

# II

## Traitement numérique des signaux

*«In fact, any operation that can be completely described to the required accuracy (if numerical) in a finite number of steps using the words « if », « or », « and », etc, can be done automatically with relays.»*

A Symbolic Analysis of Relay and Switching  
Circuits, 1937

CLAUDE ELWOOD SHANNON  
(1916-2001)

### PLAN DU CHAPITRE

<b>I</b>	<b>Echantillonnage - Conversion analogique numérique (CAN)</b>	<b>3</b>
I.1	Principe général	3
I.2	Quelques précisions sur l'échantillonneur et le C.A.N.	4
	a - Echantillonneur-bloqueur par capacité	4
	b - CAN à quantification uniforme : pleine échelle (Full Scale) - erreur de codage	5
I.3	Analyse spectrale d'un signal échantillonné	6
	a - Rappel : spectre d'un peigne de Dirac	6
	b - Spectre d'un signal échantillonné $f_{ech}(t)$	7
	c - Reconstitution d'un signal à partir de son échantillonnage (technique passe-bas) - théorème de Nyquist-Shannon	8
	d - Non respect du théorème de Nyquist-Shannon : le repliement de spectre et les fréquences "fantômes" (Expérience de cours/Simulation Python)	10
<b>II</b>	<b>Notions de base sur le filtrage numérique des signaux</b>	<b>11</b>
II.1	Principe	11
II.2	Mise en oeuvre	12

a - Filtrage numérique temporel (Expérience de cours/Simulation Python) . . .	12
b - Filtrage numérique spectral . . . . .	15
<b>III Complément TP : notions de fenêtrage . . . . .</b>	<b>15</b>
III.1 Principe . . . . .	15
III.2 Exemples d'exploitation . . . . .	16

---

## I Échantillonnage - Conversion analogique numérique (CAN)

### I.1 Principe général

Tout traitement d'un signal par un système informatique nécessite dans un premier temps son acquisition sous forme numérique. Cette étape consiste à enregistrer les valeurs prises par le signal à intervalles de temps réguliers. Une telle technique porte le nom d'**échantillonnage**.

Chaque échantillon prélevé est immédiatement converti en valeur binaire par un Convertisseur Analogique-Numérique ou C.A.N. puis stocké en mémoire.

Le signal ainsi numérisé est enregistré sous forme de liste ou tableau de valeurs numériques, généralement des flottants, et accompagnés d'erreurs d'arrondis liées à la quantification du CAN (cf *quantification* plus bas dans ce cours).

QUESTION : comment fonctionne schématiquement la chaîne d'acquisition échantillonneur-bloqueur, puis CAN ?

RÉPONSE : les valeurs de  $f(t)$  sont d'abord prélevées tous les  $T_e = \frac{1}{F_e}$  à l'aide d'un échantillonneur-bloqueur par capacité, avec  $T_e$  appelée période d'échantillonnage ( $F_e$  "fréquence d'échantillonnage"). On enregistre en mémoire la suite des nombres suivants :  $[f(0), f(T_e), f(2T_e), \dots, f(nT_e), \dots]$  à l'aide du CAN (dont on ne détaillera pas le principe dans ce cours -> cf TD).

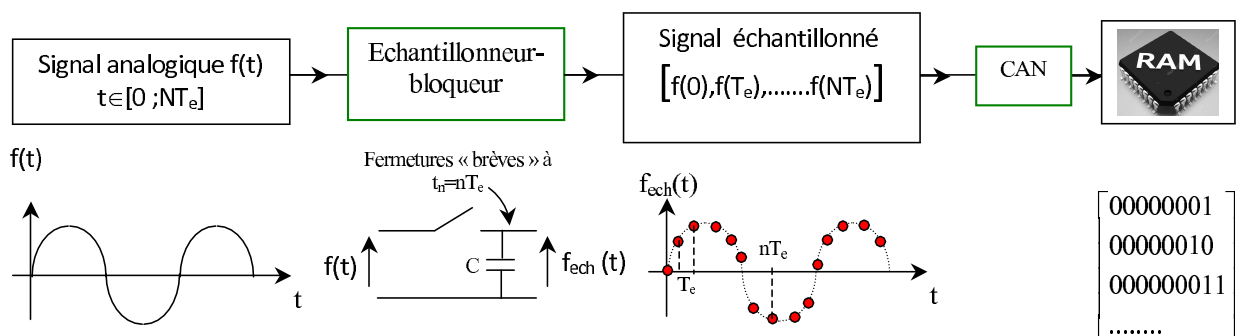


FIGURE II.1 – Principe de l'échantillonnage

Le signal échantillonné peut-être représenté par une fonction mathématique  $f_{ech}(t)$ .

ÉCRITURE FORMELLE DE  $f_{ech}(t)$  :

- pour l'échantillon  $f_{ech}(nT_e)$  pris seul à la date  $t_n = nT_e$  la valeur est donnée par le produit suivant :

$$f_{ech}(nT_e) = f(nT_e) \times \delta(t - nT_e) \quad \text{avec } \delta(t - nT_e) \text{ pic de Dirac d'amplitude 1 centré sur la date } nT_e$$

$$= f(t) \times \delta(t - nT_e) \quad \text{car } \delta(t - nT_e) = 0 \text{ pour } t \neq nT_e$$

- pour un signal continu  $f(t)$  ( $t \in ]-\infty; +\infty[$ ), la totalité du signal échantillonné  $f_{ech}(t)$  est donnée par la sommation suivante :

$$\begin{aligned}
 f_{ech}(t) &= \sum_{n=-\infty}^{+\infty} f_{ech}(nT_e) = \sum_{n=-\infty}^{+\infty} f(nT_e) \times \delta(t - nT_e) \\
 &= \sum_{n=-\infty}^{+\infty} f(t) \times \delta(t - nT_e) = f(t) \times \underbrace{\sum_{n=-\infty}^{+\infty} \delta(t - nT_e)}_{\text{peigne de Dirac période } T_e}
 \end{aligned}$$

L'opération d'échantillonnage peut donc être résumée très schématiquement ainsi :

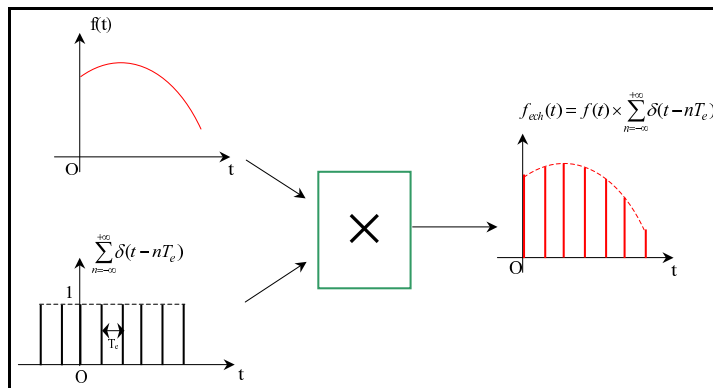


FIGURE II.2 – Schema de principe de l'échantillonnage

## I.2 Quelques précisions sur l'échantillonneur et le C.A.N.

### a - Echantillonneur-bloqueur par capacité

L'échantillonneur-bloqueur a pour rôle de fournir au C.A.N. les valeurs des tensions discrètes prises à intervalles réguliers.

PRINCIPE :

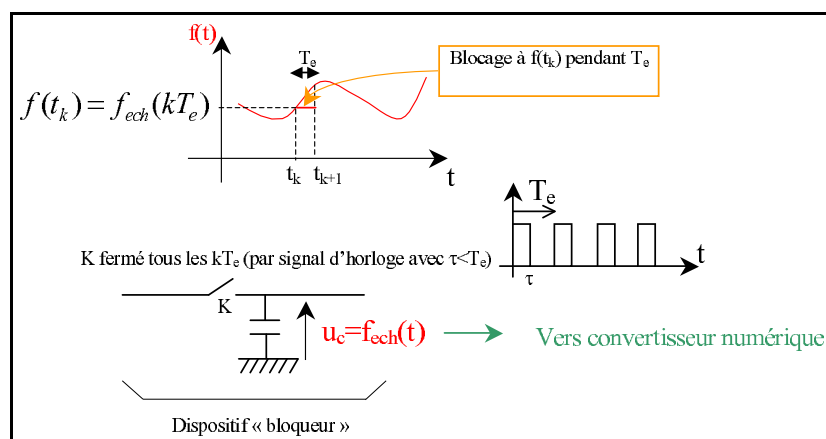


FIGURE II.3 – Principe échantillonneur-Bloqueur

Le fonctionnement obéit à des cycles de deux phases chacun, gouvernés par le signal d'horloge qui "pilote" l'interrupteur  $K$  :

- **Phase 1** :  $K$  fermé de  $t_k$  à  $t_k + \alpha \cdot T_e = t_k + \tau$  : la tension aux bornes du condensateur suit les variations du signal  $f(t)$  pendant  $\alpha \cdot T_e = \tau$  ; c'est la phase d'échantillonnage ou *Sample*  $\rightarrow U_c(t_k + \tau) = f(t_k + \tau)$
- **Phase 2** :  $K$  ouvert de  $t_k + \alpha \cdot T_e = t_k + \tau$  à  $t_{k+1} = t_k + T_e$  : la tension aux bornes du condensateur reste constante ; c'est la phase de **blocage** ou *Hold*  $\rightarrow U_c(t \in [t_k + \alpha \cdot T_e, t_{k+1}]) = f(t_k + \alpha \cdot T_e) = f_{ech}(t_k + \alpha \cdot T_e) \stackrel{\alpha \ll 1}{\simeq} f_{ech}(t_k)$

**NB** : la phase d'échantillonnage est très brève, soit  $\tau = \alpha \cdot T_e \ll T_e$  soit  $\alpha \ll 1$ .

## b - CAN à quantification uniforme : pleine échelle (Full Scale) - erreur de codage

### ■ PLEINE ÉCHELLE ET QUANTIFICATION

#### HYPOTHÈSE :

- le signal échantillonné  $f_{ech}(pT_e)$  est compris entre  $f_{ech_{min}}$  et  $f_{ech_{max}}$  ; l'intervalle  $V_{PE} = f_{ech_{max}} - f_{ech_{min}}$  est appelé **Pleine Echelle** ou bien **Full Scale** en anglais.
- le CAN est à  $N$  bits ce qui lui permet de représenter  $2^N$  nombres différents compris entre 0 à  $2^N - 1$  codant chacun pour une valeur analogique comprise dans l'intervalle de pleine échelle  $[f_{ech_{min}}, f_{ech_{max}}]$ .

Dans le cas d'un CAN à quantification uniforme, la **Pleine Echelle**  $V_{PE}$  est divisée en autant de plages d'égale dimension qu'il y a d'états possibles de la sortie numérique. Chaque plage est associée à une valeur binaire représentant la tension analogique d'entrée. La relation entre le signal d'entrée et la valeur du signal numérique de sortie est appelée **caractéristique du CAN** :

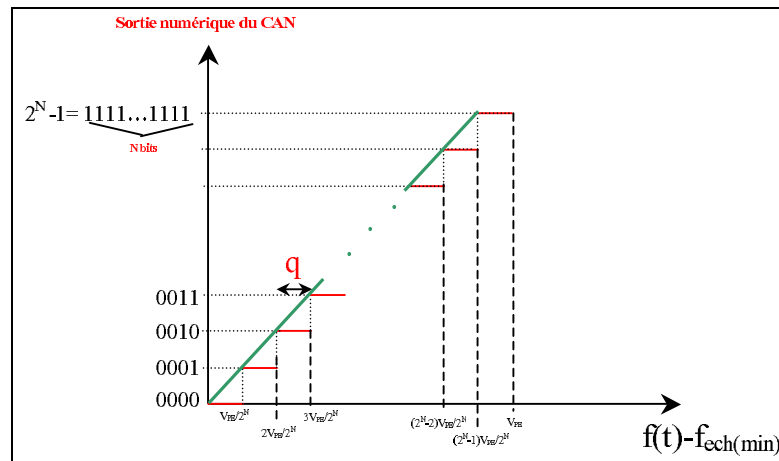


FIGURE II.4 – Caractéristique de transfert idéale d'un CAN à quantification uniforme

**A RETENIR :**

**Définition I-1:** PAS DE QUANTIFICATION OU *quantum* OU *lsb* (POUR LEAST SIGNIFICANT bit)

On appelle pas de quantification  $q$  l'incertitude sur chaque valeur numérisée du signal échantillonné  $f_{ech}$  avec :

$$q = \frac{f_{ech_{max}} - f_{ech_{min}}}{2^N}$$

CONSEQUENCE IMMÉDIATE : la quantification engendre une perte d'information d'autant plus grande que le pas  $q$  est important  $\Rightarrow$  il faut  $q$  petit, donc on code les signaux avec une profondeur de bits  $N$  élevée dans la mesure du possible.

**Exemple** : dans le cas du codage CD on prends  $N = 16$  bits, soit  $N = 2^{16} - 1 = 65536$  valeurs possibles. Si par exemple le CAN exploité pour l'encodage du CD admet un intervalle de pleine échelle de  $10\text{ V}$ , alors le pas de quantification vaudra :

$$q = \frac{10}{2^{16}} = 152,6\text{ }\mu\text{V}$$

■ ERREUR DE QUANTIFICATION OU DE CODAGE

**Définition I-2:** ERREUR DE QUANTIFICATION (CAN À CONVERSION LINÉAIRE UNIFORME)

Il s'agit de la différence entre la valeur du signal analogique d'entrée ( $x(t) = f(t) - f_{ech_{min}}$ ) et la valeur numérique de sortie correspondant  $x_{ech}(t)$ , l'erreur de codage est exprimée en *lsb*, et est forcément comprise entre 0 et 1 *lsb* :

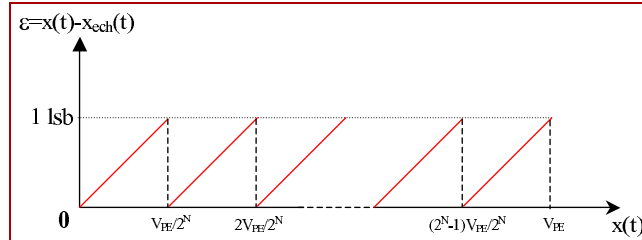


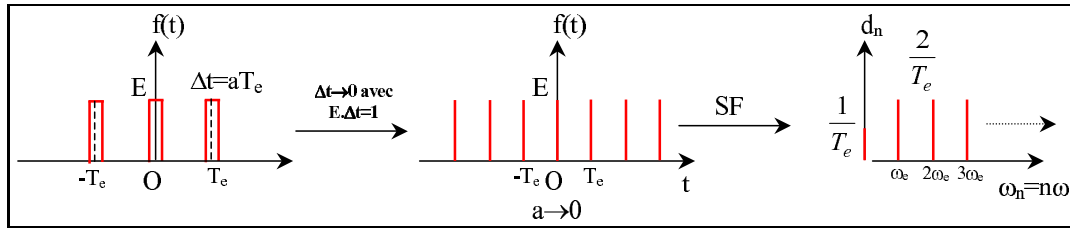
FIGURE II.5 – Erreur de codage pour la quantification linéaire uniforme

L'erreur de quantification est donc indissociable des CAN et celle-ci diminue lorsque la profondeur de codage  $N$  augmente.

### I.3 Analyse spectrale d'un signal échantillonné

#### a - Rappel : spectre d'un peigne de Dirac

Le peigne de Dirac correspond ici à un train d'impulsion de largeur tendant vers 0 ( $\Delta t \rightarrow 0$ ) de période  $T_e$  (période d'échantillonnage) et d'amplitude  $E$  (on rappelle la contrainte  $E \cdot \Delta t = 1$ ) :


 FIGURE II.6 – Série de Fourier d'un Peigne de Dirac ( $a = \frac{\Delta t}{T_e} \rightarrow 0$ )

On rappelle les coefficients de Fourier d'un tel signal :

$$\begin{cases} d_0 = |c_0| \xrightarrow{|sinc(na\pi)| \rightarrow 1 \forall n} Ea = E \frac{\Delta t}{T_e} = \frac{1}{T_e} \\ d_{n \geq 1} = |2c_n| \xrightarrow{|sinc(na\pi)| \rightarrow 1 \forall n} 2E \frac{\Delta t}{T_e} = \frac{2}{T_e} \\ \varphi_n = \arg(c_n) = 0 \end{cases}$$

Ainsi, la SF du peigne de Dirac de période  $T_e$  est :

$$\mathcal{P}_D(t) = E \cdot \sum_{n=-\infty}^{+\infty} \delta(t - nT_e) = \frac{1}{T_e} + \sum_{n=1}^{+\infty} \frac{2}{T_e} \cdot \cos(2\pi n \frac{t}{T_e})$$

### b - Spectre d'un signal échantillonné $f_{ech}(t)$

#### • CAS D'UN SIGNAL SINUSOÏDAL PUR

Considérons dans un premier temps un signal sinusoïdal pur de fréquence  $f_0$  :  $f(t) = A \cos(2\pi f_0 t)$  que l'on échantillonne à  $T_e$ . Le signal échantillonné s'écrit :

$$f_{ech}(t) = A \cos(2\pi f_0 t) \times [\text{Peigne Dirac}] = A \cos(2\pi f_0 t) \times \left[ \frac{1}{T_e} + \sum_{n=1}^{+\infty} \frac{2}{T_e} \cdot \cos(2\pi n F_e t) \right]$$

soit :

$$f_{ech}(t) = \frac{A}{T_e} \left[ \overbrace{\cos(2\pi f_0 t) + \sum_{n=1}^{+\infty} \cos(2\pi(nF_e - f_0)t) + \cos(2\pi(nF_e + f_0)t)}^{\text{décomposition en somme d'harmoniques}} \right]$$

Le spectre du signal est donc constitué d'une infinité de composantes toutes de même amplitude :

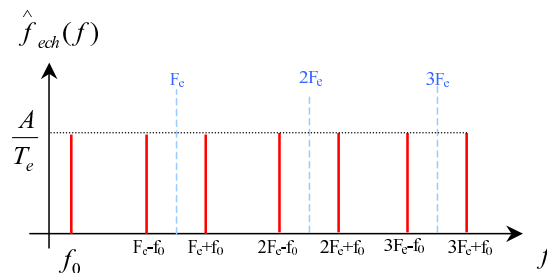


FIGURE II.7 – Spectre du signal sinusoïdal échantillonné

## • CAS D'UN SIGNAL QUELCONQUE

Considérons maintenant un signal quelconque  $s(t)$  en particulier non périodique, donc de spectre  $\hat{s}(f)$  continu. Ce dernier peut par exemple avoir l'allure suivante :

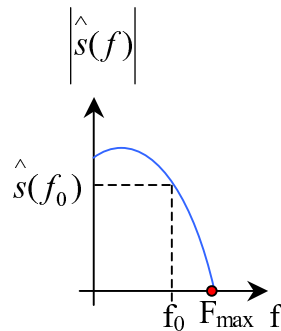


FIGURE II.8 – Spectre du signal  $s(t)$  non périodique (seules les fréquences positive de la TF sont représentées)

**Remarque I-1:** CHOIX DE  $F_{max}$ 

**NB :**  $F_{max}$  représente la fréquence maximale présente dans le spectre du signal. Lorsqu'il s'agit d'un signal périodique, nous savons que l'étendue spectrale de celui-ci est infinie et que l'amplitude des harmoniques décroît avec la fréquence ; dans ce cas, on retient pour  $F_{max}$  la fréquence du dernier harmonique de poids significatif dans le spectre <sup>a</sup>.

a. pour un signal crêteau ( $\sim \frac{1}{n}$ ), abandonner le rang 20 est suffisant ; pour un signal triangulaire ( $\sim \frac{1}{n^2}$ ), ce sera le rang 5

On peut par exemple isoler la composante de fréquence  $f_0$ , d'amplitude  $|\hat{s}(f_0)|$  ; d'après la démarche ci-dessus, le spectre de cette composante une fois échantillonnée contient les fréquences  $nF_e \pm f_0$ . En reproduisant ce raisonnement pour l'ensemble des composantes spectrales du signal  $s(t)$ , on obtient le spectre  $\hat{s}_{ech}(f)$  représenté ci-dessous, qui reproduit donc le motif spectral tous les  $nF_e$  :

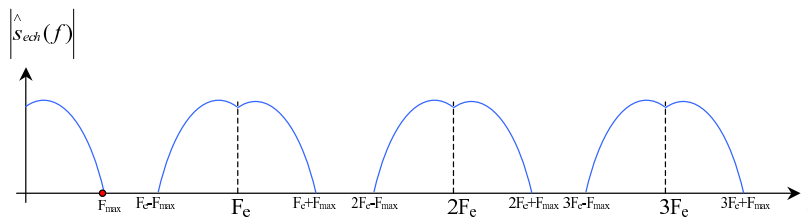


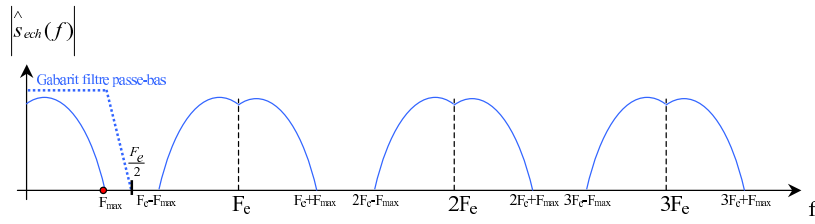
FIGURE II.9 – Spectre  $\hat{s}_{ech}(f)$  du signal échantillonné  $s_{ech}(t)$

### c - Reconstitution d'un signal à partir de son échantillonnage (technique passe-bas) - théorème de Nyquist-Shannon

QUESTION : comment obtenir le spectre  $\hat{s}(\omega)$  du signal  $s(t)$  à partir du signal échantillonné ?

RÉPONSE : en réalisant un filtrage passe-bas du signal échantillonné :




 FIGURE II.10 – Filtrage passe-bas du signal échantillonné  $s_{ech}(t)$ 

Pour restituer correctement le spectre du signal original après filtrage, il faut s'assurer qu'aucune portion du "massif spectral" centré sur  $F_e$  ne pénètre dans le "massif spectral" centré sur 0 que l'on doit récupérer ; la condition de non chevauchement s'écrit naturellement :

$$F_{Max} < F_e - F_{Max} \quad \text{soit} \quad \boxed{2F_{Max} < F_e}$$

On en déduit le **théorème de Nyquist-Shannon** imposant des conditions pour réaliser un échantillonnage correct :

**Propriété I-1: THÉORÈME DE NYQUIST-SHANNON**

La reconstitution *parfaite*, i.e. sans perte d'information (par synthèse de Fourier), d'un signal à partir de son échantillonnage à la fréquence  $F_e$  n'est possible qu'à condition qu'  $F_e$  soit au minimum deux fois supérieure à la plus haute fréquence présente dans le spectre du signal, ou bien la plus haute fréquence que l'on retiendra comme significative dans le spectre :

$$\text{Echantillonnage correct} \Leftrightarrow F_e > 2F_{Max} \quad \text{soit} \quad F_{Max} < \frac{F_e}{2} = F_{Ny}$$

en posant  $F_{Ny} = \frac{F_e}{2}$  fréquence de Nyquist.

**NB :** la fréquence de coupure du filtre est souvent fixée à  $f_c = F_{Ny} = \frac{F_e}{2}$  (cf. fig II.10)

**EXEMPLE :** cas du CD-AUDIO

Pour un CD-AUDIO, on a  $F_e = 44,1 \text{ kHz} \simeq 44 \text{ kHz}$  et le spectre des audio-fréquences  $F \in [20 \text{ Hz} \text{ } 20 \text{ kHz}]$  ; l'échantillonnage est donc correct pour une restitution ultérieure du signal :

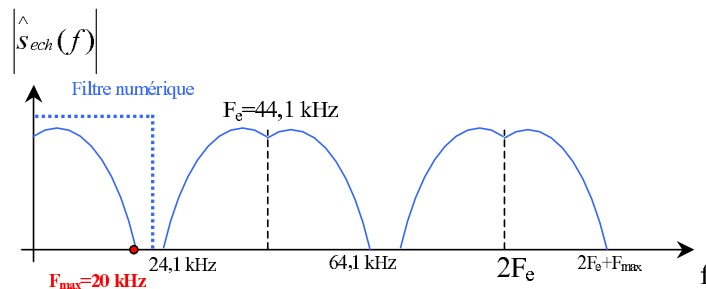


FIGURE II.11 – Échantillonnage à la norme CD-audio

### d - Non respect du théorème de Nyquist-Shannon : le repliement de spectre et les fréquences "fantômes" (Expérience de cours/Simulation Python)

EXPÉRIENCE DE COURS : illustration du repliement sur un signal sinusoïdal  $f_0 = 1 \text{ kHz}$ .

Reprenons le cas simple de l'échantillonnage d'un signal sinusoïdal et fixons sa fréquence à  $f_0 = 1 \text{ kHz}$ .

Deux situations peuvent présenter :

- La fréquence d'échantillonnage est  $F_e > 2F_{Max} = 2 \text{ kHz}$  par exemple  $F_e = 3 \text{ kHz} \Rightarrow$  **l'échantillonnage est correct (condition de Nyquist-Shannon respectée)**
- La fréquence d'échantillonnage est  $F_e < 2F_{Max} = 2 \text{ kHz}$  par exemple  $F_e = 1,5 \text{ kHz} \Rightarrow$  **l'échantillonnage est incorrect ! (condition de Nyquist-Shannon violée)**

Dans ce second cas, le spectre du signal échantillonné est le suivant :

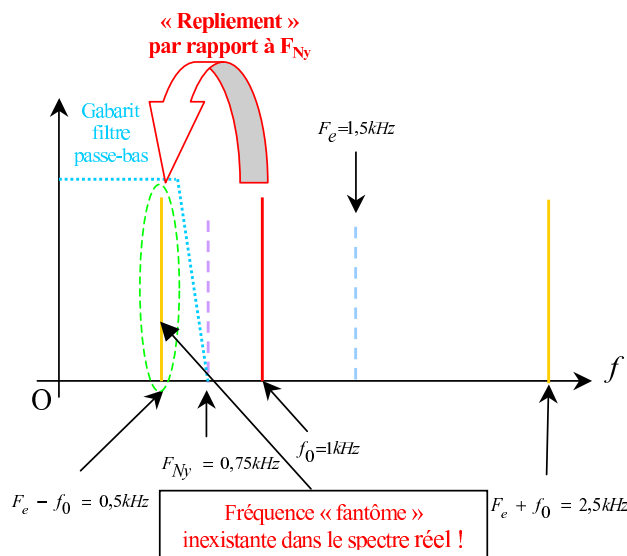


FIGURE II.12 – Phénomène de repliement de spectre pour un signal sinusoïdal

Il présente une raie dans la zone filtrée  $[0, F_{Ny} = \frac{F_e}{2} = 0,75 \text{ kHz}]$  à la fréquence  $F_e - f_0 = 1,5 \text{ kHz} - 1 \text{ kHz} = 0,5 \text{ kHz}$ . Cette raie n'existe évidemment pas dans le signal originel, et est symétrique de la raie du signal à  $f_0$  par rapport à la fréquence de Nyquist ; en effet, la fréquence médiane entre cette raie fantôme et la fréquence  $f_0$  est :

$$\frac{(F_e - f_0) + f_0}{2} = \frac{F_e}{2} = F_{Ny}$$

**A retenir** : Si  $F_e < 2f_0$  alors il apparaît  $f_{\text{fantôme}}$  symétrique de  $f_0$  par rapport à  $F_{Ny} = \frac{F_e}{2}$  :

On parle de **repliement de spectre** par rapport à la fréquence de Nyquist.

**NB** : dans le cas d'un signal non périodique (spectre continu), la situation de repliement correspond à l'allure spectrale suivante :

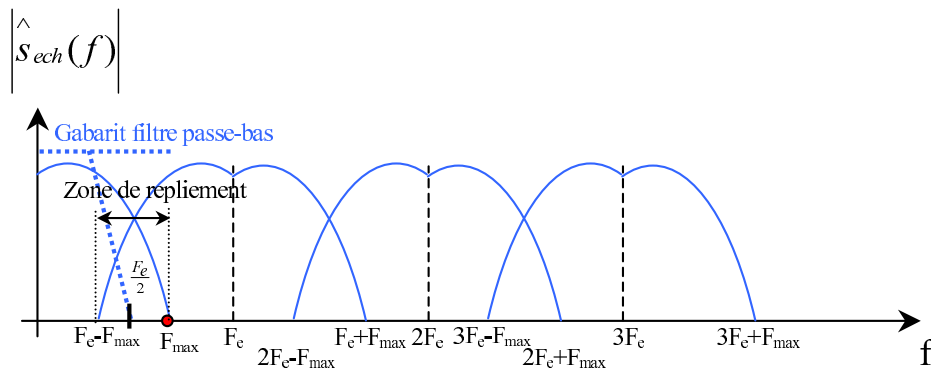


FIGURE II.13 – Phénomène de repliement de spectre

**Remarque I-2: FILTRAGE ANTI-REPLIEMENT**

Pour les signaux périodiques dont le spectre est non borné supérieurement, on procède généralement à un filtrage passe-bas avant échantillonnage appelé **FILTRAGE ANTI-REPLIEMENT**. Ceci permet de poser *arbitrairement* une borne supérieure au spectre, i.e. de définir  $F_{Max}$  afin d'assurer le respect rigoureux de la condition de Nyquist-Shannon lors du choix de la fréquence d'échantillonnage.

## II Notions de base sur le filtrage numérique des signaux

### II.1 Principe

La majorité des dispositifs modernes de traitement et analyse de signal, notamment en Hifi (élimination du bruit, enrichissement sélectif du spectre graves-médiums-aigües, effets de réverbération etc...) sont des dispositifs numériques travaillant donc sur des signaux préalablement échantillonnés. Ceci permet une latitude totale dans la nature des traitements que l'on peut faire subir aux signaux puisqu'ils sont réalisés par des algorithmes programmés sur ordinateur.

Deux opérations de base du traitement numérique du signal en Hifi sont les filtrages passe-haut (élimination du bruit très basse fréquence type "Rumble"), et passe-bas (élimination de sifflements haute fréquence).

**Question :** quel est le principe de filtrage d'un signal échantillonné ?

2 méthodes :  $\left[ \begin{array}{l} \text{filtrage temporel} \\ \text{filtrage spectral/fréquentiel} \end{array} \right.$

La suite expose les méthodes de filtrage numériques temporel et fréquentiel dont les principes sont résumés dans les synopses ci-dessous :

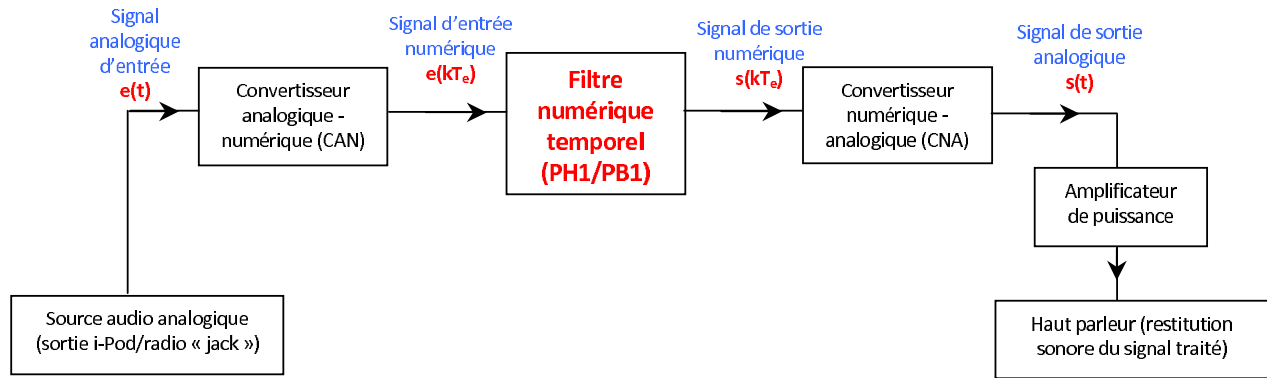


FIGURE II.14 – Chaîne complète de filtrage numérique temporel d'un signal audio

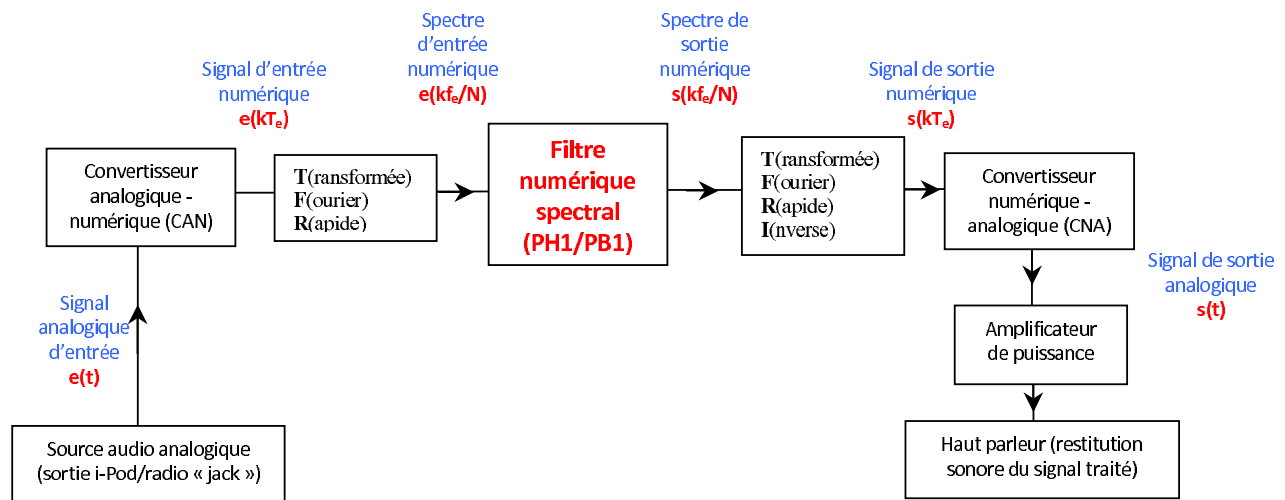


FIGURE II.15 – Chaîne complète de filtrage numérique fréquentiel d'un signal audio

## II.2 Mise en oeuvre

### a - Filtrage numérique temporel (Expérience de cours/Simulation Python)

Le signal à traiter présente deux composantes sinusoïdales l'une BF à  $f_b = 100 \text{ Hz}$  et l'autre HF à  $f_h = 4 \text{ kHz}$  :

$$e(t) = U_{0_b} \cos(2\pi f_b t) + U_{0_h} \cos(2\pi f_h t)$$

**Objectif :** On se propose de réaliser par exemple l'élimination de la composante HF à l'aide d'un **filtrage numérique temporel passe-bas**. Les différentes étapes que vous exécuterez en TP sont les suivantes :

- Acquisition du signal par conversion analogique-numérique (CAN) avec la fréquence d'échantillonnage  $f_e = \frac{1}{T_e} \geq 100 \text{ kHz} > 2f_h$  (cf th. de Shannon) avec LatisPro  $\rightarrow e(kT_e) \quad (k \in \mathbb{N})$ .
- Enregistrement du signal échantillonné au format .csv (séparation des données par une virgule), sous forme de couples  $(kT_e, e(kT_e))$

- Lecture du fichier .csv et **Filtrage numérique temporel** par un programme Python du signal échantillonné  $e(kT_e)$ , par exemple **passé-bas 1<sup>er</sup> ordre**  $e(kT_e) \xrightarrow{\text{passé bas}} s(kT_e)$
- Enregistrement par python du signal filtré  $s(kT_e)$  au format .csv
- Ouverture du fichier signal filtré par LatisPro
- Conversion numérique-analogique (CNA) du signal  $s(kT_e) \rightarrow s(t)$  et restitution par HP.

■ CALCUL DE LA "FORMULE" DE FILTRAGE DU PASSE-BAS  $e(kT_e) \xrightarrow{\text{PASSE BAS}} s(kT_e)$

La fonction de transfert d'un passe bas 1<sup>er</sup> ordre passif de pulsation de coupure  $\omega_c$  est de forme :

$$H(j\omega) = \frac{1}{1 + j\frac{\omega}{\omega_c}}$$

dont l'équation différentielle correspondante est :

$$\frac{ds(t)}{dt} + \omega_c \cdot s(t) = \omega_c \cdot e(t)$$

On intègre cette équation différentielle entre les instants "discrets"  $kT_e$  et  $(k+1)T_e$  et l'on réordonne les deux membres pour obtenir :

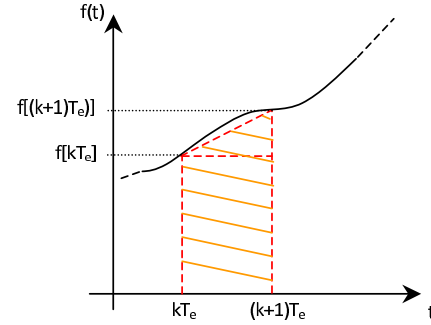


FIGURE II.16 – Intégration par méthode des trapèzes

$$s(k+1) = s(k) + \omega_c \underbrace{\int_{kT_e}^{(k+1)T_e} [e(t) - s(t)] \cdot dt}_{\text{aire sous la courbe} \Rightarrow \text{méthode des trapèzes}} \quad (\text{II.1})$$

or l'intégrale entre deux instants  $kT_e$  et  $(k+1)T_e$  d'une fonction  $f(t)$  peut être approchée par méthode des trapèzes avec :

$$\int_{kT_e}^{(k+1)T_e} f(t) \cdot dt \simeq \underbrace{T_e \cdot f(kT_e)}_{\text{aire du rectangle inférieur}} + \underbrace{\frac{T_e}{2} [f((k+1)T_e) - f(kT_e)]}_{\text{aire du triangle supérieur}} = \frac{T_e}{2} [f((k+1)T_e) + f(kT_e)]$$

Compte tenu de ceci, l'équation II.1 devient :

$$s(k+1) = s(k) + \frac{\omega_c T_e}{2} [e(k) + e(k+1) - s(k) - s(k+1)]$$

soit finalement :

$$s(k+1) = \frac{2 - \omega_c T_e}{2 + \omega_c T_e} \cdot s(k) + \frac{\omega_c T_e}{2 + \omega_c T_e} [e(k) + e(k+1)]$$

Cette dernière relation de récurrence permet d'obtenir le signal numérique (échantillonné) de sortie du filtre.

Le code Python complet sera commenté "en live" :

Script Python Filtrage numérique passe-bas et passe haut 1<sup>er</sup> ordre

```

tiny
1  # -*- coding: utf-8 -*-
2  from math import *
3  import numpy as np
4  import matplotlib.pyplot as plt
5  #ouverture du fichier de donnees source
6  mesdonnees=open('Donnees_formatees_CSV.csv','r')
7  #Donnees numeriques
8  Te=0.00001 #Période d'échantillonnage
9  wc=2*np.pi*200.0 #Pulsation de coupure des passe-bas et passe-haut
10 #Tableaux vides pour signal entree pour enregistrement de 4096 donnees
11 t=np.zeros(4096)
12 e=np.zeros(4096)
13 #Lecture en-tete du fichier de donnees
14 entete=mesdonnees.readline().rstrip('\n\r').split(',')
15
16 #Initialisation d'un compteur et du max d'écchelle graphique
17 k,max=(0,0)
18 #Boucle de construction de e par lecture du fichier source
19 for Ligne in mesdonnees:
20     tL,eL=Ligne.rstrip('\n\r').split(",") #Lit la ligne, supprime espace
21     #et retour chariot, coupe a la virgule et stocke le temps et la valeur signal
22     t[k]=float(tL) # affectation de la k-ième valeur du temps
23     e[k]=float(eL) # affectation de la k-ième valeur du signal
24     if e[k]>max:
25         max=1.05*e[k] #Max est une variable d'ajustement automatique de l'echelle
26     k+=1
27 #Fermeture du fichier source
28 mesdonnees.close()
29 #Tableaux vides pour signal sortie pour enregistrement 4096 donnees
30 spb1=np.zeros(4096)
31 sph1=np.zeros(4096)
32 #ouverture des fichiers de donnees sortie
33 masortie_PB1=open('Donnee_sortie_PB1_CSV.csv','w')
34 masortie_PH1=open('Donnee_sortie_PH1_CSV.csv','w')
35 #Ecriture des signaux de sortie des filtres par récurrence
36 for k in range(4095):
37     spb1[k+1]=((2-wc*Te)/(2+wc*Te))*spb1[k]+((wc*Te)/(2+wc*Te))*(e[k]+e[k+1])#PB
38     sph1[k+1]=((2-wc*Te)/(2+wc*Te))*sph1[k]+(2/(2+wc*Te))*(e[k+1]-e[k])#PH
39     masortie_PB1.write(str(t[k])+','+str(spb1[k])+'\n') #Ecriture de ligne dans le fichier de sortie Passe
40     bas1 apr's conversion en chaine de caractere
41     masortie_PH1.write(str(t[k])+','+str(sph1[k])+'\n') #Ecriture de ligne dans le fichier de sortie passe
42     haut1 apr's conversion en chaine de caractere
43 #Fermeture des fichiers de sortie passe-bas1 et passe-haut1
44 masortie_PB1.close()
45 masortie_PH1.close()
46 #Trace evolution des tensions entree et sortie
47 plt.grid()
48 plt.xlabel(r'$t(s)$', fontsize=10)
49 plt.ylabel(r'$s(t)$ (en V)$', fontsize=10, rotation=0)
50 plt.title(r"Tension de sortie du filtre", size=20)
51 plt.axis([t[2048], t[4095], -max, max])
52 #entree=plot(t,e)
53 #sortie_PB1=plot(t,spb1)
54 #sortie_PH1=plot(t,sph1)
55 plt.show()

```

Listing

II.1

Sources\_Python/Filtrage\_numerique\_version\_2\_boucles\_pour\_compil\_latex.py

**NB :** le fichier Python est disponible sur le site MP3, et libre de modification/amélioration si le cœur vous en dit.

- CALCUL DE LA "FORMULE" DE FILTRAGE DU PASSE-HAUT  
A faire en live à titre d'exercice.

### b - Filtrage numérique spectral

Même exemple mais cette fois **par la technique fréquentielle**  $\Rightarrow$  réalisation en TP à l'aide d'un programme python fourni.

PRINCIPE : on traite directement chaque composante du signal donc la F.F.T. du signal est requise!!!

$\Rightarrow$  on calcule l'amplitude et la phase de chaque composante du signal de sortie à partir de celles du signal d'entrée et de la fonction de transfert du filtre ;

$$\underline{s}\left(k\frac{F_e}{N}\right) = H\left(j2\pi k\frac{F_e}{N}\right) \cdot \underline{e}\left(k\frac{F_e}{N}\right) \Rightarrow \begin{cases} |s(k\frac{F_e}{N})| = G(k\frac{F_e}{N}) \times |e(k\frac{F_e}{N})| \\ \varphi_s(k\frac{F_e}{N}) = \varphi_e(k\frac{F_e}{N}) + \arg[H(j2\pi k\frac{F_e}{N})] \end{cases}$$

avec  $N$  nombre total d'échantillons prélevés et  $k \in [0, \lfloor \frac{N}{2} \rfloor]$  (afin d'assurer  $f \in [0, F_{Ny} = \frac{F_e}{2}]$ ).

## III Complément TP : notions de fenêtrage

### III.1 Principe

L'analyse de Fourier numérique d'un signal échantillonné est généralement réalisée par implémentation d'un algorithme de transformée de fourier rapide ou FFT (Fast Fourier Transform) dans les logiciels de traitement (LatisPro), ou les oscilloscopes numériques (Tektronix). Cette analyse est fatalement réalisée sur une portion limitée du signal; cette troncature du signal entraîne l'introduction de composantes spectrales inattendues; on propose d'exposer schématiquement le principe de fenêtrage qui limite ces effets.

EXEMPLE : acquisition numérique d'un signal sinusoïdal, donc tronqué :  $\begin{cases} f(t) = A \cdot \cos(\omega t) & |t| < \frac{\tau}{2} \\ 0 & \text{ailleurs} \end{cases}$

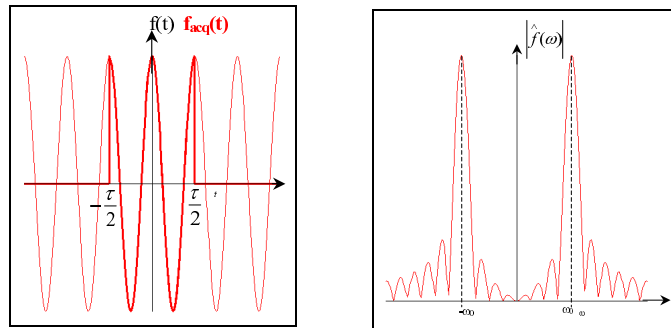


FIGURE II.17 – Sinusoïde tronquée et sa TF

La TF donne :  $\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f_{acq}(t) \cdot e^{-j\omega t} \cdot dt = \frac{A}{\sqrt{2\pi}} \int_{-\tau/2}^{\tau/2} \cos(\omega t) \cdot e^{-j\omega t} \cdot dt$

soit après calcul :

$$\hat{f}(\omega) = \frac{A\tau}{2\sqrt{2\pi}} \left[ \text{sinc}\left(\frac{\omega_0 - \omega}{2}\tau\right) + \text{sinc}\left(\frac{\omega_0 + \omega}{2}\tau\right) \right]$$

On constate sur le tracé du spectre de ce signal tronqué, que des lobes latéraux "entachent" le spectre. Deux solutions se présentent alors : soit on augmente  $\tau$  ce qui ralentit le traitement en temps réel (pas souhaitable),

soit on exploite la technique du fenêtrage.

Le fenêtrage consiste à multiplier le signal par une fonction du temps dite "fenêtre", puis réaliser la TF sur la nouvelle fonction ainsi créée. En fonction de la "forme" choisie pour la fenêtre, cette opération a pour effet de modifier l'allure spectrale du signal ainsi traité en améliorant certains défauts, notamment ceux engendrés par la troncature du signal.

On propose par exemple ici un fenêtrage de la sinusoïde tronquée par la fonction  $g(t) = e^{-\left(\frac{4t}{\tau}\right)^2}$

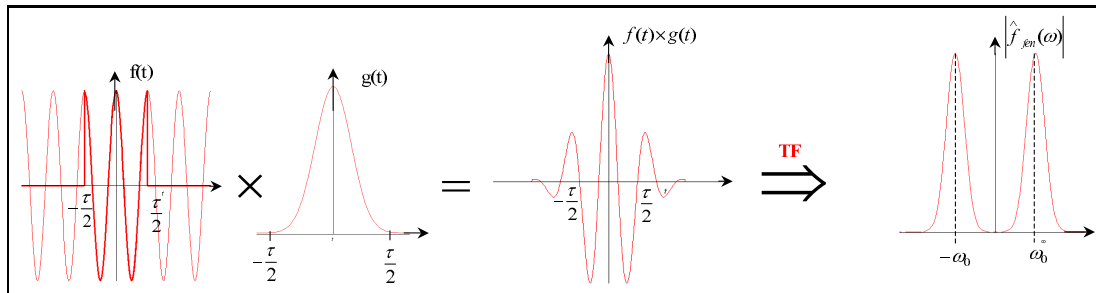


FIGURE II.18 – Fenêtrage et TF de la sinusoïde tronquée

**Bilan du fenêtrage :** on constate un effet d'affinement des raies spectrales -> meilleure lecture de la fréquence  $\omega_0$  et plus proche du spectre d'une sinusoïde infinie malgré la troncature.

**Remarque III-1: DU BON CHOIX DU FENÊTRAGE**

En pratique (cf TP), le type de fenêtrage choisi doit être adapté au signal à analyser, ainsi qu'à la mesure que l'on souhaite réaliser :

- Si l'on souhaite mesurer la fréquence d'une composante, la fenêtre la plus adaptée est la fenêtre de Hann (dite "Hanning").
- Si l'on souhaite mesurer l'amplitude d'une composante, on choisira la fenêtre type "Flatop" (sommet plat).

## III.2 Exemples d'exploitation

En «live» sur oscilloscope Tektronix en TP !