

# Echantillonnage

---

- construire un contrôleur continu et l'échantillonner ;
- échantillonner le système à contrôler et construire un contrôleur échantillonné (**théorie de la commande échantillonnée**).

On a étudié l'erreur d'échantillonnage (liée au schémas d'intégration à pas constant)

On veut étudier la possibilité de reconstruire le signal continu original à partir du signal échantillonné.

# Echantillonnage

---

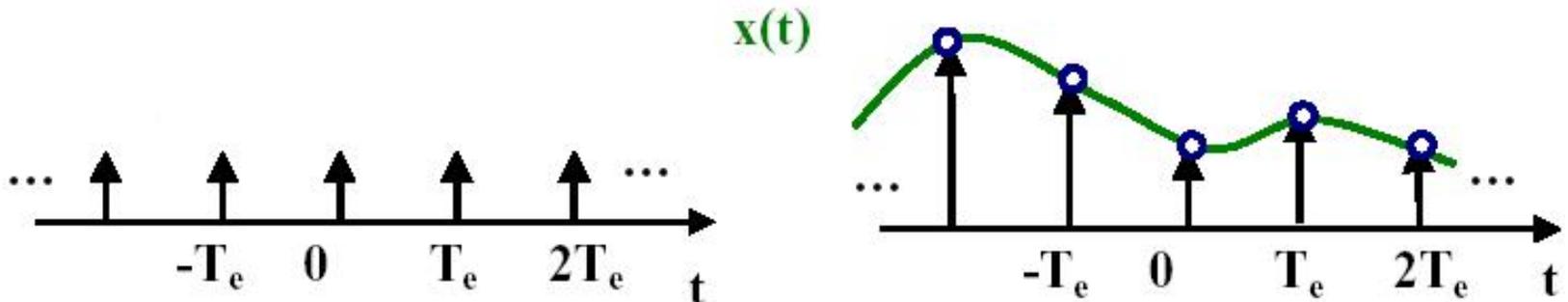
Une séquence  $(x_n)_{n \in \mathbb{Z}}$  provient d'un échantillonnage d'un signal continu  $x(t)$  :

$$x_n = x(nT_e)$$

$T_e$  est la **période d'échantillonnage**,  $F_e = \frac{1}{T_e}$  est la **fréquence d'échantillonnage**.

Le signal échantillonné est :

$$x_d(t) = \sum_{n \in \mathbb{Z}} x(nT_e) \delta(t - nT_e)$$



# Signal à bande limitée

---

## Signal à bande limitée

Un signal continu  $x(t)$  est à bande limitée sur  $[-B, B]$  si le support de sa transformée Fourier est inclu dans  $[-B, B]$ , c-a-d :

$$\forall f \in, |f| > B \Rightarrow \mathcal{F}x(f) = X(f) = 0$$

**Fréquence Nyquist** d'un signal continu  $x$  :

Soit  $F_{max}$  est la plus petite fréquence (lorsqu'elle existe) telle que  $x$  soit à bande limitée sur  $[-F_{max}, F_{max}]$ . Alors, la **fréquence Nyquist** de  $x$  est  $2F_{max}$ .

# Théorème de Shannon-Nyquist

---

Soit  $x(t)$  un signal continu à bande limitée, sa fréquence Nyquist est  $2F_{max}$ .

Alors un échantillonnage de  $x$  à la fréquence  $F_e$  permet la reconstruction complète du signal  $\{x(t)\}_{t \in \mathbb{R}}$  à condition que  $F_e/2 \geq F_{max}$ .

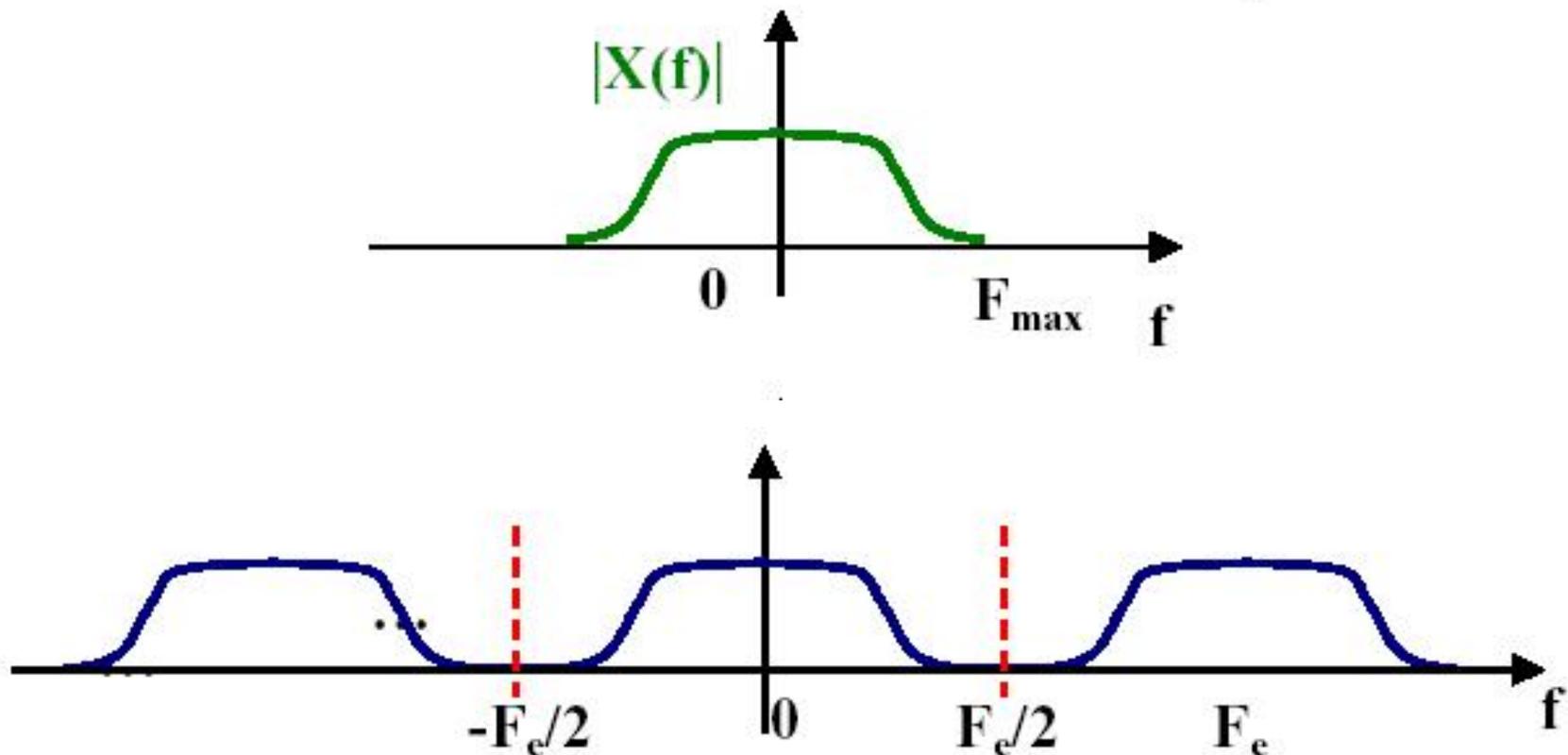
**Preuve :** On utilise le signal  $x_d(t)$

- D'abord, on examine la relation entre la transformée de Fourier  $X_d(f)$  de  $x_d(t)$  et celle  $X(f)$  de  $x(t)$
  - Puis, on montre qu'il est possible de reconstruire  $X(f)$  à partir de  $X_d(f)$ .
-

## Relation entre $X_d(f)$ et $X(f)$ (suite)

---

Le spectre  $X_d(f)$  du signal échantillonné  $x_d(t)$  est  $F_e$ -**periodique** et correspond à la **périodisation** du spectre  $X(f)$  du signal continu  $x(t)$ .



# Reconstruction de $X(f)$ à partir de $X_d(f)$ \_\_\_\_\_

$$X_d(f) = F_e \sum_{n \in \mathbb{Z}} X(f - nF_e).$$

Supposons que  $F_e/2 \geq F_{max}$ . Donc,  $\forall f, |f| \geq F_e/2 \Rightarrow X(f) = 0$ .

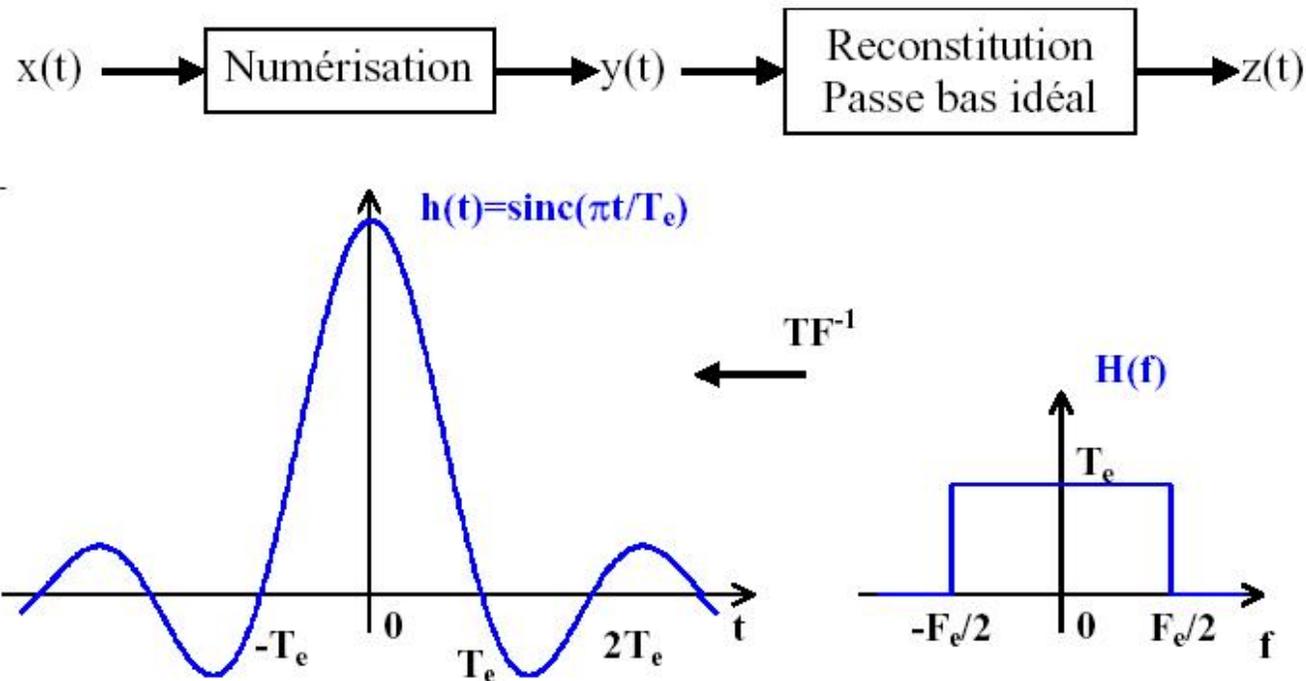
Dans ce cas, la restriction de  $X_d(f)$  à  $[-F_e/2, F_e/2]$  est exactement  $X(f)$  au facteur  $F_e$  près.

La reconstruction de  $X(f)$  peut être résumée par :

$$X(f) = T_e \text{Rect}_{[-F_e/2, F_e/2]}(f) X_d(f)$$

# Filtre passe bas idéal

---



Le filtre n'est pas causal et n'a pas un support fini.

Pour interpolier parfaitement une valeur  $x(t)$ , il faut garder en mémoire une infinité de valeurs  $(x_n) \Rightarrow$  pas réalisable en pratique.

---

# Repliement spectral

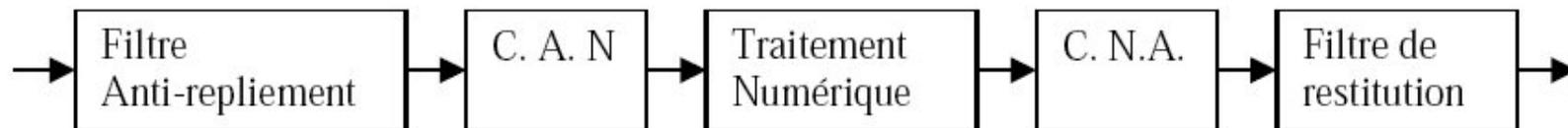
---

Quand la fréquence d'échantillonnage est trop faible ( $F_e/2 < F_{max}$ ), les conditions du théorème de Shannon-Nyquist ne sont plus respectées.

On a trouvé :  $X_d(f) = F_e \sum_{n \in \mathbb{Z}} X(f - nF_e)$ .

**Phénomène de repliement spectral** : Pour  $f \leq F_e/2$  la relation  $X(f) = T_e X_d(f)$  n'est plus valide, car la somme ci-dessus peut englober plusieurs termes non-nuls au lieu d'un seul  $\Rightarrow$  repliement des composantes hautes fréquences de  $X(f)$  et le signal  $x_r = h * s_d$  n'est pas conforme au signal original  $x$ .

Quand un signal a une fréquence de Nyquist trop grande  $F_{max} > F_e/2$ , on filtre  $x(t)$  pour éliminer des composantes fréquentielles au delà de  $F_e/2$ . Ce filtre est appelé **filtre anti-repliement**



# Filtres de restitution

---

Si  $h$  est la réponse impulsionnelle du filtre de reconstruction, on a :

$$x_r(t) = \sum_n x_n h(t - nT_e)$$

Si les conditions de Shannon sont respectées,  $X(f) = T_e \frac{\text{Rect}_{[-F_e/2, F_e/2]}(f)}{H(f)} X_r(f)$

Lorsque  $H(f)$  est une bonne approximation de  $T_e \text{Rect}_{[-F_e/2, F_e/2]}(f)$ ,  $x_r$  est "proche" de  $x$ .

$H$  doit couper toute les fréquences supérieures à  $F_e/2$ .

---

# Filtres de restitution

---

- **Bloqueur d'ordre zero** :  $h_{boz}(t) = \text{Rect}_{[0, T_e]}(t)$

$$H_{boz}(f) = T_e \exp(-i\pi f T_e) \text{sinc}(\pi f T_e).$$

Les composantes de fréquence supérieure à  $F_e/2$  sont atténuées mais ne sont pas totalement annulées.

- **Interpolation linéaire** (blocage d'ordre 1) :  $h_{lin}(t) = \text{Tri}(t/T_e)$  avec

$$\text{Tri}(t) = (1 - |t|) \text{Rect}_{[-1, 1]}(t).$$

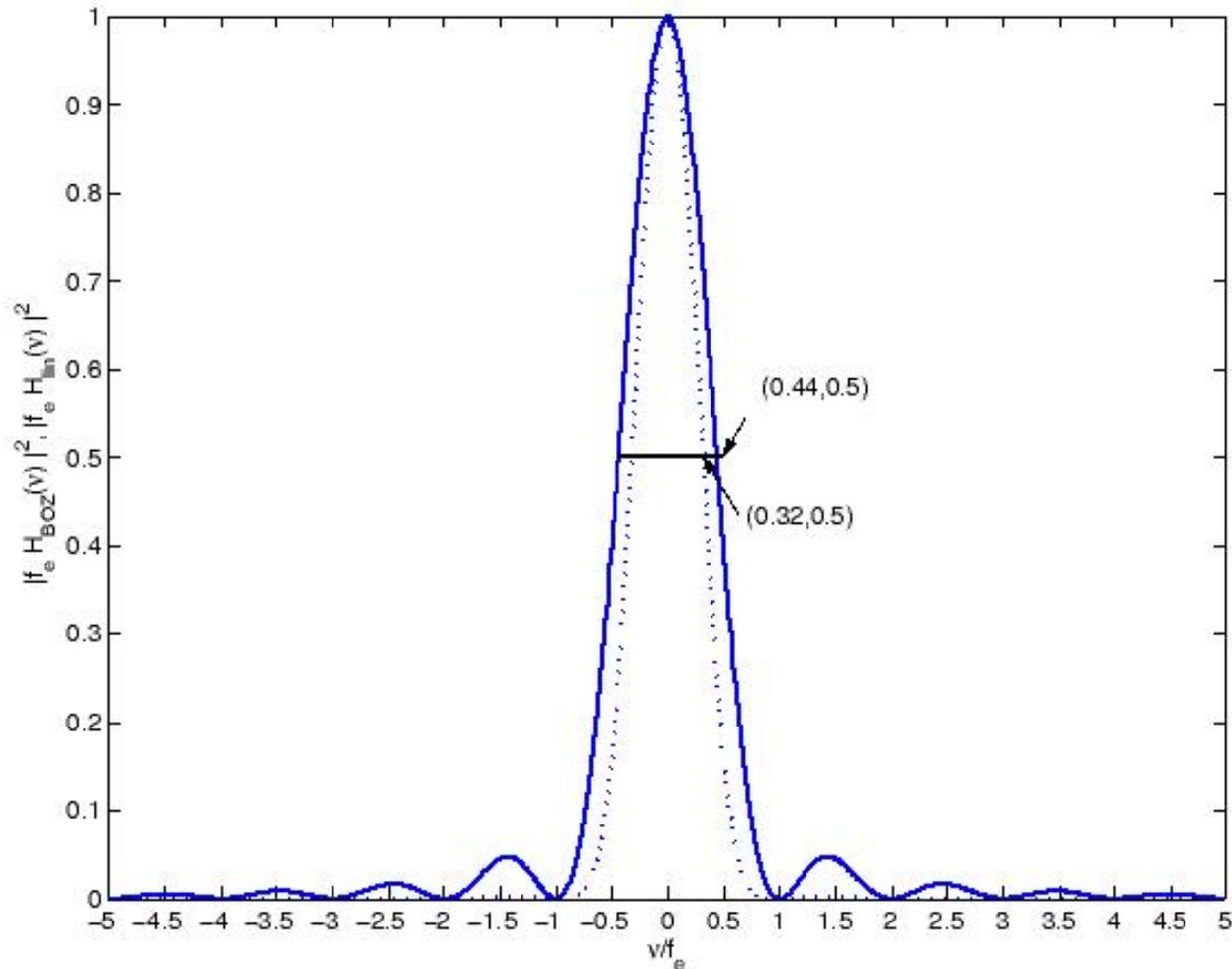
$$H_{lin}(f) = T_e \text{sinc}^2(\pi f T_e).$$

Forte atténuation des fréquences indésirables et une bande passante plus étroite  $\Rightarrow$  meilleure qualité que le bloqueur d'ordre 0.

---

# Filtres de restitution

---



*Spectres d'énergie des filtres de reconstruction BOZ (gras) et linéaire (pointillés).*

---

# Application to LEGO robots

---

The distance controller of type P needs not be discretized. We need to discretize only the

Compute the continuous-time transfer function of the open loop  $F_{BO}(s)$  for the orientation  $\theta$

Estimate the crossover frequency  $\omega_c$  of  $F_{BO}(s)$  using the function "margin" in matlab.

Choose sampling period  $T$  according to the rule :  $\omega_c T$  is **between 0.05 and 0.14**

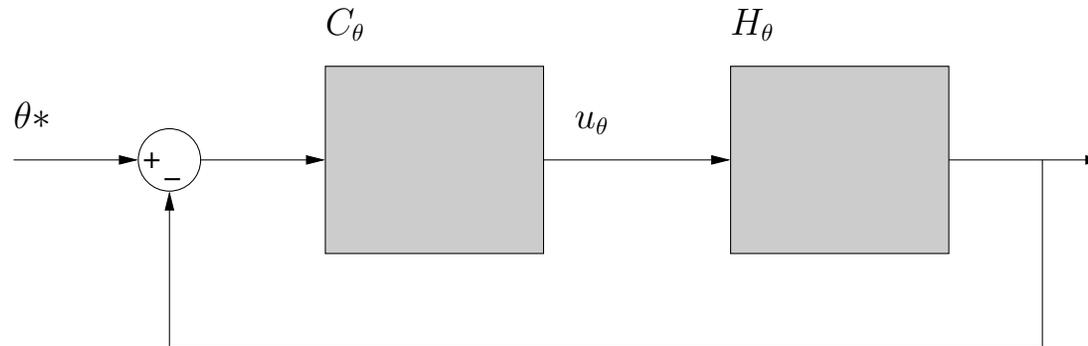
Discretize the controllers using one of the approximation methods and the chosen sampling period.

Note that instead of replacing the whole continuous-time controller with its discretized version, we can replace only its components containing continuous-time blocs, such as the integrators  $\frac{1}{s}$

---

# Commande de l'orientation du robot

---



Pour un correcteur de type PI nous avons :  $C_\theta(s) = k_{p\theta} + \frac{k_{i\theta}}{s}$

La fonction de transfert en boucle fermée

$$H(s) = \frac{C_\theta(s)H_\theta(s)}{1 + C_\theta(s)H_\theta(s)} = \frac{\frac{k_{p\theta}}{l}s + \frac{k_{i\theta}}{l}}{s^2 + \frac{k_{p\theta}}{l}s + \frac{k_{i\theta}}{l}}$$

# Robot Controller Discretization - Determining a sampling period

---

The

```
%la fonction de transfert C(s) = ki_teta/s + kp_teta
%la fonction de transfert H_teta(s) = 1/(1s)
%la fonction de transfert en boucle ouvert
%H_BO(s) = (kp_teta*s + ki_teta)/(1s^2 + 1s)

sysbo = tf([0 kp_teta ki_teta], [ 1 1 0])

[g p f wc] = margin(sysbo)
%wc est la "crossover frequency"

%T = periode d'echantillonnage
T=0.05/wc
```