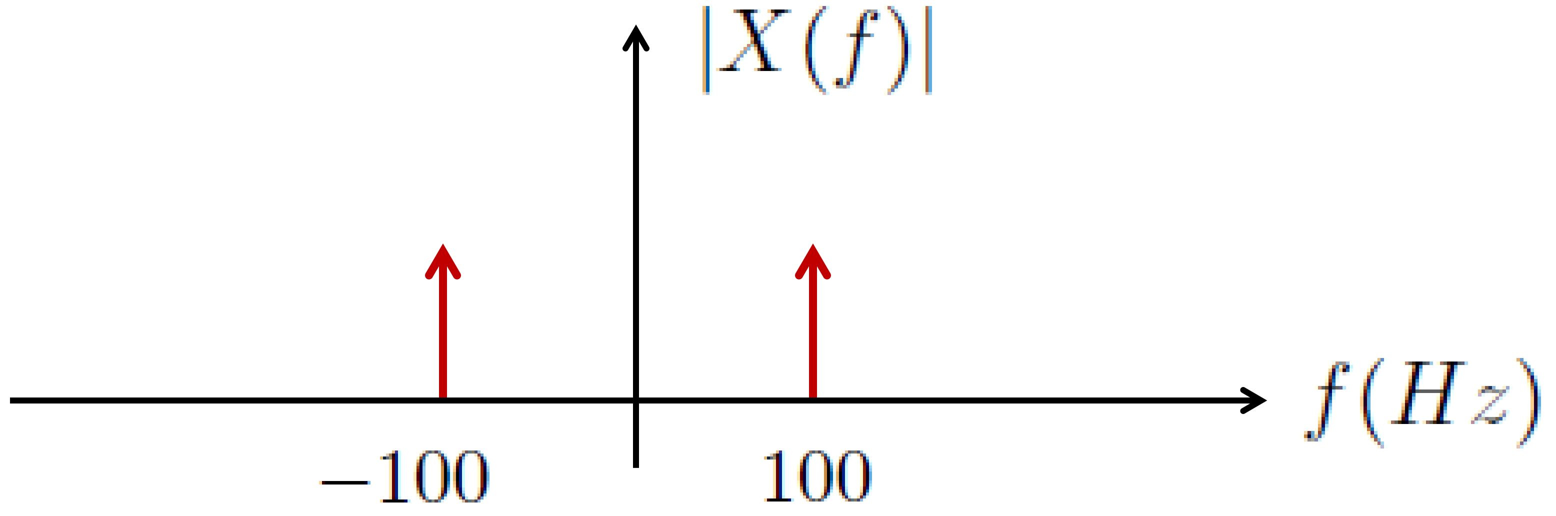
Génération d'un signal numérique

$$x(t) = \cos(2\pi f_0 t)\;;\;\; f_0 = 100\ Hz$$

<u>TF du signal :</u>

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$

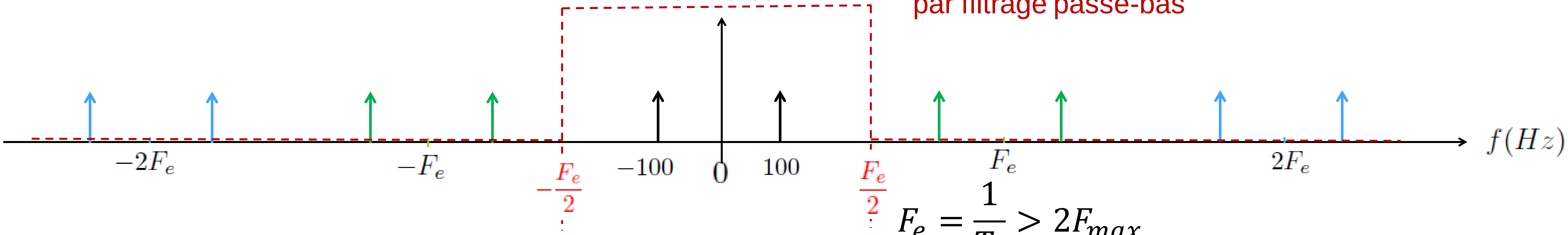
<u>Tracé du module de la TF du signal :</u>



$\Rightarrow F_{max}$=100 Hz

<u>Tracé du module de la TF du signal échantillonné :</u>

Possibilité d'isoler le spectre d'ordre 0 par filtrage passe-bas



$$F_e = \frac{1}{T_e} > 2F_{max}$$

Génération d'un signal numérique

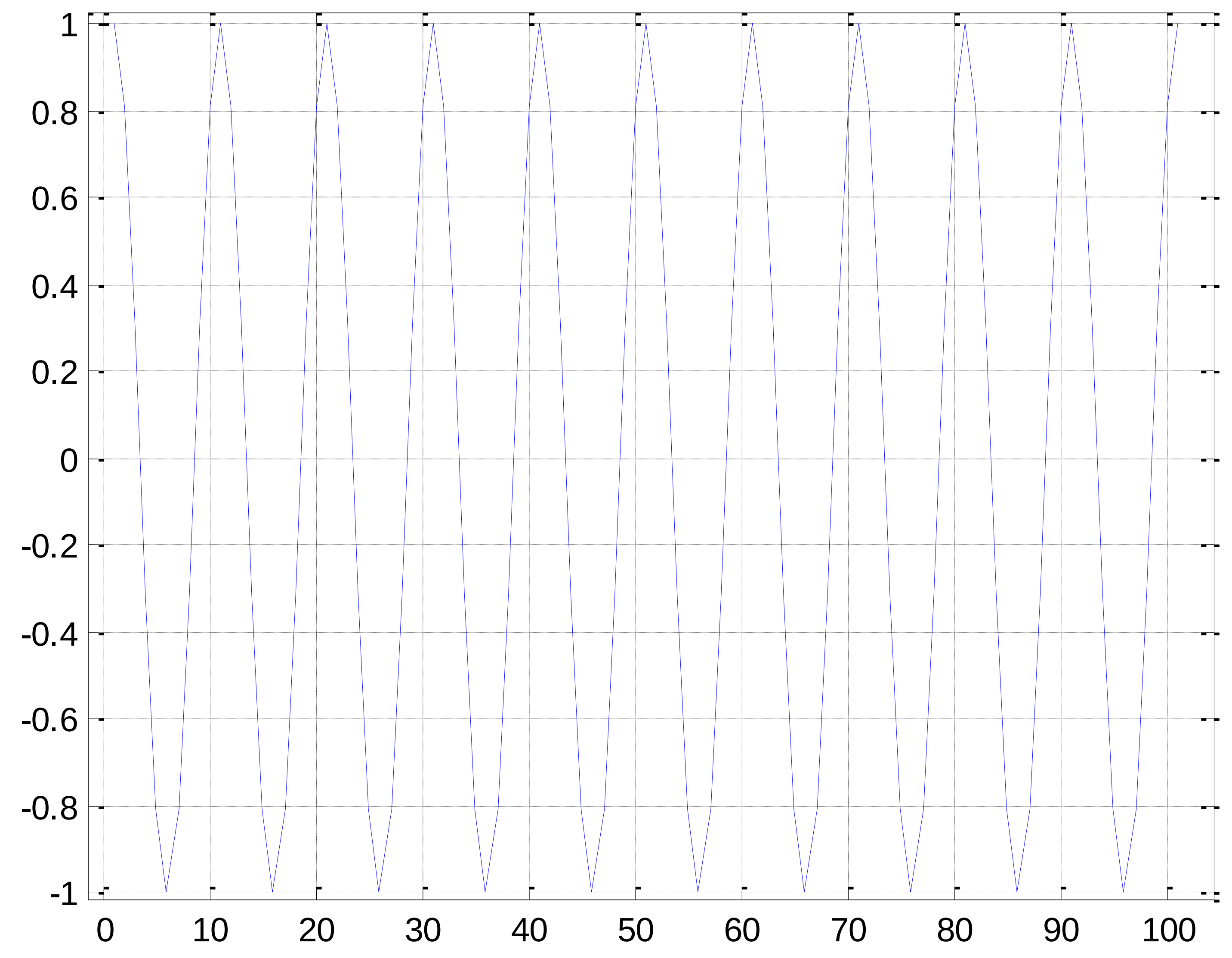
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab
(Suréchantillonnage d'un facteur 5)

```
%Paramètres
f0=100;       %fréquence du cosinus
Fe=1000;      %fréquence d'échantillonnage
Te=1/Fe;      %période d'échantillonnage
N=101;        %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)
```

Signal obtenu :



Génération d'un signal numérique

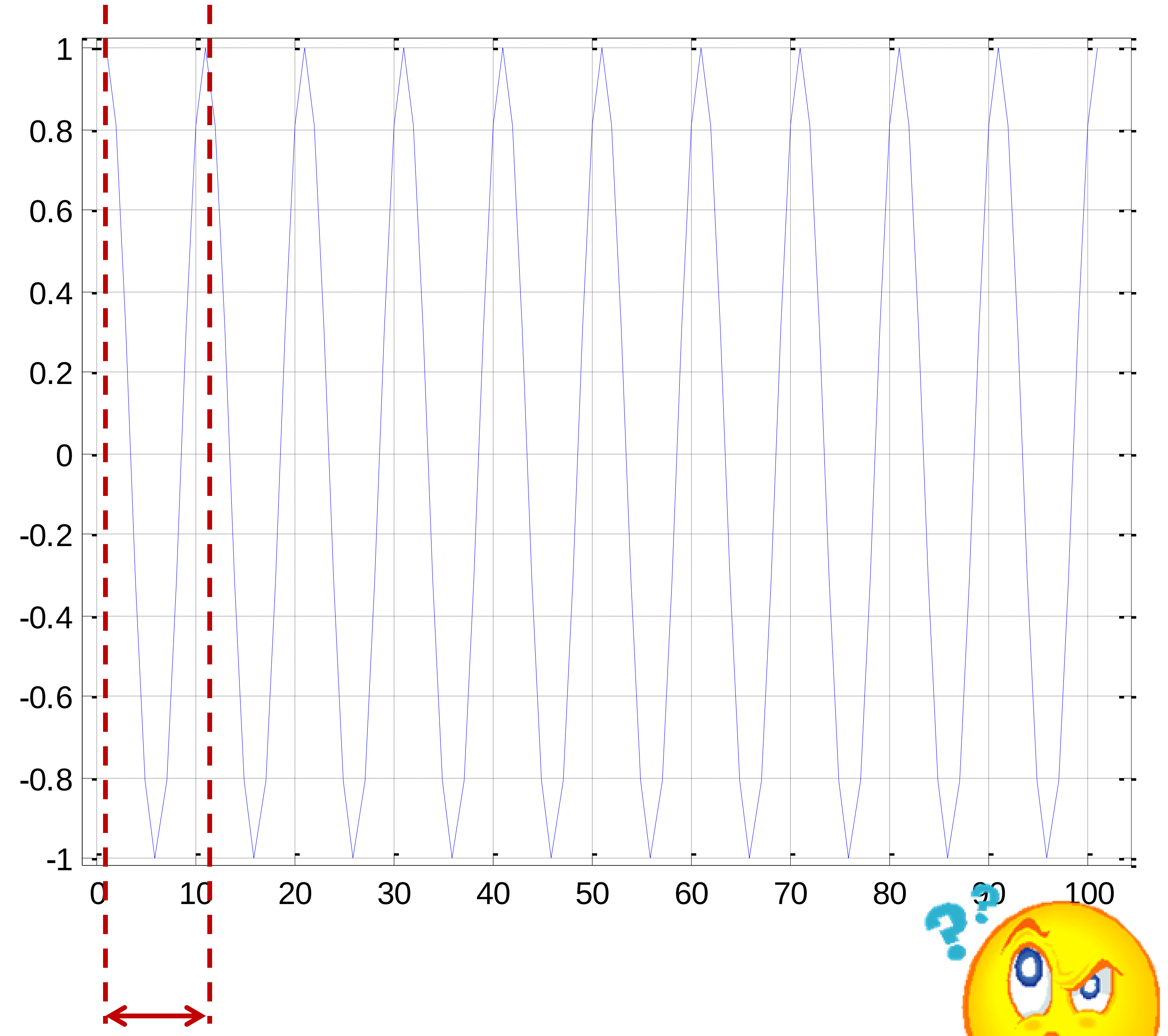
Exemple

$$x(t) = \cos\left(2\pi f_0 t\right); \;\; f_0 = 100 \; Hz$$

Code Matlab
(Suréchantillonnage d'un facteur 5)

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=101;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)
```

Signal obtenu :



Une période = 10 secondes ??

Génération d'un signal numérique

1 période = 10 échantillons de signal

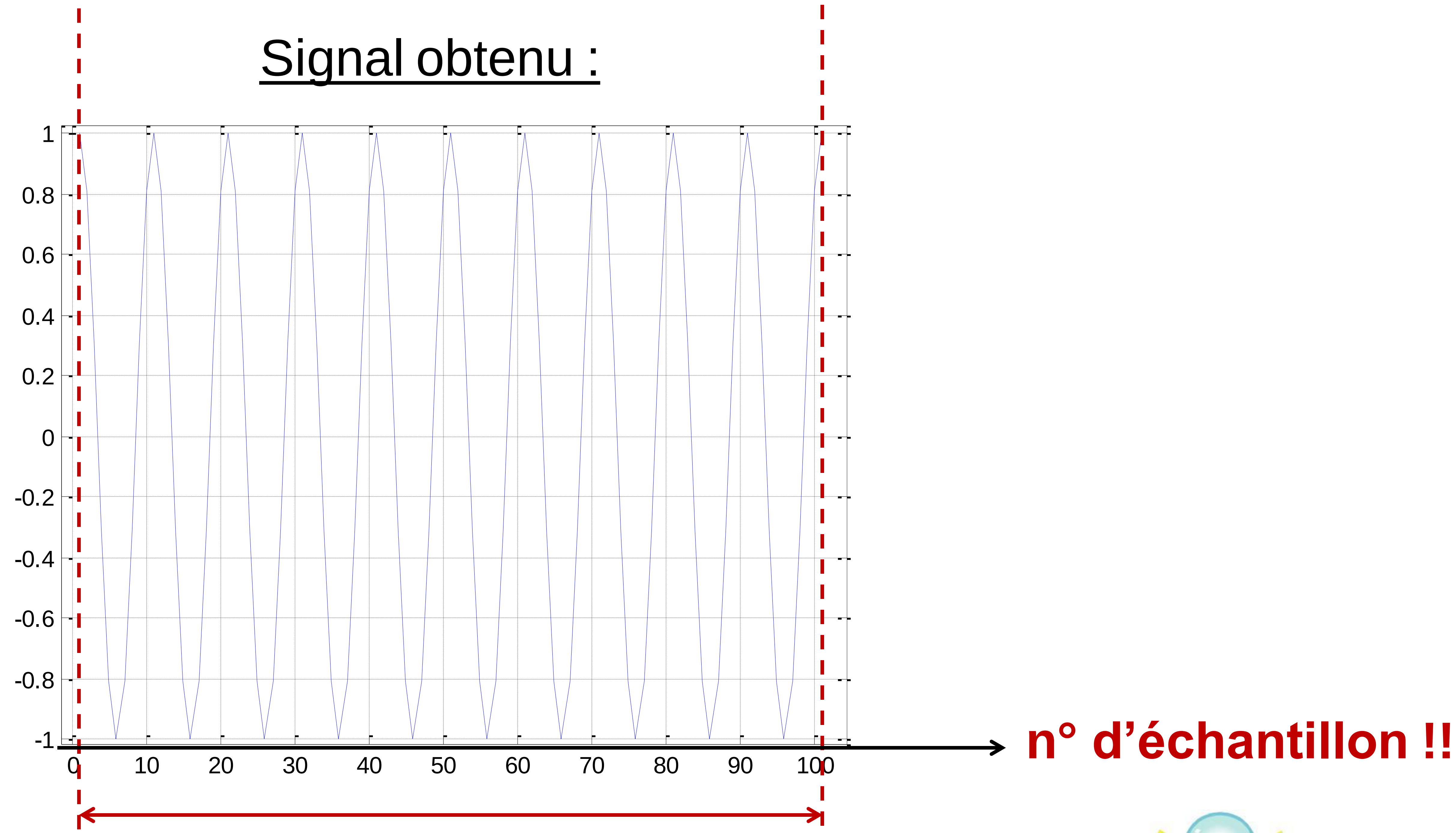
Exemple

$$x(t) = \cos(2\pi f_0 t)\ ;\ \ f_0 = 100\ Hz$$

Code Matlab
(Suréchantillonnage d'un facteur 5)

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)
```

Signal obtenu :



Taille du vecteur = 100 échantillons

Génération d'un signal numérique

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

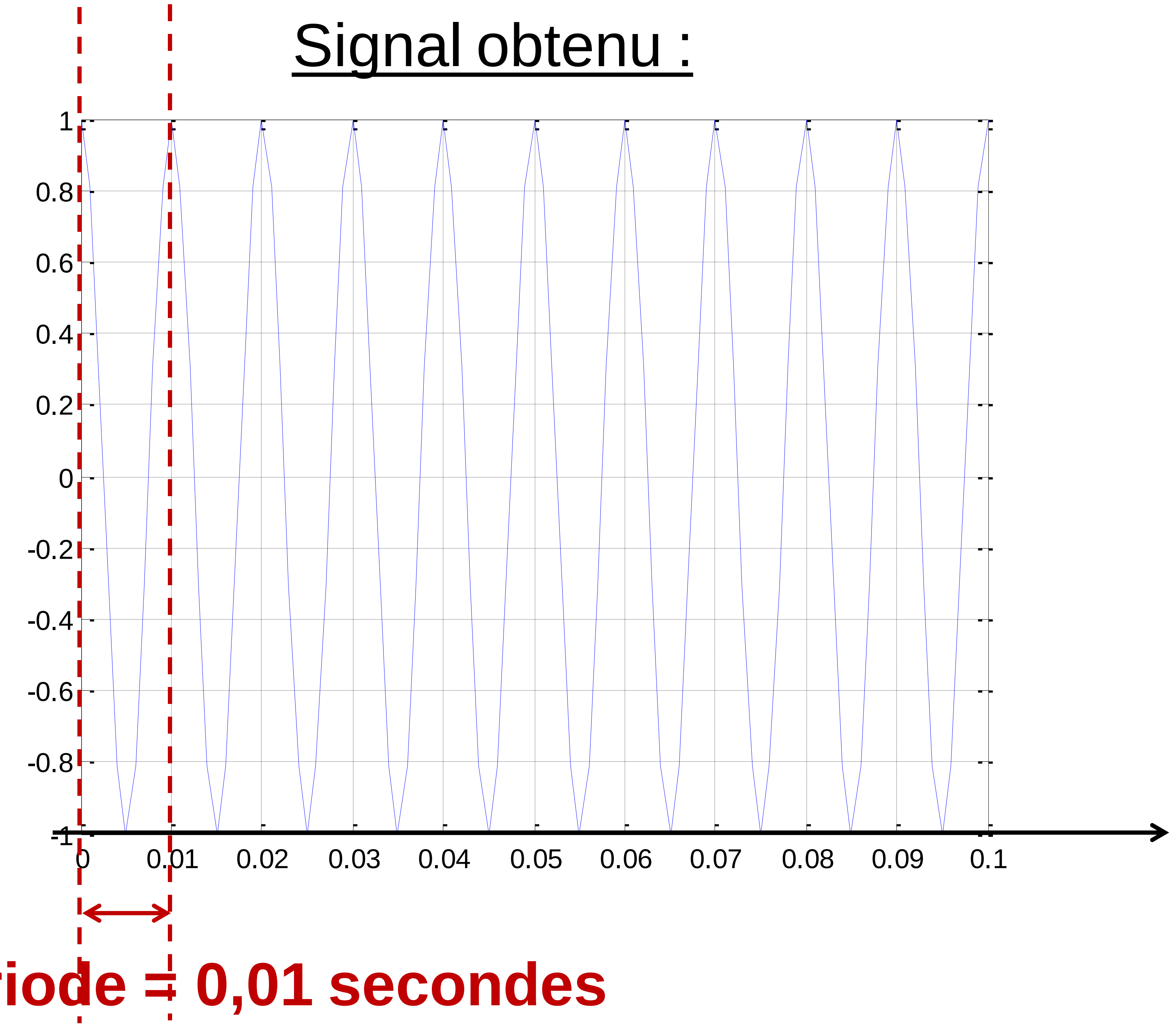
Code Matlab

(Suréchantillonnage d'un facteur 5)

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot([0:Te:(N-1)*Te],x)
```

!! Echelle temporelle à donner !!

Signal obtenu :



Une période = 0,01 secondes

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d’inter et d’auto corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Pourquoi ?

$\xrightarrow{\hspace{3cm}}$

Comment ?

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k)e^{-j2\pi\frac{kn}{N}},\ n = 0, ..., N-1$$

Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Pourquoi ?

$\longrightarrow$

Comment ?

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

1- <u>Echantillonnage temporel</u>

$$x(t) \rightarrow \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Pourquoi ?

Comment ?

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

1- <u>Echantillonnage temporel</u>

$$x(t) \; \rightarrow \; \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

2- <u>Durée limitée du signal</u>

$$x(t) \rightarrow x_L(t) = \begin{cases} x(t) & \textit{pour } t \in [0, L] \\ 0 & \textit{ailleurs} \end{cases}$$

Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Pourquoi ?

$$\longrightarrow$$

Comment ?

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

1- <u>Echantillonnage temporel</u>

$$x(t) \;\rightarrow\; \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

2- <u>Durée limitée du signal</u>

$$x(t) \rightarrow x_L(t) = \begin{cases} x(t) \;\; pour\ t \in [0, L] \\ 0 \;\; ailleurs \end{cases}$$

1+2 => <u>Echantillonnage temporel + durée limitée de signal</u>

$$x(t) \rightarrow \{x(kT_e\}_{k=0,...,N-1}$$

Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Pourquoi ?

Comment ?

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k)e^{-j2\pi\frac{kn}{N}},\ n = 0, ..., N-1$$

1- <u>Echantillonnage temporel</u>

$$x(t) \ \rightarrow \ \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

2- <u>Durée limitée du signal</u>

$$x(t) \rightarrow x_L(t) = \begin{cases} x(t) \ \textit{pour } t \in [0, L] \\ 0 \ \ \textit{ailleurs} \end{cases}$$

1+2 => <u>Echantillonnage temporel + durée limitée de signal</u>

$$x(t) \rightarrow \{x(kT_e\}_{k=0,...,N-1}$$

3- <u>Echantillonnage fréquentiel</u>

$$X(f) \ \rightarrow \ \{X(n\Delta f)\}_{n=0,...,N-1}$$

Transformée de Fourier Discrète

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

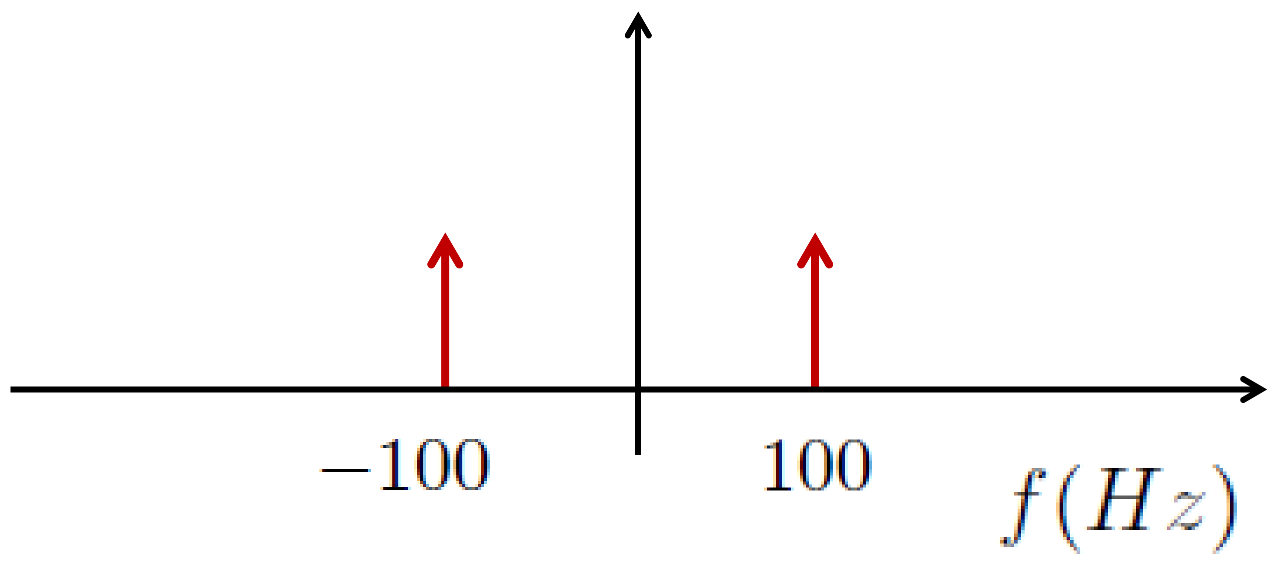
Exemple

$$x(t) = \cos(2\pi f_0 t)\,; \;\; f_0 = 100 \; Hz$$

<u>TF du signal :</u>

$$X(f) = \frac{1}{2}\left(\delta(f + f_0) + \delta(f - f_0)\right)$$

<u>Tracé du module de la TF du signal :</u>



Transformée de Fourier Discrète

$$X(n)=\sum_{k=0}^{N-1} x(k)e^{-j2\pi\frac{kn}{N}},\ n=0,\ldots,N-1$$

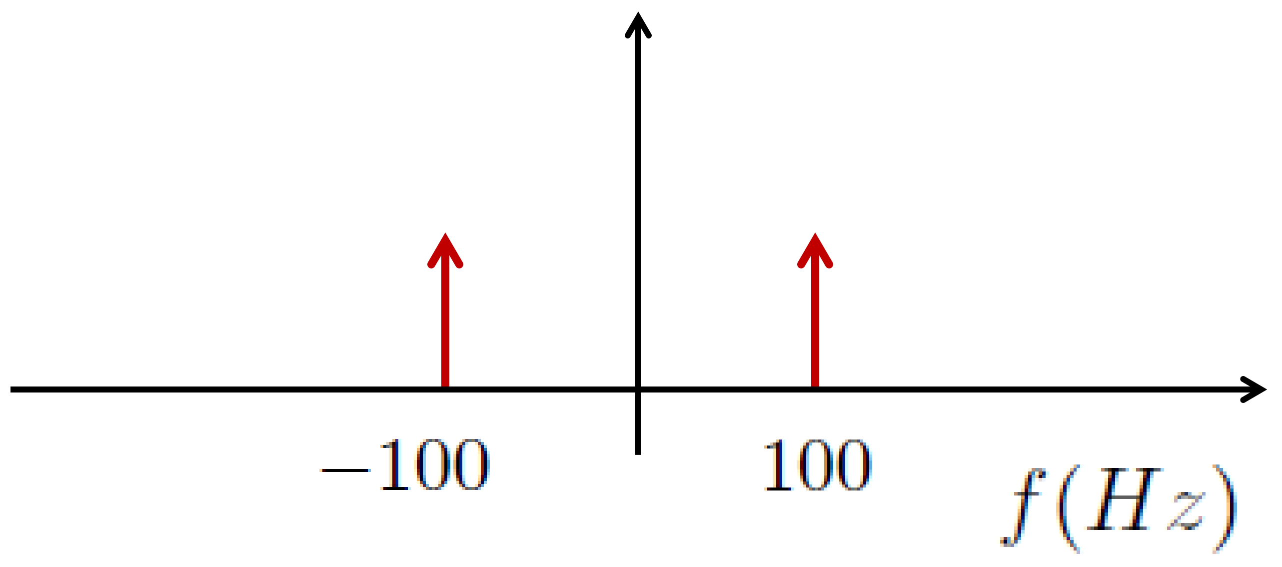
Exemple

$$x(t)=\cos\left(2\pi f_0 t\right);\ \ f_0=100\ Hz$$

TF du signal :

$$X(f)=\frac{1}{2}\left(\delta\left(f+f_0\right)+\delta\left(f-f_0\right)\right)$$

Tracé du module de la TF du signal :



Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Transformée de Fourier Discrète

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi\frac{kn}{N}}, \; n = 0, \dots, N-1$$

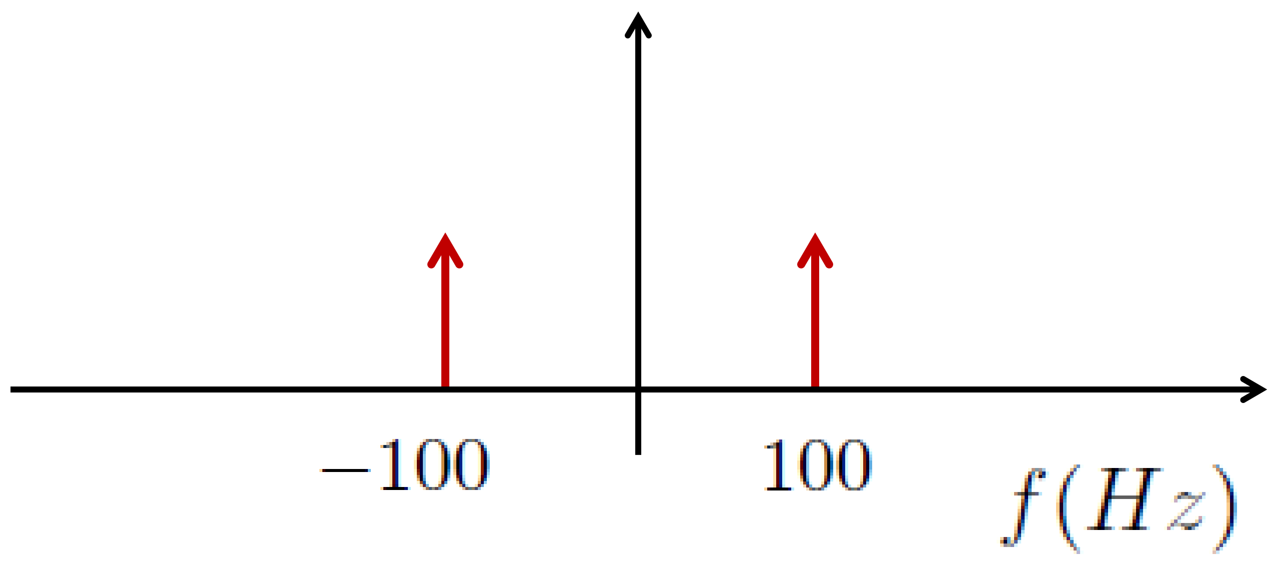
Exemple

$$x(t) = \cos(2\pi f_0 t)\,; \;\; f_0 = 100\ Hz$$

TF du signal :

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$

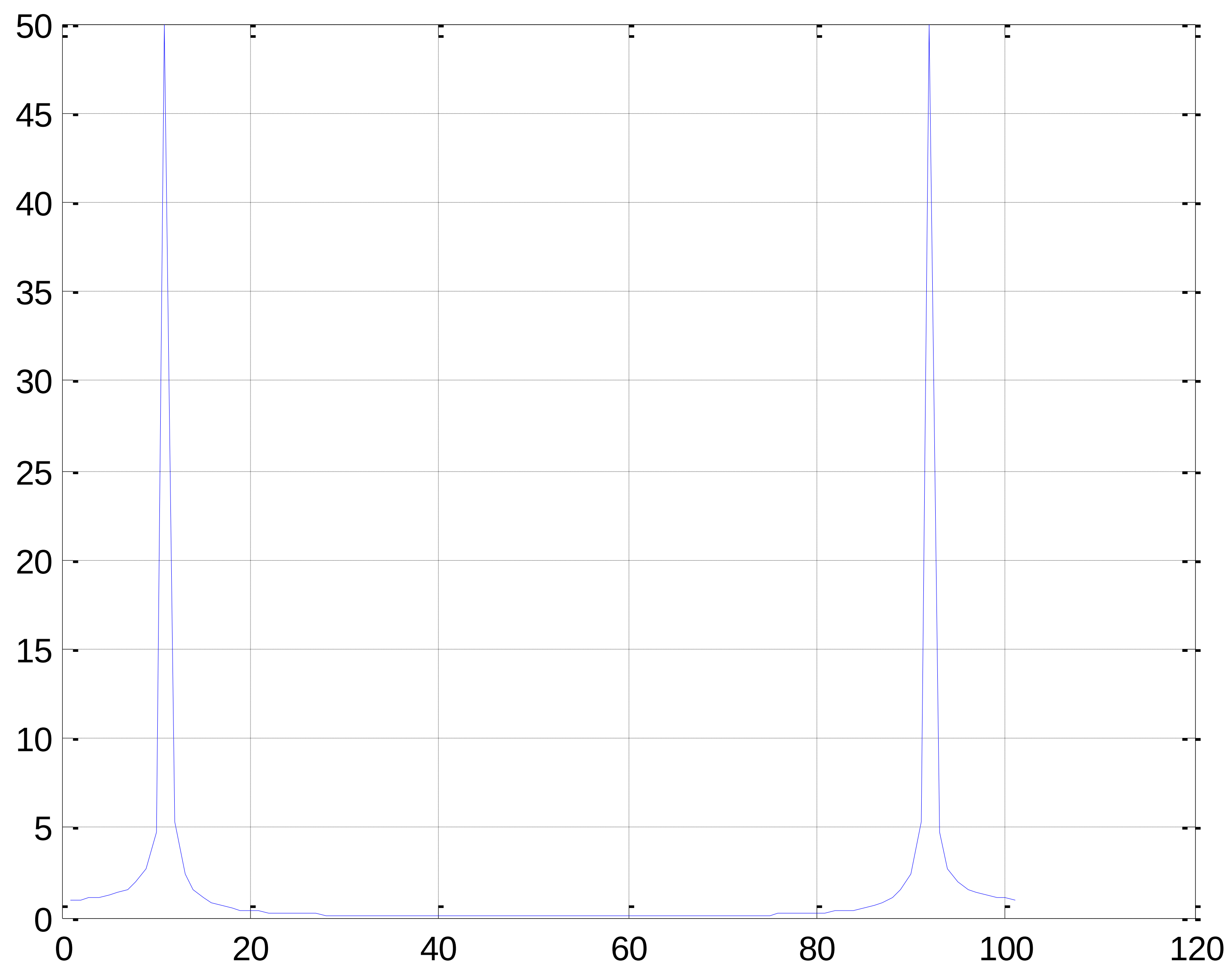
Tracé du module de la TF du signal :



Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;   %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Transformée de Fourier Discrète

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, \ldots, N-1$$

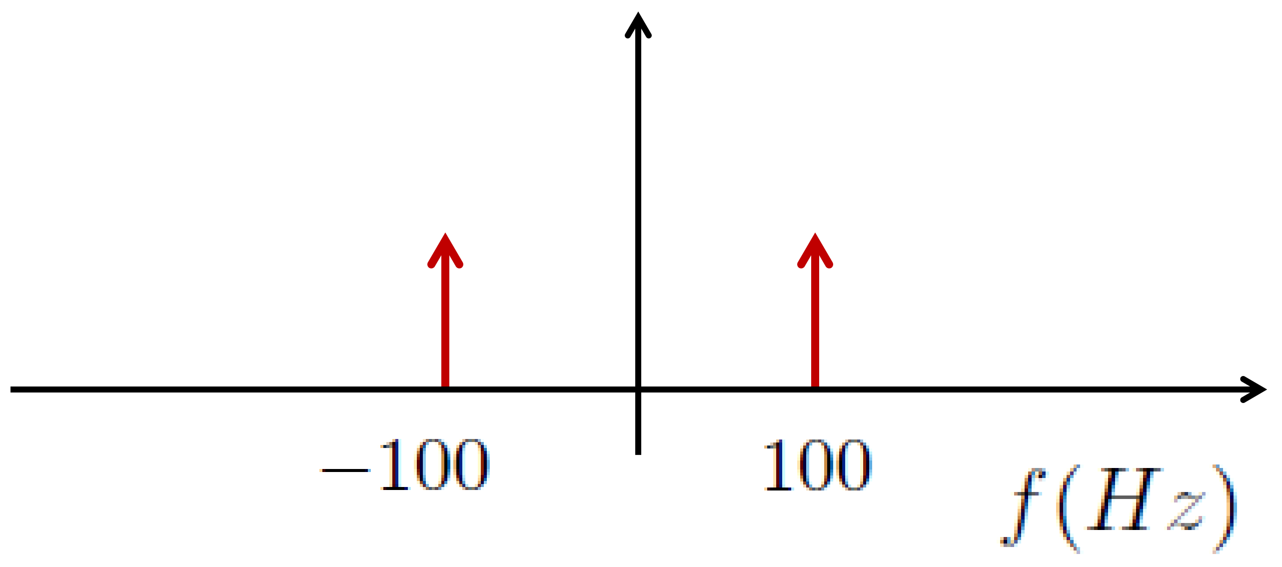
Exemple

$$x(t) = \cos(2\pi f_0 t)\,; \;\; f_0 = 100\ Hz$$

<u>TF du signal :</u>

$$X(f) = \frac{1}{2}\left(\delta\left(f + f_0\right) + \delta\left(f - f_0\right)\right)$$

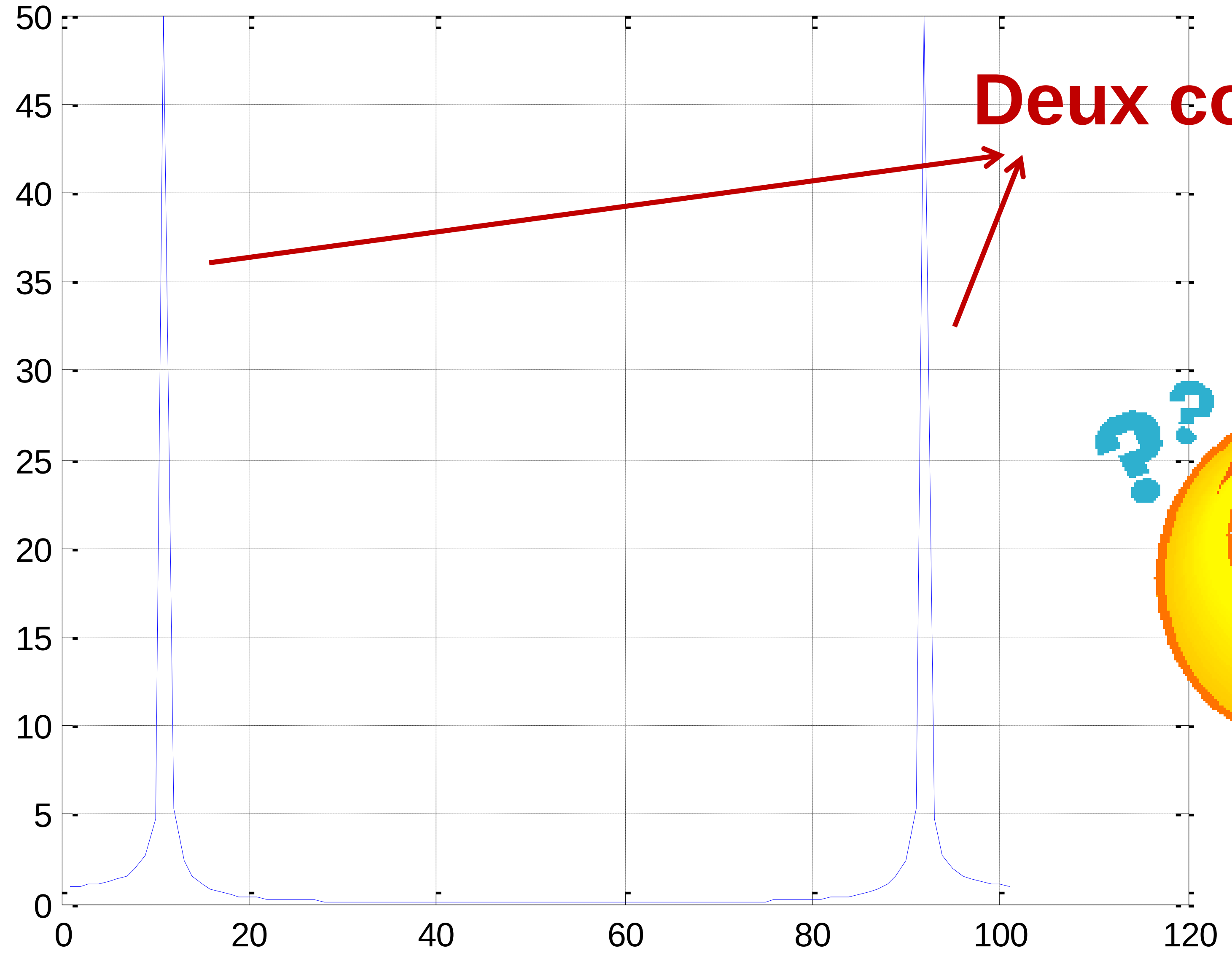
<u>Tracé du module de la TF du signal :</u>



<u>Code Matlab :</u>

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;   %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

<u>Tracé du module de la TFD obtenu :</u>



Deux cosinus ?

Transformée de Fourier Discrète

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, \ldots, N-1$$

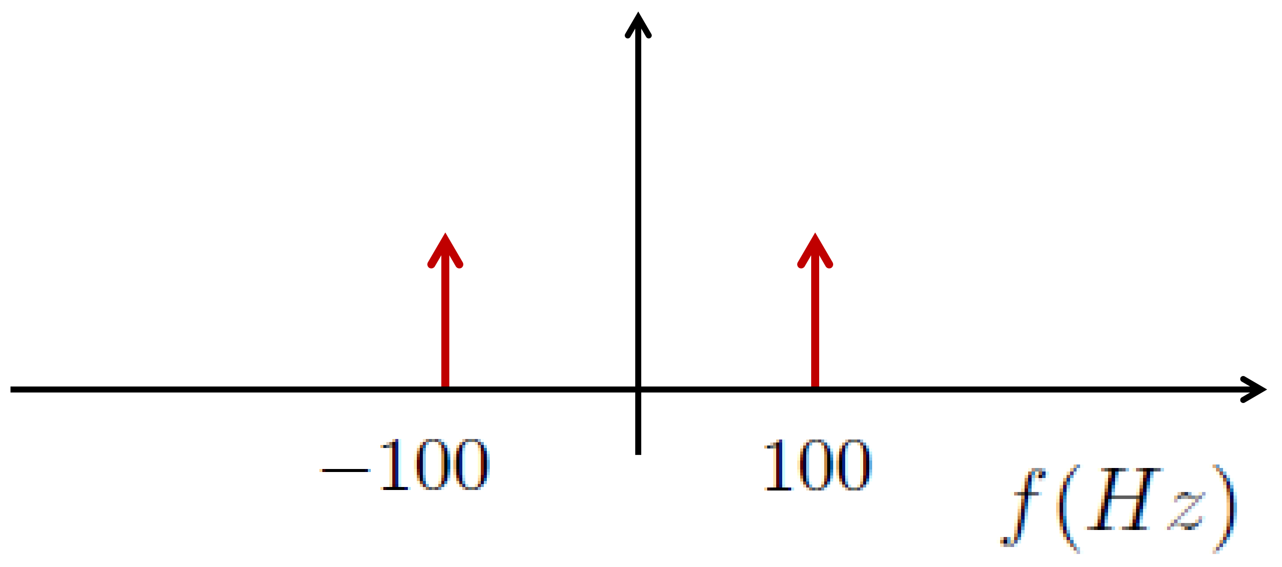
Exemple

$$x(t) = \cos(2\pi f_0 t)\,; \;\; f_0 = 100 \; Hz$$

TF du signal :

$$X(f) = \frac{1}{2}\left(\delta(f+f_0) + \delta(f-f_0)\right)$$

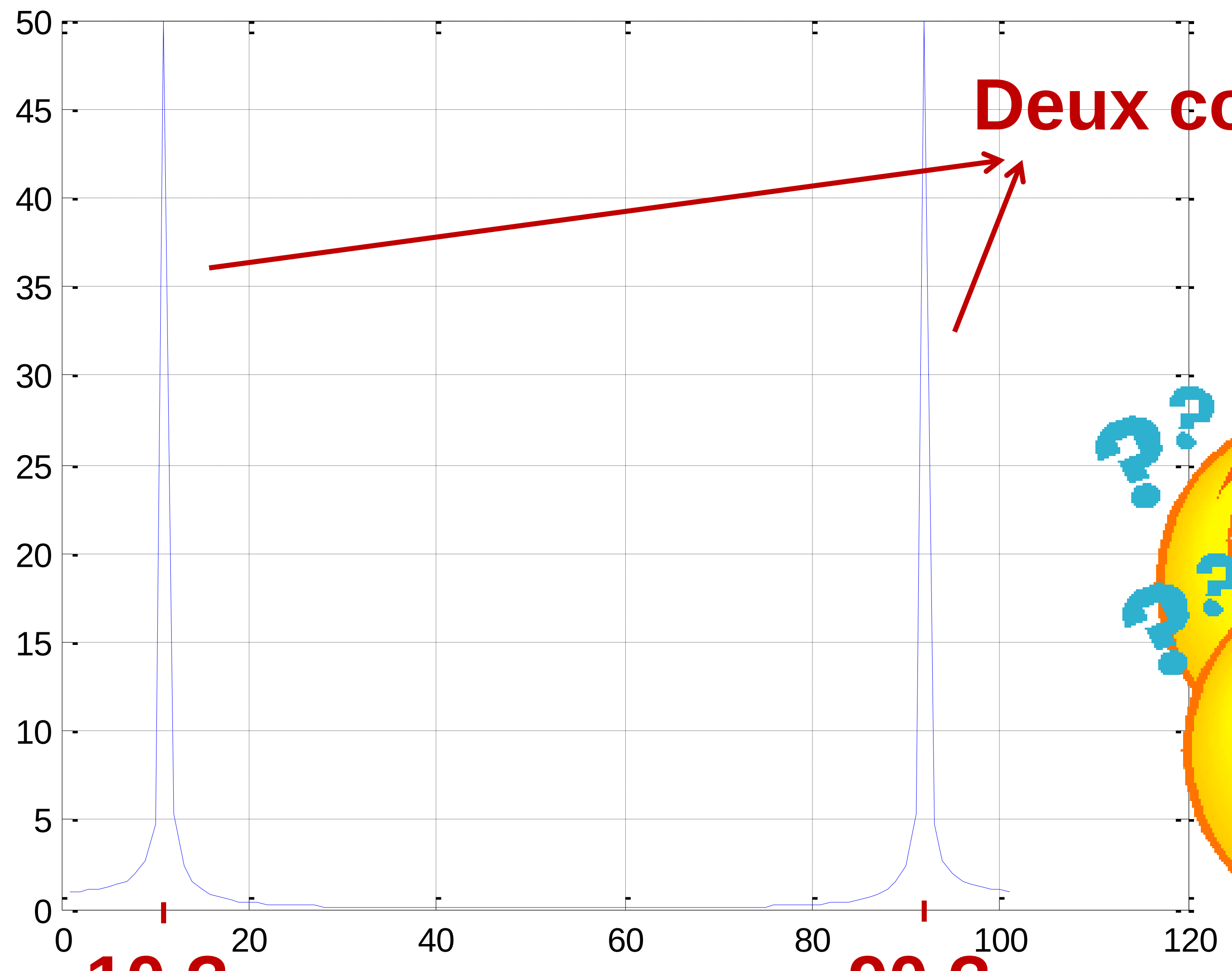
Tracé du module de la TF du signal :



Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;   %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

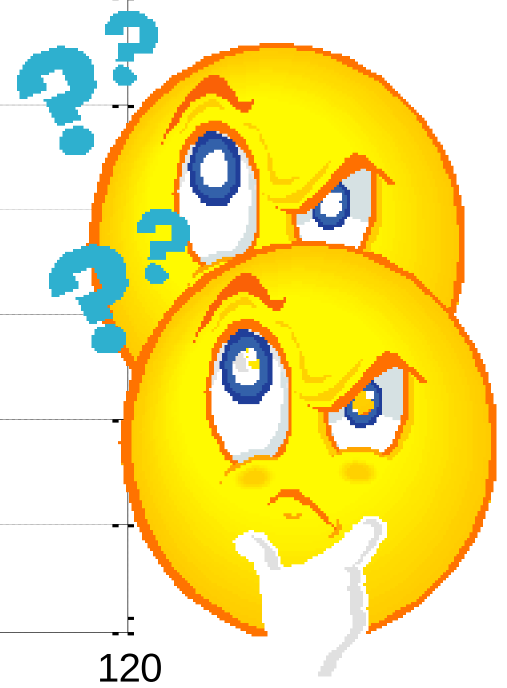
Tracé du module de la TFD obtenu :



Deux cosinus ?

10 ?

90 ?



de fréquences 10 et 90 Hz ?

Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Quels impacts ??

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

1- Echantillonnage temporel

$$x(t) \;\rightarrow\; \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

2- Durée limitée du signal

$$x(t) \rightarrow x_L(t) = \begin{cases} x(t) & \textit{pour } t \in [0, L] \\ 0 & \textit{ailleurs} \end{cases}$$

1+2 => Echantillonnage temporel + durée de signal limitée

$$x(t) \rightarrow \{x(kT_e)\}_{k=0,...,N-1}$$

3- Echantillonnage fréquentiel

$$X(f) \;\rightarrow\; \{X(n\Delta f)\}_{n=0,...,N-1}$$

Transformée de Fourier Discrète

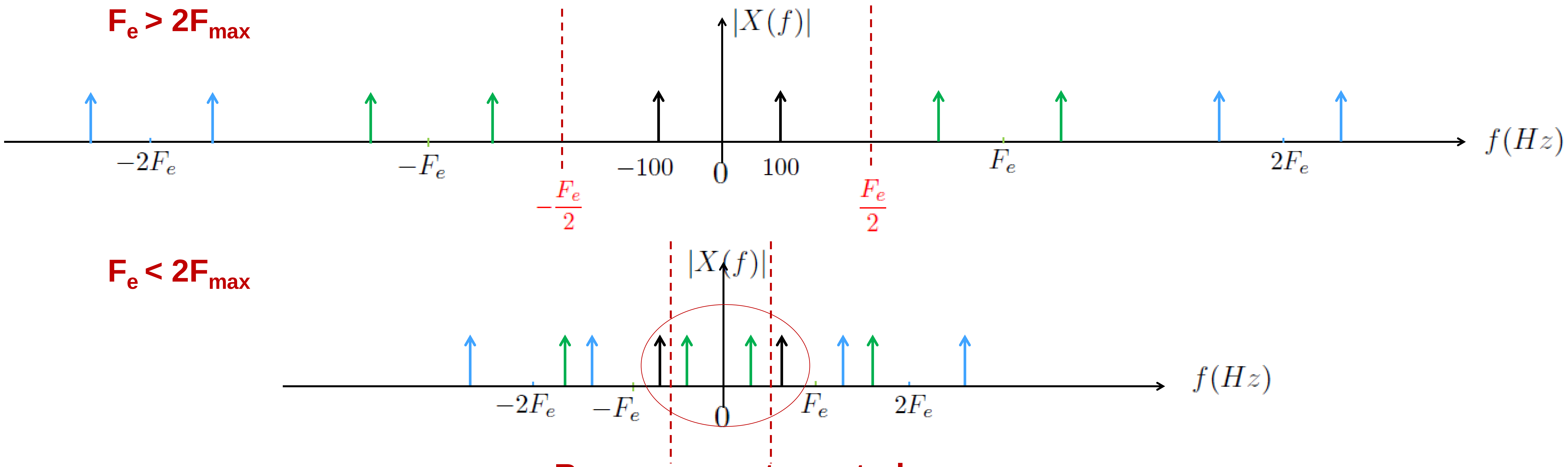
Echantillonnage temporel

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

$F_e > 2F_{max}$

$F_e < 2F_{max}$



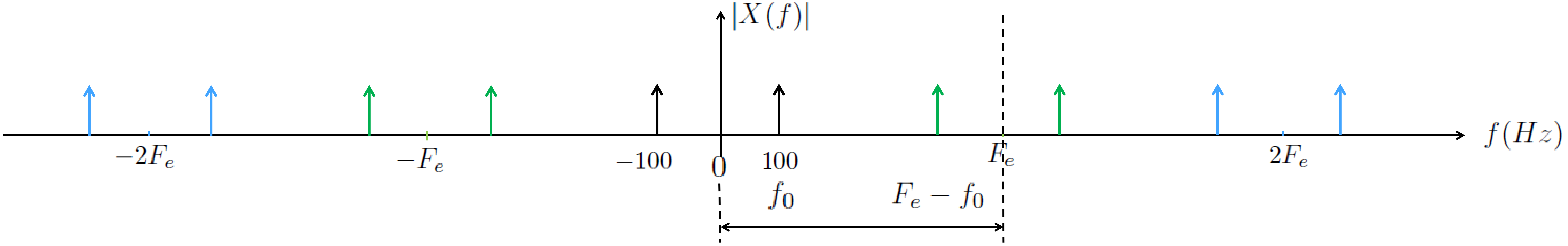
Recouvrement spectral (Aliasing)

Transformée de Fourier Discrète

Echantillonnage temporel

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$



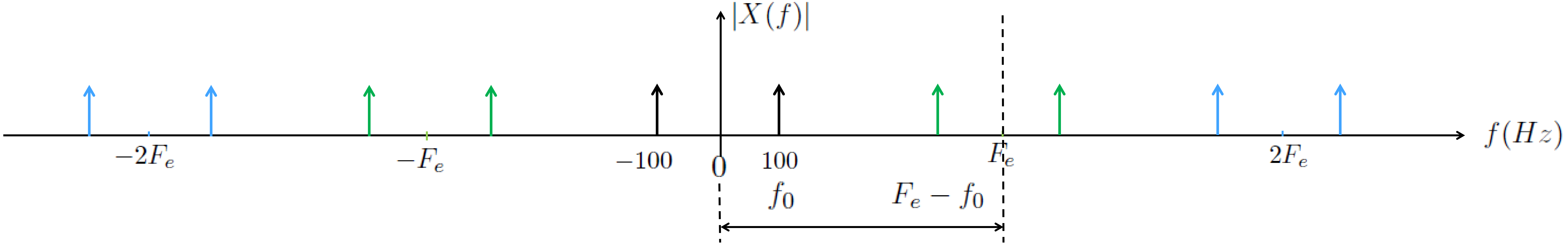
Visualisation sur [0, F_e] :

Transformée de Fourier Discrète

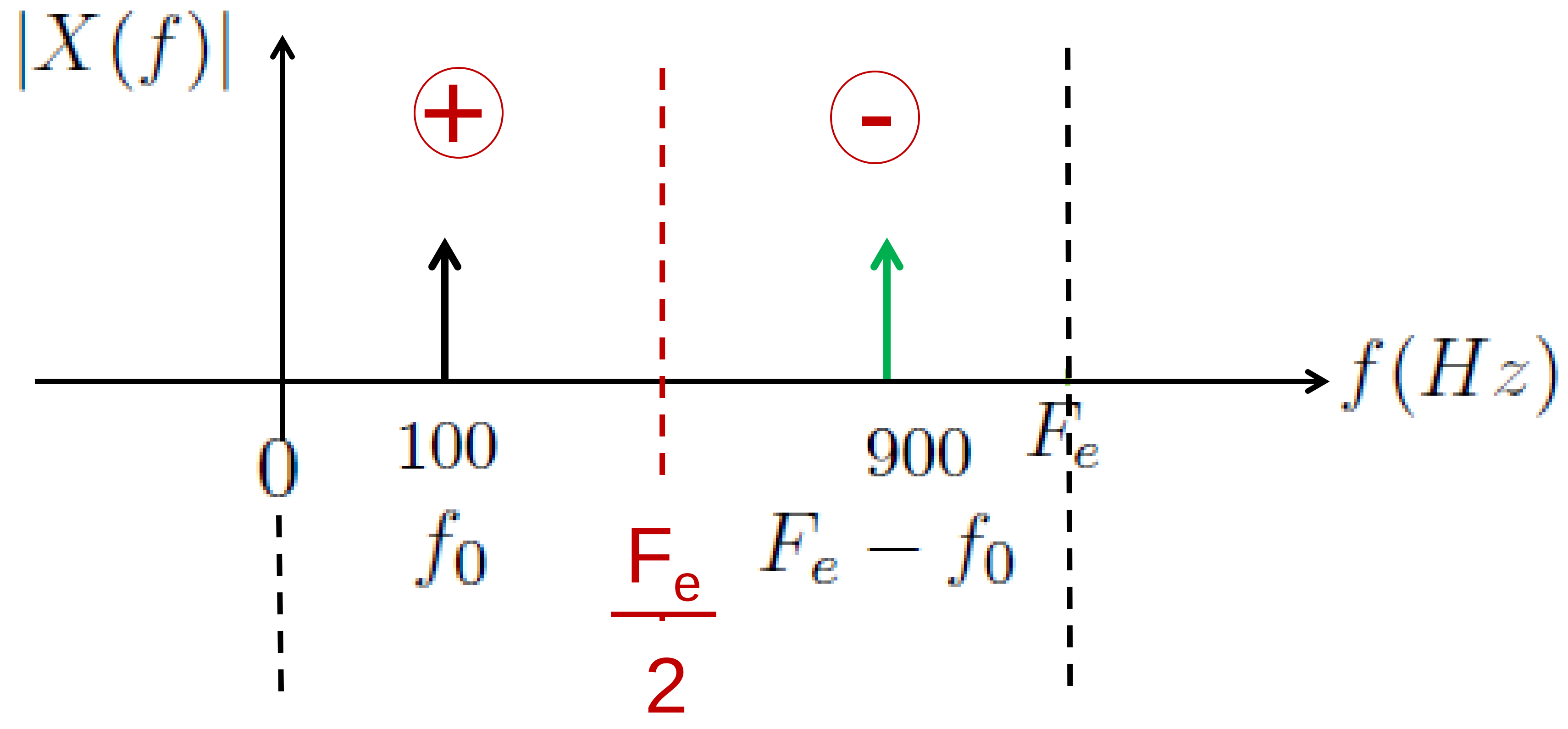
Echantillonnage temporel

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$



Visualisation sur [0, F_e] :



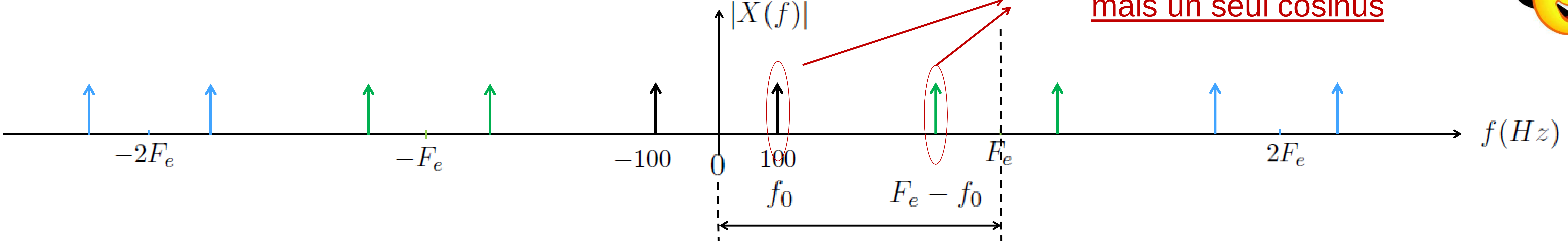
Transformée de Fourier Discrète

Echantillonnage temporel

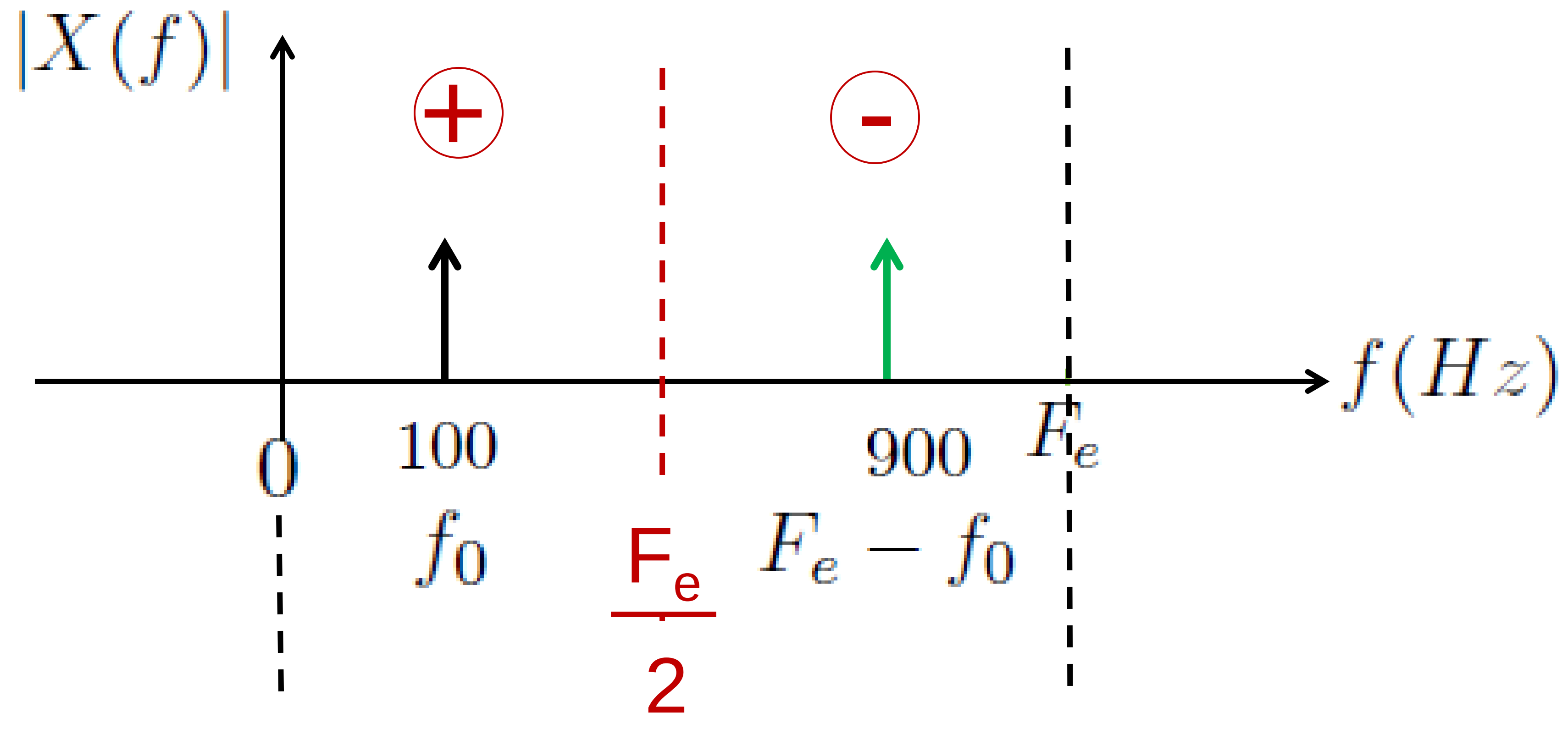
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

Deux diracs côté fréquences positives
mais un seul cosinus



Visualisation sur [0, F_e] :



Transformée de Fourier Discrète

Echantillonnage temporel

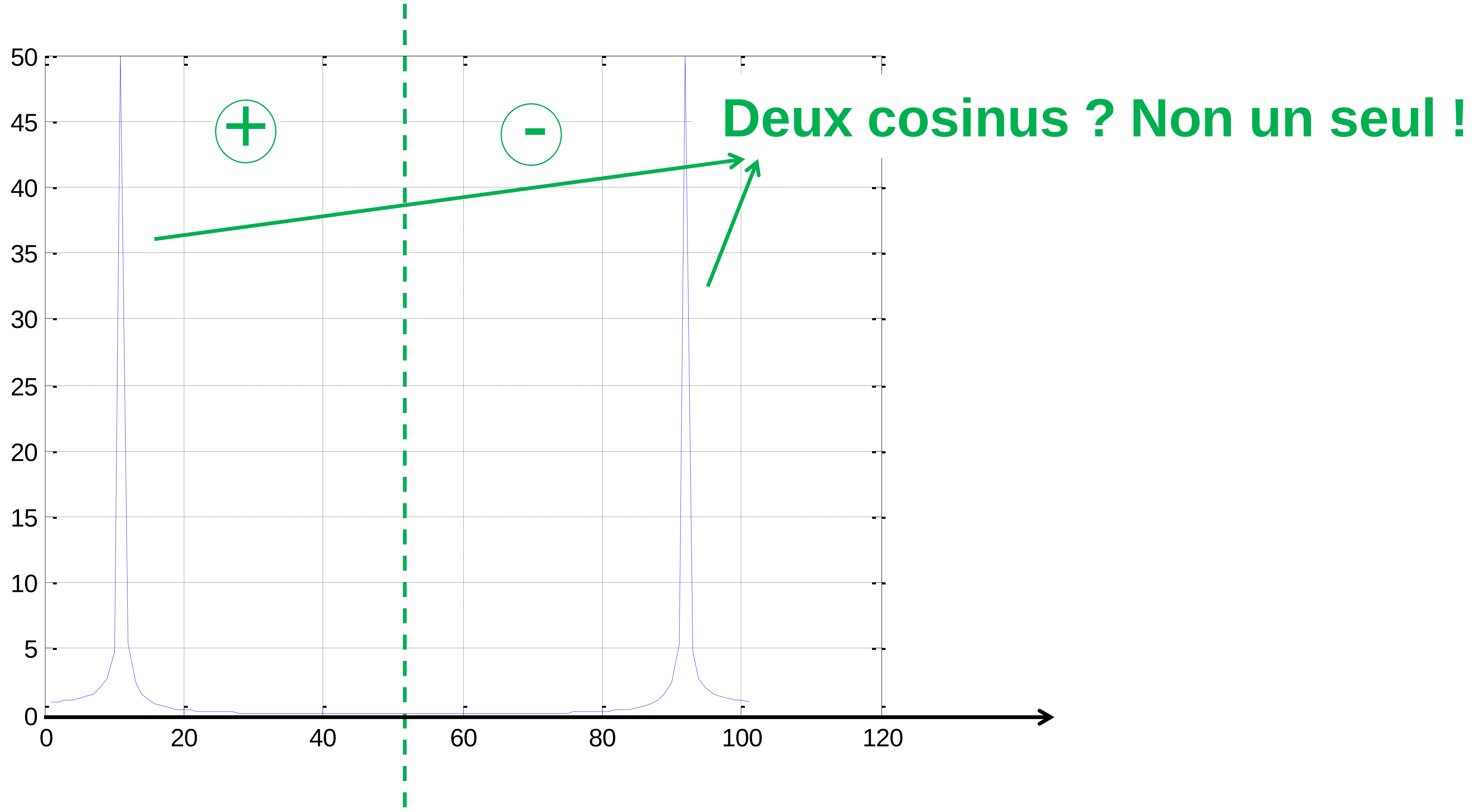
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;   %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;        %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Quels impacts ??

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

1- Echantillonnage temporel

$$x(t) \; \rightarrow \; \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

$\Rightarrow$ Périodisation de la TFD

$\Rightarrow$!! Respecter la condition de Shannon !!

$\Rightarrow$!! Lecture des tracés !!

Transformée de Fourier Discrète

!! Echelle fréquentielle !!

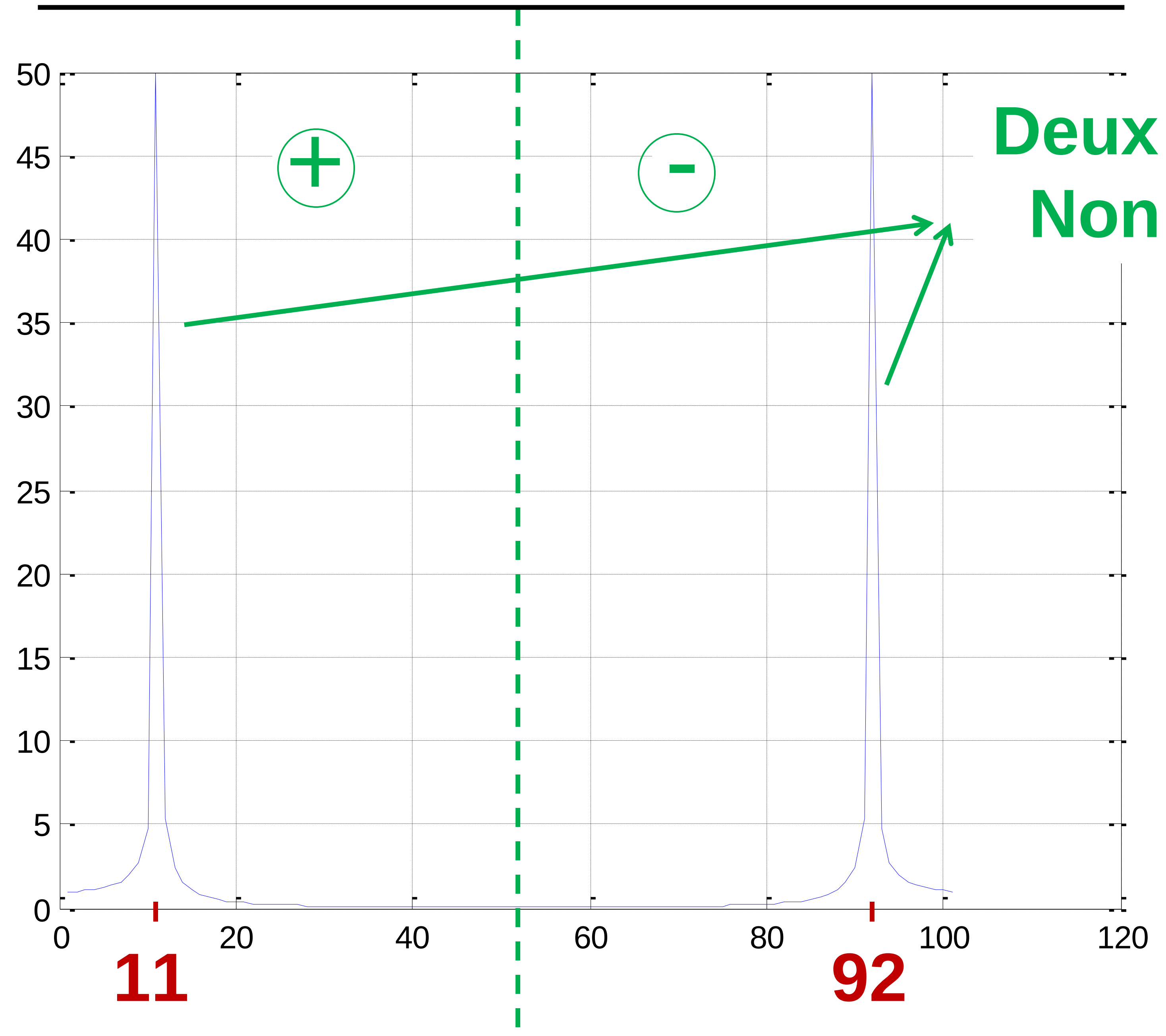
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;     %fréquence du cosinus
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;      %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Deux cosinus ?
Non un seul !

fréquences f_0=11 Hz et F_e- f_0=92 Hz ??

Transformée de Fourier Discrète

!! Echelle fréquentielle !!

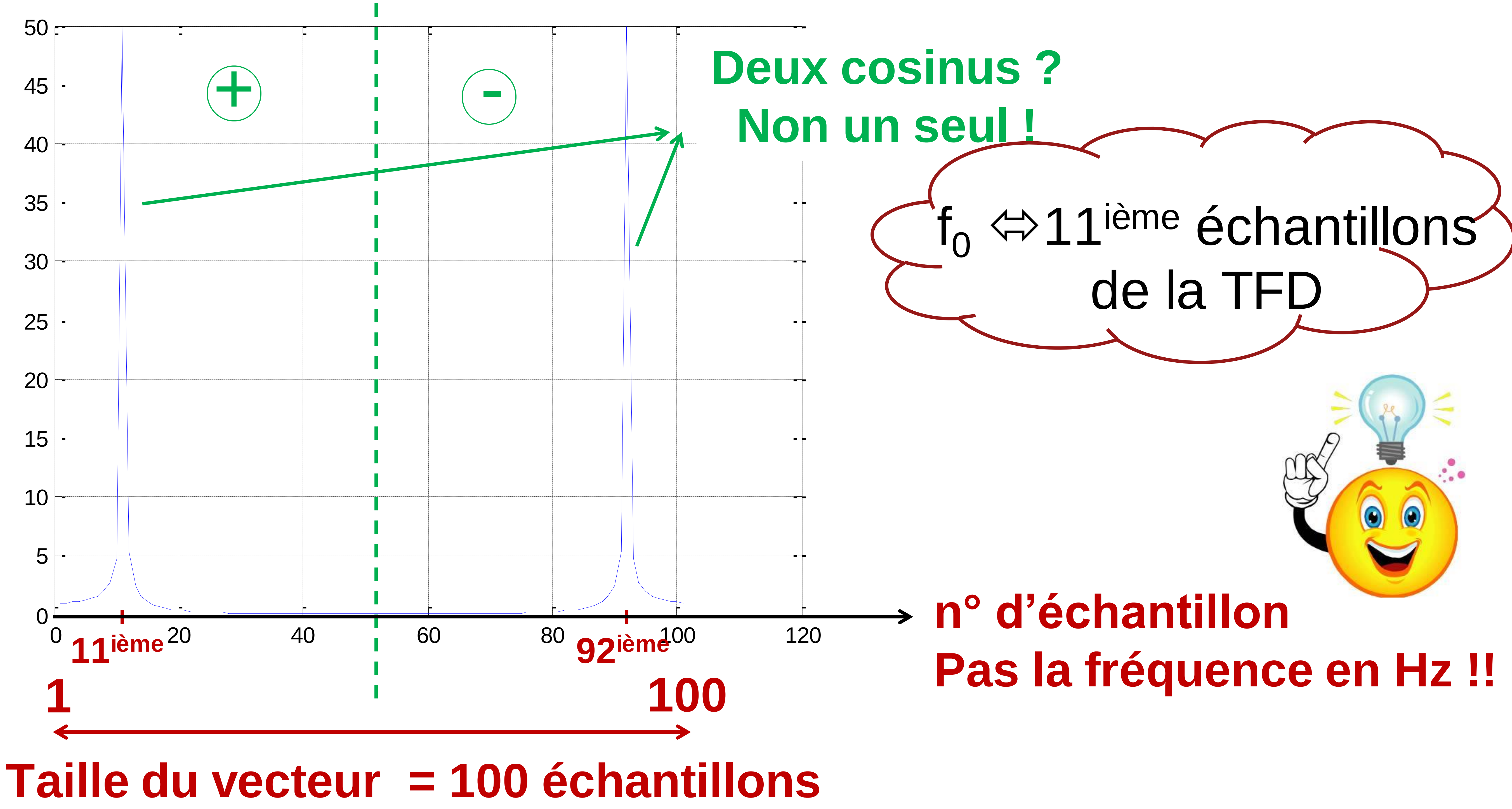
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure;  plot(abs(X))
```

Tracé du module de la TFD obtenu :



Deux cosinus ?
Non un seul !

$f_0 \Leftrightarrow 11^{\text{ième}}$ échantillons de la TFD

Transformée de Fourier Discrète

!! Echelle fréquentielle !!

Exemple

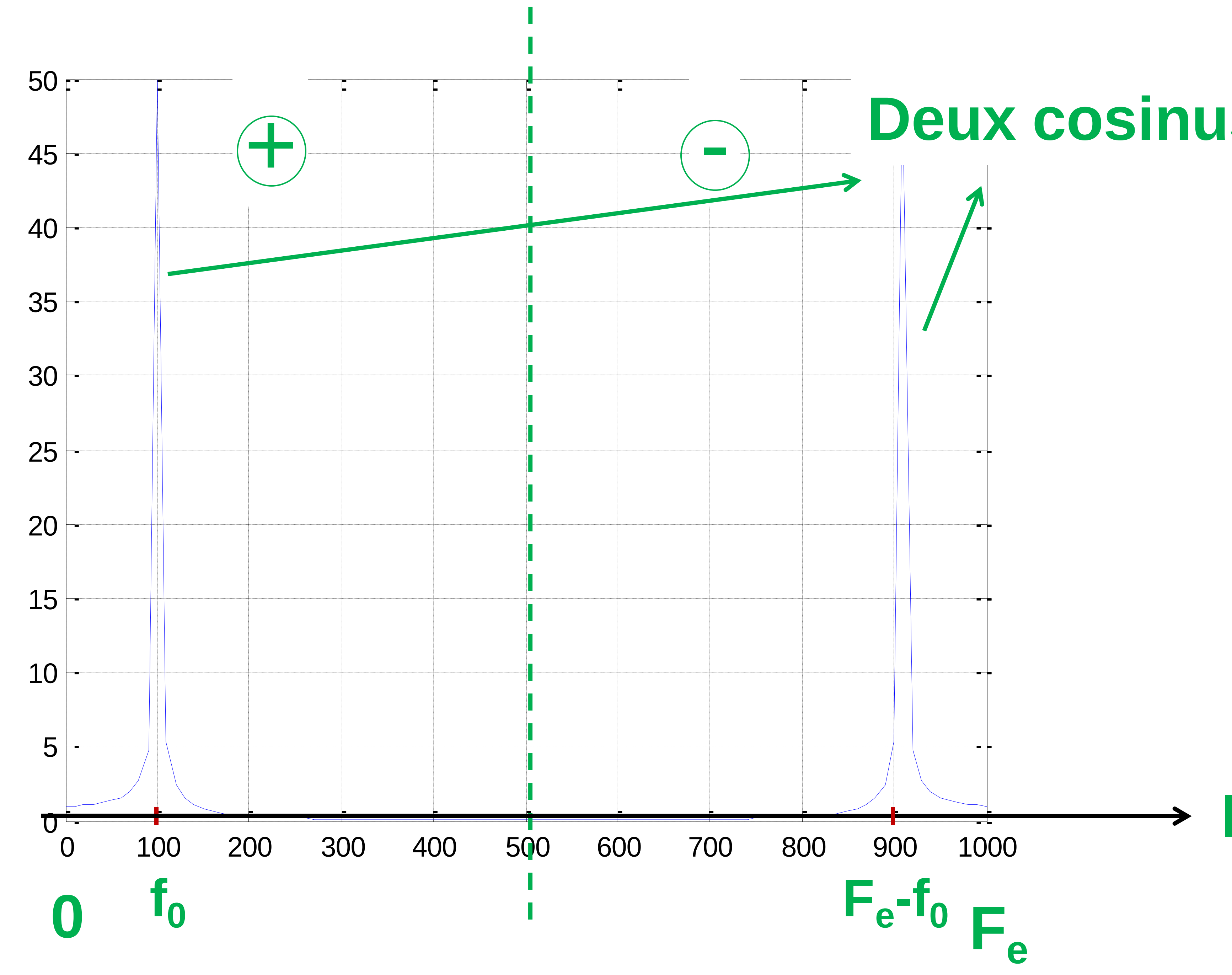
$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure; plot(linspace(0,Fe,length(X)),abs(X))
```

!! Echelle fréquentielle à donner !!

Tracé du module de la TFD obtenu :



Deux cosinus ? Non un seul !

Fréquence (Hz)

f_0 = 100 Hz

Transformée de Fourier Discrète

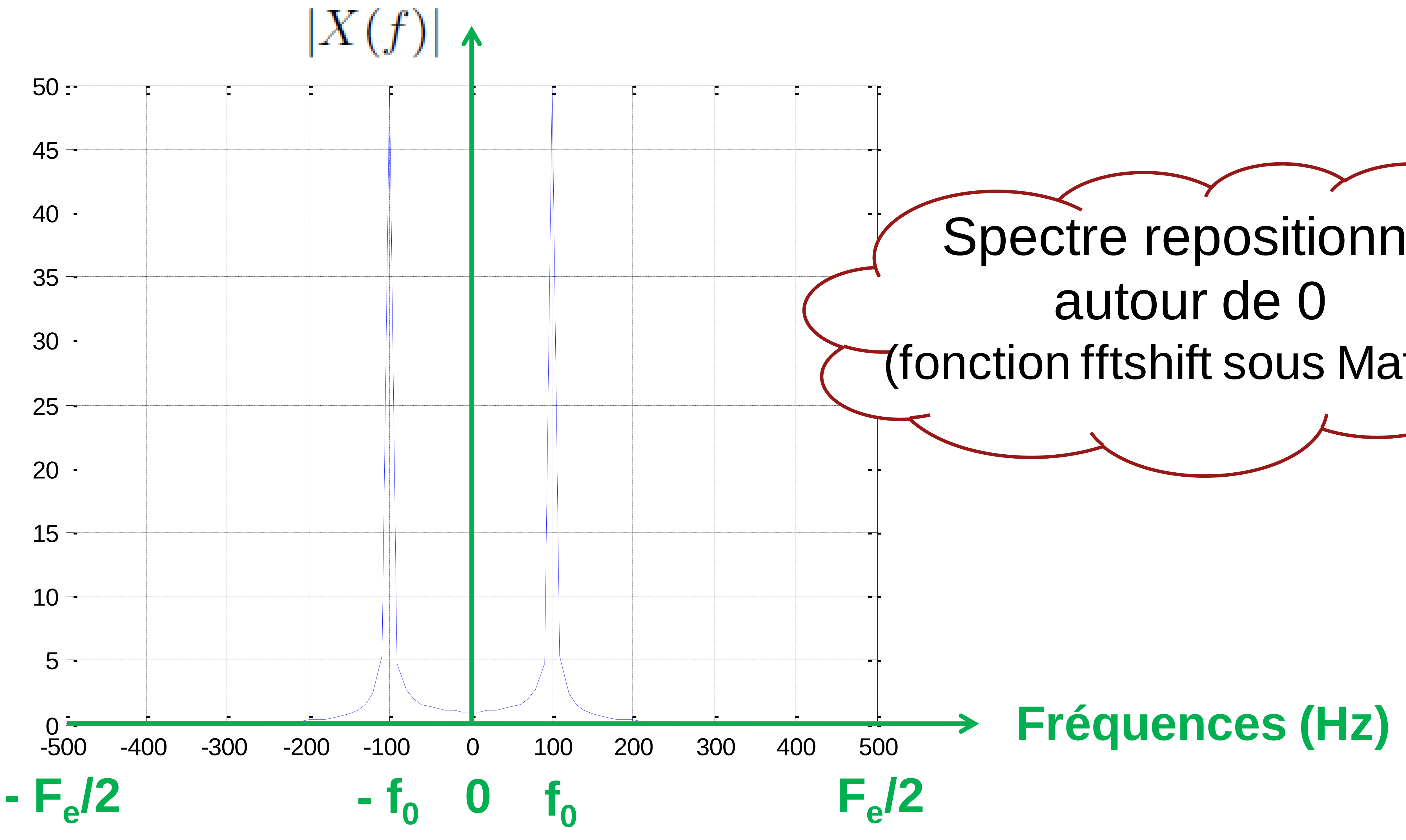
!! Echelle fréquentielle !!

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure; plot(linspace(-Fe/2,Fe/2,length(X)),fftshift(abs(X)))
```



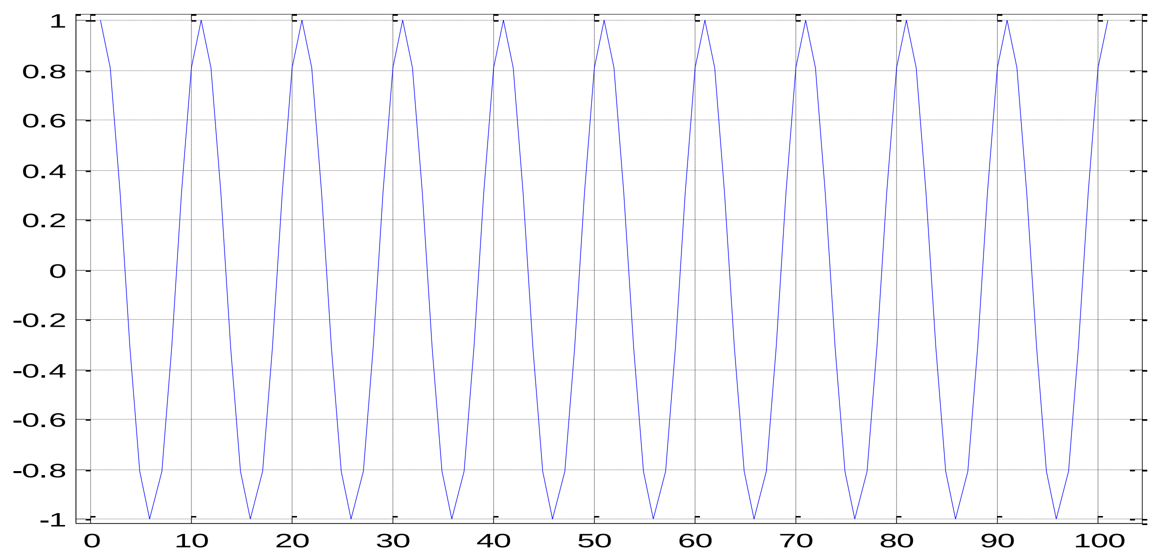
Spectre repositionné autour de 0
(fonction fftshift sous Matlab)

Transformée de Fourier Discrète

Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$



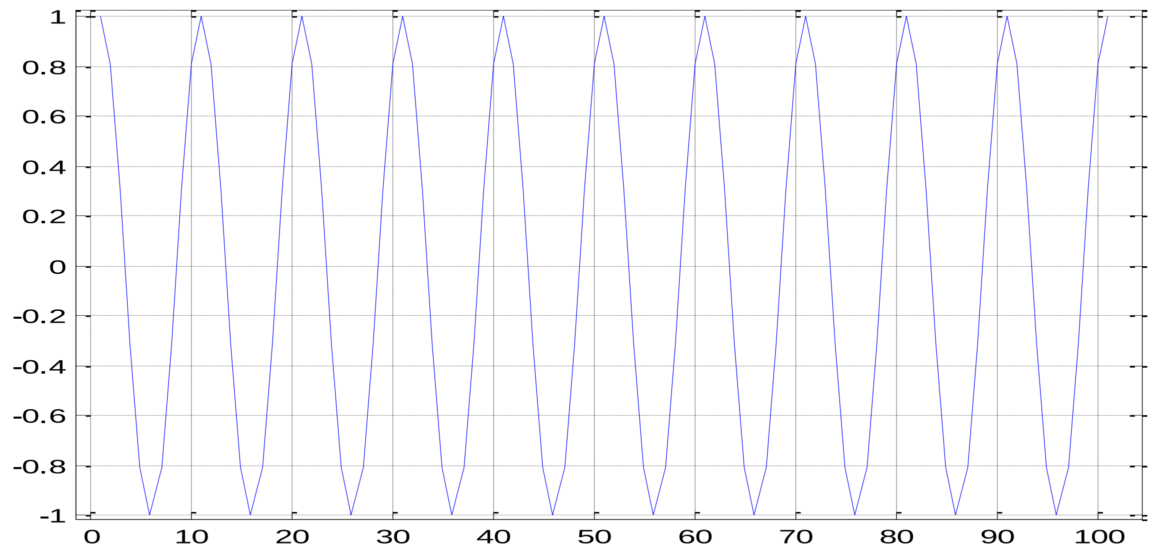
Signal de durée limitée

Transformée de Fourier Discrète

Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\ ;\ \ f_0 = 100\ Hz$$



Signal de durée limitée

Signal de durée illimitée

x Fenêtre de troncature

Transformée de Fourier Discrète

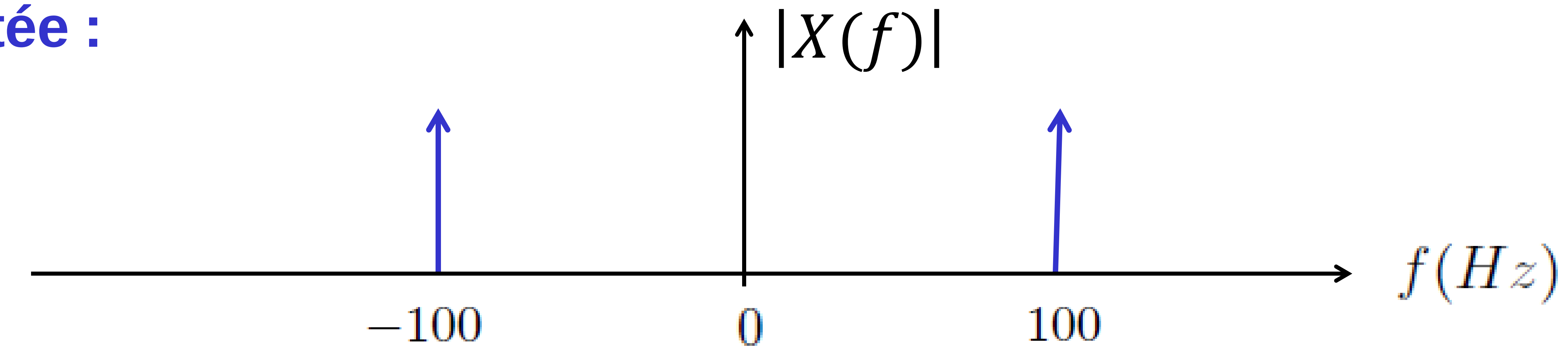
Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

TF du signal de durée illimitée :

$$x(t) \xrightarrow{TF} X(f)$$



Transformée de Fourier Discrète

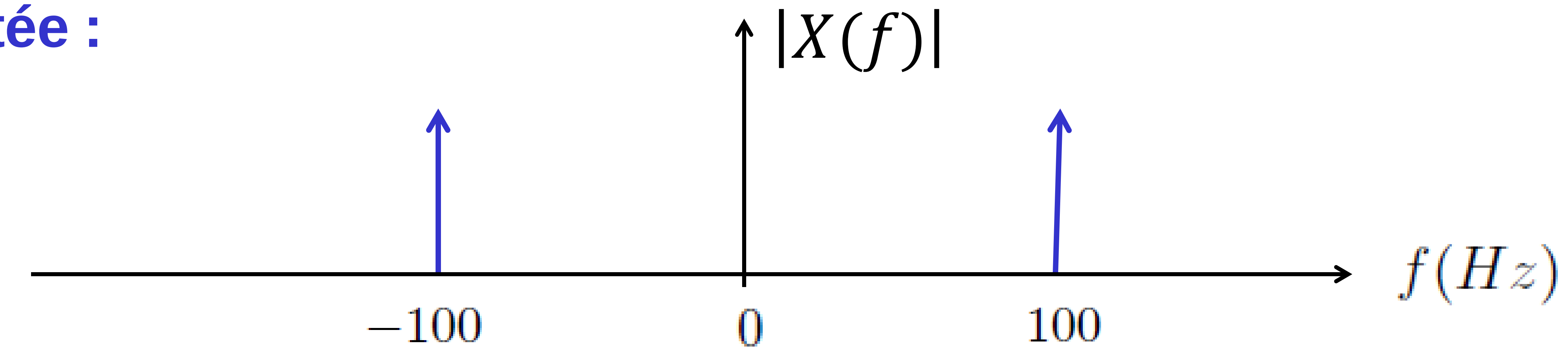
Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

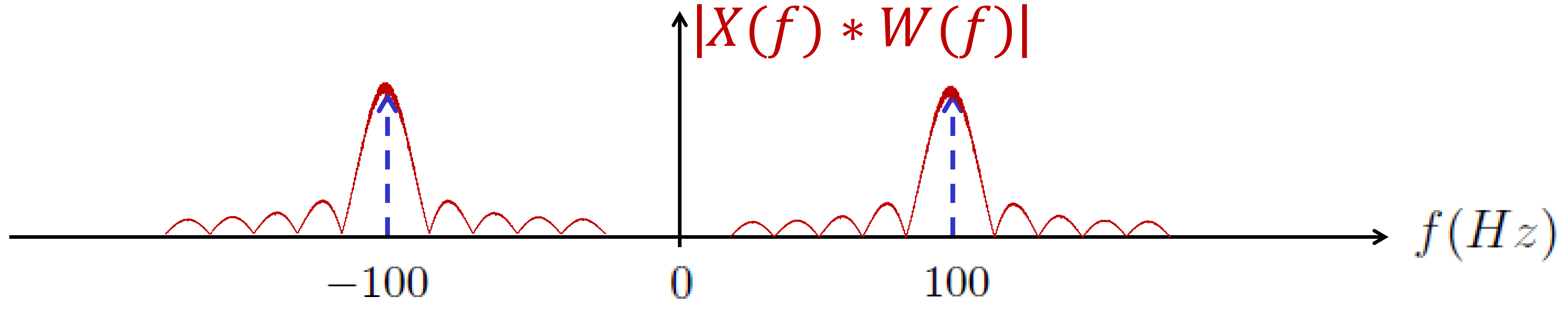
TF du signal de durée illimitée :

$$x(t) \xrightarrow{TF} X(f)$$



TF du signal à durée limitée = TF du signal à durée illimitée * TF de la fenêtre modélisant la troncature :

$$x(t)w(t) \xrightarrow{TF} X(f) * W(f)$$



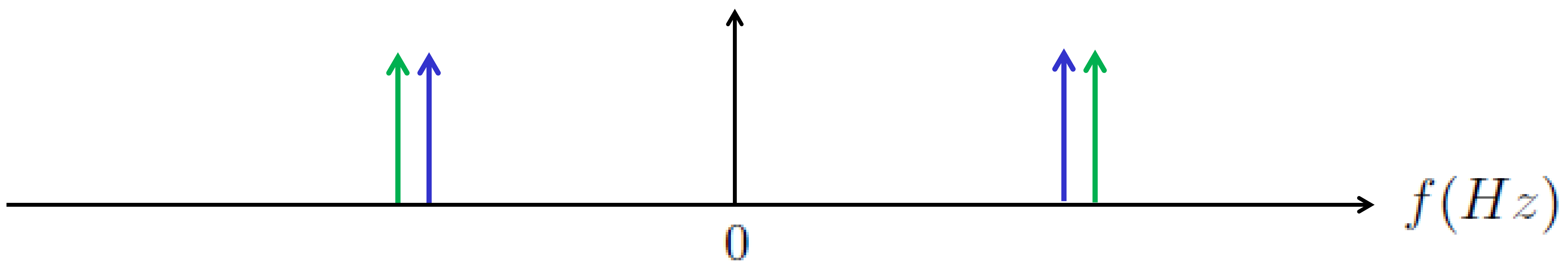
Transformée de Fourier Discrète

Signal de durée limitée

Problèmes posés :

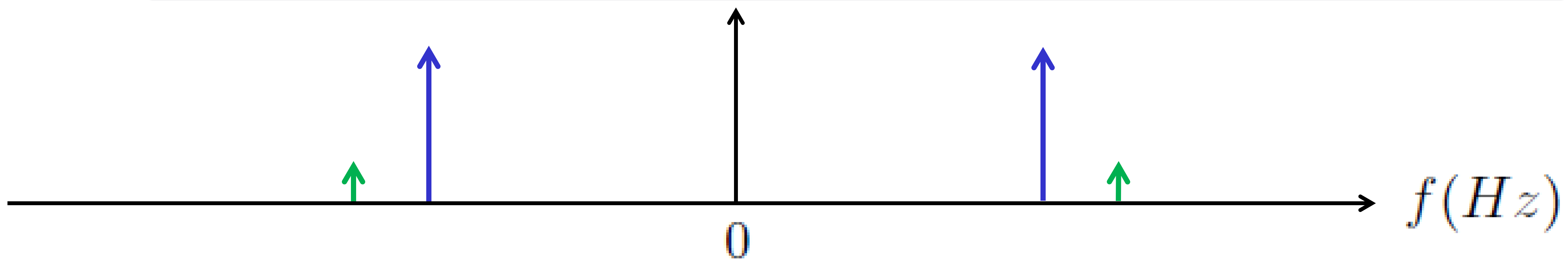
Exemple 1

Somme de deux cosinus proches en fréquences



Exemple 2

Somme de deux cosinus dont un de faible puissance



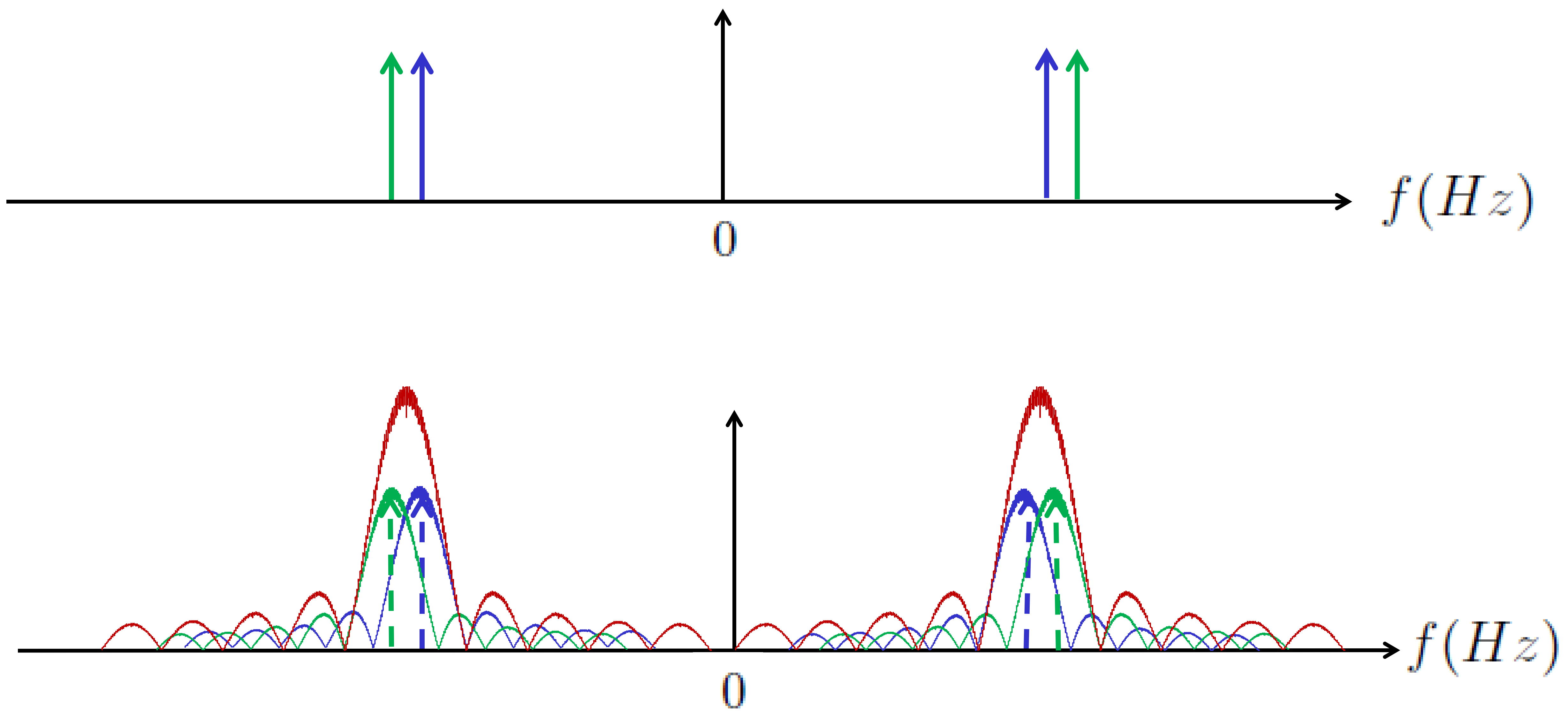
Transformée de Fourier Discrète

Signal de durée limitée

<u>Problèmes posés :</u>

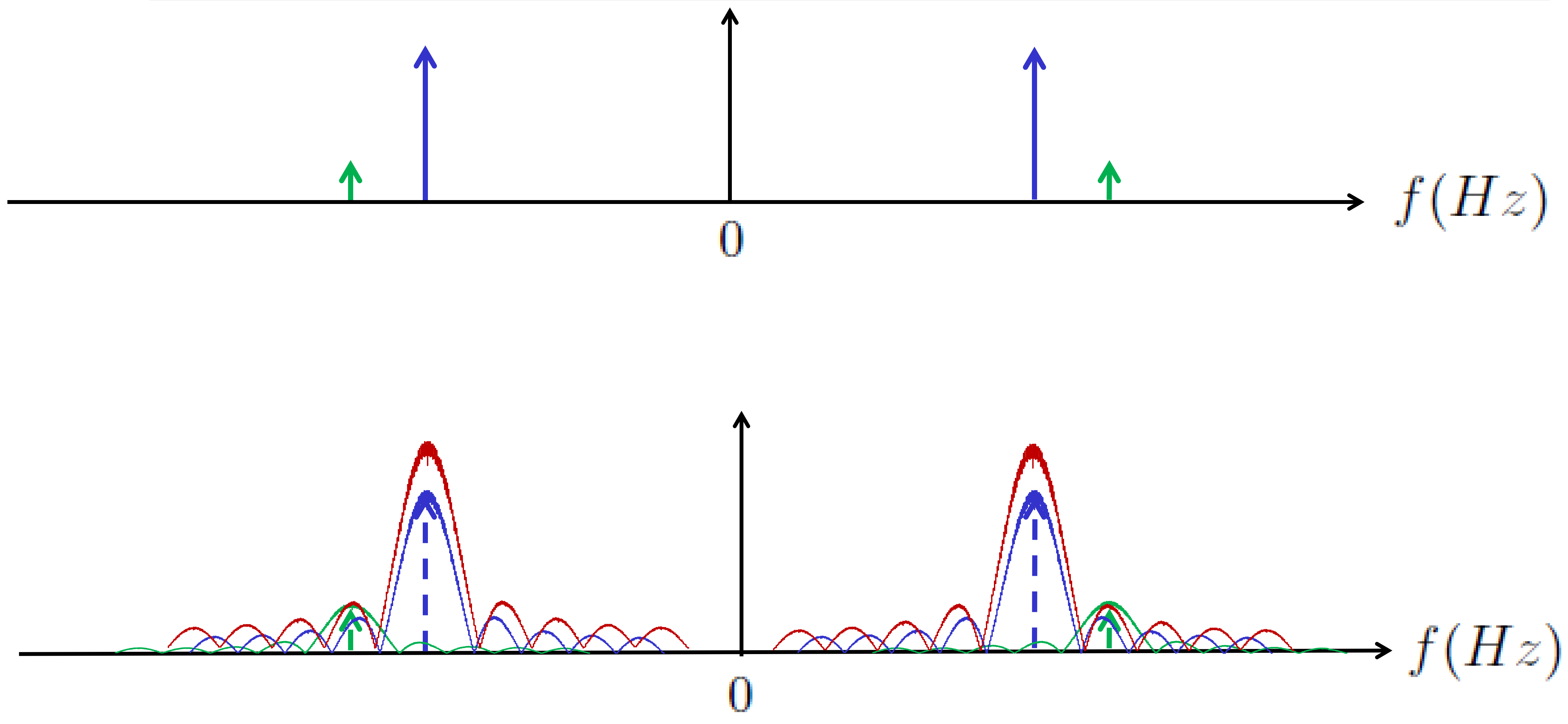
Exemple 1

Somme de deux cosinus proches en fréquences



Exemple 2

Somme de deux cosinus dont un de faible puissance



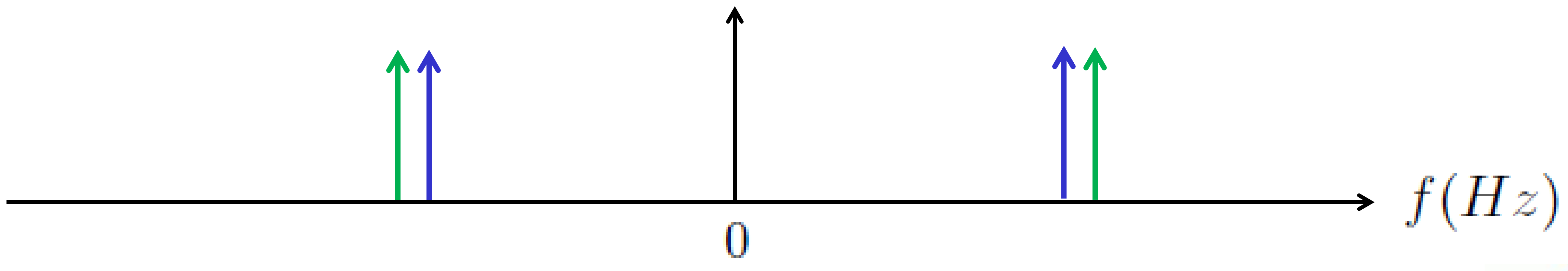
Transformée de Fourier Discrète

Signal de durée limitée

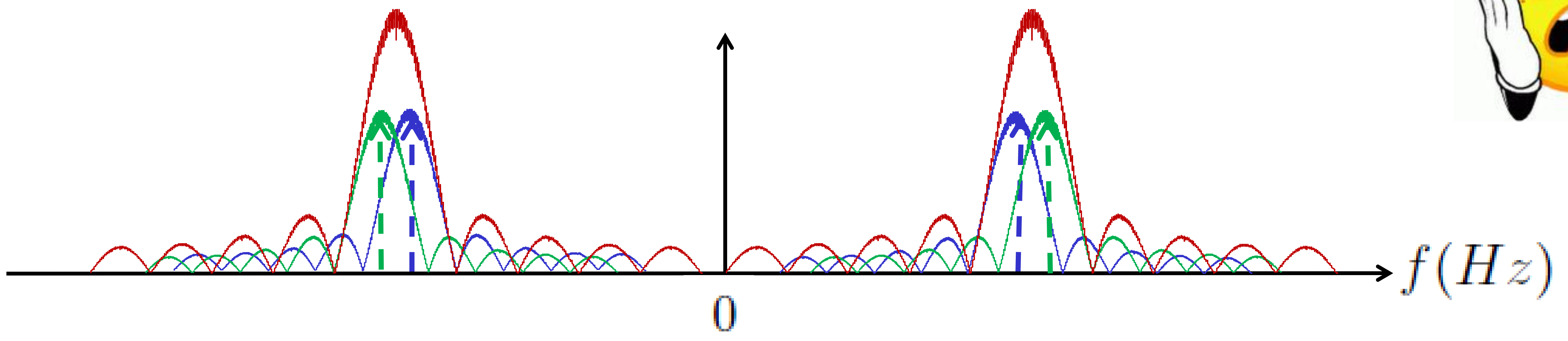
Problèmes posés :

Exemple 1

Somme de deux cosinus proches en fréquences

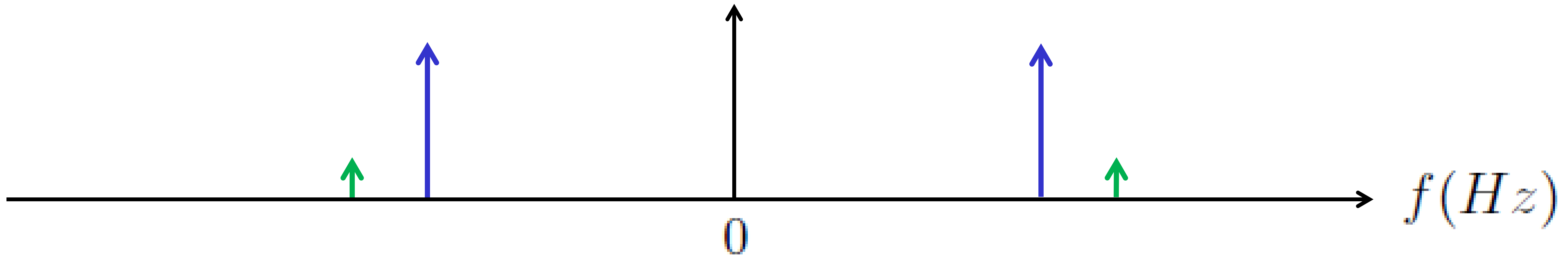


!! Un seul cosinus !!

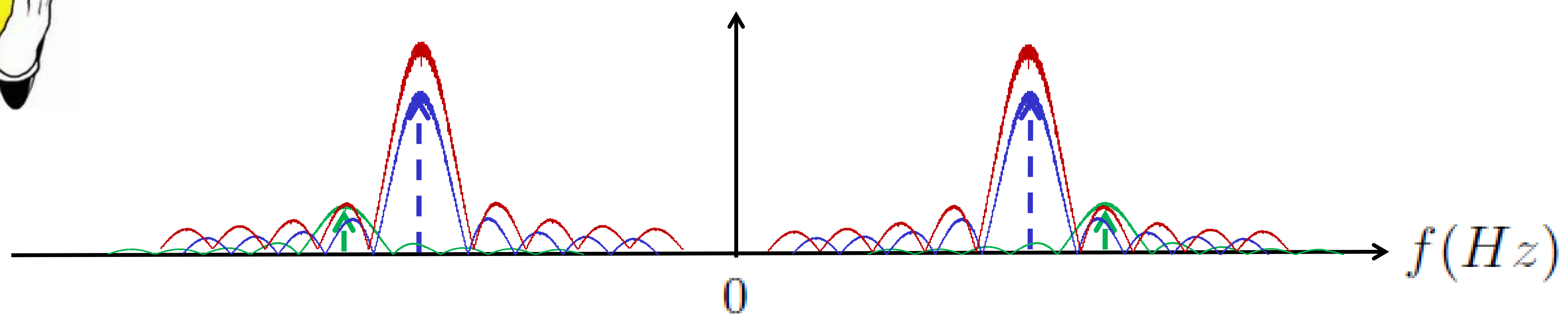


Exemple 2

Somme de deux cosinus dont un de faible puissance



!! Un seul cosinus !!



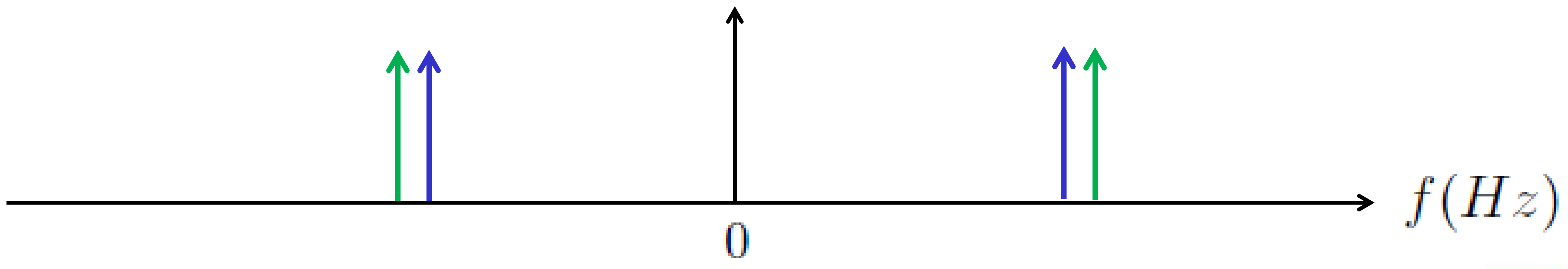
Transformée de Fourier Discrète

Signal de durée limitée

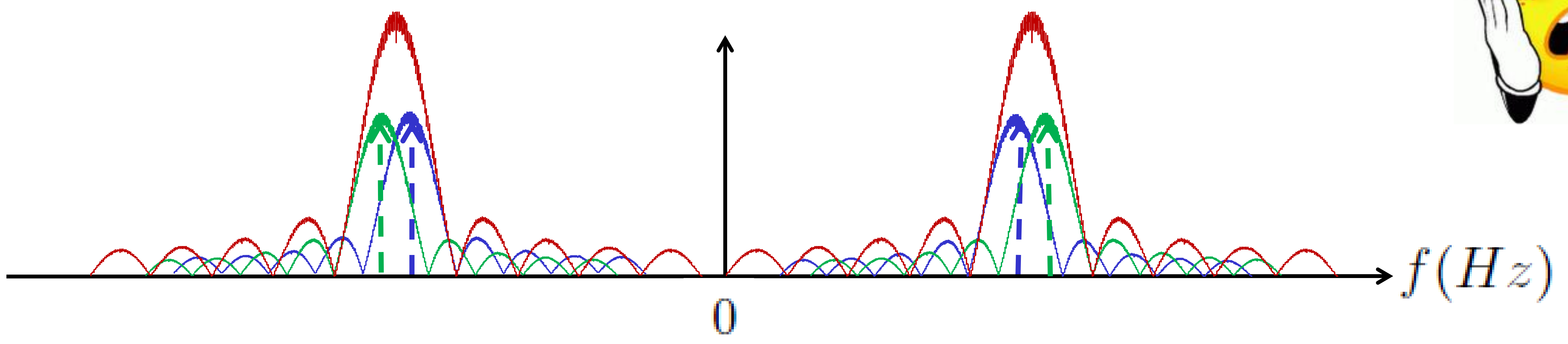
Problèmes posés :

Exemple 1

Somme de deux cosinus proches en fréquences

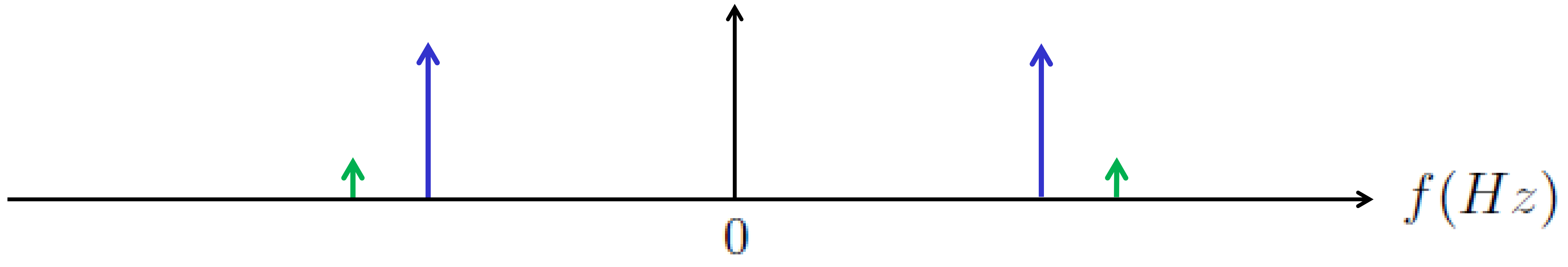


!! Un seul cosinus !!

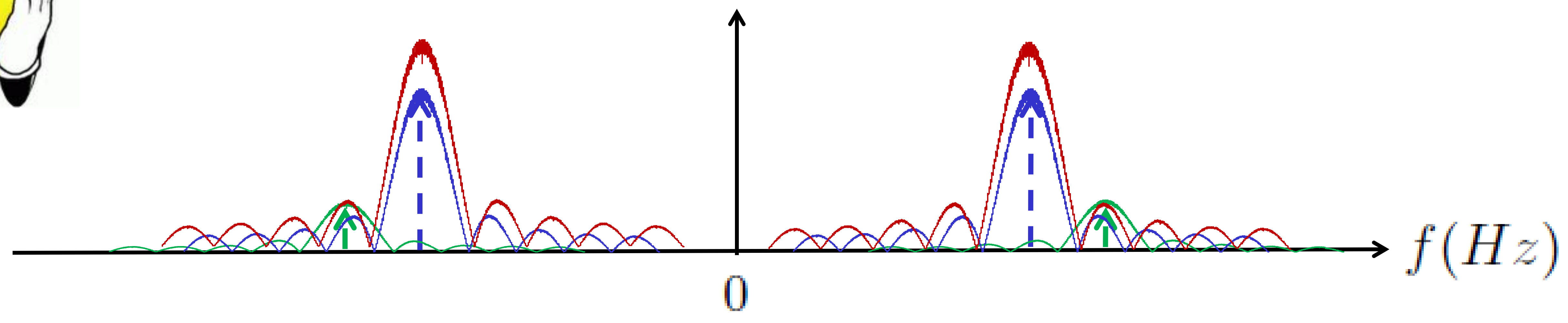


Exemple 2

Somme de deux cosinus dont un de faible puissance



!! Un seul cosinus !!



$\Rightarrow$ L'analyse spectrale NUMERIQUE aura :

un certain **pouvoir séparateur** et un certain **taux d'ondulation**

Capacité à séparer des motifs spectraux proches en fréquences

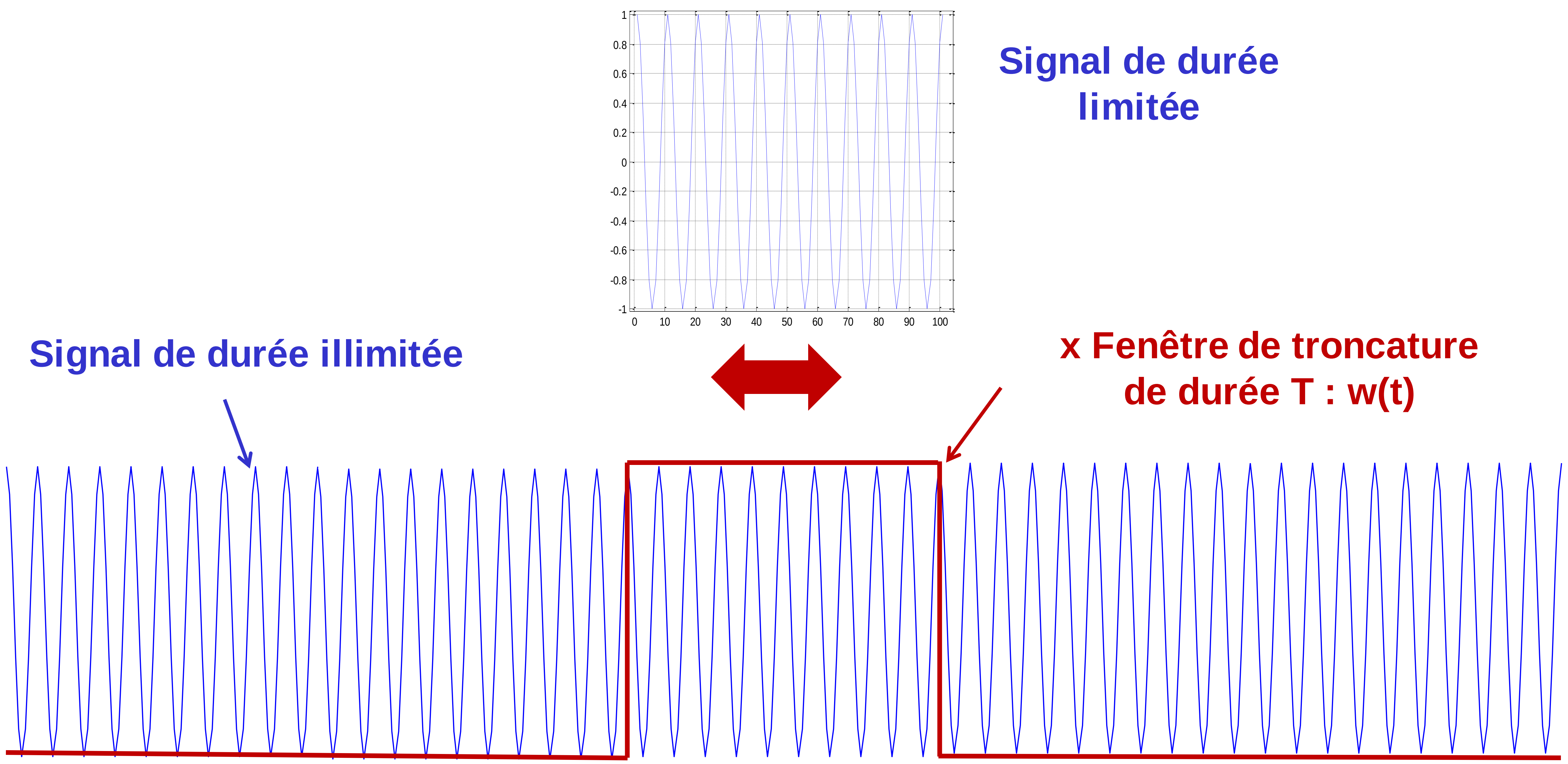
Masquage de motifs spectraux de faibles puissances

Transformée de Fourier Discrète

Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

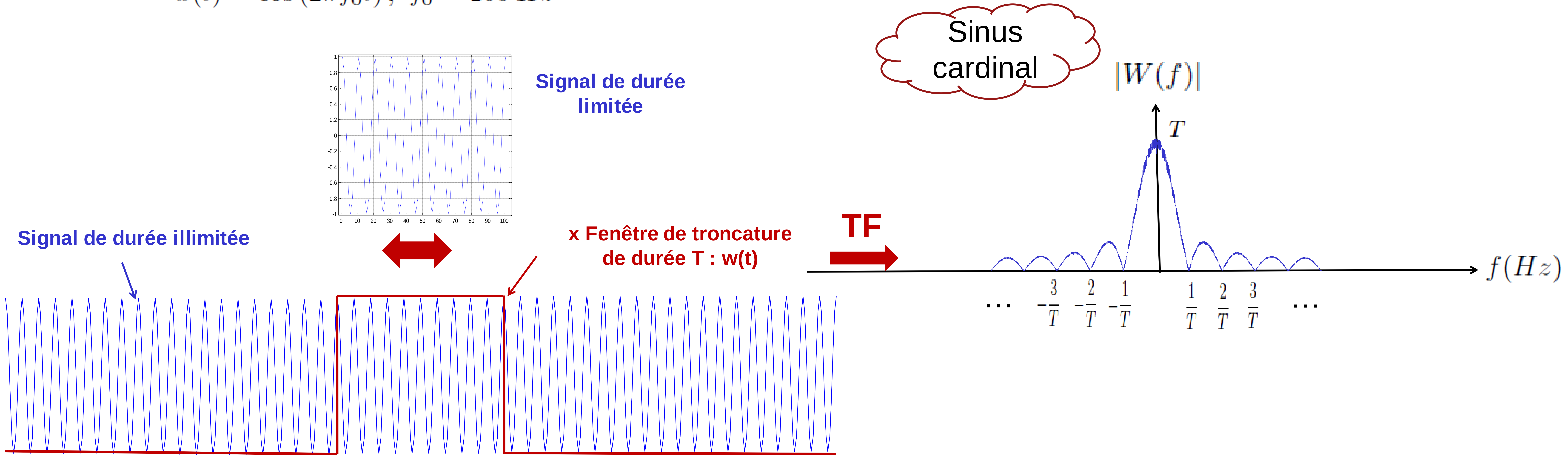


Transformée de Fourier Discrète

Signal de durée limitée

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

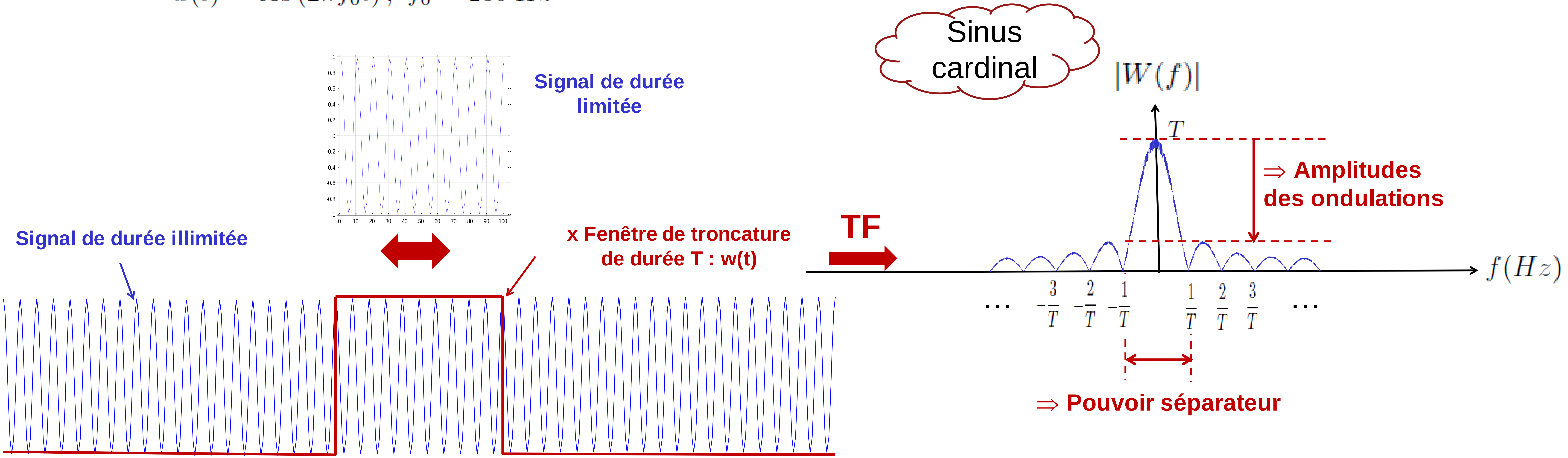


Transformée de Fourier Discrète

Signal de durée limitée

Exemple

$$x(t) = \cos\left(2\pi f_0 t\right)\,;\quad f_0 = 100\ Hz$$



Transformée de Fourier Discrète

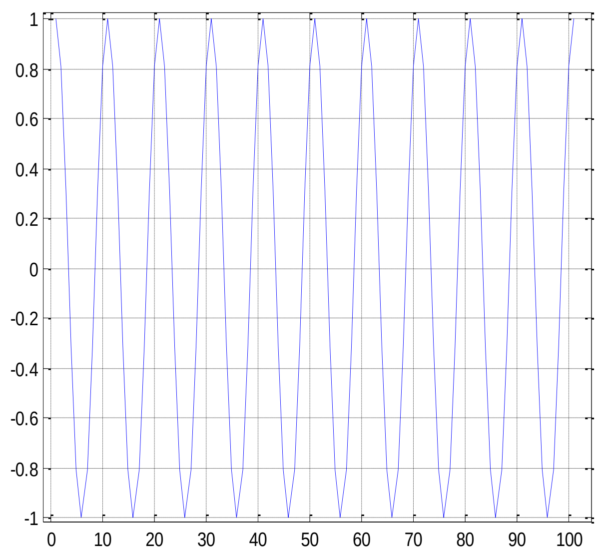
Signal de durée limitée

~~Sinus cardinal~~

Noyau de Dirichlet

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ \ f_0 = 100\ Hz$$

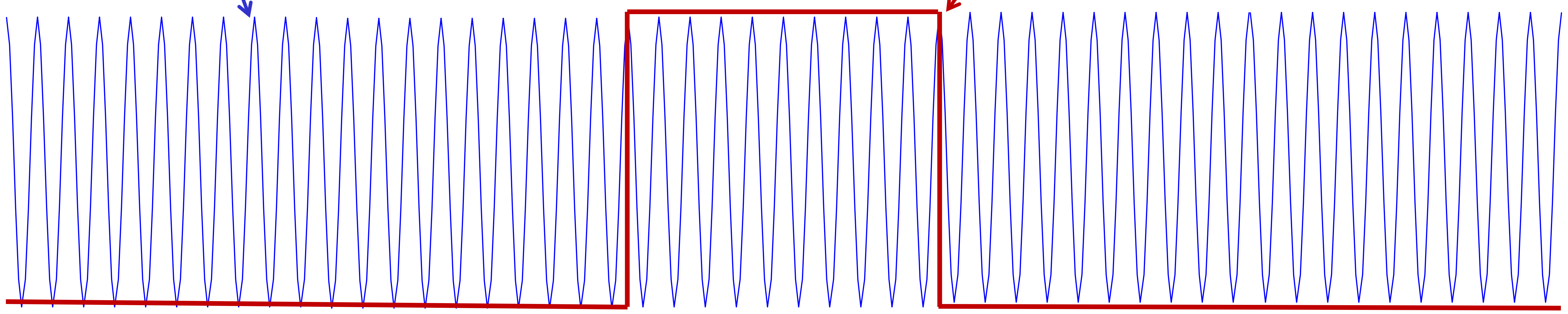


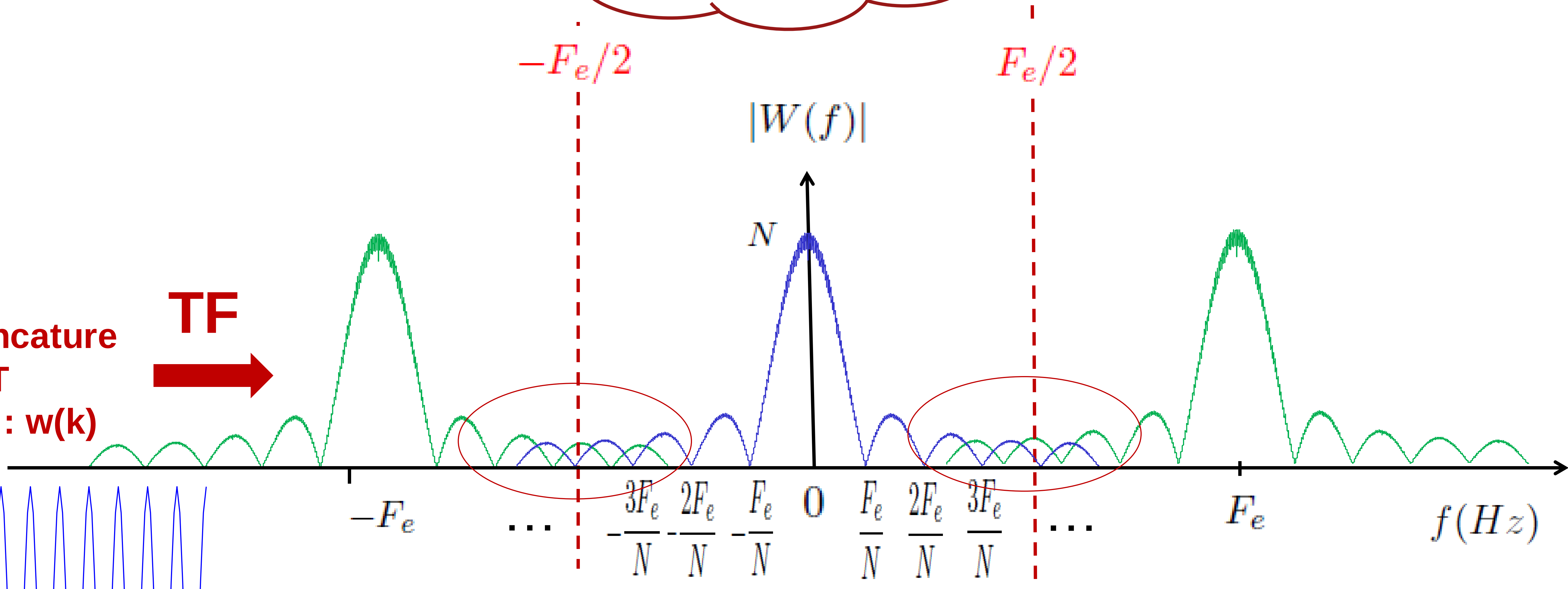
Signal de durée limitée

x Fenêtre de troncature de durée T Échantillonnée : w(k)

TF

Signal de durée illimitée



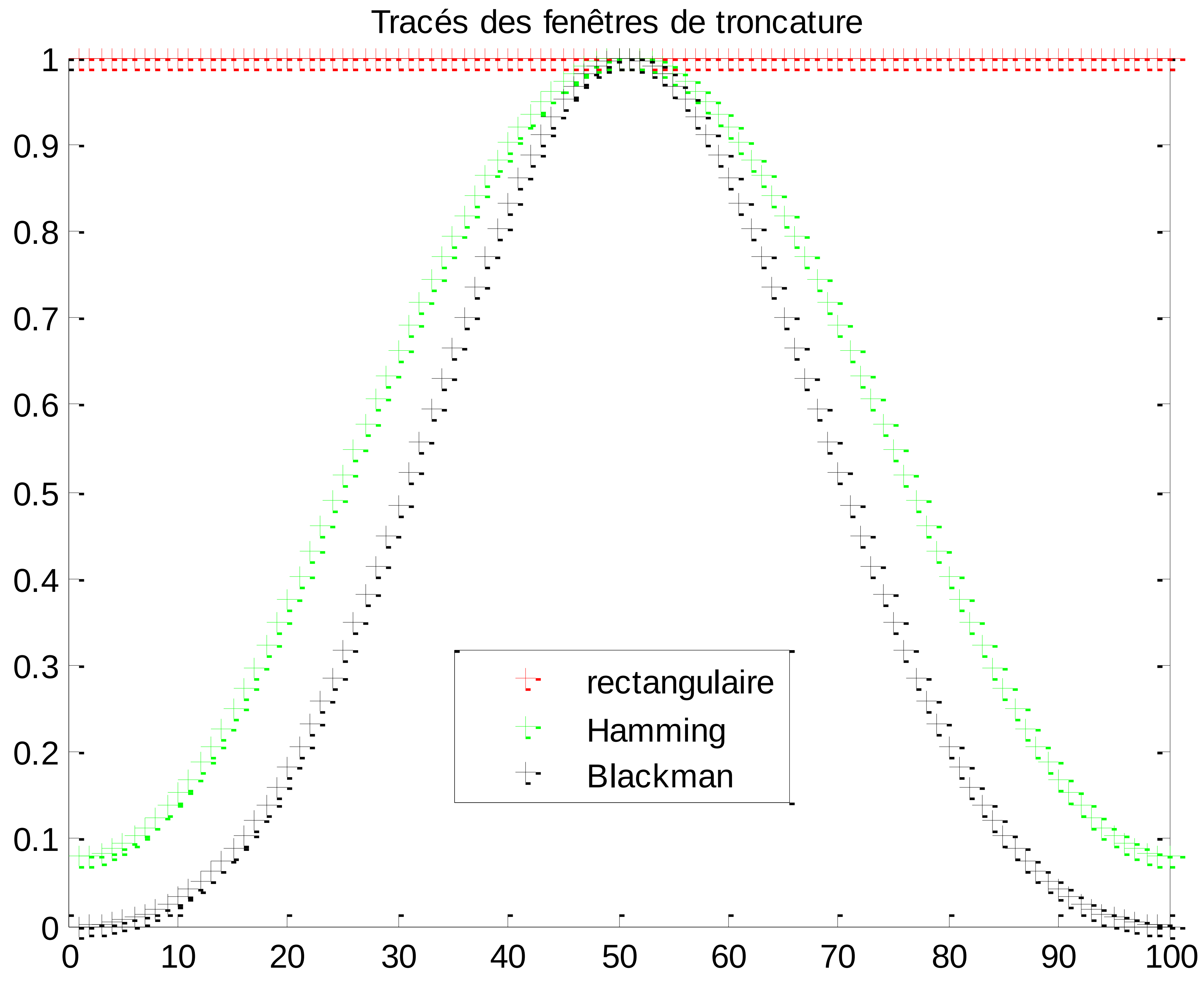


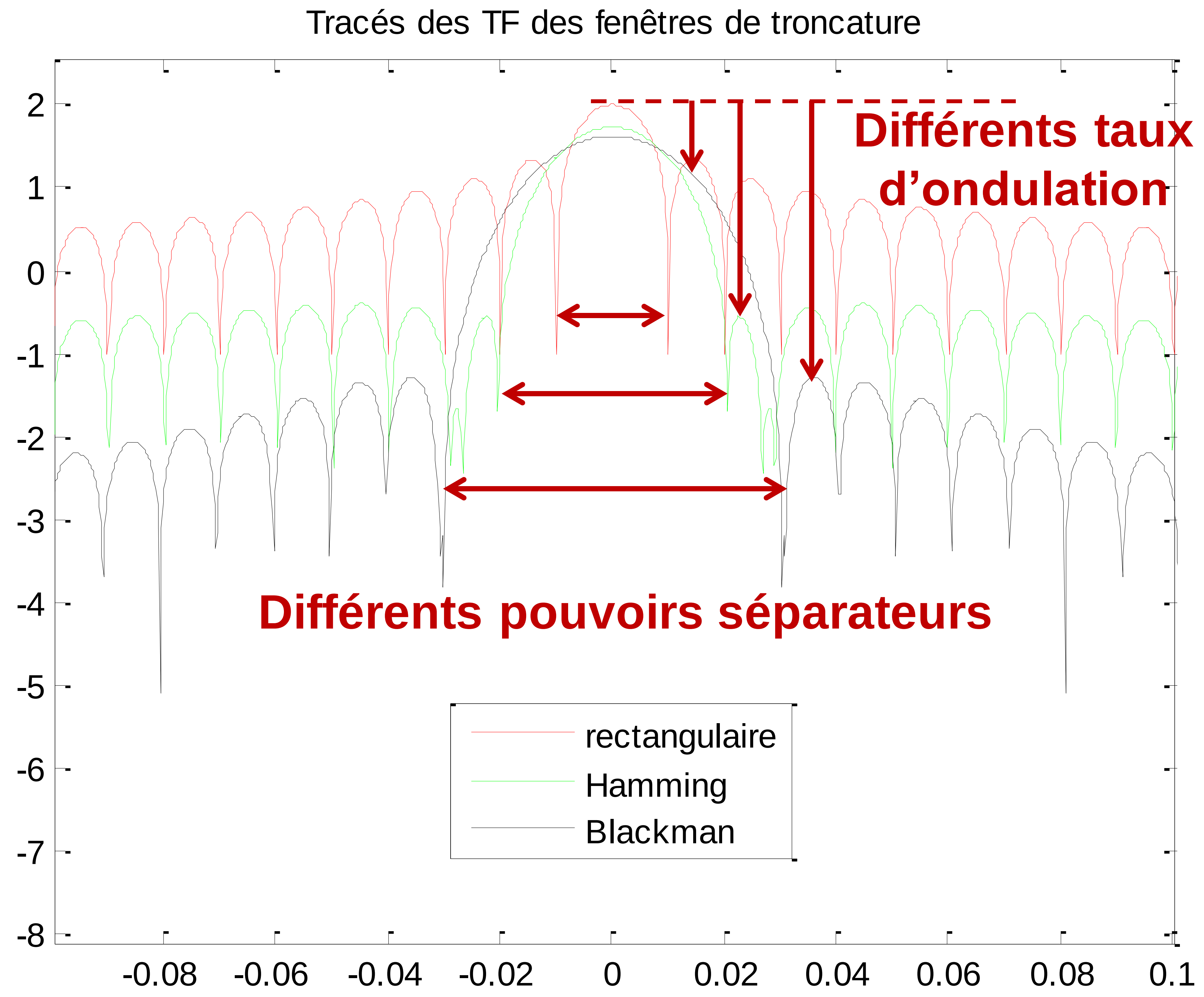
Transformée de Fourier Discrète

Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemples





Transformée de Fourier Discrète

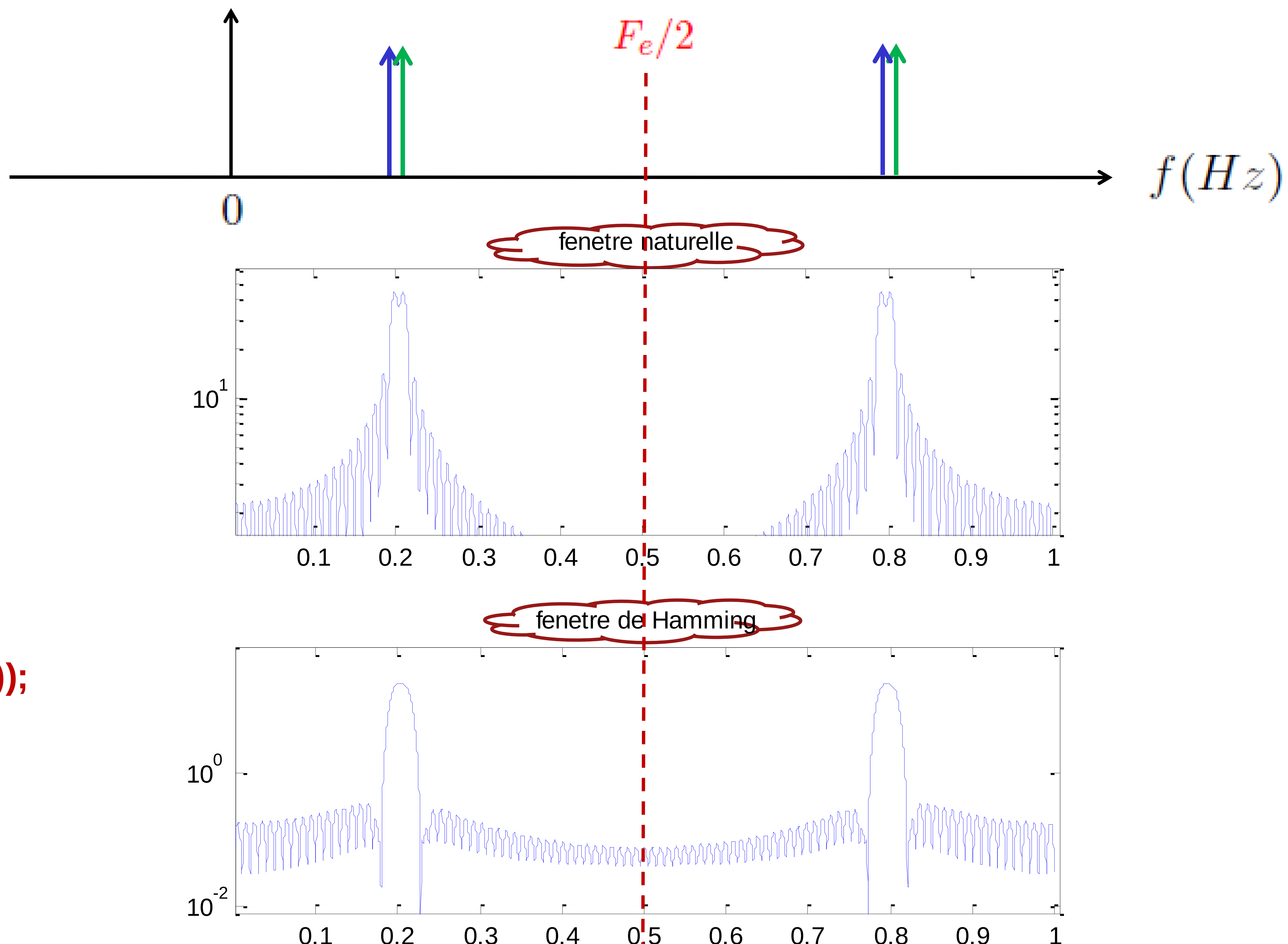
Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 1 : somme de deux cosinus proches en fréquences

```
%Exemple1
%Paramètres
f1=200;        %fréquence du cosinus 1
f2=207;        %fréquence du cosinus 2
Fe=1000;       %fréquence d'échantillonnage
Te=1/Fe;       %période d'échantillonnage
N=100;         %nombre d'échantillons

%Génération du signal
x1=cos(2*pi*f1*[0:Te:N*Te]);
x2=cos(2*pi*f2*[0:Te:N*Te]);
```

```
%fenêtre naturelle
x=x1+x2;
X_V1=fft(x,4096);
%fenêtre de hamming
w=window(@hamming,length(x));
x=(x1+x2).*w.';
X_V2=fft(x,4096);
%Tracés
figure
subplot(2,1,1)
plot(linspace(0,1,4096),log10(abs(X_V1)));
title('fenetre naturelle');
subplot(2,1,2)
plot(linspace(0,1,4096),log10(abs(X_V2)));
title('fenetre de Hamming');
```



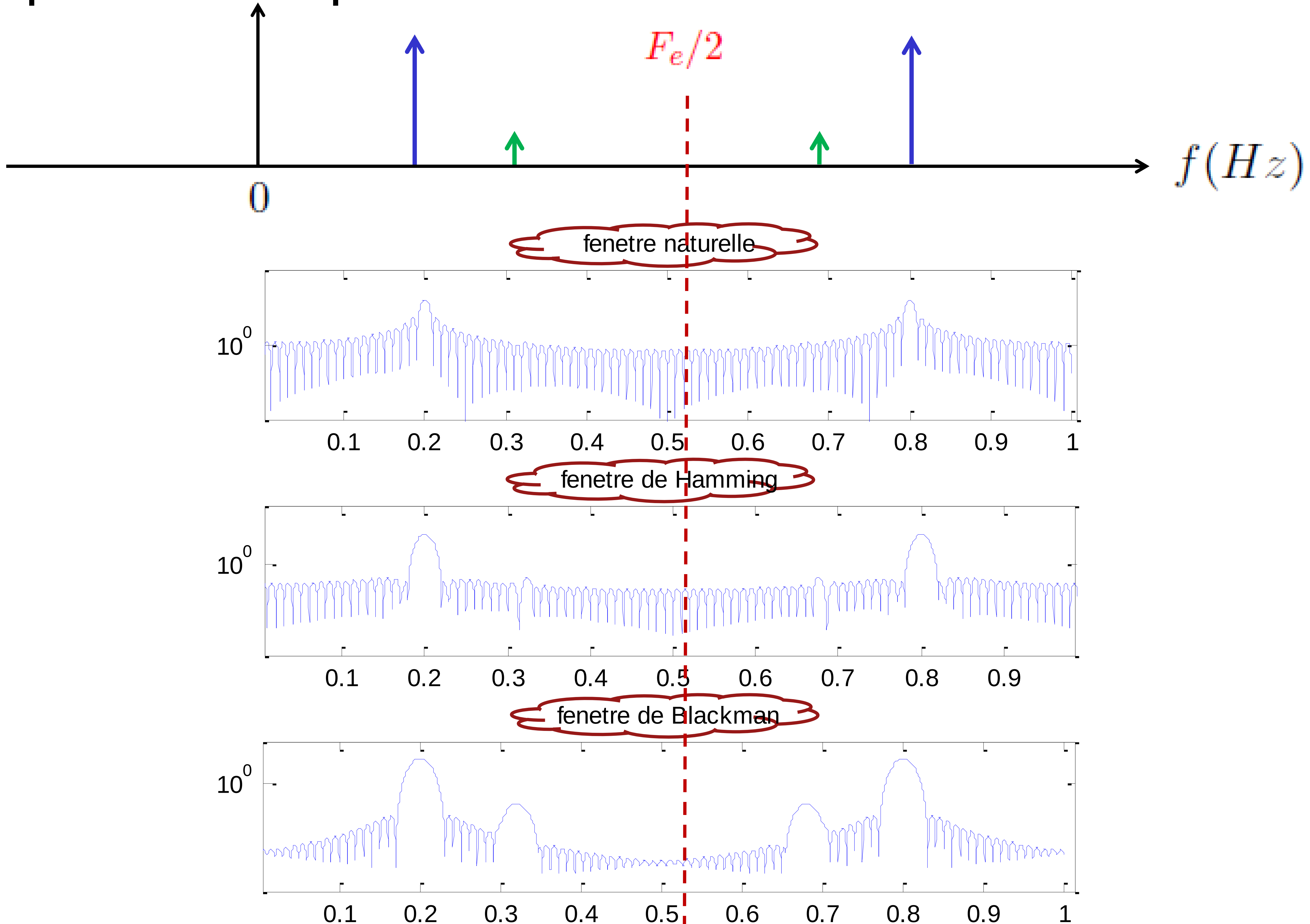
Transformée de Fourier Discrète

Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 2 : somme de deux cosinus proches de puissances différentes

```
%Exemple 2
%Paramètres
f1=200;     %fréquence du cosinus 1
f2=320;     %fréquence du cosinus 2
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;      %nombre d'échantillons

%Génération du signal
x1=cos(2*pi*f1*[0:Te:N*Te]);
x2=0.005*cos(2*pi*f2*[0:Te:N*Te]);
x=x1+x2;

%fenetre naturelle
X=fft(x,4096);
figure
subplot(3,1,1)
semilogy(linspace(0,1,4096),abs(X));
axis([0 1 10^-5 10^5])
title('fenetre naturelle');

%fenetre de hamming
w=window(@hamming,length(x));
x=(x1+x2).*w.';
X=fft(x,4096);
subplot(3,1,2)
semilogy(linspace(0,1,4096),abs(X));
title('fenetre de Hamming');

%fenetre de blackman
w=window(@blackman,length(x));
%w=blackman(N);
x=(x1+x2).*w.';
X=fft(x,4096);
subplot(3,1,3)
semilogy(linspace(0,1,4096),abs(X));
title('fenetre de Blackman');
```



Transformée de Fourier Discrète

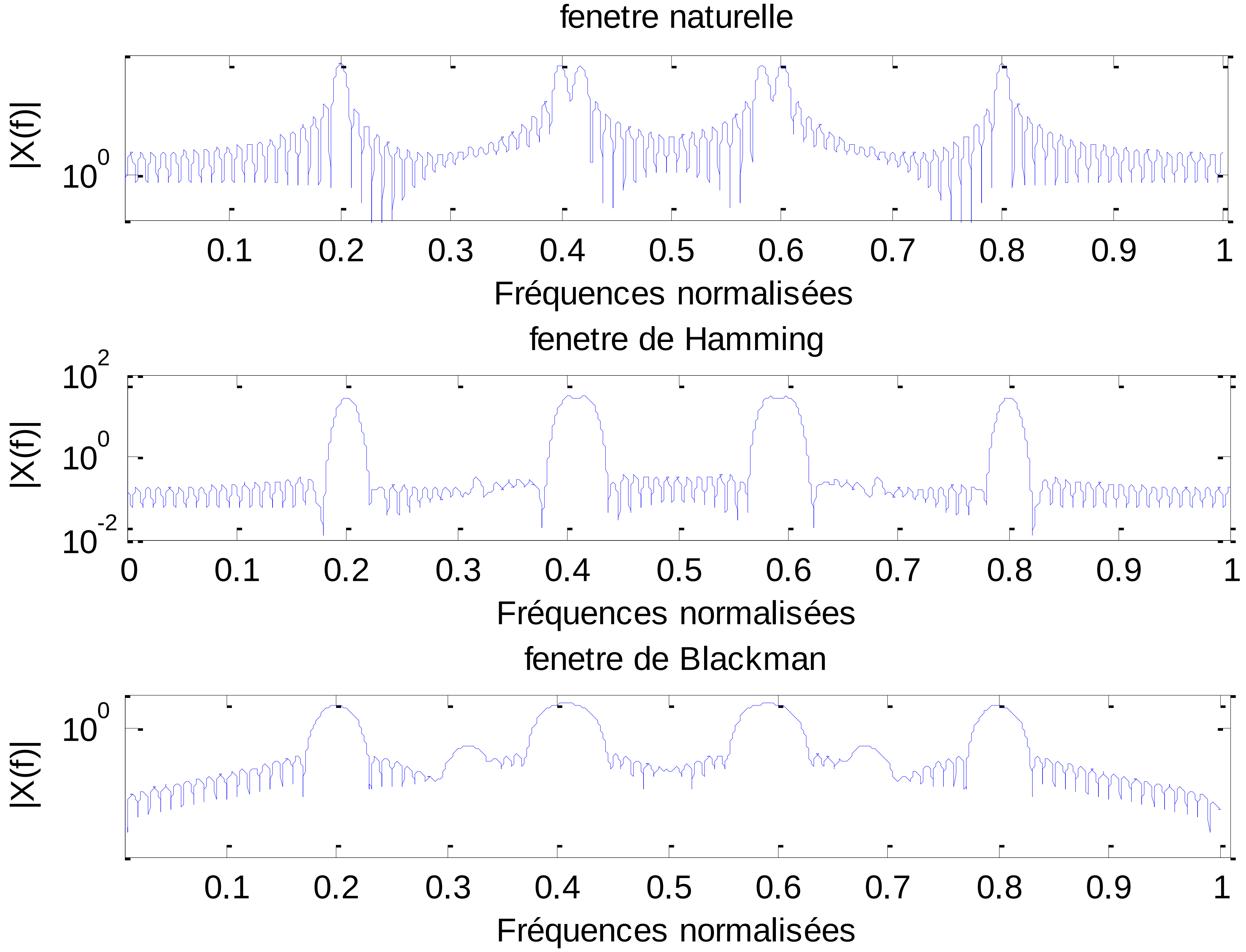
Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 3

<u>Ce signal est une somme de cosinus.</u>
<u>Il comprend :</u>

1- Deux cosinus
2- Trois cosinus
3- Quatre cosinus
4- Je ne sais pas répondre



Signal ???

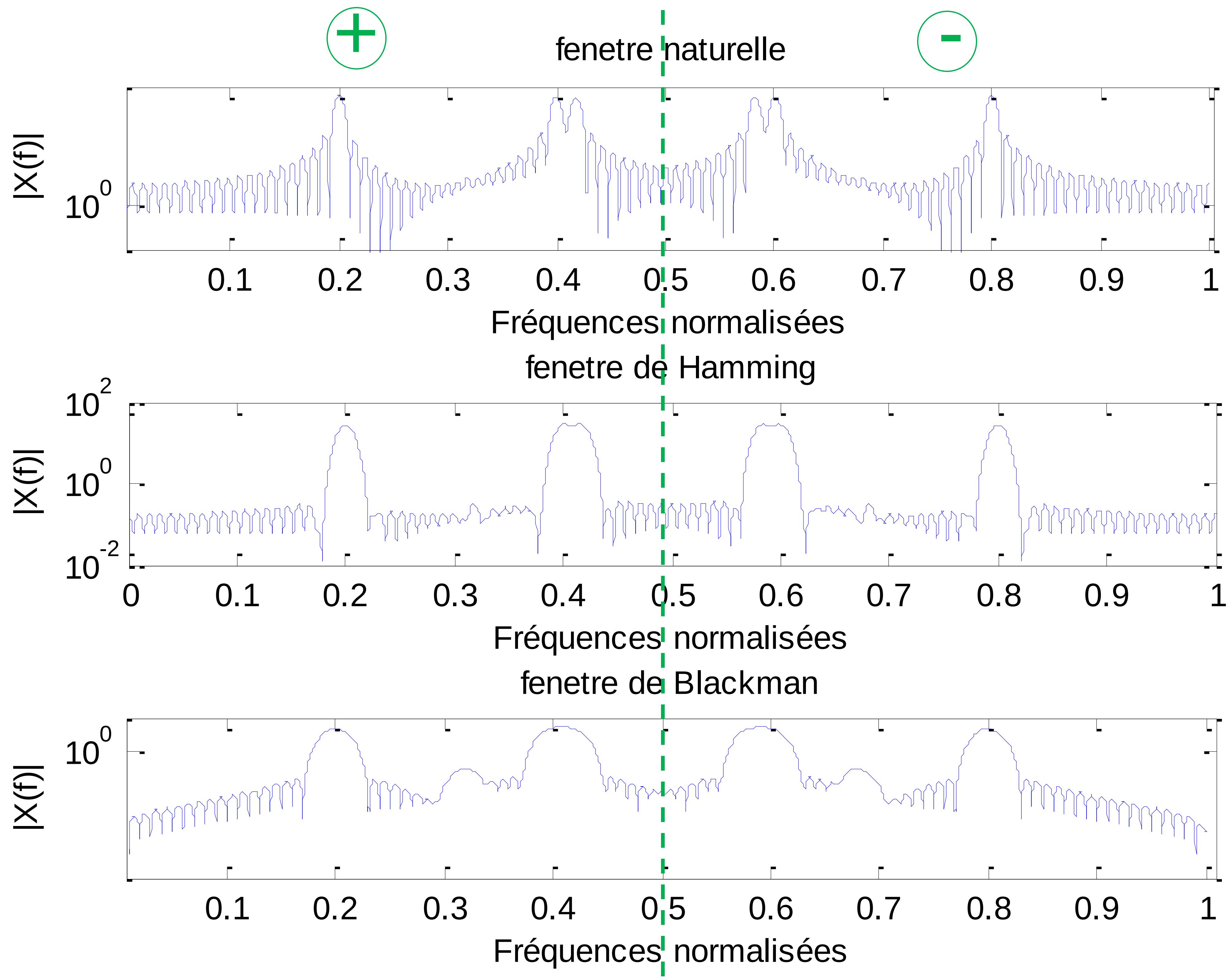
Transformée de Fourier Discrète

Signal de durée limitée

Utiliser d'autres fenêtres de troncature (fenêtre de pondération, d'apodisation) ?

Exemple 3

4 cosinus :
3 de puissances identiques
1 de plus faible puissance



3 cosinus
de puissances identiques

2 cosinus
de puissances identiques
Mais quelque chose de bizarre…

3 cosinus
de puissances différentes

Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Quels impacts ??

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

1- Echantillonnage temporel

$$x(t) \;\rightarrow\; \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

2- Signal de durée limitée

$$x(t) \rightarrow x_L(t) = \begin{cases} x(t) & \textit{pour } t \in [0, L] \\ 0 & \textit{ailleurs} \end{cases}$$

$\Rightarrow$ Périodisation de la TFD

$\Rightarrow$!! Respecter la condition de Shannon !!

$\Rightarrow$!! Lecture des tracés !!

$\Rightarrow$ Distorsion de la TFD attendue (analyse spectrale numérique avec pouvoir séparateur limité et apparition d'ondulations)

$\Rightarrow$ Utilisation de plusieurs fenêtres de pondération

Transformée de Fourier Discrète

Signal échantillonné et de durée limitée

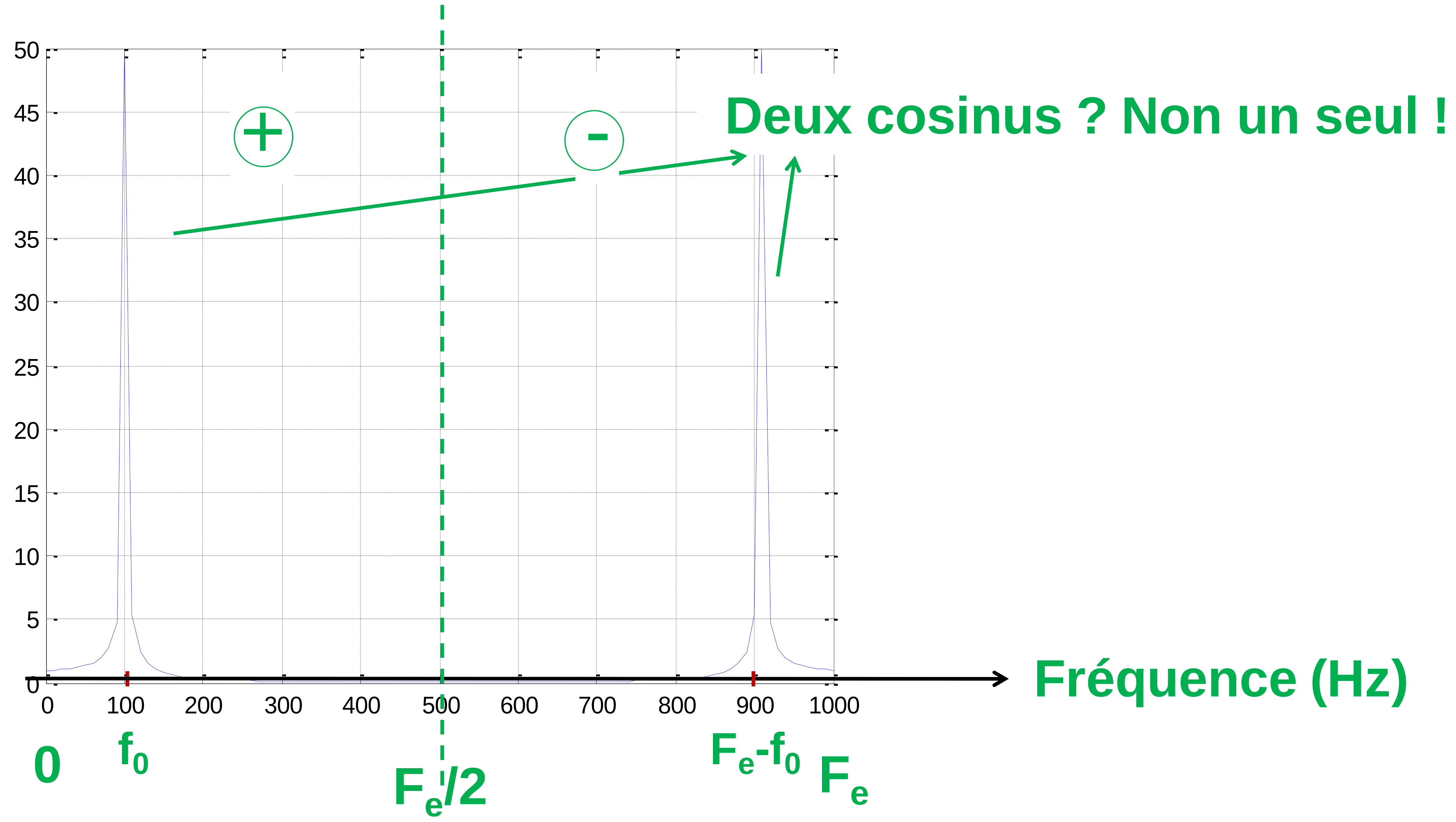
Exemple

$$x(t) = \cos\left(2\pi f_0 t\right)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=101;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure
plot(linspace(0,Fe,length(X)),abs(X))
```

Tracé du module de la TFD obtenu :



Transformée de Fourier Discrète

Signal échantillonné et de durée limitée

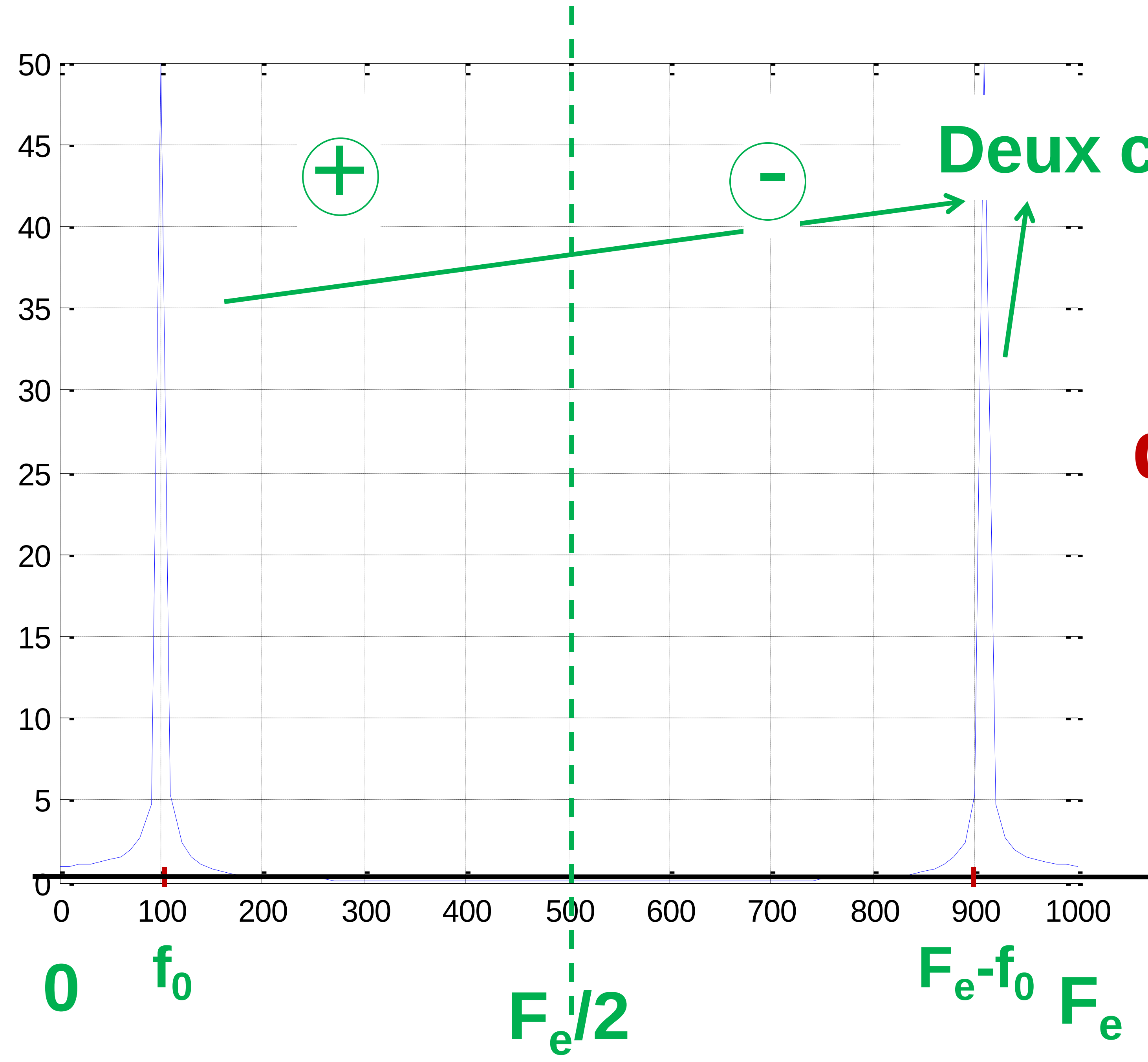
Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=101;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:(N-1)*Te]);

%Tracé du signal
figure; plot(x)

%Calcul de la TFD du signal
X=fft(x);

%Tracé du module de la TFD du signal
figure
plot(linspace(0,Fe,length(X)),abs(X))
```

Tracé du module de la TFD obtenu :



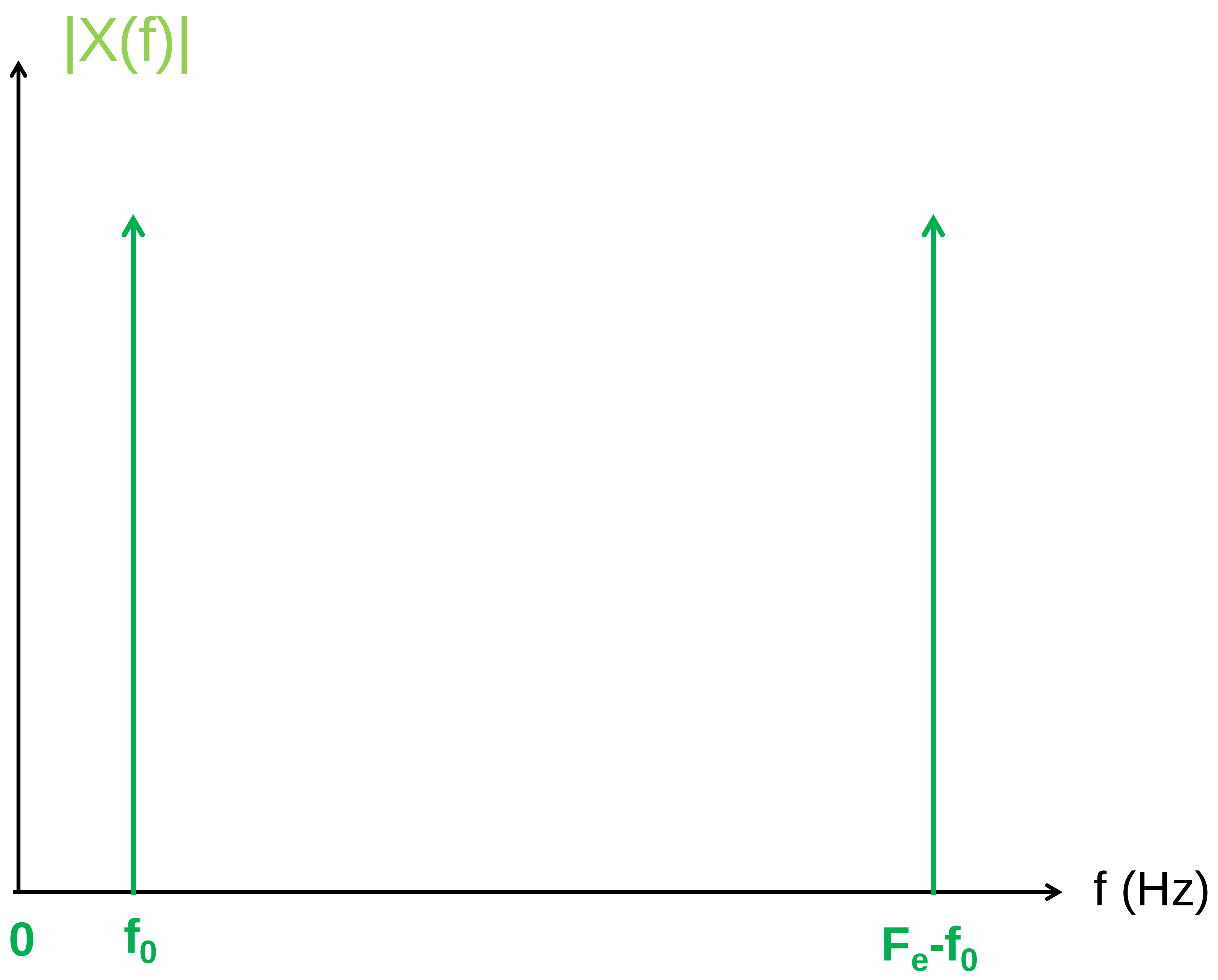
Mais où sont les noyaux de Dirichlet ???

Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

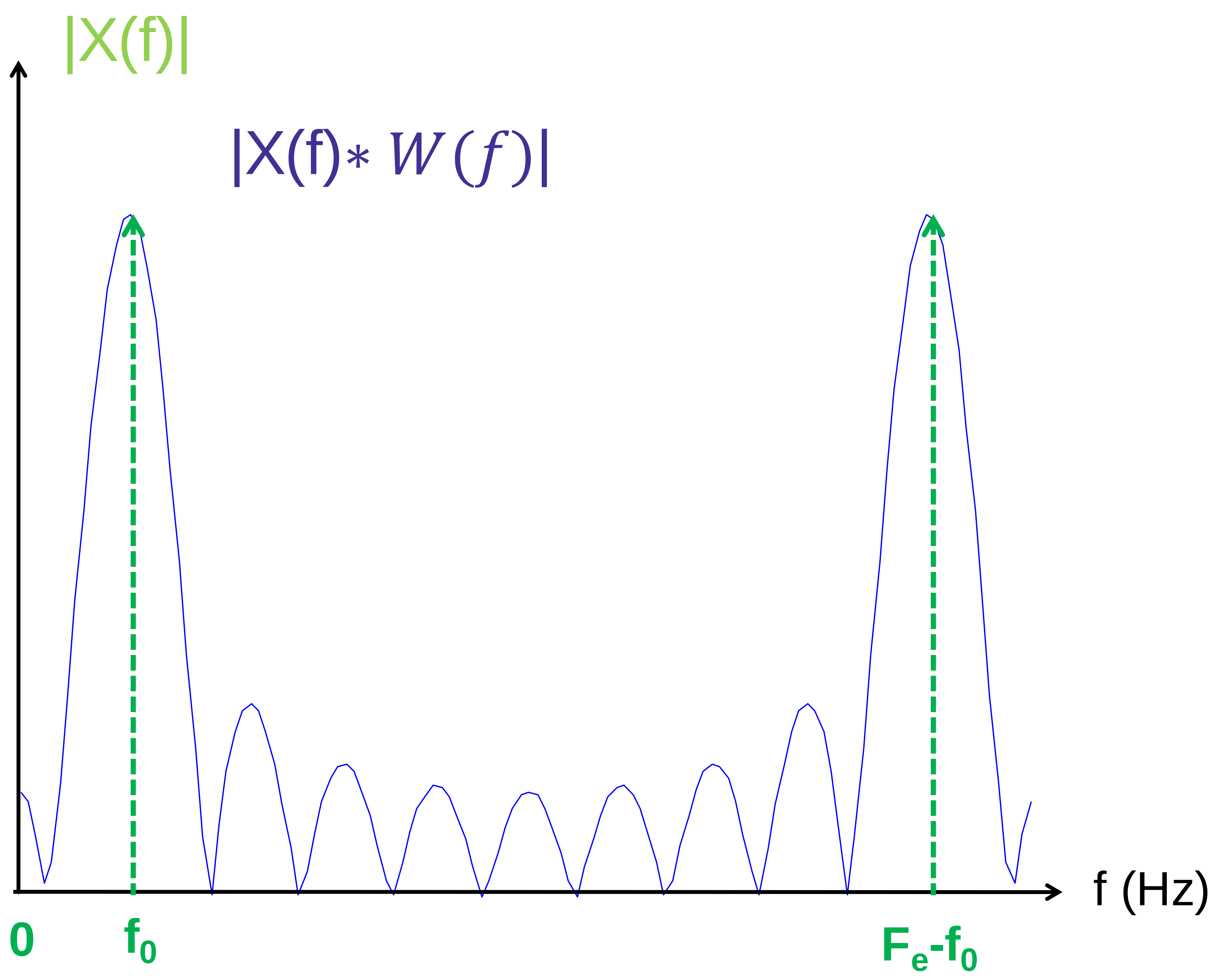


Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

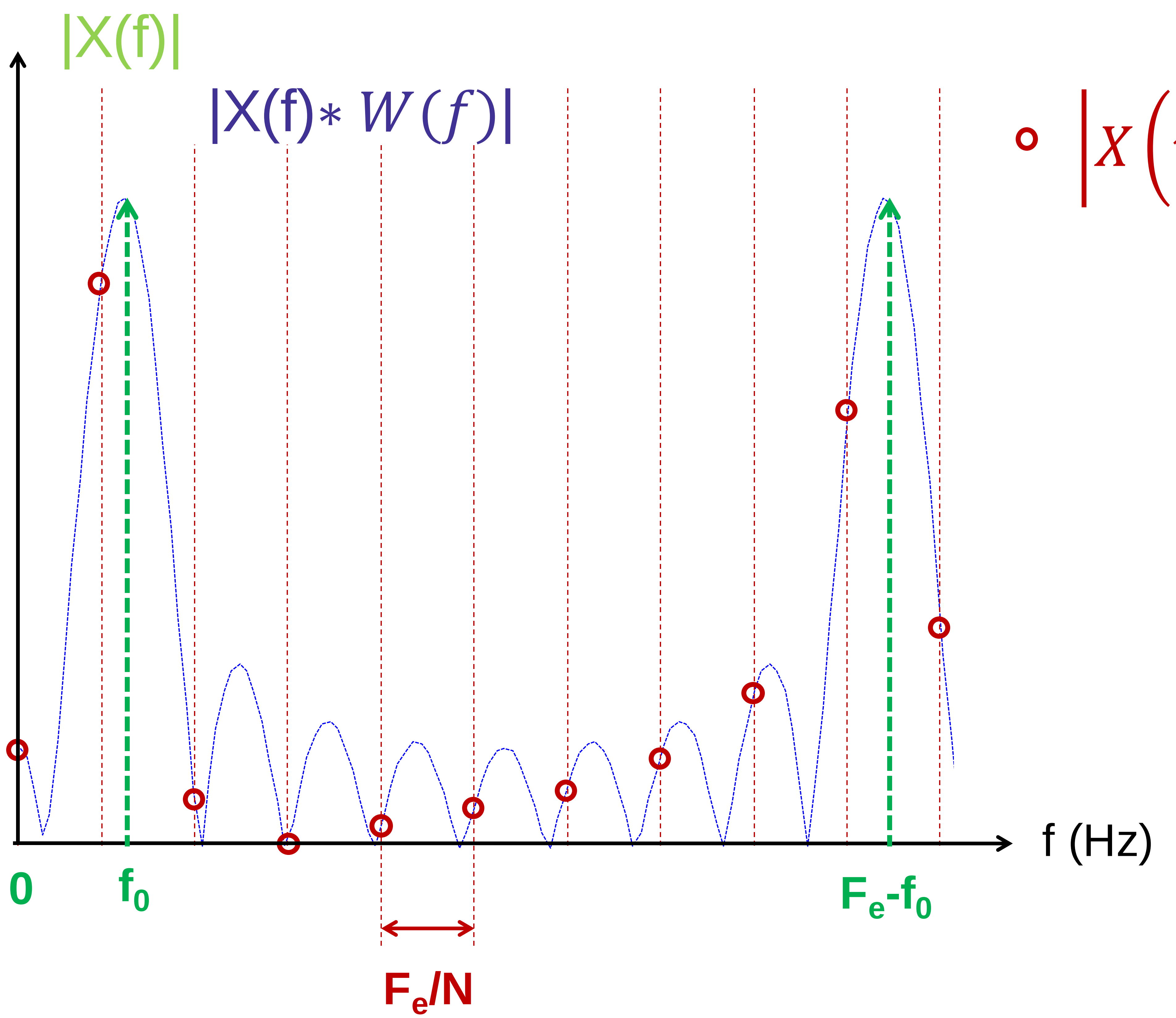


Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$



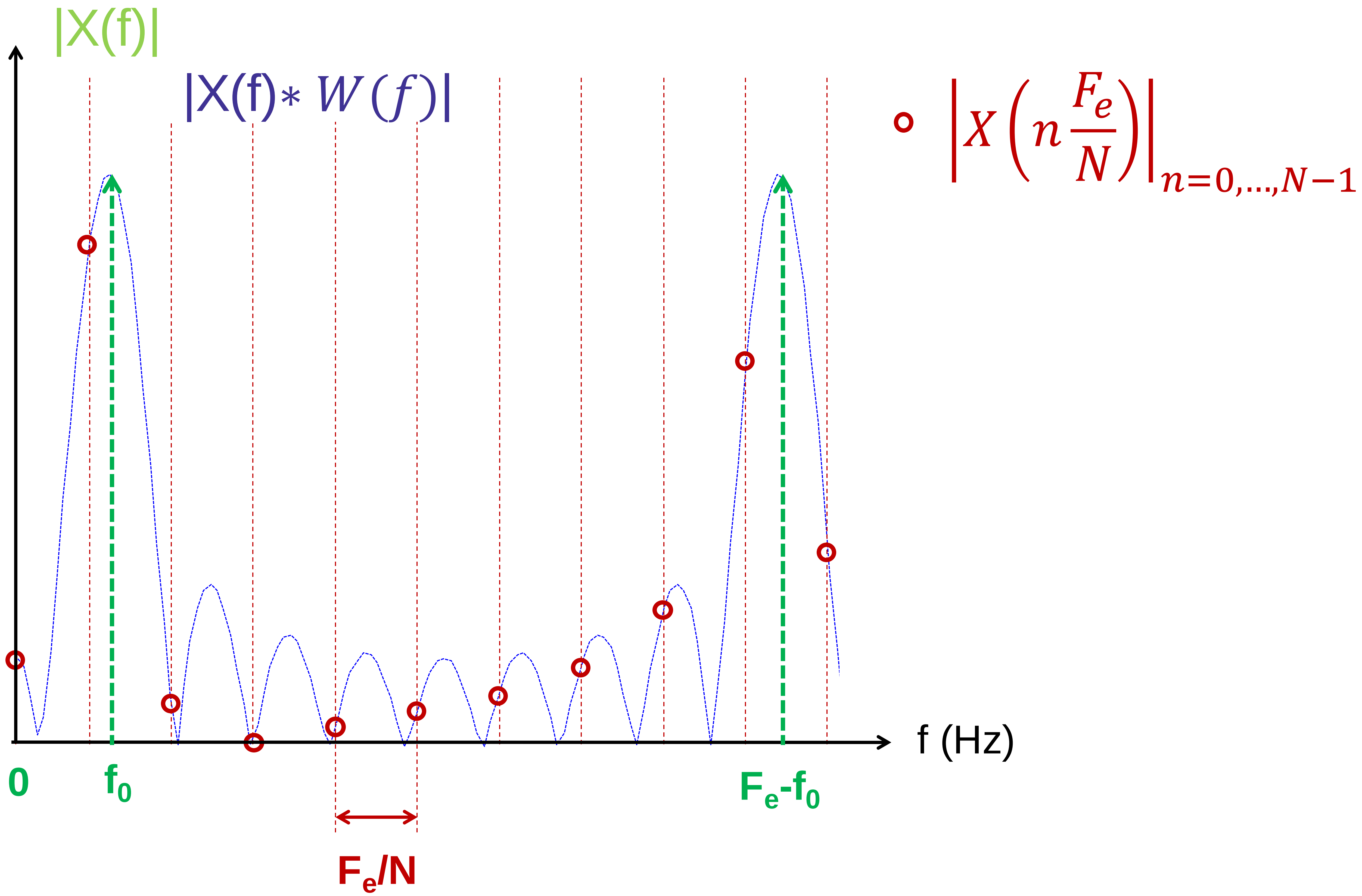
$$\left|X\left(n\frac{F_e}{N}\right)\right|_{n=0,\dots,N-1}$$

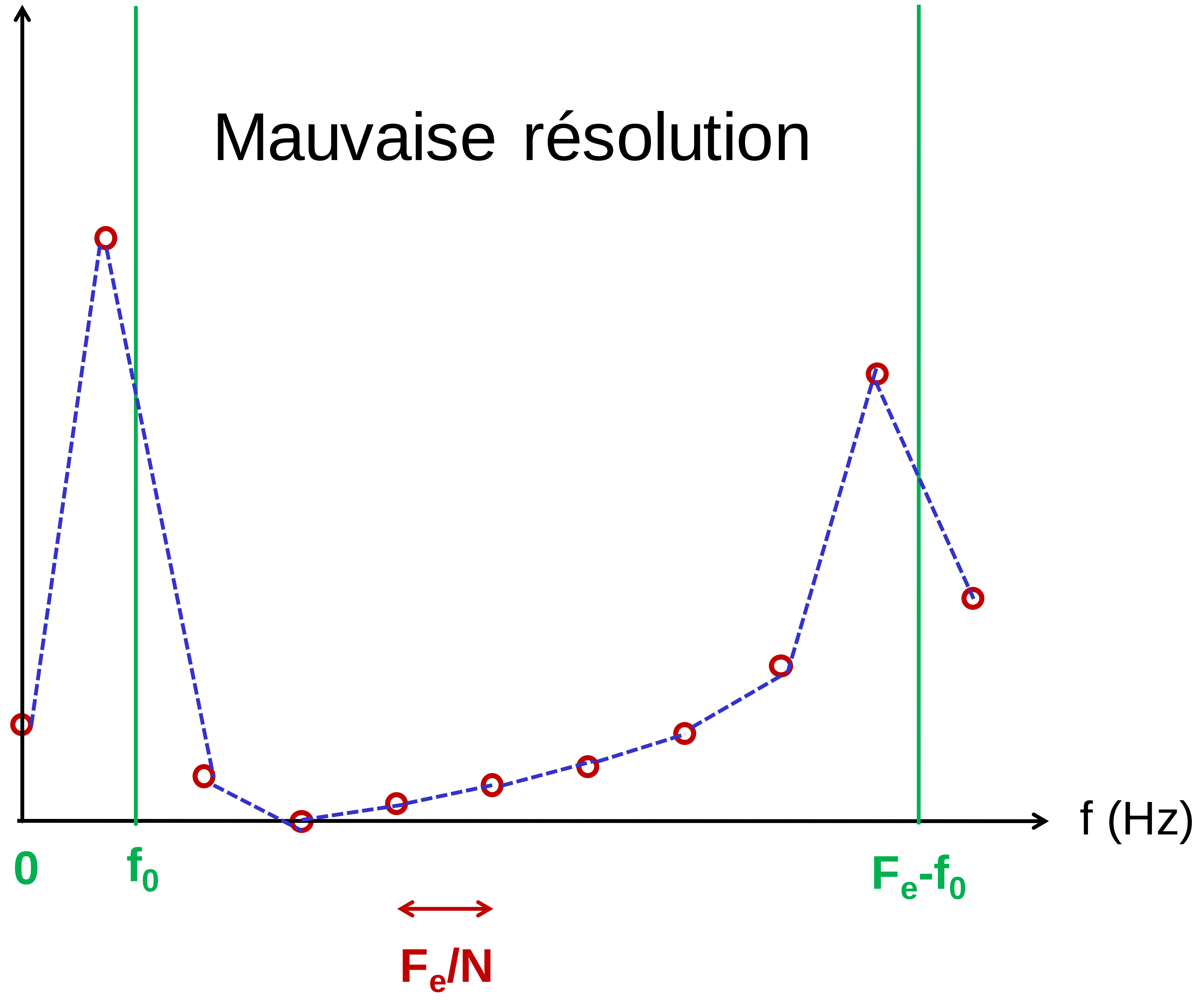
Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\ f_0 = 100\ Hz$$





Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Méthode du Zero padding :

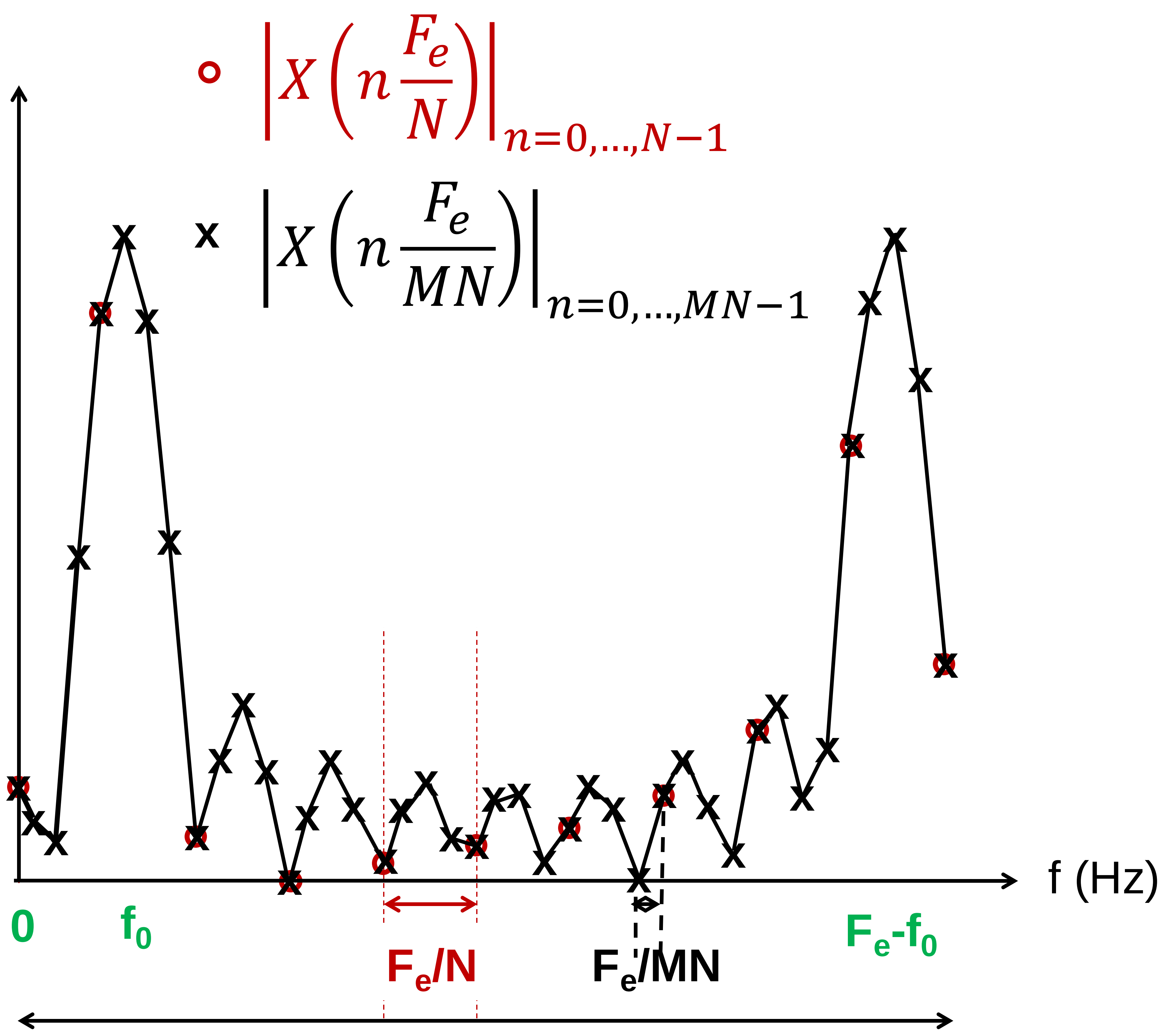
$$y(k) \begin{cases} = x(k) \;\; for \;\; k = 0, \ldots, N-1 \\ = 0 \;\; for \;\; k = N, \ldots, MN-1 \end{cases}$$

$$\Downarrow$$

$$Y(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{MN}}$$

$$for \;\; n = 0, \ldots, MN-1$$

=> Interpolation de la TFD



MN points espacés de F_e/MN entre 0 et F_e

Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

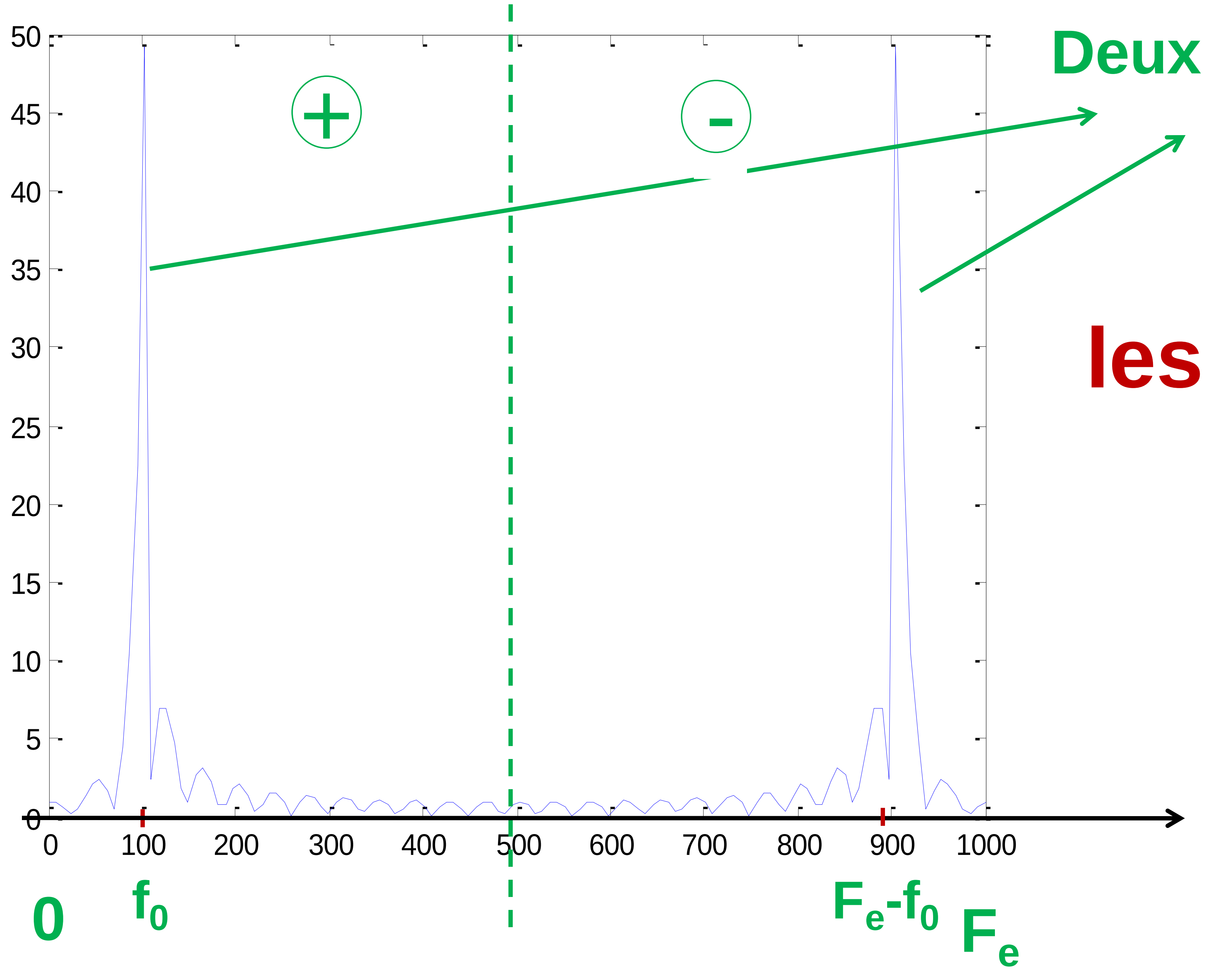
$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X=fft(x,128);
%Tracé du module de la TFD du signal
figure;
plot(linspace(0,Fe,length(X)),abs(X))
```

Utilisation de Zero Padding

Tracé du module de la TFD obtenu :



Voilà les noyaux de Dirichlet !

Transformée de Fourier Discrète

Echantillonnage de la TFD

Exemple

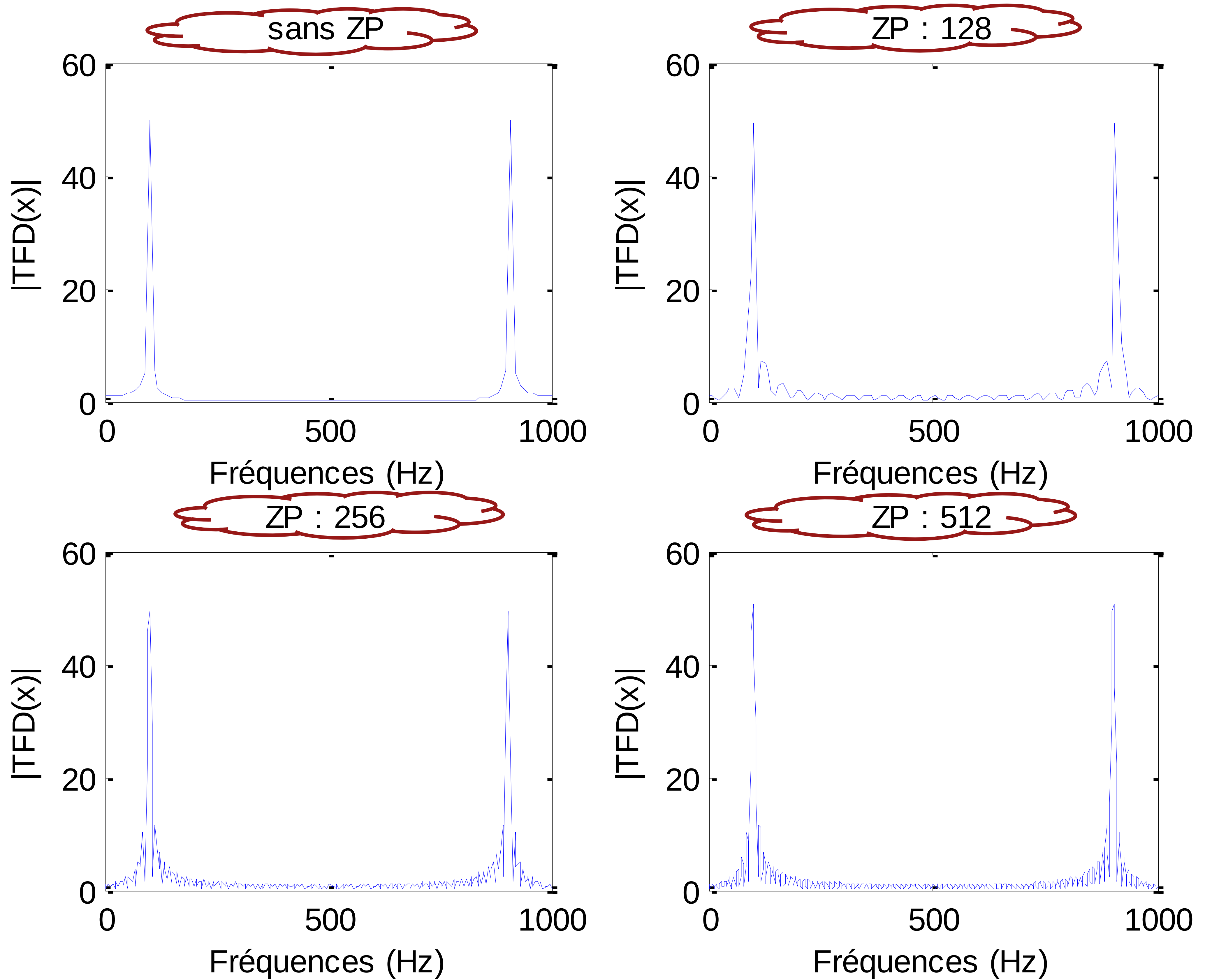
$$x(t) = \cos(2\pi f_0 t)\,;\;\; f_0 = 100\ Hz$$

Simulation sous Matlab :

```
%Paramètres
f0=100;     %fréquence du cosinus
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;    %période d'échantillonnage
N=100;      %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]);

%Tracé du signal
figure; plot([0:Te:N*Te],x)

%Calcul de la TFD du signal
X1=fft(x);
X2=fft(x,128);
X3=fft(x,256);
X4=fft(x,512);
```

```
% Tracé du module de la TFD du signal
figure;
subplot(2,2,1)
plot(linspace(0,Fe,length(X1)),abs(X1))
xlabel('Fréquences (Hz)')
subplot(2,2,2)
plot(linspace(0,Fe,length(X2)),abs(X2))
xlabel('Fréquences (Hz)')
subplot(2,2,3)
plot(linspace(0,Fe,length(X3)),abs(X3))
xlabel('Fréquences (Hz)')
subplot(2,2,4)
plot(linspace(0,Fe,length(X4)),abs(X4))
xlabel('Fréquences (Hz)')
```

Tracé du module de la TFD du signal pour différentes valeurs de Zero Padding :



Transformée de Fourier Discrète

Transformée de Fourier (TF)

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

Quels impacts ??

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, ..., N-1$$

1- Echantillonnage temporel

$$x(t) \rightarrow \{x(kT_e)\}_{k=-\infty,...,+\infty}$$

$\Rightarrow$ Périodisation de la TFD

$\Rightarrow$!! Respecter la condition de Shannon !!

$\Rightarrow$!! Lecture des tracés !!

2- Signal de durée limitée

$$x(t) \rightarrow x_L(t) = \begin{cases} x(t) & \textit{pour } t \in [0, L] \\ 0 & \textit{ailleurs} \end{cases}$$

$\Rightarrow$ Distorsion de la TFD attendue (analyse spectrale numérique avec pouvoir séparateur limité et apparition d'ondulations)

$\Rightarrow$ Utilisation de plusieurs fenêtres de pondération

3- Echantillonnage fréquentiel

$$X(f) \rightarrow \{X(n\Delta f)\}_{n=0,...,N-1}$$

$\Rightarrow$ Mauvaise visualisation de la TFD obtenue (résolution insuffisante)

$\Rightarrow$ Interpolation fréquentielle par Zero Padding

Transformée de Fourier Discrète

Transformée de Fourier Discrète (TFD)

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \; n = 0, \ldots, N-1$$

Transformée de Fourier Discrète Inverse (TFD^{-1})

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) e^{+j2\pi \frac{kn}{N}}, \; k = 0, \ldots, N-1$$

Transformée de Fourier Discrète

Propriétés

→ Linéarité $TFD\left[x_1(k) + \lambda x_2(k)\right] = TFD\left[x_1(k)\right] + \lambda TFD\left[x_2(k)\right]$

→ Translation => rotation de phase $TFD\left[x(k-k_0)\right] = X(n)e^{-j2\pi\frac{k_0 n}{N}}$

→ Symétrie hermitienne $X(N-n) = X(-n) = X^*(n).$

→ Convolution circulaire $X_1(n)X_2(n) \xrightarrow{TFD^{-1}} x_1(k) \otimes x_2(k) = \sum_{p=0}^{N-1} x_1(p)x_2\left([k-p]_{modulo\ N}\right)$

→ Egalité de Parseval $\sum_{k=0}^{N-1} |x(k)|^2 = \frac{1}{N}\sum_{n=0}^{N-1} |X(n)|^2$

→ Algorithme de calcul rapide (Fast Fourier Transform Algorithm : FFT) : $N\log_2(N)$ MAC << N^2 pour N points

Transformée de Fourier Discrète

Propriétés

→ Linéarité $TFD\left[x_1(k)+\lambda x_2(k)\right]=TFD\left[x_1(k)\right]+\lambda TFD\left[x_2(k)\right]$

→ Translation => rotation de phase $TFD\left[x(k-k_0)\right]=X(n)e^{-j2\pi\frac{k_0 n}{N}}$

→ Symétrie hermitienne $X(N-n)=X(-n)=X^*(n).$

!! La TFD et la TFD^{-1} transforment un produit en produit de convolution circulaire !!

→ Convolution circulaire $X_1(n)X_2(n)\xrightarrow{TFD^{-1}}x_1(k)\otimes x_2(k)=\sum_{p=0}^{N-1}x_1(p)x_2\left([k-p]_{modulo\ N}\right)$

→ Egalité de Parseval $\sum_{k=0}^{N-1}|x(k)|^2=\frac{1}{N}\sum_{n=0}^{N-1}|X(n)|^2$

→ Algorithme de calcul rapide (Fast Fourier Transform Algorithm : FFT) : $N\log_2(N)$ MAC << N^2 pour N points

Transformée de Fourier Discrète

Convolution circulaire

Convolution linéaire (« classique ») :

$$(x_1 * x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2(k-p)$$

Convolution circulaire :

$$(x_1 \otimes x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2\left((k-p)_{modulo\ N}\right)$$

Exemple

$x_1(p)$: . . . 0 0 0 1 2 3 0 0 0 . . .
$x_2(p)$: . . . 0 0 0 1 1 1 0 0 0 . . .

k=-3 . . 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0
k=-2 . . . 0 0 0 1 1 **1** 0 0 0 . . . $\longrightarrow$ 1
k=-1 . . . 0 0 0 1 **1 1** 0 0 0 . . . $\longrightarrow$ 3
k=0 . . . 0 0 0 **1 1 1** 0 0 0 . . . $\longrightarrow$ 6
k=1 . . . 0 0 0 **1 1** 1 0 0 0 . . . $\longrightarrow$ 5
k=2 . . . 0 0 0 **1** 1 1 0 0 0 . . . $\longrightarrow$ 3
k=3 . . . 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0

$x_1(p_{modulo\ N})$: . . . 1 2 3 1 2 3 1 2 3 . . .
$x_2(p_{modulo\ N})$: . . . 1 1 1 1 1 1 1 1 1 . . .

k=-3 . . 1 1 1 1 1 1 **1 1 1** . . . $\longrightarrow$ 6
k=-2 . . . 1 1 1 1 1 **1 1 1** 1 . . . $\longrightarrow$ 6
k=-1 . . . 1 1 1 1 **1 1 1** 1 1 . . . $\longrightarrow$ 6
k=0 . . . 1 1 1 **1 1 1** 1 1 1 . . . $\longrightarrow$ 6
k=1 . . . 1 1 **1 1 1** 1 1 1 1 . . . $\longrightarrow$ 6
k=2 . . . 1 **1 1 1** 1 1 1 1 1 . . . $\longrightarrow$ 6
k=3 . . . **1 1 1** 1 1 1 1 1 1 . . . $\longrightarrow$ 6

Transformée de Fourier Discrète

Convolution circulaire

Convolution lineaire (« classique ») :

$$(x_1 * x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2(k-p)$$

Convolution circulaire :

$$(x_1 \otimes x_2)(k) = \sum_{p=-\infty}^{+\infty} x_1(p)x_2\left((k-p)_{modulo\ N}\right)$$

En prolongeant les signaux par N zéros

. . . 0 0 0 1 2 3 0 0 0 . . .
. . . 0 0 0 1 1 1 0 0 0 . . .

. . . 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0
. . . 0 0 0 1 1 **1** 0 0 0 . . . $\longrightarrow$ 1
. . . 0 0 0 1 **1 1** 0 0 0 . . . $\longrightarrow$ 3
. . . 0 0 0 **1 1 1** 0 0 0 . . . $\longrightarrow$ 6
. . . 0 0 0 **1 1** 1 0 0 0 . . . $\longrightarrow$ 5
. . . 0 0 0 **1** 1 1 0 0 0 . . . $\longrightarrow$ 3
. . . 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0

. . . 1 2 3 0 0 0 1 2 3 0 0 0 1 2 3 0 0 0. . .
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . .

. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 1
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 3
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 6
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 5
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 3
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0
. . . 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 . . . $\longrightarrow$ 0

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $\boldsymbol{N = 2^p} \Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi \frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k) \times W_N^{-kn} \quad n = 0, ..., N-1$$

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $\boldsymbol{N = 2^p} \Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi \frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k) \times W_N^{-kn} \quad n = 0, \ldots, N-1$$

Première décomposition $\Rightarrow N + 2(N/2)^2 = N + N^2/2 < N^2$ opérations (+/x)

$$X(n) = X_1(n) + W_N^{-n} \times X_2(n), \quad n = 0, \ldots, N-1$$

N opérations (+/x)

Indices pairs (TFD d'ordre N/2)

$$X_1(n) = \sum_{i=0}^{N/2-1} x(2i) W_{N/2}^{-in}, \quad i = 0, \ldots, N/2-1$$

$$X_2(n) = \sum_{i=0}^{N/2-1} x(2i+1) W_{N/2}^{-in}, \quad i = 0, \ldots, N/2-1$$

Indices impairs TFD d'ordre N/2

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $N = 2^p$ $\Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k)\times e^{-j2\pi\frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k)\times W_N^{-kn} \quad n = 0, \dots, N-1$$

Première décomposition $\Rightarrow N + 2(N/2)^2 = N + N^2/2 < N^2$ opérations (+/x)

$$X(n) = X_1(n) + W_N^{-n}\times X_2(n), \quad n = 0, \dots, N-1$$

N opérations (+/x)

Indices pairs (TFD d'ordre N/2) $X_1(n) = \sum_{i=0}^{N/2-1} x(2i)W_{N/2}^{-in}, \ i = 0, \dots, N/2-1$

$X_2(n) = \sum_{i=0}^{N/2-1} x(2i+1)W_{N/2}^{-in}, \ i = 0, \dots, N/2-1$ Indices impairs TFD d'ordre N/2

Deuxième décomposition $\Rightarrow 2(N/2) + 4(N/4)^2 = N + N^2/4 < N^2$ opérations (+/x)

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n}X_{12}(n) \qquad X_2(n) = X_{21}(n) + W_{N/2}^{-n}X_{22}(n)$$

Indices pairs — Indices impairs — Indices pairs — Indices impairs

4 TFDs d'ordre N/4

Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

TFD d'ordre $N = 2^p \Rightarrow N^2$ opérations (+/x)

$$X(n) = \sum_{k=0}^{N-1} x(k) \times e^{-j2\pi\frac{kn}{MN}} = \sum_{k=0}^{N-1} x(k) \times W_N^{-kn} \quad n = 0, \ldots, N-1$$

Première décomposition $\Rightarrow N + 2(N/2)^2 = N + N^2/2 < N^2$ opérations (+/x)

$$X(n) = X_1(n) + W_N^{-n} \times X_2(n), \quad n = 0, \ldots, N-1$$

N opérations (+/x)

Indices pairs (TFD d'ordre N/2) $X_1(n) = \sum_{i=0}^{N/2-1} x(2i) W_{N/2}^{-in}, \; i = 0, \ldots, N/2-1$

$X_2(n) = \sum_{i=0}^{N/2-1} x(2i+1) W_{N/2}^{-in}, \; i = 0, \ldots, N/2-1$ Indices impairs TFD d'ordre N/2

Deuxième décomposition $\Rightarrow 2(N/2) + 4(N/4)^2 = N + N^2/4 < N^2$ opérations (+/x)

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n) \qquad X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n)$$

Indices pairs — Indices impairs — Indices pairs — Indices impairs

4 TFDs d'ordre N/4

⋮

$$p\frac{N}{2}\ TFD\ d'ordre\ 2 \Rightarrow p \times \frac{N}{2} \times 2 = N\log_2(N)\ opérations\ (+/\times) \ll N^2$$

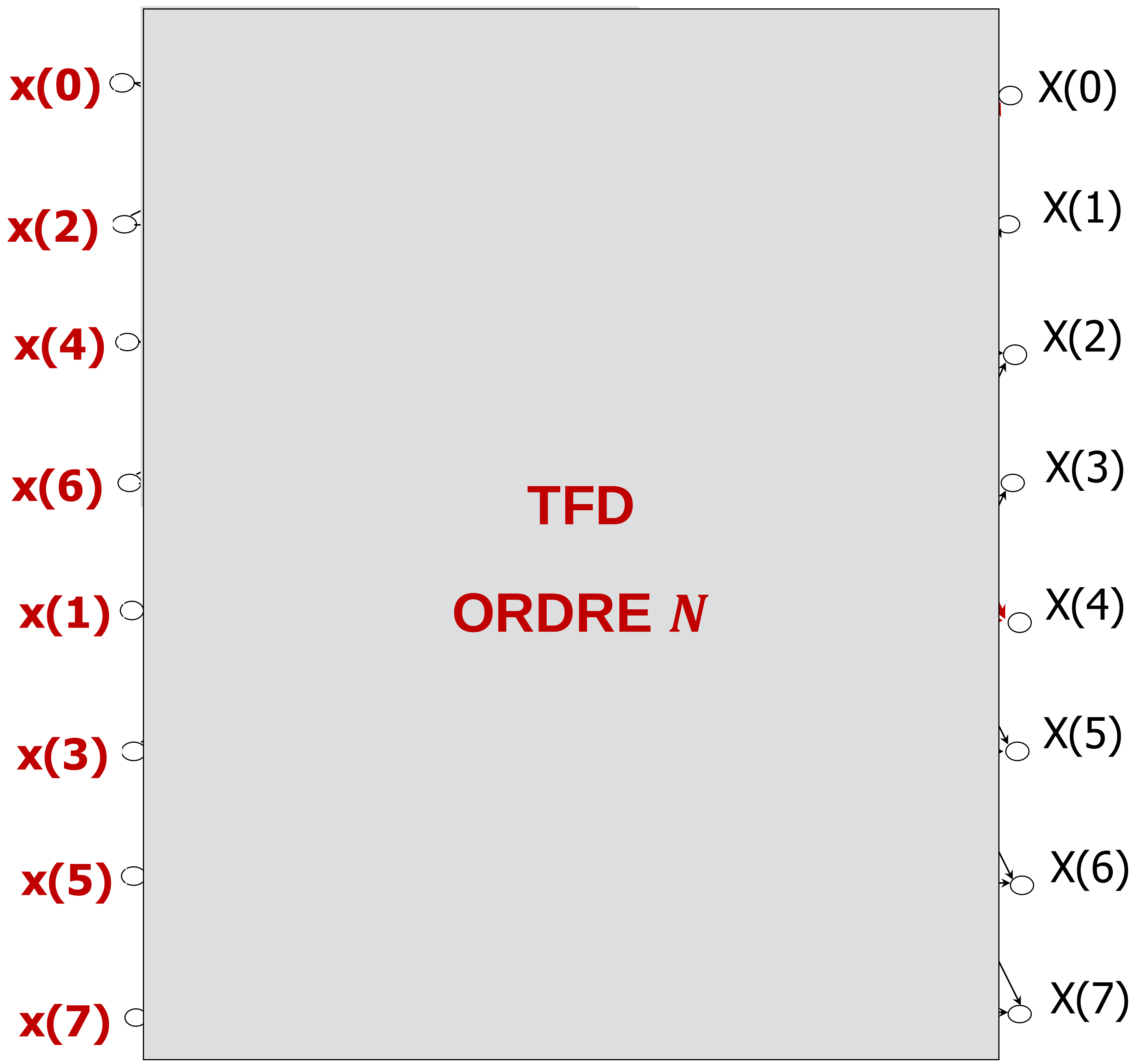
Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \ n = 0, \ldots, N-1$$

$N^2 = 64\ opérations\ d'addition/multiplication\ (+/\times)$



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = X_1(n) + W_N^{-n} X_2(n), \; n = 0, \dots, 7$$

Etape 1

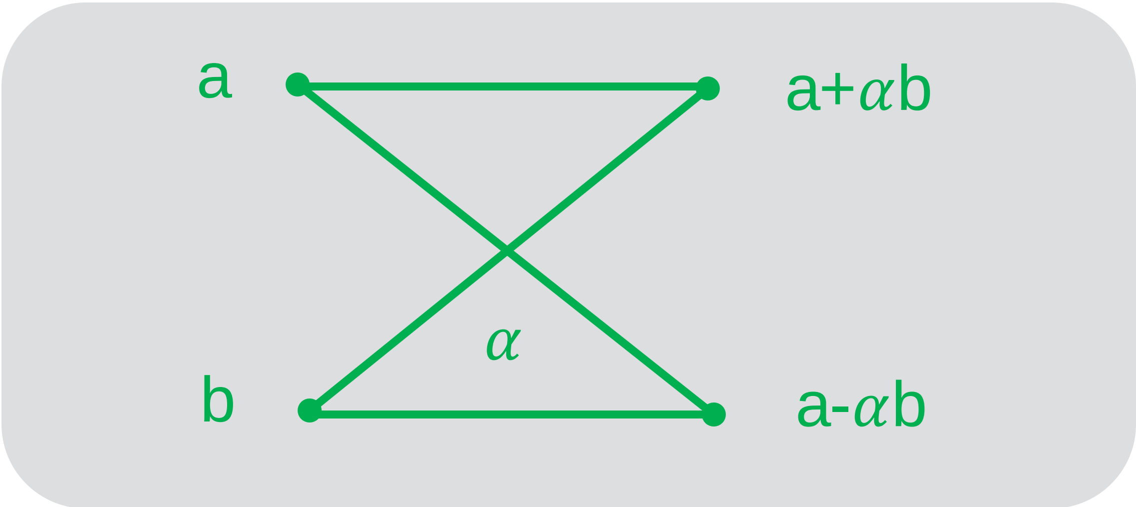
$sur\ x(0), x(2), x(4), x(6)$

Indices pairs
(TFD d'ordre N/2 = 4)

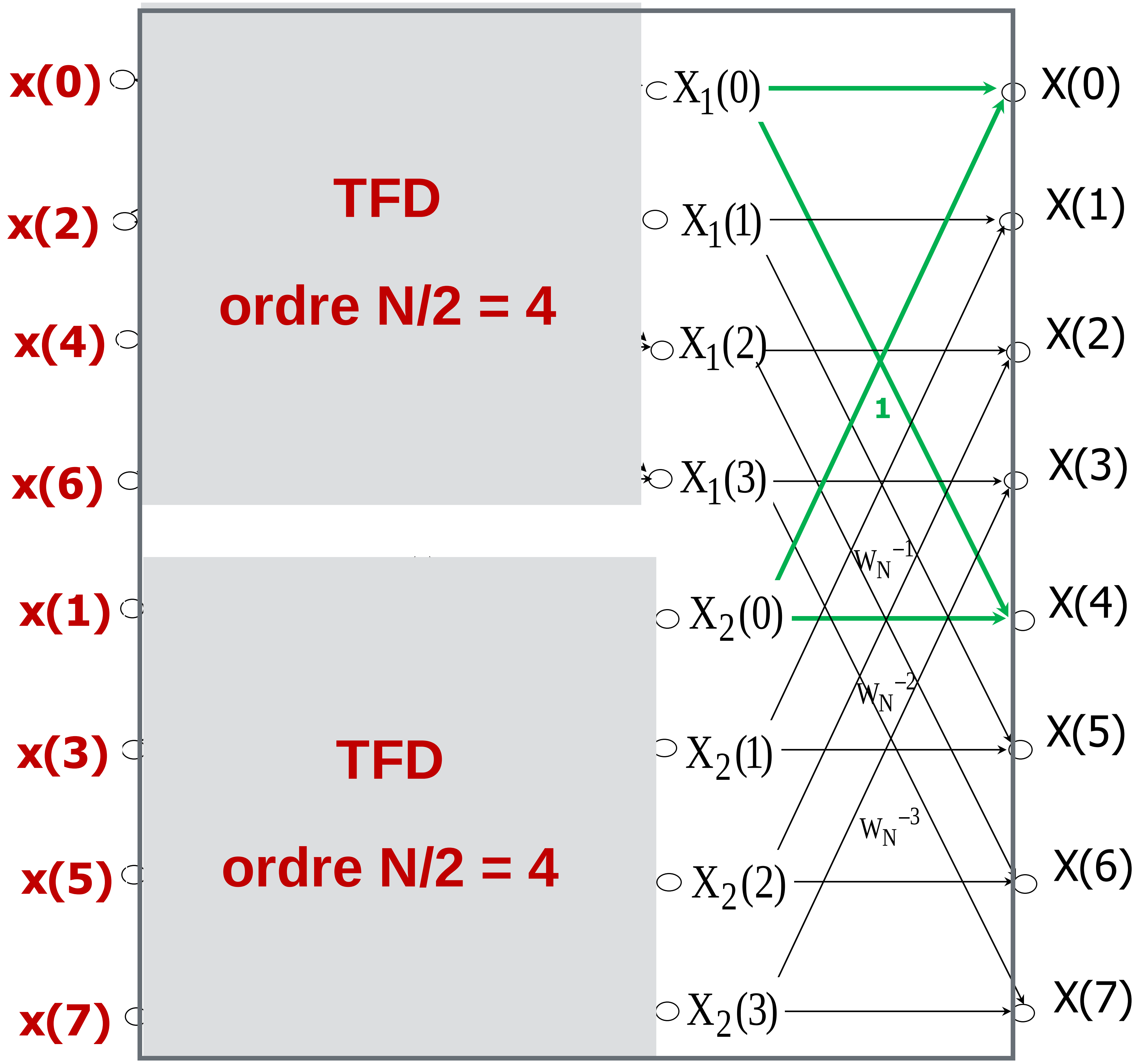
$sur\ x(1), x(3), x(5), x(7)$

Indices impairs
(TFD d'ordre N/2 = 4)

$$2 \times \left(\frac{N}{2}\right)^2 + N = \frac{N^2}{2} + N = 40 < N^2 = 64\ opérations\ (+/\times)$$



Papillon de la FFT
= 2 opérations +/x



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = X_1(n) + W_N^{-n} X_2(n),\ n = 0, \dots, 7$$

Etape 1

$sur\ x(0), x(2), x(4), x(6)$

Indices pairs
(TFD d'ordre N/2 = 4)

$sur\ x(1), x(3), x(5), x(7)$

Indices impairs
(TFD d'ordre N/2 = 4)

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n),\ n = 0, \dots, 3 \qquad X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n),\ n = 0, \dots, 3$$

Etape 2

$sur\ x(0), x(4)$

Indices pairs
(TFD d'ordre N/4=2)

$sur\ x(2), x(6)$

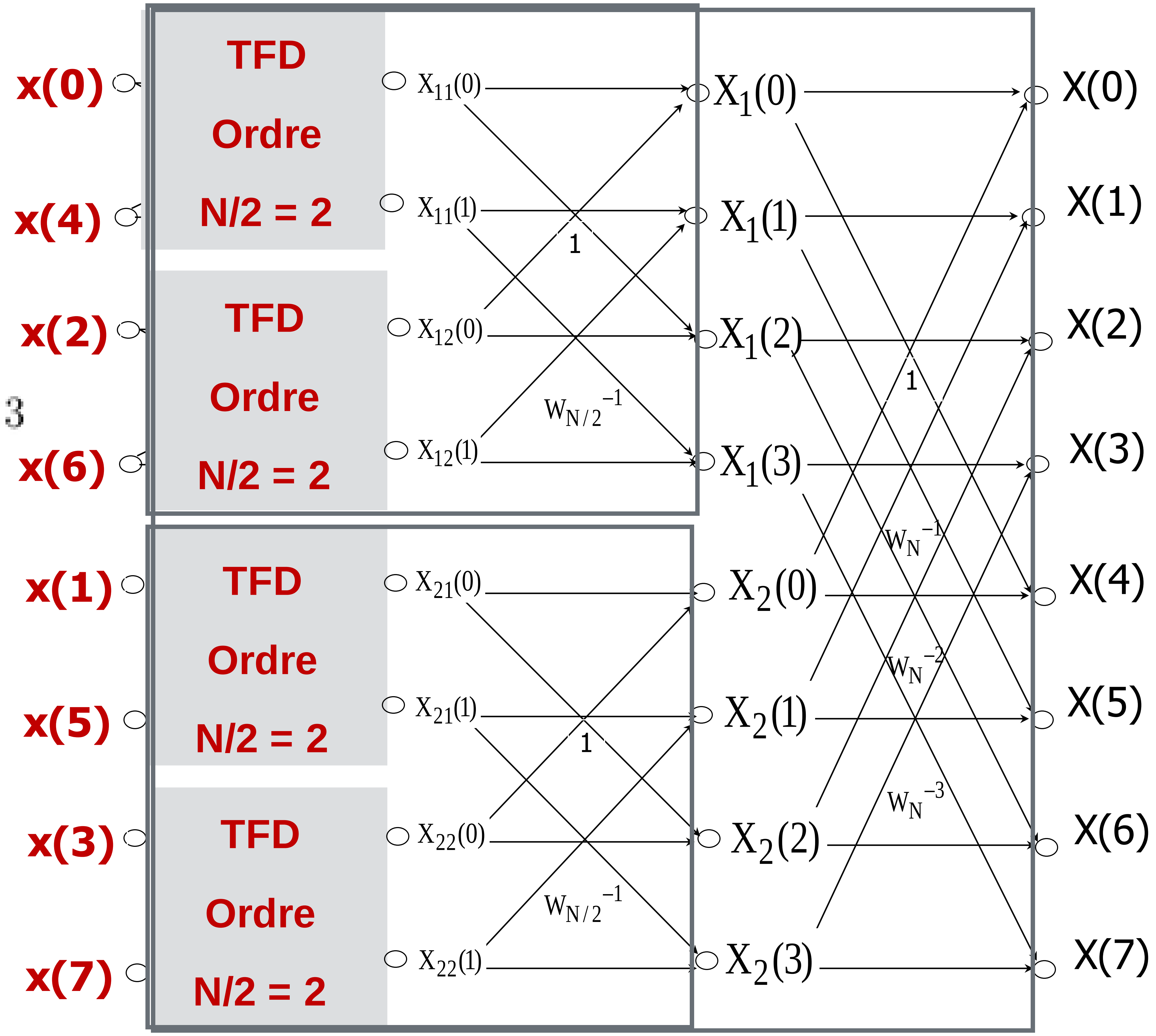
Indices impairs
(TFD d'ordre N/4=2)

$sur\ x(1), x(5)$

Indices pairs
(TFD d'ordre N/4=2)

$sur\ x(3), x(7)$

Indices impairs
(TFD d'ordre N/4=2)



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

Exemple pour $N = 2^p = 2^3 = 8$ points de signal

$$X(n) = X_1(n) + W_N^{-n} X_2(n), \; n = 0, \dots, 7$$

Etape 1

$sur\ x(0), x(2), x(4), x(6)$

Indices pairs
(TFD d'ordre N/2 = 4)

$sur\ x(1), x(3), x(5), x(7)$

Indices impairs
(TFD d'ordre N/2 = 4)

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n), \; n = 0, \dots, 3 \qquad X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n), \; n = 0, \dots, 3$$

Etape 2

$sur\ x(0), x(4)$

Indices pairs
(TFD d'ordre N/4=2)

$sur\ x(2), x(6)$

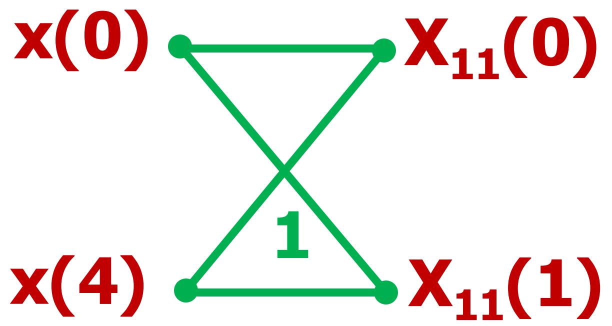
Indices impairs
(TFD d'ordre N/4=2)

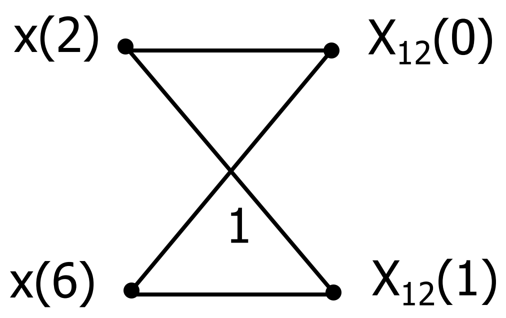
$sur\ x(1), x(5)$

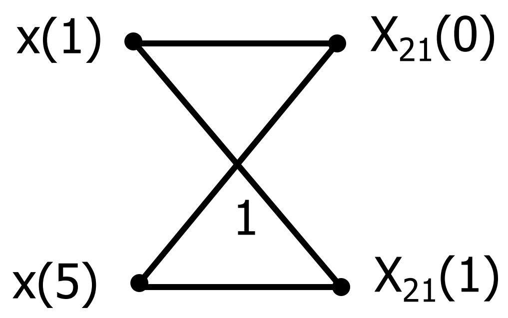
Indices pairs
(TFD d'ordre N/4=2)

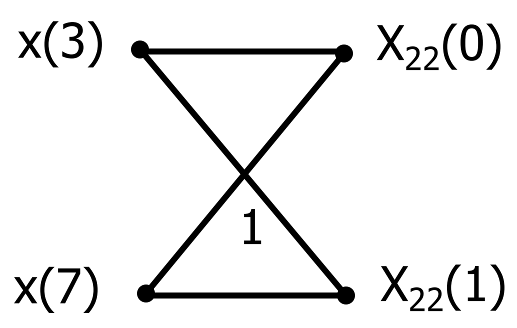
$sur\ x(3), x(7)$

Indices impairs
(TFD d'ordre N/4=2)

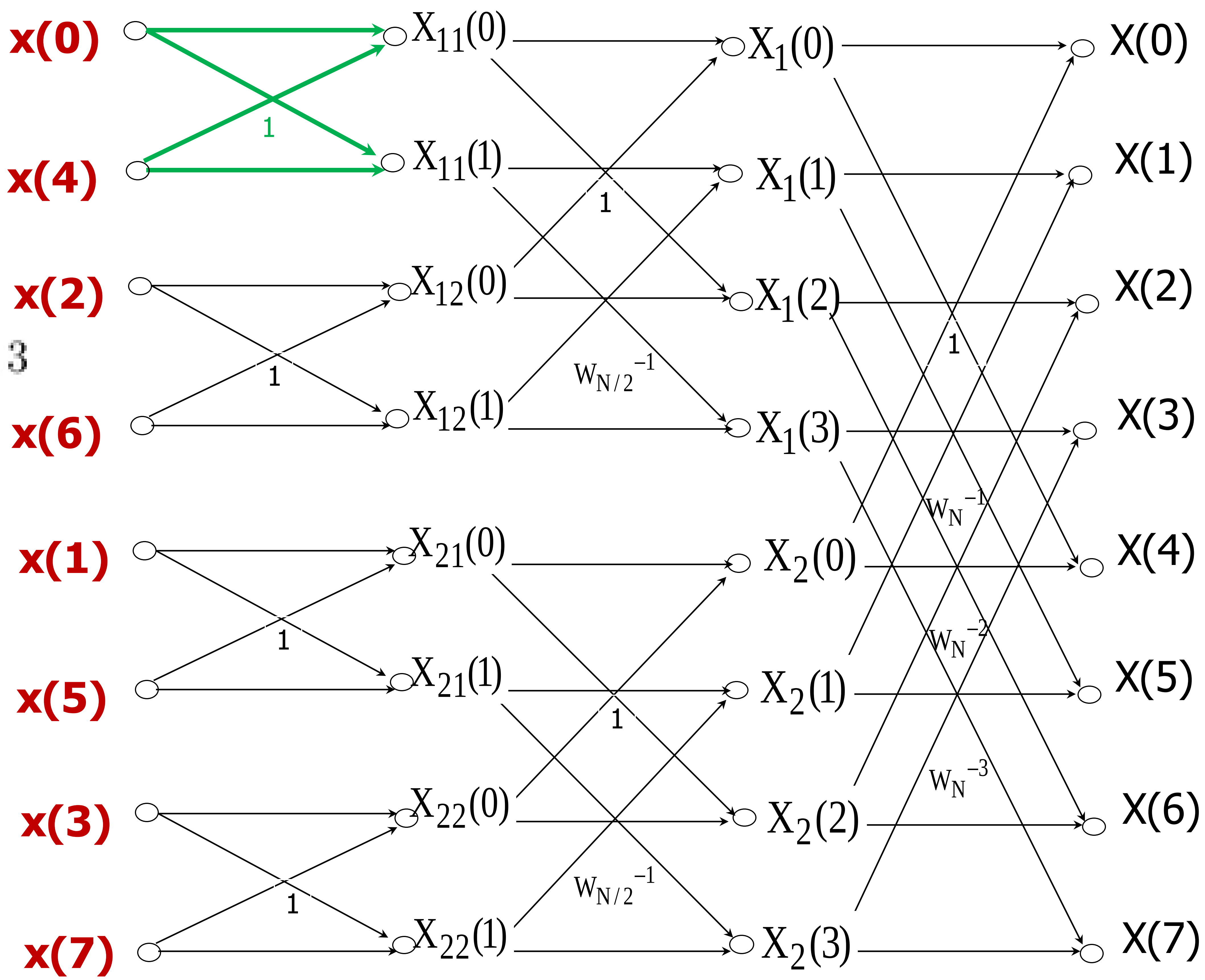








$$4 \times \left(\frac{N}{4}\right)^2 + 2 \times \left(\frac{N}{2}\right) = \frac{N^2}{4} + N = 24 < N^2 = 64\ opérations\ (+/\times)$$



Transformée de Fourier Discrète

Algorithme de la FFT (Cooley Tuckey)

<u>Exemple pour $N = 2^p = 2^3 = 8$ points de signal</u>

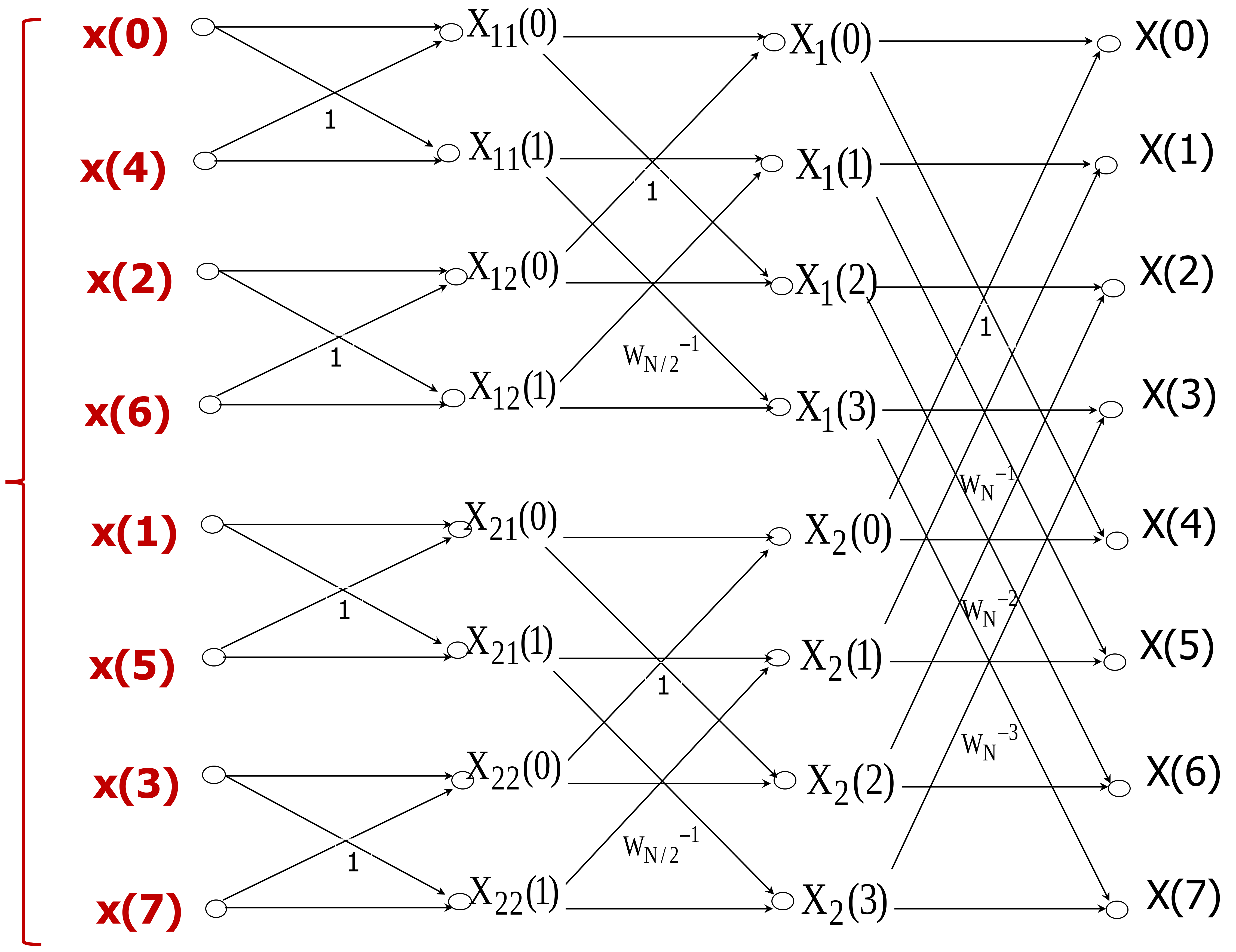
$$\boldsymbol{p \times \left(\frac{N}{2}\right) \times 2 = 24 < N^2 = 64\ opérations}\ (+/\times)$$

Présentation des données à l'algorithme :

Entrelacement temporel

k	rep. binaire	renv. Bits	nouvel ind.	echantillon
0	"000"	"000"	0	x(0)
1	"001"	"100"	4	x(4)
2	"010"	"010"	2	x(2)
3	"011"	"110"	6	x(6)
4	"100"	"001"	1	x(1)
5	"101"	"101"	5	x(5)
6	"110"	"011"	3	x(3)
7	"111"	"111"	7	x(7)

algorithme de renversement de l'adresse binaire
(« bit reversal »)



Transformée de Fourier Discrète

En résumé

→ Echantillonnage temporel => périodisation spectrale

- Respecter la condition de Shannon
- Attention à la lecture des tracés de la TFD, à la lecture de l'échelle fréquentielle

→ Signal de durée limitée => distorsion de la TFD attendue (analyse spectrale numérique avec un pouvoir séparateur limité, apparition d'ondulations)

- Utiliser plusieurs fenêtres de pondération du signal

  => différents pouvoirs séparateurs, différents taux d'ondulation pour l'analyse spectrale numérique

→ Echantillonnage spectral

- Attention à la mauvaise visualisation de la TFD (résolution insuffisante)

  => nécessité d'interpoler (méthode du zero padding)
- TFD et TFD^{-1} transforment un produit en produit de convolution circulaire

  => si besoin, convolution linéaire = convolution circulaire en prolongeant les signaux par des zéros

→ Algorithme de calcul rapide : FFT = Fast Fourier Transform

- Condition nombre de points de signal $N=2^p$ => décomposition en sous suites entrelacées
- Temps de calcul : $N\log_2(N) \ll N^2$ (temps de calcul direct)

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d’auto et d’inter corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

Fonctions d'inter et d'auto corrélation

→ <u>Signaux déterministes</u>

A énergie finie

$$R_{xy}(k) = \sum_{n=-\infty}^{+\infty} x(n)y^*(n-k)$$

A puissance moyenne finie non périodique

$$R_{xy}(k) = \lim_{N\to+\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)y^*(n-k)$$

A puissance moyenne finie périodique de période N_0

$$R_{xy}(k) = \frac{1}{N_0} \sum_{n=0}^{N_0-1} x(n)y^*(n-k)$$

→ <u>Signaux aléatoires</u>

$$R_{xy}(k) = E\left[x(n)y^*(n-k)\right]$$

Fonctions d'inter et d'auto corrélation

→ Signaux aléatoires : première estimation possible

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Exemple :

$$x(t) = \cos(2\pi f_0 t + \phi)$$

$$f_0 = 100\ Hz;$$

$$\phi\ v.a.\ u.r.\ sur\ [0, 2\pi]$$

Fonctions d’inter et d’auto corrélation

→ Signaux aléatoires : première estimation possible

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Exemple :

$$x(t) = \cos(2\pi f_0 t + \phi)$$

$$f_0 = 100\ Hz;$$

$$\phi\ v.a.\ u.r.\ sur\ \ [0, 2\pi]$$

$$\boxed{R_x(\tau) = \frac{A^2}{2}\cos(2\pi f_0 \tau)}$$

Fonctions d'inter et d'auto corrélation

→ Signaux aléatoires : première estimation possible

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \le k \le N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Exemple :

$$x(t) = \cos(2\pi f_0 t + \phi)$$

$$f_0 = 100\ Hz;$$

$$\phi\ v.a.\ u.r.\ sur\ \ [0, 2\pi]$$

$$\boxed{R_x(\tau) = \frac{A^2}{2}\cos(2\pi f_0 \tau)}$$

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;   %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;        %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]+rand*2*pi);

% Calcul et tracé de son autocorrélation biaisée
Rx=xcorr(x);
figure; plot([-N*Te:Te:N*Te],Rx);
xlabel('Temps (s)'); ylabel('R_x')
```

Fonctions d’inter et d’auto corrélation

→ <u>Signaux aléatoires : première estimation possible</u>

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Code Matlab :

Exemple :

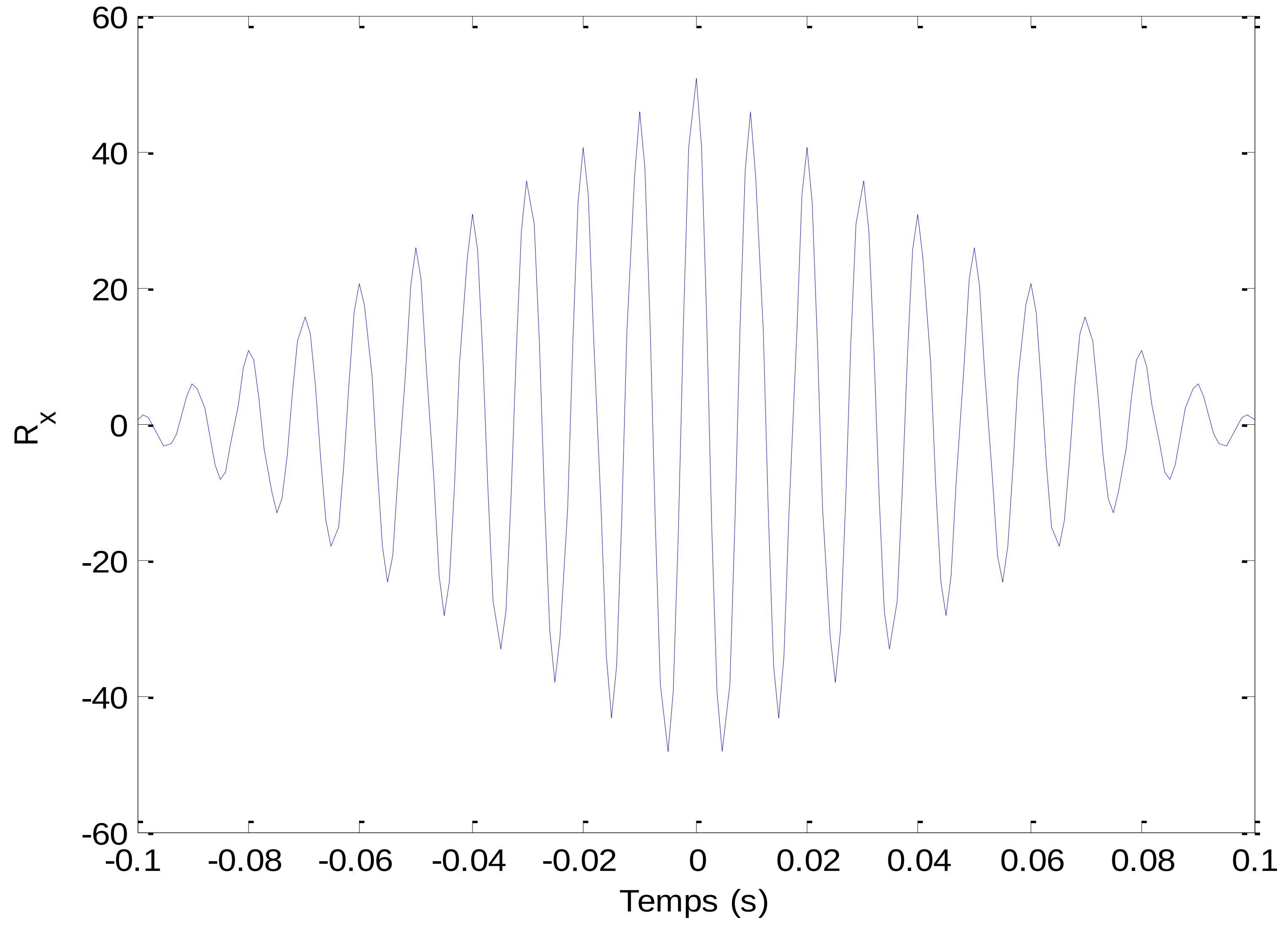
$$x(t) = \cos(2\pi f_0 t + \phi)$$

$$f_0 = 100\ Hz;$$

$$\phi\ v.a.\ u.r.\ sur\ [0, 2\pi]$$

$$\boxed{R_x(\tau) = \frac{A^2}{2}\cos(2\pi f_0 \tau)}$$

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]+rand*2*pi);

% Calcul et tracé de son autocorrélation biaisée
Rx=xcorr(x);
figure; plot([-N*Te:Te:N*Te],Rx);
xlabel('Temps (s)'); ylabel('R_x')
```



Fonctions d’inter et d’auto corrélation

→ Signaux aléatoires : première estimation possible

$$\hat{R}_{xy}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\hat{R}_{xy}(-k) = \hat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Code Matlab :

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;     %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;       %nombre d'échantillons

%Génération du signal
x=cos(2*pi*f0*[0:Te:N*Te]+rand*2*pi);

% Calcul et tracé de son autocorrélation biaisée
Rx=xcorr(x);
figure; plot([-N*Te:Te:N*Te],Rx);
xlabel('Temps (s)'); ylabel('R_x')
```

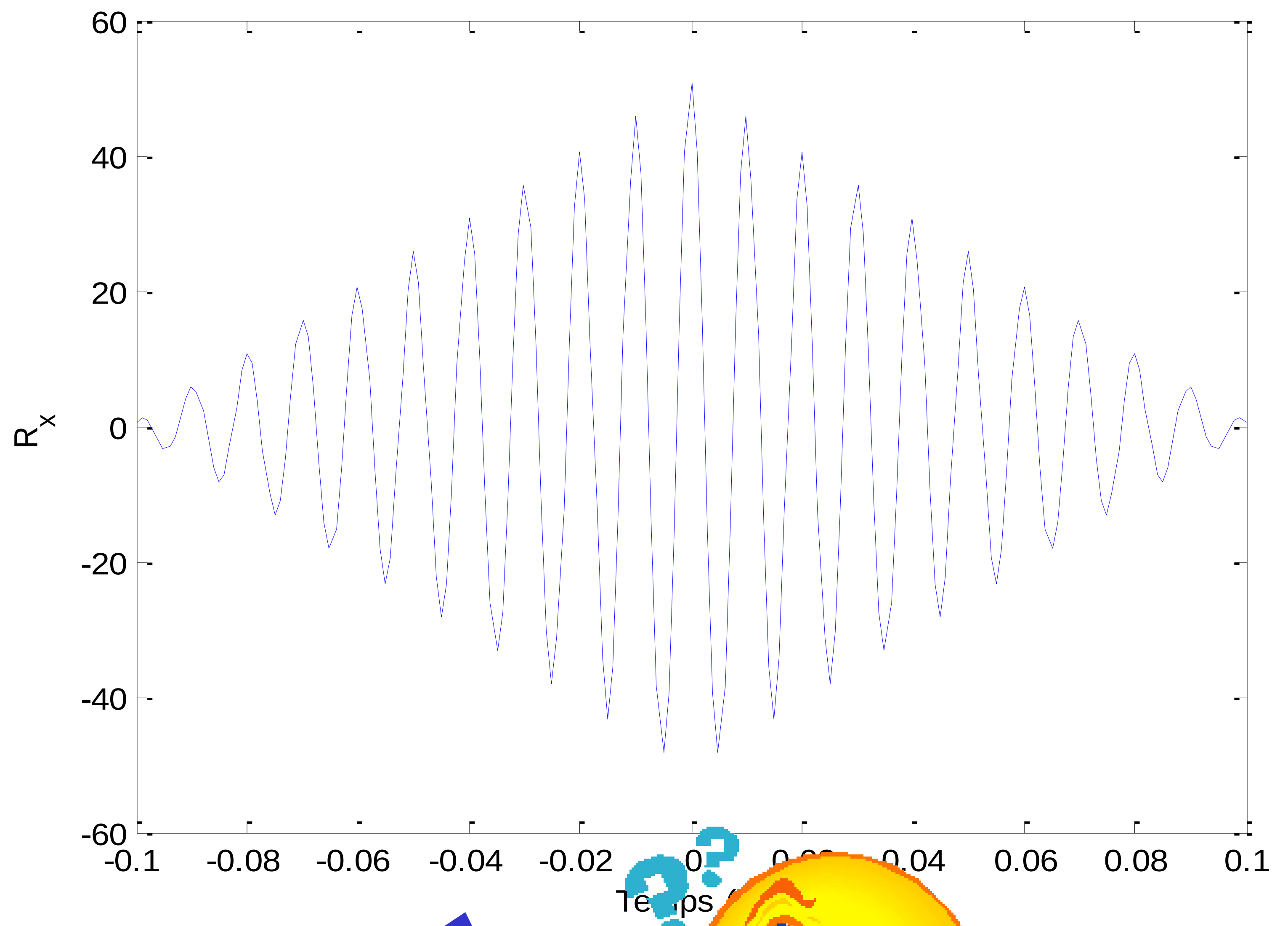
Exemple :

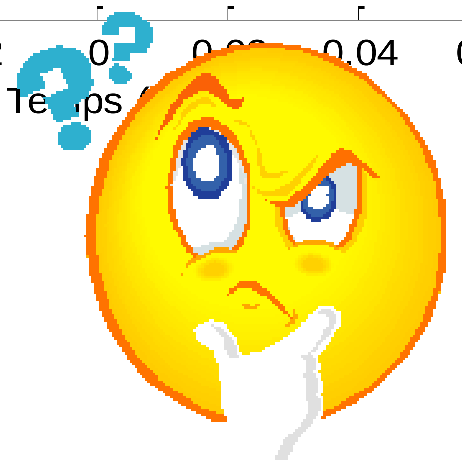
$$x(t) = \cos(2\pi f_0 t + \phi)$$

$$f_0 = 100\ Hz;$$

$$\phi\ v.a.\ u.r.\ sur\ [0, 2\pi]$$

$$\boxed{R_x(\tau) = \frac{A^2}{2}\cos(2\pi f_0 \tau)}$$





Fonctions d'inter et d'auto corrélation

→ Signaux aléatoires : première estimation possible

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Fonctions d'inter et d'auto corrélation

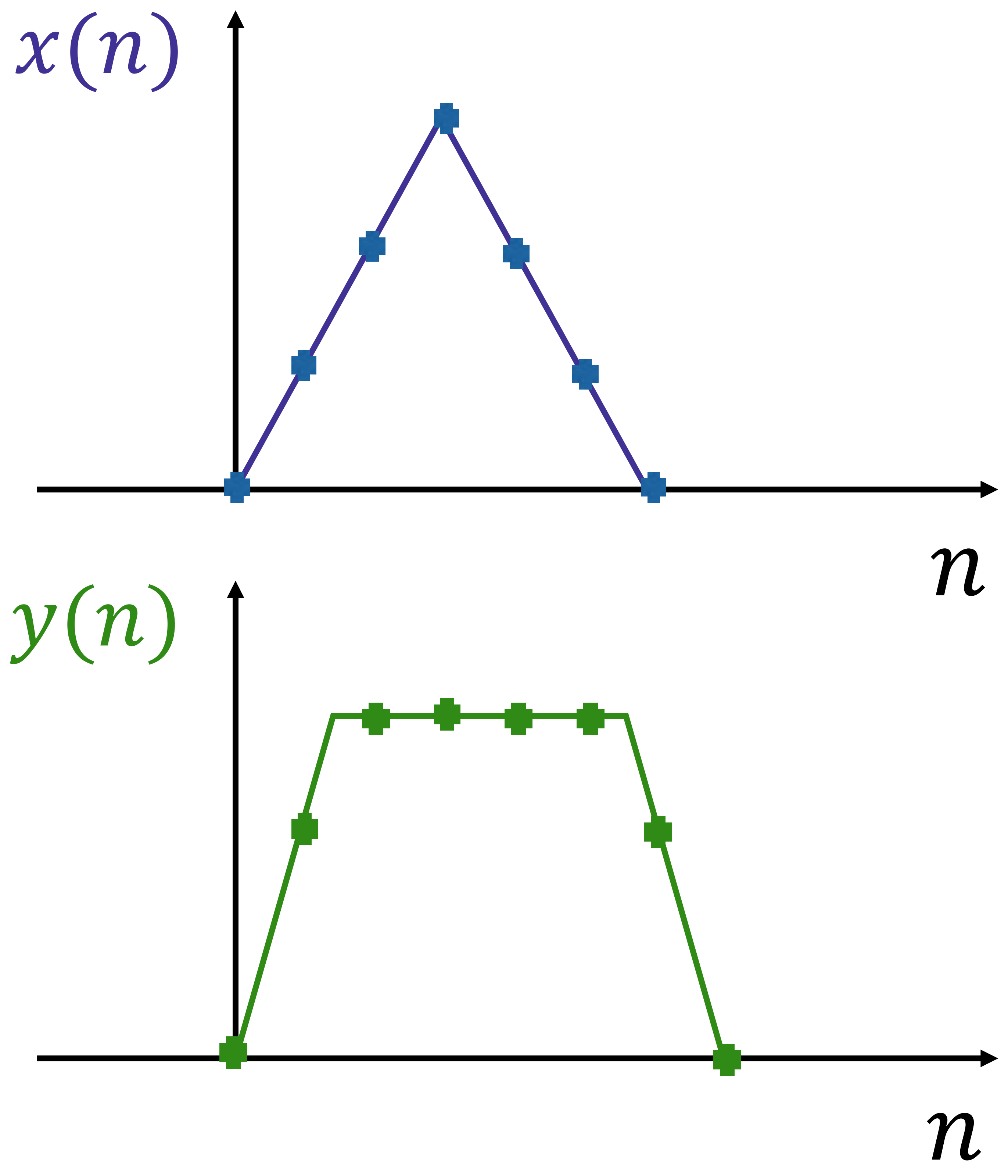
→ <u>Signaux aléatoires : première estimation possible</u>

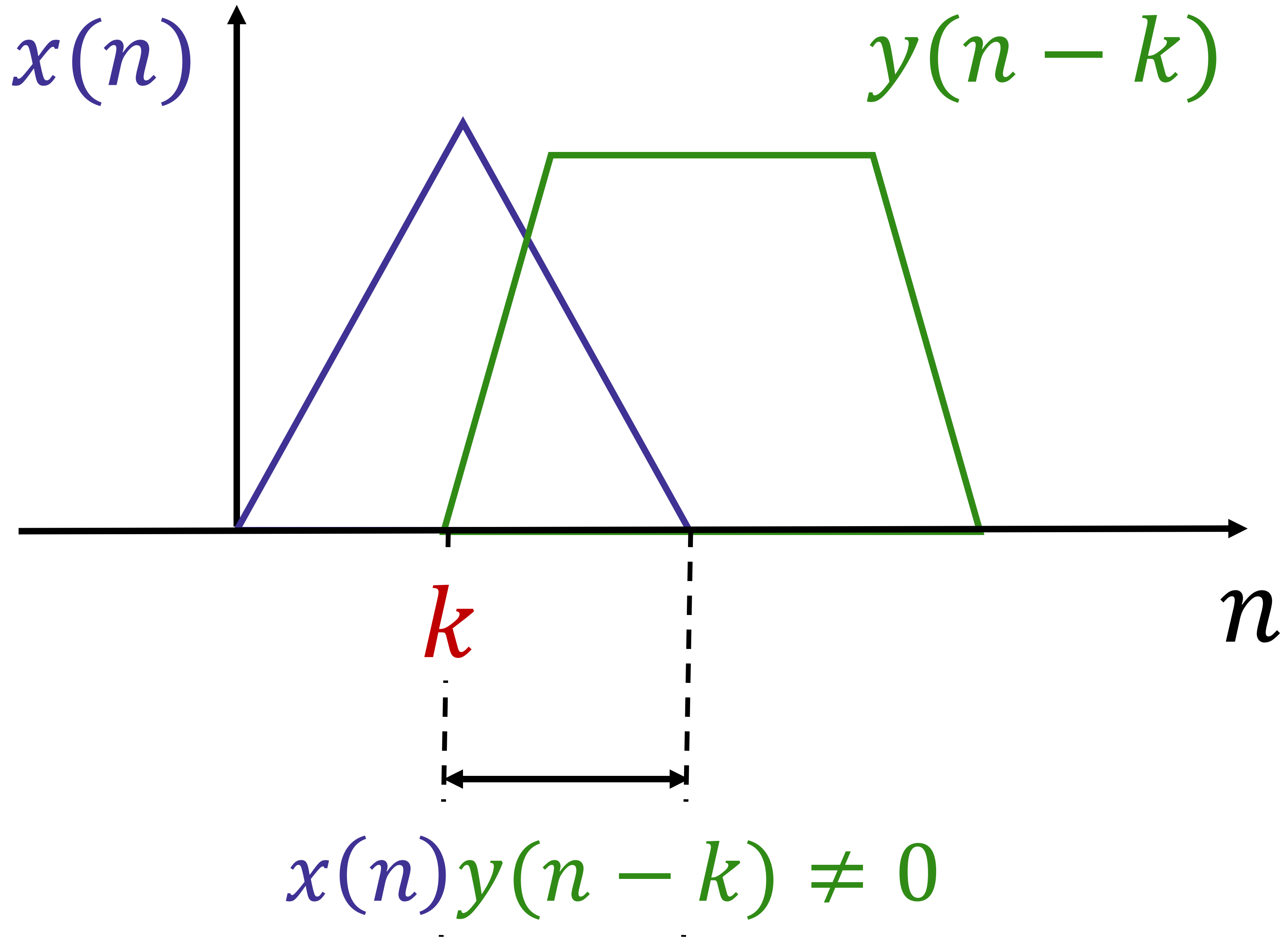
$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Exemple :





Fonctions d'inter et d'auto corrélation

→ Signaux aléatoires : première estimation possible

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=\cancel{0}\ k}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Fonctions d'inter et d'auto corrélation

→ <u>Signaux aléatoires : première estimation possible</u>

$$\widehat{R}_{xy}(k) = \frac{1}{N} \sum_{\substack{n=\cancel{0} \\ k}}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

$$\Rightarrow \quad \boxed{E\left[\widehat{R}_{xy}(k)\right] = \frac{N-|k|}{N} R_{xy}(k)}$$

Fonctions d’inter et d’auto corrélation

→ <u> Signaux aléatoires : première estimation possible</u>

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{\substack{n=\cancel{0}\\ k}}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Exemple :

$$x(t) = \cos(2\pi f_0 t + \phi)$$

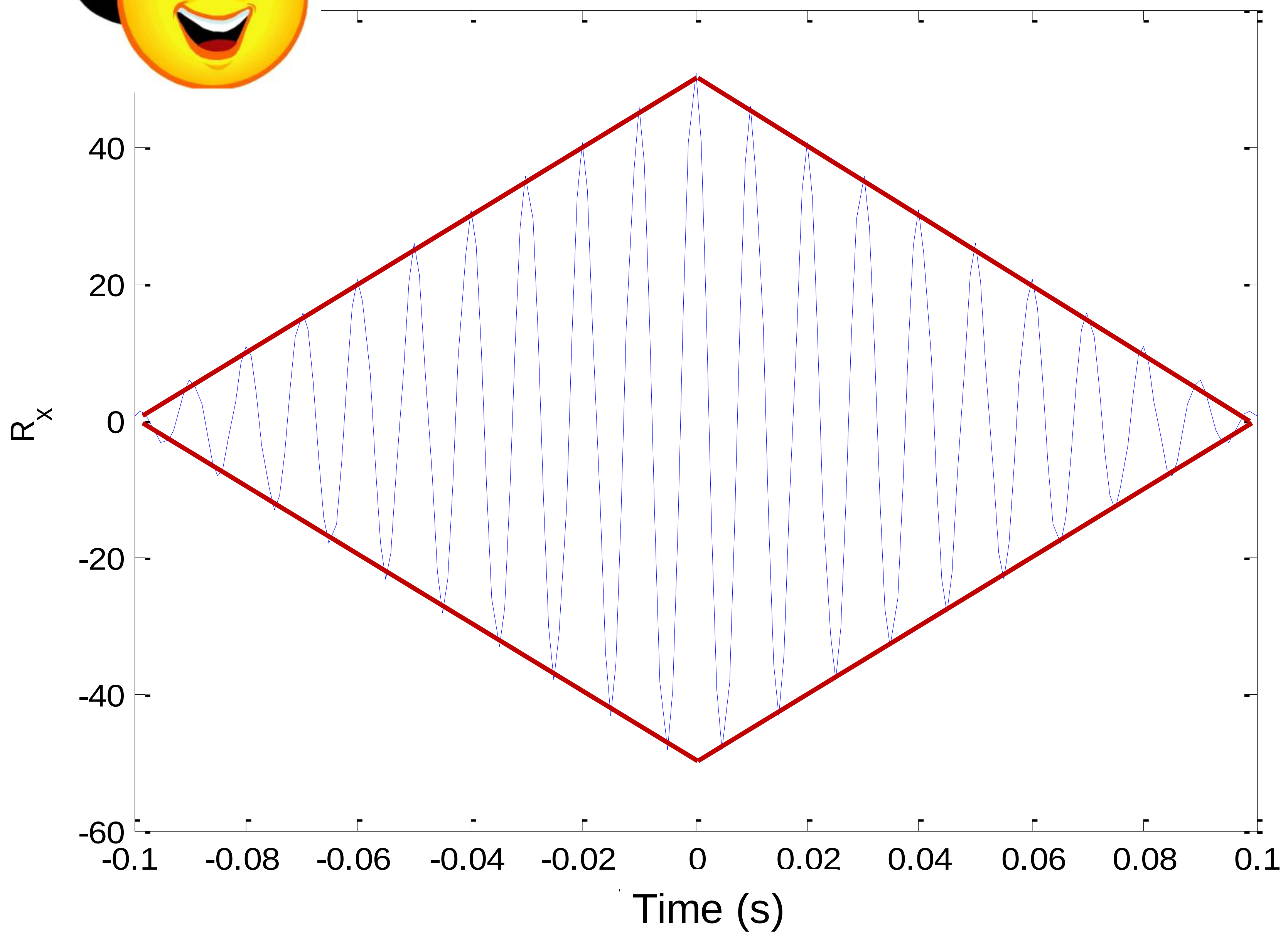
$$f_0 = 100\ Hz;$$

$$\phi\ v.a.\ u.r.\ sur\ [0, 2\pi]$$

$$\boxed{R_x(\tau) = \frac{A^2}{2}\cos(2\pi f_0 \tau)}$$

Biais Multiplicatif Triangulaire

$$E\left[\widehat{R}_{xy}(k)\right] = \frac{N-|k|}{N} R_{xy}(k)$$



Fonctions d'inter et d'auto corrélation

→ <u>Signaux aléatoires : première estimation possible</u>

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=\cancel{0}\;k}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Estimateur biaisé

Fonctions d'inter et d'auto corrélation

→ <u>Signaux aléatoires : première estimation possible</u>

$$\widehat{R}_{xy}(k) = \frac{1}{N}\sum_{n=\cancel{0}\;k}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Estimateur biaisé

→ <u>Signaux aléatoires : deuxième estimation possible</u>

$$\widehat{R}_{xy}(k) = \frac{1}{N-k}\sum_{n=\cancel{0}\;k}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1$$

$$\widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

Estimateur non biaisé

Fonctions d’inter et d’auto corrélation

→ <u>Signaux aléatoires : deuxième estimation possible</u>

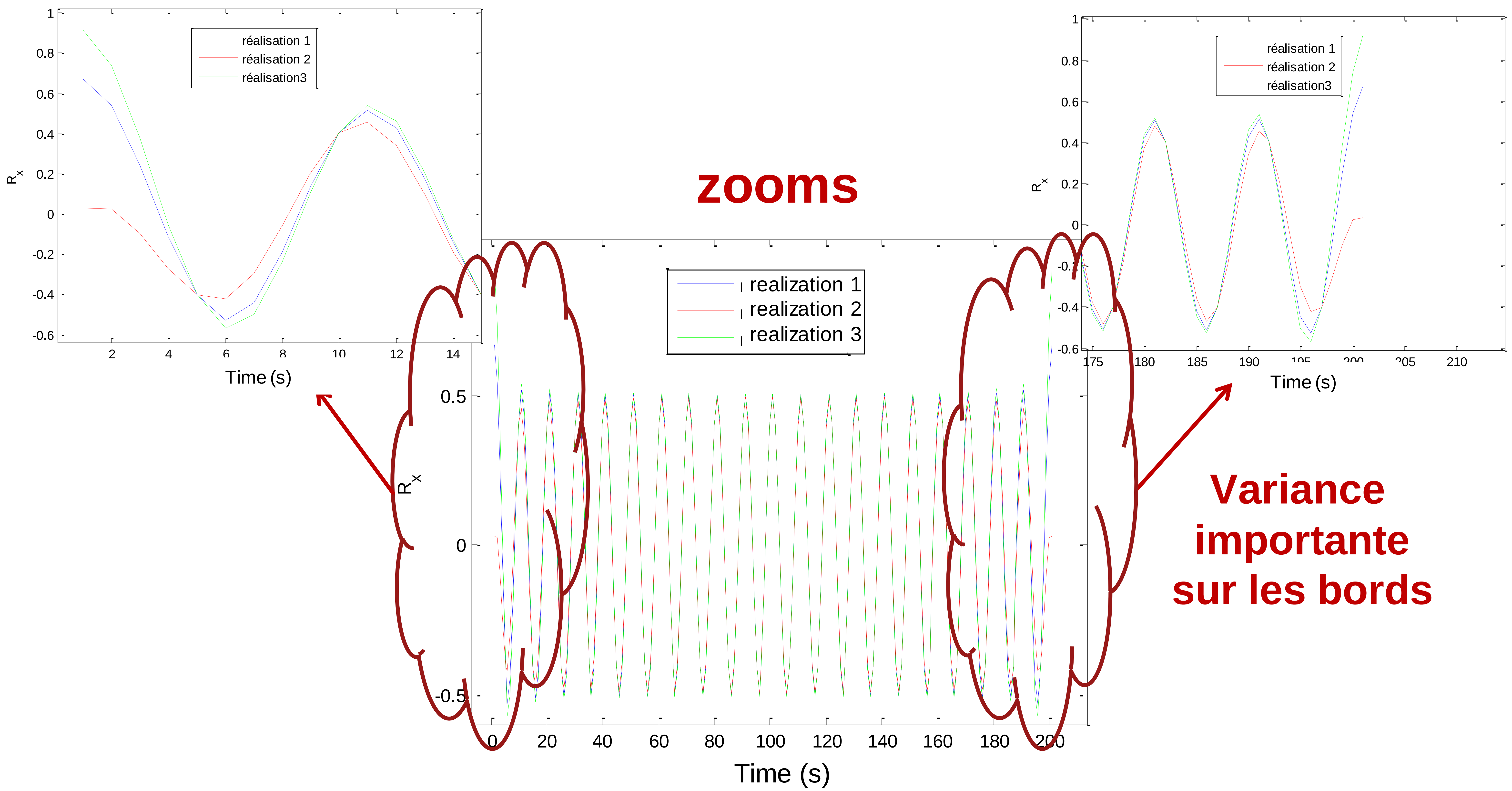
Estimateur non biaisé

$$\widehat{R}_{xy}(k) = \frac{1}{N-k}\sum_{n=\cancel{0}\;k}^{N-1} x(n)y^*(n-k) \quad 0 \le k \le N-1 \qquad \widehat{R}_{xy}(-k) = \widehat{R}^*_{xy}(k)$$

(Symétrie hermitienne)

<u>Code Matlab :</u>

```
%Paramètres
f0=100;      %fréquence du cosinus
Fe=1000;    %fréquence d'échantillonnage
Te=1/Fe;     %période d'échantillonnage
N=100;        %nombre d'échantillons

% Calcul et tracé de son autocorrélation non biaisée
% pour différentes réalisations de signal
x1=cos(2*pi*f0*[0:Te:N*Te]+rand*2*pi);
Rx1=xcorr(x1,'unbiased');
x2=cos(2*pi*f0*[0:Te:N*Te]+rand*2*pi);
Rx2=xcorr(x2,'unbiased');
x3=cos(2*pi*f0*[0:Te:N*Te]+rand*2*pi);
Rx3=xcorr(x3,'unbiased');
figure
plot(Rx1); hold on; plot(Rx2,'r'); plot(Rx3,'g')
legend('réalisation 1','réalisation 2','réalisation3')
xlabel('Temps (s)'); ylabel('R_x')
```



Variance importante sur les bords

Fonctions d'inter et d'auto corrélation

→ Estimation dans le domaine fréquentiel

Pour :

$$\widehat{R}_{xy}(k) \propto \sum_{n=-\infty}^{+\infty} x(n)y^*(n-k),\ k = 0, \ldots, N-1$$

Convolution linéaire :

$$(x_1 * x_2)(k) = \sum_{n=-\infty}^{+\infty} x_1(n)x_2(k-n)$$

$$\Downarrow$$

$$\widehat{R}_{xy}(k) \propto x(k) * y^*(-k),\ k = 0, \ldots, N-1$$

Fonctions d'inter et d'auto corrélation

→ Estimation dans le domaine fréquentiel

Pour :
$$\widehat{R}_{xy}(k) \propto \sum_{n=-\infty}^{+\infty} x(n)y^*(n-k),\ k=0,...,N-1$$

Convolution linéaire :
$$(x_1 * x_2)(k) = \sum_{n=-\infty}^{+\infty} x_1(n)x_2(k-n)$$

$\Downarrow$

$$\widehat{R}_{xy}(k) \propto x(k) * y^*(-k),\ k=0,...,N-1$$

$\Downarrow$

$$\widehat{R}_{xy}(k) \propto TFD^{-1}\left[X(n)Y^*(n)\right], 0 \leq n \leq N-1$$

Estimateur fréquentiel

Fonctions d'inter et d'auto corrélation

→ Estimation dans le domaine fréquentiel

Pour :

$$\widehat{R}_{xy}(k) \propto \sum_{n=-\infty}^{+\infty} x(n)y^*(n-k),\ k = 0, \dots, N-1$$

N^2 opérations $(+/\times)$

Pour des signaux de taille N points

Convolution linéaire :

$$(x_1 * x_2)(k) = \sum_{n=-\infty}^{+\infty} x_1(n)x_2(k-n)$$

⬇

$$\widehat{R}_{xy}(k) \propto x(k) * y^*(-k),\ k = 0, \dots, N-1$$

⬇

$$\widehat{R}_{xy}(k) \propto TFD^{-1}\left[X(n)Y^*(n)\right], 0 \leq n \leq N-1$$

Estimateur fréquentiel

Temps de calcul : $N + 3Nlog_2(N) \ll N^2$ **opérations** $(+/\times)$

Fonctions d'inter et d'auto corrélation

→ Estimation dans le domaine fréquentiel

Pour :
$$\widehat{R}_{xy}(k) \propto \sum_{n=-\infty}^{+\infty} x(n)y^*(n-k),\ k=0,\ldots,N-1$$

N^2 opérations $(+/\times)$
Pour des signaux de taille N points

Convolution linéaire :
$$(x_1 * x_2)(k) = \sum_{n=-\infty}^{+\infty} x_1(n)x_2(k-n)$$

⬇

$$\widehat{R}_{xy}(k) \propto x(k) * y^*(-k),\ k=0,\ldots,N-1$$

⬇

$$\widehat{R}_{xy}(k) \propto TFD^{-1}\left[X(n)Y^*(n)\right], 0 \le n \le N-1$$

!! Condition : si la TFD transforme un produit en produit de convolution linéaire !!

Estimateur fréquentiel

Temps de calcul : $N + 3Nlog_2(N) \ll N^2$ **opérations** $(+/\times)$

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d’auto et d’inter corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

Densité spectrale de puissance

- **Deux estimateurs de base : définition**

→ Corrélogramme

$$\widehat{S}_x(n) = TFD\left[\widehat{R}_x(k)\right]$$

Densité spectrale de puissance

- **Deux estimateurs de base : définition**

→ Corrélogramme

Vue avant

$$\widehat{S}_x(n) = \boxed{TFD}\left[\widehat{R}_x(k)\right]$$

Vu avant

Corrélogramme biaisé
$$\widehat{R}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n-k)$$

Corrélogramme non biaisé
$$\widehat{R}_x(k) = \frac{1}{N-k}\sum_{n=0}^{N-1} x(n)x^*(n-k)$$

Densité spectrale de puissance

- **Deux estimateurs de base : définition**

→ Corrélogramme

Vue avant

$$\widehat{S}_x(n) = \boxed{TFD}\left[\widehat{R}_x(k)\right]$$

Vu avant

Corrélogramme biaisé

$$\widehat{R}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n-k)$$

Corrélogramme non biaisé

$$\widehat{R}_x(k) = \frac{1}{N-k}\sum_{n=0}^{N-1} x(n)x^*(n-k)$$

→ Périodogramme

$$\widehat{S}_x(n) = \frac{1}{N}\left|TFD\left[x(k)\right]\right|^2$$

Densité spectrale de puissance

- **Deux estimateurs de base : définition**

→ Corrélogramme

Vue avant

$$\widehat{S}_x(n) = \boxed{TFD}\left[\widehat{R}_x(k)\right]$$

Vu avant

Corrélogramme biaisé $\widehat{R}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n-k)$

Corrélogramme non biaisé $\widehat{R}_x(k) = \frac{1}{N-k}\sum_{n=0}^{N-1} x(n)x^*(n-k)$

→ Périodogramme

$$\widehat{S}_x(n) = \frac{1}{N}\left|\boxed{TFD}\left[x(k)\right]\right|^2$$

Vue avant

<u>Remarque :</u> périodogramme ⬄ corrélogramme « biaisé »

Densité spectrale de puissance

- **Deux estimateurs de base : définition**

$\rightarrow$ Corrélogramme

Vue avant

$$\widehat{S}_x(n) = \boxed{TFD}\left[\widehat{R}_x(k)\right]$$

Vu avant

Corrélogramme biaisé $\quad \widehat{R}_x(k) = \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1} x(n)x^*(n-k)$

Corrélogramme non biaisé $\quad \widehat{R}_x(k) = \dfrac{1}{N-k}\displaystyle\sum_{n=0}^{N-1} x(n)x^*(n-k)$

$\rightarrow$ Périodogramme

$$\widehat{S}_x(n) = \frac{1}{N}\left|\boxed{TFD}\left[x(k)\right]\right|^2$$

Vue avant

<u>Remarque :</u> périodogramme ⬄ corrélogramme « biaisé »

- **Deux estimateurs de base : Inconvénients**

$\rightarrow$ Estimateurs non consistants

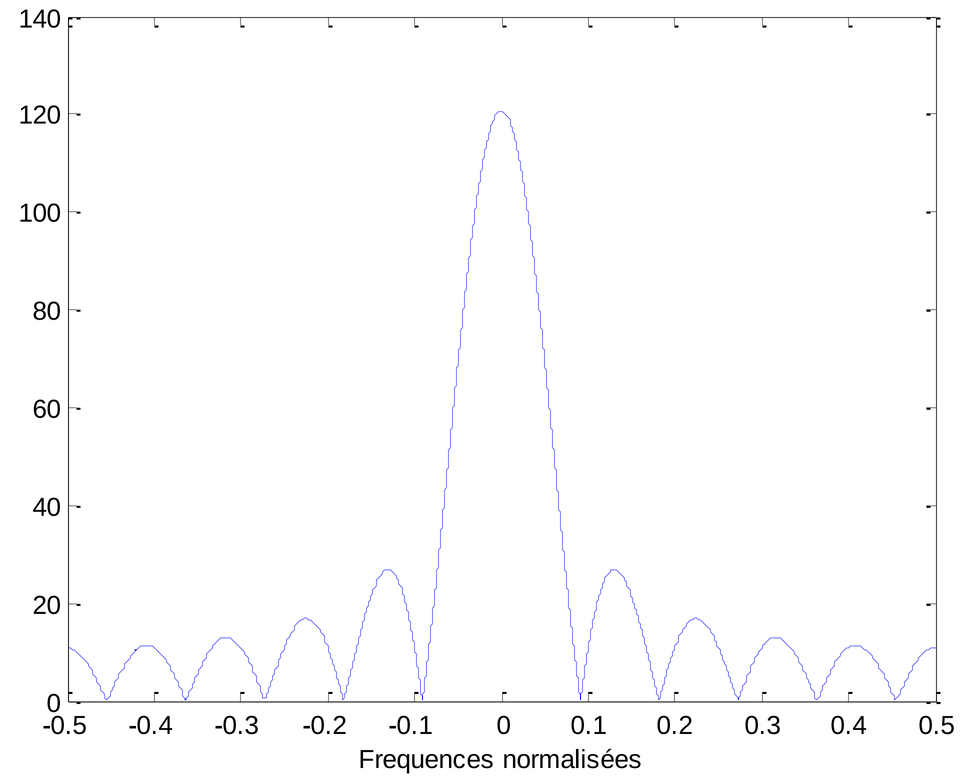
- Biais convolutif : $E\left[\widehat{S}_x(n)\right] = S_x(n) * W(n)$
- Variance indépendante de la durée d'observation du signal :

$$Var\left[\widehat{S}_x(n)\right] \xrightarrow[N\to\infty]{} S_x^2(n)$$

<u>Exemple :</u> périodogramme ⬄ corrélogramme « biaisé »

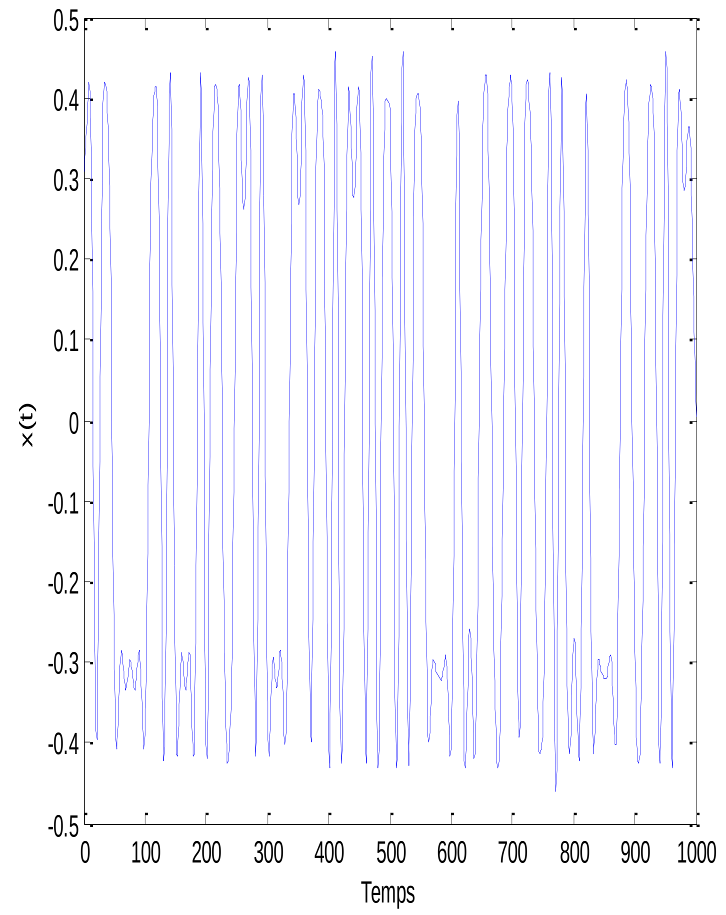
$$E\left[\widehat{S}_x(n)\right] = S_x(n) * TFD\left[1 - \frac{|k|}{N}\right]$$

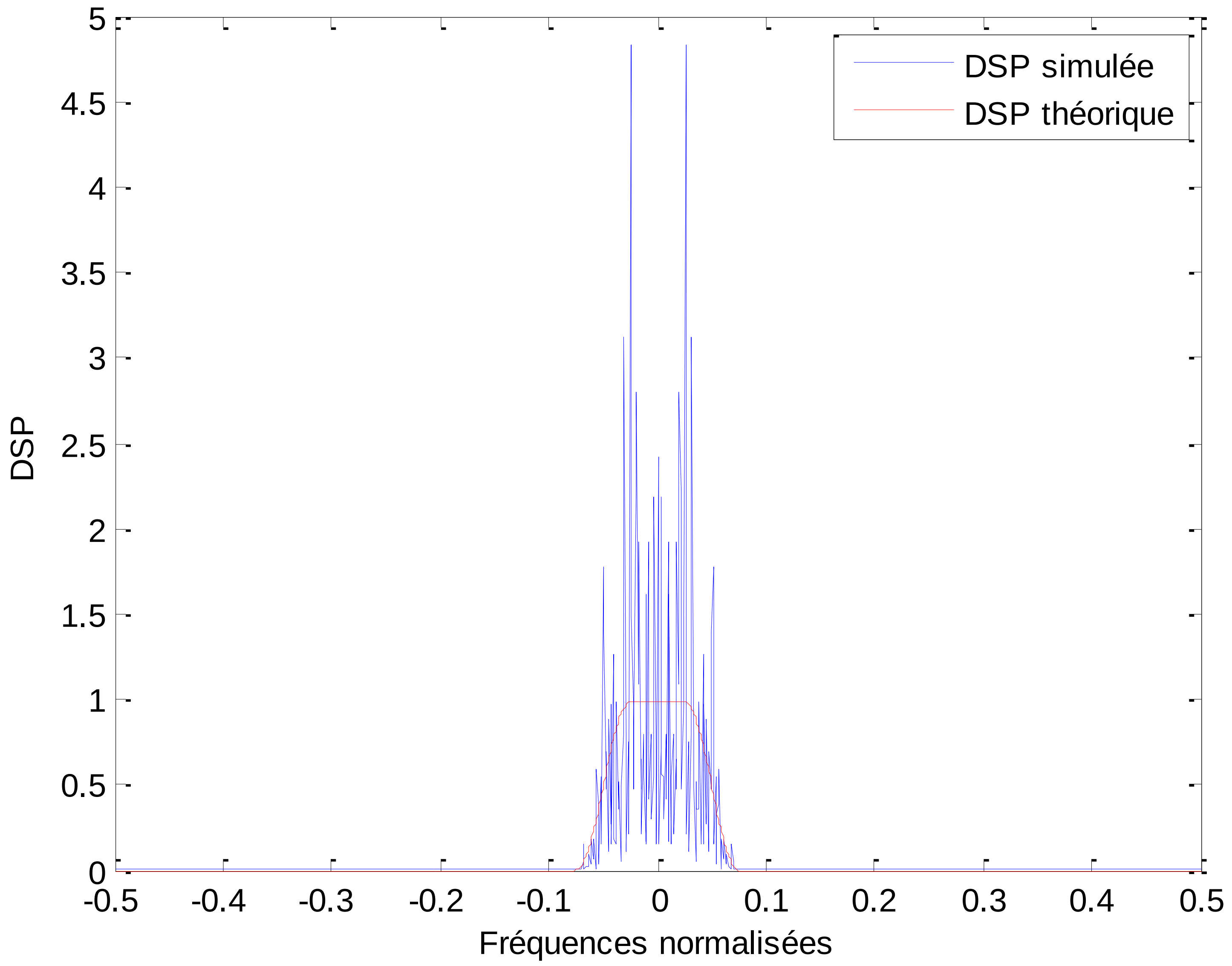
Biais convolutif : Noyau de Fejer



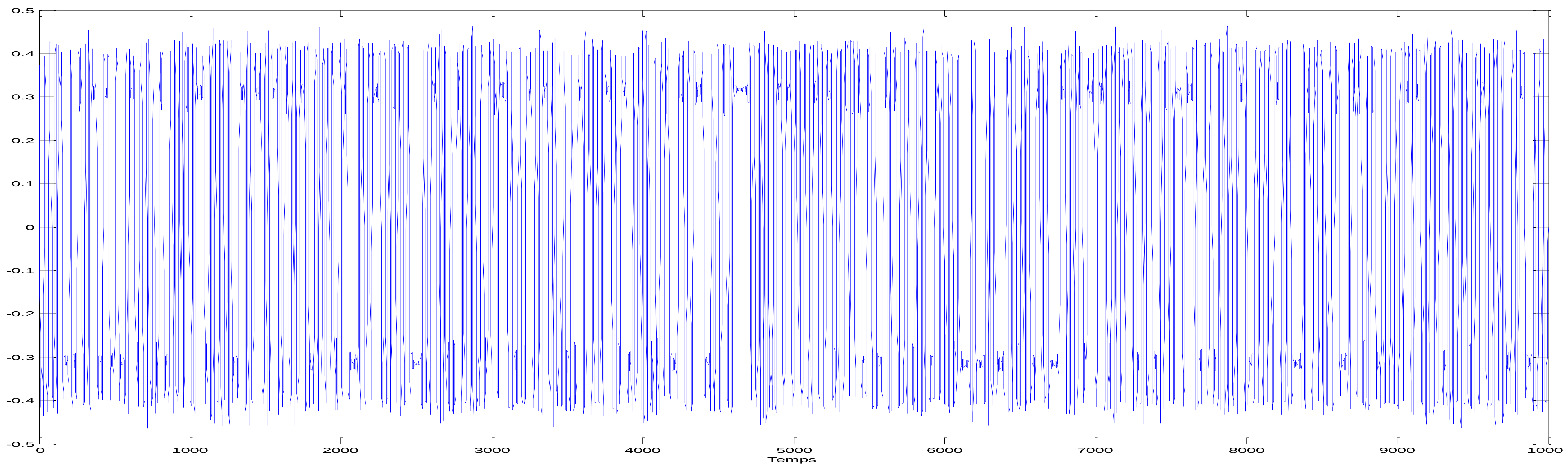
Densité spectrale de puissance

1 000 échantillons de signal



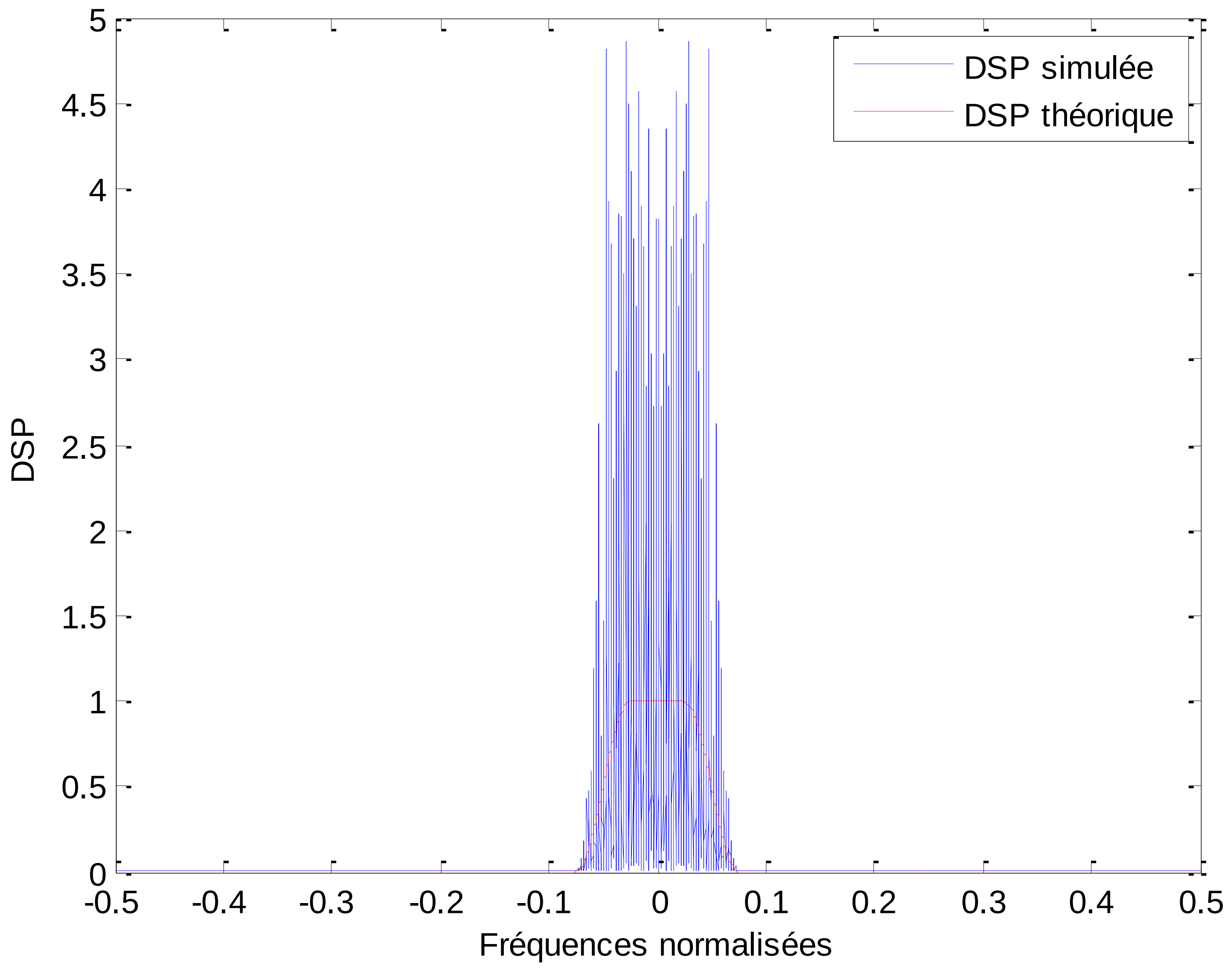


10 000 échantillons de signal



<u>Exemple :</u> périodogramme ⬄ corrélogramme « biaisé »

variance indépendante de la durée d'observation du signal

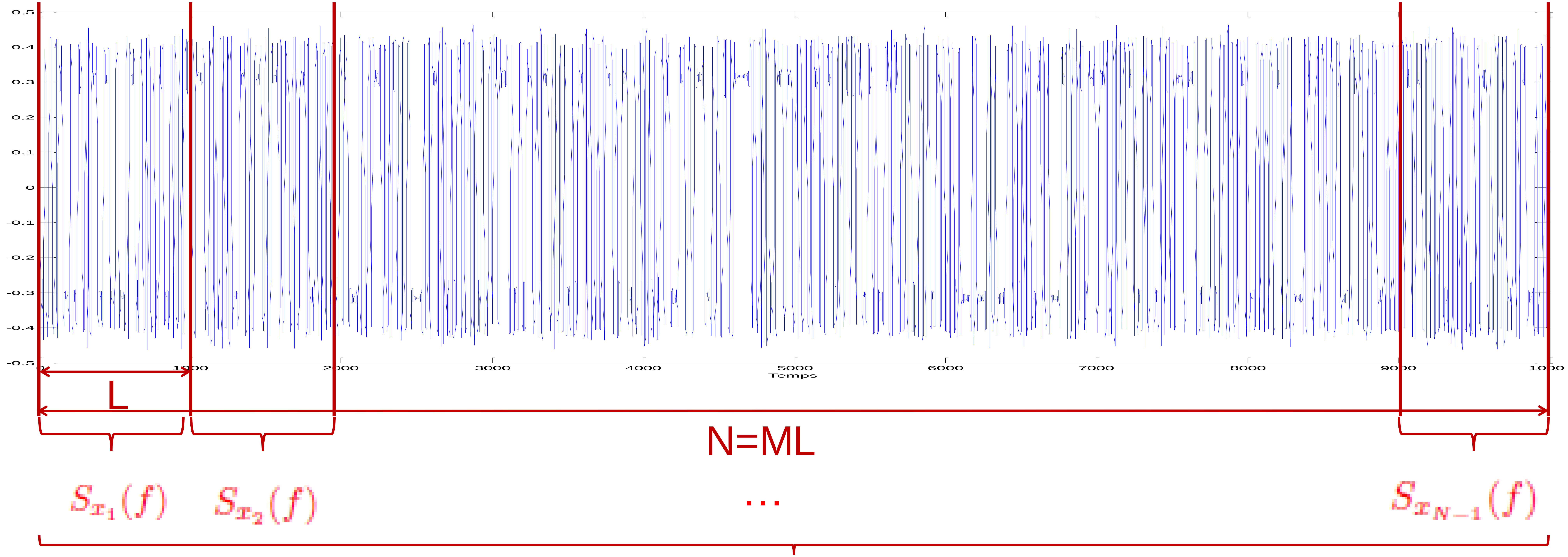


Densité spectrale de puissance

- **Autres estimateurs**

  → <u>Périodogramme cumulé (Bartlett)</u>

<u>Objectif :</u> diminuer la variance d'estimation de la DSP



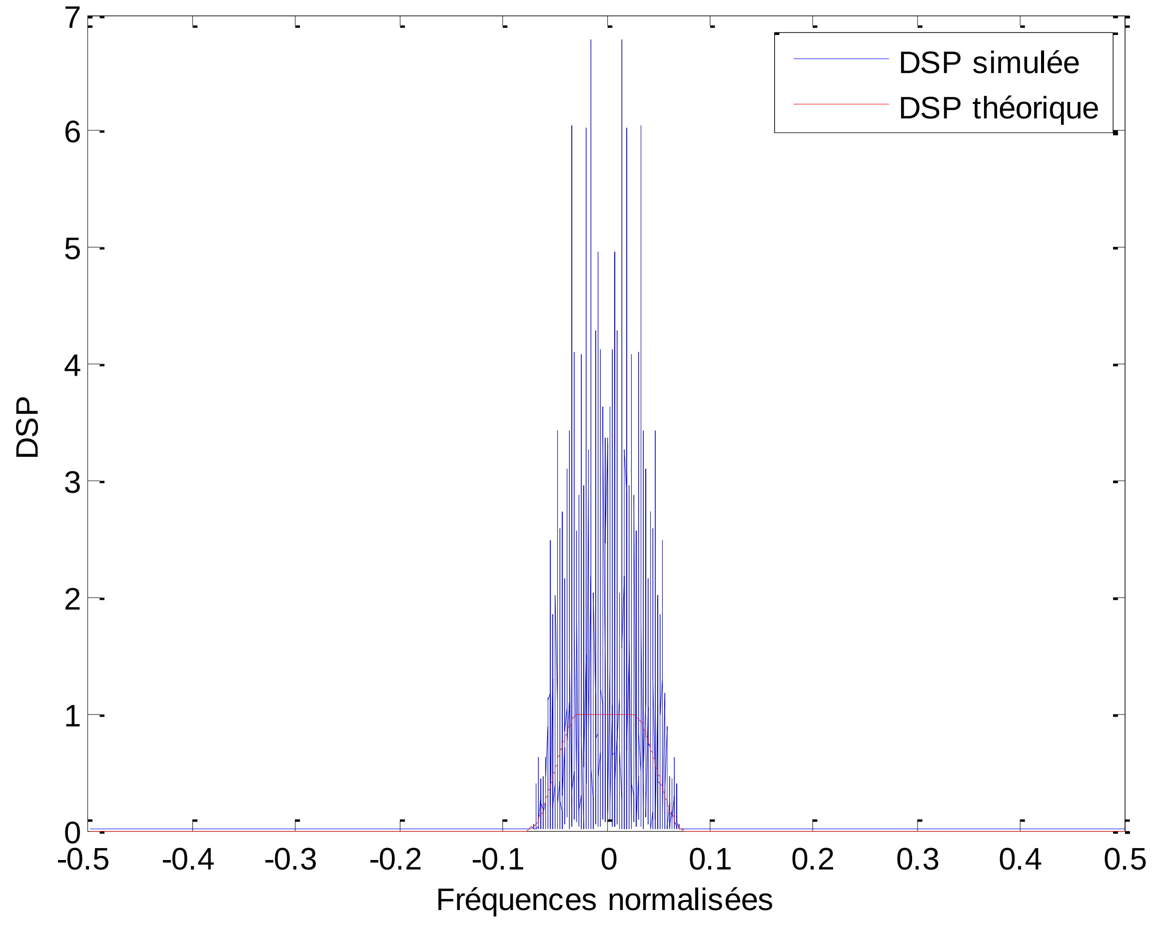
$$S_x(f) = \frac{1}{M}\sum_{i=1}^{M} S_{x_i}(f)$$

Densité spectrale de puissance

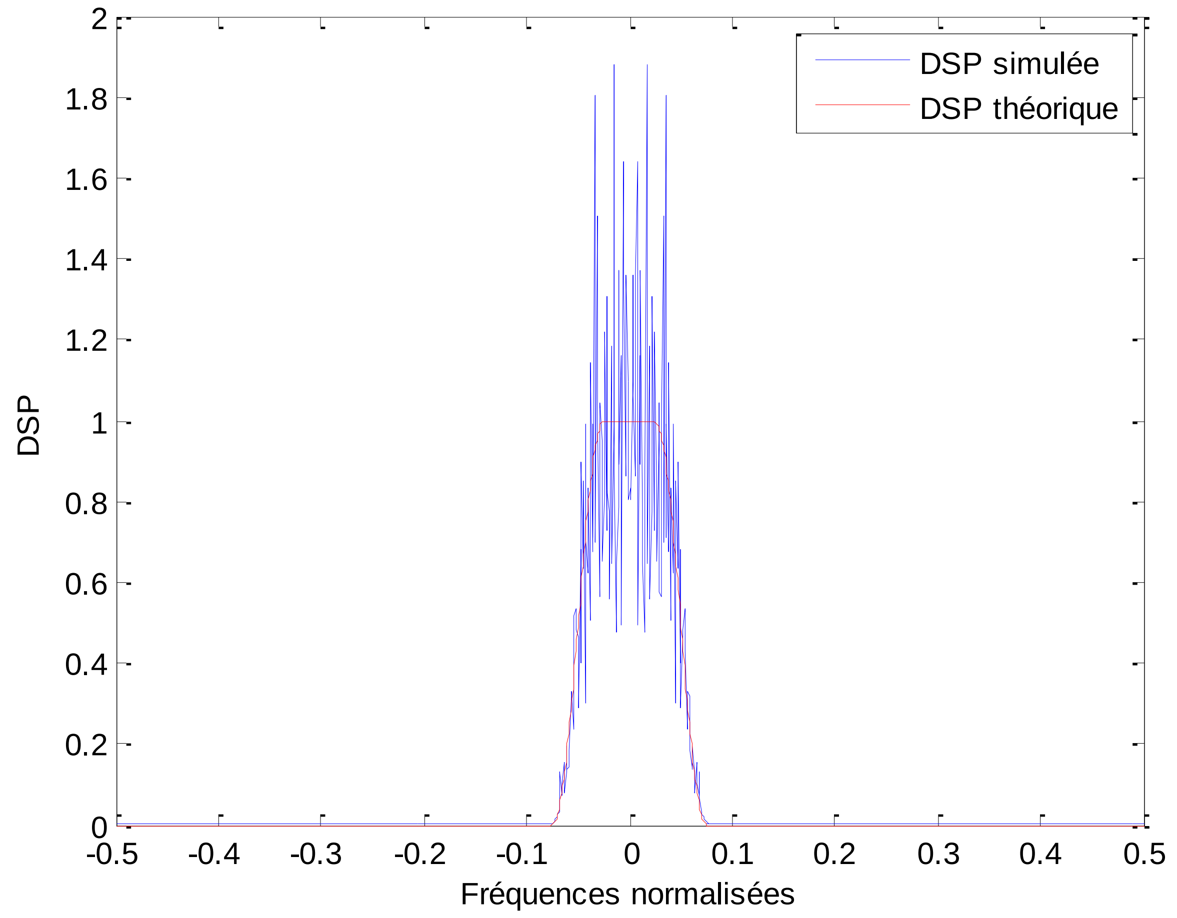
- **Autres estimateurs**

→ <u>Périodogramme cumulé (Bartlett)</u>

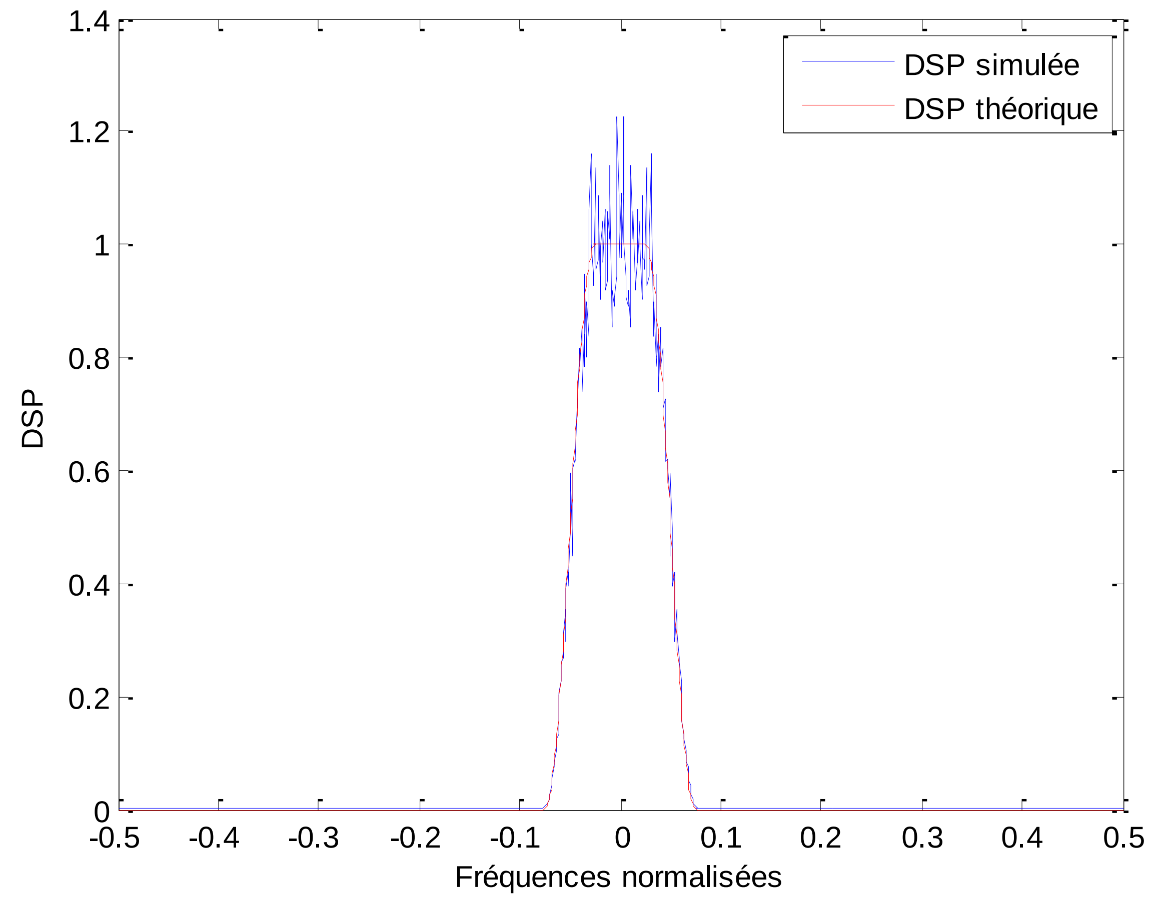
N= 10 000 échantillons, 1 fenêtre



N= 10 000 échantillons, M=10 fenêtres
de L= 1000 échantillons



N= 100 000 échantillons, M=100 fenêtres
de L=1 000 échantillons



<u>Inconvénients :</u>

- Diminution de la résolution spectrale : pas de calcul du spectre en F_e/L au lieu de F_e/N
- Pour une durée d'observation du signal donnée augmentation du biais : lobe central de W(n) de largeur 2/L au lieu de 2/N.

Densité spectrale de puissance

- **Autres estimateurs**

Fenêtre de pondération

→ Périodogramme modifié: $\widehat{S}_x(n) = \frac{1}{N}\left|TFD\left[x(k) \times w(k)\right]\right|^2$

→ Corrélogramme modifié : $\widehat{S}_x(n) = TFD\left[\widehat{R}_x(k) \times w(k)\right]$

Inconvénient : lissage des variations importantes, diminution du pouvoir séparateur

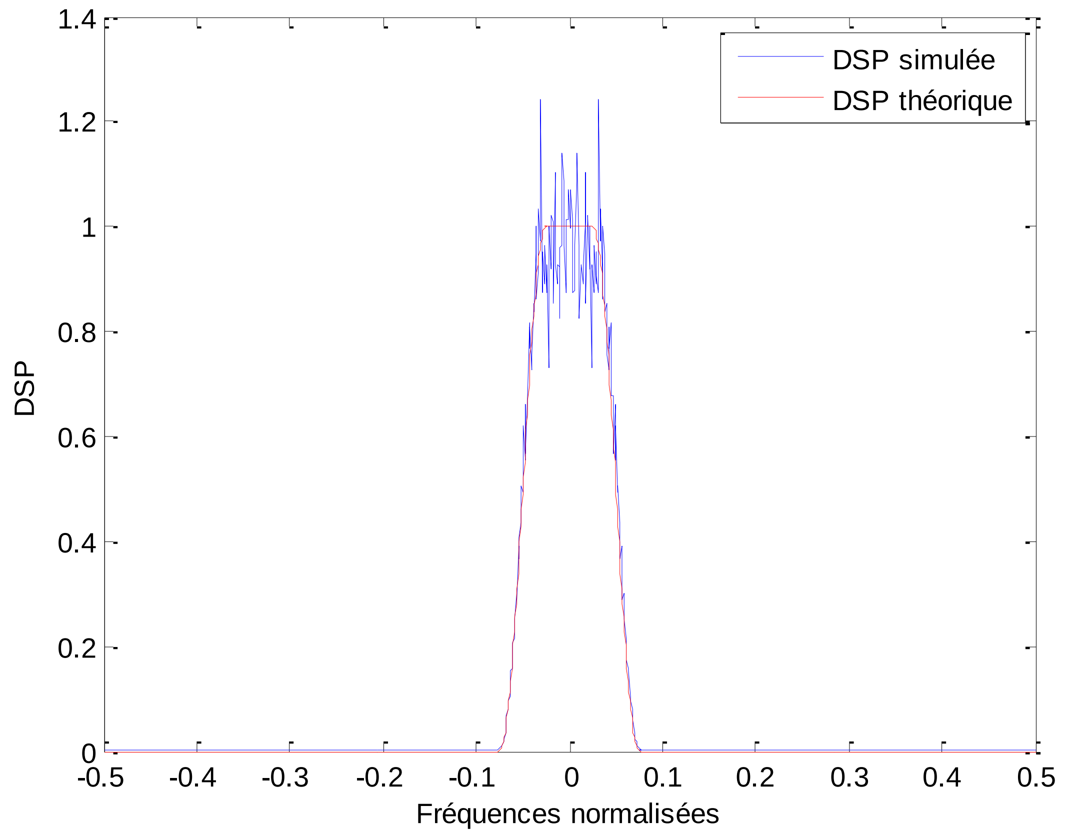
→ Périodogramme de Welch = périodogramme cumulé et modifié

- Fenêtre glissante => M' > M fenêtres de taille L en les autorisant à se recouvrir
- Périodogramme modifié sur chaque tranche de signal

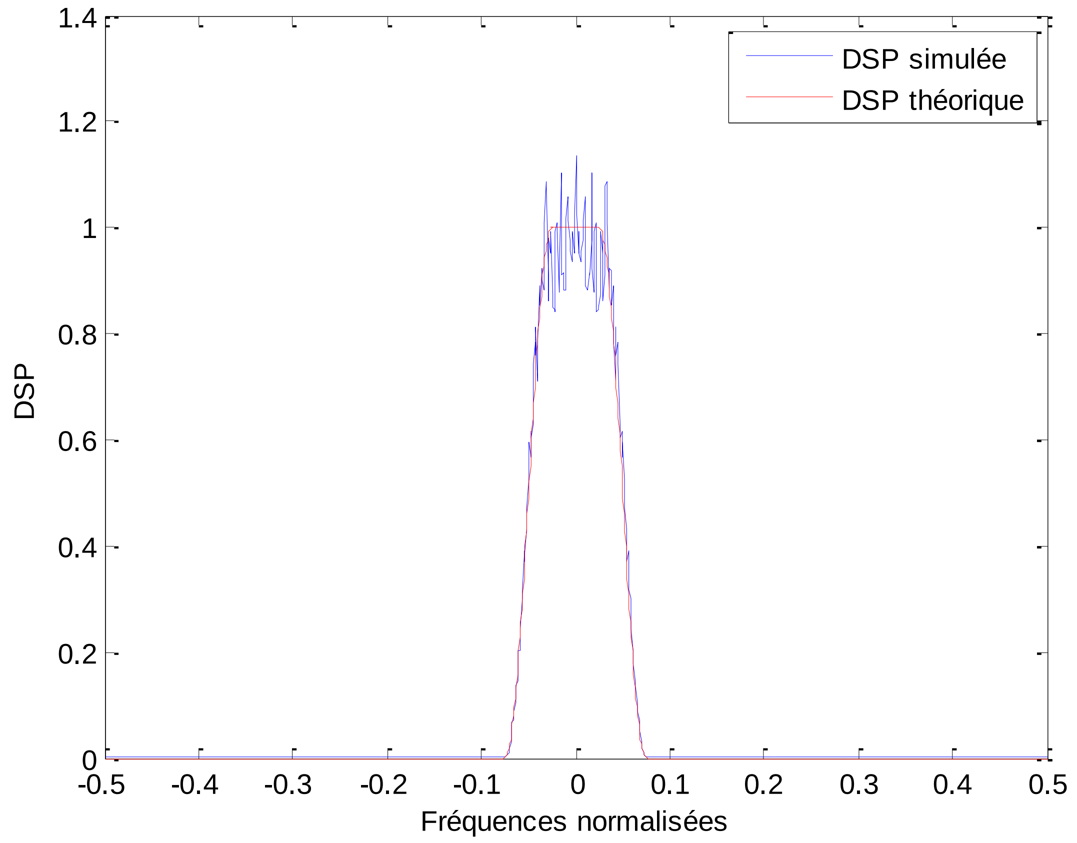
- Exemple :

N= 100 000 échantillons

M=100 fenêtres
de L=1 000 échantillons

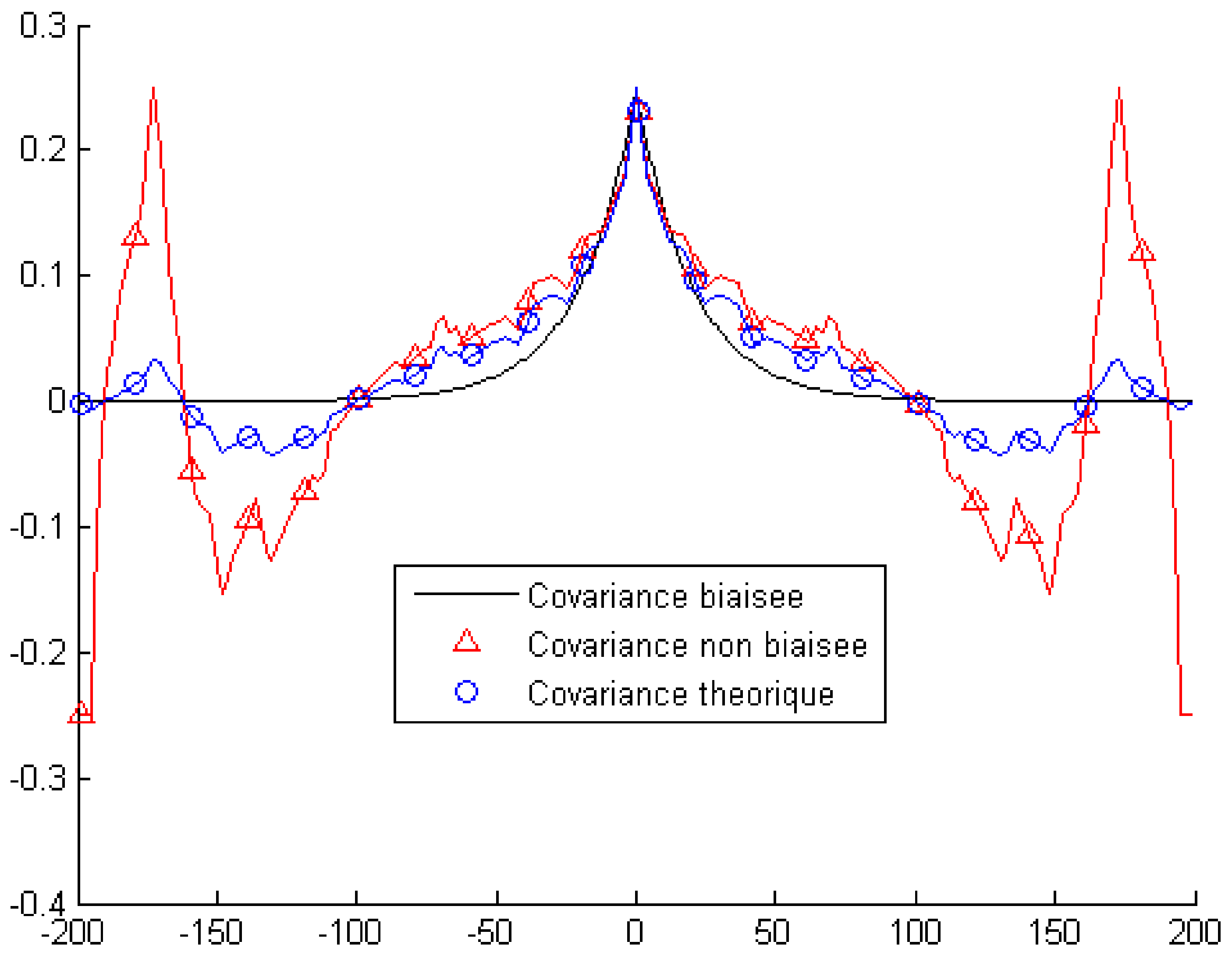


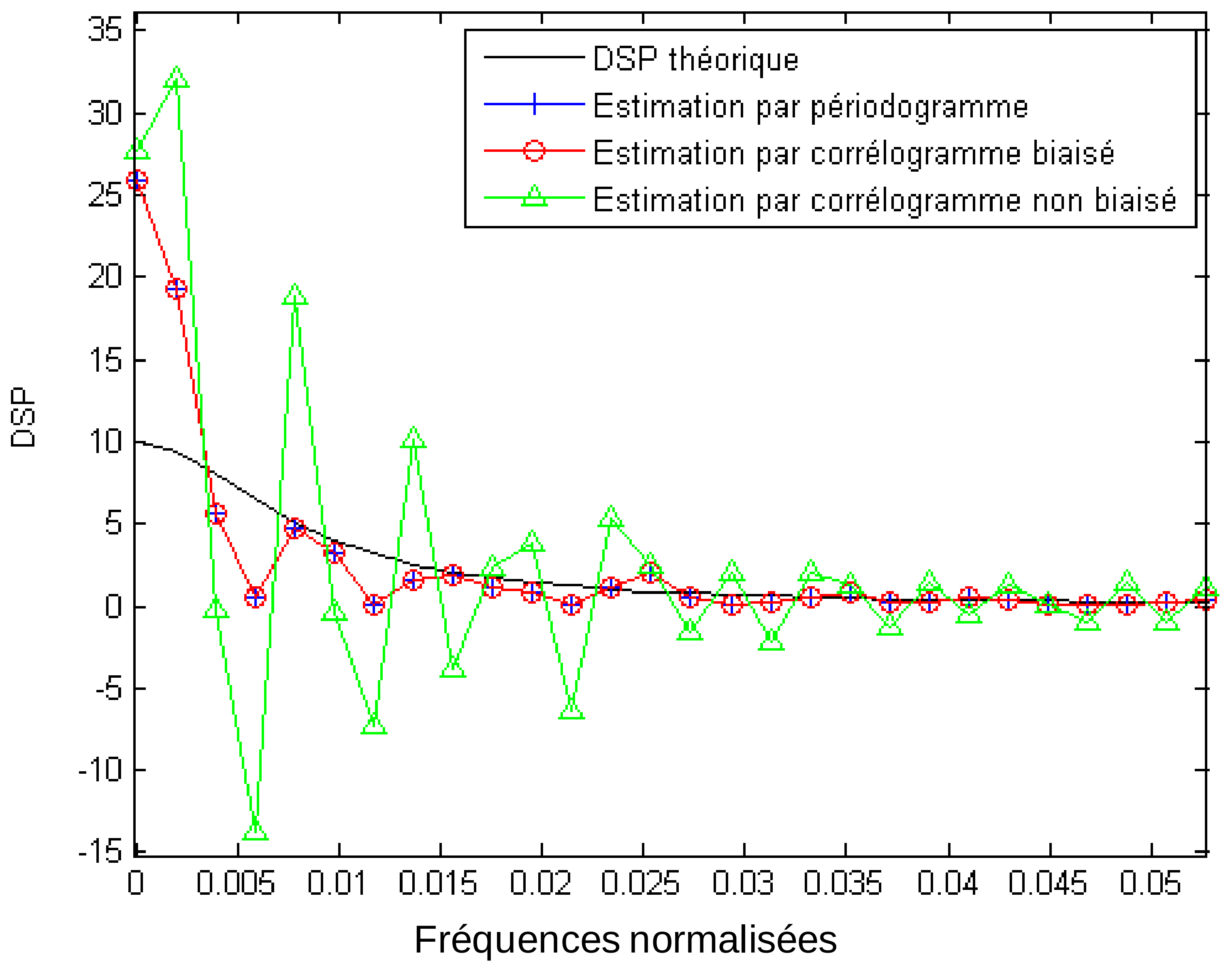
M=100 fenêtres de L=1 000 échantillons
Recouvrement de L/2, fenêtrage de Hamming

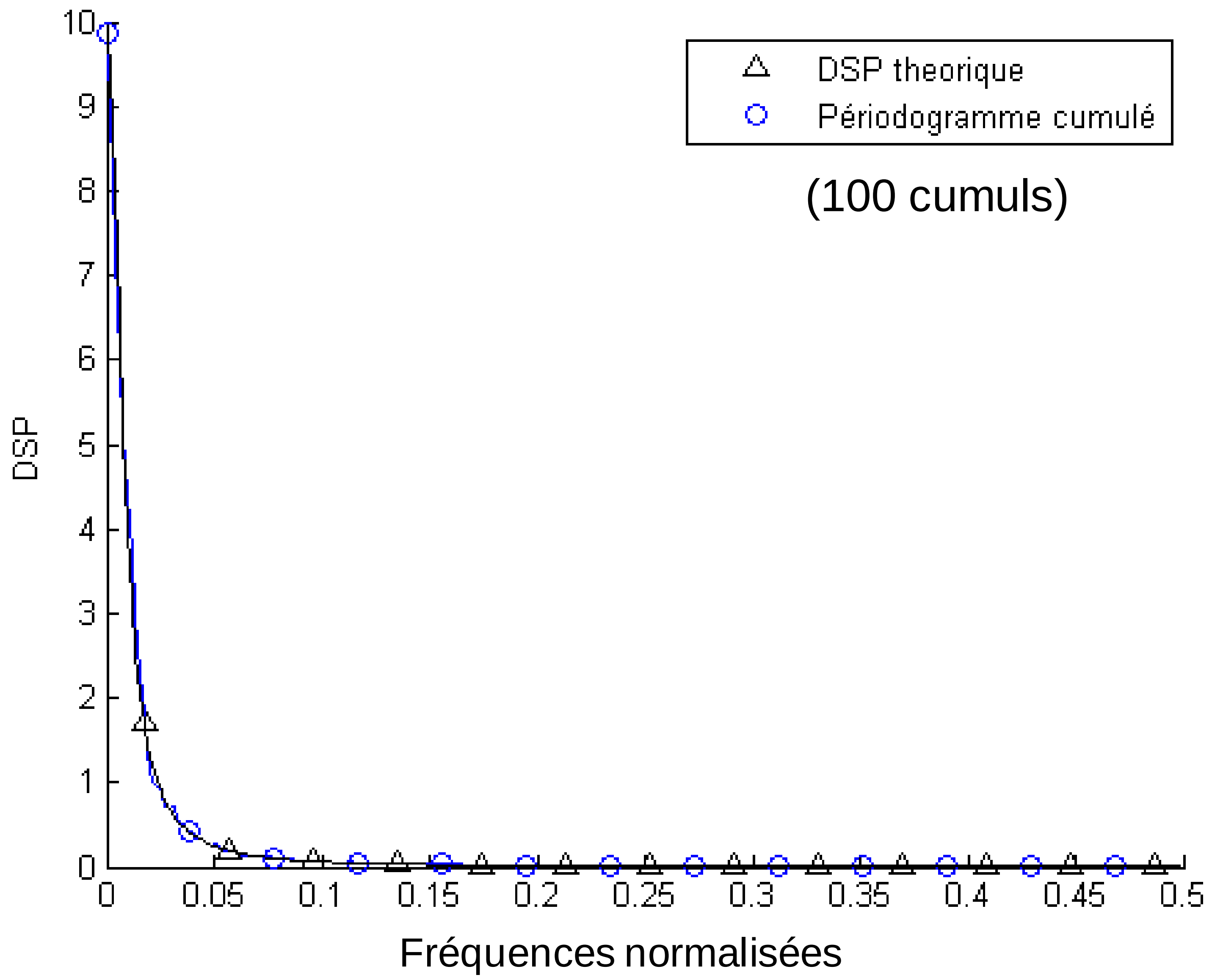


Densité spectrale de puissance

- **Exemple sur une ligne d'image SAR** (Synthetic Aperture Radar)







<u>Remarque :</u> l'estimation de la DSP par périodogramme (ou corrélogramme biaisé) est plus proche de la DSP théorique, ce qui est généralement le cas pour des signaux réels.

Traitement Numérique du signal

1- Signaux numériques
2- Transformée de Fourier Discrète (TFD)
3- Estimation des fonctions d'auto et d'inter corrélation
4- Estimation de la densité spectrale de puissance (DSP)
5- Filtrage numérique linéaire

Filtrage numérique

- **Rappel : la transformée de Laplace**

→ Définition :

$$X(p) = TL\left[x(t)\right] = \int_{0}^{+\infty} x(p)e^{-pt}dt,\ p \in \mathbb{C}$$

→ Outil d'étude des systèmes linéaires invariants dans le temps ANALOGIQUES

$$x(t) \longrightarrow \boxed{h(t) \underset{TL^{-1}}{\overset{TL}{\rightleftarrows}} H(p)} \longrightarrow y(t) = x(t) * h(t) \xrightarrow{TL} Y(p) = H(p)X(p)$$

Fonction de transfert

- Étude temporelle (réponse indicielle, réponse à une rampe): stabilité (pôles de H(p) à partie réelle négative), rapidité (temps de montée, temps de réponse à x%), précision (erreur statique, erreur de trainage)

- Etude fréquentielle (réponse à une entrée sinusoïdale) : diagrammes de Bode => fréquence de coupure, bande passante, atténuation en bande coupée, résonnance …

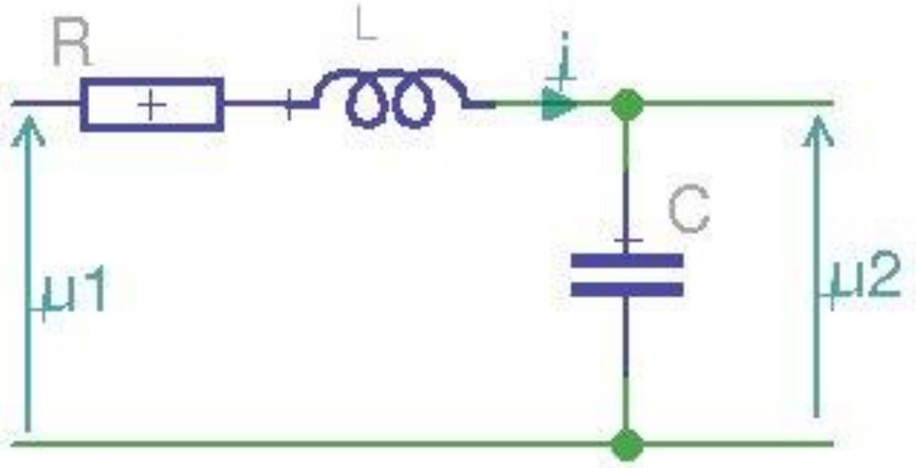
Réponse en fréquence

$$H(f) = \left[H(p)\right]_{p=j\omega=j2\pi f} \quad (TL = TF)$$

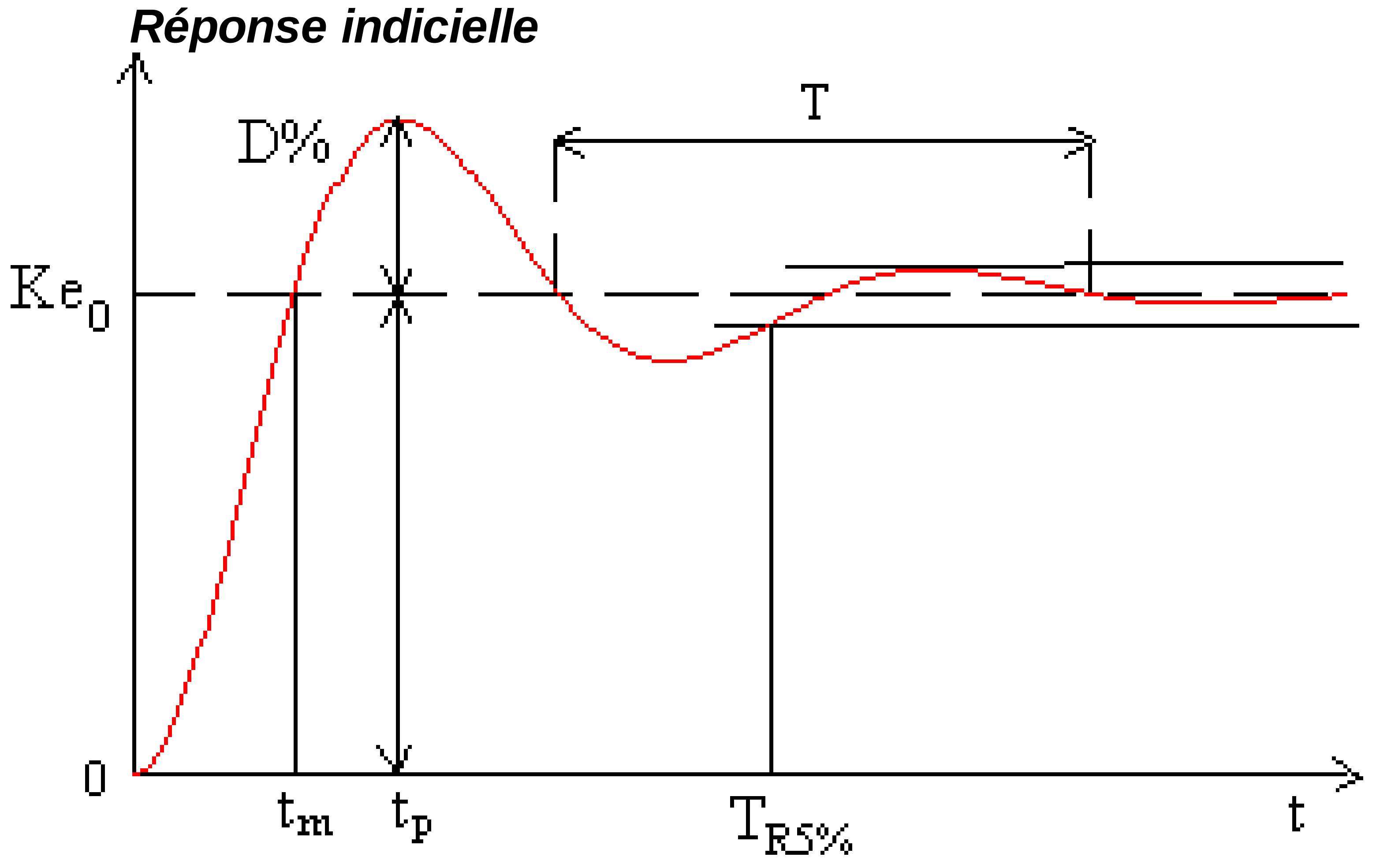
Filtrage numérique

- **Rappel : la transformée de Laplace**

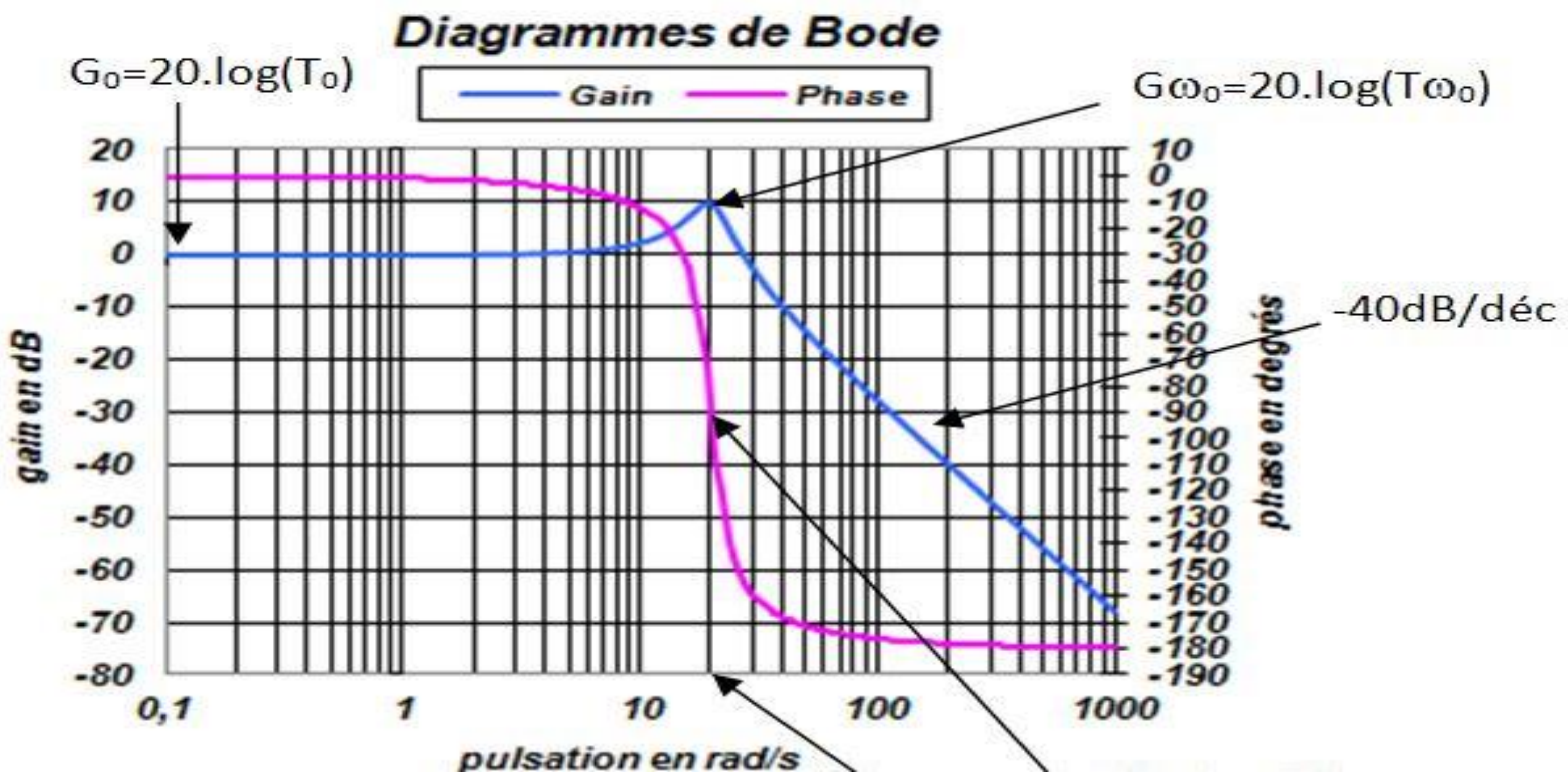
→ <u>Exemple :</u>



Étude temporelle



Étude fréquentielle



Filtrage numérique

- **La transformée en z**

→ <u>Définition :</u>

$$X(z) = TZ\left[x(n)\right] = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}, z \in \mathbb{C}$$

Filtrage numérique

- **La transformée en z**

→ Définition : $$X(z) = TZ\left[x(n)\right] = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}, z \in \mathbb{C}$$

→ Outil d'étude des systèmes linéaires invariants dans le temps NUMERIQUES

$$x(n) \longrightarrow \boxed{h(n) \underset{TZ^{-1}}{\overset{TZ}{\rightleftarrows}} H(z)} \longrightarrow \quad y(n) = x(n) * h(n) \xrightarrow{TZ} Y(z) = \underline{H(z)}X(z)$$

Fonction de transfert

- Étude temporelle : stabilité, rapidité, précision
- Etude fréquentielle :

$$\underline{H(\tilde{f})} = \left[H(z)\right]_{z=j2\pi\tilde{f}} \; (TZ = TFD) \qquad \tilde{f} = \frac{f}{F_e}$$ (fréquence normalisée)

Réponse en fréquence

Filtrage numérique

- **La transformée en z**

→ Définition : $X(z) = TZ[x(n)] = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}, z \in \mathbb{C}$

→ Outil d'étude des systèmes linéaires invariants dans le temps NUMERIQUES

$$x(n) \longrightarrow \boxed{h(n) \underset{TZ^{-1}}{\overset{TZ}{\rightleftarrows}} H(z)} \longrightarrow \quad y(n) = x(n) * h(n) \xrightarrow{TZ} Y(z) = \underline{H(z)} X(z)$$

Fonction de transfert

- Étude temporelle : stabilité, rapidité, précision
- Etude fréquentielle :

$$\underline{H(\tilde{f})} = [H(z)]_{z=j2\pi\tilde{f}} \ (TZ = TFD) \qquad \tilde{f} = \frac{f}{F_e} \quad \text{(fréquence normalisée)}$$

Réponse en fréquence

→ Principales propriétés :

- Linéarité : $TZ[ax(n) + by(n)] = aTZ[x(n)] + bTZ[y(n)]$
- Décalage temporel : $TZ[x(n - n_0)] = z^{-n_0} TZ[x(n)]$
- Produit de convolution : $TZ[x(n) * y(n)] = X(z)Y(z)$

Filtrage numérique

- **La transformée en z**

→ Définition : $$X(z) = TZ\left[x(n)\right] = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}, z \in \mathbb{C}$$

→ Outil d'étude des systèmes linéaires invariants dans le temps NUMERIQUES

$$x(n) \longrightarrow \boxed{h(n) \underset{TZ^{-1}}{\overset{TZ}{\rightleftarrows}} H(z)} \longrightarrow \quad y(n) = x(n) * h(n) \xrightarrow{TZ} Y(z) = \underline{H(z)} X(z)$$

Fonction de transfert

- Étude temporelle : stabilité, rapidité, précision
- Etude fréquentielle :

$$\underline{H(\tilde{f})} = \left[H(z)\right]_{z=j2\pi\tilde{f}} \quad (TZ = TFD)$$

Réponse en fréquence

→ Principales propriétés :

- Linéarité : $TZ\left[ax(n) + by(n)\right] = aTZ\left[x(n\right] + bTZ\left[y(n\right]$
- Décalage temporel : $TZ\left[x(n - n_0)\right] = z^{-n_0} TZ\left[x(n)\right]$
- Produit de convolution : $TZ\left[x(n) * y(n)\right] = X(z)Y(z)$

Voir Poly pour plus de détails (domaine d'existence, TZ inverse …)

Filtrage numérique

- **Filtres linéaires invariants dans le temps**

→ <u>Linéarité :</u>

$$\left.\begin{array}{l} x_1(n) \longrightarrow \boxed{\textit{Filtre}} \longrightarrow y_1(n) \\ x_2(n) \longrightarrow \boxed{\textit{Filtre}} \longrightarrow y_2(n) \end{array}\right\} \; x_1(n) + x_2(n) \longrightarrow \boxed{\textit{Filtre}} \longrightarrow y_1(n) + y_2(n)$$

→ <u>Invariance dans le temps :</u>

$$x(n) \longrightarrow \boxed{\textit{Filtre}} \longrightarrow y(n) \qquad x(n-n_0) \longrightarrow \boxed{\textit{Filtre}} \longrightarrow y(n-n_0)$$

Filtrage numérique

- **Réponse impulsionnelle et fonction de transfert**

Impulsion unité
(Impulsion de Kronecker)

$$\begin{aligned}\delta(n) &= 1 \ pour \ n = 0 \\ &= 0 \ pour \ n \neq 0\end{aligned} \longrightarrow \boxed{\textit{Filtre}} \longrightarrow h(n)$$

Réponse
impulsionnelle

Filtrage numérique

- **Réponse impulsionnelle et fonction de transfert**

Impulsion unité
(Impulsion de Kronecker)

$$\delta(n) = 1 \; pour \; n = 0$$
$$= 0 \; pour \; n \neq 0$$

$\longrightarrow$ Filtre $\longrightarrow$ $h(n)$ Réponse impulsionnelle

$$h(n) \xrightarrow{TZ} H(z)$$ Fonction de transfert

Filtrage numérique

- **Réponse impulsionnelle et fonction de transfert**

Impulsion unité
(Impulsion de Kronecker)

$$\delta(n) = 1 \; pour \; n = 0$$
$$= 0 \; pour \; n \neq 0$$

$$\longrightarrow \boxed{\textit{Filtre}} \longrightarrow h(n)$$

Réponse impulsionnelle

$$h(n) \xrightarrow{TZ} H(z)$$

Fonction de transfert

$$x(n) \longrightarrow \boxed{h(n) \underset{TZ^{-1}}{\overset{TZ}{\rightleftarrows}} H(z)} \longrightarrow y(n) = x(n) * h(n) = \sum_{n=-\infty}^{+\infty} h(k)x(n-k) \xrightarrow{TZ} Y(z) = H(z)X(z) \quad \Rightarrow H(z) = \frac{Y(z)}{X(z)}$$

Filtrage numérique

- **Réponse impulsionnelle et fonction de transfert**

Impulsion unité
(Impulsion de Kronecker)

$$\delta(n) = 1 \; pour \; n = 0$$
$$= 0 \; pour \; n \neq 0$$

$$\longrightarrow \boxed{Filtre} \longrightarrow h(n)$$

Réponse impulsionnelle

$$h(n) \xrightarrow{TZ} H(z)$$

Fonction de transfert

$$x(n) \longrightarrow \boxed{h(n) \underset{TZ^{-1}}{\overset{TZ}{\rightleftarrows}} H(z)} \longrightarrow y(n) = x(n) * h(n) = \sum_{n=-\infty}^{+\infty} h(k)x(n-k) \xrightarrow{TZ} Y(z) = H(z)X(z) \quad \Rightarrow H(z) = \frac{Y(z)}{X(z)}$$

- **Réponse en fréquence et temps de propagation de groupe**

$\rightarrow$ Réponse en fréquence (ou réponse harmonique) :

$$H(\tilde{f}) = [H(z)]_{z=j2\pi\tilde{f}} \; (TZ = TFD)$$

$$\tilde{f} = \frac{f}{F_e}$$ (fréquence normalisée)

$\rightarrow$ Temps de propagation de groupe (TPG) :

$$TPG(\tilde{f}) = -\frac{1}{2\pi}\frac{d\Phi_H(\tilde{f})}{d\tilde{f}}$$

Remarque : $\left|H(\tilde{f})\right|^2 = \left[H(z)\,H(z^{-1})\right]_{z=e^{j2\pi\tilde{f}}}$

Filtrage numérique

- **Conditions de réalisabilité (données sur la réponse impulsionnelle)**

→ Causalité :

$$h(n) = 0 \text{ pour } n < 0$$

→ Stabilité (entrée bornée => sortie bornée) :

$$\sum_{n=-\infty}^{+\infty} |h(n)| < \infty$$

→ Réponse impulsionnelle réelle

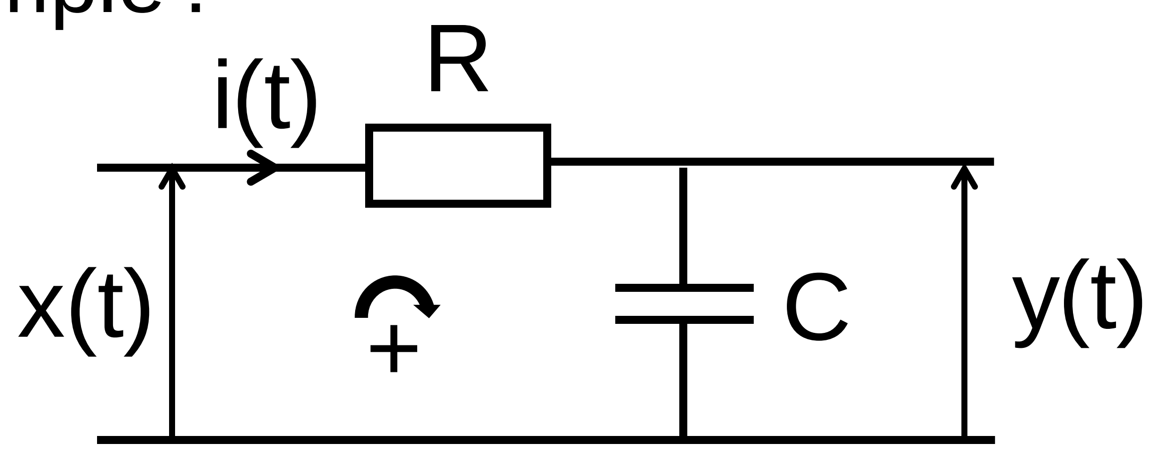
$$h(n) \in \mathbb{R}$$

Filtrage numérique

- **Filtres numériques RATIONNELS**

→ <u>Définis par analogie avec les filtres analogiques :</u>

Exemple :



$$y(t) = x(t) - RC\frac{dy(t)}{dt}.$$ Équation différentielle en temporel

TL

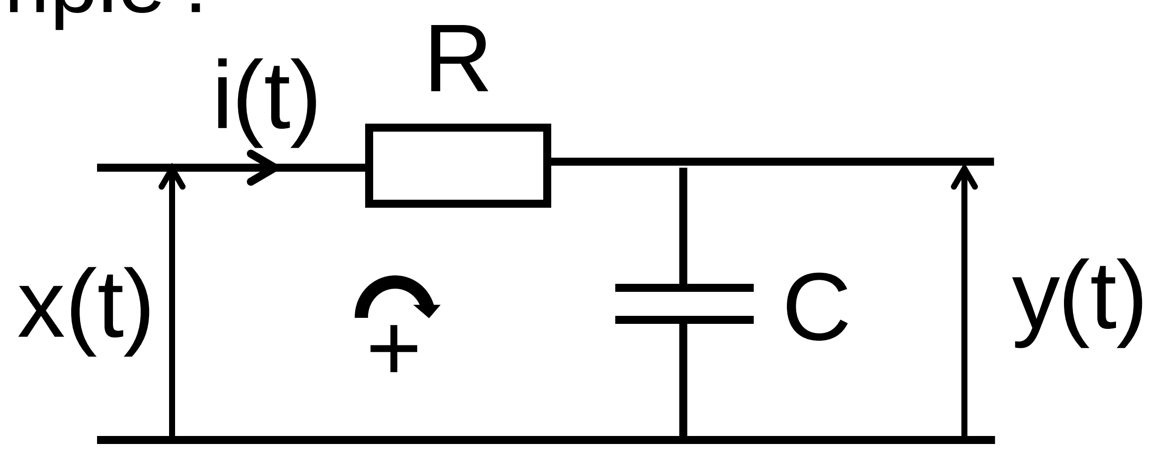
$$H(p) = \frac{Y(p)}{X(p)} = \frac{1}{1+RCp}$$ Fonction de transfert rationnelle en p

Filtrage numérique

- **Filtres numériques RATIONNELS**

→ <u>Définis par analogie avec les filtres analogiques :</u>

Exemple :



$$y(t) = x(t) - RC\frac{dy(t)}{dt}.$$

Équation différentielle en temporel

TL

$$H(p) = \frac{Y(p)}{X(p)} = \frac{1}{1+RCp}$$

Fonction de transfert rationnelle en p

$$\boxed{H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}}}$$

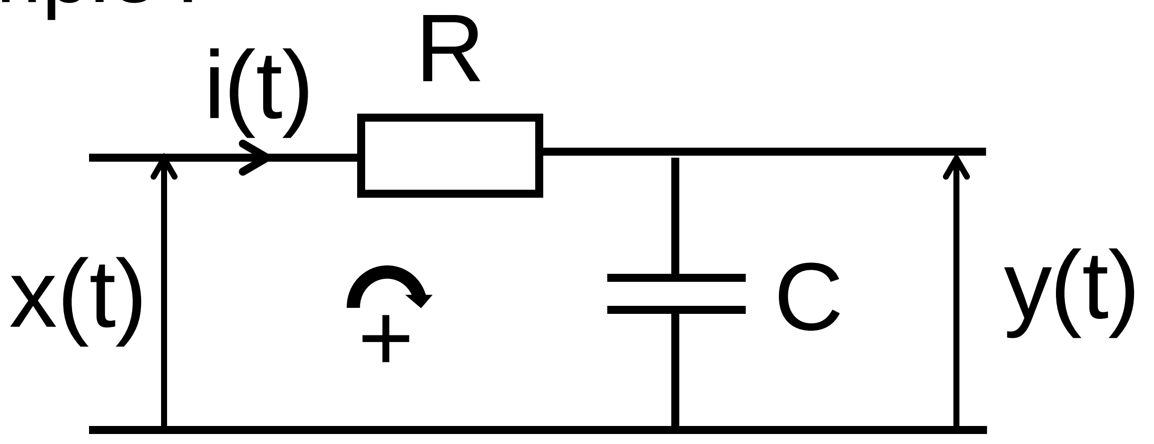
Fonction de transfert RATIONNELLE en z

Filtrage numérique

- **Filtres numériques RATIONNELS**

→ <u>Définis par analogie avec les filtres analogiques :</u>

Exemple :



$$y(t) = x(t) - RC\frac{dy(t)}{dt}$$ Équation différentielle en temporel

TL

$$H(p) = \frac{Y(p)}{X(p)} = \frac{1}{1+RCp}$$ Fonction de transfert rationnelle en p

$$\boxed{H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}}}$$ Fonction de transfert RATIONNELLE en z

TZ^{-1}

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$ Équation récurrente en temporel

(Remarque : on impose $a_0 = 1$)

Filtrage numérique

- **Filtres numériques RATIONNELS**

→ Définition :

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

Fonction de transfert RATIONNELLE en z

TZ^{-1}

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

Équation récurrente en temporel

(Remarque : on impose $a_0 = 1$)

Filtrage numérique

- **Filtres numériques RATIONNELS**

→ Définition :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}}$$ Fonction de transfert RATIONNELLE en z

TZ^{-1}

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$ Équation récurrente en temporel

(Remarque : on impose $a_0 = 1$)

Définis par deux jeux de coefficients

$$(a_0 = 1) y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

Filtrage numérique

- **Filtres numériques RATIONNELS**

→ Définition :

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

Fonction de transfert RATIONNELLE en z

TZ^{-1}

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

Équation récurrente en temporel

(Remarque : on impose $a_0 = 1$)

Définis par deux jeux de coefficients

$$(a_0 = 1)y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

```
%filtrage sous Matlab
y=filter(B,A,x);
```

Signal filtré

Signal à filtrer

Définition du filtre à utiliser (deux tableaux de coefficients)

Filtrage numérique

- **Filtres numériques RATIONNELS**

$\longrightarrow$ Définition :

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

Fonction de transfert RATIONNELLE en z

TZ^{-1}

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

Équation récurrente en temporel

(Remarque : on impose $a_0 = 1$)

Définis par deux jeux de coefficients

$$(a_0 = 1) y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

```
%filtrage sous Matlab
y=filter(B,A,x);
```

Signal filtré

Signal à filtrer

Définition du filtre à utiliser (deux tableaux de coefficients)

Système bouclé

$$y(n) = \boxed{-\sum_{k=1}^{M-1} a_k y(n-k)} + \sum_{k=0}^{N-1} b_k x(n-k)$$

Filtrage numérique

- **Filtres numériques RATIONNELS**

→ <u>Définition :</u>

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

Fonction de transfert RATIONNELLE en z

TZ^{-1}

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

Équation récurrente en temporel

(Remarque : on impose $a_0 = 1$)

Définis par deux jeux de coefficients

$$(a_0 = 1)\, y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

```
%filtrage sous Matlab
y=filter(B,A,x);
```

Signal filtré

Signal à filtrer

Définition du filtre à utiliser (deux tableaux de coefficients)

Système bouclé

$$y(n) = \boxed{-\sum_{k=1}^{M-1} a_k y(n-k)} + \sum_{k=0}^{N-1} b_k x(n-k)$$

$$\Rightarrow \quad h(n) = -\sum_{k=1}^{M-1} a_k h(n-k), \text{ pour } n \geq N$$

Filtres de type RII = à Réponse Impulsionnelle Infinie

Filtrage numérique

- **Stabilité des filtres numériques RATIONNELS**

$\longrightarrow$ Condition sur les pôles de la fonction de transfert (démo dans le poly) :

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}} = \sum_{k=0}^{M-1} \frac{A_k}{1 - p_k z^{-1}}$$

(hypothèse : N<M)

TZ^{-1}

$$h(n) = \sum_{k=0}^{M-1} A_k p_k^n u(n)$$ (solution causale[1])

Condition de stabilité des filtres numériques rationnels

$$\boxed{\sum_{n=-\infty}^{+\infty} |h(n)| < \infty \quad \text{si } |p_k| < 1 \; \forall k}$$

Filtrage numérique

- **Stabilité des filtres numériques RATIONNELS**

$\longrightarrow$ Condition sur les pôles de la fonction de transfert (démo dans le poly) :

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}} = \sum_{k=0}^{M-1} \frac{A_k}{1 - p_k z^{-1}}$$

(hypothèse : N<M)

TZ^{-1}

$$h(n) = \sum_{k=0}^{M-1} A_k p_k^n u(n)$$ (solution causale[(1)])

Condition de stabilité des filtres numériques rationnels

$$\boxed{\sum_{n=-\infty}^{+\infty} |h(n)| < \infty \quad \text{si } |p_k| < 1 \; \forall k}$$

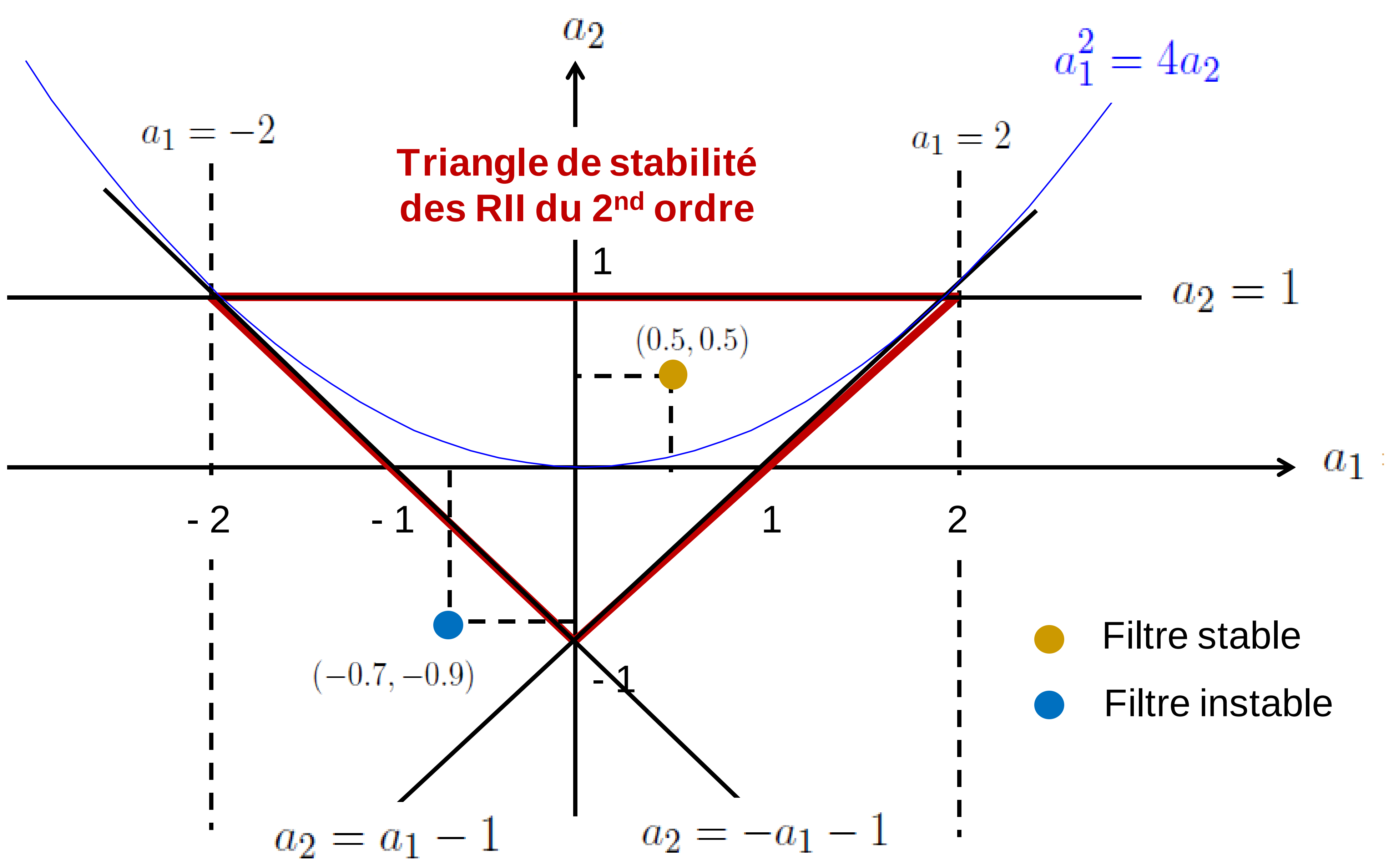
$\longrightarrow$ Condition sur les coefficients (a_k, b_k) (voir exercice 4 dans le poly) :

- Filtres du 1ier ordre $\quad H(z) = \dfrac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}$

$\boxed{|a_1| < 1}$ **Condition de stabilité des RII du 1ier ordre**

- Filtres du 2nd ordre $\quad H(z) = \dfrac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$

$\boxed{(a_1, a_2) \in \text{ triangle de stabilité}}$ **Condition de stabilité des RII du 2nd ordre**



(1)) La TZ inverse n'est pas unique. Elle sera différente selon le contour choisi dans le domaine de convergence pour la calculer : voir exercices poly sur la TZ.

Filtrage numérique

- **Filtres numériques RATIONNELS à réponse impulsionnelle finie (type RIF)**

→ Définition :

$$y(n) = \cancel{\boxed{-\sum_{k=1}^{M-1} a_k y(n-k)}} + \sum_{k=0}^{N-1} b_k x(n-k) \quad \xrightarrow{TZ} \quad H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{N-1} b_k z^{-k}$$

Équation récurrente en temporel <u>sans boucle de réaction</u>

Filtrage numérique

- **Filtres numériques RATIONNELS à réponse impulsionnelle finie (type RIF)**

→ Définition :

$$y(n) = \cancel{\boxed{-\sum_{k=1}^{M-1} a_k y(n-k)}} + \sum_{k=0}^{N-1} b_k x(n-k) \quad \xrightarrow{TZ} \quad H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{N-1} b_k z^{-k}$$

Équation récurrente en temporel sans boucle de réaction

Définis par un seul jeu de coefficients

```
%filtrage sous Matlab
y=filter(B,1,x);
```

Signal filtré

Définition du filtre (RIF) à utiliser

Signal à filtrer

```
y=conv(x,B,'same');
```

'same' returns the central part of the convolution that is the same size as x.

Filtrage numérique

- **Filtres numériques RATIONNELS à réponse impulsionnelle finie (type RIF)**

→ <u>Définition :</u>

$$y(n) = \boxed{\cancel{-\sum_{k=1}^{M-1} a_k y(n-k)}} + \sum_{k=0}^{N-1} b_k x(n-k) \quad \xrightarrow{TZ} \quad H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{N-1} b_k z^{-k}$$

Équation récurrente en temporel <u>sans boucle de réaction</u>

Définis par un seul jeu de coefficients

```
%filtrage sous Matlab
y=filter(B,1,x);
```

Signal filtré

Définition du filtre (RIF) à utiliser

Signal à filtrer

```
y=conv(x,B,'same');
```

'same' returns the central part of the convolution that is the same size as x.

Système non bouclé

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k) = \sum_{k=0}^{N-1} h(k) x(n-k)$$

- Filtres non récursifs
- Inconditionnellement stables
- Remarque : $b_k = h(k)$ mais pour un filtre rendu causal (voir plus loin)

Filtrage numérique

- **Synthèse et implantation des filtres numériques rationnels**

Spécifications à respecter :

GABARIT

Filtrage numérique

- **Synthèse et implantation des filtres numériques rationnels**

Spécifications à respecter :

GABARIT

? COEFFICIENTS ?

$$\{a_k\}_{k=0,\ldots,M-1}$$

$$\{b_k\}_{k=0,\ldots,N-1}$$

Définissant un filtre respectant le gabarit fixé

Filtrage numérique

- **Synthèse et implantation des filtres numériques rationnels**

Spécifications à respecter :

GABARIT

↓

SYNTHESE

↓

COEFFICIENTS

$$\{a_k\}_{k=0,\ldots,M-1}$$

$$\{b_k\}_{k=0,\ldots,N-1}$$

Définissant un filtre respectant le gabarit fixé

Filtrage numérique

- **Synthèse et implantation des filtres numériques rationnels**

Spécifications à respecter :

GABARIT

↓

SYNTHESE

↓

COEFFICIENTS

$$\{a_k\}_{k=0,\ldots,M-1}$$
$$\{b_k\}_{k=0,\ldots,N-1}$$

Définissant un filtre respectant le gabarit fixé

IMPLANTATION sous Matlab

Tableaux de coefficients définissant le filtre à utiliser

y=filter(B,A,x);

Signal filtré

Signal à filtrer

Remarque : A=[1] pour les filtres RIF

IMPLANTATION en temps réel :

Filtre RII

$$x(n) \longrightarrow \boxed{\{a_k\}_{k=0,\ldots,M-1} \quad \{b_k\}_{k=0,\ldots,N-1}} \longrightarrow y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$

Filtre RIF

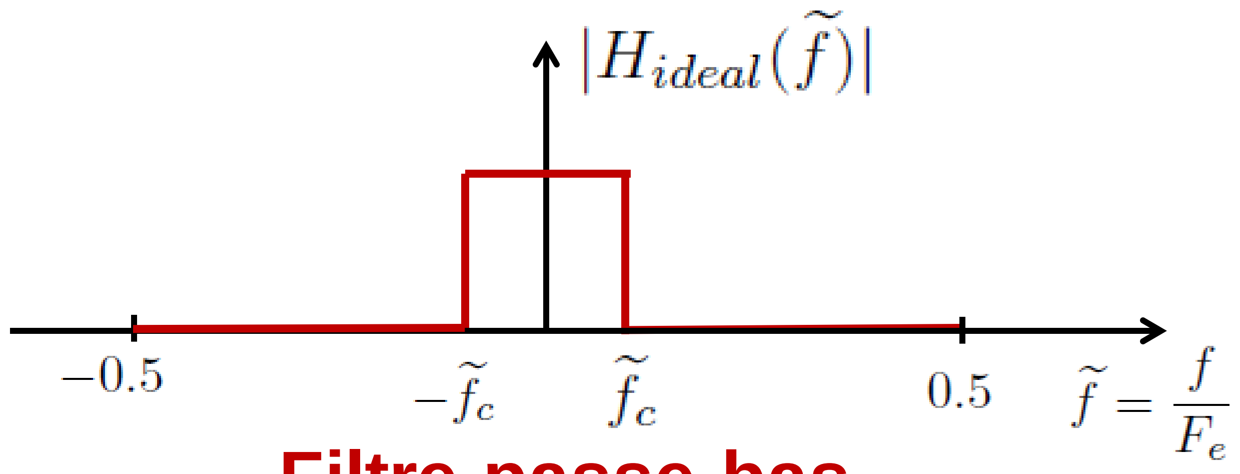
$$x(n) \longrightarrow \boxed{\{b_k\}_{k=0,\ldots,N-1}} \longrightarrow y(n) = \sum_{k=0}^{N-1} b_k x(n-k)$$

À réaliser en T_e secondes (période d'échantillonnage)

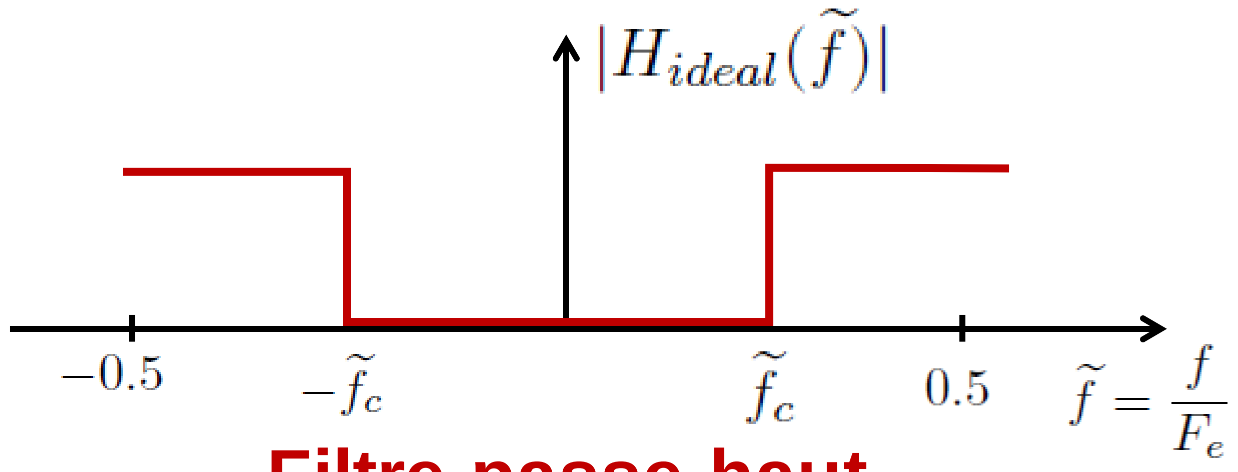
Filtrage numérique

- **Synthèse et implantation des filtres numériques rationnels**

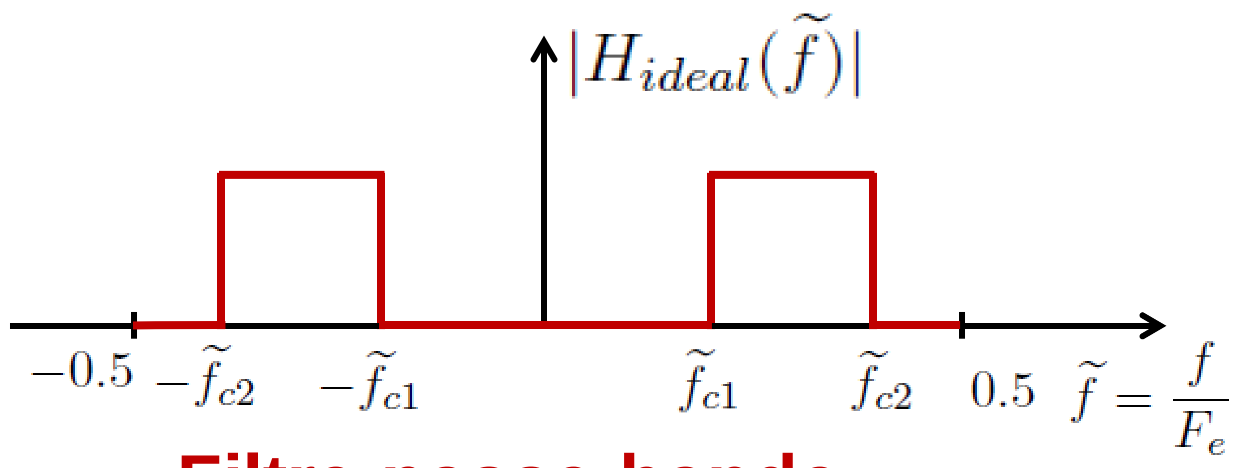
  $\rightarrow$ Réponses en fréquence cibles idéales



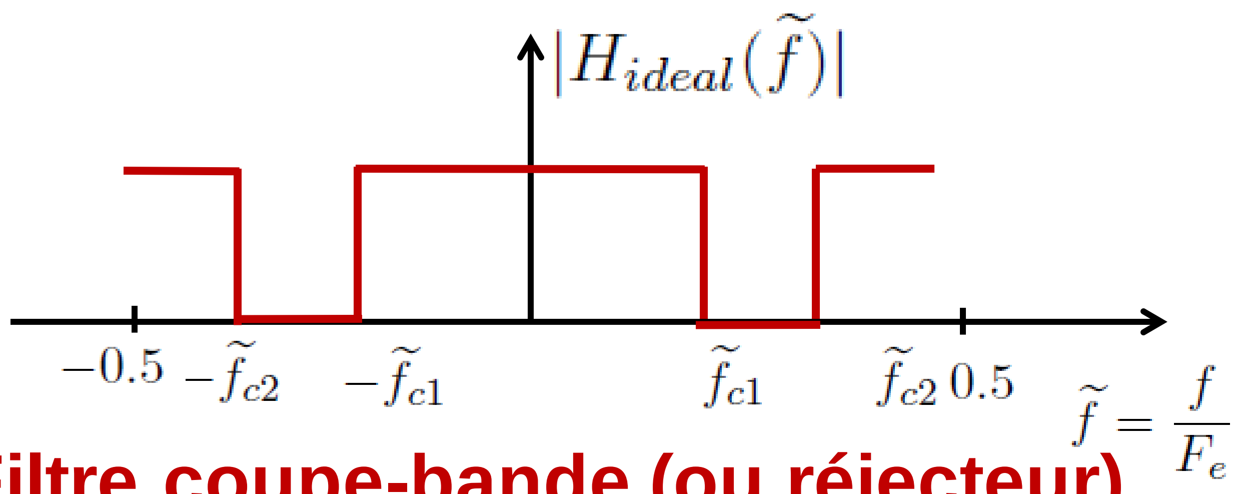
Filtre passe-bas



Filtre passe-haut



Filtre passe-bande



Filtre coupe-bande (ou réjecteur)

Filtrage numérique

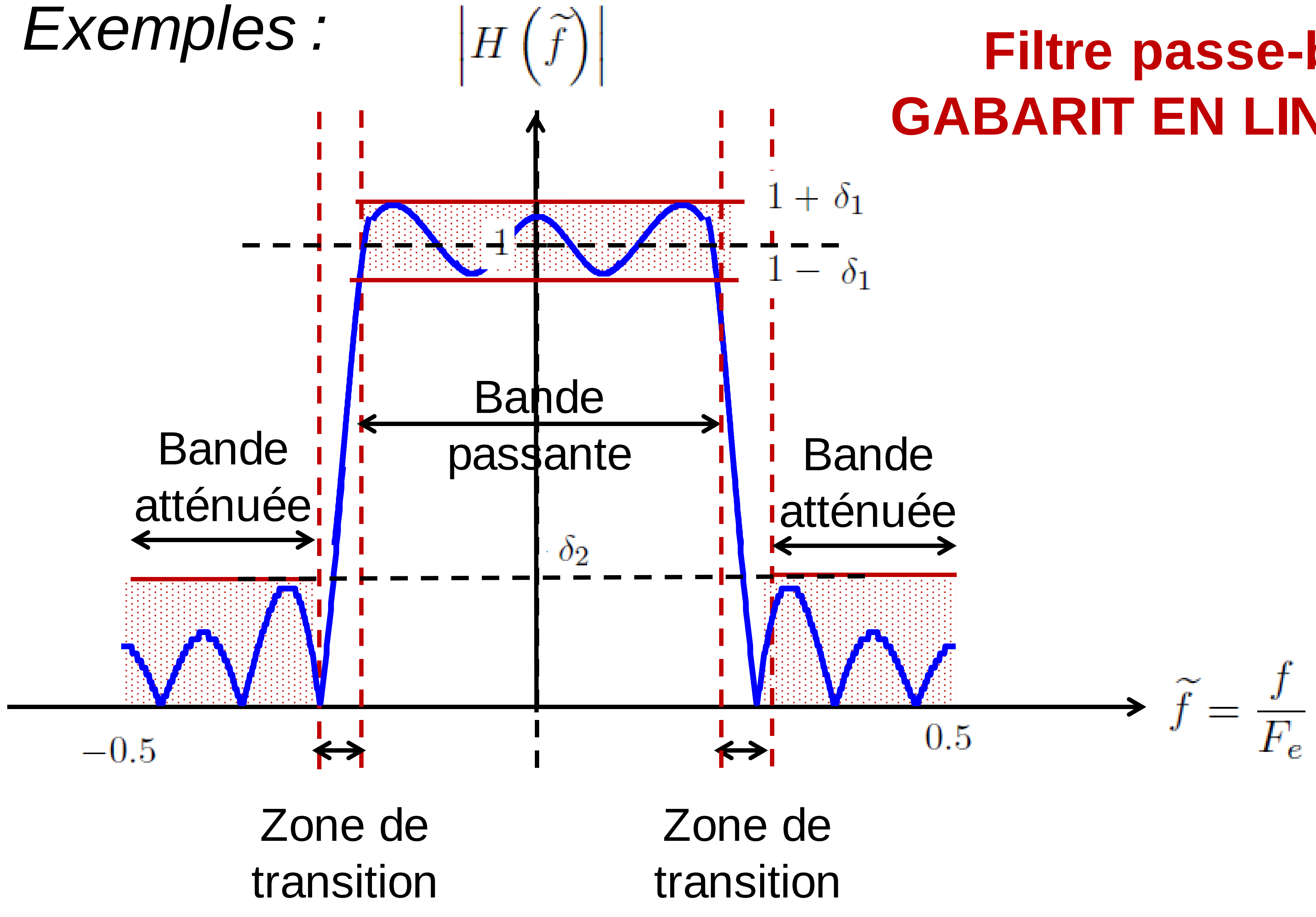
- **Synthèse et implantation des filtres numériques rationnels**

→ Notion de gabarit à respecter

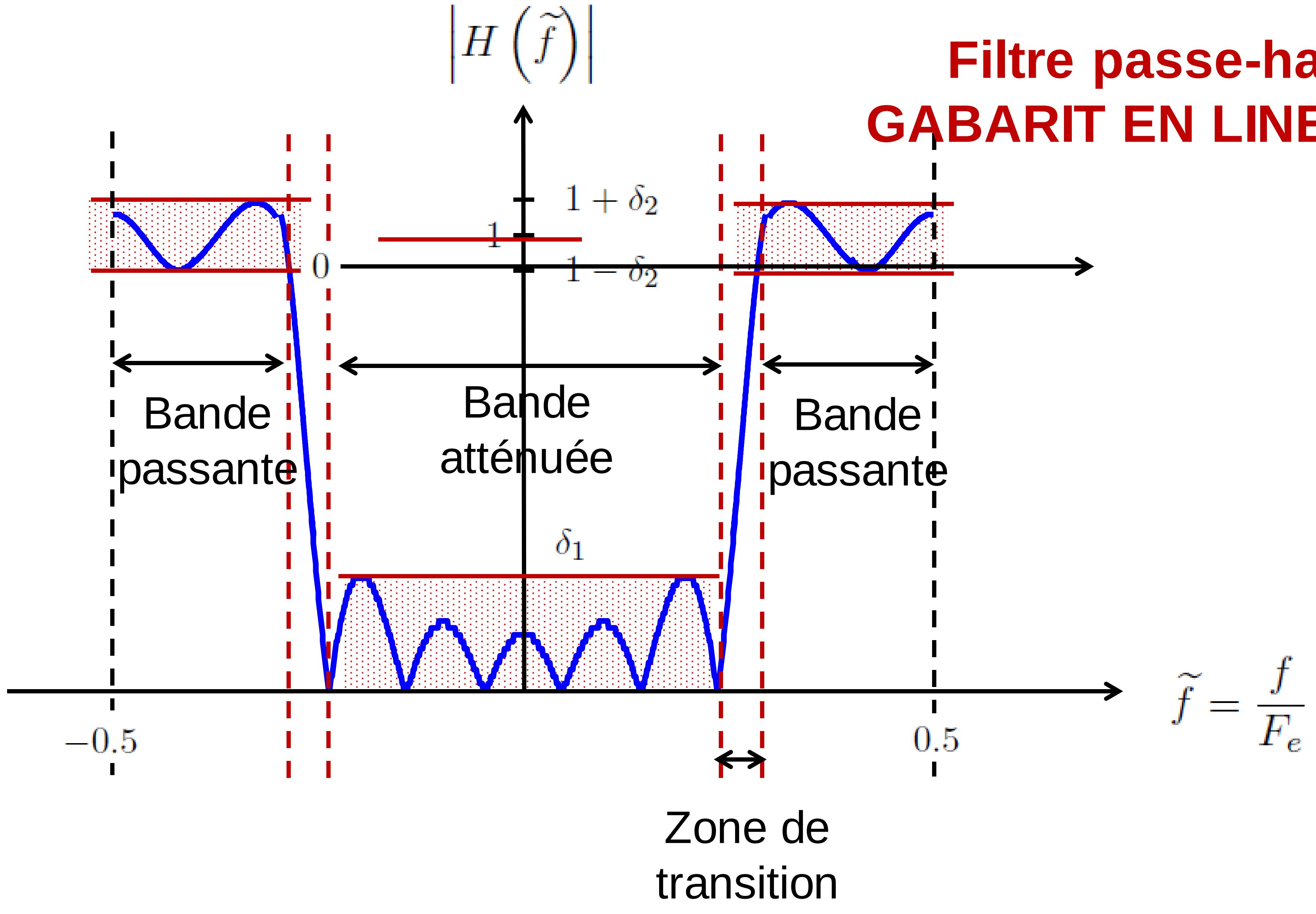
- Sur le module de la réponse en fréquence

Exemples :

Filtre passe-bas
GABARIT EN LINEAIRE



Filtre passe-haut
GABARIT EN LINEAIRE



- Sur la phase de la réponse en fréquence : filtres de phase ou passe-tout

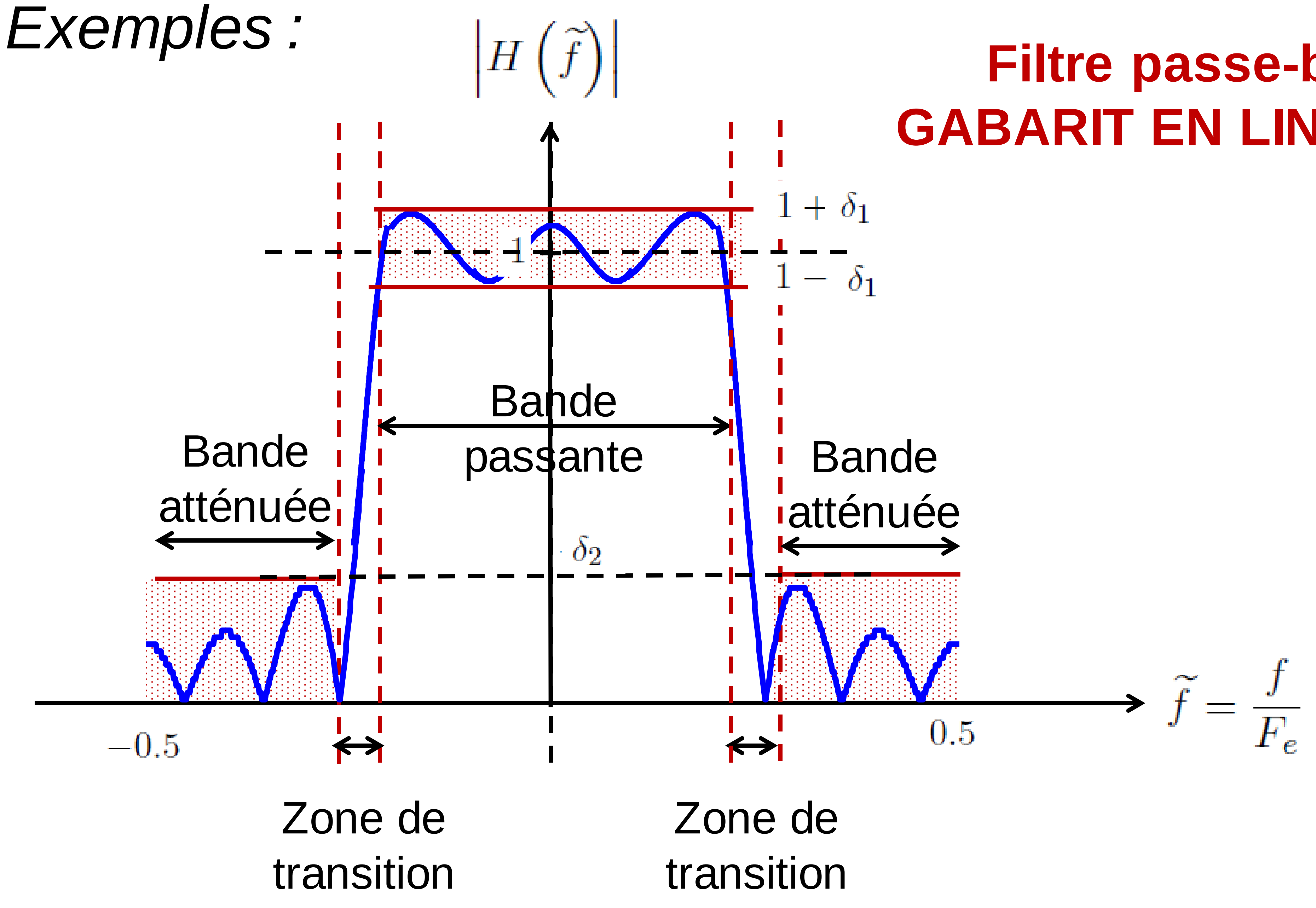
Filtrage numérique

- **Synthèse et implantation des filtres numériques rationnels**

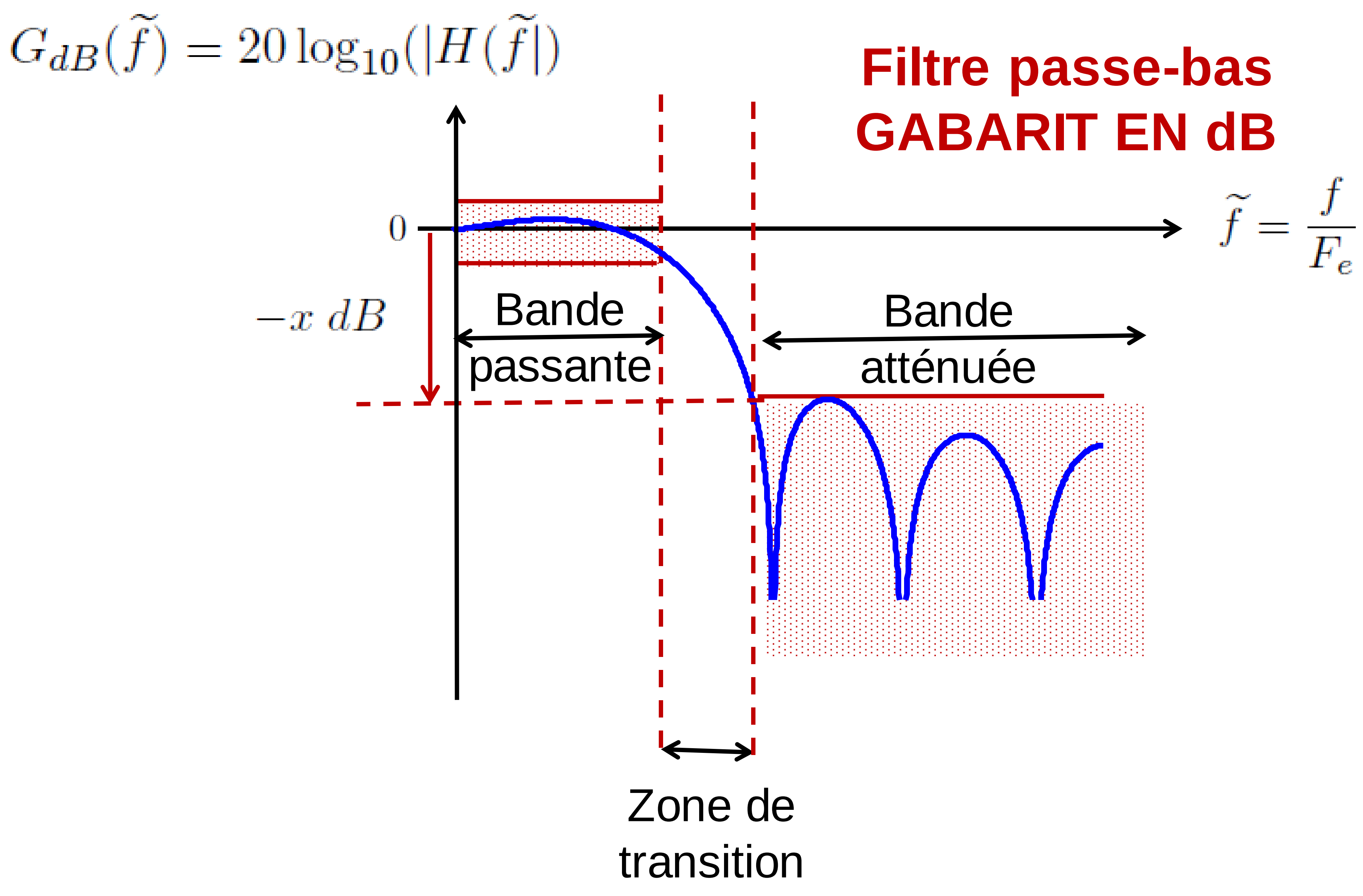
→ Notion de gabarit à respecter

- Sur le module de la réponse en fréquence

Exemples :



Filtre passe-bas
GABARIT EN LINEAIRE



Filtre passe-bas
GABARIT EN dB

- Sur la phase de la réponse en fréquence : filtres de phase ou passe-tout

Filtrage numérique

- **Synthèse et implantation des filtres numériques rationnels**

Spécifications à respecter :

GABARIT

↓

SYNTHESE

→ Synthèse de **RIF**

→ Synthèse de **RII**

Deux méthodes différentes

↓

COEFFICIENTS

$\{a_k\}_{k=0,\dots,M-1}$

$\{b_k\}_{k=0,\dots,N-1}$

Le filtre qui respecte le gabarit n'est pas unique

← **d'un filtre** respectant le gabarit fixé →

IMPLANTATION sous Matlab

Tableaux de coefficients définissant le filtre à utiliser

y=filter(B,A,x);

Signal filtré

Signal à filtrer

Remarque : A=[1] pour les filtres RIF

IMPLANTATION en temps réel :

$x(n) \longrightarrow$ **Filtre RII** $\{a_k\}_{k=0,\dots,M-1}$ $\{b_k\}_{k=0,\dots,N-1}$ $\longrightarrow y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$

$x(n) \longrightarrow$ **Filtre RIF** $\{b_k\}_{k=0,\dots,N-1}$ $\longrightarrow y(n) = \sum_{k=0}^{N-1} b_k x(n-k)$

À réaliser en T_e secondes (période d'échantillonnage)

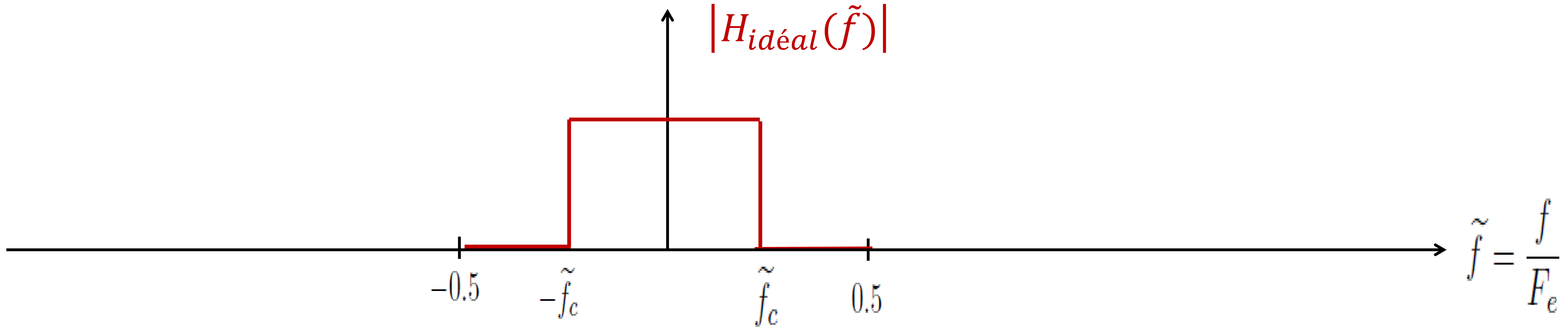
Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

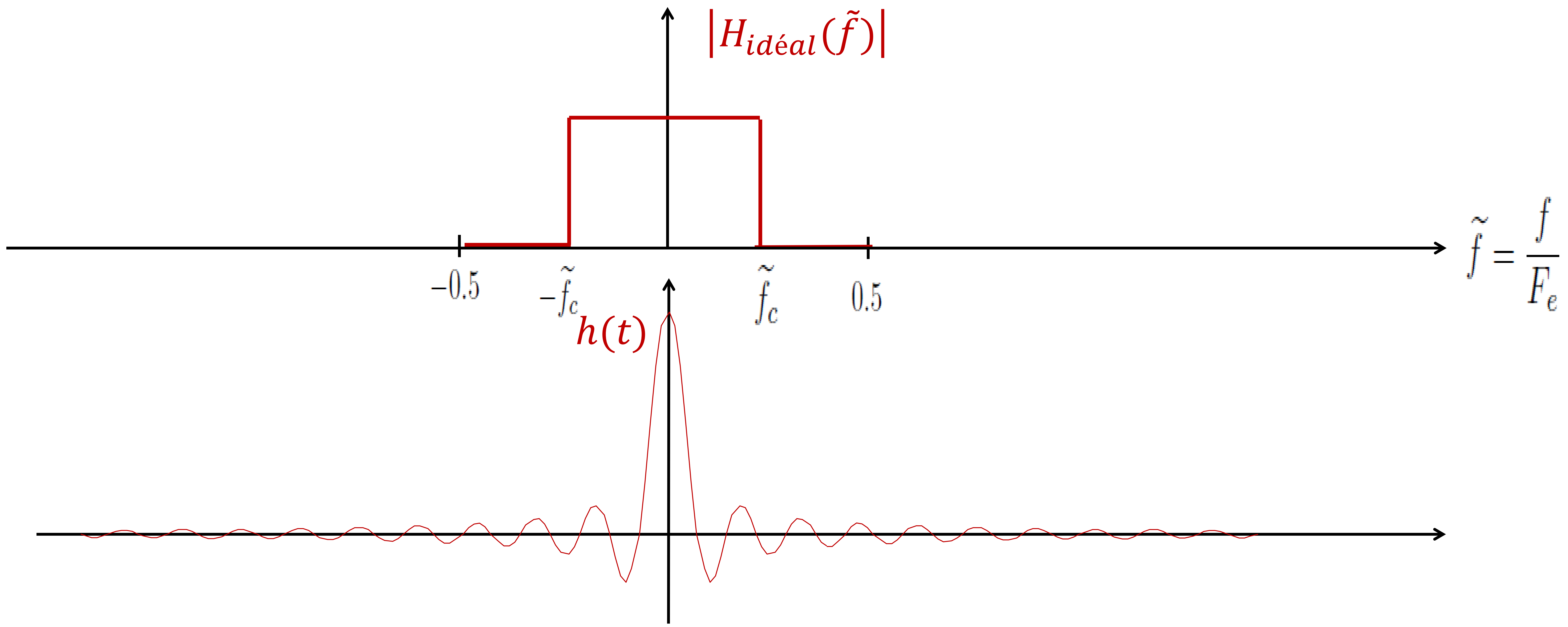
→ Exemple : synthèse d'un filtre passe-bas numérique



Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

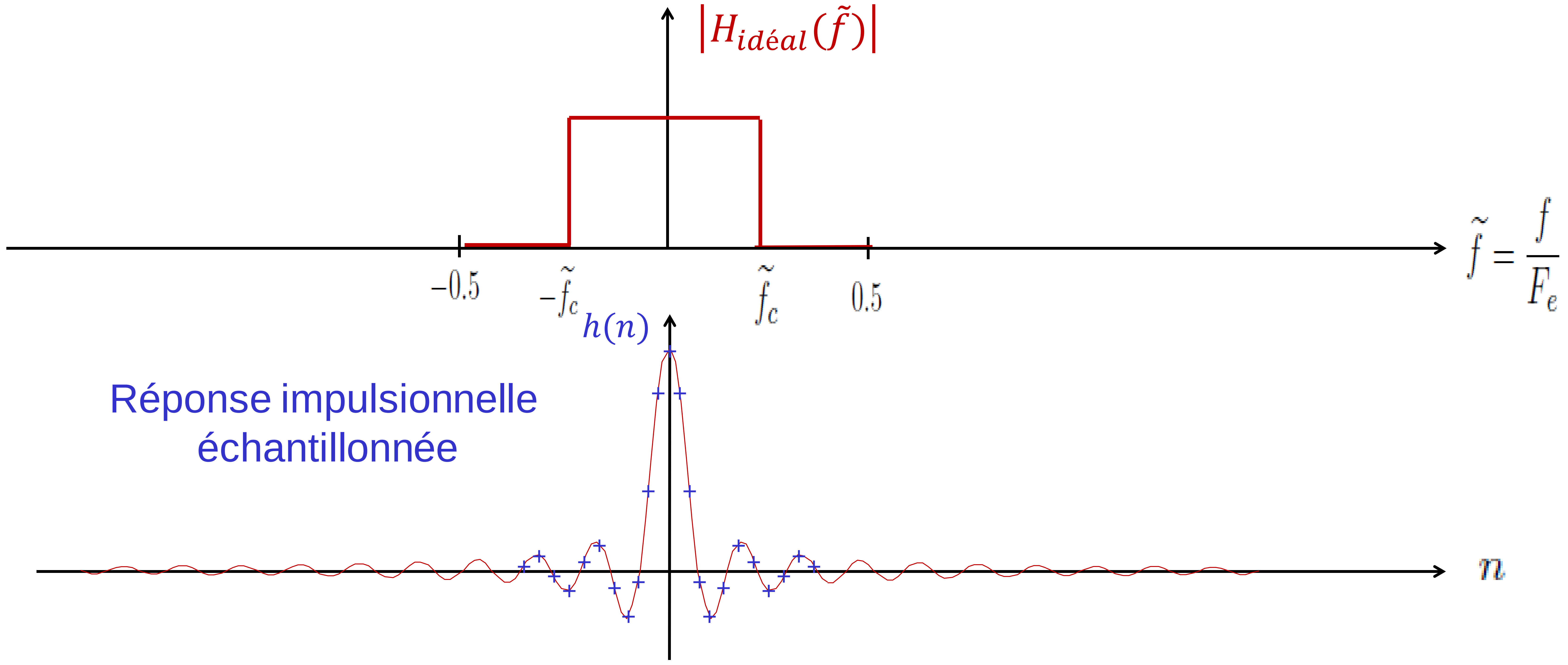
  → Exemple : synthèse d'un filtre passe-bas numérique



Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

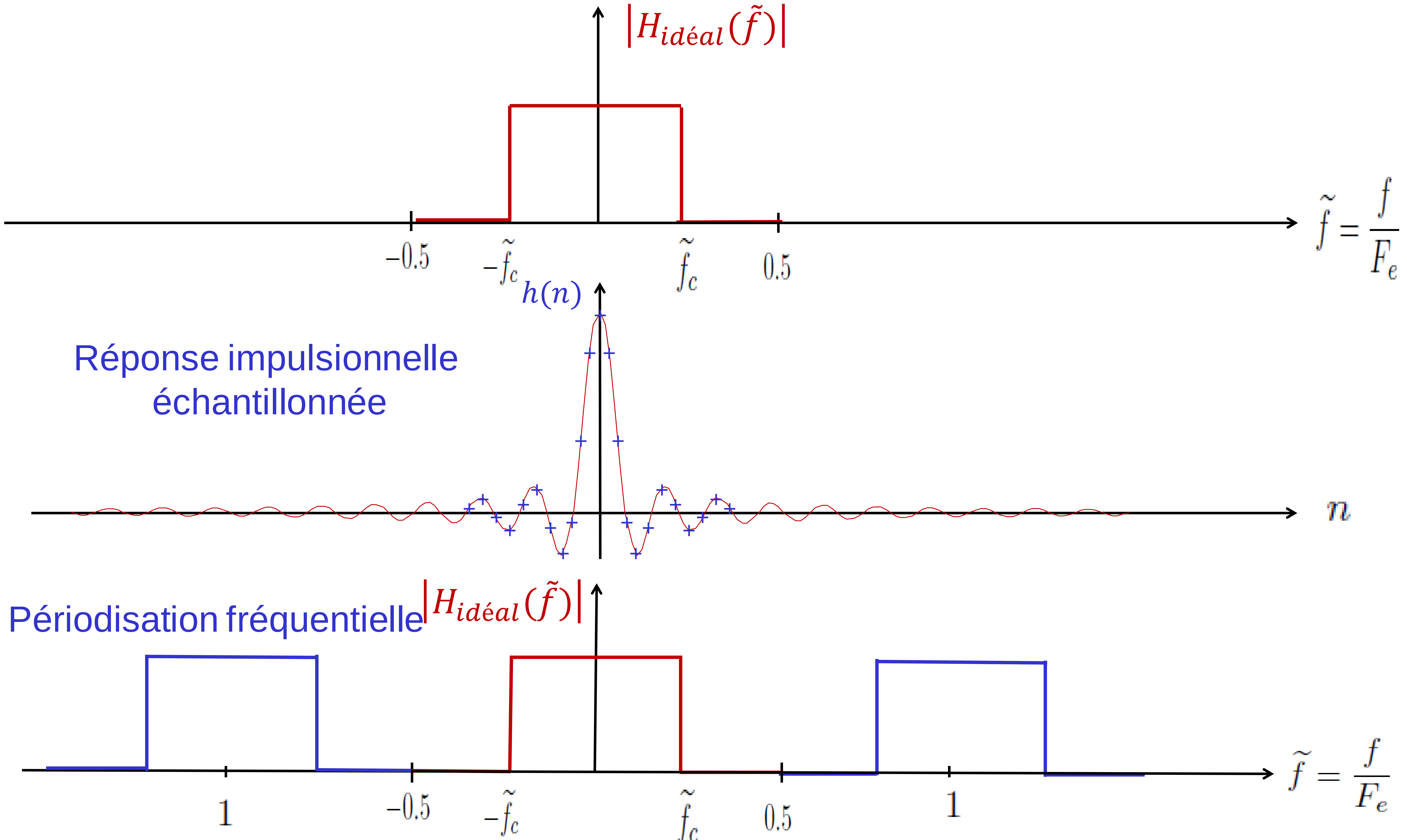
→ Exemple : synthèse d'un filtre passe-bas numérique



Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

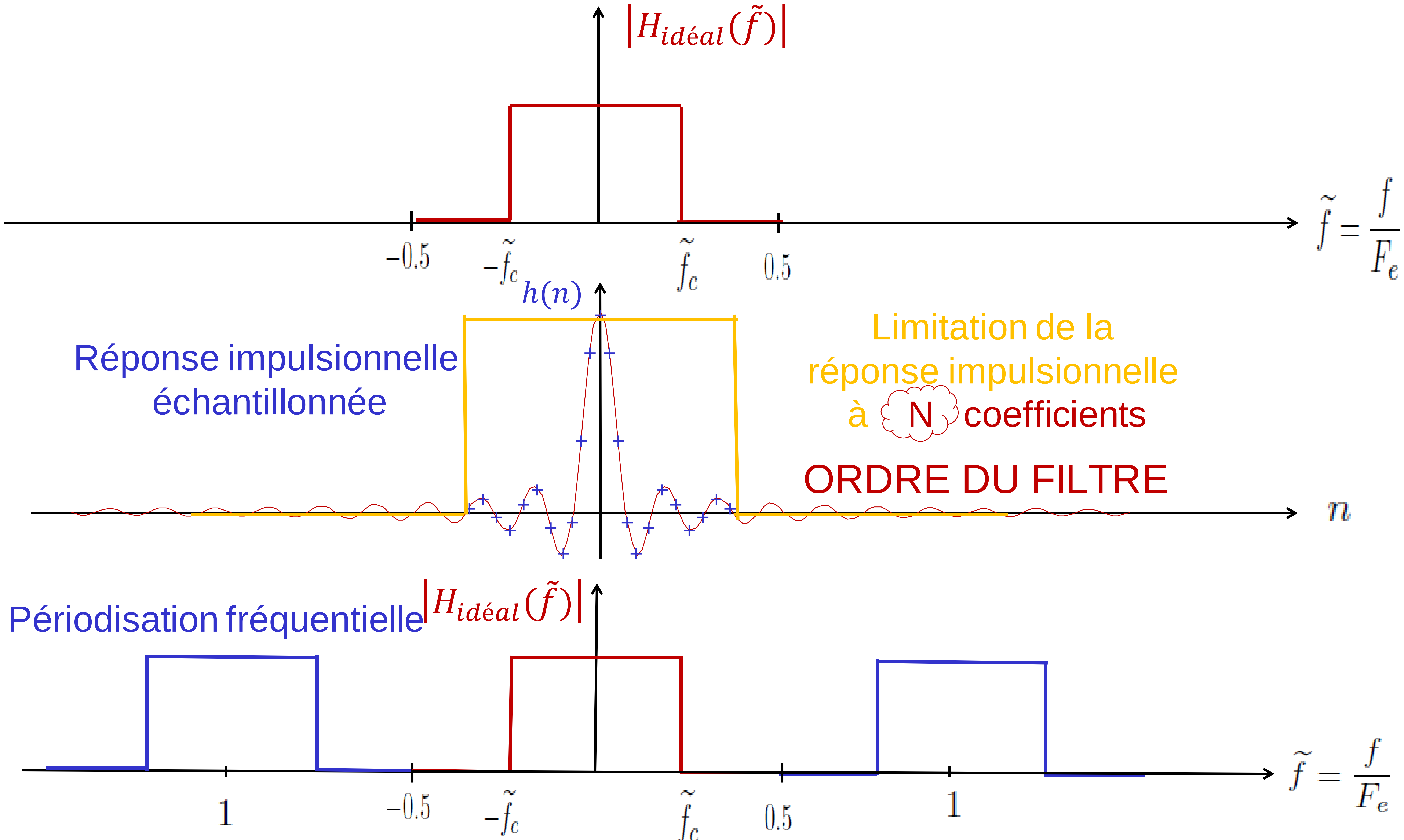
→ Exemple : synthèse d'un filtre passe-bas numérique



Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

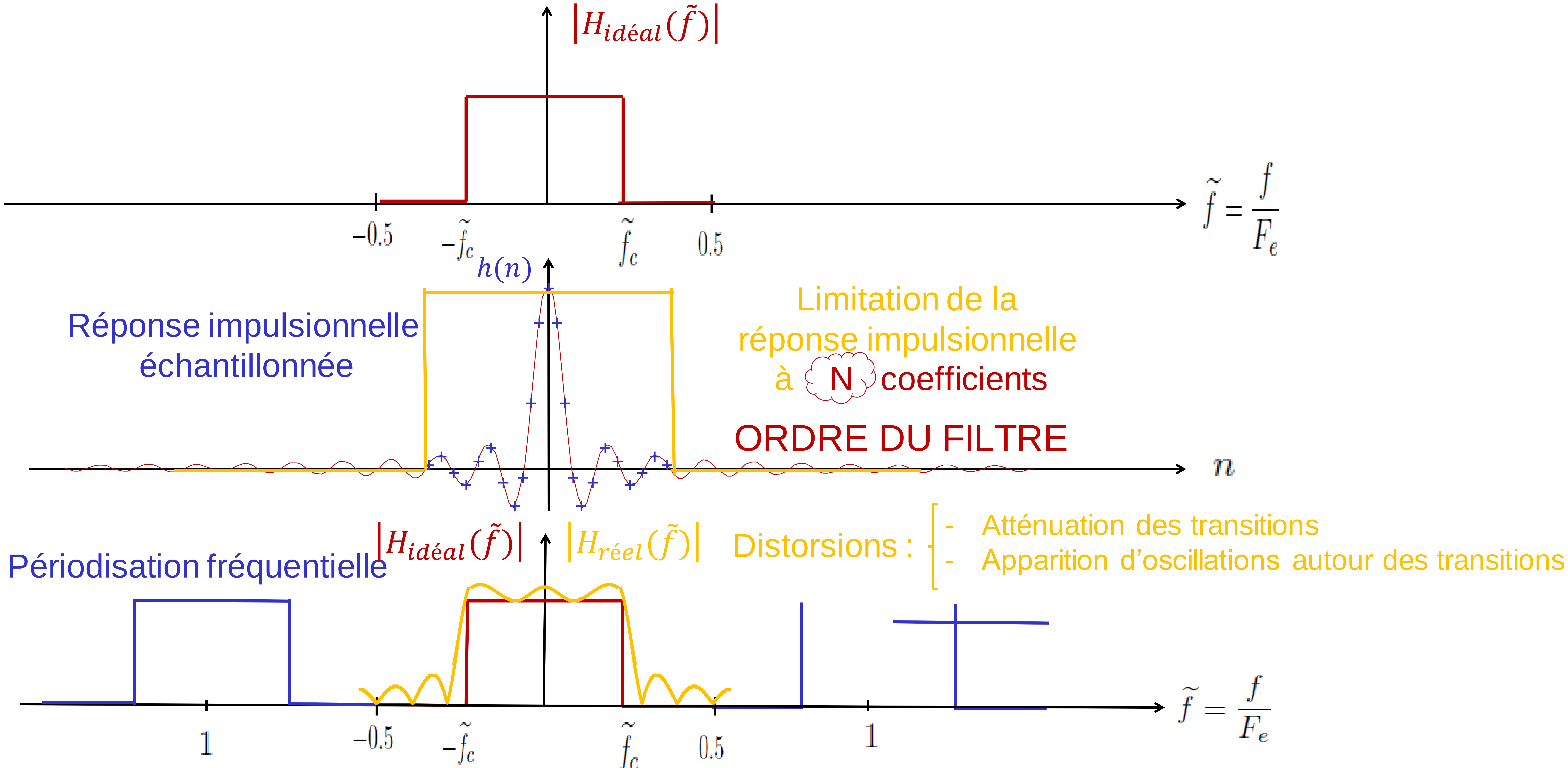
  → Exemple : synthèse d'un filtre passe-bas numérique



Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

→ Exemple : synthèse d'un filtre passe-bas numérique

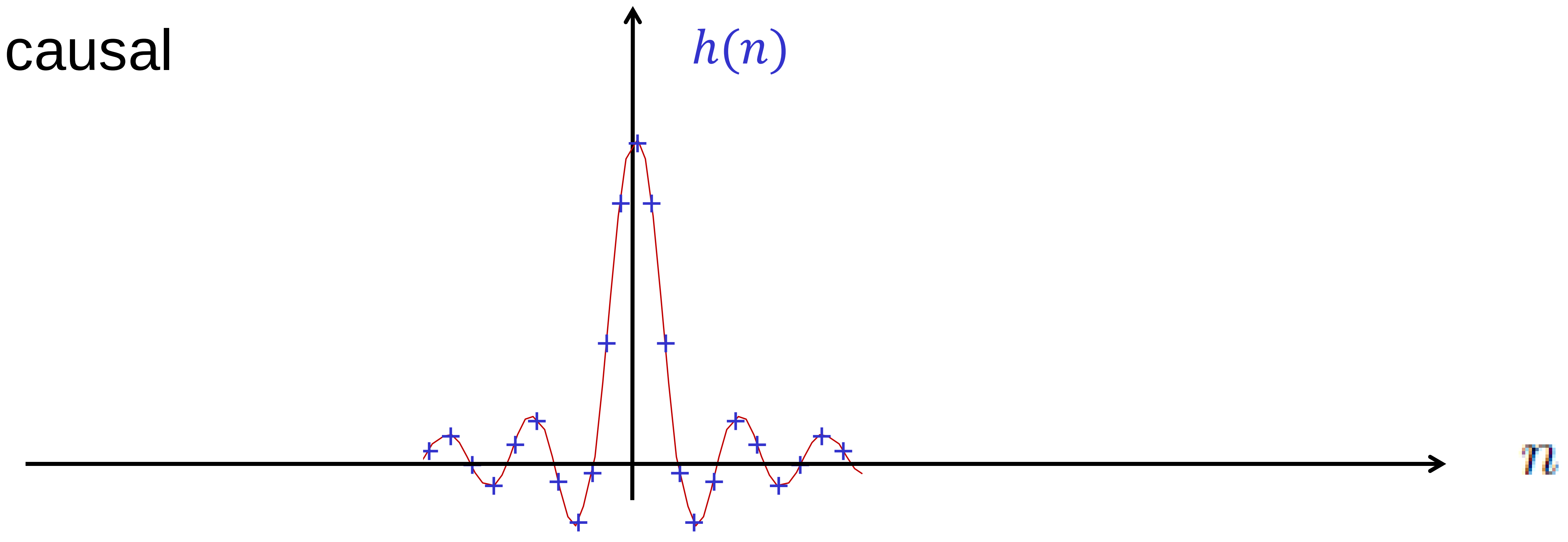


Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

  → Exemple : synthèse d'un filtre passe-bas numérique

  - Filtres non causal

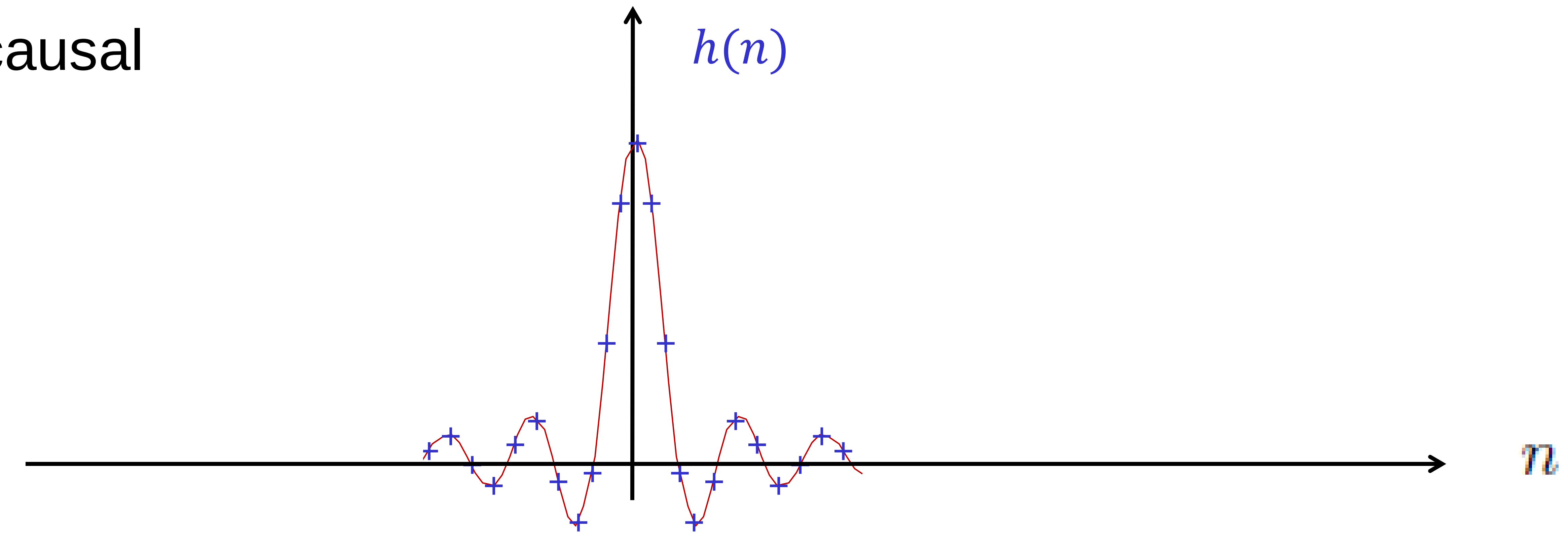


Filtrage numérique

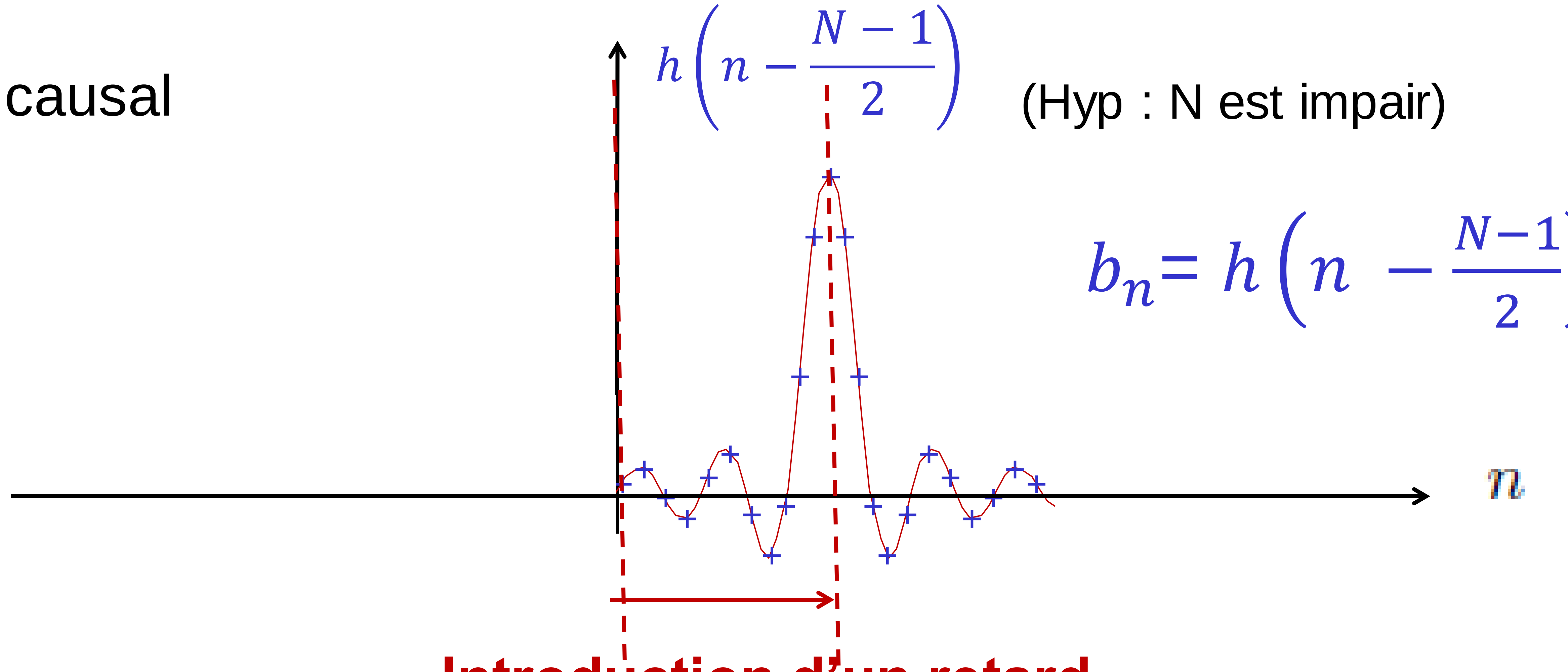
- **Synthèse des filtres numériques de type RIF**

  → Exemple : synthèse d'un filtre passe-bas numérique

  - Filtre non causal



  - Filtre rendu causal



(Hyp : N est impair)

$$b_n = h\left(n - \frac{N-1}{2}\right)$$

Introduction d'un retard

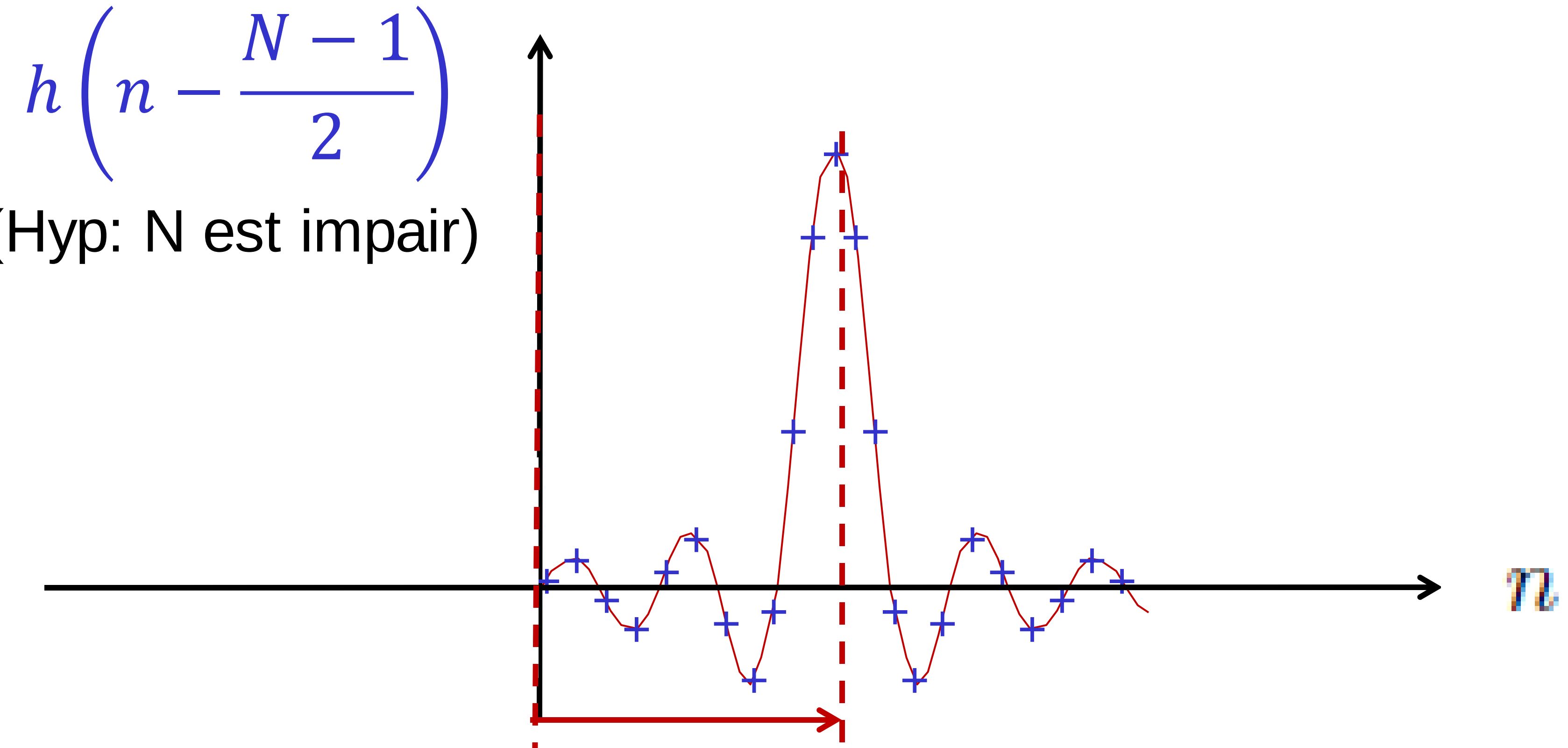
Filtrage numérique

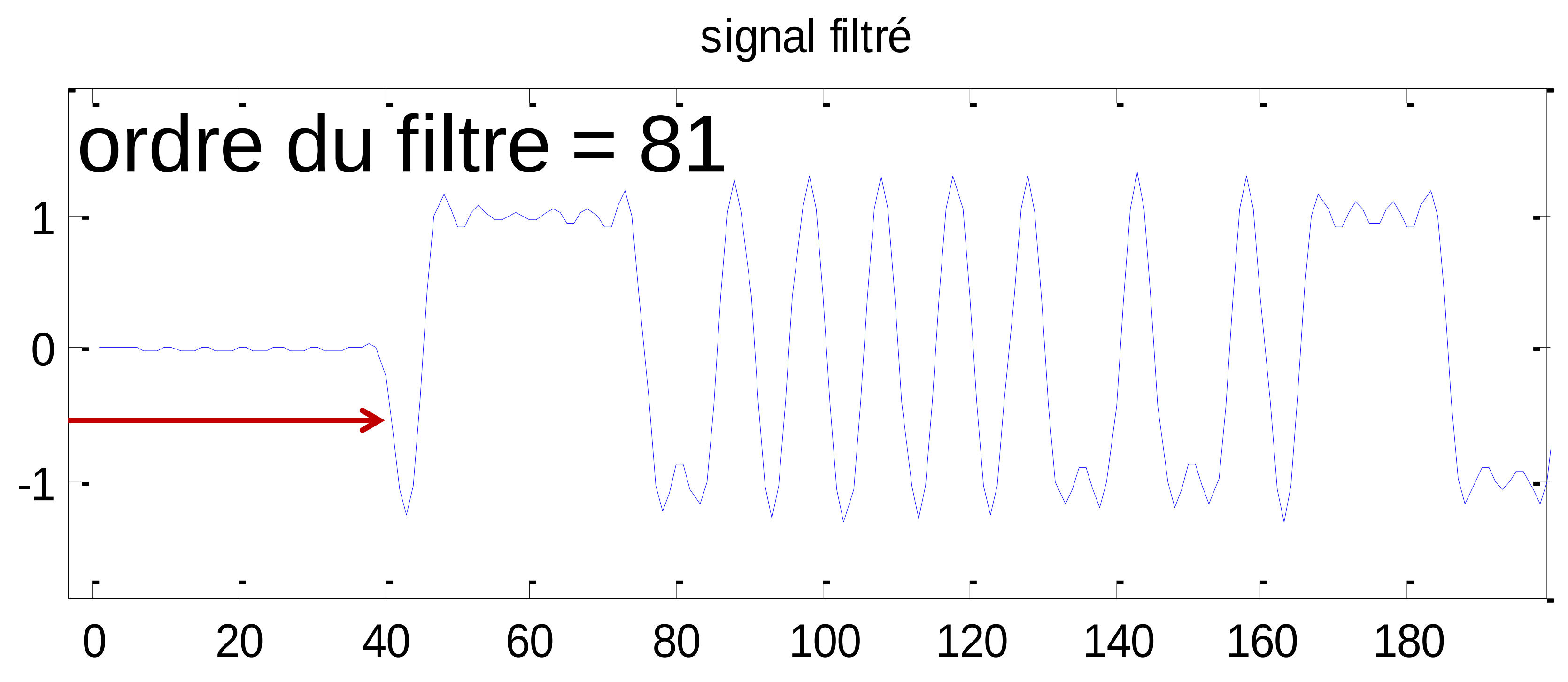
- **Synthèse des filtres numériques de type RIF**

  → Exemple : synthèse d'un filtre passe-bas numérique

$h\left(n-\frac{N-1}{2}\right)$

(Hyp: N est impair)



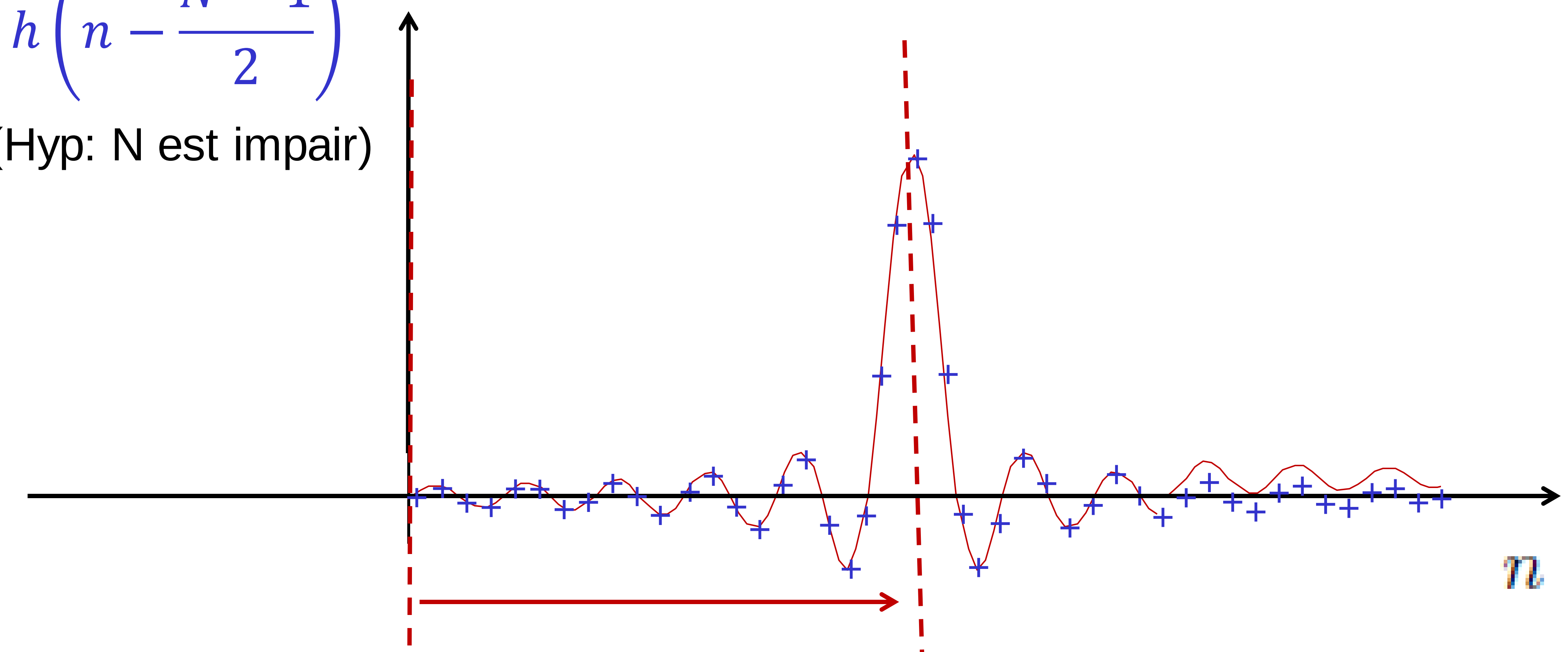


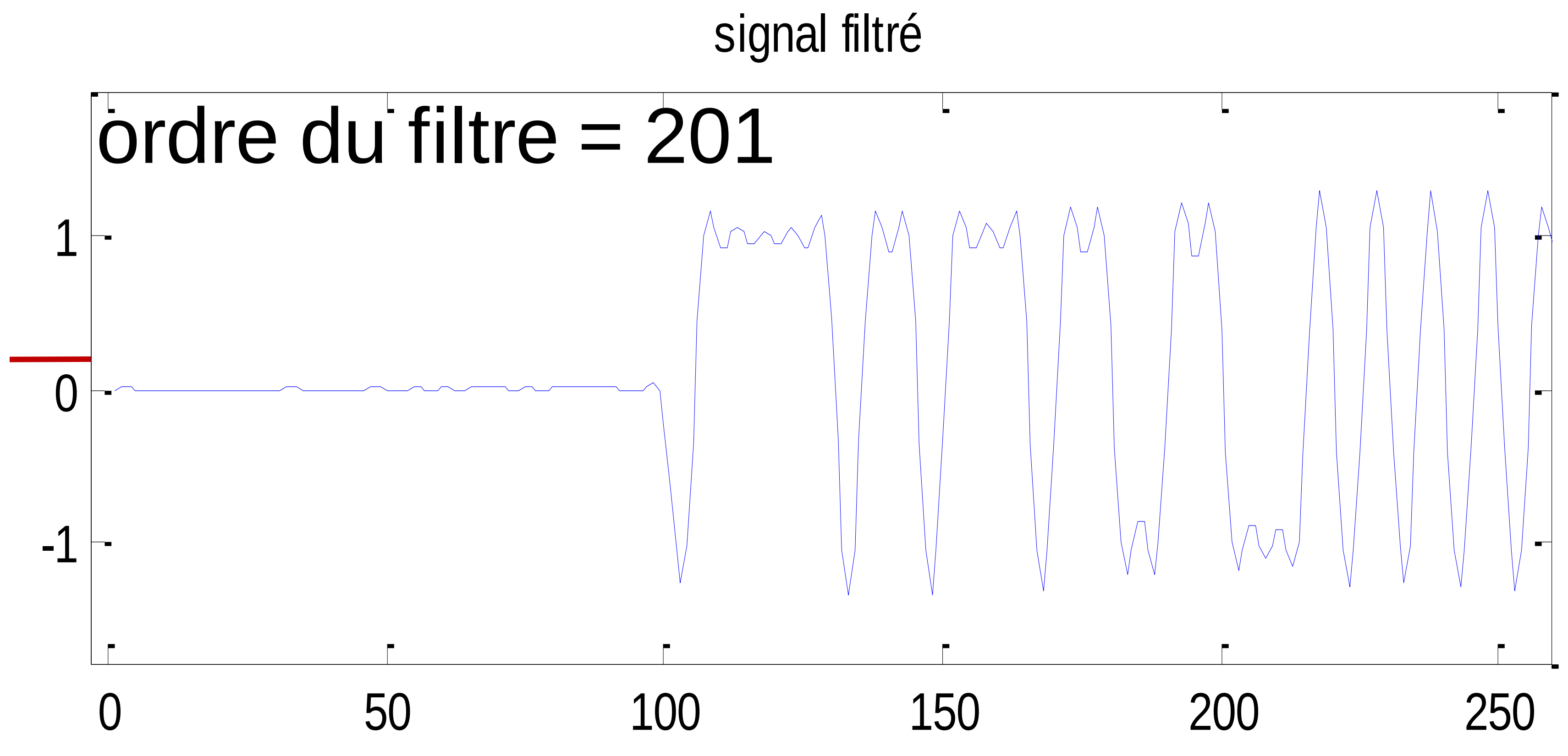
Introduction d'un retard sur la réponse impulsionnelle

Introduction d'un retard sur le signal filtré

$h\left(n-\frac{N-1}{2}\right)$

(Hyp: N est impair)

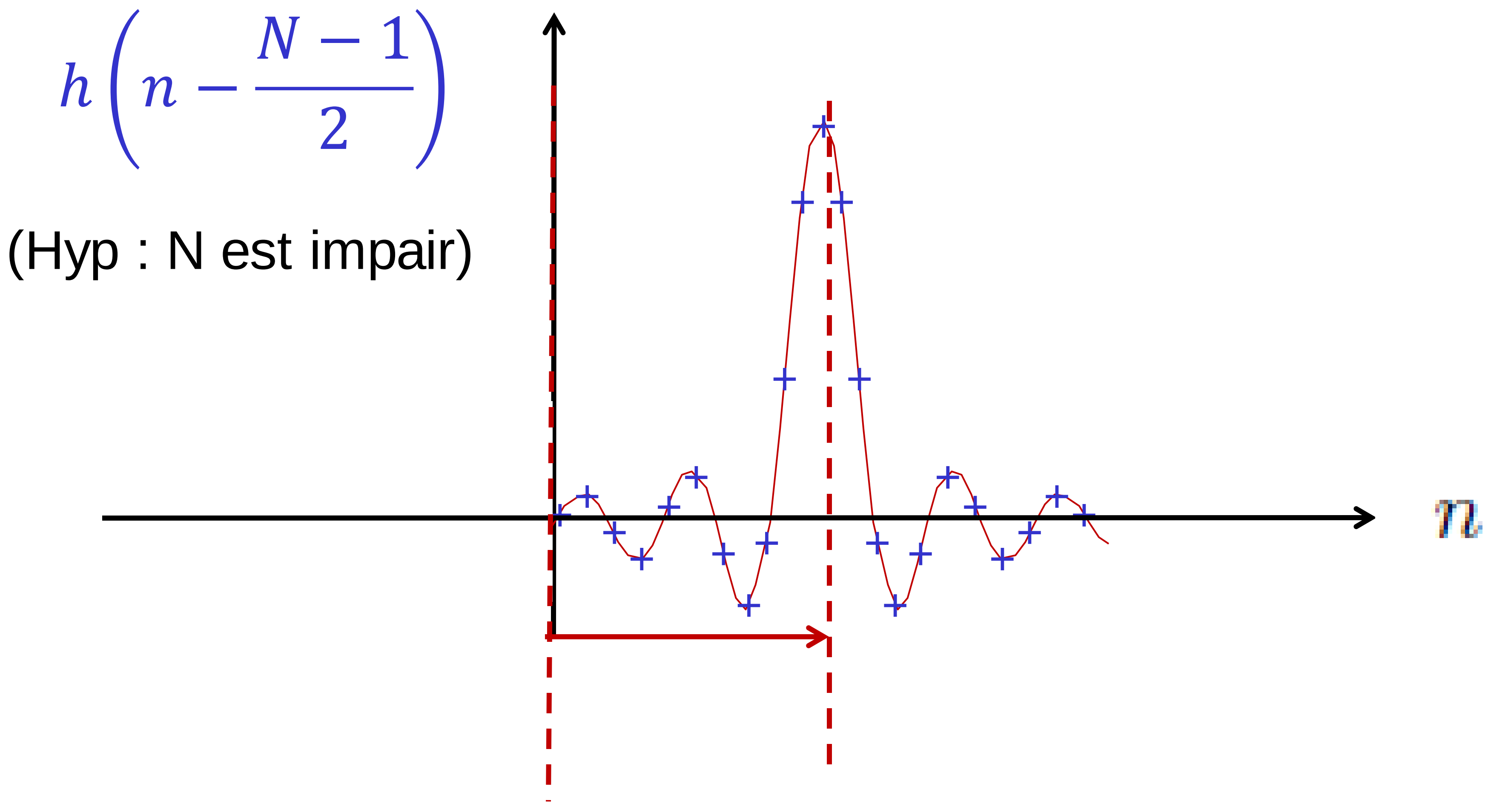




Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

  → Exemple : synthèse d'un filtre passe-bas numérique



$$Retard = \frac{Ordre - 1}{2} \times T_e$$

Retard ⇒ Constante ajoutée au TPG :

$$\left|H_{causal}(\tilde{f})\right| = \left|H_{non\ causal}(\tilde{f})\right|$$

$$\underbrace{\underbrace{TPG_{filtre\ causal}\left(\tilde{f}\right) = TPG_{filtre\ non\ causal}\left(\tilde{f}\right)}_{\text{Constant si } h(n) \text{ pair ou impair (voir poly)}} + \frac{N-1}{2}}$$

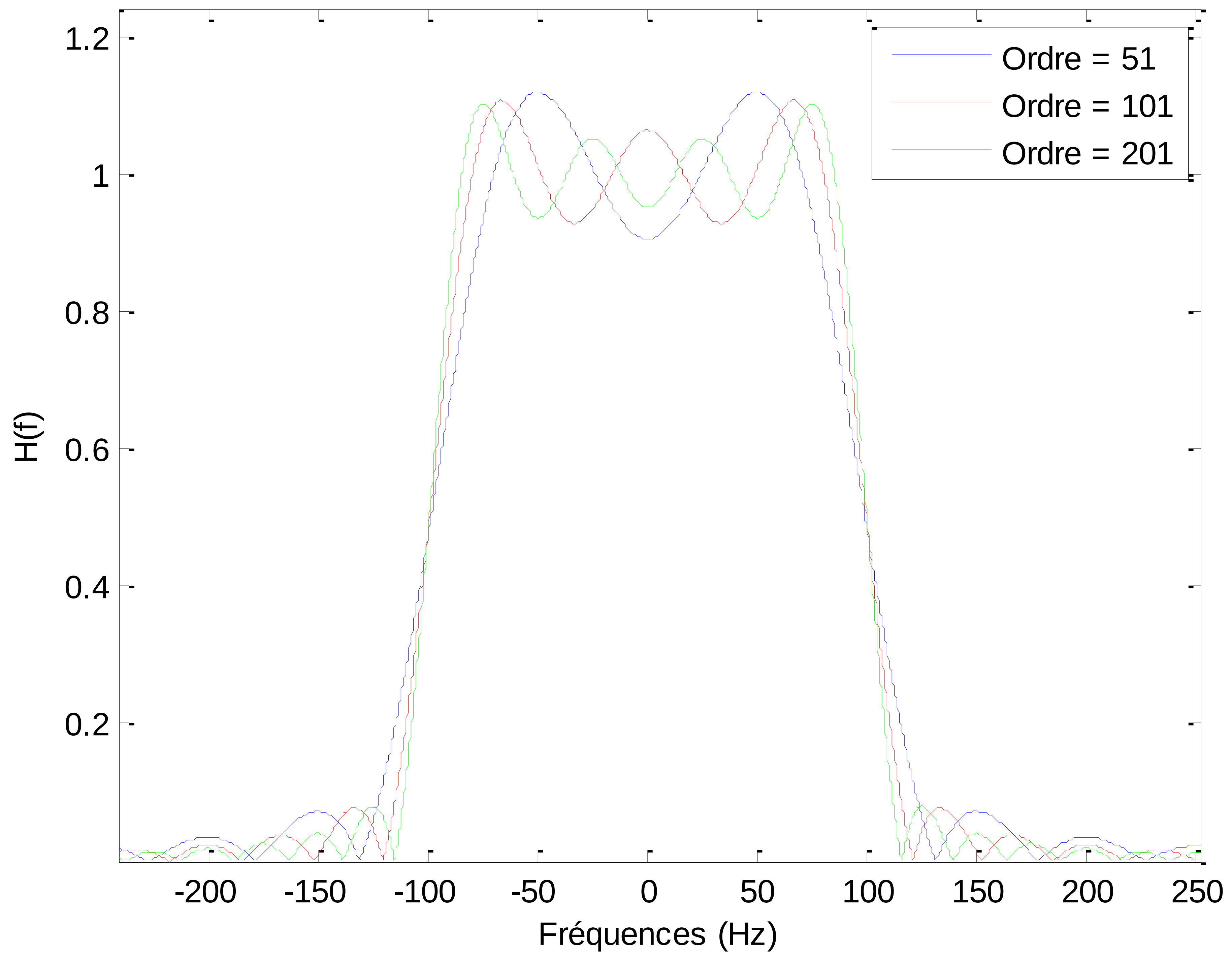
Le TPG d'un filtre RIF est constant si sa réponse impulsionnelle est paire ou impaire

Filtrage numérique

- **Synthèse des filtres numériques de type RIF**

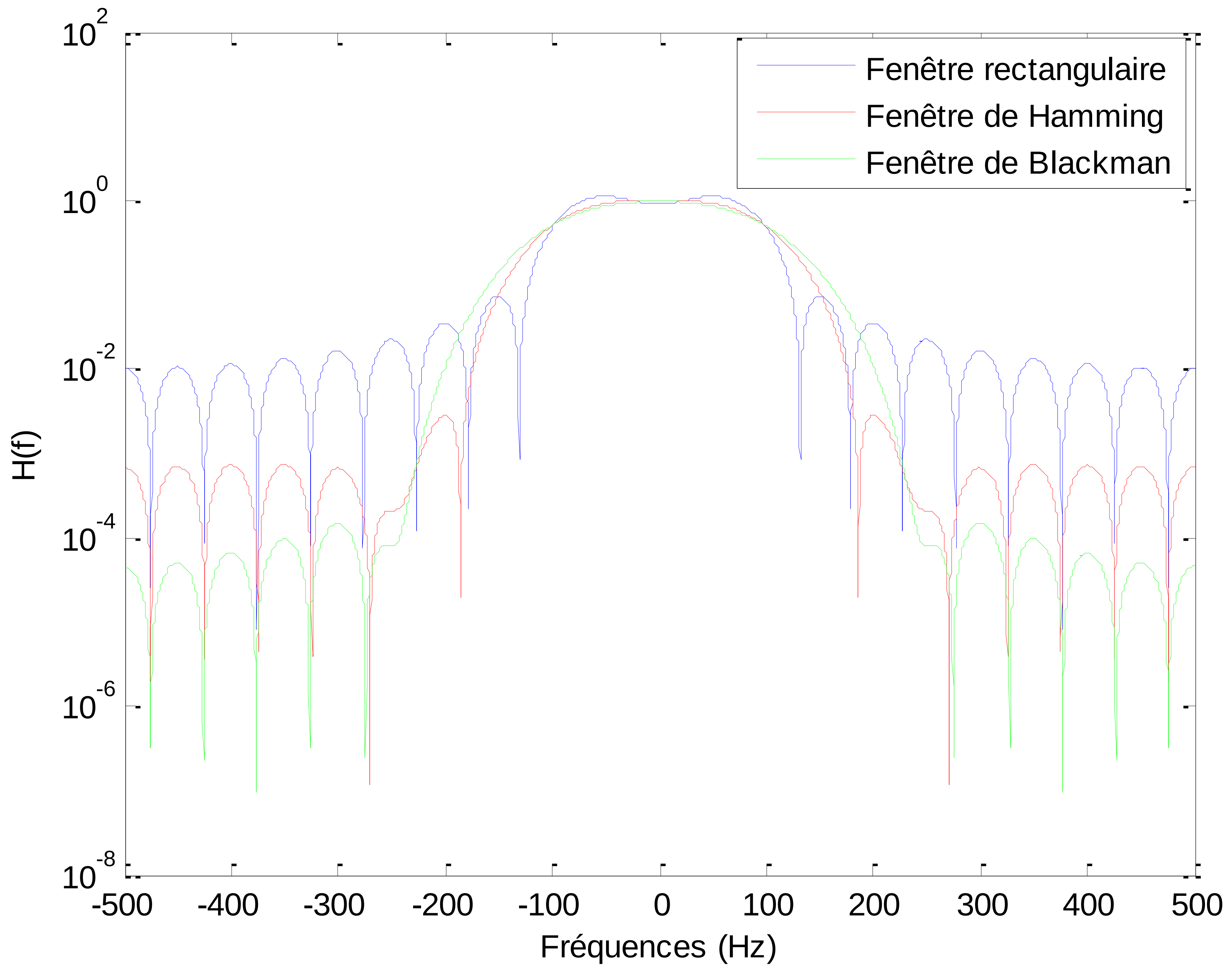
→ <u>Paramètres permettant de respecter le gabarit : ORDRE et FENETRE DE TRONCATURE</u>

→ Influence de l'ordre
(fenêtre rectangulaire)



Atténuation plus ou moins importantes des transitions

→ Influence de la fenêtre de troncature
(ordre fixé à 21)



Ondulations plus ou moins importantes autour des transitions

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Spécifications à respecter :
$H(\tilde{f})$

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

Fonction de transfert :
$H(z)$

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Spécifications à respecter :
$\mathbf{H(\tilde{f})}$

Spécifications à respecter :
$\mathbf{H(f)}$

$\downarrow$

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

$\downarrow$

Fonction de transfert :
$\mathbf{H(p)}$

Fonction de transfert :
$\mathbf{H(z)}$

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Spécifications à respecter :
$H(\tilde{f})$

?

Spécifications à respecter :
$H(f)$

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

Fonction de transfert :
$H(p)$

?

Fonction de transfert :
$H(z)$

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Spécifications à respecter :
H ($\tilde{f}$)

⬇ **?** $f = \tilde{f} F_e$

Spécifications à respecter :
H (f)

↓

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

↓

Fonction de transfert :
H (p)

⬇ **?**

Fonction de transfert :
H (z)

TRANSFORMEE BILINAIRE :

$$H(z) = [H(p)]_{p=\frac{2}{T_e}\frac{1-z^{-1}}{1+z^{-1}}}$$

Stabilité et réponse en fréquence conservées

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Spécificités à respecter :

H ($\tilde{f}$)

⬇ **?** $f = \tilde{f} F_e$

Spécifications à respecter :

H (f)

↓

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

↓

Fonction de transfert :

H (p)

⬇ **?**

Fonction de transfert :

H (z)

TRANSFORMEE BILINAIRE :

$$H(z) = [H(p)]_{p=\frac{2}{T_e}\frac{1-z^{-1}}{1+z^{-1}}}$$

Stabilité et réponse en fréquence conservées
MAIS Distorsion de l'axe des fréquences:

$$\tilde{f} = \frac{1}{\pi}\arctan(\pi f T_e)$$

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Spécifications à respecter :
H ($\tilde{f}$)

⬇ **?** ~~$f = \tilde{f} F_e$~~ $f = \dfrac{1}{\pi T_e}\tan(\pi \tilde{f})$

Prédistorsion

Spécifications à respecter :
H (f)

↓

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

↓

Fonction de transfert :
H (p)

⬇ **?**

TRANSFORMEE BILINAIRE :

$$H(z) = [H(p)]_{p=\frac{2}{T_e}\frac{1-z^{-1}}{1+z^{-1}}}$$

Stabilité et réponse en fréquence conservées
MAIS Distorsion de l'axe des fréquences:

$$\tilde{f} = \frac{1}{\pi}\arctan(\pi f T_e)$$

Fonction de transfert :
H (z)

Filtrage numérique

- **Synthèse des filtres numériques de type RII**

Spécifications à respecter :

H ($\tilde{f}$)

⬇ **?**

~~$f = \tilde{f} F_e$~~

$$f = \frac{1}{\pi T_e} \tan(\pi \tilde{f})$$

Prédistorsion

Spécifications à respecter :

H (f)

↓

Utilisation des bibliothèques
de modèles analogiques :
Butterworth, Tchebycheff, Cauer, Bessel…

↓

Fonction de transfert :

H (p)

⬇ **?**

TRANSFORMEE BILINAIRE :

$$H(z) = [H(p)]_{p=\frac{2}{T_e}\frac{1-z^{-1}}{1+z^{-1}}}$$

Stabilité et réponse en fréquence conservées
MAIS Distorsion de l'axe des fréquences:

$$\tilde{f} = \frac{1}{\pi} \arctan(\pi f T_e)$$

Fonction de transfert :

H (z)

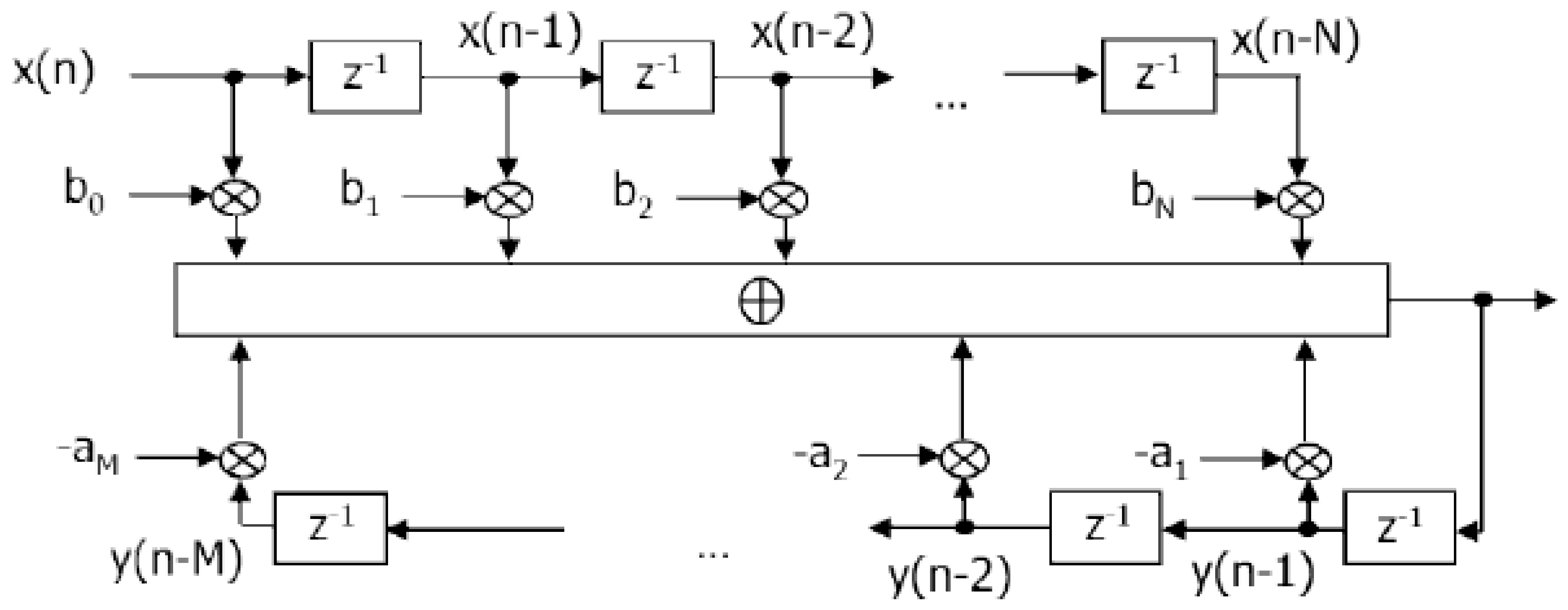
Voir poly d'exercices pour un exemple de synthèse de filtre passe-bas RII

Filtrage numérique

- **Implantation des filtres numériques**

→ <u>Structure directe :</u>

$$y(n) = -\sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k)$$



M+N+1 opération +/x, 2 files d'attente

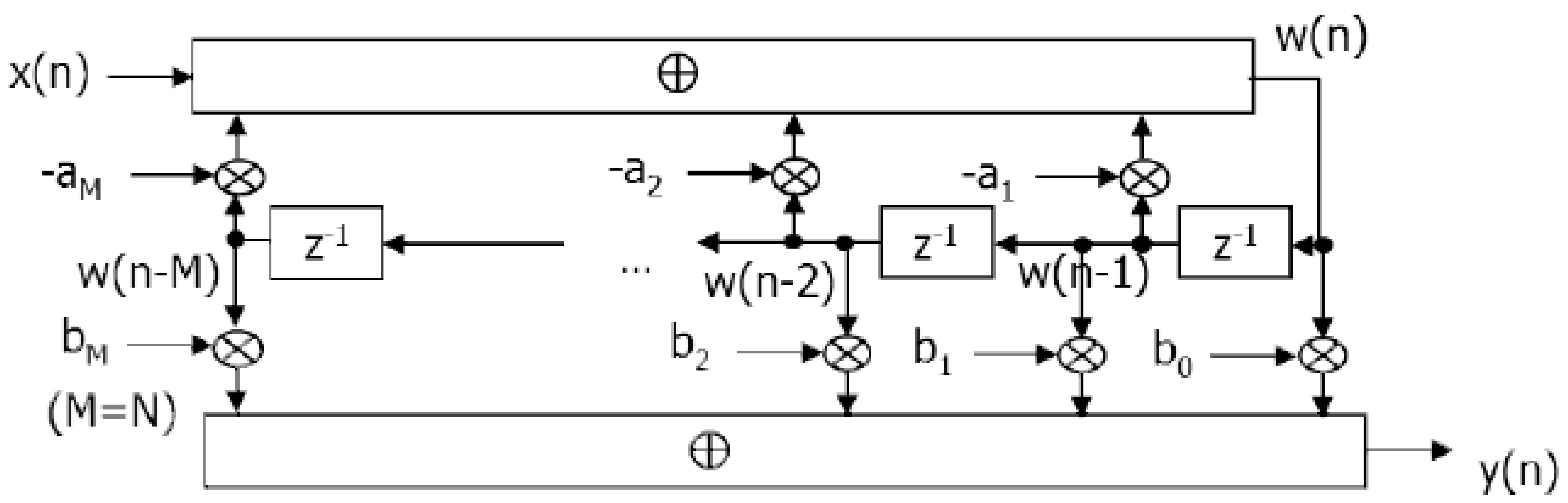
Filtrage numérique

- **Implantation des filtres numériques**

→ <u>Structure canonique :</u>

$$W(z) = \frac{X(z)}{\sum_{k=0}^{M-1} a_k z^{-k}}$$

$$\begin{cases} w(n) = -\sum_{k=1}^{M-1} a_k w(n-k) + x(n) \\ y(n) = \sum_{k=0}^{N-1} b_k w(n-k) \end{cases}$$



M+N+1 opération +/x, **1** file d'attente

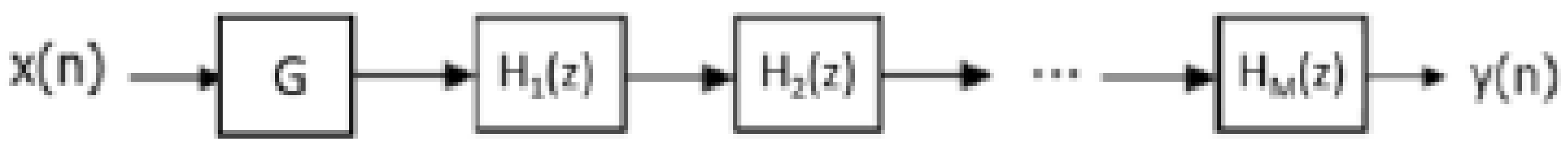
Filtrage numérique

- **Implantation des filtres numériques**

→ Structures décomposées :

→ Série (ou cascade) :

$$H(z) = G \prod_{i=0}^{M-1} H_i(z)$$

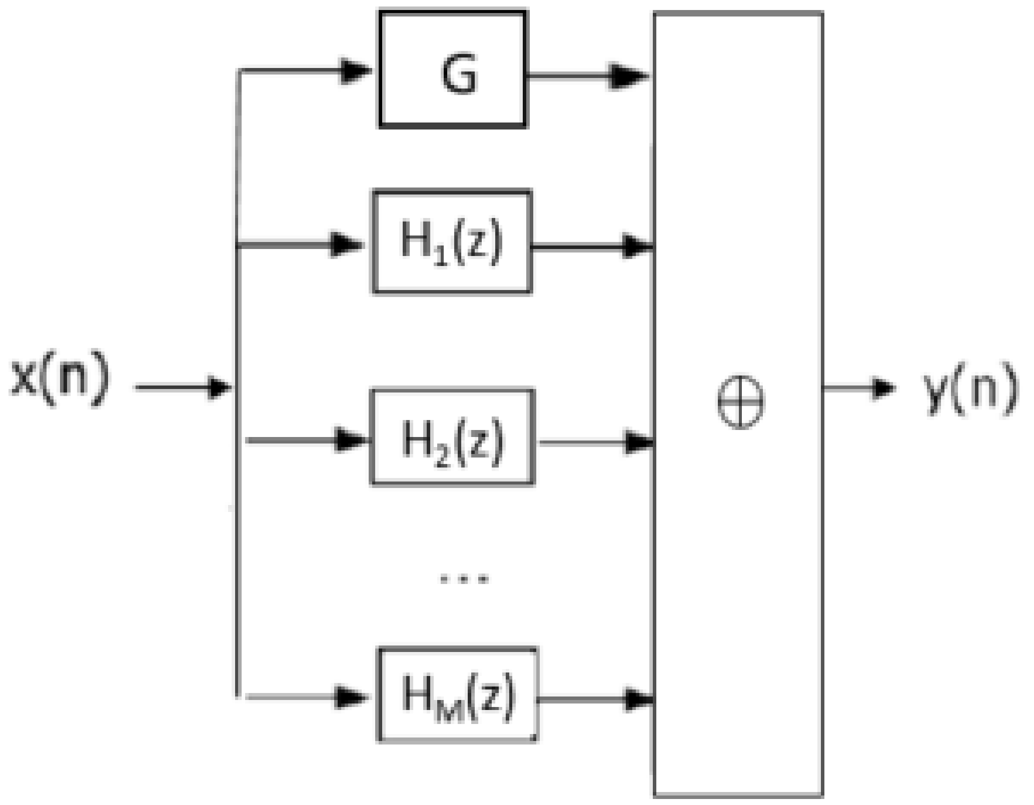


$H_i(z)$ Cellules du premier ou du deuxième ordre :

$$H_i(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \qquad H_i(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

→ Parallèle :

$$H(z) = G + \sum_{i=0}^{M-1} H_i(z)$$



$H_i(z)$ Cellules du premier ou du deuxième ordre :

$$H_i(z) = \frac{b_0}{1 + a_1 z^{-1}}$$

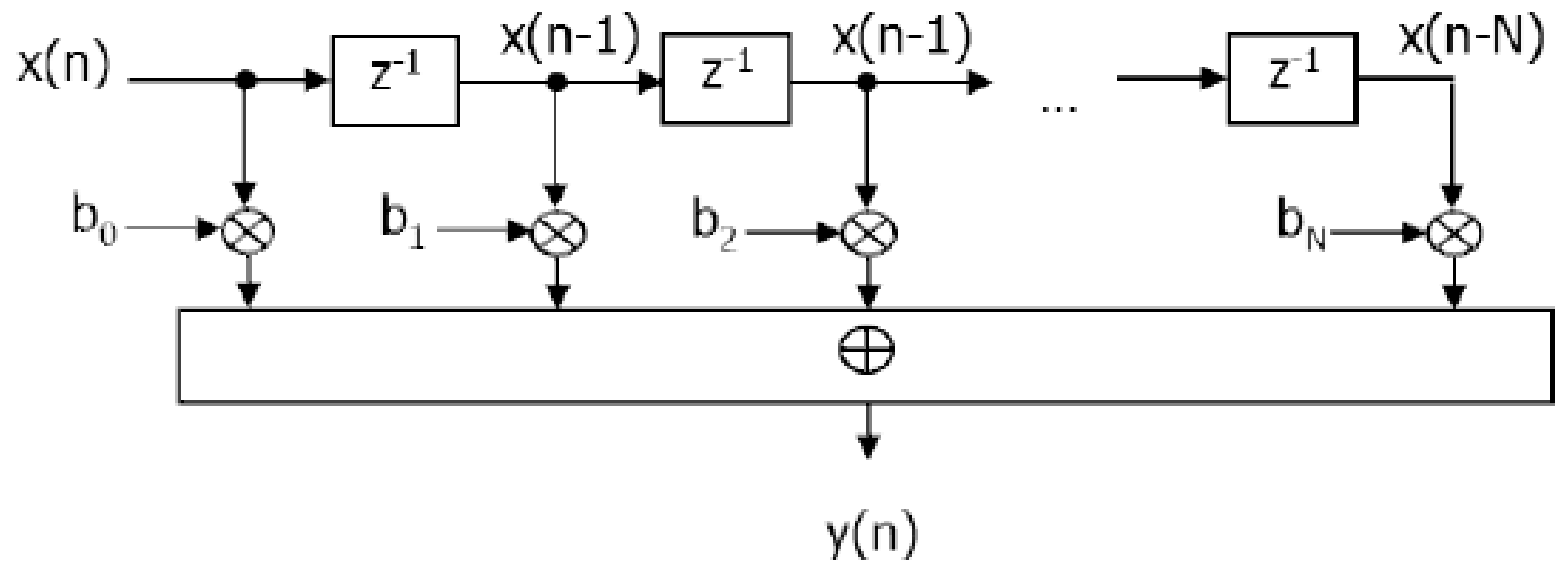
$$H_i(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Filtrage numérique

- **Implantation des filtres numériques**

→ <u>Structure non récursive :</u>

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k)$$



Références

→"Traitement numérique du signal, théorie et pratique", M. Bellanger, Masson, collection CNET-ENST.

→"Traitement numérique des signaux", M. Kunt, Dunod, Traité d'électricité, d'électronique et d'électrotechnique.

→"Traitement numérique du signal, Une introduction", A.W.M. Van Den Enden et N.A.M. Verhoeckx, Masson

→"Introduction au traitement du signal", P. Duvaut, F. Michaut, M. Chuc, Hermes, Collection traitement du signal

→Documents sur la variable complexe, la transformée de Laplace et la transformée en z :
http://dobigeon.perso.enseeiht.fr/teaching/complexe.html

→"Introduction to digital filters, with audio applications", J.O. Smith, BookSurge, 2007

→"Digital signal processing : fundamentals and applications", Tan Li, Jiang Jean, Elsevier, 2013.

→Alan V. Oppenheim, Ronald W. Schafer et J. R. Buck, Discrete-time signal processing, Upper Saddle River, N.J., Prentice Hall, 3ième édition, 2009.

→Signal and Systems, by Simon Haykin and Barry Von Veen, Wiley, 2nde édition, 2002.

→John G. Proakis, Dimitri G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, Pearson Education, 4ième édition, 2006.