Traitement Numérique du Signal

Électronique et Informatique Industrielle 2^{nde} année - EII2

Olivier SENTIEYS

ENSSAT - Université de Rennes 1

15 septembre 2003

ENSSAT
6 Rue de Kerampont - BP 447
22305 LANNION - France

Téléphone : 02-96-46-66-41
Télécopie : 02-96-46-66-75
sentieys@enssat.fr
http ://www.irisa.fr/R2D2

IRISA — ENSSAT
Institut de Recherche en Informatique et Systèmes Aléatoires
École Nationale Supérieure de Sciences Appliquées et de Technologie
Technopôle Anticipa Lannion

Table des matières

Introduction

Le traitement numérique du signal est une notion qu'il n'est pas facile de définir simplement étant donné le nombre important d'applications relevant de cette discipline. En première approximation, on peut tout d'abord tenter d'expliciter chacun des mots de cette expression.

Traitement signifie que l'on est en présence d'un processus de séquencement d'opérations programmées. Une séquence d'opérations s'applique ici à une suite de données *Numériques* qui vont représenter sous une forme discrète un paramètre variable, ou *Signal*, qui le plus souvent est extérieur au processus de traitement.

Il s'agit donc d'appliquer un traitement ou une analyse de l'information à une séquence de nombres discrets qui représente un signal provenant pour la majorité des applications du monde physique qui nous entoure.

Le traitement numérique d'un signal nécessite un support matériel permettant d'effectuer le traitement de l'information, ce peut-être du matériel électronique spécifique à une tâche particulière ou du matériel moins spécialisé comme peut-l'être un ordinateur.

Il faut pouvoir communiquer entre le monde physique extérieur et le processus par lequel s'effectue le traitement ; le signal extérieur, s'il est défini sur un support continu, doit être représenté sous une forme discrète. Cela veut dire que l'on accepte de perdre de l'information : entre deux valeurs consécutives et discrètes du signal nous faisons l'hypothèse de ne disposer d'aucune autre information. Nous verrons qu'une analyse du problème à traiter permet de définir correctement ce qui est conservé ou non dans la forme numérique d'un signal.

Il faut bien sûr définir la séquence des opérations qui transforme le signal numérique. Cette opération correspond à un objectif de traitement bien précis, par exemple supprimer l'écho des lignes téléphoniques, reconnaître une signature radar, etc...

Le développement de l'électronique permit le traitement du signal analogique ; un signal électronique *analogique* suit continuement le signal physique qui lui est relié via un capteur. C'est le développement des calculateurs numériques qui a conduit à l'essor du traitement numérique du signal. C'est en effet une approche souple - les traitements correspondent à des logiciels, les supports matériels sont polyvalents -, les états discrets d'un calculateur sont stables - ce n'est en effet pas le cas des systèmes analogiques fortement sensibles aux dérives dues par exemple aux conditions de température ou aux problèmes de vieillissement -. Cependant, l'approche numérique peut se révéler parfois complexe pour des applications très simples. La quantité d'information pouvant être traitée est corrélée à la vitesse de calcul ; la réalisation d'un traitement sous forme analogique ira toujours plus vite que par une forme numérique avec un calculateur et du logiciel.

Remarques sur la notation

Dans la suite du document, on notera par f la fréquence en Hz, f_e la fréquence d'échantillonnage, T la période d'échantillonnage et $\omega = 2\pi f$ la pulsation en rad/s. Pour des raisons de simplicité d'expression, on peut faire abstraction de T si on sait que toutes les grandeurs utilisées seront relatives à T ou f_e.

Pour ces raisons, on peut utiliser dans le domaine fréquentiel, la variable $\Omega = 2\pi f.T = 2\pi f/f_e$ appelée pulsation relative. Un filtre numérique exprimé selon la variable Ω (ou un signal discret) sera donc périodique de période 2π (c'est à dire f_e) (voir figure 6.7) et son gabarit sera défini entre 0 et π (c'est à dire entre 0 et $f_e/2$).

Utilisation de Matlab et Scilab

Matlab, Acronyme de *Matrix Laboratory*, est un environnement logiciel interactif puissant dédié au calcul numérique et à la visualisation ; il est très utilisé dans les divers domaines des sciences pour l'ingénieur, tant pour l'analyse que la conception. Il existe également un nombre important de *toolboxes* qui étendent les possibilités de Matlab à divers domaines spécialisés au moyen de fonctions supplémentaires : traitement du signal, automatique, traitement d'images, optimisation, réseaux de neurones, logique floue *etc...* Il faut considérer Matlab comme étant avant tout un outil de calcul matriciel.
Scilab (http ://www-rocq.inria.fr/scilab) est un équivalent libre de droit de Matlab qui se révèle donc intéressant si on ne possède pas de licence Matlab. Même si elle est proche dans sa philosophie, la syntaxe des commandes pour l'utilisation de Scilab n'est pas totalement compatible avec Matlab.

La plupart des exemples ou figures utilisés ont été réalisés sous Matlab ou Scilab et seront illustrés sous Matlab à travers des exemple au cours du document afin d'effectuer l'apprentissage de ce type de logiciel, aujourd'hui indispensable pour tout traiteur de signaux.

Remerciements

Merci à Michel Corazza, Daniel Ménard, Hervé Chuberre et Olivier Boëffard pour leur aide dans l'élaboration de ce document.

Chapitre 1

Signaux et Systèmes

Un *signal* est une quantité physique mesurable qui évolue en fonction d'une ou de plusieurs variables comme par exemple le temps ou des variables d'espace. Souvent on réferre un signal à la représentation mathématique de la quantité physique observée. Un signal correspond le plus souvent à une modélisation du comportement de la quantité physique observable. Cependant, il peut être extrêmement difficile d'obtenir une forme mathématique simple et concise pour un signal donné.
Un *système* est une entité physique qui réalise une opération sur un signal. Un système définit donc un signal d'entrée et un signal de sortie ; le signal de sortie correspond à la transformation opérée par le système sur le signal d'entrée. Par exemple, l'oreille humaine est un système transformant un signal correspondant à une variation de pression acoustique en des séquences parallèles de signaux électriques sur le nerf auditif. Un microphone est un système un peu analogue au précédent (en première approximation très réductrice...) dans la mesure où une variation de pression acoustique est transformée en un signal électrique monodimensionnel.
L'étude de tels systèmes conduit à analyser les transformations entre signaux d'entrée et de sortie pour des systèmes plus ou moins complexes ; cette activité est appelée *traitement du signal*. On ne parlera ici que du traitement des signaux numériques.

1.1 Système numérique de traitement du signal

Historiquement, le traitement du signal tel que défini dans l'introduction précédente fut d'abord de type analogique. En pratique, les signaux manipulés étaient des tensions ou des courants.
Un système numérique de traitement du signal peut vivre dans un monde purement numérique. Par exemple, la cotation des valeurs d'une bourse peut être vue comme un ensemble de signaux numériques.
Cependant la majorité des opérations en traitement du signal ont lieu sur des signaux analogiques, qu'il faut donc *convertir* sous une forme numérique pour que l'on puisse leur appliquer des opérations numériques. Dans la majorité des cas, il est tout aussi indispensable de convertir le signal numérique d'un traitement numérique en un signal analogique.
Ces opérations de conversion analogique/numérique, A/N, et numérique/analogique, N/A, sont les interfaces entre un monde physique et le monde du calculateur où s'exécutent les algorithmes de traitement du signal. Dans la chaîne du traitement du signal, ces interfaces de conversions sont le talon d'Achille de ces systèmes ; elles limitent la vitesse et la précision

des systèmes de traitement. La définition technologique d'une interface de conversion A/N et N/A est toujours un compromis coût performance.

Hormis la difficulté du passage analogique/numérique, les systèmes TNS ont de sérieux avantages sur leurs équivalents analogiques.
– Flexibilité, utilisation d'algorithmes sur des calculateurs.
– Précision et consistence des calculs numériques à comparer aux dérives des systèmes analogiques (tolérance des composants).
– Capacité de stockage, transmission sans altération du signal.

1.2 Classification des signaux

On rappelle qu'un signal est une fonction dépendant d'une ou de plusieurs variables. Par exemple soit le signal : $s(t)$, s est une quantité dépendant d'un paramètre t (par convention, on utilisera la lettre t pour la variable temps).
Un signal peut être classé selon différents critères : sa dimensionnalité, ses caractéristiques temporelles, les valeurs qu'il peut prendre, sa prédictibilité.

1.2.1 Dimensionnalité

On peut tenir compte de ce critère de deux manières différentes : la dimension du signal et les dimensions des variables du signal.
Considérons tout d'abord ce critère de classification comme étant la dimension de l'espace des valeurs prises par le signal (ou la fonction mathématique modélisant le signal).
On distingue alors :
– le signal scalaire, ou signal monocanal pouvant prendre des valeurs réelles ou complexes.
– le signal vectoriel, ou signal multicanal pouvant prendre des valeurs réelles ou complexes.
Prenons par exemple un signal de Télévision (TV). Si on s'intérresse aux trois couleurs constituant une image, ce signal TV prend des valeurs dans un espace à trois dimensions, une première pour le rouge, une seconde pour le vert et enfin une troisième pour le bleu ; $[R, V, B] = TV(t)$.
Par contre, si on s'interresse maintenant à la luminance, ce signal prend ses valeurs dans un espace à une dimension ; $[L] = TV(t)$.
On peut aussi considérer ce critère de classification comme la dimension du domaine de la fonction signal, c'est-à-dire, le nombre d'arguments pris par cette fonction.
On distingue alors :
– Le signal mono-dimensionnel qui correspond à des fonctions à un seul argument, comme par exemple le temps.
– Le signal multi-dimensionnel qui correspond à des fonctions à plusieurs arguments.
Le signal TV correspondant à la luminance peut être fonction du temps mais aussi des variables cartésiennes correspondant à un point de l'écran ; $[I] = TV(t, x, y)$. Il s'agit d'un signal tridimensionnel.
Les signaux abordés dans ce cours seront mono-dimensionnels fonction d'une variable que l'on considérera comme le temps. Toutes les techniques de traitement du signal se généralisent assez bien aux signaux vectoriels et multidimensionnels (voir le cours sur le traitement d'images).

1.2.2 Caractéristiques temporelles

On suppose un signal scalaire $s(t)$. On distingue alors :
- Les signaux à temps continu ou signaux *analogiques*. La variable $t \in \mathbb{R}$. On notera un signal analogique de la façon suivante : $s_a(t)$.
- Les signaux à temps discret : ces signaux sont *définis* pour certaines valeurs de la variable t.

On peut représenter un signal à temps discret par une séquence indicée de la variable t :

$$t_n, \quad n = \cdots 0, -2, -1, 0, 1, 2, \cdots$$

t_n précise un instant pour lequel le signal est défini. Attention, cela ne veut pas dire que le signal est nul *entre* deux instants ; il n'est tout simplement pas défini.
On s'intéressera ici à une répartition uniforme des instants t_n que l'on peut noter $t_n = nT$ où T est l'espace temporel entre deux échantillons consécutifs. On peut alors employer $s(n)$ ou s_n comme notation simplifiée.
On a alors les relations suivantes :

$$s_n = s(n) = s_a(t_n) = s_a(nT)$$

1.2.3 Valeurs prises par le signal

On suppose un signal scalaire $s(t)$. On distingue alors :
- Les signaux à valeurs continues pouvant prendre une valeur réelle dans un intervalle continue (par exemple, une tension ou un courant électrique).
- Les signaux à valeurs discrètes prenant seulement des valeurs parmi un ensemble fini de valeurs possibles.

Un signal numérique est un signal à temps discret et à valeurs discrètes. L'opération de discrétisation des valeurs continues d'un signal en valeurs discrètes est une *quantification*, notée q par la suite.
Soit par exemple une convertisseur Analogique/Numérique traitant des mots de 8 bits ; un signal quantifié par ce convertisseur prendra une valeur discrète parmi 256 possibles.

1.2.4 Prédictibilité des signaux

On peut distinguer deux grandes classes de signaux selon leur caractère de prédictibilité.
- Les signaux déterministes qui peuvent être représentés explicitement par une fonction mathématique.
- Les signaux aléatoires qui évoluent dans le temps d'une manière imprédictible. Il est cependant possible de décrire mathématiquement certaines caractéristiques statistiques de ces signaux.

On s'intéressera dans ce cours essentiellement aux signaux déterministes.

1.3 Représentation des signaux et systèmes numériques

1.3.1 Les signaux à temps discret

On s'intéresse ici à un signal scalaire monodimensionnel prenant des valeurs continues : $s(n)$. Le signal s prend ses valeurs dans $\mathbb{R}$ ou $\mathbb{C}$, la variable n, qu'on peut supposer être le temps, dans $\mathbb{N}$.

Le signal $s(n)$ est par définition une séquence de valeurs s_n. $s(n)$ n'est pas *défini* pour des valeurs de n non entières. On appelle $s(n)$ le nième *échantillon* de ce signal.
Quelques signaux élémentaires sont utiles pour l'étude des propriétés des systèmes de traitement du signal : l'impulsion unité et l'échelon unité.

1.3.1.1 L'impulsion unité

Il s'agit d'un signal noté $\delta(n)$ tel que :

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

Exemple 1.3.1 : Pour créer une impulsion sous Matlab, il faut tout d'abord décider de la longueur de ce signal. Le programme Matlab suivant va créer un signal *impulsion* de longueur $L = 32$.

```
L = 31;
nn = 0:(L-1);
imp = zeros(L,1);
imp(1) = 1;
stem(nn,imp);   %Trace le signal imp
```

Notons que les indices Matlab vont de 1 à L. Par conséquent, à $\delta(0)$ correspond `imp(1)`.

1.3.1.2 L'échelon unité

Il s'agit d'un signal que l'on notera $u(n)$ tel que :

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

La fonction Matlab équivalente est : `u = ones(L,1) ;`

1.3.1.3 Caractéristiques

On définit l'énergie E d'un signal à temps discret de la manière suivante :

$$E \triangleq \sum_{n=-\infty}^{\infty} |s(n)|^2$$

La puissance moyenne P d'un signal $s(n)$ est définie comme :

$$P \triangleq \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |s(n)|^2$$

Si l'énergie E est finie alors $s(n)$ est un signal *d'énergie finie* et $P = 0$. Si E est infinie alors P peut être soit finie ou infinie, si P est finie et non nulle alors $s(n)$ est un signal de *puissance finie* .

Par exemple un signal continu est un signal d'énergie infinie ($E = \infty$) mais de puissance finie, si a est l'amplitude du signal, $P = a^2$.
Un signal $s(n)$ est périodique de période P si et seulement si $s(n + P) = s(n)\ \forall n$ sinon $s(n)$ est apériodique.
Un signal $s(n)$ est symétrique ou pair si et seulement si $s(-n) = s(n)$. Un signal $s(n)$ est antisymétrique ou impair si et seulement si $s(-n) = -s(n)$. Tout signal peut se décomposer comme la somme d'un signal pair et d'un signal impair.

1.3.2 Les systèmes à temps discret

Un système à temps discret est un système qui transforme un signal d'entrée à temps discret, appelé signal d'excitation, en un signal de sortie à temps discret, appelé signal de réponse.
Un signal d'entrée $e(n)$ est *transformé* en un signal de sortie $s(n)$:

$$s(.) = \mathcal{T}[e(.)]$$

Il faut comprendre la notation précédente comme la transformation de la séquence *complète* $e(n)$. On distingue les systèmes à temps discret *sans effet de mémoire* et les systèmes *dynamiques*. Un système à temps discret sans effet de mémoire est un système pour lequel un échantillon de sortie d'instant n ne dépend que de l'échantillon d'entrée du même instant. Dans tous les autres cas, il y a un effet de mémoire et le système est dit dynamique.
Les systèmes pour lesquels le comportement entre le signal d'entrée et le signal de sortie n'évolue pas en fonction du temps sont faciles à analyser. Un système est dit *invariant* en temps (ou au décalage) si et seulement si :

$$e(n) \xrightarrow{\mathcal{T}} s(n) \qquad \Rightarrow \qquad e(n-k) \xrightarrow{\mathcal{T}} s(n-k) \quad \forall \quad e(.), \quad \forall \quad k \in (N)$$

Un système est *linéaire* si et seulement si :

$$\mathcal{T}[a \times e_1(n) + b \times e_2(n)] = a \times \mathcal{T}[e_1(n)] + b \times \mathcal{T}[e_2(n)] \quad \forall \quad e_1(.) \quad \forall \quad e_2(.) \quad \forall \quad (a, b)$$

Un système linéaire conserve donc l'opérateur d'addition et de multiplication.
Un système est *causal* si la sortie $s(n)$ à n'importe quel instant n dépend *seulement* des échantillons passés et de l'échantillon présent du signal d'entrée.

1.4 Analyse des Systèmes Linéaires Invariants

Il s'agit d'une classe de système largement utilisée en traitement du signal. On suppose un tel système *linéaire*, et *invariant* dans le temps. L'hypothèse de linéarité conduit au principe de superposition dû à la conservation de l'opérateur d'addition par la transformation ; cette hypothèse simplifie grandement les études analytiques des systèmes numériques.
La stratégie générale d'analyse d'un système linéaire invariant est la suivante :

1. Décomposition du signal d'entrée en une somme de signaux ou fonctions de base.

$$e(n) = \sum_k \alpha_k e_k^b(n)$$

2. Etude de la réponse du système pour l'ensemble des fonctions de base.

$$s_k^b(n) = \mathcal{T}[e_k^b(n)]$$

3. Recomposition de la sortie en appliquant le principe de superposition.

$$s(n) = \sum_k \alpha_k s_k^b(n)$$

L'intérêt d'une telle approche est de s'appuyer sur un ensemble de fonctions de base possédant des caractéristiques intéressantes connues (fonction $\delta(n)$ et l'exponentiel complexe par exemple).

1.4.1 Représentation d'un signal

Appliquons le premier point énuméré précédemment en utilisant des signaux δ comme fonction de base. Soit $e(n)$ un signal numérique quelconque que l'on cherche à représenter uniquement avec un ensemble de fonctions $\delta(n)$.
Développons le signal $e(n)$:

$$e(n) = \cdots, e(-2), e(-1), e(0), e(1), e(2) \cdots$$

On rappelle que $\delta(n)$ vaut 1 si $n = 0$, 0 sinon ; une somme de fonction $\delta(n)$ qui ne se recouvre pas sur le même indice vaut soit 1, soit 0.
On peut donc écrire :

$$\begin{aligned}
\vdots &= \vdots \\
e(-1) &= \cdots + 0.e(-2) + 1.e(-1) + 0.e(0) + 0.e(1) + 0.e(2) + \cdots \\
e(0) &= \cdots + 0.e(-2) + 0.e(-1) + 1.e(0) + 0.e(1) + 0.e(2) + \cdots \\
e(1) &= \cdots + 0.e(-2) + 0.e(-1) + 0.e(0) + 1.e(1) + 0.e(2) + \cdots \\
\vdots &= \vdots
\end{aligned}$$

Le signal $e(n)$ s'écrit alors :

$$e(n) = \sum_{k=-\infty}^{+\infty} e(k)\delta(n-k)$$

1.4.2 Système Linéaire Invariant

Soit maintenant un système linéaire (SL) transformant un signal d'entrée $e(n)$ en un signal de sortie $s(n)$:

$$\begin{aligned} s(n) &= \mathcal{T}[e(n)] \\ s(n) &= \mathcal{T}[\sum_{k=-\infty}^{+\infty} e(k)\delta(n-k)] \\ s(n) &= \sum_{k=-\infty}^{+\infty} \mathcal{T}[e(k)\delta(n-k)] \\ s(n) &= \sum_{k=-\infty}^{+\infty} e(k)\mathcal{T}[\delta(n-k)] \end{aligned}$$

On pose :

$$h_k(n) = \mathcal{T}[\delta(n-k)]$$

En plus d'être linéaire, si le système est *invariant*, $h_k(n)$ ne dépend plus de k, donc :

$$h(n) = \mathcal{T}[\delta(n)]$$

Pour un système linéaire invariant (SLI), on obtient alors la relation suivante entre signal d'entrée et de sortie :

$$s(n) = \sum_{k=-\infty}^{+\infty} e(k)h(n-k)$$

Un Système Linéaire Invariant est donc entièrement caractérisé par sa *réponse impulsionnelle* $h(n)$.
Cette opération d'accumulation de termes multiplicatifs porte le nom de *convolution* et se note $*$, on a :

$$s(n) = e(n) * h(n)$$

Par un simple changement de variable sous le signe somme, il est simple de montrer qu'il s'agit d'une opération commutative, on a :

$$s(n) = e(n) * h(n) = h(n) * e(n)$$

1.4.3 Exemple de convolution

Soit la réponse impulsionnelle $h(n)$ d'un système linéaire invariant telle que :

$$h(n) = \begin{cases} a^n & \text{si } n \geq 0 \\ 0 & \text{si } n < 0 \end{cases}$$

ou encore :

$$h(n) = a^n u(n)$$

Ce signal est représenté en haut de la figure 1.1.
Étudions la réponse d'un tel système à l'entrée suivante :

$$e(n) = u(n) - u(n-N) \text{pour } N \text{ fixé}$$

On trouvera au milieu de la figure 1.1 une représentation de ce signal.
On peut distinguer 3 cas :

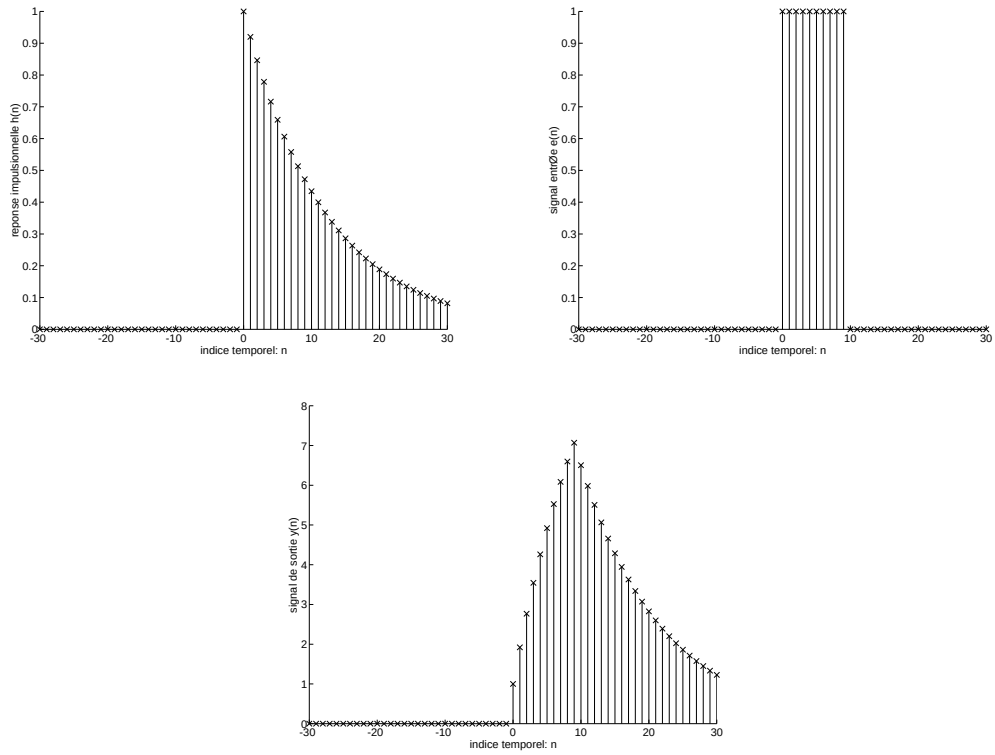


FIG. 1.1: En haut à gauche, réponse impulsionnelle du système, au haut à droite entrée du système, en bas, signal de réponse du système à l'entrée

1. Pour $n < 0$, $h(n-k)$ et $e(n)$ n'ont aucun échantillon non nuls en commun donc $s(n) = 0 \quad \text{pour} \quad n < 0$
2. Pour $0 \leq n < N$, on a un recouvrement d'échantillons non nuls pour $0 \leq k < n$, donc pour $0 \leq n < N$ on obtient :

$$s(n) = \sum_{k=0}^{n} a^{n-k} = a^n \frac{1 - a^{-(n+1)}}{1 - a^{-1}}$$

3. Pour $n \geq N$, on a n recouvrements d'échantillons non nuls pour $0 \leq k < N$, donc pour $n \geq N$ on a :

$$s(n) = \sum_{k=0}^{N-1} a^{n-k} = a^n \frac{1 - a^{-N}}{1 - a^{-1}}$$

La réponse du système est représentée en bas de la figure 1.1.

1.4.4 Stabilité

Un système est stable si à une entrée bornée correspond une sortie bornée. Soit $h(n)$ la réponse impulsionnelle d'un système linéaire invariant, la condition de stabilité d'un tel système s'écrit :

$$\sum_{k=-\infty}^{+\infty} |h(k)| < +\infty$$

Démonstration :
"$\Rightarrow$" :
Soit un signal $e(n)$ borné, c'est-à-dire :

$$\exists M < +\infty \qquad \text{tq} \quad |e(n)| < M \qquad \forall \quad n$$

Par l'opération de convolution (2ème forme) on obtient :

$$|s(n)| = |\sum_{k=-\infty}^{+\infty} h(k)e(n-k)|$$

En utilisant l'hypothèse d'une entrée bornée :

$$|s(n)| \leq M \sum_{k=-\infty}^{+\infty} |h(k)| < +\infty$$

"$\Leftarrow$" :
Il suffit d'un contre-exemple pour valider ce sens de la démonstration. Soit la sortie d'un système Linéaire Invariant supposée bornée, on suppose de plus le système instable, on recherche alors une entrée bornée.
Soit l'entrée :

$$e(n) = \begin{cases} \frac{h^*(-n)}{|h(n)|} & \text{si } h(n) \neq 0 \\ 0 & \text{si } h(n) = 0 \end{cases}$$

Pour $n = 0$ (un échantillon *contre exemple* suffit) on obtient :

$$s(0) = \sum_{k=-\infty}^{+\infty} e(-k)h(k) = \sum_{k=-\infty}^{+\infty} \frac{h^*(k)h(k)}{|h(k)|}$$

$$s(0) = \sum_{k=-\infty}^{+\infty} \frac{|h(k)|^2}{|h(k)|} = +\infty$$

□

1.4.5 Causalité

Un système est *causal* si un changement en sortie ne *précède* pas un changement en entrée. Soient deux signaux d'entrée $e_1(n)$ et $e_2(n)$ ainsi que leurs sorties respectives $s_1(n)$ et $s_2(n)$, un système est causal si et seulement si $\exists n_0$ tel que si $e_1(n) = e_2(n)$ pour $n < n_0$ alors $s_1(n) = s_2(n)$ pour $n < n_0$.
Un système linéaire invariant est causal si et seulement si $h(n) = 0$ pour $n < 0$.
Un séquence est *causale* si les échantillons de cette séquence sont nuls pour $n < 0$.

1.4.6 Equation aux différences finies

Les équations de convolutions développées au cours des paragraphes précédents font intervenir des sommes infinies de termes. Si la réponse impulsionnelle possède un nombre infini de termes il est difficilement envisageable de mettre en oeuvre cette convolution sur un calculateur. Mais, il cependant possible pour certaines classes de réponses impulsionnelles infinies de développer la convolution sous la forme d'une récursion. La relation entre l'entrée et la sortie est une combinaison linéaire à coefficients constants (et en nombre fini) des échantillons d'entrée et de sortie.
Une équation aux différences finies peut s'écrire sous la forme :

$$s(n) = -\sum_{k=1}^{N} a_k s(n-k) + \sum_{k=0}^{M} b_k e(n-k)$$

Un système régit par une équation aux différences finies du type précédent est linéaire, invariant et causal. Si $N \geq 1$:
– un échantillon de sortie au temps n, $s(n)$ dépend des N précédents échantillons de sortie $s(n-1) \cdots (n-N)$,
– un tel système est dit *récursif*,
– la réponse impulsionnelle est infinie, RII, ou IIR en anglais (Infinite Impulse Response).
Si $N = 0$:
– la sortie $s(n)$ dépend seulement de l'entrée courante $e(n)$ et de ses M échantillons précédents,
– le système est dit non-récursif,
– la réponse impulsionnelle est finie, RIF, ou FIR en anglais (Finite Impulse Response).
La réponse impulsionnelle d'un système RIF est donnée par :

$$h(n) = \sum_{k=0}^{M} b_k \delta(n-k)$$

c'est-à-dire par la séquence des $\{b_k\}$.
Puisque qu'étudier un système linéaire invariant revient à étudier sa réponse impulsionnelle, le cas particulier d'une réponse impulsionnelle mise sous la forme d'une équations aux différences finies consiste à étudier un système linéaire d'équations, c'est-à-dire à en extraire les racines du polynôme caractéristique. Ceci sera effectué de manière efficace en utilisant la transformée en Z.

1.5 Représentation fréquentielle

Au cours du paragraphe précédent, nous avons vu que les systèmes linéaires invariants ont des propriétés, notamment le principe de superposition, qui conduisent à des solutions analytiques simples. Si on applique une sinusoïde à l'entrée d'un système LI la sortie est elle aussi sinusoïdale et de même pulsation. Les amplitudes et phases dépendent par contre des caractéristiques du système. Il est donc possible d'analyser le comportement d'un système linéaire invariant par l'observation de l'évolution des paramètres d'une série de sinusoïdes. C'est cette propriété qui rend si intéressante l'utilisation de la transformée de Fourier pour l'étude des systèmes LI. Soit l'entrée $e(n) = e^{jwnT} = e^{j\Omega nT}$ pour $-\infty < n < +\infty$ d'un système linéaire invariant de réponse impulsionnelle $h(k)$. La sortie peut alors s'écrire[1] :

$$s(n) = \sum_{k=-\infty}^{\infty} h(k) e^{j\Omega(n-k)}$$

$$s(n) = e^{j\Omega n} \sum_{k=-\infty}^{\infty} h(k) e^{-j\Omega k}$$

En posant :

$$H(e^{j\Omega n}) = \sum_{k=-\infty}^{\infty} h(k) e^{-j\Omega k}$$

L'équation précédente représente la modification apportée par le système et modélisée par une amplitude complexe.
$H(e^{j\Omega n})$ est appelé *réponse fréquentielle* du système caractérisé par sa réponse impulsionnelle $h(k)$. Il s'agit d'un terme complexe qui peut s'exprimer par une partie réelle ou imaginaire ; on adopte le plus souvent une représentation de type polaire où l'on fait référence au module et à la phase de cette réponse fréquentielle :

$$H(e^{j\Omega n}) = |H(e^{j\Omega n})| e^{jarg[H(e^{j\Omega n})]}$$

La définition de la réponse fréquentielle d'un système linéaire invariant montre qu'il s'agit d'une fonction périodique de période 2π. Cela veut dire qu'à deux entrées identiques mais à une pulsation double le système répond d'une manière identique. Puisque $H(e^{j\Omega n})$ est une fonction périodique, elle peut être représentée par une série de Fourier. En fait l'équation de définition fait apparaître $H(e^{jw\Omega})$ comme une série de Fourier où les coefficients sont $h(k)$.

[1] Voir les notations de la section

Chapitre 2

Transformation en Z

La transformée en Z est un outil largement utilisé pour l'étude des systèmes de traitement numérique du signal. Ce type de transformée permet de décrire aisément les signaux à temps discret et la réponse des systèmes linéaires invariants soumis à des entrées diverses. La transformée en Z est un outil permettant de dériver la réponse impulsionnelle d'un système linéaire invariant décrit par une équation aux différences finies. De plus, à l'opérateur de convolution dans le domaine temporel correspond l'opérateur multiplicatif dans le domaine de la transformée en Z.

2.1 Définition de la transformée en Z

La transformée en Z directe d'un signal à temps discret $x(n)$ est définie par :

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{2.1}$$

La transformée en Z établit une correspondance entre l'espace des signaux à temps discret et un espace de fonctions définies sur un sous-ensemble du plan complexe. On définit le *plan en z* comme étant le plan complexe. La série des puissances introduite dans l'équation de définition précédente ne converge que pour un sous-ensemble du plan complexe. Ce sous-ensemble est appelé *région de convergence* ou *domaine de convergence*. Une région de convergence correspond à l'ensemble des valeurs de z telles que $X(z)$ soit définie et à valeurs finies. Spécifier le domaine de convergence de la transformée est tout aussi important que la transformée elle-même.

2.1.1 Exemples de transformée

Soit le signal à temps discret suivant :

$$x(n) = \delta(n)$$

on a :

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} x(n) z^{-n} \\ X(z) &= z^0, \quad \forall \quad z \in \mathbb{C} \\ X(z) &= 1, \quad \forall \quad z \in \mathbb{C} \end{aligned}$$

Pour ce premier exemple la région de convergence est $\mathbb{C}$.

Soit le signal à temps discret suivant :

$$x(n) = \delta(n-k)$$

on a :

$$X(z) = z^{-k}$$

La région de convergence dépend ici de k. Si $k = 0$, la région de convergence est $\mathbb{C}$ (exemple précédent). Si $k < 0$, $X(z)$ n'est pas à valeur finie pour $z = \infty$, donc la région de convergence est ici $\mathbb{C} - \{\infty\}$. Si $k > 0$, $X(z)$ n'est pas à valeur finie pour $z = 0$, donc la région de convergence est $\mathbb{C} - 0$.
Pour un signal à durée *finie*, la région de convergence correspond au plan complexe $\mathbb{C}$ avec l'exclusion possible de $z = 0$ ou $z = \infty$.

Soit le signal à temps discret suivant :

$$x(n) = a^n u(n) \tag{2.2}$$

on a :

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} x(n) z^{-n} \\ X(z) &= \sum_{n=0}^{\infty} a^n z^{-n} \\ X(z) &= \sum_{n=0}^{\infty} (az^{-1})^n \\ X(z) &= \frac{1}{1-az^{-1}} \end{aligned}$$

La série précédente converge si et seulement si $|az^{-1}| < 1$, c'est-à-dire ssi $|a| < |z|$. La figure 2.1 représente la région de convergence dans le plan complexe. Si $a = 1$, on obtient le cas particulier de l'échelon unité. La transformée en z de l'échelon unité est donc :

$$U(z) = \frac{1}{1-z^{-1}}, \quad |z| > 1$$

2.1.2 Description générale d'une région de convergence

Les exemples précédents ont montré qu'une région de convergence peut être l'intérieur ou l'extérieur d'un cercle.
D'une manière générale, la région de convergence est toujours un anneau, c'est-à-dire est définie par l'ensemble des points z tels que $r_1 < z < r_2$, où r_1 peut être nul et r_2 peut être ∞.

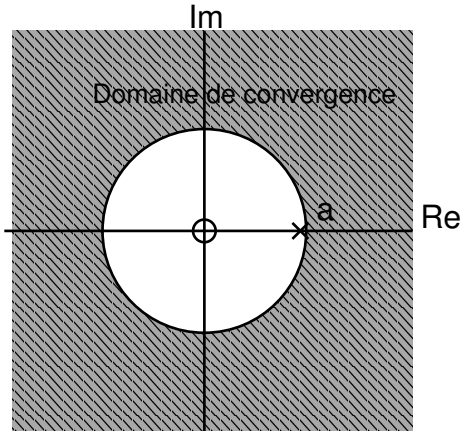


FIG. 2.1: Domaine de convergence, $X(z) = \frac{1}{1-az^{-1}}$

Soit un signal $x(n)$ et z mis sous forme polaire ($z = re^{jw}$), on a alors :

$$\begin{aligned} |X(z)| &= |\sum_{n=-\infty}^{\infty} x(n)z^{-n}| \\ &\leq \sum_{n=-\infty}^{\infty} |x(n)|r^{-n} \\ &\leq \sum_{n=-\infty}^{-1} |x(n)|r^{-n} + \sum_{n=0}^{\infty} |x(n)|r^{-n} \\ &\leq \sum_{n=1}^{\infty} |x(-n)|r^{n} + \sum_{n=0}^{\infty} \frac{|x(n)|}{r^n} \end{aligned}$$

La région de convergence de $X(z)$ correspond au sous ensemble de $\mathbb{C}$ pour lequel les *deux suites* convergent. Supposons que la seconde somme soit finie pour $r = r_2$, quel que soit $r \geq r_2$ la série converge (dans ce cas en effet, chaque élément de la somme est plus petit que pour $r = r_2$). D'une façon analogue, supposons la première somme finie pour $r = r_1$, alors elle est finie pour $r \leq r_1$.
La région de convergence de la première somme correspond au sous ensemble de $\mathbb{C}$ tel que $|z| > r_2$. La région de convergence de la seconde somme correspond au sous ensemble de $\mathbb{C}$ tel que $|z| < r_1$.

2.2 Propriétés de la transformée en z

2.2.1 Linéarité

Soient deux signaux à temps discret $x_1(n)$ et $x_2(n)$ ayant pour transformées en z respectives $X_1(z)$ et $X_2(z)$. Soit le signal $x(n) = ax_1(n) + bx_2(n)$.

La définition de la transformée en z (une somme de monômes en z) conduit directement à la relation suivante :

$$X(z) = aX_1(z) + bX_2(z) \tag{2.3}$$

Il s'agit d'une propriété très importante permettant de calculer une transformée en Z à partir d'une décomposition en transformées élémentaires de signaux connus. La région de convergence de la somme de l'équation (2.3) contient l'intersection des régions de convergence de $X_1(z)$ et $X_2(z)$.
Par exemple, soit $x(n) = \cos(\omega_0 n)u(n)$, on peut décomposer $x(n)$ de la manière suivante :

$$x(n) = \frac{1}{2}(e^{i\omega_0 n} + e^{-i\omega_0 n})u(n)$$

En reprenant l'exemple de l'équation (2.2) avec $a = e^{\pm i\omega_0 n}$, on obtient :

$$\begin{aligned} X(z) &= \frac{1}{2}(\frac{1}{1 - e^{i\omega_0} z^{-1}}) + \frac{1}{2}(\frac{1}{1 - e^{-i\omega_0} z^{-1}}) \\ &= \frac{1 - \cos(\omega_0)z^{-1}1}{1 - 2\cos(\omega_0)z^{-1} + z^{-2}} \end{aligned}$$

2.2.2 Décalage temporel

Soit un signal $x(n)$ de transformée en Z, $X(z)$. Soit k, un indice temporel quelconque et $x'(n) = x(n - k)$, on obtient simplement à partir de la définition de la transformée :

$$X'(z) = z^{-k}X(z)$$

La région de convergence reste inchangée, excepté l'ajout ou la suppression de $z = 0$ ou $z = \infty$. C'est de cette propriété que vient l'utilisation d'une cellule z^{-1} pour tenir compte d'un décalage temporel d'une unité.

2.2.3 Facteur d'échelle en z

Soit un signal $x(n)$ de transformée $X(z)$ avec $r_1 < |z| < r_2$ pour région de convergence. Soit $x'(n) = a^n x(n)$, on a alors :

$$X'(z) = X(\frac{z}{a})$$

avec $|a|r_1 < |z| < |a|r_2$ pour région de convergence.

2.2.4 Inversion de l'axe temporel

Soit $x(n)$ avec $X(z)$ pour transformée et $r_1 < |z| < r_2$ pour rayon de convergence. Soit $x'(n) = x(-n)$, on a alors :

$$X'(z) = X(\frac{1}{z})$$

avec $\frac{1}{r_2} < |z| < \frac{1}{r_1}$ comme rayon de convergence.

2.2.5 Dérivation dans l'espace en z

Soit $x(n)$ avec $X(z)$ pour transformée. Soit $x'(n) = nx(n)$. on a alors :

$$X'(z) = -z\frac{d}{dz}X(z)$$

La région de convergence reste inchangée. Démonstration :

$$\begin{aligned} X'(z) &= -z\frac{d}{dz}X(z) \\ &= \frac{d}{dz}\sum_{n=-\infty}^{\infty} x(n)z^{-n} \\ &= -z\sum_{n=-\infty}^{\infty} x(n)(-n)z^{-n-1} \\ &= \sum_{n=-\infty}^{\infty} [nx(n)]z^{-n} \end{aligned}$$

Par exemple, soit $x(n) = nu(n)$, on a vu précédemment que :

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

On obtient alors :

$$\begin{aligned} X(z) &= -z\frac{d}{dz}U(z) \\ &= -z(\frac{z^{-2}}{(1-z^{-1})^2} \\ &= \frac{z^{-1}}{(1-z^{-1})^2}, |z| > 1 \end{aligned}$$

2.2.6 Convolution

Soient $x_1(n)$ et $x_2(n)$ avec pour transformées en Z respectives $X_1(z)$ et $X_2(z)$. Soit $x(n) = x_1(n) * x_2(n)$, on a alors :

$$X(z) = X_1(z)X_2(z)$$

Démonstration :

$$\begin{aligned} X(z) &= \sum_{n=-\infty}^{\infty} x(n)z^{-n} \\ &= \sum_{n=-\infty}^{\infty} [\sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k)]z^{-n} \\ &= \sum_{k=-\infty}^{\infty} x_1(k)[\sum_{n=-\infty}^{\infty} x_2(n-k)z^{-n}] \\ &= (\sum_{k=-\infty}^{\infty} x_1(k)z^{-k})X_2(z) \\ &= X_1(z)X_2(z) \end{aligned}$$

Pour calculer la convolution de deux signaux, il peut être intéressant de multiplier les transformées respectives des deux signaux convolués et de rechercher la transformée en Z inverse de la transformée résultante.
Soit $s(n)$ la sortie d'un système linéaire invariant de réponse impulsionnelle $h(n)$ soumis à l'entrée $e(n)$, on a vu au cours du chapitre sur les signaux et systèmes que :

$$s(n) = e(n) * h(n)$$

on a alors :

$$S(z) = E(z)H(z)$$

2.3 Transformées en Z rationnelles

On entend par transformées en Z rationnelles, l'ensemble des fonctions en Z s'écrivant comme le ratio de deux polynômes en z^{-1}. Cette classe de fonction correspond aux systèmes linéaires invariants décrits par une équation aux différences finies.

2.3.1 Définition des pôles et des zéros

Les *zéros* d'une transformée en Z, $X(z)$, sont les valeurs de z telles que $X(z) = 0$.
Les *pôles* d'une transformée en Z, $X(z)$, sont les valeurs de z telles que $X(z) = \infty$
Si $X(z)$ est une fonction rationnelle, $X(z)$ peut alors s'écrire sous la forme suivante :

$$\begin{aligned} X(z) &= \frac{N(z)}{D(z)} \\ &= \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}} \\ &= \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \end{aligned} \tag{2.4}$$

En supposant $a_0 \neq 0$ et $b_0 \neq 0$, on peut réécrire l'équation (2.4) de la manière suivante :

$$\begin{aligned} X(z) &= \frac{b_0}{a_0} \frac{z^{-M}}{z^{-N}} \frac{\frac{b_M}{b_0} + \cdots + z^M}{\frac{a_N}{a_0} + \cdots + z^N} \\ &\triangleq \frac{b_0}{a_0} z^{N-M} \frac{N'(z)}{D'(z)} \end{aligned}$$

$N'(z)$ a au plus M racines simples ou multiples en $z_1 \cdots z_M$.
$D'(z)$ a au plus N pôles simples ou multiples en $p_1 \cdots p_N$.
On alors réécrire $X(z)$ sous la forme suivante :

$$\begin{aligned} X(z) &= \frac{b_0}{a_0} \frac{z^{-M}}{z^{-N}} \frac{(z - z_1) \cdots (z - z_M)}{(z - p_1) \cdots (z - p_N)} \\ &= \alpha \frac{z^{-M}}{z^{-N}} \frac{\prod_{k=1}^{M} (z - z_k)}{\prod_{k=1}^{N} (z - p_k)} \end{aligned}$$

avec $\alpha \triangleq \frac{b_0}{a_0}$. Il vient alors que :

1. $X(z)$ possède M zéros finis en $z_1 \cdots z_M$
2. $X(z)$ possède N pôles finis en $p_1 \cdots p_N$
3. si $N > M$, $X(z)$ possède $N - M$ zéros en $z = 0$
4. si $N < M$, $X(z)$ possède $M - N$ pôles en $z = 0$
5. il peut aussi y avoir des pôles ou zéros en $z = \infty$ selon que $X(\infty) = 0$ ou $X(\infty) = \infty$

En suivant la notation précédente, $X(z)$ est complètement déterminé par la position de ses pôles et de ses zéros ainsi que par le facteur d'amplitude α. Les pôles et zéros reflètent le *comportement* du système (ou signal) tandis que le facteur α n'intervient que sur l'amplitude des signaux.
$X(z)$ peut donc être représente sous la forme d'un graphique modélisant la position des pôles et des zéros dans le plan complexe. Par définition, la région de convergence de $X(z)$ exclue tous les pôles de cette fonction.

2.3.2 Fonction de transfert d'un système linéaire invariant

Au cours du premier chapitre, on a vu qu'une manière de caractériser un système linéaire invariant consiste à étudier sa réponse impulsionnelle $h(n)$. Il est donc tout aussi légitime de caractériser un système par la transformée en Z, $H(z)$, de sa réponse impulsionnelle, encore appelée *fonction de transfert* du système.
Lors de l'analyse d'un système donné, on considère le plus souvent $h(n)$ ou $H(z)$ comme inconnue. A partir d'une entrée connue, $e(n)$, on observe alors la sortie $s(n)$ caractérisée par sa transformée en Z, $S(z)$. la fonction de transfert du système est alors :

$$H(z) = \frac{S(z)}{E(z)}$$

On a vu que si on prend $e(n) = \delta(n)$, on obtient directement $s(n) = h(n)$; ce qui devient dans le plan en Z, $E(z) = 1$ donc $H(z) = S(z)$
Si on applique cette approche au systèmes linéaires invariants décrits par une équation aux différences finies, le système est décrit par la relation suivante :

$$s(n) = -\sum_{k=1}^{N} a_k s(n-k) + \sum_{k=0}^{M} b_k e(n-k)$$

Prenons la transformée en Z des membres de l'équation précédente :

$$S(z) = -\sum_{k=1}^{N} a_k z^{-k} S(z) + \sum_{k=0}^{M} b_k z^{-k} E(z)$$

ou encore :

$$S(z)[1 + \sum_{k=1}^{N} a_k z^{-k}] = E(z)[\sum_{k=0}^{M} b_k z^{-k}]$$

en posant $a_0 = 1$, sans perdre en généralité, on a :

$$\begin{aligned} H(z) &= \frac{S(z)}{E(z)} \\ &= \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \\ &= \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \end{aligned} \tag{2.5}$$

L'équation (2.5) met en évidence qu'un système linéaire invariant décrit par une équation aux différences finies a une fonction de transfert dont la transformée en Z est une fonction rationnelle. Ceci montre l'intérêt de l'étude des transformées en Z s'écrivant sous la forme de polynômes rationnels.

2.3.2.1 Systèmes *tout* zéros

Un système *tout zéros* est un système dont la transformée en Z s'exprime sous la forme de l'équation (2.5) avec $N = 0$, c'est-à-dire pour lequel $a_1 = a_2 = \cdots = a_N = 0$.
La fonction de transfert devient :

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

Le polynôme numérateur possède M racines. La réponse impulsionnelle de ce système est de type finie (RIF ou FIR).

2.3.2.2 Systèmes *tout* pôles

Un système *tout pôles* est un système dont la transformée en z s'exprime sous la forme de l'équation (2.5) avec $M = 0$, c'est-à-dire pour lequel $b_1 = b_2 = \cdots = b_M = 0$.
La fonction de transfert devient :

$$H(z) = \frac{b_0}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \frac{b_0 z^N}{\sum_{k=0}^{M} a_k z^{N-k}}$$

Le polynôme dénominateur possède N racines. La réponse impulsionnelle de ce système est de type infinie (RII ou IIR).

2.4 Transformée en Z inverse

A partir d'une liste de transformées en Z de signaux élémentaires connus, il peut être efficace de retrouver des signaux temporels à partir de transformées dérivées des opérateurs et propriétés décrits précédemment. Cependant, lorsque la transformée ne peut facilement s'écrire comme la combinaison de transformées élémentaires, il reste les techniques générales de transformation inverse :

1. L'intégration sur un contour fermé.
2. le développement en puissance de z et de z^{-1}
3. le développement en fractions élémentaires

2.4.1 Transformée inverse par intégration

Soit $X(z)$ la transformée en Z du signal $x(n)$. On définit la transformée en Z inverse, la relation déterminant $x(n)$ à partir de $X(z)$ telle que :

$$x(n) = \frac{1}{2\pi j} \oint X(z) z^{n-1} dz$$

L'intégrale précédente consiste à sommer $X(z)z^{n-1}$ pour des valeurs de z prises sur un contour fermé du plan complexe qui contient l'origine du plan tout en étant inclue dans le domaine de convergence de la fonction.

2.4.2 Transformée inverse par développement en puissance

S'il est possible d'écrire $X(z)$ comme une série de puissances de z^{-1}, l'unicité de la transformation directe conduit à prendre les coefficients de la série pour le signal temporel.
Si $X(z) = \sum_{n=-\infty}^{\infty} c_n z^{-n}$ alors $x(n) = c_n$
On recherche par exemple la réponse impulsionnelle d'un système décrit par l'équation aux différences suivante :

$$s(n) = s(n-3) + e(n)$$

On trouve aisément :

$$H(z) = \frac{1}{1 - z^{-3}}$$

En utilisant la limite des séries géométriques, on a :

$$\frac{1}{1 - z^{-3}} = \sum_{k=0}^{\infty} (z^{-3})^k$$

$$H(z) = \sum_{k=0}^{\infty} z^{-3k} = 1 + z^{-3} + z^{-6} + \cdots$$

on obtient donc :

$$h(n) = \sum_{k=0}^{\infty} \delta(n - 3k)$$

Il s'agit ici d'un cas simple d'utilisation des séries géométriques. En général, le développement de $X(z)$ en puissance de z^{-1} est un calcul assez long, difficile et fastidieux.

2.4.3 Transformée inverse par développement fractionnaire

L'idée générale de cette approche consiste à trouver pour une fonction $X(z)$ complexe un développement en fonctions en Z plus simples et pour lesquelles une transformée inverse est connue. En appliquant le principe de linéarité de la transformée, il est aisé de recomposer le signal temporel inverse à partir des signaux temporels correspondant à chacune des transformées élémentaires. En supposant :

$$X(z) = \alpha_1 X_1(z) + \alpha_2 X_2(z) + \cdots + \alpha_L X_L(z)$$

On obtient :

$$x(n) = \alpha_1 x_1(n) + \alpha_2 x_2(n) + \cdots + \alpha_L x_L(n)$$

La classe des transformées en Z rationnelles peut toujours s'écrire selon ce principe. Les formes *élémentaires* sont le plus souvent des formes telles que définies selon le tableau suivant :

Type	$X(z)$	$x(n)$
polynomial en z	$\sum_k c_k^{-k}$	$\sum_k c_k \delta(n-k)$
pôle réel simple	$\frac{1}{1-pz^{-1}}$	$p^n u(n)$
pôle réel double	$\frac{pz^{-1}}{(1-pz^{-1})^2}$	$np^n u(n)$
pôle réel double	$\frac{1}{(1-pz^{-1})^2}$	$(n+1)p^n u(n)$
pôle réel triple	$\frac{p^2 z^{-1}}{(1-pz^{-1})^3}$	$\frac{n(n-1)}{2} p^n u(n)$
pôle complexe conjugué	$\frac{r\sin(\omega_0)z}{(z-re^{i\omega_0})(z-re^{-i\omega_0})}$	$r^n \sin(\omega_0 n)u(n)$

2.5 Analyse des Systèmes LI par la transformée en Z

L'objet de ce paragraphe est d'étudier le comportement des systèmes linéaires invariants par l'analyse de leur fonction de transfert décrite par une fonction en Z. Il s'agit donc de caractériser la sortie d'un système en fonction d'une entrée et de spécifier les conditions de stabilité d'un tel système.

2.5.1 Réponse d'un système décrit par une fonction rationnelle

Soit le système décrit par l'équation suivante :

$$S(z) = H(z)E(z)$$

Il s'agit de caractériser $s(n)$. On suppose que $H(z)$ peut s'écrire sous la forme d'une fraction de polynômes en z, c'est-à-dire :

$$H(z) = \frac{N(z)}{D(z)}$$

On suppose de plus que la transformée de l'entrée du système $E(z)$ s'écrit par une fraction de polynômes :

$$E(z) = \frac{P(z)}{Q(z)}$$

On a donc :

$$S(z) = \frac{N(z)P(z)}{D(z)Q(z)}$$

Le signal de sortie est donc caractérisé par une transformée rationnelle.
Supposons :
– Les pôles du système sont uniques, $p_1 \cdots p_N$
– Les pôles de l'entrée sont uniques, $q_1 \cdots q_L$
– Les pôles du système et de l'entrée sont différents.
– Les zéros du système et de l'entrée diffèrent de l'ensemble des pôles.
On peut alors écrire $Y(z)$ sous la forme suivante :

$$S(z) = \sum_{k=1}^{N} \frac{D_k}{1-p_k z^{-1}} + \sum_{k=1}^{L} \frac{Q_k}{1-q_k z^{-1}}$$

En supposant un système causal, on obtient :

$$s(n) = \sum_{k=1}^{N} D_k p_k^n u(n) + \sum_{k=1}^{L} Q_k q_k^n u(n)$$

L'équation précédente montre que le signal de sortie peut être considéré comme composé de deux partie :
– une réponse en régime naturel (termes en p_k)
– une réponse en régime forcé (termes en q_k)

2.5.2 Régimes transitoires et permanents

On considère $s_n(n)$ la réponse du système en régime naturel :

$$s_n(n) = \sum_{k=1}^{N} D_k p_k^n u(n)$$

et $s_f(n)$ en regime forcé :

$$s_f(n) = \sum_{k=1}^{L} Q_k q_k^n u(n)$$

Si tous les pôles de la fonctions de transfert du systèmes ont des amplitudes inférieures à 1, alors le régime naturel est dit *transitoire*. Des pôles de faible module conduisent à une décroissance rapide de la réponse naturelle. Réciproquement des pôles proches du cercle unité conduisent à une réponse transitoire longue.
Si tous les pôles du signal d'entré ont un module inférieur à 1, alors la réponse forcé décroît vers 0 pour n allant vers l'infini. Si le signal d'entrée à un pôle sur le cercle unité, alors le signal est composé d'une sinusoïde persistante. Dans un tel cas, la réponse forcée correspond à ce que l'on appelle un état stable.

2.6 Causalité et Stabilité

2.6.0.1 Condition de causalité pour $H(z)$

Au cours du chapitre caractérisant les systèmes à temps discret, on a vu qu'un système linéaire invariant est causal ssi sa réponse impulsionnelle $h(n)$ est nulle si $n < 0$.
Un système linéaire invariant est causal si et seulement si le domaine de convergence de sa transformée en z est *l'extérieur* d'un cercle de rayon $r < \infty$ incluant $z = \infty$. Ainsi, $H(z) = z^2$ a une région de convergence : $\mathbb{C} - \infty$ qui est extérieur à un cercle de rayon 0 mais excluant $z = \infty$. Donc le système est non causal.

2.6.0.2 Condition de stabilité pour $H(z)$

On a vu que pour qu'un système linéaire invariant soit stable il faut et il suffit que sa réponse impulsionnelle vérifie :

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

Un système linéaire invariant est stable si le domaine de convergence de sa transformée en z inclue le cercle unité.
Pour finir, un système linéaire invariant causal est stable si et seulement si tous les pôles de sa fonction de transfert sont à l'intérieur, strictement, du cercle unité.

2.7 Détermination du module et de la phase du système

Pour un système dont le domaine de convergence de sa fonction de transfert $X(z)$ contient le cercle unité, alors la réponse fréquentielle de ce système consiste à évaluer $X(z)$ sur le cercle unité, c'est-à-dire pour $z = e^{j\Omega}$.
Prenons par exemple, un système dont la fonction de transfert s'exprime sous la forme suivante :

$$H(z) = \frac{1}{1 + 0.5z^{-1}}$$

on a :

$$H(z) = \frac{z}{z + 0.5}$$

Ce système possède donc un zéro en $z = 0$ et un pôle en $z = -\frac{1}{2}$.
En prenant z sur le cercle unité :

$$H(e^{j\Omega}) = \frac{e^{j\Omega}}{e^{j\Omega} + \frac{1}{2}}$$

En précisant explicitement amplitude et phase :

$$H(e^{j\Omega}) = |H(e^{j\Omega})|e^{jarg(H(e^{j\Omega}))}$$

On obtient alors :

$$|H(e^{j\Omega})| = \frac{1}{e^{j\Omega} + \frac{1}{2}} \tag{2.6}$$

$$arg(H(e^{j\Omega})) = \Omega - arg(e^{j\Omega} + \frac{1}{2}) \tag{2.7}$$

la double figure 2.2 représente sur sa partie gauche la position des pôles et des zéros de la fonction de transfert (un pôle est indiqué par le symbole x, et un zéro par le symbole o) et sur sa partie droite la réponse fréquentielle du système : module et phase pour $0 \leq \Omega \leq \pi$.
L'évolution du module de la réponse fréquentielle nous montre qu'il-y-a atténuation des contributions dues au basses fréquences par rapport aux hautes fréquences. Le comportement de ce système est donc celui d'un filtre passe-haut.

2.8 Transformées en z de fonctions usuelles

Le tableau 2.1 donne les transformées en z des fonctions les plus utilisées en TNS. T est la période d'échantillonnage du signal transformé dans lequel on a posé $t = nT$.

$x(t)$	$X(z)$
$\delta(t)$	1
$\delta(t-kT)$	z^{-k}
$u(t)$	$\frac{z}{z-1}$
t	$\frac{Tz}{(z-1)^2}$
$\frac{1}{2}t^2$	$\frac{T^2z(z+1)}{2(z-1)^3}$
$a^{\frac{t}{T}}$	$\frac{z}{z-a}$
e^{-at}	$\frac{z}{z-e^{-aT}}$
$\sin\omega_0 t$	$\frac{z\sin\omega_0 T}{z^2-2z\cos\omega_0 T+1}$
$\cos\omega_0 t$	$\frac{z(z-\cos\omega_0 T)}{z^2-2z\cos\omega_0 T+1}$
$e^{-at}\sin\omega_0 t$	$\frac{ze^{-aT}\sin\omega_0 T}{z^2-2ze^{-aT}\cos\omega_0 T+e^{-2aT}}$
$e^{-at}\cos\omega_0 t$	$\frac{z^2-ze^{-aT}\cos\omega_0 T}{z^2-2ze^{-aT}\cos\omega_0 T+e^{-2aT}}$
$ax_1(t)+bx_2(t)$	$aX_1(z)+bX_2(z)$
$x(t-kT)$	$z^{-k}X(z)$

TAB. 2.1: Tables des transformées en z

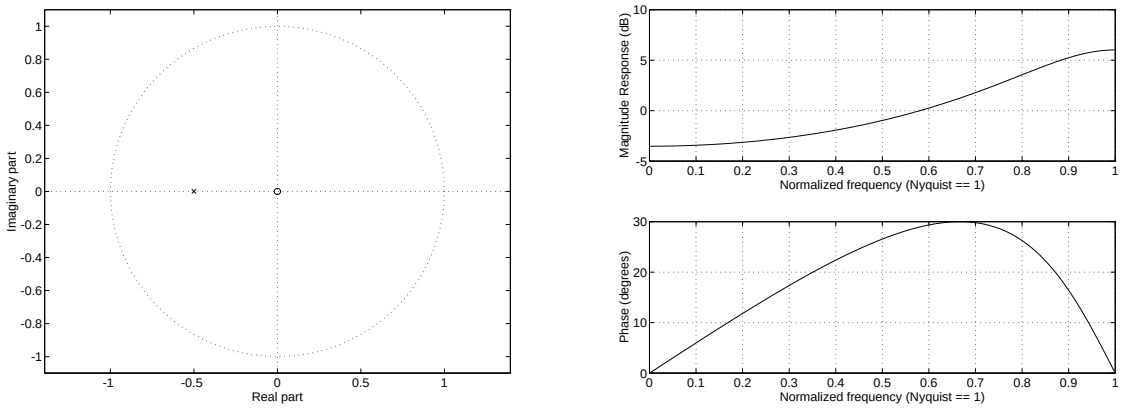


FIG. 2.2: Réponse fréquentielle, $X(z) = \frac{1}{1+0.5z^{-1}}$

Chapitre 3

Échantillonnage et reconstruction des signaux

Ce chapitre présente l'opération d'échantillonnage des signaux à temps continu permettant de représenter des signaux physiques analogiques sur une forme à temps discret. On abordera cette notion en étudiant tout d'abord le cas d'un échantillonnage idéal où les échantillons du signal à temps discret sont dérivés du signal analogique à des instants périodiques multiples d'une période élémentaire. Comme la notion d'instant a peu de sens en pratique et qu'il est plus juste de parler de durée, on abordera ensuite le cas d'un système d'échantillonnage réel, tel qu'on peut on trouver dans des convertisseurs analogique/numérique ; on précisera quelles approximations permettent de se ramener au cas de l'échantillonneur idéal.
L'opération d'échantillonnage inverse consiste à reconstruire un signal analogique à partir d'un signal à temps discret. On étudiera un modèle de reconstruction du signal tel que l'opération complète échantillonnage et reconstruction soit transparente.

3.1 Échantillonnage idéal

Soit un signal analogique $x_a(t)$ dont la transformée de Fourier est définie par :

$$X_a(j\omega) = \int_{-\infty}^{\infty} x_a(t)e^{-j\omega t}dt$$

avec $\omega = 2\pi f$.
On retrouve le signal temporel à partir de sa transformée par la transformée de Fourier inverse définie par le relation suivante :

$$x_a(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_a(j\omega)e^{j\omega t}d\omega$$

Un signal à temps discret $x(n)$ dérivé d'un signal analogique $x_a(t)$ consiste à prendre des valeurs de $x_a(t)$ à des instants multiples d'une période fixe T. T est appelée *période d'échantillonnage*. On a donc :

$$x(n) = x_a(t = nT)$$

L'inverse de la période d'échantillonnage est appelée *fréquence d'échantillonnage*, f_e ; on a $f_e = \frac{1}{T}$. On notera ω, la pulsation non normalisée s'exprimant en radian par seconde. On notera Ω, la pulsation normalisée, s'exprimant en radian par période. On a $\omega = \frac{\Omega}{T}$.

Pour se rendre compte de la relation qu'il y a entre $x(n)$ et $x_a(n)$, il est utile de relier les transformées de Fourier de ces deux signaux.
On a :

$$x(n) = x_a(nT) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X_a(j\omega)e^{j\omega nT}d\omega$$

A partir de la définition de la transformée de Fourier à temps discret, on a aussi :

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(j\Omega)e^{j\Omega n}d\Omega$$

Pour pouvoir relier les deux équations précédentes (par $x(n)\cdots$), on exprime la première en découpant l'intervalle d'intégration par morceaux, on a donc :

$$x_a(nT) = \frac{1}{2\pi}\sum_{k=-\infty}^{\infty}\int_{(2k-1)\frac{\pi}{T}}^{(2k+1)\frac{\pi}{T}} X_a(j\omega)e^{j\omega nT}d\omega$$

Par un changement de variable, on obtient :

$$x_a(nT) = \frac{1}{2\pi}\sum_{k=-\infty}^{\infty}\int_{\frac{-\pi}{T}}^{\frac{\pi}{T}} X_a(j\omega + j\frac{2\pi k}{T})e^{j\omega nT}e^{2\pi kn}d\omega$$

On peut noter que $e^{2\pi kn} = 1$ pour toutes valeurs entières de k et de n. En échangeant le signe somme et le signe intégral, on obtient :

$$x(n) = x_a(nT) = \frac{1}{2\pi}\int_{\frac{-\pi}{T}}^{\frac{\pi}{T}}[\sum_{k=-\infty}^{\infty} X_a(j\omega + j\frac{2\pi k}{T})]e^{j\omega nT}d\omega$$

En remplaçant ω par $\frac{\Omega}{T}$, on a :

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi}[\frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(\frac{j\Omega}{T} + \frac{j2\pi k}{T})]e^{j\Omega n}d\Omega$$

Il est donc maintenant aisée d'effectuer l'identification suivante :

$$X(e^{j\Omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(\frac{j\Omega}{T} + \frac{j2\pi k}{T})$$

La relation précédente montre que le spectre du signal numérique est *périodique* et correspond à des versions décalées du spectre du signal analogique correspondant.

3.1.0.3 Théorème d'échantillonnage de Shannon

Le spectre du signal numérique est composé d'une somme infinie de versions décalées du signal analogique. Il est donc possible de retrouver le signal à temps continu si tous les éléments sommés ne se recouvrent pas.
Si le signal analogique est à bande limitée, c'est-à-dire si sa transformée de Fourier à temps continu $X(\omega)$ est nulle pour $|\omega| > \omega_{\max}$, il n'y a pas recouvrement des spectres décalés pour $\omega_{\max} < \frac{2\pi}{T} - \omega_{\max}$.

On obtient alors :

$$\omega_{\max} < \frac{2\pi}{T} - \omega_{\max} \tag{3.1}$$

$$\omega_{\max} < \frac{\pi}{T} \tag{3.2}$$

$$2\frac{\omega_{\max}}{2\pi} < \frac{1}{T} \tag{3.3}$$

$$2f_{\max} < f_N \quad \text{avec} \quad f_N = \frac{1}{T} \tag{3.4}$$

f_N est la fréquence limite d'échantillonnage ou encore appelée fréquence de Nyquist. Il est donc tout d'abord s'assurer que l'on échantillonne bien un signal continu à bande limitée. Ceci est le plus souvent validé par le filtrage du signal analogique par un filtre passe-bas. Une fois la bande de base fixée, on détermine a fréquence d'échantillonnage à appliquer pour obtenir une reconstruction parfaite du signal continu à partir du signal numérique.

3.2 Exemple pratique d'échantillonnage

Soit un signal analogique $x_a(t)$ correspondant à une sinusoïde de fréquence $f = 0.5$Hz et défini par :

$$x_a(t) = 5\sin(2\pi 0.5t) \quad \text{avec} \quad -2 \leq t \leq +2$$

D'après le paragraphe précédent, la fréquence de Nyquist correspond à $f_N = 2 * 0.5 = 1$Hz.
On prend pour fréquence d'échantillonnage $f_s = 4$Hz, soit $x(n)$ le signal à temps discret résultant de l'échantillonnage de $x_a(t)$ à la fréquence f_s.

La figure 3.1 montre en trait continu la signal analogique que l'on vient échantillonné en certains instants repérés par des lignes verticales terminées par des croix.
La série des figures 3.2 à 3.4 met en évidence les problèmes de reconstruction si on ne respecte pas la contrainte du théorème de Shannon. Le signal analogique d'origine est une sinusoïde de 2 Hz. Donc une fréquence d'échantillonnage valide devra être supérieure à 4 Hz.
Les figures 3.2 et 3.3 montrent en trait pointillé le signal analogique d'origine, en trait plein le signal analogique reconstruit à partir du signal numérique noté par la séquence des points-étoiles. On voit que le signal reconstruit n'a plus rien à voir avec le signal d'origine. En effet la fréquence d'échantillonnage est inférieur à celle de Nyquist (4 Hz).
En utilisant les mêmes notations que pour les deux figures précédentes, on peut remarquer qu'il y a reconstruction parfaite pour la figure 3.4, il s'agit en fait du signal numérique échantillonné à une fréquence de 6.4 Hz. On peut aussi noter que si on ne regarde que l'enveloppe d'amplitude du signal numérique (séquence des étoiles), elle a peu de chose à voir avec une sinusoïde, du moins à l'oeil !...donc méfiance.

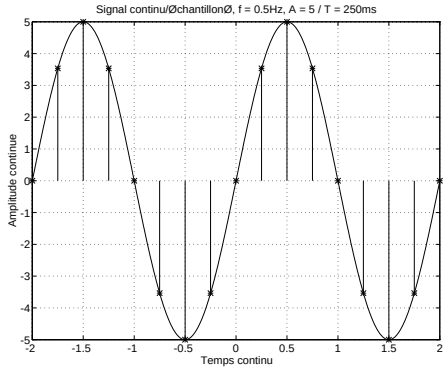


FIG. 3.1: Échantillonnage d'un signal continu ; en trait continu le signal analogique, les traits verticaux notent les valeurs du signal numérique

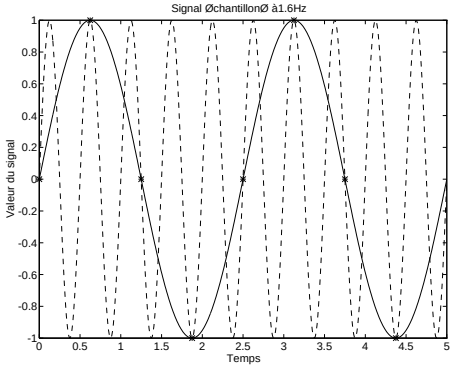


FIG. 3.2: Échantillonnage d'un signal continu et reconstruction imparfaite ; en trait discontinu le signal analogique d'origine ($F = 2Hz$), en trait plein le signal analogique reconstruit, $F_s = 1.6Hz$

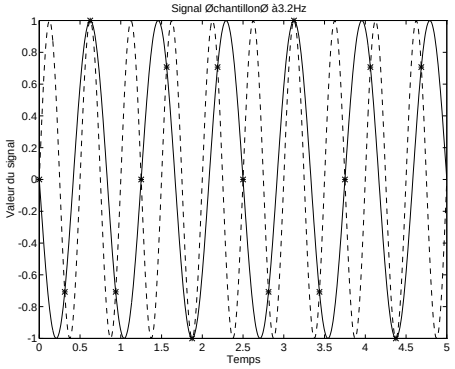


FIG. 3.3: Échantillonnage d'un signal continu et reconstruction imparfaite, $F_s = 3.2Hz$

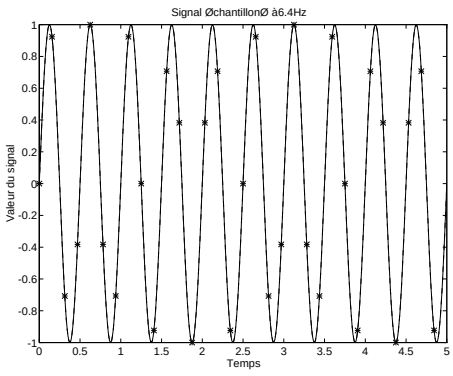


FIG. 3.4: Échantillonnage d'un signal continu et reconstruction parfaite, $F_s = 6.4Hz$

Chapitre 4

La Transformée de Fourier Discrète

Nous avons vu au cours du chapitre sur les Signaux et systèmes que la transformée de Fourier de deux signaux convolués correspond à la multiplication des transformées des deux signaux pris séparément. Comme l'opérateur de convolution est un opérateur fondamental dans l'analyse des systèmes de traitement du signal, il est donc important de disposer de solutions algorithmiques efficaces pour cet opérateur.
Pour une transformée de Fourier à temps discret la variable *temps* est discrète, donc peut être représentée sur un calculateur tandis que la variable *fréquence* est une variable continue qu'il faut aussi *discrétiser*. En fait sur un calculateur, un signal temporel, est une séquence de longueur *finie*, par exemple N points. Il n'est pas nécessaire d'avoir une précision *infinie* dans le domaine fréquentiel ; N points fréquentiels suffisent pour contenir l'information du signal temporel.
La Transformée de Fourier *Discrète*, TFD ou DFT, est donc l'outil définissant le cadre de calcul d'une transformée de Fourier à temps discret et à fréquences discrètes.
Si un signal est défini par N échantillons temporels, la transformée de Fourier Discrète est une transformée qui opère sur N points fréquentiels définis par la série des pulsations suivantes :

$$\Omega_k = \frac{2\pi}{N}, \quad k = 0, 1, \cdots, N-1$$

4.1 Rappels sur les signaux continus

Soit un signal analogique $x_a(t)$ dont la transformée de Fourier est définie par :

$$X_a(j\omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\omega t} dt \tag{4.1}$$

avec $\omega = 2\pi f$.
On retrouve le signal temporel à partir de sa transformée par la transformée de Fourier inverse définie par le relation suivante :

$$x_a(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_a(j\omega) e^{j\omega t} d\omega \tag{4.2}$$

4.2 Rappels sur les signaux discrets non périodiques

Pour un signal $x(n)$ discret quelconque non périodique, sa transformée de Fourier s'écrit :

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n} \tag{4.3}$$

On parlera également de Transformée de Fourier à Temps Discret (TFTD ou DTFT). $X(\Omega)$ étant *périodique* de période 2π à cause de la discrétisation de la variable temporelle, la TF inverse est donnée par :

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\Omega})e^{jn\Omega}d\Omega \tag{4.4}$$

4.3 Signaux discrets périodiques

Pour un signal $x_p(n)$ discret périodique, une décomposition en série de Fourier doit être utilisée sous la forme :

$$X_p(k) = \sum_{n=0}^{N-1} x_p(n).e^{(-2j\pi\frac{n.k}{N})},\ k = 0,\ 1 \ldots N-1 \tag{4.5}$$

$$x_p(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_p(k).e^{(2j\pi\frac{n.k}{N})},\ n = 0,\ 1 \ldots N-1 \tag{4.6}$$

Sa Transformée de Fourier s'écrit alors :

$$X_p(e^{j\Omega}) = \sum_{k=-\infty}^{\infty} X_p(k)\delta\left(\Omega - k\frac{2\pi}{N}\right) \tag{4.7}$$

4.4 Propriétés de la transformées de Fourier

Les principales propriétés de la TF sont énumérées ci dessous.

– Linéarité ou superposition

$$a.x(n) + b.y(n) \rightleftharpoons a.X(e^{j\Omega}) + b.Y(e^{j\Omega})$$

– Décalage en temps-fréquence

$$x(n - n_0) \rightleftharpoons e^{-jn_0\Omega}X(e^{j\Omega})$$
$$x(n)e^{jn\Omega_0} \rightleftharpoons X(e^{j(\Omega-\Omega_0)})$$

– Dérivation en fréquence

$$n.x(n) \rightleftharpoons j\frac{dX(e^{j\Omega})}{d\Omega}$$

– Produit de convolution

$$x_1(n) * x_2(n) = \sum_{i=-\infty}^{\infty} x_1(i) * x_2(n-i) \rightleftharpoons X_1(e^{j\Omega}).X_2(e^{j\Omega})$$

$$\sum_{n=-\infty}^{\infty} x_1(n) * x_2(n) \rightleftharpoons \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(e^{j\Omega}).X_2^*(e^{j\Omega})d\Omega$$

– Théorème du fenêtrage (ou de la modulation)

$$x_1(n).x_2(n) \rightleftharpoons \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(e^{j\Theta}).X_2(e^{j(\Omega-\Theta)})d\Theta$$

– Théorème de Parseval (conservation de la puissance d'un signal)

$$\sum_{i=-\infty}^{\infty} |x(i)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\Omega})|^2 d\Omega$$

– Propriétés de symétrie

$$x(n) \in \mathbb{R} \rightleftharpoons X(e^{j\Omega}) = X^*(e^{-j\Omega})$$

Si $x(n)$ est une suite réelle, alors sa TF est symétrique conjuguée. Cela implique que sa partie réelle et son module sont paires, et que sa partie imaginaire et sa phase sont impaires.

$$x(-n) \rightleftharpoons X(e^{-j\Omega})$$

$$x(-n) \rightleftharpoons X^*(e^{j\Omega}), six(n) \in \mathbb{R}$$

Le tableau 4.1 donne les transformées de Fourier des fonctions les plus utilisées en TNS. T est la période d'échantillonnage.

4.5 Échantillonnage du domaine Fréquentiel

Pour un signal $x(n)$ quelconque, sa Transformée de Fourier à Temps Discret, TFTD ou DTFT, est donnée à l'équation 4.3. On échantillonne maintenant la TFTD en N points fréquentiels, tels que :

$$X_k \triangleq X(\frac{2\pi}{N}k) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\frac{2\pi kn}{N}}$$

Essayons maintenant de trouver la relation qu'il peut y avoir entre le signal $x(n)$ et sa TFD définie par la séquence des nombres X_k. Pour cela, on découpe la somme infinie de l'équation précédente en blocs de N points et on effectue un changement de variable.

$x(n)$	$X(e^{j\Omega})$
$\delta(n)$	1
$\delta(n-n_0)$	$e^{-j\Omega n_0}$
1	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\Omega+2\pi k)$
$u(n)$	$\frac{1}{1-e^{-j\Omega}}+\sum_{k=-\infty}^{\infty}\pi\delta(\Omega+2\pi k)$
$a^n u(n) \quad (\lvert a\rvert<1)$	$\frac{1}{1-ae^{-j\Omega}}$
$(n+1)a^n u(n) \quad (\lvert a\rvert<1)$	$\frac{1}{(1-ae^{-j\Omega})^2}$
$x(n)=\begin{cases}1 & \text{si } 0\le n\le N\\ 0 & \text{ailleurs}\end{cases}$	$\frac{\sin(\Omega(N+1)/2)}{\sin(\Omega/2)}e^{-j\Omega N/2}$
$\frac{\sin\Omega_c n}{\pi n}$	$X(e^{j\Omega})=\begin{cases}1 & \text{si } \lvert\Omega\rvert<\Omega_c\\ 0 & \text{si } \Omega_c<\lvert\Omega\rvert<\pi\end{cases}$
$e^{j\Omega_0 n}$	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\Omega-\Omega_0+2\pi k)$
$\cos(\Omega_0 n+\phi)$	$\sum_{k=-\infty}^{\infty}[\pi e^{-j\phi}\delta(\Omega+\Omega_0+2\pi k)+\pi e^{j\phi}\delta(\Omega-\Omega_0+2\pi k)]$
$\sin(\Omega_0 n+\phi)$	$j\sum_{k=-\infty}^{\infty}[\pi e^{-j\phi}\delta(\Omega+\Omega_0+2\pi k)-\pi e^{j\phi}\delta(\Omega-\Omega_0+2\pi k)]$

TAB. 4.1: Tables des transformées de Fourier

$$\begin{aligned}
X_k &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\frac{2\pi kn}{N}} \\
&= \sum_{l=-\infty}^{\infty} \sum_{n=lN}^{lN+N-1} x(n)e^{-j\frac{2\pi kn}{N}} \\
&= \sum_{l=-\infty}^{\infty} \sum_{n=0}^{N-1} x(n+lN)e^{-j\frac{2\pi k(n+lN)}{N}} \\
&= \sum_{l=-\infty}^{\infty} \sum_{n=0}^{N-1} x(n+lN)e^{-j\frac{2\pi kn}{N}} \\
&= \sum_{n=0}^{N-1} \sum_{l=-\infty}^{\infty} x(n+lN)e^{-j\frac{2\pi kn}{N}} \\
&= \sum_{n=0}^{N-1} \sum_{l=-\infty}^{\infty} x(n-lN)e^{-j\frac{2\pi kn}{N}}
\end{aligned}$$

On trouve donc :

$$X_k = \sum_{n=0}^{N-1} [\sum_{l=-\infty}^{\infty} x(n-lN)]e^{-j\frac{2\pi kn}{N}}$$

Soit encore :

$$X_k = \sum_{n=0}^{N-1} x_p(n)e^{-j\frac{2\pi kn}{N}} \quad \text{où} \quad x_p(n) \triangleq \sum_{l=-\infty}^{\infty} x(n-lN)$$

On constate donc par l'équation précédente que le signal $x_p(n)$ est un signal *périodique*, et plus précisément :

$$x_p(n+N) = x_p(n)$$

Puisque le signal $x_p(n)$ est périodique, il peut s'exprimer comme une série de Fourier. Il est donc possible de retrouver les échantillons $x_p(n)$ à partir de la séquence X_k par une série de Fourier à temps discret :

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi kn}{N}} \quad \text{avec} \quad c_k = \frac{1}{N}\sum_{n=0}^{N-1} x_p(n)e^{-j\frac{2\pi kn}{N}}$$

En comparant les coefficients de la série de Fourier et ceux de la TFTD, on obtient :

$$c_k = \frac{1}{N}X_k$$

On obtient donc $x_p(n)$ à partir des X_k selon la relation suivante :

$$x_p(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j\frac{2\pi kn}{N}}$$

Il faut bien voir qu'il s'agit d'une relation entre $x_p(n)$ et la séquence des X_k, il faut déterminer la relation entre $x_p(n)$ et $x(n)$.
Cette relation s'interprète de la même manière que les versions répliquées du spectre d'un signal analogique lorsqu'on effectue un échantillonnage temporel de ce signal. Ici, le signal $x_p(n)$ est une somme de répliques du signal "de base" $x(n)$. On observe alors le même problème de reconstruction que lors de la discrétisation de la variable temporelle, c'est-à-dire, si les échantillons se recouvrent entre deux séquences consécutives, on est en présence de recouvrement dit *temporel* ou time aliasing.
Il existe un cas particulier où les sommes décalées dans la construction de $x_p(n)$ ne se recouvrent pas ; il s'agit des signaux à *temps limité*. On définit un signal à temps limité un signal $x(n)$ de durée L, prenant des valeurs non nulles *seulement* dans l'intervalle $[0 \cdots L-1]$. Ainsi, si $x(n)$ est un signal à temps limité de durée L, tel que $N \geq L$, il est possible de retrouver $x(n)$ à partir de $x_p(n)$ en isolant une séquence particulière :

$$x(n) = \begin{cases} x_p(n) & \text{si } 0 \leq n \leq L-1 \\ 0 & \text{sinon} \end{cases}$$

Puisqu'on a reconstruit le signal d'origine $x(n)$ à partir de sa TFD, séquence des valeurs X_k, il n'est pas interdit maintenant de calculer par une TFTD le spectre de $x(n)$ pour des fréquences *continues*, $X(\omega)$. Il existe une formule de passage directe entre les X_k et $X(\omega), -\pi \leq \omega \leq \pi$: il s'agit de la formule d'interpolation de Dirichlet.
Pour le reste de ce chapitre on considérera des signaux temporels à temps limité.

4.6 Transformée de Fourier Discrète

La transformée de Fourier Discrète (TFD) d'un signal $x(n)$ à temps limité de longueur N est définie par la relation suivante :

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}}$$

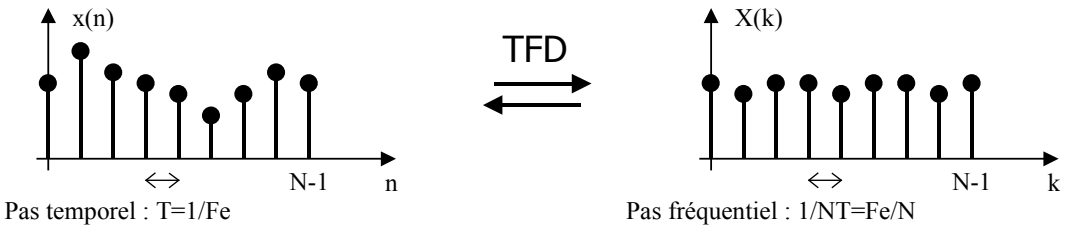


FIG. 4.1: TFD d'un signal discret à durée limitée

La figure 4.1 illustre la relation entre les pas fréquentiels et temporels. Étant donnée la définition de N permettant une reconstruction du signal temporel par sa transformée inverse, on peut aussi écrire :

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}}, \quad k = 0, \cdots, N-1 \tag{4.8}$$

On a la transformée de Fourier discrète inverse définie par :

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{j\frac{2\pi kn}{N}}, \quad n = 0, \cdots, N-1 \tag{4.9}$$

Les principales propriétés de la TFD sont énumérées ci dessous.
– Linéarité
– Décalage en temps-fréquence

$$x(n - n_0) \rightleftharpoons e^{-2j\pi\frac{kn_0}{N}} X(k)$$

– Produit de convolution circulaire

$$x_1(n) \circledast x_2(n) = \sum_{i=0}^{N-1} x_1(i) * x_2(n-i) \rightleftharpoons X_1(k).X_2(k)$$

avec $x_1(n)$ et $x_2(n)$ des signaux périodiques de période N.
– Théorème de Parseval (conservation de la puissance d'un signal)

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

– Propriétés de symétrie

$$x(n) \in \mathbb{R} \rightleftharpoons X(k) = X^*(N-k)$$

Un autre intérêt de cette transformée réside dans un calcul d'une somme finie de termes qui peut être particulièrement efficace si N est une puissance de 2, (Fast Fourier Transform, FFT).

Il est possible de trouver la transformée de Fourier discrète d'un signal, en évaluant sa transformée en z sur le cercle unité, c'est-à-dire en prenant $z = e^{j\frac{2\pi k}{N}}$.
Il est aussi possible d'exprimer $X(z)$ en fonction des X_k, on a :

$$X(z) = \frac{(1 - z^{-N})}{N} \sum_{k=0}^{N-1} \frac{X_k}{1 - e^{-j\frac{2\pi k}{N}} z^{-1}}$$

On a considéré les propriétés précédentes en prenant comme hypothèse que $x(n)$ est à temps limité. Soit maintenant un signal $x(n)$ quelconque, qu'il soit à temps limité ou non, périodique ou apériodique, on a alors les propriétés suivantes.

1. Pour un signal $x(n)$ périodique de période N, les valeurs $X(k)$ de la TFD sont les coefficients de la décomposition en série de Fourier du signal. La TFD est un calcul exact.
2. Pour un signal $x(n)$ quelconque, on a :

   $$x_p(n) = x(n), \quad \text{pour} \quad n = 0, \cdots, N-1$$

   Donc si on prend un signal à temps discret quelconque, on effectue une TFD sur N échantillons, puis une TFD inverse sur N points, on obtient exactement le signal d'origine *pour l'intervalle* $0 \leq n \leq N-1$.

3. Si $x(n)$ est à temps limité avec $L \leq N$, on a alors :

$$X_k = X(\Omega)|_{\Omega=\frac{2\pi k}{N}}$$

Si $x(n)$ n'est pas à temps limité, il existe toujours une relation en le spectre discret et le spectre continu mais qu'il est difficile d'interpréter. Comme la plupart du temps, on souhaite interpréter directement les X_k comme des composantes fréquentielles du signal on se débrouillera pour ne ne traiter que des signaux à temps limité (fenêtrage).

La transformée de Fourier Discrète est un outil efficace permettant de calculer la Transformée de Fourier à temps discret d'un signal à temps limité $x(n)$. La Transformée de Fourier Discrète est souvent utilisée pour des opérations de filtrage sous une forme rapide.

4.7 Convolution linéaire

Pour un système linéaire invariant (SLI), nous avons démontré dans la section 1.4 que l'on obtient la relation suivante entre signal d'entrée et de sortie :

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n-k) \tag{4.10}$$

Cette opération d'accumulation de termes multiplicatifs porte le nom de *convolution* et se note $*$, on a :

$$y(n) = x(n) * h(n)$$

L'opération de convolution à temps discret prend deux séquences $x(n)$ et $h(n)$ et produit une troisième séquence $y(n)$. L'équation 4.10 indique que chaque échantillon de la séquence $y(n)$ est formé d'une somme infinie des points de la séquence $x(k)$ et de $h(k)$ retournée et décalée d'un facteur n. La figure 4.2 illustre ce principe sur des séquences de durée finie. La flèche indique le calcul de $y(3) = \sum_{k=-\infty}^{+\infty} x(k)h(3-k)$. Dans notre cas la somme se limite à : $y(3) = \sum_{k=0}^{3} x(k)h(3-k)$.
L'opération de convolution linéaire est transformée en simple produit par la transformée en z :

$$y(n) = x(n) * h(n) \Rightarrow Y(z) = X(z).H(z)$$

4.7.1 Convolution périodique

Soit $x_1(n)$ et $x_2(n)$ deux séquences périodiques de période N et respectivement $X_1(k)$ et $X_2(k)$ leur coefficients de séries de Fourier discrète, si on forme le produit :

$$X_3(k) = X_1(k).X_2(k) \tag{4.11}$$

alors la séquence périodique $x_3(n)$ est donnée par :

$$x_3(n) = \sum_{k=0}^{N-1} x_1(k).x_2(n-k) \tag{4.12}$$

La démonstration de cette égalité s'effectue de manière simple en appliquant la TFD à $x_3(n)$ et en manipulant la double somme ainsi obtenue.

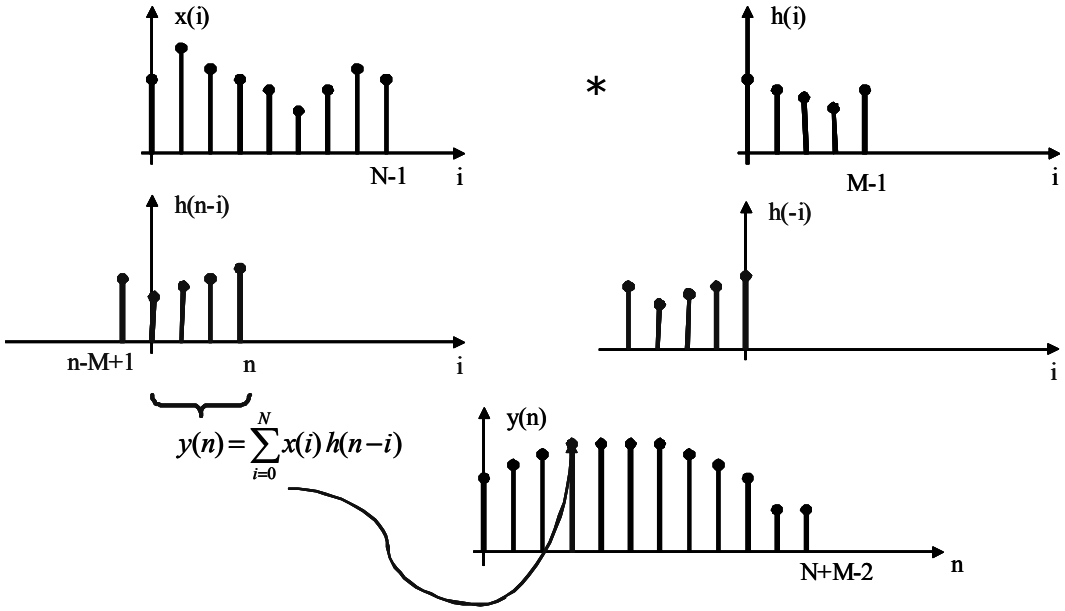


FIG. 4.2: Convolution de deux signaux discrets à durée limitée

L'équation 4.12 est également appelée convolution périodique. Les différences principales par rapport à la convolution non périodique sont que la somme est limitée à l'intervalle fini $0 \ldots N-1$ et que les valeurs de la séquence décalée $x_2(n-k)$ se répète périodiquement en dehors de l'intervalle de la somme.

4.7.2 Convolution circulaire

Dans cette section, nous étudions maintenant le cas où l'on effectue une convolution de deux séquences de durée finie $x_1(n)$ et $x_2(n)$ de longueur N, associées à leur TFD $X_1(k)$ et $X_2(k)$ et que l'on recherche à déterminer une séquence $x_3(n)$ dont la DFT est :

$$X_3(k) = X_1(k).X_2(k) \tag{4.13}$$

Pour cela, il suffit d'appliquer le résultat de l'équation 4.12 :

$$x_3(n) = \sum_{k=0}^{N-1} \tilde{x}_1(k).\tilde{x}_2(n-k), \qquad 0 \leq n \leq N-1 \tag{4.14}$$

où $\tilde{x}_1(k)$ et $\tilde{x}_2(n-k)$ sont des séquences périodiques de période N formée sur chaque période par les séquences répétées de $x_1(n)$ et $x_2(n)$. Ce résultat signifie que les séquences de durée finie sont considérées comme périodiques de période N est que leur décalage est réalisée de manière circulaire. On utilisera alors le terme de *convolution circulaire* que l'on notera :

$$x_3(n) = x_1(k) \circledast x_2(n-k) \tag{4.15}$$

Cette opération est, comme la convolution linéaire, commutative. La principale propriété de la convolution circulaire est qu'elle est transformée en un simple produit par la TFD. Ceci donnera naissance aux algorithmes dits de *convolution rapide* grâce à l'utilisation de la Transformée de Fourier Rapide (TFR) décrite dans le chapitre 5.

4.7.3 Convolution linéaire utilisant la TFD

Dans cette section nous nous intéressons au calcul efficace de la convolution linéaire de deux séquences $x_1(n)$ et $x_2(n)$ en utilisant leur TFD respective $X_1(k)$ et $X_2(k)$. Le résultat est le signal $x_3(n)$ dont la DFT est $X_3(k)$. Si l'on peut trouver une relation entre la convolution linéaire et la convolution circulaire définie dans la section précédente, alors le calcul coûteux de la convolution linéaire pourra être remplacée par la procédure suivante :

1. calcul des TFD sur N points des séquences $x_1(n)$ et $x_2(n)$,
2. calcul du produit $X_3(k) = X_1(k).X_2(k)$ pour $0 \leq k \leq N-1$ en utilisant éventuellement un algorithme de calcul rapide de la TFD,
3. calcul de la séquence $x_3(n)$ par TFD inverse de $X_3(k)$.

4.7.3.1 Cas des signaux à durée finie

Si on considère deux séquences $x_1(n)$ et $x_2(n)$, respectivement de durée finie N et M, la convolution linéaire de ces deux signaux est donnée par :

$$x_3(n) = \sum_{k=-\infty}^{+\infty} x_1(k)x_2(n-k), \qquad 0 \leq n < \infty \tag{4.16}$$

La figure 4.2 illustre cette équation sur des signaux $x(n)$ et $h(n)$ de durée $N = 9$ et $M = 5$. En observant la séquence $h(n-k)$ et pour des signaux causaux, il est aisé de comprendre que l'équation 4.16 précédente peut être réécrite de la manière suivante :

$$x_3(n) = \sum_{k=0}^{n} x_1(k)x_2(n-k), \qquad 0 \leq n \leq N+M-2 \tag{4.17}$$

Alors, $N+M-1$ est la longueur de la séquence $x_3(n)$ résultante de la convolution de $x_1(n)$ et $x_2(n)$. Par conséquent, la DFT $X_3(k)$ de $x_3(n)$ est également de longueur $N+M-1$. Ainsi, pour effectuer une convolution linéaire exacte de $x_1(n)$ et $x_2(n)$ de durée finie, il faudra appliquer la procédure suivante :

1. calcul des TFD sur $N+M-1$ points des séquences $x_1(n)$ et $x_2(n)$ complétées par des zéros,
2. calcul du produit $X_3(k) = X_1(k).X_2(k)$ pour $0 \leq k \leq N+M-2$ en utilisant éventuellement un algorithme de calcul rapide de la TFD,
3. calcul de la séquence $x_3(n)$ par TFD inverse de $X_3(k)$.

4.7.3.2 Cas des signaux à durée infinie

Dans le cas le plus général, où les séquences sont des signaux à support infini, il existe uniquement des méthodes approchées travaillant par blocs d'échantillons et utilisant la DFT pour le calcul de la convolution.
Le filtrage RIF est un cas particulier où une des séquences est à support fini (la réponse impulsionnelle $h(n)$ du filtre) et l'autre à support infini (le signal d'entrée du filtre). *Ces techniques ne sont pas détaillées dans ce support de cours, mais seront détaillées en cours.*

Chapitre 5

Transformée de Fourier Rapide

5.1 Naissance de la TFR

La transformation de Fourier rapide (TFR), ou encore Fast Fourier Transform (FFT), est directement issue d'une réorganisation du calcul des matrices de la transformée de Fourier discrète (TFD) dont la définition est donnée dans les équations 5.1 et 5.2. $X(k)$ est la TFD du signal $x(n)$. N points du signal $x(n)$ donnent N points de la TFD $X(k)$.

$$X(k) = \sum_{n=0}^{N-1} x(n).e^{(-2j\pi\frac{n.k}{N})},\ k = 0, 1 \dots N-1 \tag{5.1}$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k).e^{(2j\pi\frac{n.k}{N})},\ n = 0, 1 \dots N-1 \tag{5.2}$$

x(n) et X(k) sont, dans le cas général, des nombres complexes.
Nous pouvons réécrire l'équation 5.1 sous forme matricielle en faisant ainsi apparaître la complexité de l'algorithme de la TFD.

$$\begin{pmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{pmatrix} \times \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{pmatrix} \tag{5.3}$$

avec W_N le twiddle factor égal à $e^{(-2j\frac{\pi}{N})}$

La TFD revient à calculer un produit matrice-vecteur où chaque élément est de type complexe. La complexité de calcul de la TFD est donc de N^2 multiplications, et de $N(N-1)$ additions sur des nombres complexes. Ceci revient à une complexité de $4N^2$ multiplications réelles et $N(4N-2)$ additions réelles. Cet algorithme se comporte donc en $O(N^2)$, mais ne possède pas de problèmes d'adressage car les $x(n)$ et les W_N^k sont rangés dans l'ordre en mémoire.
En exploitant les propriétés des W_N^k (symétrie, périodicité, ...) :

$$W_N^{k(N-n)} = (W_N^{kn})^* \tag{5.4}$$
$$W_N^{k(n+N)} = W_N^{n(k+N)} = W_N^{kn} \tag{5.5}$$
$$W_N^{n+N/2} = -W_N^n \tag{5.6}$$
$$W_N^{2kn} = W_{N/2}^{kn} \tag{5.7}$$

on peut réduire le nombre d'opérations arithmétiques par un rapport de 2, mais la complexité globale reste en $O(N^2)$ [OS75].
De nombreux travaux ont essayé de réduire la complexité de la transformée de Fourier discrète, mais il a fallu attendre 1965 et la paire Cooley et Tuckey qui ont publié un algorithme applicable quand N est le produit de 2 ou plusieurs entiers. Cette publication allait engendrer de nombreuses recherches sur les algorithmes de calculs des transformées ; la FFT était née.
Depuis, et encore de nos jours, beaucoup d'algorithmes de traitement numérique du signal sont basés sur l'utilisation de cet algorithme efficace (ou de ses nombreux dérivés) pour réduire leur complexité.

Le principe fondamental de la TFR (en français dans le texte) repose sur la décomposition du calcul d'une séquence de N échantillons en TFD successives sur un nombre inférieur de points. Selon la manière dont ce principe est implanté, on obtient différents algorithmes, tous comparables au niveau de leur complexité.
– TFR partagée dans le temps ou DIT (decimation in time) :
  $x(n)$ est décomposé en sous séquence.
– TFR partagée en fréquences ou DIF (decimation in frequency) :
  $X(k)$ est décomposé en sous séquence.

5.2 TFR partagée dans le temps (DIT)

Depuis la formule 5.1, on partage $X(k)$ en une somme sur les termes d'indices pairs et une autre sur les termes d'indices impairs, dans notre exemple on se limitera au cas où N est une puissance de deux.

$$X(k) = \sum_{n\ pair} x(n).W_N^{nk} + \sum_{n\ impair} x(n).W_N^{nk} \tag{5.8}$$
$$X(k) = \sum_{n=0}^{N/2-1} x(2n).W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1).W_N^{(2n+1)k} \tag{5.9}$$
$$X(k) = \sum_{n=0}^{N/2-1} x(2n).W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1).W_{N/2}^{nk} \tag{5.10}$$
$$X(k) = G(k) + W_N^k.H(k) \tag{5.11}$$

où $G(k)$: TFD sur les $N/2$ points d'indices pairs,
$H(k)$: TFD sur les $N/2$ points d'indices impairs.

De plus, on peut réduire la complexité de calcul des échantillons $N/2$ à $N-1$ en calculant $X(k+N/2)$, puis en appliquant les propriétés de symétrie (voir équation 5.13. On obtient

ainsi le calcul d'un *papillon de la TFR.*

$$X(k+\frac{N}{2}) = \sum_{n=0}^{N/2-1} x(2n).W_{N/2}^{n(k+N/2)} + W_N^{(k+N/2)} \sum_{n=0}^{N/2-1} x(2n+1).W_{N/2}^{n(k+N/2)} \quad (5.12)$$

$$X(k+\frac{N}{2}) = G(k) - W_N^k.H(k) \quad (5.13)$$

L'application de ces équations est représentée graphiquement à la figure 5.1. On remarque qu'une TFD d'ordre N est décomposée en 2 TFD d'ordre $N/2$ suivies d'une recombinaison dont la complexité est en $O(N)$.

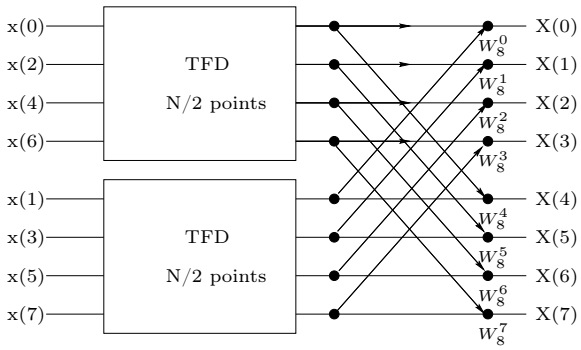


FIG. 5.1: Décomposition DIT de la TFD

Comme $N/2$ est également un nombre pair, on peut de nouveau partager chacune des TFD d'ordre $N/2$ en deux TFD d'ordre $N/4$ jusqu'à arriver à une TFD sur deux points pour laquelle le graphe flot est de complexité $O(N)$. Finalement il en résulte un graphe flot possédant $log_2 N$ étages de recombinaison (complexité $O'N$)) comme représenté à la figure 5.3 ; la complexité de la transformée de Fourier est enfin réduite à $O(Nlog_2 N)$.

L'opération élémentaire est ici appelée papillon DIT. On peut le représenter sous sa forme générale entre deux étages m et $m+1$ avec $m = 1, 2, \ldots, log_2 N$ (figure 5.2 et système d'équations 5.14).

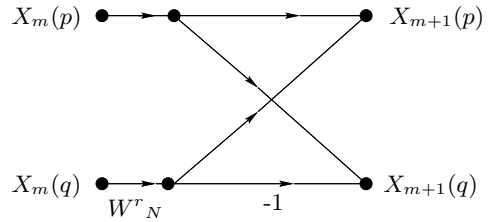


FIG. 5.2: Papillon DIT de la TFR

$$\begin{cases} X_{m+1}(p) = X_m(p) + W_N^r.X_m(q) \\ X_{m+1}(q) = X_m(p) - W_N^r.X_m(q) \end{cases} \quad (5.14)$$

Un papillon DIT, dont les équations sont données ci dessus (5.14) nécessite 1 multiplication, 1 addition et 1 soustraction sur des nombres complexes. Le graphe complet nécessite le calcul de $\frac{N}{2} \times log_2 N$ papillons ce qui donne donc en unité de calcul :
– $\frac{N}{2} log_2 N$ multiplications de nombres complexes,
– $N log_2 N$ additions/soustractions de nombres complexes, ou,

– $2N log_2 N$ multiplications de nombre réels[1],
– $3N log_2 N$ additions/soustractions de nombre réels[2].

Cette complexité précise peut servir à évaluer le temps de calcul d'une TFR complète sur n'importe quel processeur.

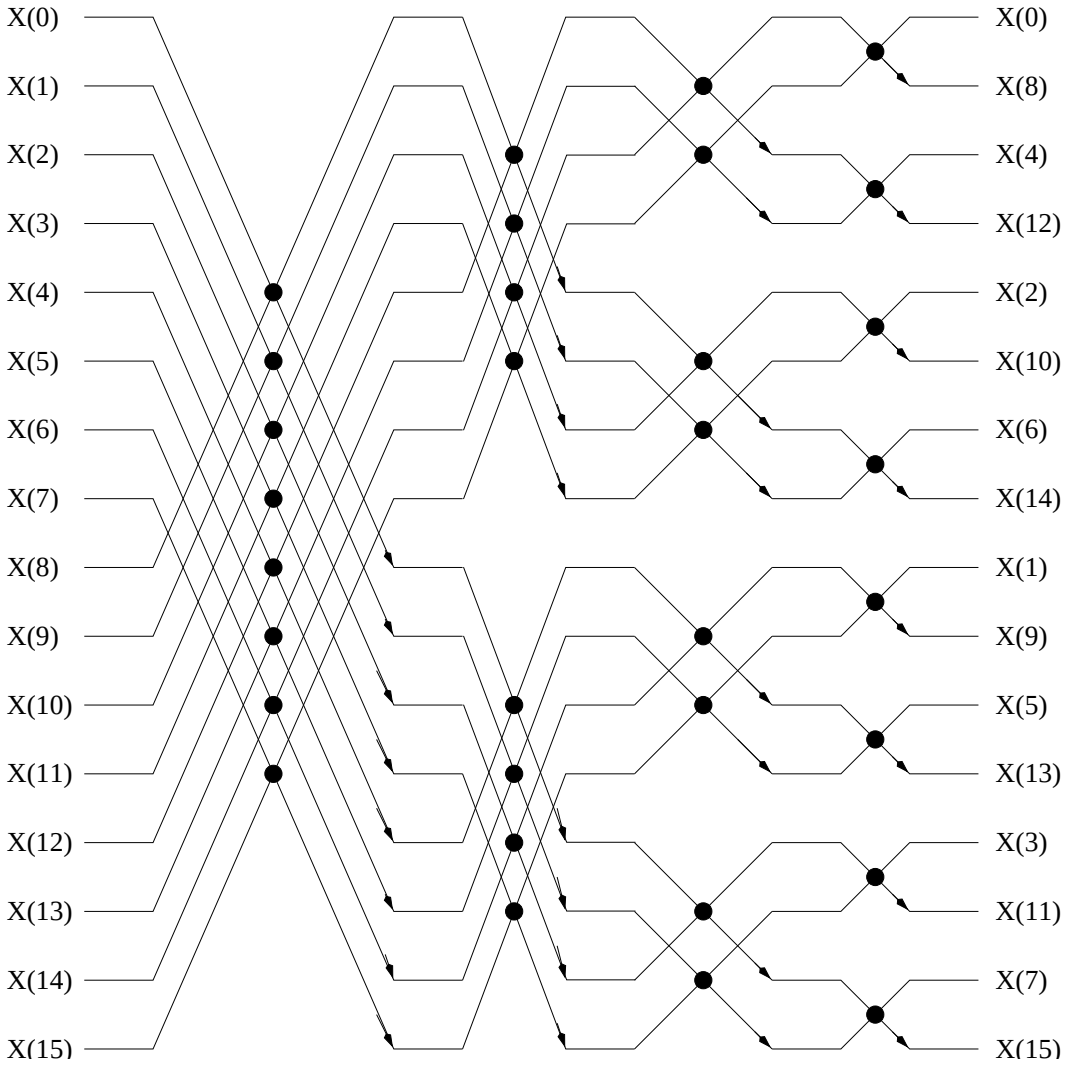


FIG. 5.3: Graphe d'une TFR DIF sur 16 échantillons

[1] 1 multiplication complexe requiert 4 multiplications et 2 additions sur des nombres réels.
[2] 1 addition complexe requiert 2 additions sur des nombres réels.

Si $X_m(.)$ et $X_{m+1}(.)$ peuvent être placés en mémoire sur le même vecteur on parle de calcul **en place**. De plus, le fait de faire apparaître la séparation pair/impair au départ (DIT) puis de séparer à nouveau en $N/4$, $N/8$,... groupes d'échantillons, implique en entrée des échantillons dont les adresses suivent un ordre appelé **bit-reverse** tel que :
– si la représentation en binaire de l'indice (n) est $(n_b \dots n_2 n_1 n_0)$ alors celle de son bit-reverse (n') est $(n_0 n_1 n_2 \dots n_b)$.

A l'opposé de la TFD, la TFR génère un calcul d'adresse non négligeable. Entre deux étages m et $m+1$, les W^k ne varient pas linéairement avec les $x(k)$. De plus, il est nécessaire de présenter à l'algorithme les échantillons dans un ordre non régulier (bit-reverse) ce qui engendre un calcul d'adresse supplémentaire. Ces problèmes ne sont pas négligeables - surtout sur un processeur général - car ils rajoutent à la complexité élémentaire une complexité d'adressage.

5.3 TFR partagée dans les fréquences (DIF)

Depuis la formule 5.1, on partage X(k) en une somme sur les $N/2$ premiers termes et une autre sur les $N/2$ autres termes :

$$X(k) = \sum_{n=0}^{N/2-1} x(n).W_N^{nk} + \sum_{n=N/2}^{N-1} x().W_N^{nk} \tag{5.15}$$

$$X(k) = \sum_{n=0}^{N/2-1} x(n).W_N^{nk} + W_N^{kN/2} \sum_{n=0}^{N/2-1} x(n+N/2).W_N^{nk} \tag{5.16}$$

$$X(k) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^k.x(n+N/2)\right] W_N^{nk} \tag{5.17}$$

On effectue une séparation des indices pairs et impairs :

$$X(2p) = \sum_{n=0}^{N/2-1} \left[x(n) + x(n+N/2)\right] W_N^{2pk} \tag{5.18}$$

$$X(2p+1) = \sum_{n=0}^{N/2-1} \left[x(n) - x(n+N/2)\right] W_N^{2pk}.W_N^n \tag{5.19}$$

$$X(2p) = \sum_{n=0}^{N/2-1} \left[x(n) + x(n+N/2)\right] W_{N/2}^{pk} \tag{5.20}$$

$$X(2p+1) = \sum_{n=0}^{N/2-1} \left(\left[x(n) - x(n+N/2)\right] W_{N/2}^{pk}\right).W_N^n \tag{5.21}$$

De même que pour le partage temporel, le remaniement des équations amène le calcul de deux TFD d'ordre inférieur selon le schéma de la figure 5.4.

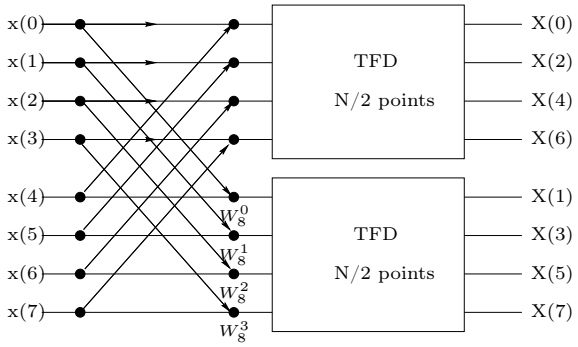


FIG. 5.4: Décomposition DIF de la TFD

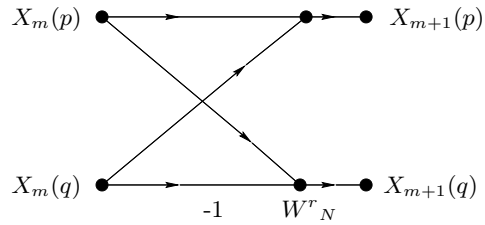


FIG. 5.5: Papillon DIF de la TFR

On peut également continuer à diviser ces transformées d'ordre inférieur comme dans le cas précédent (DIT) jusqu'à arriver au papillon unitaire dont le graphe est décrit à la figure 5.5 et dans le système d'équations 5.22).

$$\begin{cases} X_{m+1}(p) = X_m(p) + X_m(q) \\ X_{m+1}(q) = [X_m(p) - X_m(q)].W_N^r \end{cases} \tag{5.22}$$

La complexité élémentaire est identique à la TFR partagée dans le temps, la seule différence réside dans le calcul d'adresse de la table des W^k lors des étapes, qui est plus simple à programmer dans le cas DIF. De plus ce sont les échantillons en sortie qui sont placés en ordre bit-reverse.

5.4 Autres graphes de la TFR

Divers paramètres permettent de modifier la structure du graphe flot de données de la Transformée de Fourier Rapide (mode d'adressage, structure graphique, base du calcul), nous allons essayer d'en énumérer les principaux.

Le graphe à géométrie constante se caractérise par le fait que chaque passe d u calcul possède la même structure graphique que les autres passes. Celui-ci est directement issu de la décomposition de la matrice $[W]$ de la transformée de Fourier discrète en matrices de Good [Ka91] selon la relation ci dessous.

$$W = \prod_{i=1}^{log_2 N} D_i.C_i$$

avec : D_i des matrices diagonales, et C_i les matrices de Good.

Un exemple pour N=4 est représenté sous forme matricielle dans l'équation 5.23 puis sous son équivalent de graphe dans la figure 5.6 ci-dessous.

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & W \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \times \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{pmatrix} \quad (5.23)$$

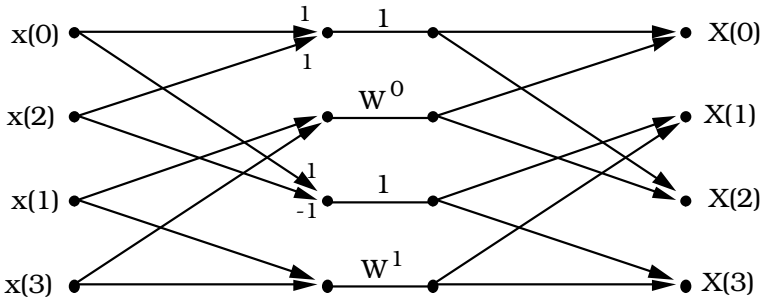


Fig. 5.6: Graphe d'une TFR à géométrie constante sur 4 échantillons

Un autre paramètre donnant des graphes de calcul différents est celui du bit-reverse. En effet on peut choisir l'entrée comme la sortie selon l'ordre normal ou l'ordre bit-reverse. Ceci a pour effet de modifier la forme du graphe ainsi que du calcul des indices des W_i. Cependant, si pour éviter un calcul d'adressage supplémentaire on désire obtenir un graphe dont à la fois les entrées comme les sorties sont dans l'ordre naturel, on passe dans une forme de calcul dit "non en place". Cette forme possède l'inconvénient majeur de perdre la notion de papillon. C'est à dire qu'entre deux passes successives on est obligé de mémoriser les vecteurs intermédiaires, ce qui a pour effet de doubler la mémoire nécessaire à l'algorithme.

On peut également utiliser d'autres bases de calcul de la transformée (nos exemples étant en base 2) qui consistent à séparer la TFD en plus de deux (quatre par exemple) TFD d'ordre inférieur ce qui a pour effet de réduire un peu encore la complexité ($O(Nlog_4 N)$ pour une TFR base (ou radix) 4). Par contre le papillon élémentaire voit son nombre de calcul augmenter. On peut citer l'exemple de la TFR DIF en base 4 dont la complexité en opération est $\frac{3N}{4} log_4 N$ multiplications complexes et $4Nlog_4 N$ additions complexes. Un exemple de graphe radix 4 est donné en annexe du document.

5.5 Annexes au chapitre sur la TFR

5.5.1 Graphe d'une TFR DIT radix 2 sur 16 points

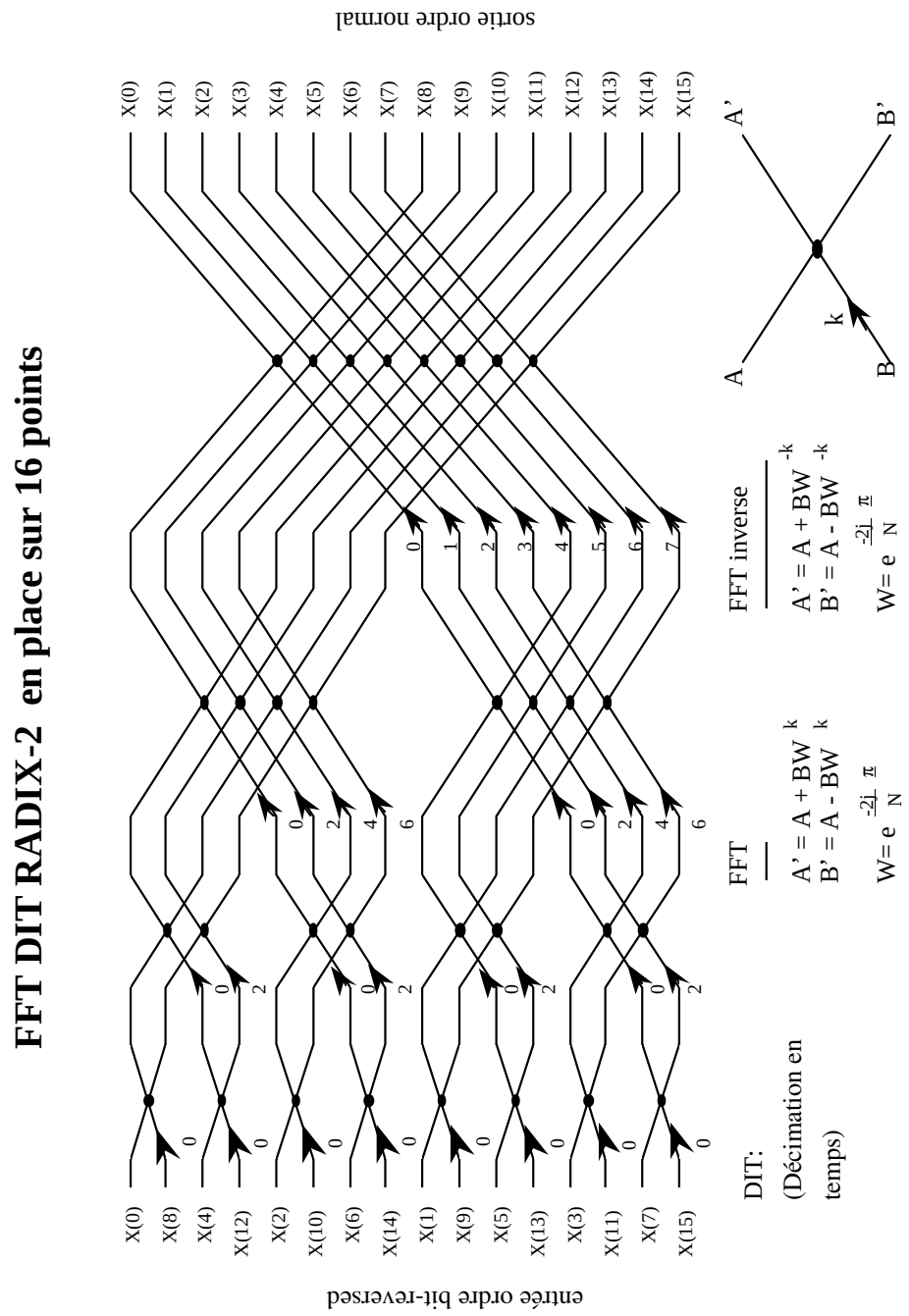


FIG. 5.7: Graphe d'une TFR DIT radix 2 sur 16 points

5.5.2 Graphe d'une TFR DIF radix 2 sur 16 points

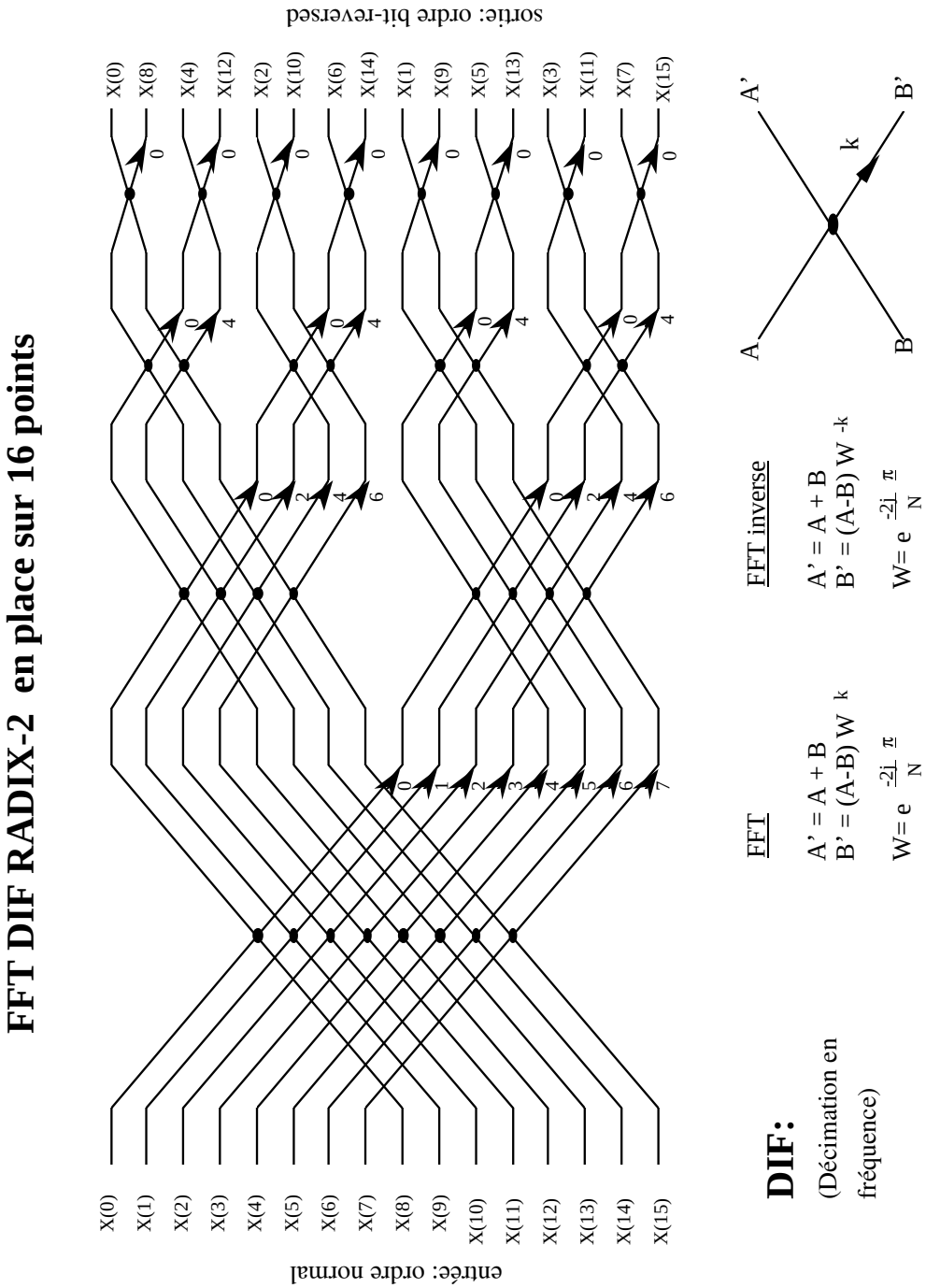


FIG. 5.8: Graphe d'une TFR DIF radix 2 sur 16 points

5.5.3 Graphe d'une TFR à géométrie constante sur 16 points

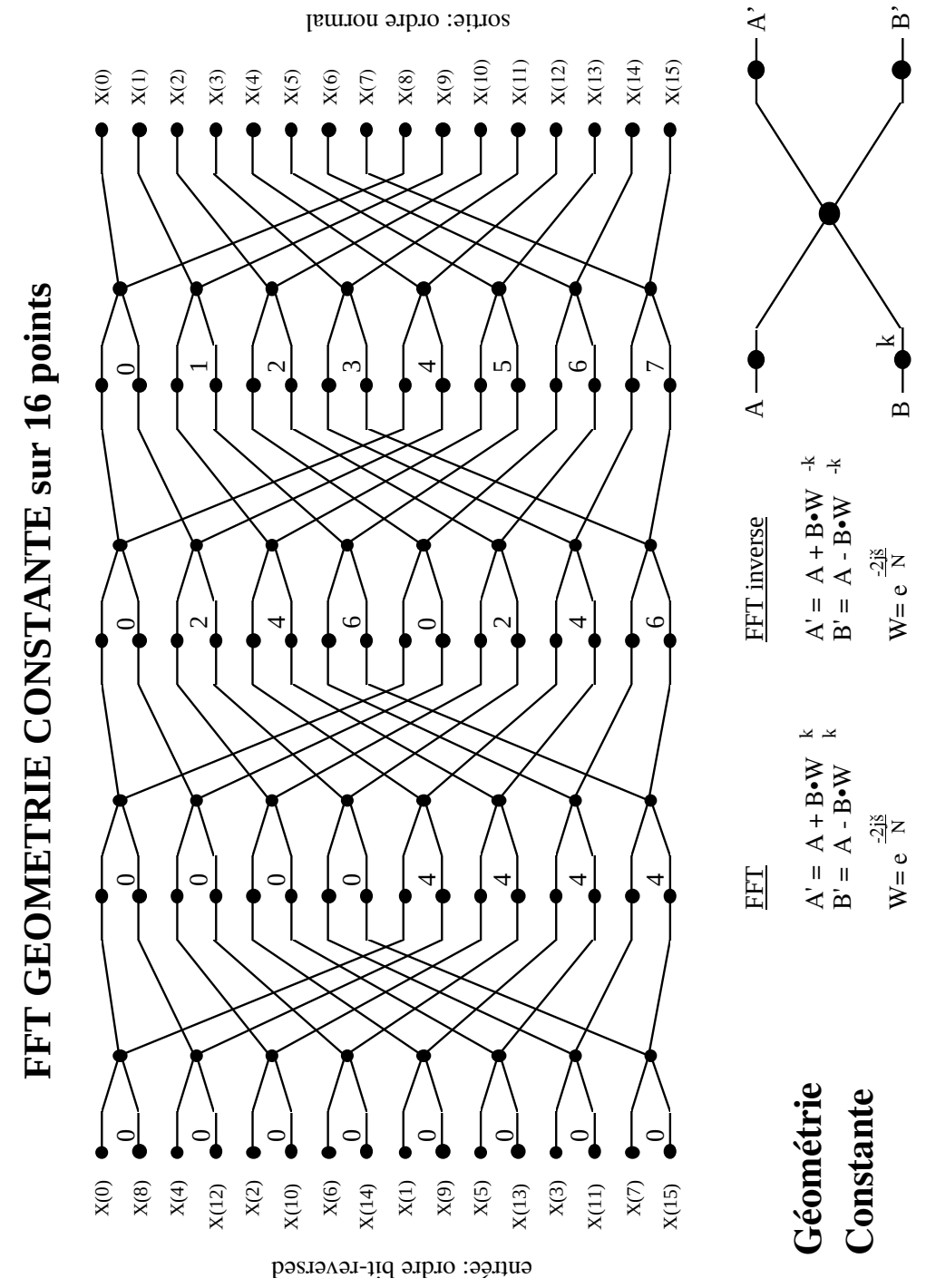


FIG. 5.9: Graphe d'une TFR à géométrie constante sur 16 points

5.5.4 Graphe d'une TFR DIF radix 4 sur 16 points

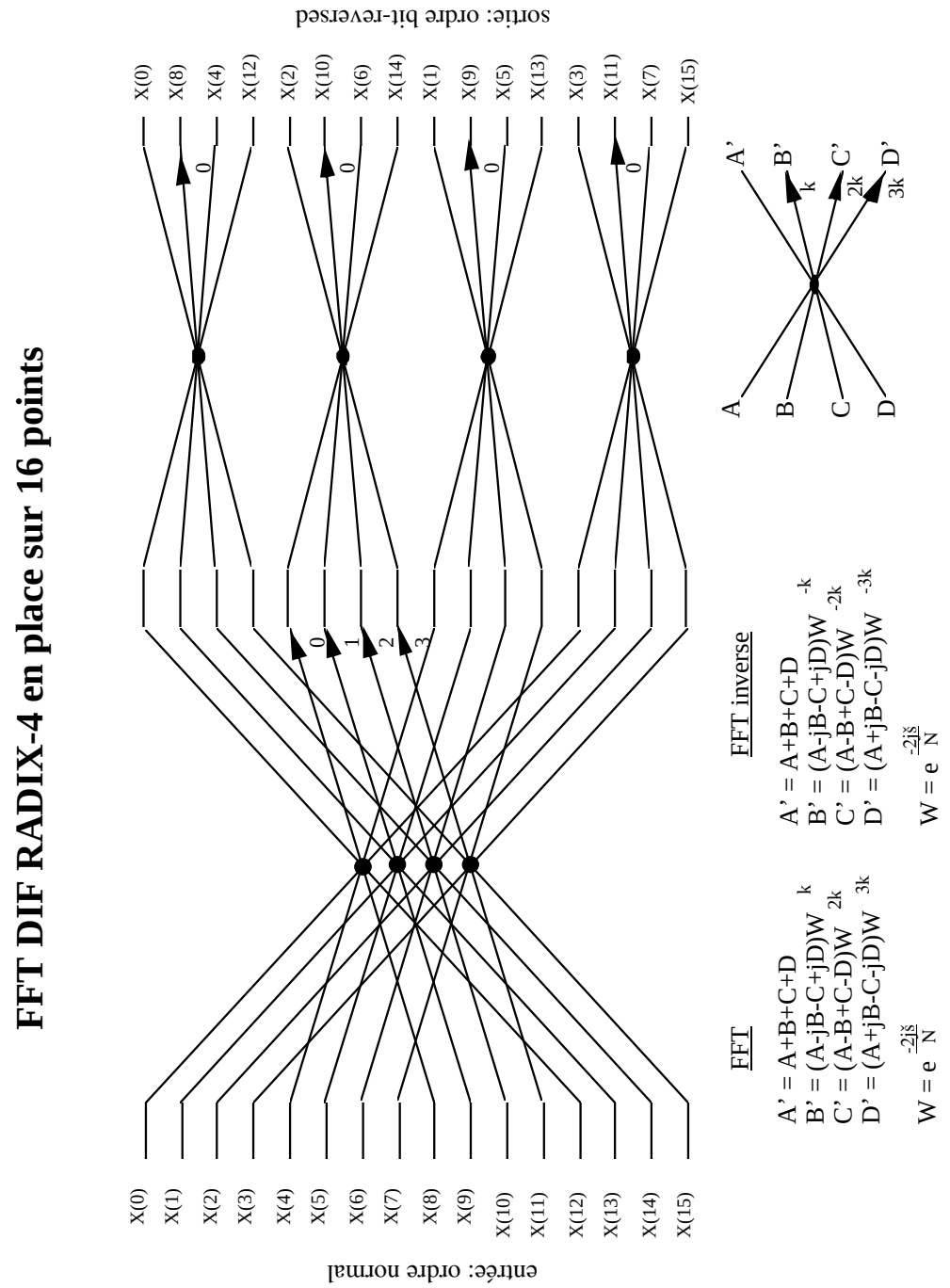


FIG. 5.10: Graphe d'une TFR DIF radix 4 sur 16 points

5.5.5 Algorithme DIF de la TFR radix 2 sur I points complexes

Complexité de calcul : $O(I.log_2 I)$
Nombre total de passes : $log_2 I$
Nombre de groupes dans une passe : 2^{passe}
Nombre de papillons calculés : $I.log_2 I/2$
Complexité d'un papillon : 2 additions et 1 multiplication complexes, ou 6 additions et 4 multiplications réelles.

```
PROCEDURE FFT (X, isign  )
    [I]complexe X: -- echantillons temporels et spatiaux
    int isign:     -- FFT / FFT inverse

        variables locales:

        [I]real: Wr, Wi:
        -- racines Nième de l'unité
        -- Wr[k]:=  COS(2p k / I)
        -- Wi[k]:= isign * SIN(2p k / I)
        int k,K:
        real C,S:

        int I1: -- nombre d'echantillons par groupe
        int I2: -- nombre de papillons par groupe
        int I3: -- nombre de groupe

        int i,l: -- indice des échantillons du papillons
        complexe t: -- variable temporaire

        DEBUT

             I2:=I;
             I3:=1;
             POUR passe DE 1 à LOG2I FAIRE

                I1:=I2; I2:= I2 / 2;    K:= I / I1;
                k:=0;

                POUR j DE 0 à I2-1 FAIRE

                     C:=Wr[k];  -- COS (2p k / I)
                     S:=Wi[k];  -- SIN(2p k / I)
                     k:=k + K;

                     POUR i DE j à I-1 PAR I1 FAIRE
                        l:=i + I2;

                        --- papillon X[i] X[l]
                        --- X[i] = X[i] + X[l]
                        --- X[l] = (X[i] - X[l])
                        t.r:=X[i].r - X[l].r;
                        t.i:=X[i].i - X[l].i;
                        X[i].r := X[i].r + X[l].r;
```

```
                X[i].i := X[i].i + X[l].i;
                X[l].r := C*t.r + S*t.i;
                X[l].i := C*t.i - S*t.r;
            FAIT
            I3:=I3 * 2;        -- nombre de groupes
        FAIT
    FAIT

    -- bit reverse : remettre la sortie dans l'ordre normal
    j:=1
    POUR i DE 1 à I-1 FAIRE
        SI i<j ALORS
            t:=X[j]
            X[j]:=X[i]
            X[i]:=t
        FINSI
        k:=I/2
        TANT QUE k<j FAIRE
            j:=j-k
            k:=k/2
        FAIT
        j:=j+k
    FAIT

    SI isign=-1 ALORS
        --- normaliser X dans le cas de la FFT inverse
        POUR tous les X[i] FAIRE
            X[i]:=X[i] / I
        FAIT
    FINSI

FIN
```

Chapitre 6

Filtrage numérique

6.1 Introduction au filtrage numérique

Il est difficile de donner une définition formelle de la notion de filtrage. L'ingénieur électronicien pense souvent à une modification des caractéristiques fréquentielles d'un signal donné d'entrée. D'un point de vue théorique, le domaine fréquentiel est couplé au domaine temporel, le filtrage modifie donc également la réponse dans ce dernier.

A une séquence d'échantillons d'un signal d'entrée à temps discret $x(n)$, un filtre numérique, défini par sa réponse impulsionnelle $h(n)$ ou par sa fonction de transfert en z $H(z)$, répond par une séquence d'échantillons d'un signal de sortie $y(n)$ (figure 6.1).

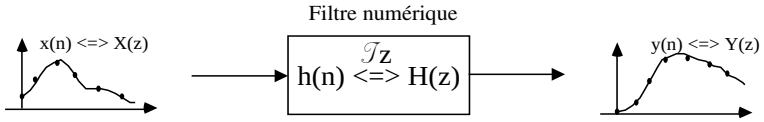


FIG. 6.1: Représentation sous forme de fonction de transfert en z

Des exemples de filtrage sont donnés ci après.
- Réduction de bruit pour des signaux radio, des images issues de capteurs, ou encore des signaux audio.
- Modification de certaines zones de fréquence dans un signal audio ou sur une image.
- Limitation à une bande fréquentielle pré-définie.
- Fonctions spéciales (dérivation, intégration, transformée de Hilbert, ...).
- Dans l'exemple du code DTMF (Digital Tone Multiple Frequency) utilisé en téléphonie, le signal transmis est la somme de deux sinusoïdes dont les fréquences sont normalisées (voir figure 6.2 gauche). Il résulte du choix de la touche appuyée sur votre téléphone. Ce principe est souvent qualifié de *fréquences vocales*. A la réception, pour reconnaître le numéro composé, une série de bancs de filtres est utilisée (voir figure 6.2 droite). Une première discrimination de deux zones fréquentielles est réalisée par un filtre passe haut et un filtre passe-bas. Puis, dans chaque zone, une série de filtres passe-bande suivis d'un détecteur permet de déterminer la présence d'une fréquence particulière.

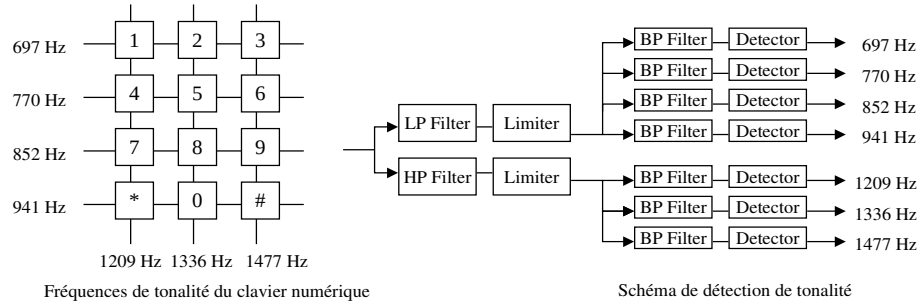


FIG. 6.2: Exemple du code DTMF en téléphonie

6.2 Représentation d'un filtre numérique

Un filtrage numérique peut être représenté en utilisant plusieurs types de spécifications.

1. **Fonction de transfert en z.** Ce mode de représentation est le plus usuel. Il permet de lier l'entrée et la sortie dans le plan z par $Y(z) = H(z).X(z)$. On posera dans la suite :

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{N} b_i.z^{-i}}{1 + \sum_{i=1}^{N} a_i.z^{-i}} \tag{6.1}$$

où $N(z)$ est le polynôme du numérateur de la fonction de transfert, tandis que $D(z)$ est son dénominateur. N est ici *l'ordre* du filtre. Dans le cas où $H(z)$ possède des pôles, on parlera de filtres RII (pour Réponse Impulsionnelle Infinie). Si $N(z) = 1$, on parlera de filtre tous-pôles. Dans le cas où $D(z) = 1$, le filtre ne possède que des zéros. Cette famille de filtre correspond au cas des filtres RIF (pour Réponse Impulsionnelle Finie). Celle ci n'a pas d'équivalent en filtrage analogique, et nous verrons que ses propriétés en font une fonction très utilisée en traitement numérique du signal.

L'équation 6.1 peut également être représentée en mettant en avant les pôles et les zéros.

$$H(z) = b_0 \frac{\prod_{i=1}^{N}(z - z_i)}{\prod_{i=1}^{N}(z - p_i)} \tag{6.2}$$

où p_i sont les pôles et z_i sont les zéros de $H(z)$. On rappelle ici que la stabilité du filtre sera déterminé par l'appartenance des pôles au cercle unité (i.e. $|p_i| < 1$), et que des zéros appartenant au cercle unité caractériseront un filtre à minimum de phase.
La figure 6.3 montre plusieurs versions de représentations de $H(z)$. La forme directe (figure6.3.a) peut être décomposée en produit ou en somme de fonctions de transfert d'ordre inférieur, généralement d'ordre 2. L'équation 6.3 et la figure 6.3.b représentent la forme parallèle, tandis que l'équation 6.4 et la figure 6.3.c représentent la forme cascade.

$$H(z) = \sum_{i=1}^{M} H_i(z) = \sum_{i=1}^{M} \frac{b_0 + b_1.z^{-1} + b_2.z^{-2}}{1 + a_1.z^{-1} + a_2.z^{-2}} \tag{6.3}$$

$$H(z) = \prod_{i=1}^{M} H_i(z) = \prod_{i=1}^{M} \frac{b_0 + b_1.z^{-1} + b_2.z^{-2}}{1 + a_1.z^{-1} + a_2.z^{-2}} \tag{6.4}$$

2. **Réponse impulsionnelle.** La réponse impulsionnelle est la fonction en z inverse de $H(z)$.

$$H(z) = \sum_{n=0}^{\infty} h(n).z^{-n} \tag{6.5}$$

Comme en filtrage analogique, la sortie d'un filtre $y(nT)$ est le résultat de la convolution du signal d'entrée représenté de manière temporelle $x(nT)$ avec la réponse impulsionnelle du filtre $h(nT)$. On a alors $y(nT) = x(nT) * h(nT)$, ou, si on fait abstraction de la période d'échantillonnage T :

$$y(n) = x(n) * h(n) = \sum_{k=0}^{\infty} x(k).h(n-k) = \sum_{k=0}^{\infty} x(n-k).h(k) \tag{6.6}$$

Dans le cas où $x(n)$ est une impulsion $\delta(n)$, on retrouve bien $y(n) = h(n)$.

Selon les cas où h(n) est à support infini ou fini, on retrouvera respectivement les deux types de filtres RII et RIF.

3. **Équation aux différences.** Une transformation en z inverse de l'équation 6.1 permet d'aboutir à la forme suivante :

$$y(n) = \sum_{i=0}^{N} b_i.x(n-i) - \sum_{i=0}^{N} a_i.y(n-i) \tag{6.7}$$

On identifie ici deux parties distinctes : une partie fonction de la valeur courante et des valeurs précédentes de l'entrée $x(n)$, et une partie fonction des valeurs précédentes de la sortie $y(n)$. Selon si les a_i sont non nuls ou nuls, on parlera donc de filtres *récursifs* ou de filtres *non récursifs*.

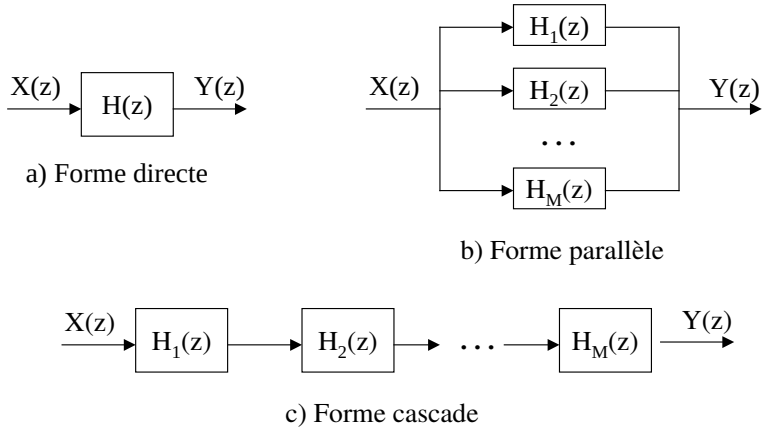


FIG. 6.3: Représentations sous forme de fonctions de transfert en z

6.3 Spécification d'un filtre numérique

Avant qu'un filtre numérique soit conçu et implanté, nous avons besoin de définir ses spécifications. Un filtre doit laisser passer certaines fréquences, alors qu'il doit en atténuer (voire éliminer) d'autres. Nous devons donc pouvoir représenter ces contraintes. Il y a quatre filtre de bases :

1. les *filtres passe-bas* laissent passer les fréquences inférieures à une fréquence de coupure f_c et bloquent celles qui lui sont supérieures (figure 6.4.a),
2. les *filtres passe-haut* bloquent les fréquences inférieures à une fréquence de coupure f_c et laissent passer celles qui lui sont supérieures (figure 6.4.b),
3. les *filtres passe-bande* laissent passer les fréquences autour d'une fréquence centrale f_0 (ou comprises entre f_1 et f_2) et bloquent les autres (figure 6.4.c),
4. les *filtres réjecteur-de-bande* bloquent les fréquences autour d'une fréquence centrale f_0 (ou comprises entre f_1 et f_2) et laissent passer les autres (figure 6.4.d).

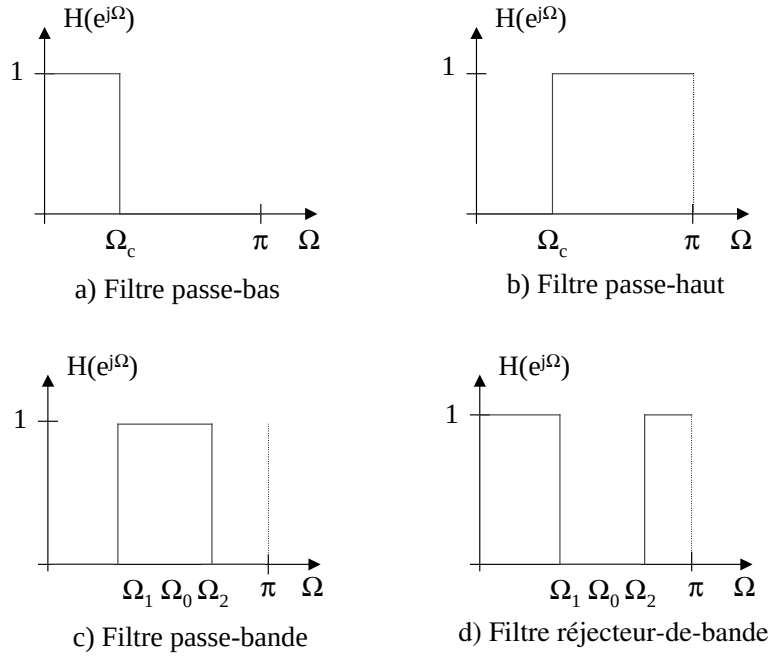


FIG. 6.4: Réponses fréquentielles idéales des 4 filtres de base

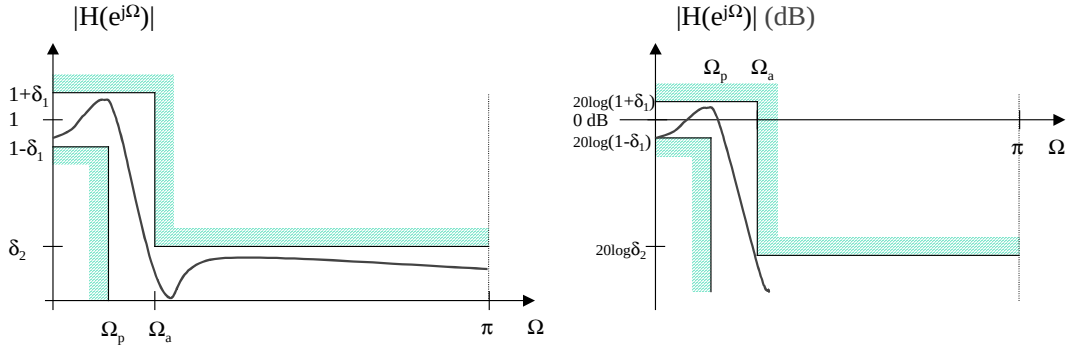
Les filtres représentés en figure 6.4 sont idéaux. Dans un cas réel il ne peut y avoir de discontinuités. Le passage entre zones passantes et zones atténuées se fait par des zones dites "de transition" dont la largeur va exprimer la *sélectivité* du filtre. Les bandes passantes et atténuées ne sont également pas idéales, elles contiennent des ondulations dont l'amplitude est exprimée par les paramètres d'*ondulation en bande passante* et d'*atténuation*.

Pour toutes ces raisons, la spécification d'un filtre est habituellement réalisée à partir d'un *gabarit* fréquentiel, défini entre 0 et π.

6.3.1 Spécifications des filtres passe-bas et passe-haut

Un filtre passe-bas possède trois zones : la bande passante ($0 \leq \Omega \leq \Omega_p$), la bande de transition ($\Omega_p \leq \Omega \leq \Omega_a$) et la bande atténuée ($\Omega_a \leq \Omega \leq \pi$). La figure 6.5.a montre une représentation graphique du gabarit fréquentiel linéaire d'un filtre passe-bas, tandis que la figure 6.5.b représente un gabarit fréquentiel en dB. Un filtre passe-haut verrait ses bandes atténuées et passantes inversées, on aurait dans ce cas $\Omega_p > \Omega_a$. δ_1 est l'ondulation en bande passante, δ_2 est l'atténuation.

La sélectivité du filtre est définie dans le tableau 6.1.



a) Gabarit fréquentiel linéaire b) Gabarit fréquentiel en dB

FIG. 6.5: Gabarit fréquentiel d'un filtre passe-bas

6.3.2 Spécifications des filtres passe-bande et réjecteur-de-bande

Un filtre passe-bande possède plusieurs zones : la bande passante ($\Omega_{p-} \leq \Omega \leq \Omega_{p+}$), deux bandes de transition et deux bandes atténuées ($0 \leq \Omega \leq \Omega_{a-}$ et $\Omega_{a+} \leq \Omega \leq \pi$). La figure 6.6 montre une représentation graphique du gabarit fréquentiel linéaire d'un filtre passe-bande. Un filtre réjecteur-de-bande verrait ses bandes atténuées et passantes inversées.

Le tableau 6.1 résume les paramètres des différents gabarits étudiés.

	Passe-bas	Passe-haut	Passe-bande	Réjecteur-de-bande
Sélectivité s	$\frac{\Omega_p}{\Omega_a}$	$\frac{\Omega_a}{\Omega_p}$	$\frac{\Omega_{p+}-\Omega_{p-}}{\Omega_{a+}-\Omega_{a-}}$	$\frac{\Omega_{a+}-\Omega_{a-}}{\Omega_{p+}-\Omega_{p-}}$
Ondulation	δ_1	δ_1	δ_1	δ_1
Atténuation	δ_2	δ_2	δ_2	δ_2
Fréquence centrale Ω_0	-	-	$\sqrt{\Omega_{p+}.\Omega_{p-}}$	$\sqrt{\Omega_{p+}.\Omega_{p-}}$
Largeur de bande B	-	-	$\frac{\Omega_{p+}-\Omega_{p-}}{\Omega_0}$	$\frac{\Omega_{p+}-\Omega_{p-}}{\Omega_0}$

TAB. 6.1: Paramètres de spécification d'un filtre numérique

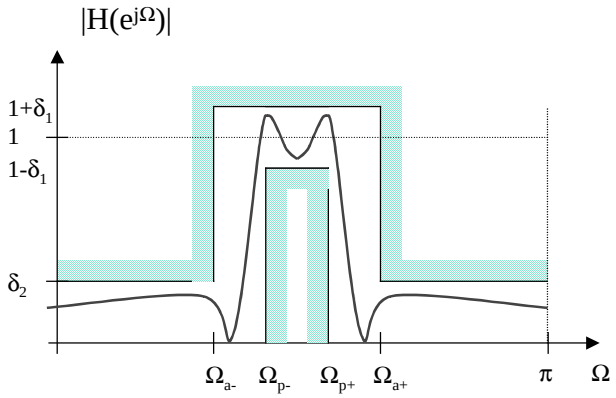


FIG. 6.6: Gabarit fréquentiel linéaire d'un filtre passe-bande

6.4 Classification des filtres numériques

Les filtres numériques peuvent être classés selon plusieurs critères :

1. la longueur de la réponse impulsionnelle implique deux types de filtres RII et RIF,
2. le type de représentation, ou de structure, implique deux types de filtres récursifs et non récursifs.

Nous verrons qu'à l'exception d'un cas particulier, les filtres récursifs et non récursifs sont respectivement équivalents aux filtres RII et RIF.

6.4.1 Filtres récursifs RII

Les filtres analogiques ont nécessairement une *réponse impulsionnelle infinie*. Les filtres numériques RII se comportent de manière similaire, mis à part les effets dus à la discrétisation. Cette catégorie de filtre est également caractérisée par une fonction de transfert en z contenant des pôles, et une équation aux différences récursives, c'est à dire lorsque la sortie $y(n)$ dépend à la fois des entrées et des sorties précédentes.
Les équations 6.8 et 6.9 montrent la fonction de transfert en z et l'équation aux différences correspondante de la forme générale d'un filtre RII. N est appelé ici l'ordre du filtre.

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{N} b_i.z^{-i}}{1 + \sum_{i=1}^{N} a_i.z^{-i}} \tag{6.8}$$

$$y(n) = \sum_{i=0}^{N} b_i.x(n-i) - \sum_{i=0}^{N} a_i.y(n-i) \tag{6.9}$$

A partir de l'équation 6.8, deux cas se présentent :

1. si $N(z)$ n'est pas divisible par $D(z)$, on a un nombre infini de termes dans la division

polynomiale de $N(z)$ par $D(z)$:

$$\begin{aligned} H(z) &= \sum_{n=0}^{N} c_n.z^{-n} \\ h(n) &= c_n \ pour \ n = 0 \dots \infty \end{aligned}$$

$H(z)$ est un filtre RII,

2. si $N(z)$ est divisible par $D(z)$, on a un nombre fini de termes dans la division polynomiale de $N(z)$ par $D(z)$:

$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} c_n.z^{-n} \\ h(n) &= c_n \ pour \ n = 0 \dots N-1 \end{aligned}$$

$H(z)$ est un filtre RIF.

Exemple 6.4.1 : Démontrer que le filtre moyenneur
$H(z) = \frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}} = \frac{1}{M}\sum_{i=0}^{M-1} z^{-i}$
peut être exprimé de manière récursive : $y(n) = s(n-1) + \frac{1}{M}[e(n) - e(n-M)]$
ou de manière non récursive : $y(n) = \frac{1}{M}\sum_{i=0}^{M-1} e(n-i)$

Les principales caractéristiques des filtres RII sont :

1. une bande de transition qui peut être étroite ;
2. des méthodes de synthèse par transposition des méthodes pour les filtres analogiques (voir chapitre 8) ;
3. une instabilité potentielle due à des pôles situés en dehors du cercle unité (i.e. $|p_i| \geq 1$ quel que soit i) ;
4. une instabilité numérique (i.e. après quantification des coefficients et du signal) potentielle due au rebouclage.

6.4.2 Filtres non récursifs RIF

Les filtres RIF ne peuvent pas être dérivés des filtres analogiques. Il sont cependant très largement utilisés car ils possèdent des propriétés uniques (phase linéaire, stabilité, flexibilité). Les équations 6.10 et 6.11 montrent la fonction de transfert en z et l'équation aux différences correspondante de la forme générale d'un filtre RII. N est appelé ici la longueur de la réponse impulsionnelle du filtre.

$$H(z) = \sum_{i=0}^{N-1} b_i.z^{-i} \tag{6.10}$$

$$y(n) = \sum_{i=0}^{N-1} b_i.x(n-i) = \sum_{i=0}^{N-1} h(i).x(n-i) \tag{6.11}$$

On remarque en exploitant l'équation 6.11 que les coefficients b_i du filtre sont également les valeurs de la réponse impulsionnelle $h(n)$, qui se trouve donc être limitée dans le temps.

$$H(z) = \sum_{i=0}^{N-1} b_i.z^{-i} \Longleftrightarrow h(n) = \sum_{i=0}^{N-1} b_i.\delta(n-i) \tag{6.12}$$

$$h(n) = \begin{cases} b_n & pour\ 0 \leq n \leq N-1 \\ 0 & ailleurs \end{cases} \tag{6.13}$$

Les principales caractéristiques des filtres RIF sont :

1. une bande de transition qui sera toujours plus large qu'un filtre RII ayant le même nombre de coefficients ;
2. des méthodes de synthèse permettant de dériver n'importe quelle réponse fréquentielle (voir chapitre 9 ;
3. une stabilité inhérente ($\sum_{n=0}^{N-1} |h(n)| < \infty$) ;
4. une plus grande stabilité numérique que les RII ;
5. une phase qui peut être exactement linéaire, par conséquent un temps de propagation de groupe constant et une absence de distorsion harmonique dans le signal ;
6. une plus grande facilité d'implantation dans un système numérique de traitement.

6.5 Analyse fréquentielle des filtres numériques

L'analyse fréquentielle est la représentation de la fonction de transfert du filtre dans le domaine des fréquences, c'est à dire celui de $\mathcal{F}$ourier. La fonction de transfert en Ω est la transformée de $\mathcal{F}$ourier du signal $h(n)$.

$$H(e^{j\Omega}) = \sum_{n=0}^{\infty} h(n).e^{-jn\Omega} = H(z)/_{z=e^{j\Omega}} \tag{6.14}$$

Cette fonction correspond à un signal discret, elle est donc périodique de période 2π. C'est pour cette raison que l'on a l'habitude d'utiliser la notation $H(e^{j\Omega})$ plutôt que $H(\Omega)$. La figure 6.7 représente un exemple de filtre passe-bas. En traitement du signal, on étudie généralement le module et la phase (ou argument) de la fonction complexe $H(e^{j\Omega})$.

$$H(e^{j\Omega}) = H_r(e^{j\Omega}) + jH_i(e^{j\Omega}) = |H(e^{j\Omega})|e^{j\Phi(\Omega)} \tag{6.15}$$

6.6 Structures des filtres RII et RIF

6.6.1 Structure des filtres RIF

La structure d'un filtre est un graphe flots de données (GFD) dans lequel les nœuds sont des opérations (les additions sont habituellement représentées par des cercles contenant un + et les multiplications par des triangles associés aux coefficients multiplicande) et les arcs les dépendances, c'est à dire le flot des données issues du signal. Certains arcs sont valués

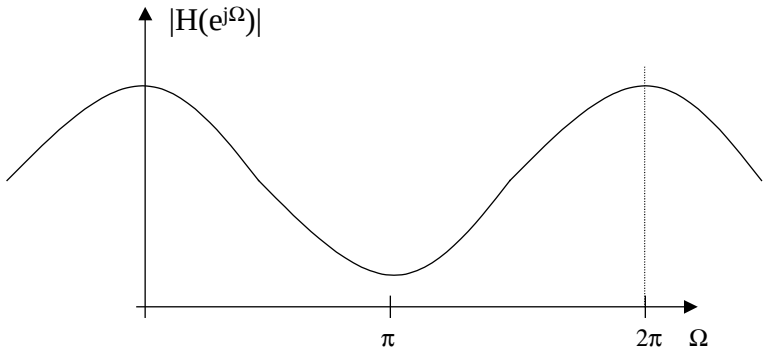


FIG. 6.7: Exemple de filtre passe-bas

d'un coefficients z^{-1} représentant un délai d'une période d'échantillonnage. Ce coefficient se représente également sous la forme d'un registre.

L'équation :

$$y(n) = \sum_{i=0}^{N} b_i.x(n-i) = b_0.x(n) + b_1.x(n-1) + \ldots + b_{N-1}.x(n-N+1) + b_N.x(n-N)$$

représente le comportement temporel d'un filtre RIF. On peut en déduire immédiatement la structure directe d'un filtre RIF qui est représentée à la figure 6.8.a. La structure transposée de la figure 6.8.b est obtenue après manipulation de cette équation.

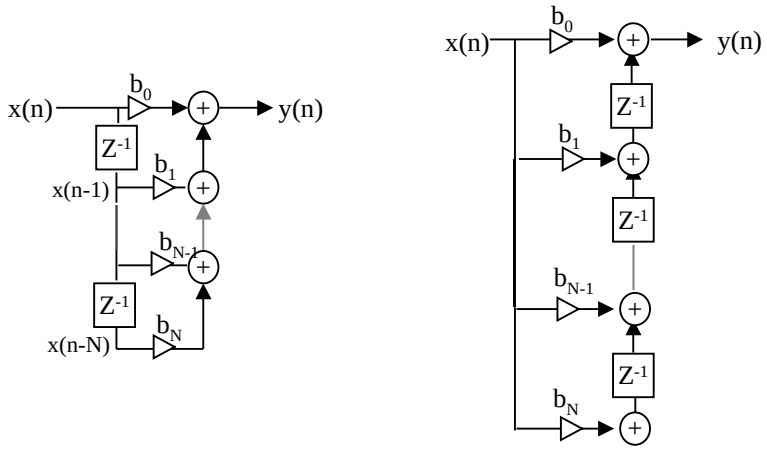


FIG. 6.8: Structures des filtres RIF

6.6.1.1 Complexité d'implantation d'un filtre RIF

Un filtre RIF necessite $N + 1$ opérations de multiplication, N opérations d'addition pour chaque nouvel échantillons à filtrer. On peut également exprimer la complexité en nombre

de multiplication-accumulation (MAC), qui, dans le cas du filtre RIF, vaut $N+1$. Le coût mémoire d'un filtrage RIF est de $2(N+1)$ ($N+1$ coefficients b_i et $N+1$ points mémoire pour le vecteur des entrées $x(i)$).
Si la fréquence d'échantillonnage du signal vaut F_e, cela signifie que le calcul d'un filtre devra être réalisé en un temps T_{calcul} inférieur à $T_e = \frac{1}{F_e}$.

Sur un processeur de type *DSP* capable d'exécuter une multiplication-accumulation (MAC) à chaque cycle, de puissance de calcul P_{calcul} exprimée en *MIPS (Million d'Instruction Par Seconde)*, Le temps de calcul sera : $T_{calcul} = (N+1).T_{cycle} = (N+1)/P_{calcul}$. Aussi, la puissance de calcul d'un DSP pour l'implantation d'un filtrage RIF vaut :

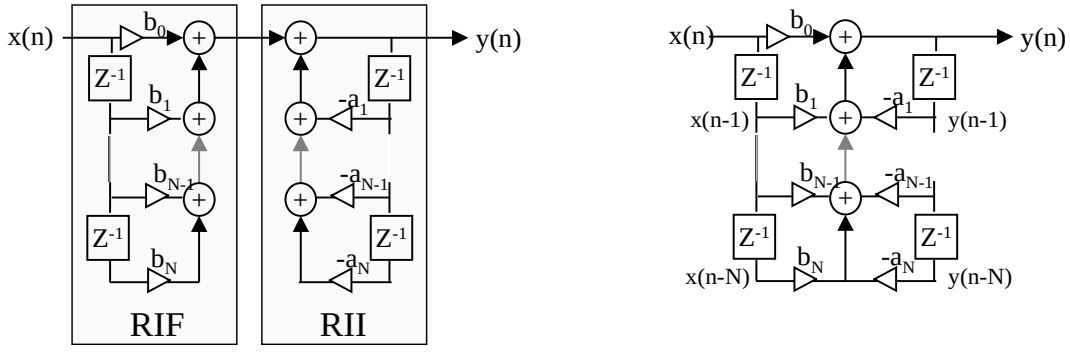
$$P_{calcul}\ (MIPS) > (N+1).F_e/10^6 \tag{6.16}$$

6.6.2 Structure des filtres RII

L'équation suivante (6.17) montre que l'on peut représenter un filtre RII $H(z)$ sous la forme du produit de 2 structures, dont une est une filtre RIF $N(z)$, et l'autre un filtre RII *tout-pôle* $1/D(z)$.

$$H(z) = \frac{N(z)}{D(z)} = [N(z)] \times \left[\frac{1}{D(z)}\right] = \left[\sum_{i=0}^{N} b_i.z^{-i}\right] \times \left[\frac{1}{1+\sum_{i=1}^{N} a_i.z^{-i}}\right] \tag{6.17}$$

La structure directe d'un filtre RII est donc obtenue en mettant en cascade un filtre RIF sous forme directe, et la représentation immédiate du filtre tout-pôle $1/D(z)$. Celle ci est donnée à la figure 6.9.a. En réunissant les additions du centre de la figure, on obtient la forme directe classique utilisée dans la littérature représentée à la figure 6.9.b.



a) Cascade **b) Structure directe**

FIG. 6.9: Structures directes des filtres RII

Il est bien entendu possible de représenter différemment un filtre RII en utilisant la propriété de commutativité de la multiplication. On a alors :

$$H(z) = \left[\frac{1}{D(z)}\right] \times [N(z)] = \left[\frac{1}{1+\sum_{i=1}^{N} a_i.z^{-i}}\right] \times \left[\sum_{i=0}^{N} b_i.z^{-i}\right] \tag{6.18}$$

On peut donc échanger sur la figure 6.9.a les 2 blocs RIF et RII. Les deux lignes à retard permettant de mémoriser les signaux x(n) se retrouvant communes, il est possible de les réunir en obtenant un vecteur unique de registres $w(n)$, comme représenté à la figure 6.10.a.

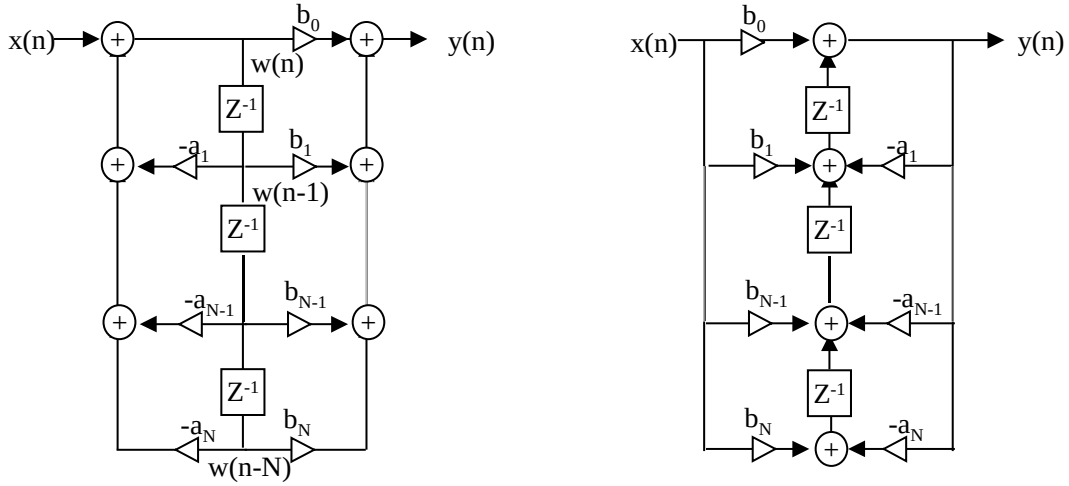


FIG. 6.10: Structures canoniques des filtres RII

On peut également représenter la structure canonique sous la forme d'un système d'équation (6.19) faisant apparaître le signal $w(n)$.

$$\begin{cases} W(z) = \frac{1}{D(z)}.X(z) \\ Y(z) = N(z).W(z) \end{cases} \quad \begin{cases} w(n) = x(n) - \sum_{i=1}^{N} a_i.w(n-i) \\ y(n) = \sum_{i=0}^{N} b_i.w(n-i) \end{cases} \tag{6.19}$$

6.6.2.1 Complexité d'implantation d'un filtre RII

Un filtre RII necessite $2N+1$ opérations de multiplication, $2N$ opérations d'addition pour chaque nouvel échantillons à filtrer ou $2N+1$ *MAC*. Le coût mémoire d'un filtrage RII en structure directe est de $4N+3$ ($2N+1$ coefficients b_i et $2(N+1)$ points mémoire pour les vecteurs des entrées $x(n)$ et des sorties $y(n)$). La structure canonique permet de diminuer le coût mémoire qui ne necessite plus que $N+1$ points mémoire pour le vecteur $w(n)$.
Si la fréquence d'échantillonnage du signal vaut F_e, cela signifie que le calcul du filtre devra être réalisé en un temps T_{calcul} inférieur à $T_e = \frac{1}{F_e}$.

Sur un processeur de type *DSP* capable d'exécuter une multiplication-accumulation (MAC) à chaque cycle, de puissance de calcul P_{calcul} exprimée en *MIPS (Million d'Instruction Par Seconde)*, Le temps de calcul sera : $T_{calcul} = (2N+1).T_{cycle} = (2N+1)/P_{calcul}$. Aussi, la puissance de calcul d'un DSP pour l'implantation d'un filtrage RIF vaut :

$$P_{calcul}\ (MIPS) > (2N+1).F_e/10^6 \tag{6.20}$$

Exemple 6.6.1 : Cellule élémentaire du premier ordre

Soit le système qui, à la suite de données $x(n)$, fait correspondre la suite $y(n)$ telle que :

$$y(n) = x(n) + b.y(n-1)$$

où b est une constante.

1. Donner les réponses impulsionnelles et indicielles de ce système. par deux méthodes (suite numérique, transformée en Z). Que peut on dire de la stabilité du filtre.
2. Étudier l'analogie avec le système continu de constante de temps t, échantillonné avec la période T.
3. Étudier la réponse fréquentielle du filtre.
4. Donner la structure de réalisation du filtre.

Exemple 6.6.2 : Cellule du second ordre purement récursive

Soit le système qui, à la suite de données $x(n)$, fait correspondre la suite $y(n)$ telle que :

$$y(n) = x(n) - b_1.y(n-1) - b_2.y(n-2)$$

1. Donner la fonction de transfert en Z du système.
2. En déduire la réponse impulsionnelle du filtre numérique.
3. Étudier la réponse fréquentielle du filtre. On regardera plus particulièrement l'influence des coefficients b_1 et b_2 sur les pôles de la fonction de transfert $H(z)$.
4. Tracer le diagramme des pôles et zéros.
5. Donner les structures de réalisation.

Chapitre 7

Effets de la quantification en traitement numérique du signal

Alfred S. se rend auprès de la société chaotique de banque afin d'effectuer un placement à long terme. Le banquier lui propose le placement suivant : « Vous faites un placement initial de $e - 1$ francs ($e = 2.718281459045\ldots$). La première année on multiplie votre capital par 1 et on prélève $1F$ de frais. La deuxième année on multiplie votre capital par 2 et on prélève $1F$ de frais. La n-ième année on multiplie votre capital par n et on prélève $1F$ de frais. Vous pouvez retirer l'argent après 25 ans. Intéressant, n'est-ce pas ? » La banque réalise une simulation sur son calculateur et obtient au bout de 25 ans $+4645987753F$. Alfred signe logiquement le contrat proposé. Rentré chez lui, il vérifie le calcul sur sa calculatrice de poche et trouve $\approx -140.10^{12}F$! En réalité un calcul symbolique donne un résultat d'environ 4 centimes. Cette anecdote prouve qu'un calcul peut être instable, même pour un petit nombre d'opérations (ici $25\otimes$ et $25\ominus$).

Les objectifs de ce chapitre sont de chercher à estimer la puissance du bruit générée par les différentes quantifications en traitement numérique du signal : conversion analogique numérique (quantification de l'amplitude du signal) et calculs en précision finie. Cela permet ensuite :

- soit de déterminer le nombre de bits nécessaires pour le CAN ou le processeur (DSP) pour un rapport signal à bruit (RSB) donné,
- soit, pour une machine fixée, de calculer le RSB et l'écart par rapport aux prévisions de simulation du système en précision infinie.

Dans ce chapitre nous présentons les caractéristiques du codage des données en virgule fixe, les conséquences de l'utilisation de données en précision finie et la modélisation du processus de codage. Tout d'abord nous détaillons les différents formats de codage des données et les paramètres associés puis nous exposons les règles de l'arithmétique virgule fixe. La seconde partie de ce chapitre concerne la modélisation de l'erreur induite par la quantification d'un signal analogique et celle issue des calculs en précision finie. Nous exposons les résultats de ces analyses et les conditions de validité de ceux-ci.

7.1 Les différents types de codage

7.1.1 Rappels sur le codage d'un entier

Un entier positif non signé x est codé en binaire sur b bits par :

$$x = \sum_{i=0}^{b-1} b_i 2^i \tag{7.1}$$

7.1.1.1 Représentation signe valeur absolue (SVA)

Pour cette représentation, la donnée x est composée d'un bit de signe S et de $b-1$ bits représentant le module de x. La valeur de cette donnée est la suivante :

$$x = (-1)^S \sum_{i=0}^{b-2} b_i 2^i \tag{7.2}$$

7.1.1.2 Représentation en complément à 2 (CA2)

La représentation en code complément à 2 de la donnée x est égale à :

$$x = -2^S S + \sum_{i=0}^{b-2} b_i 2^i \tag{7.3}$$

On écrira également que si $x > 0$ alors $-x = \bar{x} + 1$, où $\bar{x}$ est le complément de x. Le domaine de définition de ce code n'est pas symétrique par rapport à l'origine, il est composé de $2^{b-1} - 1$ valeurs positives et de 2^{b-1} valeurs négatives :

$$\mathcal{D} = [-2^{b-1}; 2^{b-1} - 1] \tag{7.4}$$

La représentation en code complément à 2 est très utilisée car elle possède des propriétés arithmétiques très intéressantes pour l'addition et la soustraction. Le résultat d'une série d'additions sera correct même si les résultats intermédiaires sont en dehors du domaine de définition du codage, il suffit que le résultat final appartienne au domaine de définition. De plus l'implantation dans les processeurs numériques des opérateurs traditionnels utilisant ce code est plus simple car elle nécessite une opérateur unique d'addition et de soustraction que les données d'entrée soient positives ou négatives.

Un nombre réel pourra être représenté par la multiplication d'un nombre entier par un coefficient $q < 1$. On parlera alors de virgule fixe.

7.1.2 Codage virgule fixe

Les données en virgule fixe sont composées d'une partie fractionnaire et d'une partie entière pour lesquelles le nombre de bits alloués reste figé au cours du traitement. L'exposant associé à chaque donnée est implicite et fixe. La figure 7.1 représente une donnée en virgule fixe composée d'un bit de signe et de $b-1$ bits répartis en m bits pour la partie entière et n bits pour la partie fractionnaire. Nous utilisons dans la suite du document la notation (b, m, n) pour

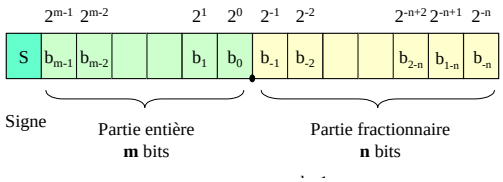


FIG. 7.1: Représentation des données en virgule fixe

définir le format d'une donnée. Nous trouvons aussi dans la littérature la notation Q_n.

Le format d'une donnée en virgule fixe est entièrement défini par la longueur de sa partie entière et de sa partie fractionnaire et de la représentation choisie. Nous présentons dans les parties suivantes les propriétés des représentations signe valeur absolue et complément à deux [Ka91].

7.1.2.1 Représentation signe valeur absolue (SVA)

Pour cette représentation, la donnée x est composée d'un bit de signe S et de $b-1$ bits représentant le module de x. La valeur de cette donnée est la suivante :

$$x = (-1)^S \sum_{i=-n}^{m-1} b_i 2^i \tag{7.5}$$

Ce type de représentation possède deux représentations de la valeur zéro ($10\ldots0$ et $00\ldots0$), ainsi le nombre de valeurs représentables N_c est égal à $2^b - 1$.

Le domaine de définition $\mathcal{D}$ correspond à l'intervalle regroupant l'ensemble des valeurs représentables par le code. Les bornes minimales et maximales de cet intervalle sont respectivement X_{min} et X_{max}. Dans le cas d'une représentation signe valeur absolue nous obtenons un domaine de définition symétrique par rapport à l'origine :

$$\mathcal{D}_R = [X_{min}; X_{max}] = [-2^m + 2^{-n}; 2^m - 2^{-n}] \tag{7.6}$$

La dynamique d'un code représente la différence entre la valeur minimale et maximale. Pour la représentation SVA, la dynamique est égale à :

$$D = X_{max} - X_{min} = 2.(2^m - 2^{-n}) \tag{7.7}$$

Le pas de quantification correspondant à la distance q entre deux valeurs successives, est fonction de la dynamique D et du nombre de valeurs représentables N_c :

$$q = \frac{D}{N_c - 1} = 2^{-n} \tag{7.8}$$

Le niveau de dynamique correspond au rapport entre les valeurs absolues maximales et minimales représentables par le code. L'expression du niveau de dynamique exprimé en dB, est la suivante :

$$N_{D\ dB} = 20 \log \left(\frac{\max(|x|)}{\min(|x|)} \right) \simeq 20.b.\log(2) \tag{7.9}$$

Représentation	cadrage à gauche	cadrage à droite
conditions	$m = 0$ $n = b - 1$	$n = 0$ $m = b - 1$
q	$2^{-(b-1)}$	1
$\mathcal{D}$	$[-1 + q; 1 - q]$	$[-2^{b-1} + q; 2^{b-1} - q]$

TAB. 7.1: Cas particuliers de la représentation signe valeur absolue

Représentation	cadrage à gauche	cadrage à droite	cadrage à n
condition	$m = 0$	$n = 0$	$n + m = b - 1$
q	$2^{-(b-1)}$	1	$2^{-(n)}$
$\mathcal{D}$	$[-1; 1 - q]$	$[-2^{b-1}; 2^{b-1} - q]$	$[-2^m; 2^m - q]$

TAB. 7.2: Cas particuliers de la représentation complément à 2

Deux représentations particulières liées à la position de la virgule sont couramment utilisées. Lorsque la virgule est cadrée à droite la valeur codée est entière et lorsque celle-ci est cadrée à gauche la donnée est fractionnaire. Les caractéristiques de ces deux représentations sont présentées dans le tableau 7.1.2.1. Plusieurs exemples de codage SVA sont présents dans le tableau 7.3.

7.1.2.2 Représentation en complément à 2 (CA2)

La représentation en code complément à 2 de la donnée x en virgule fixe est égale à :

$$x = -2^m S + \sum_{i=-n}^{m-1} b_i 2^i \tag{7.10}$$

Ce code a l'avantage de ne posséder qu'une seule représentation de la valeur zéro. Le domaine de définition de ce code n'est pas symétrique par rapport à l'origine, il est composé de $2^{b-1} - 1$ valeurs positives et de 2^{b-1} valeurs négatives :

$$\mathcal{D} = [-2^m; 2^m - 2^{-n}] \tag{7.11}$$

Le pas de quantification est identique à celui de la représentation précédente : $q = 2^{-n}$.

Les caractéristiques des représentations cadrées à gauche et cadrées à droite sont présentées dans le tableau 7.2. Plusieurs exemples de codage en CA2 sont présents dans le tableau 7.3.

7.1.3 Codage virgule flottante

Les données en virgule flottante sont composées d'un exposant et d'une mantisse représentés à la figure 7.2. L'exposant E permet d'obtenir un facteur d'échelle explicite et variable au cours du traitement, celui-ci est une puissance de 2. La mantisse représente la valeur de la donnée divisée par le facteur d'échelle. Afin d'éviter toute ambiguïté, le premier bit de la mantisse représente le coefficient $\frac{1}{2}$ et est fixé à 1. La valeur de ce bit restant fixe au cours du traitement, celui-ci n'est pas représenté dans le code.

Valeur			Représentation	
cadrage à gauche	cadrage à droite	$m = 3 \quad n = 2$	C.A.2	S.V.A.
0.96875	31	7.75	011111	011111
0.9375	30	7.5	011110	011110
⋮	⋮	⋮	⋮	⋮
0.3125	1	0.25	000001	000001
0	0	0	000000	000000
0	0	0		100000
-0.3125	-1	-0.25	111111	100001
⋮	⋮	⋮	⋮	⋮
-0.9375	-30	-7.5	100010	111110
-0.96875	-31	-7.75	100001	111111
-1	-32	-8	100000	...

TAB. 7.3: Exemples de codage

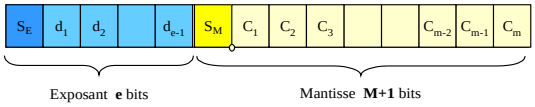


FIG. 7.2: Representation des données en virgule flottante

7.1.3.1 Représentation signe valeur absolue

La mantisse et l'exposant sont codés avec une représentation en signe valeur absolue, la valeur de la donnée x est la suivante :

$$x = 2^{u}.(-1)^{S_M}.\Big(\frac{1}{2} + \sum_{i=1}^{M} C_i 2^{-i-1}\Big) \quad \text{avec} \quad u = (-1)^{S_E}.\sum_{i=1}^{E-1} d_i 2^{i} \tag{7.12}$$

D'après l'équation 7.12 la valeur 0 n'est pas représentable, ainsi le domaine de définition est composé des deux sous intervalles suivants :

$$\mathcal{D}_R = \Big[-2^{K}; -2^{-K-1}\Big] \bigcup \Big[2^{-K-1}; 2^{K}\Big] \quad \text{avec} \quad K = 2^{E-1} - 1 \tag{7.13}$$

Le pas de quantification est fonction de la valeur représentée. Pour les valeurs de x comprises dans l'intervalle $[-2^{u}, -2^{u-1}] \bigcup [2^{u}, 2^{u-1}]$, le pas de quantification est égal à :

$$q = 2^{u}.2^{-(M+1)} \tag{7.14}$$

L'expression 7.15 détermine les bornes minimales et maximales du pas de quantification relatif. Nous pouvons considérer qu'il est pratiquement constant pour l'ensemble des valeurs de x.

$$2^{-(M+1)} < \frac{q}{|\,x\,|} < 2^{-M} \tag{7.15}$$

Le niveau de dynamique de cette représentation signe valeur absolue est égal à :

$$N_D = 20\log(2^{2K+1}) \quad \text{avec} \quad K = 2^{E-1} - 1 \tag{7.16}$$

La norme IEEE 744 utilise cette représentation en signe valeur absolue et est composée d'un exposant codé sur 8 bits et d'une mantisse sur 24 bits.

7.1.3.2 Représentation en complément à 2

Pour cette représentation la mantisse et l'exposant sont codés en complément à 2, ainsi la valeur de la donnée x est la suivante :

$$x = 2^E.\Big(- S_M + \frac{1}{2} + \sum_{i=1}^{M} C_i 2^{-i-1}\Big) \quad \text{avec} \quad E = -S_E + \sum_{i=1}^{E-1} d_i 2^i \tag{7.17}$$

Comme pour la représentation précédente le domaine de définition est composé de deux sous-ensembles :

$$\mathcal{D}_R = \Big[- 2^K; -2^{-K-2}\Big] \bigcup \Big[2^{-K-2}; 2^K\Big] \quad \text{avec} \quad K = 2^{E-1} - 1 \tag{7.18}$$

Le pas de quantification relatif est identique à celui calculé pour la représentation précédente.

7.2 Définition des règles de l'arithmétique virgule fixe

Nous considérons dans la suite de ce document que les données sont codées en virgule fixe avec une représentation en CA2.

7.2.1 Addition

L'addition de deux opérandes a et b nécessite qu'elles possèdent un format commun. Le type de représentation, la longueur de la partie entière et la longueur de la partie fractionnaire doivent être identiques pour les deux opérandes. Si cette condition n'est pas respectée il est nécessaire de modifier le format des opérandes afin d'obtenir un format identique (b_c, m_c, n_c). Le format commun garantissant l'absence de perte d'information est le suivant :

$$\begin{aligned} m_c &= \max(m_a, m_b) \\ n_c &= \max(n_a, n_b) \\ b_c &= m_c + n_c + 1 \end{aligned} \tag{7.19}$$

Pour les données ayant un format différent du format commun, il est nécessaire d'étendre le nombre de bits des parties entières et fractionnaires en suivant les règles suivantes :

- *partie fractionnaire :* les $(n_c - n_a)$ bits supplémentaires sont mis à 0.
- *partie entière :* **extension du bit de signe**. Dans le cas du complément à 2 les $(m_c - m_a)$ nouveaux bits prennent la valeur du bit de signe. L'extension de signe dans le cas d'une représentation signe valeur absolue est plus complexe, il faut décaler le bit de signe à la position b_{m_c-1} et mettre à zéro les bits b_{m_c-2} à b_{m_a-1}.

Le format du résultat de l'addition de deux opérandes au format (b_c, m_c, n_c) est présenté à l'expression 7.20. Nous obtenons un **débordement** si le résultat de l'addition des deux opérandes n'appartient pas au domaine de définition $\mathcal{D}_c = [-2^{m_c}; 2^{m_c}[$. Dans ce cas un bit supplémentaire est nécessaire pour coder la partie entière du résultat de l'addition.

$$n_{Add} = n_c \tag{7.20}$$

$$m_{Add} = \begin{cases} m_c + 1 & \text{si } a + b \notin \mathcal{D}_c \\ m_c & \text{si } a + b \in \mathcal{D}_c \end{cases} \tag{7.21}$$

7.2.2 Multiplication

Pour une multiplication, les deux opérandes doivent posséder la même représentation mais le nombre de bits réservés pour chaque partie peut être différent. Néanmoins, il est nécessaire avant d'effectuer l'opération, d'étendre le bit de signe. La multiplication de deux nombres en virgule fixe entraîne le doublement du bit de signe, celui-ci peut être éliminé automatiquement à l'aide d'un décalage à gauche. Pour un code en complément à 2 nous pouvons considérer que ce bit de signe redondant appartient à la partie entière. Le format du résultat de la multiplication de deux opérandes a et b est alors le suivant :

$$\begin{array}{c} m_{Mult} = m_a + m_b + 1 \\ n_{Mult} = n_a + n_b \\ b_{Mult} = b_a + b_b \end{array} \tag{7.22}$$

Dans le cas de la virgule fixe cadrée à gauche, le résultat de la multiplication de deux opérandes a et b appartenant à l'intervalle $[-1;1[$ reste dans le même intervalle. Il n'y a donc pas de débordement mais juste une augmentation de la précision du résultat. Cette propriété est très intéressante car elle élimine le problème du débordement. On verra plus tard que la multiplication pourra être modélisée par une quantification.

7.3 Processus de codage : lois de quantification et de dépassement

Soit x une valeur arbitraire appartenant au domaine $\mathcal{D}$ et y une valeur du domaine de définition $\mathcal{D}_\mathcal{R}$ du codage choisi. Le domaine $\mathcal{D}_\mathcal{R}$ est borné par les valeurs X_{min} et X_{max}. Nous définissons le sous-ensemble $\mathcal{D}_D$ de $\mathcal{D}$ regroupant l'ensemble des valeurs de $\mathcal{D}$ comprises dans l'intervalle $[X_{min}; X_{max}]$. Le processus de quantification correspond à l'opération de réduction d'une valeur arbitraire x à une valeur représentable y. Ce processus est régi par deux lois présentées ci-dessous :

Loi de dépassement : cette loi permet d'associer à l'ensemble des valeurs x de $\mathcal{D}$ une valeur x appartenant au domaine $\mathcal{D}_\mathcal{D}$. Elle définit plus précisément le comportement pour les valeurs présentes en dehors du domaine $\mathcal{D}_\mathcal{D}$. Nous associons à cette loi une fonction de dépassement définie ci-dessous :

$$f_D(x) = \begin{cases} x & \forall\ x \in \mathcal{D}_\mathcal{D} \\ D(x) & \forall\ x \notin \mathcal{D}_\mathcal{D} \end{cases} \tag{7.23}$$

Loi de quantification : cette loi définit les valeurs représentables y à associer à l'ensemble des valeurs x appartenant au domaine $\mathcal{D}_\mathcal{D}$. La fonction de quantification associée est la suivante :

$$f_Q(x) = Q(x) \quad \forall\ x \in \mathcal{D}_\mathcal{D} \tag{7.24}$$

Le processus de quantification global peut s'exprimer sous la forme suivante :

$$x \rightarrow f_Q(f_D(x)) = \begin{cases} Q(x) & \forall\ x \in \mathcal{D}_\mathcal{D} \\ D(x) & \forall\ x \notin \mathcal{D}_\mathcal{D} \end{cases} \tag{7.25}$$

7.3.1 Lois de dépassement

7.3.1.1 Arithmétique de saturation

Cette loi appelée *loi de saturation*, consiste à choisir la valeur du domaine $\mathcal{D}_\mathcal{D}$ la plus proche de la valeur à représenter x :

$$D(x) = \begin{cases} X_{min} & \forall\ x\ < X_{min} \\ X_{max} & \forall\ x\ > X_{max} \end{cases} \tag{7.26}$$

La caractéristique de cette fonction est représentée à la figure 7.3.a. La gestion de cette loi nécessitera des opérateurs arithmétiques spécifiques. Par contre, son utilisation permet d'éviter certains problèmes et erreurs engendrés par l'utilisation de la loi modulaire.

7.3.1.2 Arithmétique modulaire

Cette loi de dépassement modulaire substitue aux valeurs de x n'appartenant pas au domaine $\mathcal{D}_\mathcal{D}$, x *modulo* $(X_{max} - X_{min})$. La caractéristique de cette loi est présentée à la figure 7.3.b.

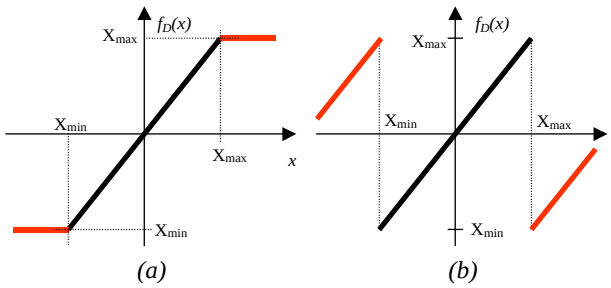


FIG. 7.3: Caractéristiques des lois de dépassement

7.3.2 Lois de quantification

Le domaine représentable $\mathcal{D}_\mathcal{R}$ est composé de N valeurs y_i avec $i = 1, 2, \ldots, N$ et le sous-domaine $\mathcal{D}_\mathcal{D}$ est subdivisé en N sous domaines juxtaposés Δ_i . La loi de quantification associe à tout x appartenant au domaine Δ_i la valeur y_i :

$$\forall\ x\ \in \Delta_i \quad Q(x) = y_i \tag{7.27}$$

7.3.2.1 Loi de quantification par arrondi

La loi de quantification par arrondi consiste à choisir la valeur représentable la plus proche de la valeur à quantifier en prenant la médiane de chaque intervalle Δ_i :

$$y_i = \frac{u_{i+1} - u_i}{2} = u_i + \frac{q}{2} \qquad \forall\ x\ \in \Delta_i = [u_i; u_{i+1}] \tag{7.28}$$

La caractéristique de cette loi est représentée à la figure 7.4.

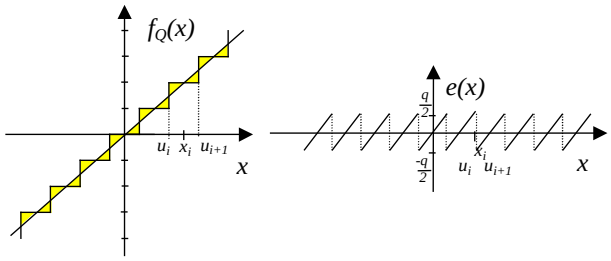


FIG. 7.4: Caractéristiques de la loi de quantification par arrondi

7.3.2.2 Loi de quantification par troncature

Cette loi de quantification consiste à tronquer un certain nombre de bits de poids faible. Les propriétés de cette loi de quantification sont fonctions du choix de la représentation (signe-valeur absolue ou complément à 2).

Représentation signe valeur absolue

Pour cette représentation la troncature d'un certain nombre de bits de poids faible revient à choisir la valeur représentable la plus proche dont le module est inférieur à la valeur à quantifier (voir figure 7.5) :

$$y_i = \begin{cases} u_i & \forall\ x > 0 \\ u_{i+1} & \forall\ x < \end{cases} \tag{7.29}$$

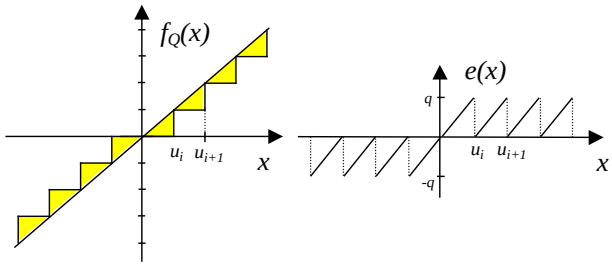


FIG. 7.5: Caractéristiques de la loi de quantification par troncature pour le codage SVA

Représentation en complément à 2

La quantification par troncature dans le cas d'une représentation en complément à 2 (voir figure 7.6) revient à prendre la valeur représentable immédiatement inférieure à la valeur à quantifier $y_i = u_i$.

7.4 Modélisation du processus de quantification

Dans cette partie nous nous intéressons à la modélisation du processus de quantification d'un signal analogique. Tout d'abord nous présentons l'analyse réalisée par Widrow qui permet de modéliser ce processus par un système linéaire où le signal quantifié est égal à la somme du

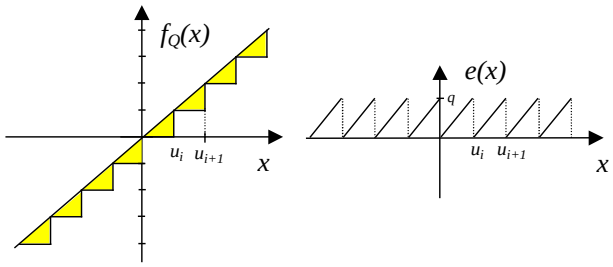


FIG. 7.6: Caractéristiques de la loi de quantification par troncature pour le codage CA2

signal d'origine et d'un bruit uniformément distribué. Ensuite nous exposons les travaux de Sripad et Snyder qui définissent les propriétés statistiques de l'erreur de quantification et les conditions de validité de cette approche. Dans les paragraphes 7.4.1 et 7.4.2, nous considérons uniquement une quantification par arrondi. Nous généraliserons ces résultats à la quantification par troncature dans le paragraphe 7.4.3.

7.4.1 Méthode de Widrow

7.4.1.1 Modélisation du processus de quantification

La quantification d'un signal x de densité de probabilité continue $p_x(x)$ conduit à un signal y de densité de probabilité discrete $p_y(y)$ composée de N valeurs p_k. L'expression de la densité de probabilité $p_y(y)$ est la suivante :

$$p_y(y) = \sum_{k=1}^{N} p_k \delta(y - k.q) \tag{7.30}$$

Chaque valeur p_k est égale à la probabilité que l'amplitude du signal x soit comprise dans l'intervalle $\Delta_k = [u_k, u_{k+1}]$. Elle correspond à l'aire de la densité de probabilité de x dans l'intervalle Δ_k :

$$p_k = \int_{\Delta_k} p_x(x) dx \tag{7.31}$$

Nous définissons la fonction $f_a(x)$ correspondant à l'aire de $p_x(x)$ dans l'intervalle Δ_k. Cette fonction est obtenue par convolution de la densité de probabilité $p_x(x)$ avec une fonction $f_q(x)$ dont la définition est la suivante :

$$f_q(x) = \begin{cases} 1/q & -q/2 < x < q/2 \\ 0 & \text{ailleurs} \end{cases} \tag{7.32}$$

La densité de probabilité discrete $p_y(y)$ correspond à l'échantillonnage de la fonction $f_a(x)$ réalisé à l'aide d'un peigne de dirac $f_d(x)$ de période q. Ainsi l'expression de cette densité de probabilité est la suivante :

$$p_y(y) = \Big(p_x(x) * f_q(x)\Big).f_d(x) \tag{7.33}$$

La fonction caractéristique d'une variable aléatoire est égale à la transformée de fourier inverse de sa densité de probabilité. En utilisant les différentes propriétés de la transformée de Fourier, nous obtenons l'expression de la fonction caractéristique du signal quantifié suivante :

$$\Phi_y(u) = \mathcal{F}^{-1}\big(p_y(y)\big) = \Big(\mathcal{F}^{-1}(p_x(x)).\mathcal{F}^{-1}(f_q(x))\Big) * \mathcal{F}^{-1}(f_d(x)) \tag{7.34}$$

soit

$$\Phi_y(u) = \frac{1}{q}\sum_{l=-\infty}^{\infty} \Phi_x(u + l\Psi).\Phi_q(u + l\Psi) \quad \text{avec } \Psi = 2\pi/q \tag{7.35}$$

La fonction caractéristique $\Phi_y(u)$ correspond à la duplication du produit des fonctions caractéristiques $\Phi_x(u)$ et $\Phi_q(u)$. A l'instar du théorème de Shannon pour l'échantillonnage des signaux analogiques, Widrow a proposé le théorème de quantification permettant de définir les conditions nécessaires pour reconstruire $p_x(x)$ à partir de $p_y(y)$ et vice-versa :

Théorème de quantification 1 *Si la fonction caractéristique de x est à bande limitée telle que :*

$$\Phi_x(u) = 0 \quad \text{pour } |u| > \frac{\Psi}{2} \tag{7.36}$$

Alors
- *la fonction caractéristique de x peut être obtenue à partir de celle de y*
- *la fonction caractéristique de y peut être obtenue à partir de celle de x*

Un exemple illustrant ce théorème est présenté à la figure 7.7.
Si la condition 7.36 du théorème de quantification est respectée, alors, en prenant la composante en bande de base de la fonction caractéristique $\Phi_y(u)$, il est possible de reconstruire $\Phi_x(u)$ à partir de $\Phi_y(u)$:

$$\Phi_y(u) = \Phi_x(u).\Phi_e(u) \tag{7.37}$$

$$\text{avec} \quad \Phi_e(u) = \frac{\sin(\frac{uq}{2})}{\frac{uq}{2}} \tag{7.38}$$

La transformée de Fourier de l'expression 7.37 permet de montrer que la densité de probabilité $p_y(y)$ est égale au produit de convolution des densités de probabilité $p_x(x)$ et $p_e(e)$. Cette dernière est obtenue par la transformée de Fourier de l'expression 7.38 :

$$p_e(e) = \frac{1}{q}\text{rect}\left(\frac{e}{q}\right) \tag{7.39}$$

Ainsi la variable aléatoire y est égale à la somme de deux variables aléatoires indépendantes x et e de densités de probabilité respectives $p_x(x)$ et $p_e(e)$. Ces résultats montrent que nous pouvons modéliser le processus de quantification par un système linéaire (figure 7.8) pour lequel la sortie est égale à la somme du signal d'entrée x avec une variable aléatoire e, appelée bruit de quantification ou erreur de quantification, dont la densité de probabilité est uniforme dans l'intervalle $[-q/2, q/2]$.

Les expressions des moments d'ordre 1 et 2 de l'erreur de quantification sont les suivantes :

$$\mu_e = \int_{-\infty}^{\infty} e\ p(e)de = \int_{-q/2}^{q/2} \frac{1}{q} e.de = 0 \tag{7.40}$$

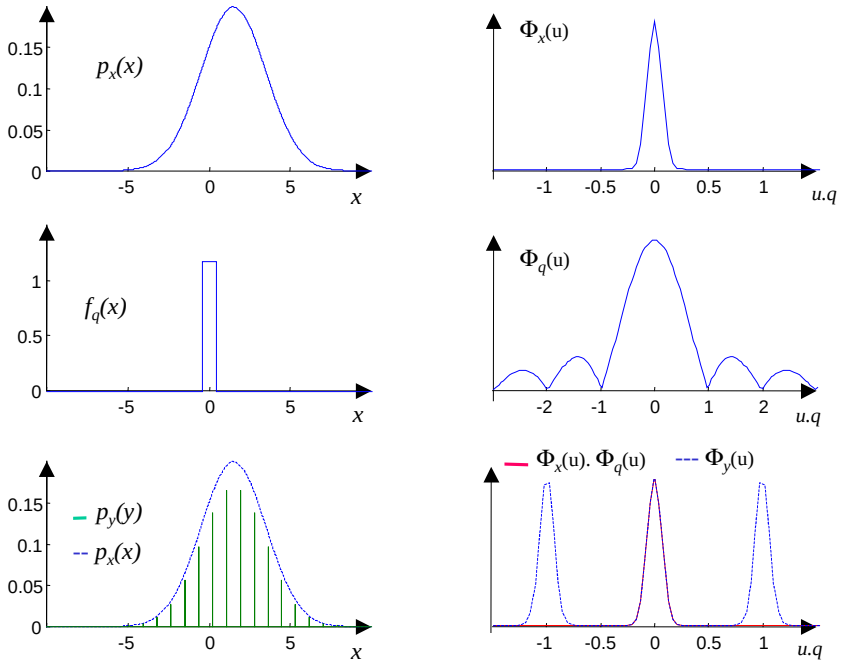


FIG. 7.7: Représentation des fonctions de distributions et de leur fonction caractéristique associée

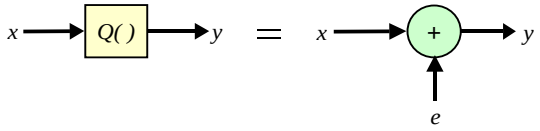


FIG. 7.8: Modélisation du bruit de quantification

$$\sigma_e^2 = \int_{-\infty}^{\infty} (e - \mu_e)^2 \; p(e) de = \int_{-q/2}^{q/2} \frac{e^2}{q} . de = \frac{q^2}{12} \tag{7.41}$$

7.4.2 Méthode de Sripad et Snyder

Sripad et Snyder proposent une modélisation de l'erreur de quantification identique à celle de Widrow mais les conditions de validité de leur approche sont moins restrictives. Le respect de ces conditions permet d'obtenir des propriétés intéressantes concernant le moment d'ordre 2 de l'erreur de quantification et la corrélation entre le signal d'origine x et l'erreur de quantification e.

7.4.2.1 Modélisation de l'erreur de quantification

La densité de probabilité de l'erreur de quantification e est déterminée à partir de celle du signal d'entrée x de la manière suivante :

$$p_e(e) = \begin{cases} \sum_{k=-\infty}^{\infty} p_x(e - k.q) & -q/2 < e < q/2 \\ 0 & \text{ailleurs} \end{cases} \tag{7.42}$$

soit

$$p_e(e) = \text{rect}\left(\frac{e}{q}\right)\Big[p_x(e) * f_d(e)\Big] \tag{7.43}$$

La fonction caractéristique de l'erreur de quantification obtenue à partir de la transformée de fourier inverse de l'expression 7.43, est égale à :

$$\Phi_e(u) = \mathcal{F}^{-\infty}\big(p_e(e)\big) = q.\frac{\sin(uq/2)}{uq/2} * \Big[\Phi_x(u).\frac{1}{q}\sum_{k=-\infty}^{\infty}\delta(u - ku_0)\Big] \qquad \text{avec} \quad u_0 = \frac{2\pi}{q} \tag{7.44}$$

soit après développement :

$$\Phi_e(u) = \sum_{k=-\infty}^{\infty}\Phi_x(ku_0)\frac{\sin[(u - ku_0)\frac{q}{2}]}{(u - ku_0)\frac{q}{2}} \tag{7.45}$$

Une condition nécessaire et suffisante pour que la densité de probabilité de l'erreur de quantification soit uniforme est que la fonction caractéristique de x soit nulle pour tout k entier différent de 0 :

$$\Phi_x(k\tfrac{2\pi}{q}) = 0 \quad \forall k \neq 0 \tag{7.46}$$

Sachant que $\Phi_x(0) = \int_{-\infty}^{\infty} p_x(x)dx = 1$, la fonction caractéristique de l'erreur de quantification est égale à celle de l'expression 7.38.

Cette condition est moins restrictive que celle proposée par Widrow. Ainsi elle permet d'augmenter la classe de signaux pour laquelle l'erreur de quantification est uniforme dans l'intervalle $[-q/2, q/2]$

7.4.2.2 Statistique d'ordre 2 de l'erreur de quantification

L'étude de la statistique d'ordre 2 de l'erreur de quantification permet d'étudier les propriétés spectrales de celle-ci. La densité de probabilité conjointe des variables aléatoires e_1 et e_2 représentant l'erreur de quantification aux instants t_1 et t_2, est égale à :

$$p_{e_1,e_2}(e_1, e_2) = \text{rect}\left(\frac{e_1}{q}\right)\text{rect}\left(\frac{e_2}{q}\right)\left[p_{x_1,x_2}(x_1, x_2) * \Big(f_d(e_1).f_d(e_2)\Big)\right] \tag{7.47}$$

Si nous suivons la même démarche que dans le paragraphe précédent, nous obtenons l'expression de la fonction caractéristique associée à $p_{e_1,e_2}(e_1, e_2)$ suivante :

$$\Phi_{e_1,e_2}(u_1, u_2) = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}\Phi_{x_1,x_2}(k_1u_0, k_2u_0)\frac{\sin[(u_1 - k_1u_0)\frac{q}{2}]}{(u_1 - k_1u_0)\frac{q}{2}}\frac{\sin[(u_2 - k_2u_0)\frac{q}{2}]}{(u_2 - k_2u_0)\frac{q}{2}} \tag{7.48}$$

Si la condition suivante, proposée par Sripad et Snyder est vérifiée

$$\Phi_{x_1,x_2}(k_1u_0, k_2u_0) = 0 \quad \forall k_1 \neq 0 \text{ et } \forall k_2 \neq 0 \tag{7.49}$$

alors nous obtenons l'expression de la fonction caractéristique associée à la densité de probabilité conjointe $p_{e_1,e_2}(e_1, e_2)$ présentée ci-dessous :

$$\Phi_{e_1,e_2}(u_1, u_2) = \frac{\sin[(u_1 - k_1u_0)\frac{q}{2}]}{(u_1 - k_1u_0)\frac{q}{2}} \frac{\sin[(u_2 - k_2u_0)\frac{q}{2}]}{(u_2 - k_2u_0)\frac{q}{2}} \tag{7.50}$$

Ainsi l'expression de la densité de probabilité conjointe $p_{e_1,e_2}(e_1, e_2)$ est égale à :

$$p_{e_1,e_2}(e_1, e_2) = \frac{1}{q}\text{rect}\left(\frac{e_1}{q}\right).\frac{1}{q}\text{rect}\left(\frac{e_2}{q}\right) = p_{e_1}(e_1).p_{e_2}(e_2) \tag{7.51}$$

D'après l'expression 7.51, les erreurs de quantification sont statistiquement indépendantes, ainsi le moment d'ordre 1 associé à $p_{e_1,e_2}(e_1, e_2)$ est égal à :

$$E(e_1, e_2) = E(e_1).E(e_2) \tag{7.52}$$

Les moments d'ordre 1 des variables aléatoires e_1 et e_2 étant nuls, nous obtenons la fonction d'autocorrélation de l'erreur de quantification suivante :

$$\varphi_{ee}(\tau) = \begin{cases} q^2/12 & \tau = 0 \\ 0 & \text{ailleurs} \end{cases} \tag{7.53}$$

Cette fonction d'autocorrélation étant égale à $q^2/12.\delta(\tau)$, l'erreur de quantification est assimilable à un bruit blanc centré.

7.4.2.3 Étude de la corrélation entre le signal d'entrée et l'erreur de quantification

La corrélation entre l'erreur de quantification et le signal d'entrée x peut être obtenue à partir de l'analyse du moment d'ordre 2 de la variable aléatoire y représentant la somme de x et de e. L'expression du moment d'ordre 2 de cette somme est la suivante :

$$E(y^2) = E(x^2) + E(e^2) + 2E(x, e) = E(x^2) + E(e^2) + R_{xe} \tag{7.54}$$

Le moment d'ordre 2 de y est égal à la valeur à l'origine de la dérivée seconde de la fonction caractéristique de y :

$$E(y^2) = (-j)^2 \frac{d^2}{du^2}\Phi_y(u)\bigg|_{u=0} \tag{7.55}$$

soit d'après l'expression 7.35

$$E(y^2) = (-j)^2 \sum_{k=-\infty}^{\infty} \left[\Phi''_x(-ku_0)\frac{sin(\pi k)}{\pi k} + q.\Phi'_x(-ku_0)\frac{\sin(\pi k) - \pi k\cos(\pi k)}{(\pi k)^2} + \Phi_x(-ku_0).\Phi''_e(-ku_0)\right] \tag{7.56}$$

Si la condition 7.46 est vérifiée alors nous pouvons simplifier l'expression 7.56 de la manière suivante : [1] [2]

[1] $\text{sinc}(\pi k) = 0 \quad \forall k \neq 0$

[2] A l'aide d'un développement limité nous obtenons

$$\lim_{u\to 0} \frac{\sin(u) - u.\cos(u)}{u^2} = \lim_{u\to 0} \frac{u}{6} + \varepsilon(u) = 0$$

$$E(y^2) = \Phi''_x(0) + \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} q.\Phi'_x(-ku_0)\frac{(-1)^k}{\pi k} + \Phi_x(0).\Phi''_e(0) \tag{7.57}$$

Par identification de l'expression 7.57 avec 7.54 nous pouvons en déduire la corrélation entre e et x :

$$E(xe) = \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} q.\Phi'_x(-ku_0)\frac{(-1)^k}{\pi k} \tag{7.58}$$

Si la condition 7.59 est vérifiée alors $E(xe)$ est nulle. Dans ce cas l'erreur de quantification e n'est pas corrélée avec le signal d'entrée x.

$$\Phi'_x(k\tfrac{2\pi}{q}) = 0 \quad \forall k \neq 0 \tag{7.59}$$

L'expression 7.58 correspond à la déviation entre la corrélation réelle et la corrélation issue de la modélisation de l'erreur de quantification e par un bruit uniformément distribué. Dans le cas d'un signal distribué selon une loi normale, la condition 7.46 n'est pas respectée, mais la déviation issue de la modélisation est très faible dès que le pas de quantification est inférieur à l'écart type du signal. Ceci a été confirmé par les simulations présentées dans le paragraphe 7.4.4.

7.4.3 Extension à la quantification par troncature

Nous pouvons étendre l'analyse effectuée par Sripad et Snyder au cas de la quantification par troncature dans le cas d'un codage en complément à 2. L'expression de la densité de probabilité de l'erreur de quantification est la suivante :

$$p_{e_t}(e) = \mathrm{rect}\left(\frac{e - q/2}{q}\right)\Big[p_x(e) * f_d(e)\Big] \tag{7.60}$$

En suivant la même démarche que celle présentée dans le paragraphe 7.4.2 de la page 82 nous obtenons l'expression de la fonction caractéristique de l'erreur de quantification suivante :

$$\Phi_{e_t}(u) = \Phi_e(u).e^{j\frac{u.q}{2}} \tag{7.61}$$

Ainsi si la condition 7.46 présentée à la la page 83, est respectée alors la densité de probabilité de l'erreur de quantification est égale à :

$$p_{e_t}(e) = \frac{1}{q}\mathrm{rect}\left(\frac{e - q/2}{q}\right) \tag{7.62}$$

Les expressions des moments d'ordre 1 et 2 de l'erreur de quantification sont égales à :

$$\mu_{e_t} = \int_{-\infty}^{\infty} e\ p(e)\ de = \int_0^q \frac{1}{q}\ e\ de = \frac{q}{2} \tag{7.63}$$

$$\sigma_{e_t}^2 = \int_{-\infty}^{\infty} (e - \mu_e)^2 \; p(e) de = \int_0^q \frac{1}{q} \Big(e - \frac{q}{2}\Big)^2 \; de = \frac{q^2}{12} \tag{7.64}$$

D'après les expressions 7.63, 7.64, 7.52 et 7.53 nous pouvons en déduire que la fonction d'autocorrélation de l'erreur de quantification est égale à :

$$\varphi_{ee_t}(\tau) = \frac{q^2}{12}\delta(\tau) + \frac{q^2}{4} \tag{7.65}$$

D'après l'expression 7.65, l'erreur de quantification peut être considérée comme un bruit blanc non centré.

De même, nous déterminons la corrélation entre le signal x et l'erreur de quantification e à partir de l'analyse du moment d'ordre 2 du signal quantifié. L'expression de la corrélation entre x et e est la suivante :

$$E(xe_t) = j\frac{q}{2}\Phi'_x(0) + \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} q.\Phi'_x(-ku_0)\frac{1}{\pi k} \tag{7.66}$$

L'erreur de quantification e ne sera pas corrélée avec l'entrée x si la condition suivante est respectée :

$$\Phi'_x(k\tfrac{2\pi}{q}) = 0 \quad \forall k \tag{7.67}$$

7.4.4 Simulation

Pour illustrer les différents résultats présentés dans cette partie, le processus de quantification peut être simulé afin d'analyser les propriétés de l'erreur de quantification. Le synoptique du système utilisé est représenté à la figure 7.9. Le signal d'entrée x est un bruit blanc gaussien d'écart type σ et non quantifié. En sortie de l'opérateur de quantification nous obtenons le signal y. Les simulations ont été réalisées pour les lois de quantification par arrondi et par troncature (codage CA2) et le nombre de bits utilisés pour coder y, varie entre 1 et 24.

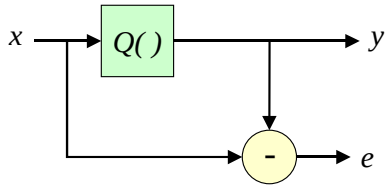


FIG. 7.9: Synoptique du système pour la simulation du processus de quantification

Si on compare la variance σ_{sim}^2 de l'erreur de quantification issue des simulations par rapport à l'expression théorique σ_{theo}^2, on obtient l'expression suivante :

$$C_\sigma = \frac{\sigma_{sim}^2}{\sigma_{theo}^2} \tag{7.68}$$

La figure 7.10 représente l'évolution de C_σ en fonction du rapport entre l'écart type σ_x du signal d'entrée et le pas de quantification q. Ces résultats sont quasiment identiques à ceux présents dans et montrent la validité de l'expression de la variance pour $\sigma_x > q$. Cette condition est toujours vérifiée en pratique.

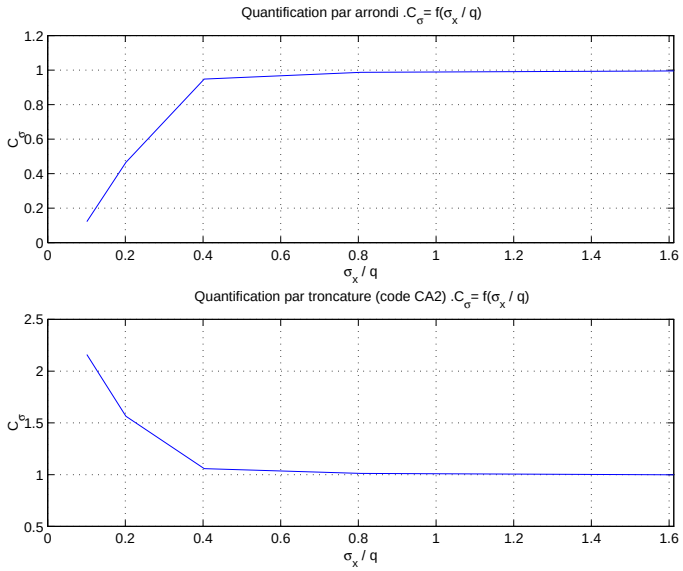


FIG. 7.10: Comparaison de la variance théorique et simulée de l'erreur de quantification

Nous avons mesuré le coefficient de corrélation entre l'entrée x et l'erreur de quantification e. Pour les deux lois de quantification, ce coefficient est inférieur à 0.02 si la condition $\sigma_x > q$ est vérifiée.

7.4.5 Résumé sur la modélisation du processus de quantification

La quantification est donc l'approximation des valeurs d'un signal $x(n)$ par un multiple entier du pas de quantification q. On notera :

$$x_Q(n) = Q[x(n)] = k.q \tag{7.69}$$

L'erreur de quantification est $e(n) = x_Q(n) - x(n)$. On modélisera cette quantification par l'addition d'une erreur $e(n)$ (voir figure 7.11), avec comme hypothèse que la suite $e(n)$ est une séquence d'un processus aléatoire stationnaire.

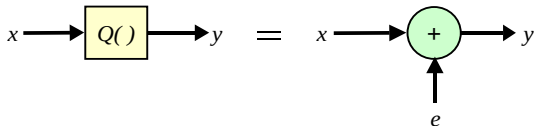


FIG. 7.11: Modélisation du bruit de quantification

On posera donc les hypothèses suivantes :
– $e(n)$ est stationnaire,

– le bruit issu de la quantification d'un signal $x(n)$ n'est pas corrélé avec ce signal,
– les bruits de quantification sont statistiquement indépendants,
– $e(n)$ est un bruit blanc uniformément réparti,
– $e(n)$ est borné par le pas de quantification,
– la distribution de probabilité de $e(n)$ est uniforme sur l'intervalle de quantification,
– l'ergodicité implique que les moyennes temporelles et statistiques sont équivalentes. En particulier on considérera que la **variance du bruit σ_e^2 est équivalente à la puissance du bruit**.

Selon la loi de quantification utilisée on aura :
– arrondi : $m_e = 0$, $\sigma_e^2 = \frac{q^2}{12}$,
– troncature : $m_e = \frac{q}{2}$, $\sigma_e^2 = \frac{q^2}{12}$.

7.5 Modélisation du bruit d'une conversion analogique

La conversion analogique numérique (CAN) consiste à passer d'un signal échantillonné $x(n)$ vers une suite de nombre $x_Q(n)$ dont les valeurs appartiennent à un intervalle de valeur (dynamique) et sont quantifiées, c'est à dire qu'elles sont un multiple entier du pas de quantification, que l'on notera q (voir figure 7.12). Dans ce cadre, on peut utiliser les modèles et lois de quantifications que nous avons vus dans les sections précédentes (en particulier voir résumé section 7.4.5).
On modélisera alors la conversion AN par l'ajout d'une erreur $e(n)$ de quantification. Dans le cas d'un algorithme de TNS, $e(n)$ représentera donc le bruit en entrée du système.

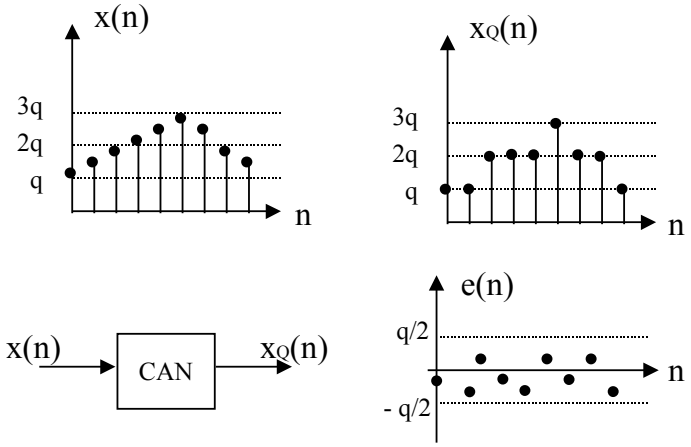


FIG. 7.12: Bruit de conversion analogique numérique

7.6 Filtrage d'un bruit de quantification

Si on considère un système composé d'un filtre numérique dans lequel on injecte un signal quantifié (voir figure 7.13). Le signal en entrée du filtre est donc composé du signal $x(n)$ de puissance σ_x^2 et d'un bruit de quantification $e(n)$ de puissance σ_e^2. On définit alors le rapport

signal à bruit en entrée du filtre par (quantification sur b bits en CA2) :

$$RSB = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{q^2/12} = 12 \times 2^{2(b-1)}\sigma_x^2 \tag{7.70}$$

$$RSB_{dB} = 10 \log RSB = 6.02 \times b + 4.77 \times 10 \log \sigma_x^2 \tag{7.71}$$

Le rapport signal à bruit augmente donc de $6dB$ par bit ajouté lors de la quantification [Bel87].

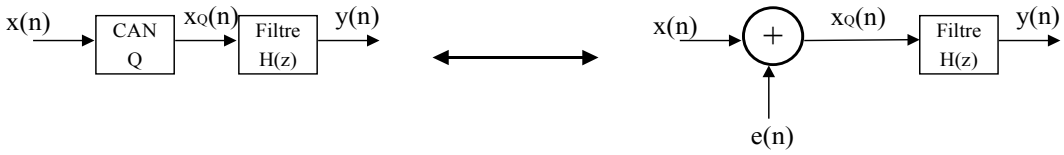


FIG. 7.13: Filtrage d'un signal et de son bruit de conversion

En sortie du filtre, si on considère que le filtre ne génère pas de bruit, le signal $y(n)$ sera composé du signal $x(n)$ et du bruit $e(n)$ filtré par $H(z)$. On aura alors :

$$y(n) = x(n) * h(n) + e(n) * h(n) = x(n) * h(n) + f(n) \tag{7.72}$$

où $f(n)$ est le bruit de quantification en sortie du filtre. On définit alors les moyennes m_f et puissance du bruit σ_f^2 du bruit $f(n)$:

$$m_f = m_e \sum_{n=-\infty}^{+\infty} h(n) = m_e H(e^{j0}) \tag{7.73}$$

$$\sigma_f^2 = \sigma_e^2 \sum_{n=-\infty}^{+\infty} |h(n)|^2 = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\Omega})|^2 d\Omega \tag{7.74}$$

Le rapport signal à bruit de sortie dépendra donc de la bande passante du filtre, mais également du comportement du filtre vis à vis du signal. La formule 7.74 permettra souvent d'étudier la puissance des bruits de calcul générés par les différents algorithmes de TNS.

7.7 Modélisation du bruit de calcul au niveau des opérateurs

La modélisation du bruit en sortie d'un opérateur nécessite de prendre en compte les deux sources de bruit suivantes :
- le *bruit propagé* par une opération en supposant que le format des données en sortie de l'opérateur est suffisant pour assurer l'absence de perte d'information.
- le *bruit généré* lors de la réduction du nombre de bits d'une donnée, liée à un changement de format.

Un opérateur va être caractérisé par la modélisation de ces deux bruits additifs comme le représente la figure 7.14.

7.7.1 Modélisation du bruit généré

Modélisation du bruit généré par la troncature

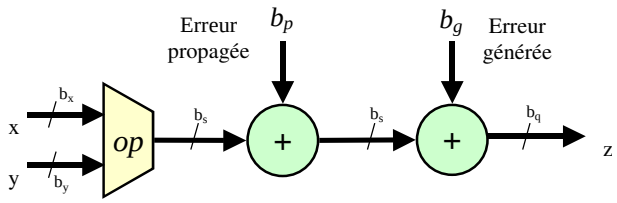


FIG. 7.14: Modélisation du bruit de calcul

Ce bruit correspond à la modélisation de la perte d'information résultant de l'élimination des k bits les moins significatifs lors d'un changement de format. Nous considérons une donnée de départ X_e et la donnée X_s après changement de format. La représentation de ces données codées en complément à 2, est présente à la figure 7.15 et les expressions de celles-ci sont les suivantes :

$$X_e = -2^m.S + \sum_{-n}^{m-1} b_i 2^i \tag{7.75}$$

$$X_s = -2^m.S + \sum_{-j}^{m-1} b_i 2^i \tag{7.76}$$

Nous définissons q le pas de quantification de la donnée après changement de format. Celui-ci est égal à 2^{-j}.

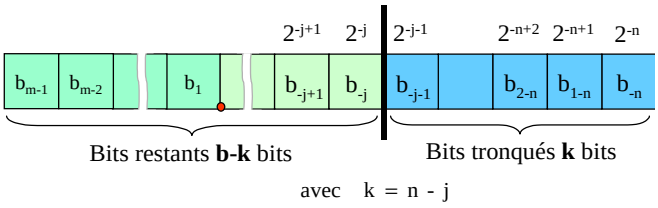


FIG. 7.15: Représentation des données lors d'une troncature

L'expression du bruit de quantification correspondant à la différence entre les deux variables X_e et X_s, est la suivante :

$$b_g = \sum_{i=j-1}^{n} b_{-i} 2^{-i} \tag{7.77}$$

L'expression 7.77 montre que le bruit généré sera toujours positif. En effet lors d'une troncature en complément à 2 la valeur tronquée est toujours inférieure à la valeur de départ.

Le bruit issu de la troncature ne pouvant prendre qu'un nombre fini de valeurs, égal à 2^k, nous pouvons modéliser le bruit généré par une variable aléatoire discrète.
Nous supposons que les valeurs binaires des bits b_{-j-1} à b_{-n} sont équiprobables, ainsi les valeurs représentant b_g sont équiprobables. Nous pouvons considérer que la densité de probabilité $p(x_q)$ représentée à la figure 7.16, est uniforme. Son expression est la suivante :

$$p(x_q) = \sum_{i=0}^{2^k-1} 2^{-k}\delta(x_q - i.2^{-n}) \tag{7.78}$$

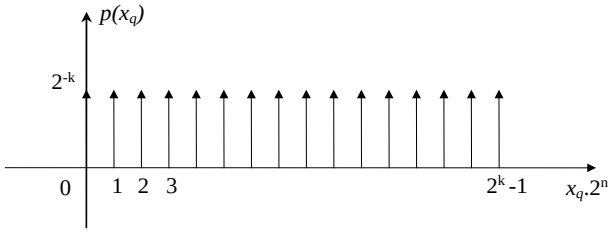


FIG. 7.16: Fonction de distribution du bruit généré lors d'une troncature

La moyenne de cette variable aléatoire est la suivante :

$$\mu_{b_g} = \sum_{i=-\infty}^{+\infty} y_i p(y_i) = \sum_{i=0}^{2^k-1} i.2^{-n}.2^{-k} \tag{7.79}$$

soit

$$\mu_{b_g} = 2^{-j-1}(1 - 2^{j-n}) = \frac{q}{2}(1 - 2^{-k}) \tag{7.80}$$

La variance de cette variable aléatoire est égale à :

$$\sigma^2_{b_g} = \sum_{i=-\infty}^{+\infty} y_i^2 p(y_i) - \mu^2_{b_g} \tag{7.81}$$

soit

$$\sigma^2_{b_g} = \frac{2^{-2j}}{12}(1 - 2^{2(j-n)}) = \frac{q^2}{12}(1 - 2^{-2k}) \tag{7.82}$$

Lorsque le nombre de bits tronqués est important les résultats tendent vers ceux obtenus dans le cas d'une loi de quantification par troncature :

$$\mu_{b_g} = 2^{-j-1} = \frac{q}{2} \tag{7.83}$$

$$\sigma^2_{b_g} = \frac{1}{12}(2^{-2j}) = \frac{q^2}{12} \tag{7.84}$$

Modélisation du bruit généré par l'arrondi

Dans le cas d'une quantification par arrondi conventionnel, la donnée correspondant à la valeur médiane de l'intervalle $[kq, (k+1)q]$ est codée à la valeur $(k+1)q$. La loi de quantification est définie de la façon suivante :

$$Q(x) = \begin{cases} k.q & \text{si } \; kq \leq x < (k+\frac{1}{2})q \\ (k+1).q & \text{si } \; (k+\frac{1}{2})q < x \leq (k+1)q \\ (k+1)q & \text{si } \; x = (k+\frac{1}{2})q \end{cases} \tag{7.85}$$

L'expression de la densité de probabilité de l'erreur de quantification b_{gac} représentée à la figure 7.17, est la suivante

$$p(x_q) = \sum_{i=-2^{k-1}}^{2^{k-1}-1} 2^{k-n}\delta(x_q - i.2^{-n}) \tag{7.86}$$

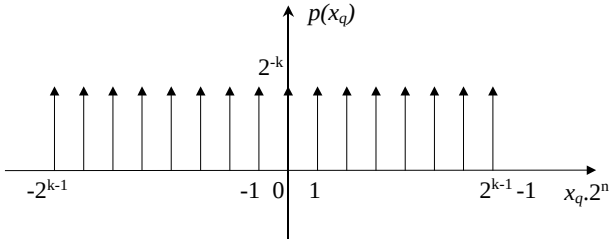


FIG. 7.17: Fonction de distribution du bruit généré lors d'un arrondi conventionnel

La moyenne de cette variable aléatoire est égale à :

$$\mu_{b_{gac}} = \sum_{i=-2^{k-1}}^{2^{k-1}-1} i.2^{-n}.2^{-k} = 2^{-n-1} = \frac{q}{2}(2^{-k}) \tag{7.87}$$

La valeur médiane de chaque intervalle $[k.q, (k+1).q]$ étant codée systématiquement à la valeur supérieure, la moyenne de l'erreur de quantification n'est pas nulle. Cette erreur est néanmoins nettement plus faible que celle présente lors d'une quantification par troncature. La variance de cette variable aléatoire est égale à :

$$\sigma^2_{b_{gac}} = \sum_{i=-2^{k-1}}^{2^{k-1}-1} (i.2^{-n})^2.2^{-k} - \mu^2_{b_g} = \frac{2^{-2j}}{12}(1 - 2^{-2k}) = \frac{q^2}{12}(1 - 2^{-2k}) \tag{7.88}$$

Afin d'éliminer le biais, un arrondi convergent peut être utilisé, celui va affecter la valeur médiane de façon équiprobable à la valeur supérieure et inférieure :

$$Q(x) = \begin{cases} k.q & \text{si } \; k.q \leq x < (k+\frac{1}{2})q \\ (k+1).q & \text{si } \; (k+\frac{1}{2})q < x \leq (k+1)q \\ k.q \text{ ou } (k+1).q & \text{si } \; x = (k+\frac{1}{2})q \end{cases} \tag{7.89}$$

$$\text{avec } \mathrm{P}\Big(Q(x) = kq \setminus x = (k+\frac{1}{2})q\Big) = \mathrm{P}\Big(Q(x) = (k+1)q \setminus x = (k+\frac{1}{2})q\Big)$$

Cette technique garantit l'équiprobabilité du codage de la valeur médiane en analysant la parité de la donnée. Les données impaires sont codées à la valeur supérieure et les données paires à la valeur inférieure. La densité de probabilité associée à l'erreur de quantification présentée à la figure 7.18, est égale à :

$$p(x_q) = \frac{2^{-k}}{2}\Big(\delta(x_q + 2^{k-n-1}) + \delta(x_q - 2^{k-n-1})\Big) + \sum_{i=-2^{k-1}+1}^{2^{k-1}-1} 2^{-k}\delta(x_q - i.2^{-n}) \tag{7.90}$$

La moyenne de cette variable aléatoire est la suivante :

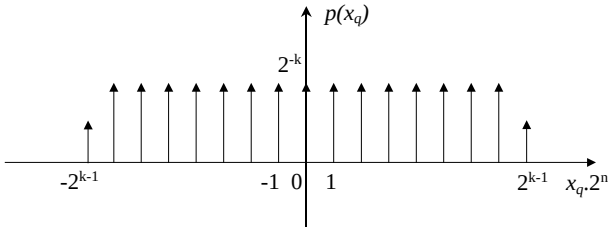


FIG. 7.18: Fonction de distribution du bruit généré lors d'un arrondi convergent

$$\mu_{b_g} = 2^{-k}\left(2^{k-n-1} - 2^{k-n-1}\right) + \sum_{i=-2^{k-1}+1}^{2^{k-1}-1} i.2^{-n}.2^{-k} = 0 \tag{7.91}$$

Ce type d'arrondi permet d'obtenir une moyenne de l'erreur de quantification nulle. L'expression de la variance de l'erreur de quantification est la suivante :

$$\sigma^2_{b_g} = 2^{-k}2^{2(k-n-1)} + \sum_{i=-2^{k-1}+1}^{2^{k-1}-1} (i.2^{-n})^2.2^{-k} \tag{7.92}$$

soit

$$\sigma^2_{b_g} = \frac{2^{-2j}}{12}(1 - 2^{-2k+1}) = \frac{q^2}{12}(1 - 2^{-2k+1}) \tag{7.93}$$

7.7.2 Simulation du bruit généré

Pour valider le modèle proposé, nous avons simulé le phénomène de changement de format et comparé les caractéristiques du bruit généré, issues de la modélisation et celles obtenues par les simulations. Le synoptique du système utilisé est représenté à la figure 7.19. Le signal d'entrée x est un bruit blanc gaussien d'écart type σ et quantifié avec b bits. En sortie de l'opérateur de changement de format nous obtenons le signal y codé sur n bits après l'élimination de k bits. Le bruit généré e_g est égal à la différence entre x et y. La figure 7.20 représente la fonction de distribution du bruit généré, obtenue pour k=4.

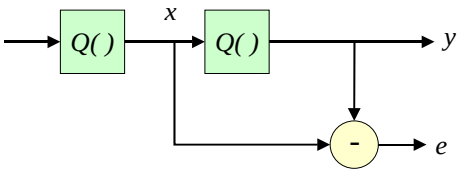


FIG. 7.19: Synoptique du système pour la simulation du processus de changement de format

Nous avons comparé la moyenne et la variance du bruit généré, obtenus par simulation avec les expressions théoriques 7.80 7.82, 7.87, 7.88. Les figures 7.21 et 7.22 représentent l'évolution de ces paramètres pour différentes valeurs de k. Les moyennes ont été normalisées par q et $q/2$ et les variances par $q^2/12$. Ces résultats montrent l'adéquation du modèle proposé pour le calcul des moments d'ordre 1 et 2 du bruit généré.

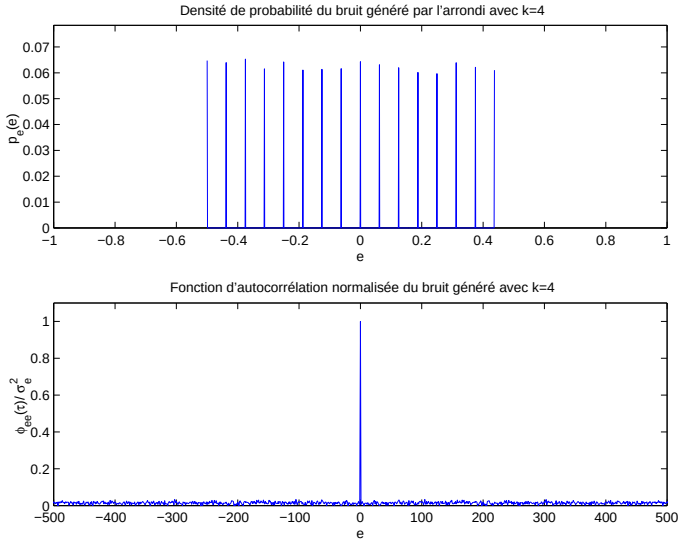


FIG. 7.20: Fonction de distribution et d'autocorrélation obtenues pour k=4

L'étude de la fonction d'autocorrélation $\varphi_{ee}(\tau)$ du bruit généré permet de déterminer les caractéristiques spectrales de ce bruit. Un exemple de $\varphi_{ee}(\tau)$ est proposé à la figure 7.20. En utilisant la moyenne et la variance de e_g, nous pouvons identifier la fonction d'autocorrélation obtenue à l'expression suivante :

$$\varphi_{ee}(\tau) = (\mu_e^2 + \sigma_e^2)\delta(\tau) \tag{7.94}$$

Ainsi le bruit généré peut être considéré comme un bruit blanc non centré.

7.7.3 Modélisation du bruit propagé

Nous considérons un opérateur dont les deux opérandes d'entrée x et y sont affectées respectivement d'un bruit b_x et b_y. Nous posons les hypothèses suivantes :
– les opérandes d'entrée de l'opérateur x et y sont indépendantes ;
– les bruits b_x et b_y associés aux opérandes x et y sont indépendants ;
– le bruit associé à chaque opérande n'est pas corrélé avec l'opérande.

La sortie est composée de deux termes z et b_z, le premier représente la sortie de l'opérateur en l'absence de bruit en entrée et le second regroupe l'ensemble des termes liés à la présence des bruits b_x et b_y. Les notations utilisées pour les paramètres statistiques (moyenne, variance, puissance) d'une variable aléatoire b_w sont présentées ci-dessous :

$$\begin{aligned} \mu_{b_w} &= E(b_w) \\ \sigma^2_{b_w} &= E\Big((b_w - \mu_{b_w})^2\Big) \\ P_{b_w} &= E(b_w^2) \end{aligned} \tag{7.95}$$

7.7.3.1 Addition

L'expression du bruit b_z en sortie de l'additionneur est la suivante :

$$b_z = b_x + b_y \tag{7.96}$$

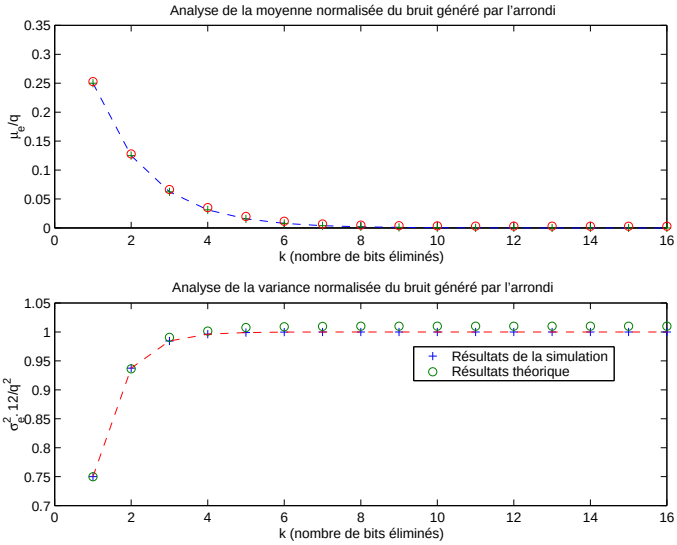


FIG. 7.21: Évolution de la moyenne et de la variance du bruit généré par l'arrondi

La moyenne et la variance du bruit b_z en sortie de l'opérateur sont égales à :

$$\begin{aligned} \mu_{b_z} &= \mu_{b_x} + \mu_{b_y} \\ \sigma^2_{b_z} &= \sigma^2_{b_x} + \sigma^2_{b_y} \end{aligned} \tag{7.97}$$

Le RSB en sortie de l'opérateur ρ_{Add} représente le rapport entre la puissance du signal utile P_z et celle du bruit P_{b_z} :

$$\rho_{Add} = \frac{E(z^2)}{E(b_z^2)} = \frac{P_x + P_y}{P_{b_x} + P_{b_y}} \tag{7.98}$$

7.7.3.2 Multiplication

L'expression du bruit en sortie du multiplieur est la suivante :

$$b_z = y.b_x + x.b_y + b_x.b_y \tag{7.99}$$

Étant donné que la puissance des bruits b_x et b_y est nettement inférieure à celle de x et y, nous pouvons négliger dans l'expression 7.99 le terme $b_x.b_y$. Ainsi les caractéristiques statistiques du bruit en sortie du multiplieur sont les suivantes :

$$\begin{aligned} \mu_{b_z} &= \mu_y \mu_{b_x} + \mu_x \mu_{b_y} \\ \sigma^2_{b_z} &= \sigma^2_y P_{b_x} + \sigma^2_x P_{b_y} + \sigma^2_{b_x} \mu^2_y + \sigma^2_{b_y} \mu^2_x \end{aligned} \tag{7.100}$$

L'expression du RSB en sortie du multiplieur est égale à :

$$\rho_{Mult} = \frac{\sigma^2_x \sigma^2_y + \mu^2_x \sigma^2_y + \mu^2_y \sigma^2_x}{\sigma^2_y P_{b_x} + \sigma^2_x P_{b_y} + \sigma^2_{b_x} \mu^2_y + \sigma^2_{b_y} \mu^2_x} \tag{7.101}$$

Multiplication d'une variable et d'une constante

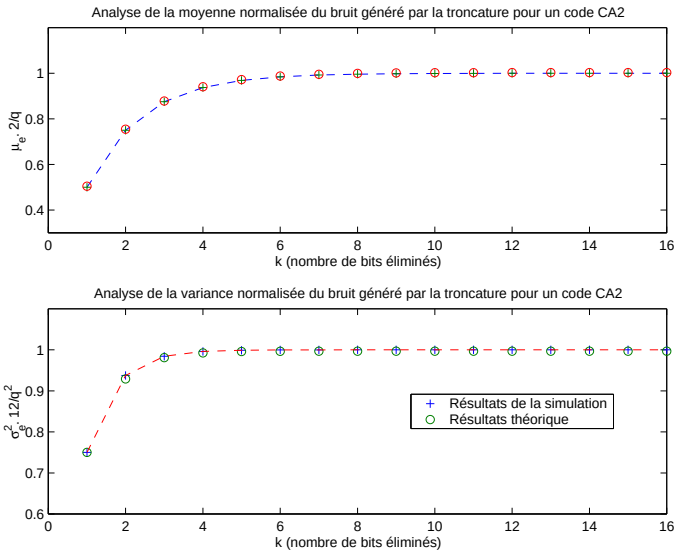


FIG. 7.22: Évolution de la moyenne et de la variance du bruit généré par troncature

Dans les algorithmes de TNS, de nombreuses opérations correspondent à la multiplication d'un signal x par une constante Y. Nous considérons que la constante Y est affectée d'une erreur de quantification Δ_Y fixe. Les expressions de la moyenne et de la variance du bruit b_z sont les suivantes :

$$\begin{aligned} \mu_{b_z} &= Y\mu_{b_x} + \mu_x\Delta_Y \\ \sigma^2_{b_z} &= Y^2\sigma^2_{b_x} + \sigma^2_x\Delta^2_Y \end{aligned} \tag{7.102}$$

Nous obtenons une expression du RSB en sortie du multiplieur $\rho_{Mult\ cste}$ fonction des RSB des deux entrées :

$$\rho^{-1}_{Mult\ cste} = \rho^{-1}_x + \rho^{-1}_y \qquad \text{avec} \quad \rho_y = \frac{Y^2}{\Delta^2_Y} \tag{7.103}$$

De l'équation 7.103 nous pouvons en déduire que le RSB en sortie du multiplieur sera toujours inférieur à celui de l'entrée x.

7.8 Comparaison des codages en virgule fixe et en virgule flottante

Dans cette partie nous comparons la dynamique et le rapport signal à bruit de quantification des données codées en virgule fixe et en virgule flottante.

7.8.1 Analyse de la dynamique

Le niveau de dynamique exprimé en dB du codage en virgule fixe est linéaire par rapport au nombre de bits b utilisés par le codage :

$$D_{N\ dB} \simeq 20.b.\log(2) \tag{7.104}$$

Le niveau de dynamique pour une représentation en virgule flottante est fonction du nombre de bits E alloués pour l'exposant :

$$D_{N\ dB} = 20\log(2^{2K+1}) \quad \text{avec} \quad K = 2^{E-1} - 1 \tag{7.105}$$

Nous avons représenté à la figure 7.23 les expressions 7.104 et 7.105 en fonction du nombre de bits utilisés. Nous avons fixé pour le codage en virgule flottante la taille de l'exposant à 1/4 de la longueur totale.

Lorsque le nombre de bits est inférieur à 16, le niveau de dynamique obtenu avec une représentation en virgule fixe est supérieur à celui d'une représentation en virgule flottante. Cette tendance s'inverse pour un nombre de bits supérieur à 16. Pour N=32, la représentation en virgule flottante montre tout son intérêt, la dynamique disponible permet d'utiliser ce codage dans la majorité des applications sans risque de débordements.

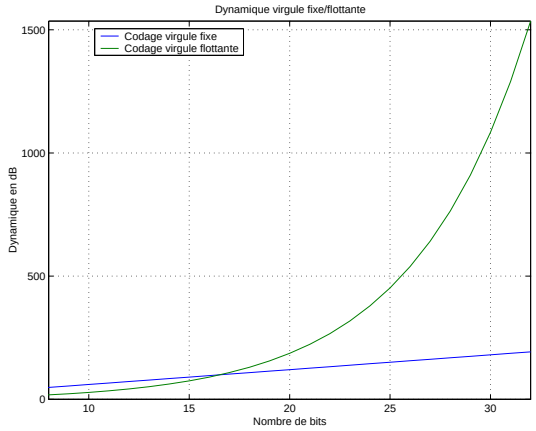


FIG. 7.23: Dynamique du codage en virgule fixe et en virgule flottante

7.8.2 Analyse du RSB

La puissance P_e du bruit de quantification correspond au moment d'ordre 2 de l'erreur de quantification e :

$$P_e = \mu_e^2 + \sigma_e^2 \tag{7.106}$$

Nous définissons K_x le rapport entre la racine carrée de la puissance du signal $x(k)$ et sa dynamique D. La puissance P_x du signal $x(k)$ est égale à :

$$P_x = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} x(k)^2 = (K_x D)^2 \tag{7.107}$$

L'expression du RSB exprimée en dB dans le cas d'un codage en virgule fixe est la suivante :

$$\rho_{dB} = 10\log(\frac{P_s}{P_e}) = 20\log(D) + 20\log(K_x) + 10\log(\mu_e^2 + \sigma_e^2) \tag{7.108}$$

L'expression 7.108 montre que le RSB est linéaire par rapport à la dynamique du signal d'entrée.

Pour un codage en virgule flottante et une quantification par arrondi, l'expression du RSB est la suivante :

$$\rho_{dB} = 20\log(K_x) + 20\log(D) - 10\log(2^E) - 10\log\left(\frac{2^{-2M}}{12}\right) \tag{7.109}$$

E représente l'exposant associé à la valeur à coder. Il est défini tel que :

$$\frac{1}{2} \leq \frac{D}{2^E} < 1 \tag{7.110}$$

Pour illustrer cette analyse, la figure 7.24 représente deux exemples de l'évolution du RSB en fonction de la dynamique du signal d'entrée. Ces exemples montrent bien que le RSB est quasiment constant dans le cas de la virgule flottante. L'utilisation d'un exposant explicite dans le codage permet de s'adapter à la dynamique du signal et de maintenir un RSB constant et indépendant de la dynamique du signal. Pour les signaux de dynamique faible, très sensibles à l'erreur de quantification, la représentation en virgule flottante permet d'obtenir un meilleur RSB.

Lorsque le nombre de bit est identique, le RSB du codage en virgule fixe est supérieur à celui en virgule flottante pour des signaux dont la dynamique d'entrée est élevée. Ceci correspond au cas ou le pas de quantification de la représentation en virgule flottante devient supérieur à celui en virgule fixe.

Pour le codage en virgule flottante, le choix du nombre de bits alloués à la mantisse et à l'exposant est un compromis entre une dynamique élevée et un RSB élevé.

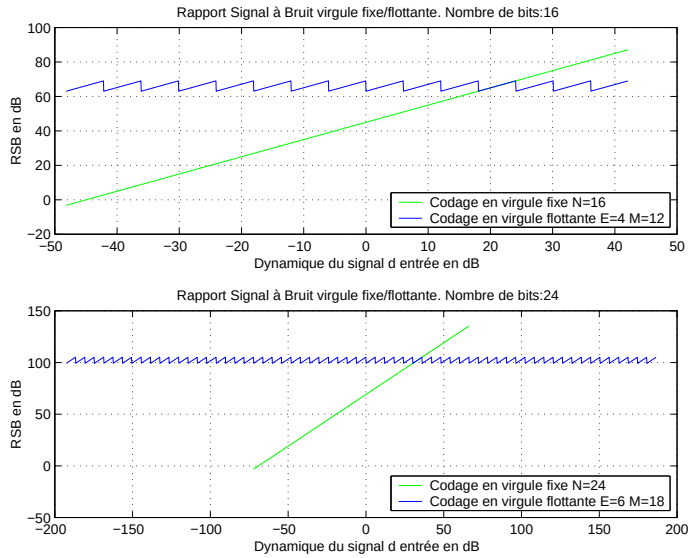


FIG. 7.24: RSB pour un codage en virgule fixe et en virgule flottante en fonction de la dynamique du signal d'entrée

7.9 Effets de la quantification sur des applications de traitement du signal

7.9.1 Filtrage RIF

Soit un filtre RIF $H(z) = \sum_{i=0}^{N-1} b_i x_{n-i}$. La figure 7.25 montre le graphe flot de calcul de ce filtre, ainsi que les différents bruits de quantification que l'on peut trouver dans cet algorithme.

- b_{gx} représente le bruit en entrée du filtre associé au signal $x(n)$.
- b_{gb_i} représente le bruit généré par les multiplications. Dans le cas d'une architecture à double précision[3], ce bruit sera nul.
- b_{gadd} représente le bruit généré par le changement de format en sortie du filtre. Dans le cas d'une architecture à simple précision, ce bruit sera nul.

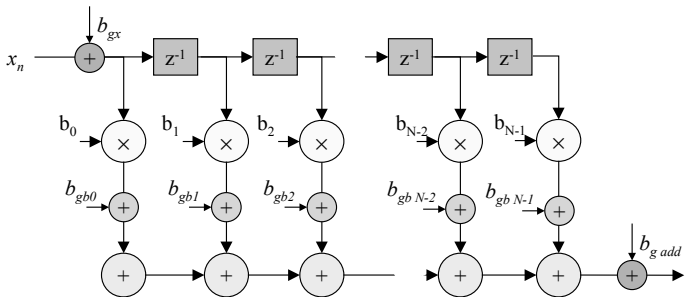


FIG. 7.25: Bruits de quantification dans un filtre RIF

7.9.1.1 Bruit en sortie du filtre RIF pour une architecture simple précision

Dans le cas d'une architecture simple précision et si on néglige le bruit en entrée, on peut écrire le bruit en sortie du filtre et sa puissance comme valant :

$$f(n) = \sum_{i=0}^{N-1} b_{gb_i}(n) \tag{7.111}$$

$$\sigma_f^2 = N\frac{q^2}{12} \tag{7.112}$$

Si on considère le bruit en entrée, la puissance devient :

$$\sigma_f^2 = \sigma_e^2 \sum_{n=0}^{N-1} |h(n)|^2 + N\frac{q^2}{12} \tag{7.113}$$

car le bruit en sortie est composé de la somme du bruit d'entrée filtré par $H(z)$ et du bruit généré par les multiplications.

[3]Une architecture simple précision est définie par un chemin de données où toutes les entrées-sorties des opérateurs sont sur b bits. Une architecture double précision possède un multiplieur $b \times b$ *bits* $\rightarrow 2b$ *bits* et un additioneur/accumulateur sur $2b$ bits. Voir cours sur les processeur de traitement du signal.

7.9.1.2 Bruit en sortie du filtre RIF pour une architecture double précision

Dans le cas d'une architecture double précision et si on néglige le bruit en entrée, on peut écrire le bruit en sortie du filtre et sa puissance comme valant :

$$f(n) = b_{gbadd}(n) \tag{7.114}$$

$$\sigma_f^2 = \frac{q^2}{12} \tag{7.115}$$

7.9.1.3 Problème de débordement du filtre RIF

Il est aisé de calculer dans le cas du filtre RIF la valeur maximum de la sortie $y(n)$:

$$|y(n)| \leq x_{max} \sum_{n=0}^{N-1} |h(n)| \tag{7.116}$$

où x_{max} est l'amplitude maximum du signal d'entrée $x(n)$. Afin de garantir qu'il n'y ait pas de débordement dans le filtre H, on effectue sur l'entrée une mise à l'échelle par un gain $A < 1$. Pour un codage en virgule fixe cadrée à gauche par exemple, garantir que $y(n)$ reste dans le domaine de codage signifie $y(n) < 1$. On peut donc en déduire le gain A par :

$$A < \frac{1}{x_{max} \sum_{n=0}^{N-1} |h(n)|}$$

ou par :

$$A < \frac{1}{x_{max} MAX\left(H(e^{j\Omega})\right)}$$

7.9.2 Filtrage RII du premier ordre

L'exemple sera traité en cours.

7.9.3 Filtrage RII du second ordre

L'exemple sera traité en cours.

7.9.4 Filtrage RII en cascade

L'exemple sera traité en TD.

Chapitre 8

Synthèse des filtres RII

8.1 Introduction et rappels en filtrage analogique

8.1.1 Introduction

La synthèse d'un filtre numérique est la recherche d'une fonction $H(z)$ (ou $h(n)$) correspondant à la spécification sous forme de gabarit. La recherche de cette fonction peut être réalisée selon diverses méthodes.

- La méthode la plus courante est l'utilisation des méthodes de synthèse des filtres analogiques aboutissant à une fonction $H(p)$ correspondant aux spécifications. Une fonction permettant le passage du plan p dans le plan z (i.e. $p = f(z)$) est ensuite utilisée pour obtenir $H(z)$. Cette fonction doit maintenir la stabilité du filtre analogique et maintenir, au mieux, les caractéristiques de la réponse fréquentielle $H(e^{j\Omega})$ du filtre numérique. Nous étudions dans la suite du chapitre trois types de transformation f.
- Une synthèse directe en z est également possible selon les mêmes principes que la synthèse analogique, mais elle ne sera pas présentée ici vu le nombre important d'outils développés dans le contexte analogique.
- Enfin, des méthodes d'optimisation issues de l'analyse numérique peuvent être utilisées afin de rechercher une fonction $H(z)$ s'approchant le plus possible d'une fonction prototype. La minimisation d'un critère d'erreur entre la courbe idéale et la courbe réelle est alors appliquée.

La figure 8.1 résume la méthodologie de synthèse de filtres analogiques et numériques RII. Les premières phases de synthèse analogique (normalisation, approximation et dénormalisation) sont explicités dans la section suivantes, tandis que les différentes transformations $p \rightarrow z$ sont présentées aux sections 8.2 à 8.4.

8.1.2 Rappels en filtrage analogique

Les filtre analogiques sont spécifiés de manière équivalente aux filtres numériques (voir section 6.3). Les fréquences sont exprimées en fonction de f (en Hz) ou de ω, la pulsation (en rad/s). La fonction de transfert $H(p)$ est exprimée en $p = j\omega$. La synthèse de cette fonction $H(p)$ est composée de trois étapes principales :

1. normalisation du gabarit,

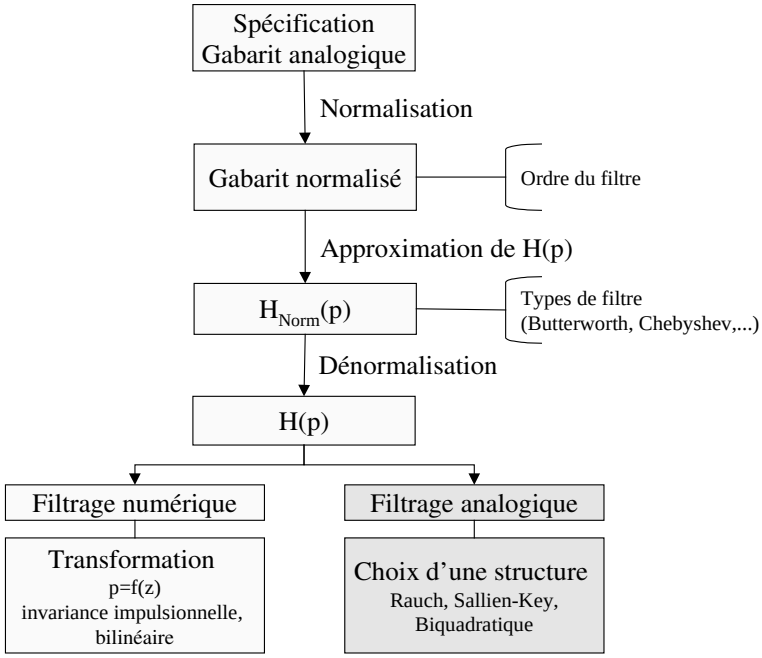


FIG. 8.1: Méthodologie de synthèse de filtres analogique et numérique RII

2. approximation de la fonction de transfert normalisée $H_{Norm}(p)$,
3. dénormalisation de $H_{Norm}(p)$ aboutissant à la fonction de transfert $H(p)$.

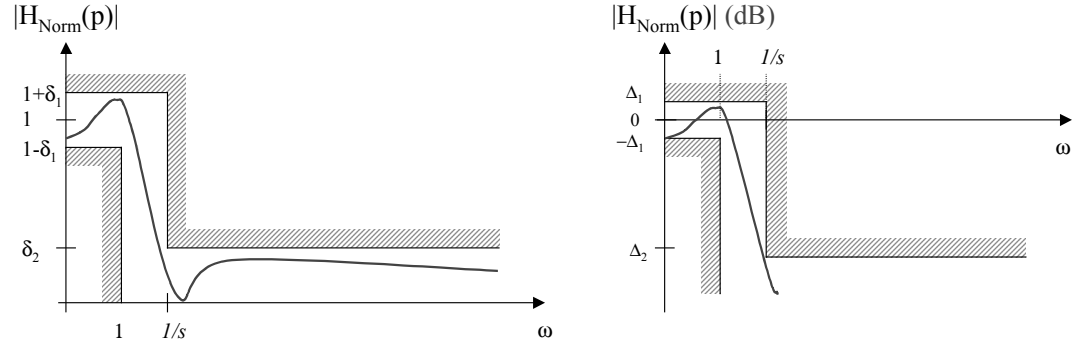
8.1.2.1 Normalisation

Cette première phase permet d'aboutir au gabarit passe-bas prototype (ou gabarit normalisé) à partir de n'importe que des 4 types principaux de filtres (passe-bas, passe-haut, passe-bande, réjecteur-de-bande). Celui ci (voir figure 8.2) est un gabarit passe-bas possédant une pulsation de coupure normalisée à 1 et une pulsation de bande atténuée valant $\frac{1}{s}$, s étant la sélectivité définie au paragraphe 6.3 (voir tableau 6.1). Les valeurs d'ondulation et d'atténuation sont inchangées.

8.1.2.2 Approximation de la fonction de transfert

La phase suivante consiste en la recherche d'une fonction $H_{Norm}(p)$ entrant dans le gabarit prototype passe-bas défini avant. Il existe plusieurs fonctions d'approximation, nous exposons ci-après les 4 fonctions les plus utilisées.
Pour la détermination de l'ordre du filtre et des coefficients de la fonction de transfert $H(p)$, on pourra soit utiliser les formules et tableaux données ci-après, soit utiliser les abaques fournies en annexe B.

1. **Filtres de Butterworth**
   Un filtre passe-bas de Butterworth d'ordre N et de pulsation de coupure ω_c est défini



a) Gabarit prototype linéaire b) Gabarit prototype en dB

FIG. 8.2: Gabarit prototype passe-bas

par le carré de sa réponse fréquentielle :

$$\left|H^B(\omega)\right|^2 = \frac{1}{1+\left(\frac{\omega}{\omega_c}\right)^{2N}} \tag{8.1}$$

Les principales propriétés des filtres de Butterworth sont :
– l'amplitude est une fonction monolithique décroissante sur ω,
– la réponse fréquentiel est plate dans les bandes passante et atténuées,
– le gain maximum est pour $\omega = 0$ et vaut 1,
– $|H^B(\omega_c)| = \sqrt{0.5}$, ω_c est donc la pulsation de coupure à -3dB,
– l'atténuation asymptotique dans les hautes fréquences vaut $20.N$ dB/décade.
L'ordre du filtre est déterminé par la relation :

$$N \geq \frac{log\left(\frac{1}{\delta_2\sqrt{\delta 1}}\right)}{log\left(\frac{1}{s}\right)} \tag{8.2}$$

N	a_1	a_2	a_3	a_4	a_5
2	1.4142				
3	2	2			
4	2.6131	3.4142	2.6131		
5	3.2361	5.2361	5.2361	3.2361	
6	3.8637	7.4641	9.1416	7.4641	3.8637

TAB. 8.1: Coefficients du polynôme du dénominateur d'un filtre passe-bas normalisé de Butterworth de degrés 2 à 6

2. **Filtres de Chebyshev**
Un filtre passe-bas de Chebyshev de type I d'ordre N et de pulsation de coupure ω_c est défini par le carré de sa réponse fréquentielle :

$$\left|H^C(\omega)\right|^2 = \frac{1}{1+\epsilon^2 T_N^2\left(\frac{\omega}{\omega_c}\right)}, \tag{8.3}$$

où $T_N(x)$ est un polynôme de Chebyshev de degré N, ϵ un paramètre permettant de régler l'amplitude de l'ondulation en bande passante ou en bande atténuée.

Les principales propriétés des filtres de Chebyshev sont :

– pour un filtre de type I, l'amplitude dans la bande passante possède une ondulation respectant :

$$0 \leq \omega \leq \omega_c \Longrightarrow \frac{1}{1+\epsilon^2} \leq \left|H^C(\omega)\right|^2 \leq 1$$

pour un filtre de type II, ces ondulations sont situées dans la bande atténuée ($\omega \geq \omega_c$),

–

$$|H^C(0)| = \begin{cases} 1/(1+\epsilon^2), & N \textit{ impair} \\ 1, & N \textit{ pair} \end{cases}$$

– l'atténuation asymptotique dans les hautes fréquences vaut $20.N$ dB/décade, mais l'ordre du filtre sera plus faible qu'un Butterworth à spécifications équivalentes.

3. **Filtres elliptiques**
Les filtres elliptiques possèdent des ondulations à la fois en bande passante et en bande atténuée. Ils sont optimaux en terme de sélectivité, c'est à dire qu'ils donnent l'ordre minimal à spécification donnée. Un filtre passe-bas elliptique d'ordre N et de pulsation de coupure ω_c est défini par le carré de sa réponse fréquentielle :

$$\left|H^E(\omega)\right|^2 = \frac{1}{1+\epsilon^2 R_N^2\left(\frac{\omega}{\omega_c}\right)}, \tag{8.4}$$

où $R_N(x)$ est une fonction rationnelle de Chebyshev de degré N, ϵ un paramètre permettant de régler l'amplitude de l'ondulation en bande passante ou en bande atténuée. L'ondulation en bande passante, équivalente à un chebyshev-I, est comprise entre 1 et $1/(1+\epsilon^2)$.

4. **Filtres de Bessel**
Les filtres de Bessel sont intéressants pour leur phase proche de la linéarité. En contrepartie, ils sont ceux ayant la sélectivité la plus faible.

8.1.2.3 Dénormalisation

Cette dernière étape permet de passer de la fonction de transfert normalisée $H_{Norm}(p_N)$ à la fonction de transfert $H(p)$. Nous appelons ici p_N la variable de la fonction normalisée. La dénormalisation consiste donc en l'application d'une fonction f permettant de transformer cette variable p_N en la variable $p = j\omega$: $p_N = f(p)$. La fonction f dépend du type de filtre spécifié. Le tableau 8.2 résume les fonctions de dénormalisation pour les 4 types de filtres.

passe-bas	passe-haut	passe-bande	réjecteur-de-bande
$p_N = \frac{p}{\omega_c}$	$p_N = \frac{\omega_c}{p}$	$p_N = \frac{1}{B}\left(\frac{p}{\omega_0} + \frac{\omega_0}{p}\right)$	$p_N = \left[\frac{1}{B}\left(\frac{p}{\omega_0} + \frac{\omega_0}{p}\right)\right]^{-1}$

Tab. 8.2: Fonctions de dénormalisation

Exemple 8.1.1 : Synthèse d'un filtre passe-haut

Soit le filtre défini par le gabarit suivant :

f_p	f_a	δ_1	δ_2
3 kHz	1 kHz	-3 dB	-20 dB

1. Dessiner le gabarit du filtre. En déduire le gabarit du filtre prototype.
2. Pour les 4 types de filtre précédents, déterminer l'ordre du filtre et la fonction de transfert normalisée $H_{Norm}(p)$.
3. En déduire les fonctions de transfert $H(p)$ des 4 types de filtre.

8.2 Méthode de l'invariance impulsionnelle

La méthode de l'invariance impulsionnelle consiste à effectuer la synthèse d'un filtre numérique dont la réponse impulsionnelle $h(nT)$ est l'échantillonnage de la réponse impulsionnelle $h_a(t)$ du filtre analogique équivalent (figure 8.3). On a alors :

$$h(nT) \triangleq h_a(t)|_{t=nT} \tag{8.5}$$

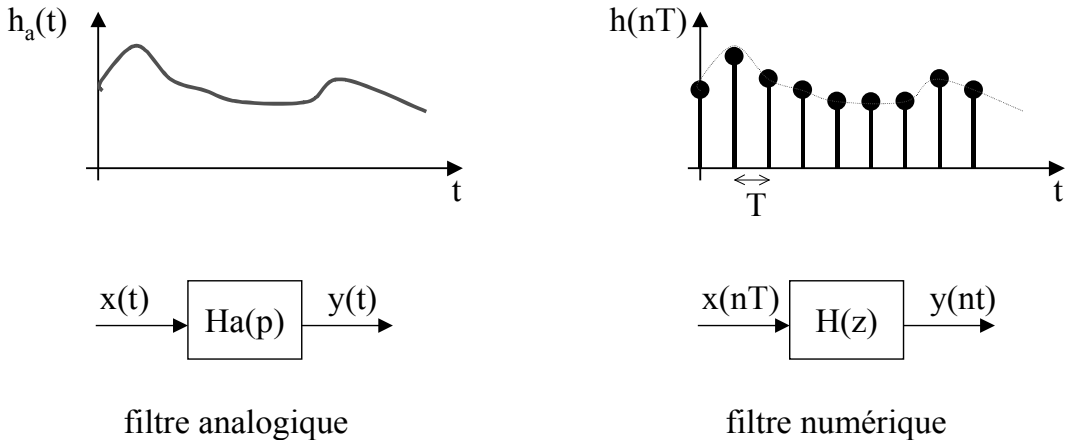


FIG. 8.3: Conservation de la réponse impulsionnelle entre les filtres analogique et numérique

Dans le cas où la fonction de transfert du filtre analogique $H_a(p)$ possède N pôles simples p_i, on peut écrire :

$$\begin{aligned} H_a(p) &= \frac{N(p)}{D(p)} = \sum_{i=1}^{N} \frac{k_i}{p + p_i} \\ h_a(t) &= \sum_{i=1}^{N} k_i e^{-p_i t} \end{aligned}$$

Si on échantillonne la réponse impulsionnelle du filtre analogique par $h(nT) = h_a(t)|_{t=nT}$, on obtient la fonction de transfert du filtre numérique $H(z)$:

$$h(nT) = \sum_{i=1}^{N} k_i e^{-p_i nT}$$

$$H(z) = \sum_{i=1}^{N} k_i \frac{z}{z - e^{-p_i T}} = \sum_{i=1}^{N} k_i \frac{1}{1 - e^{-p_i T} z^{-1}}$$

Si on étend cette formulation de $H(z)$ à tout type de fonction $H_a(p)$, on peut, en utilisant la formule des résidus, obtenir une relation directe entre $H_a(p) \longrightarrow H(z)$:

$$H(z) = \sum_{p_i \ de \ H_a(p)} Res\left(\frac{H_a(p)}{1 - e^{pT} z^{-1}}, p_i\right) \tag{8.6}$$

Cette méthode de synthèse est simple. Elle conserve la **réponse temporelle** du filtre analogique équivalent et également la stabilité du filtre. Par contre, la réponse fréquentielle du filtre n'est pas conservée. En effet, cette transformation suit logiquement le théorème d'échantillonnage et le phénomène de recouvrement de spectre. Son utilisation doit respecter les contraintes dues à l'échantillonnage, en particulier une bande passante inférieure à la moitié de la fréquence d'échantillonnage. La réponse fréquentielle est donnée par la formulation :

$$H(e^{j\omega T}) = H(z)|_{z=e^{j\omega T}} = \frac{1}{T} \sum_{k} H_a(j\omega + j\frac{2\pi k}{T}) \tag{8.7}$$

Par conséquent, la méthode de l'invariance impulsionnelle n'est applicable qu'aux filtres à bande limitée satisfaisant l'équation :

$$|H_a(j\omega)| \simeq 0 \ pour \ |\omega| > \omega_B \ avec \ \omega_B < \pi f_e \tag{8.8}$$

La figure 8.4 montre le résultat de la réponse fréquentielle d'un filtre numérique dans le cas où le filtre analogique équivalent n'est pas à bande limitée. Le phénomène de recouvrement de spectre est ici important. On remarque également que la réponse fréquentielle de $H(z)$ possède un facteur multiplicatif $1/T$ (équation 8.7). Ce facteur, dont la valeur est très élevée (e.g. 10000 pour une fréquence d'échantillonnage de 10kHz), doit être atténué si on veut un filtre réalisable. En pratique, on normalisera $H(z)$ par $(\times T)$ ou par $(/H(z = e^{j0})$. Dans ce dernier cas, on obtiendra une valeur en $\omega = 0$ équivalente au filtre analogique.

8.3 Transformation d'Euler

L'effet de recouvrement de la méthode de l'invariance impulsionnelle est due à l'utilisation d'une fonction non algébrique et non bijective entre le plan p et le plan z. Nous allons donc chercher à utiliser une transformation algébriques entre ces deux plans.
La transformation d'Euler consiste à utiliser l'approximation simple d'une dérivée :

$$\frac{de(t)}{dt}|_{t=nT} = \lim_{\Delta t \to 0} \frac{\Delta e(nT)}{\Delta t} \simeq \frac{e[nT] - e[(n-1)T]}{T} \tag{8.9}$$

Si on considère un système linéaire discret $H(z)$ réalisant cette approximation, il aura la forme :

$$s(nT) = \frac{e[nT] - e[(n-1)T]}{T} \tag{8.10}$$

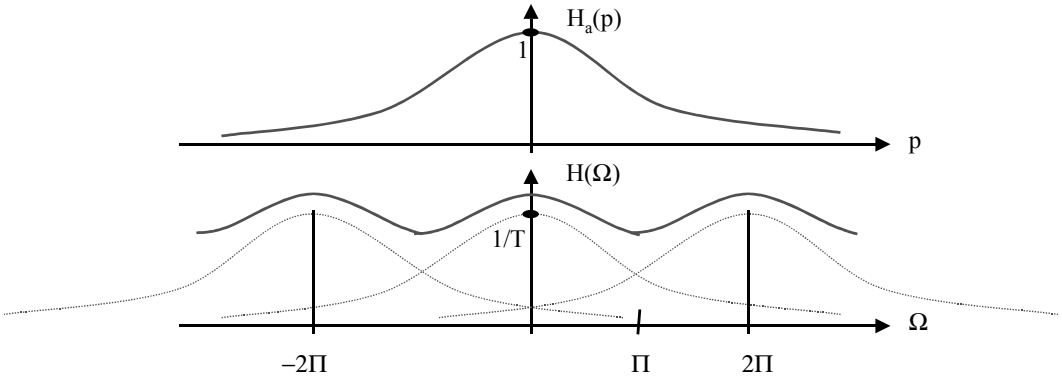


FIG. 8.4: Réponses fréquentielles des filtres analogique et numérique

Ce qui donne dans la plan z :

$$S(z) = \frac{1-z^{-1}}{T}E(z) \tag{8.11}$$

$$H(z) = \frac{1-z^{-1}}{T} \tag{8.12}$$

Dans le plan p, la dérivation est obtenue par la fonction de transfert $H_a(p) = p$. Par conséquent, la transformation d'Euler réalisant l'approximation d'une dérivée est définie par :

$$H(z) \triangleq H(p)|_{p=\frac{1-z^{-1}}{T}} \tag{8.13}$$

La transformation inverse donnée par :

$$z = \frac{1}{1-pT} \tag{8.14}$$

permet de chercher une relation entre les fréquences analogique (ω_a) et numérique (ω). On obtient :

$$z = \frac{1}{1-j\omega_a T} = e^{j\omega T} \tag{8.15}$$

$$z = \frac{1}{2}\left[1+\frac{1+j\omega_a T}{1-j\omega_a T}\right] = \frac{1}{2}\left[1+e^{j2\arctan(\omega_a T)}\right] \tag{8.16}$$

La variable z parcourt donc un cercle de centre $z = 1/2$ et de rayon $1/2$. Le cercle unité du plan z est donc transformé, impliquant une modification importante de la réponse fréquentielle du filtre donnée par l'équation 8.16. Cependant, il n'y a pas de phénomène de recouvrement de spectre. La stabilité est conservée.

La section suivante montre une transformation utilisant le même principe, mais dont les performances en terme de réponse fréquentielle sont meilleures.

8.4 Transformation bilinéaire

La transformation bilinéaire est issue d'un système linéaire discret $H(z)$ réalisant l'approximation d'une intégrale par la méthode des rectangles (voir figure 8.5. On peut l'exprimer comme un filtre récursif donné par :

$$s(nT) = s((n-1)T) + T\frac{e(nT) + e((n-1)T)}{2} \tag{8.17}$$

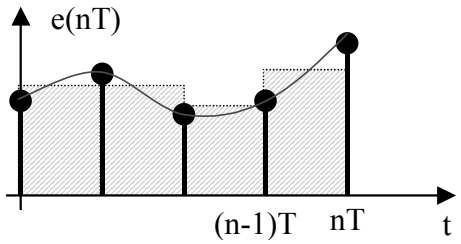


FIG. 8.5: Approximation d'une intégrale par la méthode des rectangles

Ce qui donne dans la plan z :

$$S(z) = z^{-1}S(z) + T\frac{E(z) + z^{-1}E(z)}{2} \tag{8.18}$$

$$H(z) = \frac{S(z)}{E(z)} = \frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}} \tag{8.19}$$

Dans le plan p, l'intégration est obtenue par la fonction de transfert $H_a(p) = 1/p$. Par conséquent, la transformation bilinéaire réalisant l'approximation d'une intégrale est définie par :

$$H(z) \triangleq H(p)|_{p=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}} \tag{8.20}$$

La transformation inverse donnée par :

$$z = \frac{2/T + p}{2/T - p} = \frac{2/T + j\omega_a}{2/T - j\omega_a} \tag{8.21}$$

permet de chercher une relation entre les fréquences analogique (ω_a) et numérique (ω). On obtient :

$$z = \frac{2 + j\omega_a T}{2 - j\omega_a T} = e^{j\omega T} \tag{8.22}$$

$$|z| = 1 \tag{8.23}$$

$$\arg(z) = \arctan(\frac{\omega_a T}{2}) + \arctan(\frac{\omega_a T}{2}) = 2\arctan(\frac{\omega_a T}{2})\arctan(\frac{\omega_a T}{2}) \tag{8.24}$$

$$z = e^{j\omega T} \Rightarrow \arg(z) = \omega T \tag{8.25}$$

Les équations 8.24 et 8.25 impliquent la relation entre les pulsations analogique et numérique :

$$\frac{\omega_a T}{2} = \tan\left(\frac{\omega T}{2}\right) \tag{8.26}$$

Cette relation est très importante puisqu'elle permet de connaître de manière analytique le transformation entre les réponses fréquentielles des filtres analogique et numérique. Cette relation s'appelle distorsion en fréquence ou *frequency warping*.

La figure 8.6 montre que l'ensemble de l'axe imaginaire du plan p est transformé vers le cercle unité du plan z (cf. équation 8.23 et 8.24) de manière bijective. De plus, le domaine de stabilité (demi-plan gauche du plan p) est transformé vers le disque unité (figure 8.6). Cette transformation évite donc le phénomène de recouvrement de spectre et conserve la stabilité. Cependant, une compression non linéaire de l'axe des fréquences est réalisée comme le montrent l'équation 8.26 et la figure 8.7.

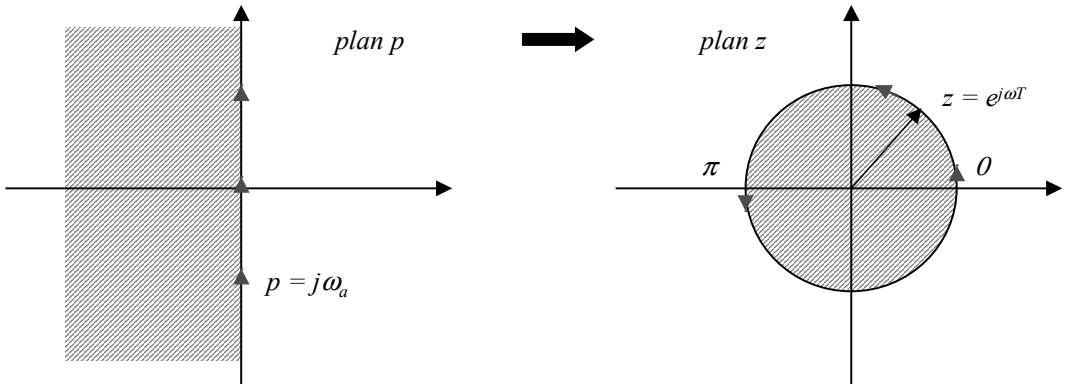


FIG. 8.6: Transformation du plan p vers le plan z

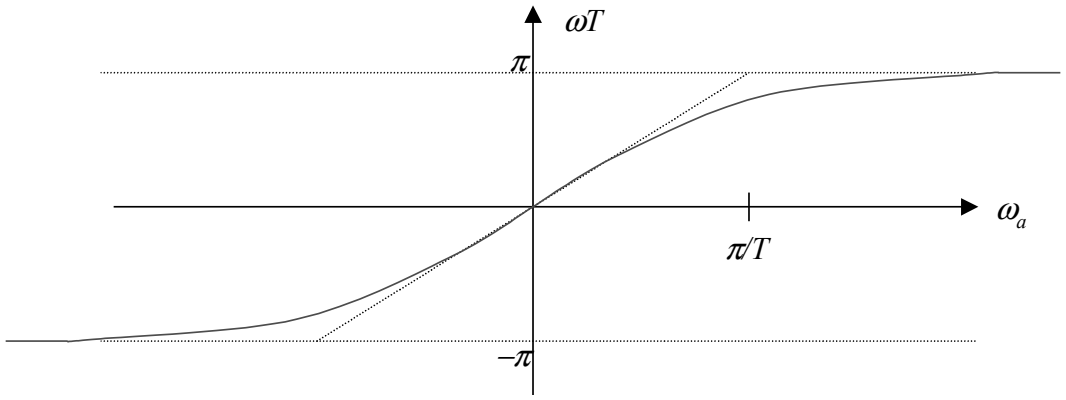


FIG. 8.7: Transformation non linéaire entre les fréquences analogique et numérique : distorsion en fréquence

En conséquence, la synthèse de filtre numérique utilisant la transformation bilinéaire est utilisable quand cette compression peut être tolérée ou compensée. La figure 8.8 illustre ce phénomène. Elle montre comment une réponse fréquentielle à temps continu et son gabarit est transformée en une réponse fréquentielle à temps discret. Si les fréquences critiques (i.e. le gabarit) sont modifiées en utilisant une **pré-distorsion** (équation 8.26), alors le filtre numérique respectera les spécifications initiales.

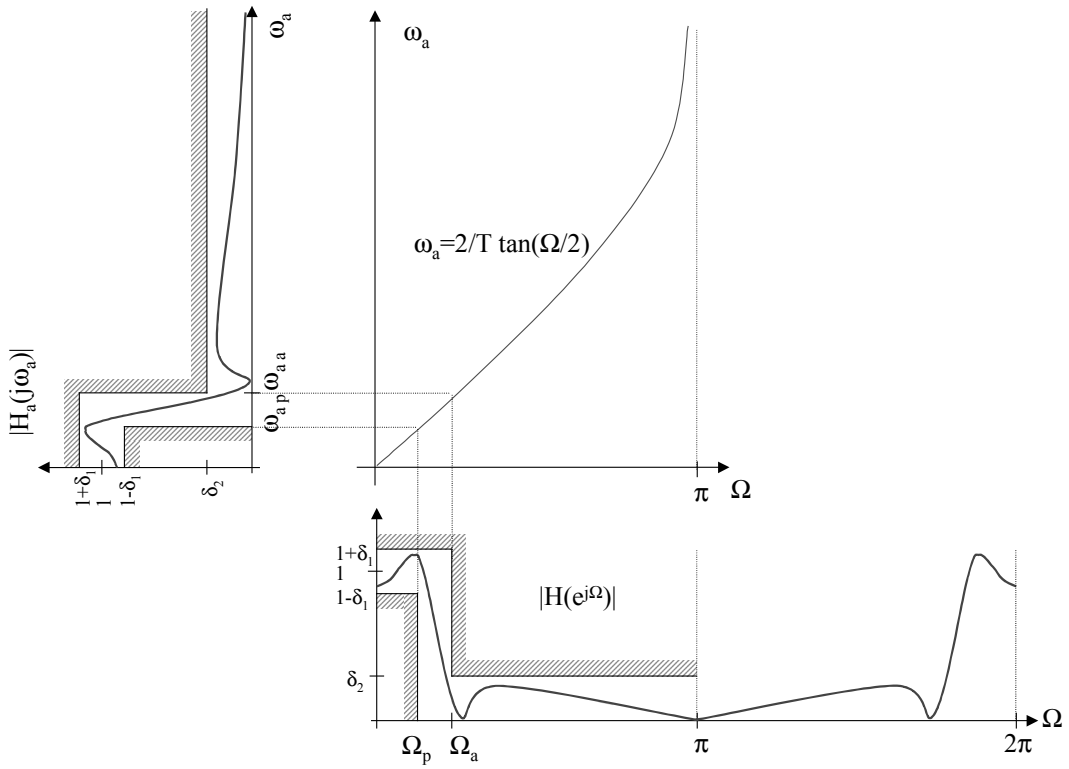


FIG. 8.8: Transformation fréquentielle du filtre analogique $H_a(\omega_a)$ vers le filtre numérique $H(\Omega)$

La méthodologie de synthèse par la transformation bilinéaire doit donc suivre l'algorithme suivant :

1. à partir des spécifications fréquentielles, effectuer une pré-distorsion des fréquences critiques du gabarit :
$$\omega_{a\ p,a} = \frac{2}{T}\tan\left(\frac{\Omega_{p,a}}{2}\right), \tag{8.27}$$
2. effectuer la synthèse du filtre analogique $H_a(p)$ (voir section 8.1),
3. utiliser la transformation bilinéaire $H(z) \triangleq H(p)|_{p=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$ pour obtenir le filtre numérique $H(z)$.

La pré-distorsion assure que le filtre numérique ainsi synthétisé entre dans le gabarit initial.

Exemple 8.4.1 : Synthèse d'un filtre IIR du premier ordre

On s'intéresse ici à la synthèse d'un filtre passe-bas RC du premier ordre par les méthodes de l'invariance impulsionnelle et bilinéaire. Soit le filtre :

$$H_a(p) = \frac{1}{1+\frac{p}{\omega_c}},$$

avec $f_c = 1kHz$ et $f_e = 10kHz$.

1. Donner la réponse fréquentielle du filtre analogique $H_a(\omega_a)$. Donner les valeurs de son module en $f = 0$, 1kHz, 5kHz et 10kHz.
2. Exprimez les filtres numériques obtenus par invariance impulsionnelle ($H_i(z)$) et transformation bilinéaire ($H_b(z)$). Pour $H_b(z)$, il faudra s'assurer de la même fréquence de coupure ω_c que le filtre analogique. Pour $H_i(z)$, il faudra s'assurer de la même valeur en $\Omega = 0$ que le filtre analogique.
3. Donner de manière analytique les réponses fréquentielles des deux filtres numériques. Donner les valeurs de leurs modules en $f = 0$, 1kHz, 5kHz et 10kHz.
4. Tracer ces deux réponses à l'aide de `Matlab` ou `Scilab`.

Chapitre 9

Synthèse des filtres RIF

9.1 Introduction

Un filtre RIF est défini par sa fonction de transfert en z :

$$H(z) = \sum_{n=0}^{N-1} h(n).z^{-n} \tag{9.1}$$

où la réponse impulsionnelle $h(n)$ est de longueur N.

A la différence des filtres RII, les filtres RIF ne sont réalisables que dans le domaine discret. Par conséquent, leurs méthodes de synthèse ne sont pas dérivées des filtres analogiques. On distingue trois principales méthodes, dont deux seront détaillées dans les sections suivantes.

1. **La méthode du fenêtrage (*windowing*)** consiste à appliquer une fenêtre de taille N au filtre idéal équivalent (voir section 9.3).
2. **La méthode de l'échantillonnage fréquentiel** utilise la Transformée de Fourier Discrète inverse depuis une fonction discrète représentative du filtre et définie en fréquence (voir section 9.4).
3. Les méthodes d'optimisation se concentrent sur la minimisation d'un critère d'erreur entre la courbe réelle et le filtre idéal. La plus utilisée est la méthode de Parks and McClellan [PM73], qui reformule le problème de synthèse de filtre sous la forme d'une approximation polynômiale.

9.2 Filtres à phase linéaire

Les filtres RIF ont la particularité de pouvoir présenter une phase linéaire. Puisque pour les systèmes causaux une phase nulle dans la bande passante n'est pas réalisable, une phase exactement linéaire évitera la modification de la forme du signal, c'est à dire le phénomène de distorsion qui peut être très gênant, en particulier dans les systèmes audio. En effet, une phase linéaire implique un temps de propagation de groupe constant, et, par conséquent, l'effet de la phase sur le signal sera un simple décalage temporel. On posera pour ce type de filtre :

$$H(e^{j\Omega}) = A(e^{j\Omega})e^{-j\alpha\Omega+j\beta}, \tag{9.2}$$

où α et β sont des constantes et $A(e^{j\Omega})$ une fonction réelle, éventuellement bipolaire, de Ω que l'on nommera plus tard pseudo-module[1] (ou *amplitude*). Dans ce cas, le temps de propagation de groupe $\tau g(\Omega) = -\frac{d\varphi(\Omega)}{d\Omega}$ sera constant et aura pour valeur α. $\varphi(\Omega)$ est ici la phase linéaire du filtre valant $\varphi(\Omega) = -\alpha\Omega + \beta$. On distinguera par la suite quatre types de filtres selon les valeurs de α et β et de la parité de N.

Le filtre $H(z)$ et sa réponse fréquentielle $H(e^{j\Omega})$ sont donnés par :

$$H(z) = \sum_{n=0}^{N-1} h(n).z^{-n} \tag{9.3}$$

$$H(e^{j\Omega}) = H(z)|_{z=e^{j\Omega}} = \sum_{n=0}^{N-1} h(n).e^{-jn\Omega} \tag{9.4}$$

$$H(e^{j\Omega}) = \Re[H(e^{j\Omega})] + j\Im[H(e^{j\Omega})] \tag{9.5}$$

$$H(e^{j\Omega}) = |H(e^{j\Omega})|e^{j\arg(\Omega)} = A(e^{j\Omega})e^{j\varphi(\Omega)} \tag{9.6}$$

•1^{er} **cas :** $\varphi(\Omega) = -\alpha\Omega$**,** $-\pi \leq \Omega \leq \pi$ Dans ce cas :

$$H(e^{j\Omega}) = A(e^{j\Omega})e^{-j\alpha\Omega} = A(e^{j\Omega})[\cos\alpha\Omega - j\sin\alpha\Omega] \tag{9.7}$$

$$H(e^{j\Omega}) = \sum_{n=0}^{N-1} h(n).e^{-jn\Omega} = \sum_{n=0}^{N-1} h(n)[\cos n\Omega - j\sin n\Omega] \tag{9.8}$$

En identifiant les parties réelles et imaginaires des deux équations précédentes, il est possible d'obtenir le système d'équation suivant :

$$\begin{cases} A(e^{j\Omega})\cos\alpha\Omega = \sum_{n=0}^{N-1} h(n)\cos n\Omega \\ A(e^{j\Omega})\sin\alpha\Omega = \sum_{n=0}^{N-1} h(n)\sin n\Omega \end{cases} \tag{9.9}$$

$$\Leftrightarrow \sum_{n=0}^{N-1} h(n)\cos n\Omega \sin\alpha\Omega - \sum_{n=0}^{N-1} h(n)\sin n\Omega\cos\alpha\Omega = 0 \tag{9.10}$$

$$\Leftrightarrow \sum_{n=0}^{N-1} h(n)\sin[(\alpha - n)\Omega] = 0 \tag{9.11}$$

L'équation 9.11 résume donc la condition sur la réponse impulsionnelle pour que le filtre soit à phase linéaire.

Si $\alpha \neq 0$, on montre que l'équation 9.11 est équivalente à :

$$\begin{cases} \alpha = \frac{N-1}{2} \\ h(n) = h(N-1-n) \ pour \ 0 \leq n \leq \alpha \end{cases} \tag{9.12}$$

La réponse impulsionnelle est donc **symétrique**. α est l'axe de symétrie de $h(n)$. α peut être entier ou non selon si N est impair ou pair.

[1] Le pseudo-module est souvent utilisé en remplacement du module car il évite les discontinuités dans le module dues à la valeur absolue et dans la phase dues à la détermination principale $[-\pi, \pi]$ de l'argument. On peut le voir comme une représentation du module intégrant ses variations de signe

•2^{nd} **cas :** $\varphi(\Omega) = \beta - \alpha\Omega$**,** $-\pi \leq \Omega \leq \pi$ Dans ce cas on obtient selon la même démonstration que pour le premier cas la relation suivante :

$$\sum_{n=0}^{N-1} h(n) \sin[(\alpha - n)\Omega + \beta] = 0 \tag{9.13}$$

On montre ensuite que l'équation 9.13 est équivalente à :

$$\begin{cases} \beta = \pm\pi/2 \\ \alpha = \frac{N-1}{2} \\ h(n) = -h(N-1-n) \textit{ pour } 0 \leq n \leq \alpha \end{cases} \tag{9.14}$$

La réponse impulsionnelle est donc **antisymétrique** par rapport à α.

On déduit de ces deux cas que selon la parité de N et le type de symétrie de $h(n)$, quatre types distincts de filtres sont à étudier.

9.2.1 Filtre RIF à phase linéaire de type I

Un filtre RIF à phase linéaire de type I est défini par une réponse impulsionnelle symétrique :

$$h(n) = h(N-1-n), \qquad 0 \leq n \leq \alpha \tag{9.15}$$

avec N impair. Le retard $\alpha = \frac{N-1}{2}$ est un entier. La réponse fréquentielle du filtre est :

$$H(e^{j\Omega}) = A(e^{j\Omega})e^{-j\alpha\Omega} = \sum_{n=0}^{N-1} h(n).e^{-jn\Omega} \tag{9.16}$$

En utilisant la relation de symétrie de l'équation 9.15 on obtient :

$$H(e^{j\Omega}) = e^{-j\frac{N-1}{2}\Omega} \sum_{n=0}^{\frac{N-1}{2}} a_n . \cos(n\Omega) \tag{9.17}$$

$$a_0 = h(\frac{N-1}{2}) \tag{9.18}$$

$$a_n = 2h(\frac{N-1}{2} - n), \qquad n = 1, \ldots \frac{N-1}{2} \tag{9.19}$$

9.2.2 Filtre RIF à phase linéaire de type II

Un filtre RIF à phase linéaire de type II est défini par une réponse impulsionnelle symétrique comme dans l'équation 9.15, avec N pair. Le retard $\alpha = \frac{N-1}{2}$ n'est pas un entier. La réponse fréquentielle du filtre est :

$$H(e^{j\Omega}) = e^{-j\frac{N-1}{2}\Omega} \sum_{n=1}^{\frac{N}{2}} b_n . \cos[(n - \frac{1}{2})\Omega] \tag{9.20}$$

$$b_n = 2h(\frac{N}{2} - n), \qquad n = 1, \ldots \frac{N}{2} \tag{9.21}$$

Remarque : si $\Omega = \pi$ alors $\cos[(n - \frac{1}{2})\pi] = 0$ ce qui implique que $H(e^{j\pi}) = 0$. Par consequent, un filtrage ne respectant pas cette contrainte (e.g. un filtre passe-haut) est impossible à réaliser avec un filtre de type II.

9.2.3 Filtre RIF à phase linéaire de type III

Un filtre RIF à phase linéaire de type III est défini par une réponse impulsionnelle antisymétrique :

$$h(n) = -h(N - 1 - n), \qquad 0 \leq n \leq \alpha \tag{9.22}$$

avec N impair. Le retard $\alpha = \frac{N-1}{2}$ est un entier, $\beta = \pm\pi/2$. La réponse fréquentielle du filtre est :

$$H(e^{j\Omega}) = je^{-j\frac{N-1}{2}\Omega} \sum_{n=1}^{\frac{N-1}{2}} c_n . \sin(n\Omega) \tag{9.23}$$

$$c_n = 2h(\frac{N-1}{2} - n), \qquad n = 1, \ldots \frac{N-1}{2} \tag{9.24}$$

Remarque : $H(e^{j0}) = H(e^{j\pi}) = 0$. Par consequent, un filtrage ne respectant pas cette contrainte est impossible à réaliser avec un filtre de type III. Les filtres pass-bande ou certains dérivateurs sont réalisables.

9.2.4 Filtre RIF à phase linéaire de type IV

Un filtre RIF à phase linéaire de type III est défini par une réponse impulsionnelle antisymétrique comme dans l'équation 9.22, avec N pair. Le retard $\alpha = \frac{N-1}{2}$ n'est pas un entier, $\beta = \pm\pi/2$. La réponse fréquentielle du filtre est :

$$H(e^{j\Omega}) = je^{-j\frac{N-1}{2}\Omega} \sum_{n=1}^{\frac{N}{2}} d_n . \sin[(n - \frac{1}{2})\Omega] \tag{9.25}$$

$$d_n = 2h(\frac{N}{2} - n), \qquad n = 1, \ldots \frac{N}{2} \tag{9.26}$$

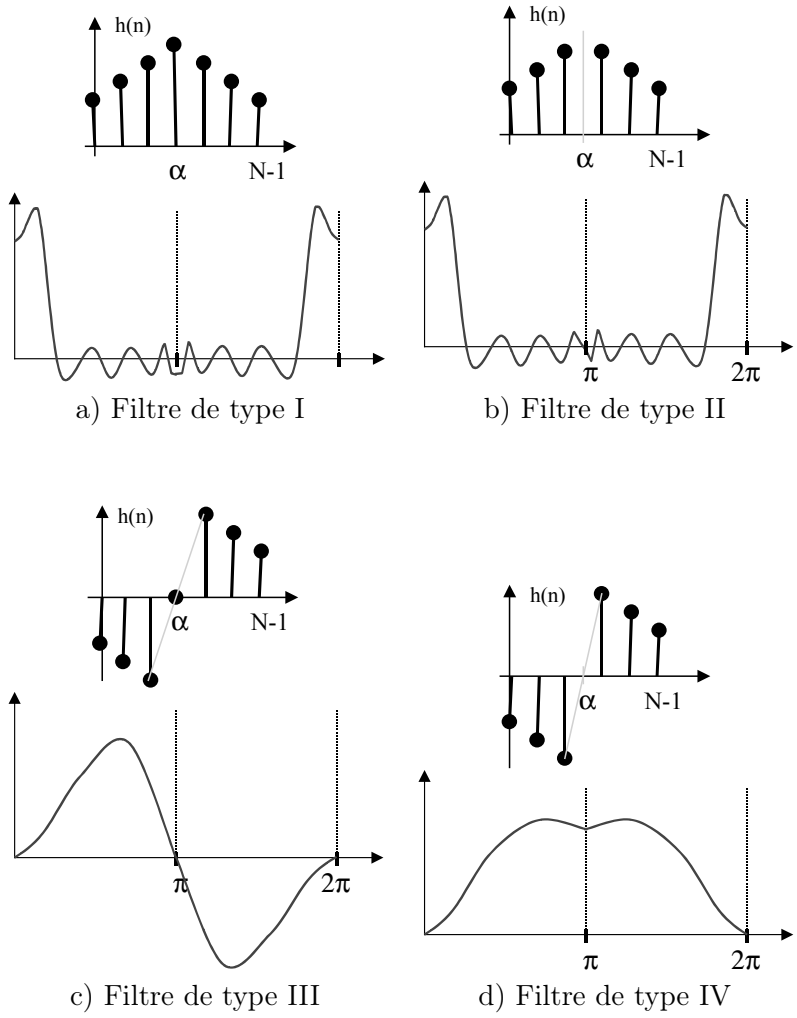
Remarque : $H(e^{j0}) = 0$, $H(e^{j\pi}) \neq 0$. Par consequent, un filtrage ne respectant pas cette contrainte est impossible à réaliser avec un filtre de type IV. Les filtres pass-haut ou certains dérivateurs sont réalisables.

La figure 9.1 résume les caractéristiques des 4 types de filtre RIF à phase linéaire. Elle trace des exemples de réponses impulsionnelle symétriques (types I et II) ou antisymétriques (types III et IV) et des exemples de réponses fréquentielles.

Exemple 9.2.1 : Analyse d'un filtre numérique RIF

Soit un filtre RIF défini par sa réponse impulsionnelle $h(n)$:

$$h(n) = 0.1\delta(n) - 0.3\delta(n-2) + 0.5\delta(n-3) - 0.3\delta(n-4) + 0.1\delta(n-6)$$



a) Filtre de type I b) Filtre de type II

c) Filtre de type III d) Filtre de type IV

FIG. 9.1: Les quatre types de filtres RIF à phase linéaire

1. Donner les expressions de l'équation aux différences finies ainsi que la fonction de transfert en Z.
2. Tracer la réponse impulsionnelle $h(n)$ du filtre.
3. Calculer la réponse fréquentielle $H(e^{j\Omega})$ du filtre. Déterminer son module et sa phase. On note que :
$$e^{-j\Omega_1} + e^{-j\Omega_2} = 2 \times e^{-j\frac{(\Omega_1+\Omega_2)}{2}} \times \cos(\frac{\Omega_2 - \Omega_1}{2})$$
4. Donner les valeurs du module en $\Omega = 0,\ \pi/2,\ ,\pi\ ,2\pi$.
5. Tracer approximativement son module. De quel type de filtre s'agit-il ?

9.3 Méthode de synthèse par fenêtrage

Soit un filtre numérique idéal $H(e^{j\Omega})$, périodique de période 2π. Il est décomposable en série de Fourier par sa réponse impulsionnelle :

$$H(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-jn\Omega} \tag{9.27}$$

Sa réponse impulsionnelle peut être exprimée en fonction de $H(e^{j\Omega})$ par :

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\Omega})e^{jn\Omega}d\Omega \tag{9.28}$$

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

$$\hat{h}(n) = \begin{cases} h(n), & 0 \leq n < N \\ 0, & n < 0 \ et \ n \geq N \end{cases}$$

On obtient dans ce cas un filtre numérique RIF approché $\hat{h}(n)$ dont la réponse fréquentielle est modifié par le phénomène de Gibbs [OS99], c'est à dire l'apparition d'ondulation dans les bandes passantes et atténuées d'amplitude convergent vers une valeur non nulle lorsque $N \to \infty$. De manière plus générale, on peut représenter $\hat{h}(n)$ comme le produit du filtre idéal $h(n)$ avec une fenêtre à durée finie $w(n)$.

$$\hat{h}(n) = h(n).w(n) \tag{9.29}$$

Dans l'exemple précédent, $w(n)$ est la fenêtre rectangulaire $r(n)$ définie par :

$$r(n) = \begin{cases} 1, & 0 \leq n < N \\ 0, & n < 0 \ et \ n \geq N \end{cases}$$

On peut alors étudier la réponse fréquentielle du filtre RIF ainsi synthétisé $\hat{H}(e^{j\Omega})$:

$$\hat{H}(e^{j\Omega}) = \hat{H}(z)|_{z=e^{j\Omega}} = \sum_{n=0}^{N-1} h(n).w(n).e^{-jn\Omega} \tag{9.30}$$

$$\hat{H}(e^{j\Omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})W(e^{j(\Omega-\theta)})d\theta \tag{9.31}$$

$$= H(e^{j\Omega}) * W(e^{j\Omega}) \tag{9.32}$$

La réponse fréquentielle du filtre $\hat{H}(e^{j\Omega})$ sera donc le résultat de la convolution entre le filtre idéal et la transformée de Fourier de la fenêtre. Les caractéristiques de cette dernière sont donc très importantes. On devra trouver un fenêtrage dont le $W(e^{j\Omega})$ se rapproche le plus des caractéristiques de $\delta(\Omega)$.

La figure 9.2 montre ce phénomène de convolution. Les effets sur le filtre idéal sont : - l'apparition d'ondulations en bandes passante et atténuée, - une zone de transition moins rapide déterminée par la largeur du lobe principal.

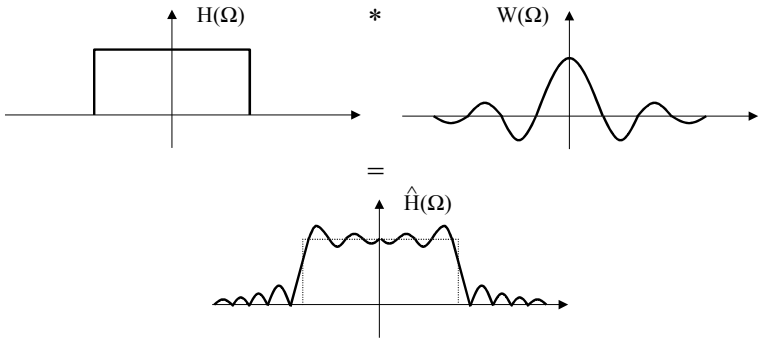


FIG. 9.2: Processus de convolution de la réponse fréquentielle du filtre idéal par la fenêtre

---

Exemple 9.3.1 : Calcul de la TF de la fenêtre rectangulaire

Soit la fenêtre rectangulaire $r(n)$ définie par : $r(n) = \begin{cases} 1, & 0 \leq n < N \\ 0, & n < 0 \ et \ n \geq N \end{cases}$

1. Calculer la transformée de Fourier $R(e^{j\Omega})$ de $r(n)$.
2. Tracer sa réponse fréquentielle entre $-\pi$ et π. Donner en particulier la valeur en 0 (amplitude du lobe principal), la largeur du lobe principal et l'amplitude du lobe secondaire.

Exemple 9.3.2 : Filtrage passe bas RIF à phase linéaire

Soit le filtre passe-bas idéal défini par :

$$H(e^{j\Omega}) = \begin{cases} e^{-j\alpha\Omega} & pour \quad -\Omega_c \leq \Omega \leq \Omega_c \\ 0 & pour \quad \Omega_c < \Omega \leq \pi \ et \ -\pi \leq \Omega < -\Omega_c \end{cases}$$

1. Tracer la réponse fréquentielle de $H(e^{j\Omega})$ en module et phase entre -2π et 2π.
2. Calculer et tracer $h(n)$.
3. Expliquer comment obtenir un filtre RIF de longueur N à partir de ce filtre. On utilisera la fenêtre étudiée dans l'exemple précédent. Donner et tracer les valeurs de $\hat{h}(n)$.
4. Donner une formule analytique de la réponse fréquentielle du filtre $\hat{H}(e^{j\Omega})$.
5. Tracer son module et sa phase.

9.3.1 Caractéristiques des principales fenêtres

Les principales fenêtres utilisées sont détaillées à la figure 9.3. Elles sont définies par les équation ci dessous :

Rectangulaire

$$w(n) = \begin{cases} 1, & 0 \leq n < N \\ 0, & n < 0 \; et \; n \geq N \end{cases} \tag{9.33}$$

Triangulaire (Bartlett)

$$w(n) = \begin{cases} \frac{2n}{N-1}, & 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N-1}{2} \leq n \leq N-1 \\ 0, & n < 0 \; et \; n \geq N \end{cases} \tag{9.34}$$

Hanning

$$w(n) = \begin{cases} 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & n < 0 \; et \; n \geq N \end{cases} \tag{9.35}$$

Hamming

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & n < 0 \; et \; n \geq N \end{cases} \tag{9.36}$$

Blackman

$$w(n) = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & n < 0 \; et \; n \geq N \end{cases} \tag{9.37}$$

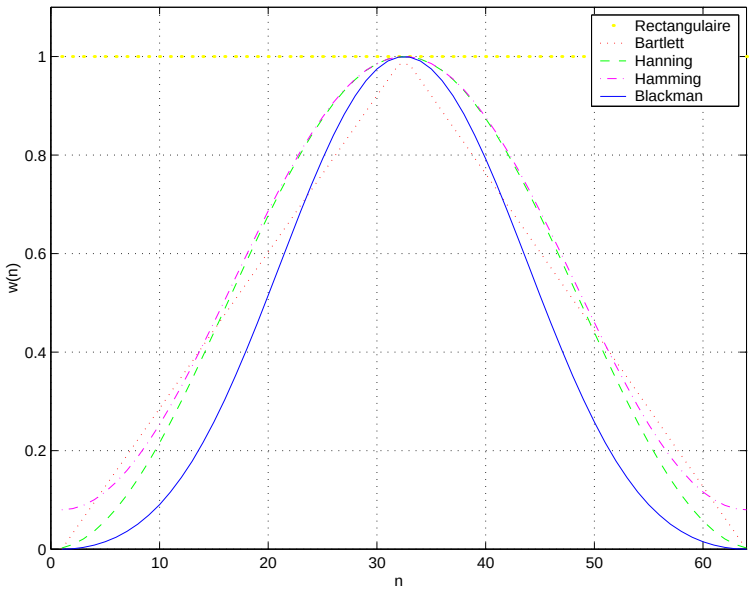


FIG. 9.3: Réponses temporelles des principales fenêtres

Ces fenêtres sont les principales fonctions utilisées en analyse spectrale et en synthèse de filtre RIF. Leurs réponses fréquentielles $20\log_{10}|W(e^{j\Omega})|$sont données à la la figure 9.4 avec $N = 64$.

On voit que selon la fenêtre, la largeur du lobe principal et l'amplitude du plus grand lobe secondaire diffèrent. Par exemple, la fenêtre rectangulaire possède le lobe principal le plus étroit ($4\pi/N$), mais le lobe secondaire le moins atténué ($-13dB$). Le tableau 9.1 résume ces caractéristiques. On y voit en particulier qu'en utilisant une fenêtre sans discontinuité et plus lisse (e.g. Hamming ou Blackman), on peut réduire de manière importante l'amplitude des lobes secondaires parasites. La colonne de droite est en relation avec la synthèse RIF et sera détaillée dans la section suivante.

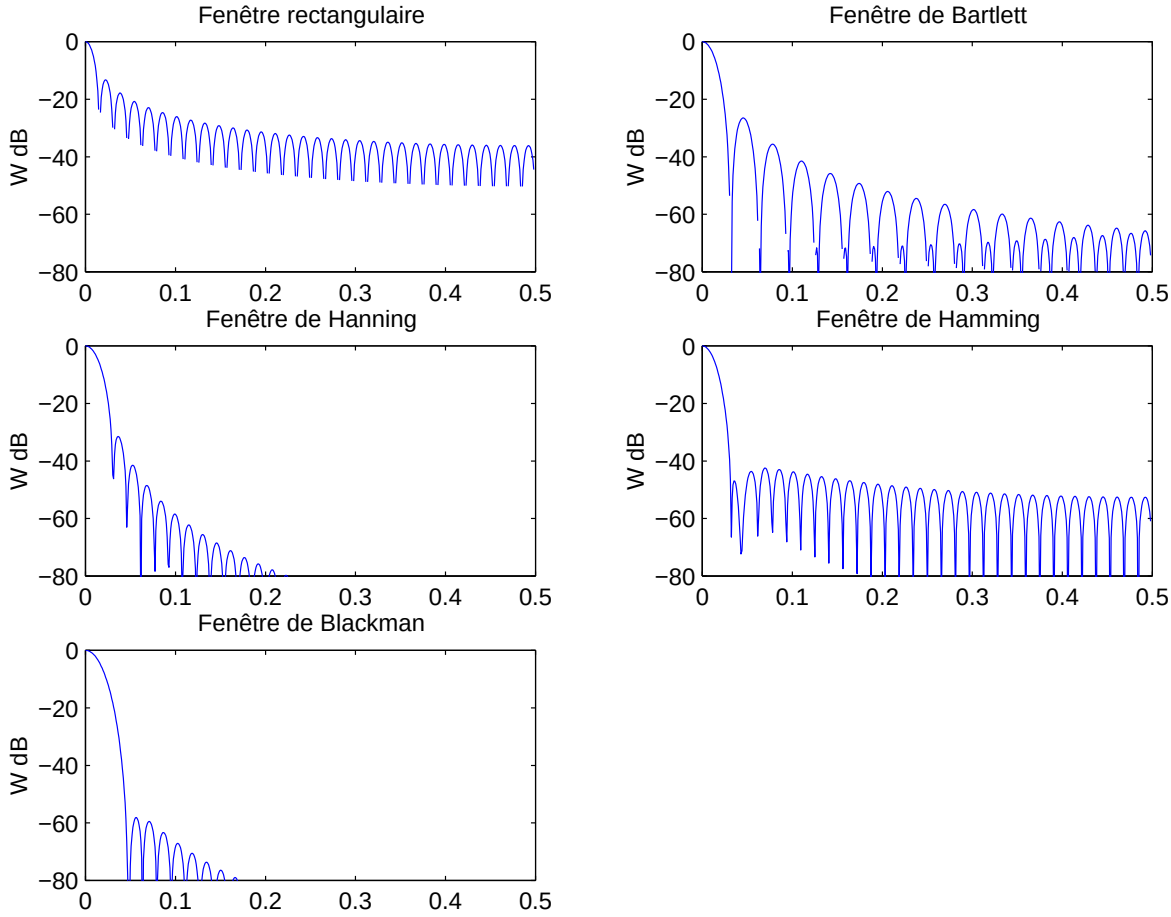


FIG. 9.4: Réponses fréquentielles des principales fenêtres

9.3.2 Choix de la fenêtre dans la méthode de synthèse

On voit donc que le choix de la fenêtre aura une influence sur les performances du filtre. En particulier, les caractéristiques importantes et leur influence sur le filtre synthétisé sont détaillées ci dessous.

– La largeur de la zone de transition $\Delta\Omega = |\Omega_p - \Omega_a|$ définie à la section 6.3.2 sera directement fonction de la largeur du lobe principal $\Delta\Omega_m$ donnée dans le tableau 9.1. Lorsque la fréquence de coupure ou de transition du filtre n'est pas trop proche de 0 ou de π, on peut considérer que $\Delta\Omega \cong \frac{\Delta\Omega_m}{2}$. On voit donc que **la transition $\Delta\Omega$ sera d'autant plus faible que** N

Type de fenêtre	Rapport d'amplitude entre lobe principal et lobe secondaire λ	Largeur du lobe principal $\Delta\Omega_m$	Atténuation minimale en bande atténuée ΔA
Rectangulaire	$-13dB$	$4\pi/N$	-21dB
Bartlett	$-25dB$	$8\pi/N$	-25dB
Hanning	$-31dB$	$8\pi/N$	-44dB
Hamming	$-41dB$	$8\pi/N$	-53dB
Blackman	$-57dB$	$12\pi/N$	-74dB

TAB. 9.1: Caractéristiques des principales fenêtres

sera grand.

– Les ondulations en bande passante et en bande atténuée seront égales, on notera $\Delta A = \delta_1 = \delta_2$ et on parlera de filtre *equiripple*. Cette atténuation est donnée par la quatrième colonne du tableau 9.1. On voit que ΔA **ne dépend que du type de fenêtre et non de** N.

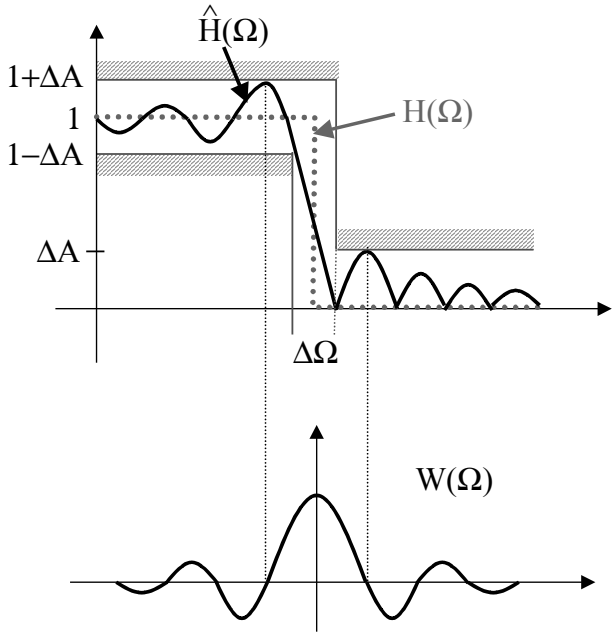


FIG. 9.5: Illustration de l'effet du fenêtrage sur le filtre idéal

La procédure de synthèse d'un filtre numérique RIF par fenêtrage à partir de la réponse impulsionnelle d'un filtre idéal $h(n)$ consiste à :

1. choisir en fonction de l'atténuation ΔA le type de fenêtre à utiliser,
2. choisir en fonction de la largeur de la zone de transition $\Delta\Omega$ et du type de la fenêtre $w(n)$ la longueur de la réponse impulsionnelle N.

Le filtre RIF résultant $\hat{h}(n)$ est ensuite défini par $\hat{h}(n) = h(n).w(n)$. Il sera de longueur N.

Remarque : un décalage de α de la réponse impulsionnelle du filtre idéal $h(n)$ peut être nécessaire afin de rendre le filtre $\hat{h}(n)$ à phase linéaire. On aura alors : $\hat{h}(n) = h(n-\alpha).w(n)$.

Le calcul de la largeur de la zone de transition $\Delta\Omega$ est ici présenté de façon empirique. Dans la plupart des cas, cette approximation reste suffisante, mais il est quelques fois nécessaire d'itérer plusieurs fois l'algorithme de synthèse afin d'assurer que le filtre entre dans le gabarit spécifié. Aussi un dernier type de fenêtre dit de Kaiser peut être utilisée. Cette méthode, permettant d'obtenir une formulation exacte des paramètres $\Delta\Omega$ et ΔA, ne sera développée ici, mais on peut se rapporter à [OS99] pour l'utiliser.

Exemple 9.3.3 : Caractéristique d'un filtre
Soit un filtre définie par une zone de transition $\Delta\Omega = 0.1\pi$ et une atténuation $> 30dB$, déterminer la taille et le type de fenêtre à utiliser.

9.4 Méthode de synthèse par échantillonnage en fréquence

La méthode de synthèse par échantillonnage en fréquence est appliquée depuis la réponse fréquentielle d'un filtre continu idéal $H(e^{j\Omega})$. A partir de celle ci et d'une valeur de N fixée, on réalise un échantillonnage en fréquence de pas $\Omega_e = \frac{2\pi}{N}$ définie par :

$$\hat{H}(e^{jk\Omega_e}) \triangleq H(e^{j\Omega})|_{\Omega=\frac{2k\pi}{N}} \tag{9.38}$$

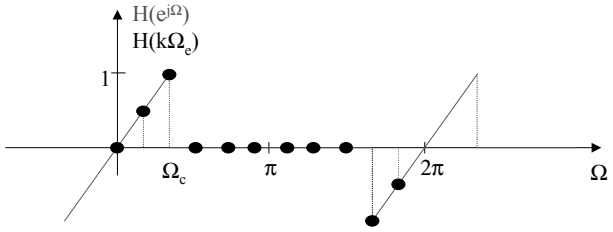


FIG. 9.6: Échantillonnage fréquentiel d'un filtre dérivateur

La figure 9.6 montre un exemple de dérivateur définie à partir d'un filtre continu. On remarquera en particulier que prendre N points entre $[0, 2\pi[$ à un pas de $\frac{2\pi}{N}$ élimine l'échantillon en $\Omega = 2\pi$. Le filtre RIF $\hat{h}(n)$ est alors trouvé par TFD inverse de $\hat{H}(e^{jk\Omega_e})$:

$$\hat{h}(n) = \frac{1}{N}\sum_{k=0}^{N-1} \hat{H}(e^{jk\Omega_e}) e^{2j\pi\frac{n.k}{N}},\ n = 0, 1 \ldots N-1 \tag{9.39}$$

$$\hat{h}(n) = 0,\ ailleurs \tag{9.40}$$

Cette méthode de synthèse est très simple et permet de réaliser toute forme de filtre. De plus, elle peut être combinée avec la méthode du fenêtrage. La fonction de transfert en Z du filtre

sera :

$$\hat{H}(z) = \sum_{n=0}^{N-1} \hat{h}(n) z^{-n} \tag{9.41}$$

En retravaillant cette fonction de transfert, on obtient :

$$\hat{H}(z) = \sum_{n=0}^{N-1} \hat{h}(n) z^{-n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} \hat{H}(e^{jk\Omega_e}) e^{2j\pi \frac{n.k}{N}} \right] z^{-n} \tag{9.42}$$

$$\hat{H}(z) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{H}(e^{jk\Omega_e}) \left[\sum_{n=0}^{N-1} e^{2j\pi \frac{n.k}{N}} z^{-n} \right] \tag{9.43}$$

$$\hat{H}(z) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{H}(e^{jk\Omega_e}) \frac{1 - e^{2j\pi \frac{N.k}{N}} z^{-N}}{1 - e^{2j\pi \frac{k}{N}} z^{-1}} \tag{9.44}$$

$$\hat{H}(z) = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \hat{H}(e^{jk\Omega_e}) \frac{1}{1 - e^{2j\pi \frac{k}{N}} z^{-1}} \tag{9.45}$$

La fonction de transfert du filtre $\hat{H}(z)$ peut donc être également exprimée comme une mise en parallèle de N filtres RII du premier ordre complexe. Cette décomposition permet d'obtenir directement les coefficients de cette structure du filtre à partir de la réponse fréquentielle recherchée.

Exemple 9.4.1 : Synthèse d'un filtre passe-bas

Soit un filtre passe bas idéal défini par :

$$H(e^{j\Omega}) = \begin{cases} 1 & pour \quad |\Omega| < \Omega_c = \frac{4\pi}{N} \\ 0 & pour \quad \frac{4\pi}{N} \leq |\Omega| < \pi \end{cases}$$

1. Tracer le filtre idéal entre $[0, 2\pi]$. Tracer ensuite le filtre échantillonné $\hat{H}(e^{jk\Omega_e})$.
2. Calculer $\hat{h}(n)$ par TFD inverse.
3. Calculer deux versions de $\hat{H}(z)$.

Chapitre 10

Analyse spectrale de signaux numériques

10.1 Introduction

Une des applications majeures utilisant la transformée de Fourier Discrète (TFD) en TNS est l'analyse du contenu fréquentiel (spectre) de signaux continus. Le principe de l'analyse de Fourier numérique d'un signal analogique déterministe est représenté à la figure 10.1. La numérisation du signal continu $x_c(t)$ est effectuée de manière classique par un filtre anti-repliement suivi d'un convertisseur analogique-numérique. Le signal discret $x(n)$ est ensuite multiplié par une fonction $w(n)$ nommée *fenêtre temporelle* sur N points. Ce fenêtrage est une conséquence de la contrainte de durée finie imposée par la TFD qui ne peut s'effectuer sur un nombre infini d'échantillons. Le signal résultant $x_N(n)$, dont on pourra ensuite analyser le spectre par TFD (ou par TFR) puis par un calcul du module et éventuellement de la phase, est donc représentatif d'une observation limitée du signal d'entrée sur un horizon d'observation $T_0 = N.T$. L'effet de cette troncature temporelle, indispensable au calcul numérique de l'analyse spectrale, va être étudié dans les sections suivantes.

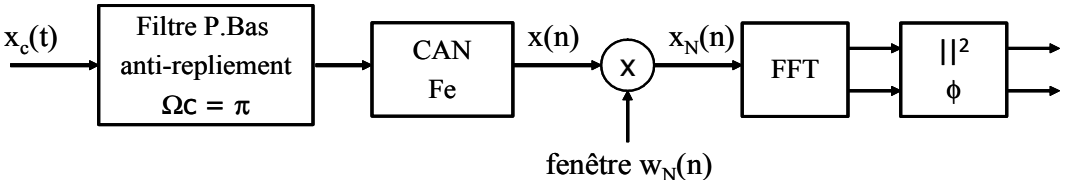


FIG. 10.1: Principe de l'analyse de Fourier numérique d'un signal analogique déterministe

10.2 Troncature d'un signal discrétisé

10.2.1 Opération dans le domaine temporel

Soit N le nombre d'échantillons manipulables par le calculateur numérique utilisé, le signal tronqué $x_N(n)$ est le résultat de la multiplication du signal discret $x(n)$ de durée infinie par une fenêtre $w(n)$ de durée $T_0 = NT$, appelée fenêtre d'observation.

La version tronquée du signal discrétisé s'écrit donc finalement :

$$x_N(n) = x(n) \cdot w(n) \tag{10.1}$$

La fenêtre la plus intuitive est une fenêtre sans pondération, c'est à dire une fenêtre rectangulaire notée $w_r(n)$ définie par :

$$w_r(n) = \begin{cases} 1 & \text{si } 0 \leq n \leq N-1 \\ 0 & \text{ailleurs} \end{cases} \tag{10.2}$$

La version tronquée du signal discrétisé s'écrit donc finalement dans le cas de la fenêtre rectangulaire :

$$x_N(n) = x(n) \cdot w_r(n) = \sum_{k=0}^{N-1} x(k) \cdot \delta(n-k) \tag{10.3}$$

10.2.2 Conséquences dans le domaine fréquentiel

La TF de (10.1) donne

$$\begin{aligned} X_N(e^{j\Omega}) &= X(e^{j\Omega}) * W(e^{j\Omega}) \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\Theta}).W(e^{j(\Omega-\Theta)})\, d\Theta \end{aligned} \tag{10.4}$$

où $W(e^{j\Omega})$ est la représentation de la fenêtre dans le domaine fréquentiel.

Dans le cas de la fenêtre rectangulaire, on a :

$$W(e^{j\Omega}) = W_r(e^{j\Omega}) = e^{-j\Omega\frac{N-1}{2}} \frac{\sin(N\Omega/2)}{\sin(\Omega/2)} \tag{10.5}$$

Une représentation de la fenêtre $w_r(n)$ ainsi que du module de sa TF est donnée à la figure 10.2.

Exemple 10.2.1 : Calculer la TF $W_r(e^{j\Omega})$ de la fenêtre rectangulaire $w_r(n)$ et démontrez qu'elle s'écrit comme dans l'équation 10.5.

Les conséquences obtenues dans le domaine fréquentiel sont issues de la convolution spectrale illustrée par l'équation 10.4. On obtient donc :
- un lissage de le représentation spectrale qui implique une perte de finesse de l'analyse en fréquence et un masquage des raies trop proches en fréquence,
- des ondulations dans la réponse fréquentielle dues aux effets des lobes secondaires de $W(e^{j\Omega})$, entraînant du bruit et du masquage en amplitude.

10.3 Analyse spectrale par TFD

La troncature du signal sur N points n'est pas le seul effet d'approximation obtenu lors de l'analyse spectrale. En effet, l'équation 10.4 de convolution spectrale est une fonction de la variable continue Ω. Ce calcul n'est donc pas réalisable sur un processeur numérique et doit donc être approximé par la transformée de Fourier discrète (TFD, voir chapitre 4).

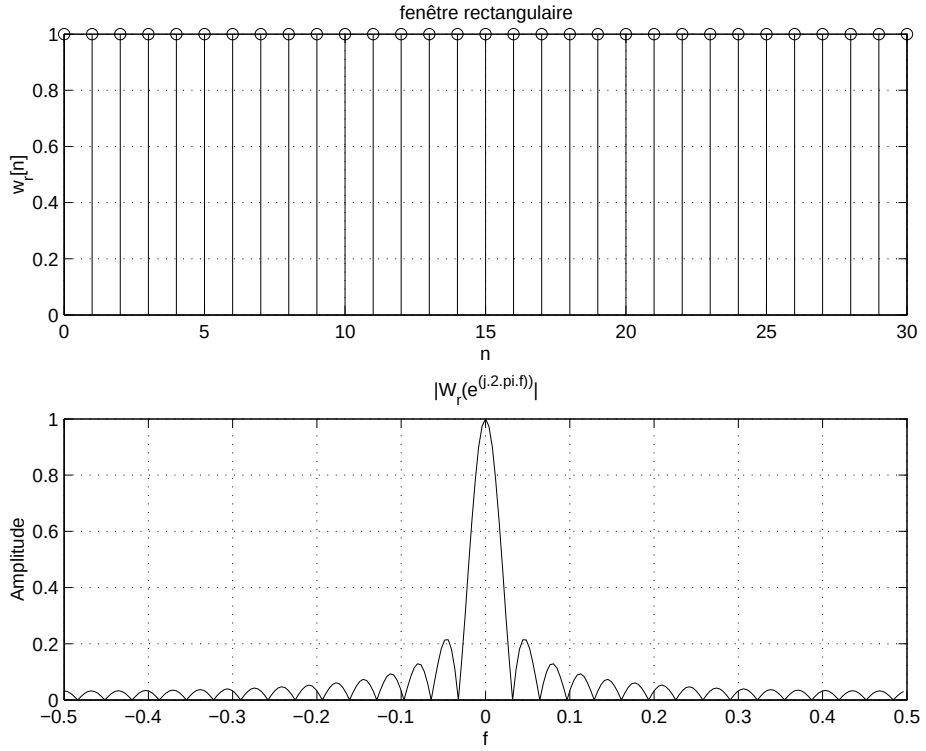


FIG. 10.2: Fenêtre rectangulaire de $N = 31$ points et le module de sa transformée de Fourier.

La dernière étape de l'analyse spectrale est donc le calcul de le TFD du signal fenêtré $x_N(n)$. On obtient alors :

$$X_N(k) = \sum_{n=0}^{L-1} x_N(n) e^{-j\frac{2\pi kn}{N}}, \qquad k = 0, 1, \ldots, L-1, \tag{10.6}$$

où la longueur de la fenêtre N est inférieure ou égale à la longueur de la TFD L. On prendra le plus souvent $L = N$, sauf dans le cas du *zero-padding* (voir section 10.4).

Comme illustrée dans le chapitre 4, la relation entre la TF $X_N(e^{j\Omega})$ de $x_N(n)$ (équation 10.4) et $X_N(k)$ s'écrit :

$$X_N(k) = X_N(e^{j\Omega})\big|_{\Omega=2\pi k/L} \tag{10.7}$$

L'espace entre deux points fréquentiels de l'analyse est donc de $2\pi/L$ et la relation entre les points d'indice k de la TFD et les fréquences continues f_k est donnée par :

$$\Omega_k = \frac{2\pi k}{L} \Leftrightarrow f_k = \frac{k}{LT} \tag{10.8}$$

Exemple 10.3.1 : Déterminez l'expression de la TFD, calculée avec une fenêtre rectangulaire ($N = 8$) du signal $x(n) = 2cos2\pi f_o n$ avec $f_o = 0,2$. Calculez la valeur numérique des $X(k)$ et dessinez à la main ce résultat.

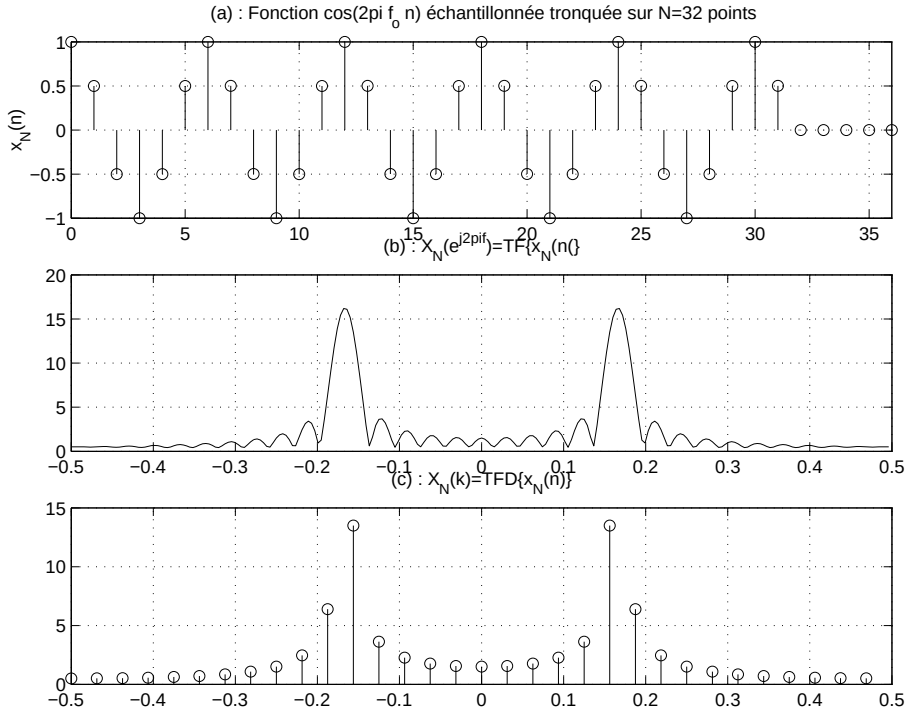


FIG. 10.3: Analyse de Fourier et TFD d'un signal sinusoïdal discret

La figure 10.3 illustre l'analyse spectrale par TFD sur un signal sinusoïdal discret. Le signal $x(t) = cos(2\pi f_0 t)$ est échantillonné puis tronqué sur $N = 32$ points par une fenêtre rectangulaire (voir figure 10.3(a)). La figure 10.3(b) représente la TF $X_N(e^{j\Omega})$ de $x_N(n)$ et illustre le problème de la troncature temporelle présenté dans la section précédente. En effet, la TF d'un cosinus étant composée de deux impulsions de Dirac située en f_0 et $-f_0$, alors la TF du signal fenêtré est composée de la somme des TF de la fenêtre décalées en f_0 et $-f_0$ *(ce que vous devez avoir trouvé si vous avez résolu l'exemple page 127).*
La figure 10.3(c) représente le résultat du calcul de la TFD sur $L = 32$ points du signal $x_N(n)$ et souligne donc l'erreur due à l'échantillonnage en fréquence effectué lors de la TFD. La TFD $X_N(k)$ est une version échantillonnée de $X_N(e^{j\Omega})$, dans laquelle l'espace entre deux points successifs représente Fe/L (ici $Fe/32$). Une erreur visible ici est donc que plusieurs raies spectrales peuvent être distinguées après TFD, pour uniquement deux raies effectives dans le spectre théorique de $x(t) = cos(2\pi f_0 t)$.

10.4 Zéro-Padding

Faire du *zéro-padding* consiste à augmenter artificiellement le nombre d'échantillons en ajoutant des zéros :

$$\{\, x(0)\,,\, x(1)\,,\, \ldots\,,\, x(N-1)\,,\, 0\,,\, 0\,,\, \ldots\,,\, 0\,\} \tag{10.9}$$

jusqu'à obtenir un nouveau nombre d'échantillons égal à L. Ainsi, si l'on calcule maintenant la TFD sur ces données, cela entraîne que les $X(k)$ seront calculés en L fréquences : $f_k = k/L$; elles seront toujours situées dans $[0\ ;\ 1[$, mais comme elles sont plus nombreuses, elles seront

plus rapprochées. Cela aura pour vertu de **révéler plus de détails** invisibles sinon.

La figure 10.4 illustre par exemple quatre analyses de Fourier faites sur une fenêtre carrée de taille initiale : $N = 8$ points. Ces quatre analyses seront effectuées successivement à l'aide de N points, $2N$ points (donc les N échantillons de la fenêtre auxquels on ajoute N zéros), $4N$ points (donc il y a $3N$ zéros) et enfin $8N$ points.

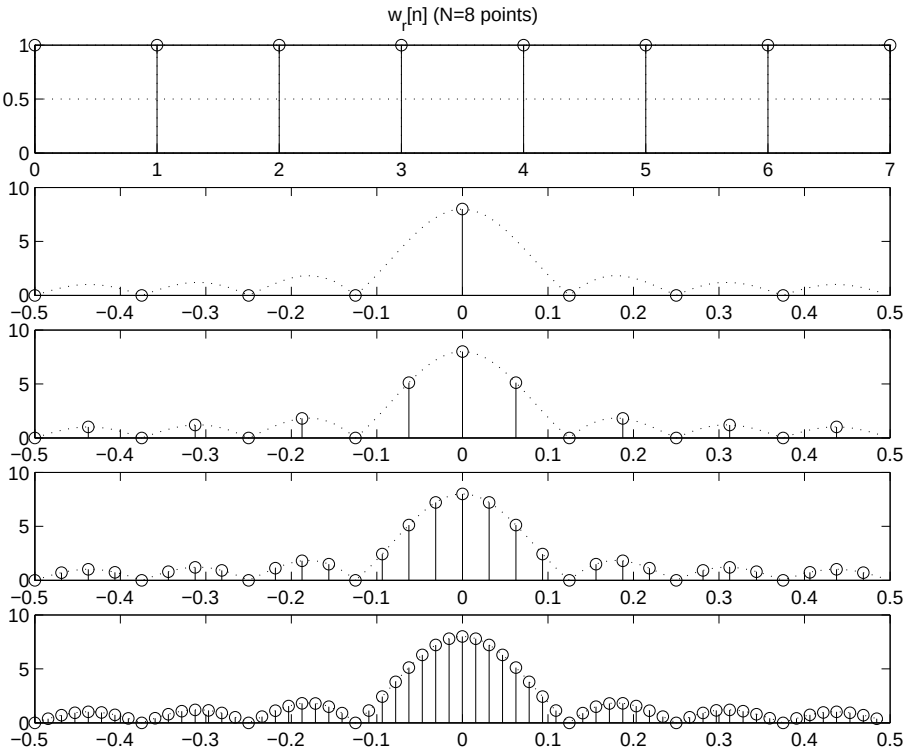


FIG. 10.4: Effet du *zéro-padding* sur l'analyse de Fourier d'une fenêtre rectangulaire de $N = 8$ points.

10.5 Paramètres d'une analyse spectrale

Deux paramètres peuvent être affinés pour améliorer l'analyse :
– N, la longueur de l'analyse (ou T_0),
– le type de fenêtre $w_N(n)$ à appliquer sur le signal.
Cette section illustre ce résultat sur des signaux de type sinusoïdal.

Soit le signal discret constitué de trois raies

$$x(n) = \cos 2\pi f_1 n + \cos 2\pi f_2 n + 0,25 \cos 2\pi f_3 n \tag{10.10}$$

avec $f_1 = 0,175$, $f_2 = 0,2$ et $f_3 = 0,4$. Sa TF s'écrit immédiatement

$$\begin{aligned} X\left(e^{j2\pi f}\right) &= \frac{1}{2}\left[\delta(f+0,175) + \delta(f-0,175) + \delta(f+0,2) + \delta(f-0,2)\right. \\ &\quad \left. +0,25\,\delta(f+0,4) + 0,25\,\delta(f-0,4)\right] \end{aligned} \tag{10.11}$$

Le spectre $X_N\left(e^{j2\pi f}\right)$ de $x_N(n)$ utilisant une fenêtre rectangulaire est donc donné par

$$\begin{aligned} X_N\left(e^{j2\pi f}\right) &= \frac{1}{2}\Big[W_r(e^{j2\pi(f+0{,}175)} + W_r(e^{j2\pi(f-0{,}175)} + W_r(e^{j2\pi(f+0{,}2)} + W_r(e^{j2\pi(f-0{,}2)} \\ &\quad +0,25\,W_r(e^{j2\pi(f+0{,}4)} + 0,25\,W_r(e^{j2\pi(f-0{,}4)}\Big] \end{aligned} \tag{10.12}$$

Ce résultat est illustré à la Figure 10.5. Il est donc clair que d'une part la taille de la fenêtre influe sur l'aptitude de l'analyse de Fourier à séparer deux raies et que d'autre part les lobes secondaires influent nécessairement sur l'aptitude de l'analyse de Fourier à détecter des raies de faible amplitude plus ou moins éloignées d'une de forte amplitude.

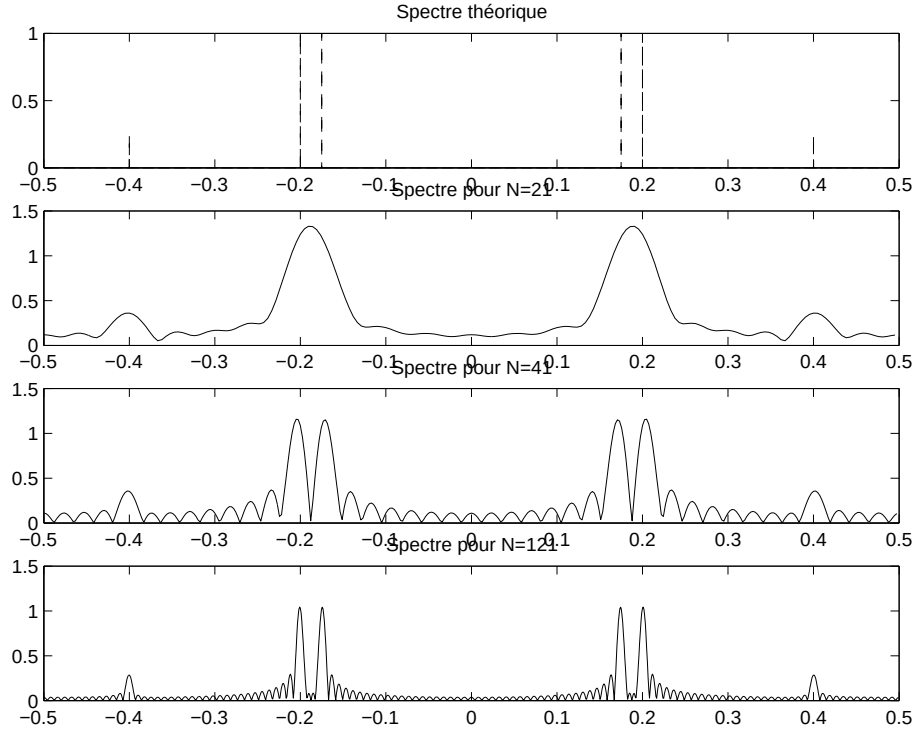


FIG. 10.5: Spectre (théorique puis utilisant des fenêtres rectangulaires de différentes taille N) de trois sinusoïdes

La relation (10.4) montre clairement que la qualité du résultat de la TF, c'est-à-dire l'adéquation avec le résultat théorique que l'on aurait sans troncature, dépend du type de fenêtre utilisé. La section 9.3.1 du chapitre 9 page 120 a résumé les principales des fenêtres utilisées en TNS ainsi que leurs caractéristiques. Le tableau 10.1 résume ces caractéristiques. La figure 10.6 représente un exemple de TF d'une fonction $w(n)$ de fenêtrage et en définit ses paramètres :

- largeur du lobe principal : $\Delta\Omega_m$,
- rapport d'amplitude entre lobe principal et lobe secondaire : $\lambda = 20\log\frac{|W(\Omega_s)|}{|W(0)|}$, où Ω_s est la fréquence correspondant au maximum d'amplitude des lobes secondaires.

Dans une analyse spectrale, on définit deux type de résolution : la résolution en fréquence et la résolution en amplitude.

Type de fenêtre	Rapport d'amplitude entre lobe principal et lobe secondaire λ	Largeur du lobe principal $\Delta\Omega_m$
Rectangulaire	$-13dB$	$4\pi/N$
Bartlett	$-25dB$	$8\pi/N$
Hanning	$-31dB$	$8\pi/N$
Hamming	$-41dB$	$8\pi/N$
Blackman	$-57dB$	$12\pi/N$

TAB. 10.1: Caractéristiques des principales fenêtres

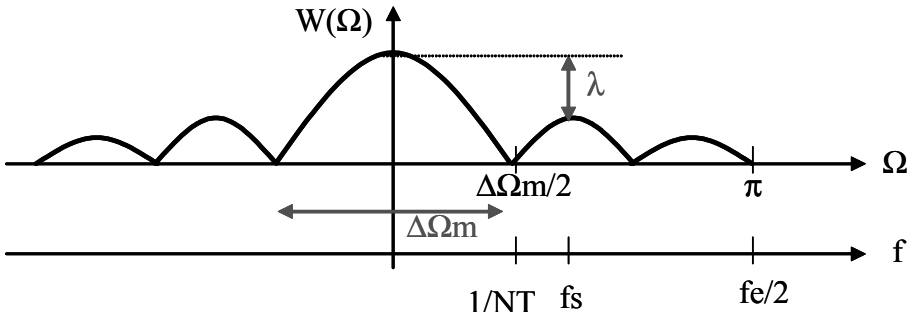


FIG. 10.6: Paramètres d'une fenêtre $w(n)$

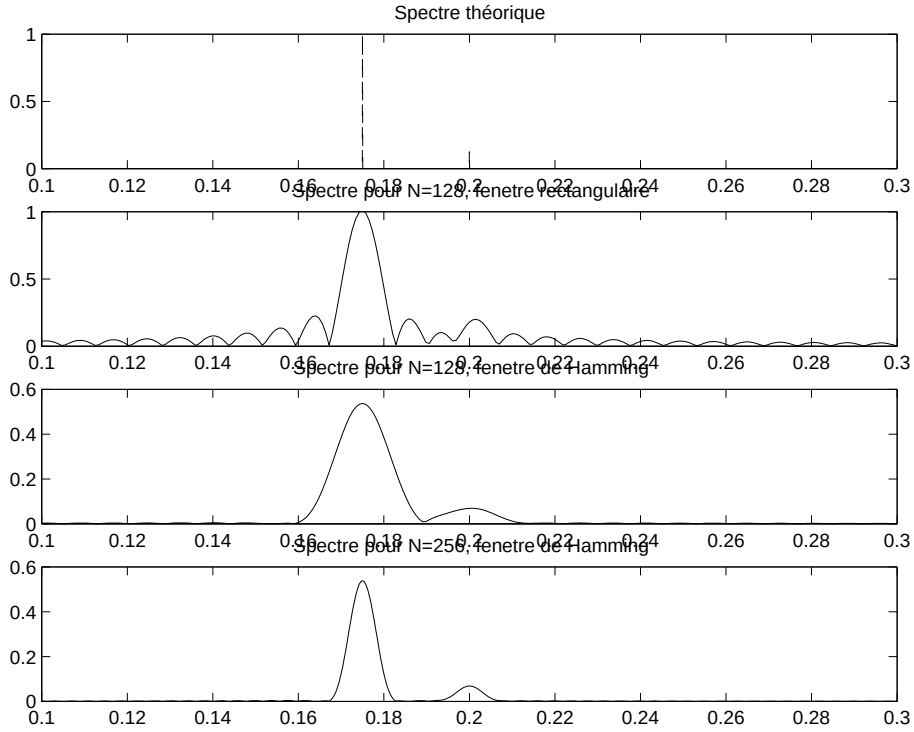


FIG. 10.7: Spectre (théorique puis utilisant des fenêtres rectangulaires et de Hamming de différentes taille N) de trois sinusoïdes

– **Résolution en fréquence** : la finesse en fréquence est la capacité de l'analyseur à distinguer deux raies spectrales proches l'une de l'autre. Clairement, ce phénomène, connu aussi sous le nom de masquage fréquentiel, dépend de la largeur du lobe principal de la réponse fréquentielle $W_N(e^{j\Omega})$ de la fenêtre ($\Delta\Omega_m$ du tableau 10.1). En effet, deux raies proches d'un espace $< \Delta\Omega_m$ se verront confondues après l'analyse par TFD. La figure 10.5 illustre ce phénomène puisque l'analyse pour $N = 21$ ne permet pas de distinguer les raies en $f = 0.2$ et $f = 0.175$, tandis que des valeurs de $N = 41$ ou $N = 121$ le permettent.

  On définit donc la résolution en fréquence comme : $\Delta\Omega_m$.
  Cette résolution peut être améliorée en augmentant le nombre de points N de l'analyse et dépend du type de fenêtre utilisée.

– **Résolution en amplitude** : la finesse en amplitude est la capacité de l'analyseur à distinguer des raies spectrales de faible amplitude ou à distinguer une raie spectrale de faible amplitude proche d'une autre plus importante. Dans cas, ce sont les lobes secondaires de de la réponse fréquentielle $W_N(e^{j\Omega})$ de la fenêtre qui viennent masquer une raie de faible amplitude en ajoutant un bruit à l'analyse. La figure 10.7 illustre ce phénomène puisque l'analyse pour $N = 128$ par fenêtre rectangulaire ne permet pas de distinguer la raie en $f = 0.2$ d'amplitude 0.25 parmi le bruit dû aux lobes secondaires, tandis qu'une fenêtre de Hamming et des valeurs de $N = 128$ ou $N = 256$ le permettent.

  On définit donc la résolution en fréquence comme le rapport d'amplitude entre lobe principal et lobe secondaire : λ.
  Cette résolution n'est pas fonction du nombre de points N de l'analyse mais dépend fortement du type de fenêtre utilisée.

10.6 Conclusion (méthodologie)

L'analyse spectrale de signaux déterministes requiert donc trois études préalables importantes :

1. le signal $x(t) \leftrightarrow X(f)$ doit être correctement **échantillonné** pour obtenir les échantillons $x(n)$ qui seront pris en compte par **bloc (segment) de N échantillons** ;
2. ces bloc seront **conditionnés par multiplication par une fenêtre, de taille** N,$w(n)$, à choisir en fonction des performances attendues :
   (a) la résolution en amplitude souhaitée $\lambda \Rightarrow$ choix du type de fenêtre,
   (b) la résolution en fréquence souhaitée $\Delta\Omega_m \Rightarrow$ choix de N ;
3. finalement les blocs fenêtrés, $x_N(n)$ sont transformés dans le domaine spectral par TFD (calculée par l'algorithme FFT). Le résultat (les $X_N(k)$) est une réplique correcte du vrai spectre $X(f)$ si les erreurs suivantes sont suffisamment faibles :
   (a) erreur de recouvrement de spectres (*Aliasing error* en anglais) (bien choisir la fréquence d'échantillonnage et utiliser un filtre passe-bas antirepliement) ;
   (b) erreur due à la longueur de la fenêtre (plus elle est longue et moins cette erreur est importante, de plus le choix de la fenêtre permet de diminuer l'erreur due aux lobes secondaires) ;

(c) erreur de reconstruction du spectre (il faut utiliser le *zéro-padding* pour diminuer cette erreur).

On peut donc finalement représenter facilement l'analyse de Fourier numérique d'un signal analogique déterministe comme indiquée à la Figure 10.1.

Exemple 10.6.1 : Simulation avec Matlab Dans un script Matlab, rédigez un programme qui permet de :

1. générer et afficher à l'écran $N = 10$ points d'une sinusoïde $x_1(t) = a\cos 2\pi f_1 t$ de $f_1 = 100Hz$ échantillonée à $F_e = 1000Hz$ et d'amplitude $a = 2$;
   (a) effectuer une analyse de Fourier élémentaire (sans *zéro-padding*) de ce signal et afficher le résultat sur l'intervalle principal de fréquences normalisées $[-1/2\ ;\ +1/2[$;
   (b) refaire une analyse de Fourier de ce signal avec divers choix de *zéro-padding* ;
   (c) explorer l'influence de divers types de fenêtrages effectués sur le signal ;
2. regénérer et afficher à l'écran le même signal que précédemment mais cette fois avec $N = 15$ points ;
   (a) refaire la même étude que précédemment ;
   (b) ajouter une seconde sinusoïde et rendre compte de la pertinence de l'analyse effectuée en fonction de l'écart en fréquence des deux composantes élémentaire du signal.
   (c) considérer deux sinusoïdes proches à la limite de résolution constatée au point précédent et ajouter une troisième sinusoïdes faible et éloignée des précédentes ; analysez alors l'influence de différentes fenêtres dans cette situation (résolution des deux sinusoïdes proches et détection de la sinusoïde faible éloignée).

Chapitre 11

Systèmes multi-cadences

Les systèmes de TNS utilisés jusqu'alors ne considèrent qu'une seule fréquence d'échantillonnage Fe. Les systèmes multi-cadences changent la fréquence d'échantillonnage au cours de la chaîne de traitement afin qu'elle soit la plus adaptée aux traitements à réaliser. Par exemple, il est aisé de comprendre qu'après un filtrage sélectif passe-bas il est possible de réduire la fréquence d'échantillonnage à une valeur équivalente à deux fois la bande passante du filtre passe-bas réalisé, sans pour autant transgresser le théorème d'échantillonnage de Shannon. Cette réduction permet de diminuer la complexité du filtrage à réaliser.

Les principaux opérateurs utilisés en traitement du signal multi-cadences sont la décimation (réduction d'un facteur M de la fréquence d'échantillonnage) et l'interpolation (augmentation d'un facteur M de la fréquence d'échantillonnage) ou une combinaison des deux.

11.1 Réduction de la fréquence d'échantillonnage

La réduction de la fréquence d'échantillonnage, ou décimation, ou encore sous-échantillonnage, par un facteur M entier est une opération simple puisqu'il suffit de ne garder que 1 échantillon sur M de la séquence d'origine. Soit un signal continu $x_c(t)$ échantillonné à la période T représenté par la séquence $x(n) = x_c(nT)$, on posera donc le résultat de la décimation de $x(n)$ d'un facteur M par la relation suivante :

$$x_d(n) = x(nM) \tag{11.1}$$

La notation graphique utilisée est $\downarrow M$ (voir figure 11.1). Si la fréquence d'échantillonnage en entrée du décimateur est Fe, alors celle de sortie vaudra $F'e = Fe/M$. La période d'échantillonnage en sortie vaudra quant à elle $T' = MT$.

Afin que l'opération de décimation s'effectue sans recouvrement de spectre, il est nécessaire que le spectre du signal $x_c(t)$ respecte le théorème de Shannon au regard de la fréquence $F'e$. Cela implique que si $X_c(j\omega)$, la TF de $x_c(n)$, est à bande limitée, i.e. $X_c(j\omega) = 0$ pour $|\omega| \leq \omega_0$, alors $x_d(n)$ est une représentation exacte de $x_c(t)$ si $\pi/T' = \pi/(MT) \geq \omega_0$ ou encore $2\pi F'e = 2\pi Fe/M \geq \omega_0$.

Dans le cas où le sous échantillonnage ne pourra s'effectuer sans recouvrement de spectre, il faudra alors en limiter la bande par un filtre passe-bas de fréquence de coupure $F'e/2$.

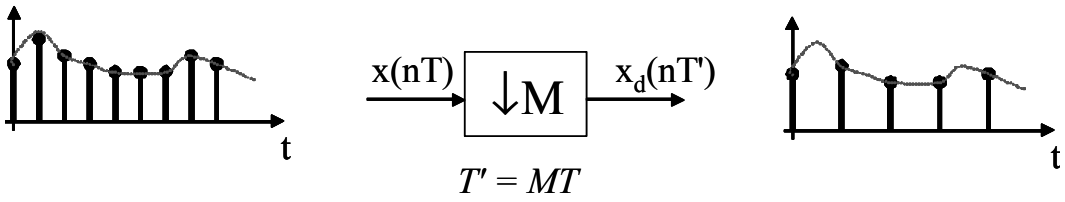


FIG. 11.1: Décimation d'un signal

Il est utile de chercher une relation entre le spectre de $x(n)$ et celui de $x_d(n)$. On peut poser :

$$X(e^{j\Omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\Omega}{T} - \frac{2\pi k}{T}\right)\right) \tag{11.2}$$

De manière similaire :

$$X_d(e^{j\Omega}) = \frac{1}{T'}\sum_{l=-\infty}^{\infty} X_c\left(j\left(\frac{\Omega}{T'} - \frac{2\pi l}{T'}\right)\right) = \frac{1}{MT}\sum_{l=-\infty}^{\infty} X_c\left(j\left(\frac{\Omega}{MT} - \frac{2\pi l}{MT}\right)\right) \tag{11.3}$$

Afin de trouver la relation entre les équations 11.2 et 11.3 les indices des sommes peuvent être exprimés par $l = i + kM$ où $0 \leq i \leq M-1$ et l'équation 11.3 peut être réécrite de la manière suivante :

$$X_d(e^{j\Omega}) = \frac{1}{M}\sum_{i=0}^{M-1}\left[\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c\left(j\left(\frac{\Omega - 2\pi i}{MT} - \frac{2\pi k}{T}\right)\right)\right] \tag{11.4}$$

On obtient alors la relation

$$X_d(e^{j\Omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X(e^{j(\Omega/M - 2\pi i/M)}) \tag{11.5}$$

Le décimation est illustrée à la figure 11.2. L'équation 11.5 montre que le spectre $X_d(e^{j\Omega})$ est composé de M copies de $X(e^{j\Omega})$ mises à l'échelle par un facteur $1/M$ et décalées par des entiers multiples de 2π.

Par conséquent, un décimateur sera en pratique composé filtre passe-bas idéal de fréquence de coupure $F_c = 1/2T' = 1/2TM$ suivi d'un opérateur de décimation $\downarrow M$ comme représenté à la figure 11.3.

11.2 Augmentation de la fréquence d'échantillonnage

L'augmentation de la fréquence d'échantillonnage, ou interpolation, ou encore sur-échantillonnage, par un facteur L entier est une opération consistant à augmenter le nombre d'échantillons de la séquence d'origine d'un facteur L. Soit un signal continu $x_c(t)$ échantillonné à la période T représenté par la séquence

$$x(n) = x_c(nT), \tag{11.6}$$

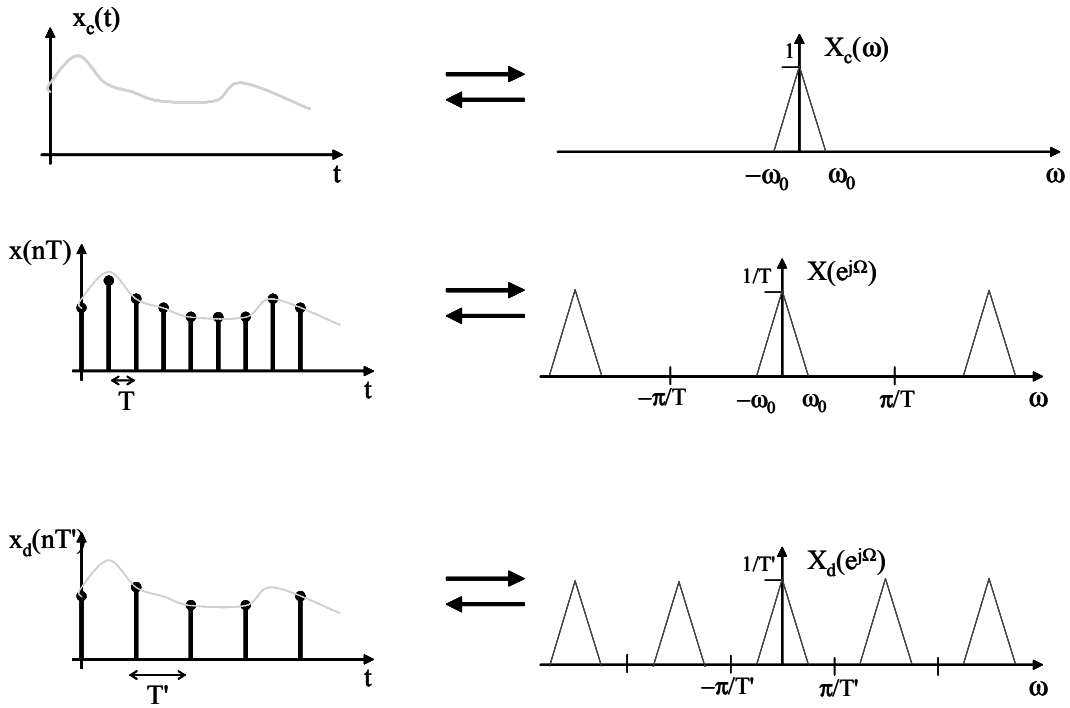


FIG. 11.2: Représentation d'un signal continu $x_c(t)$ et de son échantillonnage à 2 fréquences différentes et de leurs spectres respectifs

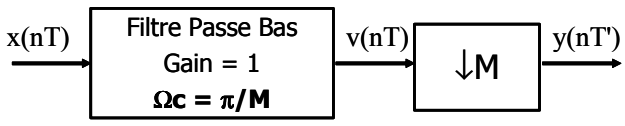


FIG. 11.3: Décimateur : filtrage passe-bas suivi d'une décimation

on posera donc le résultat de l'interpolation de $x(n)$ d'un facteur L par la relation suivante :

$$x_i(n) = x_c(nT') = x_c(nT/L), \tag{11.7}$$

avec $T' = T/L$ la période d'échantillonnage en sortie de l'interpolation ou $F'e = Fe.L$ la fréquence d'échantillonnage en sortie de l'interpolation.
À partir des équations 11.6 et 11.7, on peut déduire que :

$$x_i(n) = x_c(nT/L) = x(n/L), \qquad n = 0, \pm L, \pm 2L, \ldots \tag{11.8}$$

11.2.1 Élévateur de fréquence d'échantillonnage

On définit par élévateur de fréquence, le système décrit à la figure 11.4 consistant en l'ajout de $L-1$ zéros entre deux échantillons successifs de la séquence d'entré $x(n)$ défini par :

$$x_e(n) = \begin{cases} x(n/L), & n = 0, \pm L, \pm 2L, \ldots \\ 0, & ailleurs \end{cases} \tag{11.9}$$

ou de manière équivalente :

$$x_e(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot \delta(n - kL) \tag{11.10}$$

La notation graphique utilisée est $\uparrow L$ (voir figure 11.4). Si la fréquence d'échantillonnage en entrée de l' élévateur de fréquence est Fe, alors celle de sortie vaudra $F'e = Fe.L$. La période d'échantillonnage en sortie vaudra quant à elle $T' = T/L$.

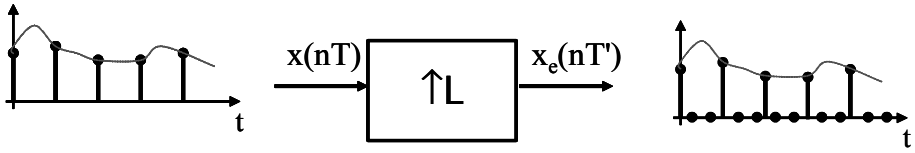


FIG. 11.4: Élévateur de fréquence d'un facteur L : ajout de $L-1$ zéros entre deux échantillons

L'opération $\uparrow L$ peut être étudiée dans le domaine fréquentiel afin de comprendre sont utilité dans le cadre de l'interpolation. La TF de $x_e(n)$ s'exprime :

$$X_e(e^{j\omega T'}) = \sum_{n=-\infty}^{\infty} x_e(nT')e^{-j\omega nT'} \tag{11.11}$$

$$= \sum_{n=0,\pm L,\pm 2L,\ldots} x\left(\frac{nT}{L}\right) e^{-j\omega nT'} \tag{11.12}$$

En posant $n = kL$, on obtient :

$$X_e(e^{j\omega T'}) = \sum_{k=-\infty}^{\infty} x(kT)e^{-j\omega kLT'} \tag{11.13}$$

$$= \sum_{k=-\infty}^{\infty} x(kT)e^{-j\omega kT} \tag{11.14}$$

$$= X(e^{j\omega T}) \tag{11.15}$$

L'ajout de zéros n'a donc aucun effet sur le spectre, si ce n'est l'écartement de l'intervalle de périodisation d'un facteur L. Ceci est illustré à la figure 11.5. On s'aperçoit alors qu'un filtrage passe-bas idéal de gain L et de fréquence de coupure $Fc = 1/2T$ permet de retrouver la forme du spectre correspondant à $x_c(t)$ échantillonné à T'.

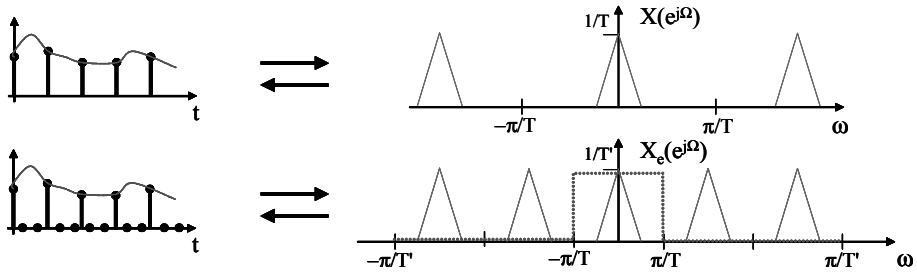


FIG. 11.5: Influence sur le spectre d'un signal interpolé par des zéros

11.2.2 Interpolation

Un interpolateur (voir figure 11.6) sera donc défini comme la succession d'un élévateur de fréquence $\uparrow L$, suivi d'un filtrage passe-bas idéal de gain L et de fréquence de coupure $Fc = 1/2T$ (ou $\Omega_c = \pi/L$).

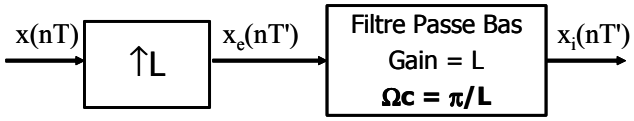


FIG. 11.6: Principe de l'interpolation d'un signal

D'autres systèmes d'interpolation sont connus (interpolation linéaire ou par TFD) mais ne seront pas détaillés ici.

11.2.3 Multiplication de la fréquence d'échantillonnage par un facteur rationnel

La figure 11.7 représente un schéma permettant de multiplier la fréquence d'échantillonnage d'un système par un facteur rationnel $R = L/M$. On aura alors $F'e = R.Fe = L.Fe/M$ ou encore $T' = T/R = T.M/L$. Pour cela, il suffit d'effectuer tout d'abord une interpolation par un facteur $\uparrow L$, puis une décimation par un facteur $\downarrow M$. Les deux filtres présents dans la décimation (voir figure 11.3) et dans l'interpolation (voir figure 11.6) peuvent alors réunis dans un seul et même filtre dont la fréquence de coupure F_c dépendra des valeurs relatives de L et M selon :

– soit $R > 1 \Leftrightarrow M < L \Leftrightarrow F'e > Fe \quad \Rightarrow \quad F_c = \frac{1}{2T}$,
– soit $R < 1 \Leftrightarrow M > L \Leftrightarrow F'e < Fe \quad \Rightarrow \quad F_c = \frac{1}{2T'}$.

On peut également écrire :

$$F_c = min\left(\frac{1}{2T}, \frac{1}{2T'}\right) \tag{11.16}$$

ou encore

$$\Omega_c = min\left(\frac{\pi}{L}, \frac{\pi}{M}\right) \tag{11.17}$$

car le filtre passe-bas travaille à une fréquence d'échantillonnage $L.Fe$ (ou encore $M.F'e$).

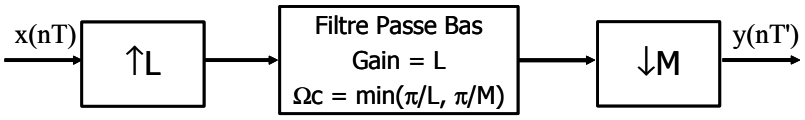


FIG. 11.7: Multiplication de la fréquence d'échantillonnage par un facteur rationnel $R = L/M$

Chapitre 12

Travaux Dirigés en Traitement Numérique du Signal

12.1 Echantillonnage

12.1.1 Chaîne de TNS

On s'intéresse à une chaîne de TNS du type de la figure 12.1

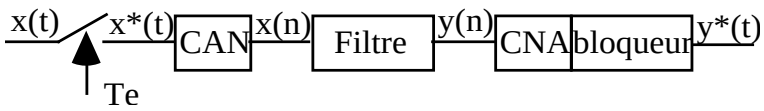


FIG. 12.1: Chaîne de TNS

Le signal $x(t) = A.e^{-at}.sin(\omega_0.t).u(t)$ correspondant à une réponse transitoire d'un système oscillant amorti est échantillonné à une période Te qui permette de limiter le recouvrement spectral.

1. Calculer la TF X(f) de ce signal.
2. Tracer le module du spectre $|X(f)|$ de ce signal, et précisez le lieu du maximum F_{max}, ainsi que sa valeur (A.N. $\omega_0 = 2\pi 4 rad/s;\ a = 7;\ A = 20$).
3. Ce spectre présente un support de durée infinie (à démontrer). Calculer et dessiner le spectre de $x*(t)$ pour une période T_e quelconque.
4. Pour limiter l'effet de recouvrement spectral, on choisit de considérer la partie utile de $x(t)$ sur un support borné. Ainsi, toute composante spectrale dont l'amplitude ne dépasse pas 1% de l'amplitude maximale F_{max} du spectre sera considérée comme négligeable. Calculer la fréquence F_M au delà de laquelle l'amplitude des raies devient négligeable.
5. Calculer F_e telle que le recouvrement n'entraîne une erreur sur le spectre initial ne dépassant pas 1% de l'amplitude du spectre en F_M.
6. Existe-t-il d'autres solutions qui permettent de limiter le recouvrement spectral.

12.1.2 Échantillonnage d'un signal

1. Soit le signal $x(t) = e^{-at}.u(t)$, calculez et dessinez sa transformée de Fourier $X(\omega)$. On donnera les valeurs du modules en $\omega = 0,\ a,\ 10a$.
2. Calculez l'énergie du signal $x(t)$.
3. On échantillonne $x(t)$ à une période T. Calculez la transformée de Fourier $X_e(\omega)$ du signal échantillonné $x_e(nT)$.
4. Dessinez approximativement le module du spectre lorsque $T = \pi/10a$. Expliquez quels problèmes peuvent survenir lors de l'échantillonnage du signal $x(t)$.
5. Donner l'expression de l'énergie en fonction de la bande de fréquence $[-B, \ldots + B]$ considérée. On rappelle que la primitive de $1/(1 + x^2)$ est $arctg(x)$. Trouver B donnant 90% de l'énergie totale du signal. Proposez, à partir de ces résultats, une solution pour limiter l'effet de l'échantillonnage en considérant qu'une bande de fréquence représentant 90% de l'énergie du signal suffit à caractériser le signal.

12.2 Analyse des filtres numériques

12.2.1 Cellule élémentaire du premier ordre RII

Soit le système qui, à la suite de données $x(n)$, fait correspondre la suite $y(n)$ telle que :

$$y(n) = x(n) + b.y(n-1)$$

où b est une constante.

1. Donner les réponses impulsionnelles et indicielles de ce système. par deux méthodes (suite numérique, transformée en Z). Que peut on dire de la stabilité du filtre.
2. Étudier l'analogie avec le système continu de constante de temps t, échantillonné avec la période T.
3. Étudier la réponse fréquentielle du filtre.
4. Donner la structure de réalisation du filtre.

12.2.2 Cellule du second ordre RII purement récursive

Soit le système qui, à la suite de données $x(n)$, fait correspondre la suite $y(n)$ telle que :

$$y(n) = x(n) - b_1.y(n-1) - b_2.y(n-2)$$

1. Donner la fonction de transfert en Z du système.
2. En déduire la réponse impulsionnelle du filtre numérique.
3. Étudier la réponse fréquentielle du filtre. On regardera plus particulièrement l'influence des coefficients b_1 et b_2 sur les pôles de la fonction de transfert $H(z)$.
4. Tracer le diagramme des pôles et zéros.
5. Donner les structures de réalisation.

12.2.3 Analyse d'un filtre numérique RIF

Soit un filtre à réponse impulsionnelle finie dont le schéma de fonctionnement dans le domaine temporel est donné figure 12.2.
On pose T_e la période d'échantillonnage du système numérique, $T_e = 1$.

12.2.3.1 Etude de la réponse fréquentielle

1. Donner les expressions de l'équation aux différences finies ainsi que la fonction de transfert en Z.
2. Déterminer et tracer la réponse impulsionnelle $h(n)$ du filtre, lorsque $h_1 = h_5 = 0.1$, $h_2 = h_4 = -0.3$, $h_3 = 0.5$.
3. Calculer la réponse fréquentielle $H(e^{j\Omega})$ du filtre. Déterminer son module et sa phase. On note que :
$$e^{-j\Omega_1} + e^{-j\Omega_2} = 2 \times e^{-j\frac{(\Omega_1+\Omega_2)}{2}} \times \cos(\frac{\Omega_2 - \Omega_1}{2})$$
4. Donner les valeurs du module en $\Omega = 0,\ \pi/2,\ ,\pi\ ,2\pi$.
5. Tracer approximativement son module. De quel type de filtre s'agit-il ?

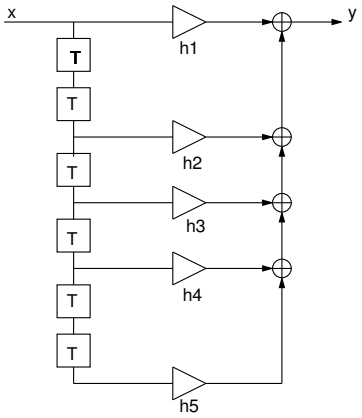


FIG. 12.2: Filtre FIR

12.2.3.2 Complexité de l'implantation du filtre sur un DSP

On suppose que l'on dispose d'un calculateur de type DSP (spécialisé dans le traitement du signal) où l'opération de base est du type $y = a \times x + b$. Ce type de calculateur est capable de mener en un cycle d'horloge une accumulation, ou une multiplication/addition.

1. Quelle est la complexité du filtre tel que réalisé figure 12.2 en nombre de multiplications et d'additions. Quel est le nombre de mots mémoires nécessaires à l'exécution du calcul. (on considérera une complexité pour N points du signal d'entrée traités).
2. On dispose d'un machine réalisant multiplication et addition en parallèle en un cycle de $100ns$. Quelle est dans ce cas la fréquence d'échantillonnage maximale du signal ?
3. Donner un schéma de principe de réalisation du filtre dans le domaine fréquentiel. Quelle est la complexité algorithmique de cette nouvelle solution (opérations et mots mémoire) ? Comparer les deux approches, la méthode fréquentielle est-elle exacte ?

12.2.3.3 Implantation du filtre en virgule fixe cadrée à gauche

Les données d'entrée et de sortie sont codées sur b bits utiles en complément à deux. La dynamique des nombres est $[-1, 1]$. La machine de traitement possède uniquement des opérateurs travaillant sur b bits. Les calculs se font par arrondi.

1. Rappeler brièvement le modèle de quantification d'un signal numérique. Quelles sont dans le cas du filtre les sources de bruits.
2. On néglige le bruit engendré par le signal en entrée du filtre. Donner la puissance de bruit σ_s^2 en sortie du filtre.
3. On tient compte maintenant du fait que le signal d'entrée est bruité par l'opération de quantification (on note la puissance de ce bruit σ_e^2). Quelle est maintenant la puissance du bruit en sortie ? Que conclure ?
4. Le signal d'entrée est une signal sinusoïdal de $1V$ crête. Quel est la puissance de ce signal ? Déterminer le rapport signal à bruit en entrée et en sortie du filtre.
5. Quel serait le nombre de bits pour obtenir un RSB en sortie supérieure à $40dB$?

6. Proposer une solution pour éviter les débordements lors du filtrage ?

On rappelle que le rapport signal à bruit est donné par la relation :

$$RSB = \frac{Puissance\ du\ signal}{Puissance\ du\ bruit} = \frac{\sigma_x^2}{\sigma_b^2}$$

$$RSB_{dB} = 10log\left(\frac{\sigma_x^2}{\sigma_b^2}\right)$$

12.2.4 Filtrage numérique RIF (1)

Soit le filtre numérique suivant : $H(z) = 0,1.(z^{-1} + z^{-3}) + 0,2.z^{-2}$
On posera T_e, période d'échantillonnage, égal à 1s.

1. Donnez et tracez sa réponse impulsionnelle $h(n)$. Quelles sont ses caractéristiques.
2. Calculez la réponse fréquentielle du système. Tracez son module et sa phase. On montrera que la phase du filtre est linéaire. Donnez la fréquence de coupure à -3dB.
3. Quel type de filtre est réalisé ?
4. Donnez l'expression de la sortie $y(n)$ du filtre en fonction de l'entrée $x(n)$. Calculez et dessinez le signal de sortie du filtre $y(n)$ pour $n = 0 \dots 7$ lorsque l'entrée est :

$$x(n) = \begin{cases} 1 & \text{n=0, 1} \\ 0 & \text{ailleurs} \end{cases}$$

12.2.5 Filtrage numérique RIF (2)

Soit le filtre de réponse impulsionnelle suivante :

$$h(n) = a_0\delta(n) + a_1\delta(n-1) + a_2\delta(n-2) + a_1\delta(n-3) + a_0\delta(n-4)$$

1. Donner l'expression de l'équation aux différences finies de ces filtres et de sa fonction de transfert en Z
2. En déduire la réponse fréquentielle $H(e^{j\Omega})$, puis l'expression de son module et de sa phase.
3. Calculer les valeurs du module pour $\Omega = 0, \pi, 2\pi, \frac{\pi}{2}$
4. Déterminer où se trouve le minimum et le maximum de ce module. En déduire quel type de filtre peut être réalisé par $h(n)$.
5. Trouver les valeurs des coefficients a_i tels que $|H(e^{j\Omega})|$ soit égal à $1, 0.5, 0$ en, respectivement, $\Omega = 0, \frac{\pi}{2}, \pi$, avec $a_i \geq 0 \quad \forall \quad i$
6. Chercher F_c la fréquence de coupure à $-3dB$ du filtre si la fréquence d'échantillonnage $F_e = 40$kHz

12.2.6 Filtrage numérique RIF cascade

Soit les filtres du second ordre suivant :

$$H^i(z) = b_0^i + b_1^i z^{-1} + b_2^i z^{-2}, \quad i = 0 \cdots 2$$

12.2.6.1 Étude des fonctions de transfert

1. Donner l'expression de l'équation aux différences finies de ces filtres
2. Donner une structure de réalisation de ces filtres
3. Donner le synoptique d'une mise sous forme parallèle de ces filtres que l'on notera $M(z)$
4. Donner le synoptique d'une mise sous une forme cascade de ces filtres que l'on notera $N(z)$
5. En déduire les fonctions de transfert $M(z)$, $N(z)$ en fonction des b_j^i

12.2.6.2 Étude de la complexité d'une implantation cascade

1. Quelle est la complexité du filtrage type $N(z)$ en nombre de multiplications et d'additions ?
2. On considère une signal audio de qualité HiFi en entrée du filtre ($F_e = 44.1kHz$), quel doit être le temps de cycle et la capacité mémoire d'une machine réalisant multiplication et addition en parallèle ?

12.2.6.3 Etude des bruits de calcul

Les données de l'entrée et de la sortie sont codées sur des mots de b bits utiles en complément à 2. La dynamique des nombres est $[-1, 1]$. La machine de traitement possède uniquement des opérateurs travaillant sur b bits.

1. Exprimer le bruit en sortie d'un filtre $H^i(z)$ en fonction du bruit en entrée dans les cas où :
   (a) l'influence des coefficients multiplicatifs est négligeable
   (b) les coefficients multiplicatifs influent sur le puissance du bruit
2. En déduire le bruit en sortie du filtre $N(z)$ en considérant que le bruit en entrée du filtre provient de la conversion analogique numérique. L'ordre de la mise en cascade a-t-elle une influence ?
3. Quelle est la valeur maximale du signal d'entrée d'un filtre $H^i(z)$ pour qu'il n'y ait pas de débordement de calcul ? Exprimer ce résultat en fonction des b_j^i
4. En déduire la valeur maximale du signal d'entrée du filtre $N(z)$ pour éviter tout débordement.

12.2.6.4 Application numérique

Les coefficients sont les suivants pour les trois filtres élémentaires :

$$b_0^i = 0.5, \quad b_1^i = 0.75, \quad b_2^i = 0.5, \quad i = 0 \cdots 2$$

1. Donner la réponse impulsionnelle puis fréquentielle des filtres $H^i(z)$, puis du filtre $N(z)$
2. Donner le bruit en sortie du filtre $N(z)$, puis sa dynamique maximale en entrée. Expliquer comment empêcher les débordements.

12.2.7 Étude des bruits de calcul dans les filtres numériques RII

On s'intéresse à l'estimation du bruit de calcul d'un filtre RII, pour un traitement en nombre réel, en virgule fixe cadrée à gauche.

12.2.7.1 Cellule du second ordre

Les calculs d'une cellule du second ordre d'un filtre RII sont donnés par l'équation ci-dessous où les a_i et b_i sont des constantes, que l'on supposera non entachées de bruit, et de module ≤ 1. La fonction de transfert $H(z)$ est donnée ci-dessous. On prendra $N = M = 2$.

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{M} b_i.z^{-i}}{1 + \sum_{i=1}^{N} a_i.z^{-i}} \Rightarrow y(n) = \sum_{i=0}^{M} b_i.x(n-i) - \sum_{i=1}^{N} a_i.y(n-i)$$

Le signal acquit est entaché d'un bruit σ_e^2, chaque résultat $y(n)$ est entaché d'un bruit σ_n^2.

1. Rappeler la modélisation d'un bruit d'arrondi en virgule fixe cadrée à gauche codé sur b bits.
2. Donner la structure directe de réalisation, puis représenter, pour le codage considéré, les bruits d'arrondis générés par les opérations. Le format de codage est constant tout au long des calculs et est celui du 1.
3. Donner la puissance totale du bruit dans le filtre. Cela représente-t-il la puissance en sortie du filtre ? Expliquer.
4. Représenter le filtre sous la forme directe en écrivant $H(z) = N(z).\frac{1}{D(z)}$ (voir cours). En déduire le bruit en sortie du filtre en utilisant la formule de filtrage d'un bruit donné en cours.
5. Représenter le filtre sous la forme canonique, c'est à dire en écrivant $H(z) = \frac{1}{D(z)}.N(z)$ (voir cours). En déduire le bruit en sortie du filtre.
6. Refaire les questions précédentes pour la structure directe d'un filtre RII du 4^o ordre.

12.2.7.2 Cellule du quatrième ordre sous forme cascade

On s'intéresse maintenant au calcul d'un filtre du 4^o ordre. Celui-ci est mis en œuvre par deux cellules du second ordre, cascadées.

1. Représentez le graphe flot de calculs de ce filtre. On fera apparaître les coefficients de la première cellule $a_{1,i}$ et $b_{1,j}$, et les coefficients de la seconde cellule $a_{2,i}$ et $b_{2,j}$.
2. Les bruits dans calcul de la première cellule sont étudiés comme la section précédente. Étudiez les bruits de calcul dans la seconde cellule, et en déduire la puissance du bruit dans la seconde cellule.
3. Déduire des questions précédentes la formule générale des bruits dans les filtre RII sous forme transverse et cascade. Que peut on conclure ?

12.2.7.3 Dynamique d'un filtre du septième ordre

On s'intéresse au codage des données pour les calculs d'une filtre à réponse impulsionnelle infinie.

$$y(n) = \sum_{i=0}^{7} b_i.x(n-i) - \sum_{i=1}^{7} a_i.y(n-i)$$

Les différentes variables sont bornées en module. On a $|a_i| < a$ et $|b_j| < b$ tout i et tout j, et $\sum_{i=0}^{7} b_i < B$, $\sum_{i=1}^{7} a_i < A$ et $|x(n-i)| < X$.
On cherchera à déterminer la dynamique du résultat du calcul dans le cas transverse.

1. Montrez que $|y_p| < B.X.(1+\beta)^p$ lorsque p est inférieur ou égal à 7. On donnera les majorants de $|y_0|$ à $|y_4|$. On rappelle que $x(i) = 0$ pour $i < 0$.
2. On suppose que pour $p >> 7$, $|y_p| < Y$, déterminez la dynamique maximale des $x(n-i)$, X, en fonction de A, B et Y pour assurer la convergence de dynamique des $|y_p|$.
3. On code les coefficients a_i, b_j et les échantillons du signal $x(n-i)$ sur 10 bits. Déterminez le format du codage des a_i, b_j et $x(n-i)$, en virgule fixe, pour que le résultat y_n soit codé en virgule fixe cadrée à gauche. On considérera que Y est la majorant de $|y_7|$. On donne $A = 5.\alpha$ et $B = 6.\beta$.
4. Formulation cascade du filtre. Le même filtre que précédemment peut se calculer sous une forme cascade, c'est à dire à partir des résultats cumulés de quatre filtres du second ordre (chaque filtre du second ordre est nommé cellule) :
   pour $c = 1$ à 4 : $y_n^c = \sum_{i=0}^{2} b_i^c.x^c(n-i) - \sum_{i=1}^{2} a_i^c.y^c(n-i)$
   et le résultat d'une cellule est l'entrée de la cellule suivante : $x_n^c = y_n^{c-1}$
   les entrées de la première cellule sont les échantillons du signal avec $|x_n| < X$
   la sortie du filtre y_n est le résultat de la dernière cellule : $y_n = y_n^4$

   Pour chaque cellule, on a : $|a_i^c| < \alpha^c$, $|b_i^c| < \beta^c$, $\sum_{i=1}^{2} |a_i^c| < A$, $\sum_{i=0}^{2} |b_i^c| < B$, avec $A = 2.\alpha$, $B = 2.\beta$. Pour la première cellule, les entrées sont bornées en module, et la borne X est connue. Montrez que $|y_p| < A.X.(1+\beta)^p$ lorsque p est inférieur ou égal à 2. On donnera les majorants de $|y_0^1|$ à $|y_2^1|$. On rappelle que $x(i) = 0$ pour $i < 0$.
5. Pour la première cellule, on suppose que pour $n >> 2$, $|y_n^1| < Y^1$, déterminez la dynamique maximale des $x(n-i)$, X, en fonction de A, B et Y^1 pour assurer la convergence de dynamique des $|y_n^1|$.
6. On code les coefficients des cellules et les échantillons du signal $x(n-i)$ sur 10 bits. Déterminez le format du codage des a_i^1, b_j^1 et $x(n-i)$, en virgule fixe, pour que le résultat soit codé en virgule fixe cadrée à gauche. On considérera que Y^1 est la majorant de $|y_n^1|$.
7. Y^1 est maintenant le majorant des entrées de la seconde cellule. Exprimez, suivant une démarche similaire à la précédente, le majorant des sorties de la second cellule Y^2.
8. Exprimer le majorant Y des sorties du filtre (obtenu à la sortie de la quatrième cellule).
9. Comparez aux résultats du 2.

12.3 Synthèse des filtres RII

12.3.1 Filtre passe bas du deuxième ordre

12.3.1.1 Étude par le gabarit

On désire réaliser un filtre numérique $H(z)$, équivalent à un filtre analogique passe-bas de Chebyshev respectant le gabarit suivant :

f_p	f_a	δ_1	δ_2
1 kHz	3 kHz	-3 dB	-20 dB

Après avoir dessiné le gabarit analogique équivalent, et déduit l'ordre du filtre, donnez la fonction de transfert obtenue par la transformation bilinéaire. On posera $f_e = 10kHz$.
Démontrer que

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

On rappelle que la transformation bilinéaire est obtenue par

$$p = f(z) = 2.Fe\frac{1 - z^{-1}}{1 + z^{-1}}$$

12.3.1.2 Étude directe

On désire réaliser un filtre numérique $H(z)$ équivalent à un filtre analogique de Chebyshev passe-bas $H(j\omega)$ du deuxième ordre qui présente une fréquence de coupure F_c de 1 kHz. La fréquence d'échantillonnage Fe sera de 10 kHz. Les fonctions de transfert du filtre de Chebyshev normalisé puis dénormalisé sont les suivantes :

$$H_{Norm}(j\omega) = \frac{1}{1 + j0.995\omega - 0.907\omega^2} \tag{12.1}$$

$$H(j\omega) = \frac{1}{1 + j0.995\omega - 0.907\omega^2} \tag{12.2}$$

$f_0 = 5.2$ kHz.
Les réponses fréquentielles des gain, phase et temps de propagation de groupe sont données par les équations suivantes :

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

$$\phi(\omega) = Arg\left(H(z)\right)$$

$$\rightarrow(\omega) = -\frac{d\phi(\omega)}{d\omega}$$

Faire la synthèse par la méthode bilinéaire du filtre $H(j\omega)$ afin d'obtenir $H(z)$. On étudiera l'influence de la distorsion en fréquence impliquée par la méthode.

12.3.2 Filtre passe haut

On désire réaliser un filtre RII dont la réponse en fréquence idéale est définie par le gabarit fréquentiel ci-dessous. La période d'échantillonnage T est fixée à $10._{\pi}\mu s$.
– Atténuation de 3dB pour $\Omega_c = 0.4\pi$ rad.
– Atténuation supérieure à 20dB pour $0 \leq \Omega \leq 0.1\pi$ rad.
– Atténuation inférieure à 1dB pour 0.5π rad $\leq \Omega \leq \pi$ rad.

1. Tracer le gabarit numérique du filtre en pulsation ω.
2. On désire réaliser le filtre numérique par la méthode de la transformation bilinéaire en ayant une réponse fréquentielle monotone dans la bande passante.
   (a) Quel type de filtre analogique doit on prendre ?
   (b) Dessiner le gabarit analogique du filtre équivalent.
   (c) A partir du gabarit prototype équivalent, déterminer l'ordre et donner la fonction de transfert normalisée $H_N(p)$.
   (d) Donnez l'expression littérale (sans application numérique) du filtre analogique équivalent $H(p)$ en se rappelant que le filtre numérique devra passer à -3dB en Ω_c.
   (e) Déterminer la fonction de transfert $H(z)$ du filtre numérique. Mettre $H(z)$ sous la forme littérale suivant. Exprimer les coefficients a_i et b_i en fonction de Ω_c.

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

   (f) Donner la fonction de transfert $H(z)$ sous forme numérique.
3. Donner l'expression de $H(ej\Omega)$, puis calculer son module pour $\Omega = 0,\ \pi/2,\ \pi$. Dessiner la réponse fréquentielle globale du filtre numérique.
4. Donner l'équation aux différences du filtre numérique puis sa structure canonique de réalisation.
5. On désire réaliser ce même filtre numérique par la méthode de l'invariance impulsionnelle.
   (a) A partir de l'expression littérale du filtre analogique équivalent $H(p)$, donner la réponse impulsionnelle $h(t)$. On rappelle que :

$$\frac{p+a}{(p+a)^2 + \omega_0^2} \Longleftrightarrow e^{-a.t} cos(\omega_0 t)$$

   (b) Donner l'expression de la fonction de transfert en z $H(z)$ du filtre.
   (c) Que peut-on dire de la réponse fréquentielle numérique par rapport au gabarit.

12.4 Synthèse des filtres RIF

12.4.1 Méthode du fenêtrage (2 heures)

On considère un filtre numérique idéal défini par la réponse fréquentielle suivante :

$$H(e^{j\Omega}) = \begin{cases} 1 & pour \quad 0 \leq |\Omega| \leq \Omega_c \\ 0 & pour \quad \Omega_c < |\Omega| < \pi \end{cases}$$

1. Dessiner la réponse en fréquence (module et phase) sur l'intervalle $[-2\pi \dots 2\pi]$, préciser le type de filtre obtenu.
2. Donner l'expression des coefficients $h(n)$ de la réponse impulsionnelle du filtre. Calculer $h(n)$ pour $n = [-5, \dots, +5]$ pour le cas où $\Omega_c = \pi/4$. Dessiner la forme générale de cette réponse. Le filtre est il causal ?
3. On recherche les relations entre cette réponse impulsionnelle et celles de filtres passe-haut, passe-bande, réjecteur-de-bande. Montrer que :
   - pour les filtres passe-haut de fréquence de coupure $\Omega'_c = \pi - \Omega_c$ les coefficients de la réponse impulsionnelle sont : $h_{PH}(n) = (-1)^n h(n)$,
   - pour les filtres passe-bande de fréquence de coupure basse $\Omega_1 = \Omega_0 - \Omega_c$ et de fréquence de coupure haute $\Omega_2 = \Omega_0 + \Omega_c$ (Ω_0 fréquence centrale), les coefficients de la réponse impulsionnelle sont : $h_{PB}(n) = 2h(n)cos(n\Omega_0)$,
   - pour les filtres rejecteur-de-bande de fréquence de coupure basse $\Omega_1 = \Omega_0 - \Omega_c$ et de fréquence de coupure haute $\Omega_2 = \Omega_0 + \Omega_c$ (Ω_0 fréquence centrale), les coefficients de la réponse impulsionnelle sont : $h_{RB}(0) = 1 - h_{PB}(0), h_{RB}(n) = -h_{PB}(n)$.
4. Calculer les coefficients $h_{PH}(n)$, $h_{PB}(n)$, $h_{RB}(n)$ pour $n = [-5, \dots, +5]$, lorsque $\Omega_c = \pi/4$ et $\Omega_0 = \pi/2$. Dessiner les réponses correspondantes.
5. On s'intéresse au premier filtre $h(n)$ que l'on veut transformer en filtre causal à phase linéaire ayant une réponse impulsionnelle limitée à 11 points sans pondération de la réponse en ces points. Comment peut-on obtenir ce résultat ?
6. En déduire l'expression de la fonction de transfert $H_a(z)$, ainsi que $H_a(e^{j\Omega})$ dont on calculera les coefficients a_n $(a_0 \dots a_5)$ lorsque $\Omega_c = \pi/4$.
7. Quelle est la largeur de la zone de transition de $H_a(e^{j\Omega})$ et l'amplitude maximale de l'ondulation dans la zone atténuée.

12.4.2 Méthode de l'échantillonnage fréquentiel

On désire réaliser un filtre dérivateur à Réponse Impulsionnelle Finie ayant une caractéristique en phase linéaire par la méthode de l'échantillonnage fréquentiel sur N points.
La réponse fréquentielle entre $-\pi$ et π du filtre idéal est donc définie par :

$$H(e^{j\Omega}) = \begin{cases} j\frac{\Omega}{\Omega_c} & pour \quad -\Omega_c \leq \Omega \leq \Omega_c \\ 0 & pour \quad \Omega_c < \Omega \leq \pi \ et \ -\pi \leq \Omega < -\Omega_c \end{cases}$$

On fixe $\Omega_c = \frac{4\pi}{N}$

1. Dessiner le pseudo-module $A(\Omega)$ et la phase $\phi(\Omega)$ de la réponse fréquentielle pour $-2\pi \leq \Omega \leq 2\pi$.
2. Donner le type de réponse impulsionnelle issu de la classification vue en cours pouvant réaliser au mieux ce filtre RIF à phase linéaire.
3. On échantillonne le filtre idéal à $\Omega_e = \Omega_c/2$ pour $0 \leq k\Omega_e < 2\pi$.
   - Représenter la réponse fréquentielle du filtre échantillonné $H_a(k\Omega_e)$.
   - Exprimer la réponse impulsionnelle $h_a(n)$ en fonction de N.
   - Calculez et dessinez $h_a(n)$ pour le cas particulier où $N = 7$.
4. Donner l'expression de l'équation aux différences du filtre. En déduire la fonction de transfert $H_a(z)$ du filtre obtenu.

5. Donner un deuxième version de $H_a(z)$ directement déduite de $H_a(k\Omega_e)$ sous forme de cellules du second ordre en parallèle réelles.
6. Montrer que les deux versions du 4. et du 5. sont équivalentes.
7. A votre avis quels sont les problèmes sur la réponse fréquentielle de ce filtre. Donner une nouvelle version du filtre $h_b(n)$ déduite du filtre $h_a(n)$ dont le comportement en fréquence serait optimisé.

12.5 Transformée de Fourier Discrète et Rapide (TFD et TFR)

12.5.1 TFD bidimensionnele

Soit une transformée de Fourier discrète d'un signal bidimensionnel ; par exemple une image de taille $N \times N$:

$$X(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(k,l) \times W_N^{mk} \times W_N^{nl}$$

avec :

$$W_N = e^{\frac{-2j\pi}{N}}$$

Montrer que cette transformée peut s'exprimer comme étant la succession d'une TF 1D sur les lignes de l'image et une TF 1D sur les colonnes de l'image.

12.5.2 Transformée de Fourier Glissante

On considère une séquence temporelle $x(k)$ que l'on échantillonne continûment dans le temps à une cadence fixe. On désire obtenir en permanence le spectre de cette séquence sur les N derniers points échantillonnés. A l'instant $i+N$, le spectre est obtenu par :

$$X_i(n) = \sum_{k=i}^{i+N-1} x(k).W_N^{n(k-i)},\ pour 0 \leq n \leq N-1$$

1. Donner l'expression du spectre $X_{i+1}(n)$.
2. Comment peut on calculer ce spectre de manière récurrente.
3. Comparer le nombre de calculs à effectuer entre la solution précédente et la solution consistant à calculer une TFD ou une TFR sur chaque séquence.

12.5.3 Transformée de Fourier en Base 4

Démontrer comment on peut obtenir une TFR à base 4 à partir d'une TFD. Sous quelle condition sur N peut y arriver. On utilisera l'exemple sur 16 points pour supporter la démonstration. On précisera les calculs d'un papillon.

12.5.4 Optimisation du calcul de la TFR d'une suite de nombres réels

Soit une séquence temporelle d'échantillons réels $u(k),\ 0 \leq k \leq 2N-1$ dont nous cherchons à calculer le spectre en minimisant le nombre de calculs à effectuer. Pour cela on forme une séquence temporelle complexe $x(n) = x_r(n) + j.x_i(n)$ telle que :

$x_r(n) = u(k),\ k = 2n$, partie réelle de la séquence $x(n)$,

$x_i(n) = u(k),\ k = 2n+1$, partie imaginaire de la séquence $x(n)$,

$X(p) = X_r(p) + j.X_i(p)$ est la TFD de $x(n)$, $U(p) = U_r(p) + j.U_i(p)$ est la TFD de $u(k)$.

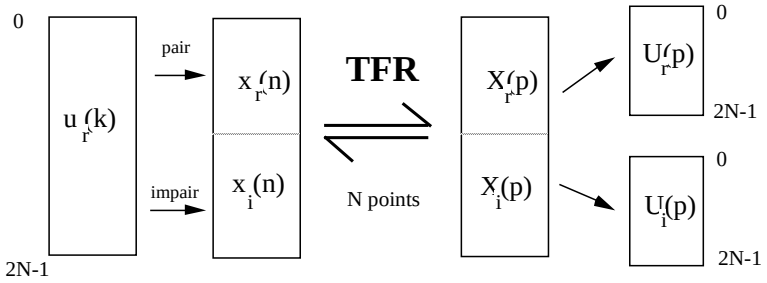


FIG. 12.3: TFR d'une suite de $2N$ nombres réels

1. Trouver $U(p)$ et $X(p)$ en fonction de $x_r(n)$ et $x_i(n)$. Démontrer que :

$$\begin{aligned} X(p) &= A + j.B \\ U(p) &= A + B.e^{-\frac{j\pi p}{N}} \end{aligned}$$

   où A et B sont des nombres complexes. Exprimer $X(N-p)$ en fonction de A et B afin d'en déduire les valeurs de A et B. Donner finalement les relations permettant de retrouver $U(p)$ à partir de $X(p)$ et $X(N-p)$.
2. Évaluer le gain en nombre de calculs ($\oplus$ et $\otimes$) que l'on obtient entre l'application de la méthode précédente et l'application directe de la TFR de $u(k)$.

12.5.5 Optimisation du calcul de la TFR de deux suites de nombres réels

Soit deux séquences temporelles réelles $u(k)$ et $v(l)$ sur N points dont nous cherchons à calculer les TFR en minimisant le nombre de calculs à effectuer. Pour cela on forme une séquence temporelle complexe $x(i)$ telle que :

$x_r(i) = u(k),\ k = i$, partie réelle de la séquence $x(i)$,

$x_i(i) = v(l),\ l = i$, partie imaginaire de la séquence $x(i)$,

$x(i) = x_r(i) + j.x_i(i)$.

1. Donner les relations permettant de retrouver les TFD $U(p)$ et $V(q)$ de $u(k)$ et $v(l)$ à partir du spectre $X(n)$ de $x(i)$.
2. Évaluer le gain en nombre de calculs ($\oplus$ et $\otimes$) que l'on obtient entre l'application de la méthode précédente et l'application directe de la TFR de $u(k)$ et de $v(l)$.

12.5.6 Comparaison entre TFTD et TFD

Soit le signal :

$$x(n) = e^{-a.n} \times u(n)$$

avec $u(n)$ l'échelon unité, la période d'échantillonnage étant $T_e = \frac{1}{F_e}$.

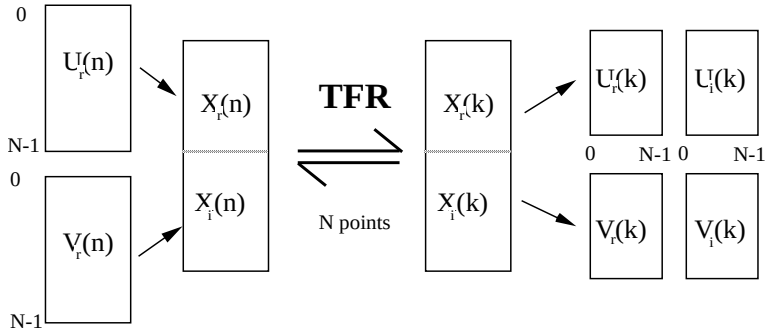


FIG. 12.4: TFR de 2 suites de N nombres réels

– Déterminer la TFTD, Transformée de Fourier à Temps Discret de $x(n)$ que l'on notera $X_{TFTD}(f)$
– Déterminer la TFD, Transformée de Fourier Discrète de $x(n)$ que l'on notera $X_{TFD}(k)$
– Comparer les résultats de la TFTD et de la TFD, d'où peut provenir l'écart entre ces résultats et évaluer son comportement ? (on cherchera à exprimer une relation entre $X_{TFD}(k)$ et $X_{TFTD}(f)$)

12.5.7 TFD par convolution

1. En utilisant la relation $n.k = (n^2 + k^2 - (n-k)^2)/2$, montrez que l'on peut exprimer une TFD à partir d'un convolution.
2. Donner le schéma de principe de la TFD par convolution. Quel avantage peut comporter cette solution ?

12.5.8 Bruits dans la TFD

On rappelle l'expression de la transformée de Fourier Discrète $X(k)$ d'un signal $x(n)$ que l'on supposera réel :

$$X(k) = \sum_{n=0}^{N-1} x(n) \times W_N^{nk}, \quad 0 \leq k \leq N-1, \quad W_N^{nk} = e^{-j\frac{2\pi nk}{N}}$$

1. Évaluer le nombre de multiplications et le nombre d'additions de la TFD.
2. Rappeler le modèle statistique de quantification par arrondi d'un signal sur b bits, on prendra comme application numérique $b = 8$ (préciser l'intervalle de variation de l'erreur, la densité de probabilité, moyenne et variance de ce bruit)
3. On suppose que le signal d'entrée est entâché d'un bruit de puissance σ_e^2, évaluer la puissance de bruit σ_s^2 en sortie de la TFD en fonction de σ_e^2 et de b.
4. Dans quel rapport (exprimé en linéaire ou log) diminue-t-on la puissance de σ_s^2 lorsqu'on multiplie le nombre de bits de représentation par 2.
5. Montrer que chaque sortie d'une TFD, $X(k)$, peut être obtenue à partir des entrées $x(n)$

par une relation de récurrence du type suivant :

$$\begin{aligned} y(m) &= A_k \times y(m-1) + B_m \\ y(0) &= 0 \\ X(k) &= y(N) \end{aligned}$$

6. Cette équation comporte-t-elle des avantages ?

12.5.9 Étude des bruits de calcul dans la transformée de Fourier Rapide

On s'intéresse à l'estimation du bruit de calcul d'une transformée de Fourier rapide, à base 2 et à entrelacement temporel (DIT). Les calculs de cette transformation numérique reposent sur l'enchaînement de papillons, suivant la structure donnée dans les documents.

Le papillon de la TFR est la structure de calcul qui se répète. On étudiera donc les bruits de calcul qui s'y produisent. Pour cela on considère un modèle de graphe où les données sont complexes (figure 12.5 à gauche), ou réelles (figure 12.5 à droite).

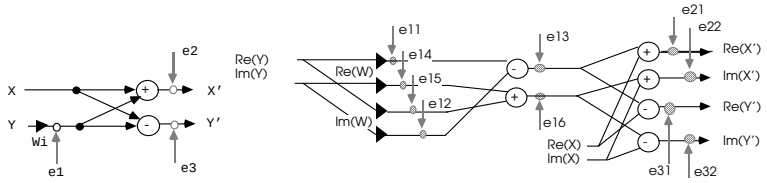


FIG. 12.5: Modèles de bruits de calcul dans un papillon

12.5.9.1 Calcul en virgule fixe cadrée à gauche [-1. . . +1] sur b bits

1. Précisez le bruit de calcul additionnel à la sortie de chaque opération. Calculez le bruit de calcul σ_s^2 sur les données sortant du papillon, en fonction du bruit de calcul et du bruit σ_s^e sur les données complexes en entrée du papillon. On demande de spécifier tous les bruits de calcul en fonction de q, l'échelon de quantification. Exprimez q en fonction de b, le nombre de bits du format de codage.
2. Pour une transformée de Fourier sur N points, le nombre d'étapes de calcul est $log_2(N)$. Si les échantillons sont entachés d'un bruit de conversion σ_o^2, quel est le bruit σ_n^2 sur les sorties de la TFR sur N points (on considère que N est une puissance de 2).
3. Les calculs sont opérés sur un processeur de traitement du signal qui permet de sauvegarder les résultats des multiplications en double précision, sur $2b$ bits. Les additions peuvent être opérées sur $2b$ bits (double précision), mais la mémorisation des données calculées à la sortie de chaque papillon n'est faite que sur b bits (on considère que les calculs intermédiaires du papillon peuvent être mémorisés en double précision). Un arrondi est donc effectué uniquement lors de la mémorisation des résultats d'un papillon. Indiquez à partir du graphe flot de calcul du 2. la valeur des différents bruits de calcul. Calculez comme au 1. σ_s^2 en fonction de σ_e^2, puis comme au 2. σ_n^2 en fonction de σ_o^2.

4. Pour résoudre des problèmes de dynamique de calcul, une division par 2 des données calculées à la sortie des papillons est opérée systématiquement, soit globalement une division par N. On appellera σ^2_{div} la valeur du bruit d'arrondi de la division, on admettra qu'une division complexe entraîne un bruit de puissance $q^2/4$. Préciser sur le graphe flot de calcul du papillon les sources de bruit (on ne considère plus la mémorisation des résultats intermédiaires en double précision).
5. Indiquez pour les conditions précédentes la valeur de σ^2_s en fonction de σ^2_e, puis de σ^2_n en fonction de σ^2_o.

12.5.10 Calculs de TFD

1. Donner la représentation matricielle de la TFD d'un vecteur de N échantillons. Donner en particulier la matrice de transformation de Fourier discrète lorsque $N = 4$. Simplifier au maximum les éléments de la matrice.
2. Soient les signaux $x(n)$ et $h(n)$ suivants :

$$h(n) = \begin{cases} \frac{(n+1)}{10} & \text{si } n = 0 \cdots 3 \\ 0 & \text{sinon} \end{cases} \qquad x(n) = \begin{cases} 0 & \text{si } n = 4k \\ 1 & \text{si } n = 4k+1, 4k+3 \\ 2 & \text{si } n = 4k+2 \end{cases}$$

   $x(n)$ est un signal périodique. Calculer $X(k)$ et $H(k)$ les TFD sur 4 points des signaux $x(n)$ et $h(n)$. Tracer le module.
3. Comparer qualitativement $H(k)$ et $X(k)$ avec les transformées de Fourier des signaux $x(n)$ et $h(n)$.
4. Exprimer $y(n)$, résultat du filtrage de $x(n)$ par un filtre de réponse impulsionnelle $h(n)$.
5. Expliquer comment obtenir $Y(k)$, la représentation spectrale du signal $y(n)$.

12.5.11 Transformée en cosinus discret rapide

On s'intéresse à l'estimation du bruit de calcul d'une transformée en cosinus rapide (TCR). La TCR prend en entrée un vecteur de signal réel et fournit un vecteur réel de même dimension. Les calculs de cette transformation reposent sur l'enchaînement de papillons dont la structure est donnée figure 12.6. Le calcul d'un papillon élémentaire est représenté figure 12.6. On a, pour chaque papillon, $x_s = x_e + c_k.y_e$ et $y_s = x_e - c_k.y_e$. L'indice k des coefficients C_k varie pour chaque papillon et $c_k = \cos(\frac{2\pi k}{N})$.

12.5.11.1 Complexité de calcul

1. Quelle est la complexité en nombre de multiplications et d'additions ainsi qu'en nombre de mots mémoires pour une TCR sur N points.
2. On veut effectuer cette transformée en continu sur le signal, par bloc de N échantillons, sans recouvrement. Sur une machine réalisant une multiplication ou une addition en un cycle de $50ns$, quelle est la taille maximale du bloc que l'on peut traiter si le signal est échantillonné à 1MHz.

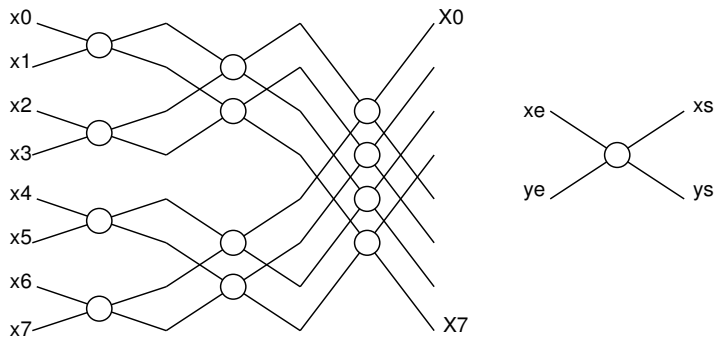


FIG. 12.6: Structure d'une TCR

12.5.11.2 Implantation en virgule fixe cadrée à gauche

On travaille avec des nombres sur b bits utiles, en virgule fixe centrée à gauche de dynamique $[-1, 1]$ et en complément à deux.

1. Dessiner le graphe flot de calcul d'un papillon et préciser le bruit de calcul additionnel à la sortie de chaque opération.

2. Calculer le bruit de calcul b_s sur les données x_s et y_s sortant du papillon en fonction du bruit de calcul et du bruit b_e sur les données x_e et y_e en entrée du papillon et des C_k. En déduire la puissance du bruit en sortie σ_s^2 en fonction de la puissance du bruit en entrée.

3. Pour une TCR sur N points, le nombre d'étapes de calcul est $\log_2(N)$. Si les échantillons sont modifiés par un bruit de quantification, quelle est la puissance du bruit σ_{fi}^2 sur les N sorties X_i de la TCR. On considérera que $|C_k| < 1$ et que le bruit en entrée de la TCR sur les x_i est équivalent à un bruit de conversion AN.

4. Les calculs sont réalisés sur un DSP qui permet de sauvegarder les résultats sur $2b$ bits de précision. Les additions peuvent être menées sur $2b$ bits mais la mémorisation des données x_s et y_s calculées à la sortie de chaque papillon n'est faite que sur b bits. Un arrondi est effectué uniquement lors de la mémorisation des résultats d'un papillon. Indiquer à partir du graphe flot précédent la présence des différents bruits. Calculer comme précédemment, σ_s^2 et σ_{fi}^2.

5. Évaluer les problèmes de débordement dans un papillon élémentaire, puis dans la TCR complète.

6. Pour résoudre les problèmes de dynamique de calcul, une division par 2 des données calculées à la sortie de chaque papillon est opérée systématiquement, soit globalement une division par N. On admettra qu'une division par 2 entraîne un bruit de puissance $\frac{q^2}{4}$. Préciser sur le graphe flot de calcul d'un papillon les sources de bruit (on ne considère plus la mémorisation des résultats intermédiaires en double précision). Déterminer comme précédemment σ_s^2 et σ_{fi}^2.

12.6 Analyse spectrale

12.6.1 Questions

1. Un signal analogique est échantillonné à $F_e = 10$ kHz. On calcule son spectre à partir de 1024 points de la séquence temporelle. Quel est l'intervalle de fréquence entre deux points successifs du spectre ?
2. On rappellera le schéma de principe d'une analyse spectrale. La bande occupée par un signal à analyser s'étend de 0 à 10 kHz. La résolution fréquentielle recherchée est de 1 Hz. La résolution en amplitude doit être supérieure à 40 dB.
   (a) Quelle doit être la fréquence d'échantillonnage.
   (b) Quelle longueur d'enregistrement doit on prélever pour faire une telle analyse.
   (c) Déterminez les caractéristiques d'une machine 8 bits capable de réaliser une telle analyse spectrale (capacité de mémoire, temps d'addition et de multiplication). On désire que le résultat de l'analyse soit affiché sur un écran à une fréquence de 25 images/s.

12.6.2 Analyse spectrale d'un signal sinusoïdal

On effectue l'analyse spectrale par voie numérique d'un signal sinusoïdal de fréquence f_s. On sait que l'observation du signal temporel durant un temps limité à $[0,\ N.T]$ amène à une pondération du signal temporel par une fenêtre d'observation. On se propose d'étudier l'effet sur le spectre du signal observé.

1. Dans le cas d'une fenêtre rectangulaire, tracer le spectre du signal continu x(t), puis celui du signal discrétisé, tronqué et pondéré $x_{T_0}(n.T)$. Calculer l'erreur maximale en % que l'on fait sur l'estimation de l'amplitude du spectre lorsque $f_0 \neq N/T_0$.
2. Calculer le TF d'une fenêtre triangulaire entre 0 et T_0, valant 1 en $T_0/2$. Reprendre dans le cas d'une fenêtre triangulaire (Bartlett) la question 1.
3. Même question que le 1. lorsque la fenêtre d'observation est une fenêtre de Hanning.
4. On veut une résolution fréquentielle de 1 Hz entre deux raies du spectre, avec des amplitudes pouvant varier de 1 à 10. Calculer T0 dans les cas où on utilise une des trois fenêtres précédentes.

12.6.3 Analyse spectrale d'un signal

1. Calculer la TFD $X(n)$ de la suite $x(k) = sin\left[2\pi\frac{k}{3.5}\right]$ avec $0 \leq k \leq 15$. Quel type de fenêtre est implicitement utilisé pour l'analyse de $x(k)$?
2. Tracer $X(k)$ et la TF du signal sinusoïdal complet sur le même graphique. Calculez, pour cet exemple, les erreurs d'analyse (en fréquence et en amplitude) induites par le fenêtrage. Ceci peut être fait même si la TFD n'est pas calculée.

12.7 Convolution

12.7.1 Calcul d'une convolution

1. Soit $x(n) = a^n.u(n)$ et $h(n) = b^n.u(n)$, trouver par la méthode directe $y(n) = x(n) * h(n)$. $*$ est la convolution de 2 signaux.

2. Retrouver ce résultat par l'application de la transformée en Z de la convolution.
3. On tronque $x(n)$ sur $N = 8$ points et $h(n)$ sur $M = 4$ points. Donner l'expression de $y(n)$ dans ce cas. On donnera les valeurs et on dessinera $y(n)$ pour $n = 0 \dots 15$.

12.7.2 Complexité de calcul d'une convolution

1. Donner la complexité de calcul de la convolution par la méthode directe.
2. Donner la complexité de calcul de la convolution par la méthode rapide en utilisant la Transformée de Fourier Rapide.

12.8 Interpolation et décimation

12.8.1 Interpolation linéaire

On considère l'interpolation d'ordre 1 d'une fonction $x(t)$: $x(t+\epsilon) = x(t) + \epsilon.\frac{dx(t)}{dt}$, avec ϵ petit. On peut appliquer cette formule à une séquence $x(k)$ échantillonnée :

$$x(k+\epsilon) = x(k).(1-\epsilon) + \epsilon.x(k+1)$$

On appelle $X(n)$ la TFD de cette séquence. On appelle $Y(p)$, $0 \leq p \leq 2N-1$, la TFD de la séquence $y(l)$, où $y(l)$ est égale à la séquence $x(k)$ interpolée avec $\epsilon = 1/2$. Si l est pair, $y(l) = x(k)$ $(k = l/2)$ est l'échantillon contenu primitivement dans la séquence temporelle, si l est impair $y(l)$ est un échantillon interpolé.

1. Montrer que le spectre Y(p) peut s'écrire :

$$\begin{aligned} Y(p) &= X(p).[1 + cos(\pi p/N)],\ 0 \leq p \leq N-1 \\ Y(p) &= X(p).[1 - cos(\pi p/N)],\ N \leq p \leq 2N-1 \end{aligned}$$

   On considérera les conditions initiales $x(0) = x(N) = 0$.
2. Indiquer comment on peut effectuer une interpolation linéaire en utilisant la TFR.

12.8.2 Suréchantillonnage

On considère la séquence temporelle $x_N(k)$ sur N points et $X_N(n)$ son spectre. On complète $x(k)$ par des zéros pour obtenir une séquence sur M points $(M > N)$. Cette nouvelle suite $x_M(k)$ à un spectre $X_M(n)$.

1. Qu'il y a t-il de changé au niveau spectral ?
2. Déterminez la relation entre M et N pour que toutes les composantes de $X_N(n)$ soient contenues dans $X_M(n)$.

Chapitre 13

Corrections des Travaux Dirigés en TNS

13.1 Corrigés des TD sur l'échantillonnage

13.1.1 Chaîne de TNS

Soit le signal : $x(t) = A \cdot e^{-at} \cdot sin(\omega_0.t) \cdot u(t)$

$$\begin{aligned} X(f) &= \int_{-\infty}^{+\infty} x(t) \cdot e^{-2j\pi ft} dt = A \cdot \int_{0}^{+\infty} e^{-at} \cdot sin(\omega_0.t) \cdot e^{-2j\pi ft} dt \\ X(\omega) &= \frac{A\omega_0}{(a+j\omega)^2 + \omega_0^2} \\ y(n) &= a^2.y(n-2) + a.x(n-1) + x(n) \end{aligned}$$

$$\begin{aligned} y(n) &= a^3.y(n-3) + a^2.x(n-2) + a.x(n-1) + x(n) \\ y(n) &= a^N.y(n-N) + D(n,N) \\ D(n,N) &= \sum_{j=0}^{N-1} a^j.x(n-j) \end{aligned}$$

1. $X(f) = \frac{A.\omega_0}{a^2+\omega_0^2-\omega^2+2.a.j.\omega}$
2. $\omega_{MAX} = 24.14 rad/s;\ |X(0)| = 0.73;\ |X(\omega_{MAX})| = A/2a = 1,43$
3. $|X(F_m)|^2 = 10^{-4}|X(F_{max})|^2 \Rightarrow F_m = 30Hz$
4. $|X(F_r)| = 0.01|X(F_{max})| = 0.0001|X(F_m)| \Rightarrow F_r = 298Hz$

13.1.2 Échantillonnage d'un signal

1. $X(f) = \frac{1}{a+j2\pi f}$
2. $E_x = 1/(2a)$
3. $X_e(\Omega) = \frac{1}{1-e^{-aT}e^{-j\Omega}}$

4. Problème de recouvrement de spectre.
5. $E_{[-B..+B]} = \int_{-B}^{B} |X(f)|^2 df = \frac{\arctan(2\pi B/a)}{\pi a}$, pour 90% : $B = a$.

13.2 Analyse des filtres numériques

13.2.1 Cellule élémentaire du premier ordre RII

13.2.2 Cellule du second ordre RII purement récursive

13.2.3 Analyse d'un filtre numérique RIF

13.2.3.1 Étude de la réponse fréquentielle

Un retard d'une période T_e du signal analogique correspond à un décalage de 1 de l'indice temporel du signal numérique. On obtient comme équation aux différences finies la relation suivante :

$$y(n) = 0.1x(n) - 0.3x(n-2) + 0.5x(n-3) - 0.3x(n-4) + 0.1x(n-6)$$

En appliquant la relation $TZ(x(n_k)) = z^{-k}TZ(x(n))$, on prend la transformée de l'équation aux différences :

$$Y(z) = 0.1X(z) - 0.3z^{-2}X(z) + 0.5z^{-3}X(z) - 0.3z^{-4}X(z) + 0.1z^{-6}X(z)$$

On obtient alors la fonction de transfert :

$$H(z) = \frac{Y(z)}{X(z)} = 0.1 - 0.3z^{-2} + 0.5z^{-3} - 0.3z^{-4} + 0.1z^{-6}$$

A partir d'une transformée en Z décrite comme la somme de monômes en z^{-1}, la réponse impulsionnelle du système correspond simplement aux coefficients de chacun des monômes.

$$h(n) = 0.1\delta(n) - 0.3\delta(n-2) + 0.5\delta(n-3) - 0.3\delta(n-4) + 0.1\delta(n-6)$$

La réponse fréquentielle du filtre se calcule en évaluant $H(z)$ sur le cercle unité, donc pour $z = e^{j\Omega}$. On obtient :

$$\begin{aligned} H(e^{j\Omega}) &= 0.1e^{j0\Omega} - 0.3e^{-j2\Omega} + 0.5e^{-j3\Omega} - 0.3e^{-j4\Omega} + 0.1e^{-j6\Omega} \\ &= 0.1[e^{j0\Omega} + e^{-j6\Omega}] - 0.3[e^{-j2\Omega} + e^{-j4\Omega}] + 0.5e^{-j3\Omega} \\ &= 2 \times (0.1e^{-j3\Omega}\cos(3\Omega) - 0.3e^{-j3\Omega}\cos(\Omega) + 0.5e^{-j3\Omega}) \\ &= 2 \times e^{-j3\Omega}[0.5 - 0.3\cos(\Omega) + 0.1\cos(3\Omega)] \end{aligned}$$

Puisque seul le terme mis en facteur est complexe, le module et la phase du filtre viennent facilement :

$$\begin{aligned} |H(e^{j\Omega})| &= 1.0 - 0.6\cos(\Omega) + 0.2\cos(3\Omega) \\ Arg(H(e^{j\Omega})) &= -3\Omega \end{aligned}$$

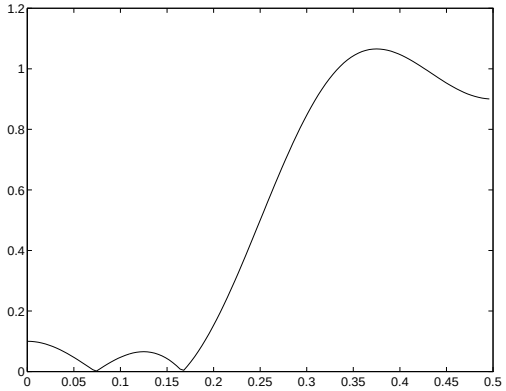


FIG. 13.1: Module de la réponse fréquentielle du filtre étudié, en abscisse la fréquence normalisée et en ordonnée le module sur une échelle linéaire

La figure 13.1 représente le module de la réponse fréquentielle du filtre avec en abscisse une fréquence normalisée $f \in [0, \frac{1}{2}]$ et en ordonnée le module sur une échelle linéaire. La figure 13.2 représente en trait plein la phase modulo π et en trait pointillé la phase déroulée.
On constate qu'il s'agit d'un filtre de type passe-haut ; notons de plus que ce filtre est à phase linéaire.

13.2.3.2 Complexité de l'implantation du filtre sur un DSP

Le graphe de traitement montre que pour un échantillon d'entrée, il faut 5 multiplications et 4 additions pour calculer un échantillon de sortie. Pour N échantillons en entrée, la complexité est donc de $5N$ multiplications et de $4N$ additions.
Concernant le nombre de mots mémoires nécessaires au calcul du filtre, il faut tout d'abord 6 registres pour stocker les retards de l'entrée et 1 autre pour stocker les calculs intermédiaires pour les additions. Enfin, il faut compter 5 points mémoire pour stocker les 5 coefficients du filtre.
On dispose d'un processeur de type DSP traitant en parallèle multiplications et additions, il faut donc 5 temps de cycles (le nombre max entre les multiplications et les additions) pour calculer un échantillon de sortie. Puisque les échantillons de sortie sont échantillonnés *à la même fréquence* que l'entrée, on dispose d'un temps min d'échantillonnage $T_{min} = 5 \times 100ns$, c'est-à-dire une fréquence d'échantillonnage maximum de 2MHz.
L'opération réalisée dans le domaine temporel par ce filtre est une convolution linéaire entre les signaux $x(n)$ et $h(n)$. La convolution de ces deux signaux correspond à une multiplication de leurs transformées de Fourier respectives. Un autre schéma de calcul consiste donc à prendre les transformées de Fourier de $x(n)$ et de $h(n)$, respectivement $X(f)$ et $H(f)$, de multiplier ces deux transformées $Y(f) = H(f)X(f)$ et finalement d'effectuer une transformée de Fourier inverse pour obtenir le signal filtré $y(n)$. La transformation de Fourier est ici une Transformée de Fourier Discrète (temps et fréquence sont des variables discrètes) mise en oeuvre sous la forme d'une FFT. On notera qu'il suffit de ne calculer qu'une seule fois $H(f)$, les caractéristiques du filtre étant supposées constantes quel que soit le temps.

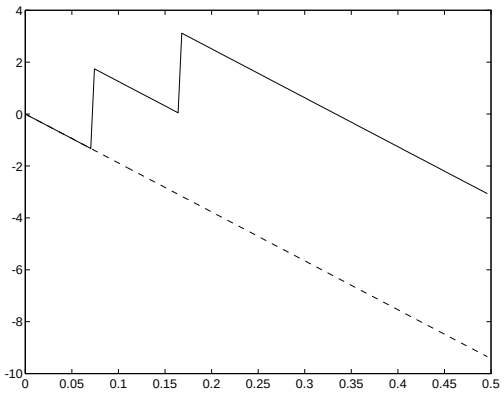


FIG. 13.2: Phase de la réponse fréquentielle du filtre étudié. En abscisse, la fréquence normalisée et en ordonnée la phase en radian. En trait plein, la phase modulo π , en trait pointillé la phase déroulée

La complexité de calcul est de 2 fois celle d'une FFT sur N points plus la multiplication des deux transformées (nombre complexes...). Ce qui fait : $2\times 4\times \frac{N}{2}\log(N)+4\times N$ multiplications réelles et $2\times 3\times N\log(N)+2\times N$ additions réelles. Pour trouver ces relations, revoir dans le cours le coût de calcul d'un papillon élémentaire multiplié par le nombre de papillons mis en oeuvre dans la FFT d'ordre N.

Il ne faut pas oublier que la méthode fréquentielle *ne peut être* exacte puisque que l'on traite des blocs de N points dans le domaine fréquentiel, cela revient dans notre cas à fenêtrer le signal par une fenêtre rectangulaire de longueur N. Ce qui conduit à un spectre calculé correspondant à la convolution de $X(f)$ par $Rect(f)$...ce qui pour faire simple n'est pas $X(f)$!

13.2.3.3 Implantation du filtre en virgule fixe

Les hypothèses sont les suivantes :
- les nombres sont représentés sur b bits utiles.
- le codage est de type virgule fixe cadrée à gauche.
- la dynamique est $[-1, 1]$.
- les nombres négatifs sont codés en complément à deux.
- Le mode de calcul est de type arrondi.

On adopte un modèle statistique de représentation de l'influence des erreurs de calcul sur un signal. On considère que pour chaque traitement faisant intervenir une erreur de calcul on ajoute un bruit au signal traité. Cette source de bruit additive est relativement simpliste mais facilite le calcul des puissances de bruit aux entrées et sorties des systèmes de traitement numérique. Les hypothèses sur les sources de bruit sont les suivantes :
- pour un mode de calcul par arrondi, il s'agit d'un bruit de moyenne $\mu = 0$ et de puissance $\sigma^2 = \frac{q^2}{12}$ avec $q = 2^{-b}$
- bruit blanc non corrélé avec les signaux traités et les autres sources de bruit (les puissances s'ajoutent...)

On placera sur le schéma du filtre autant de sources de bruit qu'il y a de *multiplications* car les additions en format virgule fixe tel que précisé ne génère pas d'erreurs d'arrondi. Soit σ_s^2 la puissance de bruit en sortie du filtre, on a donc :

$$\sigma_s^2 = 5 \times \frac{q^2}{12}$$

On considère maintenant que le signal d'entrée est affecté par une opération de quantification, on lui ajoute donc une source de bruit de puissance σ_e^2 [cette source est située à l'entrée du filtre, elle se superpose à $x(n)$]. On a en sortie le bruit de l'entrée *mis en forme* par le filtre (il s'agit du filtrage s'un signal aléatoire...) superposé au bruit dû aux 5 multiplications. On obtient donc ici :

$$\sigma_s^2 = \sigma_e^2[\Sigma_{k=0}^{6}|h(k)|^2] + 5\frac{q^2}{12}$$

Ce qui fait :

$$\sigma_s^2 = (0.45 + 5)\frac{q^2}{12}$$

On constate que numériquement l'influence du bruit d'entrée sur la sortie est négligeable devant le bruit des 5 multiplications.
On suppose maintenant un signal sinusoïdal en entrée du filtre d'amplitude 1 et de pulsation ω. La puissance de ce signal s'écrit :

$$\begin{aligned} P_x &= \frac{1}{T}\int_0^T \sin^2(\omega t)dt \\ &= \frac{1}{T}\int_0^T \frac{1-\cos(2\omega t)}{2}dt \\ &= \frac{1}{2} \end{aligned}$$

Un rapport signal à bruit est le rapport de la puissance du signal sur celle du bruit. En entrée du filtre, on a le rapport signal/bruit :

$$\begin{aligned} RSB_e &= \frac{P_x}{\sigma_e^2} \\ &= \frac{1/2}{q^2/12} \\ &= \frac{6}{q^2} \end{aligned}$$

En sortie du filtre le RSB devient :

$$\begin{aligned} RSB_s &= \frac{1}{2}\frac{12}{5.45q^2} \\ &= \frac{1.1}{q^2} \end{aligned}$$

Si on exprime le RSB de sortie en log, on obtient :

$$RSB_{dB} = 10\log(1.1) + 2b\log(2) = 0.41 + 6.02b$$

On recherche un nombre de bits suffisant pour que le RSB soit supérieur à $40dB$, en prenant la relation précédente on trouve $b > 6.57$, donc $b > 7$. On notera que l'on travaille avec b bits *utiles*, il faut donc rajouter 1 bit de signe, ce qui fait des mots de 8 bits pour représenter les calculs.
Travailler en virgule fixe peut conduire à des débordements lors de l'enchaînement des additions (ce qui est particulièrement sensible dans le cas d'un filtre). Le signal d'entrée est supposé de dynamique comprise dans $[-1, 1]$ donc $|x(n)| < 1 \quad \forall \quad n$. Une condition suffisante pour éviter un débordement en sortie est de contraindre chaque échantillon dans la même dynamique, soit $|y(n)| < 1 \quad \forall \quad n$. on a alors :

$$|y(n)| \quad \leq \quad x(n)\Sigma_i|h(i)| \quad < \quad 1$$

on pose $x'(n) = Ax(n)$ le facteur d'échelle à appliquer à l'entrée pour éviter le débordement, on obtient alors :

$$A < \frac{1}{\Sigma_i|h(i)|}$$

On trouve $A < 0.77$, pour simplifier et tomber sur une opération binaire, on prendra une division par 2 du signal d'entrée.

13.2.4 Filtrage numérique RIF (1)

1. $h(n) = 0.1[\delta(n-1) + \delta(n-3)] + 0.2.\delta(n-2)$: RIF symétrique
2. $H(\Omega) = 0.2.e^{-2j\Omega}[1 + cos\Omega]$
   $|H(\Omega)| = 0.2(1 + cos\Omega)$
   $Arg[H(\Omega)] = -2\Omega \Rightarrow$ phase linéaire
   $f_c = 182Hz$
3. Filtre passe bas
4. $y(0) = y(5) = 0;\ y(1) = y(4) = 0.1;\ y(2) = y(3) = 0.3$

13.2.5 Filtrage numérique RIF (2)

A partir de la réponse impulsionnelle d'un filtre exprimée comme une somme d'impulsions, on trouve aisément en fonction d'un signal d'entrée la sortie suivante :

$$\begin{aligned} y(n) &= x(n) * h(n) \\ &= a_0x(n) + a_1x(n-1) + a_2x(n-2) + a_1x(n-3) + a_0x(n-4) \end{aligned}$$

On trouve la fonction de transfert :

$$H(z) = a_0 + a_1z^{-1} + a_2z^{-2} + a_1z^{-3} + a_0z^{-4}$$

Pour trouver la réponse fréquentielle du filtre on évalue la transformée en z sur le cercle unité :

$$\begin{aligned} H(e^{j\Omega}) &= a_0 + a_1e^{-j\Omega} + a_2e^{-2j\Omega} + a_1e^{-3j\Omega} + a_0e^{-4j\Omega} \\ &= 2a_0e^{-2j\Omega}\cos(2\Omega) + 2a_1e^{-2j\Omega}\cos(\Omega) + a_2e^{-2j\Omega} \\ &= e^{-2j\Omega}[a_2 + 2a_1\cos(\Omega) + 2a_0\cos(2\Omega)] \end{aligned}$$

On trouve alors les modules et arguments suivants :

$$\begin{aligned} |H(e^{j\Omega})| &= a_2 + 2a_1\cos(\Omega) + 2a_0\cos(2\Omega) \\ \angle H(e^{j\Omega}) &= -2\Omega \end{aligned}$$

On trouve les modules et les phases pour les quelques valeurs de Ω suivantes :

	0	$\pi/2$	π	2π
$\lvert H(e^{j\Omega})\rvert$	$\lvert a_2 + 2a_1 + 2a_0\rvert$	$\lvert a_2 - 2a_0\rvert$	$\lvert a_2 + 2a_0 - 2a_1\rvert$	$\lvert a_2 + 2a_1 + 2a_0\rvert$
$\angle H(e^{j\Omega})$	0	$-\pi$	-2π	-4π

On peut tout d'abord noter que le module en 0 est toujours supérieur au module en π puisque les coefficients sont positifs. Il n'est donc pas possible de réaliser un filtre passe-haut avec une telle équation.
On recherche a_0, a_1, a_2 tels que $|H(e^{j0})| = 1$, $|H(e^{j\frac{\pi}{2}})| = \frac{1}{2}$ et $|H(e^{j\pi})| = 0$. Il faut tout d'abord noter que l'on travaille avec une norme L_2, c'est-à-dire des valeurs absolues pour des nombres réels. On cherche tout d'abord à faire disparaître les valeurs absolues en distinguant éventuellement plusieurs solutions selon les coefficients a_i. Pour cela on dispose d'une contrainte supplémentaire : les a_i sont supérieurs ou égaux à 0. On a donc $\forall a_i \quad |H(e^{j0})| \geq |H(e^{j\pi})|$, c'est-à-dire $\forall a_i \quad |H(e^{j0})| \geq 0$. On trouve alors comme première équation $(a_2 + 2a_1 + 2a_0) = 1$, comme secondes $(a_2 - 2a_0) = \frac{1}{2}$ ou $-(a_2 - 2a_0) = \frac{1}{2}$, et comme troisième $(a_2 + 2a_0 - 2a_1) = 0$. On distingue deux solutions donnant chacune un jeu de coefficients différents.

13.2.5.1 Première solution

On a le système suivant :

$$\begin{aligned} a_2 + 2a_1 + 2a_0 &= 1 \\ a_2 - 2a_0 &= \frac{1}{2} \\ a_2 + 2a_0 - 2a_1 &= 0 \end{aligned}$$

On trouve $a_0 = 0$, $a_1 = \frac{1}{4}$, $a_2 = \frac{1}{2}$.

13.2.5.2 Deuxième solution

$$\begin{aligned} a_2 + 2a_1 + 2a_0 &= 1 \\ a_2 - 2a_0 &= -\frac{1}{2} \\ a_2 + 2a_0 - 2a_1 &= 0 \end{aligned}$$

On trouve $a_0 = \frac{1}{4}$, $a_1 = \frac{1}{4}$ et $a_2 = 0$.
Une fréquence de coupure à $-3dB$ correspond à trouver la pulsation Ω_c telle que

$$\frac{|H(e^{j\Omega_c}|^2}{|H(e^{j\Omega_{max}})|^2_{max}} = \frac{1}{2}$$

En prenant les coefficients de la première solution, on obtient :

$$H(e^{j\Omega}) = e^{-j2\Omega}[\frac{1}{2} + \frac{1}{2}\cos(\Omega)]$$

On cherche Ω_c vérifiant :

$$0.5 + 0.5cos(\Omega_c) = \frac{1}{\sqrt{2}}$$

On trouve donc $\cos(\Omega) = \sqrt{2} - 1$, soit $\Omega = 65.5° = \frac{2\pi}{5.5}$. On les relations $\frac{\Omega}{T_e} = 2\pi f$, donc :

$$F_c = \frac{F_e}{5.5} = 7273Hz$$

Il n'existe pas de solution qui respecte le théorème de Shannon en utilisant l'autre jeu de coefficients.

13.2.6 Filtrage Numérique RIF cascade

13.2.6.1 Fonctions de transfert

L'équation au différences finies s'écrit :

$$y(n) = b_0^i x(n) + b_1^i x(n-1) + b_2^i x(n-2)$$

La figure 13.3 présente la structure de réalisation du filtre sous une forme canonique.

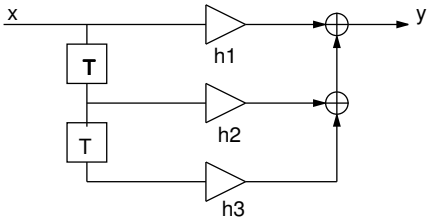


FIG. 13.3: Structure d'implantation d'un filtre non récursif de longueur 2

La mise en parallèle des trois filtres conduit à la fonction de transfert suivante :

$$M(z) = H^0(z) + H^1(z) + H^2(z)$$

En posant $M(z) = m_0 + m_1 z^{-1} + m_2 z^{-2}$, on a :

$$m_j = \sum_{i=0}^{2} b_j^i$$

La mise en cascade des trois filtres conduit à la fonction de transfert suivante :

$$N(z) = H^0(z) \times H^1(z) \times H^2(z)$$

En posant $N(z) = n_0 + n_1 z^{-1} + n_2 z^{-2}$, on a :

$$n_0 = \prod_{i=0}^{2} b_0^i \cdots$$

On s'intéressera dans la suite du problème à l'étude de $N(z)$.

13.2.6.2 Complexité

Il faut 3 multiplications, 2 additions et 2/3 mots mémoires pour un filtre élémentaire. Pour le filtre $N(z)$ il faut donc 9 multiplications et 6 additions.
On considère que les multiplications et les additions sont réalisées en parallèle, le temps de traitement est donc $9 \times T_c$ avec T_c le temps de cycle d'une opération élémentaire. Avec une fréquence d'échantillonnage de $F_e = 44.1kHz$, on a $T_e = 22.67\mu s$. Il faut donc mener tous les calculs en un temps maximum T_e, on alors :

$$9 \times T_c \leq T_e \quad T_c \leq 2.5\mu s$$

13.2.6.3 Étude du bruit de calcul

On dispose d'un filtre élémentaire mettant en œuvre 3 multiplications, nous aurons donc trois sources de bruit interne. Un bruit de quantification se superpose au signal d'entrée. Si l'influence des coefficients multiplicatifs est négligeable, le modèle de bruit du filtre comporte 6 sources élémentaires de variances $\frac{q^2}{12}$, 3 sources en entrée des branches en parallèle (e_e^i)et 3 autres pour les multiplications ($e_m^i(n)$), on a donc :

$$\begin{aligned}
\sigma_s^2 &= 3 \times \sigma_e^2 + 3 \times \sigma_m^2 \\
\sigma_s^2 &= 3 \times \frac{q^2}{12} + 3 \times \frac{q^2}{12} \\
\sigma_s^2 &= \frac{q^2}{2}
\end{aligned}$$

Si maintenant les coefficients multiplicatifs sont significatifs, on a :

$$\begin{aligned}
\sigma_s^2 &= 3 \times \sigma_e^2 + 3 \times \sigma_m^2 \\
\sigma_s^2 &= \frac{q^2}{12} \times (\sum_{j=0}^{2} (b_j^i)^2) + \frac{q^2}{4}
\end{aligned}$$

On enchaîne maintenant en cascade les trois filtres élémentaires pour constituer le filtre $N(z)$. On considère un bruit de quantification en entrée du système et on pose σ_1^2 la puissance du bruit à la sortie du premier filtre, σ_2^2 celle à la sortie du second et σ_N^2 celle en bout du filtre cascade (dernier étage). On a :

$$\begin{aligned}
\sigma_1^2 &= \frac{q^2}{4} + \frac{q^2}{12} \sum_{j=0}^{2} (b_j^0)^2 \\
\sigma_2^2 &= \frac{q^2}{4} + \sigma_1^2 \sum_{j=0}^{2} (b_j^1)^2 \\
\sigma_N^2 &= \frac{q^2}{4} + \sigma_2^2 \sum_{j=0}^{2} (b_j^2)^2 \\
&= \frac{q^2}{4}[1 + \sum_{j=0}^{2} (b_j^2)^2 + \sum_{j=0}^{2} (b_j^2)^2 \sum_{j=0}^{2} (b_j^1)^2] + \frac{q^2}{12} \sum_{j=0}^{2} (b_j^2)^2 \sum_{j=0}^{2} (b_j^1)^2 \sum_{j=0}^{2} (b_j^0)^2
\end{aligned}$$

Il faut donc mettre le filtre qui coupe le moins (coefficients les plus faible) en fin de chaîne.

Une sortie maximale y_{max} d'un filtre élémentaire peut s'écrire de la manière suivante en fonction d'une entrée maximale :

$$y_{max} = x_{max} \sum_{j=0}^{2} |b_j^i|$$

pour éviter tout débordement, on impose $y_{max} < 1$, donc :

$$x_{max} \leq \frac{1}{\sum_{j=0}^{2} |b_j^i|}$$

Si on considère maintenant le filtre cascade $N(z)$, on a :

$$x_{max} \leq \frac{1}{(\sum_{j=0}^{2} |b_j^0|)(\sum_{j=0}^{2} |b_j^1|)(\sum_{j=0}^{2} |b_j^2|)}$$

13.2.6.4 Application Numérique

En prenant :

$$b_0^i = 0.5, \quad b_1^i = 0.75, \quad b_2^i = 0.5, \forall i$$

On trouve la réponse impulsionnelle suivante :

$$\begin{aligned} N(n) &= 0.125\delta(n) + 0.5625\delta(n-1) + 1.21875\delta(n-2) + 1.546875\delta(n-3) \\ &+ 1.21875\delta(n-4) + 0.5625\delta(n-5) + 0.125\delta(n-6) \end{aligned}$$

On a la réponse fréquentielle suivante :

$$N(e^j\Omega) = e^{-3j\Omega}(0.75 + cos(\Omega))^3$$

On a $\sum b_j^2 = 1.0626$, on trouve donc :

$$\sigma_N^2 = \frac{q^2}{4} + 1.0625 \times (\frac{q^2}{4} + 1.0625 \times (\frac{q^2}{4} + 1.0625 \times \frac{q^2}{12}))$$

ce qui fait $\sigma_N^2 \approx 0.9 \times q^2$.
On a :

$$x_{max} \leq \frac{1}{3 \times (0.5 + 0.75 + 0.5)}$$

ce qui donne $x_{max} \leq 0.187$. En fait en prenant $x_{max} = \frac{1}{6}$ on évite les débordements. Il suffit alors de diviser par 2 le signal en entrée de chaque filtre élémentaire.

13.4.2 Méthode de l'échantillonnage fréquentiel

On désire réaliser un filtre dérivateur à Réponse Impulsionnelle Finie ayant une caractéristique en phase linéaire par la méthode de l'échantillonnage fréquentiel sur N points.
On fixe $\Omega_c = \frac{4\pi}{N}$

1. $A(\Omega) = \frac{\Omega}{\Omega_c}$ entre $-\Omega_c$ et Ω_c. La phase $\phi(\Omega)$ est constante entre $-\Omega_c$ et Ω_c et vaut $\pi/2$ (voir figure 13.4).

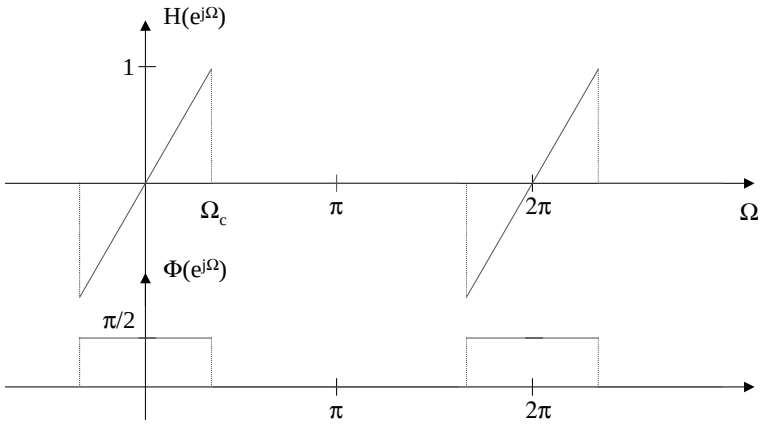


FIG. 13.4: Pseudo module et phase du dérivateur

2. Le type de réponse permettant de réaliser au mieux ce filtre RIF à phase linéaire est le type III (réponse impulsionnelle antisymétrique et N impair). Cela implique deux zéros en $\Omega = 0$ et $\Omega = \pi$.

3. On échantillonne le filtre idéal à $\Omega_e = \Omega_c/2$ pour $0 \leq k\Omega_e < 2\pi$.
   – La réponse fréquentielle du filtre échantillonné $H_a(k\Omega_e)$ est donnée figure 13.4.

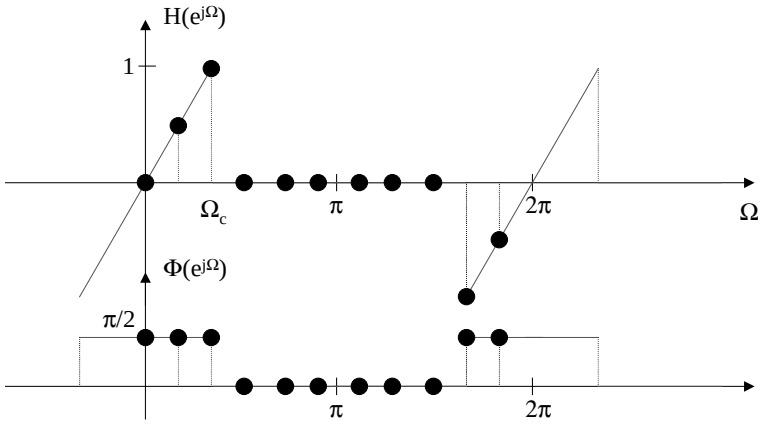


FIG. 13.5: Pseudo module et phase du dérivateur échantillonné

– Calcul de $h_a(n)$ pour $n = 0 \dots N-1$:

$$\begin{aligned}
h_a(n) &= \frac{1}{N}\sum_{k=0}^{N-1} H(k\Omega_e)e^{j2\pi k.n/N} \\
h_a(n) &= j\frac{1}{N}[0.5e^{j2\pi n/N} + e^{j4\pi n/N} - e^{j2\pi(N-2)n/N} - 0.5e^{j2\pi(N-1)n/N}] \\
h_a(n) &= j\frac{1}{N}[0.5e^{j2\pi n/N} + e^{j4\pi n/N} - e^{-j2\pi 4n/N} - 0.5e^{-j2\pi n/N}] \\
h_a(n) &= j\frac{1}{N}[j.sin(2\pi n/N) + j.sin(4\pi n/N)] \\
h_a(n) &= -\frac{1}{N}[sin(2\pi n/N) + sin(4\pi n/N)]
\end{aligned}$$

– Pour $N = 7 : h_a(n) = -\frac{1}{7}[sin(2\pi n/7) + sin(4\pi n/7)]$ pour $n = 0 \dots 6$.

n	0	1	2	3	4	5	6	7
h(n)	0	-0.39	-0.0153	0.1614	-0.1614	0.0153	0.39	0

4. Equation aux différences du filtre et fonction de transfert en z :

$$\begin{aligned}
y(n) &= -0.39x(n-1) - 0.0153x(n-2) + 0.1614x(n-3) \\
&\quad -0.1614x(n-4) + 0.0153x(n-5) + 0.39x(n-6) \\
H_a(z) &= -0.39(z^{-1} - z^{-6}) - 0.0153(z^{-2} - z^{-5}) + 0.1614(z^{-3} - z^{-4})
\end{aligned}$$

5. On peut également déduire $H_a(z)$ directement de $H_a(k\Omega_e)$ sous forme de cellules du second ordre en parallèle réelles en utilisant la formule vue en cours.

$$\begin{aligned}
H_a(z) &= \frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k\Omega_e)}{1-z^{-1}.e^{\frac{j2\pi k}{N}}} \\
H_a(z) &= \frac{1-z^{-N}}{N}\left[\frac{0.5}{1-z^{-1}.e^{\frac{j2\pi}{N}}} + \frac{1}{1-z^{-1}.e^{\frac{j4\pi}{N}}} - \frac{1}{1-z^{-1}.e^{\frac{-j4\pi}{N}}} - \frac{0.5}{1-z^{-1}.e^{\frac{-j2\pi}{N}}}\right] \\
H_a(z) &= \frac{1-z^{-N}}{N}\left[\frac{-z^{-1}sin(2\pi/N)}{1-2cos(2\pi/N)z^{-1}+z^{-2}} + \frac{-2z^{-1}sin(4\pi/N)}{1-2cos(4\pi/N)z^{-1}+z^{-2}}\right] \\
H_a(z) &= (1-z^{-7})\left[\frac{-0.11z^{-1}}{1-1.247z^{-1}+z^{-2}} + \frac{-0.28z^{-1}}{1-0.445z^{-1}+z^{-2}}\right]
\end{aligned}$$

6. Montrer que les deux versions du 4. et du 5. sont équivalentes : il suffit de réduire au même dénominateur la formule précédente, puis de faire une division polynomiale.
7. Les problèmes sur le filtre (module de $H_a(\Omega)$ très différent d'un dérivateur) viennent de la contrainte sur la phase qui a été implicitement posée. En effet, la spécification impose une phase nulle. D'autre part, on a ici $N = 7$ alors que le centre de symétrie est placé en 3.5, différent du $\alpha = \frac{N-1}{2}$. Il faut donc utiliser un filtre à phase linéaire $h_b(n)$. Pour cela, nous devons décaler $h_a(n)$ de $\alpha = 3 : h_b(n) = h_a(n-3)$. Dans ce cas le filtre se comporte beaucoup mieux en fréquence. Ce résultat est illustré dans le TP.

13.5 Transformée de Fourier Discrète et Rapide (TFD et TFR)

13.5.1 TFD bi-dimensionnelle

$$\begin{aligned} X(m,n) &= \sum_{k=0}^{N-1}\left[\sum_{l=0}^{N-1} x(k,l)e^{-2j\frac{\pi nl}{N}}\right]e^{-2j\frac{\pi mk}{N}} \\ &= \sum_{k=0}^{N-1}\overrightarrow{X(k)}e^{-2j\frac{\pi mk}{N}} \end{aligned}$$

où $\overrightarrow{X(k)}$ est le vecteur formé par la TFD de ligne k de l'image composée des pixels $x(k,l)$ avec $l = 0 \dots N-1$. $X(m,n),\ m,n = 0 \dots N-1$ est donc calculé à partir de deux TFD successives sur les lignes puis sur es colonnes (ou inversement).
Complexité : $O(N^2 \log_2 N)$

13.5.2 Transformée de Fourier Glissante

2. $X_{i+1}(n) = [X_i(n) - x(i) + x(i+N)]e^{-2j\frac{\pi n}{N}}$

3. TFR : $3N \log_2 N.Tcycle < Te$
TFD : $2N^2.Tcycle < Te$
TFR glissante : $4N.Tcycle < Te$

13.5.3 Transformée de Fourier en Base 4

13.5.4 Optimisation du calcul de la TFR d'une suite de nombres réels

1. $A = \frac{X(p)+X^*(N-p)}{2}$, $B = \frac{X(p)-X^*(N-p)}{2}$

$$\Re U(p) = \Re A + \Re B\cos\beta + \Im B\sin\beta \quad (13.1)$$
$$\Im U(p) = \Im A + \Im B\cos\beta - \Re B\sin\beta \quad (13.2)$$
$$\beta = \pi p/N \quad (13.3)$$

2. Méthode directe : $4N\log_2(2N)\otimes$ et $6Nlog_2(2N)\oplus$.
Méthode optimisée : $2N\log_2(N) + 4N\otimes$, $6Nlog_2(N) + 8N\oplus$ et $4N$ divisions par 2.

13.5.5 Optimisation du calcul de la TFR de deux suites de nombres réels

13.5.6 Comparaison TFTD et TFD

On a le signal suivant, avec $u(n)$ l'échelon unité, échantillonné à $T_e = 1$:

$$x(n) = e^{-an} \times u(n)$$

Soit $X_{TFTD}(f)$ la transformée de Fourier à temps discret de $x(n)$, on a :

$$\begin{aligned} X_{TFTD}(f) &= \sum_{n=-\infty}^{\infty} x(n) \times u(n) \times e^{-j2\pi nf} \\ &= \sum_{n=0}^{\infty} e^{-an} \times e^{-j2\pi nf} \\ &= \sum_{n=0}^{\infty} (e^{-a} \times e^{-j2\pi f})^n \end{aligned}$$

Si $|a| < 1$ la série précédente converge, on obtient alors :

$$X_{TFTD}(f) = \frac{1}{1 - e^{-a} \times e^{-j2\pi f}}$$

Soit $X_{TFD}(k)$ la transformée de Fourier discrète de $x(n)$ pour $n = 0 \cdots N-1$, on a :

$$\begin{aligned} X_{TFD}(k) &= \sum_{n=0}^{N-1} x(n) \times e^{-j2\pi \frac{nk}{N}} \\ &= \sum_{n=0}^{N-1} e^{-an} \times e^{-j2\pi \frac{nk}{N}} \\ &= \sum_{n=0}^{N-1} (e^{-a} \times e^{-j2\pi \frac{k}{N}})^n \\ &= \frac{1 - (e^{-a} \times e^{-j2\pi \frac{k}{N}})^N}{1 - e^{-a} \times e^{-j2\pi \frac{k}{N}}} \\ &= \frac{1 - e^{-aN} \times e^{-j\frac{2\pi kN}{N}}}{1 - e^{-a} \times e^{-j2\pi \frac{k}{N}}} \\ &= \frac{1 - e^{-a}}{1 - e^{-aN} \times e^{-j2\pi \frac{k}{N}}} \end{aligned}$$

Dans le calcul de la TFD, la variable fréquence, f, de la transformée de Fourier est échantillonnée pour des valeurs $f_k = k\frac{F_e}{N}$, avec ici $F_e = 1$. On a alors la relation suivante entre $X_{TFD}(k)$ et $X_{TFTD}(f)$:

$$\begin{aligned} X_{TFD}(k) &= \frac{1}{1 - e^{-a} \times e^{-j2\ pi\frac{k}{N}}} - \frac{e^{-aN}}{1 - e^{-a} \times e^{-j2\pi \frac{k}{N}}} \\ X_{TFD}(k) &= X_{TFTD}(\frac{kF_e}{N}) \times [1 - e^{-aN}] \\ X_{TFD}(k) &= X_{TFTD}(\frac{kF_e}{N}) \times [1 - \varepsilon(N, a)] \end{aligned}$$

La dernière équation du système précédent montre que la TFD de $x(n)$ correspond à l'échantillonnage fréquentiel de la TFTD de ce même signal mutiplié par un terme d'erreur. On a :

$$\lim_{n\to\infty}[1-\varepsilon(N,a)]=1$$

On s'arrange pour qu'en pratique on puisse négliger $\varepsilon(N,a)$

13.5.7 TFD par convolution

13.5.8 Bruits dans la TFD

La TFD, $X(k)$, d'un signal $x(n)$ à durée limité est donnée par la relation suivante :

$$X(k)=\sum_{n=0}^{N-1}x(n)\times W_N^{nk},\quad 0\le k\le N-1,\quad W_N^{nk}=e^{-j\frac{2\pi nk}{N}}$$

Pour le calcul d'un échantillon de la sortie, l'équation précédente nous indique qu'il faut effectuer N multiplications complexes et $N-1$ additions complexes. Soit encore, $2N$ multiplications réelles et $2(N-1)$ additions réelles. Pour traiter N échantillons, la complexité est de N^2 multiplications complexes et $N(N-1)$ additions complexes.
Le modèle statistique permettant de représenter l'opération de quantification est indiqué figure 13.6.

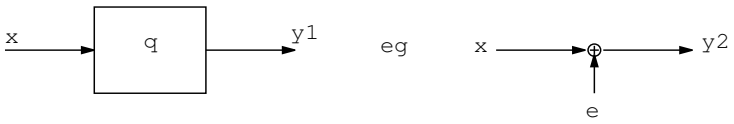


FIG. 13.6: modélisation statistique de l'opération de quantification

la source de bruit additif $e(n)$ est un signal aléatoire, stationnaire, de valeurs comprises dans l'intervalle $[0,q]$ s'il s'agit d'une troncature ou $[-\frac{q}{2},+\frac{q}{2}]$ s'il s'agit d'un arrondi avec $q=2^{-b}$ si les échantillons sont codés sur b bits utiles. On suppose une densité de probabilité uniforme ($=\frac{1}{q}$). S'il s'agit d'un arrondi la moyenne de $e(n)$ est nulle et sa variance $=\frac{q^2}{12}$. S'il s'agit d'une troncature, la moyenne $=\frac{q}{2}$ et la variance $=\frac{q^2}{12}$. On suppose ici une quantification par arrondi.
le signal de sortie est un signal complexe, il faut donc distinguer le bruit sur les parties réelles et imaginaires de la transformée :
Soit σ_s^2 la puissance du bruit en sortie de la TFD sur la partie réelle ou imaginaire de la transformée, on a :

$$\sigma_s^2=\sum_{n=0}^{N-1}(\sigma_e^2+\frac{q^2}{12})$$

On a donc au total (en composant les variances des parties réelles et imaginaires) :

$$\sigma_t^2 = 2 \times (N \times (\frac{q^2}{12} + \frac{q^2}{12})) = N \times \frac{q^2}{3}$$

En exprimant le RSB sur une échelle logarithmique, on obtient avec P_s la puissance du signal utile et P_b la puissance du bruit :

$$RSB_{dB} = 10\log(\frac{P_s}{P_b}) = 10\log(P_s) - 10\log N\frac{q^2}{3}$$

En considérant la puissance du signal et le terme en $N/3$ comme une constante relativement au nombre de bits, on a :

$$RSB_{dB} = K + 20 \times b \times \log(2) = K + b \times 6.02$$

Si on double le nombre de bits servant à coder les échantillons, on augmente de RSB de $b \times 6_{dB}$.
En prenant $b = 8$ bits, si on passe à 16 on a un gain de $+48dB$.
Reprenons la définition de la TFD, on a la relation de récurrence suivante :

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) \times W_N^{nk} \\ X(k) &= \sum_{n=0}^{N-1} x(n) \times (W_N^k)^n \end{aligned}$$

En développant le signe somme, on a :

$$X(k) = \left(\left(\cdots \Big(\big(x(N-1) \times W_N^k \big) + x(N-2) \Big) \times W_N^k + \cdots \right) \times W_N^k \right) + x(0)$$

On trouve bien une relation de récurrence de la forme :

$$y(m) = y(m-1) \times W_N^k + x(N-m),$$

en prenant $y(0) = x(0)$ comme condition initiale, et $m = 1 \cdots N$, on a alors $X(k) = y(N)$.
L'intérêt de cette mise en forme des calculs tient au fait que l'on adresse toujours le même élément W_N^k, il s'agit de l'algorithme de Goertzel.

13.5.9 Étude des bruits de calcul dans la transformée de Fourier Rapide

1. 1 multiplication complexe = 4 multiplications réelles.

$$e_1 = q^2/3; \quad e_2 = e_3 = 0; \quad B_s = 2B_e + q^2/3$$

2. $B_n = 2B_{n-1} + q^2/3 \approx N.B_0 + Nq^2/3$
3. $B_s = 2B_e + q^2/6; \quad B_n = N.B_0 + Nq^2/6$
4. La puissance du bruit est divisée par 4 : $B_s = 2q^2/4 + (2B_e + q^2/3)/4$
5. $Bn \approx q^2$

13.5.10 Calculs de TFD

La Transformée de Fourier discrète d'un signal composé de N échantillons s'écrit :

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

Avec $x(n)$ le signal d'entrée et $X(k)$ sa transformée, on a la relation matricielle suivante :

$$\begin{pmatrix} X(0) \\ \vdots \\ X(N-1) \end{pmatrix} = \begin{pmatrix} W_N^{0\times 0} & \cdots & W_N^{0\times(N-1)} \\ \vdots & W_N^{k\times n} & \vdots \\ W_N^{0\times(N-1)} & \cdots & W_N^{(N-1)\times(N-1)} \end{pmatrix} \times \begin{pmatrix} x(0) \\ \vdots \\ x(N-1) \end{pmatrix}$$

Il faut noter les propriétés de périodicité des racines W, on a :

$$\begin{aligned} W_N^{k(N-n)} &= (W_N^{kn})^* \\ W_N^{kn} &= W_N^{k(n+N)} \end{aligned}$$

On prend maintenant $N = 4$, on obtient alors, après simplification :

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & (W_4^1)^* \\ 1 & W_4^2 & W_4^4 & W_4^2 \\ 1 & (W_4^1)^* & W_4^2 & W_4^1 \end{pmatrix} \times \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{pmatrix}$$

Ou encore :

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & j \\ 1 & j & -1 & -j \end{pmatrix} \times \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{pmatrix}$$

Soit le signal $h(n)$ de durée finie, $N = 4$, pour trouver $H(k)$ il suffit d'une TFD sur 4 points, en utilisant la relation matricielle précédente, on a :

$$\begin{pmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & j \\ 1 & j & -1 & -j \end{pmatrix} \times \begin{pmatrix} \frac{1}{10} \\ \frac{2}{10} \\ \frac{3}{10} \\ \frac{4}{10} \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{5}(1-j) \\ -\frac{1}{5} \\ -\frac{1}{5}(1+j) \end{pmatrix}$$

$x(n)$ est un signal périodique de période $N = 4$, pour trouver $X(k)$, pour $k = 0 \cdots 3$ il suffit d'une TFD sur 4 points :

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & j \\ 1 & j & -1 & -j \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{5}(1-j) \\ -\frac{1}{5} \\ -\frac{1}{5}(1+j) \end{pmatrix}$$

On a :

$$\begin{aligned} H(k) &= \frac{1}{10} + \frac{2}{10}(-j)^k + \frac{3}{10}(-1)^k + \frac{4}{10}(j)^k \\ X(k) &= -e^{-j\frac{\pi k}{2}} + 2e^{-j\pi k} + e^{j\frac{3\pi k}{2}} \\ &= e^{-j\pi k}(2 + 2\cos(\frac{\pi k}{2})) \end{aligned}$$

Comme $h(n)$ est à durée limitée on a $H_{TFD}(k) = H_{TFTD}(e^{j\frac{2\pi k}{N}})$. Sa transformée de Fourier discrète correspond à un échantillonnage de sa transformée de Fourier. Comme $x(n)$ est périodique sa TFD est exactement sa TF.
Si on écrit $y(n)$ résultat du filtrage de $x(n)$ par $h(n)$, on a :

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

On constate que le schéma périodique temporel de $y(n)$ est de longueur 7 et non 4 !. On a donc $Y(k) \neq X(k) \times H(k)$! Pour obtenir $Y(k)$ il faut compléter par des zéros les signaux $h(n)$ et une période de $x(n)$ pour traiter le support $n = 0, .., 6$. Il est ensuite possible d'appliquer le produit des TFD.

13.5.11 Transformée en Cosinus Rapide

13.5.11.1 Complexité des calculs

Le calcul d'un papillon élémentaire est représenté par l'équation suivante :

$$\begin{aligned} X' &= X + c_k \times Y \\ Y' &= X - c_k \times Y \end{aligned}$$

Le signal d'entrée est réel, les c_k sont réels donc tous les calculs se font sur des nombres réels. Un papillon nous donne 1 multiplication réelle et 2 additions/soustractions. Comme pour la FFT, il y a $\frac{N}{2}\log_2(N)$ papillons si la dimension du vecteur de travail est N. Ce qui fait pour N échantillons traités, $\frac{N}{2}\log_2(N)$ multiplications et $N\log_2(N)$ additions. En faisant des calculs "in place", il suffit de N mots mémoires pour stocker les résultats ; il faut cependant $\frac{N}{2}$ mots supplémentaires pour stocker la table des cosinus multiplicatifs.
On considère une machine effectuant en parallèle une addition et une multiplication en un temps de cycle de $T_c = 50ns$. Si $T_e = 10^{-6}s$ est la période d'échantillonnage, on a alors la relations suivante :

$$(N + \frac{N}{2})\log_2(N) \times T_c < N \times T_e$$

On trouve alors $N < 2^{13}$.

13.5.11.2 Implantation en virgule fixe

Le graphe flot du calcul d'un papillon élémentaire est représenté figure 13.7. On modélise les imprécisions de calcul par trois sources de bruit : e_x et e_y modélisant le bruit superposé au signal de l'entrée (quantification du signal d'entrée), e_m modélisant le bruit de calcul dû à la multiplication de Y par C_k.

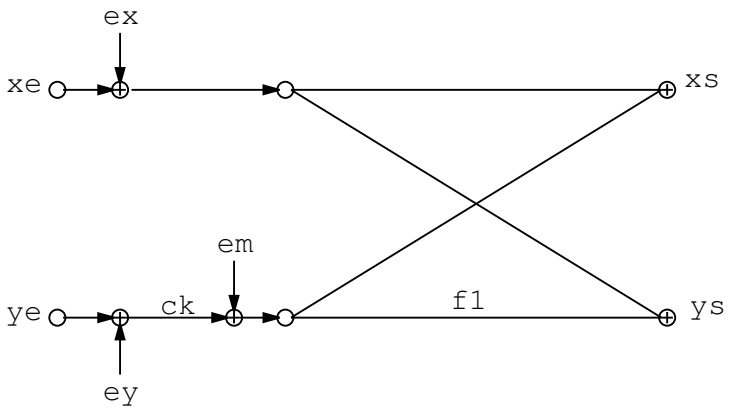


FIG. 13.7: Graphe flot des calculs d'un papillon élémentaire de la TCR

On cherche à déterminer la caractéristique du bruit en sortie du papillon élémentaire. Pour cela on suppose les entrées $X = e_x$ et $Y = e_y$ (on ne travaille qu'avec les sources de bruit...). On pose s_x et s_y les sorties respectives en X' et Y' correspondant aux entrées e_x et e_y. En tenant compte de la source de bruit e_m due à la multiplication, on a alors les relations suivantes :

$$\begin{aligned} s_x &= e_x + c_k \times e_y + e_m \\ s_y &= e_x - c_k \times e_y + e_m \end{aligned}$$

En faisant l'hypothèse de bruits blancs non corrélés entre eux, on additionne les variances :

$$\sigma_{sx}^2 = \sigma_{sy}^2 = \sigma_{ex}^2 + c_k^2 \sigma_{ey}^2 + \frac{q^2}{12}$$

On cherche à déterminer la puissance du bruit en sortie du graphe complet, on a pour un papillon :

$$\sigma_s^2 \leq \frac{q^2}{12} + 2\sigma_{ex}^2$$

Au total la TCR comporte $m = log_2(N)$ étages de traitement, on obtient après m étages :

$$\sigma_s^2 \leq 2^m \sigma_e^2 + \frac{q^2}{12}[\sum_{k=0}^{m-1} 2^k]$$

les sources de bruit en entrée modélise un bruit de quantification, on a donc $\sigma_e^2 = \frac{q^2}{12}$, en fin de traitement on obtient donc :

$$\sigma_{fi}^2 = \frac{q^2}{12} \sum_{k=0}^{m} 2^k = (2^{m+1} - 1)\frac{q^2}{12} = (2N - 1)\frac{q^2}{12}$$

La question sur les arrondis conduit aux mêmes résultats que précédemment puisque les puissances des sources de bruit sont les mêmes. On remplace alors les puissances dues aux multiplications par celles dues à la quantification.

Les débordement proviennent des additions. Si $|X| < 1$ à l'entrée du premier papillon, à la sortie on obtient la relation suivante :

$$|X'| < 1 + |C_k| < 2$$

Si maintenant, on se place à la sortie de l'étage de traitement m, on obtient :

$$|X'| < 2^m$$

Un échantillon de sortie de la TCR vérifie donc la relation suivante :

$$|X'| < N$$

Il suffit donc de diviser le signal d'entrée par N pour que chaque échantillon de sortie soit inférieur à 1 et évite tout débordement de calcul. On a vu en cours que diminuer la dynamique d'un signal en entrée d'un processus de traitement contribue à diminuer le rapport signal à bruit. Il existe une autre solution pour éviter les débordements consistant à diviser par 2 le signal à l'entrée de chaque papillon. Il est clair que la sortie de chaque papillon est bornée par 1. Cependant la division par 2 introduit un bruit de calcul supplémentaire que l'on modélisera comme une source de bruit de puissance $\frac{q^2}{4}$. On reprend le graphe flot de la figure 13.7 et on lui ajoute une source de bruit avant la multiplication par c_k. On obtient alors pour un papillon élémentaire, la relation suivante :

$$\begin{aligned}
\sigma_{sx}^2 &= \sigma_{ex}^2 + c_k^2(\sigma_{ey}^2 + \frac{q^2}{4}) + \frac{q^2}{12} \\
\sigma_s^2 &\leq 2\sigma_e^2 + \frac{q^2}{4} + \frac{q^2}{12} \\
\sigma_s^2 &\leq 2\sigma_e^2 + \frac{q^2}{3}
\end{aligned}$$

Au dernier étage de la TCR, $m = \log_2(N)$, on a :

$$\begin{aligned}
\sigma_{fi}^2 &= 2^m\sigma_e^2 + \frac{q^2}{3}[\sum_{k=0}^{m-1} 2^k] \\
\sigma_{fi}^2 &= N\frac{q^2}{12} + (N-1)\frac{q^2}{3} \\
\sigma_{fi}^2 &\approx N\frac{5q^2}{12}
\end{aligned}$$

13.6 Analyse spectrale

13.6.1 Questions

1. $10kHz/1024 = 9.765Hz$. Attention, cette valeur est différente de la finesse en fréquence.
2. Une bande de 0 à $10kHz$ implique $f_e \geq 20kHz$.
   Atténuation $> 40dB \Longrightarrow$ Fenêtre de Hamming (ou Blackman).
   Hamming : $\Delta\Omega = 8\pi/N \Longrightarrow N \geq 4f_e/1Hz = 80.000 \Longrightarrow N = 2^{17}$
   $Tcalcul = (N + 3Nlog_2N + 2N).Tcycle < 1/25 \Longrightarrow Tcycle = 25ns$

13.6.2 Analyse spectrale d'un signal sinusoïdal

13.6.3 Analyse spectrale d'un signal

A vérifier sous `Matlab` ou `Scilab`.

13.7 Convolution

13.7.1 Calcul d'une convolution

$$y(n) = \frac{b^{n+1}\left[1-(a/b)^{n+1}\right]}{b-a}u(n)$$

13.7.2 Complexité de calcul d'une convolution

Voir le TP de TNS.

13.8 Interpolation et décimation

13.8.1 Interpolation linéaire

$$x(k+1/2) = [x(k)+x(k+1)]/2$$

1. Avec $x(0) = x(N) = 0$:

$$Y(p) = X(p)[1+\cos(\pi p/N)],\ 0 \leq p \leq N-1 \quad (13.4)$$
$$Y(p) = X(p)[1-\cos(\pi p/N)],\ N \leq p \leq 2N-1 \quad (13.5)$$

2. On fait une TFR, puis on recombine la sortie pour obtenir $Y(p)$.
   Méthode directe : $N\otimes, N\oplus$
   Méthode TFR : $Nlog_2(N)/2 + N/2 \otimes \mathbb{C}, Nlog_2(N) + N \oplus \mathbb{C}$

13.8.2 Suréchantillonnage

1. Il s'agit du même signal mais échantillonné à une fréquence f'_e différente. En fait, ajouter des zéros puis filtrer revient à suréchantillonner.
2. M multiple de N.

Bibliographie

[Bel87] M. Bellanger. *Traitement Numérique du Signal.* Collection CNET-ENST, MASSON, 1987.

[BL80] R. Boite and H. Leich. *Les Filtres Numériques Analyse et Synthèse des filtres unidimensionnels.* Collection CNET-ENST, Masson, 1980.

[BP85] C.S. Burrus and T.W. Parks. *DFT/FFT and Convolution Algorithms.* Topics in DSP : John Wiley & Sons, 1985.

[EW92] Van Den Enden and Werdeckh. *Traitement Numérique du Signal : Une Introduction.* Masson, 1992.

[HL97] D. Hanselman and B. Littlefield. *Matlab : the language of technical computing.* Prentice Hall, 1997.

[Ka91] M. Kunt and al. *Techniques modernes de Traitement Numérique du Signal.* Collection CNET-ENST, Presses Romandes, Masson, 1991.

[Kun81] M. Kunt. *Traitement Numérique des Signaux.* Collection Traité d'Electricité Presses Romandes, 1981.

[ME93] C. Marven and G. Ewers. *a simple approach to Digital Signal Processing.* Texas Instruments Mentors, 1993.

[MSY98] J. McClellan, R. Schafer, and M. Yoder. *DSP First : a Multimedia Approach.* Prentice Hall, 1998.

[OS75] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice-Hall, 1975.

[OS99] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing, second edition.* Prentice-Hall, 1999.

[PB87] T.W. Parks and C.S. Burrus. *Digital Filter Design.* Topics in DSP, John Wiley & Sons, 1987.

[PM73] T.W. Parks and J.H. McClellan. A unified approach to the design of optimum linear phase digital filters. *IEEE Transactions on Circuit Theory*, 20 :697–701, Nov. 1973.

[PM96] J. Proakis and D. Manolakis. *Digital Signal Processing : Principles, Algorithms and Applications.* Prentice Hall, 1996.

[Poa97] B. Poart. *A Course in Digital Signal Processing.* John Wiley & Sons, 1997.

[SS88] R. David S. Stearns. *Signal Processing Algorithms.* Prentice Hall, 1988.

Annexe A

Examens

A.1 DS novembre 2001

UNIVERSITE DE RENNES 1

ENSSAT EII2
DS Traitement Numérique du Signal
Tous documents autorisés
Vendredi 30 novembre 2001

ENSSAT

Problème 1 : Synthèse d'un filtre réjecteur-de-bande (≈ 7 points)

1. On souhaite réaliser un filtre réjecteur de bande RIF à phase linéaire, de fréquences de coupure $\Omega_1 = \Omega_0 - \Omega_c$ et $\Omega_2 = \Omega_0 + \Omega_c$. On prendra $\Omega_0 = \pi/4$ et $\Omega_c = \pi/8$. Représenter l'amplitude et la phase de $H(e^{j\Omega})$ sur l'intervalle $[-\pi, \pi]$.

2. Prévoir le type de la réponse impulsionnelle ainsi que la parité de sa longueur N.

3. Donner l'expression des coefficients $h(n)$ de la réponse impulsionnelle du filtre idéal.

4. On souhaite transformer $h(n)$ en un filtre causal à phase linéaire, de longueur finie N la plus petite possible, respectant le gabarit ci-dessous figure A.1. Comment s'y prendre ? Quelle valeur de N choisir ?

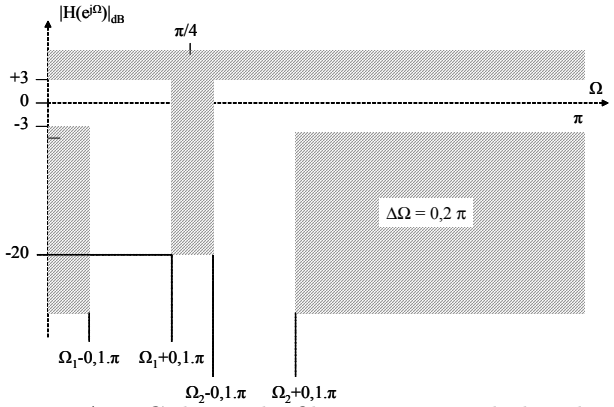


FIG. A.1: Gabarit du filtre réjecteur de bande

5. On cherche maintenant à réaliser un filtre RII respectant ce gabarit, tel que la phase reste proche de la linéarité. Quel type de filtre choisir ?

6. Ce filtre RII est synthétisé par la méthode de la transformation bilinéaire. La fréquence d'échantillonnage est fixée à $8kHz$. On veut bien sur que ce le filtre numérique entre dans le gabarit de la figure A.1.
– Dessiner le gabarit analogique équivalent.
– En déduire le gabarit du prototype passe-bas.

– Déterminer l'ordre et donner la fonction de transfert normalisée $H_N(p)$.
– Exprimer la fonction de transfert $H(p)$ du filtre analogique équivalent en fonction de la largeur de bande analogique B et de la pulsation centrale analogique ω_{centr}.

7. Soit $H(z)$ la fonction de transfert du filtre numérique obtenu par transformation bilinéaire à partir de $H(p)$. Sans faire de calcul, pouvez-vous dire si le filtre RII ainsi obtenu sera plus intéressant que la réalisation RIF précédente, en terme de complexité ?

Problème 2 : Implantation d'un RII en virgule fixe (≈ 8 points)

Nous souhaitons implanter un filtre passe-bas de type Butterworth dont la fonction de transfert $H(z)$ est définie à l'équation A.1. Soit $x(n)$ l'entrée de ce filtre et $y(n)$ sa sortie. Nous considérons que l'entrée $x(n)$ est comprise dans l'intervalle $]-1,1[$.

$$H(z) = \frac{0.2 + 0.4z^{-1} + 0.2z^{-2}}{1 - 0.4z^{-1} + 0.2z^{-2}} \tag{A.1}$$

Le processeur utilisé est un DSP de type TMS320C50. Les différentes caractéristiques de ce processeur sont les suivantes :
– les calculs sont réalisés en double précision ;
– les données en entrée du multiplieur sont codées sur 16 bits et la sortie sur 32 bits ;
– les données en entrée et en sortie de l'additionneur sont codées sur 32 bits ;
– les données sont stockées en mémoire sur 16 bits.
Pour simplifier le fonctionnement du processeur nous considérons que nous possédons une instruction assembleur permettant de charger une donnée stockée en mémoire sur 16 bits dans la partie haute de l'accumulateur et une instruction permettant de transférer directement le contenu de la partie haute de l'accumulateur vers la mémoire.
Nous considérons que le bit de signe redondant en sortie de la multiplication est automatiquement éliminé.

1. Nous considérons une donnée issue d'un processus de quantification et possédant le format (b, m, n)[1]
– Déterminer l'expression du pas de quantification associé à cette donnée.
– En déduire la puissance du bruit de quantification associé à cette donnée en considérant que le mode de quantification utilisé est l'arrondi.

Structure directe non canonique

2. Nous utilisons une structure directe non canonique pour implanter ce filtre. Donner la structure de réalisation de ce filtre (graphe flot de signal).

3. Pour étudier la dynamique de la sortie du filtre nous utilisons la norme de Chebychev. Démontrer que la dynamique de la sortie du filtre $y(n)$ est inférieure à 1 ($y \in]-1,1[$).

[1] b représente le nombre total de bits utilisés pour coder la donnée, m représente le nombre de bits pour la partie entière et n représente le nombre de bit pour la partie fractionnaire

4. En déduire le codage des données et des coefficients. Pour ajuster les formats nous réalisons un recadrage des coefficients.

5. Nous considérons que le signal d'entrée est issu de la quantification d'un signal analogique. Après avoir identifié les différentes sources de bruit vous déterminerez l'expression et la valeur numérique de la puissance de chaque source de bruit. En déduire l'expression de la puissance du bruit en sortie du filtre.

6. Déterminer l'expression de la puissance du bruit en sortie du filtre dans le cas d'une architecture ne permettant que de réaliser des calculs en simple précision (tous les chemins de données sont limités à 16 bits).

Structure canonique transposée

7. Nous utilisons maintenant une structure canonique transposée pour implanter ce filtre. Donner la structure de réalisation de ce filtre (graphe flot de signal).

8. Nous allons déterminer la dynamique des données en sortie de chaque additionneur. D'après les résultats obtenus à la question 3, la dynamique de la sortie de l'additionneur générant $y(n)$ est inférieure à 1. Pour les deux autres additionneurs nous allons déterminer le domaine de définition de leur sortie en se plaçant dans le pire cas. Déterminer la dynamique des sorties des deux additionneurs à partir de celle de $x(n)$ et de $y(n)$ en se plaçant dans le pire cas.

9. En déduire le codage des données et des coefficients. Nous souhaitons sauvegarder en mémoire le résultat des additions avec le maximum de précision, ainsi, vous pouvez insérer des opérations de décalage si cela est nécessaire. *(Remarque : chaque coefficient peut posséder son propre codage).*

10. *(Question subsidiaire)* Analyser le comportement de cette structure en terme de bruit par rapport à la structure directe non canonique (sans faire de calcul). Quelles sont les principales différences avec la structure précédente ?

Problème 3 : Zoom sur TFD (≈ 5 points)

Dans ce problème nous allons étudier une méthode permettant de faire un *zoom* sur une zone fréquentielle particulière. A partir des N points d'une TFD $X_N(k)$ d'un signal $x(n)$ échantillonné à $F_e = 1MHz$, on souhaite donc effectuer un zoom sur la région $[\Omega_c - \Delta\Omega, \Omega_c + \Delta\Omega]$ et obtenir L points avec interpolation de cette zone.

La méthode est résumée dans le schéma figure A.2. A partir des N points de la TFD $X_N(k)$, $x(n)$ est calculé par TFD inverse, puis multiplié par $f(n) = e^{-j\Omega_c n}$ et filtré passe-bas par $h(n)$ pour former $x_1(n)$, décimé par un facteur Mpour obtenir $x_2(n)$. Le spectre zoom est alors obtenu par une TFD sur P points ($P \geq L$) à partir du signal $x_Z(n)$ formé de $x_2(n)$ éventuellement complété par des zéros. Le filtre $h(n)$ est un filtre passe bas idéal de fréquence de coupure $\Delta\Omega$ défini par :

$$H(e^{j\Omega}) = \begin{cases} 0 & pour \quad -\pi \leq \Omega < \Delta\Omega \quad et \quad \Delta\Omega < \Omega \leq \pi \\ 1 & pour \quad -\Delta\Omega \leq \Omega \leq \Delta\Omega \end{cases}$$

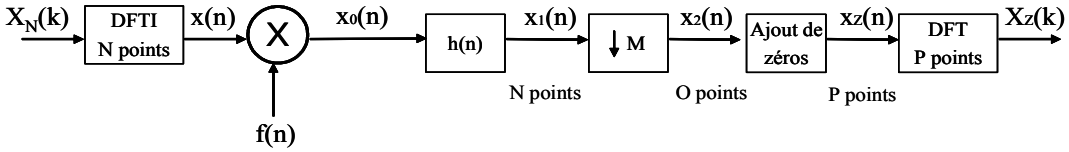


FIG. A.2: Schéma de principe de la *zoom transform*

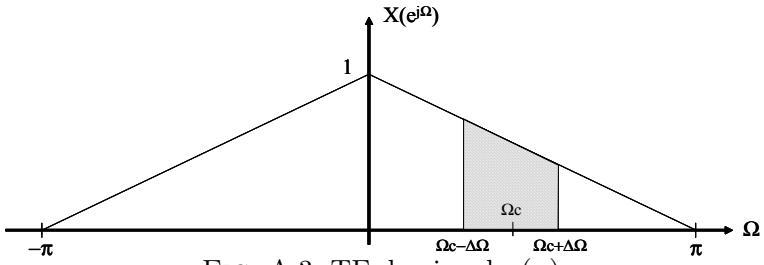


FIG. A.3: TF du signal $x(n)$

1. Soit $X(e^{j\Omega})$ la TF du signal $x(n)$, exprimez la TF du signal $x_0(n)$. Quelle opération a t-on effectué sur le signal $x(n)$? Représenter le spectre $X_0(e^{j\Omega})$ du signal $x_0(n)$ et celui de $x_1(n)$ si on considère que $X(e^{j\Omega})$ la TF du signal $x(n)$ est celle de la figure A.3.

2. Quelle valeur de M doit on prendre au mieux ? Expliquer votre choix et donner sa valeur en fonction des différents paramètres de la chaîne de traitement. Quelle est la nouvelle valeur de fréquence d'échantillonnage après traitement ? Combien de points composent la suite $x_2(n)$ (on posera O pour la suite) ?

3. Exprimer (sans chercher à en calculer le résultat) la TFD de $x_2(n)$. Le bloc suivant ajoute $P-O$ zéros à la fin du signal $x_2(n)$. On a : $x_Z(n) = \{x_2(n)\}\{0 \ldots 0\}$. Exprimer la TFD $X_Z(k)$ de $x_Z(n)$ et trouver une relation avec $X_2(k)$. Expliquer le résultat obtenu. Représenter $X_Z(k)$ entre 0 et $P-1$.

4. Donner la complexité de chaque bloc puis de la chaîne complète. On considérera que $N = 2^n$, $P = 2^p$ et que le filtre $h(n)$ est un filtre RIF avec K coefficients.

Pour les réponses aux questions précédentes vous prendrez les valeurs numériques suivantes : $N = 256$, $P = L = 256$, $K = 128$, $\Omega_c = \pi/3$, $\Delta\Omega = \pi/4$.

A.2 DS décembre 2000

UNIVERSITE DE RENNES 1

ENSSAT EII2

DS Traitement Numérique du Signal

Tous documents autorisés

Mercredi 20 décembre 2000

ENSSAT

Problème 1 : Synthèse d'un filtre RIF passe-bande (≈ 9 points)

On désire réaliser un filtre RIF passe-bande à phase linéaire dont la réponse en fréquence idéale $H(e^{j\Omega})$ possède une bande passante étendue entre les pulsations de coupure normalisées : $\Omega_{cl} = \pi/3$ et $\Omega_{ch} = 2\pi/3$.

1. Dessiner le module du filtre idéal $H(e^{j\Omega})$ entre $-\pi$ et 2π. Dans le cas où la fréquence d'échantillonnage est fixée à 20kHz, quelles sont les fréquences de coupure du filtre passe bande.
2. Calculer la réponse impulsionnelle $h(n)$ du filtre idéal $H(e^{j\Omega})$. Application numérique pour $-8 \leq n \leq +8$. Comment modifier $h(n)$ pour que le filtre présente une phase linéaire $\phi(\Omega) = -\alpha\Omega$, avec $\alpha > 0$.
3. On souhaite approcher ce filtre idéal par un filtre à réponse impulsionnelle finie, synthétisé par la méthode du **fenêtrage**. Comment obtenir, à partir de $h(n)$, un filtre causal, de longueur N **impaire**, à **phase linéaire**, tel que : - la largeur de sa bande de transition soit inférieure à $pi/4$, - l'atténuation hors bande soit supérieure à 40 dB. Calculer ses coefficients $h_1(n)$. (*Remarque* : les valeurs des coefficients des fenêtres sont données en fin de page.
4. On remplace le fenêtrage précédent par un fenêtrage rectangulaire sur 9 points. Quelles sont alors l'atténuation hors bande et la largeur de la bande de transition du filtre ainsi synthétisé ?

   On appelle $h_2(n)$ ce filtre. Expliquer pourquoi le filtre équivalent à la mise en cascade de deux réalisations de $h_2(n)$ respecte les spécifications de la question 3 (largeur de la bande de transition et atténuation hors bande).
5. Le signal en entrée est issu d'un convertisseur analogique numérique et codé en virgule fixe cadrée à gauche, sur b bits utiles. On considérera donc que le signal d'entrée est affecté d'un bruit de quantification. Toutes les opérations sont réalisées en simple précision.
   - Représenter le filtre $h_2(n)$ sous forme directe.
   - Calculer, en fonction de b, la puissance totale de bruit en sortie du filtre $h_2(n)$, en déduire la puissance totale de bruit en sortie du filtre constitué par la mise en cascade de deux réalisations de $h_2(n)$.
   - Calculer la puissance totale de bruit en sortie du filtre $h_1(n)$.
   - Après calcul des applications numériques, comparer les puissances de bruit des deux structures. Conclure.

   (*Remarque* : si vous n'avez pas trouvé les coefficients des filtres des premières questions, le problème peut être traité de manière analytique en utilisant les coefficients $h_1(n)$ et $h_2(n)$.)

Problème 2 : Etude d'un filtre RII (≈ 4 points)

Soit le filtre numérique RII du premier ordre suivant : $y(n) = (1-b).x(n) + b.y(n-1)$, avec $0 < b < 1$.

1. Déterminer la fonction de transfert en z $H(z)$, puis la réponse fréquentielle $H(e^{j\Omega})$ de ce filtre.

2. Donner les valeurs de $|H(e^{j\Omega})|$ en $\Omega = 0$ et $\Omega = \pi$. Quel type de filtre est réalisé ?
3. Déterminer la relation entre la pulsation de coupure normalisée à -3dB Ω_c et le coefficient b.
4. Pour une fréquence d'échantillonnage de 8kHz, on veut que ce filtre ait une fréquence de coupure à -3dB à 1kHz. Quelle est la valeur de b ?
5. Que deviennent les coefficients du filtre lorsqu'on les code en virgule fixe cadrée à gauche sur 4 bits, puis 8 bits. Que devient, dans les 2 cas, la valeur de la fréquence de coupure ?
6. D'après l'équation $y(n) = (1-b).x(n)+b.y(n-1)$, si $x(n)$ est codé en virgule fixe cadrée à gauche, sous quelle condition peut on garantir que le filtre ne déborde pas.

Problème 3 : Algorithme de Bluestein (≈ 3 points)

Soit le filtre numérique dont la réponse impulsionnelle est : $h(n) = \begin{cases} 0 & pour \quad n > N-1 \\ e^{j\pi n^2/N} & pour \quad 0 \leq n \leq N-1 \end{cases}$

On l'utilise dans la chaîne de traitement numérique du signal définie par la figure A.4, avec $f(n) = g(n) = e^{-j\pi n^2/N}$, et $x(n)$ un signal de durée finie N.

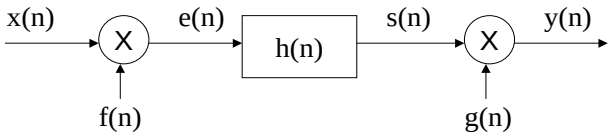


FIG. A.4: Filtrage de Bluestein

1. Démontrer que cette chaîne algorithmique de TNS est équivalente à un autre algorithme bien connu de tous. Pour cela, exprimer $y(n)$ en fonction de $x(n)$.
2. Calculer la complexité en nombre d'opération de l'algorithme pour le calcul d'un point du signal de sortie.
3. Donnez les avantages de cette solution dans le cas d'un flot continu du signal $x(n)$ par rapport à la version classique.

Problème 4 : Conception d'un filtre RII (≈ 4 points)

Soit le filtre analogique défini par $H(p) = \frac{1-p}{1+p}$. On désire réaliser un filtre numérique équivalent par la méthode de la transformation bilinéaire.

1. Dessiner et calculer le module et l'argument de la réponse fréquentielle du filtre analogique $H(j\omega)$.
   Quel type de filtre réalise $H(p)$?
   Quelle est la valeur de ω pour laquelle la phase du filtra analogique vaut $-\pi/2$?
2. Déterminer la fonction de transfert $H_b(z)$ du filtre numérique.
3. Calculer $H_b(e^{j\Omega})$ pour une période d'échantillonnage $T = 0.5$.
4. Calculer le module et la phase de $H_b(e^{j\Omega})$ pour $\Omega = 0$ et π.
5. Expliquer comment trouver la valeur de ω pour laquelle la phase du filtre numérique vaut $-\pi/4$?

n	0	1	2	3	4	5	6	7	8
Triangle	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
Hamming	0.08	0.1150154	0.2147309	0.3639656	0.54	0.7160344	0.8652691	0.9649846	1
Blackman	0	0.0146288	0.0664466	0.1720897	0.34	0.5547732	0.7735534	0.9385083	1

Valeurs de différentes fenêtres pour $N = 17$

A.3 DS janvier 2000

UNIVERSITE DE RENNES 1

ENSSAT EII2
DS Traitement Numérique du Signal
Tous documents autorisés
Vendredi 7 janvier 2000

ENSSAT

Problème 1 : Synthèse de filtre RIF (≈ 7 points)

On désire réaliser un filtre RIF dont la réponse s'approche du filtre idéal défini ci dessous.

$$H(e^{j\Omega}) = \begin{cases} -j & pour \quad \Omega \leq -\Omega_c \\ j\frac{\Omega}{\Omega_c} & pour \quad -\Omega_c < \Omega < \Omega_c \\ j & pour \quad \Omega \geq \Omega_c \end{cases}$$

1. Représenter le pseudo-module et la phase de $H(e^{j\Omega})$.
2. Prévoir le type de la réponse impulsionnelle h(n), ainsi que la parité de N.
3. Donner l'expression des termes h(n) de la réponse impulsionnelle pour $\Omega_c = \pi/2$. Dessinez h(n) pour $n = -4... + 4$.
4. On veut réaliser un filtre RIF $H_a(z)$ à phase linéaire dont la réponse impulsionnelle $h_a(n)$ est h(n) limitée à N points sans pondération. Indiquez comment obtenir ce filtre RIF. Dessinez $h_a(n)$ dans le cas ou N=7.
5. Donner les expressions du pseudo-module $A_a(e^{j\Omega})$ et de la phase $\Phi_a(\Omega)$.
6. Dessiner l'allure de $A_a(e^{j\Omega})$ dans le cas précédent ainsi que dans le cas où on pondérerait h(n) par une fenêtre de Hamming. Expliquez vos résultats.

Problème 2 : Etude d'un filtre RIF (≈ 9 points)

Soit le filtre numérique RIF à phase linéaire suivant : $y(n) = b_0.x(n) + b_1.x(n-1) + b_2.x(n-2)$

1. Déterminer la fonction de transfert en z $H(z)$, puis la réponse fréquentielle $H(e^{j\Omega})$ de ce filtre.
2. Déterminer les coefficients du filtre précédent pour qu'il rejète complètement une composante fréquentielle $\Omega_0 = 2\pi/3$, et que sa réponse fréquentielle soit normalisée telle que $H(0) = 1$.
3. Déterminer la réponse fréquentielle $H(e^{j\Omega})$ du filtre trouvé pour vérifier s'il respecte bien les spécifications précédentes.
4. Que deviennent les coefficients b_i lorsqu'on les code en virgule fixe cadrée à gauche sur 4 bits. Les conditions de la question 2 sont elles toujours remplies ?

Etude de l'implémentation d'un filtre RIF

5. Dessiner la structure directe du filtre précédent $H(z)$ dans le cas où $b_0 = 1$.

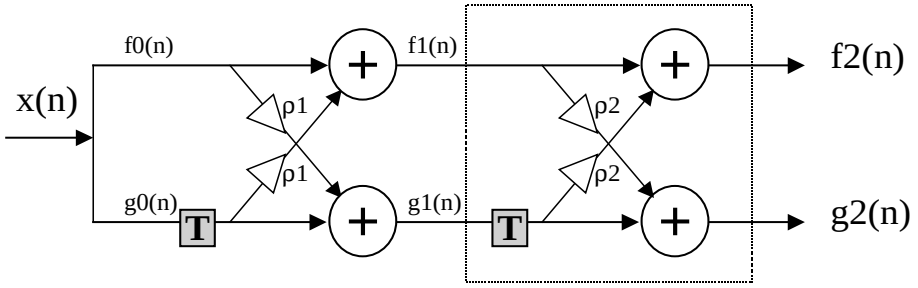


FIG. A.5: Filtre en structure treillis.

6. Donner la puissance du bruit en sortie en fonction du bruit en entrée sur une machine simple précision.
7. Exprimer les signaux $f_i(n)$ et $g_i(n)$ en fonction de $f_{i-1}(n)$ et $g_{i-1}(n)$. On appelle ce bloc (figure A.5) une structure élémentaire de treillis. En déduire l'expression de $f_2(n)$ et $g_2(n)$ en fonction de x(n).
8. Exprimer les coefficients ρ_1 et ρ_2 en fonction de b_1 et b_2 pour que $\frac{F_2(z)}{X(z)} = H(z)$. Dans ces conditions exprimez $\frac{G_2(z)}{X(z)}$ en fonction de $H(z^{-1})$.
9. Donner la puissance du bruit en sortie sur les signaux $f_i(n)$ et $g_i(n)$ en fonction du bruit sur les entrées de la cellule treillis, toujours sur une machine simple précision.
10. En déduire la puissance du bruit sur la sortie $f_2(n)$ du filtre.
    Dans le cas où les coefficients b_1 et b_2 valent tous deux 0.33, comparez les puissances de bruit de calcul en sortie des deux structures directe et treillis. Que peut on en conclure ?
11. Donner la complexité en nombre d'opérations arithmétiques des deux structures dans le cas où la réponse impulsionnelle de ce filtrage est étndue sur N points. Donner ensuite le temps de cycle d'un processeur de traitement du signal capable de réaliser en temps réel ce filtrage de taille 1024 sur un signal échantillonné à 20 kHz.

Problème 3 : Analyse spectrale ($\approx$ 4 points)

1. Donner la transformée de Fourier d'une fenêtre rectangulaire (c'est à dire un signal valant 1 pour $n = 0 \dots N-1$ et 0 ailleurs) de taille N et le module de celle ci $|W(e^{j\Omega})|$. Tracer ce dernier entre $[0 \dots 2\pi]$.
2. Nous avons vu en cours que le lobe principal d'une telle fonction a une largeur de $4\pi/N$, et que le second lobe possède une atténuation de -13dB. On notera également que l'amplitude du lobe principal $|W(0)| = N$. Quelle est approximativement la valeur de l'atténuation du plus petit lobe, c'est à dire la valeur du rapport entre l'amplitude maximale de ce lobe et celle du lobe principal.

   On considérera dans la suite le signal $x(n) = sin\left(2\pi\frac{k_0}{128}n\right) + 0.001sin\left(2\pi\frac{k_1}{128}n\right)$.
3. Dessiner le résultat d'une analyse spectrale par TFD sur N=128 points du signal x(n) si $k_0 = 6$ et $k_1 = 56$ sont des entiers.
4. Tracer maintenant approximativement le cas ou $k_0 = 6.3$ et $k_1 = 56$. Expliquer pourquoi une fenêtre rectangulaire ne permet pas de détecter correctement la deuxième composante de $x(n)$.
5. Des fenêtres de Hanning et de Hamming, laquelle est la meilleure pour cette analyse. Dans ce dernier cas, quel doit être l'écart minimum entre k_0 et k_1 pour distinguer les deux composantes.

$\int_a^b u.v' = [u.v]_a^b - \int_a^b u'.v$

A.4 DS mars 1999

UNIVERSITE DE RENNES 1

ENSSAT EII2

DS Traitement Numérique du Signal

Tous documents autorisés

2 mars 1999

ENSSAT

Problème 1 : Filtrage numérique (17 pts)

Le but de ce problème est de comparer la synthèse de filtres numériques à réponse impulsionnelle infinie et finie par les méthodes bilinéaires et de fenêtrage. Le filtre à réaliser est un filtre passe-haut dont les caractéristiques sont : ondulation en BP $\delta_1 = 3dB$, atténuation $\delta_2 = 20dB$, fréquence de coupure à $-3dB$ $f_c = 2,5kHz$, fréquence en bande atténuée $f_a = 1kHz$, fréquence d'échantillonnage $f_e = 10kHz$.
Les question suivantes sur les deux types de filtre peuvent être traitées séparément. Les questions sur le bruit de calcul peuvent être traitées sans nécessairement avoir réussi le début du problème.

1. **Synthèse de filtre RII par la méthode bilinéaire (8 pts)**
   Après avoir tracé le gabarit du filtre numérique, donnez les gabarits analogique et passe bas normalisé correspondants. En déduire l'ordre et la fonction du filtre de butterworth normalisé $H_n(p)$.
2. Après dénormalisation du filtre passe-haut $H(p)$, donnez la fonction de transfert $H_{bi}(z)$ du filtre numérique entrant dans le gabarit numérique de départ.
3. Donnez l'équation aux différences et dessinez la structure canonique du filtre.
4. Quelle est la complexité en opérations et mémoire de ce filtre ? Expliquez vos résultats.
5. **Etude des bruits de calcul du filtre RII** On considérera que le signal en entrée est bruité (bruit de puissance $q^2/12$), et que la puissance du signal n'est pas modifiée après filtrage. Donnez la puissance du bruit en sortie du filtre lorsque :
   - tous les chemins de données sont limités à b bits (simple précision),
   - les opérations s'effectuent en double précision, mais toutes les sauvegardes en mémoire sont effectuées sur b bits. Pour ce deuxième cas, vous expliquerez clairement votre raisonnement, et où se situent les quantifications.
6. **Application numérique** On approximera la réponse fréquentielle du filtre par le filtre idéal de fréquence de coupure f_c. Donnez la valeur de la puissance du bruit en sortie lorsque b = 8 bits.
7. **Synthèse de filtre RIF par fenêtrage (6 pts)**

On considérera pour la synthèse un filtre passe-haut idéal de fréquence de coupure f_c. Après avoir tracé de manière précise le filtre idéal, calculez $h(n)$, puis donnez ses valeurs pour $n = [-4 \dots + 4]$. Il vous faudra calculer $h(0)$ de manière isolée ($h(0) > 0$).

8. En fonction de l'atténuation et de la sélectivité, quel type de fenêtre faut il utiliser, et quelle est la longueur N d'un filtre RIF dont la phase serait linéaire ? Tracez sa réponse impulsionnelle $h_a(n)$ et donnez sa fonction de transfert en z $H_a(z)$.
9. Donnez l'équation aux différences et dessinez la structure directe du filtre.
10. Quelle est la complexité en opérations et mémoire de ce filtre. Expliquez vos résultats.
11. **Etude des bruits de calcul du filtre RIF** On se placera dans les mêmes conditions qu'au 1.5. Donnez la puissance du bruit en sortie du filtre lorsque :
    - tous les chemins de données sont limités à b bits (simple précision),
    - les opérations s'effectuent en double précision, mais toutes les sauvegardes en mémoire sont effectuées sur b bits. Pour ce deuxième cas, vous expliquerez clairement votre raisonnement, et où se situent les quantifications.
12. **Comparaison des filtres RII et RIF (3 pts)**
    Calculez les réponses fréquentielles $H_{bi}(e^{j\Omega})$ et $H_a(e^{j\Omega})$.Tracez l'allure de chacune des réponses fréquentielles. Expliquez les principales différences entre les résultats.
13. Comparez de manière qualitative les complexités et bruits de calcul des deux types de filtre.
14. Comparez rapidement les problèmes de débordement des deux solutions RII et RIF.

Remarques : si les coefficients des filtres ne sont pas obtenus aux questions 1.3 et 1.9, un filtre RII du deuxième ordre générique et un filtre RIF de longueur N générique peuvent être utilisés dans les dernières questions.

Problème 2 : Transformée en cosinus discret (3 points)

Une version de la TCD d'un bloc de N échantillons peut être donnée par la formule suivante :

$$X_C(k) = \sum_{n=0}^{N-1} x(n).cos\left(\frac{2\pi(4n+1)k}{4N}\right), k = 0 \dots N-1$$

1. Quelle est la complexité de la TCD. Justifiez votre réponse.
2. Démontrer que $X_C(k) = cos(\frac{2\pi k}{4N})Re[X(k)] + sin(\frac{2\pi k}{4N})Im[X(k)]$ où $X(k)$ est la TFD de $x(n)$.
3. Peut on en déduire une version "rapide" de la TCD. Quelle est sa complexité (opérations et mémoire) ? Pour quelle valeur de N cette solution est elle plus efficace ?

$|1 - e^{-j\Omega}| = |2sin(\Omega/2)|$ $\qquad$ $cos(a+b) = cosa.cosb - sina.sinb$

A.5 Correction du DS de novembre 2001

Problème 1 : Synthèse d'un filtre réjecteur-de-bande (≈ 9 points)

1. RIF à phase linéaire réjecteur de bande.
2. Type I : RI symétrique, N impair.
3. $h(n) = \delta(n) - \frac{2\Omega_c}{\pi} sinc(n\Omega_c) cos(n\Omega_0)$.
4. La phase linéaire implique un décalage de $\alpha = \frac{N-1}{2}$. La fenêtre rectangulaire convient. $N = 11$.
5. RII de type Bessel.
6. Filtre RII :
   - Même gabarit que précédemment mais avec prédistorsion. Les nouvelles pulsation des bandes passantes et atténuées sont dans l'ordre : $628.6 rad/s, 5903 rad/s, 7376 rad/s, 14790 rad/s$
   - Voir figure A.6 gauche. $1/s = 9.6 \rightarrow ordre 2$.
   - $H_N(p)$ Bessel : $H_N(p) = \frac{1}{0.618p^2+1.3613p+1}$.
   - $H(p) = H_N(\frac{B}{p/\omega_{centr}+\omega_{centr}/p}) = \frac{(p^2+\omega_{centr}^2)^2}{0.618\omega_{centr}^2 B^2 p^2+1.3616B(\omega_{centr}p^3+\omega_{centr}^3 p)+(p^2+\omega_{centr}^2)^2}$
7. $H(p)$ d'ordre 4 $\rightarrow$ $H(z)$ d'ordre 4 $\rightarrow$ 9 MAC. RIF de longueur 11 $\rightarrow$ 11 MAC. RII moins complexe.

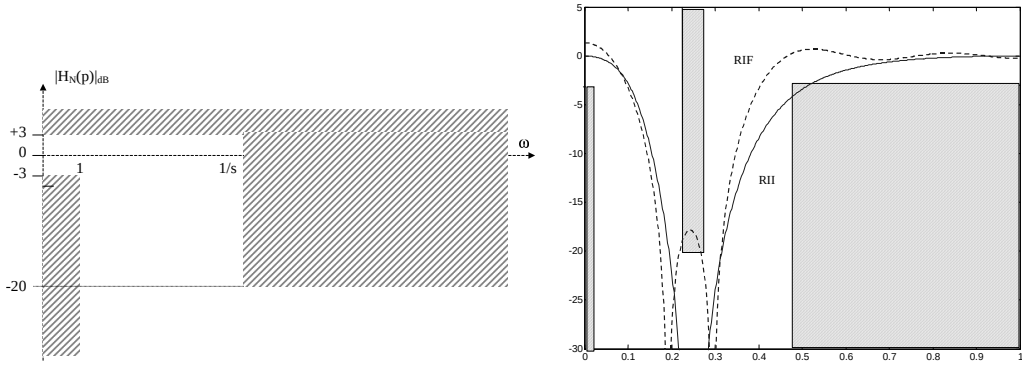


FIG. A.6: Gabarit et réponses fréquentielles des filtres RII et RIF

Problème 2 : Implantation d'un RII en virgule fixe (≈ 8 points)

1. Pas de quantification $q = 2^{-n}$, puissance du bruit $\sigma_b^2 = \frac{q^2}{12} = \frac{2^{-2n}}{12}$.

3. D'après les propriétés des filtres de Butterworth : le gain maximal est obtenu pour $\omega = 0$ et il est égal à 1. Donc nous obtenons $\max_\omega(|H(\omega)|) = 1$
4. Format des données :
- x : (16,0,15), y : (16,0,15)
- format de l'additionneur (32,0,31)
- a_i : (16,0,15), b_i : (16,0,15)

5. Deux sources de bruit lorsque nous sommes en double précision :
- quantification du signal d'entrée : $\sigma_{b_e}^2 = \frac{q^2}{12}$ avec $q = 2^{-15}$

– renvoi de la sortie de l'additionneur en mémoire $\sigma^2_{b_{ADD}} = \frac{q^2}{12}$ avec $q = 2^{-15}$

$$\sigma^2_{b_y} = \frac{q^2}{12}\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|H(e^{j\Omega})\right|^2 d\Omega + \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|H_D(e^{j\Omega})\right|^2 d\Omega\right) \tag{A.2}$$

avec

$$H_D(z) = \frac{1}{1 - 0.4z^{-1} + 0.2z^{-2}} \tag{A.3}$$

6. En simple précision nous avons 1 source de bruit en sortie de chaque multiplication :

$$\sigma^2_{b_y} = \frac{q^2}{12}\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|H(e^{j\Omega})\right|^2 d\Omega + \frac{5}{2\pi}\int_{-\pi}^{\pi}\left|H_D(e^{j\Omega})\right|^2 d\Omega\right) \tag{A.4}$$

8. Dynamique de la sortie des additionneurs ADD2 et ADD1
– $s_2(n) = -a_2 y(n-2) + b_2 x(n-2)$ d'ou $s_2 \in]-0.4, 0.4[$
– $s_1(n) = -a_1 y(n-1) + b_1 x(n-1) + s_2(n-1)$ d'ou $s_1 \in]-1.2, 1.2[$

9. Format des données :
– ADD_2 : (32,-1,32), s_2 :(16,-1,16), a_2 : (16,-1,16), b_2 : (16,-1,16)
– ADD_1 : (32,1,30), s_1 :(16,1,14), a_1 : (16,1,14), b_1 : (16,1,14), la donnée s_2 est décalée à droite de deux bits lors de son chargement dans l'accumulateur
– ADD_0 : (32,1,30), b_0 : (16,1,14), la sortie de $ADD0$ est décalée à gauche de 1 bit pour son renvoi en mémoire dans y.

10. La puissance du bruit dans cette structure est plus élevée pour deux raisons :
– la sortie de chaque additionneur est renvoyée en mémoire sur 16 bits
– la puissance de la source de bruit en sortie de ADD1 est plus élevée $\left(\sigma^2_{b_{ADD1}} = \frac{2^{-28}}{12}\right)$

Problème 3 : Zoom sur TFD (≈ 5 points)

1. $X_0(e^{j\Omega}) = X(e^{j(\Omega+\Omega_c)})$

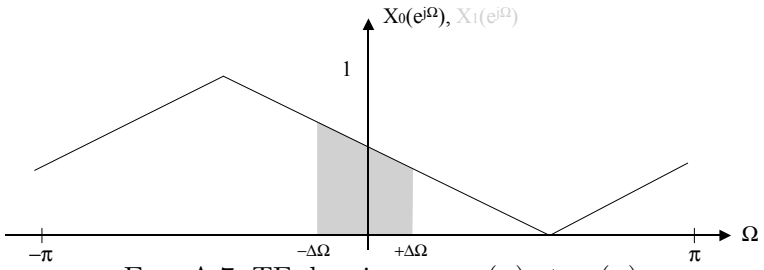


FIG. A.7: TF des signaux $x_0(n)$ et $x_1(n)$

2. Le signal $x_1(n)$ est composé de N points. Après décimation d'un facteur M, le signal $x_2(n)$ est composé de $O = N/M$ points. La valeur de M est limitée par le théorème de Shannon. La bande maximale du signal $x_2(n)$ est $[-\Delta\Omega, +\Delta\Omega]$. Par conséquent, la nouvelle valeur de fréquence d'échantillonnage après traitement peut être ramenée à : $F'_e = 2\Delta\Omega F_e/2\pi = 250kHz, M = \pi/\Delta\Omega = 4,0 = 64$.

3. $X_2(k) = \sum_{n=0}^{O-1} x_2(n)e^{-j2\pi kn/O}$. $X_Z(k) \sum_{n=0}^{O-1} x_2(n)e^{-j2\pi kn/P}$. $X_Z(k)$ est donc également la TFD de $x_2(n)$, mais possédant plus de points. Globalement, on obtient donc bien un zoom sur la partie souhaitée.

4. Complexité d'une FFT + N multiplications d'un réel par un complexe + un filtrage RIF à K coefficients + complexité d'une FFT.

A.6 Correction du DS de décembre 2000

Problème 1 : Synthèse de filtre RIF

1. Les fréquences de coupure valent 3.33kHz et 6.66kHz. La fréquence centrale vaut 5kHz.

2. $h(n) = 2\frac{\Omega}{\pi} sinc(n\Omega_c).cos(n\Omega_0)$ avec $\Omega_0 = \pi/2$ et $\Omega_c = \pi/6$.

n	0	1	2	3	4	5	6	7	8
$h(n)$	0.33333	0	-0.2756644	0	-0.1378322	0	0	0	-0.0689161

   Pour obtenir une phase linéaire il suffit de décaler $h(n)$ de α.

3. L'atténuation de 40dB implique une fenêtre de hamming, hanning ou blackman. Hamming est optimale dans notre cas. On a donc $\Delta\Omega = \pi/4 = 4\pi/N \Leftrightarrow N \geq 16$. On prendra donc $N = 17$. Soit $w(n)$ la fenêtre de Hamming, $h_1(n) = h(n-\alpha).w(n)$.

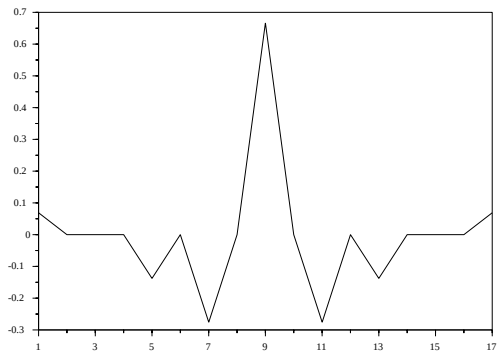
n	0	1	2	3	4	5	6	7	8
$h_1(n)$	0.333333	0	-0.2385239	0	0.0744294	0	0	0	-0.0055133

   Dans le cas d'un fenêtrage rectangulaire sur 9 points, l'atténuation vaut -21dB, mais la bande de transition vaut toujours $\pi/4$. La mise en cascade permet alors de retrouver les spécifications initiales de la question 3.

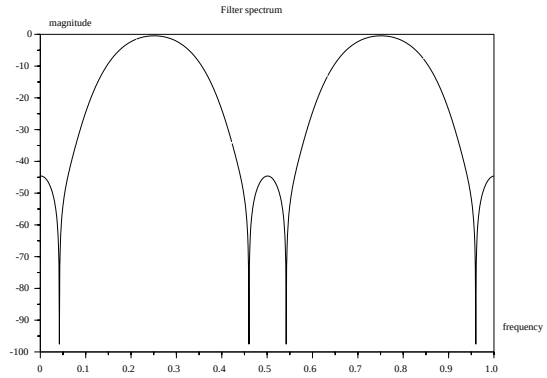
4. Le filtre h_2 possède 5 multiplications, tandis que h_1 en possède 7.

   - $\sigma^2_{h_2} = 5q^2/12 + q^2/12 \sum_n h_2^2(n) = 5.30q^2/12$
   - $\sigma^2_{h_2.h_2} = 5q^2/12 + 5.30q^2/12 \sum_n h_2^2(n) = 6.59q^2/12$
   - $\sigma^2_{h_1} = 7q^2/12 + q^2/12 \sum_n h_1^2(n) = 7.24q^2/12$

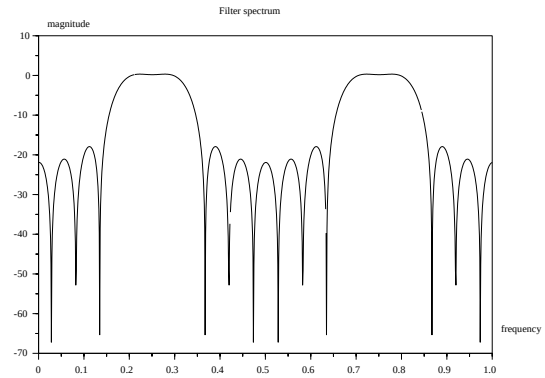
   La structure cascade est moins bruyante.



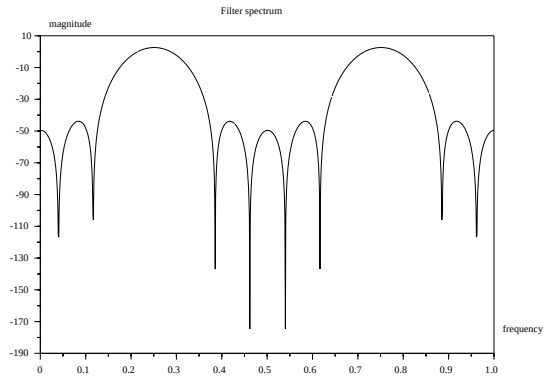
$h_1(n)$



$H_1(\Omega)$



$H_2(\Omega)$



$H_2(\Omega).H_2(\Omega)$

Problème 2 : Etude d'un filtre RII

1. $H(z) = \frac{1-b}{1-b.z^{-1}}$, $H(e^{j\Omega}) = \frac{1-b}{1-b.e^{-j\Omega}}$
2. $|H(0)| = 1$, $|H(\pi)| = \frac{1-b}{1+b}$. On a un filtre passe-bas.
3. $|H(e^{j\Omega})|^2 = \frac{(1-b)^2}{(1-b\cos\Omega)^2+b^2\sin^2\Omega} = \frac{(1-b)^2}{1+b^2-2b\cos\Omega}$
   On cherche Ω_c tel que $|H(e^{j\Omega_c})|^2 = \frac{1}{2}$.
   $\Rightarrow cos\Omega_c = \frac{1-4b+b^2}{-2b}$
4. $\Omega_c = 2\pi f_c/f_e = \pi/4 \Rightarrow b = 0.4733977$
5. Sur 4 bits : $b = 0.5 : 0,100 \Rightarrow cos\Omega_c = 0.75 \Leftrightarrow \Omega_c = 0.7227342 \Leftrightarrow f_c = 920.2$
   Sur 8 bits : $b = 0.4765625 : 0,0111101 \Rightarrow cos\Omega_c = 0.7125384 \Leftrightarrow \Omega_c = 0.7776869 \Leftrightarrow f_c = 990.2$
6. Le filtre ne débordera pas car la somme des coefficients ($1-b$ et b) vaut 1.

Problème 3 : Algorithme de Bluestein

1. $e(n) = x(n).e^{-j\pi n^2/N}$, $s(n) = \sum_{k=0}^{N-1} e(k).h(n-k) = \sum_{k=0}^{N-1} x(k).e^{-j\pi k^2/N}.e^{j\pi(n-k)^2/N}$
   $s(n) = \sum_{k=0}^{N-1} x(k).e^{-2j\pi n.k/N}.e^{j\pi n^2/N}$
   $y(n) = \sum_{k=0}^{N-1} x(k).e^{-2j\pi n.k/N} = TFD[x(n)]$
2. $(2\otimes) + (4\otimes + 2N \oplus + 2(N-1)\oplus) + (4\otimes + 2\oplus) : (4N+6)\otimes$ et $4N\oplus$

3. Dans le cas d'un flot continu, cette TFD glissante évite le retard de N points du traitement par blocs de la TFD classique.

Problème 4 : Conception d'un filtre RII

1. $|H(j\omega)| = 1$, $Arg[H(j\omega)] = -2arctg(\omega)$. Il s'agit d'un filtre déphaseur pur. La phase vaut $-\pi/2$ en $\omega = 1$.
2. $H_b(z) = \frac{-3+5.z^{-1}}{5-3.z^{-1}} = \frac{T-2+(T+2).z^{-1}}{T+2+(T-2).z^{-1}}$
3. $H_b(e^{j\Omega}) = \frac{-3+5.e^{-j\Omega}}{5-3.e^{-j\Omega}}$
4. $|H_b(e^{j\Omega})| = 1$, $H_b(0) = 1$, $Arg[H_b(0)] = 0$, $H_b(\pi) = -1$, $Arg[H_b(\pi)] = -\pi$
5. Il faut utiliser la formule de distorsion bilinéaire. La nouvelle valeur de pulsation ω' pour laquelle la phase vaut $-\pi/2$ est donnée par $\omega'.T/2 = arctg(\omega.T/2)$, $\omega' = 4.arctg(0.5/2) = 0.9799147$.

A.7 Correction du DS de janvier 2000

Problème 1 : Synthèse de filtre RIF (8 points)

1. Attention, le pseudo-module et la phase sont sans discontinuité.
2. type 3 : RI antisymétrique, N impair.
3. $h(n) = \frac{(-1)^n}{n\pi} - \frac{sin(n\Omega_c)}{n^2\pi\Omega_c}$
   $h(0) = 0$
   h(n) pour $n = -4... + 4$ = [-0.0796 0.0836 -0.1592 0.521 0 -0.521 0.1592 -0.0836 0.0796].
4. Décalage de $h(n)$ de $N - 1/2$.
5. $A_a(e^{j\Omega}) = 2.[h_a(0).sin3\Omega + h_a(1).sin2\Omega + h_a(2).sin\Omega]$
   $\Phi_a(\Omega) = \pi/2 - 3\Omega$
6. moins d'oscillation en bande passante mais bande de transition plus étroite.

Problème 2 : Etude d'un filtre RIF (8 points)

1. $H(z) = b_0 + b_1z^{-1} + b_2z^{-2}$,$H(e^{j\Omega}) = b_0 + b_1e^{-j\Omega} + b_2e^{-2j\Omega}$
2. $b_0 = b_1 = b_2 = 1/3$
3. $H(\Omega) = 1/3e^{-j\Omega}(1 + 2cos(\Omega))$, $H(2\pi/3) = 0$
4. Les coefficients b_i valent 0.375, H(0) = 1.125
5. Voir cours
6. $2q^2/12 + \sigma_e^2(1 + b_1^2 + b_2^2) = 0,268q^2$
7. $f_0(n) = x(n)$, $g_0(n) = x(n)$
   $f_1(n) = f_0(n) + \rho_1.g_0(n-1)$, $g_1(n) = g_0(n-1) + \rho_1.f_0(n)$
   $f_2(n) = x(n) + \rho_1(1 + \rho_2).x(n-1) + \rho_2.x(n-2)$
8. $\rho_2 = b_2$, $\rho_1 = b_1/(1 + b_2)$, $\frac{G_2(z)}{X(z)} = z^{-2}.H(z^{-1})$
9. $\sigma_{f_i}^2 = \sigma_{f_{i-1}}^2 + \sigma_{g_{i-1}}^2.\rho_i^2 + q^2/12$
10. $\sigma_{f_1}^2 = \frac{q^2}{12} + \frac{q^2}{12}.\rho_1^2 + \frac{q^2}{12} = \sigma_{g_1}^2 = 2,06\frac{q^2}{12}$
    $\sigma_{f_2}^2 = \sigma_{f_1}^2 + \sigma_{f_1}^2.\rho_2^2 + \frac{q^2}{12} = \sigma_{g_1}^2 = 3,29\frac{q^2}{12}$

11. Structure directe : N multiplications, N-1 additions, N cycles, $Tcycle < 49ns$
Structure treillis : 2N multiplications, 2N additions, 2N cycles, $Tcycle < 24ns$

Problème 3 : Analyse spectrale (4 points)

1. $W(\Omega) = e^{-j\Omega(N-1)/2} \frac{sin(\Omega N/2)}{sin(\Omega/2)}$
2. $-1/128$
3. Une raie en k=6 d'amplitude 1/2 et une raie d'amplitude 0.001/2 en k=56, complétées par leur symétrique par rapport à $N/2$.
4. Si k_0 n'est plus entier, on trouve des résidus des lobes de la fenêtre qui viennent masquer la raie de faible amplitude.
5. Hamming évidemment. L'écart minimum est la largeur d'un lobe soit dans ce cas $8\pi/N$.

A.8 Correction du DS de mars 1999

Correction du Problème 1

1. **Synthèse de filtre RII par la méthode bilinéaire** Le gabarit du filtre passe haut doit être recalculé afin de tenir compte de la prédistorsion. f_c et f_a doivent être modifiées par $f' = \frac{tg(\pi.f.T)}{\pi.T}$, et donnent respectivement 1034 Hz et 3183 Hz. Le filtre prototype équivalent est un passe-bas de sélectivité 3.08, ce qui donne un filtre du deuxième ordre.
2. Passe-haut normalisé $H_n(p)$, la dénormalisation s'effectue par rapport à $w'_c = 20000rad/s$, $w'_c.T = 2$

$$\begin{aligned} H_n(p) &= \frac{p^2}{p^2 + \sqrt{2}p + 1} \\ H(p) &= \frac{p^2}{p^2 + \sqrt{2}p.w'_c + {w'_c}^2} \\ H(z) &= \frac{4(1-z^{-1})^2}{(4 + 2\sqrt{2}w'_c.T + 4) + (4 - 2\sqrt{2}w'_c.T + 4)z^{-2}} \\ H(z) &= \frac{1 - 2z^{-1} + z^{-2}}{3.414 + 0.586.z^{-2}} \\ H(z) &= \frac{0.29(1 - 2z^{-1} + z^{-2})}{1 + 0.171z^{-2}} \end{aligned}$$

3. $y(n) = 0.29x(n) - 0.58x(n-1) + 0.29x(n-2) - 0.17y(n-2)$
4. 4 multiplications, 3 additions, 2 cases mémoire.
5. **Etude des bruits de calcul du filtre RII**
   – Les multiplications sont sources de bruit. Certaines sont injectées en entrée, d'autres directement sur la sortie.

$$P_s = 3\frac{q^2}{12} + 2\frac{q^2}{12}\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\Omega})|^2 d\Omega$$

– Ici les bruits sont générés par les mémorisations. Le bruit est donc généré avant les registres (en sortie du premier additionneur) et peut être ramener sur l'entrée. Un bruit est également généré sur la sortie (du au filtrage RIF).

$$P'_s = \frac{q^2}{12} + 2.\frac{q^2}{12}.\frac{1}{2\pi}\int_{-\pi}^{\pi}|H(e^{j\Omega})|^2 d\Omega$$

6. **Application numérique** Dans ce cas $\frac{1}{2\pi}\int_{-\pi}^{\pi}|H(e^{j\Omega})|^2 d\Omega \approx 1/2$

$$P_s = \frac{q^2}{3},\ P'_s = 2.02\frac{q^2}{12}$$

7. **Synthèse de filtre RIF par fenêtrage**

$$h(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} d\Omega = 1 - \Omega_c/\pi = 1/2$$

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi}|H(e^{j\Omega})|^2 d\Omega = -\frac{sinn\Omega_c}{n\pi} = -\frac{1}{2}sinc(n\pi/2)$$

n	0	1	2	3	4
h(n)	1/2	-0.32	0	0.106	0

8. Une fenêtre rectangulaire est suffisante (en théorie). Sa sélectivité vaut $\Delta f = f_e/N = 1500Hz \rightarrow N = 7$.

$$H_a(z) = 0.106 - 0.32z^{-2} + 0.5z^{-3} - 0.32z^{-4} + 0.106z^{-6}$$

9. $y(n) = 0.106x(n) - 0.32x(n-2) + 0.5x(n-3) - 0.32x(n-4) + 0.106x(n-6)$
10. 5 multiplications, 4 additions, 6 cases mémoire.
11. **Etude des bruits de calcul du filtre RIF**
    – $P_s = 5\frac{q^2}{12}$
    – $P_s = \frac{q^2}{12}$
12.

$$H_{bi}(e^{j\Omega}) = \frac{0.29(1 - e^{-j\Omega})^2}{1 + 0.17e^{-j2\Omega}}$$

$$|H_{bi}(e^{j\Omega})|^2 = \frac{|0.29(2sin(\Omega/2)|^2}{(1 + 0.17cos(2\Omega))^2 - sin^2(2\Omega)}$$

$$H_a(e^{j\Omega}) = e^{-3j\Omega}\,(0.5 - 0.64cos\Omega + 0.212cos3\Omega)$$

13. RII est moins complexe à sélectivité équivalente, et, dans ce cas, moins bruyant. Le RIF possède une phase linéaire et son implémentation en double précision est très efficace.
14. Les deux filtres doivent être limités en entrée par une division par 2.

Correction du Problème 2

1. N^2 multiplications et$N(N-1)$ additions
2. $$X(k) = \sum_{n=0}^{N-1} x(n).e^{-2j\pi\frac{n.k}{N}} = \sum_{n=0}^{N-1} x(n).\left[cos\left(\frac{2\pi n.k}{N}\right) - j.sin\left(\frac{2\pi n.k}{N}\right)\right]$$

   Il suffit de développer le cosinus de la formule de $X_C(k)$ pour obtenir :

   $$X_C(k) = cos\left(\frac{2\pi k}{4N}\right) Re[X(k)] + sin\left(\frac{2\pi k}{4N}\right) Im[X(k)]$$

3. La version rapide consiste à effectuer une FFT, puis multiplier respectivement les parties réelles et imaginaires d'indice n par $cos(\frac{2\pi k}{4N})$ et $sin(\frac{2\pi k}{4N})$, et enfin additionner l'ensemble. La complexité est maintenant : $2Nlog_2(N) + 2N$ multiplications et$3Nlog_2(N) + N$ additions

Annexe B

Abaques de filtrage analogique