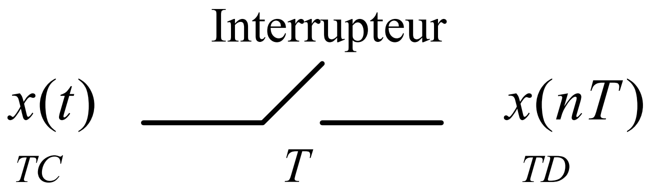
5. Echantillonnage - Interpolation - Quantification

Echantillonnage : du TC au TD

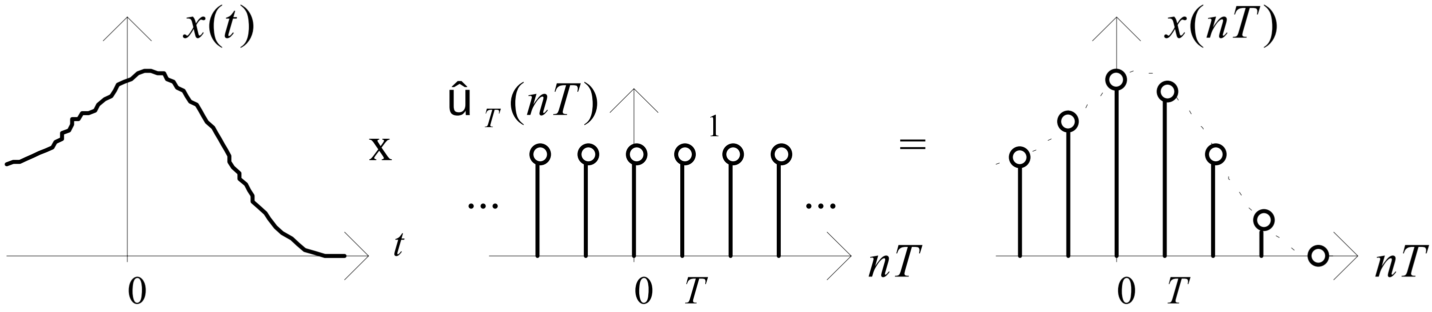
Signal échantillonné (comme signal à TD)

Représentation :



Echantillonnage d'un signal à TC = Remplacer t par nT

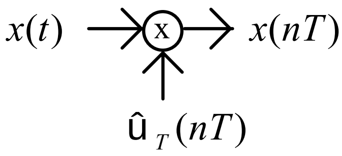
Echantillonnage d'un signal à TC ≡ multiplication par $\text{û}_T(nT)$

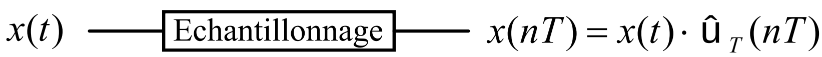


$$x(nT) = \sum_{k=-\infty}^{\infty} x(t)\delta_{n,k} = x(t)\sum_{k=-\infty}^{\infty}\delta_{n,k} = x(t)\,\text{û}_T(nT)$$

$$\boxed{x(nT) = x(t)\,\text{û}_T(nT)}$$

avec : $\text{û}_T(nT) = \sum_{k=-\infty}^{\infty}\delta_{n,k}$





Signal échantillonné comme signal à signal à TC

$$\boxed{x^*(t) = x(t)\cdot \hat{u}_T(t)}$$

$$x^*(t) = x(t)\sum_{n=-\infty}^{\infty}\delta(t-nT) = \sum_{n=-\infty}^{\infty}x(nT)\delta(t-nT)$$

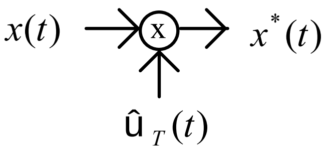
Echantillonnage d'un signal à TC ≡ multiplication par $\hat{u}_T(t)$

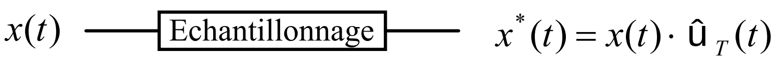
Avantage de représenter le signal échantillonné comme signal à TC :

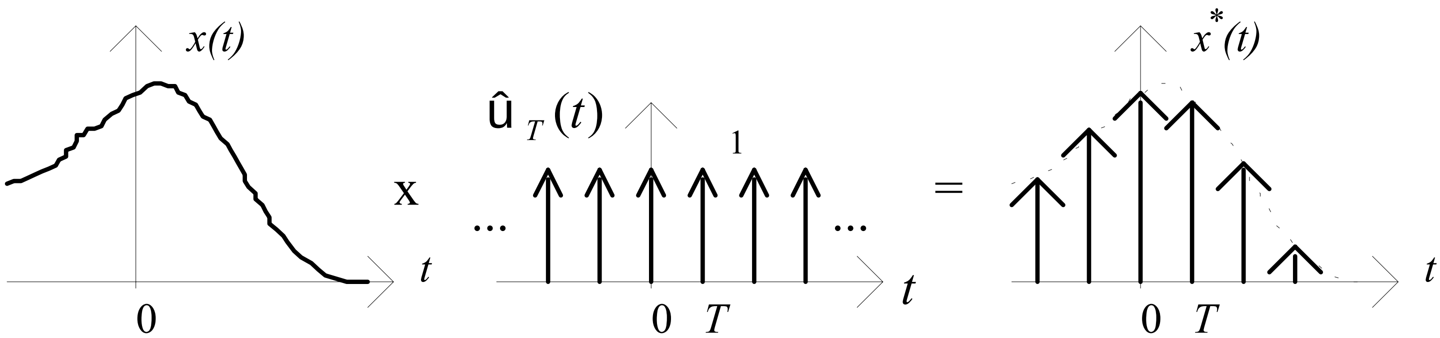
Etude spectrale facilitée

$x^*(t)$ = équivalent continu du signal échantillonné à TD $x(nT)$

(ils ont la même TL)







Théorème de Shannon

$$x^*(t) = x(t) \cdot \text{û}_T(t)$$

Spectre de $x^*(t)$ *:*

$$TF\left[x^*(t)\right] = TF\left[x(t) \cdot \text{û}_T(t)\right] = TF\left[x(t)\right] * TF\left[\text{û}_T(t)\right] = X(\nu) * TF\left[\text{û}_T(t)\right]$$

On a vu : $TF\left[\text{û}_T(t)\right] = F\ \text{û}_F(\nu)$ avec : $F = 1/T$

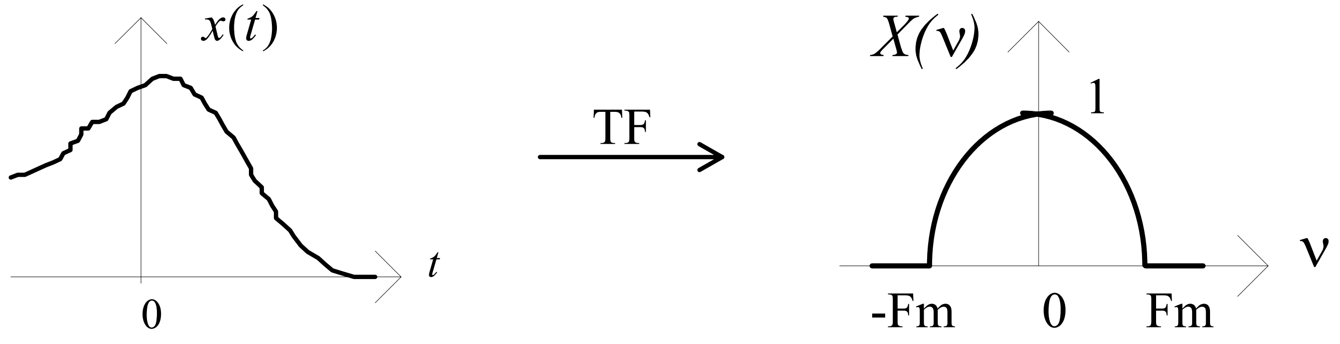
$$TF\left[x^*(t)\right] = X(\nu) * F\ \text{û}_F(\nu) = X(\nu) * F \sum_{k=-\infty}^{\infty} \delta(\nu - kF) = F \sum_{k=-\infty}^{\infty} X(\nu) * \delta(\nu - kF)$$

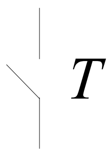
Rappel : $X(\nu) * \delta(\nu - kF) = X(\nu - kF)$

$$\rightarrow \quad \boxed{TF\left[x^*(t)\right] = F \sum_{k=-\infty}^{\infty} X(\nu - kF)}$$

TEMPS		FREQUENCE
Echantillonner *x(t)*	=	Périodiser le spectre de *x(t)* à la période $F = 1/T$

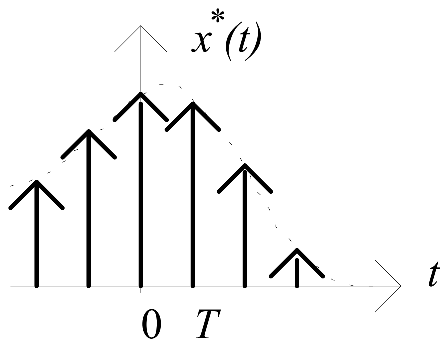
Exemple : Spectre $X(\nu)$ de *x(t)*





Fm : fréquence max. contenue dans le spectre de *x(t)*

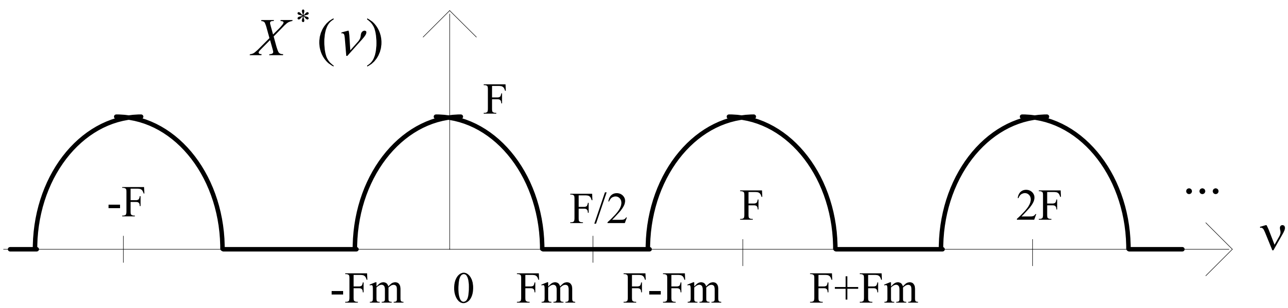
(Fm est finie, sinon le signal est infiniment rapide !)



Spectre $X^*(\nu)$ *de* $x^*(t)$ *:*

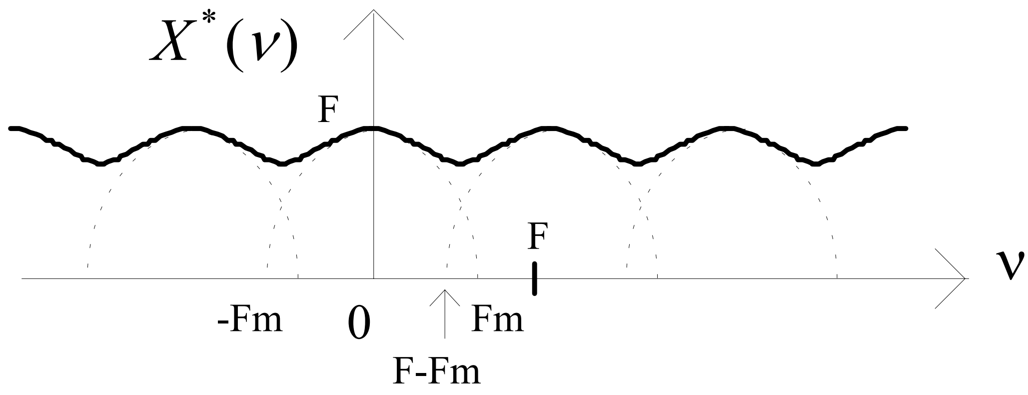
2 cas de figure :

(1) $F = \dfrac{1}{T} \geq 2Fm$:



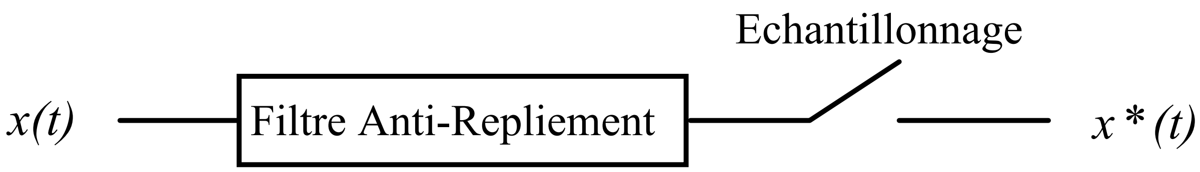
Pas de chevauchement : F-Fm ≥ Fm → F ≥ 2Fm

(2) $F < 2Fm$: sous-échantillonnage
repliement de spectre (aliasing)



Reconstitution impossible de *x(t)* à partir de ses échantillons $x^*(t)$

→ perte d'information



Théorème de Shannon :

Echantillonnage sans perte d'un signal : $\boxed{F \geq 2Fm}$

Application : *Transmission de la parole en temps réel*
(1 octet par échantillon - mono)

Débit ≥ 64 kbit/s (Spectre parole : Fm = 4 kHz → Fe ≥ 8 kHz)

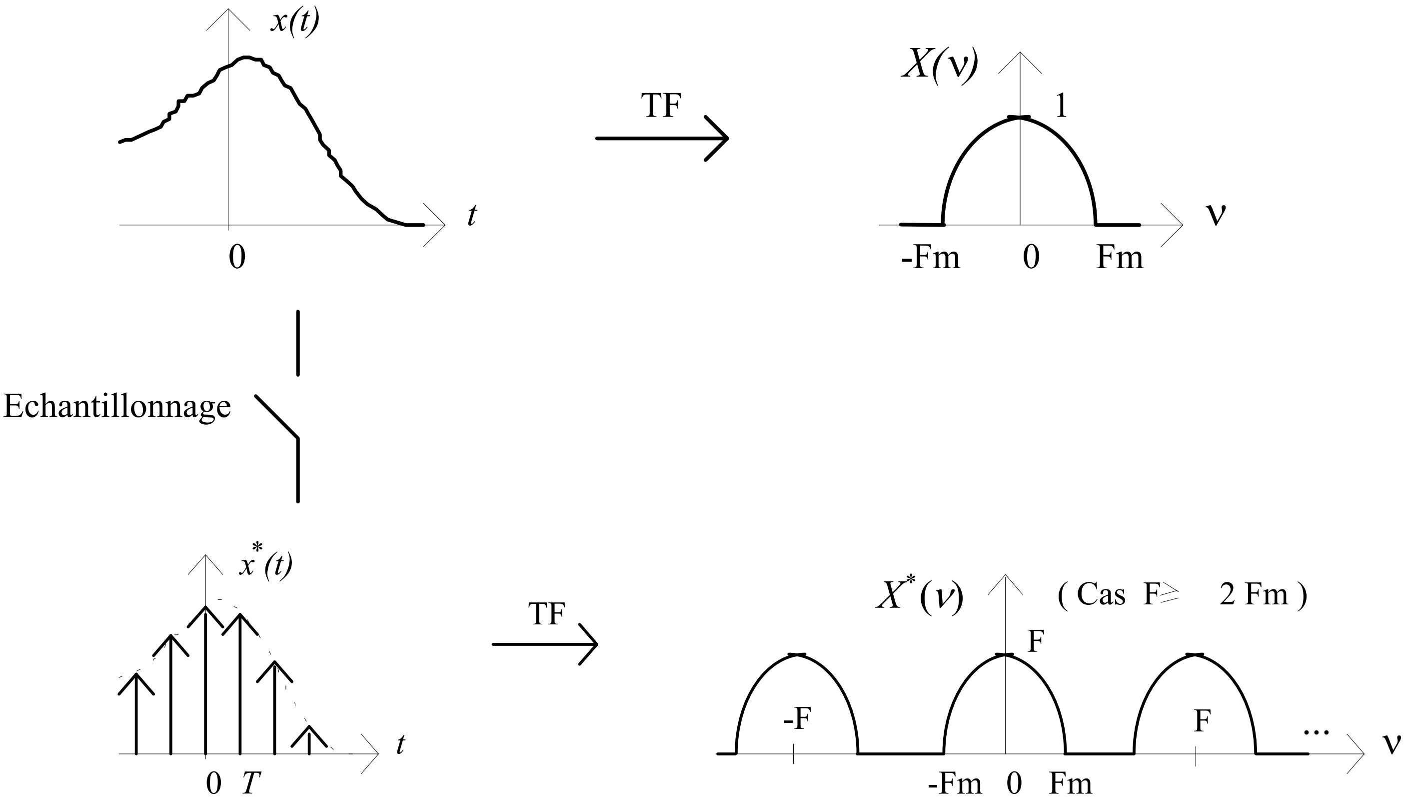
Opérations sur l'échantillonnage

TEMPS	LAPLACE	Z
$\left[x(t)+y(t)\right]^* = x^*(t)+y^*(t)$	$\left[X(p)+Y(p)\right]^* = X^*(p)+Y^*(p)$	$Z\left[X(p)+Y(p)\right] = X(z)+Y(z)$
$\left[x^*(t)\right]^* = x^*(t)$	$\left[X^*(p)\right]^* = X^*(p)$	$Z\left[X(z)\right] = X(z)$
$\left[x^*(t)*y(t)\right]^* = x^*(t)*y^*(t)$	$\left[X^*(p)\cdot Y(p)\right]^* = X^*(p)\cdot Y^*(p)$	$Z\left[X(z)\cdot Y(p)\right] = X(z)\cdot Y(z)$
$\left[x(t)*y^*(t)\right]^* = x^*(t)*y^*(t)$	$\left[X(p)\cdot Y^*(p)\right]^* = X^*(p)\cdot Y^*(p)$	$Z\left[X(p)\cdot Y(z)\right] = X(z)\cdot Y(z)$
$\left[x(t)*y(t)\right]^* \neq x^*(t)*y^*(t)$	$\left[X(p)\cdot Y(p)\right]^* \neq X^*(p)\cdot Y^*(p)$ $\left[X(p)\cdot Y(p)\right]^*$ noté $\overline{XY}^*(p)$	$Z\left[X(p)\cdot Y(p)\right] \neq X(z)\cdot Y(z)$ $Z\left[X(p)\cdot Y(p)\right]$ noté $\overline{XY}(z)$

Interpolation (≡ Reconstitution) : du TD au TC

Interpolation d'un signal à TD = filtrage passe-bas

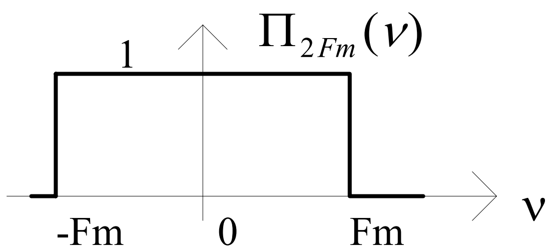
Interpolateur idéal et Bloqueur d'ordre 0



Si F ≥ 2Fm :

On peut reconstituer *x(t)* à partir de ses échantillons *x*(t)*
par filtrage de *x*(t)* avec le filtre passe-bas idéal de FT $\Pi_{2Fm}(\nu)$
(au facteur d'atténuation 1/F près)

FT du filtre interpolateur idéal



En fréquence, le filtrage s'écrit :

$$X^*(\nu) \longrightarrow \otimes \longrightarrow X(\nu) \quad \text{avec } \Pi_{2Fm}(\nu) \text{ en entrée du multiplieur}$$

ou encore :

$$X^*(\nu) \longrightarrow \boxed{\Pi_{2Fm}(\nu)} \longrightarrow X(\nu)$$

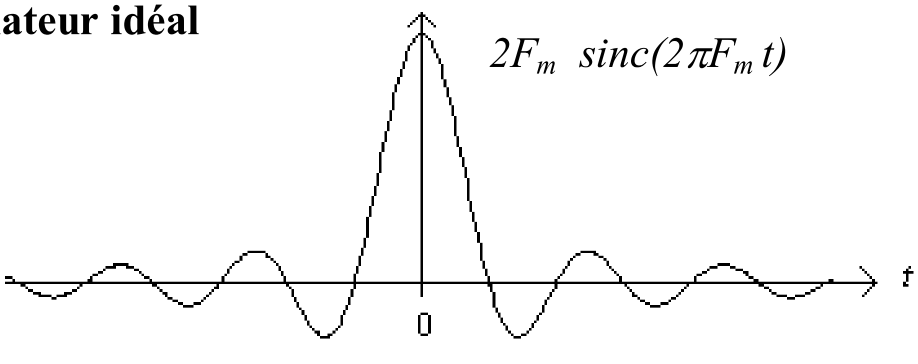
En temps, on a, sachant que $TF^{-1}[\Pi_{2Fm}(\nu)] = 2F_m \sin c(2\pi F_m t)$:

$$x^*(t) \longrightarrow \circledast \longrightarrow x(t) \quad \text{avec } 2F_m \operatorname{sinc}(2\pi F_m t) \text{ en entrée}$$

ou encore :

$$x^*(t) \longrightarrow \boxed{2F_m \operatorname{sinc}(2\pi F_m t)} \longrightarrow x(t)$$

RI du filtre interpolateur idéal



Le filtre $\Pi_{2Fm}(\nu)$ constitue l'interpolateur idéal

(Il reconstitue *x(t)* à partir de ses échantillons → interpolateur)

Remarques :

- Un filtre passe-bas de raideur infinie comme $\Pi_{2Fm}(\nu)$ n'existe pas, il ne peut être qu'approché

- L'interpolateur idéal est impossible à utiliser en Temps Réel (filtre non réalisable car non causal)

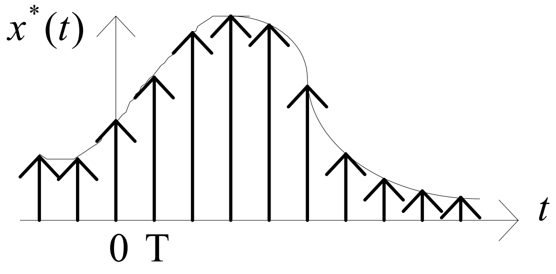
$\rightarrow$ En pratique : Interpolateur = Bloqueur d'ordre 0
(simple et causal)

Bloqueur d'ordre 0 :

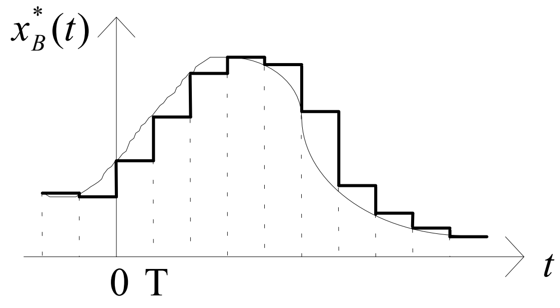
$$x(t) = x(nT + \tau) = x(nT) \quad pour : \quad 0 \leq \tau < T$$

La réponse du Bloqueur d'ordre 0 à $x^*(t)$ est notée $x_B^*(t)$:
(mémorisation de la valeur de l'échantillon pendant *T*)

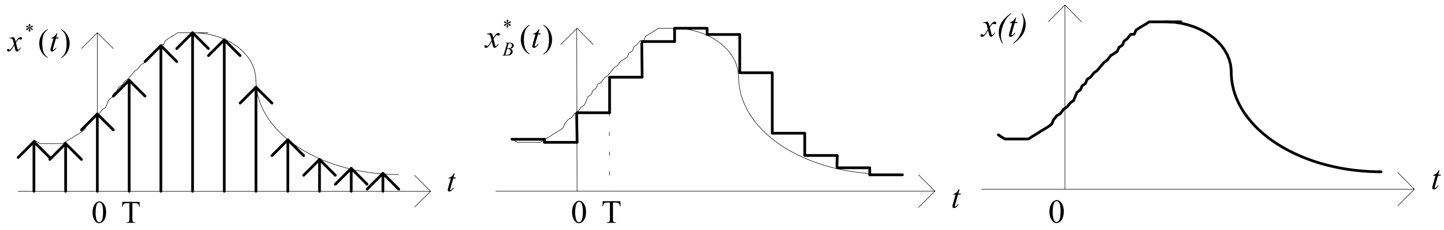
Interpolation par bloqueur d'ordre 0 :



$\underset{Bloqueur\ d'ordre\ 0}{\overset{Interpolation}{\longrightarrow}}$



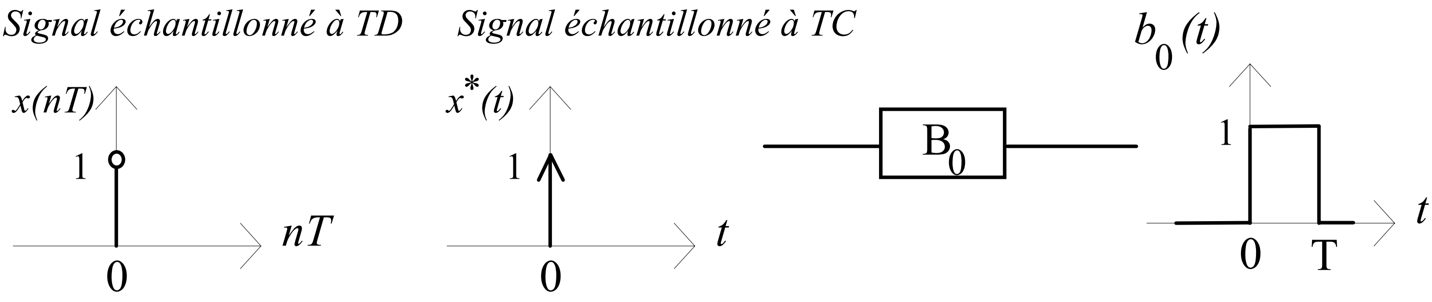
$$x^*(t) \longrightarrow \boxed{\text{Bloqueur d'ordre 0 } B_0} \xrightarrow{x^*_B(t)} \boxed{\text{Filtre passe-bas de lissage}} \longrightarrow x(t)$$



Propriétés du Bloqueur d'ordre 0 B_0 (Interpolateur réel)

- RI $b_0(t)$ (causale) *du Bloqueur* B_0 *:*

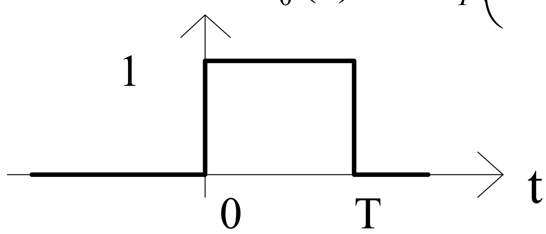
Signal échantillonné à TD *Signal échantillonné à TC*



RI du filtre interpolateur réel

$$b_0(t) = \Pi_T\left(t - \frac{T}{2}\right)$$

$$b_0(t) = \Pi_T\left(t - \frac{T}{2}\right)$$

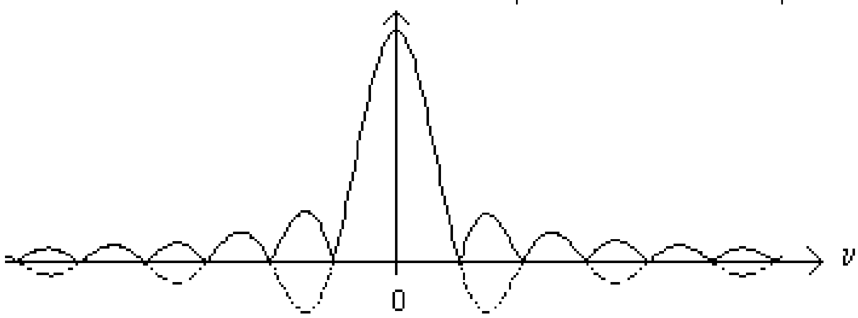


- *TL :* $$B_0(p) = TL\left[b_0(t)\right] = \int_0^T 1.e^{-pt}dt = \frac{1-e^{-pT}}{p}$$

- TF (*Spectre) :* $$TF\left[b_0(t)\right] = B_0(\nu) = \int_0^T 1.e^{-i2\pi\nu t}dt = T.e^{-i\pi\nu T}.\sin c(\nu T)$$

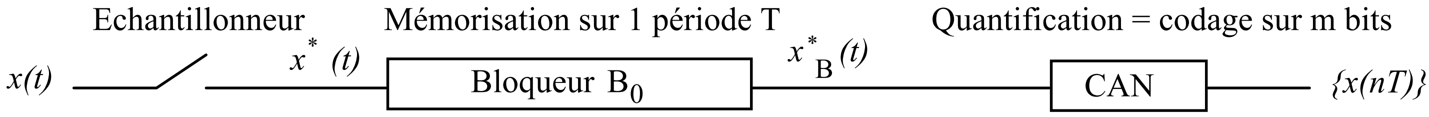
FT du filtre interpolateur réel $\left|B_0(\nu)\right| = \left|TF\left[b_0(t)\right]\right| \quad = \left|Te^{-i\pi\nu T}\sin c(\nu T)\right|$

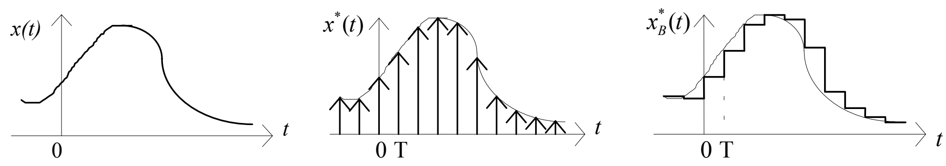
$$B_0(\nu) = Te^{-i\pi\nu T}\sin c(\nu T)$$



Chaîne d'acquisition numérique et de restitution d'un signal

- Acquisition :

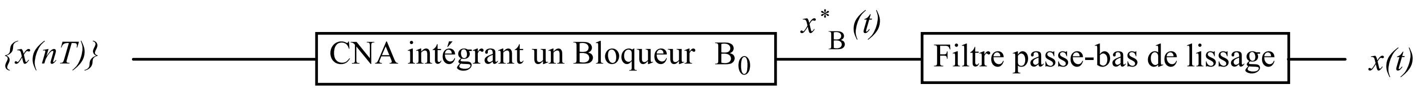




Séquence numérique *{x(nT)}*

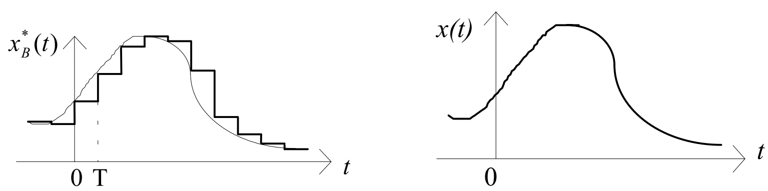
0	0	1	0	0	0	*x(0)*
0	1	1	0	1	0	*x(T)*
...						
1	0	1	0	0	0	*x(kT)*

- Restitution :



Séquence numérique *{x(nT)}*

0	0	1	0	0	0	*x(0)*
0	1	1	0	1	0	*x(T)*
...						
1	0	1	0	0	0	*x(kT)*

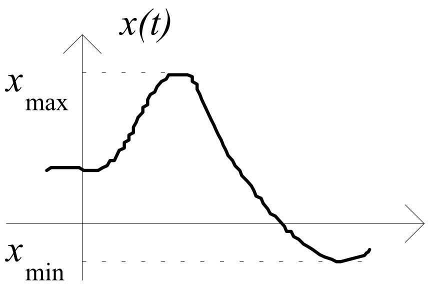


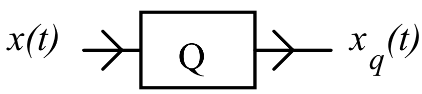
Quantification - Codage

Quantification = Bruit additif

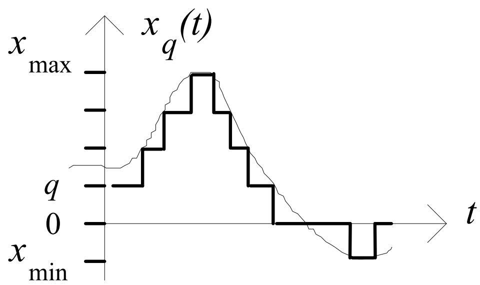
Numérisation

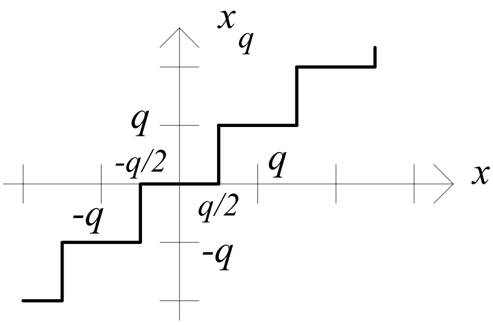
$$\text{Numérisation} \equiv \begin{cases} \text{Echantillonnage : } \quad \underset{(TC)}{x(t)} \rightarrow \underset{(TD)}{x(nT)} \\ et \\ \text{Quantification : } \quad x(nT) \rightarrow x_q(nT) \end{cases}$$



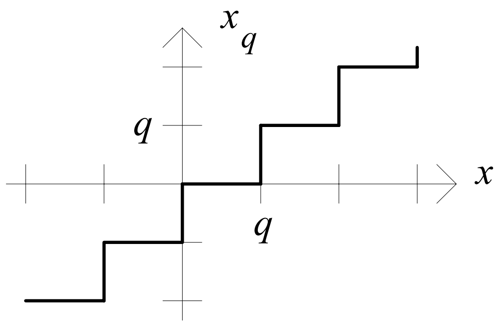


Troncature





Quantification par Arrondi



Quantification par Troncature

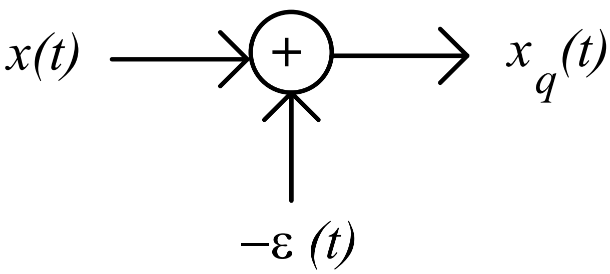
Bruit de Quantification

On peut écrire :

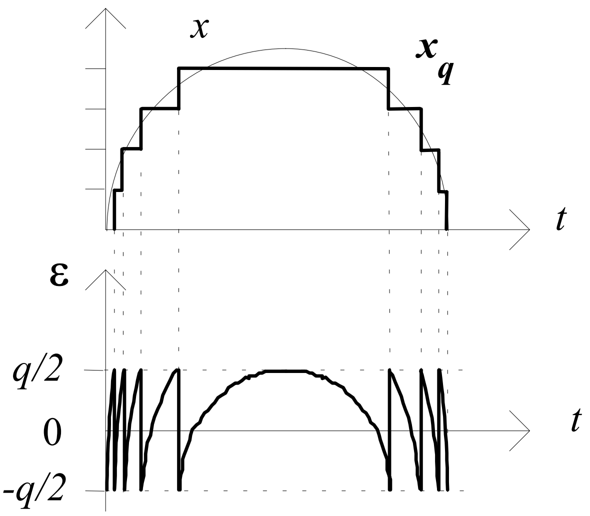
$$x(t) = x_q(t) + \varepsilon(t) \quad \rightarrow \quad x_q(t) = x(t) - \varepsilon(t)$$

$$\varepsilon(t) = x(t) - x_q(t)$$: Bruit de quantification

Bruit Additionnel

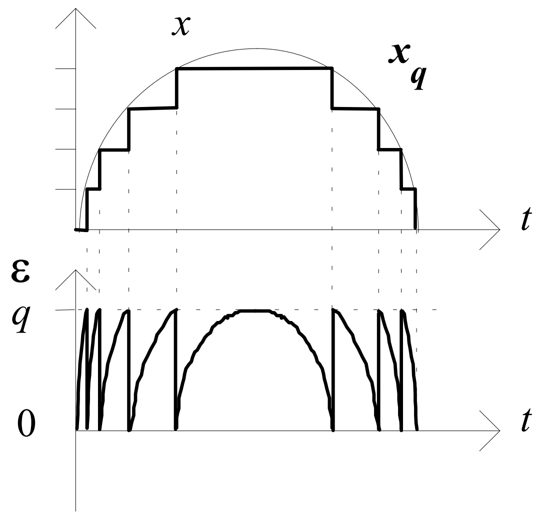


Quantification par Arrondi



$$-\frac{q}{2} \leq \varepsilon \leq \frac{q}{2}$$

Quantification par Troncature



$$0 \leq \varepsilon \leq q$$

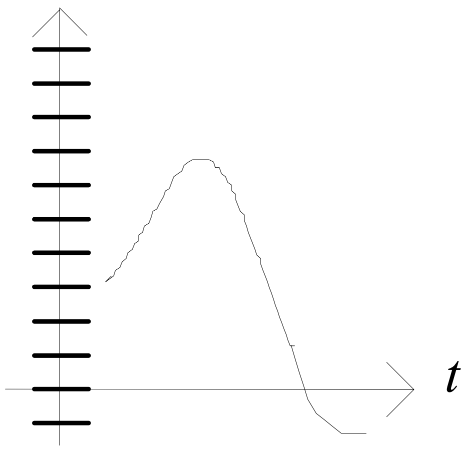
Codage

Après échantillonnage et quantification :

$x_q(nT)$ est représenté par une séquence de valeurs numériques

Codage binaire linéaire

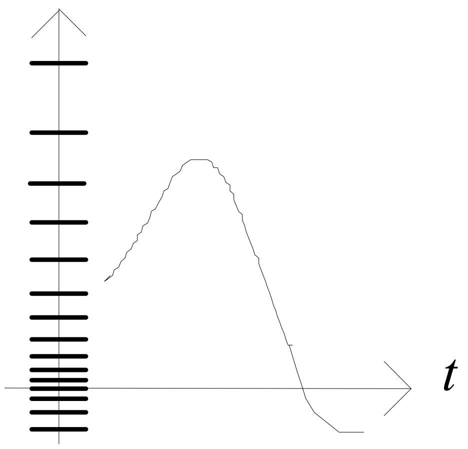
Pas de quantification constant avec l'amplitude du signal



Codage non linéaire

Car les signaux ne sont pas souvent à leurs valeurs extrêmales

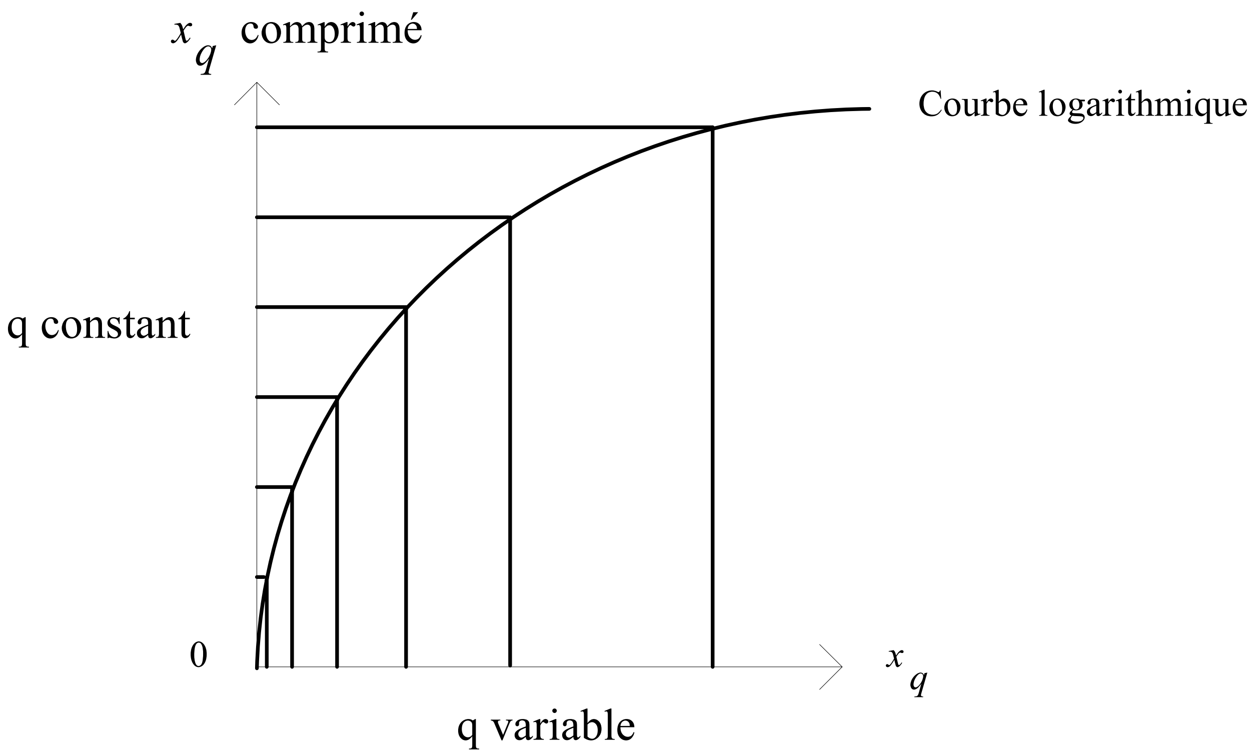
Pas de quantification augmentant avec l'amplitude du signal



- Variation de q exponentielle

- Coder x_q avec q exponentiel $\equiv$

comprimer x_q sur les faibles amplitudes et le quantifier avec q constant :



Exemple : Téléphone :