Information, Calcul et Communication: Cours 8

Résumé de la dernière leçon:

- Représentation binaire des nombres entiers positifs

Plan de cette leçon:

- Représentation binaire des nombres entiers relatifs
- Représentation binaire des nombres réels
- Echantillonnage de signaux
- Bande passante, théorème d'échantillonnage

Représentation binaire des nombres entiers relatifs (+ et -)

Rappel : Avec 11 bits, on peut représenter les nombres entiers positifs de 0 à $2^{11}-1 = 2048-1 = 2047$:

$$N = \sum_{i=0}^{10} b_i \cdot 2^i \qquad b_i \in \{0,1\}$$

(Ex : $1984 = 11111000000$ en binaire)

En changeant une seule chose à cette représentation, on peut représenter les nombres entiers de -1024 à $+1023$:

$$N = \underbrace{-b_{10} \cdot 2^{10}}_{\text{soit } 0\text{, soit } -1024} + \underbrace{\sum_{i=0}^{9} b_i\, 2^i}_{\text{nombres entiers de } 0 \text{ à } +1023} \qquad b_i \in \{0,1\}$$

Ex :
$$\begin{cases} 10000000000 = -1024, & 11111111111 = -1024+1023 = -1 \\ 00000000000 = 0, & 01111111111 = +1023 \\ 11111000000 = -1024+960 = -64 & \end{cases}$$

Remarques

- Dans les deux cas (nombres entiers positifs ou relatifs), le nombre d'éléments que l'on peut représenter avec n bits ne change pas : on décale juste la fenêtre à gauche !

- Le fait de n'avoir qu'un certain nombre de bits à disposition pour représenter les nombres entiers (dans un ordinateur, ceux-ci sont en effet enregistrés dans des registres de taille fixe) peut engendrer des dépassements de capacité ou "overflows" lors d'opérations arithmétiques (addition, soustraction, multiplication) sur ces nombres.

Représentation binaire des nombres réels

- Première remarque : il n'est pas possible de représenter tous les nombres réels dans un intervalle donné avec un nombre fixé de bits $\Rightarrow$ approximations nécessaires !
- Représentation en virgule flottante sur 64 bits :

$$X = (-1)^S \cdot M \cdot 2^E$$

$$\begin{array}{ll}
1 \text{ bit} & \left\{\begin{array}{l}
S = \text{bit de signe} \quad \in \{0,1\} \\[2ex]
M = \text{"mantisse"} = 1 + \sum_{j=1}^{52} B_j \, 2^{-j}, \quad B_1 \dots B_{52} \in \{0,1\} \\[2ex]
E = \text{"exposant"} = -b_{10} \cdot 2^{10} + \sum_{i=0}^{9} b_i \cdot 2^i, \quad b_0 \dots b_{10} \in \{0,1\}
\end{array}\right. \\
52 \text{ bits} & \\
11 \text{ bits} &
\end{array}$$

Avec cette représentation :

- la mantisse $M \in [1, 2]$, et on peut représenter 2^{52} nombres différents dans cet intervalle !
- l'exposant $E \in \{-1024, \dots, +1023\}$, donc $2^E \in \sim 10^{-300} \dots \sim 10^{+300}$

On peut donc représenter des nombres positifs et négatifs sur un intervalle gigantesque de -10^{+300} à $+10^{+300}$, et ce avec une précision relative de $2^{-52} \sim 10^{-16}$!!!

Malgré cela, il faut faire très attention lorsqu'on utilise cette représentation pour effectuer des soustractions ; on peut alors perdre soudainement de la précision !

Ne jamais tester non plus une condition du type "Est-ce que $X=0$?"

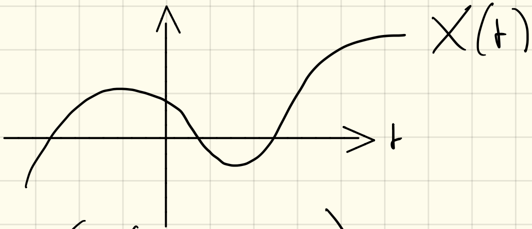
Echantillonnage de signaux

Maintenant qu'on a en quelque sorte "réglé" le problème de la représentation binaire des nombres réels, se pose la question de comment enregistrer un signal physique dans un ordinateur.

L'exemple le plus simple de signal physique est une fonction $X : \mathbb{R} \longrightarrow \mathbb{R}$, décrivant par exemple les variations de pression générées en un point de l'espace par une onde sonore provenant d'une source à quelque distance de là (p.ex. lorsque quelqu'un parle et qu'une autre personne écoute).

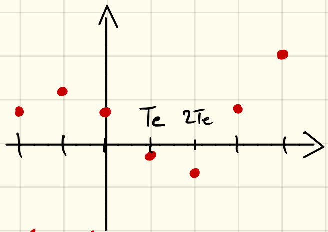
Comment donc enregistrer une fonction $X : \mathbb{R} \to \mathbb{R}$ sous forme de bits ? Il faut l'échantillonner, c'est-à-dire enregistrer les valeurs de la fonction à des intervalles réguliers (c'est mieux) dans le temps.

Ainsi, on passe de :



$(X(t),\ t \in \mathbb{R})$

à



$(X(nT_e),\ n \in \mathbb{Z})$

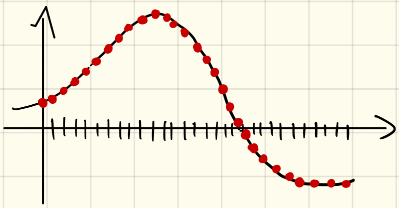
où T_e est appelée la période d'échantillonnage du signal (et on suppose que chaque valeur $X(nT_e) \in \mathbb{R}$ est encodée en virgule flottante avec un nombre fixe de bits, disons 64)

- La période d'échantillonnage T_e est la durée entre deux échantillons du signal, qui se mesure p.ex. en secondes.
- De manière équivalente, on définit $f_e = \frac{1}{T_e}$, la fréquence d'échantillonnage du signal, qui se mesure en Hertz $= \frac{1}{\text{sec}}$: la fréquence d'échantillonnage indique le nombre d'échantillons enregistrés par seconde.

Lorsqu'on échantillonne un signal $(X(t), t \in \mathbb{R})$, on a bien sûr envie de pouvoir le reconstruire après coup à partir de sa version échantillonnée $(X(nT_e), n \in \mathbb{Z})$.
La question est : à quelle fréquence f_e faut-il l'échantillonner pour que ça fonctionne ?

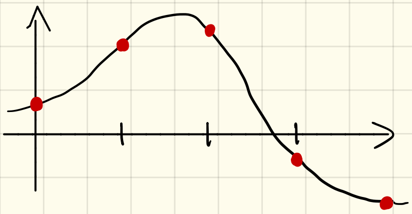
Deux problèmes peuvent se poser :

- échantillonnage à une fréquence fe trop élevée :



⟹ trop de données à enregistrer

- échantillonnage à une fréquence fe trop basse :



⟹ risque de perdre de l'information sur le signal

Pour tenter de répondre à la question ci-dessus, nous allons d'abord nous concentrer sur des signaux d'un type particulier : <u>les sinusoïdes</u>.

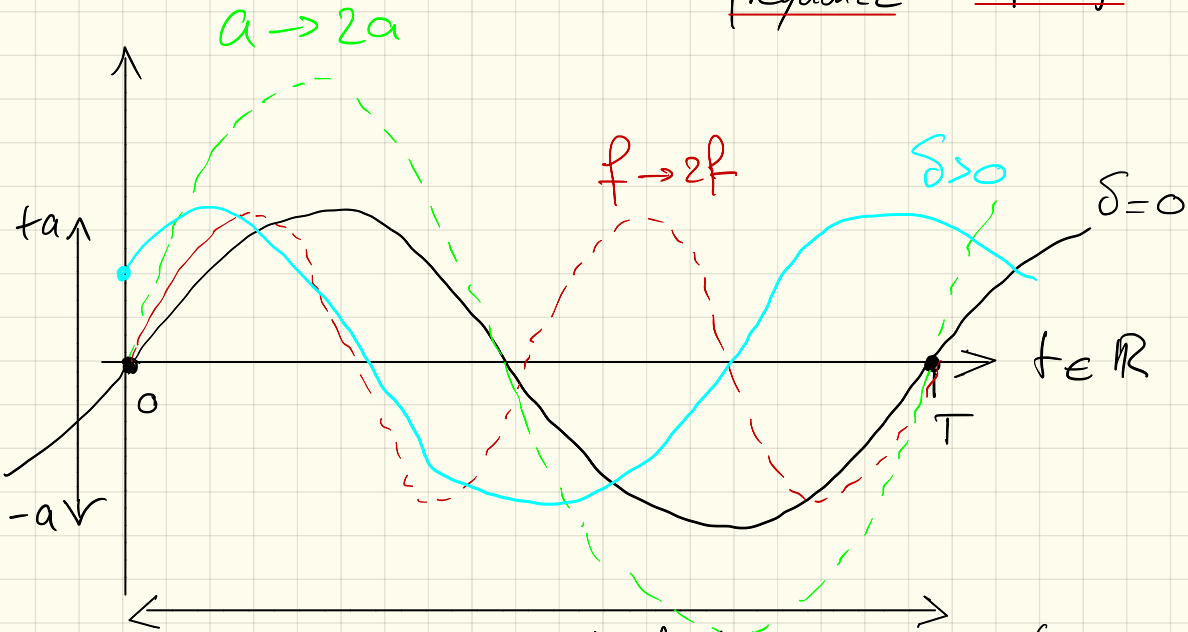
Ceci pour <u>deux raisons</u> principales :

1. Pour de tels signaux, un critère simple existe.

2. Il se trouve que tout signal peut être décomposé sous forme de sinusoïdes (comme nous allons le voir ci-dessous).

La sinusoïde : $X(t) = \boxed{a} \cdot \sin(2\pi \boxed{f} t + \boxed{\delta})$, $t \in \mathbb{R}$

amplitude fréquence déphasage



T est la période de la sinusoïde (mesurée en sec.)

$f = \frac{1}{T}$ est sa fréquence (mesurée en Hz)

Joseph Fourier, 1822 :

" Tout signal X est une somme de sinusoïdes ! "

Ceci veut dire que tout signal X s'écrit :

$$X(t) = \sum_{j=1}^{n} a_j \sin(2\pi f_j t + \delta_j) \quad , \quad t \in \mathbb{R}$$

(On triche un peu ici, car en réalité, la somme ci-dessus est parfois (souvent, même) infinie !)

On définit ainsi la bande passante de ce signal :

$$B = \max(f_1, f_2, \dots, f_n)$$

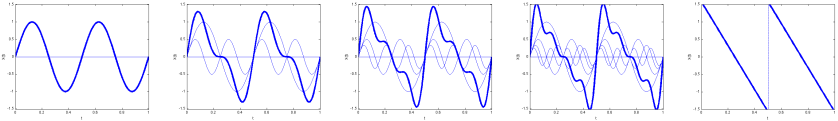
"Preuve visuelle" de l'énoncé de Fourier

Somme de sinusoïdes

$$X_n(t) = a_1 \sin(2\pi f_1\, t + \delta_1) + \ldots + a_n \sin(2\pi f_n\, t + \delta_n), \quad t \in \mathbb{R}$$

avec $a_j = 1/j$, $f_j = 2j$, $\delta_j = 0$.

- $n = 1$: $X_1(t) = \sin(4\pi t)$
- $n = 2$: $X_2(t) = \sin(4\pi t) + \frac{1}{2}\sin(8\pi t)$
- $n = 3$: $X_3(t) = \sin(4\pi t) + \frac{1}{2}\sin(8\pi t) + \frac{1}{3}\sin(12\pi t)$
- $n = 4$: $X_4(t) = \sin(4\pi t) + \frac{1}{2}\sin(8\pi t) + \frac{1}{3}\sin(12\pi t) + \frac{1}{4}\sin(16\pi t)$
- "$n = \infty$" : $X_\infty(t) = \sum_{j \geq 1} \frac{1}{j}\sin(4\pi j t)$

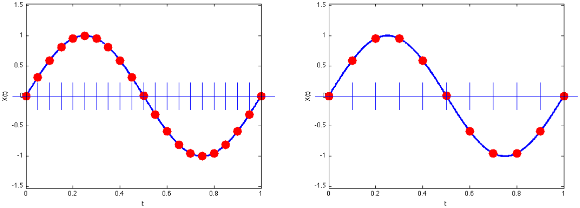


Echantillonnage d'une sinusoïde

On considère le signal $X(t) = \sin(2f\pi t)$ échantillonné à fréquence f_e.

Fixons d'abord $f = 1$ Hz et diminuons progressivement la fréquence d'échantillonnage f_e:

- $f_e = 20$ Hz
- $f_e = 10$ Hz

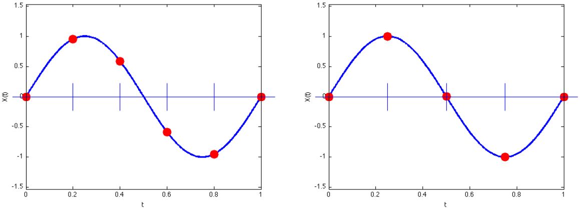


Echantillonnage d'une sinusoïde

On considère le signal $X(t) = \sin(2f\pi t)$ échantillonné à fréquence f_e.

Fixons d'abord $f = 1$ Hz et diminuons progressivement la fréquence d'échantillonnage f_e:

- $f_e = 5$ Hz
- $f_e = 4$ Hz

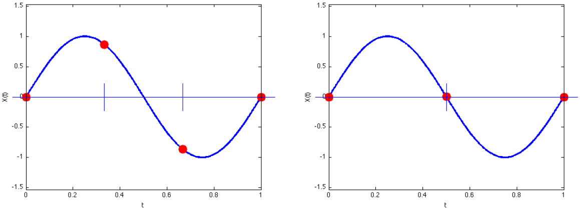


Echantillonnage d'une sinusoïde

On considère le signal $X(t) = \sin(2f\pi t)$ échantillonné à fréquence f_e.

Fixons d'abord $f = 1$ Hz et diminuons progressivement la fréquence d'échantillonnage f_e:

- $f_e = 3$ Hz
- $f_e = 2$ Hz



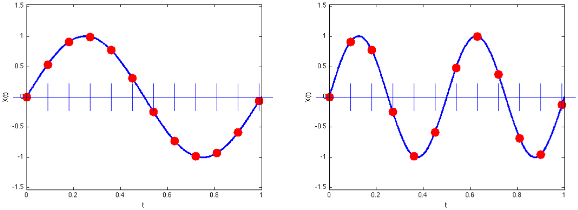
On a clairement un problème lorsque $f_e = 2$ Hz $(= 2f)$!

Echantillonnage d'une sinusoïde (bis)

On considère le signal $X(t) = \sin(2f\pi t)$ échantillonné à fréquence f_e.

Fixons maintenant $f_e = 11.11$ Hz (correspondant à $T_e = 0.09$ sec) et augmentons progressivement la fréquence f du signal:

- $f = 1$ Hz
- $f = 2$ Hz

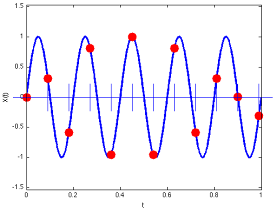


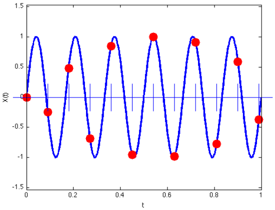
Echantillonnage d'une sinusoïde (bis)

On considère le signal $X(t) = \sin(2f\pi t)$ échantillonné à fréquence f_e.

Fixons maintenant $f_e = 11.11$ Hz (correspondant à $T_e = 0.09$ sec) et augmentons progressivement la fréquence f du signal:

- $f = 5$ Hz
- $f = 6$ Hz



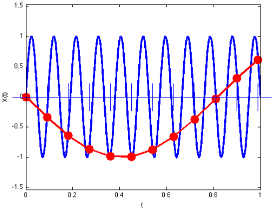


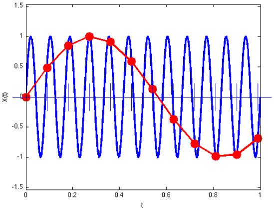
Echantillonnage d'une sinusoïde (bis)

On considère le signal $X(t) = \sin(2f\pi t)$ échantillonné à fréquence f_e.

Fixons maintenant $f_e = 11.11$ Hz (correspondant à $T_e = 0.09$ sec) et augmentons progressivement la fréquence f du signal:

- $f = 10.5$ Hz
- $f = 12$ Hz





$f_e < 2f$: le signal est sous-échantillonné $\longrightarrow$ effet stroboscopique!

Conclusion :

Dans les deux cas, on observe que quand la fréquence d'échantillonnage f_e est plus petite ou égale à $2f$, il est difficile (et même impossible) de reconstruire la sinusoïde d'origine $(x(t), t \in \mathbb{R})$ à partir de ses valeurs échantillonnées $(x(nT_e), n \in \mathbb{Z})$.

La condition nécessaire pour échantillonner correctement une sinusoïde de fréquence f est donc :

$$\boxed{f_e > 2f}$$

Qu'en est-il si le signal X est la somme de deux sinusoïdes ?

$$X(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) \quad , \quad t \in \mathbb{R}$$

Pour pouvoir reconstruire correctement chaque sinusoïde à partir de sa version échantillonnée, il faut que $f_e > 2 f_1$ et $f_e > 2 f_2$.

On obtient donc la condition nécessaire :

$$\boxed{f_e > 2 \max(f_1, f_2)}$$

Et finalement, lorsque le signal est une somme de sinusoïdes :

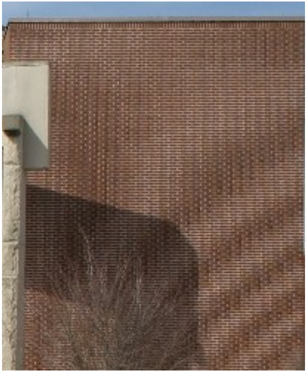
$$x(t) = \sum_{j=1}^{n} a_j \sin(2\pi f_j t + \delta_j) \quad , \quad t \in \mathbb{R}$$

Bande passante : $B = \max(f_1, f_2, \ldots, f_n)$

La condition nécessaire pour un échantillonnage correct du signal x devient : $\boxed{f_e > 2B}$

Nous verrons la semaine prochaine que cette condition est aussi suffisante pour une reconstruction parfaite du signal x : c'est l'énoncé du théorème d'échantillonnage.

Et pour finir, revenons à l'effet stroboscopique...



Aussi:

https://www.youtube.com/watch?v=LVwmtwZLG88

https://www.youtube.com/watch?v=2IghwseoISc