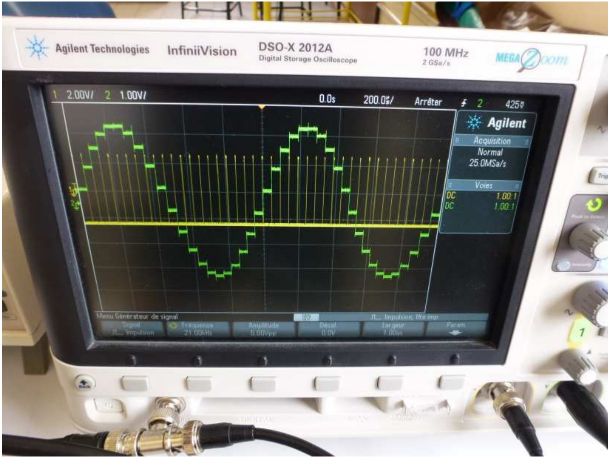
Échantillonnage

JR Seigne MP*, Clemenceau Nantes

September 11, 2023

JR Seigne
MP*,
Clemenceau
Nantes

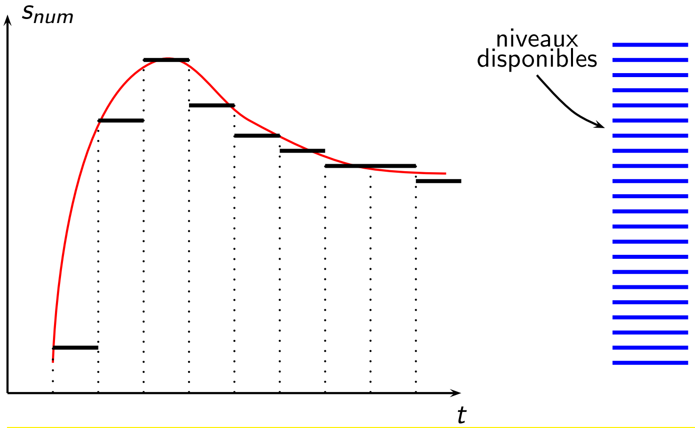


Informatique : mode binaire 0 ou 1.
Signal analogique : évolution continue au cours du temps.
Signal numérique : valeurs discrètes au cours du temps.
Ces valeurs discrètes sont codées par un ensemble de 0 et 1 dont la base de représentation est l'octet.

La Physique n'apprécie guère les discontinuités mais l'ordinateur ne peut pas traiter une fonction continue.
À quel rythme va-t-on répéter les évaluations d'un signal analogique ?

T_e : période d'échantillonnage
$F_e = 1/T_e$: fréquence d'échantillonnage

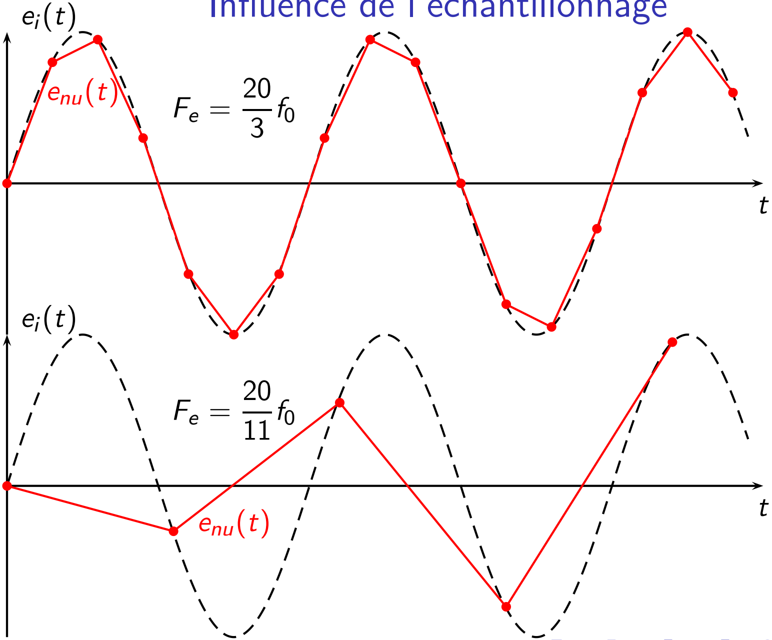
Fréquence F_e adaptée à la rapidité d'évolution d'un signal...
Avoir une idée des fréquences présentes dans le signal... analyse de Fourier



Travailler en 8 bits : $2^8 = 256$ niveaux

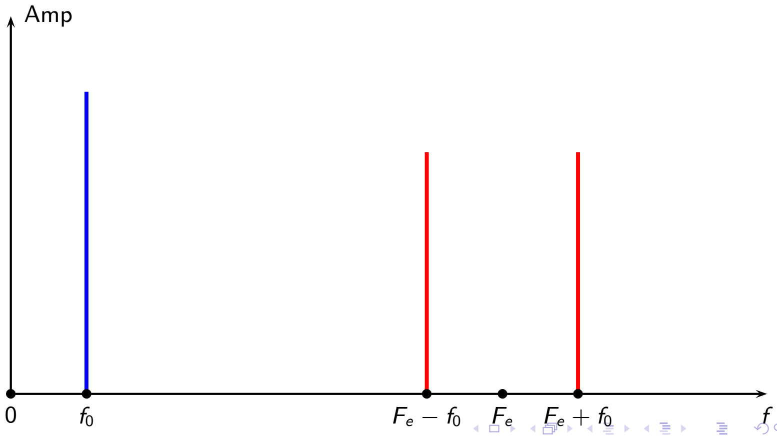
Travailler en 64 bits : $2^{64} = 2 \times 10^{19}$ niveaux

Influence de l'échantillonnage

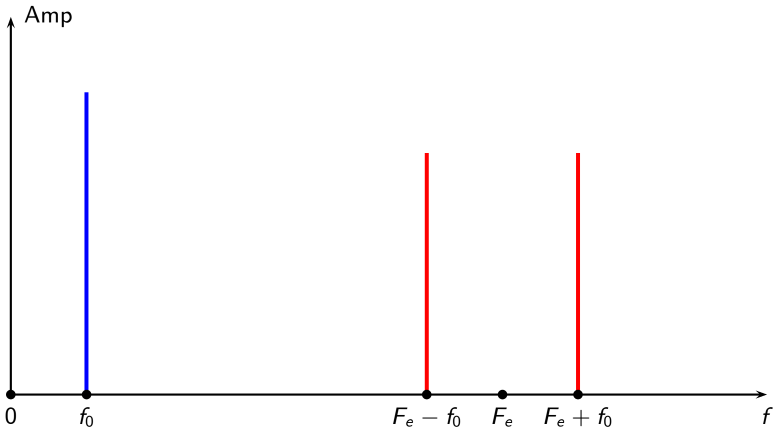


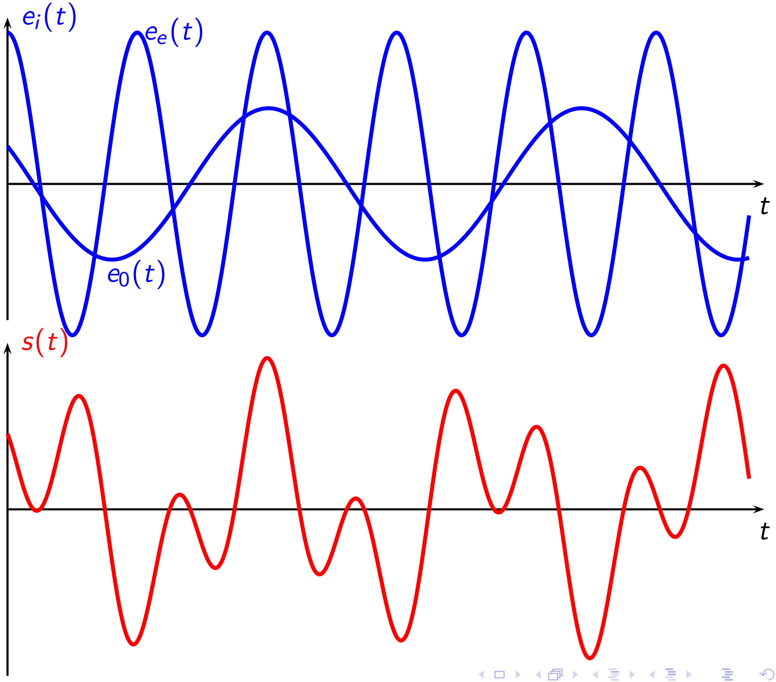
$$e_0(t) = E_1 \cos(2\pi f_0 t + \varphi_0) \quad e_e(t) = E_e \cos(2\pi F_e t)$$

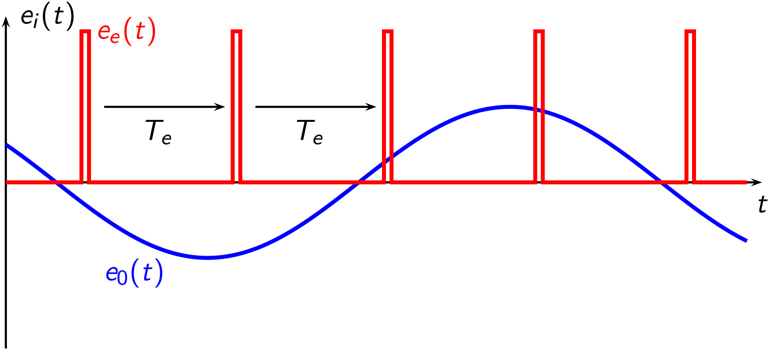
$$s(t) = \frac{e_1(t) \times e_e(t)}{V_0}$$

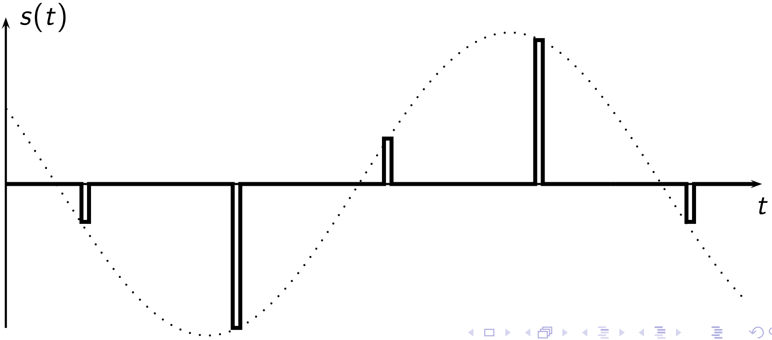


$$s(t) = \frac{E_1 E_2}{2V_0}\left[\cos\left(2\pi(F_e+f_0)t+\varphi_0\right)+\cos\left(2\pi(F_e-f_0)t-\varphi_0\right)\right]$$




$e_i(t)$
$e_e(t)$
$e_0(t)$
t
$s(t)$
t


$e_i(t)$
$e_e(t)$
T_e
T_e
t
$e_0(t)$


$s(t)$
t

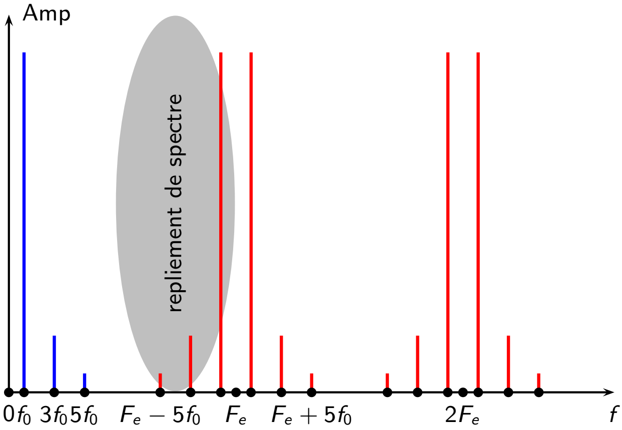
Fréquence du signal : f_0

Fréquences du peigne (signal d'échantillonnage) : nF_e

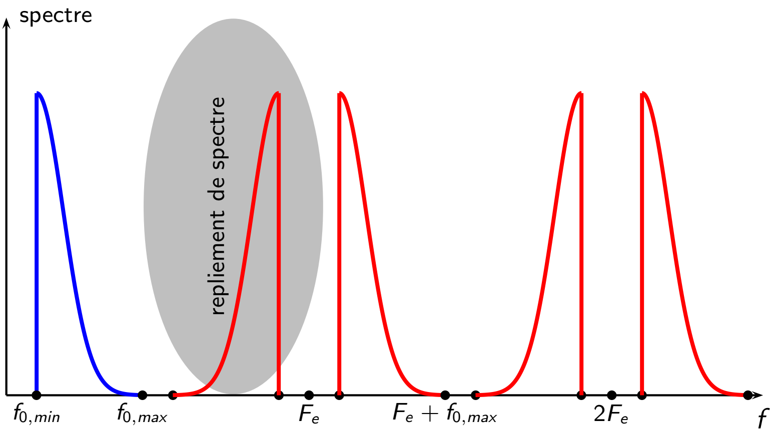
Fréquences du signal échantillonné

$$nF_e \pm f_0$$

Spectre d'un signal triangulaire échantillonné

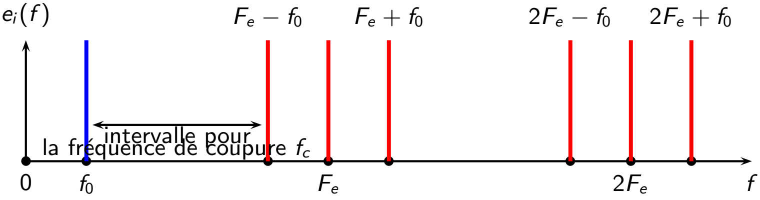


Spectre d'un signal échantillonné possédant au départ un spectre continu

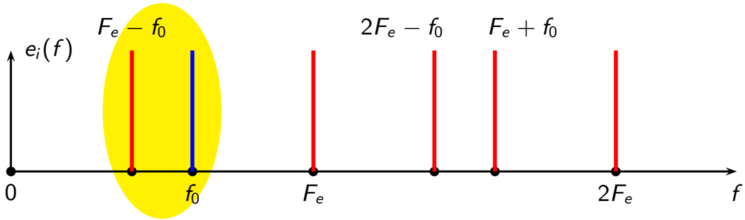


Signal bien échantillonné

Après une numérisation, on met en forme le signal pour assurer une bonne transmission du signal ou bien un traitement. Une fois que cela est terminé, il faut être capable de retrouver une image fidèle du signal. Il faut extraire par filtrage le signal de départ du signal réceptionné.



Signal mal échantillonné



Il faut éviter que $F_e - f_0$ soit inférieur à f_0 !

Critère de Shannon

$$F_e \geq 2f_0$$

Principe du CAN

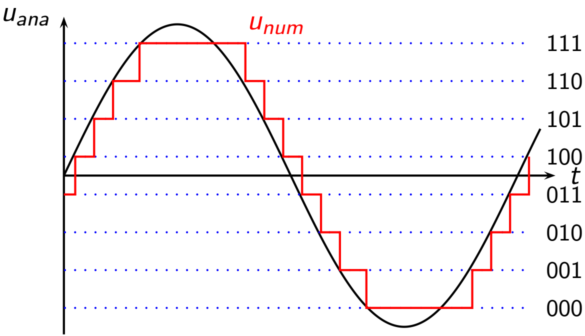
Objectif : transformer la valeur du signal échantillonné en une suite de 0 et de 1

- Choisir un nombre de bits n
- Créer une liste de 2^n valeurs accessibles dans un intervalle de tension Δu_{num} appelé *dynamique du convertisseur*
- Associer à chaque valeur une suite de 0 et 1
- Attribuer à chaque valeur échantillonnée une valeur dans celles accessibles
- Faire correspondre à chaque valeur échantillonnée au départ la suite de 0 et de 1 correspondante
- Stocker le résultat en mémoire

Le pas d'échantillonnage q est :

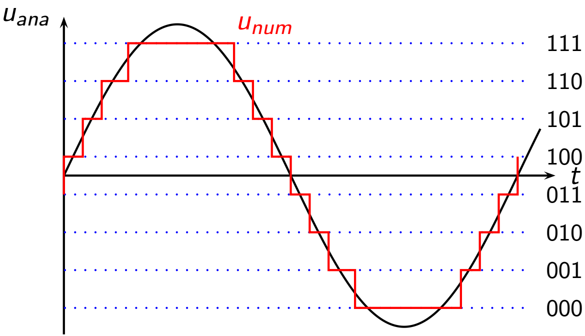
$$q = \frac{\Delta u_{num}}{2^n - 1} \simeq \frac{\Delta u_{num}}{2^n}$$

Erreurs de quantification



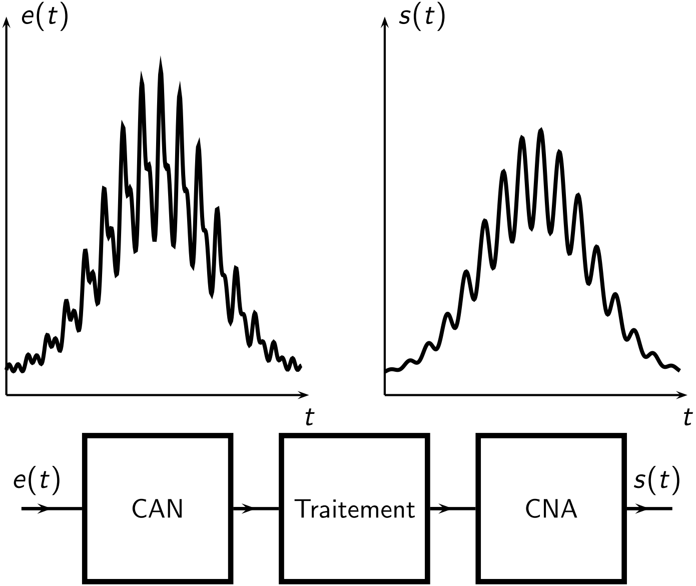
Bruit de quantification $b(t) = u_{num}(t) - u_{ana}(t)$ pour $2^3 = 8$ pas (3 bits) avec basculement en fin de pas de quantification

Erreurs de quantification



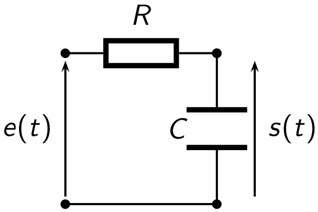
Bruit de quantification $b(t) = u_{num}(t) - u_{ana}(t)$ pour $2^3 = 8$ pas (3 bits) avec basculement au milieu du pas de quantification

Chaîne de traitement



Passe-bas 1er ordre

Filtre analogique du premier ordre :



$$\underline{H}(j\omega) = \frac{\underline{s}}{\underline{e}} = \frac{1}{1 + jRC\omega} = \frac{1}{1 + j\omega\tau}$$

En version temporelle :

$$\tau \frac{\mathrm{d}s}{\mathrm{d}t} + s = e(t)$$

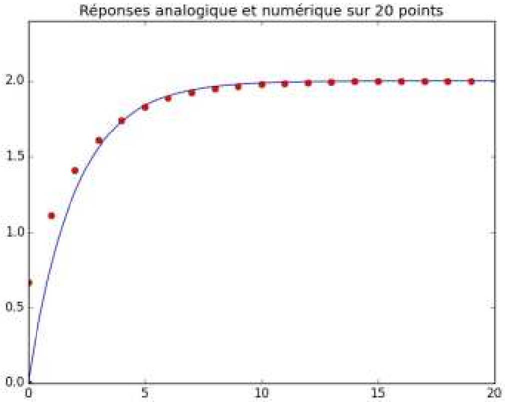
Filtre numérique 1er ordre

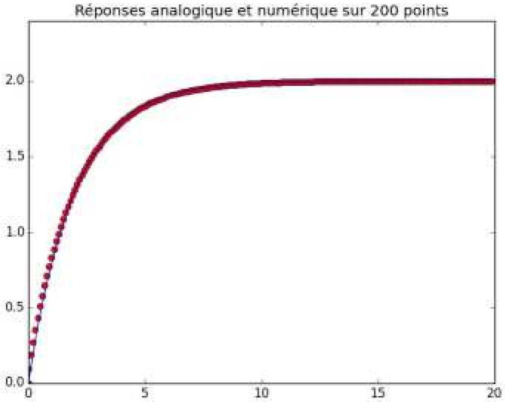
On passe au filtre numérique en assimilant la dérivée à un taux de variation sur la période d'échantillonnage T_e :

$$\tau \frac{\mathrm{d}s}{\mathrm{d}t} + s = e(t) \qquad \longrightarrow \qquad \tau \frac{s_n - s_{n-1}}{T_e} + s_n = e_n$$

$$s_n = \frac{\theta}{1+\theta}\, s_{n-1} + \frac{1}{1+\theta}\, e_n$$

L'état de la sortie au rang n dépend de l'état de l'entrée au rang n et de la sortie au rang $n-1$. Le poids de chaque contribution est fixé par $\theta = \tau / T_e$ qui compare le temps caractéristique du filtre à celui de la période d'échantillonnage.


Réponses analogique et numérique sur 20 points
2.0
1.5
1.0
0.5
0.0
0
5
10
15
20


Réponses analogique et numérique sur 200 points
0.0
0.5
1.0
1.5
2.0
0
5
10
15
20

Moyenne glissante

$$a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}\ a_{11}\ a_{12}\ a_{13}\ a_{14}\ a_{15}$$

a_3^{moy}

a_4^{moy}

calcul d'une moyenne sur une portion de la liste

affectation arbitraire du résultat à une position

décalage d'une place de la moyenne...

reconstitution d'une liste plus courte

$$?\quad ?\quad a_3^{moy}\, a_4^{moy}\, a_5^{moy}\, a_6^{moy}\, a_7^{moy}\, a_8^{moy}\, a_9^{moy}\, a_{10}^{moy}\, a_{11}^{moy}\, a_{12}^{moy}\, a_{13}^{moy}\quad ?\quad ?$$

on peut compléter la liste

$$a_1\quad a_2\ a_3^{moy}\, a_4^{moy}\, a_5^{moy}\, a_6^{moy}\, a_7^{moy}\, a_8^{moy}\, a_9^{moy}\, a_{10}^{moy}\, a_{11}^{moy}\, a_{12}^{moy}\, a_{13}^{moy}\ a_{14}\quad a_{15}$$

Processus

- Calculer la transformée de Fourier du signal
- Récupèrer le tableau associant à une fréquence son amplitude et sa phase
- Supprimer dans le tableau les fréquences non voulues, par exemple les hautes fréquences si on veut faire agir un filtre passe-bas
- Calculer la transformée de Fourier inverse du spectre tronqué

Le résultat de la Transformée de Fourier inverse donne le signal filtré.