

Chapitre 2

TP Echantillonnage et Quantification

2.1 Rappels

L'échantillonnage et la quantification permettent l'acquisition et le codage d'un signal en vue d'un traitement ou d'un stockage. Le but de ce TP est d'illustrer les notions d'échantillonnage et de quantification et d'analyser leurs effets sur différents signaux.

2.1.1 Echantillonnage

L'échantillonnage consiste à représenter un signal à temps continu $s(t)$ par ses valeurs $s(nT_e)$ à des instant multiples de T_e , T_e étant la **période d'échantillonnage**. La **condition de Shannon** permet d'échantillonner un signal sans perte d'information aucune perte d'information si la fréquence d'échantillonnage $f_e = \frac{1}{T_e}$ est au moins 2 fois supérieure à la plus grande fréquence intervenant dans le spectre (répartition de la puissance du signal en fonction des fréquences) du signal. Si cette condition n'est pas respectée on observe un "repliement de spectre" (voir figure suivante). On note ce signal échantillonné $s_e(t)$:

$$s_e(t) = s(t) \times \sum_{-\infty}^{+\infty} \delta(t - nT_e)$$

Dans le plan fréquentiel :

$$S_e(f) = S(f) * \frac{1}{T_e} \sum_{-\infty}^{+\infty} \delta\left(f - \frac{n}{T_e}\right) = \frac{1}{T_e} \sum_{-\infty}^{+\infty} S\left(f - \frac{n}{T_e}\right)$$

L'échantillonnage à la période T_e a donc introduit une périodicité du spectre du signal échantillonné, de période F_e . Lorsqu'on observe des spectres de signaux échantillonnés, on préfère remplacer les fréquences (F) en Hertz par des fréquences normalisées par rapport à la fréquence d'échantillonnage (F_e) :

$$\tilde{f} = F/F_e$$

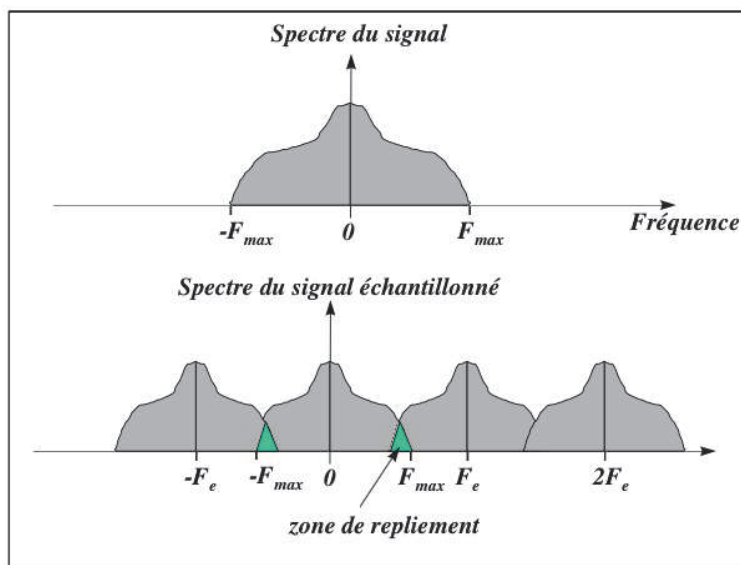


FIGURE 2.1 – Effet du repliement de spectre

Remarquons qu'après échantillonnage, le spectre entier du signal est disponible entre 0 et 1 en fréquence normalisée. En pratique, par raison de symétrie, on ne s'intéresse qu'à la partie centrale entre 0 et 0.5 en fréquence normalisée.

2.1.2 Restitution après échantillonnage du signal d'origine

Plusieurs méthodes d'interpolation existent pour restituer le signal entre deux points d'échantillonnage successifs. Entre deux points d'échantillonnage situés entre kT_e et $(k+1)T_e$, comment reconstituer le signal $\hat{s}(kT_e + \tau)$?

Par bloqueur :

$$\hat{x}(kT_e + \tau) = x(kT_e) \quad 0 \leq \tau \leq T_e$$

Par extrapolateur linéaire :

$$\hat{x}(kT_e + \tau) = x(kT_e) + \frac{\tau}{T_e} (x(kT_e) - x((k-1)T_e)) \quad 0 \leq \tau \leq T_e$$

Les marches d'escalier sont remplacées par des rampes.

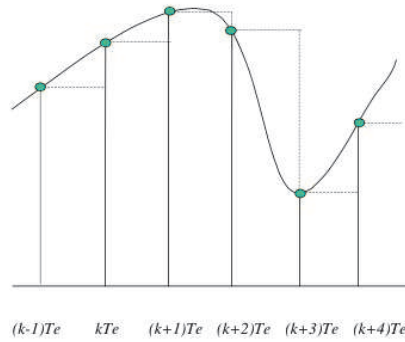


FIGURE 2.2 – Restitution par bloqueur

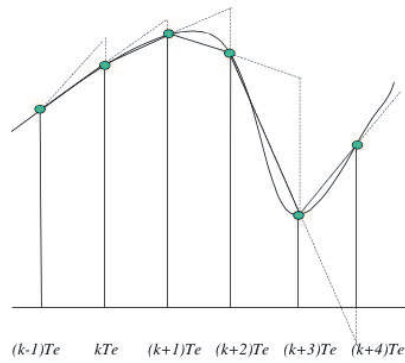


FIGURE 2.3 – Restitution par extrapolateur linéaire

Par interpolateur linéaire :

$$\hat{x}(kT_e + \tau) = x(kT_e) + \frac{\tau}{T_e} (x((k+1)T_e) - x(kT_e)) \quad 0 \leq \tau \leq T_e$$

Dans la restitution, il y aura un retard de T_e : il est nécessaire d'attendre la valeur en kT_e pour restituer l'intervalle $[(k-1)T_e, kT_e]$.

Par filtrage

Après échantillonnage, le spectre du signal est :

$$S_e(f) = F_e \sum_{n \in \mathbb{Z}} S(f - nF_e), \quad F_e = \frac{1}{T_e}$$

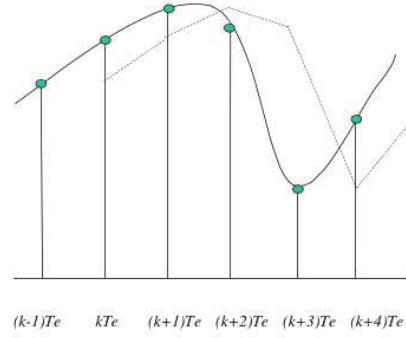


FIGURE 2.4 – Restitution par interpolateur linéaire

Si $x(t)$ est à spectre borné $(-F_M, +F_M)$, et si la condition de Shannon est vérifiée, on peut restituer le signal par filtrage de façon à ne récupérer que le spectre d'ordre 0 :

$$S(f) = \frac{1}{F_e} [S_e(f)]_{n=0}$$

Donc

$$S(f) = S_e(f)H_r(f)$$

où $H_r(f)$ est le filtre de restitution :

$$\begin{aligned} H_r(f) &= \frac{1}{F_e} \text{ si } f \in (-F_M, F_M) \\ H_r(f) &= 0 \text{ si } |f| \geq F_e - F_M \end{aligned}$$

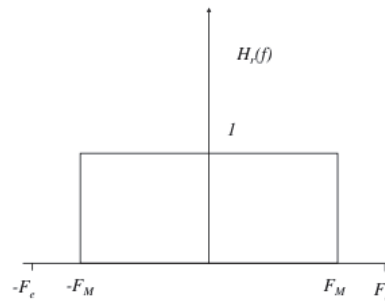


FIGURE 2.5 – Filtre de restitution

Dans le cas limite où $F_e = 2F_M$:

$$H_r(f) = \frac{1}{F_e} \Pi_{F_e}(f)$$

La réponse impulsionnelle associée s'écrit :

$$h_r(t) = \frac{\sin(\pi F_e t)}{\pi F_e t}$$

soit, pour le signal restitué :

$$s(t) = \sum_k s_e(kT_e) \frac{\sin\left(\frac{\pi}{T_e}(t - kT_e)\right)}{\frac{\pi}{T_e}(t - kT_e)}$$

La courbe restituée passe exactement par les valeurs des échantillons.

2.1.3 Quantification

La quantification consiste à allouer aux échantillons un nombre fini de valeurs d'amplitude. Deux grandeurs caractérisent un quantificateur : le nombre de niveaux et la dynamique de codage. On distingue des quantificateurs uniformes (l'écart entre chaque valeur quantifiée est constant) et non uniformes (l'écart entre chaque valeur quantifiée est variable) comme l'illustre la figure suivante.

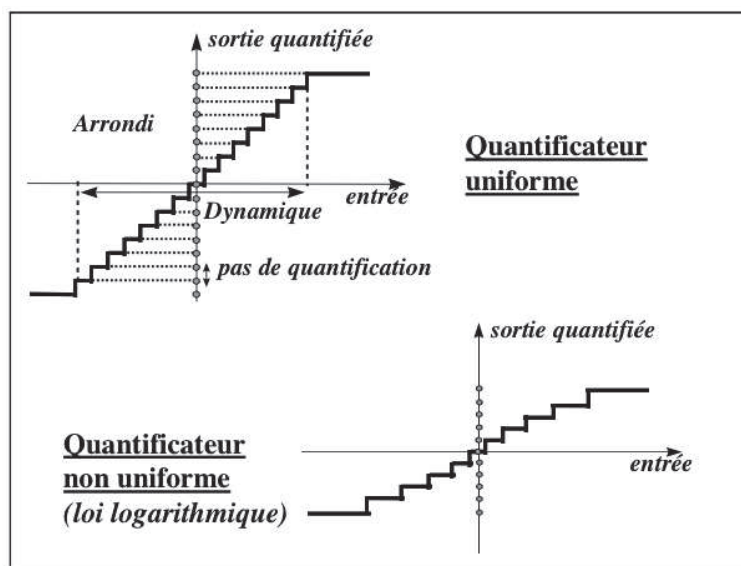


FIGURE 2.6 – Quantificateurs uniforme et non uniforme

On note N le nombre de valeurs quantifiées ou nombre de niveaux du quantificateur. Le codage nécessite alors b bits, où 2^b est la puissance de 2 immédiatement supérieure ou égale à N .

L'effet de la quantification revient, en première approximation sous certaines conditions réalisées en pratique (approximation dite de Sheppard) à ajouter au signal $s(t)$ un signal d'erreur $e(t)$ appelé bruit de quantification, non corrélé avec $s(t)$. Le pas de quantification doit être choisi de façon à minimiser l'erreur en sortie du quantificateur.

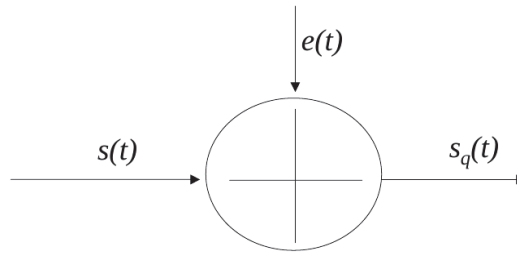


FIGURE 2.7 – Bruit de quantification

Quantification uniforme

Le pas de quantification est alors constant et vaut :

$$q = \frac{\text{pleine échelle du quantificateur}}{N}$$

On peut distinguer deux types de quantificateurs uniformes : type A et type B.

type A : si $\left(n - \frac{1}{2}\right)q \leq s(t) \leq \left(n + \frac{1}{2}\right)q$ alors $s_Q(t) = nq$

type B : $nq \leq s(t) \leq (n+1)q$ alors $s_Q(t) = \left(n + \frac{1}{2}\right)q$

Leurs caractéristiques sont données sur la figure suivante. Dans les deux cas, le domaine de quantification est symétrique par rapport à zéro.

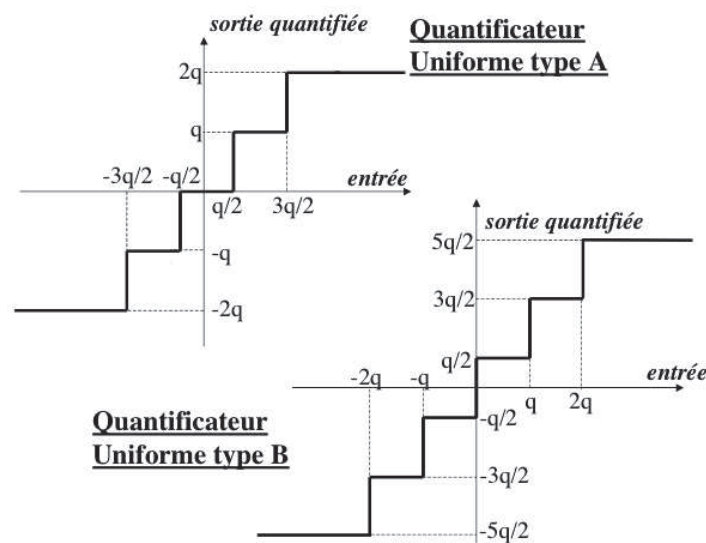


FIGURE 2.8 – Quantificateurs uniformes de type A et B

L'amplitude du signal d'erreur est comprise entre $-q/2$ et $q/2$. Quand les variations du signal sont grandes par rapport au pas de quantification, c'est-à-dire que la quantification est faite avec suffisamment de finesse, et lorsque le signal n'est pas échantillonné, le signal d'erreur peut être supposé à distribution uniforme, de densité de probabilité :

$$\begin{aligned} p_e(x) &\approx \frac{1}{q} && \text{pour } x \in \left[-\frac{q}{2}, \frac{q}{2}\right] \\ p_e(x) &\approx 0 && \text{ailleurs} \end{aligned}$$

Dans ce cas, la puissance du bruit de quantification est donnée par :

$$\sigma_e^2 = \frac{q^2}{12}$$

Rapport signal à bruit de quantification

Considérons la quantification uniforme de type B. La gamme des amplitudes qu'il est possible de coder est soumise à une double limitation : vers les faibles valeurs, elle est limitée par le pas de quantification q et vers les fortes valeurs par $\frac{Nq}{2}$. Toute amplitude qui dépasse cette valeur ne peut être représentée et il y a **écrêtage du signal**. Il s'en suit une dégradation par distorsion harmonique, par exemple si le signal est sinusoïdal.

Pour définir et calculer le rapport signal à bruit de quantification, considérons un signal sinusoïdal non échantillonné et occupant la pleine échelle du quantificateur.

On appelle **puissance crête** P_c d'un codeur la puissance du signal sinusoïdal ayant l'amplitude maximale A_m admissible sans écrêtage :

$$A_m = \frac{Nq}{2} \quad P_c = \frac{1}{2} \left(\frac{Nq}{2} \right)^2$$

Le rapport signal à bruit de quantification est alors le rapport de la puissance crête et de la puissance du bruit de quantification.

$$\frac{P_c}{\sigma_e^2} = \frac{3}{2} N^2$$

soit en décibels et pour $N = 2^b$:

$$10 \log_{10} \left(\frac{P_c}{\sigma_e^2} \right) = 20 \log_{10} N + 1.76 \text{dB} \simeq 6b + 1.76 \text{dB}$$

Le rapport signal à bruit de quantification est proportionnel à N^2 . Doubler N (ajouter un bit) améliore le rapport signal à bruit de 6dB .

Quantification non uniforme : codage non linéaire suivant une loi segmentée

Le rapport signal à bruit de quantification varie fortement avec le niveau du signal dans le cas de la quantification uniforme. La quantification non uniforme vise à maintenir l'**erreur relative**

de quantification constante, quelle que soit l'amplitude du signal alors que la quantification uniforme introduit une **erreur absolue** maximum de $\pm \frac{1}{2}q$.

La solution théorique est de faire varier le pas de quantification q proportionnellement à l'amplitude de l'entrée. Or ceci est incompatible avec la nécessité d'avoir un nombre fini de niveaux de quantifications. En pratique, la quantification non-uniforme utilise une quantification uniforme avec compression-extension : on réalise tout d'abord une compression de la dynamique du signal dans laquelle le signal x est transformé en un signal y , puis une quantification uniforme du signal y et finalement une extension (opération inverse de la compression) du signal quantifié.

La compression du signal x est destinée à amplifier les faibles amplitudes et à minimiser l'effet des fortes amplitudes. En pratique, deux caractéristiques de compression (correspondant à deux approximations de la loi logarithmique) sont normalisées au niveau international par le C.C.I.T.T. : la loi A et la loi μ .

- la loi A :

$$y = \text{sign}(x) \frac{1 + \text{Ln}(A|x|)}{1 + \text{Ln}(A)} \quad \text{pour } \frac{1}{A} < |x| \leq 1$$

$$y = \text{sign}(x) \frac{A|x|}{1 + \text{Ln}(A)} \quad \text{pour } 0 \leq |x| \leq \frac{1}{A}$$

- la loi μ :

$$y = \text{sign}(x) \frac{\text{Ln}(1 + \mu|x|)}{\text{Ln}(1 + \mu)} \quad \text{pour } -1 \leq x \leq 1$$

Les paramètres de A et μ déterminent l'augmentation de la dynamique du codeur. La valeur retenue pour A est 87.6, celle retenue pour μ est 255. Ces lois sont ensuite approchées par des segments de droite, de façon à coder sur un nombre fini de bits le signal ainsi distordu. La loi A est normalisée par le C.C.I.T.T. comme une loi à 13 segments alors que la loi μ est normalisée à 15 segments. La caractéristique de compression de la loi A à 13 segments est donnée sur la figure suivante. La caractéristique fait apparaître 7 segments dans le quadrant positif, on en imagine autant dans le quadrant négatif et, les deux segments entourant l'origine étant colinéaires, on obtient bien 13 segments. Les lois A et μ sont utilisées en téléphonie. Elles constituent deux approximations de la loi logarithmique proposées respectivement en Europe et aux Etats-Unis. Ces lois permettent d'obtenir un rapport signal à bruit de quantification constant à partir d'un seuil ($1/A$ ou $1/\mu$).

L'intérêt de telles lois réside dans la diminution du nombre d'éléments binaires pour coder les échantillons, tout en conservant la dynamique du signal d'entrée (au détriment du RSB aux forts niveaux).

Si on prend l'exemple de la loi A , pour coder l'amplitude du signal distordu par cette loi, il faut :

- 1 bit de signe
- 3 bits indiquant le numéro de segments

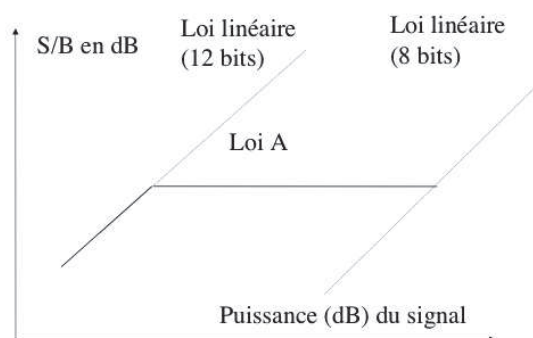


FIGURE 2.9 – Rapport signal à bruit de quantification avec quantification non uniforme

- n autres bits pour coder le niveau où on se trouve dans le segment.

La dynamique du signal téléphonique est d'environ $65dB$, nécessitant 12 éléments binaires dans le cas d'un codage linéaire. Avec la compression de la loi A , le nombre de bits descend à 8, ce qui signifie que $n = 4$ bits suffisent pour un codage correct du niveau des segments.

Approximation de la loi A à l'aide de 13 segments :

si $0 \leq x \leq \frac{1}{64}$	alors	$y = 16x$
$\frac{1}{64} \leq x \leq \frac{1}{32}$	alors	$y = 8x + \frac{1}{8}$
$\frac{1}{32} \leq x \leq \frac{1}{16}$	alors	$y = 4x + \frac{1}{4}$
$\frac{1}{16} \leq x \leq \frac{1}{8}$	alors	$y = 2x + \frac{3}{8}$
$\frac{1}{8} \leq x \leq \frac{1}{4}$	alors	$y = x + \frac{1}{2}$
$\frac{1}{4} \leq x \leq \frac{1}{2}$	alors	$y = \frac{1}{2}x + \frac{5}{8}$
$\frac{1}{8} \leq x \leq \frac{1}{4}$	alors	$y = x + \frac{1}{2}$
$\frac{1}{2} \leq x \leq 1$	alors	$y = \frac{1}{4}x + \frac{3}{4}$

2.1.4 Annexe : Test de Kolmogorov

Le test de Kolmogorov permet de décider (avec une marge d'erreur donnée) si une variable aléatoire X suit une loi de fonction de répartition $F(x)$. Dans ce TP, il est utilisé pour l'étude de la distribution de l'erreur de quantification.

Le déroulement du test est le suivant :

1) Estimation de la fonction de répartition $\hat{F}(x)$ à partir de K observations x_k $k = 1, \dots, K$ de la variable aléatoire X . **Remarque importante** : pour se rapprocher au maximum des conditions

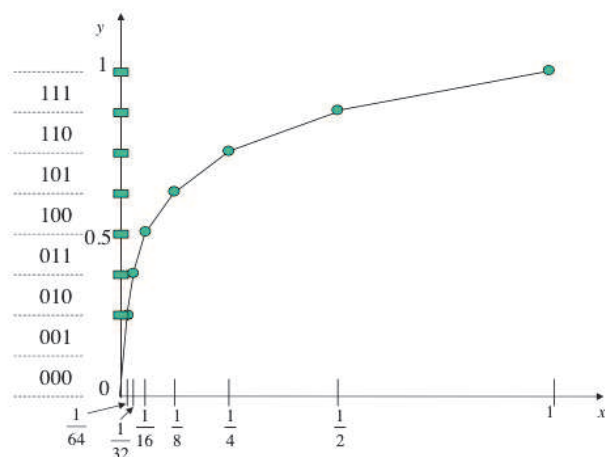


FIGURE 2.10 – Quantification non uniforme : approximation de la loi A

de validité du test de Kolmogorov, le nombre de classes pour l'estimation de la fonction de répartition doit être pris égal au nombre de points de signal.

2) Recherche de l'écart maximum Δ_{\max} entre $\hat{F}(x)$ et $F(x)$.

3) Pour un risque de première espèce α donné (1% ou 5% généralement) vérifier que la valeur donnée dans la table de Kolmogorov est supérieure à Δ_{\max} . Si c'est le cas, on décide que X suit la loi de fonction de répartition $F(x)$ avec un risque d'erreur $\alpha\%$. Rappelons la notion de risque lorsqu'on fait un test d'hypothèse H_0 contre H_1 : 2 risques sont définis :

$\alpha = P[\text{rejeter } H_0 \text{ sachant } H_0 \text{ vraie}]$ risque de non détection

$\beta = P[\text{accepter } H_0 \text{ sachant } H_0 \text{ fausse}]$ risque de fausse alarme

Pour expliquer qualitativement le rôle joué par les risques α et β , imaginons la situation suivante où on effectue la surveillance militaire d'un terrain. Prenons l'hypothèse H_0 : "il y a des ennemis" et l'hypothèse H_1 : "il n'y a pas d'ennemis". Le risque α est le risque de croire qu'il n'y pas d'ennemis et donc, de ne pas tirer alors qu'il y a effectivement des ennemis sur le terrain... Dangereux!... Le risque β est le risque de tirer en croyant qu'il y a des ennemis alors qu'il n'y a personne. Le second risque β est moins important que le premier qui peut avoir des conséquences désastreuses... Théoriquement, le résultat d'un test n'est valable que si l'on précise les valeurs des 2 risques. Le test de Kolmogorov, très simple à mettre en oeuvre, ne prend malheureusement en compte que le risque de première espèce α .

La table suivante donne l'écart maximal théorique, Δ_{kolmo} , entre fonctions de répartition empirique et théorique pour accepter l'hypothèse H_0 avec un risque α de 5 ou 1%, en fonction de Nc (nombre de points de calcul de la fonction de répartition, c'est-à-dire nombre de classes de l'histogramme). Si l'écart maximal mesuré Δ_{\max} est inférieur à Δ_{kolmo} , on accepte l'hypothèse

H_0 avec un risque α . Cette table n'est valable que pour N_c inférieur à 100. Pour N_c plus grand, on sait calculer $P(\max |\hat{F} - F| > \Delta_{kolmo})$ sous H_0 (Vladimir N. Vapnik, "The Nature of Statistical Learning Theory", Springer Ed. p87). Or :

$$\alpha = P[\text{rejeter } H_0 \mid H_0 \text{ vraie}] = P(\max |\hat{F} - F| > \Delta_{kolmo}) \simeq \sum_{k=1}^{k=+\infty} 2(-1)^{k-1} \exp(-2k^2 N_c \Delta_{kolmo}^2)$$

On mesure tout d'abord $\Delta_{\max} = \max |\hat{F} - F|$, écart maximal entre la fonction de répartition théorique et la fonction de répartition estimée. On calcule ensuite :

$$P_{\Delta_{\max}} = P(\max |\hat{F} - F| > \Delta_{\max}) \simeq \sum_{k=1}^{k=+\infty} 2(-1)^{k-1} \exp(-2k^2 N_c \Delta_{\max}^2)$$

Or cette probabilité est une fonction décroissante de Δ_{\max} . Donc si $P_{\Delta_{\max}} > \alpha$, on peut en déduire que $\Delta_{\max} < \Delta_{kolmo}$. Il n'est donc pas nécessaire de calculer Δ_{kolmo} pour prendre la décision. En effet, on accepte l'hypothèse H_0 si :

$$\sum_{k=1}^{k=+\infty} 2(-1)^{k-1} \exp(-2k^2 N_c \Delta_{\max}^2) > \alpha$$

Le calcul de cette probabilité sera effectué sous Matlab, la somme infinie pouvant être approchée par une somme de 1 à K "très grand" (par exemple $K = 1000$).

N_c	$\alpha = 5\%$	$\alpha = 1\%$
5	0.5633	0.6685
10	0.4087	0.4864
15	0.3375	0.4042
20	0.2939	0.3524
25	0.2639	0.3165
30	0.2417	0.2898
40	0.2101	0.2521
50	0.1884	0.2260
60	0.1723	0.2067
70	0.1597	0.1917
80	0.1496	0.1795
90	0.1412	
100	0.1340	

2.2 Travail à réaliser

Lancer `ech_quant` sous Matlab.

2.2.1 Echantillonnage

Echantillonnage et périodisation du contenu spectral.

Générer $N=100$ points d'un signal sinusoïdal de fréquence 5000 Hz échantillonné à 100 000 Hz.

Dans une période du sinus, combien y a-t-il de points ?

Effectuer le "spectre" du signal (cette fonction sera étudiée en détail dans le TP Corrélations et Spectres).

Expliquer ce que l'on obtient, en particulier la présence des deux "pics" et leurs fréquences respectives (utiliser le zoom...).

Changer le nombre de points $N=1000$. Que se passe-t-il ? (utiliser le zoom)

Passer en fréquences normalisées (bouton "changement échelle fréquentielle"). Retrouver les valeurs des fréquences intéressantes.

Effet de la fréquence d'échantillonnage sur les représentations temporelle et spectrale d'un signal : le repliement

Reprendre le signal généré au départ : $N=100$ points, sinus de fréquence 5000 Hz, échantillonné à 100 000 Hz.

Pour différentes fréquences d'échantillonnage F_e , observez le signal en temporel et en spectral :

$F_e = 50\,000$ Hz

$F_e = 20\,000$ Hz : combien de points par période du sinus ?

$F_e =$ Fréquence de Shannon : combien de points par période du sinus ? Que se passe-t-il du point de vue spectral ?

$F_e = 7\,500$ Hz : sur le spectre, d'où proviennent les pics observés ? (Faire un dessin). La fréquence du sinus reconstitué est de quelle valeur ? Vérifier sur la représentation temporelle que le nombre de points par période du sinus correspond bien à cette fréquence.

$F_e = 4\,000$ Hz : mêmes questions.

Pour un signal carré

Générer $N=1000$ points d'un signal carré de fréquence fondamentale de 800 Hz, échantillonné à 100 000 Hz.

Visualiser son spectre. Qu'observez-vous ?

Changer la fréquence d'échantillonnage : $F_e=10\,000$ Hz. Observez le spectre. Que se passe-t-il ? Un tel signal est-il échantillonnable en théorie ?

Pour un signal sonore

Lors de la numérisation d'un signal sonore, il est nécessaire de l'échantillonner. Classiquement, la fréquence d'échantillonnage choisie est de 44100 Hz. Sachant que l'oreille humaine perçoit des fréquences allant de 20 Hz à 20 kHz, ce choix vous semble-t-il judicieux ?

Lancez l'interface `ech_quant_son` sous Matlab. Cette interface permet d'échantillonner un enregistrement sonore. L'enregistrement de base est à 44100 Hz. Essayer plusieurs fréquences d'échantillonnage (10000, 5000, 2000, 1000 Hz) sans filtre anti-repliement et écoutez le résultat. Que constatez vous ?

Le filtre anti-repliement correspond à une filtrage des hautes fréquences en fonction de la fréquence d'échantillonnage afin d'éviter le repliement spectral. Essayer à nouveau les différentes fréquences d'échantillonnage avec le filtre anti-repliement. Que constatez vous (notamment pour de faibles fréquences d'échantillonnage) ?

Comparer les spectres des signaux échantillonnés avec et sans filtre anti-repliement.

Pour une image

Il est également possible d'échantillonner des images. Dans ce cas, la notion d'échantillonnage correspond au nombre de pixels par unité de longueur. On parle alors de fréquence spatiale. Lancer l'interface `ech_quant_img` sous Matlab. Cette interface permet d'échantillonner une image de taille 512×512 pixels. Diminuer l'échantillonnage de l'image. Que constatez-vous ? A votre avis, dans une image, quel type d'information contient les hautes fréquences ? Et les basses fréquences ?

2.2.2 Quantification

Quantification uniforme grossière

Générer $N=1000$ points d'un signal sinusoïdal de fréquence 20 Hz échantillonné à 2000 Hz.

Quantifier ce signal à l'aide d'un quantificateur uniforme à 4 niveaux. Pour voir l'erreur de quantification, utiliser "traitement". Que se passe-t-il en temporel ?... Bien caler la dynamique!!!

Visualiser le spectre du signal quantifié. Quel est l'effet d'une quantification grossière sur un signal sinusoïdal ?

Considérer une dynamique de 40 pour une quantification uniforme à 4 niveaux. Effectuer la quantification uniforme de type A et de type B du signal sinusoïdal précédent. Conclusions ?

Quantification uniforme haute résolution

Vérifier la formule liant SNR et nombre de niveaux de quantification dans le cas haute résolution (nombre de niveaux de quantification 2^b avec $b = 16, 15, 14, \dots$) :

$$\frac{P_c}{\sigma_e^2} = \frac{3}{2}N^2$$

Montrer en particulier que doubler N améliore le rapport signal à bruit de 6dB.

Une des hypothèses couramment utilisée en quantification est de dire que *l'erreur de quantification est uniformément répartie sur $[-q/2, +q/2]$* .

Générer $N=1000$ points d'un signal sinusoïdal de fréquence 2100 Hz échantillonné à 100 000 Hz. Effectuer la quantification uniforme de ce sinus sur 14 bits (attention à la dynamique!). Observer l'erreur de quantification puis sa densité de probabilité. En appuyant sur OK, différentes réalisations du même processus aléatoires sont générées. Que penser de cette hypothèse ?

Pour s'assurer de la validité de cette hypothèse, on effectue un test de Kolmogorov. Jusqu'à quel nombre de bits de quantification peut-on considérer que cette hypothèse est valable ?

2.2.3 Restitution

Générer $N=100$ points d'un signal sinusoïdal de fréquence 2100 Hz, échantillonné à 100 000 Hz. Comparer les méthodes de restitution par bloqueur, interpolateur linéaire, extrapolateur et filtrage tant sur le plan temporel que sur le plan spectral.

Mêmes questions lorsque le signal sinusoïdal est échantillonné à 10 000 Hz.

Mêmes questions pour un signal carré de fréquence fondamentale 2100 Hz, échantillonné à 100 000 Hz.

2.2.4 Signal sonore

Lancer l'interface `ech_quant_son` et sélectionner '*Quantifier*'. Observer le signal en temporel, remarquer le caractère non stationnaire du signal. Observer la densité de probabilité du signal et retrouver la validité apparente de l'hypothèse de loi double exponentielle souvent énoncée pour des signaux sonores.

Essayer une quantification uniforme grossière à 8 niveaux et écoutez le résultat. Que constatez vous ? Observez également l'erreur de quantification.

Comparer la quantification uniforme et non uniforme (logarithmique) de ce signal à nombre de niveaux fixé à $N=8$ (cad 256 en uniforme). Attention à bien choisir la dynamique. En particulier, comparez la distribution de l'erreur de quantification.

2.2.5 Image

Lancer l'interface `ech_quant_img` et observer l'effet de la quantification uniforme sur une image.

2.3 Elements de programmation Matlab

2.3.1 Quelques fonctions Matlab utiles dans le TP

Pour connaître le mode d'appel de ces fonctions, penser à l'aide en ligne de Matlab : `help` 'nom de la fonction' ou `lookfor` 'mot'.

`fft` : transformée de Fourier discrète

`ifft` : transformée de Fourier inverse

`interp1q`: interpolation linéaire

`quantiz`: quantification uniforme du signal.

`lin2mu`: conversion à la loi μ .

`mu2lin`: conversion inverse (de la loi μ à la loi linéaire).

2.3.2 Echantillonnage

Génération d'une sinusoïde : Pour générer le signal obtenu après l'échantillonnage d'une sinusoïde de fréquence 20 KHz échantillonnée à un rythme de 1 échantillon toute les 10 micro-secondes, on peut procéder soit en utilisant les fréquences physiques :

```
N = 1000;
```

```
Te = 10*10^(-6);
```

```
fsin = 20*103;
```

```
temps = (0:N-1)*Te;
```

```
signal = sin(2*pi*fsin*temps);
```

soit en raisonnant directement en fréquences normalisées :

```
N = 1000;
```

```
Fe = 1/(10*10^(-6));
```

```
fsin_norm = 20*103/Fe;
```

```
indice = 0:N-1;
```

```
signal = sin(2*pi*fsin_norm*indice);
```

Visualisation de la densité spectrale de puissance : Le calcul et le tracé de la densité spectrale de puissance seront détaillés au cours du TP Corrélations et Spectres.

L'instruction de base pour réaliser la transformée de Fourier discrète d'un signal \mathbf{x} est `fft(x)`.

Les commandes suivantes permettent de tracer la densité spectrale de \mathbf{x} soit en fréquence normalisée :

```
nfft = 2^nextpow2(N); %calcul de la puissance de 2 immédiatement supérieure à N.
```

```
plot(linspace(0,1,nfft),abs(fft(x,nfft)).^2./nfft).
```

soit en fréquence physique :

```
plot(linspace(0,Fe,nfft),abs(fft(x,nfft)).^2./nfft).
```

Restitution : la commande `y = kron(x, ones(1,Nrestit))`; permet de restituer par bloqueur d'ordre 0 le signal `x` échantillonné à `Te`. Le signal `y` résultant est une version suréchantillonnée de `x` de période d'échantillonnage `Te/Nrestit`.

La commande `interp1q` permet de réaliser l'interpolation linéaire (taper `help interp1q`);

La commande `interp` permet de générer un filtre passe-bas d'interpolation.

2.3.3 Quantification

Quantification uniforme : La fonction `quantiz` permet de réaliser directement la quantification sous Matlab.

Etude statistique de l'erreur de quantification : La fonction `var` donne une estimation de la variance. Pour l'estimation de la densité de probabilité et le test de Kolmogorov se reporter au TP 1.

Quantification non uniforme selon la loi μ : voir les fonctions `lin2mu`, `mu2lin` et `auwrite`.

Chapitre 3

TP Analyse Spectrale - Corrélations et Spectres

Le but de ce TP est d'analyser des estimateurs de la fonction de corrélation et de la densité spectrale de puissance (DSP).

La fonction de corrélation constitue une mesure de ressemblance entre deux signaux (intercorrélation) ou, si elle est appliquée à un seul signal, cette fonction, appelée alors autocorrélation, mesure le lien qui existe entre les différentes valeurs du signal à des instants différents. Cette quantité dépend d'un paramètre de décalage temporel (déphasage entre les deux signaux dans le cas de l'intercorrélation) et possède selon la nature des signaux, plusieurs définitions.

La densité spectrale de puissance (transformée de Fourier de la fonction d'autocorrélation) reflète la contribution qu'apporte chaque fréquence à la puissance moyenne du signal.

3.1 Rappels

3.1.1 Propriétés des fonctions de corrélation

La fonction de corrélation se définit de différentes façons suivant la classe de signaux à laquelle on s'adresse. Nous donnons ci-après les différentes définitions de la fonction d'intercorrélation. Pour la fonction d'autocorrélation, il suffit de faire $y = x$.

Signaux déterministes

— Energie finie

$$C_{xy}(\tau) = \int_{-\infty}^{+\infty} x(t)y^*(t - \tau)dt$$

— Puissance finie

$$C_{xy}(\tau) = \lim_{T \rightarrow +\infty} \frac{1}{T} \int_0^T x(t)y^*(t - \tau)dt$$

Cas particulier des signaux périodiques :

$$C_{xy}(\tau) = \frac{1}{T} \int_0^T x(t)y^*(t-\tau)dt$$

Signaux aléatoires

$$C_{xy}(\tau) = E[x(t)y^*(t-\tau)]$$

Lorsque $y = x$ on parle de fonction d'**autocorrélation** ($C_{xx}(\tau) = C_x(\tau)$).

Propriétés de la fonction d'autocorrélation

Parité : $C_x(-\tau) = C_x(\tau)$

Maximum en zéro : $|C_x(\tau)| \leq C_x(0)$

Puissance moyenne du signal = $C_x(0)$

Propriétés de la fonction d'intercorrélation

Symétrie hermitienne : $C_{xy}^*(-\tau) = C_{xy}(\tau)$

Majoration : $|C_{xy}(\tau)| \leq \frac{1}{2}(C_x(0) + C_y(0))$

3.1.2 Algorithmes de calcul des fonctions de corrélation

Pour estimer la fonction d'autocorrélation ou d'intercorrélation, on utilise deux types d'estimateurs.

Dans une première approche, la fonction d'autocorrélation est estimée comme la valeur moyenne de $x(n)x(n+k)$. Si on ne dispose que de N échantillons du signal $x(n)$ (réel), $C_{xx}(k)$ ne peut être estimée qu'à partir de $N-k$ valeurs :

$$\hat{C}_b(k) = \frac{1}{N} \sum_{n=0}^{N-k-1} x(n)x(n+k) \quad 0 \leq k \leq N-1 \quad (3.1)$$

Lorsque k tend vers $N-1$ peu de termes interviennent dans le calcul de la moyenne alors que le terme de normalisation reste égal à $\frac{1}{N}$. Cela a pour conséquences d'introduire un **biais** dans l'estimation : la corrélation est pondérée par une fenêtre **triangulaire**. Ce premier estimateur $\hat{C}_b(k)$ est donc biaisé.

Pour éliminer ce biais, un second estimateur peut être défini de la façon suivante :

$$\hat{C}_{nb}(k) = \frac{1}{N-k} \sum_{n=0}^{N-k-1} x(n)x(n+k) \quad 0 \leq k \leq N-1 \quad (3.2)$$

Le choix entre ces deux estimateurs se porte normalement sur l'estimateur non biaisé mais, pour les valeurs k proches de N , la variance de cet estimateur augmente considérablement ce

qui n'était pas le cas du précédent estimateur. D'autre part, il peut être intéressant de choisir l'estimateur biaisé ce qui est équivalent à prendre la fonction de corrélation multipliée par une fenêtre temporelle triangulaire.

Remarque : Le calcul de ces deux estimateurs requiert de l'ordre de $\frac{N^2}{2}$ multiplications et additions. Pour réduire le coût calculatoire, on peut utiliser un algorithme à base de transformée de Fourier rapide (FFT). En effet, les deux estimateurs ci-dessus représentent à un facteur multiplicatif près la même opération de convolution discrète $x(k) * x(-k)$. Pour calculer cette convolution, on peut utiliser le fait que

$$TF(x(k) * x(-k)) = X(f)X^*(f) = |X(f)|^2$$

avec $X(f)$ la transformée de Fourier de $x(n)$. Ainsi, les sommes temporelles coûteuses en coût calculatoire contenues dans (3.1) et (3.2) peuvent être remplacées par $FFT^{-1} \left[|FFT(x(n))|^2 \right]$, ce qui rend le calcul plus rapide.

3.2 Exemples d'utilisation des fonctions de corrélation

Nous donnons ci-après deux exemples d'utilisation des fonctions de corrélation.

3.2.1 Détection d'un signal périodique noyé dans un bruit

Par détection, on entend détection de présence. Il ne s'agit pas de retrouver la forme du signal périodique mais de détecter sa présence, de savoir si ce signal existe ou non.

Soit $x(n)$ le signal périodique de période inconnue noyé dans un bruit centré $b(n)$:

$$y(n) = x(n) + b(n)$$

La fonction d'autocorrélation du signal $y(n)$ s'écrit :

$$C_{yy}(k) = C_{xx}(k) + C_{xb}(k) + C_{bx}(k) + C_{bb}(k)$$

Si le bruit $b(n)$ est indépendant du signal $x(n)$, les intercorrélations $C_{xb}(k)$ et $C_{bx}(k)$ sont nulles pour tout k . Si de plus la densité spectrale du bruit est absolument continue, on a :

$$\lim_{k \rightarrow +\infty} C_{bb}(k) = 0$$

D'où :

$$C_{yy}(k) \rightarrow C_{xx}(k) \text{ lorsque } k \rightarrow +\infty$$

En calculant la fonction d'autocorrélation de $y(n)$, on va voir apparaître celle du signal périodique $x(n)$ aux grandes valeurs de k .

3.2.2 Identification d'un filtre

Considérons un filtre ou d'une façon plus générale un système linéaire invariant dans le temps, de réponse temporelle $h(t)$ et de réponse fréquentielle $H(f)$ (fonction de transfert). La relation temporelle qui relie la sortie $y(t)$ à l'entrée $x(t)$ est la suivante :

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(u)h(t-u)du$$

On obtient la même relation sur les fonctions de corrélation :

$$C_{yx}(\tau) = C_{xx}(\tau) * h(\tau)$$

où $C_{yx}(\tau)$ est l'intercorrélation entre la sortie et l'entrée du filtre.

Si le signal $x(t)$ est un bruit blanc, c'est-à-dire $C_{xx}(\tau) = \delta(\tau)$, on a :

$$C_{yx}(\tau) = \delta(\tau) * h(\tau) = h(\tau)$$

L'intercorrélation entre la sortie et l'entrée du filtre correspond à la réponse temporelle du filtre. La transformée de Fourier de l'intercorrélation permet d'obtenir la réponse fréquentielle. Une autre possibilité consiste à estimer le spectre de la sortie du filtre :

$$Y(f) = X(f)H(f)$$

La fonction de transfert $H(f)$ permet une interprétation plus visuelle de la nature du filtre (passe-haut, passe-bas, passe-bande).

3.3 Analyse spectrale

3.3.1 Différents estimateurs

Il existe 2 grandes familles d'estimateurs de la densité spectrale de puissance (DSP) :

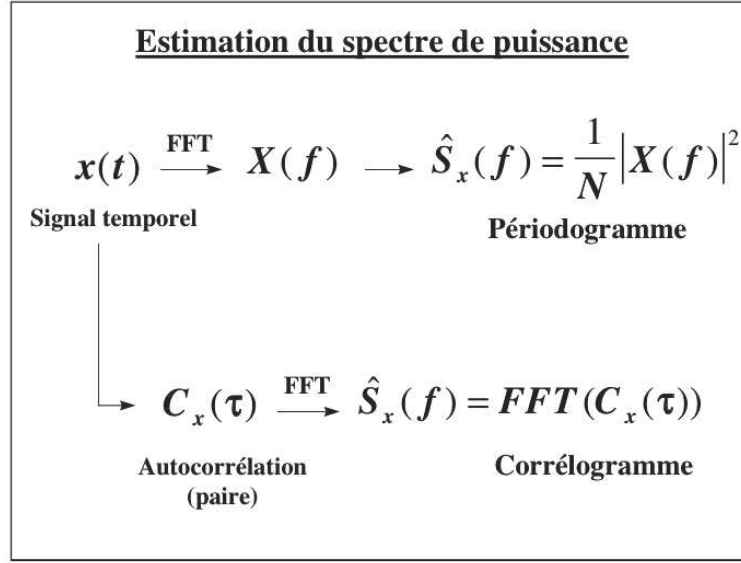
- les méthodes du PERIODOGRAMME
- les méthodes du CORRELOGRAMME

Ces deux catégories de méthodes mettent en oeuvre la transformée de Fourier discrète définie par :

$$X(f) = \sum_{n=0}^{N-1} x(n)e^{-i2\pi fn}$$

L'algorithme de transformée de Fourier discrète rapide, FFT, requiert un nombre de points fréquentiels N_f égal à une puissance de 2 afin d'obtenir la meilleure efficacité en termes de temps de calcul. Le spectre $S_x(f)$ est calculé aux fréquences normalisées :

$$f = \frac{k}{N_f} \quad k = 0, \dots, N_f - 1$$



3.3.2 Intérêt du zero-padding

A titre d'exemple, la transformée de Fourier discrète (TFD) d'un sinus

$$x(n) = A \cos(2\pi f_0 n + \phi) \quad n = 0, \dots, N-1$$

donne :

$$\begin{aligned}
 X(f) = & \frac{A}{2} \left(e^{-i(\pi(f-f_0)(N-1)+\phi)} \frac{\sin(\pi(f-f_0)N)}{\sin(\pi(f-f_0))} \right. \\
 & \left. + e^{-i(\pi(f+f_0)(N-1)+\phi)} \frac{\sin(\pi(f+f_0)N)}{\sin(\pi(f+f_0))} \right)
 \end{aligned}$$

Ceci correspond à 2 noyaux de Dirichlet (ressemblant à 2 sinus cardinaux), l'un centré en f_0 , l'autre en $-f_0$. Chaque motif correspondant à un noyau de Dirichlet se présente sous forme d'un lobe principal et de lobes secondaires, avec une pseudo période de $\frac{1}{N}$, et le résultat du calcul est représenté sur seulement N points. Ceci donne 1 point par pseudo-période et c'est trop peu pour que, visuellement, on voit clairement apparaître le noyau de Dirichlet centré sur la fréquence d'intérêt. Afin d'améliorer la visualisation du résultat de la TFD, il faut prendre un nombre de points en fréquence N_f grand devant le nombre d'échantillons du signal et faire ainsi du "zero-padding".

3.3.3 Corrélations théoriques de signaux particuliers

— Sinusoïde à phase aléatoire

$$x(n) = A \cos(2\pi f n + \phi) \rightarrow C_x(k) = \frac{A^2}{2} \cos(2\pi f k) \quad (3.3)$$

— Bruit blanc

$$C_x(k) = \sigma^2 \delta(k) \quad (3.4)$$

3.3.4 Périodogramme d'une sinusoïde bruitée et estimation du SNR

On considère N échantillons d'une sinusoïde de fréquence f_0 et d'amplitude A perturbée par un bruit blanc de puissance σ^2 . Le périodogramme de ce signal est :

$$S_x(f) = \frac{1}{N} |X(f)|^2 \approx \sigma^2 + \frac{A^2}{4} N (\text{sinc}^2(\pi N(f - f_0)) + \text{sinc}^2(\pi N(f + f_0))) \quad (3.5)$$

avec $\text{sinc}(x) = \frac{\sin(x)}{x}$.

L'amplitude du pic à la fréquence $f = f_0$ correspond donc à $\frac{A^2}{4} N + \sigma^2$ ce qui permet d'estimer le rapport signal-à-bruit (SNR) :

$$SNR = 10 \log \left(\frac{\text{Puissance de la sinusoïde} = \frac{A^2}{2}}{\text{Puissance du bruit} = \sigma^2} \right) \quad (3.6)$$

En effet, l'observation de la DSP obtenue permet de mesurer le niveau de bruit moyen σ^2 ainsi que l'amplitude du sinus A , comme l'illustre la figure suivante.

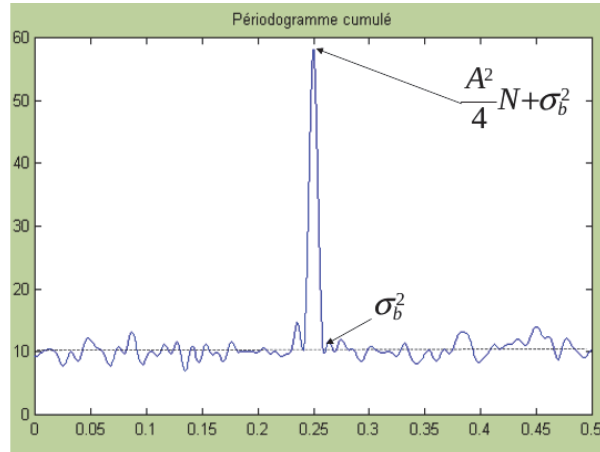


FIGURE 3.1 – Exemple de DSP estimée d'une sinusoïde bruitée en échelle linéaire

3.4 Travail à effectuer

Lancer `analyse_spectrale` sous Matlab.

Le but de ce TP est d'effectuer de l'analyse spectrale par Transformée de Fourier (TF) et de comprendre les intérêts, inconvénients et nature des paramètres des estimateurs existants.

Charger successivement les fichiers *Eng8_e5.txt* et *Eng8_e11.txt* (dans le logiciel analyse_spectrale, faire "choix du signal", "fichier") : ces signaux représentent des essais de fatigue de trains épicycloïdaux. La roue de petite vitesse est menante. Elle possède 56 dents et a une fréquence de rotation de 12,43 Hz. La roue de grande vitesse possède 15 dents et une fréquence de rotation de 46,4 Hz. On dispose de mesures issues de capteurs en position horizontale, la fréquence d'échantillonnage est de 6365 Hz. Le premier fichier correspond à des mesures effectuées le 5ème jour et le deuxième fichier à des mesures effectuées le 11ème jour. Le 5ème jour, l'état des dentures est sain tandis que le 11ème jour, il est dégradé : la dent 51 est usée à 100%, les dents 24, 26, 27, 31 et 55 sont usées à 75% et la dent 56 est usée à 15%. La question posée est de savoir si l'analyse spectrale ou le calcul de corrélation nous permettrait de mettre en évidence l'usure des dents.

C'est pourquoi, dans ce TP, nous allons étudier différents estimateurs de la fonction d'autocorrélation et de la Densité Spectrale de Puissance (DSP). Mais avant de travailler sur des signaux réels, nous allons valider les différents estimateurs sur des signaux tests.

Remarque : toutes les fréquences sont données en fréquences normalisées.

Autre remarque : il reste encore des "bugs" dans ce logiciel. En cas de comportement bizarre (refus d'afficher un signal qui devrait devoir s'afficher), utiliser le bouton "reset".

3.4.1 Autocorrélations

Autocorrélation d'un sinus

Générer 50 échantillons d'une sinusoïde de fréquence 0.134 (phase uniformément répartie sur $[0, 2\pi]$). Observer les estimations de son autocorrélation en utilisant le bouton "ok" pour générer à chaque fois une nouvelle réalisation de signal.

Pour l'estimateur biaisé, quelle est l'allure du biais ?

Pour l'estimateur non biaisé, dans quelle partie de l'autocorrélation la variance est-elle la plus importante ? Expliquer.

Retrouver les caractéristiques du signal (puissance et fréquence).

Rappel : l'expression théorique de la corrélation est donnée par (3.3).

Autocorrélation d'un bruit blanc

Générer un bruit blanc ($N = 100$) de variance $\sigma^2 = 4$. Utiliser le bouton "ok" pour voir des réalisations différentes de ce processus aléatoire. Observer les estimations de son autocorrélation, utiliser le bouton "ok" pour visualiser différentes estimations correspondant à différentes réalisations du signal.

Des 2 estimations (biaisée ou non biaisée), laquelle paraît la plus satisfaisante ? Expliquer.

Retrouver les caractéristiques du bruit (remarquer que, pour plus de facilité, la fenêtre "Informations et mesures" fournit la valeur et l'indice du maximum de la fonction).

Rappel : l'expression théorique de la corrélation est donnée par (3.4).

Autocorrélation d'un sinus bruité

Générer 500 échantillons d'une sinusoïde bruitée (bruit blanc additif) de fréquence 0.0134 ($SNR = -7\text{ dB}$). Pour cela, sachant que les sinusoïdes générées sont d'amplitude $A = \sqrt{2}$ (i.e. de puissance unité), régler le niveau de bruit nécessaire en utilisant la définition du SNR donnée par (3.6). Observer le signal et l'estimation biaisée de son autocorrélation (ne pas oublier d'utiliser le bouton "ok" pour générer plusieurs réalisations du processus). Montrer que l'estimation de l'autocorrélation permet de retrouver les caractéristiques du signal : présence d'un signal périodique, puissances du sinus et du bruit.

3.4.2 Estimation spectrale

Périodogramme

Générer une sinusoïde ($N = 128$, $f = 0.134$). Observer le périodogramme en utilisant une taille de FFT = 128 (en échelle linéaire et logarithmique).

Remarque : *On rappelle que pour le périodogramme, le seul paramètre à fixer est la taille de la FFT.*

Recommencer en faisant du zero-padding avec une taille de FFT de 4096. Expliquer les différences observées : l'expression théorique de la DSP est donnée par (3.5) et des rappels sur le zero-padding sont fournis au paragraphe 3.3.2). Retrouver les caractéristiques du signal : fréquence et amplitude du sinus.

Charger le fichier *quiusje0.txt* et cliquer sur le bouton "signal" pour observer le signal en temporel. En faire l'analyse spectrale en utilisant le périodogramme : retrouver les caractéristiques du signal. Ne pas oublier d'observer l'estimateur spectral obtenu en échelle log ! Que représente la DSP à la fréquence $f = 0$?

Périodogramme modifié

Pour une sinusoïde ($N = 100$, $f = 0.25$), analyser et comparer les effets des différentes fenêtres en utilisant le périodogramme modifié. Classer-les en fonction de leur pouvoir à réduire l'amplitude des lobes secondaires (utiliser la représentation en échelle logarithmique et le bouton *hold*).

Remarque : *Pour le périodogramme modifié, les paramètres à choisir sont le type de fenêtre d'apodisation et la taille de la FFT.*

Analyse de signaux sinusoïdaux : intérêt du périodogramme modifié

Charger le fichier *quiusisje1.txt* et cliquer sur le bouton "signal" pour observer le signal en temporel. En utilisant le périodogramme modifié et les différentes fenêtres d'apodisation, donner les caractéristiques de ce signal.

Refaire la même analyse sur le fichier *quiusisje2.txt*.

Corrélogramme

Générer une sinusoïde ($N = 100$, $f = 0.134$). Observer le corrélogramme avec l'estimateur de la corrélation non biaisée en utilisant une taille de FFT = 4096. Comparer avec le corrélogramme utilisant la corrélation biaisée. Qu'en concluez-vous ? Comparer corrélogramme biaisé et périodogramme (utiliser le bouton "hold").

Remarque : *Pour le corrélogramme, les seuls paramètres à choisir sont la nature de l'estimation de la corrélation (biaisée ou non biaisée) et la taille de la FFT.*

Blackman-Tukey

Pour le même signal, utiliser le corrélogramme de Blackman-Tukey.

Remarque : *Pour la méthode de Blackman-Tukey, il est préconisé de l'utiliser sur la corrélation biaisée afin de garantir une DSP positive. Toutefois, il est intéressant de voir ce qu'on obtient en utilisant l'estimation de la corrélation non biaisée. C'est pourquoi les paramètres à choisir pour cette méthode sont la nature de l'estimation de la corrélation (biaisée ou non biaisée), le type de fenêtre d'apodisation, la taille de la fenêtre d'apodisation (**attention, appliquée sur la corrélation qui est de taille double par rapport au signal**) et la taille de la FFT.*

Faire varier tous les paramètres afin d'en observer leur effet (en particulier sur la positivité de la DSP estimée et sur les lobes secondaires des fenêtres spectrales équivalentes) : choix du type de corrélation, choix de la fenêtre, choix de la taille de la fenêtre.

Générer 1000 échantillons d'une sinusoïde bruitée (bruit blanc additif) de fréquence 0.134 ($SNR = -7\text{ dB}$). Comparer l'analyse spectrale obtenue avec le périodogramme modifié et avec Blackman-Tukey, utilisé par exemple en prenant la corrélation biaisée et la fenêtre de Blackman (permettant d'avoir une DSP positive). Que remarquez vous sur la variance de l'estimation spectrale ?

Périodogramme de Bartlett

Générer un bruit blanc ($N = 4000$) de variance $\sigma^2 = 5$. Observer son périodogramme en utilisant une taille de FFT = 8192. Comparer le résultat obtenu à l'expression théorique de la

DSP d'un bruit blanc. En remarquant que la fenêtre "informations et mesures" fournit la valeur moyenne de la DSP ainsi que sa variance, expliquer la différence entre la théorie et l'estimation obtenue de cette DSP. Ne pas hésiter à utiliser le bouton "ok" pour générer plusieurs réalisations du processus aléatoire.

Pour ce même type de signal, utiliser le périodogramme de Bartlett en choisissant des tailles de fenêtre différentes : 1000 et 100 par exemple. Qu'observe-t-on ?

Remarque : *Pour le périodogramme de Bartlett, les paramètres à choisir sont la taille de la fenêtre d'analyse et la taille de la FFT.*

L'utilisation de la méthode de Bartlett se révèle intéressante sur le bruit blanc mais elle s'accompagne d'un inconvénient. Afin de le mettre en évidence, recommencer la même expérience sur un signal constitué d'un sinus bruité ($N = 4000$, $f = 0.2$) en conservant la même variance de bruit. Faire varier la taille de la fenêtre d'analyse de 4000, à 1000 puis à 100. Pour ce type de signal, le logiciel n'affiche pas la valeur moyenne et la variance du spectre mais la variance peut être observée en se plaçant en échelle logarithmique. En déduire les avantages et inconvénients de la méthode de Bartlett.

Périodogramme de Welch

Pour améliorer la méthode de Bartlett, on peut utiliser le périodogramme de Welch qui permet d'utiliser une fenêtre d'analyse et un recouvrement entre les tranches.

Remarque : *Pour le périodogramme de Welch, les paramètres à choisir sont la taille et le type de fenêtre d'analyse, le recouvrement (indiqué en % ici) des tranches et la taille de la FFT.*

Générer un bruit blanc de $N = 50000$ échantillons (beaucoup d'échantillons!), de variance $\sigma^2 = 5$. Choisir une fenêtre d'analyse rectangulaire de taille 1000 et une taille de la FFT de 8192.

Pour un recouvrement de 0%, afficher plusieurs réalisations successives du périodogramme de Welch en notant la variance du spectre (ou en faire une moyenne à l'oeil!).

Refaire de même pour un recouvrement de 30%, 50% et 80%. Quel est l'intérêt du recouvrement ? Son inconvénient ?

Pour un recouvrement fixé à 30% par exemple, faire varier la nature de la fenêtre d'analyse (pas sa taille). Qu'observe-t-on ? Que gagne-t-on à utiliser des fenêtres autre que la rectangulaire ? Ne pas oublier leur inconvénient...

Résumer l'influence (*avantages et inconvénients*) des divers paramètres de la méthode de Welch (la plus utilisée) :

- taille de la FFT :
- type de fenêtre d'analyse :

- taille de la fenêtre d'analyse :
- pourcentage de recouvrement :

3.4.3 Analyse spectrale de signaux réels

Après avoir étudié les estimateurs de corrélation et de DSP, analyser les signaux d'engrenage cités au début du TP. Donner une description (sommaire) de ces signaux. A votre avis, pourrait-on distinguer ces deux signaux et donc signaler l'usure des engrenages à partir de l'analyse spectrale ?

3.5 Programmation Matlab (pour ceux qui le souhaitent)

3.5.1 Quelques fonctions Matlab utiles dans le TP

`randn` : génération d'un bruit blanc normal centré et de variance unité
`fft` : transformée de Fourier discrète
`fftshift` : recentre le spectre
`ifft` : transformée de Fourier inverse
`xcorr` : autocorrélation
`specgram` : spectrogramme (utile pour le calcul du périodogramme cumulé)
`psd` : densité spectrale de puissance Pour connaître le mode d'appel de ces fonctions, penser à l'aide en ligne de MATLAB : `help 'nom de la fonction'` ou `lookfor 'mot'`

3.5.2 Autocorrélations

Elle peut être calculée de la façon suivante :

```
for k=0:N-1,
autocorb(k+1) = fact*signal(1:N-k)*signal(k+1:N)';
end
```

avec `fact = 1/N` pour l'autocorrélation biaisée ou `1/(N-k)` pour l'autocorrélation non biaisée. Il existe sous Matlab la fonction `xcorr`, avec les options `'biased'` et `'unbiased'`. Pour la génération des différents signaux (sinus, carré, bruit blanc gaussien) voir les indications Matlab du TP1.

3.5.3 Estimations spectrales

Transformée de Fourier Discrète

L'instruction de base pour réaliser la transformée de Fourier discrète d'un signal `x` est `fft(x)`. L'algorithme de transformée de Fourier rapide est utilisé par Matlab si et seulement si la longueur

du vecteur \mathbf{x} , N_{ech} , est une puissance de 2. Si ce n'est pas le cas, on peut utiliser la technique du zero-padding pour s'y ramener :

```
nfft = 2^nextpow2(Nech); %calcul de la puissance de 2 immédiatement supérieure à Nech
fft(x,nfft);
```

Exemple : Génération de 512 points d'un sinus de fréquence normalisée $f_0 = 0.2$

```
x=sin(2*pi*f0*(0:511));
```

Calcul de sa FFT et tracé du module (fonction `abs`) de sa FFT :

```
fft_de_x = fft(x,nfft);
```

Si l'axe des \mathbf{x} , n'est pas spécifié dans la commande `plot` :

```
plot(abs(fft_de_x))
```

cet axe porte alors par défaut les indices des éléments du vecteur tracé (ici $0, 1, 2, \dots, n_{\text{nfft}}-1$). Pour interpréter correctement le tracé de la Transformée de Fourier Discrète, il est nécessaire de graduer l'axe des \mathbf{x} , par exemple en fréquences normalisées par rapport à la fréquence d'échantillonnage :

```
axe_des_x = linspace(0,1,nfft)
plot(axe_des_x,abs(fft_de_x))
```

L'affichage du spectre peut être également recentré autour de zéro à l'aide de la commande `fftshift`, l'axe des \mathbf{x} doit alors être modifié :

```
plot(linspace(-0.5,0.5,nfft),fftshift(abs(fft_de_x)))
```

Périodogramme et périodogramme de Welch

La densité spectrale de puissance (DSP) peut être estimée à l'aide du **périodogramme** :

```
den_puis = abs(fft(x,nfft)).^2./Nech;
```

On peut également utiliser le **périodogramme de Welch**. Cela consiste à :

- couper le signal en tranches de N_t points (**N_t doit être évidemment plus faible que la longueur totale du signal**) avec recouvrement possible des tranches.
- faire une FFT de chacune des tranches (avec du zero-padding et une fenêtre d'apodisation) et prendre le module au carré de la FFT,
- faire la moyenne de toutes les FFT en module au carré

Ces opérations peuvent être réalisées avec la fonction `pwelch` (voir le help sous Matlab).

Corrélogramme

Il suffit d'utiliser successivement `xcorr` et `fft`.