Arithmétique à virgule fixe et

à virgule flottante

Arithmétique à virgule fixe et à virgule flottante

I. Introduction :

Dans une structure de traitement numérique du signal, les coefficients (par exemple : coefficients d'un filtre RIF) et les grandeurs physiques à l'entrée ou à la sortie de la chaine intervenants dans l'élaboration du programme à exécuter par le processeur chargé du traitement sont des grandeurs **réelles**. Après numérisation ses grandeurs sont représentées sous l'une des deux formes binaires suivantes :

- ❖ Format Virgule fixe ;
  - Format courant : 16/24 bits
  - Idéal avec CAN/CNA 12/14 bits
  - Applications : Contrôle industriel, communications, instrumentation, parole, médical, militaire…
- ❖ Format Virgule flottante
  - Format courant : 32 bits
  - Idéal pour le traitement sur une grande dynamique
  - Applications : Audio professionnel, vidéo, médical…

II. Numérisation des signaux :

La conversion d'un signal analogique en signal numérique est caractérisée par deux discrétisations :

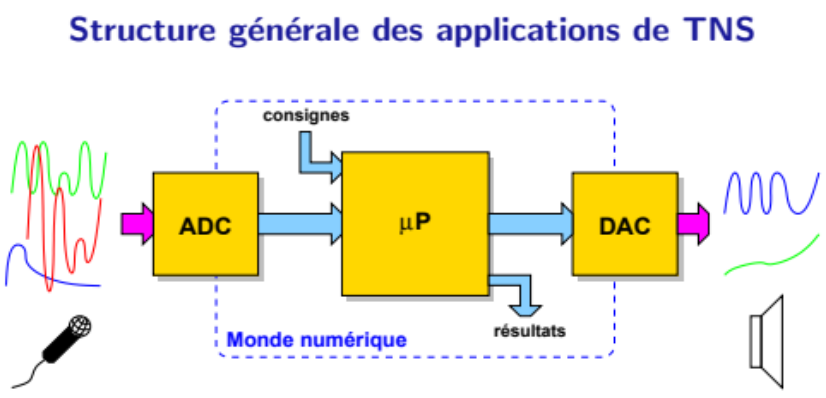
- La première concerne le temps et porte le nom d'**échantillonnage** :

Cela consiste à prendre des échantillons du signal analogique à des instants régulièrement espacés. Le signal fonction du temps s(t) est remplacé par ses valeurs s(nTe) a des instants multiples entiers d'une durée Te;

- La deuxième concerne l'amplitude et porte le nom de **quantification** :

Cela consiste à coder l'amplitude du signal sur un nombre fini d'éléments binaires.

Chaque valeur s(nTe) est approchée par un multiple entier d'une quantité élémentaire q; c'est l'opération de quantification. Cependant la numérisation d'un signal doit avoir un certain nombre de conditions sur **l'opération d'Echantillonnage et sur l'opération de quantification pour garantir la précision nécessaire** pour que le signal obtenu représente au mieux le signal d'origine sans perte d'information.



La structure générale des applications de TNS est donc constituée des éléments suivants :

- ADC ou CAN
- Microprocesseur
- DAC ou CNA

II-1 Echantillonnage

L'échantillonnage consiste donc à représenter un signal à temps continu s(t) par ses valeurs s(nTe) à des instant multiples de Te, Te étant la période d'échantillonnage.

- La précision de discrétisation est obtenue via la fréquence d'échantillonnage fe
- La fréquence fe doit être suffisamment élevée si l'on ne veut pas perdre trop d'informations sur le signal.
- Cependant plus fe est élevée (Te faible), plus le nombre d'échantillons à traiter sera important et plus le temps disponible pour effectuer les traitements numériques sera court
- La condition de Shannon permet d'échantillonner un signal sans aucune perte d'information si la fréquence d'échantillonnage fe = 1/ Te est au moins 2 fois supérieure à la plus grande fréquence intervenant dans le spectre (répartition de la puissance du signal en fonction des fréquences) du signal.
- Les échantillons prélevés devront être représentatifs du signal analogique c'est un sondage !
- Le signal analogique devra pouvoir être reconstitué (interpolé) à partir des échantillons (il existe une infinité de signaux qui passent par ces échantillons)
- L'échantillonnage à la période Te, introduit une périodicité du spectre du signal échantillonné, de période Fe.
- Si la condition de Shanoon n'est pas respectée on observe un "repliement de spectre"

Echantillonnage d'un signal sinusoïdal :

Domaine temporel

$s(t)=\sin(2\pi f_o t)$ $T_o=\frac{1}{f_o}$

Considérons 3 choix de période d'échantillonnage T_e

- **1er choix :**

$T_e=\frac{T_o}{8}$ $f_e=8f_o$

- **2ème choix** :

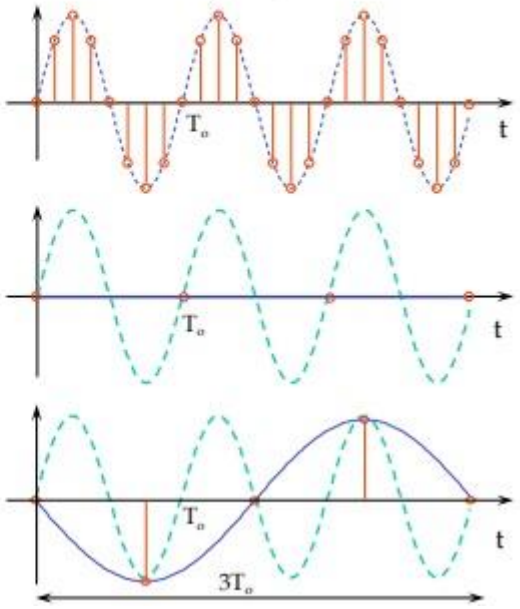
$T_e=T_o$ $f_e=f_o$

- **3ème choix** :

$T_e=\frac{3}{4}T_o$ $f_e=\frac{4}{3}f_o$

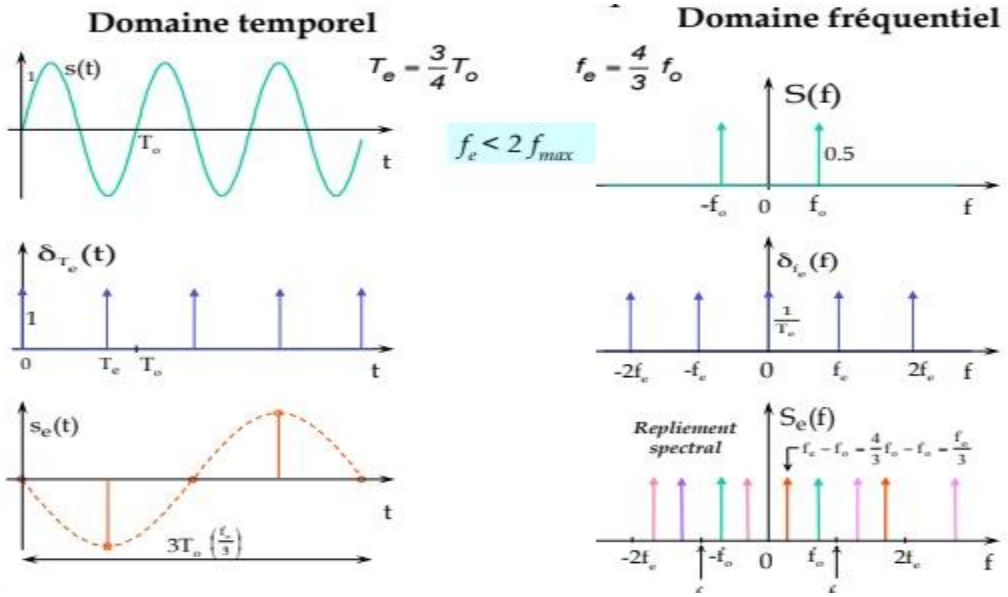
→ ***Repliement spectral***

$f=f_e-f_o=\frac{1}{3}f_o$



Echantillonnage d'un signal sinusoïdal :

Domaine fréquentiel :



II-1-a : Filtre anti-repliement :

En pratique, il est indispensable de faire précéder l'opération d'échantillonnage par un filtre passe-bas appelé filtre anti-repliement de fréquence de coupure un peu inférieure à la fréquence de Nyquist fe /2 pour éviter le repliement du spectre.

II-1-b : Traitements temps réel :

C'est le Traitement "en ligne" . Soit Te la période d'échantillonnage du signal s(t) à traiter. Ce traitement consiste en trois phases:

- Phase1 : Acquisition d'un échantillon Xn,
- Phase2 : traitement (à partir de Xn, et d'un certain nombre d'échantillons précédents pouvant provenir de différentes sources),
- Phase3 : sortie d'un échantillon Yn du signal de sortie Y.

On doit toujours conserver la même cadence d'échantillonnage pour le signal de sortie traité Y (Fech entrée = Fech sortie), l'échantillon Yn devra être fourni avant l'acquisition du nouvel échantillon Xn+1. pour faire du traitement en temps réel Il faudra donc que :

Tacquisition + Tcalcul + Tsortie < Te.

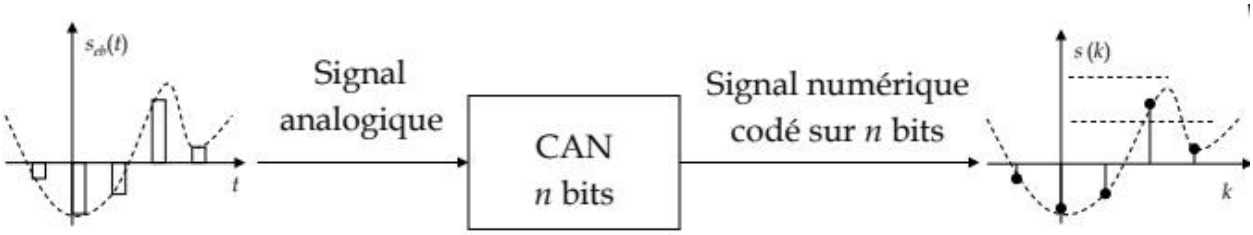
II-1-c- Exemples de Fréquences d'échantillonnage

Le tableau ci-dessous donne quelques valeurs de fréquences d'échantillonnage utilisées dans le domaine de l'audiovisuel :

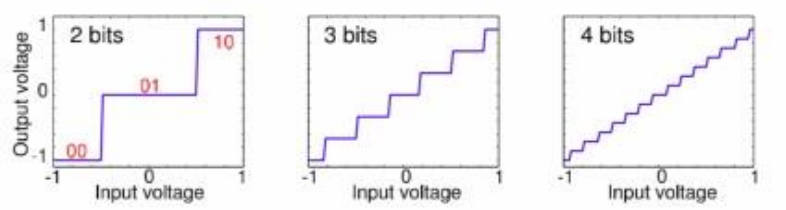
Quelques valeurs de fréquences d'échantillonnage			
Signaux	**Largeur de bande**	**Fréquence d'échantillonnage**	**Période d'échantillonnage**
Parole sur le réseau RTC	0 - 3400 Hz	8 KHZ	125 µs
Musique sur un CD Laser	0 - 22 KHz	44,1 kHz	22,6 µs
Audio	0 - 22 kHz	48 kHz	20,8 µs
Video	0 - ≈5 MHz	10 MHz	100 ns

II-2 **Quantification**

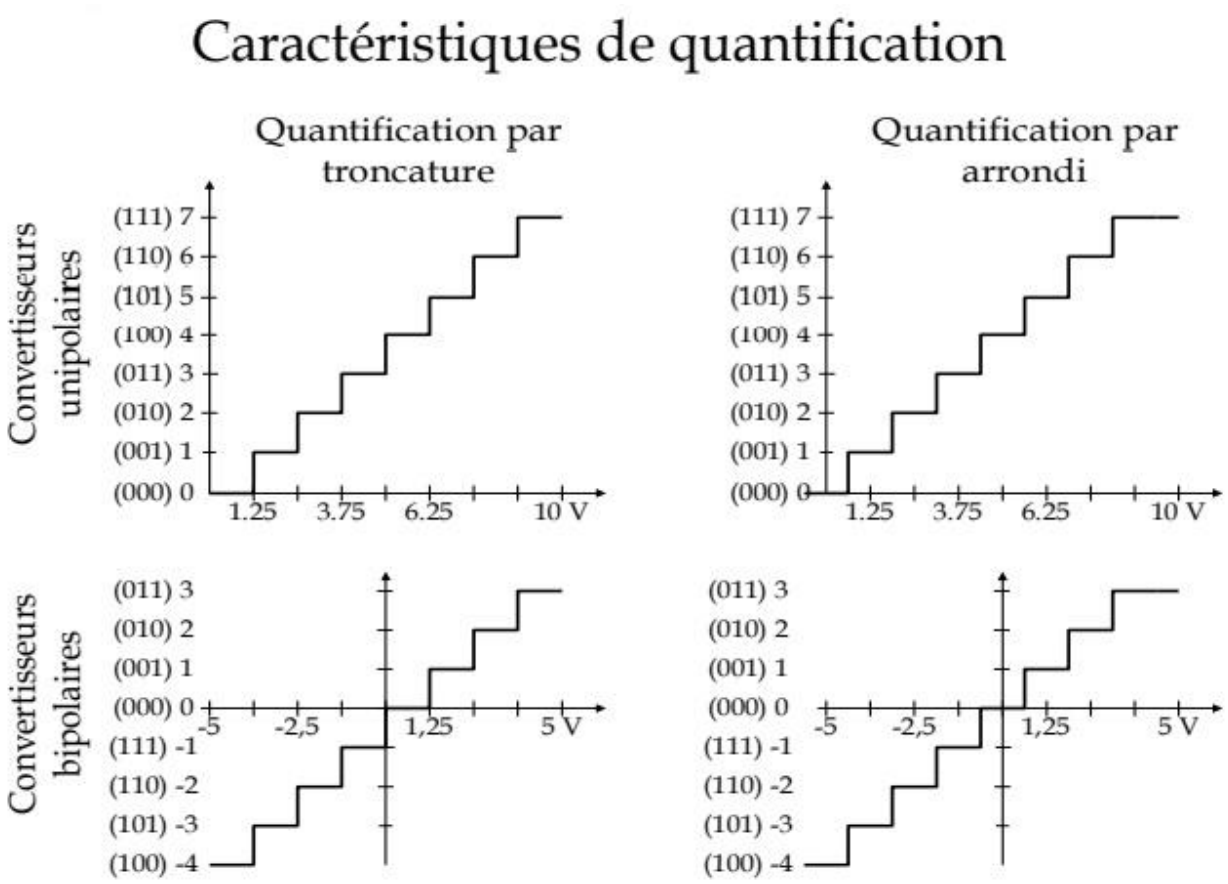
- La précision de numérisation d'un signal analogique est obtenue via le pas de quantification
- Ce problème peut assez facilement être traité en augmentant le nombre de bits du convertisseur analogique numérique CAN de la structure de traitement numérique du signal.



- Plage d'entrée analogique du CAN
  - Plage positive $0 \leq A < A_{FS}$
  - Plage centrée $-A_{FS}/2 \leq A < A_{FS}/2$
- Sortie d'un CAN *n* bits :
  - 2^n valeurs distinctes
  - dynamique en dB = $20 \log 2^n$
- Résolution du CAN
  - Plus petit incrément de signal continu discernable
  - La pleine échelle A_{FS} est divisée en 2^n intervalles



- Le convertisseur Analogique/numérique peut être unipolaire ou bipolaires et peut fonctionner en quantification par arrondie ou quantification par troncature. Voir les deux figures ci-dessous.



III. Formats de représentation des nombres

En général dans un calculateur les grandeurs numériques manipulés sont soient des entiers ou des nombres réels.

1. Représentation binaire des nombres entiers avec signe:

Il existe différentes représentations possibles pour les entiers signés :

- Représentation Signe- valeur absolue
- Représentation Complément à 1(complément restreint)
- Représentation Complément à 2 (complément vrai)

On considère la représentation en complément vrai (ou en complément à 2) des nombres entiers signés qui donne une représentation unique pour le zéro.

Soit N un nombre entier sur n bits :

$N=b_{n-1}.b_{n-2}.....b_0$;

On sait que $N+$**CA1**$(N)=2^n-1$; **CA1** : Complément à 1

$N+ (CA1(N) +1)=2^n$ pour le même nombre de bits on obtient $N+ (CA1(N) +1)=0$

On a donc : $N+$**CA2**$(N)=0$;

CA2(N)=**CA1**$(N) +1=$**CV**(N) : complément vrai de N

CV(N)= –N : Le nombre **–N** est représenté par le complément vrai de N

On montre que N est obtenu par :

$$N = -b_{n-1}.2^{(n-1)} + \sum_{i=0}^{n-2} b_i.2^i$$

Sur 8 bits par exemple on peut représenter les entiers relatifs N: allant de (-128) jusqu' a (+127)

- Dans cette représentation, le bit du poids fort nous indique le signe.
- On remarque que le zéro n'a pas une double représentation.

Si on travaille sur n bits, l'intervalle des valeurs qu'on peut représenter en CA2 :

Sur n bits on peut représenter les entiers signés N : $-(2^{(n-1)}) \leq N \leq +(2^{(n-1)} - 1)$

2. Représentation binaire des nombres réels :

Il existe deux méthodes pour représenter les nombre réel :

- Virgule fixe : la position de la virgule est fixe
- Virgule flottante : la position de la virgule change (dynamique)

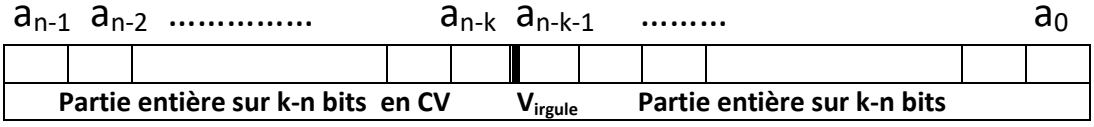
a. Représentation binaire des nombres réels en Virgule fixe :

Soit X un nombre réel càd réel X ∈ R. X se décompose en :

- une partie entière E(x) et
- une partie fractionnaire F(x) tel que :

X= E(X) + F(X), où E(x) ∈ **Z ;** E(X) est un nombre signé représenté en CV et
F(X) = x − E(X) ∈ **[0, 1[**.
Le nombre X est représenté en format binaire virgule fixe Q_k sur **n** bits comme suit :

$$X=a_{n-1}\ a_{n-2}\ .....a_{n-k}\ a_{n-k-1}.....a_0$$



$$X = -a_{n-1}.2^{(n-k-1)} + \sum_{i=-k}^{n-k-2} a_{i+k}.2^i$$

Précision :

En format Q_k sur n bits la précision est donnée par :
$q=2^{-k}$, la marge d'erreur est définie par **$q/2=2^{-(k+1)}$**
Exemple : En format Q_5 sur 8 bits le nombre 2,33 sera codé : 010.01010 avec une errer de 2^{-6}.
En format Q_{15} sur 16 bits on peut coder les nombres réels de -1 à 0.99997.

b. Représentation binaire des nombres réels en Virgule flottante

De nombreux défauts pour la représentation en virgule fixe

• Pour un nombre très grand comme le nombre d'Avogadro N_A (environ $6{,}022\times 10^{23}$), en écriture décimale cela nécessite au moins 24 chiffres pour une approximation à l'entier près.

• Pour un nombre très petit comme la charge élémentaire d'un électron (environ $-1{,}602 \times 10^{-19}$ Coulombs), en écriture décimale cela nécessite au moins 20 chiffres pour une approximation. Virgule flottante.

La représentation binaire des nombres réels en Virgule flottante est Inspirée de l'écriture scientifique

Exemple: $173{,}95 = +\ 1{,}7395 \times 10^2$

Généralisation:

Un nombre X réel est représenté en virgule flottante de base 2 par :

• Son signe S (codage sur un bit : 0 pour X positif et 1 pour X négatif)

• Son exposant e, un entier de k chiffres compris entre e_{min} et e_{max}

• Sa mantisse M de n + 1 bits

tel que :

$$\mathbf{X= (-1)^S . M . 2^{e-127}}$$

. Pour des questions de précision, on exige que la mantisse soit normalisée.

Sur 32 bits, On prend alors : **M=1,f ; M = 1 + f**

$$\mathbf{X= (-1)^S . (1,f). 2^{e-127}} \quad \textbf{avec :}$$

$$\mathbf{e= \sum_{i=0}^{7} e_i .2^i} \qquad \text{et} \qquad f=\sum_{i=0}^{22} f_i . 2^{(i-23)}$$

X est représenté en **binaire format des nombres réels Virgule flottante** par :

31	30							23	22														0
S	$\mathbf{e_7}$							$\mathbf{e_0}$	$\mathbf{f_{22}}$														$\mathbf{f_0}$
	Exposant								**Partie fractionnaire de la mantisse**														

Exemples :

Exemple 1. Quel est le nombre réel représenté par : 110000010 01010101010101010101010

1 10000010 01010101010101010101010

Signe exposant mantisse

• le bit de signe est 1 : le nombre est négatif.

• l'exposant biaisé est 10000010 correspondant à 130=e-127 :

L'exposant non biaisé est donc e=130−127 = 3

• la mantisse codée est 01010101010101010101010 correspondant donc à :

1,01010101010101010101010 c'est-à-dire : 1+2 −2 +2 −4 +2 −6 +2 −8 +2 −10 +2 −12 +2 −14 +2 −16 +2 −18 +2 −20 +2 −22 ≈ 1.3333333134651184 Le nombre codé est donc environ égal à −1.3333333134651184×2 3 c'est-à-dire environ égal à −10,66666

Exemples : 2.

Codons le nombre décimal −118,625

- Premièrement, nous avons besoin du signe, de l'exposant et de la partie fractionnaire. C'est un nombre négatif, le signe est donc "1".
- Puis nous écrivons le nombre (sans le signe) en binaire. Nous obtenons 1110110,101 (avec divisions par deux successives pour la partie décimale) Ensuite, nous décalons la virgule vers la gauche, de façon à ne laisser qu'un 1 sur sa gauche : 1110110,101 = 1,110110101×2 ^{6}

  C'est un nombre flottant normalisé : la mantisse est la partie à droite de la virgule, remplie de 0 vers la droite pour obtenir 23 bits. Cela donne 11011010100000000000000 (on omet le 1 avant la virgule, qui est implicite). L'exposant est égal à 6, et nous devons le décaler puis le convertir en binaire : 6+127 = 133 codé par 10000101.

  On a donc −118,625 qui est codé par 11000010111011010100000000000000

Le codage limite selon la norme IEE754 sur 32 bits.

Nombre représentable	signe	exposant	mantisse	valeur approchée
le plus grand	0	1111 1110	111 1111 1111 1111 1111 1111	$3{,}40282346\times10^{38}$
le positif non nul le plus proche de 0	0	0000 0001	000 0000 0000 0000 0000 0000	$1{,}17549435\times10^{-38}$
le négatif non nul le plus proche de 0	1	0000 0001	000 0000 0000 0000 0000 0000	$-1{,}17549435\times10^{-38}$
le plus petit	1	1111 1110	111 1111 1111 1111 1111 1111	$-3{,}40282346\times10^{38}$

La représentation de 0 :

Le zéro « 0 » "ne peut pas avoir une mantisse commençant par un 1, il n'est donc pas possible de le représenter comme décrit ci-dessus : par convention, il a été décidé qu'un nombre vaut 0 si, et seulement si, tous les chiffres de son exposant et sa mantisse valent 0 (il y a donc +0 et −0)

c-Comparatif représentation Virgule fixe/ Virgule flottante

	Virgule fixe (95% des DSP)	Virgule flottante
Complexité hard	Simple	Complexe
Coût	Faible	Elevé
Consommation	Faible	Elevée
Dynamique max.	190 dB	1500 dB
Complexité progr.	Elevée	Faible