

# Traitement du signal

## Notes de cours

Edition du 01/09/2016

Copyright © 2001-2016 Faculté Polytechnique de Mons



**Thierry Dutoit**  
**Faculté Polytechnique de Mons**  
**TCTS Lab**

Bvd Dolez, 31  
B-7000 Mons  
Belgium

Ph: +32 65 374774  
Fax: +32 65 374729  
Thierry.Dutoit@fpms.ac.be  
<http://tcts.fpms.ac.be/~dutoit>

# CHAPITRE 2

## INTRODUCTION

*I just wondered how things were put together.*

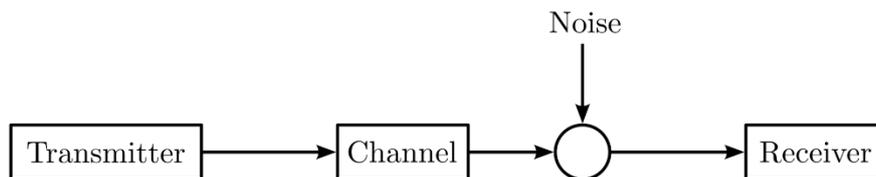
Claude E. Shannon

### 2.1 Classification des signaux

Un *signal* est une *variation* d'une grandeur physique (le plus souvent électrique, délivrée par un capteur<sup>1</sup>) qui porte de *l'information*<sup>2</sup>.

On classe souvent les signaux les plus étudiés en fonction de leur origine, c'est-à-dire en fonction du type de *systèmes* qui les produisent. On trouve ainsi :

- Les signaux de télécommunications, qui transportent l'information sous forme de signaux obtenus par modulation d'une porteuse, dans un émetteur, et démodulation dans un récepteur (Fig. 2.1).



---

<sup>1</sup> Dans certains cas, lorsqu'il serait dangereux d'utiliser des signaux électriques pour transporter de l'information – on pense par exemple aux signaux de commande d'une vanne d'une colonne à distiller le pétrole – l'industrie utilise des signaux hydrauliques.

<sup>2</sup> La notion d'*information* intervient dans celle de *signal* afin de ne pas considérer comme signal une variation de grandeur physique portant, non pas de l'information, mais de l'*énergie*, comme c'est le cas par exemple de la sinusoïde à 50Hz du réseau de distribution électrique.

Fig. 2.1 Schéma-bloc typique d'un système de télécommunication.  
[[https://en.wikipedia.org/wiki/Communications\\_system](https://en.wikipedia.org/wiki/Communications_system)].

- Les signaux géophysiques, tels que l'évolution du niveau (Fig. 2.2), de la température, de la pression, de la vitesse d'écoulement de fluides terrestres, les signaux acoustiques, les ondes sismiques, etc.

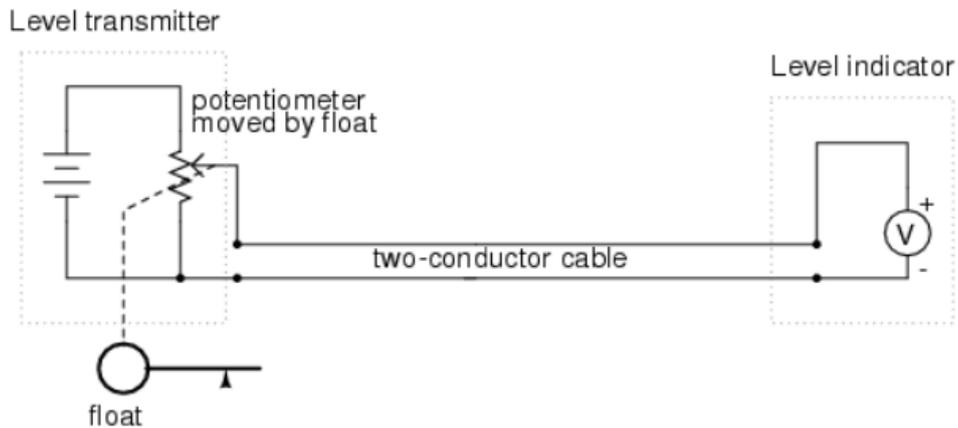


Fig. 2.2 Mesure du niveau de liquide dans un réservoir, par modification d'une résistance électrique en fonction de la position d'un flotteur dans le réservoir.  
[<http://www.allaboutcircuits.com/textbook/direct-current/chpt-9/voltage-signal-systems/>].

Un cas particulier très important est l'image, variation de luminosité et de couleur perçue par l'œil, l'appareil photo (Fig. 2.3) ou la caméra vidéo.



Fig. 2.3 Image de Lena, souvent utilisée comme image test en Traitement de l'Image.

- Les signaux biologiques, mesurés à divers endroits du corps humain et renseignant sur son fonctionnement. On peut citer par exemple l'électrocardiogramme (Fig. 2.4), l'électro-encéphalogramme, l'électro-oculogramme ou l'électro-myogramme, etc.

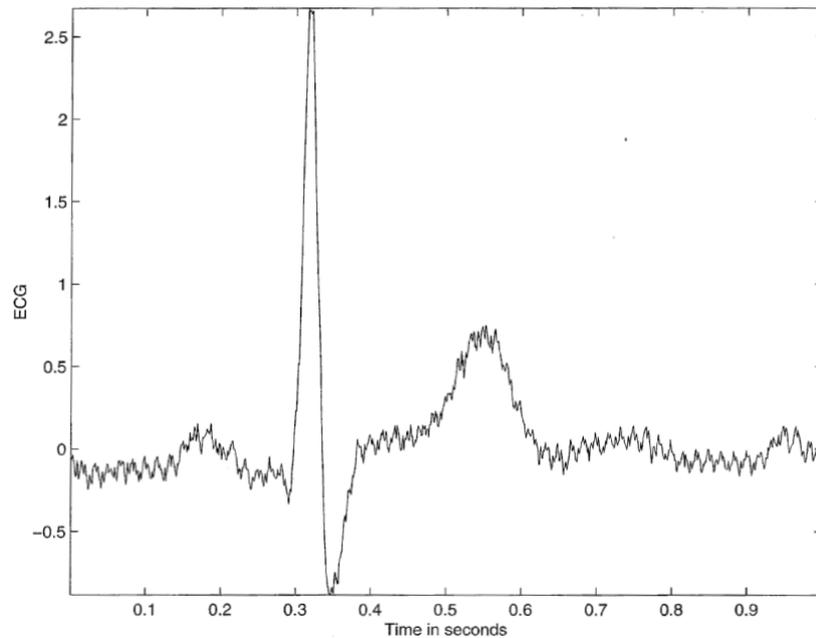


Fig. 2.4 Un battement d'ECG, pollué ici par le réseau électrique. Un tel signal devra être *filtré* avant d'être utilisé (par exemple pour l'analyse de rythme cardiaque). Le filtrage permettra d'éliminer la contribution à 50 Hz du réseau.

- Les signaux de communication interpersonnelle, tels la parole ou l'écriture (Fig. 2.5).

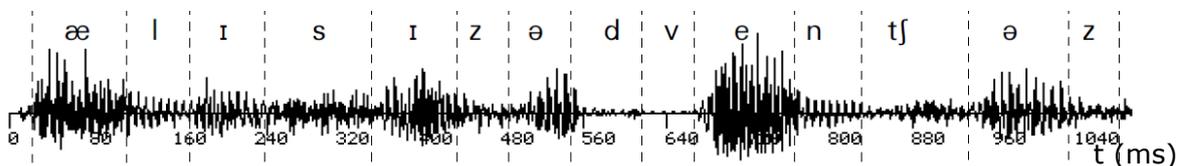


Fig. 2.5 Signal de parole (« Alice's adventures »). La grandeur portée en ordonnée est une image de la pression mesurée par un micro ; la ligne de base est la pression atmosphérique. Le signal se décompose en unités plus ou moins distinctes, qui sont perçues par l'oreille comme des sons articulés porteurs d'information (ces sons sont notés sur le graphique au moyen de l'alphabet phonétique international).

## 2.2 Caractérisation des signaux

### Temps continu – Temps discret ; Analogique – Numérique

Un signal à *temps continu* peut être décrit par une fonction continue du temps  $f(t)$ . Au contraire, un signal à *temps discret* n'a de valeurs qu'à des instants discrets, souvent répartis de manière régulière sur l'axe du temps. On le notera plutôt  $f(n)$  (où  $n$  est un entier).

L'opération de discrétisation dans le temps s'accompagne systématiquement d'une opération de discrétisation sur les valeurs prises par le signal.

On parle de signal *analogique* lors qu'on est à temps continu (et donc en général à valeurs continues également) et de signal *numérique* lorsqu'on est à temps discret (et donc en général en valeurs discrètes).

Cette distinction fera l'objet d'une section spécifique sur l'échantillonnage et la quantification (0.3).

### Monodimensionnel – Multidimensionnel

On peut classer les signaux en fonction de leur dimension. Ainsi, la plupart des signaux ci-dessus sont représentables mathématiquement par des fonctions d'une seule variable réelle. On les appelle donc signaux *monodimensionnels*. L'image, au contraire, est un signal bidimensionnel. Dans le cadre de ce cours, nous nous intéresserons aux signaux monodimensionnels.

### Déterministe - Aléatoire

Il est également fréquent de classer les signaux en fonction de la connaissance que l'on peut en avoir. Bien que tous les signaux soient par essence déterministes (ils sont tous créés par des causes physiques précises, bien que parfois non mesurables), on réserve en général le qualificatif de *déterministe* pour les signaux dont on peut expliquer l'allure temporelle (ce qui implique qu'on peut alors souvent lui associer une équation analytique). Les autres signaux sont qualifiés d'*aléatoires* et sont traités par des outils particuliers (Fig. 2.6). Dans ce cours, nous nous intéresserons uniquement aux signaux déterministes. Les signaux aléatoires seront étudiés dans des cours plus spécialisés.

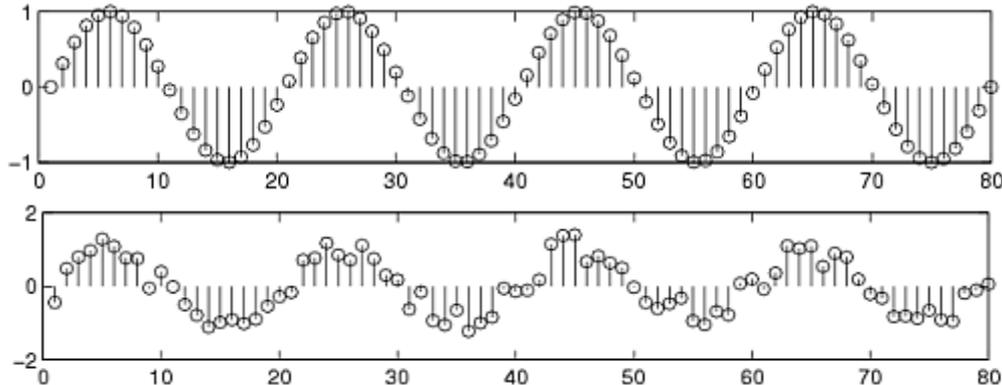


Fig. 2.6 Haut : signal périodique (ici à temps discret); Bas : signal non périodique.

[<http://pilot.cnxproject.org/content/collection/col10064/latest/module/m10057/latest>].

### Causal – Non causal

Enfin, un signal est dit *causal* lorsqu'il est identiquement nul pour tous les instants négatifs. Un tel signal résulte souvent d'une action particulière à l'instant  $t=0$ . Il est dit *anticausal* s'il est nul pour tous les instants positifs. Dans le cas général, il est *non-causal* (Fig. 2.7).

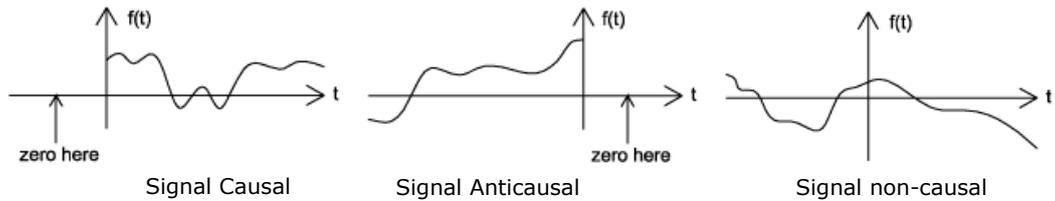


Fig. 2.7 Signal causal, anticausal, et non-causal.

[<http://pilot.cnxproject.org/content/collection/col10064/latest/module/m10057/latest>].

### Périodique – Apériodique

Les signaux périodiques (Fig. 2.8) se répètent après une période  $T_0$  :

$$f_{T_0}(t) = f_{T_0}(t + T_0) \quad (2.1)$$

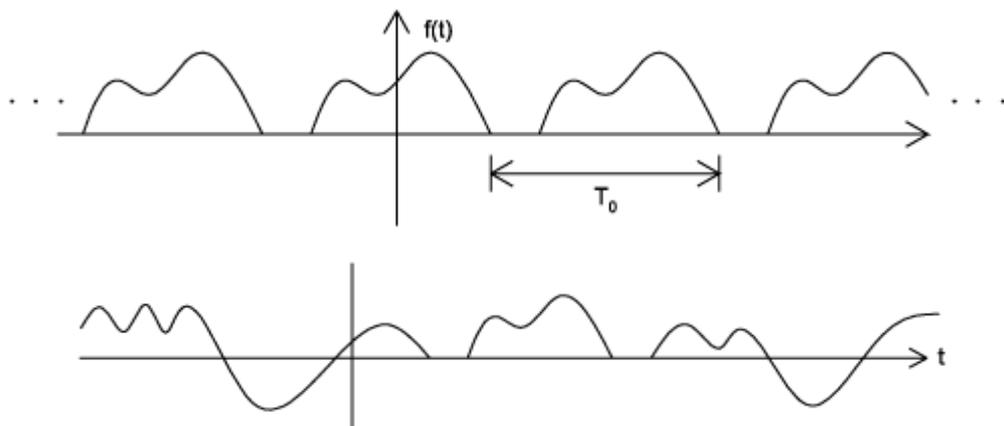


Fig. 2.8 Haut : signal périodique de période  $T_0$ ; Bas : signal non périodique  
[<http://pilot.cnxproject.org/content/collection/col10064/latest/module/m10057/latest>].

### Pair – Impair

Un signal est pair si  $f(t) = f(-t)$ ; il est impair si  $f(t) = -f(-t)$ .

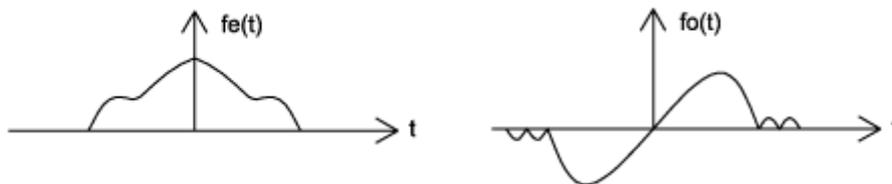


Fig. 2.9 Gauche : signal pair; Droite : signal impair

[<http://pilot.cnxproject.org/content/collection/col10064/latest/module/m10057/latest>].

### Energie finie – puissance finie

L'énergie  $E$  d'un signal, homologue du carré de la norme d'un vecteur, et définie par :

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = E \quad (2.2)$$

Cette expression n'est définie que pour des signaux d'énergie finie. Les signaux d'énergie infinie sont par contre souvent *de puissance finie*. On définit la *puissance*  $P$  d'un signal comme la moyenne du carré du signal :

$$P = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |f(t)|^2 dt \quad (2.3)$$

En particulier, les signaux périodiques sont d'énergie infinie. L'intégrale précédente peut n'être calculée que sur une période, et l'expression de leur puissance se simplifie en :

$$P = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} |f_{T_0}(t)|^2 dt \quad (2.4)$$

Les signaux périodiques sont donc de puissance finie si leur période est un signal d'énergie finie (ce qui est le cas pour tous les signaux périodiques utilisés en pratique).

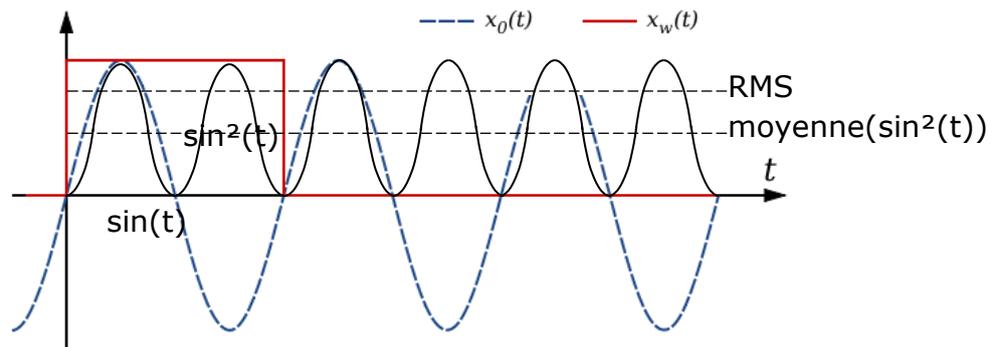


Fig. 2.10 La puissance d'un signal périodique (ici  $\sin(t)$ ) peut ne se calculer que sur une période.

[[https://en.wikibooks.org/wiki/Signals\\_and\\_Systems/Print\\_version](https://en.wikibooks.org/wiki/Signals_and_Systems/Print_version)]

On définit également la *valeur efficace*  $F_{RMS}$  (où RMS est mis pour Root Mean Square) d'un signal  $f(t)$  (Fig. 2.10) comme la racine carrée de sa puissance, c'est-à-dire comme l'amplitude d'un signal constant qui aurait la même puissance que  $f(t)$  :

$$F_{RMS} = \sqrt{\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |f(t)|^2 dt} \quad (2.5)$$

ou, pour un signal périodique :

$$F_{RMS} = \sqrt{\frac{1}{T_0} \int_{-T_0/2}^{T_0/2} |f(t)|^2 dt} \quad (2.6)$$

### Exemple 1 – Puissance et valeur efficace d'un sinus

La puissance d'un signal sinusoïdal de type  $A \sin(\omega_0 t)$ , avec  $\omega_0 = 2\pi/T_0$ , est donnée par :

$$P = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} A^2 \sin^2(\omega_0 t) dt = \frac{A^2}{T_0} \int_{-T_0/2}^{T_0/2} 1/2 + \sin(2\omega_0 t) dt = A^2/2$$

Sa valeur efficace est donc égale à  $A/\sqrt{2}$ .

## 2.3 Echantillonnage et quantification

La plupart des signaux (du moins ceux qui ont une origine naturelle) sont intrinsèquement analogiques. Il est donc logique que les machines permettant de créer ou de modifier ces signaux aient longtemps été elles-mêmes exclusivement analogiques. Le téléphone en est un bel exemple. Ce sont ces signaux analogiques qui nous intéresseront dans ce cours.

Cependant, depuis les années 70, on dispose de systèmes électroniques (CAN : convertisseur analogique-numérique, ou ADC : analog-digital converter) permettant d'échantillonner et de quantifier les signaux analogiques, les transformant ainsi en signaux numériques (Fig. 2.11). Ces derniers sont caractérisés par le fait qu'ils ne sont définis qu'à des instants discrets, appelés *instants d'échantillonnage* et multiples de la *période d'échantillonnage*  $T_e$ , et qu'ils ne peuvent prendre qu'un nombre fini de valeurs discrètes<sup>3</sup>. Inversement, on dispose de systèmes (CNA : convertisseur numérique-analogique, ou DAC : digital-analog converter) permettant de reconvertir un signal numérique en signal analogique (Fig. 2.12). Cette brèche entre l'analogique et le numérique a ouvert la voie vers la mise au point de machines, numériques cette fois (c'est-à-dire basées sur l'utilisation d'un ordinateur spécialisé ou *DSP*, pour *Digital Signal Processor*) permettant de manipuler les signaux.

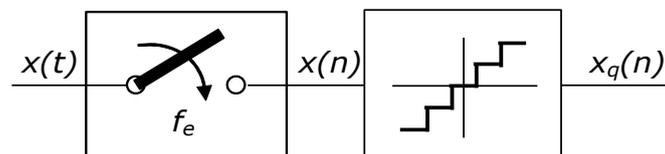


Fig. 2.11 Représentation schématique de l'échantillonnage et de la quantification d'un signal analogique.

<sup>3</sup> En pratique, évidemment, ces signaux numériques sont codés à l'aide de signaux analogiques, par *modulation numérique*. Dans le plus simple des cas, chaque échantillon est représenté à l'aide d'un nombre fini de bits (0 ou 1), et chaque bit est transcodé en analogique sous la forme de deux niveaux de tension. Le signal analogique sous-jacent est cependant toujours interprété comme un signal numérique.

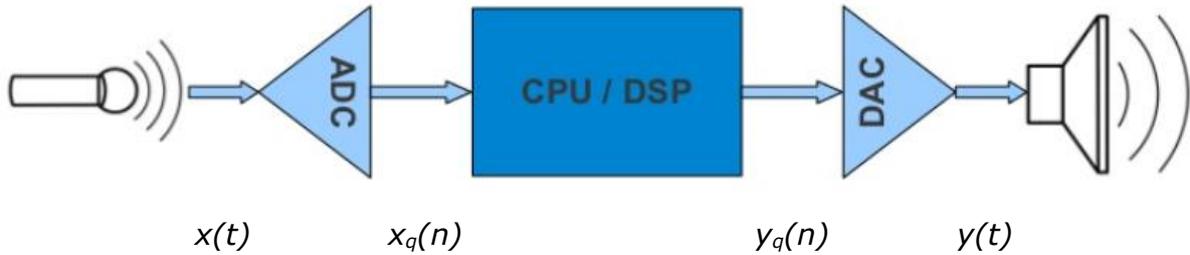


Fig. 2.12 Chaîne de traitement numérique du signal. L'entrée et la sortie du système sont analogiques ; le traitement est numérique.

L'échantillonnage d'un signal analogique représenté par une fonction  $x(t)$  consiste à construire, à partir de  $x(t)$ , un signal à temps discret  $x(n)$  obtenu en mesurant la valeur de  $x(t)$  toutes les  $T_e$  secondes :

$$x(n) = x(nT_e) \quad (2.7)$$

Intuitivement, à condition que la fréquence d'échantillonnage  $F_e$  ( $F_e=1/T_e$ ) soit suffisamment grande par rapport aux fréquences contenues dans le signal, l'effet de discrétisation dû à l'échantillonnage peut être négligé (Fig. 2.13). On montre en pratique, dans le cadre du *théorème de l'échantillonnage*, que  **$F_e$  doit être supérieure à deux fois la plus haute fréquence contenue dans le signal à échantillonner**<sup>4</sup>. Au-delà de cette valeur, le signal apparent est en effet différent du signal original (tout se passe comme si la fréquence de  $F_e/2$  jouait un effet de miroir : Fig. 2.14). Ainsi, sur un CD audio, dont le contenu fréquentiel est inférieur à 20kHz,  $F_e$  est fixé à 44100 Hz. Cette notion très importante sera revue en détail au chapitre 3.

---

<sup>4</sup> Cette condition est parfois appelée condition de Shannon. En pratique, pour assurer que cette condition est vérifiée, on place souvent, avant échantillonnage, un filtre passe-bas de fréquence de coupure inférieure ou égale à  $F_e/2$  : le *filtre de garde*.

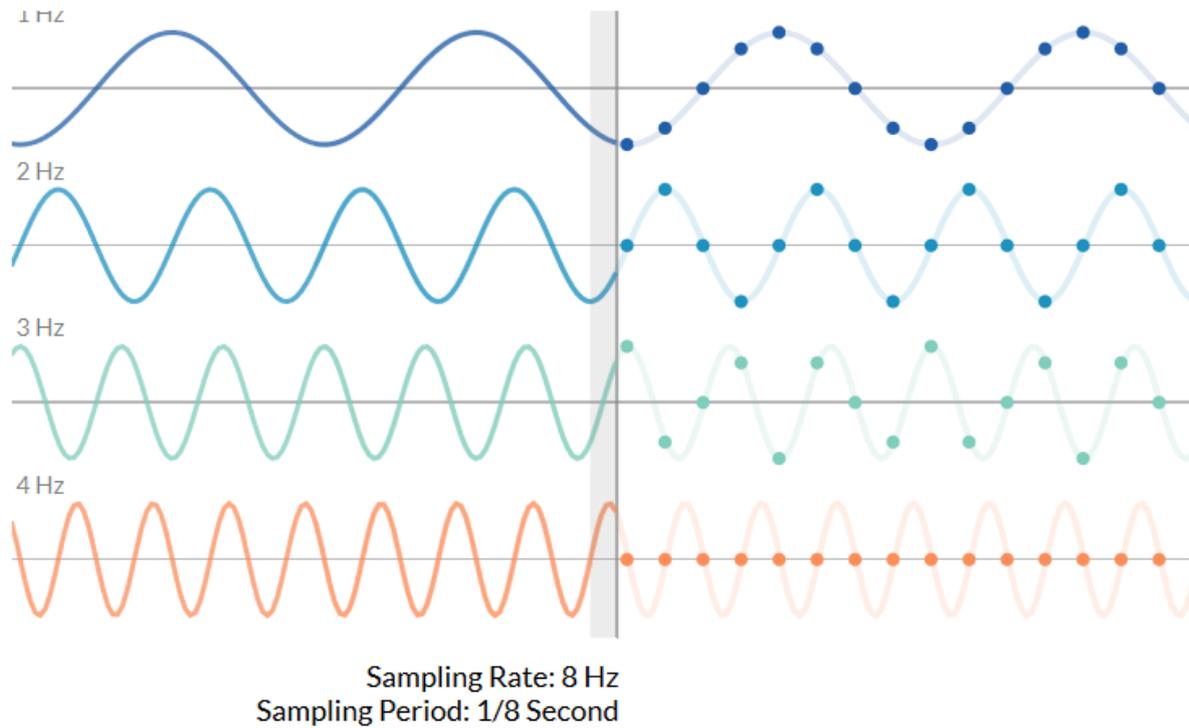


Fig. 2.13 Echantillonnage de signaux sinusoïdaux de fréquences 1, 2, 3, et 4 Hz, avec une fréquence d'échantillonnage de 8 Hz. [<http://jackschaedler.github.io/circles-sines-signals/sampling.html>]

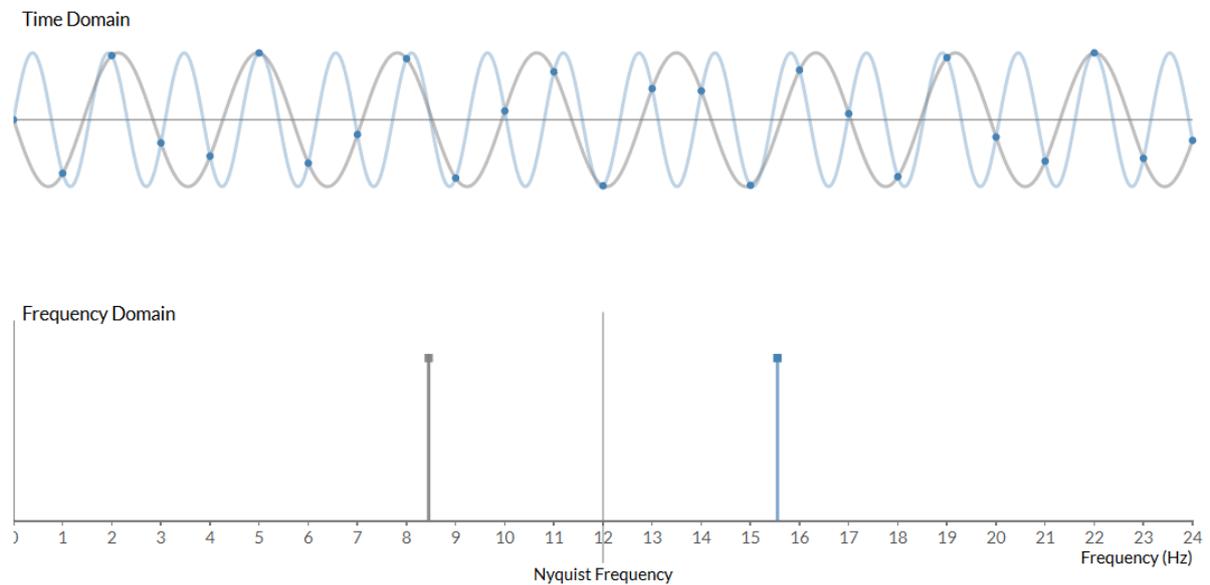


Fig. 2.14 Echantillonnage d'une sinusoïde de fréquence 15,5 Hz avec une fréquence d'échantillonnage de 12 Hz. Le signal apparent a une fréquence de 8,5 Hz. [<http://jackschaedler.github.io/circles-sines-signals/sampling2.html>]

La *quantification* d'un signal consiste à en discrétiser les valeurs sur une grille de valeurs admissibles. Typiquement, ce nombre de valeurs est un exposant de 2, de la forme  $2^b$  (ce qui permet d'associer à chaque valeur admissible un mot de  $b$

bits). Le quantificateur le plus simple est le quantificateur uniforme, caractérisé par un *pas de quantification* constant, noté  $q$  (Fig. 2.15).

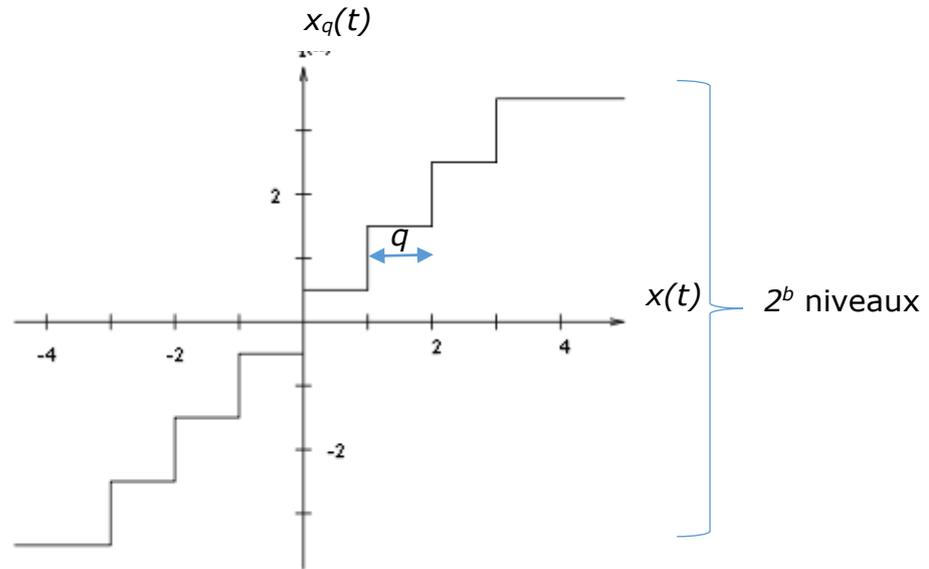


Fig. 2.15 Fonction de quantification uniforme. Le pas de quantification  $q$  est ici égal à 1. Le nombre de niveaux admissibles est ici de 8 (soit une quantification sur 3 bits). [[https://fr.wikipedia.org/wiki/Quantification\\_\(signal\)](https://fr.wikipedia.org/wiki/Quantification_(signal))]

La quantification du signal crée une *erreur de quantification*  $e(t)$  :

$$e(t) = x_q(t) - x(t) \quad -q/2 \leq e(t) \leq q/2 \quad (2.8)$$

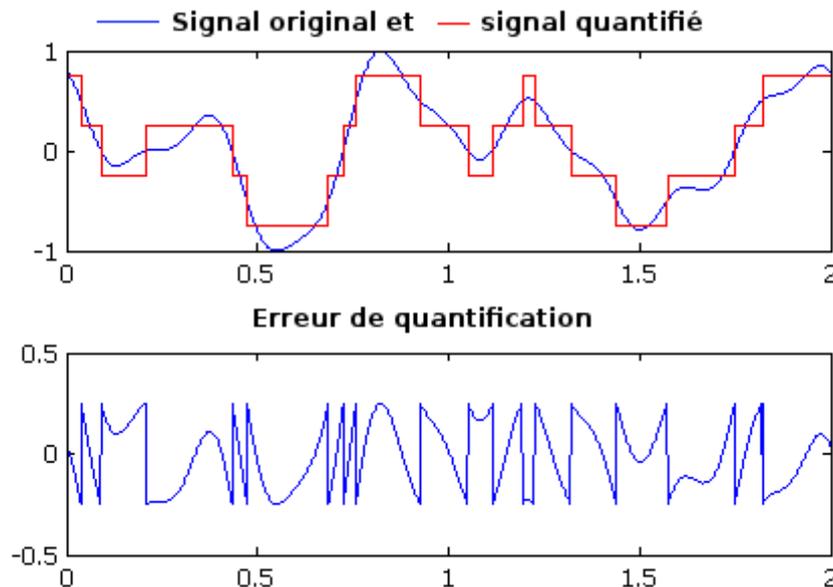


Fig. 2.16 Effet de la quantification sur un signal analogique (ici une quantification sur 2 bits et un pas  $q$  égal à  $1/2$ )

[[https://fr.wikipedia.org/wiki/Quantification\\_\(signal\)](https://fr.wikipedia.org/wiki/Quantification_(signal))]

Sur un signal audio, on perçoit cette erreur comme un bruit qui vient s'ajouter au signal à temps discret, d'où le nom usuel de *bruit de quantification* (Fig. 2.16). Pour que cet effet devienne négligeable, il faut **choisir un nombre de bits de quantification suffisant**. Sur un CD audio, la quantification est faite sur 16 bits, ce qui correspond au fait que chaque échantillon est codé comme un nombre entier entre  $-32\,768$  à  $32\,767$ , soit de  $-2^{15}$  à  $2^{15}-1$ .

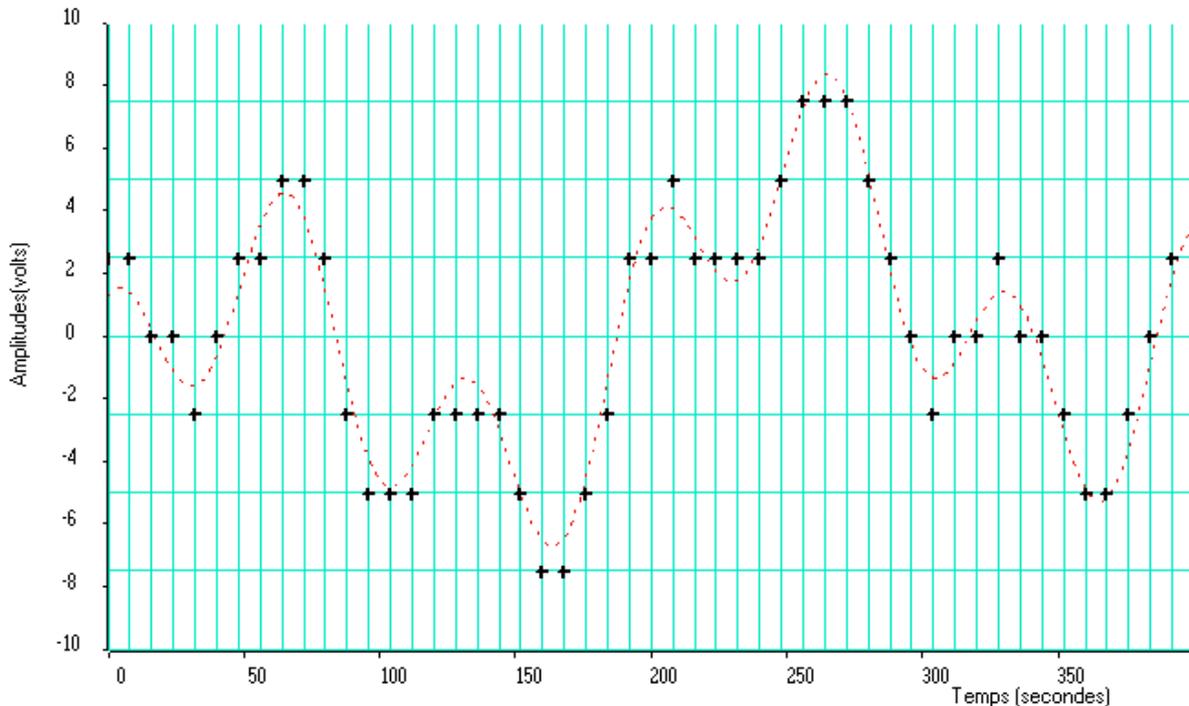


Fig. 2.17 Effet combiné de l'échantillonnage et de la quantification d'un signal analogique (en pointillé). Le résultat est une suite d'échantillons prenant des valeurs discrètes (« + »)

[<http://chamilo2.grenet.fr/inp/courses/ENSE32A4EMAAAK0/document/html/Intro.htm>]

Notons pour terminer qu'après échantillonnage et quantification, les signaux devenus numériques peuvent évidemment être transmis sous forme d'un signal numérique à temps continu, qui portera explicitement les bits d'information contenus dans le signal.

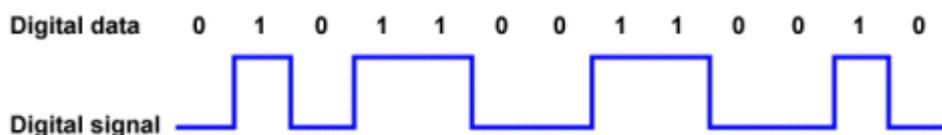


Fig. 2.18 Signal numérique à temps continu correspondant aux bits des valeurs de  $f(n)$

[[http://www.technologyuk.net/telecommunications/telecom\\_principles/digital\\_signals.shtml](http://www.technologyuk.net/telecommunications/telecom_principles/digital_signals.shtml)]

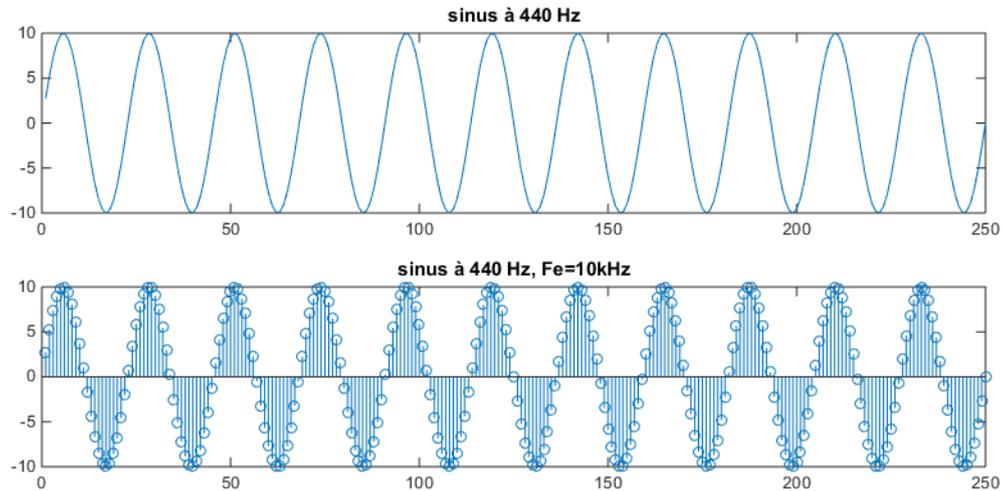
### Exemple 2 – Echantillonnage d'un sinus

```

A=10; f=440; Fe=10000;
t=(1:20000)/Fe;
signal = A * sin(2*pi*f*t); % f=440 Hz, Fe=10000 Hz

soundsc(signal,10000);
subplot(2,1,1); plot(signal(1:250)); title('sinus à 440 Hz');
subplot(2,1,2); stem(signal(1:250)); title('sinus à 440 Hz, Fe=10kHz');

```



### Exemple 3 – Quantification d'un sinus échantillonné

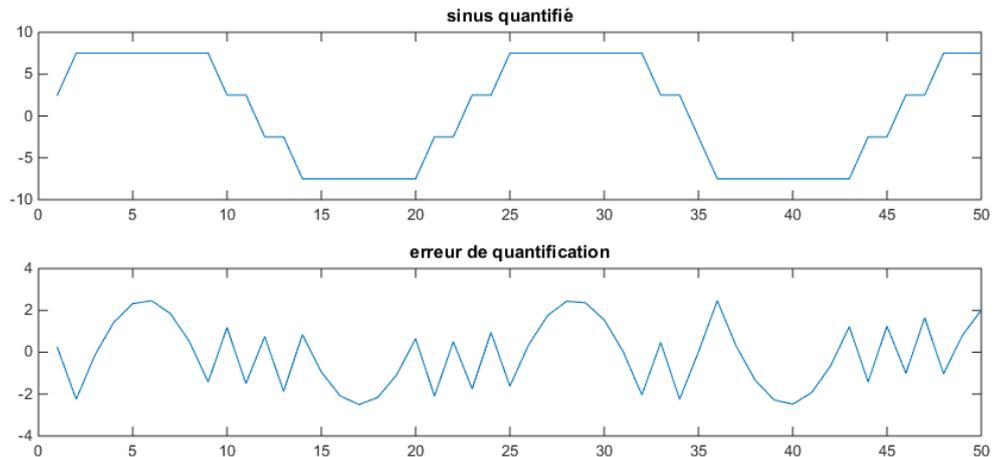
```

A = 10; f = 440; Fe = 10000;
signal = A * sin(2*pi*f*(1:20000)/Fe); % f=440 Hz, Fe=10000 Hz

b = 2;
quantizer_max = max(signal);
q = (2*quantizer_max) / 2^b;
signal_quantized = floor(signal/q)*q + q/2;
error = signal - signal_quantized;

soundsc(signal_quantized,10000);
plot(signal(1:50));
hold on;
subplot(2,1,1), plot(signal_quantized(1:50)); title('sinus quantifié');
subplot(2,1,2), plot(error(1:50)); title('erreur de quantification');

```



Il est à noter que, même sans cette opération formelle de quantification (ici à 2 bits), les échantillons sont en réalité *déjà* stockés par MATLAB sous forme quantifiée : MATLAB stocke ses variables en virgule flottante sur 8 octets, ce qui correspond en pratique à une quantification négligeable.

## 2.4 Rapport signal sur bruit

La notion de signal est souvent indissociable de celle de *bruit*, perturbation indésirable qui se superpose au signal utile. Ce qui est bruit et ce qui ne l'est pas dépend évidemment du contexte. Ainsi, par exemple, pour un opérateur sonar, le signal utile est émis par les navires et les sous-marins, et le bruit peut provenir des poissons et des crustacés qui émettent des signaux perturbateurs ; au contraire, pour l'opérateur sonar d'un bâtiment de pêche, le signal utile est celui émis par les bancs de poissons, les autres signaux sont donc des perturbations.

Un des problèmes fondamentaux en traitement du signal sera donc d'extraire le signal utile du bruit. La difficulté du problème dépend en particulier de la proportion entre signal et bruit, mesurée par le *rapport signal à bruit* (RSB, ou SNR en anglais pour signal-to-noise ratio) :

$$RSB = \frac{P_S}{P_B} \quad \text{ou, en dB :} \quad RSB_{dB} = 10 \log \frac{P_S}{P_B} = 20 \log \frac{S_{RMS}}{B_{RMS}} \quad (2.9)$$

où  $P_S$  et  $P_B$  sont respectivement la puissance du signal et du bruit. On trouvera dans le tableau ci-dessous un rappel des valeurs souvent utilisées en dB et leur lien avec des rapports de puissance et de valeur efficace (racine carrée de la puissance). Les dB sont utilisés en ingénierie lorsque les rapports s'étalent sur une large plage.

En pratique, la puissance dans (2.9) est souvent estimée sur une portion de signal :

$$P = \frac{1}{|b-a|} \int_a^b |f(t)|^2 dt \quad (2.10)$$

Notons que, sur le signal échantillonné  $f(n)$ <sup>5</sup>, cette même puissance peut être estimée comme :

$$P = \frac{1}{N} \sum_0^{N-1} |f(n)|^2 \quad (2.11)$$

Gain en puissance	Gain en amplitude	dB
1/4	1/2	$-20 \log(2) \cong -6dB$

<sup>5</sup> On suppose que la fréquence d'échantillonnage suffisante pour que le signal échantillonné donne une image réaliste du signal original.

$\frac{1}{2}$	$1/\sqrt{2}$	$-10\log(2) \cong -3\text{dB}$
4	2	6 dB
100	10	20 dB
10000	100	40 dB
1000000	1000	60 dB

Tableau 2.1 Valeurs usuelles et équivalents en dB

**Exemple 4 – Parole bruitée avec divers niveaux de RSB**

```

Signal = audioread('par8.wav'); % Fe=8000 Hz; nbits=16

% out = awgn(in,snr) : additive white Gaussian noise with given SNR
signal_plus_noise = awgn(signal, 40, 'measured');
soundsc(signal_plus_noise,8000);
subplot(5,1,1); plot(signal_plus_noise); title('signal+noise; SNR=40dB');
pause;

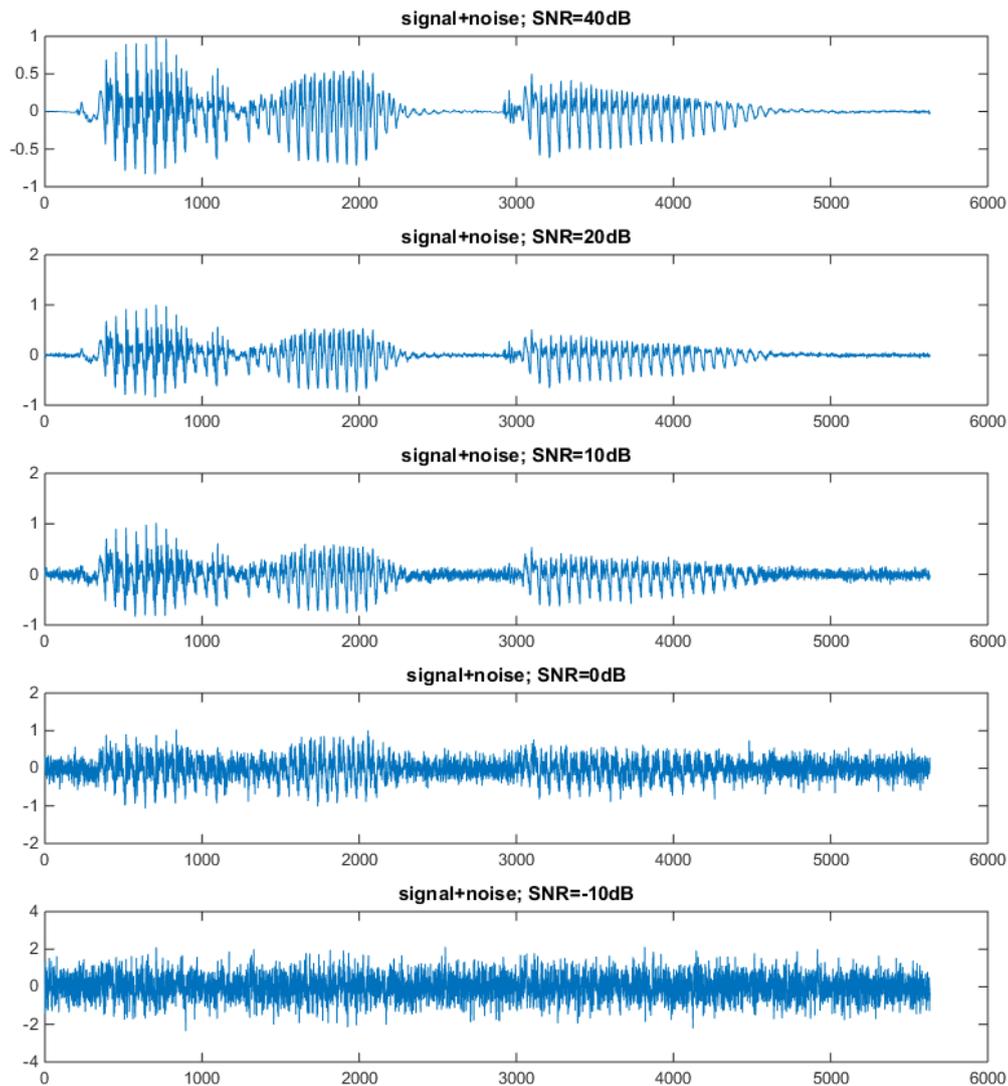
signal_plus_noise = awgn(signal, 20, 'measured');
soundsc(signal_plus_noise,8000);
subplot(5,1,2); plot(signal_plus_noise); title('signal+noise; SNR=20dB');
pause;

signal_plus_noise = awgn(signal, 10, 'measured');
soundsc(signal_plus_noise,8000);
subplot(5,1,3); plot(signal_plus_noise); title('signal+noise; SNR=10dB');
pause;

signal_plus_noise = awgn(signal, 0, 'measured');
soundsc(signal_plus_noise,8000);
subplot(5,1,4); plot(signal_plus_noise); title('signal+noise; SNR=0dB');
pause;

signal_plus_noise = awgn(signal, -10, 'measured');
soundsc(signal_plus_noise,8000);
subplot(5,1,5); plot(signal_plus_noise); title('signal+noise; SNR=-10dB');

```



Il est à noter que le RSB d'une ligne téléphonique est typiquement de 30 dB.

### Exemple 5 – Image bruitée avec divers niveaux de RSB

```

signal = rgb2gray(imread('lena.jpg')); % convert the jpg to B&W (luminance)
signal_plus_noise = imnoise(signal, 'gaussian', 0, 0.0001);
subplot(2,2,1); imshow(signal_plus_noise); title('signal+noise; SNR=40dB');

signal_plus_noise = imnoise(signal, 'gaussian', 0, 0.001);
subplot(2,2,2); imshow(signal_plus_noise); title('signal+noise; SNR=30dB');

signal_plus_noise = imnoise(signal, 'gaussian', 0, 0.01);
subplot(2,2,3); imshow(signal_plus_noise); title('signal+noise; SNR=20dB');

signal_plus_noise = imnoise(signal, 'gaussian', 0, 0.1);
subplot(2,2,4); imshow(signal_plus_noise); title('signal+noise; SNR=10dB');

```

signal+noise; SNR=40dB



signal+noise; SNR=30dB



signal+noise; SNR=20dB



signal+noise; SNR=10dB



On constate dans la pratique que le bruit est difficilement perceptible sur l'intensité d'une image au-dessus d'un RSB de 40 dB.

### Exemple 6 – Quantification et RSB

Reprenons le signal sinusoïdal précédemment quantifié. Le rapport des puissances du signal et du bruit est égal au rapport des variances des vecteurs d'échantillons correspondants.

```
A = 10; f = 440; Fe = 10000;
signal = 10 * sin(2*pi*f*(1:20000)/Fe); % f=440 Hz, Fe=10000 Hz
b = 2;
quantizer_max = max(signal);
q = (2*quantizer_max) / 2^b;
signal_quantized = floor(signal/q)*q + q/2;
error = signal - signal_quantized;
SNR = var(signal)/var(error)
SNR_dB = 10*log10(SNR)
```

```
> SNR = 19.0679
> SNR_dB = 12.8030
```

On verra dans le cours de Traitement du Signal que le RSB lié à la quantification d'un signal musical est théoriquement donné par :

$$RSB_{dB} = 6,02b + 1,76 \quad (2.12)$$

ce qui implique que chaque bit supplémentaire de quantification apporte une augmentation de 6 dB de RSB. En particulier, sur un CD audio, le RSB est de 98 dB !

## 2.5 Traitement numérique du signal

Le traitement (numérique) du signal consiste en un ensemble de théories et de méthodes, relativement indépendantes du signal traité, permettant de créer, d'analyser, de synthétiser (c.-à-d créer à partir de rien), de quantifier, de classer, de prédire, ou de reconnaître les signaux. Il s'agit donc d'une science appliquée, puisque le signal numérique n'existe pour ainsi dire pas dans la nature. Il est une invention de l'homme, qui a pour but principal de permettre une manipulation aisée de signaux analogiques à l'aide de calculateurs numériques. Ses applications sont nombreuses dans des domaines aussi variés que les télécommunications, le traitement du son musical, le traitement de la parole, le radar, le sonar, l'étude des vibrations sismiques, le contrôle non destructif, l'ingénierie bio-médicale, l'imagerie, l'économie (avec l'étude des séries chronologiques), etc.

On peut estimer que cette science de l'ingénieur a réellement vu le jour dans les années 60, avec la découverte d'algorithmes rapides de calcul de la transformée de Fourier discrète par Cooley et Tukey. Les ouvrages de base ont été publiés entre la fin des années 70 et le début des années 90. Ce n'est que récemment (fin des années 90), cependant, que les applications pratiques se sont multipliées, avec l'avènement des technologies numériques (RNIS-CD-GSM-DVD-MP3, etc.) et des processeurs spécialisés en traitement du signal (DSP : digital signal processor). On trouvera sur le site MOODLE du cours, outre des pointeurs vers divers sites et outils logiciels intéressants, les biographies d'un certain nombre de chercheurs qui ont marqué de leur empreinte l'évolution de ce que l'on qualifie aujourd'hui de *révolution numérique*. La plupart sont toujours en poste ou récemment retraités.

De nombreux journaux scientifiques sont publiés mensuellement (notamment par l'IEEE et EURASIP) sur ce sujet, que ce soit pour le développement de nouvelles idées et techniques (comme récemment l'acquisition comprimée - *sparse sampling* - qui permet d'échantillonner des signaux sous la limite de Shannon mentionnée plus haut) ou dans des domaines plus appliqués (comme par exemple l'analyse et la compréhension du contenu des images numériques).

## 2.6 Systèmes analogiques – Systèmes numériques

Les systèmes numériques possèdent sur leurs homologues analogiques un ensemble d'avantages décisifs :

- *Simplicité*. Les systèmes numériques sont intrinsèquement plus simples à analyser (et donc à synthétiser) que les systèmes analogiques. La récurrence linéaire qui caractérise un filtre numérique, par exemple, est accessible à un tout jeune enfant. Cette propriété des systèmes numériques est due en partie à l'adéquation parfaite entre simulation et traitement : *simuler un traitement numérique, c'est en faire*.

- *Possibilités de traitement accrues.* La simplicité des opérations numériques de base ne doit pas tromper : il s'ensuit qu'il est possible de réaliser, en numérique, des opérations beaucoup plus complexes qu'en analogique, notamment des opérations non-linéaires.
- *Robustesse aux bruits.* Les systèmes numériques sont par essence insensibles aux bruits parasites électromagnétiques. Le transcodage de l'information sous forme numérique joue un peu le rôle de « firewall ».
- *Précision et stabilité.* Puisque les seuls « bruits » sont liés à la précision des calculs, cette dernière dépend uniquement du calculateur utilisé ; elle est insensible à la température et ne varie pas avec l'âge du système.
- *Flexibilité.* Dans un grand nombre de systèmes numériques, le traitement est défini par un logiciel chargé en mémoire. Il est dès lors très facile de modifier ce traitement, sans devoir modifier la machine qui le réalise. On pense par exemple aux modems numériques actuels, qui peuvent s'adapter facilement aux normes futures par simple reprogrammation.

## 2.7 De l'Embarqué à l'Internet des objets

On a assisté ces vingt dernières années à une révolution plus silencieuse mais bien plus importante pour l'essor du traitement du signal : celle des réseaux informatiques et des *systèmes embarqués (embedded systems)*. Les technologies sous-jacentes au traitement numérique des signaux (CAN-CNA-DSP) sont en effet devenues tellement bon marché que les objets qui nous entourent en sont progressivement équipés. L'ancien paradigme de l'unité centrale intelligente reliée à des périphériques se trouve ainsi balayé et remplacé par une myriade de petits calculateurs embarqués et dialoguant entre eux à l'aide de signaux portés par des réseaux informatiques. Les caméras et appareils photos numériques, lecteurs de MP3 portables, agendas électroniques, tous aujourd'hui réunis dans un seul smartphone ne sont que la partie visible par le grand public de ce bouleversement en marche. L'apparition plus récente encore de systèmes informatiques programmables embarquant des CAN et CNA, à bas prix (comme les cartes Arduino, RaspBerry Pi et leur clones - Fig. 2.19), a permis aux étudiants du monde entier (et pas seulement dans les universités) de s'adonner, sans toujours le savoir, aux joies du traitement du signal.

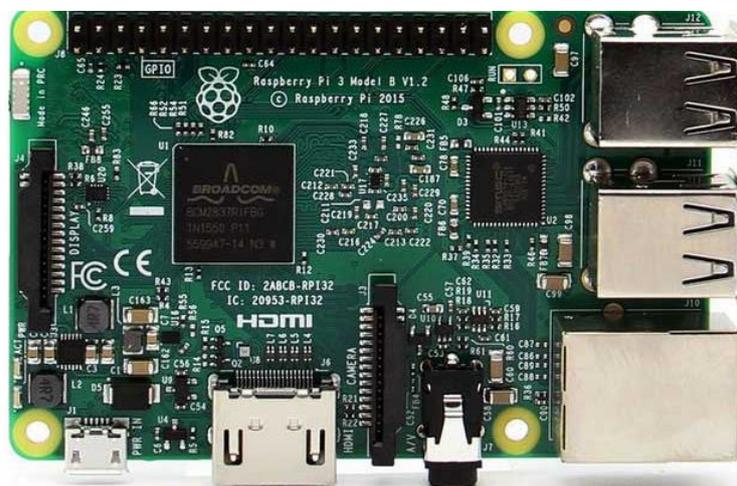


Fig. 2.19 Carte RaspBerry Pi3 : un ordinateur QuadCore et 1 GO de RAM pour 30 EU.

L' étape suivante dans ce processus est déjà en route : les objets ainsi rendus autonomes possèdent désormais une adresse IP et se mettent à communiquer à travers *l'internet des objets*. L'apparition de protocoles réseaux informatiques à bas débit et basse puissance permet en effet déjà à des objets autonomes de rester connectés pendant plusieurs années (ce qui correspond souvent par ailleurs à leur durée de vie). En particulier, les réseaux LoRA se développent en ce moment même partout dans le monde, et en particulier en Wallonie. Basé sur une technologie libre, les nœuds LoRa (Fig. 2.20) peuvent envoyer quelques centaines d'octets par jour, sur une portée de 10 km (contre 100m pour le wifi !). Un réseau mondial (et citoyen) des nœuds LoRa existe même déjà : the THINGS network.

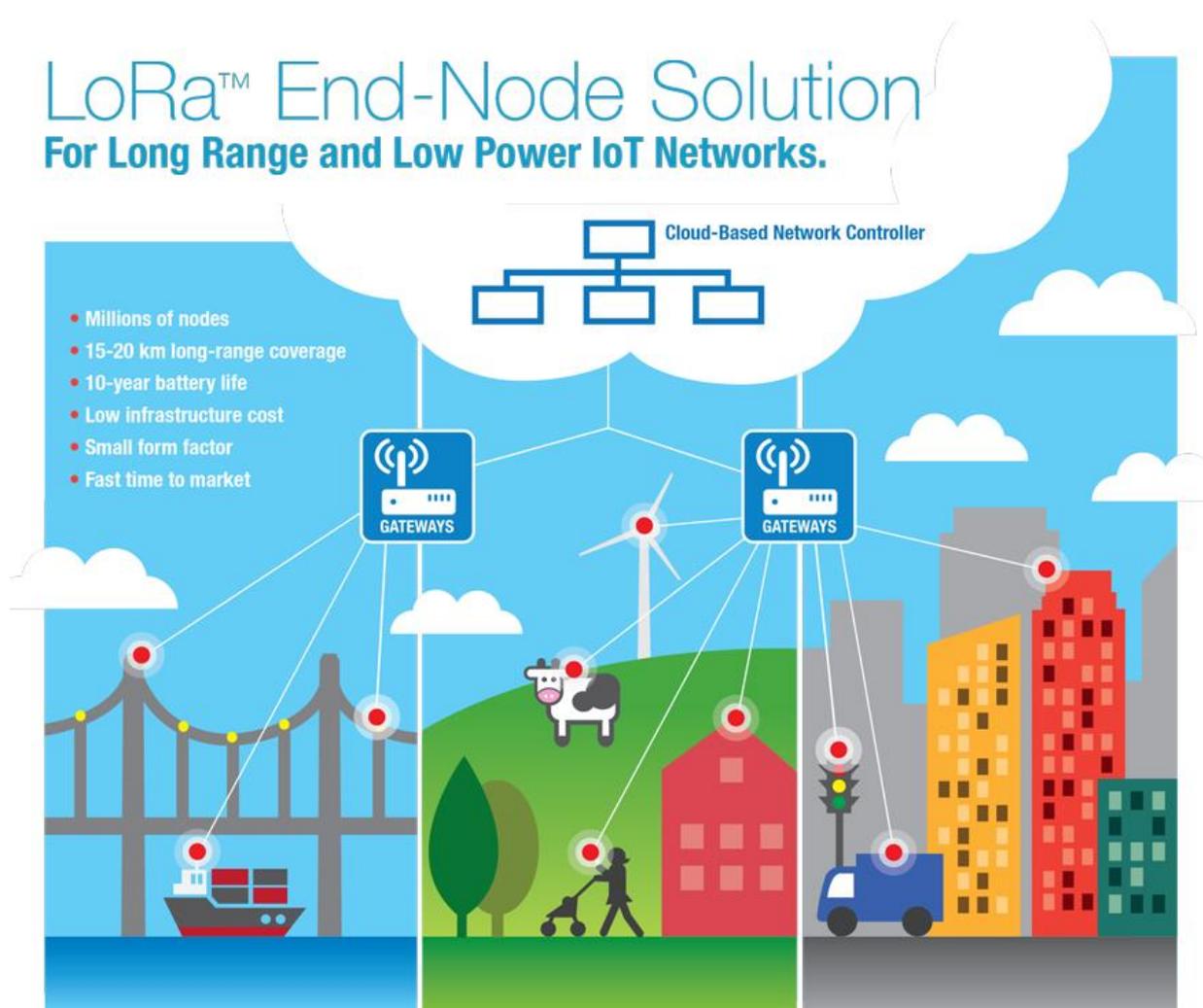


Fig. 2.20 Caractéristiques des réseaux LoRa.

[<http://www.microchip.com/design-centers/wireless-connectivity/embedded-wireless/lora-technology>]

## Organisation pratique du cours

Depuis le décret « paysage » et son application, le cours de Traitement du Signal a été séparé en deux parties : Traitement du signal 1 (16 de cours, 8h d'exercices, 24h de TP), qui est suivi par les BAB3 FPMs/IG, BAB3 FS/Info, et MAB1 FPMs/Elec, et Traitement du Signal 2 (4h de cours et 8h de projet) qui est suivi par les MAB1 FPMs/ELEC. Ces trois publics ont des formations différentes, ce qui implique une forme de redite (comme certains ont déjà pu le constater dans cette introduction), notamment en ce qui concerne la transformée de Fourier et les transmittances des systèmes linéaires et invariants, qui seront résumés aux endroits adhoc.

Les séances d'exercices sont TRES importantes, et doivent être préparées par les étudiants. En particulier, chaque séance suppose la préparation préliminaire d'un exercice (court) par les étudiants. En début de séance, un étudiant au hasard est envoyé au tableau pour y transcrire sa solution. Le reste de la séance d'exercices est consacré à une liste d'exercices que les étudiants peuvent consulter *avant* la séance sur le site MOODLE du cours. Aucun résolu n'est donné, mais le professeur et des assistants passent parmi les étudiants pour les aider.

Les 6 TP de 4h passent en revue les outils de traitement du signal sous MATLAB et SIMULINK, en relation avec le cours. Cette première partie est sanctionnée par une épreuve pratique en 7<sup>e</sup> séance. Pour les MA1, les deux dernières séances sont consacrées à un avant-projet se rapportant à un "projet grand public" basé sur un principe de traitement du signal.

L'examen ne porte que sur des exercices. La répartition des points est la suivante: TS1 (4 crédits) = EP (35/100) Examen (65/100) ; TS2 (1 crédit) = Projet (35/100) Examen (65/100).

## Remerciements

Je tiens à remercier Dominique Wynsberghe, Stéphanie Devuyst, Nicolas Riche et Alexis Moinet : leurs remarques et corrections ont apporté beaucoup à ces notes de cours. Je remercie également Thomas Dubuisson et Fabien Rogister, qui ont contribué à augmenter ces notes d'exercices progressifs et judicieux. Enfin, je suis reconnaissant aux étudiants et étudiantes de la FPMs qui ont bien voulu me signaler les coquilles présentes dans ce document (il en reste sûrement!) et qui, par leur interaction, m'ont permis de présenter cette matière d'une manière qui, je l'espère, correspond bien à la formation de l'ingénieur civil électricien.

## **Bibliographie**

Les présentes notes de cours ont bénéficié de l'apport de nombreux ouvrages. Parmi les principaux, que je recommande chaudement (dans l'ordre de priorité, du plus important au moins important) :

DENBIGH, P. (1998) *System Analysis and Signal Processing*, 513 pp. Harlow : Addison-Wesley

AUGER, F. (1999) *Introduction à la théorie du signal et de l'information*, 461 pp. Paris : Technip

BAHER, H. (2001) *Analog and Digital Signal Processing*, 497 pp. Chichester : Wiley & Sons

LYONS, R.G. (1998) *Understanding Digital Signal Processing*, 517pp. Harlow : Addison-Wesley

*T. Dutoit*

*30/08/16*

# CHAPITRE 1

## INTRODUCTION AUX SYSTEMES NUMERIQUES LINEAIRES ET INVARIANTS

Leonardo Pisano Fibonacci (1170-1250), mathématicien italien spécialiste de la théorie des nombres. Parmi les nombreux problèmes qu'il a traités, celui qui a donné naissance à la *séquence de Fibonacci* est resté célèbre : « *Un homme place un couple de lapins dans un enclos. Si on suppose que chaque couple donne naissance à un nouveau couple chaque mois, et que chaque nouveau couple devient lui-même productif à partir de son deuxième mois, combien de couples aura-t-il de mois en mois ?* ». Ce problème caractérise un système linéaire numérique, dont la réponse impulsionnelle est  $\{1,1,2,3,5,8,13\dots\}$ .



Ce chapitre constitue une première introduction aux systèmes numériques, et en particulier aux systèmes linéaires et invariants. Il doit permettre d'en comprendre l'extrême simplicité, sur le plan des éléments et des lois d'associations entre ces éléments (par opposition aux systèmes électriques à temps continus étudiés dans un cours de théorie des circuits pour les électriciens).

L'étude précise du comportement dynamique des systèmes à temps discret est également laissée à un chapitre ultérieur, consacré à la transformée en Z.

### 1.1 Signaux numériques

Un signal numérique<sup>1</sup>  $x(n)$  est défini comme une séquence de nombres (souvent appelés *échantillons* parce qu'ils proviennent en général de l'échantillonnage d'un signal analogique; cf. Chap. 3):

---

<sup>1</sup> Ceci définit en fait un signal numérique *scalaire*. On peut également définir des signaux *vectoriels* (séquence de vecteurs) ou *matriciels* (séquence de matrices). Ainsi, une séquence d'images numériques est un signal numérique matriciel. On parle également de signaux *mono-* et *multi-dimensionnels*. Une image peut être vue comme un signal bi-dimensionnel vectoriel (chaque échantillon représente un pixel, par exemple sous la forme de ses trois composantes RGB).

## 2 INTRODUCTION AUX SYSTEMES NUMERIQUES LINEAIRES ET INVARIANTS

$$\{x(n)\} = \{\dots, x(-n), \dots, x(-1), x(0), x(1), \dots, x(n), \dots\} \quad (1.1)$$

Dans de nombreux cas, on manipule des séquences dites *causales*, à partir d'une valeur de départ  $x(0)$  (les valeurs pour  $n$  négatif sont supposées nulles):

$$\{x(n)\} = \{x(0), x(1), \dots, x(n), \dots\} \quad (1.2)$$

### Exemple 1.1

□ *Impulsion numérique* :  $\delta(n) = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{si } n \neq 0 \end{cases}$

Ce signal est facilement obtenu dans Matlab (ici les 50 premières valeurs):

```
impulse=[1, zeros(1,49)];
```

□ *Train d'impulsions numérique* :  $\delta_{n_0}(n) = \begin{cases} 1 & \text{si } n = kn_0 \quad (k \text{ entier}) \\ 0 & \text{si } n \neq kn_0 \end{cases}$

Sous Matlab (ici avec  $n_0=10$ ):

```
pulsetrain=[1, zeros(1,9)];
```

```
pulsetrain=[pulsetrain pulsetrain pulsetrain pulsetrain pulsetrain];
```

□ *Echelon numérique* :  $\varepsilon(n) = \begin{cases} 1 & \text{si } n \geq 0 \\ 0 & \text{si } n < 0 \end{cases}$

```
step=[ones(1,50)];
```

□ *Exponentielle imaginaire numérique* :  $f(n) = e^{jn\varphi}$

Sous Matlab (ici avec  $\varphi=\pi/20$ ):

```
imagexp=[exp(j*(0:49)*pi/20)];
```

La Fig. 1.1 montre comment Matlab permet d'afficher ces séquences numériques de base :

```
Subplot(2,2,1); stem(impulse);
```

```
Subplot(2,2,2); stem(pulsetrain);
```

```
Subplot(2,2,3); stem(step);
```

```
Subplot(2,2,4); stem(real(imagexp)); hold on; stem(imag(imagexp));
```

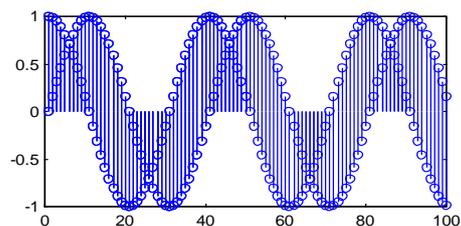
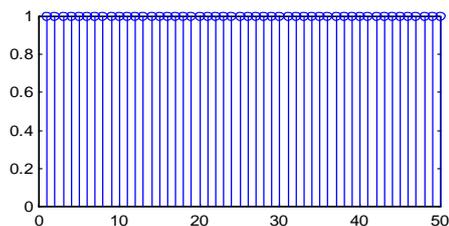
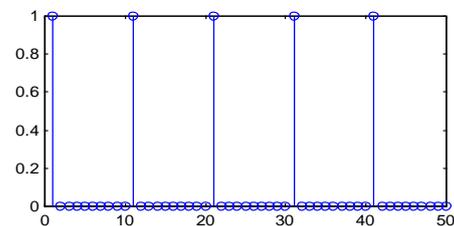
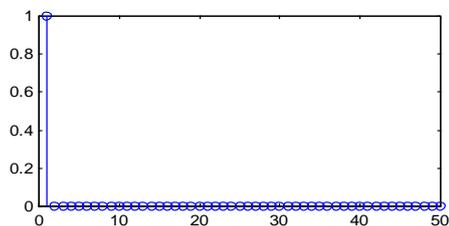


Fig. 1.1 Séquences numériques de base

## 1.2 Systèmes numériques

Un système numérique (Fig. 1.2) reçoit en entrée une séquence d'échantillons  $\{x(0), x(1), x(2), \dots\}$ , notée plus simplement  $x(n)$ , et produit en sortie une séquence d'échantillons  $y(n)$  obtenue à partir de l'entrée après application d'une série d'opérations algébriques<sup>2</sup>.

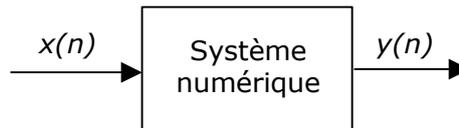


Fig. 1.2 Système numérique

Un système numérique est *linéaire et invariant* s'il répond en outre aux conditions classiques illustrées à la Fig. 1.3.

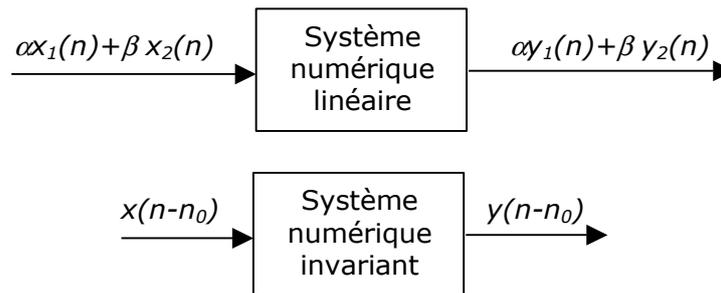


Fig. 1.3 Système numérique linéaire et invariant

Un système numérique est *stable* si, lorsqu'on lui présente une entrée finie (c.-à-d. dont tous les échantillons sont de grandeur finie), il produit une sortie finie.

### Exemple 1.2

Le système défini par la relation  $y(n) = Kx(n) + A$  (où  $K$  et  $A$  sont des constantes réelles) est non linéaire, invariant, et stable.

L'élément « délai » défini par la relation  $y(n) = x(n-1)$  est linéaire, invariant, et stable.

Le système défini par la relation  $y(n) = n x(n)$  est linéaire, non invariant, et instable.

Le système défini par la relation  $y(n) = |x(n)|^2$  est non linéaire, invariant, et stable.

## 1.3 Systèmes récurifs – Systèmes non récurifs

<sup>2</sup> En pratique un système peut avoir plusieurs entrées et/ou plusieurs sorties. On parle de système SISO (Single Input – Single Output) ou MIMO (Multiple Inputs – Multiple Outputs).

La classe la plus importante des systèmes numériques linéaires et invariants (SLI) peut être décrite par des équations aux différences finies, linéaires et à coefficients constants, de la forme :

$$y(n) + a_1 y(n-1) + a_2 y(n-2) + \dots + a_N y(n-N) = b_0 x(n) + b_1 x(n-1) + \dots + b_M x(n-M) \quad (1.3)$$

Une telle équation exprime le fait que la valeur de la sortie  $y(n)$  à l'instant courant est une combinaison linéaire des  $N$  sorties précédentes, de l'entrée courante, et des  $M$  entrées précédentes. On la note souvent de façon plus compacte :

$$y(n) + \sum_{i=1}^N a_i y(n-i) = \sum_{i=0}^M b_i x(n-i) \quad (1.4)$$

De tels systèmes sont dits *récurifs* étant donnée le caractère récursif de l'équation correspondante : il faut avoir déjà calculé toutes les sorties précédentes ( $n-1, n-2, \dots$ ) pour pouvoir calculer la sortie courante ( $n$ ).

Un cas particulier est celui des systèmes dont les valeurs de sortie sont calculées uniquement à partir du signal d'entrée :

$$y(n) = \sum_{i=0}^M b_i x(n-i) \quad (1.5)$$

De tels systèmes sont dits *non récurifs* <sup>3</sup>.

L'ordre d'un SLI numérique récursif est donné par le degré de la récursivité de l'équation aux différences finies associée :  $N$ . Pour un filtre non récursif, l'ordre est défini égal à  $M$ .

De la même façon qu'un système linéaire et invariant analogique (caractérisé par une équation intégro-différentielle entre l'entrée et la sortie, toutes deux analogiques) définit un filtre analogique, on appelle *filtre numérique* un système numérique linéaire et invariant. On parlera donc dans la suite de *filtre récursif* et de *filtre non récursif*.

Il est possible de visualiser l'équation de récurrence associée à un filtre numérique, sous la forme d'une *structure* (ou *graphe de fluence*) faisant apparaître les éléments de base suivants :

- L'additionneur, symbolisé par  $\Sigma$ , qui additionne les signaux à ses entrées.
- Le multiplieur, symbolisé par  $\triangleleft a$ , qui multiplie un signal par un scalaire  $a$
- L'élément « délai », symbolisé par  $z^{-1}$ , qui produit une sortie retardée de une valeur par rapport à son entrée.

La structure associée à (1.3) est donnée à la Fig. 1.4 <sup>4</sup>

---

<sup>3</sup> Nous verrons plus loin que tout système récursif peut se réécrire sous une forme non-récursive, mais que le nombre de coefficients du filtre non-récursif équivalent peut être très grand.

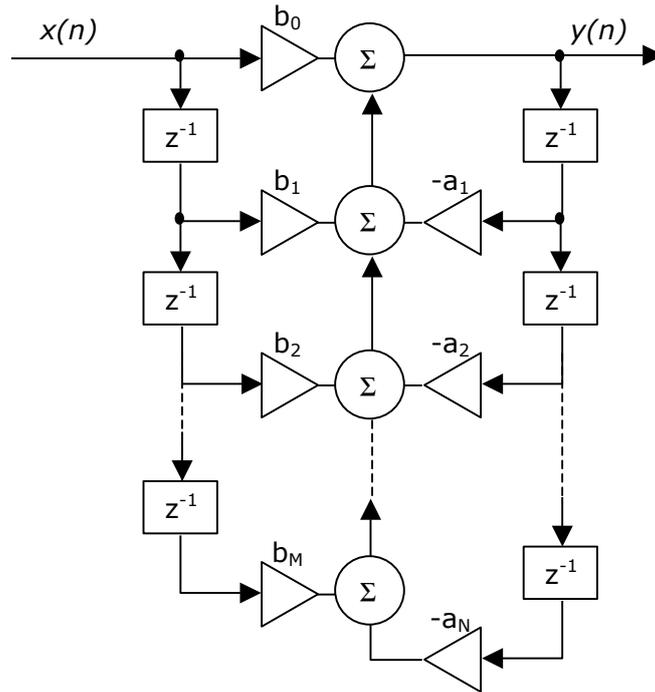


Fig. 1.4 Exemple de structure d'un filtre numérique récursif

On constate que le caractère récursif ou non récursif d'un filtre numérique est lié à la présence ou à l'absence de boucles orientées dans sa structure.

**Exemple 1.3 – Intérêts composés**

Le principe des intérêts composés sur un compte bancaire peut être facilement défini par une équation aux différences finies :

$$y(n) = y(n-1) + \frac{p}{100} y(n-1) + x(n) \tag{1.6}$$

où  $y(n)$  est la somme disponible au temps  $t=nT$  sur le compte,  $x(n)$  est le dépôt ou le retrait courant et  $p$  est le taux d'intérêt (en %). La valeur de  $T$  dépend de la politique financière de l'établissement (actualisation par jour, semaine, mois, année...).

Le système sous-jacent est récursif et d'ordre 1. Sa structure est donnée à la Fig. 1.5.

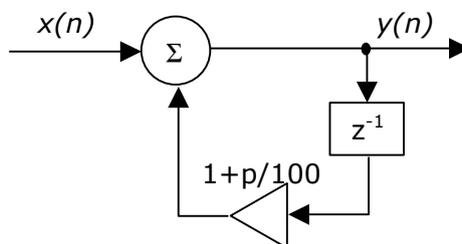


Fig. 1.5 Structure du SLI correspondant au calcul des intérêts composés

<sup>4</sup> Plus précisément, il s'agit là d'une des structures possibles. Nous verrons au chapitre consacré au filtrage numérique que plusieurs structures existent, avec chacune leurs avantages et leurs inconvénients.

**Exemple 1.4 – Séquence de Fibonacci**

La séquence de Fibonacci (voir citation en début de chapitre) peut être définie par une équation aux différences finies :

$$y(n) = y(n-1) + y(n-2) + x(n) \tag{1.7}$$

Où  $y(n)$  est le nombre de couples de lapins dans l'enclos au  $n^{\text{ème}}$  mois et  $x(n)$  est le nombre de couples que l'on ajoute à la population de l'enclos au  $n^{\text{ème}}$  mois.

Le système sous-jacent est récursif et d'ordre 2. Sa structure est donnée à la Fig. 1.6

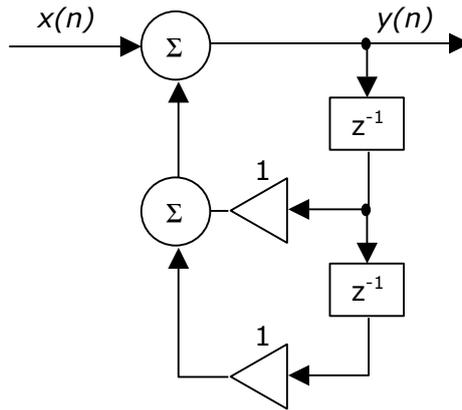


Fig. 1.6 Structure du SLI correspondant au calcul de la séquence de Fibonacci

**Exemple 1.5 – Filtre à moyenne mobile**

Imaginons que l'on veuille créer un filtre numérique dont chaque valeur de sortie  $y(n)$  soit la moyenne locale du signal d'entrée, estimée sur  $2L+1$  valeurs autour de l'échantillon courant  $x(n)$  (Fig. 1.7):

$$y(n) = \frac{1}{2L+1} \sum_{i=-L}^L x(n-i) \tag{1.8}$$

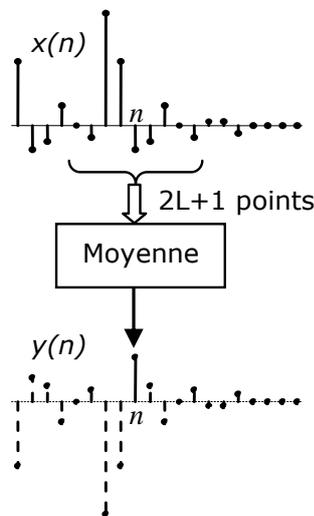


Fig. 1.7 Filtre à moyenne mobile

L'équation précédente définit bien un filtre numérique non récursif, appelé *filtre à moyenne mobile*.

Notons que ce filtre, tel que défini, est *non causal* : le calcul de la valeur courante de sortie courant nécessite de connaître les  $L$  valeurs d'entrée *suivantes*. L'existence de systèmes non causaux n'est pas nécessairement un problème dans le domaine du numérique : ces systèmes seront en pratique implémentés comme des systèmes causaux à *décali*, le décali correspondant au nombre de valeurs futures nécessaires au calcul de la valeur de sortie courante. Le système causal à décali correspondant à l'équation précédente est donné par :

$$y(n) = \frac{1}{2L+1} \sum_{i=0}^{2L} x(n-i) \tag{1.9}$$

et représenté la Fig. 1.8.

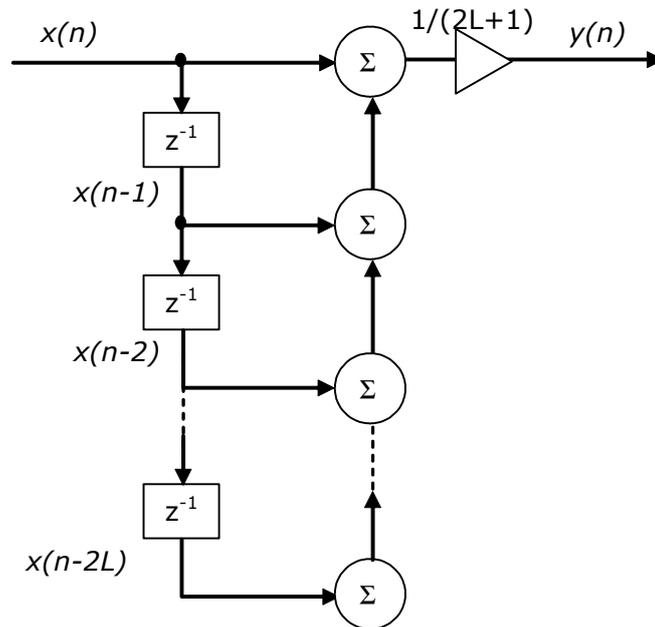


Fig. 1.8 Structure du SLI causal à décali ( $L$ ) correspondant au filtre à moyenne mobile

### 1.4 Réponse impulsionnelle

La *réponse impulsionnelle*, notée  $h(n)$ , d'un SLI numérique est alors définie comme sa réponse forcée  $y(n)$  à l'impulsion numérique (Fig. 1.9), en supposant que les conditions initiales sont nulles :  $y(n)=0$  pour  $n<0$ .

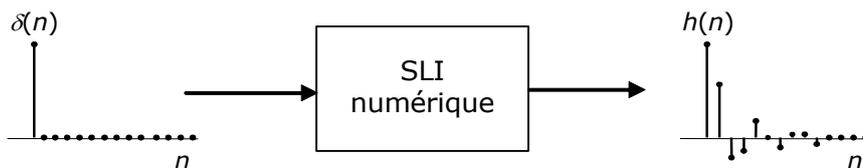


Fig. 1.9 Réponse impulsionnelle

Au contraire des systèmes analogiques, le calcul de la réponse impulsionnelle d'un SLI numérique est immédiat : il suffit d'effectuer la récurrence numérique sous-jacente<sup>5</sup>.

### Exemple 1.6

La réponse impulsionnelle du système des intérêts composés décrit par (1.6) est :

$$h(n) = \{1, 1+p/100, (1+p/100)^2, (1+p/100)^3, \dots\}$$

On peut l'obtenir sous Matlab grâce à la fonction `filter`, qui implémente la récurrence générale (1.4). Le résultat est affiché à la Fig. 1.10 (on suppose ici que  $p$  vaut 5) :

```
impresp=filter([1],[1 -1.05],impulse); stem(impresp)6;
```

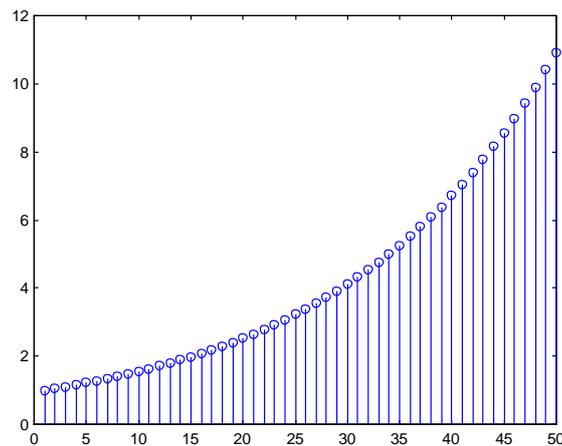


Fig. 1.10 Réponse impulsionnelle du filtre correspondant au calcul des intérêts composés

On constate en passant que ce système est instable (mais qui s'en plaindra ?), puisque sa réponse à une séquence de nombre finis produit une séquence de nombres qui tendent vers l'infini.

### Exemple 1.7

La séquence de Fibonacci est précisément la réponse impulsionnelle du système répondant à (1.7), puisqu'on suppose commencer l'élevage avec un couple de lapins et ne plus en ajouter par la suite :  $h(n) = \{1, 1, 2, 3, 5, 8, 13, 21, \dots\}$

Sous Matlab :

```
impresp=filter([1],[1 -1 -1],impulse);
impresp(1:12)
1 1 2 3 5 8 13 21 34 55 89 144
stem(impresp);
```

<sup>5</sup> Ceci illustre bien ce qui avait été annoncé dans l'introduction de ce cours : *simuler un traitement numérique, c'est l'effectuer*.

<sup>6</sup> La fonction `stem` permet de visualiser une séquence numérique sans interpolation linéaire entre les échantillons successifs, ce qui permet de ne pas la confondre avec un signal analogique.

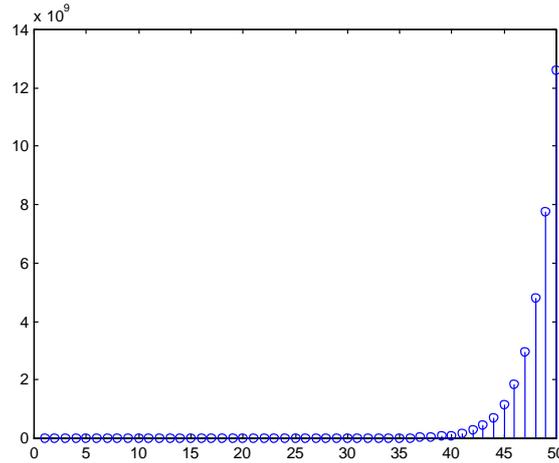


Fig. 1.11 Réponse impulsionnelle du filtre correspondant à la séquence de Fibonacci

Comme on pouvait s’y attendre, ce système, tel que modélisé ici par une simple récurrence linéaire, est également instable ☺.

On notera que la réponse impulsionnelle d’un filtre non récursif n’est rien d’autre que la suite de ses coefficients  $\{b_0, b_1, b_2, \dots\}$  (Fig. 1.12), qui se trouvent successivement poussés vers la sortie par l’impulsion qui se propage le long de la chaîne d’éléments à délai.

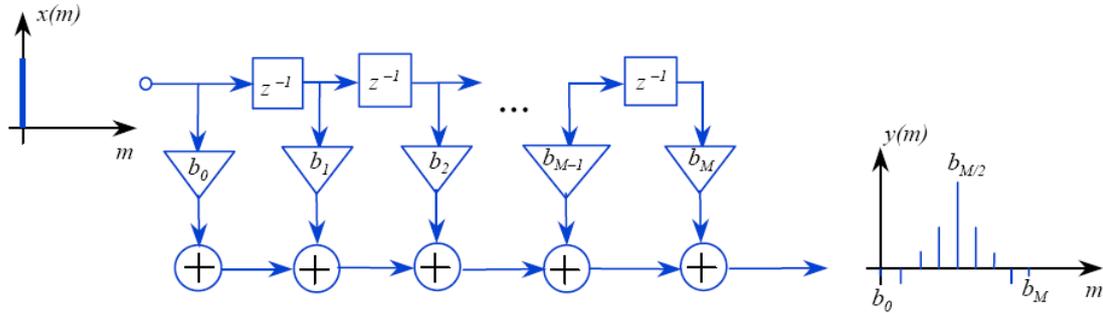


Fig. 1.12 Réponse impulsionnelle d’un filtre non récursif

**Exemple 1.8**

L’examen de la structure du filtre à moyenne mobile suffit à se convaincre que sa réponse impulsionnelle est donnée par la séquence  $1/(2L+1)\{1, 1, 1, \dots, 1, 0, 0, 0, \dots\}$  comprenant  $2L+1$  valeurs «  $1/(2L+1)$  » et une infinité de « 0 ».

Pour l'exemple qui nous occupe (avec  $L = 3$ ):

```
impresp=filter(1/7*[1 1 1 1 1 1 1],[1],impulse); stem(impresp);
```

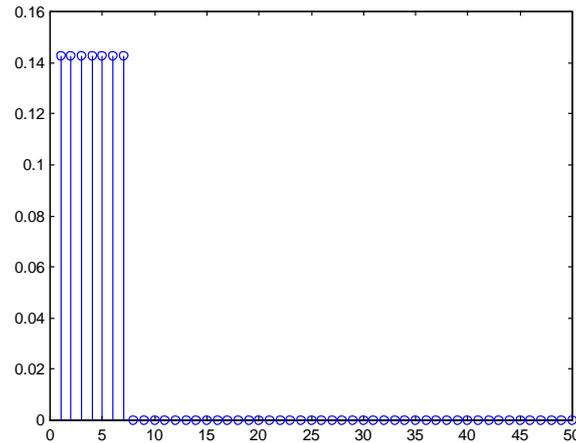


Fig. 1.13 Réponse impulsionnelle du filtre à moyenne mobile causal à délai

Nous pouvons donc penser que ce filtre est stable, bien que rien ne nous permette pour le moment d'affirmer que la réponse impulsionnelle soit suffisante pour vérifier la stabilité d'un filtre. Nous verrons à la Section suivante que c'est bien le cas.

On constate sur les exemples précédents que la réponse impulsionnelle d'un filtre récursif est une séquence, a priori illimitée, de valeurs non nulles (et ce même si le filtre est stable), puisque chaque valeur de sortie dépend des valeurs de sorties précédentes. Par contre, la réponse impulsionnelle d'un filtre non récursif est donnée par la suite de ses coefficients, et constitue donc une séquence qui s'annule après un nombre fini de valeurs. En conséquence, les filtres récursifs sont appelés *filtres à réponse impulsionnelle infinie (RII)* et que les filtres non récursifs sont dits à *réponse impulsionnelle finie (RIF)*.

## 1.5 Réponse forcée à une entrée quelconque – Convolution numérique

Le calcul de la réponse à une entrée quelconque est tout aussi trivial que celui de la réponse impulsionnelle : il suffit d'effectuer la récurrence numérique. Il est cependant intéressant de constater que la réponse à une entrée quelconque peut être déterminée facilement si on connaît déjà la réponse impulsionnelle, en effectuant une *convolution numérique*.

**Calcul par produit de convolution.** On peut en effet toujours considérer qu'une séquence d'entrée  $\{x(n)\}=\{x(0), x(1), x(2), \dots\}$  est une somme d'impulsions numériques pondérées et décalées :

$$x(n) = x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + \dots \quad (\text{pour tout } n) \quad (1.10)$$

Vu les propriétés de linéarité et d'invariance du système, la réponse à  $x(n)$  est donnée par :

$$y(n) = x(0)h(n) + x(1)h(n-1) + x(2)h(n-2) + \dots \quad (1.11)$$

ce qui n'est rien d'autre qu'une combinaison linéaire des réponses impulsionnelles à chacun des nombres de la séquence d'entrée (et donc décalées d'autant; Fig. 1.16).

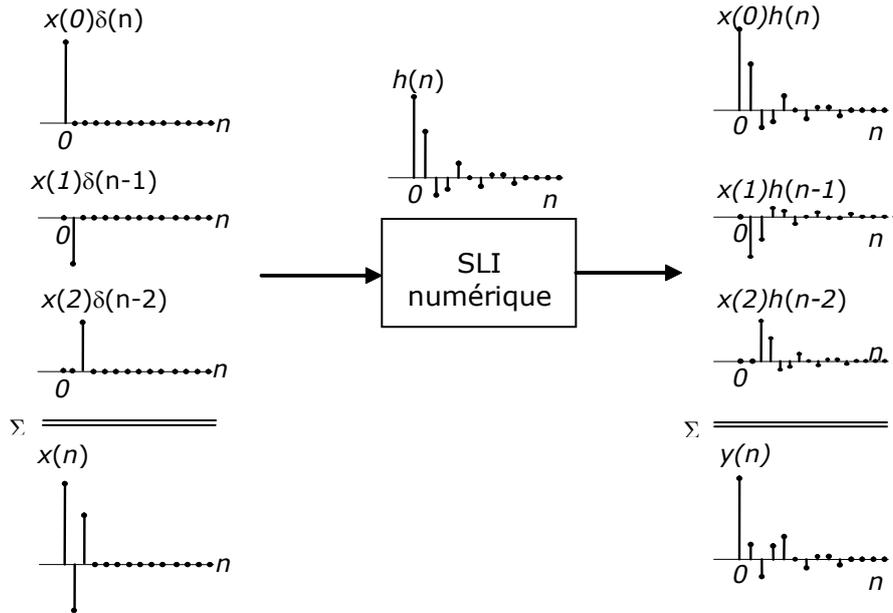


Fig. 1.14 Convolution numérique par somme de réponses impulsionnelles décalées et pondérées

Cette équation définit ce que l'on appelle une *convolution numérique*, notée <sup>7</sup> :

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i) \tag{1.12}$$

**Exemple 1.9**

Considérons à nouveau le système de Fibonacci et supposons que le propriétaire de l'enclos y apporte, non pas un seul couple de lapins le premier mois, mais bien : un couple le premier mois, deux couples le troisième, et un couple le quatrième mois. On demande combien de couples se trouveront dans l'enclos après sept mois.

Il est facile de lancer le calcul de la récurrence sous Matlab :

```
resp=filter([1],[1 -1 -1],[1,0,2,1,0,0,0])
1      1      4      6      10     16     26
```

On peut retrouver facilement ce résultat en utilisant la réponse impulsionnelle {1,1,2,3,5,8,13,21,34,55,89,144} : le premier couple de lapins produira 13 couples en sept mois, les deux couples placés au troisième mois en produiront 2\*5, et le quatrième en produira 3, ce qui fait bien en tout 26.

**Stabilité et réponse impulsionnelle.** Une des conséquences importantes de l'équation (1.12) est que la stabilité d'un filtre numérique peut bel et bien être vérifiée uniquement sur base de sa réponse impulsionnelle. On montre assez facilement (mais la démonstration n'a pas d'intérêt ici) que la condition de stabilité s'exprime comme :

---

<sup>7</sup> On notera que, dans cette dernière équation, la sommation est définie pour *i* allant de  $-\infty$  à  $+\infty$ , et non de 0 à  $+\infty$  comme on pourrait s'y attendre vu (1.11). Cette extension permet de prendre en compte les systèmes non causaux, dont la réponse impulsionnelle *h(n)* peut être définie pour *n* négatif.

$$\sum_{i=-\infty}^{\infty} |h(n-i)| < \infty \tag{1.13}$$

Dans la pratique, on ne mesure jamais que les premiers échantillons de la réponse impulsionnelle d'un système (ceux dont la valeur est supérieure à un seuil, typiquement 1e-3 ou 1e-4, et on ignore les autres), de sorte que cette condition se réduit à :

$$|h(i)| < \infty \quad (-\infty < i < \infty) \tag{1.14}$$

**Vision duale du produit de convolution.** Il est facile de montrer que le produit de convolution numérique ainsi défini est bien commutatif :

$$y(n) = x(n) * h(n) = \sum_{i=0}^{\infty} h(i)x(n-i) \tag{1.15}$$

Cette autre façon d'écrire le produit de convolution possède elle aussi une interprétation simple : pour obtenir une valeur particulière  $y(n_0)$  de la séquence  $y(n)$ , il suffit d'inverser  $h(n)$ , de repositionner  $h(0)$  sur  $x(n_0)$ , et de calculer le produit scalaire entre les séquences  $x(n)$  et  $h(n)$  ainsi définies. On constate ainsi qu'il est possible d'obtenir directement une valeur  $y(n_0)$  de la séquence  $y(n)$  par combinaison linéaire des valeurs de  $x(n)$  autour de  $x(n_0)$ , combinaison linéaire dont les coefficients sont les valeurs de  $h(n)$  (Fig. 1.16).

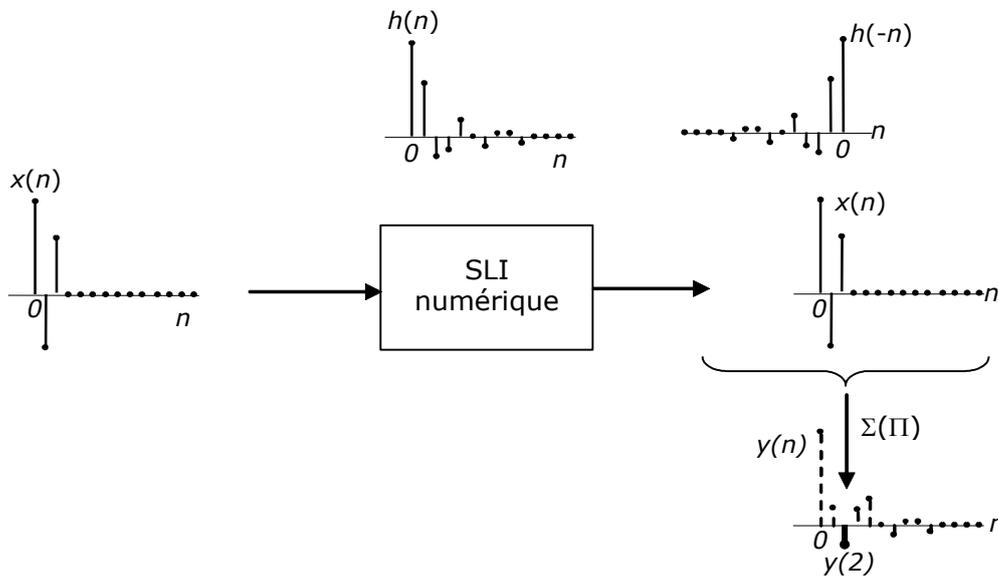


Fig. 1.15 Convolution numérique par retournement de la réponse impulsionnelle et somme pondérée des échantillons d'entrée : calcul de  $y(2)$

Pour rappel, l'équation de récurrence (1.4) nécessite au contraire le calcul de toutes les valeurs de  $y(n)$  précédant  $y(n_0)$ <sup>8</sup>. Ce calcul est réalisé ici indirectement par le calcul préliminaire de  $h(n)$ .

**Exemple 1.10**

Soit  $\{x(n)\}=n$  et  $\{h(n)\}= \{1,1/2,1/3,1/4,0,0,\dots\}$ . Le calcul de  $y(6)$  par (1.15) donne 10.5833. Cette valeur correspond à  $6+5/2+4/3+3/4$  (Fig. 1.16).

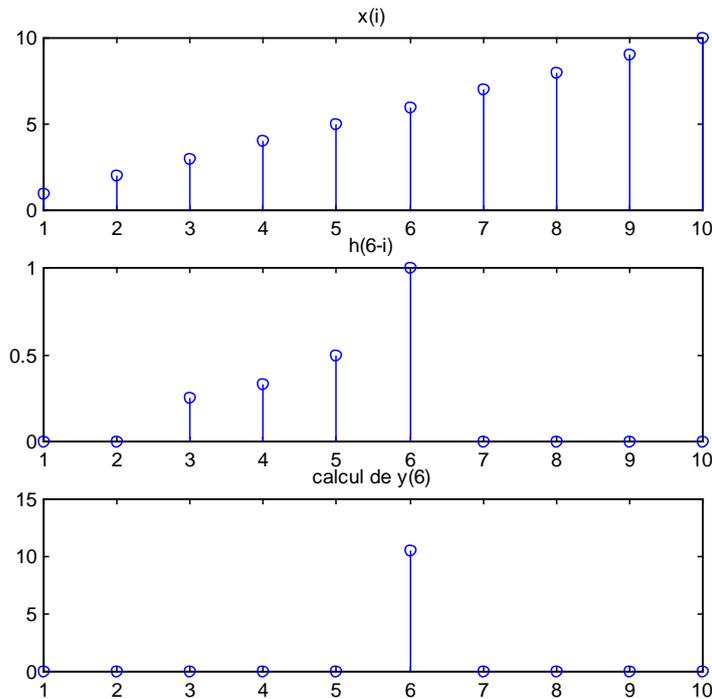


Fig. 1.16 Calcul de  $y(6)$  par combinaison linéaire des valeurs de  $x$  autour de  $x(6)$

**1.6 Transformée en Z – Pôles et zéros d’un SLI**

Les opérations effectuées pour réaliser la convolution entre deux séquences ne sont pas sans rappeler celles qui permettent de calculer un produit de polynômes. Le produit de deux polynômes se réduit en effet à une somme de produits des coefficients du premier par les coefficients du second, décalés des puissances correspondantes. On montre facilement que ces deux opérations sont exactement identiques : la convolution entre deux séquences numériques est exactement équivalente au produit de deux polynômes dont les coefficients sont précisément les séquences à convoluer.

**Exemple 1.11**

---

<sup>8</sup> Sauf si le système est non-récursif, auquel cas on a déjà vu que sa réponse impulsionnelle est égale à la suite de ses coefficients. Le calcul de  $y(n_0)$  par (1.5) est alors exactement identique à (1.15).

Soit  $\{x(n)\}=\{1,2,3,0,0,\dots\}$  et  $\{h(n)\}=\{2,3,4,0,0,\dots\}$ . La convolution de  $\{x(n)\}$  avec  $\{h(n)\}$  par (1.12) donne  $\{2,7,16,17,12,0,0,\dots\}$ . Construisons deux polynômes  $X(z)$  et  $H(z)$  à partir de  $\{x(n)\}$  et  $\{h(n)\}$  :

$$X(z)=1+2z^{-1}+3z^{-2}=z^{-2}(z^2+2z+3) \quad H(z)=2+3z^{-1}+4z^{-2}=z^{-2}(2z^2+3z+4)$$

Le produit de ces deux polynômes donne bien le polynôme  $Y(z)=z^{-4}(2z^4+7z^3+16z^2+17z+12)$

On peut également le vérifier par Matlab :

```
conv([2,3,4],[1 2 3])
ans =
     2     7    16    17    12
```

On comprend mieux maintenant pourquoi, dans Matlab, la fonction qui réalise le produit de deux polynômes s'appelle `conv`.

L'exemple précédent a montré qu'il est intéressant d'associer à une séquence numérique  $\{x(n)\}$  un polynôme en  $z^{-1}$  (où  $z$  est la variable complexe), que l'on appelle *transformée en Z* de  $\{x(n)\}$  et que l'on note  $X(z)$  :

$$\{x(n)\} \stackrel{Z}{\Leftrightarrow} X(z) = \sum_{i=-\infty}^{\infty} x(i)z^{-i} \tag{1.16}$$

Si  $H(z)$  est la transformée en  $z$  de la réponse impulsionnelle  $\{h(n)\}$  d'un SLI numérique :

$$\{h(n)\} \stackrel{Z}{\Leftrightarrow} H(z) = \sum_{i=-\infty}^{\infty} h(i)z^{-i} \tag{1.17}$$

La transformée en  $z$  de la sortie du système est alors donnée par :

$$y(n) \stackrel{Z}{\Leftrightarrow} Y(z) = X(z)H(z) \tag{1.18}$$

On voit donc (pour ceux qui ont eu un cours sur le sujet) que la transformée en  $z$  (qui associe une fonction de la variable complexe  $z$  à un signal numérique) est l'homologue, dans le domaine numérique, de la transformée de Laplace (qui associe une fonction de la variable complexe  $p$  à un signal analogique). On avait en effet :

$$\begin{aligned} x(t) \stackrel{L}{\Leftrightarrow} X(p) &= \int_{-\infty}^{\infty} x(t)e^{-pt} dt \\ h(t) \stackrel{L}{\Leftrightarrow} H(p) &= \int_{-\infty}^{\infty} h(t)e^{-pt} dt \\ y(t) \stackrel{L}{\Leftrightarrow} Y(p) &= X(p)H(p) \end{aligned} \tag{1.19}$$

Le passage de l'analogique au numérique correspond au passage de l'intégrale à la somme, et la substitution de  $z$  à  $e^p$ .

**Transformée en Z de signaux usuels.** On peut dresser rapidement une liste des valeurs de  $X(z)$  pour certains signaux simples. Ainsi par exemple:

$$\{a^n \varepsilon(n)\} \stackrel{Z}{\Leftrightarrow} \sum_{i=0}^{\infty} a^i z^{-i} = \frac{1}{1-az^{-1}} \tag{1.20}$$

L'exponentielle décroissante  $a^n$  ( $|a|<1$ ) correspond donc (de façon similaire à ce qui se passe en analogique) à un pôle réel, en  $z=a$  (Fig. 1.17) et un zéro en 0.

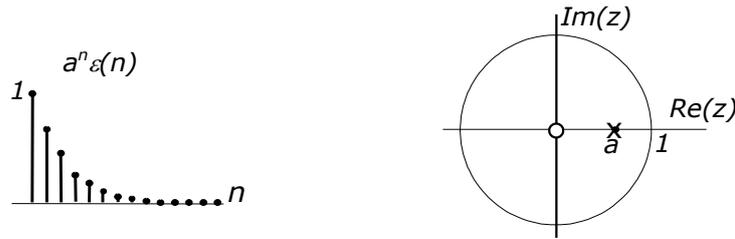


Fig. 1.17 Exponentielle numérique réelle et son pôle réel dans le plan z

Cette conclusion reste valable si  $a$  est un nombre complexe  $\rho e^{j\varphi}$  :

$$\{\rho^n e^{jn\varphi} \varepsilon(n)\} \stackrel{Z}{\Leftrightarrow} \sum_{i=0}^{\infty} \rho^i e^{ji\varphi} z^{-i} \stackrel{|\rho z^{-1}| < 1}{=} \frac{1}{1 - \rho e^{j\varphi} z^{-1}} \tag{1.21}$$

On en déduit qu'un cosinus amorti correspond à deux pôles complexes conjugués et deux zéros (un en zéro et l'autre sur l'axe réel, à la verticale des pôles) :

$$\begin{aligned} \{\rho^n \cos(n\varphi) \varepsilon(n)\} &\stackrel{Z}{\Leftrightarrow} \sum_{i=0}^{\infty} \rho^i \cos(i\varphi) z^{-i} = \sum_{i=0}^{\infty} \frac{\rho^i}{2} (e^{ji\varphi} + e^{-ji\varphi}) z^{-i} \\ &\stackrel{\rho < 1}{=} \frac{1/2}{1 - \rho e^{j\varphi} z^{-1}} + \frac{1/2}{1 - \rho e^{-j\varphi} z^{-1}} \\ &= \frac{1 - \rho \cos(\varphi) z^{-1}}{1 - 2\rho \cos(\varphi) z^{-1} + \rho^2 z^{-2}} \end{aligned} \tag{1.22}^9$$

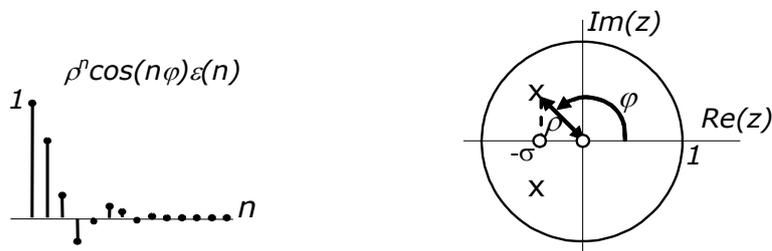


Fig. 1.18 Sinusoïde numérique amortie et pôles complexes conjugués dans le plan z

**Fonction de réponse opérationnelle (en z) d'un SLI numérique.** La transformée en z est cependant largement moins indispensable en numérique que son homologue de Laplace ne l'est en analogique. La récurrence (1.4), qui se substitue à l'équation intégral-différentielle décrivant un SLI analogique, permet en effet le calcul direct de la sortie, sans devoir passer par le calcul de sa transformée en z.

Comme la transformée de Laplace, la transformée en z ne converge pas nécessairement pour toutes les valeurs de z (on parle de *domaine de*

<sup>9</sup> Notons que le dénominateur de cette dernière expression est similaire à celle obtenue en analogique:  $1 - 2\rho \cos(\varphi) z^{-1} + \rho^2 z^{-2} = z^{-2} (z^2 + 2\sigma z + \rho^2)$  si l'on pose  $\sigma = -\rho \cos(\varphi)$ , c'est-à-dire si  $-\sigma$  est la partie réelle des pôles.

convergence; voir par exemple la condition apparue en (1.20)). Elle jouit également de nombreuses propriétés (que nous ne verrons pas ici in extenso).

La seule propriété fondamentale de la transformée en  $z$  que nous utiliserons ici est celle liée au retard de  $\{x(n)\}$  de  $n_0$  échantillons:

$$\{x(n-n_0)\} \Leftrightarrow \sum_{i=-\infty}^{\infty} x(i-n_0)z^{-i} = \sum_{k=-\infty}^{\infty} x(k)z^{-(k+n_0)} = z^{-n_0} \sum_{k=-\infty}^{\infty} x(k)z^{-k} = z^{-n_0} X(z) \quad (1.23)$$

On peut en effet avantageusement appliquer cette propriété à la récurrence (1.4) :

$$Y(z) + Y(z) \sum_{i=1}^N a_i z^{-i} = X(z) \sum_{i=0}^M b_i z^{-i} \quad (1.24)$$

ce qui conduit à une autre formulation de  $H(z)$  :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \quad (1.25)$$

que l'on réécrit parfois sous la forme plus simple :

$$H(z) = \frac{\sum_{i=0}^M b_i z^{-i}}{\sum_{i=0}^N a_i z^{-i}} \quad (a_0 = 1) \quad (1.26)$$

Cette dernière équation définit la fonction de transfert en  $z$  d'un SLI, et est à comparer à l'équation donnant la forme de la fonction de transfert en  $p$  d'un système analogique. On définit les *zéros* et les *pôles* d'un SLI numérique comme les racines du numérateur et du dénominateur de  $H(z)$ .

**Pôles et stabilité.** L'équation (1.18) permet alors, si l'on connaît  $X(z)$  et  $H(z)$ , de calculer la sortie  $Y(z)$  d'un SLI numérique, et d'en déduire  $\{y(n)\}$ . On peut d'ailleurs également utiliser pour ce faire la décomposition en fractions simples (en pratique, on ne le fait presque jamais: on obtient plus rapidement la réponse à partir de la récurrence du filtre). On trouve alors, comme en analogique, que la réponse est en général une somme pondérée de termes de régime dont la forme dépend de l'excitation, et de termes transitoires dont l'allure ne dépend que de la position des pôles de  $H(z)$ .

En conséquence, on peut établir un lien entre la position des pôles d'un système numérique et sa stabilité: un SLI numérique est strictement stable si ses pôles sont tous à l'intérieur du cercle de rayon unité (cercle non compris), et stable si on accepte aussi les pôles simples<sup>10</sup> sur le cercle de rayon unité. Cette propriété vient directement du fait que la réponse impulsionnelle du système peut être obtenue par décomposition en fractions simples de  $H(z)$ , et inversion de chacun

---

<sup>10</sup> Rappelons qu'en analogique un pôle multiple correspond à un terme  $t \exp(-\sigma t) \cos(\omega t)$ . Un pôle multiple sur l'axe imaginaire conduit donc à une réponse non bornée. Pour les mêmes raisons, un pôle multiple sur le cercle de rayon unité correspond à un système numérique instable.

des termes. Si tous les pôles sont à l'intérieur du cercle de rayon unité, tous ces termes seront de forme décroissante<sup>11</sup>.

Le demi-plan de droite de l'analogique se trouve ainsi transformé en l'intérieur du cercle de rayon unité, ce qui correspond à la substitution de  $z$  à  $e^p$  déjà évoquée plus haut (Fig. 1.19).

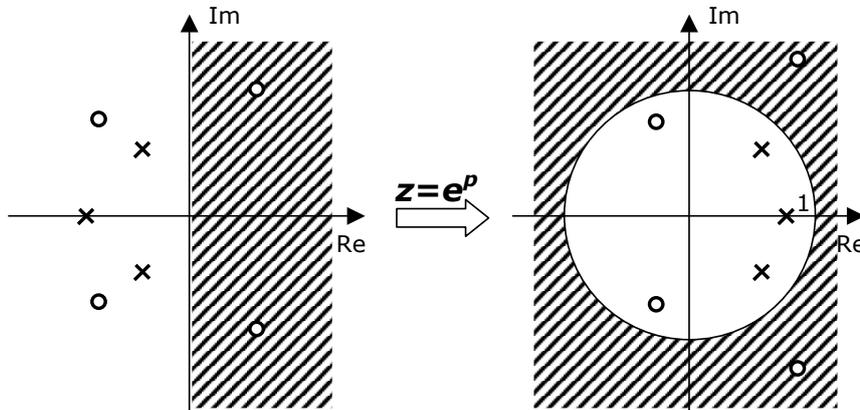


Fig. 1.19 Zone de stabilité en  $z$  et lien avec la zone de stabilité en  $p$ .

**Exemple 1.12**

Nous avons déjà vu que le système de Fibonacci est instable. Nous pouvons maintenant le vérifier sur sa fonction de transfert :

$$H(z) = \frac{1}{1 - z^{-1} - z^{-2}}$$

Les pôles de  $H(z)$  sont réels et égaux à  $(1 \pm \sqrt{5})/2$ . Il y a deux zéros en 0, qui n'ont pas d'influence sur le module de la réponse en fréquence. On peut les afficher facilement sous Matlab, qui les positionne automatiquement par rapport au cercle de rayon unité :

```
zplane([1], [1 -1 -1])
```

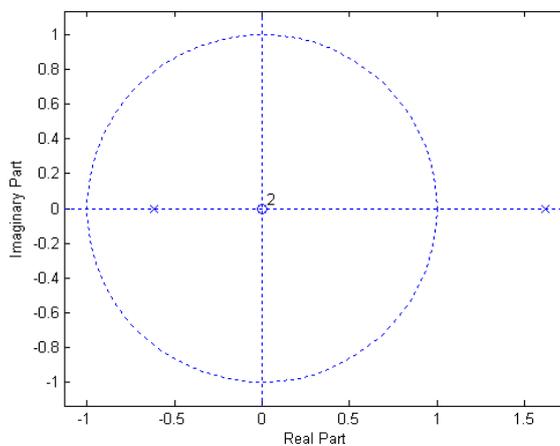


Fig. 1.20 Affichage des pôles et zéros par Matlab

<sup>11</sup> La démonstration stricte de cette propriété n'est cependant pas simple.

**Exemple 1.13**

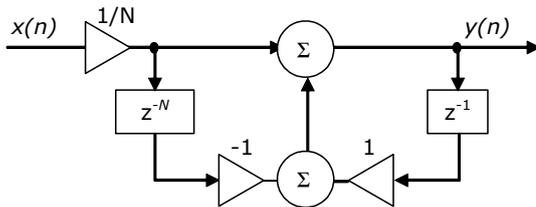
L'intérêt de l'expression d'un filtre numérique sous sa forme  $H(z)$  (1.26) est également de permettre de manipulations de la structure du filtre sans changer sa réponse impulsionnelle. Ainsi, le filtre à moyenne mobile de l'Exemple 1.8 possède une forme récursive plus simple que sa forme non récursive:

$$H(z) = \frac{1}{N} \sum_{i=0}^{N-1} z^{-i} = \frac{1}{N} \frac{1 - z^{-N}}{1 - z^{-1}}$$

à laquelle il correspond la récurrence :

$$y(n) - y(n-1) = \frac{1}{N} (x(n) - x(n-N))$$

et la structure :



On peut vérifier que les réponses impulsionnelles de ces deux filtres sont identiques. La forme récursive est plus compacte à calculer, mais elle nécessite, pour le calcul d'un échantillon donné  $y(n_0)$  de la sortie, de calculer tous les échantillons de sortie précédents.

Notons qu'il s'agit ici d'un cas assez rare (celui d'une structure récursive qui possède un équivalent exact non récursif). En général, un filtre récursif possède une réponse impulsionnelle longue, qui ne s'annule théoriquement jamais mais devient en pratique négligeable au bout d'un temps fini. On pourrait dès lors le remplacer par son équivalent non récursif obtenu en tronquant la réponse impulsionnelle du filtre récursif. On ne le fait en général pas, car le degré du filtre non récursif (et donc la charge de calcul associée) est systématiquement très supérieur à celui du filtre récursif.

**Exercices**

**Exercice 1.1**

Un SLI est décrit par l'équation :

$$y(n) + a_1 y(n - 1) = b_0 x(n) + b_2 x(n - 2)$$

On demande :

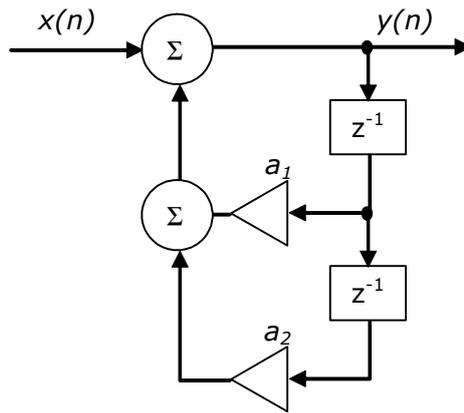
- De dessiner la structure du filtre correspondant
- De trouver l'expression *analytique* de sa réponse impulsionnelle
- D'en déduire une condition pour que le filtre soit stable
- De vérifier cette condition à partir de la fonction de transfert en  $z$  du système

Solution

$$h(n) = h_1(n) + h_2(n) = b_0 (-a_1)^n \varepsilon(n) + b_2 (-a_1)^{n-2} \varepsilon(n-2), \text{ stable si } |a_1| \leq 1$$

**Exercice 1.2**

Le système suivant constitue, pour certaines valeurs de ses coefficients, un *oscillateur numérique*. Il implémente un calcul purement récursif de  $\cos(x)$ .



On demande :

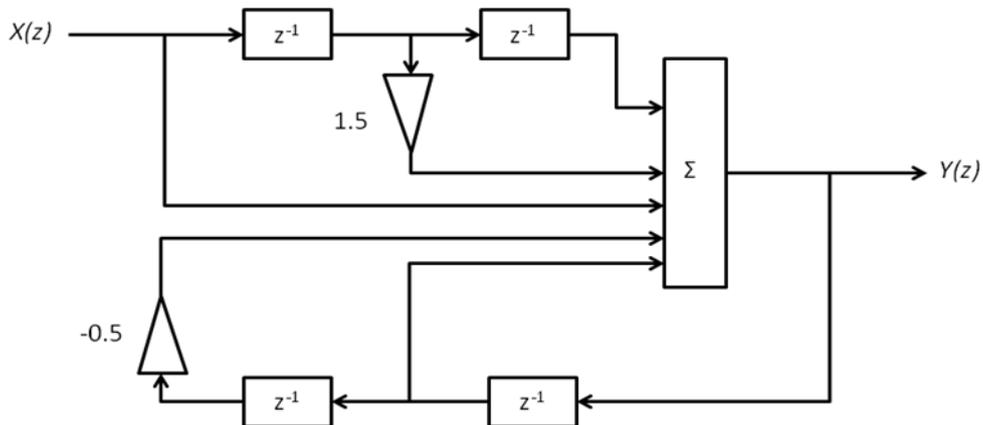
- D'écrire la récurrence numérique correspondante
- De trouver les valeurs de  $a_1$  et  $a_2$  permettant à ce filtre d'avoir une réponse impulsionnelle de type  $\rho^n \cos(n\varphi)$  pulsation réduite  $\varphi$  et d'amortissement nul (c.à-d. ayant un  $\rho$  unitaire)
- De calculer les pôles de ce système et de les dessiner dans le plan complexe

Solution

$$a_1 = -2\cos(\varphi); a_2 = 1 ; \text{ pôles complexes} = (e^{j\varphi}, e^{-j\varphi})$$

**Exercice 1.3**

Un système du second ordre est décrit par le diagramme suivant :



On demande :

- De déterminer la fonction de transfert  $H(z) = \frac{Y(z)}{X(z)}$  du système ainsi que l'équation aux différences correspondante
- De calculer les 10 premiers échantillons de la réponse impulsionnelle du système en utilisant l'équation aux différences (le système est supposé initialement au repos)
- De dessiner le diagramme pôles-zéros de ce système.

Solution

...  $h(9) = .1563$ ; zéros =  $\exp[+ -j \arccos(-0.75)]$ ; pôles =  $\sqrt{.5} \exp[+ -j \pi/4]$

**Exercice 1.4**

Un système numérique est décrit par l'équation aux différences :

$$y(n) + 0.9 y(n - 2) = 0.3 x(n) + 0.6 x(n - 1) + 0.3 x(n - 2)$$

On demande :

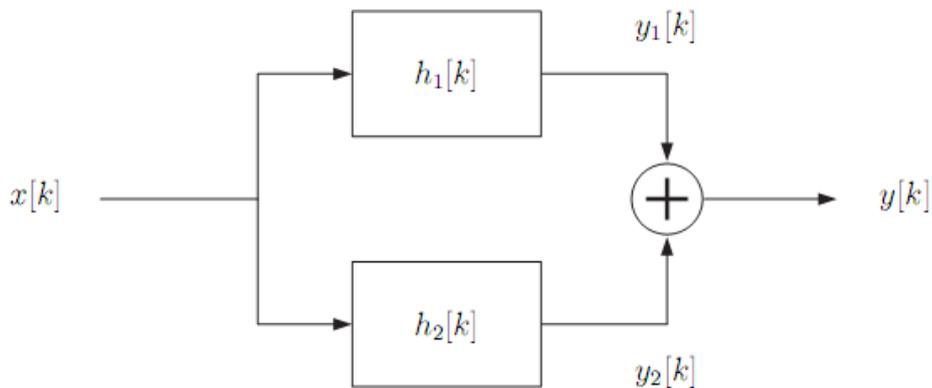
- De calculer la fonction de transfert du système  $H(z) = \frac{Y(z)}{X(z)}$
- De calculer les 5 premiers échantillons de la réponse du système à  $x(n) = \delta(n)$  (pour les conditions initiales  $y(-2)=y(-1)=0$ )
- De donner la pulsation réduite  $\varphi$  et le coefficient d'amortissement  $\rho$  de la réponse impulsionnelle.

Solution

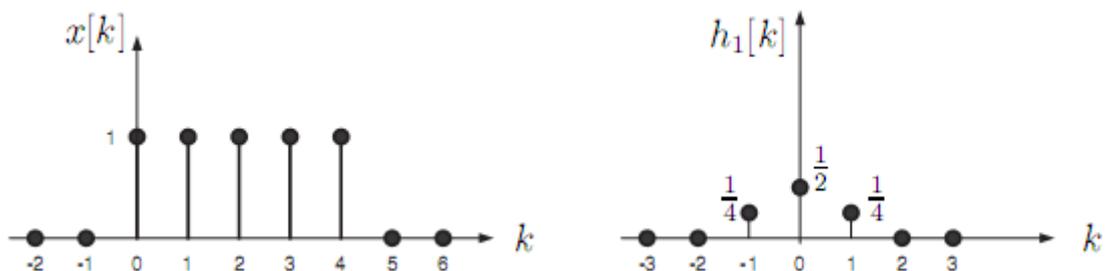
...  $h(4) = -0.0027$ ;  $\rho = \sqrt{0.9}$ ;  $\varphi = \pi/2$

**Exercice 1.5**

Un système numérique est constitué de la mise en parallèle de deux systèmes numériques :



Le signal d'entrée  $x(k)$  ainsi que la réponse impulsionnelle  $h_1(k)$  sont fournis :



On demande :

- De tracer le signal  $y_1(k)$  pour  $k$  compris entre -2 et 6
- De déterminer  $h_2(k)$  tel que la sortie  $y(k)$  du système soit identique à l'entrée  $x(k)$

Solution

...  $h_2(-1) = -0.25$ ;  $h_2(0) = 0.5$ ;  $h_2(1) = -0.25$ ;

**Exercice 1.6**

Soit les signaux  $x_1(n)$  et  $x_2(n)$  :

$$x_1(n) = \{-1, 2, 1\} \quad 0 \leq n \leq 2$$

$$x_2(n) = 1 \quad 0 \leq n \leq 5$$

$$x_2(n) = 0 \quad \text{ailleurs}$$

On demande :

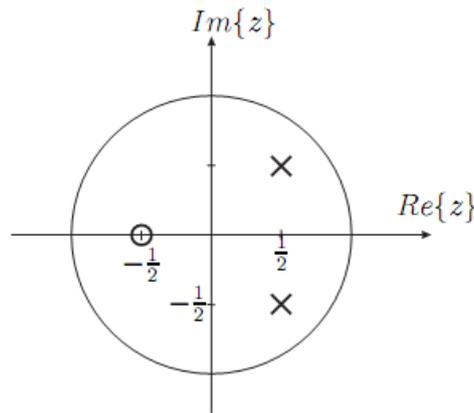
- De calculer les 3 premiers échantillons de leur convolution numérique :  $y(0)$ ,  $y(1)$ ,  $y(2)$ .
- De calculer le 8<sup>ème</sup> échantillon *sans passer par le calcul des échantillons précédents*.

Solution

...  $y(2) = 2$ ;  $y(7) = 1$

**Exercice 1.7**

Un système numérique est caractérisé par le diagramme pôles-zéros :

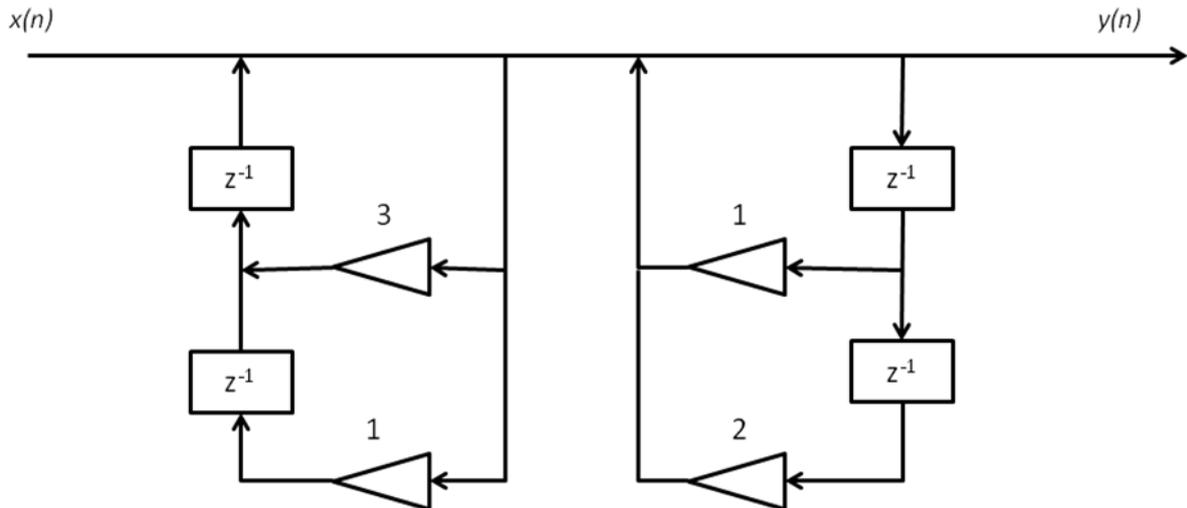


On demande :

- De déterminer la fonction de transfert du système (à une constante près) à partir du diagramme pôles-zéros
- D'en déduire l'équation aux différences décrivant le système
- De dessiner la structure du filtre correspondant

**Exercice 1.8**

Un système numérique est décrit par la structure suivante :



On demande de :

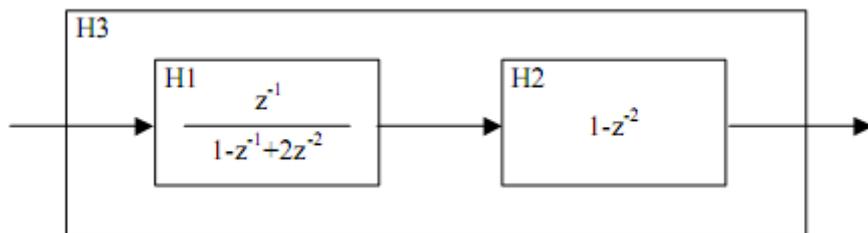
- Déterminer l'équation aux différences correspondant à cette structure
- De déterminer la fonction de transfert  $H(z)$  du système
- De déterminer combien de multiplications et d'additions sont nécessaires selon cette structure pour obtenir chaque échantillon de sortie (on fait comme hypothèse que  $x(n)$  est réel et qu'une multiplication par 1 n'est pas comptée).

Solution

... 2 produits et 4 sommes

**Exercice 1.9**

A partir de leurs récurrences respectives, calculer les 5 premiers échantillons de la réponse impulsionnelle des filtres ci-dessous  $H_1$ ,  $H_2$ ,  $H_3$  et vérifier que la réponse de  $H_3$  est bien la convolution numérique de celle de  $H_1$  et  $H_2$ .



Solution

...  $h_3(n) = \{0, 1, 1, -2, -4, \dots\}$

**Exercice 1.10**

Discuter de la stabilité du système donné par la récurrence suivante en fonction de  $k$  (réel).

$$y(n) = y(n - 1) + k^2 y(n - 2) + x(n) - 3x(n - 3)$$

Solution

*toujours instable)*

**Exercice 1.11**

a. Donner la fonction de transfert en z correspondant à la récurrence suivante, étudier sa stabilité, et proposer une structure pour le SLI correspondant.

$$y(n) = x(n) + 2x(n-1) + y(n-1) + 4y(n-2)$$

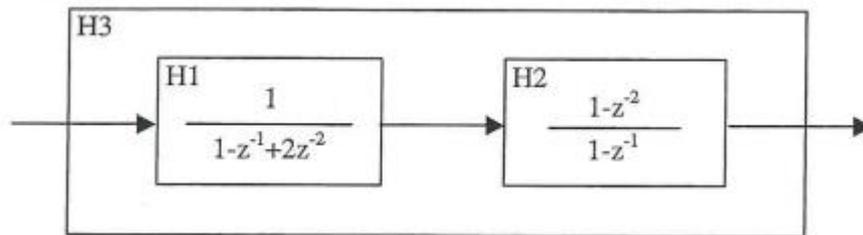

b. Donner l'équation de récurrence du SLI *inverse* (celui qui produirait x(n) en sortie si on lui présentait y(n) en entrée) et proposer une structure.

Solution

*a. instable b...*

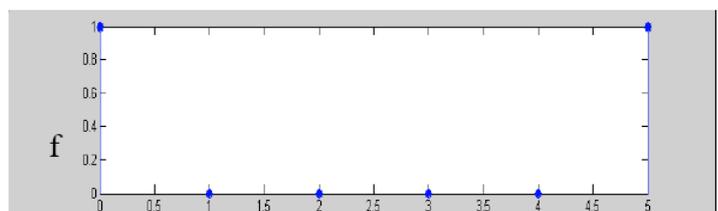
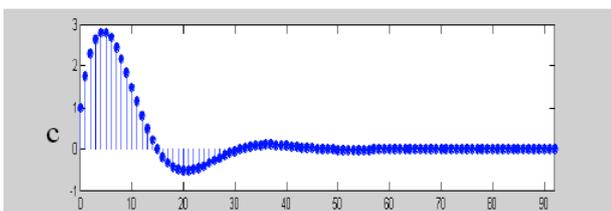
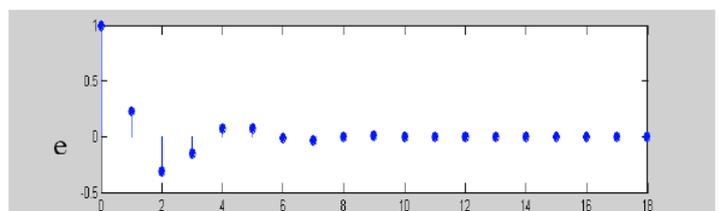
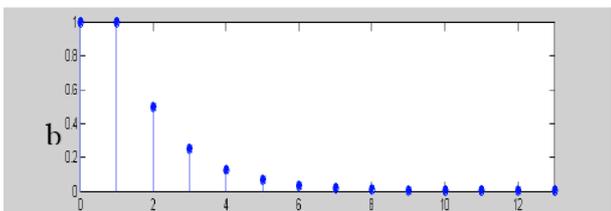
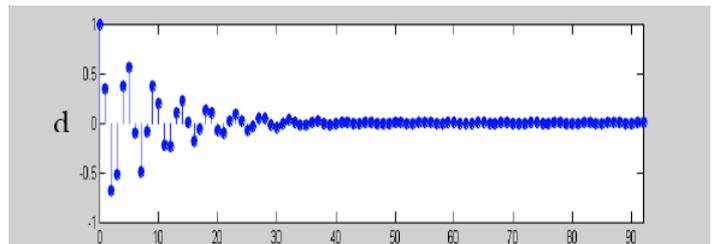
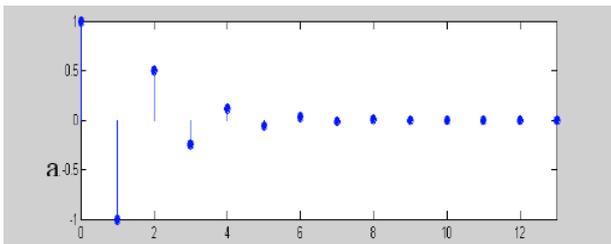
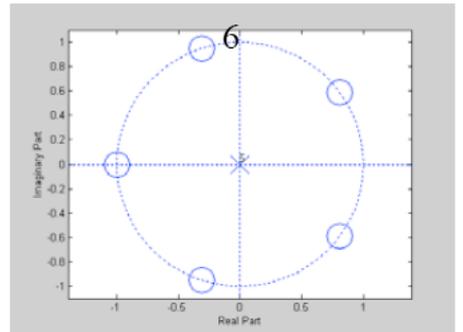
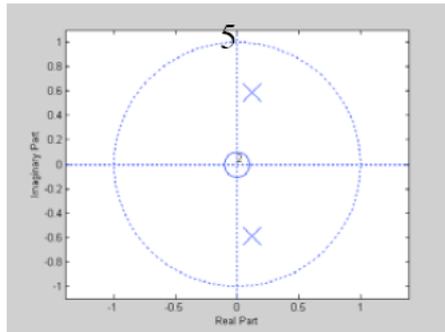
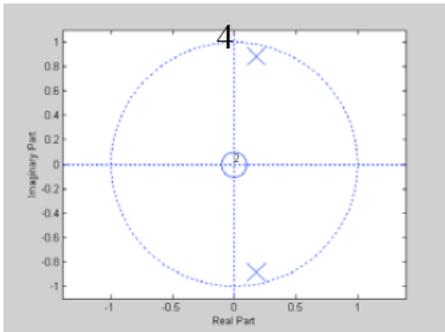
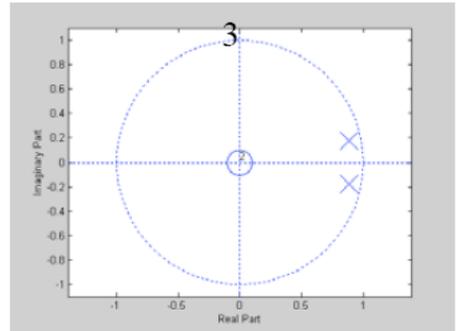
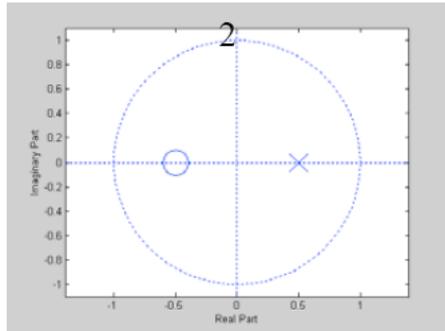
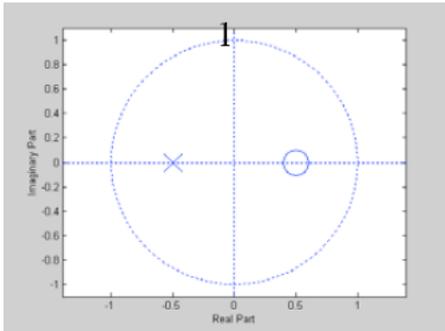
**Exercice 1.12**

A partir de leurs récurrences respectives, calculer les 5 premiers échantillons de la réponse impulsionnelle des filtres ci-dessous H<sub>1</sub>, H<sub>2</sub>, H<sub>3</sub> et vérifier que la réponse de H<sub>3</sub> est bien la convolution numérique de celle de H<sub>1</sub> et H<sub>2</sub>.



**Exercice 1.13**

Etablir les correspondances entre les diagrammes pôles zéros et les réponses impulsionnelles suivants et **justifier** vos choix :



# CHAPITRE 2

## ANALYSE FREQUENTIELLE DES SIGNAUX ET SYSTEMES NUMERIQUES

Witold Hurewicz (1904-1956). Mathématicien russe, spécialiste en topologie. Emigré aux Etats-Unis en 1936, il décida d'y rester et devint professeur au MIT, de 1945 à sa mort. Durant la seconde guerre mondiale, il participa à l'analyse des signaux radar, intrinsèquement discrets puisque la position des objets y est connue une seule fois par tour d'antenne. C'est dans ce contexte qu'il inventa en 1947 ce que nous connaissons aujourd'hui sous le nom de *transformée en Z*.



Le chapitre 1 a montré l'essentiel de ce qui peut être dit des signaux et systèmes numériques (linéaires et invariants) dans le domaine temporel.

L'analyse fréquentielle des signaux numériques nécessite d'introduire la transformée de Fourier à temps discret (TFTD), homologue de la transformée de Fourier dans le monde numérique. Nous le ferons ici après un rappel sur l'analyse fréquentielle des signaux analogiques, par souci de cohérence.

Nous pourrions alors définir la réponse en fréquence d'un système numérique, comme facteur multiplicatif intervenant dans le calcul de sa réponse en régime sinusoïdal. Nous verrons que ce facteur multiplicatif peut être obtenu à partir de la fonction de transfert en  $Z$  du système. Nous terminerons par le calcul pratique de quelques filtres simples.

### 2.1 Signaux simples usuels

On utilisera souvent dans ce cours les signaux ci-dessous, dont nous donnons ici les versions analogiques et numériques (et l'implémentation MATLAB de ces dernières).

#### Echelon unité

L'échelon unité, noté  $\varepsilon(t)$  en temps continu (Fig. 2.1) ou  $\varepsilon(n)$  en temps discret, est défini par :

$$\varepsilon(t) = \begin{cases} 1 & (t \geq 0) \\ 0 & (t < 0) \end{cases} \quad \varepsilon(n) = \begin{cases} 1 & (n \geq 0) \\ 0 & (n < 0) \end{cases} \quad (2.1)$$

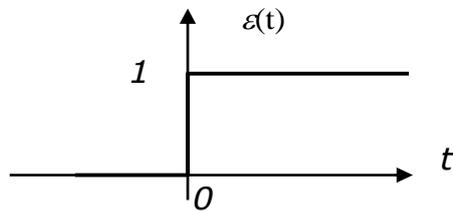


Fig. 2.1 Signal échelon à temps continu.

### Rectangle

Le signal rectangulaire  $rect(t)$  est constant  $-1/2$  à  $+1/2$ , d'amplitude 1.

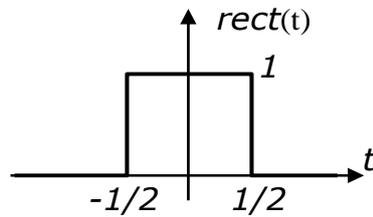
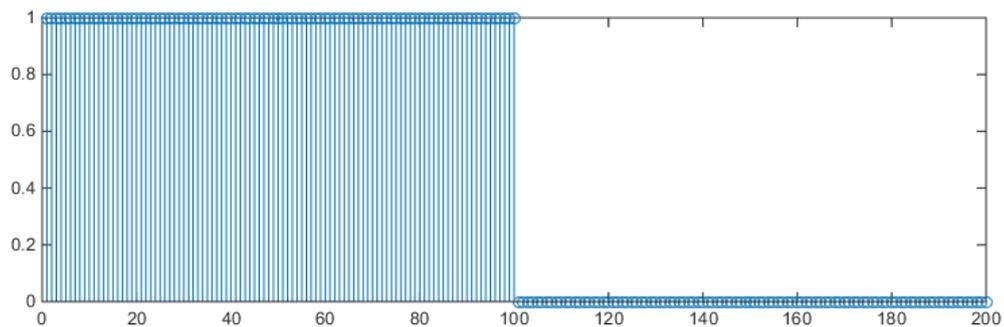


Fig. 2.2 Signal rectangulaire à temps continu.

### Exemple 2.1 – Signal rectangulaire à temps discret

```
rect=[ones(1,100) zeros(1,100)]; %% 1s of a rectangle sampled at 100Hz
stem(rect);
```



### Triangle

Le signal triangulaire est défini par :

$$tri(t) = \begin{cases} 1-|t| & (|t| \leq 1) \\ 0 & \text{sinon} \end{cases} \quad (2.2)$$

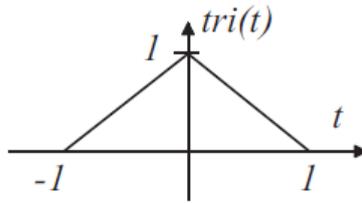


Fig. 2.3 Signal triangulaire à temps continu.

### Sinus cardinal

Le signal sinus cardinal (souvent appelé « fonction pieuvre ») est donné par :

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \quad (2.3)$$

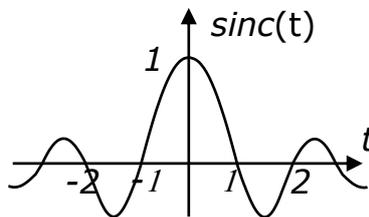
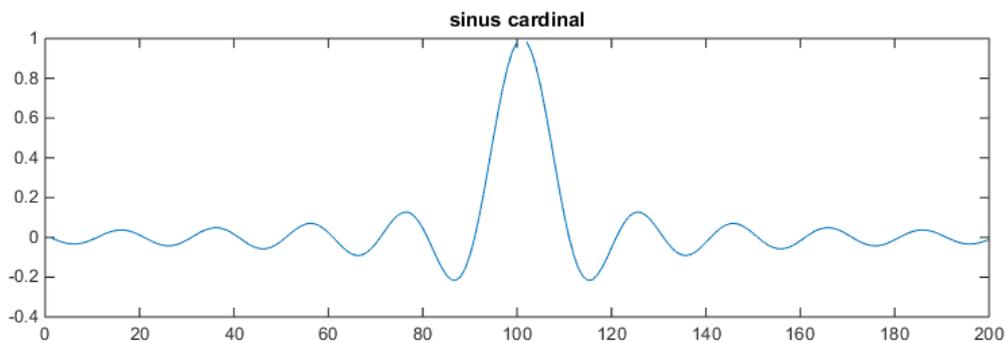


Fig. 2.4 Sinus cardinal

### Exemple 2.2 – Sinus cardinal à temps discret

```
Fe=100 ; t=(-100:99)* 10 /Fe;      %Fe = 100 ; 200 samples ; 10 oscillations
sincard=sin(pi*t) ./ (pi*t);
plot(sincard); title ('sinus cardinal');
```



### Exponentielle imaginaire - phaseur

L'exponentielle imaginaire  $\underline{A}e^{j\omega_0 t}$ , où  $\underline{A}$  est une constante complexe  $|\underline{A}|e^{j\angle \underline{A}}$  correspond, dans l'espace à 3 dimensions comprenant le plan complexe et un axe temporel orthogonal, à une hélice de rayon  $|\underline{A}|$ , de phase initiale  $\angle \underline{A}$ , et de vitesse angulaire  $\omega_0$ . Le sens de rotation dépend du signe de  $\omega_0$ , ce qui donne un sens à la notion de fréquence négative. On parle souvent de *phaseur* lorsqu'on considère la projection de ce signal sur le plan complexe.

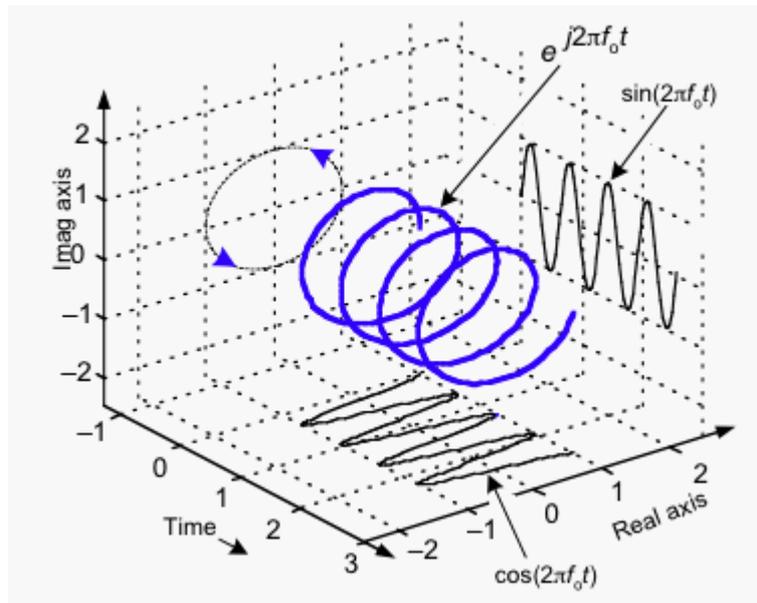


Fig. 2.5 Exponentielle imaginaire en 3D.

[<http://dsp.stackexchange.com/questions/431/what-is-the-physical-significance-of-negative-frequencies>]

### Signal sinusoïdal

Un signal sinusoïdal est de la forme générale  $a \cos(\omega_0 t + \phi)$ , ce qui comprend en particulier les formes  $a \sin(\omega_0 t)$  et  $a \cos(\omega_0 t)$ .  $a$  est la *valeur de crête*,  $\omega_0$  est la pulsation (en radians/s) et  $\phi$  est la phase à l'origine (en radians). La fréquence de la sinusoïde est donnée par  $f_0 = \omega_0 / 2\pi$  et la période par  $T_0 = 2\pi / \omega_0$ .

Ces signaux sont obtenus par projection d'un phaseur sur les axes réels ou imaginaires (Fig. 2.6).

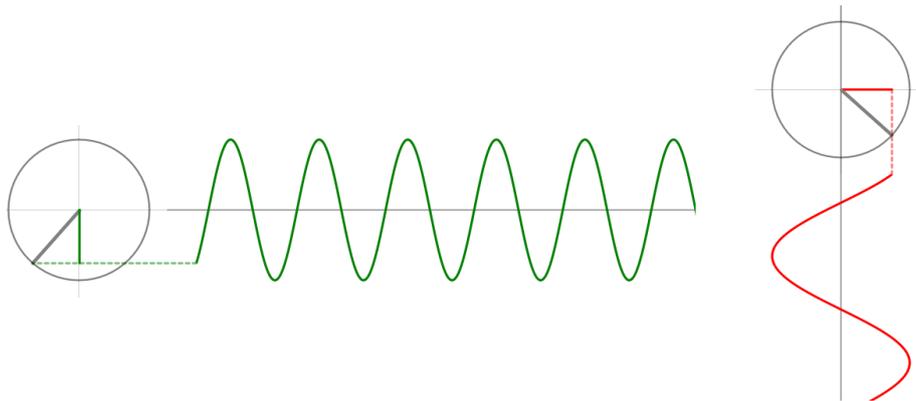


Fig. 2.6 Signaux sinusoïdaux [<http://jackschaedler.github.io/circles-sine-signals/sincos.html>]

La puissance de ce signal est donnée par :

$$P = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} |f_{T_0}(t)|^2 dt = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} a^2 |\cos(\omega_0 t + \phi)|^2 dt = a^2 / 2 \quad (2.4)$$

La valeur efficace de ce signal est donc donnée par  $a/\sqrt{2}$ .

**Exemple 2.3**

La Fig. 2.7 représente un signal sinusoïdal.

Il est facile d'en déduire l'expressions analytique :

$$U_{\max} = 14.1 \approx 10\sqrt{2} \quad ; \quad \omega_0 = 2\pi/T_0 = 2\pi/0.02 \approx 315 \text{ rad/s} ;$$

$$\frac{\theta}{2\pi} = \frac{\partial t}{T_0} \Rightarrow \theta = \frac{2\pi \cdot 0.0032}{0.02} \approx 1 \text{ rad/s}$$

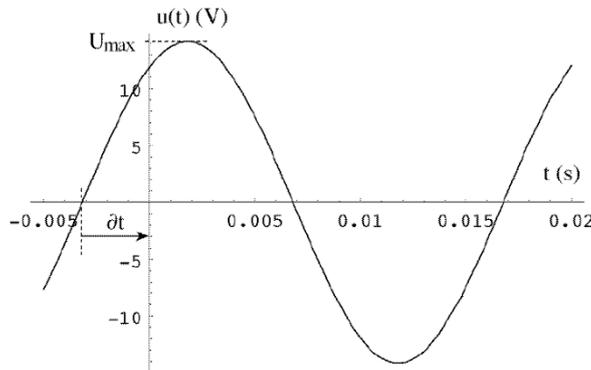


Fig. 2.7 Grandeur sinusoïdale

**Exponentielle complexe**

L'exponentielle complexe est définie par  $A e^{(\sigma+j\omega)t}$ . Elle peut être décomposée en trois facteurs :  $A$ , l'amplitude,  $e^{\sigma t}$ , où  $\sigma$  est lié à l'amortissement du signal, et  $e^{j\omega t}$  où  $\omega=2\pi f$ , et  $f$  est la fréquence de rotation de l'exponentielle imaginaire  $e^{j\omega t}$  autour de l'origine. Le signal est réellement amorti si  $\sigma$  est négatif. L'inverse de  $\sigma$  est noté  $\tau$  et appelé *constante de temps*.

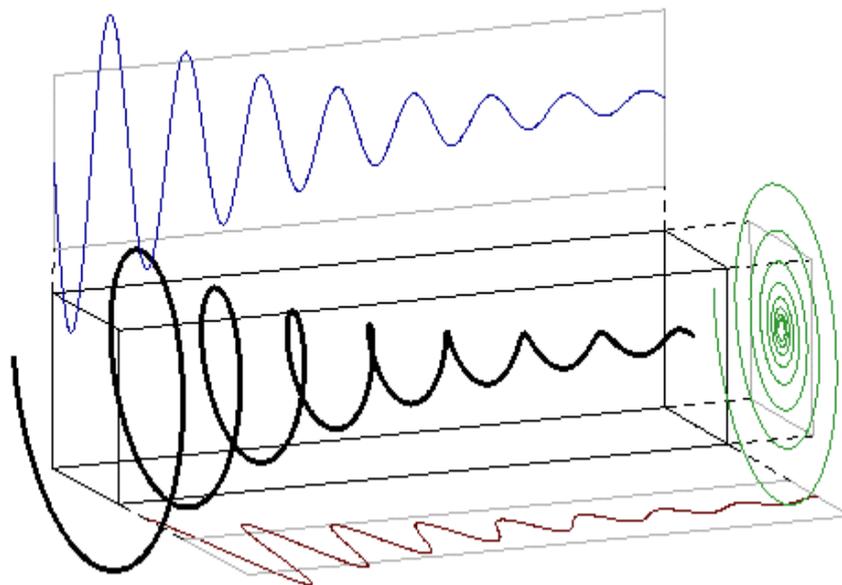


Fig. 2.8 Exponentielle complexe (avec amortissement :  $\sigma < 0$ )  
[\[https://www.quora.com/Laplace-Transforms/What-are-the-best-possible-](https://www.quora.com/Laplace-Transforms/What-are-the-best-possible-)

pedagogical-functions-examples-to-use-when-intuitively-describing-Laplace-Transforms-to-someone].

**Impulsion de Dirac**

Un des signaux discrets les plus simples est l'impulsion de Dirac  $\delta(n)$  définie par :

$$\delta(n) = \begin{cases} 1 & (n = 0) \\ 0 & (n \neq 0) \end{cases} \tag{2.5}$$

On peut donc extraire la valeur d'un signal  $f(n)$  à l'instant  $n=0$  en le multipliant (Fig. 2.9) par l'impulsion  $\delta(n)$  :

$$f(n)\delta(n) = f(0)\delta(n) \tag{2.6}$$

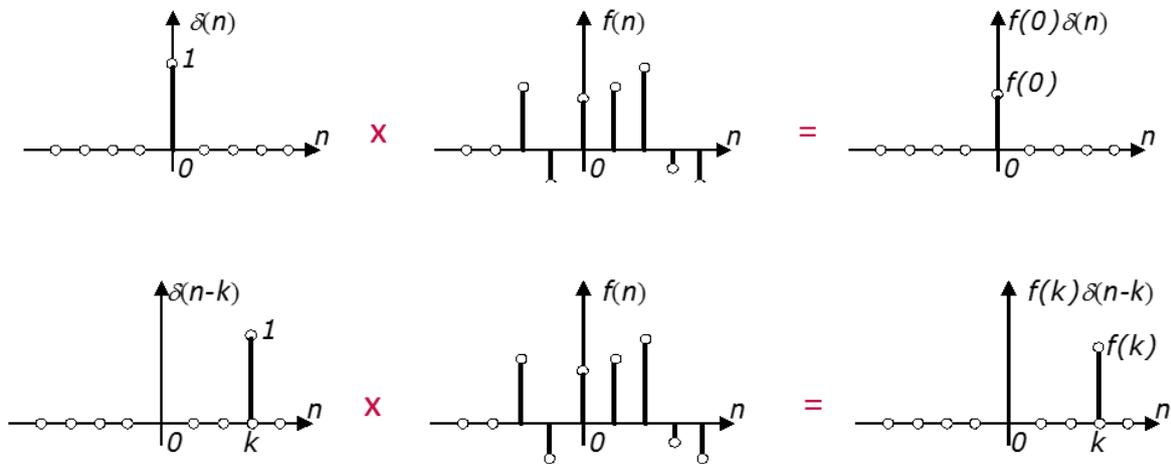


Fig. 2.9 Haut : produit de  $\delta(n)$  par un signal  $f(n)$  ; Bas : produit de  $\delta(n-k)$  par un signal  $f(n)$ .

Le résultat de cette opération est un signal qui vaut  $f(0)$  à l'instant  $n=0$  et est nul partout ailleurs.

En généralisant (2.6), on peut tout aussi bien extraire la valeur d'un signal  $f(n)$  à l'instant  $n=k$  en le multipliant (Fig. 2.9) par l'impulsion  $\delta(n-k)$ , qui ne vaut 1 qu'en  $n=k$  :

$$f(n)\delta(n-k) = f(k)\delta(n-k) \tag{2.7}$$

On peut donc finalement écrire que :

$$f(n) = \sum_{k=-\infty}^{\infty} f(k)\delta(n-k) \tag{2.8}$$

ce qui exprime que le signal  $f(n)$  est vu comme une somme pondérée d'impulsions de Dirac décalées dans le temps.

En particulier, l'échelon à temps discret  $\varepsilon(n)$  peut être décrit comme :

$$\varepsilon(n) = \sum_{k=0}^{\infty} \delta(n-k) \tag{2.9}$$

On peut également tirer de (2.6) que :

$$\sum_{-\infty}^{\infty} f(n)\delta(n) = \sum_{-\infty}^{\infty} f(0)\delta(n) = f(0)\sum_{-\infty}^{\infty} \delta(n) = f(0) \quad (2.10)$$

En particulier évidemment :

$$\sum_{-\infty}^{\infty} \delta(n) = 1 \quad (2.11)$$

L'analogie à temps continu de l'impulsion de Dirac est noté  $\delta(t)$  et devrait jouir des mêmes propriétés que  $\delta(n)$  (Fig. 2.10) :

$$f(t)\delta(t-\tau) = f(\tau)\delta(t-\tau) \quad (2.12)$$

$$f(t) = \int_{\tau=-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau \quad (2.13)$$

$$\varepsilon(t) = \int_{\tau=0}^{\infty} \delta(t-\tau)d\tau \quad (2.14)$$

$$\int_{-\infty}^{\infty} f(t)\delta(t) dt = f(0) \quad (2.15)$$

$$\int_{-\infty}^{\infty} \delta(t)dt = 1 \quad (2.16)$$

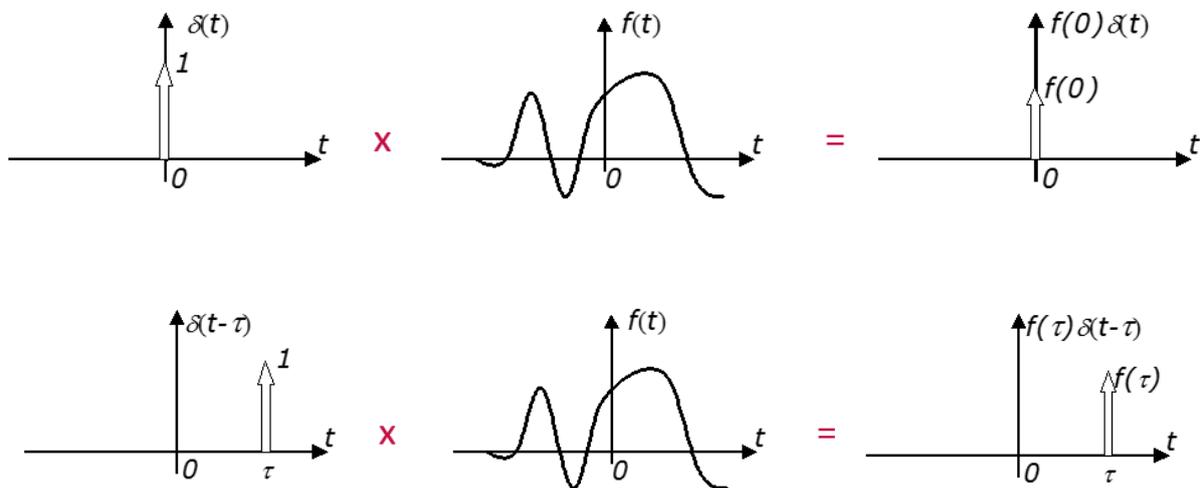


Fig. 2.10 Haut : produit de  $\delta(t)$  par un signal  $f(t)$  ; Bas : produit de  $\delta(t-\tau)$  par un signal  $f(t)$ .

La propriété (2.14) peut se réécrire, après changement de variable  $s=t-\tau$  :

$$\varepsilon(t) = \int_{-\infty}^t \delta(s)ds \quad (2.17)$$

ce qui implique que  $\varepsilon(t)$  soit la primitive de  $\delta(t)$ , et donc que  $\delta(t)$  soit la dérivée de  $\varepsilon(t)$ . Mathématiquement, ceci pose problème car la fonction  $\varepsilon(t)$  est

discontinue, et ne possède donc pas de dérivée au sens classique. Une théorie a donc dû être établie pour formaliser cette impulsion, qui est formellement définie comme une *distribution*, c'est-à-dire comme un objet mathématique dont on ne peut en principe pas donner la valeur (la dérivée de  $\varepsilon(t)$  est infinie en 0) mais dont on peut connaître l'effet par intégration grâce à (2.15).

En pratique, cependant, on peut utiliser  $\delta(t)$  comme une fonction classique, qu'on assimile, pour respecter (2.16), à la limite de  $rect(t/a)/a$  pour  $a$  tendant vers 0. On ne lui donne pas de valeur, mais plutôt un *poinds* égal à la surface du rectangle, c'est-à-dire 1 (Fig. 2.11)

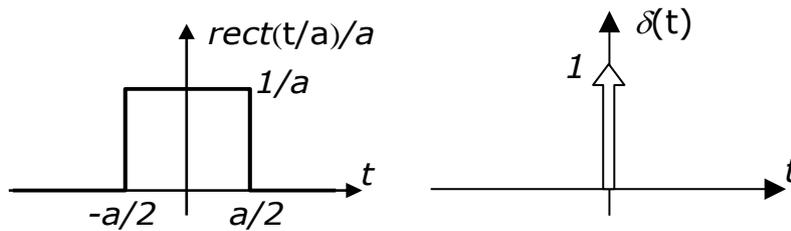


Fig. 2.11 L'impulsion de Dirac à temps continu, vue comme limite de  $rect(t/a)/a$  quand  $a$  tend vers 0.

**Train d'impulsions de Dirac (ou peigne de Dirac)**

Le train d'impulsion de Dirac  $\delta_{T_0}(t)$  est un signal périodique obtenu à partir d'une impulsion de Dirac (Fig. 2.12).

$$\delta_{T_0}(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_0) \tag{2.18}$$

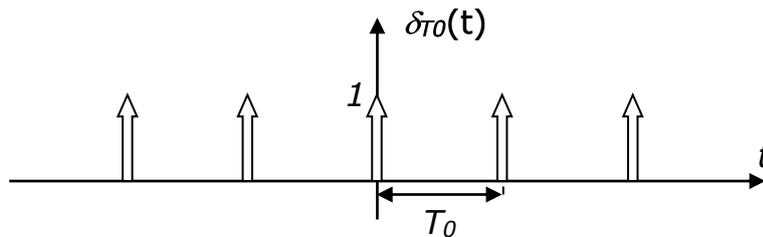


Fig. 2.12 Train d'impulsions de Dirac

**2.2 Opérations sur les signaux**

**2.2.1 Opérations élémentaires sur les signaux**

Les signaux étant assimilés à des fonctions, on peut leur faire subir des opérations simples.

**Décalage temporel**

Le décalage temporel (ou "time-shift") est défini par :

$$f(t) \longrightarrow f(t-t_0) \quad f(n) \longrightarrow f(n-n_0) \quad (2.19)$$

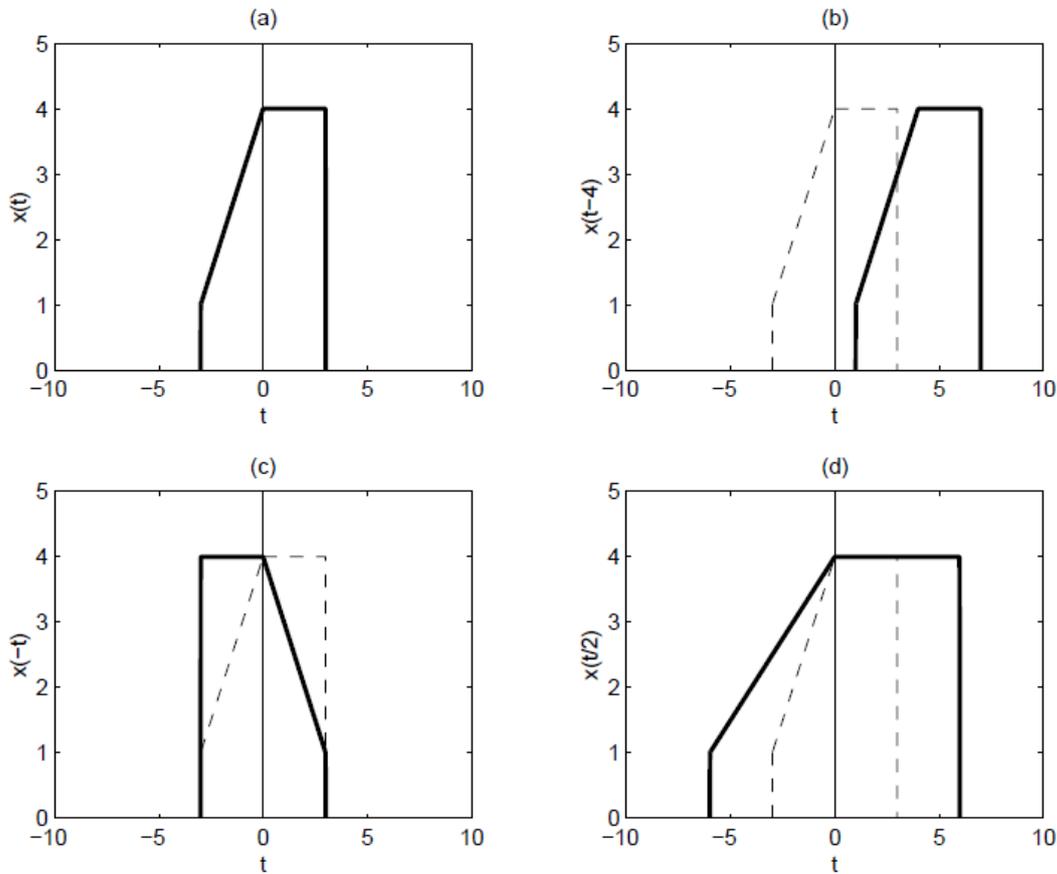


Fig. 2.13 Opérations de base sur les signaux : (a) signal original (b) signal retardé de 4s (c) signal réfléchi (d) signal dilaté d'un facteur 2. [www.montefiore.ulg.ac.be/systems/SYST002/syst002-syllabus.pdf]

### Réflexion

La réflexion (ou "time-reversal") est définie par :

$$f(t) \longrightarrow f(-t) \quad f(n) \longrightarrow f(-n) \quad (2.20)$$

### Changement d'échelle

Le changement d'échelle est défini par :

$$f(t) \longrightarrow f(\alpha t) \quad f(n) \longrightarrow f(\alpha n) \quad (2.21)$$

et contracte ( $\alpha > 1$ ) ou dilate ( $\alpha < 1$ ) le signal selon l'axe temporel (Fig. 2.13).

### Somme

La somme de signaux correspond évidemment à la somme des fonctions sous-jacentes, c'est-à-dire à la somme, pour tout instant  $t$ , de valeurs prises par ces deux fonctions pour ce même instant  $t$  (Fig. 2.14).

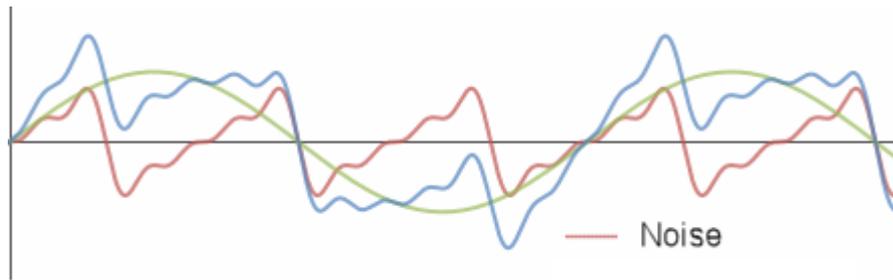


Fig. 2.14 Somme de deux signaux (rouge et bleu) et d'un signal plus complexe (bleu) [http://www.intmath.com/blog/mathematics/how-do-noise-canceling-headphones-work-7619]

**Exemple 2.4**

On peut créer un signal périodique  $f_{T_0}(t)$  de période fondamentale  $T_0$  par la somme infinie de signaux obtenus par décalage, à intervalles réguliers  $kT_0$ , d'un signal de base non périodique  $f(t)$  (Fig. 2.15):

$$f_{T_0}(t) = \sum_{k=-\infty}^{\infty} f(t - kT_0) \tag{2.22}$$

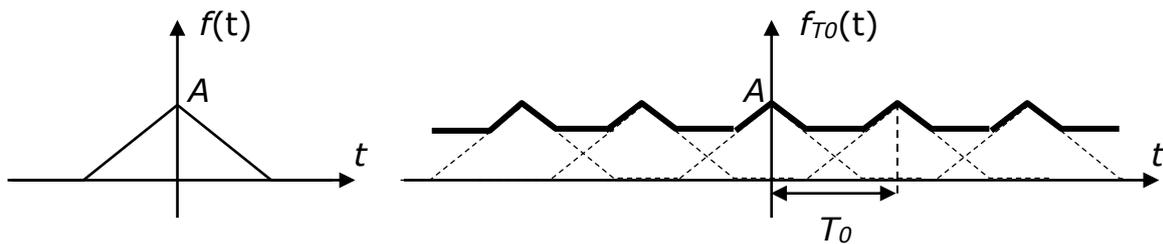


Fig. 2.15 Périodification (à droite) d'un signal de base non périodique (à gauche)

**Produit**

De même, le produit de signaux correspond au produit des fonctions sous-jacentes, c'est-à-dire au produit, pour tout instant  $t$ , de valeurs prises par ces deux fonctions pour ce même instant  $t$  (Fig. 2.14).

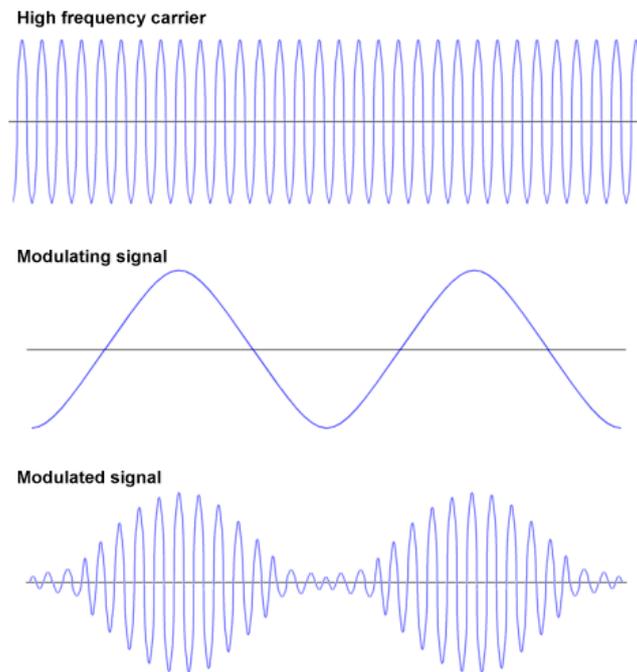


Fig. 2.16 Produit de deux signaux. On parle, en télécommunications, de *modulation d'amplitude*. En musique, on parlera de *trémolo*.  
 [http://www.technologyuk.net/telecommunications/telecom\_principles/amplitude\_modulation.shtml]

### 2.2.2 Produit scalaire de signaux analogiques

Le produit scalaire de deux signaux complexes  $f(t)$  et  $g(t)$  est défini par (Fig. 2.17, gauche):

$$\langle f(t), g(t) \rangle = \int_{-\infty}^{\infty} f(t)g^*(t)dt \quad (2.23)$$

Il peut être vu comme une extension, à l'espace des fonctions, du produit scalaire défini dans un espace vectoriel (complexe, en général).

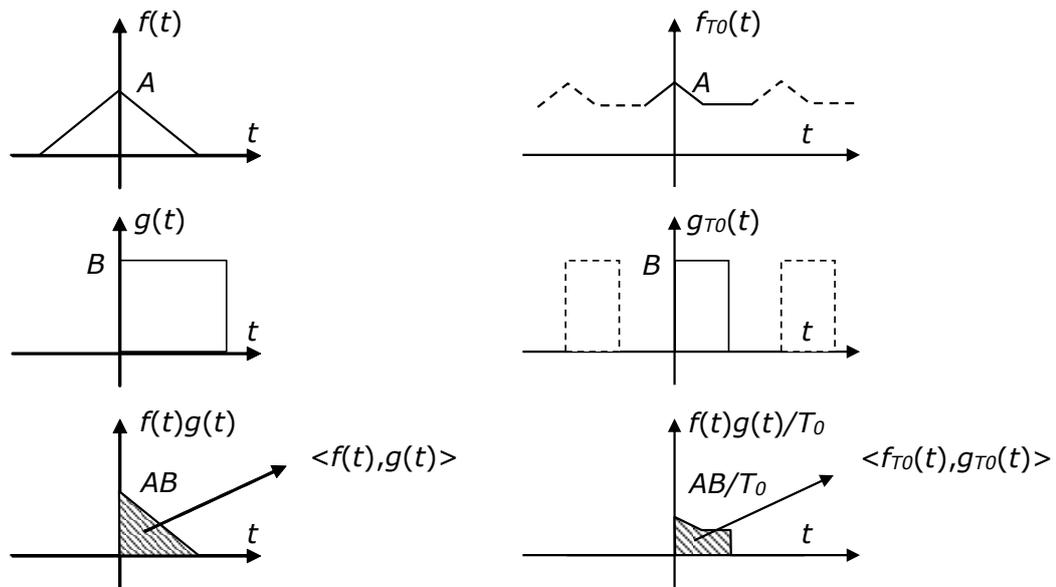


Fig. 2.17 Produit scalaire de deux fonctions (réelles) aperiodiques (gauche) et periodiques de même periode  $T_0$  (droite)

En particulier le produit scalaire d'une fonction par elle-même est égal à son énergie  $E$ , homologue du carré de la norme d'un vecteur :

$$\langle f(t), f(t) \rangle = \int_{-\infty}^{\infty} f(t) f^*(t) dt = \int_{-\infty}^{\infty} |f(t)|^2 dt = E \quad (2.24)$$

L'expression précédente n'est définie que pour les signaux d'énergie finie.

Dans le cas des signaux periodiques, qui sont d'énergie infinie, mais en général de puissance finie, on redéfinit le produit scalaire (voir par exemple Fig. 2.17, droite) comme :

$$\langle f_{T_0}(t), g_T(t) \rangle = \begin{cases} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} f_{T_0}(t) g_T^*(t) dt & (\text{si } T = T_0/k; \quad k=0,1,2,\dots) \\ 0 & (\text{si } T \neq T_0/k; \quad k=0,1,2,\dots) \end{cases} \quad (2.25)$$

Autrement dit : deux fonctions periodiques dont les periodes ne sont pas multiples l'une de l'autre sont de facto *orthogonales*. Dans le cas contraire, leur produit scalaire peut être calculé comme le produit scalaire classique, estimé sur la periode la plus longue et rapporté à celle-ci.

En particulier, le produit scalaire d'un signal periodique par lui-même donne sa puissance  $P$ :

$$\langle f_{T_0}(t), f_{T_0}(t) \rangle = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} |f_{T_0}(t)|^2 dt = P \quad (2.26)$$

### Exemples 2.5

$$\langle \text{rect}(t), \varepsilon(t) \rangle = \int_{-\infty}^{\infty} \text{rect}(t)\varepsilon(t)dt = 1/2 \tag{2.27}$$

(Fig. 2.18)

$$\langle f(t), \delta(t-\tau) \rangle = \int_{-\infty}^{\infty} f(t)\delta(t-\tau)dt = f(\tau) \tag{2.28}$$

(Fig. 2.18)

$$\langle \sin(\omega_0 t), \cos(\omega_0 t) \rangle = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} \frac{\sin(2\omega_0 t)}{2} dt = \frac{1}{T_0} \left[ -\frac{\cos(2\omega_0 t)}{4\omega_0} \right]_{-T_0/2}^{T_0/2} = 0 \tag{2.29}$$

(Fig. 2.19)

$$\langle \cos(\omega_0 t), \cos(\omega_0 t) \rangle = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} \frac{\cos(2\omega_0 t) + 1}{2} dt = \frac{1}{2} \tag{2.30}$$

(Fig. 2.19)

$$\langle e^{j\omega_0 t}, e^{j\omega t} \rangle = \begin{cases} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} e^{j\omega_0 t} e^{-j\omega t} dt & (\text{si } \omega = k\omega_0) \\ 0 & (\text{si } \omega \neq k\omega_0) \end{cases}$$

On démontrera à titre d'exercice que le produit scalaire ci-dessus s'annule même quand  $\omega = k\omega_0$ , sauf pour  $k=1$ . Il reste donc : (2.31)

$$\langle e^{j\omega_0 t}, e^{j\omega t} \rangle = \begin{cases} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} e^{j\omega_0 t} e^{-j\omega_0 t} dt = 1 & (\text{si } \omega = \omega_0) \\ 0 & (\text{si } \omega \neq \omega_0) \end{cases}$$

Nous constatons ici que les exponentielles imaginaires sont des fonctions périodiques très particulières<sup>1</sup> : elle sont toutes orthogonales et de norme unitaire.

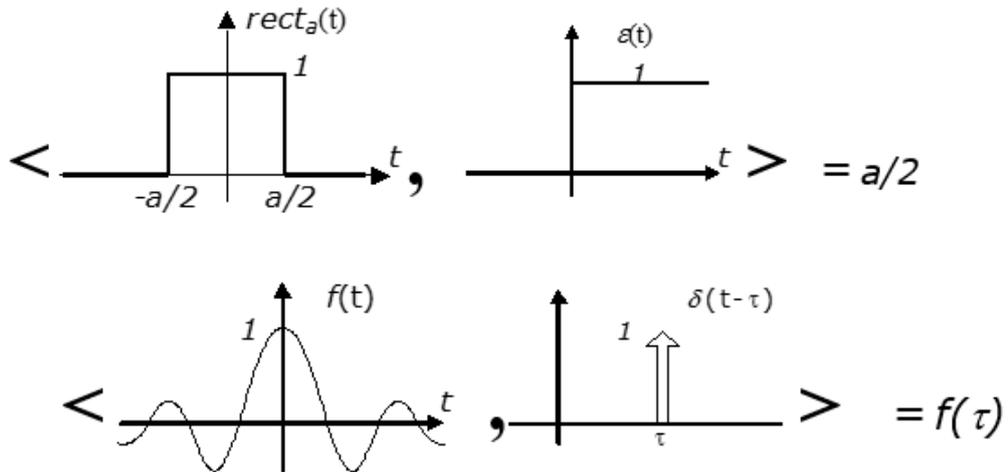


Fig. 2.18 Produits scalaires évidents

---

<sup>1</sup> Ce ne sont pas les seules fonctions orthogonales. Le traitement du signal foisonne d'ensembles de signaux orthogonaux. Les *ondelettes* (dont la théorie a été développée à la fin des années 80 par une chercheuse belge, Ingrid Daubechies) en sont un bel exemple.

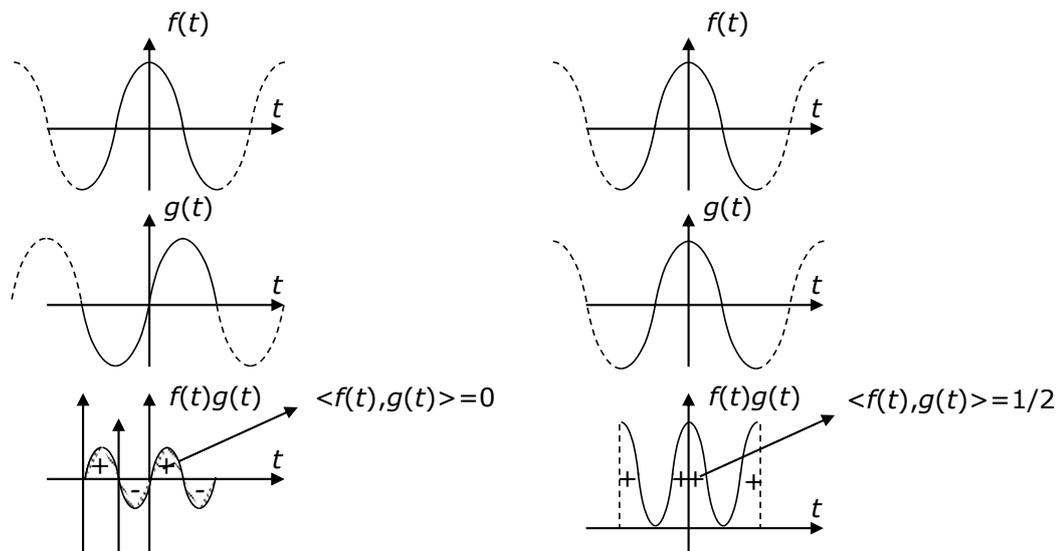


Fig. 2.19  $\langle \sin(\omega_0 t), \cos(\omega_0 t) \rangle$  et  $\langle \cos(\omega_0 t), \cos(\omega_0 t) \rangle$

### 2.2.3 Convolution

La convolution est une opération fondamentale. Elle consiste, pour deux signaux  $x(t)$  et  $y(t)$  à temps continu et à énergie finie, à créer un signal  $z(t)$  donné par :

$$z(t) = x(t) * y(t) = \int_{-\infty}^{\infty} x(\tau) y(t - \tau) d\tau \tag{2.32}$$

Il faut comprendre cette équation de la manière suivante : pour toute valeur de  $t$ ,  $z(t)$  est donné par le produit scalaire entre  $x(\tau)$  et  $x(t - \tau)$ .

Il est donc possible de lui donner une interprétation graphique simple :  $y(t - \tau)$  est en effet simplement le résultat d'une réflexion de la fonction  $y(\tau)$  en  $y(-\tau)$ , suivie d'un décalage de  $t$  vers la droite pour donner  $y(t - \tau)$ . L'intégrale de convolution est alors l'aire sous-tendue par le produit de  $x(t)$  et cette fonction réfléchi-décalée. L'intégrale est à calculer pour toutes les valeurs possibles du décalage  $t$ , ce qui permet de tracer  $y(t)$  point par point (Fig. 2.20).

On montre facilement que le produit de convolution est commutatif.

$$x(t) * y(t) = y(t) * x(t) \tag{2.33}$$

Dans le cas de deux signaux périodiques de fréquences multiples l'une de l'autre, l'intégrale (2.32) est infinie, et le produit scalaire entre  $x(\tau)$  et  $y(t - \tau)$  s'écrit plutôt :

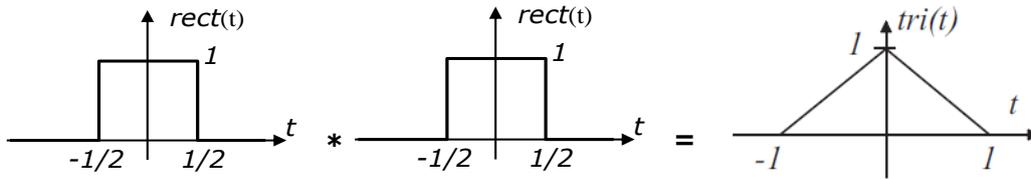
$$z_{T_0}(t) = x_{T_0}(t) * y_T(t) = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(\tau) y(t - \tau) d\tau \quad \text{avec } T_0 = kT \quad (k = 1, 2, 3, \dots) \tag{2.34}$$

Il est nul si les fréquences ne sont pas multiples (selon (2.25)).

#### Exemple 2.6 - Convolution entre deux rectangles

On montre très facilement (graphiquement ou analytiquement) que :

$$\text{rect}(t) * \text{rect}(t) = \text{tri}(t) \quad (2.35)$$



### Exemple 2.7 - Convolution avec une impulsion de Dirac

En utilisant l'interprétation graphique donnée ci-dessus (ou suivant (2.13)), il est facile de comprendre que :

$$f(t) * \delta(t) = \int_{-\infty}^{\infty} f(\tau) \delta(t - \tau) d\tau = f(t) \quad (2.36)$$

et plus généralement encore :

$$f(t) * \delta(t - t_0) = f(t - t_0) \quad (2.37)$$

On en déduit que convoluer un signal avec une impulsion de Dirac revient à déplacer l'origine de ce signal au droit de l'impulsion (Fig. 2.21).

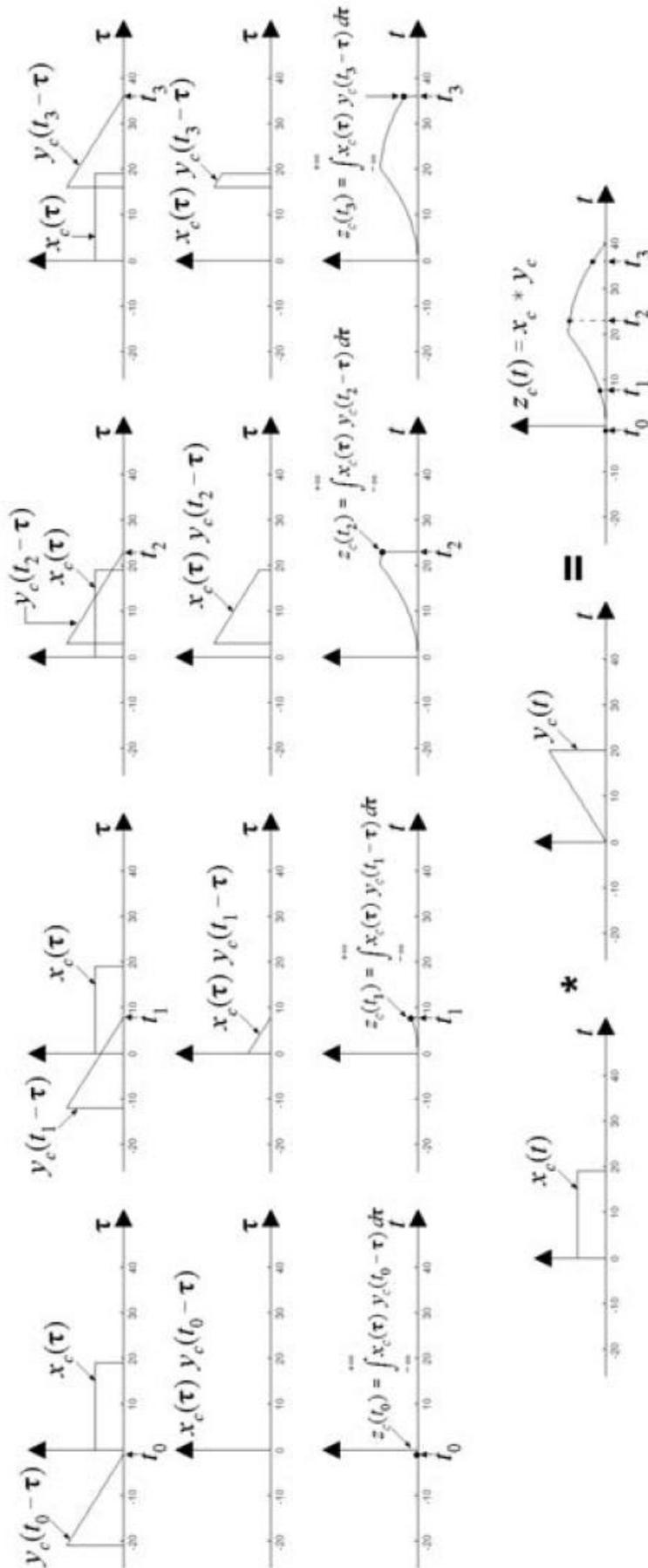


Fig. 2.20 Construction graphique de  $z(t)=x(t)*y(t)$  pour différentes valeurs de  $t$ .

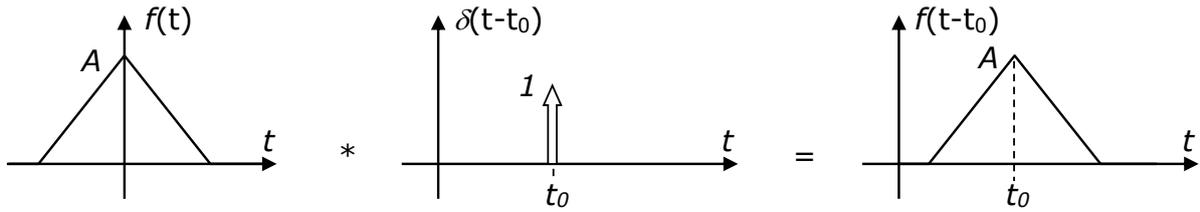


Fig. 2.21 Convolution d'un signal de base non périodique avec une impulsion de Dirac.

**Exemple 2.8 - Convolution avec un peigne de Dirac**

On a vu plus haut (2.22) comment on pouvait créer un signal périodique.

En utilisant l'exemple ci-dessus, on voit qu'on obtient le même résultat par convolution entre  $\delta_{T_0}(t)$  et une fonction  $f(t)$  non périodique de départ (Fig. 2.22) :

$$f(t) * \delta_{T_0}(t) = \int_{-\infty}^{\infty} f(\tau) \delta_{T_0}(t - \tau) d\tau = f_{T_0}(t) \tag{2.38}$$

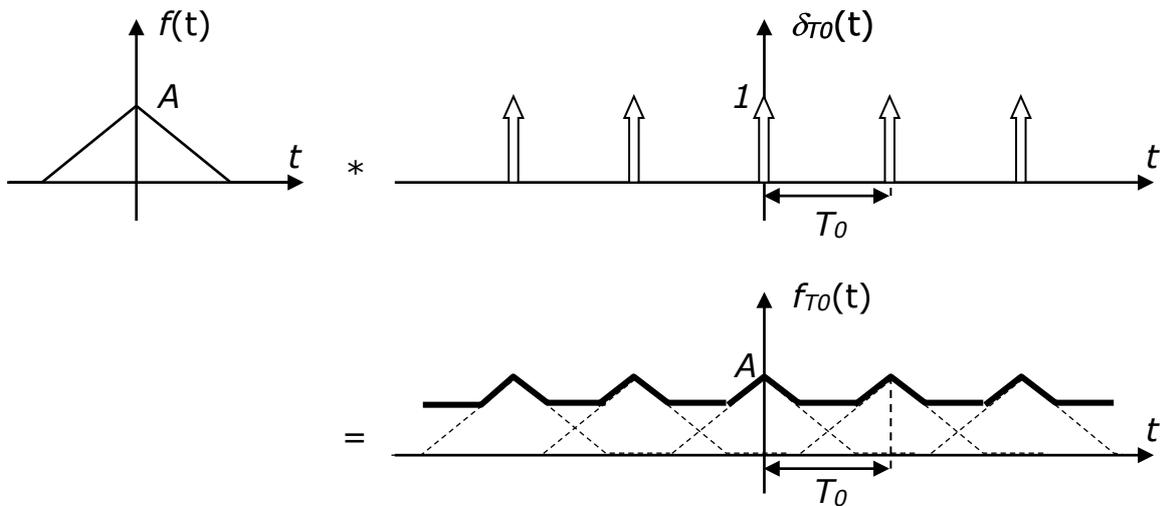


Fig. 2.22 Convolution d'un signal de base non périodique avec un peigne de Dirac

**2.3 Analyse fréquentielle des signaux analogiques**

Le but de l'analyse fréquentielle est de décomposer un signal quelconque en une somme de signaux élémentaires bien connus (Fig. 2.23).

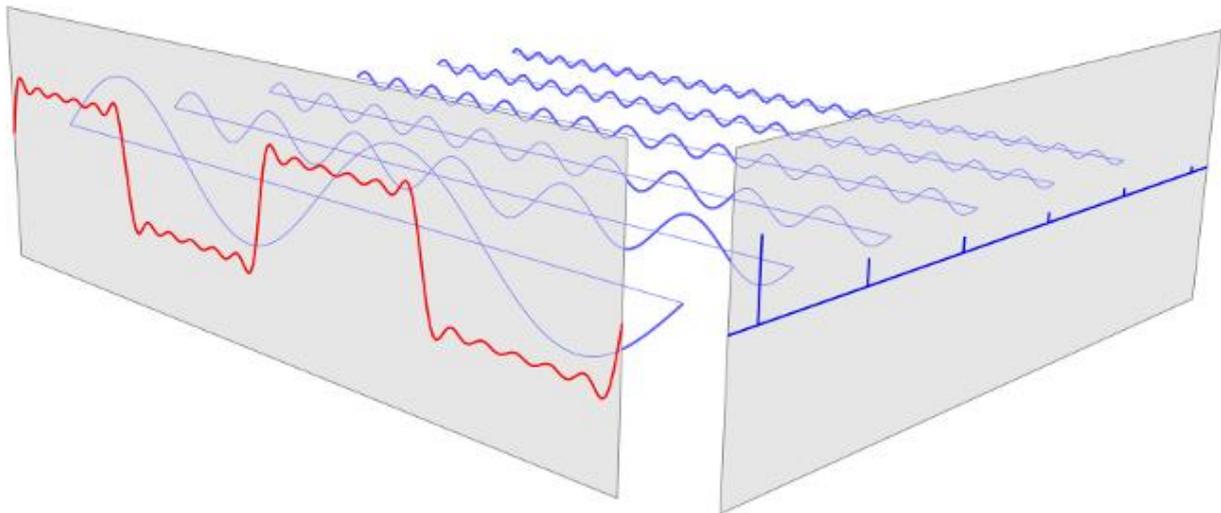


Fig. 2.23 Décomposition d'un signal (à gauche ; ici, un signal périodique) en une somme de sinusoides (au centre) par calcul de leurs amplitudes (à droite) [https://www.quora.com/Laplace-Transforms/What-are-the-best-possible-pedagogical-functions-examples-to-use-when-intuitively-describing-Laplace-Transforms-to-someone]

Rappelons tout d'abord (Fig. 2.24) que dans un espace vectoriel à deux dimensions, on peut toujours exprimer un vecteur quelconque  $\vec{x}$  comme combinaison linéaire des vecteurs d'une base orthonormée  $\{\vec{u}_1, \vec{u}_2\}$  :

$$\vec{x} = x_1\vec{u}_1 + x_2\vec{u}_2 \tag{2.39}$$

où les coefficients de cette combinaison linéaire sont donnés par :

$$x_1 = \langle \vec{x}, \vec{u}_1 \rangle \text{ et } x_2 = \langle \vec{x}, \vec{u}_2 \rangle \tag{2.40}$$

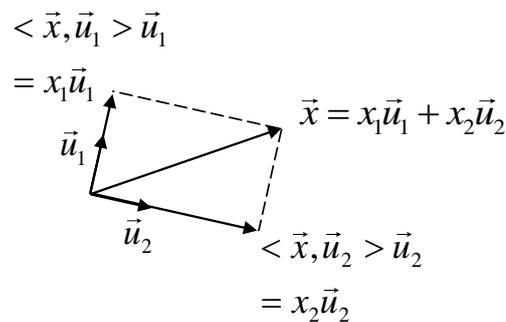


Fig. 2.24 Décomposition d'un vecteur sur une base orthonormée.

Nous verrons ci-dessous que l'on peut procéder de la même façon pour les signaux, à condition de définir une base de l'espace des signaux.

### 2.3.1 Transformée de Fourier d'un signal d'énergie finie

#### Impulsions de Dirac et base orthonormée

Rappelons l'équation (1.50) :

$$\int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau = f(t) \quad (2.41)$$

Cette équation peut être interprétée de la façon suivante : un signal analogique  $f(t)$  peut être décomposé comme une somme pondérée (et continue, d'où l'intégrale qui remplace la somme) de fonctions de base orthonormées  $\{\delta(t-\tau) (\tau \in [-\infty, +\infty])\}$ . Les coefficients de cette décomposition sont fonction de la variable  $\tau$  et valent tout simplement  $f(\tau)$ .

$$\begin{aligned} & \langle f(t), \delta(t-t_1) \rangle \delta(t-t_1) \\ &= f(t_1)\delta(t-t_1) \end{aligned}$$

$$f(t) = f(t_1)\delta(t-t_1) + f(t_2)\delta(t-t_2)$$

$$\begin{aligned} & \langle f(t), \delta(t-t_2) \rangle \delta(t-t_2) \\ &= f(t_2)\delta(t-t_2) \end{aligned}$$

Fig. 2.25 Décomposition d'une fonction sur des impulsions de Dirac.<sup>2</sup>

L'équation (1.42) exprime alors simplement le fait que les coefficients de cette décomposition sont obtenus par produit scalaire de  $f(t)$  avec les fonctions de base, comme on le ferait dans un espace vectoriel classique.

### Exponentielles imaginaires et base orthonormée

Un des résultats importants de l'analyse mathématique (que l'on admettra ici) est de montrer que l'on peut également décomposer un signal d'énergie finie comme une somme pondérée (et continue) de fonctions de base  $\{e^{j\omega t} (\omega \in [-\infty, +\infty])\}$ : les exponentielles imaginaires (Fig. 2.26). Ceci s'exprime comme suit :

$$f(t) = \int_{-\infty}^{\infty} F(f)e^{j\omega t} df \quad (2.42)$$

Les coefficients (complexes)  $F(f)$  de cette somme pondérée continue sont fonction de la variable  $f$  (fréquence:  $f = \omega/2\pi$ ) et constituent la *transformée de Fourier* de  $f(t)$ , notée  $F(f)$ .

<sup>2</sup> Ce graphique est à prendre symboliquement : les deux impulsions de Dirac ne constituent pas une base ; il faut en réalité décomposer sur l'ensemble infini des impulsions.

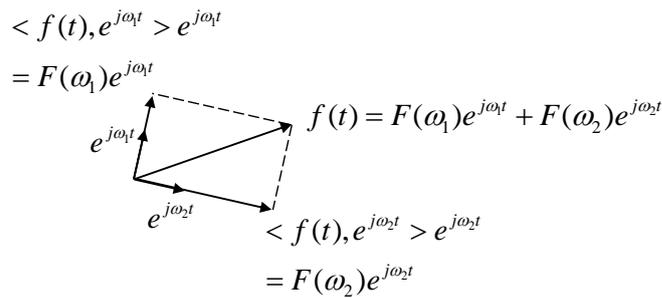


Fig. 2.26 Décomposition d'une fonction sur des exponentielles imaginaires.<sup>3</sup>

Le calcul de  $F(f)$  est simple : il est basé sur la projection de  $f(t)$  sur les fonctions de base, à l'aide du produit scalaire. Si l'on admet en effet que les exponentielles imaginaires constituent une base de l'ensemble des fonctions d'énergie finie, cette base est orthonormée, vu (1.45). Il s'ensuit que les coefficients intervenant dans la décomposition de  $f(t)$  sur ces fonctions de base peuvent être obtenus par simple projection de  $f(t)$  sur les fonctions de base :

$$F(f) = \langle f(t), e^{j\omega t} \rangle \tag{2.43}$$

ce qui conduit à l'expression bien connue de  $F(f)$  :

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \tag{2.44}$$

Le couple  $f(t)$  et  $F(f)$  constitue une *paire de transformées de Fourier*. On note souvent cette relation sous la forme :

$$f(t) \overset{\mathbb{F}}{\Leftrightarrow} F(f) \tag{2.45}$$

### Amplitude et phase

Si on définit l'*amplitude*  $A(f)$  et la *phase*  $\varphi(f)$  de la transformée de Fourier par :

$$F(f) = A(f) e^{j\varphi(f)} \tag{2.46}$$

Il est facile de comprendre que, pour que  $f(t)$  soit une fonction réelle (ce qui sera toujours le cas ici, à l'exception de l'exponentielle imaginaire),  $A(f)$  doit être une fonction paire en  $f$  en tandis que  $\varphi(f)$  est une fonction impaire en  $f$  (Fig. 2.27). Les exponentielles complexes dont est constitué un signal réel vont en effet par paires complexes conjuguées : de même amplitude, de fréquences égales et opposées, et avec des phases initiales opposées (Fig. 2.28).

<sup>3</sup> Ce graphique est à prendre symboliquement : les deux fonctions ne constituent pas une base ; il faut en réalité décomposer sur l'ensemble infini des exponentielles imaginaires.

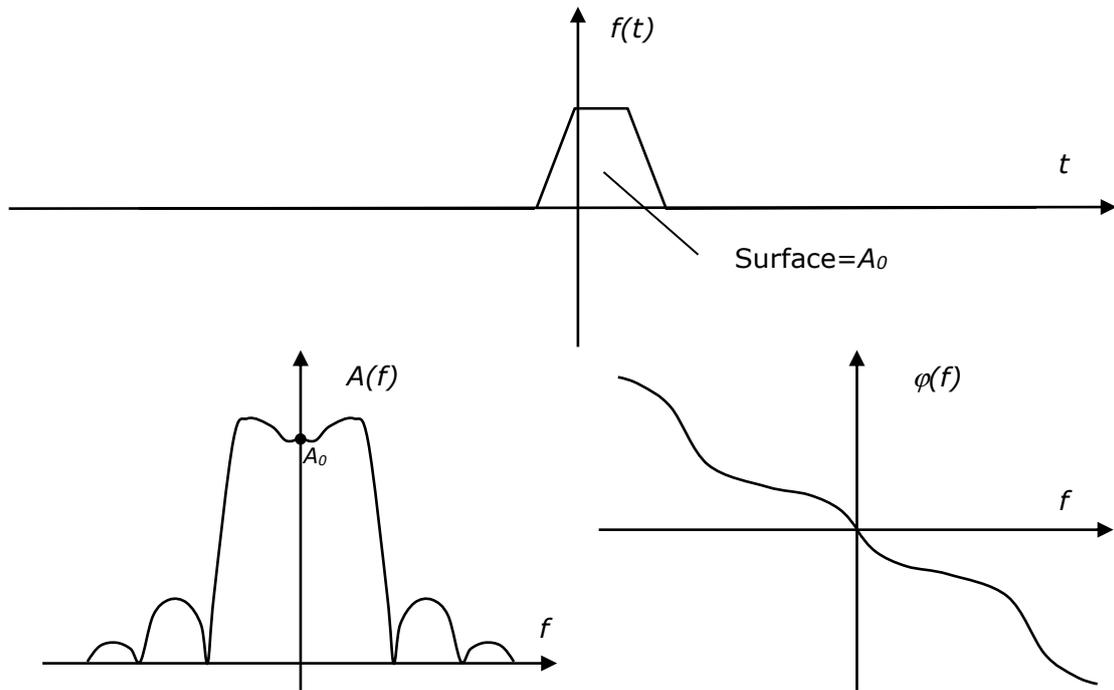


Fig. 2.27 Amplitude (paire) et phase (impaire) de la transformée de Fourier d'un signal réel non périodique

En particulier, on a pour la valeur en  $f=0$  :

$$F(0) = \langle f(t), e^{j0t} \rangle = \langle f(t), 1 \rangle = \int_{-\infty}^{\infty} f(t) dt = A(0) \tag{2.47}$$

ce qui correspond à la somme des surfaces sous-tendues par  $f(t)$ , comptées avec leur signe.

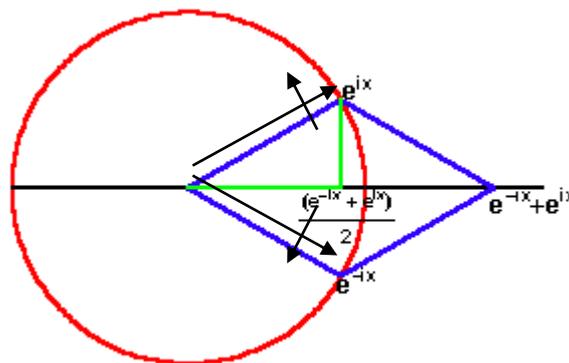


Fig. 2.28 Somme de phaseurs d'amplitudes identiques et de phases opposées.

### 2.3.2 Propriétés de la Transformée de Fourier

Lorsqu'on applique sur les signaux les opérations mentionnées au chapitre précédent, leurs transformées de Fourier se modifient. Ces propriétés ont été étudiées dans un cours de Mathématiques Appliquées, et sont rappelées au Tableau 2.1.

(dualité)  $F(t) \Leftrightarrow f(-f)$

(décalage temporel)	$f(t-t_0)$	$\Leftrightarrow$	$F(f)\exp(-jt_0\omega)$
(réflexion)	$f(-t)$	$\Leftrightarrow$	$F^*(f)$
(changement d'échelle)	$f(at)$	$\Leftrightarrow$	$\frac{1}{ a }F\left(\frac{f}{a}\right)$
(dérivée)	$\frac{df(t)}{dt}$	$\Leftrightarrow$	$j\omega F(f)$ avec $\omega = 2\pi f$
(somme)	$\sum_i a_i \cdot f_i(t)$	$\Leftrightarrow$	$\sum_i a_i \cdot F_i(f)$
(produit)	$f(t)g(t)$	$\Leftrightarrow$	$F(f)*G(f)$
(convolution)	$f(t)*g(t)$	$\Leftrightarrow$	$F(f)G(f)$
(produit scalaire)	$\langle f(t), g(t) \rangle$	$\Leftrightarrow$	$\langle F(f), G(f) \rangle$
(Parseval)	$\langle f(t), f(t) \rangle$	$\Leftrightarrow$	$\langle F(f), F(f) \rangle$

Tableau 2.1 Propriétés principales de la transformée de Fourier

La propriété de dualité vient du fait que l'expression de la transformée de Fourier inverse est identique, au signe près, à celle de la transformée de Fourier).

Retarder un signal ne change rien à l'amplitude de sa transformée car les exponentielles imaginaires qui composent le signal sont simplement retardées. Ce retard correspond à une phase linéaire qui vient s'ajouter à la phase de la transformée. On le comprend facilement sur la Fig. 2.29 : retarder un signal composé de deux sinusoïdes à  $f_0$  et  $2f_0$  implique de déphaser la seconde deux fois plus que la première (sans quoi le signal reconstruit n'aurait pas la même forme).

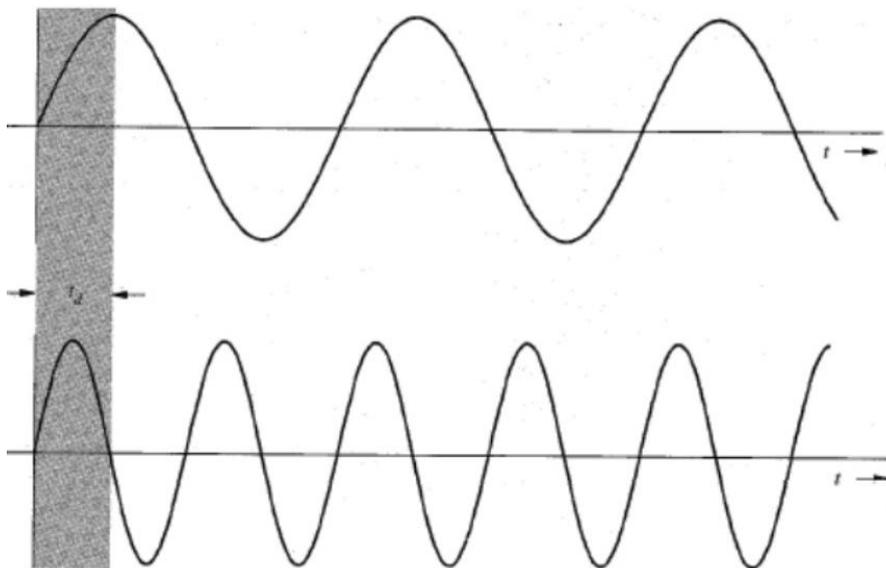


Fig. 2.29 Retard d'un signal composé de deux sinusoïdes..

La réflexion d'un signal ne modifie très logiquement que la phase de la transformée : toutes les exponentielles imaginaires qui le composent doivent être elles-mêmes réfléchies, et voient donc leur phase changer de signe.

Dilater un signal d'un facteur  $a$  revient donc à compresser sa transformée du même facteur. En effet, le produit scalaire du signal original avec une exponentielle imaginaire de période  $T$  (et donc de fréquence  $1/T$ ) est identique (à un facteur près) à celui du signal dilaté avec une exponentielle imaginaire elle-même dilatée, de période  $aT$  (et donc de fréquence  $1/aT$ ). Le facteur de proportionnalité est facile à comprendre : la surface du signal dilaté étant multipliée par  $a$ , l'amplitude de la transformée est elle aussi multipliée par  $a$  (Fig. 2.30).

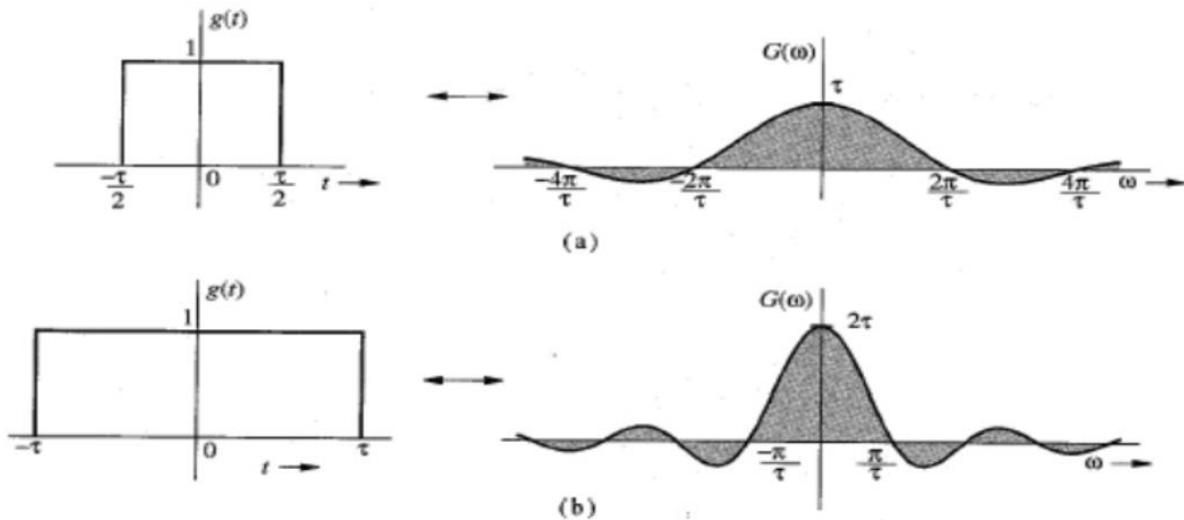


Fig. 2.30 Transformée d'un signal dilaté (ici d'un facteur 2).

La dérivée temporelle d'un signal correspond à la dérivée temporelle des exponentielles imaginaires qui le composent, et fait donc apparaître dans la transformée un facteur  $j\omega$  d'amplitude linéaire en fréquence et de phase constante de  $+90^\circ$  et  $-90^\circ$  (Fig. 2.31).

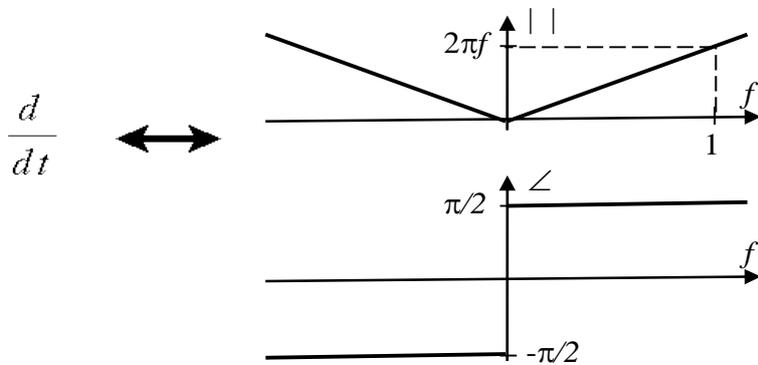


Fig. 2.31 Transformée de Fourier de la dérivée : facteur multiplicatif d'amplitude linéaire en fréquence et de phase égale à  $\pm 90^\circ$ .

Les théorèmes du produit scalaire et de Parseval (qui n'est qu'une forme particulière du précédent) prennent un sens évident si on comprend que  $f(\tau)$  et  $F(f)$  sont les composantes de  $f(t)$  sur deux ensembles de fonctions de base orthonormées :  $\{ \delta(t-\tau) (\tau \in [-\infty, +\infty]) \}$  et  $\{ e^{j\omega t} (\omega \in [-\infty, +\infty]) \}$ . Ce théorème n'exprime en effet rien d'autre que le fait que le produit scalaire de deux fonctions (et donc en particulier pour Parseval le carré de la norme d'une fonction) est indépendant du repère sur lequel on les projette. Rappelons en effet que pour des vecteurs dans un espace vectoriel, le produit scalaire de deux vecteurs est égal au produit de leurs normes par le cosinus de l'angle qu'ils font entre eux, et est donc indépendant de la base sur laquelle on projette les vecteurs (Fig. 2.32).

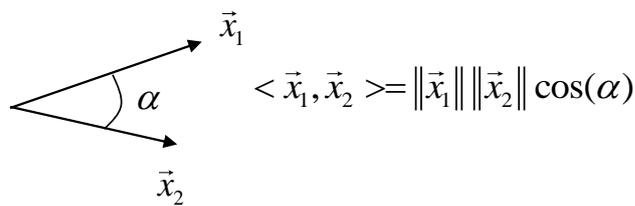
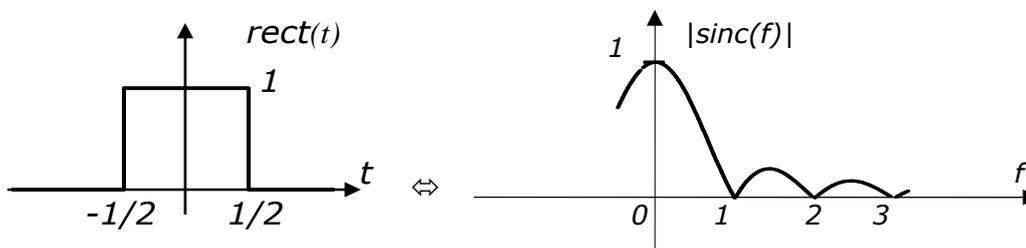


Fig. 2.32 Produit scalaire dans un espace vectoriel.

**Exemple 2.9**

$$\begin{aligned}
 \text{rect}(t) &\stackrel{\mathbb{F}}{\Leftrightarrow} \int_{-1/2}^{1/2} e^{-j\omega t} dt = \frac{1}{-j\omega} [e^{-j\omega/2} - e^{j\omega/2}] \\
 &= 2 \frac{\sin \omega/2}{\omega} = \frac{\sin \omega/2}{\omega/2} \\
 &= \text{sinc}(f)
 \end{aligned}
 \tag{2.48}$$



On peut retrouver ce résultat intuitivement. Il est clair, si l'on se souvient que la transformée de Fourier est un produit scalaire, que sa valeur en  $f=0$  est l'intégrale de la fonction  $f(t)$  (donc ici 1). Par ailleurs, il est évident que  $F(f)$ , étant un produit scalaire entre  $f(t)$  et les exponentielles imaginaires, s'annule pour toutes les exponentielles imaginaires de périodes égales à  $1/k$  (avec  $k$  entier positif) .

**Exemple 2.10 – Transformée du signal triangulaire**

On a vu précédemment que la convolution de deux signaux rectangulaires donne un signal triangulaire. On en déduit (Fig. 2.33) que :

$$tri(t) \stackrel{\mathbb{F}}{\Leftrightarrow} sinc^2(f)$$

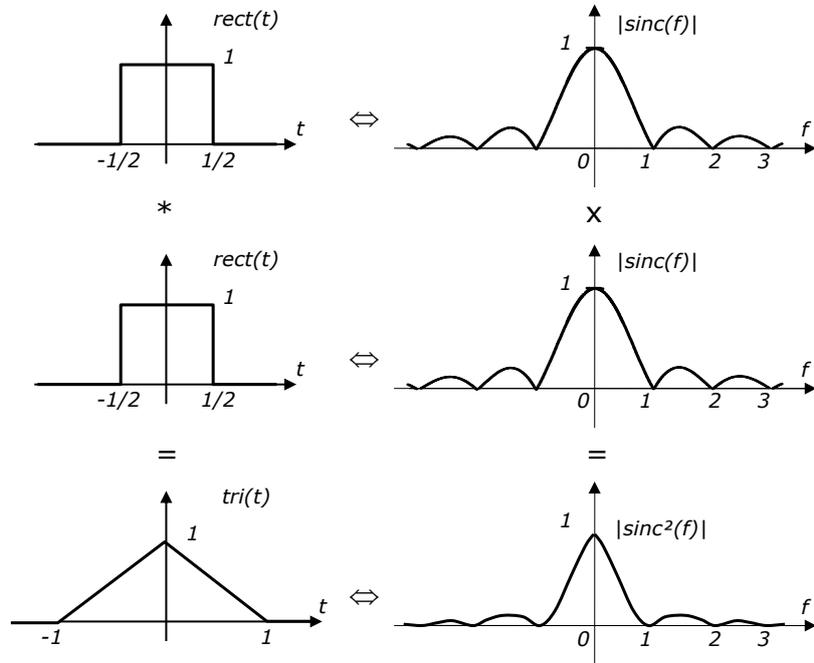
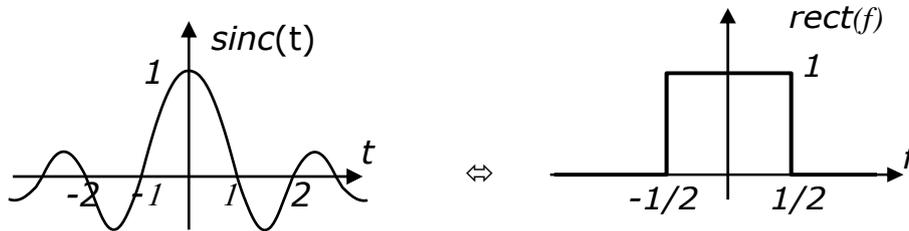


Fig. 2.33 Lien entre transformée de Fourier d'un signal carré et triangulaire

**Exemple 2.11 – Transformée du sinus cardinal**

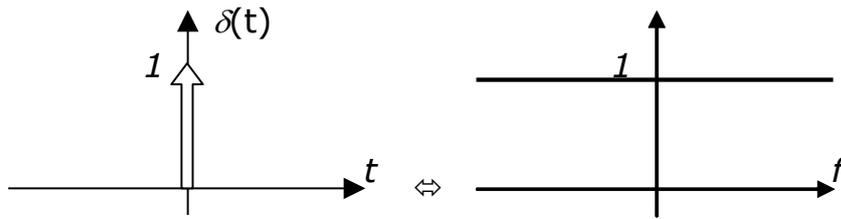
Par dualité de (2.49), il vient :

$$sinc(t) \stackrel{\mathbb{F}}{\Leftrightarrow} rect(-f) = rect(f)$$



**Exemple 2.12**

$\delta(t) \stackrel{\mathbb{F}}{\Leftrightarrow} 1$  (évident vu l'interprétation de la transformée de Fourier comme un calcul de produits scalaires entre la fonction de départ et les exponentielles imaginaires)



### 2.3.3 Transformée de Fourier d'une fonction périodique - Séries de Fourier

La propriété (2.42) reste valable pour un signal périodique (de puissance finie)  $f_{T_0}(t)$  à condition d'utiliser, pour la projection (2.44), le produit scalaire *au sens des distributions*:

$$\langle f_{T_0}(t), e^{j\omega t} \rangle = \begin{cases} 0 & \text{si } \omega \neq k\omega_0 \\ \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} f_{T_0}(t) e^{-j\omega t} dt & \text{si } \omega = k\omega_0 \end{cases} \quad (\omega_0 = \frac{2\pi}{T_0}) \quad (2.49)$$

et de considérer que les valeurs non-nulles de ce produit scalaire sont en réalité les poids d'impulsions de Dirac<sup>4</sup> aux fréquences correspondantes :

$$F_{T_0}(f) = \sum_{k=-\infty}^{\infty} F_k \delta(f - kf_0) \quad \text{avec} \quad F_k = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} f_{T_0}(t) e^{-jk\omega_0 t} dt \quad (2.50)$$

Les signaux périodiques possèdent donc une transformée de Fourier constituée de raies (Fig. 2.34). L'amplitude de ces raies est égale à l'amplitude de la transformée de Fourier d'une période de  $f_{T_0}(t)$ , divisée par la période  $T_0$ .

<sup>4</sup> La justification théorique de l'apparition d'impulsions de Dirac en fréquence est la suivante :

$$\delta(t) \stackrel{\mathbb{F}}{\leftrightarrow} 1 \text{ (Exemple 9)} \rightarrow \delta(t - t_0) \stackrel{\mathbb{F}}{\leftrightarrow} e^{j\omega t_0} \text{ (décalage temporel)} \rightarrow e^{j\omega t} \stackrel{\mathbb{F}}{\leftrightarrow} \delta(f - f_0) \text{ (dualité).}$$

Ainsi, plus généralement, décomposer un signal périodique sur les exponentielles imaginaires conduit à l'apparition d'impulsions de Dirac en fréquence, dont les poids sont les coefficients des composantes.

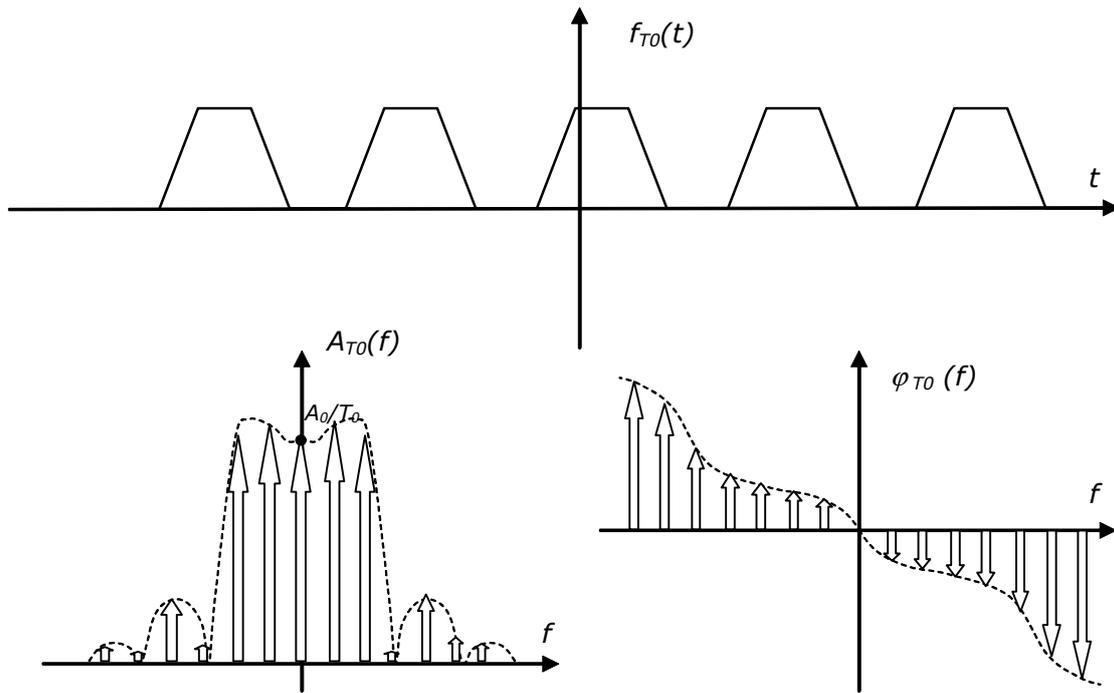


Fig. 2.34 Amplitude et phase de la transformée de Fourier d'un signal périodique

La somme continue (2.42) se réduit finalement une somme discrète, que l'on appelle *série de Fourier* associée à  $f_{T0}(t)$  :

$$f_{T_0}(t) = \sum_{k=-\infty}^{\infty} F_k e^{jk\omega_0 t} \tag{2.51}$$

Notons que, comme les composantes exponentielles imaginaires de la transformée de Fourier d'un signal périodique sont orthogonales et de valeur efficace unitaire (car de puissance unitaire), on a :

$$F_{RMS} = \sqrt{F_{-N}^2 + \dots + F_{-1}^2 + F_0^2 + F_1^2 + \dots + F_N^2} \tag{2.52}$$

où  $F_{-N}, \dots, F_N$  sont les amplitudes des  $2N+1$  exponentielles imaginaires (de fréquence positive, nulle, et négative) qui composent le signal (voir exemple Fig. 2.35).

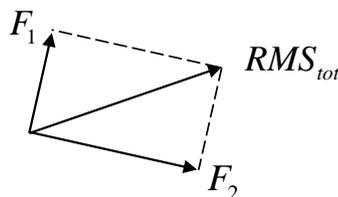


Fig. 2.35 Valeur RMS totale d'un signal composé de 2 exponentielles imaginaires. Un cas particulier est celui d'un sinus, composé de 2 raies d'amplitude  $1/2$ , et de valeur efficace égale à  $\sqrt{2}/2$ .

En particulier pour un signal réel de moyenne nulle ( $F_0=0$ ) :

$$F_{RMS} = \sqrt{2(F_1^2 + \dots + F_N^2)} \tag{2.53}$$

On peut alors définir, pour les signaux périodiques, un *taux de distorsion harmonique total* (THD, Total Harmonic Distortion), qui indique le taux d'harmoniques présent dans le signal et est le rapport (en %) entre la valeur efficace de l'ensemble des harmoniques d'ordre supérieur à 1 et la valeur efficace du signal entier :

$$THD = 100 \frac{\sum_{k=2}^{\infty} 2F_k^2}{\sum_{k=1}^{\infty} 2F_k^2} = 100 \frac{\sum_{k=2}^{\infty} F_k^2}{\sum_{k=1}^{\infty} F_k^2} \tag{2.54}$$

**Exemple 2.13**

Il est évident qu'une exponentielle imaginaire de pulsation  $\omega_0$  peut être décomposée comme une somme continue et pondérée d'exponentielles imaginaires : la seule exponentielle intervenant dans cette décomposition est bien entendu l'exponentielle de départ (Fig. 2.36).

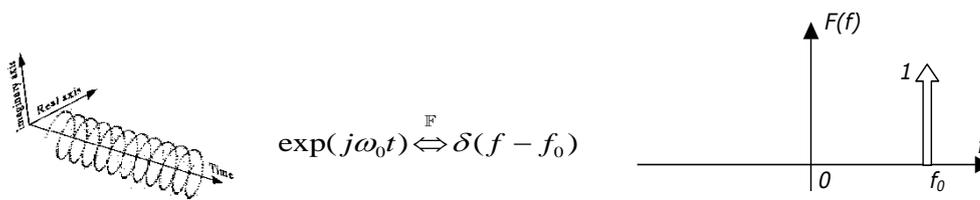


Fig. 2.36 Transformée de Fourier d'une exponentielle imaginaire

**Exemple 2.14**

De même, on sait déjà qu'un cosinus est la somme de deux exponentielles imaginaires conjuguées et d'amplitude 1/2; la décomposition s'exprime donc comme à la Fig. 2.37.

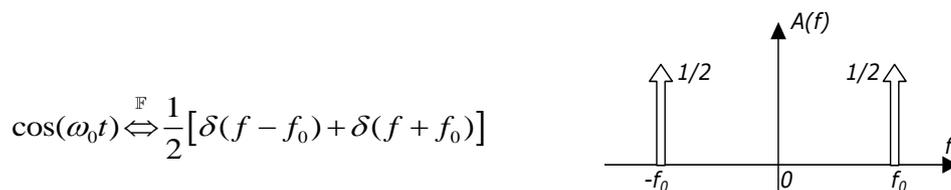


Fig. 2.37 Module de la transformée de Fourier d'un cosinus

En particulier, la fonction  $f(t)=1$  n'est rien d'autre qu'un cosinus de fréquence nulle :



Fig. 2.38 Module de la transformée de Fourier d'une constante

On trouve tout aussi facilement, et sans aucun calcul d'intégrale :

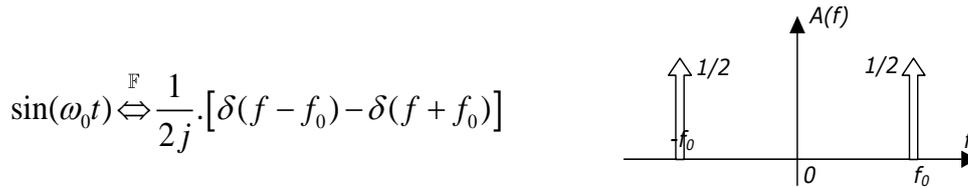


Fig. 2.39 Module de la transformée de Fourier d'un sinus

**Exemple 2.15**

La projection d'un *peigne* de Dirac temporel de période  $T_0$  sur les fonctions de base donne un peigne de Dirac fréquentiel de période  $f_0$  et d'amplitude  $1/T_0$ .

$$\delta_{T_0}(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_0) \overset{\mathbb{F}}{\leftrightarrow} \frac{1}{T_0} \delta_{f_0}(f)$$

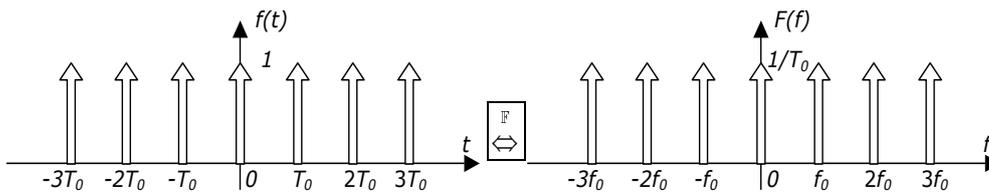


Fig. 2.40 Transformée de Fourier d'un peigne de Dirac

Cette dernière relation permet d'ailleurs de comprendre l'équation (2.50) et la Fig. 2.34 différemment. En effet, puisqu'on a :

$$f_{T_0}(t) = f(t) * \delta_{T_0}(t) \tag{2.55}$$

cette convolution se traduit en transformée de Fourier par un produit (Fig. 2.41):

$$F_{T_0}(f) = \frac{1}{T_0} F(f) \delta_{f_0}(f) \tag{2.56}$$

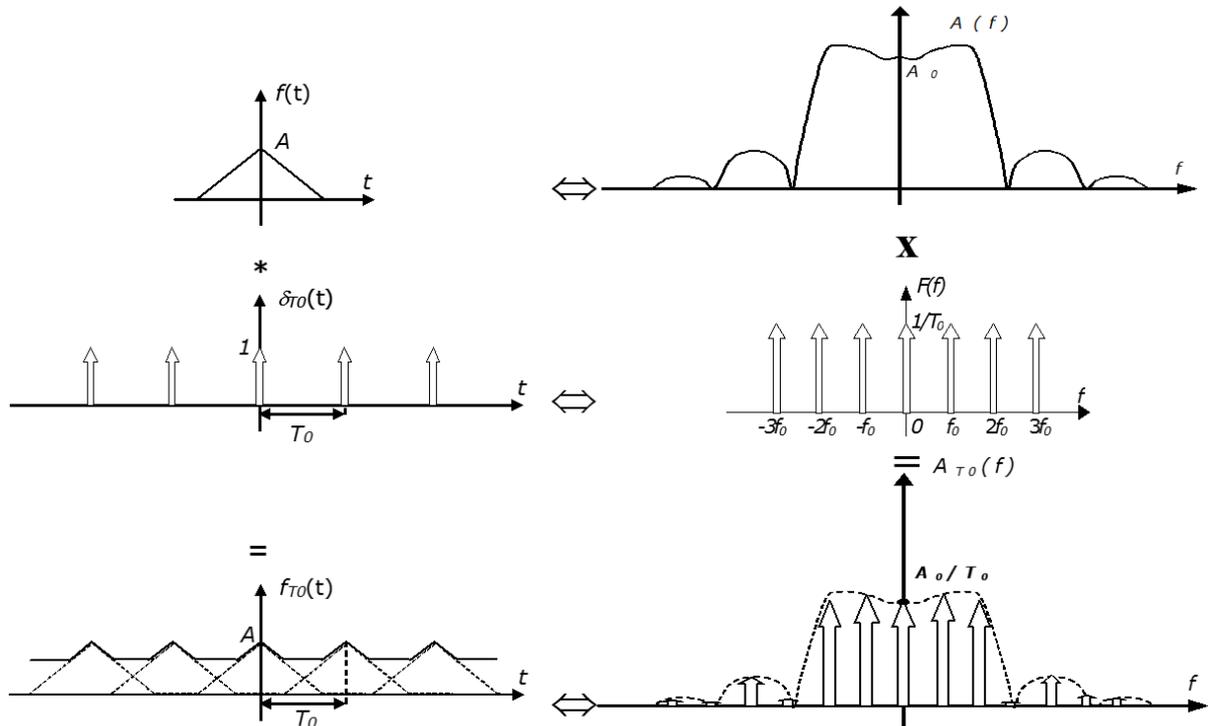
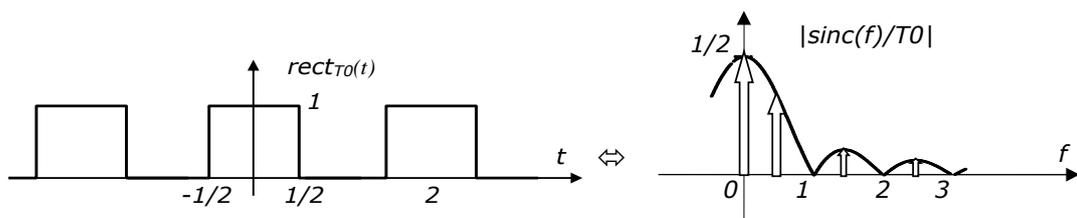


Fig. 2.41 Lien entre transformée de Fourier d'un signal non périodique et série de Fourier du signal périodifié.

**Exemple 2.16 – signal rectangulaire périodique**

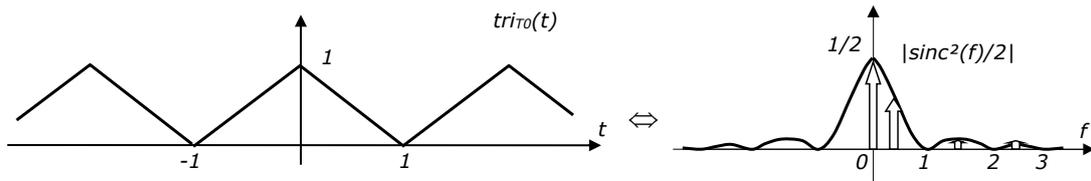
Rendons le signal rectangulaire  $rect(t)$  périodique de période égale à 2s. Sa transformée de Fourier fera apparaître des raies en  $f=0, 1/2, 1, 3/2, etc.$ , de valeur  $sinc(f)/2$ .



On constate que la transformée fait apparaître une raie en  $f=0$  (composante continue= $1/2$ ), puis uniquement des harmoniques  $F^k$  impaires, d'amplitudes décroissant en  $1/k$ .

**Exemple 2.17 – signal triangulaire périodique**

Rendons le signal rectangulaire  $tri(t)$  périodique de période égale à 2s. Sa transformée de Fourier fera apparaître des raies en  $f=0, 1/2, 1, 3/2, etc.$ , de valeur  $sinc^2(f)/2$ .



On constate que la transformée fait apparaître une raie en  $f=0$  (composante continue= $1/2$ ), puis uniquement des harmoniques  $F^k$  impaires, d'amplitudes décroissant en  $1/k$ .

### 2.3.4 Autocorrélation – Densité spectrale d'énergie et de puissance

On a constaté dans la Section précédente l'intérêt du produit scalaire de fonctions pour l'interprétation de la transformée de Fourier comme une technique de décomposition sur une base orthonormée.

L'intérêt du produit scalaire de fonctions va bien au-delà. En effet, on définit également, pour les fonctions d'énergie finie, une fonction caractéristique, appelée *fonction d'autocorrélation* et définie par :

$$\phi_{xx}(\tau) = \int_{-\infty}^{\infty} x(t)x^*(t-\tau)dt = \langle x(t), x(t-\tau) \rangle \quad (2.57)$$

La transformée de Fourier de cette fonction est donnée par :

$$S_{xx}(f) = \int_{-\infty}^{\infty} \phi_{xx}(t)e^{-j\omega t} dt \quad (2.58)$$

Cette fonction est appelée *densité spectrale d'énergie* car son intégrale sur l'axe des fréquences est égale à l'énergie du signal :

$$\int_{-\infty}^{\infty} S_{xx}(f)e^{j\omega t} df = \phi_{xx}(0) = E \quad (2.59)$$

Nous verrons tout l'intérêt de ces fonctions au chapitre 5, consacré aux signaux aléatoires (qui sont des fonctions à puissance finie).

Notons pour terminer que l'intégrale de convolution est elle-même une fonction d'autocorrélation particulière (et est donc également basée sur des calculs de produit scalaire) :

$$x(t) * y(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau = \langle x(t), y(\tau-t) \rangle \quad (2.60)$$

### 2.3.5 Notation en $\omega$

On représente parfois la transformée de Fourier (et les séries de Fourier) en utilisant la variable de pulsation  $\omega$  plutôt que la fréquence  $f$ .

Cette notation en  $F(\omega)$  a pour avantage de rendre l'écriture plus compacte. Puisqu'on ne décompose que sur des exponentielles imaginaires,  $\omega$  apparaît en effet tout naturellement. La notation en  $F(f)$  utilisée ici devrait en principe faire

apparaître explicitement  $f$  dans les expressions utilisées jusqu'ici. Ainsi, l'expression (2.42) devrait s'écrire :

$$f(t) = \int_{-\infty}^{\infty} F(f)e^{j2\pi ft} df \tag{2.61}$$

Pour ne pas alourdir les notations, nous avons maintenu  $\omega$  dans les expressions de  $F(f)$ , en supposant implicitement que le lecteur remplacera de lui-même automatiquement  $\omega$  par  $2\pi f$ .

Par contre, la notation en  $f$  a pour avantage de faire ressortir la symétrie entre la transformée de Fourier et son inverse. Elle élimine les nombreux facteurs  $2\pi$  qui apparaissent dans les propriétés de la transformée lorsqu'elles sont exprimées en  $\omega$ . Ainsi par exemple, l'expression (2.61) devient, en  $\omega$  :

$$f(t) = \int_{-\infty}^{\infty} F(f)e^{j2\pi ft} df = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} d\omega \tag{2.62}$$

En principe, lorsqu'une fonction  $f(t)$  possède une transformée  $F(f)$ , le passage à  $F(\omega)$  est immédiat :  $F(\omega)=F(f)$ . Le changement d'échelle de  $F(f)$  en  $F(\omega)$  ne change en effet rien à la valeur de la fonction ; seul l'axe des abscisses change, dans la représentation. Ainsi par exemple, l'expression analytique de la transformée de  $rect(t)$  n'est pas modifiée (Fig. 2.42).

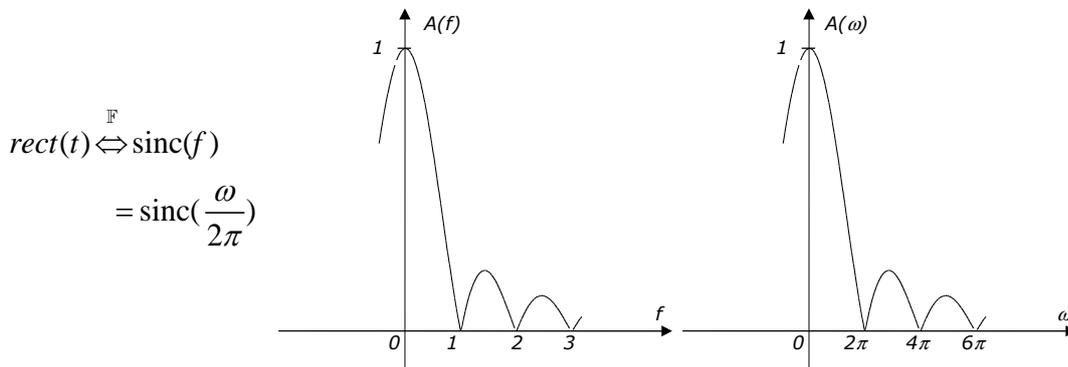


Fig. 2.42 Transformée de Fourier en  $f$  et en  $\omega$  : cas général

Le seul cas où le changement d'échelle modifie la valeur de la transformée est celui de l'impulsion de Dirac. En effet, l'impulsion de Dirac n'est pas définie en tant que fonction (par sa valeur), mais uniquement est tant que distribution (à travers une intégration sur sa variable de définition). Ainsi, l'écriture :

$$\int_{-\infty}^{\infty} \delta(f)df = 1 = \int_{-\infty}^{\infty} \delta(\omega) d\omega \tag{2.63}$$

impose que  $\delta(f) = 2\pi\delta(\omega)$  (Fig. 2.43). Autrement dit, l'impulsion de Dirac est modifiée par l'opération de changement d'échelle. Il n'y a rien d'étonnant à cela, si on se souvient que l'impulsion de Dirac  $\delta(t)$  peut être vue comme la limite de  $rect(t/a)/a$  pour tendant vers 0 (Fig. 1.33), et que le poids de l'impulsion est la surface sous-tendue : un changement d'échelle sur  $rect(t/a)/a$  modifie la largeur du rectangle, et donc sa surface.

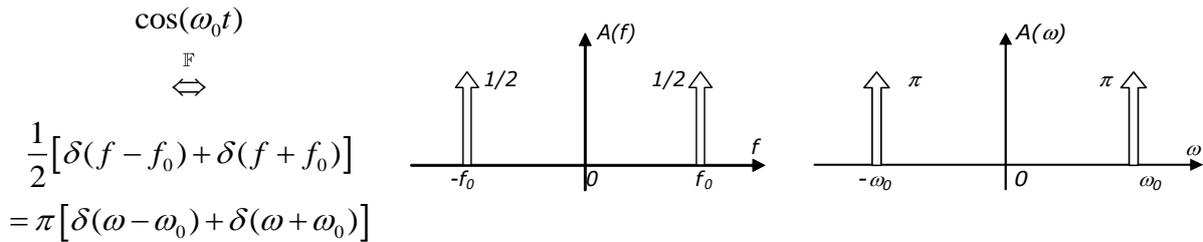


Fig. 2.43 Transformée de Fourier en  $f$  et en  $\omega$  : cas des transformées en  $\delta(f)$

## 2.4 Analyse fréquentielle des signaux à temps discret

### 2.4.1 Transformée de Fourier à temps discret (TFTD)

Pour étudier la contenu fréquentiel d'un signal *signal à temps discret* composé d'une séquence d'échantillons  $\{f(n)\}$ , il faut associer aux échantillons une échelle de temps. On le fait en créant la fonction (analogique!)  $f^+(t)$  composée d'une suite d'impulsions de Dirac espacées d'un temps  $T_e$  appelé *période d'échantillonnage*<sup>5</sup>, et de valeurs égales aux échantillons  $f(n)$  (Fig. 2.44, haut):

$$f^+(t) = \sum_{n=-\infty}^{\infty} f(n)\delta(t - nT_e) \tag{2.64}$$

Dans la suite, nous appellerons  $F_e = 1/T_e$  *fréquence d'échantillonnage* et  $\omega_e = 2\pi F_e$  *pulsation d'échantillonnage*.

La *transformée de Fourier à temps discret* d'un signal à temps discret  $\{f(n)\}$  est alors définie comme la transformée de Fourier du signal  $f^+(t)$  qui lui est associé. Ce dernier est en effet décomposable sur les fonctions exponentielles imaginaires :

$$f^+(t) = \int_{-\infty}^{\infty} F^+(f)e^{j\omega t} df \tag{2.65}$$

Les coefficients  $F^+(f)$  de cette décomposition sont obtenus par projection :

$$F^+(f) = \int_{-\infty}^{\infty} f^+(t)e^{-j\omega t} dt = \int_{-\infty}^{\infty} \left( \sum_{n=-\infty}^{\infty} f(n)\delta(t - nT_e) \right) e^{-j\omega t} dt = \sum_{n=-\infty}^{\infty} f(n)e^{-j\omega nT_e} \tag{2.66}$$

que l'on réécrit plus simplement :

$$F^+(F) = \sum_{n=-\infty}^{\infty} f(n)e^{-jn2\pi F} \quad \text{avec} \quad F = \frac{T_e}{T} = \frac{f}{F_e} \tag{2.67}$$

où  $F$  est la *fréquence normalisée*<sup>6</sup>, sans dimension.

On définit souvent aussi la *pulsation normalisée*  $\varphi^7$  (parfois aussi notée  $\Omega$ ), comme égale à  $2\pi F$ , ce qui donne la notation plus compacte :

<sup>5</sup> On verra en effet au chapitre suivant que la plupart du temps un signal à temps discret est le résultat de l'échantillonnage d'un signal continu.

<sup>6</sup>  $F$  est égale à la fréquence  $f$  si la période d'échantillonnage est égale à 1.

$$F^+(\varphi) = \sum_{n=-\infty}^{\infty} f(n)e^{-jn\varphi} \quad \text{avec} \quad \varphi = 2\pi F = 2\pi \frac{f}{F_e} \quad (2.68)$$

L'examen de ces dernières relations montre que  $F^+(F)$  est une fonction périodique (Fig. 2.44, bas) en  $F$  de période 1 (ce qui correspond à une période de  $f_e$  en fréquence non normalisée), et que  $F^+(\varphi)$  est une fonction périodique en  $\varphi$  de période  $2\pi$  (ce qui correspond à une période de  $\omega_e$  en pulsation non normalisée).

On peut dès lors réécrire l'expression (2.65) au sens des distributions sous la forme :

$$\int_{-\infty}^{\infty} F^+(f)e^{j\omega t} df = \begin{cases} 0 & \text{si } t \neq nT_e \\ \frac{1}{f_e} \int_{-f_e/2}^{f_e/2} F^+(f)e^{j\omega nT_e} df & \text{si } t = nT_e \end{cases} \quad (2.69)$$

$$= \begin{cases} 0 & \text{si } t \neq nT_e \\ \int_{-1/2}^{1/2} F^+(F)e^{jn\varphi} dF & \text{si } t = nT_e \end{cases}$$

On en déduit la formule d'inversion en  $F$  ou en  $\varphi$ :

$$f(n) = \int_{-1/2}^{1/2} F^+(F)e^{jn\varphi} dF \quad \text{ou} \quad f(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} F^+(\varphi)e^{jn\varphi} d\varphi \quad (2.70)$$

Le signal  $\{f(n)\}$  et la fonction  $F^+(F)$  constituent une paire de transformées de Fourier à temps discret, notée :

$$\{f(n)\} \stackrel{TFTD}{\Leftrightarrow} F^+(F) \quad (2.71)$$

ou plus simplement, en laissant tomber l'indice « + » pour ne pas alourdir les notations <sup>8</sup>:

$$\{f(n)\} \stackrel{TFTD}{\Leftrightarrow} F(F) \quad (2.72)$$

<sup>7</sup>  $\varphi$  est égale à la pulsation  $\omega$  si la période d'échantillonnage est égale à 1.

<sup>8</sup> Il ne faudra cependant pas oublier que  $F(F)$  (où on a oublié l'indice "+") désigne une TFTD. Nous verrons en effet au chapitre 3 que lorsqu'on échantillonne un signal continu  $f(t)$  de transformée de Fourier (non discrète)  $F(f)$ , la TFTD  $F^+(F)$  obtenue à partir de ses échantillons est donnée par :  $F^+(F) = F(f)/T_e$  (où  $T_e$  est le pas d'échantillonnage et  $f = F \cdot F_e$ ). On écrira donc, par simplicité :  $F(F) = F(f)/T_e$ , mais il ne faudra donc pas confondre  $F(f)$  et  $F(F)$ : ce sont *pas* les mêmes fonctions.

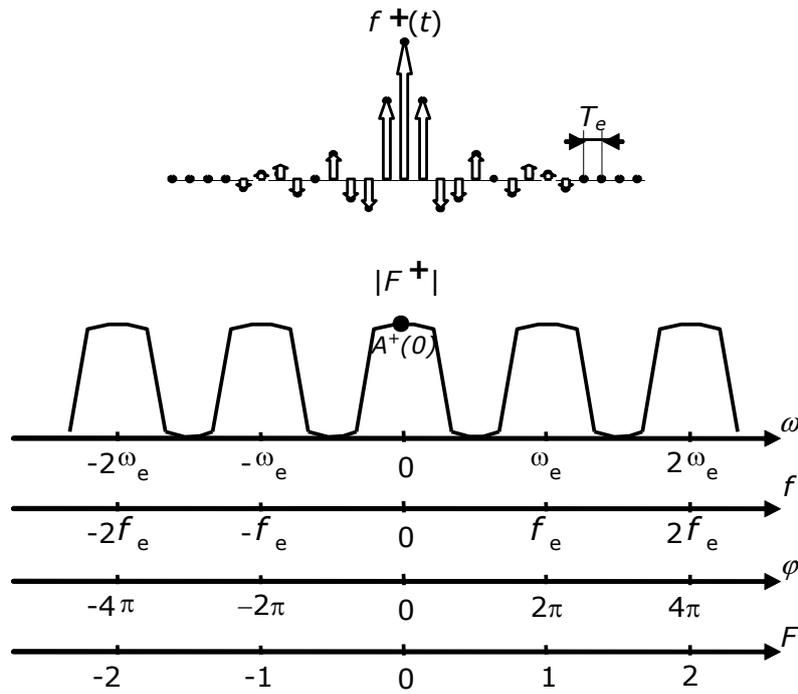


Fig. 2.44 Un signal numérique non périodique (haut) et la TFTD associée (en module)

Comme c'était déjà le cas pour la transformée de Fourier, la TFTD  $F(F)$  d'un signal  $\{f(n)\}$  réel est telle que  $F(-F) = F(F)^*$ , c'est-à-dire que les valeurs de  $F(F)$  entre 0 et 1 ne sont rien d'autre que les conjuguées des valeurs de  $F(F)$  entre  $-1$  et 0 : le module de la TFTD est pair et son argument est impair.

### Interprétation géométrique

L'expression (2.67) possède une interprétation géométrique immédiate : pour toute fréquence normalisée  $F$ ,  $F(F)$  est la somme vectorielle de vecteurs complexes de type  $f(n)e^{jn\varphi}$ .

#### Exemple 2.18

Soit le signal numérique causal  $\{f(n)\} = \{1, 2, 1, 0, 0, 0, \dots\}$ . On peut facilement calculer la valeur de sa TFTD  $F(F)$  pour  $F=1/16$  et  $1/8$ , c'est-à-dire  $F(\varphi)$  pour  $\varphi = \pi/8$  et  $\pi/4$  (Fig. 2.45) :

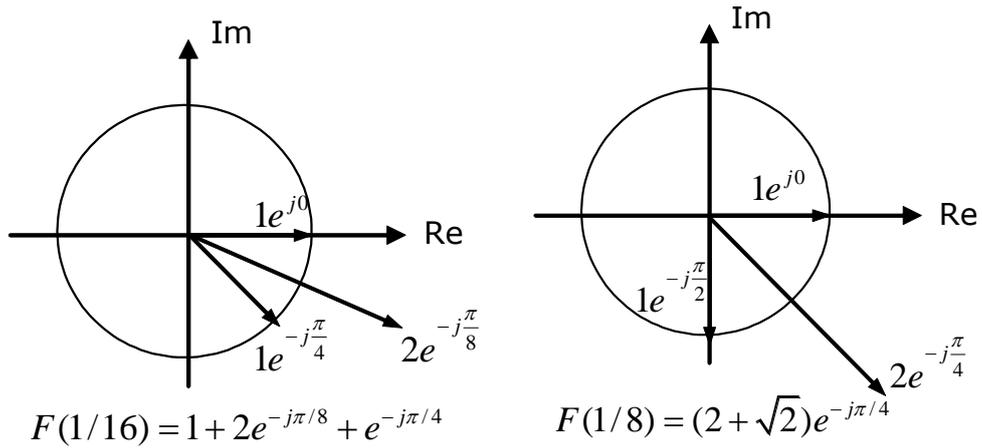


Fig. 2.45 Calcul géométrique de la TFTD pour  $F=1/16$  et  $1/8$  ( $\varphi = \pi/8$  et  $\pi/4$ )

L'estimation complète et l'affichage de la TFTD  $F(F)$  d'un signal  $x(n)$  peut être faite sous Matlab en utilisant la fonction `freqz(x,1)` où  $x$  est le vecteur de la séquence  $x(n)$  (nous verrons plus loin pourquoi le deuxième argument doit être 1) :

```
freqz([1 2 1],1) % calcule et affiche la TFTD de x sur 512 points entre F=0 et 1/2
```

On constate (Fig. 2.46) que Matlab n'affiche la TFTD que dans la zone utile  $\varphi=0 \dots \pi$ .

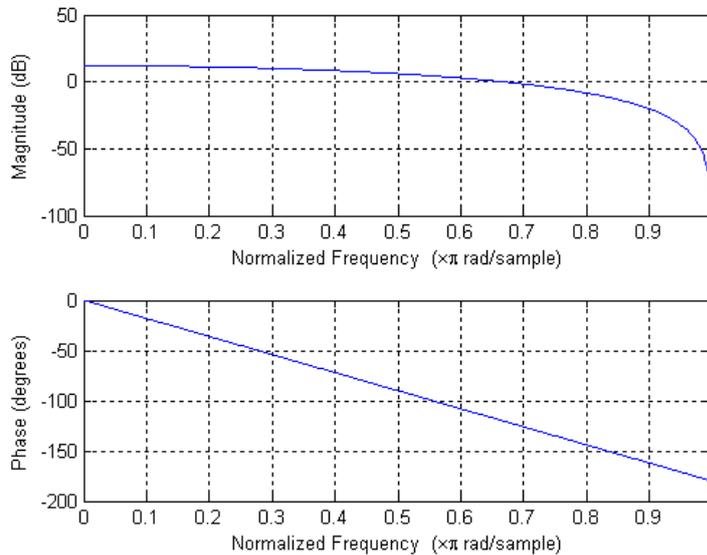


Fig. 2.46 Affichage de la TFTD par Matlab

### Propriétés

Les propriétés de la TFTD sont les mêmes que celles de la transformée de Fourier (la TFTD n'étant rien d'autre elle-même qu'une transformée de Fourier appliquée à un signal de type particulier). Elles sont présentées au Tableau 2.2. La

propriété de produit temporel et le théorème de Parseval prennent ici une forme particulière (que l'on admettra sans démonstration<sup>9</sup>), due à la périodicité des transformées : les intégrales sont calculées sur une seule période (pour  $F$  allant de  $-1/2$  à  $+1/2$ ).

(linéarité)	$\sum_i a_i f_i(n)$	$\Leftrightarrow$	$\sum_i a_i F_i(F)$
(dilatation)	$f(an)$	$\Leftrightarrow$	$\frac{1}{ a } F(F/a)$
(retard)	$f(n-n_0)$	$\Leftrightarrow$	$F(F) \exp(-jn_0\varphi)$
(convolution)	$f(n)*g(n)$	$\Leftrightarrow$	$F(F)G(F)$
(produit)	$f(n)g(n)$	$\Leftrightarrow$	$\int_{-1/2}^{1/2} F(F-\Phi)G(\Phi)d\Phi$
(produit scalaire)	$\sum_n f(n)g^*(n)$	$\Leftrightarrow$	$\int_{-1/2}^{1/2} F(F)G^*(F)dF$
(Parseval)	$\sum_n  f(n) ^2$	$\Leftrightarrow$	$\int_{-1/2}^{1/2}  F(F) ^2 dF$

Tableau 2.2 Propriétés principales de la TFTD

### 2.4.2 Séries de Fourier à temps discret (SFTD)

Dans le cas d'un signal *périodique* à temps discret de période égale à  $n_0$  échantillons ( $n_0$  entier), que nous noterons  $\{f_{n_0}(n)\}$ , l'expression (2.64) peut se réécrire sous la forme de la fonction (analogique)  $f_{T_0}^+(t)$ , qui serait obtenue par périodification d'une fonction correspondant  $f^+(t)$  de départ :

$$f_{T_0}^+(t) = \sum_{k=-\infty}^{\infty} f^+(t - kn_0T_e) \tag{2.73}$$

où l'on a explicité le fait que la période  $T_0$  doit correspondre à un nombre entier  $n_0$  d'échantillons :  $T_0 = n_0T_e$ .

L'application de la TFTD à un signal périodique  $\{f_{n_0}(n)\}$  de période  $n_0$  échantillons correspond à la projection de la fonction  $f_{T_0}^+(t)$  sous-jacente sur les fonctions de base  $\{e^{j\omega t} (\omega \in [-\infty, +\infty])\}$ . Ceci ne peut se faire qu'au sens des distributions, suivant (2.49) :

---

<sup>9</sup> La TFTD d'un produit de signaux numériques  $f(n)$  et  $g(n)$  peut se comprendre de la façon suivante : elle est donnée par la convolution des transformées de Fourier  $F^+(f)$  et  $G^+(f)$ . Or, la convolution est calculée par produit scalaire, et les fonctions  $F^+(f)$  et  $G^+(f)$  sont périodiques de période  $f_e$ . La TFTD est donc obtenue par convolution  $F^+(f)$  et  $G^+(f)$  sur une seule période (de 0 à  $f_e$ ), le tout divisé par la période  $f_e$ . Ou, si on fait le calcul en fréquence normalisée, sur une période (de 0 à 1), le tout divisé par 1. Les deux propriétés suivantes s'expliquent de la même façon.

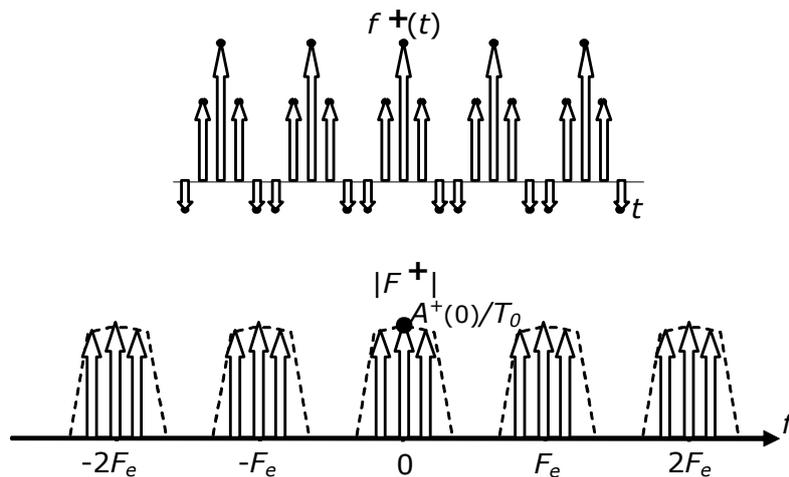
$$\begin{aligned}
 \langle f_{T_0}^+(t), e^{j\omega t} \rangle &= \begin{cases} 0 & \text{si } f \neq kf_0 \\ \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} f_{T_0}^+(t) e^{-j\omega t} dt & \text{si } f = kf_0 \end{cases} \quad (f_0 = \frac{1}{T_0}) \\
 &= \begin{cases} 0 & \text{si } f \neq kf_0 \\ \frac{1}{T_0} \sum_{n=0}^{n_0-1} f(n) e^{-j\omega n T_e} & \text{si } f = kf_0 \end{cases} \quad (2.74) \\
 &= \begin{cases} 0 & \text{si } F \neq kF_0 \\ \frac{1}{T_0} \sum_{n=0}^{n_0-1} f(n) e^{-jn\varphi} & \text{si } F = kF_0 \end{cases} \quad (F_0 = \frac{1}{n_0})
 \end{aligned}$$

La TFTD du signal périodique est alors, comme dans le cas non discret, une suite d'impulsions de Dirac en fréquence, dont les valeurs sont données par les produits scalaires obtenus par (2.74):

$$F_{T_0}(F) = \sum_{k=-\infty}^{\infty} F_k \delta(\varphi - k\varphi_0) \quad \text{avec} \quad F_k = \frac{1}{T_0} \sum_{n=0}^{n_0-1} f(n) e^{-jn k \varphi_0} \quad (2.75)$$

On constate, comme dans le cas d'un signal à temps discret non périodique, que  $F_{T_0}(F)$  est périodique de période 1.

Les signaux périodiques à temps discret possèdent donc un spectre de raies de pas  $1/n_0$  (en  $F$ ; ou  $2\pi/n_0$  en  $\varphi$ ; ou  $F_e/n_0$  en fréquence non normalisée). Ce spectre de raies est lui-même périodique (Fig. 2.47) de période 1 (en  $F$ , ou  $2\pi$  en  $\varphi$ ; ou  $F_e$  en fréquence non normalisée). L'amplitude des raies est égale l'amplitude du spectre du signal non périodique sous-jacent  $f^+(t)$  (à partir duquel le signal  $f^+_{T_0}(t)$  a été obtenu), divisée par la période  $T_0$ .<sup>10</sup>



<sup>10</sup> La transformée de Fourier d'un signal périodique numérique est donc liée à la transformée de Fourier du signal non périodique sous-jacent de la même façon qu'en analogique. En particulier, les expressions (2.55) et (2.56) restent valables.

Fig. 2.47 Signal numérique périodique (haut) et TFTD correspondante, en module (bas) <sup>11</sup>

La somme continue (2.65) se réduit finalement à une somme discrète, que l'on appelle *série de Fourier à temps discret* associée à  $f_{T_0}^+(t)$  :

$$f_{T_0}^+(t) = \sum_{k=-\infty}^{\infty} F_k e^{jk\omega_0 t} \quad (2.76)$$

## 2.5 Réponse en fréquence d'un SLI numérique

### 2.5.1 Réponse de régime à une exponentielle imaginaire numérique

Soit un SLI numérique excité par l'exponentielle imaginaire  $x(n) = e^{jn\varphi}$  ( $n = -\infty, \dots, +\infty$ ). La réponse de régime est donnée par :

$$\begin{aligned} y(n) &= x(n) * h(n) = \sum_{i=-\infty}^{\infty} h(i)x(n-i) \\ &= \sum_{i=-\infty}^{\infty} h(i)e^{j(n-i)\varphi} \\ &= e^{jn\varphi} \sum_{i=-\infty}^{\infty} h(i)e^{-ji\varphi} \end{aligned} \quad (2.77)$$

Ce qui donne finalement :

$$y(n) = e^{jn\varphi} H(\varphi) \quad \text{avec } H(\varphi) = \sum_{n=-\infty}^{\infty} h(n)e^{-jn\varphi} \quad (2.78)$$

On constate donc que l'exponentielle imaginaire (numérique) est une *fonction propre* de tout SLI numérique : la réponse de régime d'un SLI à une exponentielle imaginaire n'est autre que l'exponentielle imaginaire d'entrée, multiplié par un facteur complexe  $H(\varphi)$  qui dépend de la pulsation normalisée  $\varphi$  de l'exponentielle.

La grandeur  $H(\varphi)$  ainsi définie est appelée *fonction de réponse en fréquence*, ou *transmittance isochrone*, du SLI. L'équation (2.78) montre par ailleurs, par comparaison avec (2.68), que la réponse en fréquence d'un SLI est la TFTD de sa réponse impulsionnelle.

La réponse en régime sinusoïdal d'un SLI est évidemment directement liée à la réponse en fréquence. Sachant que  $x(n) = A \cos(n\varphi + \theta) = A/2(e^{j(n\varphi + \theta)} + e^{-j(n\varphi + \theta)})$ , il vient immédiatement :

---

<sup>11</sup> Remarquons que la Fig. 2.47 donne la TFTD en fonction de la fréquence *non normalisée* ( $f$ ). En effet, en vertu de 2.3.5, un affichage en fonction de  $F$  ferait apparaître des impulsions de Dirac  $T_e$  fois plus grandes que celles de la Fig. 2.47.

$$\begin{aligned}
 y(n) &= A \frac{H(\varphi)e^{j(n\varphi+\theta)} + H(-\varphi)e^{-j(n\varphi+\theta)}}{2} \quad \text{avec } H(\varphi) = |H(\varphi)|e^{j\arg(H(\varphi))} \\
 &= A \frac{H(\varphi)e^{j(n\varphi+\theta)} + H^*(\varphi)e^{-j(n\varphi+\theta)}}{2} \\
 &= A |H(\varphi)| \cos(n\varphi + \theta + \arg(H(\varphi)))
 \end{aligned}
 \tag{2.79}$$

**Exemple 2.19**

Calculons sous Matlab la réponse en fréquence des SLI suivants :

- Passe-bas :  $h(n)=0.9^n \varepsilon(n)$  (Fig. 2.48)
- Passe-haut :  $h(n)=(-0.9)^n \varepsilon(n)$  (Fig. 2.49)
- Passe-bande :  $h(n)=2(0.9)^n \cos(n\pi/2) \varepsilon(n)$  (Fig. 2.50)

```

h=0.9.^[0 :50] ;
stem(h) ;
figure(2) ;
freqz(h,1) ;
    
```

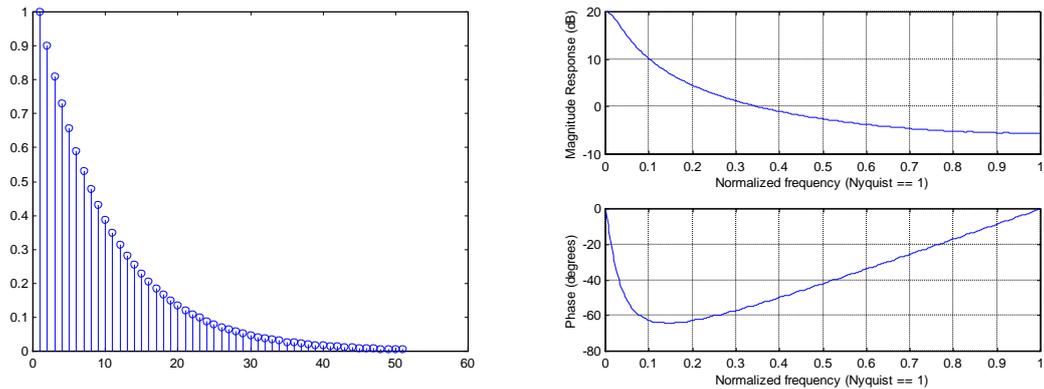


Fig. 2.48 Réponse impulsionnelle et réponse en fréquence du filtre passe-bas

```

h=(-0.9).^[0 :50] ;
stem(h) ;
figure(2) ;
freqz(h,1) ;
    
```

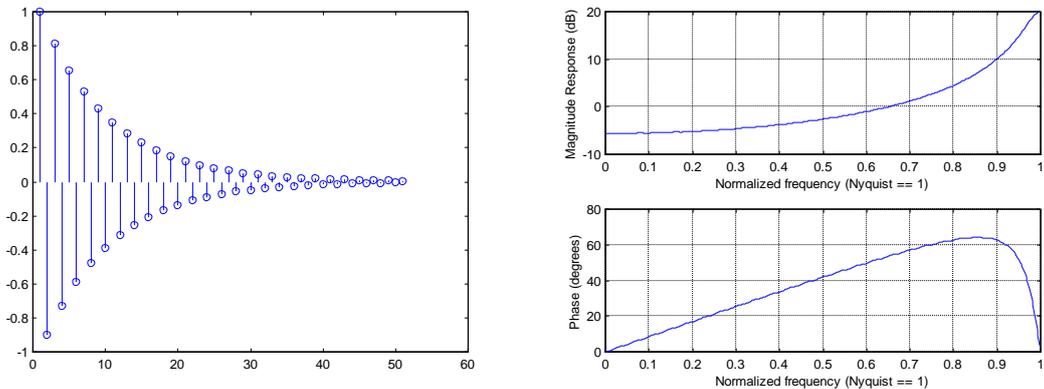


Fig. 2.49 Réponse impulsionnelle et réponse en fréquence du filtre passe-haut

```
h=2*(0.9).^[0 :50].*cos([0 :50]*pi/2) ;
stem(h) ;
figure(2) ;
freqz(h,1) ;
```

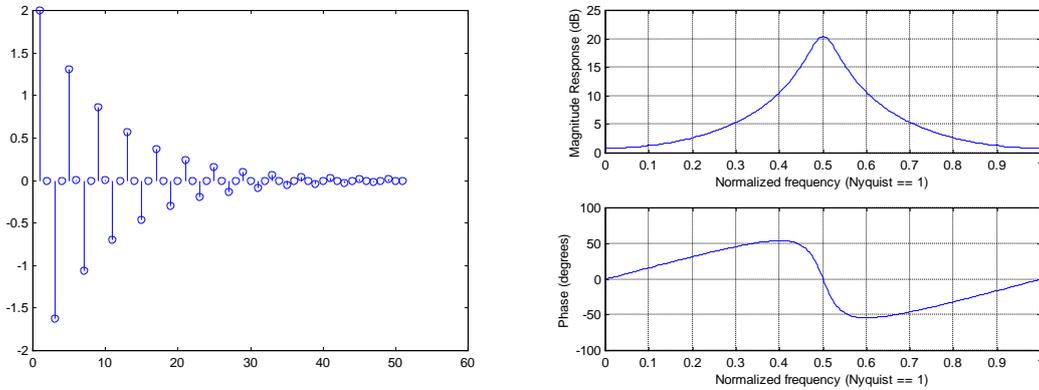


Fig. 2.50 Réponse impulsionnelle et réponse en fréquence du filtre passe-bande

### 2.5.2 Lien entre réponse opérationnelle et réponse isochrone

Tout comme en analogique, il est possible d’obtenir la réponse en fréquence d’un SLI à partir de l’équation aux différences finies qui le caractérise :

$$y(n) + \sum_{i=1}^N a_i y(n-i) = \sum_{i=0}^M b_i x(n-i) \tag{2.80}$$

En substituant  $e^{jn\varphi}$  à  $x(n)$  et  $H(\varphi)e^{jn\varphi}$  à  $y(n)$ <sup>12</sup>, il vient :

$$H(\varphi)e^{jn\varphi} + \sum_{i=1}^N a_i H(\varphi)e^{j(n-i)\varphi} = \sum_{i=0}^M b_i e^{j(n-i)\varphi} \tag{2.81}$$

ce qui donne finalement, en isolant  $H(\varphi)$  :

$$H(\varphi) = \frac{\sum_{i=0}^M b_i e^{-ji\varphi}}{1 + \sum_{i=1}^N a_i e^{-ji\varphi}} \tag{2.82}$$

On prendra l’habitude, pour la suite, d’écrire cette relation sous la forme :

$$H(\varphi) = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \Bigg|_{z=e^{j\varphi}} = H(z) \Bigg|_{z=e^{j\varphi}} \tag{2.83}$$

où l’on retrouve les séquences  $\{b_0, b_1, \dots, b_M\}$  et  $\{1, a_1, \dots, a_N\}$  associées à des polynômes en la variable complexe  $z$ . On lui donne le nom de *transformée en*

<sup>12</sup> Il est important de noter qu’on ne peut ainsi substituer  $H(\varphi)e^{jn\varphi}$  à  $y(n)$  que si une réponse de régime existe, c’est à dire si le filtre est stable.

$z$  associée à l'équation aux différences finies (2.80)<sup>13</sup>. On constate donc que la réponse en fréquence d'un SLI numérique est égale à sa transformée en  $z$  calculée sur le cercle de rayon unité.

### Exemple 2.20

Un filtre numérique passe-bande (que l'on supposera stable) est défini par l'équation aux différences finies :

$$y(n) + 0.81 y(n-2) = x(n) - x(n-2)$$

On demande :

- de déterminer l'expression analytique de sa réponse en fréquence
- de déterminer sa réponse de régime pour l'entrée  $x(n) = 10 + 10 \cos(n\pi/2) + 10 \cos(n\pi)$
- de calculer et d'afficher la réponse en fréquence sous Matlab.

La filtre est de type RII. La réponse en fréquence est donnée, soit par la TTFD de  $h(n)$ , que l'on pourrait calculer à partir de l'équation aux différences finies, mais qui est théoriquement infinie. Il est plus simple de calculer  $H(\varphi)$  à partir des coefficients  $a_i$  et  $b_i$  :

$$H(\varphi) = \frac{1 - z^{-2}}{1 + 0.81z^{-2}} \Bigg|_{z=e^{j\varphi}} = \frac{1 - e^{-j2\varphi}}{1 + 0.81e^{-j2\varphi}}$$

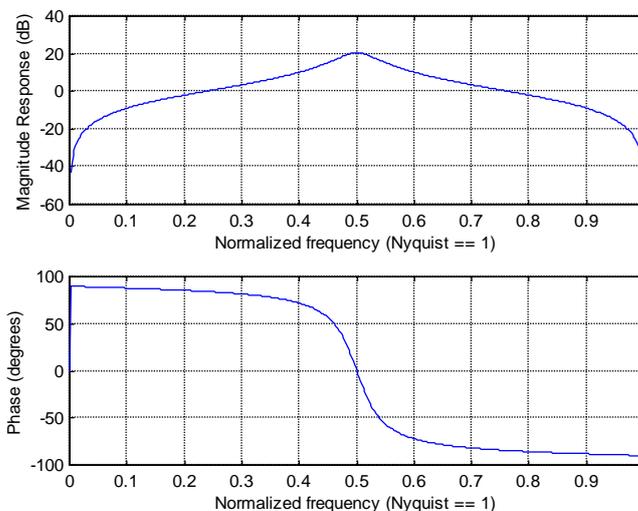
Il vient donc :

$$H(0) = 0 \quad H(\pi/2) = 2/0.19 = 10.53 \quad H(\pi) = 0$$

On en déduit que la réponse de régime sinusoïdal est donnée par  $105.3 \cos(n\pi/2)$ , ce qui constitue une amplification d'à peu près 20dB.

Matlab permet très facilement de calculer et d'afficher la réponse en fréquence d'un filtre numérique connaissant son équation aux différences finies. La fonction `freqz` est en effet prévue pour être utilisée selon le synopsis `freqz(b,a)` où  $b$  et  $a$  sont les vecteurs des  $a_i$  et  $b_i$  :

`freqz([1 0 -1],[1 0 0.81]) ;`



<sup>13</sup> L'origine de cette notation sera développée dans un chapitre ultérieur consacré à la transformée en  $z$ .

Fig. 2.51 Réponse en fréquence du filtre passe-bande

On retrouve bien sur ce graphique les 20dB d’amplification calculés plus haut.

### 2.5.3 Interprétation géométrique

L’équation (2.83) se prête à une interprétation géométrique similaire à celle permettant de pressentir la réponse en fréquence d’un système numérique en fonction de la position de ses pôles et zéros. On peut en effet la réécrire sous la forme :

$$H(\varphi) = H(z) \Big|_{z=e^{j\varphi}} = K z^{-(M-N)} \frac{\prod_{i=1}^M (z - z_i)}{\prod_{i=1}^N (z - p_i)} \Big|_{z=e^{j\varphi}} \quad (2.84)$$

$$|H(\varphi)| = |K| \frac{\prod_{i=1}^M |z - z_i|}{\prod_{i=1}^N |z - p_i|} \Big|_{z=e^{j\varphi}}$$

On en conclut (Fig. 2.52) que le module de la réponse en fréquence, pour une pulsation réduite  $\varphi_0$  donnée, est obtenu (à  $K$  près) en multipliant les normes des vecteurs joignant les zéros du système à un point  $e^{j\varphi_0}$  du cercle de rayon unité, et en divisant le résultat par le produit des normes des vecteurs joignant les pôles à ce même points. De même la réponse en phase correspond à la somme des arguments des vecteurs issus des zéros moins la somme des arguments des vecteurs issus des pôles<sup>14</sup>.

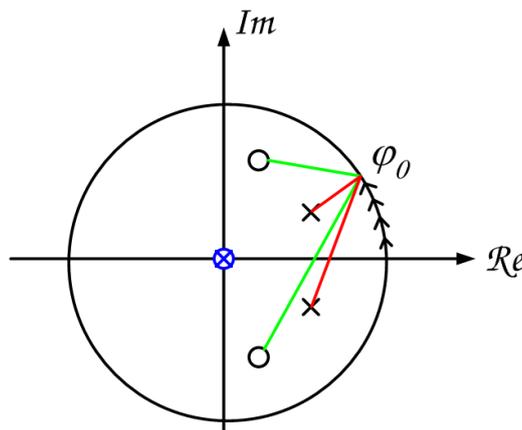


Fig. 2.52 Estimation graphique de la réponse en fréquence d’un SLI numérique.

Il s’ensuit que la position de pôles près (mais à l’intérieur) du cercle de rayon unité à une pulsation  $\theta$  fait apparaître une résonance dans la réponse en

<sup>14</sup> On notera que les zéros ou les pôles en  $z=0$  (contribution en  $z^{-(M-N)}$ ) n’ont pas d’effet sur l’amplitude de la réponse, mais bien sur sa phase.

fréquence pour  $\varphi=\theta$ . De même, des zéros, près du (ou sur le) cercle de rayon unité induisent une antirésonance dans la réponse à fréquence.

### Exemple 2.21

On peut prévoir la réponse en amplitude du filtre numérique passe-bande de l'exercice précédent :

```
zplane([1 0 -1],[1 0 0.81]) ;
```

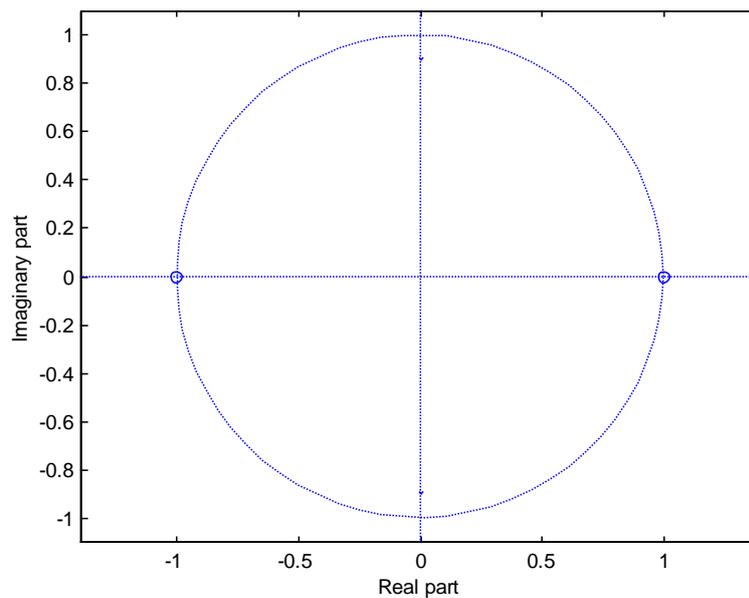


Fig. 2.53 Réponse en fréquence du filtre passe-bande

On constate deux zéros en  $\varphi=0$  et  $\pi$  qui imposeront une réponse nulle en  $F=0$  et  $1/2$ , et deux pôles en  $\varphi=\pm\pi/2$ , proches du cercle de rayon unité. Ces pôles provoqueront une résonance en  $F=1/4$ , qui sera d'autant plus marquée que les pôles se rapprochent du cercle.

## 2.5.4 Résumé comparatif entre SLI analogiques et numériques

Nous pouvons à présent résumer le lien entre systèmes numériques et analogiques sous la forme de deux graphiques (Fig. 2.54 et Fig. 2.55).

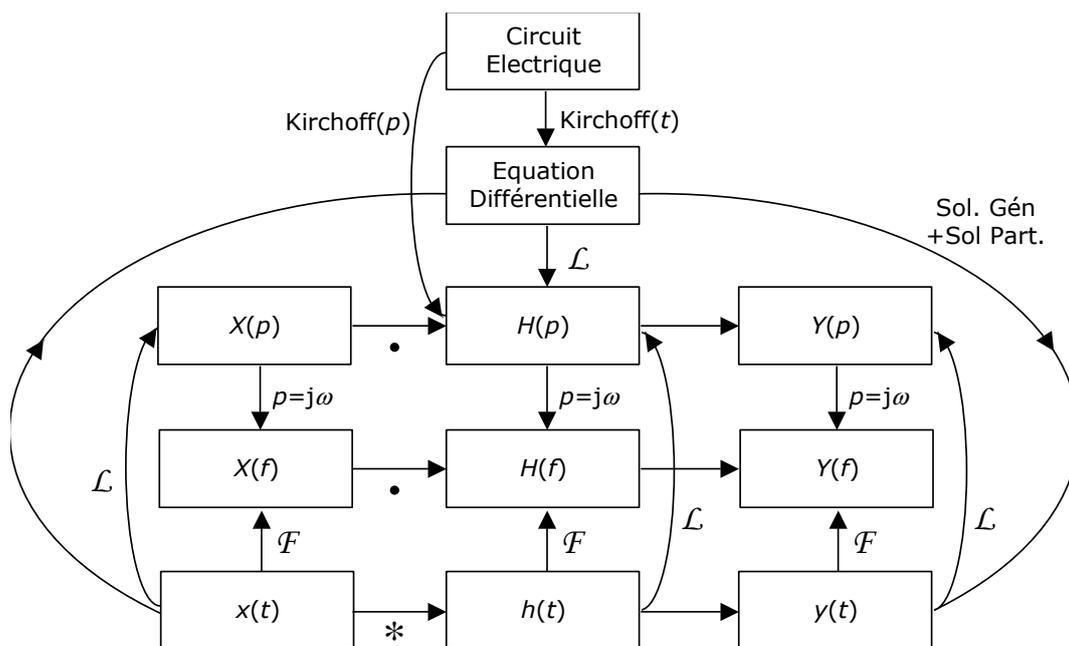


Fig. 2.54 Relations entre éléments d'un SLI analogique

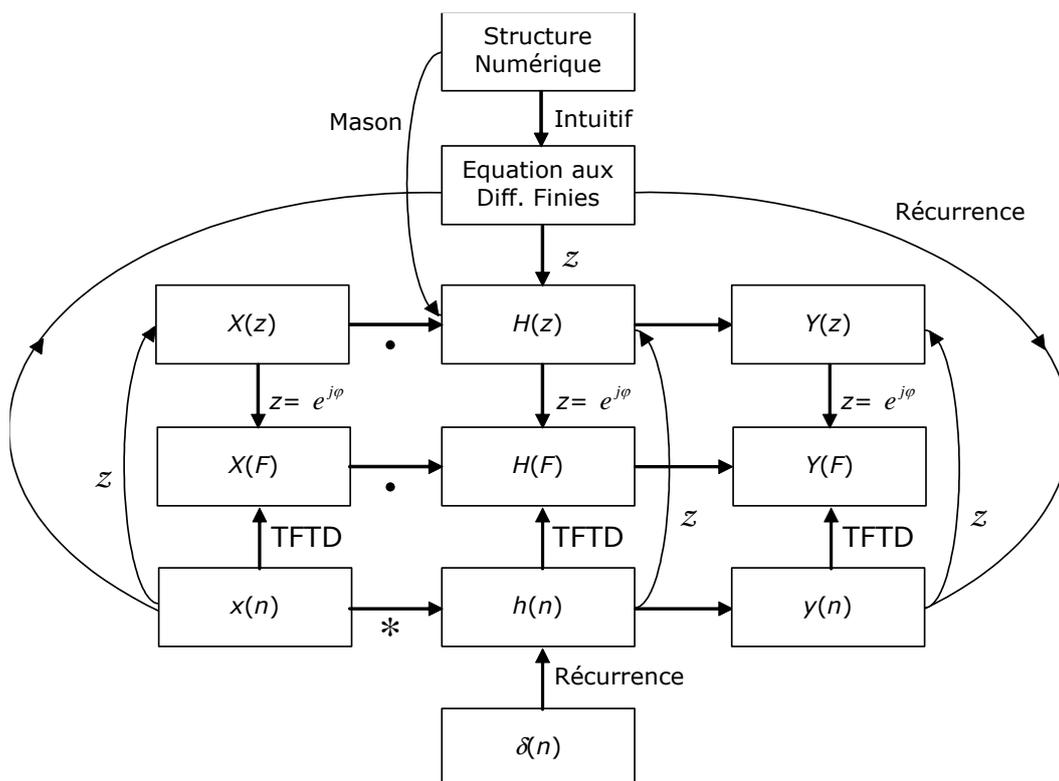


Fig. 2.55 Relations entre éléments d'un SLI numérique

## Exercices

### Exercice 2.1

Un système causal est décrit par l'équation aux différences :

$$y(n) = -0.1 y(n-1) + 0.3 y(n-2) + x(n) + x(n-1)$$

On demande :

- De calculer la fonction de transfert  $H(z)$  du système
- De tracer le diagramme pôles-zéros correspondant
- De calculer la réponse en fréquence  $H(e^{j\omega})$  et d'esquisser le spectre d'amplitude  $|H(e^{j\omega})|$  de  $f=0\dots 400$  Hz en calculant ce spectre aux fréquences  $f=[50, 100, 150$  Hz] ( $F_e=200$  Hz).

Solution

... 2 pôles réels en  $-0.6$  et  $+0.5$ ;  $H(\pi/2)=H(3\pi/2)=.7$  dB;  $H(\pi)=-\infty$ dB

### Exercice 2.2

Soit le filtre défini par la relation :

$$y(n)=x(n)+\alpha y(n-10) , \text{ où } 0<\alpha.$$

On demande :

- a. De déterminer l'expression analytique de sa réponse en fréquence
- b. De tracer son diagramme pôles-zéros (séparer le cas  $\alpha<1$  et le cas  $\alpha>1$ )
- c. D'en déduire une esquisse du graphique du module de sa réponse en fréquence (séparer le cas  $\alpha<1$  et le cas  $\alpha>1$ ).

### Exercice 2.3

Un filtre en peigne est défini par la relation :

$$y(n) = x(n) - x(n - n_0)$$

On demande de déterminer analytiquement sa réponse en fréquence, d'esquisser le graphique de son module à partir de son diagramme pôles-zéros et de justifier l'appellation filtre en peigne.

### Exercice 2.4

Jeu des 8 familles (D'après J. Lebrun, EPFL) (voir Figs1-3 dans les pages suivantes)

Pour chaque diagramme de pôles-zéros (Z1-Z8 ; Fig. 1), retrouver la réponse en fréquence (F1-F8 ; Fig. 2) et la réponse impulsionnelle (I1-I8 ; Fig. 3) correspondantes. JUSTIFIEZ TOUS VOS CHOIX.

**Complément théorique :** La Fig. 1 présente certains systèmes comportant des pôles en dehors du cercle de rayon unité. En général, de tels systèmes sont

instables, et leur réponse en fréquence n'est donc pas définie, ... sauf si l'on calcule uniquement la réponse impulsionnelle  $h(n)$  correspondante pour  $n \leq 0$  et qu'on suppose cette réponse nulle pour  $n > 0$  (on obtient alors des systèmes dits *anti-causaux*).

En effet, si un signal  $x(n)$  causal (c.à.d. nul pour  $n < 0$ ) admet pour transformée en  $z$  :  $X(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + x(3)z^{-3} + \dots$ , alors le signal  $y(n) = x(-n)$  (anti-causal) admet pour transformée en  $z$  :  $Y(z) = x(0) + x(1)z^1 + x(2)z^2 + x(3)z^3 + \dots$

Les zéros du  $Y(z)$  sont donc les inverses (complexes) des zéros de  $X(z)$ , ce qui implique que l'intérieur du cercle de rayon unité correspond alors à l'extérieur de ce même cercle lorsqu'on se restreint aux signaux anti-causaux.

On supposera donc ici que les systèmes de la Fig. 1 dont les pôles sont en dehors du cercle de rayon unité sont anti-causaux, et donc stables.

Solution

Z1 F8 I4; Z2 F6 I7; Z3 F2 I2; Z4 F4 I5; Z5 F1 I8; Z6 F3 I1; Z7 F5 I3; Z8 F7 I6

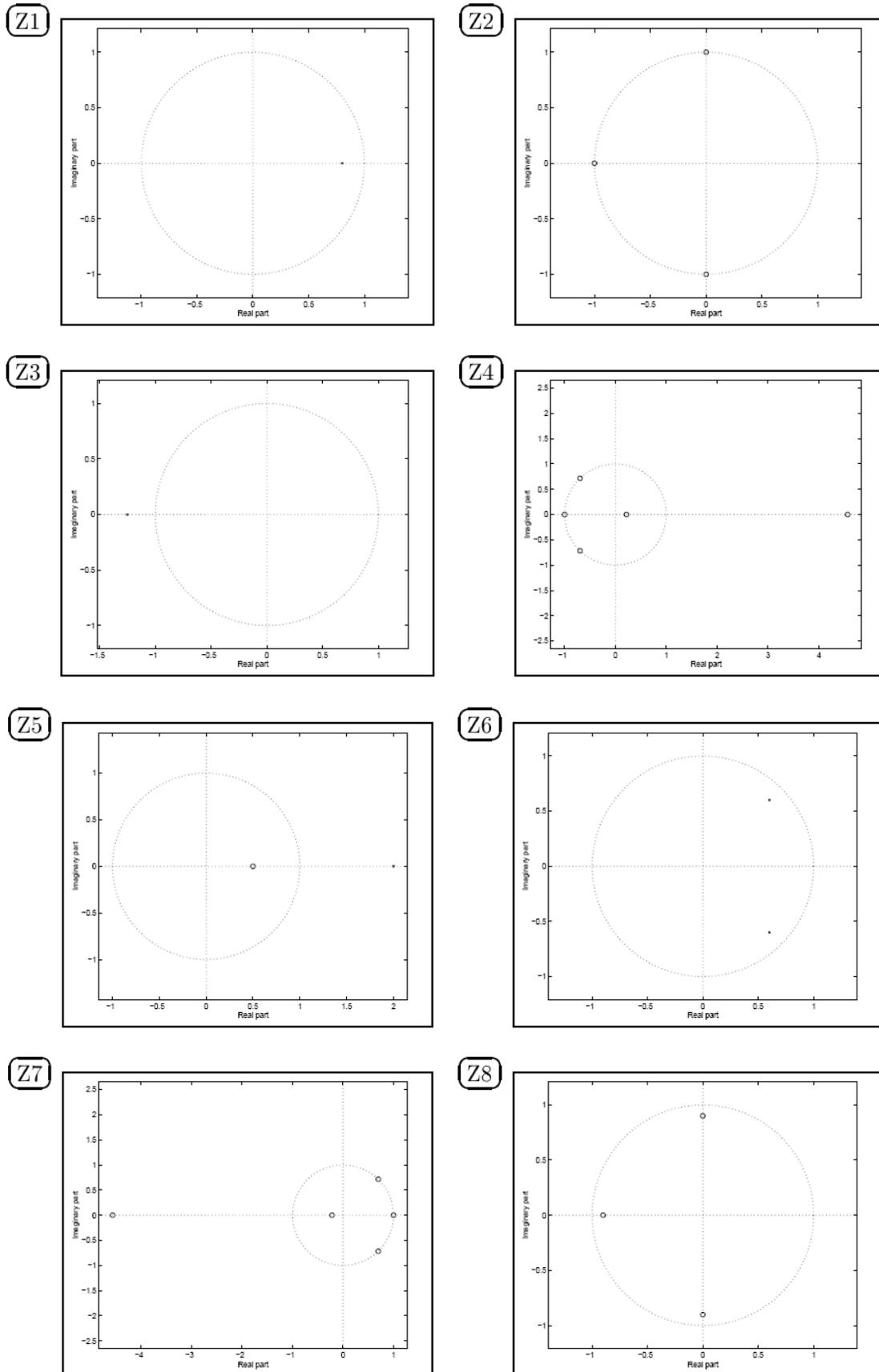
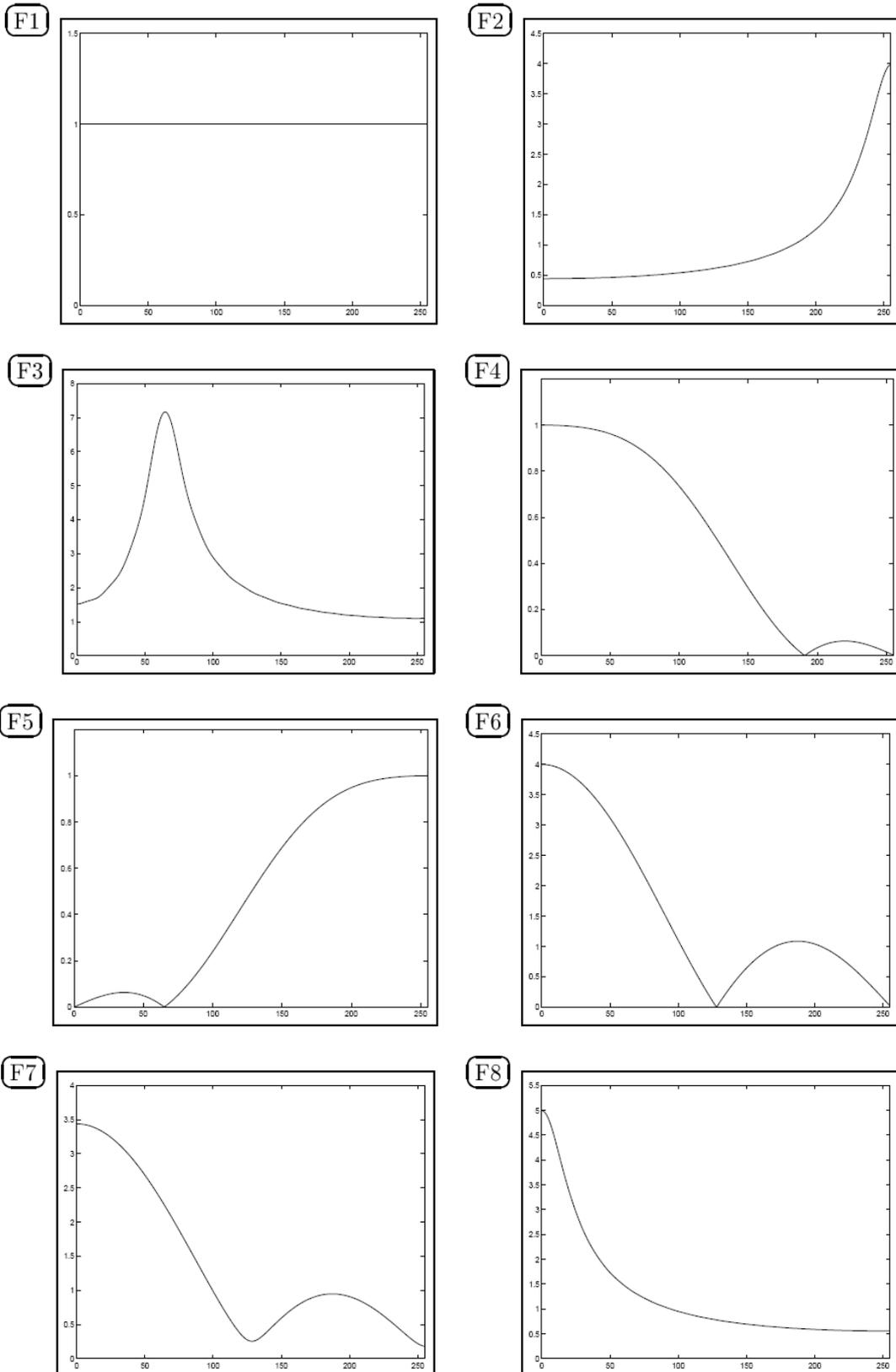


Figure 1: diagramme zéros/pôles dans le plan des  $z$  des 8 différents systèmes linéaires invariants dans le temps.



**Figure 2:** Graphe de l'amplitude des réponses fréquentielles  $|H(e^{j\omega})|$  obtenues par DFT sur 512 points. L'abscisse correspond à la pulsation  $\omega$  (l'indice 255 correspondant à la pulsation  $\omega = \pi$ ).

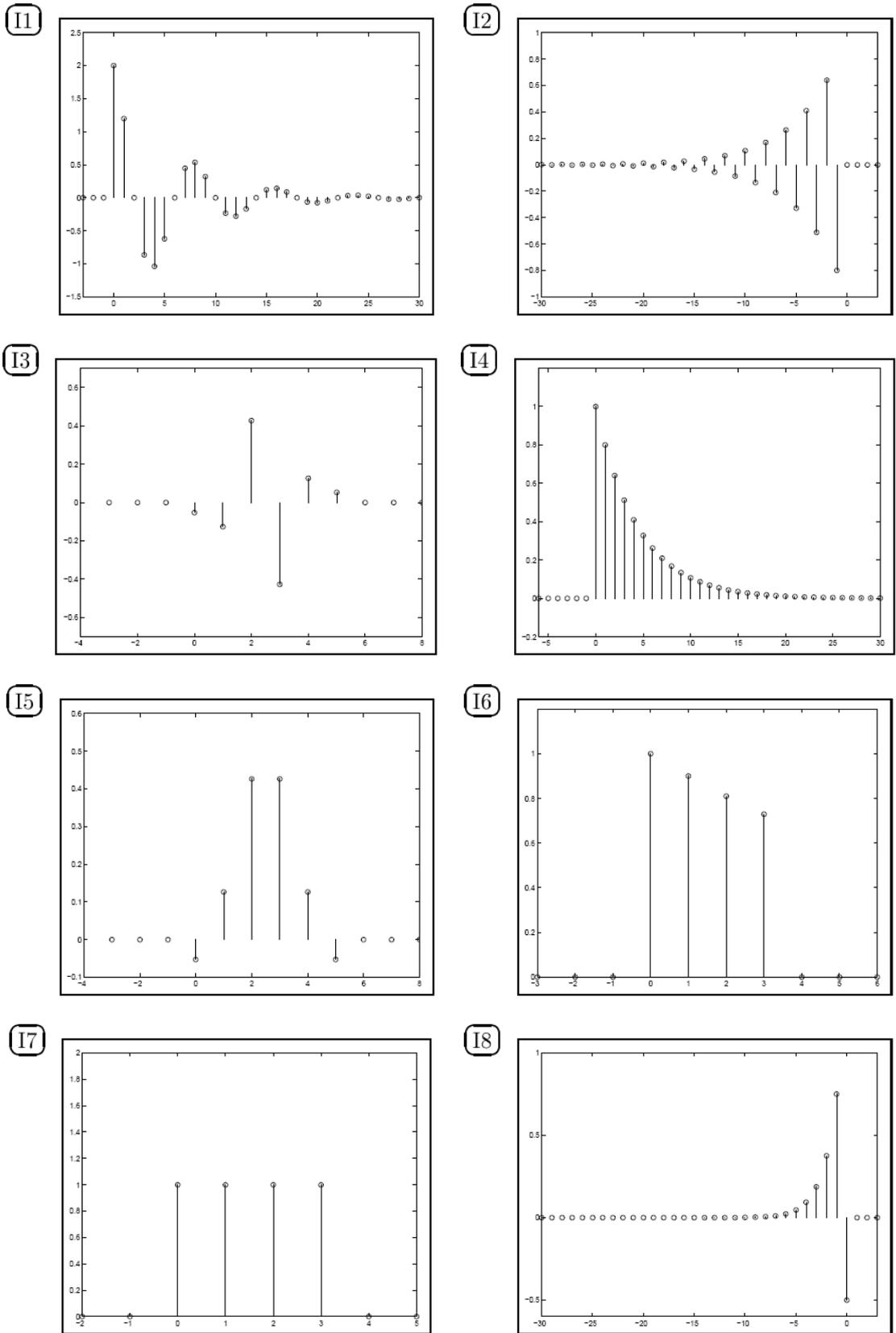


Figure 3: Réponses impulsionnelles  $h[n]$ . L'abscisse correspond au temps discret  $n$ .

**Exercice 2.5**

On mesure la réponse impulsionnelle d'un système numérique linéaire et invariant et on trouve :  $h(n) = 2 * (-0.8)^n \varepsilon(n)$ . Donner la récurrence numérique du système, en déduire sa fonction de transfert opérationnelle  $H(z)$  et esquisser sa réponse en fréquence.

**Exercice 2.6**

Soit le système numérique linéaire de fonction de transfert  $T(z) = 1/A(z)$ . On demande de choisir  $A(z)$  de façon à ce que  $T(z)$  constitue un oscillateur numérique (à la limite de stabilité) de pulsation réduite  $\varphi_0 = \frac{\pi}{2}$ . On demande ensuite de calculer les 5 premiers échantillons de sa réponse impulsionnelle.

**Exercice 2.7**

Soit le système numérique linéaire défini par :

$$y(n] = x(n] + k_1 y(n - 1] + k_2 y(n - 2]$$

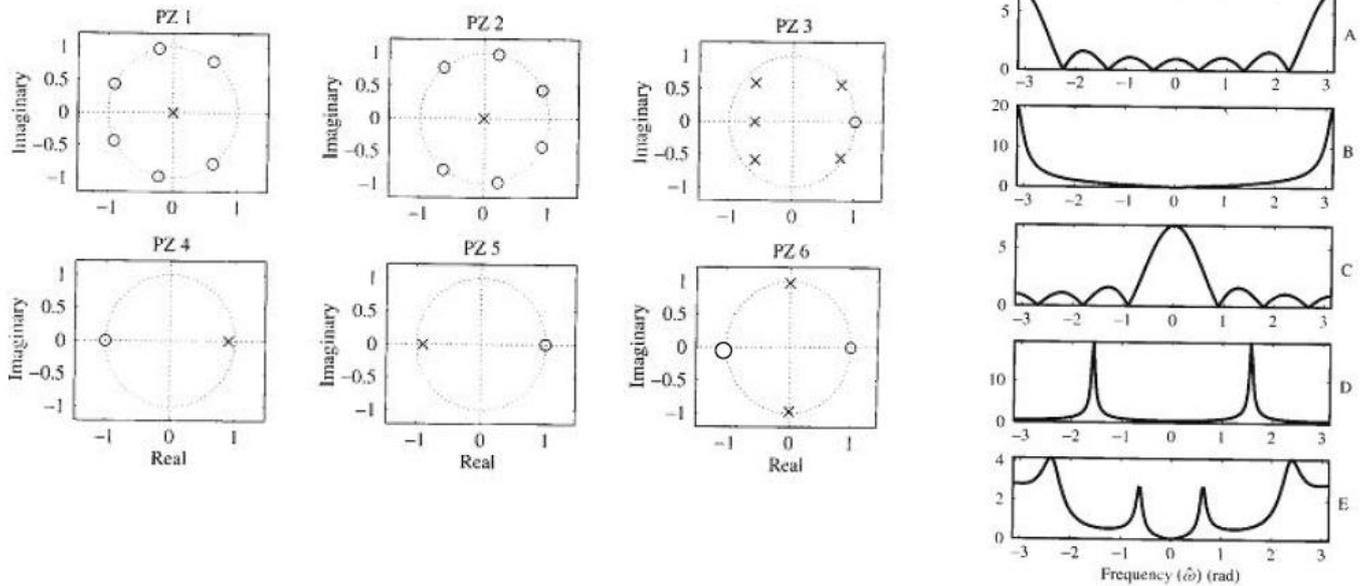
Où  $k_1 = 2 r \cos \theta$   $0 < \theta < \pi$   $k_2 = -r^2$   $r > 0$

On demande de :

- Donner sa fonction de transfert opérationnelle
- D'en déduire la position des pôles et des zéros
- De donner les conditions de stabilité du système
- De dessiner l'allure de sa réponse en fréquence pour  $r=0.9$  et  $\theta = 36^\circ$
- De dessiner une structure pour ce système

**Exercice 2.8**

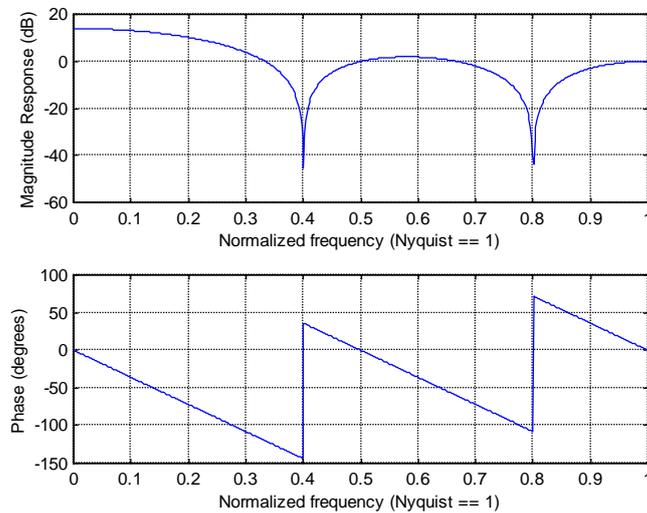
Etablir les correspondances entre les diagrammes pôles zéros et les réponses en fréquences suivants et **justifier** vos choix :



**Exercice 2.9**

On demande de calculer analytiquement et graphiquement la TFTD de la séquence  $\{1,1,1,1,1,0,0,\dots\}$ , et de vérifier le résultat avec Matlab.

Solution



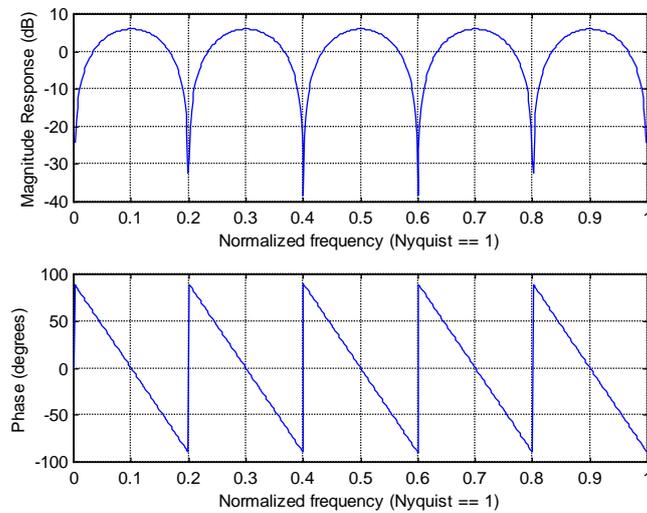
**Exercice 2.10**

Un *filtre en peigne* est défini par la relation :

$$y(n) = x(n) - x(n - n_0]$$

On demande de déterminer analytiquement sa réponse en fréquence, d'esquisser le graphique de son module à partir de son diagramme pôles-zéros, et de justifier l'appellation de filtre en peigne. Vérifier la réponse sous Matlab pour  $n_0 = 10$ .

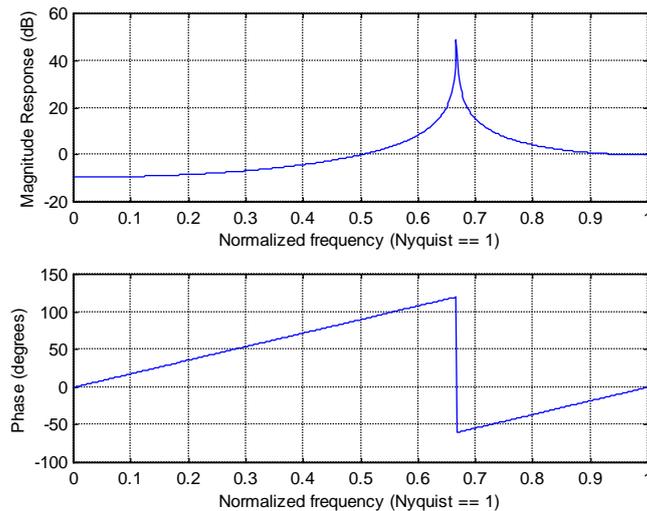
Solution



**Exercice 2.11**

Il a été vu à l'exercice 1.2 que la récurrence  $y(n)=x(n)+2\cos(\theta)y(n-1)-y(n-2)$  définit un oscillateur numérique à la pulsation réduite  $\varphi$ . On demande de calculer l'expression analytique de la réponse en fréquence de ce système pour  $\theta=2\pi/3$ , d'expliquer sur le diagramme pôles-zéros pourquoi il y a résonance en  $F=1/3$ , et de vérifier la réponse sous Matlab.

Solution



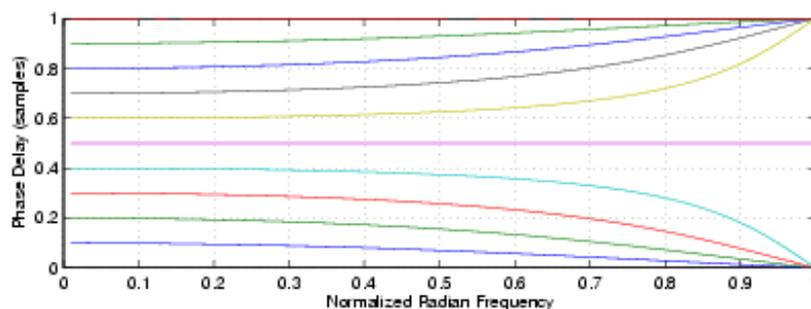
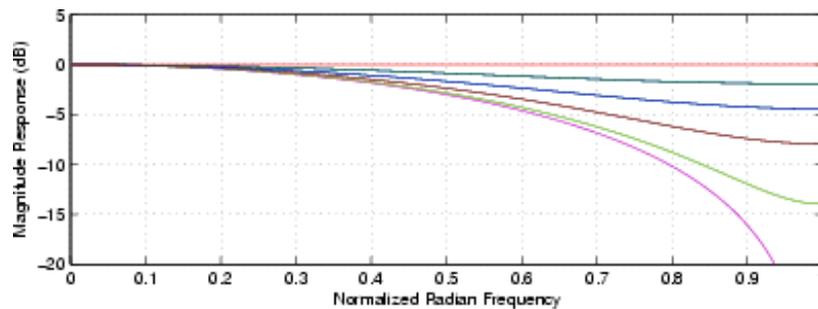
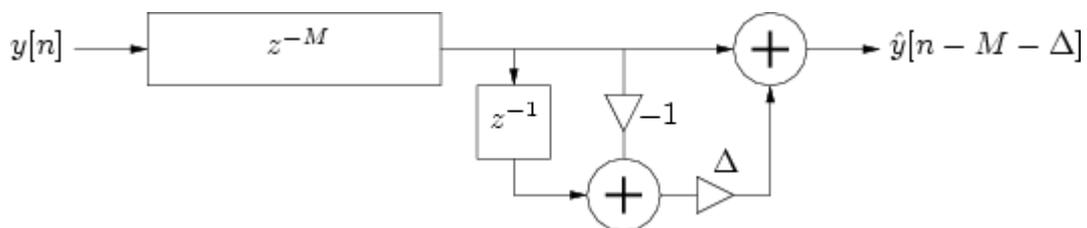
**Exercice 2.12**

Un élément délai de type  $z^{-M}$  permet de retarder un signal de  $M$  échantillons. Il est en pratique également possible de simuler le retard d'une fraction  $\Delta$  d'échantillon. En effet, on a :

$$\begin{aligned}\hat{y}[n + \Delta] &= (1 - \Delta) \cdot y[n] + \Delta \cdot y[n + 1] \\ &= y[n] + \Delta \cdot (y[n + 1] - y[n]).\end{aligned}$$

Dessiner la structure du filtre correspondant, et calculer sous Matlab sa réponse en phase et en fréquence pour des valeurs de  $\Delta$  comprises entre 0 et 1.

Solution



On constate que la simulation du retard est mauvaise en HF.

# CHAPITRE 3

## ECHANTILLONNAGE

Claude E. Shannon (1916-2001), mathématicien et ingénieur électricien américain. Après deux thèses aux MIT, Shannon entra aux Bell Labs, en 1941, où il établit les fondements de la théorie de l'information. Il décrit le premier la possibilité d'envoyer du son, du texte, et des images, par le biais d'un flux binaire (il est d'ailleurs l'inventeur du « bit », contraction de « binary digit »), dans un article magistral : "*A mathematical theory of communication*", Bell Labs Technical Journal, 1948. Il s'intéressa plus tard à l'intelligence artificielle, et fut un des premiers à programmer un jeu d'échecs.



La grande majorité des signaux numériques représentent en réalité des signaux analogiques sous-jacents, auxquels on veut pouvoir appliquer toute la puissance des traitements numériques. La conversion analogique-numérique est appelée *échantillonnage*. Nous étudierons dans ce chapitre l'interprétation spectrale de l'échantillonnage, et en déduirons l'expression du *théorème de Shannon*.

Nous examinerons de plus près les problèmes liés à la conception du filtre de garde, et introduirons l'interpolation-décimation ainsi que le sur-échantillonnage. Nous verrons de même comment reconstruire un signal analogique à partir de son homologue numérique, par extrapolation et lissage.

Nous concluons ce chapitre par l'examen du *théorème de Shannon généralisé* et de son application pour le filtrage en sous-bande.

### 3.1 Principe

Les signaux numériques peuvent être créés directement par un système numérique (un processeur, sous toutes ses formes actuelles). On parle alors de *synthèse numérique* (comme dans le cas de la synthèse d'images ou de sons numériques). La plupart du temps, cependant, ils sont obtenus par échantillonnage de signaux analogiques.

L'*échantillonnage* d'un signal analogique représenté par une fonction  $f(t)$  consiste à construire, à partir de  $f(t)$ , un signal à temps discret  $f(n) = f(nT_e)$  obtenu en mesurant la valeur de  $f(t)$  toutes les  $T_e$  secondes (Fig. 3.1) :

$$f(n) = f(nT_e) \tag{3.1}$$

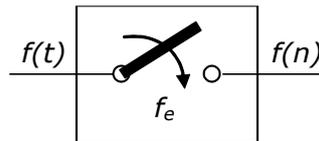


Fig. 3.1 Représentation schématique de l'échantillonnage

Si  $f(t)$  subit une discontinuité par saut à un instant d'échantillonnage, on convient de poser :

$$f(n) = 1/2. [f(nT_e^+) + f(nT_e^-)] \quad (3.2)$$

Le schéma de principe de l'échantillonnage est décrit à la Fig. 3.2. Il exprime le fait qu'on peut considérer que  $f^+(t)$  est obtenu par multiplication de  $f(t)$  par un train d'impulsions de Dirac de période  $T_e$  :

$$f^+(t) = f(t)\delta_{T_e}(t) = \sum_n f(nT_e)\delta(t - nT_e) \quad (3.3)$$

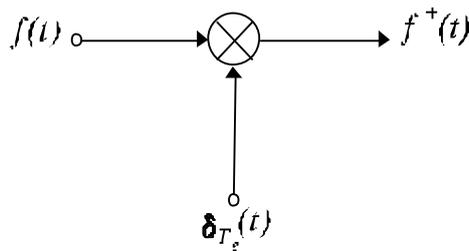


Fig. 3.2 Représentation mathématique de l'échantillonnage

On peut alors interpréter la transformée de Fourier  $F^+(\omega)$  de  $f^+(t)$  (c'est-à-dire la TFTD de  $\{f(n)\}$ ) comme celle d'un produit, et en calculer la transformée de Fourier :

$$F^+(f) = F(f) * \left[ \frac{1}{T_e} \delta_{f_e}(f) \right] = \frac{1}{T_e} \sum_{k=-\infty}^{\infty} F(f - kf_e) \quad (3.4)$$

Nous savions déjà que la TFTD d'un signal échantillonné est périodique en  $f$ , de période  $f_e$  (ou, en  $F$ , de période 1). Nous comprenons maintenant que, lorsque le signal numérique est obtenu par échantillonnage d'un signal analogique, cette périodicité résulte de la superposition de toutes les translatées (à des multiples entiers de  $f_e$ ) de la transformée de Fourier du signal original divisée par  $T_e$ .

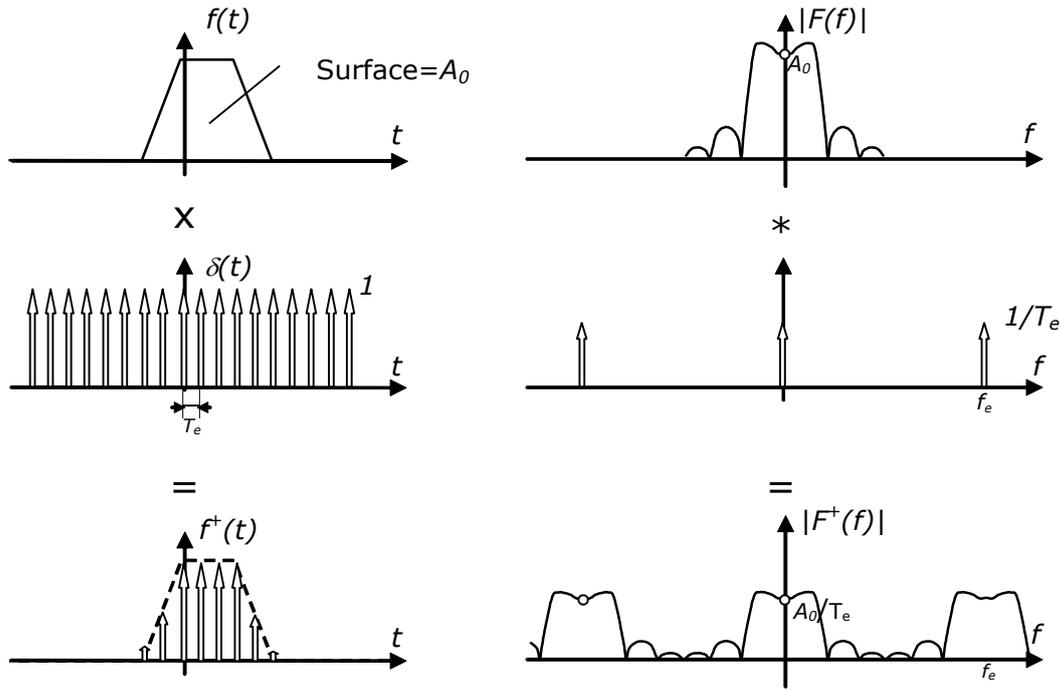


Fig. 3.3 Spectre d'un signal avant (en haut) et après (en bas) échantillonnage (cas sans recouvrement)

### 3.2 Recouvrement spectral (aliasing)

Si le spectre  $F(f)$  du signal analogique  $f(t)$  n'est pas nul au delà de  $f_e/2$ , la superposition (3.4) peut conduire à des empiétements des translattées (Fig. 3.4). Ce phénomène est appelé *recouvrement* (ou *repliement*) spectral (*aliasing*).

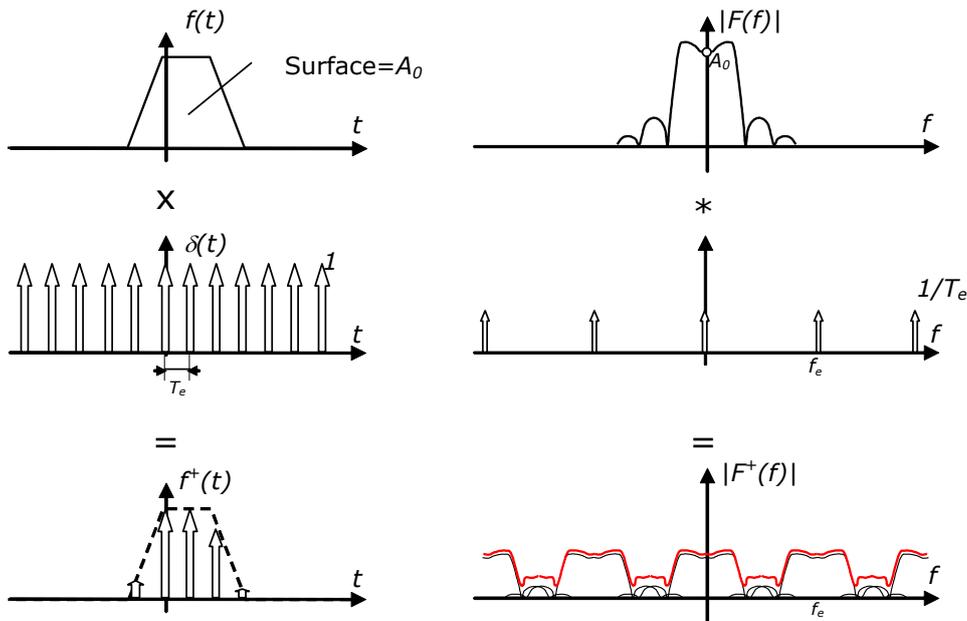


Fig. 3.4 Phénomène de recouvrement spectral<sup>1</sup>

Le recouvrement spectral a pour conséquence que le signal à temps discret  $f(n)$  obtenu par échantillonnage n'est plus une image correcte de  $f(t)$ , mais bien du signal  $\tilde{f}(t)$  dont le spectre est donné par  $F^+(f)$  entre  $-f_e$  et  $+f_e$ . Ainsi, par exemple, une sinusoïde de fréquence  $f$  juste légèrement inférieure à  $f_e$  sera perçue, après échantillonnage, comme une sinusoïde à basse fréquence  $f_e - f$ . C'est l'effet stroboscopique, bien connu (Fig. 3.5).

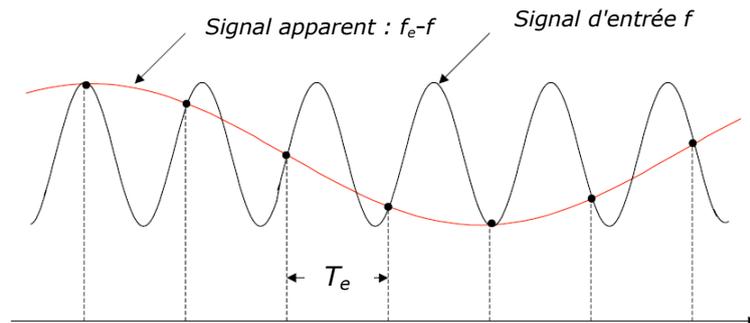
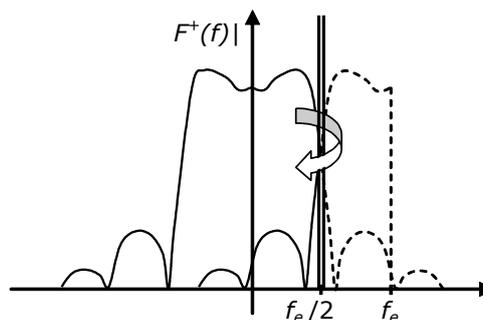


Fig. 3.5 Effet stroboscopique (extrait de *Mixed-Signal and DSP Design Techniques*, edited by Walt Kester, Newnes, 2003, [http://www.analog.com/library/analogDialogue/archives/39-06/mixed\\_signal.html](http://www.analog.com/library/analogDialogue/archives/39-06/mixed_signal.html))

Le terme de *repliement spectral* est d'ailleurs tout à fait justifié (plus encore que celui de *recouvrement*). En effet, tout se passe comme si la partie de  $F(f)$  inférieure à  $f_e/2$  se trouvait additionnée à la partie de ce même  $F(f)$  supérieure à  $f_e/2$ , repliée autour de  $f_e/2$  et conjuguée (Fig. 3.6).

Fig. 3.6 Repliement spectral<sup>2</sup>

<sup>1</sup> Ce graphique est ici à des fins didactiques, mais il peut porter à confusion : en réalité, c'est le spectre complexe  $F(\omega)$  qui est additionné à des translattées de lui-même, et pas le spectre d'amplitude.

<sup>2</sup> Ce graphique est en réalité assez réducteur : non seulement ce ne sont pas les spectres d'amplitude qui s'additionnent (mais bien les spectres complexes), mais de plus l'opération de repliement est en réalité associée à la conjugaison de  $F(f)$ .

**Exemple 3.1**

On échantillonne une sinusoïde  $f(t) = \sin(2\pi f_0 t)$  à la fréquence d'échantillonnage de 10000 Hz. Dessinons l'allure des échantillons (c'est-à-dire l'allure de la fonction  $f^*(t)$  correspondante) pour des valeurs de  $f_0$  égales à : 1000 Hz, 2500 Hz, 5000 Hz, 7500 Hz, 9000 Hz, et 31000 Hz.

Pour que les graphiques possèdent des axes temporels identiques, choisissons de montrer les 10 premières ms des signaux (Fig. 3.7).

```
subplot(6,1,1); stem(sin(2*pi*1000*[0:99]/10000));
subplot(6,1,2); stem(sin(2*pi*2500*[0:99]/10000));
subplot(6,1,3); stem(sin(2*pi*5000*[0:99]/10000));
subplot(6,1,4); stem(sin(2*pi*7500*[0:99]/10000));
subplot(6,1,5); stem(sin(2*pi*9000*[0:99]/10000));
subplot(6,1,6); stem(sin(2*pi*31000*[0:99]/10000));
```

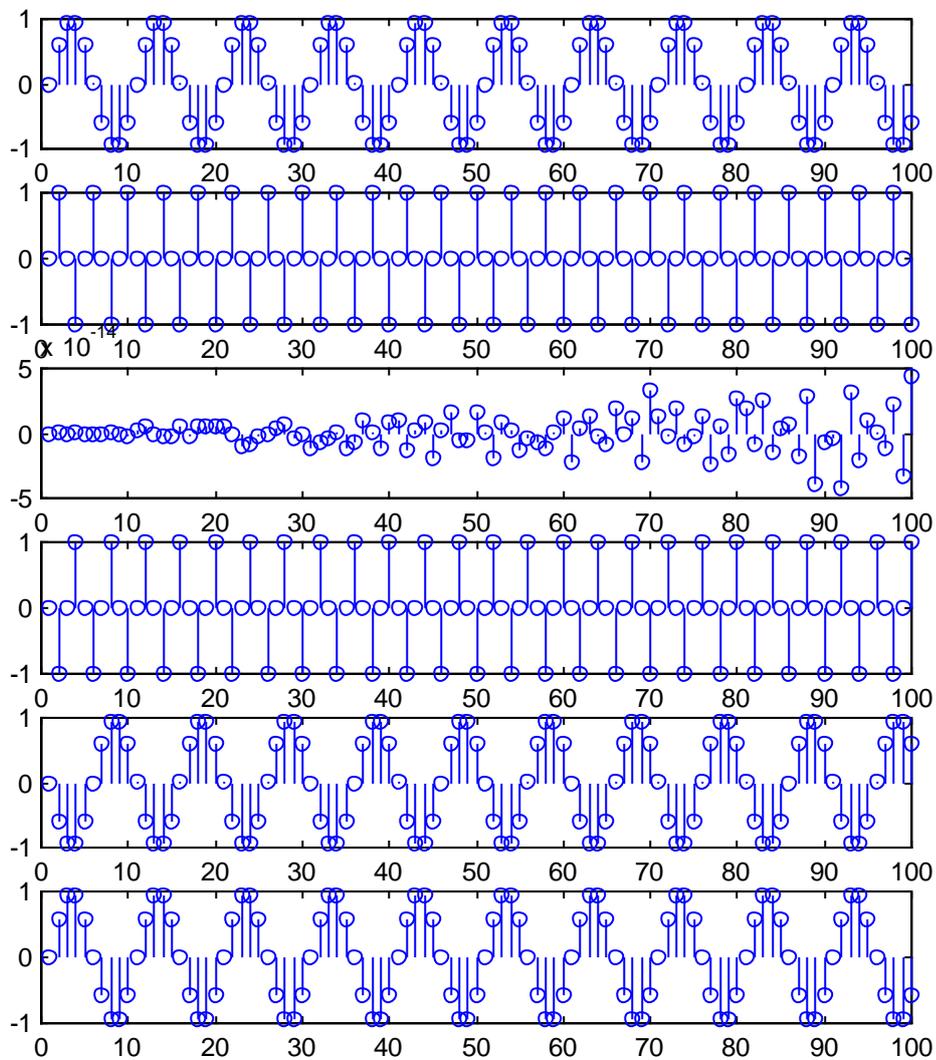


Fig. 3.7 Echantillonnage de sinusoïdes

On constate :

- Que seuls les échantillonnages des sinusoïdes à 1000 et 2500 Hz donnent une image réaliste des signaux sous-jacents (on peut retrouver la fréquence des sinusïdes correspondantes en mesurant leur période sur le graphique).
- Que l'échantillonnage de la sinusoïde à 5000 Hz fait apparaître un signal d'amplitude très petite. En théorie, les échantillons devraient être tous nuls. Les valeurs non nulles apparaissant sur le graphique sont le résultat d'erreurs d'arrondis de Matlab.
- Que les échantillonnages des sinusoïdes à 7500 Hz et 9000 Hz donnent des résultats identiques (au signe près) à ceux des sinusoïdes à 2500 Hz et 1000 Hz. Il est par conséquent impossible, après échantillonnage, de retrouver la fréquence exacte des sinusoïdes sous-jacentes. C'est le résultat du repliement spectral des sinusoïdes de départ autour de la fréquence de Nyquist (5000 Hz).
- Que l'échantillonnage de la sinusoïde à 31000 Hz donne exactement le même résultat que celui d'une sinusoïde à 1000 Hz. C'est le résultat de la superposition de toutes les translatées (à des multiples entiers de  $f_e$ ) de la transformée de Fourier du signal original.

Un moyen simple d'éviter le repliement spectral est d'échantillonner à une fréquence supérieure à deux fois la plus haute composante fréquentielle du signal. Certains signaux cependant (typiquement, les fonctions qui présentent des discontinuités), ont un spectre théoriquement infini. Dans ce cas, on peut toujours choisir la fréquence d'échantillonnage *de façon à imposer que le recouvrement spectral soit inférieur à un seuil.*

### Exemple 3.2

On cherche à calculer la transformée de Fourier de  $rect_a(t)/a$ . On demande :

- de choisir la fréquence d'échantillonnage de façon que la valeur maximale de la transformée de Fourier de  $rect_a(t)/a$  soit affectée de moins de 1% d'erreur par repliement spectral (on suppose qu'il n'y a pas de filtre de garde);
- de vérifier ce résultat sous Matlab.

Il est clair qu'il y aura repliement spectral, quelle que soit la fréquence d'échantillonnage choisie. En effet, la transformée de Fourier de  $rect_a(t)/a$  est donnée par :

$$rect_a(t)/a \stackrel{\mathbb{F}}{\Leftrightarrow} \frac{\sin(a\omega/2)}{a\omega/2}$$

Son enveloppe est donc donnée par  $2/a\omega$ . Le maximum du spectre se trouve en  $f=0$  et vaut 1. Si l'on suppose que l'effet du repliement est surtout du au premier spectre image à droite, qu'on approxime ce spectre par son enveloppe (Fig. 3.8), et qu'on suppose que l'amplitude du spectre résultant est la somme des amplitudes du spectre de base et de ce premier spectre image<sup>3</sup>, il vient :

$$|F(f_e)| < 0.01 \max(|F(f)|) = 0.01$$

$$2/(a2\pi f_e) < 0.01$$

$$f_e > 100/\pi a$$

---

<sup>3</sup> En réalité la somme étant complexe, l'effet du recouvrement sera en général moindre que celui calculé ici.

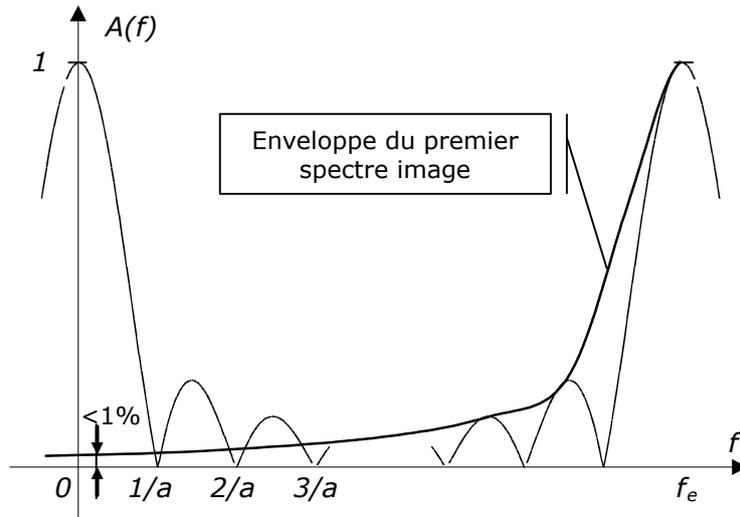
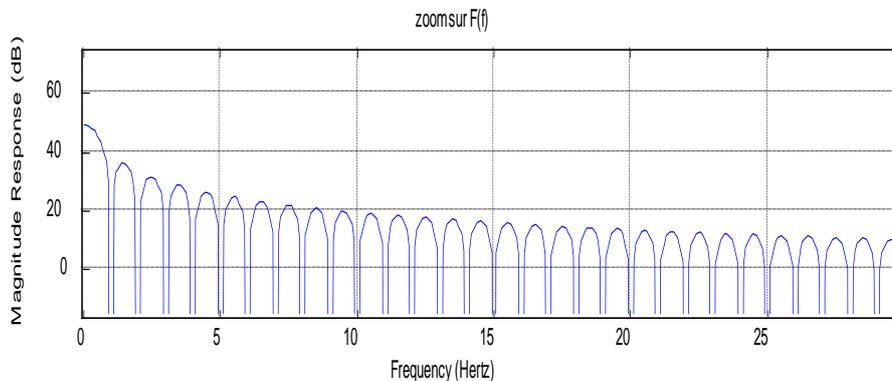


Fig. 3.8 Effet approximatif du repliement

Si on prend par exemple  $a=1$  (ce qui correspond à un carré d'une seconde), on trouve à peu près  $f_e > 30$ . Comme chaque lobe a une largeur de 1 Hz, on constate qu'il faut 30 lobes pour que l'erreur soit de moins de 1%.

Pour voir cet effet sous Matlab, on affiche la transformée  $F^+(f)$  du signal obtenu par échantillonnage de  $rect_1(t)$  à une fréquence largement supérieure à 30 Hz (ex : 300 Hz). Le signal est donc constitué de 300 échantillons égaux à 1. On constate que le spectre obtenu est pratiquement exempt de repliement spectral entre 0 et 30 Hz, et correspond donc pratiquement à  $F(f)$ .

```
freqz([ones(1,300)],1,0:1/10:30,300) % F(f) entre 0 et 30 Hz => 30 lobes
```

Fig. 3.9 Transformée de Fourier de  $rect_1(t)$ 

On constate que l'enveloppe de  $F(f)$  à 30 Hz est bien 40 dB sous  $F(0)$  (c'est-à-dire 1% de  $F(0)$ ).

Notons qu'en pratique, on ne connaît pas la transformée de Fourier que l'on cherche à calculer. Le calcul ci-dessus pour le choix de  $f_e$  est alors remplacé par une estimation itérative : partant d'une première valeur de  $f_e$ , on augmente progressivement  $f_e$  jusqu'à ce que la transformée de Fourier mesurée ne change plus beaucoup. On a alors atteint une valeur suffisante pour éviter le recouvrement spectral. Une alternative consiste à utiliser dès le départ une fréquence d'échantillonnage suffisamment élevée pour que la condition de non-recouvrement soit vérifiée à coup sûr.

### 3.3 Filtre de garde (anti-aliasing)

On cherche donc en général à éviter ce phénomène, en faisant en sorte que le spectre de la fonction  $f(t)$  soit à support borné  $[-f_M, +f_M]$  avec  $f_M < f_e/2$  (comme sur la Fig. 3.3).

On sait cependant qu'une fonction dont le spectre est à support borné est illimitée dans le temps. Un signal physique n'est jamais illimité dans le temps et par conséquent son spectre n'est pas à support borné. Son échantillonnage provoque donc en principe toujours du recouvrement spectral. Par ailleurs, tout signal analogique est affecté par des bruits additifs, qui portent des composantes à haute fréquence. Ces bruits peuvent venir dégrader le spectre utile du signal par repliement spectral.

La condition de non recouvrement n'est donc en général pas vérifiée par les signaux usuels, ce qui implique la nécessité de faire précéder l'échantillonnage d'un filtre dont les spécifications seront définies pour vérifier la condition de non recouvrement; ce filtre sera appelé *filtre de garde* ou *filtre anti-repliement*. (Fig. 3.10).

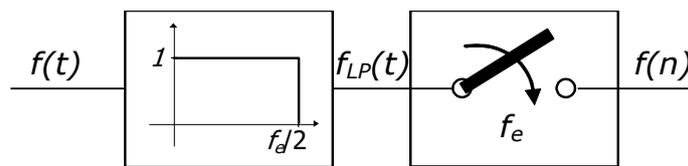


Fig. 3.10 Filtre de garde idéalisé

On commet toujours une certaine « erreur » en fixant la limite du spectre à  $f_e/2$  : les composantes du signal à plus haute fréquence ne sont pas prises en compte.<sup>4</sup>

### 3.4 Théorème de Shannon

Le *théorème de Shannon*, également appelé *théorème de l'échantillonnage*, est un des fondements du traitement numérique des signaux :

"Toute fonction  $f(t)$  dont le spectre est à support borné ( $F(f) = 0$  pour  $|f| > f_M$ ) est complètement définie par ses échantillons  $f(nT_e)$  si  $f_e \geq 2f_M$ ."

Ce théorème découle immédiatement de l'analyse précédente concernant le phénomène de recouvrement : l'échantillonnage à une fréquence  $f_e$  d'une fonction  $f(t)$  qui respecte le théorème de Shannon ne produit pas de recouvrement spectral. Par conséquent, le signal original  $f(t)$  peut être reconstitué par le filtrage du signal impulsionnel  $f^+(t)$  par un passe-bas idéal de fréquence de coupure  $f_M$  (Fig. 3.11, droite).

<sup>4</sup> Cette erreur n'est pas nouvelle. Tout système de mesure analogique souffre du même inconvénient : les composantes des signaux mesurés supérieures à la bande passante du système de mesure ne sont pas prises en compte.

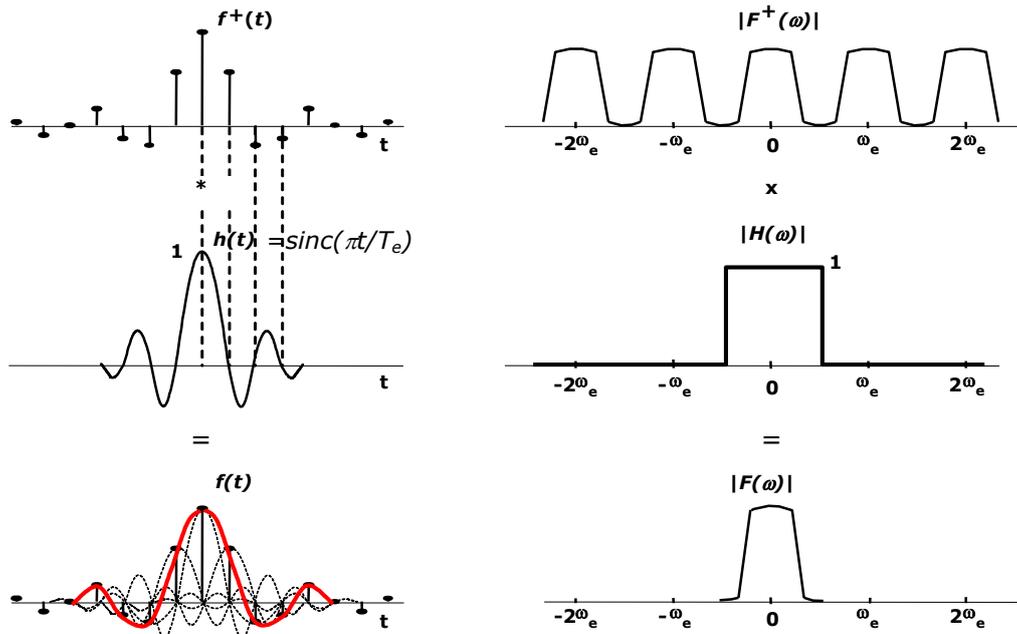


Fig. 3.11 Filtre de reconstitution (Gauche : convolution temporelle; Droite : multiplication fréquentielle)

Cette opération de filtrage correspond à la convolution du signal  $f^+(t)$  par la réponse impulsionnelle du filtre idéal, ce qui revient pratiquement à réaliser la somme des réponses impulsionnelles dues à chaque impulsion de  $f^+(t)$  :

$$f(t) = \sum_n f(nT_e) \cdot \frac{\sin[\omega_M \cdot (t - nT_e)]}{\omega_M \cdot (t - nT_e)} \tag{3.5}$$

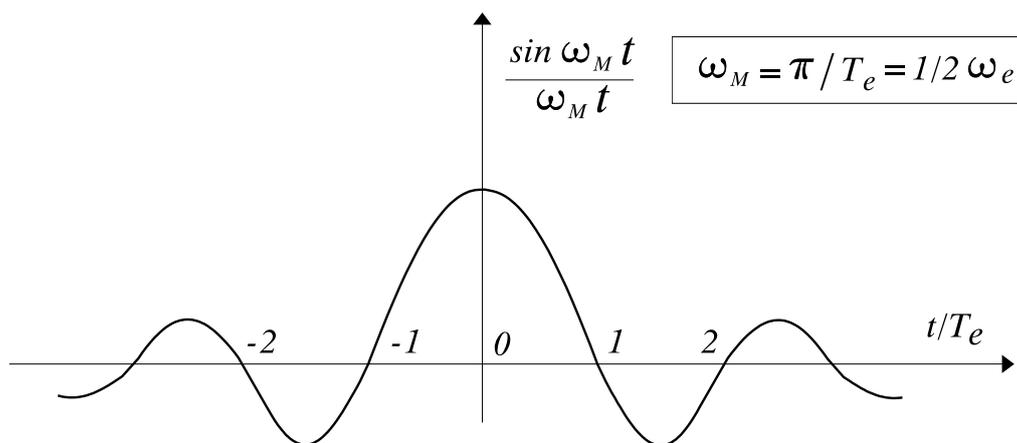


Fig. 3.12 La fonction d'interpolation idéale

La fonction  $\sin(\omega_M t)/(\omega_M t)$  est la *fonction d'interpolation idéale* (Fig. 3.12); elle s'annule aux points  $t = nT_e$ , ce qui confirme l'indépendance des échantillons  $f(n)$  dans la reconstruction du signal  $f(t)$  (Fig. 3.11, gauche).

Notons pour terminer que la fréquence  $f_e/2$ , qui donne la limite supérieure du spectre du signal à échantillonner, est souvent appelée *fréquence de Nyquist*.

### Exemple 3.3

Parmi les sinusoïdes examinées à l'exemple précédent, seules celles à 1000 et 2500 Hz respectaient le théorème de Shannon, ce qui explique qu'on puisse retrouver à l'œil les sinusoïdes sous-jacentes à partir des échantillons.

## 3.5 Reconstruction du signal à temps continu – filtre de lissage (anti-imaging)

A partir d'un signal à temps discret, on peut vouloir reconstruire un signal à temps continu. En pratique, il est évidemment impossible de filtrer le signal impulsionnel  $f^+(t)$ , qui n'est qu'une abstraction mathématique utile à la compréhension des phénomènes.

On procède alors en deux étapes :

- On construit un vrai signal analogique  $f^*(t)$  à partir des échantillons du signal à temps discret. Cette opération porte le nom d'*extrapolation*. Le signal  $f^*(t)$  est en quelque sorte une première ébauche de  $f(t)$ .
- On fait suivre l'extrapolateur d'un filtre est appelé *filtre de lissage*, qui affine  $f^*(t)$  et le rend plus proche de  $f(t)$ .

Le principe le plus simple de la reconstruction du signal à temps continu consiste en l'utilisation d'un *extrapolateur d'ordre zéro* (en anglais : *sample and hold*) dont le schéma de base est décrit à la Fig. 3.13. On peut, pour mieux comprendre son effet, le considérer comme un filtre de réponse impulsionnelle unitaire de 0 à  $T_e$  et nulle partout ailleurs : l'extrapolation de  $f(n)$ , tout comme le filtrage de  $f^+(t)$ , conduit à un signal  $f^*(t)$  évoluant par paliers.

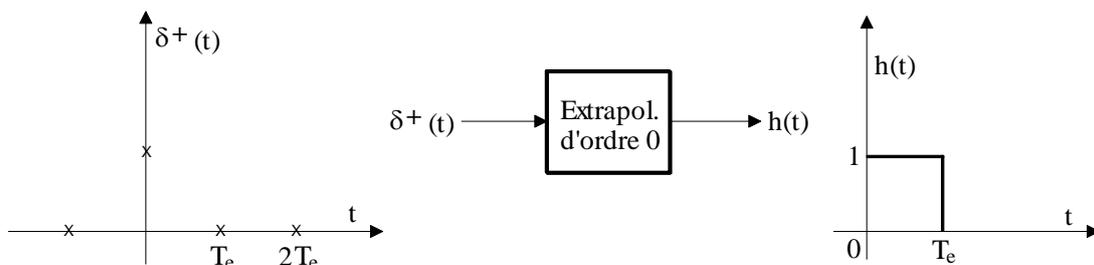


Fig. 3.13 Extrapolateur d'ordre 0

Connaissant la réponse impulsionnelle du filtre équivalent à l'extrapolateur d'ordre 0, on peut calculer sa fonction de transfert et sa réponse en fréquence :

$$H(p) = \frac{1}{p} \cdot (1 - e^{-pT_e}) = \frac{2}{p} \cdot e^{-pT_e/2} \cdot \sinh(pT_e/2) \quad (3.6)$$

ce qui conduit à :

$$H(j\omega) = T_e \cdot e^{-j\omega T_e/2} \cdot \frac{\sin(\omega T_e/2)}{\omega T_e/2} \quad (3.7)$$

Le principe de la reconstitution d'un signal à temps continu à partir d'un signal impulsionnel est alors décrit à la Fig. 3.14. L'examen de l'amplitude spectrale du signal à temps continu montre que l'amplitude spectrale dans la zone utile  $(0, f_e/2)$  est déformée par la fonction  $\sin(\omega T_e/2)/(\omega T_e/2)$  (qui n'est qu'une pâle approximation de la fonction de transfert du filtre de lissage idéal) et de plus que des résidus non négligeables des spectres translétés subsistent. Un filtrage passe-bas supplémentaire est donc nécessaire pour éliminer ces résidus dans le spectre: c'est le *filtre de lissage* dont les spécifications sont fort proches de celles du filtre de garde.

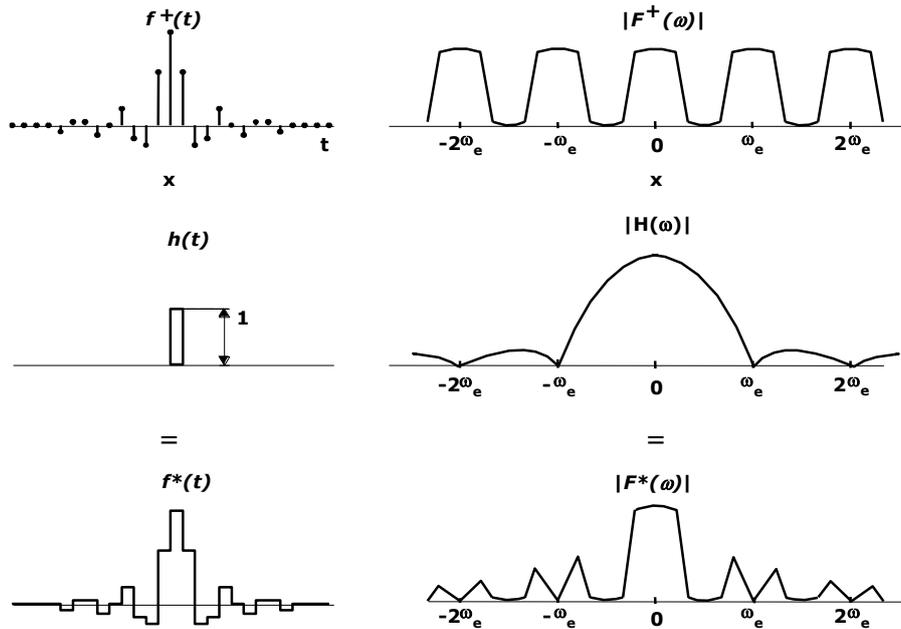


Fig. 3.14 L'extrapolateur d'ordre 0 vu comme un filtre, suivi d'un (vrai) filtre de lissage analogique.

En conséquence, le traitement numérique d'un signal (comme un filtrage numérique) implique l'utilisation de deux filtres analogiques (Fig. 3.15).

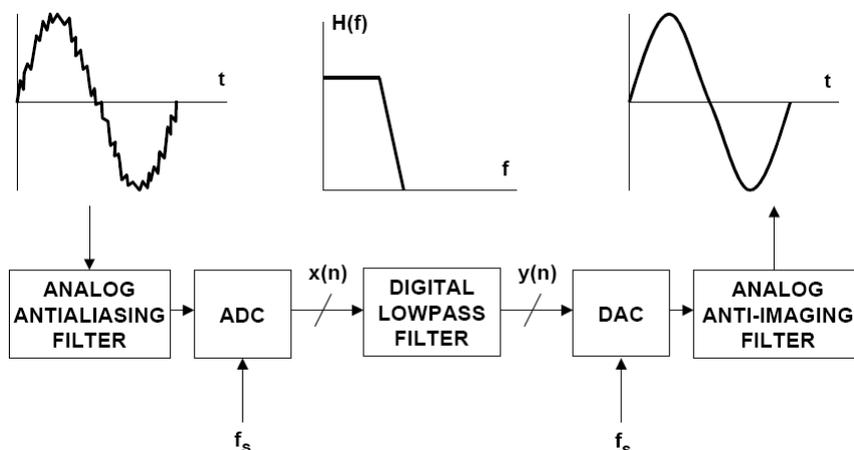


Fig. 3.15 Chaîne de filtrage numérique à fréquence d'échantillonnage  $f_s$ , avec filtre de Garde et filtre de Lissage (anti-imaging). (extrait de *Mixed-Signal and DSP Design Techniques*, edited by Walt Kester, Newnes, 2003, [http://www.analog.com/library/analogDialogue/archives/39-06/mixed\\_signal.html](http://www.analog.com/library/analogDialogue/archives/39-06/mixed_signal.html))

### 3.6 Changement de fréquence d'échantillonnage - Décimation – Interpolation

Il arrive que l'on cherche à augmenter ou à diminuer la fréquence d'échantillonnage d'un signal *déjà* échantillonné. Nous nous intéresserons ici au cas le plus courant : celui d'une division ou de la multiplication de la fréquence d'échantillonnage par un nombre entier.

#### 3.6.1 Décimation – Sous-échantillonnage (downsampling)

Considérons un signal  $x_1(n)$  obtenu par échantillonnage d'un signal analogique  $x(t)$  à une fréquence d'échantillonnage valant  $f_e$ . En vertu de ce qui a été dit plus haut, le spectre utile du signal est limité à l'intervalle  $[0, f_e/2]$  ; un filtre de garde veille d'ailleurs normalement à ce que cet intervalle ne soit pas perturbé par du repliement spectral.

*Sous-échantillonner* ce signal à une fréquence  $f_e'$ ,  $k$  fois inférieure à  $f_e$ , correspond en principe à ne retenir qu'un échantillon de  $x_1(n)$  sur  $k$  dans  $x_2(n)$ , à *décimer*  $x_1(n)$  par  $k$  (Fig. 3.16) :

$$x_2(n) = x_1(kn) \quad (3.8)$$

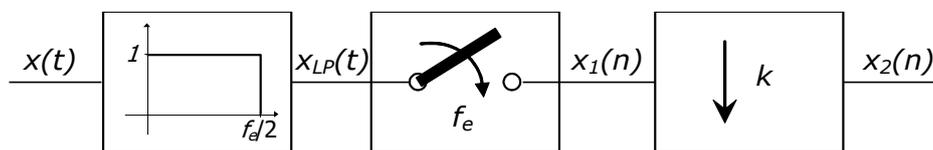


Fig. 3.16 Sous-échantillonnage d'un signal numérique  $x_1(n)$

Ceci pose cependant un problème important : le résultat de l'échantillonnage de  $x(t)$  à  $f_e$  suivi d'un second échantillonnage à  $f_e/k$  est évidemment équivalent à un échantillonnage direct à  $f_e/k$  : le spectre du signal décimé est celui du signal analogique de départ, rendu périodique de période  $f_e/k$ , et dont l'amplitude spectrale est divisée par  $kT_e$  (ce qui constitue donc une division par  $k$  par rapport au spectre du signal échantillonné à  $f_e$ ). Or, comme le filtre de garde a été prévu pour un échantillonnage à  $f_e$ , le sous-échantillonnage par  $k$  introduit, au niveau de la TFTD de  $x_2(n)$ , un repliement spectral des composantes de  $x_1(n)$  situées entre  $f_e/2k$  et  $f_e/2$ . Il est donc nécessaire de faire précéder le sous-échantillonnage d'un filtre *numérique* (au contraire du filtre de garde, analogique) passe-bas jouant le rôle d'un « adaptateur de filtre de garde », de fréquence de coupure égale à  $f_e/2k$  (Fig. 3.17). Le résultat final, appelé *décimation* est donné à la Fig. 3.18.

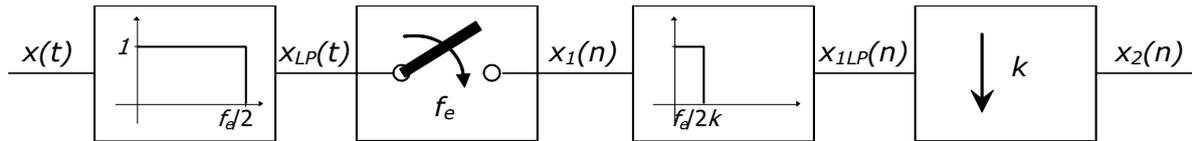


Fig. 3.17 Décimation par filtrage et sous-échantillonnage

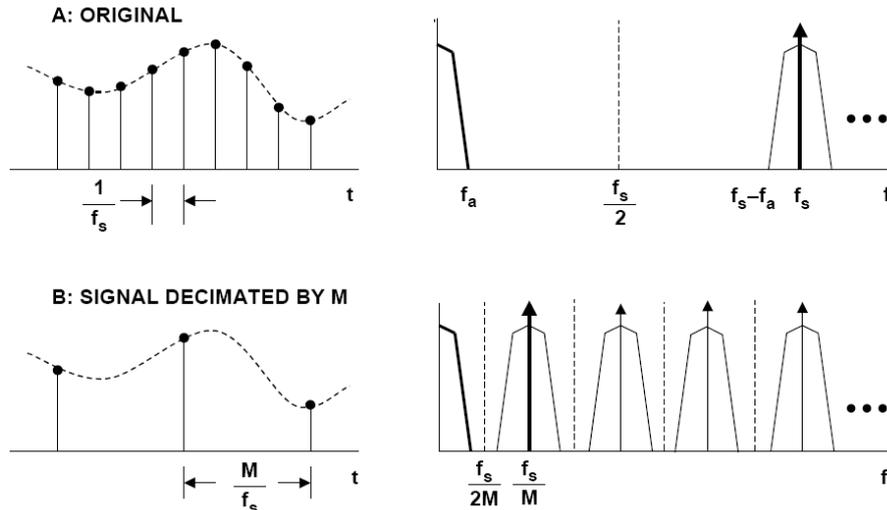


Fig. 3.18 Effet spectral de la décimation d'une fréquence d'échantillonnage  $f_s$  à  $f_s/M$  (extrait de *Mixed-Signal and DSP Design Techniques*, edited by Walt Kester, Newnes, 2003, [http://www.analog.com/library/analogDialogue/archives/39-06/mixed\\_signal.html](http://www.analog.com/library/analogDialogue/archives/39-06/mixed_signal.html))

### 3.6.2 Interpolation – Sur-échantillonnage (upsampling)

Pour *interpoler* le signal  $x_1(n)$  à une fréquence  $f_e'$ ,  $k$  fois supérieure à  $f_e$ , il faut calculer  $k-1$  échantillons intermédiaires entre deux échantillons connus de  $x_1(n)$ . Ce calcul est possible, puisqu'en vertu du théorème de Shannon il est même possible de reconstituer complètement  $x_{LP}(t)$  en utilisant l'interpolateur idéal.

Le calcul des  $k-1$  échantillons intermédiaires de  $x_{LP}(t)$  se fait en pratique de la manière suivante : on commence par sur-échantillonner le signal (au sens du mot anglais « up-sampling »), c.-à-d. par augmenter sa fréquence d'échantillonnage de façon brutale en insérant  $k-1$  échantillons nuls entre chaque paire d'échantillons, aux endroits requis :

$$\begin{aligned} x_2(n) &= x_1(n/k) \quad (n \text{ multiple de } k) \\ x_2(n) &= 0 \quad (n \text{ non multiple de } k) \end{aligned} \quad (3.9)$$

Puisqu'on n'a fait qu'ajouter des zéros, cette opération ne change en rien la TFTD du signal sur-échantillonné  $x_2(n)$ , qui reste périodique de période  $f_e$ . Par contre, le spectre utile de ce signal s'étend maintenant de 0 à  $kf_e/2$ . Ce spectre fait donc apparaître des composantes de  $x_1(n)$  qui ne sont pas présentes dans  $x_{LP}(t)$  (Fig. 3.19, où l'on a expressément représenté les TFTD en fréquence vraie, non normalisée).

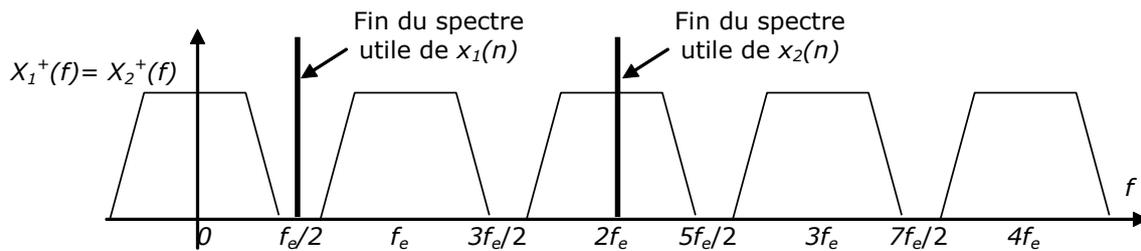


Fig. 3.19 Effet spectral du sur-échantillonnage par insertion de zéros (exemple :  $k=4$ )

Ces composantes perturbatrices peuvent être éliminées en faisant suivre le sur-échantillonnage par un filtre passe-bas *numérique* de fréquence de coupure égale à  $f'_e/2k$  (Fig. 3.20). Le résultat est appelé *interpolation* du signal. Son effet temporel est illustré à la Fig. 3.21; son résultat fréquentiel apparaît à la Fig. 3.22.

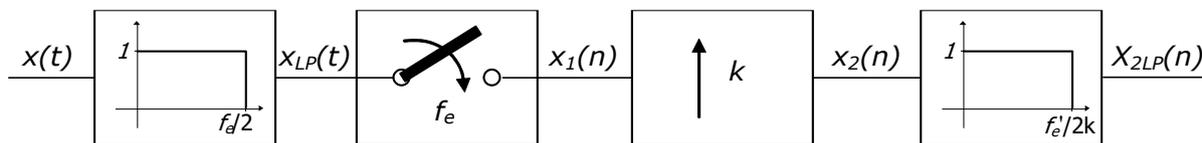


Fig. 3.20 Interpolation par sur-échantillonnage et filtrage

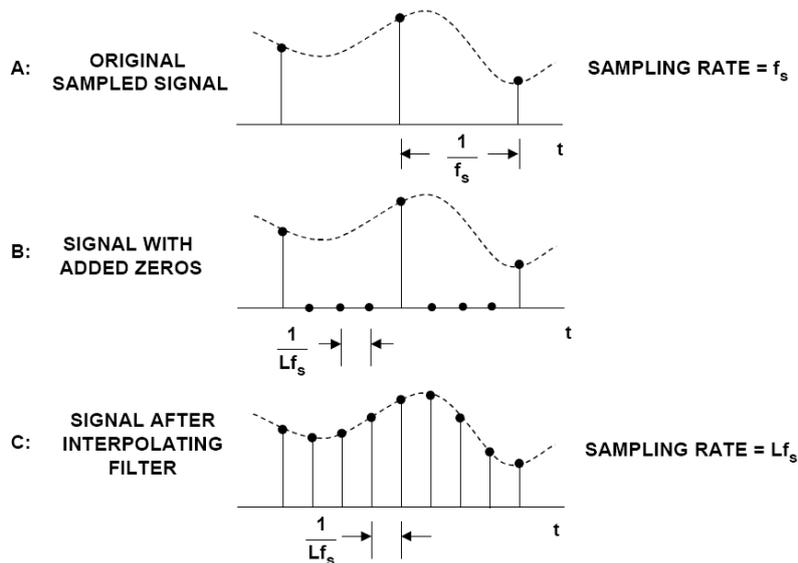


Fig. 3.21 Principe de l'interpolation par sur-échantillonnage et filtrage (d'une fréquence d'échantillonnage  $f_s$  à  $Lf_s$ ) par interpolation et filtrage (extrait de *Mixed-Signal and DSP Design Techniques*, edited by Walt Kester, Newnes, 2003, [http://www.analog.com/library/analogDialogue/archives/39-06/mixed\\_signal.html](http://www.analog.com/library/analogDialogue/archives/39-06/mixed_signal.html))

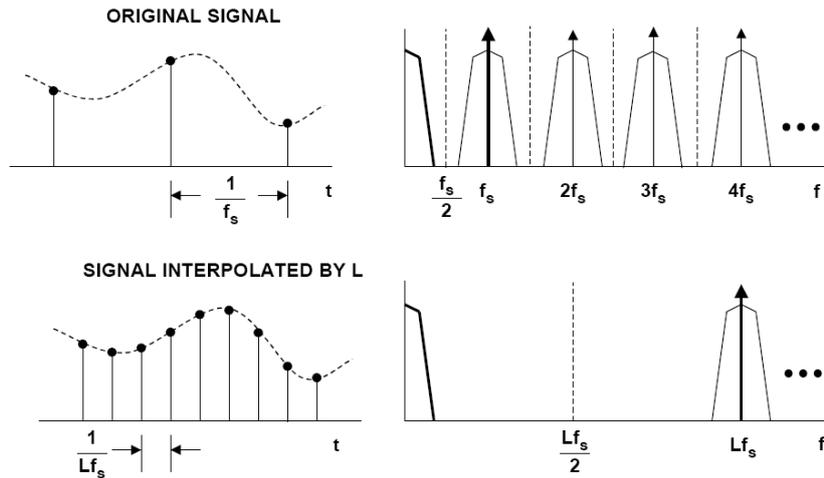


Fig. 3.22 Effet spectral de l'interpolation (extrait de *Mixed-Signal and DSP Design Techniques*, edited by Walt Kester, Newnes, 2003, [http://www.analog.com/library/analogDialogue/archives/39-06/mixed\\_signal.html](http://www.analog.com/library/analogDialogue/archives/39-06/mixed_signal.html))

On notera que les filtres numériques utilisés pour décimer et interpoler par un facteur  $k$  sont en réalité identiques. En effet, leur fréquence de coupure *normalisée* (rapport de la fréquence de coupure à la fréquence d'échantillonnage des signaux d'entrées et de sortie) vaut dans les deux cas  $1/2k$ .

### 3.7 Filtre de garde réel – Sur-échantillonnage (oversampling)

Le filtre de garde idéal est impossible à réaliser en pratique. Pour obtenir une transition aussi radicale entre bande passante et bande atténuée, il faudrait un filtre d'ordre infini. Le filtre de garde utilisé en pratique n'est qu'une approximation du filtre idéal (Fig. 3.23), dans laquelle on acceptera de commencer à atténuer le signal à partir d'une fréquence  $f_M$  inférieure à  $f_e/2$ , et où la bande atténuée commencera à une fréquence  $f_S$  supérieure à  $f_e/2$ . En général, cette approximation est bonne pour les fréquences éloignées de la fréquence de coupure du filtre. Par contre, le filtre réel possède une bande de transition de largeur finie, non nulle. Il s'ensuit que le spectre du signal à temps discret obtenu par échantillonnage du signal sortant du filtre de garde différera un peu du spectre du signal analogique de départ autour de  $f_M$ , à cause du repliement des composantes résiduelles supérieures à  $f_e/2$ .

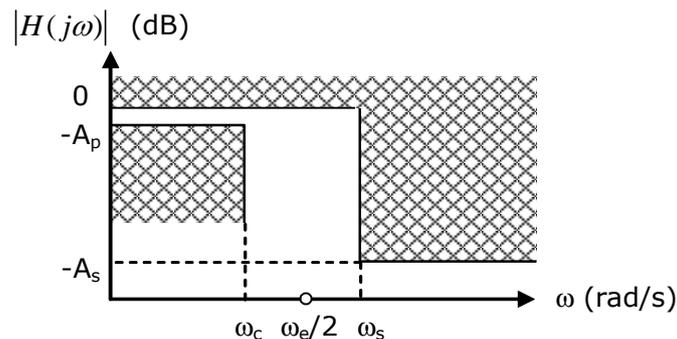


Fig. 3.23 Filtre de garde réel

Cet effet peut être compensé de deux façons :

- Si la fréquence d'échantillonnage est imposée, on peut faire en sorte que la bande passante du filtre de garde soit franchement plus étroite que la limite théorique de  $f_e/2$  (ce qui permet d'avoir une bande de transition plus large, et donc, pour un filtre de degré donné, un ripple plus faible en bande passante, et une atténuation plus importante en bande atténuée). Les recouvrements spectraux sont alors minimisés, au prix d'une perte des composantes à plus hautes fréquences du signal.
- Si on a la liberté de choisir la fréquence d'échantillonnage, on peut la choisir fortement supérieure à  $2f_M$ . En effet, pour une fréquence utile maximale  $f_M$  donnée, le filtre de garde est d'autant moins critique que la fréquence d'échantillonnage de départ est choisie supérieure à 2 fois cette fréquence utile : on peut élargir considérablement la zone de transition tout en gardant la même bande utile de fréquence. On paie alors par un nombre d'échantillons plus importants par seconde (et donc plus de calculs pour les traitements numériques qui suivent) le gain obtenu sur les spécifications du filtre de garde.

Dans certains cas, on choisit même d'échantillonner à plusieurs fois la fréquence désirée, ce qui permet de simplifier nettement le filtre de garde. Une fois le signal échantillonné, on procède alors à un sous-échantillonnage par filtrage et décimation. Une partie de la complexité du filtre de garde se trouve ainsi déplacée en numérique, dans le filtre de sous-échantillonnage. Il est en effet nettement plus facile (et moins coûteux) de créer des filtres sélectifs en numérique qu'en analogique.

### Exemple 3.4

On cherche comme précédemment à échantillonner la fonction rectangulaire  $rect_1(t)$ . Si l'on choisit  $f_e=10\text{Hz}$ , et qu'on choisit comme filtre de garde un filtre passe-bas de degré 1 de fréquence de coupure égale à 5Hz, on demande :

- De vérifier si la valeur maximale de la transformée de Fourier de  $rect_1(t)$  est affectée de moins de 1% d'erreur par repliement spectral ;
- Si ce n'est pas le cas, de donner le facteur de sur-échantillonnage minimal pour que cette condition soit respectée (en supposant qu'on ne change pas le filtre de garde).

Le filtre de garde a une pente de 20 dB/décade (degré 1) à partir de 5 Hz. Comme précédemment, on ne considérera que l'effet du premier spectre image, réduit à son enveloppe. L'atténuation du filtre de garde à 10 Hz est de  $20\text{dB} \cdot \log(10/5)/\log(10) = 6\text{dB}$ . Comme la transformée de Fourier de  $rect_1(t)$  tombe déjà de 30 dB entre  $F(0)$  et  $F(10)$  (Fig. 3.9), les 6 dB supplémentaires du filtre de garde ne permettent pas vraiment de vérifier la condition demandée (1% = atténuation de 40dB).

On peut refaire le même calcul pour des fréquences d'échantillonnages de 15 (il est clair qu'on ne doit pas aller jusque 30Hz, puisque c'est la valeur qui vérifie la condition *sans filtre de garde*). L'atténuation  $y$  est respectivement de  $20\text{dB} \cdot \log(15/5)/\log(10) = 9.5\text{dB}$ , ce qui fait à peu près 45 dB si on tient compte du fait que  $F(15)$  est déjà 35 dB au dessous de  $F(0)$ . En procédant par approximations successives, on trouve facilement que la fréquence  $f_e$  minimale est de 12 Hz :  $20\text{dB} \cdot \log(12/5)/\log(10) = 7.6\text{dB}$ , qui s'ajoutent aux 32 dB d'écart entre de  $F(12)$  et  $F(0)$ . Le facteur de sur-échantillonnage doit donc être de 1.2.

### 3.8 Filtre de lissage réel – Sur-échantillonnage (oversampling)

Le filtre de lissage peut également se trouver grandement simplifié par l'utilisation d'une fréquence d'échantillonnage intermédiaire, supérieure à la fréquence de départ, avant l'extrapolateur d'ordre 0. On l'obtient facilement par interpolation (up-sampling), ce qui implique une certaine charge de calcul au niveau du filtre numérique, que l'on gagne en simplification du gabarit du filtre de lissage.

#### Exemple 3.5 – Application : Sur-échantillonnage dans un lecteur de CD Audio

Le sur-échantillonnage est souvent utilisé dans l'étage de sortie des lecteurs de CD audio. La Fig. 3.24 illustre les contraintes sur le filtre de lissage sans et avec sur-échantillonnage.<sup>5</sup> On constate que la zone de transition du filtre est beaucoup plus large dans le second cas.

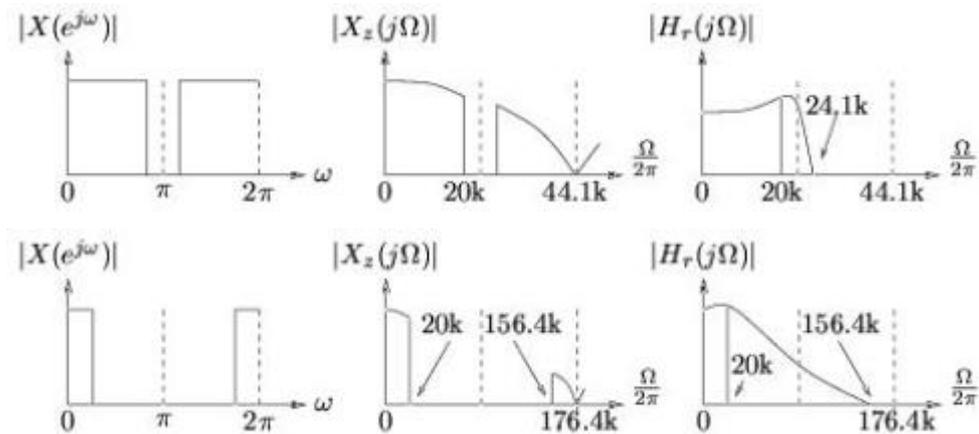


Fig. 3.24 a. Réponse en fréquence typique pour le filtre de lissage dans l'étage DAC d'un lecteur de CD, sans sur-échantillonnage (haut) et avec sur-échantillonnage d'un facteur 4 (bas) [d'après Phil Shniter, *Digital Signal Processing*, <http://cnx.rice.edu/content/m11006/latest/>].

### 3.9 Théorème de Shannon généralisé

Le *théorème de Shannon* peut être généralisé pour les signaux à bande étroite, c'est-à-dire pour les signaux pour lesquels l'amplitude spectrale se trouve confinée dans une bande de fréquence de largeur  $B$  centrée autour de  $f_0$  (Fig. 3.25).

<sup>5</sup> Paradoxalement, l'industrie du lecteur de CD se sert même du sur-échantillonnage comme un argument commercial.

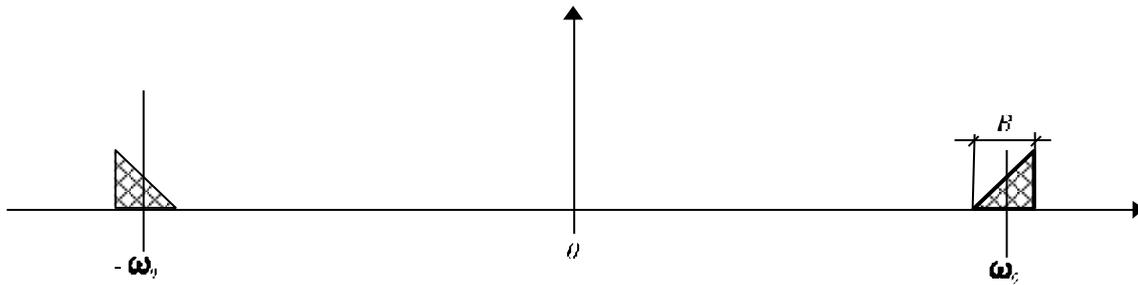


Fig. 3.25 Signal à bande étroite

Imaginons que l'on échantillonne un tel signal à une fréquence telle que  $Kf_e = f_0 - B/2$  ou  $Kf_e = f_0 + B/2$ , avec  $K$  entier (Fig. 3.26 et Fig. 3.27). On constate que des spectres image apparaissent de part et d'autre du spectre initial, mais que le recouvrement spectral est nul tant que  $f_e \geq 2B$ . Le signal original peut donc être reconstitué par le passage du signal impulsionnel  $f^+(t)$  dans un filtre passe-bande idéal dont les fréquences de coupure sont  $f_{c\pm} = f_0 \pm B/2$ .

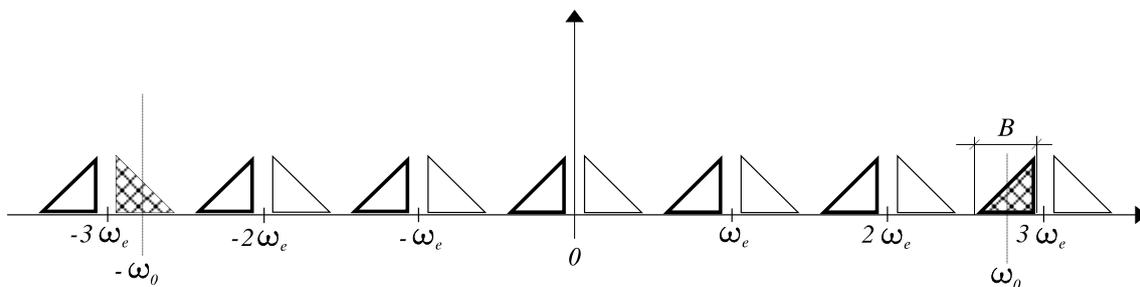


Fig. 3.26 Signal à bande étroite échantillonné avec  $3f_e = f_0 + B/2$

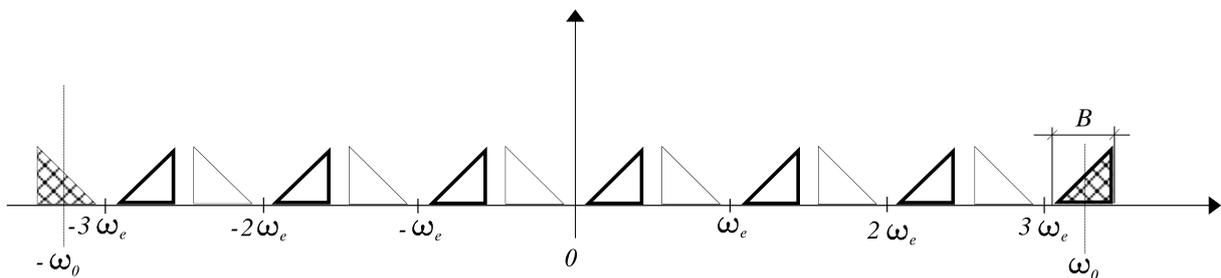


Fig. 3.27 Signal à bande étroite échantillonné avec  $3f_e = f_0 - B/2$

La fréquence d'échantillonnage nécessaire est donc fonction de la largeur effective  $B$  et non de la valeur centrale  $f_0$  (comme on aurait pu s'y attendre avec la version simple du théorème de Shannon) :

"Toute fonction  $f(t)$  dont le spectre est à bande étroite ( $f_0, B$ ) est complètement définie par ses échantillons  $f(nT_e)$  si  $f_e \geq 2B$  et que  $f_e$  respecte en outre l'une des conditions suivantes :  $Kf_e = f_0 - B/2$  ou  $Kf_e = f_0 + B/2$ , avec  $K$  entier "

On peut noter en passant que les signaux dont les spectres sont donnés à la Fig. 3.26 et la Fig. 3.27 sont identiques à ceux que l'on aurait obtenus en translatant tout d'abord le spectre du signal de  $Kf_e$  (obtenant ainsi un signal analogique à basse fréquence, entre 0 et  $f_e/2$ , et en échantillonnant ce signal à  $f_e$ . La reconstruction du signal de départ implique donc bien entendu de *se souvenir* que le signal, avant échantillonnage, avait son spectre dans l'intervalle  $f_0 \pm B/2$

### Exemple 3.6 – Application : Codage en sous-bandes

Le théorème de Shannon généralisé est mis à profit dans les systèmes dit *d'analyse en sous-bandes*, où un signal de départ est décomposé, par filtrage passe-bande, en plusieurs signaux à bande étroite, dont la somme fournit le signal de départ. Ces signaux sont alors échantillonnés séparément en respectant le théorème de Shannon généralisé. Ceci permet de transformer un flux d'échantillons de départ (large-bande, de largeur de bande  $f_M$  donnée) en  $M$  flux à bande étroite ( $f_M/M$ ) en conservant le débit total en nombre d'échantillons par seconde.

En pratique, la décomposition en sous-bandes est généralement réalisée directement dans le domaine numérique, c.-à-d. à partir d'un signal déjà échantillonné. Ainsi par exemple (Fig. 3.28), le signal  $x(n)$  dont le spectre  $A^+(f)$ , de largeur  $f_e/2$ , est donné à la Fig. 3.29 est décomposé en deux sous-bandes de spectre  $A^+_L(f)$  et  $A^+_H(f)$  de largeur  $f'_e/2$  (avec  $f'_e = f_e/2$ ). Les deux signaux numériques  $x_L(t)$  et  $x_H(t)$  correspondants sont tous deux sous-échantillonnés à  $f'_{e=} f'_e/2$ . Le rééchantillonnage de  $x_L(t)$  respecte donc le théorème de Shannon, et celui de  $x_H(t)$  respecte le théorème de Shannon généralisé. Les deux signaux numériques correspondants  $x'_L(n)$  et  $x'_H(n)$  portent bien la même information que le signal  $x(n)$  qui aurait été obtenu par échantillonnage direct de  $x(t)$  à une fréquence d'échantillonnage  $f_e$  (pour respecter le théorème de Shannon sur l'échantillonnage de ce signal). Le nombre total d'échantillons utilisés pour stocker cette information dans  $x'_L(n)$  et  $x'_H(n)$  est bien identique à celui utilisé dans  $x(n)$ . Le signal  $x'_L(n)$  peut être vu comme une « approximation grossière », basse-fréquence, de  $x(n)$ , tandis que  $x'_H(n)$  porte au contraire les détails haute-fréquence du signal.

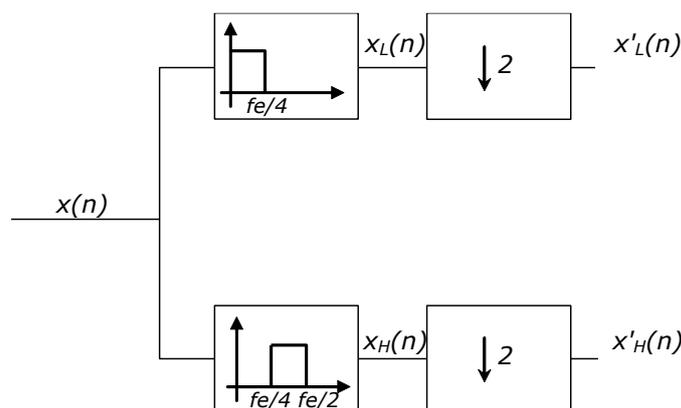


Fig. 3.28 Décomposition en deux sous-bandes

Notons que la même opération peut alors être réitérée sur  $x'_L(n)$  et  $x'_H(n)$  séparément, ce qui conduit à une décomposition récursive en un nombre de sous-bandes de plus en plus élevé, et de fréquence d'échantillonnage de plus en plus faible. Il est également possible de réaliser

directement une décomposition en  $M$  sous-bandes spectrales, en utilisant les filtres et les décimateurs *ad hoc*.

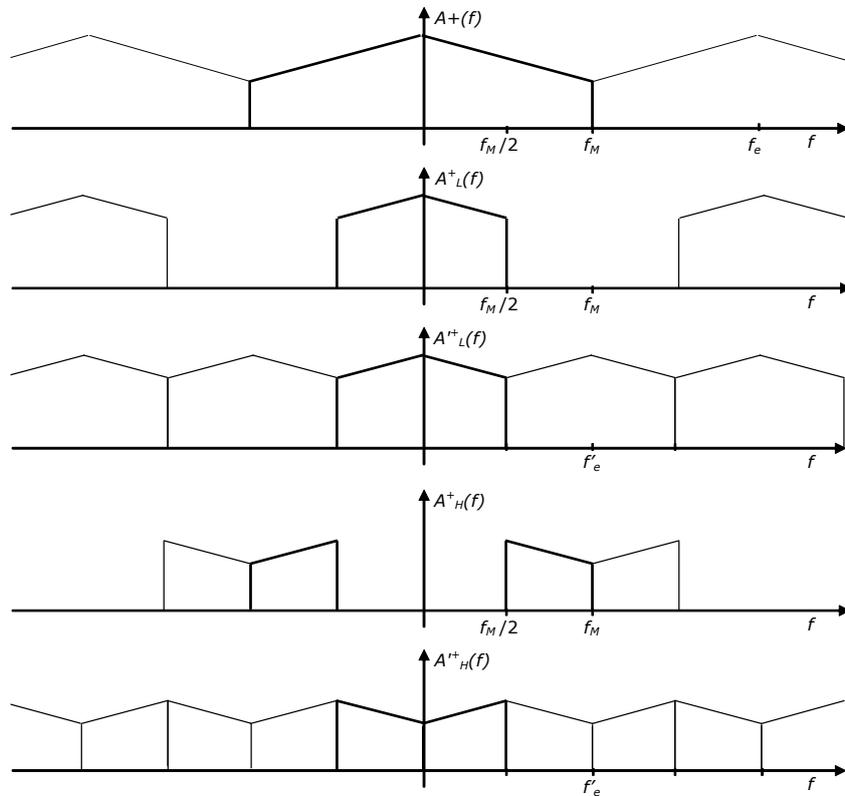


Fig. 3.29 Effet spectral de la décomposition d'un signal en 2 sous-bandes. En gras, la bande de fréquence utile.

La reconstruction du signal original est alors possible (Fig. 3.30 et Fig. 3.31).

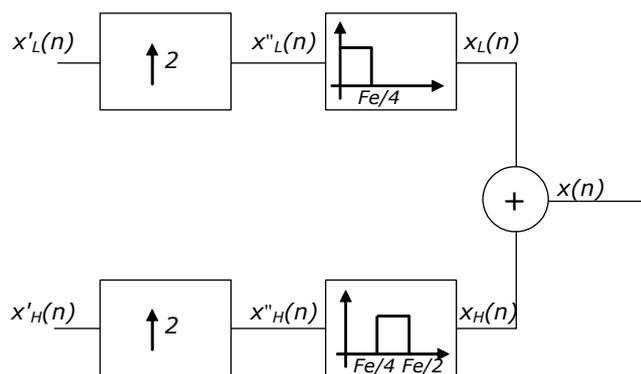


Fig. 3.30 Reconstitution à partir de deux sous-bandes

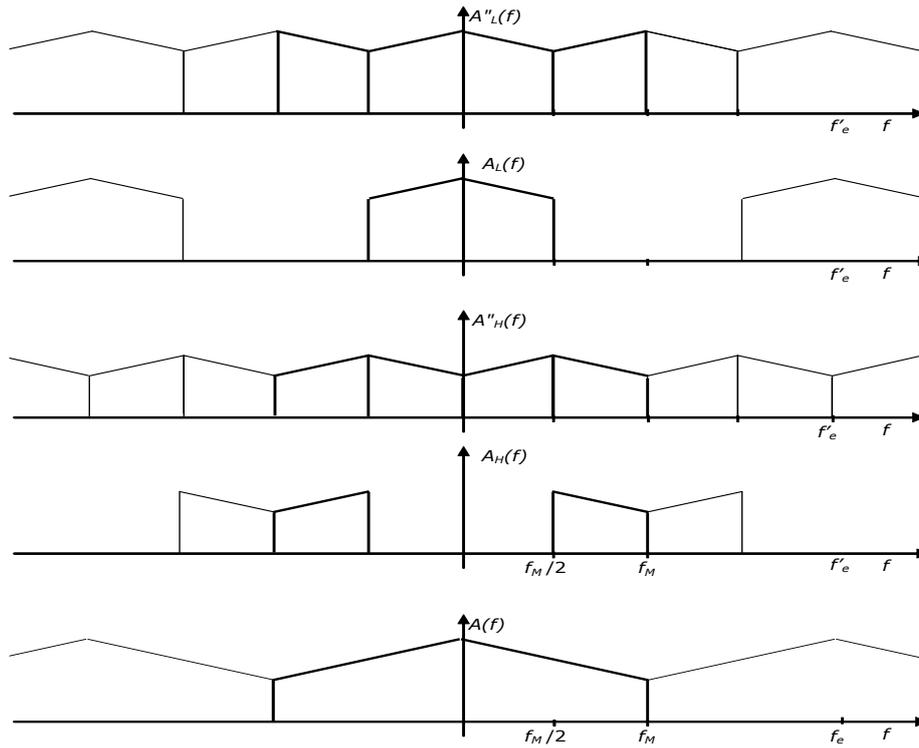


Fig. 3.31 Effet spectral de la reconstitution d'un signal à partir de 2 sous-bandes. En gras, la bande de fréquence utile.

Ce type de décomposition est utilisé par exemple pour le codage d'images selon la norme JPEG2000, ainsi que pour le codage/décodage audio selon la norme MPEG1-layer3 (plus communément appelée mp3). L'idée est alors d'estimer dans quelle mesure l'œil ou l'oreille est sensible à l'information contenue dans chaque sous-bande, en fonction du contenu des sous-bandes adjacentes, et de quantifier cette sous-bande avec un pas de quantification (voir Chapitre 5) qui est fonction de cette sensibilité. On a en effet pu mettre en évidence que la perception d'un signal audio ou visuel dans une bande de fréquences dépend du contenu des bandes adjacentes (effet dit *psycho-acoustique* ou *psycho-visuel*). Une sous-bande peu importante au niveau perceptuel est alors quantifiée avec peu de bits, ce qui permet une compression importante du signal en minimisant la dégradation perçue (Fig. 3.32)

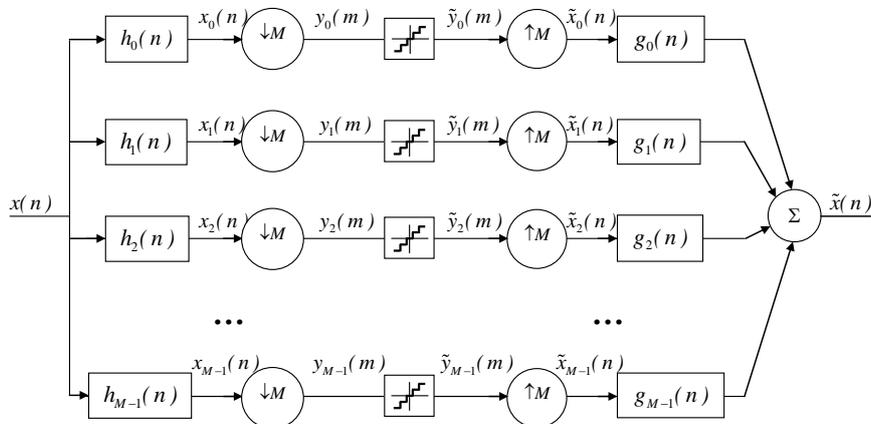
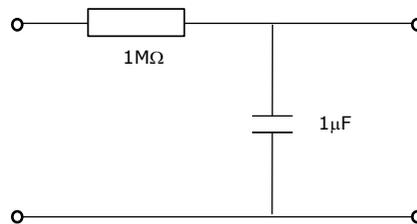


Fig. 3.32 Codeur en  $M$  sous-bandes : la quantificateur est adapté dynamiquement au contenu de chaque sous-bande.

## Exercices

### Exercice 3.1

Soit  $x(t)$  la réponse impulsionnelle du quadripôle ci-dessous :



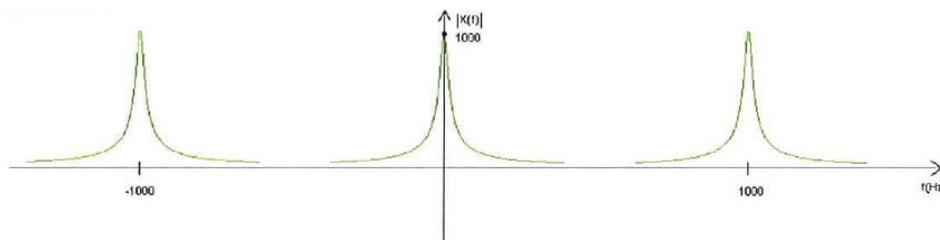
- Calculer et dessiner  $x(t)$
- Calculer l'expression analytique et dessiner (module uniquement; entre -3000 et 3000 Hz) la transformée de Fourier du signal  $x(t)$ .
- Déduire de la réponse *b* le graphique de la transformée de Fourier (module uniquement; entre -3000 et 3000 Hz) du signal  $x^+(t)$  échantillonné avec une fréquence d'échantillonnage de 1000 Hz.
- Dessiner  $x_1(t)$ , réponse du circuit à un train d'impulsions de Dirac  $\delta_2(t)$  de grandeur unitaire et de période égale à 2 secondes (supposé exister depuis  $t=-\infty$  jusque  $t=+\infty$ ).
- Déduire de la réponse *b* le graphique de la transformée de Fourier (module uniquement; entre -5 et 5 Hz) du signal  $x_1(t)$ .

### Solution

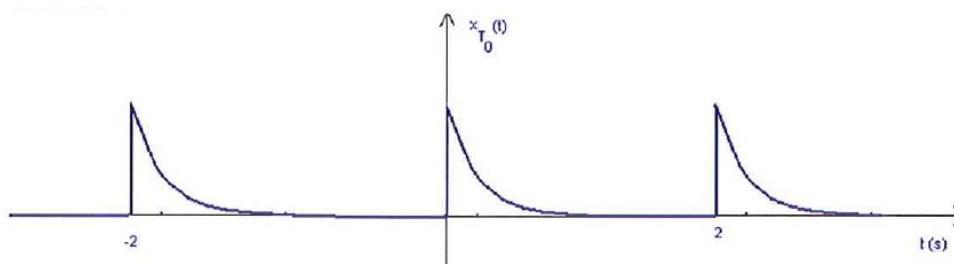
(a)  $x(t) = (1/RC) \exp(-t/RC) \varepsilon(t) \rightarrow x(t) = \exp(-t) \varepsilon(t)$

(b)  $X(f) = \frac{1}{1 + j2\pi f}$

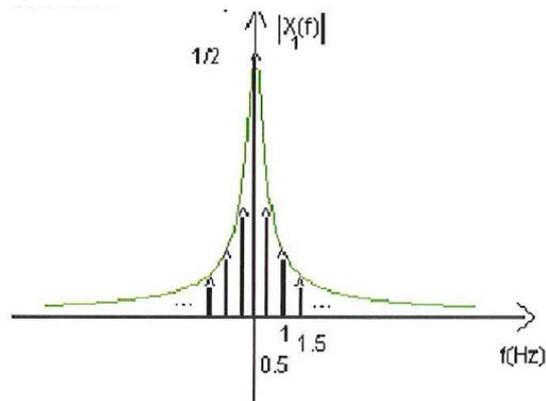
(c)



(d)

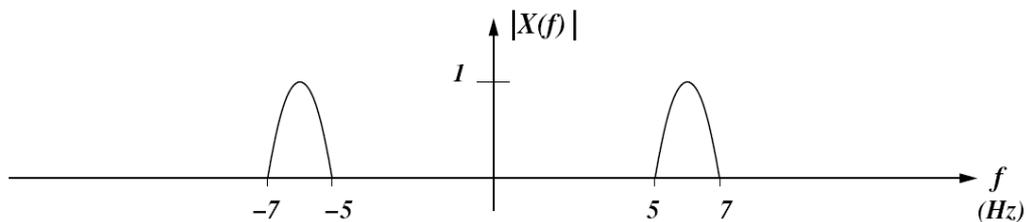


(e)



### Exercice 3.2

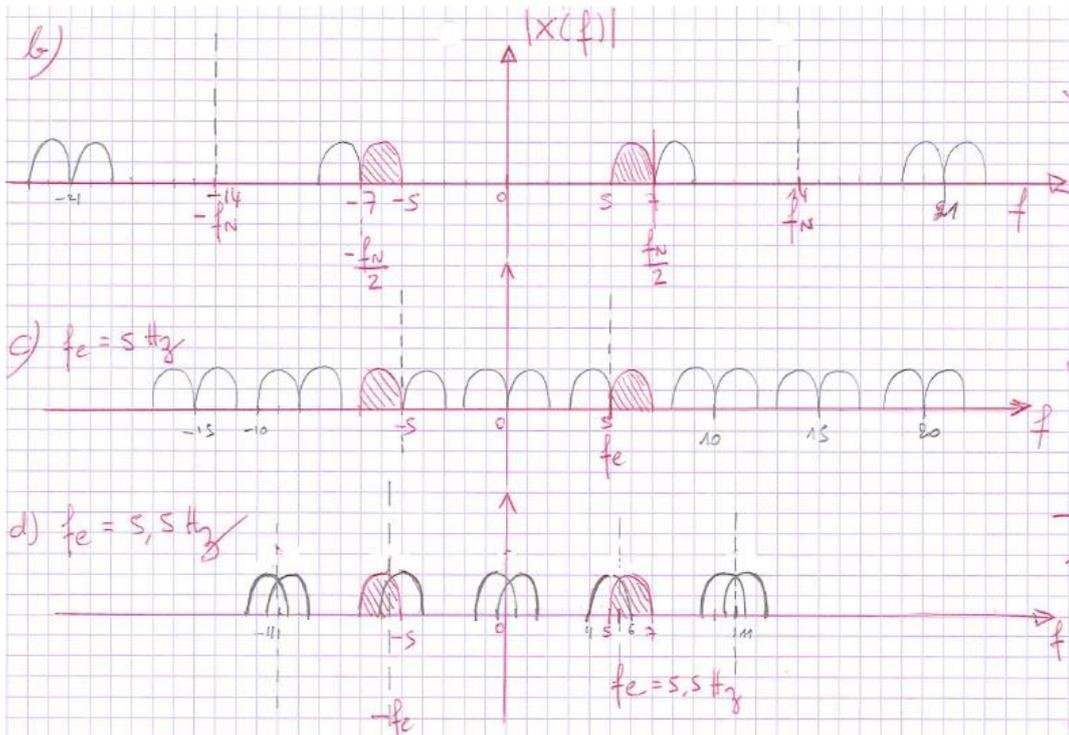
- (a) En appliquant le théorème de Shannon (théorème de l'échantillonnage), déterminez la fréquence d'échantillonnage minimale  $f_e$  du signal  $x(t)$  dont le module  $|X(f)|$  du spectre est représenté à la figure ci-dessous.



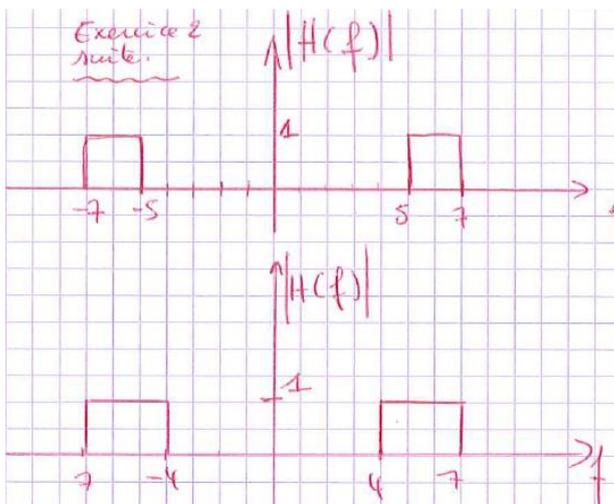
- (b) Tracer le module du spectre du signal échantillonné à la fréquence de Nyquist dans l'intervalle  $-23 \text{ Hz} < f < 23 \text{ Hz}$ .
- (c) Tracer le module du spectre du signal échantillonné à la fréquence  $f_e = 5 \text{ Hz}$  dans l'intervalle  $-10 \text{ Hz} < f < 10 \text{ Hz}$ .
- (d) Tracer le module du spectre du signal échantillonné à la fréquence  $f_e = 5.5 \text{ Hz}$  dans l'intervalle  $-10 \text{ Hz} < f < 10 \text{ Hz}$ .
- (e) Peut-on utiliser des fréquences d'échantillonnage inférieures à celle qui a été déterminée au point (a) ? Quelle serait la fréquence d'échantillonnage minimale ? (Utilisez le théorème de Shannon généralisé).
- (f) En utilisant le théorème de Shannon généralisé, déterminez quelle serait la fréquence d'échantillonnage minimale si le contenu spectral n'était pas compris dans l'intervalle  $5 \text{ Hz} - 7 \text{ Hz}$  mais dans l'intervalle  $4 \text{ Hz} - 7 \text{ Hz}$ .
- (g) Déterminez la réponse en fréquence d'un filtre d'interpolation permettant de reconstruire parfaitement le signal original  $x(t)$ . Plusieurs bandes passantes étant possibles, représentez le filtre idéal avec la plus petite bande passante.

### Solution

- (a)  $f_e = 14 \text{ Hz}$



- (e) 5 Hz
- (f) 7 Hz
- (g)



**Exercice 3.3**

Considérons le signal  $x(t) = 2 \sin(12\pi t) + 3 \cos(25\pi t) - \sin(30\pi t)$ .

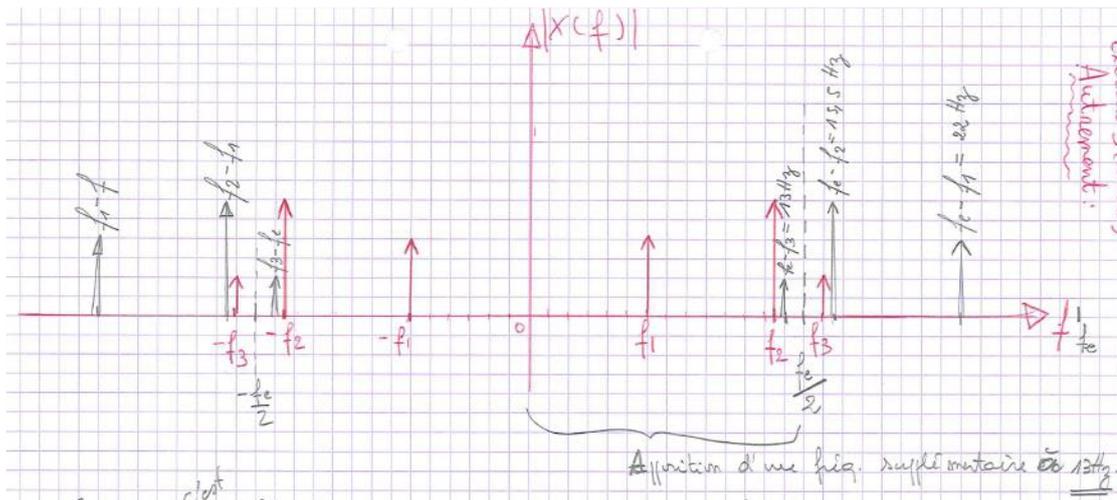
- (a) Pour quelles fréquences d'échantillonnage évite-t-on l'aliasing ?
- (b) Supposons que l'on filtre le signal  $x(t)$  avec un filtre d'antialiasing idéal de fréquence 14 Hz puis qu'on l'échantillonne à la fréquence  $f_e = 28$  Hz. Quel est le signal échantillonné  $x(n)$  ? Quel est le signal  $y(t)$  reconstruit à partir de  $x(n)$  si l'on utilise l'interpolateur idéal ?

- (c) Supposons qu'on échantillonne le signal  $x(t)$  avec la même fréquence  $f_e = 28$  Hz, mais, cette fois, sans préfiltrage. Quel est le signal échantillonné  $x(n)$  dans ce cas-ci ? Quel est son spectre ? Quel est le signal  $y(t)$  reconstruit à partir de  $x(n)$  si l'on utilise l'interpolateur idéal ?
- (d) Même questions qu'au point (c) mais avec le signal
- $$x(t) = 2 \sin(12\pi t) + 3 \cos(25\pi t) - \cos(30\pi t).$$

Indice pour les points (c) et (d) : n'oubliez pas la phase des signaux !

**Solution**

- (a) supérieures ou égales à 30 Hz
- (b)  $x(n) = 2 \sin(2\pi F_1 n) + 3 \cos(2\pi F_2 n)$  avec  $F_1 = 6/28$  et  $F_2 = 12.5/28$   
 $y(t) = 2 \sin(12\pi t) + 3 \cos(25\pi t)$
- (c)



Cependant c'est  $X(f)$ , complexe, qui est dupliqué et traduit. Pour la phase, réfléchir avec  $X(f)$  complexe ou bien faire comme suit :

$$x(n) = 2 \sin(2\pi F_1 n) + 3 \cos(2\pi F_2 n) - \sin(2\pi F_3 n)$$

$$= 2 \sin(2\pi F_1 n) + 3 \cos(2\pi F_2 n) + \sin(2\pi F_4 n)$$

où  $F_3 = 15/28 = 1 - F_4$  avec  $F_4 = 13/28$

$$y(t) = 2 \sin(12\pi t) + 3 \cos(25\pi t) + \sin(26\pi t)$$

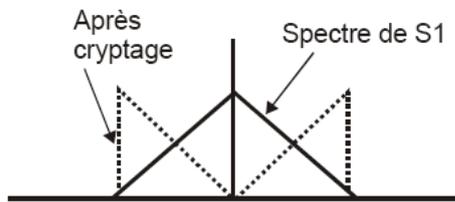
- (d)  $x(n) = 2 \sin(2\pi F_1 n) + 3 \cos(2\pi F_2 n) - \cos(2\pi F_4 n)$   
 $y(t) = 2 \sin(12\pi t) + 3 \cos(25\pi t) - \cos(26\pi t)$

**Exercice 3.4 (d'après J. Auvray)**

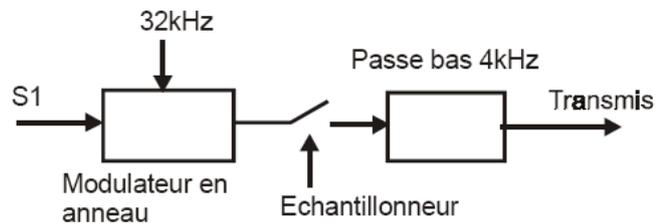
Un signal vocal S1 est soigneusement filtré de façon à ne posséder aucune composante spectrale au-delà de 4 kHz.

On désire le crypter de façon à le rendre incompréhensible sur la ligne téléphonique. Pour cela on procède de la façon suivante : le signal S1 est utilisé pour moduler en amplitude, avec suppression de porteuse, une porteuse de fréquence 32 kHz (ce qui peut se faire avec un modulateur en anneau). Le signal obtenu est échantillonné à une fréquence  $f_e$  et les impulsions obtenues filtrées par un filtre passe-bas idéal de fréquence de coupure 4 kHz. Cette procédure permet de renverser le contenu spectral initial.

- (1) Quelle valeur minimale doit-on donner à  $f_e$  pour que le spectre du signal transmis soit le résultat du renversement suivant l'axe des fréquences du spectre de S1 (fréquences basses et hautes permutées comme schématisé sur la figure (a)) ?
- (2) Représentez dans la bande  $[-10,+10]$  kHz l'allure du spectre à la sortie de l'échantillonneur avant filtrage par le filtre passe-bas.
- (3) Comment peut-on réaliser le décryptage du signal transmis, c'est-à-dire reconstituer le signal initial ?



(a)

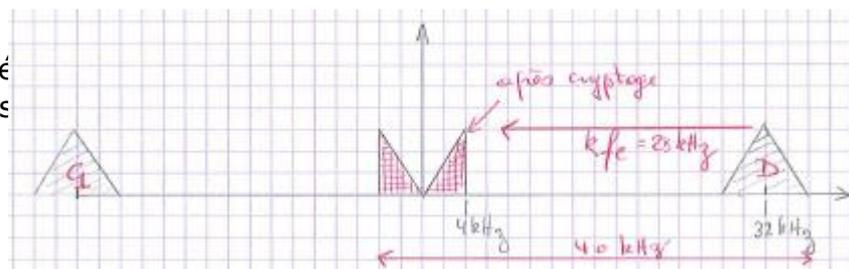


(b)

**Solution**

Réponses : a) 12kHz

NB : ce procédé était transmis



ette chaîne

On doit translater le spectre de  $kf_e = 28$  kHz. A la translation suivante, on ne doit pas toucher la bande  $]-4, 4[$  kHz  $\rightarrow (k+1)f_e \geq 40$  kHz or  $f_e = 28$  kHz/k. On trouve que  $k \leq 7/3 = 2.33$  kHz avec k entier donc on prend  $k = 2$ . D'où  $f_e = 14$  kHz.

!!! autre approche : mettre le coin droit (celui à 36) en 0  $\Rightarrow kf_e = 36$  ; éviter que l'image  $k-1$  ne recouvre la bande 0-4 kHz par son coin gauche (celui à 28)  $\Rightarrow (k-1)f_e \leq 24$ . On trouve alors  $k=3$  et  $f_e=12$  !

(2) à faire

On constate alors que Shannon généralisé n'est pas respecté et il y a recouvrement. Mais ce n'est pas grave, vu le filtre de lissage.

(3) Il suffit d'effectuer la même opération (montrez-le graphiquement)

### Exercice 3.5

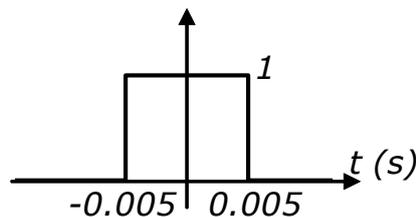
Comment un lecteur de CD à facteur de suréchantillonnage = 10 diffère-t-il d'un lecteur de CD sans suréchantillonnage ?

#### Solution

voir notes de cours

### Exercice 3.6

On cherche à échantillonner le signal ci-dessous sans utiliser de filtre de garde :

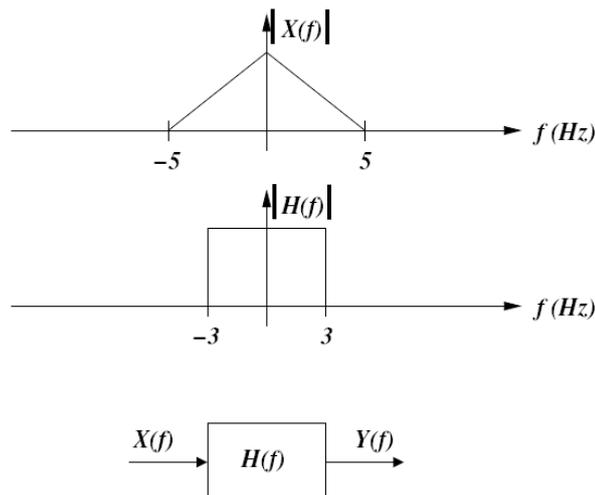


Quelle est la fréquence d'échantillonnage minimale qui permet d'assurer que le repliement spectral au droit du lobe principal du spectre du signal soit de moins de 5% de sa valeur réelle ?

### Exercice 3.7

Le module du spectre d'un signal temps continu  $x(t)$  est représenté ci-dessous.

(a) Si  $x(t)$  est échantillonné à la fréquence  $f_e = 8$  Hz, tracez le module du spectre du signal échantillonné dans l'intervalle  $[-13 \text{ Hz}, 13 \text{ Hz}]$ .

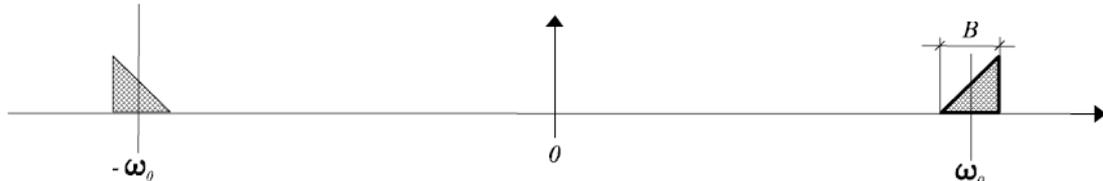


(b)  $x(t)$  est appliqué à un système dont le module de la réponse en fréquence  $H(f)$  est représenté à la deuxième figure. Tracez le module du spectre du signal de sortie  $y(t)$ .

(c) Si  $y(t)$  est ensuite échantillonné à une fréquence  $f_e = 8$  Hz, tracez le spectre de ce signal échantillonné dans l'intervalle  $[-13 \text{ Hz}, 13 \text{ Hz}]$ .

**Exercice 3.8**

Le signal dont la transformée de Fourier (en module) est schématisée ci-dessous (où  $f_0 = 1000$  Hz et  $B = 100$  Hz) doit être échantillonné.



Donnez la plus petite fréquence d'échantillonnage qui vérifie le théorème de Shannon généralisé (non recouvrement des spectres).

**Exercice 3.9**

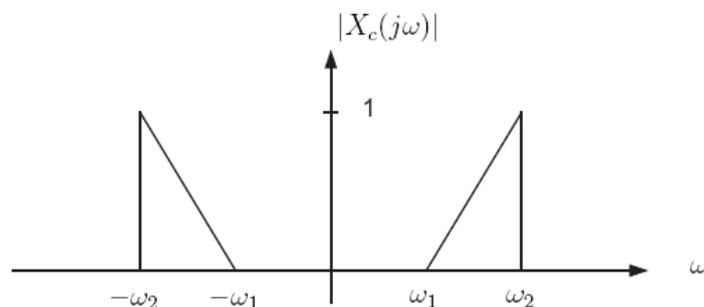
On considère le signal à temps continu  $x(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$  avec  $-\infty < t < \infty$  et  $f_1 = 500$  Hz et  $f_2 = 3000$  Hz.

- Déterminez les fréquences normalisées correspondantes ainsi que le signal échantillonné  $x(n)$  pour la fréquence d'échantillonnage  $f_e = 8000$  Hz.
- Quelle est la période fondamentale  $N_p$  de ce signal à temps discret ?
- Quelle est la fréquence d'échantillonnage minimale qui satisfait le théorème de l'échantillonnage ?
- Quelle est la fréquence d'échantillonnage minimale telle que l'échantillonnage se produit sur les valeurs crêtes des deux sinusoïdes ?

**Exercice 3.10**

Le module du spectre d'un signal réel à temps continu et à bande étroite est schématisé à la figure suivante où  $\omega_B = \omega_2 - \omega_1$  et  $\omega_2 = 4\omega_B$ .

- Déterminez la fréquence d'échantillonnage minimale qui permette la reconstruction parfaite du signal.
- Représentez le module du spectre du signal échantillonné à cette fréquence.
- Déterminez le filtre d'interpolation permettant la reconstruction parfaite du signal original.
- Supposons que  $\omega_2 = 4.5$ . Déterminez la fréquence d'échantillonnage minimale qui autorise une reconstruction parfaite.



**Exercice 3.11**

L'échantillonnage idéal suppose l'échantillonnage d'un signal  $x_c(t)$  aux instants  $t = kT$ .

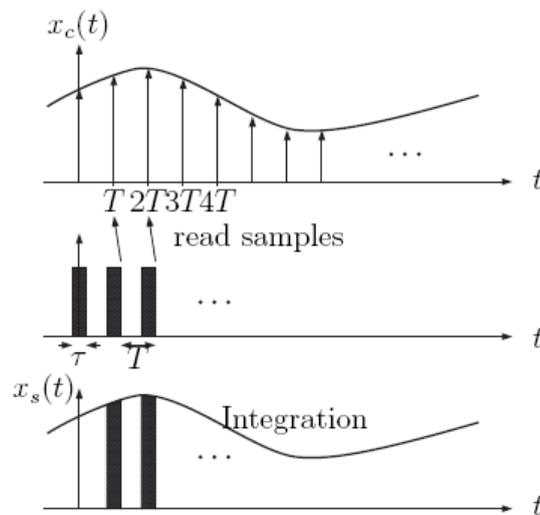
En pratique, un convertisseur A/D échantillonne le signal en chargeant un condensateur pendant la période  $\tau$ . La tension du condensateur est alors une mesure de la valeur échantillonnée. L'action de charger le condensateur correspond à l'intégration du signal sur cette période  $\tau$  :

$$\frac{1}{\tau} \int_{t-\frac{\tau}{2}}^{t+\frac{\tau}{2}} x_c(\nu) d\nu = x_c(t) * \frac{1}{\tau} \text{rect} \left( \frac{t}{\tau} \right); \quad t < \tau \quad .$$

L'expression du signal échantillonné est

$$x_s(t) = \left( x_c(t) * \frac{1}{\tau} \text{rect} \left( \frac{t}{\tau} \right) \right) \sum_{k=-\infty}^{\infty} \delta(t - kT) \quad .$$

- (a) Déterminez la transformée de Fourier du signal échantillonné  $x_s(t)$  et représentez son module lorsque  $\tau = T$ .  
 (b) Quelle est l'influence de  $\tau$  dans le domaine fréquentiel ? Calculez l'atténuation maximale du spectre  $X_c$ .

**Solution**

$$X_s(f) = X_c(f) \frac{1}{\tau} \frac{\sin(\tau\pi f)}{\pi f} * \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta(f - \frac{k}{T})$$

# CHAPITRE 4

## TRANSFORMEE DE FOURIER DISCRETE

John Tukey, (1915-2000), statisticien américain. Après une licence en chimie et deux en mathématiques à Brown et Princeton University, il devint professeur de statistiques à Princeton et continua en parallèle une carrière de chercheur aux Bell Labs. Il travailla notamment dans le domaine de l'analyse statistique robuste. Avec l'aide de James Cooley, il mit au point l'algorithme de la Transformée de Fourier Rapide (FFT). John Tukey est également l'inventeur du mot « software ».



James Cooley (1926-), mathématicien américain, pionnier dans le développement du traitement numérique du signal. Alors qu'il travaillait dans les laboratoires de recherche de IBM à New York, il co-inventa la FFT avec J. Tukey. Cet algorithme est aujourd'hui l'un des plus utilisés en sciences et en ingénierie, de la géophysique à l'astronomie en passant la biologie. Il a volontairement été rendu public par IBM dès sa découverte en 1965, sans aucune protection intellectuelle (sans brevet), afin de favoriser le développement de logiciels.



Ce troisième chapitre est consacré à l'étude du seul outil réellement utilisable en pratique pour calculer la transformée de Fourier d'un signal déterministe : la transformée de Fourier Discrète (4.1). Nous verrons que la transformée de Fourier discrète correspond à la décomposition d'un vecteur sur une base orthogonale, et qu'elle n'est donc qu'une simplification (en termes de calculs) de la transformée de Fourier à temps discret introduite au chapitre 2. Cette simplification introduit cependant quelques artefacts dont il faut être conscient, ne fût-ce que pour pouvoir les minimiser lorsque c'est possible.

Nous montrerons qu'il est possible de calculer la transformée de Fourier discrète de façon très efficace, à l'aide d'algorithmes spécialisés, dits de transformée de Fourier rapide (4.2).

La transformée de Fourier discrète doit être considérée comme un outil, d'autant plus efficace qu'on l'utilise sous sa forme rapide. En pratique, elle est essentiellement utilisée comme :

- Outil d'observation spectrale (4.3), dans le but de calculer la TTFD de signaux numériques, et même le plus souvent la transformée de Fourier de signaux analogiques sous-jacents.

- Outil de convolution linéaire rapide (4.4).

La plupart des conclusions qui seront tirées dans ce chapitre seront également applicables à l'analyse spectrale des signaux aléatoires, qui sera l'objet du chapitre suivant.

## 4.1 Transformée de Fourier Discrète

Si la transformée de Fourier à temps discret introduite au chapitre 2 est conceptuellement plus simple que la transformée de Fourier d'un signal analogique, il n'en reste pas moins que le calcul de la TTFD nécessite en principe une charge de calcul infinie, puisque la série qui la définit comporte un nombre infini de termes, et que l'estimation doit en être faite pour toutes les valeurs de la fréquence normalisée  $F$  entre 0 et 1.<sup>1</sup>

C'est la raison pour laquelle la *transformée de Fourier Discrète* ou *TFD* (en anglais *Discrete Fourier Transform*, ou *DFT*) a été introduite. Son calcul est en effet limité à un nombre fini de valeurs de  $n$  et pour un nombre fini de valeurs de  $F$ .

### 4.1.1 Définition

Considérons une suite finie de  $N$  échantillons  $\{x(n)\} = \{x(0), x(1), \dots, x(N-1)\}$ . On définit sa *transformée de Fourier Discrète* comme la suite  $\{X(k)\}$  (Fig. 4.1):

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jn(k\frac{2\pi}{N})} \quad (k = 0 \dots N-1) \quad (4.1)$$

Il ne s'agit donc ni plus ni moins de la TTFD d'un signal numérique dont on ne considère que les  $N$  premiers échantillons, calculée pour les  $N$  pulsations normalisées  $\varphi = \{0, 2\pi/N, 4\pi/N, \dots, 2(N-1)\pi/N\}$ , c.-à-d. pour les  $N$  fréquences normalisées  $F = \{0, 1/N, 2/N, \dots, (N-1)/N\}$ . A ce titre, la TFD jouit des mêmes interprétations que la TTFD : interprétation géométrique (cf. 4.1.2) et décomposition sur une base (cf. 4.1.3). Nous verrons également qu'on peut l'interpréter comme le TTFD d'un signal périodique (cf. 4.1.4).

Nous verrons plus loin que la *transformée de Fourier Discrète inverse* est donnée par :

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{jnk\frac{2\pi}{N}} \quad (n = 0 \dots N-1) \quad (4.2)$$

---

<sup>1</sup> Dans les exemples donnés au chapitre précédent, le premier problème n'apparaissait pas, puisque les séquences numériques étaient toujours limitées en nombre d'échantillons. Le second était « résolu » par Matlab en calculant les TTFD sur une grille de valeurs entre  $F=0$  et  $\frac{1}{2}$ . La fonction `freqz` permet d'ailleurs de fixer le nombre de points de cette grille.

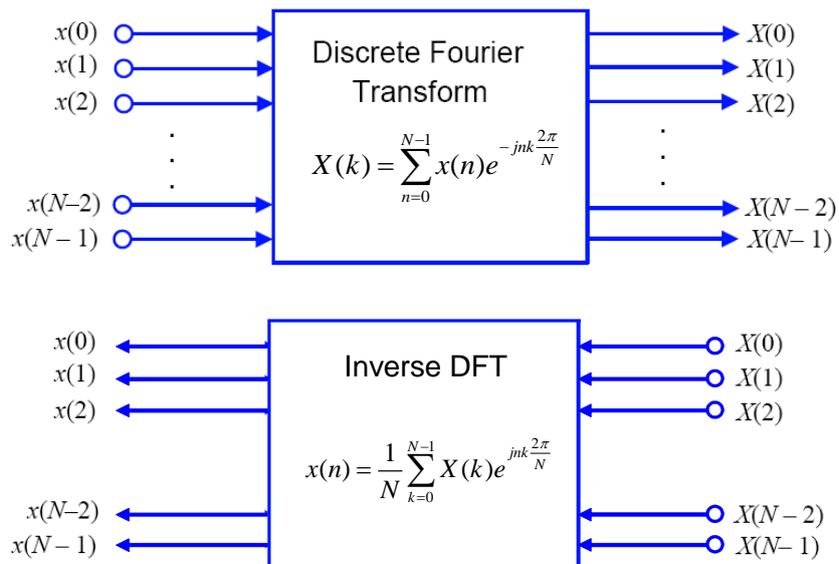


Fig. 4.1 TFD et TFD inverse

### 4.1.2 Interprétation géométrique

Comme pour la TFTD, l'évaluation de l'expression (4.1) pour  $k$  fixé correspond à une somme de vecteurs complexes  $e^{-jn\varphi}$ , avec  $\varphi = k \frac{2\pi}{N}$ , pondérés par les échantillons  $x(n)$ .

#### Exemple 4.1

La TFD de la suite de 4 échantillons  $\{x(n)\} = \{x(0), x(1), x(2), x(3)\}$  correspond aux quatre valeurs complexes  $\{X(k)\} = \{X(0), X(1), X(2), X(3)\}$  obtenues comme à la Fig. 4.2, où l'on a symboliquement placé les échantillons près des vecteurs complexes auxquels ils se rapportent dans la somme (4.1).

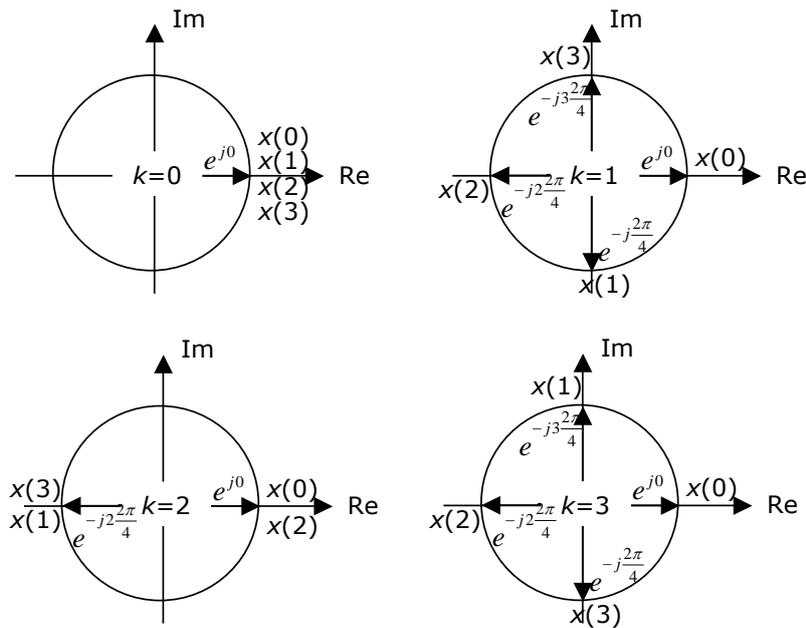


Fig. 4.2 Calcul géométrique de la TFD de  $\{x(0), x(1), x(2), x(3)\}$

### 4.1.3 Interprétation vectorielle

Les éléments de la suite  $\{x(0), x(1), \dots, x(N-1)\}$  peut être vue comme les composantes d'un vecteur  $\mathbf{x} = [x(0), x(1), \dots, x(N-1)]^T$  dans un espace à  $N$  dimensions.

L'équation (4.2) exprime alors que  $\mathbf{x}$  comme une combinaison linéaire de  $N$  vecteurs de base  $\mathbf{w}_k$  ( $k=0, \dots, N-1$ ), avec  $\mathbf{w}_k = [e^{j0}, e^{jk\frac{2\pi}{N}}, e^{j2k\frac{2\pi}{N}}, \dots, e^{j(N-1)\frac{2\pi}{N}}]^T$  (Fig. 4.4).

Il est facile de montrer que ces vecteurs de base sont orthogonaux<sup>2</sup> entre eux (leurs composantes décrivent en effet des exponentielles imaginaires dont les périodes sont entières et sont entre elles dans des rapports entiers).

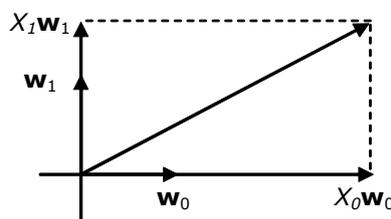


Fig. 4.3 Décomposition sur les vecteurs de base (cas  $N=2$ )

<sup>2</sup> Le produit scalaire est ici celui défini classiquement dans les espaces vectoriels.

### Exemple 4.2

Considérons une TFD sur 16 valeurs d'entrée ( $N=16$ ). Il est facile d'afficher sous Matlab les parties réelles des composantes des cinq premiers vecteurs de base  $\mathbf{w}_k$  ( $k=0, \dots, 4$ ).

```
for k=0:4
    w=exp(j*2*pi/16*k*(0:15));
    subplot(5,1,k+1);
    stem(real(w));
end
```

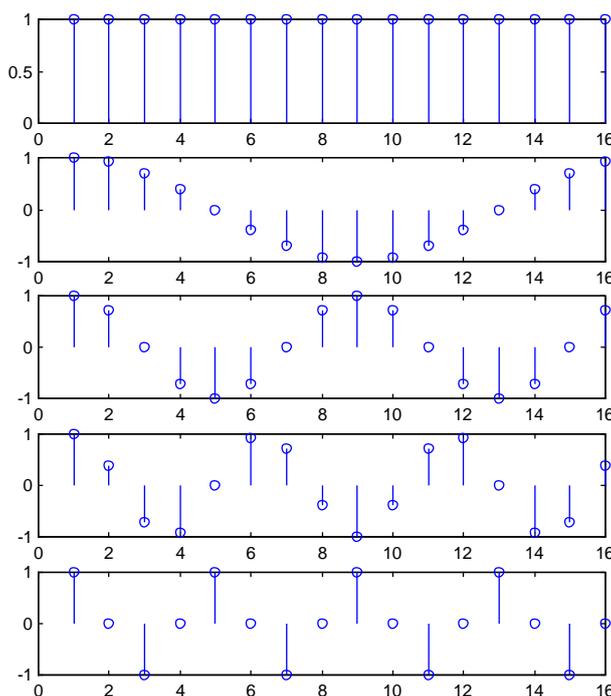


Fig. 4.4 Partie réelle des cinq premières fonctions de base pour une TFD sur 16 valeurs ( $N=16$ )

On constate que les vecteurs de base complexes sous-jacents sont bien les exponentielles imaginaires de périodes entières sur 16 échantillons (la première correspondant au cas particulier d'une période de 1 échantillon).

La décomposition peut être exprimée sous forme matricielle :

$$\mathbf{W}\mathbf{X} = \mathbf{x}$$

$$\text{avec } \mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{N-1}] \quad (4.3)$$

$$\mathbf{X} = [X_0, X_1, \dots, X_{N-1}]^T$$

La matrice  $\mathbf{W}$  (dont les colonnes sont les vecteurs de base de la décomposition) étant orthogonale, on obtient donc facilement le vecteur  $\mathbf{X}$  (vecteurs des coefficients de la décomposition) par multiplication pour la matrice adjointe :

$$(\mathbf{W}^H \mathbf{W})\mathbf{X} = \mathbf{W}^H \mathbf{x}$$

avec  $\mathbf{W}^H \mathbf{W} = \begin{bmatrix} \langle \mathbf{w}_0, \mathbf{w}_0 \rangle & 0 & \dots & 0 \\ 0 & \langle \mathbf{w}_1, \mathbf{w}_1 \rangle & & \\ \dots & \dots & \dots & \\ 0 & 0 & \dots & \langle \mathbf{w}_{N-1}, \mathbf{w}_{N-1} \rangle \end{bmatrix}$  (4.4)

Les éléments de la diagonale principale de cette dernière matrice sont les carrés des normes des vecteurs de base, et valent  $N$  (les vecteurs de base ne sont donc pas orthonormés, comme c'était le cas en transformée de Fourier) :

$$\langle \mathbf{w}_k, \mathbf{w}_k \rangle = \sum_{n=0}^{N-1} e^{jnk\frac{2\pi}{N}} e^{-jnk\frac{2\pi}{N}} = N \quad \text{pour tout } k$$
 (4.5)

Par conséquent :

$$\mathbf{X} = \frac{1}{N} \mathbf{W}^H \mathbf{x}$$
 (4.6)

Cette dernière expression est à comparer à celle de la TFD (4.2) : les coefficients  $X_k$  de la décomposition sont donnés par les valeurs  $X(k)$  de la TFD (produits scalaires de chaque  $\mathbf{w}_k$  avec  $\mathbf{x}$ ) divisés par  $N$  (carré de la norme des vecteurs de base) :

$$X_k = \frac{X(k)}{N} \quad (\text{pour } k = 0, 1, \dots, N-1)$$
 (4.7)

Autrement dit, les valeurs de la TFD  $X(k)$  d'une suite d'échantillons sont égales à  $N$  fois les coefficients  $X_k$  de sa décomposition sur les vecteurs de base.

On constate en passant qu'en combinant (4.7) et (4.3), on retrouve l'expression annoncée pour la TFD inverse (4.2).

### Exemple 4.3

Considérons la TFD d'un des vecteurs de base, par exemple  $\mathbf{w}_2$  pour  $N=8$ . Il est clair en vertu du raisonnement précédent que toutes les valeurs  $X(k)$  de la TFD seront nulles, sauf  $X(2)$ , qui doit valoir 8. On obtient effectivement ce résultat sous Matlab :

```
x=exp(j*2*pi/8*2*(0:7));
X=fft(x,8)3
subplot(2,1,1); stem(real(x)); %on n'affiche que la partie réelle de x
subplot(2,1,2); stem(abs(X)); %on n'affiche que le module de X
```

<sup>3</sup> On notera que la fonction Matlab utilisée est `fft` (Fast Fourier Transform, cf. 4.2), et non plus `freqz` comme dans le chapitre précédent. En réalité, `freqz` utilise `fft`.

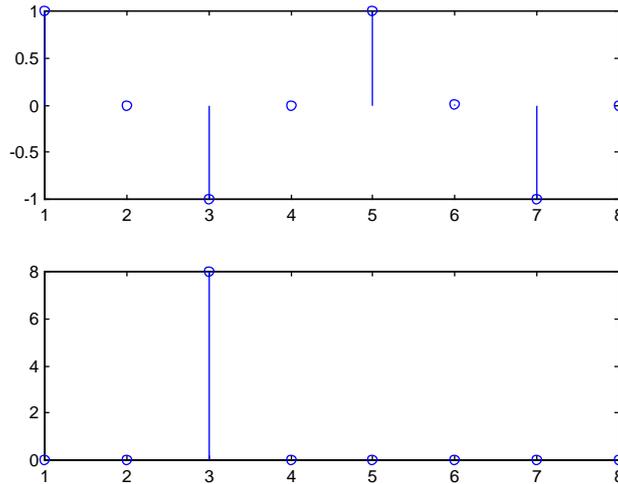


Fig. 4.5 Partie réelle de  $w_2$  pour  $N=8$ <sup>4</sup> et module de sa TFD

#### 4.1.4 Interprétation spectrale

Il a été vu au Chapitre 2 que la TFD d'un signal périodique de période  $T_0 = n_0 T_e$  est égale à  $1/T_0$  fois la TFD de ses  $n_0$  premiers échantillons (c'est-à-dire à la TFD d'une seule période du signal) :

$$F_{T_0}(F) = \sum_{k=-\infty}^{\infty} F_k \delta(\varphi - k\varphi_0) \quad \text{avec} \quad F_k = \frac{1}{T_0} \sum_{n=0}^{n_0-1} f(n) e^{-jn k \varphi_0} \quad \text{et} \quad \varphi_0 = \frac{2\pi}{n_0} \quad (4.8)$$

Par comparaison avec l'équation (4.1), on constate que, si l'on considère que  $n_0 = N$  :

$$X(k) = T_0 F_k \quad (4.9)$$

Autrement dit : les valeurs des coefficients  $\{X_k\} = \{X(0), X(1), \dots, X(N-1)\}$  de la TFD d'une suite  $\{x(n)\} = \{x(0), x(1), \dots, x(N-1)\}$  ne sont rien d'autre que les raies  $F_k$  de la TFD du signal périodique de période  $N$  ayant  $\{x(n)\}$  comme période, corrigées par un facteur multiplicatif  $T_0 = N T_e$  (Fig. 4.6).

<sup>4</sup> Notons que la raie en 3 correspond bien à  $X(2)$  : Matlab commence toujours ses indices à 1.

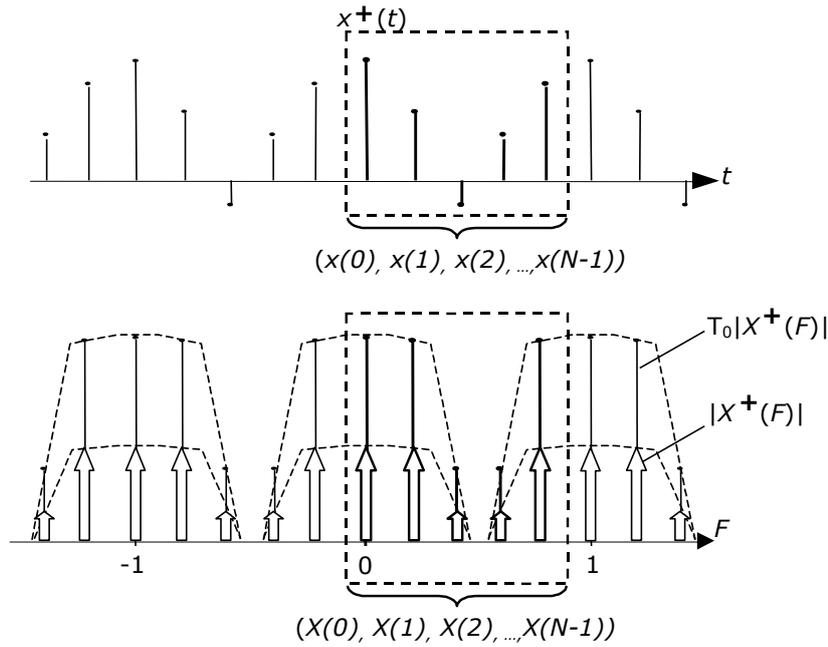


Fig. 4.6 Interprétation de la TFD d'un signal numérique comme la TTFD d'un signal périodique sous-jacent

### 4.1.5 Propriétés

Les propriétés de la TFD sont présentées au Tableau 4.1. Vu l'interprétation spectrale qui vient d'être donnée, elles sont les mêmes que celles de la TTFD, à condition de supposer que le signal d'entrée est périodique de période égale à  $N$  échantillons. Ainsi, il est logique de voir apparaître un *modulo*  $N$  dans la propriété de retard : si on retarde le signal périodique sous-jacent d'une période (c.-à-d. de  $N$  échantillons), on retrouve les échantillons de départ.

(linéarité)	$\sum_i a_i f_i(n)$	$\Leftrightarrow$	$\sum_i a_i F_i(F)$
(retard)	$f((n-n_0) \bmod N)$	$\Leftrightarrow$	$F(k) e^{-j\frac{2\pi}{N}kn_0}$
(convolution <b>circulaire</b> )	$f(n) \otimes g(n)$	$\Leftrightarrow$	$F(k)G(k)$
(produit)	$f(n)g(n)$	$\Leftrightarrow$	$\frac{1}{N} F(k) \otimes G(k)$
(Parseval)	$\sum_{n=0}^{N-1}  f(n) ^2$	$\Leftrightarrow$	$\frac{1}{N} \sum_{n=0}^{N-1}  F(k) ^2$

Tableau 4.1 Propriétés principales de la transformée de Fourier à temps discret

La *convolution circulaire*, qui apparaît dans le tableau précédent, est définie par :

$$f(n) \otimes g(n) = \sum_{i=0}^{N-1} f(i)g((n-i) \bmod N) \quad (4.10)$$

Elle correspond en fait à la convolution entre deux signaux périodiques sous-jacents. Elle n'est donc calculée que sur une période ( $N$  échantillons), et les indices des échantillons sont calculés *modulo*  $N$ .

Enfin, si l'on exprime  $F(k)$  sous la forme :

$$F(k) = A(k)e^{j\varphi(k)} \quad (4.11)$$

on montre facilement (par exemple à partir de l'interprétation géométrique de la TFD ou tout naturellement à partir de son interprétation spectrale) que, dans le cas d'un signal réel, le module  $A(k)$  est pair modulo  $N$  et la phase  $\varphi(k)$  est impaire modulo  $N$ . On dit que  $\{F(k)\}$  est une suite *conjuguée modulo*  $N$  :

$$F(N-k) = F(k)^* \quad (4.12)$$

La parité de  $A(k)$  modulo  $N$  apparaît très bien à la Fig. 4.6.

#### Exemple 4.4

Considérons la TFD de la partie imaginaire (le sinus) d'un des vecteurs de base, par exemple  $\mathbf{w}_2$  pour  $N=8$ . Puisque  $\text{Im}(\mathbf{w}_2) = -j/2 (\mathbf{w}_2 - \mathbf{w}_{-2}) = -j/2 (\mathbf{w}_2 - \mathbf{w}_{N-2})$ , toutes les valeurs  $X(k)$  de la TFD seront nulles, sauf  $X(2)$  et  $X(6)$  qui vaudront  $-4j$  et  $+4j$ , ce qui correspond bien à une suite conjuguée modulo  $N$ . On obtient effectivement ce résultat sous Matlab :

```
x=sin(2*pi/8*2*(0:7));
X=fft(x,8)
subplot(3,1,1); stem(x);
subplot(3,1,2); stem(abs(X));
subplot(3,1,3); stem(angle(X));
```

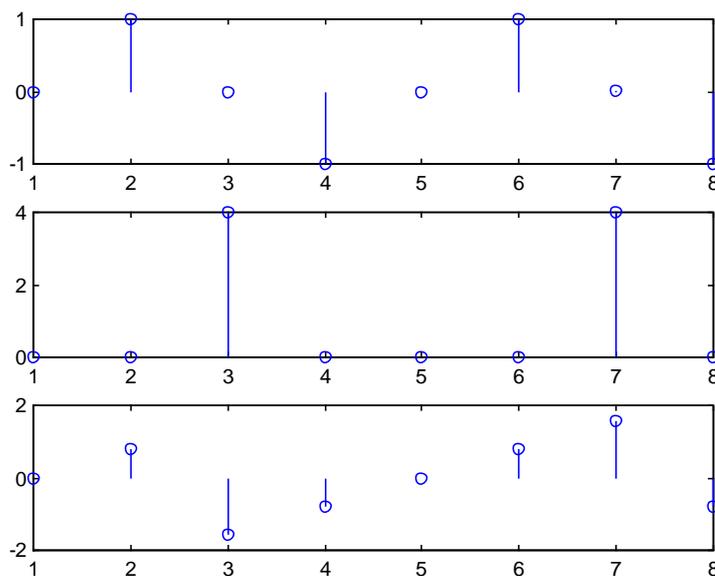


Fig. 4.7 Module et argument de  $\text{Im}(\mathbf{w}_2)$  pour  $N=8$  : suite conjuguée modulo 8

## 4.2 Transformée de Fourier Rapide

En 1965, Cooley et Tukey proposèrent une méthode qui permet de réduire considérablement le temps de calcul de la TFD d'une suite dont le nombre d'échantillons  $N$  est décomposable en facteurs (typiquement, une puissance de 2). Par la suite, de nombreux algorithmes ont été publiés; ils sont connus sous le terme général de *transformation de Fourier rapide* (TFR ; ou FFT : *Fast Fourier Transform*). Tous ces algorithmes sont basés sur un même principe qui consiste à décomposer le calcul de la TFD en plusieurs TFD de longueur plus petite. La mise en œuvre de ce principe conduit à différentes méthodes dont les performances sont comparables. Nous décrivons ici plus particulièrement l'algorithme dit *radix 2 avec entrelacement dans le temps*.

### 4.2.1 FFT radix 2 avec entrelacement dans le temps

Cette méthode, qui exige une séquence dont la longueur est une puissance de 2 ( $N = 2^M$ ), a rendu envisageable le calcul de TFD de plusieurs milliers de points. Le nom de *radix 2* provient du fait que l'on ramène le calcul d'une TFD de  $N$  points à un certain nombre de calculs de TFD de 2 points. L'appellation *entrelacement dans le temps* est lié à la décomposition de la suite  $\{x(n)\}$  en suites plus courtes.

Soit  $\{X(k)\}$  la TFD d'une suite  $\{x(n)\}$  de longueur  $N = 2^M$  :

$$X(k) = \sum_{n=0}^{N-1} x(n).W^{-nk} \quad (k = 0, 1, \dots, N-1) \quad \text{avec} \quad W = e^{-j\frac{2\pi}{N}} \quad (4.13)$$

Soient les deux suites  $a(n)$  et  $b(n)$  de longueur  $N/2$  et leurs TFD  $A(k)$  et  $B(k)$  :

$$a(n) = x(2n) \quad a(n) \Leftrightarrow A(k) \quad n, k = 0, 1, \dots, N/2 - 1 \quad (4.14)$$

$$b(n) = x(2n+1) \quad b(n) \Leftrightarrow B(k)$$

On montre facilement que les  $X(k)$  peuvent être calculés partir des  $A(k)$  et  $B(k)$  :

$$\begin{aligned} X(k) &= \sum_{n=0}^{N/2-1} x(2n).W^{-2nk} + \sum_{n=0}^{N/2-1} x(2n+1).W^{-(2n+1)k} \\ &= A(k) + W^{-k}.B(k) \end{aligned} \quad (4.15)$$

$$\begin{aligned} X\left(\frac{N}{2} + k\right) &= \sum_{n=0}^{N/2-1} x(2n).W^{-2n(k+N/2)} + \sum_{n=0}^{N/2-1} x(2n+1).W^{-(2n+1)(k+N/2)} \\ &= A(k) - W^{-k}.B(k) \end{aligned}$$

au moyen de  $N/2$  modules élémentaires appelés, connus dans la littérature anglo-saxonne sous le nom de *butterfly* (*croisillon*) et définis par les relations (4.15) dans lesquelles  $W^{-k}$  est appelé *facteur de rotation*. Ces relations sont illustrées la Fig. 4.8 dans laquelle le cercle symbolise la TFD d'une suite de longueur 2 (addition et soustraction des deux éléments).

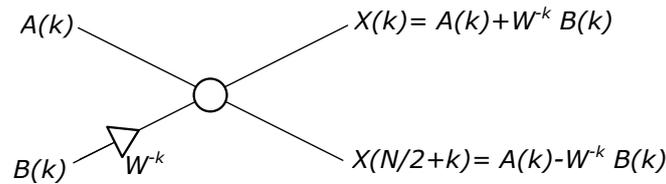


Fig. 4.8 Elément de base pour le calcul de la FFT (croisillon)

Le problème initial est donc ramené à deux sous problèmes (Fig. 4.9) : le calcul de la TFD des deux suites de  $N/2$  points  $\{a(n)\}$  et  $\{b(n)\}$ . Ces deux problèmes sont eux-mêmes divisés en problèmes élémentaires de  $N/4$  points (Fig. 4.10), et ainsi de suite.

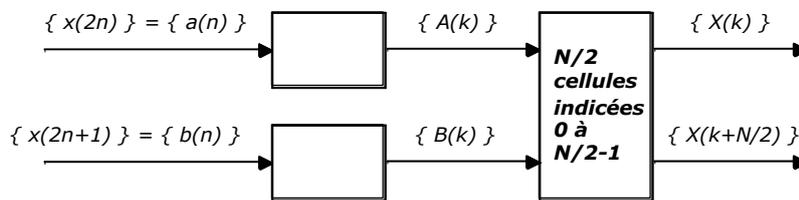


Fig. 4.9 Décomposition en deux sous problèmes de  $N/2$  points

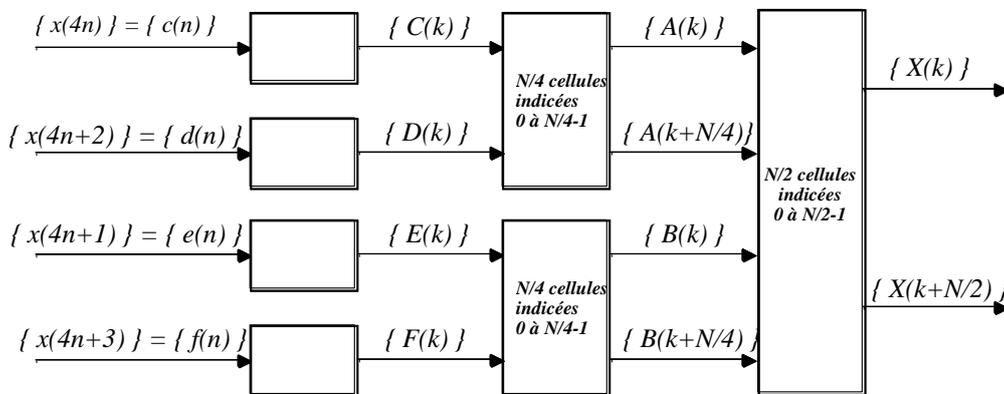


Fig. 4.10 Décomposition en quatre sous problèmes de  $N/4$  points

Comme  $N$  est une puissance de 2, on peut itérer le procédé; ainsi pour  $N=8$ , on obtient le graphe de la Fig. 4.11. D'une manière générale, pour  $N = 2^M$ , le calcul s'effectue en  $M$  étages et chaque étage comprend  $N/2$  modules élémentaires; il faut donc au total:

$N \log_2(N)$  additions(ou soustractions) complexes

$\frac{N}{2} \log_2(N)$  multiplications complexes<sup>5</sup>

<sup>5</sup> Certaines de ces multiplications sont mêmes triviales (celles par 1 et par  $j$ ).

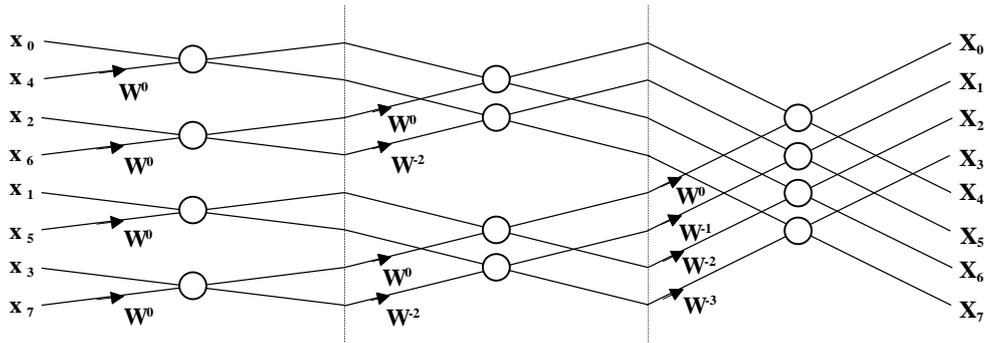


Fig. 4.11 FFT d'une suite de 8 points

On retiendra que la complexité algorithmique de la FFT est de l'ordre de  $O(N \log_2(N))$ , à comparer au  $O(N^2)$  de la TFD. Le gain, de l'ordre de  $O(N/\log_2(N))$ , est énorme si  $N$  est suffisamment grand. Ainsi, pour  $N=1024$ , le gain en charge (et donc en temps) de calcul vaut 100.

### 4.2.2 Cas particulier des suites réelles

Lorsque la suite d'entrée  $\{x(n)\}$  est réelle, on sait que sa TFD est une suite conjuguée paire modulo  $N$  ( $X(N-k) = X(k)^*$ ). Il est donc inutile de calculer les valeurs de  $X(k)$  pour  $k=N/2+1, \dots, N-1$  : la TFD est entièrement définie par deux nombres réels ( $X(0)$  et  $X(N/2)$ ) et  $(N/2)-1$  nombres complexes ( $X(k)$   $k=1, \dots, N/2-1$ ). Nous allons voir qu'on peut effectivement éviter ce calcul inutile en préparant correctement les données en entrée.

Il est clair que si on disposait des TFD  $A(k)$  et  $B(k)$  des deux suites  $\{a(n)\}$  et  $\{b(n)\}$  de longueur  $N/2$  définies à la section précédente, on pourrait calculer directement les seules valeurs de  $X(k)$  qui nous intéressent, puisque :

$$\begin{aligned} X(0) &= A(0) + B(0) \\ X(N/2) &= A(0) - B(0) \\ X(k) &= A(k) + W^{-k} \cdot B(k) \quad k = 1, \dots, N/2 - 1 \end{aligned} \tag{4.16}$$

Or on peut obtenir  $A(k)$  et  $B(k)$  ( $k=0, 1, \dots, N/2-1$ ) en calculant la TFD d'une suite de  $N/2$  valeurs complexes  $\{y(n)\}$  définie par :

$$y(n) = a(n) + jb(n) \quad k = 0, 1, \dots, N/2 - 1 \tag{4.17}$$

En effet, comme les suites  $\{a(n)\}$  et  $\{b(n)\}$  sont réelles, leurs TFD sont des suite conjuguées paires modulo  $N/2$  :

$$\begin{aligned} A(k) &= A^*(N/2 - k) \\ B(k) &= B^*(N/2 - k) \end{aligned} \quad (k = 0, 1, \dots, N/2 - 1) \tag{4.18}$$

Il vient donc :

$$\begin{aligned} Y(k) &= A(k) + jB(k) \\ Y(N/2 - k) &= A^*(k) + jB^*(k) \end{aligned} \quad (k = 0, 1, \dots, N/2 - 1) \tag{4.19}$$

D'où l'on extrait facilement :

$$\begin{aligned}
 A(k) &= \frac{1}{2} (Y(k) + Y^*(N/2 - k)) \\
 B(k) &= -\frac{j}{2} (Y(k) - Y^*(N/2 - k))
 \end{aligned}
 \quad (k = 0, 1, \dots, N/2 - 1) \quad (4.20)$$

Le passage par la TFD des  $N/2$  valeurs complexes de  $\{y(n)\}$  permet donc de calculer directement les  $X(k)$ . De cette façon, on réduit le volume des mémoires d'un facteur 2 et on diminue le temps de calcul (d'environ 40 % pour  $N=1024$ ).

### 4.3 Observation spectrale

Le principal usage de la TFD est lié à la mise en évidence des caractéristiques spectrales des signaux, le plus souvent analogiques. Nous allons voir qu'il est en effet possible d'estimer la transformée de Fourier d'un signal analogique quelconque (périodique ou pas) à l'aide de la TFD.

Rappelons qu'au chapitre 3 nous avons défini la TTFD d'une suite d'échantillons résultant de l'échantillonnage d'une fonction  $x(t)$  comme la transformée de Fourier de la fonction impulsionnelle  $x^+(t)$  obtenue par multiplication de  $x(t)$  avec un train d'impulsions de Dirac illimité dans le temps.

Nous pouvons à présent considérer la TFD d'une séquence  $(x(0), x(1), \dots, x(N-1))$  comme la transformée de Fourier de la fonction impulsionnelle  $x_N^+(t)$  obtenue par multiplication de  $x(t)$  avec un train de  $N$  impulsions de Dirac, transformée dont on n'a gardé que les  $N$  valeurs correspondant aux fréquences  $f=0, F_e/N, 2F_e/N, \dots, (N-1)F_e/N$ .

Il nous reste à examiner l'effet de cette troncation dans le temps et de cet échantillonnage en fréquence.

Considérons tout d'abord le train de  $N$  impulsions de Dirac (signal fenêtre rectangulaire numérique)  $w^+(t)$  valant 1 pour  $t=0, T_e, \dots, (N-1)T_e$  et 0 partout ailleurs (Fig. 4.13.c). La TF  $W^+(f)$  de ce signal est donnée par la TTFD de la suite d'échantillons  $\{w(n)\}$  correspondants :

$$\begin{aligned}
 W^+(F) &= \sum_{n=-\infty}^{\infty} w(n) e^{-jn\varphi} \\
 &= 1 + e^{-j\varphi} + e^{-j2\varphi} + \dots + e^{-j(N-1)\varphi} \\
 &= \frac{1 - e^{-jN\varphi}}{1 - e^{-j\varphi}} \\
 &= \left( \frac{e^{-jN/2\varphi}}{e^{-j1/2\varphi}} \right) \frac{\sin(N/2\varphi)}{\sin(1/2\varphi)}
 \end{aligned} \quad (4.21)$$

Cette fonction a l'allure d'une fonction *sinc* **périodique** ; elle vaut  $N$  en  $F=0$  et s'annule en  $F=1/N, 2/N, \dots, (N-1)/N$  (Fig. 4.12).

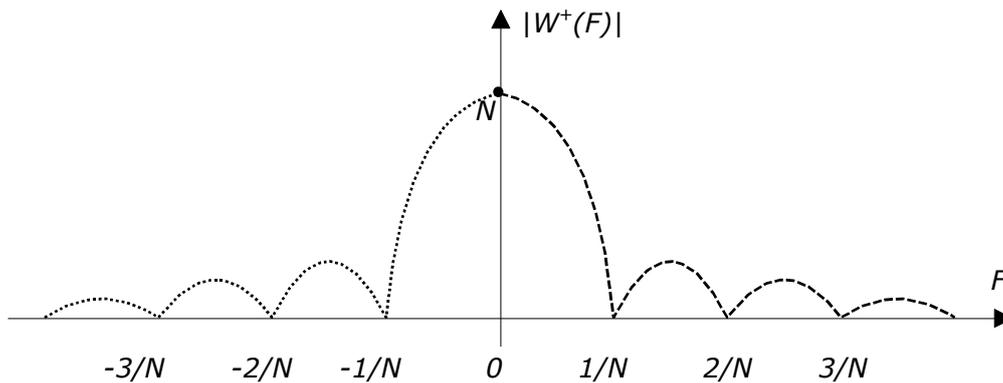


Fig. 4.12 Zoom sur la TTFD de la fenêtre rectangulaire numérique de largeur  $N$

La TF  $X_N^+(f)$  de  $x_N^+(t)$  correspond donc à la convolution des TF respectives de  $x(t)$  et de  $w^+(t)$  (Fig. 4.13.d)

La TFD de  $[x(0), x(1), \dots, x(N-1)]$  n'est alors rien d'autre que la séquence des valeurs de  $X_N^+(f)$  pour  $f = \{0, F_e/N, 2 F_e/N, \dots, (N-1) F_e/N\}$ .

### 4.3.1 Signal périodique

On sait que la TTFD du signal  $x_{T0}(n)$  résultant de l'échantillonnage d'un signal analogique périodique  $x_{T0}(t)$  avec un pas d'échantillonnage  $T_e$  est un spectre de raies périodique (Fig. 4.13.a et b).

Nous pouvons maintenant utiliser (4.21) pour établir le lien qui unit la transformée de Fourier d'un signal périodique analogique  $x_{T0}(t)$  à la TFD de la suite d'échantillons correspondant  $x_{T0}(n)$  ( $n=0, \dots, N-1$ ).

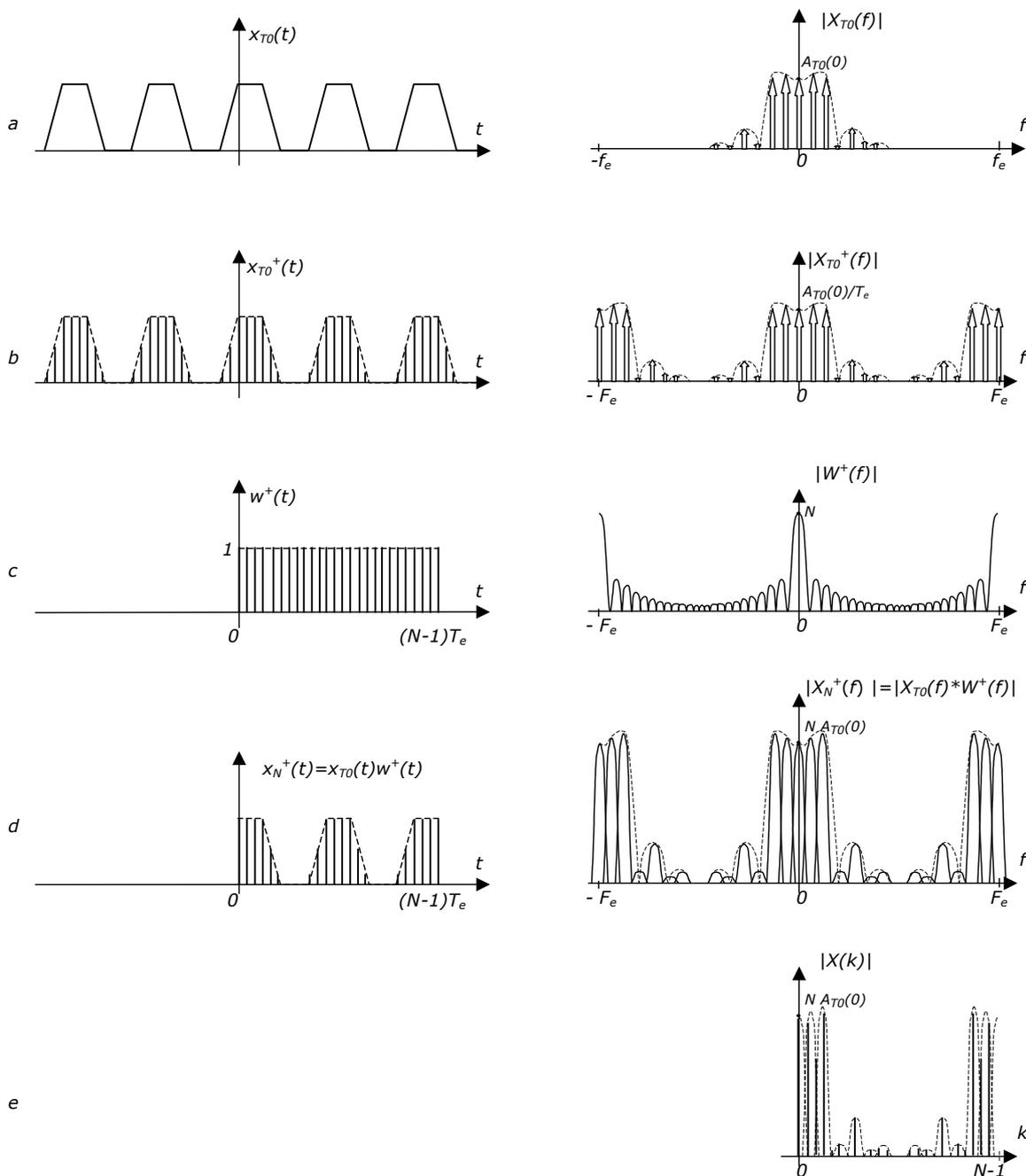


Fig. 4.13 Estimation de la transformée de Fourier d'un signal périodique par TFD : TF (a) ; TTFD (b) ; TFD (e)

En effet si, plutôt que de multiplier  $x_{T0}(t)$  par un train infini d'impulsions de Dirac, on multiplie  $x_{T0}(t)$  par  $w^+(t)$ , on constate que le lobe principal de la TF de la fenêtre  $w^+(t)$ , d'amplitude égale à  $N$ , vient se positionner au droit de chacune des raies du spectre de départ.

Dans le cas où  $F_e/N \ll f_0$  (c'est-à-dire dans le cas où  $N \gg T_0/T_e$ , ce qui revient à dire que les  $N$  échantillons retenus couvrent plusieurs périodes du signal de départ; cas de la Fig. 4.13.c), les copies du lobe principal ne se recouvrent pas trop, et leur somme laisse encore deviner les raies de départ, multipliées par  $N$

(Fig. 4.13.d). Dans le cas contraire, la superposition des lobes ne permet plus de distinguer les raies.

La TFD donne alors les valeurs de cette TTFD pour  $F=\{0, 1/N, 2/N, \dots, (N-1)/N\}$  (Fig. 4.13.e).

### Exemple 4.5

Affichons la TFD sur  $N=16$  points d'une exponentielle imaginaire de fréquence normalisée  $F_0=4/16$  et répétons le calcul pour  $F_0=4.1/16$ ,  $4.5/16$ , et  $4.9/16$  (Fig. 4.14). Profitons-en pour afficher la fréquence normalisée en abscisse.

```
x=exp(j*2*pi*4/16*(0:15));
X=fft(x);
subplot(4,1,1); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
x=exp(j*2*pi*4.1/16*(0:15));
X=fft(x);
subplot(4,1,2); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
x=exp(j*2*pi*4.5/16*(0:15));
X=fft(x);
subplot(4,1,3); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
x=exp(j*2*pi*4.9/16*(0:15));
X=fft(x);
subplot(4,1,4); stem((0:15)/16,abs(X)); %on n'affiche que le module de X
xlabel('fréquence normalisée');
```

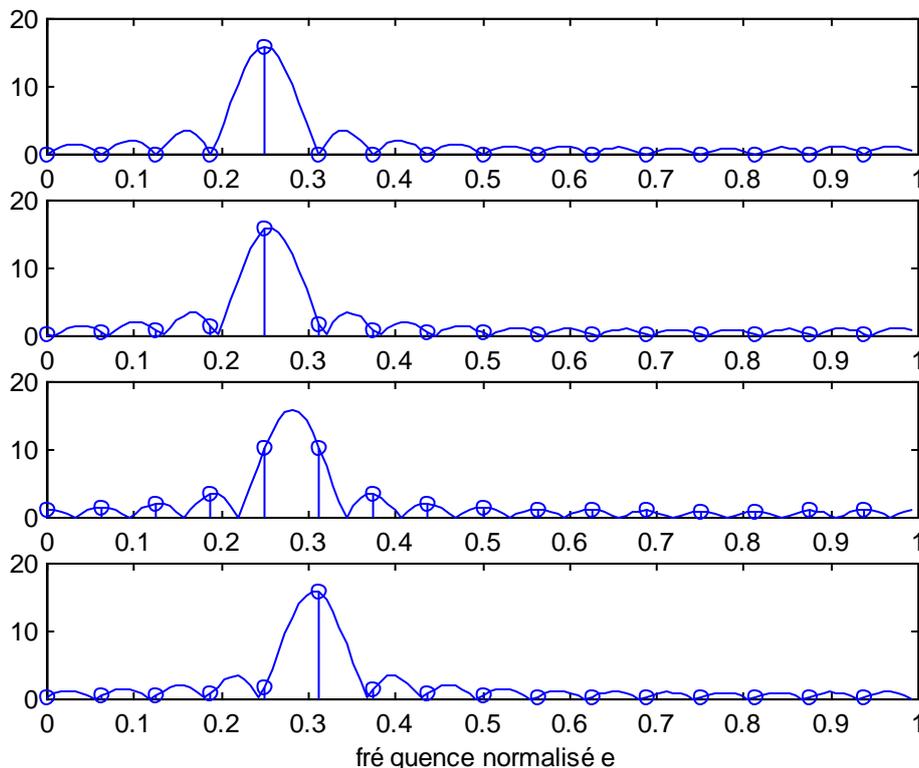
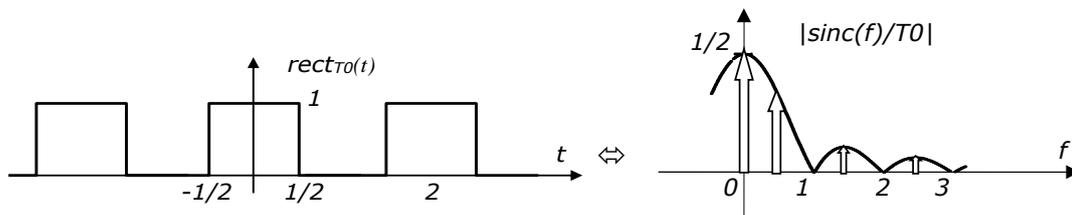


Fig. 4.14 Module de la TFD ( $N=16$ ) d'exponentielles imaginaires de fréquences normalisées  $F_0=4/16$ ,  $4.1/16$ ,  $4.5/16$ , et  $4.9/16$  (la fonction *sinc* périodique a été ajoutée en superposition à titre purement didactique)

On a superposé aux graphiques la TFD de la fenêtre rectangulaire à 16 échantillons, pour mieux comprendre ce qui se passe. On constate que la raie de valeur unitaire attendue se présente sous la forme d'une raie unique de valeur 16 pour  $F_0=4/16$  (multiple de  $1/N$ ). Lorsque la fréquence normalisée du signal n'est pas un multiple de  $1/N$ , la TFD de la fenêtre rectangulaire n'est plus centrée sur une des fréquences normalisées correspondant aux valeurs de la TFD, et plusieurs raies apparaissent. Le cas le pire est celui de  $4.5/16$ , où les deux raies ont la même valeur et où les lobes secondaires de  $W(F)$  sont échantillonnés par la TFD.

**Exemple 4.6 – signal rectangulaire périodique**

Rendons le signal rectangulaire  $rect(t)$  périodique de période égale à 2s. Sa transformée de Fourier fera apparaître des raies en  $f=0, 1/2, 1, 3/2, etc.$ , de valeur  $sinc(f)/2$ .

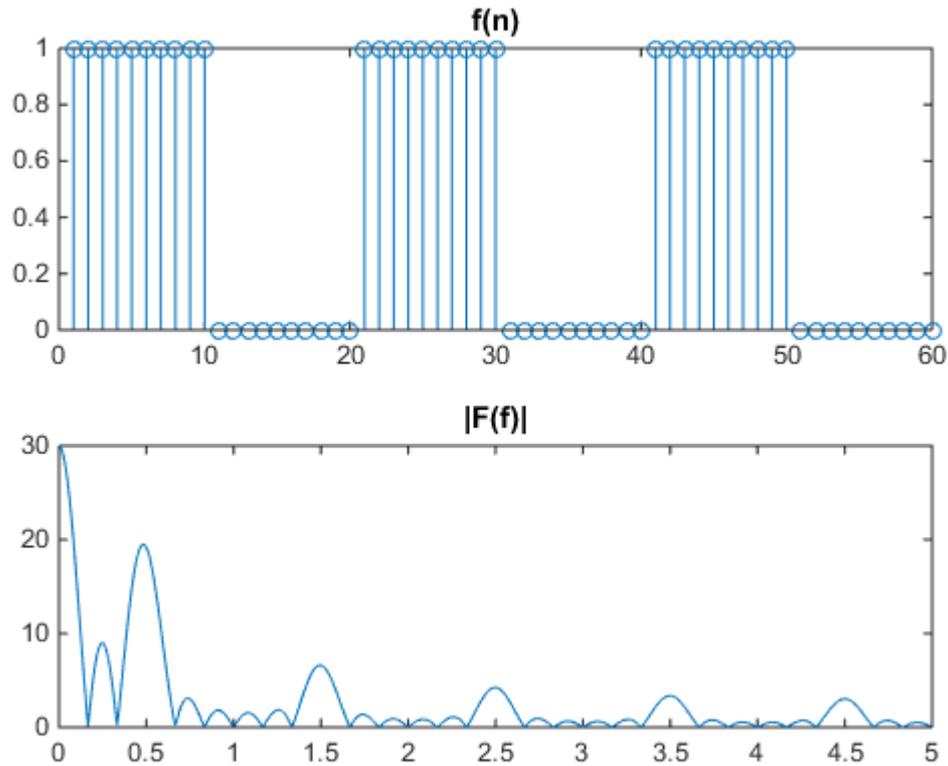


On constate que la transformée fait apparaître une raie en  $f=0$  (composante continue= $1/2$ ), puis uniquement des harmoniques  $F^k$  impaires, d'amplitudes décroissant en  $1/k$ .

```

signal=[rectwin(10) ; zeros(10,1)]; % rectangle width : 1 s = 10 samples
signal = [signal; signal; signal]
[fourier,frequencies]=freqz(signal) % computes the FT in f=[0,Fe/2]

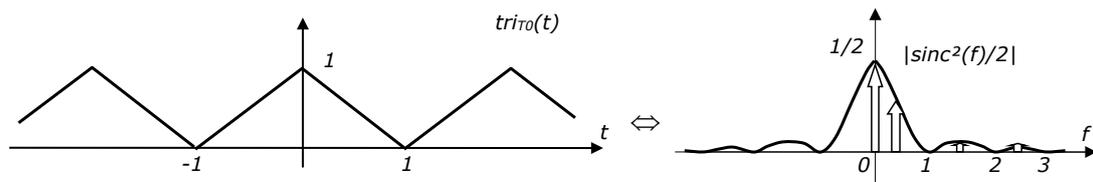
subplot(211); stem(signal); title('f(n)')
Fe=10;
subplot(212); plot(frequencies/pi*(Fe/2),abs(fourier)); title('|F(f)|');
    
```



On retrouve bien ici les raies attendues, avec une amplitude multipliée par le nombre  $N$  d'échantillons retenus (ici 60).

### Exemple 4.7 – signal triangulaire périodique

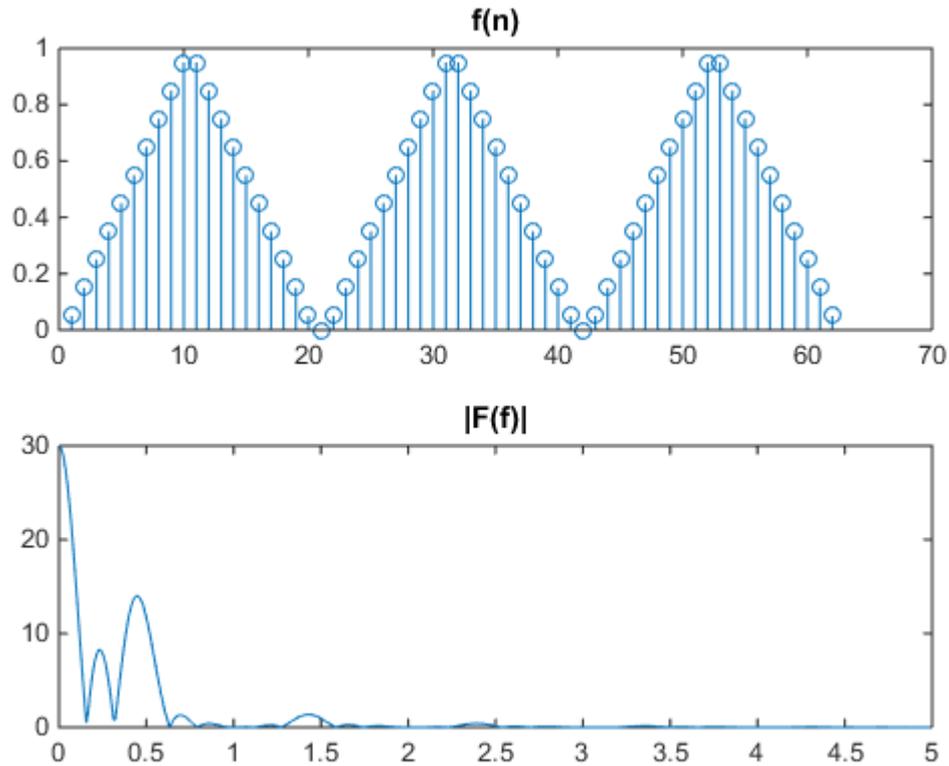
Rendons le signal rectangulaire  $\text{tri}(t)$  périodique de période égale à 2s. Sa transformée de Fourier fera apparaître des raies en  $f=0, 1/2, 1, 3/2, \text{etc.}$ , de valeur  $\text{sinc}^2(f)/2$ .



On constate que la transformée fait apparaître une raie en  $f=0$  (composante continue= $1/2$ ), puis uniquement des harmoniques  $F^k$  impaires, d'amplitudes décroissant en  $1/k$ .

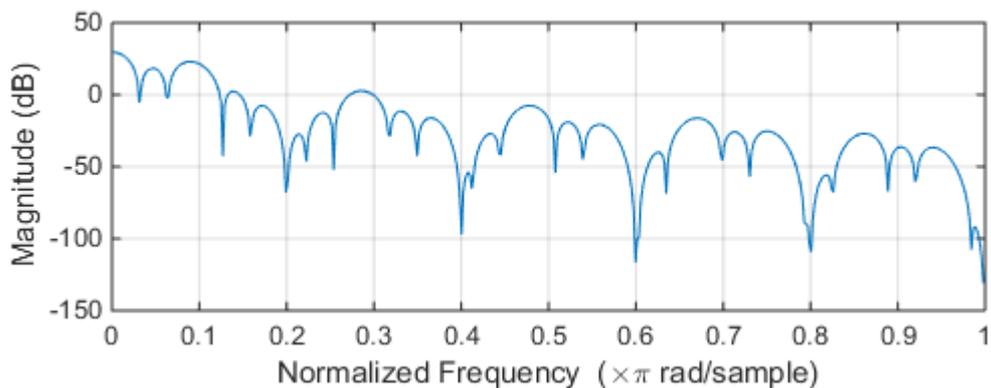
```
signal=[triang(20)]; % triangle width : 2 s = 20 samples
signal = [signal; signal; signal]
[fourier,frequencies]=freqz(signal) % computes the FT in f=[0,Fe/2]
```

```
subplot(211); stem(signal); title('f(n)')
Fe=10;
subplot(212); plot(frequencies/pi*(Fe/2),abs(fourier)); title('|F(f)|');
```



Comme les amplitudes des raies sont petites, on peut avantageusement afficher le module de la transformée de Fourier en dB. C'est ce que fait par défaut la fonction `freqz`, sur un axe de fréquence normalisé entre 0 et 1.

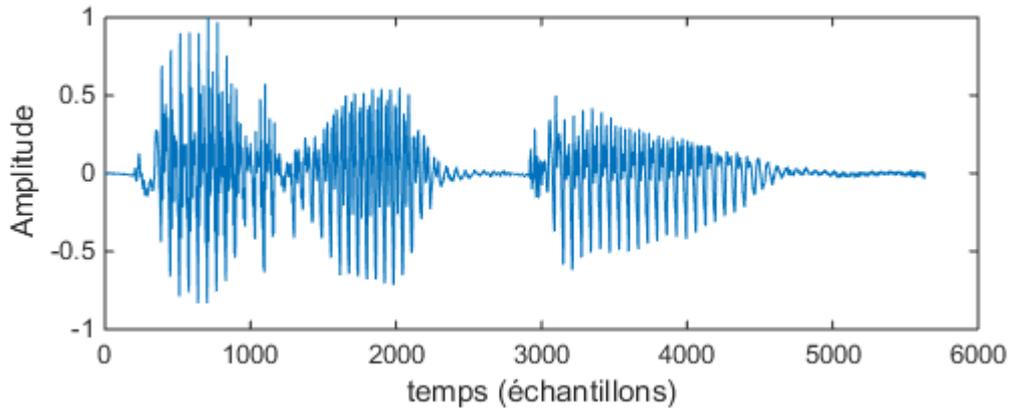
```
freqz(signal)
```



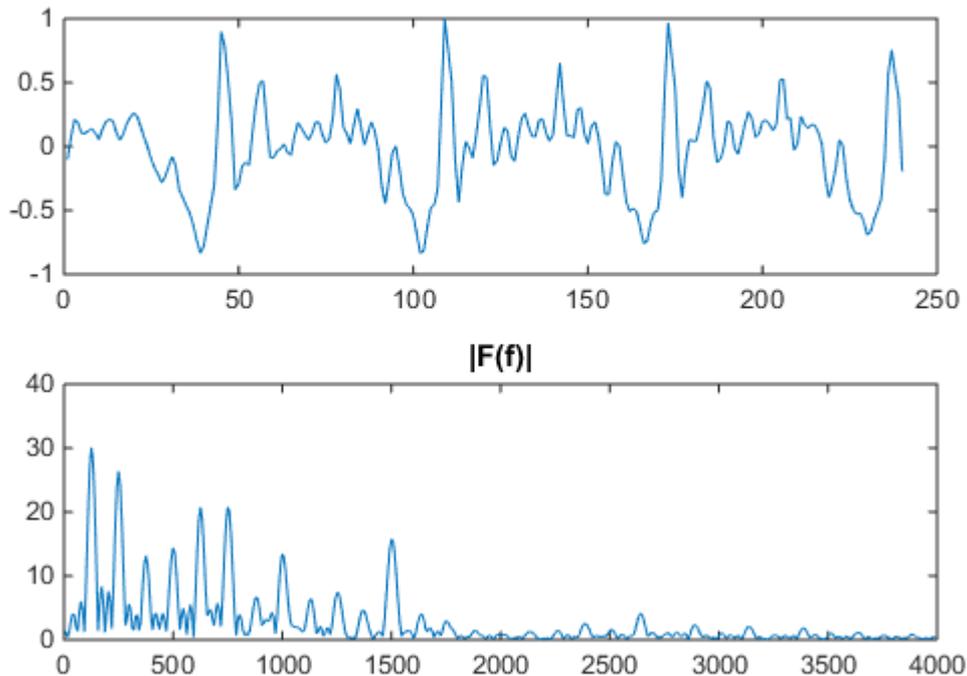
### Exemple 4.8

Il est bien pratique de pouvoir estimer rapidement la transformée de Fourier d'un segment de signal de parole. Par exemple ici pour un extrait du « a » de « parenthèse ».

```
signal=audioread('par8.wav');
Fe=8000;
plot(signal)
xlabel('temps (échantillons)'); ylabel('Amplitude');
soundsc(signal,8000);
```



```
segment=signal(600:839);
subplot(211); plot(segment);
[fourier,frequencies]=freqz(segment) % computes the FT in f=[0,Fe/2]
subplot(212); plot(frequencies/pi*(Fe/2),abs(fourier)); title('|F(f)|');
```



On constate que la période du signal (de l'ordre de 65 échantillons) se manifeste bien par une structure spectrale harmonique de fréquence fondamentale  $8000/65=123$  Hz.

### 4.3.2 Signal non périodique

#### Cas où la fenêtre de $N$ échantillons n'a pas d'influence

Si le signal dont on cherche à observer le spectre est de durée finie, et que le nombre  $N$  d'échantillons dont on dispose couvre une plage temporelle supérieure ou égale à cette durée, l'effet de  $N$  est évidemment transparent : la TFD affiche alors  $N$  valeurs de la TTFD originale.

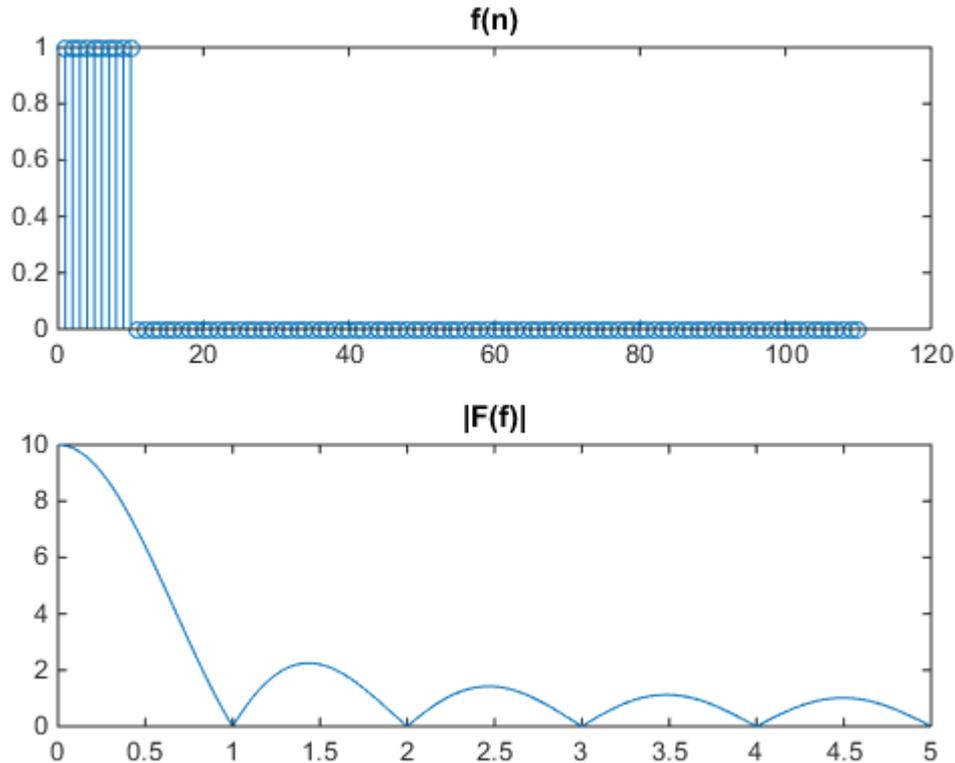
#### Exemple 4.9 – signal rectangulaire

Sous MATLAB, affichons la TFD de  $\text{rect}(t)$ , échantillonné à 10 Hz, et complété par cent échantillons nuls. Notons que la transformée de Fourier de ce signal ne possède pas vraiment de fréquence maximale. On la choisit en pratique suffisamment grande pour que l'effet de miroir spectral puisse être négligé.

```

signal=[rectwin(10) ; zeros(100,1)]; % rectangle width : 1 s = 10 samples
[fourier,frequencies]=freqz(signal) % computes the FT in f=[0,Fe/2]

subplot(211); stem(signal); title('f(n)')
Fe=10;
subplot(212); plot(frequencies/pi*(Fe/2),abs(fourier)); title('|F(f)|');
    
```



MATLAB ne renvoie que les composantes de fréquence positive. On constate que les fréquences renvoyées par `freqz` sont par défaut dans  $[0, \pi]$ .

La forme de la réponse en amplitude correspond bien à la forme attendue ; sa valeur (ici en dB), est 100 fois supérieure à la valeur théorique  $1/10$ , vu le facteur  $1/T_e$  annoncé.

### Exemple 4.10 – Transformée du signal triangulaire

On a vu au chapitre précédent que la convolution de deux signaux rectangulaires donne un signal triangulaire. On en déduit (Fig. 4.15) que :

$$\text{tri}(t) \stackrel{\mathbb{F}}{\Leftrightarrow} \text{sinc}^2(f)$$

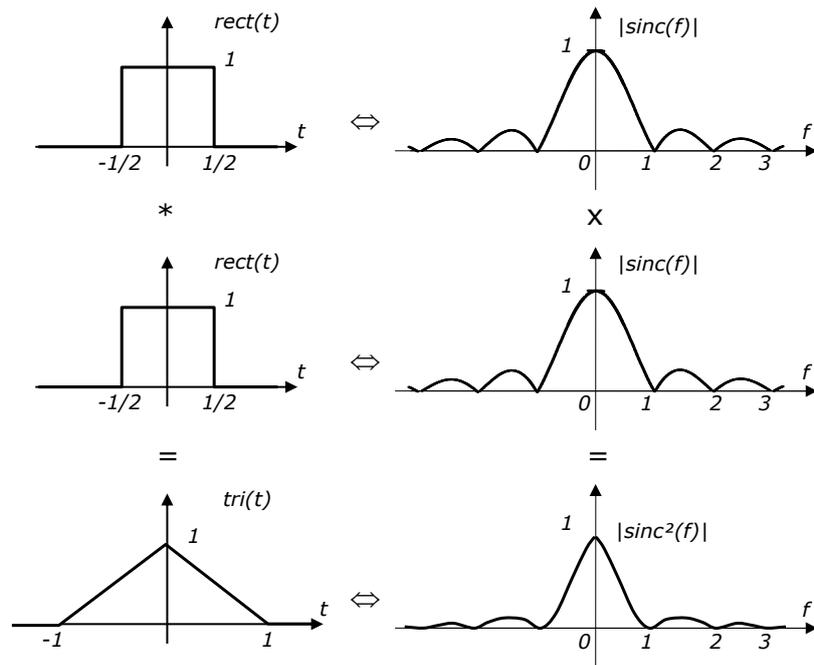
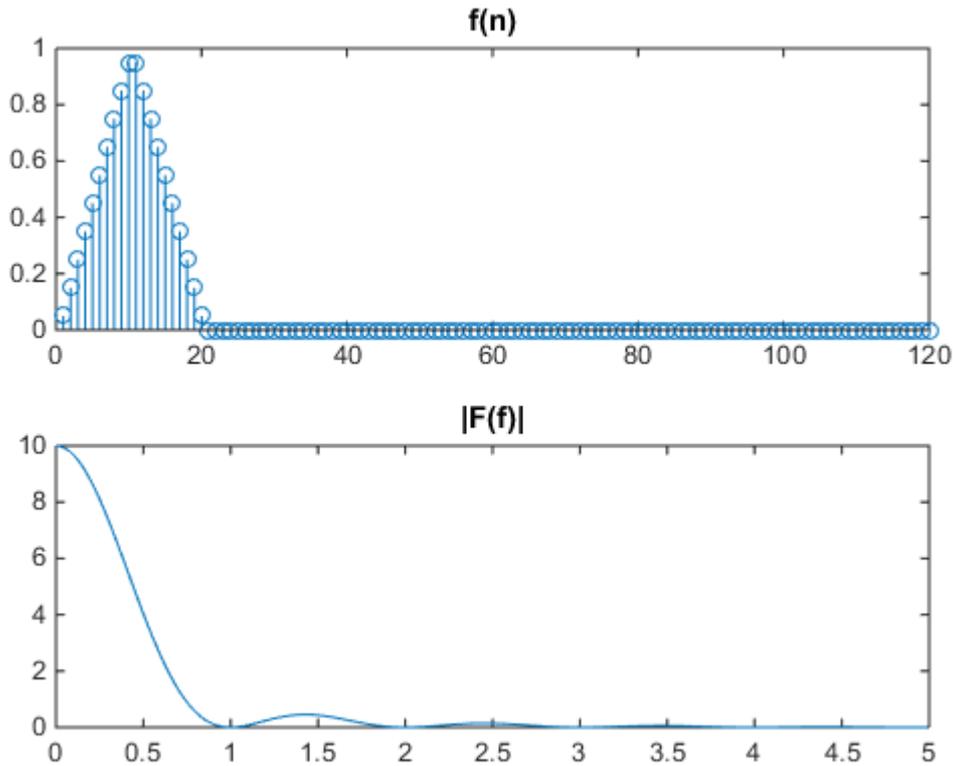


Fig. 4.15 Lien entre transformée de Fourier d'un signal carré et triangulaire

Calculons cette transformée numériquement sur un triangle de largeur 2s, échantillonné à 10 Hz.

```
signal=[triang(20) ; zeros(100,1)]; % triangle width : 2 s = 20 samples
[fourier,frequencies]=freqz(signal) % computes the FT in f=[0,Fe/2]
```

```
subplot(211); stem(signal); title('f(n)')
Fe=10;
subplot(212); plot(frequencies/pi*(Fe/2),abs(fourier)); title('|F(f)|');
```



### Cas où la fenêtre de $N$ échantillons conduit à une troncature du signal dans le temps

Dans le cas où le nombre d'échantillons ne permet pas de couvrir tout le signal, l'interprétation donnée plus haut est également applicable à un signal non périodique  $x(t)$ , la seule différence étant que la transformée de Fourier de départ,  $X(f)$ , est continue (Fig. 4.16).

L'effet de la convolution de  $X(f)$  avec  $W^+(f)$  correspond à une périodification accompagnée d'un filtrage passe-bas de  $X(f)$  (Fig. 4.16.d): les variations brusques de  $X(f)$  sont adoucies, et sa dynamique est limitée par celle de  $W^+(f)$  : on voit parfois apparaître sur  $X_N^+(f)$  les lobes secondaires de  $W^+(f)$  dans les plages de fréquences où  $X(f)$  est très faible.

Le niveau d'amplitude des raies données par la TFD est lié à celui de la transformée de Fourier du signal par la relation :

$$X^+(F=0) = \sum_{n=-\infty}^{\infty} x(n) \quad X_N^+(F=0) = \sum_{n=0}^{N-1} x(n) \quad (4.22)$$

D'où l'on déduit que :

$$X_N^+(0) = kX^+(0) \quad \text{avec } k = \frac{\sum_{n=0}^{N-1} x(n)}{\sum_{n=-\infty}^{\infty} x(n)} \quad (4.23)$$

Il est clair que si la fenêtre est suffisamment longue (c'est-à-dire si  $N$  est suffisamment grand), l'effet de filtrage passe-bas sur  $X(f)$  deviendra négligeable ( $W^+(f)$  tendra vers une impulsion) et l'amplitude de  $X_N^+(f)$  tendra vers celle de

$X^+(f)$  (c'est-à-dire vers celle de la transformée de Fourier recherchée  $X(f)/T_e$ , si l'on suppose que le théorème de Shannon a été respecté).

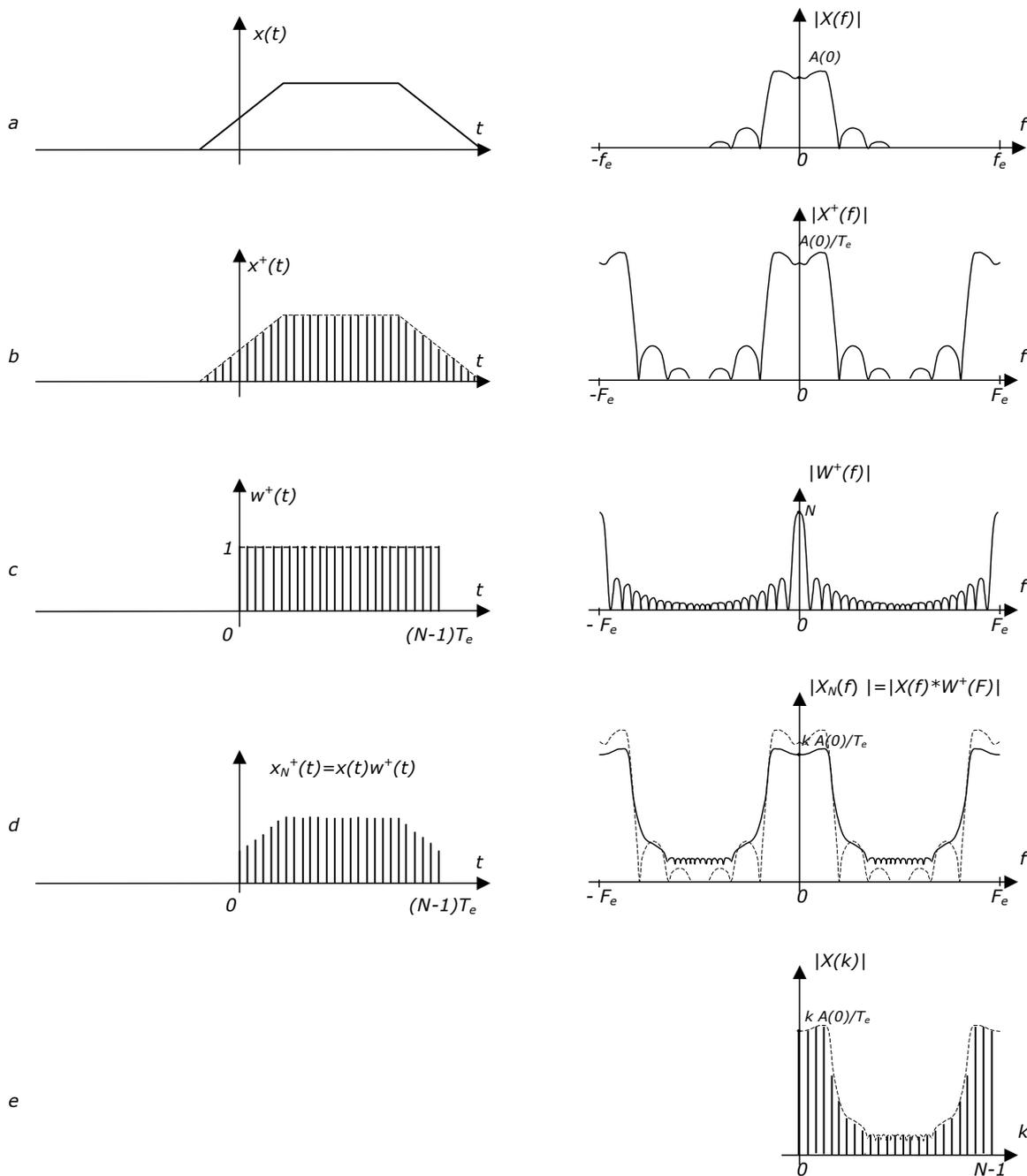
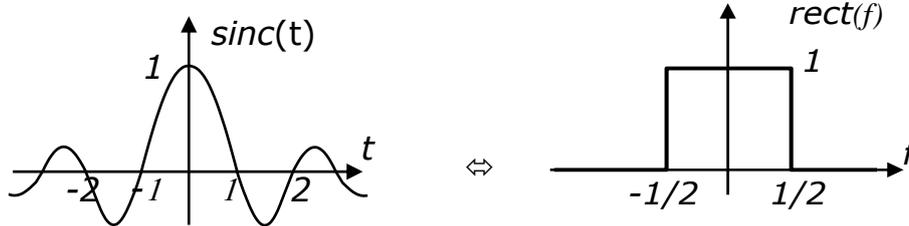


Fig. 4.16 Estimation de la transformée de Fourier d'un signal non périodique par TFD : TF (a) ; TTFD (b) ; TFD (e)

### Exemple 4.11 – Transformée du sinus cardinal

Par application de la propriété de dualité de la transformée de Fourier, il vient :

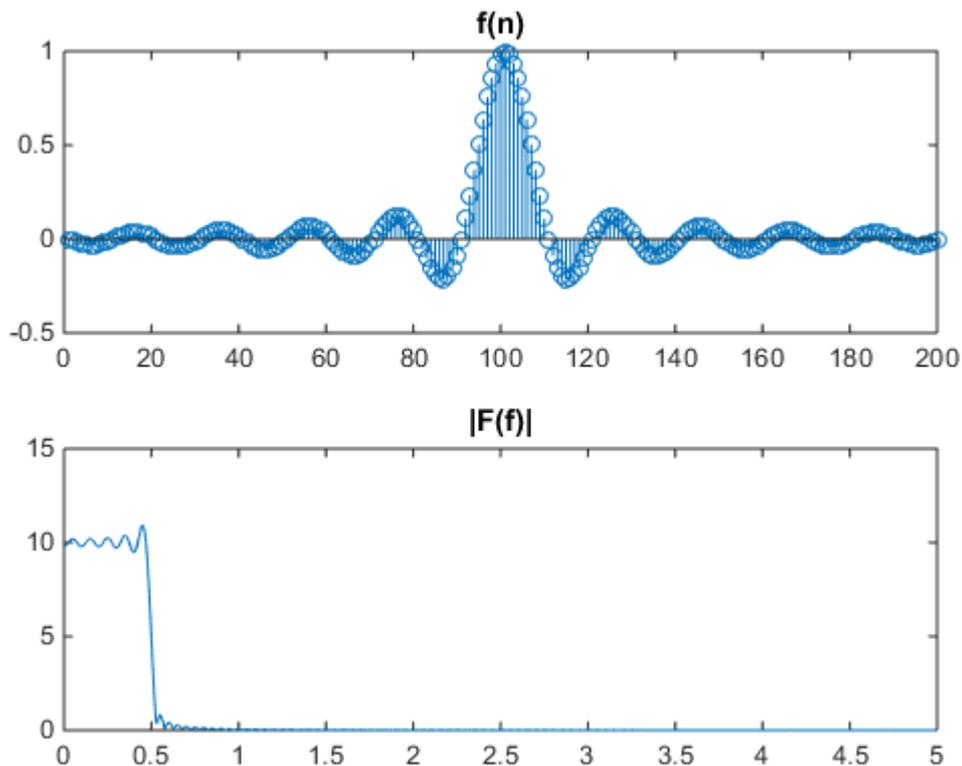
$$\text{sinc}(t) \stackrel{\mathbb{F}}{\Leftrightarrow} \text{rect}(-f) = \text{rect}(f)$$



```

Fe=10 ; t=(-100:99)* 1/Fe;    %Fe = 10 ; 200 samples ; 20 oscillations
signal=sinc(t),
[fourier,frequencies]=freqz(signal) % computes the FT in f=[0,Fe/2]

subplot(211); stem(signal); title('f(n)')
Fe=10;
subplot(212); plot(frequencies/pi*(Fe/2),abs(fourier)); title('|F(f)|');
    
```



**Exemple 4.12**

Soit un SLI analogique du second ordre à pôles complexes conjugués, de fonction de transfert :

$$H(p) = \frac{\rho^2}{p^2 + 2\sigma p + \rho^2} \quad \text{avec } \rho = 2\pi 1000 \quad \sigma = 2\pi 5 \quad (4.24)$$

Il y correspond une réponse en amplitude :

$$A(\omega) = 20 \log |H(j\omega)|$$

$$= 20 \log \rho^2 - 20 \log \sqrt{(\rho^2 - \omega^2)^2 + 4\sigma^2 \omega^2} \quad (4.25)$$

Le diagramme de Bode correspondant est donné à la Fig. 4.17, où l'on a représenté également la réponse affichée par Matlab avec :

```
rho=2*pi*1000; sigma=2*pi*5;
freqs([rho*rho], [1, 2*sigma, rho*rho], logspace(3, 5, 200)) ;
```

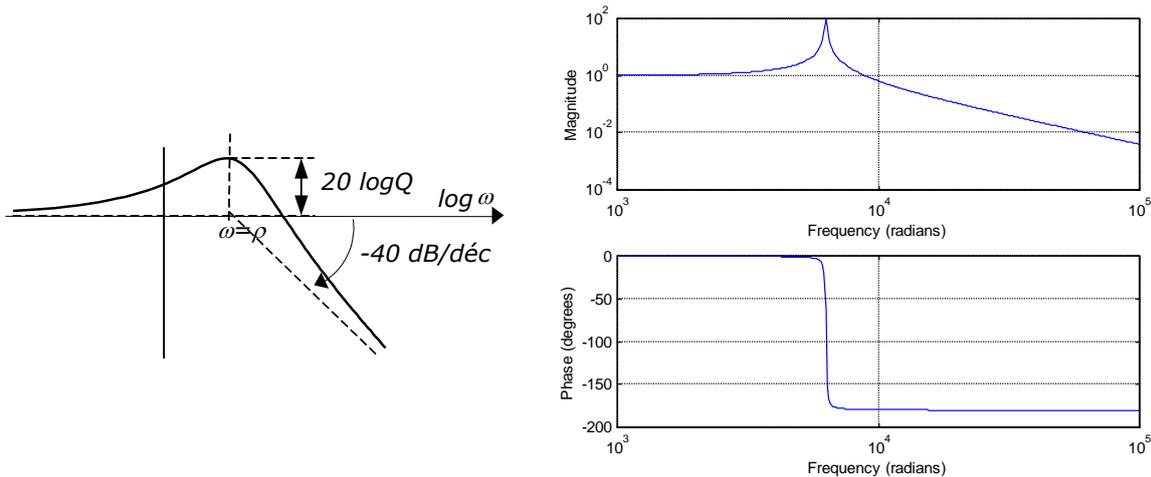


Fig. 4.17 Diagramme de Bode de  $H(p) = \frac{\rho^2}{p^2 + 2\sigma p + \rho^2}$

Sa réponse impulsionnelle est donnée par :

$$h(t) = \frac{\rho^2}{\omega} \exp(-\sigma t) \sin(\omega t) \varepsilon(t) \quad \text{avec } \omega = \sqrt{\rho^2 - \sigma^2} \quad (4.26)$$

Supposons que l'on mesure cette réponse impulsionnelle sans connaître la fonction de transfert de système, et que l'on veuille afficher sa réponse en amplitude  $A(f)$ .

Nous devons pour ce faire fixer deux paramètres : la période d'échantillonnage  $T_e$  et le nombre  $N$  de points du signal à utiliser pour le calcul de la FFT. La Fig. 4.18 montre les résultats obtenus avec  $F_e=10000$  (cette valeur est choisie de façon que  $A(F_e) = A(F_e) - 40$  dB) et différentes valeurs de  $N$ . Les commandes Matlab sont du type :

```
omega=sqrt(rho*rho-sigma*sigma)
N=1024; Te=1/10000;
n=(0:N-1);
h=rho*rho/omega*exp(-sigma*n*Te).*sin(omega*n*Te);
H=fft(h)*Te; % multiplier par Te pour tenir compte de l'échantillonnage
w=n(1:N/2+1)/N*(2*pi/Te); H=H(1:N/2+1); % On ne prend que H(F) pour F=0...1/2
subplot(2,1,1); plot(n*Te,h); title('h(t) pour Fe=10000 et N=256');
subplot(2,1,2); semilogx(w,20*log10(abs(H))); title('A(w) (dB)');
```

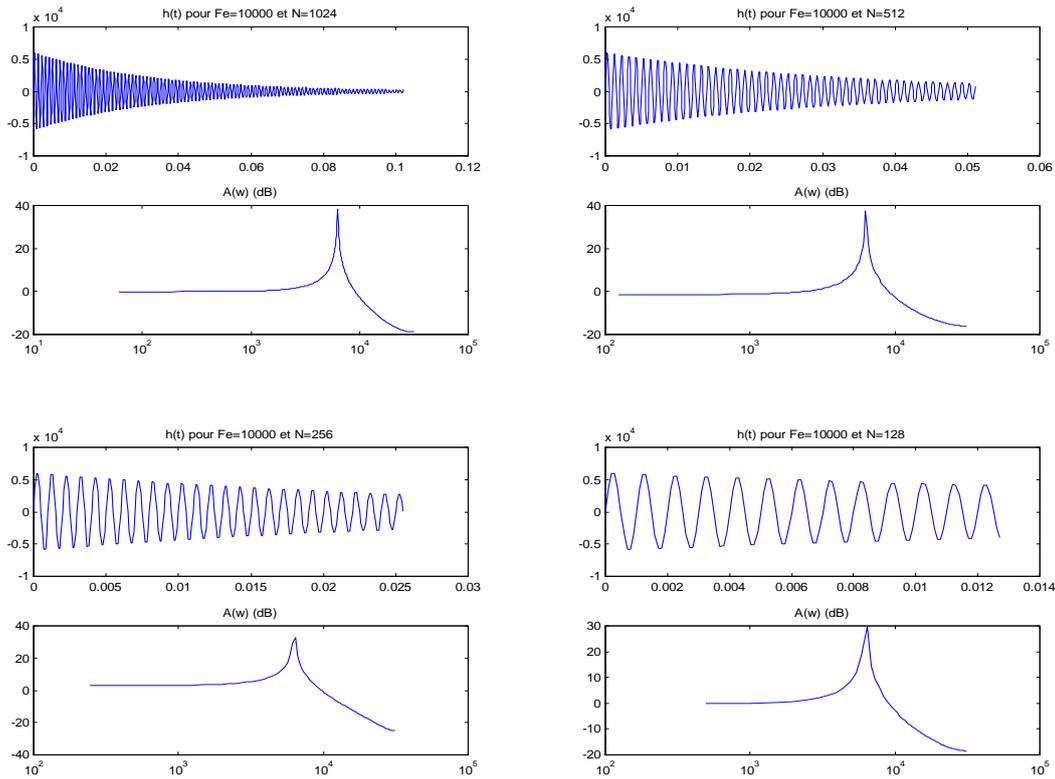


Fig. 4.18 Estimation de la réponse en fréquence d'un SLI analogique d'ordre 2 à partir du module de la TFD de sa réponse impulsionnelle échantillonnée ( $F_e=10000$  ;  $N$  variable)

On constate à la Fig. 4.18 que, pour une fréquence d'échantillonnage suffisamment élevée, la qualité du résultat est fonction du choix de  $N$  : lorsque  $N$  diminue, l'effet de la convolution avec  $W(F)$  se manifeste par un élargissement de la résonance, et une modifications des amplitudes.

Notons qu'une diminution de  $N$  implique aussi une diminution de la résolution fréquentielle : seules  $N/2+1$  valeurs de la TTFD sont affichées par la TFD, entre  $F=0$  et  $1/2$  .

On comprend, au vu de ces figures, qu'il faut être prudent sur le choix de  $N$  lorsqu'on évalue une transformée de Fourier à l'aide de la TFD.

### 4.3.3 Calcul d'une TTFD sur $N_{TFD}$ points en fréquence

On a constaté jusqu'ici que la valeur de  $N$  détermine simultanément sur la largeur du lobe principal de  $W^+(f)$  (et par là sur l'importance de l'effet convolutif associé) et sur la discrétisation fréquentielle réalisée par la TFD. Ces deux effets peuvent en réalité être maîtrisés séparément, en réalisant le calcul de la TTFD du signal fenêtré  $\{x_N(n)\}$  sur un nombre de points en fréquence  $N_{TFD}$  différent de  $N$ .

#### Complétion par des zéros

Si l'on ajoute des échantillons nuls à la suite des  $N$  valeurs ( $x_N(0), x_N(1), \dots, x_N(N-1)$ ), de façon à obtenir une séquence de  $N_{TFD}$  valeurs, on obtient une représentation plus réaliste de  $\{x_N(n)\}$ . Le calcul de la TFD de cette séquence fournit donc un échantillonnage de la TTFD de  $\{x_N(n)\}$ , cette fois sur  $N_{TFD}$  valeurs

en fréquence. La complétion d'une séquence par des zéros fournit donc une meilleure représentation de  $X_N^+(f)$ , sans pour autant augmenter la précision du résultat.

### Exemple 4.13

Calculons la TTFD d'une fenêtre rectangulaire de 16 points, par FFT sur 16, 32, et 128 points :

```
x=ones(1,16);
subplot(6,1,1); stem(x);
subplot(6,1,2); plot((0:15)/16, abs(fft(x,16)));
x=[ones(1,16) zeros(1,16)];
subplot(6,1,3); stem(x);
subplot(6,1,4); plot((0:31)/32, abs(fft(x,32)));
x=[ones(1,16) zeros(1,112)];
subplot(6,1,5); stem(x);
subplot(6,1,6); plot((0:127)/128, abs(fft(x,128)));
```

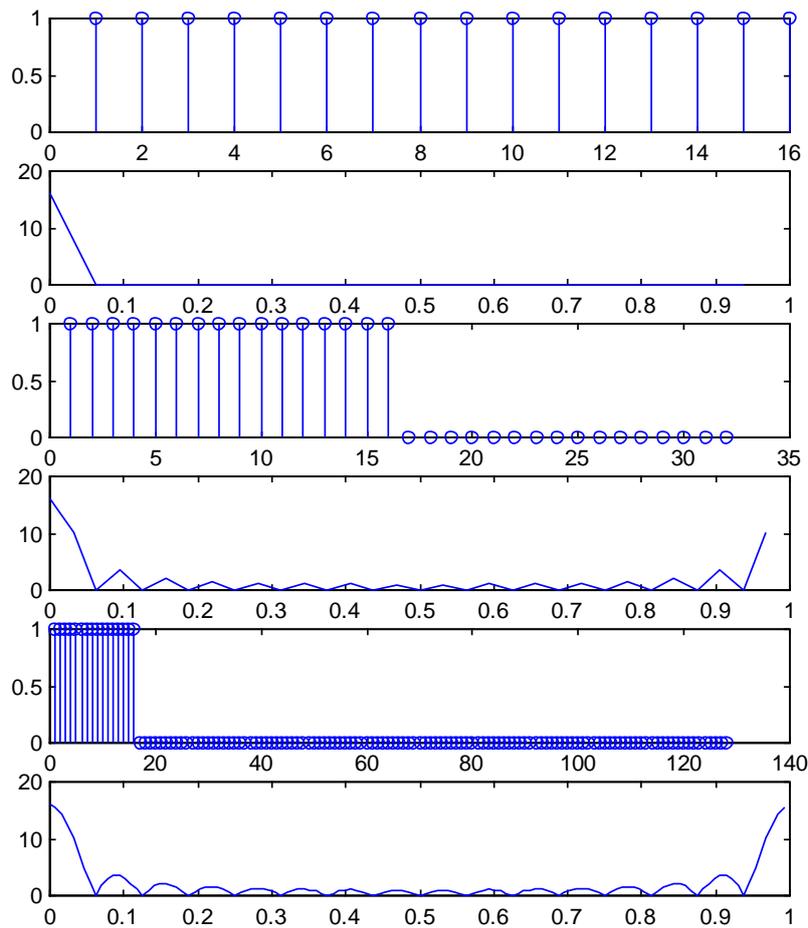


Fig. 4.19 FFT sur 16, 32, et 128 points de 16 échantillons d'une fenêtre rectangulaire.

**Périodification de la séquence d'échantillons**

De façon similaire, il est possible de calculer la TTFD de  $\{x_N(n)\}$  sur un nombre de points  $N_{TFD}$  en fréquence *plus petit que*  $N$ . Il suffit en effet de découper la séquence  $(x_N(0), x_N(1), \dots, x_N(N-1))$  en blocs de  $N_{TFD}$  valeurs, et de sommer les échantillons occupant les mêmes positions dans leurs blocs respectifs. Ceci revient à périodifier la séquence d'entrée avec une période de  $N_{TFD}$  échantillons :

$$x_{N_{TFD}}(n) = \sum_{i=-\infty}^{\infty} x_N(n - iN_{TFD}) \tag{4.27}$$

Il est facile de montrer que la TFD de la séquence  $\{x_{N_{TFD}}(n)\}$  ainsi constituée fournit bien les valeurs de la TTFD de  $\{x_N(n)\}$  sur  $N_{TFD}$  valeurs entre  $F=0$  et 1.

**Exemple 4.14**

Soit une séquence de  $N=8$  échantillons  $(x(0), x(1), \dots, x(7))$ . Constituons une séquence de longueur  $N_{TFD}=4$  par (4.27) :  $(x(0)+x(5), x(1)+x(6), x(2)+x(6), x(3)+x(7))$ . La TFD de cette séquence est donnée géométriquement à la Fig. 4.2. Il est clair que les 4 valeurs obtenues par cette TFD sont aussi 4 des 8 valeurs que l'on aurait obtenues par TFD de  $(x(0), x(1), \dots, x(7))$ .

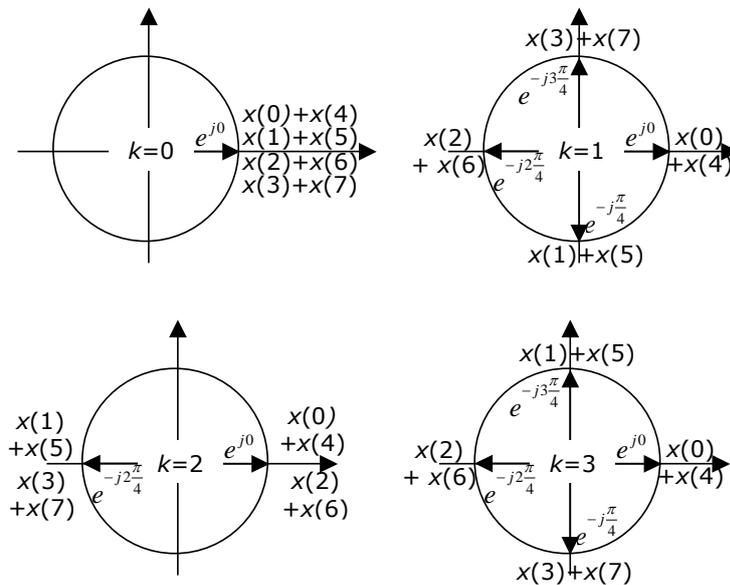


Fig. 4.20 Calcul géométrique de la TFD de  $(x(0)+x(5), x(1)+x(6), x(2)+x(6), x(3)+x(7))$

L'intérêt de cette technique se fait surtout sentir lorsque le signal dont on veut calculer la TTFD possède un grand nombre d'échantillons non nuls et que l'on veut éviter un effet convolutif trop important par utilisation d'une fenêtre trop courte. La calcul sur tous les échantillons impliquerait alors une charge de calcul importante, même pour la FFT.

On retiendra que, de la même façon que la complétion par des zéros n'augmente la résolution fréquentielle sans augmenter la précision de la TTFD sous-jacente,

la périodification diminue la résolution fréquentielle sans affecter la précision de la TFTD sous-jacente.

### Exemple 4.15

Calculons la TFTD d'une fenêtre rectangulaire de 16 points, par FFT sur 16, 8, et 4 points :

```
x=ones(1,16);
subplot(6,1,1); stem(x);
subplot(6,1,2); stem((0:15)/16, abs(fft(x,16)));
x=[ones(1,8)+ones(1,8)];
subplot(6,1,3); stem(x);
subplot(6,1,4); stem((0:7)/8, abs(fft(x,8)));
x=[ones(1,4)+ones(1,4)+ones(1,4)+ones(1,4)];
subplot(6,1,5); stem(x);
subplot(6,1,6); stem((0:3)/4, abs(fft(x,4)));
```

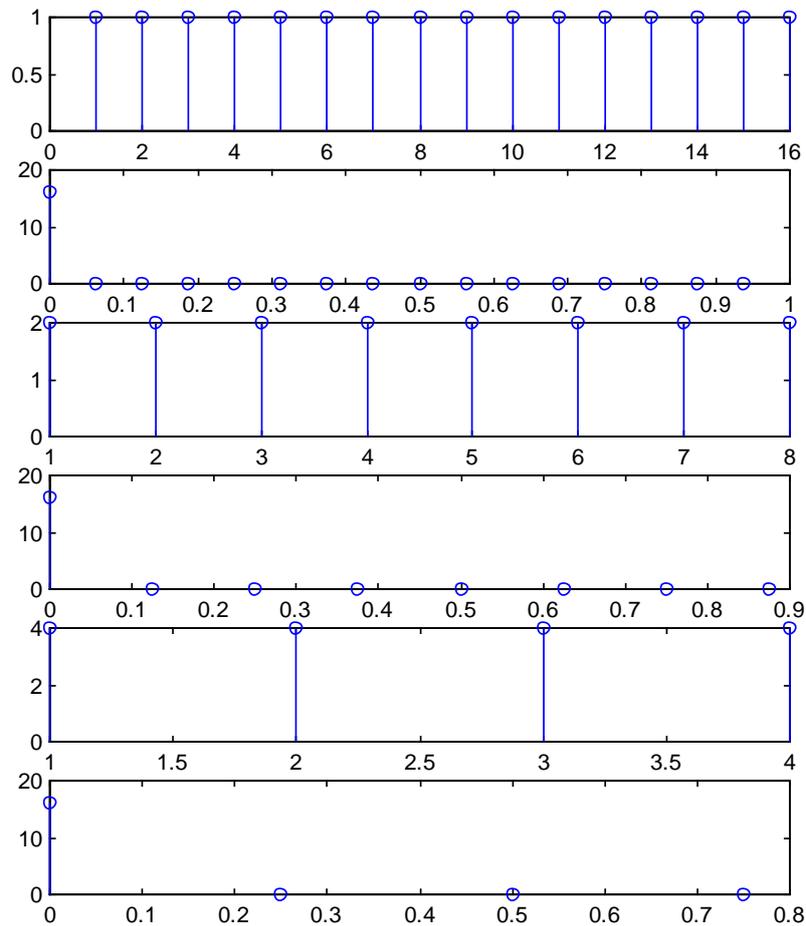


Fig. 4.21 FFT sur 16, 8, et 4 points de 16 échantillons d'une fenêtre rectangulaire.

En résumé, en complétant les  $N$  échantillon initiaux par de 0 ou en périodifiant cette séquence d'échantillons, on peut calculer une TTFD sur un nombre de points  $N_{TFD}$  plus petit ou plus grand que  $N$ . Le pas fréquentiel de la TFD est égale à  $N_{TFD}$  (Fig. 4.22).

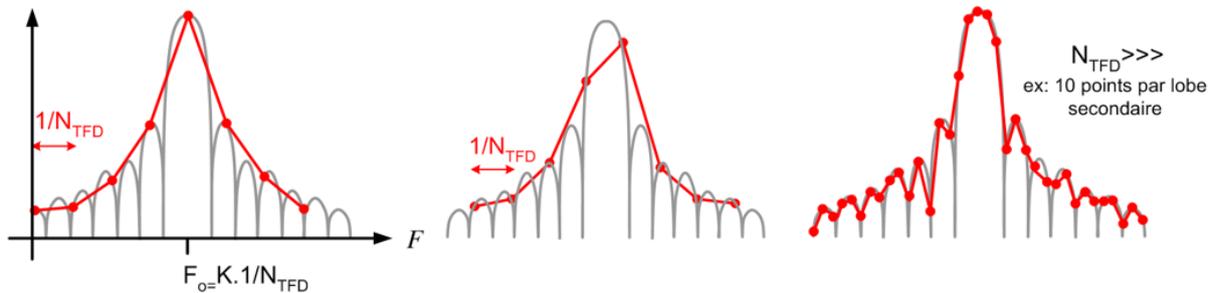


Fig. 4.22 Effet d'une valeur de  $N_{TFD}$  différente de  $N$ . En gris, la TTFD sous-jacente ; en groupe, la TTFD dont les points ont été reliés par des segments de droite.

#### 4.3.4 Pondération par une fenêtre

Nous avons vu que le fait de ne conserver que  $N$  échantillons d'un signal  $\{x(n)\}$  peut être interprété comme la multiplication de ce signal par une fenêtre rectangulaire. La TTFD de cette fenêtre possède un lobe principal de largeur  $1/N$ , et l'amplitude de ses lobes secondaires est à peu près 13 en dessous de celle de son lobe principal. Ceci peut être gênant lorsqu'il s'agit de mesurer l'amplitude des raies d'un signal périodique, si certaines raies sont largement à plus de 13 dB sous la raie d'amplitude maximale.

#### Exemple 4.16

Soit un signal constitué de deux sinusoïdes à 130 et 300 Hz, d'amplitudes 1 et 1/00, échantillonnées à 1000 Hz. On utilise une fenêtre rectangulaire de 64 points et le calcul de la FFT se fait sur 512 points en fréquence :

```
x=sin(2*pi*130*(0 :63)/1000)+ 0.01*sin(2*pi*300*(0 :63)/1000);
subplot(2,1,1); % affichage du signal
plot((0:63)/1000,x);
subplot(2,1,2); % affichage de la TTFD du signal fenêtré
plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
```

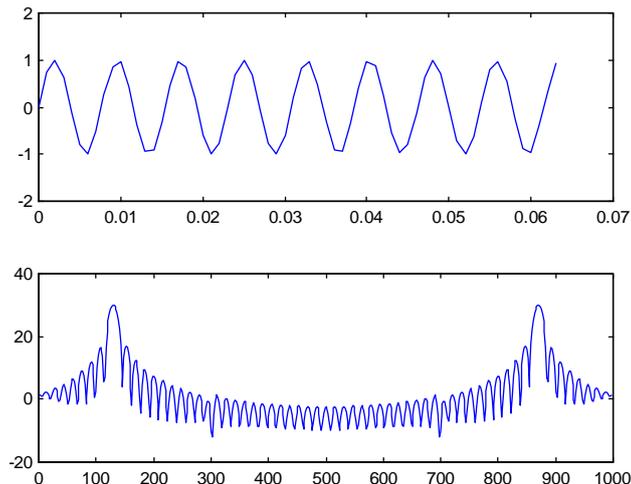


Fig. 4.23 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100

On constate que la raie de la sinusoïde à 300 Hz est couverte par les lobes secondaires de la TFTD de la fenêtre, placée sur la raie à 130 Hz.

On peut alors utiliser sciemment une fenêtre de pondération dont les caractéristiques fréquentielles (demi-largeur du lobe principal, amplitude des lobes secondaires) sont différentes de celles de la fenêtre rectangulaire. Les fenêtres utilisées le plus couramment sont (Fig. 4.24) :

$$\text{Hanning : } w(n) = \begin{cases} 0.5 - 0.5 \cos(2\pi n / N) & \text{si } n = 0 \dots N - 1 \\ 0 & \text{sinon} \end{cases} \quad (4.28)$$

$$\text{Hamming : } w(n) = \begin{cases} 0.54 - 0.46 \cos(2\pi n / N) & \text{si } n = 0 \dots N - 1 \\ 0 & \text{sinon} \end{cases} \quad (4.29)$$

$$\text{Blackman : } w(n) = \begin{cases} 0.42 - 0.5 \cos(2\pi n / N) + 0.08 \cos(4\pi n / N) & \text{si } n = 0 \dots N - 1 \\ 0 & \text{sinon} \end{cases} \quad (4.30)$$

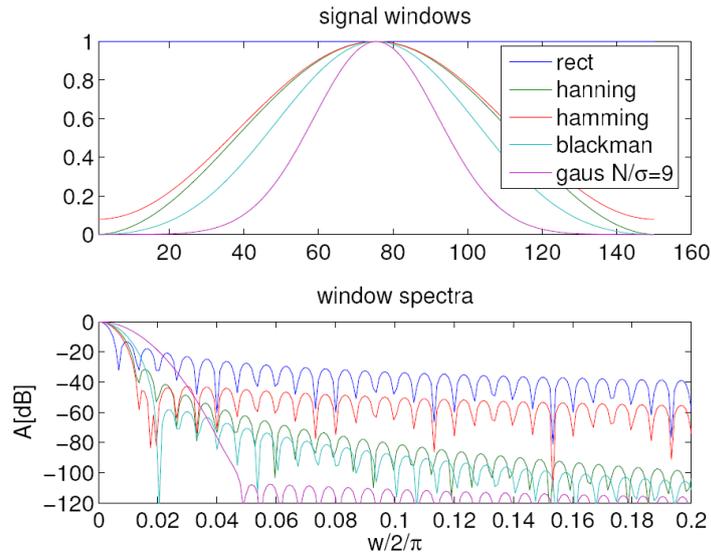


Fig. 4.24 Fenêtres de pondération usuelles et leur TFD

La demi-largeur du lobe principal et le niveau des lobes secondaires pour ces différentes fenêtres est donné au Tableau 4.2. On constate que la diminution de l’amplitude des lobes secondaires se fait au prix d’une augmentation de la largeur du lobe principal.

Type	Demi-largeur du lobe principal (en $F$ )	Niveau des lobes secondaires
<b>Rectangulaire</b>	$1/N$	-13 dB
<b>Hanning</b>	$1.5/N$	-30dB
<b>Hamming</b>	$2/N$	-40 dB
<b>Blackman</b>	$2.75/N$	-60 dB

Tableau 4.2 Propriétés des fenêtres de pondération usuelles

**Exemple 4.17**

Reprenons le signal de l’exemple précédent, et pondérons le avec la fenêtre de Hamming.

```
x=sin(2*pi*130*(0 :63)/1000)+ 0.01*sin(2*pi*300*(0 :63)/1000);
x=x'.*hamming(64);
subplot(2,1,1); % affichage du signal
plot((0:63)/1000,x);
subplot(2,1,2); % affichage de la TFTD du signal fenêtré
plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
```

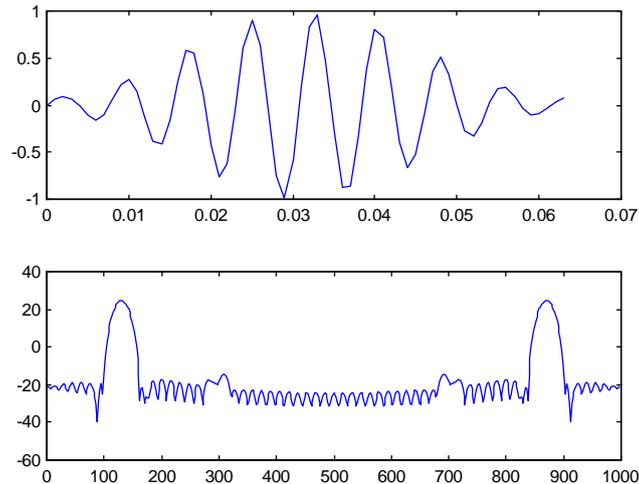


Fig. 4.25 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100, pondérés par une fenêtre de Hamming

Même chose avec une fenêtre de Blackman :

```
x=sin(2*pi*130*(0 :63)/1000)+ 0.01*sin(2*pi*300*(0 :63)/1000);
x=x'.*blackman(64);
subplot(2,1,1); % affichage du signal
plot((0:63)/1000,x);
subplot(2,1,2); % affichage de la TFTD du signal fenêtré
plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
```

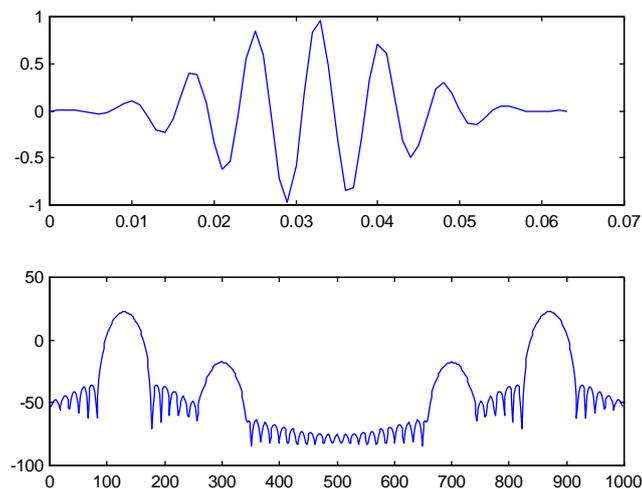


Fig. 4.26 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100, pondérés par une fenêtre de Blackman

L'augmentation de la largeur du lobe principal peut évidemment poser problème, par exemple dans le cas de sinusoides de fréquences fort rapprochées.

### Exemple 4.18

Soit un signal constitué de deux sinusoïdes à 130 et 150 Hz, d'amplitudes identiques, échantillonnées à 1000 Hz. On utilise une fenêtre rectangulaire de 64 points et le calcul de la FFT se fait sur 512 points en fréquence. Montrons l'influence de la fenêtre de pondération :

```
x=sin(2*pi*130*(0 :63)/1000)+ sin(2*pi*150*(0 :63)/1000);
subplot(3,1,1); plot((0:511)/512*1000,20*log10(abs(fft(x,512))));
subplot(3,1,2); plot((0:511)/512*1000,20*log10(abs(fft(x'.*hamming(64),512))));
subplot(3,1,3); plot((0:511)/512*1000,20*log10(abs(fft(x'.*blackman(64),512))));
```

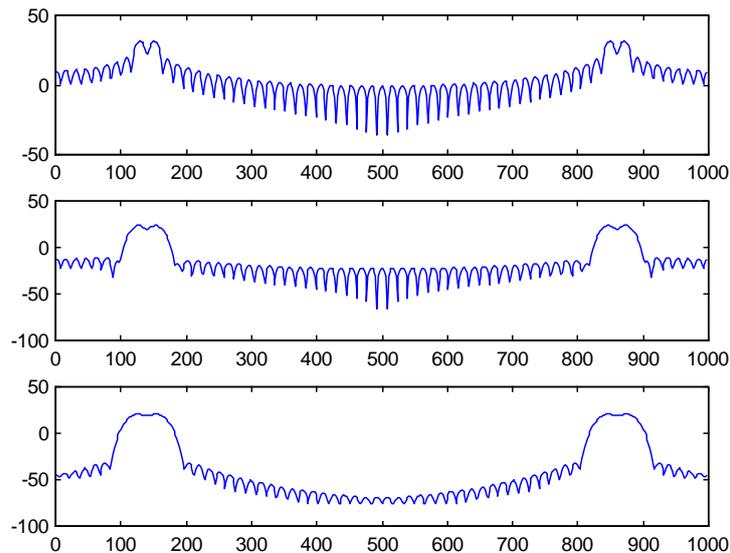


Fig. 4.27 FFT sur 512 points de 64 échantillons d'une somme de deux sinus (à 130 et 300 Hz, échantillonnées à 1000 Hz) d'amplitudes dans un rapport 100

On constate que seule la fenêtre rectangulaire permet dans ce cas de distinguer les deux raies.

### 4.3.5 Analyse à court-terme

Comme l'analyse en fréquence est une opération qui effectue une moyenne sur tout l'axe du temps, certains aspects temporels peuvent être complètement cachés. Il peut alors être intéressant de découper le signal en tranches, pour faire apparaître l'ordonnancement de portions du signal. On parle alors de *transformée de Fourier à court-terme* (STFT : short-term Fourier Transform).

On constate cependant toujours que le découpage en tranches plus petites augmente la résolution temporelle, mais diminue la résolution fréquentielle.

### Exemple 4.19

Soit un signal constitué de deux sinusoïdes successives de fréquences normalisées égales à 0.1 et 0.2 et calculons-en une TFD sur 256 points :

```
T1=128; T2=64;
```

```

x1=sin(2*pi*0.1*(0 :T1-1));
x2=sin(2*pi*0.2*(0 :T2-1));
x=[x1 x2];
subplot(2,1,1) ; plot(x);
subplot(2,1,2) ; plot((0:255)/256,abs(fft(x,256)));6

```

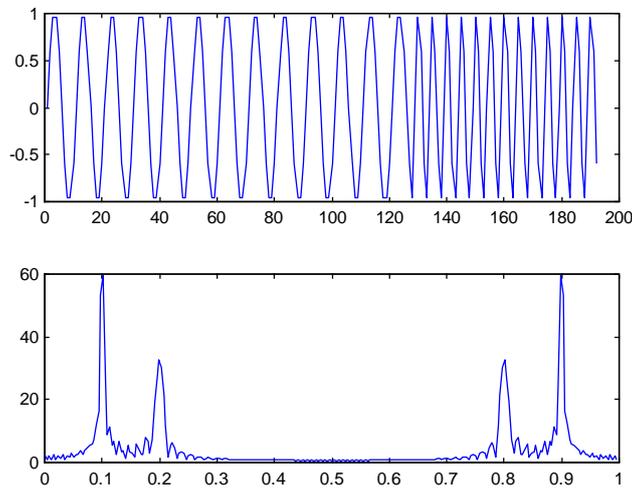


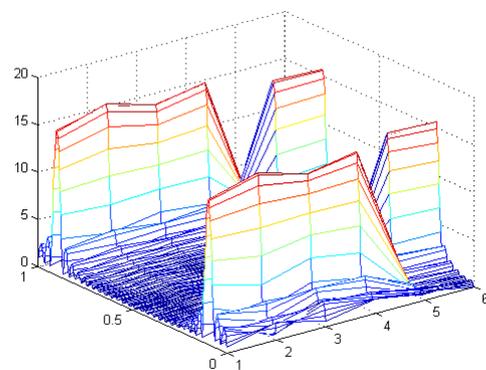
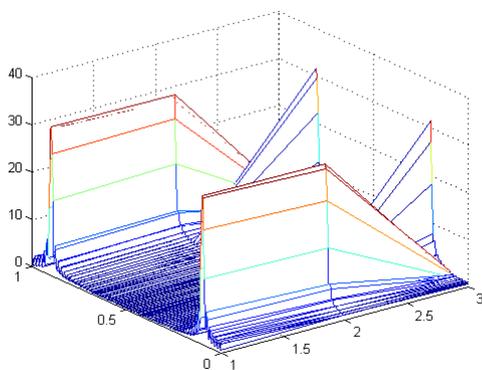
Fig. 4.28 FFT sur 256 points d'une suite de deux sinusoïdes

On constate que la FFT fait bien apparaître les raies des deux sinusoïdes, mais ne renseigne pas sur pas l'ordre dans lequel elles apparaissent. On peut découper le signal en 3 tranches de 64 échantillons et en afficher les TFD successives, et répéter l'opération avec 6 tranches de 32 échantillons:

```

xt=[x(1:64);x(65:128);x(129:192)];
ft=abs(fft(xt',256));
mesh(1:3,(0:255)/256,ft);
xt=[x(1:32); x(33:64);x(65:96);x(97:128);x(129:160);x(161:192)];
ft=abs(fft(xt',256));
mesh(1:6,(0:255)/256,ft);

```



<sup>6</sup> On notera au passage que la fonction `fft` effectue d'elle-même la complétion par des 0 si nécessaire.

Fig. 4.29 FFT successives de tranches d'une suite de deux sinusoides (gauche : 3 tranches de 64 échantillons ; droite : 6 tranches de 32 échantillons).

### 4.4 Convolution linéaire

Le calcul numérique de la convolution (linéaire) est un des grands classiques du traitement du signal. Lorsqu'on veut calculer la réponse, à une entrée  $x(t)$ , d'un SLI analogique caractérisé par une réponse impulsionnelle  $h(t)$ , il est souvent plus facile d'échantillonner l'entrée et de la convoluer numériquement avec une version échantillonnée de la réponse impulsionnelle, que de calculer l'intégrale de convolution analogique. Nous verrons au chapitre 5 que la convolution numérique est également utilisée pour calculer la fonction d'autocorrélation d'un signal aléatoire.

Rappelons que le produit de convolution numérique (linéaire) est donné par :

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i) \tag{4.31}$$

Le nombre de valeurs non nulles retournées par la convolution de deux séquences finies de  $N$  et  $M$  valeurs est de  $N+M-1$ . Il est facile de montrer que le nombre d'opérations nécessaire à la convolution de ces séquences est de l'ordre de  $NM$ .

Il est possible de réduire considérablement cette charge de calcul en calculant la TFD (par FFT) de chacune des séquences à convoluer, en multipliant ces TFDs, et en prenant la TFD inverse du résultat. Cette opération nécessite cependant quelques précautions.

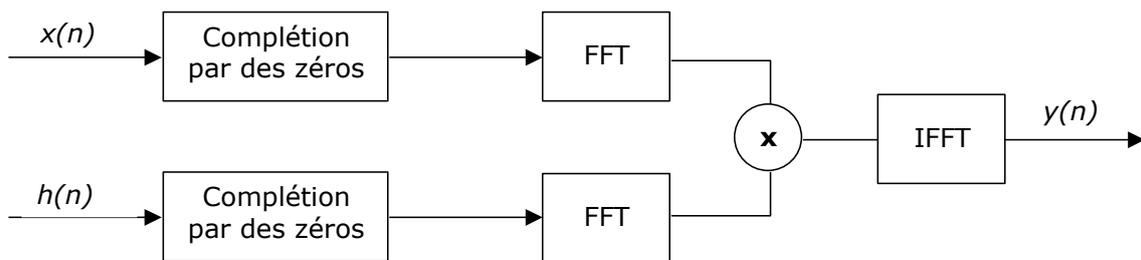


Fig. 4.30 Calcul d'un produit de convolution par FFT/IFFT

Nous avons vu en effet à la section 4.1.5 que le produit de deux TFD est équivalent à la TFD de la convolution *circulaire* des séquences de départ :

$$y_c(n) = x(n) \otimes h(n) = \sum_{i=0}^{N-1} x(i)h((n-i) \bmod N) \tag{4.32}$$

La convolution circulaire implique que les deux séquences soient de même longueur (ou qu'on les complète par des zéros pour qu'elles le soient), et la séquence résultat est de même longueur que les séquences d'entrée.

On comprend aisément que le résultat du produit de convolution circulaire  $y_c(n)$  est identique à celui du produit de convolution linéaire si les deux séquences sont complétées par des zéros de façon à former des séquences de  $N+M-1$  valeurs.

En pratique, on ajoutera souvent plus de zéros que nécessaire (jusqu'à atteindre une puissance de 2), afin de profiter de la rapidité de la FFT.

Le gain en charge de calcul par rapport au calcul du produit de convolution par (4.31) est évidemment plus faible que celui de la FFT sur la TFD, puisqu'on utilise ici 3 FFT (2 directes et une inverse). Il reste que, pour des valeurs élevées de  $N$  ou  $M$ , le calcul de convolution est incomparablement plus efficace par FFT.

### Exemple 4.20

Soit  $a(n)$  et  $b(n)$  constitués respectivement de 200 et 180 échantillons unitaires. On peut en calculer la convolution linéaire à partir de (4.31), puis comparer au calcul par FFT sur 256 et 512 points :

```
a=ones(1,200) ; b=ones(1,180) ;
y1= conv(a,b);
y1(600)=0; % forcer le vecteur à avoir 600 valeurs
subplot(3,1,1); plot(y1);
a(256)=0;b(256)=0; %completer par des zeros
y2=real(ifft(fft(a).*fft(b)));
y2(600)=0 ; % forcer le vecteur à avoir 600 valeurs
subplot(3,1,2); plot(y2);
a(512)=0;b(512)=0; %completer par des zeros
y3=real(ifft(fft(a).*fft(b)));7
y3(600)=0 ; % forcer le vecteur à avoir 600 valeurs
subplot(3,1,3); plot(y3);
```

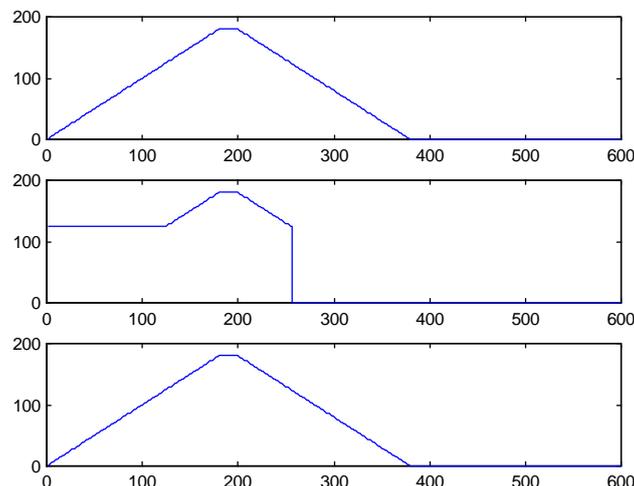


Fig. 4.31 Calcul d'une convolution linéaire (haut : force brute ; centre : par FFT sur 256 points; bas : par FFT sur 512 points)

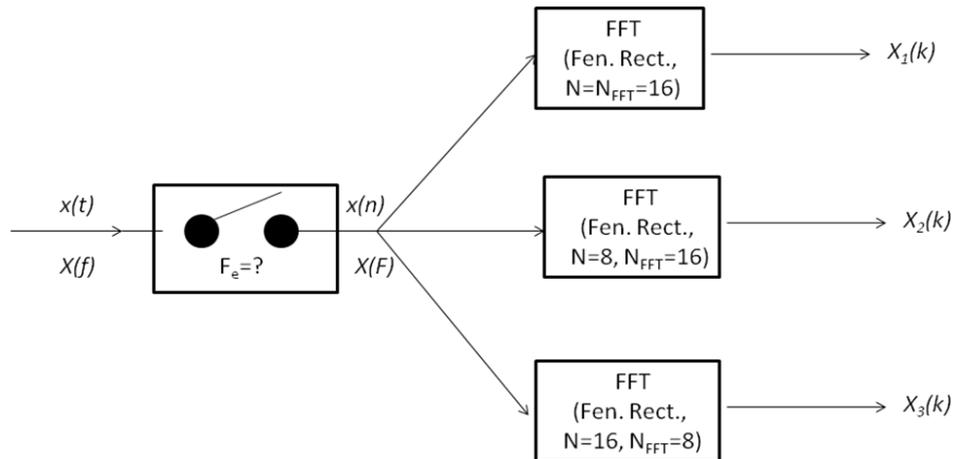
<sup>7</sup> L'appel à la fonction `real` est indispensable : vu les erreurs de calcul, l'IFFT renvoie un résultat très légèrement complexe.

On constate que le calcul par FFT sur 256 points ne conduit pas au bon résultat :  $256 < 200 + 180 - 1$ .

## Exercices

### Exercice 4.1

Soit le système suivant :



Dans le cas où  $x(t) = 10 \cos(30\pi t)$ , échantillonné à  $F_e = 120$  Hz, on demande :

- De déterminer la fréquence d'échantillonnage minimale.
- D'établir les relations entre  $f$ ,  $F$  (fréquence normalisée) et  $k$  (indice de la FFT).
- D'esquisser la Transformée de Fourier  $X(F)$ .
- D'esquisser les FFT  $X_1(k)$ ,  $X_2(k)$  et  $X_3(k)$ , avec le plus de précision possible.
- D'esquisser ces mêmes FFT si on utilise une fenêtre de Hamming.

### Exercice 4.2

Soit le signal  $x(n) = 2\delta(n) + \delta(n-1) + \delta(n-3)$

On demande :

- De calculer la Transformée de Fourier Discrète  $X(k)$  de la séquence numérique  $x(n)$  pour  $N_{TFD} = 4$  par la formule consacrée et par l'interprétation géométrique.
- De calculer les échantillons des signaux  $y_l(n) = x(n) * x(n)$  et  $y_c(n) = x(n) \otimes x(n)$ , où  $\otimes$  indique la convolution circulaire sur  $N_{TFD}$  points.
- Vérifier que, pour  $N_{TFD} = 8$ ,  $y_c(n)$  et  $y_l(n)$  sont identiques.
- De calculer la Transformée de Fourier Discrète  $Y_c(k)$  de la séquence numérique  $y_c(n)$  avec  $N_{TFD} = 4$ , et de vérifier que  $Y_c(k) = X(k)^2$

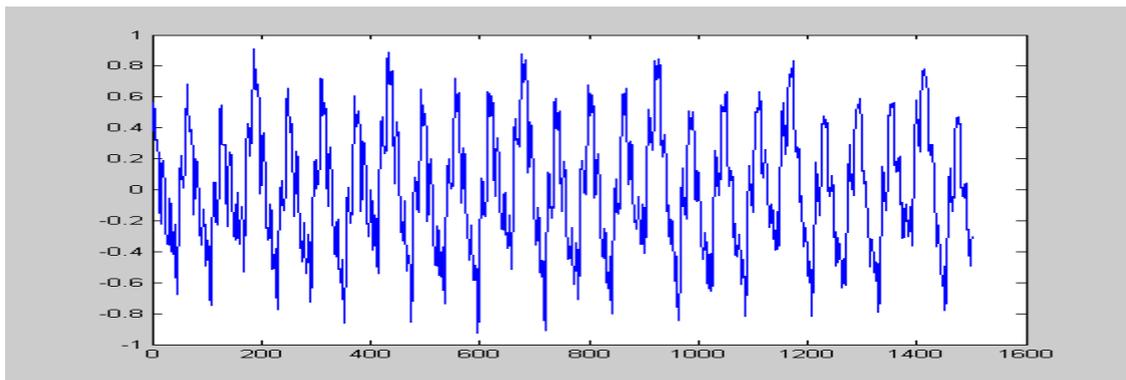
**Solution**

$$b) y_i(n) = \{4, 4, 1, 4, 2, 0, 1, \dots\} \quad y_c(n) = \{6, 4, 2, 4, 6, 4, 2, \dots\}$$

**Exercice 4.3**

Soit le signal suivant, échantillonné à 16 kHz (l'axe des abscisses est gradué en échantillons). On demande :

- (a) combien d'échantillons ( $N_{\text{TFD}, \text{min}}$ ) de ce signal il faut prendre au minimum pour que leur TFD (avec fenêtre rectangulaire) permette de distinguer les harmoniques entre elles (Justifier).
- (b) Et si on utilise une fenêtre de Hamming, comment ce nombre minimum évolue-t-il?
- (c) Si l'amplitude du fondamental et des harmoniques est donnée par  $A_i = 0.5/i$  ( $i$  est l'indice de l'harmonique;  $i=1$  pour le fondamental), esquisser la TFD obtenue avec  $N_{\text{TFD}} = 10 * N_{\text{TFD}, \text{min}}$  (mentionner les amplitudes des raies qui seront mesurées).

**Solution**

- a) 500      b) 1000

**Exercice 4.4**

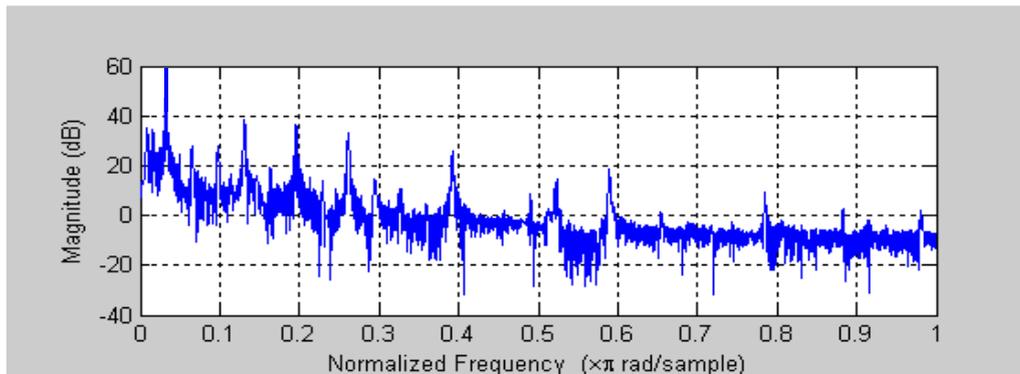
Si la TFD de  $\{x_0, x_1, \dots, x_{N-2}, x_{N-1}\}$  est donnée par  $\{X_0, X_1, X_2, \dots, X_{N-1}\}$ , que vaut la TFD de  $\{x_0+x_1, x_1+x_2, x_2+x_3, \dots, x_{N-2}+x_{N-1}, x_{N-1}+x_0\}$  ?

**Solution**

$$X'_k = X_k (1 + e^{j\frac{2\pi}{N}k})$$

**Exercice 4.5**

Soit la TFD suivante (où l'axe des abscisses est donné pour une pulsation normalisée allant de 0 à  $\pi$  et où on suppose avoir utilisé une fenêtre rectangulaire):



On demande :

- (a) la fréquence du fondamental, si  $F_e=1000$  Hz
- (b) la période du signal, en échantillons
- (c) un ordre de grandeur du nombre d'échantillons du signal qui ont été utilisés ( $N_{TFD}$ )
- (d) l'amplitude réelle du fondamental

**Solution**

- c)  $\sim 400$  échantillons    d) 5

# CHAPITRE 5

## TRAITEMENT NUMERIQUE DES SIGNAUX ALEATOIRES

Steven L. Kay, ingénieur électricien américain, professeur de traitement du signal à l'université de Rhodes Island. Spécialiste en statistique mathématique en liaison avec le traitement du signal, il est l'auteur d'un ouvrage de référence dans le domaine de la détection de signaux (*Fundamentals of Statistical Signal Processing*, Prentice Hall). Membre d'honneur de l'IEEE pour sa contribution à la détection et à l'estimation spectrale paramétrique.



Les signaux étudiés jusqu'ici possèdent une forme simple, dont on peut rendre compte sous forme analytique. L'utilisation pratique de tels signaux est assez restreinte : elle touche principalement les cas où l'ingénieur contrôle le type des signaux se manifestant dans un système, parce qu'il en assure la *synthèse* (par exemple dans le cas du signal émis par un radar ou un sonar ou dans le cas des sons ou des images de synthèse). A contrario il est rare que les signaux naturels, résultant de causes multiples et souvent incontrôlables et non mesurables, puissent être caractérisés par une expression analytique simple. C'est le cas par exemple des signaux reçus par un sonar ou un radar, de la parole, des signaux biologiques (Fig. 5.1) comme l'électro-cardiogramme ou de l'électro-encéphalogramme, des signaux géodésiques ou encore des signaux boursiers étudiés par les économistes. On les qualifie alors d'*aléatoires*, par opposition aux premiers que l'on qualifie de *déterministes*<sup>1</sup>.

---

<sup>1</sup> En réalité, le qualificatif de *déterministe* ou *aléatoire* n'est pas vraiment une caractéristique des signaux (qui sont tous fondamentalement déterministes) mais plutôt du degré de connaissance (ou de l'ignorance) que nous en avons. Tout ce qui sera dit ici sur les signaux « aléatoires » est donc applicable a fortiori aux signaux « déterministes ».

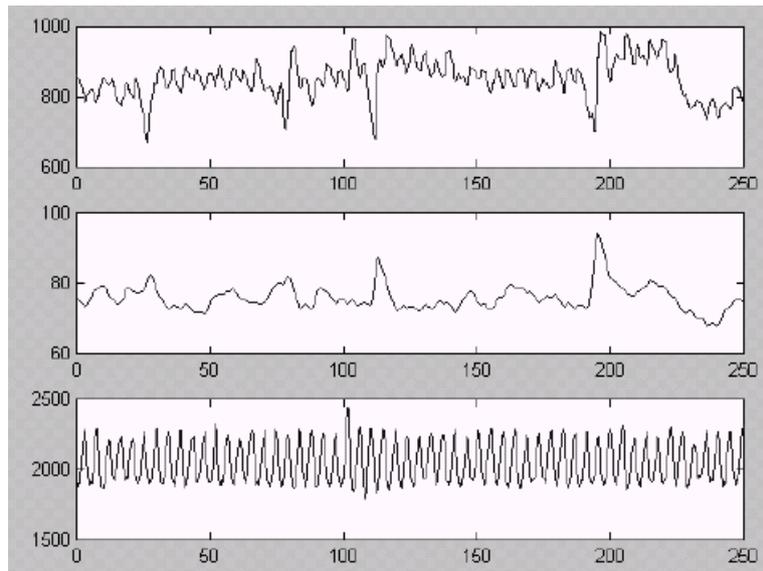


Fig. 5.1 Exemples de signaux bio-médicaux.  
 Haut : intervalles entre les battements du cœur (ms) ;  
 Centre : Pression artérielle moyenne (mm Hg);  
 Bas : Volume respiratoire.

Après un rappel sur les variables aléatoires et une application importante à la quantification uniforme des signaux (5.1), nous examinerons les propriétés théoriques des signaux aléatoires : densité de probabilité, moyenne, variance, autocorrélation et densité spectrale de puissance (5.2). Nous verrons comment ce que deviennent ces grandeurs lorsqu'on échantillonne un signal aléatoire (5.3). Nous poursuivrons en considérant le filtrage d'un signal aléatoire à temps discret et montreront comment les propriétés de ces signaux s'en trouvent modifiées (5.4). Nous étudierons pour terminer divers estimateurs de ces grandeurs à partir des signaux aléatoires eux-mêmes (5.5). Une difficulté fréquente résulte de ce que les enregistrements disponibles de ces signaux sont souvent de durée limitée, soit par manque réel de données, soit par manque de stationnarité du signal, ou encore par soucis de limiter le volume de calcul; il faut en général accepter un *compromis* entre la *résolution* et la *précision* de l'estimation.

## 5.1 Variables aléatoires

### 5.1.1 Rappel théorique

Une variable aléatoire continue  $X$  est complètement caractérisée par sa fonction de *densité de probabilité*  $p_X(x)$  qui permet le calcul de la probabilité que  $X$  appartienne à l'intervalle  $(a,b)$  :

$$P(X \in (a,b)) = \int_a^b p_X(x) dx \quad (5.1)$$

Il est équivalent de spécifier sa *fonction de répartition*  $F_X(x)$ , qui est l'intégrale de la fonction de densité de probabilité :

$$F_X(x) = \int_{-\infty}^x p_X(\chi) d\chi \tag{5.2}$$

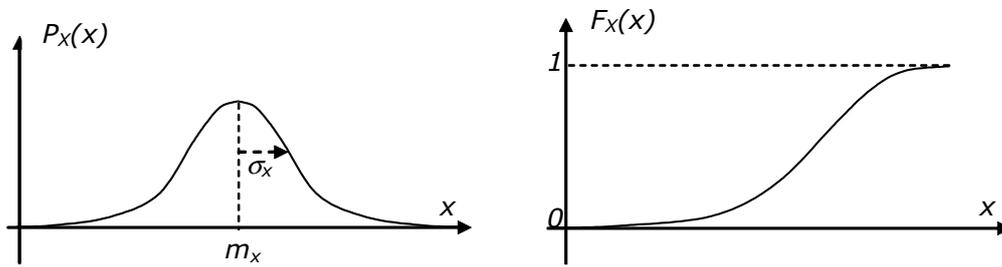


Fig. 5.2 Densité de probabilité et fonction de distribution d'une v.a.

On caractérise souvent (et souvent de façon incomplète) une variable aléatoire par sa *moyenne*  $m_X$  et sa *variance*  $\sigma_X^2$ , définies par :

$$m_X = E[X] = \int_{-\infty}^{\infty} x p_X(x) dx \tag{5.3}$$

$$\sigma_X^2 = E[(X - m_X)^2] = \int_{-\infty}^{\infty} (x - m_X)^2 p_X(x) dx \tag{5.4}$$

De façon similaire, on caractérise complètement la relation statistique entre deux variables aléatoires continues  $X$  et  $Y$  par une *densité de probabilité conjointe*  $p_{XY}(x, y)$  (Fig. 5.3) telle que :

$$P(X \in (a, b), Y \in (c, d)) = \int_a^b \int_c^d p_{XY}(x, y) dx dy \tag{5.5}$$

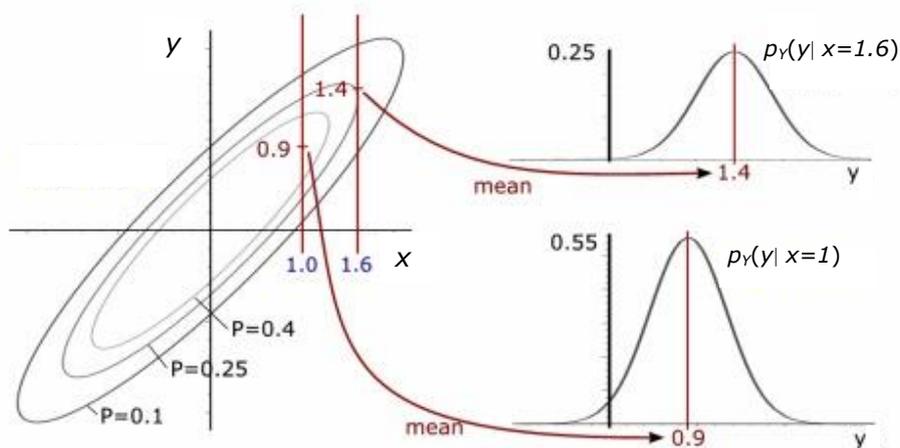


Fig. 5.3 Densité de probabilité conjointe de deux v.a. (ici deux v.a. *dépendantes*)

On caractérise souvent cette relation de façon plus simple (et souvent incomplète) par la *covariance*  $\text{cov}(X,Y)$  entre  $X$  et  $Y$ , ou par sa valeur normalisée entre  $-1$  et  $1$ , la *corrélation*  $\rho(X,Y)$  :

$$\text{cov}(X,Y) = E[(X - m_X)(Y - m_Y)] \quad (5.6)$$

$$\rho(X,Y) = \frac{\text{cov}(X,Y)}{\sqrt{\sigma_X \sigma_Y}} \quad (5.7)$$

En particulier, deux variables sont dites *indépendantes* ssi la valeur prise lors du tirage aléatoire de la première n'a aucune influence sur le tirage de la seconde :

$$p_{XY}(x,y) = p_X(x)p_Y(y) \quad (5.8)$$

Elle sont dites *non corrélées* ssi :

$$\text{cov}(X,Y) = 0 \quad (5.9)$$

On montre que deux variables aléatoires indépendantes sont non corrélées, mais qu'en général la réciproque est fautive. La propriété (5.8) doit en effet être vérifiée pour tout  $x$  et  $y$ , alors que (5.9) est une propriété plus globale.

Comme les variables déterministes, les variables aléatoires possèdent une algèbre. L'opération la plus simple est la somme  $Y$  de deux variables aléatoires indépendantes  $X_1$  et  $X_2$ . Le résultat est bien entendu également une variable aléatoire, et sa densité de probabilité est facile à obtenir : la probabilité d'une valeur particulière  $y$  de  $Y$  est bien entendu la somme (continue) des probabilités de chaque paire de valeurs particulières  $(x_1, y-x_1)$  dont la somme donne  $y$ . Si les variables sont indépendantes, la probabilité de chaque paire est donnée par le produit des probabilités de leurs éléments, et on trouve tout naturellement que la densité de probabilité de  $Y$  est le produit de convolution des densités de probabilité initiales :

$$p_Y(y) = \int_{-\infty}^{\infty} p_{X_1}(x)p_{X_2}(y-x)dx \quad (5.10)$$

$$p_Y(y) = p_{X_1}(x) * p_{X_2}(x)$$

### Exemple 5.1 - Variable aléatoire à distribution uniforme sur $(-q/2, q/2)$

Cette v.a. est caractérisée par une densité de probabilité constante (Fig. 5.4) .

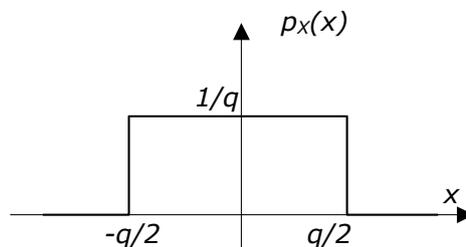


Fig. 5.4 Densité de probabilité de la v.a. uniforme

Sa moyenne est nulle. Sa variance est donnée par :

$$\sigma_x^2 = \int_{-q/2}^{q/2} \frac{x^2}{q} dx = \left[ \frac{x^3}{3q} \right]_{-q/2}^{q/2} = \frac{q^2}{12} \tag{5.11}$$

Il est facile de générer un tirage de cette v.a. sous Matlab, d'estimer sa moyenne et sa variance, et d'afficher un histogramme donnant une idée de sa densité de probabilité<sup>2</sup> :

```
x=rand(1,1000); %v.a. uniforme entre 0 et 1
x=(x-1/2)*10; %v.a. uniforme entre -5 et 5
m=mean(x)
m =
    0.0211
s=std(x)^2 % std = standard deviation = écart-type
s =
    8.3717
hist(x,20);
```

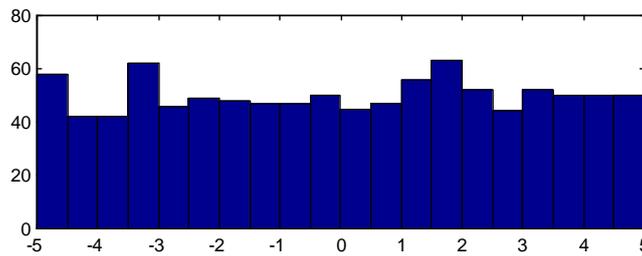


Fig. 5.5 Histogramme d'un tirage de v.a. uniforme

**Exemple 5.2 - Variable aléatoire à distribution normale  $N(m,\sigma)$**

Cette v.a. est caractérisée par une densité de probabilité Gaussienne (Fig. 5.6) :

$$p_x(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \tag{5.12}$$

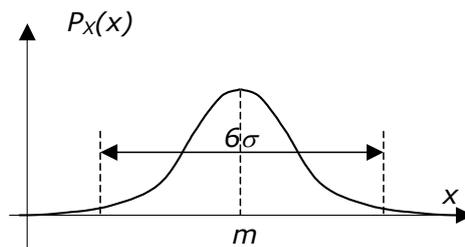


Fig. 5.6 Densité de probabilité de la v.a. Gaussienne  $N(m,\sigma)$

La moyenne de cette v.a. est égale à  $m$ . Sa variance est égale à  $\sigma^2$ . On montre facilement que plus de 99% des valeurs d'une v.a. Gaussienne tombent dans l'intervalle  $(m-3\sigma, m+3\sigma)$ .

<sup>2</sup> On constate en passant que, bien que la moyenne, la variance, et la densité de probabilité d'une v.a. soient des grandeurs déterministes, leurs estimateurs à partir d'un tirage de cette v.a. sont eux-mêmes des variables aléatoires. Nous étudierons plus loin les caractéristiques de ces estimateurs.

## 6 TRAITEMENT NUMERIQUE DES SIGNAUX ALEATOIRES

Il est tout aussi facile de générer un tirage de cette v.a. sous Matlab et d'en estimer les caractéristiques :

```
x=randn(1,1000); %v.a. gaussienne N(0,1)
x=x*10+8; %v.a. gaussienne N(8,10)
m=mean(x)
m =
    7.5692
s=std(x)^2 % std = standard deviation = écart-type
s =
    89.0175
hist(x,20);
```

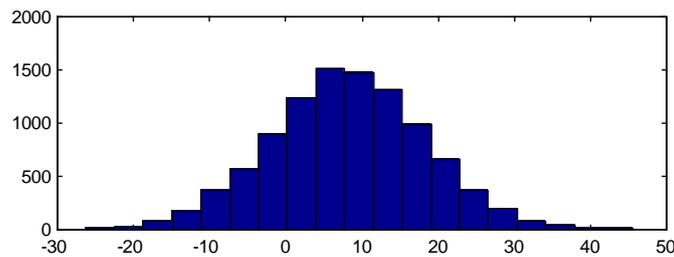


Fig. 5.7 Histogramme d'un tirage de v.a. gaussienne  $N(8,10)$

Rappelons pour terminer qu'en vertu du *théorème central limite*, la somme d'un grand nombre  $N$  de v.a. indépendantes de densités de probabilité identiques (de moyenne  $m_x$  et de variance  $\sigma_x^2$ ) tend vers une v.a. gaussienne :  $N(Nm_x, \sigma_x / \sqrt{N})$ <sup>3</sup>. Ceci explique en soi l'importance de cette variable aléatoire.

On montre facilement qu'il est possible d'obtenir le tirage d'une v.a.  $X$  à densité de probabilité quelconque  $p_X(x)$  à partir d'un tirage de v.a.  $U$  à densité uniforme, en associant à chaque réalisation  $u$  de  $U$  la valeur  $x=F_X^{-1}(u)$  où  $F$  est la fonction de répartition de  $X$  (Fig. 5.8).

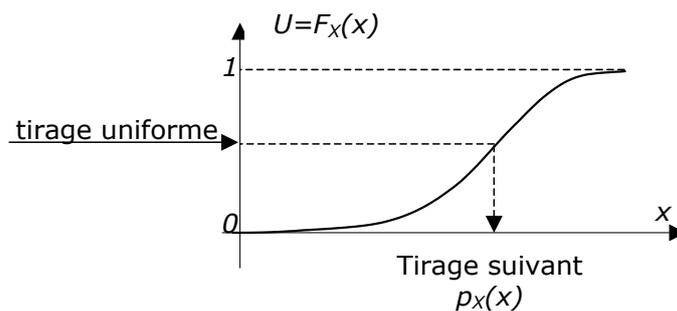


Fig. 5.8 Tirage d'une v.a. de densité de probabilité quelconque à partir d'une v.a. uniforme

## 5.2 Signaux aléatoires (rappel)

<sup>3</sup> Ce qui revient à dire, vu (5.10), que la convolution d'un grand nombre de fonctions de densités de probabilités tend toujours vers une fonction Gaussienne.

Au contraire d'un signal déterministe, la valeur prise à tout instant  $t$  par un signal aléatoire  $x(t)$  n'est pas connue de façon univoque, mais plutôt de façon probabiliste : elles sont considérées comme le résultat du tirage d'une v.a.  $X(t)$ , dont la densité de probabilité peut en toute généralité dépendre de  $t$  (Fig. 5.15) :

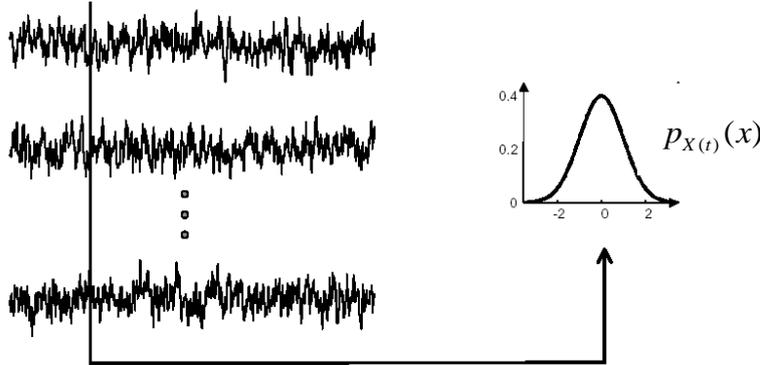


Fig. 5.9 Pour  $t$  fixé, les valeurs  $x(t)$  sont des tirages d'une variable aléatoire  $X(t)$

On associe ainsi à chaque instant  $t$  une variable aléatoire  $X(t)$  de fonction de densité de probabilité  $p_{X(t)}(x)$ , que l'on peut estimer à partir d'un grand nombre de tirages  $x(t)$  de ce signal au temps  $t$ . L'ensemble des fonctions  $p_{X(t)}(x)$  pour tous les instants  $t$  définit ce que l'on appelle la *statistique d'ordre 1* du signal aléatoire.

Bien que les variables aléatoires  $X(t_1)$  et  $X(t_2)$  puissent être totalement indépendantes les unes des autres (voir plus loin : « bruit blanc »), il existe le plus souvent un lien statistique entre elles. Cette relation peut être caractérisée par le densité de probabilité conjointe  $p_{X(t_1)X(t_2)}(x, y)$  des deux variables aléatoires. L'ensemble des fonctions  $p_{X(t_1)X(t_2)}(x, y)$  en fonction de  $(t_1, t_2)$  définit ce que l'on appelle la *statistique d'ordre 2* du signal aléatoire.

En pratique, on caractérise plus simplement (et plus approximativement) la statistique d'ordre 1 par des moyennes et des écarts-types (plutôt que par les fonctions de densité de probabilités complètes), et la statistique d'ordre 2 par la *fonction d'autocorrélation* du signal (5.2.1) ou par sa transformée de Fourier, la *densité spectrale de puissance* (5.2.2).

## 5.2.1 Fonction d'autocorrélation

### Définitions

La *fonction d'autocorrélation* d'un signal aléatoire (complexe dans le cas général) est donnée par :

$$\phi_{XX}(t_1, t_2) = E[X(t_1)X^*(t_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xy^* p_{X(t_1)X(t_2)}(x, y) dx dy \quad (5.13)$$

De même, on parle d'*intercorrélation* (*cross-correlation*) entre deux signaux aléatoires :

$$\phi_{XY}(t_1, t_2) = E[X(t_1)Y^*(t_2)] \quad (5.14)$$

### Stationnarité au sens large et ergodicité

Dans la suite de ce chapitre, nous considérerons souvent les signaux aléatoires *ergodiques* et *stationnaires au sens large (SSL)*. La stationnarité au sens large impose que la statistique d'ordre 1 soit indépendante de l'instant  $t$  où on la mesure :

$$p_{X(t_1)}(x) = p_{X(t_2)}(x) \stackrel{\Delta}{=} p_X(x)$$

$$\text{et donc : } m_{X(t_1)} = m_{X(t_2)} \stackrel{\Delta}{=} m_X \quad (5.15)$$

$$\sigma_{X(t_1)} = \sigma_{X(t_2)} \stackrel{\Delta}{=} \sigma_X$$

et que la statistique d'ordre 2 ne dépende que de l'intervalle  $t_2 - t_1$ , et pas de la valeur absolue de  $t_1$  ou  $t_2$  :

$$p_{X(t_1)X(t_2)}(x, y) = p_{X(t_1+\tau)X(t_2+\tau)}(x, y) \quad (5.16)$$

$$\text{et donc : } \phi_{XX}(t_1, t_2) \stackrel{\Delta}{=} \phi_{XX}(t_2 - t_1)$$

L'ergodicité correspond quant à elle au fait que l'estimation des caractéristiques statistiques peut se faire sur la suite des valeurs  $x(t)$  d'un seul tirage du signal aléatoire<sup>4</sup>  $X(t)$ .

Il s'ensuit que :

$$m_X = \overline{x(t)} = \lim_{T \rightarrow \infty} \left[ \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt \right]$$

$$\sigma_X^2 = \overline{|x(t) - m_X|^2} = \lim_{T \rightarrow \infty} \left[ \frac{1}{T} \int_{-T/2}^{T/2} |x(t) - m_X|^2 dt \right] \quad (5.17)$$

$$\phi_{XX}(\tau) = \overline{x(t)x^*(t-\tau)} = \lim_{T \rightarrow \infty} \left[ \frac{1}{T} \int_{-T/2}^{T/2} x(t)x^*(t-\tau) dt \right]$$

On notera que l'équation la définition de la variance par (5.36) est analogue à la définition de la puissance d'un signal périodique par (2.6). En effet, tout comme les signaux périodiques, les signaux aléatoires sont de puissance finie mais d'énergie infinie (puisque'ils sont sensés durer un temps infini). Dans la suite, on utilisera indifféremment les mots *puissance* et *variance* pour un signal aléatoire stationnaire et ergodique.

Plus généralement encore, on notera que la définition de la fonction d'autocorrélation  $\phi_{XX}(\tau)$  dans (5.36) correspond au produit scalaire (tel que défini au Chapitre 2)  $\langle x(t), x(t-\tau) \rangle$  entre le signal  $x(t)$  et une version décalée d'un temps  $\tau$  de ce même signal. Comme les signaux aléatoires sont des signaux de puissance finie, l'expression de  $\phi_{XX}(\tau)$  ci-dessus fait tout naturellement intervenir, non pas simplement une intégrale  $dt$  comme dans (2.25), mais bien une intégrale rapportée à un temps.

### Propriétés

<sup>4</sup> La plus souvent, d'ailleurs, on ne dispose que d'une réalisation (parole, image, signaux bio-médicaux), sur laquelle il faut mesurer des caractéristiques.

On voit que, par construction, la fonction d'autocorrélation d'un signal aléatoire stationnaire et ergodique possède la propriété de *symétrie hermitienne* :

$$\phi_{XX}(-\tau) = \phi_{XX}^*(\tau) \quad (5.18)$$

La valeur de  $\phi_{XX}(0)$  est quant à elle liée à la variance par :

$$\begin{aligned} \phi_{XX}(0) &= E[XX^*] \\ &= E[(X - m_X + m_X)(X - m_X + m_X)^*] \\ &= E[(X - m_X)(X - m_X)^*] + E[m_X m_X^*] \\ &\quad + E[(X - m_X)m_X^*] + E[m_X(X - m_X)^*] \\ &= \sigma_X^2 + |m_X|^2 \end{aligned} \quad (5.19)$$

Pour un signal de moyenne nulle,  $\phi_{XX}(0)$  est même égale à la variance.

### 5.2.2 Densité spectrale de puissance

On appelle *spectre* ou *densité spectrale de puissance (PSD: Power Spectral Density)*  $S_{XX}(f)$  d'un signal aléatoire stationnaire, la transformée de Fourier de sa fonction d'autocorrélation :

$$S_{XX}(f) = \int_{-\infty}^{\infty} \phi_{XX}(t) e^{-j\omega t} dt \quad \text{avec } \omega = 2\pi f \quad (5.20)$$

Cette fonction est l'équivalent, pour les signaux de puissance finie, de ce qu'est la densité spectrale d'énergie (2.26) pour les signaux d'énergie finie.

Comme la fonction d'autocorrélation possède la symétrie hermitique, la densité spectrale de puissance est purement réelle. Si de plus la variable aléatoire  $X(t)$  est réelle, alors la fonction d'autocorrélation est réelle paire, et la densité spectrale de puissance l'est aussi.

La transformée de Fourier inverse de la PSD est bien entendu la fonction d'autocorrélation :

$$\phi_{XX}(t) = \int_{-\infty}^{\infty} S_{XX}(f) e^{j\omega t} df \quad \text{avec } \omega = 2\pi f \quad (5.21)$$

En particulier, on notera que, pour un signal de moyenne nulle, la variance (ou puissance) est donnée par la surface sous-tendue par  $S_{XX}(f)$  (Fig. 5.10) :

$$\sigma_X^2 = \phi_{XX}(0) = \int_{-\infty}^{\infty} S_{XX}(f) df \quad (5.22)$$

Cette dernière équation justifie l'appellation de "densité spectrale de puissance" pour  $S_{XX}(f)$ .

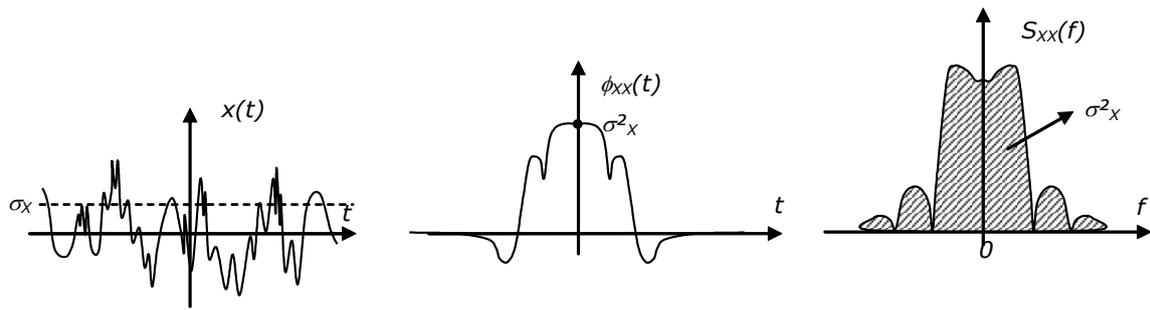


Fig. 5.10 Exemple de fonction d'autocorrélation et de densité spectrale de puissance d'un signal aléatoire à valeurs réelles et à moyenne nulle

**Exemple 5.3 – Bruit blanc (théorique)**

Un bruit blanc (théorique) est un signal aléatoire dont la densité spectrale de puissance est constante, quelle que soit la bande de fréquences où on l'examine (tout comme une lumière blanche est composée d'ondes lumineuse dans tout le spectre visible) :

$$S_{xx}(f) = \sigma_x^2 \tag{5.23}$$

En conséquence, sa fonction d'autocorrélation est une impulsion de Dirac en  $t=0$  (Fig. 5.11) :

$$\phi_{xx}(t) = \sigma_x^2 \delta(t) \tag{5.24}$$

Les v.a.  $X(t_1)$  et  $X(t_2)$  d'un tel signal en deux instants différents sont non corrélées et de moyennes nulles.

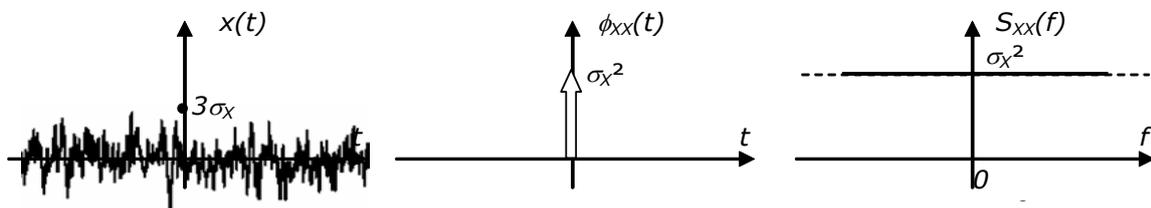


Fig. 5.11 Fonction d'autocorrélation et densité spectrale de puissance d'un bruit blanc théorique (ici, un bruit blanc Gaussien)

Les bruits blancs diffèrent entre eux par leur statistique du 1<sup>er</sup> ordre (la densité de probabilité de la v.a. sous-jacente) : bruit blanc uniforme, gaussien, etc. <sup>5</sup>

**Exemple 5.4 – Bruit blanc (réel)**

En pratique, un bruit blanc réel ne peut avoir qu'une largeur de bande finie  $B$  (Fig. 5.13). Sa fonction d'autocorrélation a donc la forme d'un sinus cardinal.

<sup>5</sup> Contrairement à ce qu'on pense parfois, le fait qu'un bruit soit Gaussien n'implique pas qu'il soit blanc, et réciproquement.

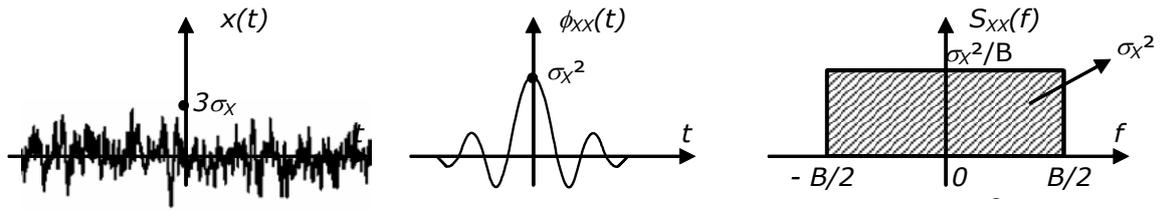


Fig. 5.12 Fonction d'autocorrélation et densité spectrale de puissance d'un bruit blanc réel

### Exemple 5.5 – Signal périodique

La fonction d'autocorrélation d'un signal périodique (que l'on peut toujours considérer comme un signal aléatoire) est elle-même périodique. En effet, considérons pour commencer un signal  $x(t) = Ae^{j(\omega_0 t + \phi)}$ . Sa fonction d'autocorrélation est donnée par :

$$\begin{aligned} \phi_{XX}(t) &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} Ae^{j(\omega_0 \tau + \phi)} Ae^{-j(\omega_0(\tau+t) + \phi)} d\tau \\ &= \frac{A^2}{T_0} \int_{-T_0/2}^{T_0/2} e^{j\omega_0 \tau + \phi} e^{-j\omega_0(\tau+t) - \phi} d\tau \\ &= A^2 e^{-j\omega_0 t} \end{aligned} \quad (5.25)$$

Sa densité spectrale de puissance est donc une impulsion de Dirac :

$$S_{XX}(f) = A^2 \delta(f - f_0) \quad \text{avec} \quad f_0 = \omega_0 / 2\pi \quad (5.26)$$

Comme un signal périodique est une somme d'exponentielles de fréquences harmoniques :

$$x(t) = \sum_{i=0}^p A_i e^{j(i\omega_0 t + \phi_i)} \quad (5.27)$$

et comme ces exponentielles sont orthogonales entre elles, l'autocorrélation et la densité spectrale de puissance d'une fonction périodique  $x(t)$  sont données par :

$$\phi_{XX}(t) = \sum_{i=0}^p A_i^2 e^{-j i \omega_0 t} \quad S_{XX}(f) = \sum_{i=0}^p A_i^2 \delta(f - i f_0) \quad (5.28)$$

En particulier, la fonction d'autocorrélation associée à une fonction cosinoïdale  $x(t) = A \cos(\omega_0 t + \phi)$  est donnée par :

$$\phi_{XX}(t) = \frac{A^2}{2} \cos(\omega_0 t) \quad S_{XX}(f) = \frac{A^2}{4} \delta(f - f_0) + \frac{A^2}{4} \delta(f + f_0) \quad (5.29)$$

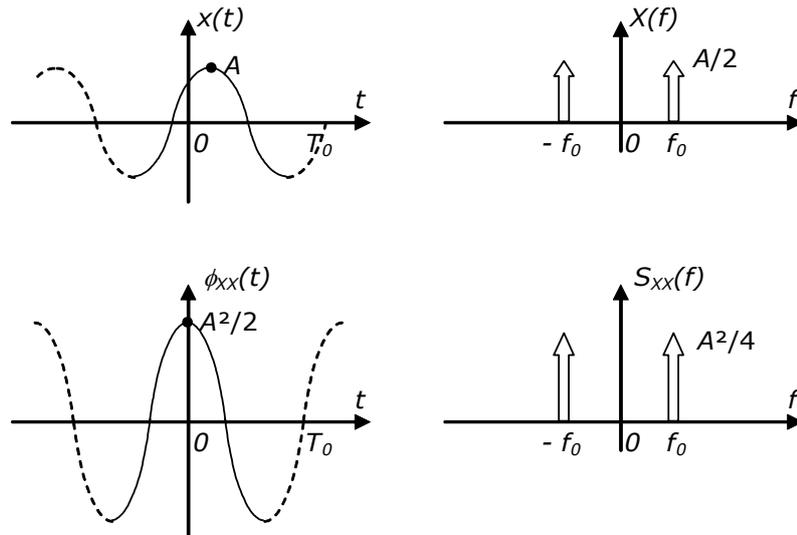


Fig. 5.13 Fonction d'autocorrélation et densité spectrale de puissance de  $x(t) = A \cos(\omega_0 t + \phi)$  et comparaison avec la transformée de Fourier du signal

**Exemple 5.6 – Signal aléatoire de moyenne non nulle**

La fonction d'autocorrélation d'un signal aléatoire  $X(n)$  de moyenne nulle, auquel on ajoute une constante  $m_X$  est donnée par :

$$\begin{aligned} \phi_{(X+m_X)(X+m_X)}(\tau) &= E[(X(t) + m_X)(X(t - \tau) + m_X)^*] \\ &= \phi_{XX}(\tau) + |m_X|^2 \end{aligned} \tag{5.30}$$

Sa densité spectrale de puissance est donc égale à celle du signal de départ, à laquelle on ajoute une impulsion de Dirac en 0 :

$$S_{(X+m_X)(X+m_X)}(f) = S_{XX}(f) + |m_X|^2 \delta(f) \tag{5.31}$$

Appliquées au signal correspondant à la Fig. 5.10, ces conclusions conduisent à la Fig. 5.14.

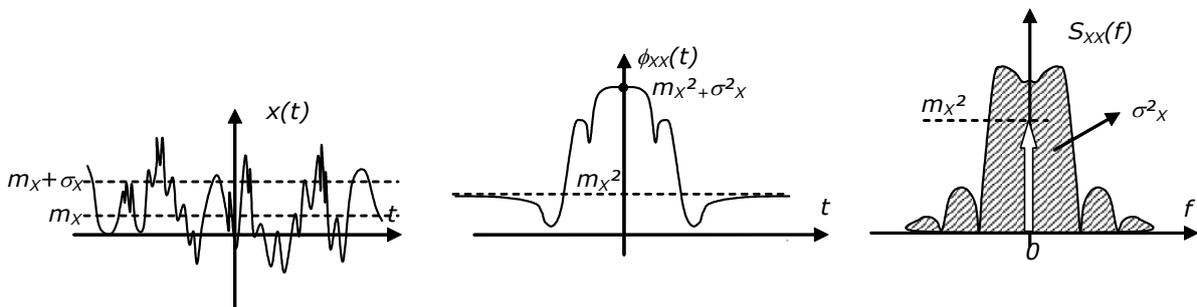


Fig. 5.14 Exemple de fonction d'autocorrélation et de densité spectrale de puissance d'un signal aléatoire à valeurs réelles et à moyenne non-nulle

**Exemple 5.7 – Somme de deux signaux aléatoires non-corrélés**

La fonction d'autocorrélation d'une somme de deux signaux aléatoires  $X(t)$  et  $Y(t)$  est donnée par :

$$\begin{aligned}\phi_{(X+Y)(X+Y)}(\tau) &= E[(X(t)+Y(t))(X(t-\tau)+Y(n-\tau))^*] \\ &= \phi_{XX}(\tau) + \phi_{YY}(\tau) + \phi_{XY}(\tau) + \phi_{YX}(\tau)\end{aligned}\quad (5.32)$$

Si les deux signaux sont non-corrélés, elle se réduit donc à :

$$\phi_{(X+Y)(X+Y)}(\tau) = \phi_{XX}(\tau) + \phi_{YY}(\tau) \quad (5.33)$$

La densité spectrale de puissance de la somme est alors égale à la somme des densités spectrales de puissance :

$$S_{X+Y}(F) = S_{XX}(F) + S_{YY}(F) \quad (5.34)$$

L'exemple précédent n'est d'ailleurs qu'un cas particulier de ce principe.

### 5.3 Signaux aléatoires à temps discret

L'échantillonnage d'un signal aléatoire ne pose pas de problème théorique particulier conduit à créer un signal aléatoire à temps discret, qui peut être formalisé mathématiquement comme une séquence temporelle discrète de variables aléatoires (Fig. 5.15) :

$$\{X(n)\} = \{\dots, X(-n), \dots, X(-1), X(0), X(1), \dots, X(n), \dots\} \quad (5.35)$$

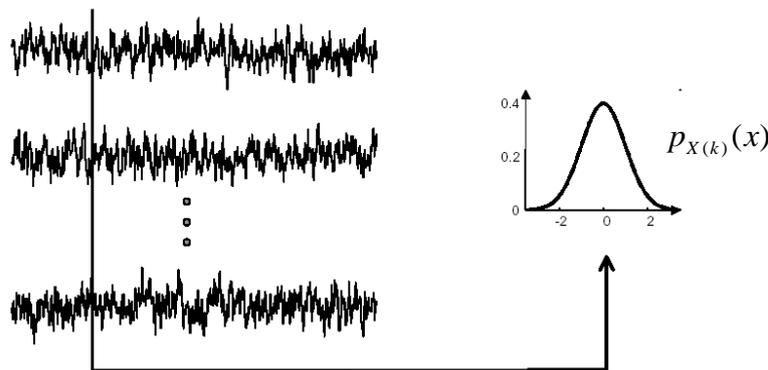


Fig. 5.15 Pour  $k$  fixé, les échantillons  $x(k)$  sont des tirages d'une variable aléatoire  $X(k)$

On associe ainsi à chaque échantillon  $k$  une variable aléatoire  $X(k)$  de fonction de densité de probabilité  $p_{X(k)}(x)$ . La séquence des fonctions  $p_{X(k)}(x)$  en fonction de  $k$  définit la *statistique d'ordre 1* du signal aléatoire et la séquence des fonctions  $p_{X(i)X(j)}(x, y)$  en fonction de  $(i, j)$  définit sa *statistique d'ordre 2*.

On supposera le plus souvent que les signaux aléatoires discrets sont stationnaires et ergodiques, et on caractérisera plus simplement un signal aléatoire par sa moyenne, sa variance, sa fonction d'autocorrélation, et sa densité spectrale de puissance à *temps discret*.

$$\begin{aligned}m_X &= \overline{x(n)} = \lim_{N \rightarrow \infty} \left[ \frac{1}{2N+1} \sum_{i=-N}^N x(n) \right] \\ \sigma_X^2 &= \overline{|x(n) - m_X|^2} = \lim_{N \rightarrow \infty} \left[ \frac{1}{2N+1} \sum_{i=-N}^N |x(n) - m_X|^2 \right]\end{aligned}\quad (5.36)$$

$$\phi_{xx}(k) = \overline{x(n)x^*(n-k)} = \lim_{N \rightarrow \infty} \left[ \frac{1}{2N+1} \sum_{n=-N}^N x(n)x^*(n-k) \right]$$

Il est clair que la moyenne et la variance du signal discret sont identiques à celles du signal continu. De même, les valeurs de la fonction d'autocorrélation estimées par (5.36) ne sont rien d'autre que des échantillons de la fonction d'autocorrélation du signal continu sous-jacent (Fig. 5.16).

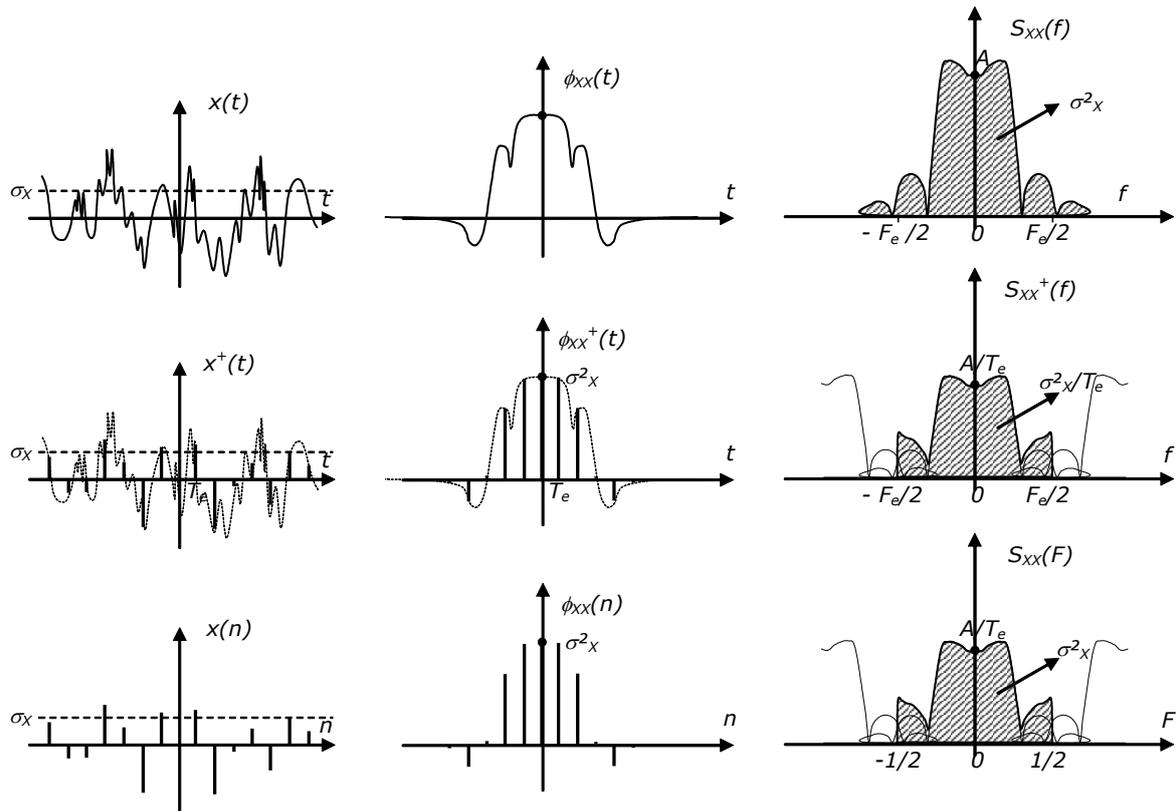


Fig. 5.16 Exemple de fonction d'autocorrélation et de densité spectrale de puissance à temps discret d'un signal aléatoire à temps discret à valeurs réelles et à moyenne nulle, comparées à celles du signal aléatoire continu sous-jacent

Dans le cas d'un signal à temps discret, la fonction d'autocorrélation est elle aussi à temps discret, et la PSD à temps discret se réduit à une TFTD :

$$S_{XX}(F) = \sum_{n=-\infty}^{\infty} \phi_{XX}(n) e^{-jn\varphi} \quad \text{avec } \varphi = 2\pi F \quad (5.37)^6$$

Il est clair de la condition de non-recouvrement d'applique ici également : la fréquence d'échantillonnage  $F_e$  doit être choisie supérieure à deux fois la fréquence maximale de la PSD  $S_{XX}(f)$  du signal continu (cette condition n'est

<sup>6</sup> On se souviendra que la notation  $S_{XX}(F)$  est en réalité une notation simplifiée pour  $S_{XX}^+(f)$  avec  $f = FF_e$ . Il ne faut en effet pas confondre la PSD du signal à temps continu,  $S_{XX}(f)$ , et celle du signal à temps discret,  $S_{XX}^+(f)$ .

De même, on parlera parfois, par abus de langage, de "PSD" pour désigner la "PSD à temps discret"  $S_{XX}(F)$ .

pas respectée dans l'exemple de la Fig. 5.16) pour que  $S_{xx}(F)$  donne une image correcte de  $S_{xx}(f)$  dans  $[-F_e/2, F_e/2]$ .

On constate également sur la Fig. 5.16 que l'amplitude de la PSD à temps discret  $S_{xx}^+(f)$  n'est pas égale à celle de la PSD du signal continu. Un facteur  $1/T_e$  est apparu, comme toujours lorsqu'on échantillonne un signal continu.

La TFTD inverse de la densité spectrale de puissance est bien entendu :

$$\phi_{xx}(n) = \int_{-1/2}^{1/2} S_{xx}(F) e^{jn\omega} dF \quad (5.38)$$

En particulier, on notera que, pour un signal de moyenne nulle, la variance (ou puissance) est donnée par la surface sous-tendue par  $S_{xx}(F)$  dans la bande de fréquences utile  $[-1/2, 1/2]$  (Fig. 5.16) :

$$\sigma_x^2 = \phi_{xx}(0) = \int_{-1/2}^{1/2} S_{xx}(F) dF \quad (5.39)$$

### Exemple 5.8 – Bruit blanc à temps discret

Un bruit blanc à temps discret (que nous supposerons de moyenne nulle) est un signal aléatoire à temps discret dont les échantillons successifs sont des variables aléatoires non corrélées. Il s'ensuit que la fonction d'autocorrélation est nulle partout sauf en 0 :

$$\phi_{xx}(k) = \begin{cases} \sigma_x^2 & \text{pour } k = 0 \\ 0 & \text{pour } k \neq 0 \end{cases} \quad (5.40)$$

La densité spectrale de puissance d'un bruit blanc est donc une constante :

$$S_{xx}(F) = \sigma_x^2 \quad (5.41)$$

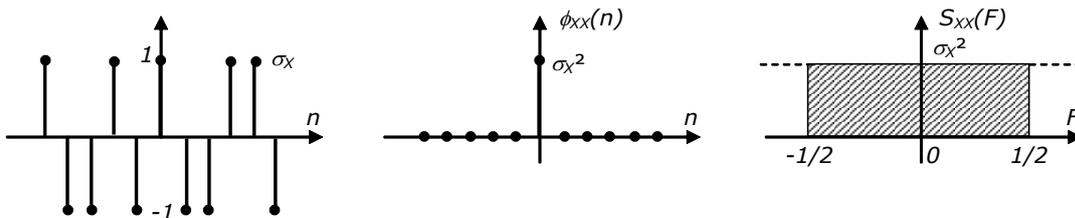


Fig. 5.17 Fonction d'autocorrélation et densité spectrale de puissance (à temps discret) d'un bruit blanc à temps discret (ici un bruit blanc binaire)

Les fonctions Matlab déjà mentionnées pour le tirage des variables aléatoires sont donc directement utilisables pour produire les bruits blancs correspondants.

```
subplot(2,1,1) ; plot(rand(1,300)); title('bruit blanc uniforme (0,1)');
subplot(2,1,2) ; plot(randn(1,300)); title('bruit blanc gaussien (0,1)');
```

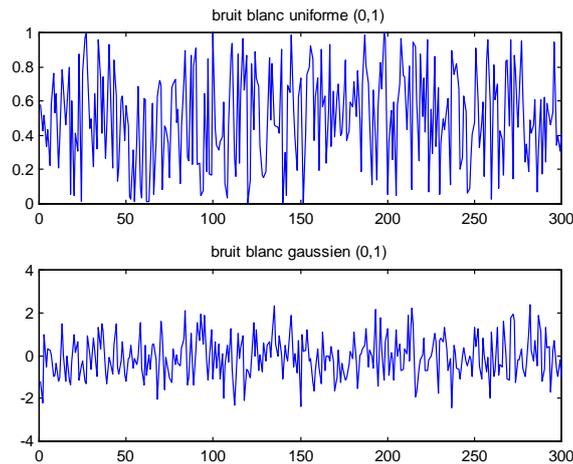


Fig. 5.18 Tirage de bruits blancs

**Exemple 5.9 – Signal périodique à temps discret**

Les conclusions de l'exemple 5.5 sont directement transposables aux signaux périodiques à temps discret. La fonction d'autocorrélation du signal numérique correspond à l'échantillonnage de la fonction d'autocorrélation du signal continu. La densité spectrale de puissance à temps discret est un spectre de raies périodique de période  $F_e$ . L'amplitude des raies dépend de la variable choisie comme abscisse. En particulier, les raies apparaissent sur  $S_{xx}(F)$  avec la même amplitude que sur  $S_{xx}(f)$  (Fig. 5.19).

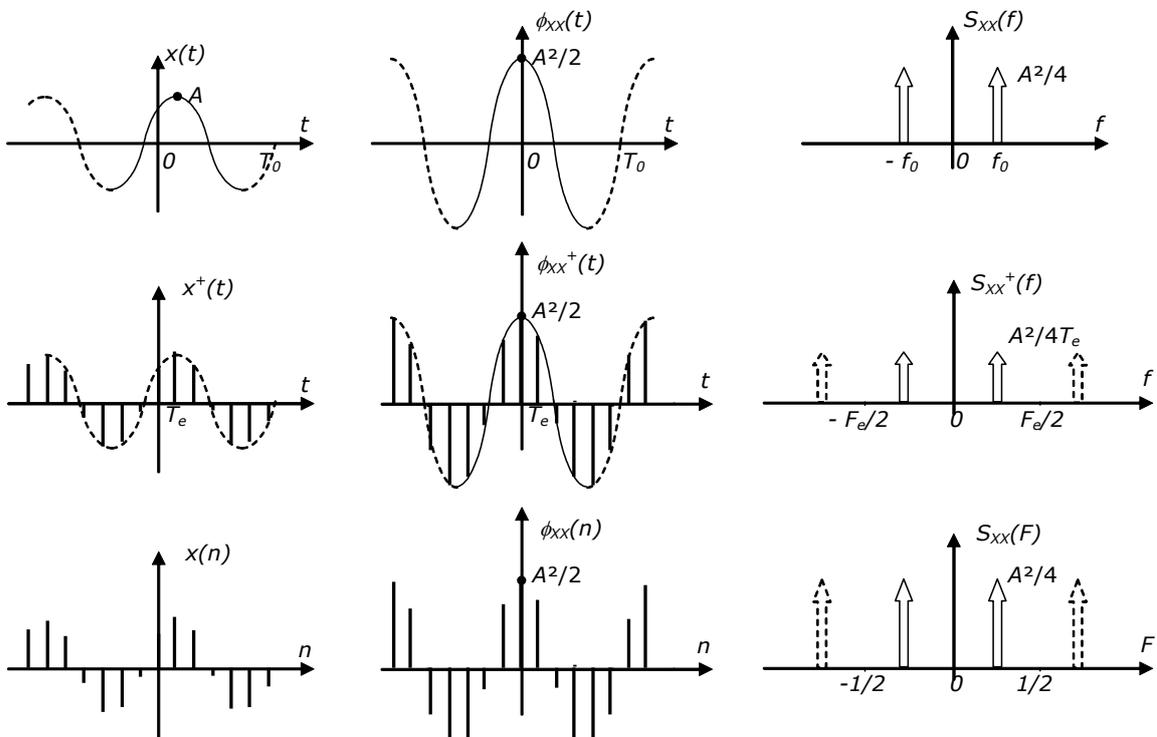


Fig. 5.19 Exemple de fonction d'autocorrélation et de densité spectrale de puissance à temps discret d'un signal périodique, comparées à celles du signal continu sous-jacent

En particulier, pour un signal cisoïdal :

$$\begin{aligned}\phi_{xx}(k) &= \frac{A^2}{2} \cos(n\varphi_0) \quad (\varphi_0 = \omega_0 / f_e) \\ S_{xx}(F) &= \frac{A^2}{4} \delta(F - iF_0) + \frac{A^2}{4} \delta(F + iF_0) \quad (\text{pour } 0 < F (= f / f_e) < 1)\end{aligned}\tag{5.42}$$

**On retiendra donc que, que ce soit pour un signal périodique ou aléatoire, (5.39) est toujours valable.**

### 5.3.1 Application : Quantification uniforme d'un signal numérique

On a étudié au chapitre 3 l'échantillonnage des signaux analogiques, qui conduit à l'obtention de signaux à temps discret. Par la suite, on a souvent utilisé abusivement le terme de *signal numérique* pour désigner le résultat de l'échantillonnage. En réalité, un signal numérique est forcément représenté, dans la mémoire d'un ordinateur, à l'aide d'un nombre fini de bits. La plupart du temps, les signaux numériques sont stockés sous la forme de nombre entiers (représentation dite *en virgule fixe*), obtenus par *quantification uniforme* du signal à temps discret (Fig. 5.20).

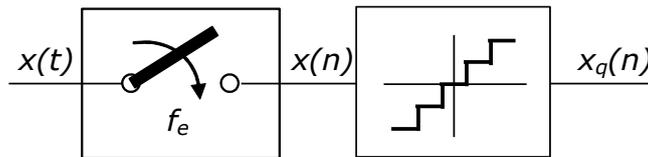


Fig. 5.20 Numérisation d'un signal analogique

L'opération de quantification uniforme sur  $N$  bits consiste à diviser un intervalle  $[-A, +A]$  en  $2^N$  intervalles de même longueur  $q = 2A/2^N$  appelée *pas de quantification*. On associe alors (non-linéairement) à chaque échantillon  $x(n)$  une valeur  $x_q(n)$  représentée par un nombre binaire sur  $N$  bits. Cette opération revient à remplacer toutes les valeurs de  $x(n)$  dans un intervalle donné  $(kq - q/2, kq + q/2)$  par une valeur unique  $x_q(n)$ , typiquement la valeur centrale de l'intervalle :  $kq$ .

L'erreur commise, appelée *erreur de quantification*, est donc donnée par (Fig. 5.21):

$$e(n) = x_q(n) - x(n) \quad -q/2 \leq e(n) \leq q/2 \tag{5.43}$$

On peut considérer cette erreur comme un bruit qui vient s'ajouter au signal à temps discret, d'où le nom usuel de *bruit de quantification* (Fig. 5.24).

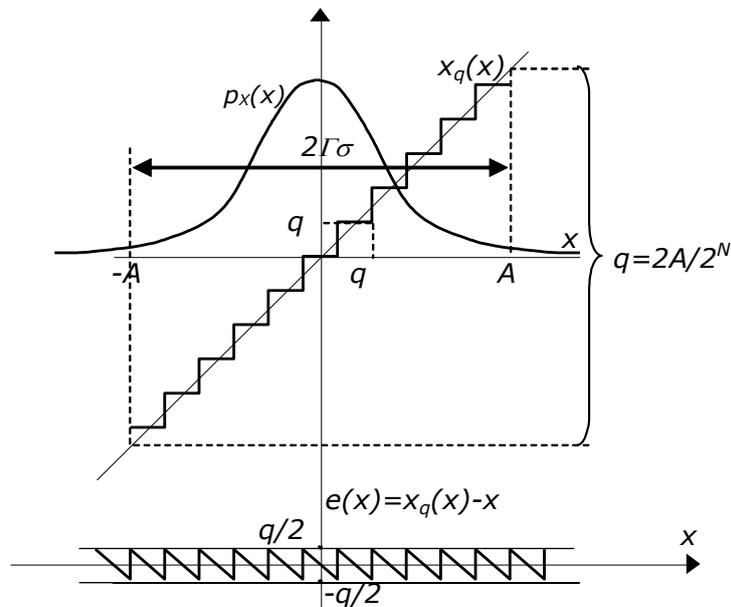
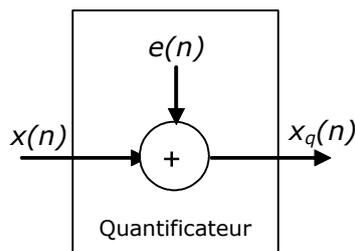
Fig. 5.21 Quantification uniforme d'une v.a.  $x$ 

Fig. 5.22 La quantification vue comme un bruit additif

Il est intéressant de chercher à quantifier l'importance de ce bruit. Supposons que le signal soumis à quantification soit tel que :

- Entre 2 instants d'échantillonnage, le signal à quantifier traverse plusieurs intervalles de quantification (il suffit pour cela que l'écart-type du signal vaille au moins quelques pas de quantification ; Fig. 5.23).
- La probabilité que l'amplitude du signal dépasse  $A$  (dite *probabilité de dépassement*) soit négligeable.

Dans ces conditions, il est clair que le signal d'erreur peut être assimilé à un bruit blanc uniforme entre  $-q/2$  et  $+q/2$ . Sa moyenne est donc nulle, et sa variance est donnée par :

$$\sigma_e^2 = \frac{q^2}{12} = \frac{A^2}{3 \cdot 2^{2N}} \quad (5.44)$$

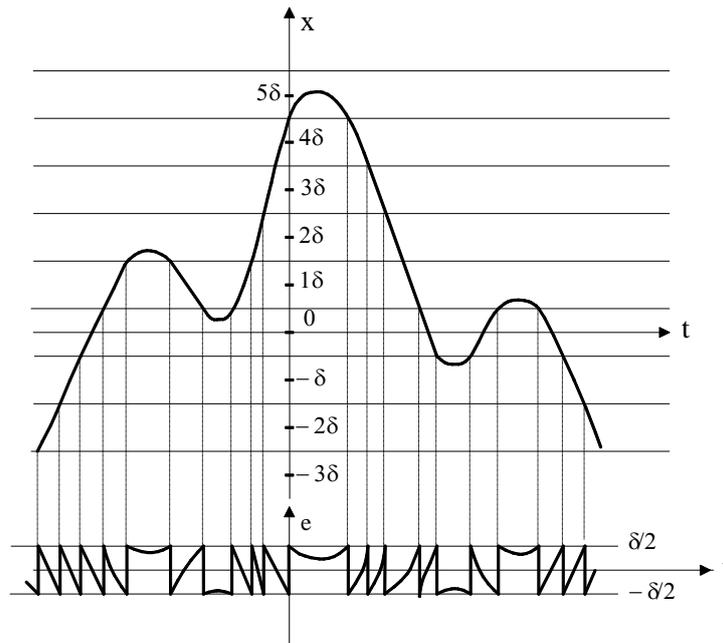


Fig. 5.23 Quantification d'un signal dont l'écart-type vaut plusieurs fois le pas de quantification (noté ici  $\delta$ )

Le *rapport signal à bruit*, noté *RSB* est défini par le logarithme du rapport de la variance du signal à celle du bruit :

$$RSB = 10 \cdot \log \left( \frac{\sigma_x^2}{\sigma_e^2} \right) \text{ dB} \quad (5.45)$$

Si l'on désigne par  $\Gamma$  le *facteur de charge* défini par :

$$\Gamma = \frac{A}{\sigma_x}$$

on obtient :

$$RSB = 10 \log \left( \frac{1}{\Gamma^2} 3 \cdot 2^{2N} \right) \approx 6,02N + 4,44 - 20 \cdot \log \Gamma \quad (5.46)$$

Cette expression est illustrée à la Fig. 5.24 pour quelques valeurs de  $N$ . Elle montre bien que *chaque bit de quantification apporte 6 dB supplémentaire de RSB*. La valeur  $N=16$  correspond à la quantification adoptée usuellement pour les signaux audio (comme pour le CD audio). Elle permet d'obtenir des RSB allant jusqu'à 90 dB. On remarque également sur le graphique une chute brutale du RSB lorsque  $\Gamma$  diminue (c.-à-d. lorsque  $-20 \log \Gamma$  augmente) : l'hypothèse de non dépassement n'est alors plus vérifiée, et la variance du bruit augmente de façon très importante.

En pratique, si la variable à quantifier est de type Gaussien, le facteur de charge doit être supérieur ou égal à 3 ( $A=3\sigma$ ) pour éviter le dépassement. Cette valeur est à adapter à la statistique du signal dans le cas non Gaussien.

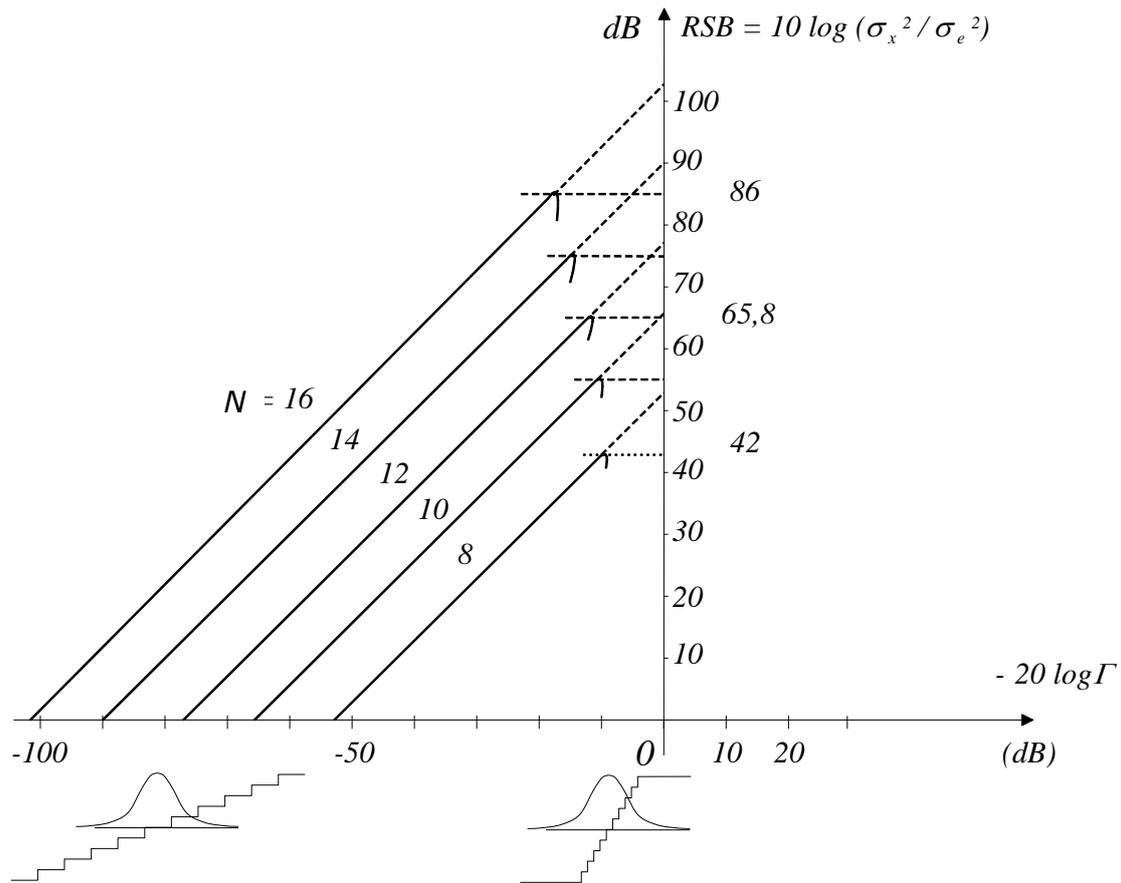


Fig. 5.24 RSB pour la quantification uniforme d'un signal avec un pas de quantification petit par rapport à son écart-type (hypothèse de non-dépassement)

### 5.4 Filtrage d'un signal aléatoire SSL

Soit un signal aléatoire  $\{x(n)\}$  stationnaire placé en entrée d'un filtre numérique de réponse impulsionnelle  $\{h(n)\}$  :

$$y(n) = \sum_{i=-\infty}^{\infty} h(i)x(n-i) \tag{5.47}$$

En vertu de (5.10), la densité de probabilité du signal  $y(n)$  résulte de la convolution d'un grand nombre de densités de probabilités similaires (les coefficients de la réponse impulsionnelle agissant comme facteurs d'échelle). Elle est donc en toute généralité différente de départ, et tend vers une distribution gaussienne en vertu du théorème central limite. En particulier, si  $x(n)$  est un bruit gaussien,  $y(n)$  l'est aussi.

Le signal de sortie  $\{y(n)\}$  est stationnaire. Sa moyenne et sa fonction d'autocorrélation sont données par :

$$m_y = E[Y(n)] = \sum_{i=-\infty}^{\infty} h(i)E[X(n-i)] = m_x \sum_{i=-\infty}^{\infty} h(i) \tag{5.48}$$

$$\begin{aligned} \phi_{YY}(k) &= E[Y(n)Y^*(n-k)] = E\left[\sum_{i=-\infty}^{\infty} h(i)X(n-i) \sum_{j=-\infty}^{\infty} h(j)X^*(n-k-j)\right] \\ &= \sum_{i=-\infty}^{\infty} h(i) \sum_{j=-\infty}^{\infty} h(j) E[X(n-i)X^*(n-k-j)] \\ &= \sum_{i=-\infty}^{\infty} h(i) \sum_{j=-\infty}^{\infty} h(j) \phi_{XX}(-k-j+i) \end{aligned}$$

On peut mettre ce dernier résultat sous une forme plus simple :

$$\begin{aligned} \phi_{YY}(k) &= \sum_{i=-\infty}^{\infty} h(i) \sum_{l=-\infty}^{\infty} h(i-l) \phi_{XX}(l-k) \quad \text{avec } l = i - j \\ &= \sum_{l=-\infty}^{\infty} \phi_{XX}(l-k) \sum_{i=-\infty}^{\infty} h(i)h(i-l) \\ &= \sum_{k=-\infty}^{\infty} \phi_{XX}(l-k) \phi_{HH}(l) \\ &= \sum_{k=-\infty}^{\infty} \phi_{XX}(k-l) \phi_{HH}(l) \end{aligned} \tag{5.49}$$

La fonction d'autocorrélation de la sortie est donc la convolution entre la fonction d'autocorrélation de l'entrée et la fonction  $\phi_{HH}(k)$  donnée par:

$$\phi_{HH}(k) = \sum_{i=-\infty}^{\infty} h(i)h(i-k) = \{h(k)\} * \{h(-k)\} \tag{5.50}$$

On en conclut que *la fonction d'autocorrélation du signal de sortie est le résultat du filtrage de la fonction d'autocorrélation de l'entrée par la « fonction d'autocorrélation » de la réponse impulsionnelle.*

La TFTD de  $\phi_{HH}(k)$  vaut :

$$S_{HH}(F) = H(F)H^*(F) = |H(F)|^2 \tag{5.51}$$

D'où l'expression simple de la densité spectrale de puissance de la sortie (Fig. 5.25) :

$$S_{YY}(F) = |H(F)|^2 S_{XX}(F)$$
(5.52)

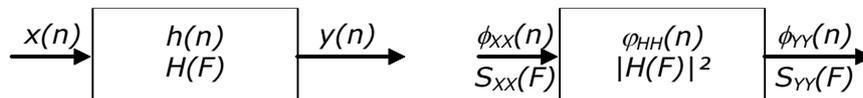


Fig. 5.25 Filtrage d'un signal aléatoire

L'exemple qui suit montre comment on peut se servir du filtrage pour créer un bruit *coloré* (c'est-à-dire un bruit de densité spectrale de puissance non constante) à partir d'un bruit blanc.

**Exemple 5.10 – Filtrage d'un bruit blanc par un passe-bande idéal**

Dans le cas particulier où le signal  $x(n)$  est un bruit blanc de variance  $\sigma_x^2$  et de moyenne nulle, il vient :

$$S_{YY}(F) = \sigma_x^2 |H(F)|^2$$

$$\sigma_y^2 = \varphi_{YY}(0) = \sigma_x^2 \int_{-1/2}^{+1/2} |H(F)|^2 dF \quad (5.53)$$

Si le filtre est un passe-bande idéalisé dont la bande passante normalisée vaut  $B$  ( $B < 1$ ), on trouve donc (Fig. 5.26):

$$\sigma_y^2 = 2\sigma_x^2 B \quad (5.54)$$

La variance du signal de sortie est donc plus faible que celle du signal d'entrée puisque le filtre élimine une partie du spectre du bruit.

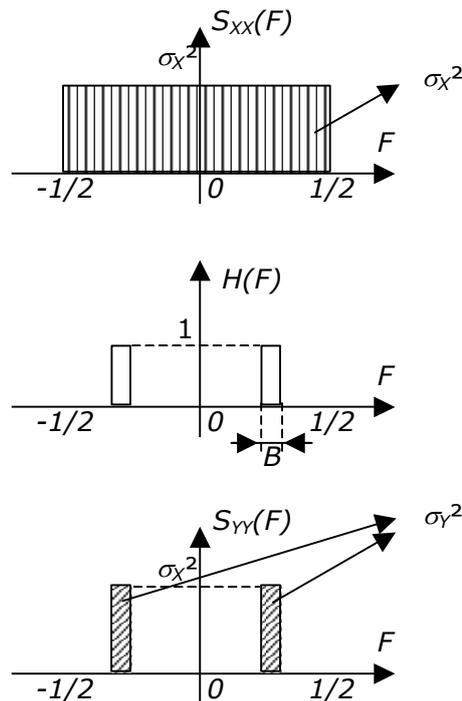


Fig. 5.26 Densité spectrale de puissance avant et après filtrage

### Exemple 5.11 – Filtrage du bruit de quantification – Augmentation du SNR

Lorsqu'on passe un signal numérique (et quantifié) à travers un filtre, on modifie en général le rapport signal sur bruit, puisque la puissance du signal et celle du bruit sont en général modifiées de façon différentes.

Ainsi par exemple, considérons une sinusoïde échantillonnée sur  $N$  bits avec un facteur de charge unitaire (c'est-à-dire que la valeur maximale en sortie du quantificateur est choisie égale à l'amplitude de la sinusoïde). Le RSB correspondant est de :

$$RSB_{in} = 10 \log\left(\frac{A^2/2}{q^2/12}\right) \quad (5.55)$$

où  $q$  est la valeur du pas de quantification ( $q = 2A/2^N$ ).

Si on passe cette sinusoïde à travers le filtre précédent, et qu'on suppose que la fréquence de la sinusoïde est dans la bande passante du filtre, le RSB à la sortie du filtre est de :

$$\begin{aligned}
 RSB_{in} &= 10 \log\left(\frac{1}{2B} \frac{A^2/2}{q^2/12}\right) \\
 &= RSB_{in} - 10 \log(2B)
 \end{aligned}
 \tag{5.56}$$

et comme  $B$  est inférieur à 1, la RSB a augmenté.

## 5.5 Estimation de la statistique du second ordre

Tout comme la TFTD, l'autocorrélation et la densité spectrale de puissance supposent des calculs sur un nombre infini d'échantillons. En pratique, les signaux aléatoires sont rarement stationnaires sur une durée infinie, et les calculs se font donc toujours à partir d'*estimateurs* opérant dans une fenêtre de  $N$  valeurs.

### 5.5.1 Statistique d'un estimateur

Il est important de connaître l'effet d'une troncation à  $N$  échantillons sur la qualité de l'estimation. Comme les signaux sont aléatoires, les estimateurs le sont aussi. Ils sont donc eux-mêmes caractérisés par leur moyenne, variance, et densité de probabilité.

Lorsque la moyenne d'un estimateur est différente de la valeur qu'il estime, on dit que l'estimateur est *biaisé*. Lorsque la variance de l'estimateur ne peut être rendue aussi petite que l'on veut (même en augmentant le nombre de points sur lesquels se base l'estimation), on dit que l'estimateur est *non consistant*. Il est clair qu'on cherche en général des estimateurs non biaisés et consistants.

La densité de probabilité de l'estimateur permet également d'associer un intervalle de confiance à l'estimation qu'il fournit.

#### Exemple 5.12 – Estimation de la moyenne d'un bruit blanc de moyenne non-nulle

Considérons un exemple simple : l'estimation de la moyenne d'un signal aléatoire gaussien stationnaire dont la densité spectrale de puissance est supposée constante (bruit blanc) excepté la présence d'une raie en  $F = 0$  due précisément à la composante moyenne :

$$S_{XX}(F) = \sigma_x^2 + \mu_x^2 \delta(F) \quad |F| \leq 1 \tag{5.57}$$

L'estimateur  $\hat{\mu}_x$  sera donné par :

$$\hat{\mu}_x = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \tag{5.58}$$

Cette expression est une variable aléatoire car c'est une somme de variables aléatoires. L'estimateur est non biaisé car :

$$E[\hat{\mu}_x] = E\left[\frac{1}{N} \sum_{n=0}^{N-1} x(n)\right] = \frac{1}{N} \sum_{n=0}^{N-1} E[x(n)] = \mu_x \tag{5.59}$$

Quant à la variance de l'estimateur, elle est donnée par :

$$\text{VAR}[\hat{\mu}_x] = E\left[(\hat{\mu}_x - E[\hat{\mu}_x])^2\right] = E\left[(\hat{\mu}_x - \mu_x)^2\right] \tag{5.60}$$

$$\begin{aligned}
 \text{VAR}[\hat{\mu}_x] &= E \left[ \left( \left( \frac{1}{N} \sum_{n=0}^{N-1} x(n) \right) - \mu_x \right)^2 \right] \\
 &= \frac{1}{N^2} E \left[ \left( \left( \sum_{n=0}^{N-1} x(n) \right) - N\mu_x \right)^2 \right] \\
 &= \frac{1}{N^2} E \left[ \left( \sum_{n=0}^{N-1} (x(n) - \mu_x) \right)^2 \right]
 \end{aligned} \tag{5.61}$$

Puisque  $x(n) - \mu_x$  est un bruit blanc de moyenne nulle, les variables aléatoires correspondant aux échantillons successifs sont indépendantes. L'espérance mathématique de leur produit est donc nulle, et il vient :

$$\begin{aligned}
 \text{VAR}[\hat{\mu}_x] &= \frac{1}{N^2} E \left[ \sum_{n=0}^{N-1} (x(n) - \mu_x)^2 \right] \\
 &= \frac{1}{N^2} N\sigma_x^2 = \frac{1}{N} \sigma_x^2
 \end{aligned} \tag{5.62}$$

On observe qu'en l'occurrence, la variance de l'estimateur tend vers zéro lorsque le nombre d'échantillons tend vers l'infini: cet estimateur est consistant.

Sous Matlab (en prenant ici un bruit blanc Gaussien, par exemple):

```

x= randn(100,100)+1 ; % 100 échantillons de 100 bruits blancs gaussiens (1,1)
for N=2:100
    mux= sum(x(1:N ,:))/N ; % somme de N bruits blancs
    meanmux (N)=mean (mux) ;
    varmux (N)=std (mux) *std (mux) ;
end
subplot (2,1,1) ; plot (meanmux) ; title ('moyenne (mux)') ;
subplot (2,1,2) ; plot (varmux) ; title ('variance (mux)') ;

```

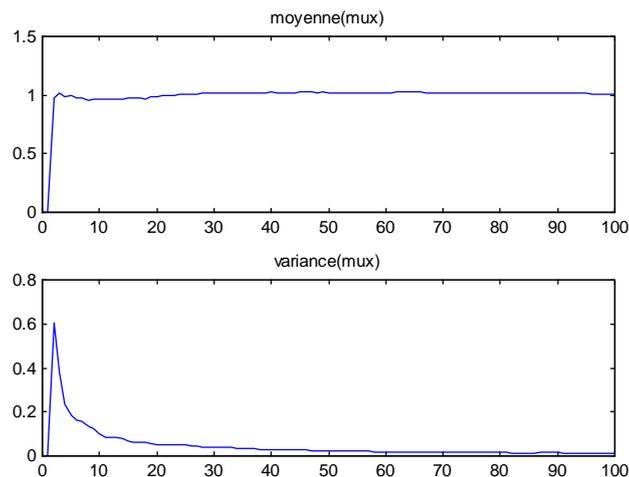


Fig. 5.27 Calcul approché de la moyenne et de la variance de l'estimateur  $\hat{\mu}_x$ , en fonction de  $N$ .

On peut, pour terminer, associer un intervalle de confiance à  $\hat{\mu}_x$ . Cet estimateur est en effet une gaussienne de moyenne  $\mu_x$  et de variance  $\sigma_x^2/N$ . La vraie moyenne  $\mu_x$  se trouve donc dans l'intervalle  $[\mu_x - d, \mu_x + d]$  avec une probabilité de 90% si :

$$\text{prob} \left[ -\alpha \leq \frac{\hat{\mu}_x - \mu_x}{\sigma_x / \sqrt{N}} \leq \alpha \right] = 0,9 \quad d = \alpha \cdot \sigma_x / \sqrt{N} \quad (5.63)$$

L'intervalle de confiance à 90% sera  $[\hat{\mu}_x - d, \hat{\mu}_x + d]$ , ce qui signifie qu'il y a 90% de chances pour que cet intervalle contienne la vraie valeur  $\mu_x$ . Le paramètre  $\alpha$  est obtenu en tenant compte de ce que :

$$\frac{\hat{\mu}_x - \mu_x}{\sigma_x / \sqrt{N}} \text{ est une v.a. gaussienne } N[0,1] \Rightarrow \alpha = 1.645 \quad (5.64)$$

## 5.5.2 Estimation de la fonction d'autocorrélation

### Estimateur non biaisé

Un premier estimateur est défini comme suit :

$$\bar{\phi}_{xx}(k) = \frac{1}{N - |k|} \cdot \sum_{n=0}^{N-|k|-1} x(n)x^*(n-k) \quad (5.65)$$

On peut vérifier<sup>7</sup> que  $E[\bar{\phi}_{xx}] = \phi_{xx}$  : cet estimateur n'est donc pas biaisé. Pour  $k \ll N$ , on montre que sa variance vaut :

$$\text{VAR}[\bar{\phi}_{xx}] \approx \frac{N}{(N - |k|)^2} \cdot \sum_l [\phi_{xx}^2(l) + \phi_{xx}(l+k)\phi_{xx}(l-k)] \quad (5.66)$$

et comme elle varie comme  $1/N$ , l'estimateur est consistant. Par contre, la variance dépend de  $k$  : si  $k$  est proche de  $N$ , le nombre de points sur lequel est estimée la fonction d'autocorrélation est très petit, et la variance de l'estimateur est donc tout naturellement grande.

### Estimateur biaisé

On peut définir un autre estimateur :

$$\hat{\phi}_{xx}(k) = \frac{1}{N} \cdot \sum_{n=0}^{N-|k|-1} x(n)x^*(n-k) \quad (5.67)$$

qui est lié au précédent par :

$$\hat{\phi}_{xx}(k) = \frac{N - |k|}{N} \bar{\phi}_{xx}(k) \quad (5.68)$$

et pour lequel on a donc :

$$E[\hat{\phi}_{xx}(k)] = \frac{N - |k|}{N} \phi_{xx}(k) \quad (5.69)$$

D'autre part, on montre que la variance est donnée par :

---

<sup>7</sup> Nous avons traité complètement le cas de l'estimateur de la moyenne d'un signal Gaussien. Nous ne donnerons plus que les résultats finaux pour le traitement des estimateurs de la fonction de l'autocorrélation et de la densité spectrale de puissance. Les calculs sont effets assez fastidieux.

$$\text{VAR}[\hat{\phi}_{XX}(k)] \approx \frac{1}{N} \cdot \sum_l [\phi_{XX}^2(l) + \phi_{XX}(l+k)\phi_{XX}(l-k)] \quad (5.70)$$

Ce nouvel estimateur est donc *biaisé* (surtout pour les grandes valeurs de  $k$ ), mais reste *consistant*.

### Exemple 5.13 – Estimation de la fonction d'autocorrélation d'une cisoïde

OIN a vu que la fonction d'autocorrélation d'une cisoïde d'amplitude  $A$  est un cosinus de même fréquence et d'amplitude  $A^2/2$ . Il est facile de le retrouver numériquement, en utilisant l'estimateur non-biaisé. L'estimateur biaisé, quant à lui, fait clairement apparaître la pondération de l'estimateur biaisé par la fenêtre triangulaire (5.68).

```
x= 10*cos(2*pi*1/10*(0:99)+pi/5) ;
% 100 échantillons d'un cosinus à Fe/10, d'amplitude 10 et de phase initiale pi/5
subplot(2,1,1) ; plot(xcorr(x, 'unbiased'));
title('autocorrélation sur 100 points, estimateur non-biaisé');
subplot(2,1,2) ; plot(xcorr(x, 'biased'));
title('autocorrélation sur 100 points, estimateur biaisé');
```

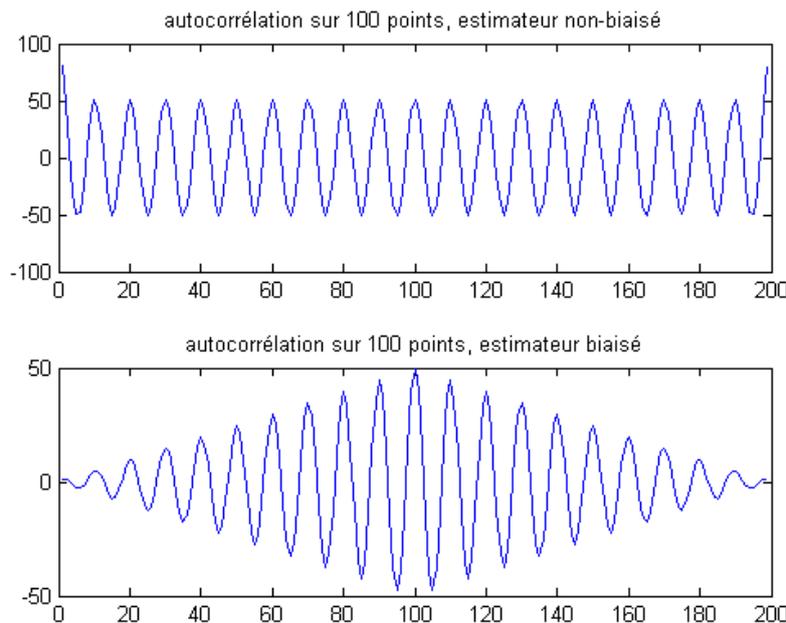


Fig. 5.28 Calcul de  $\hat{\phi}_{XX}(k)$  (estimateur biaisé) et  $\bar{\phi}_{XX}(k)$  (non-biaisé) sur 100 points d'un cisoïde.

### Exemple 5.14 – Estimation de la fonction d'autocorrélation d'un bruit blanc Gaussien

Considérons l'exemple d'un bruit blanc Gaussien stationnaire de moyenne nulle et de variance égale à 4 :

$$S_{XX}(F) = 4 \quad (5.71)$$

$$\phi_{XX}(k) = 4\delta(k)$$

Sous Matlab, les deux estimateurs de la fonction d'autocorrélation sont appelés par `xcorr(x, 'unbiased')` et `xcorr(x, 'biased')` :

```
x= 2*randn(1,100) ; % 100 échantillons d'un bruit blanc gaussien (0,2)
```

```
subplot(2,1,1) ; plot(xcorr(x, 'unbiased'));
title('autocorrélation sur 100 points, estimateur non-biaisé');
subplot(2,1,2) ; plot(xcorr(x, 'biased'));
title('autocorrélation sur 100 points, estimateur biaisé');
```

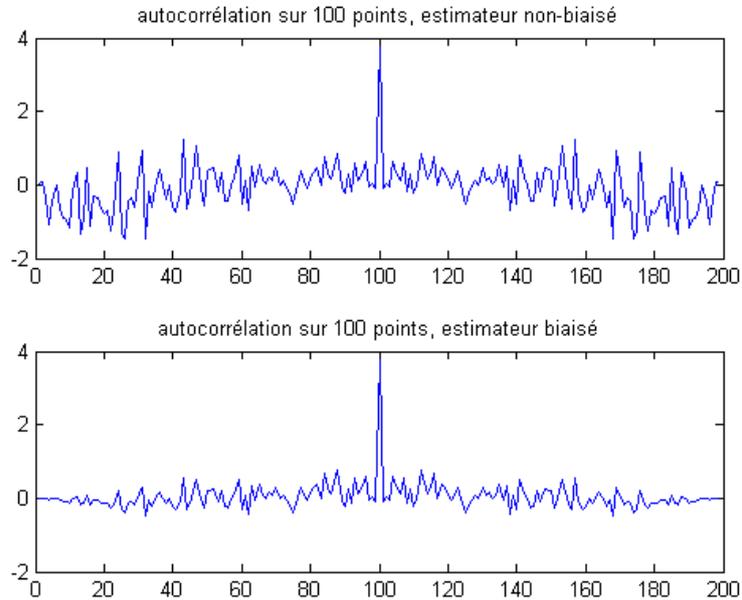


Fig. 5.29 Calcul de  $\bar{\phi}_{xx}(k)$  (non-biaisé) et  $\hat{\phi}_{xx}(k)$  (estimateur biaisé) sur 100 points d'un bruit blanc.

On observe que lorsque  $k$  approche du nombre d'échantillons  $N$ , la variance de l'estimateur non biaisé  $\bar{\phi}_{xx}(k)$  devient excessive, alors que celle de l'estimateur biaisé  $\hat{\phi}_{xx}(k)$  varie beaucoup moins. C'est une des raisons pour lesquelles cet estimateur est souvent utilisé par la suite, malgré son biais.

Une deuxième raison est qu'il est possible de calculer (5.67) par FFT. Il apparaît en effet que :

$$\hat{\phi}_{xx}(k) = \frac{1}{N} \{x_N(n)\} * \{x_N^*(-n)\} \tag{5.72}$$

où  $\{x_N(n)\} = \{x(0), x(1), \dots, x(N-1), 0, 0, \dots\}$ . On peut donc calculer l'autocorrélation comme une convolution numérique linéaire. On a vu au chapitre 3 qu'il est possible de le faire de façon efficace en simulant une convolution circulaire par FFT/IFFT.

Enfin, comme nous allons le voir au point suivant, l'estimateur biaisé permet un calcul très rapide de la densité spectrale de puissance, par le biais d'une FFT.

### 5.5.3 Estimation de la densité spectrale de puissance

#### Le périodogramme

L'estimateur spectral le plus simple est le *périodogramme* qui correspond à la transformée de Fourier d'une estimation de la fonction d'autocorrélation, à l'aide de l'estimateur biaisé  $\hat{\phi}_{xx}(k)$  :

$$\hat{S}_{XX}(F) = \sum_{k=-(N-1)}^{N-1} \hat{\phi}_{XX}(k) e^{-jk\varphi} \quad \text{avec } \varphi = 2\pi F \quad (5.73)$$

Si on remplace  $\hat{\phi}_{XX}(k)$  par son expression dans (5.72), on trouve immédiatement:

$$\hat{S}_{XX}(F) = \frac{1}{N} |X_N(F)|^2 \quad (5.74)$$

où  $X_N(F)$  est la TFTD de la suite  $x(n)$  :

$$X_N(F) = \sum_{n=0}^{N-1} x(n) e^{-jn\varphi} \quad \text{avec } \varphi = 2\pi F \quad (5.75)$$

Par conséquent, pour  $\varphi = k/N$  ( $k=0,1,\dots,N-1$ ), la densité spectrale peut donc être calculée par un algorithme de FFT qui fournit  $X_N(F)$  en  $N$  points uniformément répartis entre  $F = 0$  et  $1$ .

Toutefois, le périodogramme est un estimateur biaisé, puisqu'il est lui-même basé sur un estimateur biaisé :

$$E[\hat{S}_{XX}(F)] = \sum_{k=-(N-1)}^{N-1} E[\hat{\phi}_{XX}(k)] e^{-jk\varphi} = \sum_{k=-(N-1)}^{N-1} \frac{N-|k|}{N} \phi_{XX}(k) e^{-jk\varphi} \quad (5.76)$$

Cette expression représente la TFTD de la fonction d'autocorrélation  $\phi_{XX}(k)$  vue à travers une fenêtre triangulaire de durée  $2N-1$ ; en fréquence, cela correspond à une convolution entre la vraie densité spectrale de puissance et la TFTD de cette fenêtre triangulaire.

D'autre part, on montre que, dans le cas particulier d'un bruit blanc Gaussien de variance  $\sigma_x^2$ , on trouve :

$$\text{VAR}[\hat{S}_{XX}(F)] = S_{XX}^2(F) \quad (5.77)$$

Cette variance, égale au carré de la densité spectrale, est très élevée et elle est indépendante du nombre  $N$  d'échantillons ! Le périodogramme simple n'est donc pas consistant et il n'est utilisé que lorsque le nombre d'échantillons dont on dispose pour l'estimation est faible. Dans le cas contraire, on lui préfère le périodogramme moyenné (voir plus bas).

Notons que, tout comme dans le cas du calcul de la TFTD d'un signal déterministe, il est possible de pondérer les échantillons par une fonction fenêtre  $w(n)$  sur  $N$  points avant la TFTD. L'estimateur spectral doit alors être calculé comme :

$$\ddot{S}_{XX}(F) = \frac{1}{NP} \left| \sum_{n=0}^{N-1} x(n) w(n) e^{-jn\varphi} \right|^2 \quad \text{avec } \varphi = 2\pi F \quad (5.78)$$

où  $P$  est un facteur de normalisation (destiné à éliminer le biais introduit par la fenêtre) défini par:

$$P = \frac{1}{N} \sum_{n=0}^{N-1} w(n)^2 \quad (5.79)$$

Ce nouvel estimateur est tout aussi biaisé que le précédent (où la fenêtre appliquée sur les échantillons avant calcul de la FFT est simplement

rectangulaire). Le choix de la fenêtre est cependant important lorsque la densité spectrale de puissance du signal étudié comprend des raies superposées à un bruit large bande. L'effet des fenêtres est similaire à ce qui avait déjà été mis en évidence au chapitre 4.

### Le périodogramme moyenné – Estimateur de Welsh

Pour réduire la variance d'un estimateur, il suffit, en principe, de calculer la moyenne de plusieurs estimateurs identiques mais indépendants. Le signal de durée  $N$  est donc décomposé en  $L$  segments de durée  $M=N/L$ ; pour chacun de ces segments, on calcule par FFT un périodogramme simple (avec ou sans fenêtre de pondération). On calcule alors la moyenne des estimations pour chaque fréquence  $F = k/M$ . Les différentes tranches considérées peuvent se recouvrir ou non. On a donc :

$$\bar{S}_{XX}(F) = \frac{1}{L} \cdot \sum_{i=1}^L \ddot{S}_{XX,i}(F) \quad (5.80)$$

avec

$$\ddot{S}_{x,i}(F) = \frac{1}{MP} \left| \sum_{n=0}^{M-1} x_i(n) w(n) e^{-jn\varphi} \right|^2 \quad i = 1, 2, \dots, L \quad (5.81)$$

Si on le compare au précédent, le biais introduit par cet estimateur est plus important, puisque la fenêtre de pondération est plus courte (sa TFTD est donc plus large). On peut montrer par contre que sa variance est donnée par:

$$\text{VAR}[\bar{S}_{XX}(F)] \approx \frac{1}{L} S_{XX}^2(F) \quad (5.82)$$

si l'on peut considérer les  $L$  sections comme indépendantes. En conséquence, *cet estimateur est consistant*.

Pour  $N$  fixé, il y a malgré tout conflit entre une grande valeur de  $L$  qui diminue la variance mais conduit à une pauvre résolution fréquentielle car  $M$  diminue.

#### Exemple 5.15 – Bruit blanc

Sous Matlab, on peut calculer un périodogramme simple très facilement. Considérons le cas du bruit blanc Gaussien  $N(0,2)$  de l'exemple précédent (Fig. 5.31).

```
x= 2*randn(1,512) ; % 512 échantillons d'un bruit blanc gaussien (0,2)
plot(10*log10( abs(fft(x,512)).^2 /512 )) ; % périodogramme simple
```

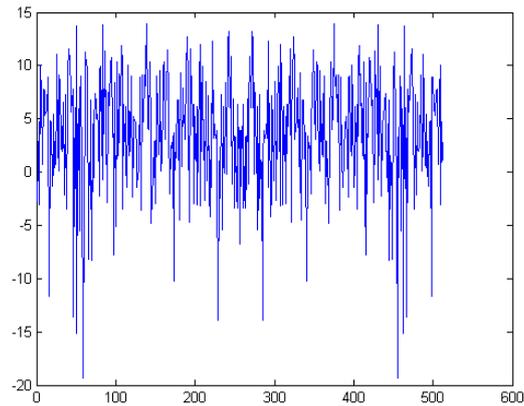


Fig. 5.30 Périodogramme simple d'un bruit blanc de variance 4, calculé sur  $N_{FFT}=512$  points dans l'intervalle  $F=[0,1]$ .

Il est cependant plus pratique, pour le périodogramme moyenné (et même pour le périodogramme simple) d'utiliser par la fonction `pwelch(x,window,NOVERLAP,NFFT,Fe)`, qui découpe le signal  $x$  en tranches de  $N_{FFT}$  échantillons, avec un recouvrement de  $NOVERLAP$  échantillons entre tranches, et une pondération par le fenêtre  $window$ , et affiche la moyenne de leur FFT selon (5.80). Comme pour `freqz`, un appel à `pwelch` sans variable de sortie provoque l'affichage de la densité spectrale de puissance, en dB ( $=10 \log_{10}(S_{xx})$  !)

Le résultat est cependant multiplié par un facteur correctif égal à  $2*Te$ . Dans ce facteur, le rôle de  $Te$  est de renvoyer une estimation de la PSD du signal analogique sous-jacent (on a vu en effet à la Fig. 5.16 que le passage en numérique divise la PSD par  $Te$ ). Ce facteur correctif permet donc à la fonction `pwelch` de calculer, non pas la PSD  $S_{xx}(F)$  du signal échantillonné, mais bien la PSD  $S_{xx}(f)$  du signal analogique de départ. Le facteur 2 est utilisé par MATLAB pour ne renvoyer que la PSD  $S_{xx}(f)$  sur  $[0,Fe/2]$  tout en faisant en sorte que son intégrale sur  $[0,Fe/2]$  soit égale à  $\sigma_x^2$ . **On retiendra que, pour calculer (5.80) par `pwelch`, il suffit de compenser ce facteur en imposant artificiellement  $Te=1/2$  (c.-à-d.  $Fe=2$ ).**

Appliquons cette fonction au signal  $x$  (Fig. 5.31). Pour le périodogramme simple, on constate une grande dispersion des valeurs autour de la valeur vraie (une constante égale à  $10\log_{10}(4)=6\text{dB}$ ), qui ne diminue pas lorsque  $N_{FFT}$  augmente. Le calcul du périodogramme avec pondération par une fenêtre lisse un peu la variation du spectre mais ne réduit pas son amplitude. Lorsque le même bruit blanc est estimé au moyen du périodogramme moyenné (ici avec une fenêtre rectangulaire), on observe une nette diminution de l'amplitude des variations de l'estimation.

```
subplot(2,2,1) ; pwelch(x(1:128),boxcar(128),0,128,2); % périodogramme simple8
subplot(2,2,2) ; pwelch(x(1:512),boxcar(512),0,512,2); % périodogramme simple
subplot(2,2,3) ; pwelch(x(1:128),hamming(128),0,128,2); % périodogramme avec fenêtre
subplot(2,2,4) ; pwelch(x,hamming(128),0,128,2); % périodogramme moyenné avec fenêtre

subplot(2,2,1) ; ax=axis; subplot(2,2,2) ; axis(ax);
subplot(2,2,3) ; axis(ax); subplot(2,2,4) ; axis(ax); % axes identiques
```

<sup>8</sup> On spécifie ici, comme expliqué plus haut,  $Fe=2$ . Il faudra donc interpréter l'axe des fréquences affiché  $f=[0,1]$  comme correspondant à une fréquence normalisée  $F=[0,1/2]$ .

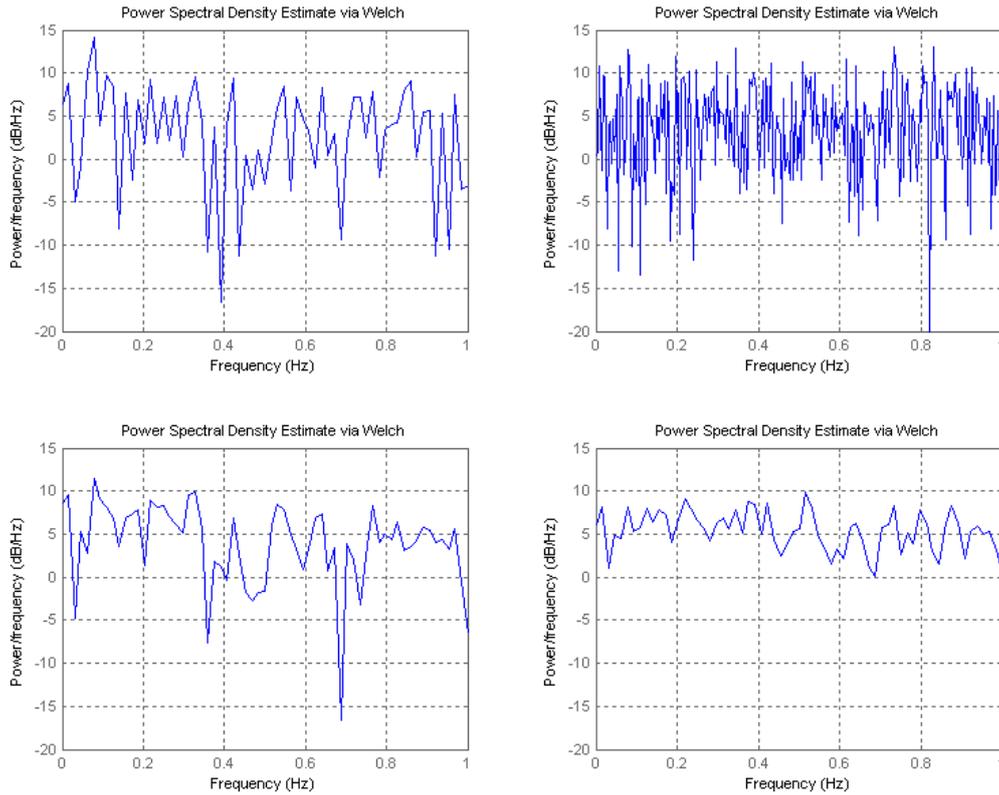


Fig. 5.31 Périodogrammes d'un bruit blanc. En haut à gauche : simple ( $N=128$ ) ; en haut à droite : simple ( $N=512$ ) ; en bas à gauche : simple avec pondération par une fenêtre de Hamming ( $N=128$ ) ; en bas à droite : moyenné :  $N=128$  et  $L=4$ .<sup>9</sup>

Notons que, par défaut, `pwelch` divise le vecteur d'échantillons  $x$  en 8 tranches avec un recouvrement de 50%, une fenêtre de Hamming, et un choix de NFFT fonction de la largeur des tranches. On peut donc le plus souvent laisser faire la fonction, en lui spécifiant simplement (où `[]` signifie que MATLAB utilisera ses valeurs par défaut):

```
pwelch(x, [], [], [], 2)
```

<sup>9</sup> Dans toutes les figures qui suivent, il faut interpréter l'axe des fréquences affiché  $f=[0,1]$  comme correspondant à une fréquence normalisée  $F=[0,1/2]$ .

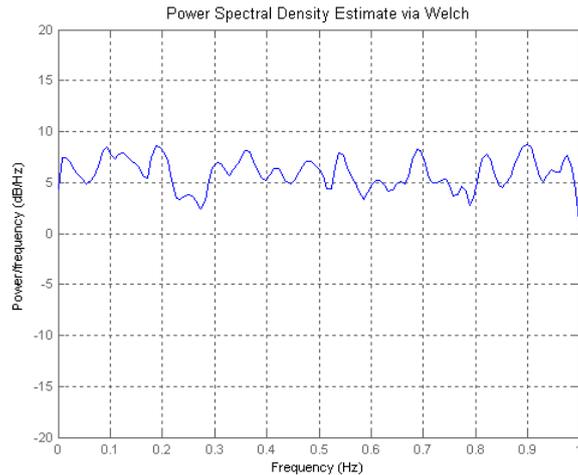


Fig. 5.32 Périodogramme moyenné d'un bruit blanc (utilisation de `pwelch` avec valeurs par défaut).

Il est clair que l'estimation de la variance du bruit peut être faite par intégration de la densité spectrale de puissance estimée, selon (5.39). Il faut alors simplement se souvenir, comme montré à la Fig. 5.36, que la FFT utilisée pour calculer la PSD dans `pwelch` ne renvoie que  $N_{FFT}/2$  échantillons de cette PSD estimée pour  $F=[0,1/2]$  (la PSD étant symétrique, ses valeurs pour les fréquences négatives ne sont pas renvoyées). Le calcul de l'intégrale nécessite donc de multiplier chaque valeur de la PSD renvoyée par le pas en fréquence  $1/N_{FFT}$  et de le multiplier à nouveau par 2, pour tenir compte de la PSD pour les fréquences négatives :

```
variance= sum(x.*x)/length(x)
variance = 4.0372
variance=sum(pwelch(x, [], [], 128, 2))/128*2
variance = 3.9154
```

On note que dans l'estimation précédente on a dû imposer  $N_{FFT}$ , pour pouvoir corriger son effet.

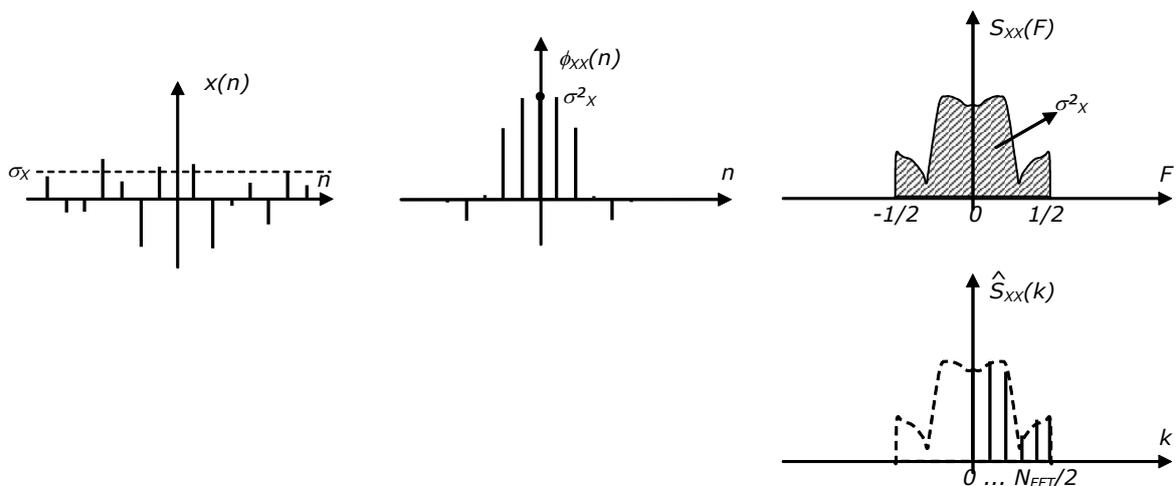


Fig. 5.33 Valeurs du périodogramme  $\hat{S}_{xx}(F)$  d'un signal aléatoire, telles que renvoyées par `pwelch` pour  $F=k/N_{FFT}$  ( $k=0,1,\dots,N_{FFT}/2$ )

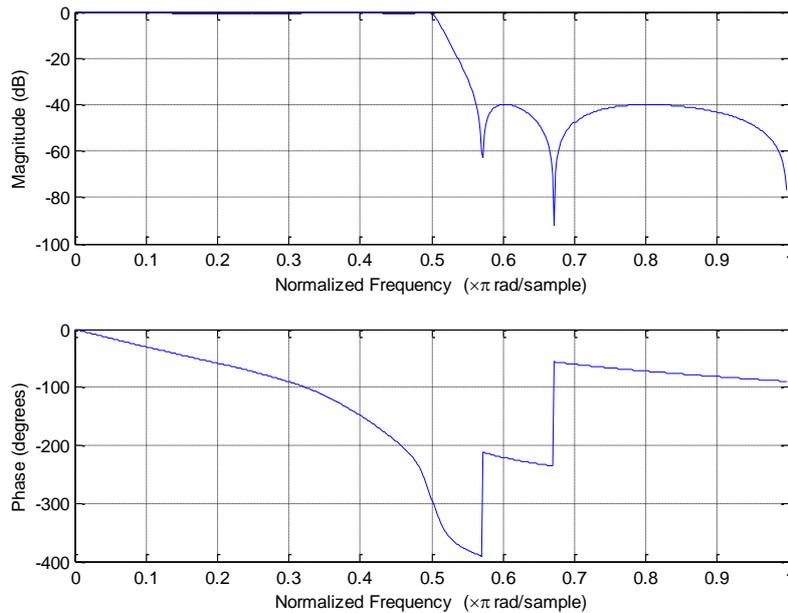
**Exemple 5.16 – Bruit coloré**

On construit un bruit coloré par filtrage du bruit blanc de l'exercice précédent. La PSD de ce bruit est estimée par différentes méthodes (Fig. 5.34) qui montrent l'influence du nombre de points et du nombre de tranches dans le cas du périodogramme moyenné.

```
x=2*randn(1,512) ; % 512 échantillons d'un bruit blanc gaussien (0,2)
[N,Wn]=ellipord(0.5,0.6,1,40) ;
[B,A]=ellip(N,1,40,Wn) ; % synthèse d'un filtre passe-bas10
freqz(B,A) % vérifier la réponse en fréquence du filtre
xf=filter(B,A,x) ; % filtrage

subplot(2,2,1) ; pwelch(xf(1:128),boxcar(128),0,128,2) ; % périodogramme simple
subplot(2,2,2) ; pwelch(xf(1:512),boxcar(512),0,512,2) ; % périodogramme simple
subplot(2,2,3) ; pwelch(xf(1:128),hamming(128),0,128,2) ; % périodogramme avec fenêtre
subplot(2,2,4) ; pwelch(xf,hamming(128),0,128,2) ; % périodogramme moyenné avec fenêtre

subplot(2,2,1) ; ax=axis; subplot(2,2,2) ; axis(ax);
subplot(2,2,3) ; axis(ax); subplot(2,2,4) ; axis(ax); % axes identiques
```



<sup>10</sup> La synthèse de filtres numériques fait l'objet d'un chapitre ultérieur.

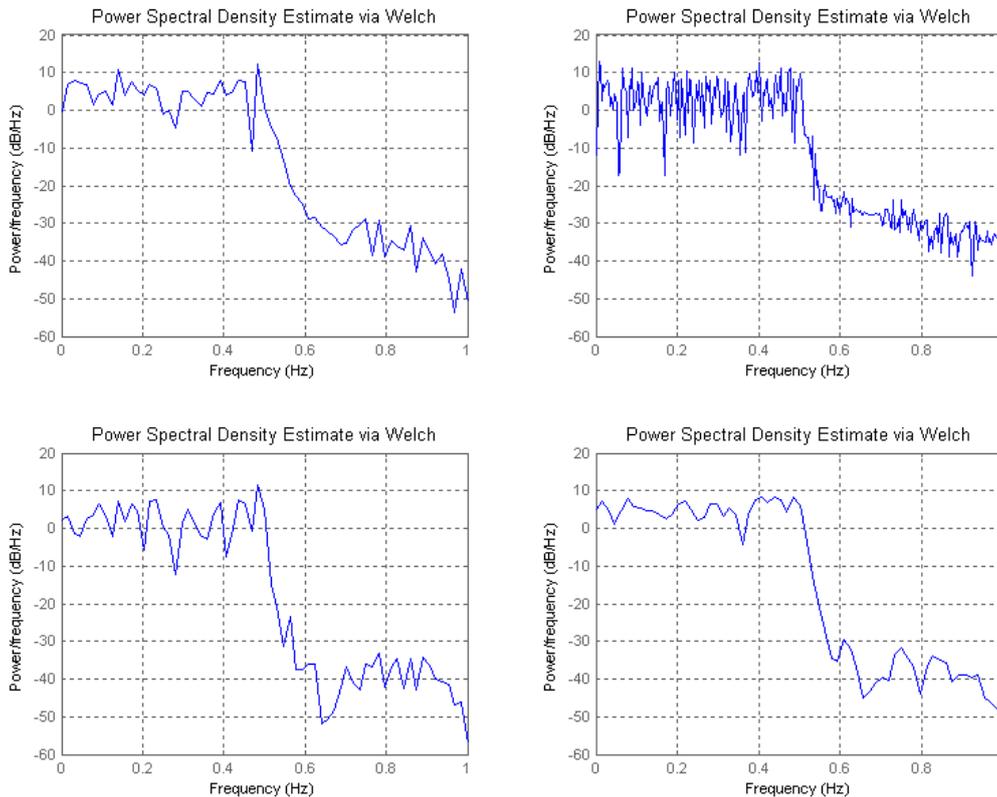


Fig. 5.34 Estimateurs d'un bruit coloré. (haut) Réponse en fréquence du filtre; (milieu gauche) périodogramme simple ( $N=128$ ); (milieu droite) périodogramme simple ( $N=512$ ); (bas gauche) périodogramme simple avec pondération par fenêtre de Hamming ( $N=128$ ); (bas droite) périodogramme moyenné ( $N=512; L=4$ )

On constate que l'utilisation d'une fenêtre de Hamming permet, grâce à ses lobes secondaires bas, de bien voir la bande atténuée à  $-40\text{dB}$ , au contraire d'une fenêtre rectangulaire. On constate également que le périodogramme moyenné est plus précis en amplitude (car sa variance est plus faible) mais moins en fréquence (car il est calculé sur moins de points en fréquence).

De même, la variance (ici à peu près 2) peut être calculée sur la PSD estimée :

```
variance= sum(xf.*xf)/length(xf)
variance = 2.0736
variance=sum(pwelch(xf, [], [], 128, 2))/128*2
variance = 1.9973
```

### Exemple 5.17 – Cisoïde

Le cas de la cisoïde d'amplitude  $A$  est particulier : sa fonction d'autocorrélation est un cosinus d'amplitude  $A^2/2$ , et sa densité spectrale de puissance devrait donc faire apparaître deux impulsions de Dirac d'amplitude  $A^2/4$ .

Comme le périodogramme et le périodogramme moyenné calculent la densité spectrale de puissance comme le carré d'une TFD divisé par  $N$  (voir par exemple (5.74)), ils sont soumis aux mêmes artifices que pour le calcul de la TFD d'une cisoïde : en lieu et place des impulsions de Dirac en  $A^2/4$ , ils font apparaître le carré des TFD des fonctions fenêtres,

divisés par  $N$ , et multiplié par  $A^2/4$ . Dans le cas du périodogramme simple, l'amplitude du lobe principal vaut  $A^2/4$  multiplié par un coefficient  $N^2/N=N$  (Fig. 5.35). <sup>11</sup>

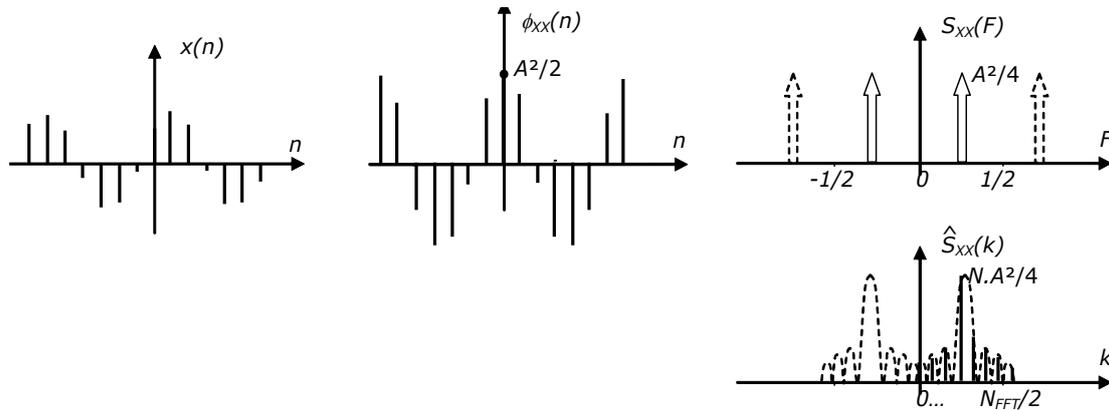


Fig. 5.35 Valeurs du périodogramme  $\hat{S}_{XX}(F)$  d'un signal cisoïdal, telles que renvoyées par `pwelch` pour  $F=k/N_{FFT}$  ( $k=0,1,\dots,N_{FFT}/2$ ), et pour une fenêtre rectangulaire.

D'une manière générale, si on pondère par une fenêtre, le coefficient multiplicatif vaut :

$$\frac{1}{NP} \left( \sum_{n=0}^{N-1} w(n) \right)^2 = \frac{\left( \sum_{n=0}^{N-1} w(n) \right)^2}{\left( \sum_{n=0}^{N-1} w(n)^2 \right)} \quad (5.83)$$

ce qui donne bien  $N$  dans le cas particulier d'une fenêtre rectangulaire.

Pour le périodogramme moyenné, le coefficient multiplicatif prend la même expression que (5.83) où l'on remplace  $N$  par  $M$ , le nombre de points dans chaque tranche.

On peut vérifier tout ceci facilement sous Matlab.

```
x= 10*cos(2*pi*1/10*(0:511)+pi/5) ;
% 512 échantillons d'un cosinus à Fe/10, d'amplitude 10 et de phase initiale pi/5

subplot(2,2,1) ; pwelch(x(1:128),boxcar(128),0,128,2) ; % périodogramme simple
subplot(2,2,2) ; pwelch(x(1:512),boxcar(512),0,512,2) ; % périodogramme simple
subplot(2,2,3) ; pwelch(x(1:128),hamming(128),0,128,2) ; % périodogramme avec fenêtre
subplot(2,2,4) ; pwelch(x,hamming(128),0,128,2) ; % périodogramme moyenné avec fenêtre

subplot(2,2,1) ; ax=axis; subplot(2,2,2) ;axis(ax);
subplot(2,2,3) ;axis(ax); subplot(2,2,4) ;axis(ax); % axes identiques
```

<sup>11</sup> Il est à noter que le  $N$  mentionné ici, et apparaissant à la Fig. 5.35, fait référence au *nombre de points effectivement utilisés pour calculer le périodogramme*. Dans le cas d'un périodogramme moyenné, il faut donc remplacer  $N$  par  $M$ , le nombre de points dans chacune des  $L$  tranches du signal (avec  $M=N/L$ ).

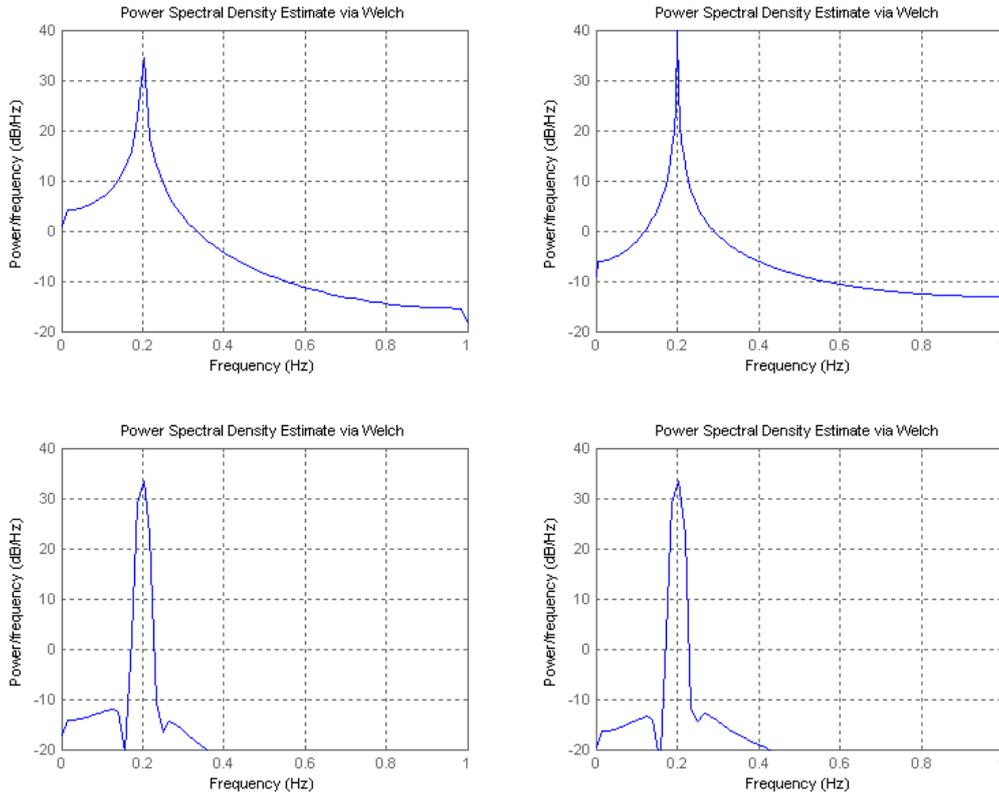


Fig. 5.36 Estimateurs de la densité spectrale de puissance d'une cisoïde d'amplitude 10 et de fréquence  $F=Fe/10$ ; (haut gauche) périodogramme ( $N=128$ ); (haut droite) périodogramme ( $N=512$ ); (bas gauche) périodogramme simple avec fenêtre de Hamming ( $N=128$ ); (bas droite) périodogramme moyenné ( $N=512$ ;  $L=4$ ; fen. de Hamming).

On retrouve bien les valeurs maximales attendues :

- Périodogramme simple (128) :  $10\log_{10}(NA^2/4)=10\log_{10}(25*128)=35$  dB
- Périodogramme simple (512) :  $10\log_{10}(25*512)=41$  dB
- Périodogramme moyenné (128 ; hamming) :

$$10\log_{10}\left(\frac{A^2/4 * \left(\sum_{n=0}^{N-1} w(n)\right)^2}{\left(\sum_{n=0}^{N-1} w(n)^2\right)}\right) = 10 * \log_{10}(25 * 93.4) = 33.6 \text{ dB}$$

Ici aussi la variance (ici 50, puisque la densité spectrale de puissance théorique est constituée de deux raies de valeur 25) peut être calculée sur la PSD estimée :

```
variance=sum(x.*x)/length(x)
variance = 50.0584
variance=sum(pwelch(x, [], [], 128, 2))/128*2
variance = 49.9994
```

### Exemple 5.18 – Cisoïdes dans du bruit blanc

Voyons enfin le cas d'une cisoïde d'amplitude 10 ( $\sigma_s^2=A^2/2=50$ ) contaminée par un bruit blanc de moyenne nulle et d'écart-type égal à 10 ( $\sigma_b^2=100$ ). Les deux signaux étant non corrélés, leurs densités spectrales de puissance s'additionnent. La Fig. 5.37 montre à nouveau bien l'influence du choix de l'estimateur et de ses paramètres.

```
x=10*randn(1,512)+10*cos(2*pi*1/10*(0 :511));
    % 512 échantillons d'un bruit blanc gaussien (0,10) + cisoïde

pwelch(x,hamming(128),0,128,2); % périodogramme moyenné avec fenêtre
```

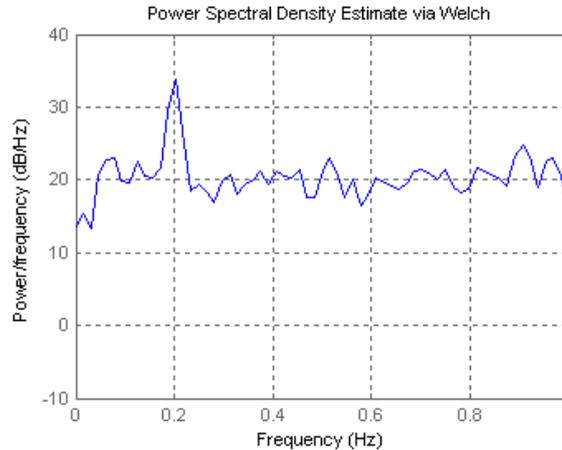


Fig. 5.37 Estimation de la densité spectrale de puissance d'un bruit blanc de variance 100 (c.-à-d. 20 dB) superposé à une cisoïde d'amplitude 10, par un périodogramme moyenné (N=512, L=4).

On peut lire sur le périodogramme moyenné de la Fig. 5.37 la valeur de  $\sigma_b^2$  (20 dB, ce qui correspond à  $\sigma_b^2=100$ ) et on peut obtenir  $\sigma_s^2=A^2/2$  en lisant sur la PSD la valeur du maximum du lobe principal correspondant à la sinusoïde (ici : 33.6 dB=  $MA^2/4$ <sup>12</sup>, ce qui donne  $A\sim 10$ , et donc  $\sigma_s^2\sim 50$ ). Le RSB mesuré correspond donc au RSB réel, soit  $10*\log_{10}(1/2)=-3$  dB.

## Exercices

### Exercice 5.1

Un bruit blanc  $N(n)$  de moyenne nulle et de puissance  $\sigma_N^2=2$  est filtré par un filtre à moyenne mobile  $y(n) = x(n) - 2x(n-1) + x(n-2)$ .

- Exprimez analytiquement  $\phi_{YY}(k)$  et déterminez ses valeurs pour  $k = -3, -2, \dots, 2, 3$ .
- Idem mais pour  $\phi_{HH}(k)$ .
- Exprimez analytiquement la densité spectrale de puissance de la sortie du filtre et esquissez-la.

### Solution

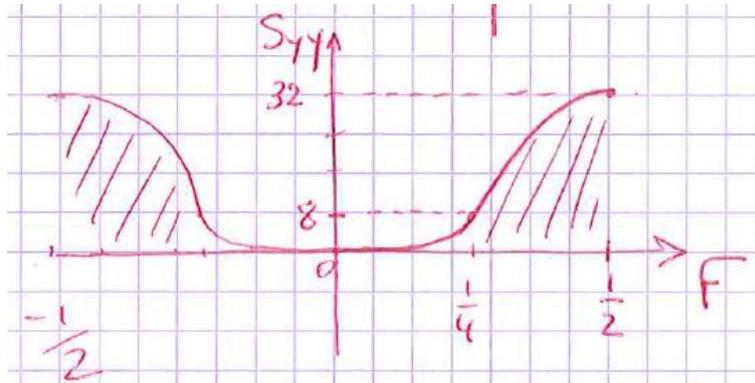
a) 
$$\phi_{YY}(k) = 2\delta(k+2) - 8\delta(k+1) + 12\delta(k) - 8\delta(k-1) + 2\delta(k-2)$$

$$\phi_{YY} = \{\dots, 0, 2, -8, 12, -8, 2, 0, \dots\}$$

<sup>12</sup> En réalité, il faudrait tenir compte du fait que, comme on utilise une fenêtre de Hamming, le maximum vaut un peu moins que  $MA^2/4$  (voir l'exemple précédent). On a négligé ici cet effet.

$$b) \quad \phi_{HH}(k) = \frac{1}{2} \phi_{YY}(k)$$

$$c) \quad S_{YY}(F) = |H(F)|^2 S_{NN}(F) = \sigma_N^2 |1 - \exp(-j2\pi F)|^2 = 8[1 - \cos(2\pi F)]^2$$

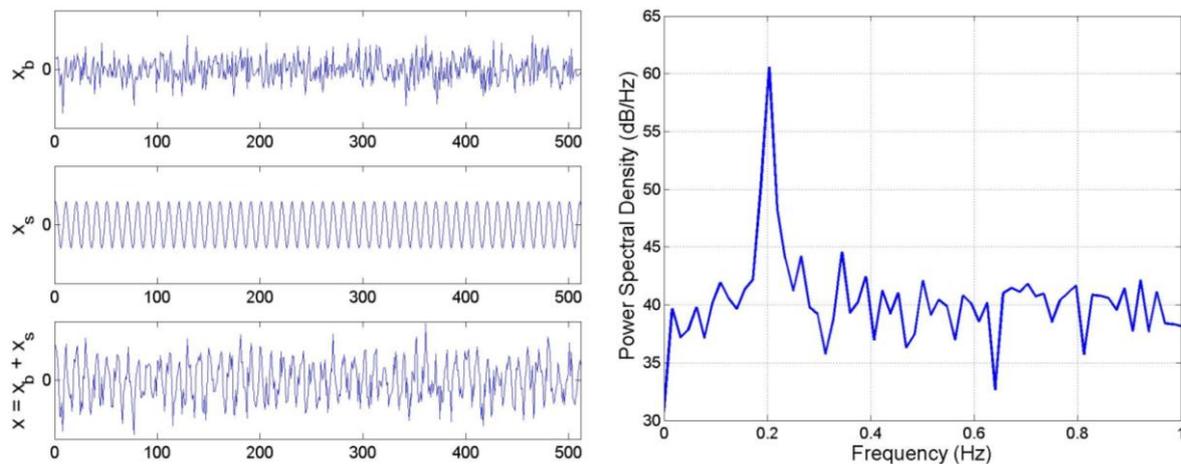


### Exercice 5.2

On considère une cisoïde d'amplitude  $A_s$  contaminée par un bruit blanc de moyenne nulle et d'écart type  $A_b$ . Ces signaux et leur somme sont représentés à la figure de gauche. Une estimation de leur densité spectrale de puissance par un périodogramme moyenné ( $N=512$ ,  $L=4$ , fenêtres rectangulaires) est représentée à la figure de droite.

On demande d'estimer  $A_s$  et  $A_b$  au moyen des informations contenues dans la figure de droite puis d'en déduire une estimation du rapport signal à bruit (RSB).

NB : comme indiqué dans le cours, « Frequency = 1 » correspond en réalité à  $F = 1/2$  où  $F$  est la fréquence normalisée



### Solution

$$\sigma_b^2 = 10^4 \Rightarrow A_b = 100$$

Avec une DSP  $\cong 61$  dB en  $F = 0.1$ , (soit « frequency » = 0.2), on trouve :

$$10 \log \frac{MA_s^2}{4} = 61 \text{ dB}$$

donc  $A_s \cong 198.3$  vu que chacune des  $L = 4$  fenêtres a 128 points ( $M = N / L$ )  
 (En réalité,  $A_s = 200$ ).

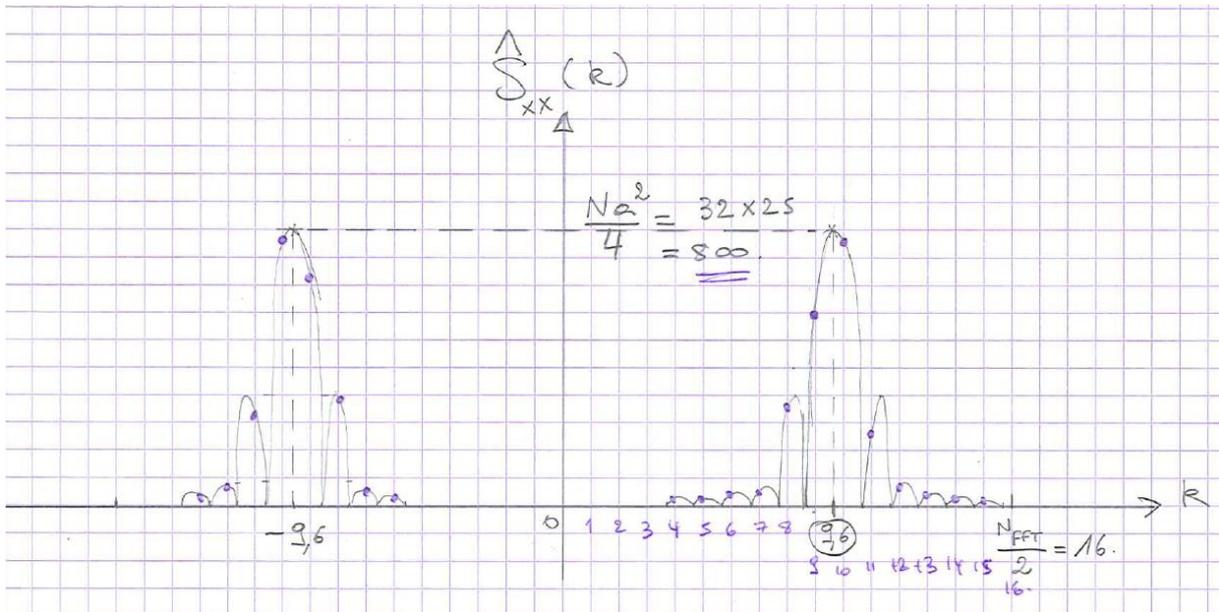
Puissance signal =  $2 \cdot 10^4$ . Puissance bruit =  $10^4$ .  $\rightarrow$  RSB = 2 (3 dB).

**Exercice 5.3**

Soit un signal  $x(t) = 10 \cos(2\pi 700 t + \pi/4)$  que l'on échantillonne à une fréquence de 1000 Hz.

On demande d'esquisser le périodogramme simple (avec fenêtre rectangulaire) obtenu en prenant 32 points du signal précédent et en faisant le calcul avec NFFT=32. Comment retrouver l'amplitude de la sinusoïde à partir de ce périodogramme ? Et sa phase ?

**Solution**



Lorsque la raie de la sinusoïde ne tombe pas sur une valeur entière de  $k$  (c'est le cas ici), on est réduit à estimer l'amplitude  $a$ .

L'information sur la phase est perdue.

**Exercice 5.4**

Soit  $x(n)$  un bruit blanc de moyenne égale à 2 et de variance unitaire. On passe ce bruit à travers un filtre dont l'équation de récurrence est :

$$y(n) = x(n) + x(n-1) + x(n-2)$$

On demande :

- a. de donner l'expression de  $\phi_{xx}(k)$  ainsi que sa représentation graphique
- b. d'exprimer analytiquement  $\phi_{yy}(k)$  en fonction de  $\phi_{xx}(k)$

- c. d'en déduire les valeurs de  $\phi_{yy}(k)$  pour  $k = -5, -4 \dots 4, 5$  et d'en donner une représentation graphique  
 d. de calculer la fonction  $\phi_{HH}(k)$  pour  $k = -5, -4 \dots 4, 5$   
 e. d'utiliser  $\phi_{HH}(k)$  et  $\phi_{xx}(k)$  pour déterminer  $\phi_{yy}(k)$

**Solution**

- a)  $\phi_{xx}(k) = 4 + \delta(k)$   
 b)  $\phi_{yy}(k) = \phi_{xx}(k+2) + 2\phi_{xx}(k+1) + 3\phi_{xx}(k) + 2\phi_{xx}(k-1) + \phi_{xx}(k-2)$   
 c)  $\phi_{yy}(k) = 36 + \delta(k+2) + 2\delta(k+1) + 3\delta(k) + 2\delta(k-1) + \delta(k-2)$   
 $\phi_{yy} = \{\dots, 36, 36, 37, 38, 39, 38, 37, 36, 36, \dots\}$   
 d)  $\phi_{HH} = \{h(n)\} * \{h(-n)\} = \{\dots, 0, 1, 2, 3, 2, 1, 0, \dots\}$   
 e) En utilisant la méthode graphique de calcul de  $\phi_{yy}(k) = \phi_{xx}(k) * \phi_{HH}(k)$  on retrouve plus rapidement  $\phi_{yy} = \{\dots, 36, 36, 37, 38, 39, 38, 37, 36, 36, \dots\}$

**Exercice 5.5**

Un signal  $x(t)$  est transmis à travers un canal de communication qui a pour effet de le multiplier par une constante  $k$ , de le retarder d'un temps  $t_0$ , et de lui ajouter un bruit  $b(t)$  :  $y(t) = k x(t-t_0) + b(t)$ . Si  $x(t)$  et  $b(t)$  sont non corrélés, exprimez la densité spectrale de puissance de  $y$  en fonction de celles de  $x$  et  $b$  ?

**Solution**

- a)  $\phi_{yy}(\tau) = k^2 \phi_{xx}(\tau) + \phi_{bb}(\tau)$   
 b)  $S_{yy}(f) = k^2 S_{xx}(f) + S_{bb}(f)$

**Exercice 5.6 - Problème dit « du confort du professeur de traitement du signal »**

Un professeur enseigne à 30 étudiants bruyants dans une salle en hémicycle. Dix étudiants se trouvent à la première rangée (à 10m du professeur), 10 autres à la seconde rangée (11m) et 10 autres à la troisième (12m). On assimile le signal de parole du professeur à un bruit blanc  $N(0, \sqrt{10} \sigma)$ , et le signal émis par chaque étudiant à un bruit blanc  $N(0, \sigma)$  (indépendant des autres). On suppose par ailleurs que la fonction de transfert correspondant à la transmission dans l'air entre la première rangée et le professeur est celle d'un filtre passe-bas idéal de fréquence de coupure  $f_c = 6000$  Hz et de gain 0.9 (deuxième rangée :  $f_c = 5000$  Hz, gain = 0.8 ; troisième rangée :  $f_c = 4000$  Hz, gain = 0.7). On suppose enfin que l'auditoire est une chambre sourde (pas de réverbération) et que personne ne bouge. On demande le rapport signal à bruit perçu par le professeur (dont l'ouïe s'étend jusque 15000 Hz).

Pour rappel,  $N(m, \sigma)$  indique un signal Gaussien de moyenne  $m$  et d'écart type  $\sigma$ .

**Solution**

$$Puissance_{professeur} = \sigma^2 \times 3.10^5 \text{ Hz [W]}$$

$$Puissance_{etudiants} = 10 \times \sigma^2 \times [0.7^2 \times 8 + 0.8^2 \times 10 + 0.9^2 \times 12] \text{ kHz [W]}$$

$$10 \log RSB = 10 \log \frac{Puissance_{professeur}}{Puissance_{etudiants}} = 1.75 \text{ dB}$$

**Exercice 5.7 (librement adapté d'un exercice de J.Auvray)**

Un signal sinusoïdal  $s$  de fréquence 500Hz et d'amplitude  $a=100\text{mV}$  est noyé dans un bruit  $b$  que l'on peut considérer comme un bruit blanc filtré par un filtre passe-bas idéal de fréquence de coupure 1kHz. La puissance totale du bruit est de  $10\text{V}^2$ .

- a) Représentez la densité spectrale de puissance de ce signal bruité  $x=s+b$ . Quel est en dB le rapport signal sur bruit de  $x$  ?

Le signal est échantillonné à 4 kHz.

- b) Représentez les densités spectrales de puissance  $S_{xx}^+(f)$  et  $S_{xx}(F)$  du signal bruité échantillonné et déterminez le rapport signal sur bruit en dB.
- c) Prendre une fréquence d'échantillonnage de 1.2 kHz modifierait-il le RSB ?

Pour sortir le signal échantillonné du bruit on utilise un filtre passe-bande idéal (fonction de transfert rectangulaire) de fréquence centrale 500 Hz et de largeur de bande 10 Hz.

- d) Représentez le spectre du signal  $y$  à la sortie du filtre et le rapport signal bruit obtenu.

Pour tenter d'améliorer les choses on effectue une détection synchrone en multipliant le signal original (non échantillonné)  $x$  par une référence d'amplitude 1V en phase avec  $s$ . Le multiplicateur effectue l'opération  $p(t) = x(t).r(t)/10$ . Le signal  $p$  à la sortie de ce multiplicateur a un spectre comportant des raies et un fond continu.

- e) Représentez d'abord les raies du spectre de ce signal non échantillonné et leur amplitude.
- f) Sachant que la fonction d'autocorrélation de  $p(t)$  est

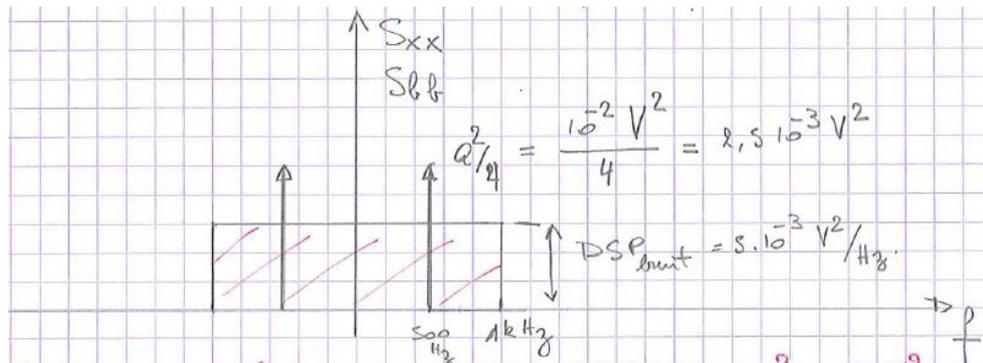
$$\phi_{pp}(\tau) = \frac{1}{100} \left[ \frac{a^2}{4} + \frac{a^2}{8} \cos 2\omega\tau + \frac{1}{2} \phi_{bb}(\tau) \cos \omega\tau \right]$$

où  $\phi_{bb}(\tau)$  est la fonction d'auto-corrélation du bruit, déterminez la densité spectrale de puissance de  $p$  que vous dessinerez en précisant les amplitudes et fréquences.

- g) Le signal  $p$  est filtré par un filtre intégrateur RC de constante de temps 10 secondes. Soit  $Q$  le signal final. Représentez la densité spectrale de  $Q$ .
- h) Quel est maintenant le rapport signal sur bruit en dB en précisant bien ce qu'est le signal à cette étape-ci.

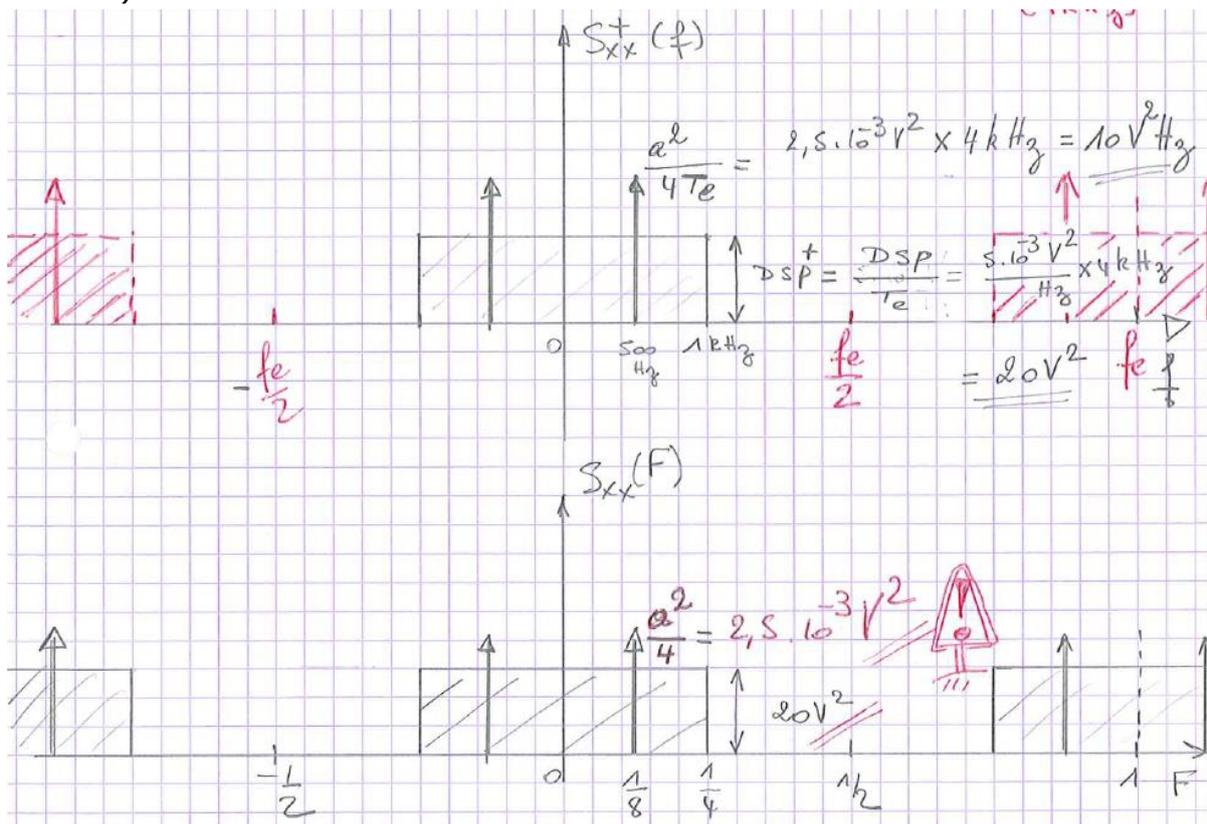
**Solution**

a)



-33 dB

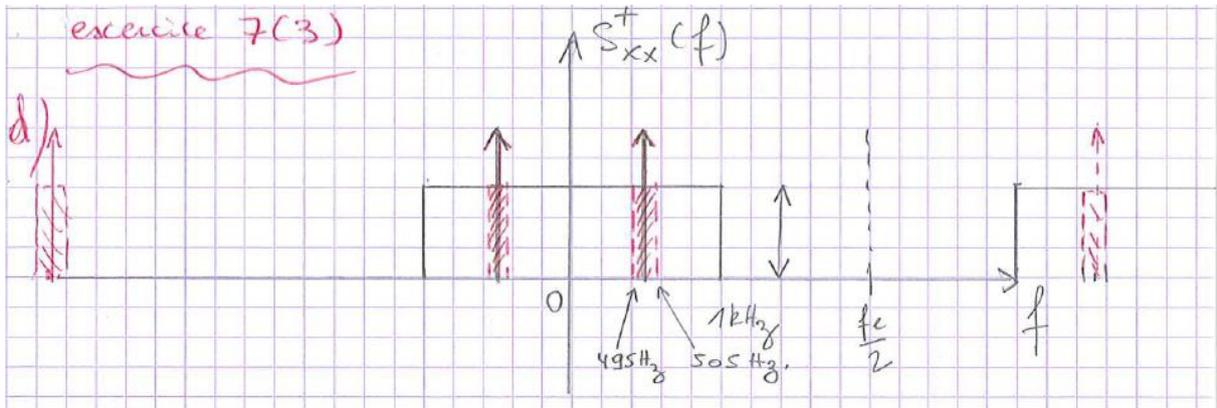
b)



Le RSB ne change pas.

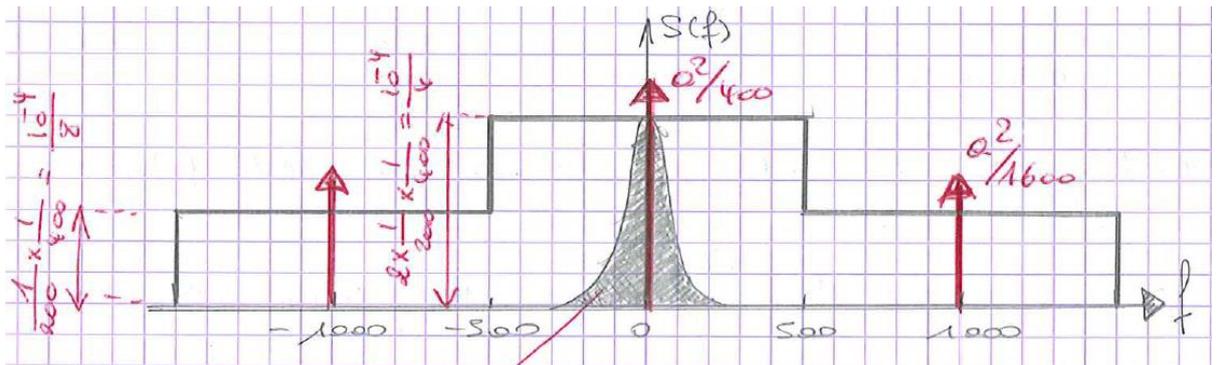
c) Bien que l'on observerait un recouvrement dans ce cas, le RSB ne changerait pas.

d)



-13 dB

e)



(les 3 raies ne sont pas représentées avec la même échelle)

f)

$$S_{pp}(f) = \frac{1}{100} \left[ \frac{a^2}{4} \delta(f) + \frac{a^2}{16} [\delta(f - 2f_0) + \delta(f + 2f_0)] + \frac{1}{4} [S_{bb}(f - f_0) + S_{bb}(f + f_0)] \right]$$

avec  $f_0 = 500$  Hz

$$g) X_{filtre}(f) = \frac{1}{1 + j2\pi f RC} \quad \int_{-\infty}^{\infty} |X_{filtre}|^2 df = \frac{1}{2RC} = 0.05 \text{ Hz}$$

Le filtre RC n'affecte pas la composante continue (raie en 0 Hz).

En très bonne approximation (en négligeant la contribution des intervalles  $]-\infty, -500$  Hz] et  $[500 \text{ Hz}, +\infty[$ , ce que l'on peut faire vu la constante de temps  $RC = 10$  s), la surface hachurée (puissance du bruit après filtrage)

$$\text{est } \frac{10^{-4}}{4} \times \frac{1}{2RC} = \frac{10^{-5}}{8}.$$

La puissance du signal (raie) en  $f=0$  est  $\frac{a^2}{400} = \frac{10^{-4}}{4}$ .

Donc  $10\log RSB = 13 \text{ dB}$ .

NB : sur la figure, la DSP du bruit après filtrage a été dilatée pour une meilleure représentation.

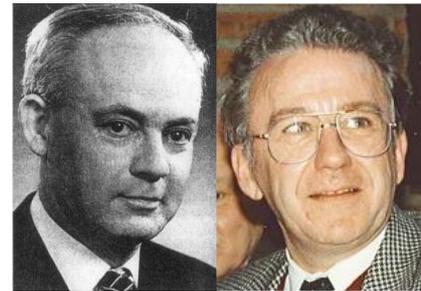
# CHAPITRE 6

## SYNTHESE DES FILTRES NUMERIQUES

Thomas W. Parks and James H. McClellan ont marqué le traitement du signal au début des années 70, alors que J. McClellan était encore étudiant à Rice University (Houston, Texas), par l'invention d'une technique d'approximation numérique de filtres FIR à phase linéaire. Connue aujourd'hui sous le nom d'algorithme de Parks-McClellan, ce procédé est très largement utilisé dans les systèmes de télécommunications numériques.



R. Boite (1928-2002) et H. Leich (1941-), professeurs émérites de la FPMs, ont fortement contribué au développement des techniques de filtrage numérique au début des années 80, notamment par la publication de leur ouvrage "Les filtres numériques" (Masson, Paris)



Avec la FFT, le filtrage numérique est un des outils majeurs du traitement du signal. Si les propriétés temporelles et fréquentielles des filtres numériques linéaires ont été *analysées* aux Chapitres 1 et 2, il reste à voir comment on les *approxime* pour obtenir une réponse en fréquence donnée avec un filtre IIR (6.2) ou un FIR (6.3) et comment on les *synthétise* (c.-à-d. comment on les implémente en pratique). Nous commençons (6.1) par le cas particulier des filtres du second degré, qui sont fort utilisés pour construire des filtres plus complexes.

### 6.1 La cellule du second degré

Pour des applications peu exigeantes, on pourra utiliser un filtre d'ordre un, deux, ou trois. Les propriétés de ces filtres en fonction de leurs coefficients ont été largement étudiées dans la littérature, et sont résumée par exemple dans l'article "*The Swiss Army Knife of Digital Networks*" de R. Lyons et A. Bell (IEEE Signal Processing Magazine, Mai 2004, pp. 90-100).

Nous nous contenterons ici d'étudier une cellule de base du second degré et deux exemples d'application.

La transmittance rationnelle du second degré :

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (6.1)$$

Si on pose (Fig. 6.1) que la forme générale des pôles et des zéros est donnée par :

$$\text{racines} = \underbrace{-\sigma}_{\text{partie réelle}} \pm j \underbrace{\omega}_{\text{partie imaginaire}} \quad (6.2)$$

$$\rho = \text{module} \quad (6.3)$$

$$\sigma = -\rho \cos \theta$$

Il vient :

$$H(z) = K \frac{1 - 2\rho_n \cos \theta_n z^{-1} + \rho_n^2 z^{-2}}{1 - 2\rho_d \cos \theta_d z^{-1} + \rho_d^2 z^{-2}} \quad (6.4)$$

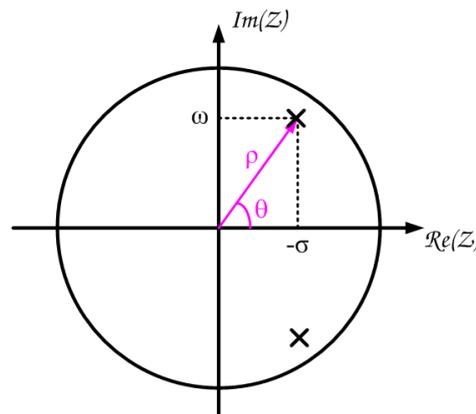


Fig. 6.1 Lien entre  $\sigma, \omega, \rho$ , et  $\theta$

Cette forme caractérise une *cellule biquadratique* ou *cellule du second degré*, très utilisée en pratique.

### 6.1.1 Stabilité

L'expression (6.1) peut être mise sous une forme utilisée classiquement en théorie des circuits analogiques :

$$H(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2} = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + 2\sigma z + \rho^2} \quad (6.5)$$

La stabilité impose des conditions sur la valeur des coefficients du dénominateur:

- si  $a_2 > a_1^2 / 4$  ( $\rho^2 > \sigma^2$ ), les pôles de  $H(z)$  sont complexes et il faut que l'on ait  $a_2 < 1$  ( $\rho < 1$ )

- si  $a_2 \leq a_1^2/4$ , les pôles sont réels. On obtient alors, après quelques développements partant du fait que le module des racines doit être inférieur à 1, qu'il faut que  $(1-a_1+a_2)$  et  $(1+a_1+a_2)$  soient positifs.

Ces conditions de stabilité sont illustrées à la Fig. 6.2.

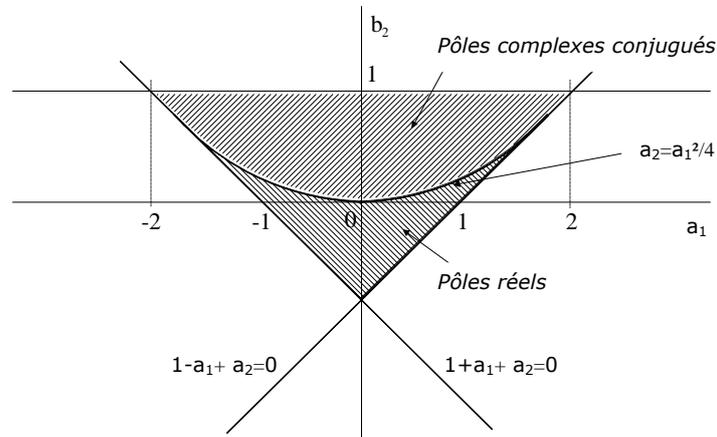


Fig. 6.2 Domaine de stabilité (ordre 2)

### 6.1.2 Le résonateur du second ordre

La transmittance (6.1) avec pôles complexes situés dans le voisinage du cercle unité, et dans laquelle  $b_2 = 0$  (ce qui correspond à un seul zéro, réel), caractérise le résonateur du second ordre (Fig. 6.3).

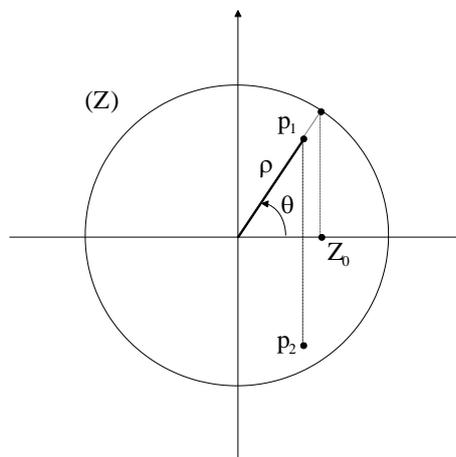


Fig. 6.3 Pôles et zéros d'un résonateur du second ordre

La fonction de transfert est donnée par :

$$H(z) = K \frac{1+bz^{-1}}{1+a_1z^{-1}+a_2z^{-2}} = K \frac{1+bz^{-1}}{(1-p_1z^{-1})(1-p_2z^{-1})} \quad (6.6)$$

avec  $p_{1,2} = \rho e^{\pm j\theta}$      $a_1 = -2\rho \cos \theta$      $a_2 = \rho^2$

Ou plus simplement :

$$H(z) = K \frac{1 - \cos \theta z^{-1}}{1 - 2\rho \cos \theta z^{-1} + \rho^2 z^{-2}} \quad (6.7)$$

Le calcul de l'amplitude conduit alors à :

$$\begin{aligned} A^2(\varphi) &= H(e^{j\varphi})H(e^{-j\varphi}) \\ &= K^2 \cdot \frac{1 + b^2 + 2b \cos \varphi}{\left[1 + \rho^2 - 2\rho \cos(\varphi - \theta)\right] \cdot \left[1 + \rho^2 - 2\rho \cos(\varphi + \theta)\right]} \end{aligned} \quad (6.8)$$

Si l'on suppose que le zéro est placé comme sur la Fig. 6.3 <sup>1</sup>, on a  $b = -\cos \theta$ , ce qui conduit à :

$$A^2(\varphi) = K^2 \cdot \frac{\sin^2 \varphi}{\left[1 + \rho^2 - 2\rho \cos(\varphi - \theta)\right] \cdot \left[1 + \rho^2 - 2\rho \cos(\varphi + \theta)\right]} \quad (6.9)$$

Comme on pouvait le prévoir en vertu de l'interprétation géométrique de la réponse en fréquence, cette fonction atteint un maximum pour  $\varphi = \theta$ . Autour de cette fréquence, l'allure de la réponse en fréquence est surtout influencée par le premier facteur du dénominateur. Si on approxime le second facteur par  $1 + \rho^2 - 2\rho \cos(2\theta)$ , la réponse en fréquence est approximativement donnée par :

$$A^2(\varphi) \approx K^2 \cdot \frac{\sin^2 \varphi}{\left[1 + \rho^2 - 2\rho \cos(\varphi - \theta)\right] \cdot \left[1 + \rho^2 - 2\rho \cos(2\theta)\right]} \quad (\text{pour } \varphi \approx \theta) \quad (6.10)$$

Si de plus on suppose que  $\rho$  est assez voisin de 1 (ce qui correspond au cas d'un filtre très sélectif), on peut même écrire:

$$\begin{aligned} (1 - \rho)^2 &\approx 0 \\ 1 + \rho^2 - 2\rho &\approx 0 \\ 1 + \rho^2 &\approx 2\rho \end{aligned} \quad (6.11)$$

D'où :

$$\begin{aligned} 1 + \rho^2 - 2\rho \cos(2\theta) &= 1 + \rho^2 - 2\rho \cos^2 \theta + 2\rho \sin^2 \theta \\ &\approx 2\rho(1 - \cos^2 \theta) + 2\rho \sin^2 \theta \\ &= 4\rho \sin^2 \theta \end{aligned} \quad (6.12)$$

Ce qui donne finalement :

$$A^2(\varphi) \approx \frac{K^2}{\left[1 + \rho^2 - 2\rho \cos(\varphi - \theta)\right] \cdot 4\rho} \quad (\text{si } \rho \approx 1) \quad (6.13)$$

Le maximum vaut alors :

$$\boxed{A_M = \frac{K}{2(1 - \rho) \cdot \sqrt{\rho}}} \quad (6.14)$$

---

<sup>1</sup> Il n'est pas indispensable d'avoir un zéro pour créer une résonance. On montre par contre que l'utilisation d'un zéro tel que sur la Fig. 6.3 conduit à une filtre dont l'amplitude de la résonance est indépendante de  $\theta$ .

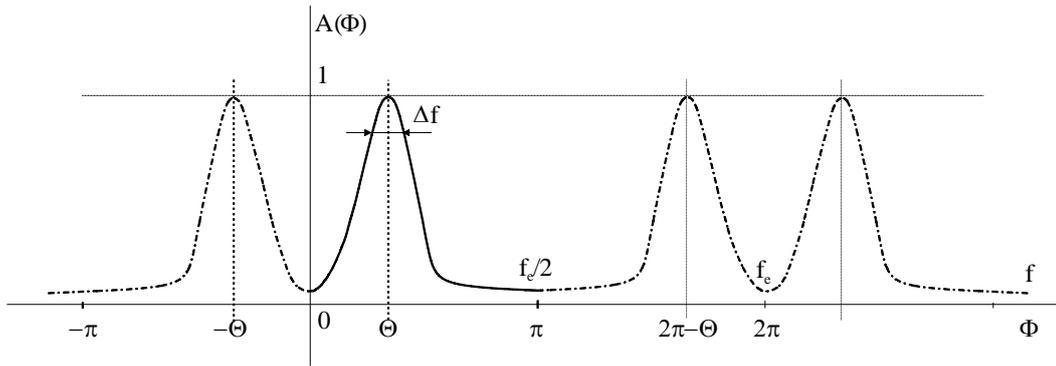


Fig. 6.4 Réponse en fréquence d'un résonateur du second ordre

La loi de l'amplitude en fonction de  $\varphi$  (ou de la fréquence  $f$ ) est représentée à la Fig. 6.4

L'intervalle de fréquences  $\Delta\varphi$  dans lequel  $A$  est supérieur à  $1/\sqrt{2}$  (la bande passante à 3 dB) se calcule comme suit :

$$A(\varphi)^2 = \frac{1}{2} \cdot A_M^2 = \frac{K^2}{8 \cdot \varepsilon^2 \cdot \rho} \quad \text{si } \rho = 1 - \varepsilon \quad (6.15)$$

Comme  $\cos x \approx 1 - x^2/2$ , il vient :

$$1 + \rho^2 - 2\rho \cdot (1 - x^2/2) \cong (1 - \rho)^2 + x^2 = \varepsilon^2 + x^2 \quad (6.16)$$

On en tire donc la relation :

$$(\varepsilon^2 + x^2) \cdot 4\rho = 8\varepsilon^2 \rho \Rightarrow x = \pm \varepsilon \quad (6.17)$$

c'est-à-dire

$\Delta\varphi = 2\varepsilon = 2(1 - \rho) \quad \text{radians}$ $\Delta f = 2(1 - \rho) \cdot F_c / 2\pi \quad \text{Hertz}$	(6.18)
--	--------

### 6.1.3 Le filtre coupe-bande du second ordre

La transmittance possédant deux zéros sur le cercle de rayon unité et deux pôles proches de ces zéros et de mêmes arguments :

$$H(z) = K \cdot \frac{1 - 2\cos\theta \cdot z^{-1} + z^{-2}}{1 - 2\rho \cos\theta \cdot z^{-1} + \rho^2 z^{-2}} \quad \rho = 1 - \varepsilon \quad \varepsilon \ll 1 \quad (6.19)$$

caractérise un filtre destiné à éliminer une composante spectrale de fréquence

$$f_0 = \theta \cdot \frac{F_c}{2\pi} \quad (\text{figure 4.18}).$$

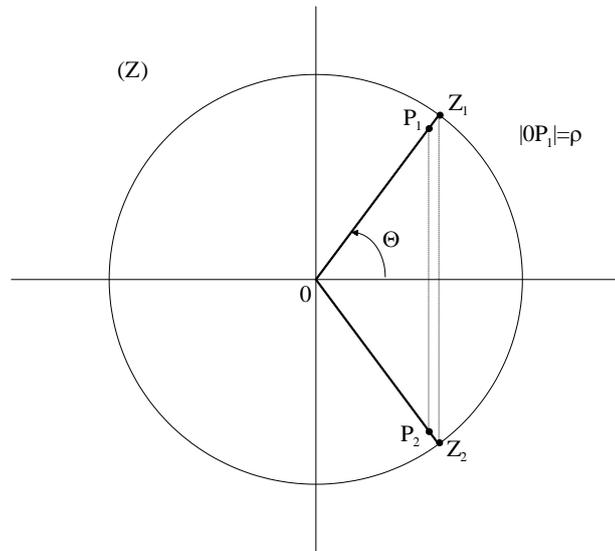


Fig. 6.5 Coupe-bande du second ordre (pôles et zéros)

La loi de l'amplitude est représentée à la Fig. 6.5. On peut montrer, par un raisonnement similaire à ce qui a été fait précédemment, que l'intervalle de fréquences autour de  $f_0$  dans lequel l'amplitude est inférieure à  $1/\sqrt{2}$  (la bande coupée à 3 dB) vaut encore :

$$\begin{aligned} \Delta\varphi &= 2\varepsilon = 2(1-\rho) \text{ radians} \\ \Delta f &= 2(1-\rho).F_c / 2\pi \text{ Hertz} \end{aligned} \tag{6.20}$$

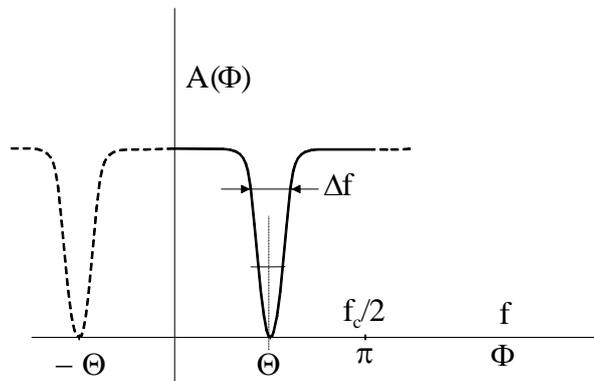
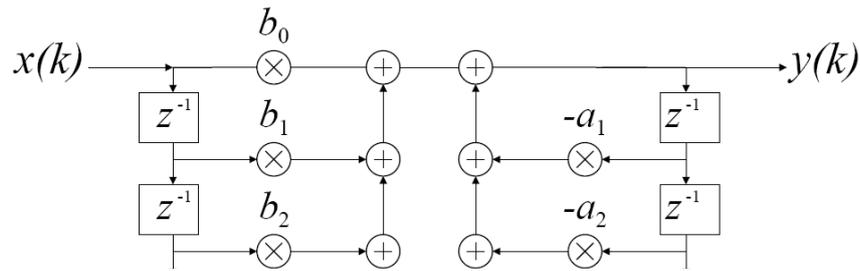


Fig. 6.6 Réponse en fréquence d'un filtre coupe-bande

### 6.1.4 Implémentation d'une cellule du second degré

Nous avons présenté au Chapitre 1 une structure possible pour l'implémentation de la récurrence numérique d'un filtre récursif (le plus général). Cette structure, appelée structure *directe I* (ou *immédiate*), implémente en fait  $B(z)/A(z)$  comme la cascade de  $B(z)$  suivi de  $A(z)$  (Fig. 6.7). Ce n'est pas la seule structure possible, et en pratique ce n'est pas celle qui est utilisée le plus souvent.

Fig. 6.7 Structure directe I (dite *immédiate*)

Il est facile de trouver la structure dite *directe II*, en intervertissant l'ordre de l'implémentation de  $A(z)$  et  $B(z)$  (Fig. 6.8). Il est clair que les valeurs  $w(k)$ ,  $w(k-1)$ , et  $w(k-2)$  apparaissant dans les parties gauche ( $A(z)$ ) et droite ( $B(z)$ ) de cette structure sont identiques dans la partie  $A(z)$  et  $B(z)$ . On peut donc en pratique fusionner les éléments délais, ce qui divise par deux le nombre de délais nécessaires par rapport à la structure directe I. Ce nombre de délai est minimum, d'où le nom de *canonique* pour cette structure.

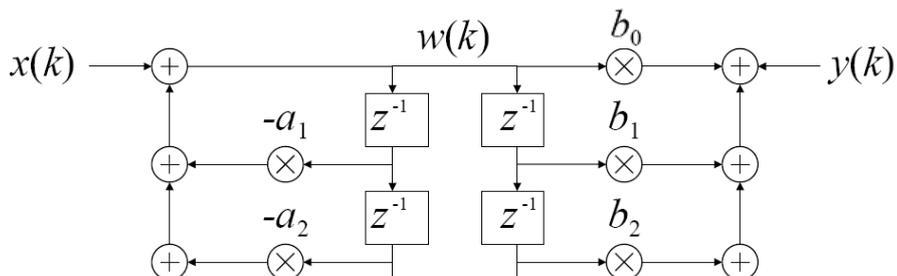
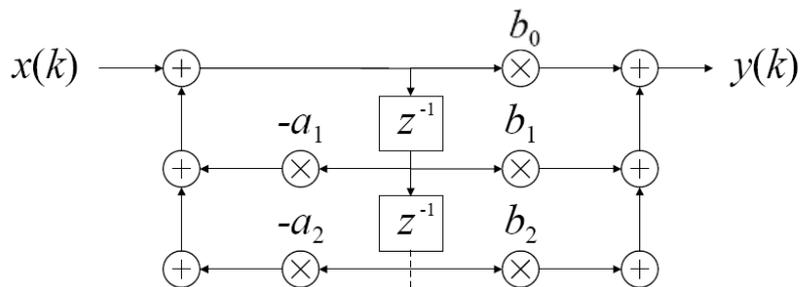
Fig. 6.8 Intervernion de  $B(z)$  et  $A(z)$ 

Fig. 6.9 Structure directe II (version finale)

Il est également possible d'implémenter la même récurrence avec une structure dite *directe II transposée* (Fig. 6.10). Le principe de la *transposition* a été établi par Mason, sur base d'une règle simple, dite *de Mason*, pour le calcul de la fonction de transfert d'un filtre :

$$H(z) = \frac{\sum_i P_i(z)}{1 - \sum_j B_j(z)} \quad (6.21)$$

où  $P_i(z)$  représente la fonction de transfert associé à un parcours dans la structure du filtre, et où  $B_j(z)$  représente celle d'une boucle (les sommes sur  $i$  et  $j$  s'étendent sur tous les parcours et toutes les boucles). On peut vérifier que les structures précédentes vérifient bien cette règle.

Sur base de cette règle, on peut toujours obtenir une structure dite *transposée* à partir d'une structure de base, en réalisant les opérations suivantes :

1. Remplacer les nœuds de sommation par des nœuds de dispersion, et inversement
2. Inverser tous les sens de parcours

Appliquée à la structure directe II, cette opération conduit à la structure directe II transposée. On notera que cette structure n'est autre que la structure directe I dont on a déplacé (et fusionné) les éléments délais des extrémités vers le centre de la structure.

MATLAB implémente par défaut la structure directe II transposée. D'autres structures sont disponibles via la fonction `dfilt`

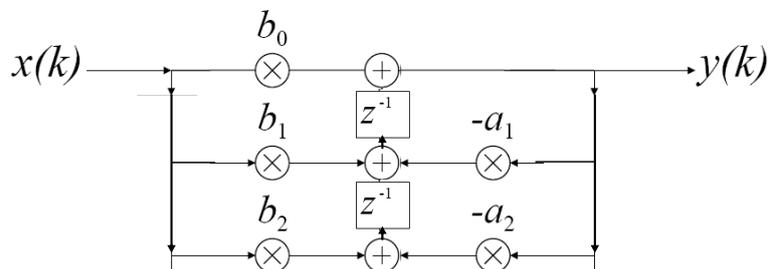


Fig. 6.10 Structure directe II transposée

Le choix d'une structure ou d'une autre est pratique dicté par un critère de bruit de calcul (voir 6.2.4).

## 6.2 Approximation et synthèse des filtres IIR

Le problème de l'approximation ayant largement été étudié dans le cadre des filtres analogiques (cf. cours de Théorie des Circuits : Butterworth, Chebyshev, Cauer, etc.), une des méthodes de conception consiste à trouver des transformations qui permettent de passer du domaine analogique (plan  $p$ ) vers le domaine numérique (plan  $z$ ).

On suppose que la fonction de transfert d'un filtre analogique :

$$H_a(p) = \frac{N(p)}{D(p)} \quad (6.22)$$

a été déterminée par l'une des méthodes classiques d'approximation des filtres analogiques. On doit alors trouver une fonction de transfert numérique :

$$H(z) = \frac{B(z)}{A(z)} \quad (6.23)$$

qui ait la même réponse du filtre analogique sans en augmenter le degré, et en assurant le maintien de la stabilité: les pôles de  $H(z)$  doivent donc être situés à l'intérieur du cercle unité.

Parmi les transformations existantes, on ne considèrera que les deux plus importantes : la transformation qui préserve la réponse impulsionnelle et la transformation bilinéaire.

### 6.2.1 La transformation qui préserve la réponse impulsionnelle

Si on veut un filtre numérique dont la réponse impulsionnelle  $h(n)$  soit une version échantillonnée de celle du filtre analogique  $h_a(t)$  :

$$h(n) = h_a(nT_e) \quad (6.24)$$

on peut évidemment imposer:

$$H(z) = \sum_{i=0}^{\infty} h(i)z^{-i} = \sum_{i=0}^{\infty} h_a(iT_e)z^{-i} \quad (6.25)$$

ce qui conduit tout naturellement à un filtre FIR, dont la réponse en fréquence est la TFTD de  $h(n)$ . Il faut donc, pour éviter un recouvrement des images des réponses en fréquences du filtre analogique, échantillonner  $h_a(t)$  avec une période d'échantillonnage respectant le théorème de Shannon.

L'implémentation d'un tel FIR pose cependant le problème de la longueur de la suite  $h(n)$  : même si on ne considère la réponse  $h_a(t)$  que pour  $t=0\dots t_{max}$  (où  $t_{max}$  est tel que  $h_a(t > t_{max}) < 0.001 * \max(h_a(t))$ ), il reste que la condition de non recouvrement conduit le plus souvent à une suite  $h(n)$  très longue, et la charge de calcul est alors importante.

On peut heureusement respecter (6.25) en passant par un filtre IIR, nettement moins gourmand en calcul. Considérons par exemple un filtre analogique d'ordre un :

$$H_a(p) = \frac{A}{p-a} \quad (6.26)$$

Sa réponse impulsionnelle est donnée par :

$$h_a(t) = Ae^{at} \quad (6.27)$$

On peut facilement obtenir  $h(n)$  par le filtre numérique suivant :

$$H(z) = \frac{A}{1 - e^{aT_e} z^{-1}} \quad (6.28)$$

Ce principe peut être généralisé en passant par une décomposition en fractions simples de (6.22) :

$$H_a(p) = K \sum_{i=1}^N \frac{A_i}{p - p_i} \quad (6.29)$$

ce qui donne:

$$H(z) = K \sum_{i=1}^N \frac{A_i}{1 - e^{p_i T_e} z^{-1}} \quad (6.30)$$

On obtient donc un filtre numérique de même degré que le filtre analogique de départ, et dont la stabilité est bien entendu préservée par la transformation.

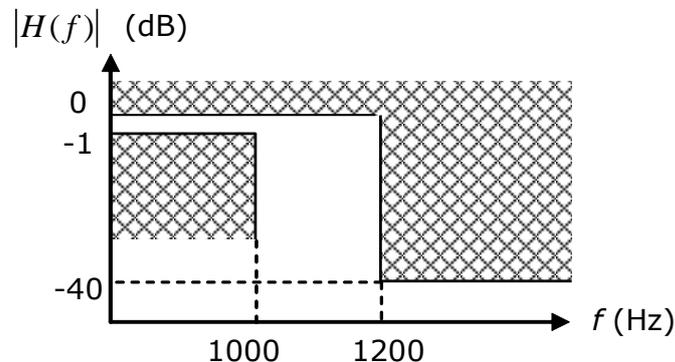
En pratique, on multiplie également  $H(z)$  par  $T_e$  pour compenser le fait que la TFTD de  $h(n)$  ne correspond à  $H_a(f)$  qu'à un facteur  $1/T_e$  près :

$$H(z) = K T_e \sum_{i=1}^N \frac{A_i}{1 - e^{p_i T_e} z^{-1}} \quad (6.31)$$

L'inconvénient de cette approche est que, si la réponse en fréquence  $H_a(f)$  n'est pas négligeable pour  $f > F_e/2$ , elle fait apparaître un recouvrement spectral qui peut diminuer les performances du filtre.

### Exemple 6.1

On cherche à créer un filtre passe-bas numérique ( $F_e=3000$ ) par approximation de Chebyshev de type I, qui respecte les spécifications suivantes :



On réalise l'approximation du filtre analogique et utilise la fonction `impinvar(N,D,Fe)` :

```
Fe=3000;
[n,wn]=cheblord(2*pi*1000,2*pi*1200,1,40,'s');
[N,D]=cheby1(n,1,wn,'s');
[B,A]=impinvar(N,D,Fe); % on multiplie aussi par Te

% Calcul de la réponse impulsionnelle analogique
[r,p,k]=residue(N,D);
t=0:1/(5*Fe):.02;
ha=real([exp(t.*(p.))*r]);

% Calcul de la réponse impulsionnelle numérique
h=filter(B,A,[1;zeros(49,1)]);

% Comparaison des réponses impulsionnelles
plot(t,ha);
hold on;
```

```

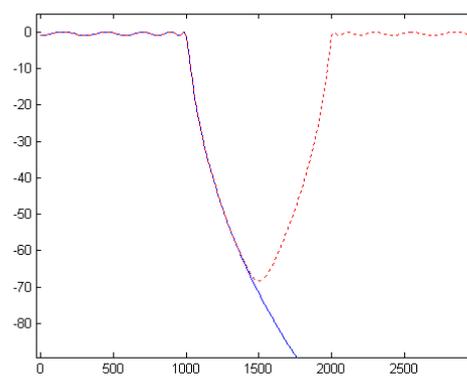
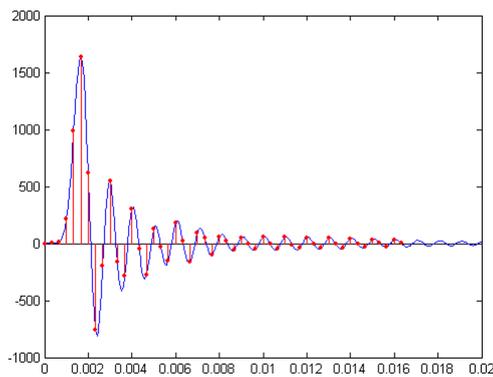
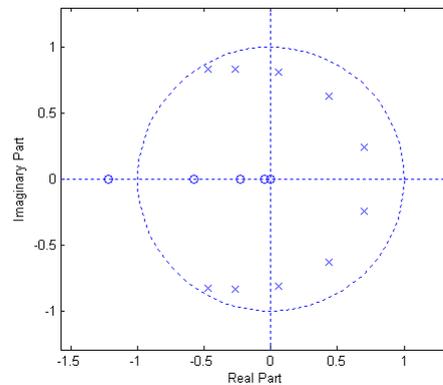
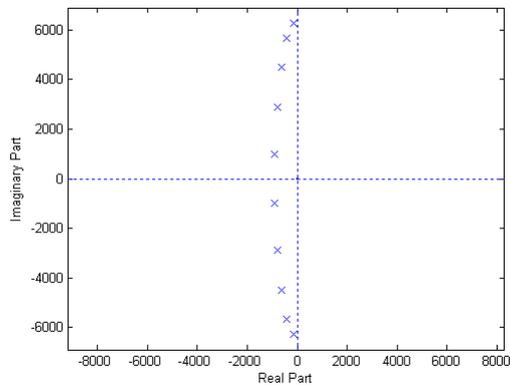
stem(0:1/Fe:49/Fe, Fe*h,'r.')

% Comparaison des pôles et zéros
figure;
zplane(N,D)
figure
zplane(B,A)

% Calcul des réponses en fréquence
[Ha,Wa]=freqs(N,D,2*pi*(1:10:Fe));
[H,W]=freqz(B,A,1:10:Fe,Fe);

% Comparaison des réponses en fréquence
figure;
plot(Wa/(2*pi),20*log10(abs(Ha)));
hold on;
plot(W,20*log10(abs(H)),'r:')

```



On constate effectivement que le degré du filtre est conservé, ainsi que la réponse impulsionnelle, et que la réponse en fréquence est rendue périodique par l'échantillonnage de cette réponse, ce qui rend le filtre légèrement moins efficace autour de  $F_e/2$

## 6.2.2 La transformation bilinéaire

Une transformation simple qui élimine cet inconvénient est la *transformation bilinéaire* (ou homographique) définie par:

$$p = \frac{1 - z^{-1}}{1 + z^{-1}} \quad z = \frac{1 + p}{1 - p} \quad (6.32)$$

qui établit une correspondance entre l'axe des imaginaires du plan  $p$  et le cercle unité du plan  $z$  (Fig. 6.11).

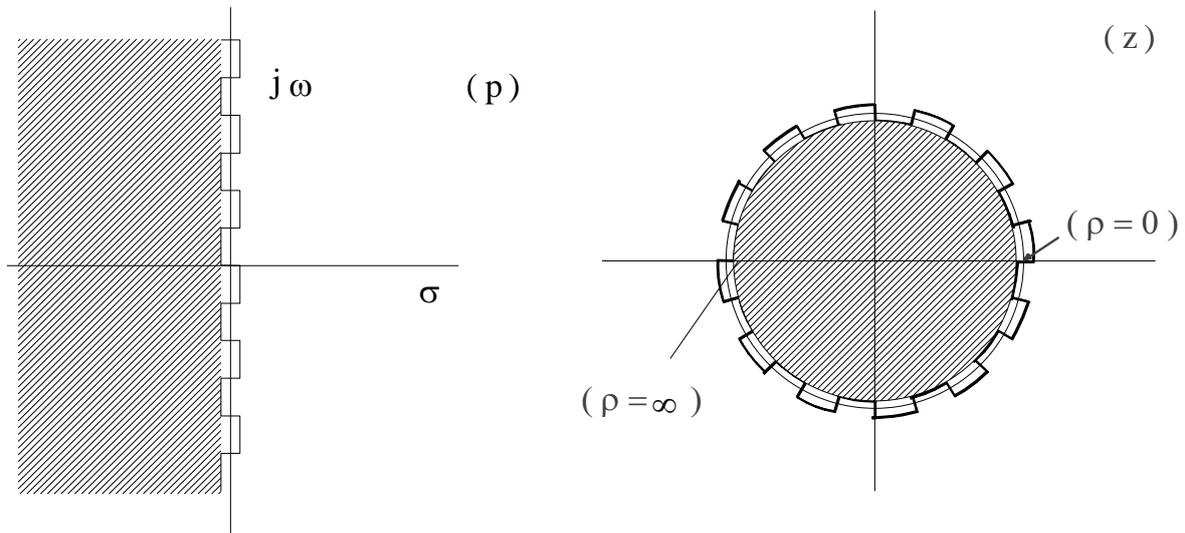


Fig. 6.11 Correspondance  $p - z$  établie par la transformation bilinéaire

Par conséquent, au demi-plan de gauche du plan  $p$  correspond l'intérieur du cercle unité et au demi-plan de droite l'extérieur du même cercle, l'axe imaginaire étant transformé en le cercle lui-même (figure 4.21) : la stabilité est préservée.

En particulier, on peut en effet vérifier que :

$$p = j\omega \Rightarrow z = e^{j\varphi} \quad (6.33)$$

On peut ainsi parler d'une correspondance entre la pulsation  $\omega_a$  du système à temps continu et la pulsation réduite  $\varphi$  du système à temps discret: en effet, pour  $p$  imaginaire pur, l'équation (6.32) devient :

$$j\omega_a = \frac{1 - e^{-j\varphi}}{1 + e^{-j\varphi}} \frac{1 + e^{j\varphi}}{1 + e^{j\varphi}} = \frac{2j \sin \varphi}{2 + 2 \cos \varphi} = \frac{j2 \sin \frac{\varphi}{2} \cos \frac{\varphi}{2}}{\cos^2 \frac{\varphi}{2}} \quad (6.34)$$

$$= j \tan\left(\frac{\varphi}{2}\right)$$

En conséquence, si on calcule la fonction de transfert d'un filtre numérique  $H(z)$  en partant d'une fonction de transfert analogique  $H_a(p)$  en remplaçant  $p$  par sa valeur dans (6.32) :

$$H(z) = H_a(p) \Big|_{p=(1-z^{-1})/(1+z^{-1})} \quad (6.35)$$

on aura :

$$H(\varphi) = H_a(\omega_a) \text{ avec } \omega_a = \tan\left(\frac{\varphi}{2}\right), \text{ ou } \varphi = 2 \arctan(\omega_a) \quad (6.36)$$

La relation entre la pulsation analogique  $\omega$  et la pulsation numérique  $\varphi$  est représentée à la Fig. 6.12. La réponse en fréquence du filtre analogique peut donc s'étendre jusqu'à l'infini: il n'y a plus aucun risque de recouvrement.

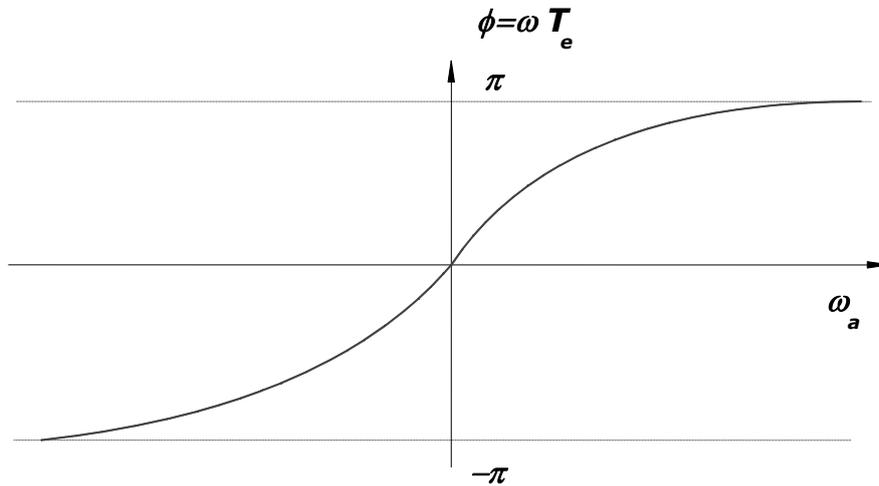


Fig. 6.12 Correspondance  $\omega_a - \varphi$  établie par la transformation bilinéaire

La transformation bilinéaire occasionne cependant une distorsion de l'axe des fréquences comme le montre la figure 4.22; elle ne pourra donc être utilisée que si la distorsion peut être tolérée ou bien compensée par une prédistorsion adéquate. Dans la plupart des cas rencontrés en pratique, les spécifications sont constantes dans des domaines de fréquences donnés, comme l'illustre par exemple la Fig. 6.13. Les spécifications du filtre analogique destiné à être transformé sont alors obtenues simplement : on calcule les fréquences critiques analogiques par prédistorsion de celles qui sont imposées pour le filtre numérique, et on ne modifie pas les atténuations.

Une fois les spécifications du filtre analogique obtenues, on peut en réaliser l'approximation(analogique) de Butterworth, de Chebyshev ou de Cauer, et repasser ensuite au domaine numérique par (6.35)

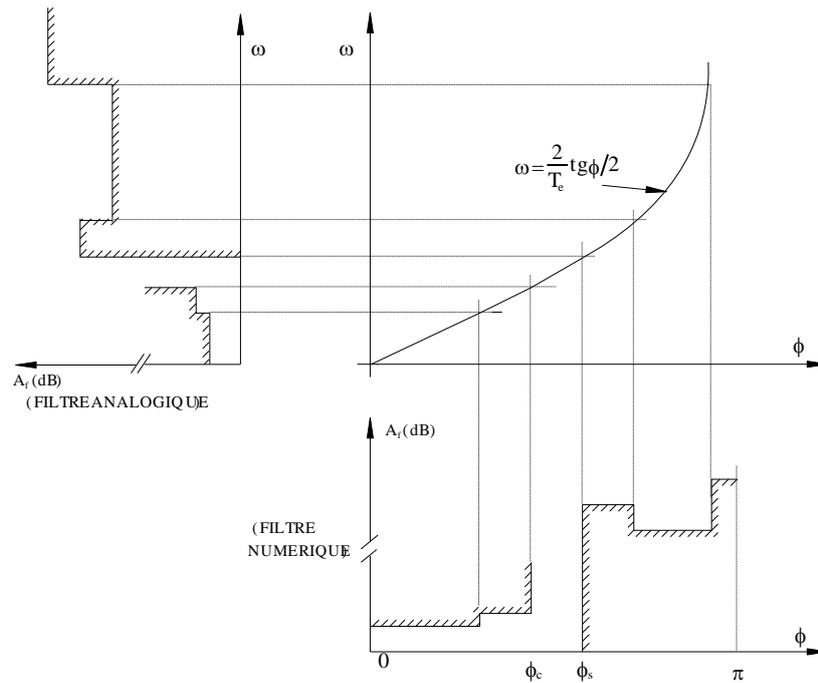


Fig. 6.13 Technique de compensation de la distorsion des fréquences due à la transformation bilinéaire (exemple d'un filtre passe-bas).

### Exemple 6.2

Reprenons l'exemple précédent, cette fois par transformation bilinéaire. Le calcul sous MATLAB est immédiat : Si l'on ne spécifie pas 's' dans l'appel des fonctions (ce qui veut dire que l'on demande, non pas une approximation analogique, mais bien une approximation numérique), MATLAB intègre par défaut la prédistorsion, l'approximation analogique, et le passage au numérique par la transformation bilinéaire. Ainsi par exemple, pour un filtre de Chebyshev :

```

Fe=3000;
[n,wn]=cheblord(1000/(Fe/2),1200/(Fe/2),1,40);
[B,A]=cheby1(n,1,wn);

% Calcul du filtre analogique, pour comparaison
[n,wn]=cheblord(2*pi*1000,2*pi*1200,1,40,'s');
[N,D]=cheby1(n,1,wn,'s');

% Comparaison des pôles et zéros
zplane(N,D)
figure
zplane(B,A)

% Calcul de la réponse impulsionnelle analogique
[r,p,k]=residue(N,D);
t=0:1/(5*Fe):.02;
ha=real([exp(t*(p.'))*r]);

% Calcul de la réponse impulsionnelle numérique
h=filter(B,A,[1;zeros(49,1)]);

% Comparaison des réponses impulsionnelles
figure;
plot(t,ha);

```

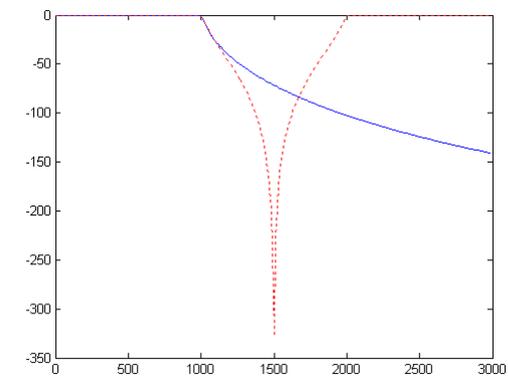
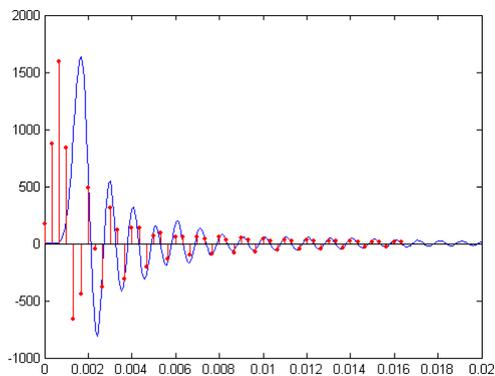
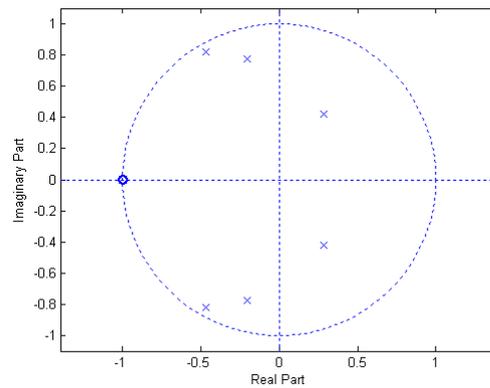
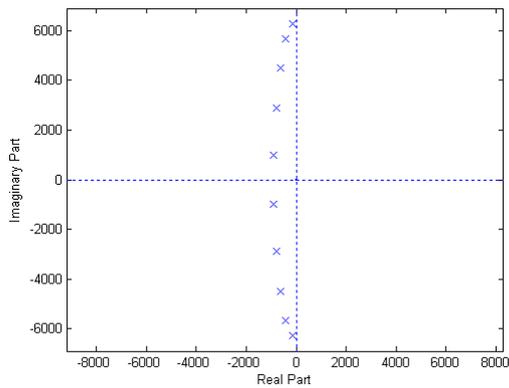
```

hold on;
stem(0:1/Fe:49/Fe, Fe*h, 'r. ')

% Calcul des réponses en fréquence
[Ha, Wa]=freqs(N,D,2*pi*(1:10:Fe));
[H,W]=freqz(B,A, 1:10:Fe, Fe);

% Comparaison des réponses en fréquence
figure;
plot(Wa/(2*pi), 20*log10(abs(Ha)));
hold on;
plot(W, 20*log10(abs(H)), 'r: ')

```



On constate que la réponse impulsionnelle et la réponse en fréquence du filtre numérique sont différentes de celle du filtre analogique, mais que la différence est favorable au filtre numérique, de par la présence de zéros en  $z=-1$ .

### 6.2.3 Le problème de la quantification des coefficients

Lorsqu'un filtre numérique est implémenté sur un processeur en virgule fixe (ou lorsqu'il est directement implémenté en hardware dans un ASIC en virgule fixe), les coefficients calculés par approximation doivent être quantifiés sur un nombre fini de bits, ce qui en modifie la valeur. Malheureusement, un polynôme étant très sensible à une modification de ses coefficients, la simple quantification des coefficients  $b_i$  ou  $a_i$  peut conduire à une modification majeure de la réponse en fréquence du filtre.

#### Exemple 6.3

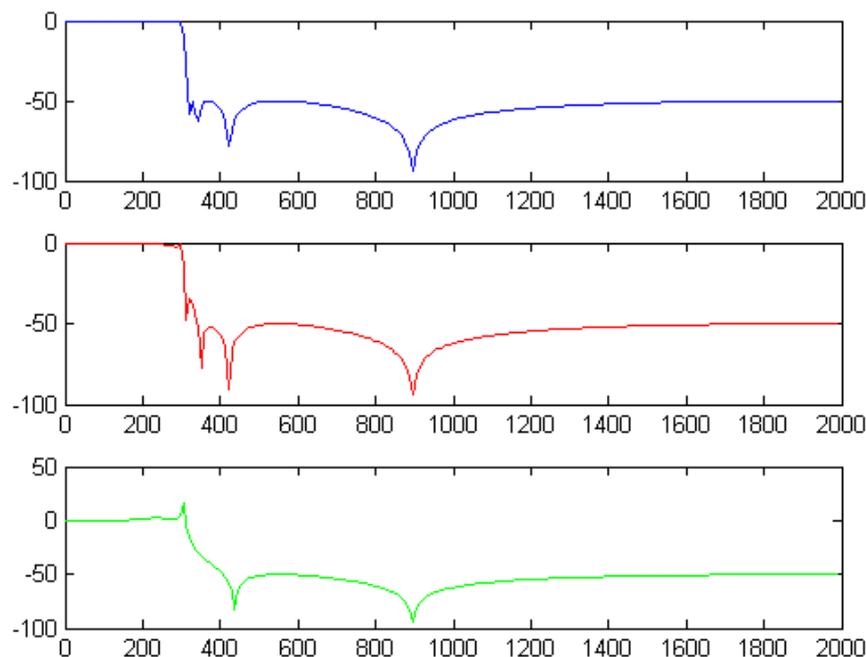
Nous allons faire l'approximation d'un filtre passe-bas IIR de Cauer, de degré 8, avec une fréquence de coupure de 300 Hz, un ripple en bande passante de 0.5 dB et une atténuation minimale en bande atténuée de 50 dB (la fréquence d'échantillonnage est de 4 kHz). Nous comparerons ensuite la réponse en fréquence du filtre à celle des filtres obtenus en diminuant la précision des coefficients à 6, puis 5 chiffres après la virgule (le calcul initial de MATLAB étant affectué avec 16 chiffres après la virgule).

```
% Approximation de départ
[b,a]=ellip(8, .5,50,300/2000);
[h,f]=freqz(b,a,256,4000);

% 6 chiffres décimaux
b6=round(b*1^6)/1^6;
a6=round(a*1^6)/1^6;
[h6,f]=freqz(b6,a6,256,4000);

% 5 chiffres décimaux
b5=round(b*1^5)/1^5;
a5=round(a*1^5)/1^5;
[h5,f]=freqz(b5,a5,256,4000);

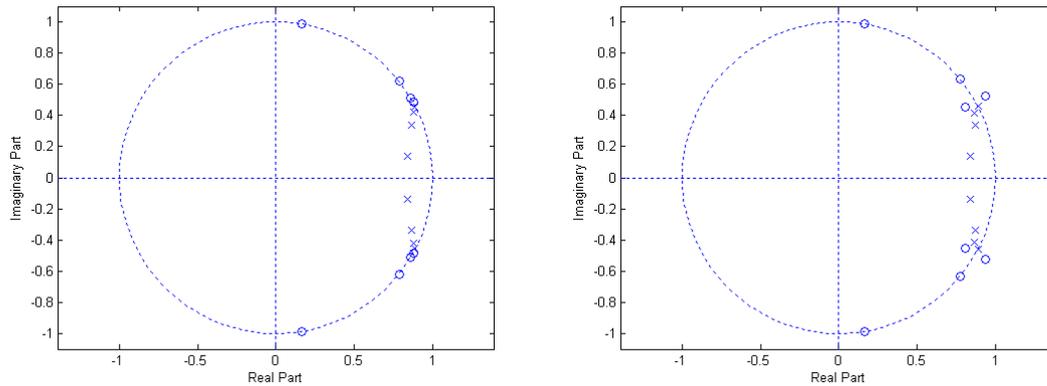
% Comparaison
subplot(3,1,1);
plot(f,20*log10(abs(h)));
subplot(3,1,2);
plot(f,20*log10(abs(h6)), 'r');
subplot(3,1,3);
plot(f,20*log10(abs(h5)), 'g');
```



On constate que la réponse en fréquence est fort modifiée lors du passage à 5 chiffres après la virgule. Pire : l'examen des positions des pôles et zéros révèle un problème : le passage à 5 chiffres après la virgule a poussé certaines pôles en dehors du cercle de rayon unité.

```
% Comparaison
zplane(b,a);
figure;
```

zplane(b5,a5);



La réponse en fréquence donnée dans la fenêtre inférieure du graphique ci-dessus n'a donc aucune signification physique : le filtre est devenu instable!

### 6.2.4 Le problème du bruit de calcul

Il ne faut pas oublier que dans un filtre en virgule fixe, les échantillons sont eux aussi quantifiés sur un nombre de bits fini. Ceci a deux conséquences importantes :

1. En sortie de chaque multiplicateur, les valeurs numériques sont connues sur  $2N$  bits (où  $N$  est le nombre de bits des valeurs d'entrée : la multiplication de deux nombres sur  $N$  bits produit un nombre sur  $2N$  bits). Si on ne conserve pas ces  $2N$  bits mais qu'on requantifie plutôt les valeurs, la requantification des résultats crée au droit de chaque multiplieur un bruit que l'on appelle *bruit de calcul*. Il est clair alors que l'ordre des cellules du second degré influence sur la transmission de ce bruit de calcul vers la sortie du filtre.

Pour atténuer cet effet, on réalise en général les calculs internes sur  $N_c$  bits (où  $N_c$  est le nombre de bits *interne*, qui peut être différent du nombre de bits sur lequel les échantillons d'entrée et de sortie sont quantifiés). La valeur de  $N_c$  est alors choisie telle qu'il en résulte un bruit de calcul en sortie du filtre dont la puissance  $\sigma_c^2$  soit du même ordre de grandeur que la puissance  $\sigma_s^2$  du bruit en sortie du filtre, dû à la quantification des échantillons d'entrée. Il en résulte une valeur de  $N_c$  qui est comprise entre  $N$  et  $2N$ .

2. Les additionneurs posent un problème différent : ils n'opèrent pas de requantification, mais peuvent être soumis au *dépassement*, si le résultat de la somme dépasse la valeur maximale admissible en sortie. Le dépassement est le plus souvent interprété par le calculateur comme un changement de signe, ce qui produit des erreurs importantes (assimilables à des bruits) et des comportements transitoires qui peuvent perturber le bon fonctionnement du filtre, jusqu'à même devenir des bruits de régime (on parle de *cycles limites* lorsque les coefficients du filtre conduisent à ce que ces transitoires de dépassement s'entretiennent d'eux-mêmes).

L'étude complète de ces effets sort du cadre de ce cours. On montre cependant que les structures des cellules du second degré (Directe I, II, transposée) possèdent des caractéristiques de bruit de calcul fort différentes

l'une de l'autre. Il est donc intéressant, avant de choisir une implémentation, d'en tester les performances de bruit

### 6.2.5 Synthèse des filtres IIR en virgule fixe, par cascade de cellules du second degré

Les problèmes mentionnés à la Section précédente sont d'autant plus importants que le degré du filtre est élevé. En effet, la quantification de chacun des coefficients du dénominateur modifie la position de *tous* les pôles. Tout comme en synthèse des filtres analogiques, une solution pour minimiser le problème est de réaliser le filtre :

$$H(z) = \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} = K \frac{\prod_{i=1}^M (1 - z_i z^{-i})}{\prod_{i=1}^N (1 - p_i z^{-i})} \quad (6.37)$$

(où  $z_i$  et  $p_i$  sont les zéros et les pôles du filtre) sous la forme d'une cascade de cellules du second degré (avec si nécessaire une section du premier degré)<sup>2</sup>:

$$H(z) = K_0 \frac{1 - z_0 z^{-1}}{1 - p_0 z^{-1}} \prod_{i=1}^{M/2} \left( K_i \frac{(1 + b_{i1} z^{-1} + b_{i2} z^{-2})}{(1 + a_{i1} z^{-1} + a_{i2} z^{-2})} \right) \quad (6.38)$$

Il reste alors à décider quelles paires de zéros on associe à quelles paires de pôles, et dans quel ordre placer les cellules, et comment répartir la constante  $K$ .

Le problème du bruit de calcul, mentionné à la Section précédente, intervient ici directement.

On retiendra simplement que (Fig. 6.14) :

1. Pour le choix de la répartition de la constante  $K$ , on cherchera à minimiser les risques de dépassement, en répartissant  $K$  de façon à ce que la dynamique des sorties des cellules soient toujours adaptées à celle du quantificateur. Ainsi, si une section du second degré présente une amplification importante à sa fréquence de résonance, on utilisera pour cette section un facteur  $K_i$  égal à l'inverse de ce facteur d'amplification (de manière à éliminer l'amplification).
2. Pour le choix de l'association pôles-zéros, on cherchera toujours à associer aux pôles les plus résonants les zéros les plus proches, de manière à en compenser au maximum la résonance.
3. Pour le choix de l'ordre des cellules, on placera toujours en premier les cellules les moins résonantes, pour terminer par les cellules les plus résonantes.

<sup>2</sup> On peut en vérifier le principe sous MATLAB grâce à la fonction `sosdemo`.

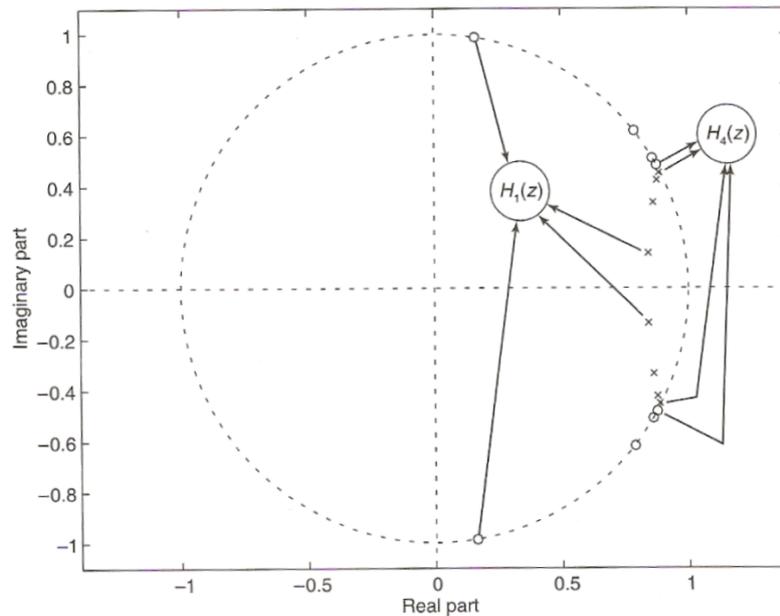


Fig. 6.14 Exemple d'association pôles-zéros pour un filtre elliptique

### Exemple 6.4

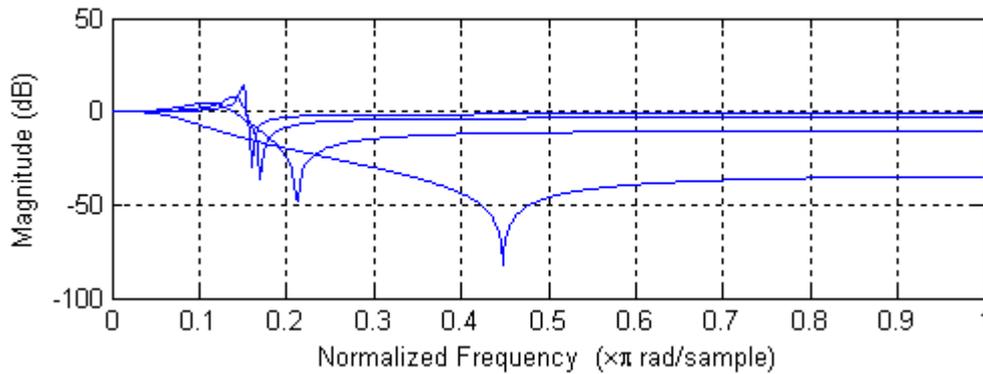
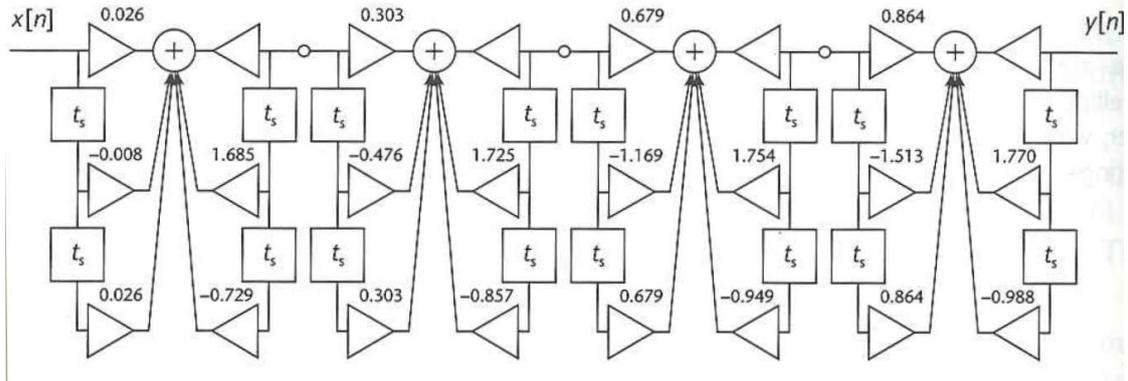
Sous MATLAB le calcul des sections du second degré nécessaires à l'implémentation cascade d'un filtre numérique est directement disponible dans la fonction `tf2sos(b,a,ordre,scale)` (où 'tf' signifie "transfer function" et "sos" signifie "second order section", où `ordre` est mis à "up" par défaut et spécifie l'ordre des cellules en terme de facteur de qualité, et où `scale` spécifie comme on répartie la constante  $K$  : "Inf" correspond à la procédure proposé ci-dessus, qui normalise le maximum de la réponse en fréquence de chaque cellule à 1). Reprenons l'approximation de l'exemple précédent :

```
[b,a]=ellip(8, .5, 50, 300/2000);
sos=tf2sos(b,a, 'up', Inf)
```

```
>sos=
    0.0261   -0.0085    0.0261    1.0000   -1.6853    0.7290
    0.3027   -0.4761    0.3027    1.0000   -1.7250    0.8573
    0.6795   -1.1687    0.6795    1.0000   -1.7542    0.9488
    0.8641   -1.5131    0.8641    1.0000   -1.7705    0.9882
```

```
sos1n=sos(1,1:3); sos1d=sos(1,4:6);
sos2n=sos(2,1:3); sos2d=sos(2,4:6);
sos3n=sos(3,1:3); sos3d=sos(3,4:6);
sos4n=sos(4,1:3); sos4d=sos(4,4:6);
freqz(sos1n,sos1d); hold on;
freqz(sos2n,sos2d);
freqz(sos3n,sos3d);
freqz(sos4n,sos4d);
```

La structure du filtre correspondant est donnée ci-dessous (où  $t_s$  symbolise un élément délai), ainsi que la réponse en fréquence de ses cellules du second degré.



Il est noter que MATLAB fournit deux toolboxes spéciales pour l'étude des filtres en virgule fixe : la Filter Design Toolbox, couplée à la Fixed-Point Toolbox.

Notons pour terminer que ces problèmes ne concernent pas les filtres en virgule flottante.

### 6.3 Approximation et synthèse des filtres FIR

#### 6.3.1 Filtres à phase linéaire

Les filtre FIR ont une fonction de transfert de la forme :

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \tag{6.39}$$

Ils sont toujours stables et leur principal intérêt est qu'ils peuvent facilement implémenter des filtres à phase linéaire. Par contre, pour une même sélectivité, leur degré est souvent fort supérieur à celui d'un filtre IIR. Les filtres FIR et IIR ne sont donc pas utilisés dans les mêmes applications. Une utilisation typique des filtres FIR est par exemple celle des filtres d'interpolation ou de décimation, où la phase linéaire de FIR permet de conserver la forme du signal.

La propriété de linéarité de la phase est obtenue en imposant que la réponse impulsionnelle  $h(n)$  (qui est aussi la suite des coefficients du filtre) soit symétrique (Fig. 6.15) <sup>3</sup>:

$$h(n) = h(-n) \quad (6.40)$$

En effet, considérons la réponse impulsionnelle  $h(n)$  comme la somme de deux réponses distinctes :

$$h(n) = h_1(n) + h_2(n)$$

$$\text{avec } h_1(n) = \begin{cases} h(n) & \text{pour } n < 0 \\ h(n)/2 & \text{pour } n = 0 \\ 0 & \text{pour } n > 0 \end{cases} \quad (6.41)$$

$$\text{et } h_2(n) = \begin{cases} 0 & \text{pour } n < 0 \\ h(n)/2 & \text{pour } n = 0 \\ h(n) & \text{pour } n > 0 \end{cases}$$

Clairement, on a :

$$H(\varphi) = H_1(\varphi) + H_2(\varphi) \quad (6.42)$$

Or, si  $h(n)$  est symétrique, on a :

$$h_1(n) = h_2(-n) \quad (6.43)$$

de sorte que:

$$H_1(\varphi) = H_2^*(\varphi) \quad (6.44)$$

ce qui donne bien une phase rigoureusement nulle pour  $H(\varphi)$ . <sup>4</sup>

En pratique, la phase d'un FIR n'est pas nulle mais linéaire car pour rendre le filtre causal on décale sa réponse vers la droite (Fig. 6.15). Le délai  $n_0$  (en nombre d'échantillons) du filtre est alors égal à la moitié de sa longueur, et la phase linéaire vaut  $-\varphi n_0$ .

---

<sup>3</sup> On peut montrer qu'elle est également obtenue lorsque la réponse est antisymétrique :  $h(n) = -h(-n)$ .

<sup>4</sup> C'est d'ailleurs cette même propriété qui explique que la densité spectrale de puissance des signaux aléatoires réels soit purement réelle (leur fonction d'autocorrélation est en effet symétrique).

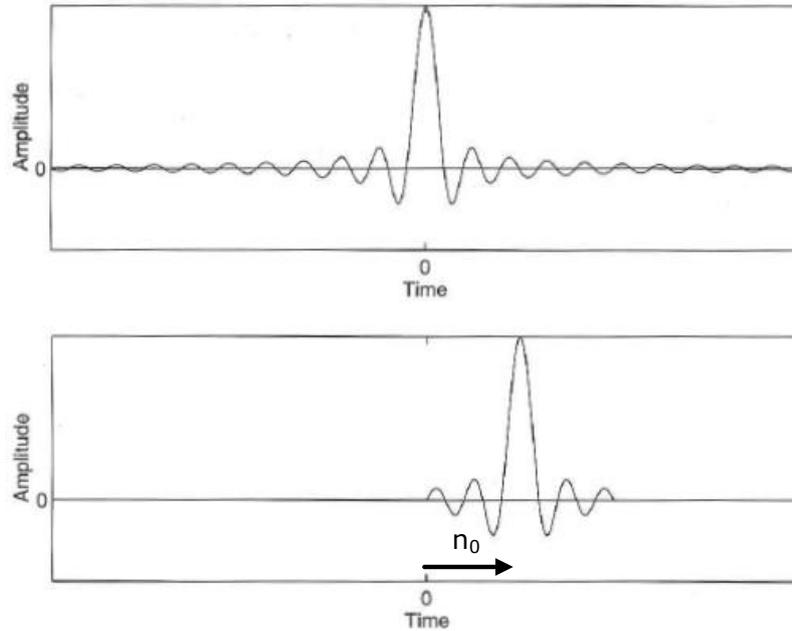


Fig. 6.15 Réponse impulsionnelle d'un filtre à phase nulle (haut) et à phase linéaire (bas)

### 6.3.2 Approximation optimale des filtres FIR à phase linéaire

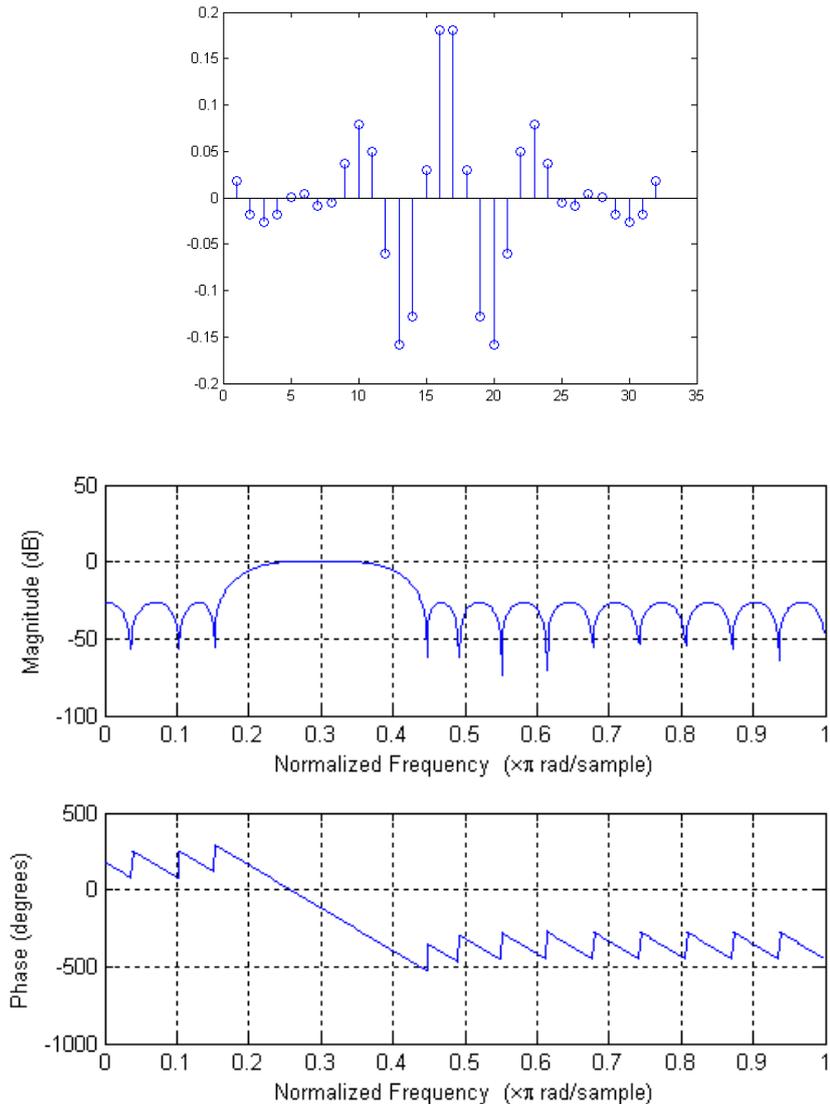
L'approximation numérique due à Parks et McClellan fournit un degré minimal de la transmittance obtenue pour des spécifications données. Le critère utilisé consiste à minimiser le maximum de  $|H(\varphi) - H_d(\varphi)|$  où  $H_d$  est la transmittance idéale à réaliser. L'explication cet algorithme est complexe et sort du cadre de ce chapitre. Fort heureusement, MATLAB en fournit une implémentation simple à utiliser.

Cette technique d'approximation est très utilisée.

#### Exemple 6.5

Pour une fréquence d'échantillonnage de 5 kHz, on cherche à approximer un filtre FIR à phase linéaire, à 32 coefficients, dont la bande passante se situe entre 600 et 900 Hz, et qui impose une forte atténuation avant 400 Hz et au-delà de 1100 Hz.

```
f=[0 400/2500 600/2500 900/2500 1100/2500 1];
m=[0 0 1 1 0 0];
b=remez(31,f,m);
stem(b)
freqz(b,1)
```



On constate que le filtre est bien symétrique, et que sa phase est donc bien linéaire (les accidents de la phase sont des imprécisions de calcul).

### 6.3.3 Synthèse des filtres FIR à phase linéaire

Les filtres à phase linéaire sont beaucoup moins sensibles que les filtres IIR à une quantification de leurs coefficients ou au bruit de calcul. Leur implémentation ne pose pas de problème particulier : la structure directe (il n'y a plus de distinction entre directe I et II) ou la structure transposée (qui ne diffère de la structure directe que par la position des ses éléments délais) conviennent très bien. Il n'est pas nécessaire de réaliser la filtre sous la forme d'une cascade de cellules du second degré.