

Transformée de Fourier

4.1 Objectif du TP

Dans le TP précédent, nous avons étudié les signaux périodiques et leur développement en séries de sinus et de cosinus, ou d'exponentielles complexes, appelées *séries de Fourier*. Un signal périodique de période $T = \frac{2\pi}{\omega}$ apparaissait donc comme une somme infinie d'harmoniques de fréquence $n\omega, n = 1, 2, \dots$. De même, il est possible de représenter les signaux non-périodiques par un outil analogue à une série de Fourier, **la transformée de Fourier**. Dans ce cas, la série de Fourier est remplacée par une intégrale de Fourier, par exemple un son qui n'est pas répété ou une impulsion unique de tension. L'intégrale de Fourier fait intervenir un spectre continu de fréquences.

L'objectif de ce TP est d'aborder **les notions de base de la transformée de Fourier continue et discrète**. Puis, effectuer *l'analyse spectrale* des signaux non périodiques à l'aide des outils disponibles sous MatLab.

4.2 Rappels

4.2.1 La transformée de Fourier continue

Définition

Soit $x(t)$ un signal non périodique. On appelle transformée de Fourier du signal $x(t)$, le signal complexe $X(f)$ de la variable réelle f défini par l'intégrale :

$$X(f) = F[x(t)] = \int_{-\infty}^{+\infty} x(t) \exp(-i2\pi ft) dt$$

- La transformée de Fourier d'un signal $x(t)$ est une intégrale dépendant du paramètre ω .
- $X(f)$ indique la *quantité de fréquence* f présente dans le signal $x(t)$ sur l'intervalle $] -\infty, +\infty[$.
- Le signal $X(f)$ définie pour tout $\omega \in \mathbb{R}$ est une fonction complexe qui admet un spectre d'amplitude $A_f = |X(f)|$ et un spectre de phase $\Phi(f) = \arg|X(f)|$.

Conditions d'existence de la TF

La transformée de Fourier de $x(t)$ est notamment définie dans chacun des cas suivants :

1. Le signal $x(t)$ est nul pour $|t|$ assez grand.
2. L'intégrale $\int_{-\infty}^{+\infty} |x(t)| dt$ est convergente.
3. Le signal $x(t)$ à énergie totale finie.

Transformée de Fourier inverse

La transformée de Fourier inverse du signal $x(t)$ est défini par l'intégrale :

$$x(t) = F^{-1}[X(f)] = \int_{-\infty}^{+\infty} X(f) \exp(i2\pi ft) dt$$

On note que $X(f)$ et $x(t)$ sont deux descriptions équivalentes (temporelle ou fréquentielle) du même signal $x(t) \iff X(f)$.

4.2.2 Propriétés de la transformée de Fourier continue

Continuité :

Lorsque l'intégrale $\int_{-\infty}^{+\infty} |x(t)| dt$ existe, la transformée $X(f)$ est une fonction continue sur $\mathbb{R}$ et de plus $X(f)$ tend vers zéro pour $|\omega|$ tendant vers ∞ .

Linéarité :

La transformée de Fourier est linéaire, autrement dit :

$$a_1 x_1(t) + a_2 x_2(t) \iff a_1 X_1(f) + a_2 X_2(f)$$

Parité et imparité :

Si $x(t)$ est un signal pair alors $X(f)$ est à valeurs réelles et

$$X(f) = 2 \int_0^{+\infty} x(t) \cos(2\pi \omega t) dt$$

Si $x(t)$ est un signal impair alors $X(f)$ est à valeurs imaginaires pures et

$$X(f) = -2i \int_0^{+\infty} x(t) \sin(2\pi\omega t) dt$$

Dérivation : $\frac{\partial x(t)}{\partial t} \Longleftrightarrow j2\pi f X(f)$

Décalage temporel : $x(t - t_o) \Longleftrightarrow \exp(j2\pi f t_o) X(f)$

Cette propriété montre qu'il y a déphasage fréquentiel.

L'amplitude A_f ne change pas et la phase est modifiée de $-j2\pi f t_o$.

Décalage fréquentiel : $\exp(j2\pi f_o t) x(t) \Longleftrightarrow X(f - f_o)$

Inversion temporelle : $x(-t) \Longleftrightarrow X(-f)$

Conjugaison complexe : $x^*(t) \Longleftrightarrow X^*(-f)$

Changement d'échelle : La contraction dans le domaine temporel ($a \geq 1$) correspond à la dilatation dans le domaine fréquentiel et inversement :

$$x(at) \Longleftrightarrow \frac{1}{a} X\left(\frac{f}{a}\right)$$

4.2.3 La transformation de Fourier discrète (TFD)

Définition

La transformée de Fourier d'un signal discret $x(n)$ est donnée par l'expression suivante :

$$X_e(f) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j2\pi n f}$$

- La TF d'un signal discret est une fonction continue de la variable continue f .
- La fonction $X(f)$ est une fonction complexe. Si $x(n)$ est réel, pour un intervalle de fréquence $f \in [0, 1/2]$: le spectre d'amplitude $|X(f)|$ est une fonction paire et le spectre de phase $\arg(X(f))$ est une fonction impaire.

Conditions d'existence de la TFD

L'existence de la TFD est liée à la convergence absolue de la série $x(n)$. Donc, la TF d'un signal discret $x(n)$ existe si le signal est absolument sommable :

$$\sum_{n=-\infty}^{+\infty} |x(n)| < \infty$$

Pour un signal échantillonné à la fréquence Fe , sa TFD ($Xe(f)$) est périodique de période Fe
 $\Rightarrow$ l'information fréquentielle est contenue dans la bande $[+Fe/2, -Fe/2]$:

La transformation de Fourier discrète inverse

L'expression de la TFD inverse est donnée par l'intégrale suivant :

$$x(n) = \int_{-1/2}^{1/2} X(f) e^{j2\pi n f} df$$

4.2.4 Propriétés de la transformée de Fourier discrète (TFD)

Linéarité : $ax(n) + by(n) \iff aX(f) + bY(f)$

Décalage temporel : $x(n - n_o) \iff X(f) e^{-j2\pi f n_o}$

Décalage fréquentiel : $x(n) e^{j2\pi f_o n} \iff X(f - f_o)$

Dérivation : $\partial x(n) / \partial n \iff j2\pi f X(f)$

Relation de Parseval (conservation de l'énergie)

$$\sum_{n=-\infty}^{+\infty} |x(n)|^2 = \int_{-1/2}^{1/2} |X(f)|^2 df$$

Relations de Plancherel

$$\left\{ \begin{array}{l} x(n) * y(n) \iff X(f) \cdot Y(f) \\ x(n) \cdot y(n) \iff X(f) * Y(f) = \int_{-1/2}^{1/2} X(u) Y(f - u) \end{array} \right.$$

4.3 Manipulation

4.3.1 La transformée de Fourier d'un signal triangulaire

Soit $x(t)$ le signal triangulaire défini par :

$$x(t) = \begin{cases} 1 - |t| & \text{Si } t \in [-1, +1] \\ 0 & \text{Sinon} \end{cases}$$

Ecrire un script **MatLab** commenté pour réaliser les opérations suivantes :

1. Tracer le graphe du signal $x(t)$.
2. Tracer la transformée de Fourier $X(f)$ du signal $x(t)$ calculée de manière formelle (*expression algébrique*).
3. Tracer la dérivée $x'(t) = \partial x(t)/\partial t$ (là où elle existe) calculée de manière formelle.
4. Vérifier que la dérivée $x'(t)$ s'exprime en fonction du signal $p(t)$ comme suit :

$$x'(t) = p(t + 1/2) - p(t - 1/2)$$

Où $p(x)$ est le signal porte défini comme suit :

$$p(x) = \begin{cases} 1 & \text{Si } x \in] - 1/2, 1/2[\\ 0 & \text{Sinon} \end{cases}$$

5. Tracer la transformée de Fourier de cette porte $P(f)$ calculée de manière formelle.
6. En utilisant le signal $P(f)$, donner la transformée de Fourier du signal $x'(t)$.
7. Tracer la transformée de Fourier $X'(f)$.
8. En utilisant la convolution des signaux (formule $x(t) = p(t) * p(t)$), recalculer la transformée de Fourier du signal $x(t)$. Que concluez-vous ?

4.3.2 La transformée de Fourier d'un signal complexe

Soit $x(t)$ un signal exponentiel complexe :

$$x(t) = Ae^{j\omega t}$$

Ecrire un programme **MatLab** qui permet de réaliser les opérations suivantes :

1. Générer et tracer 128 échantillons d'une somme de deux exponentiels complexes de fréquence $f_1 = 1680 \text{ Hz}$ et $f_2 = 1780 \text{ Hz}$ échantillonnées à $f_e = 8 \text{ kHz}$, et d'amplitude $A = 1$.
2. Tracer la partie réelle et imaginaire du signal $x(n)$.
3. Tracer la transformée de Fourier discret (TFD) $X(f)$ des premiers échantillons ($M = 32$) du signal $x(k)$ sur N points fréquentiels ($N = 256$ et 1024). Qu'observez-vous ?

4. Les sinusoides sont-elles résolues ? Interpréter les résultats obtenus en calculant $f_2 - f_1$ et $f_e/32$?
5. Refaire les calculs pour $M = 128$ échantillons et $N = 256$ points fréquentiels. Est-ce que les sinusoides sont résolues ?
6. Comment interpréter les résultats obtenus ?

Analyse et synthèse de filtres numériques (RIF et RII)

5.1 Objectif du TP

Un filtre numérique est un élément qui modifie le contenu spectral du signal d'entrée en atténuant ou éliminant certaines composantes spectrales non désirées. Contrairement aux filtres analogiques, qui sont réalisés à l'aide d'un arrangement de composants électroniques (résistance, condensateur, etc.), les filtres numériques sont réalisés soit par des circuits intégrés dédiés ou par des processeurs programmables (microprocesseur, microcontrôleur, FPGA etc...). Ces filtres numériques sont classés en deux catégories : filtre à réponse impulsionnelle finie (RIF) et filtres à réponse impulsionnelle infinie (RII).

L'objectif de ce TP est d'**analyser et synthétiser des filtres numériques RIF et RII**. Nous intéressons, plus particulièrement, à la synthèse de filtres RIF par la *méthode de la fenêtre* (rectangulaire, triangulaire, de Hamming et de Kaiser), et de filtres RII par la *transformation bilinéaire*.

5.2 Rappels

5.2.1 Filtre numérique : définition et caractéristiques

Un filtre numérique est un *système Linéaire Discret (SLD)* qui modifie la représentation temporelle et fréquentielle de signaux discrets. Par exemple, pour des signaux radio ou des images ces filtres peuvent être utilisés pour réduire le bruit ou modifier certaines plages de fréquences.

Un filtre numérique peut être représenté par plusieurs types de spécifications :

1. **Fonction de transfert en z :**

La fonction de transfert du filtre $H(z)$ est rationnelle :

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^N b_i \cdot z^{-i}}{1 + \sum_{i=0}^N a_i \cdot z^{-i}}$$

2. **Réponse impulsionnelle :** La transformée en z inverse de $H(z)$:

$$H(z) = \sum_{n=0}^{\infty} h(n) \cdot z^{-n}$$

3. **Equation aux différences :**

$$y(n) = \sum_{i=0}^N b_i \cdot x(n-i) - \sum_{i=0}^N a_i \cdot y(n-i)$$

Remarques

- Si $a_n = 0$, le filtre s'appelle non récursif ou RIF (Réponse Impulsionnelle Finie) ;
- Si $a_n \neq 0$, le filtre s'appelle récursif ou RII (Réponse Impulsionnelle Infinie) ;
- Si les pôles de $H(z)$ sont à l'intérieur du cercle unité, le filtre est stable ;
- Si la réponse impulsionnelle est causale $h(n) = 0, \forall n < 0$, le filtre est causal.

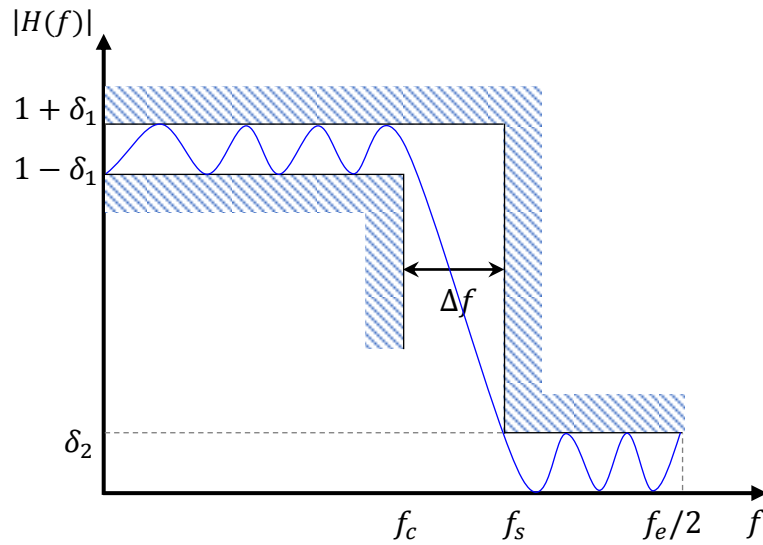
Gabarit d'un filtre numérique

Le gabarit d'un filtre numérique $H(e^{j\omega})$ est défini par l'utilisateur parmi les quatre catégories suivantes :

1. Filtre passe-bas
2. Filtre passe-haut
3. Filtre coupe-bande
4. Filtre passe-bande

Avec :

- F_e la fréquence d'échantillonnage.
- Δf la largeur de la zone de transition.
- δ_1 l'amplitude des ondulations en bande passante.
- δ_2 l'amplitude des oscillations en bande atténuée.



5.2.2 Filtre à Réponse Impulsionnelle Finie (RIF)

Définition et caractéristiques

Les filtres RIF sont des systèmes à réponse impulsionnelle finie, de fonction de transfert $H(z)$ défini par la relation suivante :

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k}$$

dont les coefficients $h(k)$ sont tels que :

$$\begin{cases} h(k) \neq 0 & \text{pour } k \in [0, N-1] \\ h(k) = 0 & \text{pour } k \notin [0, N-1] \end{cases}$$

L'équation de récurrence qui liant l'entrée et la sortie montre clairement le caractère non-récursif du filtre :

$$h(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

Les *principales caractéristiques* des filtres RIF sont :

- Les filtres RIF sont toujours stables, car la sortie revient toujours à zéro après suppression de l'excitation ($\sum_{k=0}^{N-1} |h(k)| < \infty$) ;
- Les méthodes de synthèse des filtres RIF permettant de dériver n'importe quelle réponse fréquentielle ;
- Pour le même nombre de coefficients, la bande de transition des filtres RIF est toujours plus large par rapport aux filtres RII ;
- Les filtres RIF ont un temps de propagation de groupe constant et une absence de distorsion harmonique dans le signal ;
- Les filtres RIF sont plus faciles d'implémenter dans un système numérique de traitement.

5.2.3 Synthèse de filtre à RIF par la méthode de la fenêtre

Faire la synthèse d'un filtre c'est rechercher la fonction de transfert $H(z)$ correspondant à la spécification d'un gabarit imposé.

La méthode de la fenêtre consiste à appliquer une fenêtre de taille N au filtre idéal équivalent ;

Soit un filtre numérique idéal, périodique de période 2π de fonction de transfert $H(e^{j\omega})$:

$$H(e^{j\omega}) = \sum_{-\infty}^{+\infty} h(n) e^{-jn\omega}$$

et de réponse impulsionnelle $h(n)$

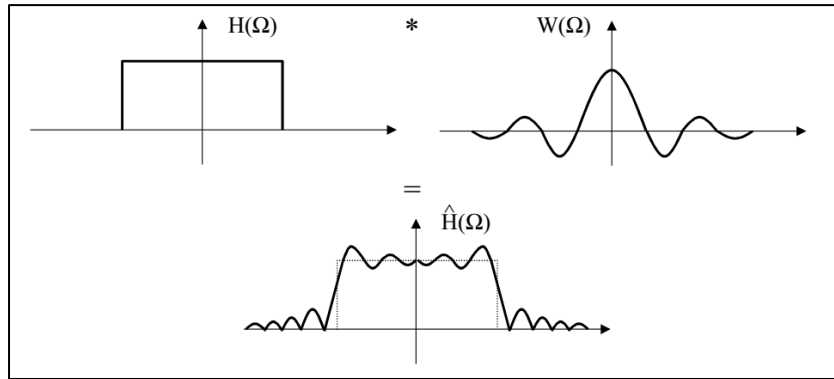
$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{jn\omega} d\omega$$

Ce filtre est non causal et à réponse impulsionnelle infinie. La manière la plus simple d'obtenir un filtre RIF approchant $H(e^{j\omega})$ est de limiter $h(n)$ par :

$$\hat{h}(n) = \begin{cases} h(n) \cdot w(n) & \text{Si } 0 \leq n < N \\ 0 & \text{Sinon} \end{cases}$$

$$\begin{aligned} \hat{H}(e^{j\omega}) &= \sum_{n=0}^{N-1} h(n) \cdot w(n) \cdot e^{-jn\omega} \\ &= H(e^{j\omega}) * W(e^{j\omega}) \end{aligned}$$

Principales fenêtres



Rectangulaire

$$w(n) = \begin{cases} 1 & \text{Si } 0 \leq n < N \\ 0 & \text{Sinon} \end{cases}$$

Hanning

$$w(n) = \begin{cases} 0.5 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) & \text{Si } 0 \leq n < N \\ 0 & \text{Sinon} \end{cases}$$

Hamming

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) & \text{Si } 0 \leq n < N \\ 0 & \text{Sinon} \end{cases}$$

Blackman

$$w(n) = \begin{cases} 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{24\pi n}{N-1}\right) & \text{Si } 0 \leq n < N \\ 0 & \text{Sinon} \end{cases}$$

5.2.4 Filtre à Réponse Impulsionnelle Infinie (RII)

Définition et caractéristiques

Les filtres RII sont des systèmes caractérisés par des réponses impulsionnelles de durée infinie et donc les coefficients $h(k)$ sont non nuls sur l'intervalle $[0, +\infty[$. Ceci est réalisé par la présence de pôles dans la fonction de transfert du filtre :

$$H(z) = \frac{\sum_{k=0}^{p-1} b_k z^{-k}}{\sum_{i=0}^{m-1} a_i z^{-i}}$$

ce qui traduit par l'équation suivante :

$$y(n) = \sum_{k=0}^{p-1} b_k x(n-k) - \sum_{i=1}^{m-1} a_i y(n-i)$$

Les *principales caractéristiques* des filtres RII sont :

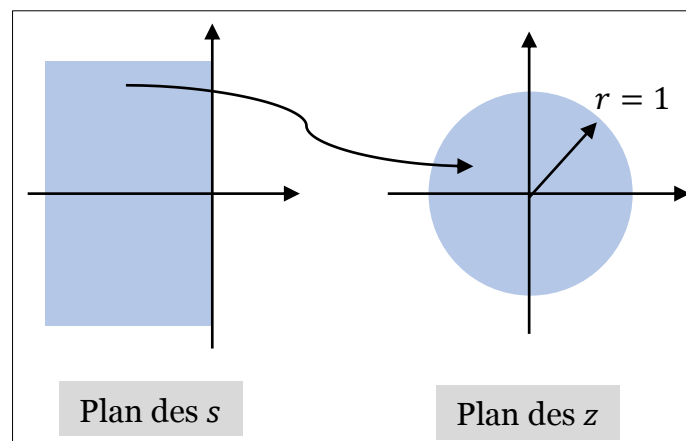
- Les filtres RII ont une instabilité potentielle due à des pôles de $H(z)$ situés en dehors du cercle unité ;
- La réponse impulsionnelle infinie permet d'obtenir un filtrage plus sélectif qu'un filtre RIF pour une quantité de calcul inférieure ;
- Les filtres RII ont une bande de transition étroite ;
- Les filtres RII ont une complexité plus faible qu'un filtre RIF à sélectivité équivalente.

5.2.5 Synthèse de filtre à RII par la transformée bilinéaire

Une méthode pour faire la synthèse du filtre $H(z)$ à partir du gabarit numérique consiste à transformer ce gabarit en un gabarit analogique. On choisit une fonction analogique modèle et on détermine ses paramètres afin d'obtenir la fonction de transfert $H(p)$ du filtre respectant le gabarit analogique souhaité. Enfin, on obtient la fonction de transfert $H(z)$ du filtre numérique par application de la transformation bilinéaire :

$$H(z) = \begin{cases} H(p) \\ p = \frac{2}{T_e} \frac{1 - z^{-1}}{1 + z^{-1}} \end{cases}$$

Avec : T_e fréquence d'échantillonnage.



Les pôles de l'équation sont définis par cette équation :

$$z = \frac{2/T_e + s}{2/T_e - s}$$

Si s a une partie réelle négative, z est de module inférieur à 1 $\Rightarrow$ conservation de la stabilité.

Remarque : La transformation bilinéaire entraîne une relation non linéaire entre les fréquences analogiques f_a et les fréquences numérique f_d : $f_d = \frac{1}{\pi T_e} \tan^{-1}(\pi f_a T_e)$

5.3 Manipulation

5.3.1 Synthèse d'un filtre numérique *RIF* par la méthode de la fenêtre

Pour concevoir un filtre numérique RIF qui répond au cahier des charges (c'est-à-dire un filtre qui possédant les caractéristiques du gabarit à respecter), il est possible de calculer l'ordre N par la formule suivante :

$$N = \frac{-20 \log(\sqrt{\delta_1 \delta_2}) - 13}{14.6(f_s - f_p)/f_c}$$

	filtre RIF passe-haut	filtre RIF passe-bande
Fréquence de coupure	2400 Hz	//
Bande passante	//	1500-2500 Hz
Largeur de la zone de transition $\Delta f = f_s - f_c$	1000 Hz	500 Hz
Amplitude des ondulations en bande passante δ_1	0.01 dB	0.1 dB
Amplitude des oscillations en bande atténuée δ_2	> 40 dB	> 60 dB
Fréquence d'échantillonnage f_e	20 kHz	10 kHz

1. Écrire le script **MatLab** permettant de visualiser les fonctions de transfert des filtres RIF obtenues en utilisant les différents types de fenêtres disponibles.
2. Quels sont les avantages et les inconvénients de chacune fenêtre ?
3. Écrire le script **MatLab** permettant de visualiser la réponse impulsionnelle des filtres RIF obtenue.
4. Faire varier l'ordre du filtre dans une plage incluant l'ordre optimal N . Observer son influence sur la position de la fréquence de coupure et la réponse impulsionnelle du filtre.
5. Quelles remarques peut-on faire sur la réponse impulsionnelle (*allure générale, nombre d'oscillations en fonction de l'ordre, ...*) ?

5.3.2 Synthèse d'un filtre numérique *RII* par la transformée bilinéaire

On désire réaliser un filtre numérique $H(z)$ équivalent à un filtre analogique de Chebyshev $H(j\omega)$. Le filtre numérique obéissant aux caractéristiques ci-dessous :

$$\left\{ \begin{array}{l} \text{Fréquence de la bande passante : } f_c = 1\text{kHz} \\ \text{Fréquence de la coupe bande : } f_p = 3\text{kHz} \\ \text{Amplitude des ondulations en bande passante : } \delta_1 = 1\text{dB} \\ \text{Amplitude des oscillations en bande atténuée : } \delta_2 = -20\text{dB} \\ \text{Fréquence d'échantillonnage : } f_e = 10\text{kHz} \end{array} \right.$$

1. Montrer par la méthode bilinéaire que la fonction du transfert du filtre $H(z)$ est donnée par la relation suivante :

$$H(z) = \frac{0.079(z+1)^2}{z^2 - 1.2z + 0.516}$$

2. Écrire le script `MatLab` permettant de visualiser la réponse fréquentielle en module et en phase du filtre numérique.
3. Trouver la relation entre l'échelle des abscisses de la commande `plot` de `MatLab` et les fréquences. Le filtre respecte-t-il le gabarit spécifié ?
4. Quelles est la relation entre la fréquence d'échantillonnage et la fréquence de coupure ?
5. Écrire le script `MatLab` permettant de visualiser la réponse impulsionnelle du filtre numérique obtenue.
6. Tracez dans le plan complexe les pôles et les zéros du filtre. Le filtre est-il stable ? Que concluez-vous ?

Processus aléatoires

6.1 Objectif du TP

Un processus aléatoire (PA) (ou *processus stochastique*) est un processus qui décrit l'évolution d'une grandeur aléatoire en fonction du temps (ou de l'espace). Les applications principales des processus aléatoires sont : l'administration des réseaux et bien entendu dans les télécommunications. L'étude des caractéristiques de processus aléatoire on fait appel aux théories de probabilités et de statistiques dont elle constitue l'un des objectifs les plus profonds. Leur étude soulève des problèmes mathématiques intéressants et souvent très difficiles.

Ce TP aura pour objet de **familiariser les étudiants avec quelques notions de bases de processus aléatoires** : calculer les propriétés statistiques d'un ensemble de N variables aléatoire uniforme et gaussien ; calculer la fonction d'autocorrélation et la densité spectrale de puissance (DSP) des signaux aléatoire.

6.2 Rappels

Définition

Un processus aléatoire $X(t, \zeta)$ est un processus qui pour chaque temps t , $X(t, \zeta)$ est une variable aléatoire. Par ailleurs, pour $\zeta = \zeta_o$, $X(t, \zeta)$ est une fonction déterministe qui traduit une courbe appelée *trajectoire du signal*.

Exemple :

$$X(t, \zeta) = A \cos(\omega t + \theta)$$

Si θ est une variable aléatoire qui prend des valeurs entre $[0, 2\pi]$, nous pouvons avoir plusieurs réalisations. Par exemple, pour une valeur donnée θ_o de θ , $X(t, \zeta)$ est une fonction déterministe qui traduit une *trajectoire du signal aleatoire*.

6.2.1 Moyennes, moment et variance d'un processus aléatoire

Esperance mathématiques ou moyenne d'un PA

On appelle espérance mathématique ou valeur moyenne d'un PA la quantité suivante définie comme suit :

$$m_X(t) = E[X(t, \zeta)] = \begin{cases} \int_{-\infty}^{\infty} x(t) F_1(x, t) dx & \text{Si X continue} \\ \sum_k x_k F_k(x, t) & \text{Si X discrète} \end{cases}$$

La moyenne $m_X(t)$ est définie comme moyenne au sens du calcul des probabilités.

- Si $m_X(t) = 0$, on dit que le PA est centré.
- Si tel n'est pas le cas, le signal $X(t, \zeta) - m_X(t)$ est centré.
- Si $F_1(x, t) = F_1(x)$, alors : $m_X(t) = m$ est indépendant du temps t .

Propriétés

Si X et Y sont deux VA définies sur le même univers Ω , admettant une espérance :

1. $E[X + Y] = E[X] + E[Y]$
2. $E[aX] = a.E[X], \forall a \in \mathbb{R}$
3. Si $X \geq 0$, alors : $E[X] \geq 0$

Moments d'ordre n

Le moment d'ordre n d'un PA est défini comme suit :

$$E[X(t, \zeta)^n] = \begin{cases} \int_{-\infty}^{\infty} x(t)^n F_1(x, t) dx & \text{Si X continue} \\ \sum_k x_k^n F_k(x, t) & \text{Si X discrète} \end{cases}$$

Variance d'un processus aléatoire

La variance d'un PA reel $X(t, \zeta)$ est définit comme suit :

$$\sigma_X^2(t) = \text{Var}[X] = \begin{cases} \int_{-\infty}^{\infty} [x - E[X]]^2 F_1(x, t) dx & \text{Si } X \text{ continue} \\ \sum_k [x_k - E[X]]^2 F_k(x, t) & \text{Si } X \text{ discrète} \end{cases}$$

Si $\sigma_X^2(t) = 0$, le signal perd son caractère aléatoire. Il s'agit alors d'un signal déterministe.

Il est important de remarquer que l'espérance mathématique, le moment d'ordre n et la variance d'un processus aléatoire sont calculer à un instant t particulier et sont déduits uniquement à partie de la fonction de densité de probabilité $F_1(x, t)$.

6.2.2 Autocovariance, autocorrélation et stationnarité

Fonction d'autocovariance

La fonction d'auto covariance d'un processus aléatoire réel $X(t, \zeta)$, notée $C_X(t_1, t_2)$, est définie par :

$$C_X(t_1, t_2) = \text{cov}(X(t_1, \zeta), X(t_2, \zeta)) = E[X(t_1, \zeta), X(t_2, \zeta)] - m_X(t_1)m_X(t_2)$$

Dans cette définition $E[X(t_1, \zeta), X(t_2, \zeta)]$ est calculé à partir de $F_2(x_1, x_2; t_1, t_2)$.

En outre, nous remarquons que lorsque $t = t_1 = t_2$, nous retrouvons la variance du PA, c'est-à-dire : $C_X(t_1, t_2) = \sigma_X^2(t)$.

Cas des signaux complexe : Si $X(t, \zeta)$ est un PA complexe alors :

$$C_X(t_1, t_2) = E[X(t_1, \zeta), X^*(t_2, \zeta)] - m_X(t_1)m_X^*(t_2)$$

Cas des signaux discrets : Si $X(t, \zeta)$ est un PA discret alors :

$$C_X(n_1, n_2) = E[X(n_1), X(n_2)]$$

Fonction d'autocorrélation

La fonction d'autocorrélation d'un PA réel $X(t, \zeta)$, notée $\gamma_X(t_1, t_2)$, est définie par :

$$\gamma_X(t_1, t_2) = E[X(t_1, \zeta), X(t_2, \zeta)]$$

Cette fonction peut être obtenue à partir de la fonction d'autocovariance comme suit :

$$\gamma_X(t_1, t_2) = C_X(t_1, t_2) + m_X(t_1)m_X(t_2)$$

Cas des signaux complexe : Si $X(t, \zeta)$ est un PA complexe alors :

$$\gamma_X(t_1, t_2) = E[X(t_1, \zeta), X^*(t_2, \zeta)]$$

Cas des signaux discrets : Si $X(t, \zeta)$ est un PA discret alors :

$$\gamma_X(n_1, n_2) = C_X(n_1, n_2) - m_X(n_1)m_X(n_2)$$

Si $X(t, \zeta)$ est un PA centré alors : $\gamma_X(t_1, t_2) = C_X(t_1, t_2)$.

si $\gamma_X(t_1, t_2) = 0$, alors $X(t_1, \zeta)$ et $X(t_2, \zeta)$ sont orthogonaux ($X(t_1, \zeta) \perp X(t_2, \zeta)$).

Stationnarité des processus aléatoires

La notion de stationnarité est liée au cours du temps de propriétés statistiques d'un PA. Elle peut être définie de manières différentes :

Stationnaire au sens strict (SSS) : Un PA $X(t, \zeta)$ est dit SSS, si toutes ses propriétés statistiques sont invariantes dans toute translation de l'origine du temps. Ceci signifie que $X(t, \zeta)$ et $X(t + \tau, \zeta)$ ont les mêmes propriétés statistiques. C'est-à-dire :

$$\forall n \quad F_n(x_1, x_2, \dots, x_n; t_1, t_2, \dots, t_n) = F_n(x_1, x_2, \dots, x_n; t_1 + \tau, t_2 + \tau, \dots, t_n + \tau)$$

Stationnaire au sens large : un PA $X(t, \zeta)$ est dit stationnaire du second ordre ou stationnaire au sens large si :

$$\begin{cases} m_X(t) = m_X = \text{constante} \\ \gamma_X(t_1, t_2) = \gamma_X(t_1 - t_2) \end{cases}$$

La moyenne du PA est indépendante du temps et la fonction d'autocorrélation dépend uniquement de la différence $t_1 - t_2$.

La fonction d'autocovariance dépend uniquement du retard entre les instants t_1 et t_2 : $C_X(t_1, t_2) = C_X(t_1 - t_2)$

Remarque : Un SSL n'implique pas la SSS. L'inverse est également vrai

6.2.3 Intercovariance et intercorrélations

Fonction d'intercovariance

La fonction d'intercovariance de deux processus aléatoire $X(t)$ et $Y(t)$, notée $C_{XY}(t_1, t_2)$, est définie comme suit :

$$C_{XY}(t_1, t_2) = \text{cov}(X(t_1), Y(t_2)) = E[X(t_1)Y(t_2)] - m_X(t_1)m_Y(t_2)$$

Si $C_{XY}(t_1, t_2) = 0$, alors $X(t_1)$ et $Y(t_2)$ sont non corrélés. Rappelons à cet effet que deux PA non corrélés ne sont pas forcément indépendants.

En effet, soit deux PA $X(t)$ et $Y(t)$ indépendant alors :

$$\begin{aligned} C_{XY}(t_1, t_2) &= E[X(t_1)Y(t_2)] - m_X(t_1)m_Y(t_2) \\ &= E[X(t_1)]E[Y(t_2)] - m_X(t_1)m_Y(t_2) \\ &= 0 \end{aligned}$$

Cas des signaux complexe : Si $X(t)$ et $Y(t)$ sont deux PA complexe alors :

$$\begin{cases} C_{XY}(t_1, t_2) = E[X(t_1)Y^*(t_2)] - m_X(t_1)m_Y^*(t_2) \\ C_{YX}(t_1, t_2) = E[Y(t_1)X^*(t_2)] - m_Y(t_1)m_X^*(t_2) \end{cases}$$

Cas des signaux discrets : Si $X(n)$ et $Y(n)$ sont deux PA discrets, alors :

$$C_{XY}(n_1, n_2) = E[X(n_1)Y(n_2)] - m_X(n_1)m_Y(n_2)$$

Fonction d'intercorrélation

La fonction d'intercorrélation de deux processus aléatoire $X(t)$ et $Y(t)$, notée $\gamma_{XY}(t_1, t_2)$, est définie comme suit :

$$\gamma_{XY}(t_1, t_2) = E[X(t_1)Y(t_2)]$$

Cette fonction peut être obtenue à partir de la fonction d'intercovariance comme suit :

$$\gamma_{XY}(t_1, t_2) = C_{XY}(t_1, t_2) + m_X(t_1)m_Y(t_2)$$

Cas des signaux complexe : Si $X(t)$ et $Y(t)$ sont deux PA complexe alors :

$$\begin{cases} \gamma_{XY}(t_1, t_2) = E[X(t_1)Y^*(t_2)] \\ \gamma_{YX}(t_1, t_2) = E[Y(t_1)X^*(t_2)] \end{cases}$$

Cas des signaux discrets : Si $X(n)$ et $Y(n)$ sont deux PA discrets, alors :

$$\gamma_{XY}(n_1, n_2) = C_{XY}(n_1, n_2) - m_X(n_1)m_Y(n_2)$$

Orthogonalité de deux PA

Deux processus aléatoire $X(t_1)$ et $Y(t_2)$ sont dits orthogonaux ($X(t_1) \perp Y(t_2)$) si $\gamma_{XY}(t_1, t_2) = 0$

Si les deux PA sont centrés alors : $\gamma_{XY}(t_1, t_2) = C_{XY}(t_1, t_2)$

6.2.4 Matrice de corrélation et de covariance

Soit $\underline{X}(t)$ un vecteur aléatoire :

$$\underline{X}(t) = \begin{bmatrix} X_1(t) \\ X_2(t) \\ \vdots \\ X_N(t) \end{bmatrix} \quad (6.1)$$

Moyenne du VA : la moyenne du vecteur $\underline{X}(t)$ est définie comme suit :

$$E[\underline{X}(t)] = \begin{bmatrix} E[X_1(t)] \\ E[X_2(t)] \\ \vdots \\ E[X_N(t)] \end{bmatrix} \quad (6.2)$$

Matrice de corrélation : la matrice de corrélation du vecteur $\underline{X}(t)$ est définie par :

$$\Gamma_{\underline{X}}(t_i, t_j) = E[\underline{X}(t_i)\underline{X}^H(t_j)]$$

Matrice de covariance : la matrice de covariance du vecteur $\underline{X}(t)$ est définie par :

$$C_{\underline{X}}(t_i, t_j) = \Gamma_{\underline{X}}(t_i, t_j) - E[\underline{X}(t_i)]E[\underline{X}^H(t_j)]$$

6.2.5 Densité spectrale de puissance

La densité spectrale de puissance d'un processus aléatoire correspond à la répartition de la puissance moyenne du processus dans l'espace de fréquences. Cette densité est obtenue à partir de la transformée de Fourier de la fonction d'autocorrélation et réciproquement.

$$\begin{cases} \Phi_X(f) = \int_{-\infty}^{+\infty} \gamma_X(\tau) e^{-2\pi j f \tau} d\tau \\ \gamma_X(\tau) = \int_{-\infty}^{+\infty} \Phi_X(f) e^{2\pi j f \tau} df \end{cases}$$

Ces relations sont appelées relations d'*EINSTEIN-WIENER-KINTCHINE*.

6.2.6 Ergodicité

Un processus stochastique $X(t, \zeta)$ est ergodique si toutes les moyennes temporelles existent et ont même valeur pour tout échantillon.

Un processus est dit à moyenne ergodique si :

$$E[X(t, \zeta)] = m_X(t) = \langle X(t, \zeta) \rangle$$

Un processus est dit à autocorrélation ergodique si :

$$\gamma_X(t, \zeta) = \langle X(t, \zeta)X(t + \tau, \zeta) \rangle = \bar{\gamma}_X(t, \zeta)$$

6.3 Manipulation

6.3.1 Etude d'un processus aléatoire uniforme et gaussien

Ecrire un programme **MatLab** qui permet de réaliser les opérations suivantes :

1. Générer et représenter graphiquement N points d'un processus aléatoire uniforme de moyenne nulle et de variance σ^2 égale à 1.
2. Générer et représenter graphiquement N points d'un processus aléatoire gaussien de moyenne nulle et de variance σ^2 égale à 1.
3. Tracer et commenter l'histogramme de chaque processus aléatoire.
4. Générer un ensemble de N variables aléatoires $\underline{X}(t)$ gaussiennes indépendantes et identiquement distribuées de variance σ^2 (bruit blanc gaussien).
5. Calculer la matrice de covariance du vecteur $\underline{X}$ correspondant.
6. Estimer la fonction de corrélation de ce processus et comparer à la fonction de corrélation théorique. ? Que concluez-vous ?

6.3.2 Etude des propriétés statistiques d'un signal aléatoire

On considère le signal aléatoire $x(t) = A \cos(\omega_o(t) + \Theta)$, où A et ω_o sont constantes et Θ est une variable aléatoire uniforme sur $[0, \pi]$.

Ecrire un programme **MatLab** qui permet de réaliser les opérations suivantes :

1. Générer et représenter graphiquement N points du signal aléatoire $x(t)$.
2. Calculer la moyenne du signal $E[x(t)]$.

3. Calculer la variance du signal.
4. Calculer la fonction d'autocorrélation.
5. Calculer et représenter graphiquement la densité spectrale de puissance $\Phi_X(f)$.
6. Quelle est la puissance du signal $x(t)$?

Annexe **A**

Annexe : Les fonctions usuelles de MatLab

A.1 Commandes générales

Lancement et arrêt de MatLab	
<code>matlab.rc</code>	Fichier M de démarrage principal
<code>quit</code>	Permet de quitter Matlab
<code>startup.m</code>	Fichier M de démarrage secondaire
Gestion de l'environnement	
<code>addpath</code>	Ajoute un répertoire à la liste des chemins de recherche de Matlab
<code>doc</code>	Charge la documentions hypertexte
<code>help</code>	Lance l'aide en ligne
<code>lasterr</code>	Fournit le dernier message d'erreur
<code>lookfor</code>	Lance une recherche par mot-clé
<code>path</code>	Affiche la liste des chemins de recherche
<code>profile</code>	Mesure le temps d'exécution d'un fichier M
<code>rmpath</code>	Enlève un répertoire de la liste des chemins de recherche MatLab
<code>type</code>	Liste le contenu d'un fichier
<code>version</code>	Affiche la version de Matlab en cours d'exécution
<code>what</code>	Donne la liste des fichiers M, MEX et MAT présents dans le répertoire courant
<code>whatsnew</code>	Affiche les fichiers <code>readme</code> pour Matlab et ses boîtes à outils
<code>which</code>	Localiser les fichiers et les fonctions

Gestion des variables et de l'espace mémoire	
clear	Efface les variables choisies de la mémoire
disp	Afficher du texte ou des tableaux
load	Charge des variables depuis un fichier
pack	Lance la défragmentation de l'espace mémoire de travail
save	Sauvegarde des variables dans un fichier
size	Fournit les dimensions d'une matrice ou vecteur
who, whos	Liste toutes les variables connues de l'espace de travail
Gestion de la fenêtre de commande	
echo	Affiche les commandes du fichier M couramment exécuté
format	Définit le format d'affichage des nombres
more	Active et désactive le mode d'affichage par page

A.2 Opérateurs et caractères spéciaux

Opérateurs et caractères usuels	
+	Addition de réels ou de matrices
−	Soustraction de réels ou de matrices
*	Produit de réels ou de matrices
.*	Produit élément par élément de matrices
\	Division à gauche de réels ou de matrices
/	Division à droite de réels ou de matrices
.\	Division à gauche élément par élément de matrices
./	Division à droite élément par élément de matrices
.	Point décimal
..	Désigne le répertoire parent du répertoire courant
...	Symbole de continuation (pour terminer une instruction ou une expression à la ligne suivante)
%	Commentaire
'	Matrice adjointe (transposée + conjuguée) et délimiteur de chaîne de caractère
.'	Matrice transposée
!	Lance une commande système dont le nom suit, tout en restant sous MatLab
=	Affectation

Constantes et variables spéciales	
<code>ans</code>	Dernier résultat calculé
<code>computer</code>	Identifie le type d'ordinateur sur lequel est exécuté Matlab
<code>eps</code>	Précision relative des calculs menés en virgule flottante
<code>flops</code>	Compte le nombre d'opérations élémentaires
<code>i</code> ou <code>j</code>	Racine de -1 (<code>sqrt(-1)</code>) nombre imaginaire
<code>Inf</code> ou <code>inf</code>	Infini, Par exemple : 10/0
<code>inputname</code>	Nom de l'argument en entrée
<code>NaN</code> ou <code>nan</code>	Indéterminé, par exemple : 0/0
<code>nargin</code>	Nombre d'arguments passés en entrée à une fonction
<code>nargout</code>	Nombre d'arguments fournis en sortie par une fonction
<code>pi</code>	Désigne la constante trigonométrique $\pi = 3.141592653589793$
<code>realmax</code>	Plus grand petit réel positif (double précision)
<code>realmin</code>	Plus petit réel positif (double précision)
<code>varargin</code>	Liste d'arguments d'entrée contenant les arguments optionnels d'appel d'une fonction
<code>varargout</code>	Liste d'arguments de retour contenant les arguments optionnels de retour d'une fonction
<code>true</code> ou <code>1</code>	Vrai
<code>false</code> ou <code>0</code>	Faux
Opérateurs relationnels	
<code>a == b</code> ou <code>a eq b</code>	Test d'égalité qui retourne 1 si $a = b$ et 0 sinon
<code>isequal(a,b)</code>	Test d'égalité valable pour des vecteurs et des matrices
<code>a ~= b</code> ou <code>a ne b</code>	Test de différence qui retourne 0 si $a = b$ et 1 sinon
<code>a > b</code> ou <code>a gt b</code>	Test de supériorité qui retourne 1 si inégalité vérifiée et 0 sinon
<code>a < b</code> ou <code>a lt b</code>	Test d'infériorité qui retourne 1 si inégalité vérifiée et 0 sinon
<code>a >= b</code>	1 Si inégalité vérifiée et 0 sinon
<code>a <= b</code>	1 Si inégalité vérifiée et 0 sinon
Opérateurs logiques	
<code>~a</code> ou <code>not(a)</code>	Négative logique qui retourne 1 si $a = 0$ et 0 si $a=1$
<code>a & b</code> ou <code>and(a,b)</code>	ET logique ($0 \text{ ET } 0 \Rightarrow 0$; $0 \text{ ET } 1 \Rightarrow 0$; $1 \text{ ET } 1 \Rightarrow 1$)
<code>a b</code> ou <code>or(a,b)</code>	OU logique ($0 \text{ OU } 0 \Rightarrow 0$; $0 \text{ OU } 1 \Rightarrow 1$; $1 \text{ OU } 1 \Rightarrow 1$)
<code>xor(a,b)</code>	OU EXCLUSIF ($0 \text{ OU EXCL } 0 \Rightarrow 0$; $0 \text{ OU EXCL } 1 \Rightarrow 1$; $1 \text{ OU } 1 \Rightarrow 0$)

A.3 Matrices et vecteurs remarquables

Matrices et vecteurs remarquables	
<code>eye</code>	Matrice identité
<code>linspace</code>	Génère un vecteur ligne de valeurs également espacées
<code>logspace</code>	Génère un vecteur ligne de valeurs logarithmiquement espacées
<code>ones</code>	Matrices composées de 1
<code>rand</code>	Matrice (ou nombre) généré aléatoirement selon une distribution uniforme
<code>randn</code>	Matrice (ou nombre) généré aléatoirement selon une distribution normale
<code>zeros</code>	Matrice nulle

A.4 Opérations sur les vecteurs et les matrices

Opérations sur les vecteurs et les matrices	
<code>cat</code>	Permet la concaténation de tableaux
<code>diag</code>	Permet la définition de matrices diagonales ou l'extraction de la diagonale d'une matrice
<code>fliplr</code>	Permutation circulaire des colonnes
<code>flipud</code>	Permutation circulaire des lignes
<code>repmat</code>	Permet la réplication d'une valeur dans une matrice
<code>reshape</code>	Permet de transformer et redimensionner une matrice
<code>rot90</code>	Transformation équivalente à l'application successive de la transposition et de flipud
<code>tril</code>	Extrait le triangle inférieur d'une matrice
<code>triu</code>	Extrait le triangle supérieur d'une matrice
<code>numel(M)</code>	Nombre d'éléments de matrice M
<code>size(M)</code>	Taille d'une matrice (nb lignes $\times$ nb colonnes)
<code>rank(M)</code>	Rang d'une matrice (nb de lignes ou colonnes linéairement indépendants)
<code>trnspose(M) = M'</code>	Transposée d'une matrice
<code>M1+M2, M1-M2</code>	Addition et soustraction de deux matrices (<code>!dim(M1) = dim(M2)</code>)

$\alpha * M$	Multiplication d'une matrice par scalaire α
$M1 * M2$ ou <code>mtimes</code>	Multiplication de deux matrices (nb colonnes M1 = nb colonne M2)
$M1.M2$ ou <code>times</code>	Multiplication élément par élément ($!dim(M1) = dim(M2)$)
<code>inv(M)</code> ou M^{-1}	Inversion d'une matrice (M = matrice carrée)
<code>det(M)</code>	Déterminant d'une matrice
<code>trace(M)</code>	Trace d'une matrice
<code>[P, D]=eig(M)</code>	Valeurs et vecteurs propres d'une matrice
<code>mean(M)</code>	Moyenne d'une matrice
<code>std(M)</code>	Ecart-type d'une matrice
<code>[A,B,C]</code>	Juxtaposition horizontale des matrices A, B et C
<code>[A;B;C]</code>	Juxtaposition verticale des matrices A, B et C
$N=0*J$	Astuce pour obtenir une matrice N nulle de même taille que J sans connaître la taille de J
<code>sparse</code>	Création d'une matrice «creuse» (cf help <code>sparse</code>)
<code>length(V)</code>	Dimension (ou nombre d'éléments) du vecteur V
<code>min(V)</code>	Minimum d'un vecteur
<code>max(V)</code>	Maximum d'un vecteur
<code>norm(V)</code>	Norme du vecteur V
<code>mean(V)</code>	Moyenne arithmétique
<code>dot(V1, V2)</code>	Produit scalaire
<code>cross(V1, V2)</code>	Produit vectoriel
<code>expm(M)</code>	Exponentielle matricielle
<code>sqrtm(M)</code>	Racine carrée matricielle (cf. help <code>sqrtm</code>)
<code>logm(M)</code>	Logarithme matriciel (cf. help <code>logm</code>)
$M * V$	Image du vecteur colonne V par la matrice carrée M
<code>cumsum(V)</code>	Sommes cumulatives des entrées du vecteur V
<code>cumsum(M)</code>	Matrice des sommes cumulatives des colonnes de M
<code>cumprod(V)</code>	Produits cumulatifs des entrées du vecteur V
<code>cumprod(M)</code>	Matrice des produits cumulatifs des colonnes de M
<code>diag</code>	création ou extraction de la diagonale d'une matrice

A.5 Fonctions mathématiques usuelles

Fonctions mathématiques usuelles	
<code>abs</code>	Valeur absolue d'un réel ou module d'un complexe
<code>acos</code> , <code>asin</code> , <code>atan</code> , <code>acot</code>	Fonctions circulaires réciproques
<code>acosh</code> , <code>asinh</code> , <code>atanh</code> , <code>acoth</code>	Fonctions hyperboliques réciproques
<code>acsc</code> , <code>asec</code>	Fonctions réciproques de la cosécante et de la séquante circulaires
<code>acsch</code> , <code>asech</code>	Fonctions réciproques de la cosécante et de la séquante hyperboliques
<code>ceil</code>	Plus petit entier supérieur au nombre flottant considéré
<code>cos</code> , <code>sin</code> , <code>tan</code> , <code>cot</code>	Fonctions circulaires directes
<code>cosh</code> , <code>sinh</code> , <code>tanh</code> , <code>coth</code>	Fonctions hyperboliques directes
<code>csc</code> , <code>sec</code>	Fonctions directes de la cosécante et de la séquante circulaires
<code>csch</code> , <code>sech</code>	Fonctions directes de la cosécante et de la séquante hyperboliques
<code>exp</code>	Fonction exponentielle népérienne
<code>fix</code>	Arrondi entier vers 0
<code>floor</code>	Plus grand entier inférieur au nombre flottant considéré (partie entière)
<code>gcd</code>	Plus grand commun diviseur
<code>imag</code>	Partie imaginaire d'un nombre complexe
<code>log</code>	Fonction logarithme népérien
<code>log2</code>	Fonction logarithme en base 2
<code>log10</code>	Fonction logarithme en base 10
<code>mod</code>	Reste (signé) d'une division euclidienne (fonction modulo)
<code>rem</code>	Reste d'une division euclidienne
<code>round</code>	Arrondi au plus proche entier
<code>sign</code>	Signe
<code>sqrt</code>	Racine carrée

A.6 Représentations graphiques

Représentation graphique en dimension 2	
<code>plot</code>	Graphique en coordonnées linéaires
<code>stem</code>	Graphe d'un signal (style <i>discret</i>)
<code>loglog</code>	Graphique en coordonnées logarithmiques
<code>polar</code>	Graphique en coordonnées polaires
<code>semilogx</code> , <code>semilogy</code>	Graphique en coordonnées semi logarithmique
<code>hist</code>	Histogramme
<code>contour</code>	Trace les lignes de niveau
Représentation graphique en dimension 3	
<code>fill</code>	Représentation à base de polygones
<code>mesh</code>	Représentation d'une nappe par maillage
<code>plot3</code>	Représentation de segments et de points en dimension 3
<code>surf</code>	Représentation d'une nappe par une surface
<code>specgram</code>	Spectrogramme (utile pour le calcul du périodogramme cumulé)
<code>psd</code>	Densité spectrale de puissance
<code>quantiz</code>	Quantification uniforme du signal
Gestion des axes	
<code>axes</code>	Création d'axes en positions arbitraires
<code>axis</code>	Contrôle de l'apparence et de l'échelle des axes
<code>box</code>	Encadre le graphique avec des axes
<code>grid</code>	Crée un quadrillage superposé au graphique
<code>hold</code>	Gèle la figure graphique courante
<code>subplot</code>	Juxtapose des graphiques dans la même figure
<code>zoom</code>	Agrandit ou diminue une figure
Annotation d'une figure	
<code>gtext</code>	Permet de placer du texte sur une figure à l'aide de la souris
<code>legend</code>	Ajoute une légende à la figure
<code>text</code>	Place du texte sur la figure
<code>title</code>	Place un titre sur la figure
<code>xlabel</code> , <code>ylabel</code> , <code>zlabel</code>	Mettre une étiquette sur l'axe correspondant

A.7 Analyse de données

Fonctions élémentaires	
<code>convhull</code>	Enveloppe convexe
<code>cumprod</code>	Produit cumulé
<code>cumsum</code>	Somme cumulée
<code>dsearch</code>	Recherche du point le plus proche
<code>median</code>	Valeur médiane d'un tableau
<code>perms</code>	Liste toutes les permutations possibles
<code>prodv</code>	Effectue le produit des éléments d'un tableau
<code>sort</code>	Trie les éléments d'un tableau
<code>sortrows</code>	Trie les lignes d'une matrice par ordre croissant
<code>sum</code>	Effectue la somme des éléments d'un tableau
<code>trapz</code>	Intégration numérique selon la méthode des trapèzes
<code>tsearch</code>	Recherche d'un triangle de Delaunay englobant le point considère
<code>lin2mu</code>	Conversion à la loi μ
<code>mu2lin</code>	Conversion inverse (de la loi μ à la loi linéaire)
<code>dtrend</code>	Centrage d'un signal aléatoire
Filtrage, Corrélation et convolution	
<code>corrcoef</code>	Coefficient de corrélation
<code>cov</code>	Matrice de covariance
<code>xcorr</code>	Autocorrélation
<code>conv</code>	Convolution et produit de polynômes
<code>conv2</code>	Convolution bidimensionnelle
<code>deconv</code>	Déconvolution et division de polynôme
<code>filter</code>	Filtrage numérique avec filtre de type RIF (réponse impulsion-nelle finie) et avec filtres de type RII (réponse impulsionnelle infinie)
<code>filter2</code>	Filtrage numérique bidimensionnel
<code>freqz</code>	Réponse en fréquence d'un filtre
<code>fir1</code>	Conception d'un filtre à réponse impulsionnelle finie
<code>buttord,</code> <code>cheb1ord,</code> <code>cheb2ord,</code> <code>ellipord</code>	Synthèse des filtres numériques

Transformation de Fourier	
real	Partie réelle d'un nombre complexe
imag	Partie imaginaire d'un nombre complexe
conj	Conjugué d'un nombre complexe
abs	Module d'un nombre complexe
angle	Argument d'un nombre complexe
fft	Transformée de Fourier rapide monodimensionnelle
fft2	Transformée de Fourier rapide bidimensionnelle
fftshift	Permet le centrage du spectre d'une transformée de Fourier rapide
ifft	Transformée de Fourier inverse monodimensionnelle
ifft2	Transformée de Fourier inverse bidimensionnelle
nextpow2	Détermine la puissance de 2 immédiatement supérieure au nombre passé en argument
unwrap	Corrige les angles des phases