



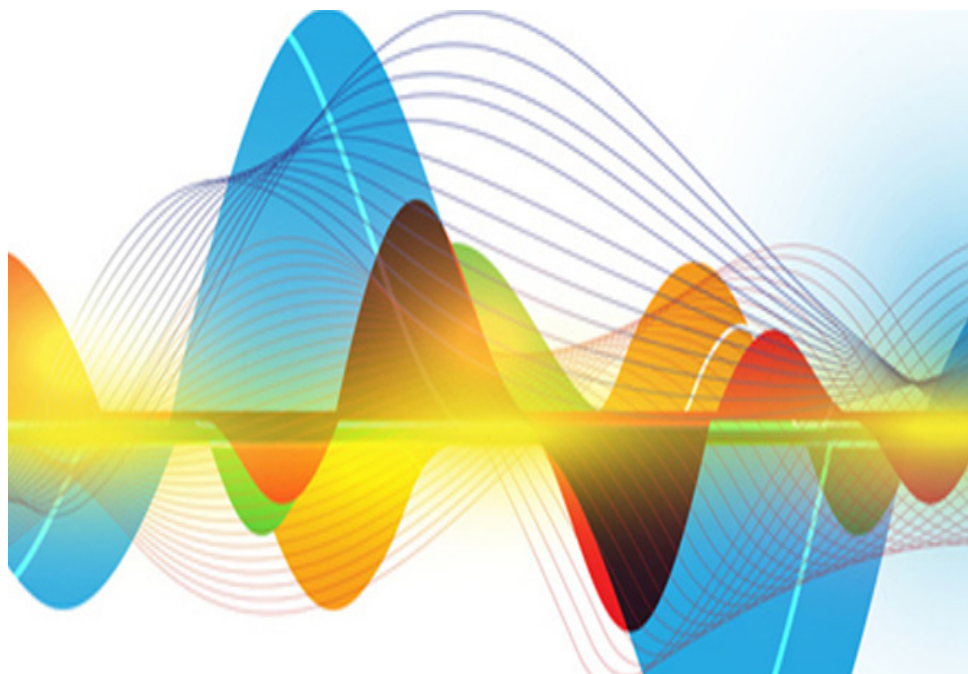
Département Sciences du Numérique

Première année

Traitement Numérique du Signal

Cours, Exercices, Travaux Pratiques et Projets

Auteur : N. Thomas



5 novembre 2020

Table des matières

1	Introduction	5
1.1	Numérisation du signal : échantillonnage	5
1.1.1	Principe, impact	5
1.1.2	Restitution par filtrage	6
1.1.3	Remarques	7
1.2	Numérisation du signal : Quantification	8
1.2.1	Quantification : principe, impact	8
1.2.2	Remarque	9
1.3	Numérisation du signal : exemples sur une image	10
1.4	Outils de traitement du signal à numériser	10
1.5	Notion de temps de traitement et de traitement temps réel	11
1.6	Exercices avec éléments de correction	12
1.6.1	Exercice 1 : Effet de l'échantillonnage	12
1.6.2	Exercice 2 : Echantillonnage d'un signal passe-bande	13
1.6.3	Exercice 3 : Echantillonneur moyennneur	14
1.6.4	Exercice 4 : Echantillonneur bloqueur	15
1.6.5	Exercice 5 : Signal à spectre non borné - Recherche de la F_e	15
1.6.6	Exercice 6 : Quantification d'un sinusoïde	16
2	Transformée de Fourier Discrète (TFD)	17
2.1	De la TF à la TFD	17
2.1.1	Echantillonnage	17
2.1.2	Limitation de la durée du signal à N points	18
2.1.3	Calcul de N points du spectre	20
2.1.4	Expressions de la TFD et de la TFD inverse	22
2.2	Propriétés de la TFD	22
2.2.1	Linearité	22
2.2.2	Translation => rotation de phase	22
2.2.3	Symétrie Hermitienne	22
2.2.4	Convolution circulaire	23
2.2.5	Egalité de Parseval	25
2.2.6	Algorithme de calcul rapide	25
2.3	Algorithme de calcul rapide (FFT) de Cooley Tuckey	25
2.3.1	Principe	25
2.3.2	Exemple pour $N = 2^3 = 8$	26
2.4	Exercices	27

2.4.1	Exercice 1 : Etude de la TFD d'un signal à spectre continu : effet de la limitation de la durée du signal	27
2.4.2	Exercice 2 : Etude de la TFD d'un signal à spectre continu : effet de l'échantillonnage du signal	29
2.4.3	Exercice 3 : Etude de la TFD d'un signal à spectre continu : échantillonnage et limitation de la durée du signal	30
2.4.4	Exercice 4 : Etude de la TFD d'un signal à spectre discontinu : calcul d'un nombre fini de points du spectre	30
3	Fonctions d'inter et d'auto correlations	33
3.1	Calcul dans le domaine temporel	33
3.1.1	Signaux déterministes	33
3.1.2	Signaux aléatoires	33
3.2	Calcul dans le domaine fréquentiel	35
3.3	Quelques propriétés	36
3.3.1	Symétrie Hermitienne	36
3.3.2	Bornes	36
4	Densité Spectrale de Puissance (DSP)	37
4.1	Estimateurs "de base" : corrélogramme, périodogramme	37
4.1.1	Définitions	37
4.1.2	Problèmes posés par ces estimateurs "de base"	38
4.2	Périodogramme avec fenêtrage	38
4.3	Périodogramme cumulé (ou de Bartlett)	38
4.4	Périodogramme de Welch	39
4.5	Exemple sur une ligne d'image SAR	39
5	La transformée en z	40
5.1	Définition	40
5.2	Convergence	40
5.3	Propriétés	41
5.3.1	Linéarité	41
5.3.2	Décalage temporel	41
5.3.3	Changement d'échelle	41
5.3.4	Dérivabilité	41
5.3.5	Produit de convolution	41
5.4	Transformée en z inverse	41
5.4.1	Définition	41
5.4.2	Preuve	42
5.5	Exercices	42
5.5.1	Exercice 1	42
5.5.2	Exercice 2	44
6	Filtrage numérique	45
6.1	Un outil d'étude : la transformée en z	45
6.2	Définitions	45
6.2.1	Linéarité	45

6.2.2	Invariance temporelle	45
6.2.3	Réponse impulsionnelle et fonction de transfert	45
6.2.4	Filtres numériques rationnels	46
6.2.5	Réponse en fréquence et temps de propagation de groupe (TPG)	46
6.3	Réalisabilité d'un filtre numérique	46
6.4	Classification des filtres numériques rationnels : RIF, RII	47
6.4.1	Filtres à réponse impulsionnelle infinie (RII)	47
6.4.2	Filtres à réponse impulsionnelle finie (RIF)	47
6.5	Stabilité des filtres numériques rationnels récurrents (RII)	48
6.5.1	Condition de stabilité	48
6.5.2	Remarques	49
6.6	Synthèse des filtres numériques rationnels	49
6.6.1	Introduction	49
6.6.2	Synthèse des filtres à réponse impulsionnelle finie	51
6.6.3	Synthèse des filtres à réponse impulsionnelle infinie	54
6.7	Implantation des filtres numériques rationnels	57
6.7.1	Structure directe	57
6.7.2	Structure canonique	58
6.7.3	Structures décomposées	58
6.7.4	Structure non récurrente	59
6.8	Exercices	60
6.8.1	Exercice 1	60
6.8.2	Exercice 2	60
6.8.3	Exercice 3 : synthèse d'un filtre passe-bas de type RIF	61
6.8.4	Exercice 4 : étude de la cellule du second ordre	65
6.8.5	Exercice 5 : synthèse RII guidée	68
7	TPs (séances 1 et 2)	71
7.1	Introduction	71
7.2	Séance 1 : Génération et étude d'un signal numérique : le cosinus	71
7.2.1	Représentation temporelle	71
7.2.2	Représentation fréquentielle	71
7.3	Séance 2 : Réalisation d'un filtrage de type passe-bas	72
7.3.1	Génération du signal à filtrer	72
7.3.2	Synthèse du filtre passe-bas	72
7.3.3	Réalisation du filtrage	73
8	Projet (séances 3 à 7)	
	Réalisation d'un modem de fréquence selon la recommandation V21 de l'Union Internationale des Télécommunications (UIT)	74
8.1	Introduction	74
8.2	Information binaire à transmettre	75
8.3	Modem de fréquence - Démodulation par filtrage	76
8.3.1	Construction du signal modulé en fréquence	76
8.3.2	Canal de transmission à bruit additif, blanc et Gaussien	78
8.3.3	Démodulation par filtrage	78
8.3.4	Application de la recommandation V21	80

8.4	Modem de fréquence V21 - Démodulateur FSK	80
8.4.1	Démodulateur FSK - Contexte de synchronisation idéale	80
8.4.2	Démodulateur FSK avec gestion d'une erreur de synchronisation de phase porteuse	81
8.5	Evaluation	82
8.5.1	Le rapport	82
8.5.2	Les codes	82
9	Références	83

Chapitre 1

Introduction

Le traitement numérique du signal désigne l'ensemble des opérations effectuées sur un signal à traiter, représenté par un nombre fini de valeurs, pour fournir un autre nombre fini de valeurs représentant le signal traité. Travailler avec des signaux et des traitements numériques présente un certain nombre d'avantages. Les principaux étant :

- Une plus grande robustesse vis à vis du bruit (dans le cadre des transmissions).
- Une meilleure stabilité et reproductibilité des équipements. Il est, en effet, possible, en numérique, de construire des systèmes identiques, comme il est possible d'anticiper les dérives temporelles liées aux conditions extérieures (température, pression...). Les marges à prendre en compte au moment de la conception des équipements sont donc réduites.
- La possibilité de définir des fonctions nouvelles, notamment des fonctions évoluant dans le temps tel que, par exemple, le filtrage adaptatif, mais également des fonctions de compression, de codage correcteur d'erreurs...

Le nombre fini de valeurs représentant le signal numérique à traiter peut être issu d'un processus numérique (on dit aussi discret) ou provenir d'un signal analogique (représentant une grandeur physique qui évolue dans le temps) qui a été numérisé (on dit aussi discrétisé). Pour numériser un signal analogique (défini à tout instant par des valeurs réelles) deux opérations sont nécessaires : une opération d'échantillonnage (discrétisation dans le domaine temporel) et une opération de quantification (discrétisation dans le domaine des amplitudes).

Nous allons considérer, dans la suite, des signaux déterministes et regarder plus en détail les opérations d'échantillonnage et de quantification. Les résultats obtenus restent valables pour des signaux aléatoires. Seul le formalisme utilisé pour les démonstrations changerait en faisant appel aux équations normales.

1.1 Numérisation du signal : échantillonnage

1.1.1 Principe, impact

Nous considérons ici un échantillonnage idéal périodique et un signal à échantillonner, $x(t)$, déterministe. Si nous notons T_e la période d'échantillonnage, le signal $x(t)$ échantillonné de manière périodique à T_e est constitué d'une succession d'éléments prélevés tous les T_e : $\{x(kT_e)\}$ avec $k \in \mathbb{Z}$. La figure 1.1 présente un exemple d'échantillonnage d'une fonction sinusoïdale.

Afin d'étudier l'effet de cet échantillonnage temporel, nous allons associer à $\{x(kT_e)\}$, $k \in \mathbb{Z}$, le modèle à temps continu suivant :

$$x_e(t) = x(t) \text{III}_{T_e}(t),$$

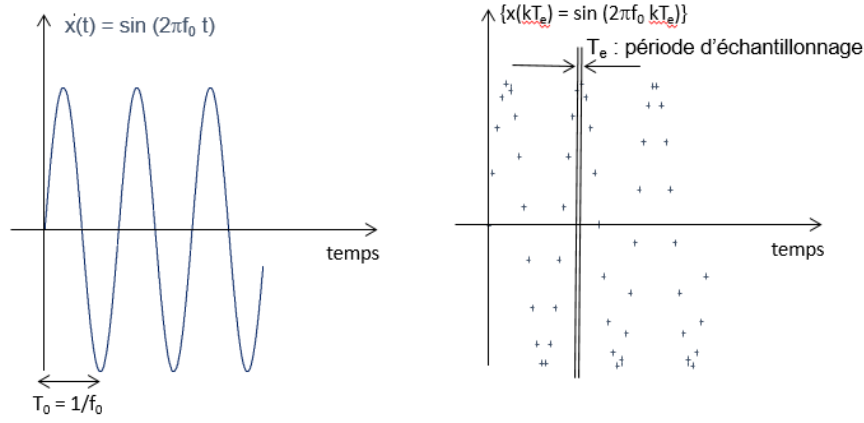


FIGURE 1.1 – Exemple de sinusoïde échantillonnée

où $\text{III}_{T_e}(t)$ représente le peigne de Dirac de largeur T_e . Si $X(f)$ est la Transformée de Fourier de $x(t)$ alors la transformée de Fourier de $x_e(t)$ est donnée par :

$$X_e(f) = \frac{1}{T_e} X(f) * \delta(f - kF_e) = F_e \sum_n X(f - kF_e) \quad (1.1)$$

où $F_e = \frac{1}{T_e}$ représente la fréquence d'échantillonnage (nombre d'échantillons prélevés par unité de temps). La transformée de Fourier du signal $x(t)$ est donc périodisée par l'opération d'échantillonnage. Afin de conserver la même information dans le signal échantillonné et dans le signal à temps continu, F_e doit être choisie de manière à respecter la condition suivante :

$$F_e > 2F_{max}, \quad (1.2)$$

si F_{max} représente la fréquence maximale de $X(f)$. Cette condition s'appelle la condition de Shannon. Claude Shannon est un ingénieur en génie électrique et un mathématicien né en 1916 aux Etats Unis. Il est un des pères de la théorie de l'information. Il a montré en 1949 que tout signal à temps continu dont le spectre est limité en bande pouvait être représenté, sans perte d'information, par une série d'échantillons du signal d'origine, à condition de correctement choisir la fréquence à laquelle on prélève ces échantillons.

1.1.2 Restitution par filtrage

- Si la condition de Shannon est respectée, il est alors possible de reconstituer le signal $x(t)$, à partir de la suite des échantillons prélevés tous les T_e , en utilisant un filtre passe-bas de fréquence de coupure $f_c \in [F_{max} F_e - F_{max}]$. En notant $y(t)$ le signal reconstitué et $Y(f)$ sa transformée de Fourier, nous avons :

$$Y(f) = H_{PB}(f)X_e(f)$$

et donc :

$$y(t) = h_{PB}(t) * x_e(t) = h_{PB}(t) * \sum_{k \in \mathbb{Z}} x(kT_e) \delta(t - kT_e) = \sum_{k \in \mathbb{Z}} x(kT_e) h_{PB}(t - kT_e)$$

La reconstitution par filtrage est donc équivalente à une interpolation. Le signal est reconstitué en sommant les décalages, tous les T_e , de la fonction d'interpolation $h_{PB}(t)$ pondérée par les échantillons du signal.

Lorsque $F_e = 2F_{max}$ (limite de Shannon), un seul filtre passe-bas est possible : c'est un filtre passe-bas idéal de réponse en fréquence $H_{PB}(f) = \Pi_{F_e}(f)$. Sa réponse impulsionnelle est donc $h_{PB}(t) = F_e \text{sinc}(\pi F_e t)$ et :

$$y(t) = F_e \sum_{k \in \mathbb{Z}} x(kT_e) \text{sinc}(\pi F_e(t - kT_e))$$

Cette expression est appelée formule de reconstitution de Shannon. Elle permet de voir l'échantillonnage idéal d'une autre manière : comme étant la décomposition du signal sur la base orthogonale des fonctions $\text{sinc}(\pi F_e(t - kT_e))$, $k \in \mathbb{Z}$.

- Si la condition de Shannon n'est pas respectée, il n'est alors plus possible de reconstituer le signal $x(t)$, à partir de la suite des échantillons prélevés tous les T_e , car les périodisations de $X(f)$ tous les F_e vont venir se superposer à $X(f)$. On parle de repliement ou d'aliasing (voir exercice 1.6.1).

1.1.3 Remarques

- Lorsque le signal $x(t)$ a un spectre de type passe-bande, il est possible de ne pas respecter la condition de Shannon tout en étant capable de reconstituer le signal de départ. La condition est alors que, par un choix astucieux de F_e , les repliements puissent se faire dans les trous du spectre de départ (voir exercice 1.6.2).
- L'échantillonnage présenté plus haut est un échantillonnage idéal. En pratique, il est impossible de prélever un échantillon de signal toutes les T_e secondes. On pourra, par exemple,
 - Utiliser un filtre, mais qui ne sera pas un filtre idéal. Il est alors nécessaire de suréchantillonner par rapport à la limite de Shannon.
 - Procéder par extrapolation : $x(kT_e + \tau) = x(kT_e)$ pour $0 \leq \tau \leq T_e$ (bloqueur d'ordre 0) ou $x(kT_e + \tau) = x(kT_e) + \frac{\tau}{T_e} (x(kT_e) - x((k-1)T_e))$ pour $0 \leq \tau \leq T_e$ (bloqueur d'ordre 1 ou extrapolateur linéaire)
 - Procéder par interpolation : par exemple $x(kT_e + \tau) = x(kT_e) + \frac{\tau}{T_e} (x((k+1)T_e) - x(kT_e))$, $0 \leq \tau \leq T_e$, pour un interpolateur d'ordre 1 (causal à un retard près)...

Les exercices 1.6.3 et 1.6.4 étudient deux méthodes "pratiques" d'échantillonnage.

- Lorsque le signal $x(t)$ ne présente pas de fréquence maximale F_{max} mais que son spectre décroît quand la fréquence augmente, il est alors possible de réaliser une opération d'échantillonnage avec une réversibilité acceptable. On se définit alors une fréquence maximale, correspondant à l'atténuation minimale souhaitée par rapport à la valeur maximale du spectre, on positionne un filtre dit filtre anti-repliement, qui va couper le spectre au delà de la fréquence maximale choisie, puis on échantillonne (voir exercice 1.6.5).
- On définit en numérique des fréquences normalisées : $\tilde{f} = \frac{f}{F_e}$ qui sont donc sans dimension et qui permettent de s'affranchir de la connaissance de la valeur de F_e dans les traitements à réaliser sur le signal.

1.2 Numérisation du signal : Quantification

1.2.1 Quantification : principe, impact

Un signal quantifié est un signal dont les amplitudes ne pourront prendre qu'un nombre fini de valeurs. Chaque valeur du signal sera approchée par un multiple entier d'une quantité élémentaire appelée pas de quantification. Le nombre de valeurs possibles, pour l'amplitude du signal après quantification, va être donné par le nombre de bits de quantification utilisés : avec nb bits on pourra coder 2^{nb} niveaux sur la dynamique D du signal. Dans le cas d'une quantification uniforme (pas de quantification q constant sur toute la dynamique du signal), le pas de quantification est donné par : $q = \frac{D}{2^{nb}}$. La figure 1.2 présente un exemple de quantification uniforme d'une fonction sinusoïdale.

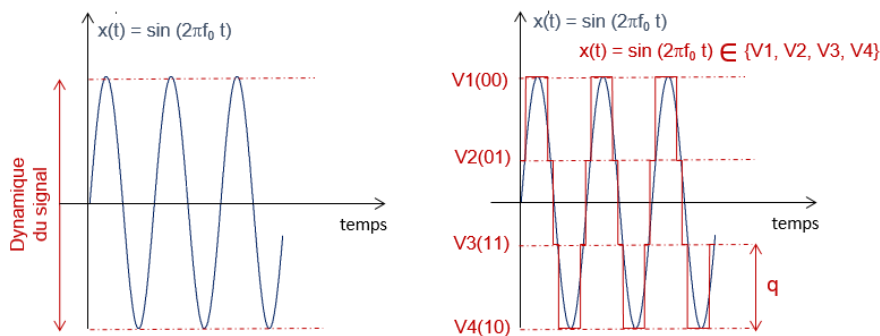


FIGURE 1.2 – Exemple de sinusoïde quantifiée.

Plusieurs types de quantification existent. On peut, par exemple, affecter tous les échantillons de signal appartenant à un niveau donné à la valeur min de ce niveau (quantification par troncature : on approche par nq toutes les valeurs de signal comprises entre nq et $(n+1)q$) ou bien affecter à la valeur nq toutes les valeurs de signal comprises entre $(n - \frac{1}{2})q$ et $(n + \frac{1}{2})q$ (quantification par arrondi). Dans tous les cas, l'opération de quantification est une opération non linéaire irréversible. Cependant, si elle est effectuée dans de bonnes conditions (pas d'écroulement du signal, pas de quantification suffisamment fin) elle est équivalente à l'ajout d'un bruit, $n_Q(t)$, sur le signal non quantifié de départ $x(t)$ pour donner le signal quantifié $x_Q(t)$ (voir figure 1.3 pour un exemple) :

$$x_Q(t) = x(t) + n_Q(t), \quad (1.3)$$

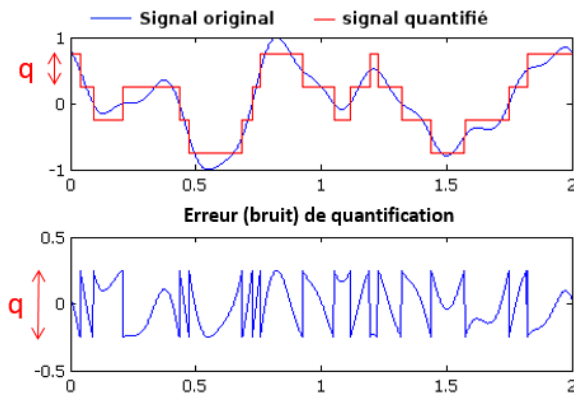


FIGURE 1.3 – Erreur, ou bruit, de quantification.

Ce bruit peut être modélisé comme un signal aléatoire de moyenne nulle, suivant une loi uniforme sur $[-\frac{q}{2}, \frac{q}{2}]$ et le rapport signal à bruit de quantification s'écrit :

$$SNR_Q(dB) = 10 \log_{10} \left(\frac{P_x}{P_{n_Q}} \right)$$

où

$$P_{n_Q} = E[n_Q^2] = \int_{\mathbb{R}} n_Q^2 p_{n_Q} dn_Q = \int_{-\frac{q}{2}}^{\frac{q}{2}} n_Q^2 \times \frac{1}{q} dn_Q = \frac{q^2}{12}$$

ce qui conduit à

$$SNR_Q(dB) = 6 nb + \text{constante},$$

où la constante dépend du signal considéré (voir exercice 1.6.6).

A l'heure actuelle, du fait du nombre de bits de quantification disponibles sur les processeurs, cette opération s'avère alors quasi transparente.

1.2.2 Remarque

Afin de diminuer le nombre de bits nécessaires pour quantifier un signal, il faudrait pouvoir adapter le pas de quantification à l'amplitude du signal d'entrée. On a alors une quantification non uniforme. En pratique cette opération est réalisée en utilisant une compression avant quantification uniforme, de manière à amplifier les faibles amplitudes et à minimiser l'effet des fortes amplitudes. Deux lois de compression sont normalisées et utilisées : la loi A et la loi μ . Ce sont deux approximations de la loi logarithmique utilisée pour les signaux audio dans les applications traitant la voix humaine. L'échantillon de signal, y , en sortie de la compression est donné, en fonction de l'échantillon du signal en entrée x , par :

— Loi A :

$$\begin{aligned} y &= \operatorname{sgn}(x) \frac{A|x|}{1 + \ln(A)} \quad 0 \leq |x| \leq \frac{1}{A} \\ &= \operatorname{sgn}(x) \frac{1 + \ln(A|x|)}{1 + \ln(A)} \quad \frac{1}{A} \leq |x| \leq 1 \end{aligned}$$

La loi A est utilisée principalement en Europe avec un paramètre de compression de 87,6. Elle permet, en téléphonie par exemple, d'utiliser 8 bits de quantification au lieu des 12 qui seraient nécessaires avec un quantifieur uniforme étant donnée la dynamique du signal à quantifier (figure 1.4).

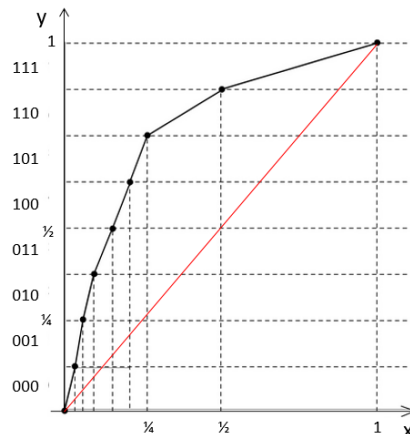


FIGURE 1.4 – Caractéristique de compression à 13 segments (loi A)

- Loi $\mu : y = \operatorname{sgn}(x) \frac{\ln(1+\mu|x|)}{\ln(1+\mu)}$, $-1 \leq x \leq 1$, utilisée principalement aux États-Unis et au Japon avec $\mu = 255$.

1.3 Numérisation du signal : exemples sur une image

La figure 1.5 présente une image de 512×512 pixels, codée sur $nb = 8$ bits et sa version sous échantillonnée d'un facteur 4 (128×128 pixels). Cette image (Barbara) est très utilisée en traitement d'images. Elle permet ici de voir apparaître le phénomène de Moiré : on voit apparaître sur l'image sous échantillonnée des structures différentes de celles contenues dans l'image d'origine.

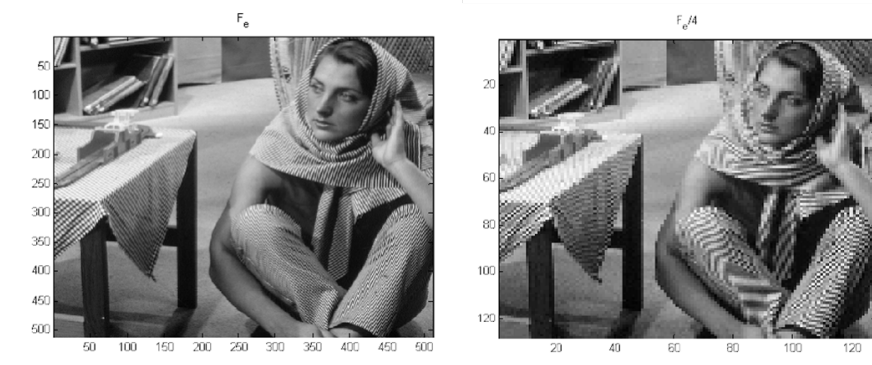


FIGURE 1.5 – Image de départ (quantifiée sur 8 bits, 512×512 pixels), Image sous échantillonnée d'un facteur 4

La figure 1.6 présente Barbara codée sur $nb = 4$ bits et sur $nb = 2$ bits. Dans le dernier cas, par exemple, on ne voit alors plus que $2^{nb} = 2^2 = 4$ niveaux de gris (au lieu de $2^8 = 256$ allant du noir au blanc sur l'image de départ).

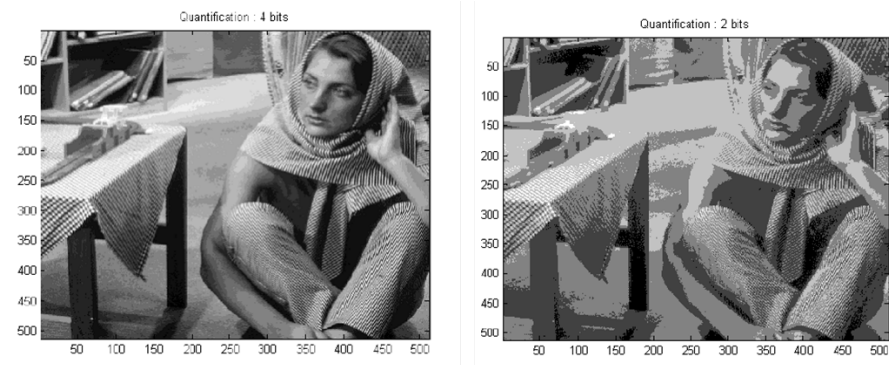


FIGURE 1.6 – Image quantifiée sur $nb = 4$ bits, Image quantifiée sur $nb = 2$ bits

1.4 Outils de traitement du signal à numériser

Un certain nombre d'outils de traitement du signal sont définis pour des signaux analogiques (signaux définis à chaque instant par des valeurs réelles) :

- La transformée de Fourier et la densité spectrale de puissance, permettant de visualiser la représentation fréquentielle des signaux pour en extraire de l'information (exemples : bande occupée par le signal en vue d'une transmission, détection de défauts apparaissant comme des composantes fréquentielles particulières).

- Les fonctions d’auto et d’inter corrélation, permettant d’accéder à la densité spectrale de puissance du signal, mais également utiles pour d’autres fonctions (exemple : extraction d’un signal dans du bruit).
- Les filtres, permettant de construire et de modifier des signaux (exemples : modulation, réduction du bruit, suppression de certaines composantes fréquentielles).

Il va être nécessaire de réaliser un certain nombre d’approximations, d’estimations pour passer des outils précédents, définis de manière théorique sur des signaux analogiques, aux outils que l’on va être capable d’implanter en numérique. L’objectif de ce cours de traitement numérique du signal est de présenter, d’expliquer ces modifications et leurs impacts sur les résultats attendus afin d’être capable d’utiliser correctement les outils numériques et d’analyser correctement les résultats obtenus.

1.5 Notion de temps de traitement et de traitement temps réel

Comme on pourra le constater par la suite, tous les traitements numériques des signaux sont basés sur une même opération de base qui est l’opération d’addition/multiplication (ou MAC = Multiplication Accumulation). Le temps nécessaire à un traitement sera donc évalué en nombre d’addition/multiplication nécessaires. Un traitement est accompli en temps réel quand il délivre un échantillon de signal en sortie de traitement pour un échantillon de signal en entrée (c’est-à-dire à chaque période d’échantillonnage T_e).

1.6 Exercices avec éléments de correction

1.6.1 Exercice 1 : Effet de l'échantillonnage

Soit le signal suivant : $x(t) = \cos(2\pi f_0 t)$, $f_0 = 10$ kHz.

1. Tracer la transformée de Fourier de $x(t)$: $X(f)$.

La transformée de Fourier de $x(t)$, $X(f)$, est tracée sur la figure 1.7.

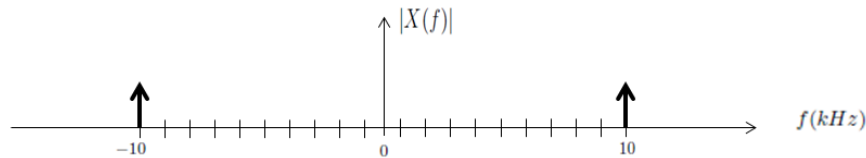


FIGURE 1.7 – Transformée de Fourier de $x(t) = \cos(2\pi f_0 t)$, $f_0 = 10$ kHz.

2. Est-il possible d'échantillonner $x(t)$ sans perte d'information ? Si oui à quelle condition ?

Il est possible d'échantillonner $x(t)$ sans perte d'information en utilisant une fréquence d'échantillonnage $F_e > 2f_0 = 20$ kHz (respect de la condition de Shannon).

3. Tracer, entre 0 et F_e , la transformée de Fourier de $x(t)$ échantillonné à $T_e = 1/F_e$ quand :

(a) $F_e = 30$ kHz.

(b) $F_e = 8$ kHz.

La transformée de Fourier de $x(t)$, échantillonné à $T_e = 1/F_e$, est tracée entre 0 et F_e sur la figure 1.8 quand $F_e = 30$ kHz et sur la figure 1.9 quand $F_e = 8$ kHz.

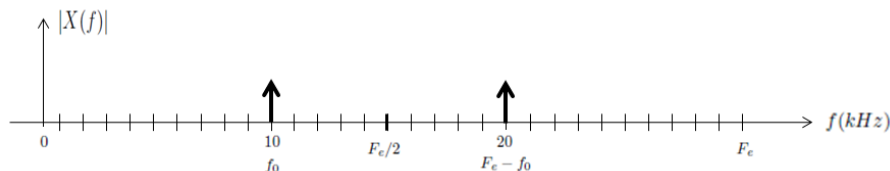


FIGURE 1.8 – Transformée de Fourier de $x(t) = \cos(2\pi f_0 t)$, $f_0 = 10$ kHz, $F_e = 30$ kHz.

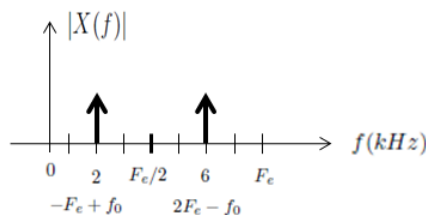


FIGURE 1.9 – Transformée de Fourier de $x(t) = \cos(2\pi f_0 t)$, $f_0 = 10$ kHz, $F_e = 8$ kHz.

4. A partir des échantillons nous souhaitons reconstruire $x(t)$ par filtrage passe-bas à $F_e/2$.

Quels seront les signaux obtenus pour chaque fréquence d'échantillonnage précédente ?

Par filtrage passe-bas à $F_e/2$, nous obtenons $x(t) = \cos(2\pi f_0 t)$, avec $f_0 = 10$ kHz pour $F_e = 30$ kHz, et $x(t) = \cos(2\pi f_1 t)$, avec $f_1 = 2$ kHz pour $F_e = 8$ kHz.

1.6.2 Exercice 2 : Echantillonnage d'un signal passe-bande

On considère le signal $x(t) = x^+(t) + x^-(t)$, avec $x^+(t) = B \frac{\sin(\pi B t)}{\pi B t} e^{j2\pi f_0 t}$ et $x^-(t) = B \frac{\sin(\pi B t)}{\pi B t} e^{-j2\pi f_0 t}$, $f_0 = 8 \text{ kHz}$ et $B = 2 \text{ kHz}$.

1. Déterminer la transformée de Fourier du signal $x(t)$ et la représenter graphiquement.

$$X(f) = X^+(f) + X^-(f) = \Pi_B(f) * \delta(f - f_0) + \Pi_B(f) * \delta(f + f_0) = \Pi_B(f - f_0) + \Pi_B(f + f_0)$$

(voir figure 1.10)

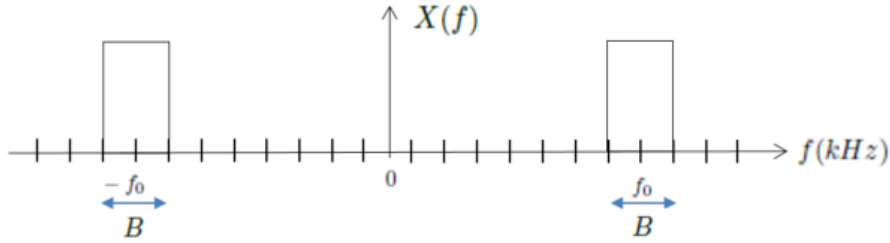


FIGURE 1.10 – Transformée de Fourier de $x(t)$.

2. Comment s'écrit la condition de Shannon pour le signal $x(t)$?

$$F_e > 2F_{max} \text{ avec } F_{max} = f_0 + \frac{B}{2} = 9 \text{ kHz ici.}$$

3. On échantillonne le signal $x(t)$ à la fréquence $F_e = 6 \text{ kHz}$.

- (a) Représenter graphiquement la transformée de Fourier du signal échantillonné $x_e(t)$ dans la bande $[-9 \text{ kHz}, 9 \text{ kHz}]$

Voir sur la figure 1.11

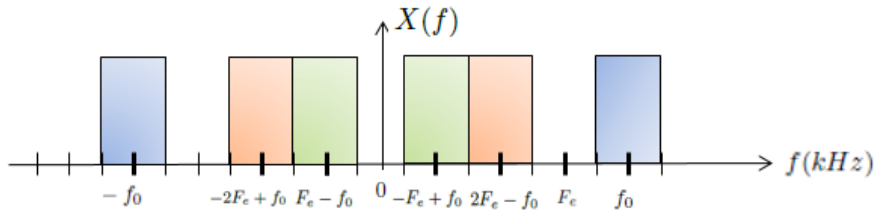


FIGURE 1.11 – Transformée de Fourier de $x(t)$ avec $F_e = 8 \text{ kHz}$.

- (b) On désire restituer le signal $x(t)$ à partir de $x_e(t)$ par un filtrage de réponse en fréquence $H(f)$.

- 1^{ier} cas : $H(f) = \Pi_F(f)$ avec $F = 6 \text{ kHz}$. Quel sera le signal restitué par ce filtre ?
Voir la figure 1.12, on retrouvera $x(t) = B \frac{\sin(\pi B t)}{\pi B t} e^{j2\pi f_1 t} + B \frac{\sin(\pi B t)}{\pi B t} e^{-j2\pi f_1 t} = 2B \text{sinc}(\pi B t) \cos(2\pi f_1 t)$, avec $f_1 = -F_e + f_0 = 2 \text{ kHz}$.

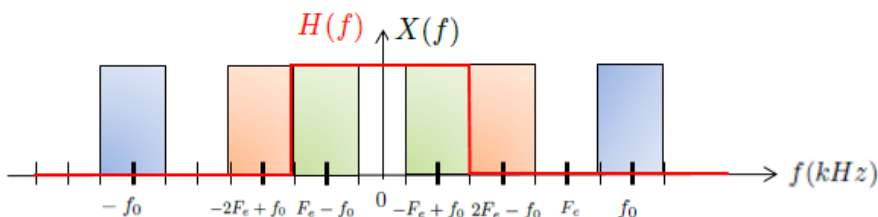


FIGURE 1.12 –

- 2^{me} cas : $H(f) = \Pi_B(f + f_0) + \Pi_B(f - f_0)$ avec $f_0 = 8\text{kHz}$ et $B = 2\text{kHz}$. Quel sera le signal restitué par ce filtre ?

Voir la figure 1.13, on retrouvera $x(t) = B \frac{\sin(\pi Bt)}{\pi Bt} e^{j2\pi f_0 t} + B \frac{\sin(\pi Bt)}{\pi Bt} e^{-j2\pi f_0 t} = 2B \text{sinc}(\pi Bt) \cos(2\pi f_0 t)$, avec $f_0 = 8\text{kHz}$.

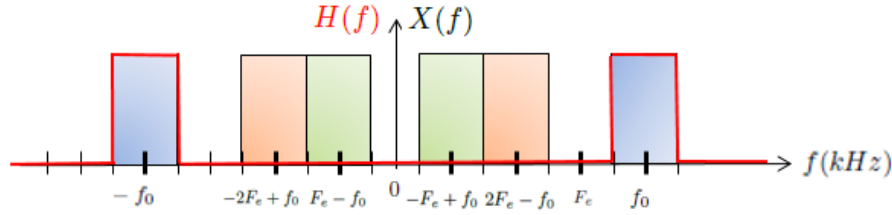


FIGURE 1.13 –

- Conclusion ?

Il est possible d'échantillonner un signal de type passe-bande sans respecter la condition de Shannon tout en assurant une reconstitution parfaite (par filtrage passe-bande), à condition que les repléments se fassent dans les trous du spectre de départ.

1.6.3 Exercice 3 : Echantillonneur moyennneur

L'échantillonneur moyennneur est une méthode pratique d'échantillonnage qui consiste à calculer, toutes les T_e secondes (période d'échantillonnage), la valeur moyenne du signal pendant un intervalle de temps θ ($\theta < T$) et à affecter cette valeur moyenne à l'échantillon discrétisé :

$$y(kT_e) = \frac{1}{\theta} \int_{kT_e - \theta}^{kT_e} x(u) du$$

$$x_{ech}(t) = \sum_k y(kT_e) \delta(t - kT_e)$$

1. Démontrer que le signal échantillonné $x_{ech}(t)$ peut se mettre sous la forme :

$$x_{ech}(t) = \frac{1}{\theta} \left[\Pi_{\theta}(t) * x\left(t - \frac{\theta}{2}\right) \right] \cdot \text{III}_{T_e}(t)$$

où $\Pi_{\theta}(t)$ et $\text{III}_{T_e}(t)$ représentent respectivement la fenêtre rectangulaire de largeur θ et le peigne de Dirac de période T_e .

$x_{ech}(t) = \sum_k y(kT_e) \delta(t - kT_e) = y(t) \sum_k \delta(t - kT_e) = y(t) \cdot \text{III}_{T_e}(t)$. Reste à montrer que $y(t) = \frac{1}{\theta} [\Pi_{\theta}(t) * x(t - \frac{\theta}{2})]$:

$$y(t) = \frac{1}{\theta} \int_{t-\theta}^t x(u) du = \frac{1}{\theta} \int_{-\infty}^{+\infty} x(u) \times \Pi_{\theta}(u - (t - \frac{\theta}{2})) du = \frac{1}{\theta} \int_{-\infty}^{+\infty} x(u) \times \Pi_{\theta}((t - \frac{\theta}{2}) - u) du = \frac{1}{\theta} (x * \Pi_{\theta})(t - \frac{\theta}{2})$$

2. En déduire la transformée de Fourier correspondante $X_{ech}(f)$. $X_{ech}(f) = Y(f) * \frac{1}{T_e} \Pi_{1/T_e}(f) = \frac{1}{T_e} \sum_k Y\left(f - \frac{k}{T_e}\right)$, avec $Y(f) = \text{sinc}(\pi f \theta) X(f) e^{-j\pi f \theta}$
3. En considérant un signal à support spectral borné $2\Delta f$ et en prenant en compte que la fonction $\text{sinc}(\pi \theta f)$ peut être supposé constante sur l'intervalle $[-\frac{1}{3\theta}, \frac{1}{3\theta}]$

$$\text{sinc}(\pi \theta f) = \frac{\sin(\pi \theta f)}{\pi \theta f} \approx 1 \text{ pour } f \in \left[-\frac{1}{3\theta}, \frac{1}{3\theta}\right]$$

- (a) quelle(s) condition(s) doit vérifier θ pour que le signal $x(t)$ puisse être restitué par filtrage de $x_{ech}(t)$?

Il faut que $\Delta f \leq \frac{1}{3\theta} \Leftrightarrow \theta \leq \frac{1}{3\Delta f}$

(b) Dans ces conditions peut-on échantillonner à la fréquence de Shannon ?

Après filtrage antialiasing on pourra prendre F_e tel que $\Delta f < \frac{F_e}{2} = \frac{1}{2T_e} \Leftrightarrow T_e < \frac{1}{2\Delta f}$

1.6.4 Exercice 4 : Echantillonneur bloqueur

L'échantillonneur bloqueur est un échantillonneur réalisable en pratique qui consiste à acquérir un échantillon du signal, $x(t)$, toutes les T_e secondes (période d'échantillonnage) et à le bloquer pendant τ secondes ($\tau \ll T_e$).

1. Proposer une écriture du signal échantillonné de cette manière, $x_e(t)$, en fonction de l'expression du signal échantillonné de manière idéale : $x_{ei}(t) = \sum_{k \in \mathbb{Z}} x(kT_e) \delta(t - kT_e)$.

Le signal échantillonné par bloqueur va être constitué d'une somme de fonctions porte espacées de T_e , de largeur τ et de hauteur $x(kT_e)$ si $x(kT_e)$ représente la valeur de l'échantillon prélevé sur le signal $x(t)$ à l'instant kT_e . On peut donc écrire le signal échantillonné, $x_e(t)$, de la manière suivante : $x_e(t) = \sum_{k \in \mathbb{Z}} x(kT_e) \Pi_\tau(t - \frac{\tau}{2} - kT_e) = \Pi_\tau(t - \frac{\tau}{2}) * \sum_{k \in \mathbb{Z}} x(kT_e) \delta(t - kT_e) = \Pi_\tau(t - \frac{\tau}{2}) * x_{ei}(t)$.

2. Calculer la transformée de Fourier du signal échantillonné à l'aide de cette méthode. L'écrire en fonction de la transformée de Fourier, $X(f)$, du signal de départ.

$X_e(f) = \tau \text{sinc}(\pi f \tau) e^{-j\pi f \tau} * X_{ei}(f) = \tau \text{sinc}(\pi f \tau) e^{-j\pi f \tau} * F_e \sum_{k \in \mathbb{Z}} X(f - kF_e)$, où $F_e = \frac{1}{T_e}$ représente la fréquence d'échantillonnage du signal.

3. Est-il possible de dimensionner τ pour que l'échantillonnage par bloqueur se rapproche d'un échantillonnage idéal ?

Si le critère de Shannon est vérifié, on pourra récupérer $X(f)$ à condition que $\frac{1}{\tau} \gg F_{max}$, en appelant F_{max} la fréquence maximale du signal $x(t)$. On aura alors, en effet, $\text{sinc}(\pi f \tau) \simeq 1$ sur la bande du signal.

1.6.5 Exercice 5 : Signal à spectre non borné - Recherche de la F_e

Soit le signal $x(t)$ défini par :

$$x(t) = \begin{cases} e^{-at} & \text{si } t \geq 0, a > 0 \\ 0 & \text{si } t < 0. \end{cases} \quad (1.4)$$

1. Déterminer la transformée de Fourier $X(f)$ du signal $x(t)$. Tracer $|X(f)|$.

$$X(f) = \int_0^{+\infty} e^{-(a+j2\pi f)t} dt = \frac{1}{a+j2\pi f}, |X(f)| = \frac{1}{\sqrt{a^2+4\pi^2 f^2}}.$$

2. En théorie le signal $x(t)$ est-il échantillonnable sans perte d'information ? Expliquez votre réponse.

Non car le spectre non borné \Rightarrow forcément du repliement quand on va échantillonner \Rightarrow signal distordu.

3. En considérant la transformée de Fourier comme négligeable pour une atténuation minimale de 40 dB par rapport à sa valeur maximum, dimensionner la fréquence d'échantillonnage, F_e , à utiliser.

On a le maximum du spectre pour $f = 0$. On souhaite donc trouver F_{max} telle que :

$$10 \log_{10} |X(F_{max})|^2 \leq 10 \log_{10} |X(0)|^2 - 10 \log_{10} (10^4) = 10 \log_{10} \frac{|X(0)|^2}{10^4}$$

D'où $\frac{1}{\sqrt{a^2 + 4\pi^2 F_{max}^2}} \leq \frac{1}{10^4 a^2}$ et donc $F_{max}^2 \geq \frac{(10^4 - 1)a^2}{4\pi^2}$. Soit, en négligeant 1 devant 10^4 :
 $F_{max} \geq \frac{100a}{2\pi}$ et donc $F_e \geq \frac{100a}{\pi}$.

4. Une fois F_e déterminée, quel traitement doit-on appliquer au signal avant de l'échantillonner ?

Un filtre anti repliement afin de tronquer le spectre du signal à F_{max} .

1.6.6 Exercice 6 : Quantification d'un sinusoïde

Soit un signal sinusoïdal $x(t) = A_0 \sin(2\pi f_0 t + \phi)$, avec $f_0 = 50\text{Hz}$, $A_0 = 220\sqrt{2}\text{V}$ et ϕ une phase aléatoire uniformément répartie entre 0 et 2π . On suppose que la quantification de cette sinusoïde est effectuée dans de bonnes conditions : pas d'écrtage du signal, pas de quantification $q = \frac{D}{2^{nb}}$ suffisamment fin (D représentant la dynamique du signal et nb le nombre de bits de quantification). Elle est donc équivalente à l'ajout d'un bruit, $n_Q(t)$, sur le signal non quantifié de départ, bruit aléatoire, centré qui suit une loi uniforme sur $[-\frac{q}{2}, \frac{q}{2}]$. Déterminer le rapport signal à bruit de quantification en fonction de nb .

$SNR_{dB} = 10 \log_{10} \left(\frac{P_x}{P_n} \right)$ si P_x représente la puissance du signal $x(t)$ et P_n la puissance du bruit de quantification, $n_Q(t)$, qui vient s'ajouter au signal de départ. $P_x = \frac{A_0^2}{2}$ (résultat classique pour la puissance d'un sinus ou d'un cosinus, calculé en TD dans le cas d'un cosinus) et $P_n = E[n_Q^2(t)] = \int_{-\frac{q}{2}}^{\frac{q}{2}} \frac{1}{q} n_Q^2(t) dn_Q = \frac{1}{q} \left[\frac{n_Q^3(t)}{3} \right]_{-\frac{q}{2}}^{\frac{q}{2}} = \frac{q^2}{12}$, d'où $SNR_{dB} = 10 \log_{10} \left(\frac{3}{2} 2^{2nb} \right) \simeq 1.76 + 6nb$

Chapitre 2

Transformée de Fourier Discrète (TFD)

Pour un signal $x(t)$, la transformée de Fourier $X(f)$ et son inverse sont données par :

$$X(f) = \int_{\mathbb{R}} x(t) e^{-j2\pi ft} dt \quad (2.1)$$

$$x(t) = \int_{\mathbb{R}} X(f) e^{+j2\pi ft} df \quad (2.2)$$

2.1 De la TF à la TFD

Un signal en numérique est un tableau de points contenant un nombre fini, N , de valeurs de signal : $[x(0) \ x(1) \dots \ x(N-1)]$, le $k^{\text{ième}}$ élément $x(k)$ représentant en réalité $x(kT_e)$ si on considère un échantillonnage temporel périodique de période T_e . On travaille donc avec des signaux échantillonnés et limités dans le temps et il n'est pas possible d'utiliser l'expression 2.3 pour déterminer leurs transformées de Fourier. Des approximations doivent être effectuées à partir de 2.3 pour obtenir un outil numérique capable d'estimer une représentation fréquentielle et les effets induits par ces approximations doivent être connus de manière à être capable de mener correctement une analyse spectrale en numérique.

L'objectif de ce paragraphe va être de lister les approximations à réaliser pour passer de la Transformée de Fourier (TF) définie par l'expression 2.3 à la Transformée de Fourier numérique ou Discrète (TFD) qui sera finalement donnée par l'expression suivante :

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \ n = 0, \dots, N-1 \quad (2.3)$$

2.1.1 Echantillonnage

Un signal numérique est forcément échantillonné :

$$x(t) \rightarrow \{x(kT_e)\}_{k \in \mathbb{Z}} \quad (2.4)$$

L'échantillonnage du signal va avoir pour effet de périodiser sa transformée de Fourier, qui est approchée par la somme des aires des rectangles sous la courbe, au facteur T_e près :

$$X(f) \rightarrow X_1(f) = \sum_{k=-\infty}^{+\infty} x(kT_e) e^{-j2\pi f k T_e} \quad (2.5)$$

$X_1(f)$ est bien périodique de période $F_e = \frac{1}{T_e}$: $X_1(f + F_e) = X_1(f)$. On devra donc faire attention au respect du théorème d'échantillonnage de Shannon pour ne pas provoquer de recouvrement lors de l'échantillonnage. On devra également faire attention à la manière dont on lit le spectre numérique. En effet si la TFD est observée sur une période, entre 0 et F_e , la partie positive du spectre sera observée entre 0 et $F_e/2$, tandis que la partie négative le sera entre $F_e/2$ et F_e .

2.1.2 Limitation de la durée du signal à N points

Un signal numérique est forcément observé sur un nombre fini de points :

$$\{x(kT_e)\}_{k \in \mathbb{Z}} \rightarrow \{x(kT_e)\}_{k=0, \dots, N-1} \quad (2.6)$$

Cette connaissance du signal sur un nombre limité de points (dimension du tableau représentant le signal numérique) conduit à une distorsion de la transformée de Fourier :

$$X_1(f) \rightarrow X_2(f) = \sum_{k=0}^{N-1} x(kT_e) e^{-j2\pi f k T_e} = \sum_{k=-\infty}^{+\infty} x(kT_e) w(kT_e) e^{-j2\pi f k T_e} \quad (2.7)$$

qui donne :

$$X_2(f) = X_1(f) * W_1(f) \quad (2.8)$$

$$\text{où } W_1(f) = \sum_{k=-\infty}^{+\infty} w(kT_e) e^{-j2\pi f k T_e} \text{ avec } w(kT_e) = \begin{cases} 1 & k=0, \dots, N-1 \\ 0 & \text{ailleurs.} \end{cases}$$

Cette distorsion de la transformée de Fourier implique un pouvoir séparateur limité pour l'analyse spectrale (possibilité de dissocier 2 motifs spectraux de fréquences proches) et un certain taux d'ondulation (des ondulations apparaissent autour des transitions brutales du spectre : phénomène de Gibbs).

Ces paramètres (pouvoir séparateur et taux d'ondulation) sont liés à la forme de $W_1(f)$, plus précisément à la largeur de son lobe principal et à l'amplitude de ses lobes secondaires (voir figure 2.1).

On pourra faire varier le pouvoir séparateur et le taux d'ondulation de l'analyse spectrale en tronquant le signal étudié avec différentes fenêtres $w(kT_e)$, autres que rectangulaire (fenêtres de pondération ou d'apodisation), de manière à obtenir différentes formes pour $W_1(f)$ et donc différentes versions de la TFD du même signal.

La figure 2.2 présente quelques fenêtres classiques d'apodisation, tandis que la figure 2.3 présente leurs transformées de Fourier. On peut constater que les lobes centraux sont plus ou moins larges, conduisant à un pouvoir séparateur plus ou moins grand pour l'analyse spectrale. On peut également constater que les lobes secondaires sont plus ou moins atténués, conduisant à un taux d'ondulation plus ou moins grand pour l'analyse spectrale. Lorsque des fenêtres autres que rectangulaires sont utilisées on parle de transformée de Fourier pondérée.

On réalisera chaque analyse spectrale en utilisant plusieurs fenêtres de pondération. Chacune permettra d'avoir une visualisation différente du spectre d'un même signal et d'en extraire des informations différentes. La figure 2.4 présente un exemple dans lequel les différentes fenêtres utilisées permettent de mettre en exergue différentes composantes spectrales. Mais quel est donc le signal qui pourrait correspondre à ces différents spectres ?

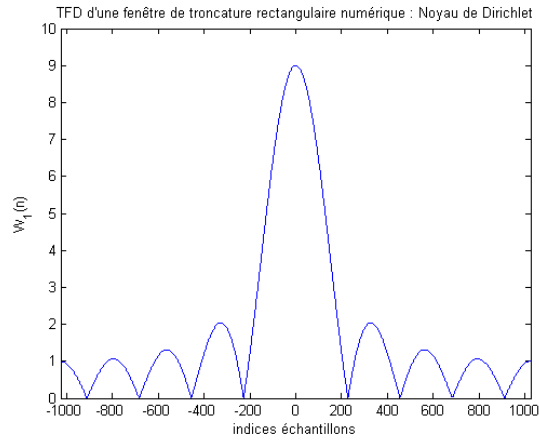


FIGURE 2.1 – Transformées de Fourier de la fenêtre rectangulaire : Noyau de Dirichlet.

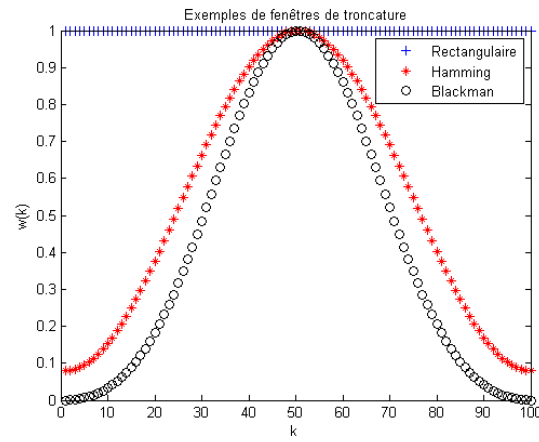


FIGURE 2.2 – Quelques exemples de fenêtres d'apodisation.

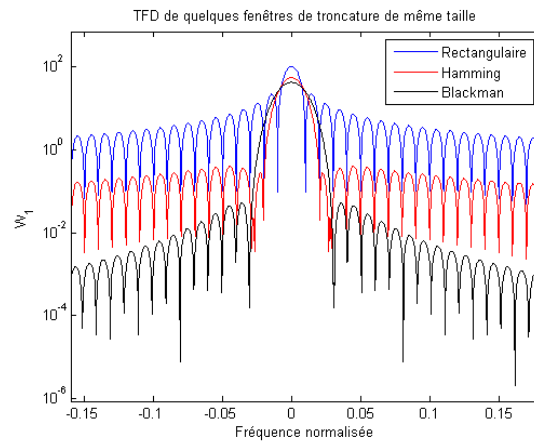


FIGURE 2.3 – Transformées de Fourier de quelques fenêtres de troncature.

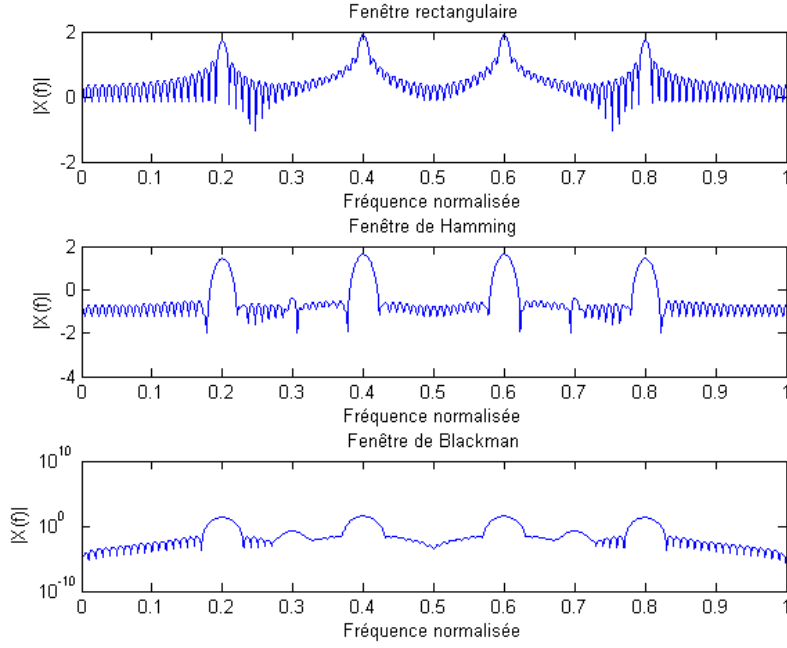


FIGURE 2.4 – Différentes versions de la transformée de Fourier d'un même signal.

2.1.3 Calcul de N points du spectre

Tout comme le signal numérique ne peut pas être à temps continu, il ne sera possible de calculer qu'un nombre fini d'échantillons de la TFD :

$$X_2(f) \rightarrow \{X_2(n\Delta f)\}_{n=0,\dots,N-1} \quad (2.9)$$

En considérant que l'on calcule, sur une période F_e , un nombre de point de la TFD identique au nombre N de points de signal, on obtient un pas de calcul $\Delta f = \frac{F_e}{N}$ et donc une transformée de Fourier discrète qui s'écrit :

$$X\left(n\frac{F_e}{N}\right) = \sum_{k=0}^{N-1} x(kT_e) e^{-j2\pi n \frac{F_e}{N} kT_e} = \sum_{k=0}^{N-1} x(kT_e) e^{-j2\pi \frac{nk}{N}}, \quad n = 0, \dots, N-1 \quad (2.10)$$

Le fait de ne pouvoir calculer qu'un certain nombre de points de la transformée de Fourier numérique a un impact sur la résolution de l'analyse spectrale. Celle-ci sera liée au nombre de points calculés sur une période de la TFD : pas de calcul $\Delta f = \frac{F_e}{N}$. Afin d'augmenter la résolution spectrale, on pourra utiliser une technique d'interpolation fréquentielle, la plus connue et utilisée étant celle du Zero Padding. On construit un nouveau signal, à partir du signal $x(kT_e)$ donné sur N points, en le prolongeant par des zéros :

$$y(kT_e) = \begin{cases} x(kT_e) & k=0, \dots, N-1 \\ 0 & k=N, \dots, MN-1. \end{cases}$$

La TFD de ce nouveau signal :

$$Y\left(n\frac{F_e}{N}\right) = \sum_{k=0}^{N-1} x(kT_e) e^{-j2\pi \frac{kn}{MN}}, \quad n = 0, \dots, MN-1$$

dispose d'un pas de calcul plus fin : $\frac{F_e}{MN}$. On calcule MN points distants de $\frac{F_e}{MN}$ entre 0 et F_e (une période de la TFD), au lieu de N points distants de $\frac{F_e}{N}$ entre 0 et F_e . La résolution spectrale est donc améliorée.

Les figures 2.5 et 2.6 proposent des tracés du module de la transformée de Fourier numérique d'un cosinus numérique observé sur $N = 1000$ points, pour différentes valeurs du paramètre MN . Ces tracés sont donnés avec une échelle en fréquences normalisées : $\tilde{f} = \frac{f}{F_e}$.

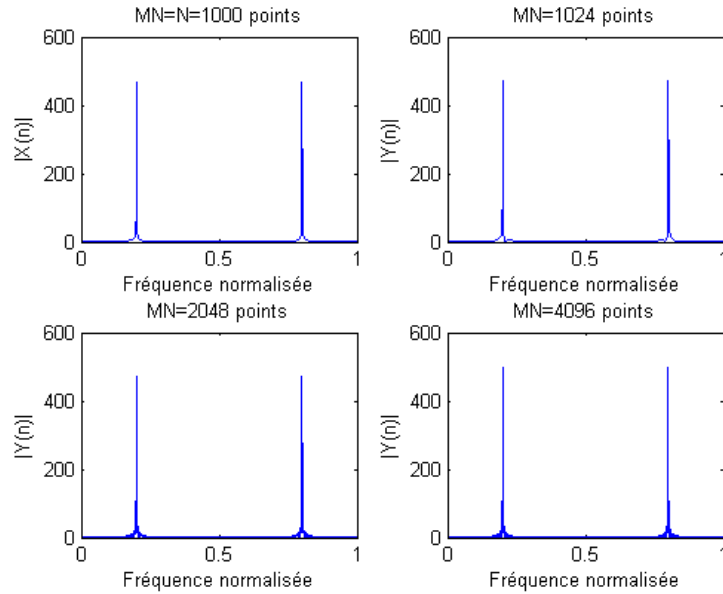


FIGURE 2.5 – Transformées de Fourier d'un cosinus de fréquence normalisée 0.2 pour différentes valeurs du paramètre de zero padding.

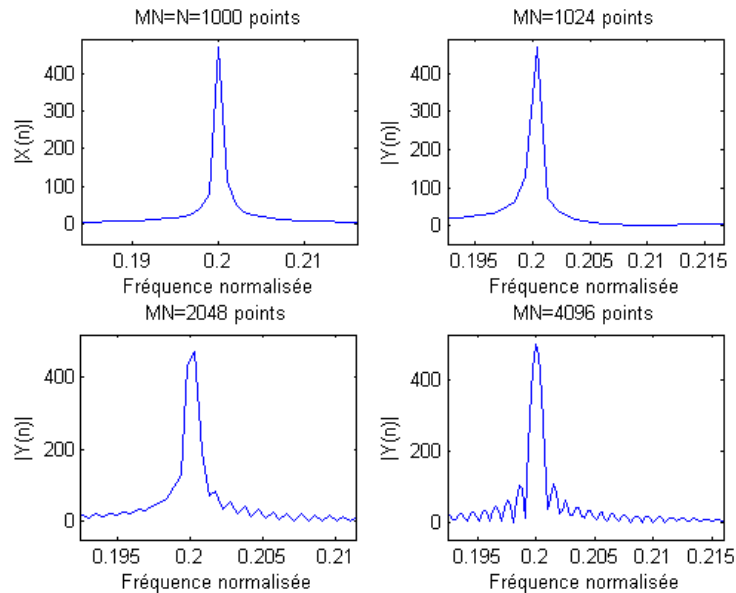


FIGURE 2.6 – Transformées de Fourier d'un cosinus de fréquence normalisée 0.2 pour différentes valeurs du paramètre de zero padding : zoom un pic de la figure 2.5

Un autre impact de la discrétisation fréquentielle est une périodisation temporelle. On doit, en effet, considérer en numérique que les signaux sont périodisés, de même que leurs transformées de Fourier. Ce dernier point a une conséquence importante : cela ne permet pas de conserver à

la TFD la propriété très intéressante de la TF qui consiste à transformer un produit de convolution en produit (et inversement). En effet, comme nous le verrons par la suite (section 2.2), la transformée de Fourier Discrète transforme un produit en produit de convolution circulaire, c'est-à-dire en produit de convolution entre des signaux périodisés. Néanmoins, il est possible de faire en sorte que le produit de convolution linéaire (produit de convolution "classique") et le produit de convolution circulaire soient identiques en prolongeant les signaux de longueur N par au moins N zéros.

2.1.4 Expressions de la TFD et de la TFD inverse

Les trois approximations précédemment étudiées (échantillonnage temporel, limitation de la durée du signal et échantillonnage fréquentiel) conduisent finalement à l'outil suivant pour calculer la transformée de Fourier en numérique :

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}}, \quad n = 0, \dots, N-1 \quad (2.11)$$

Le même cheminement conduirait à obtenir l'expression de la TFD inverse :

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) e^{+j2\pi \frac{kn}{N}}, \quad k = 0, \dots, N-1 \quad (2.12)$$

Notons que, dans ces expressions, $x(k)$ est une notation simplifiée pour $x(kT_e)$, de même que $X(n)$ est une notation simplifiée pour $X(n \frac{F_e}{N})$. On utilisera ces notations par la suite.

2.2 Propriétés de la TFD

2.2.1 Linéarité

La transformée de Fourier est linéaire :

$$TFD[x_1(k) + \lambda x_2(k)] = TFD[x_1(k)] + \lambda TFD[x_2(k)], \quad \lambda \text{ scalaire.} \quad (2.13)$$

2.2.2 Translation \Rightarrow rotation de phase

Une translation dans le domaine temporel entraîne un déphasage dans le domaine fréquentiel :

$$TFD[x(k - k_0)] = TFD[x(k)] e^{-j2\pi \frac{k_0 n}{N}} = X(n) e^{-j2\pi \frac{k_0 n}{N}} \quad (2.14)$$

Démonstration :

$$TFD[x(k - k_0)] = \sum_{k=k_0}^{N-1+k_0} x(k - k_0) e^{-j2\pi \frac{kn}{N}} = \sum_{m=0}^{N-1} x(m) e^{-j2\pi \frac{(m+k_0)n}{N}} = e^{-j2\pi \frac{k_0 n}{N}} X(n)$$

2.2.3 Symétrie Hermitienne

Soit $X(n)$ la transformée de Fourier discrète d'un signal réel $x(k)$, on a :

$$X(N - n) = X(-n) = X^*(n) \quad (2.15)$$

Démonstration :

$$\text{Si } x(k) \text{ réel alors } x^*(k) = x(k) \text{ et } X(-n) = \sum_{k=0}^{N-1} x(k) e^{j2\pi \frac{kn}{N}} = \left[\sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}} \right]^* = X^*(n).$$

D'autre part $X(N - n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{k(N-n)}{N}} = \sum_{k=0}^{N-1} x(k) e^{j2\pi \frac{kn}{N}} = X(-n)$.

2.2.4 Convolution circulaire

Le produit de convolution linéaire ("classique") est donné en numérique par :

$$(x_1 * x_2)(k) = \sum_{p=0}^{N-1} x_1(p) x_2(k - p) \quad (2.16)$$

Attention : la transformée de Fourier discrète ne transforme pas un produit en produit de convolution linéaire mais en produit de convolution circulaire :

$$X_1(n) X_2(n) \xrightarrow{TFD^{-1}} (x_1 \otimes x_2)(k) = \sum_{p=0}^{N-1} x_1(p) x_2([k - p]_{\text{modulo } N}) \quad (2.17)$$

si $X_1(n)$ est la transformée de Fourier discrète de $x_1(k)$ et $X_2(n)$ la transformée de Fourier discrète $x_2(k)$.

Démonstration :

$$\begin{aligned} TFD^{-1}[X_1(n) X_2(n)] &= \frac{1}{N} \sum_{n=0}^{N-1} X_1(n) X_2(n) e^{j2\pi \frac{kn}{N}} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \left(\sum_{p=0}^{N-1} x_1(p) e^{-j2\pi \frac{pn}{N}} \right) \left(\sum_{q=0}^{N-1} x_2(q) e^{-j2\pi \frac{qn}{N}} \right) e^{j2\pi \frac{kn}{N}} \\ &= \frac{1}{N} \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} x_1(p) x_2(q) \sum_{n=0}^{N-1} e^{-j2\pi \frac{ln}{N}}, \text{ avec } l = p + q - k. \end{aligned}$$

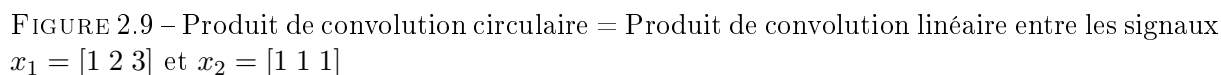
Or nous avons :

$$\sum_{n=0}^{N-1} e^{-j2\pi \frac{ln}{N}} = \begin{cases} 0 & l \neq iN \\ N & l = iN \Leftrightarrow q = k - p + iN. \end{cases}$$

Ce qui donne bien :

$$TFD^{-1}[X_1(n) X_2(n)] = \sum_{p=0}^{N-1} x_1(p) x_2([k - p]_{\text{modulo } N}) = (x_1 \otimes x_2)(k)$$

Les figures 2.7 et 2.8 illustrent la différence entre un produit de convolution linéaire et un produit de convolution circulaire sur un exemple considérant deux signaux de $N = 3$ points : $x_1 = [1 \ 2 \ 3]$ et $x_2 = [1 \ 1 \ 1]$. Le produit de convolution linéaire considère que les signaux sont nuls en dehors de leurs N points (figure 2.7), tandis que le produit de convolution circulaire considère des signaux périodisés (figure 2.8). On constate que les résultats des ces deux produits de convolution sont différents : $[0 \ 1 \ 3 \ 6 \ 5 \ 3 \ 0]$ pour le produit de convolution linéaire et $[6 \ 6 \ 6 \ 6 \ 6 \ 6 \ 6]$ pour le produit de convolution circulaire. Néanmoins, il est possible de rendre le résultat du produit de convolution circulaire égal à celui du produit de convolution linéaire en prolongeant les signaux par un nombre de zéros au moins égal au nombre de point de signal (figure 2.9).



2.2.5 Egalité de Parseval

$$\sum_{k=0}^{N-1} |x(k)|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |X(n)|^2 \quad (2.18)$$

Démonstration :

$$\begin{aligned} \sum_{k=0}^{N-1} |x(k)|^2 &= \sum_{k=0}^{N-1} x(k)x^*(k) = \sum_{k=0}^{N-1} x(k) \frac{1}{N} \sum_{n=0}^{N-1} X^*(n) e^{-j2\pi \frac{kn}{N}} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} X^*(n) \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} X^*(n) X(n) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} |X(n)|^2 \end{aligned}$$

2.2.6 Algorithme de calcul rapide

La transformée de Fourier discrète se prête à un algorithme de calcul rapide que l'on nomme FFT pour "Fast Fourier Transform". Son principe, détaillé au paragraphe suivant, consiste à décomposer le signal de départ (sur N points) en une succession de sous suites entrelacées en effectuant à chaque étape de l'algorithme des transformées de Fourier disjointes sur les points d'indices pairs et les points d'indices impairs du tableau représentant le signal numérique. La condition de départ est que le nombre de points N soit une puissance de 2.

2.3 Algorithme de calcul rapide (FFT) de Cooley Tuckey

2.3.1 Principe

La transformée de Fourier de départ $X(n)$ porte sur les N points d'un signal numérique $x(k)$:

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}} = \sum_{k=0}^{N-1} x(k) W_N^{-kn}, \quad n = 0, \dots, N-1 \quad (2.19)$$

en notant $W_N = e^{j\frac{2\pi}{N}}$. On parle de TFD d'ordre N . Son temps de calcul est évalué à N^2 opérations d'additions/multiplications.

On suppose que N est une puissance de 2 : $N = 2^p$ et on va commencer à décomposer le signal en sous suites entrelacées.

Première décomposition

$$X(n) = X_1(n) + W_N^{-n} X_2(n), \quad n = 0, \dots, N-1, \quad (2.20)$$

où :

$$X_1(n) = \sum_{i=0}^{N/2-1} x(2i) W_{N/2}^{-in}, \quad X_2(n) = \sum_{i=0}^{N/2-1} x(2i+1) W_{N/2}^{-in}, \quad i = 0, \dots, N/2-1 \quad (2.21)$$

On peut évaluer le temps de calcul suite à cette première décomposition à $2\left(\frac{N}{2}\right)^2 + N$ opérations d'additions/multiplications, ce qui est déjà $\ll N^2$, surtout si N est grand.

Deuxième décomposition

$$X_1(n) = X_{11}(n) + W_{N/2}^{-n} X_{12}(n), \quad X_2(n) = X_{21}(n) + W_{N/2}^{-n} X_{22}(n), \quad n = 0, \dots, N/2 - 1, \quad (2.22)$$

où :

$$X_{11}(n) = \sum_{i=0}^{N/4-1} x_1(2i) W_{N/4}^{-in}, \quad X_{12}(n) = \sum_{i=0}^{N/4-1} x_1(2i+1) W_{N/4}^{-in}, \quad i = 0, \dots, N/4 - 1 \quad (2.23)$$

$$X_{12}(n) = \sum_{i=0}^{N/4-1} x_2(2i) W_{N/4}^{-in}, \quad X_{22}(n) = \sum_{i=0}^{N/4-1} x_2(2i+1) W_{N/4}^{-in}, \quad i = 0, \dots, N/4 - 1 \quad (2.24)$$

avec $x_1(p) = x(2i)$ et $x_2(p) = x(2i+1)$, $i, p = 0, \dots, N/2 - 1$.

...

Dernière décomposition

On continue jusqu'à arriver à la plus petite transformée de Fourier possible qui porte sur deux points et que l'on appelle le "papillon" de la transformée de Fourier numérique (figure 2.10). On aura alors réalisé Np transformées de Fourier d'ordre 2 ou "papillons", soit un temps de calcul de $N \log_2(N) \ll N^2$ opérations d'additions/multiplications.

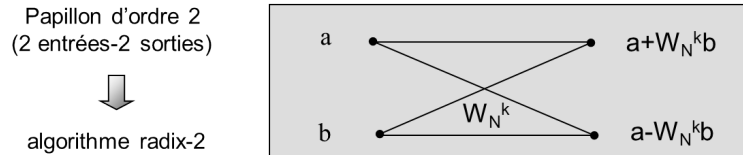


FIGURE 2.10 – Papillon de la FFT : transformée de Fourier d'ordre 2 = 2 opérations d'addition/multiplication

2.3.2 Exemple pour $N = 2^3 = 8$

Première décomposition

$$X(n) = X_1(n) + W_8^{-n} X_2(n), \quad n = 0, \dots, 7, \quad (2.25)$$

où $X_1(n)$ porte sur $x(0), x(2), x(4), x(6)$ et $X_2(n)$ porte sur $x(1), x(3), x(5), x(7)$.

Deuxième et dernière décomposition

$$X_1(n) = X_{11}(n) + W_4^{-n} X_{12}(n), \quad n = 0, \dots, 3, \quad (2.26)$$

où $X_{11}(n)$ porte sur $x(0), x(4)$ et $X_{12}(n)$ porte sur $x(2), x(6)$.

$$X_2(n) = X_{21}(n) + W_4^{-n} X_{22}(n), \quad n = 0, \dots, 3, \quad (2.27)$$

où $X_{21}(n)$ porte sur $x(1), x(5)$ et $X_{22}(n)$ porte sur $x(3), x(7)$.

Graphes correspondant

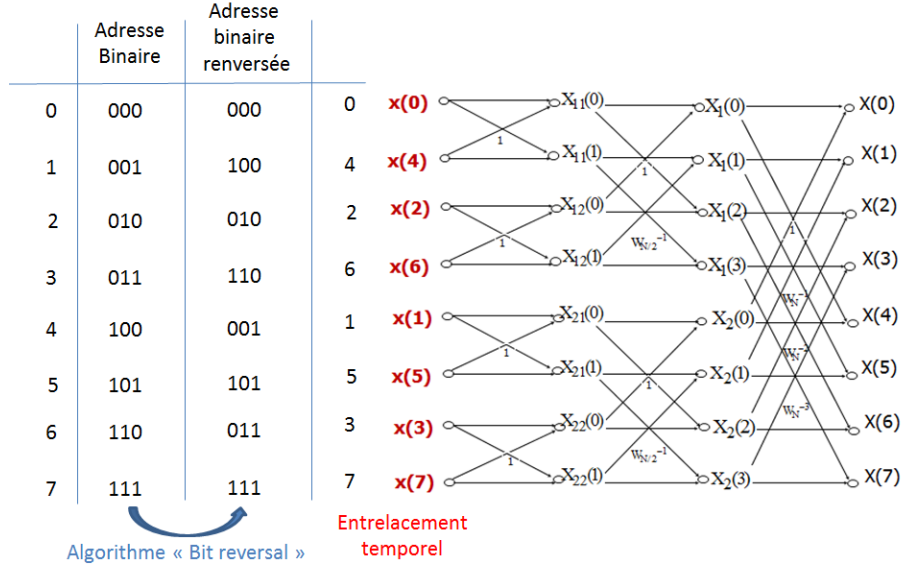


FIGURE 2.11 – Graphe de la FFT pour $N = 2^3$

On a $p = 3$ colonnes de $\frac{N}{2} = 4$ papillons, soit $p \times \left(\frac{N}{2}\right) \times 2 = pN = \log_2(N)N = 3 \times 8 = 24$ opérations d'additions multiplications, au lieu de $N^2 = 64$ si nous n'avions pas utilisé l'algorithme de FFT. On remarque que les échantillons de signal ne se présentent pas dans leur ordre naturel à l'entrée de l'algorithme. On peut utiliser un algorithme de renversement de l'adresse binaire ("bit reversal") afin de les présenter dans l'ordre voulu.

2.4 Exercices

2.4.1 Exercice 1 : Etude de la TFD d'un signal à spectre continu : effet de la limitation de la durée du signal

$$x(t) = \begin{cases} e^{-at} & \text{si } t \geq 0, a > 0 \\ 0 & \text{si } t < 0. \end{cases} \quad (2.28)$$

On observe le signal sur une durée limitée L .

1. Montrer que la transformée de Fourier du signal observé sur une durée $[0, L]$ s'écrit $X_L(f) = X(f)G(f, L)$

Calculons tout d'abord la transformée de Fourier du signal $x(t)$:

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{j2\pi f t} dt = \int_0^{+\infty} e^{-(a+j2\pi f)t} dt = \frac{1}{a + j2\pi f} \quad (2.29)$$

puis celle du signal observé sur une durée $[0, L]$:

$$X_L(f) = \int_0^L x(t)e^{j2\pi ft} dt = \int_0^L e^{(-a+j2\pi f)t} dt = X(f).G(f, L)$$

avec

$$G(f, L) = 1 - e^{-(a+j2\pi f)L} = 1 - e^{-aL} (\cos(2\pi fL) - j \sin(2\pi fL))$$

- Déterminer le module de $G(f, L)$.

$$|G(f, L)|^2 = (1 - e^{-aL} \cos(2\pi fL))^2 + e^{-2aL} \sin^2(2\pi fL) = 1 - 2e^{-aL} \cos(2\pi fL) + e^{-2aL}$$

- Montrer que $|G(f, L)|$ est compris entre $1 - e^{-aL}$ et $1 + e^{-aL}$.

$$(1 - e^{-aL}) \text{ (quand } \cos(2\pi fL) = +1) \leq |G(f, L)| \leq (1 + e^{-aL}) \text{ (quand } \cos(2\pi fL) = -1)$$

- Chiffrer ces bornes pour $L = \frac{4}{a}$.

Pour $L = \frac{4}{a}$ on a $e^{-aL} = e^{-4} = 0.0183$, d'où $0.9817 \leq |G(f, L)| \leq 1.0183$,

d'où $|G(f, L)| \simeq 1$, d'où $|X_L(f)| = |X(f)||G(f, L)| \simeq |X(f)|$

- Déterminer la phase de $G(f, L)$.

$$\text{Arg}[G(f, L)] = \text{Arctan} \left(\frac{e^{-aL} \sin(2\pi fL)}{1 - e^{-aL} \cos(2\pi fL)} \right)$$

- En utilisant les développements limités dans le cas où $L \gg \frac{1}{a}$, montrer qu'on peut arriver à la valeur approchée de la phase suivante :

$$\text{Arg}[G(f, L)] \simeq e^{-aL} \sin(2\pi fL)$$

Pour $L \ll \frac{1}{a}$ on a $e^{-aL} \ll 1$ et donc $\text{Arg}[G(f, L)] \simeq \text{Arctan}(e^{-aL} \sin(2\pi fL))$ car $e^{-aL} \cos(2\pi fL) \ll 1$. Puis $\text{Arg}[G(f, L)] \simeq e^{-aL} \sin(2\pi fL)$ car $\text{Arctan}(x) \simeq x$ quand $x \ll 1$.

- Borner la valeur approchée de la phase et la chiffrer pour $L = \frac{4}{a}$.

$|\text{Arg}[G(f, L)]| \leq e^{-aL} = 0.0183 \simeq 0$ et donc $\text{Arg}[X_L(f)] = \text{Arg}[X(f)] + \text{Arg}[G(f, L)] \simeq \text{Arg}[X(f)]$

- Quelle conclusion peut-on tirer de ces calculs sur l'effet de la troncature du signal $x(t)$?
On n'abime donc pas trop ni le module ni l'argument du spectre en observant le signal sur une durée suffisante (vrai pour $L = \frac{4}{a}$, alors $e^{-aL} \simeq 0.02$).

!!Attention!! En nommant $w_L(t)$ la fenêtre de troncature du signal à une durée L on a bien $TF[x(t)w_L(t)] = X(f) * W_L(f)$, où $W_L(f)$ représente la transformée de Fourier de la fenêtre de troncature (rectangulaire ici). On a juste montré que, dans ce cas, la dégradation du spectre due à la convolution par la transformée de Fourier de la fenêtre de troncature se ramène à une

erreur multiplicative que l'on a appelé $G(f, L)$, c'est-à-dire : $TF[x(t)w_L(t)] = X(f) * W(f) = X(f)G(f, L)$.

2.4.2 Exercice 2 : Etude de la TFD d'un signal à spectre continu : effet de l'échantillonnage du signal

Soit le signal $x(t)$ défini par :

$$x(t) = \begin{cases} e^{-at} & \text{si } t \geq 0, a > 0 \\ 0 & \text{si } t < 0. \end{cases} \quad (2.30)$$

1. Déterminer la transformée de Fourier $X(f)$ du signal $x(t)$. Tracer $|X(f)|$.
 $X(f)$ a été calculée précédemment : voir équation 2.29.
2. En théorie le signal $x(t)$ est-il échantillonnable sans perte d'information ? Expliquez votre réponse.
 $X(f)$ (voir équation 2.29) est à support non borné. En théorie $x(t)$ n'est donc pas échantillonnable sans perte d'information. Cependant $X(f)$ tend vers 0 quand $f \rightarrow \infty$. En pratique le signal $x(t)$ pourra donc être échantillonné avec une perte d'information "acceptable". Reste à définir un critère d'acceptabilité pour la perte d'information. Dans l'exercice on demande de considérer que $X(f)$ est nul pour une atténuation de plus de 40 dB par rapport à son maximum (voir question suivante).
3. En considérant la transformée de Fourier comme négligeable pour une atténuation minimale de 40 dB par rapport à sa valeur maximum, dimensionner la fréquence d'échantillonnage à utiliser F_e . On va considérer que $X(f)$ est nul pour une atténuation de plus de 40 dB par rapport à son maximum pour en déduire une fréquence maximale F_{max} et l'utiliser pour dimensionner la fréquence d'échantillonnage F_e : $20\log|X(F_{max})| = 20\log|X_{max}(f)| - 40dB = 20\log|X(0)| - 20\log 10^2 \Leftrightarrow |X(F_{max})| = \frac{|X(0)|}{10^2}$, ce qui conduit à $F_{max} \simeq \frac{100a}{2\pi}$ et donc $F_e > \frac{100a}{\pi}$.
4. Donner l'expression de la transformée de Fourier d'un signal $x(t)$ échantillonné à T_e , c'est-à-dire la transformée de Fourier de $\{x(kT_e)\}$ pour $k = -\infty, \dots, +\infty$. On la notera $X_e(f)$.

$$X_e(f) = \sum_{k=-\infty}^{+\infty} x(kT_e)e^{-j2\pi f k T_e} \quad (2.31)$$

5. Déterminer $X_e(f)$ pour le signal donné par (6.11). Vérifier qu'elle est périodique de période F_e . La comparer à $X(f)$.

$$X_e(f) = \sum_{k=0}^{+\infty} \left[e^{-(a+j2\pi f T_e)} \right]^k = \frac{1}{1 - e^{-(a+j2\pi f T_e)}}$$

$X(f)$ est bien périodique de période F_e : $X(f + pF_e) = X(f)$, $p \in \mathbb{Z}$.

Remarques :

- Pour $f \ll \frac{F_e}{2} \Rightarrow \frac{2f}{F_e} \ll 1$ on peut faire un développement limité de l'exponentielle (on a aussi $aT_e = \frac{100}{\pi} \ll 1$) qui donne $X_e(f) \sim \frac{1}{a+j2\pi f T_e} = F_e X(f)$. Ce résultat s'explique par le fait que l'influence de la périodisation d'ordre 1 devient importante quand f approche de $\frac{F_e}{2}$ mais est faible quand $f \ll \frac{F_e}{2}$. Le facteur F_e est dû au fait que l'on définit la TFD (équation (2.31)) à un facteur T_e près.

— on pensera à utiliser un filtre anti repliement avant d'échantillonner $x(t)$.

2.4.3 Exercice 3 : Etude de la TFD d'un signal à spectre continu : échantillonnage et limitation de la durée du signal

Soit le signal $x(t)$ défini par :

$$x(t) = \begin{cases} e^{-at} & \text{si } t \geq 0, a > 0 \\ 0 & \text{si } t < 0. \end{cases} \quad (2.32)$$

1. Donner l'expression de la transformée de Fourier d'un signal $x(t)$ échantillonné à T_e et limité à N points, c'est-à-dire la transformée de Fourier de $\{x(kT_e)\}$ pour $k = 0, \dots, N-1$. On la notera $X_D(f)$.

$$X_D(f) = \sum_{k=0}^{N-1} x(kT_e) e^{-j2\pi f k T_e} \quad (2.33)$$

2. Déterminer $X_D(f)$ pour le signal donné par (2.32). La comparer à $X(f)$.

$$X_D(f) = \sum_{k=0}^{N-1} \left[e^{-(a+j2\pi f T_e)} \right]^k = \frac{1 - e^{-(a+j2\pi f) N T_e}}{1 - e^{-(a+j2\pi f) T_e}}$$

Se combinent ici les deux approximations précédentes.

2.4.4 Exercice 4 : Etude de la TFD d'un signal à spectre discontinu : calcul d'un nombre fini de points du spectre

Soit le signal $x(t)$ défini par :

$$x(t) = A e^{j(2\pi f_0 t + \phi)}, \quad t \in \mathbb{R}, \quad \phi = \text{constante} \quad (2.34)$$

1. Déterminer la transformée de Fourier $X(f)$ du signal $x(t)$.
 $x(t) = A e^{j\phi} e^{j2\pi f_0 t} \rightarrow X(f) = A e^{j\phi} \delta(f - f_0)$
2. Déterminer la transformée de Fourier du signal observé sur une durée limitée $[0, L]$. On la note $X_L(f)$.

Pour obtenir $X_L(f)$, on peut procéder de deux manières :

$$\begin{aligned} X_L(f) &= \int_0^L x(t) e^{-j2\pi f t} dt = A e^{j\phi} \int_0^L e^{-j2\pi(f-f_0)t} dt \\ &\rightarrow A L e^{j\phi} \text{sinc}(\pi(f-f_0)L) e^{-j\pi(f-f_0)L} \end{aligned}$$

ou

$$\begin{aligned} X_L(f) &= TF \left[x(t) \Pi_L \left(t - \frac{L}{2} \right) \right] = X(f) * \{L \text{sinc}(\pi f L) e^{-j\pi f L}\} \\ &= A L e^{j\phi} \text{sinc}(\pi(f-f_0)L) e^{-j\pi(f-f_0)L} \end{aligned} \quad (2.35)$$

$\Pi_L(t - \frac{L}{2})$ représente une fenêtre rectangulaire (ou fonction porte) de largeur L centrée en $\frac{L}{2}$.

3. Déterminer la transformée de Fourier du signal échantillonné à T_e et observé sur N points. On la note $X_D(f)$.

$$\begin{aligned} X_D(f) &= Ae^{j\phi} \sum_{k=0}^{N-1} \left[e^{-j\pi(f-f_0)T_e} \right]^k = \frac{1 - e^{-j2\pi(f-f_0)NT_e}}{1 - e^{-j2\pi(f-f_0)T_e}} \\ &= Ae^{j\phi} e^{-j\pi(f-f_0)(N-1)T_e} \frac{\sin(\pi(f-f_0)NT_e)}{\sin(\pi(f-f_0)T_e)} \end{aligned} \quad (2.36)$$

4. La transformée de Fourier numérique (spectre du signal) ne sera calculée que pour un nombre fini, N , de points : $X_D(f) \rightarrow \{X_D(n \frac{F_e}{N})\}$ pour $n = 0, \dots, N-1$. Dans le cas où $f_0 = \frac{n_0}{N} F_e$, avec n_0 entier, déterminer $X_D(n)$ (notation pour $X_D(n \frac{F_e}{N})$) puis tracer $|X_D(n)|$ pour $n = 0, \dots, N-1$. Que constate-t-on ?

On calcule N points du spectre sur une période F_e , d'où le pas de calcul de $\frac{F_e}{N}$. La variable fréquentielle f devient donc $n \frac{F_e}{N}$, avec $n = 0, \dots, N-1$.

Dans le cas où $f_0 = n_0 \frac{F_e}{N}$, on a :

$$X_D(n) = Ae^{j\phi} e^{-j\pi \frac{(n-n_0)}{N} (N-1)} \frac{\sin(\pi(n-n_0))}{\sin(\pi \frac{(n-n_0)}{N})}$$

qui donne $X_D(n) = Ae^{j\phi} N$ pour $n = n_0$ et 0 ailleurs. La figure 2.12 donne un exemple de représentation de $X_D(n)$ dans le cas où $n_0 = 4$. On remarque que l'on retrouve la transformée de Fourier théorique d'une exponentielle de fréquence f_0 , soit un Dirac en f_0 .

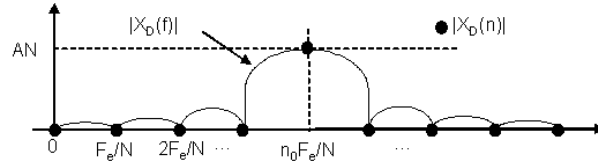


FIGURE 2.12 – Exemple de représentation de $X_D(n)$ dans le cas où $n_0 = 4$.

5. Tracer $|X_D(n)|$, pour $n = 0, \dots, N-1$, dans le cas où $f_0 = \frac{n_0 + \epsilon}{N} F_e$, n_0 entier et $0 < \epsilon < 1$. Ce résultat est-il satisfaisant (permet-il une analyse spectrale correcte) ?

Le cas où $f_0 = n_0 \frac{F_e}{N}$ est un cas très particulier. La plupart du temps on a $f_0 = (n_0 + \epsilon) \frac{F_e}{N}$. Un exemple de la transformée de Fourier numérique $X_D(n)$ correspondante est tracé sur la figure 2.13. A partir des points représentant la transformée de Fourier numérique du signal nous n'avons plus aucun moyen de décider qu'il s'agit du spectre d'une exponentielle.

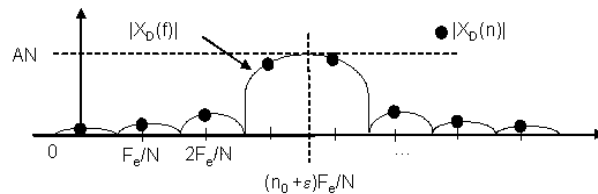


FIGURE 2.13 – Exemple de représentation de $X_D(n)$ dans le cas où $n_0 = 4 + \epsilon$.

6. Quelle méthode peut-on utiliser pour améliorer la visualisation de la transformée de Fourier numérique (et donc le résultat de l'analyse spectrale) ?

La solution permettant de mieux visualiser le spectre est d'interpoler. On le fera grâce à la technique du Zero Padding.

Chapitre 3

Fonctions d'inter et d'auto correlations

Les fonctions d'inter et d'auto corrélations peuvent être obtenues en numérique dans le domaine temporel ou dans le domaine fréquentiel.

3.1 Calcul dans le domaine temporel

3.1.1 Signaux déterministes

Le $k^{\text{ième}}$ échantillon de la fonction d'intercorrélation numérique entre les signaux x et y s'écrit :

- $R_{xy}(k) = \sum_{n=-\infty}^{+\infty} x(n)y^*(n-k)$, pour un signal déterministe à énergie finie.
- $R_{xy}(k) = \frac{1}{N_0} \sum_{n=0}^{N_0-1} x(n)y^*(n-k)$ pour un signal déterministe à puissance moyenne finie périodique de période N_0 .

3.1.2 Signaux aléatoires

Estimateur biaisé

En supposant le signal ergodique et en utilisant un estimateur de la moyenne, le $k^{\text{ième}}$ élément de la fonction d'intercorrélation numérique entre les signaux aléatoires x et y aléatoires s'écrit naturellement comme :

$$\hat{R}_{xy}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)y^*(n-k) \quad (3.1)$$

si N représente le nombre de points des signaux considérés. En réalité, si on considère les signaux numériques comme des tableaux de N points ($[x(0) \dots x(N-1)]$ et $[y(0) \dots y(N-1)]$), la somme précédente porte sur $N-k$ échantillons du produit $x(n)y^*(n-k)$ (signaux décalés, produit non nul uniquement entre k et $N-1$). L'estimateur précédent est donc, en réalité, donné par :

$$\hat{R}_{xy}(k) = \frac{1}{N} \sum_{n=k}^{N-1} x(n)y^*(n-k) \quad (3.2)$$

et il est biaisé :

$$E \left[\hat{R}_{xy}(k) \right] = \frac{N-|k|}{N} R_{xy}(k). \quad (3.3)$$

Le biais est ici multiplicatif et triangulaire (voir figure 3.1 dans les exemples).

Estimateur non biaisé

Afin de supprimer le biais, on peut définir un deuxième estimateur temporel de la manière suivante :

$$\hat{R}_{xy}(k) = \frac{1}{N-k} \sum_{n=0}^{N-1} x(n)y^*(n-k) \quad 0 \leq k \leq N-1 \quad (3.4)$$

On a bien alors $E[\hat{R}_{xy}(k)] = R_{xy}(k)$. Cependant cet estimateur possède une grande variance sur les bords. En effet, lorsque $k \rightarrow N$ peu de points vont être utilisés pour réaliser l'estimation. Celle-ci variera donc beaucoup entre différentes réalisations de signal (voir figure 3.2 dans les exemples). Ceci reste vrai avec l'estimateur biaisé mais le phénomène est masqué par le fait que le biais soit triangulaire.

Exemples : cosinus, ligne d'image SAR (Synthese Aperture Radar)

La figure 3.1 trace une estimation biaisée de la fonction d'autocorrélation ($x = y$) d'un cosinus, obtenue grâce à la fonction `xcorr.m` de matlab. On voit apparaître le biais multiplicatif triangulaire. La figure 3.2 trace une estimation non biaisée de la fonction d'autocorrélation d'un cosinus, obtenue grâce à la fonction `xcorr.m` de matlab, avec le paramètre `'unbiased'`. On voit apparaître la variance de l'estimation sur les bords de la fenêtre d'analyse.

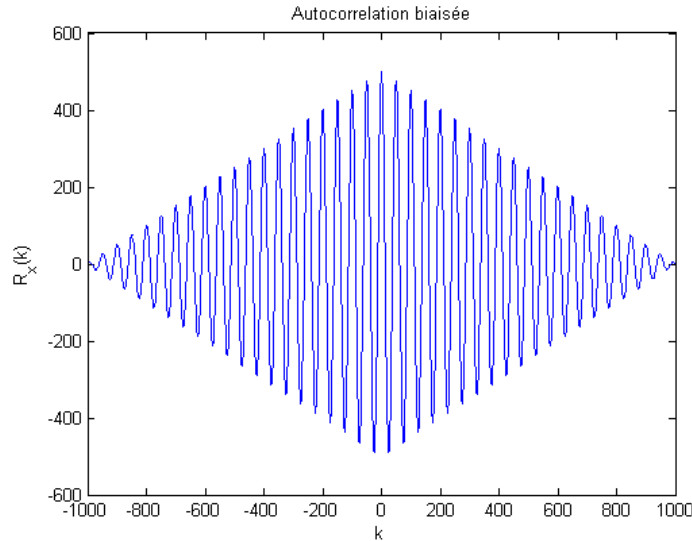


FIGURE 3.1 – Estimation biaisée de la fonction d'autocorrélation du cosinus

La figure 3.3 compare les estimations biaisée et non biaisée de la fonction de covariance (*autocorrelation – moyenne*²) d'un signal réel (ligne d'image SAR (Synthese Aperture Radar)) à la covariance théorique. On constate que la covariance estimée avec un estimateur biaisé semble donner un résultat plus proche de la covariance théorique que la covariance estimée avec un estimateur non biaisé. Cette constatation reste généralement vraie pour les signaux réels. Notons que pour une comparaison valable des estimateurs biaisés et non biaisés, il faudrait comparer leur erreur quadratique moyenne. Il a été montré que, dans de nombreux cas en effet, l'erreur quadratique moyenne de l'estimateur biaisé est nettement inférieure à celle de l'estimateur non biaisé. Dans les deux cas cependant, on peut améliorer les estimations en augmentant la durée d'observation. L'estimateur biaisé est, en effet, asymptotiquement non biaisé, tandis que la variance de l'estimateur non biaisé apparaît sur les bords.

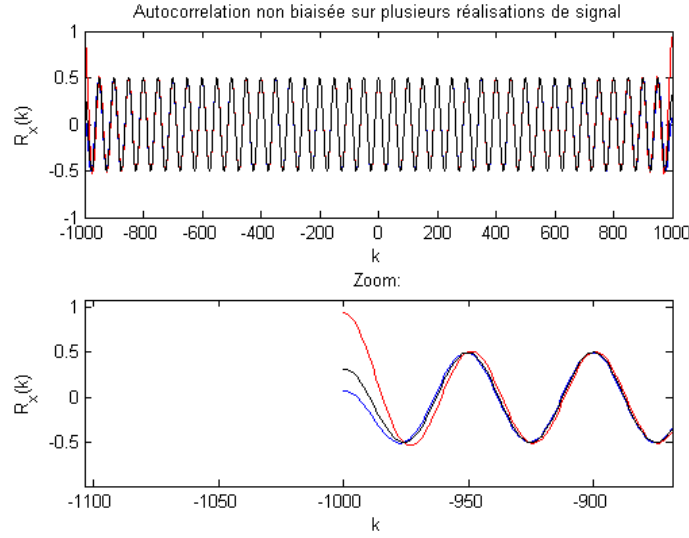


FIGURE 3.2 – Estimations non biaisées de la fonction d'autocorrelation du cosinus

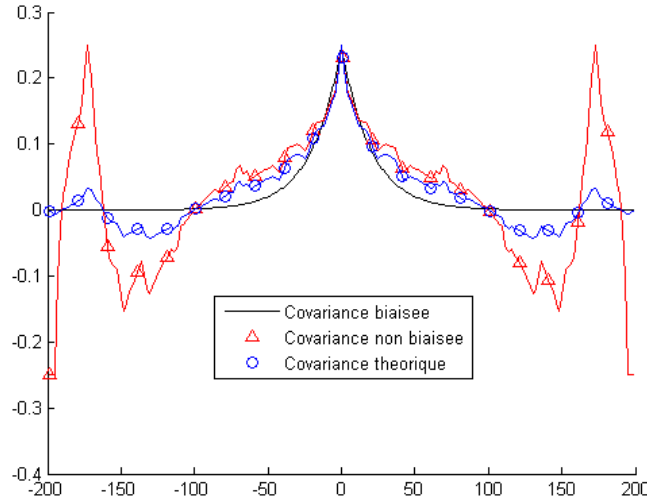


FIGURE 3.3 – Estimations de la covariance sur une ligne d'image SAR

3.2 Calcul dans le domaine fréquentiel

En partant des expressions $R_{xy}(k) = \sum_{n=-\infty}^{+\infty} x(n)y^*(n-k)$ ou $\hat{R}_{xy}(k) = \frac{1}{N} \sum_{n=k}^{N-1} x(n)y^*(n-k)$, on remarque qu'il est possible d'écrire la fonction d'intercorrélation comme un produit de convolution :

$$\hat{R}_{xy}(k) = x(k) * y^*(-k) \quad (3.5)$$

pour des signaux représentés par des tableaux de N valeurs. On peut alors estimer la fonction d'intercorrélation dans le domaine fréquentiel de la manière suivante :

$$\hat{R}_{xy}(k) = TFD^{-1}[X(n)Y^*(n)] \quad (3.6)$$

où $X(n) = TFD[x(k)]$ et $Y(n) = TFD[y(k)]$. L'intérêt de cette estimation fréquentielle est qu'elle permet de réduire le temps de calcul. En effet, en utilisant l'algorithme de la FFT (voir paragraphe 2.3), le temps de calcul pour obtenir N éléments de \hat{R}_{xy} est donné par $\sim 3N \log_2(N) +$

N opérations d'addition/multiplication (2 FFT, 1 FFT inverse et N produits) si on fait le calcul avec un estimateur fréquentiel, au lieu de $\frac{N^2}{2}$ en utilisant une estimation dans le domaine temporel ($N + (N-1) + \dots + 1 = \frac{N(N+1)}{2} \sim \frac{N^2}{2}$: N pour $k = 0$, $N-1$ pour $k = 1$, ... 1 pour $k = N-1$). Et $3N \log_2(N) + N \ll \frac{N^2}{2}$, surtout quand N est grand. Cependant ce calcul, utilisant un passage dans le domaine fréquentiel, suppose que la transformée de Fourier discrète inverse transforme un produit en produit de convolution linéaire. Or nous avons vu que la transformée de Fourier discrète (ou TFD inverse) transformait un produit en produit de convolution circulaire, c'est-à-dire un produit de convolution entre signaux périodisés (période N). Néanmoins nous avons également vu plus tôt qu'il est possible de se ramener à un produit de convolution linéaire en utilisant du Zero Padding (au moins autant de zéros ajoutés que de longueur de signal avant de passer dans le domaine transformé).

3.3 Quelques propriétés

3.3.1 Symétrie Hermitienne

Comme c'est le cas à temps continu, la fonction d'intercorrélation numérique possède la propriété de symétrie hermitienne :

$$\hat{R}_{xy}(-k) = \hat{R}_{xy}^*(k) \quad (3.7)$$

Soit pour des signaux réels : $\hat{R}_{xy}(-k) = \hat{R}_{xy}(k)$. Cette propriété est intéressante car elle permet d'économiser la moitié du temps de calcul.

3.3.2 Bornes

La fonction d'autocorrélation numérique est bornée par sa valeur en 0 qui représente la puissance du signal :

$$\hat{R}_x(k) \leq \hat{R}_x(0) = P_x \quad (3.8)$$

$P_x = R_x(0)$ représentant la puissance du signal x . Mais aussi :

$$\hat{R}_{xy}(k) \leq \frac{1}{2} \left(\hat{R}_x(0) + \hat{R}_y(0) \right) \quad (3.9)$$

Chapitre 4

Densité Spectrale de Puissance (DSP)

La Densité Spectrale de Puissance reflète la contribution de chaque fréquence à la puissance moyenne du signal. Elle permet de fournir une représentation fréquentielle aux signaux aléatoires pour lesquels nous ne pouvons calculer une transformée de Fourier (mais reste utilisable pour les signaux déterministes). Elle est donnée par la transformée de Fourier de la fonction d'autocorrélation du signal :

$$S_x(f) = TF[R_x(\tau)] \quad (4.1)$$

Il est donc naturel d'envisager d'estimer en numérique la densité spectrale de puissance d'un signal aléatoire en prenant la TFD de l'estimation de sa fonction d'autocorrélation. On appellera cet estimateur l'estimateur "de base". Malheureusement cet estimateur "de base" s'avère ne pas être consistant : il est biaisé et sa variance ne tend pas vers 0 lorsque l'on augmente la durée d'observation pour réaliser l'estimation. On définit donc des estimateurs de la DSP plus satisfaisants, à partir de cet estimateur "de base".

4.1 Estimateurs "de base" : corrélogramme, périodogramme

4.1.1 Définitions

La manière la plus naturelle d'estimer la densité spectrale de puissance du signal x est d'utiliser sa définition :

$$\hat{S}_x(n) = TFD \left[\hat{R}_x(k) \right] = \sum_{k=0}^{N-1} \hat{R}_x(k) e^{-j2\pi \frac{kn}{N}}, \quad n = 0, \dots, N-1 \quad (4.2)$$

en utilisant l'estimée de la fonction d'autocorrélation et la transformée de Fourier discrète. On appelle cet estimateur le corrélogramme, biaisé ou non biaisé selon que l'on utilise un estimateur biaisé ou non biaisé pour l'estimation de la fonction d'autocorrélation. Le corrélogramme utilisant un estimateur biaisé pour la fonction d'autocorrélation est également appelé périodogramme et est donné par :

$$\hat{S}_x(n) = \frac{1}{N} |X(n)|^2 \left(= \frac{1}{N} TFD [x(k) * x^*(-k)] \right) \quad (4.3)$$

en notant $X(n)$ la TFD d'une réalisation du signal x (tableau de N points). Notons que cette définition de la DSP par périodogramme permet un lien plus direct avec le cadre des signaux déterministes : la densité spectrale de puissance ou d'énergie est en effet proportionnelle au module de la transformée de Fourier du signal au carré pour les signaux déterministes.

4.1.2 Problèmes posés par ces estimateurs "de base"

On constate qu'en moyenne les estimateurs de la fonction d'autocorrélation se comportent comme la fonction d'autocorrélation théorique multipliée par une fenêtre (rectangulaire pour l'estimateur non biaisé, triangulaire pour l'estimateur biaisé) :

$$\begin{aligned} E \left[\widehat{R}_x(k) \right] &= \Pi_N(k) R_x(k) \quad \text{dans le cas d'un estimateur non biaisé.} \\ &= \frac{N - |k|}{N} R_x(k) \quad \text{dans le cas d'un estimateur biaisé.} \end{aligned}$$

Ce qui donne pour l'estimée de la densité spectrale de puissance :

$$E \left[\widehat{S}_x(n) \right] = S_x(n) * W(n)$$

où $W(n)$ représente la transformée de Fourier de la fenêtre. Les estimateurs de base de la DSP possèdent donc un biais convolutif avec :

- $W(n) = \frac{\sin(\pi n)}{\sin(\frac{\pi n}{N})}$ dans le cas de l'utilisation d'un estimateur non biaisé pour la fonction d'autocorrélation.
- $W(n) = \left(\frac{\sin(\pi n)}{\sin(\frac{\pi n}{N})} \right)^2$ (noyau de Fejer) dans le cas de l'utilisation d'un estimateur biaisé pour la fonction d'autocorrélation.

et leur variance ne tend pas vers 0 quand N tend vers l'infini. Il est difficile d'établir la variance de ces estimateurs pour un signal de distribution quelconque. Cette variance est donc calculée dans le cas particulier du bruit blanc et généralisée par approximation aux autres signaux (voir ref [2], page 258). Ces estimateurs de la DSP (corrélogramme, périodogramme) ne sont donc pas consistants. De plus, l'estimateur utilisant une estimation non biaisée pour la fonction d'autocorrélation peut donner des valeurs négatives pour la DSP.

4.2 Périodogramme avec fenêtrage

Comme nous l'avons évoqué dans le chapitre sur la transformée de Fourier discrète en parlant de TFD pondérée, on peut également, pour estimer la DSP, utiliser un périodogramme sur le signal pondéré :

$$\widehat{S}_x(n) = \frac{1}{N} |TFD[x(k)w(k)]|^2 \quad (4.4)$$

où $w(k)$ représente la fenêtre de pondération choisie (voir chapitre sur la transformée de Fourier).

4.3 Périodogramme cumulé (ou de Bartlett)

Cette méthode tire son nom du statisticien anglais M. S. Bartlett, qui a été le premier à la proposer. Pour calculer un périodogramme cumulé, le signal de N points x est divisé en M parties : x_i , $i = 1, \dots, M$ de $L = \frac{N}{M}$ points. Un périodogramme moyenné est alors effectué :

$$\widehat{S}_x(n) = \frac{1}{LM} \sum_{i=1}^M |TFD[x_i(k)]|^2, \quad (4.5)$$

afin de diminuer la variance d'estimation de la DSP. La variance sera diminuée d'un facteur M lorsque l'on moyenne M périodogrammes indépendants.

L'inconvénient de cet estimateur va être l'augmentation du biais car le lobe central du noyau

de Fejer est alors plus large ($2/L$ au lieu de $2/N$). De plus, on diminue la résolution du spectre obtenu (L points calculés sur F_e au lieu de N).

4.4 Périodogramme de Welch

Cette méthode a été proposée par Peter D. Welch en 1967. Le principe est le même que pour le périodogramme cumulé mais les différentes parties x_i sont autorisées à se recouvrir, ce qui réduit encore la variance d'estimation de la DSP pour une même résolution, et le signal est fenêtré avant calcul de la TFD. Typiquement on laisse les fenêtres se recouvrir sur la moitié de leur longueur et la fenêtre de troncature par défaut est celle de Hamming.

4.5 Exemple sur une ligne d'image SAR

La figure 4.1 trace la DSP théorique d'une ligne d'image SAR (Synthese Aperture Radar) et la compare aux estimations par périodogramme, par corrélogramme biaisé et corrélogramme non biaisé. On constate effectivement que le périodogramme et le corrélogramme biaisé sont identiques et plus proches de la DSP théorique que le corrélogramme non biaisé.

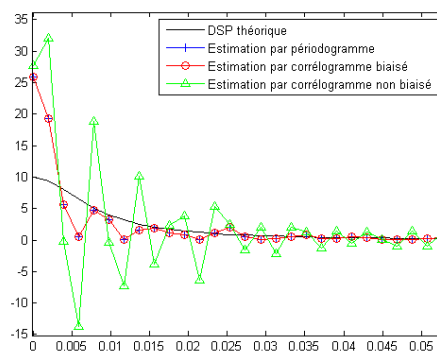


FIGURE 4.1 – Calcul de la DSP sur une ligne d'image SAR

La figure 4.2 trace la DSP théorique d'une ligne d'image SAR et son estimation par périodogramme cumulé. On peut effectivement constater que la variance d'estimation a réduit du fait du cumul.

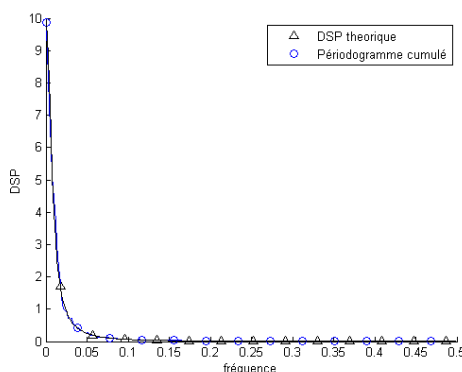


FIGURE 4.2 – Calcul de la DSP sur une ligne d'image SAR en utilisant un périodogramme cumulé

Chapitre 5

La transformée en z

Tout comme la transformée de Laplace permet l'étude des systèmes analogiques linéaires et invariants dans le temps, la transformée en z va permettre l'étude des systèmes numériques linéaires invariants dans le temps, et donc l'étude des filtres numériques linéaires invariants dans le temps qui feront l'objet du prochain chapitre. Ce chapitre permet de donner les éléments essentiels concernant la transformée en z afin de l'utiliser comme un outil d'étude. Pour plus de détails, voir, par exemple, la référence [5].

5.1 Définition

$$X(z) = TZ[x(n)] = \sum_{n=-\infty}^{+\infty} x(n)z^{-n}, z \in \mathbb{C} \quad (5.1)$$

5.2 Convergence

La région de convergence est l'ensemble des nombres complexes z tels que la série $X(z)$ converge. On utilisera le critère de Cauchy :

$$\lim_{n \rightarrow \infty} \sqrt[n]{|v(n)|} < 1 \quad \Rightarrow \quad \sum_{n=0}^{\infty} v(n) \quad \text{converge} \quad (5.2)$$

pour avoir une condition suffisante de convergence :

$$X(z) = \sum_{n=0}^{+\infty} x(n)z^{-n} + \sum_{n=1}^{+\infty} x(-n)z^n \text{ converge pour } 0 \leq R_x^- \leq |z| < R_x^+ < \infty \quad (5.3)$$

avec $R_x^- = \lim_{n \rightarrow \infty} \sqrt[n]{|x(n)|}$ et $R_x^+ = \frac{1}{\lim_{n \rightarrow \infty} \sqrt[n]{|x(-n)|}}$

Exemple :

$$X(z) = \sum_{n=0}^{+\infty} z^{-n} \quad \text{converge pour } |z| > 1 : R_x^- = 1 \text{ and } R_x^+ = \infty \quad (5.4)$$

5.3 Propriétés

5.3.1 Linéarité

$$TZ [ax(n) + by(n)] = aTZ [x(n)] + bTZ [y(n)] \quad (5.5)$$

Convergence : si $R^+ = \min(R_x^+, R_y^+)$ et $R^- = \max(R_x^-, R_y^-)$ alors le domaine de convergence contient $]R^-, R^+[$.

5.3.2 Décalage temporel

$$TZ [x(n - n_0)] = z^{-n_0} TZ [x(n)] \quad (5.6)$$

Même domaine de convergence que pour $X(z) = TZ [x(n)]$.

5.3.3 Changement d'échelle

$$TZ [a^n x(n)] = X\left(\frac{z}{a}\right) \quad (5.7)$$

Convergence : $aR_x^- \leq |z| < aR_x^+$

5.3.4 Dérivabilité

La transformée en z définit une série de Laurent qui est indéfiniment dérivable terme à terme dans son domaine de convergence. On en déduit :

$$TZ [nx(n)] = -z \frac{dX(z)}{dz} \quad (5.8)$$

Même domaine de convergence que pour $X(z) = TZ [x(n)]$.

5.3.5 Produit de convolution

Le produit de convolution entre les suites $x(n)$ et $y(n)$ est défini par :

$$x(n) * y(n) = \sum_{k=-\infty}^{+\infty} x(k)y(n-k) \quad (5.9)$$

On a alors :

$$TZ [x(n) * y(n)] = X(z)Y(z) \quad (5.10)$$

et sa région de convergence peut être plus large que l'intersection des régions de convergence de $X(z)$ et $Y(z)$.

5.4 Transformée en z inverse

5.4.1 Définition

La transformée en z inverse est définie par :

$$x(n) = \frac{1}{j2\pi} \int_{C^+} X(z)z^{n-1}dz \quad (5.11)$$

où C^+ est un contour fermé incluant l'anneau de convergence.

5.4.2 Preuve

L'expression de la transformée en z inverse découle directement du calcul de l'intégrale :

$$J(n, k) = \int_{C^+} z^{n-k-1} dz \quad (5.12)$$

A l'aide du théorème des résidus on montre : $J(n, k) = \begin{cases} 0, & n \neq k; \\ j2\pi, & n = k. \end{cases}$

On en déduit alors :

$$\begin{aligned} \frac{1}{j2\pi} \int_{C^+} X(z) z^{n-1} dz &= \frac{1}{j2\pi} \int_{C^+} \left(\sum_{k=-\infty}^{+\infty} x(k) z^{-k} \right) z^{n-1} dz \\ &= \frac{1}{j2\pi} \left(\sum_{k=-\infty}^{+\infty} x(k) \right) J(n, k) = x(n) \end{aligned} \quad (5.13)$$

Remarque : il existe des tables de transformées en z et transformées en z inverse.

Rappel : Si z_i est un pôle simple de $g(z)$:

$$\text{Residu}[g(z)]_{z=z_i} = \lim_{z \rightarrow z_i} (z - z_i) g(z) \quad (5.14)$$

Si z_i est un pôle d'ordre α de $g(z)$:

$$\text{Residu}[g(z)]_{z=z_i} = \frac{1}{(\alpha-1)!} \left[\frac{\partial^{\alpha-1}}{\partial z^{\alpha-1}} (g(z)(z - z_i)^\alpha) \right]_{z=z_i} \quad (5.15)$$

5.5 Exercices

5.5.1 Exercice 1

Soit un réel $a \in]0, 1[$ et $u(n)$ l'échelon de Heaviside (ou échelon unité) :

$$u(n) = \begin{cases} 1 & \text{pour } n > 0 \\ 0 & \text{pour } n \leq 0. \end{cases} \quad (5.16)$$

1. Déterminer la transformée en z du signal $x(n) = a^n u(n)$, avec $|a| < 1$, et préciser avec soin la région de convergence de $X(z)$.

$$x(n) = a^n u(n), |a| < 1 \rightarrow X(z) = \sum_{n=0}^{+\infty} [az^{-1}]^n = \frac{1}{1 - az^{-1}}$$

Domaine d'existence :

$$\lim_{n \rightarrow +\infty} \sqrt[n]{|a^n z^{-n}|} < 1 \text{ pour } |z| > |a|$$

2. Déterminer la transformée en z du signal $y(n) = -a^n u(-n-1)$, avec $|a| < 1$, et préciser avec soin la région de convergence de $Y(z)$.

$$y(n) = -a^n u(-n-1), |a| < 1$$

$$\rightarrow Y(z) = \sum_{n=-\infty}^{-1} -[az^{-1}]^n = \sum_{n=1}^{+\infty} -[a^{-1}z]^n = -\left\{ \sum_{n=0}^{+\infty} [a^{-1}z]^n - 1 \right\} = \frac{1}{1 - az^{-1}}$$

Domaine d'existence :

$$\lim_{n \rightarrow +\infty} \sqrt[n]{|a^{-n}z^n|} < 1 \text{ pour } |z| < |a|$$

Conclusion : la transformée en z inverse n'est pas unique. Son expression dépend du contour choisi pour la calculer. S'il inclut tous les points singuliers (premier cas dans cet exercice) on obtient une solution causale, sinon elle ne l'est pas (deuxième cas dans cet exercice).

3. Soit b un réel tel que $b > a$ et $|b| < 1$. On considère un système de fonction de transfert :

$$H(z) = \frac{1}{(1 - az^{-1})(1 - bz^{-1})} \quad (5.17)$$

Déterminer la réponse impulsionnelle $h(n)$ du système dans les trois cas suivants :

- la région de convergence de $H(z)$ est $|z| < a$,
- la région de convergence de $H(z)$ est $a < |z| < b$,
- la région de convergence de $H(z)$ est $|z| > b$.

Plusieurs méthodes sont possibles pour résoudre cette question. Par exemple, par décomposition en éléments simples, on obtient :

$$H(z) = \frac{1}{a-b} \left\{ \frac{a}{1 - az^{-1}} - \frac{b}{1 - bz^{-1}} \right\}$$

- pour $|z| < a$:

$$h(n) = \frac{1}{a-b} \{ -a^{n+1}u(-n-1) + b^{n+1}u(-n-1) \}.$$

Aucun des points singuliers n'est inclut dans le contour considéré. $h(n)$ est non causale.

- pour $a < |z| < b$:

$$h(n) = \frac{1}{a-b} \{ a^{n+1}u(n) + b^{n+1}u(-n-1) \}.$$

Le contour considéré inclut le point singulier $z = a$ et exclut $z = b$. $h(n)$ est constituée d'une partie causale et d'une partie non causale.

- pour $|z| > b$:

$$h(n) = \frac{1}{a-b} \{ a^{n+1} - b^{n+1} \} u(n).$$

Le contour considéré inclut tous les points singuliers ($z = a$ et $z = b$). La transformée en z inverse est causale.

5.5.2 Exercice 2

Soit le système d'entrée $x(n)$ et de sortie $y(n)$ défini par l'équation récurrente suivante : $y(n) - ay(n-1) = x(n)$, avec $|a| < 1$.

1. Soit $x(n) = b^n u(n)$ avec $|b| < 1$. Déterminer sa transformée en z , ainsi que son domaine d'existence.

$$x(n) = b^n u(n), |b| < 1 \rightarrow X(z) = \sum_{n=0}^{+\infty} [bz^{-1}]^n = \frac{1}{1 - bz^{-1}}$$

Domaine d'existence :

$$\lim_{n \rightarrow +\infty} \sqrt[n]{|b^n z^{-n}|} < 1 \text{ pour } |z| > |b|$$

2. Déterminer la réponse du système à l'entrée $x(n)$ définie à la question précédente, en supposant que le système est causal.

$$\begin{aligned} y(n) - ay(n-1) &= b^n u(n) \rightarrow Y(z) - az^{-1}Y(z) = \frac{1}{1 - bz^{-1}} \\ \rightarrow Y(z) &= \left\{ \frac{1}{1 - az^{-1}} \right\} \left\{ \frac{1}{1 - bz^{-1}} \right\} \end{aligned}$$

Soit en considérant un système causal :

$$y(n) = \frac{1}{a-b} \{a^{n+1} - b^{n+1}\} u(n)$$

3. Déterminer la fonction de transfert, ainsi que la réponse impulsionnelle du système.

$$y(n) - ay(n-1) = x(n) \rightarrow Y(z) - az^{-1}Y(z) = X(z) \rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - az^{-1}}$$

$$h(n) = TZ^{-1}[H(z)] = a^n u(n) \text{ (filtre causal).}$$

Chapitre 6

Filtrage numérique

On va s'intéresser aux filtres numériques linéaires, invariants dans le temps (LIT), rationnels.

6.1 Un outil d'étude : la transformée en z

Tout comme la transformée de Laplace permet l'étude des filtres analogiques de type LIT, la transformée en z va permettre l'étude des filtres numériques LIT. Tout comme, la transformée de Laplace permet de réaliser à la fois une étude temporelle et une étude fréquentielle (quand $p = j\omega$) des filtres analogiques LIT, la transformée en z va permettre à la fois de réaliser à la fois une étude temporelle et une étude fréquentielle (quand $z = e^{j\omega} = e^{j2\pi f}$) des filtres numériques LIT. Les principaux éléments concernant la transformée en z sont donnés dans le chapitre précédent.

6.2 Définitions

6.2.1 Linéarité

Si $y_1(n)$ et $y_2(n)$ sont les sorties du filtre qui correspondent respectivement aux entrées $x_1(n)$ et $x_2(n)$, le filtre est linéaire s'il fait correspondre $y_1(n) + y_2(n)$ à une entrée $x_1(n) + x_2(n)$.

6.2.2 Invariance temporelle

Si $y(n)$ est la sortie du filtre qui correspond à l'entrée $x(n)$, le filtre est dit invariant dans le temps s'il fait correspondre $y(n - n_0)$ à l'entrée $x(n - n_0)$, n_0 représentant un retard de n_0 échantillons.

6.2.3 Réponse impulsionnelle et fonction de transfert

On peut écrire tout signal numérique de la manière suivante : $x(n) = \sum_{k=-\infty}^{+\infty} x(k)\delta(n - k)$, où $\delta(n)$ représente le Dirac numérique qui vaut 1 pour $n = 0$ et 0 ailleurs. Si l'on met ce signal $x(n)$ à l'entrée d'un filtre numérique linéaire invariant dans le temps, que l'on nomme $y(n)$ la sortie correspondante (réponse) et $h(n)$ la sortie correspondant à l'entrée $\delta(n)$, on obtient alors :

$$y(n) = \sum_{k=-\infty}^{+\infty} h(k)x(n - k) = \sum_{k=-\infty}^{+\infty} h(n - k)x(k) = x(n) * h(n)$$

On peut donc écrire toute sortie $y(n)$ d'un filtre numérique linéaire invariant dans le temps en fonction de l'entrée appliquée $x(n)$ et de $h(n)$ qui est appelée réponse impulsionnelle du filtre (réponse à une impulsion représenté par $\delta(n)$). Cette réponse impulsionnelle $h(n)$ définit donc le

filtre. Sa transformée en z , $H(z)$, est appelée fonction de transfert du filtre. Si $Y(z)$ et $X(z)$ sont respectivement les transformées en z de $y(n)$ et $x(n)$, on a alors :

$$H(z) = \frac{Y(z)}{X(z)}$$

6.2.4 Filtres numériques rationnels

Les filtres numériques rationnels sont définis par une fonction de transfert rationnelle en z (par analogie avec les filtres analogiques qui, eux, sont définis par une fonction de transfert rationnelle en p) :

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}} \quad (6.1)$$

Un filtre numérique rationnel sera donc défini par deux vecteurs de coefficients : $A = [a_0 \ a_1 \dots \ a_{M-1}]$ et $B = [b_0 \ b_1 \dots \ b_{N-1}]$ stockés en mémoire. On pourra, par exemple, réaliser le filtrage sous Matlab en utilisant la fonction *filter.m* de la manière suivante : $y = \text{filter}(B, A, x)$, si x représente le signal à filtrer, y le signal filtré et B et A les tableaux de coefficients définissant le filtre à utiliser. Nous reviendrons par la suite plus en détail sur cette catégorie de filtres.

6.2.5 Réponse en fréquence et temps de propagation de groupe (TPG)

La réponse en fréquence, ou réponse harmonique, du filtre est donnée par :

$$H(\tilde{f}) = [H(z)]_{z=e^{j2\pi\tilde{f}}}$$

où $\tilde{f} = \frac{f}{F_e}$ (fréquence normalisée), avec F_e qui représente la fréquence d'échantillonnage. On peut également écrire : $|H(\tilde{f})|^2 = [H(z)H(\frac{1}{z})]_{z=e^{j2\pi\tilde{f}}}$, ce qui peut faciliter certains calculs (voir exercices).

Le temps de propagation de groupe (TPG) est donné en fréquences normalisées par :

$$TPG(\tilde{f}) = -\frac{1}{2\pi} \frac{d\varphi_H(\tilde{f})}{d\tilde{f}}, \text{ avec } \varphi_H(\tilde{f}) = \text{Arg} [H(\tilde{f})]$$

$TPG(\tilde{f})$ représente le temps que met un paquet d'onde à la fréquence \tilde{f} pour traverser le filtre. Lorsque l'on souhaite réaliser un filtre numérique les spécifications portent généralement sur le module de sa réponse en fréquence et on souhaite un TPG le plus constant possible, au moins sur la/les bande(s) passante(s). Notons cependant qu'il existe des filtres de phase, pour lesquels les spécifications portent sur la phase de la réponse en fréquence (afin de corriger, par exemple, un TPG non constant).

6.3 Réalisabilité d'un filtre numérique

Un filtre numérique est réalisable si les trois conditions suivantes sont respectées :

- Le filtre doit être causal. Cette condition se décline sur la réponse impulsionnelle du filtre de la manière suivante : $h(n) = 0$ pour $n < 0$.
- Le filtre doit être stable : pour toute entrée bornée, la sortie doit être bornée. Cette condition se décline sur la réponse impulsionnelle du filtre de la manière suivante : $\sum_{n=-\infty}^{+\infty} |h(n)| < \infty$. Nous verrons par la suite, pour les filtres rationnels, que cette

condition conduit à une condition portant sur les pôles de la fonction de transfert du filtre.

— La réponse impulsionnelle du filtre $h(n)$ doit être réelle.

6.4 Classification des filtres numériques rationnels : RIF, RII

Les filtres numériques rationnels sont donc définis par une fonction de transfert rationnelle en z de la manière suivante :

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}} \quad (6.2)$$

Cette définition conduit dans le domaine temporel à une équation récurrente (là où nous avons une équation différentielle pour les filtres analogiques) :

$$y(n) = - \sum_{k=1}^{M-1} a_k y(n-k) + \sum_{k=0}^{N-1} b_k x(n-k). \quad (6.3)$$

On a en utilisé, pour passer de (6.2) à (6.3), le fait que $TZ^{-1} [z^{-k} X(z)] = x(n-k)$ et on a fixé a_0 à 1 toujours par analogie avec les filtres analogiques. Les filtres rationnels sont divisés en deux catégories : les filtres rationnels dits "à réponse impulsionnelle infinie" (ou RII) et les filtres rationnels dits "à réponse impulsionnelle finie" (ou RIF)

6.4.1 Filtres à réponse impulsionnelle infinie (RII)

L'équation 6.3 définit la catégorie des filtres numériques rationnels dits à "Réponse Impulsionnelle Infinie" (RII, IIR en anglais) ou encore filtres récurrents. Ce nom vient du fait que ces filtres possèdent une boucle de réaction qui permet d'obtenir une réponse impulsionnelle qui s'auto-entretient :

$$h(n) = - \sum_{k=1}^{M-1} a_k h(n-k), \text{ pour } n \geq N \quad (6.4)$$

Chaque nouvelle valeur ne dépend que des valeurs passées et on peut donc ainsi obtenir un nombre infini de valeurs (ou coefficients) pour la réponse impulsionnelle.

6.4.2 Filtres à réponse impulsionnelle finie (RIF)

On définit les filtres numériques rationnels à "Réponse Impulsionnelle Finie" (RIF, FIR en anglais) ou filtres non récurrents, de la manière suivante :

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k} \quad (6.5)$$

en supprimant tout bonnement la boucle de réaction. Ce qui conduit dans le domaine temporel à :

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k). \quad (6.6)$$

Les coefficients (points, éléments) de la réponse impulsionnelle d'un filtre RIF sont donc directement donnés par les coefficients b_k :

$$h(n) = \sum_{k=0}^{N-1} b_k \delta(n-k) = b_n, \quad n = 0, \dots, N-1 \quad (6.7)$$

car

$$\delta(n-k) = \begin{cases} 1 & \text{pour } n = k \\ 0 & \text{pour } n \neq k. \end{cases} \quad (6.8)$$

Un des intérêts fondamentaux de ces filtres est qu'ils ne posent pas de problèmes d'instabilité. Ils présentent également très simplement un TPG constant, la condition étant que leur réponse impulsionnelle soit paire ou impaire, comme nous le verrons par la suite.

6.5 Stabilité des filtres numériques rationnels récurrents (RII)

6.5.1 Condition de stabilité

En supposant dans (6.2) que $N < M$ et en réalisant une décomposition en éléments simples on obtient :

$$H(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{\sum_{k=0}^{M-1} a_k z^{-k}} = \sum_{k=0}^{M-1} \frac{A_k}{1 - p_k z^{-1}} \quad (6.9)$$

Ce qui donne par transformée en z inverse (solution causale) :

$$h(n) = \sum_{k=0}^{M-1} A_k p_k^n u(n), \quad (6.10)$$

où $u(n)$ représente la fonction de Heaviside :

$$u(n) = \begin{cases} 1 & \text{pour } n > 0 \\ 0 & \text{pour } n \leq 0. \end{cases} \quad (6.11)$$

Afin d'assurer la stabilité du système il faut que $\sum_{n=-\infty}^{+\infty} |h(n)|$ soit borné. Or

$$\begin{aligned} \sum_{n=-\infty}^{+\infty} |h(n)| &\leq \sum_{n=-\infty}^{+\infty} \sum_{k=0}^{M-1} |A_k| |p_k|^n u(n) \\ \sum_{n=-\infty}^{+\infty} |h(n)| &\leq \sum_{k=0}^{M-1} |A_k| \sum_{n=0}^{+\infty} |p_k|^n \\ \sum_{n=-\infty}^{+\infty} |h(n)| &\leq \sum_{k=0}^{M-1} |A_k| \frac{1}{1 - |p_k|} \text{ si } |p_k| < 1 \end{aligned}$$

$\sum_{n=-\infty}^{+\infty} |h(n)|$ sera donc borné, et le système stable, si $|p_k| < 1 \forall k$.

Un filtre numérique rationnel de type RII est donc stable si tous les pôles (racines du dénominateur) de sa fonction de transfert $H(z)$ sont de modules inférieurs à 1 (on dit souvent dans le cercle unité \Leftrightarrow cercle de rayon 1).

Rappelons qu'il existe une condition similaire pour les filtres analogiques : un filtre analogique est stable si les pôles de sa fonction de transfert $H(p)$ sont à parties réelles négative.

6.5.2 Remarques

- Si tous les zéros (racines du numérateur) de la fonction de transfert du filtre, $H(z)$, sont de modules inférieurs à 1 (dans le cercle unité), le filtre est dit à minimum de phase. Ce type de filtres est très intéressant car il est d'inverse stable, ce qui est une propriété recherchée dans beaucoup d'applications.
- La position des pôles dans le cercle unité est également intéressante. Pour un filtre stable résonnant, on montre (voir exercice 4) que plus le module des pôles associés est proche de 1, plus la résonance est forte. Lorsque, pour un filtre stable, les pôles sont proches du cercle unité, on sait qu'il faudra faire davantage attention aux effets de la quantification des coefficients du filtre qui pourraient amener les pôles à sortir de la zone de stabilité.
- Nous verrons par la suite (exercice 4) qu'il est également possible de définir une zone de stabilité dans le plan des coefficients du filtre.
- Il existe des méthodes de conception de filtres numériques récurrents basées sur le placement des zéros et des pôles de leurs fonctions de transfert (voir références [6] et [7])

6.6 Synthèse des filtres numériques rationnels

6.6.1 Introduction

La synthèse d'un filtre numérique correspond au passage entre les spécifications à respecter, généralement données sur le module de la réponse en fréquence du filtre, et l'ensemble des coefficients définissant le filtre numérique. Les spécifications à respecter correspondent à un gabarit que l'on se donne et qui fixe des marges dans lesquelles la réponse en fréquence doit se trouver. Les figures 6.1 et 6.2 montrent un exemple de gabarit pour un filtrage de type passe-bas. La figure 6.3 présentent les 4 réponses en fréquence idéales classiques : pour un filtre

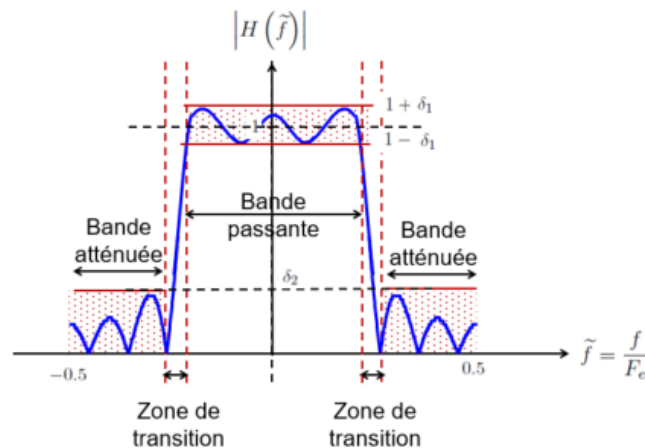


FIGURE 6.1 – Exemple de gabarit à respecter (en rouge) pour un filtre passe-bas.

passe-bas qui laisse passer les fréquences autour de 0 ("basses fréquences") jusqu'à la fréquence \tilde{f}_c (fréquence de coupure normalisée), pour un filtre passe-haut qui coupe les fréquences autour de 0 ("basses fréquences") jusqu'à la fréquence \tilde{f}_c et laisse passer les fréquences au-delà de \tilde{f}_c ("hautes fréquences"), pour un filtre passe-bande qui ne laisse passer que les fréquences se trouvant sur une certaine bande (entre \tilde{f}_{c1} et \tilde{f}_{c2}) et coupe toutes les autres, pour un filtre coupe-bande (ou réjecteur) qui laisse passer toutes les fréquences sauf celles se trouvant entre \tilde{f}_{c1} et \tilde{f}_{c2} .

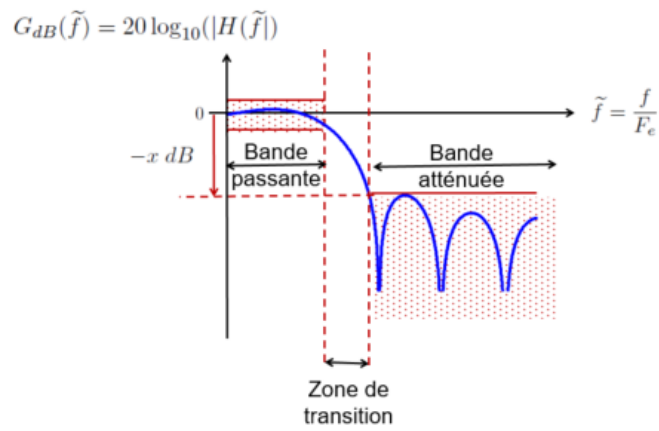


FIGURE 6.2 – Exemple de gabarit (en dB) à respecter (en rouge) pour un filtre passe-bas.

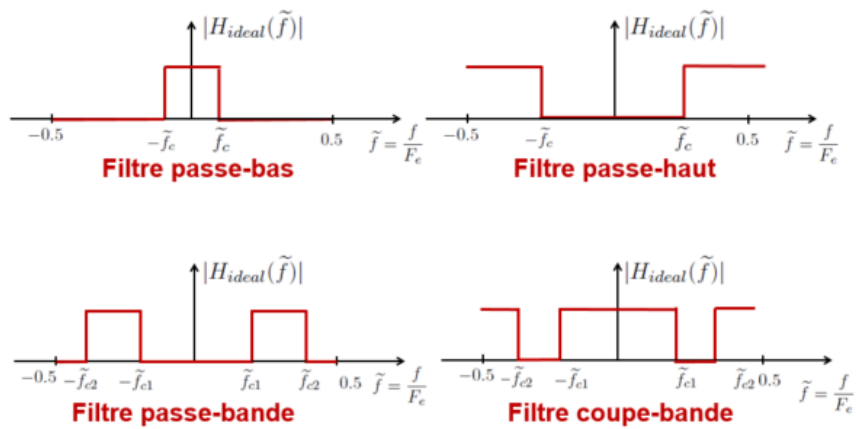


FIGURE 6.3 – Réponse en fréquences idéales (cibles pour la synthèse).

Nous pouvons envisager en numérique de réaliser un filtrage idéal : on calcule le spectre de notre signal (TFD), on met à 0 dans le tableau correspondant toutes les fréquences que l'on souhaite éliminer puis on revient au signal filtré par TFD inverse. Cependant, ce filtrage ne se fait pas en temps réel car ils sous entend que l'on dispose de tout les échantillons de signal pour en calculer le spectre. Or, dans le cadre d'une transmission, les échantillons de signal sont envoyés et arrivent les uns après les autres (toutes les T_e secondes si T_e représente la période d'échantillonnage). Un filtrage ne pourra être réalisé en temps réel que s'il est réalisé dans le domaine temporel, en implantant l'expression (6.3) ou l'expression (6.6), selon que l'on souhaite réaliser un filtre de type RII ou bien de type RIF et en s'assurant que le calcul de la/des somme(s) puisse se faire en T_e secondes. C'est cette implantation dans le domaine temporel qui va conduire, comme nous allons le voir par la suite, à un filtrage qui sera non idéal et donc à la nécessité de définir un gabarit à respecter. Notons également qu'il n'y aura pas un filtre unique respectant le gabarit. On pourra choisir celui qui satisfait le mieux aux critères que l'on se donne (le plus simple d'implantation, celui qui présente la charge de calcul la plus faible, le plus stable numériquement ...)

Les synthèses des filtres RIF et RII se font de manière très différente. Nous allons les décrire dans les paragraphes suivants.

6.6.2 Synthèse des filtres à réponse impulsionnelle finie

Synthèse par développement en série de Fourier

La synthèse de filtres RIF est assez intuitive et facile à mettre en oeuvre. On se donne une réponse en fréquences idéale, $H_I(\tilde{f})$, du filtre à réaliser et un gabarit à respecter. On travaille en numérique, on doit donc considérer que la réponse en fréquence est périodique de période 1 en fréquences normalisées (TF d'une réponse impulsionnelle qui devra être échantillonnée). Un filtre RIF peut donc être synthétisés directement par un développement en série de Fourier de la réponse en fréquence idéale :

$$H_I(\tilde{f}) = \sum_{k=-\infty}^{+\infty} h_I(k) e^{-j2\pi\tilde{f}k}. \quad (6.12)$$

où les coefficients de la série de Fourier $h_I(k)$ représentent les éléments de la réponse impulsionnelle, ou "coefficients", du filtre :

$$h_I(k) = \int_{-\frac{1}{2}}^{\frac{1}{2}} H_I(\tilde{f}) e^{+j2\pi k\tilde{f}} d\tilde{f} \quad (6.13)$$

$h_I(k)$ représente ici le $k^{\text{ième}}$ point de la reponse impulsionnelle $h_I(t)$ du filtre qui est échantillonnée avec une période d'échantillonnage $T_e = \frac{1}{F_e}$ (il s'agit en réalité de $h_I(kT_e)$).

En pratique le nombre de coefficients de la réponse impulsionnelle devra être limité. En effet celle-ci sera représentée par un tableau contenant un nombre fini de valeurs. On modélise cette limitation par l'application d'une fenêtre de troncature, $w(n)$, sur la réponse impulsionnelle idéale. Cela conduit à la réponse impulsionnelle réelle suivante :

$$h(n) = h_I(n) \times w(n)$$

et donc à une réponse en fréquence qui n'est plus idéale :

$$H(\tilde{f}) = H_I(\tilde{f}) * W(\tilde{f})$$

En effet, cette convolution de la réponse en fréquence idéale par la transformée de Fourier de la fenêtre de troncature va générer des transitions adoucies et des oscillations autour des transitions (voir tracé en bleu dans la figure 6.1). Nous retrouvons les mêmes phénomènes que ceux déjà évoqués dans le chapitre sur la transformée de Fourier d'un signal de durée limitée.

Lorsque l'on conserve tout simplement un certain nombre de points de la réponse impulsionnelle idéale pour former la réponse impulsionnelle réelle : $[h_I(-N)...h_I(N)]$, c'est comme si on utilisait une fenêtre de troncature rectangulaire. Dans les autres cas nous avons une pondération de la réponse impulsionnelle idéale, en plus de la troncature : $[h_I(-N)w(-N)...h_I(N)w(N)]$, en supposant que l'on conserve $2N+1$ éléments de $h_I(k)$ et que l'on utilise une fenêtre de troncature, w , de $2N+1$ échantillons : $[w(-N)...w(N)]$.

Le nombre de points, $2N+1$, conservé sur la réponse impulsionnelle idéale pour former le tableau représentant la réponse impulsionnelle réelle est appelé *ORDRE* du filtre. Les éléments du tableau représentant la réponse impulsionnelle réelle sont appelés *COEFFICIENTS* du filtre. La synthèse d'un filtre RIF va alors consister à déterminer l'ordre du filtre, ainsi que la fenêtre de troncature à utiliser, afin que celui-ci satisfasse au gabarit souhaité. Les figures 6.4 et 6.5 présentent des exemples d'influence de l'ordre et de la fenêtre de troncature utilisée dans la cas d'un filtrage passe-bas.

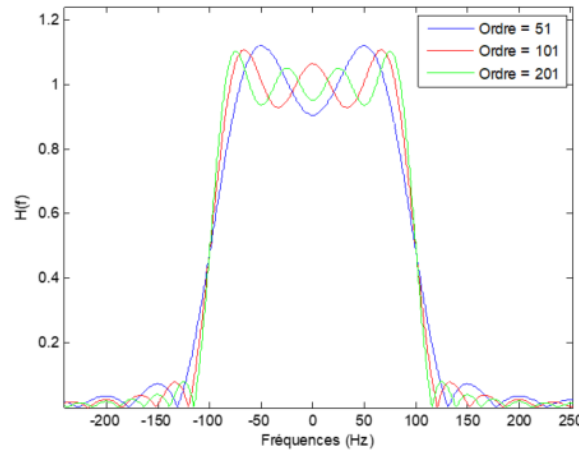


FIGURE 6.4 – Exemple d'influence de l'ordre dans la synthèse d'un filtre RIF de type passe-bas - Fenêtre rectangulaire

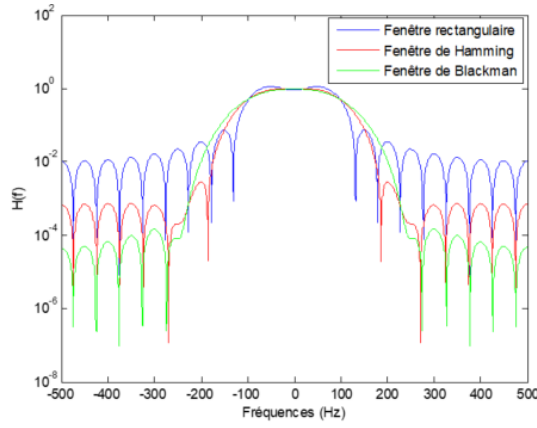


FIGURE 6.5 – Exemple d'influence de la fenêtre de troncature dans la synthèse d'un filtre RIF de type passe-bas - Ordre = 21

Causalité des filtres RIF

Attention la réponse impulsionnelle obtenue doit être causale pour que le filtre soit réalisable. Si elle ne l'est pas en sortie de la synthèse on devra la décaler afin qu'elle le devienne et, ce faisant, on introduira un retard : $h_{causal}(n) = h\left(n - \frac{N}{2}\right)$ si N est pair ou $h_{causal}(n) = h\left(n - \frac{N-1}{2}\right)$ si N est impair. Ce décalage ne modifie pas le module de la réponse en fréquence mais ajoute une phase linéaire en fréquence. En supposant N pair et en notant $H_{causal}(\tilde{f}) = TFD[h_{causal}(n)]$, on a :

$$H_{causal}(\tilde{f}) = H(\tilde{f})e^{-j\pi\tilde{f}N}$$

et donc :

$$\begin{aligned} |H_{causal}(\tilde{f})| &= |H(\tilde{f})| \\ Arg[H_{causal}(\tilde{f})] &= Arg[H(\tilde{f})] - \pi\tilde{f}N \end{aligned}$$

Remarques

- Il existe un calcul (très approximatif) permettant de se donner un ordre de départ pour la synthèse : $Ordre = \frac{2}{3} \log_{10} \left(\frac{1}{\delta_1 \delta_2} \right) \frac{F_c}{\Delta f}$, où $2\delta_1$ représente la largeur admise autour de 1 en bande passante (en linéaire pas en dB), δ_2 représente l'atténuation minimum en bande coupée (en linéaire pas en dB) et Δf représente la largeur de la zone de transition.
- En général le choix de la fenêtre fixe l'amplitude des oscillations et, pour une fenêtre donnée, l'ordre $(2N + 1)$ va fixer la largeur de la zone de transition : voir tableau 6.1. Comme cela a été vu dans le chapitre sur la transformée de Fourier, on retrouve ici le compromis à réaliser entre la largeur de la transition et l'amplitude des oscillations : une fenêtre qui permet d'avoir des oscillations de plus faible amplitude, comparée à une autre, fournit une zone de transition plus large pour un même ordre.
- Les résultats obtenus par synthèse directe peuvent être ensuite optimisés grâce, par exemple, à la méthode des moindres carrés (minimisation au sens des moindres carrés de la distance entre le gabarit désiré et le gabarit du filtre obtenu par la synthèse en série de Fourier, fonction *firls.m* sous Malab) ou à l'algorithme de Remez (pour obtenir la meilleure approximation du gabarit présentant des ondulations d'amplitude constante, fonction *firgr.m* sous Malab).

Type de fenêtre :	Largeur de la zone de transition ($\Delta\omega$)	Atténuation min en bande coupée (en dB)
Rectangulaire	$\frac{1.8\pi}{\text{Ordre}}$	21
Barlett	$\frac{5.6\pi}{\text{Ordre}}$	25
Hanning	$\frac{6.2\pi}{\text{Ordre}}$	44
Hamming	$\frac{6.6\pi}{\text{Ordre}}$	53
Blackman	$\frac{11\pi}{\text{Ordre}}$	74
Kaiser ($\beta = 4.54$)	$\frac{5.8\pi}{\text{Ordre}}$	50
Kaiser ($\beta = 5.67$)	$\frac{7.8\pi}{\text{Ordre}}$	60

TABLE 6.1 – Caractéristiques des fenêtres de troncature

- Il existe des fonctions matlab fournissant les coefficients du filtre souhaité sans avoir à réaliser le calcul de la impulsionnelle (voir, par exemple, *fir1.m* pour les filtres "classiques" ou *rcosdesign.m* pour réaliser un filtre en cosinus surélevé très utilisé en transmission). Il est également possible de rentrer la réponse en fréquence souhaitée comme un tableau de point et d'obtenir la réponse impulsionnelle correspondante en utilisant la TFD inverse (*ifft.m* sous Matlab).

Avantages/Inconvénients des filtres RIF

- Les filtres RIF sont fréquemment désignés par le terme de filtres non-récurrents, car ils ne présentent pas de boucle de réaction de la sortie vers l'entrée, ce qui assure leur stabilité de manière inconditionnelle (pour peu que les coefficients du filtre soient bornés).
- Les filtres RIF peuvent avoir très simplement un temps de propagation de groupe constant. En effet, si $\text{Arg}[H(\tilde{f})] = \text{constante}$ alors $\text{Arg}[H_{\text{causal}}(\tilde{f})] = \text{constante} - \pi\tilde{f}N$ et donc :

$$TPG(\tilde{f}) = -\frac{1}{2\pi} \frac{d\text{Arg}[H_{\text{causal}}(\tilde{f})]}{d\tilde{f}} = \frac{N}{2}$$

Or $\text{Arg}[H(\tilde{f})]$ est constant si la réponse impulsionnelle $h(n)$ est paire ou impaire :

$$\begin{aligned} \text{Arg}[H(\tilde{f})] &= \text{Arg}\left[\sum_{-\infty}^{+\infty} h(n) \left(\cos(2\pi\tilde{f}n) + j\sin(2\pi\tilde{f}n)\right)\right] \\ &= \text{atan}\left(\frac{\sum_{-\infty}^{+\infty} h(n) \sin(2\pi\tilde{f}n)}{\sum_{-\infty}^{+\infty} h(n) \cos(2\pi\tilde{f}n)}\right) \\ &= \begin{cases} 0 & \text{si } h(n) \text{ pair} \\ \frac{\pi}{2} & \text{si } h(n) \text{ impair.} \end{cases} \end{aligned}$$

- La synthèse des filtres RIF est très simple.
- Certains gabarits très contraints peuvent nécessiter l'utilisation de beaucoup trop de coefficients et donc un temps de calcul beaucoup trop élevé. Dans ce cas on utilisera plutôt un filtre de type RII.

6.6.3 Synthèse des filtres à réponse impulsionnelle infinie

Principe de la méthode

Les filtres RII sont des filtres qui présentent une boucle de réaction de la sortie vers l'entrée (filtres récurrents). L'idée de leur synthèse est de s'appuyer sur les bibliothèques très fournies de

modèles pour le filtrage analogique.

La synthèse de filtre analogique consiste, à partir d'un gabarit défini sur la réponse en fréquence souhaitée $H(f)$, à choisir un modèle dans la bibliothèque de modèles et à adapter ses paramètres de façon à obtenir un filtre satisfaisant au gabarit. Elle fournit une fonction de transfert en p : $H(p)$.

En numérique le gabarit est donné sur $H(\tilde{f})$ et nous souhaitons obtenir une fonction de transfert en z : $H(z)$.

Il "suffirait" donc de trouver un "passage" entre $H(\tilde{f})$ et $H(f)$, puis entre $H(p)$ et $H(z)$ pour utiliser ce qui existe en synthèse analogique à des fins numériques.

Mise en pratique

La figure 6.6 résume la manière de réaliser une synthèse de filtre RII. Les explications sont données dans la suite.

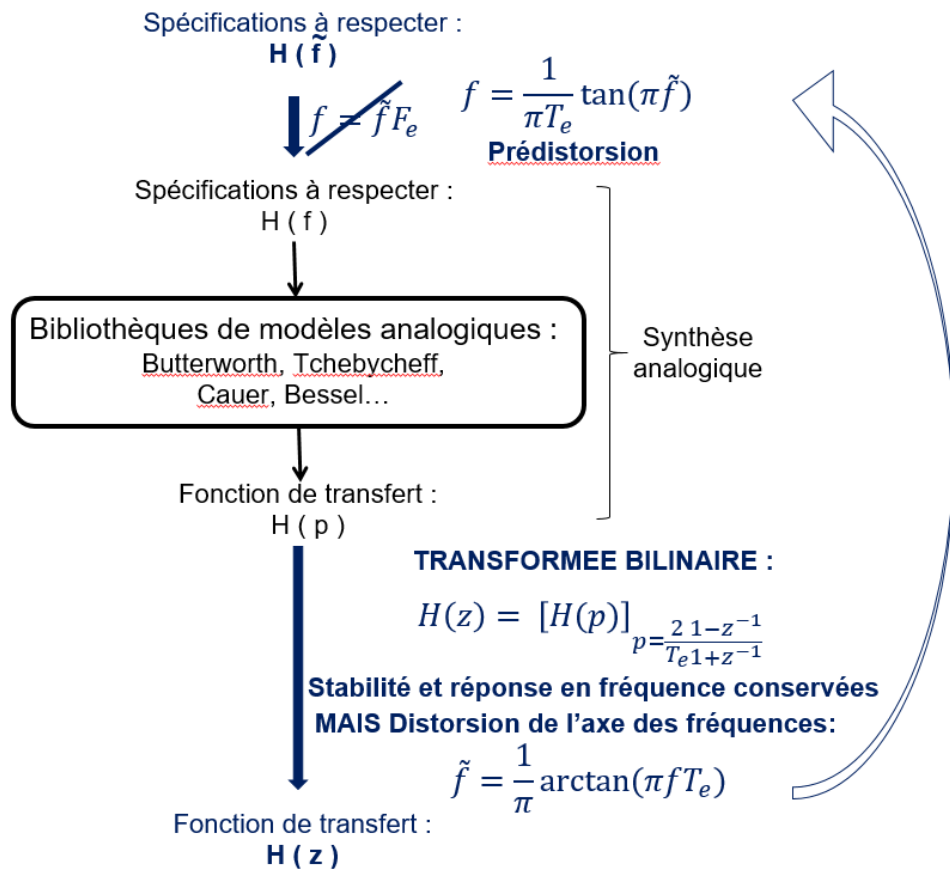


FIGURE 6.6 – Méthode de synthèse d'un filtre numérique rationnel de type RII (filtre récursif).

1. Le passage entre $H(\tilde{f})$ et $H(f)$ qui semble naturel est de prendre : $f = \tilde{f} F_e$. Nous allons cependant voir par la suite qu'il va falloir le modifier.
2. Le passage entre $H(p)$ et $H(z)$ doit permettre de conserver au filtre numérique la stabilité et la conformité au gabarit fixé obtenue à l'issue de la synthèse analogique. Pour cela on doit trouver une transformée du plan des p vers le plan des z qui transforme :
 - le demi plan gauche (lieu de stabilité en analogique : pôles de $H(p)$ à parties réelles négatives) en l'intérieur du cercle unité (lieu de stabilité en numérique : pôles de $H(z)$ de modules inférieurs à 1)

— l'axe imaginaire (lieu de parcours de la réponse en fréquence en analogique : $H(f) = [H(p)]_{p=j\omega=j2\pi f}$) en le cercle unité (lieu de parcours de la réponse en fréquence en numérique : $H(\tilde{f}) = [H(z)]_{z=e^{j2\pi\tilde{f}}}$).

Cette transformée est la transformée bilinéaire. Elle est obtenue par approximation numérique de l'opérateur intégrale définissant le passage dans le filtre analogique (voir figure 6.7) et donne $H(z)$, à partir de $H(p)$ de la manière suivante :

$$H(z) = [H(p)]_{p=\frac{2}{T_e} \frac{1-z^{-1}}{1+z^{-1}}} \quad (6.14)$$

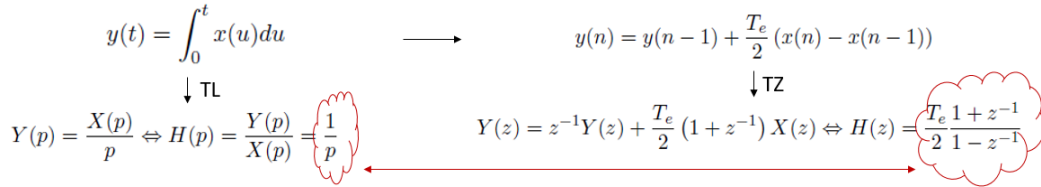


FIGURE 6.7 – Obtention de la transformée bilinéaire.

- Attention cependant car, même si la transformée bilinéaire permet de passer de $H(p)$ à $H(z)$ en conservant la stabilité et la réponse en fréquence du filtre, elle ne transforme pas les fréquences f en $\tilde{f} = \frac{f}{F_e}$. En effet, quand $p = j2\pi f$ alors $z = \frac{1+j2\pi f T_e/2}{1-j2\pi f T_e/2}$: on a bien transformé l'axe imaginaire en le cercle unité mais pas f en $\tilde{f} = \frac{f}{F_e}$: en écrivant $z = \frac{1+j2\pi f T_e/2}{1-j2\pi f T_e/2} = e^{j2\pi\tilde{f}}$ on obtient $\tilde{f} = \frac{1}{\pi} \arctan(\pi f T_e)$. Ceci n'est néanmoins pas gênant car il suffit de passer du gabarit sur $H(\tilde{f})$ au gabarit sur $H(f)$, non pas en faisant $f = \tilde{f} F_e$ mais en faisant $f = \frac{1}{\pi T_e} \tan(\pi \tilde{f})$. La distorsion de l'axe des fréquences introduite par la transformée bilinéaire est ainsi compensée par une prédistorsion lors du passage de $H(\tilde{f})$ vers $H(f)$.

Une fois établis les passages de $H(\tilde{f})$ vers $H(f)$ et de $H(p)$ vers $H(z)$, on constate donc que la synthèse analogique revient à faire une synthèse analogique : choix du modèle et adaptation des paramètres pour satisfaire au gabarit. Parmi les modèles les plus courants on trouve :

- Le modèle de Butterworth, qui est le plus simple : $|H(f)|^2 = \frac{1}{1+(\frac{f}{f_c})^{2n}}$, où f_c représente la fréquence de coupure. Ce modèle ne présente pas d'ondulations, ni en bande passante, ni en bande atténuée. L'ordre du filtre n permet de régler la raideur de la transition entre la bande passante et la bande atténuée. La zone de transition est plus large qu'avec les autres modèles pour un même ordre mais la phase de la réponse en fréquence est modérément non linéaire (TPG à peu près constant sur la bande passante).
- Le modèle de Tchebychef (ou Chebyshev) présente des ondulations en bande passante ou en bande atténuée (type I ou type II), ce qui permet une zone de transition plus étroite qu'un modèle de Butterworth pour un même ordre. La phase de la réponse en fréquence est, par contre, non linéaire (TPG non constant).
- Le modèle de Cauer (ou Elliptique) présente des ondulations en bande passante et en bande atténuée, ce qui permet une zone de transition encore plus étroite que les modèles précédents pour un même ordre. La phase de la réponse en fréquence est, pour ce modèle aussi, non linéaire (TPG non constant).

- Le modèle de Bessel présente une phase à peu près linéaire pour la réponse en fréquence, ne présente pas d'ondulations ni en bande passante ni en bande atténuée mais sa zone de transition va être beaucoup plus large que les autres filtres pour un même ordre. C'est un filtre de phase.

Des exemples de synthèse de filtres RII se trouvent dans la section exercices corrigés. Notons que de nombreuses fonctions sont proposées par Matlab pour réaliser une synthèse de RII en utilisant les différents modèles issus des bibliothèques analogiques.

Avantages/Inconvénients des filtres RII

- Du fait de la présence de pôles dans leur fonction de transfert (boucle de réaction), les filtres RII présentent des risques d'instabilité.
- Le temps de propagation de groupe des filtres RII n'est pas constant. Néanmoins, selon le modèle analogique utilisé, le TPG sera plus ou moins perturbé en bande passante.
- La synthèse des filtres RII est plus complexe que celle des filtres RIF.
- Le principal avantage des filtres RII va être de satisfaire certains gabarits avec un coût calculatoire plus faible que les filtres RIF (ordre, ou nombre de coefficients nécessaires, moins élevé).

6.7 Implantation des filtres numériques rationnels

Comme nous l'avons évoqué précédemment, un filtrage numérique ne pourra être réalisé en temps réel que s'il est réalisé dans le domaine temporel, en implantant l'expression (6.3) ou l'expression (6.6). Ces relations s'appuient sur 3 opérations élémentaires : le retard, la multiplication et l'addition. La fonction de transfert d'un système numérique effectuant un retard de T_e (un coup d'échantillonnage) étant z^{-1} , il est d'usage de représenter les retards unités dans les schémas bloc d'implantation des filtres par la notation z^{-1} .

6.7.1 Structure directe

La structure la plus directe pour implanter un filtre numérique rationnel est donné par le schéma de la figure 6.8. Cette implantation nécessite deux files d'attente et $\sim M + N + 1$ opérations d'additions/multiplications. Il est cependant possible de n'utiliser qu'une seule file d'attente : c'est la structure canonique présentée dans le paragraphe suivant.

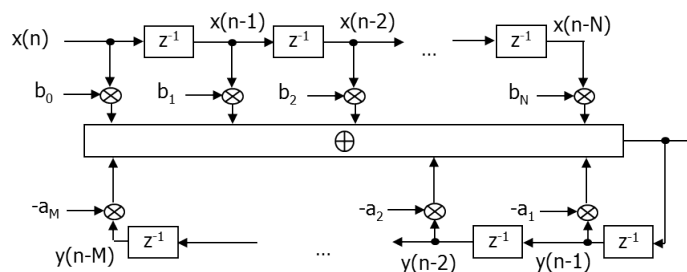


FIGURE 6.8 – Structure directe

6.7.2 Structure canonique

En utilisant une variable intermédiaire :

$$W(z) = \frac{X(z)}{\sum_{k=0}^{M-1} a_k z^{-k}}$$

on obtient $y(n) = \sum_{k=0}^{N-1} b_k w(n-k)$ avec $w(n) = -\sum_{k=1}^{M-1} a_k w(n-k) + x(n)$, et donc une structure d'implantation simplifiée, dite canonique et présentée dans la figure 6.9. Cette implantation ne nécessite qu'une seule file d'attente et $\sim M + N + 1$ opérations d'additions/multiplications.

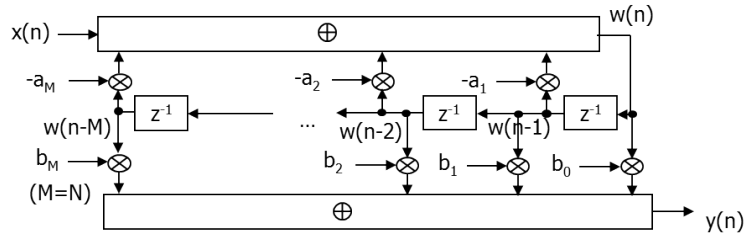


FIGURE 6.9 – Structure canonique

6.7.3 Structures décomposées

La plupart du temps on plante un filtre numérique en le décomposant en cellules du premier et du deuxième ordre. L'implantation peut se faire de deux manières : en série ou en parallèle.

Structure décomposée en série (ou cascade)

La figure 6.10 illustre la structure d'implantation par décomposition série. Elle est obtenue en exprimant la fonction de transfert globale du filtre comme un produit de fonctions de transfert plus simples :

$$H(z) = G \prod_{i=0}^{M-1} H_i(z)$$

où G est une constante et les $H_i(z)$ sont des cellules du premier et du deuxième ordre de la forme :

$$H_i(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$

ou

$$H_i(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

obtenues en factorisant le numérateur et le dénominateur de $H(z)$. Les cellules du premier ordre ont des pôles et des zéros réels, tandis que les cellules du second ordre ont des zéros et des pôles qui sont complexes conjugués, les coefficients a_k et b_k eux sont toujours réels.

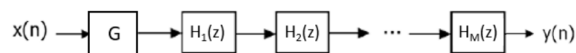


FIGURE 6.10 – Structure série

Structure décomposée en parallèle

La figure 6.11 représente une décomposition parallèle. Elle est obtenue en exprimant la fonction de transfert globale du filtre comme une somme de fonctions de transfert plus simples :

$$H(z) = G + \sum_{i=0}^{M-1} H_i(z)$$

où G est une constante et les $H_i(z)$ sont des cellules du premier et du deuxième ordre de la forme :

$$H_i(z) = \frac{b_0}{1 + a_1 z^{-1}}$$

ou

$$H_i(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

obtenues en décomposant $H(z)$ en éléments simples.

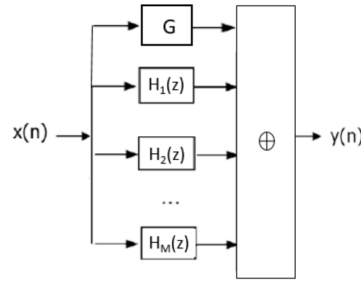


FIGURE 6.11 – Structure parallèle

6.7.4 Structure non réursive

Il s'agit de la structure des filtres à réponses impulsionnelles finies, ou filtres RIF. Elle est représentée dans la figure 6.12.

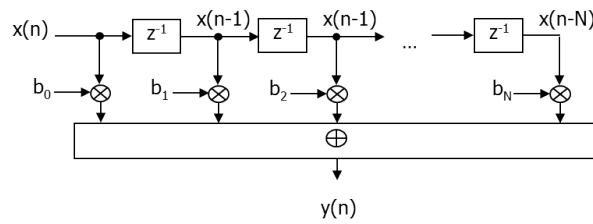


FIGURE 6.12 – Structure non réursive

6.8 Exercices

6.8.1 Exercice 1

On considère un filtre de fonction de transfert :

$$H(z) = \frac{1}{(1 - az^{-1})(1 - bz^{-1})} \quad (6.15)$$

où a et b sont deux réels $\in]0, 1[$ avec $b > a$.

1. Quel est l'ordre du filtre défini par la fonction de transfert $H(z)$?
Le filtre défini par la fonction de transfert $H(z)$ est d'ordre 2 (degré du dénominateur).
2. Déterminer l'équation récurrente définissant le filtre dans le domaine temporel.
 $Y(z)(1 - (a+b)z^{-1} + abz^{-2}) = X(z) \xrightarrow{TZ^{-1}} y(n) = x(n) + (a+b)y(n-1) - aby(n-2)$
3. Quel type de filtre rationnel (RIF, RII) est défini par $H(z)$? Justifiez votre réponse.
Ce filtre est de type RII car il présente une boucle de réaction : la sortie à l'instant n dépend de l'entrée à l'instant n mais également des valeurs passées de la sortie ($y(n-1)$ et $y(n-2)$).
4. Le filtre défini par $H(z)$ est-il stable ? Justifiez votre réponse.
Ce filtre sera stable si les pôles de $H(z)$ sont inclus dans le cercle de rayon 1. Pour cela il faut que $|a| < 1$ et $|b| < 1$, ce qui est le cas ici. Le filtre est donc stable.
5. En réutilisant les résultats de l'exercice 5.5.1 déterminer la réponse impulsionnelle $h(n)$ permettant de pouvoir réaliser le filtre.
 $h(n) = \frac{1}{a-b} \{a^{n+1} - b^{n+1}\} u(n)$ conduit à un filtre causal.

6.8.2 Exercice 2

Soit le filtre d'entrée $x(n)$ et de sortie $y(n)$ défini par l'équation récurrente suivante :

$$y(n) = x(n) - ax(n-1) \quad (6.16)$$

1. Déterminer sa fonction de transfert $H(z)$.
 $Y(z) = X(z) - az^{-1}X(z) \rightarrow H(z) = \frac{Y(z)}{X(z)} = 1 - az^{-1}$
2. Déterminer la transformée en z de $\delta(n)$ et de $\delta(n-1)$, où $\delta(n)$ représente le Dirac numérique :

$$\delta(n) = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{for } n \neq 0. \end{cases} \quad (6.17)$$

$$\delta(n) \rightarrow \Delta(z) = \sum_{n=-\infty}^{+\infty} \delta(n)z^{-n} = \delta(0)z^0 = 1, \quad TZ[\delta(n-1)] = z^{-1}TZ[\delta(n)] = z^{-1}.$$

En déduire la réponse impulsionnelle du filtre.

$$H(z) = \frac{Y(z)}{X(z)} = 1 - az^{-1} \rightarrow h(n) = TZ^{-1}[H(z)] = \delta(n) - a\delta(n-1)$$

Remarque : on pouvait obtenir directement $h(n)$ pour $x(n) = \delta(n)$ en utilisant l'équation de récurrence définissant le système.

3. Déterminer la transformée en z de la fonction échelon unité $u(n)$, ainsi que son domaine d'existence. En déduire la réponse indicielle du filtre (réponse à un échelon).

$$U(z) = \sum_{n=0}^{+\infty} z^{-n} = \frac{1}{1 - z^{-1}}$$

Domaine d'existence :

$$\lim_{n \rightarrow +\infty} \sqrt[n]{|z^{-n}|} < 1 \text{ pour } |z| > 1$$

Réponse indicielle :

$$Y(z) = H(z)U(z) = \frac{1 - az^{-1}}{1 - z^{-1}} = \frac{1}{1 - z^{-1}} - \frac{az^{-1}}{1 - z^{-1}} \rightarrow y(n) = u(n) - au(n-1)$$

Remarque : on pouvait obtenir directement la réponse indicielle (= réponse à un échelon) pour $x(n) = u(n)$ en utilisant l'équation de récurrence définissant le système.

4. Le filtre défini par l'équation 6.16 est il un filtre RIF ou un filtre RII ? Justifiez votre réponse.

Ce filtre est un filtre à Réponse Impulsionnelle Finie (RIF) car il ne présente pas de boucle de réaction : la sortie à l'instant n ne dépend pas de ses valeurs passées.

5. Le filtre défini par l'équation 6.16 est il stable ? Justifiez votre réponse.

Un filtre RIF est stable si ses coefficients sont finis, ce qui est le cas ici.

6. Le filtre défini par l'équation 6.16 est il causal ? Justifiez votre réponse.

Ce filtre est causal car sa réponse impulsionnelle est nulle pour $n < 0$.

6.8.3 Exercice 3 : synthèse d'un filtre passe-bas de type RIF

On veut synthétiser un filtre passe-bas en essayant d'approcher par un filtre RIF la fonction de transfert idéale de la figure 6.13. Donner l'expression de la réponse impulsionnelle d'un filtre à $2N+1$ coefficients utilisant une fenêtre rectangulaire de troncature et d'un filtre à $2N+1$ coefficients utilisant une fenêtre de troncature de Hamming donnée par $w(n) = 0.54 + 0.46 \cos(\frac{2\pi n}{2N+1})$.

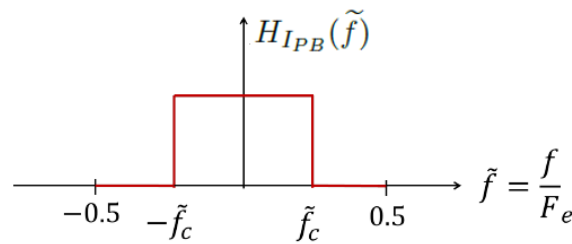


FIGURE 6.13 – Filtre passe-bas - Fonction de transfert idéale

La réponse en fréquence idéale $H_{IPB}(\tilde{f})$ est périodique, donc décomposable en série de Fourier :

$$H_{IPB}(\tilde{f}) = \sum_{k=-\infty}^{+\infty} h_{IPB}(k) e^{-j2\pi \tilde{f}k} \quad (6.18)$$

où les coefficients de la série de Fourier $h_{IPB}(k)$ représentent les éléments de la réponse impulsionnelle, ou "coefficients", du filtre. Ici :

$$h_{IPB}(k) = \int_{-\infty}^{+\infty} H_{IPB}(\tilde{f}) e^{+j2\pi \tilde{f}k} d\tilde{f} = \int_{-\tilde{f}_c}^{\tilde{f}_c} e^{+j2\pi \tilde{f}k} d\tilde{f}. \quad (6.19)$$

conduisant, après calculs, à $h_{I_{PB}}(k) = 2\tilde{f}_c \text{sinc}(2\pi\tilde{f}_c k)$. En pratique le nombre de coefficients du filtre devra être limité à un nombre $2N+1$, appelé "ordre" du filtre. On modélise cette limitation par l'utilisation d'une fenêtre de troncature, ou de pondération, $w(k)$, de longueur $2N+1$:

$$h_{PB}(k) = h_{I_{PB}}(k) \times w(k)$$

Elle conduit à une réponse en fréquence approchée :

$$H_{PB}(\tilde{f}) = H_{I_{PB}}(\tilde{f}) * W(\tilde{f})$$

où $W(\tilde{f})$ est la transformée de Fourier de $w(k)$.

On propose d'utiliser ici une fenêtre rectangulaire de troncature ou une fenêtre de Hamming. Dans le cas de la fenêtre rectangulaire on a :

$$\begin{aligned} h_{PB}(k) &= 2\tilde{f}_c \text{sinc}(2\pi\tilde{f}_c k) \text{ pour } k = -N, \dots, N \\ &= 0 \text{ ailleurs} \end{aligned} \quad (6.20)$$

Dans le cas de la fenêtre de Hamming on a :

$$\begin{aligned} h_{PB}(k) &= 2\tilde{f}_c \text{sinc}(2\pi\tilde{f}_c k) \times \left(0.54 + 0.46 \cos\left(\frac{2\pi k}{2N+1}\right) \right) \text{ pour } k = -N, \dots, N \\ &= 0 \text{ ailleurs} \end{aligned} \quad (6.21)$$

Les figures 6.14 et 6.15 tracent les réponses impulsionnelles et les fonctions de transfert des deux filtres pour une fréquence de coupure $f_c = 100$ Hz, une fréquence d'échantillonnage $F_e = 800$ Hz et un ordre $N = 31$.

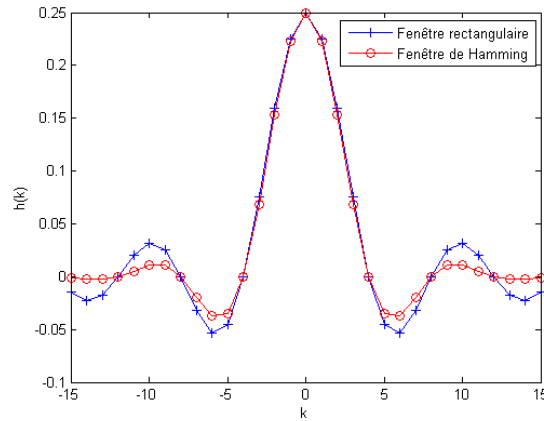


FIGURE 6.14 – Réponses impulsionnelles du filtre passe-bas de type RIF synthétisé avec les paramètres suivants : $f_c = 100$ Hz, $F_e = 800$ Hz, $\text{ordre} = 31$, pour deux fenêtres de troncature : rectangulaire et Hamming.

La figure 6.16 trace un exemple de signal d'entrée et de signal de sortie du filtre utilisant la fenêtre rectangulaire. On peut constater le retard du signal de sortie comparé au signal d'entrée. Celui-ci est dû au décalage de la réponse impulsionnelle appliqué afin de rendre le filtre causal. Le signal de sortie est déformé par rapport au signal d'entrée. Certaines fréquences ont effectivement été supprimées, comme le montre la figure 6.17 qui trace la TFD du signal d'entrée et celle du signal de sortie correspondant.

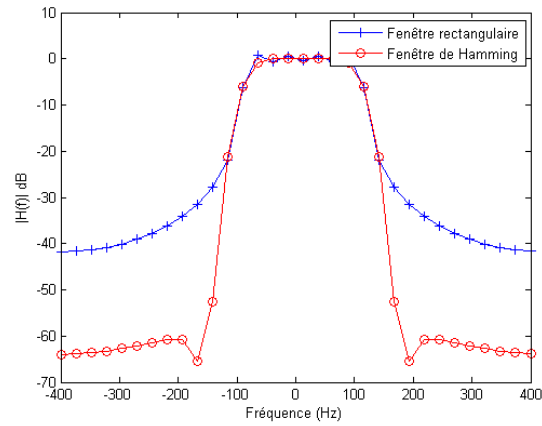


FIGURE 6.15 – Fonctions de transfert du filtre passe-bas de type RIF synthétisé avec les paramètres suivants : $f_c = 100$ Hz, $F_e = 800$ Hz, $ordre = 31$, pour deux fenêtres de troncature : rectangulaire et Hamming.

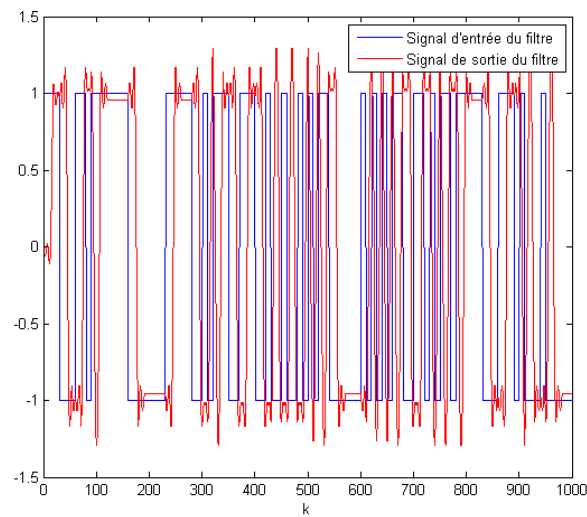


FIGURE 6.16 – Exemple de signaux d'entrée et de sortie du filtre passe-bas de type RIF synthétisé avec les paramètres suivants : $f_c = 100$ Hz, $F_e = 800$ Hz, $ordre = 31$, fenêtre rectangulaire de troncature.

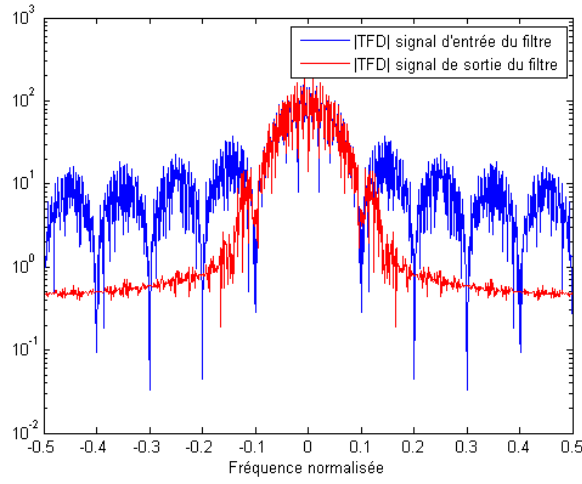


FIGURE 6.17 – TFD des signaux d’entrée et de sortie du filtre passe-bas de type RIF synthétisé avec les paramètres suivants : $f_c = 100$ Hz, $F_e = 800$ Hz, $ordre = 31$, fenêtre rectangulaire de troncature.

Remarques :

- A partir de la réponse impulsionnelle du filtre passe-bas il est possible d’obtenir les réponses impulsionnelles des autres filtres de base (passe-haut, passe-bande, rejeteur). Par exemple, un filtre passe-haut idéal de fréquence de coupure \tilde{f}_c présente une réponse en fréquence

$$H_{IPH}(\tilde{f}) = 1 - H_{IPB}(\tilde{f})$$

où $H_{IPB}(\tilde{f})$ est la réponse en fréquence du filtre passe-bas idéal de même fréquence de coupure. On peut donc en déduire la réponse impulsionnelle du filtre passe-haut idéal, $h_{IPH}(n)$, à partir de celle du filtre passe-bas idéal correspondant, $h_{IPB}(n)$:

$$h_{IPH}(n) = \delta(n) - h_{IPB}(n)$$

Cette réponse impulsionnelle idéale sera ensuite tronquée à $2N+1$ coefficients pour donner $h_{PH}(n)$, réponse impulsionnelle recherchée.

- La fonction `filter.m` de Matlab permet de réaliser le filtrage du signal x pour donner le signal y en utilisant un filtre défini par les tableaux de coefficients $A = [a_0 \dots a_{M-1}]$ et $B = [b_0 \dots b_{M-1}]$: $y = \text{filter}(B, A, x)$. Dans le cas d’un filtre RIF le tableau de coefficients A se résume à un seul coefficient $a_0 = 1$ et le tableau B contient les $2N + 1$ éléments conservés sur la réponse impulsionnelle pour un filtre d’ordre $2N + 1$: $B = [h_{PB}(-N) \dots h_{PB}(N)]$. Notons qu’un filtrage de type RIF peut également être réalisé en utilisant la fonction `conv.m` de Matlab : $y = \text{conv}(x, B, 'same')$, le paramètre 'same' n’est pas obligatoire mais il permet de ne pas avoir à gérer soi même le retard introduit par le filtre (permet de gérer les effets de bord de la convolution et renvoie la partie centrale du résultat de la convolution avec une taille égale à celle du premier vecteur).

6.8.4 Exercice 4 : étude de la cellule du second ordre

Cellule du second ordre purement récurrente

On la définit par l'équation aux différences suivantes :

$$y(n) = x(n) - a_1 y(n-1) - a_2 y(n-2)$$

1. Exprimez sa fonction de transfert en z .

$$y(n) = x(n) - a_1 y(n-1) - a_2 y(n-2) \xrightarrow{TZ} Y(z) = X(z) - a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z).$$

D'où la fonction de transfert du filtre :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

2. Dans le plan des coefficients (a_1 en abscisse, a_2 en ordonnées), tracez le domaine de stabilité du filtre.

Un filtre numérique est stable si les pôles de sa fonction de transfert $H(z)$ sont inclus dans le cercle de rayon 1. Cherchons les pôles de $H(z)$:

$$H(z) = \frac{z^2}{z^2 + a_1 z + a_2} = \frac{z^2}{a(z - z_1)(z - z_2)}$$

Il faut résoudre $z^2 + a_1 z + a_2 = 0$. On calcule donc $\Delta = a_1^2 - 4a_2$ et on doit considérer les 3 cas possibles :

- (a) $\Delta < 0$: Deux pôles complexes conjugués de modules $= \sqrt{a_2}$. Il faudra donc que $|a_2| < 1$. On est dans le cas d'une vraie cellule du second ordre.
- (b) $\Delta = 0$: Un pôle double $z_0 = -\frac{a_1}{2}$. Il faudra donc que $|a_1| < 2$.
- (c) $\Delta > 0$: Deux pôles réels (deux cellules du premier ordre) :

$$z_1 = \frac{-a_1 - \sqrt{\Delta}}{2}, \quad z_2 = \frac{-a_1 + \sqrt{\Delta}}{2}$$

En remarquant que $-1 \leq z_1 \leq z_2 \leq 1$, on arrive aux conditions suivantes : $a_2 \geq a_1 - 1$, $a_2 \geq -a_1 - 1$ et $-2 \leq a_1 \leq 2$.

On regroupe ensuite tous les cas possibles pour obtenir, dans le plan des coefficients (a_1 , a_2), le domaine de stabilité du filtre qui est un triangle : figure 6.18.

3. Donnez l'expression de la réponse en fréquence en fonction de a_1 et a_2 .

$$|H(\tilde{\omega})|^2 = \left[H(z) H\left(\frac{1}{z}\right) \right]_{z=e^{j\tilde{\omega}}}$$

Ce qui donne :

$$|H(\tilde{\omega})|^2 = \frac{1}{1 + a_1^2 + a_2^2 + 2a_1(1 + a_2) \cos(\tilde{\omega}) + 2a_2 \cos(2\tilde{\omega})}$$

4. A quelle condition existe-t-il une pulsation de résonance $\tilde{\omega}_0$ ($\tilde{\omega} = 2\pi f$) ?

En dehors de $\tilde{\omega} = 0$ et $\tilde{\omega} = 2\pi \times 0.5$, l'annulation de la dérivée de $|H(\tilde{\omega})|^2$ montre qu'elle passe par un maximum pour $\cos(\tilde{\omega}_0) = -\frac{a_1(1+a_2)}{4a_2}$. On a donc résonance si $\left| \frac{a_1(1+a_2)}{4a_2} \right| \leq 1$.

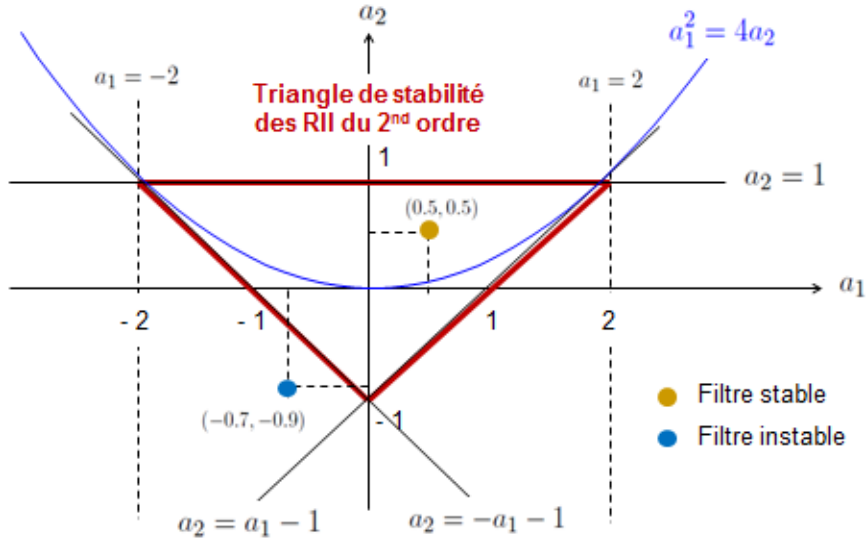


FIGURE 6.18 – Triangle de stabilité des filtres numériques du second ordre.

5. Montrez que la valeur du module de la réponse harmonique à la résonance est inversement proportionnelle à la distance des pôles au cercle de rayon 1. On se placera dans le cas où $a_1^2 < 4a_2$ (vraie cellule du second ordre) et on écrira la réponse en fréquence en $\tilde{\omega}_0$ sous forme polaire. On donne :

$$|H(\tilde{\omega}_0)| = \frac{2\sqrt{a_2}}{(1 - a_2)\sqrt{4a_2 - a_1^2}}$$

On a $\cos(\tilde{\omega}_0) = -\frac{a_1(1+a_2)}{4a_2}$ et $\cos(2\tilde{\omega}_0) = 2\frac{a_1^2(1+a_2)^2}{16a_2} - 1$, ce qui donne, après calculs :

$$|H(\omega_0)| = \frac{2\sqrt{a_2}}{(1 - a_2)\sqrt{4a_2 - a_1^2}}$$

Pour $z_1 = re^{j\theta}$ et $z_2 = re^{-j\theta}$ (pôles complexes conjugués) : $a_2 = r^2$ et $a_1 = -2r \cos(\theta)$ et donc :

$$|H(\omega_0)| = \frac{1}{(1 - r)(1 + r) \sin(\theta)}$$

La valeur du module de la réponse harmonique à la résonance est bien inversement proportionnelle à la distance des pôles au cercle unité. Plus le pôle est proche du cercle (a_2 proche de 1) plus la résonance sera forte. On peut donc lier la position des pôles dans le cercle de rayon 1 du plan complexe avec l'effet spectral produit.

6. Donnez l'expression de la réponse impulsionnelle en fonction des coordonnées polaires r et θ des pôles dans le cas où $a_1^2 < 4a_2$.

$$H(z) = \frac{1}{(1 - z_1 z^{-1})(1 - z_2 z^{-1})} = \frac{z_1}{z_1 - z_2} \frac{1}{1 - z_1 z^{-1}} + \frac{z_2}{z_2 - z_1} \frac{1}{1 - z_2 z^{-1}}$$

$$\xrightarrow{TZ^{-1}} h(n) = \frac{z_1^{n+1} - z_2^{n+1}}{z_1 - z_2} u(n) = r^n \frac{\sin((n+1)\theta)}{\sin(\theta)} u(n)$$

7. Proposez une structure de réalisation de ce filtre.

La figure 6.19 propose une structure de réalisation pour ce filtre.

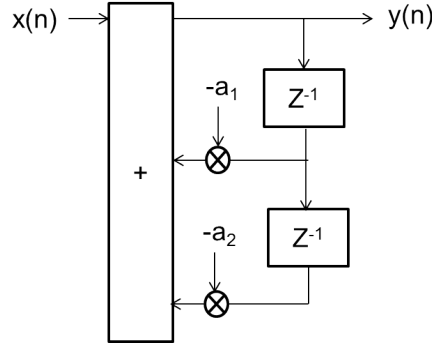


FIGURE 6.19 – Structure de réalisation de la cellule du second ordre purement récurrente.

Cellule du second ordre générale

On considère une équation générale de la cellule du second ordre :

$$y(n) = x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2)$$

1. Exprimez sa fonction de transfert en z .

$$y(n) = x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2)$$

$$\xrightarrow{TZ} Y(z) = X(z) + b_1z^{-1}X(z) + b_2z^{-2}X(z) - a_1z^{-1}Y(z) - a_2z^{-2}Y(z)$$

D'où la fonction de transfert :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$$

2. Montrez que cette cellule du second ordre peut être considérée comme la mise en cascade de la cellule purement récurrente précédente et d'un filtre RIF.

$$H(z) = \frac{1}{1 + a_1z^{-1} + a_2z^{-2}} \times (1 + b_1z^{-1} + b_2z^{-2})$$

3. En déduire une structure de réalisation.

En cascade de la cellule du second ordre purement récurrente (RII) et le filtre RIF (figure 6.20), on peut écrire les relations suivantes :

$$Y_1(z) = X(z) \frac{1}{1 + a_1z^{-1} + a_2z^{-2}} \xrightarrow{TZ^{-1}} y_1(n) = x(n) - a_1y(n-1) - a_2y(n-2)$$

$$Y(z) = Y_1(z) (1 + b_1z^{-1} + b_2z^{-2}) \xrightarrow{TZ^{-1}} y(n) = y_1(n) + b_1y_1(n-1) + b_2y_1(n-2)$$

D'où la structure de réalisation donnée par la figure 6.21. Il s'agit d'une structure canonique (une seule ligne à retard en cascade de RII puis RIF).

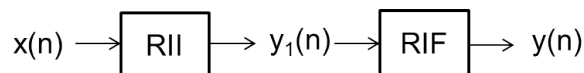


FIGURE 6.20 – Cascade d'un filtre RIF et d'un filtre RII.

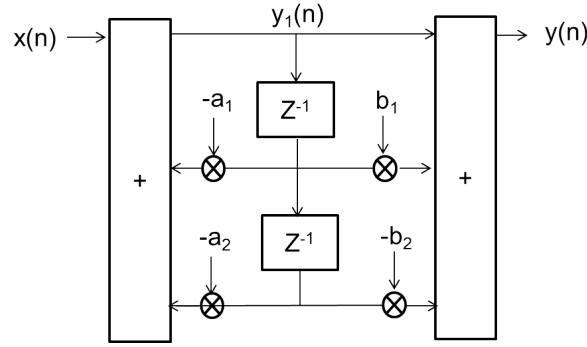


FIGURE 6.21 – Struture canonique de réalisation de la cellule du second ordre.

4. Pour $b_2 = 1$ montrez que la phase du RIF est linéaire.

Pour $b_2 = 1$ on a : $H_{RIF}(z) = 1 + b_1 z^{-1} + z^{-2}$. Et donc :

$$H_{RIF}(\tilde{\omega}) = e^{-j\tilde{\omega}}(b_1 + 2 \cos(\tilde{\omega})) \Rightarrow \text{Arg}[H_{RIF}(\tilde{\omega})] = -\tilde{\omega}$$

On a bien une phase linéaire. On aurait pu dire que le temps de propagation de groupe était constant en regardant $h_{RIF} = [1 \ b_1 \ 1]$ qui est symétrique.

6.8.5 Exercice 5 : synthèse RII guidée

On se propose de synthétiser un filtre passe-bas numérique de type RII. Le gabarit à respecter est donné par la figure 6.22. Afin de simplifier les calculs la fréquence d'échantillonnage sera considérée égale à 1Hz.

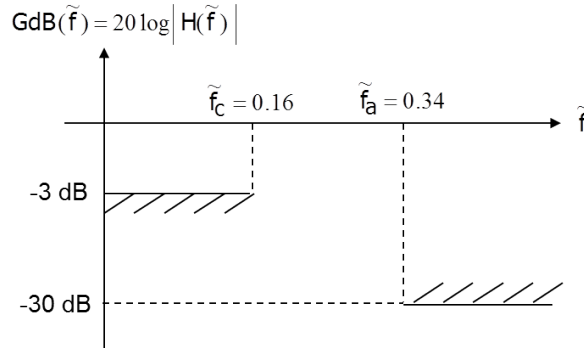


FIGURE 6.22 – Gabarit numérique à respecter

1. La synthèse de filtre RII est une méthode de synthèse numérique qui utilise la synthèse analogique. Cette synthèse analogique a besoin en entrée d'un gabarit analogique à respecter. On passera donc dans un premier temps du gabarit numérique sur $H(\tilde{f})$ au gabarit d'entrée de la synthèse analogique portant sur $H(f)$. Tracez le gabarit à respecter par $H(f)$.

Les atténuations du gabarit analogique restent identiques à celle du gabarit numérique mais l'axe de fréquence doit subir la prédistorsion suivante : $f = \frac{1}{\pi T_e} \tan(\pi \tilde{f})$, qui conduit au gabarit de la figure 6.23.

2. Réalisation de la synthèse analogique :

- (a) Première étape : on doit choisir une fonction modèle analogique et régler ses paramètres afin de satisfaire le gabarit souhaité. On impose dans cet exercice d'utiliser le modèle

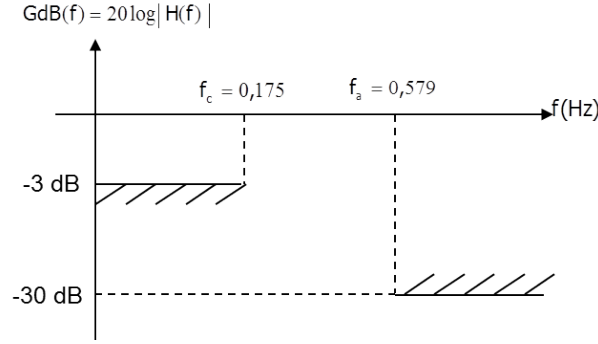


FIGURE 6.23 – Gabarit analogique à respecter

passé-bas de Butterworth, dont la fonction de transfert est donnée par :

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}},$$

où ω_c représente la pulsation de coupure. Montrez que le paramètre N permettant au modèle de Butterworth de satisfaire le gabarit à moindre coût est égal à 3.

L'atténuation pour $\omega = \omega_c = 2\pi f_c$ est fixée à -3 dB, le paramètre N va permettre de satisfaire à l'atténuation minimale souhaitée en bande atténuée. On souhaite :

$$10 \log_{10} \frac{1}{1 + \left(\frac{\omega_a}{\omega_c}\right)^{2N}} \leq -30 \text{ dB} = 10 \log \frac{1}{10^3}$$

où $\omega_a = 2\pi f_a$. Connaissant $\omega_c = 2\pi \times 0,175$ rad/s, cela conduit à choisir $N = 3$.

(b) Deuxième étape : passer de $|H(\omega)|^2$ à $H(p)$.

$$|H(\omega)|^2 = \left[|H(p)|^2 \right]_{p=j\omega}$$

d'où pour $N = 3$:

$$|H(p)|^2 = \frac{1}{1 - p^6} = H(p)H(-p)$$

(on oublie pour l'instant le $\frac{1}{\omega_c}$, sachant qu'on remplacera p par $\frac{p}{\omega_c}$ à la fin).

Parmi les 6 pôles de $|H(p)|^2$ (qui sont les racines sixièmes de l'unité), 3 appartiennent à $H(p)$, 3 appartiennent à $H(-p)$. On choisira comme pôles pour $H(p)$: $p_0 = -1$, $p_1 = -\frac{1}{2} - j\frac{\sqrt{3}}{2}$, $p_2 = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$. Expliquez ce qui conduit à ce choix.

Ces pôles sont à parties réelles négatives et conduiront donc à un filtre stable.

On en déduit donc la fonction de transfert suivante :

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

soit en remplaçant p par $\frac{p}{\omega_c}$:

$$H(p) = \frac{\omega_c^3}{(p + \omega_c)(p^2 + p\omega_c + \omega_c^2)}$$

(c) Troisième étape : Application de la transformée bilinéaire. Après application de la

transformée bilinéaire sur $H(p)$ on obtient la fonction de transfert suivante :

$$H(z) = H_1(z)H_2(z)$$

avec

$$H_1(z) = \frac{0.43(1 + z^{-1})}{1 - 0.29z^{-1}}, \quad H_2(z) = \frac{0.135 + 0.27z^{-1} + 0.135z^{-2}}{1 - 0.753z^{-1} + 0.4z^{-2}}$$

3. Le filtre obtenu est-il stable ? Justifiez votre réponse.

Le filtre obtenu est décomposé en un filtre du premier ordre qui est stable car son pôle $(0, 29)$ est de module inférieur à 1 et un filtre du second ordre qui est également stable car $(a_1, a_2) = (-0.753, 0.4)$ appartient au triangle de stabilité (voir exercice précédent).

4. Le filtre obtenu est-il résonnant ? Justifiez votre réponse.

Le filtre obtenu n'est pas résonnant car $\left| \frac{a_1(1+a_2)}{4a_2} \right| = 6.588 > 1$ (voir exercice précédent).

5. On souhaite filtrer un signal x en utilisant le filtre RII synthétisé. En appelant y le signal de sortie, proposer un programme matlab permettant de passer de x à y . Ce programme pourra être testé pour filtrer des sinusoides lors des séances de TP.

Afin de filtrer un signal x en utilisant ce filtre RII, on pourra écrire sous Matlab :

$$y1 = filter([0.43 \ 0.43], [1 \ -0.29], x);$$

$$y = filter([0.135 \ 0.27 \ 0.135], [1 \ -0.753 \ 0.4], y1);$$

Chapitre 7

TPs (séances 1 et 2)

7.1 Introduction

Les deux premières séances de TP seront consacrées à la mise en place des outils nécessaires à la réalisation du projet et à la bonne compréhension de leur utilisation. Aucun compte-rendu ne vous sera demandé sur ces parties là.

7.2 Séance 1 : Génération et étude d'un signal numérique : le cosinus

7.2.1 Représentation temporelle

Un signal numérique, x , est représenté sous Matlab par un tableau contenant un nombre fini, N , de valeurs représentant des échantillons de signal prélevés toutes les T_e secondes (échantillonnage uniforme) : $[x(0) \ x(1) \dots \ x(N-1)]$, le $k^{\text{ième}}$ élément $x(k)$ représentant en réalité $x(kT_e)$.

1. Générer 90 échantillons d'un cosinus (fonction *cos.m* sous matlab), d'amplitude 1 (V), de fréquence $f_0 = 1100$ Hz et échantillonné à $F_e = 10000$ Hz.
2. En utilisant la fonction *plot.m* de Matlab, tracer le cosinus généré précédemment avec une échelle temporelle en secondes. On doit pouvoir retrouver, à partir du tracé, la fréquence et l'amplitude du cosinus. Pensez à ajouter des labels sur vos axes en utilisant les fonctions *xlabel.m* et *ylabel.m* de Matlab
3. Générer 90 échantillons d'un cosinus (fonction *cos.m* sous matlab), d'amplitude 1 (V), de fréquence $f_0 = 1100$ Hz et échantillonné à $F_e = 1000$ Hz.
4. En utilisant la fonction *plot.m* de Matlab, tracer le cosinus généré précédemment avec une échelle temporelle en secondes (figure 2 de votre programme). La fréquence mesurée sur le cosinus tracé n'est pas $f_0 = 1100$ Hz. Vous devez être capables d'expliquer pourquoi et d'où vient la valeur de la fréquence observée.

7.2.2 Représentation fréquentielle

La transformée de Fourier numérique du signal x devra être estimée à partir du tableau de points représentant le signal numérique. Elle sera, elle même, représentée par un tableau, contenant un nombre fini de valeurs, représentant des échantillons de la TFD du signal calculés avec un certain pas. On obtient en effet sous Matlab,

en utilisant la fonction *fft.m* sur le signal x , un tableau : $[X(0) X(1) \dots X(N-1)]$, le $n^{\text{ième}}$ élément $X(n)$ représentant en réalité $X_D(n\Delta f)$, si Δf désigne le pas de calcul de la TFD. En considérant que l'on calcule N points pour la TFD entre 0 et F_e , le $n^{\text{ième}}$ élément $X(n)$ du tableau représentant la TFD du signal x est alors $X_D(n\frac{F_e}{N})$.

1. Vous devez être capable de répondre à la question suivante : qu'est-ce qui peut justifier que l'on calcule la transformée de Fourier numérique entre 0 et F_e ?
2. Pour chaque cosinus généré, estimez sa transformée de Fourier en utilisant la fonction *fft.m* de Matlab et tracez en le module (fonction *abs.m* de Matlab), en échelle log grâce à la fonction *semilogy.m*. L'échelle fréquentielle devra être en Hz. Vous devez être capables d'expliquer le tracé obtenu : retrouvez-vous les fréquences des cosinus générés ?
3. Choisissez un des cosinus générés précédemment et estimez sa transformée de Fourier sur un nombre de points, N' , supérieur au nombre de points de signal, en utilisant la technique du zero padding : $X = \text{fft}(x, N')$. Tracez le module de la transformée de Fourier obtenue (en échelle log avec une échelle fréquentielle en Hz) pour plusieurs valeurs de N' . En comparant les résultats obtenus, vous devez être capable d'en déduire l'intérêt de la technique dite du Zero Padding.
Attention afin d'utiliser l'algorithme de calcul rapide de la TFD (FFT) on devra utiliser un nombre de points N' égal à une puissance de 2.
4. Un autre outil permet de visualiser la représentation fréquentielle d'un signal, notamment pour les signaux aléatoires. Il s'agit de la Densité Spectrale de Puissance (DSP). Choisissez un des cosinus générés précédemment, estimez, puis tracez, sa DSP. Vous devez pouvoir retrouver, sur le tracé effectué, la fréquence du cosinus généré.

7.3 Séance 2 : Réalisation d'un filtrage de type passe-bas

7.3.1 Génération du signal à filtrer

1. Générez 100 échantillons d'une somme de deux cosinus d'amplitude 1 (V), de fréquences $f_1 = 1000$ Hz et $f_2 = 3000$ Hz, échantillonnés à $F_e = 10000$ Hz.
2. Tracez le signal obtenu avec une échelle temporelle en secondes.
3. Tracez une représentation fréquentielle du signal obtenu avec une échelle fréquentielle en Hz (module de la TFD ou DSP). Vous devez retrouver les fréquences des deux cosinus présents dans le signal somme.

7.3.2 Synthèse du filtre passe-bas

1. Déterminez la réponse impulsionnelle du filtre passe-bas permettant de ne conserver que le cosinus de fréquence $f_1 = 1000$ Hz.
2. Implantez cette réponse impulsionnelle sous Matlab et tracez là pour un ordre du filtre égal à 11 et pour un ordre du filtre égal à 61. Vous devez être capable d'expliquer la différence obtenue entre les deux tracés.
3. Estimez la réponse en fréquence du filtre passe-bas synthétisé avec un ordre de 11 et celle du filtre passe-bas synthétisé avec un ordre de 61. Tracez le module de ces deux réponses en fréquence. Vous devez être capable d'expliquer la différence obtenue entre les deux tracés.

7.3.3 Réalisation du filtrage

1. Superposez, sur une même figure, le tracé de la représentation fréquentielle (module de la TFD ou DSP) du signal somme des deux cosinus et celui du module de la réponse en fréquence d'un des deux filtres passe-bas synthétisés auparavant. Ce tracé doit vous permettre de déterminer si le filtre que vous avez synthétisé va correctement filtrer le cosinus souhaité.

Pour superposer plusieurs tracés sur une même figure vous pouvez utiliser *hold on* après le premier tracé.

Lorsque vous avez plusieurs tracés sur une même figure vous pouvez ajouter une légende permettant de les identifier en utilisant la fonction *legend.m* de Matlab.

Pensez à utiliser le zero padding afin d'obtenir des vecteurs de même taille à tracer sur la même figure.

2. En utilisant la fonction *filter.m* de Matlab, filtrez le signal somme des deux cosinus en utilisant chaque filtre passe-bas synthétisé auparavant (ordre égal à 11 et ordre égal à 61) et visualisez les deux signaux obtenus.

Vous devez également être capables d'identifier les paramètres à envoyer à la fonction *filter.m* de Matlab afin de réaliser le filtrage.

Vous devez être capables de comprendre la différence entre les signaux obtenus et le signal somme des deux cosinus, ainsi que la différence entre les deux signaux obtenus.

Chapitre 8

Projet (séances 3 à 7)

Réalisation d'un modem de fréquence selon la recommandation V21 de l'Union Internationale des Télécommunications (UIT)

8.1 Introduction

Un modem (acronyme pour modulateur/démodulateur) est utilisé, d'une part, pour transformer une information numérique à transmettre en un signal susceptible de se propager sur le support physique à traverser entre l'émetteur et le récepteur (modulation) et, d'autre part, pour retrouver, à partir du signal abîmé par le lien physique de communication, l'information binaire transmise (démodulation).

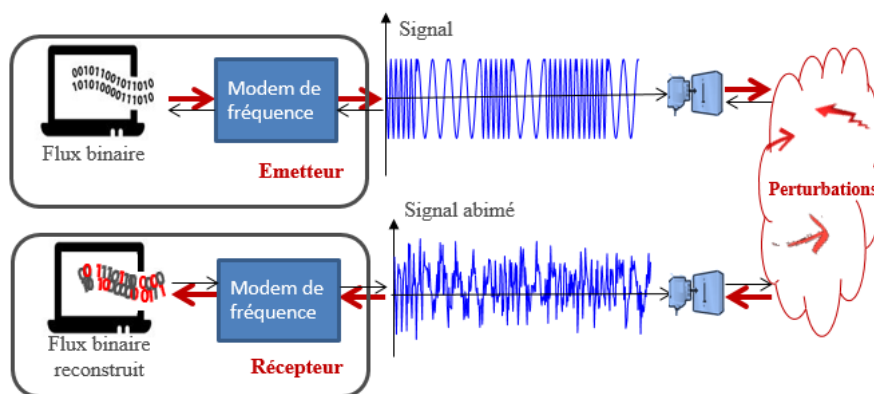


FIGURE 8.1 – Opération de modulation/démodulation

Le modem à implanter dans ce projet devra suivre la recommandation V21 du CCIT (Comité Consultatif International de Téléphonie et de Télégraphie, renommé depuis 1990 Union Internationale des Télécommunications (UIT)). Il s'agit d'un modem travaillant, avec pour support de communication la ligne téléphonique, à un débit maximal de 300 bits par secondes (également exprimé en bauds, en hommage à Emile Baudot (11 septembre 1845 - 28 mars 1903), célèbre ingénieur français ayant oeuvré dans les télécommunications. Ce modem est bien évidemment un

vieux modem mais la technique de transmission utilisée, la modulation de fréquence numérique, reste présente dans de nombreux systèmes de transmissions à l'heure actuelle, dont, par exemple, l'internet des objets (technologie LoRa), le bluetooth ou bien la télémessure/télécommande par satellite. Elle présente, en effet, l'avantage de générer des signaux à enveloppes constantes, ce qui est très intéressant notamment en présence de non linéarités dans une chaîne de transmission (amplificateurs embarqués travaillant à saturation par exemple). Plusieurs versions de modulations de fréquence existent : FSK (Frequency Shift Keying), CP-FSK (Continuous Phase FSK), GMSK (Gaussian Minimum Shift Keying) ou plus généralement les CPM (Continuous Phase Modulation). La modulation de fréquence présente dans la recommandation V21 est une modulation de fréquence à 2 états (2-FSK). Les bits 0 et 1 à transmettre sont transformés en morceaux de cosinus de fréquences différentes pour former le signal transmis (voir figure 8.2).

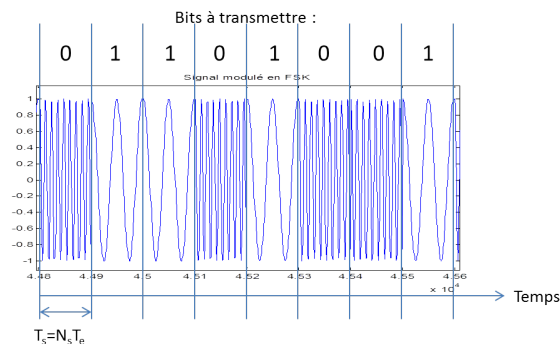


FIGURE 8.2 – Signal modulé en fréquence

Le modem V21 fonctionne en mode duplex. Nous simulerons dans ce projet la voie de transmission n°1 (appel). La fréquence moyenne nominale est de $F_c = 1080$ Hz. L'excursion de fréquence de $\pm\Delta f = \pm 100$ Hz. La fréquence la plus élevée ($F_0 = F_c + \Delta f = 1180$ Hz) devra correspondre à l'état binaire 0, tandis que la plus faible ($F_1 = F_c - \Delta f = 980$ Hz) devra correspondre à l'état binaire 1. La première étape consistera à former le signal 2-FSK à transmettre à partir d'un fichier d'information binaire (modulation). A partir d'un signal reçu abimé par un canal à bruit additif blanc et Gaussien (canal AWGN : Additive White Gaussian Noise), il s'agira ensuite de mettre en place le modem de réception permettant de retrouver l'information binaire. Plusieurs versions du récepteur devront être implantées. Une première version procédera par filtrage. Cette version nécessitant l'utilisation de fréquences suffisamment éloignées pour coder les bits 0 et 1, elle n'est pas utilisable dans le contexte de la norme V21 souhaité. On implantera donc une deuxième version de récepteur étant capable de travailler avec des fréquences proches (occupation spectrale plus faible pour le signal généré).

8.2 Information binaire à transmettre

Afin de tester vos codes vous utiliserez, dans un premier temps, une suite de bits à transmettre (bits 0 et 1 aléatoires et indépendants) générés grâce à la fonction *randi.m* de Matlab.

Vous pourrez ensuite appliquer vos codes à l'information qui vous sera fournie, par binôme, dans un fichier *donnees.mat*. Chaque fichier contiendra un message binaire à transmettre dans la variable *bits* que vous pourrez placer dans votre espace de travail en utilisant la commande *load donnees.mat*. Une fois transmis en utilisant le modem réalisé, vous accéderez en réception à un morceau d'image (voir comment par la suite). L'ensemble des morceaux d'images décodés dans votre groupe de TP et correctement repositionnés vous permettra de retrouver un lieu et

un personnage.

8.3 Modem de fréquence - Démodulation par filtrage

8.3.1 Construction du signal modulé en fréquence

La première étape du projet va consister à réaliser la modulation de fréquence, i.e. transformer l'information binaire à transmettre en un signal modulé en fréquence (exemple sur la figure 8.2). Pour cela, on codera chacun des bits à transmettre par un morceau de cosinus durant T_s secondes, où $T_s = N_s T_e$, N_s étant le nombre d'échantillons distant de T_e formant chaque morceau de cosinus et $T_e = 1/F_e$ la période d'échantillonnage. Afin de pouvoir réaliser la démodulation par filtrage on utilisera un morceau de cosinus à la fréquence $F_0 = 6000$ Hz pour coder les bits 0 et un morceau de cosinus à la fréquence $F_1 = 2000$ Hz pour coder les bits 1. La fréquence d'échantillonnage F_e sera fixée à 48 kHz.

La figure 8.3 présente un exemple de signal NRZ généré à partir d'une information binaire à transmettre, avec $N_s = 3$ échantillons distants de T_e par niveau 0 ou 1 de durée T_s .

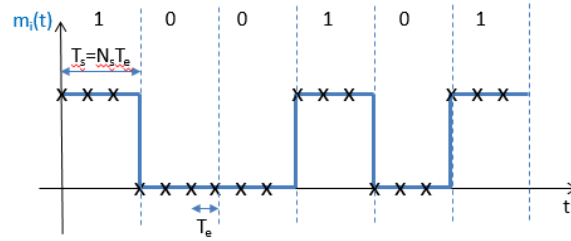


FIGURE 8.3 – Exemple de signal NRZ polaire

Le signal modulé en fréquence $x(t)$ sera alors généré de la manière suivante :

$$x(t) = (1 - NRZ(t)) \times \cos(2\pi F_0 t + \phi_0) + NRZ(t) \times \cos(2\pi F_1 t + \phi_1) \quad (8.1)$$

où $NRZ(t)$ est un signal de type NRZ polaire formé à partir de la suite de bits à transmettre en codant les 0 et les 1 par des niveaux 0 et 1 de durée T_s secondes. ϕ_0 et ϕ_1 sont des variables aléatoires indépendantes uniformément réparties sur $[0, 2\pi]$ qui peuvent être obtenues sous matlab en utilisant $rand*2*pi$.

Génération du signal NRZ

1. Générez le signal NRZ à partir de la suite de bits à transmettre. Le nombre d'échantillons N_s à générer par niveau 0 ou 1 de durée T_s sera déduit du débit souhaité pour la transmission (300 bits par secondes).
2. Tracez le signal NRZ avec une échelle temporelle en secondes et vérifiez que le débit souhaité pour la transmission est le bon.
3. Estimez puis tracez la densité spectrale de puissance du signal NRZ, en utilisant un périodogramme. L'échelle fréquentielle devra être en Hz. La densité spectrale de puissance théorique d'un tel signal est donnée par (voir cours de télécommunications pour la démonstration) :

$$S_x(f) = \frac{1}{4} T_s \text{sinc}^2(\pi f T_s) + \frac{1}{4} \delta(f)$$

Génération du signal modulé en fréquence

1. A partir du signal NRZ précédent, et de la génération de deux cosinus numériques aux fréquences souhaitées, générez le signal modulé en fréquence $x(t)$ définie par l'équation 8.1.
2. Tracez le signal $x(t)$ avec une échelle temporelle en secondes.
3. Calculez la densité spectrale de puissance du signal modulé en fréquence $x(t)$ en fonction de celle du signal NRZ que l'on notera $S_{NRZ}(f)$.
4. Estimez puis tracez la densité spectrale de puissance du signal modulé en fréquence, en utilisant un périodogramme et avec une échelle fréquentielle en Hz. Commentez le résultat obtenu en le comparant à l'expression théorique précédente.

8.3.2 Canal de transmission à bruit additif, blanc et Gaussien

Nous allons considérer que le canal de propagation ajoute au signal émis un bruit que l'on suppose blanc et Gaussien et qui modélise les perturbations introduites.

Vous simulerez ce canal de transmission en ajoutant au signal modulé en fréquence un signal gaussien généré grâce à la fonction *randn.m* de Matlab. Afin de générer N échantillons d'un signal gaussien de moyenne nulle et de puissance égale à σ^2 vous devez l'utiliser de la manière suivante $\sigma * \text{randn}(1, N)$.

La puissance du bruit Gaussien à ajouter devra être déduite du rapport signal sur bruit (SNR : Signal to Noise Ratio) souhaité pour la transmission donné en dB :

$$SNR_{dB} = 10 \log_{10} \frac{P_x}{P_b}$$

où P_x représente la puissance du signal modulé en fréquence et P_b la puissance du bruit ajouté. La puissance d'un signal aléatoire $x(t)$ étant donnée par $E[|x(t)|^2]$, elle peut être obtenue sous Matlab en utilisant $P_x = \text{mean}(\text{abs}(x).^2)$, si x est le vecteur contenant les échantillons du signal. Vous pouvez, dans un premier temps, fixer le rapport signal sur bruit à 50dB, puis vous le ferez évoluer afin de visualiser son impact sur le taux d'erreur binaire obtenu pour la transmission et sur la qualité du morceau d'image restitué.

8.3.3 Démodulation par filtrage

La figure 8.4 présente le récepteur que vous devez implanter pour retrouver, à partir du signal modulé en fréquence bruité, le message binaire envoyé.

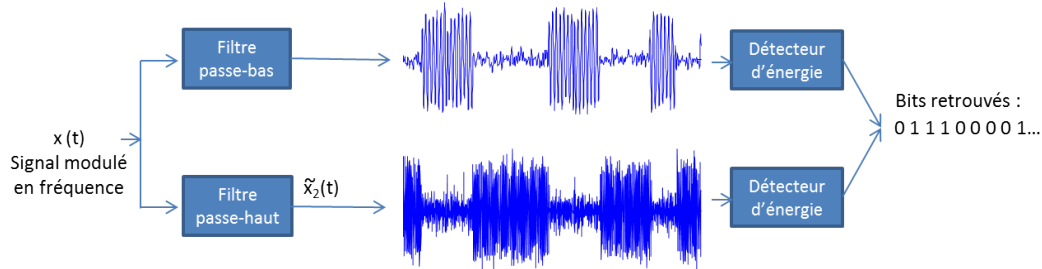


FIGURE 8.4 – Démodulation par filtrage.

Un filtre passe-bas permet de filtrer les morceaux de cosinus à la fréquence $F_0 = 6000\text{Hz}$, tandis qu'un filtre passe-haut permet de filtrer les morceaux de cosinus à la fréquence $F_1 = 2000\text{Hz}$. Une détection d'énergie réalisée tous les T_s secondes permet de récupérer, à partir des signaux filtrés, les bits 0 et 1 transmis.

Remarque : un seul des deux filtres est bien entendu nécessaire pour démoduler votre signal. Néanmoins nous vous demanderons d'implanter les deux.

Synthèse du filtre passe-bas

Vous pouvez utiliser le filtre passe-bas réalisé lors du TP2, en adaptant ses paramètres afin de récupérer le signal permettant de retrouver les 1 émis.

Synthèse du filtre passe-haut

La réponse en fréquence d'un filtre passe-haut idéal est donné par : $H_{I_{PH}}(\tilde{f}) = 1 - H_{I_{PB}}(\tilde{f})$, où $H_{I_{PB}}(\tilde{f})$ représente la réponse en fréquence du filtre passe-bas idéal de même fréquence de

coupure et $\tilde{f} = \frac{f}{F_c}$ la fréquence normalisée. Utilisez cette expression pour déterminer la réponse impulsionnelle idéale d'un filtre passe-haut : $h_{IPH}(n) = TFD^{-1} [H_{IPH}(\tilde{f})]$. Une fois cette réponse impulsionnelle idéale obtenue, il vous restera à adapter ses paramètres afin de pouvoir l'implanter de manière à retrouver le signal permettant de récupérer les 0 émis en sortie du filtre.

Filtrage

Procédez aux filtrages (passe-bas et passe-haut) du signal reçu en utilisant la fonction *filter.m* de Matlab. Attention aux retards introduits par les filtres qui doivent être pris en compte afin de ne pas perdre d'information et de synchroniser correctement votre détecteur d'énergie (voir plus loin).

Tracés à réaliser

Attention les signaux devront être tracés avec une échelle temporelle en secondes et les densités spectrales de puissance avec une échelle fréquentielle en Hz. Pour chaque voie du récepteur :

1. Tracez la réponse impulsionnelle et la réponse en fréquence du filtre réalisé.
2. Tracez, sur un même graphique, la densité spectrale de puissance du signal modulé en fréquence reçu, $x(t)$, et la réponse en fréquences du filtre implanté. Commentez le résultat obtenu.
3. Tracez le signal en sortie du filtre ainsi que sa densité spectrale de puissance. Commentez le résultat obtenu.

Détection d'énergie

Afin de retrouver les bits émis à partir des signaux filtrés, on va utiliser un détecteur d'énergie.

1. Chaque signal filtré sera divisé en périodes temporelles de durée T_s secondes. Sur chaque tranche de signal, notée X et constituée de N_s échantillons : $X = \{x_1, \dots, x_{N_s}\}$, un calcul d'énergie devra être effectué et comparé à un seuil.

Si on considère, par exemple, le signal filtré par le passe-bas, on aura la condition suivante :

$$\text{Si } \sum_{n=1}^{N_s} x_n^2 > K \text{ alors bit 1 sinon bit 0}$$

où K représente le seuil qui sera fixé expérimentalement après observation de l'énergie calculée sur les différentes périodes T_s de signal.

2. Une fois le message binaire retrouvé vous pourrez :
 - Calculer le taux d'erreur binaire obtenu : nombre de bits erronés sur nombre de bits transmis. Bien entendu ce taux d'erreur binaire dépendra du rapport signal sur bruit considéré dans votre transmission, ce que vous pourrez constater en le modifiant.
 - Utiliser la fonction *reconstitution_image.p* qui vous est fournie, sur la suite binaire retrouvée en réception (notée *suite_binaire_reconstruite*), de la manière suivante :
`code_reconstitution_image ;`
`reconstitution_image(suite_binaire_reconstruite) ;`
`which_reconstitution_image ;`
 afin d'afficher le morceau d'image qui vous a été fournie sous forme binaire dans le fichier *donnees.mat*. L'ensemble des morceaux d'images décodés dans votre groupe de

TP et correctement repositionnés vous permettra de retrouver une image complète faisant référence à un lieu et à un personnage. Qui et où ?

8.3.4 Application de la recommandation V21

Testez la chaîne de transmission implantée en utilisant les fréquences préconisées dans la recommandation V21, à savoir $F_0 = 1180$ Hz et $F_1 = 980$ Hz.

- Retrouvez-vous un taux d'erreur binaire nul lorsqu'il n'y a aucun bruit (ou pour un rapport signal sur bruit suffisamment élevé) ?
- Est-il possible de modifier les filtres (fréquence de coupure, nombre de coefficients, fenêtre de troncature) pour retrouver un taux d'erreur binaire nul lorsqu'il n'y a aucun bruit (ou pour un rapport signal sur bruit suffisamment élevé) ? Si oui réalisez cette nouvelle implantation, si non expliquez ce qui vous en empêche.

8.4 Modem de fréquence V21 - Démodulateur FSK

8.4.1 Démodulateur FSK - Contexte de synchronisation idéale

Nous vous proposons d'implanter ici le modem qui fonctionne avec les fréquences préconisées dans la recommandation V21, à savoir $F_0 = 1180$ Hz et $F_1 = 980$ Hz, en supposant que la synchronisation entre l'émetteur et le récepteur est parfaite. L'émetteur reste le même à l'exception des fréquences F_0 et F_1 . Le canal de propagation reste le même (ajout d'un bruit additif blanc et gaussien sur le signal émis). Le récepteur, par contre, va devoir être modifié afin de démoduler correctement les données. La figure 8.5 présente le récepteur que vous devez implanter pour retrouver, à partir du signal modulé en fréquence suivant la recommandation V21, le message binaire envoyé.

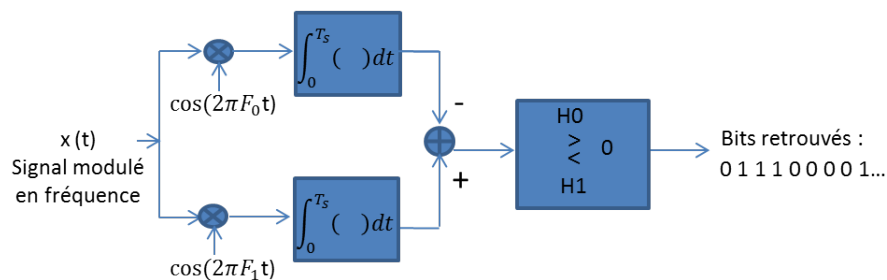


FIGURE 8.5 – Démodulation FSK. Synchronisation supposée idéale.

1. Expliquez le principe de fonctionnement de ce récepteur.
2. Implantez le démodulateur proposé et vérifiez qu'il fonctionne correctement. Vous devez retrouver un taux d'erreur binaire nul sans bruit, ou pour un rapport signal sur bruit suffisamment élevé, et un morceau d'image de bonne qualité si vous utilisez les données binaires fournies dans le fichier *donnees.mat*.

8.4.2 Démodulateur FSK avec gestion d'une erreur de synchronisation de phase porteuse

Le problème de la synchronisation entre l'émetteur et le récepteur est un problème important lorsque l'on réalise une transmission. Les deux doivent être parfaitement synchronisés en temps et en fréquence pour que le démodulateur idéal implanté précédemment fonctionne, ce qui en pratique n'est bien entendu pas possible.

1. Introduisez, dans le modem implanté précédemment, une erreur de phase porteuse, c'est-à-dire utilisez en émission et en réception des cosinus de phases différentes. Votre modem continue-t-il à fonctionner ? Si ce n'est pas le cas expliquez pourquoi.
 2. Afin que votre modem puisse continuer à fonctionner en présence d'une erreur de phase porteuse, celui-ci doit être modifié. La figure 8.6 présente un démodulateur permettant de s'affranchir de problèmes de synchronisation de phase entre les oscillateurs d'émission et de réception.
- (a) Expliquez le principe de fonctionnement de ce nouveau récepteur : en quoi la modification apportée permet-elle de retrouver l'information même s'il existe une erreur de phase entre les cosinus d'émission et les cosinus et sinus de réception ?
- (b) Implantez ce nouveau démodulateur et testez le en présence d'une erreur de phase entre les oscillateurs d'émission et de réception pour vérifier que le taux d'erreur binaire obtenu sans bruit est bien égal à 0.

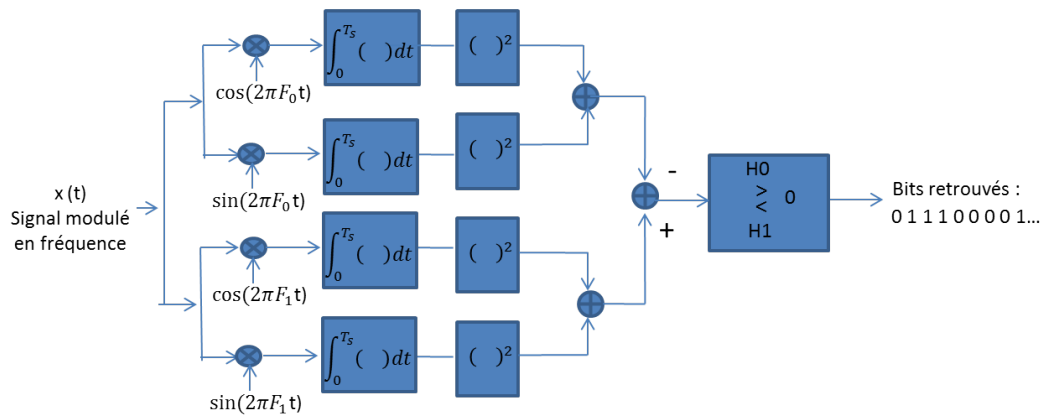


FIGURE 8.6 – Démodulation FSK - Gestion d'une erreur de phase porteuse.

8.5 Evaluation

Le travail sera réalisé en binôme et nous vous demanderons de déposer vos codes et un rapport au format pdf sur moodle, dans l'espace réservé à votre groupe de TP et prévu pour cela, avant le 22/01/2021.

8.5.1 Le rapport

1. Comme tout rapport, il devra comporter un sommaire, une introduction présentant les objectifs des TPs, une conclusion synthétisant les principaux résultats obtenus et une bibliographie comprenant les références éventuellement utilisées. On pourra y ajouter une table des illustrations.
2. Les équations devront être réalisées avec un éditeur d'équation.
3. Lorsque vous commentez une figure vous devez y faire référence dans votre texte : par exemple "comme le montre la figure 1, ..."
4. Tous vos tracés doivent comporter des labels sur les axes et un titre.
5. Si plusieurs courbes sont tracées sur la même figure, celle-ci devra comporter une légende permettant de les différencier.
6. Toutes vos explications/justifications/commentaires devront utiliser les bons termes techniques (provenant des cours/TDs/TPs, des livres/sites consultés et cités), pas d'à peu près. "En gros" est à proscrire...
7. Attention votre rapport doit être relu, éventuellement passé au correcteur orthographique et grammatical.

Un fichier exemple au format latex vous est fourni. Vous pouvez l'utiliser en le complétant, en le modifiant comme vous le souhaitez. Il vous donne quelques exemples pour réaliser un plan, l'écriture d'équations, l'insertion et l'appel à figure, l'appel à section ... Vous pouvez, par exemple, utiliser texmaker qui est gratuit pour l'éditer et le compiler.

8.5.2 Les codes

1. Vos codes doivent être commentés de manière suffisante et claire. Un nouvel utilisateur doit pouvoir comprendre ce que vous avez souhaité implanter.
2. Vous pouvez regrouper l'ensemble du code du projet dans un seul fichier .m. Si vous en fournissez plusieurs, un mode d'emploi devra être fourni dans le rapport pour savoir ce qui doit être lancé pour réaliser les différentes fonctions implantées.

Chapitre 9

Références

- [1] : "Traitement numérique du signal, théorie et pratique", M. Bellanger, Masson, collection CENT-ENST.
- [2] : "Traitement numérique des signaux", M. Kunt, Dunod, Traité d'électricité, d'électronique et d'électrotechnique.
- [3] : "Traitement numérique du signal, Une introduction", A.W.M. Van Den Enden et N.A.M. Verhoeckx, Masson.
- [4] : "Introduction au traitement du signal", P. Duvaut, F. Michaut, M. Chuc, Hermes, Collection traitement du signal.
- [5] : Documents sur la variable complexe, la transformée de Laplace et la transformée en z : <http://dobigeon.perso.enseiht.fr/teaching/complex.html>
- [6] : "Introduction to digital filters, with audio applications", J.O. Smith, BookSurge, 2007. [7] : "Digital signal processing : fundamentals and applications", Tan Li, Jiang Jean, Elsevier, 2013.
- [8] : Alan V. Oppenheim, Ronald W. Schaffer et J. R. Buck, Discrete-time signal processing, Upper Saddle River, N.J., Prentice Hall, 3rd edition, 2009. [9] : Signal and Systems, by Simon Haykin and Barry Van Veen, Wiley, 2nd edition, 2002. [10] : John G. Proakis, Dimitri G. Manolakis, Digital Signal Processing : Principles, Algorithms and Applications, Pearson Education, 4th edition, 2006.