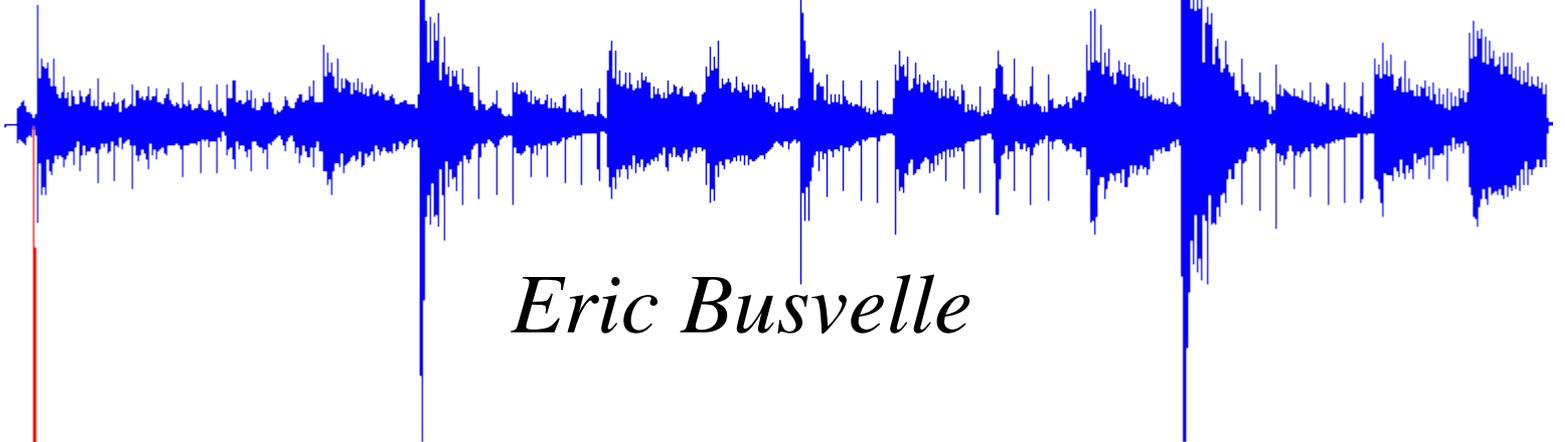


Traitement numérique du signal



Eric Busvelle



Objectifs et pré-requis

- Objectifs
 - Acquérir les outils mathématiques utiles au traitement du signal
 - Maîtriser les techniques de traitement du signal
- Pré-requis
 - M6: Mathématiques pour le signal discret
 - TR1: Modulations numériques et notions de compression

Sommaire PPN

- Notions de signaux aléatoires (bruit)
- Corrélation, orthogonalité
- Transformée de Fourier discrète 1D/2D
- Rappels sur la transformée en Z
- Filtrage numérique
- Applications à la compression audio/vidéo

Mots clés:

FFT, filtrage numérique, compression

Indications de mise en oeuvre

- Applications aux DSP (Digital Signal Processor)
- Logiciels de calcul numérique:
 - Matlab
 - Scilab
 - Octave



Supports de cours

TECHNOSUP

Les FILIÈRES TECHNOLOGIQUES des ENSEIGNEMENTS SUPÉRIEURS

SIGNAUX ET SYSTÈMES

Débuter en traitement
numérique du signal

Applications au filtrage et au traitement des sons
Cours et exercices résolus

Jean-Noël MARTIN



Débuter en traitement numérique du signal
Application au filtrage et au traitement des sons

Jean-Noël Martin
Prof. À l'IUT d'Annecy

**Analyse et contrôle
numérique du signal**

Philippe Destuynder, Prof. CNAM

Françoise Santi, Prof. CNAM

TECHNOSUP

Les FILIÈRES TECHNOLOGIQUES des ENSEIGNEMENTS SUPÉRIEURS

CALCUL SCIENTIFIQUE

Analyse et contrôle
numérique du signal

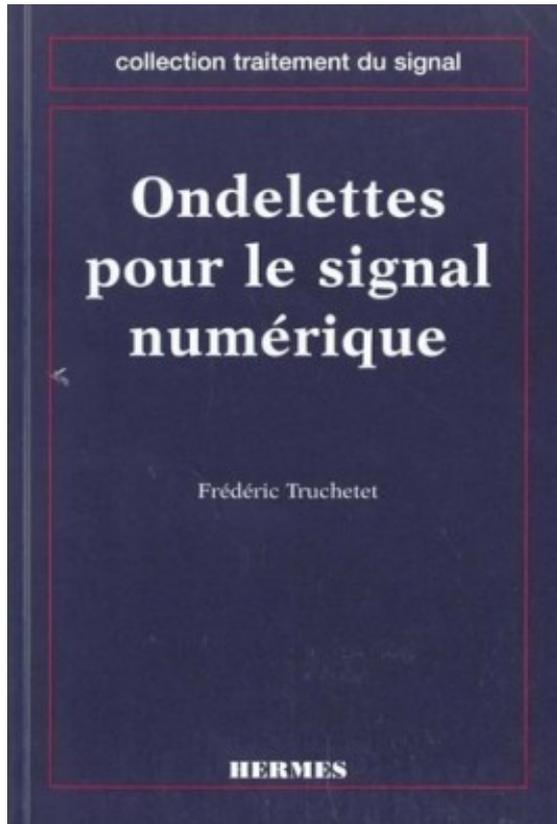
Philippe DESTUYNDER Françoise SANTI



Pour aller plus loin...

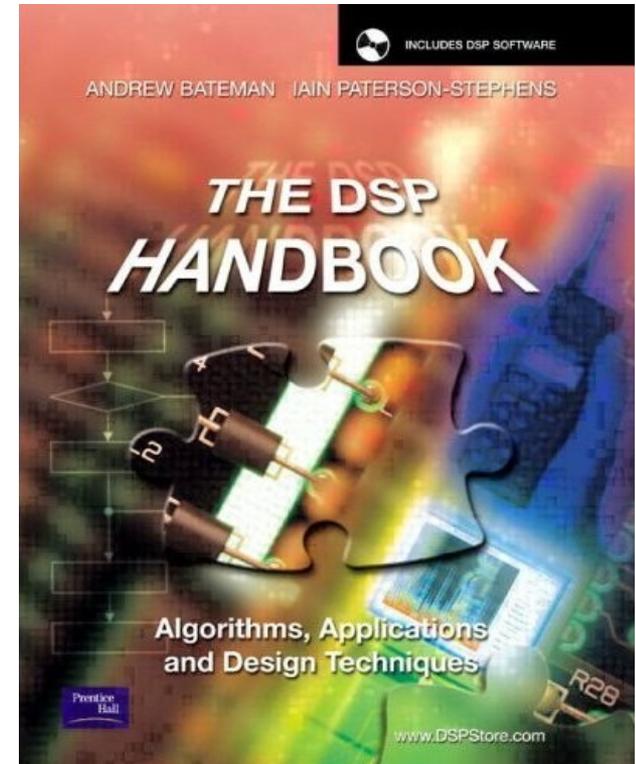
Ondelettes pour le signal numérique

Frédéric Truchetet, Prof. à l'uB



The DSP handbook Algorithms, Applications, and design techniques

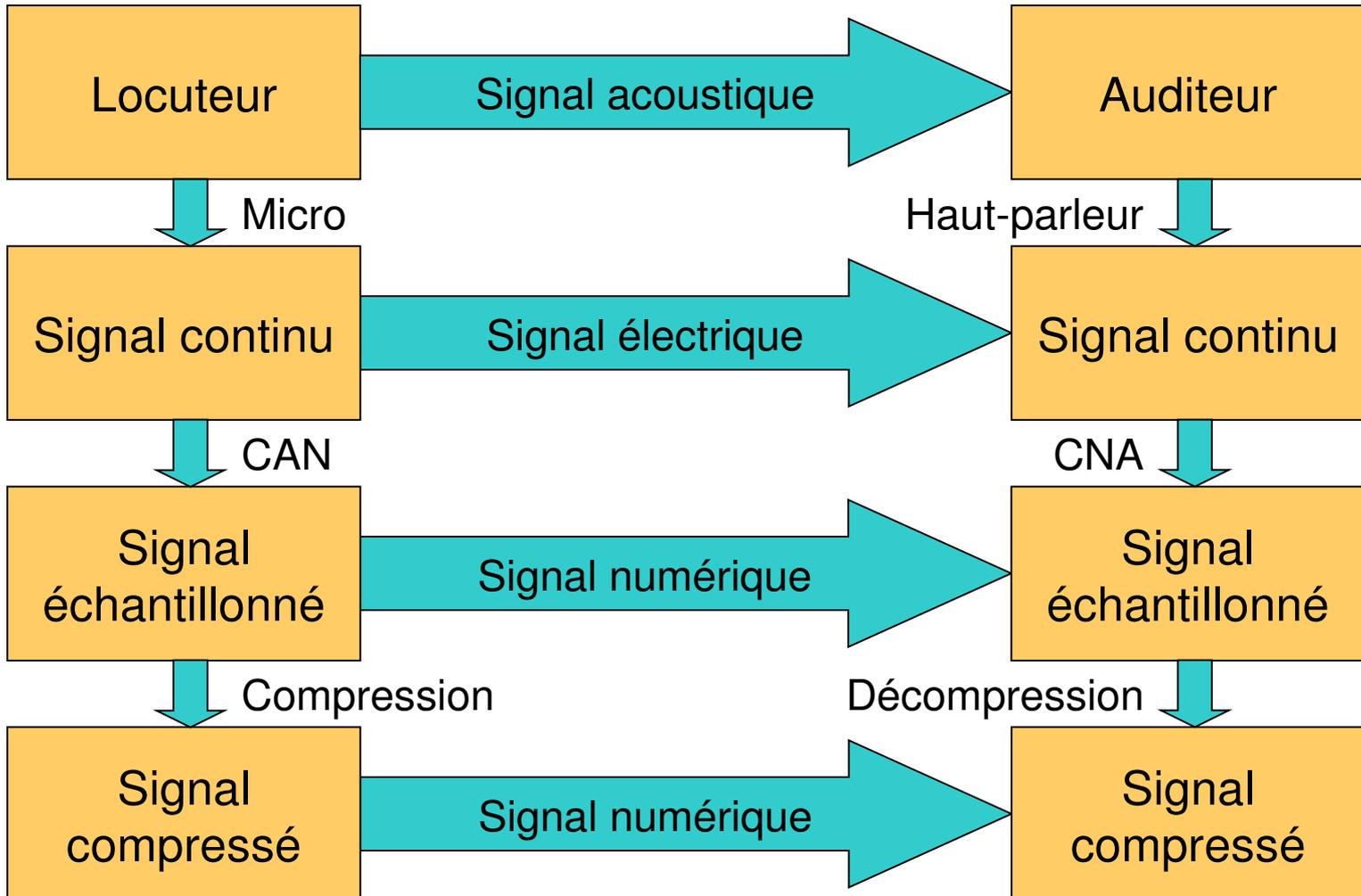
Bateman, Bateman and Paterson-Stephens



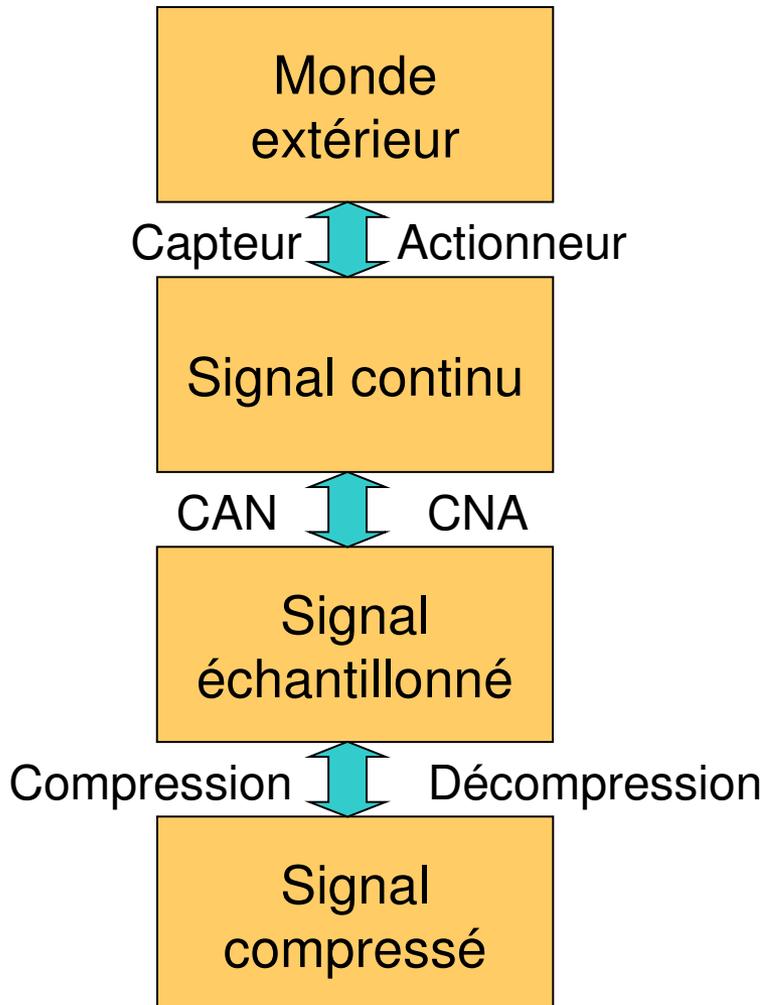
Introduction

- 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

Chaîne de transformations



Des compétences variées



- Physique/Electronique
 - Capteurs
 - Filtres
 - DSP
- Informatique
 - Programmation: Matlab (R&D)
C, assembleur, ...
- Mathématiques
 - Probabilités
 - Analyse de Fourier
 - Transformée (Z, Laplace)

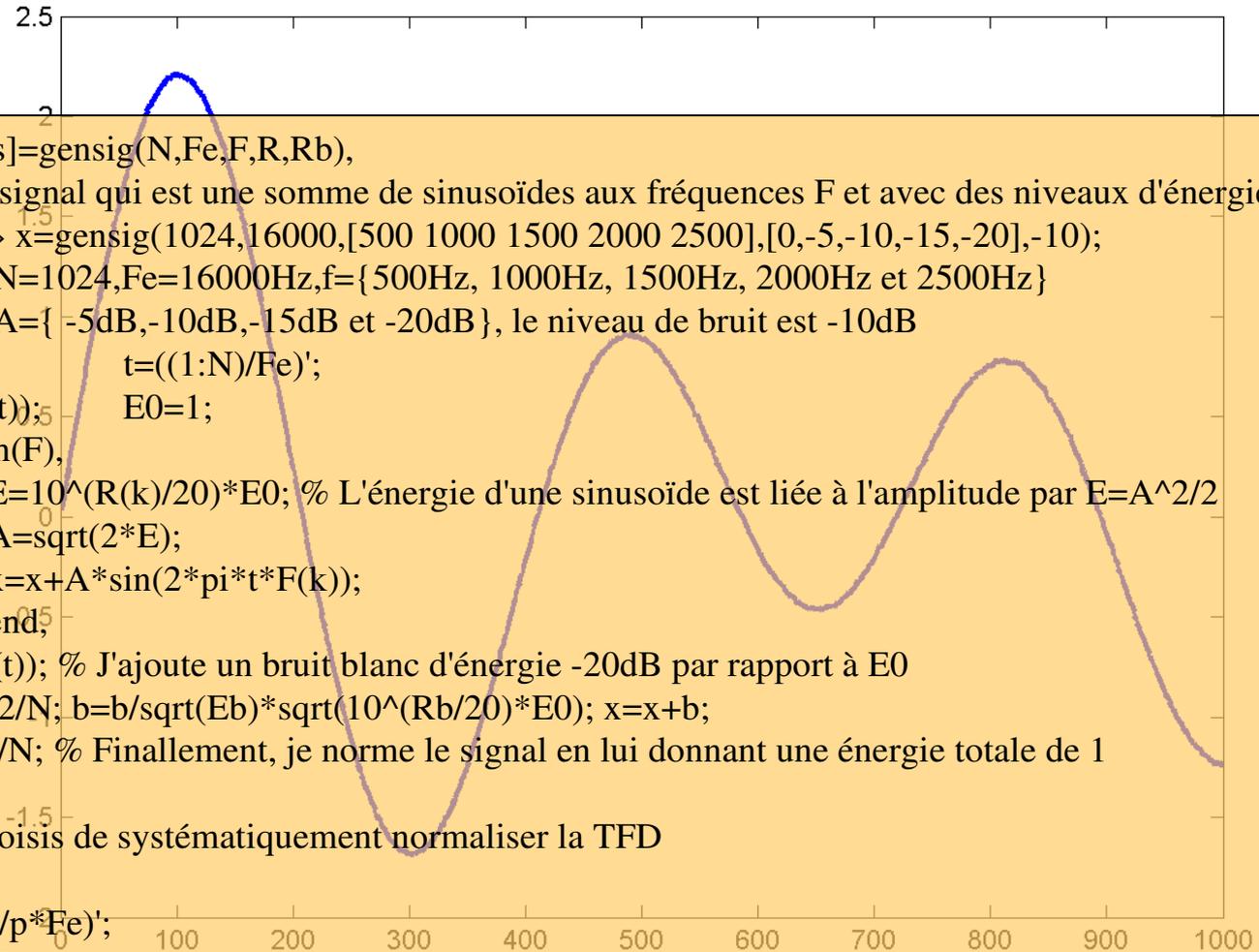
Plan du cours

La structure du cours reprend les étapes de traitement:

- I Échantillonnage
- II Conversion Analogique ↔ Numérique
- III Analyse des systèmes numériques
- IV Filtrage numérique
- V Application: Traitement numérique des sons (TPs)

Echantillonneur-bloqueur

Signal continu:



```

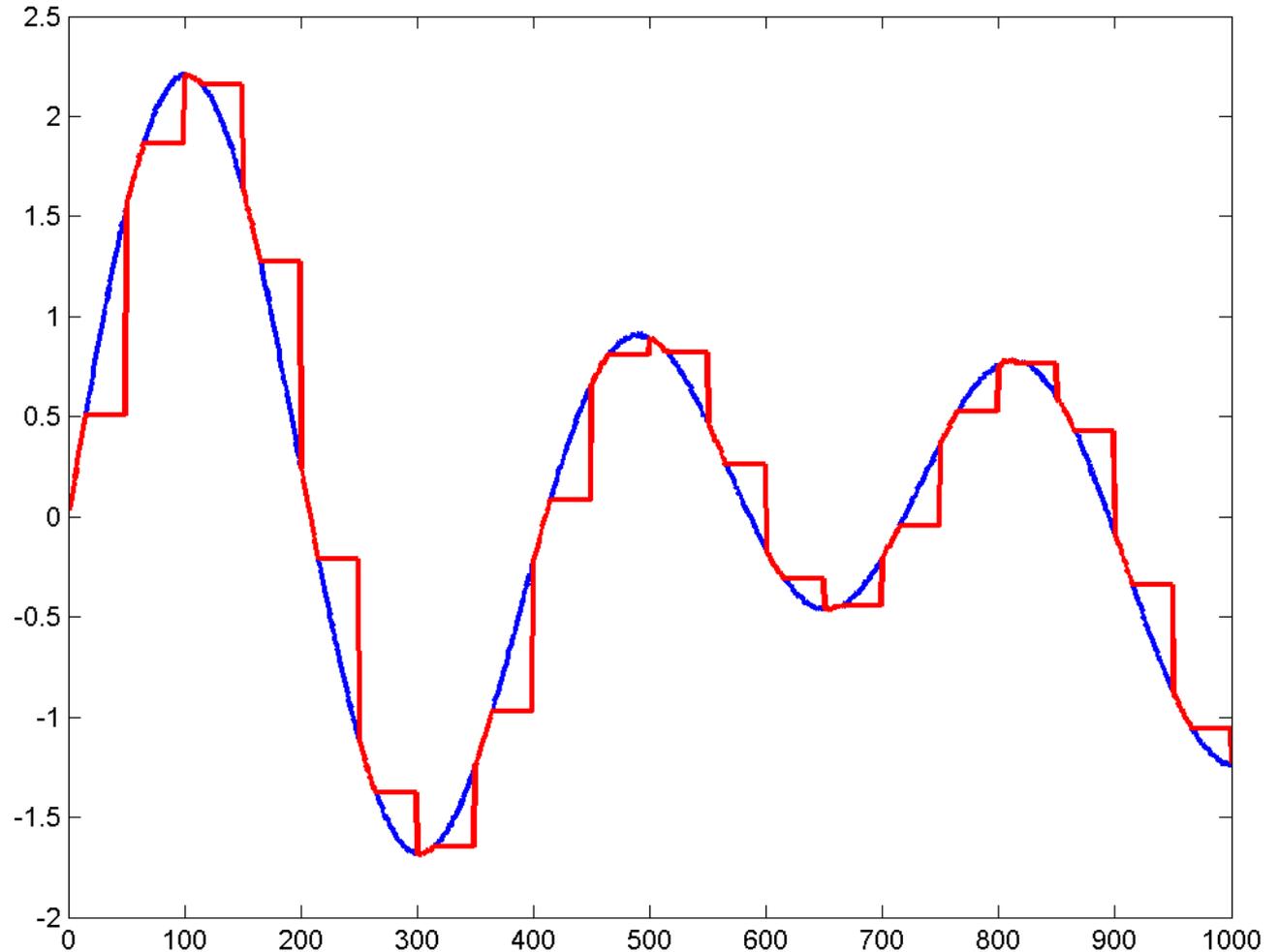
function [x,f,s]=gensig(N,Fe,F,R,Rb),
% Génère un signal qui est une somme de sinusôides aux fréquences F et avec des niveaux d'énergie R.
% Exemple: » x=gensig(1024,16000,[500 1000 1500 2000 2500],[0,-5,-10,-15,-20],-10);
>> N1=1000; % N1=1000;
>> [x,f,s]=gensig(N1, N=1024,Fe=16000Hz,f={500Hz, 1000Hz, 1500Hz, 2000Hz et 2500Hz}
16000, % A={ -5dB,-10dB,-15dB et -20dB}, le niveau de bruit est -10dB
[30 40 45] Te=1/Fe; t=((1:N)/Fe)';
[0,0,0], -8 x=zeros(size(t)); E0=1;
for k=1:length(F),
E=10^(R(k)/20)*E0; % L'énergie d'une sinusôide est liée à l'amplitude par E=A^2/2
A=sqrt(2*E);
x=x+A*sin(2*pi*t*F(k));
end;
b=randn(size(t)); % J'ajoute un bruit blanc d'énergie -20dB par rapport à E0
Eb=norm(b)^2/N; b=b/sqrt(Eb)*sqrt(10^(Rb/20)*E0); x=x+b;
E=norm(x)^2/N; % Finalement, je norme le signal en lui donnant une énergie totale de 1
x=x/sqrt(E);
p=N; % Je choisis de systématiquement normaliser la TFD
y=fft(x,p)/p;
f=((0:(p/2-1))/p*Fe)';
s=abs(y(1:(p/2)));

```

Echantillonneur-bloqueur

Signal bloqué:

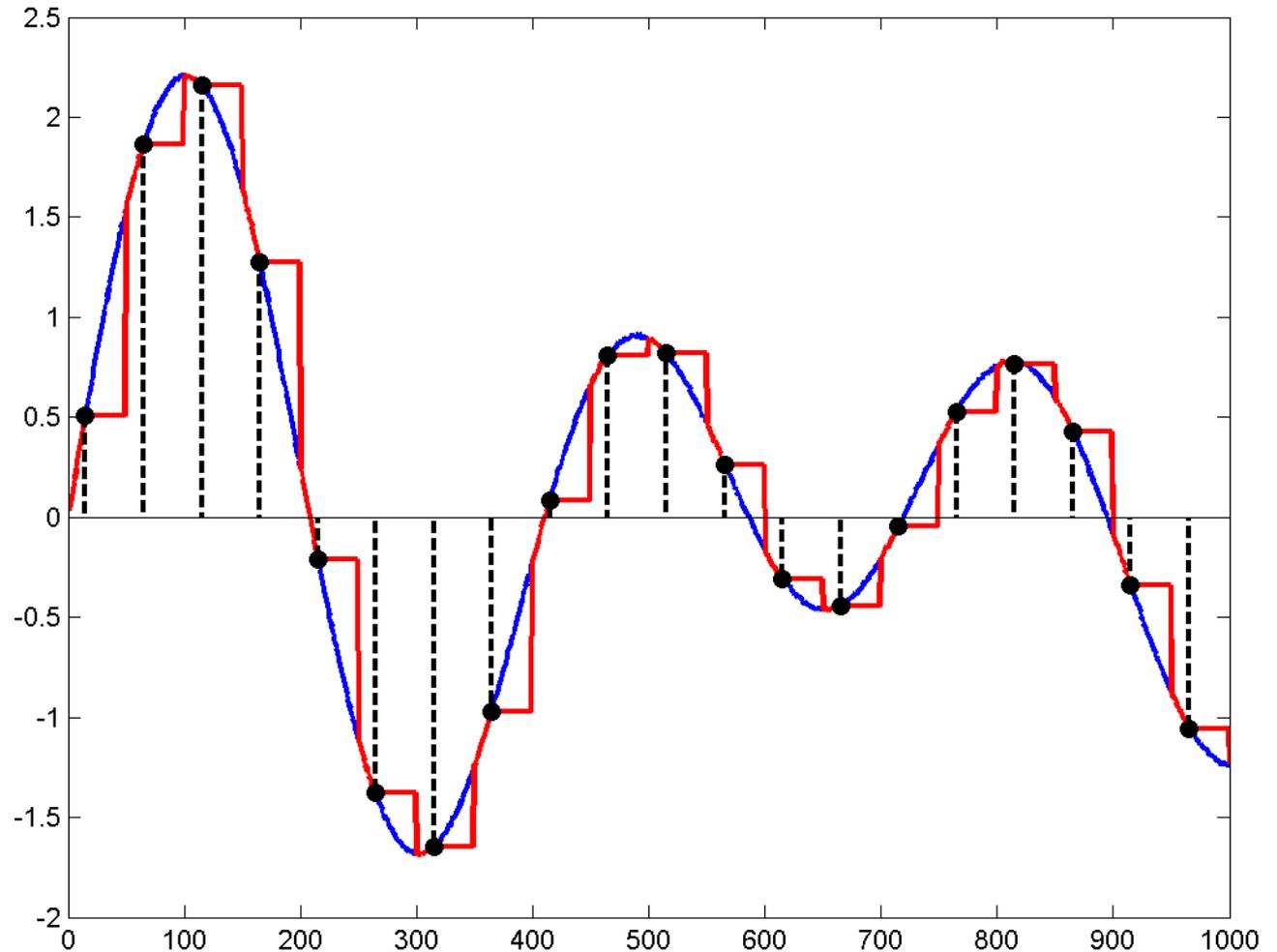
```
>> N0=15;  
>> dN=50;  
>> xb=x;  
>> for i=N0:dN-1;  
>>     xb(i:dN:N1)  
       =xb(N0:dN:N1);  
>> end,
```



Echantillonneur-bloqueur

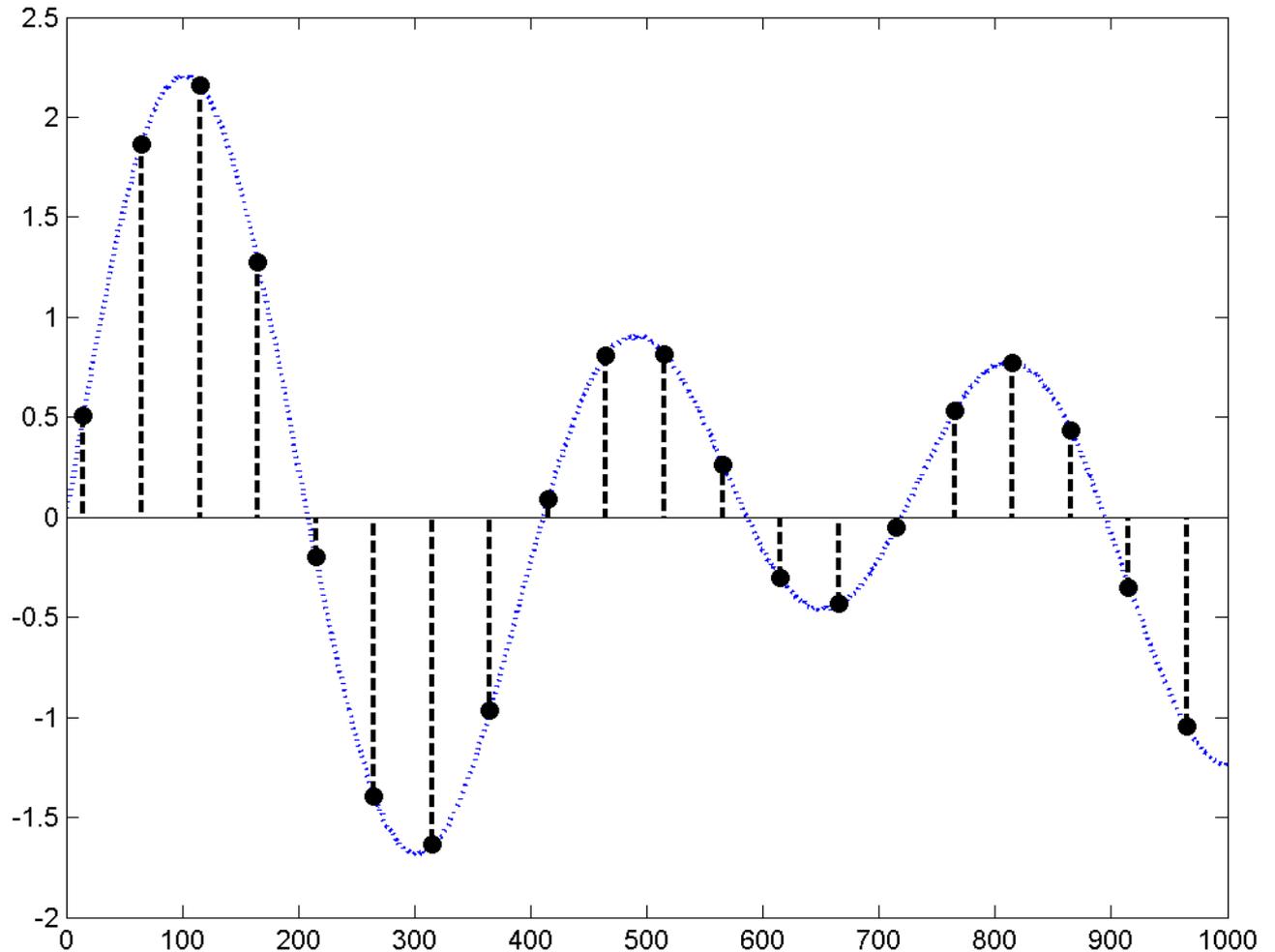
Signal
échantillonné:

```
>> xd=xb (N0:dN:N1);
```



Reconstruction

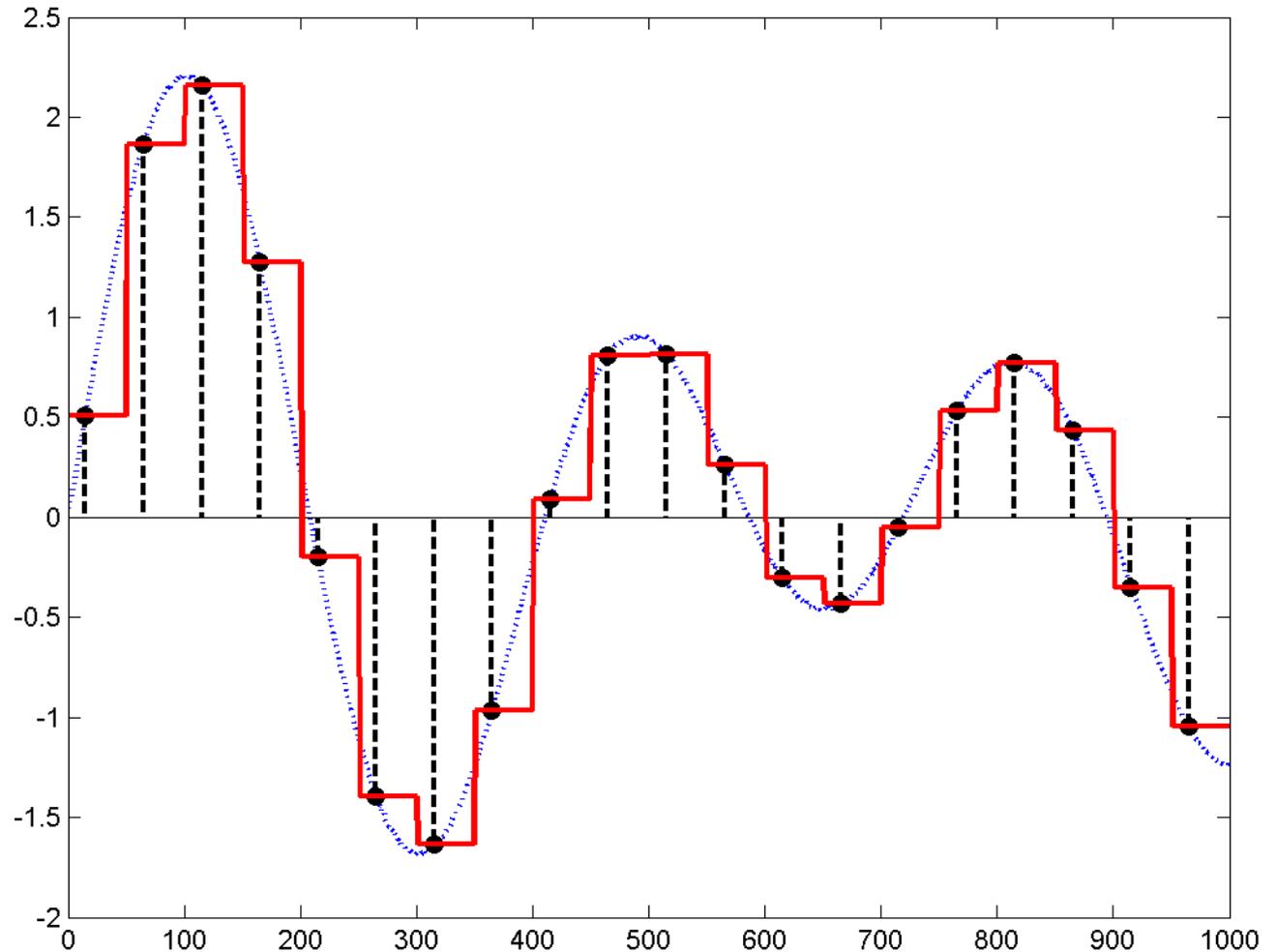
Signal
échantillonné:



Reconstruction

Signal bloqué
reconstruit:

```
>> for i=1:dN;  
>>     xr(i:dN:N1)=xd;  
>> end,
```



Reconstruction

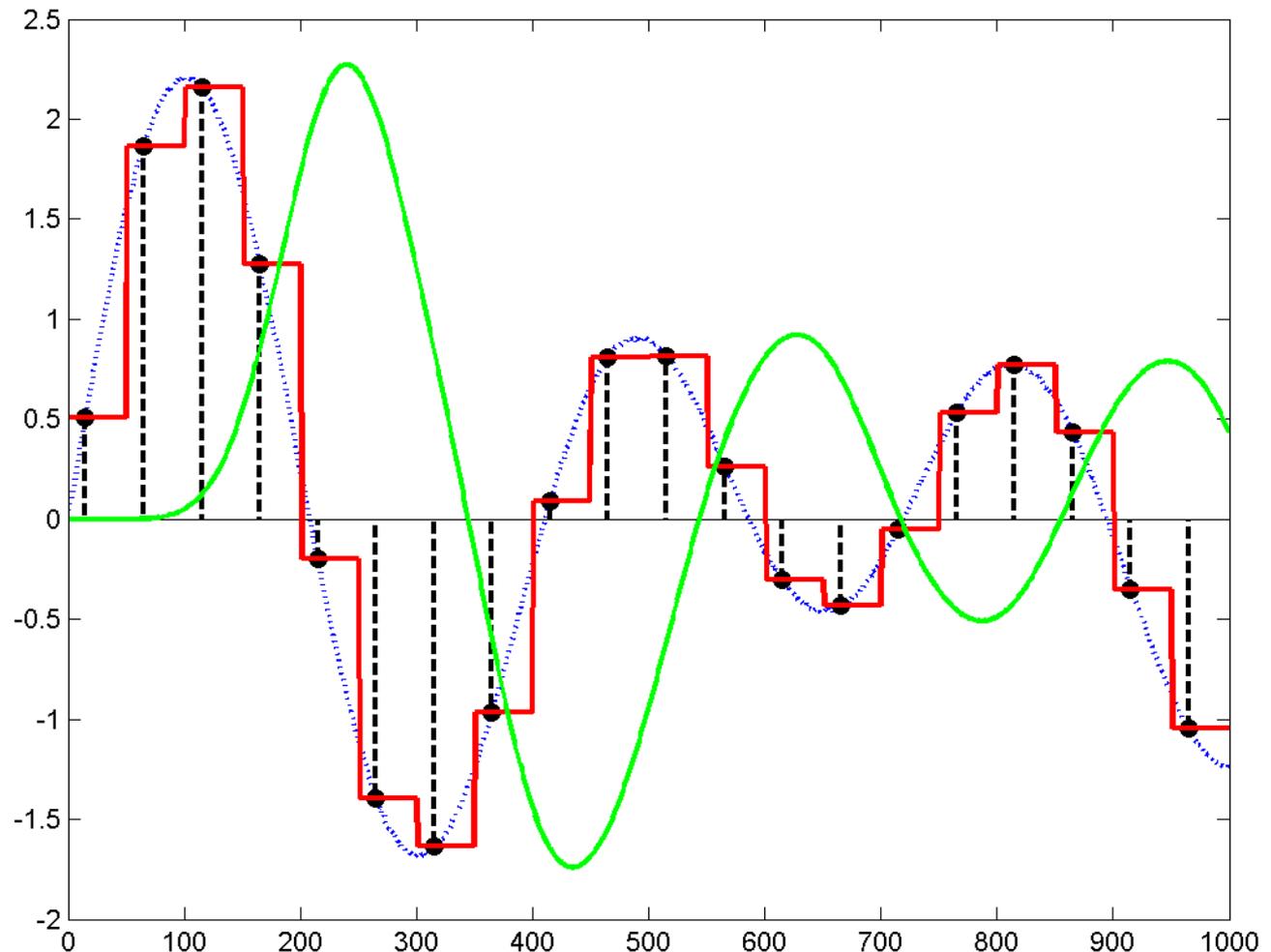
Signal filtré:

Synthèse du filtre
numérique passe-
bas (Chebyshev):

```
>> Fc=200;
>> Rp=0.5;
>> Wn=1/dN;
>> [b, a]
    =cheby1(9, Rp, Wn);
```

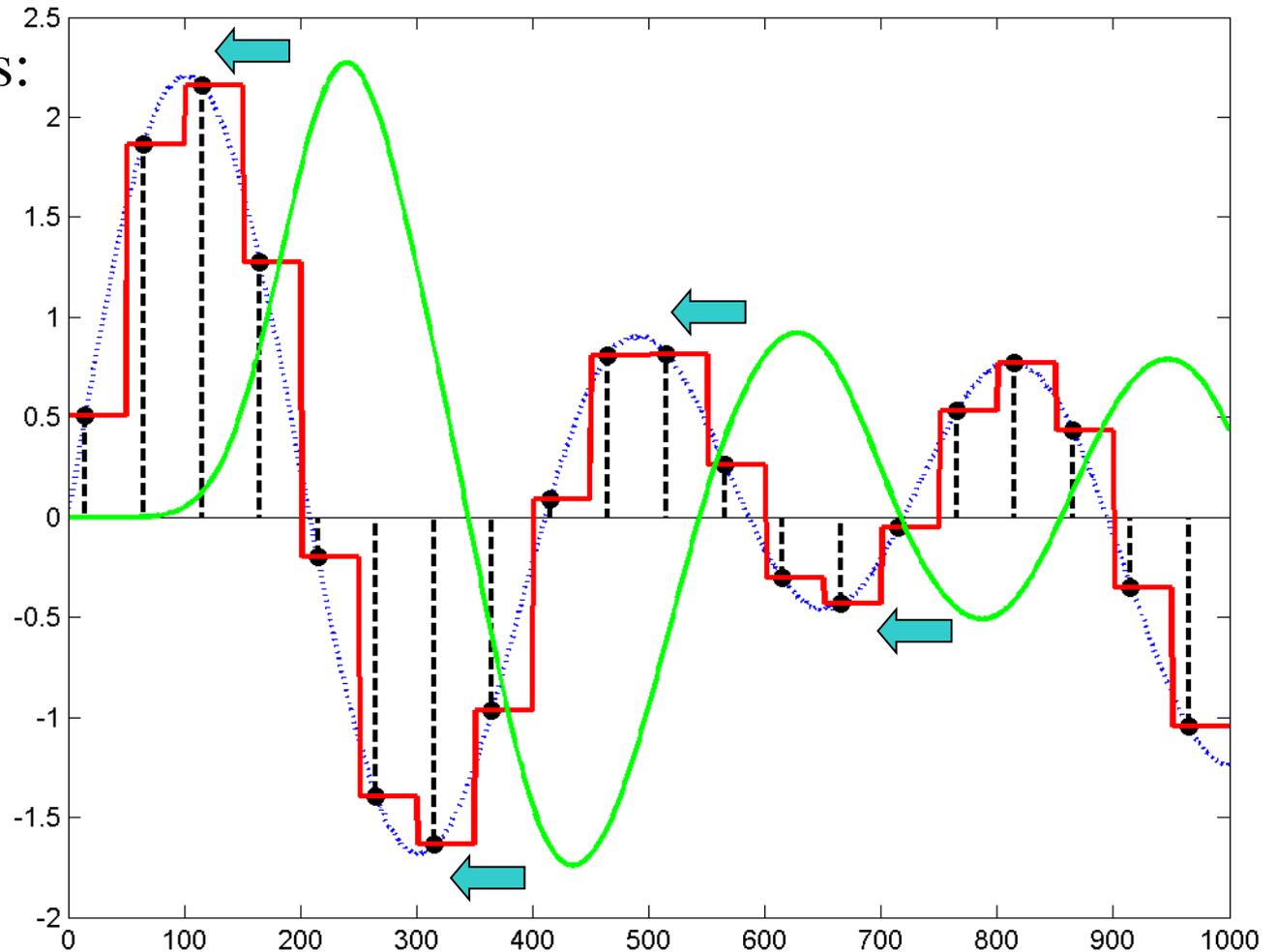
Filtrage:

```
>> xf=filter(b, a, xr);
```



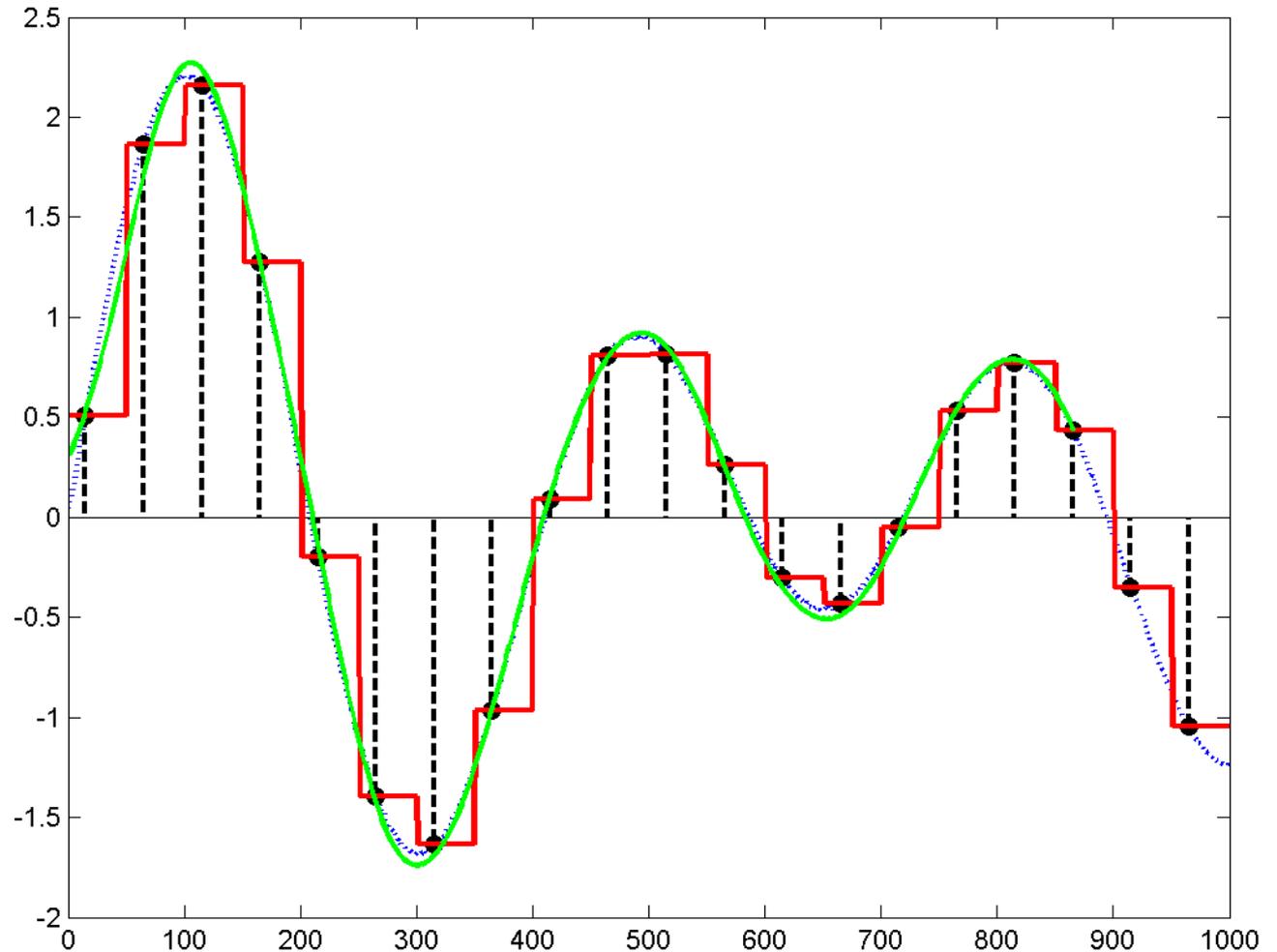
Reconstruction

Signaux superposés:

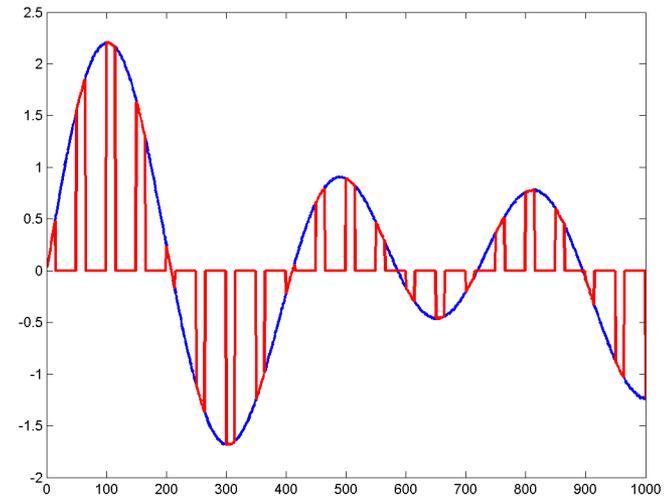
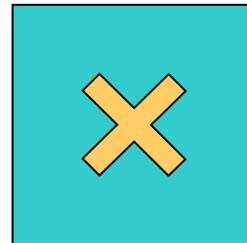
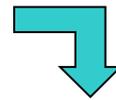
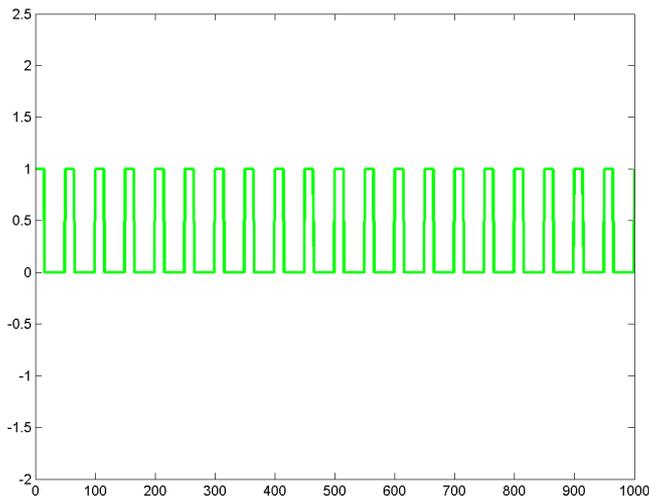
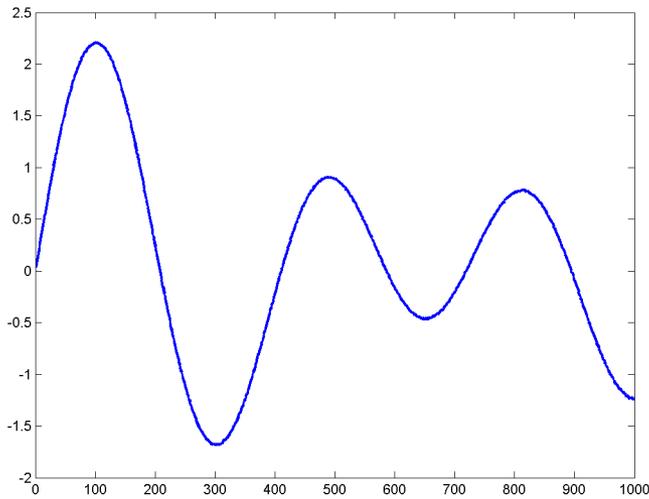


Reconstruction

Résultat final:



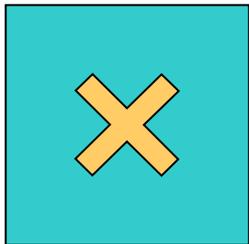
Etude spectrale d'un échantillonneur-bloqueur



Un échantillonneur est le produit entre le signal et un peigne, idéalement un peigne de Dirac.

Produit de deux sinusoides (déphasées)

$$X_1(t) = A_1 \sin(2\pi f_1 t)$$



$$\begin{aligned} X_1(t) X_2(t) &= A_1 A_2 \sin(2\pi f_1 t) \cos(2\pi f_2 t) \\ &= \frac{A_1 A_2}{2} \left[\sin(2\pi(f_1 + f_2)t) + \sin(2\pi(f_1 - f_2)t) \right] \end{aligned}$$



$$X_2(t) = A_2 \cos(2\pi f_2 t)$$

Le produit de deux sinusoides de fréquences f_1 et f_2 fait apparaître deux autres sinusoides à des fréquences $f_1 + f_2$ et $f_1 - f_2$

Etude spectrale d'un peigne

Etudions le cas classique d'une sinusoïde de fréquence f_0

$$X(t) = A_0 \sin(2\pi f_0 t)$$

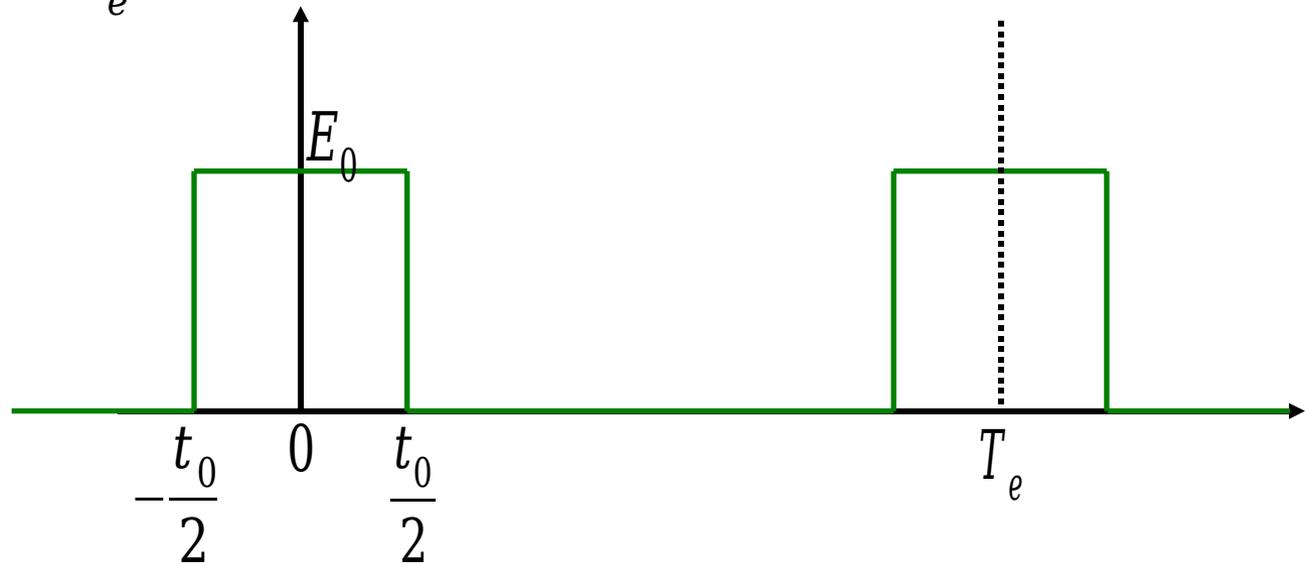
et du peigne

$$P(t) = \begin{cases} E_0 & \text{si } \text{frac}\left(\frac{t}{T_e}\right) \leq \frac{t_0}{2T_e} \\ 0 & \text{sinon} \end{cases}$$

Signaux continus
périodiques



Séries de Fourier



Développement en série de Fourier

Le développement en série de Fourier présente un instrument d'analyse des signaux analogiques T_0 -**périodiques**.



$$X(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(2\pi k f_0 t) + b_k \sin(2\pi k f_0 t)$$

$$a_0 = \frac{1}{T_0} \int_0^{T_0} X(t) dt$$

$$a_k = \frac{2}{T_0} \int_0^{T_0} X(t) \cos(2\pi k f_0 t) dt$$

$$b_k = \frac{2}{T_0} \int_0^{T_0} X(t) \sin(2\pi k f_0 t) dt$$

Convention: $f_0 = \frac{1}{T_0}$ et $\omega_0 = 2\pi f_0$

Joseph Fourier, né à Auxerre en 1768

Développement en série de Fourier

La série de Fourier peut aussi s'exprimer sous la forme équivalente

$$X(t) = \sum_{k=-\infty}^{\infty} c_k \exp(2i\pi k f_0 t) \text{ avec } c_k = \frac{1}{T_0} \int_0^{T_0} X(t) \exp(-2i\pi k f_0 t) dt$$

avec les relations entre les deux définitions:

$$c_k = \begin{cases} a_0 & \text{si } k=0 \\ \frac{a_k - b_k i}{2} & \text{si } k \geq 1 \\ \bar{c}_{-k} & \text{sinon} \end{cases}$$

Identité de Parseval

Si le signal est d'énergie finie $\int_0^{T_0} |X(t)|^2 dt < \infty$ on a les relations

$$\int_0^{T_0} |X(t)|^2 dt = a_0 + \frac{1}{2} \sum_{k=1}^{\infty} a_k^2 + b_k^2 = \sum_{k=-\infty}^{\infty} |c_k|^2$$

Remarque: Pour un signal entre a et b , on a, en posant $c = \frac{a+b}{2}$

$$X(t) = \sum_{k=-\infty}^{\infty} c_k \exp\left(2i\pi k \frac{t-m}{b-a}\right) \text{ avec } c_k = \frac{1}{b-a} \int_a^b X(t) \exp\left(-2i\pi k \frac{t-c}{b-a}\right) dt$$

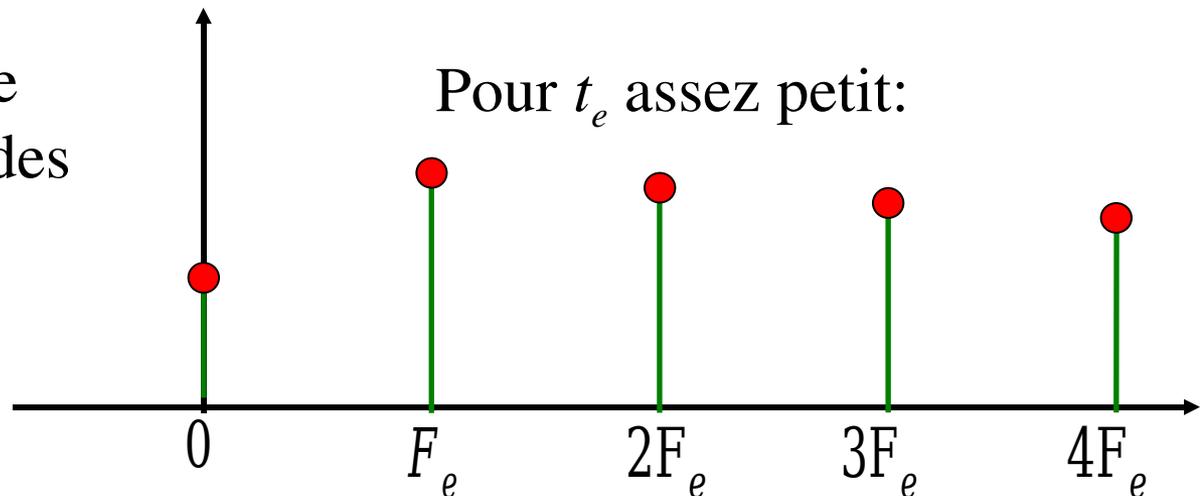
Exercice: Trouver la formule équivalente pour la définition en sinus-cosinus

Analyse de Fourier de l'échantillonnage

D'après les formules précédentes:

$$P(t) = \frac{E_0 t_0}{T_e} \left(1 + 2 \sum_{k=1}^{\infty} \frac{\sin(\pi k F_e t_0)}{\pi k F_e t_0} \cos(2\pi k F_e t) \right)$$

Le développement en série de Fourier fait apparaître des raies aux fréquences multiples de la fréquence d'échantillonnage F_e .

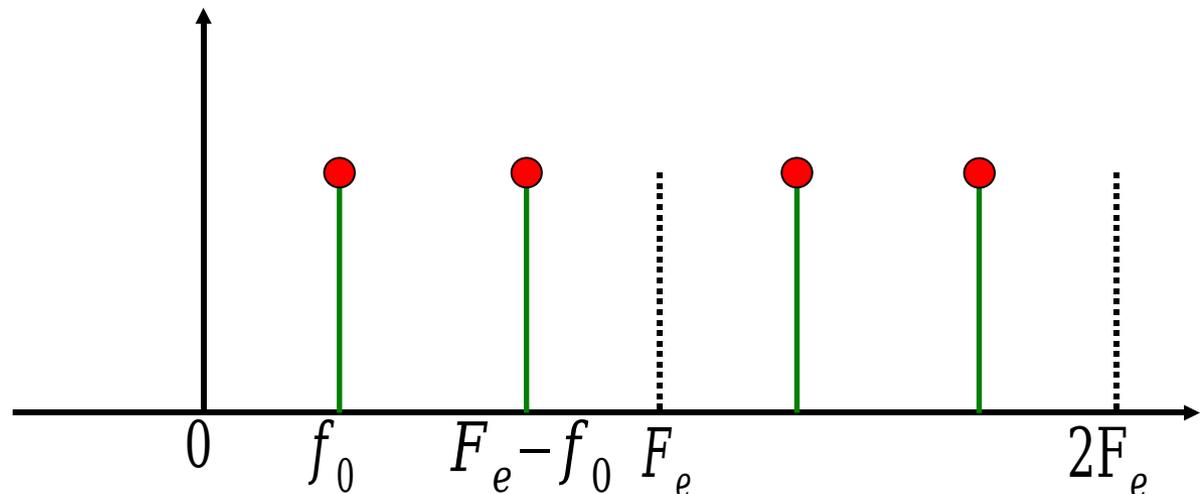


Analyse de Fourier de l'échantillonnage

En vertu du principe de superposition, le produit d'une sinusoïde de fréquence f_0 et du peigne de Dirac fera apparaître des raies aux fréquences $f_0, F_e \pm f_0, 2F_e \pm f_0, \dots$

Si $f_0 < F_e - f_0$ alors on retrouvera bien le pic correspondant au signal.

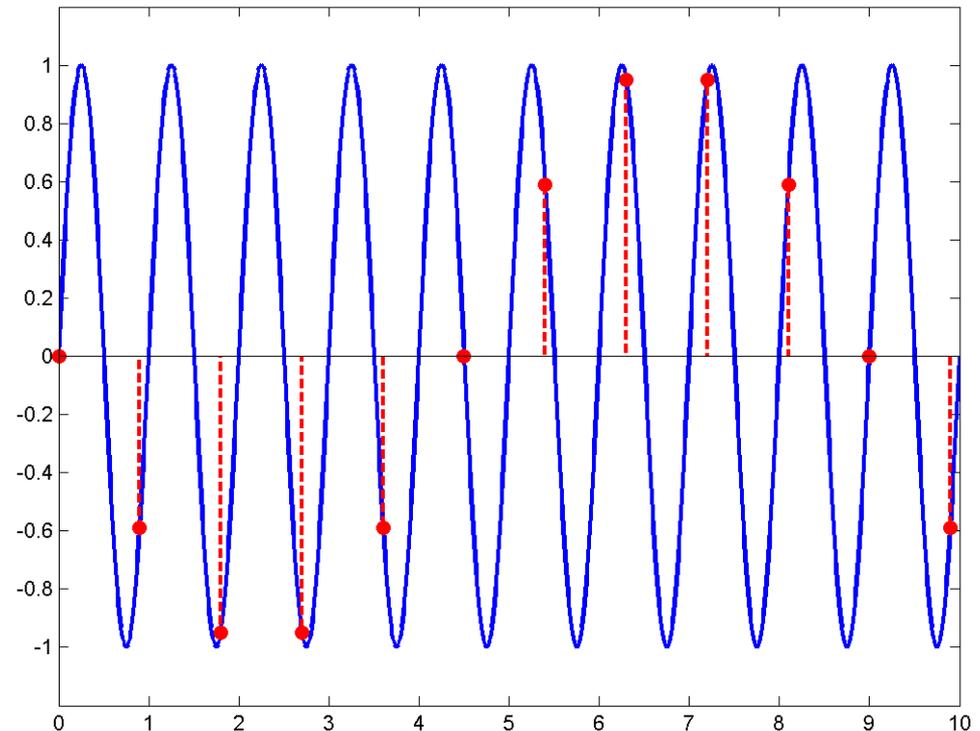
Mais sinon, aïe aïe !



Analyse de Fourier de l'échantillonnage

Si les fréquences du signal sont toutes inférieures à $F_e/2$ alors le signal ne sera pas détérioré.

Si des fréquences sont supérieures à la moitié de la fréquence d'échantillonnage, alors il y a déformation du spectre par recouvrement.

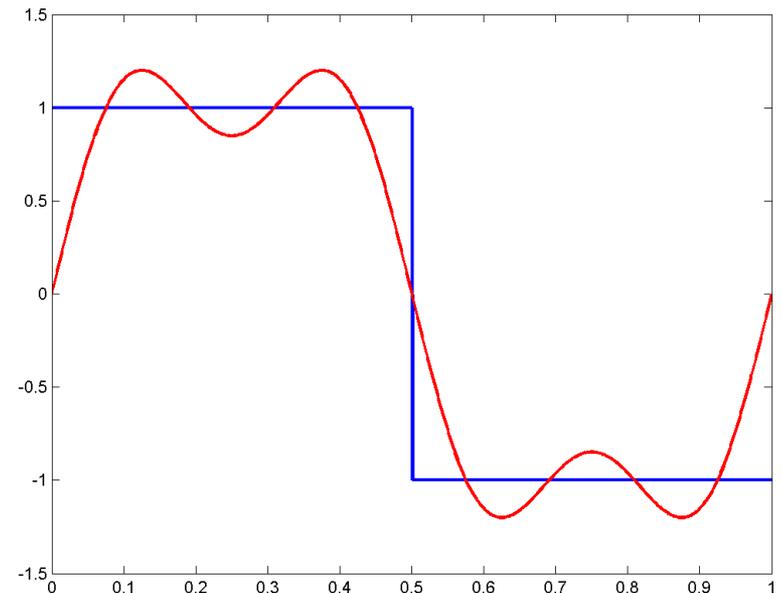


Phénomène de Gibbs

Les signaux discontinus sont eux aussi très détériorés par l'oubli des hautes fréquences: c'est le **phénomène de Gibbs**.

$$S(t) = \frac{4}{\pi} \sum_{n=1}^{+\infty} \frac{\sin(2\pi nt)}{n}$$

$$S(t) = \frac{4}{\pi} \left(\sin(2\pi t) + \frac{1}{2} \sin(4\pi t) \right)$$

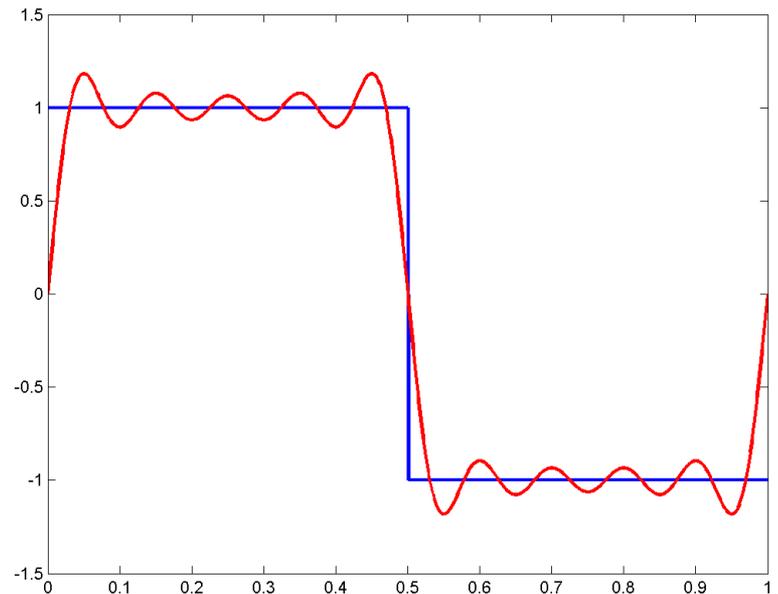


Phénomène de Gibbs

Les signaux discontinus sont eux aussi très détériorés par l'oubli des hautes fréquences: c'est le **phénomène de Gibbs**.

$$S(t) = \frac{4}{\pi} \sum_{n=1}^{+\infty} \frac{\sin(2\pi nt)}{n}$$

$$S(t) = \frac{4}{\pi} \sum_{n=1}^5 \frac{\sin(2\pi nt)}{n}$$

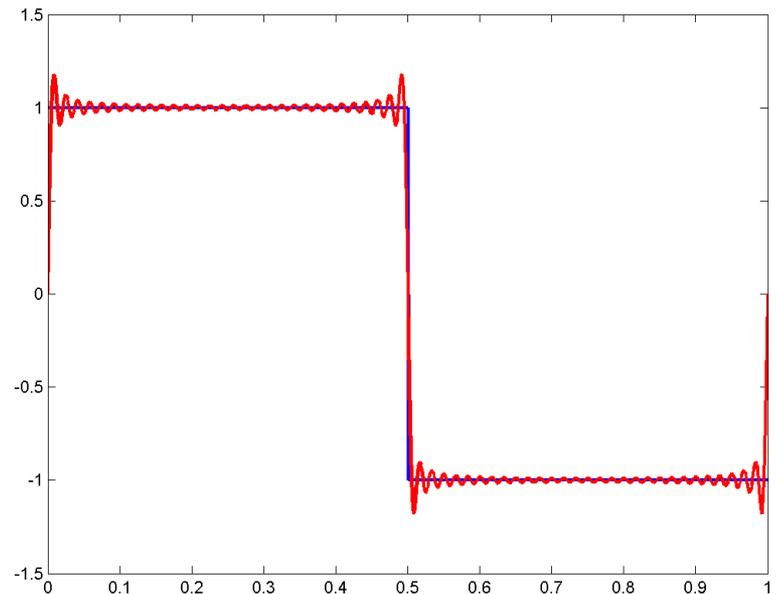


Phénomène de Gibbs

Les signaux discontinus sont eux aussi très détériorés par l'oubli des hautes fréquences: c'est le **phénomène de Gibbs**.

$$S(t) = \frac{4}{\pi} \sum_{n=1}^{+\infty} \frac{\sin(2\pi nt)}{n}$$

$$S(t) = \frac{4}{\pi} \sum_{n=1}^{30} \frac{\sin(2\pi nt)}{n}$$

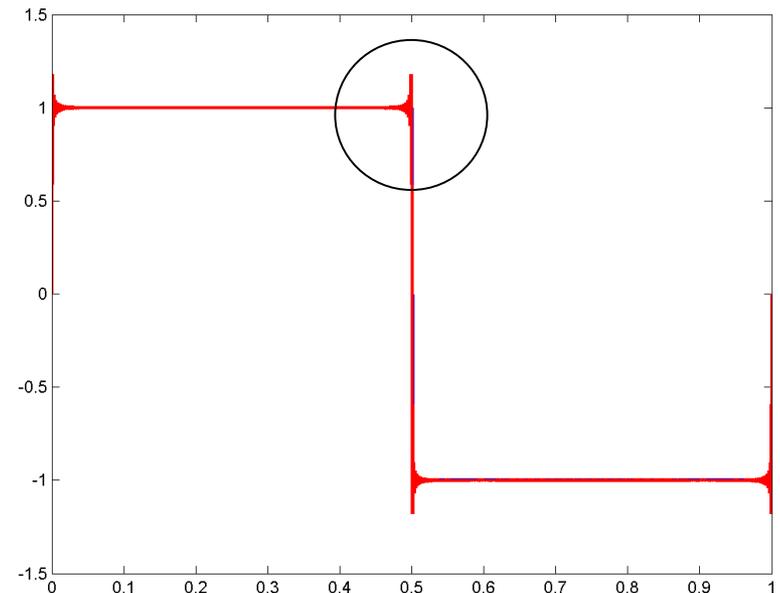


Phénomène de Gibbs

Les signaux discontinus sont eux aussi très détériorés par l'oubli des hautes fréquences: c'est le **phénomène de Gibbs**.

$$S(t) = \frac{4}{\pi} \sum_{n=1}^{+\infty} \frac{\sin(2\pi nt)}{n}$$

$$S(t) = \frac{4}{\pi} \sum_{n=1}^{250} \frac{\sin(2\pi nt)}{n}$$



Théorème de Shannon-Nyquist

Réciproquement, si la fréquence d'échantillonnage est assez grande par rapport au signal étudié, on a le théorème fondamental:

Théorème. Soit $X(t)$ une fonction telle que toutes les fréquences sont inférieures à f . Soit F_e une fréquence d'échantillonnage telle que $F_e > 2f$.

Alors:

$$X(t) = \sum_{n=-\infty}^{+\infty} X(nT_e) \frac{\sin(2\pi f(t - nT_e))}{2\pi f(t - nT_e)}$$

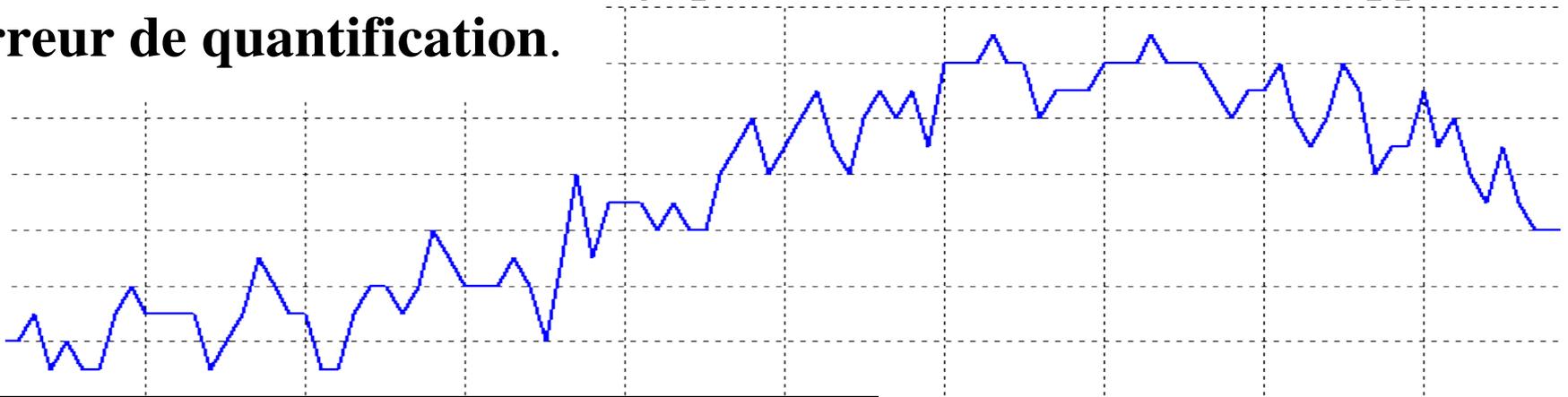
Conclusion. Il est impératif de faire passer le signal par un filtre passe-bas **analogique** dont la fréquence de coupure est inférieure à la moitié de la fréquence d'échantillonnage, **avant d'échantillonner**.

Conversion analogique-numérique

La représentation numérique en virgule fixe(*) d'une quantité analogique est caractérisée par la plage des valeurs représentables $[a, b]$ et la précision ε , c'est-à-dire la différence entre deux valeurs consécutives.

Si N est le nombre de bits de la représentation,
$$\varepsilon = \frac{b-a}{2^N}$$

L'erreur entre la valeur analogique et la valeur numérisée s'appelle **erreur de quantification**.



(*) La représentation en virgule flottante ne sera pas étudiée ici.

Conversion analogique-numérique

En plus des erreurs de quantification, un CAN $y \rightarrow b$

$$b = \text{int} \left(2^N \frac{y-a}{b-a} \right)$$

peut présenter les erreurs suivantes:

- Biais systématique (erreur sur a)
- Erreur de gain (erreur sur $b-a$)
- Non-linéarités (distorsion de la fonction linéaire)

La conversion réciproque CNA peut présenter le même type d'erreur.

Notions de processus

Un processus aléatoire (discret) est une suite de variables aléatoires $X(n)$ définie sur un espace de probabilité (Ω, A, P) .

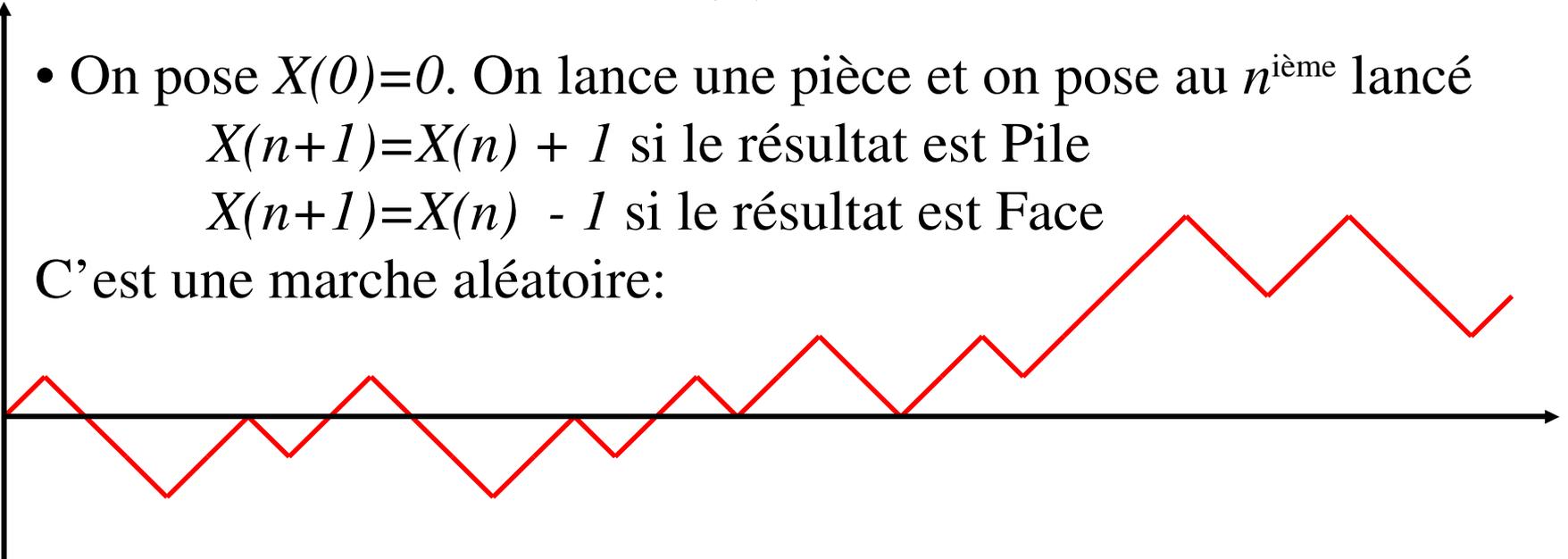
Exemples:

- Toute suite numérique est aussi un processus aléatoire
- On lance un dé et on note $X(n)$ le résultat du $n^{\text{ième}}$ résultat
- On pose $X(0)=0$. On lance une pièce et on pose au $n^{\text{ième}}$ lancé

$$X(n+1)=X(n) + 1 \text{ si le résultat est Pile}$$

$$X(n+1)=X(n) - 1 \text{ si le résultat est Face}$$

C'est une marche aléatoire:



Notions de processus

Exemples (suite):

- On note $X(n)$ le **nombre de requêtes** arrivant sur un serveur par unité de temps. C'est un cas de loi de Poisson:

$$P(X(n)=k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

- On note $X(n)$ le **temps d'attente** avant l'arrivée de la $n^{\text{ième}}$ requête sur un serveur. C'est la loi exponentielle:

$$P(a \leq X(n) < b) = \lambda \int_a^b e^{-\lambda x} dx \text{ pour } 0 \leq a < b$$

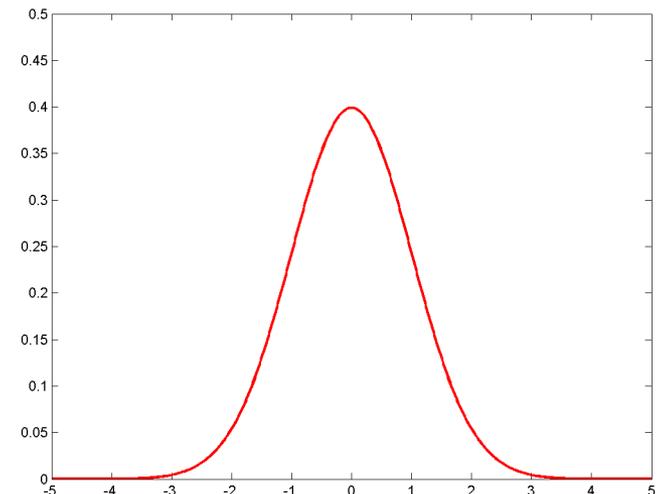
Notions de processus

Exemples (suite):

- Si les variables aléatoires $X(n)$ sont indépendantes et suivent la loi normale centrée réduite (loi de Gauss), le processus $X(n)$ s'appelle un **bruit blanc**. C'est un processus très important !

Densité de la loi de Gauss de moyenne m et de variance σ^2 :

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \frac{(x-m)^2}{\sigma^2}\right)$$



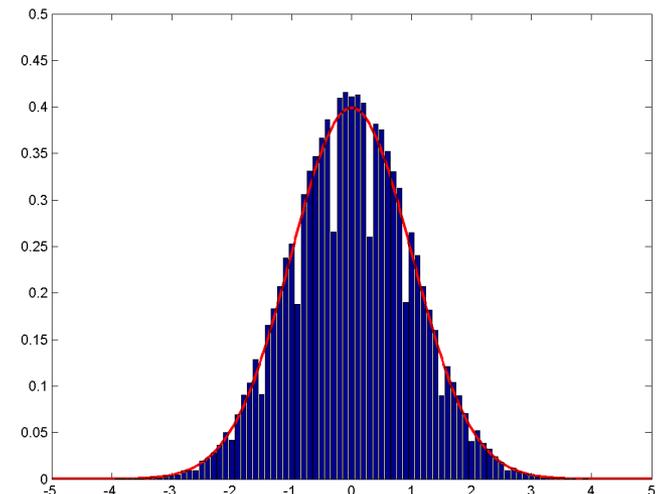
Bruit blanc

Le bruit blanc est souvent utilisé pour modéliser un bruit physique, ceci en raison du résultat mathématique suivant:

Théorème « central limite ». Si $Y(n)$ est une suite de variables aléatoires indépendantes et de même loi, de moyenne μ et de variance σ^2 , alors la loi de $S(n) = Y(1) + \dots + Y(n)$ tend vers la loi de Gauss de moyenne $n.\mu$ et de variance $n.\sigma^2$. ou de façon équivalente mais plus rigoureuse

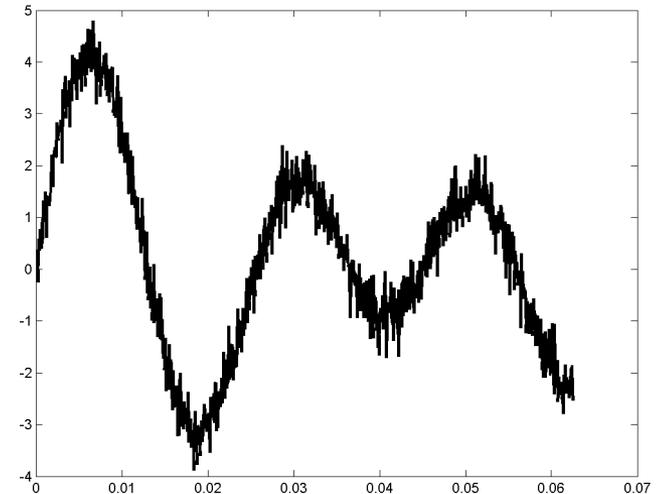
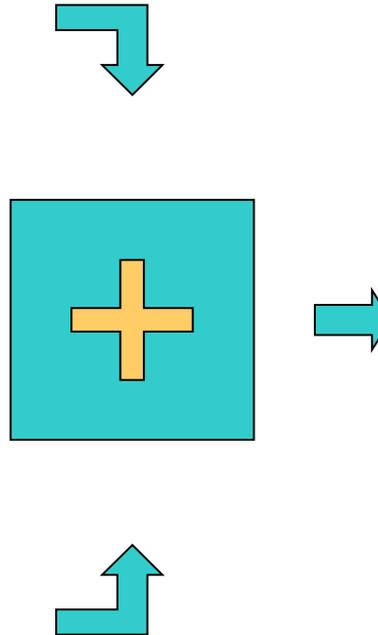
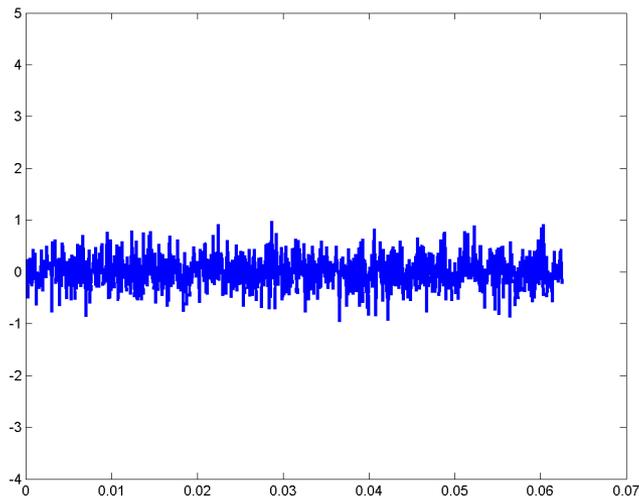
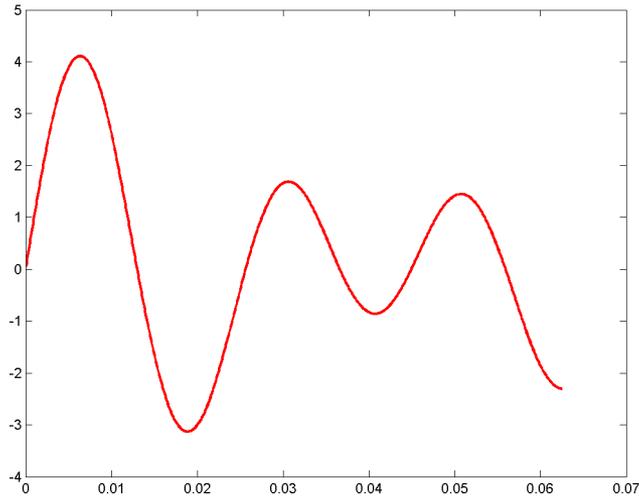
$$\frac{S(n) - n\mu}{\sigma \sqrt{n}} \xrightarrow{n \rightarrow \infty} N(0, 1)$$

Somme de n variables aléatoires
uniformes discrètes, $n = \underline{1}, \underline{2}, \underline{5}, \underline{10}, \underline{100}$



Bruit blanc

Signal utile + Bruit de mesure



Problématique: éliminer le bruit et récupérer les informations utiles, c.a.d. inverser cette opération

Processus stationnaires

On dit d'un processus aléatoire $X(n)$ qu'il est **stationnaire** si la loi de $X(n)$ ne dépend pas de n .

On dit que $X(n)$ est **stationnaire du 1^{er} ordre** si $E[X(n)] = \mu$ existe et est indépendant de n .

On dit que $X(n)$ est **stationnaire du 2nd ordre** s'il est stationnaire du 1^{er} ordre et $E[(X(i) - \mu)(X(j) - \mu)] = \gamma(|i - j|)$ existe et ne dépend pas de n .

Théorème.

Stationnaire \Rightarrow Stationnaire du 2nd ordre
 \Rightarrow Stationnaire du 1^{er} ordre

Autocovariance et autocorrélation

Pour un processus stationnaire du 2nd ordre, on note pour tout entier naturel k

$$\Gamma_X(k) = E \left[(X(i) - \mu)(X(i-k) - \mu) \right]^{(*)}$$

et $\Gamma_X(k)$ (indépendant de i) s'appelle l'**autocovariance** de X .

On définit aussi l'**autocorrélation** de X la normalisation de l'autocovariance par la variance de X

$$\gamma_X(k) = \frac{\Gamma_X(k)}{\sigma^2}$$

de sorte que $\gamma_X(0) = 1$.

(*) On a noté $\mu = E[X(n)]$ et $\sigma^2 = E[(X(n) - \mu)^2]$

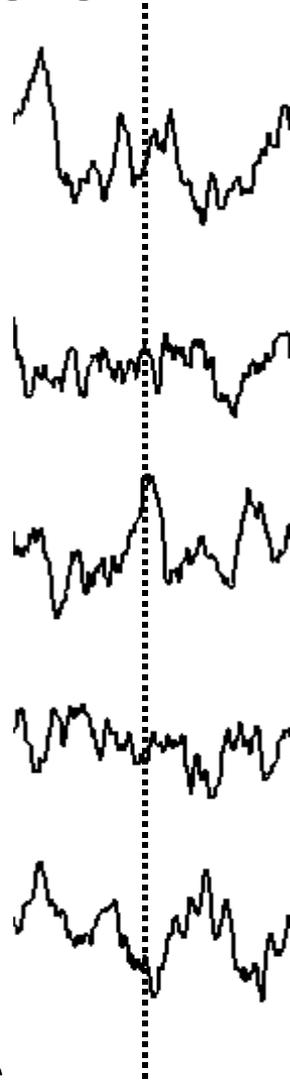
Processus ergodique

Le calcul de l'autocorrélation nécessite un calcul probabiliste, donc la connaissance précise de la loi du processus, ou son estimation en réalisant un grand nombre de fois le processus (**moyenne statistique**).

Sous certaines conditions (en particulier la stationnarité), la connaissance du processus sur une durée assez longue peut suffire à calculer l'autocorrélation, par une **moyenne temporelle**.



Moyenne temporelle
$$\bar{X} = \frac{1}{N} \sum_{k=1}^N X(k)$$



Moyenne statistique $E(X(n))$

Processus ergodique

Un processus stationnaire est dit ergodique si on peut calculer moyenne et covariances par des statistiques temporelles.

Ainsi, pour un tel processus $X(n)$, n allant de 1 à N , on a:

$$\mu = E[X(n)] \approx \frac{1}{N} \sum_{k=1}^N X(k)$$

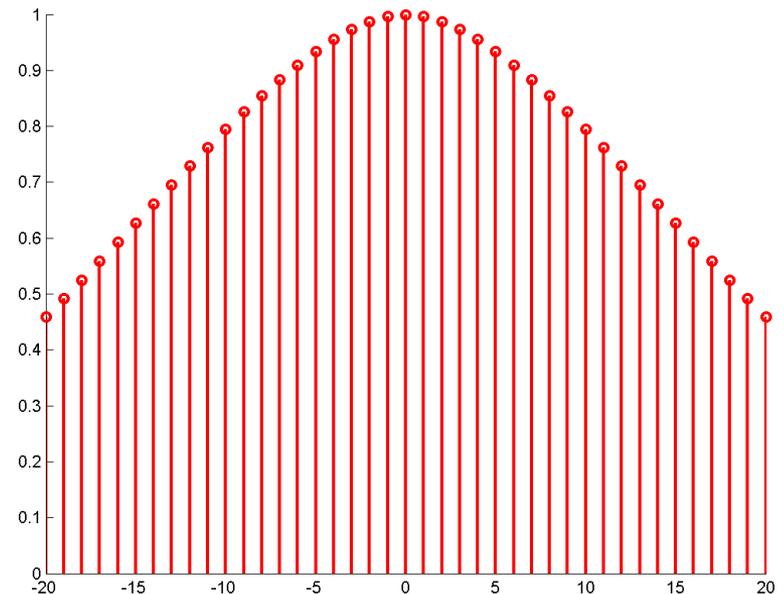
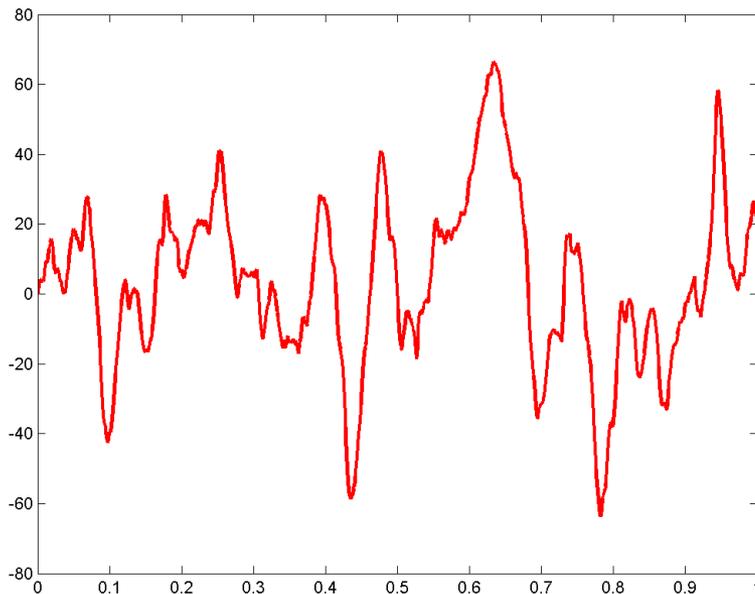
$$\sigma^2 = E[(X(n) - \mu)^2] \approx \frac{1}{N} \sum_{k=1}^N (X(k) - \mu)^2$$

$$\Gamma_X(k) \stackrel{!}{=} E[(X(i) - \mu)(X(i-k) - \mu)]$$

$$\stackrel{!}{=} \frac{1}{N-k} \sum_{i=k+1}^N (X(i) - \mu)(X(i-k) - \mu)$$

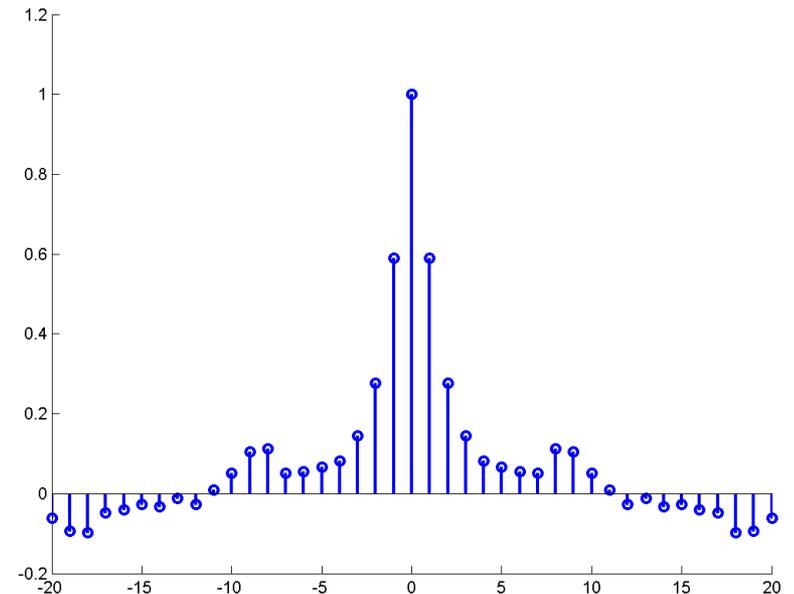
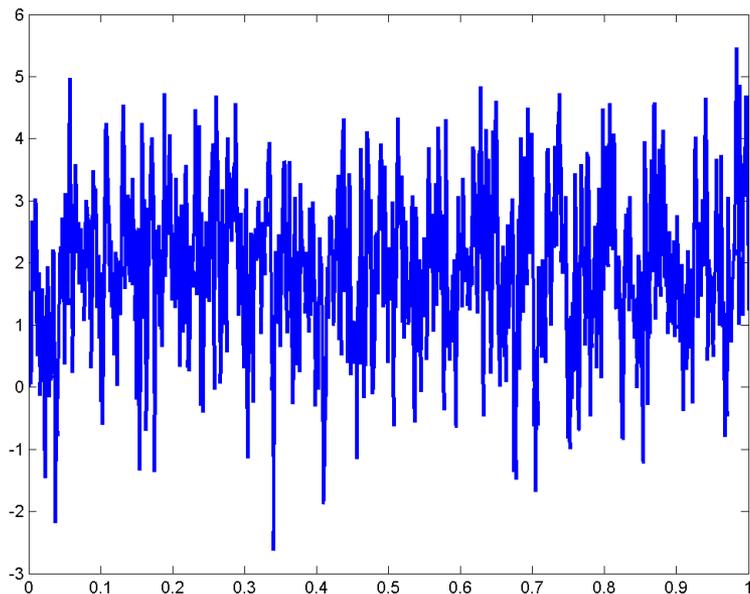
Exemple de calcul d'autocorrélation

```
>> N0=100; N=1000;  
>> X=zeros(N0+N,1);  
>> B=randn(N0+N,1);  
>> for k=3:N0+N,  
    X(k)=1.85*X(k-1)-0.855*X(k-2)+B(k);  
>> end,  
>> X=X(N0+1:N0+N)-X(N0+1);  
>> [gamma,lags]=xcov(X,20,'coeff');
```



Exemple de calcul d'autocorrélation

```
>> N0=100; N=1000;
>> X=zeros(N0+N,1);
>> B=randn(N0+N,1);
>> for k=3:N0+N,
        X(k)=0.63*X(k-1)-0.098*X(k-2)+B(k);
>> end,
>> X=X(N0+1:N0+N)-X(N0+1);
>> [gamma,lags]=xcov(X,20,'coeff');
```



Covariance et cross-corrélation

Pour deux processus stationnaires du 2nd ordre, on note pour tout entier relatif k

$$\Gamma_{X_1 X_2}(k) = E \left[\left(X_1(i) - \mu_1 \right) \left(X_2(i-k) - \mu_2 \right) \right]^{(*)}$$

et $\Gamma_{X_1 X_2}(k)$ s'appelle la **covariance** de X_1 et X_2 .

On définit aussi la **cross-corrélation** de X_1 et X_2 comme étant la normalisation de l'autocovariance par les écart-types de X_1 et de

$$\gamma_{X_1 X_2}(k) = \frac{\Gamma_{X_1 X_2}(k)}{\sigma_1 \sigma_2}$$

(*) On a noté $\mu_j = E[X_j(n)]$ et $\sigma_j^2 = E[(X_j(n) - \mu_j)^2]$, $j=1,2$

Covariance et cross-corrélation

Remarque. L' autocorrélation de X est la cross-corrélation de X avec lui-même.

Sous l'hypothèse d'ergodicité, on peut calculer la covariance de deux processus X_1 et X_2 après avoir calculé les moyennes empiriques μ_1 et μ_2 par

$$\Gamma_{X_1 X_2}(k) = E\left[\left(X_1(i) - \mu_1\right)\left(X_2(i-k) - \mu_2\right)\right]$$
$$= \frac{1}{N-k} \sum_{i=k+1}^N \left(X_1(i) - \mu_1\right)\left(X_2(i-k) - \mu_2\right)$$

Puis, σ_1 et σ_2 désignant les écart-types, on a toujours:

$$\gamma_{X_1 X_2}(k) = \frac{\Gamma_{X_1 X_2}(k)}{\sigma_1 \sigma_2}$$

Applications

- Géologie: Détecter la profondeur des couches géologiques (autocorrélation).
- GPS: calculer le délai de propagation entre deux signaux émis par deux satellites (cross-corrélation).
- Sonar: Détecter la distance d'une surface quelconque (autocorrélation).
- Economie: Repérer les corrélations entre plusieurs variables économiques (cross-corrélation).
- FBI: localiser un signal particulier dans un enregistrement plus long, ou dans un ensemble d'enregistrements (convolution).
- Réseau: estimer des retards de propagation (autocorrélation).

Exemple de calcul de cross-corrélation

```

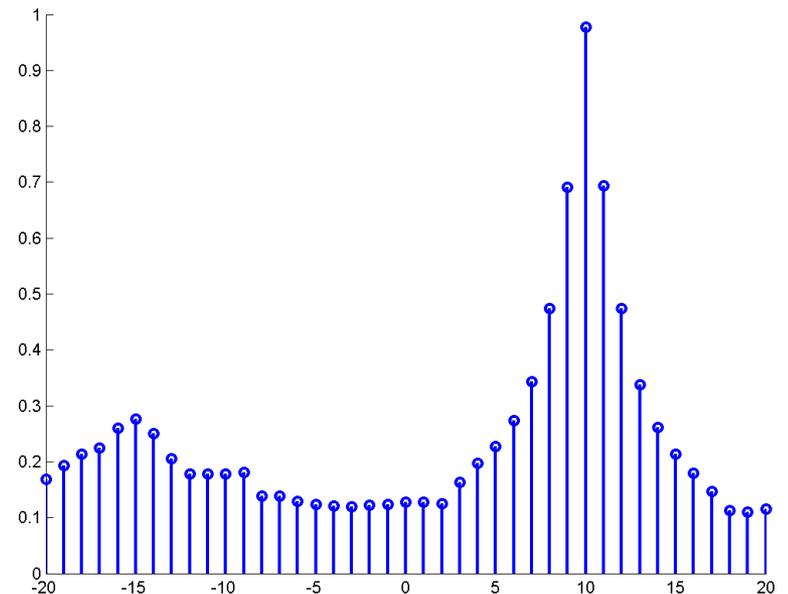
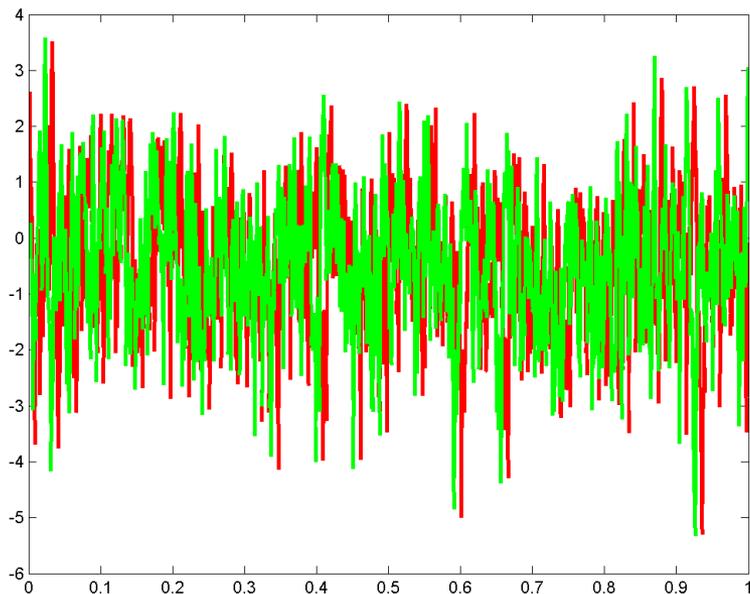
N0=130;
N=1000;
X=zeros(N0+N,1);
B=randn(N0+N,1);
a=0.3;b=0.4;
for k=3:N0+N,
    X(k)=(a+b)*X(k-1)-a*b*X(k-2)+B(k);
end,

```

```

Y=X(N0+1:N0+N)-X(N0+1);
X=X(N0-10+1:N0-10+N)-X(N0+1);
X=X+randn(size(X))*0.1;
Y=Y+randn(size(Y))*0.1;
[gamma,lags]=xcorr(X,Y,20,'coeff');

```



Filtre de Wiener

Le **filtre de Wiener** est un outil permettant d'estimer une variable aléatoire ou un processus aléatoire entaché de bruit, en supprimant autant que possible le bruit (on « filtre » le bruit).



Ce n'est pas nécessairement un filtre passe-bas: c'est une méthodologie permettant de synthétiser des filtres de tous ordres.

Vous avez tous un filtre de Wiener dans votre poche !

Filtre de Wiener: exemples

- Problèmes d'égalisation de canal (déconvolution):
 - $X(n) = S(n) + W(n)$ où $S(n)$ est le signal et $W(n)$ est le bruit.
On mesure $X(n)$ et on veut estimer $S(n)$.
 - $X(n) = S(n) - 2S(n-1) + S(n-2) + W(n)$
- Calcul de moyenne récursif:
 - $X(n) = \theta + W(n)$
- Régénération d'un signal binaire b :
 - $X(n) = b(n) + \alpha b(n-1) + W(n)$
- Annulation d'écho:
 - $X(n) = (h_0 * S(n)) + W(n)$

L'oreillette Bluetooth 1.1 Nokia **BH-602** appartient au milieu de gamme et offre 11 heures d'autonomie avec une qualité audio profitant de son DSP (Digital Signal Processing) d'annulation d'écho et de réduction de bruit, pour des conversations plus nettes.



Filtre de Wiener

Soit $X(n)$ un processus aléatoire quelconque que l'on suppose centré (sinon, on remplacerait $X(n)$ par $X(n)-\mu$).

On suppose que θ est une variable aléatoire inconnue mais que l'on veut estimer linéairement à partir des $p+1$ récentes valeurs de $X(n)$:

$$\hat{\theta} = a_0 X(n) + a_1 X(n-1) + \dots + a_p X(n-p)$$

On cherche les paramètres a_0, a_1, \dots, a_p .

On suppose pouvoir calculer

$$\Gamma_X(k) \text{ et } \Gamma_{X\theta}(k)$$

pour $k=0, \dots, p$, où on note

$$\Gamma_{\theta X}(k) = E[\theta X(n-k)]$$

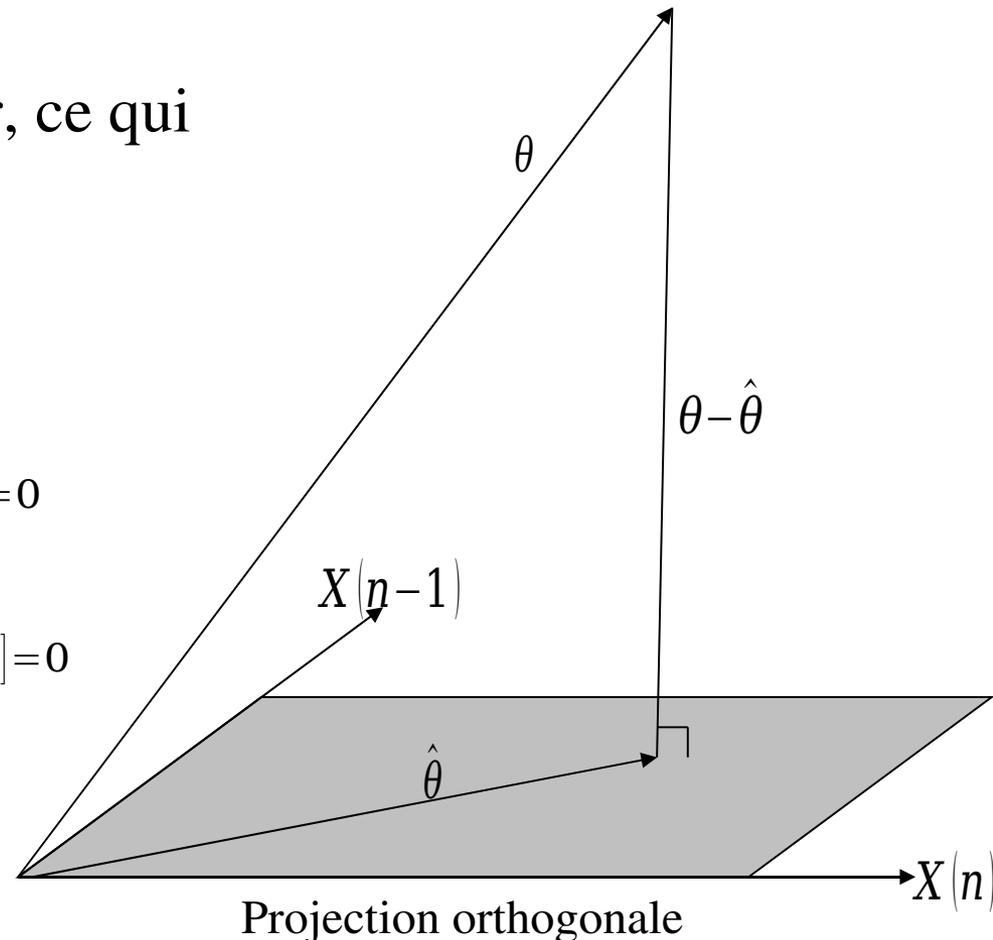
Filtre de Wiener

On écrit que $\theta - \hat{\theta}$ et $X(n-k)$ sont orthogonaux pour $k=0, \dots, p$

$$E[(\theta - \hat{\theta}) X(n-k)] = 0$$

Puis on remplace $\hat{\theta}$ par sa valeur, ce qui donne les équations

$$\begin{cases} E[\theta X(n)] - \sum_{i=0}^p a_i E[X(n)X(n-i)] = 0 \\ E[\theta X(n-1)] - \sum_{i=0}^p a_i E[X(n-1)X(n-i)] = 0 \\ \vdots \\ E[\theta X(n-p)] - \sum_{i=0}^p a_i E[X(n-p)X(n-i)] = 0 \end{cases}$$



Filtre de Wiener

En remplaçant par les covariances, le système s'écrit donc:

$$\begin{cases} \sum_{i=0}^p a_i \Gamma_X(i) = \Gamma_{\theta X}(0) \\ \sum_{i=0}^p a_i \Gamma_X(|i-1|) = \Gamma_{\theta X}(1) \\ \vdots \\ \sum_{i=0}^p a_i \Gamma_X(|i-p|) = \Gamma_{\theta X}(p) \end{cases}$$

Matriciellement, on obtient les équations de Wiener-Hopf

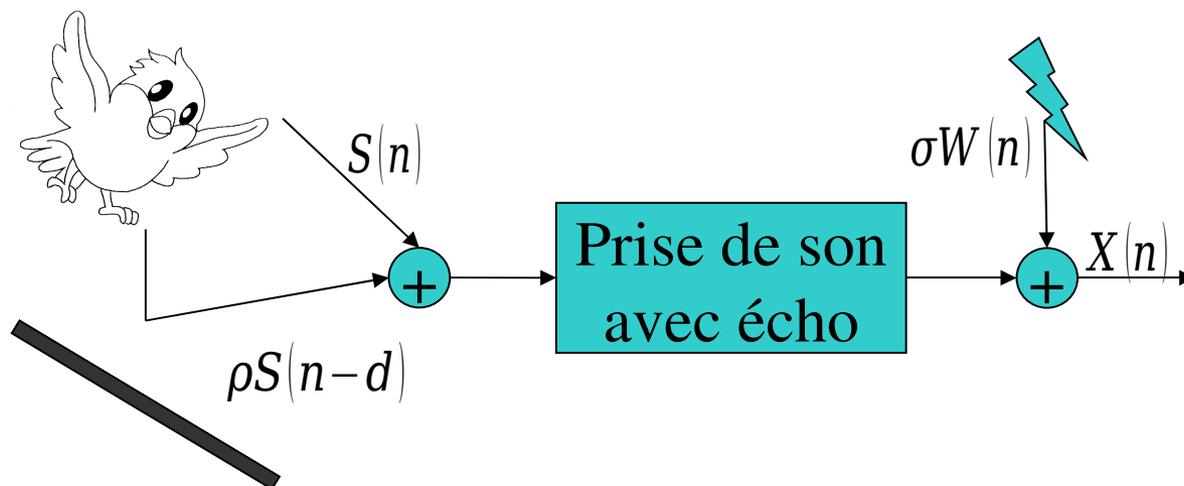
$$\begin{pmatrix} \Gamma_X(0) & \Gamma_X(1) & \cdots & \Gamma_X(p) \\ \Gamma_X(1) & \Gamma_X(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Gamma_X(1) \\ \Gamma_X(p) & \cdots & \Gamma_X(1) & \Gamma_X(0) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} \Gamma_{\theta X}(0) \\ \Gamma_{\theta X}(1) \\ \vdots \\ \Gamma_{\theta X}(p) \end{pmatrix}$$

(Matrice de Toëplitz)

Annulation d'écho

On mesure $X(n) = S(n) + \rho S(n-d) + \sigma W(n)$ où

- $S(n)$ est le signal « utile »
- $W(n)$ un bruit blanc
- ρ et σ sont des paramètres d'amplitude
- d est le retard du à l'écho que l'on veut annuler



Annulation d'écho

On suppose (car c'est plus simple) que S est centré, stationnaire et surtout que $E[S(n)^2]=s^2$ et $E[S(n)S(n-k)]=0$ si $k>0$.

On obtient par un calcul simple que l'on fera en TD :

$$\Gamma_{\theta X}(k) = E[S(n) X(n-k)] = \begin{cases} s^2 & \text{si } k=0 \\ \rho s^2 & \text{si } k=d \\ 0 & \text{sinon} \end{cases}$$

$$\Gamma_X(k) = E[X(n) X(n-k)] = \begin{cases} s^2(1+\rho^2)+\sigma^2 & \text{si } k=0 \\ \rho s^2 & \text{si } k=d \\ 0 & \text{sinon} \end{cases}$$

Annulation d'écho

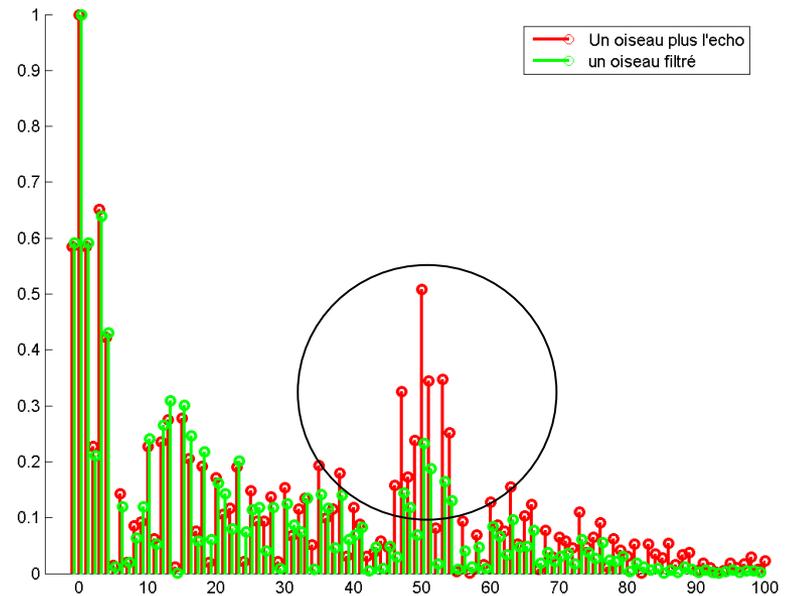
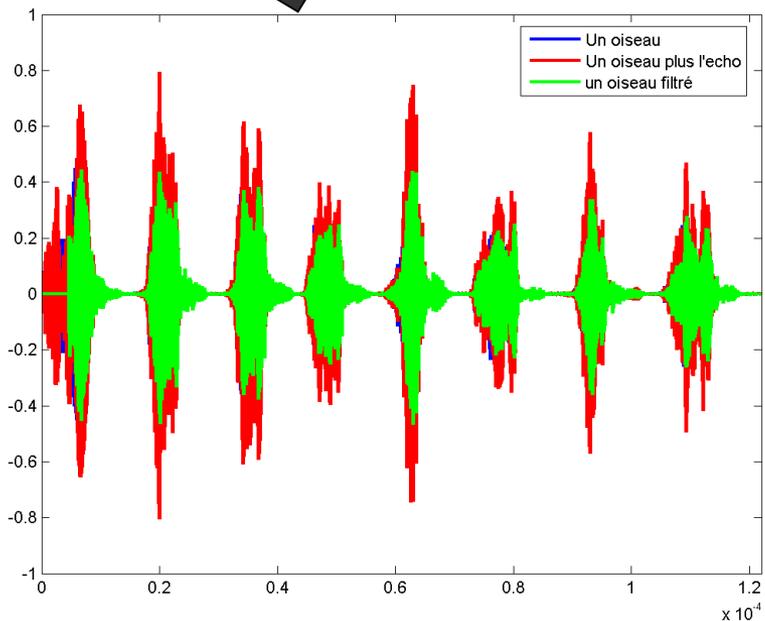
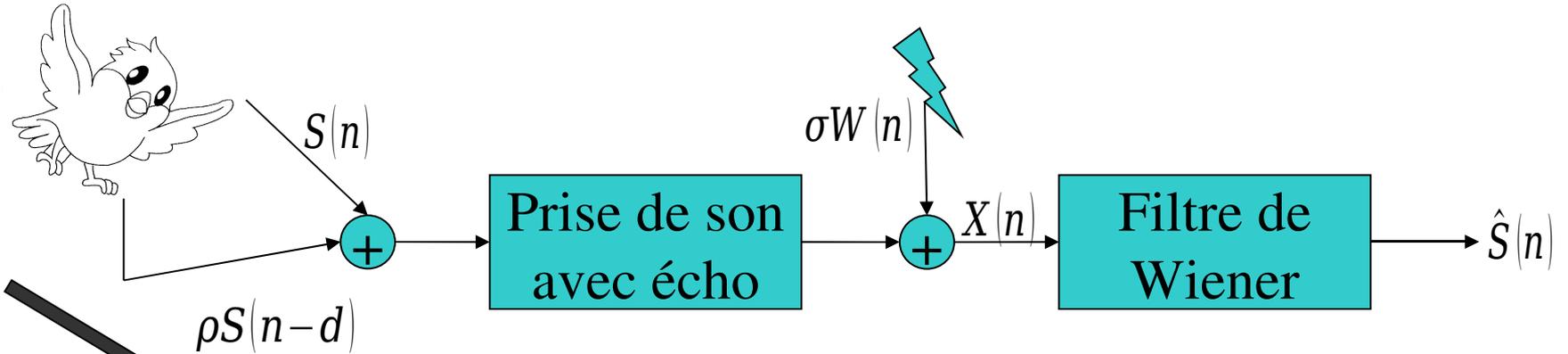
ce qui permet de poser le système de Wiener-Hopf et de le résoudre (numériquement):

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} \Gamma_X(0) & \Gamma_X(1) & \cdots & \Gamma_X(p) \\ \Gamma_X(1) & \Gamma_X(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Gamma_X(1) \\ \Gamma_X(p) & \cdots & \Gamma_X(1) & \Gamma_X(0) \end{pmatrix}^{-1} \begin{pmatrix} \Gamma_{\theta X}(0) \\ \Gamma_{\theta X}(1) \\ \vdots \\ \Gamma_{\theta X}(p) \end{pmatrix}$$

et on appliquera le filtre linéaire

$$\hat{S}(n) = a_0 X(n) + a_1 X(n-1) + \cdots + a_p X(n-p)$$

Annulation d'écho



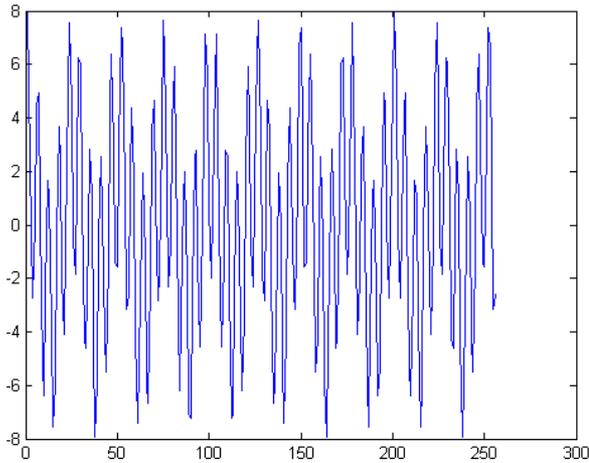
Rappels sur la DFT (Discrete Fourier Transform)

La transformée de Fourier discrète est l'analogie pour les signaux **numériques** de la transformée de Fourier continue utilisée pour les signaux **analogiques** (et dont on ne parlera donc pas dans ce cours).

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) \exp\left(-2i\pi k \frac{n}{N}\right)$$
$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}(k) \exp\left(2i\pi n \frac{k}{N}\right)$$

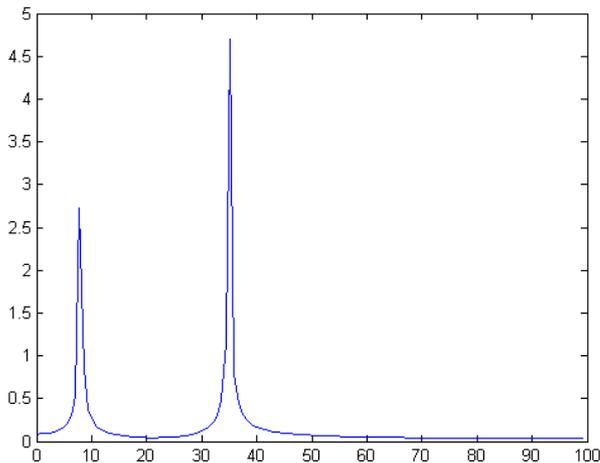
La TFD est une méthode d'analyse dans le domaine fréquentiel très puissante. Elle est aussi à la base de la compression de signaux (audio et vidéo).

Un exemple en Matlab



```
N=256;  
Fe=200;  
omega1=8;  
omega2=35;
```

```
t=(0:N-1)/Fe;  
x=3*cos(2*pi*omega1*t)  
+5*cos(2*pi*omega2*t);
```



```
figure(1),  
plot(x);
```

```
y=fft(x);  
f=(0:N-1)*Fe/N;
```

```
figure(2),  
plot(f(1:N/2),
```

Résolution

La TFD d'un signal réel est caractérisée par le nombre de points du signal N et la fréquence d'échantillonnage F_e . La plage de fréquence détectable vaut $F_e/2$ et la finesse de résolution vaut F_e/N

$$x(n) = 3 \cos(2\pi f_1 t) + \cos(2\pi f_2 t) + \cos(2\pi f_3 t) + 5 \cos(2\pi f_4 t)$$

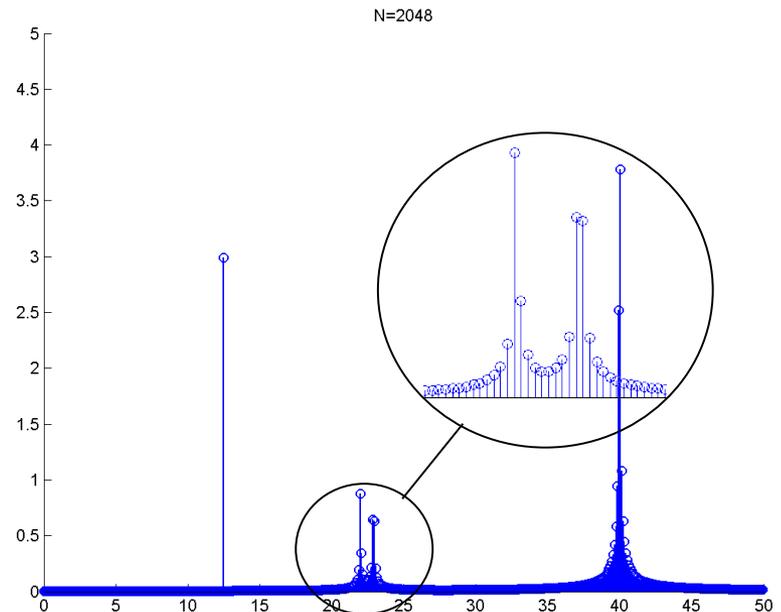
où $f_1 = 12,5$

$f_2 = 22$

$f_3 = 22.9$

$f_4 = 40$

f_1 est de la forme $f_1 = k \frac{F_e}{N}$
 et $f_2 \approx f_3$



Extension: la DTFT

La transformée de Fourier en temps discret (Discrete time Fourier Transform) est un outil plus théorique

$$\tilde{x}(\omega) = \sum_{n=-\infty}^{+\infty} x(n) \exp(-i\omega n)$$

où $\omega \in [-\pi, \pi[$.

On a la formule d'inversion

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \tilde{x}(\omega) \exp(i\omega n) d\omega$$

Puisque N est infini, la finesse de résolution est nulle et les fréquences sont toutes distinguables.

Algorithme FFT

La transformée de Fourier rapide (Fast Fourier Transform, FFT) est un algorithme de calcul de la transformée de Fourier discrète. Cet algorithme a permis de rendre possible le calcul de TFD dans des applications embarquées, et on désigne souvent la TFD par FFT.

Exemple numérique. Un signal audio de 1 seconde échantillonné à 40KHz nécessite 40000 valeurs numériques. Le calcul de sa FFT prend un temps de l'ordre de 400 ms (*)
La TFD classique prend un temps de l'ordre de 26 minutes !

(*) à raison d'une opération pas milli-seconde.

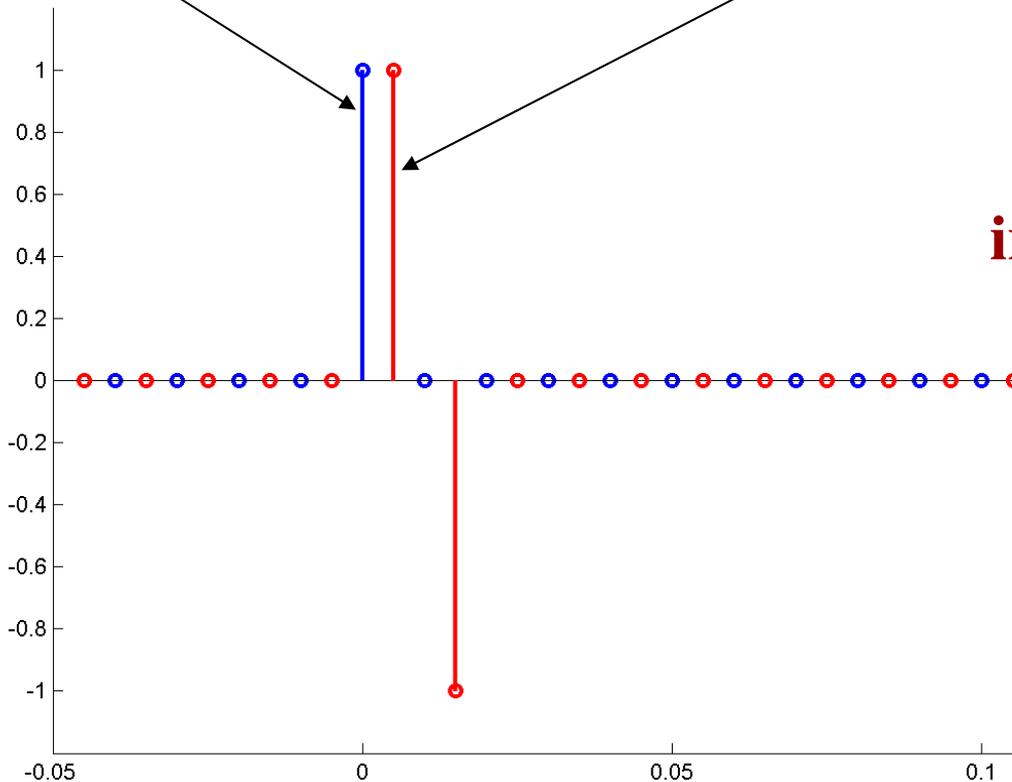
Réponse impulsionnelle

$$u(n) = \begin{cases} 1 & \text{si } n=0 \\ 0 & \text{sinon} \end{cases}$$

Filtre dérivateur

$$y(n) = u(n) - u(n-1]$$

Filtre à réponse impulsionnelle finie (FIR)



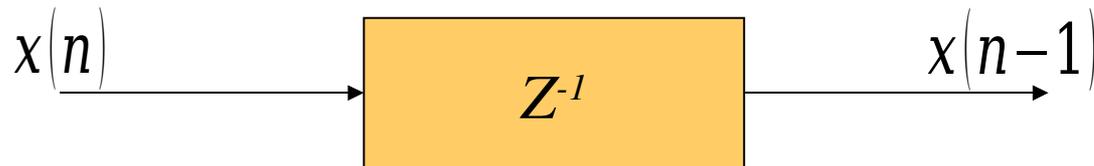
NB: On a représenté la sortie y décalée d'une demi-période dans le temps, pour des raisons de représentation graphique.

Rappel sur la transformée en Z

Transformée en Z: $X(Z) = \sum_{n=-\infty}^{+\infty} x(n)Z^{-n}$

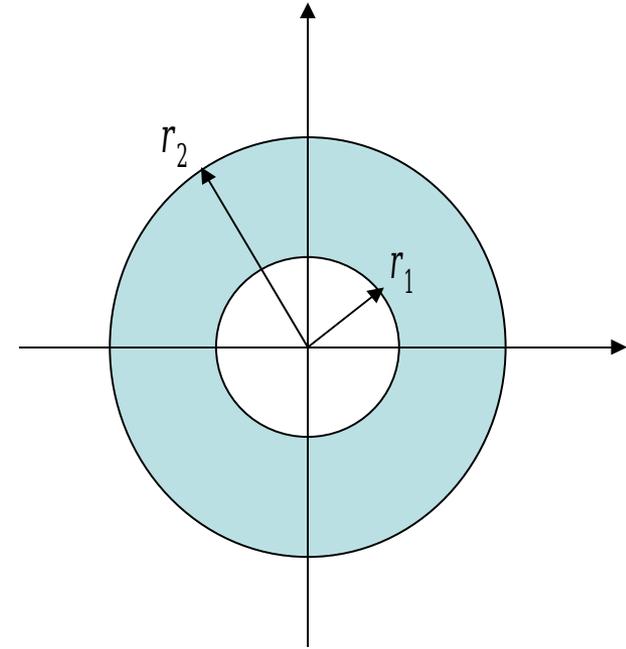
Z^{-1} est l'opérateur retard:

$$Z^{-1} X(Z) = \sum_{n=-\infty}^{+\infty} x(n)Z^{-(n+1)} = \sum_{n=-\infty}^{+\infty} x(n-1)Z^{-n}$$



Rappel sur la transformée en Z

La transformée en Z converge à l'intérieur d'un anneau délimité par deux cercles de rayons r_1 et r_2 définis à partir des rayons de convergence des séries:



$$\sum_{n=0}^{+\infty} x(n)Z^{-n} \text{ converge} \Leftrightarrow \frac{1}{Z} < \rho_1 \Leftrightarrow Z > \frac{1}{\rho_1} \text{ donc } r_1 = \frac{1}{\rho_1}$$

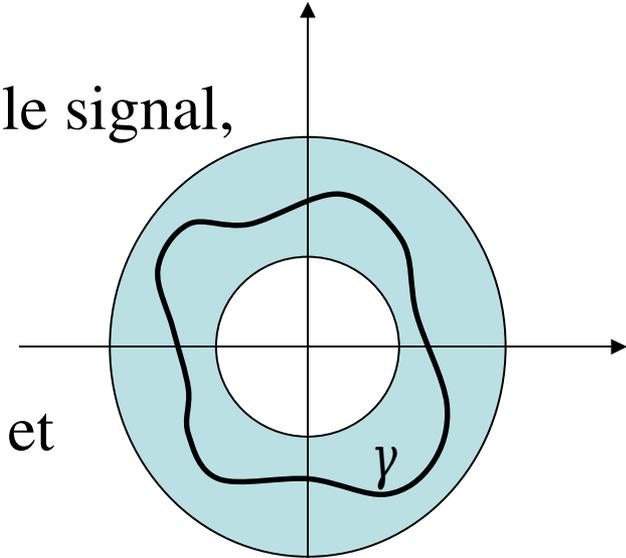
$$\sum_{n=0}^{-\infty} x(n)Z^{-n} \text{ converge} \Leftrightarrow \sum_{n=0}^{-\infty} x(-n)Z^n \text{ converge donc } r_2 = \rho_2$$

Rappel sur la transformée en Z

La transformée en Z caractérise complètement le signal, puisqu'on a la formule d'inversion

$$x(n) = \frac{1}{2i\pi} \int_{\gamma} Z^{n-1} X(Z) dZ$$

où γ est une courbe fermée entourant l'origine et incluse dans le disque unité.



Remarque. Si l'anneau de convergence contient le disque unité

$$x(n) = \frac{1}{2\pi} \int_0^{2\pi} \exp(in\omega) X(\exp(i\omega)) d\omega$$

en faisant le changement de variable $Z = \exp(i\omega)$

On retrouve aussi la DTFT: $X(\exp(i\omega)) = \sum_{n=-\infty}^{+\infty} x(n) \exp(-i\omega n) = \tilde{x}(\omega)$

Théorème des résidus (utile pour certains calculs)

Théorème. Notons a_1, \dots, a_n les n pôles de la fonction rationnelle f , et soit γ un lacet ne passant par aucun pôle. Alors

$$\int_{\gamma} f(z) dz = 2i\pi \sum_{k=1}^n \text{ind}_{\gamma}(a_k) \text{Res}(f, a_k)$$

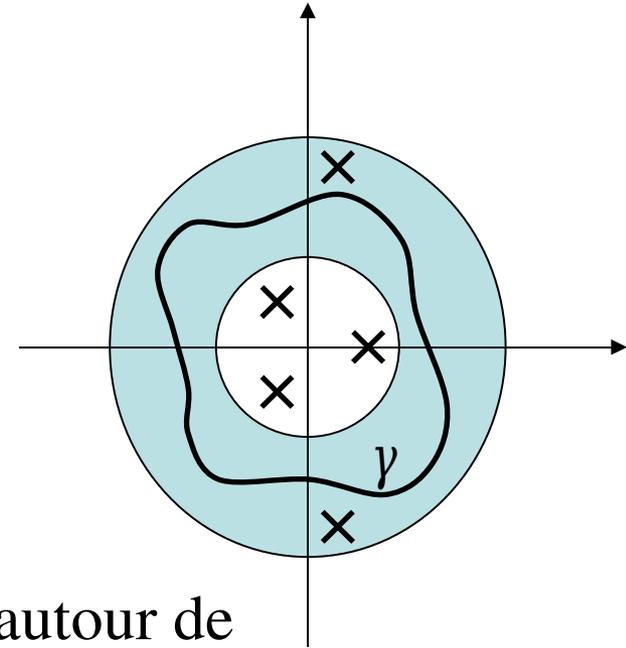
où

$\text{ind}_{\gamma}(a)$ est le nombre de tours que fait γ autour de

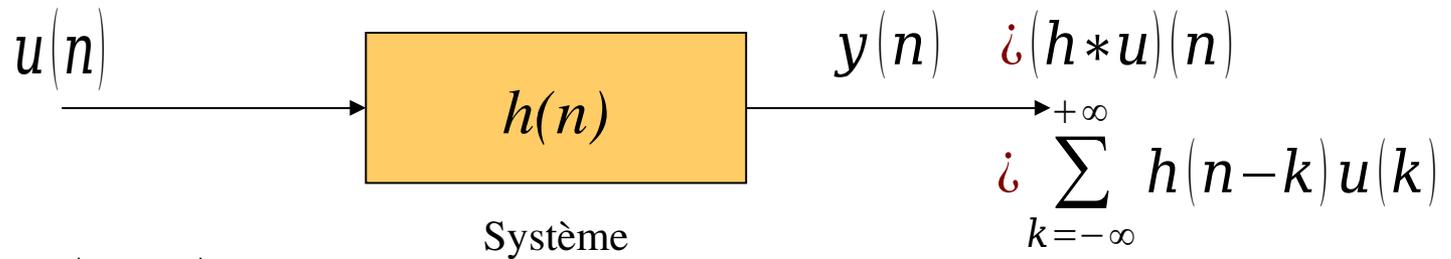
$$\text{Res}(f, a) \stackrel{\text{def}}{=} \lim_{z \rightarrow a} \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} \left((z-a)^m f(z) \right)$$

$$\stackrel{\text{def}}{=} \lim_{z \rightarrow a} (z-a) f(z) = \frac{p(a)}{q'(a)}$$

en notant m l'ordre du pôle, et en écrivant $f(z) = \frac{p(z)}{q(z)}$



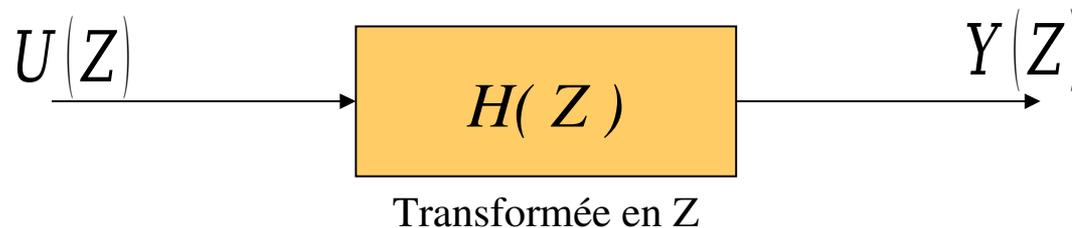
Réponse impulsionnelle: Définition



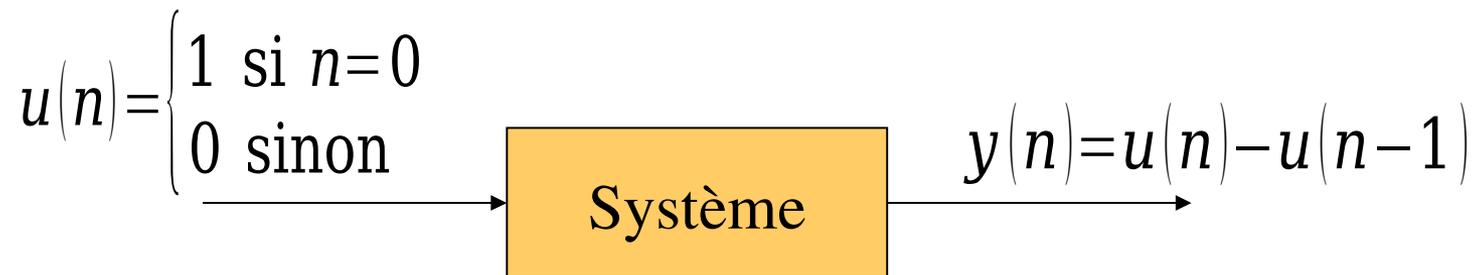
$$Y(Z) = \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} h(n-k)u(k)Z^{-n}$$

$$= \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} h(n-k)u(k)Z^{-(n-k)}Z^{-k}$$

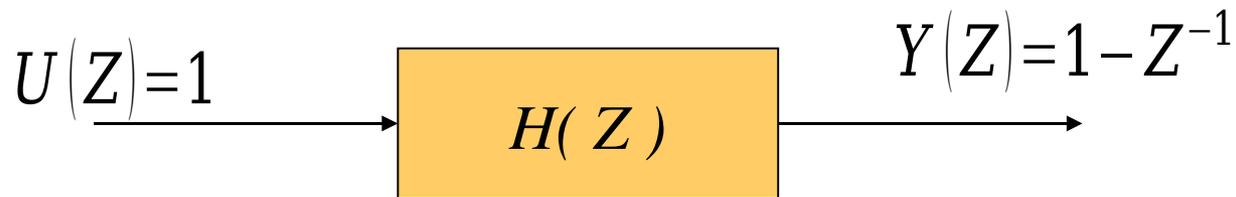
$$= \sum_{n=-\infty}^{+\infty} h(m)Z^{-m} \sum_{k=-\infty}^{+\infty} u(k)Z^{-k} = \boxed{H(Z)} U(Z)$$



Réponse impulsionnelle: Identification



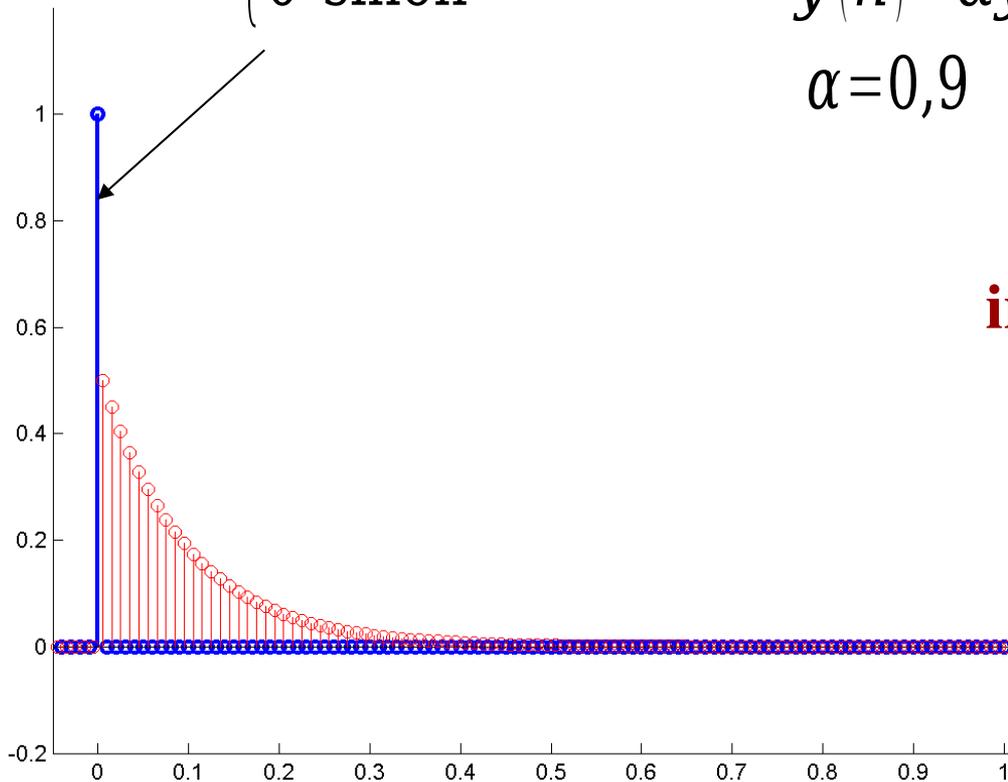
Transformée en Z



$$H(Z) = \frac{Y(Z)}{U(Z)} = Y(Z)$$

Réponse impulsionnelle

$$u(n) = \begin{cases} 1 & \text{si } n=0 \\ 0 & \text{sinon} \end{cases}$$



Système du 1^{er} ordre

$$y(n) = \alpha y(n-1) + (1-\alpha) u(n)$$

$$\alpha = 0,9$$

**Filtre à réponse
impulsionnelle infinie (IIR)**

NB: La sortie $y(n)$, en rouge, a été multipliée par un facteur 5 pour des raisons graphiques.

Réponse impulsionnelle: 1^{ère} méthode de calcul

1) Equation sans second membre (ssm)

$$y(n) = \alpha y(n-1)$$

Solution de la forme $y(n) = ar^n$

$$ar^n = \alpha ar^{n-1} \Leftrightarrow r = \alpha \quad (\text{équation caractéristique})$$

2) Réponse à l'entrée $u(0)=1$

$$y(0) = a \stackrel{?}{=} \alpha y(-1) + (1-\alpha)u(0)$$

$$\stackrel{?}{=} 1 - \alpha$$

d'où
$$y(n) = \begin{cases} (1-\alpha)\alpha^n & \text{si } n \geq 0 \\ 0 & \text{sinon} \end{cases}$$

donc
$$H(Z) = Y(Z) = \sum_{n=0}^{+\infty} (1-\alpha)\alpha^n Z^{-n} = \frac{1-\alpha}{1-\alpha Z^{-1}}$$

Réponse impulsionnelle: 2^{nde} méthode de calcul

On utilise la transformée en Z

$$\begin{array}{c} y(n) = \alpha y(n-1) + (1-\alpha)u(n) \\ \updownarrow \qquad \qquad \updownarrow \qquad \qquad \updownarrow \\ Y(Z) = \alpha Z^{-1} Y(Z) + (1-\alpha)U(Z) \end{array}$$

puis on raisonne « formellement »

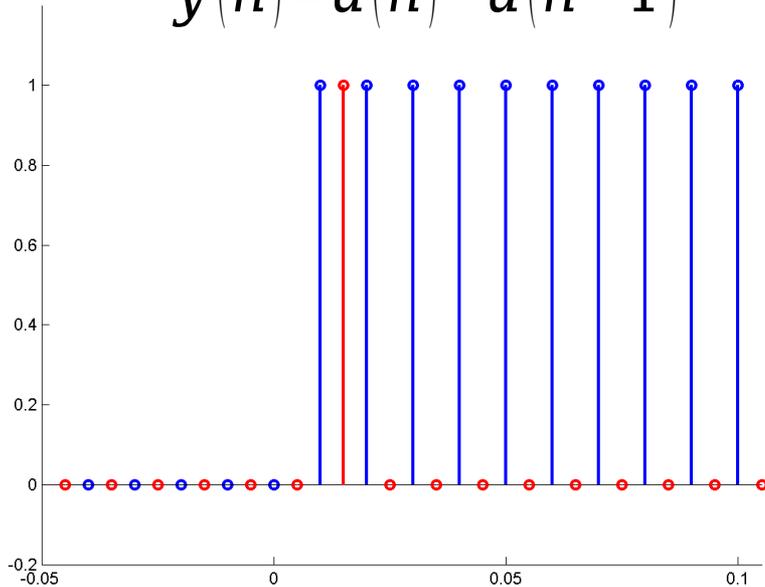
$$\Leftrightarrow (1 - \alpha Z^{-1}) Y(Z) = (1 - \alpha) U(Z)$$

$$\Leftrightarrow \frac{Y(Z)}{U(Z)} = \frac{1 - \alpha}{1 - \alpha Z^{-1}} = H(Z)$$

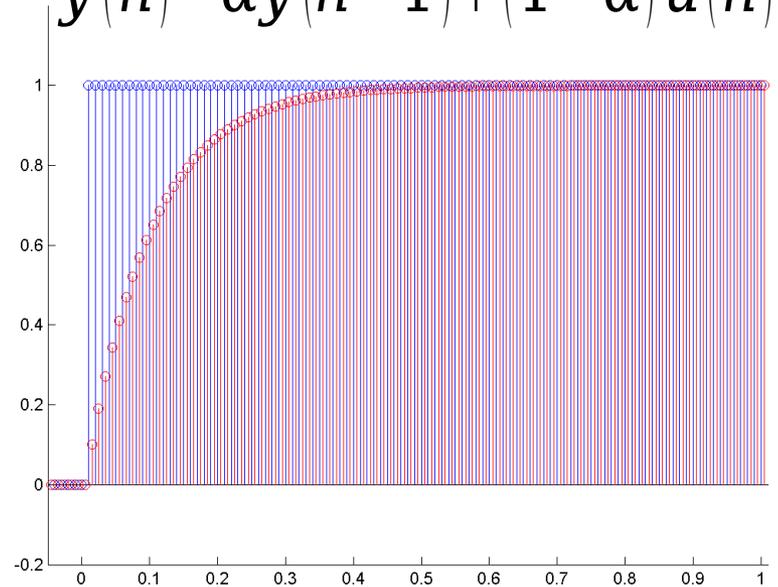
Réponse à un échelon

C'est la méthode employée pour étudier la nature et identifier la réponse d'un système physique.

$$y(n) = u(n) - u(n-1]$$



$$y(n) = \alpha y(n-1) + (1-\alpha)u(n]$$



Etude de la réponse du second système

$$y(n) \text{ 坵 } = \alpha y(n-1) + (1-\alpha)$$

$$\text{坵} = \alpha^2 y(n-2) + \alpha(1-\alpha) + (1-\alpha)$$

$$\text{坵} = \alpha^n y(0) + \alpha^{n-1}(1-\alpha) + \dots + \alpha(1-\alpha) + (1-\alpha)$$

$$\text{坵} = \alpha^n (1-\alpha) + \alpha^{n-1}(1-\alpha) + \dots + \alpha(1-\alpha) + (1-\alpha)$$

$$\text{坵} = (1-\alpha) \sum_{k=0}^n \alpha^k = (1-\alpha) \frac{1-\alpha^{n+1}}{1-\alpha} = 1-\alpha^{n+1}$$

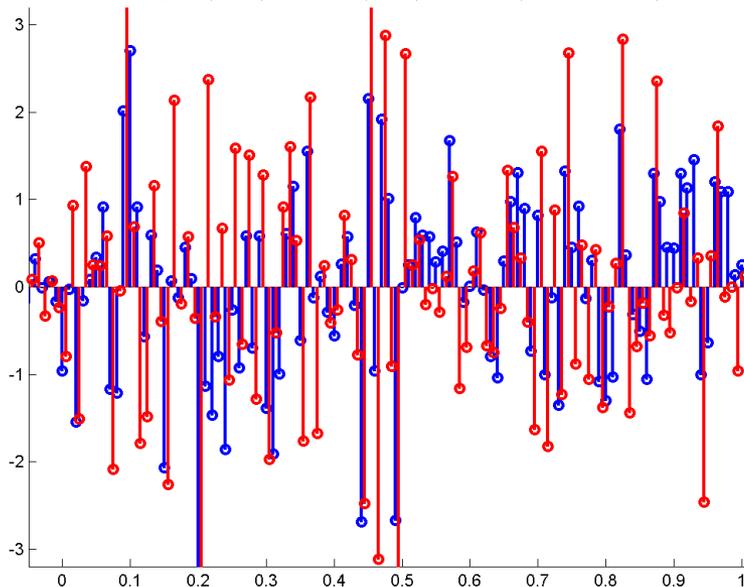
d'où la forme exponentielle de la réponse, caractéristique d'un 1^{er} ordre.

Réponse à un bruit blanc

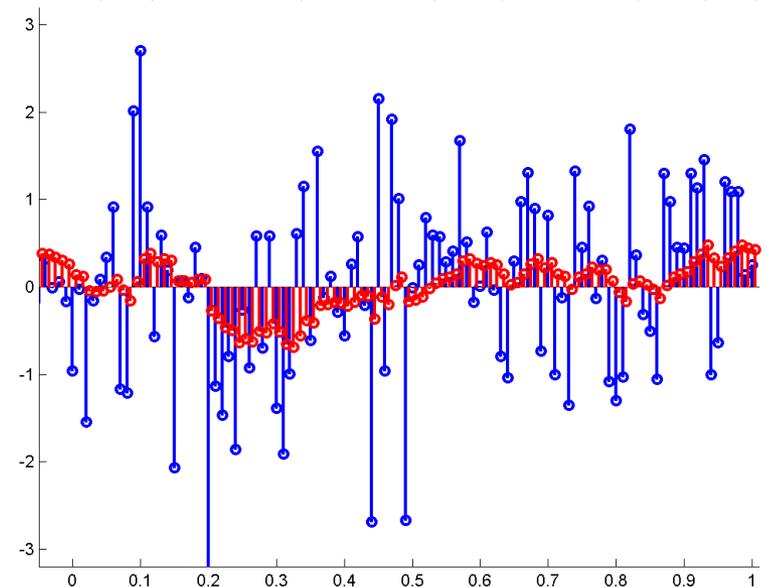
C'est une méthode employée pour identifier avec précision et robustesse un système en utilisant un logiciel d'identification (tel que Matlab).

Noter le comportement du système du 1^{er} ordre: c'est un filtre passe-bas.

$$y(n) = u(n) - u(n-1]$$

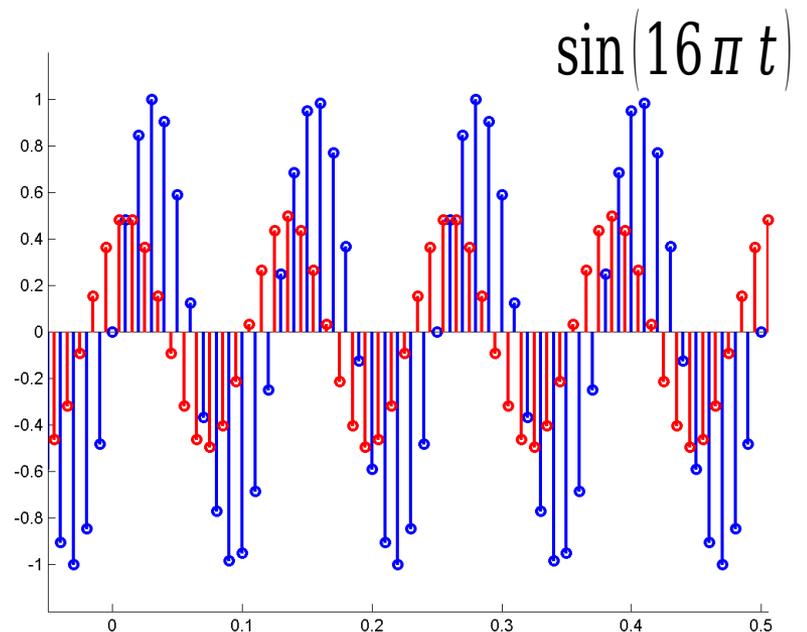
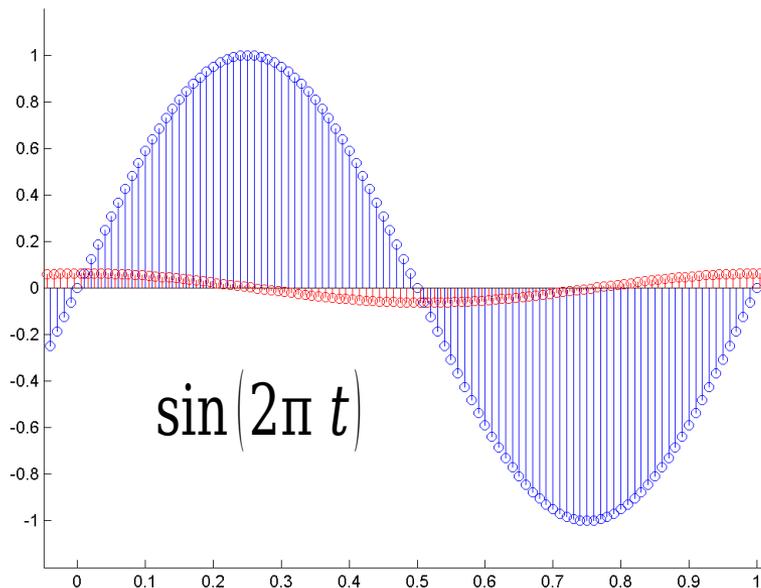


$$y(n) = \alpha y(n-1) + (1-\alpha)u(n]$$



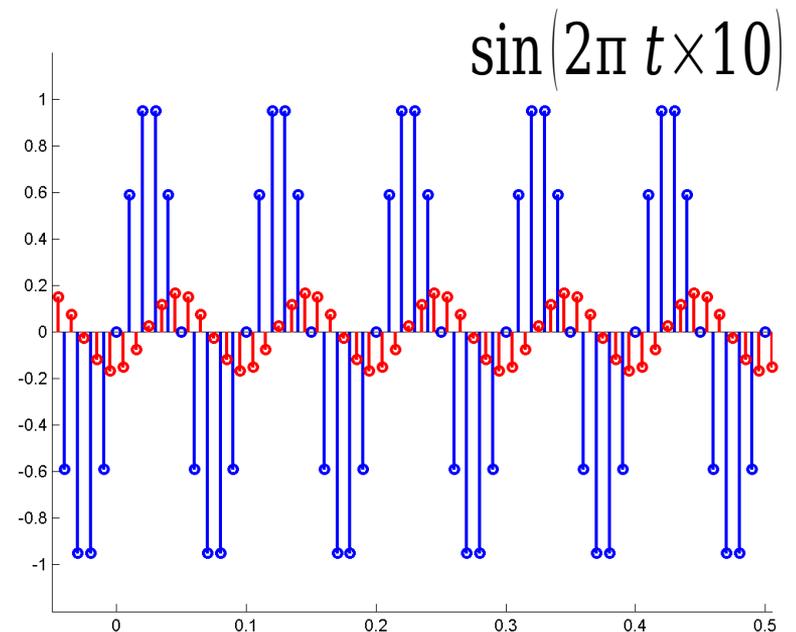
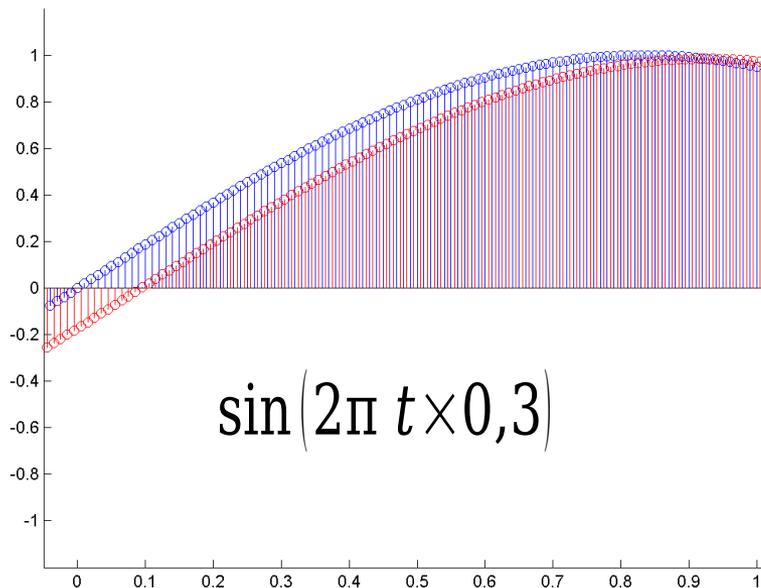
Réponse à un signal périodique: le dérivateur

$$y(n) = u(n) - u(n-1]$$

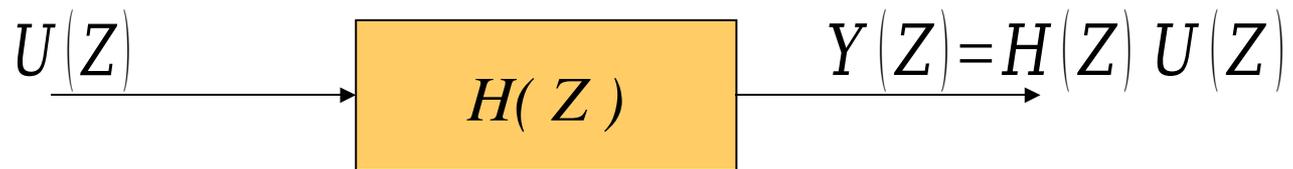


Réponse à un signal périodique: le passe-bas

$$y(n) = \alpha y(n-1) + (1-\alpha) u(n) \quad (\alpha = 0,9)$$



Réponse à un signal périodique



Posons $Z = \exp(i\omega)$

$$\begin{array}{ccc} Y(\exp(i\omega)) & = & H(\exp(i\omega))U(\exp(i\omega)) \\ \parallel & & \parallel \\ \tilde{y}(\omega) & & \tilde{u}(\omega) \end{array}$$

Cette équation relie le spectre de l'entrée à celui de la sortie. Le module et l'argument de $H(\exp(i\omega))$ montrent le comportement en fréquence du système, en fonction d'une fréquence normalisée.

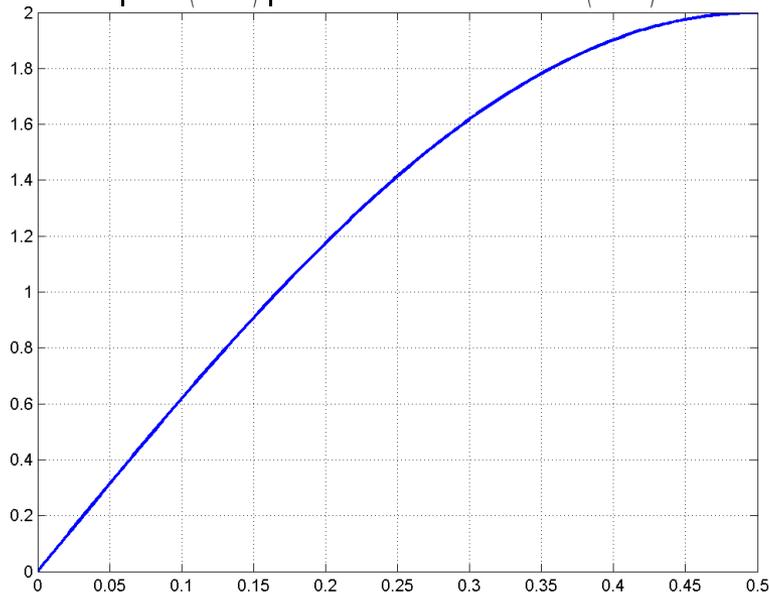
Reprenons nos deux exemples...

Réponse à un signal périodique: le dérivateur

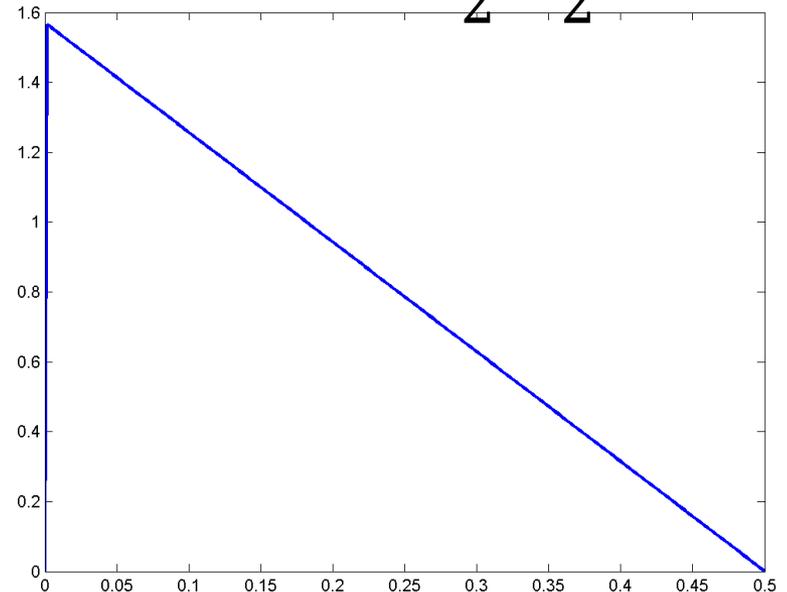
$$y(n) = u(n) - u(n-1) \quad \longrightarrow \quad H(Z) = 1 - Z^{-1}$$

$$F(\omega) = H(\exp(i\omega)) = 1 - \exp(i\omega) = 1 - \cos(\omega) - i\sin(\omega)$$

$$|F(\omega)| = \sqrt{2 - 2\cos(\omega)}$$

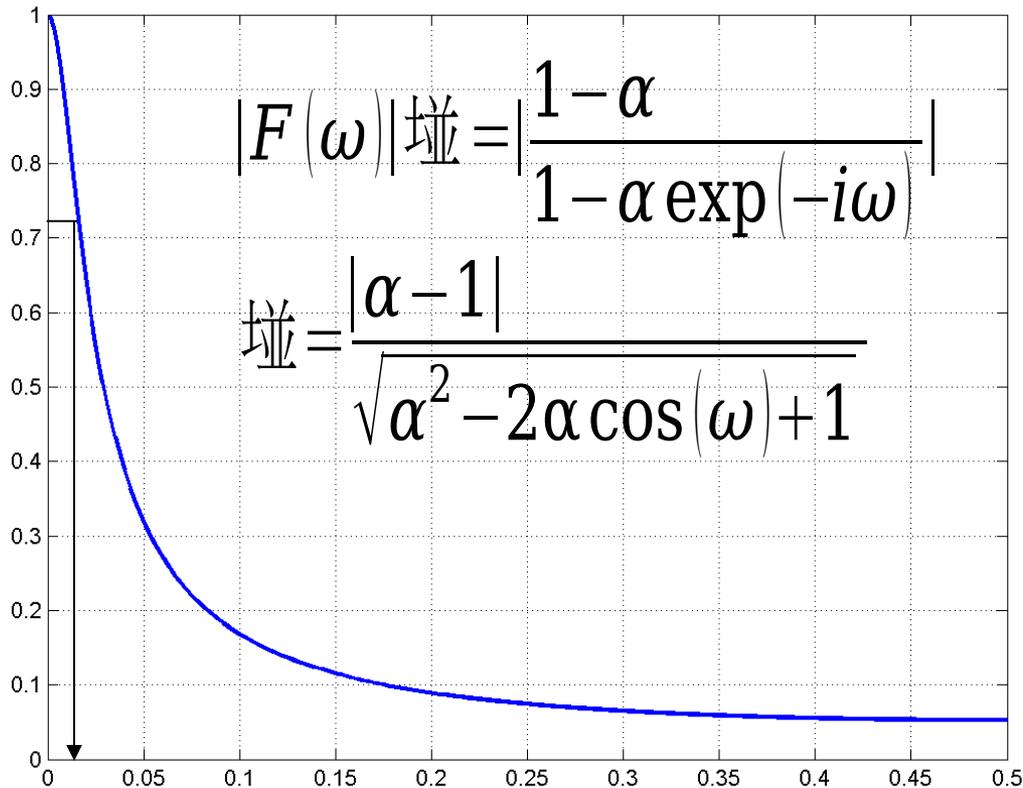


$$\arg(F(\omega)) = \frac{\pi}{2} - \frac{1}{2}\omega$$



Réponse à un signal périodique: le passe-bas

$$y(n) = \alpha y(n-1) + (1-\alpha)u(n) \quad (\alpha = 0,97)$$



ω est une fréquence normalisée: $\omega = 2\pi \frac{f}{F_e}$

La fréquence de coupure est l'atténuation à 3 dB:

$$|F(\omega_c)| = \frac{1}{\sqrt{2}} \max_{0 \leq \omega \leq \pi} (|F(\omega)|)$$

Ici, cela donne

$$\omega_c \approx 0,02$$

$$f_c \approx 0,32 \text{ Hz}$$

$$(F_e = 100 \text{ Hz})$$

Synthèse de filtre

FIR
$$H(Z) = \sum_{n=N_1}^{N_2} h(n) Z^{-n}$$

- Toujours **stable** (*i.e.* une entrée bornée génère une sortie bornée)
- **Causal** si $N_1=0$ donc **réalisable**.

IIR

- Stabilité à étudier
- Réalisable s'il est rationnel *i.e.*

$$H(Z) = \frac{P(Z)}{Q(Z)} = \frac{\sum_{n=0}^p a_n Z^{-n}}{\sum_{n=0}^q b_n Z^{-n}}$$

où P et Q sont des polynômes en Z^{-1} .

Synthèse de filtre

Si un filtre IIR a une fonction de transfert rationnelle $H(Z) = \frac{P(Z)}{Q(Z)}$
alors on appelle

- zéros de H les racines (complexes) du polynôme P
- pôles de H les racines du polynôme Q

Ce type de filtre est stable si les pôles de H sont inclus dans le disque unité.

Exemple: le système du 1^{er} ordre $y(n) = \alpha y(n-1) + (1-\alpha)u(n)$
a pour fonction de transfert

$$H(Z) = \frac{1-\alpha}{1-\alpha Z^{-1}}$$

Il est stable ssi $|\alpha| < 1$

Synthèse de filtre à partir de filtres analogiques

Partir de la transformée de Laplace d'un filtre analogique:



Signal dans le domaine temporel $y(t) = \int_{-\infty}^{+\infty} h(t-\tau)u(\tau)d\tau$

Transformée de Laplace (s complexe) $X(s) = \int_0^{+\infty} x(t)\exp(-st)dt$

C'est l'équivalent (point de vue utilisation) de la transformée en Z

Synthèse de filtre à partir de filtres analogiques

Soit la fonction de transfert d'un filtre passe-bas analogique de pulsation de coupure $\omega_0 = 2\pi f_0$

$$H(s) = \frac{1}{1 + \frac{s}{\omega_0}}$$

La relation fondamentale entre analogique et numérique est

$$Z = \exp(s T_e)$$

On ne peut l'inverser telle quelle et obtenir un filtre rationnel !

On utilisera des approximations:

$$\exp(s T_e) \approx 1 + s T_e + \frac{s^2 T_e^2}{2} + o(s^2) \quad (2^{\text{nd}} \text{ ordre})$$

$$\approx 1 + s T_e + o(s) \quad (1^{\text{er}} \text{ ordre})$$

Synthèse de filtre à partir de filtres analogiques

Au 1^{er} ordre:

$$Z = 1 + s T_e \Leftrightarrow s = \frac{1}{T_e} \frac{1 - Z^{-1}}{Z^{-1}}$$

que l'on substitue dans la fonction de transfert analogique:

$$H(Z) = \frac{\omega_0 T_e Z^{-1}}{1 + (\omega_0 T_e - 1) Z^{-1}}$$

A.N. $F_e = 48$ kHz, $f_0 = 1$ kHz

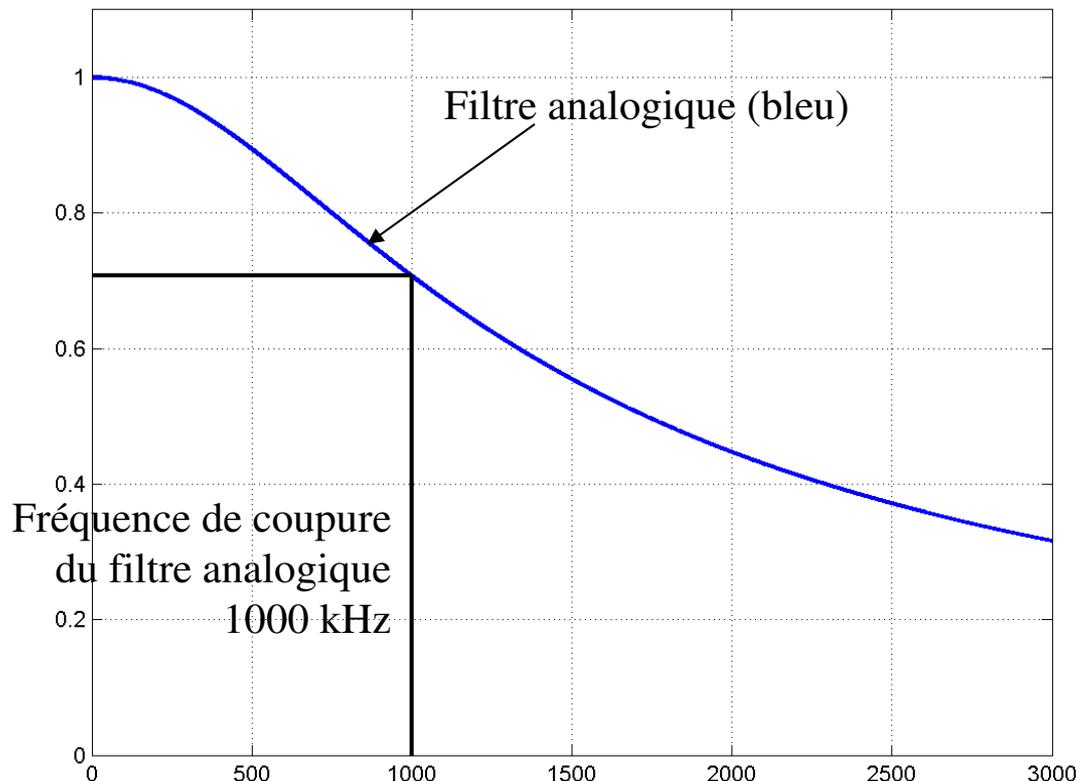
$$H(Z) = \frac{0,131 Z^{-1}}{1 + 0,869 Z^{-1}}$$

$$y(n) = 0,131 u(n-1) + 0,869 y(n-1)$$

(on retrouve un passe-bas numérique classique, retardé, avec $\alpha = 0,131$)

Synthèse de filtre à partir de filtres analogiques

$$H(s) = \frac{\omega_0}{\omega_0 + s} \Rightarrow F(\omega) = |H(i\omega)| = \left| \frac{\omega_0}{\omega_0 + i\omega} \right| = \frac{\omega_0}{\sqrt{\omega^2 + \omega_0^2}}$$

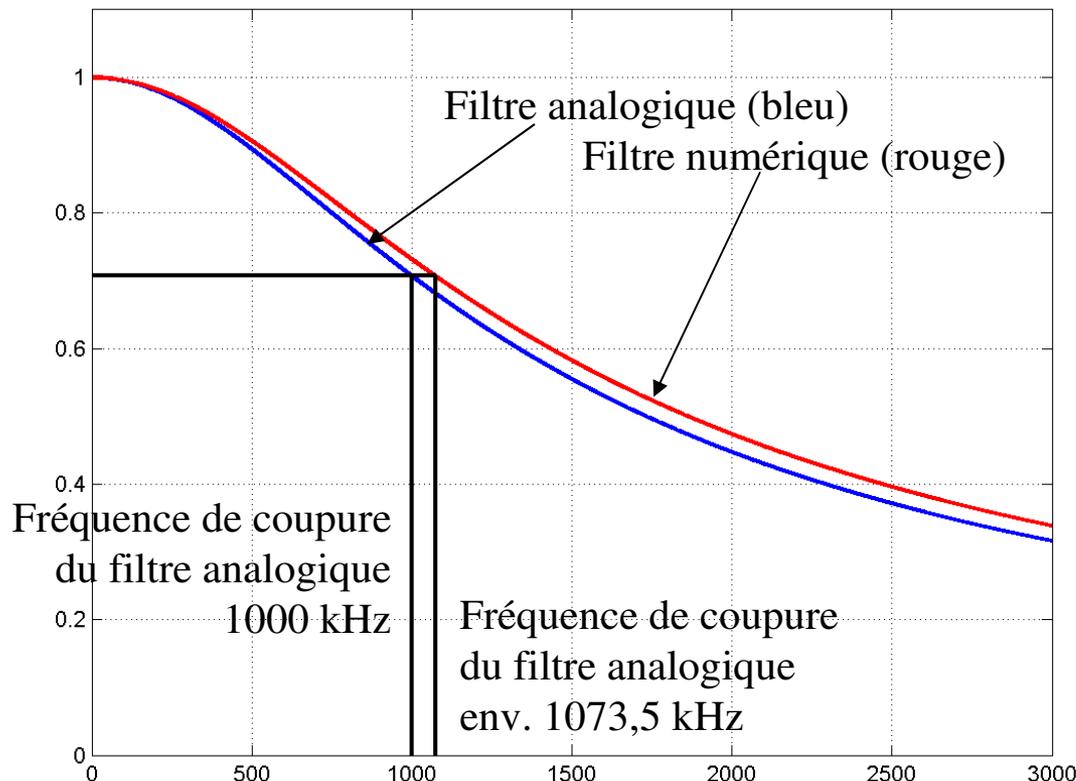


```
f0=1000;
w0=2*pi*f0;
Fe=48000;
Te=1/Fe;
f=0:0.1:5000;
w=2*pi*f;
i=sqrt(-1);
Fa=abs( 1./(1+i*w/w0) );
W0=w0*Te;
w=w/Fe;
```

```
h=plot(f,Fa,'b');
```

Synthèse de filtre à partir de filtres analogiques

$$H(Z) = \frac{OMEGA_0 Z^{-1}}{1 + (OMEGA_0 - 1) Z^{-1}} \Rightarrow F(\omega) = \left| \frac{OMEGA_0 \exp(-i\omega)}{1 + (OMEGA_0 - 1) \exp(-i\omega)} \right| \text{ où } OMEGA_0 = \omega_0 T_e$$



```
f0=1000;
w0=2*pi*f0;
Fe=48000;
Te=1/Fe;
f=0:0.1:5000;
w=2*pi*f;
i=sqrt(-1);
Fa=abs( 1./(1+i*w/w0) );
W0=w0*Te;
w=w/Fe;
Fn=abs( W0*exp(-i*w) ./
        (1+(W0-1)*exp(-
        i*w) ) );
h=plot(f,Fa,'b',f,Fn,'r');
```

Synthèse de filtre à partir de filtres analogiques

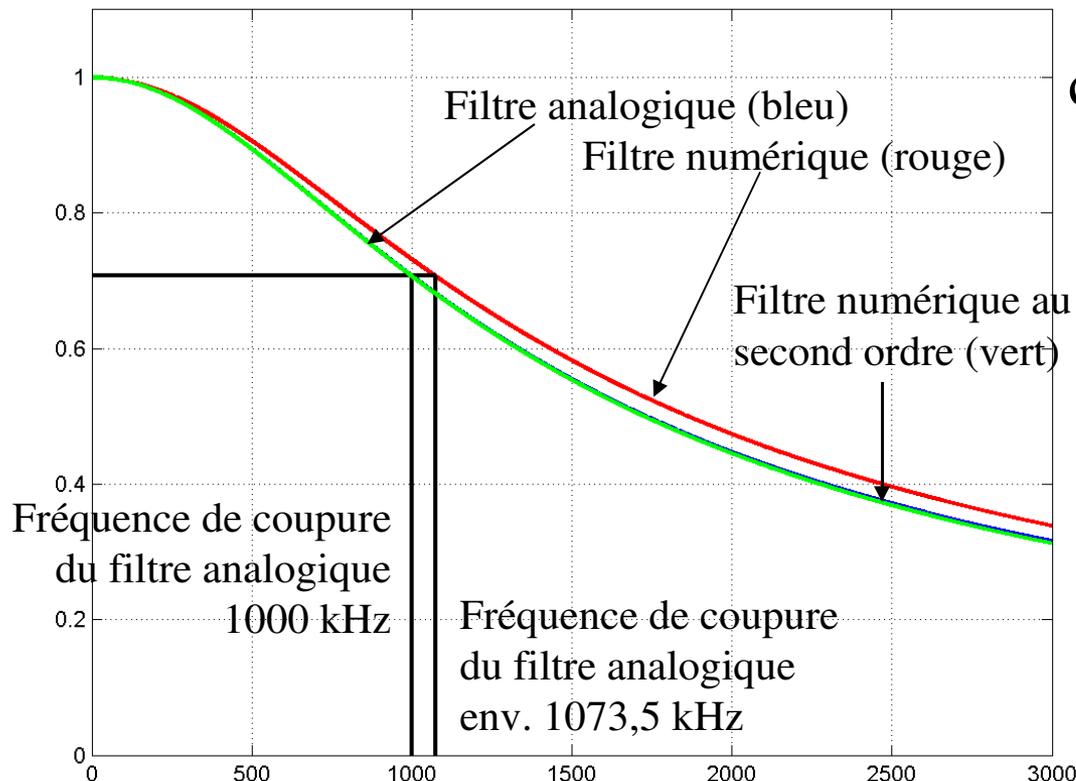
Dans $H(s) = \frac{\omega_0}{\omega_0 + s}$ on remplace s par son expression en Z tirée de

$$Z = \exp(s T_e) = 1 + s T_e + \frac{s^2 T_e^2}{2}$$

ce qui donne

$$s = \frac{2}{T_e} \frac{1 - Z^{-1}}{1 + Z^{-1}}$$

(transformation bilinéaire)



L'approximation (ci-contre) est très bonne, la fréquence de coupure vaut 999 Hz.

Synthèse de filtre FIR méthode directe (gabarit)

$$\text{Gabarit } H^i(\omega) = I_{[-\omega_0, \omega_0]}(\omega) = \begin{cases} 1 & \text{si } -\omega_0 \leq \omega \leq \omega_0 \\ 0 & \text{sinon} \end{cases}$$

On cherche h par la formule d'inversion (obtenue en prenant le chemin particulier qui est le cercle unité $Z = \exp(i\omega)$)

$$h(n) = \frac{T_e}{2\pi} \int_{-\infty}^{+\infty} H^i(\omega) \exp(in\omega T_e) d\omega = \frac{T_e}{2\pi} \int_{-\omega_0}^{+\omega_0} \exp(in\omega T_e) d\omega$$

$$h(n) = \begin{cases} \omega_0 T_e & \text{si } n=0 \\ \frac{1}{k\pi} \sin(n\omega_0 T) & \text{sinon} \end{cases}$$

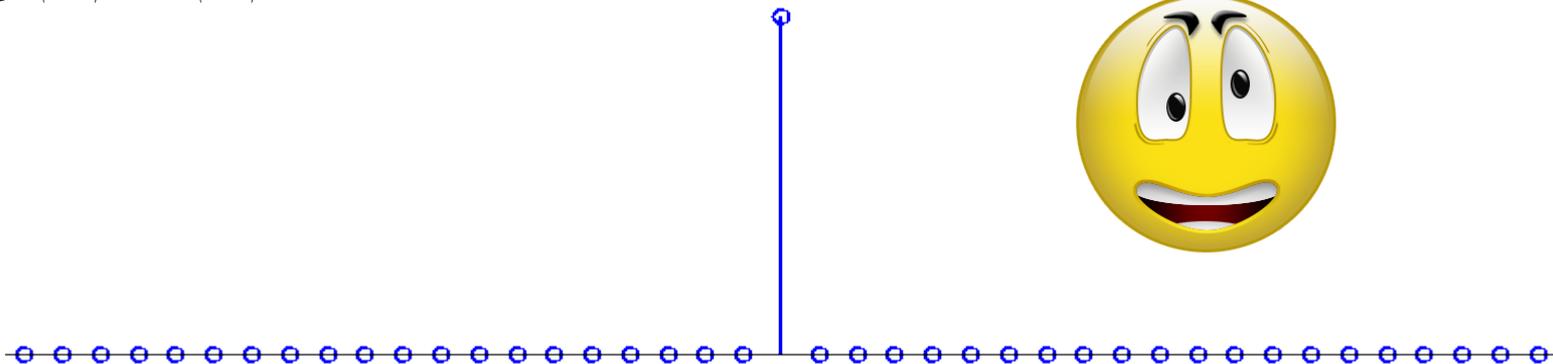
Synthèse de filtre FIR méthode directe (gabarit)

1^{ère} essai $\omega_0 = 2\pi f_0$ avec $f_0 = \frac{F_e}{2}$ la fréquence de Nyquist

Calcul trivial:

$$h(n) = \begin{cases} \frac{\omega_0 T_e}{\pi} = 1 & \text{si } n=0 \\ \frac{1}{k\pi} \sin(n\omega_0 T) = 0 & \text{sinon} \end{cases}$$

i.e. $y(n) = u(n)$



Synthèse de filtre FIR méthode directe (gabarit)

2nd essai $\omega_0 = 2\pi f_0$ avec $f_0 = \frac{F_e}{4}$

Calcul élémentaire:

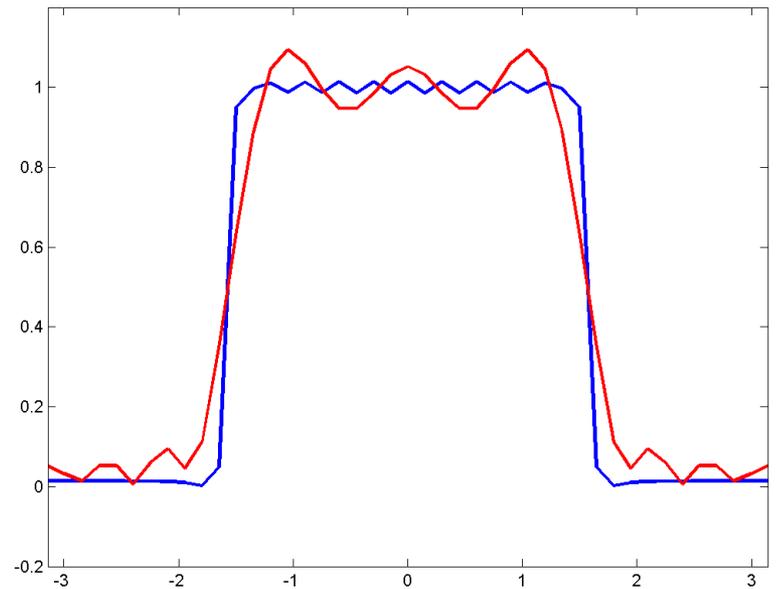
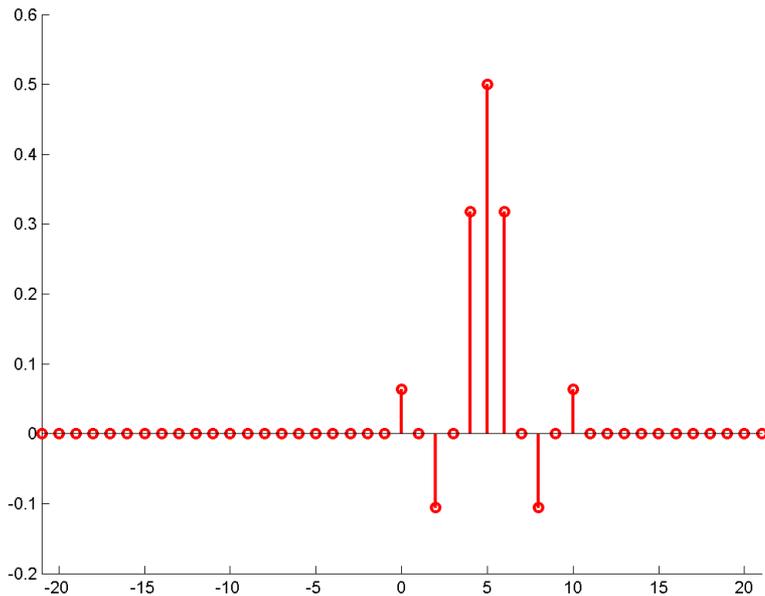
$$h(n) = \begin{cases} \frac{1}{2} & \text{si } n=0 \\ 0 & \text{si } n \text{ est pair non nul} \\ \frac{1}{n\pi} (-1)^{\frac{n-1}{2}} & \text{si } n \text{ est impair} \end{cases}$$

i.e. filtre **non causal** et **non rationnel**

- Il faut tronquer la série: risque d'apparition du phénomène de Gibbs
- Il faut mettre un retard: pas trop problématique en général...

Synthèse de filtre FIR méthode directe (gabarit)

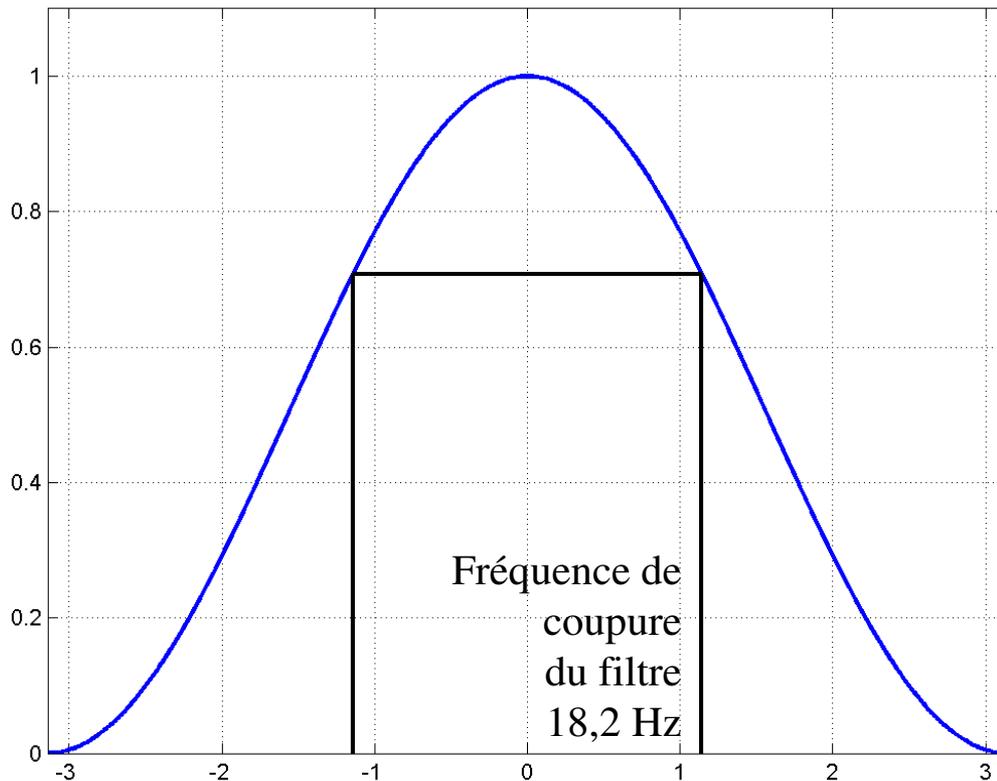
$$H(Z) = \sum_{n=-\infty}^{+\infty} h(n) Z^{-n}$$



Synthèse de filtre FIR méthode directe (gabarit)

Fenêtre de Von Hann $H^i(\omega) = \begin{cases} \frac{1}{2} (1 + \cos(\omega T_e)) & \text{si } -\omega_0 \leq \omega \leq \omega_0 \\ 0 & \text{sinon} \end{cases}$

Pour $\omega_0 = 2\pi f_0$ avec $f_0 = \frac{F_e}{2}$
on obtient



$$h(n) = \begin{cases} \frac{1}{2} & \text{si } n=0 \\ \frac{1}{4} & \text{si } n=\pm 1 \\ 0 & \text{sinon} \end{cases}$$

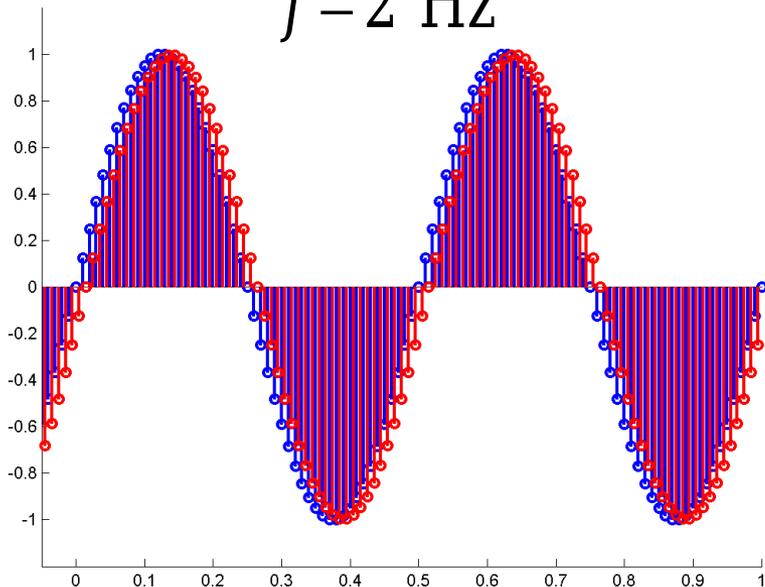
Synthèse de filtre FIR méthode directe (gabarit)

$$y(n) = \frac{1}{4} (u(n) + 2u(n-1) + u(n-2))$$

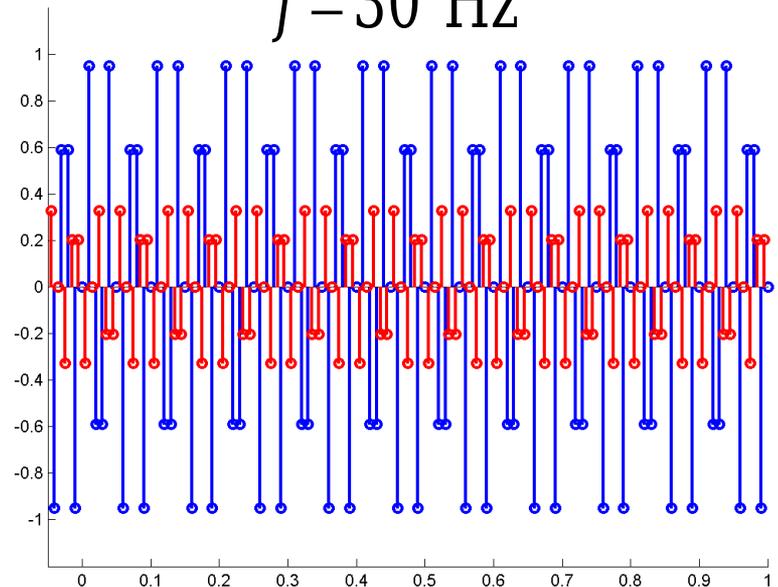
$$u(n) = \cos(2\pi f n T_e)$$

$$F_e = 100 \text{ Hz}$$

$f = 2 \text{ Hz}$



$f = 30 \text{ Hz}$

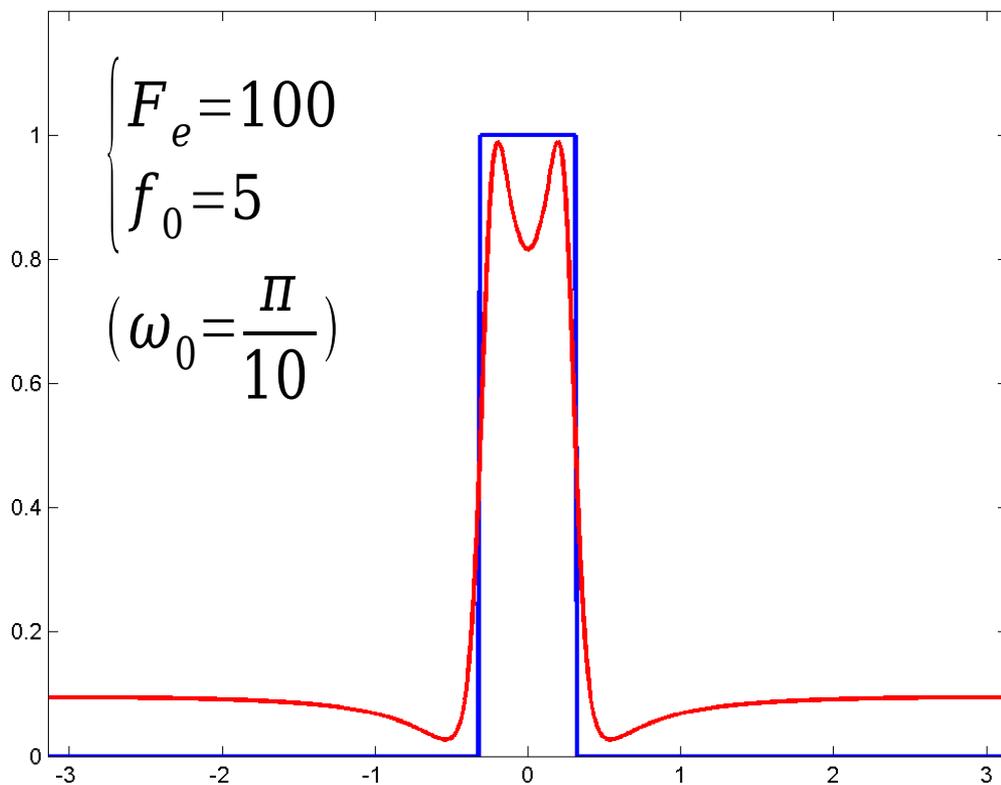


Synthèse de filtre IIR

Approximants de Padé

$$H^i(\omega) = I_{[-\omega_0, \omega_0]}(\omega) = \frac{a + bZ^{-1} + cZ^{-2}}{1 + dZ^{-1} + eZ^{-2}}$$

dont on doit surveiller la stabilité.



```

x0=randn(5,1);
x=fminsearch(@pade,x0);
w=-pi:0.01:pi;
g=(abs(w)<pi/10);
z=exp(sqrt(-1)*w);
h=(x(1)*z.^2+x(2)*z+x(3))./(z.^2+x(4)*z+x(5));
x(1:3)=x(1:3)/max(abs(h));

```

```
abs(roots([1;x(4:5)]))
```

```

function y=pade(x);
w=0:0.01:pi;
g=(w<pi/10);
z=exp(sqrt(-1)*w);
h=(x(1)*z.^2+x(2)*z+x(3))./(z.^2+x(4)*z+x(5));
y=norm(h-g,2);

```

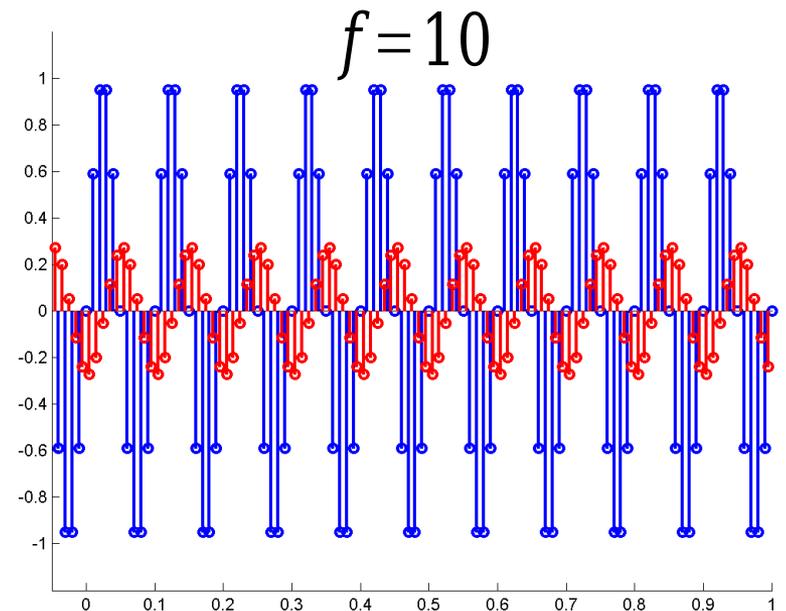
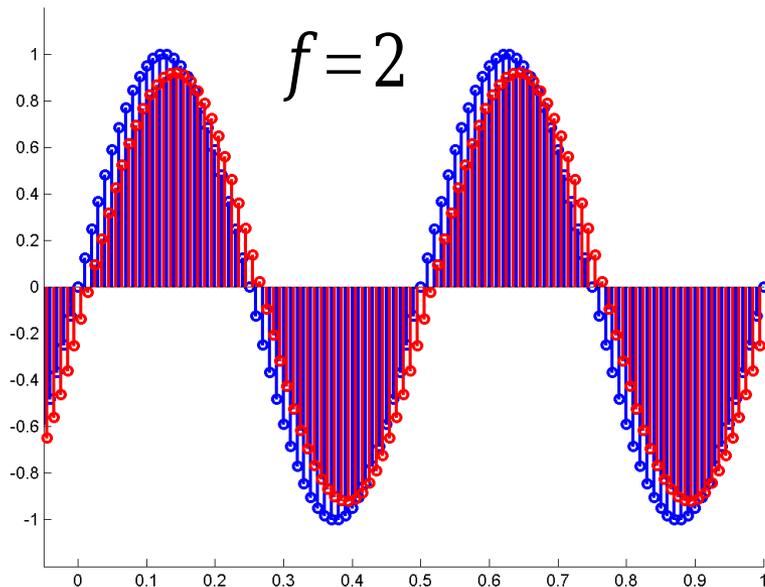
Synthèse de filtre IIR

Approximants de Padé

Si les pôles (racines de $Z^2+dZ+e=0$) ont un module inférieur à 1, le filtre est stable. Il s'écrit

$$y(n) = -dy(n-1) - ey(n-2) + au(n) + bu(n-1) + cu(n-2)$$

$$u(n) = \sin(2\pi ft):$$



Compression MP3

- 20%: Codage de Huffman (compression sans perte de données binaires)
- 80%: Modèle psycho-acoustique
 - Filtres passe bande pour récupérer 32 signaux
 - Changement de représentation: MDCT (Modified Discrete Cosine Transform), proche DTFT
 - Suppression des fréquences inaudibles
 - Mélange des fréquences indistinguables

Mise en œuvre (en fonction des applications)

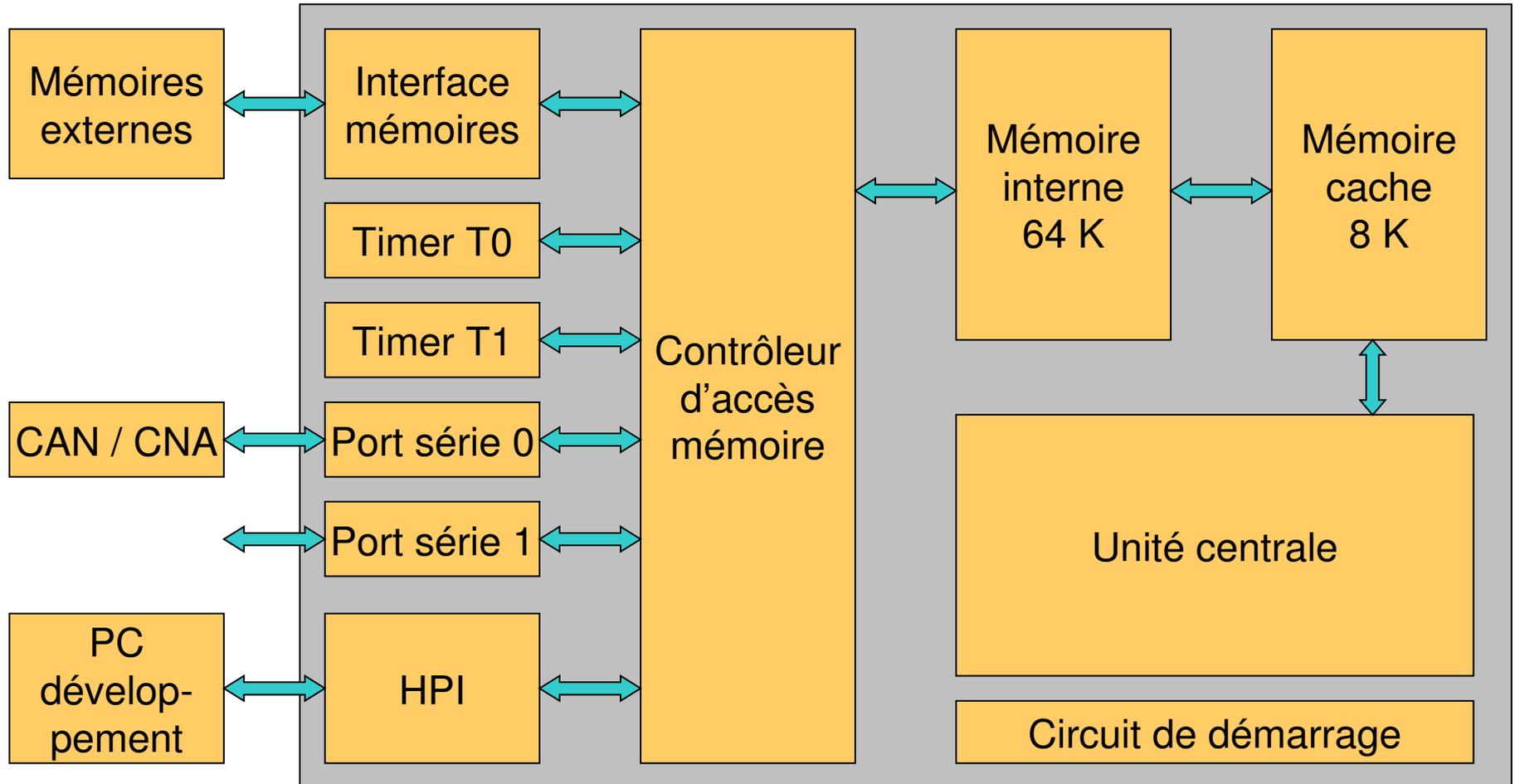
	Fréquence du signal	F_e	T_e	Nombre de cycles(*)
Système de mesures	1 Hz	4 Hz	0,25 s	50 millions
Traitement de la parole	2 kHz	8 kHz	125 μ s	25 000
CD audio	16 kHz	44,1 kHz	22,7 μ s	4535
Analyse de vibrations	40 kHz	100 kHz	10 μ s	2000
Traitement d'image	2,5 MHz	12,5 MHz	80 ns	16

(*) Nombre d'instructions par seconde, pour un processeur cadencé à 200 MHz

Mise en œuvre (en fonction des applications)

- PC
- Microcontrôleurs
 - traitement de la parole
- DSP à virgule fixe: quantification en nombre de bits, vitesse en Mips, ex: TMS320C64x (Texas Instrument) 4800 Mips 32 bits
 - traitement d'image
 - analyse de vibrations
- DSP à virgule flottante: quantification en nombre de bits, vitesse en MFlops, ex: TMS320C67x (Texas Instrument) 1350 Mflops 64 bits
 - cartes son
 - lecteurs CD
 - téléphones portables
 - ...
- FPGA (Field-Programmable Gate Array)
 - Applications dédiées

Digital Signal Processor



C.I. carte son

