

وزارة التربية الوطنية
MINISTRE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT

Electronique

المدرسة الوطنية المتعددة التخصصات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

*Etude et simulation
des filtres numériques RIF
implémentés par la méthode
de fenêtrage*

Proposé par :

M^r B. DERRAS

Etudié par :

*M^{lle} M. ZAIT
M^{lle} S. CHOUAÏ*

Dirigé par

M^r B. DERRAS

PROMOTION

juillet 1993

ERRATUM

Titre : lire " ... par la méthode de fenêtrage "
au lieu " ... pour la méthode de fenêtrage ".

Page 7: équation (I.22), lire : $E = \int_{-\infty}^{+\infty} |f(t)|^2 dt$.

Page 71: lire " Filtre multi-passe bandes avec bande à l'origine "
au lieu de " Filtre de mise en forme des impulsions ".

وزارة التربية الوطنية
MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT *Electronique*



PROJET DE FIN D'ETUDES

SUJET

*Etude et simulation
des filtres numériques RIF
synthétisés pour la méthode
de fenêtrage*

Proposé par :

M^r B. DERRAS

Etudié par :

*M^{lle} M. ZAIT
M^{lle} S. CHOUAI*

Dirigé par

M^r B. DERRAS

PROMOTION

juillet 1993

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

A mes parents,

A mes amis,

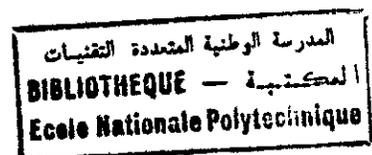
A tous ceux qui me sont chers

Z.Malika

A mes chers parents,

A tous ceux que j'estime

S.Sihem



A monsieur B.DERRAS.

Qu'il trouve ici l'expression de notre gratitude et notre profond respect.

Pour toute l'aide qu'il nous a prodiguée,

Pour sa large disponibilité, et,

Pour l'intérêt qu'il a manifesté à l'égard de notre travail,

Nous lui exprimons également notre reconnaissance et nos remerciements.

Title: Study and simulation of FIR digital filters, designed by the windowing method

Abstract : the purpose of this work consists of a simulation of FIR digital filters. To reach that goal we have at first studied and then designed some filters such as bands selectors filters, differentiator, Hilbert transformer and wave form shaping filters, by the windowing method. The coefficients of the impulse response of the filter that are computed with infinite precision are quantized, and then an optimization method is used to improve the performance of the quantized filter. At last, some signals such as white noise and sinusoids with different frequencies are used to simulate some filters in their non quantized and quantized forms. We can conclude about the quality of filtering by observing the output signals.

Titre: Etude et Simulation des filtres numériques R.I.F synthétisés par la méthode de fenêtrage.

Résumé: Le but essentiel de ce travail consiste à simuler des filtres numériques R.I.F. Pour y parvenir, nous avons effectué une étude, ensuite une synthèse de quelques filtres tels que les filtres sélecteurs de bandes, les différentiateurs, le transformateur de Hilbert et les filtres de mise en forme par la méthode de fenêtrage temporel. Les coefficients de la réponse impulsionnelle du filtre évalués en précision infinie, sont ensuite quantifiés puis optimisés afin d'améliorer les performances du filtre.

L'étape de simulation consiste à appliquer quelques signaux tels que des composantes sinusoidales, un bruit blanc ou une combinaison de ces signaux à l'entrée du filtre sous sa forme non quantifiée et quantifiée. L'observation du signal de sortie nous permet de conclure quant à la qualité du filtrage.

- العنوان : دراسة ومحاكاة المرشحات العددية ذات الاستجابة
الدفعية المنتهية، المركبة بطريقة العنصر بالترافذ.
- ملخص : الهدف الرئيسي من هذا العمل يتمثل في محاكاة
المرشحات العددية ذات الاستجابة الدفعية المنتهية للتحويل
إلى ذلك، قمنا بدراسة ثم بتركيب بعض المرشحات المبرزة لأشكال
ترددية معينة، والمرشحات التفاضلية، ومحولة هيلبرت، وكذلك
مرشحات التشكيل الموجي وذلك بواسطة التحويل الزمني وبغدها قمنا
بتكميم معاملات الاستجابة الدفعية ذات الدقة اللامتناهية،
وتجويدها بهدف تحسين خصائص وأداء المرشح.
وبتطبيق إشارات جيبية متعددة ومنجزة بيضاء على مدخل المرشح
وبملاحظة الإشارات عند مخرجه يمكننا استخلاص نوعية
الترشيح المستعمل.

SOMMAIRE

INTRODUCTION.

1

CHAPITRE I : GENERALITES SUR LES TRANSFORMEES, LES SYSTEMES ET LES FILTRES

I.1. INTRODUCTION.	
I.2. PROPRIETES DES OUTILS MATHÉMATIQUES	3
I.2.1. Linéarité et invariance temporelle d'un système.	3
I.2.2. Causalité et stabilité.	3
I.2.3. Convolution	3
I.2.4. Transformée de Fourier.	4
I.2.5. Transformée en Z	4
I.2.6. Energie d'un signal analogique.	6
I.3. FILTRAGE ANALOGIQUE.	7
I.4. FILTRAGE NUMÉRIQUE.	7
I.5. PASSAGE DU PLAN S AU PLAN Z.	8
I.5.1. Invariance impulsionnelle.	8
I.5.2. transformation bilinéaire.	9
I.6. COMPARAISON ENTRE UN FILTRE ANALOGIQUE ET UN FILTRE NUMÉRIQUE.	10
	11

CHAPITRE II : FILTRES NUMÉRIQUES.

II.1. INTRODUCTION.	
II.2. CLASSIFICATION DES FILTRES NUMÉRIQUES.	12
II.2.1. Classification selon la réponse impulsionnelle.	12
II.2.2. Classification selon la réponse fréquentielle.	12
II.3. COMPARAISON ENTRE LES FILTRES R.L.I ET R.I.F.	13
	16

CHAPITRE III : SYNTHÈSE DES FILTRES R.I.F A PHASE LINEAIRE.

III.1. INTRODUCTION.	
III.2. FILTRES R.I.F A PHASE LINEAIRE.	17
III.3. METHODE DE SYNTHÈSE DES FILTRES R.I.F A PHASE LINEAIRE.	17
III.3.1. Méthode d'échantillonnage en fréquence.	18
III.3.2. méthode du minimax.	19
III.3.3. Méthode de fenêtrage temporel.	19
III.4. SYNTHÈSE DES FILTRES NUMÉRIQUES R.I.F PAR LA METHODE DE FENETRAGE.	20
III.4.1. Présentation de la méthode.	21
III.4.2. Propriétés des fenêtres.	21
III.4.3. Exemples de quelques fenêtres.	25
III.4.4. Elaboration d'un programme et résultats.	31
III.5. CONCLUSION.	75

CHAPITRE IV : STRUCTURE DE REALISATION DES FILTRES NON RECURSIFS.	76
IV.1. INTRODUCTION.	76
IV.2. STRUCTURE DIRECTE ET STRUCTURE TRANSPOSEE.	77
IV.3. STRUCTURE EN CASCADE.	78
IV.4. STRUCTURE EN TREILLIS..	80
IV.5. STRUCTURE DE REALISATION PAR F.F.T.	81
IV.6. CONCLUSION.	
CHAPITRE V : OPTIMISATION DES FILTRES NUMERIQUES R.L.F A PHASE LINEAIRE ET COEFFICIENTS DE LONGUEUR FINIE PAR UNE METHODE DE RECHERCHE LOCALE	82
V.1. INTRODUCTION.	83
V.2. QUANTIFICATION.	83
V.2.1. Types de représentation des nombres binaires.	84
V.2.2. Quantification par l'arrondi.	84
V.2.3. quantification par troncature.	86
V.3. EFFET DE LA QUANTIFICATION SUR LA REPONSE FREQUENTIELLE DU FILTRE.	94
V.4. METHODE DE LA RECHERCHE LOCALE.	96
V.5. RESULTATS ET INTERPRETATION.	98
V.6. APPROCHE STATISTIQUES POUR L'ESTIMATION DE LA LONGUEUR DES MOTS.	101
V.7. CONCLUSION.	
	102
CHAPITRE VI: SIMULATION DES FILTRES R.L.F A PHASE LINEAIRE	102
VI.1. INTRODUCTION.	103
VI.2. EFFET DES ERREURS DE CALCUL SUR LES CARACTERISTIQUES DU FILTRE.	103
VI.3. TYPES DE SIGNAUX APPLIQUES A L'ENTREE DU FILTRE SIMULE.	
VI.4. ELABORATION D'UN PROGRAMME ET RESULTATS.	
VI.4.1. Description.	103
VI.4.2. Exemples.	104
VI.4.3. Résultats et interprétation.	105
VI.5. CONCLUSION.	117
CONCLUSION.	118
BIBLIOGRAPHIE.	
ANNEXE 1.	
ANNEXE 2.	
ANNEXE 3	

INTRODUCTION

Un filtre est un système, qui, excité par un signal d'entrée, produit un signal de sortie dont la distribution fréquentielle diffère de celle du signal d'excitation.

Selon la nature continue ou échantillonnée du signal à traiter, et selon la nature du traitement, on distingue deux catégories de filtres : les filtres analogiques et les filtres numériques.

Un filtre est caractérisé dans le domaine fréquentiel par sa fonction de transfert, et dans le domaine temporel par sa réponse impulsionnelle, qui en fait, est la transformée de Fourier inverse de la fonction de transfert. Les filtres numériques ont été développés et étudiés dans le but de pouvoir simuler les filtres analogiques sur ordinateur. Ceci a permis de vérifier les performances et d'optimiser les paramètres de ces filtres avant leur éventuelle réalisation. Le développement de la technologie des circuits intégrés numériques augmentent l'intérêt économique des filtres numériques. C'est pourquoi, en complément aux résultats bien établis pour la simulation, les méthodes propres pour la synthèse des filtres numériques ont été développées.

Lorsque le nombre de coefficients de la réponse impulsionnelle du filtre numérique est infini, il est dit à réponse impulsionnelle infinie (R.I.I), sinon on parle de filtre à réponse impulsionnelle finie (R.I.F). L'un des principaux avantages des filtres R.I.F est qu'ils peuvent donner des phases exactement linéaires.

Le présent travail consiste en l'étude, la synthèse et la simulation des filtres numériques R.I.F à phase linéaire. Pour entamer ce travail, il nous a semblé nécessaire d'introduire les différents outils mathématiques dont on fera usage. Suite à cela, une étude comparative entre le filtrage analogique et le filtrage numérique sera faite (chapitre I). L'accent sera mis sur les filtres numériques.

Dans le cadre des filtres numériques, un bref aperçu sur deux méthodes de classification sera présenté dans le chapitre II. La classification selon la réponse fréquentielle et la classification selon la réponse impulsionnelle pour laquelle on distingue les filtres R.I.I et les filtres R.I.F. Les avantages et les inconvénients de ces deux types de filtres feront également objet du chapitre II.

Nous parlerons après cela, dans le chapitre III, et de manière succincte, sur quelques méthodes de synthèse des filtres numériques R.I.F à phase linéaire. On détaillera la méthode de fenêtrage temporel, qui sera appliquée pour synthétiser quelques filtres dont nous citons les filtres sélecteurs de bandes (passe-bas, passe-haut, etc...), le transformateur de Hilbert, le différentiateur et le filtre de mise en forme.

Dans un premier temps, les coefficients du filtre seront calculés avec une très grande précision (celle de l'ordinateur) ou une précision infinie, nous procéderons ensuite (chapitre V) à une quantification puis à une optimisation des coefficients. La méthode utilisée à cet effet est celle de la recherche locale.

Le chapitre IV portera sur les différentes structures de réalisation des filtres R.I.F.

Pour terminer cette étude, l'étape de simulation (chapitre VI) consistera à appliquer quelques signaux aux filtres synthétisés avec une précision infinie puis avec une précision finie (après optimisation).

Enfin, une conclusion sur tout ce qui a été fait clôturera notre travail.

CHAPITRE I

GENERALITES SUR LES TRANSFORMEES LES SYSTEMES ET LES FILTRES

I.1. INTRODUCTION :

Dans le présent chapitre seront exposés, un bref rappel sur les principaux outils mathématiques dont, il a été fait usage durant cette étude; ainsi que des notions sur le filtrage analogique et numérique, suivis de définitions sur les différentes transformations permettant le passage du plan S au plan Z, et enfin d'une étude comparative mettant l'accent sur l'importance du filtrage numérique.

I.2. PROPRIETES DES OUTILS MATHÉMATIQUES :

I.2.1. Linéarité et invariance temporelle d'un système [1] :

Soit un espace de signaux S, nous appellerons "Système linéaire et invariant dans le temps", un opérateur L réalisant une application de S dans S, tel que :

- Linéarité : $\forall x_1(t), x_2(t) \in S \text{ et } \forall \alpha, \beta \in \mathbb{C}$

$$L[\alpha \cdot x_1(t) + \beta \cdot x_2(t)] = \alpha \cdot L[x_1(t)] + \beta \cdot L[x_2(t)] \quad (I.1)$$

La relation (I.1) montre qu'un système linéaire traite une somme d'excitations comme si elles étaient traitées séparément puis additionnées.

- Invariance temporelle :

$$\forall x(t), y(t) \in S \text{ et } \forall \tau \in \mathbb{R}$$

$$\text{on a : } L[x(t - \tau)] = y(t - \tau) \quad (I.2)$$

Cela veut dire qu'une excitation spécifique produira toujours la même réponse indépendamment du temps d'application.

On dira alors d'un système qu'il est invariant dans le temps, si les paramètres internes ne changent pas avec le temps.

I.2.2. Causalité et stabilité [2] :

Un système physiquement réalisable doit être causal et stable.

- Causalité :

Un système est dit causal si pour tout signal d'entrée $x(t)$, vérifiant $x(t) = 0$ pour $t < 0$; la sortie doit vérifier $y(t) = 0$ pour $t < 0$.

Pour un tel système, la réponse à un instant donné ne peut précéder l'excitation. L'opérateur L est contraint de ne pas dépendre des valeurs futures de celle-ci.

- Stabilité :

A toute entrée bornée, un système stable associe une sortie bornée, soit :

$$\text{Si } |x(t)| < M \forall t \text{ alors } |y(t)| < N \forall t \quad (\text{I.3})$$

Dans ce cas, le système est dit stable au sens BIBO (Banded Input - Banded Output).

I.2.3. Convolution [3],[4] :

La propriété d'invariance vue précédemment (I.2) permet une formulation particulièrement simple de la réponse impulsionnelle d'un système linéaire. On peut donc représenter un système linéaire invariant dans le temps par la relation suivante :

$$y(t) = \int_{-\infty}^{+\infty} h(\tau) x(t - \tau) d\tau \quad (\text{I.4})$$

La réponse impulsionnelle $h(t)$ caractérise entièrement ce système.

$y(t)$ étant la réponse à l'excitation $x(t)$.

La notation usuelle représentant un produit de convolution est la suivante :

$$y(t) = h(t) * x(t) \quad (\text{I.5})$$

Par symétrie, on a ainsi :

$$y(t) = x(t) * h(t) \quad (\text{I.6})$$

I.2.4. Transformée de Fourier :

L'analyse de Fourier occupe une place privilégiée dans la théorie et le traitement des signaux.

La transformée de Fourier est un outil efficace et d'utilisation facile pour la résolution de très nombreux problèmes en traitement des signaux.

- Définition :

Soit $f(t)$ une fonction de la variable t ; ainsi, on démontre que sous certaines conditions on a [1] :

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) e^{j\omega t} d\omega \quad (\text{I.7})$$

$$\text{avec : } F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt \quad (\text{I.8})$$

La fonction $F(\omega)$ est la transformée de Fourier de $f(t)$.

On peut l'écrire aussi :

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t} dt = A(\omega) e^{j\Phi(\omega)} \quad (\text{I.9})$$

On appelle : $A(\omega)$: Le spectre d'amplitude.

$A^2(\omega)$: Le spectre d'énergie.

$\Phi(\omega)$: Le spectre de phase.

La transformée de Fourier de $f(t)$ existe si $f(t)$ est absolument intégrable; c'est à dire :

$$\int_{-\infty}^{+\infty} |f(t)| \cdot dt < \infty \quad (\text{I.10})$$

- Propriété :

La propriété fondamentale de la transformée de Fourier est qu'elle transforme un produit de convolution dans un domaine (temporel ou fréquentiel), en un simple produit dans l'autre domaine (fréquentiel ou temporel).

- Convolution temporelle :

Soient $f_1(t)$ et $f_2(t)$ deux signaux à énergie finie, on a :

$$f_1(t) * f_2(t) \xrightarrow{\text{TF}} F_1(\omega) \cdot F_2(\omega) \quad (\text{I.11})$$

- Convolution fréquentielle :

$$f_1(t) \cdot f_2(t) \xrightarrow{\text{TF}} F_1(\omega) * F_2(\omega) \cdot \frac{1}{2\pi} \quad (\text{I.12})$$

- TFD :

La transformée de Fourier discrète offre la possibilité de calculer la transformée de Fourier continue sur calculateur numérique.

C'est essentiellement, grâce à cet outil que la synthèse et la simulation des filtres numériques est possible.

En fait, un algorithme très rapide a été développé [1],[5], permettant le calcul de cette transformée d'une manière efficace. Cet algorithme est appelé : La transformée de Fourier rapide ou fast Fourier transform (FFT).

a - Définition :

On appelle transformée de Fourier discrète d'une séquence $x(k)$ de N valeurs, la séquence :

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j\frac{2\pi}{N}kn} \quad ; 0 \leq k \leq N-1 \quad (\text{I.13})$$

$X(n)$ et $x(k)$ pouvant être complexes.

La transformée de Fourier inverse sera :

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) e^{j\frac{2\pi}{N}kn} \quad (\text{I.14})$$

b - Représentation matricielle de la TFD :

Posons : $W_N = e^{j\frac{2\pi}{N}}$ (I.15)

En substituant $W_N = e^{j\frac{2\pi}{N}}$ dans l'équation (I.13), on obtient :

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot W_N^{-nk} \quad (\text{I.16a})$$

$$X(n) = x(0) + x(1) \cdot W_N^{-n} + x(2) \cdot W_N^{-2n} + \dots + x(N-1) \cdot W_N^{-(N-1)n} \quad (I.16b)$$

D'où la représentation matricielle suivante de la TFD :

$$X = W \cdot x \quad (I.17a)$$

Où X et x sont des vecteurs de dimension N et W est une matrice carrée, $W(n,k) = W_N^{-nk}$.
Pour la transformée inverse, on obtient :

$$x = W^{-1} \cdot X \quad (I.17b)$$

Cette représentation matricielle est très utile pour développer l'algorithme FFT [5] et pour déduire certaines propriétés de la TFD.

Il est à noter que le calcul de la TFD nécessite N^2 opérations lorsqu'on utilise la forme directe, alors que ce même calcul ne nécessite que $N \cdot \log_2 N$ opérations lorsqu'on utilise l'algorithme FFT, avec $N = 2^v$.

1.2.5. La transformée en Z :

La transformée en Z joue dans le domaine numérique le même rôle que celui de la transformée de Laplace dans le domaine analogique [3],[4].

- Définition :

L'analyse des systèmes discrets peut s'effectuer grâce à la transformée en Z .

Soit $x(k)$ un signal discret défini pour $0 \leq k < \infty$, sa transformée en Z est définie par :

$$Z[x(k)] = X(Z) = \sum_{k=0}^{\infty} x(k) Z^{-k} \quad (I.18)$$

Cette transformée est appelée aussi la T.Z unilatérale, avec Z une variable complexe (Nb: La notation T.Z veut dire transformée en Z).

- Fonction de transfert :

On sait que le signal de sortie d'un système linéaire invariant est donné par le produit de convolution :

$$y(k) = \sum_{n=-\infty}^{+\infty} x(n) \cdot h(k-n) \quad (I.19)$$

Où $x(k)$ est le signal d'entrée et $h(k)$ est la réponse impulsionnelle du système.

La T.Z. de cette expression est :

$$Y(Z) = H(Z) \cdot X(Z) \quad (I.20)$$

La fonction $H(Z)$ est appelée " fonction de transfert ".

Si le système est de plus causal, on a :

$$H(Z) = \sum_{k=0}^{+\infty} h(k) \cdot Z^{-k} \quad (I.21)$$

I.2.6. Energie d'un signal analogique :

Soit un signal $f(t)$ à énergie finie, sa transformée de Fourier existe et est donnée par :

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

Avec

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

L'énergie totale E s'exprime dans le domaine fréquentiel ou dans le domaine temporel :

$$E = \int_{-\infty}^{+\infty} |f^2(t)| dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |F(\omega)|^2 dt \quad (I.22)$$

Cette équivalence s'appelle l'égalité de Parseval.

Si l'énergie d'un signal $x(t)$ est infinie et si sa puissance moyenne totale est finie, alors $x(t)$ est dit à puissance moyenne finie.

$$P = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-T/2}^{+T/2} \frac{|X_T(\omega)|^2}{T} d\omega \quad (I.23)$$

Avec
$$X_T(\omega) = T.F(x_T(t)) \text{ et } x_T(t) = \begin{cases} x(t) & |t| < T/2 \\ 0 & \text{ailleurs} \end{cases}$$

Les signaux à énergie finie sont généralement des signaux transitoires de courte durée.

Les signaux à puissance moyenne finie sont généralement des signaux périodiques, quasi-périodiques et aléatoires permanents [3].

I.3. FILTRAGE ANALOGIQUE :

Dans le domaine analogique, le rôle d'un filtre consiste à transmettre sans déformation toutes les composantes utiles d'un signal analogique (courants, tensions), et à éliminer toutes celles qui n'appartiennent pas à sa bande passante.

Le filtrage analogique demeure un élément indispensable malgré le développement du filtrage numérique. Il représente un guide dans la synthèse des filtres numériques.

Les filtres analogiques sont caractérisés par leur :

- Fonction de transfert :

Elle est déterminée par le rapport, en fonction de la fréquence, d'une grandeur de sortie $Y(j\omega)$ à une grandeur d'entrée $X(j\omega)$

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} \quad (I.24)$$

$$H(j\omega) = A(\omega) \cdot e^{j\phi(\omega)} \quad (I.25)$$

- Le Gabarit :

C'est une représentation graphique des conditions limites amplitude-fréquence, nécessaire pour réaliser un filtrage donné; il délimite la bande de fréquence à l'intérieur de laquelle le gain doit être maintenu quasi-constant, c'est à dire avec une marge tolérable.

Il donne les limites entre lesquelles, le gain peut fluctuer, ainsi que les fréquences au delà desquelles le signal devra subir une atténuation spécifique.

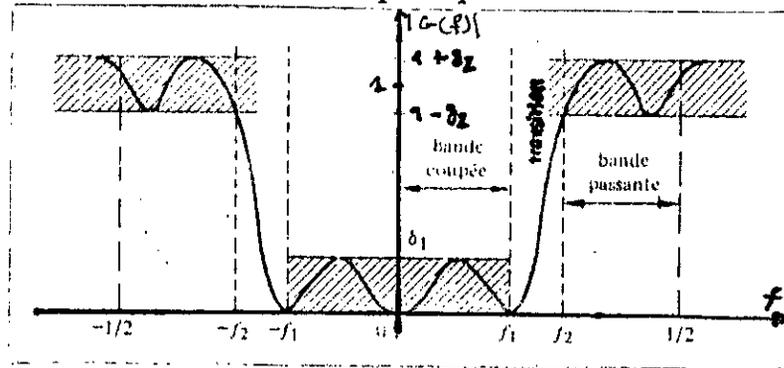


Fig I.1

La définition énoncée au début de la section (I.3) est relative à un filtre idéal, qui n'existe pas. En réalité, on approxime ces derniers par des filtres réels.

Plusieurs filtres peuvent être ainsi synthétisés par des techniques d'approximations [3].

I.4. FILTRAGE NUMÉRIQUE

Le filtrage numérique résoud très souvent, le problème de mise en forme d'un signal, c'est l'opération de traitement qui consiste à modifier la distribution fréquentielle des composantes d'un signal, à l'aide d'un système numérique.

Par rapport au filtrage analogique qu'il remplace dans bien des domaines, le filtre numérique possède, des avantages que l'on pourrait résumer comme suit :

- Souplesse de sa mise en forme.

La réponse en fréquence peut être modifiée en changeant les coefficients.

- Sa stabilité dans le temps, pas de vieillissement des composants, pas d'influence de la température.

- Se prête à la miniaturisation, même aux très basses fréquences, ce qui n'est pas le cas des filtres analogiques, utilisant des composants et circuits bien encombrants [3] [6].

I.5. PASSAGE DU PLAN S AU PLAN Z

Pour pouvoir tirer partie des méthodes de synthèse des filtres analogiques déjà établies, en les transposant au cas des filtres numériques, il faut pouvoir établir une correspondance entre le plan S de la transformée de Laplace et le plan Z de la transformée en Z, tout en prenant soin de conserver les propriétés essentielles des filtres telles que : stabilité et causalité.

Ainsi, seront exposés brièvement quelques unes de ces méthodes de transposition dans ce qui suit. [3], [4], [7].

1.5.1. Invariance impulsionnelle :

Cette méthode consiste en l'échantillonnage de la réponse impulsionnelle du filtre analogique. Il s'agit de trouver une réponse impulsionnelle échantillonnée $h(k)$ d'un filtre numérique qui soit la réponse échantillonnée de la réponse impulsionnelle approximée $h_a(t)$ du filtre analogique désiré.

Considérons un filtre analogique dont la fonction de transfert est :

$$H_a(S) = \sum_{i=1}^N \frac{\alpha_i}{(s - s_i)} \quad (I.26)$$

Où S_i sont des pôles complexes de $H_a(S)$.

Sa réponse impulsionnelle analogique est donnée par :

$$h_a(t) = \sum_{i=1}^N \alpha_i e^{s_i t} \varepsilon(t) \quad (I.27)$$

Avec :

$$\varepsilon(t) = \begin{cases} 1 & \text{pour } t \geq 0 \\ 0 & \text{ailleurs} \end{cases}$$

L'échantillonnage de $h_a(t)$ donne :

$$h(n) = T_e \cdot h_a(nT_e) = \sum_{i=1}^N \alpha_i e^{s_i \cdot n \cdot T_e} \varepsilon(nT_e) \quad (I.28)$$

Où T_e : période d'échantillonnage.

Aux instants d'échantillonnage, on a bien une parfaite correspondance entre les réponses impulsionnelles des deux filtres numérique et analogique.

La fonction de transfert $H(Z)$ du filtre numérique, étant la transformée en Z de sa réponse impulsionnelle, on a :

$$\begin{aligned} H(Z) &= \sum_{n=0}^{\infty} h(nT_e) Z^{-n} \\ &= \sum_{n=0}^{\infty} \sum_{i=1}^N \alpha_i e^{(s_i \cdot n \cdot T_e)} Z^{-n} \end{aligned}$$

$$H(Z) = \sum_{n=0}^{\infty} \frac{\alpha_i}{1 - e^{s_i T_e} Z^{-1}} \quad \text{pour } |Z| > e^{s_i T_e} \quad (I.29)$$

Avec

$$Z[(e^{-s_i T_e})] = \frac{1}{1 - e^{s_i T_e} Z^{-1}} \quad (I.30)$$

On a donc, la correspondance des pôles suivante :

$$S_i \leftrightarrow Z_i = e^{S_i T_e}$$

Elle permet d'associer à l'axe imaginaire le cercle de rayon unité; et aux deux domaines mitoyens (gauche et droit), les deux domaines intérieurs et extérieurs délimités par le cercle de rayon unité centré à l'origine.

Il faut aussi noter que si $H_a(S)$ est stable alors $H(Z)$ est aussi stable.

I.5.2. Transformation bilinéaire :

Cette méthode consiste à construire pour chaque filtre analogique $H_a(S)$, un filtre numérique $H_z(Z)$ tel que :

$$H(Z = e^{j\omega T_e}) = H_a(S = j\Omega(\omega)) \tag{I.31}$$

Où $\Omega(\omega)$ est appelée la distorsion de fréquence.

Si $\omega T_e \in]-\pi, \pi[$ (théorème d'échantillonnage) alors $\Omega(\omega) \in]-\infty, +\infty[$

Cette transformation est définie par : [3]

$$Z = \frac{1+S}{1-S} \leftrightarrow S = \frac{Z-1}{Z+1} \tag{I.32}$$

D'où
$$j\Omega(\omega) = \frac{e^{j\omega T_e} - 1}{e^{j\omega T_e} + 1} \tag{I.33}$$

Avec
$$\omega_a = \Omega(\omega) = \text{tg}\left(\frac{\omega T_e}{2}\right) \tag{I.34}$$

et
$$\omega = \frac{2}{T_e} \text{Arctg} \omega_a \tag{I.35}$$

où ω_a est la fréquence analogique correspondant à la fréquence numérique ω .

La transformation bilinéaire transforme la partie gauche du plan S en l'intérieur du cercle unité et la partie droite en l'extérieur de ce cercle. donc cette transformation conserve la stabilité.

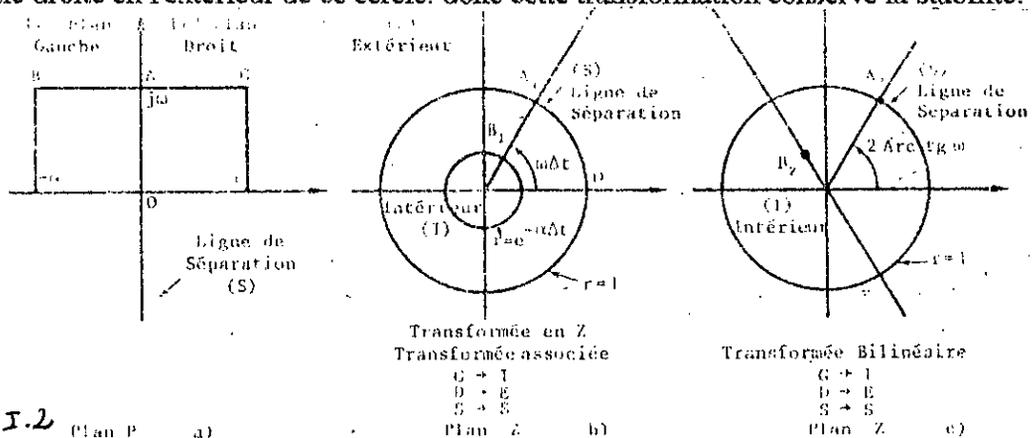


Fig 1.2 Plan P a)

Cette approche présente par rapport à l'invariance impulsionnelle l'avantage d'éviter les problèmes de recouvrement spectral.

En effet, à cause de l'échantillonnage, la réponse fréquentielle du filtre numérique dans le cas de l'invariance impulsionnelle est obtenue par une répétition périodique de celle du filtre analogique et comme ces derniers utilisés en pratique ne sont pas limités en fréquences, un recouvrement spectral est inévitable.

I.6 COMPARAISON ENTRE UN FILTRE ANALOGIQUE ET UN FILTRE NUMERIQUE

En effet, il n'existe pas d'analogie stricte entre le filtrage numérique et analogique. Les deux types sont basés sur des approches fort différentes (Tableau I.1).

Néanmoins, et comme on vient de le voir, les notions de transformée de Fourier et de fonction de transfert, en particulier les transformées en Z et de Laplace, restent applicables et permettant de transposer la plupart des résultats connus sur les filtres analogiques aux filtres numériques.

Tableau 1.1 : Comparaison entre un filtre analogique et un filtre numérique.

Caractéristiques principales	Filtre analogique	Filtre numérique
Processus Fondamentaux	Propriétés électriques, des résistances, capacités et inductances.	Manipulation des nombres
Temps	Continu	Discret
Représentation du Signal	Fonction continue du temps.	Distribution ou Série du temps
Outil Mathématique d'analyse	Transformation de Laplace	Transformation en Z
Fonction de transfert	$H(s)$ variable complexe $s=j\omega$	$H(Z)$ variable complexe $Z = e^{j\omega T}$
Implémentation	Par des composants passifs : R,L,C et actifs : transistors et amplis opérationnels.	Circuits logiques et numériques ou à base d'un microprocesseur.

CHAPITRE II

FILTRES NUMERIQUES

II.1. INTRODUCTION

Dans le chapitre, une idée générale sur les filtres numériques sera présentée. L'accent sera mis sur les filtres numériques à réponse impulsionnelle finie, dont fait l'objet cette présente étude.

Nous présenterons essentiellement deux types de classification des filtres numériques :

- La classification selon la réponse impulsionnelle.
- La classification selon la réponse fréquentielle.

Une étude comparative entre les filtres R.I.F et R.I.I clôturera ce chapitre, mettant en évidence le domaine d'application des filtres à réponse impulsionnelle finie (R.I.F).

II.2. CLASSIFICATION DES FILTRES NUMERIQUES

II.2.1 Classification selon la réponse impulsionnelle :

Dans cette classe on distingue deux types de filtres numériques :

- Les filtres numériques à réponse impulsionnelle finie (R.I.F)

La réponse impulsionnelle de ces filtres présente un nombre fini de coefficients non nuls. Tous les autres le sont .

Les filtres numériques R.I.F sont des systèmes linéaires, discrets, invariants dans le temps, décrits par une équation selon laquelle, un échantillon du signal filtré est donné par une combinaison linéaire du signal d'entrée pondérée par les éléments de la réponse impulsionnelle.

$$S(k) = h(k) * e(k) \quad (II.1)$$

$$S(k) = \sum_{n=0}^{N-1} h(n).e(k-n) \quad (II.2)$$

où $h(k)$ est la réponse impulsionnelle du filtre.

La fonction de transfert est donnée par :

$$H(Z) = \sum_{n=0}^{N-1} h(n).Z^{-n} = \frac{S(Z)}{E(Z)} \quad (II.3)$$

La réponse impulsionnelle $h(k)$ étant un signal discret, la réponse fréquentielle $H(f)$ est une fonction périodique de période $2\pi.f_e$ où f_e est la fréquence d'échantillonnage du domaine temporel.

La réponse fréquentielle $H(f)$ est donc la fonction de transfert du filtre évaluée sur le cercle unité:

$$H(f) = H(Z) \Big|_{Z=e^{j.2\pi.f/f_c}} \quad (\text{II.4})$$

Ces filtres bénéficieront, dans les chapitres suivants, d'une étude plus détaillée.

- Les filtres numériques à réponse impulsionnelle infinie (R.I.I).

La réponse impulsionnelle de ces filtres présente un nombre infini de coefficients non nuls.

Un filtre R.I.I est mis en oeuvre selon l'équation aux différences :

$$s(k) = \sum_{n=0}^M b_n e(k-n) - \sum_{n=1}^N s(k-n) a_n \quad (\text{II.5})$$

Sa fonction de transfert est donnée par :

$$H(f) = \sum_{i=(M-N)}^{+\infty} h(i) e^{-j.2\pi.f.i} \quad (\text{II.6})$$

C'est un système linéaire, discret, invariant dans le temps. Il est à mémoire infinie, du fait de présence d'une boucle de réaction de la sortie sur l'entrée, d'où la dénomination courante de filtre récursif.

Les filtres numériques R.I.I, présentent des propriétés proches de celle des filtres analogiques, ainsi que des techniques de synthèse pratiquement analogues.

Les méthodes de synthèse de ce type de filtres sont classées en deux catégories :

- Les méthodes classiques établissant une correspondance entre les domaines analogique et numérique, pour ne citer que : la méthode de l'invariance de la réponse impulsionnelle et la méthode de transformation bilinéaire.

- Les méthodes algorithmiques d'optimisation assistée par ordinateur.

Tout cela pourra être vu plus en détails en consultant les références [5],[7],[8].

II.2.2. Classification selon la réponse fréquentielle :

On peut classer les filtres selon leurs réponses fréquentielles, on distingue alors :

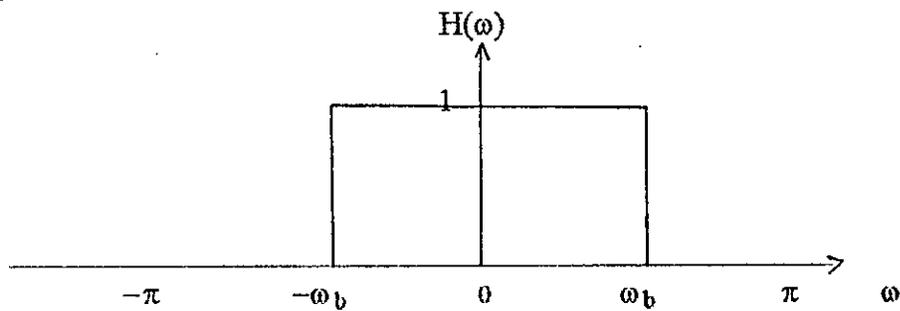
- Les filtres " passe-bas ", qui laissent passer les fréquences inférieures à leur fréquence de coupure f_c
- Les filtres " passe-haut ", qui contrairement favorisent les fréquences supérieures à f_c
- Les filtres " passe-bande ", qui ne laissent passer que les signaux dont les fréquences sont comprises entre deux fréquences de coupures f_1 et f_2 .
- Les filtres " coupe-bande " qui au contraire éliminent les signaux dans une bande $[f_1, f_2]$

On peut ainsi rencontrer d'autres types de filtres comme :

- Le filtre de mise en forme.
- Le différentiateur.
- Le transformateur de Hilbert.
- Le filtre passe-tout.

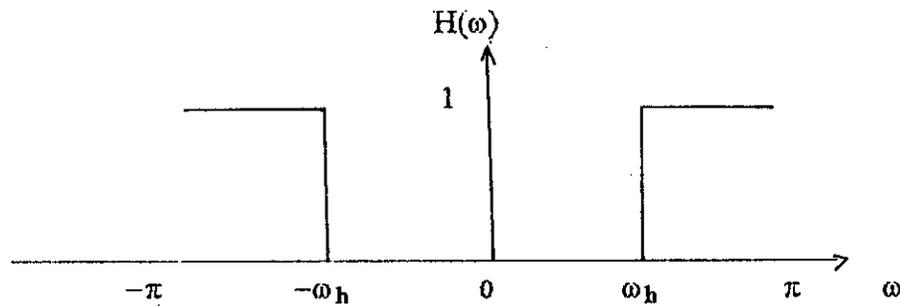
* Filtres idéaux :

a - Filtre passe-bas :



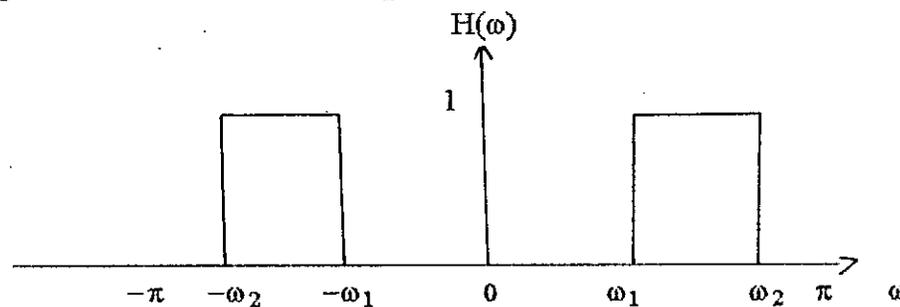
$$H(\omega) = \begin{cases} 1 & \text{pour } |\omega| < \omega_b \\ 0 & \text{pour } \omega_b < |\omega| < \pi \end{cases} \quad (\text{II.7})$$

b - Filtre passe-haut :



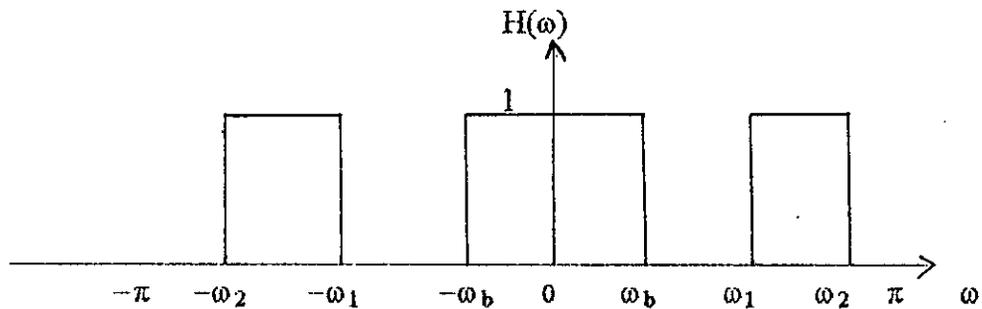
$$H(\omega) = \begin{cases} 1 & \text{pour } \omega_h < |\omega| < \pi \\ 0 & \text{pour } |\omega| < \omega_h \end{cases} \quad (\text{II.8})$$

c - Filtre passe-bande sans bande à l'origine :



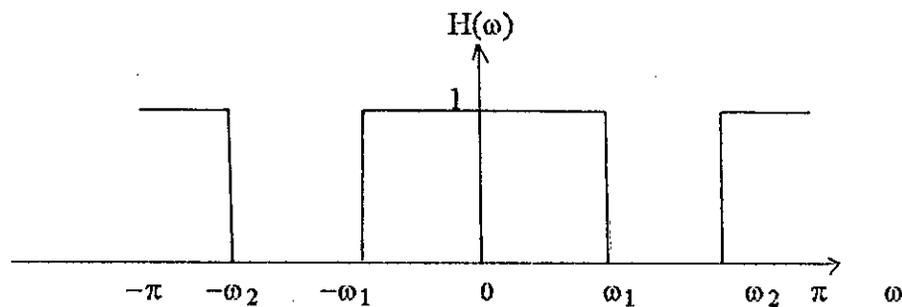
$$H(\omega) = \begin{cases} 1 & \text{pour } \omega_1 < |\omega| < \omega_2 \\ 0 & \text{pour } |\omega| < \omega_1, \omega_2 < |\omega| < \pi \end{cases} \quad (\text{II.9})$$

d - Filtre passe-bande avec bande à l'origine :



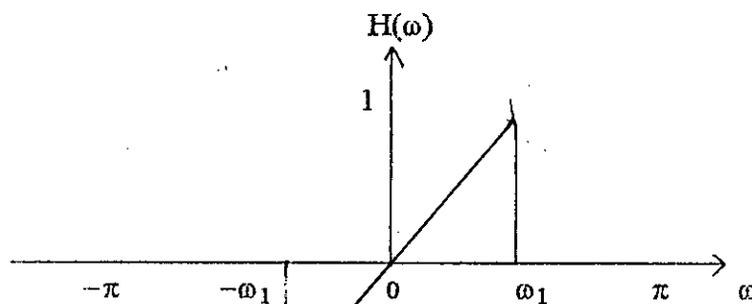
$$H(\omega) = \begin{cases} 1 & \text{pour } |\omega| < \omega_b \text{ et } \omega_1 < |\omega| < \omega_2 \\ 0 & \text{pour } \omega_b < |\omega| < \omega_1, \omega_2 < |\omega| < \pi \end{cases} \quad (\text{II.10})$$

e - Filtre coupe-bande :



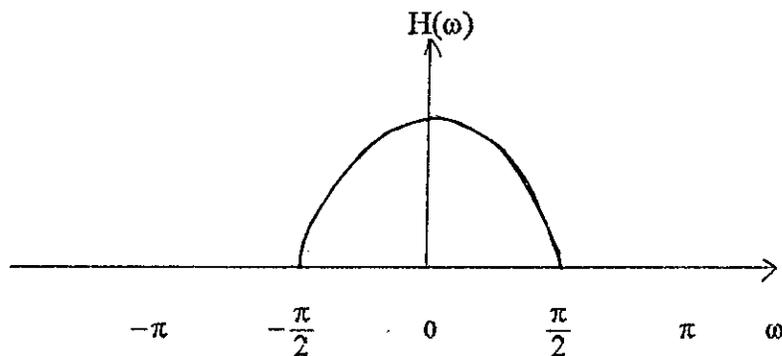
$$H(\omega) = \begin{cases} 1 & \text{pour } |\omega| < \omega_1, \omega_2 < |\omega| < \pi \\ 0 & \text{pour } \omega_1 < |\omega| < \omega_2 \end{cases} \quad (\text{II.11})$$

f - Différentiateur :



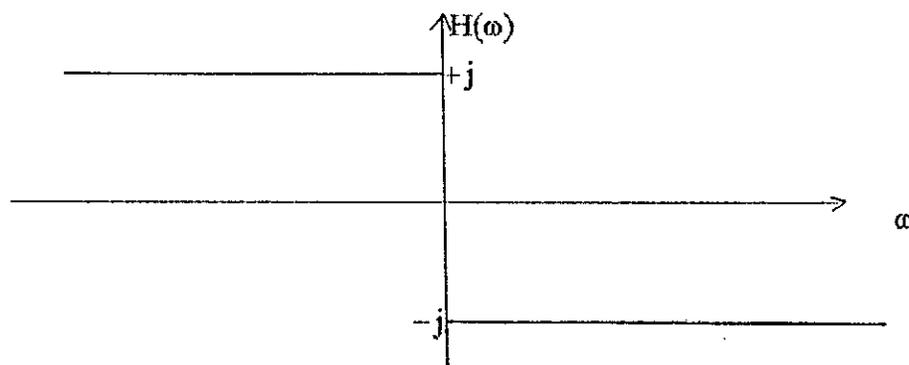
$$H(\omega) = \begin{cases} j \cdot \omega & \text{pour } |\omega| < \omega_1 \\ 0 & \text{pour } \omega_1 < |\omega| < \pi \end{cases} \quad (\text{II.12})$$

g - Filtre de mise en forme :



$$H(\omega) = \begin{cases} \cos \omega \cdot \text{rect}\left(\frac{\omega}{\pi}\right) & \text{pour } |\omega| < \frac{\pi}{2} \\ 0 & \text{pour } \frac{\pi}{2} < |\omega| < \pi \end{cases} \quad (\text{II.13})$$

h - Transformateur de Hilbert :



$$H(\omega) = -j \cdot \text{Sgn}(\omega) \quad \text{avec} \quad \text{Sgn}(\omega) = \begin{cases} 1 & \omega > 0 \\ 0 & \omega < 0 \end{cases} \quad (\text{II.14})$$

II.3. COMPARAISON ENTRE FILTRE R.I.F ET R.I.I

Les filtres R.I.I donnent une réponse fréquentielle satisfaisante pour un nombre de coefficients relativement réduit, ce qui n'est pas le cas pour les filtres R.I.F qui eux, au contraire, nécessitent un nombre assez élevé de coefficients pour arriver à une telle réponse.

Les filtres R.I.I peuvent être calculés par des méthodes classiques, semblables à celles utilisées pour les filtres analogiques.

Néanmoins, les filtres R.I.F présentent des avantages tout aussi méritants ou même plus vu leur formulation mathématique simple et la facilité avec laquelle ils peuvent être implémentés.

Ils sont toujours stables et peuvent être synthétisés facilement avec une phase exactement linéaire.

Les filtres R.I.F sont beaucoup moins sensibles aux erreurs de calcul, qui ne sont pas cumulatives comme dans le cas des filtres R.I.I, vu leur structure non récursive.

CHAPITRE III

SYNTHESE DES FILTRES R.I.F A PHASE LINEAIRE

III.1. INTRODUCTION

La réalisation des filtres numériques se fait, en général selon les étapes suivantes :

- 1 - Spécification des propriétés du filtre désiré. Ces propriétés sont souvent données sous forme fréquentielle: fréquence de coupure, largeur de bande transition, niveau de l'atténuation, etc...
- 2- Approximation des spécifications, en utilisant un système causal invariant dans le temps, et une précision infinie.
- 3 - Approximation des coefficients du filtre en précision finie.
- 4 - Choix d'une structure de réalisation.
- 5 - Simulation des performances du filtre.
- 6 - Implémentation software ou hardware.

III.2. FILTRES RIF A PHASE LINEAIRE

Un filtre est dit à phase linéaire si le spectre de phase de sa réponse fréquentielle est linéaire par rapport à la fréquence.

Soit $\theta(\omega)$ le spectre de phase du filtre, il s'écrit :

$$\theta(\omega) = -\alpha\omega + \beta \quad (\text{III.1})$$

où α et β sont des constantes à déterminer.

La réponse fréquentielle du filtre est de la forme:

$$H(\omega) = \pm |H(\omega)| e^{j\theta(\omega)} \quad (\text{III.2})$$

De même:

$$H(\omega) = \sum_{n=0}^{N-1} h(n) \cdot e^{-j\omega n} \quad (\text{III.3})$$

Les coefficients $h(n)$, $n = 0 \dots N-1$ représentent la réponse impulsionnelle du filtre.

La condition de linéarité du spectre est vérifiée si et seulement si la réponse impulsionnelle du filtre présente une symétrie positive ou négative par rapport au point milieu [1], soit :

- Symétrie positive :

$$\begin{aligned}
 h(n) &= h(N-1-n) & n &= 0, N-1 \\
 \alpha &= (N-1)/2 \\
 \beta &= 0
 \end{aligned}
 \tag{III.4a}$$

- Symétrie négative ou antisymétrie :

$$\begin{aligned}
 h(n) &= -h(N-1-n) & n &= 0, N-1 \\
 \alpha &= (N-1)/2 \\
 \beta &= \pm \pi/2
 \end{aligned}
 \tag{III.4b}$$

Selon que N est pair ou impair, nous distinguons quatre types de filtres R.I.F. à phase linéaire. Les réponses fréquentielles de ces différents filtres sont résumées dans le tableau suivant (tableau III.1) : [4],[8]

Tableau (III.1) : Expressions des réponses fréquentielles des quatre types de filtres RIF

h(n)	N	H(ω)
symétrique	impair	$e^{-j\omega(\frac{N-1}{2})} \sum_{n=0}^{\frac{N-1}{2}} a_n \cos \omega n$
	pair	$e^{-j\omega(\frac{N-1}{2})} \sum_{n=1}^{\frac{N}{2}} b_n \cos \omega(n - \frac{1}{2})$
antisymétrique	impaire	$e^{-j\omega(\frac{N-1}{2})} e^{j\frac{\pi}{2}} \sum_{n=1}^{\frac{N-1}{2}} a_n \sin \omega n$
	pair	$e^{-j\omega(\frac{N-1}{2})} e^{j\frac{\pi}{2}} \sum_{n=1}^{\frac{N}{2}} b_n \sin \omega(n - \frac{1}{2})$
$a_0 = h(\frac{N-1}{2})$	$a_n = 2h(\frac{N-1}{2} - n)$	$b_n = 2h(\frac{N}{2} - n)$

III.3. METHODES DE SYNTHÈSE DES FILTRES RIF A PHASE LINÉAIRE

Le but de cette synthèse est de trouver les coefficients du filtre satisfaisant à des contraintes imposées sous forme fréquentielle (gabarit), il s'agit donc de trouver une méthode de calcul qui conduit à la meilleure approximation du filtre désiré.

III.3.1. Méthode d'échantillonnage en fréquence :

La réponse fréquentielle du filtre désiré est échantillonnée en N points sur le cercle unité. [8],[9],[10].

Les coefficients de la réponse impulsionnelle sont déterminés par la transformée de Fourier discrète inverse :

$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) e^{j\frac{2\pi}{N}nk} \quad (\text{III.5})$$

où $H(k)$ est la réponse fréquentielle échantillonnée.

La transformée en Z des échantillons $h(n)$ est donnée par :

$$\begin{aligned} H(Z) &= \sum_{n=0}^{N-1} h(n) Z^{-n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} H(k) \frac{1 - Z^{-N}}{1 - Z^{-1} e^{j\frac{2\pi}{N}k}} \end{aligned} \quad (\text{III.6})$$

Pour trouver l'approximation du filtre désiré on remplace Z dans l'équation (III.6) par $e^{j\omega}$, soit :

$$\hat{H}(\omega) = H(Z)|_{Z=e^{j\omega}} = \frac{1}{N} \sum_{k=0}^{N-1} H(k) S(\omega, k) \quad (\text{III.7})$$

$$\text{où} \quad S(\omega, k) = \pm e^{j\frac{\pi}{N}k} \frac{\sin[N(\omega/2 - \pi k/N)]}{\sin[\omega/2 - \pi k/N]}$$

est appelée fonction d'interpolation [8],[9].

L'erreur d'approximation est nulle aux points donnés par l'échantillonnage, et reste finie ailleurs.

Le choix du nombre N se fait selon les contraintes imposées sur le niveau des ondulations et sur la largeur de la bande de transition [8],[9].

III.3.2. Méthode du minimax :

Cette méthode consiste à trouver une approximation qui minimise la norme de Chebyshev [8] :

$$\|H(\omega) - \hat{H}(\omega)\| = \max_{\omega_{\min} \leq \omega \leq \omega_{\max}} |W(\omega) (H(\omega) - \hat{H}(\omega))| \quad (\text{III.8})$$

$W(\omega)$ est une fonction de pondération qui permet de contrôler l'erreur sur tout l'intervalle de définition de $H(\omega)$ et de son approximation.

Cette méthode est basée sur la décomposition de $H(\omega)$ en polynôme de Chebyshev de variable $\cos \omega$ et de degré inférieur ou égal au nombre d'échantillons de la réponse impulsionnelle du filtre [11].

Cette méthode garantit des ondulations d'amplitudes égales dans la bande passante et dans la bande coupée d'après le théorème d'alternance [11].

Une étude détaillée sur la synthèse des filtres numériques RIF par la méthode de minimax est donnée dans la référence [12] dans le cadre d'une thèse de magister.

III.3.3. Méthode de fenêtrage temporel :

La réponse fréquentielle d'un filtre numérique RIF étant périodique de période normalisée égale à 2π , elle peut être décomposée en série de Fourier [8],[9],[13] :

$$H(\omega) = \sum_{n=-\infty}^{+\infty} h(n) e^{-j\omega n} \quad (\text{III.9})$$

La réponse impulsionnelle $h(n)$ est donnée par la transformée de Fourier inverse :

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(\omega) \cdot e^{j\omega n} \quad (\text{III.10})$$

La réponse impulsionnelle d'un tel filtre est non causale et de durée infinie, c'est donc un filtre à réponse impulsionnelle infinie (R.L.I.). Pour limiter le nombre de coefficients, des fenêtres temporelles $w(n)$ ont été élaborées; Celles-ci ont la propriété d'être nulles en dehors d'un intervalle $[n_1, n_2]$.

Ainsi, la réponse impulsionnelle approximée

$$\hat{h}(n) = h(n) \cdot W(n) \quad (\text{III.11a})$$

est de durée finie.

La causalité du filtre est obtenue par la translation à droite des coefficients. Cette opération n'a d'effet que sur le spectre de phase qui reste toujours linéaire. Le spectre d'amplitude est inchangé.

Si n_1 est positif, cette dernière opération n'a pas lieu d'être. Pour n_1 négatif, on a :

$$h(n) = \begin{cases} h(n + n_1) & 0 \leq n \leq n_2 - n_1 \\ 0 & \text{ailleurs} \end{cases} \quad (\text{III.11b})$$

A un produit simple dans le domaine temporel, correspond un produit de convolution dans le domaine fréquentiel. La réponse fréquentielle du filtre calculé est :

$$H(\omega) = \frac{1}{2\pi} H(\omega) * W(\omega)$$

$$H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(\Omega) W(\omega - \Omega) d\Omega \quad (\text{III.12})$$

Le but de cette méthode est de trouver la meilleure approximation du filtre désiré. Dans le cas idéal $\hat{H}(\omega)$ coïnciderait avec $H(\omega)$. Dans ce cas la transformée de Fourier de la fenêtre serait une impulsion de Dirac. Ceci ne pouvant être réalisé pratiquement, les fenêtres sont choisies de façon à tendre vers l'impulsion de Dirac.

III.4. SYNTHÈSE DES FILTRES NUMÉRIQUES RIF PAR LA MÉTHODE DE FENÊTRAGE

III.4.1. Présentation de la méthode :

Le passage d'une réponse impulsionnelle de durée infinie à une réponse impulsionnelle de durée finie, peut se faire, de manière brute, par une troncature directe des coefficients. Cette troncature est le siège, pour la réponse fréquentielle, de perturbations connues sous le nom du phénomène de GIBBS [8]; il se traduit par une évolution plus ou moins lente de la réponse fréquentielle autour des discontinuités: on parle de bandes de transitions, et par l'apparition d'oscillations autour de ces mêmes points.

Le phénomène de GIBBS étant très gênant, il serait souhaitable de pouvoir réduire les bandes de transition, réduire les amplitudes des ondulations dans les bandes passantes et augmenter l'atténuation en bandes coupées.

Cependant, ces objectifs ne pouvant être atteints simultanément, les fonctions fenêtres temporelles sont établies dans le sens de réaliser le meilleur compromis entre eux.

III.4.2. Propriétés des fenêtres :

Les fonctions fenêtres temporelles sont toutes positives, paires et nulles en dehors de l'intervalle considéré. La parité des fenêtres est une propriété très importante, elle permet de conserver la linéarité de la phase du filtre.

Dans le domaine fréquentiel, les fenêtres se présentent sous forme d'un lobe principal et de lobes secondaires, comme le montre la figure III.1.

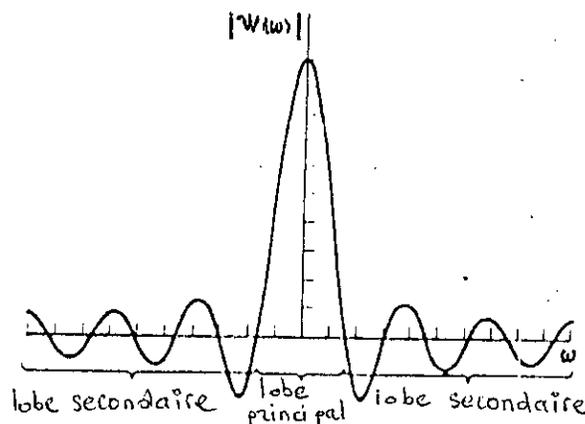


Figure III.1 Définition des lobes principal et secondaires.

Le lobe principal est défini comme étant la partie de la transformée de Fourier de la fenêtre comprise entre les deux premiers points d'intersection de la courbe avec l'axe des fréquences. Le reste de la courbe définit les lobes secondaires [13].

La convolution (III.12) de la réponse fréquentielle du filtre par la transformée de Fourier de la fenêtre est à l'origine des perturbations observées dans le spectre calculé [1]. Pour mieux voir la cause de ces perturbations, décomposons le spectre d'amplitude de la fenêtre $W(\omega)$ en deux signaux, l'un représente le lobe principal $P(\omega)$, l'autre les lobes secondaires $S(\omega)$. Par conséquent on a :

$$W(\omega) = P(\omega) + S(\omega) \quad (\text{III.13})$$

La relation (III.12) est alors exprimée par :

$$H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(\Omega) P(\omega - \Omega) d\Omega + \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(\Omega) S(\omega - \Omega) d\Omega \quad (\text{III.14})$$

Les effets des deux signaux seront étudiés séparément pour plus de simplicité.

La convolution de $H(\omega)$ par $P(\omega)$ est illustrée par la figure III.2.

- Lorsque le lobe principal est totalement contenu dans la bande passante du filtre, la convolution donne une constante égale à son aire. Dans ce cas le point de symétrie est inférieur à ω_1 .

- Lorsque le lobe principal est totalement contenu dans la bande coupée du filtre, le résultat de la convolution est nul. Dans ce cas, le point de symétrie est supérieur à ω_5 .

- Lorsque le centre de symétrie du lobe principal est translaté entre les deux positions extrêmes ω_1 et ω_5 , l'aire calculée par la convolution diminue graduellement jusqu'à s'annuler pour ω_5 , provoquant ainsi une bande de transition égale à la largeur du lobe principal.

La convolution de $H(\omega)$ par $S(\omega)$ est illustrée par la figure III.3.

En bandes passantes, le résultat de la convolution est pratiquement constant. Au niveau de la discontinuité du filtre désiré, la convolution conduit à des oscillations dont les amplitudes dépendent des amplitudes des lobes secondaires [13].

En résumé, les bandes de transition sont liées à la largeur du lobe principal, les oscillations, quant à elles, sont proportionnelles au niveau du premier lobe secondaire. Pour minimiser ce phénomène, et donc pour se rapprocher au mieux de l'impulsion de Dirac, comme il a été vu précédemment, l'énergie des fenêtres doit être concentrée aux basses fréquences, c'est à dire dans le lobe principal.

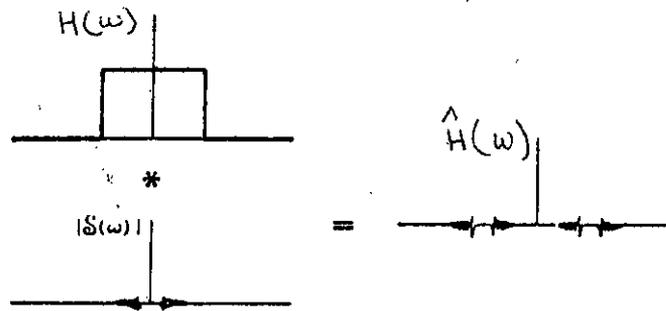


Figure III.3 Effet des lobes secondaires sur le spectre d'amplitude du filtre

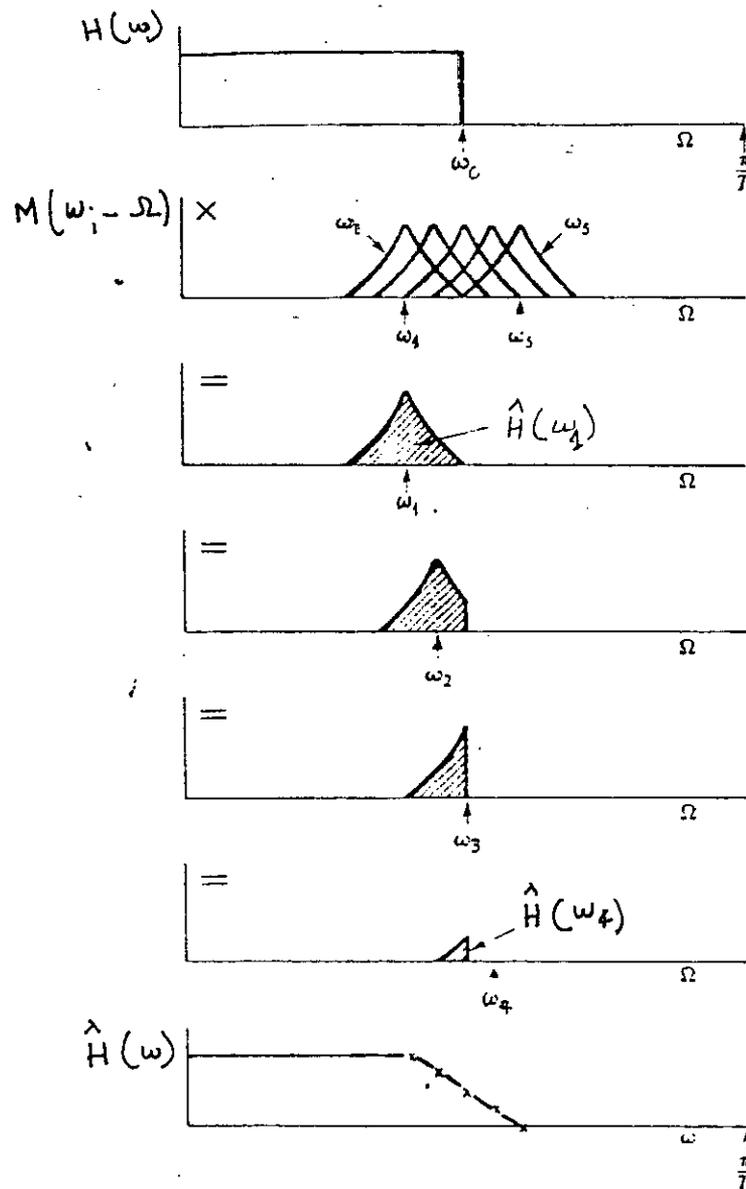


Figure III.2 Effet du lobe principal sur le spectre d'amplitude du filtre

III.4.3. Exemples de quelques fenêtres :

Dans les expressions des fenêtres données ci-dessous, le nombre N est supposé impair.

- Fenêtre rectangulaire :

Elle correspond à une simple troncature des coefficients du filtre [14] :

$$D(n) = \begin{cases} 1 & |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad \text{(III.15a)}$$

Sa transformée de Fourier est (figure III.4) :

$$W_d(\theta) = \frac{\sin N\theta/2}{\sin \theta/2} \quad \text{(III.15b)}$$

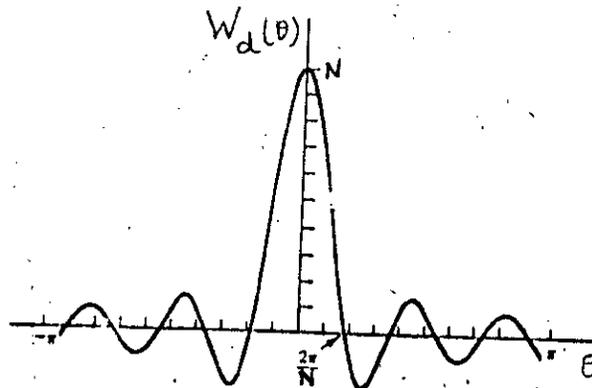


Figure III.4 Domaine fréquentiel de la fenêtre rectangulaire.

La largeur du lobe principal est inversement proportionnelle au nombre de coefficients N .

$$\Delta\omega = \frac{4\pi}{N} \quad \text{(III.15c)}$$

- Fenêtre triangulaire :

Son expression est la suivante : [14]

$$W(n) = \begin{cases} 1 - 2|n|/(N-1) & |n| \leq (N-1)/2 \\ 0 & \text{ailleurs} \end{cases} \quad \text{(III.16a)}$$

Sa transformée de Fourier est :

$$W(\theta) = \frac{2}{N} \left[\frac{\sin(\frac{N-1}{4}\theta)}{\sin(\theta/2)} \right]^2 \quad \text{(III.16b)}$$

Les oscillations en bande coupée sont plus atténuées que celles de la fenêtre rectangulaire, cependant, la largeur du lobe principal est augmentée; Elle reste toujours inversement proportionnelle au nombre de coefficients considéré, avec un facteur de proportionnalité supérieur à celui de fenêtre rectangulaire.

$$\Delta\omega = \frac{8\pi}{N} \quad (\text{III.16c})$$

- Fenêtres cosinusoidales :

C'est une famille de fenêtres qui dépendent d'un paramètre. Elles sont de la forme : [14]

$$\omega(n) = \begin{cases} \cos^\alpha\left(\frac{n}{N-1}\pi\right) & |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{III.17a})$$

Dans le cas particulier, où le paramètre est égal à 2, la fenêtre est appelée fenêtre de HANNING, son expression est la suivante :

$$\omega(n) = \begin{cases} 0.5 + 0.5\cos\left(\frac{2\pi}{N-1}n\right) & |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{III.17b})$$

Notons que la fenêtre de HANNING est liée à la fenêtre rectangulaire par la relation :

$$\omega(n) = 0.5D(n) + 0.5D(n)\cos\left(\frac{2\pi}{N-1}n\right) \quad (\text{III.17c})$$

Sa transformée de Fourier s'écrit en fonction de la transformée de Fourier de la fenêtre rectangulaire selon (III.17.d)

$$W(\theta) = 0.5W_d(\theta) + 0.25\left(W_d\left(\theta - \frac{2\pi}{N-1}\right) + W_d\left(\theta + \frac{2\pi}{N-1}\right)\right) \quad (\text{III.17d})$$

Pour des valeurs de N suffisamment grandes, N-1 est approximée à N, la relation (III.17d) peut s'écrire :

$$W(\theta) = 0.5W_d(\theta) + 0.25\left(W_d\left(\theta - \frac{2\pi}{N}\right) + W_d\left(\theta + \frac{2\pi}{N}\right)\right) \quad (\text{III.17e})$$

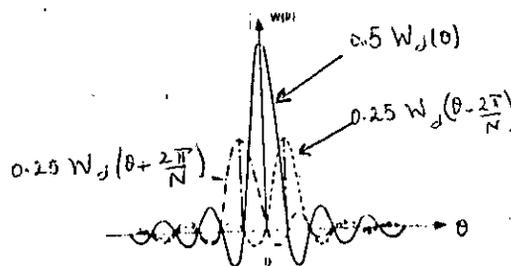


Figure III.5 Composantes de la T.F de la fenêtre de HANNING.

Les positions des lobes principaux de $W_d(\theta - \frac{2\pi}{N})$ et $W_d(\theta + \frac{2\pi}{N})$ font que les premiers zéros de la transformée de Fourier de la fenêtre de HANNING se situent à $\theta = \pm \frac{4\pi}{N}$, de sorte que la largeur de base du lobe principal $\Delta\omega$ soit égal à $\frac{8\pi}{N}$, soit le double de la largeur du lobe principal de la fenêtre rectangulaire.

Le premier lobe secondaire de cette fenêtre est plus atténué que celui de la fenêtre rectangulaire, du fait de la sommation d'éléments en opposition de phase.

- Fenêtre de Hamming :

Se basant sur les résultats obtenus par VON-HANN [15], HAMMING se propose de minimiser le niveau du lobe secondaire, en gardant constante la largeur de base du lobe principal. L'expression de la fenêtre de HAMMING généralisée est [14],[15] :

$$w(n) = \begin{cases} \alpha + (1-\alpha) \cdot \cos(\frac{2\pi}{N-1}n) & |n| < \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{III.18a})$$

Sa transformée de Fourier s'écrit pour des ordres élevés :

$$W(\theta) = \alpha \cdot W_d(\theta) + 0.5 \cdot (1-\alpha) \cdot (W_d(\theta - \frac{2\pi}{N}) + W_d(\theta + \frac{2\pi}{N})) \quad (\text{III.18b})$$

HAMMING démontre empiriquement que son objectif est atteint lorsque le spectre $W(\theta)$ s'annule au point $\theta = 2.5 \frac{2\pi}{N}$ [15]. La valeur de α correspondante est déterminée en posant :

$$W(\theta) \Big|_{\theta = \frac{5\pi}{N}} = 0 \quad (\text{III.18c})$$

On a :

$$W_d(\theta) \Big|_{\theta = \frac{5\pi}{N}} = \frac{\sin 2.5\pi}{\sin 2.5 \frac{\pi}{N}} = \frac{1}{\sin 2.5 \frac{\pi}{N}} \quad (\text{III.18d})$$

$$W_d(\theta - \frac{2\pi}{N}) \Big|_{\theta = \frac{5\pi}{N}} = \frac{\sin 1.5\pi}{\sin 1.5 \frac{\pi}{N}} = \frac{-1}{\sin 1.5 \frac{\pi}{N}} \quad (\text{III.18e})$$

$$W_d(\theta + \frac{2\pi}{N}) \Big|_{\theta = \frac{5\pi}{N}} = \frac{\sin 3.5\pi}{\sin 3.5 \frac{\pi}{N}} = \frac{-1}{\sin 3.5 \frac{\pi}{N}} \quad (\text{III.18e})$$

Pour N grand, on fait l'approximation suivante:

$$\sin(\frac{1}{N}) \approx \frac{1}{N} \quad (\text{III.18g})$$

En tenant compte de (III.18.g), et en remplaçant (III.18d), (III.18e) et (III.18f) dans (III.18c), on aboutit à la valeur de α , soit :

$$\alpha = 0.54$$

D'où l'expression de la fenêtre de HAMMING:

$$W(n) = \begin{cases} 0.54 + 0.46 \cos\left(\frac{2\pi}{N-1}n\right) & |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{III.18h})$$

- Fenêtre de BLACKMAN :

Son expression est [14] :

$$W(n) = \begin{cases} \sum_{m=0}^K a_m \cos\left(\frac{2\pi}{N-1}m.n\right) & |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{III.19a})$$

Sa transformée de Fourier est:

$$W(\theta) = \sum_{m=0}^K 0.5 a_m (W_d(\theta - \frac{2\pi}{N}m) + W_d(\theta + \frac{2\pi}{N}m)) \quad (\text{III.19b})$$

avec la contrainte:

$$\sum_{m=0}^K a_m = 1 \quad (\text{III.19c})$$

Les fenêtres de HANNING et de HAMMING ont la même forme que la fenêtre de BLACKMAN, avec deux coefficients non nuls a_0 et a_1 . On peut construire des fenêtres avec K coefficients non nuls et ainsi réaliser $(2K-1)$ sommations de fonctions SINC (transformée de Fourier de la fenêtre rectangulaire).

Pour réaliser des fenêtres dont le lobe principal est étroit, on doit restreindre la valeur de K . Blackman a étudié le cas pour lequel $K=3$, et démontre empiriquement que le niveau du premier lobe secondaire est minimum lorsque la transformée de Fourier de sa fenêtre s'annule aux points $\theta_1 = 3.52\pi/N$ et $\theta_2 = 4.52\pi/N$.

En remplaçant ces deux valeurs dans l'expression (III.19b) et en annulant cette dernière, il obtient

les valeurs que voici :

$$\begin{aligned} a_0 &= 0.42 \\ a_1 &= 0.50 \\ a_2 &= 0.08 \end{aligned} \quad (\text{III.19d})$$

- Fenêtre de KAISER :

Elle est définie par la formule [1],[8],[16]:

$$\omega(n) = \begin{cases} \frac{I_0\left(\alpha \sqrt{1 - \left(\frac{n}{\frac{N-1}{2}}\right)^2}\right)}{I_0(\alpha)} & |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{III.20a})$$

$I_0(\cdot)$ est la fonction de Bessel modifiée d'ordre zéro et de première espèce. Elle peut être évaluée en utilisant la série suivante [1] :

$$I_0(x) = \sum_{k=0}^{+\infty} \frac{(x/2)^{2k}}{k!^2} \quad (\text{III.20b})$$

α est un paramètre qui permet au niveau du premier lobe secondaire de varier de façon continue. La largeur du lobe principal, fonction de ce paramètre et du nombre d'échantillons considéré, est ajustée par un choix adéquat de ce dernier.

Pour un filtre passe bas (ou un filtre passe haut) l'influence du paramètre α sur le niveau du lobe secondaire et sur la largeur de base du lobe principal est exprimée par les relations ci-dessous [4] :

$$\alpha = \begin{cases} 0 & A_a \leq 21 \\ 0.5842(A_a - 21)^{0.4} + 0.07886(A_a - 21) & 21 < A_a \leq 50 \\ 0.1102(A_a - 8.7) & A_a > 50 \end{cases} \quad (\text{III.20c})$$

A_a est l'atténuation en bande coupée exprimée en dB.

Le nombre N de coefficients vérifie l'inégalité :

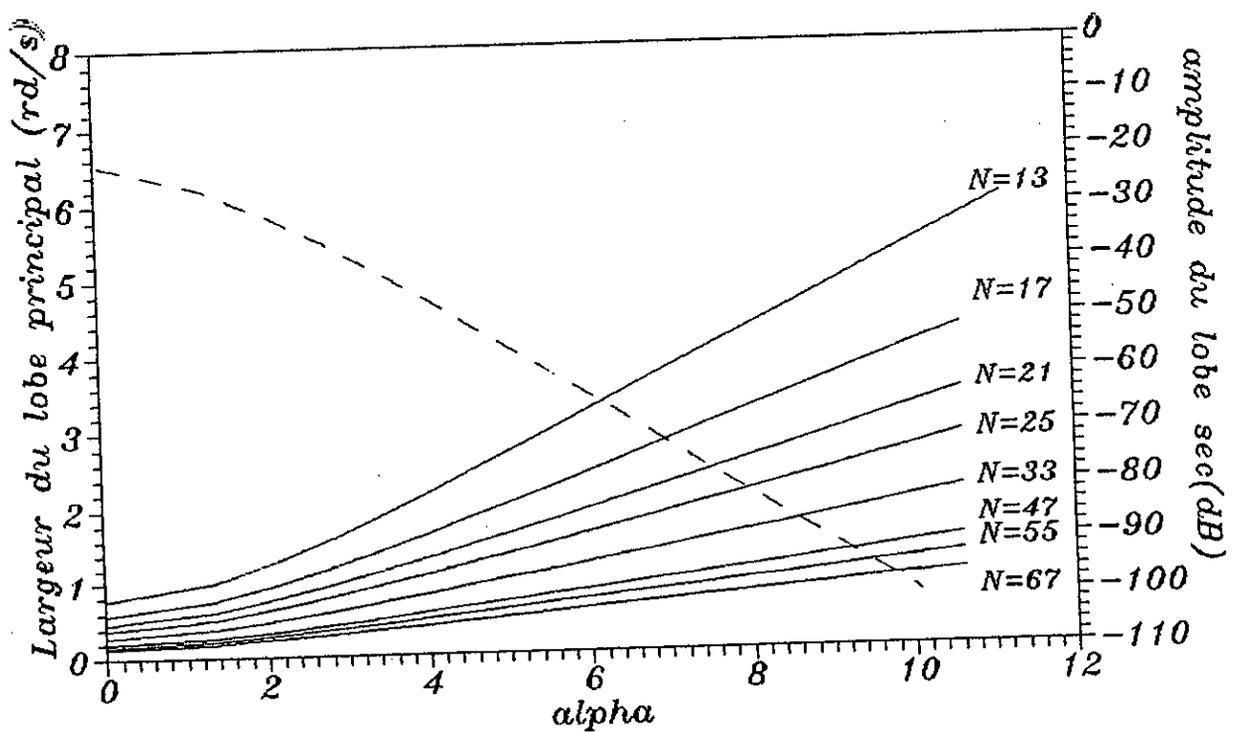
$$N \geq \frac{\omega_s D}{B} + 1 \quad (\text{III.20d})$$

où D vérifie la relation :

$$D = \begin{cases} 0.9222 & A_a \leq 21 \\ \frac{A_a - 7.95}{14.36} & A_a > 21 \end{cases} \quad (\text{III.20e})$$

ω_s est la fréquence d'échantillonnage de $W(n)$ exprimée en rd/s, et B est la largeur de base du pic principal.

La figure (III.6) illustre cet effet.



PARAMETRES DE LA FENETRE DE KAISER
 frequence d'echantillonnage: 10 rd/s
 ——— largeur du lobe principal
 - - - amplitude du lobe secondaire

Le tableau ci-dessous (tableau III.2) résume les propriétés des fenêtres étudiées :

Fenêtre	Largeur du lobe princip.	Niveau du lobe secondaire
Rectangulaire	$4\pi/N$	- 13 dB
Triangulaire	$8\pi/N$	- 25 dB
Hanning	$8\pi/N$	- 31 dB
Hamming	$8\pi/N$	- 41 dB
Blackman	$12\pi/N$	- 57 dB

III.4.4. Elaboration d'un programme et résultats :

- DESCRIPTION :

Comme il a été indiqué dans la section (III.2), la condition de linéarité de la phase du filtre implique une réponse impulsionnelle symétrique ou antisymétrique. Dans notre étude, nous considérons les cas pour lesquels l'ordre du filtre est impair. Ayant la réponse fréquentielle du filtre désiré, on calcule les réponses impulsionnelles selon la relation (III.10).

Le tableau III.3 résume les expressions des différents filtres et les puissances moyennes correspondantes (P).

Le transformateur de Hilbert et le filtre différentiateur sont par définition des filtres antisymétriques.

Tableau III.3 Expression des réponses impulsionnelles des différents filtres.		
	Filtre symétrique (N impair)	Filtre antisymétrique (N pair)
Filtre différentiateur ω_c : fréquence de coupure (rd/s)		$h(n) = \frac{\omega_c}{\pi} \cos n\omega_c - \frac{1}{n\pi^2} \sin n\omega_c$ $h(0) = 0$ $p = \frac{\omega_c}{3\pi}$
Transformateur de Hilbert		$h(n) = \frac{1}{\pi n} - \frac{1}{\pi^2 n} \cos n\pi$ $h(0) = 0$ $p = 1$
Filtre passe-bas ω_c : fréquence de coupure (rd/s)	$h(n) = \frac{1}{\pi n} \sin n\omega_c$ $h(0) = \frac{\omega_c}{\pi}$ $p = \frac{\omega_c}{\pi}$	$h(n) = \frac{1}{\pi n} - \frac{1}{\pi n} \cos n\omega_c$ $h(0) = 0$ $p = \frac{\omega_c}{\pi}$
Filtre passe-haut ω_c : fréquence de coupure (rd/s)	$h(n) = -\frac{1}{\pi n} \sin n\omega_c$ $h(0) = 1 - \frac{\omega_c}{\pi}$ $p = 1 - \frac{\omega_c}{\pi}$	$h(n) = \frac{1}{\pi n} \cos n\omega_c - \frac{1}{\pi n} \cos n\pi$ $h(0) = 0$ $p = 1 - \frac{\omega_c}{\pi}$
Filtre passe-bande k : nombre de bandes $\omega(2i)$: fréquence de coupure haute $\omega(2i-1)$: fréquence de coupure basse	$h(n) = \frac{1}{\pi n} \left\{ \sum_{i=1}^k \sin(n\omega(2i)) \sin(n\omega(2i-1)) \right\}$ $h(0) = \frac{1}{\pi} \sum_{i=1}^k \left\{ \omega(2i) - \omega(2i-1) \right\}$ $p = \frac{1}{\pi} \sum_{i=1}^k \left\{ \omega(2i) - \omega(2i-1) \right\}$	$h(n) = \frac{1}{\pi n} \left\{ \sum_{i=1}^k \cos(n\omega(2i-1)) - \cos(n\omega(2i)) \right\}$ $h(0) = 0$ $p = \frac{1}{\pi} \sum_{i=1}^k \left\{ \omega(2i) - \omega(2i-1) \right\}$
Filtre coupe bande ω_1 : fréquence coupe basse ω_2 : fréquence coupe haute	$h(n) = \frac{1}{\pi n} \left\{ \sin n\omega_1 - \sin n\omega_2 \right\}$ $h(0) = \frac{\omega_1 - \omega_2}{2} + 1$ $p = \frac{\omega_1 - \omega_2}{2} + 1$	$h(n) = \frac{1}{\pi n} \left\{ 2 + \cos n\omega_2 - \cos n\omega_1 \right\}$ $h(0) = 0$ $p = 1 + \frac{\omega_1 - \omega_2}{2}$
Filtre de mise en forme des impulsions	$h(n) = \frac{-1}{\pi(n^2-1)} \cos n\frac{\pi}{2}$ $h(0) = \frac{1}{\pi}$ $h(1) = 1/4$ $p = 1/4$	$h(n) = \frac{1}{\pi(n^2-1)} (n - \sin n\frac{\pi}{2})$ $h(0) = 0$ $h(1) = 1/2\pi$ $p = 1/4$

Le programme que nous avons élaboré contient les étapes suivantes:

Etape 1 :

- Choix du filtre à synthétiser : filtre passe bas, filtre passe haut, passe bande etc..., et de ses caractéristiques (spécifications).

- Choix d'une réponse impulsionnelle symétrique ou antisymétrique. Généralement le cas symétrique est réservé aux filtres sélecteurs de bandes, et le cas antisymétrique convient au transformateur de Hilbert et au filtre différentiateur.

- Choix du critère d'évaluation de l'ordre du filtre :

Critère 1: évaluation de l'ordre par un choix personnel

Critère 2: évaluation par comparaison entre la puissance du filtre calculé et celle du filtre désiré.

Critère 3: évaluation selon des contraintes imposées sur le gabarit du filtre à calculer.

- Etape 2 :

Évaluation de l'ordre N, calcul des coefficients de la fenêtre, de la réponse impulsionnelle du filtre désiré, et utilisation de la relation (III.11a) pour calculer les coefficients de l'approximation du filtre.

Critère 1: l'ordre N et la fenêtre à utiliser sont choisis selon le bon vouloir de l'opérateur.

Critère 2: on initialise N à une valeur impaire fixée par le programme, puis on calcule les coefficients du filtre après choix d'une fenêtre.

On compare l'erreur relative, entre la puissance du filtre calculé et celle du filtre désiré, à une tolérance introduite au préalable. Cette tolérance est donnée par :

$$\text{Tolérance (\%)} = \frac{\text{Puissance du filtre désiré} - \text{Puissance du filtre calculé}}{\text{puissance du filtre désiré}} \quad (\text{III.21})$$

Si l'erreur est supérieure à la tolérance, on incrémente N d'une valeur paire, et on refait les calculs, sinon, l'ordre du filtre est égal à la dernière valeur de N.

Critère 3: on fixe une valeur de l'atténuation en bande coupée. La fenêtre est choisie en conséquence.

On choisit une largeur de la bande de transition à partir de laquelle on évalue la valeur de N.

On peut aussi évaluer N en choisissant un rapport R entre la largeur de la bande de transition et la bande passante (coupée) du filtre.

- Etape 3 :

Calcul de la réponse fréquentielle du filtre (spectre d'amplitude). On utilise à cet effet la transformée de Fourier discrète (TFD). Cette dernière est calculée par l'algorithme FFT (Fast Fourier Transform).

Les détails de cet algorithme sont donnés dans l'annexe 2. L'organigramme de la figure (III.7) décrit ces différentes étapes.

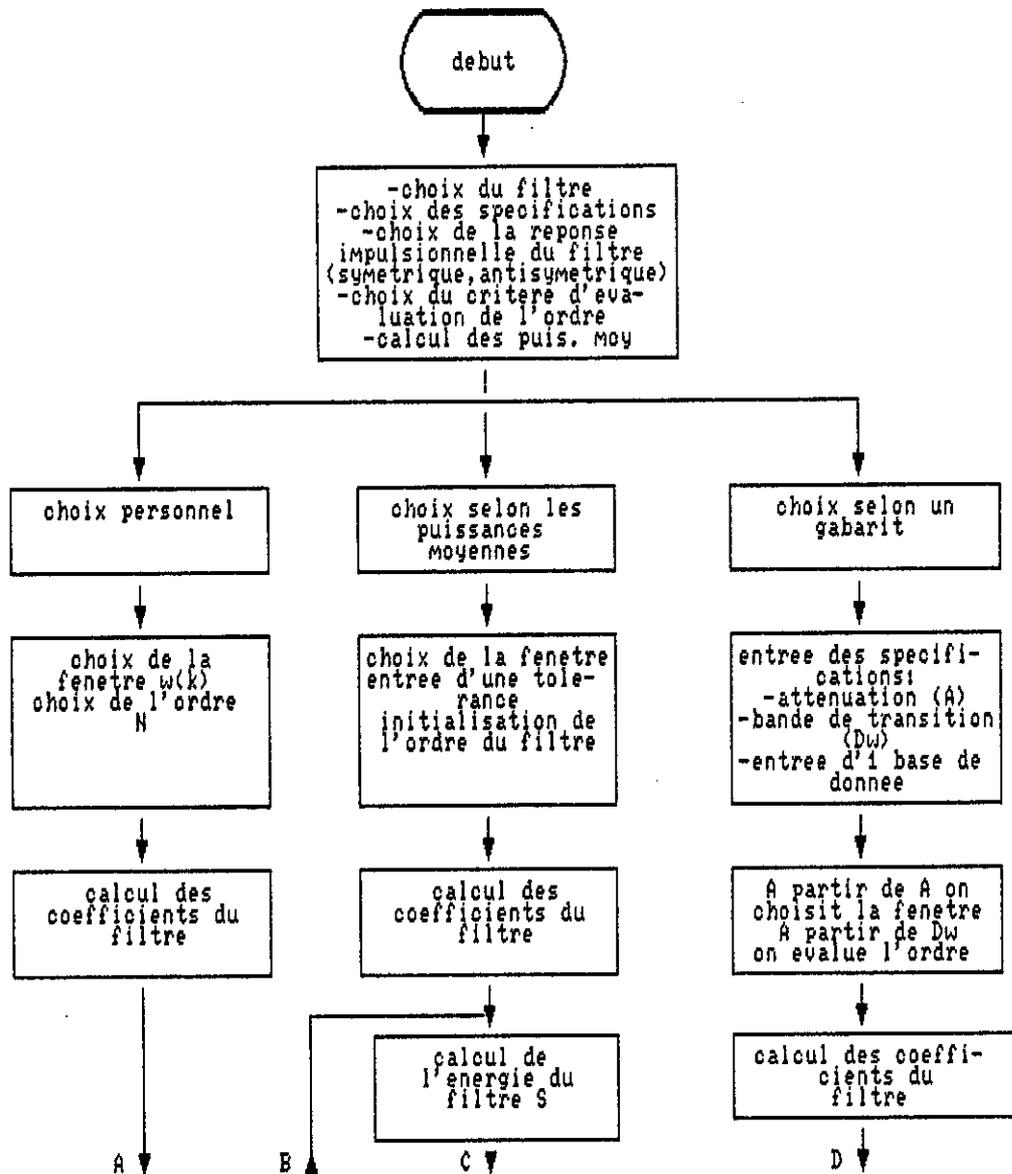
La réponse impulsionnelle du filtre étant symétrique ou antisymétrique, il suffit de calculer les coefficients sur la moitié de l'intervalle, soit $[0, (N-1)/2]$.

La réponse fréquentielle est calculée selon la relation :

$$H(\omega) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \hat{h}(n) \cdot e^{j\omega n} \quad (\text{III.22a})$$

$$H(\omega) = \sum_{n=0}^{\frac{N-1}{2}} \hat{h}(n) \cdot (e^{j\omega n} \pm e^{-j\omega n}) - h(0)$$

$$H(\omega) = \begin{cases} 2 \sum_{n=0}^{\frac{N-1}{2}} \hat{h}(n) \cdot \cos n\omega - \hat{h}(0) & \text{cas symétrique} \\ 2j \sum_{n=0}^{\frac{N-1}{2}} \hat{h}(n) \cdot \sin n\omega - \hat{h}(0) & \text{cas antisymétrique} \end{cases} \quad (\text{III.22b})$$



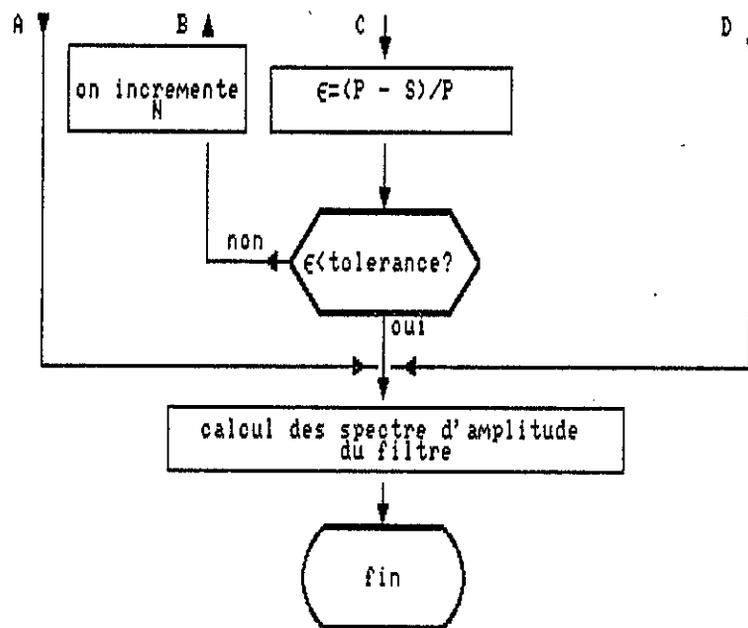


Fig.3.7 Organnigramme de synthese des filtres RIF par la methode de fenetrage

Il est à noter que la réponse impulsionnelle calculée est celle d'un filtre non causal. Pour se ramener à un filtre causal, on translate les coefficients de $(N-1)/2$ points vers la droite. Cette translation n'affectant en rien le spectre d'amplitude.

En effet si : $h(n) \xrightarrow{T.F} H(\omega)$

$$h(n-a) \xrightarrow{T.F} G(\omega) = H(\omega).e^{-j\omega a}$$

Le spectre d'amplitude de $G(\omega)$ est identique à celui de $H(\omega)$.

- RESULTATS ET INTERPRETATIONS :

Dans tous les exemples considérés, le paramètre ω_c indique une fréquence de coupure normalisée par rapport à 2π . Dans ce cas la fréquence maximum à utiliser est égale à 0.5.

- Exemple 1 :

Soit à synthétiser un filtre passe bas d'ordre 51, de fréquence de coupure normalisée $\omega_c = 0.1$. On applique à cet effet quatre différentes fenêtres.

Les résultats sont illustrés par les figures (III.8a) à (III.8e).

Lorsqu'on utilise la fenêtre rectangulaire, le phénomène de GIBBS est très apparent. Quoique la bande de transition est relativement étroite, l'atténuation en bande coupée est faible, elle est de l'ordre de -21 dB.

La fenêtre triangulaire améliore l'atténuation en bande coupée qui est de -27 dB (comparativement à la fenêtre rectangulaire), au prix d'une bande de transition plus large (le double).

La fenêtre de HAMMING donne une atténuation en bande coupée de l'ordre de -50 dB. La largeur du lobe principal est le double de celle du filtre utilisant la fenêtre rectangulaire.

La fenêtre de KAISER utilisée pour $\alpha = 7.87$, donne une bonne atténuation en bande coupée (environ -80 dB). La bande de transition est la plus large.

Cet exemple met en évidence l'effet du fenêtrage sur la réponse fréquentielle du filtre (phénomène de GIBBS). Il montre aussi que l'amélioration de l'atténuation en bande coupée provoque un élargissement de la bande de transition et vice versa.

- Exemple 2 :

Soit à synthétiser un filtre numérique passe bande, dont le rapport R est fixé à 15% .

$$R = \frac{\text{Largeur de la bande de transition}}{\text{Largeur de la bande passante}}$$

Dans cet exemple, on désire avoir une atténuation de -50 dB pour les trois dans suivants :

$$\begin{array}{lcl} \omega_{c1} = 0.1 & , & \omega_{c2} = 0.39 \\ \omega_{c1} = 0.1 & , & \omega_{c2} = 0.25 \\ \omega_{c1} = 0.1 & , & \omega_{c2} = 0.21 \end{array}$$

Les résultats sont illustrés par les figures (III.9a) à (III.9e). Pour avoir l'atténuation citée la fenêtre choisie est celle de HAMMING.

Pour différentes valeurs de la bande passante, nous obtenons différentes valeurs de l'ordre N.

Tableau III.4. : Evolution de l'ordre N en fonction de la bande de transition

ω_{c1}	ω_{c2}	$\Delta\omega$	N
0.1	0.35	0.25	105
0.1	0.25	0.15	177
0.1	0.22	0.12	217
0.1	0.21	0.11	241
0.1	0.20	0.10	265
0.1	0.18	0.08	329
0.1	0.17	0.07	377
0.1	0.15	0.05	529

Le rétrécissement de la largeur relative de la bande de transition suite à un rétrécissement de la bande passante provoque une augmentation de l'ordre du filtre. Ceci est justifié par la relation de proportionnalité inverse qui existe entre ces deux paramètres.

- Exemple 3 :

Soit à synthétiser un filtre passe haut de fréquence de coupure normalisée $\omega_c = 0.3$, pour trois ordres différents. On applique deux fenêtres (Riemann et bohman)

Les résultats sont illustrés par les figures (III.10a) à (III.10d) et de(III.11a) à (III.11d).

Lorsque N augmente la bande de transition rétrécit, les oscillations en bande coupée se resserrent, c'est à dire que leur fréquence augmente, cependant le niveau de l'atténuation reste constant pour une fenêtre donnée.

- Exemple 4 :

Soient à synthétiser un filtre passe bas et un filtre passe haut de fréquences de coupure respectives 0.1 et 0.3. On applique la fenêtre de KAISER pour trois valeurs différentes du paramètre α .

Les résultats sont illustrés par les figures (III.12a) à (III.12d), (III.13a) à (III.13d) et de (III.14a) à (III.14d).

L'augmentation de α provoque un élargissement de la bande de transition et donne une meilleure atténuation en bande coupée. Pour une valeur donnée de α , l'atténuation est indépendante de l'ordre du filtre.

- Exemple 5 :

On utilise la fenêtre de Blackman pour synthétiser un filtre de mise en forme, un filtre différentiateur, un transformateur d'Hilbert et un filtre multi-passe bandes avec une bande à l'origine.

Cette fenêtre possède une bonne atténuation en bandes coupées, elle est utilisée pour deux ordres différents.

Les figures (III.15a) à (III.15d) et (III.16a) à (III.16d) illustrent les résultats obtenus.

lorsque l'ordre est faible, on remarque que les bandes de transition sont assez importantes, notamment pour le filtre multi-passe bandes, dont la première et la seconde bandes sont pratiquement confondues. pour un ordre plus élevé, les bandes de transition rétrécissent et l'approximation du filtre est meilleure.

-Exemple 6 :

On veut synthétiser un filtre passe haut de fréquence de coupure normalisée 0.3 . On utilise le critère de comparaison des puissances moyennes. Pour différentes valeurs de la tolérance (définie suivant la relation III.21), on obtient le tableau III.5 .

Les fenêtres utilisées sont celles de BLACKMAN et RIEMANN. Les caractéristiques de la fenêtre de RIEMANN sont exposées dans l'annexe 1.

Tableau III.5. Evolution de l'ordre N en fonction de la tolérance

Tolérance %	ORDRE N	
	BLACKMAN	RIEMANN
30	11	11
25	13	11
15	19	13
10	27	19
8	35	23
5	53	35
3	87	57
1	259	169

Pour la première itération, N est initialisé à 11. L'obtention d'une bonne approximation de filtre désiré nécessite un nombre de coefficients important.

Notons que pour toutes les valeurs de N, l'atténuation en bande coupée reste constante pour une fenêtre donnée, elle est plus importante pour la fenêtre de BLACKMAN.

En termes de puissance, nous pouvons dire que la fenêtre de RIEMANN permet une meilleure approximation du filtre désiré. Pour un ordre donné, la puissance moyenne filtre limité par la fenêtre de RIEMANN est plus proche de la puissance moyenne du filtre désiré.

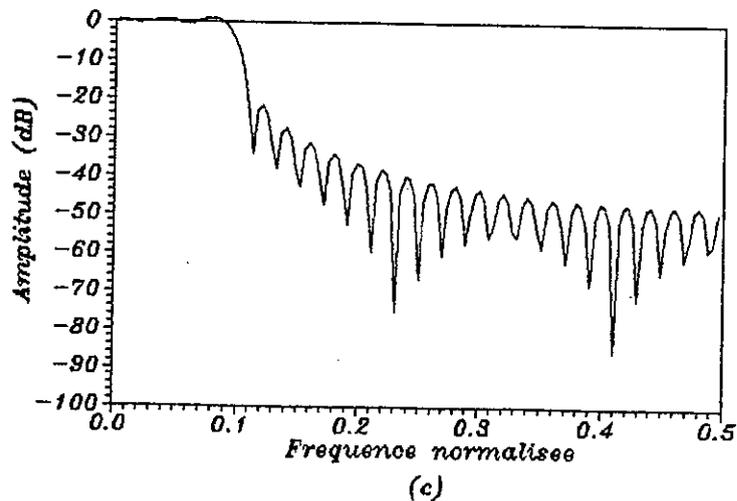
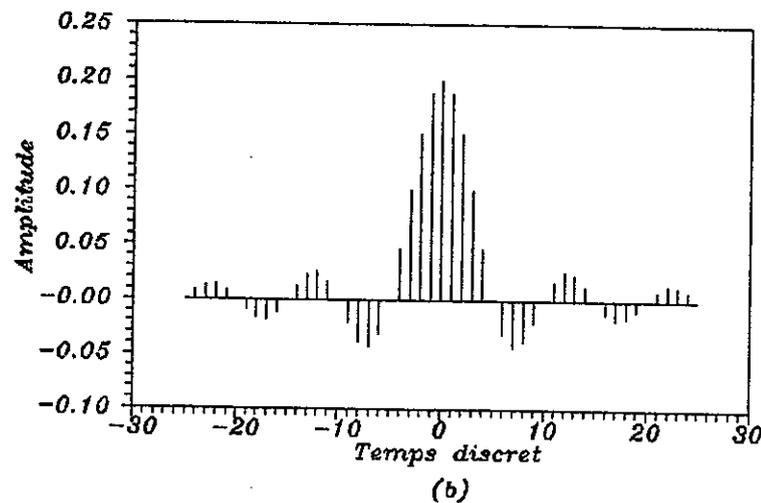
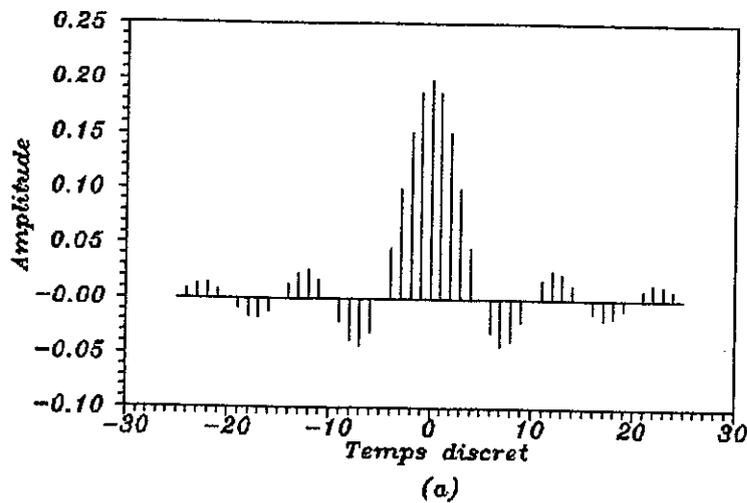


Fig. 3.8.a
 Filtre passe bas limite
 par
 la fenetre rectangulaire
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 $N = 51$

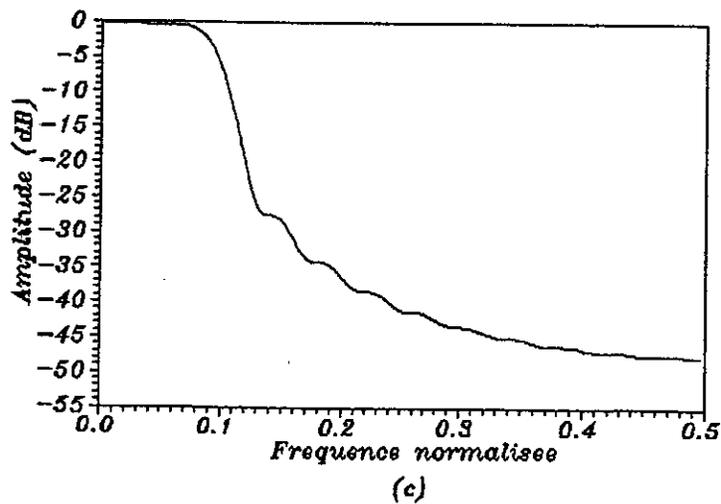
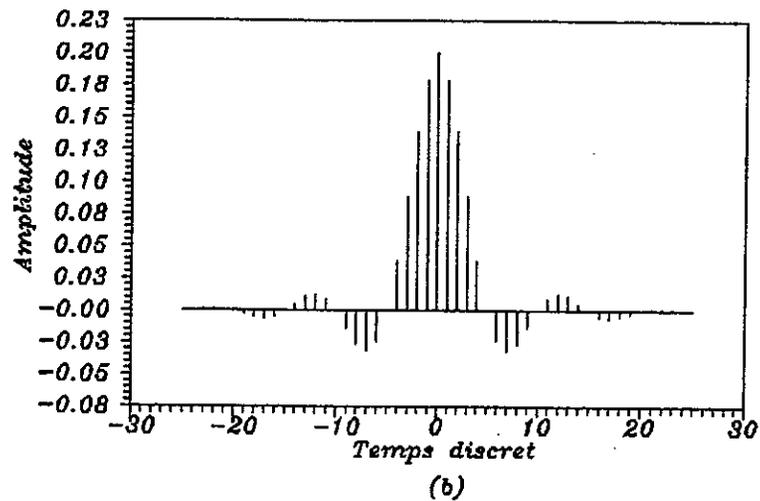
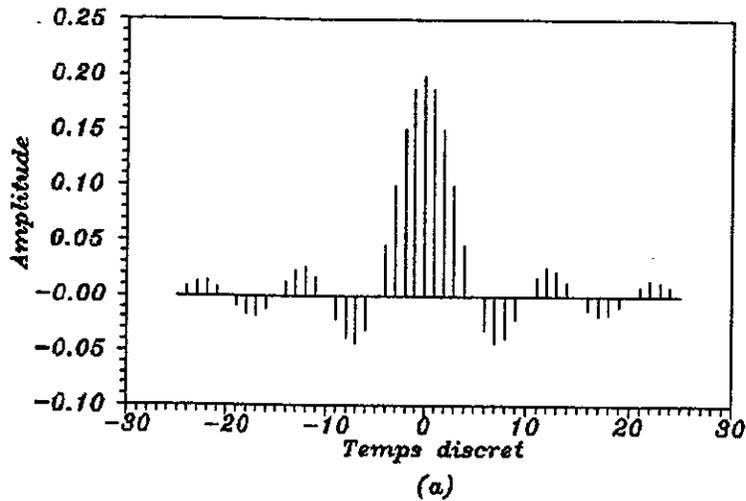


Fig. 3.8.b
 Filtre passe bas limite
 par
 la fenetre triangulaire
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 $N = 51$

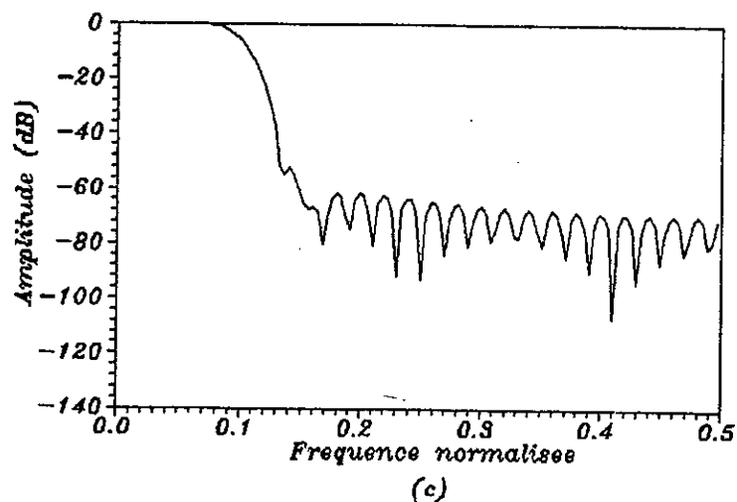
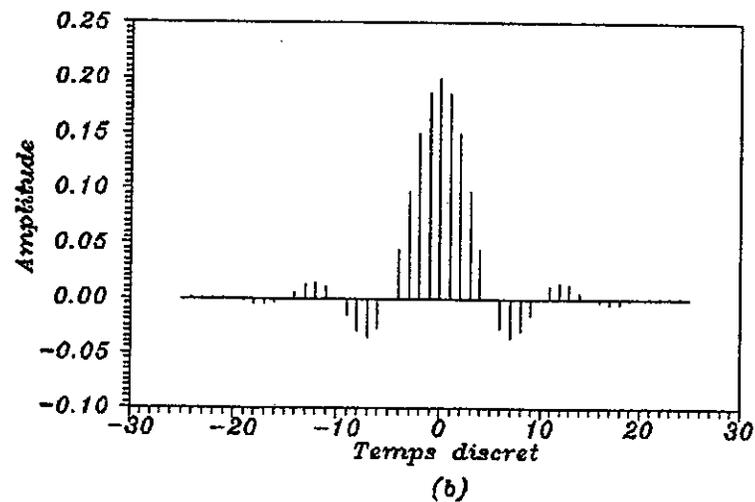
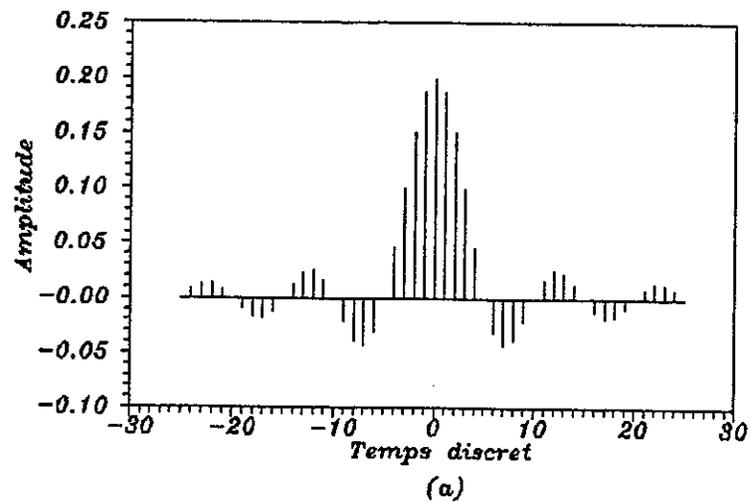


Fig. 3.8.c
 Filtre passe bas limite
 par
 la fenetre de Hamming
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 $N = 51$

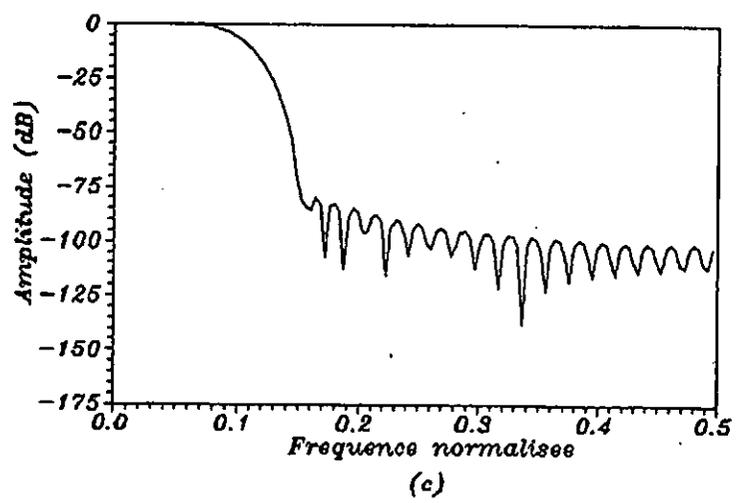
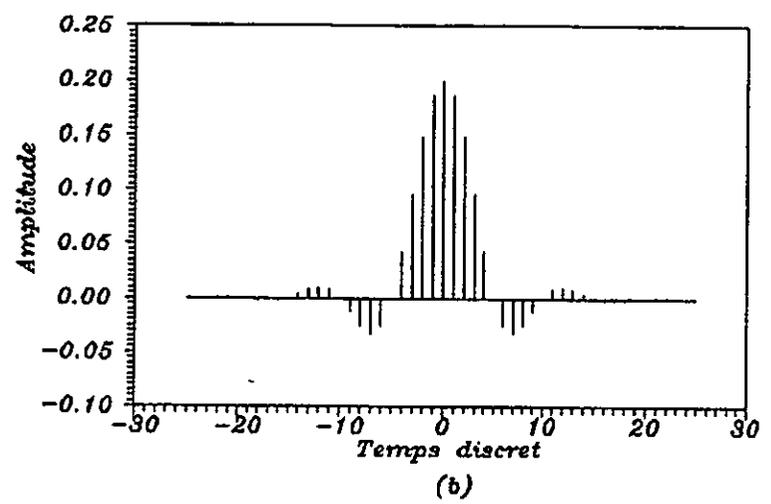
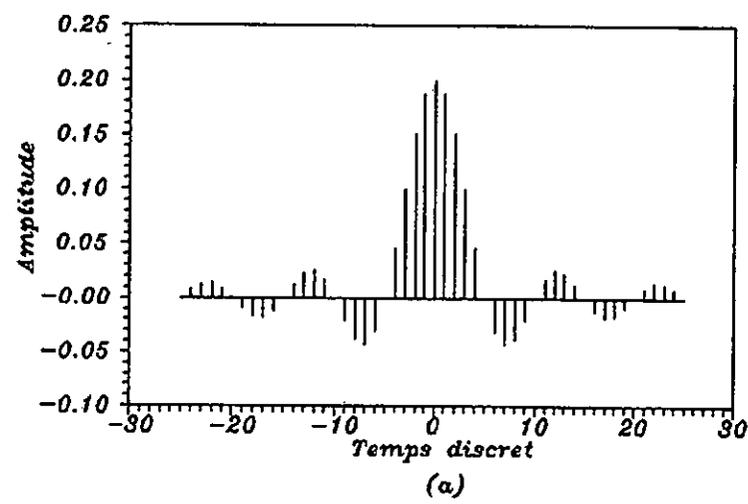


Fig.3.8.d
 Filtre passe bas limite
 par
 la fenetre de Kaiser $\alpha = 2.5 \cdot \pi$
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 $N = 51$

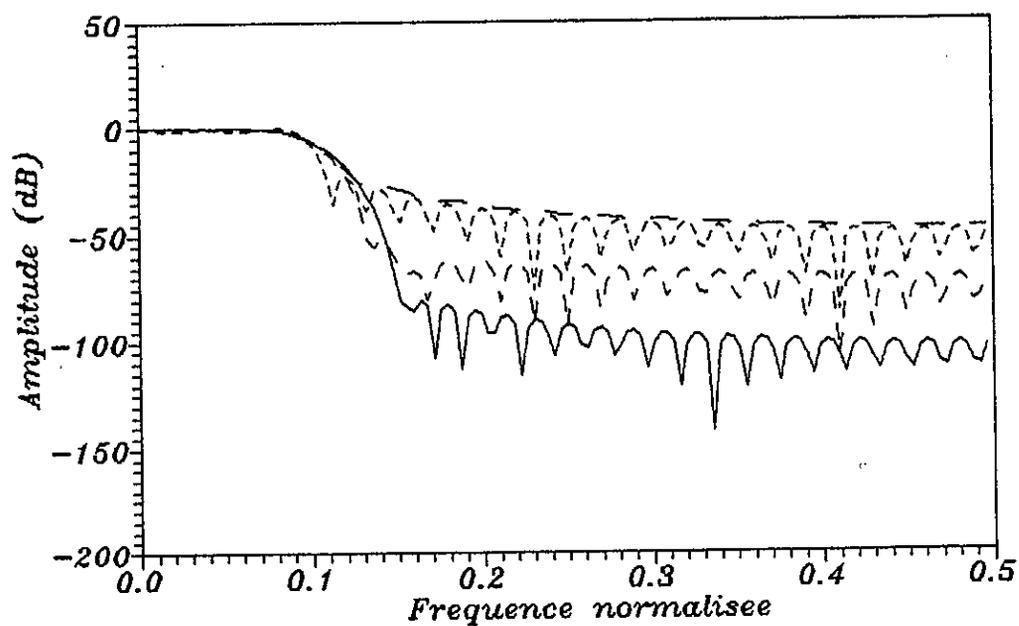


fig 3.8.e
 Spectre d'amplitude du filtre passe bas
 limité par la fenêtre

----- Rectangulaire
 - - - Triangulaire
 - - - de Hamming
 ——— de Kaiser ($\alpha=2\pi$)
 $N=51$ $W_c=0.1$

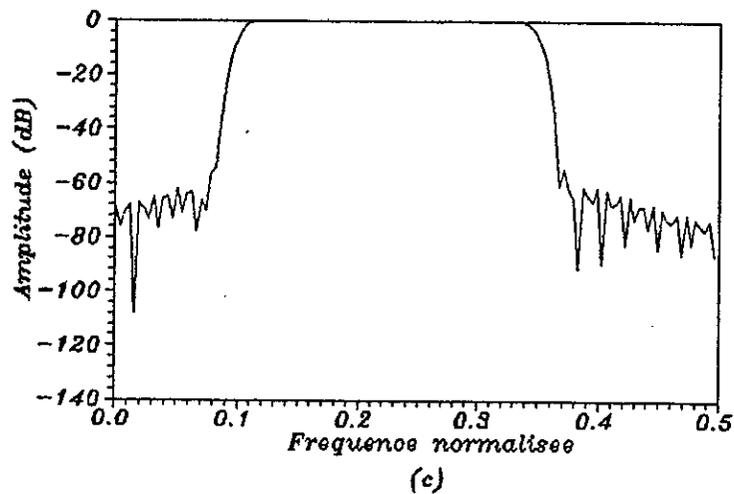
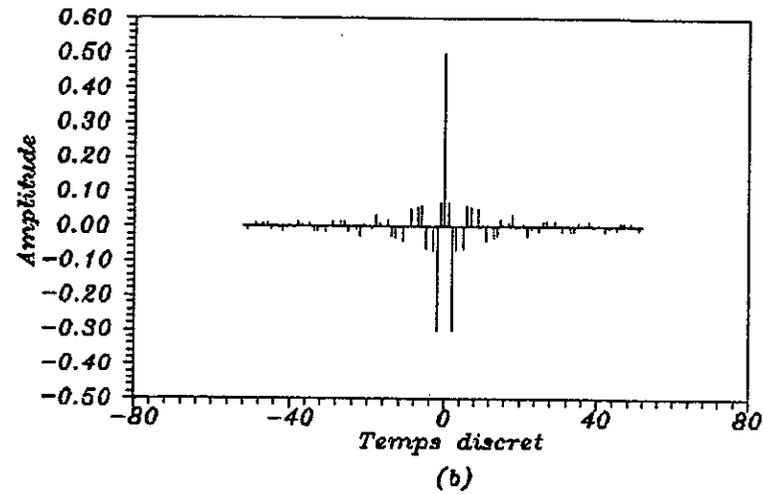
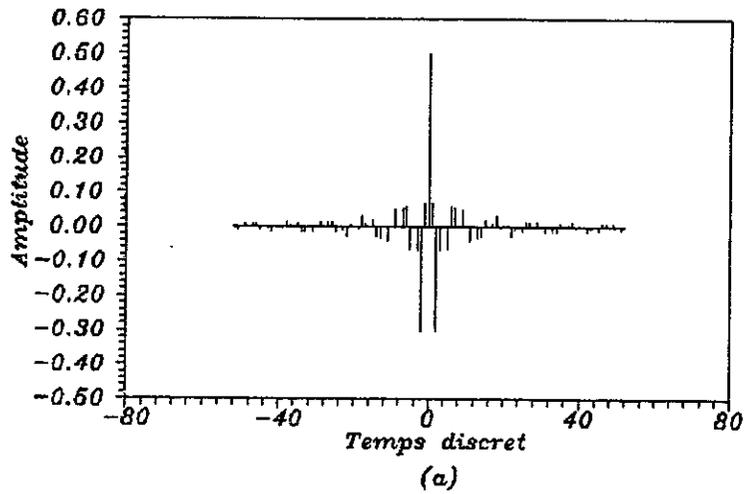


fig 3-9-a
 Filtre passe bande limite
 par
 la fenetre de Hamming
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequences de coupure normalisees :
 $Wc1=0.1$ $Wc2=0.35$
 $N=105$

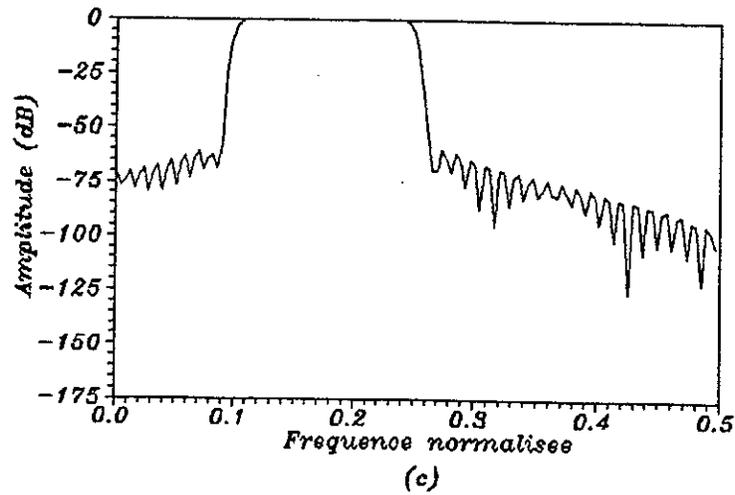
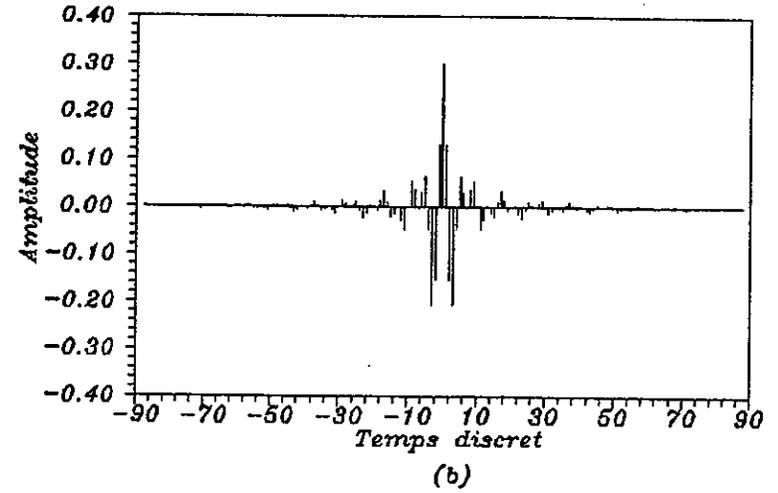
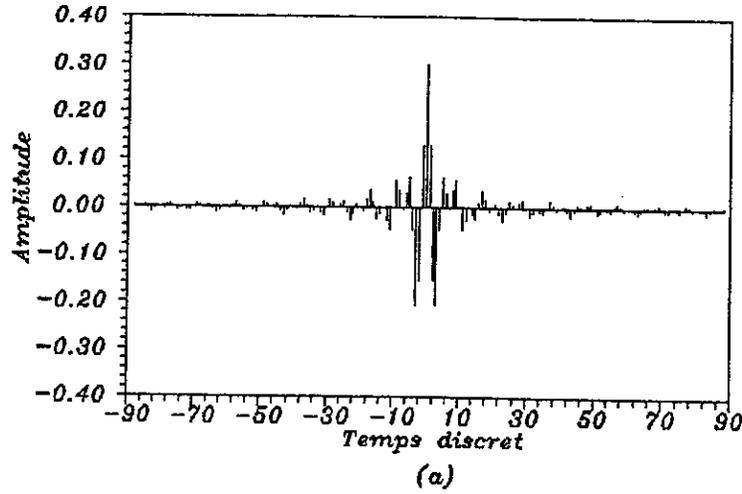
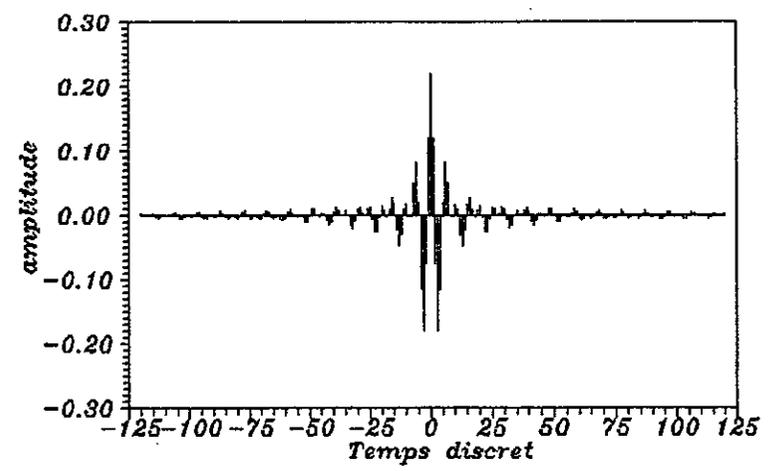
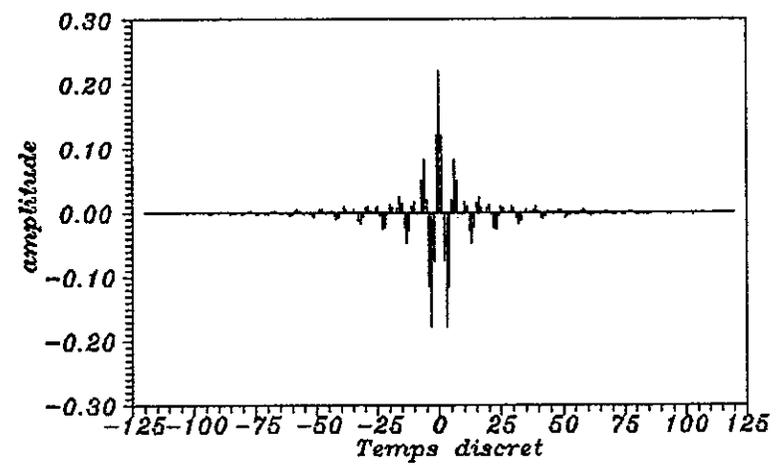


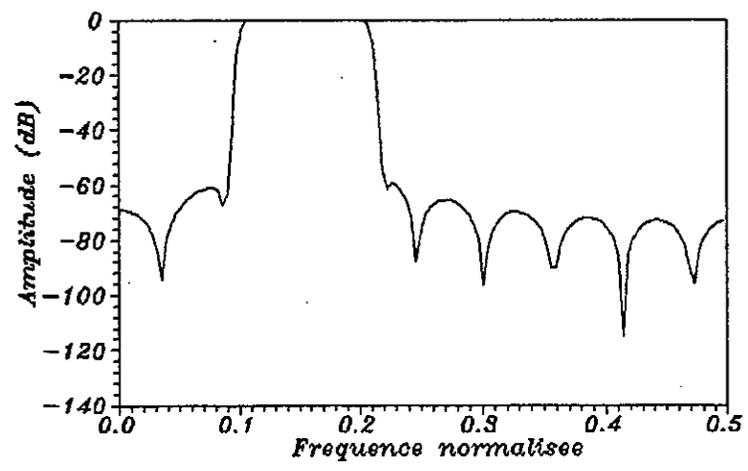
fig 3.9.b
 Filtre passe bande limite
 par
 la fenetre de Hamming
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequences de coupure normalisees :
 $Wc1=0.1$ $Wc2=0.25$
 $N=177$



(a)



(b)



(c)

Fig 3.9-c
 Filtre passe bande limite
 par
 la fenetre de Hamming
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequences de coupure normalisees :
 $Wc1=0.1$ $Wc2=0.21$
 $N=241$

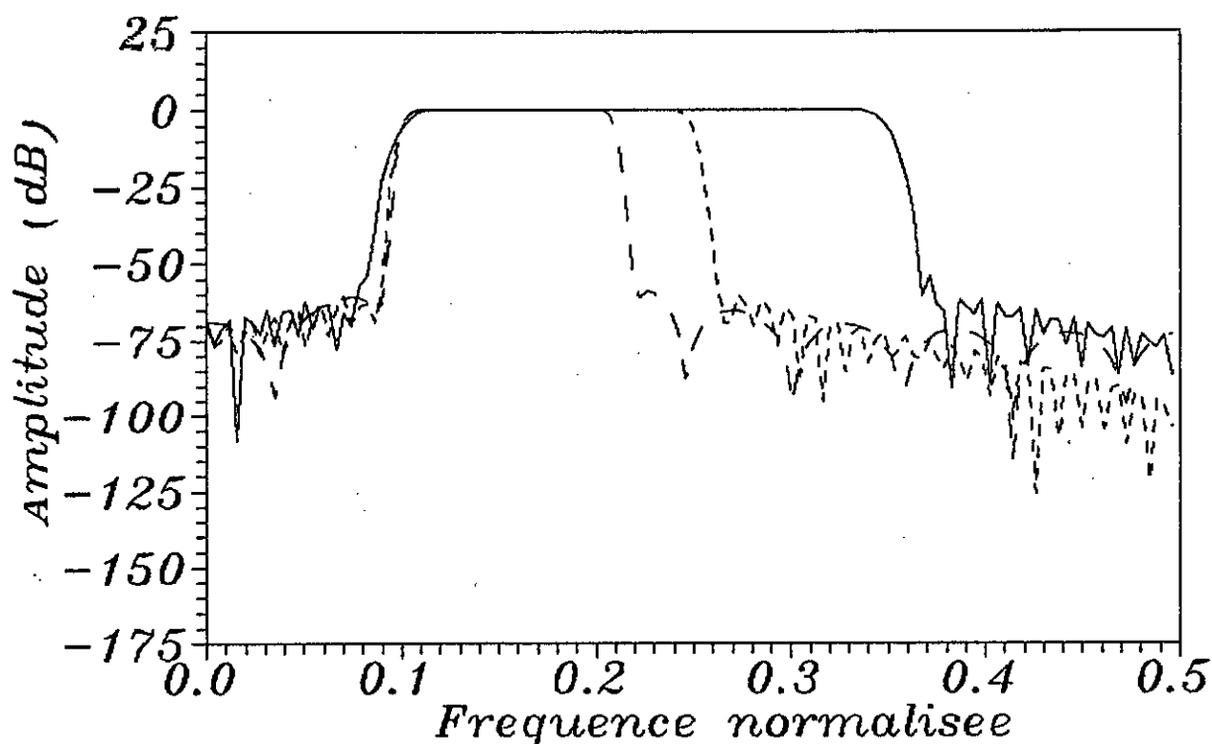


Fig 3.9.d

Spectre d'amplitude du filtre passe bande limite par la fenetre de hamming

—	$Wc1=0.1$	$Wc2=0.35$	$N=105$
----	$Wc1=0.1$	$Wc2=0.25$	$N=177$
- - -	$Wc1=0.1$	$Wc2=0.21$	$N=241$

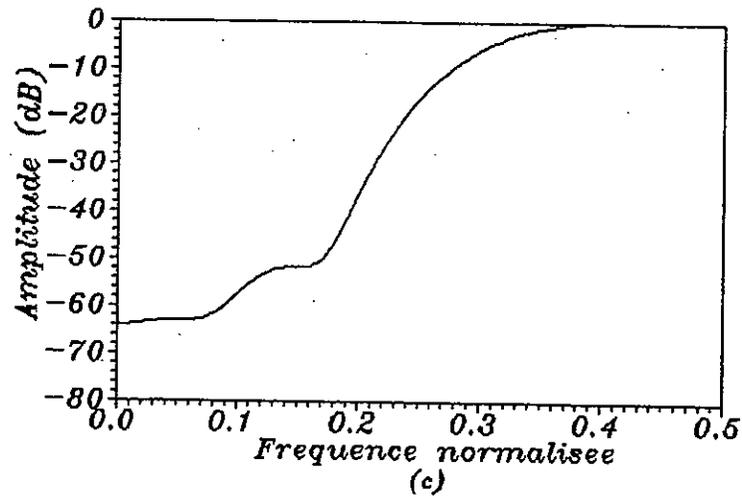
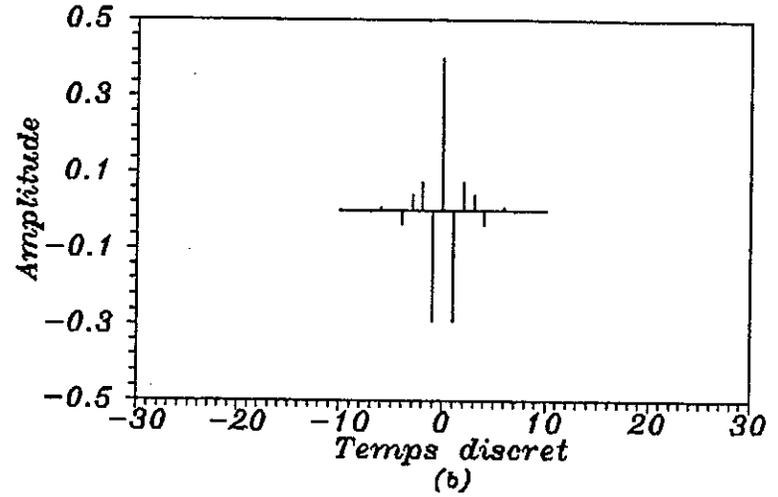
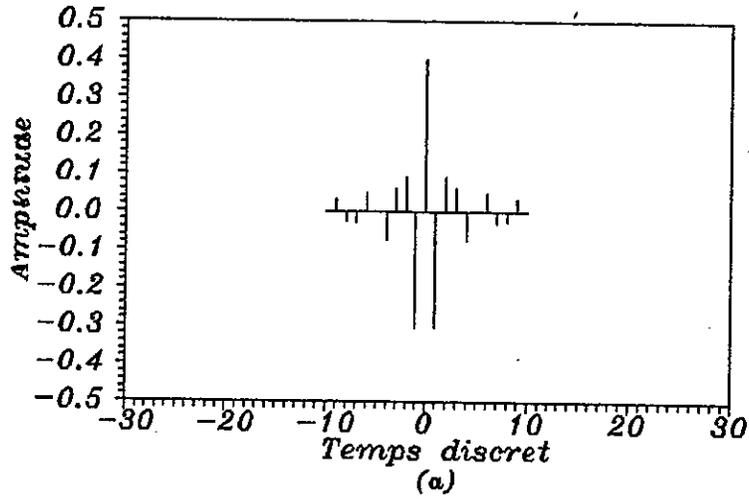


Fig. 3.10.a
 Filtre passe haut limite
 par
 la fenetre de Bohman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 $N = 21$

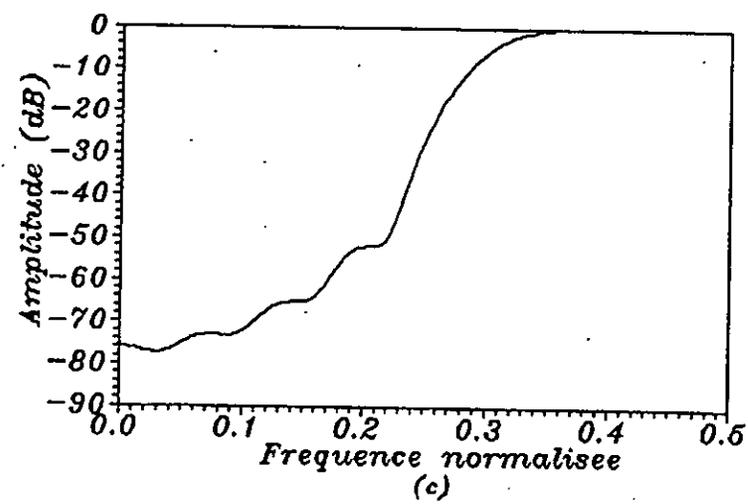
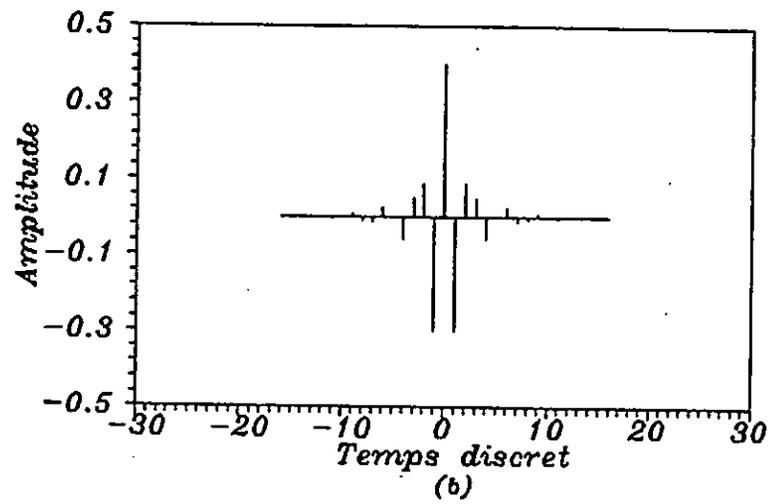
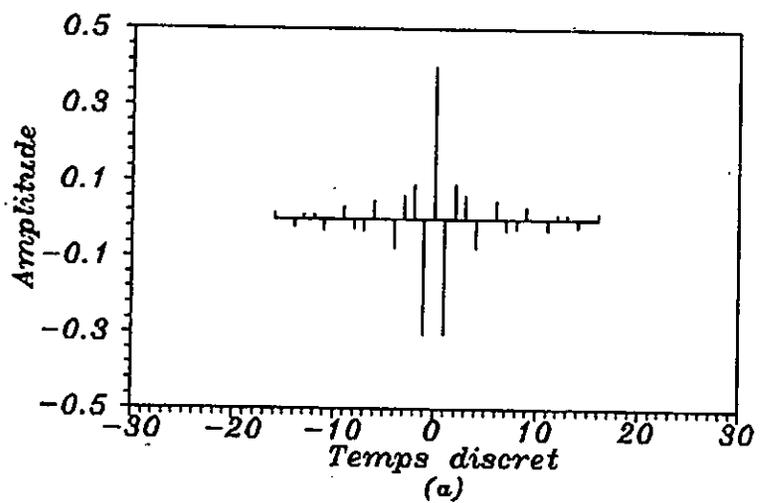


Fig. 3.10.b
 Filtre passe haut limite
 par
 la fenetre de Bohman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 N = 33

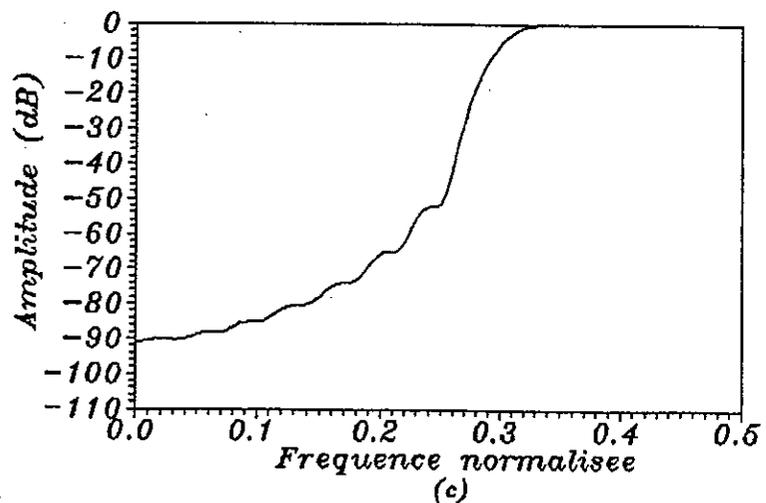
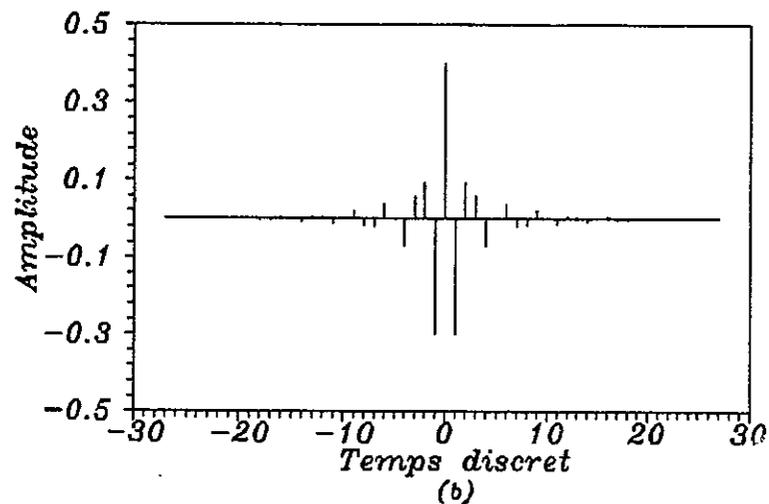
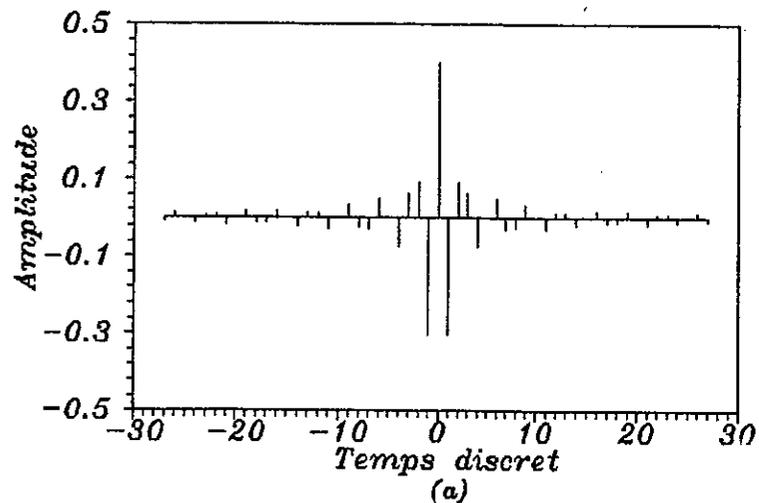
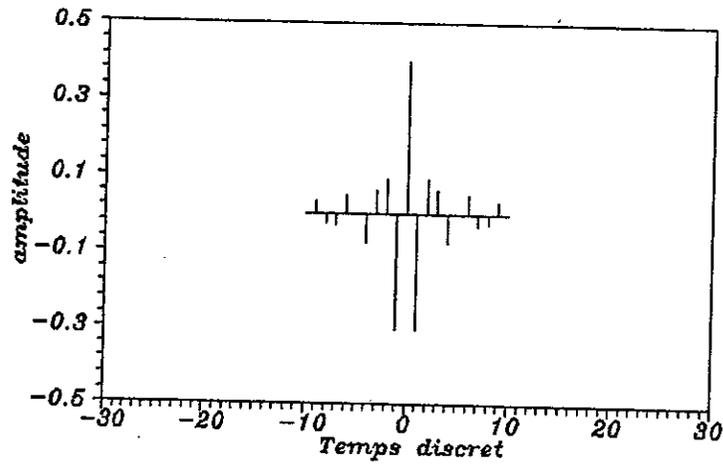
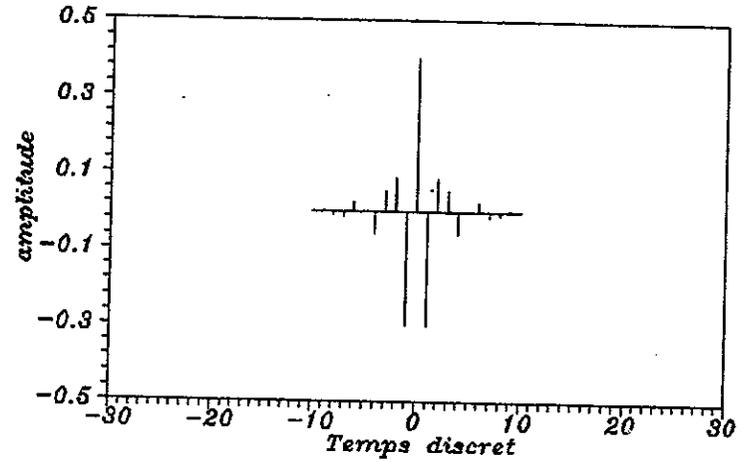


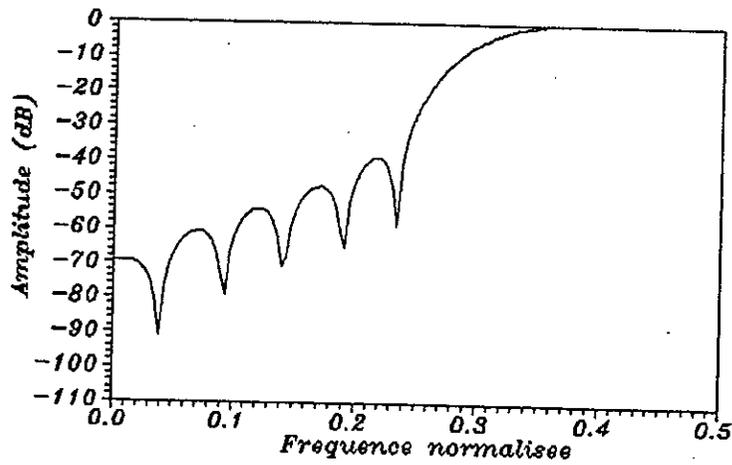
Fig. 3.10.C
 Filtre passe haut limite
 par
 la fenetre de Bohman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 $N = 55$



(a)

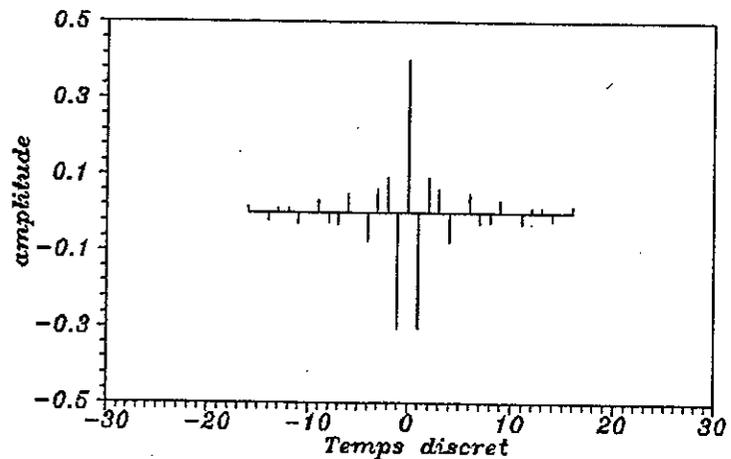


(b)

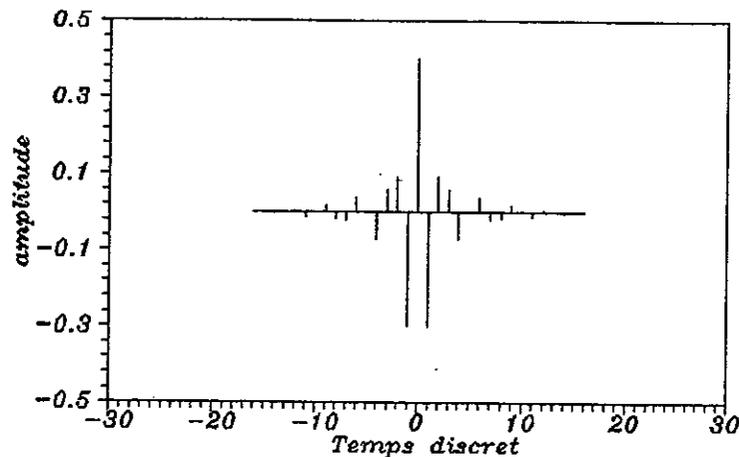


(c)

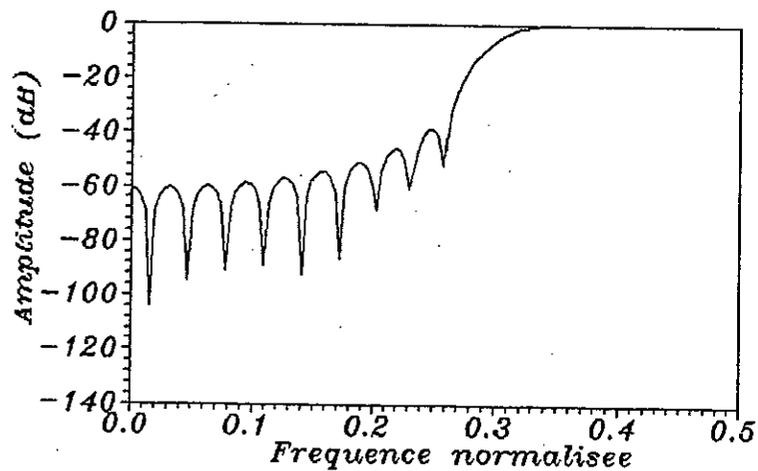
Fig. 3.11.a
 Filtre passe haut limite
 par
 la fenetre de Riemann
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 N=21



(a)

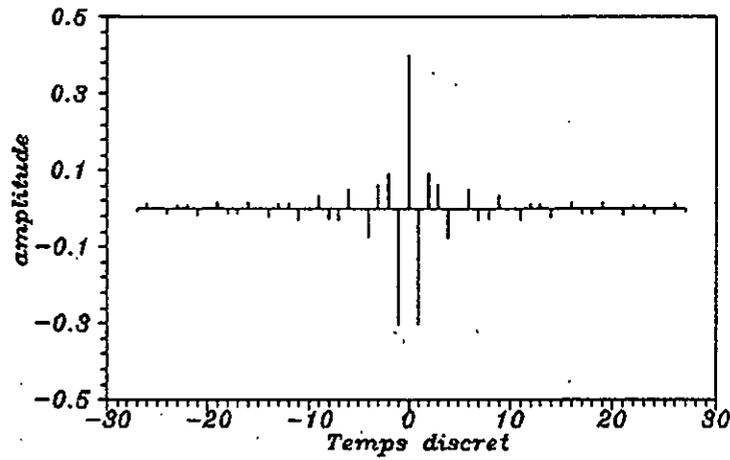


(b)

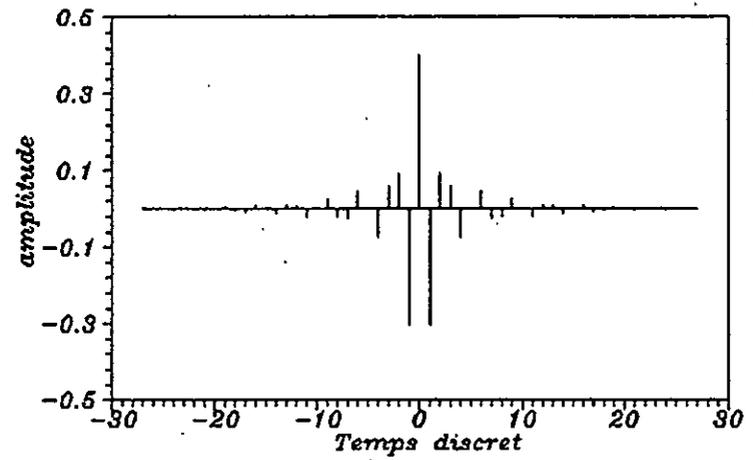


(c)

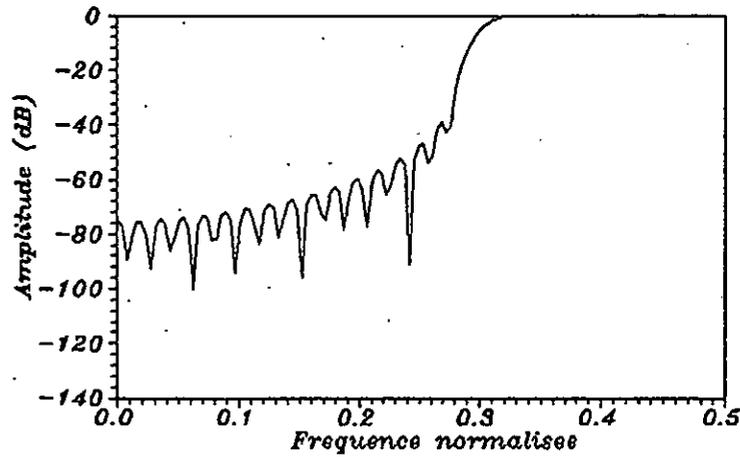
Fig. 3.11.b
 Filtre passe haut limite
 par
 la fenetre de Riemann
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 $N = 33$



(a)



(b)



(c)

Fig. 3.11.C
 Filtre passe haut limite
 par
 la fenetre de Riemann
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 N=55

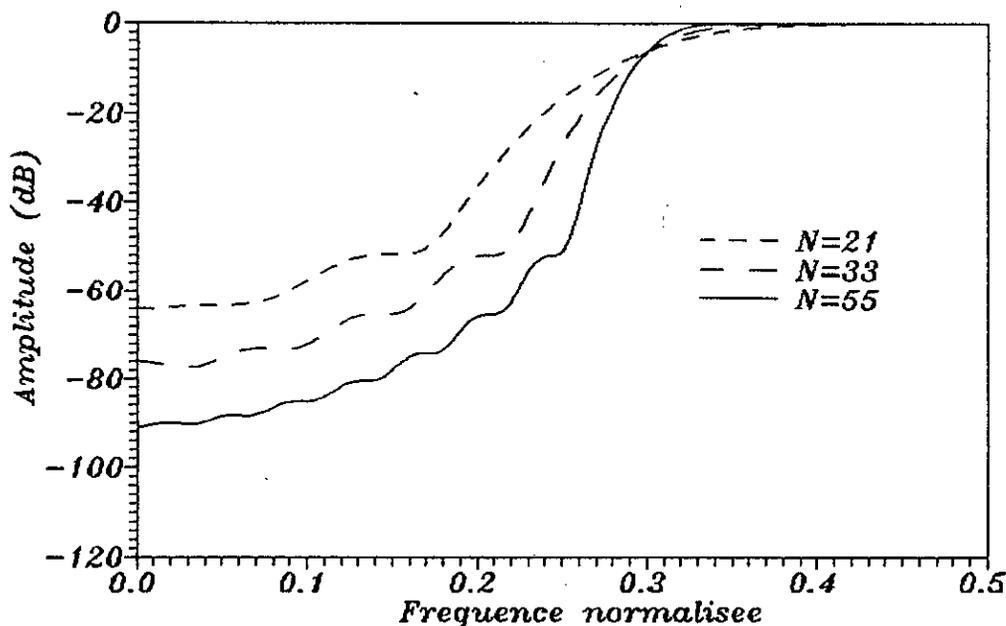


fig. 3.10. d
 Spectre d'amplitude du filtre passe haut
 limite par la fenetre de BOHMAN

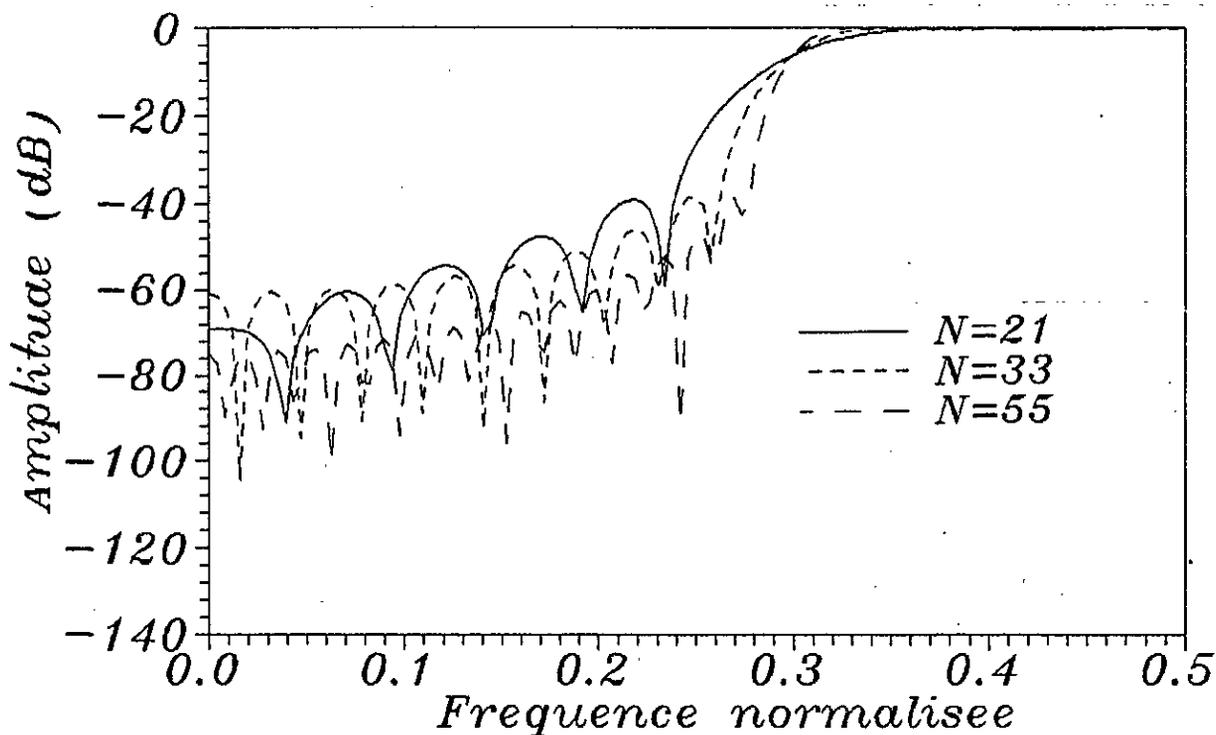


Fig. 3.11. d
 Spectre d'amplitude du filtre passe haut
 limite par la fenetre de Riemann

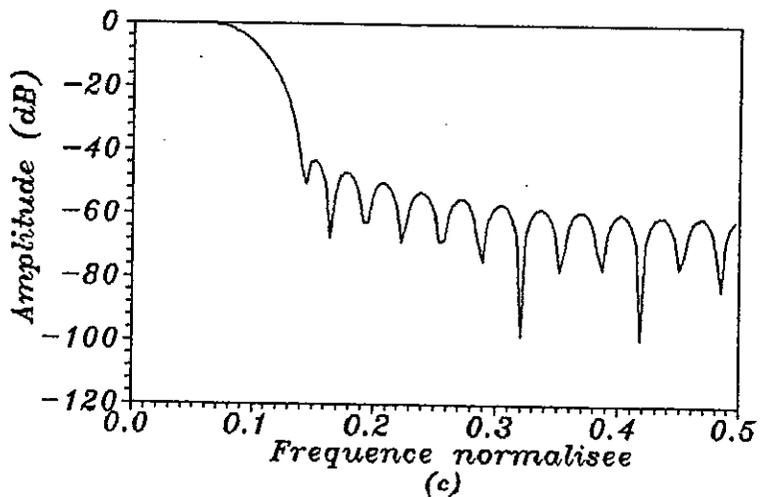
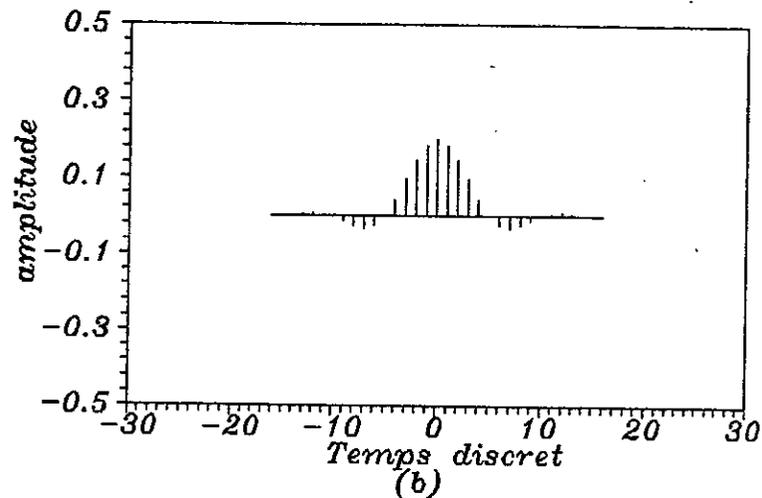
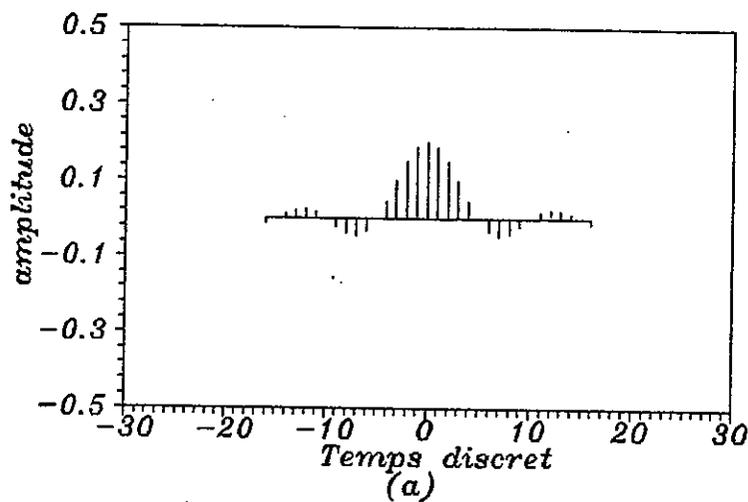


Fig 3.12. a
 Filtre passe bas limite
 par
 la fenetre de Kaiser
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 alpha=4 N = 33

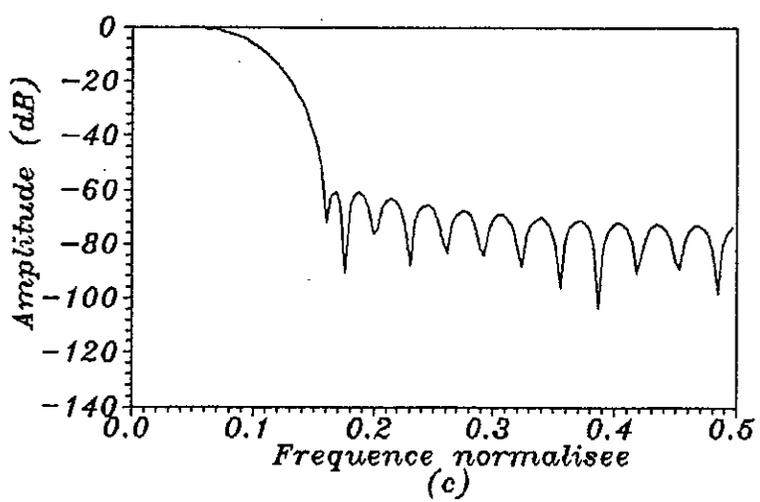
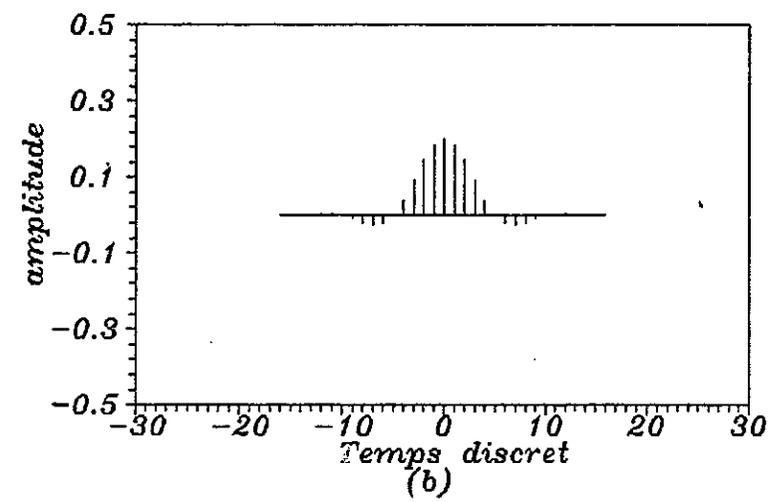
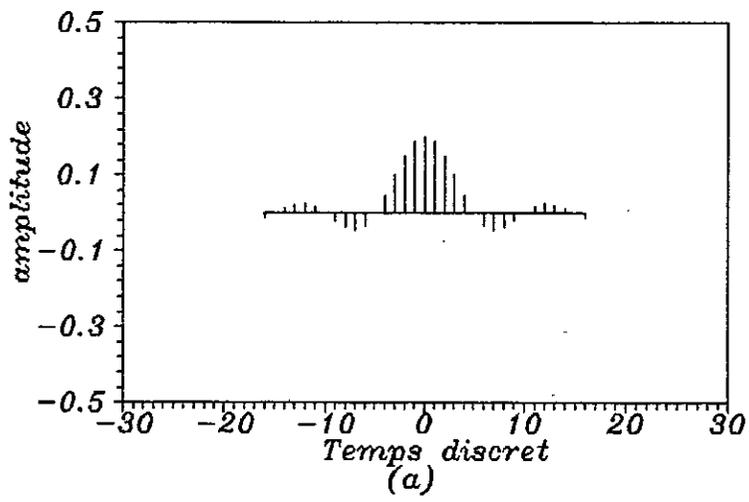


Fig 3-12-b
 Filtre passe bas limite
 par
 la fenetre de Kaiser
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 alpha=6 N = 33

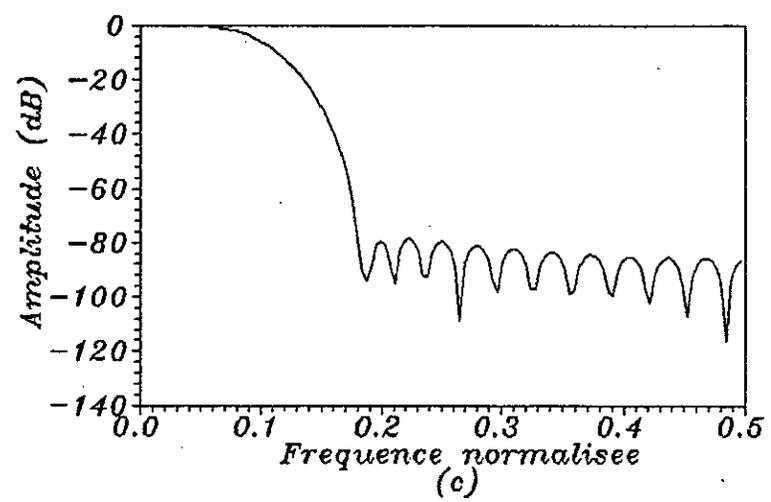
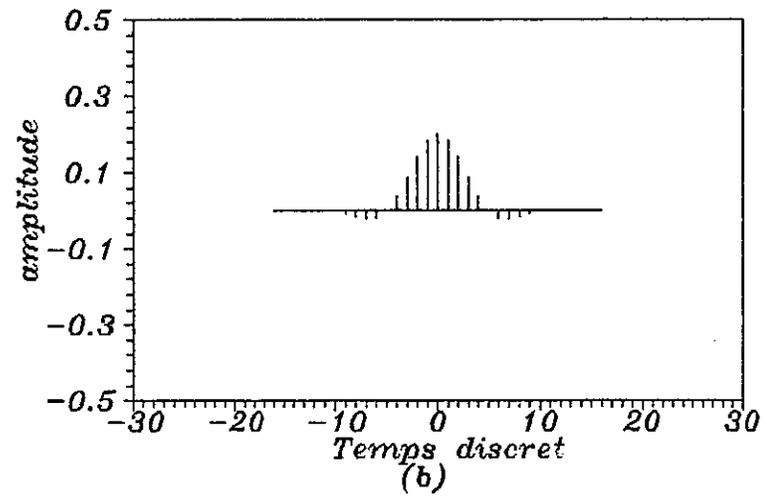
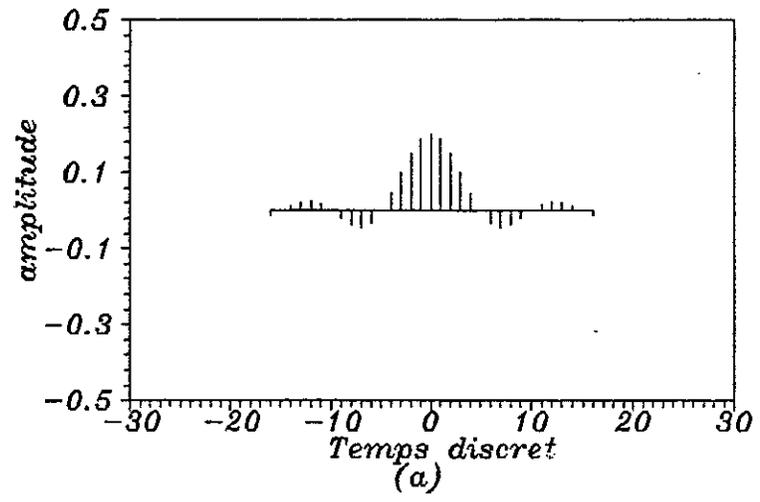


Fig 3.12-c
 Filtre passe bas limité
 par
 la fenetre de Kaiser
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 alpha=8 N = 33

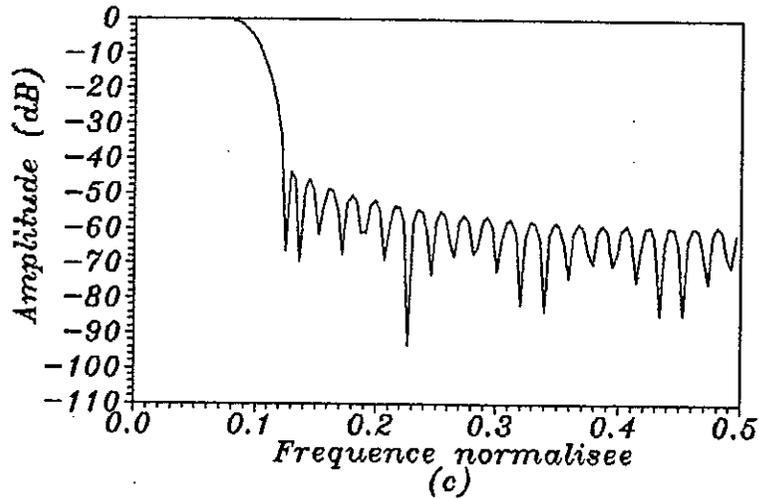
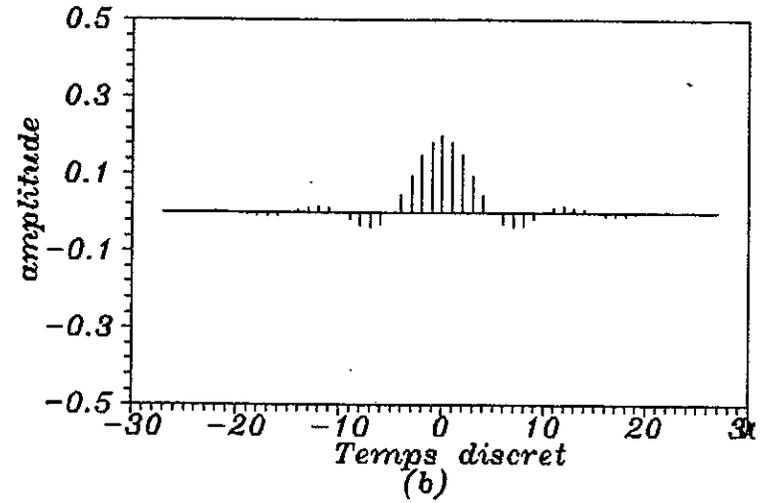
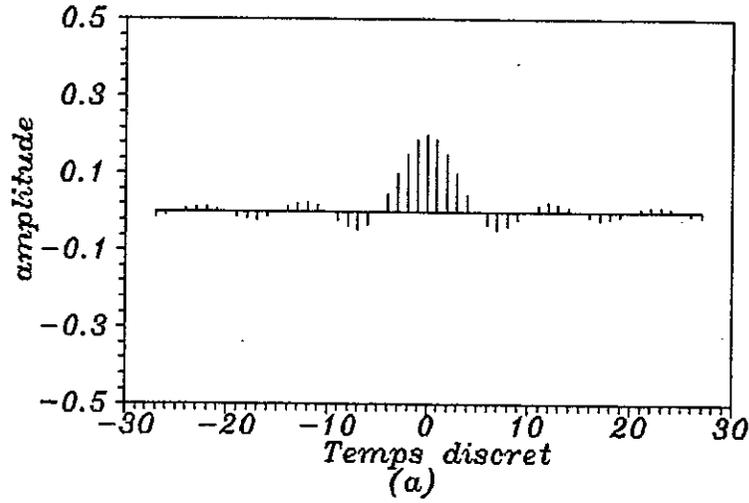


Fig 3.13. a
 Filtre passe bas limite
 par
 la fenetre de Kaiser
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 alpha=4 N = 55

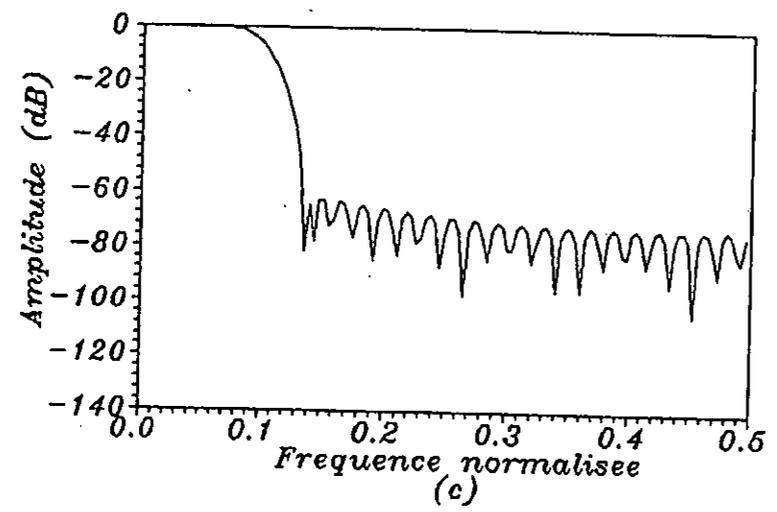
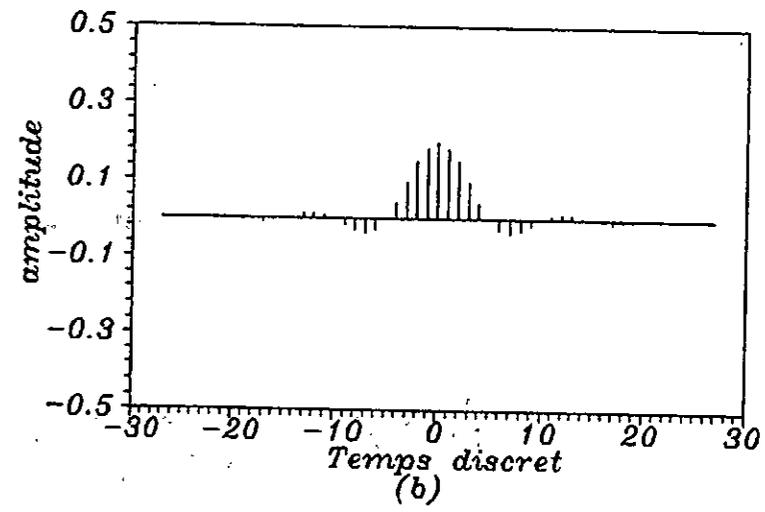
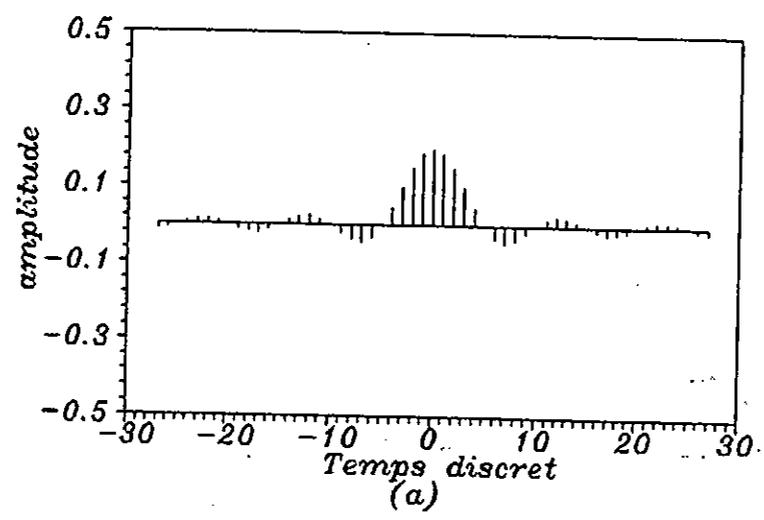


Fig 3.13.b
 Filtre passe bas limite
 par
 la fenetre de Kaiser
 a) Reponse impulsionnelle non fenetre
 b) Reponse impulsionnelle fenetre
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.1
 alpha=6 N = 55

09

19

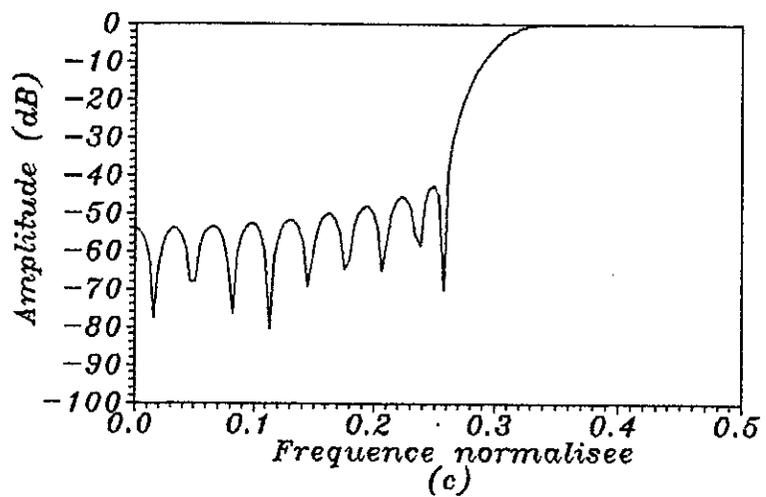
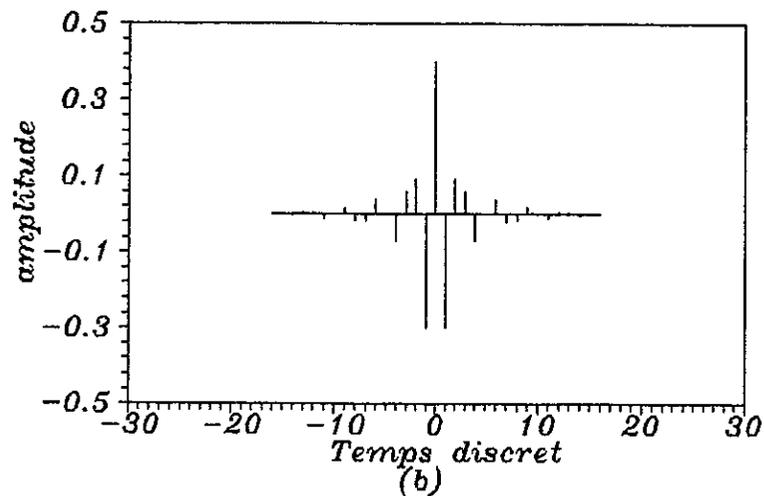
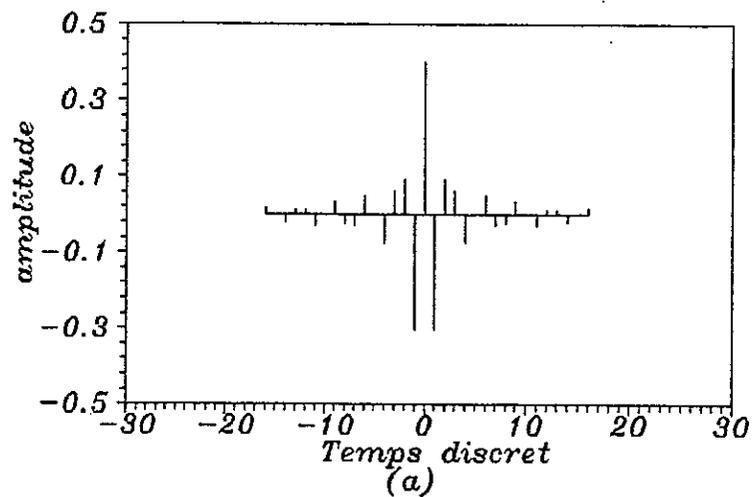


Fig 3-14-a
 Filtre passe haut limite
 par
 la fenetre de Kaiser
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 alpha=4 N = 33

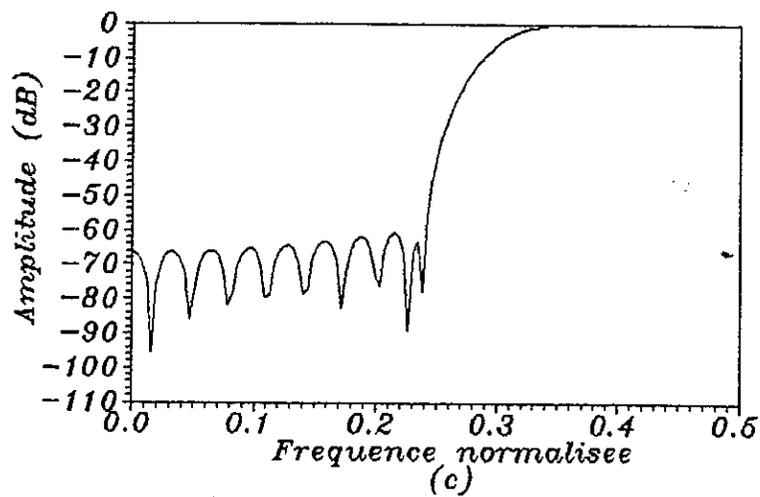
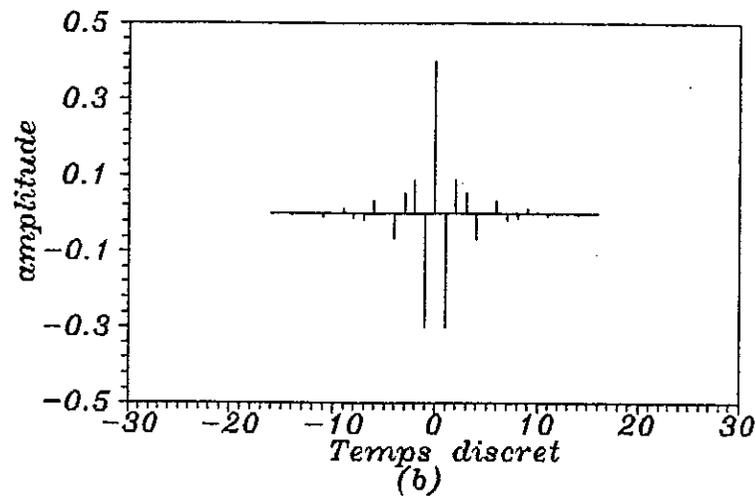
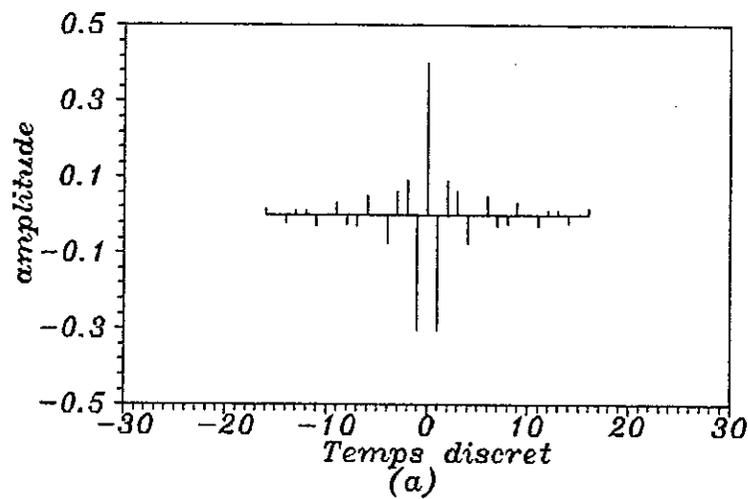


Fig 3.14.b
 Filtre passe haut limite
 par
 la fenetre de Kaiser
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 Gabarit :
 Frequence de coupure normalisee : 0.3
 alpha=6 N = 33

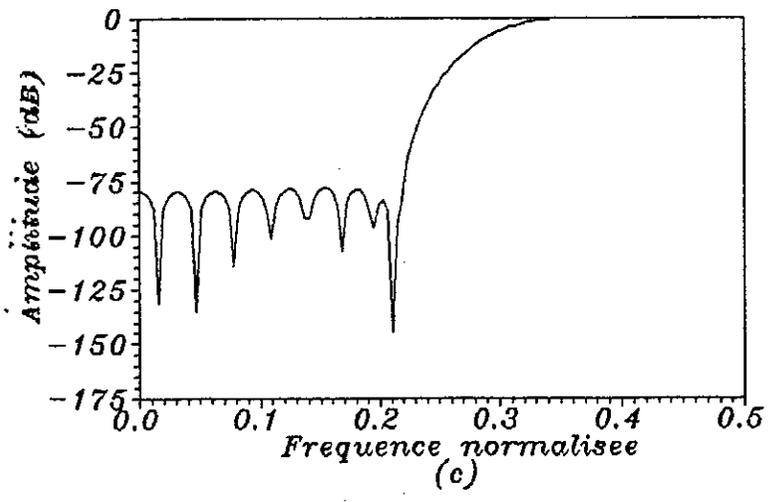
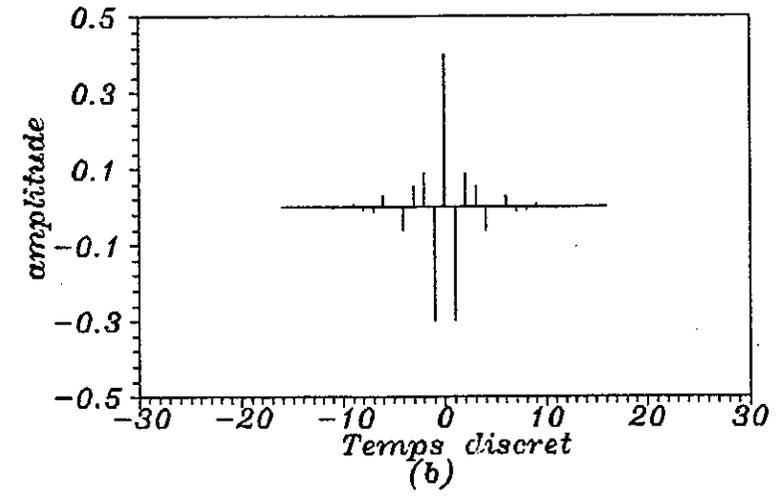
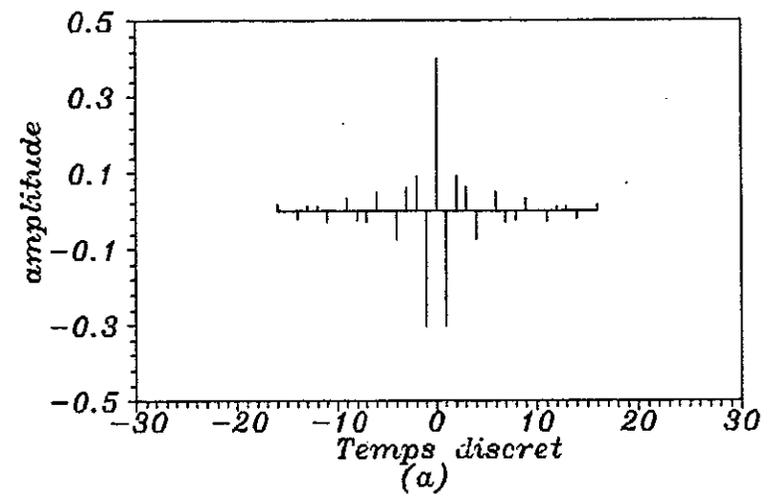


Fig 3-14.C
Filtre passe haut limite
par
la fenetre de Kaiser
a) Reponse impulsionnelle non fenetree
b) Reponse impulsionnelle fenetree
c) Spectre d'amplitude
Gabarit :
Frequence de coupure normalisee : 0.3
alpha=8 N = 33

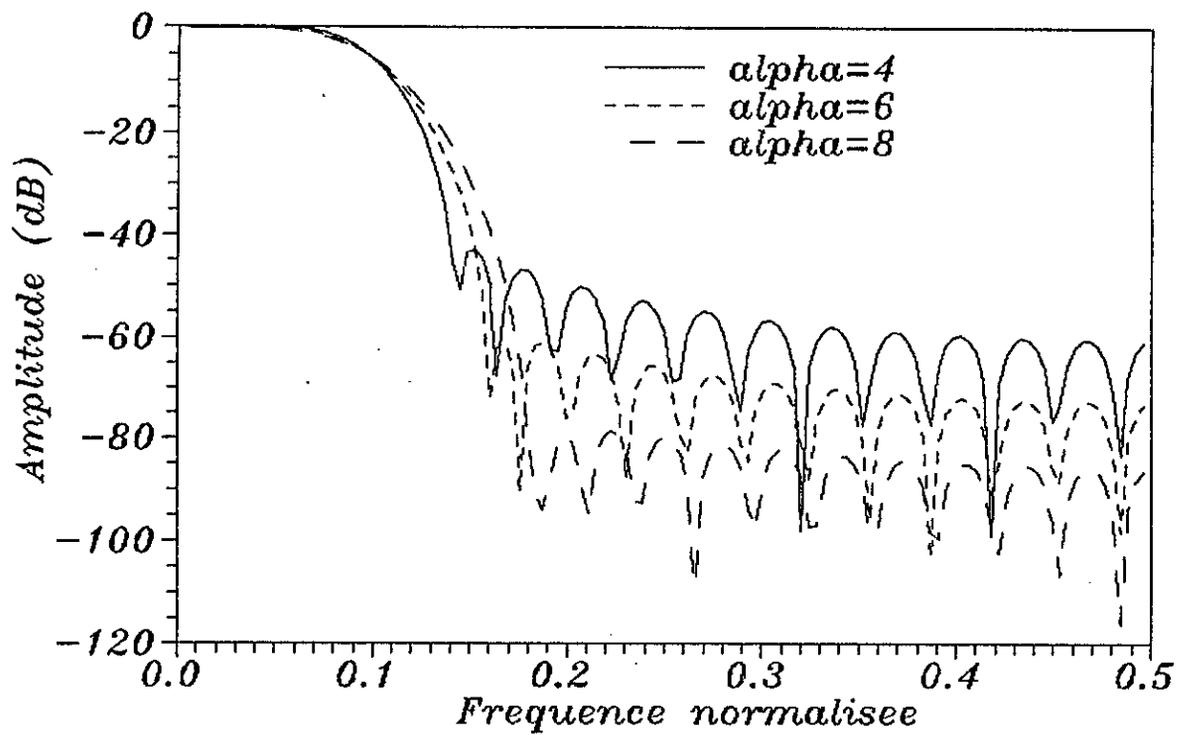


Fig 3.12 - d
 spectre d'amplitude du filtre passe
 bas limite par la fenetre de
 Kaiser
 frequence de coupure normalisee : 0.1
 N=33

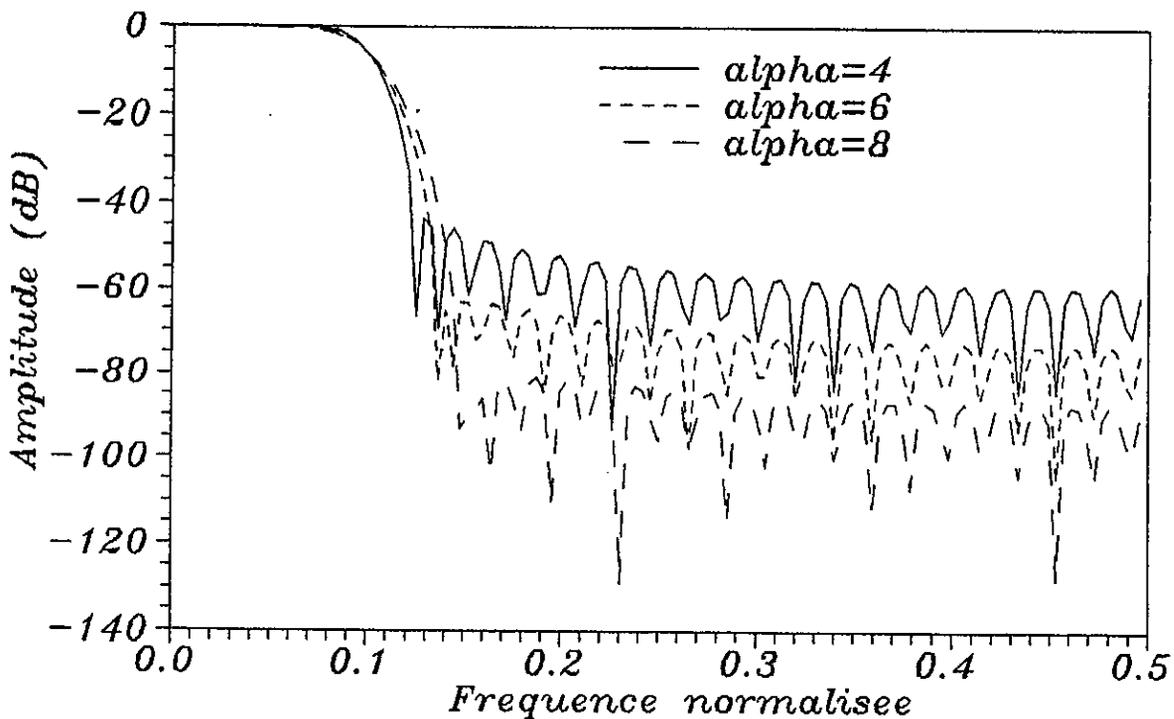


Fig 3.13 - d
 spectre d'amplitude du filtre passe
 bas limite par la fenetre de
 Kaiser
 frequence de coupure normalisee : 0.1
 N=55

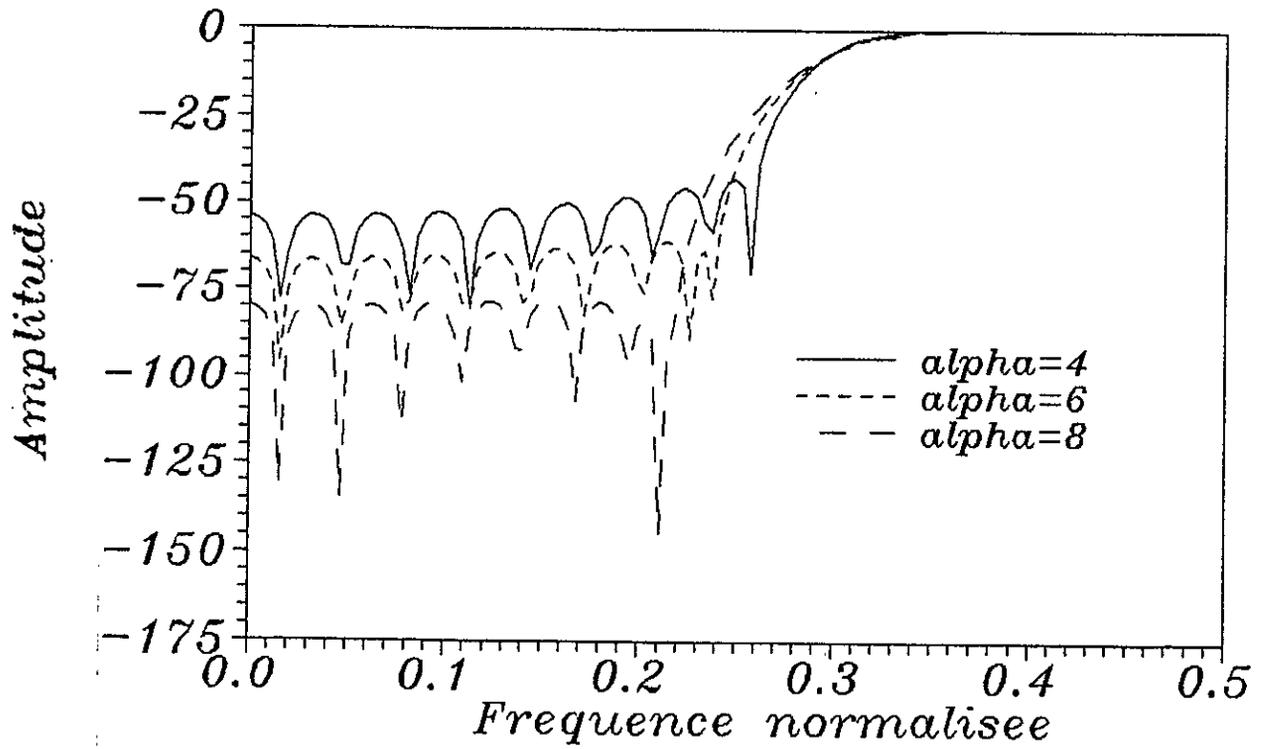


Fig 3-14-d
Spectre d'amplitude du filtre passe
haut limite par la fenetre de
Kaiser
frequence de coupure normalisee : 0.3
 $N=33$

67

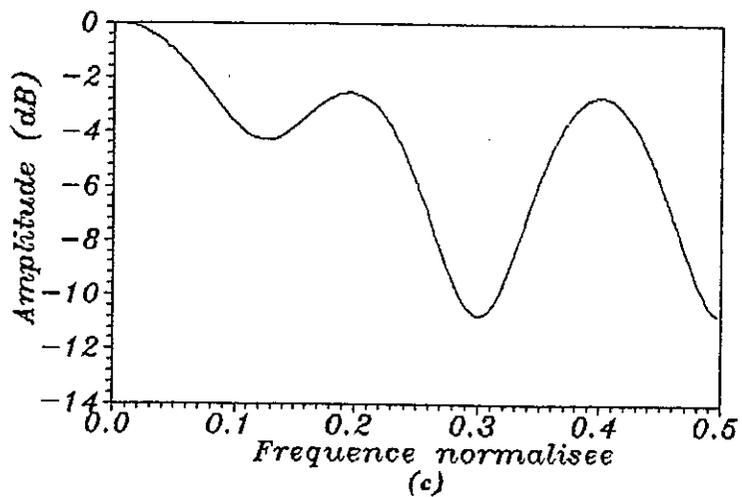
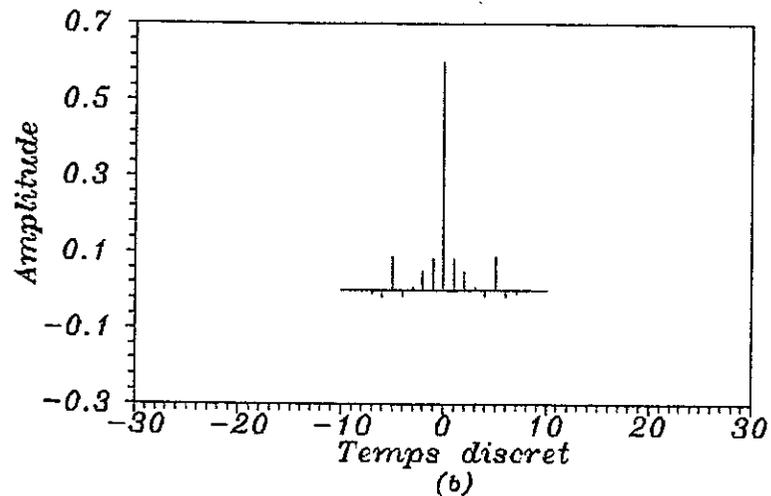
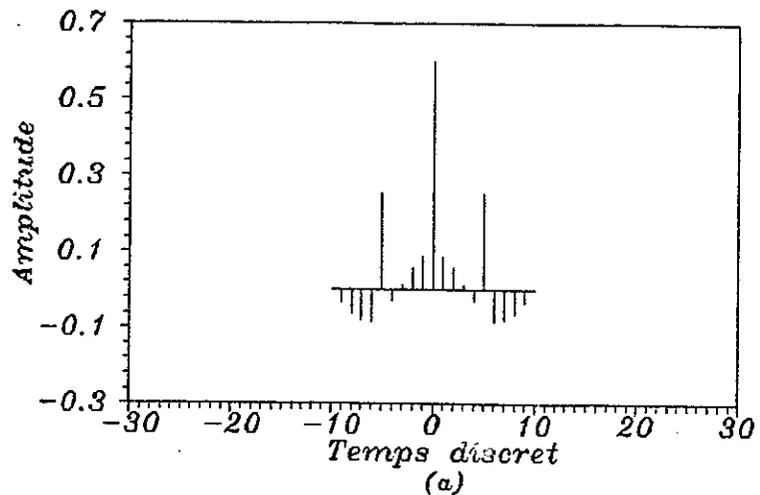


Fig. 3.15.α
 Filtre multi-passe bandes avec bande
 a l'origine limite par
 la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 $N=21$
 bande a l'origine: 0 - 0.1
 bande 1: 0.15 - 0.25
 bande 2: 0.35 - 0.45

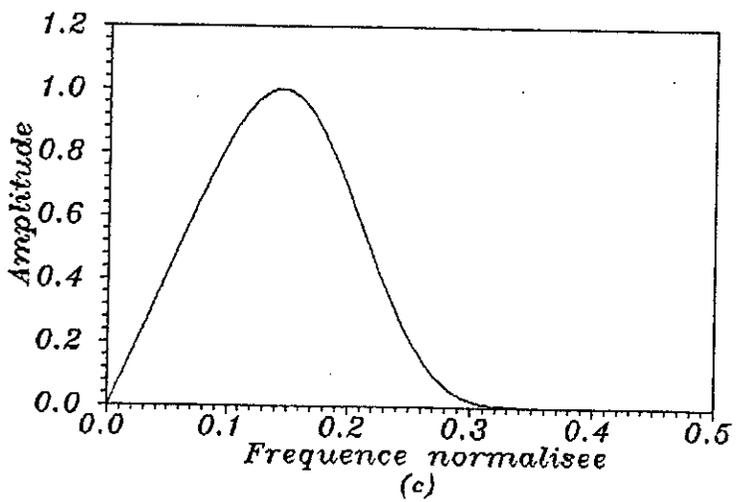
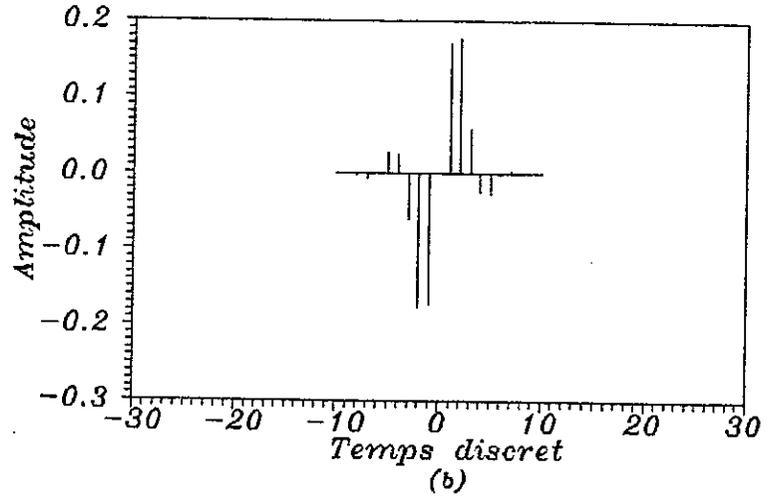
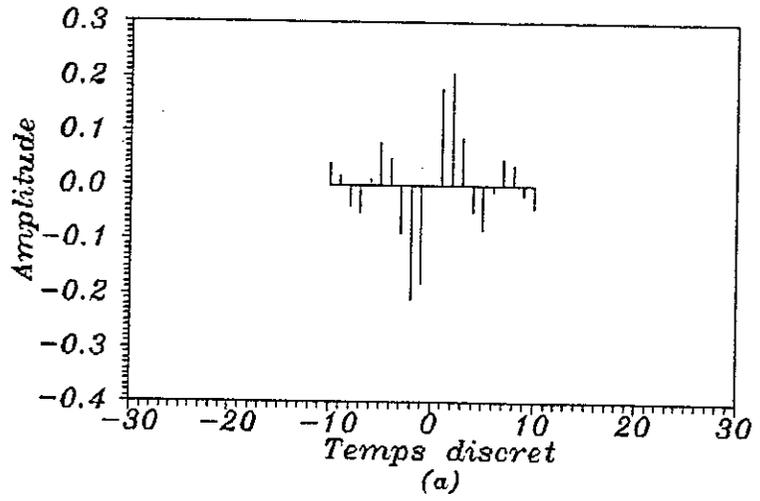


Fig. 3.15.b
 Filtre differentiateur
 limite par
 la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 $N=21$ $Wc=0.2$

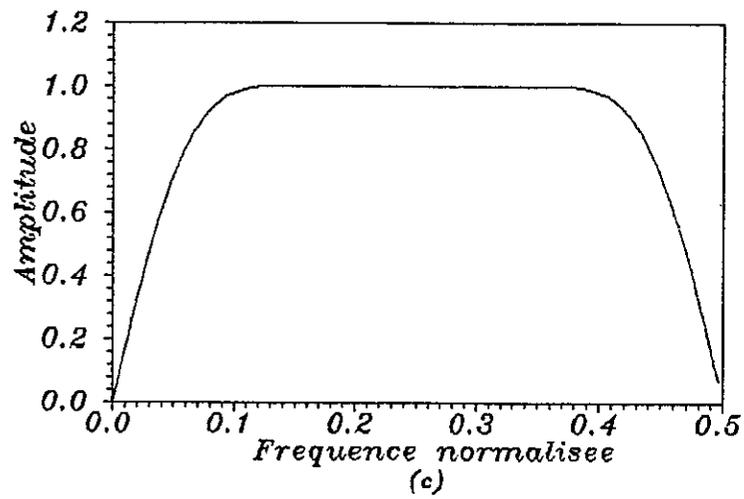
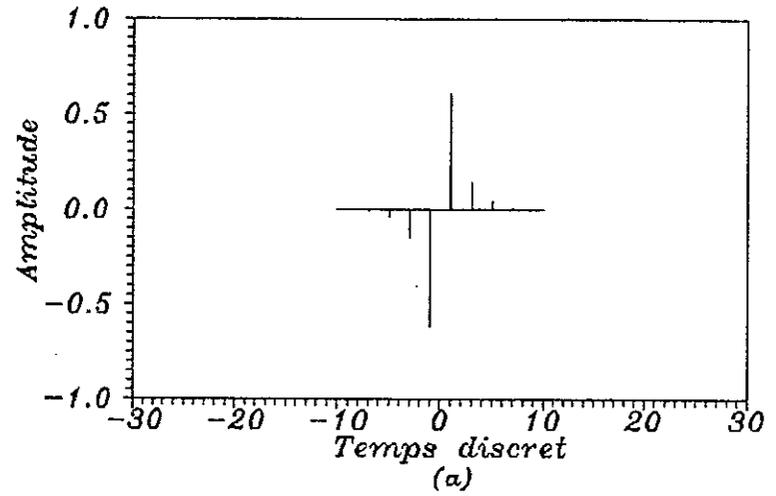
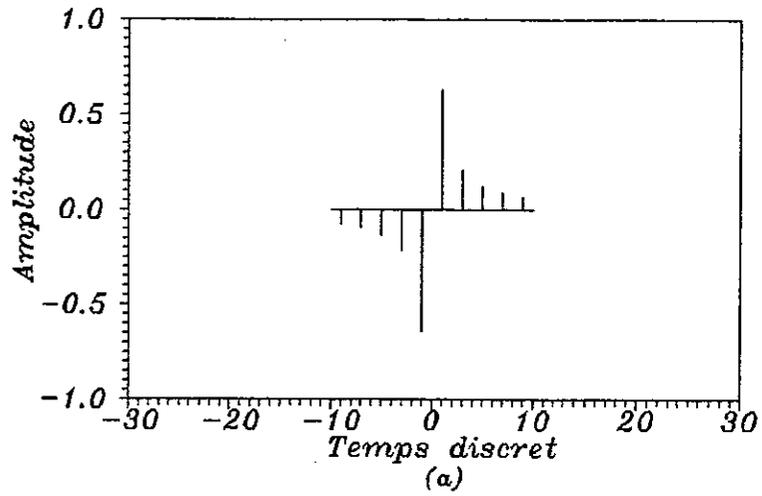


Fig. 3.45. c
 Transformateur de Hilbert
 limite par
 la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 $N=21$

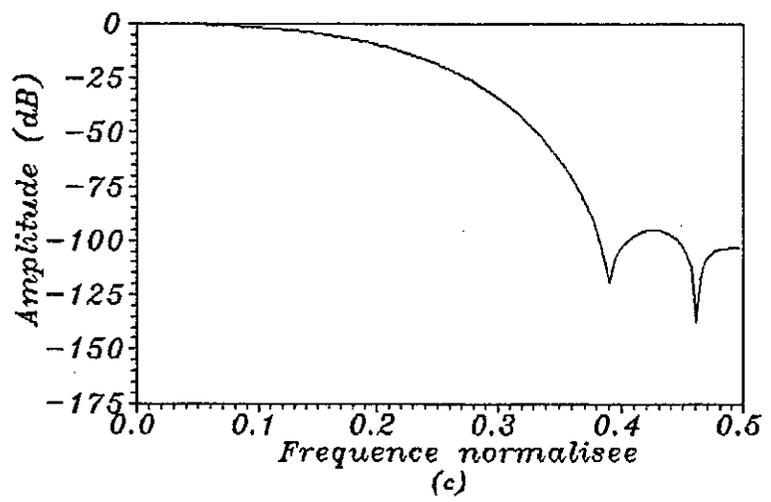
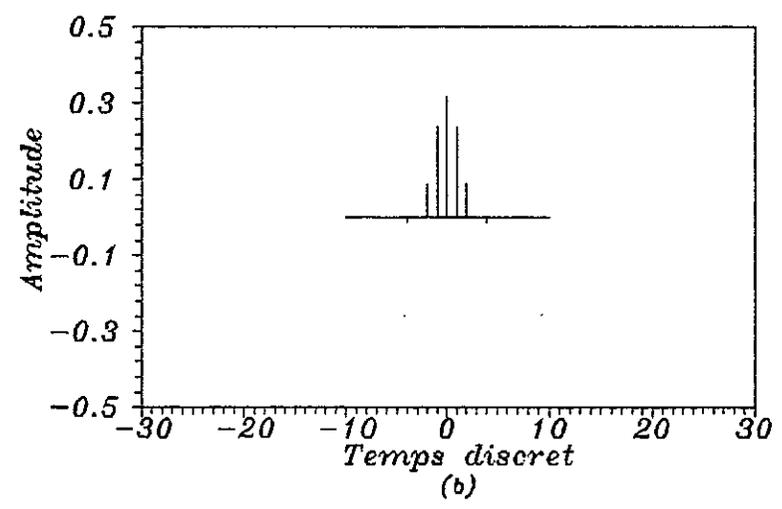
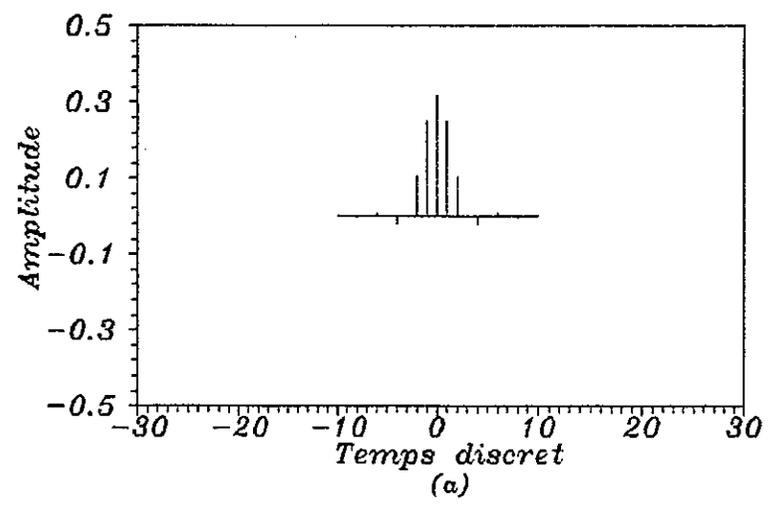


Fig. 3.15.d
 Filtre de mise en forme des impulsions
 limite par
 la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 $N=21$

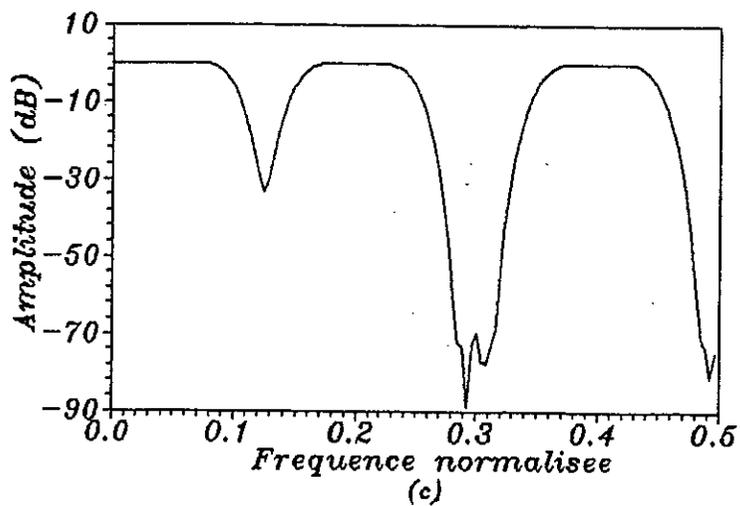
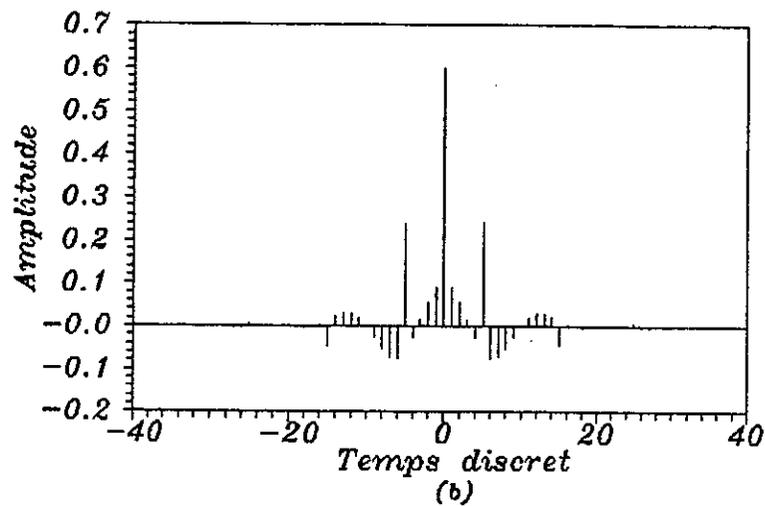
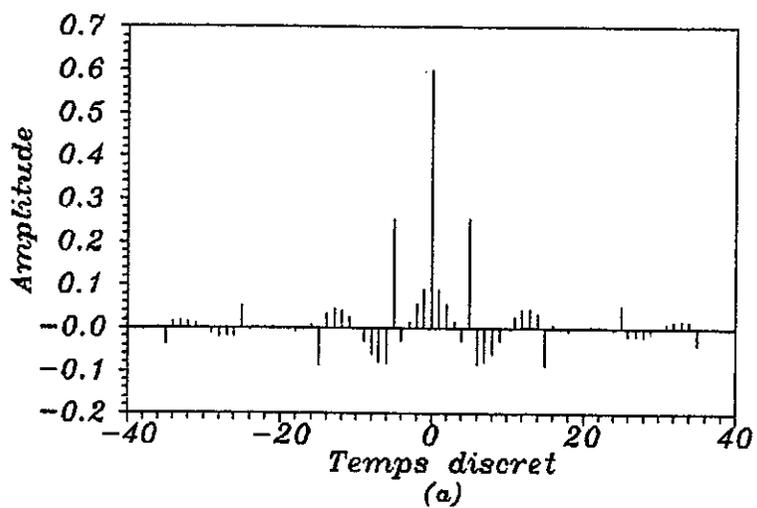


Fig. 3.16.a
 Filtre de mise en forme des impulsions limite par la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 $N=81$
 bande a l'origine: 0 - 0.1
 premiere bande: 0.15 - 0.25
 seconde bande: 0.35 - 0.45

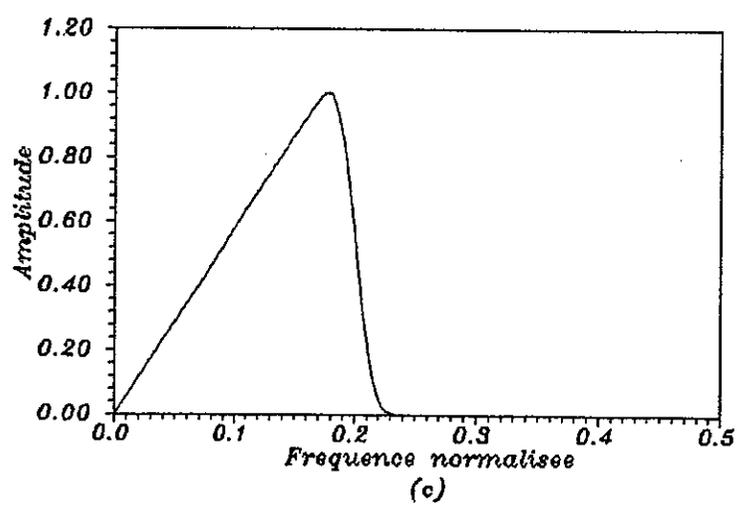
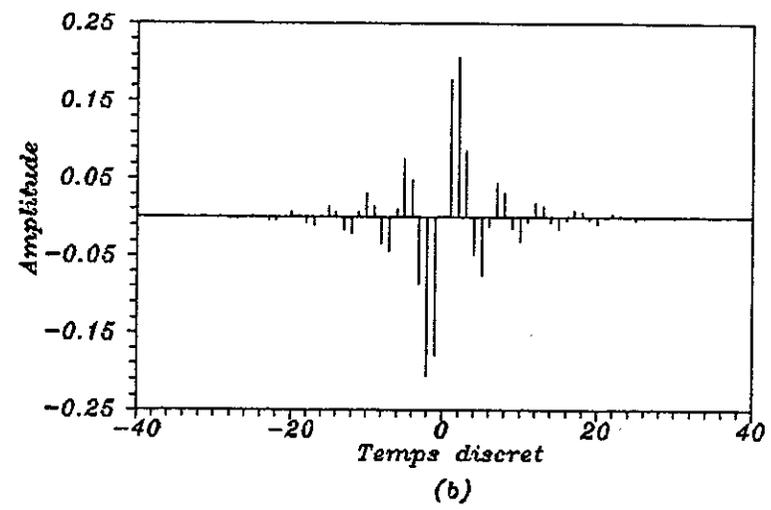
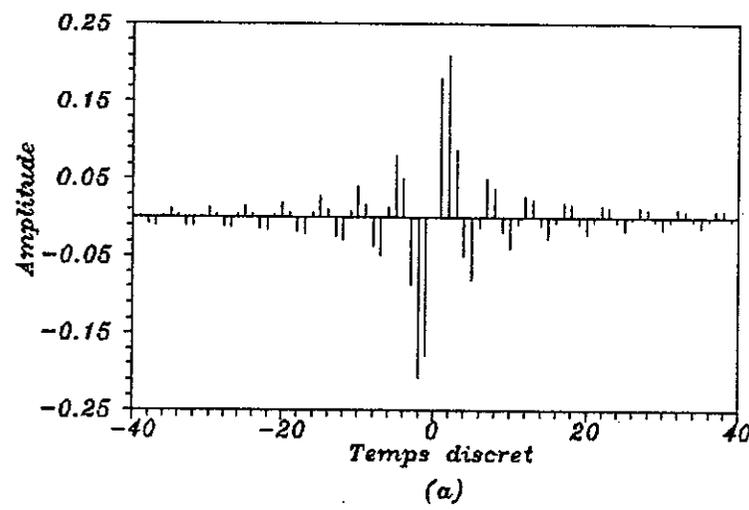


fig. 3.16.b
 Filtre differentiateur limite
 par la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 $N=81$ $Wc1=0.2$

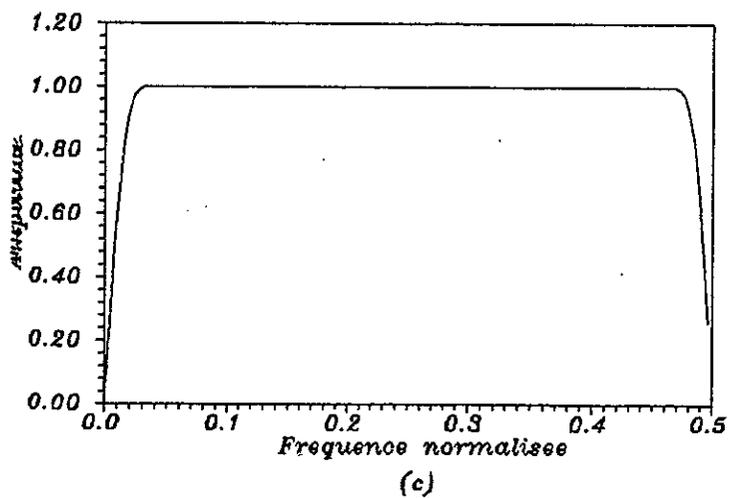
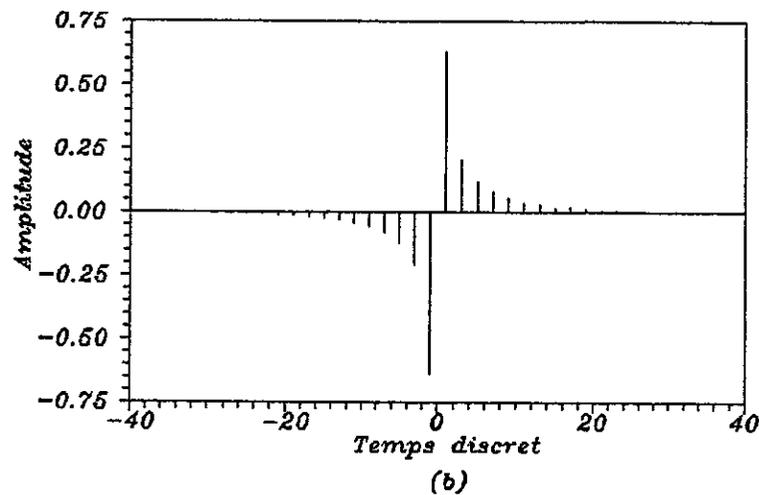
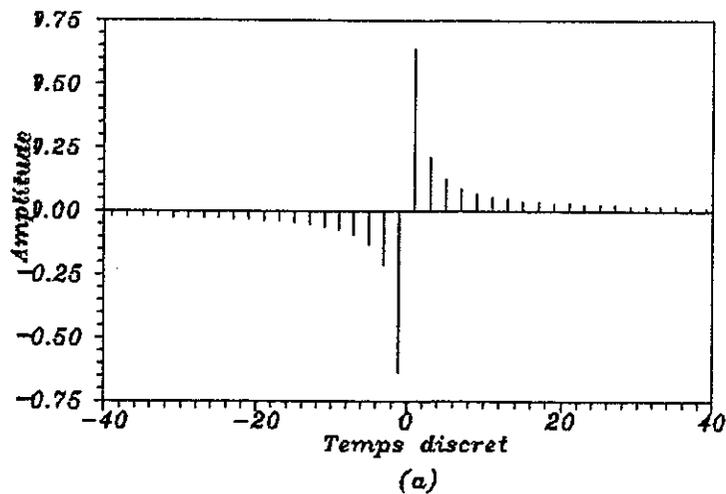


Fig. 3.16.C
 Transformateur de Hilbert limite
 par la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 N=81

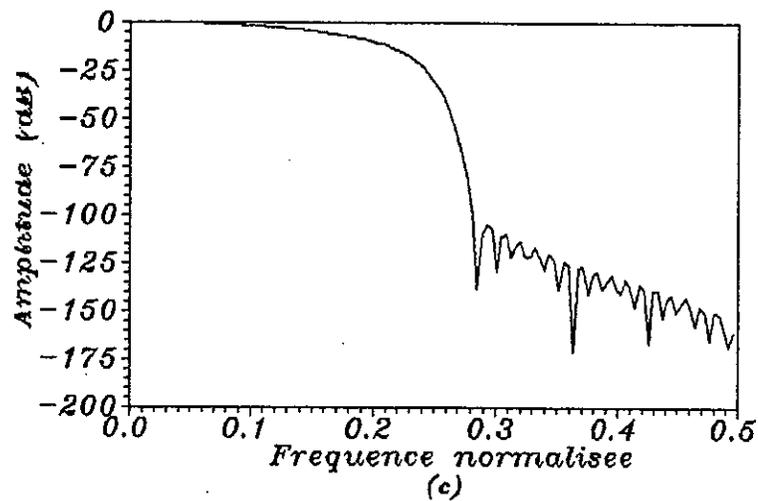
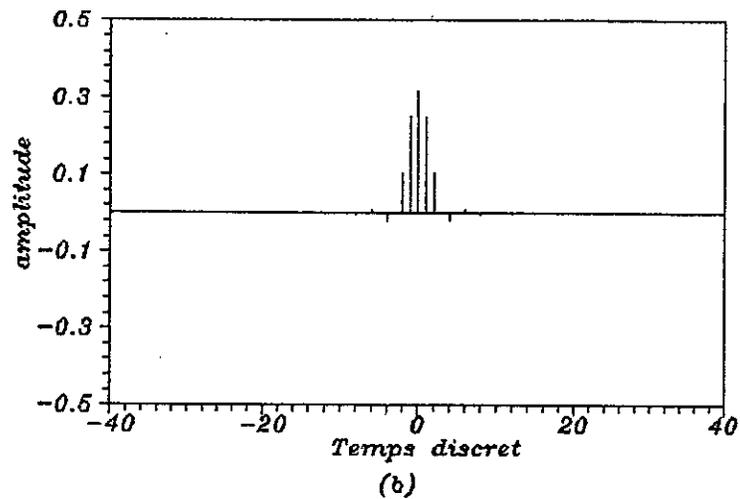
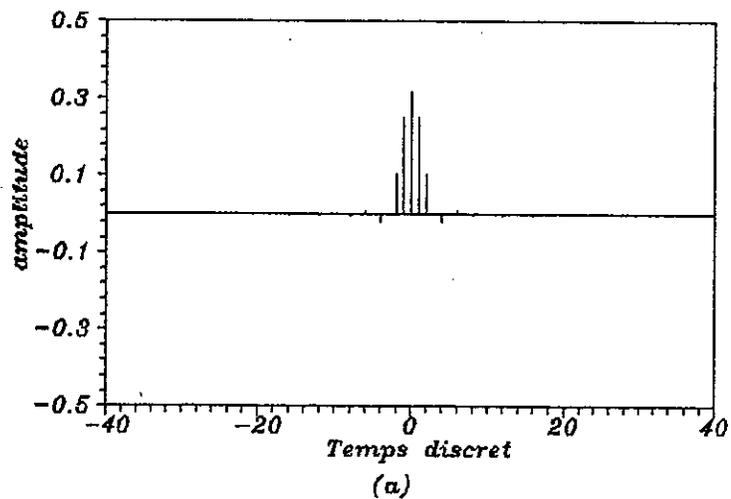


Fig. 3.16.d
 Filtre de mise en forme limite
 par
 la fenetre de Blackman
 a) Reponse impulsionnelle non fenetree
 b) Reponse impulsionnelle fenetree
 c) Spectre d'amplitude
 $N=81$

III.5. CONCLUSION

Dans ce chapitre, nous avons étudié une méthode de synthèse des filtres numériques RIF: la méthode de fenêtrage temporel. Ce concept a été introduit pour étudier, en premier lieu, l'effet de la limitation de la durée de la réponse impulsionnelle d'un filtre sur sa réponse fréquentielle, et en second lieu pour minimiser cet effet.

Pour une fenêtre donnée, et à partir d'un certain ordre (assez faible en général), l'atténuation en bande coupée reste constante, cependant la fréquence des oscillations devient importante lorsque l'ordre N est augmenté.

La bande de transition est inversement proportionnelle à l'ordre N .

En général, lors de la synthèse des filtres RIF, le premier objectif consiste à satisfaire les contraintes imposées sur l'atténuation en bande coupée par un choix adéquat de la fenêtre. La bande de transition est ensuite ajustée à la valeur désirée par le choix de l'ordre N .

Le problème qui se pose avec les fenêtres classiques telles que la fenêtre de HAMMING et celle de BLACKMAN est qu'à chaque fenêtre correspond un niveau de l'atténuation. La variation de cette dernière est donc limitée par le nombre de fenêtres existantes. En introduisant des fenêtres qui sont fonctions, non seulement de l'ordre N du filtre, mais aussi d'un paramètre indépendant, l'atténuation peut varier de façon continue, permettant ainsi une meilleure approximation du filtre désiré.

Parmi ces fenêtres, nous citons la fenêtre de KAISER. Celle-ci est sans doute celle qui s'adapte à une grande variété de contraintes imposées sur sa réponse fréquentielle. Elle permet d'avoir simultanément une atténuation très élevée et une largeur du lobe principal assez faible. Cela n'étant pas possible avec les fenêtres classiques citées plus haut.

CHAPITRE IV

STRUCTURE DE REALISATION
DES FILTRES NON RECURSIFS

IV.1. INTRODUCTION

Une fois la synthèse des filtres numériques effectuée, les opérations de filtrage sont organisées par un choix d'une structure de réalisation. Par réalisation, on entend le processus qui permet la conversion de la fonction de transfert d'un filtre à un ensemble de circuits qui réalisent les fonctions fondamentales :

- Addition.
- Multiplication.
- Mise en mémoire.

Pour mettre en oeuvre ces trois opérations, plusieurs arrangements des circuits peuvent être envisagés, chacun définit une structure de réalisation.

Les filtres R.I.F sont souvent désignés par les filtres non récurrents, car ils ne nécessitent pas de boucle de réaction dans leur réalisation, comme c'est le cas pour les filtres R.L.I qui sont récurrents. Pour les filtres non récurrents, l'efficacité est mesurée par rapport au temps de calcul, c'est à dire que pour un tel filtre, une structure est d'autant plus efficace, qu'elle nécessite moins de temps pour effectuer les opérations de filtrage. Quant aux filtres récurrents, le problème principal est la minimisation de la propagation des erreurs qui sont cumulatives [1].

Dans le présent chapitre, on se contentera de présenter quelques structures envisageables dans le cas non récurrent.

IV.2. STRUCTURE DIRECTE ET STRUCTURE TRANSPOSEE [5], [8],[10] :

La structure directe est connue pour être la plus simple des structures non récurrentes. Elle consiste à faire le produit de convolution du signal d'entrée $x(n)$ par les coefficients $(h(n))$ du filtre correspondants conformément à la relation (I.1) que nous rappelons :

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

où $y(n)$ est le signal de sortie.

Chaque valeur $x(n-i)$ est stockée dans un registre, les contenus de tous les registres sont ensuite pondérés par les coefficients $h(i)$ ensuite sommés pour conduire au signal de sortie $y(n)$.

Une telle structure est illustrée par la figure (IV.1)

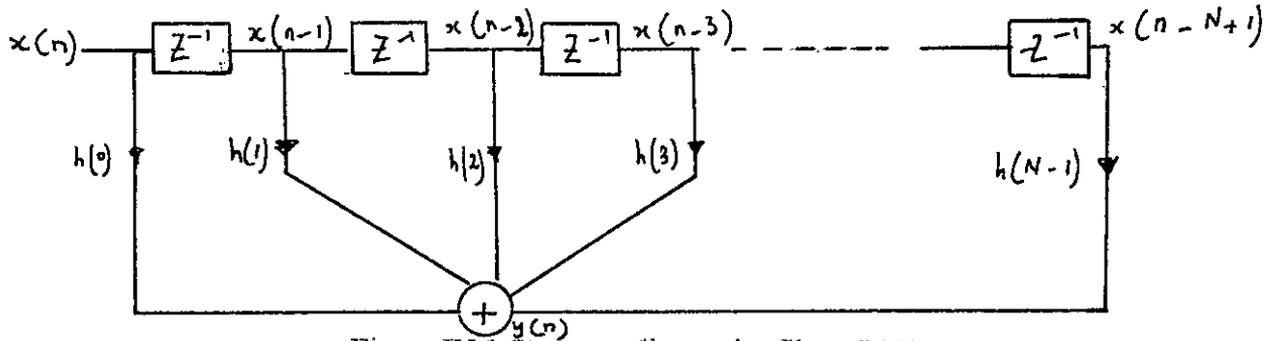


Figure IV.1 Structure directe des filtres R.I.F

On peut envisager une structure dite structure transposée. Elle diffère de la précédente par le fait qu'à chaque instant, la donnée $x(n)$ est multipliée par tous les coefficients du filtre, comme le montre la figure IV.2. Par ailleurs, les mémoires stockent des sommes partielles [5].

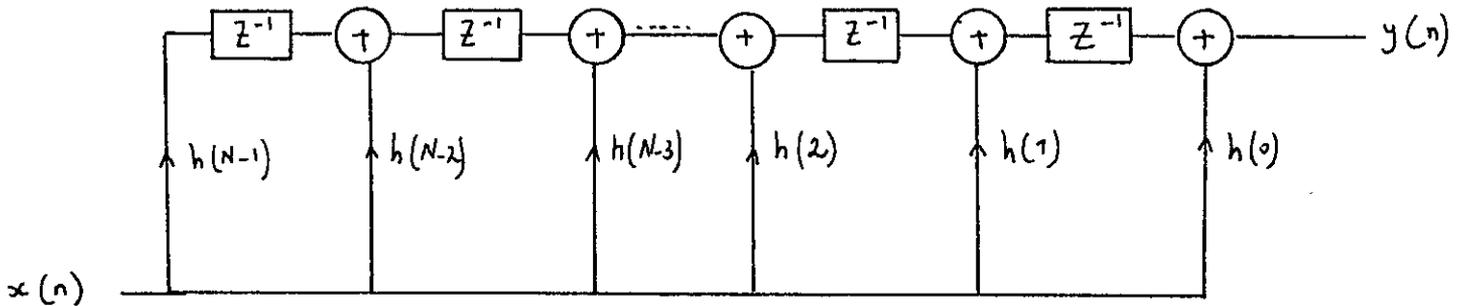


Figure IV.2 Structure transposée du filtre R.I.F

IV.3. STRUCTURE EN CASCADE : [9], [10]

Pour cette structure, la fonction de transfert $H(Z)$ du filtre est factorisée en un nombre de fonctions du second ordre.

$$H(Z) = \prod_{i=1}^{\frac{N-1}{2}} H_i(Z) \tag{IV.1}$$

où $H_i(Z) = a_i + b_i Z^{-1} + c_i Z^{-2}$ (IV.2)

Dès lors, cette réalisation est la mise en cascade de $(N-1) / 2$ blocs similaires comme le montre la figure (IV.3). Chacun de ces blocs est décrit sur la figure (IV.4).

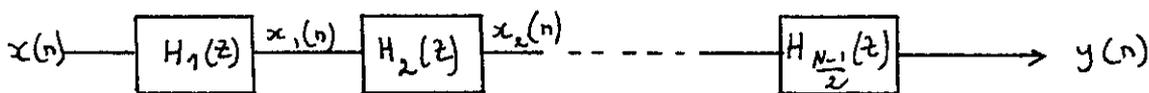


Figure IV.3 Structure cascade des filtres R.I.F

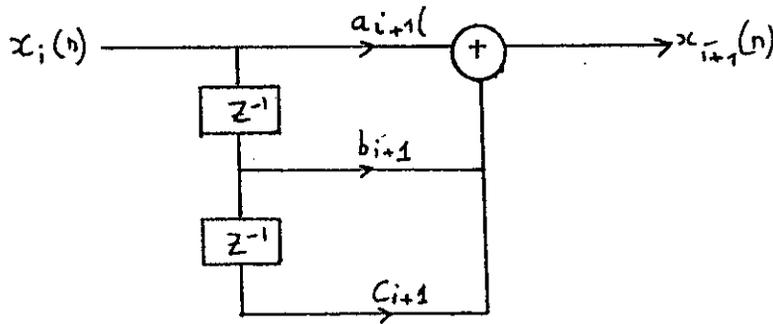


Figure IV.4 Structure interne d'un bloc ($H_{i+1}(Z)$)

IV.4. STRUCTURE EN TREILLIS

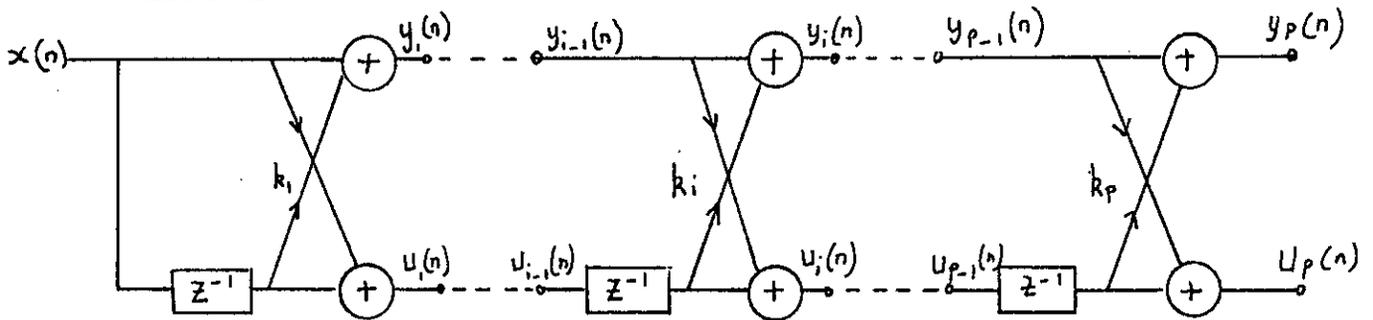


Figure IV.5 Structure en treillis d'un filtre non récursif

Cette structure est illustrée par la figure (IV.5). Elle est constituée d'un nombre de cellules identiques dont les entrées $y_{i-1}(n)$ et $u_{i-1}(n)$ et les sorties $y_i(n)$ et $u_i(n)$ de chacune sont liées par les relations :

$$\begin{cases} y_i(n) = y_{i-1}(n) + k_i u_{i-1}(n-1) \\ u_i(n) = k_i y_{i-1}(n) + u_{i-1}(n-1) \end{cases} \quad i=1,2,\dots,p \tag{IV.3}$$

où k_i est un paramètre appelé coefficient de réflexion caractérisant la i ème cellule et p représente l'ordre du filtre à réaliser avec cette structure.

Notons ici que : $y_0(n) = u_0(n) = x(n)$

où $x(n)$ est l'entrée du filtre

Dans le plan complexe Z , le système d'équation (IV.3) est équivalent au système (IV.4) qui peut être écrit sous forme matricielle (v. 5) :

$$\begin{cases} Y_i(Z) = Y_{i-1}(Z) + k_i Z^{-1} U_{i-1} \\ U_i(Z) = k_i Y_{i-1}(Z) + Z^{-1} U_{i-1} \end{cases} \tag{IV.4}$$

$$\begin{bmatrix} Y_{i+1}(Z) \\ U_{i+1}(Z) \end{bmatrix} = \begin{bmatrix} 1 & k_i Z^{-1} \\ k_i & Z^{-1} \end{bmatrix} \begin{bmatrix} Y_i(Z) \\ U_i(Z) \end{bmatrix} \tag{IV.5}$$

La structure en treillis nécessite la détermination des paramètres k_i $i=1,p$ qui sont en fait les derniers coefficients des polynômes d'ordres inférieurs à P obtenus successivement à partir du polynôme du filtre R.I.F donné. On est amené à exprimer $y_{i-1}(Z)$ et $u_{i-1}(Z)$ en fonction de $y_i(Z)$ et $u_i(Z)$.

Il vient la relation suivante :

$$\begin{bmatrix} Y_{i-1}(Z) \\ U_{i-1}(Z) \end{bmatrix} = \frac{1}{Z^{-1} - Z^{-1}k_i} \begin{bmatrix} Z^{-1} & -k_i Z^{-1} \\ -k_i & 1 \end{bmatrix} \begin{bmatrix} Y_i(Z) \\ U_i(Z) \end{bmatrix} \quad (IV.6)$$

d'où
$$y_{i-1}(Z) = \frac{1}{1 - k_i^2} \{ y_i(Z) - k_i u_i(Z) \} \quad (IV.7)$$

Pour la cellule d'ordre P, on peut écrire [5] :

$$\begin{aligned} y_p(n) &= \sum_{i=0}^p a_i x(n-i) \\ u_p(n) &= \sum_{i=0}^p a_{p-i} x(n-i) \end{aligned} \quad (IV.8)$$

encore :

$$\begin{aligned} y_p(Z) &= \sum_{i=0}^p a_i Z^{-i} \\ u_p(Z) &= \sum_{i=0}^p a_{p-i} Z^{-i} \end{aligned} \quad (IV.9)$$

En tenant compte de (IV.7) et (IV.9), on peut écrire :

$$\begin{bmatrix} a_{i-1}(0) \\ a_{i-1}(1) \\ \vdots \\ a_{i-1}(i-1) \\ 0 \end{bmatrix} = \frac{1}{1 - k_i^2} \begin{bmatrix} a_i(0) \\ a_i(1) \\ \vdots \\ a_i(i-1) \\ a_i(i) \end{bmatrix} - k_i \begin{bmatrix} a_i(i) \\ a_i(i-1) \\ \vdots \\ a_i(1) \\ a_i(0) \end{bmatrix} \quad i=p, p-1, \dots, 2 \quad (IV.10)$$

Les coefficients $a_i(0)$, $i=1, p$ sont normalisés à l'unité. les paramètres k_i sont égaux aux derniers coefficients de chaque polynôme, soit $k_i = a_i(i)$, $i=1, 2, \dots, p$.

Un filtre R.I.F d'ordre P est complètement défini par les paramètres k_i . Ces derniers étant inférieurs à l'unité en valeur absolue, une telle structure s'adapte parfaitement au cas d'une représentation en virgule fixe. Par ailleurs, un treillis d'ordre P inclut tous les treillis d'ordres inférieurs, on peut donc augmenter l'ordre en ajoutant de nouvelles cellules, sans modifier les cellules existantes.

Remarque : La sortie du filtre (qui est le signal filtré) est donnée par $y_p(n)$ qui est en fait la première sortie de la pième cellule. $u_i(n)$ n'interviennent que comme des quantités intermédiaires pour calculer à la fin $y_p(n)$.

IV.5. STRUCTURE DE REALISATION PAR F.F.T

Cette structure est basée sur l'utilisation d'algorithmes rapides d'évaluation de la transformée de Fourier discrète (voir annexe 2), pour mettre en oeuvre l'opération de convolution définie dans la section IV.2.

La transformée de Fourier discrète $Y(k)$ du signal de sortie $y(n)$ est égale au produit des transformées de Fourier discrètes des suites $x(n)$ et $h(n)$, soit :

$$Y(k) = X(k).H(k) \quad (\text{IV.12})$$

Il s'agit donc de calculer les DFT des suites $x(n)$ et $h(n)$, de faire le produit des transformées résultantes, et par une IDFT (Inverse Discrete Fourier Transform) retrouver la suite de sortie $y(n)$. Cette structure est illustrée par la figure (IV.6).

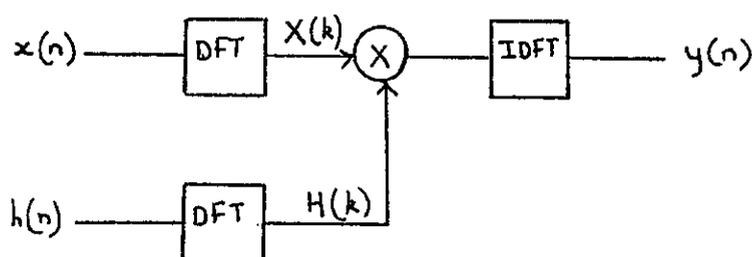


Figure IV.9 structure de réalisation par F.F.T

Les suites $x(n)$ et $h(n)$, ainsi que leurs transformées de Fourier discrètes respectives, sont périodiques. Désignons par N_1 la période de $x(n)$ et de sa D.F.T, et par N_2 celle de $h(n)$ et de sa D.F.T. Le produit de convolution des deux suites $x(n)$ et $h(n)$ engendre une suite $y(n)$ de largeur N_1+N_2 . En d'autre termes, on peut dire que les coefficients $y(n)$ sont non nuls sur un intervalle de longueur N_1+N_2 .

Pour pouvoir réaliser le produit (IV.11) , il faudrait égaliser les périodes de $X(k)$ et $H(k)$, qui sont celles de $x(n)$ et $h(n)$, d'où il vient :

$$h'(n) = \begin{cases} h(n) & 0 \leq n \leq N_2 - 1 \\ 0 & N_2 \leq n \leq L \end{cases}$$

$$x'(n) = \begin{cases} x(n) & 0 \leq n \leq N_1 - 1 \\ 0 & N_1 \leq n \leq L \end{cases}$$

Cette opération consiste à compléter les suites $x(n)$ et $h(n)$ par des coefficients nuls jusqu'à atteindre une longueur L qui définira la période de $Y(k)$ ainsi que celle de $y(n)$.

Sachant que $y(n)$ présente des coefficients non nuls sur l'intervalle $[0, N_1+N_2-1]$, il faut veiller à ce que la longueur L soit supérieur ou égale à la valeur N_1+N_2-1 pour éviter tout recouvrement dans la suite $y(n)$. La solution la plus évidente est l'égalité des deux quantités, soit $L = N_1+N_2-1$.

Cette structure est surtout utilisée lorsque l'ordre du filtre est élevé, dans ce cas l'évolution de la convolution temporelle entraînerait des temps de calcul plutôt importants. Pour des ordres plus faibles, il n'y a pas d'inconvénients à utiliser la convolution directe.

IV.6. CONCLUSION

Dans ce chapitre nous avons présenté quelques structures de réalisation des filtres non récursifs.

Le choix d'une structure ou d'une autre dépend de certains critères dont nous citons, le temps de calcul, le type et le nombre d'opérations à effectuer, la place mémoire et la facilité de mise en oeuvre ou de conception.

CHAPITRE V

OPTIMISATION DES FILTRES R.I.F PAR LA METHODE DE RECHERCHE LOCALE

V.1. INTRODUCTION

En général, les méthodes de synthèse des filtres numériques ne tiennent pas en compte le fait, que dans de nombreuses applications, les coefficients sont codés sur un nombre de bits limité. Les résultats de synthèse ainsi obtenus sont codés sur un nombre de bits infini (ou du moins un nombre très grand).

La quantification des coefficients conduit à un filtre dont la réponse fréquentielle peut s'écarter fortement de celle du filtre d'origine.

A priori, la longueur des registres dans lesquels seront accumulés les coefficients du filtre est inconnue, cependant elle doit être la plus petite possible afin de réduire le coût de la réalisation et d'accroître la vitesse de calcul, tout en respectant les spécifications du filtre. Soient:

$M_1(\omega)$ le spectre d'amplitude du filtre idéal,

$M_Q(\omega)$ le spectre d'amplitude du filtre à coefficients quantifiés,

$M(\omega)$ le spectre d'amplitude du filtre calculé en précision infinie,

et δ l'écart maximum admissible entre le filtre idéal et le filtre évalué en précision infinie.

L'erreur de quantification :

$$|\Delta M| = |M(\omega) - M_Q(\omega)| \quad (V.1)$$

doit être telle que le spectre $M_Q(\omega)$ s'inscrive dans le gabarit

$$M_1(\omega) - \delta \leq M_Q(\omega) \leq M_1(\omega) + \delta \quad (V.2)$$

Des techniques d'optimisation dans l'espace discret des coefficients ont été établies pour minimiser l'effet de la quantification, nous citons la méthode de la recherche locale, qui, basée sur l'analyse combinatoire, consiste à examiner le voisinage immédiat des coefficients quantifiés, et à modifier ces derniers en conséquence, afin d'obtenir la meilleure approximation du filtre désiré [12],[13].

Ces techniques ne permettent de déterminer la longueur optimale des mots, qu'après calcul des coefficients pour différentes longueurs et évaluation de l'erreur pour chaque cas traité, entraînant ainsi des temps de calcul plus ou moins importants.

Des approches statistiques proposées par *Avenhaus* [19], et par la suite modifiées par *Crochiere* [20], viennent combler cette insuffisance, elles permettent l'estimation stochastique de la longueur appropriée au cas traité.

V.2. QUANTIFICATION

Pour pouvoir stocker les paramètres d'un filtre évalués avec une grande précision (celle de l'ordinateur) dans des registres de longueur finie, on a recours à la quantification. Cette dernière est une méthode approximative, basée sur le principe de discrétisation des amplitudes en un nombre fini de niveaux. La distance entre deux niveaux successifs définit le pas de la quantification.

Pour des raisons technologiques, nous sommes amenés à décomposer les nombres de la base 10 et à les représenter dans une autre base, le plus souvent la base 2.

La quantification dépend du type de représentation des nombres décimaux dans une autre base.

V.2.1. Types de représentation des nombres binaires :

Il existe diverses façons d'établir la correspondance entre l'ensemble des amplitudes quantifiées et l'ensemble des nombres qui doivent les représenter. Les plus courantes sont [1],[3],[9] :

- Représentation en virgule fixe :

Soit un nombre N, s'écrivant dans une base r :

$$N = (b_n \dots b_0 \cdot b_{-1} \dots b_{-m}) \tag{V.3}$$

Sa représentation décimale est :

$$N = \sum_{i=-m}^n b_i r^i \quad \text{avec } 0 \leq b_i \leq r-1 \tag{V.4}$$

Pour r = 2, nous avons :

$$N = \sum_{i=-m}^n b_i 2^i \quad b_i = 0 \text{ ou } 1$$

Dans ce qui suivra, les nombres à traiter sont fractionnaires (inférieurs à 1 en valeur absolue). Par conséquent, la virgule binaire se situe entre le premier et le second bits les plus significatifs. Le premier bit étant réservé au signe.

On distingue trois formes de représentation des nombres signés :

1- Signe et valeur absolue :

Dans cette représentation, un nombre N s'écrit [1] :

$$N_{sva} = \begin{cases} 0.b_{-1}b_{-2} \dots b_{-m} & N \geq 0 \\ 1.b_{-1}b_{-2} \dots b_{-m} & N \leq 0 \end{cases} \tag{V.5}$$

2 - Complément à 1 [1] :

$$N_{cl} = \begin{cases} N & N \geq 0 \\ 2 - 2^{-B} - |N| & N \leq 0 \end{cases} \tag{V.6}$$

B est le nombre de bits à droite de la virgule binaire.

3 - Complément à 2 [1] :

$$N_{c2} = \begin{cases} N & N \geq 0 \\ 2^{-|N|} & N < 0 \end{cases} \quad (V.7)$$

Chacune de ces trois représentations possède des avantages et des inconvénients [1]. Le choix de l'une ou l'autre est dicté par le type d'opérations à effectuer.

Dans une représentation en virgule fixe, on rencontre souvent le problème de dépassement de capacité. Pour pallier à cet inconvénient, il faut veiller à utiliser des facteurs d'échelle afin de respecter la gamme dynamique choisie [3] tout en gardant une précision suffisante.

- Représentation en virgule flottante :

Un nombre F est représenté par :

$$F = M b^e \quad (V.8)$$

e est un nombre entier, (exposant)

M est un nombre fractionnaire normalisé : $\frac{1}{b} < M < 1$. (mantisse)

La représentation en virgule flottante permet une extension de la gamme dynamique des registres, cependant, elle entraîne une complication des opérations arithmétiques et des circuits, du fait du traitement séparé de la mantisse et de l'exposant.

Une fois le type de représentation choisi, on peut adopter le principe de la troncature ou de l'arrondi pour réaliser la quantification.

V.2.2. Quantification par troncature :

Soit un nombre représenté sur un nombre de bits donné b (signe non compris). On désire le stocker dans un registre de longueur b_1 avec $b_1 \leq b$.

La troncature consiste à ignorer les $(b-b_1)$ bits les moins significatifs. Le pas de quantification est égal à 2^{-b_1} .

L'erreur résultant de cette quantification

$$E = x - Q(x) \quad (V.9)$$

où $Q(x)$ est la valeur quantifiée de x
dépend du type de représentation utilisé.

En virgule fixe et pour la forme signe et valeur absolue, on a [1] :

$$\begin{aligned} 0 \leq E \leq 2^{-b_1} - 2^{-b} & \quad \text{pour } x \geq 0 \\ -(2^{-b_1} - 2^{-b}) \leq E \leq 0 & \quad \text{pour } x \leq 0 \end{aligned} \quad (V.10)$$

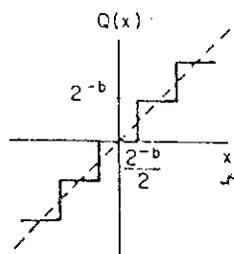
V.2.3. Quantification par l'arrondi :

Avec les mêmes hypothèses que précédemment, l'arrondi consiste à approximer un coefficient par le niveau le plus proche. Pratiquement, ceci revient à ajouter 1 à la position b_1+1 , ensuite réaliser une troncature à b_1 bits. (avec la convention que la position zéro correspond au bit le plus significatif).

L'erreur d'arrondi est donnée par [1] :

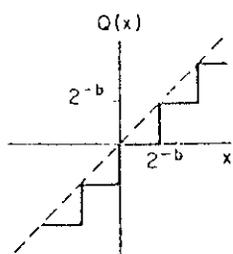
$$-\frac{2^{-b_1}}{2} \leq E \leq +\frac{2^{-b_1}}{2} \tag{V.11}$$

En conclusion, la quantification revient à faire passer le signal dans un organe dont les caractéristiques sont les suivantes :



Rounding

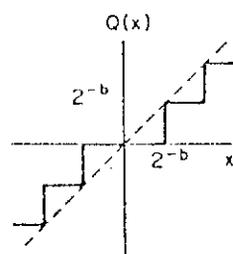
$$-\frac{1}{2} \cdot 2^{-b} < Q(x) - x \leq \frac{1}{2} \cdot 2^{-b}$$



Truncation

(two's complement)

$$-2^{-b} < Q(x) - x \leq 0$$



Truncation

(one's complement and sign and magnitude)

$$-2^{-b} < Q(x) - x \leq 0; x > 0$$

$$0 \leq Q(x) - x < 2^{-b}; x < 0$$

V.3. EFFET DE LA QUANTIFICATION SUR LA REPONSE FREQUENTIELLE DU FILTRE

Dans ce qui suivra, on adoptera la forme signe et valeur absolue en virgule fixe. Pour un filtre dont les coefficients de la réponse impulsionnelle calculés en précision infinie sont $h(0)...h(N-1)$, la quantification est réalisée par l'opération suivante :

$$h'(i) = \pm \left[|h(i)| \cdot 2^{b-1} \right] \cdot 2^{-(b-1)} \quad (V.12)$$

où b est la longueur des mots signe compris et les crochets dénotent la valeur quantifiée de la quantité entre eux.

Les nombres positifs et négatifs sont traités de façon identique, sauf qu'il faut ajuster le bit de signe pour les seconds.

L'opération définie par la relation (V.12) consiste d'abord, à décaler les coefficients à gauche (par rapport à la virgule) d'un nombre de bits égal à la longueur effective des registres (par la longueur effective on entend le nombre de bits significatifs, le bit de signe exclu), ensuite on ignore les bits les moins significatifs après qu'on ait ajusté le bit de poids le plus faible (premier bit à gauche de la virgule) dans le cas de l'arrondi, ceci est réalisé en rajoutant 2^{-1} au module du coefficient à traiter, soit une unité au premier bit à droite de la virgule.

La partie entière ainsi obtenue est décalée à droite du même nombre de bits, et le nombre quantifié $h'(i)$ est alors obtenu.

Pour voir l'effet de la quantification sur la réponse fréquentielle du filtre, on prend quelques exemples de filtres dont les spécifications sont les suivantes :

- Filtre passe-bas d'ordre 23, de fréquence de coupure normalisée 0.1, limité par la fenêtre triangulaire.
- Filtre passe-bas d'ordre 51, de fréquence de coupure normalisée 0.1, limité par la fenêtre triangulaire.
- Filtre passe-bas d'ordre 81, de fréquence de coupure normalisée 0.1, limité par la fenêtre de Kaiser pour une valeur de α égale à 7.5 .

Après évaluation des coefficients en précision infinie, on procède à une quantification par l'arrondi. Les résultats sont illustrés par les graphes des figures (V.2a) à (V.2d), (V.3a) à (V.3d) et (V.4a) à (V.4d). Les longueurs des mots utilisés sont 4,6,8 et 12 bits.

Pour le troisième exemple, nous avons opté pour une représentation des amplitudes en unités normales pour mieux voir l'écart entre la réponse idéale et la réponse après quantification. Le graphe de la figure (V.5) montre l'évolution de cet écart en fonction de la fréquence.

Nous avons noté que pour les différentes longueurs des registres considérées, l'écart maximal se produit au voisinage de la fréquence de coupure du filtre (c'est à dire au niveau de la discontinuité) et ne diminue pas automatiquement lorsque le nombre de bits est augmenté. De part et d'autre de la discontinuité, l'écart entre les réponses fréquentielles diminue au fur et à mesure que le nombre de bits augmente. A partir de 8 bits l'écart est très faible, le filtre à coefficients quantifiés est pratiquement confondu avec le filtre synthétisé avec précision infinie.

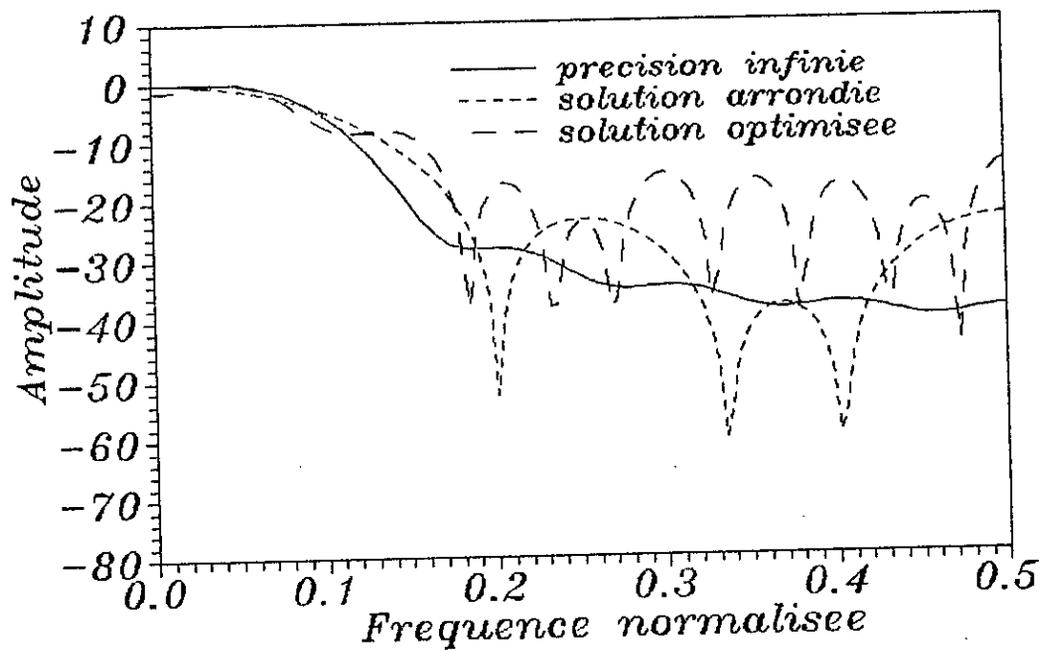


Fig 5.2.a
 spectre d'amplitude d'un filtre passe bas
 d'ordre 23 ,de frequence de coupure normalisee
 0.1 .La longueur des mots quantifies : 4

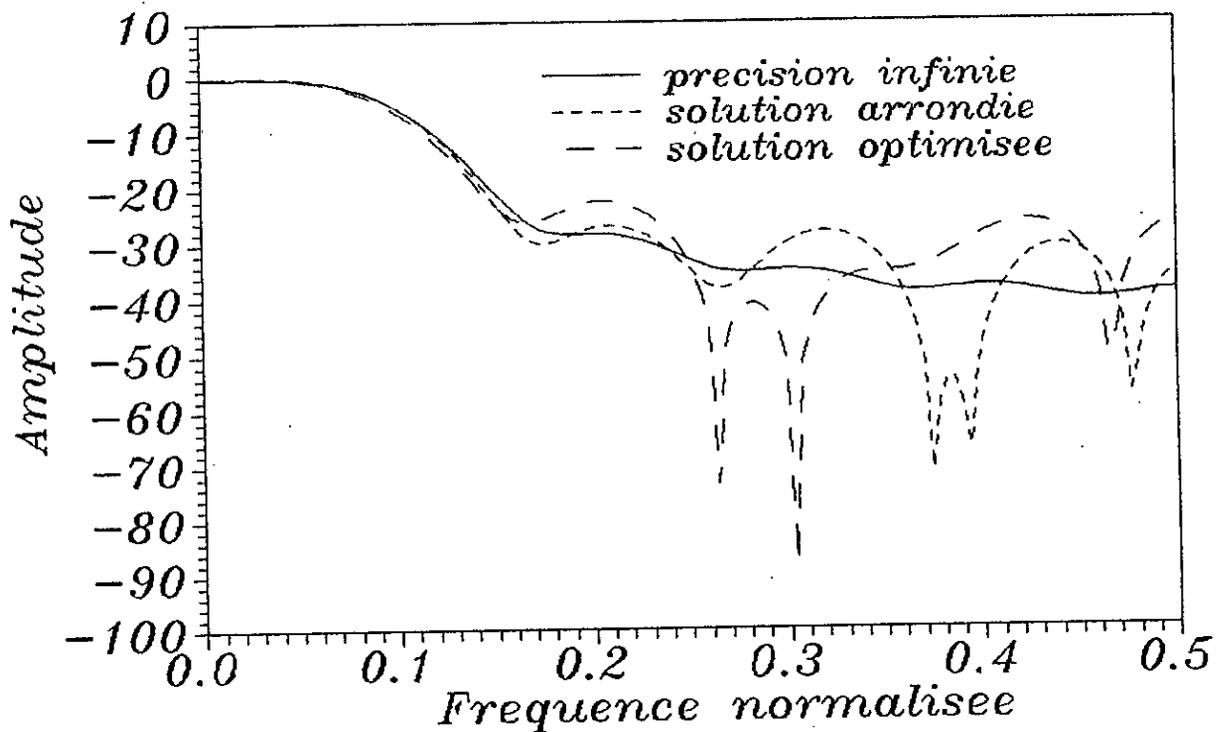


Fig 5.2.b
 spectre d'amplitude d'un filtre passe bas
 d'ordre 23 ,de frequence de coupure normalisee
 0.1 .La longueur des mots quantifies : 6

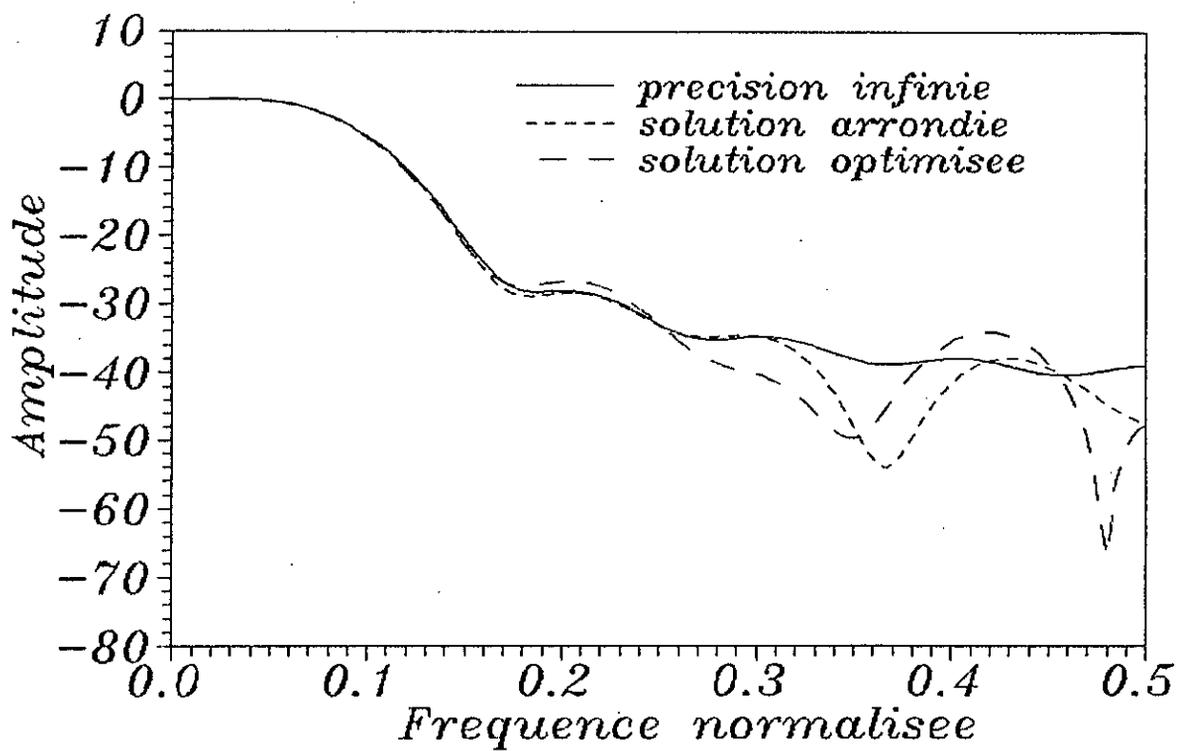


Fig 5.2.c
 spectre d'amplitude d'un filtre passe bas
 d'ordre 23 ,de frequence de coupure normalisee
 0.1 .La longueur des mots quantifies : 8

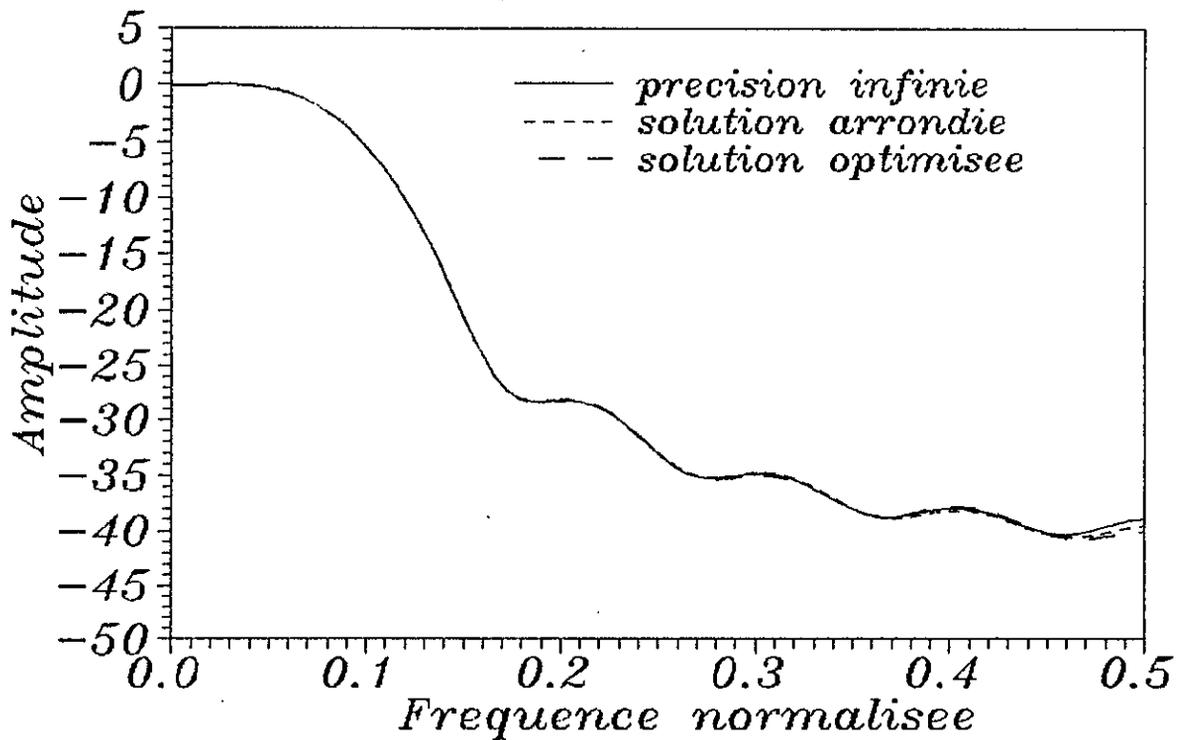


Fig 5.2.d
 spectre d'amplitude d'un filtre passe bas
 d'ordre 23 ,de frequence de coupure normalisee
 0.1 .La longueur des mots quantifies : 12

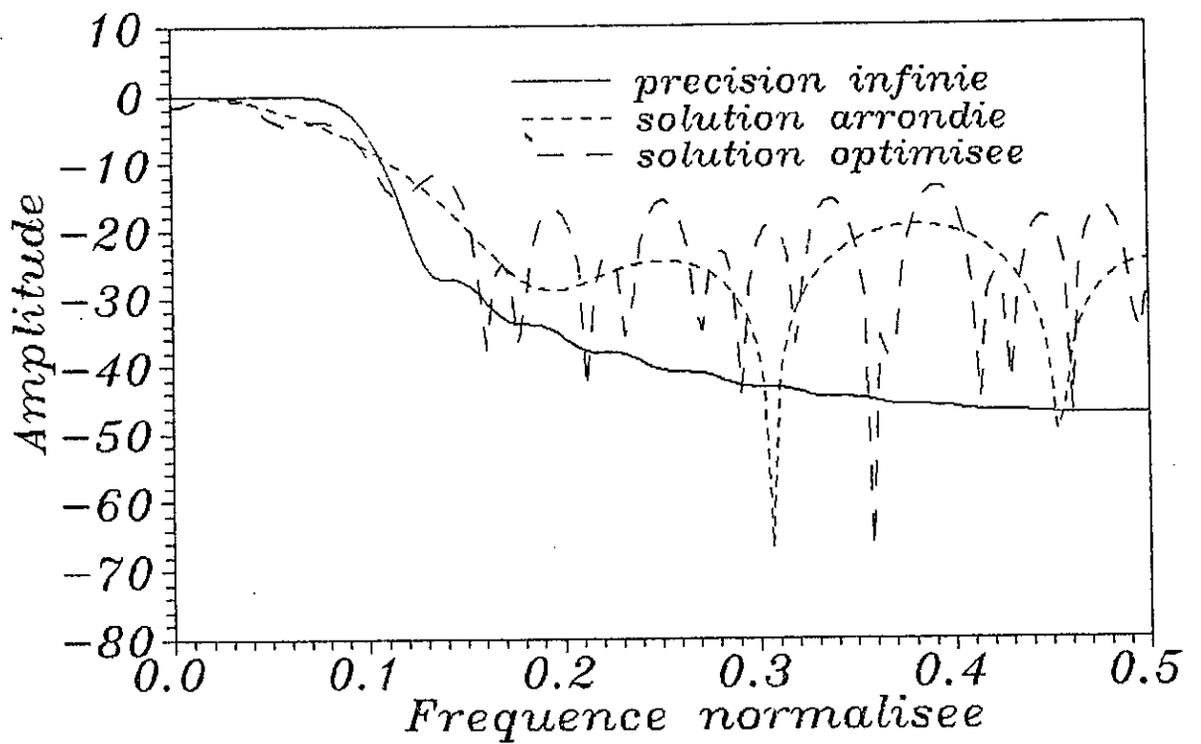


Fig 5.3.a
 spectre d'amplitude d'un filtre passe bas
 d'ordre 51 ,de frequence de coupure normalisee
 0.1 .La longueur des mots quantifies : 4

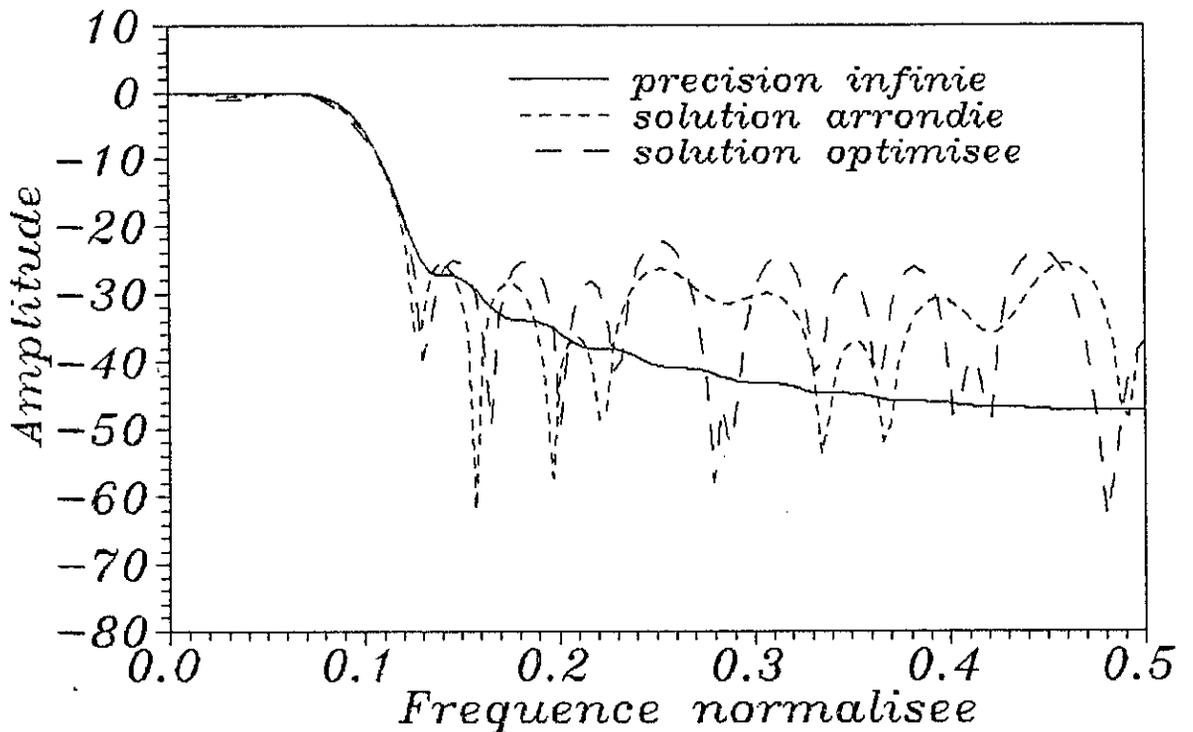


Fig. 5.3. b
 spectre d'amplitude d'un filtre passe bas
 d'ordre 51 ,de frequence de coupure normalisee
 0.1 .La longueur des mots quantifies : 6

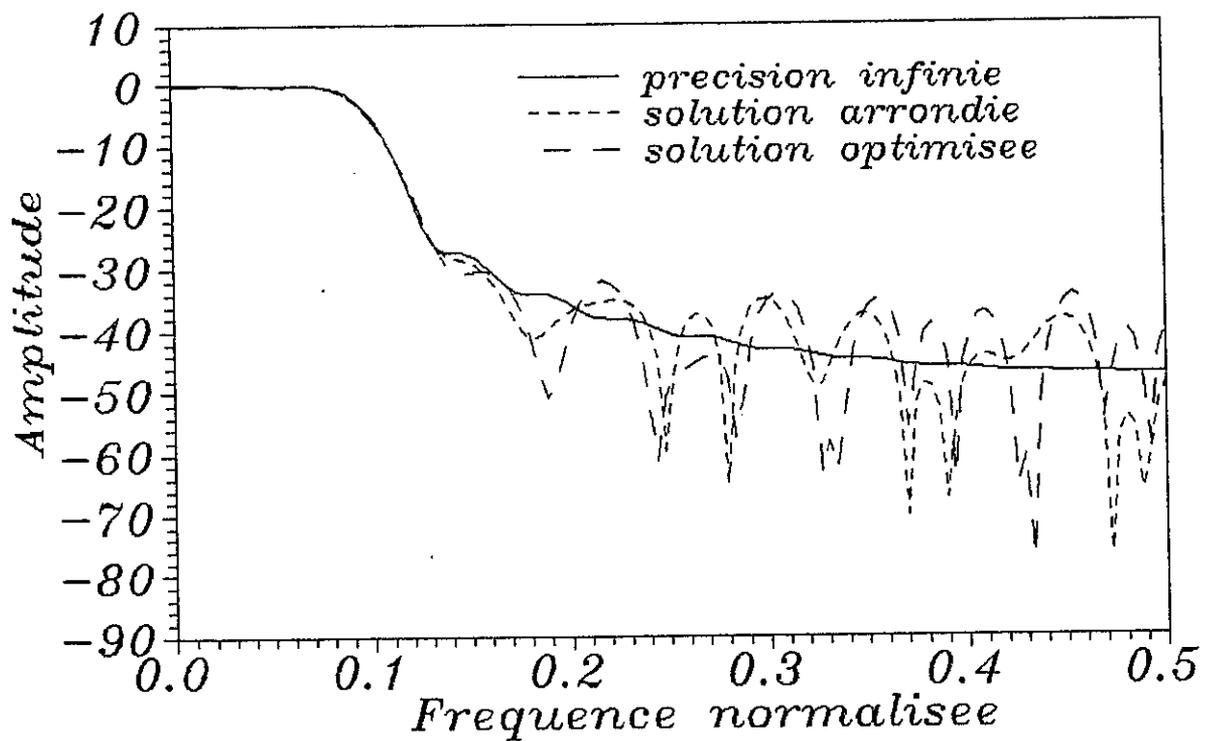


Fig 5.3.c
spectre d'amplitude d'un filtre passe bas
d'ordre 51 ,de frequence de coupure normalisee
0.1 .La longueur des mots quantifies : 8

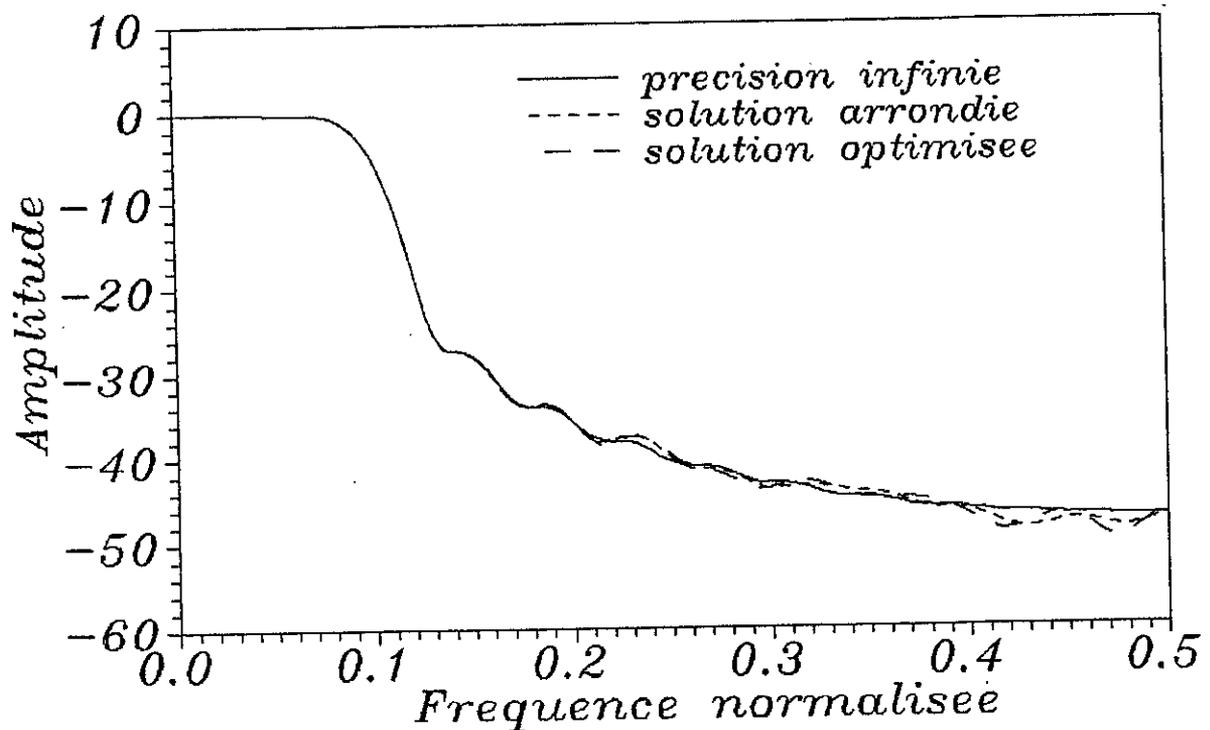


Fig 5.3.d
spectre d'amplitude d'un filtre passe bas
d'ordre 51 ,de frequence de coupure normalisee
0.1 .La longueur des mots quantifies : 12

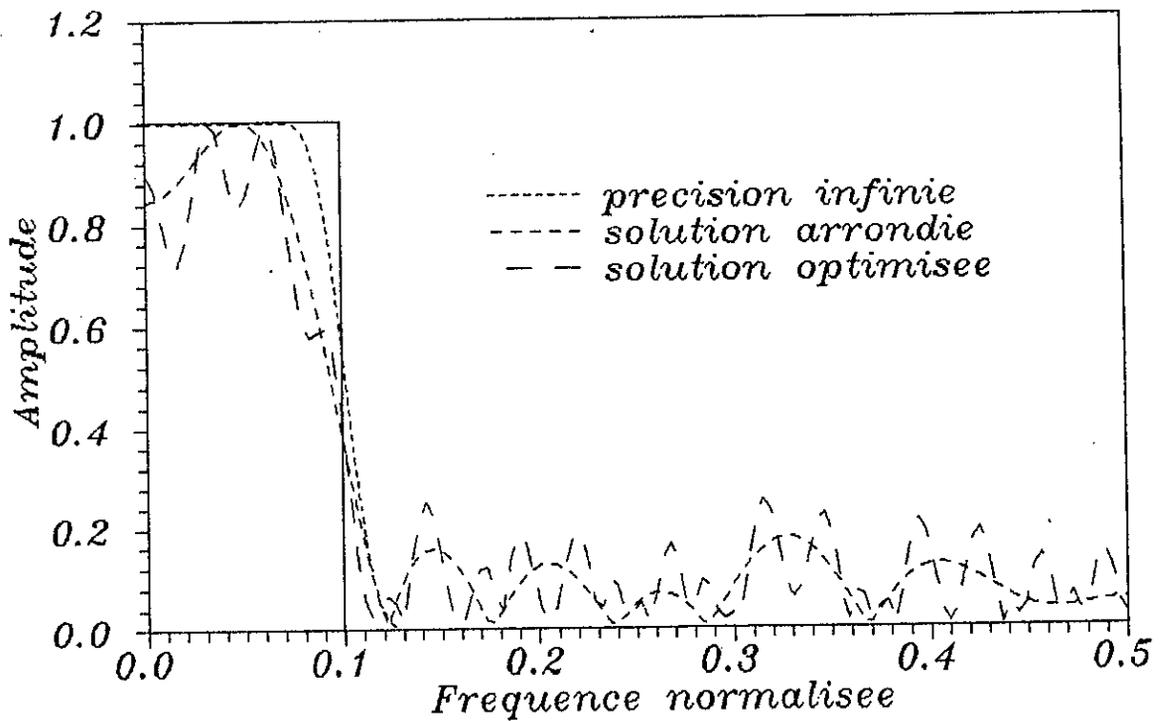


Fig 5.4. a filtre passe bas limite par la fenetre de KAISER, d'ordre 81, de frequence de coupure normalisee 0.1 , longueur des mots : 4 bits

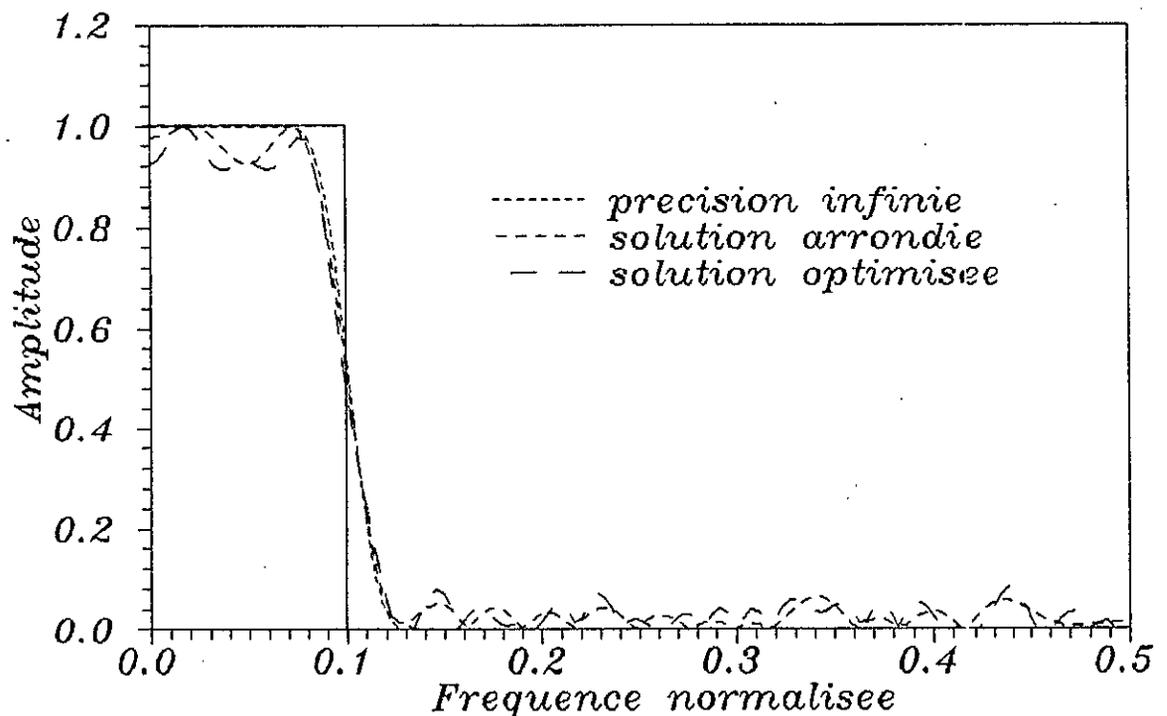


Fig 5.4. b filtre passe bas limite par la fenetre de KAISER, d'ordre 81, de frequence de coupure normalisee 0.1 , longueur des mots : 6 bits

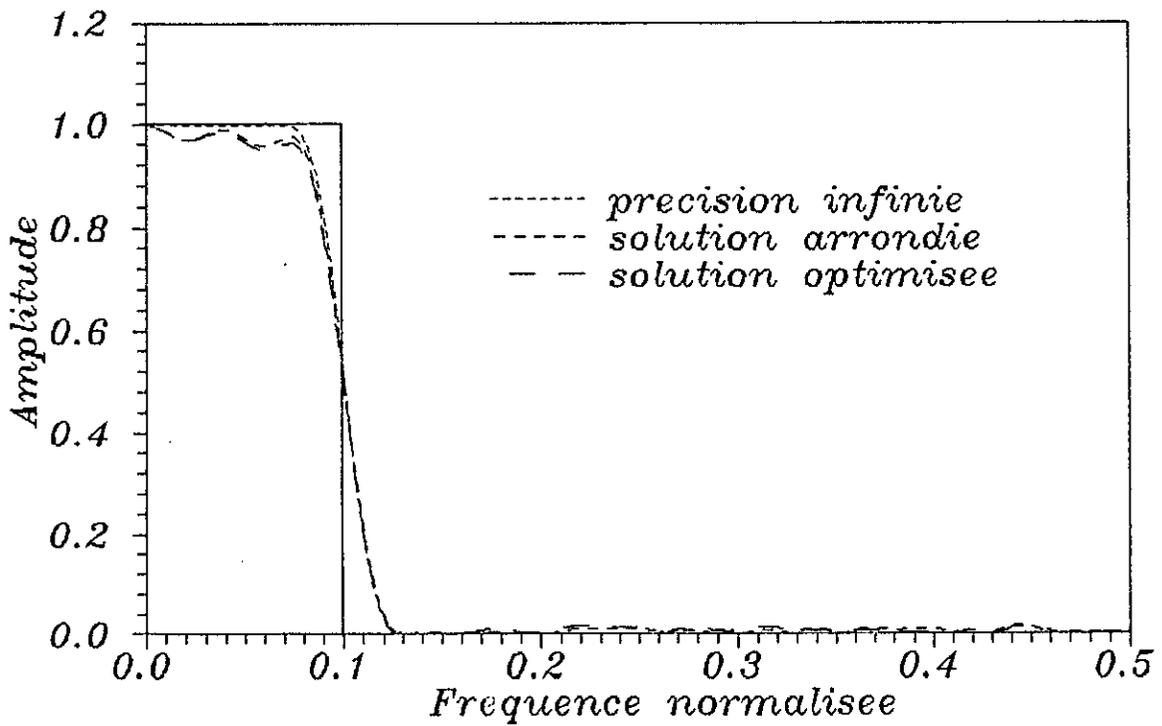


Fig 5.4.c filtre passe bas limite par la fenetre de KAISER, d'ordre 81, de frequence de coupure normalisee 0.1, longueur des mots : 8 bits

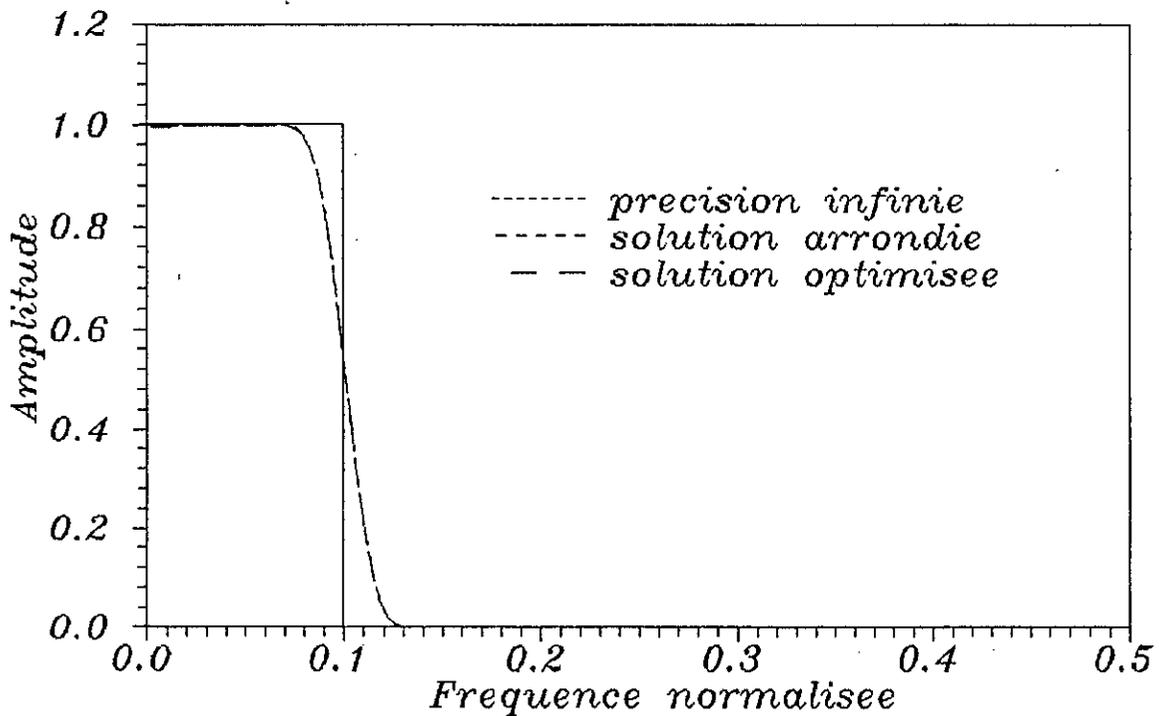


Fig 5.4.d filtre passe bas limite par la fenetre de KAISER, d'ordre 81, de frequence de coupure normalisee 0.1, longueur des mots : 12 bits

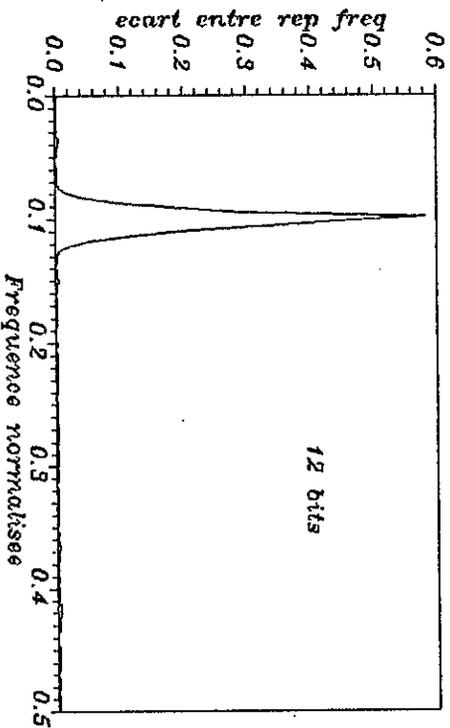
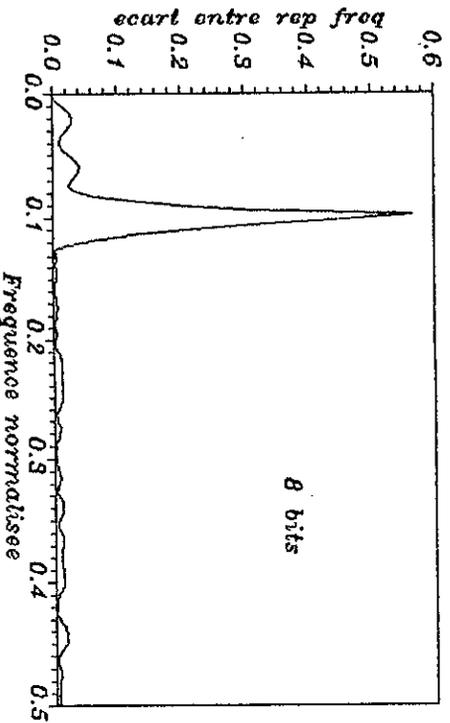
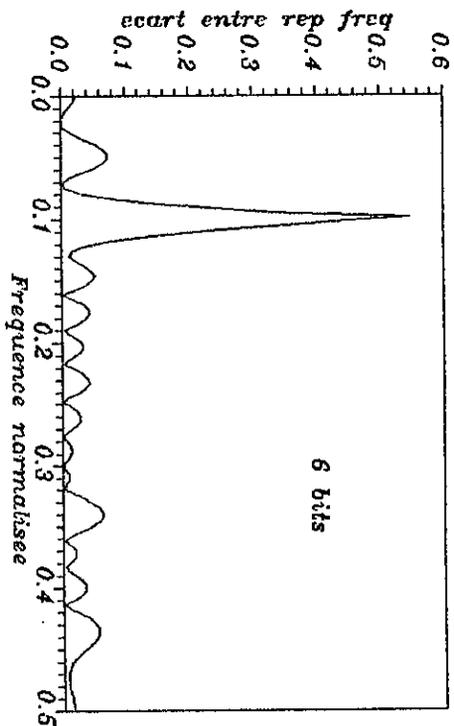
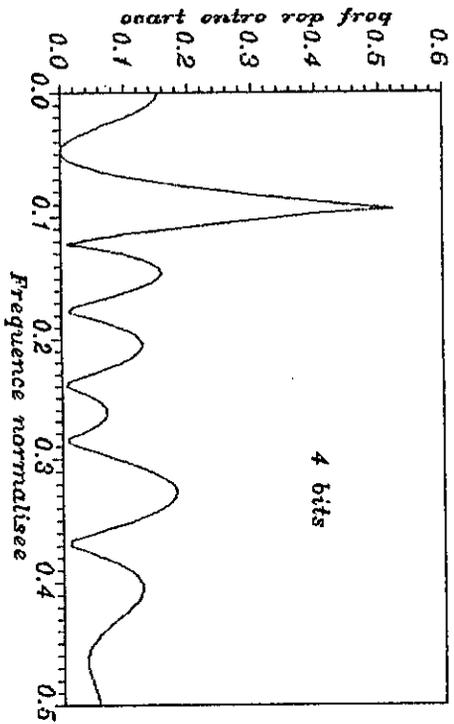


Fig 5.5 Evolution de l'ecart entre la reponse ideale et la reponse apres quantification des coefficients par l'arrondi pour un filtre passe bas de frequence de coupure normalisee 0.1

V.4. METHODE DE LA RECHERCHE LOCALE

Les algorithmes de la recherche locale sont utilisés pour optimiser les filtres numériques à coefficients quantifiés. L'objectif de cette méthode est d'atteindre une solution pour laquelle l'erreur définie par la relation suivante :

$$E = \max |M_1(\omega) - M_0(\omega)| \quad (V.13)$$

où $M_0(\omega)$ est le spectre d'amplitude correspondant à la solution recherchée, est minimale.

La méthode de la recherche locale consiste à explorer un voisinage de la solution arrondie (le terme solution désigne l'ensemble des coefficients de la réponse impulsionnelle du filtre); ceci tient du fait que la solution recherchée ou la solution sub-optimale ne peut s'écarter complètement de la solution arrondie, et d'en extraire la combinaison qui approcherait au mieux le filtre idéal. Par ailleurs, pour aboutir à des résultats satisfaisants, il suffit de perturber quelques coefficients seulement.

On définit un voisinage V_M d'ordre M d'un coefficient h_i comme étant l'ensemble des valeurs h_i' qui s'écartent au plus de $M \cdot 2^{-(b-1)}$ unités par rapport à h_i . D'où il vient :

$$V_M = \left\{ h_i' \text{ tel que } h_i' = h_i + m \cdot 2^{-(b-1)} \right. \\ \left. \begin{array}{l} -M \leq m \leq +M \\ 0 \leq i \leq L \end{array} \right\} \quad (V.14)$$

où b désigne le nombre de bits considérés et L désigne le nombre maximum de coefficients sur lesquels on peut exercer simultanément une perturbation.

Les étapes de la recherche locale, décrites par l'organigramme de la figure (V.6), peuvent être résumées comme suit :

Partant de la solution en précision infinie, on procède à la quantification des coefficients par l'arrondi ou par troncature (selon la relation V.12) et on évalue l'erreur correspondante (selon la relation V.13). On initialise la solution sub-optimale à la solution arrondie, suite à quoi, on génère toutes les combinaisons, en perturbant L coefficients d'une quantité $m \cdot 2^{-(b-1)}$, m variant de $-M$ à $+M$. A chaque étape de calcul (pour chaque valeur de m) on évalue l'erreur correspondante et on la compare à l'erreur optimale obtenue précédemment. La combinaison retenue sera celle qui présentera l'erreur la plus faible. Pour la génération des combinaisons, le point de départ est toujours la solution arrondie.

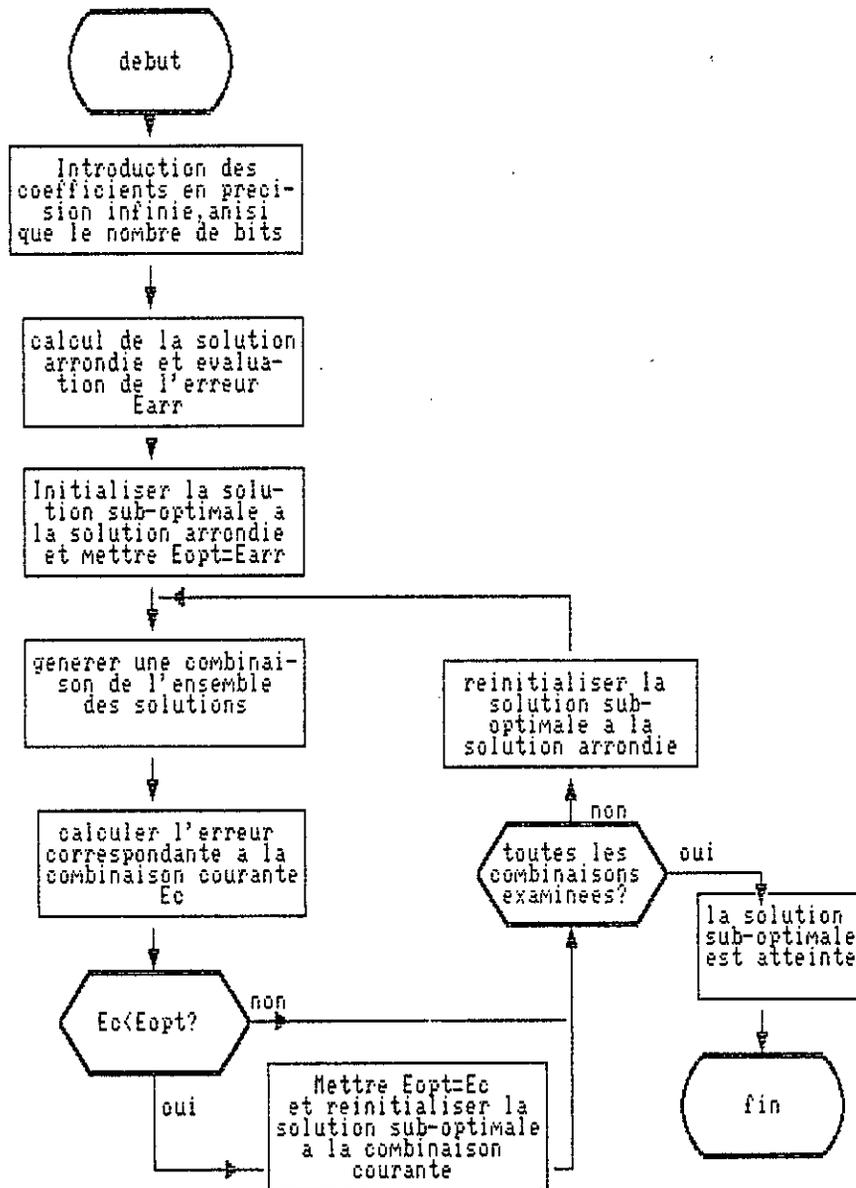


Fig. 5.6 Organigramme de la recherche locale

V.5. RESULTATS ET INTERPRETATIONS

Les exemples traités sont ceux déjà vus dans la section (V.3). La méthode de recherche locale est appliquée en perturbant un élément à la fois de plus ou moins une unité.

L'écart entre la réponse fréquentielle idéale et la réponse fréquentielle du filtre après optimisation est illustré par les graphes de la figure (V.7). Comme pour le cas de l'arrondi, le maximum est atteint au voisinage de la discontinuité, il est inférieur à celui obtenu dans le cas de l'arrondi. En dehors de la bande de transition du filtre, l'écart après optimisation est supérieur à l'écart dû à l'arrondi, cependant, pour des longueurs de mots dépassant 8 bits, il devient très faible. D'après les figures (V.2a) à (V.2d), (V.3a) à (V.3d) et (V.4a) à (V.4d), nous constatons que lorsque la longueur des registres est réduite, la solution arrondie, ainsi que celle obtenue après optimisation par la méthode de recherche locale, s'écartent fortement de la solution obtenue en précision infinie. Cet écart se manifeste par une atténuation assez faible dans la bande d'affaiblissement, la largeur de la bande de transition n'est que légèrement affectée. Lorsque la longueur des registres est augmentée, l'effet de la quantification est de moins en moins visible, si bien que la largeur de la bande de transition et le niveau de l'atténuation sont améliorés au fur et à mesure que le nombre de bits croît.

Par ailleurs, il est à noter que la méthode d'optimisation permet une amélioration des performances du filtre pour des longueurs très faibles des registres. Cette amélioration consiste notamment en la réduction de la largeur de la bande de transition. Cependant, pour des longueurs plus importantes, il devient inutile d'optimiser les performances du filtre, car celles-ci se rapprochent de la solution en précision infinie, et toute amélioration s'avère inutile.

Toutefois, la méthode de recherche locale, qui à l'origine, conçue pour optimiser les filtres numériques synthétisés par la méthode du minimax à coefficients quantifiés, ne s'avère pas très efficace dans le cas de la méthode de synthèse que nous avons utilisé, à savoir la méthode de fenêtrage temporel. Le filtre obtenu par cette méthode n'est optimal en aucun sens, donc, il ne répond à aucun critère d'optimalité comme c'est le cas pour la méthode du minimax, pour laquelle le filtre est optimal dans le sens de minimisation de la norme de Chebyshev définie par la relation (III.8).

La recherche locale est basée sur le même critère, c'est à dire que la solution sub-optimale sera celle qui présentera le minimum de la norme définie par la relation (III.8).

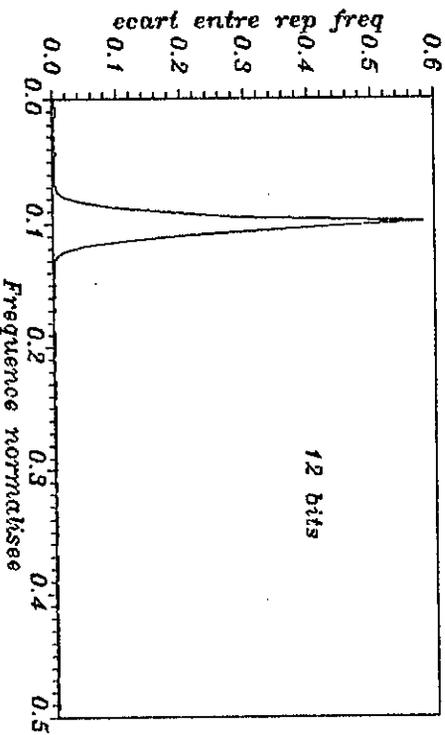
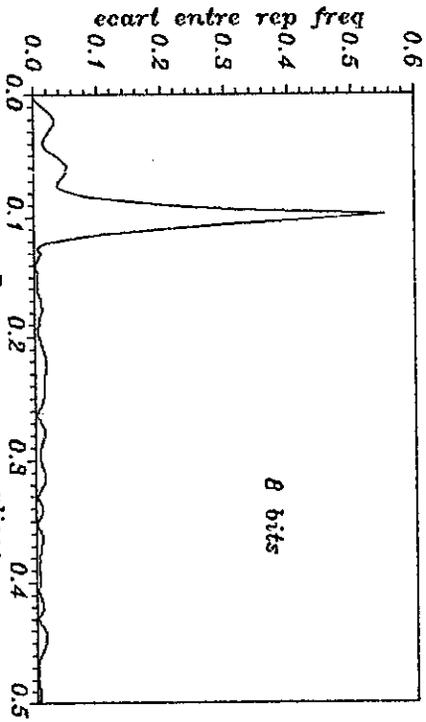
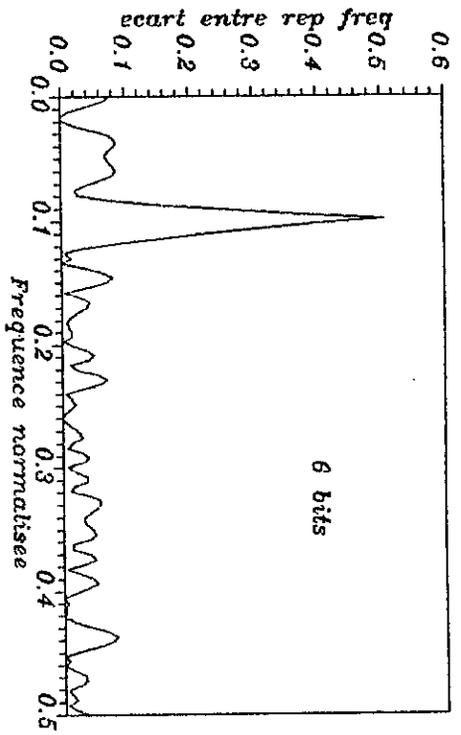
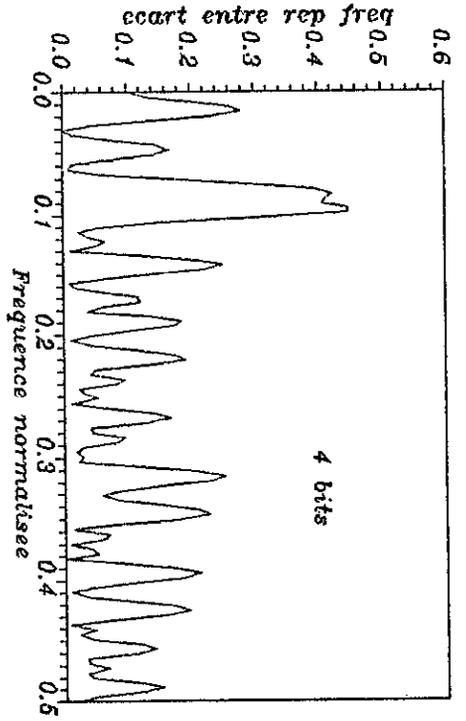


Fig 5.7 Evolution de l'ecart entre la reponse ideale et la reponse apres optimisation par la methode de recherche locale pour un filtre passe bas de frequence de coupure normalisee 0.1

V.5. APPROCHE STATISTIQUE POUR L'ESTIMATION DE LA LONGUEUR DES MOTS [1],[19]

Soient $h_i, i=1, N$ les coefficients de la réponse impulsionnelle du filtre désiré, représentés en virgule fixe.

Considérons le cas d'une quantification par l'arrondi. L'erreur Δh_i introduite sur les coefficients h_i est une variable aléatoire qui prend ses valeurs entre $-q/2$ et $+q/2$, où q est le pas de quantification.

On suppose que la densité de probabilité est uniforme dans cet intervalle. La moyenne Δh_i est alors nulle, et la variance $\sigma_{\Delta h_i}^2$ vaut $q^2/12$ [1],[19].

On suppose aussi que les variables aléatoires Δh_i et Δh_j pour $i \neq j$ sont indépendantes.

La quantité ΔM telle que :

$$\Delta M = \sum_{i=0}^{N-1} \Delta h_i \cdot S_{h_i} \quad (V.15)$$

$$\text{où } S_{h_i} = \frac{\delta M(\omega)}{\delta h_i} \text{ est la sensibilité de } M(\omega) \quad (V.16)$$

est une variable aléatoire de moyenne ΔM nulle et de variance :

$$\sigma_{\Delta M}^2 = \sum_{i=0}^{N-1} \sigma_{\Delta h_i}^2 \cdot S_{h_i}^2 = Q^2 \frac{S_T^2}{12} \quad (V.17)$$

$$\text{où } S_T^2 = \sum_{i=0}^{N-1} S_{h_i}^2 \quad (V.18)$$

En vertu du théorème de la limite centrale, pour de grandes valeurs de N [1], ΔM est une variable aléatoire gaussienne de densité de probabilité :

$$P(\Delta M) = \frac{1}{\sigma_{\Delta M} \sqrt{2\pi}} e^{-\Delta M^2 / 2\sigma_{\Delta M}^2} \quad -\infty < \Delta M < +\infty \quad (V.19)$$

Il est donc possible de borner la variable ΔM dans l'intervalle $[-\Delta M_1, +\Delta M_1]$, soit :

$$y = P(|\Delta M| \leq \Delta M_1) = \frac{2}{\sigma_{\Delta M} \sqrt{2\pi}} \int_0^{\Delta M_1} e^{-\frac{\Delta M^2}{2\sigma_{\Delta M}^2}} d(\Delta M) \quad (V.20)$$

par le changement de variable suivant :

$$\Delta M = x \cdot \sigma_{\Delta M} \quad (V.21)$$

$$\Delta M_1 = x_1 \cdot \sigma_{\Delta M_1} \quad (V.22)$$

on se ramène à l'équation :

$$y = \frac{2}{\sqrt{2\pi}} \int_0^{x_1} e^{-x^2/2} dx \quad (V.23)$$

Cette intégrale est évaluée par une méthode de calcul numérique, les valeurs de y et x_1 sont tabulées dans les ouvrages de mathématiques.

De cette relation, nous pouvons dire que le filtre obtenu remplit les conditions imposées sur son spectre d'amplitude avec une certaine probabilité γ qui doit être aussi proche que possible de la valeur unité.

Compte tenu des équations (V.17), (V.21), (V.22) et du fait que la quantité ΔM_1 doit être inférieure à l'écart maximum, au delà duquel le filtre ne satisfait plus les spécifications (ΔM_{\max}), le pas de quantification est alors majoré comme le montre l'équation suivante :

$$q \leq \frac{\sqrt{12} \Delta M_{\max}(\omega)}{x_1 \cdot S_T} \quad (\text{V.24})$$

Lors de la réalisation, les registres doivent être suffisants pour contenir le coefficient le plus

long. Si ce dernier s'écrit $\sum_{i=-k}^j b_i \cdot 2^i$ avec $b_{-k} \neq 0$ et $b_j \neq 0$, la longueur L du registre doit vérifier la relation :

$$L = 1 + J + K \quad (\text{V.25})$$

Etant donné $q = 2^{-K}$ et sachant que la relation (V.24) est vérifiée, l'équation (V.25) devient :

$$L \geq L(\omega) = 1 + J + \log_2 \frac{x_1 \cdot S_T}{\sqrt{12} \Delta M_{\max}} \quad (\text{V.26})$$

Parfois, il est plus aisé de calculer la sensibilité de la réponse fréquentielle $S_{h_i}^H$, que de calculer la sensibilité du spectre d'amplitude S_{h_i} . Ces deux grandeurs sont liées par la relation [1] :

$$S_{h_i} = \frac{\delta M(\omega)}{\delta h_i} = \cos[\theta(\omega)] \operatorname{Re}[S_{h_i}^H(\omega)] + \sin[\theta(\omega)] \operatorname{Im}[S_{h_i}^H(\omega)] \quad (\text{V.27})$$

où $\theta(\omega)$ est le spectre de phase du filtre.

Pour l'estimation de l'erreur statistique des registres (définie par la relation V.26), prenons le cas d'une structure de réalisation directe définie dans la section 2 du chapitre IV. La relation entre la réponse fréquentielle du filtre et sa réponse impulsionnelle est donnée par l'équation (III.22b).

Nous pouvons écrire dans le cas d'un filtre symétrique :

$$H(\omega) = h(0) + 2 \sum_{i=1}^{\frac{N-1}{2}} h(i) \cos i\omega \quad (\text{V.28})$$

Le module $M(\omega)$ de $H(\omega)$ s'écrit :

$$M(\omega) = \left| h(0) + 2 \sum_{i=1}^{\frac{N-1}{2}} h(i) \cos i\omega \right|$$

D'où il vient :

$$H(\omega) = M(\omega) \cdot e^{j\theta(\omega)}$$

Il est facile de voir que $\theta(\omega)$ est égale à $2m\pi$ lorsque la somme est positive et $(2m+1)\pi$ lorsqu'elle est négative.

Remarque :

La sensibilité que nous calculons est celle d'un filtre non causal. Et comme il a été dit dans le chapitre III, les spectres d'amplitude des deux filtres causal et non causal sont identiques, les sensibilités des spectres d'amplitude des deux filtres sont alors égales.

Compte tenu des relations (V.16) et (V.28), il vient :

$$S_{h_i}^H(\omega) = \begin{cases} 1 & i = 0 \\ 2 \cos i\omega & 1 \leq i \leq \frac{N-1}{2} \end{cases} \quad (\text{V.29})$$

La sensibilité $S_{h_i}(\omega)$ est calculée selon la relation (V.27)

$$S_{h_i}(\omega) = \pm S_{h_i}^H(\omega) \quad (\text{V.30})$$

d'où
$$S_T = \sqrt{S^2(\omega)} = \left(1 + 4 \sum_{i=1}^{\frac{N-1}{2}} \cos^2 i\omega \right)^{1/2} \quad (\text{V.31})$$

Dans le cas traité, les coefficients h_i sont tous inférieurs à 1 en valeur absolue, la valeur de J est alors nulle, et la relation (V.26) s'écrit :

$$L \geq L(\omega) = 1 + \log_2 \frac{x_1 \cdot S_T}{\sqrt{12} \Delta M_{\max}} \quad (\text{V.32})$$

Pour une tolérance ΔM_{\max} fixée à 0.05, la longueur des registres est estimée à 9 bits avec une probabilité $y = 0.95$ (la valeur de x_1 correspondante est 2), et à 8 bits pour une probabilité inférieure correspondant à $x_1 = 1.5$.

Pour une même valeur de la probabilité (y), l'augmentation de la tolérance ΔM_{\max} , entraîne une diminution de la longueur L des registres, comme le montre le tableau (V.1) ci-dessous :

Tableau V.1 : Estimation de la longueur L en fonction de la tolérance ΔM_{\max}

	ΔM_{\max}	0.05	0.1	0.15	0.2
$x = 2$	L	9	8	7	7
$x = 1.5$	L	8	7	7	6

En conclusion, on peut dire que la longueur estimée est d'autant plus faible, que l'on est moins sévère quant à l'erreur imposée sur le spectre d'amplitude du filtre à calculer.

V.6. CONCLUSION

Dans ce chapitre, nous avons introduit le concept de la quantification en l'appliquant aux coefficients d'un filtre synthétisé en précision infinie (ou plus précisément celle de l'ordinateur). Nous avons constaté que cette opération provoque une modification de la réponse fréquentielle du filtre. Ces perturbations sont d'autant plus importantes que le nombre de bits alloués à chaque coefficient est réduit.

Pour essayer de minimiser ces effets, et donc pour optimiser le filtre calculé, nous avons appliqué une méthode d'optimisation à savoir la méthode de la recherche locale.

La solution obtenue ne permet pas une amélioration importante des caractéristiques du filtre, ceci nous amène à dire, que mis à part le cas où le nombre de bits alloués à chaque coefficient est très réduit, toute optimisation du filtre s'avère sans intérêt. Donc, lorsque le nombre de bits est satisfaisant, il est suffisant d'utiliser le filtre avec des coefficients tout simplement arrondis ou tronqués. On remarque aussi, pour la plupart des exemples étudiés, qu'à partir de 8 bits, toutes les réponses fréquentielles (avec précision infinie, arrondies et optimisées) sont pratiquement confondues, il n'est donc pas nécessaire de procéder à l'optimisation.

CHAPITRE VI

SIMULATION DES FILTRES NUMERIQUES R.I.F A PHASE LINEAIRE

VI.1. INTRODUCTION

Une fois la synthèse du filtre établie (chapitre III) et la structure de réalisation choisie (chapitre IV), la simulation sur ordinateur d'un filtre numérique permet de juger les performances du filtre et d'apprécier la qualité d'une éventuelle mise en oeuvre.

Dans ce chapitre, on a donc opté pour l'implantation sur ordinateur de deux types de structure du filtre numérique R.I.F. à phase linéaire : la structure directe et la structure par FFT.

Tout d'abord seront exposés les effets dus à la limitation du nombre de bits des mémoires internes et ceux dus au choix du type d'accumulation des produits (simple ou double). Par la suite quelques exemples de signaux d'entrée générés seront utilisés pour la simulation en précision infinie puis finie selon un programme de calcul.

Une étude comparative entre les performances des deux structures illustrée par des tracés et une conclusion clôtureront ce chapitre.

VI.2. EFFETS DES ERREURS DE CALCUL SUR LES CARACTERISTIQUES DU FILTRE [5],[7],[21]

Lors de l'étape de synthèse (chapitre III), les coefficients des filtres R.I.F. ont été évalués avec une grande précision (précision de l'ordinateur de 54 bits).

En pratique, quand on veut réaliser un tel filtre numérique, les circuits disponibles réalisant les opérations fondamentales de mise en mémoire, addition et multiplication utilisent des registres d'accumulation de longueur très limitée par rapport à celle de l'ordinateur. Il est même préférable que cette longueur soit la plus petite possible pour réduire le coût de la mise en oeuvre. Ce qui revient à quantifier les coefficients des filtres. Cette limitation du nombre de bits des coefficients engendre une altération de la réponse fréquentielle (chapitre V).

- Erreurs dues à la limitation du nombre de bits des mémoires internes pour stocker les coefficients :

Lorsqu'on applique un signal à un filtre dont le nombre de bits des mémoires internes a été limité, le signal filtré se trouve altéré.

En effet, soit b_e le nombre de bits du signal d'entrée et b_c celui des coefficients du filtre, le produit réalisé se trouve sur $(b_e + b_c)$ bits, or la longueur des registres d'accumulation est préalablement fixée à b_r bits pour un filtre donné. Donc, les produits devront être quantifiés à b_r bits pour pouvoir réaliser ce filtre. Une telle opération produit un bruit appelé : bruit de calcul.

Ces erreurs de calcul varient aussi en fonction du type d'accumulateur utilisé propre à la structure en direct.

* Cas de l'accumulateur simple :

La quantification dans ce cas est effectuée après chaque produit avant la sommation selon l'expression (I.19), donc l'erreur est accumulée autant de fois que le produit a lieu.

* Cas de l'accumulateur double :

La quantification s'effectue après sommation de tous les produit de l'expression (I.19).

On peut déjà remarquer que théoriquement l'erreur engendrée par l'accumulateur double doit être inférieure à celle de l'accumulateur simple.

VI.3. TYPES DE SIGNAUX APPLIQUES A L'ENTREE DU FILTRE SIMULE

Les signaux d'entrée générés sont :

1 - Un signal S1 somme de deux sinusoides dont l'une des fréquences choisies se trouve à l'intérieur de la bande passante du filtre utilisé et l'autre complètement à l'extérieur . (figures VI.1a et VI.1b)

Ce signal s'exprime comme suit : (voir fig. VI.1c)

$$S_1 = 0.2 \sin(2\pi f_1 n) + 0.3 \sin(2\pi f_2 n) \quad (VI.1)$$

avec $f_1 = 0.05$ et $f_2 = 0.25$.

2 - Un signal S2 somme de trois sinusoides dont l'une des fréquences choisies est dans la bande passante du filtre, la deuxième dans la bande de transition et la troisième fréquence complètement à l'extérieur (fig. VI.3a, VI.3b et VI.3c).

Ce deuxième signal est donné par : (fig. VI.3d)

$$S_2 = 0.2 \sin(2\pi f_1 n) + 0.3 \sin(2\pi f_2 n) + 0.1 \sin(2\pi f_3 n) \quad (VI.2)$$

avec $f_1 = 0.05$, $f_2 = 0.025$ et $f_3 = 0.1$.

3 - Un signal S3 (fig. VI.5d), somme d'une sinusoïde S avec un bruit blanc préalablement filtré (fig. VI.5a et VI.5b).

$$S(n) = 0.4 \sin(2\pi f n) \quad (VI.3)$$

avec $f = 0.3$ (fig. VI.5c).

Tous ces signaux serviront d'exemples pour simuler différents filtres.

VI.4. ELABORATION D'UN PROGRAMME ET RESULTATS

VI.4.1. Description :

Le programme de simulation sur ordinateur du filtre R.I.F, consiste en :

- 1 - L'introduction des coefficients du filtre synthétisé au préalable.
- 2 - L'introduction des données du signal d'entrée.
- 3 - Le choix de la structure à appliquer (directe ou FFT).
- 4 - Le calcul des échantillons du signal de sortie du filtre.

VI.4.2. Exemples :

Pour mieux illustrer nos exemples, nous avons opté pour quelques filtres nous permettant, en vue des signaux de sortie obtenus, de juger les qualités et performances effectives de ces filtres R.L.F synthétisés auparavant par la méthode de fenêtrage.

Les filtres simulés dans les exemples qui suivront sont :

- Un filtre passe-bas d'ordre 61 de fréquence de coupure normalisée $f_c = 0.1$, qu'on appellera : Filtre 1.
- Un filtre passe-bas d'ordre 21 de même fréquence de coupure que le précédent qu'on appellera : Filtre 2.
- Un filtre passe-bande d'ordre 81, sans bande à l'origine, de fréquences de coupures normalisées : $fc_1 = 0.2$ et $fc_2 = 0.4$.

Ces trois types de filtres ont été synthétisés en leur appliquant la fenêtre de Kaiser avec $\alpha = 7$.

- Exemple 1 :

Il consiste à appliquer à l'entrée du filtre 1, le signal S1, somme de deux sinusoides (VI.3).

Le filtrage s'effectue selon les deux types de structures directe et FFT.

En précision infinie, les résultats sont illustrés par les figures (VI.1) et (VI.2). En précision finie, la longueur des mots binaires des coefficients du filtre a été fixée à 4,6,8 et 12 bits.

La sortie obtenue pour les quatre cas est représentée par les figures (VI.7) et (VI.8).

- Exemple 2 :

Aux entrées des filtres 1 et 2, on applique le signal S2, somme de trois sinusoides.

Les résultats en précision infinie sont illustrés par les figures (VI.3) et (VI.4), et en précision finie par les figures (VI.9), (VI.10) en ce qui concerne le filtre 1 et (VI.11), (VI.12) pour le filtre 2.

- Exemple 3 :

Tout d'abord, on commence par générer un bruit blanc, centré de variance unité (fig. (VI.5a) et (VI.6a)), qu'on filtre à travers le filtre 1 pour les deux types de structure (fig. (VI.5b) et (VI.6b)).

Le bruit blanc filtré est ensuite superposé à un signal sinusoidal S (VI.3) dont la fréquence de coupure se trouve à l'extérieur de la bande passante du filtre 1.

Ce signal S3 est alors appliqué au filtre passe-bande décrit au début de cette section (VI.4).

Les résultats en précision infinie sont illustrés par les figures (VI.5) et (VI.6).

VI.4.3. Résultats et interprétations :**- Exemple 1 :**

On constate effectivement d'après les figures citées, que le filtrage de S1 est correctement effectué. La fréquence f_2 , se trouvant à l'extérieur de la bande passante du filtre 1, a bien été rejetée. cependant, on remarque que le signal de sortie pour les 25 ou 30 premières valeurs est pratiquement nul et ceci est très net dans le cas de la structure directe. Ceci s'explique par le fait que pour les tous premiers termes, la convolution est incomplète.

C'est la phase transitoire, il faudra donc attendre un certain ordre pour avoir le régime permanent et récupérer la sinusoïde de fréquence $f_1 = 0.05$. Etant donné que le gain en fréquence des filtres utilisés est égal à l'unité, l'amplitude des signaux d'entrées après filtrage s'en trouve préservée. En précision finie, on remarquera cependant que la sinusoïde de la sortie pour 4 bits est distordue, beaucoup moins pour le cas de 6 bits.

Le filtrage devient aussi correct que pour une précision infinie dans le cas 8 et 12 bits.

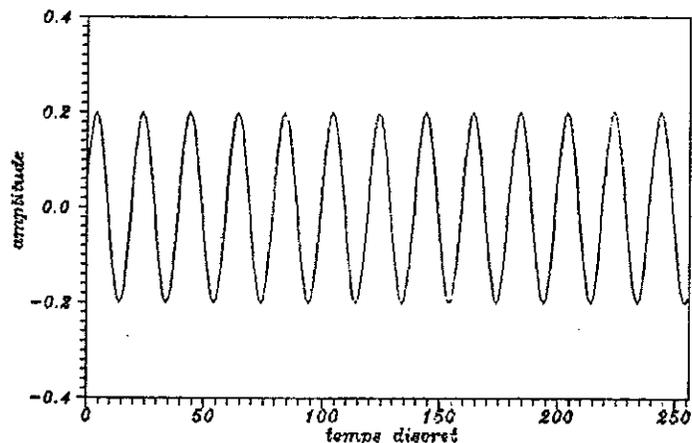
- Exemple 2 :

Les figures (VI.3) pour la forme directe et (VI.4) pour la forme FFT, montrent que le signal de sortie est quelques peu plus distordu en (f) qu'en (e) et que l'influence de la sinusoïde de fréquence normalisée 0.1 est plus évidente en (f) qu'en (e). Ceci s'explique par le fait que le filtre 1 d'ordre 61 a une bande de transition beaucoup plus étroite que celle du filtre 2 d'ordre 21; et donc une transition au niveau de la fréquence de coupure de 0.1, beaucoup plus raide.

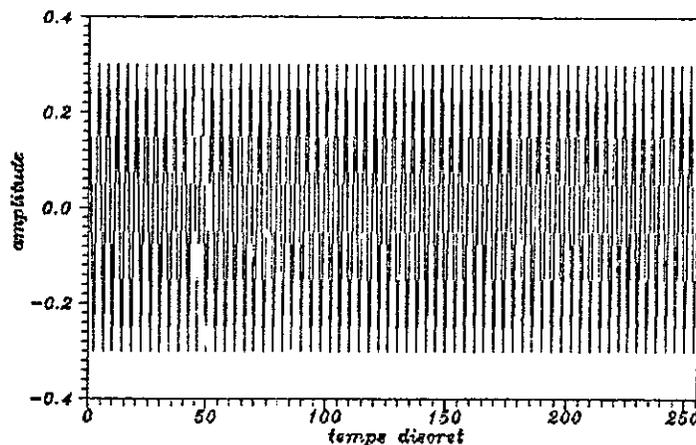
- Exemple 3 :

Le filtrage du bruit effectué (fig. (VI.5b) et (VI.6b)), montre que les fréquences supérieures à la fréquence de coupure du filtre 1, ont été éliminées.

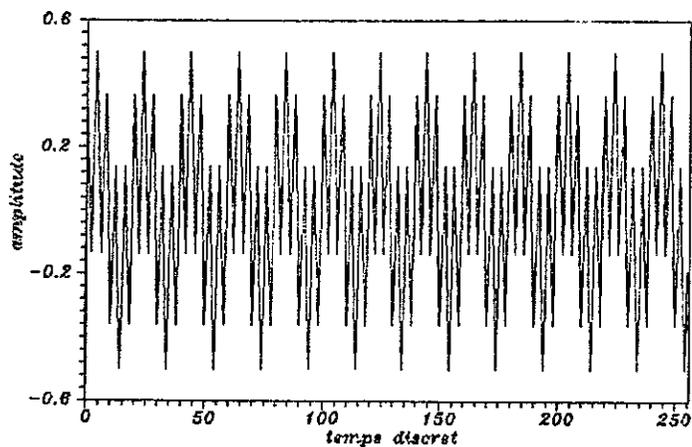
Par contre, le filtre passe bande a permis d'éliminer complètement le bruit filtré qu'on avait superposé au signal $S = 0.4 \sin(2\pi f.n)$, $f = 0.3$.



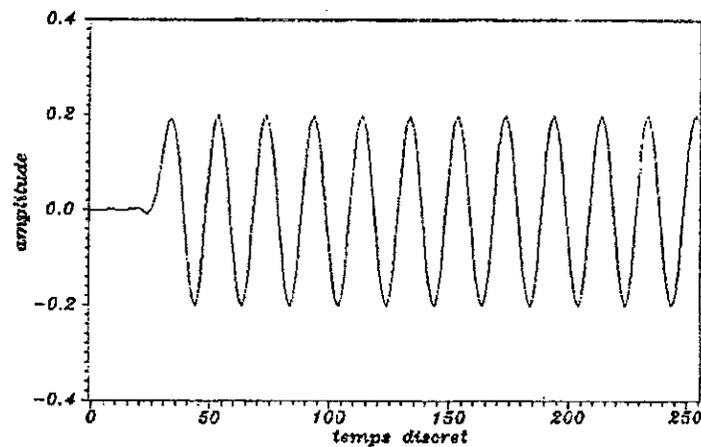
(a) SINUSOIDE DE FREQUENCE NORMALISEE 0.05



(b) SINUSOIDE DE FREQUENCE NORMALISEE 0.25

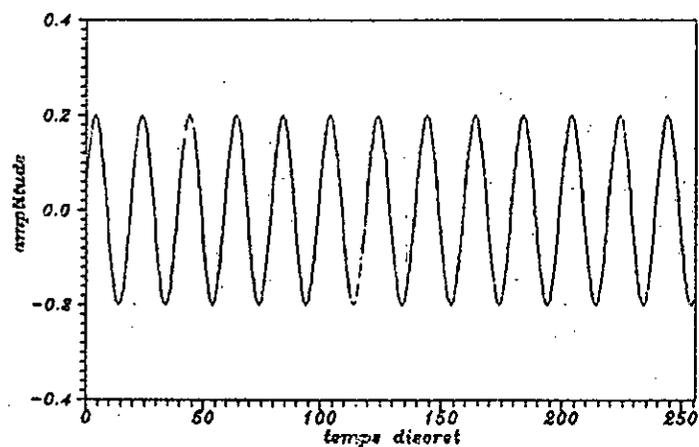


(c) SIGNAL D'ENTREE DU FILTRE -1-
(somme des deux sinusoides (a) et (b))

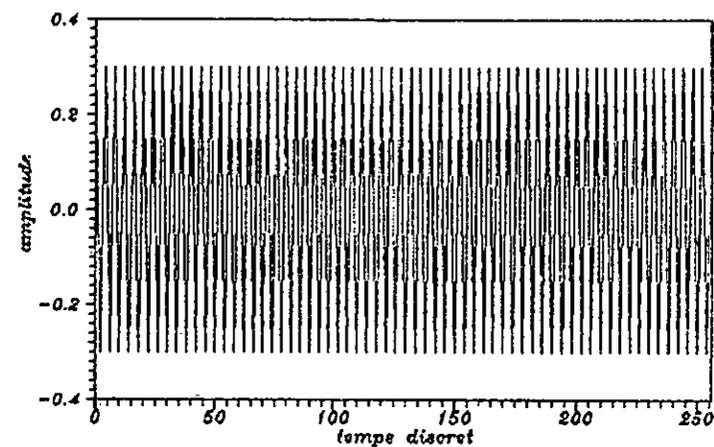


(d) SIGNAL DE SORTIE DU FILTRE -1-

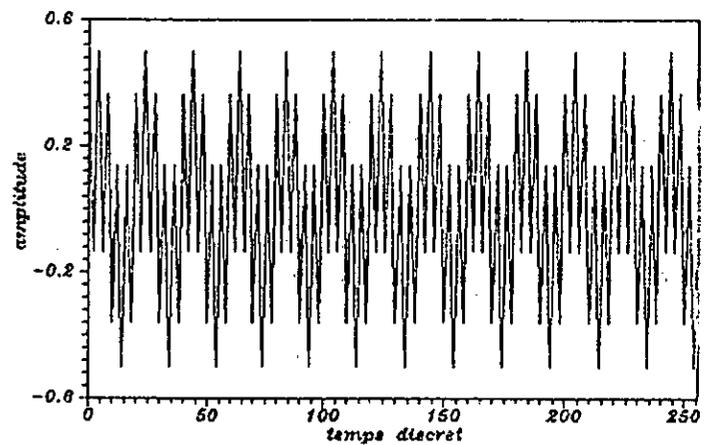
Fig 6.1 : FILTRAGE D'UN SIGNAL SOMME DE DEUX SINUSOIDES
AVEC UNE PRECISION INFINIE SELON UNE STRUCTURE EN DIRECT



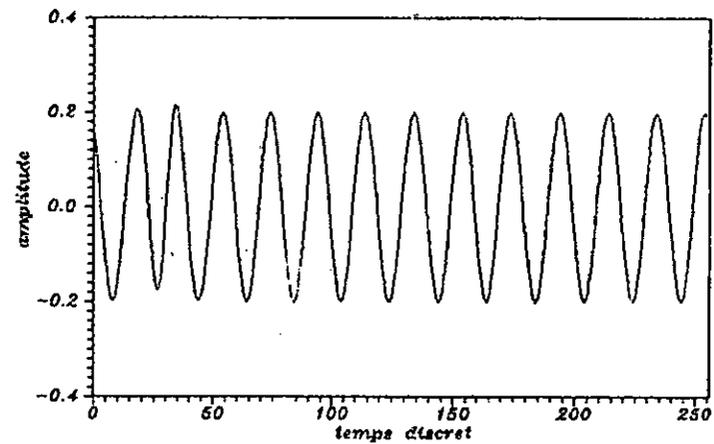
(a) SINUSOIDE DE FREQUENCE NORMALISEE 0.05



(b) SINUSOIDE DE FREQUENCE NORMALISEE 0.25



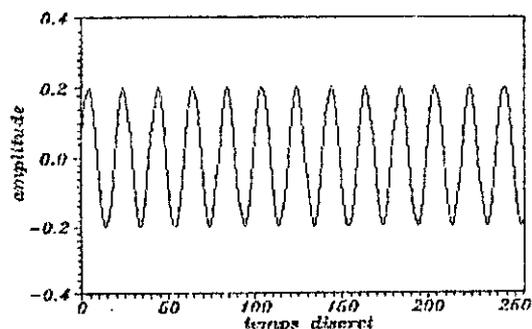
(c) SIGNAL D'ENTREE DU FILTRE -1-
(somme des deux sinusoides (a) et (b))



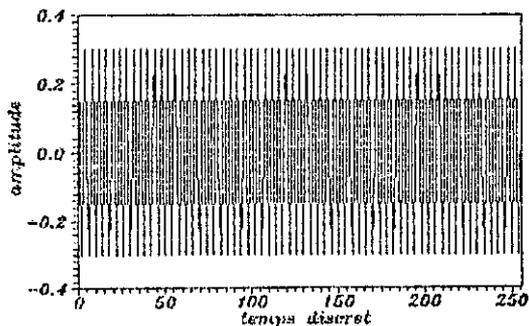
(d) SIGNAL DE SORTIE DU FILTRE -1-

107

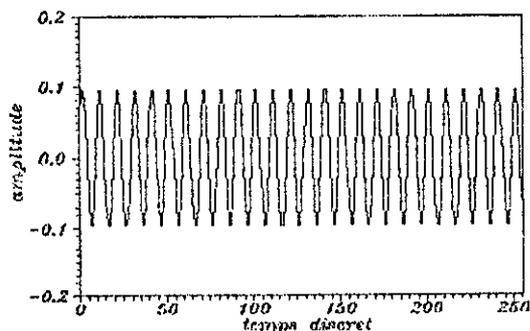
Fig 6.2 : FILTRAGE D'UN SIGNAL SOMME DE DEUX SINUSOIDES
AVEC UNE PRECISION INFINIE SELON UNE STRUCTURE EN FFT.



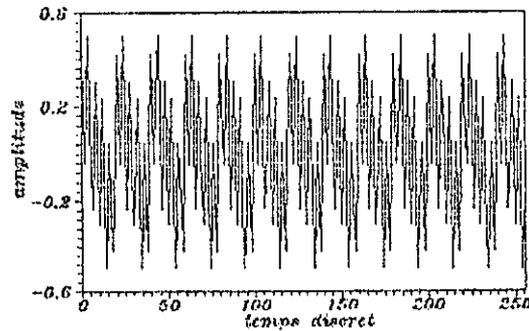
(a) SINUSOIDE DE FREQUENCE NORMALISEE 0.05



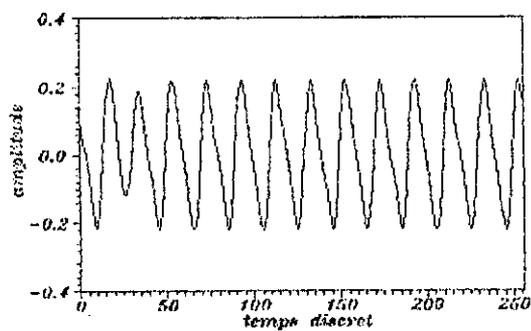
(b) SINUSOIDE DE FREQUENCE NORMALISEE 0.25



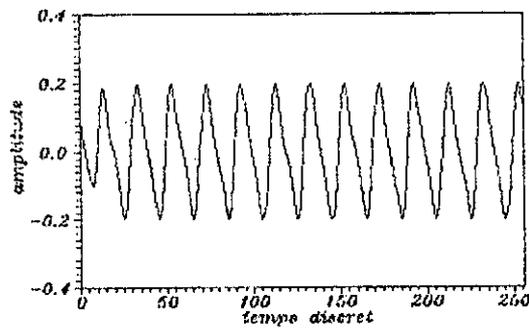
(c) SINUSOIDE DE FREQUENCE NORMALISEE 0.1



(d) SIGNAL D'ENTREE DU FILTRE
(somme des trois sinusoides (a), (b) et (c))



(e) SIGNAL DE SORTIE DU FILTRE -1-



(f) SIGNAL DE SORTIE DU FILTRE -2-

Fig 6.4 : FILTRAGE D'UN SIGNAL SOMME DE TROIS SINUSOIDES
AVEC UNE PRECISION INFINIE SELON UNE STRUCTURE EN FFT

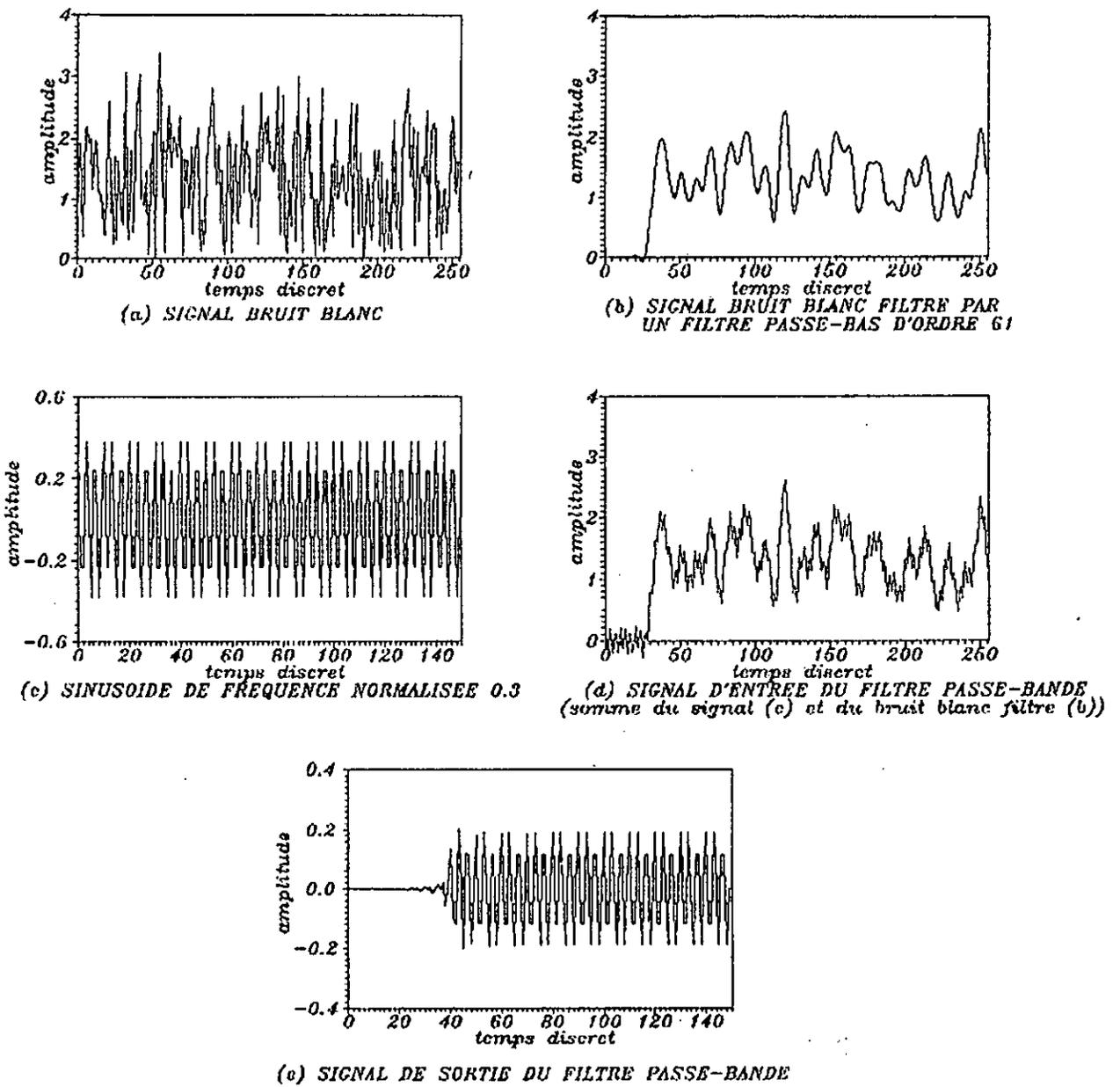


Fig 6.5 : FILTRAGE D'UN SIGNAL BRUIT BLANC PREALABLEMENT FILTRE SUPPERPOSE A UNE SINUSOIDE AVEC UNE PRECISION INFINIE SELON UNE STRUCTURE EN DIRECT

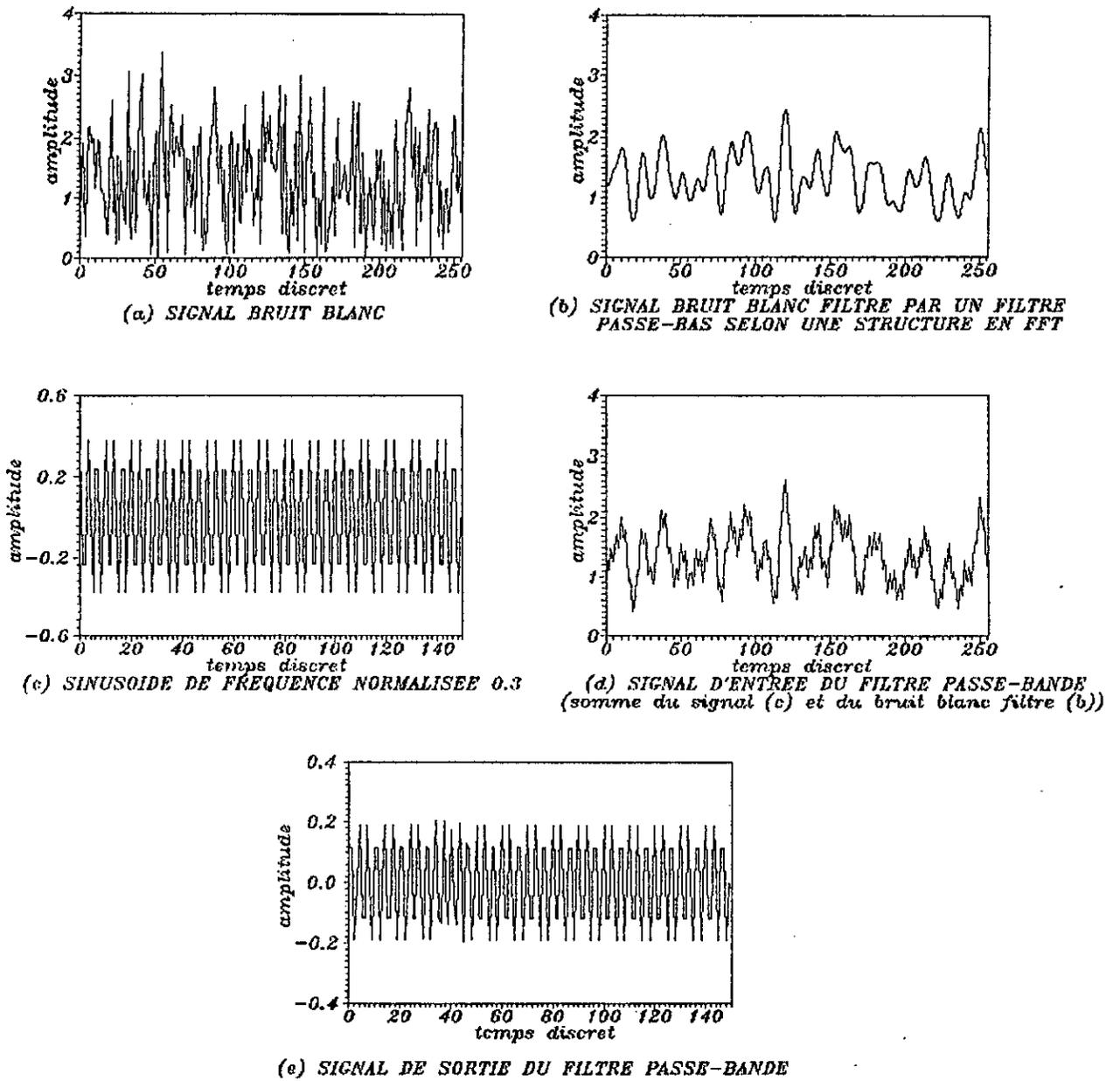
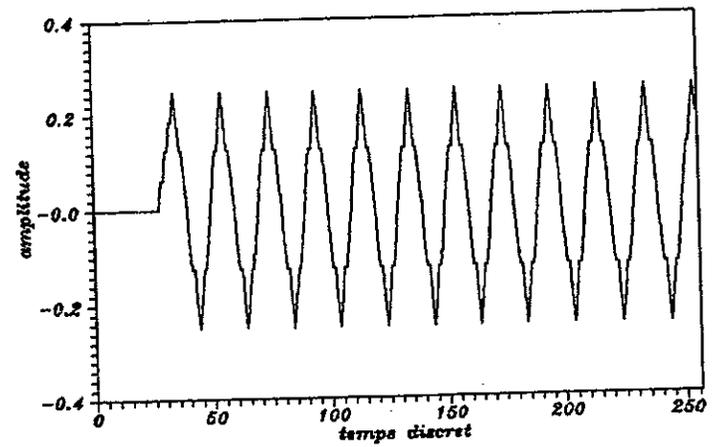
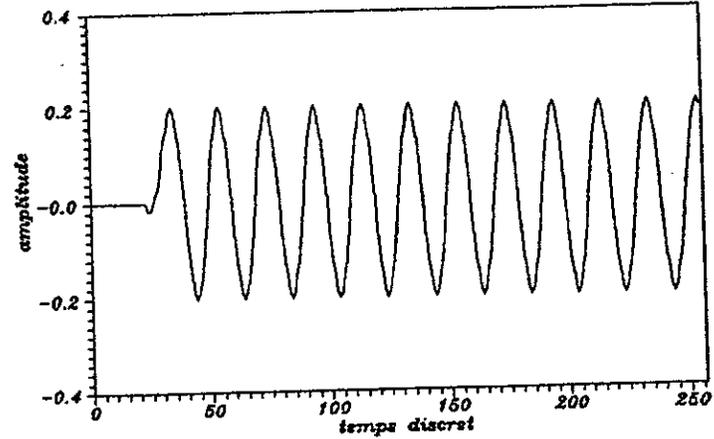


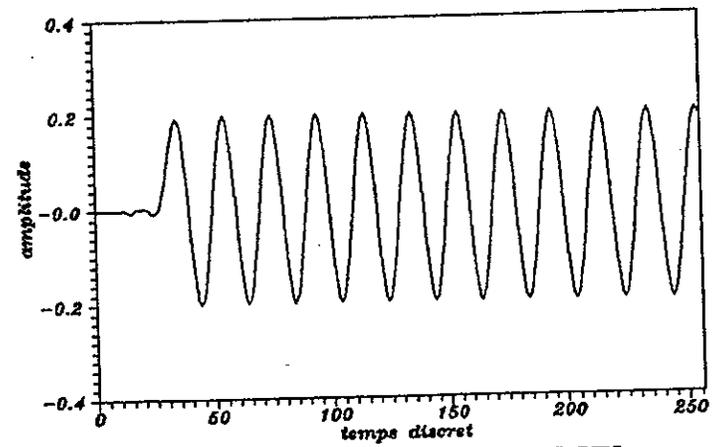
Fig 6.6 : FILTRAGE D'UN SIGNAL BRUIT BLANC PREALABLEMENT FILTRE SUPPERPOSE A UNE SINUSOIDE AVEC UNE PRECISION INFINIE SELON UNE STRUCTURE EN FFT



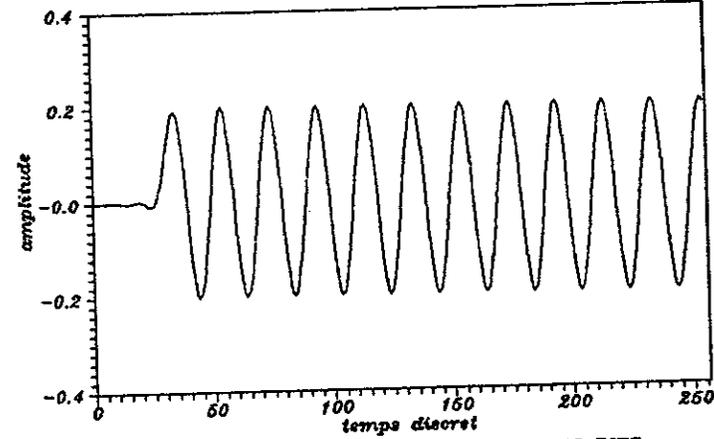
(a) SIGNAL DE SORTIE ACCUMULE SUR 4 BITS



(b) SIGNAL DE SORTIE ACCUMULE SUR 6 BITS

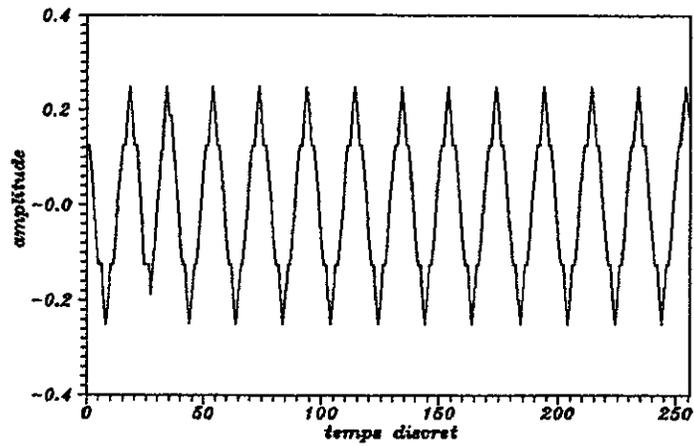


(c) SIGNAL DE SORTIE ACCUMULE SUR 8 BITS

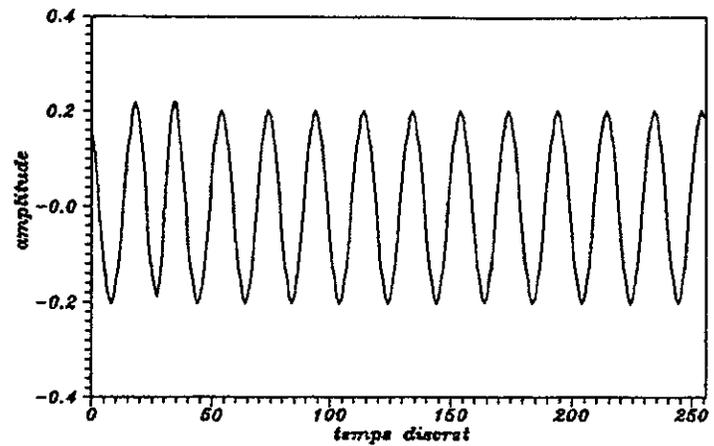


(d) SIGNAL DE SORTIE ACCUMULE SUR 12 BITS

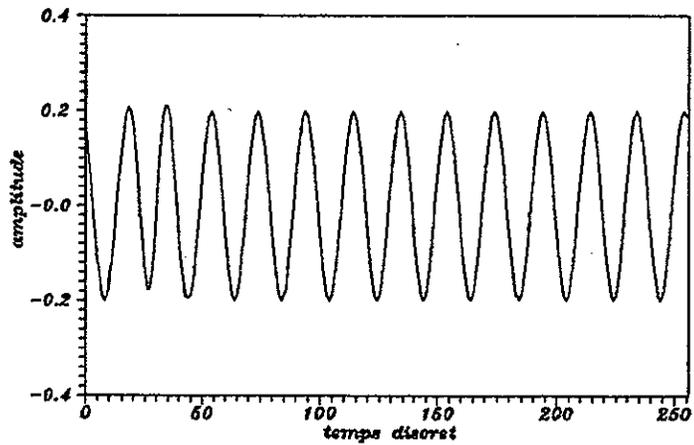
Fig 6.7 : FILTRAGE DU SIGNAL SOMME DE DEUX SINUSOIDES
AVEC PRECISION FINIE SELON UNE STRUCTURE EN DIRECT



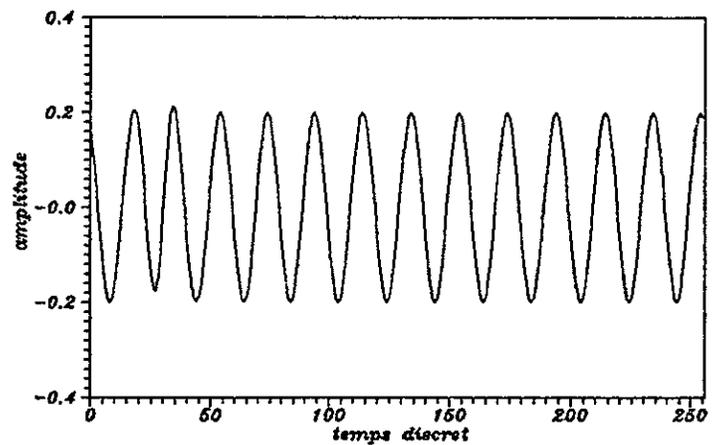
(a) SIGNAL DE SORTIE ACCUMULE SUR 4 BITS



(b) SIGNAL DE SORTIE ACCUMULE SUR 6 BITS

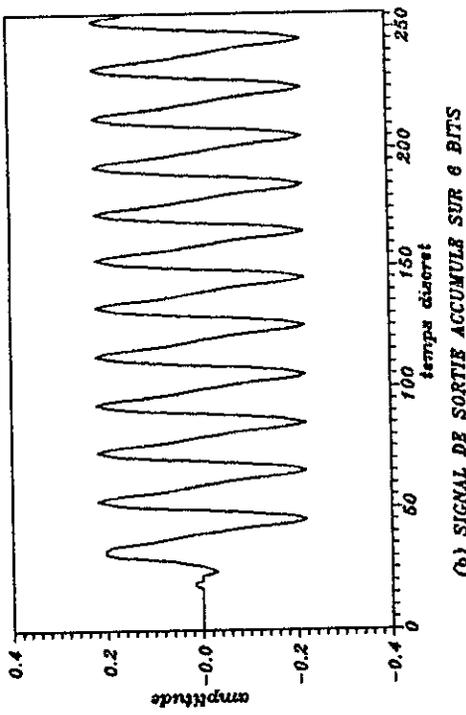


(c) SIGNAL DE SORTIE ACCUMULE SUR 8 BITS

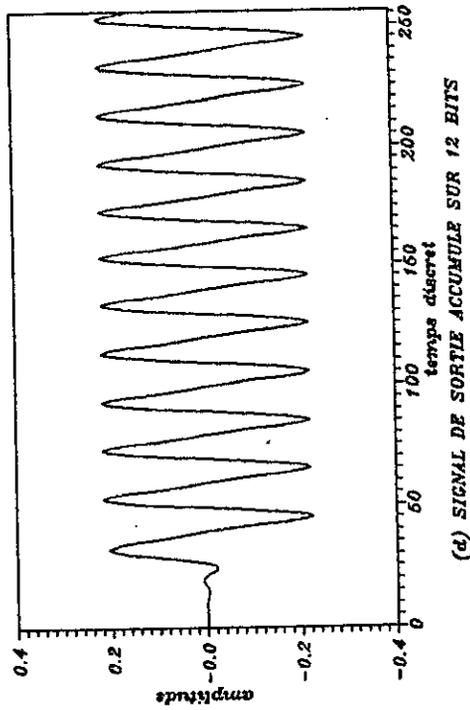


(d) SIGNAL DE SORTIE ACCUMULE SUR 12 BITS

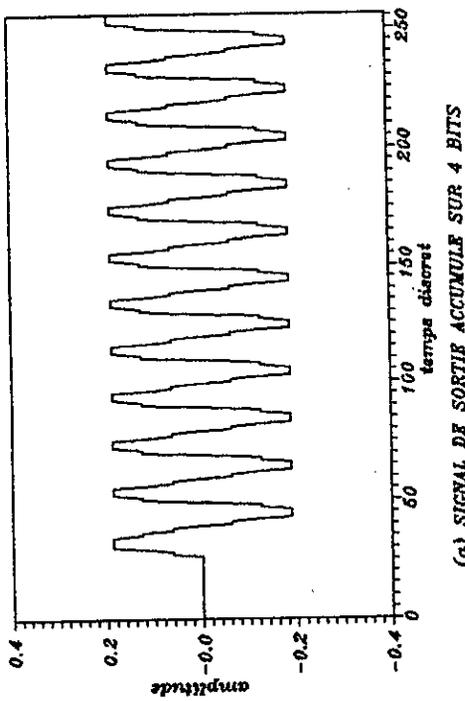
Fig 6.8 : FILTRAGE DU SIGNAL SOMME DE DEUX SINUSOIDES
AVEC PRECISION FINIE SELON UNE STRUCTURE EN FFT



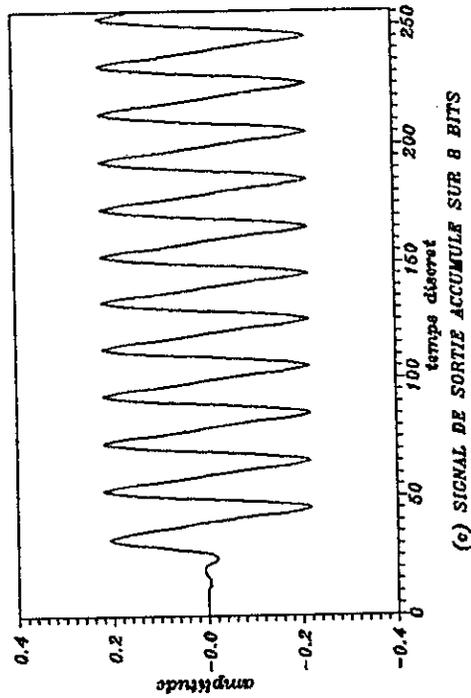
(b) SIGNAL DE SORTIE ACCUMULE SUR 6 BITS



(d) SIGNAL DE SORTIE ACCUMULE SUR 12 BITS

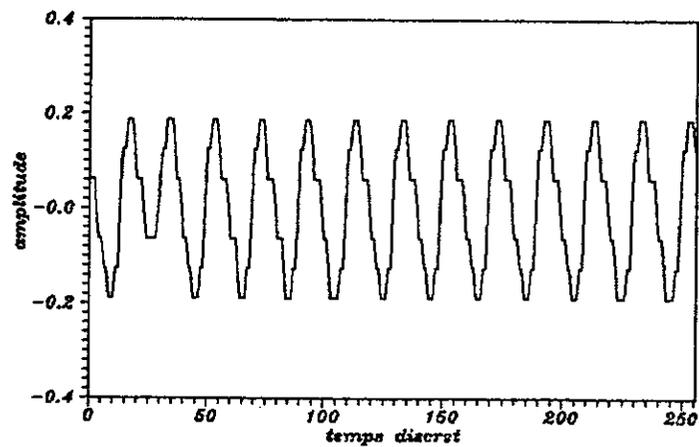


(a) SIGNAL DE SORTIE ACCUMULE SUR 4 BITS

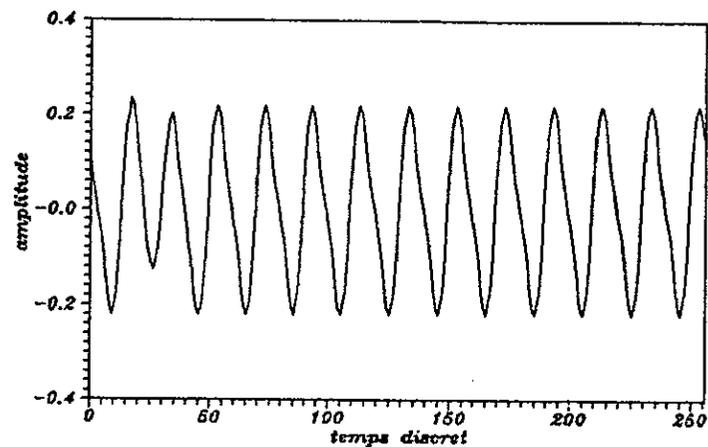


(c) SIGNAL DE SORTIE ACCUMULE SUR 8 BITS

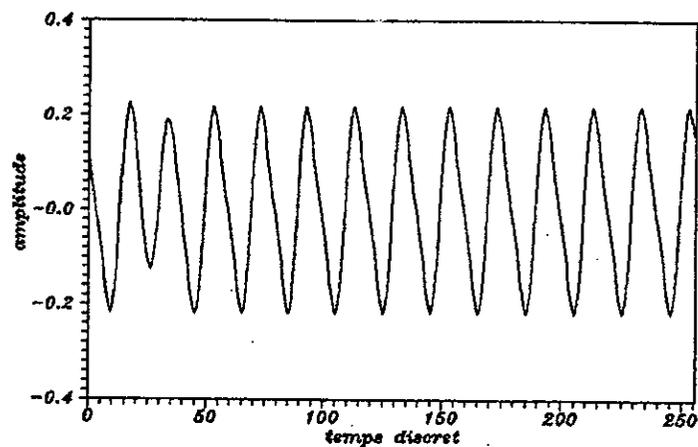
Fig. 6.9 : FILTRAGE DU SIGNAL SOMME DE TROIS SINUSOÏDES AVEC PRECISION FINIE SELON UNE STRUCTURE EN DIRECT (FILTRE -1-)



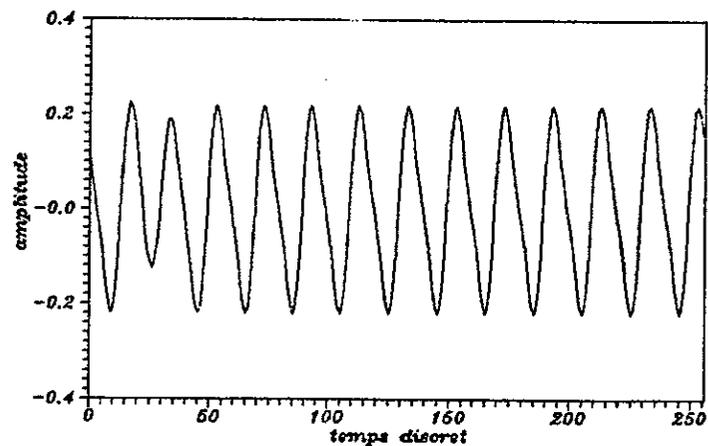
(a) SIGNAL DE SORTIE ACCUMULE SUR 4 BITS



(b) SIGNAL DE SORTIE ACCUMULE SUR 6 BITS

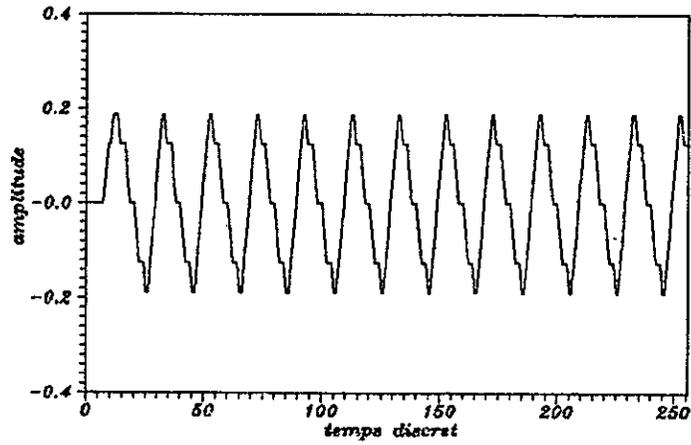


(c) SIGNAL DE SORTIE ACCUMULE SUR 8 BITS

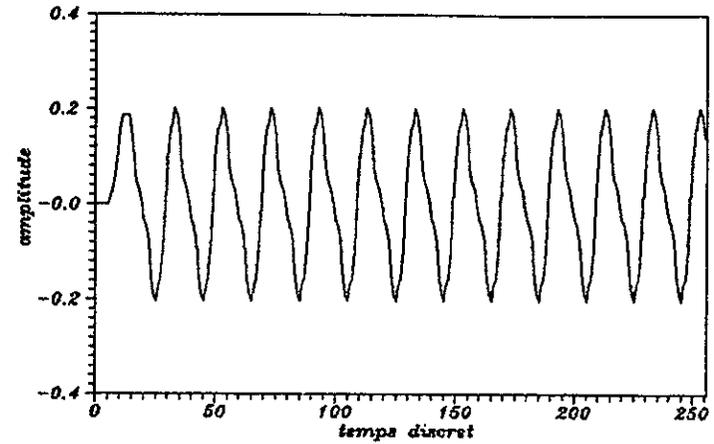


(d) SIGNAL DE SORTIE ACCUMULE SUR 12 BITS

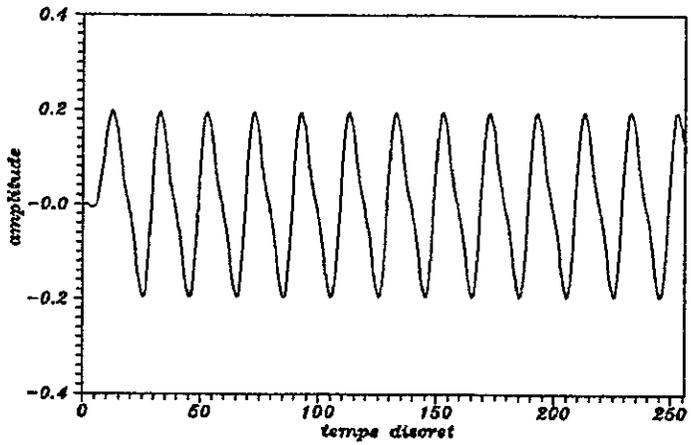
Fig 6.10 : FILTRAGE DU SIGNAL SOMME DE TROIS SINUSOIDES
AVEC PRECISION FINIE SELON UNE STRUCTURE EN FFT (FILTRE -1-)



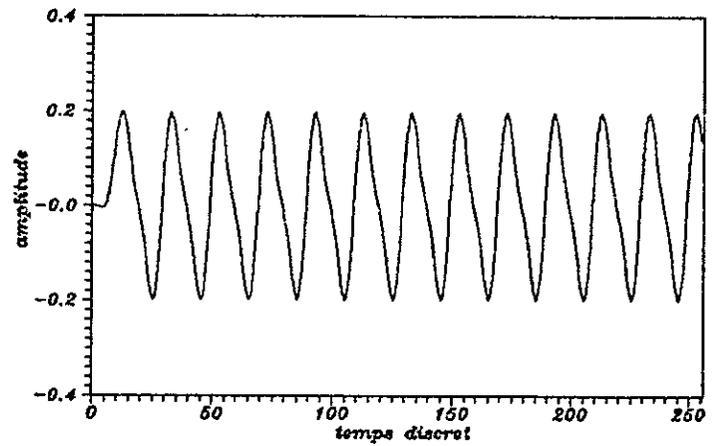
(a) SIGNAL DE SORTIE ACCUMULE SUR 4 BITS



(b) SIGNAL DE SORTIE ACCUMULE SUR 6 BITS

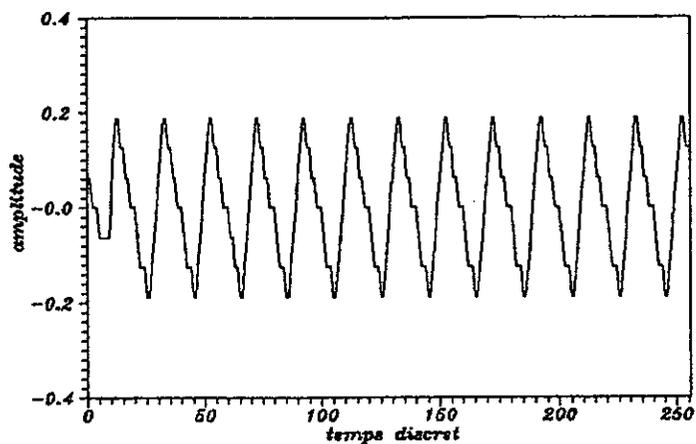


(c) SIGNAL DE SORTIE ACCUMULE SUR 8 BITS

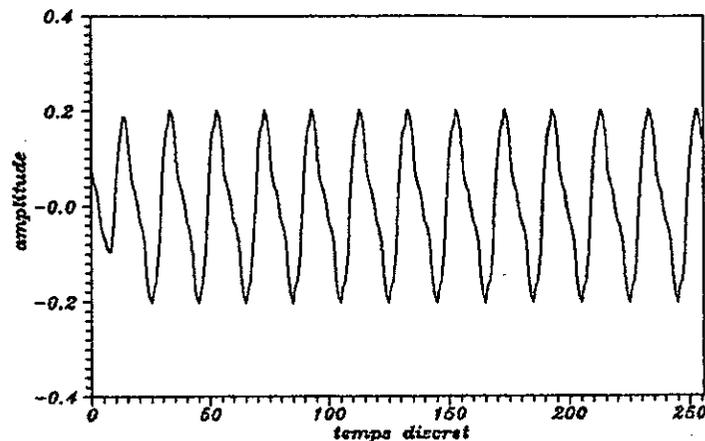


(d) SIGNAL DE SORTIE ACCUMULE SUR 12 BITS

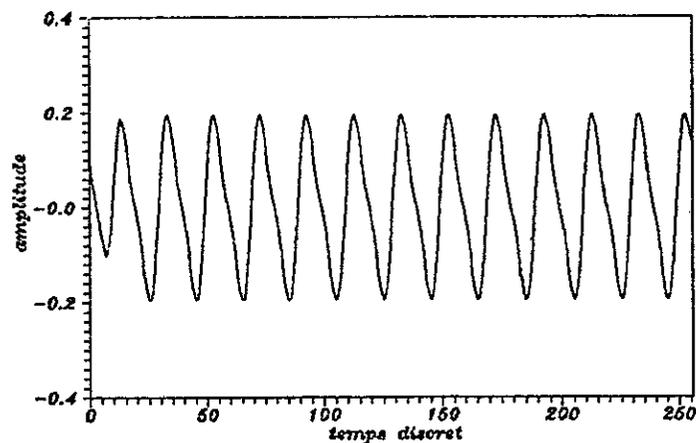
Fig 6.11 : FILTRAGE DU SIGNAL SOMME DE TROIS SINUSOIDES
AVEC PRECISION FINIE SELON UNE STRUCTURE EN DIRECT (FILTRE -2-)



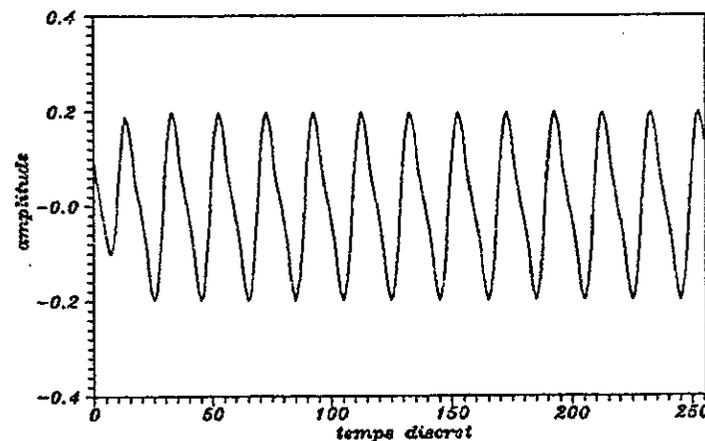
(a) SIGNAL DE SORTIE ACCUMULE SUR 4 BITS



(b) SIGNAL DE SORTIE ACCUMULE SUR 6 BITS



(c) SIGNAL DE SORTIE ACCUMULE SUR 8 BITS



(d) SIGNAL DE SORTIE ACCUMULE SUR 12 BITS

Fig 6.12 : FILTRAGE DU SIGNAL SOMME DE TROIS SINUSOIDES
AVEC PRECISION FINIE SELON UNE STRUCTURE EN FFT (FILTRE -2-)

VI.5. CONCLUSION

La simulation d'un filtre numérique R.I.F. à phase linéaire nous a permis d'étudier les performances des deux types de structures, directe et FFT, puis de les comparer du point de vue qualité de filtrage et temps d'exécution.

Il est évident que la structure en FFT est celle qui permet un gain important en temps d'exécution et surtout lorsque l'ordre du filtre et le nombre d'échantillons à filtrer sont importants.

Pour ce qui est de la qualité de filtrage, la différence n'est pas très appréciable.

Globalement, la simulation a permis de voir le comportement des filtres R.I.F. lorsqu'ils sont implémentés avec une précision très limitée.

On a pu remarquer qu'avec un nombre de bits très faible (4 bits), l'allure du signal filtré (malgré quelques distorsions) a été nette et claire (fig. (VI.7) et (VI.8)). Ceci n'est pas possible dans le cas d'un filtre R.I.I., où l'accumulation des erreurs de calcul peut masquer complètement le signal de sortie lorsque le nombre de bits est de l'ordre de 4 [7].

Enfin, nous pouvons dire que la simulation est une étape indispensable avant la mise en oeuvre d'un filtre numérique. Cette étape nous permet de corriger toutes anomalies associées avec la précision limitée et de choisir le nombre de bits qui donne un bon compromis entre la qualité du filtrage (performances) et le coût du filtre.

CONCLUSION

Lors de cette étude, notre premier objectif était l'application de la méthode de fenêtrage temporel pour synthétiser des filtres numériques R.I.F. à phase linéaire. Cette méthode introduit des perturbations dans le spectre d'amplitude du filtre calculé. Celles-ci sont connues sous le nom du phénomène de Gibbs et se présentent sous forme de bandes de transition au niveau des discontinuités, et d'ondulations autour de ces mêmes points. Pour obtenir un bon filtre, il faut pouvoir réduire les bandes de transitions et augmenter l'atténuation en bandes d'affaiblissement simultanément.

Dans ce sens, des fenêtres particulières telle que le fenêtre de Kaiser sont assez performantes, ce qui n'est pas le cas avec les fenêtres classiques qui, elles, nous permettent de faire seulement un compromis entre les deux paramètres.

Une fois la synthèse effectuée en précision infinie, nous avons procédé à une quantification des coefficients par l'arrondi, nous avons ensuite essayé d'optimiser le filtre obtenu par la méthode de recherche locale.

Les résultats obtenus nous laissent dire que l'optimisation ne permet pas une amélioration notable des spécifications du filtre, et particulièrement lorsque la longueur des registres est suffisante (à partir de 8 bits).

La méthode de recherche locale ne s'est pas avérée d'une très grande efficacité dans le cas d'une synthèse avec la méthode de fenêtrage temporel, dans la plupart des cas traités, exception faite du cas où le nombre de bits est très réduit, il suffit d'effectuer un simple arrondi.

La mise en oeuvre d'un filtre numérique nécessite un arrangement des opérations d'addition, de multiplication et de mise en mémoire. Tout arrangement définit une structure de réalisation.

Parmi les avantages des filtres R.I.F., on compte :

- Un déphasage linéaire.
- Une structure non récursive qui empêche toute propagation des erreurs.
- Une possibilité d'utilisation des algorithmes rapides pour implémenter la fonction filtrage.

Pour ce qui est de la simulation, et étant donné les résultats de l'optimisation, il nous a semblé suffisant d'utiliser un filtre, en premier lieu, avec une précision infinie, et en second lieu avec des coefficients quantifiés par l'arrondi pour différentes valeurs de la longueur du mot. La qualité du filtrage dépend essentiellement de la précision avec laquelle le filtre a été synthétisé et les caractéristiques du filtre utilisé. En d'autres termes, le filtrage est d'autant meilleur que le filtre utilisé présente une bande de transition étroite, une bonne atténuation en bande d'affaiblissement, et que ses coefficients soient calculés avec une bonne précision, et ceci pour une structure donnée.

REFERENCES BIBLIOGRAPHIQUES

- [1] A.Antoniou, *Digital filters : Analysis and design*, Mc Graw-hill, New york , 1979.
- [2] F.DE COULON , *Signal theory and processing*, Artech house, Washington , 1986 (dedham)
- [3] M.Labarrere , J.P.Krief & B.Gimonet , *Le filtrage numérique et ses applications* , Cepadus-édition , Toulouse , 1981.

- [4] M.Kunt , *Traitement numérique des signaux* , DUNOD Paris, 1981.
- [5] M.Bellanger , *Traitement numérique du signal* , Masson , Paris , 1987.
- [6] L.Mitiche & M.A.A Belouchrani, *Etude unifiée des transformations fréquentielles numériques et application à quelques filtres prototypes*, P.F.E 1991.
- [7] N.Sadi-Ahmed, *Simulation des filtres numériques R.L.I en virgule fixe*, P.F.E 1991.

- [8] L.R.Rabiner & B.Gold , *Theory and application of digital signal processing*, Prentice hall, New Jersey, 1975.
- [9] A.V.Oppenheim & R.W.Schafer, *Digital signal processing* , Prentice hall , New Jersey , 1975.
- [10] C.H.Chen, *Signal processing handbook* , Marcel Dekker, New York 1988.
- [11] C.T.Mullis & R.A.Roberts , *Digital signal processing*, addison Wesley, Readings MA, 1987.
- [12] A.Laib , *Optimisation minimax des filtres numériques R.L.F avec contraintes supplémentaires*, Magister 1992.
- [13] C.S.Williams , *Designing digital filters*, PHI , 1986.
- [14] F.J. Harris , " *On the use of windows for harmonic analysis with the direct Fourier Transform* ", Proceeding of IEEE , Vol 56 N° 1 pp.51-83 , January 1978.
- [15] R.B.Blackman & J.W.Tukey , *The measurement of power spectra*, Dover, New York 1958.

-
- [16] J.F Kaiser of R.W.Schofer, " *On the use of the lo-Sinh windows for spectrum analysis*"
IEEE Trans.on acoustic, Speech of signal processing , Vol ASSP.28 N°1 , February 1980.
- [17] J.L.ferman, *Les méthodes rapides de transformation du signal : Fourier, Walsh, Hadalard, Hoar*, Ed Masson, Paris, 1980
- [18] P.Siohan & A.Benslimane," *Synthèse des filtres numériques non récursifs à phase linéaire et coefficients de longueur finie* ", Annales des télécom, 39 N° 7-8, pp.307-322, France 1984.
- [19] A.Avenhaus, " *On the design of digital filters with coefficients of word lenght*", IEEE tra s Audio Electroacoust., vol AV 20, pp.206-212, Aug.1972.
- [20] R.E. Crochiere, " *A new statistical approch to the coefficient word lenght problem for digital filters*", IEEE Trans. on circuits and systems, vol.CAS-22 , N° 3 , March 1975.
- [21] F.Grenez, " *Synthèse des filtres numériques non récursifs à coefficints quantifiés*", Annales de Télécom, 34, N° 1-2, pp.33-39, France 1979.
- [22] R.Boite & MKunt, *Trattement de la parole*, Presses polytechniques romandes, Lausanne, 1987.
- [23] A.Fipoulis, *Signal processing*, Mc Graw hill, USA, 1977

4 - Fenêtre de De Lavalé-poussin :

$$w(n) = \begin{cases} 1 - 6\left(\frac{n}{(N-1)/2}\right)^2 \cdot \left(1 - \frac{|n|}{(N-1)/2}\right) & 0 \leq |n| \leq \frac{N-1}{4} \\ 2\left(1 - \frac{|n|}{(N-1)/2}\right)^3 & \frac{N-1}{4} \leq |n| \leq \frac{N-1}{2} \end{cases} \quad (\text{A1.5})$$

5 - Fenêtre de Tuckey :

$$w(n) = \begin{cases} 1 & 0 \leq |n| \leq \alpha \frac{N-1}{2} \\ 0.5 \cdot \left[1 + \cos\left(\pi \frac{n - \alpha \frac{N-1}{2}}{2(1-\alpha)\frac{N-1}{2}}\right) \right] & \alpha \frac{N-1}{2} \leq |n| \leq \frac{N-1}{2} \end{cases} \quad (\text{A1.6})$$

6 - Fenêtre de Bohman :

$$w(n) = \begin{cases} \left(1 - \frac{|n|}{(N-1)/2}\right) \cos\left(\pi \frac{|n|}{(N-1)/2}\right) + \frac{1}{\pi} \sin\left(\pi \frac{|n|}{(N-1)/2}\right) & 0 \leq n \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{A1.7})$$

7 - Fenêtre de Poisson :

$$w(n) = \begin{cases} \exp\left(-\alpha \frac{|n|}{(N-1)/2}\right) & 0 \leq |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{A1.8})$$

8 - Fenêtre de Hanning-Poisson :

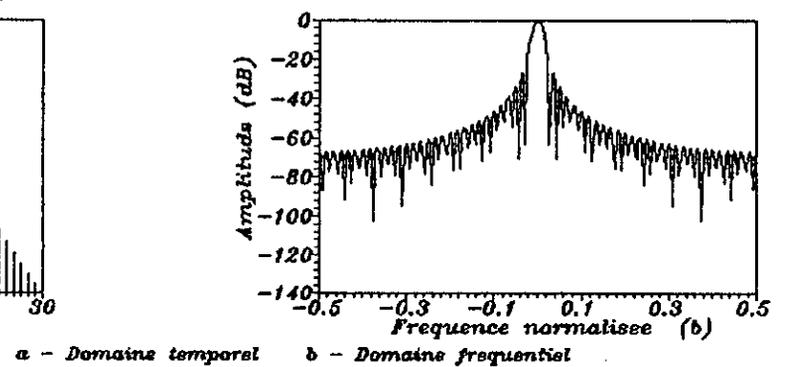
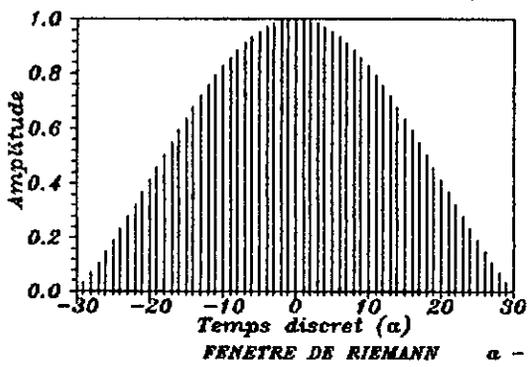
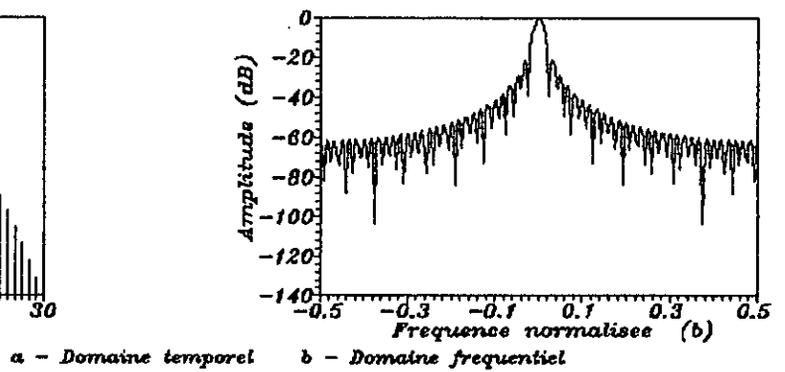
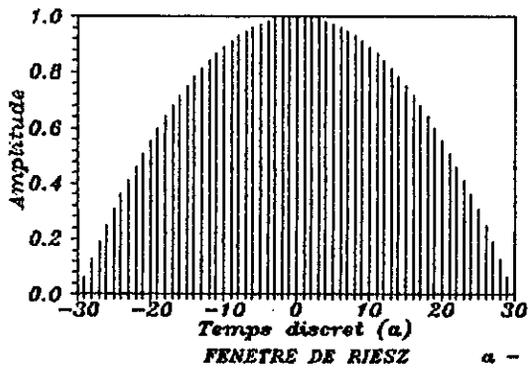
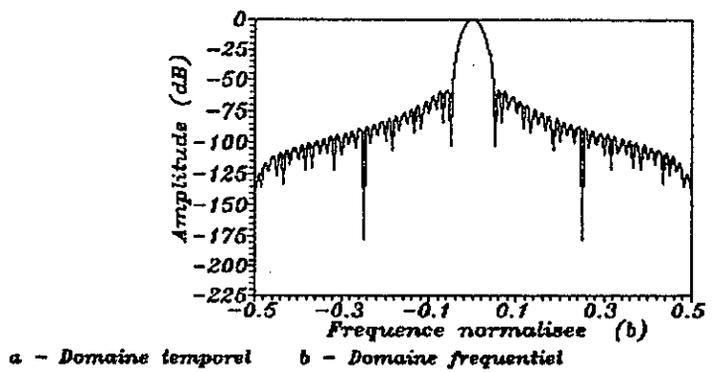
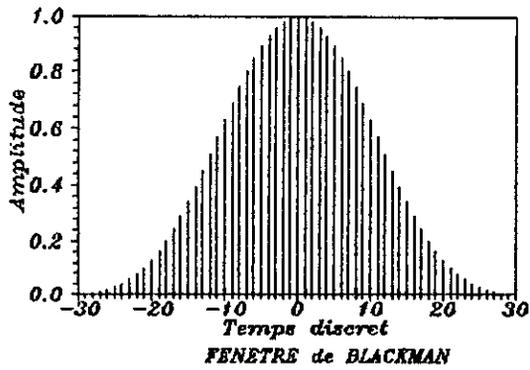
$$w(n) = \begin{cases} 0.5 \cdot \left[1 + \cos\left(\pi \frac{n - \alpha \frac{N-1}{2}}{2(1-\alpha)\frac{N-1}{2}}\right) \right] \exp\left(-\alpha \frac{|n|}{(N-1)/2}\right) & 0 \leq |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{A1.9})$$

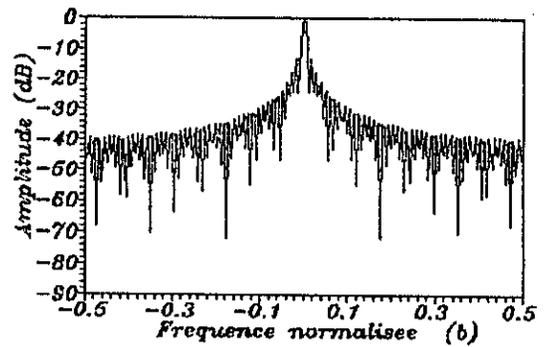
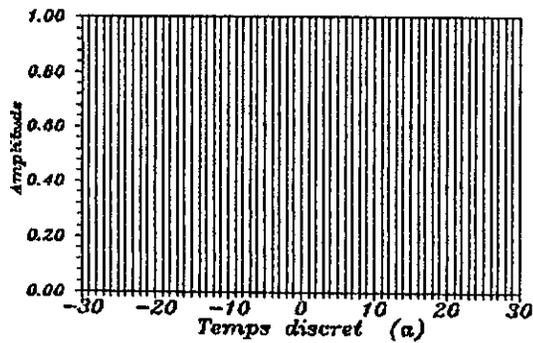
9 - Fenêtre de Cauchy :

$$w(n) = \begin{cases} \frac{1}{1 + \left(\alpha \frac{n}{(N-1)/2}\right)^2} & 0 \leq |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{A1.10})$$

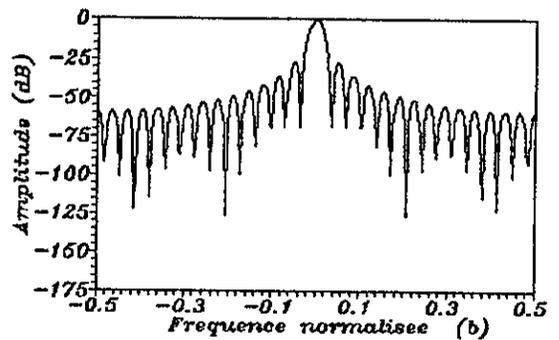
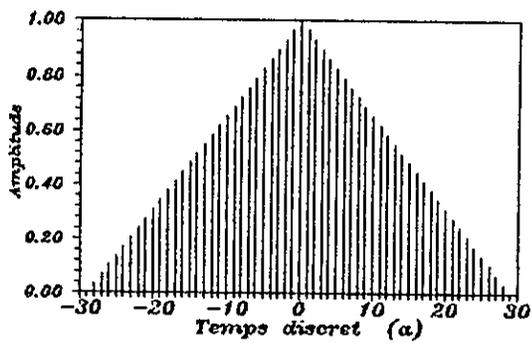
10 - Fenêtre de Gauss (Weiersrass):

$$w(n) = \begin{cases} \exp\left(-\frac{1}{2}\left(\alpha \frac{n}{(N-1)/2}\right)^2\right) & 0 \leq |n| \leq \frac{N-1}{2} \\ 0 & \text{ailleurs} \end{cases} \quad (\text{A1.11})$$

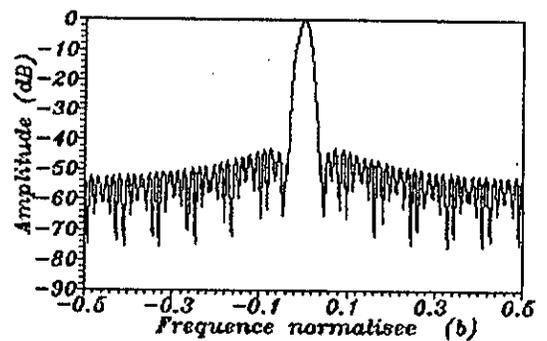
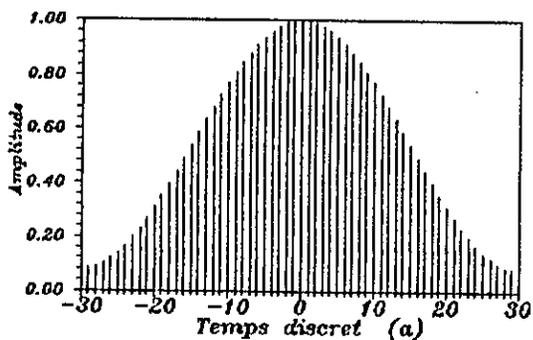




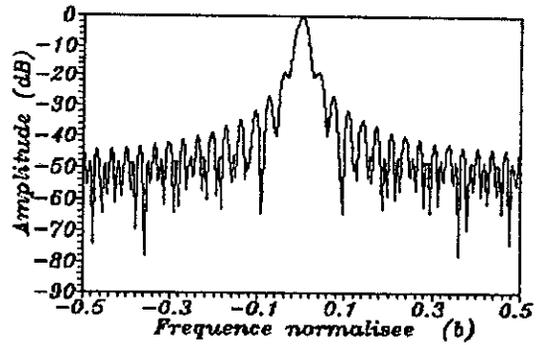
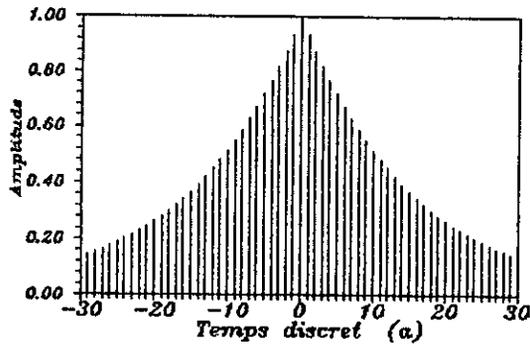
FENETRE RECTANGULAIRE a-Domaine temporel b-Domaine frequentiel



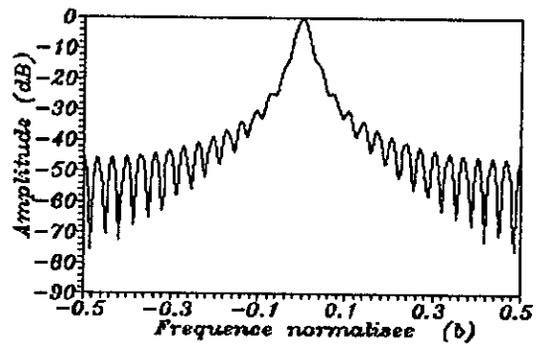
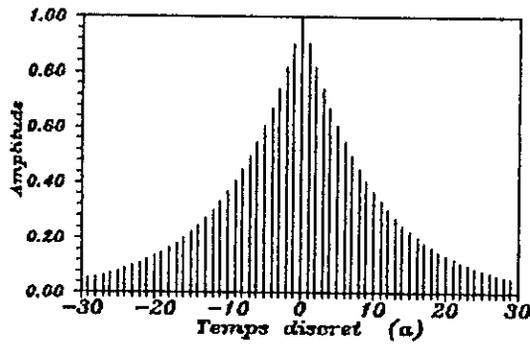
FENETRE TRIANGULAIRE a-Domaine temporel b-Domaine frequentiel



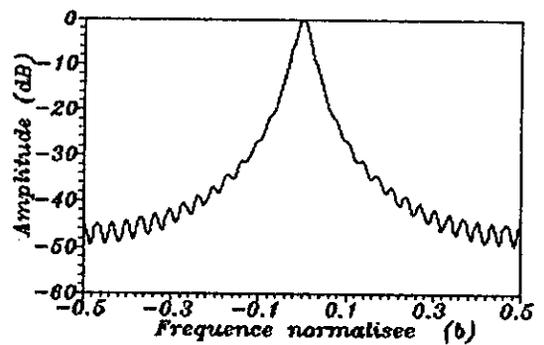
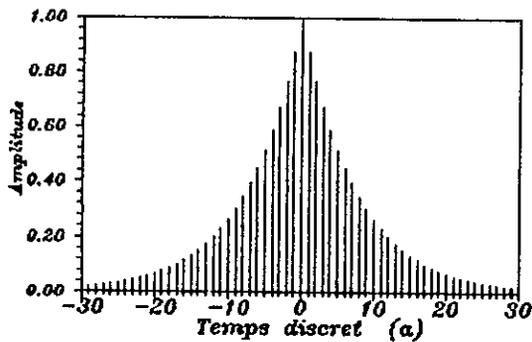
FENETRE DE HAMMING a-Domaine temporel b-Domaine frequentiel



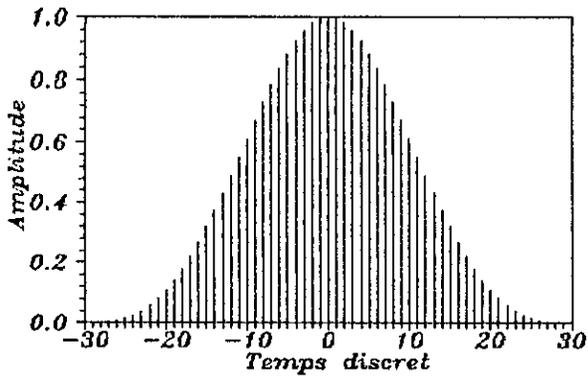
FENETRE DE POISSON $\alpha=2$ a-Domaine temporel b-Domaine frequentiel



FENETRE DE POISSON $\alpha=3$ a-Domaine temporel b-Domaine frequentiel

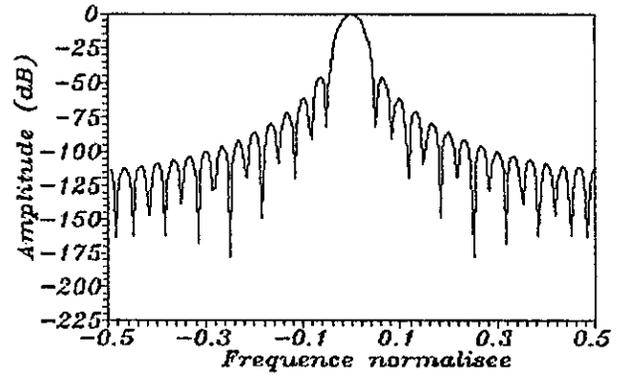


FENETRE DE POISSON $\alpha=4$ a-Domaine temporel b-Domaine frequentiel

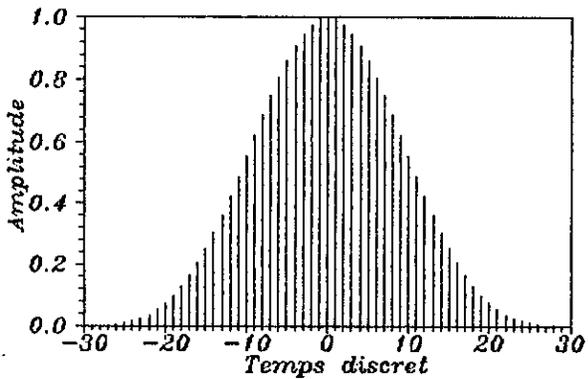


FENETRE de BOHMAN

a - Domaine temporel

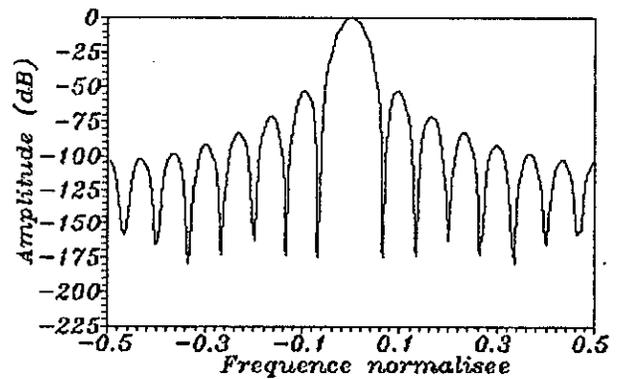


b - Domaine frequenciel

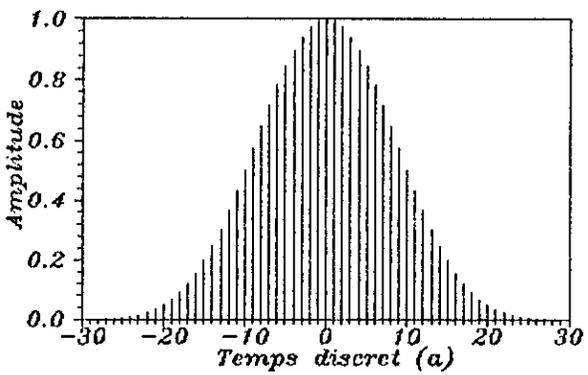


FENETRE de DE-LA-VALLE-POUSSIN

a - Domaine temporel

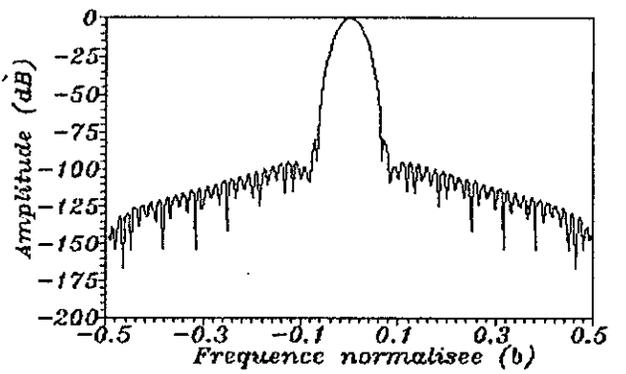


b - Domaine frequenciel

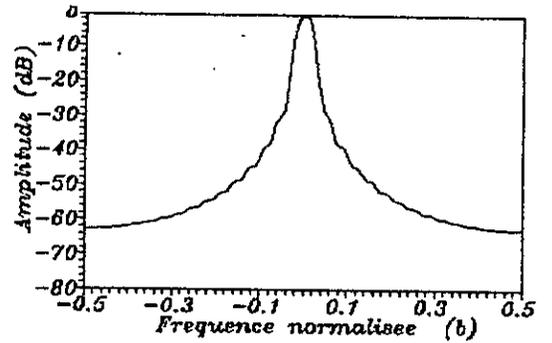
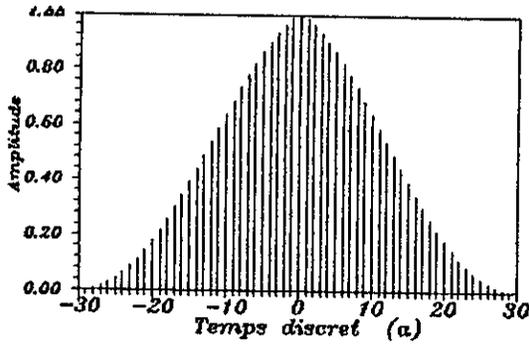


FENETRE de BLACKMAN-HARRIS (alpha=4)

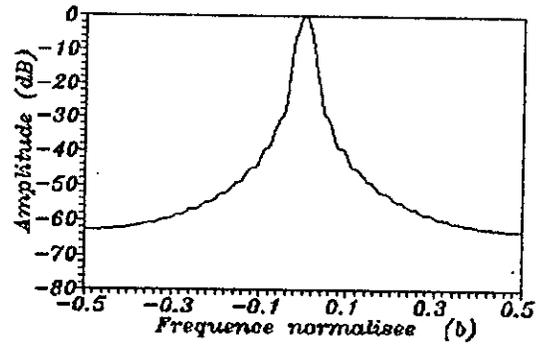
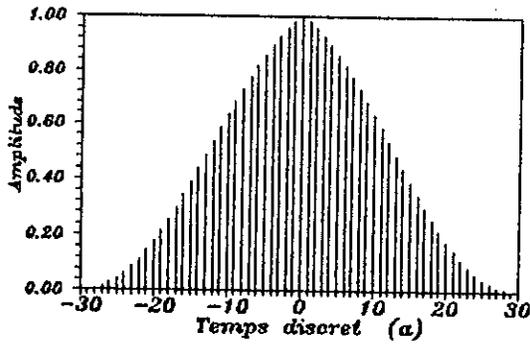
a - Domaine temporel



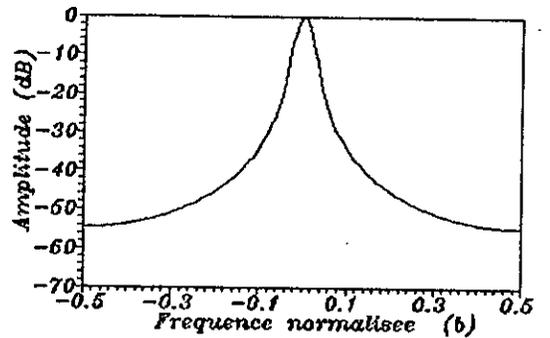
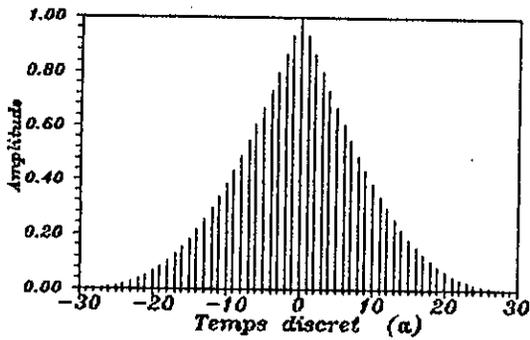
b - Domaine frequenciel



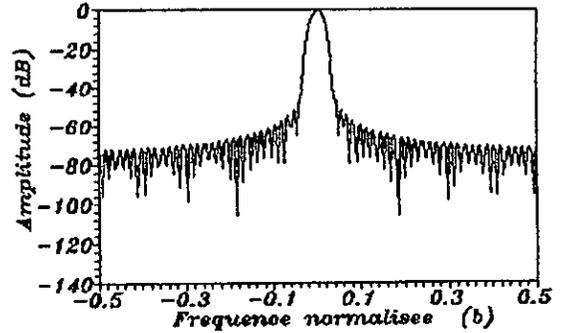
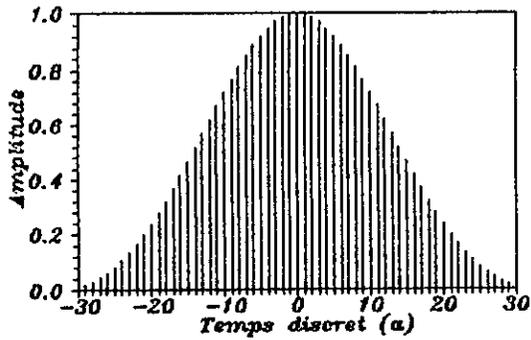
FENETRE DE HANNING-POISSON $\alpha=0.5$ a-Domaine temporel b-Domaine frequentiel



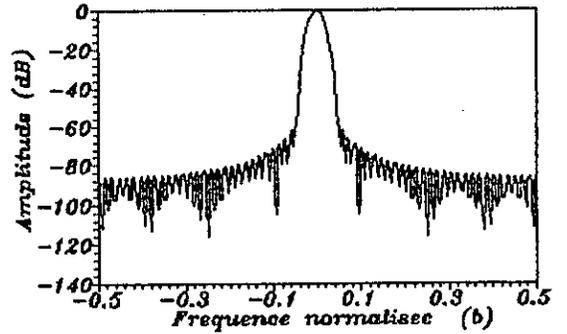
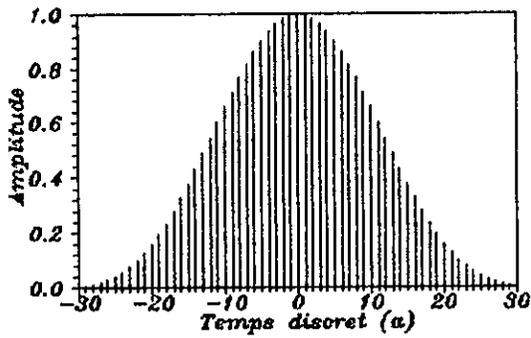
FENETRE DE HANNING-POISSON $\alpha=1$ a-Domaine temporel b-Domaine frequentiel



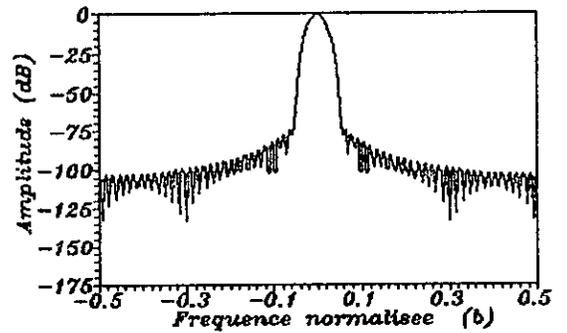
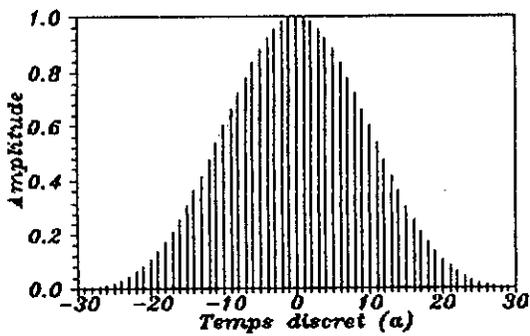
FENETRE DE HANNING-POISSON $\alpha=2$ a-Domaine temporel b-Domaine frequentiel



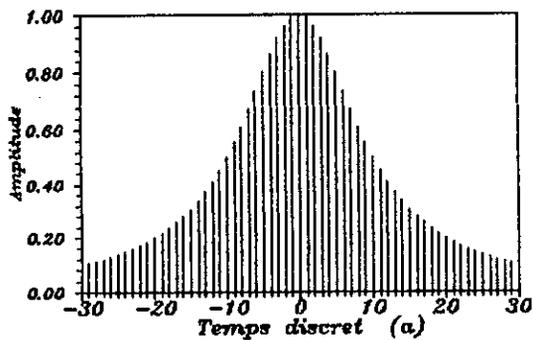
FENETRE DE KAISER $\alpha=2*\pi$ a-Domaine temporel b-Domaine frequentiel



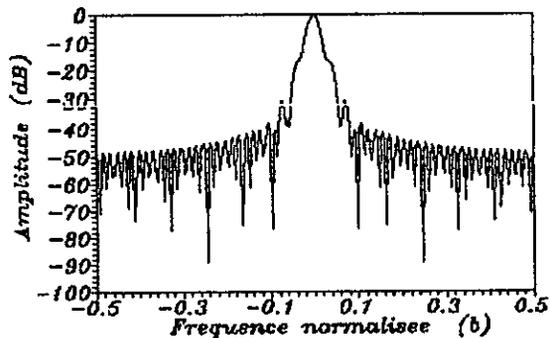
FENETRE DE KAISER $\alpha=2.5*\pi$ a-Domaine temporel b-Domaine frequentiel



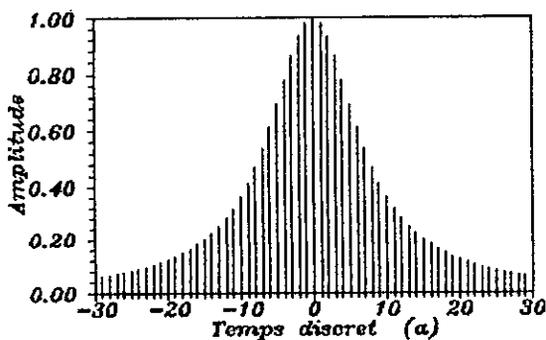
FENETRE DE KAISER $\alpha=3*\pi$ a-Domaine temporel b-Domaine frequentiel



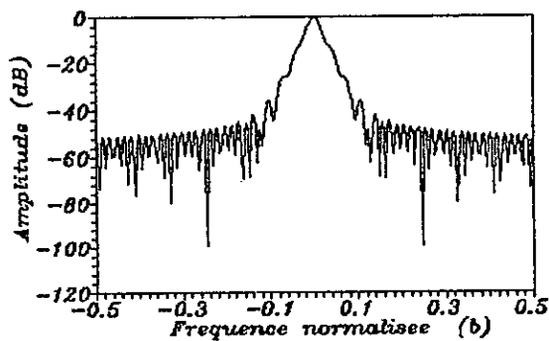
FENETRE DE CAUCHY $\alpha=3$



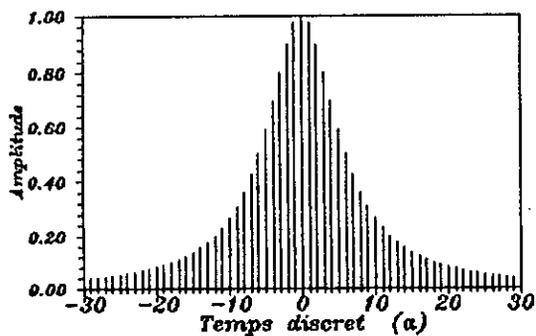
a-Domaine temporel b-Domaine frequentiel



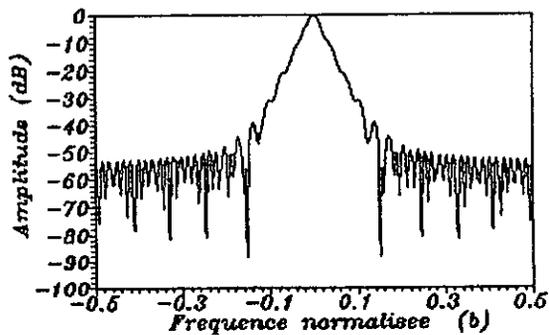
FENETRE DE CAUCHY $\alpha=4$



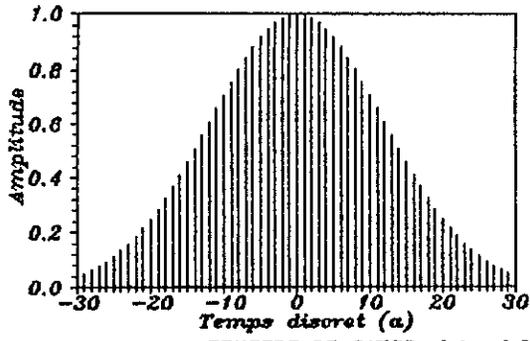
a-Domaine temporel b-Domaine frequentiel



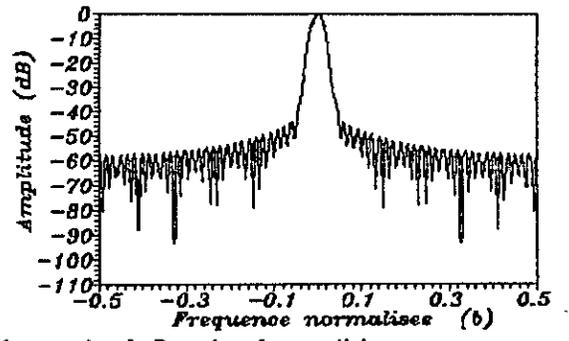
FENETRE DE CAUCHY $\alpha=5$



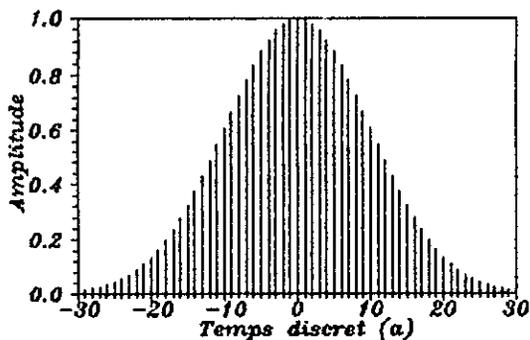
a-Domaine temporel b-Domaine frequentiel



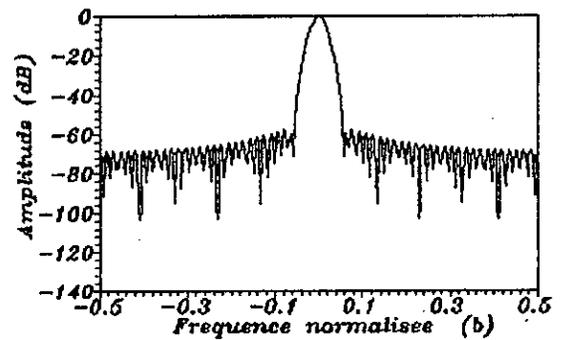
FENETRE DE GAUSS alpha=2.5



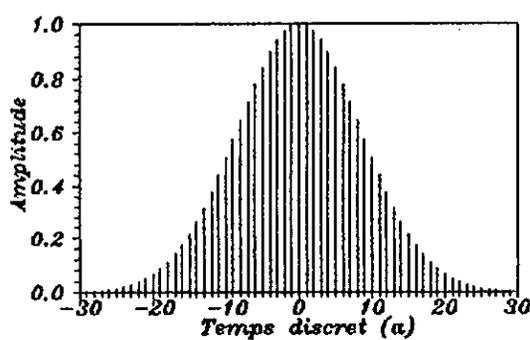
a-Domaine temporel b-Domaine frequenciel



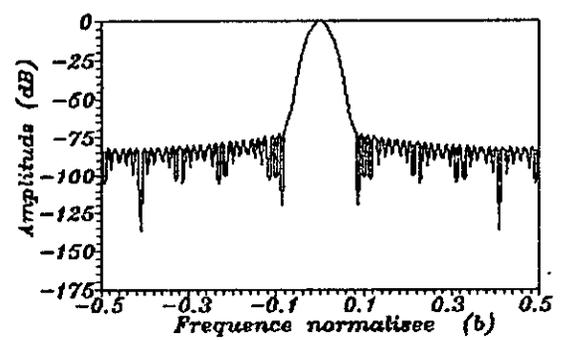
FENETRE DE GAUSS alpha=3



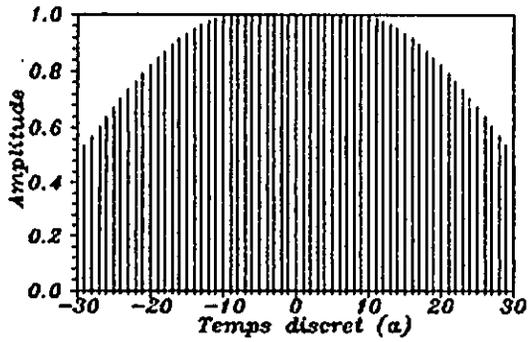
a-Domaine temporel b-Domaine frequenciel



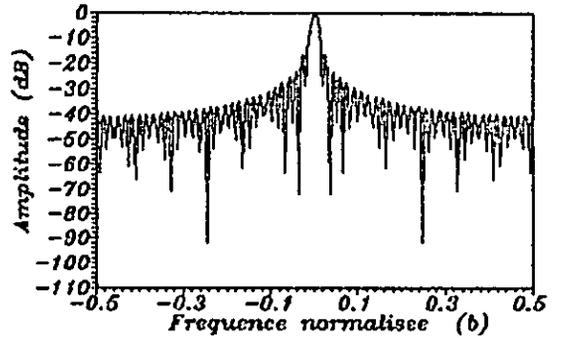
FENETRE DE GAUSS alpha=3.5



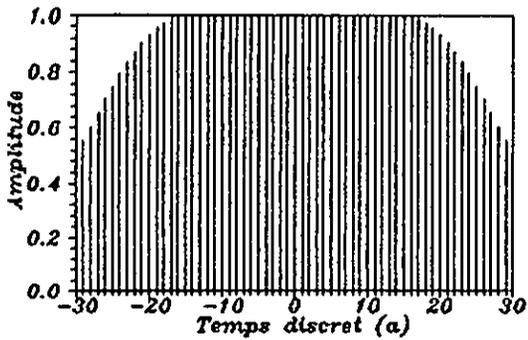
a-Domaine temporel b-Domaine frequenciel



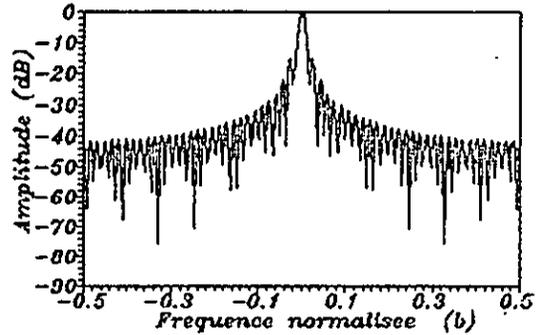
FENETRE DE TUKEY $\alpha=0.25$



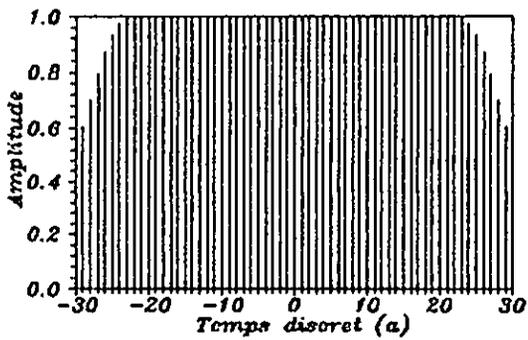
a -Domaine temporel b-Domaine frequentiel



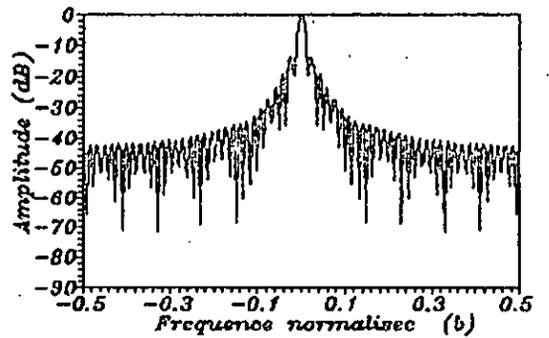
FENETRE DE TUKEY $\alpha=0.5$



a -Domaine temporel b-Domaine frequentiel



FENETRE DE TUKEY $\alpha=0.75$



a -Domaine temporel b-Domaine frequentiel

ANNEXE 2

ALGORITHME DE LA TRANSFORMEE DE FOURIER RAPIDE (FFT)

La transformée de Fourier discrète, définie dans le chapitre I, joue un rôle important dans l'analyse, la synthèse et la réalisation des filtres numériques.

La transformée de Fourier rapide FFT (Fast Fourier Transform) est un algorithme d'évaluation de la DFT (discret Fourier Transform). Son avantage principal est un temps de calcul réduit relativement à l'évaluation directe de la DFT.

Rappelons que la DFT d'une suite $x(n)$, $n = 0, N-1$ s'écrit :

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{-nk} \quad k = 0, N-1 \quad (\text{A2.1})$$

$$\text{où } W_N^{-nk} = e^{-j \frac{2\pi}{N} nk}$$

$$\text{et } x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{nk} \quad (\text{A2.2})$$

est la transformée de Fourier discrète inverse de $X(k)$ (IDFT)

On peut voir que W_N^{nk} est une fonction périodique de période N .

Dès lors, on peut écrire :

$$W_N^{nk} = W_N^{(n+mN)(k+lN)} \quad (\text{A2.3})$$

L'équation (A2.1) peut s'écrire sous une forme matricielle (voir chapitre I, relation I.17a).

Le nombre d'opérations à effectuer s'évalue à N^2 multiplications complexes et $N(N-1)$ additions complexes.

Lorsque le nombre N est élevé, le nombre d'opérations nécessaires pour le calcul direct de la DFT devient très important, entraînant des temps de mise oeuvre prohibitifs. Les algorithmes de la FFT

exploitent les propriétés de W_N^{nk} (périodicité) pour réduire le nombre d'opérations, e par la même occasion le temps de calcul.

L'algorithme que nous avons utilisé est basé sur la décomposition de l'ensemble $x(n)$, en première étape, en deux sous ensemble selon la parité du paramètre n .

(N est supposé une puissance de 2).

La première étape se fait selon le graphe de fluence de la figure ci-dessous :

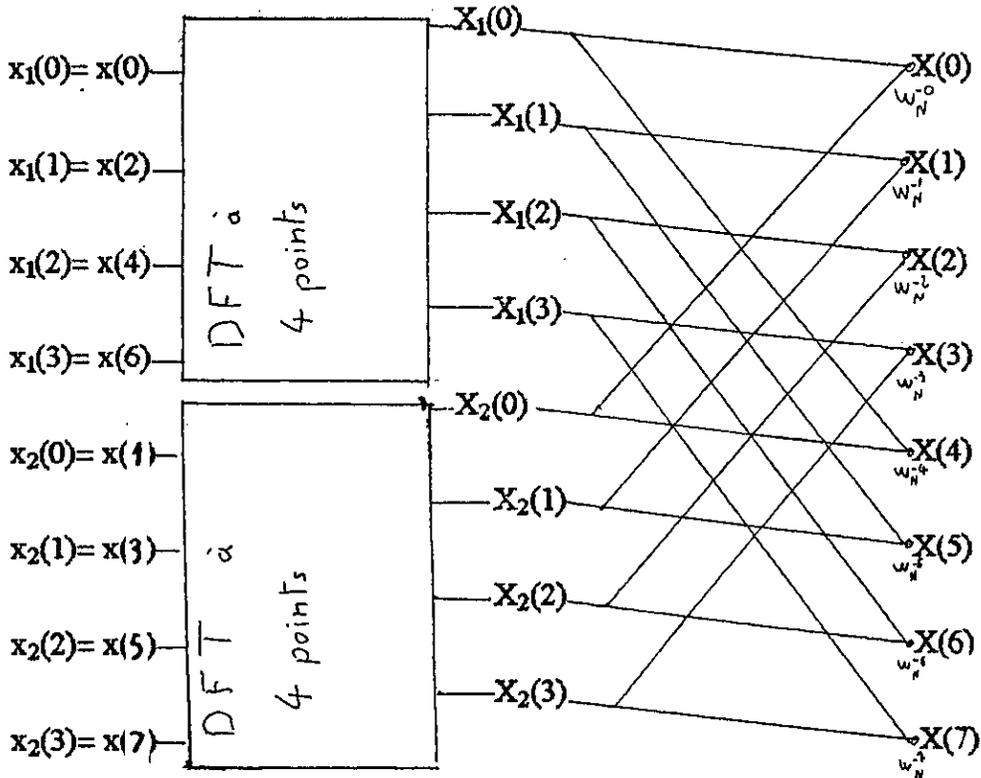


Figure A2.1 : Graphe de fluence de la décomposition d'une DFT à 8 points en deux DFT à 4 points

En seconde étape, les DFT à 4 points sont décomposées chacune en 2 DFT à deux points.

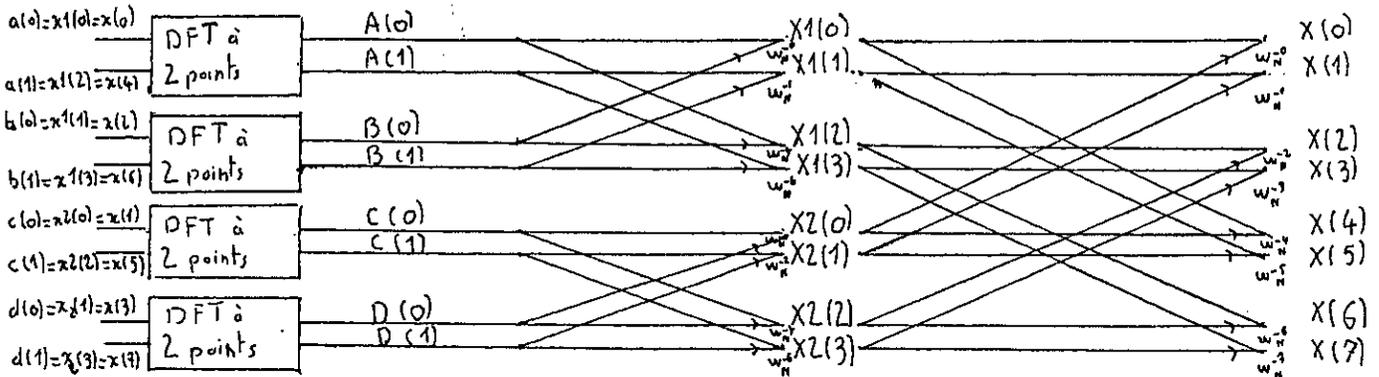


Figure A2.2 : Graphe de fluence de la décomposition d'une D.F.T à 8 points en 4 DFT à 2 points. Le graphe de fluence total est donné par la figure ci-dessous :

Nous pouvons alors écrire :

$$\begin{aligned}
 X(k) &= \sum_{n \text{ pair}} x(n) W_N^{-nk} + \sum_{n \text{ impair}} x(n) W_N^{-nk} \\
 &= \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{-2nk} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{-(2n+1)k}
 \end{aligned} \tag{A2.4}$$

En posant : $x(2n) = x_1(n)$
 $x(2n+1) = x_2(n)$

il vient :

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{\frac{N}{2}-1} x_1(n) W_{N/2}^{-nk} + W_N^{-k} \sum_{n=0}^{\frac{N}{2}-1} x_2(n) W_{N/2}^{-nk} \\
 &= X_1(k) + W_N^{-k} X_2(k)
 \end{aligned} \tag{A2.5}$$

Après cette première étape, de décomposition, on réduit le nombre de multiplications complexes de

$$N^2 \text{ à } \frac{N^2}{2} + N.$$

Etant donnée que $X_1(k)$, $X_2(k)$ sont de période $N/2$, $X(k)$ s'écrit :

$$X(k) = \begin{cases} X_1(k) + W_N^{-k} X_2(k) & 0 \leq k \leq \frac{N}{2} - 1 \\ X_1(-\frac{N}{2} + k) + W_N^{-k} X_2(-\frac{N}{2} + k) & \frac{N}{2} \leq k \leq N - 1 \end{cases} \tag{A2.6}$$

En seconde étape, on décompose $x_1(n)$ et $x_2(n)$ en sous ensembles selon la parité de n , on aboutit alors à des DFT à $N/4$ points. Si $N = 2^v$, on refait la même opération v fois (v étapes). chaque étape nécessite N multiplications complexes, soit un total $Nv = N \log_2 N$ multiplications

complexes, qui peut être réduit d'avantage à $\frac{N}{2} \log_2 N$.

Lorsque N est faible, la différence entre N^2 et $N \log_2 N$ n'est pas très grande, cependant, quand N augmente, elle devient de plus en plus importante [17].

Pour illustrer les étapes de décomposition du signal, on prend l'exemple d'un signal $x(n)$ défini dans l'intervalle $[0,7]$, soit $N = 8$ et $v = 3$.

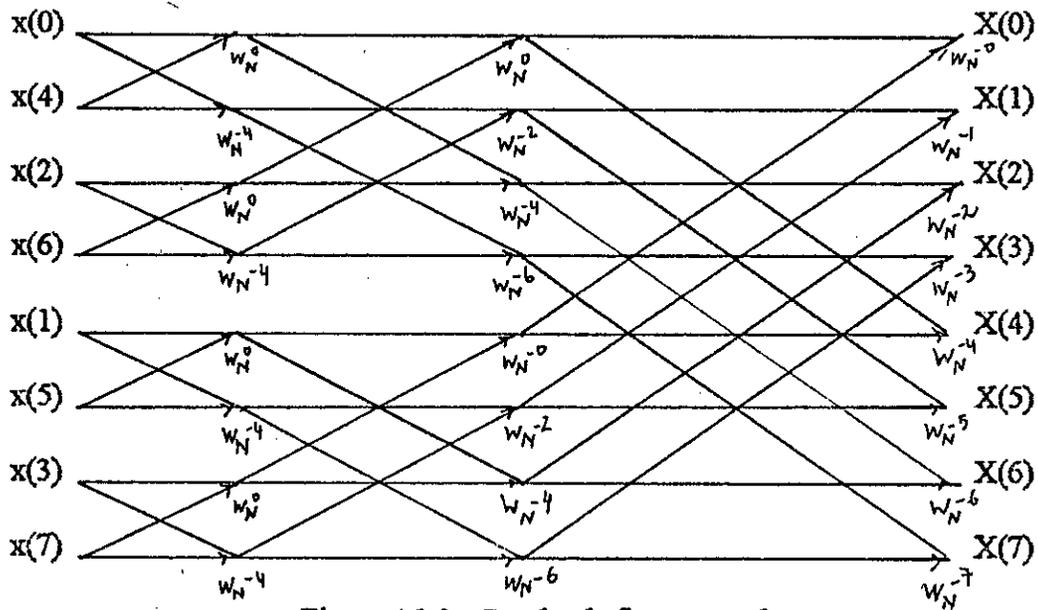
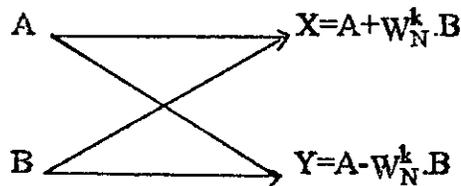


Figure A2.3 : Graphe de fluence total.

Il est à remarquer que ce diagramme de fluence est formé de $\frac{N}{2}$ cellules de la forme :



Pour pouvoir récupérer un signal $X(k)$ ordonné par ordre croissant de k , il est nécessaire de permuter entre certains coefficients de $x(n)$. Cette perturbation est en fait une inversion des bits du paramètre n .

Pour l'exemple considéré, on peut écrire [8] :

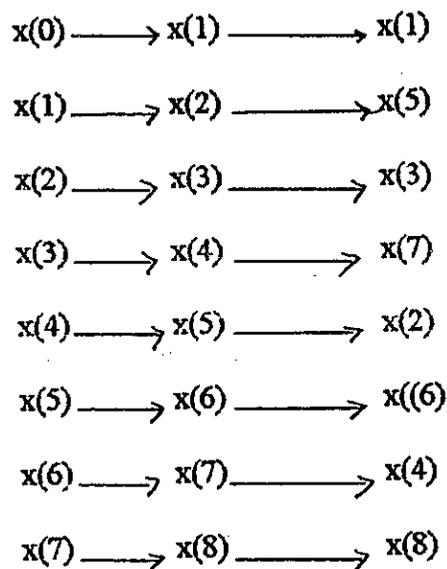
indice avant inversion	représentation binaire	représentation binaire	indice après inversion
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

La subroutine F.F.T que nous avons utilisé, a été élaborée par Cooley, Lewis et Welch. Elle consiste en :

- Choix du mode direct ou inverse (D.F.T ou I.D.F.T).
- Lecture de N ou v .
- Lecture des coefficients du signal à traiter.
- Permutation des coefficients.
- Calcul des différentes cellules (En papillon).

Il faut noter que ce programme travaille sur des signaux $x(n)$ dont l'indice n varie de 1 à N (avec $N=2^v$) et dont la valeur réelle est définie pour des indices appartenant à l'intervalle $[0, N-1]$. De même le signal de sortie $X(k)$ défini pour $k=0, N-1$ est décalée à droite d'une position.

Pour se fixer les idées, prenons le cas où $v = 3$. Le signal $x(0).....x(N-1)$ est représenté par $x(1).....x(N)$. Ainsi :



Pour plus de détails sur les différentes algorithmes de la F.F.T, nous orientons le lecteur vers des références que voici [8], [9], [17]

Annexe 3

PROGRAM FILTRES RIF

C Ce programme permet la synthèse des filtres numériques RIF à phase
C linéaire . En première étape, les fonctions fenêtrés temporelles
C sont évaluées. En seconde étape, on calcule la réponse impulsion-
C nelle du filtre désiré, puis on déduit les coefficients du filtre
C à synthétiser.

C Le choix du filtre à synthétiser, de la fenêtré à utiliser et le
C calcul des transformées de Fourier discrètes se font par des
C sousroutines .

C Paramètres d'entrée:

C
C =====
C CN=1====> choix de l'ordre selon les puissances des filtres
C CN=2====> choix personnel de l'ordre
C CN=3====> choix de l'ordre selon un gabarit
C CHOIX1=1====> choix du filtre passe bas
C CHOIX1=2====> choix du filtre passe haut
C CHOIX1=3====> choix du filtre passe bandes avec passe bas
C CHOIX1=4====> choix du filtre passe bandes sans passe bas
C CHOIX1=5====> choix du filtre coupe bande
C CHOIX1=6====> choix du filtre différentiateur
C CHOIX1=7====> choix du filtre de mise en forme
C CHOIX1=8====> choix du transformateur de Hilbert
C CHOIX3=1====> réponse symétrique
C CHOIX3=2====> réponse antisymétrique
C La valeur de CHOIX2 permet de choisir la fenêtré
C Wci,W(i): fréquences de coupure normalisées
C K: nombre de bandes du filtre désiré
C N: ordre du filtre
C ALPHA: paramètre caractérisant certaines fenêtrés

C
C PARAMETRES DE SORTIE:

C
C =====
C Win(I): Vecteur contenant les coefficients de la fenêtré
C H(I) : Vecteur contenant les coefficients de la réponse
C impulsionnelle non fenêtrée
C H1(I) : Vecteur contenant les coefficients de la réponse
C impulsionnelle fenêtrée
C X(I) : Vecteur contenant les coefficients du spectre
C d'amplitude du filtre calculé
C X2(I) : Vecteur contenant les coefficients du spectre
C d'amplitude des fenêtrés

C
C FICHIERS DES RESULTATS:

C
C =====
C 'FENT.DAT': coefficients de la fenêtré choisie
C 'FENF.DAT': coefficients de la transformée de Fourier de la fenêtré
C 'HIMP.DAT': coefficients de la réponse impulsionnelle non fenêtrée
C 'REPI.DAT': coefficients de la réponse impulsionnelle fenêtrée
C 'REFP.DAT': coefficients du spectre d'amplitude du filtre calculé

DECLARATIONS

C
C implicit real*8 (A-H,O-Z)
C real*8 X(300),h1(300),X2(300)
C INTEGER CN,CHOIX1,CHOIX2,CHOIX3,REP,PAS,si,sig
C COMPLEX*16 h1c(300),WINC1(300),h2c(300),WINC2(300)
C CHARACTER*40 titr1,titr2
C COMMON/B1/ WIN(300)
C COMMON/B2/ H(300)
C COMMON/B3/W(20) ,wc
C PI=4.DO*DATAN(1.DO)

OUVERTURE DES FICHIERS

C
C
C open(1,FILE='FENT.DAT',status='new')

```

OPEN(2,FILE='FENF.DAT',status='new')
OPEN(3,FILE='REPI.DAT',status='new')
OPEN(4,FILE='REPF.DAT',status='new')
OPEN(5,FILE='HIMP.DAT',status='new')
10 WRITE(*,*) 'MENU'
WRITE(*,*) 'ORDRE DU FILTRE:'
WRITE(*,*) '1-CHOIX AUTOMATIQUE'
WRITE(*,*) '2-CHOIX PERSONNEL'
WRITE(*,*) '3-CHOIX SELON GABARIT'
WRITE(*,*) 'DONNER VOTRE CHOIX'
READ(*,*) CM

```

```

MENU
ORDRE DU FILTRE:
1-CHOIX AUTOMATIQUE
2-CHOIX PERSONNEL
3-CHOIX SELON GABARIT

```

```

IF(CM.LT.1.OR.CM.GT.3)THEN

```

```

WRITE(*,*) 'ERREUR'
write(*,*) 'ERREUR'
write(*,*) 'ERREUR'

```

```

ERREUR

```

```

GOTO 10
ENDIF 11

```

```

WRITE(*,*) 'MENU'
WRITE(*,*) '1-FILTRE PASSE BAS'
WRITE(*,*) '2-FILTRE PASSE HAUT'
WRITE(*,*) '3-FILTRE PASSE BANDES AVEC PASSE BAS'
WRITE(*,*) '4-FILTRE PASSE BANDES SANS PASSE BAS'
WRITE(*,*) '5-FILTRE COUPE BANDE'
WRITE(*,*) '6-DIFFERENTIEATEUR'
WRITE(*,*) '7-FILTRE DE MISE EN FORME'
WRITE(*,*) '8-TRANSFORMATEUR DE HILBERT'

```

```

MENU
1-FILTRE PASSE BAS
2-FILTRE PASSE HAUT
3-FILTRE PASSE BANDES AVEC PASSE BAS
4-FILTRE PASSE BANDES SANS PASSE BAS
5-FILTRE COUPE BANDE
6-DIFFERENTIEATEUR
7-FILTRE DE MISE EN FORME
8-TRANSFORMATEUR DE HILBERT

```

```

WRITE(*,*) 'DONNER VOTRE CHOIX'
WRITE(*,*) 'DONNER VOTRE CHOIX'

```

```

READ(*,*) CHOIX1
IF(CHOIX1.LT.1.OR.CHOIX1.GT.8)THEN
WRITE(*,*) 'ERREUR'
GOTO 11
ENDIF

```

```

IF(CHOIX1.EQ.1.OR.CHOIX1.EQ.2)THEN

```

```

K=1
W(1)=0.d0
W(2)=0.d0
CALL LECWC
IF(CHOIX1.EQ.2)THEN
SIG=-1
P=1.D0-WC/PI
titr2='filtre passe haut'
h0=P
ELSE
SIG=1
P=WC/Pi
titr2='filtre passe bas'
h0=wc/pi
ENDIF

```

```

ELSE
IF(CHOIX1.EQ.3.OR.CHOIX1.EQ.4)THEN

```

```

WRITE(*,*) 'INTRODUIRE LE NOMBRE DE BANDES :K'
WRITE(*,*) 'LA BANDE A L'ORIGINE NON COMPRISE'
READ(*,*) K
CALL LECWI(K)
CALL ORD(K)
IF(CHOIX1.EQ.3)THEN
CALL LECWC

```

```

if(w(1).le.wc)then
  write(*,*) '
  write(*,*) '
  write(*,*) '
  goto 201
endif
  titr2='filtre passe bandes avec passe bas'
  ELSE
  WC=0.d0
  titr2='filtre passe bandes sans passe bas'
ENDIF
SIG=1
  s1=0.D0
  S2=0.D0
  DO 12 I=1,K
    S1=S1+W(2*I)
    S2=S2+W(2*I-1)
    P=((S1-S2)/PI)+(WC/PI)
12 CONTINUE
h0=P
ELSE
  IF(CHOIX1.EQ.5)THEN
    K=1
    CALL LECWI(K)
    CALL ORD(K)
    SIG=-1
    WC=0.d0
    P=(W(1)-W(2)/PI)+1.D0
    titr2='filtre coupe bande'
    h0=1+((w(1)-w(2))/pi)
  ELSE
    SIG=0
    IF(CHOIX1.EQ.6)THEN
      CALL LECWC
      P=WC**3/(3*PI)
      titr2='différentiateur'
      h0=0.d0
    ELSE
      IF(CHOIX1.EQ.7)THEN
        K=2
        WC=0.d0
        P=25.D-2
        titr2='filtre de mise en forme'
        h0=1/PI
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
C
C CHOIX DU MODE SYMETRIQUE OU ANTISYMETRIQUE
C
32 WRITE(*,*) '
WRITE(*,*) '
WRITE(*,*) '
WRITE(*,*) '
  titr2='MENU'
  1-REPONSE SYMETRIQUE
  2-REPONSE ANTISYMETRIQUE

WRITE(*,*) '
write(*,*) '
READ(*,*) CHOIX3
IF(CHOIX3.LT.1.OR.CHOIX3.GT.2)THEN
  WRITE(*,*) 'ERREUR'
  GOTO 32
ENDIF
IF(CHOIX3.EQ.1.AND.CHOIX1.EQ.6.OR.CHOIX3.EQ.2.AND.CHOIX1.EQ.8)then

```

```

write(*,*)
write(*,*)
write(*,*)
write(*,*) Le transformateur de Hilbert & le Differentiateur
write(*,*) Sont Antisymetriques
goto 32
endif
IF(CHOIX3.EQ.1)SI=1
IF(CHOIX3.EQ.2)then
SI=-1
h0=0.d0
endif
if(cn.ne.3)then
WRITE(*,*) CHOIX DE LA FENETRE
WRITE(*,*)
WRITE(*,*) 1-RECTANGULAIRE
WRITE(*,*) 2-TRIANGULAIRE
WRITE(*,*) 3-HANNING
WRITE(*,*) 4-HAMMING
WRITE(*,*) 5-BLACKMAN
WRITE(*,*) 6-DE RIETZ
WRITE(*,*) 7-DE RIEMAN
WRITE(*,*) 8-DE DE LA VALLE-POUSSIN
WRITE(*,*) 9-DE TUKEY
WRITE(*,*) 10-DE BOHMAN
WRITE(*,*) 11-DE POISSON
WRITE(*,*) 12-DE HANNING-POISSON
WRITE(*,*) 13-DE CAUCHY
WRITE(*,*) 14-DE GAUSS-WEIERSTRASS
WRITE(*,*) 15-DE KAISER BESSEL
WRITE(*,*) 16-DE BLACKMAN-HARRIS
13 WRITE(*,*) INTRODUIRE UN NUMERO
READ(*,*) CHOIX2
IF(CHOIX2.LT.1.OR.CHOIX2.GT.16)THEN
write(*,*)
write(*,*) ERREUR
write(*,*)
GOTO 13
ENDIF
C INTRODUCTION DES CARACTERISTIQUES DES FENETRES
IF(CHOIX2.EQ.9)THEN
14 WRITE(*,*) 'INTRODUIRE ALPHA '
WRITE(*,*) '<ALPHA<1'
READ(*,*) ALPHA
IF(ALPHA.LE.0.OR.ALPHA.GE.1)THEN
WRITE(*,*) 'ERREUR'
GOTO 14
ENDIF
ELSE
IF(CHOIX2.EQ.11.OR.CHOIX2.EQ.12.OR.CHOIX2.EQ.13.OR.CHOIX2
*.EQ.14.OR.CHOIX2.EQ.15.OR.CHOIX2.EQ.16)THEN
15 WRITE(*,*) 'INTRODUIRE ALPHA '
WRITE(*,*) '>2ALPHA '
read(*,*) alpha
IF(ALPHA.LT.2)THEN
write(*,*)
write(*,*) ERREUR
write(*,*)
GOTO 15
ENDIF
endif
endif
IF(CN.EQ.2)THEN
WRITE(*,*) 'DONNER UN ORDRE IMPAIR DU FILTRE '

```

```

      READ(*,*)N
      ELSE
        WRITE(*,*) ' DONNER LE PAS P(PAIR)'
        read(*,*) pas
        write(*,*) 'donner LA VALEUR MAX DE N'
        read(*,*) nmax
        write(*,*) 'donner l''erreur en pourcentage eop '
        read(*,*) eop
        eop=eop/100.d0
        n=11
      endif
16    s=h0**2
      i=1
      m=(n-1)/2
17    if(choix3.eq.1) call rif (sig,i,k)
      if(choix3.eq.2) call rifa (i,k,choix1)
      CALL FENET(CHOIX2,I,M,ALPHA,TITR1)
      h1(i)=h(i)*win(i)
      s=s+(h1(i)**2)*2
      i=i+1
      if(i.LE.m) GOTO 17
      if(cn.eq.2)GOTO 100
      E=(p-s)/p
      if(e.lt.eop)then
        write(*,*) 'l''ordre du filtre est:'
        write(*,*) n
        goto 100
      else
        n=n+pas
        if(n.lt.nmax)GOTO 16
101    write(*,*) 'vous avez depasse la valeur nmax '
        write(*,*) 'voulez vous? '
        write(*,*) '1-changer nmax '
        write(*,*) '2-changer eop '
        read(*,*) rep
        IF (REP.LT.1.OR.REP.GT.2) THEN
          WRITE(*,*)'ERREUR'
          GOTO 101
        ENDIF
        if (rep.eq.1) then
          write(*,*) 'introduisez la nouvelle valeur NMAX'
          read(*,*) nmax
        else
          write(*,*) 'introduisez la nouvelle valeur de eop'
          read(*,*) eop
          eop=eop/100.d0
          n=11
        endif
        goto 16
      endif
      else
        CALL GABARI(N,H1,TITR1,SIG,K,choix3,choix1)
      endif
      m=(N-1)/2
100    h(1)=h0
      h1(1)=h0
      win(1)=1.D0
      write(5,154) titr2
154    format(14X,'reponse impulsionnelle du'/20X,A30)
      write(1,150) titr1
150    format(14X,'COEFFICIENTS DE LA FENETRE'/20X,A30)
      DO 151 I=1,M+1
        write(1,*) I-1,0
        write(1,152) I-1,WIN(I)
        write(1,*) I-1,0

```

```

        write(5,*) I-1,0
        write(5,153) I-1,H(I)
        write(5,*) I-1,0
        write(3,*) I-1,0
        write(3,155) I-1,H1(I)
        write(3,*) I-1,0
151  continue
        DO 159 I=M+1,2,-1
        write(1,*) -I+1,0
        write(1,152) -I+1,WIN(I)
        write(1,*) -I+1,0
        write(5,*) -I+1,0
        write(5,153) -I+1,H(I)
        write(5,*) -I+1,0
        write(3,*) -I+1,0
        write(3,155) -I+1,H1(I)
        write(3,*) -I+1,0
159  continue
152  format(2X,'win('I4,')=' ,D20.14)
152  format(2X,I4,e20.5)
153  format(2X,I4,2X,E20.14)
155  format(2X,I4,2X,E20.14)
        close(1)
        close(3)
        close(5)
        NU=INT(DLOG(FLOAT(N)+1.DO)/DLOG(2.DO))+1

```

C
C
C

CALCUL DES FFT

```

        N1=2**NU
        DO 30 I=(N+3)/2,N1
        h1(i)=0.d0
        h(i)=0.d0
        win(i)=0.d0
30  continue
        DO 31 I=1,N1
        WINC1(I)=CMPLX(WIN(I),0.DO)
        WINC2(I)=CMPLX(WIN(I),0.DO)
        H1C(I)=CMPLX(H1(I),0.DO)
        H2C(I)=CMPLX(H1(I),0.DO)
31  continue
        mode=0
        CALL FFT(WINC1,NU,MODE)
        CALL FFT(H1C,NU,MODE)
        MODE=-1
        CALL FFT(H2C,NU,MODE)
        CALL FFT(WINC2,NU,MODE)
        write(4,160) titr2
160  format(14X,'module de la reponse frequentielle du '/20x,A40)
        WRITE(4,161)TITR1
161  FORMAT(14X,'utilisant la fenetre ',1X,A30)
        write(2,165) titr1
165  format(14X,'domaine frequentiel de la fenetre',1X,A30)
        DO 33 I=1,N1
        H01=H0*float(si)
        N11=n1*si
        H2C(I)=H2C(I)*cmplx(float(N11),0.DO)
        WINC2(I)=WINC2(I)*CMPLX(FLOAT(N1),0.DO)
        H1C(I)=H1C(I)+H2C(I)-cmplx(h01,0.DO)
        WINC1(I)=WINC1(I)+WINC2(I)-CMPLX(WIN(1),0.DO)
        X(I)=cdabs(H1C(I))
        X2(I)=cdabs(WINC1(I))
33  continue
        AX=X(1)
        AX2=x2(1)

```

```

DO 333 I=1,N1
  IF(X(I).LE.AX) GOTO 333
  AX=X(I)
333  continue
DO 3333 I=1,N1
  IF(x2(I).LE.AX2) GOTO 3333
  AX2=X2(I)
3333  continue
      do 162 i=1,N1
        X(I)=X(I)/AX
        X2(I)=X2(i)/AX2
IF(choix3.eq.1) X(I)=20*DLOG(X(I))/DLOG(10.DO)
IF(choix3.eq.1) X2(i)=20*DLOG(X2(I))/dlog(10.DO)
162  continue
      do 34 i=1,N1/2
        r=(float(i)-1.d0)/N1
        WRITE(4,*) r,x(i)
        write(2,*) r,x2(i)
34  continue
      close(2)
      close(4)
      stop
      END

```

```

C-----
C      SUBROUTINE FENET
C      PERMET LE CHOIX D'UNE FENETRE
C-----

```

```

SUBROUTINE FENET(CHOIX2,I,M,ALPHA,titr1)
implicit real*8 (A-H,O-Z)
common/B1/win(300)
character*40 titr1
integer choix2
if(choix2.eq.1) call rect(i,titr1)
if(choix2.eq.2) call tria(i,m,titr1)
if(choix2.eq.3) call HAMN(I,M,titr1)
if(choix2.eq.4) call HAMM(I,M,titr1)
if(choix2.eq.5) call BLACK(I,M,titr1)
if(choix2.eq.6) call RIEZ(I,M,titr1)
if(choix2.eq.7) call RIEM(I,M,titr1)
if(choix2.eq.8) call DVP(I,M,titr1)
if(choix2.eq.9) call TUK(I,M,ALPHA,titr1)
if(choix2.eq.10) call BOHM(I,M,titr1)
if(choix2.eq.11) call POISS(I,M,ALPHA,titr1)
if(choix2.eq.12) call POHA(I,M,ALPHA,titr1)
if(choix2.eq.13) call CAUCH(I,M,ALPHA,titr1)
if(choix2.eq.14) call GAUS(I,M,ALPHA,titr1)
if(choix2.eq.15) call KAIS(I,M,ALPHA,titr1)
if(choix2.eq.16) call BHA(I,M,ALPHA,titr1)
RETURN
END

```

```

C-----
c      subroutine rif
c      calcul des echantillons des differents filtres
C-----

```

```

subroutine rif(sig,i,k)
implicit real*8 (A-H,O-Z)
integer sig
common/B2/H(300)
common/B3/w(20),wc
PI=4.DO*DATAN(1.DO)
c1=float(i)*pi
if(i.eq.1) then
  fact=(-1+float(k))/4
else
  fact=-(float(k)-1.DO)*dcos(c1/2)/(pi*((float(i)**2)-1.DO))

```

```

endif
C=FLOAT(I)*WC
s2=Dsin(C)/c1
s1=0.D0
do 20 j=1,k
  l=2*j
  m=2*j-1
  wh=FLOAT(i)*w(l)
  wl=FLOAT(i)*w(m)
  s1=s1+(Dsin(wh)-Dsin(wl))/c1
  coef1=(s1+s2)*float(sig)
20  continue
  coef2=fact*(1.D0-iabs(sig))
  h(i+1)=coef1+coef2
  return
end

C-----
C      SUBROUTINE RIFA
C      CALCULE LES ECHANTILLONS DES DIFFERENTS FILTRES -CAS ANTISYMETRIQUE
C-----

subroutine rifa(I,K,CHOIX1)
implicit real*8 (A-H,O-Z)
integer choix1
common/B2/ H(300)
common/B3/ w(20),WC
pi=4.D0*Datn(1.D0)
c1=pi*float(i)
c=WC*float(i)
if(choix1.eq.1) H(I+1)=2*(Dsin(c/2)**2)/c1
write(*,*) h(i+1)
if(choix1.eq.2) H(I+1)=(Dcos(c)-Dcos(c1))/c1
if(choix1.eq.3.or.choix1.eq.4) then
  H(I+1)=0.D0
  do 200 J=1,K
    L=2*J
    M=2*J-1
    WH=float(I)*W(L)
    WL=float(I)*W(M)
    H(I+1)=H(I+1)+((Dcos(WL)-Dcos(WH))/C1)
200  continue
  H(I+1)=H(I+1)+(2.D0*(Dsin(c/2)**2)/c1)
endif
if(choix1.eq.5) H(I+1)=(2.D0+Dcos(I*w(2))-Dcos(I*w(1)))/C1
if(choix1.eq.6) H(I+1)=((-WC*dcos(c))+Dsin(C)/float(I))/c1
if(choix1.eq.7) then
  if(I.eq.1)then
    H(I+1)=1/(2.D0*PI)
  else
    H(I+1)=(float(I)-Dsin(C1/2))/(pi*((float(I)**2)-1))
  endif
endif
if(choix1.eq.8) H(I+1)=(1.D0-dcos(C1))/c1
return
end

C-----
C      SUBROUTINE LECWC
C      PERMET LA LECTURE DE LA FREQUENCE DE COUPURE
C      *- DE LA BANDE A L'ORIGINE OU
C      *- DU FILTRE PASSE HAUT OU
C      *- DU DIFFERENTIEATEUR
C-----

SUBROUTINE LECWC
implicit real*8 (A-H,O-Z)
common/B3/w(20),wc
PI=4.D0*DATAN(1.D0)

```

```

21  WRITE(*,*) 'DONNER LA FREQUECE WC'
    WRITE(*,*) '0<WC<0.5'
    READ(*,*) WC
    IF(WC.LE.0.OR.WC.GE.5.D-1)THEN
        WRITE(*,*) 'ERREUR'
        GOTO 21
    ENDIF
    WC=WC*2*PI
    RETURN
    END

```

```

C-----
C  SUBROUTINE LECWI
C  PERMET LA LECTURE DES FREQUENCES DE COUPURE W(I)
C  *- DU FILTRE PASSE BANDES
C  *- DU FILTRE COUPE BANDE
C  ET VERIFIE QU'ILS SONT TOUS DIFFERENTS
C-----

```

```

SUBROUTINE LECWI(K)
implicit real*8 (A-H,O-Z)
common/b3/ w(20),wc
PI=4.D0*DATAN(1.D0)
k=2*k
22  WRITE(*,*) 'DONNER LES FREQUENCES DE COUPURE W(I) '
    WRITE(*,*) 'DANS L'ORDRE CROISSANT '
    WRITE(*,*) '0<W(I)<5.D-1'
    DO 24 I=1,K
        READ(*,*) W(I)
        IF(W(I).LE.0.OR.W(I).GE.5.D-1)THEN
            WRITE(*,*) 'ERREUR'
            GOTO 22
        ENDIF
        DO 23 J=1,I-1
            IF(W(I).EQ.W(J))THEN
                WRITE(*,*) 'ERREUR'
                GOTO 22
            ENDIF
        CONTINUE
        W(I)=W(I)*2*pi
        write(*,*) w(i)
    CONTINUE
    K=K/2
    RETURN
    END

```

```

C-----
C  SUBROUTINE ORD
C  ORDONNE LES FREQUENCES DE COUPURE DANS L'ORDRE CROISSANT
C-----

```

```

SUBROUTINE ORD(K)
implicit real*8 (A-H,O-Z)
common/B3/ w(20),wc
K=2*K
DO 25 I=1,K-1
    DO 25 J=I+1,K
        IF(W(J).LT.W(I))THEN
            P=W(J)
            W(J)=W(I)
            W(I)=P
        ENDIF
    CONTINUE
    K=K/2
    RETURN
    END

```

```

C-----
C  SUBROUTINE GABARI
C  CALCULE LES COEFFICIENTS DES FILTRES A PARTIR D'UN GABARIT

```

C

```

SUBROUTINE GABARI(N,H1,TITR1,SIG,K,choix3,choix1)
implicit real*8 (a-h,o-z)
integer sig,choix2,choix1,choix3
character*40 titr1
real*8 H1(300)
common/b1/ win(300)
common/B2/ h(300)
common/B3/w(20),wc
pi=4.D0*DATAN(1.D0)
write(*,*) 'attenuation dB negative'
read(*,*) Aa
write(*,*) 'Rapport entre la bande de transision et'
write(*,*) 'la bande passante(coupée) du filtre (en %)'
read(*,*) R
R=R/100.D0
if(choix1.EQ.1) B=WC
if(choix1.EQ.2) B=PI-WC
if(choix1.EQ.3) B=min(WC,W(2)-W(1))
if(choix1.EQ.4.OR.choix1.EQ.5) B=W(2)-W(1)
DW=B*R
if(Aa.GT.-21.D0) then
    N=4*int(PI/dw)
    choix2=1
else
if(Aa.LE.-21.D0.AND.Aa.GT.-25.D0) THEN
    N=8*int(PI/DW)
    CHOIX2=2
ELSE
IF(Aa.LT.-25.D0.AND.Aa.GT.-44.D0) THEN
    N=8*int(PI/DW)
    CHOIX2=3
ELSE
IF(Aa.LT.-44.D0.AND.Aa.GT.-53.D0) THEN
    N=8*int(PI/DW)
    CHOIX2=4
ELSE
IF(Aa.LT.-53.D0.AND.Aa.GT.-74.D0) THEN
    N=12*int(PI/DW)
    CHOIX2=5
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
    if(int(n/2).eq.int((n+1)/2)) n=n+1
    M=(N-1)/2
    do 400 i=1,M
        call fenet(choix2,I,M,0.d0,titr1)
        if(choix3.eq.1)call rif(sig,i,k)
        if(choix3.eq.2)call rifa(i,k,choix1)
400    h1(i)=h(i)*win(i)
        continue
    write(*,*) 'la fenetre choisie est :'
    write(*,*) titr1
    write(*,*) 'N=',N
    return
end

```

C

C

C

C

```

SUBROUTINE RECT
CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE RECTANGULAIRE

```

```

SUBROUTINE RECT(I,titr1)
implicit real*8 (A-H,O-Z)
character*40 titr1

```

```
common/B1/ WIN(300)
WIN(I+1)=1.DO
titr1='rectangulaire'
RETURN
END
```

```
C-----
C      SUBROUTINE TRIA
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE TRIANGULAIRE
C-----
```

```
SUBROUTINE TRIA(I,M,titr1)
  implicit real*8 (A-H,O-Z)
  character*40 titr1
  common/B1/ WIN(300)
  WIN(i+1)=1.DO-(FLOAT(I)/FLOAT(M))
  titr1='triangulaire'
  RETURN
  END
```

```
C-----
C      SUBROUTINE HANN
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE HANNING
C-----
```

```
SUBROUTINE HANN(I,M,titr1)
  implicit real*8 (A-H,O-Z)
  character*40 titr1
  common/B1/ WIN(300)
  PI=4.DO*DATAN(1.DO)
  WIN(I+1)=50.D-2+(5.D-1*DCOS(PI*FLOAT(I)/FLOAT(M)))
  titr1='de HANNING'
  RETURN
  END
```

```
C-----
C      SUBROUTINE HAMM
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE HAMMING
C-----
```

```
SUBROUTINE HAMM(I,M,titr1)
  implicit real*8 (A-H,O-Z)
  character*40 titr1
  common/B1/ WIN(300)
  PI=4.DO*DATAN(1.DO)
  WIN(I+1)=54.D-2+(46.D-2*DCOS(PI*FLOAT(I)/FLOAT(M)))
  titr1='DE HAMMING'
  RETURN
  END
```

```
C-----
C      SUBROUTINE BLACK
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE BLACKMAN
C-----
```

```
SUBROUTINE BLACK(I,M,titr1)
  implicit real*8 (A-H,O-Z)
  character*40 titr1
  common/B1/ WIN(300)
  PI=4.DO*DATAN(1.DO)
  WIN(I+1)=42.D-2+(5.D-1*DCOS(PI*FLOAT(I)/FLOAT(M)))+(8.D-2*
*DCOS(2.DO*PI*FLOAT(I)/FLOAT(M)))
  titr1='DE BLACKMAN'
  RETURN
  END
```

```
C-----
C      SUBROUTINE BHA
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE BLACKMAN-HARRIS
C-----
```

```
SUBROUTINE BHA(I,M,alpha,titr1)
  implicit real*8 (A-H,O-Z)
  DIMENSION B(20),A(20)
  character*40 titr1
```

```

common/B1/ WIN(300)
PI=4.DO*DATAN(1.DO)
c   C=(dexp(pi*alpha)-dexp(-pi*alpha))/2*Pi*alpha
    c=dsinh(pi*alpha)/(pi*alpha)
    B1=C
    nalp=int(alpha)
    do 1000 j=1,nalp-1
    R=Dsqr(alpha**2-j**2)
    B(j)=dsinh(pi*R)/(pi*R)
    C=C+2*B(J)
1000 continue

    A1=B1/C
    do 1001 J=1,nalp
    A(J)=2*B(J)/C
1001 continue
WIN(I+1)=A1
do 1002 J=1,nalp
WIN(I+1)=Win(I+1)+A(J)*Dcos(Pi*float(I)*float(J)/float(M))
1002 continue
titr1='DE BLACKMAN-HARRIS'
RETURN
END

```

```

C-----
C   SUBROUTINE DE RIETZ
C   CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE RIEZ
C-----

```

```

SUBROUTINE RIEZ(I,M,titr1)
implicit real*8 (A-H,O-Z)
character*40 titr1
common/B1/ WIN(300)
WIN(I+1)=1.DO-(FLOAT(I)/FLOAT(M))**2
titr1='DE RIEZ'
RETURN
END

```

```

C-----
C   SUBROUTINE RIEM
C   CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE RIEMAN
C-----

```

```

SUBROUTINE RIEM(I,M,titr1)
implicit real*8 (A-H,O-Z)
character*40 titr1
common/B1/ WIN(300)
PI=4.DO*DATAN(1.DO)
WIN(I+1)=DSIN(PI*FLOAT(I)/FLOAT(M))/(PI*FLOAT(I)/FLOAT(M))
titr1='DE RIEMAN'
RETURN
END

```

```

C-----
C   SUBROUTINE DVP
C   CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE DE LA VALLE POUSSIN
C-----

```

```

SUBROUTINE DVP(I,M,titr1)
implicit real*8 (A-H,O-Z)
character*40 titr1
common/B1/ WIN(300)
M1=INT(M/2)+1
IF (I.GE.O.AND.I.LT.M1) THEN
WIN(I+1)=1.DO-6*((FLOAT(I)/FLOAT(M))**2)*(1-FLOAT(I)/FLOAT(M))
ELSE
WIN(I+1)=2.DO*(1-FLOAT(I)/FLOAT(M))**3
ENDIF
titr1='DE DE LA VALLE POUSSIN'
RETURN
END

```

```

C-----
C      SUBROUTINE DE TUKEY
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE TUCKEY
C-----
      SUBROUTINE TUK(I,M,ALPHA,titr1)
      implicit real*8 (A-H,O-Z)
      character*40 titr1
      common/B1/ WIN(300)
      PI=4.DO*DATAN(1.DO)
      OALPHA=FLOAT(M)*ALPHA
      IF (I.GE.O.AND.I.LE.OALPHA) THEN
        WIN(I+1)=1.DO
      ELSE
        WIN(I+1)=5.D-1*(1.DO+DCOS(PI*(I-OALPHA)/(2.DO*(1.d0-ALPHA)*
*FLOAT(M))))
      ENDIF
      titr1='DE TUKEY'
      RETURN
      END
C-----
C      subroutine DE BOHMAN
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE BOHMAN
C-----
      SUBROUTINE BOHM(I,M,titr1)
      implicit real*8 (A-H,O-Z)
      character*40 titr1
      common/B1/ WIN(300)
      PI=4.DO*DATAN(1.DO)
      WIN(I+1)=(1-(FLOAT(I)/FLOAT(M)))*DCOS(PI*FLOAT(I)/FLOAT(M))+
*(1/PI)*DSIN(PI*FLOAT(I)/FLOAT(M))
      titr1='DE BOHMAN'
      RETURN
      END
C-----
C      SUBROUTINE POISSON
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE POISSON
C-----
      SUBROUTINE POISS(I,M,ALPHA,titr1)
      implicit real*8 (A-H,O-Z)
      character*40 titr1
      common/B1/ WIN(300)
      WIN(I+1)=dEXP(-alpha*float(i)/float(m))
      titr1='DE POISSON'
      RETURN
      END
C-----
C      SUBROUTINE HANNING-POISSON
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE HANNING-POISSON
C-----
      SUBROUTINE POha(I,M,ALPHA,titr1)
      implicit real*8(A-H,O-Z)
      character*40 titr1
      common/B1/ WIN(300)
      PI=4.DO*DATAN(1.DO)
      wi=pi*float(I)/float(m)
      WIN(I+1)=(dEXP(-alpha*float(i)/float(m)))*5.D-1*(1.DO+Dcos(wi))
      titr1='DE HANNING-POISSON'
      RETURN
      END
C-----
C      SUBROUTINE CAUCHY
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE CAUCHY
C-----
      SUBROUTINE CAUCH(I,M,ALPHA,titr1)
      implicit real*8 (A-H,O-Z)

```

```

character*40 titr1
common/B1/ WIN(300)
WIN(I+1)=1.D0/(1.D0+(ALPHA*FLOAT(I)/FLOAT(M))**2)
titr1='DE CAUCHY'
RETURN
END

```

```

C-----
C      SUBROUTINE GAUSS
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE GAUSS
C-----

```

```

SUBROUTINE GAUS(I,M,ALPHA,titr1)
  implicit real*8 (A-H,O-Z)
  character*40 titr1
  common/B1/ WIN(300)
  WIN(I+1)=DEXP(-5.D-1*(ALPHA *FLOAT(I)/FLOAT(M))**2)
  titr1='DE GAUSS'
  RETURN
END

```

```

C-----
C      SUBROUTINE KAISER
C      CALCUL DU COEFFICIENT D'ORDRE I DE LA FENETRE DE KAISER
C-----

```

```

SUBROUTINE KAIS(I,M,ALPHA,titr1)
  implicit real*8 (A-H,O-Z)
  character*40 titr1
  common/B1/ WIN(300)
  PI=4.D0*DATAN(1.D0)
  x11=1-(float(i)/float(m))**2
  X1=ALPHA*DSQRT(x11)
  CALL INO(X1,N3)
  X2=ALPHA
  CALL INO(X2,N2)
  WIN(I+1)=X1/X2
  titr1='de kaiser'
  write(*,*) win(i)
  RETURN
END

```

```

C-----
C      SUBROUTINE INO
C      CALCULE LA FONCTION DE BESSEL IO(X) ET DONNE LE NOMBRE
C      D'OPERATIONS A EFFECTUER (N) POUR UNE VALEUR DONNEE DE L'ERREUR
C-----

```

```

SUBROUTINE INO(X,N)
DOUBLE PRECISION X
Y=X/2
T=1.D-08
E=1
DE=1
DO 1 I=1,25
  DE=DE*Y/FLOAT(I)
  SDE=DE**2
  E=E+SDE
  IF(E*T-SDE)1,1,2
1   CONTINUE
2   X=E
  N=I
  RETURN
END

```

```

C-----
C      SUBROUTINE FFT
C      XA:VECTEUR COMPLEX DE DIMENSION N
C      NU:ENTIER TEL QUE N=2**NU
C      MODE:ENTIER POUR CONTROLER LA FONCTION PERFORMEE PAR CETTE SUBROUTINE
C      MODE=0  CALCUL DE LA FFT
C      MODE=-1 CALCUL DE LA FFT INVERSE

```

C

```
SUBROUTINE FFT(X,NU,MODE)
implicit real*8 (a-h,o-z)
COMPLEX*16 X(1),W,A,B
N=2**NU
J=1
PI2=8.DO*DATAN(1.DO)
DO 27 I=1,N-1
  IF(I.LT.J)THEN
    B=X(J)
    X(J)=X(I)
    X(I)=B
  ENDIF
  K=N/2
2  IF(K.GE.J) GOTO 27
  J=J-K
  K=K/2
  goto 2
27 J=J+K
DO 28 L=1,NU
  LE=2**L
  ANG=-PI2/float(LE)
  LE2=LE/2
  A=(1.DO,0.DO)
  W=CMPLX(dCOS(ANG),+dSIN(ANG))
  IF(MODE.EQ.-1) W=CMPLX(dCOS(ANG),-dSIN(ANG))
  DO 28 J=1,LE2
    DO 3 I=J,N,LE
      ID=I+LE2
      B=X(ID)*A
      X(ID)=X(I)-B
3      X(I)=X(I)+B
28      A=A*W
      IF(MODE.EQ.-1)THEN
        DO 29 I=1,N
29          X(I)=X(I)/CMPLX(FLOAT(N),0.DO)
      ENDIF
    RETURN
  END
```

PROGRAM SIMULATION DES FILTRES RIF

C Ce programme permet la simulation des filtres RIF à phase
 C linéaire. En premier lieu, les coefficients de la réponse
 C impulsionnelle du filtre à simuler et les données du signal
 C d'entrée lui sont introduits.
 C Puis le choix de la structure du filtre à simuler doit être
 C fait.

C Paramètre d'entrée:

C -----
 C CHOIX=1----->choix de la structure en DIRECT
 C CHOIX=2----->choix de la structure en FFT

C Paramètre de sortie:

C -----
 C Y(I) : vecteur contenant les données du signal de sortie
 C du filtre simulé

C Fichiers d'entrée:

C -----
 C FICH : coefficients de la réponse impulsionnelle
 C FICH1 : coefficients du signal d'entrée

C Fichiers de sortie:

C -----
 C 'FSOR.DAT' : coefficients du signal de sortie du
 C filtre simulé

C -----
 C DECLARATIONS
 C -----

IMPLICIT REAL*8 (A-H,O-Z)
 DIMENSION X(1000),Y(1000),H(300)
 COMPLEX*16 XC(1000),YC(1000),HC(300)
 CHARACTER*40 FICH,FICH1
 INTEGER CHOIX,MODE

WRITE(*,*) 'INTRODUIRE LE FICHIER DE LA REPONSE IMPULSIONNELLE'

READ(*,*) FICH
 OPEN(1,file=FICH,status='old')
 READ(1,*) NFILT
 DO 11 I=1,NFILT
 READ(1,*) H(I)
 11 CONTINUE

WRITE(*,*) 'FICHIER DU SIGNAL A TRAITER'

READ(*,*) FICH1
 OPEN(2,file=FICH1,status='old')
 READ(2,*) NSIG
 DO 12 I=1,NSIG
 READ(2,*) X(I)
 12 CONTINUE

NU=8
 N=2**NU

DO 1 I=NFILT+1,N
 H(I)=0.DO
 1 CONTINUE

c DO 211 I=NSIG+1,N
 c X(I)=0.DO
 c 211 CONTINUE

13 WRITE(*,*) '*****'
 WRITE(*,*) '** CHOIX DE LA STRUCTURE DE REALISATION **'

```

WRITE(*,*) `*****`
WRITE(*,*) `**1-FORME DIRECTE (CONVOLUTION) **`
WRITE(*,*) `**2-FFT **`
WRITE(*,*) `*****`
READ(*,*) CHOIX

IF(CHOIX.LE.2) GOTO 4
WRITE(*,*) `RECOMMENCER,CHOISIR 1 OU 2`
GOTO 13

4    IF(CHOIX.EQ.2) GOTO 7

c    APPLICATION DE LA CONVOLUTION
c    *****
DO 6 K=1,NSIG
    Y(K)=0
    DO 5 I=1,NFILT
        KI1=K-I+1
        IF(KI1.GT.0) THEN
            Y(K)=Y(K)+H(I)*X(KI1)
        ENDIF
5    CONTINUE
6    CONTINUE

GOTO 9

c    APPLICATION DE LA FFT
C    *****
7    MODE=0
    DO 23 I=1,N
        HC(I)=cplx(H(I),0.d0)
        XC(I)=cplx(X(I),0.d0)
23   CONTINUE
    CALL FFT(NU,MODE,HC)
    CALL FFT(NU,MODE,XC)

DO 8 I=1,N
    YC(I)=XC(I)*HC(I)
8    CONTINUE

    MODE=-1
    CALL FFT(NU,MODE,Yc)
DO 33 I=1,N
    Y(I)=REAL(YC(I))
33   CONTINUE

C    CREATION DU FICHER DE SORTIE
C    *****
9    OPEN(3,file='FSOR.DAT',status='new')
    DO 10 I=1,NSIG
        WRITE(3,*) I-1,Y(I)
10   CONTINUE
    CLOSE(3)
    STOP
    END

C-----
C    SUBROUTINE FFT
C    V:VECTEUR COMPLEX DE DIMENSION N
C    NU:ENTIER TELQUE N=2**NU
C    MODE:ENTIER POUR CONTROLER LA FONCTION PERFORMEE
C    PAR CETTE SUBROUTINE
C    MODE=0   CALCUL DE LA FFT
C    MODE=-1  CALCUL DE LA FFT INVERSE
C-----

SUBROUTINE FFT(NU,MODE,V)
IMPLICIT REAL*8 (A-H,O-Z)

```

```

COMPLEX*16 W,A,B,V(300)
N=2**NU
J=1
PI2=8.DO*DATAN(1.DO)
DO 27 I=1,N-1
  IF(I.LT.J) THEN
    B=V(J)
    V(J)=V(I)
    V(I)=B
  ENDIF
  K=N/2
2   IF(K.GE.J) GOTO 27
    J=J-K
    K=K/2
    GOTO 2
27  J=J+K
DO 28 L=1,NU
  LE=2**L
  ANG=-PI2/float(LE)
  LE2=LE/2
  A=cmplx(1.DO,0.DO)
  W=CMPLX(DCOS(ANG),DSIN(ANG))
  IF(MODE.EQ.-1) THEN
    W=CMPLX(DCOS(ANG),-DSIN(ANG))
  ENDIF
  DO 28 J=1,LE2
    DO 3 I=J,N,LE
      ID=I+LE2
      B=V(ID)*A
      V(ID)=V(I)-B
3     V(I)=V(I)+B
28    A=A*W
      IF(MODE.EQ.-1) THEN
        DO 29 I=1,N
          V(I)=V(I)/CMPLX(FLOAT(N),0.DO)
29        ENDIF
      RETURN
    END
  END

```