MU4RBI08 / MU4MEA05
Traitement du signal audio

Henri Boutin
boutin@ircam.fr

2021/2022

Plan

- 1- Introduction : contexte et objectifs

- 2- Chaine de traitement d'un signal sonore
  - Rappels
  - Acquisition : le CAN
  - Restitution/reconstruction : le CNA
  - Quantification

- 3- Analyse en fréquence des signaux discrets
  - Rappels : définitions et propriétés
  - Analyse temps-fréquence
    - Principe d'incertitude
    - Transformée de Fourier à Court Terme

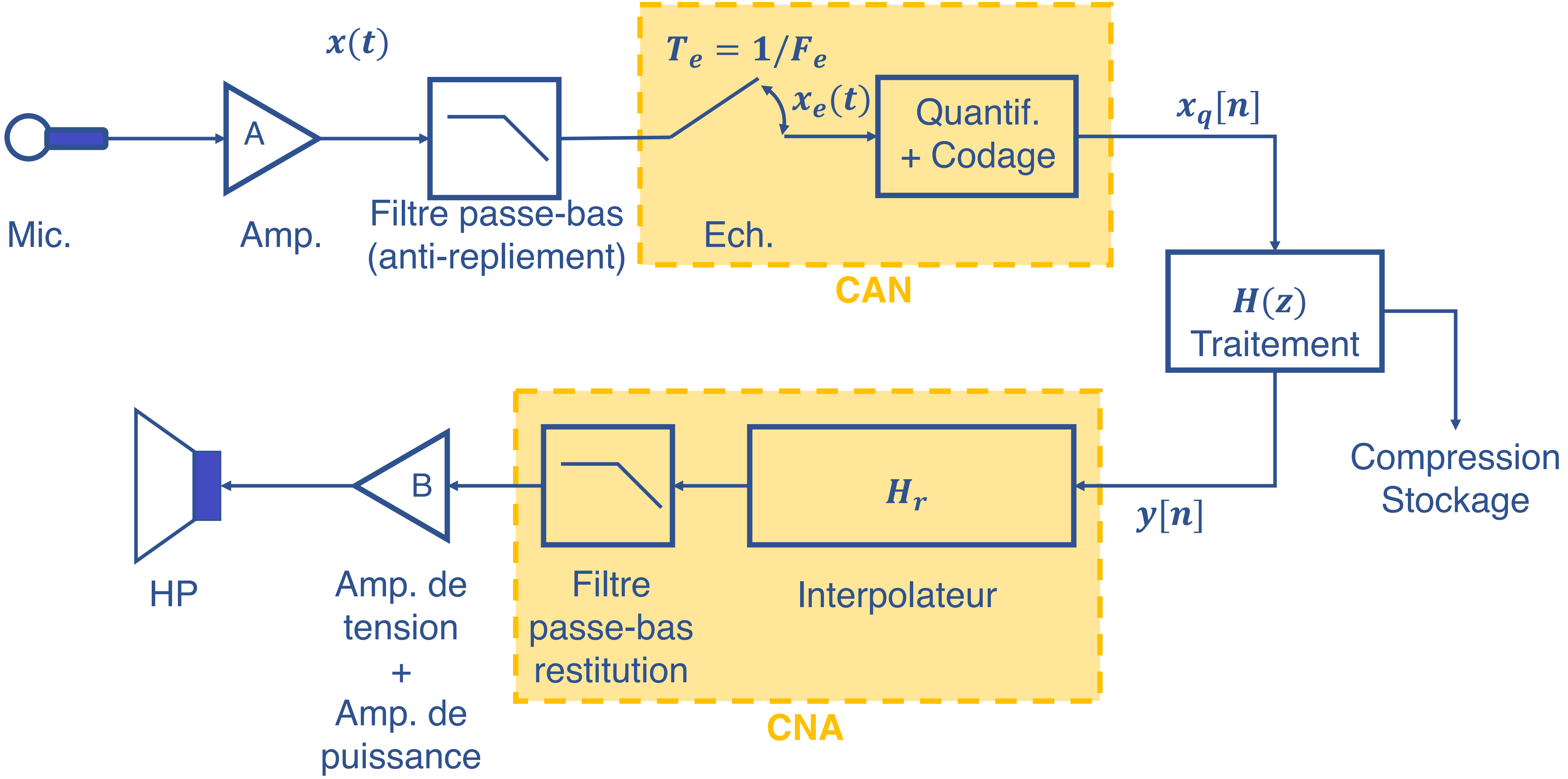
Plan

- 1- Introduction : contexte et objectifs

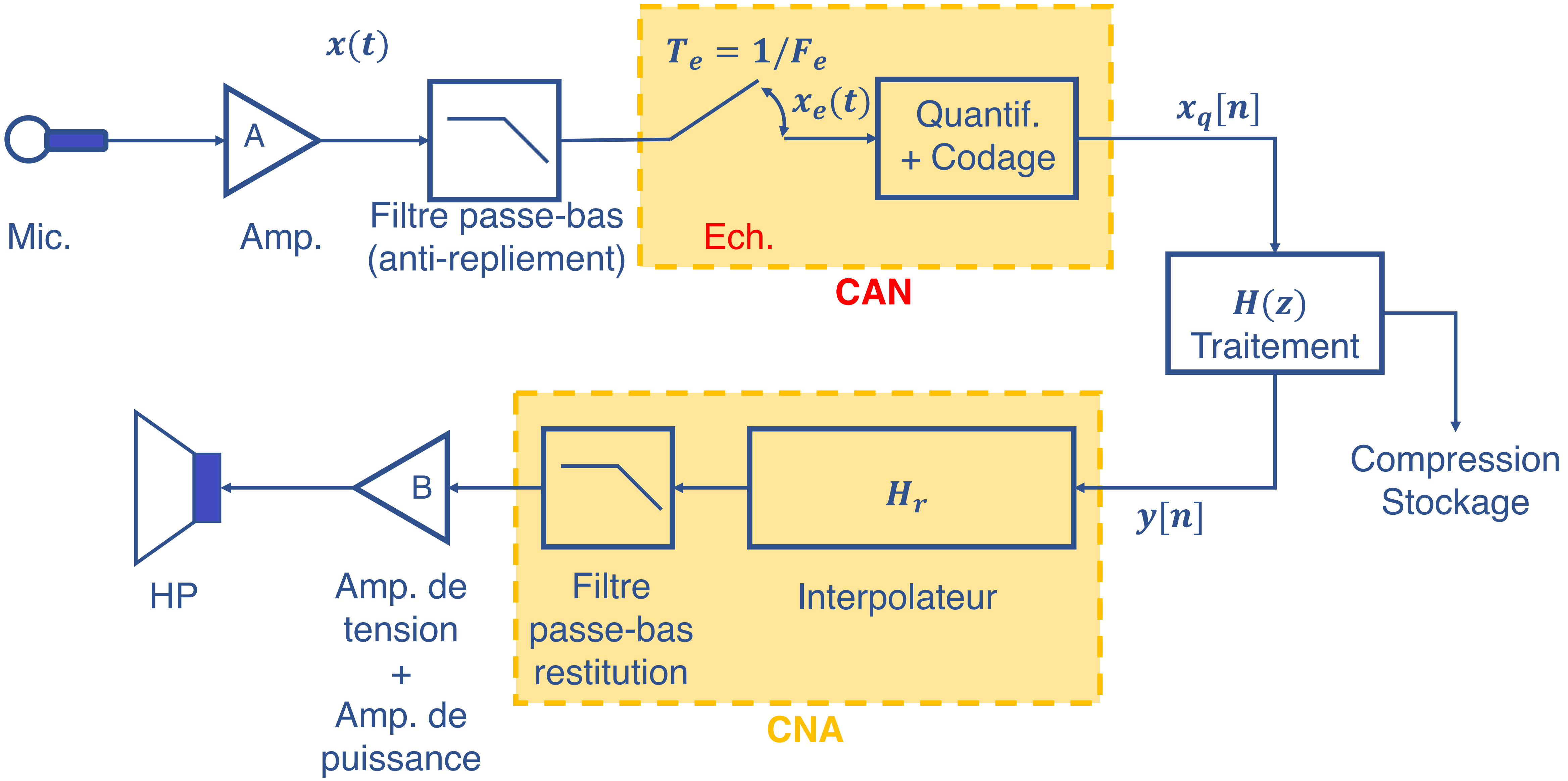
- 2- Chaine de traitement d'un signal sonore
  - Rappels
  - Acquisition : le CAN
  - Restitution/reconstruction : le CNA
  - Quantification

- 3- Analyse en fréquence des signaux discrets
  - Rappels : définitions et propriétés
  - Analyse temps-fréquence
    - Principe d'incertitude
    - Transformée de Fourier à Court Terme

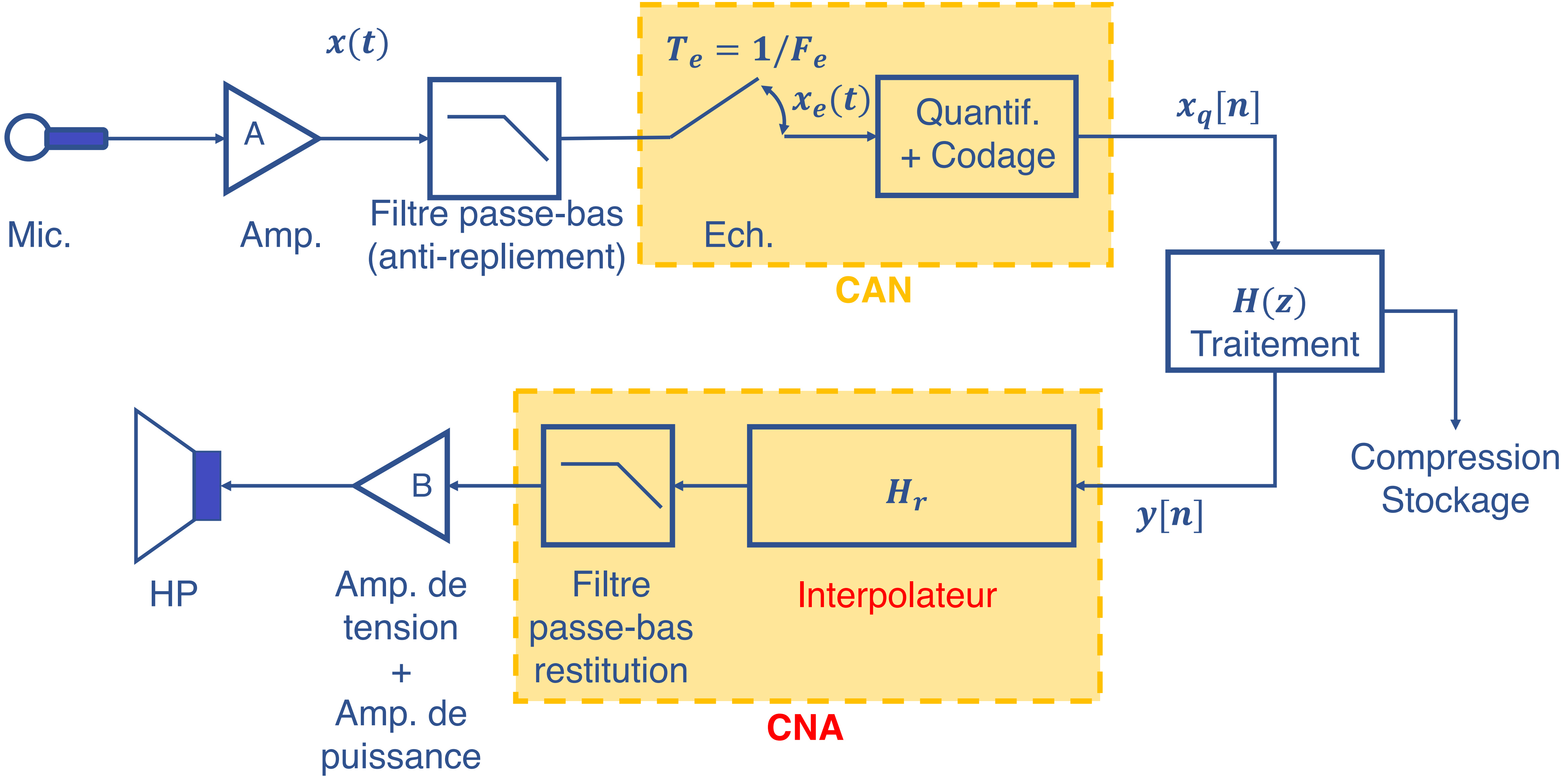
RAPPEL : Chaine d'acquisition et de traitement d'un signal sonore



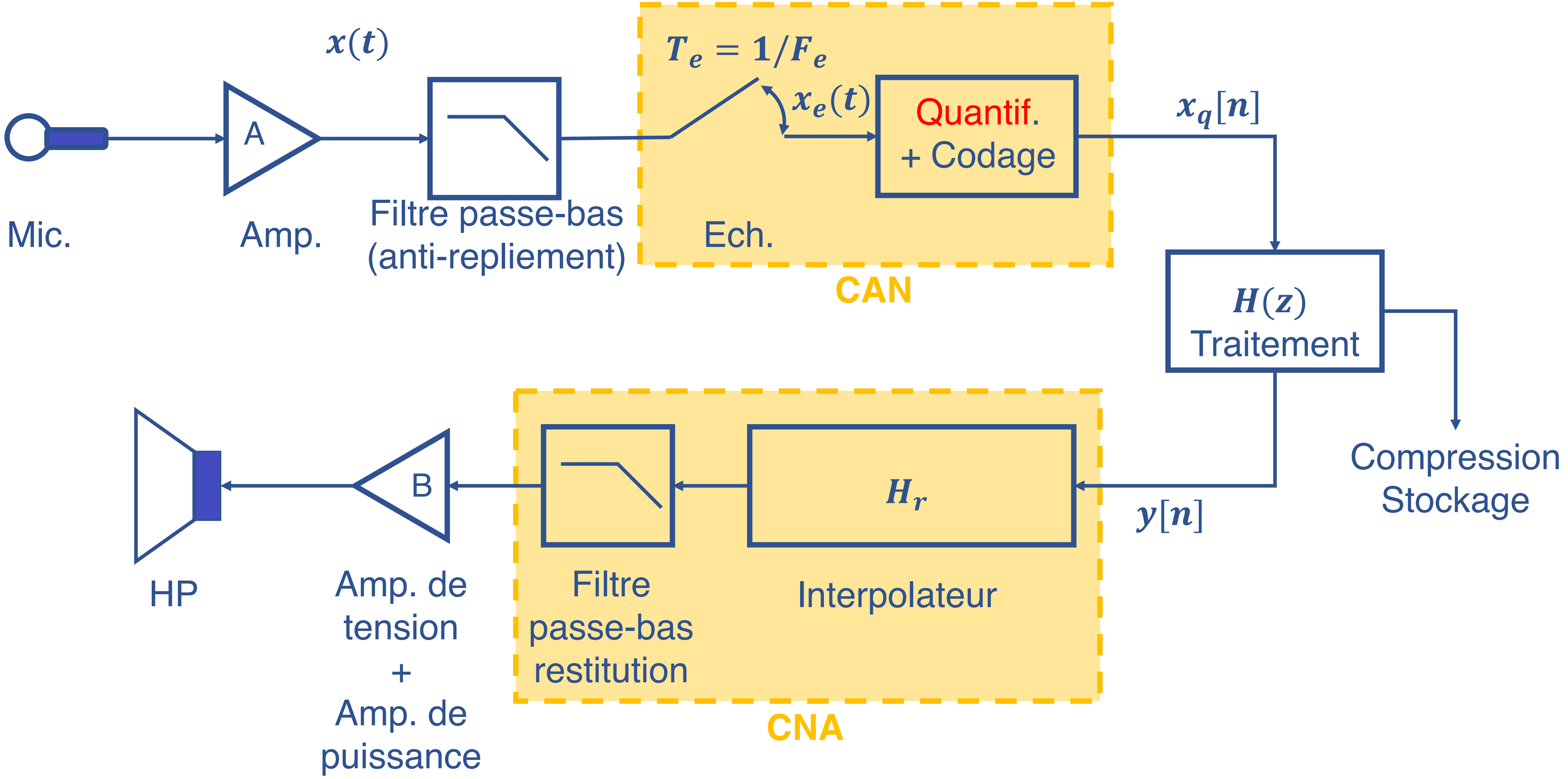
RAPPEL : Chaine d'acquisition et de traitement d'un signal sonore



RAPPEL : Chaine d'acquisition et de traitement d'un signal sonore



RAPPEL : Chaine d'acquisition et de traitement d'un signal sonore



2.1 Rappels

- Impulsion de Dirac : $\delta(t)$

Définition: $\delta(t) = \lim_{T \to 0} \Pi_T\,(t)$

Propriétés : $\int_{\mathbb{R}} \delta(t)\phi(t)\,dt =$

$\int_{\mathbb{R}} \delta_{t_0}(t)\phi(t)\,dt =$

$\phi(t)\delta(t) =$ et $\phi(t)\delta_{t_0}(t) =$

$\phi(t) * \delta(t - t_0) =$

$TF\left[\delta_{t_0}(t)\right] =$

$TF\left[e^{2j\pi f_0 t}\right] =$

- Train d'impulsions de Dirac : $Ш_{T_e}(t)$

$TF\left[Ш_{T_e}\right] =$

- Représentation :

2.2 Acquisition : le CAN: RAPPELS

- **Echantillonnage:**

  Domaine temporel : $x_e(t) = \sum_{n \in Z} x(nT_e)\delta(t - nT_e)$

2.2 Acquisition : le CAN: RAPPELS

- Echantillonnage:

  Domaine temporel : $x_e(t) = \sum_{n \in Z} x(nT_e)\delta(t - nT_e)$

  Domaine fréquentiel : $X_e(f) = F_e \sum_{k \in Z} X(f - kF_e)$

2.2 Acquisition : le CAN: RAPPELS

➢ Echantillonnage:

Domaine temporel : $x_e(t) = \sum_{n \in Z} x(nT_e)\delta(t - nT_e)$

Domaine fréquentiel : $X_e(f) = F_e \sum_{k \in Z} X(f - kF_e)$

Théorème de Shannon : échantillonnage sans perte d'information ⇔ $F_e \geq 2F_{max}$, sinon : repliement spectral

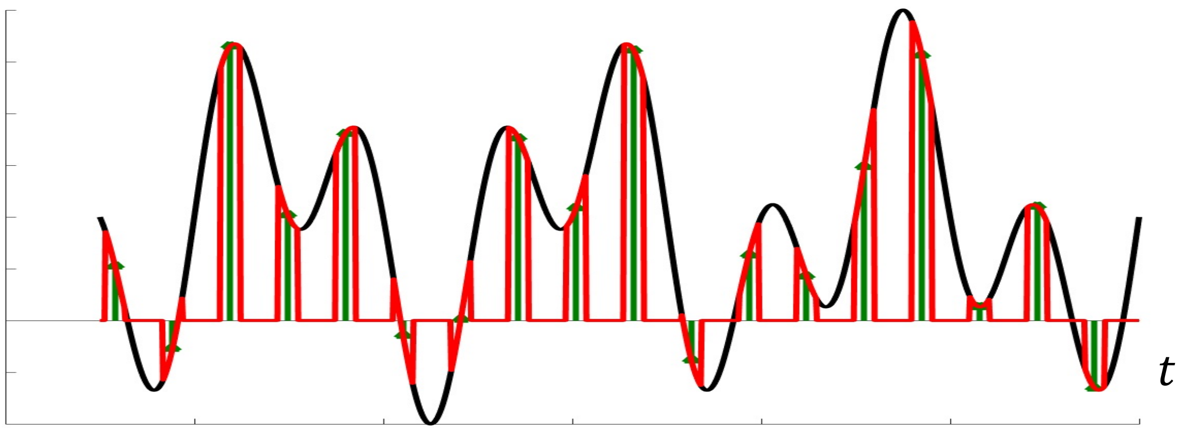
2.2 Acquisition : le CAN: RAPPELS

- **Echantillonnage:**
  - Domaine temporel : $x_e(t) = \sum_{n\in Z} x(nT_e)\delta(t - nT_e)$
  - Domaine fréquentiel : $X_e(f) = F_e \sum_{k\in Z} X(f - kF_e)$
  - Théorème de Shannon : échantillonnage sans perte d'information $\Leftrightarrow F_e \geq 2F_{max}$, sinon : repliement spectral

- **Echantillonnage réel :**
  - domaine temporel : $x_{en}(t) = x(t) \times r(t)$

Domaine temporel:

$\boldsymbol{x(t)}$ $\boldsymbol{x_e(t)}$ $\boldsymbol{x_{en}(t)}$



2.2 Acquisition : le CAN: RAPPELS

- **Echantillonnage:**

  Domaine temporel : $x_e(t) = \sum_{n\in Z} x(nT_e)\delta(t - nT_e)$

  Domaine fréquentiel : $X_e(f) = F_e \sum_{k\in Z} X(f - kF_e)$

  Théorème de Shannon : échantillonnage sans perte d'information $\Leftrightarrow F_e \geq 2F_{max}$, sinon : repliement spectral

- **Echantillonnage réel :**

  domaine temporel : $x_{en}(t) = x(t) \times r(t)$

  domaine fréquentiel: $X_{en}(f) = \epsilon \sum_{k\in Z} \mathrm{sinc}(\pi k\epsilon)\, X_e(f - kF_e)$

2.2 Acquisition : le CAN: RAPPELS

- **Echantillonnage:**
  Domaine temporel : $x_e(t) = \sum_{n \in Z} x(nT_e)\delta(t - nT_e)$
  Domaine fréquentiel : $X_e(f) = F_e \sum_{k \in Z} X(f - kF_e)$
  Théorème de Shannon : échantillonnage sans perte d'information $\Leftrightarrow F_e \geq 2F_{max}$, sinon : repliement spectral

- **Echantillonnage réel :**
  domaine temporel : $x_{en}(t) \; = x(t) \times r(t)$
  domaine fréquentiel: $X_{en}(f) = \epsilon \sum_{k \in Z} \mathrm{sinc}(\pi k \epsilon)\, X_e(f - kF_e)$

SORBONNE UNIVERSITÉ CRÉATEURS DE FUTURS DEPUIS 1257

MASTER Spi

2.2 Acquisition : le CAN: RAPPELS

- **Echantillonnage:**

  Domaine temporel : $x_e(t) = \sum_{n\in Z} x(nT_e)\delta(t - nT_e)$

  Domaine fréquentiel : $X_e(f) = F_e \sum_{k\in Z} X(f - kF_e)$

  Théorème de Shannon : échantillonnage sans perte d'information $\Leftrightarrow F_e \geq 2F_{max}$, sinon : repliement spectral

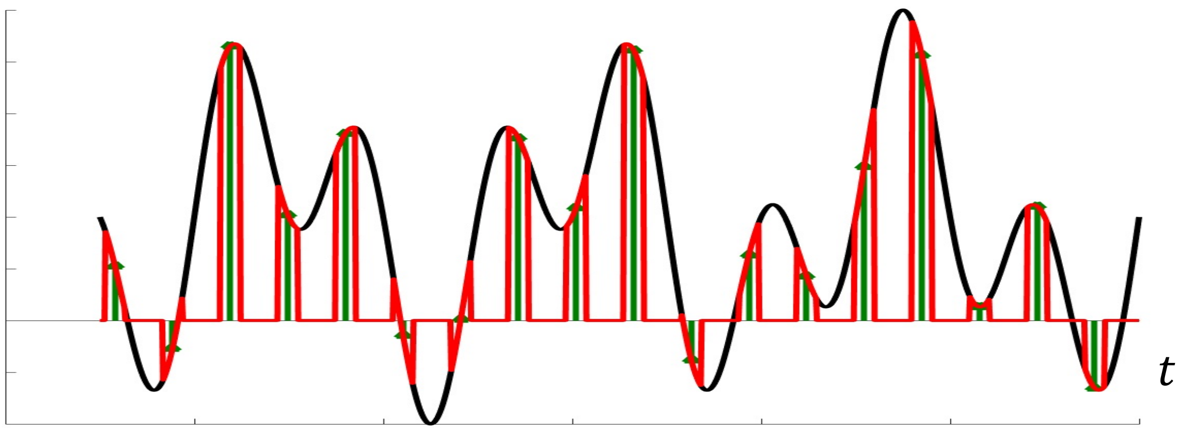
- **Echantillonnage réel :**

  domaine temporel : $x_{en}(t) = x(t) \times r(t)$

  domaine fréquentiel: $X_{en}(f) = \epsilon \sum_{k\in Z} \text{sinc}(\pi k\epsilon)\, X_e(f - kF_e)$

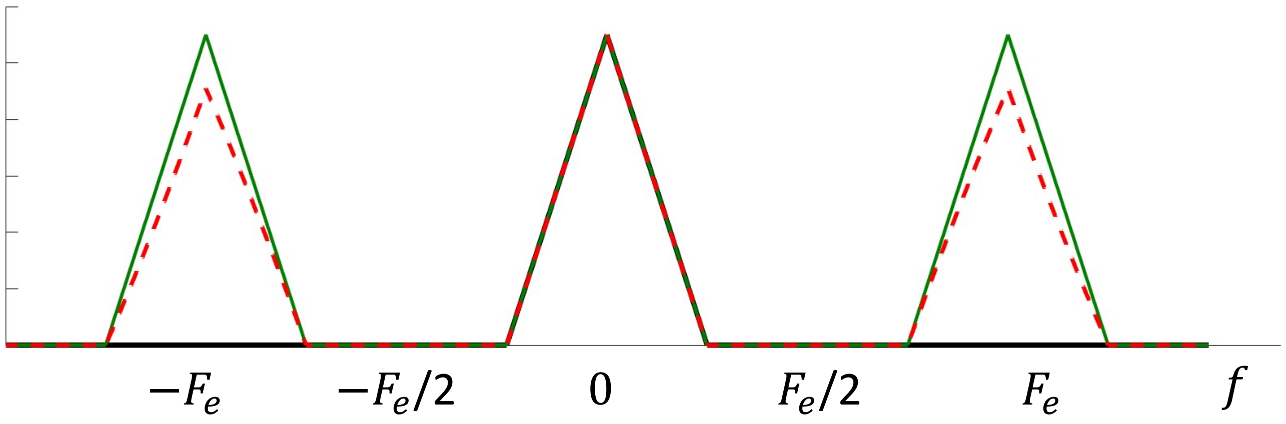
Domaine temporel:

$\boldsymbol{x(t)}$ $\boldsymbol{x_e(t)}$ $\boldsymbol{x_{en}(t)}$

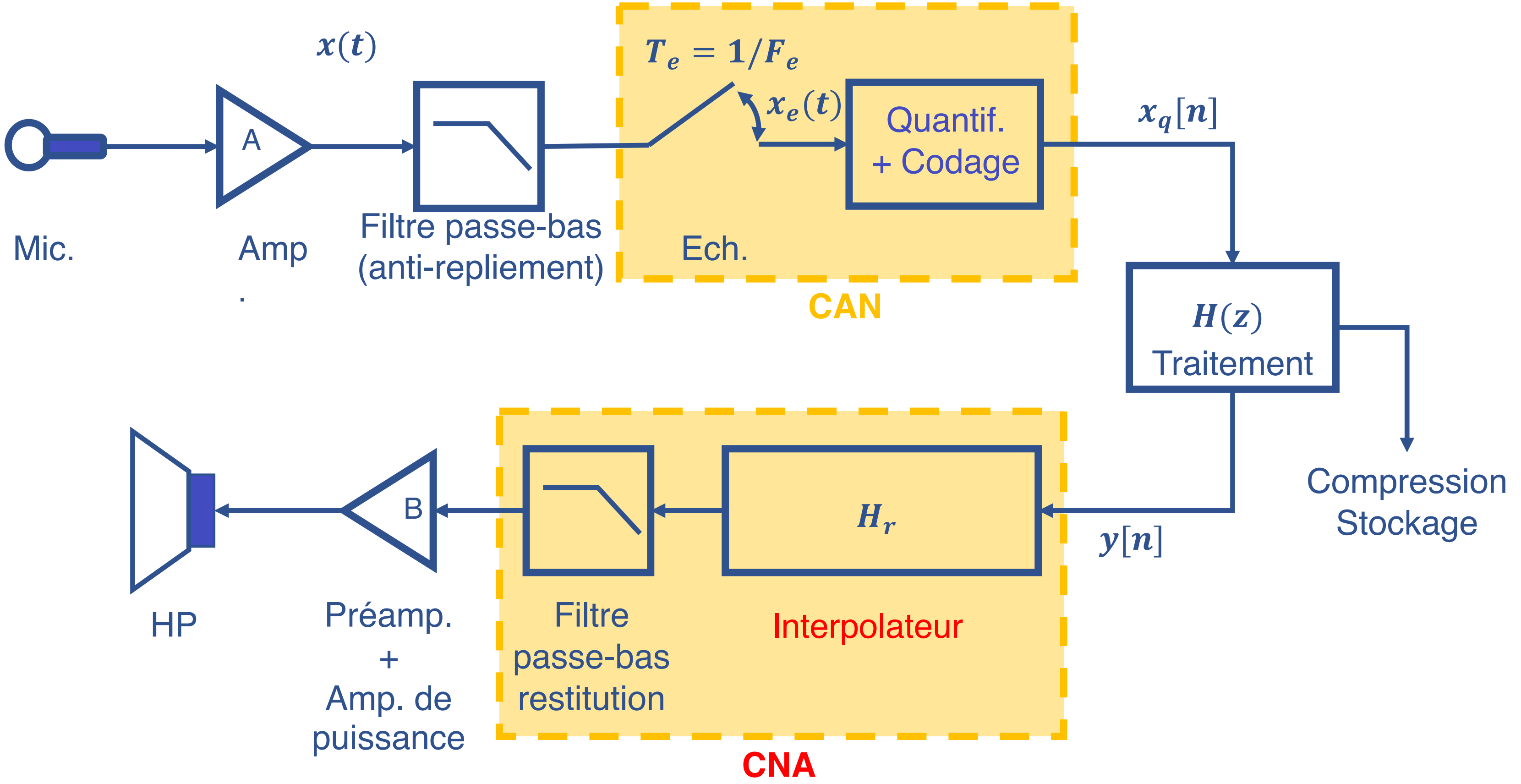


Domaine fréquentiel:

$|\boldsymbol{X(f)}|$ $|\boldsymbol{X_e(f)}|/\boldsymbol{F_e}$ $|\boldsymbol{X_{en}(f)}|/\epsilon$



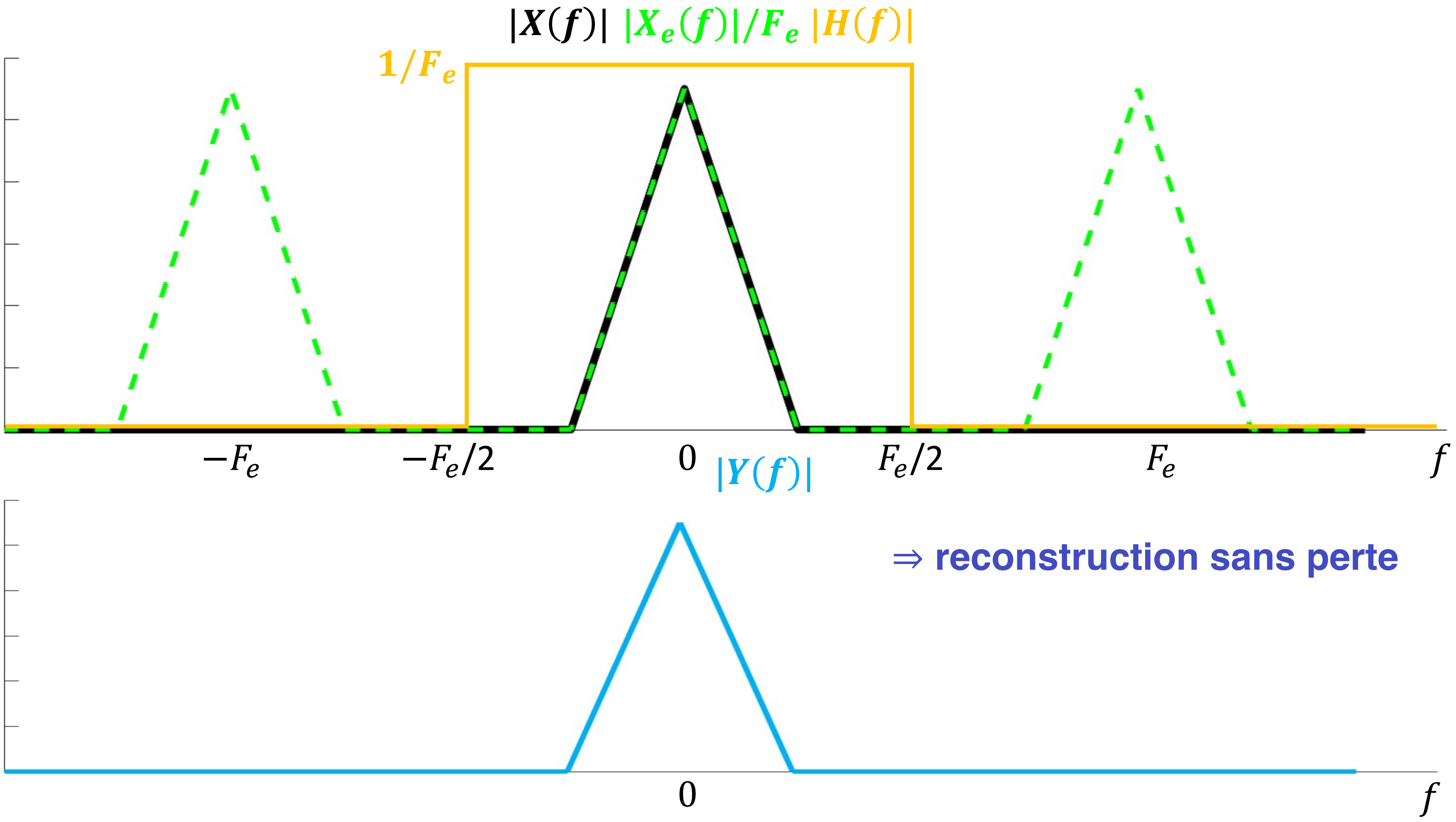
2.3 Restitution / Reconstruction



Chaine d'acquisition et de traitement d'un signal sonore

2.3 Restitution / Reconstruction

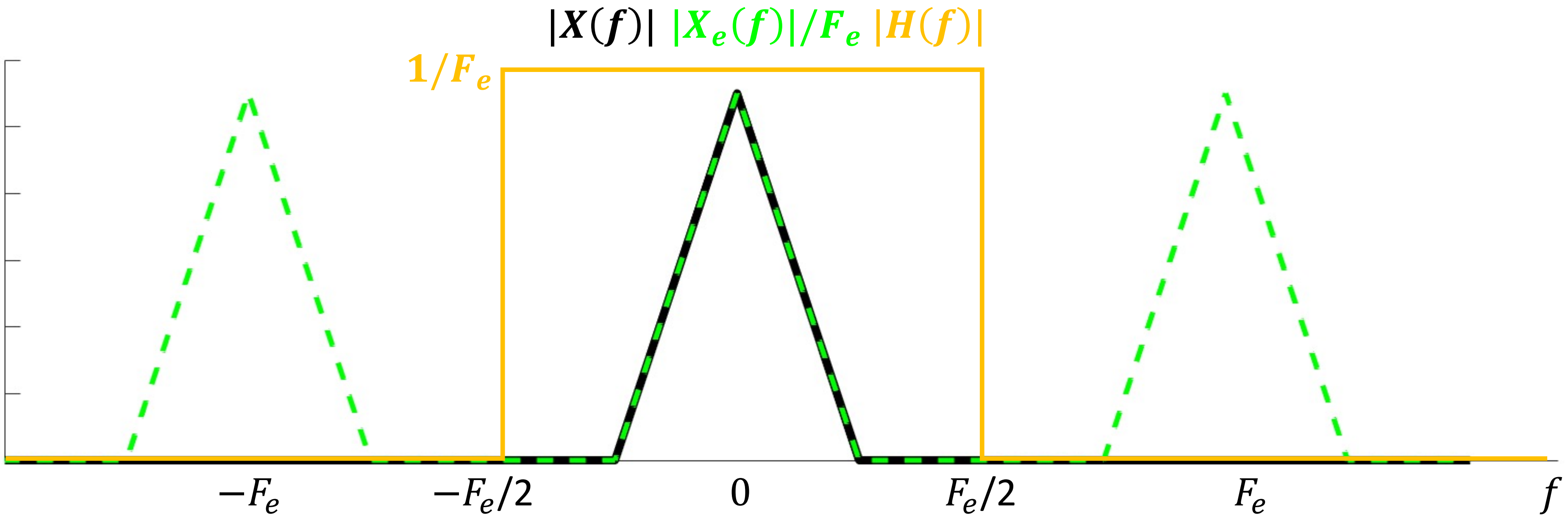
- <u>Solution1</u> : Filtre « cardinal » : ***domaine fréquentiel***



⇒ reconstruction sans perte

2.3 Restitution / Reconstruction

➢ Solution1 : Filtre « cardinal » : ***domaine fréquentiel***



i.e. $Y(f) = X_e(f) \times H(f)$ avec

$$H(f) = \Pi_{F_e}(f) \begin{cases} 1/F_e \text{ sur } \left[-\frac{F_e}{2}, \frac{F_e}{2}\right], \\ 0 \text{ ailleurs} \end{cases} \Rightarrow h(t) =$$

2.3 Restitution / Reconstruction

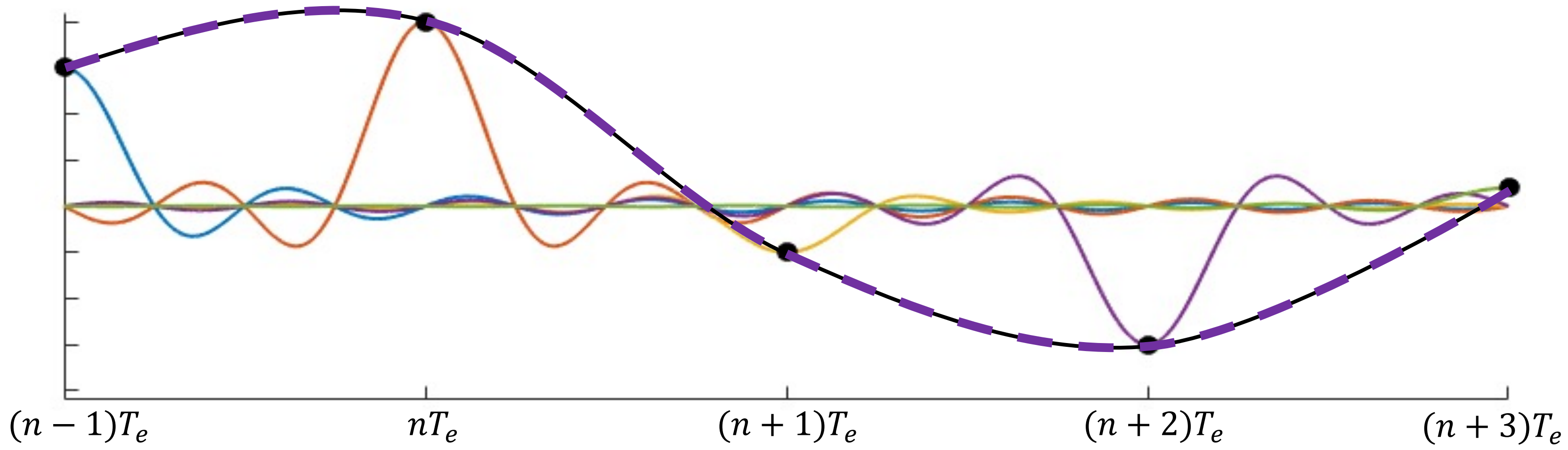
- Solution1 : Filtre « cardinal » : ***domaine temporel***

$$y(t) = x_e(t) * \mathrm{sinc}(\pi F_e t) = \cdots = \sum_{n \in Z} x_e(nT_e)\, \mathrm{sinc}(\pi F_e (t - nT_e))$$

2.3 Restitution / Reconstruction

- Solution1 : Filtre « cardinal » : ***domaine temporel***

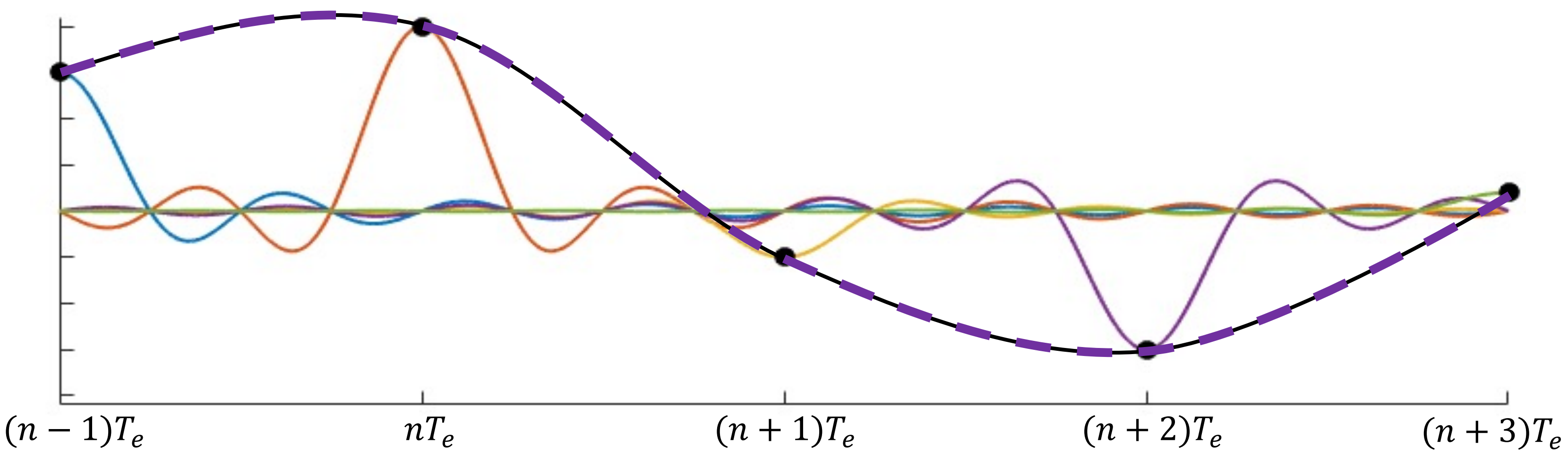
$$\boldsymbol{y(t) = x_e(t) * \mathrm{sinc}(\pi F_e t) = \cdots = \sum_{n\in Z} x_e(nT_e)\,\mathrm{sinc}(\pi F_e(t - nT_e))}$$



2.3 Restitution / Reconstruction

➢ Solution1 : Filtre « cardinal » : ***domaine temporel***

$$y(t) = x_e(t) * \mathrm{sinc}(\pi F_e t) = \cdots = \sum_{n\in Z} x_e(nT_e)\,\mathrm{sinc}(\pi F_e(t - nT_e))$$



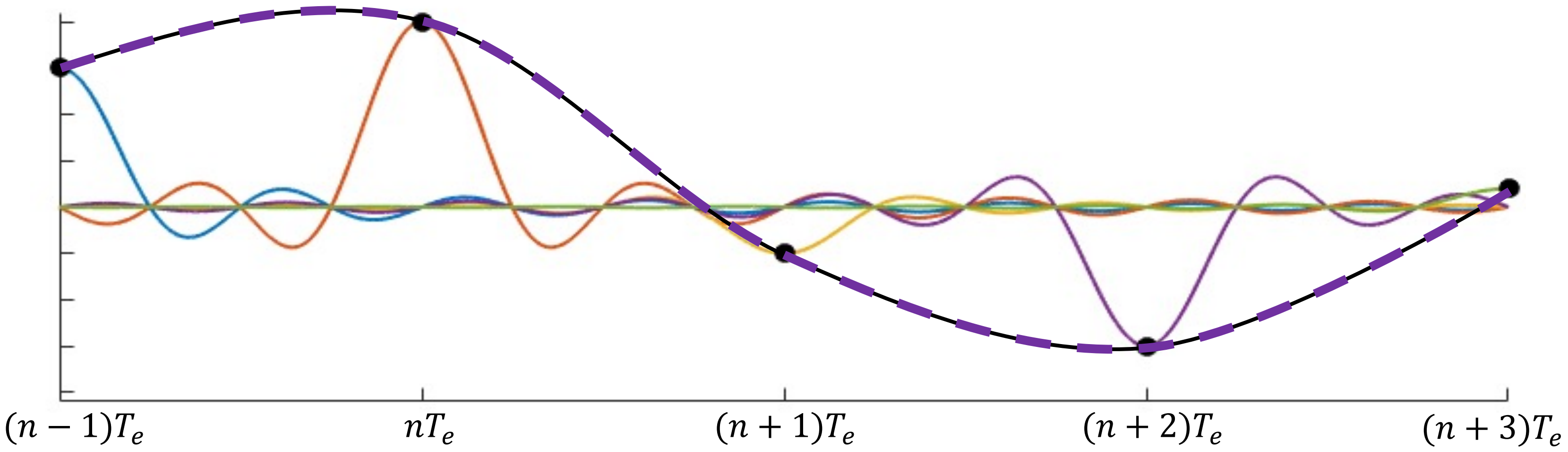
Problème :

$$y(t) = x_e(t) * \mathrm{sinc}(\pi F_e t) = \int_R \mathrm{sinc}(\pi F_e u)\, x_e(t-u)\,du$$

2.3 Restitution / Reconstruction

➢ <u>Solution1</u> : Filtre « cardinal » : ***domaine temporel***

$$\boldsymbol{y(t) = x_e(t) * \mathrm{sinc}(\pi F_e t) = \cdots = \sum_{n\in Z} x_e(nT_e)\,\mathrm{sinc}(\pi F_e(t - nT_e))}$$



Problème :

$$\boldsymbol{y(t) = x_e(t) * \mathrm{sinc}(\pi F_e t) = \int_R \mathrm{sinc}(\pi F_e u)\, x_e(t-u)\,du}$$

et … sinc non causal…

$\Rightarrow$ **latence!**

$\Rightarrow$ réalisable **<u>en temps différé mais pas en temps réel</u>**

2.3 Restitution / Reconstruction

- Solution2 : Interpolation

$$\hat{y}(t) \quad = \sum_{n\in Z} x_e(nT_e)\ h_r(t - nT_e) = \cdots = h_r * x_e\ (t)$$

2.3 Restitution / Reconstruction

- <u>Solution2</u> : Interpolation d'ordre 0 / bloqueur d'ordre 0

$$\hat{y}(t) = \sum_{n\in Z} x_e(nT_e)\ h_r(t-nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t-nT_e) + x_e((n+1)T_e)\ h_r(t-(n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k\in N} \frac{x^{(k)}(nT_e)}{k!}(t-nT_e)^k$

- **Ordre 0** : $x(t) =$

2.3 Restitution / Reconstruction

- Solution2 : Interpolation d'ordre 0 / bloqueur d'ordre 0

$$\hat{y}(t) = \sum_{n \in Z} x_e(nT_e)\ h_r(t - nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t - nT_e) + x_e((n+1)T_e)\ h_r(t - (n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k \in N} \frac{x^{(k)}(nT_e)}{k!}(t - nT_e)^k$

- **Ordre 0 :** $x(t) \approx x_e(nT_e)$ sur $[nT_e, (n+1)T_e], \forall n \in \mathbb{Z}$

Question : Que vaut h_r, notée h_0 pour avoir $\hat{y}(t) = x_e(nT_e)$ sur $[nT_e, (n+1)T_e]$?

2.3 Restitution / Reconstruction

- Solution2 : Interpolation d'ordre 0 / bloqueur d'ordre 0

$$\hat{y}(t) = \sum_{n\in Z} x_e(nT_e)\ h_r(t-nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t-nT_e) + x_e((n+1)T_e)\ h_r(t-(n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k\in N} \frac{x^{(k)}(nT_e)}{k!}(t-nT_e)^k$

- **Ordre 0 :** $x(t) \approx x_e(nT_e)$ sur $[nT_e, (n+1)T_e], \forall n \in \mathbb{Z}$

Question : Que vaut h_r, notée h_0 pour avoir $\hat{y}(t) = x_e(nT_e)$ sur $[nT_e, (n+1)T_e]$?
Réponse :

2.3 Restitution / Reconstruction

- <u>Solution2</u> : Interpolation d'ordre 0 / bloqueur d'ordre 0

$$\hat{y}(t) = \sum_{n\in Z} x_e(nT_e)\ h_r(t-nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t-nT_e) + x_e((n+1)T_e)\ h_r(t-(n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k\in N} \frac{x^{(k)}(nT_e)}{k!}(t-nT_e)^k$

- **<u>Ordre 0</u>** : $x(t) \approx x_e(nT_e)$ sur $[nT_e, (n+1)T_e], \forall n \in \mathbb{Z}$

Question : Que vaut h_r, notée h_0 pour avoir $\hat{y}(t) = x_e(nT_e)$ sur $[nT_e, (n+1)T_e]$?
Réponse : $h_0(t)\begin{cases} = 1 \text{ sur } [0, T_e] = \cdots \\ = 0 \text{ ailleurs} \end{cases}$

2.3 Restitution / Reconstruction

- Solution2 : Interpolation d'ordre 0 / bloqueur d'ordre 0

$$\hat{y}(t) = x_e(t) * T_e\, \Pi_{T_e}(t - \frac{T_e}{2})$$

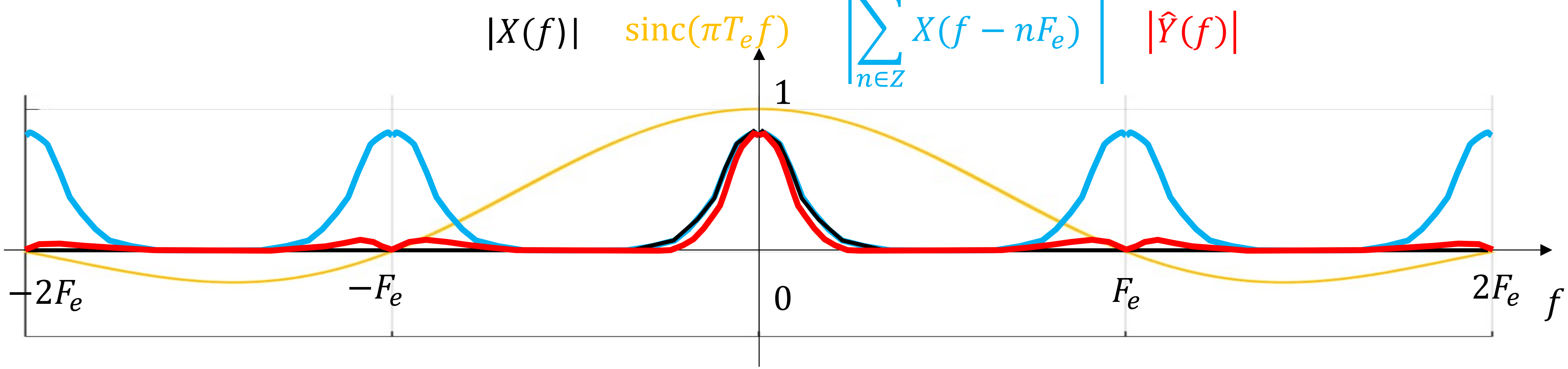
- ***Distorsion : domaine fréquentiel :***

2.3 Restitution / Reconstruction

➢ Solution2 : Interpolation d'ordre 0 / bloqueur d'ordre 0

$$\hat{y}(t) \quad = x_e(t) * T_e\, \Pi_{T_e}(t - \frac{T_e}{2})$$

- ***Distorsion : domaine fréquentiel :*** $\hat{Y}(f) = \mathrm{sinc}(\pi T_e f) \sum_{n \in Z} X(f - nF_e)\, e^{-j\pi f T_e}$



⇒Distorsion

2.3 Restitution / Reconstruction

➢ Solution2 : Interpolation d'ordre 1 / bloqueur d'ordre 1

$$\hat{y}(t) = \sum_{n\in Z} x_e(nT_e)\ h_r(t-nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t-nT_e) + x_e((n+1)T_e)\ h_r(t-(n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k\in N} \frac{x^{(k)}(nT_e)}{k!}(t-nT_e)^k$

➢ **Ordre 1 :** $x(t) =$

2.3 Restitution / Reconstruction

- Solution2 : Interpolation d'ordre 1 / bloqueur d'ordre 1

$$\hat{y}(t) = \sum_{n\in Z} x_e(nT_e)\ h_r(t-nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t-nT_e) + x_e((n+1)T_e)\ h_r(t-(n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k\in N} \frac{x^{(k)}(nT_e)}{k!}(t-nT_e)^k$

- **Ordre 1 :** $x(t) \approx x_e(nT_e) + \frac{x_e((n+1)T_e) - x_e(nT_e)}{T_e}(t-nT_e)$ sur $[nT_e, (n+1)T_e], \forall n \in \mathbb{Z}$

Question : Que vaut h_r, notée h_1 pour avoir $\hat{y}(t) = x_e(nT_e) + \frac{x_e((n+1)T_e) - x_e(nT_e)}{T_e}(t-nT_e)$ sur $[nT_e, (n+1)T_e]$?

2.3 Restitution / Reconstruction

- Solution2 : Interpolation d'ordre 1 / bloqueur d'ordre 1

$$\hat{y}(t) = \sum_{n \in Z} x_e(nT_e)\ h_r(t - nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t - nT_e) + x_e((n+1)T_e)\ h_r(t - (n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k \in N} \frac{x^{(k)}(nT_e)}{k!}(t - nT_e)^k$

- **Ordre 1 :** $x(t) \approx x_e(nT_e) + \frac{x_e((n+1)T_e) - x_e(nT_e)}{T_e}(t - nT_e)$ sur $[nT_e, (n+1)T_e], \forall n \in \mathbb{Z}$

Question : Que vaut h_r, notée h_1 pour avoir $\hat{y}(t) = x_e(nT_e) + \frac{x_e((n+1)T_e) - x_e(nT_e)}{T_e}(t - nT_e)$ sur $[nT_e, (n+1)T_e]$?

Réponse :

2.3 Restitution / Reconstruction

➢ <u>Solution2</u> : Interpolation d'ordre 1 / bloqueur d'ordre 1

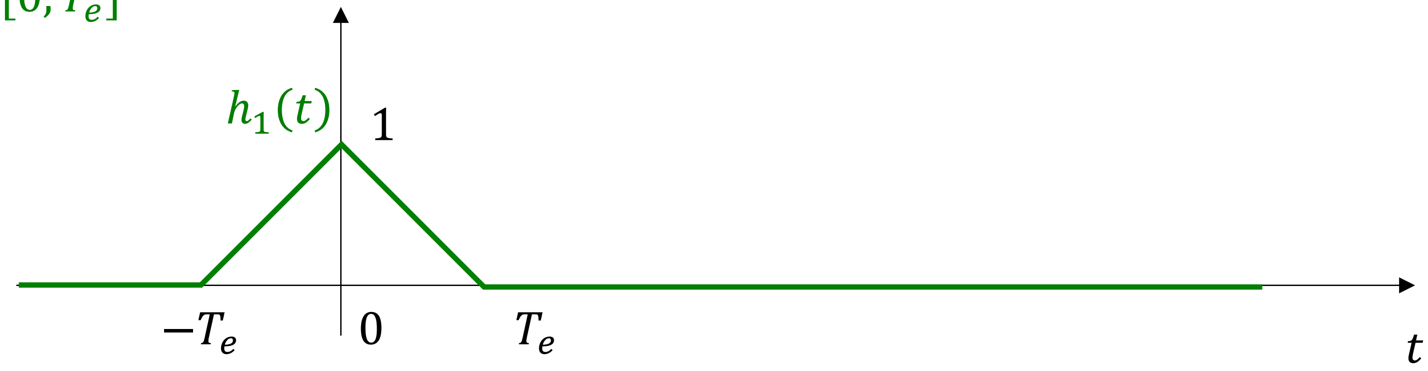
$$\hat{y}(t) = \sum_{n \in Z} x_e(nT_e)\ h_r(t - nT_e) = \cdots = h_r * x_e\ (t)$$
$$= \ldots + x_e(nT_e)\ h_r(t - nT_e) + x_e((n+1)T_e)\ h_r(t - (n+1)T_e) + \ \ldots$$

Or, pour $t \in [nT_e, (n+1)T_e]$: $x(t) = \sum_{k \in N} \frac{x^{(k)}(nT_e)}{k!}(t - nT_e)^k$

➢ **Ordre 1 :** $x(t) \approx x_e(nT_e) + \frac{x_e((n+1)T_e) - x_e(nT_e)}{T_e}(t - nT_e)$ sur $[nT_e, (n+1)T_e], \forall n \in \mathbb{Z}$

Question : Que vaut h_r, notée h_1 pour avoir $\hat{y}(t) = x_e(nT_e) + \frac{x_e((n+1)T_e) - x_e(nT_e)}{T_e}(t - nT_e)$ sur $[nT_e, (n+1)T_e]$?

Réponse : $h_1(t) \begin{cases} = \frac{t}{T_e} + 1 \text{ sur } [-T_e, 0] \\ = -\frac{t}{T_e} + 1 \text{ sur } [0, T_e] \\ = 0 \text{ ailleurs} \end{cases}$ $= \cdots$



2.3 Restitution / Reconstruction

➢ <u>Solution2</u> : Interpolation d'ordre 1 / bloqueur d'ordre 1

$$\hat{y}(t) \qquad = x_e(t) * \Lambda_{T_e}(t)$$

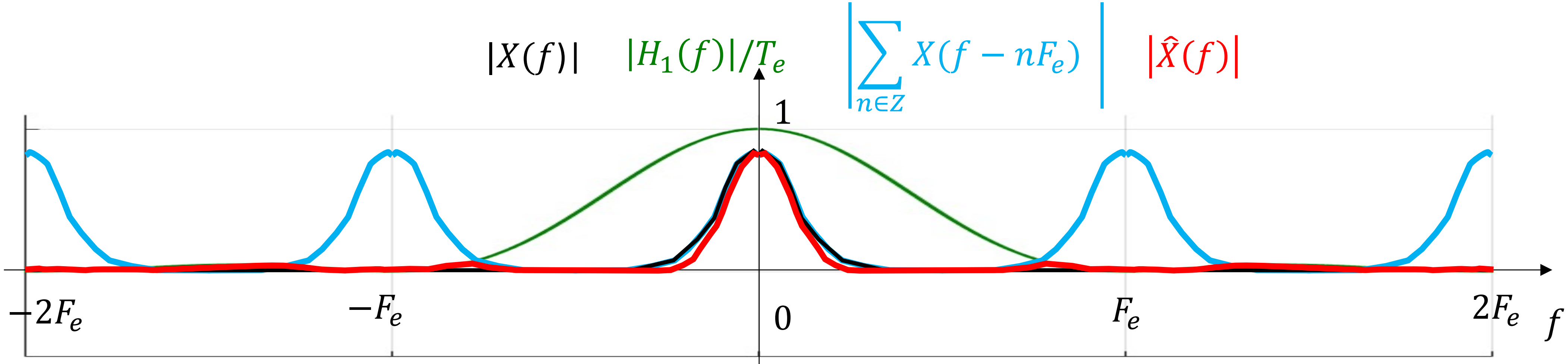
- ***Distorsion : domaine fréquentiel :***

2.3 Restitution / Reconstruction

➢ Solution2 : Interpolation d'ordre 1 / bloqueur d'ordre 1

$$\hat{y}(t) \quad = x_e(t) * \Lambda_{T_e}(t)$$

- ***Distorsion : domaine fréquentiel :*** $\hat{Y}(f) = \mathrm{sinc}^2(\pi T_e f)\sum_{n\in Z} X(f - nF_e)$



⇒ Distorsion plus importante entre –Fe/2 et Fe/2
Lobes secondaires plus faibles ⇒ distorsion résiduelle plus faible en dehors de [–Fe/2, Fe/2]

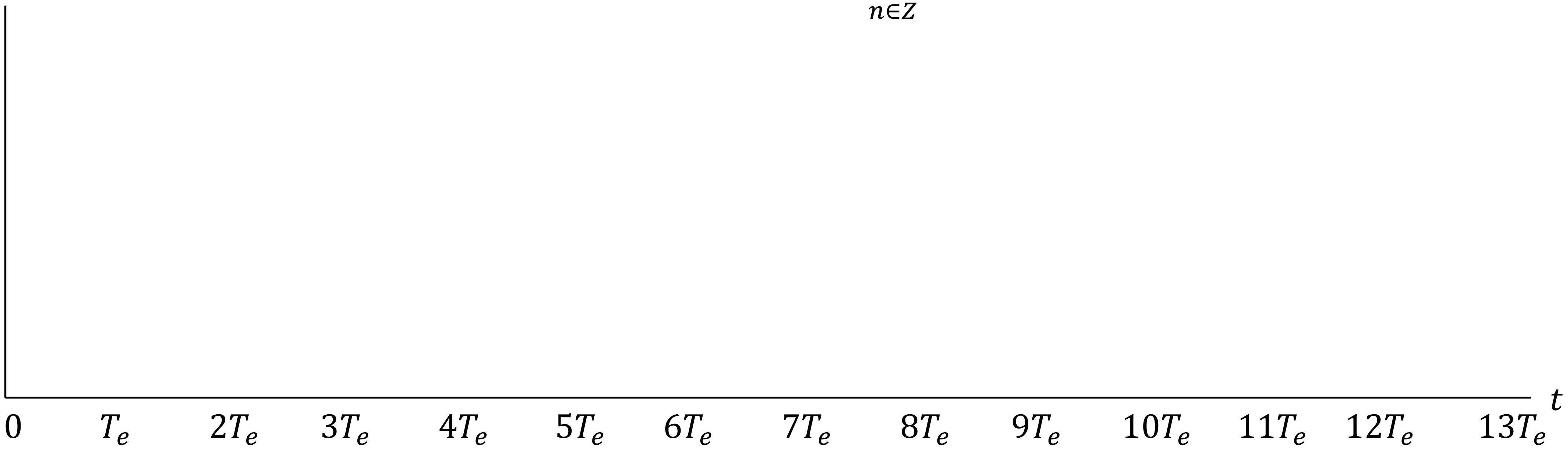
2.3 Restitution / Reconstruction

- Solution2 : Interpolation
  - « bloqueur d'ordre 0 »:

$$\hat{X}(f) = H_0(f)X_e(f) = \text{sinc}\,(\pi T_e f)e^{-j\pi f T_e} \times \sum_{n\in Z} X(f - nF_e)$$ ⇒ **latence**

  - « bloqueur d'ordre 1 »:

$$\hat{X}(f) = H_0(f)X_e(f) = \text{sinc}^2\,(\pi T_e f) \times \sum_{n\in Z} X(f - nF_e)$$



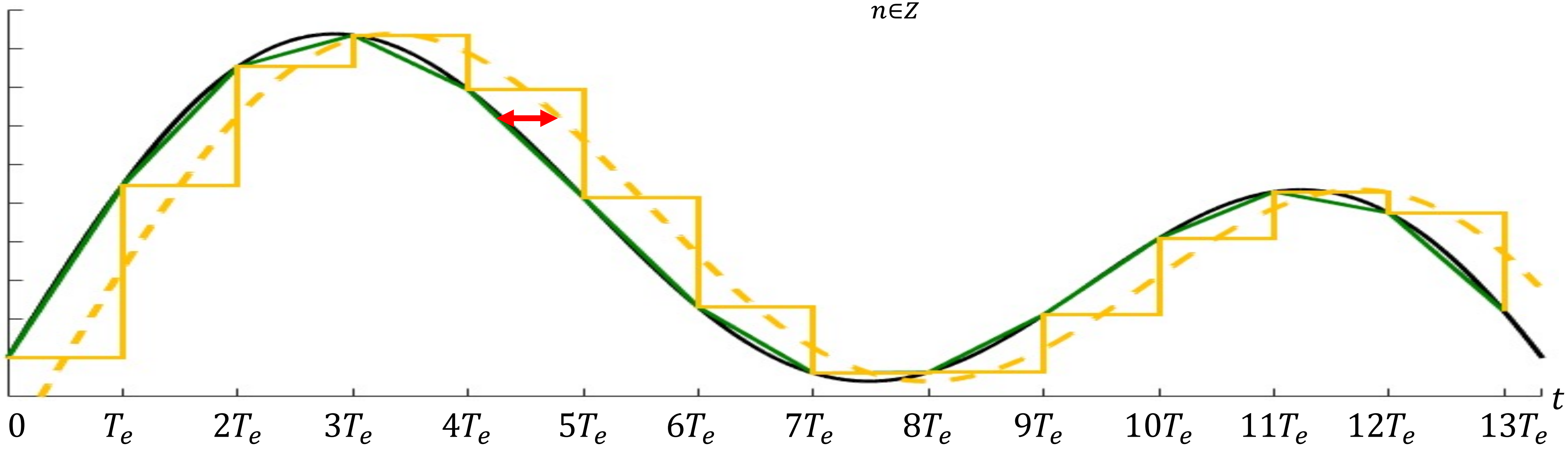
2.3 Restitution / Reconstruction

- Solution2 : Interpolation
  - « bloqueur d'ordre 0 »:

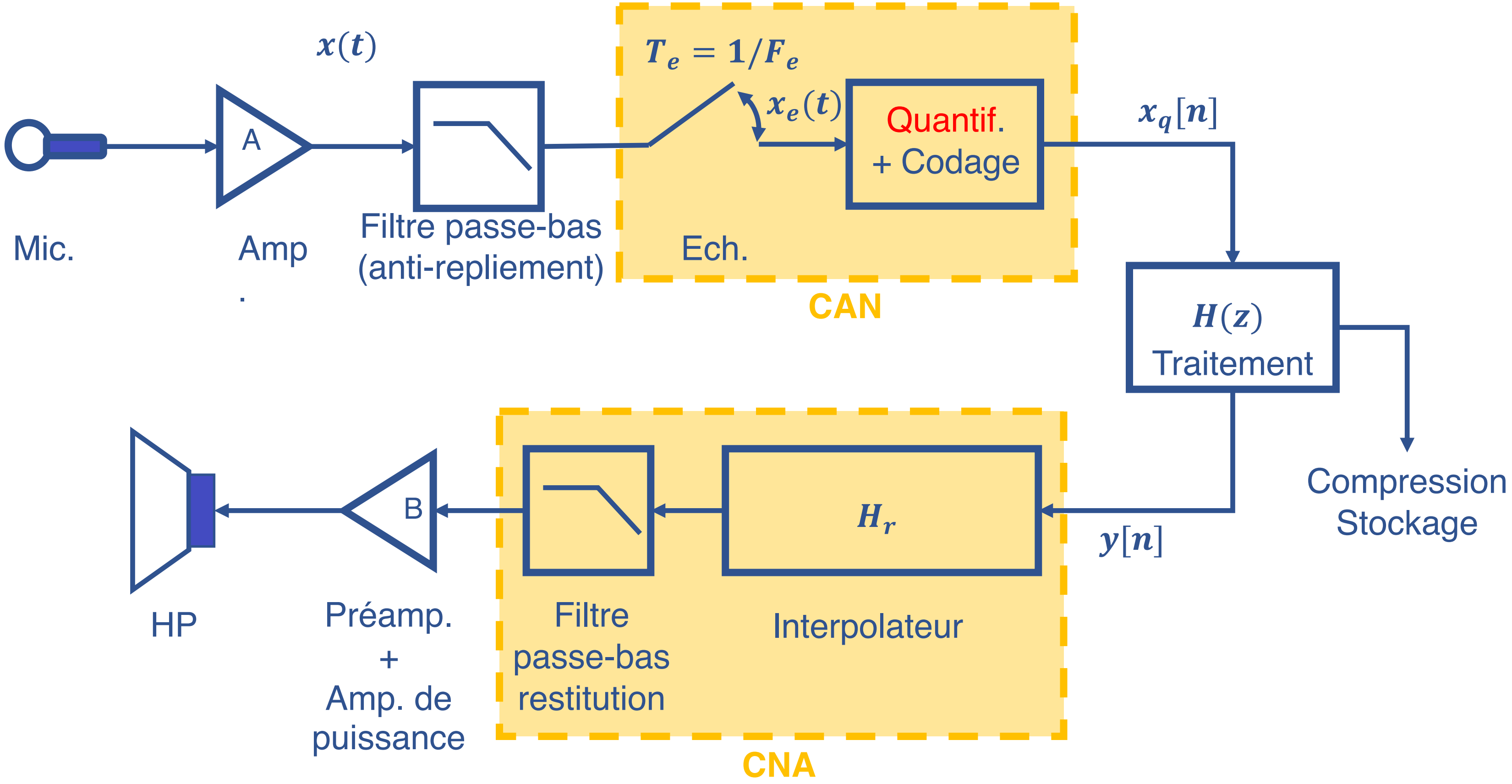
$$\hat{X}(f) = H_0(f)X_e(f) = \text{sinc}\,(\pi T_e f)e^{-j\pi f T_e} \times \sum_{n\in Z} X(f - nF_e) \Rightarrow \textbf{latence}$$

  - « bloqueur d'ordre 1 »:

$$\hat{X}(f) = H_0(f)X_e(f) = \text{sinc}^2\,(\pi T_e f) \times \sum_{n\in Z} X(f - nF_e)$$



2.4 Quantification



Chaine d'acquisition et de traitement d'un signal sonore

2.4 Quantification

- Définition: La quantification est un **arrondi** à une valeur autorisée. Elle entraîne une **perte systématique d'information**.

- Notations: - ensemble *fini* des valeurs autorisées : $\boldsymbol{\Omega}$,
  - signal quantifié : $\boldsymbol{x_q} = \left(\boldsymbol{x_q[n]}\right)_{\boldsymbol{n}}$ tel que $\boldsymbol{x_q[n]} \in \Omega, \forall n$

- Exemples:- N bits de quantification $\Rightarrow \mathrm{card}\,\boldsymbol{\Omega} =$

  - Pour $N = 3$ bits : $\boldsymbol{\Omega} =$

2.4 Quantification

- Loi de quantification : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

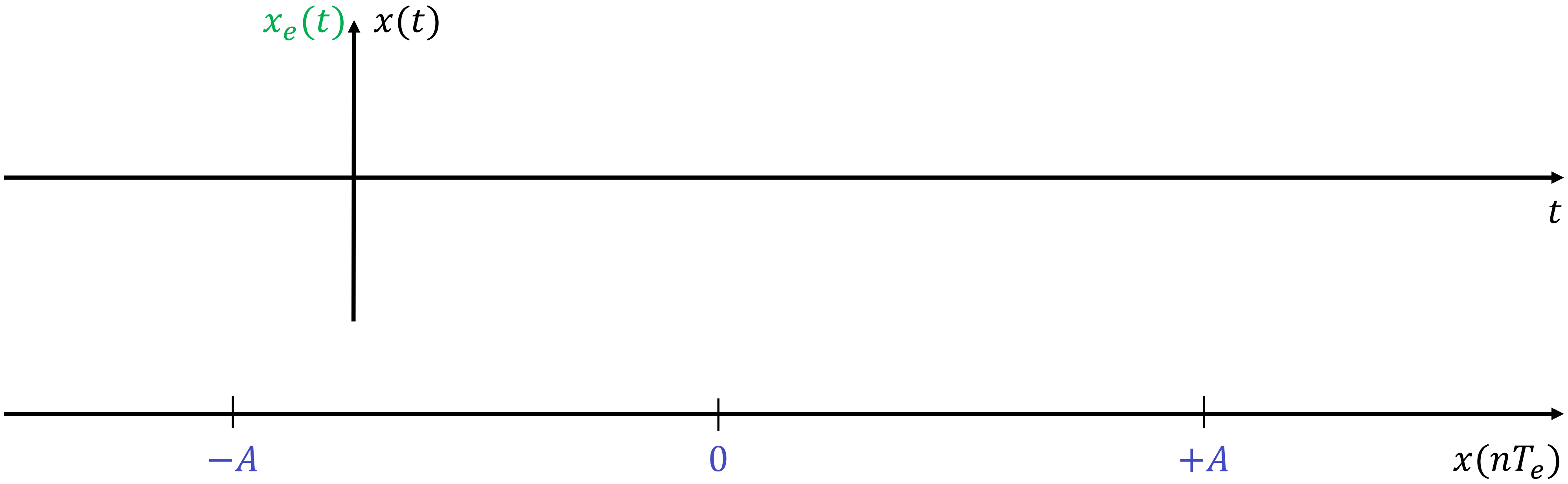
Elle repose sur 3 choix :

2.4 Quantification

- <u>Loi de quantification</u> : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

Elle repose sur 3 choix :

- la valeur de **<u>pleine échelle A</u>** : valeur max de $|x(nT_e)|$ pouvant être quantifiée



SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

MASTER Spi

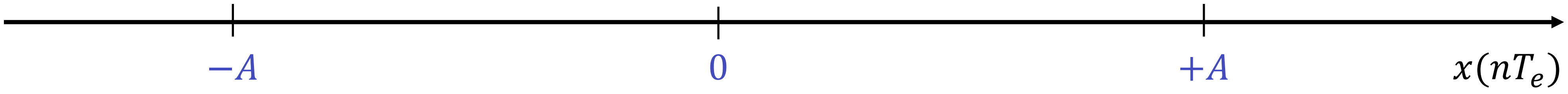
2.4 Quantification

- <u>Loi de quantification</u> : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

Elle repose sur 3 choix :
- la valeur de **<u>pleine échelle A</u>**
- la répartition des 2^N valeurs $x_q[n]$ de $\boldsymbol{\Omega}$

Loi de quantification **<u>non-uniforme</u>**

Loi de quantification **<u>uniforme</u>**



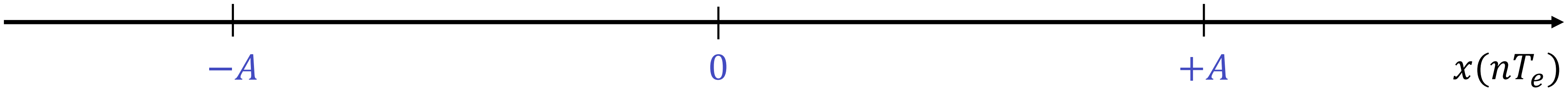
2.4 Quantification

➢ <u>Loi de quantification</u> : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

Elle repose sur 3 choix :

- la valeur de **<u>pleine échelle A</u>**
- la répartition des 2^N valeurs $x_q[n]$ de $\boldsymbol{\Omega}$
- la méthode d'arrondi

Exemple :



2.4 Quantification

- Loi de quantification : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

- Loi de quantification uniforme :
  - On appelle quantum la distance entre 2 valeurs consécutives de $\boldsymbol{\Omega}$.

  $q = 2A/2^N$, où $A =$ valeur de pleine échelle et $N =$ nombre de bits.

2.4 Quantification

- Loi de quantification : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

- Loi de quantification uniforme :
  - On appelle quantum la distance entre 2 valeurs consécutives de $\boldsymbol{\Omega}$.
  $q = 2A/2^N$, où $A =$ valeur de pleine échelle et $N =$ nombre de bits.

  - Exemples : $\forall k \in \mathbb{Z}$
    - a- $kq \leq x(nT_e) < (k+1)q \Rightarrow x_q[n] = kq =$ arrondi par défaut

2.4 Quantification

- Loi de quantification : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

- Loi de quantification uniforme :
  - On appelle quantum la distance entre 2 valeurs consécutives de $\boldsymbol{\Omega}$.
  $q = 2A/2^N$, où $A =$ valeur de pleine échelle et $N =$ nombre de bits.

  - Exemples : $\forall k \in \mathbb{Z}$
    - a- $kq \leq x(nT_e) < (k+1)q \Rightarrow x_q[n] = kq =$ arrondi par défaut
    - b- $kq < x(nT_e) \leq (k+1)q \Rightarrow x_q[n] = (k+1)q =$ arrondi par excès

2.4 Quantification

- <u>Loi de quantification</u> : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

- <u>Loi de quantification uniforme</u> :
  - On appelle quantum la distance entre 2 valeurs consécutives de $\boldsymbol{\Omega}$.

  $q = 2A/2^N$, où $A =$ valeur de pleine échelle et $N =$ nombre de bits.

  - Exemples : $\forall k \in \mathbb{Z}$
    - a- $kq \leq x(nT_e) < (k+1)q \Rightarrow x_q[n] = kq =$ arrondi par défaut
    - b- $kq < x(nT_e) \leq (k+1)q \Rightarrow x_q[n] = (k+1)q =$ arrondi par excès
    - c- $\left(\boldsymbol{k} - \frac{1}{2}\right)\boldsymbol{q} \substack{< \\ \leq} \boldsymbol{x}(nT_e) \substack{\leq \\ <} \left(\boldsymbol{k} + \frac{1}{2}\right)\boldsymbol{q} \Rightarrow x_q[n] = kq =$ arrondi au plus proche

2.4 Quantification

- Loi de quantification : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

- Loi de quantification uniforme :
  - On appelle quantum la distance entre 2 valeurs consécutives de $\boldsymbol{\Omega}$.
  $q = 2A/2^N$, où $A =$ valeur de pleine échelle et $N =$ nombre de bits.

  - Exemples : $\forall k \in \mathbb{Z}$
    - a- $kq \leq x(nT_e) < (k+1)q \Rightarrow x_q[n] = kq =$ arrondi par défaut
    - b- $kq < x(nT_e) \leq (k+1)q \Rightarrow x_q[n] = (k+1)q =$ arrondi par excès
    - c- $\left(\boldsymbol{k} - \frac{1}{2}\right)\boldsymbol{q} \genfrac{}{}{0pt}{}{<}{\leq} \boldsymbol{x}(nT_e) \genfrac{}{}{0pt}{}{\leq}{<} \left(\boldsymbol{k} + \frac{1}{2}\right)\boldsymbol{q} \Rightarrow x_q[n] = kq =$ arrondi au plus proche

- Bruit de quantification : $\boldsymbol{\varepsilon_q = x_q - x}$

2.4 Quantification

- Loi de quantification : loi qui détermine la valeur quantifiée $x_q[n]$ pour une valeur $x(nT_e)$ du signal échantillonné temps continu.

- Loi de quantification uniforme :
  - On appelle quantum la distance entre 2 valeurs consécutives de $\boldsymbol{\Omega}$.
  $q = 2A/2^N$, où $A =$ valeur de pleine échelle et $N =$ nombre de bits.

  - Exemples : $\forall k \in \mathbb{Z}$
    - a- $kq \leq x(nT_e) < (k+1)q \Rightarrow x_q[n] = kq =$ arrondi par défaut $\qquad \boldsymbol{E(\varepsilon_q) > 0}$
    - b- $kq < x(nT_e) \leq (k+1)q \Rightarrow x_q[n] = (k+1)q =$ arrondi par excès $\qquad \boldsymbol{E(\varepsilon_q) < 0}$
    - c- $\left(\boldsymbol{k} - \frac{1}{2}\right)\boldsymbol{q} \stackrel{<}{\leq} \boldsymbol{x}(nT_e) \stackrel{\leq}{<} \left(\boldsymbol{k} + \frac{1}{2}\right)\boldsymbol{q} \Rightarrow x_q[n] = kq =$ arrondi au plus proche $\qquad \boxed{\boldsymbol{E(\varepsilon_q) = 0}}$

- Bruit de quantification : $\boldsymbol{\varepsilon_q = x_q - x}$

2.4 Quantification

- Rapport signal sur bruit de quantification :
  Signal $x[n]$, bruit de quantification $\boldsymbol{\varepsilon_q[n] = x_q[n] - x[n]}$

$RSB_q = \frac{\sigma_x^2}{\sigma_{\boldsymbol{\varepsilon_q}}^2}$ avec : σ_x^2 = variance de x,

$\sigma_{\boldsymbol{\varepsilon_q}}^2$ = variance du bruit

2.4 Quantification

- <u>Rapport signal sur bruit de quantification :</u>

Hyp.: bruit $\boldsymbol{\varepsilon_q}$ = processus aléatoire:
- centré en 0: $E(\boldsymbol{\varepsilon_q}) = \mathbf{0}$
- de loi de probabilité $p_{\varepsilon_q}(\boldsymbol{\varepsilon_q})$ uniforme entre $-\frac{q}{2}$ et $\frac{q}{2}$

$$\Rightarrow p_{\varepsilon_q}(\boldsymbol{\varepsilon_q}) = \Pi_q(\boldsymbol{\varepsilon_q}) = \begin{cases} 1/q \text{ sur } [-\frac{q}{2}, \frac{q}{2}] \\ 0 \text{ ailleurs} \end{cases}$$

2.4 Quantification

- Rapport signal sur bruit de quantification : $RSB_q = \frac{\sigma_x^2}{\sigma_{\varepsilon_q}^2} = 6N - 10\log\frac{F^2}{3}$

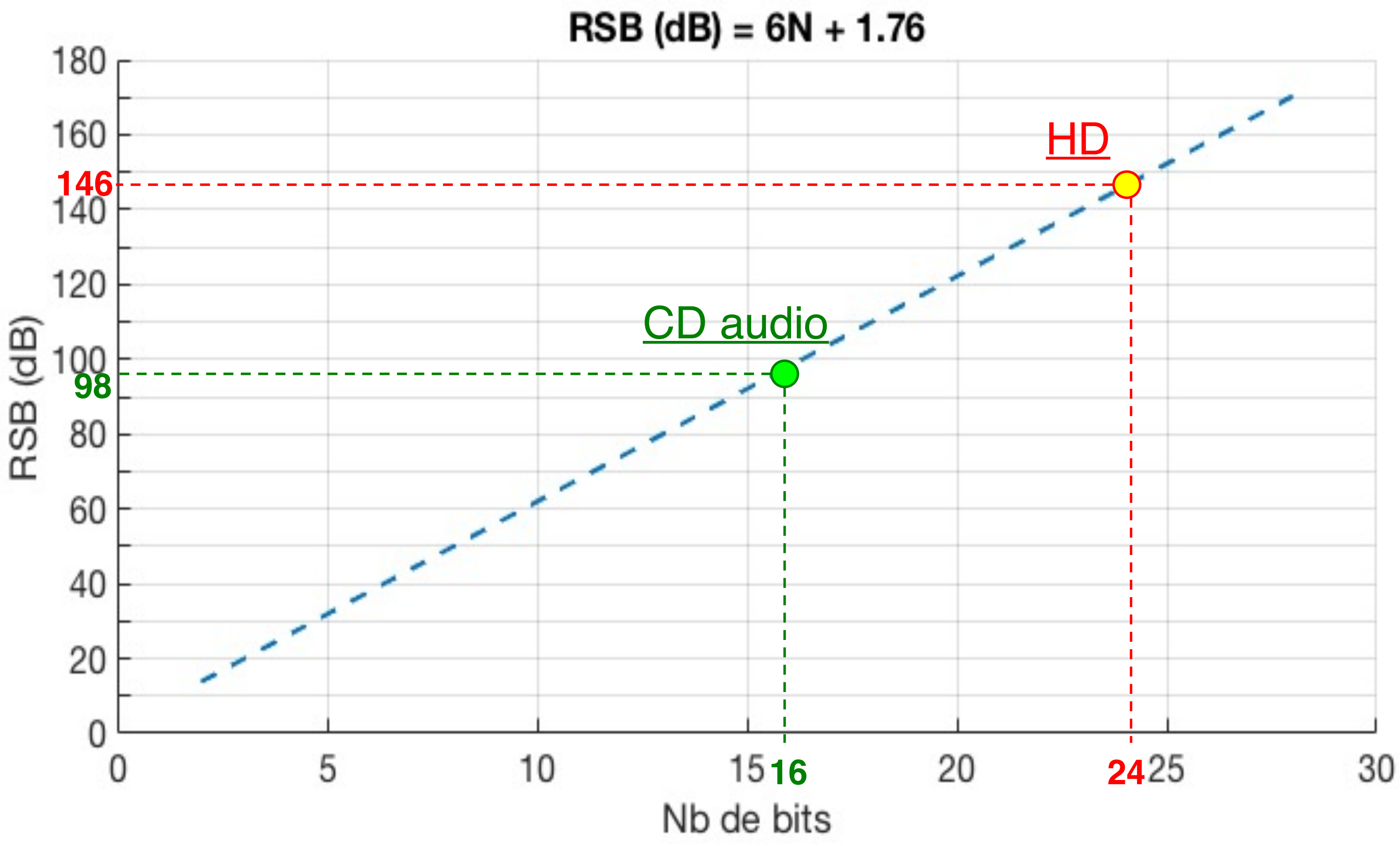
où : N = nombre de bits et $F = \frac{A}{\sigma_x}$ est appelé le facteur de charge

2.4 Quantification

- Rapport signal sur bruit de quantification : $RSB_q = \frac{\sigma_x^2}{\sigma_{\varepsilon q}^2} = 6N - 10\log\frac{F^2}{3}$
- Exemple1 : format CD audio : $N = 16$ bits, et format HD audio : $N = 24$ bits

2.4 Quantification

- Rapport signal sur bruit de quantification : $RSB_q = \frac{\sigma_x^2}{\sigma_{\varepsilon_q}^2} = 6N - 10\log\frac{F^2}{3}$
- Exemple1 : format CD audio : $N = 16$ bits, et format HD audio : $N = 24$ bits



2.4 Quantification

- Rapport signal sur bruit de quantification : $RSB_q = \frac{\sigma_x^2}{\sigma_{\varepsilon q}^2} = 6N - 10\log\frac{F^2}{3}$
- Exemple2 : tracer $RSB_{CD\ \mathrm{audio}}$ en fonction du facteur de charge, pour σ_x^2 fixé à 1.

2.4 Quantification

- Rapport signal sur bruit de quantification : $RSB_q = \frac{\sigma_x^2}{\sigma_{\varepsilon_q}^2} = 6N - 10\log\frac{F^2}{3}$
- Exemple2 : tracer $RSB_{CD\ \mathrm{audio}}$ en fonction du facteur de charge, pour σ_x^2 fixé à 1.

RSB = fonction linéaire décroissante de $\log F^2$
pour $\boldsymbol{\varepsilon_q}$ de loi uniforme
$(1/q$ sur $[-\frac{q}{2},\frac{q}{2}]$ et 0 ailleurs)