Université IBN TOFAIL – Kénitra
ECOLE NATIONALE DES SCIENCES APPLIQUEES

Cours de Traitement de Signal et Communication Numérique

CH4: Numérisation d'un signal analogique

Par:

Hassan EL FADIL

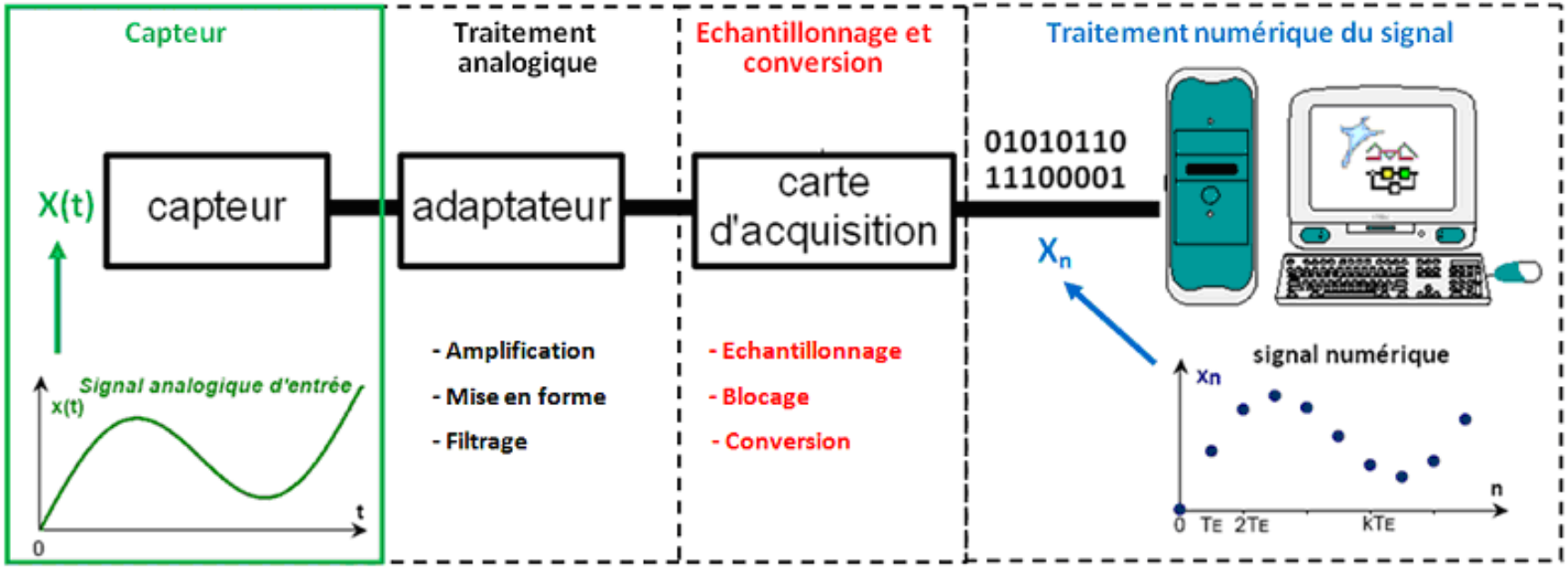
elfadil.h@gmail.com

1. Introduction

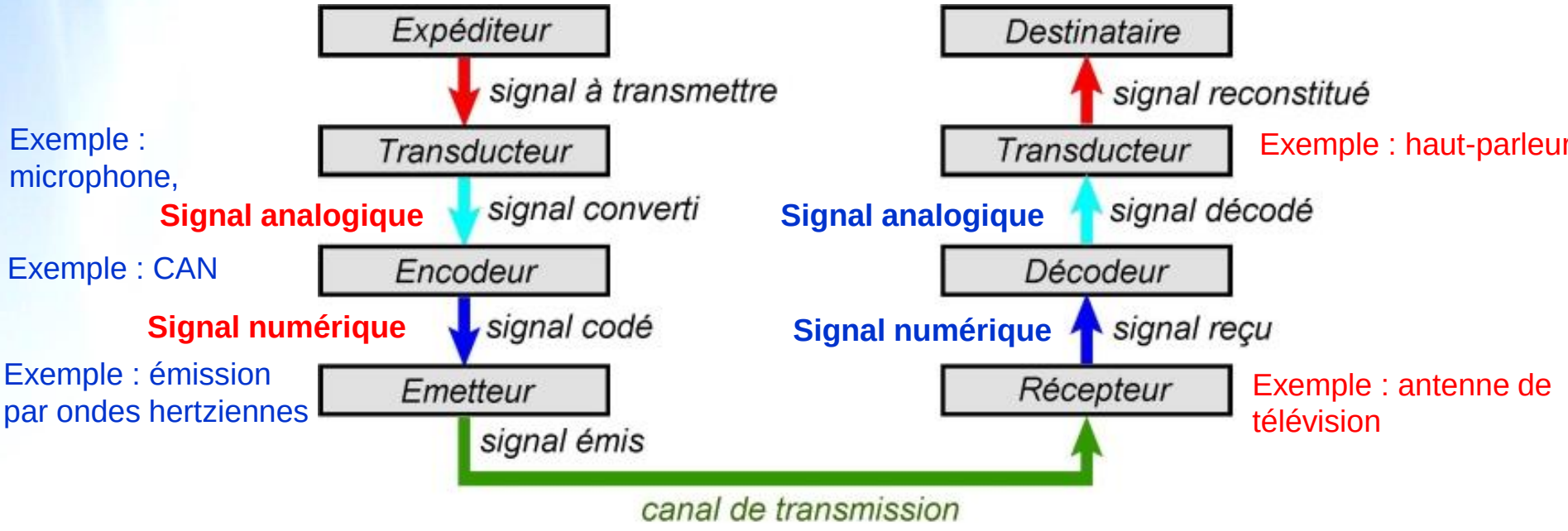
- Pourquoi numériser un signal?
  - *le stocker, dupliquer, transmettre sans perte, ou au contraire pour le compresser.*
  - *lui faire subir des traitements numériques (filtrage,*
  - *analyse, prédiction,…)*
  - *mélanger différents signaux de différentes natures sur de mêmes supports (multimédia: son, image,…)*
- Avantages de la numérisation
  - *précision et stabilité dans le temps et dans la production*
  - *souplesse, polyvalence, modularité et évolutivité*

Exemple d'applications:

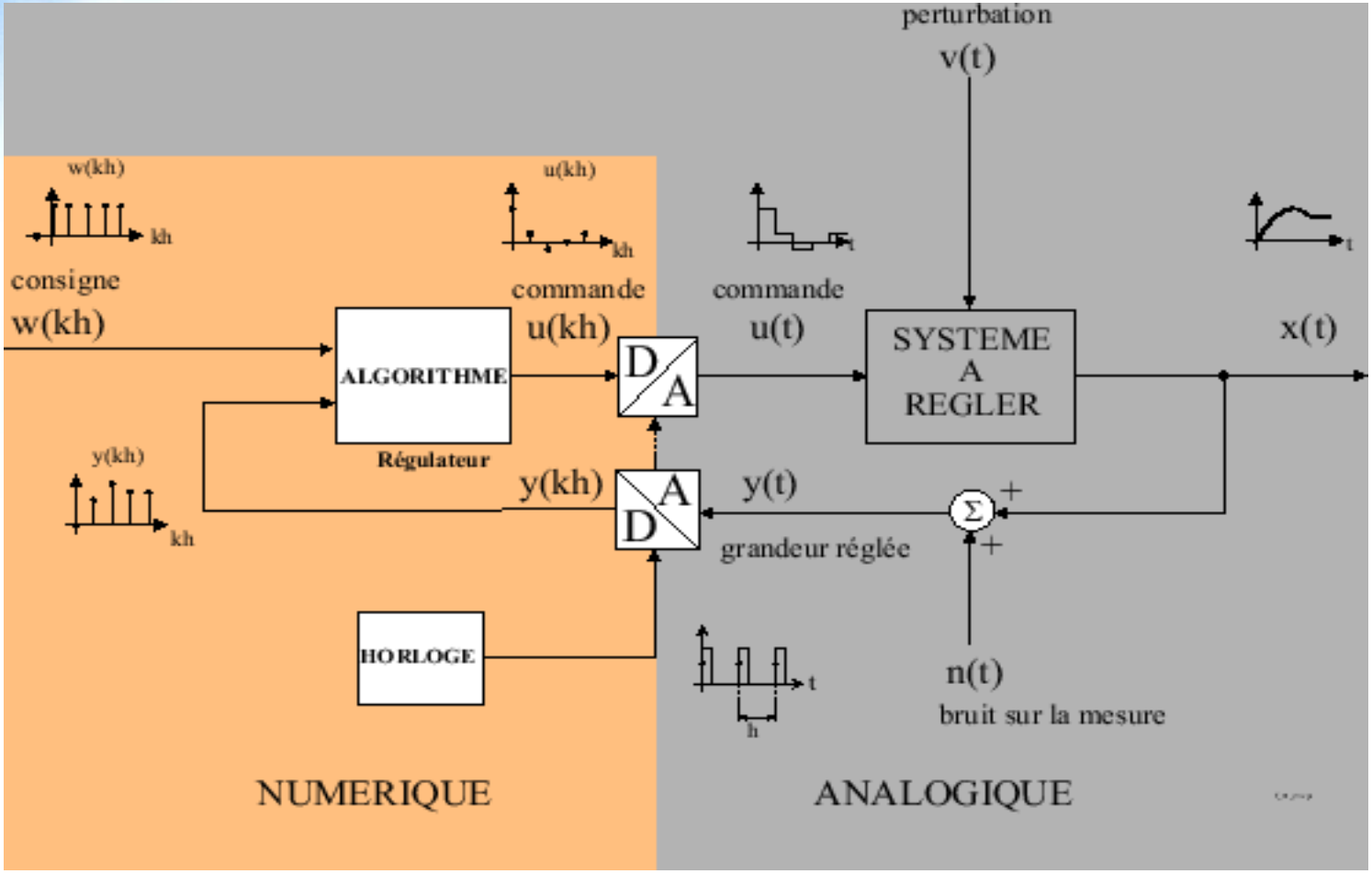
Exemple 1: Acquisition numérique du signal:



Exemple2: Transmission numérique de l'information:



Exemple 3: Régulation numérique:



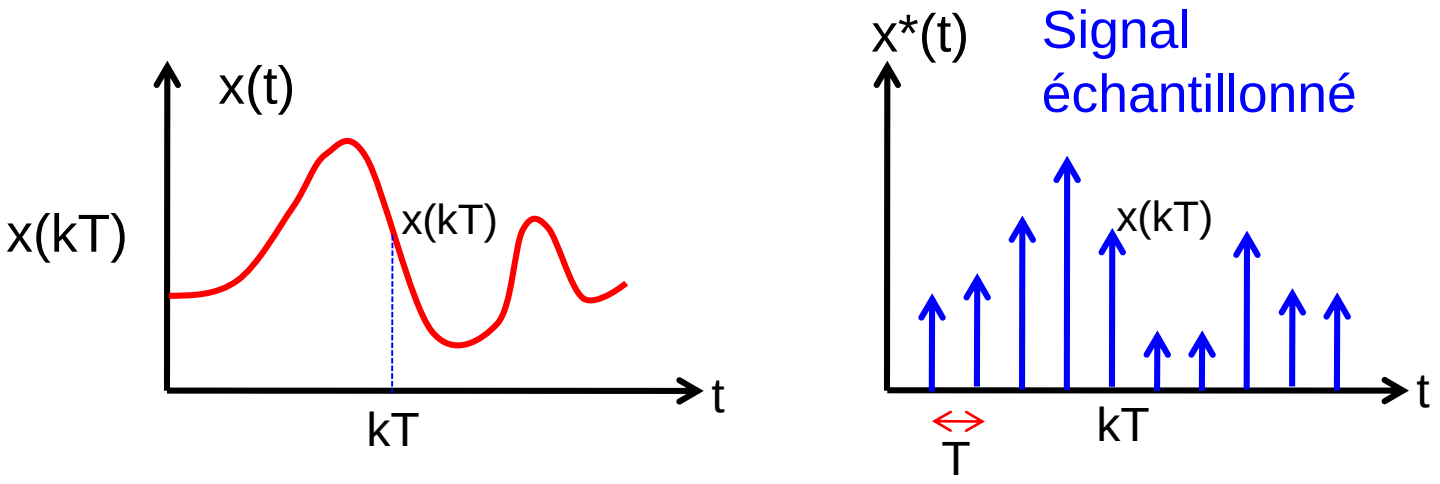
2. Signaux échantillonnés

- Échantillonner un signal analogique signifie le remplacer par une suite de valeurs prises à des instants bien définis.

- L'échantillonnage joue un rôle capital en traitement de signal du fait qu'un ordinateur traite des nombres plutôt que des grandeurs analogiques.

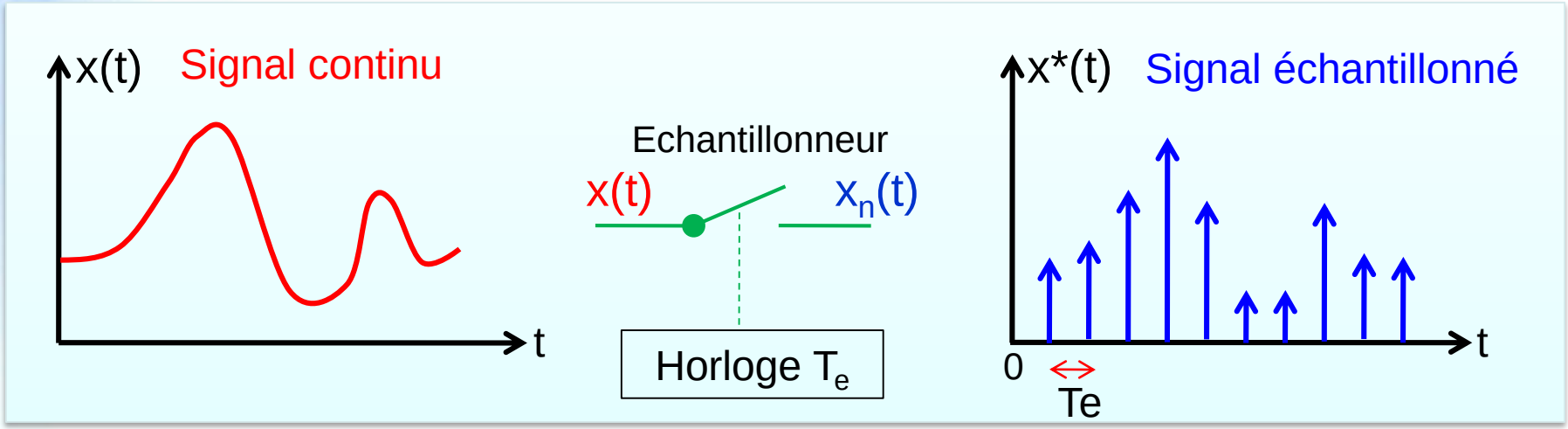
…Signal échantillonné

- ***Le signal échantillonné x*(t)*** , associé à un signal continu x(t), est composé d'une série d'impulsions de Dirac.



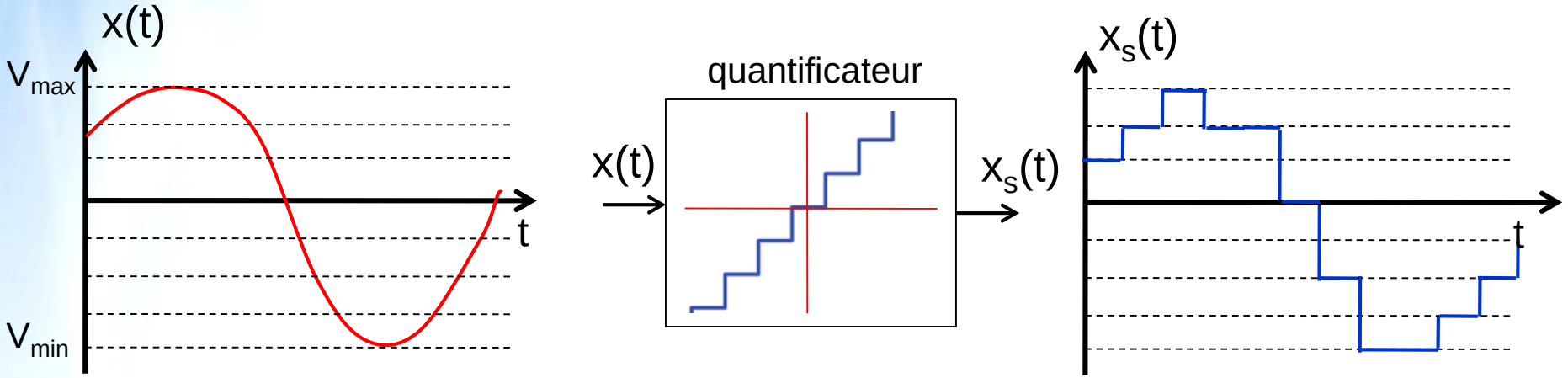
3. Les éléments de la chaine de numérisation

3.1. Echantillonnage idéal



3.2. Quantification:

- Quantification: transformation du signal échantillonné en différents niveaux qui sont codés en binaire (C'est le rôle du CAN: Convertisseur Analogique Numérique)



Ordres de grandeurs:

Type de support de sons	Quantification choisie
CD audio	16 bits
DVD	24 bits
Téléphonie	8 bits
Radio numérique	8 bits

…Quantification

- **La valeur Pleine échelle** (PE) : Full-Scale voltage (also called 'span'):

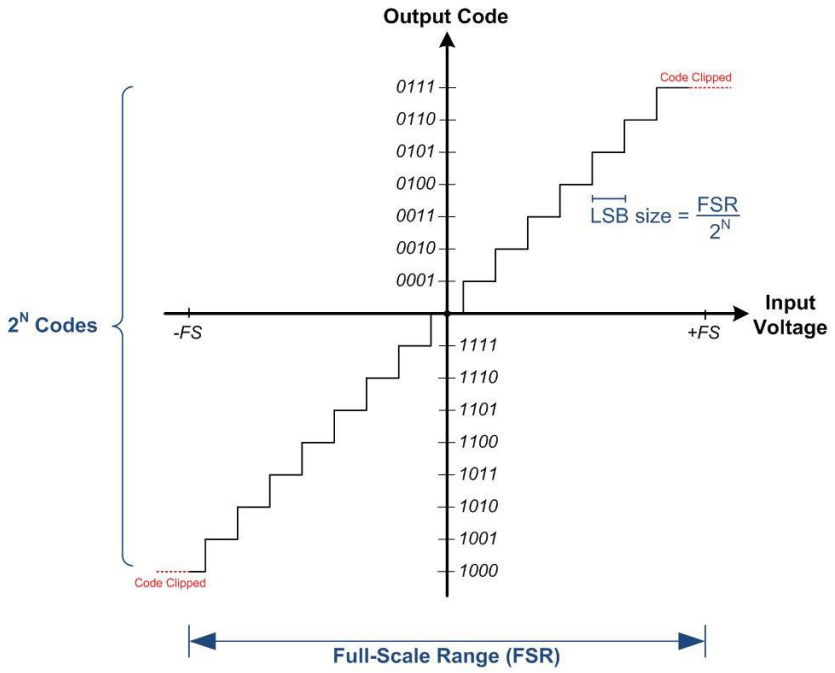
$$V_{PE} = V_{max} - V_{min}$$

- **La résolution**: c'est le nombre de bits N du CAN.
- **Le quantum** (LSB voltage), noté **q**, correspond à la quantité élémentaire de variation du signal analogique pour une variation d'un bit:

$$q = LSB = \frac{V_{PE}}{2^N}$$

- L'erreur maximale de quantification (Quantization Error) est:

$$\varepsilon = \pm\frac{1}{2}q = \pm\frac{1}{2}LSB$$



…Quantification

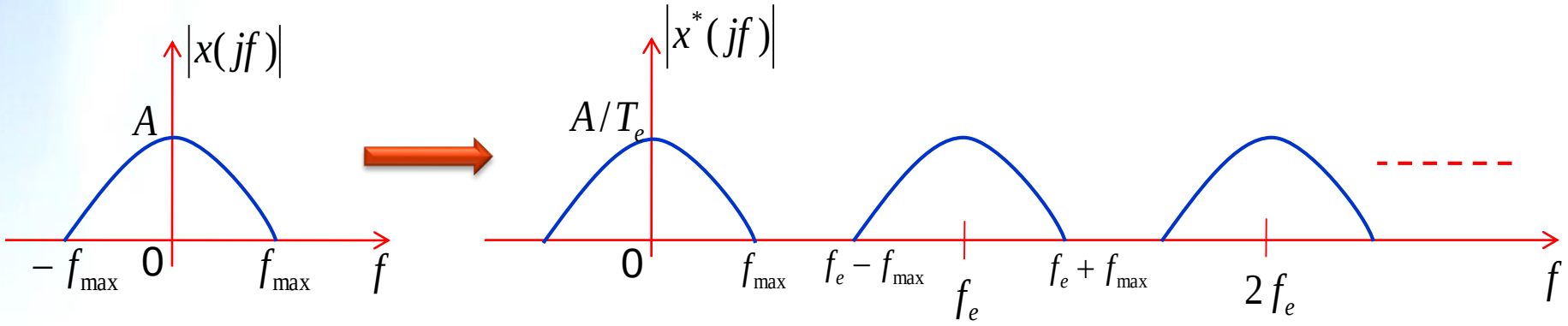
- La plupart des convertisseurs CAN actuels ont un nombre de bits supérieurs à 8: 10, 12 ou 16; exceptionnellement plus.
- Le tableau ci-dessous donne l'erreur absolue maximum commise pour une plage d'entrée du CAN s'étendant de 0 à 10V.

Nombre de bits n	8	10	12	16	20	24
Nombre de niveaux	256	1024	4096	65536	$1,05.10^{6}$	$16,8.10^{6}$
ε absolue maxi (mV)	40	10	2,5	0,15	0,01	0,0006
Résolution %	0,4	0,1	0,025	$1,5.10^{-3}$	$1,1.10^{-4}$	6.10^{-6}

- Pour des raisons de précision, il est en général nécessaire de choisir une quantification très fine , c'est-à-dire inférieure à 1%.

4. Reconstitution d'un signal échantillonné

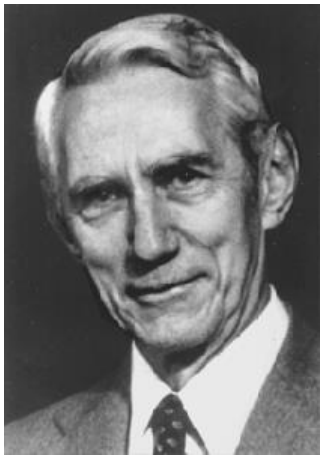
Spectres:



Le spectre du signal échantillonné est bien plus étendu que celui du signal continu qui lui a donné naissance.

4.1. Théorème de Shannon:

Claude Elwood Shannon (1916 - 2001) est un ingénieur en génie électrique et mathématicien américain. Il est l'un des fondateurs, de la théorie de l'information.



Claude Shannon

Théorème de Shannon: Pour pouvoir reconstituer un signal continu à partir d'un train d'échantillons de période T_e, on choisit F_e telle que:

$$F_e \geq 2F_{max}$$

$F_e = 1/T_e$;

4.2. Quelques fréquences d'échantillonnage:

Signal	Fréquence d'échantillonnage
Parole	8 kHz (téléphonie) 16 kHz (conférence audio)
Audio	32 kHz 44,1 kHz (Qualité CD) 48 kHz (DVD)
Radio Numérique	22,5 kHz
Vidéo	Environ 10 MHz (définition PAL)

5. Transformée en Z

- On définit ainsi le *transformée en z* du signal *x(t)*:

$$X(z) = \sum_{n=0}^{+\infty} x_n z^{-n}$$

$$x_n = x(nT_e)$$

VOIR la table de transformées en z de signaux et de fonctions habituellement utilisées.

6. Equation de récurrence

- Cette méthode consiste à déduire la valeur de l'échantillon $x(nT_e)$ de la connaissance des échantillons précédents aux instants: $(n-1)T_e$, $(n-2)T_e$, …
- S'apprête mieux à la programmation sur calculateurs.

Exemple: Soit un système à temps discret **S** de fonction de transfert en z:

$$\frac{Y(z)}{U(z)} = \frac{1}{z-0.5}$$

- On peut écrire:

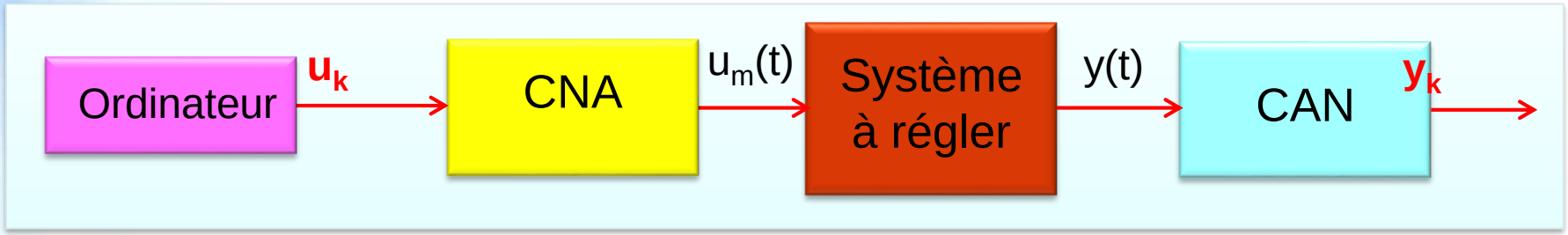
$$Y(z) - 0.5z^{-1}Y(z) = z^{-1}U(z)$$

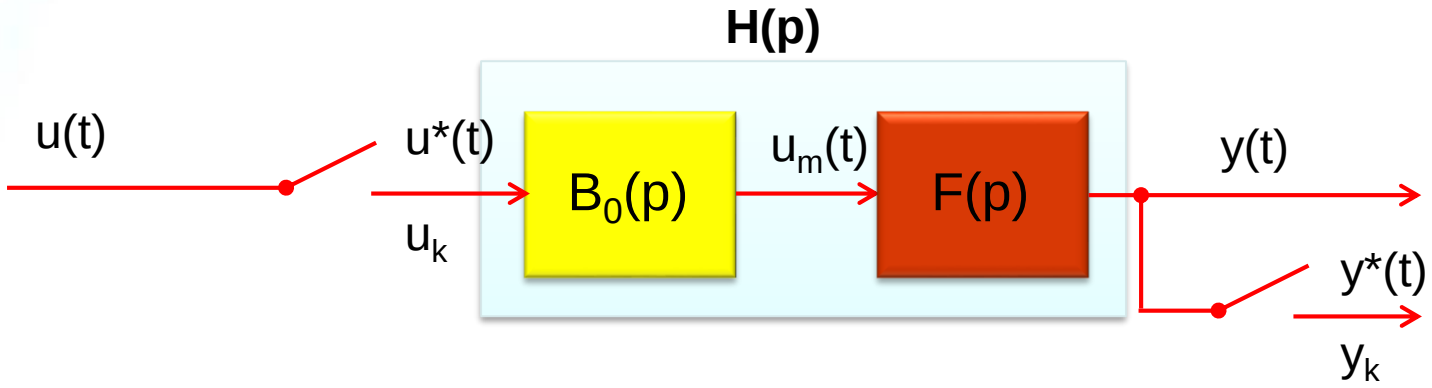
- soit:

$$y(kT_e) = 0.5y\left[(k-1)T_e\right] + u\left[(k-1)T_e\right] \Rightarrow y_k = 0.5y_{k-1} + u_{k-1}$$

7. Fonction de transfert discrète

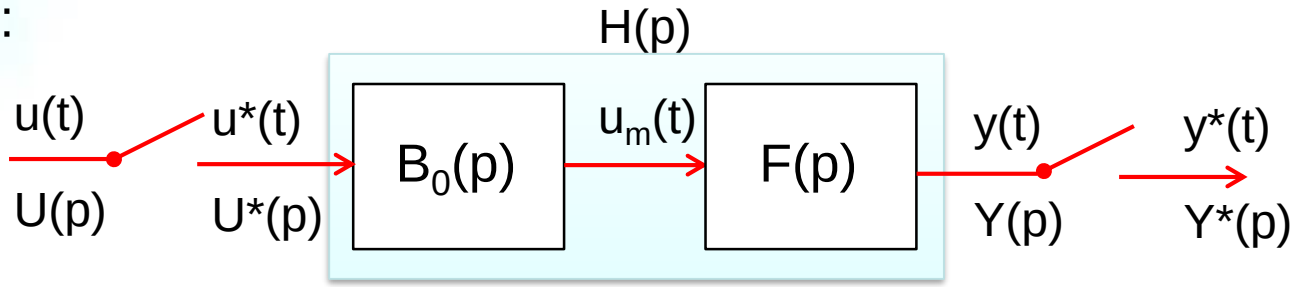
- ***fonction de transfert échantillonnée*** lie le signal discret d'entrée u_k au signal de sortie échantillonné y_k.





- F(p) :FT du système (analogique) à commander par ordinateur.
- H(p) : FT de tous les éléments compris entre deux échantillonneurs.

- Soit le système linéaire à temps discret (ou système échantillonné) suivant:



- On montre que:

$$Y(z) = H(z) \cdot U(z) \quad \Rightarrow \quad H(z) = \frac{Y(z)}{U(z)}$$

- Nous avons:

$$H(z) = (1 - z^{-1}) Z\left[\frac{F(p)}{p}\right]$$

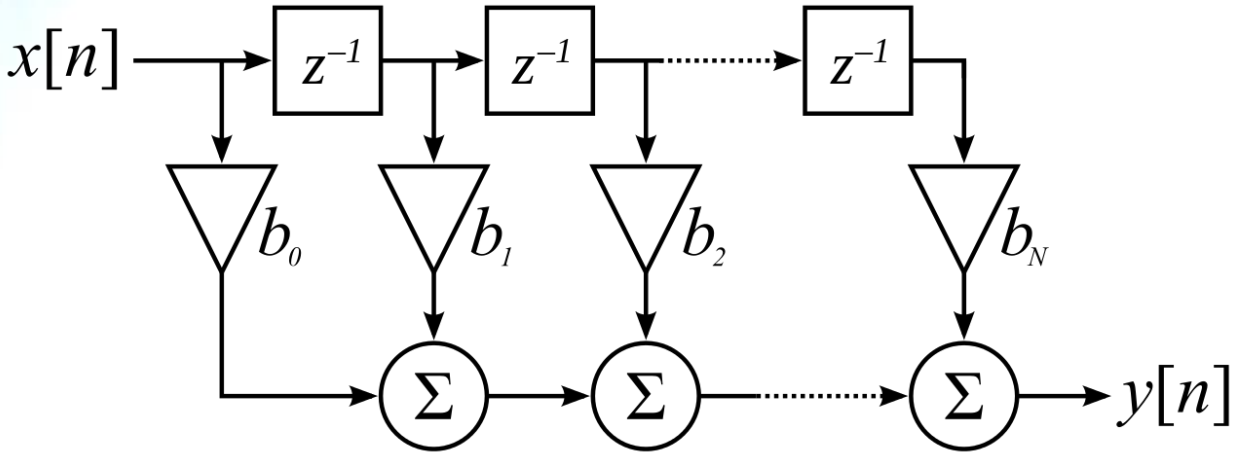
8. Filtres numérique

- Il y a deux grandes familles de filtres numériques : RIF et RII.

8.1. Les filtres RIF:

- Les filtres RIF (filtres à réponse impulsionnelle finie), en anglais FIR (*finite impulse response*).
- Ce type de filtre est dit fini, car sa réponse impulsionnelle se stabilisera à zéro.
- Un filtre FIR est non récursif, c'est-à-dire que la sortie dépend uniquement de l'entrée du signal, il n'y a pas de contre-réaction.

Réalisation d'un filtre RIF



$$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_N x(n-N)$$

$$y(n) = \sum_{k=0}^{N} b_k x(n-k)$$

Fonction de transfert

$$H(z) = \sum_{k=0}^{N} b_k z^{-k}$$

8.2. Les filtres RII:

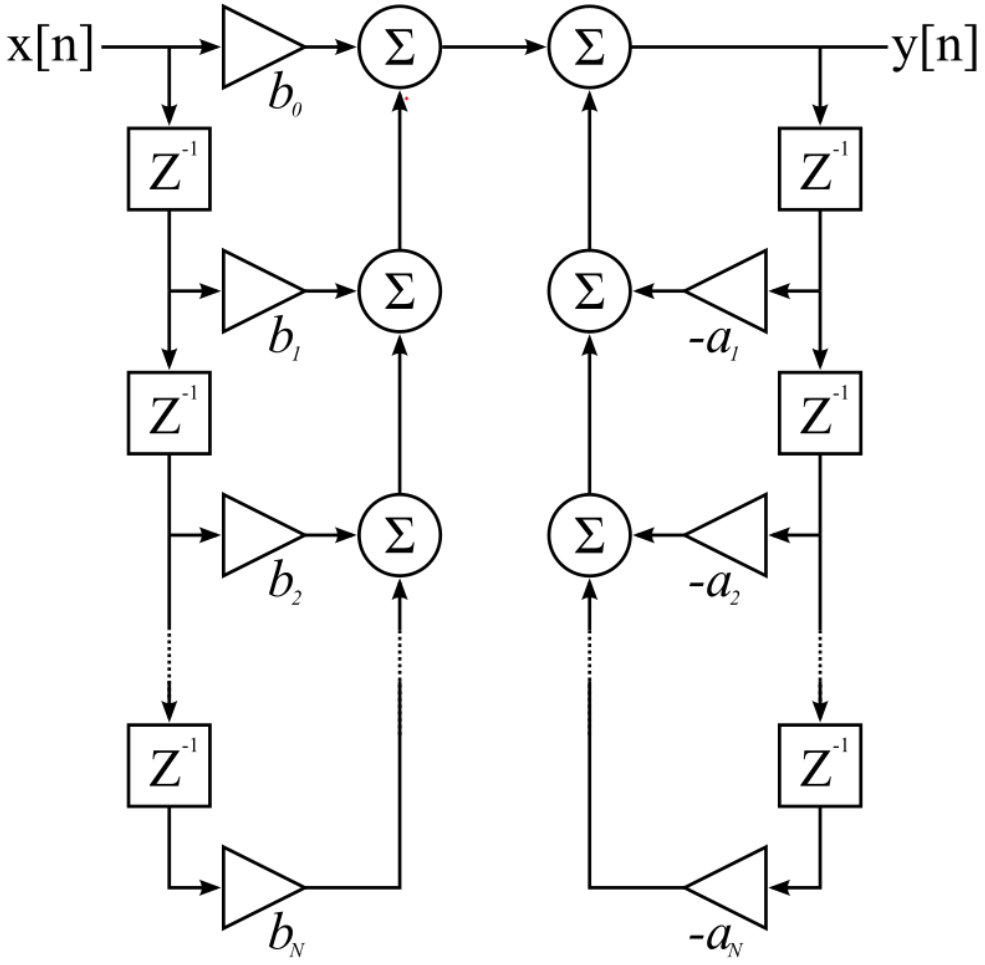
- Les filtres RII (Filtre à réponse impulsionnelle infinie), en anglais IIR (*infinite impulse response*), possèdent une réponse impulsionnelle qui ne s'annule jamais définitivement ou qui converge éventuellement vers zéro à l'infini.
- Ce type de filtre est récursif, c'est-à-dire que la sortie du filtre dépend à la fois du signal d'entrée et du signal de sortie, il possède ainsi une boucle de contre-réaction *(feedback)*.
- Les filtres IIR sont principalement la version numérique des filtres analogiques traditionnels

Réalisation d'un filtre RII

$$y(n) = \sum_{k=0}^{N} b_k x(n-k) - \sum_{k=1}^{M} a_k y(n-k)$$

Fonction de transfert discrète:

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k z^{-k}}$$



8.3. Comparaison RIF-RII

Filtre RIF	Filtre RII
• sont toujours stables. • La complexité d'un filtre RIF est moindre que celle d'un filtre RII du même ordre. • Généralement, les filtres RIF sont moins sensibles aux erreurs de quantification que les filtres RII. • Un filtre RIF est moins sélectif qu'un filtre RII du même ordre. C'est-à-dire que la transition entre la bande passante et la bande rejetée est moins rapide que dans le cas du filtre RII.	• Les filtres RII ne sont pas forcément stables, la stabilité dépend de la position des pôles dans le plan complexe ; • Beaucoup moins de calculs par rapport à un filtre RIF équivalent au niveau des performances ; • Généralement, les filtres RII sont plus sensibles aux erreurs de quantification que les filtres RIF. La récursivité peut générer des erreurs cumulatives ; • Un filtre RII est plus sélectif qu'un filtre RIF du même ordre.

9. Applications sous Matlab/Simulink

9.1. Signaux échantillonnés:

Affichage signal discret et bloqué

```
%%%%%%%%%%%%%%%%%%%% Affichage d'un signal discret  %%%%%%%%%%%%%%%%%%

f1=60;                          %%% Fréquence du signal
N1=256;                         %%% Nombre d'échantillons
Te1=1/1024;                     %%% Période d'échantillonnage
t1=0:Te1:(N1-1)*Te1;            %%% Axe du temps
x1=cos(2*pi*f1*t1);             %%% Signal cosinusoidal
figure(1)
subplot(211) ;
stem(t1,x1,'filled') ;          %%% Echantillons
axis([0 0.03 -1.2 1.2]);
title('Echantillons') ;
grid
subplot(212) ;
stairs(t1,x1) ;                 %%% Signal bloqué
title('Blocage d''ordre 0') ;
axis([0 0.03 -1.2 1.2]);
grid
xlabel('temps')
```

Applications Matlab:
…signal échantillonné

FFT d'un signal échantillonné

```
%%%%%%%%%%%%%%%%%%% spectre signa échantillonné  %%%%%%%%%%%%%%

f=100;                                  %%% fréquence du signal
Fs=1000;                                %%% fréquence d'échantillonnage
Ts=1/Fs;
N=10000;                                %%% Longueur de la séquence
t=Ts*(0:N-1);
x=5*sin(2*pi*f*t);                      %%% signal sinusoidal d'amplitude 5V
y=fft(x)/N;

f=Fs*linspace(0,1,length(y));
figure(2)
plot(f,abs(y));
title('spectre de X(f)');
xlabel('Fréquence (Hz)');
```

9.2. Fonction de transfert discrète

Soit un système décrit par sa FT continue:

$$G(p) = \frac{2p+1}{p^2+2p+1}$$

1) A l'aide de Matlab, discrétiser cette fonction de transfert avec une période d'échantillonnage **Te=0.1s** en utilisant l'instruction **c2d (Help c2d)**. Soit G_d la fonction de transfert obtenue.
2) Visualiser sur la même figure les réponses indicielles et impulsionnelles de G et Gd .
3) Répéter cette opération pour les périodes d'échantillonnage 0.5, 1 et 2. Conclure.
4) Tester les 2 méthodes de discrétisation suivantes : **'zoh'**, **'tustin'**. Conclure.

9.3. Filtres numérique:

Analyse temporelle et fréquentielle d'un filtre numérique

On considère une filtre numérique de fonction de transfert:

$$H(z)=\frac{Y(z)}{X(z)}=\frac{z^2+2z}{z^2+0.2}=\frac{1+2z^{-1}}{1+0.2z^{-2}}$$

Il s'agit bien d'un filtre RII

Applications Matlab:
…Filtres numériques

```
%%%%%%%%%%%%%%%%%%%%%%%%%% Filtre numérique %%%%%%%%%%%%%%%%%%%%%%%%%%
Num = [1 2];                                %%%% Coefficients de numérateur
Den = [1 0 0.2];                            %%%%%% coefficients de dénominateur
[b a] = eqtflength(Num,Den);                %%% force le même ordre pour le Num et le Den
[z,p,k] = tf2zpk(b,a);                      %%% affiche les zéros, pôles et gain
mod_p=abs(p);                  %%% calcule le module de p (pour vérifier  la stabilité)
figure(1)
zplane(b,a) ;                               %%% affiche dans le plan (Im,Re) les pôles et les zéros

%%%%%%%%%%%%%%%%%% Réponse impulsionnelle %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2)
impz(b,a,10,1e3);              %%% Affiche 10 échantillons de la réponse impulsionnelle du filtre
                               %%% les échantillons sont espacés d'une période de 1/1e3

%%%%%%%%%%%%%%%%%% Réponse fréquentielle du filtre  %%%%%%%%%%%%%%%%%%%%%%
Ts=1;
Fs=1/Ts;
N=512;
w=0:1/N:1-1/N;
[H,f]=freqz(b,a,N,Fs);
% la réponse en fréquence est calculée à l'aide de N=512 échantillons et d'une fréquence
%d'échantillonnage spécifiée par Fs en Hz.  % Le vecteur de fréquence f est calculé en Hz et a des
% valeurs allant de 0 à fs/2 Hz.
figure(3);
subplot(211), plot(f,abs(H))
subplot(212), plot(f,(180/pi)*angle(H))
```

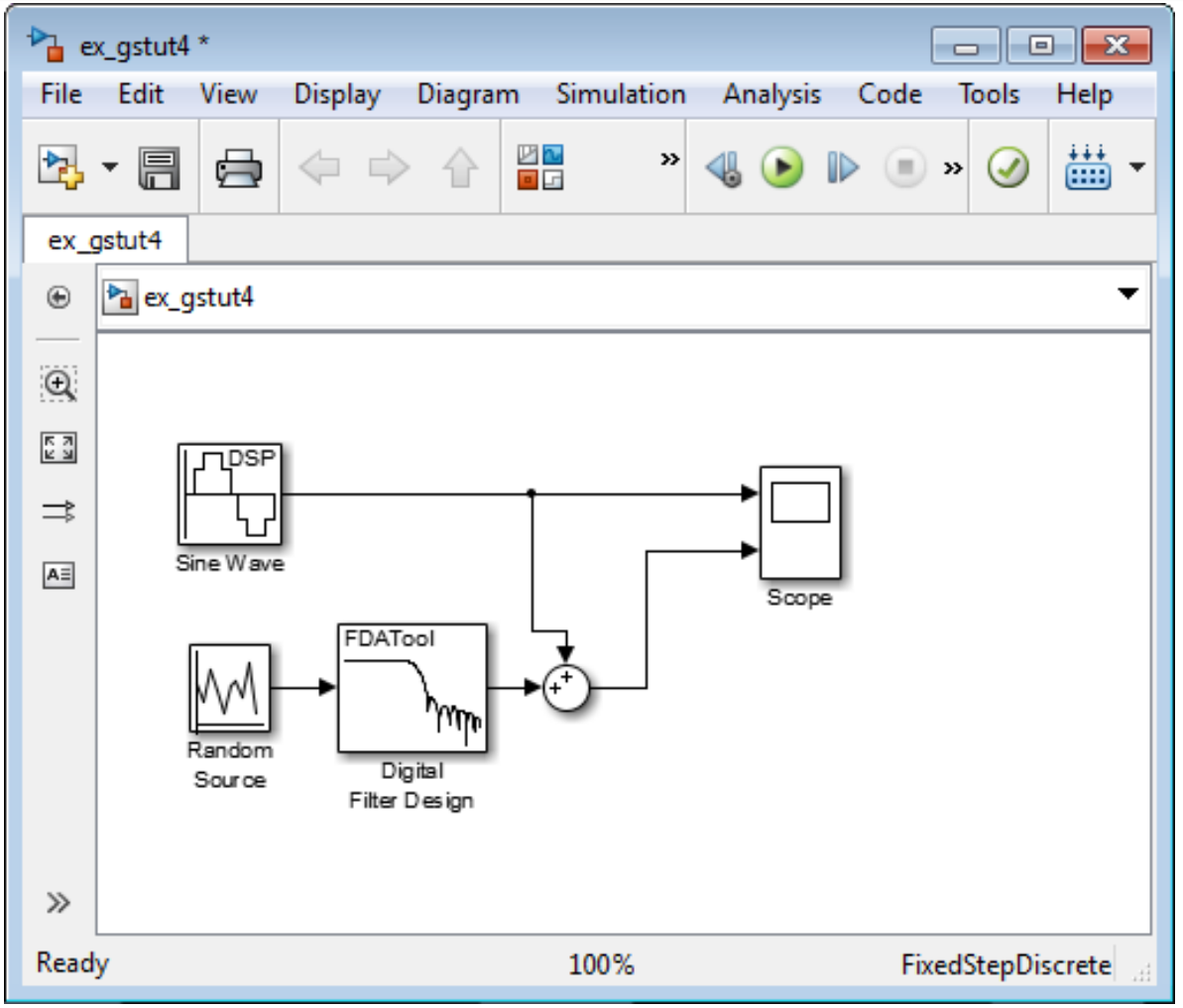
Applications Matlab:
…signal échantillonné

Filtrage numérique d'un signal

```
%%%%%%%%%%%%%%%% Filtrage numérique d'un signal bruité %%%%%%%%%%
Fs = 1e4;                                   %% fréquence d'échantillonnage
f0=262;                                     %% fréquence du signal sinusoïdal 262Hz
tstop=5/f0; t = 0:1/Fs:tstop;               %% axe temps
s = sin(2*pi*f0*t);                         %% signal sinusoïdal sans bruit
n = 0.1*randn(size(s));                     %% bruit
sn = s + n;                                 %% signal bruité
figure(1)
plot(t,sn),
%%%%%%%%%%%%%%%% coefficients du filtre  %%%%%%%%%%%%%%%%
b=[.25 .25 .25 .25];
a=[1 0 0 0];
%%%%%%%%%%%%%%%% Filtrage du signal  %%%%%%%%%%%%%%%%
snf1=filter(b,a,sn);
figure(2)
plot(t,sn,t,snf1),
```

Exercice 1:

En utilisant le toolboxe ***"DSP System Toolbox"****, réaliser un filtrage numérique sous* SIMULINK (voir ci-contre).



Voir le lien:

https://www.mathworks.com/help/dsp/gs/design-and-implement-a-filter.html

Exercice 2:

Considérons un filtre moyenneur sur 2 points décrit par l'équation de récurrence suivante:

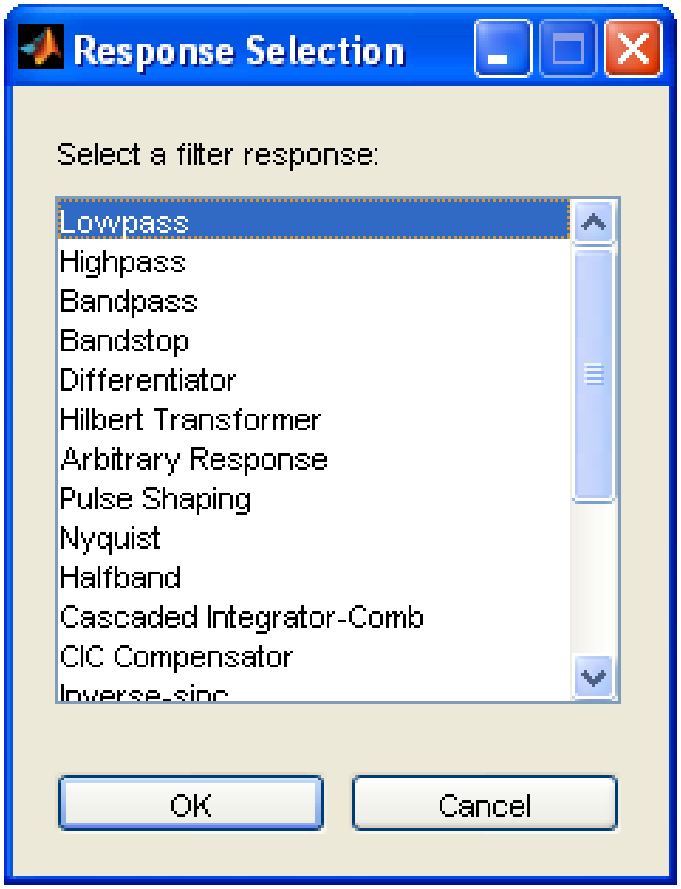
$$s(k) = \frac{1}{2}e(k) + \frac{1}{2}e(k-1)$$

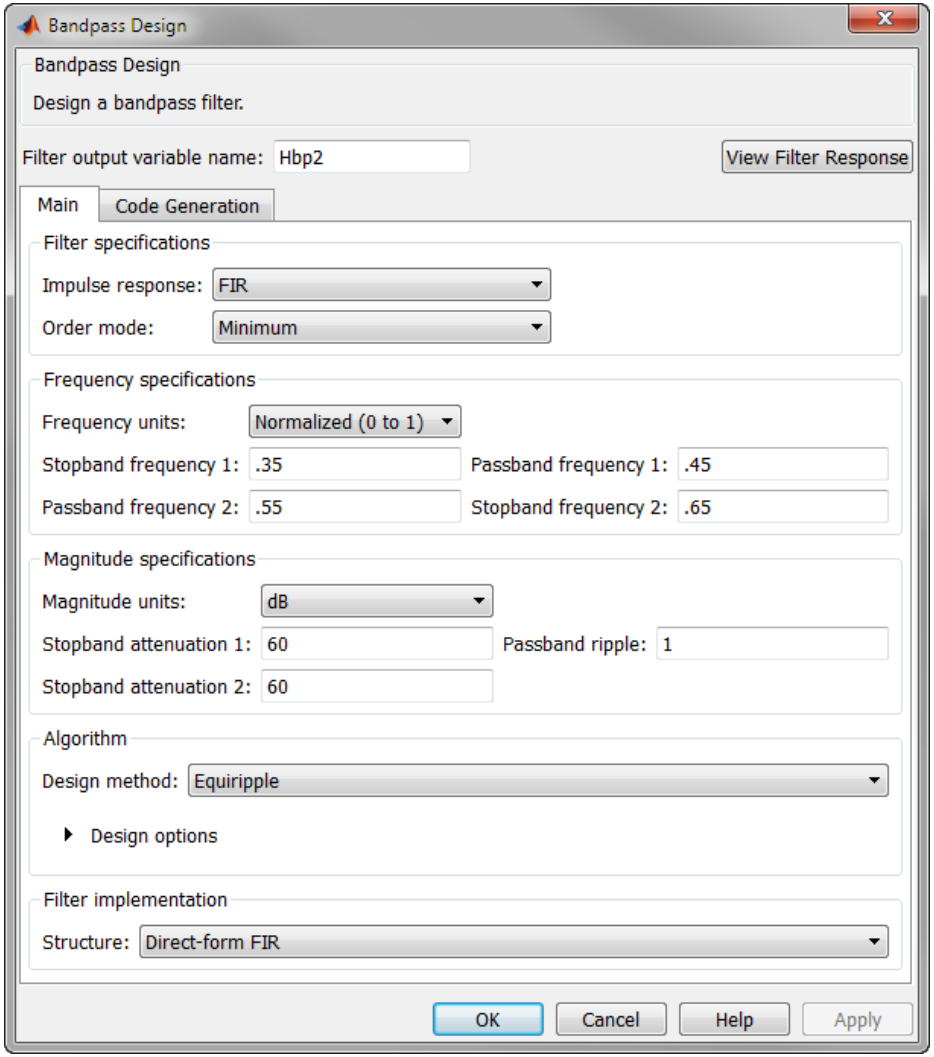
1) Donner sa fonction de transfert H(z)?
2) Tracer sa réponse impulsionnelle?
3) Spécifier son type: RIF ou RII?
4) Tracer ses pôles et ses zéros dans le paln Z
5) Le filtre est-il stable?
6) Tracer sa réponse fréquentielle (module et phase) dans l'intervalle 0<f<Fs/2. Avec Fs: fréquence d'échantillonnage.
7) Déduire des réponses en fréquence le type de filtrage (passe-bas, passe-haut,...) Et la fréquence de coupure du filtre numérique.

PROJET A REALISER

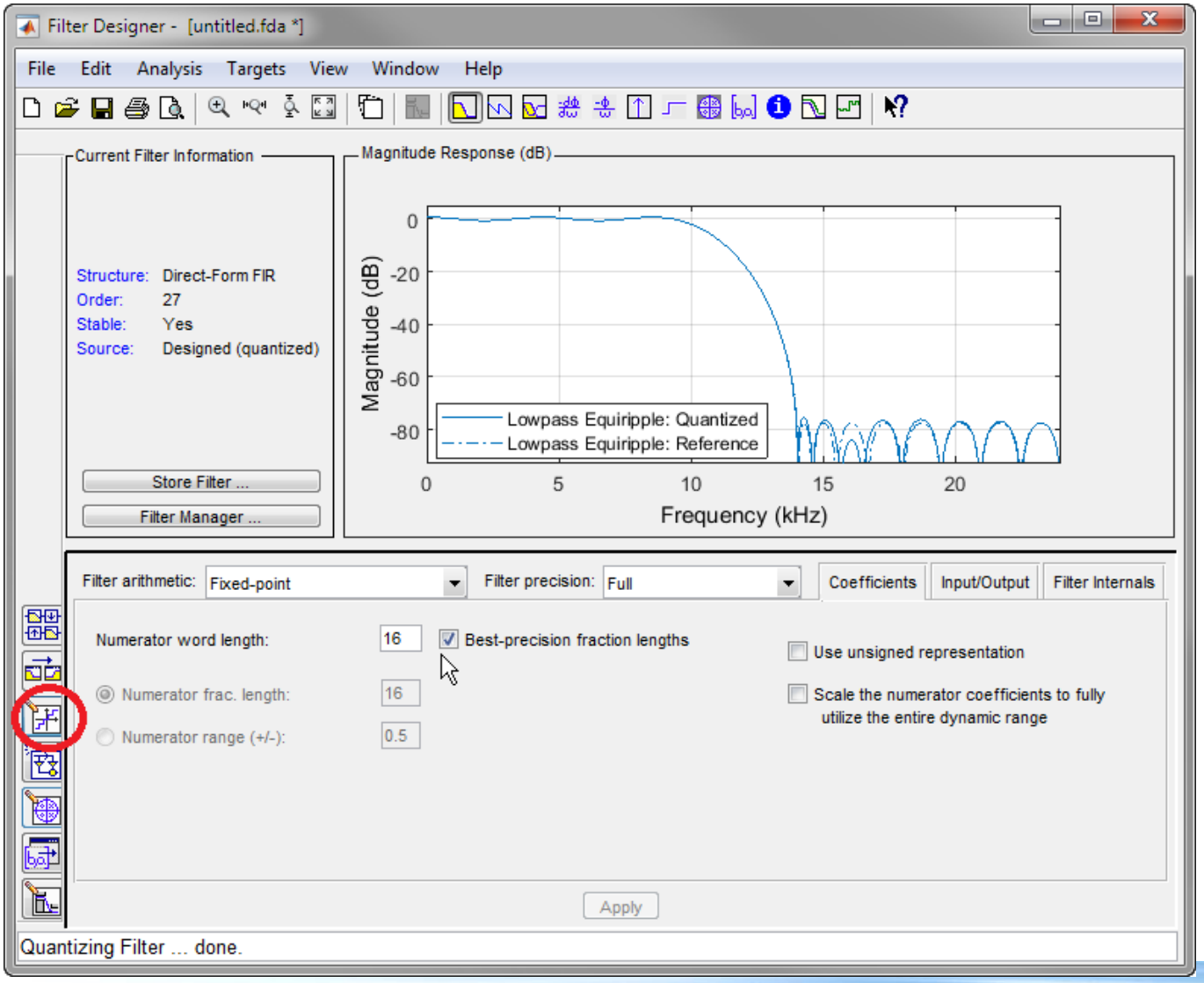
- Utiliser les **toolboxes** suivants pour synthétiser un filtre numérique
  - ***Filterbuilder***
  - ***fdatool***
- On testera pour chaque type (RIF, RII) plusieurs formes (passe bas, passe haut, passe bande et réjecteur de bande).
- Implémenter le filtre synthétisé en utilisant une carte ARDUINO.

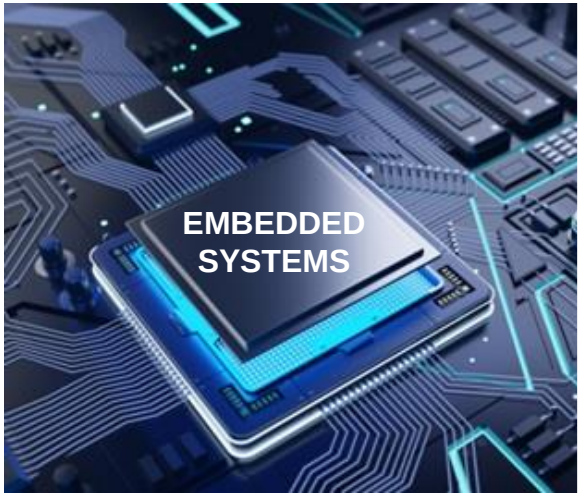
>>Filterbuilder





>>fdatool


Filter Designer - [untitled.fda *]
File Edit Analysis Targets View Window Help
Current Filter Information
Structure: Direct-Form FIR
Order: 27
Stable: Yes
Source: Designed (quantized)
Store Filter ...
Filter Manager ...
Magnitude Response (dB)
Magnitude (dB)
0
-20
-40
-60
-80
Lowpass Equiripple: Quantized
Lowpass Equiripple: Reference
0
5
10
15
20
Frequency (kHz)
Filter arithmetic: Fixed-point
Filter precision: Full
Coefficients
Input/Output
Filter Internals
Numerator word length: 16
Best-precision fraction lengths
Use unsigned representation
Numerator frac. length: 16
Scale the numerator coefficients to fully utilize the entire dynamic range
Numerator range (+/-): 0.5
Apply
Quantizing Filter ... done.



FIN !
Questions ?