ANALYSE, SYNTHESE ET COMPLEXITE DE CALCUL DE BANCS DE FILTRES NUMERIQUES

THESE No 617 (1986)

PRESENTEE AU DEPARTEMENT D'ELECTRICITE

ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ES SCIENCES

PAR

MARTIN VETTERLI

Master of Science, Stanford University
originaire de Stafa (ZH)

acceptée sur proposition du jury:

Prof. H. Nussbaumer, rapporteur
Prof. M. Kunt, corapporteur
Dr. C. Galand, corapporteur
Dr. M. Bellanger, corapporteur

Lausanne, EPFL
1986

Sommaire

Remerciements

> ... Je n'avais d'abord projeté qu'un mémoire de quelques pages; mon sujet m'entraînant malgré moi, ce mémoire devint insensiblement une espèce d'ouvrage trop gros, sans doute, pour ce qu'il contient, mais trop petit pour la matière qu'il traite. J'ai balancé longtemps à le publier; et souvent il m'a fait sentir, en y travaillant, qu'il ne suffit pas d'avoir écrit quelques brochures pour savoir composer un livre. Après de vains efforts pour mieux faire, je crois devoir le donner tel qu'il est, jugeant qu'il importe de tourner l'attention publique de ce côté-là; et que, quand mes idées seraient mauvaises, si j'en fais naître de bonnes à d'autres, je n'aurai pas tout à fait perdu mon temps.
>
> J.J.Rousseau, préface, "Emile ou de l'Education"

Tout d'abord, j'aimerais remercier le Professeur Nussbaumer pour avoir dirigé ce travail et pour la liberté qu'il m'a accordée dans l'exécution de celui-ci. Je lui suis reconnaissant pour ses compétences scientifiques, sa très grande disponibilité ainsi que la confiance qu'il m'a accordée.

Ensuite, je tiens à remercier le Professeur Kunt, d'une part pour avoir accepté de faire partie du jury de cette thèse et, d'autre part, pour son enthousiasme et ses encouragements constants. Je tiens à remercier les Docteurs Bellanger (TRT, Paris) et Galand (CER-IBM, La Gaude) pour avoir lu et critiqué ce travail et pour leur participation au jury, ainsi que le Professeur Bühler pour avoir présidé le jury de cette thèse.

Ma gratitude va également à Pierre Duhamel (CNET, Paris) pour une collaboration fort intéressante et fructueuse, à Riccardo Leonardi pour sa relecture sémantique critique du manuscript et de nombreuses discussions, ainsi qu'à Brigitte Kocher pour sa relecture syntaxique.

Mes remerciements vont au "Fonds National Suisse de la Recherche Scientifique" qui a soutenu en partie la recherche présentée dans ce travail.

Ensuite, je remercie les Professeurs Wohlhauser et Zwahlen ainsi que C.El-Hayek (Département de Mathématiques de l'EPFL) pour les discussions intéressantes et leur collaboration constructive, le Professeur Hasler (DE-EPFL, pour ses conseils concernant l'optimisation), J.Masson (TRT-Paris, pour ses critiques et l'élaboration de la figure

3.8) et les Dr. Haskell et Ninke (AT&T Bell Laboratories) pour avoir rendu possible mon séjour au sein de leur groupe de recherche.

Bien sûr, ma reconnaissance va à ma famille, aux collaborateurs et collaboratrices des Laboratoires d'Informatique Technique et de Traitement des Signaux (entre autres, à leurs secrétaires respectives pour leur sens de la réalité), à Michel Kocher (pour le Zen et l'art du traitement d'image), à Michael Unser (pour les nombreuses discussions et son programme de traitement de formules), à Adriaan Ligtenberg (pour la collaboration dans le N.J), à Eric Debourse (pour le travail sur le DSP), aux étudiants qui ont souffert sur mes projets (en particulier, M.Kardan et F.Schmitt), à mes amis musiciens (pour l'échappatoire) et à mes colocataires divers (pour leur patience).

Finalement et surtout, je tiens à remercier Marie-Laure Renevey, sans laquelle je n'écrirais pas ces lignes.

Résumé

Le traitement numérique du signal à l'aide de bancs de filtres, c'est-à-dire le filtrage d'un signal par plusieurs filtres, suivi éventuellement d'une réduction de la fréquence d'échantillonnage (ou l'opération duale, c'est-à-dire le multiplexage), est un sujet important aux applications multiples (codage en sous-bandes, transmultiplexage).

Ce travail développe d'abord une méthode d'analyse pour le traitement du signal par bancs de filtres. Cette méthode est basée sur une notation matricielle et une représentation polyphase généralisée des bancs de filtres. Ce formalisme puissant est utilisé afin de dériver des résultats fondamentaux sur les bancs de filtres. On montre que les signaux peuvent toujours être reconstruits sans repliements spectraux (ou diaphonie) à la sortie d'un banc de filtres, et dans quels cas la reconstruction peut être parfaite. La dualité entre codeurs en sous-bandes et transmultiplexeurs est démontrée, unifiant ainsi le traitement de ces deux cas.

On considère ensuite la synthèse de bancs de filtres en présentant d'abord des méthodes de conception (analytiques et par optimisation) et des exemples de filtres dans le cas où la reconstruction est parfaite avec des filtres à réponse impulsionnelle finie (RIF). Les bancs de filtres modulés (pseudo-QMF entre autres) ainsi que des extensions (filtres QMF complexes, cas bidimensionnel) sont ensuite introduits.

La complexité de calcul des bancs de filtres est alors évaluée, et on montre comment réduire sensiblement la charge de calcul dans des cas particuliers importants (arbres de filtres, bancs de filtres modulés). De ces considérations ressort l'importance de transformations rapides pour l'évaluation des bancs de filtres.

Le dernier chapitre traite du calcul de transformations rapides (Fourier, cosinus) et introduit un nouvel algorithme pour des transformeés de longueur $N=2^m$. Celui-ci est généralisé à toute une série de problèmes et permet toujours d'atteindre le nombre minimum d'opérations. Deux implantations de cet algorithme, l'une matérielle (une puce VLSI pour le codage vidéo en temps réel par transformée) et l'autre logicielle (du code efficace pour processeur de traitement du signal), sont décrites.

En résumé, ce travail montre que le problème des bancs de filtres est radicalement différent du problème classique des filtres isolés. Par les résultats apportés, il ouvre un certain nombre de perspectives nouvelles pour le traitement du signal par bancs de filtres.

Summary

Digital signal processing with filter banks, that is filtering a single signal with several filters and subsequent subsampling (or the converse, that is multiplexing), is an important subject with numerous applications (sub-band coding, transmultiplexing).

This work first develops an analysis framework for signal processing by means of filter banks. The framework is based on matrix notation and a generalized polyphase representation of filter banks. This powerful formalism is used to derive fundamental results on filter banks. It is shown that signals can always be reconstructed without aliasing (or crosstalk) and in which cases the reconstruction can be perfect. The duality between sub-band coders and transmultiplexers is also demonstrated, thus unifying the analysis of these two cases.

The synthesis of filter banks is considered next. Analytical and optimisation methods are presented, together with examples of perfect FIR reconstruction for arbitrary N. Modulated filter banks (among them, pseudo-QMF banks) and generalizations (complex QMF and the bidimensional case) are then introduced.

The computational complexity of filter banks is then considered, and it is shown how to substantially reduce the number of operations in some important cases (filter trees, modulated filter banks). This highlights the importance of fast transforms for the evaluation of filter banks.

The last chapter is concerned with the computation of fast transforms (Fourier, cosine) and introduces a new algorithm for length $N=2^m$ transforms. This algorithm is generalized to a whole set of problems and always achieves the minimum known number of operations. Two implementations of this algorithm, one in hardware (a VLSI chip for a real-time video coder) and the other in software (efficient code for a signal processor) are described.

In short, this work shows that filter bank problems are radically different from classical single filter problems. With the obtained results, a number of new perspectives on signal processing by filter banks have been opened.

Notations:

En traitement numérique du signal, la numérotation des vecteurs commence habituellement à zéro. Nous avons gardé cette convention dans tous les cas, même si elle est moins usuelle dans le contexte de l'algèbre linéaire.

D m,n : définition numéro n dans le chapitre m

R m,n : remarque numéro n dans le chapitre m

C m,n : condition numéro n dans le chapitre m

T m,n : théorème numéro n dans le chapitre m

Co m,n : corollaire se référant au théorème Tm,n

E m,n : Exemple numéro n dans le chapitre m

A m,n : annexe numéro n se référant au chapitre m

x : scalaire

$x(n)$: suite de scalaire

$X(z)$: transformée en z de x(n), fonction rationnelle en z^{-1}

$\mathbf{H}(z)$: matrice de fonctions rationnelles en z^{-1}

$\mathbf{H}_{i,j}(z)$: élément de la ligne i et de la colonne j de $\mathbf{H}(z)$

$[\mathbf{H}(z)]_{i,j}$: idem

$\mathbf{g}(z)$: vecteur colonne de fonctions rationnelles en z^{-1}

$\mathbf{g}_i(z)$: élément de la ligne i de $\mathbf{g}(z)$

$[\mathbf{g}(z)]_i$: idem

$\Delta(z)$: déterminant d'une matrice $\mathbf{H}(z)$

$\mathbf{C}(z)$: matrice des cofacteurs d'une matrice $\mathbf{H}(z)$

$\mathbf{H}_p(z)$: matrice de filtres polyphases

$\mathbf{H}_m(z)$: matrice de filtres modulés

N : nombre de canaux d'un banc de filtres

N' : facteur de sur/sous échantillonnage, s'il est différent du nombre de canaux d'un banc de filtres

M : longueur de filtres RIF

$H_i(z)$: filtre i

$h_{i,j}$: coefficient j du ième filtre RIF

j : racine de l'unité égale à $(-1)^{1/2}$

x^* : complexe conjugué de x

W : N-ième racine de l'unité, égale à $\exp(-j2\pi/N)$

$\mathbf{F}$: matrice de Fourier

$\mathbf{F}_{m,n}$: élément i,j de la matrice de Fourier égal à $\exp(-j2\pi mn/N)$

$\mathbf{H}^T$: transposée de $\mathbf{H}$

$\mathbf{H}^H$: transposée hermitienne de $\mathbf{H}$, égale à $[\mathbf{H}^{T*}]$

$\lfloor n/m \rfloor$: plus grand nombre entier plus petit ou égal à n/m

$\lceil n/m \rceil$: plus petit nombre entier plus grand ou égal à n/m

$\sim$: similaire, au sens d'une structure similaire

Diag[v] : matrice diagonale dont les éléments sont donnés par ceux du vecteur ligne ou colonne v

Rang[H] : donne le rang de la matrice H

$\{ x_i \}$: ensemble de valeurs

$\mathrm{Min}[\{x_i\}]$: donne la valeur minimum de l'ensemble des x_i

$\mathrm{Max}[\{x_i\}]$: donne la valeur maximum de l'ensemble des x_i

$O_a[..]$: donne le nombre d'additions

$O_m[..]$: donne le nombre de multiplications

m Mod N : donne la valeur de m modulo N

$\Pi\ x_i$: multiplication des termes x_i

$\delta(n)$: impulsion au temps discret n

1 Introduction

"Ce ne sont pas les choses qui nous troublent,
mais l'opinion que nous nous faisons des choses"
Epictète

1.1 Remarques liminaires

Le traitement numérique du signal, introduit il y a une quarantaine d'année, a connu depuis un essort considérable. Les raisons principales de cette évolution sont, outre la maîtrise accrue des problèmes à résoudre, la disponibilité de puissances de calcul de plus en plus grandes (par exemple avec l'introduction de processeurs spécialisés pour le traitement du signal) ainsi que des impératifs économiques (l'évolution du coût de la bande passante en télécommunication par exemple).

Le travail présenté ci-dessous touche principalement à deux domaines classiques du traitement numérique du signal: le filtrage numérique d'une part, et les algorithmes rapides pour l'évaluation du processus de filtrage d'autre part.

Rappelons qu'un banc de filtres permet d'obtenir à partir d'un seul signal d'entrée N signaux de sortie (ou vice versa), où N représente le nombre de filtres impliqués.

La notion fondamentale liée aux bancs de filtres est celle de simultanéité du traitement, puisque les N filtres sont appliqués à un seul signal d'entrée (ou bien ne produisent qu'un seul signal de sortie). Cette simultanéité a des conséquences importantes sur les propriétés des bancs de filtres et sur la complexité de calcul qui leur est associée. Nous faisons donc la remarque suivante:

"Le problème du filtrage simultané d'un signal par N filtres est radicalement différent de celui du filtrage effectué séparément".

Intrinsèquement, le problème des bancs de N filtres est un problème à N dimensions, et nous avons par conséquent été amené à introduire un formalisme matriciel afin

de le traiter comme tel. Ce formalisme permet d'établir une théorie des bancs de filtres relativement générale qui nous a permis d'apporter trois contributions nouvelles. Premièrement, cette théorie place, classe et explique les résultats actuellement connus sur les bancs de filtres. Ensuite, elle démontre que certains résultats nouveaux sont possibles avec les bancs de filtres, et prouve que d'autres sont impossibles. Finalement, elle permet d'explorer des horizons nouveaux et de découvrir ainsi des méthodes et des applications jusqu'alors inconnues.

Nous ne tenterons pas de masquer le fait que peu de résultats "pratiques" sont présentés sur les bancs de filtres à l'aide de cette théorie. Nous le justifions simplement par le fait que nous avons trouvé le niveau théorique d'une part relativement vierge, d'autre part passablement fructueux.

Outre la problématique des propriétés des bancs de filtres, il est important de considérer la complexité de calcul que demande leur mise en oeuvre. En particulier, ceci nous a amené à étudier des classes de bancs de filtres dont l'implantation requiert une complexité fortement réduite. Des résultats obtenus dans ce contexte et concernant la transformée de Fourier et en cosinus discrète rapide dépassent le cadre des bancs de filtres vu leurs applications multiples, et seront donc présentés exhaustivement.

1.2 Historique

1.2.1 Remarque sur le traitement numérique du signal

Historiquement, de nombreux résultats en traitement numérique du signal ont été obtenus en se référant au traitement analogique du signal. Pourtant, très tôt, une école s'est développée (par exemple [sit57]) qui considère le problème dans sa généralité, c'est à dire comme un traitement appliqué à une suite de nombres, sans faire référence au monde analogique dont cette suite peut être éventuellement issue. Cette approche directe, si elle est faite en toute généralité, a l'avantage, d'une part, d'inclure tous les résultats qui pourraient être obtenus en faisant référence au traitement analogique, mais d'autre part, surtout, de pouvoir obtenir des résultats nouveaux qui n'ont pas d'équivalent dans le monde analogique. Le travail ci-dessous s'inscrit dans cette école, et il sera évité tant que possible dans la suite de faire référence à une quelconque source analogique des signaux discrets que nous traiterons, ceci afin d'isoler clairement le problème considéré.

1.2.2 Historique des bancs de filtres

Les bancs de filtres peuvent se diviser en 3 classes principales définies par le type de traitement du signal pour lequel ils sont utilisés: l'analyse spectrale, le traitement de la parole et les transmultiplexeurs. Même si les contraintes sont fort différentes, le formalisme utilisé est commun. Dans ces systèmes, le banc de filtres associe un signal à N composantes (ou réciproquement) et en général, la fréquence d'échantillonnage du signal est supérieure à celle des composantes (typiquement d'un facteur N), ceci afin de préserver un débit constant (nombre d'échantillons par unité de temps).

a) Bancs de filtres pour l'analyse spectrale

L'analyse spectrale est vraisemblablement la plus ancienne application des bancs de filtres. En particulier, le spectrogramme [pot47] utilisé en phonétique acoustique est le résultat de sortie d'un banc de filtres réalisant une analyse spectrale glissante.

L'analyse de Fourier glissante [rab75] peut également être considérée comme un banc de filtres pour l'analyse spectrale. Quoique moins puissante que d'autres formes d'analyse simultanée en temps et en fréquence comme par exemple la représentation de Wigner [cla80], elle est néanmoins très populaire en raison de sa simplicité. Elle a été abondamment étudiée [all77,por80,gri84], également en raison de son implantation efficace [uns83]. Il est à noter que la mise en oeuvre se fait souvent avec un résonateur complexe réalisant l'annulation d'un pôle par un zéro. Quoique cette méthode soit numériquement mal conditionnée (voire instable) [gol80], la solution équivalente à réponse impulsionnelle finie [bru78,vet83] est rarement utilisée. Finalement, notons que si pour l'analyse spectrale en temps différé (off line) on peut utiliser des méthodes beaucoup plus puissantes parce qu'adaptatives (par exemple la méthode de l'entropie maximum [abl78]), l'analyse spectrale à l'aide de bancs de filtres fixes reste importante pour le traitement en temps réel en raison de sa simplicité.

b) Bancs de filtres pour le traitement de la parole

Avec l'introduction du codage de la parole en sous-bandes [crc76] et des filtres miroirs en quadrature [cro76], les bancs de filtres ont connu une apparition tardive mais prometteuse en compression de la parole.

Le concept de filtres miroirs en quadrature [gal84] (ou QMF pour "quadrature mirror

filters") est important, car il introduit la notion de filtres simultanés. Le repliement spectral apparaissant en raison du sous-échantillonnage est éliminé par annulation cohérente entre les canaux lors de la synthèse. Ceci est une utilisation explicite du fait que l'on sous-échantillonne simultanément plusieurs versions d'un même signal original. La méthode habituelle de suppression des repliements spectraux utilise un filtre passe-bas "idéal", ce qui est la seule méthode possible si l'on ne sous-échantillonne qu'une seule version du signal.

Le perfectionnement des techniques de codage en sous-bandes [gal83] a permis d'obtenir une compression appréciable avec une qualité adéquate pour la transmission téléphonique du signal de parole, donc d'établir cette méthode comme candidate valable dans le contexte de la numérisation du réseau téléphonique.

Nombreux sont les développements qui ont suivi les travaux initiaux sur le codage en sous-bandes et les filtres QMF. Le perfectionnement et la conception des filtres QMF non-récursifs [joh80], puis leur réalisation récursive [ram80,mil85] ont été explorés. La représentation analytique du signal a été rendue possible avec l'introduction des QMF complexes (CQMF) 81[nus83a]. L'implantation efficace de bancs QMF a été réalisée avec les bancs dits "pseudo-QMF" [nus81,rot83, nus84a,nus84b,mas85,chu85], qui allient une propriété de suppression partielle du repliement spectral (en fait, du repliement spectral dominant) avec une complexité de calcul réduite (en utilisant l'approche des transmultiplexeurs, voir ci-dessous). Finalement, un type de filtres QMF parfaits a été introduit [smi84,wac85], permettant une reconstruction parfaite avec des filtres d'analyse et de synthèse non-récursifs.

c) Bancs de filtres utilisés dans les transmultiplexeurs

Les transmultiplexeurs permettent de réaliser la transformation d'un multiplexage temporel en un multiplexage fréquentiel (ou inversement), un processus important en télécommunications [com78,com82]. Essentiellement, il s'agit d'évaluer simultanément N filtres qui sont dérivés par modulation à partir d'un seul filtre prototype. La démonstration que ce problème était équivalent à celui d'évaluer N filtres réduits d'un facteur N (appelés filtres polyphases) ainsi qu'une transformée de Fourier [bel84] a permis des implantations très efficaces de ces bancs de filtres [bel76,nar79]. De tels bancs de filtres ont également été utilisés pour l'analyse spectrale [var80,heu81] et permettent en quelque sorte une généralisation de la transformée de Fourier glissante: la transformée de Fourier utilisée dans ces bancs donne la notion d'analyse à des points équidistants dans le domaine fréquentiel, et les filtres polyphases déterminent la résolution spectrale de cette analyse.

En résumé, deux notions fondamentales apparaissent dans le développement historique des bancs de filtres: d'abord, la notion de filtres simultanés permettant la maîtrise de certains effets indésirables apparaissant avec des filtres isolés (en particulier, le contrôle des repliements spectraux); ensuite la notion de filtres modulés pouvant être interprétée comme une transformée de Fourier conjuguée avec des filtres polyphases.

1.2.3 Historique des transformées rapides

Cet historique comporte trois parties, une sur la transformée de Fourier, une sur les transformées utilisées en codage et enfin une dernière sur les implantations des transformées rapides.

a) Complexité de calcul de la transformée de Fourier

Nous ne tenterons pas de faire un historique complet du développement des transformées rapides (celle de Fourier en particulier), puisqu'une bibliographie récente [hei84a] recense plus de deux milles articles et livres à ce sujet. Un excellent traitement historique peut être trouvé dans [hei84b], où il est entre autres montré que l'algorithme communément connu comme algorithme de FFT (fast Fourier transform) de Cooley-Tukey [coo65] remonte en fait à Gauss!

La réintroduction de cet algorithme en 1965 a pourtant constitué un évènement majeur, puisqu'il a donné lieu à un foisonnement de développements ultérieurs. Parmi ceux-ci, notons celui de la FFT à base mixte [sin69] et de la FFT à facteurs réels [rad76,bru78]. En 1968, une contribution restée négligée jusqu'à une période récente [yav68] établit un nombre d'opérations minimal connu à ce jour pour la FFT réelle et complexe sur des longueurs égales à des puissances de 2.

Pour la FFT portant sur des séquences dont les longueurs sont le produit de facteurs premiers entre eux, Good [goo60] avait déjà montré qu'elle était équivalente à une transformée multidimensionnelle bien avant la publication de l'algorithme de FFT de Cooley et Tukey, mais l'importance de ce résultat avait été négligée. La publication par Rader [rad68] d'une méthode permettant l'évaluation efficace d'une transformée de Fourier sur un nombre premier d'échantillons a permis d'utiliser avec profit la FFT sur des longueurs ayant des facteurs premiers entre eux, puisque les FFT de petite longueur peuvent ainsi être évaluées efficacement. La conséquence immédiate a été le "Prime Factor Algorithm" [kol77], dans lequel une FFT sur N échantillons (où est $N=\prod N_i$ et les N_i sont premiers entre eux) est évaluée comme une transformée multidimensionnelle par la méthode ligne/colonne. Une méthode encore

plus performante a été introduite par Winograd [win76] pour évaluer ces transformées multidimensionnelles et est réalisée par un calcul "emboîté" sur de petites transformées. L'algorithme de Rader combiné avec celui de Winograd permet d'obtenir une complexité linéaire pour le calcul de la transformée de Fourier. En plus d'un nombre réduit de multiplications, le nombre d'additions reste également intéressant. Si le nombre d'échantillons n'est pas un produit de facteurs premiers entre eux, mais par exemple une puissance de 2, on peut utiliser des méthodes similaires pour démontrer la complexité linéaire [nus82,duh84b,hei85]. Dans ce cas, par contre, le nombre d'additions est nettement plus élevé que dans les algorithmes classiques et rend donc cette approche peu attrayante.

Pour les transformées multidimensionnelles, outre l'approche évidente utilisant la séparabilité ainsi que l'approche dite du "vector radix" (qui est une généralisation multidimensionnelle de l'algorithme de FFT), la contribution majeure a été l'introduction par Nussbaumer des transformées polynômiales [nus78,nus79,nus82]. Les transformées polynômiales sont utilisées pour réduire une transformée de Fourier multidimensionnelle en une série de transformées monodimensionnelles, et ceci d'une manière beaucoup plus efficace que dans les approches traditionnelles, résultant en un gain d'un facteur 2 et plus dans le nombre d'opérations.

Récemment, les algorithmes FFT classiques (pour des longueurs égales à des puissances de 2) ont connu un regain d'intérêt par l'introduction d'algorithmes atteignant le nombre minimum connu d'opérations [duh84a,vet84b,mar84] et qui sont bien adaptés au traitement de séquences réelles ou symétriques/antisymétriques [duh85,vet85d].

b) Les transformées utilisées pour le codage

Si la transformée de Fourier est fort utile entre autres grâce à sa propriété de diagonalisation des matrices de Toeplitz circulantes (donc à la possibilité de faire du filtrage dans le domaine transformé à l'aide d'une multiplication complexe par point fréquentiel), il est souvent nécessaire de diagonaliser des matrices de Toeplitz qui n'ont pas la propriété de circularité. Ceci est le cas dans les applications de codage et de reconnaissance de formes, où l'on cherche à décorréler les échantillons entre eux, donc à diagonaliser une matrice de corrélation. Dans ce cas, la transformée optimale est la transformée de Karhunen-Loève [pap65] où la matrice de transformation est obtenue à partir des vecteurs propres de la matrice de corrélation. Outre le fait que cette transformation dépend des propriétés statistiques du signal (donc n'est pas une transformation fixe), elle utilise $O[N^2]$ opérations (où N est le nombre d'échantillons à décorréler). De ce fait, des approximations ont été

proposées qui remplissent deux conditions: i) elles sont asymptotiquement équivalentes (N tendant vers l'infini) à la transformée de Karhunen-Loève [uns84] et ii) elles peuvent être mises en oeuvre avec un algorithme rapide.

La première et la plus populaire des approximations qui ait été proposée est la transformation en cosinus discrète [ahm74]. Cette transformée diagonalise la matrice d'autocorrélation d'un processus de Markov du premier ordre lorsque la corrélation tend vers 1 ou la dimension tend vers l'infini. Jain [jai79] a proposé toute une série de transformées sinusoïdales permettant de diagonaliser des matrices d'autocorrélation particulières, entre autres la transformée en sinus discrète, utile lorsqu'un processus de Markov du premier ordre a une corrélation très faible.

Les algorithmes rapides pour ces transformées sont de deux types distincts; d'une part les algorithmes directs [che77,wan83] qui ont l'avantage d'être réels, d'autre part les algorithmes passant par la transformée de Fourier [nar78] qui peuvent ainsi utiliser les meilleurs algorithmes de FFT et sont avantageux du point du vue du nombre d'opérations.

Notons finalement que la transformée en cosinus trouve également des applications dans les bancs de filtres modulés, car pour des bancs de filtres réels, on utilise une modulation en cosinus plutôt qu'une modulation en exponentiels complexes. Dans ce cas, le banc de filtres s'évalue à l'aide d'une transformée en cosinus et de filtres polyphases [nar79].

c) Implantation des transformées rapides

Jusqu'ici, nous nous sommes intéressés uniquement à la complexité de calcul théorique, voire même qu'au nombre de multiplications utilisé par un algorithme donné. Ceci est dû au fait que ces mesures de complexité se basent sur des fondements théoriques bien établis, et que la comparaison entre différents algorithmes est aisée. Pourtant, les résultats médiocres obtenus avec l'implantation logicielle de l'algorithme de Winograd en particulier (où les gains en nombre de multiplications n'ont pas pu être traduits en amélioration de vitesse) ont relancé le débat sur les mesures d'efficacité d'un algorithme. Foncièrement, la seule question pertinente d'un utilisateur d'algorithme reste de savoir lequel s'exécute le plus rapidement possible sur son ordinateur. Evidemment, cette question ne conduit pas à un formalisme mathématique évident, mais elle replace le débat sur l'efficacité dans un contexte plus proche de la réalité.

Une conséquence directe de cette remarque est la prise en compte (au niveau de la complexité de calcul) du nombre total d'opérations, où l'on ne compte pas seulement les multiplications, mais également les additions et les transferts de données [mor78,naw79]. La motivation pour cette nouvelle approche vient du fait que le rapport (temps de multiplication)/(temps d'additions) et (temps de multiplication)/(temps de transfert) s'approche de plus en plus de l'unité sur les ordinateurs modernes. Une autre conséquence est la tentative de générer du code machine aussi efficace que possible, appelé code linéaire [mor77], afin d'éliminer toute opération inutile pour le résultat du traitement (telles que boucles, compteurs et décisions) et de ne garder que les opérations qui affectent les variables. Néanmoins, surtout dans le contexte d'ordinateurs évolués, l'efficacité d'un algorithme donné reste une fonction non-triviale du nombre total d'opérations ainsi que de sa complexité structurelle [nus83b].

Du côté des implantations matérielles, les possibilités de l'intégration à très large échelle ont permis d'envisager la réalisation de transformées rapides sur des puces de silicium. Là encore, la notion d'efficacité prend un sens différent, car même si le nombre de multiplications reprend tout son poids, il faut tenir compte du stockage et surtout des transferts de données. Le coût des transferts de données peut devenir prépondérant, surtout si la structure est complexe. Les algorithmes rapides de transformée de Fourier discrète ont en général un coût élevé en ce qui concerne les transferts de données, ce qui a donné lieu dans un premier temps à un retour à l'algorithme réalisant simplement la multiplication du vecteur d'échantillons par la matrice de Fourier (utilisant donc N^2 opérations mais ne requérant que de la communication locale). Cet algorithme simple est donc adapté aux réseaux systoliques [mea80], ce qui n'est pas le cas des algorithmes rapides.

D'un point de vue théorique, la mesure d'efficacité est la surface de silicium utilisée multipliée par le temps d'exécution (voire le temps d'exécution au carré si le délai est important) [mea80]. Des limites inférieures pour cette mesure d'efficacité [vui83] et des algorithmes atteignant ces limites [bau83] ont bien été dérivés, mais trop peu d'implantations réelles ont été réalisées jusqu'à présent pour qu'il soit possible de mesurer l'efficacité réelle des algorithmes proposés.

En résumé, il ressort de cet historique des transformées rapides que la complexité théorique de la transformée de Fourier (tant pour des longueurs composites que pour des longueurs égales à des puissances d'un même facteur) semble approcher un minimum pour la somme des multiplications et des additions requises. Par contre, pour les implantations tant logicielles que matérielles, un grand effort reste à fournir afin de traduire ces avances théoriques en gains effectifs.

1.3 Contributions de l'étude

Après un bref survol des résultats présentés dans cette étude, nous placerons ces résultats dans le contexte actuel des recherches dans le domaine, afin de préciser nos contributions et leurs applications éventuelles.

1.3.1 Survol de l'étude

Le chapitre 2 présente l'analyse des bancs de filtres. En particulier, le problème des bancs de filtres sous-échantillonnés permettant la reconstruction parfaite du signal original est exploré en détail. Pour ce faire, une méthode d'analyse matricielle du problème est introduite. Des matrices de filtres sont définies. Leurs éléments sont donc des fonctions rationnelles de z. Il en ressort deux constatations pertinentes: l'importance du déterminant de la matrice de filtres, ainsi qu'une interprétation générale des bancs de filtres sous-échantillonnés comme étant des bancs de filtres polyphases. Le problème dual du multiplexage de N signaux sur un seul signal est également considéré.

Après la présentation du problème, on introduit deux modes de représentation complémentaires, la représentation dans le plan de modulation et celle dans le plan polyphase. Cette dernière est particulièrement intéressante, car elle évite la redondance implicite à la première. On analyse ensuite les propriétés intrinsèques des matrices de filtres liées aux bancs avec reconstruction (tel que le déterminant, les cofacteurs et l'inverse) et on démontre quelques théorèmes de base liés aux bancs de filtres, entre autres l'équivalence entre les bancs d'analyse et de synthèse avec reconstruction.

Le chapitre 3 concerne la conception de bancs de filtres. Quoique très lié au chapitre précédent, il prend un point de vue plus pratique. Le cas à deux canaux est d'abord adressé, tant pour des filtres à réponse impulsionnelle finie (RIF) que pour des filtres à réponse impulsionnelle infinie (RII). La classe de solutions générales où les filtres d'analyse et de synthèse sont RIF, et où le signal est parfaitement reconstitué, est présentée. Cette classe de filtres est beaucoup plus étendue que les cas particuliers connus jusqu'à présent [smi84,wac85]. En particulier, nous démontrons l'existence de filtres à phase linéaire, et nous présentons des exemples de tels filtres.

Le cas des bancs de N filtres est ensuite analysé dans le cas RIF et RII. Dans le cas à réponse impulsionnelle finie, nous montrons comment trouver des bancs de filtres

de taille quelconque (nombre de filtres et longueur des filtres arbitraires) qui permettent la reconstruction parfaite à l'aide de filtres d'analyse et de synthèse FIR. En ce qui concerne les bancs de filtres modulés, nous avons montré que la reconstruction ne peut jamais être parfaite avec des filtres RII (sauf si la longueur des filtres est égale à leur nombre). Les bancs de filtres QMF complexes, permettant d'obtenir une approximation du signal analytique, sont ensuite analysés. Finalement, certains résultats sont étendus au cas bidimensionnel.

Le chapitre 4 traite le problème de la complexité de calcul liée aux bancs de filtres. On considère d'abord l'évaluation des arbres de filtres, et ceci tant dans le domaine temporel que fréquentiel. Une méthode efficace pour l'évaluation des arbres de filtres dans le domaine de Fourier est proposée. Le cas des bancs de filtres modulés est traité ensuite. Celui-ci est intéressant, puisque l'évaluation d'un banc de filtres modulés peut être faite à l'aide d'un banc de filtres réduits (filtres polyphases) et d'une transformée rapide (Fourier, cosinus). Outre les approches connues, on propose une manière différente de calculer de tels bancs de filtres dans le domaine transformé (et faisant appel à des transformations bidimensionnelles).

Le chapitre 5 est consacré aux transformées rapides dont on a vu l'utilité pour l'évaluation des bancs de filtres modulés. Un algorithme simple, appelé algorithme de FFCT ("fast Fourier cosine transform"), est développé et nous montrons que cet algorithme permet d'atteindre le nombre minimum connu d'opérations pour toute une série de problèmes (transformée de Fourier sur des séquences complexes, réelles, symétriques et antisymétriques, transformée en cosinus, transformées impaires etc), et ceci pour des longueurs égales à des puissances de 2. L'application au calcul des transformées bidimensionnelles est également proposée.

La faisabilité d'une réalisation matérielle de cet algorithme a été démontrée grâce à la conception d'un circuit intégré à très haute densité (35000 transistors) calculant une transformée en cosinus avec un débit de 120 Mbits/sec (suffisant pour le traitement vidéo en temps réel). L'étude de la complexité de ce circuit nous a amené à considérer brièvement une méthodologie de développement pour l'implantation d'algorithmes de traitement du signal en silicium. Une méthodologie similaire a également été utilisée pour produire du logiciel réalisant les algorithmes rapides développés précédemment sur un processeur spécialisé. Ceci a permis d'obtenir de manière grandement automatisée du code très efficace réalisant des transformées à structure complexe. Les résultats avec le matériel et le logiciel ont permis de démontrer l'utilité pratique des algorithmes introduits.

1.3.2 Contributions de l'étude présentée

En ce qui concerne les bancs de filtres, un résultat intéressant que nous avons obtenu est le formalisme d'analyse matriciel utilisé [vet85c,vet86b]. Même si ce formalisme a été proposé presque simultanément dans deux autres publications [ram84,smi85], son application systématique, ainsi que sa version polyphase restée originale, nous a permis de trouver des résultats nouveaux, en particulier sur les bancs de filtres RIF. Dans ce cas, il n'existait jusqu'alors qu'une classe particulière de filtres RIF permettant une synthèse RIF parfaite, et ceci uniquement pour le cas N=2 [smi84,wac85]. A présent, non seulement la classe pour N=2 a été généralisée (incluant des solutions à phase linéaire), mais le cas N>2 a également pu être résolu. Le problème de la suppression des repliements spectraux et celui de la reconstruction parfaite du signal original dans les bancs de filtres sous-échantillonnés ont pu être traités en détail. Plusieurs autres résultats, entre autres sur les bancs de filtres modulés et sur les approximations pseudo-QMF, ont été démontrés, établissant ainsi ce formalisme matriciel des bancs de filtres comme un outil d'analyse puissant. La généralisation des filtres QMF au cas multidimensionnel a également été proposée [vet84a].

Pour l'évaluation des bancs de filtres, nous avons poursuivi les approches classiques des filtres polyphases suivis d'une FFT [bel74,bel76], en cherchant des fonctions de modulation adaptées aux problèmes donnés mais ayant tout de même des algorithmes efficaces. En particulier, les bancs de filtres pseudo-QMF, d'abord introduits dans [nus81] puis développés dans [rot83,nus84a,nus84b], allient l'implantation efficace du type transmultiplexeur avec la propriété de reconstruction quasi parfaite du type bancs de filtres QMF. Des bancs de filtres orthogonaux à complexité réduite ont également été proposés [vet83].

Dans le domaine des transformées rapides, il a été possible de développer un algorithme de transformée de Fourier et cosinus original [vet84b] qui atteint un nombre minimum connu d'opérations [duh84a,mar84], puis de le généraliser à toute une classe de problèmes [vet85a,vet85b,vet85d,duh86a]. Cet algorithme a donné lieu à une implantation matérielle [vet86a] et logicielle [vet85e], ainsi qu'à la proposition d'une méthodologie de transformation d'algorithmes de traitement du signal pour l'implantation [lig86]. Ces travaux nous ont permis de démontrer l'utilité pratique de l'algorithme.

En résumé, plusieurs résultats nouveaux intéressants ont été obtenus sur les bancs de filtres et les problèmes théoriques qui leurs sont liés, sur leur implantation en particulier dans le cas de filtres modulés, et finalement sur les transformées rapides utilisées lors de l'implantation de ces bancs de filtres.

Références bibliographiques du chapitre 1:

[abl78] J.B.Ables, "Maximum Entropy Spectral Analysis", in **Modern Spectrum Analysis**, D.G.Childers Ed., IEEE Press, New York, 1978.

[ahm74] N.Ahmed, T.Natarajan, and K.R. Rao, "Discrete Cosine Transform", IEEE Trans. on Computers, Vol. C-23, pp.88-93, Jan. 1974.

[all77] J.B.Allen, "Short Term Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform", IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-25, pp.235-238, June 1977.

[bau83] G.M. Baudet, F.P. Preparata, and J.E. Vuillemin, "Area-Time Optimal VLSI Circuits for Convolution", IEEE Trans. on Computers, Vol. C-32, No. 7, pp.684-687, July 1983.

[bel74] M.G.Bellanger, and J.L.Daguet, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT", IEEE Trans. on Communications, Vol. COM-22, No.9, pp. 1199-1204, Sept. 1974.

[bel76] M.G.Bellanger, G.Bonnerot and M.Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample-Rate Alteration and Filter Banks", IEEE Trans. on Acoust., Speech, Signal Processing, Vol. ASSP-24, No.2, pp. 109-114, April 1976.

[bru78] G.Bruun, "z-Transform DFT Filters and FFT's", IEEE Trans. Acoust. Speech Signal Process., Vol. ASSP-26, No.1, Feb. 1978, pp 56-63.

[che77] W-H.Chen, C.H.Smith, and S.C.Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Trans. on Communications, Vol. COM-25, pp.1004-1009, Sept. 1977.

[chu85] P.L.Chu, "Quadrature Mirror Filter Design for an Arbitrary Number of Equal Bandwidth Channels", IEEE Trans. Acoust. Speech Signal Process., Vol. ASSP-33, No.1, pp.203-218, Feb. 1985.

[cla80] T.A.C.M.Claasen, and W.F.G.Mecklenbräuker, "The Wigner Distribution - A Tool for Time-Frequency Signal Analysis", Part I, II and III, Philips Journal of Research, Vol.35, No.3, pp.217-250, No.4/5, pp.276-300, No.6, pp.372-389, 1980.

[com78] Special Issue on TDM-FDM Conversion, IEEE Trans. on Communications, Vol. COM-26, No.5, May 1978.

[com82] Special Issue on transmultiplexers, IEEE Trans. on Communications, Vol. COM-30, No.7, July 1982.

[coo65] J.W.Cooley, and J.W.Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Math. of Comput., Vol.19, pp.297-301, April 1965.

[cro76] R.E.Crochiere, S.A.Webber, and J.L.Flanagan, "Digital Coding of Speech in Sub-bands", Bell System Technical Journal, Vol.55, No.8, Oct.1976, pp.1069-1085.

[cro83] R.E.Crochiere, and L.R.Rabiner, **Multirate Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1983.

[cro76] A.Croisier, D.Esteban, and C.Galand, "Perfect Channel Splitting by Use of Interpolation, Decimation, Tree Decomposition Techniques", Int. Conf. on Information Sciences/Systems, Patras, pp. 443-446, Aug. 1976.

[duh84a] P.Duhamel, and H. Hollmann, "Split-Radix FFT Algorithms", Electronics Letters, Vol.20, No.1, 5th Jan. 1984.

[duh84b] P.Duhamel, and H. Hollmann, "Existence of a 2^n FFT Algorithm with a Number of Multiplications lower than 2^{n+1}", Electronics Letters, Vol.20, No.17, pp.690-692, Aug. 1984.

[duh85] P.Duhamel, and H. Hollmann, "Implementation of 'Split-Radix' FFT Algorithms for Complex, Real and Real-Symmetric Data", Proceedings of the IEEE Intl. Conf. on ASSP,Tampa, March 1985.

[duh86a] P.Duhamel, and M.Vetterli, "Cyclic Convolution of Real Sequences: Hartley versus Fourier and new Schemes", Proc. of the 1986 IEEE Intl. Conf. on ASSP, Tokyo, April 1986.

[gal84] C.R.Galand, and H.J.Nussbaumer, "New Quadrature Mirror Filter Structures", IEEE Trans. on Acoust., Speech and Signal Proc., Vol.32, No.3, June 1984, pp.522-531.

[gri84] D.W.Griffin, and J.S.Lim, "Signal Estimation from Modified Short-Time Fourier Transform", IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-30, No.3, pp.236-242, April 1984.

[goo60] I.J.Good, "The Interaction Algorithm and Practical Fourier Analysis", J.Roy.Stat.Soc., Vol. B-20, pp. 361-372 (1958), Vol. B-22, pp. 372-375, (1960).

[gol80] B.G.Goldberg, "A Continuous Recursive DFT Analyser - The Discrete Coherent Memory Filter", IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-28, No. 6, pp.760-762, Dec. 1980.

[hei84a] M.T.Heideman, and C.S.Burrus, "A Bibliography of Fast Transform and Convolution Algorithms II", Technical Report No.8402, Rice University, Feb.24 1984.

[hei84b] M.T.Heideman, D.H.Johnson, and C.S.Burrus, "Gauss and the History of the FFT", IEEE ASSP Magazine, Vol.1, No.4, Oct.1984, pp.14-21.

[hei85] M.T.Heideman, and C.S.Burrus, "Multiply/Add Tradeoffs in Length-2^n FFT Algorithms", Proc. IEEE Conf. on ASSP, Tampa, Fl, 1985, pp.780-783.

[heu81] U.Heute, and P.Vary, "A Digital Filter Bank with Polyphase Network and FFT Hardware: Measurements and Applications", Signal Processing, Vol.3, No.4 pp. 307-319, Oct. 1981.

[jai79] A.K. Jain, "A Sinusoidal Family of Unitary Transforms", IEEE Trans. on PAMI, Vol.1, No.4, pp.356-365, Oct.1979.

[joh80] J.D.Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks", Int. Conf. On ASSP, ICASSP 80, pp.291-294, Denver, 1980.

[kol77] D.P.Kolba, and T.W.Parks, "A Prime Factor Algorithm Using High-Speed Convolution", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-25, pp.281-294, Aug.1977.

[lig86] A.Ligtenberg, M.Vetterli, and J.H.O'Neill, "MOVAL: A Framework for Turning Digital Signal Processing Algorithms into Custom Chips", à paraître dans Signal Processing, Juin 1986.

[mar84] J.B.Martens, "Recursive Cyclotomic Factorisation - A New Algorithm for calculating the Discrete Fourier Transform", IEEE Trans. on ASSP, Vol. ASSP-32, N0.4, Aug.1984.

[mas85] J.Masson, and Z.Picel, "Flexible Design of Computationaly Efficient Nearly Perfect QMF Filter Banks", Proc. 1985 IEEE Conf. on ASSP, Tampa, March 1985.

[mea80] C. Mead, and L. Conway, **Introduction to VLSI**, Addison-Wesley, 1980.

[mil85] P.C.Millar, "Recursive Quadrature Mirror Filters .- Criteria, Specification and Design Method", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-33, No.2, pp.413-420, April 1985.

[mor77] L.R.Morris, "Automatic Generation of Time Efficient Digital Signal Processing Software", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-25, pp.74-78, Feb. 1977.

[mor78] L.R.Morris, "A Comparative Study of Time Efficient FFT and WFTA Programs for General Purpose Computers", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-26, pp.141-150, April 1978.

[nar78] M.J.Narasimha, and A.M.Peterson, "On the Computation of the Discrete Cosine Transform", IEEE Trans. on Communications, Vol. COM-26, pp.934-936, June 1978.

[nar79] M.J.Narasimha, and A.M.Peterson, "Design of a 24-Channel Transmultiplexer", IEEE Trans. on ASSP, Vol. ASSP-27, No.6, pp. 752-762, Dec.1979.

[nus78] H.J. Nussbaumer, and P. Quandalle, "Computation of Convolutions and Discrete Fourier Transforms by Polynomial Transforms", IBM J. Res. Dev., Vol. 22, pp. 134-144, 1978.

[nus79] H.J. Nussbaumer, and P. Quandalle, "Fast Computation of Discrete Fourier Transforms Using Polynomial Transforms", IEEE Trans. on ASSP, Vol. ASSP-27, pp. 169-181, 1979.

[nus81] H.J.Nussbaumer, "Pseudo QMF Filter Bank", IBM Technical Disclosure Bulletin, Vol.24, No.6, pp 3081-3087, Nov.1981.

[nus82] H.J.Nussbaumer, **Fast Fourier Transform and Convolution Algorithms**, Springer, Berlin, 1982.

[nus83a] H.J.Nussbaumer, "Complex Quadrature Mirror Filters", Proc. 1983 Int. IEEE Conf. on ASSP, Boston, March 1983.

[nus83b] H.J.Nussbaumer, "Efficient Algorithms for Signal Processing", Second European Signal Processing Conference, EUSIPCO-83, Erlangen, Sept. 1983.

[nus84a] H.J.Nussbaumer, and M.Vetterli, "Computationally Efficient QMF Filter Banks", Proc. 1984 Int. IEEE Conf. on ASSP, San Diego, March 1984.

[nus84b] H.J.Nussbaumer, and M.Vetterli, "Pseudo Quadrature Mirror Filters", Proc. of Int. Conf. on Digital Signal Processing, Florence, Sept. 1984.

[pap65] A.Papoulis, "Probability, Random Variables, and Stochastic Processes", McGraw-Hill Book Co., New York, 1965.

[por80] M.R.Portnoff' "Time-Frequency Representation of Digital Signals and Systems based on Short-Time Fourier Analysis", IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-28, pp.55-69, Feb. 1980.

[pot47] R.K.Potter, G.A.Kopp, and H.C.Green, "**Visible Speech**", New York, Van Nostrand 1947.

[rab75] L.R.Rabiner, and B.Gold, **Theory and Application of Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1975.

[rad68] C.M.Rader, "Discrete Fourier Transforms when the Number of Data Samples is Prime", Proc. IEEE, Vol. 56, pp.1107-1008, 1968.

[rad70] C.M. Rader, "An Improved Algorithm for High Speed Autocorrelation with Application to Spectral Estimation", IEEE Trans. Audio and Electroacoust., Vol. AU-18, No. 4, pp.439-441, Dec. 1970.

[rad76] C.M.Rader, and N.M.Brenner, "A New Principle for Fast Fourier Transformation", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-24, pp.264-265, June 1976.

[ram80] T.A.Ramstad, and O.Foss, "Sub-band Coder Design Using Recursive Quadrature Mirror Filters", in **Signal Processing: Theories and Applications**, EUSIPCO 1980, North-Holland, Amsterdam.

[ram84] T.A.Ramstad, "Analysis/synthesis filterbanks with critical sampling", Intl. Conf. on DSP, Florence, Sept.1984, pp. 130-134.

[rot83] J.H.Rothweiler, "Polyphase Quadrature Filters - A New Subband Coding Technique", Proc. 1983 Int. IEEE Conf. on ASSP, pp 1280-1283, Boston, March 1983.

[sin69] R.Singleton, "An Algorithm for Computing the Mixed Radix Fast Fourier Transform", IEEE Trans. on Audio. and Electroacoustics, Vol. AU-17, pp. 93-103, June 1969.

[sit57] R.W.Sittler, "Lectures on Sampled Data Systems Analysis", Lecture Notes, MIT Lincoln Labs, 1957.

[smi84] M.J.T.Smith, T.P.Barnwell, "A procedure for designing exact reconstruction filterbanks for tree structured sub-band coders", Proc. IEEE ICASSP-84, San Diego, March 1984.

[smi85] M.J.T.Smith, T.P.Barnwell, "A Unifying Framework for Analysis/Synthesis Systems Based on Maximally Decimated Filter Banks", Proc. IEEE ICASSP-85, pp. 521-524, Tampa, March 1985.

[Uns83] M.Unser, "Recursion in Short-Time Signal Analysis", Signal Processing, Vol.5, No.3, pp. 229-240, May 1983.

[Uns84] M.Unser, "On the Approximation of the Discrete Karhunen-Loève Transform for Stationary Processes", Signal Processing, Vol.7, No.3, pp. 231-249, Dec.1984.

[var80] P.Vary, and U.Heute, "A Short-Time Spectrum Analyzer with Polyphase-Network and DFT", Signal Processing, Vol.2, No.1, pp. 55-65, Jan. 1980.

[var83] P.Vary, and G.Wackersreuther, "A Unified Approach to Digital Polyphase Filter Banks", AEU, Band 37, 1983, Heft 1/2, pp.29-34.

[vet83] M.Vetterli, "Tree Structures for Orthogonal Transforms and Application to the Hadamard Transform", Signal Processing, Vol.5 , No.6, pp. 473-484, Nov.1983.

[vet84a] M.Vetterli, "Multi-Dimensional Sub-Band Coding: Some Theory and Algorithms", Signal Processing, Vol. 6, No.2, pp. 97-112, Feb. 1984.

[vet84b] M.Vetterli, and H.J.Nussbaumer, "Simple FFT and DCT Algorithms with Reduced Number of Operations", Signal Processing, Vol.6, No.4, pp.267-278, Aug. 1984.

[vet85a] M.Vetterli, "Fast 2-D Discrete Cosine Transform", Proc. of the 1985 IEEE Intl. Conf. on ASSP, Tampa, March 1985. pp.1538-1541.

[vet85b] M.Vetterli, and H.J.Nussbaumer, "Algorithmes de Transformée de Fourier et Cosinus Mono et Bidimensionnels", Annales des Telecommunications, Tome 40, Sept.- Oct. 1985, No.9-10, pp.466-476.

[vet85c] M.Vetterli, "Splitting a Signal into Subsampled Channels Allowing Perfect Reconstruction", Proc. of the IASTED Conf. on Applied Signal Processing and Digital Filtering, Paris, June 1985.

[vet85d] M.Vetterli, "FFT's of Signals with Symmetries and Applications" Proc. of MELECON-85, Madrid, Oct. 1985.

[vet85e] M.Vetterli, E.Debourse, and M.Kardan, "Fast Fourier Transforms on the TMS 320", Proc. of the Journées d'Electronique 1985, EPFL, Lausanne, Oct. 1985, pp.193-204.

[vet86a] M.Vetterli, and A.Ligtenberg, "A Discrete Fourier-Cosine Transform Chip", IEEE Trans. on Communications, Special Issue on VLSI in Telecommunications. January 1986.

[vet86b] M.Vetterli, "Filter Banks Allowing Perfect Reconstruction", à paraître dans Signal Processing, April 1985.

[vui83] J. Vuillemin, "A Combinatorial Limit to the Computing Power of VLSI Circuits", IEEE Trans. on Computers, Vol. C-32, No. 3, pp.294-300, March 1983.

[wac85] G.Wackersreuther, "On the Design of Filters for Ideal QMF and Polyphase Filter Banks", AEU, Band 39, 1985, Heft 2, pp.123-130.

[wan83] Z.Wang, and B.R.Hunt, The Discrete Cosine Transform: A New Version", IEEE Int. Conf. on Acoust., Speech, Signal Processing, ICASSP-83, Boston, 1983.

[win76] S.Winograd, "On Computing the Discrete Fourier Transform", Proc. Nat. Acad. Sci. USA, Vol. 73, pp. 1005-1006, April 1976.

[Yav68] R.Yavne, "An Economical Method for Calculating the Discrete Fourier Transform", AFIPS Proc., Vol.33, pp.115-125, 1968 Fall Joint Computer Conf., Washington.

2 Analyse de bancs de filtres

"On ne révolutionne pas en révolutionnant,
on révolutionne en "solutionnant""

Le Corbusier

Dans ce chapitre, nous allons analyser les bancs de filtres, en particulier ceux, très importants en pratique, qui permettent de reconstruire les signaux après le traitement par le banc de filtre.

Les bancs de filtres numériques ont une grande importance pratique, puisqu'ils sont utilisés pour le codage (par exemple, le codage de la parole en sous-bandes [crc76,est77,gal83]) ou dans les transmultiplexeurs (par exemple pour passer d'un multiplexage temporel à un multiplexage fréquentiel [bel74,com78,com82]).

Dans ce chapitre, nous poserons le problème de façon générale afin de développer une théorie unifiée des bancs de filtres. A cet effet, nous introduirons une notation matricielle que nous avons proposée simultanément avec d'autres chercheurs [ram84,smi85,vet85c,vet86b], ce qui montre bien la nécessité d'un tel formalisme. Ensuite, la notion de bancs de filtres polyphases introduite par Bellanger [bel74] pour les bancs de filtres modulés sera généralisée aux bancs de filtres quelconques.

Nous montrerons ensuite comment cette théorie générale permet d'obtenir des résultats nouveaux, en particulier pour la reconstruction parfaite des signaux originaux et la reconstruction sans repliement spectral ou sans diaphonie.

Notons qu'un banc de filtres utilisé en codage analyse un signal en N composantes, alors qu'un transmultiplexeur pour passer d'un multiplexage temporel à un multiplexage fréquentiel synthétise un signal à partir de N composantes. Ces deux types de bancs de filtres sont liés par dualisme, et nous verrons par la suite les similarités qui existent entre ces deux types et les systèmes qui en sont dérivés. Remarquons qu'en général, le signal aura une fréquence d'échantillonnage différente de celle des composantes. Typiquement, les fréquences diffèrent d'un facteur N, et

ceci donne lieu à des repliements spectraux indésirables.

Dans la suite, nous nous placerons du point de vue selon lequel le signal doit pouvoir être reconstitué à partir de ses composantes d'analyse, ou de façon duale, que les composantes doivent pouvoir être retrouvées à partir du signal synthétisé. Mathématiquement, ces deux problèmes sont équivalents, mais correspondent à deux problèmes réels bien différents. Le premier apparaît lors de l'analyse de signaux pour le codage, et dans ce cas, la suppression des repliements spectraux dus au sous-échantillonnage est impérative. Nous montrerons que la solution des filtres miroirs en quadrature [est77,gal84] peut être généralisée et améliorée. En particulier, nous verrons que la reconstruction parfaite est possible (ceci également pour le cas de plus de deux canaux), et des solutions utilisant des filtres à phase linéaire peuvent être utilisés. Le second cas apparaît par exemple dans les bancs de filtres réalisant le multiplexage en fréquence utilisé en télécommunications. Dans ce cas, le problème majeur est celui de la diaphonie entre les différents canaux. Etant donné l'équivalence mathématique qui sera démontrée entre les deux problèmes, la diaphonie peut dès lors être supprimée avec des méthodes similaires à celle utilisée pour la suppression des repliements spectraux, et plus généralement, toute solution applicable à l'un des problèmes est utilisable pour l'autre.

Dans notre traitement des bancs de filtres, nous insistons sur la propriété de reconstruction parfaite car elle représente un critère objectif pour l'évaluation de la qualité d'une conception. Une grande place est consacrée aux bancs de filtres avec sur- ou sous-échantillonnage d'un facteur égal au nombre des canaux (appelé sur- ou sous-échantillonnage critique). D'une part, c'est le cas théorique le plus intéressant puisqu'il représente le cas limite où une reconstruction est encore possible. D'autre part, c'est également le cas pratique le plus fréquent en codage et multiplexage (où la contrainte de bande passante minimum conduit toujours à des facteurs de sur- ou sous-échantillonnage critiques).

Dans la suite, nous utiliserons selon les besoins la représentation dans le domaine temporel et dans le domaine de la transformée en z des signaux et des filtres numériques. Nous ferons donc appel aux propriétés de la transformée en z, comme par exemple celle de convolution ou de modulation, tout en étant conscients qu'à toute transformée en z correspond un domaine de convergence hors duquel la transformée et par conséquent ses propriétés ne sont pas définies. Notons que toutes les variables utilisées par la suite sont supposées complexes à moins d'une spécification différente explicite.

Notre présentation sera organisée de la façon suivante. La section 2.1 donne les

définitions de base utiles à ce chapitre et présente les problèmes que l'on vise à résoudre. La section 2.2 introduit un formalisme matriciel adapté aux problèmes des bancs de filtres, et ceci dans deux plans de représentation: le plan de modulation (en quelque sorte la représentation fréquentielle) et le plan polyphase (similaire à une représentation temporelle). Ces représentations sont complémentaires et permettent ainsi d'éclairer les problèmes sous différents angles. La section 2.3 présente les résultats généraux obtenus grâce au formalisme précédemment introduit et la section 2.4 conclut le chapitre en rappelant les résultats obtenus les plus importants.

2.1 Définitions et présentation des problèmes

Tout d'abord, et ceci afin d'éviter des répétitions inutiles dans la suite de l'exposé, nous définirons quelques notions de base utiles à nos développements sur les bancs de filtres. Ces définitions sont en accord avec la littérature standard sur le sujet [opp75,rab75,kun80,bel81,crc83]. Ensuite, les deux problèmes centraux à ce chapitre seront présentés. En premier lieu, le problème de l'analyse d'un signal en N composantes par un banc de filtres et permettant la reconstruction du signal original sera posé. Ensuite, le problème de la synthèse d'un signal à partir de N composantes et permettant la reconstruction des composantes sera présenté.

2.1.1 Définitions

Dans les définitions qui vont suivre, on omettra sciemment le terme "numérique", puisque qu'il est tacitement supposé que tous les filtres et signaux impliqués sont numériques.

Définition D2.1: Un banc de filtres d'analyse de dimension N est constitué par N filtres $H_i(z)$ produisant à partir d'un seul signal d'entrée (avec transformée en z X(z)) N signaux de sortie (avec transformées en z $Y_i(z) = H_i(z)\ X(z)$, i = 0..N-1). Ces sorties sont souvent appelées canaux d'un banc de filtres. Un tel banc est représenté dans la figure 2.1.

Définition D2.2: Un banc de filtres de synthèse de dimension N est constitué par N filtres $H_i(z)$. Ceux-ci filtrent N signaux d'entrée (avec transformées en z $X_i(z)$), et leurs N sorties (avec transformées en z $Y_i(z) = H_i(z) \cdot X_i(z)$) sont ensuite sommées pour produire le signal de synthèse Y(z), aussi appelé signal de canal. Un tel banc est représenté dans la figure 2.2.

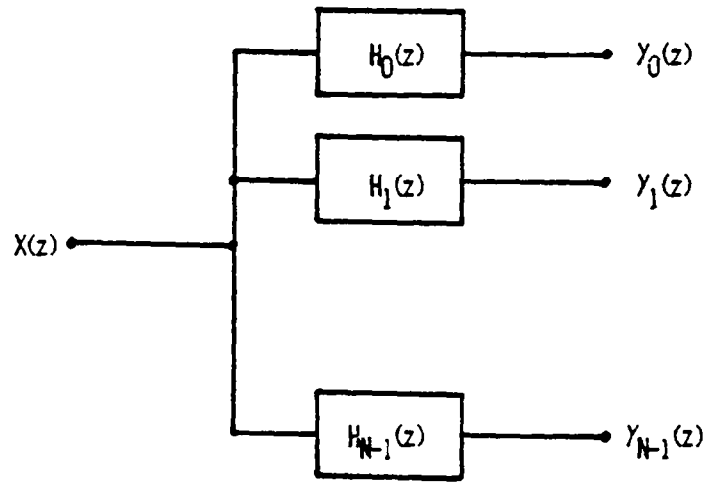

Figure 2.1: Banc de filtres d'analyse de taille N

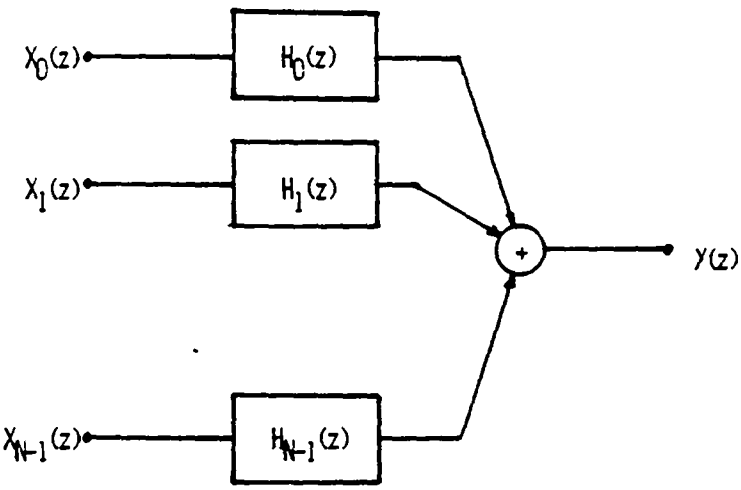

Figure 2.2: Banc de filtres de synthèse de taille N

Définition D2.3: Le sous-échantillonnage par un facteur N est l'opération linéaire qui transforme une suite x(n) en une suite x'(n) = x(nN). Dans le domaine de la transformée en z, on obtient la relation suivante [sha73,crc83]:

$$X'(z) = 1/N \sum_{k=0}^{N-1} X(W^k z^{1/N}) \qquad (2.1$$

avec $W = e^{-j2\pi/N}$ et $j = \sqrt{-1}$

Notons qu'en lieu et place de sous-échantillonnage on parle souvent de décimation.

Définition D2.4: Le suréchantillonnage par un facteur N est l'opération linéaire qui transforme une suite x(n) en une suite x'(n) tel que x'(nN) = x(n) et x'(n) zéro ailleurs. Dans le domaine de la transformée en z, X'(z) devient:

$$X'(z) = X(z^N) \tag{2.2}$$

Notons que cette définition est différente de la notion d'échantillonnage d'un signal analogique à une fréquence plus élevée que la fréquence de Nyquist.

Remarque R2.1: Le changement d'échantillonnage par un facteur rationnel P/Q est obtenu par mise en cascade d'un suréchantillonnage par un facteur P puis d'un sous-échantillonnage par un facteur Q ou inversément [crc83]. S'il n'y a pas d'opération (filtrage) effectuée entre le sur- et le sous-échantillonnage, l'ordre n'a pas d'influence.

Remarque R2.2: La mise en cascade d'un suréchantillonnage par un facteur N suivi d'un sous-échantillonnage par un facteur N ne modifie rien au signal si les changements d' échantillonnage sont en phase.

Remarque R2.3: La mise en cascade d'un sous-échantillonnage par un facteur N suivi d'un suréchantillonnage par un facteur N correspond à la modulation par une fonction m(n) donnée par [vet84a]:

$$m(n) = 1/N \cdot \sum_{k=0}^{N-1} W^{nk} \tag{2.3}$$

Donc, si un signal x(n) est d'abord sous-échantillonné par un facteur N puis suréchantillonné par un facteur N, alors la transformée en z du résultat x'(n) est égale à:

$$X'(z) = 1/N \cdot \sum_{k=0}^{N-1} X(W^k z) \tag{2.4}$$

Définition D2.5: Un banc de filtres d'analyse sous-échantillonné est un banc de filtres de dimension N dont les sorties sont sous-échantillonnées par un facteur N'. Si N=N', on parle d'un banc de filtres d'analyse à sous-échantillonnage critique [ram84], et lorsque N' n'est pas précisé, on supposera tacitement N'=N. Si N'<N, on parle de bancs de filtres d'analyse à sous-échantillonnage sous-critique, alors que pour N'>N, le sous-échantillonnage est qualifié de surcritique.

Notons que la signification des notions de sur- et sous-critique deviendra claire par

la suite. Remarquons cependant qu'une simple observation du nombre d'échantillons par unité de temps devant (signal d'entrée) et derrière (somme des signaux de sortie) le banc de filtres est déjà suffisante: si N'=N, le nombre d'échantillons est constant, si N'<N, leur nombre augmente, et enfin si N'>N, leur nombre diminue. Il est dès lors intuitivement compréhensible que dans les deux premiers cas, une reconstruction devrait être en général possible, alors qu'elle est impossible dans le dernier cas.

Définition D2.6: Un banc de filtres de synthèse suréchantillonné est un banc de filtres de dimension N dont les entrées sont suréchantillonnées par un facteur N'. De façon équivalente au banc d'analyse, on parle d'un banc de filtres de synthèse à suréchantillonnage critique si N'=N. Lorsque N' n'est pas précisé, on supposera tacitement N'=N. Si N'<N, on parle de bancs de filtres de synthèse à suréchantillonnage surcritique, alors que pour N'>N, le suréchantillonnage est qualifié de sous-critique.

Là aussi, une remarque similaire peut être faite à propos des notions d'échantillonnage sur- et sous-critique si l'on considère le nombre total d'échantillons par unité de temps devant et derrière le banc de filtre. Il est aisé de voir que N'=N garde et que N'>N augmente le nombre d'échantillons par unité de temps entre l'entrée et la sortie du banc de filtres, donc une reconstruction semble possible. Par contre, pour N'<N, le nombre d'échantillons diminue, rendant en général une reconstruction impossible.

Définition D2.7: Un système linéaire variant dans le temps de façon périodique (avec une période N égale à un multiple de la période d'échantillonnage) est un système linéaire dont les caractéristiques se répètent avec une période N. Il est donc caractérisé parfaitement par N réponses impulsionnelles successives correspondant à $\delta(0)$, $\delta(1)$,...$\delta(N-1)$ (ou bien leurs N transformées en z).

Remarque R2.4: Le sur- et le sous-échantillonnage font partie des systèmes linéaires variant dans le temps de façon périodique. Il en est donc de même pour tout système comportant ces opérations (et dont les autres composantes sont soit invariantes dans le temps, soit à variation périodique également).

Remarque R2.5: Si le quotient de la période de variation sur la période d'échantillonnage est égal à un nombre rationnel P/Q, alors le système échantillonné variera dans le temps avec période égale au plus petit commun multiple de P et Q.

Définition D2.8: Un banc de filtres modulés est un banc de filtres où les N filtres sont obtenus à partir d'un seul filtre prototype par modulation. Si les fonctions de

modulation ne sont pas précisées, on supposera par défaut qu'elles sont égales aux N racines de l'unité, c'est-à-dire que si $H_p(z)$ est la transformée en z du filtre prototype, alors les N filtres $H_i(z)$ sont donnés par:

$$H_i(z) = H_p(W^i z) \tag{2.5}$$

Remarque R2.6: Pour N=2 et un filtre prototype RIF à phase linéaire, le banc de filtres modulés est appelé banc de filtres miroirs en quadrature ("quadrature mirror filters" ou QMF) [cro76,gal84].

2.1.2 Bancs de filtres d'analyse avec reconstruction du signal

Le problème général de reconstruction parfaite est posé ci-dessous pour le cas des bancs de filtres d'analyse. Comme nous le verrons par la suite, si une reconstruction parfaite ne peut pas être obtenue, il peut être également intéressant de reconstruire le signal de façon à éviter tout repliement spectral du signal original dans le signal de sortie.

La classe des bancs de filtres d'analyse où la reconstruction du signal original doit être possible à partir des échantillons de sortie est importante en pratique. Ces bancs de filtres sont utilisés entre autres en codage de la parole en sous-bandes [crc76,cro76,gal83], une application où la propriété de reconstruction est cruciale. Dans les applications liées au codage, les bancs de filtres sont en général sous-échantillonnés de façon critique afin de ne pas accroître le nombre d'échantillons entre l'entrée et la sortie du banc de filtre, et de maintenir ainsi la même largeur de bande pour le signal d'entrée et pour l'ensemble des canaux.

Afin de reconstruire le signal original, les canaux sous-échantillonnés sont d'abord suréchantillonnés, puis interpolés et enfin sommés afin d'obtenir le signal reconstruit. Le banc de filtres d'analyse à sortie sous-échantillonnée est donc suivi d'un banc de filtres de synthèse avec une entrée suréchantillonnée. Un tel banc de filtres est représenté dans la figure 2.3.

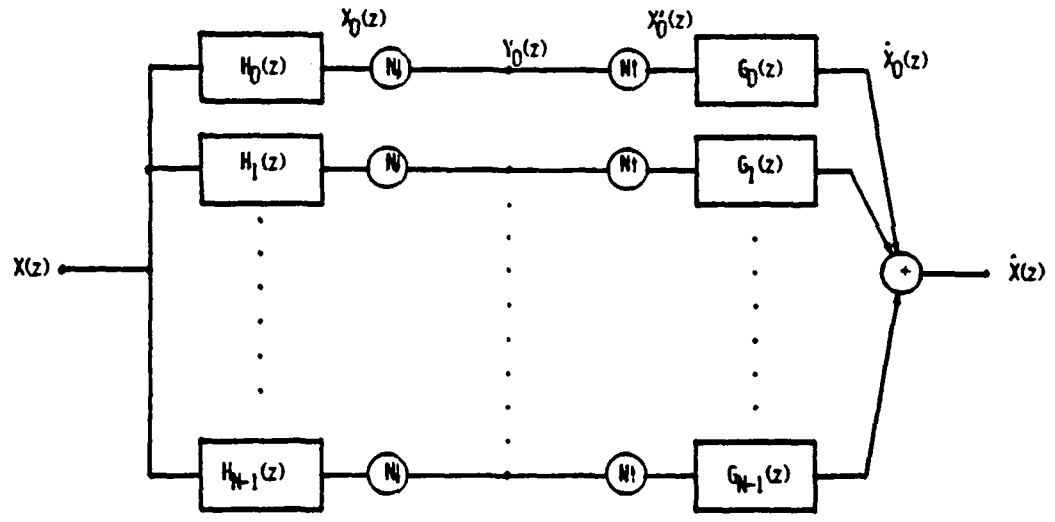

Figure 2.3: Banc de filtres d'analyse avec reconstruction du signal d'entrée

Nous allons définir la notion de reconstruction parfaite dans le cadre de bancs de filtres d'analyse:

Définition D2.9 Un banc de filtres d'analyse permettant la reconstruction parfaite du signal original possède une réponse impulsionnelle globale, correspondant à la mise en cascade des bancs d'analyse et de reconstruction (*entrée-sortie*), égale à un délai. Les bancs de filtres d'analyse et de synthèse ne doivent comporter que des filtres stables et causals (afin d'être réalisables en pratique).

Dans la figure 2.3, les notations suivantes sont utilisées:

$X(z)$: transformée en z du signal original

$X_i(z)$: transformée en z du i-ième signal après filtrage de $X(z)$ par un filtre d'analyse $H_i(z)$

$Y_i(z)$: transformée en z du signal dans le i-ième canal, obtenu après sous-échantillonnage de $X_i(z)$ par un facteur N

$X'_i(z)$: transformée en z du signal dérivé de $Y_i(z)$ par un suréchantillonnage par un facteur N

$\hat{X}_i(z)$: transformée en z du signal dérivé de $X'_i(z)$ par passage dans un filtre interpolateur $G_i(z)$

$\hat{X}(z)$: transformée en z du signal reconstruit égal à la somme des $\hat{X}_i(z)$

Les relations suivantes lient les grandeurs introduites ci-dessus:

$$X_i(z) = H_i(z)\, X(z) \tag{2.6}$$

$$Y_i(z) = 1/N \sum_{k=0}^{N-1} X_i(W^k z^{1/N}) \tag{2.7}$$

$$X'_i(z) = 1/N \sum_{k=0}^{N-1} X_i(W^k z) \tag{2.8}$$

$$\hat{X}_i(z) = G_i(z)\, X'_i(z) \tag{2.9}$$

$$\hat{X}(z) = \sum_{i=0}^{N-1} \hat{X}_i(z) \tag{2.10}$$

où (2.7) et (2.8) correspondent à (2.1) et (2.2) respectivement. Ainsi, et avec les relations (2.6) à (2.10), le problème des bancs de filtres d'analyse avec reconstruction est posé. Notons que l'exemple physique le plus connu d'un tel système est le codage en sous-bandes [gal83].

2.1.3 Bancs de filtres de synthèse avec reconstruction des signaux

Une autre classe importante de bancs de filtres est celle où N signaux sont multiplexés sur un seul canal à l'aide d'un banc de filtres de synthèse. Ces bancs de filtres sont par exemple utilisés en multiplexage fréquentiel [bel74,crc83], où il est évidemment nécessaire de pouvoir retrouver les N signaux d'entrée à partir du signal de canal. Dans ce cas, un problème majeur est l'apparition, par diaphonie, de signaux indésirables dans les sorties du système. Une contrainte importante est la largeur de bande du signal de canal. Le banc de filtres est donc suréchantillonné de façon critique afin de ne pas augmenter la largeur de bande utilisée par le signal de canal par rapport à celle utilisée par les signaux d'entrée.

Un tel système est représenté dans la figure 2.4. Les N signaux d'entrée sont tout d'abord suréchantillonnés par un facteur N, filtrés puis additionnés pour former le

signal de canal (banc de filtres de synthèse). Afin de reconstruire les signaux originaux, le signal de canal est filtré en N signaux qui sont alors sous-échantillonnés par un facteur N (banc de filtres d'analyse).

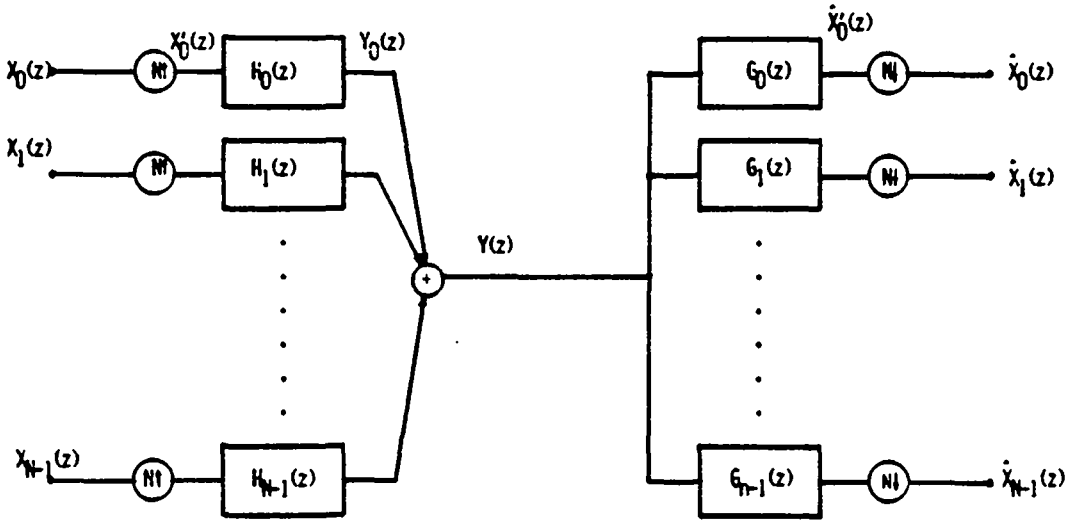

Figure 2.4: Banc de filtres de synthèse avec reconstruction des signaux d'entrée

La définition de la notion de reconstruction parfaite dans le cadre de bancs de filtres de synthèse est similaire à celle pour les bancs d'analyse (définition D2.9):

Définition D2.10 Un banc de filtres de synthèse permettant la reconstruction parfaite des signaux originaux possède une réponse impulsionnelle de l'entrée i à la sortie j égale à un délai si i=j et à zéro si i≠j. Les bancs de filtres de synthèse et d'analyse ne doivent comporter que des filtres stables et causals (afin d'être réalisables en pratique).

Les notations suivantes sont utilisées dans la figure 2.4:

$X_i(z)$: transformée en z du i-ième signal original

$X'_i(z)$: transformée en z du signal dérivé du i-ième signal original par suréchantillonnage par un facteur N

$Y_i(z)$: transformée en z du signal dérivé de $X'_i(z)$ après passage dans un filtre interpolateur $H_i(z)$

$Y(z)$: transformée en z du signal obtenu par sommation des $Y_i(z)$

$\hat{X}'_i(z)$: transformée en z du signal obtenu à partir de $Y(z)$ après filtrage par $G_i(z)$

$\hat{X}_i(z)$: transformée en z du i-ième signal reconstruit; obtenu après sous-échantillonnage de $\hat{X}'_i(z)$ par un facteur N

Notons que le système de la figure 2.4 est dual à celui de la figure 2.3, puisque les opérations sont simplement effectuées dans l'ordre inverse.

En utilisant les relations (2.1) et (2.2), on trouve que les grandeurs introduites ci-dessus sont liées par les relations suivantes:

$$X'_i(z) = X_i(z^N) \tag{2.11}$$

$$Y_i(z) = H_i(z)\, X'_i(z) \tag{2.12}$$

$$Y(z) = \sum_{i=0}^{N-1} Y_i(z) \tag{2.13}$$

$$\hat{X}'_i(z) = G_i(z)\, Y(z) \tag{2.14}$$

$$\hat{X}_i(z) = 1/N \sum_{k=0}^{N-1} \hat{X}'_i(W^k z^{1/N}) \tag{2.15}$$

On a donc posé, avec les relations (2.11) à (2.15), le problème des bancs de filtres de synthèse avec reconstruction, problème dont l'exemple physique le plus commun est la conversion du multiplexage temporel en multiplexage fréquentiel (et réciproquement) réalisée par un transmultiplexeur [crc83].

2.2 Formalisme matriciel

Les relations (2.6) à (2.10) ainsi que (2.11) à (2.15) gagnent à être représentées dans un formalisme matriciel. Le but de l'introduction de ce formalisme est d'établir les conditions de reconstruction parfaite ou de reconstruction sans repliement spectral (ou diaphonie) dans les bancs de filtres. Une transposition directe du problème donne lieu à ce que nous appellerons la représentation dans le plan de modulation. Celle-ci est similaire aux représentations introduites dans [ram84,smi85,vet85c,vet86b]. Une transposition quelque peu différente donne lieu à la représentation dans le plan polyphase. Ces deux représentations sont complémentaires, et il est possible de passer de l'une à l'autre par la transformée de Fourier. Par analogie, nous pouvons parler d'une représentation fréquentielle pour la première (elle fait appel aux versions modulées du signal original) et d'une représentation temporelle pour la seconde (elle décompose le signal original dans le temps).

Afin de développer ce formalisme matriciel, nous prendrons d'abord le cas des bancs de filtres d'analyse avec reconstruction (section 2.1.2). Le formalisme sera ensuite appliqué aux bancs de filtres de synthèse avec reconstruction (section 2.1.3).

2.2.1 Représentation dans le plan de modulation

Dans la suite, N est le facteur de sous-échantillonnage apparaissant dans un banc de filtres de dimension N, et W est la N-ième racine de l'unité. Comme un sous-échantillonnage par un facteur N fait apparaître N versions modulées du signal (y compris le signal lui-même), l'espace dans le plan de modulation est de dimension N, donc tous les vecteurs et matrices utilisées seront de dimension N et NxN. Le cas d'un sous-échantillonnage différent du nombre de filtres est relégué à la fin de cette section (voir Remarque R2.7). Notons que l'indice m indique une représentation dans le plan de modulation.

Introduisons les définitions suivantes:

Définition D2.11: Le vecteur des transformées en z modulées d'un signal (avec transformée en z X(z)), appelé $\mathbf{x}_m(z)$, est égal à:

$$\mathbf{x}_m(z) = [\ X(z),\ X(Wz),\ X(W^2z),\ \ldots\ ,\ X(W^{N-1}z)\ ]^T \qquad (2.16)$$

Donc, par exemple, $\mathbf{h}_{i,m}(z)$ est égal à:

$$\mathbf{h}_{i,m}(z) = [\ H_i(z),\ H_i(Wz),\ H_i(W^2z),\ .\ .\ .\ .\ ,\ H_i(W^{N-1}z)\ ]^T \tag{2.17}$$

Définition D2.12: La matrice des filtres $H_i(z)$ modulés, appelée $\mathbf{H}_m(z)$, est définie par:

$$\mathbf{H}_m(z) = [\ \mathbf{h}_{0,m}(z),\ \mathbf{h}_{1,m}(z),\ \mathbf{h}_{2,m}(z),\ .\ .\ .\ .\ ,\ \mathbf{h}_{N-1,m}(z)\ ]$$

$$= \begin{bmatrix} H_0(z) & H_1(z) & H_2(z) & . & . & . & . & . & H_{N-1}(z) \\ H_0(Wz) & H_1(Wz) & H_2(Wz) & . & . & . & . & . & H_{N-1}(Wz) \\ \vdots & \vdots & \vdots & & & & & & \vdots \\ H_0(W^{N-1}z) & H_1(W^{N-1}z) & H_2(W^{N-1}z) & . & . & . & . & . & H_{N-1}(W^{N-1}z) \end{bmatrix} \tag{2.18}$$

Introduisons également la notation suivante:

$$\mathbf{x}(z) = [\ X_0(z),\ X_1(z),\ X_2(z),\ .\ .\ .\ .\ ,\ X_{N-1}(z)\ ]^T \tag{2.19}$$

avec des définitions équivalentes pour $\mathbf{y}(z)$, $\mathbf{x}'(z)$, $\hat{\mathbf{x}}(z)$, $\mathbf{h}(z)$ et $\mathbf{g}(z)$.

Dès lors, les relations (2.6), (2.8) et (2.10) peuvent s'écrire:

$$\mathbf{x}(z) = \mathbf{h}(z) \cdot X(z) \tag{2.20}$$

$$\mathbf{x}'(z) = 1/N\ [\ \mathbf{H}_m(z)\ ]^T \cdot \mathbf{x}_m(z) \tag{2.21}$$

$$\hat{X}(z) = [\ \mathbf{g}(z)\ ]^T \cdot \mathbf{x}'(z) \tag{2.22}$$

A partir des relations (2.21) et (2.22), il est aisé de voir que le signal reconstruit est une fonction linéaire du signal original et des N-1 versions modulées de celui-ci. Que le système de la figure 2.3 soit linéaire est évident, puisqu'il qu'il ne comporte que des opérations linéaires (telles que filtrage et modulation). Néanmoins, il n'est pas rare de voir dans la littérature la notion de "distorsion non-linéaire" utilisée à propos des repliements spectraux apparaissant dans un tel système. Comme l'apparition de versions modulées du signal original est une caractéristique importante du système, nous allons introduire explicitement des fonctions de transfert $M_i(z)$ pour chacune des composantes $X(W^iz)$. Dans ces conditions, le signal de sortie (figure 2.5) devient:

$$\hat{X}(z) = \sum_{k=0}^{N-1} M_k(z) \cdot X(W^k z) \qquad (2.23)$$

où $M_k(z)$ est la k-ième composante du vecteur $\mathbf{m}(z)$ dont la définition suit ci-dessous.

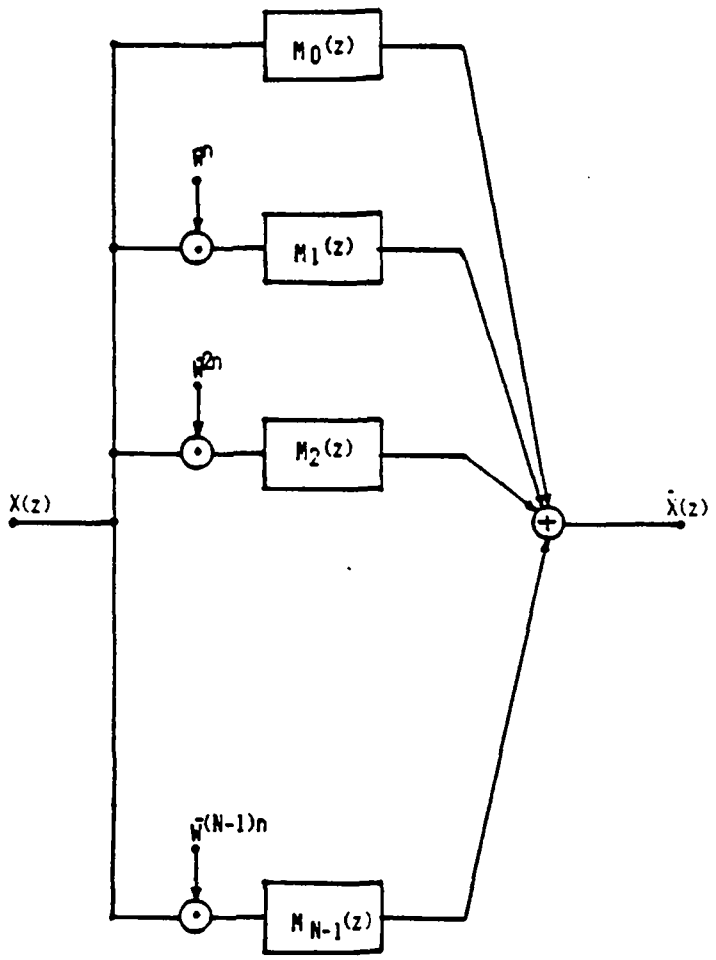

Figure 2.5: Représentation dans le plan de modulation du problème des bancs de filtres d'analyse avec reconstruction

Introduisons un vecteur $\mathbf{m}(z)$ défini de façon similaire à $\mathbf{x}(z)$ en (2.19) et donné par:

$$\mathbf{m}(z) = 1/N \quad \mathbf{H}_m(z) \cdot \mathbf{g}(z) \qquad (2.24)$$

Alors, la relation (2.23) devient:

$$\hat{X}(z) = [\ \mathbf{m}(z)\ ]^T \cdot \mathbf{x}_m(z) \qquad (2.25)$$

La reconstruction sera parfaite si:

$$\mathbf{m}(z) = [\ z^{-k}\ 0\ 0\ 0\ .\ .\ .\ .\ 0\ ]^T \tag{2.26}$$

puisqu'alors:

$$\hat{X}(z) = z^{-k}\ X(z) \tag{2.27}$$

D'une manière similaire, on parlera de reconstruction sans repliement spectral (mais avec modification spectrale) lorsque:

$$\mathbf{m}(z) = [\ S(z)\ 0\ 0\ .\ .\ .\ 0\ ]^T \tag{2.28}$$

puisqu'alors seule la version filtrée du signal original apparaît à la sortie et que toutes les versions modulées sont éliminées:

$$\hat{X}(z) = S(z)\ X(z) \tag{2.29}$$

Jusqu'à présent, seul le cas d'un sous-échantillonnage critique du banc de filtres d'analyse a été considéré, entre autres parce qu'il est le plus intéressant pour le codage. La remarque ci-dessous donne les modifications nécessaires à la résolution du problème de sous-échantillonnage sur- ou sous-critique.

Remarque R2.7: Lorsque le banc de filtres d'analyse n'est pas sous-échantillonné de façon critique, c'est-à-dire que $N \neq N'$ (N étant le nombre de filtres et N' le facteur de sous-échantillonnage), nous remarquons que:
- la matrice $\mathbf{H}_m(z)$ a N' lignes et N colonnes
- les vecteurs $\mathbf{x}_m(z)$, $\mathbf{h}_{i,m}(z)$ et $\mathbf{m}(z)$ sont de longueur N'
- les vecteurs $\mathbf{x}(z)$, $\mathbf{y}'(z)$, $\mathbf{x}'(z)$, $\hat{\mathbf{x}}_i(z)$, $\mathbf{h}(z)$ et $\mathbf{g}(z)$ sont de longueur N
- W est la N'-ième racine de l'unité

Dans la suite, nous nous référerons aux relations (2.24-2.25) comme à la représentation dans le plan de modulation du problème des bancs de filtres d'analyse avec reconstruction du signal. En quelque sorte, ces relations donnent une représentation fréquentielle du problème des bancs de filtres comportant un sous-échantillonnage.

2.2.2 Représentation dans le plan polyphase

Une représentation complémentaire de celle qui est basée sur le plan de modulation permet d'observer les effets temporels des bancs de filtres d'analyse avec reconstruction. Comme base de représentation, on utilisera non pas le signal d'entrée et ses N-1 versions modulées, mais ce que nous conviendrons d'appeler la représentation polyphase du signal (qui sera indiquée par l'indice p).

La dualité vient du fait qu'un système variant dans le temps avec une période N peut être représenté soit par les fonctions de transfert du signal et de ses N-1 versions modulées, soit par N réponses impulsionnelles aux temps 0..N-1 (voir également la définition D2.7 et la remarque R2.4).

La représentation dans le plan polyphase est une généralisation des bancs de filtres polyphases introduits par Bellanger pour les transmultiplexeurs [bel74]. Dans ce cas, le banc de filtres est obtenu par modulation à partir d'un seul filtre prototype, et peut être interprété comme N filtres polyphases suivis d'une transformée de Fourier. Dans notre cas, nous ne nous restreignons pas aux bancs de filtres modulés, mais nous considérons des filtres arbitraires. Chaque filtre peut être décomposé en N filtres polyphases, et de ce fait, un banc de N filtres sous-échantillonné de façon critique peut être interprété comme N^2 filtres polyphases. Cette généralisation originale nous semble importante pour l'interprétation des bancs de filtres numériques. Même si la représentation polyphase peut paraître lourde au premier abord, elle permettra un traitement concis de nombreux problèmes par la suite, et elle éclairera également ces problèmes sous un jour nouveau.

Rappelons d'abord brièvement la transformation d'un filtre général en filtre polyphase. Ce premier pas est similaire à la dérivation dans [bel74]. Cette transformation vise à décomposer un filtre en différents sous-filtres qui ne comportent que des puissances N-ièmes de z et qui ont chacun un facteur de phase différent (d'où le terme polyphase). Cette représentation est fort utile afin d'exprimer un filtre à deux fréquences d'échantillonnage différant d'un facteur N.

Prenons la fonction de transfert d'un filtre général $H_p(z)$ à réponse impulsionnelle infinie (RII) qui peut être représenté de la façon suivante:

$$H_p(z) = \frac{N(z)}{D(z)} = \frac{\sum_{j=0}^{J} n_j z^{-j}}{\sum_{i=0}^{I} d_i z^{-i}} = c \cdot \frac{\prod_{j=0}^{J-1} (1 - q_j z^{-1})}{\prod_{i=0}^{I-1} (1 - p_i z^{-1})} \qquad (2.30)$$

où le facteur constant c est égal à n_0/d_0. Utilisons maintenant l'identité suivante, basée sur la factorisation de polynômes:

$$\frac{1}{1-p_i z^{-1}} = \frac{\sum_{l=0}^{N-1} (p_i z^{-1})^l}{1-p_i^N z^{-N}} \qquad (2.31)$$

Dans la relation (2.31), N est arbitraire (en fait, N est choisi égal au facteur d'échantillonnage qui suit ou précède le filtre $H_p(z)$). L'équivalence exprimée par (2.31) remplace un pôle en p_i par N pôles en $W^k p_i$, k=0..N-1, ainsi que N-1 zéros en $W^k p_i$, k=1..N-1 (voir également la figure 2.6). Avec (2.31), $H_p(z)$ peut être modélisé par un filtre $H'_p(z)$ de structure différente, mais de même fonction de transfert, avec:

$$H'_p(z) = c \cdot \frac{\prod_{j=0}^{J-1} (1 - q_j z^{-1}) \cdot \prod_{i=0}^{I-1} \left(\sum_{l=0}^{N-1} (p_i z^{-1})^l \right)}{\prod_{i=0}^{I-1} (1 - p_i^N z^{-N})}$$

$$= \frac{\sum_{j=0}^{J+IN-I} \alpha_j z^{-j}}{\sum_{i=0}^{I} \beta_{Ni} z^{-Ni}} = \frac{N'(z)}{D'(z^N)} \qquad (2.32)$$

où le dénominateur ne comporte que des puissances multiples de N. Ce fait est explicité par la notation z^N dans $D'(z^N)$. Notons que la stabilité du filtre n'a pas été affectée par cette transformation, mais que les propriétés numériques peuvent être différentes.

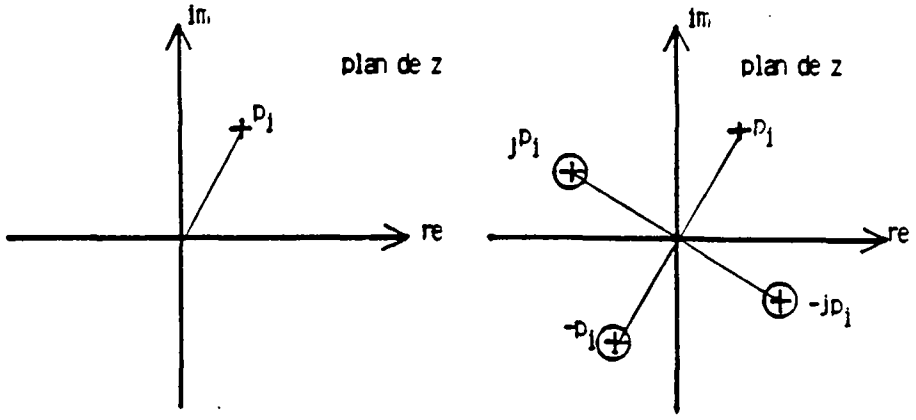

Figure 2.6: Expansion d'un pôle en N pôles et N-1 zéros selon la relation (2.31) (ici pour N=4)

Décomposons maintenant le numérateur de $H'_p(z)$ en filtres polyphases de la façon suivante:

$$H''_p(z) = \frac{\sum_{k=0}^{N-1} z^{-k} H_{p,k}(z^N)}{D'(z^N)} \tag{2.33}$$

avec:

$$H_{p,k}(z^N) = a_k + a_{k+N} z^{-N} + \ldots\ldots + a_{k+LN} z^{-LN}$$

$$= \sum_{l=0}^{L_k} a_{Nl+k} z^{-Nl} \qquad L_k = \lfloor (J+IN-I-k)/N \rfloor \tag{2.34}$$

Rappelons que $H_p(z)$, $H'_p(z)$ et $H''_p(z)$, quoique de structures différentes, ont tous trois la même réponse impulsionnelle (donc la même fonction de transfert). L'avantage de la représentation en (2.33) est qu'un filtre n'ayant que des puissances N-ième de z (comme c'est le cas des différentes composantes polyphases dans (2.33)) est aisément représenté à deux fréquences d'échantillonnage qui ont un rapport égal à N. Par exemple, un filtre $H(z^N)$ précédant un sous-échantillonnage de N peut être remplacé par un filtre H'(z)=H(z) à la suite du sous-échantillonnage. Comme exemple, la figure 2.7 montre comment un filtre général précédant un sous-échantillonnage est d'abord décomposé en composantes polyphases (comme en (2.33)), puis comment celles-ci sont passées de l'autre côté du sous-échantillonnage. Notons que la section d'entrée de la figure 2.7c (composée de délais suivis de sous-échantillonnage) est souvent représentée comme un distributeur d'échantillons avec un sens de rotation contraire à celui des aiguilles d'une montre [crc83].

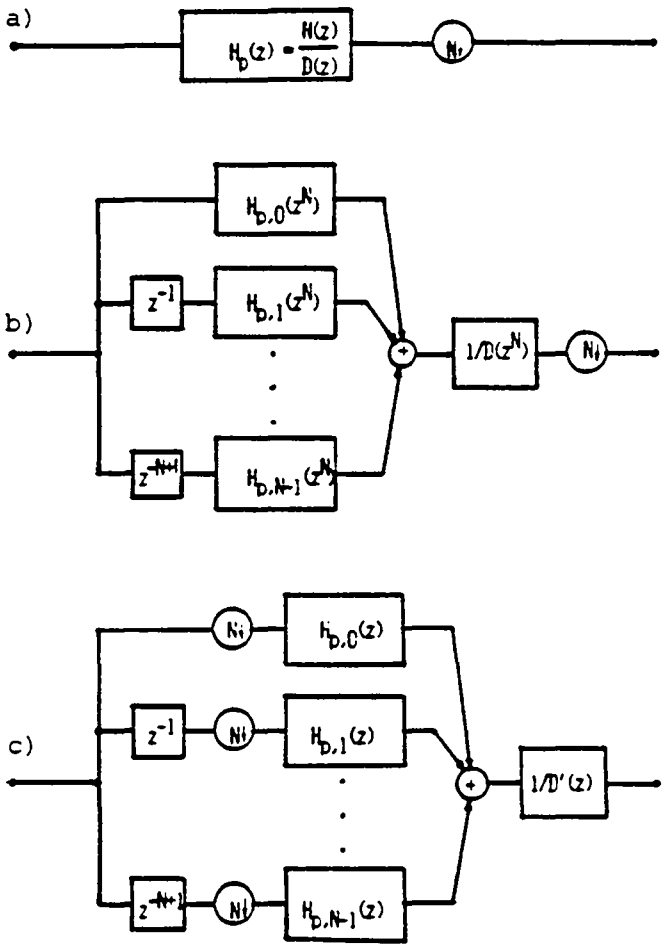

Figure 2.7: Transformation d'un filtre général en filtres polyphases sous-échantillonnés

a) Filtre original suivi d'un sous-échantillonnage

b) Filtre décomposé en composantes polyphases avec un dénominateur commun en z^N

c) Filtre exprimé à la fréquence d'échantillonnage réduite

Les relations (2.30) à (2.34) sont équivalentes à la dérivation des filtres polyphases dans [bel74], et nous les avons rappelées ci-dessus pour plus de clarté dans notre développement. Résumons les relations (2.30) à (2.34) dans la définition suivante:

Définition D2.13: La représentation polyphase d'ordre N d'un filtre général comporte un dénominateur n'ayant que des puissances N-ième de z^{-1} et un numérateur constitué par la somme de N filtres n'ayant également que des puissances multiples de N, mais dont chacun est retardé d'un délai différent allant de 0 à -N+1.

La généralisation de la notion de bancs de filtres polyphases aux bancs de filtres quelconques se base sur la définition suivante:

Définition D2.14: La matrice des filtres polyphases, appelée $\mathbf{H_p}(z)$, est définie comme suit:

$$\mathbf{H_p}(z)=\begin{bmatrix} \frac{H_{0,0}(z^N)}{D_0(z^N)} & \frac{H_{1,0}(z^N)}{D_1(z^N)} & \cdots & \frac{H_{N-1,0}(z^N)}{D_{N-1}(z^N)} \\ \frac{z^{-1}H_{0,1}(z^N)}{D_0(z^N)} & \frac{z^{-1}H_{1,1}(z^N)}{D_1(z^N)} & \cdots & \frac{z^{-1}H_{N-1,1}(z^N)}{D_{N-1}(z^N)} \\ \vdots & \vdots & & \vdots \\ \frac{z^{-N+1}H_{0,N-1}(z^N)}{D_0(z^N)} & \frac{z^{-N+1}H_{1,N-1}(z^N)}{D_1(z^N)} & \cdots & \frac{z^{-N+1}H_{N-1,N-1}(z^N)}{D_{N-1}(z^N)} \end{bmatrix} \quad (2.35)$$

Il est aisé de reconnaître que les élements d'une colonne de la matrice $\mathbf{H_p}(z)$ sont simplement les N filtres polyphases qui apparaissent lors de la décomposition polyphase d'un filtre $H_i(z)$ comme dans (2.33).

Une représentation plus explicite de la matrice $\mathbf{H_p}(z)$ est obtenue par la factorisation suivante:

$$\mathbf{H_p}(z) = \mathbf{I_d}(z) \cdot \mathbf{N_p}(z^N) \cdot \mathbf{D_p}(z^N) \quad (2.36)$$

où:

$$\mathbf{I_d}(z) = \begin{bmatrix} 1 & & & & \\ & z^{-1} & & & \\ & & z^{-2} & & \\ & & & \ddots & \\ & & & & z^{-N+1} \end{bmatrix} \quad (2.37)$$

$$\mathbf{N}_p(z^N)= \begin{bmatrix} H_{0,0}(z^N) & H_{1,0}(z^N) & \cdots & H_{N-1,0}(z^N) \\ H_{0,1}(z^N) & H_{1,1}(z^N) & \cdots & H_{N-1,1}(z^N) \\ \vdots & \vdots & & \vdots \\ H_{0,N-1}(z^N) & H_{1,N-1}(z^N) & \cdots & H_{N-1,N-1}(z^N) \end{bmatrix} \tag{2.38}$$

$$\mathbf{D}_p(z^N)= \begin{bmatrix} \frac{1}{D_0(z^N)} & & & 0 \\ & \frac{1}{D_1(z^N)} & & \\ & & \ddots & \\ 0 & & & \frac{1}{D_{N-1}(z^N)} \end{bmatrix} \tag{2.39}$$

Donc, $\mathbf{I}_d(z)$ représente le délai commun associé avec chaque ligne de la matrice $\mathbf{H}_p(z)$, $\mathbf{N}_p(z^N)$ la décomposition polyphase des numérateurs des filtres $H_i(z)$ et $\mathbf{D}_p(z^N)$ le dénominateur commun associé avec chaque colonne de $\mathbf{H}_p(z)$.

Cette factorisation de $\mathbf{H}_p(z)$ sera fort utile par la suite. D'une part, elle donne une interprétation physique du banc de filtres en séparant les délais et les dénominateurs communs des numérateurs polyphases. D'autre part, le traitement mathématique de la matrice $\mathbf{H}_p(z)$ sera grandement simplifié par cette factorisation.

Notons encore qu'un avantage de la représentation polyphase par rapport à la représentation dans le plan de modulation est l'absence de redondance dans la première des deux. Prenons par exemple le cas où les filtres du banc sont RIF. Dans ce cas, un coefficient donné d'un filtre apparaît dans un seul élément de la matrice $\mathbf{H}_p(z)$, tandis qu'il apparaît dans tous les éléments d'une colonne de la matrice $\mathbf{H}_m(z)$, c'est-à-dire N fois. Ceci se vérifie aisément en considérant les définitions de $\mathbf{H}_p(z)$ et de $\mathbf{H}_m(z)$.

Finalement, introduisons la représentation polyphase d'un signal:

Définition D2.15: Le vecteur de représentation polyphase d'un signal (avec transformée en z X(z)), appelé $\mathbf{x}_p(z)$, est formé des composantes suivantes:

$$[\ \mathbf{x}_p(z)\ ]_k = z^{-k}\ X_{p,k}(z^N) \tag{2.40}$$

où $X_{p,k}(z^N)$ est la transformée en z du signal $x_{p,k}(n)$. Ce dernier est obtenu à partir du signal x(n) de la façon suivante:

$$\begin{aligned} x_{p,k}(n) &= x(n+k) \qquad n = lN,\quad l = -\infty\ldots\infty \\ &= 0 \qquad\qquad \text{ailleurs} \end{aligned} \tag{2.41}$$

Cette définition est similaire à la définition des filtres polyphases en (2.33).

Notons les relations suivantes qui lient les représentations polyphases aux représentations dans le plan de modulation:

$$\mathbf{x}_p(z) = 1/N \cdot \mathbf{F} \cdot \mathbf{x}_m(z) \tag{2.42}$$

$$\mathbf{H}_p(z) = 1/N \cdot \mathbf{F} \cdot \mathbf{H}_m(z) \tag{2.43}$$

où $\mathbf{F}$ est la matrice de Fourier définie par:

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)(N-1)} \end{bmatrix} \tag{2.44}$$

Il est caractéristique que les deux représentations soient liées par une transformée de Fourier puisqu'elles correspondent à la représentation temporelle (polyphase) et fréquentielle (modulation) d'un même problème.

En utilisant la relation réciproque à (2.42), c'est-à-dire:

$$\mathbf{x}_m(z) = N \cdot \mathbf{F}^{-1} \cdot \mathbf{x}_p(z) = \mathbf{F}^* \cdot \mathbf{x}_p(z) \tag{2.45}$$

nous pouvons écrire (2.25) comme suit:

$$\hat{X}(z) = 1/N \cdot [\ \mathbf{H}_m(z) \cdot \mathbf{g}(z)\ ]^T \cdot \mathbf{F}^* \cdot \mathbf{x}_p(z) \tag{2.46}$$

Notons que, à partir de (2.43), on peut écrire:

$$[\ \mathbf{H}_m(z)\ ]^T \cdot \mathbf{F}^* = N \cdot [\ \mathbf{H}_p(z)\ ]^T \cdot \mathbf{J} \tag{2.47}$$

où nous avons utilisé le fait que $\mathbf{F}^2 = [\mathbf{F}^*]^2 = N \cdot \mathbf{J}$, et:

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 1 \\ 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & & \iddots & \vdots \\ 0 & 0 & 1 & & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{2.48}$$

Donc, la relation (2.46) devient:

$$\hat{X}(z) = [\ \mathbf{H}_p(z) \cdot \mathbf{g}(z)\ ]^T \cdot \mathbf{J} \cdot \mathbf{x}_p(z) \tag{2.49}$$

Par la suite, on se référera à la relation (2.49) comme à la représentation dans le plan polyphase du problème des bancs de filtres d'analyse avec reconstruction. Notons que la matrice de permutation $\mathbf{J}$ correspond à la distribution des échantillons dans un banc de filtres polyphases, et que cette distribution se fait dans le sens contraire à la rotation des aiguilles d'une montre [crc83].

Appelons $P_k(z)$ la réponse du système à une impulsion unitaire au temps k. Alors le vecteur $\mathbf{p}(z)$ des N réponses aux impulsions aux temps k=0...N-1 est égal à:

$$[\ \mathbf{p}(z)\ ]^T = [\ \mathbf{H}_p(z) \cdot \mathbf{g}(z)\ ]^T \cdot \mathbf{J} \cdot \mathbf{I}_d(z) \tag{2.50}$$

puisque les colonnes de $\mathbf{I}_d(z)$ correspondent bien à des impulsions aux temps 0,1,...N-1 dans une représentation polyphase. La reconstruction parfaite est obtenue si le vecteur $\mathbf{p}(z)$ est égal à:

$$\mathbf{p}(z) = [\ z^{-k} \quad z^{-k-1} \quad \ldots \quad z^{-k-N+1}\ ]^T \tag{2.51}$$

La relation (2.50) correspond à (2.26). Le système est invariant dans le temps si toutes les réponses sont identiques, donc si $\mathbf{p}(z)$ est égal à:

$$\mathbf{p}(z) = [\ S(z) \quad z^{-1}\cdot S(z) \quad \ldots \quad z^{-N+1}\cdot S(z)\ ]^T \tag{2.52}$$

Notons que l'invariance dans le temps correspond à l'absence de modulation (donc de repliements spectraux), d'où il ressort que (2.52) est équivalent à (2.28).

Introduisons un vecteur $\mathbf{p}'(z)$ obtenu à partir de $\mathbf{p}(z)$ en simplifiant les délais et la permutation de la façon suivante:

$$\mathbf{p}'(z) = \mathbf{J}\cdot [\ \mathbf{I}_d(z)\ ]^{-1}\cdot \mathbf{p}(z) = \mathbf{H}_p(z)\cdot \mathbf{g}(z) \tag{2.53}$$

Donc, $P'_k(z)$ (qui est le k-ième élement de $\mathbf{p}'(z)$) représente la transmission d'une impulsion au temps N-k, sans tenir compte du délai implicite de z^{-N+k}. A l'aide de $\mathbf{p}'(z)$, on peut représenter le banc de filtres avec reconstruction comme dans la figure 2.8, qui est l'équivalent polyphase de la représentation dans le plan de modulation de la figure 2.5. Dans la figure 2.8, la section d'entrée réalisée par des délais suivis d'un sous/suréchantillonnage de N (qui garde uniquement les échantillons à des instants multiples de N et remplace les autres par zéro) est, comme déjà remarqué auparavant, équivalente à un distributeur (en sens contraire aux aiguilles d'une montre) suivi d'un suréchantillonnage de N.

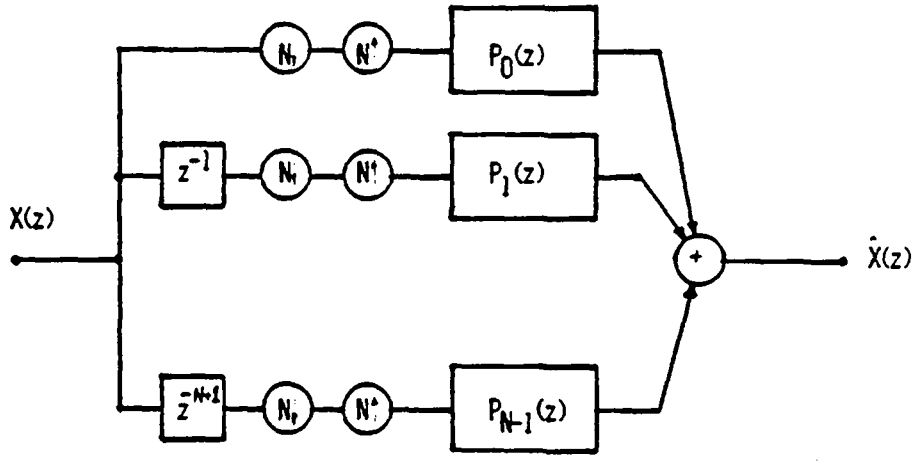

Figure 2.8: Représentation dans le plan polyphase du problème des bancs de filtres d'analyse avec reconstruction

Notons que si le suréchantillonnage n'est pas critique (N'≠N), alors une remarque similaire à R2.7 est applicable:

Remarque R2.8: Si le sous-échantillonnage n'est pas critique, alors les dimensions des matrices et vecteurs dans le plan polyphase sont modifiées de la façon suivante:
- la matrice $\mathbf{H}_p(z)$ a N' lignes et N colonnes
- les vecteurs $\mathbf{x}_p(z)$, $\mathbf{h}_{i,p}(z)$ et $\mathbf{p}(z)$ sont de longueur N'
- les vecteurs $\mathbf{x}_i(z)$, $\mathbf{y}'(z)$, $\mathbf{x}'(z)$, $\hat{\mathbf{x}}_i(z)$, $\mathbf{h}(z)$ et $\mathbf{g}(z)$ sont de longueur N
- W est la N'-ième racine de l'unité

Avec la représentation polyphase introduite ci-dessus, on a obtenu en quelque sorte une vue des bancs de filtres dans le domaine temporel.

2.2.3 Résumé et synthèse du formalisme matriciel des bancs de filtres

Comme le formalisme matriciel introduit ci-dessus pour les bancs de filtres d'analyse avec reconstruction est central au chapitre 2, nous rappelons les équations les plus importantes ci-dessous, et ceci en présentant côte à côte la représentation dans le plan polyphase (indice p) et dans le plan de modulation (indice m).

Les deux représentations du signal d'entrée d'un banc de filtre comportant un sous-échantillonnage de N sont liées par les relations suivantes:

$$\mathbf{x}_m(z) = \mathbf{F} \cdot \mathbf{J} \cdot \mathbf{x}_p(z) \tag{2.54}$$

$$\mathbf{x}_p(z) = 1/N \cdot \mathbf{F} \cdot \mathbf{x}_m(z) \tag{2.55}$$

où nous avons utilisé le fait que:

$$\mathbf{F}^{-1} = 1/N \cdot \mathbf{F} \cdot \mathbf{J} = 1/N \cdot \mathbf{J} \cdot \mathbf{F} \tag{2.56}$$

La sortie du banc de filtre de synthèse $\hat{X}(z)$ peut être écrite en fonction de $\mathbf{x}_m(z)$ et $\mathbf{H}_m(z)$ ou de $\mathbf{x}_p(z)$ et $\mathbf{H}_p(z)$ comme:

$$\hat{X}(z) = 1/N \cdot [\, H_m(z) \cdot g(z) \,]^T \cdot x_m(z) \tag{2.57}$$

$$\hat{X}(z) = [\, H_p(z) \cdot g(z) \,]^T \cdot J \cdot x_p(z) \tag{2.58}$$

Les deux représentations des matrices de filtres sont liées comme suit:

$$H_m(z) = F \cdot J \cdot H_p(z) \tag{2.59}$$

$$H_p(z) = 1/N \cdot F \cdot H_m(z) \tag{2.60}$$

Le problème de la reconstruction parfaite ou de la reconstruction sans repliement spectral est succinctement formulé par:

$$\begin{aligned} H_m(z) \cdot g(z) &= [\, z^{-k} \; 0 \; 0 \; \ldots \; 0 \,]^T \text{ reconstruction parfaite} \\ &= [\, S(z) \; 0 \; 0 \; \ldots \; 0 \,]^T \text{ reconstruction sans repliement spectral} \end{aligned} \tag{2.61}$$

$$\begin{aligned} H_p(z) \cdot g(z) &= [\, z^{-k} \; z^{-k} \; \ldots \; z^{-k} \,]^T \text{ reconstruction parfaite} \\ &= [\, S(z) \; S(z) \; \ldots \; S(z) \,]^T \text{ reconstruction sans repliement spectral} \end{aligned} \tag{2.62}$$

Les relations (2.54) à (2.62) résument parfaitement le problème des bancs de filtres d'analyse avec reconstruction, et nous nous y référerons continuellement dans la suite lors de la recherche de solutions au dit problème.

2.2.4 Application du formalisme matriciel aux bancs de filtres de synthèse

Le formalisme matriciel introduit ci-dessus peut également être appliqué aux bancs de filtres de synthèse avec reconstruction des signaux (voir section 2.1.3 et figure 2.4). Notons que la puissance du formalisme matriciel introduit dans les sections 2.2.1 à 2.2.3 devient évident, puisque le cas des bancs de filtres de synthèse se traite maintenant en quelques équations.

Récrivons d'abord le i-ième signal de sortie (avec transformée en z $\hat{X}_i(z)$) comme suit (d'après 2.11-2.15):

$$\hat{X}_i(z) = 1/N \sum_{k=0}^{N-1} G_i(W^k z^{1/N}) \; Y(W^k z^{1/N})$$

$$= 1/N \sum_{k=0}^{N-1} G_i(W^k z^{1/N}) \sum_{j=0}^{N-1} H_j(W^k z^{1/N}) \; X_j(z) \qquad (2.63)$$

Remplaçons d'abord z par z^N afin de simplifier la notation (ceci revient à exprimer toutes les transformées en z dans la fréquence du signal de canal plutôt que dans celle des signaux d'entrée). Ensuite, nous remarquons que la relation (2.63) s'exprime aisément à l'aide des matrices de filtres modulés introduits à la section 2.2.1, et ceci comme suit:

$$\hat{\mathbf{x}}(z^N) = 1/N \; [\; \mathbf{G}_m(z) \;]^T \cdot \mathbf{H}_m(z) \cdot \mathbf{x}(z^N) \qquad (2.64)$$

où:

$$\mathbf{x}(z^N) = [\; X_0(z^N) \; X_1(z^N) \; . \; . \; . \; X_{N-1}(z^N) \;]^T \qquad (2.65)$$

ainsi qu'une définition similaire pour $\hat{\mathbf{x}}(z^N)$. Introduisons une matrice $\mathbf{T}(z)$ définie comme suit:

$$\mathbf{T}(z) = 1/N \; [\; \mathbf{G}_m(z) \;]^T \cdot \mathbf{H}_m(z) \qquad (2.66)$$

Notons que $\mathbf{T}(z)$ est une fonction de z^N puisque chaque élément est égal à la somme de N versions modulées. Avec (2.66), la relation (2.64) devient:

$$\hat{\mathbf{x}}(z^N) = \mathbf{T}(z) \cdot \mathbf{x}(z^N) \qquad (2.67)$$

Donc, l'élément i,j de la matrice $\mathbf{T}(z)$ indique comment le signal d'entrée j est transmis à la sortie i d'un banc de filtres de synthèse avec reconstruction. De ce fait, nous appellerons $\mathbf{T}(z)$ la matrice de transition entre les entrées et les sorties d'un tel banc de filtres. Les relations (2.66) et (2.67) sont la représentation dans le plan de modulation du problème, puisqu'elles font appel aux matrices de filtres modulés.

Afin d'obtenir la représentation dans le plan polyphase, nous appliquerons la transformation suivante:

$$\begin{aligned} \mathbf{T}(z) &= 1/N \; [\; \mathbf{G}_m(z) \;]^T \cdot \mathbf{F}^{-1} \cdot \mathbf{F} \cdot \mathbf{H}_m(z) \\ &= 1/N \; [\; \mathbf{G}_m(z) \;]^T \cdot 1/N \cdot \mathbf{F} \cdot \mathbf{J} \cdot N \cdot \mathbf{H}_p(z) \\ &= [\; \mathbf{G}_p(z) \;]^T \cdot \mathbf{J} \cdot \mathbf{H}_p(z) \end{aligned} \tag{2.68}$$

où nous avons utilisé les relations (2.56) et (2.60). Donc, nous pouvons récrire (2.64) comme suit:

$$\hat{\mathbf{x}}(z^N) = [\; \mathbf{G}_p(z) \;]^T \cdot \mathbf{J} \cdot \mathbf{H}_p(z) \cdot \mathbf{x}(z^N) \tag{2.69}$$

Ceci est la représentation dans le plan polyphase des bancs de filtres de synthèse avec reconstruction.

Similairement aux relations (2.61-2.62), on peut parler de reconstruction parfaite si:

$$\mathbf{T}(z) = \begin{bmatrix} z^{-Nk} & & & \\ & z^{-Nk} & & \\ & & \ddots & \\ & & & z^{-Nk} \end{bmatrix} \tag{2.70}$$

Si le délai est différent d'un multiple de N, la reconstruction parfaite peut être obtenue, mais alors le sous-échantillonnage à la sortie n'est plus en phase, et doit être décalé dans le temps de façon appropriée. La reconstruction sans diaphonie est obtenue si:

$$\mathbf{T}(z) = \begin{bmatrix} S_0(z^N) & & & \\ & S_1(z^N) & & \\ & & \ddots & \\ & & & S_{N-1}(z^N) \end{bmatrix} \tag{2.71}$$

Les relations (2.64), (2.69) et (2.70-71) sont l'équivalent, pour les bancs de filtres de synthèse, des relations (2.54-62) pour les bancs de filtres d'analyse. De ce fait, elles jouent un rôle central dans l'étude des bancs de filtres de synthèse avec reconstruction du signal, ainsi que nous le montrerons par la suite.

2.3 Quelques résultats généraux

Dans cette section, nous démontrerons quelques théorèmes liés à la propriété de reconstruction des bancs de filtres d'analyse et de synthèse. Lorsque dans la suite nous parlerons de matrices de filtres, nous parlerons de matrices de fonctions rationnelles en z^{-1} ayant la forme de $\mathbf{H}_m(z)$ ou de $\mathbf{H}_p(z)$ et qui ont été introduites dans la section précédente. Notons que ces matrices ont des structures bien particulières et ne sont donc pas des matrices générales de fonctions en z^{-1}. Nous présenterons d'abord quelques propriétés des matrices de filtres. Nous indiquerons ensuite les principaux résultats qu'il est possible d'obtenir sur les bancs de filtres d'analyse et de synthèse, en particulier dans le cas de la reconstruction parfaite et de la reconstruction sans repliement spectral ou sans diaphonie. Finalement, nous montrerons qu'il existe des liens étroits entre les bancs de filtres d'analyse et de synthèse.

A notre connaissance, la plupart des résultats de cette section sont originaux. En fait, nous pensons que leur obtention n'est possible que grâce à un formalisme suffisamment puissant tel que celui que nous avons introduit dans la section précédente.

2.3.1 Propriétés de matrices de filtres

Nous allons utiliser ce que l'on pourrait appeler une algèbre sur un espace vectoriel de fonctions rationnelles en z^{-1}. Comme les opérations fondamentales restent inchangées (si ce n'est que les scalaires sont remplacés par des fonctions rationnelles en z^{-1}) et que l'on respecte les domaines de convergence des transformées en z impliquées (les solutions hors de ces domaines sont sans intérêt pour la pratique), cette algèbre est suffisamment bien définie pour les problèmes que nous cherchons à résoudre. Posons quelques définitions qui seront utiles pour l'analyse des matrices de filtres.

Définition D2.16: Une matrice $\mathbf{H}(z)$ dont les éléments sont des fonctions rationnelles de z^{-1} est singulière si son déterminant $\Delta(z)$ est nul quel que soit z.

Notons qu'au sens habituel, la matrice $\mathbf{H}(z)$ est également singulière pour tous les zéros de $\Delta(z)$, mais que ces singularités isolées correspondent ici à des pôles des fonctions rationnelles en z^{-1} de l'inverse de $\mathbf{H}(z)$.

Définition D2.17: Le rang r d'une matrice H(z) est la dimension de l'espace de fonctions rationnelles en z^{-1} que l'on peut couvrir par combinaison linéaire des vecteurs ligne ou colonne de cette matrice.

Notons que la combinaison linéaire admet des facteurs de pondération qui sont également des fonctions rationnelles en z^{-1}.

Définition D2.18: Une matrice H(z) est stable si tous ses éléments sont des filtres stables, donc des fonctions rationnelles en z^{-1} avec des pôles strictement à l'intérieur du cercle unité.

Remarque R2.9: La matrice des cofacteurs C(z) d'une matrice H(z) stable est une matrice stable.

Cette dernière remarque se vérifie aisément, par exemple en ramenant tous les éléments à un dénominateur commun D(z) (dont les zéros sont à l'intérieur du cercle unité), puis en calculant les cofacteurs. Ceux-ci ont alors tous un dénominateur égal à $[D(z)]^{N-1}$ et sont donc stables.

Définition D2.19: Une matrice H(z) est qualifiée de causale si tous ses éléments sont des filtres causals.

Remarque R2.10: La matrice des cofacteurs C(z) d'une matrice H(z) causale est une matrice causale, puisque les éléments de C(z) sont obtenus par multiplication et addition de filtres causals.

Nous allons maintenant considérer les inverses des matrices de filtres, donc également les déterminants et les matrices de cofacteurs de telles matrices. Rappelons que les matrices $H_p(z)$ et $H_m(z)$ sont étroitement liées par la matrice de Fourier (qui est évidemment non-singulière). Donc, les propriétés de l'une sont immédiatement transmissibles à l'autre (par exemple, la stabilité, la causalité ou le rang).

Dans ce qui suit, nous allons montrer que la structure particulière de $H_p(z)$ et de $H_m(z)$ se répercute sur leurs inverses, cofacteurs et déterminants. Une première sous-section examine la matrice des cofacteurs d'une matrice de filtres, et une seconde le déterminant d'une telle matrice. Ensuite, les caractéristiques de l'inverse sont explorées en détails, en particulier la causalité et la stabilité de cet inverse.

a) Matrices de cofacteurs des matrices de filtres

Evaluons d'abord la matrice des cofacteurs de $\mathbf{H}_p(z)$ que nous appellerons $\mathbf{C}_p(z)$. Nous allons montrer que celle-ci est similaire à une matrice de filtres polyphases. Nous dirons que deux matrices sont similaires (dénoté par "$\sim$") si elles ont une structure comparable. Par exemple, si toutes deux sont diagonales et ont des éléments qui ne sont fonction que de z^{-N}, alors on pourra dire qu'elles sont similaires. Toutefois, si l'une a un élément diagonal non-nul ou un élément diagonal comportant une puissance de z différente d'un multiple de N, elles ne sont plus similaires. La matrice $\mathbf{C}_p(z)$ peut s'écrire, en utilisant la décomposition donnée en (2.36), et en exploitant le fait que la matrice des cofacteurs d'un produit est égale au produit des matrices de cofacteurs, avec:

$$\mathbf{C}_p(z) = \mathbf{C}_i(z) \cdot \mathbf{C}_n(z^N) \cdot \mathbf{C}_d(z^N) \tag{2.72}$$

où $\mathbf{C}_i(z)$, $\mathbf{C}_n(z^N)$ et $\mathbf{C}_d(z^N)$ sont les matrices de cofacteurs de $\mathbf{I}_d(z)$, $\mathbf{N}_p(z^N)$ et $\mathbf{D}_p(z^N)$ respectivement. Il est aisé de voir que $\mathbf{C}_n(z^N)$ et $\mathbf{C}_d(z^N)$ sont des matrices similaires à $\mathbf{N}_p(z^N)$ et $\mathbf{D}_p(z^N)$. Par contre, $\mathbf{C}_i(z)$ est égal à:

$$\mathbf{C}_i(z) = \mathrm{Diag}[\, z^{-N(N-1)/2} , z^{(-N(N-1)/2)+1} , \ldots\ldots , z^{(-N(N-1)/2)+N-1} \,] \tag{2.73}$$

Donc, les lignes successives de $\mathbf{C}_p(z)$ ont des délais décroissants (et non pas croissants comme dans $\mathbf{H}_p(z)$).

Si N est impair, alors $z^{-N(N+1)/2}$ est un délai multiple de N. Dans ce cas, en raison de (2.72) et (2.73), on voit que:

$$\mathbf{J} \cdot \mathbf{C}_p(z) \sim \mathbf{H}_p(z) \tag{2.74}$$

où $\mathbf{J}$ est la matrice de permutation définie en (2.48). Notons que maintenant les délais de la matrice $\mathbf{J} \cdot \mathbf{C}_p(z)$ sont dans un ordre croissant, d'où la similarité. Si N est pair, alors $z^{-N(N-1)/2} z^{-N/2}$ est un délai multiple de N, donc:

$$z^{-N/2} \cdot \mathbf{J} \cdot \mathbf{C}_p(z) \sim \mathbf{H}_p(z) \tag{2.75}$$

Pour montrer les particularités de la matrice de cofacteurs $\mathbf{C}_m(z)$ de $\mathbf{H}_m(z)$, notons que les déterminants de $\mathbf{H}_p(z)$ et de $\mathbf{H}_m(z)$ sont liés, à cause de (2.60), par la relation suivante:

$$\Delta_p(z) = 1/N^N \cdot \Delta_f \cdot \Delta_m(z) \tag{2.76}$$

où Δ_f est le déterminant de la matrice de Fourier en (2.44) égal à:

$$\begin{aligned} \Delta_f &= N^{N/2} \qquad \lfloor (N+1)/2 \rfloor \text{ impair} \\ &= j \cdot N^{N/2} \qquad \lfloor (N+1)/2 \rfloor \text{ pair} \end{aligned} \tag{2.77}$$

Egalement suivant (2.60), les matrices de cofacteurs $\mathbf{C}_p(z)$ et $\mathbf{C}_m(z)$ sont liées par la relation suivante:

$$\begin{aligned} \mathbf{C}_p(z) &= N \cdot \Delta_p(z)/\Delta_m(z) \cdot \mathbf{F}^{-1} \cdot \mathbf{C}_m(z) \\ &= \Delta_p(z)/\Delta_m(z) \cdot \mathbf{J} \cdot \mathbf{F} \cdot \mathbf{C}_m(z) \end{aligned} \tag{2.78}$$

où $\Delta_p(z)/\Delta_m(z)$ est une constante scalaire définie par (2.76-77) que nous appellerons dorénavant c_d. Puisque $\mathbf{J}$ est autoinverse, il ressort que:

$$\mathbf{J} \cdot \mathbf{C}_p(z) = c_d \cdot \mathbf{F} \cdot \mathbf{C}_m(z) \tag{2.79}$$

Lorsque N est impair, et en comparant (2.79) à (2.60), on voit que $\mathbf{C}_m(z)$ est une matrice similaire à une matrice $\mathbf{H}_m(z)$ de filtres modulés puisque $\mathbf{J} \cdot \mathbf{C}_p(z)$ est similaire à une matrice $\mathbf{H}_p(z)$ selon (2.74). Lorsque N est pair, le délai de $z^{\mathbf{p}_{N/2}}$ dans (2.75) provoque un changement de signe des lignes impaires, et dans ce cas, la matrice $\mathbf{C}'_m(z)$ donnée par:

$$\mathbf{C}'_m(z) = \text{Diag}[\ 1,\ -1,\ 1,\ \ldots\ ,\ -1] \cdot \mathbf{C}_m(z) \tag{2.80}$$

est similaire à une matrice $\mathbf{H}_m(z)$ de filtres modulés.

Le but de cette partie sur les matrices de cofacteurs était de montrer qu'elles ont également une structure très particulière. Tout comme la matrice $\mathbf{H}_m(z)$, la matrice $\mathbf{C}_m(z)$ est définie complètement par une seule de ses lignes, les autres étant obtenues simplement par modulation.

b) déterminants de matrices de filtres

Rappelons que les relations (2.76) et (2.77) montrent que les déterminants $\Delta_p(z)$ et $\Delta_m(z)$ sont liés par une constante scalaire. Dans la suite, nous n'évaluerons plus que $\Delta_p(z)$, puisque $\Delta_m(z)$ peut en être immédiatement déduit. Toujours en vertu de (2.36), on peut écrire $\Delta_p(z)$ comme:

$$\Delta_p(z) = \Delta_i(z) \cdot \Delta_n(z^N) \cdot \Delta_d(z^N) \tag{2.81}$$

où $\Delta_i(z)$, $\Delta_n(z^N)$ et $\Delta_d(z^N)$ sont les déterminants de $\mathbf{I}_d(z)$, $\mathbf{N}_p(z^N)$ et $\mathbf{D}_p(z^N)$ respectivement. A nouveau, nous avons utilisé la notation z^N pour indiquer une fonction qui n'a que des puissances N-ièmes de z, et évidemment, le déterminant d'une matrice en z^N ne comportera que des puissances N-ièmes de z. Les différents facteurs de (2.81) sont égaux à (suivant (2.36-39)):

$$\Delta_i(z) = z^{-N(N-1)/2} \tag{2.82}$$

$$\Delta_n(z^N) = \quad \alpha_0 \quad + \quad \alpha_1 \cdot z^{-N} \quad + \quad \alpha_2 \cdot z^{-2N} \quad + \; . \; . \tag{2.83}$$

$$\Delta_d(z) = \quad 1/\Big(\prod_{i=0}^{N-1} D_i(z^N) \Big) \tag{2.84}$$

Dans (2.84), les α_i sont des facteurs scalaires obtenus lors de l'évaluation du déterminant de $\mathbf{N}_p(z^N)$. Donc, le déterminant $\Delta(z)$ d'une matrice de filtres $\mathbf{H}(z)$ peut, selon (2.81-84), s'exprimer comme suit:

$$\begin{aligned} \Delta(z) &= f(z^N) \qquad N \text{ impair} \\ z^{-N/2} \cdot \Delta(z) &= f(z^N) \qquad N \text{ pair} \end{aligned} \tag{2.85}$$

Deux conséquences importantes découlent de (2.85). Premièrement, outre d'éventuels zéros à l'origine, tous les pôles et zéros de $\Delta(z)$ se répètent N fois sur un cercle, puisque si p_i est un pôle ou zéro de $\Delta(z)$, alors $W^k \cdot p_i$ (k=1...N-1) est également un pôle ou zéro de $\Delta(z)$. Deuxièmement, et nous verrons l'importance de ce fait par exemple lors de la dérivation de bancs de filtres RIF, le fait d'imposer certaines contraintes au déterminant $\Delta(z)$ d'une matrice de filtres (typiquement, l'annulation de coefficients) est grandement simplifié puisque seul un coefficient tous les N coefficients est différent de zéro.

c) Inverse des matrices de filtres

Deux problèmes sont liés aux matrices inverses: la causalité et la stabilité. A nouveau, comme les inverses de $H_p(z)$ et de $H_m(z)$ sont étroitement liés par (2.60), on a:

$$[H_p(z)]^{-1} = N \cdot [H_m(z)]^{-1} \cdot F^{-1} \tag{2.86}$$

Nous ne considérerons donc en général que l'inverse de $H_p(z)$.

Examinons d'abord le problème de la causalité de l'inverse. En général, l'inverse d'une matrice de filtres causale (au sens de la définition D2.19) n'est pas causale. Ecrivons l'inverse de $H_p(z)$ comme suit:

$$\begin{aligned}[H_p(z)]^{-1} &= 1/\Delta_p(z) \cdot [C_p(z)]^T \\ &= z^{N(N-1)/2} \cdot 1/\Delta_n(z^N) \cdot 1/\Delta_d(z^N) \cdot C_d(z^N) \cdot [C_n(z^N)]^T \cdot C_i(z)\end{aligned} \tag{2.87}$$

où nous avons mis à profit les relations (2.72) et (2.81). Maintenant, tous les facteurs en (2.87), à l'exception de $z^{N(N-1)/2}$, sont causals (entre autres en vertu de la remarque R2.10). Notons que:

$$z^{N(N-1)/2} \cdot C_i(z) = \text{Diag}[\ 1\ ,\ z\ ,\ z^2\ ,\ \ldots\ ,\ z^{N-1}\ ] \tag{2.88}$$

Donc, nous pouvons dire que si une matrice $H_p(z)$ est causale, alors la matrice:

$$G_p(z) = z^{-N+1} \cdot [H_p(z)]^{-1} \tag{2.89}$$

est également une matrice causale. Ce résultat est important, car il donne une limite inférieure au délai produit par un système avec reconstruction, comme noté dans la remarque ci-dessous.

Remarque R2.11: Un système d'analyse ou de synthèse avec reconstruction et comportant N canaux produit au moins un délai de N-1 échantillons (échantillons correspondant à la fréquence la plus élevée) si tous les filtres impliqués sont causals.

Concernant le délai appliqué dans (2.89), la remarque suivante est appropriée:

Remarque R2.12: En général, on choisira de multiplier l'inverse dans (2.89) par z^{-N} plutôt que par z^{-N+1}. La raison en est qu'un système variant dans le temps avec une période N reste inchangé par l'introduction d'un délai multiple de N, ce qui n'est pas forcément le cas pour des délais différents.

Le problème de la stabilité de l'inverse est plus délicat. Pour le cas le plus général, il sera seulement possible de donner une condition suffisante pour cette stabilité. On montrera que cette condition est également nécessaire pour N=2 ainsi que pour les bancs de filtres modulés (voir la définition D2.8).

Ecrivons l'inverse de $H_p(z)$ de la façon suivante (suivant la factorisation en (2.36)):

$$[H_p(z)]^{-1} = [D_p(z^N)]^{-1} \cdot [N_p(z^N)]^{-1} \cdot [I_d(z)]^{-1} \tag{2.90}$$

L'inverse de $D_p(z^N)$ est une matrice diagonale de filtres RIF égaux à $D_i(z^N)$ selon (2.39). Notons que ces filtres RIF sont à phase minimum (zéros à l'intérieur du cercle unité), puisque nous avons supposé que $H_p(z)$ était une matrice stable. L'inverse de $I_d(z)$ ne présente pas de problème de stabilité (la causalité ayant été considérée plus haut). Donc seul $[N_p(z^N)]^{-1}$ influence la stabilité de $[H_p(z)]^{-1}$.

Rappelons que $\Delta_n(z^N)$ est le déterminant de $N_p(z^N)$. La condition suffisante suivante peut alors être posée pour la stabilité de l'inverse d'une matrice de filtres:

Condition C2.1: Si les zéros de $\Delta_n(z^N)$ sont à l'intérieur du cercle unité, alors la stabilité de la matrice $[H_p(z)]^{-1}$ est garantie.

Ce résultat est évident, puisque:

$$[N_p(z^N)]^{-1} = 1/\Delta_n(z^N) \cdot [C_n(z^N)]^T \tag{2.91}$$

Selon la remarque R2.9, $C_n(z^N)$ est stable. Si $\Delta_n(z^N)$ est à phase minimale, alors son inverse est un filtre stable, donc $[N_p(z^N)]^{-1}$ et par conséquent aussi $[H_p(z)]^{-1}$ sont des matrices stables.

Démonstration de la nécessité de la condition C2.1 pour le cas N=2

Pour dériver la nécessité de cette condition, il faut considérer le cas d'un pôle de $1/\Delta_n(z^N)$ à l'extérieur du cercle unité. Manifestement, pour que $[H_p(z)]^{-1}$ soit stable, ce pôle doit se simplifier avec un zéro. Comme il est à l'extérieur du cercle unité, il n'y a pas de simplification possible avec les zéros de $[D_p(z^N)]^{-1}$ ou de $[I_d(z)]^{-1}$. Seuls peuvent entrer en ligne de compte les zéros de $C_n(z^N)$. Nous allons montrer l'impossibilité d'une telle simplification pour le cas N=2. Pour ce faire, nous reformulerons $N_p(z^N)$ de la façon suivante (où nous omettons la notation z^N par simplicité):

$$N_p(z) = \begin{bmatrix} \alpha_0(z) & 0 \\ 0 & \alpha_1(z) \end{bmatrix} \cdot \begin{bmatrix} H'_{00}(z) & H'_{10}(z) \\ H'_{01}(z) & H'_{11}(z) \end{bmatrix} \cdot \begin{bmatrix} \beta_0(z) & 0 \\ 0 & \beta_1(z) \end{bmatrix} \quad (2.92)$$

Les $\alpha_i(z)$ représentent les facteurs communs aux éléments d'une même ligne i, et les $\beta_j(z)$ représentent ceux communs aux éléments d'une même colonne j de la matrice $N_p(z)$. L'inverse $[N_p(z)]^{-1}$ peut dès lors s'écrire comme suit:

$$[N_p(z)]^{-1} = \begin{bmatrix} 1/\beta_0(z) & 0 \\ 0 & 1/\beta_1(z) \end{bmatrix} 1/\Delta'_n(z) \begin{bmatrix} H'_{11}(z) & -H'_{10}(z) \\ -H'_{01}(z) & H'_{00}(z) \end{bmatrix} \cdot \begin{bmatrix} 1/\alpha_0(z) & 0 \\ 0 & 1/\alpha_1(z) \end{bmatrix} \quad (2.93)$$

où:

$$\Delta'_n(z) = H'_{00}(z) \cdot H'_{11}(z) - H'_{01}(z) \cdot H'_{10}(z) \quad (2.94)$$

Evidemment, les $\alpha_i(z)$ et les $\beta_i(z)$ doivent avoir leurs zéros à l'intérieur du cercle unité. Si maintenant $\Delta'_n(z)$ possède un zéro à l'extérieur du cercle unité, il doit se simplifier afin que $[N_p(z)]^{-1}$ soit stable. Prenons l'élément avec l'indice 0,0 de la matrice centrale de (2.93), qui est donné par:

$$e_{0,0} = H'_{11}(z) \; / \; (H'_{00}(z) \cdot H'_{11}(z) - H'_{01}(z) \cdot H'_{10}(z)) \quad (2.95)$$

Si un pôle p_i doit être annulé par un zéro en (2.95), alors $e_{0,0}$ devra être de la forme suivante:

$$e_{0,0} = (z^{-1}p_i - 1)\cdot H''_{11}(z) \;/\; ((z^{-1}p_i - 1)\cdot H'_{00}(z)\cdot H''_{11}(z) - H'_{01}(z)\cdot H'_{10}(z)) \qquad (2.96)$$

Comme $(z^{-1}p_i - 1)$ est un facteur du dénominateur de (2.96), il est également un facteur de $H'_{01}(z)\cdot H'_{10}(z)$. $(z^{-1}p_i - 1)$ est donc un facteur de $H'_{01}(z)$ ou de $H'_{10}(z)$ mais aussi de $H'_{11}(z)$, ce qui est en contradiction avec la factorisation que nous avons effectuée en (2.92). La même démarche peut être appliquée aux autres éléments de la matrice centrale en (2.93) et par conséquent, un pôle instable ne peut pas être éliminé par un zéro en (2.93). Nous pouvons donc poser la condition suivante:

Condition C2.2: Pour N=2, il est nécessaire et suffisant que les zéros de $\Delta_n(z^N)$ soient à l'intérieur du cercle unité pour que $[H_p(z)]^{-1}$ soit une matrice de filtres stable.

Pour N>2, la situation est plus complexe car d'une part les cofacteurs sont des sous-déterminants (et non plus simplement des filtres), et d'autre part, le déterminant est une somme de plus de deux termes (ce qui ne permet plus la démonstration qui suit (2.96)). Comme les zéros d'une somme de polynômes sont non-trivialement liés aux zéros des polynômes sommés, nous pouvons dire que la nécessité de la condition C2.1 reste un problème ouvert pour N>2 et des filtres généraux.

Par contre, pour des bancs de filtres modulés, les conditions pour la stabilité de la matrice inverse peuvent être bien établies. Dans ce cas, il est plus aisé de considérer l'inverse de la matrice $\mathbf{H_m}(z)$, de laquelle l'inverse de $\mathbf{H_p}(z)$ suit immédiatement selon (2.83). Notons que si le filtre $H_i(z)$ est égal à la i-ième version modulée d'un filtre prototype $H_\Omega(z)$, avec:

$$H_i(z) = H_\Omega(W^i z) \qquad (2.97)$$

alors la matrice $\mathbf{H_m}(z)$ possède une forme très particulière:

$$H_m(z) = \begin{bmatrix} H_\Omega(z) & H_\Omega(Wz) & H_\Omega(W^2z) & \cdots & H_\Omega(W^{N-1}z) \\ H_\Omega(Wz) & H_\Omega(W^2z) & \ddots & \ddots & H_\Omega(z) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ H_\Omega(W^{N-1}z) & H_\Omega(z) & H_\Omega(Wz) & \cdots & H_\Omega(W^{N-2}z) \end{bmatrix} \quad (2.98)$$

Notons que si $H_\Omega(z)$ est un filtre RII, on le transformera d'abord en filtre polyphase, que l'on peut écrire comme $N(z)/D(z^N)$, avec la méthode décrite au début de la section 2.2.2. Le dénominateur en z^N est commun à tous les éléments et peut donc être mis en facteur. Prémultiplions de surcroît $H_m(z)$ en (2.98) par J pour obtenir:

$$J \cdot H_m(z) = 1/D(z^N) \begin{bmatrix} N_\Omega(z) & N_\Omega(Wz) & N_\Omega(W^2z) & \cdots & N_\Omega(W^{N-1}z) \\ N_\Omega(W^{N-1}z) & N_\Omega(z) & N_\Omega(Wz) & \ddots & N_\Omega(W^{N-2}z) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ N_\Omega(Wz) & N_\Omega(W^2z) & N_\Omega(W^3z) & \cdots & N_\Omega(z) \end{bmatrix} \quad (2.99)$$

La matrice en (2.99) est une matrice de Toeplitz circulante, c'est-à-dire une matrice de convolution. Elle peut donc être diagonalisée à l'aide de la matrice de Fourier. Les éléments sur la diagonale sont alors les éléments de la transformée de Fourier inverse de la première ligne de la matrice de convolution (voir annexe A2.1):

$$J \cdot H_m(z) = 1/D(z^N) \cdot F^{-1} \cdot L(z) \cdot F \quad (2.100)$$

où:

$$L(z) = \mathrm{Diag}[\ l(z)\ ] \quad (2.101)$$

avec:

$$\begin{aligned} l(z) &= N \cdot F^{-1} \cdot [\ N_\Omega(z),\ N_\Omega(Wz),\ N_\Omega(W^2z), \ldots N_\Omega(W^{N-1}z)\ ]^T \\ &= N \cdot J\ [\ N_{\Omega 0}(z^N),\ z^{-1} \cdot N_{\Omega 1}(z^N),\ z^{-2} \cdot N_{\Omega 2}(z^N), \ldots\ z^{-N+1} \cdot N_{\Omega N-1}(z^N)\ ]^T \end{aligned} \quad (2.102)$$

$\mathbf{I}(z)$ est donc le vecteur des filtres polyphases de $N_\Omega(z)$. Dans ces conditions, la matrice inverse $[\mathbf{H}_m(z)]^{-1}$ devient avec (2.100):

$$[\ \mathbf{H}_m(z)\ ]^{-1} = D(z^N) \cdot \mathbf{F}^{-1} \cdot [\ \mathbf{L}(z)\ ]^{-1} \cdot \mathbf{F} \cdot \mathbf{J} \tag{2.103}$$

où nous avons utilisé le fait que $\mathbf{J}^{-1} = \mathbf{J}$. L'inverse de $\mathbf{L}(z)$ est égal à:

$$[\ \mathbf{L}(z)\ ]^{-1} = 1/N \cdot \mathrm{Diag}[\ 1/N_{\Omega 0}(z^N), z^{N-1}/N_{\Omega N-1}(z^N), \ldots, z/N_{\Omega 1}(z^N)\ ] \tag{2.104}$$

Nous remarquons à nouveau qu'il faudra multiplier $[\mathbf{H}_m(z)]^{-1}$ au moins par z^{-N+1} afin d'obtenir une matrice inverse causale. En fait, une multiplication par z^{-N} simplifiera les notations (voir également les remarques R2.11 et R2.12). Considérons donc la matrice $\mathbf{G}_m(z)$ égale à:

$$\mathbf{G}_m(z) = z^{-N} \cdot [\ \mathbf{H}_m(z)\ ]^{-1} \tag{2.105}$$

En utilisant les relations (2.103) et (2.104), il peut être vérifié que $\mathbf{G}_m(z)$ est également une matrice de filtres modulés donnée par:

$$\mathbf{G}_m(z) = \begin{bmatrix} G_\Omega(z) & G_\Omega(Wz) & G_\Omega(W^2z) & \cdots & G_\Omega(W^{N-1}z) \\ G_\Omega(Wz) & G_\Omega(W^2z) & \iddots & \iddots & G_\Omega(z) \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ G_\Omega(W^{N-1}z) & G_\Omega(z) & G_\Omega(Wz) & \cdots & G_\Omega(W^{N-2}z) \end{bmatrix} \tag{2.106}$$

où, de (2.103-105):

$$G_\Omega(z) = D(z^N) \cdot [\ z^{-N}/N_{\Omega 0}(z^N) + z^{-N+1}/N_{\Omega 1}(z^N) + \ldots + z^{-1}/N_{\Omega N-1}(z^N)\ ] \tag{2.107}$$

Dans l'annexe A2.2, il est montré que pour que $\mathbf{G}_m(z)$ soit une matrice stable, il est nécessaire et suffisant que tous les zéros des $N_{\Omega i}(z^N)$ soient à l'intérieur du cercle unité. De façon équivalente, il faut que les N filtres polyphases qui composent $H_\Omega(z)$ soient à phase minimale.

Considérons encore le déterminant de la matrice $\mathbf{H}_m(z)$ en (2.100). Celui-ci est égal à:

$$\Delta_m(z) = [1/D(z^N)]^N \cdot 1/\Delta_j \cdot \Delta_1(z) \tag{2.108}$$

où Δ_j et $\Delta_1(z)$ sont les déterminants de **J** et **L**(z) respectivement. Puisque seuls les zéros de $\Delta_m(z)$ nous intéressent pour la stabilité (ils seront les pôles de l'inverse), nous allons évaluer $\Delta_1(z)$:

$$\Delta_1(z) = N^N \cdot z^{-N(N-1)/2} \cdot \prod_{i=0}^{N-1} N_{\Omega i}(z^N) \tag{2.109}$$

Afin que l'inverse de $\mathbf{H}_m(z)$ soit stable, on voit qu'il est suffisant que tous les zéros du déterminant soient à l'intérieur du cercle unité, et ceci suivant les relations (2.107) et (2.109). De surcroît, et ceci en vertu de l'annexe A2.2, il est également nécessaire que tous les zéros soient à l'intérieur du cercle unité. On peut donc poser la condition suivante pour les bancs de filtres modulés:

Condition C2.3: Pour que l'inverse d'une matrice de filtres modulés soit stable, il est nécessaire et suffisant que les zéros du déterminant soient à l'intérieur du cercle unité ou que les filtres polyphases correspondant au filtre prototype soient à phase minimale. Les deux conditions sont absolument équivalentes.

Notons finalement que dans le cas de bancs de filtres modulés, la matrice $\mathbf{H}_p(z)$ est égale à, selon les relations (2.56), (2.60) et (2.100):

$$\begin{aligned} \mathbf{H}_p(z) &= 1/N \cdot \mathbf{F} \cdot \mathbf{H}_m(z) \\ &= 1/N \cdot \mathbf{F} \cdot \mathbf{J} \cdot \mathbf{F} \cdot \mathbf{L}(z) \cdot \mathbf{F}^{-1} \\ &= \mathbf{L}(z) \cdot \mathbf{F}^{-1} \end{aligned} \tag{2.110}$$

Considérons encore quelques propriétés générales des inverses de matrices de filtres. De la même façon que pour les matrices de cofacteurs (relations (2.72-80), nous allons montrer que les inverses de $\mathbf{H}_p(z)$ et $\mathbf{H}_m(z)$ ont des structures particulières. Cette remarque est importante, car si les matrices inverses n'étaient pas similaires à des matrices de filtres, il serait évidemment impossible de leur associer un banc de filtres correspondant. En inversant pas à pas $\mathbf{H}_p(z)$ donnée par sa factorisation en (2.36), il est aisé de vérifier que:

$$z^{-N} \cdot [H_p(z)]^{-1} \cdot J \sim [H_p(z)]^T \tag{2.111}$$

En utilisant la relation (2.86) puis (2.56), on peut voir que

$$z^{-N} \cdot [H_m(z)]^{-1} \sim [H_m(z)]^T \tag{2.112}$$

Donc, tant l'inverse de $H_p(z)$ que l'inverse de $H_m(z)$ ont une structure similaire à la matrice de départ transposée. Ceci était bien sûr prévisible au vu des similarités que nous avions vues entre les matrices et les matrices de cofacteurs associées.

Très souvent, il ne sera pas possible d'inverser parfaitement la matrice de filtres $H_p(z)$ ou $H_m(z)$ (par exemple pour des questions de stabilité). Si la matrice $[G_p(z)]^T$ ou $[G_m(z)]^T$ (qui correspond aux filtres de synthèse) n'est que partiellement l'inverse de la matrice $H_p(z)$ ou $H_m(z)$ (correspondant aux filtres d'analyse), on s'intéresse à la "qualité" de cet inverse partiel (c'est-à-dire dans quelle mesure le produit de la matrice de filtres par cet inverse partiel est proche d'une matrice unité ou diagonale). Dans ce cas, les relations apparaissant ci-dessous seront utiles.

Pour le cas des bancs de filtres d'analyse, nous étendons d'abord $g(z)$ (à partir de (2.61) par exemple) en une matrice $G_m(z)$, et ceci en modulant $g(z)$ par W^i afin d'obtenir la i-ième colonne de la matrice $G_m(z)$. On obtient alors la relation suivante:

$$H_m(z) \cdot [G_m(z)]^T = S(z) \tag{2.113}$$

Un cas important est celui où $S(z)$ est une matrice diagonale, qui a dès lors la forme suivante:

$$S(z) = \text{Diag}[\ S(z),\ S(Wz),\ S(W^2z),\ \ldots\ .\ S(W^{N-1}z)\ ] \tag{2.114}$$

En pré- et post-multipliant (2.113) par la matrice de Fourier (puisque $F=F^T$), on obtient, suivant (2.60), la relation équivalente pour les matrices polyphases:

$$H_p(z) \cdot [G_p(z)]^T = 1/N^2 \cdot F \cdot S(z) \cdot F \tag{2.115}$$

Pour les bancs de filtres de synthèses, rappelons explicitement les relations (2.66) et (2.68):

$$[G_m(z)]^T \cdot H_m(z) = T(z) \tag{2.116}$$

$$[G_p(z)]^T \cdot J \cdot H_p(z) = T(z) \tag{2.117}$$

Notons que les relations (2.113) et (2.115) sont équivalentes à (2.116-117) si et seulement si [str80]:

$$S(z) = T(z) = \alpha(z) \cdot I \tag{2.118}$$

où $\alpha(z)$ est une fonction de transfert arbitraire et I la matrice unité. La relation (2.118) est importante, car elle indique dans quels cas les résultats obtenus pour des bancs de filtres d'analyse (section 2.1.2) peuvent être sans autre transférés aux bancs de filtres de synthèse (section 2.1.3) et réciproquement.

Pour conclure cette partie sur les inverses de matrices de filtres, nous allons considérer deux cas particuliers, celui de filtres RIF de longueur N (donc égale au nombre de canaux) et celui de filtres passe-bandes idéaux.

Dans la cas de filtres RIF ayant une longueur égale au nombre de canaux, la solution de reconstruction parfaite après l'analyse est très simple (comme nous l'avions montré dans [vet83]), donc l'inverse doit également être simple. La matrice $N_p(z^N)$ en (2.36) se réduit à une matrice purement scalaire, alors que $D_p(z^N)$ est égal à l'identité (puisque les filtres sont RIF). L'inverse de $H_p(z)$ devient:

$$[\ H_p(z)\ ]^{-1} = N_p^{-1} \cdot [\ I_d(z)\ ]^{-1} \tag{2.119}$$

où la notation N_p^{-1} montre qu'il s'agit d'une matrice inverse dont les éléments sont des scalaires (et non pas des fonctions de z). Pour rendre la matrice $[H_p(z)]^{-1}$ causale, il faut la multiplier par z^{-N+1} (selon (2.86)). Dès lors, l'inverse est une matrice de filtres RIF de longueur N, et dont les coefficients ont été obtenus par inversion de la matrice N_p. Donc, la reconstruction parfaite ne pose pas de problème dans ce cas particulier qui n'a malheureusement que peu d'intérêt en pratique vu la longueur extrêmement réduite des filtres.

Le cas de filtres idéaux a bien sûr surtout un intérêt théorique, mais il est intéressant de le traiter dans le cadre de notre formalisme matriciel. Introduisons les deux types de filtres suivants (toujours pour N canaux complexes):

i) filtre A(z) avec atténuation hors-bande idéale. A(z) possède la propriété suivante:

$$A(z) \cdot A(W^k z) = [\ A(z)\ ]^2 \qquad k = n \cdot N,\ n=0,1,..$$
$$= 0 \qquad k \neq nN \tag{2.120}$$

ii) filtre P(z) passe-bande parfait. Sa transformée en z sur le cercle unité a la propriété suivante:

$$P(e^{j\omega}) = 1 \qquad \omega \in \{\ 0\ ..\ 2\pi/N\ \}$$
$$= 0 \qquad \text{ailleurs} \tag{2.121}$$

Notons que ce filtre idéal n'est pas causal et que sa réponse impulsionnelle est infiniment longue. Evidemment, (2.121) entraîne (2.120) mais la réciproque n'est pas vraie. Soient maintenant $\mathbf{A}_m(z)$ et $\mathbf{P}_m(z)$ les matrices de filtres obtenues par modulation de A(z) et P(z) respectivement (par modulation équidistante sur le cercle unité aux points $(k2\pi)/N$, k=0..N-1). Il est dès lors aisé de vérifier que:

$$\mathbf{A}_m(z) \cdot [\ \mathbf{A}_m(z)\ ]^T = \text{Diag}[\ S(z)\ S(z)\ \ldots\ S(z)\ ] \tag{2.122}$$

où:

$$S(z) = \sum_{k=0}^{N-1} [\ A(W^k z)\ ]^2 \tag{2.123}$$

et:

$$\mathbf{P}_m(z) \cdot [\ \mathbf{P}_m(z)\ ]^T = \mathbf{I} \tag{2.124}$$

Evidemment, la relation (2.124) est la solution idéale à tous les problèmes de bancs de filtres comportant des changements de fréquence d'échantillonnage, mais elle est absolument irréaliste puisqu'elle nécessite des filtres dont la longueur tend vers l'infini (pourtant, on a parfois tenté de l'approximer en utilisant des filtres de très grande longueur). La relation (2.122) correspond à un cas plus réaliste, où seul le repliement spectral (ou la diaphonie) doit être éliminé, et ceci au prix d'une reconstruction approximative, puisque (2.123) ne correspond en général pas à un filtre passe-tout. En fait, des filtres similaires à A(z) (donc ayant une atténuation hors-bande quasi parfaite) sont utilisés dans les transmultiplexeurs, et des filtres approximativement idéaux (donc similaires à P(z)) ont été utilisés dans les codeurs

en sous-bandes [crc76] avant l'introduction des filtres QMF.

Mais outre le fait que ces filtres ont une relation avec des systèmes réels, il se trouve que les matrices $\mathbf{A}_m(z)$ et $\mathbf{P}_m(z)$ qui en sont dérivées ont une propriété fort intéressante: le produit de la matrice fois sa transposée est une matrice diagonale. L'annexe A2.3 explore plus en détail les matrices de filtres dotées de cette propriété.

2.3.2 Résultats concernant les bancs de filtres d'analyse

Considérons un banc de filtres d'analyse comme celui de la figure 2.3, stable et sous-échantillonné de façon critique. Le théorème de reconstruction est alors le suivant:

Théorème T2.1: Pour éviter qu'une version modulée de signal d'entrée n'apparaisse dans la sortie d'un banc de filtre d'analyse avec reconstruction, les deux conditions suivantes doivent être remplies:
a) Il est nécessaire que la matrice $\mathbf{H}_p(z)$ ou $\mathbf{H}_m(z)$ soit non-singulière, c'est-à-dire d'un rang égal à N.
b) Il est nécessaire et suffisant de choisir les filtres de synthèse égaux aux éléments de la première ligne de la matrice des cofacteurs de $\mathbf{H}_m(z)$, ceux-ci pouvant être multipliés par un terme constant (qui peut être une fonction rationnelle de z^{-1}).

La partie a) se démontre par l'absurde. Si l'on utilise la factorisation de $\mathbf{H}_p(z)$ donnée en (2.36), alors la condition (2.52) devient:

$$\mathbf{N}_p(z^N) \cdot \mathbf{D}_p(z^N) \cdot \mathbf{g}(z) = [\ T(z) \quad z\cdot T(z) \ \ldots \quad z^{N-1}\cdot T(z)\ ]^T \tag{2.125}$$

Le fait que $\mathbf{H}_p(z)$ soit une matrice singulière implique qu'au moins une des lignes (ou colonnes) de $\mathbf{N}_p(z^N)\cdot \mathbf{D}_p(z^N)$ est linéairement dépendante des autres (ou totalement nulle). En appelant $\mathbf{v}_i(z^N)$ la i-ième ligne de $\mathbf{N}_p(z^N)\cdot \mathbf{D}_p(z^N)$, cela signifie qu'il existe au moins une ligne $\mathbf{v}_k(z^N)$ telle que:

$$\mathbf{v}_k(z^N) = \sum_{\substack{i=0 \\ i\neq k}}^{N-1} \alpha_i \cdot \mathbf{v}_i(z^N) \tag{2.126}$$

La relation (2.125) implique que:

$$\mathbf{v}_i(z^N) \cdot \mathbf{g}(z) = z^i \cdot T(z) \tag{2.127}$$

En utilisant (2.126) et (2.127), on obtient:

$$\mathbf{v}_k(z^N) \cdot \mathbf{g}(z) = \sum_{\substack{i=0 \\ i \neq k}}^{N-1} \alpha_i \cdot z^i \cdot T(z) \tag{2.128}$$

ce qui est en contradiction avec (2.127). Notons ci-dessus qu'en toute généralité, les α_i peuvent être des fonctions de z, mais alors uniquement des fonctions de z^N (à cause de 2.126), et que la démonstration reste donc inchangée.

Le fait que la condition b) soit nécessaire peut être démontré de la façon suivante. $\mathbf{H}_m(z)$ a N lignes (ou colonnes) indépendantes. La condition:

$$\mathbf{H}_m(z) \cdot \mathbf{g}(z) = [\ T(z)\ 0\ \ldots\ 0\ ]^T \tag{2.129}$$

signifie que $\mathbf{g}(z)$ doit être orthogonal aux lignes 1 à N-1 de $\mathbf{H}_m(z)$. Puisque ce sous-espace est de dimension N-1, le sous-espace orthogonal est de dimension 1. Il est défini de façon univoque [str80] par la première ligne de la matrice des cofacteurs de $\mathbf{H}_m(z)$, que nous appellerons $\mathbf{w}_0(z)$.

Le fait que la condition b) est suffisante se démontre de la façon suivante. Si $\mathbf{g}(z)$ est choisi parallèle à $[\ \mathbf{w}_0(z)\ ]^T$ avec:

$$\mathbf{g}(z) = E(z) \cdot [\ \mathbf{w}_0(z)\ ]^T \tag{2.130}$$

où E(z) est une fonction rationnelle en z^{-1}. On a donc:

$$\mathbf{H}_m(z) \cdot \mathbf{g}(z) = [\ E(z) \cdot \Delta_m(z) \quad 0\ \ldots\ 0\ ]^T \tag{2.131}$$

où $\Delta_m(z)$ est le déterminant de $\mathbf{H}_m(z)$.

Les corollaires suivants découlent immédiatement de ce théorème.

Corollaire Co2.1a: Si le sous-échantillonnage est surcritique (N'>N), alors la reconstruction sans repliement spectral est impossible.

Ce corollaire découle du fait qu'alors la condition a) du théorème T2.1 est violée. Ce résultat est par ailleurs intuitivement évident, puisque autrement, il serait possible de représenter un signal quelconque par un nombre réduit de ses échantillons.

Corollaire Co2.1b: Si le sous-échantillonnage est sous-critique (N'<N), il est nécessaire et suffisant que la matrice $\mathbf{H}_m(z)$ (ou $\mathbf{H}_p(z)$) de dimension N' x N ait un rang égal à N' afin de reconstruire le signal sans repliement spectral.

Le rang r de la matrice $\mathbf{H}_m(z)$ donne le nombre de canaux linéairement indépendants. Le système peut dès lors être réduit à ces canaux-là (puisque les autres peuvent être retrouvés par combinaison linéaire). Si r est plus petit que N', nous nous trouvons dans le cas du corollaire Co2.1a. Si r est égal à N', alors on applique le théorème T2.1 afin de reconstruire le signal original à partir de ces N' canaux.

En résumé, nous avons vu que la grandeur pertinente est le facteur de sous-échantillonnage N', puisqu'il donne le nombre d'équations à satisfaire afin de réaliser une reconstruction sans repliement spectral. Le cas limite apparaît lorsque le nombre de canaux N est égal au facteur de sous-échantillonnage N'. Ce cas est traité par le théorème T2.1. Si le nombre de canaux est plus petit, c'est-à-dire si le sous-échantillonnage est surcritique, alors une reconstruction est impossible. Si le nombre de canaux est plus grand, c'est-à-dire si le sous-échantillonnage est sous-critique, il est nécessaire et suffisant de trouver un sous-système de taille N' qui satisfasse les conditions du cas limite donné par le théorème T2.1.

Considérons maintenant le cas de la reconstruction parfaite lorsque le sous-échantillonnage est critique:

Corollaire Co2.1c: Pour obtenir une reconstruction parfaite dans le cas d'un sous-échantillonnage critique, il est suffisant que les zéros du déterminant soient à l'intérieur du cercle unité. Pour N=2 ou pour des bancs de filtres modulés, cette condition est également nécessaire.

Il est aisé de démontrer que cette condition est suffisante. Si les zéros de $\Delta_m(z)$ sont à l'intérieur du cercle unité, alors $1/\Delta_m(z)$ est un filtre stable. On peut donc choisir $\mathbf{g}(z)$ égal à:

$$g(z) = w_0(z) \;/\; \Delta_m(z) \tag{2.132}$$

qui est un vecteur de filtres stables en vertu de la remarque R2.9 et du fait que le numérateur de $\Delta_m(z)$ est à phase minimale. Alors:

$$H_m(z) \cdot g(z) = [\ 1\ 0\ 0\ \ldots\ 0\ ]^T \tag{2.133}$$

La nécessité de cette condition pour N=2 ou pour les bancs de filtres modulés correspond aux conditions C2.2 et C2.3 respectivement, où il est montré que dans ces cas, la stabilité de la matrice inverse (donc de $g(z)$) est directement liée à celle de $1/\Delta_m(z)$.

Pour le cas de sous-échantillonnage surcritique, la reconstruction parfaite est évidemment impossible en vertu du corollaire Co2.1a. Pour le cas sous-critique, il est suffisant de trouver un sous-système de dimension N' qui satisfasse au corollaire Co2.1c afin de pouvoir reconstruire parfaitement le signal original. Notons toutefois qu'il n'est pas nécessaire de trouver un tel sous-système, car les degrés de liberté supplémentaires (dus au fait que nous avons N filtres, mais uniquement N' équations à remplir, où N'<N) permettent de travailler sur des combinaisons linéaires des canaux et de déplacer ainsi les zéros afin de les amener éventuellement tous à l'intérieur du cercle unité. Donc, la condition de phase minimale du numérateur de $\Delta_m(z)$ (pour N'<N) est différente de celle de rang, qui elle est nécessaire (voir corollaire Co2.1b) et ne dépend pas d'éventuelles combinaisons linéaires des colonnes de $H_m(z)$.

Remarque R2.13: Une reconstruction sans distorsion spectrale (mais avec une distorsion de phase) est possible si le déterminant $\Delta(z)$ de la matrice de filtres n'a pas de zéros sur le cercle unité.

Dans ce cas, tous les pôles instables de $1/\Delta(z)$ peuvent être remplacés par des pôles miroirs à l'intérieur du cercle unité, et la fonction de transfert devient dès lors passe-tout.

Le théorème T2.1 (en particulier sa partie b) fait avant tout référence à la matrice de filtres modulés $H_m(z)$. Nous allons voir sa signification pour les matrices de filtres polyphases $H_p(z)$ et $G_p(z)$. En fait, le théorème T2.1 indique qu'il est nécessaire et suffisant de choisir $g(z)$ parallèle au vecteur $w_0(z)$ qui correspond à la première ligne

de la matrice des cofacteurs de $\mathbf{H}_m(z)$. Parallèle signifie ici égal à $\mathbf{w}_0(z)\cdot C(z)$, où C(z) est une constante qui peut être une fonction rationnelle de z^{-1}. Notons que la constante C(z) est alors commune à tous les filtres de synthèse, puisque le vecteur des filtres de synthèse est $\mathbf{g}(z)=C(z)\cdot \mathbf{w}_0(z)$. Puisque ce facteur est commun, il peut être placé après la sommation des sorties des filtres de synthèse (voir la figure 2.3). Mais surtout, il n'influence en rien la propriété d'annulation des repliements spectraux, car celle-ci est uniquement donnée par la "direction" de $\mathbf{w}_0(z)$. Choisissons donc g(z) avec:

$$\begin{aligned} \mathbf{g}(z) &= \mathbf{w}_0(z) \qquad N \text{ impair} \\ &= z^{-N/2}\cdot \mathbf{w}_0(z) \qquad N \text{ pair} \end{aligned} \tag{2.134}$$

Notons que le choix de (2.134) est effectué en toute généralité, puisque tous les autres choix possibles qui réalisent l'annulation des repliements spectraux peuvent être obtenus en ajoutant un filtre commun à la sortie du système. On obtient donc, avec (2.131) et (2.85):

$$\begin{aligned} \mathbf{H}_m(z)\cdot \mathbf{g}(z) &= [\ \Delta_m(z)\ 0\ \ldots\ 0\ ]^T = f'(z^N) \qquad N \text{ impair} \\ &= [\ z^{-N/2}\cdot\Delta_m(z)\ 0\ \ldots\ 0\ ]^T = f''(z^N) \qquad N \text{ pair} \end{aligned} \tag{2.135}$$

Notons que le résultat est une fonction de z^N uniquement. Dans ces conditions, la matrice des filtres modulés correspondant aux filtres de synthèse est égale à:

$$[\ \mathbf{G}_m(z)\ ]^T = [\ \mathbf{g}(z)\ \mathbf{g}(Wz)\ \ldots\ \mathbf{g}(W^{N-1}z)\ ] \tag{2.136}$$

et, en vertu de (2.135), le produit correspondant à (2.113) devient:

$$\begin{aligned} \mathbf{H}_m(z)\cdot [\ \mathbf{G}_m(z)\ ]^T &= \Delta_m(z)\cdot \mathbf{I} \qquad N \text{ impair} \\ &= z^{-N/2}\cdot \Delta_m(z)\cdot \mathbf{I} \qquad N \text{ pair} \end{aligned} \tag{2.137}$$

Les matrices de filtres polyphases correspondantes obéissent donc à la relation suivante, selon (2.115), (2.137) et (2.56):

$$\mathbf{H}_p(z) \cdot [\, \mathbf{G}_p(z) \,]^T = 1/N^2 \cdot \Delta_m(z) \cdot \mathbf{F} \cdot \mathbf{I} \cdot \mathbf{F}$$

$$= 1/N \cdot \Delta_m(z) \cdot \mathbf{J} \qquad N \text{ impair}$$

$$= 1/N \cdot z^{-N/2} \cdot \Delta_m(z) \cdot \mathbf{J} \qquad N \text{ pair} \tag{2.138}$$

La relation (2.138) donne lieu à la remarque importante suivante:

Remarque R2.14: Les différentes lignes de $\mathbf{H}_p(z)$ et de $\mathbf{J} \cdot \mathbf{G}_p(z)$ doivent non seulement être orthogonales les unes par rapport aux autres, mais les produits scalaires de deux mêmes lignes doivent tous avoir une valeur égale.

L'explication physique de cette "normalisation" est la suivante: l'annulation du repliement spectral se fait par soustraction cohérente entre les différents canaux. Si les canaux ont une pondération différente, le résultat de la soustraction est évidemment non nul. Nous verrons plus tard que cette normalisation est spécifique aux bancs de filtres d'analyse avec reconstruction, donc au problème de l'annulation cohérente des repliements spectraux.

Notons que le choix de $\mathbf{g}(z)$ qui a été fait en (2.134) simplifie considérablement les démonstrations. Un choix différent ne change rien au résultat (puisqu'il en résulte un facteur multiplicatif commun à tous les filtres) mais complique sensiblement la démonstration. Ceci est dû au fait que si S(z) en (2.113-114) n'est pas une fonction en z^N, mais une fonction générale, alors les pré- et post-multiplications par $\mathbf{F}$ en (2.115) produisent une matrice circulante dont les éléments sont les composantes polyphases de S(z). Toutefois, la démonstration ci-dessus reste générale dans le sens que le vecteur $\mathbf{g}(z)$ doit être parallèle au vecteur choisi en (2.134) (en vertu de la partie b) du théorème T2.1).

Résumons brièvement les résultats de cette section. La reconstruction sans repliements spectraux dans le cas d'un banc de filtres d'analyse sous-échantillonné nécessite que le rang de la matrice $\mathbf{H}_m(z)$ ou $\mathbf{H}_p(z)$ soit égal au facteur de sous-échantillonnage. Pour une reconstruction parfaite, une condition suffisante est que le numérateur du déterminant $\Delta_m(z)$ ou $\Delta_p(z)$ soit à phase minimale. Cette condition est également nécessaire pour N=2 ou des filtres modulés lorsque le sous-échantillonnage est critique. Le vecteur des filtres de reconstruction doit être perpendiculaire aux lignes 1 à N-1 de la matrice $\mathbf{H}_m(z)$, et sa direction est donc indiquée par la première ligne des cofacteurs de $\mathbf{H}_m(z)$. Pour la matrice des filtres

polyphases, la condition ci-dessus signifie que les lignes de $\mathbf{G}_p(z)$ doivent être perpendiculaires à celle de $\mathbf{H}_p(z)$ sauf une, et que les produits non nuls doivent tous être égaux. Un éventuel facteur commun à tous les filtres de synthèse se traduit par la multiplication de $\mathbf{G}_p(z)$ par une matrice circulante, mais ne modifie en rien la solution puisqu'il peut être considéré comme un filtre commun placé à la sortie du système.

2.3.3 Résultats concernant les bancs de filtres de synthèse

Considérons un banc de filtres de synthèse, stable et suréchantillonné de façon critique, correspondant à la figure 2.4. De la même façon que pour les bancs d'analyse, nous allons dériver le théorème de reconstruction pour les bancs de filtres de synthèse avec reconstruction des signaux originaux.

Théorème T2.2: Afin d'éviter l'apparition de diaphonie entre les canaux de sortie d'un banc de filtres de synthèse avec reconstruction, il faut que les deux conditions suivantes soient remplies:

a) Il est nécessaire que la matrice $\mathbf{H}_p(z)$ ou $\mathbf{H}_m(z)$ soit non-singulière, c'est-à-dire d'un rang égal à N.

b) Il est nécessaire et suffisant de choisir le vecteur des filtres de synthèse $\mathbf{g}(z)$ égal à la première ligne transposée de la matrice des cofacteurs de $\mathbf{H}_m(z)$ (multipliée par $z^{-N/2}$ si N est pair), multipliée par une matrice diagonale de fonctions rationnelles en z^{-N}.

Evidemment, ce théorème est très similaire au théorème équivalent pour les bancs de filtres d'analyse. La différence sera d'ailleurs montrée à la section suivante qui comparera les deux solutions.

La multiplication par $z^{-N/2}$ dans le cas où N est pair est nécessaire afin d'être en phase avec le sous-échantillonnage de sortie (voir aussi la relation (2.75)).

La partie a) se démontre à nouveau par l'absurde. Selon (2.71), $\mathbf{T}(z)$ doit être une matrice diagonale afin d'éliminer la diaphonie, donc avoir un rang égal à N. Avec (2.66), $\mathbf{T}(z)$ est égal au produit de $\mathbf{H}_m(z)$ par $[\mathbf{G}_m(z)]^T$. Rappelons que le rang de $\mathbf{H}_m(z)$ est au plus égal à N et que le rang d'un produit est borné par le plus petit des rangs des facteurs du produit [kai80]:

$$\text{Rang}[\mathbf{T}(z)] \leq \text{Min}[\ \text{Rang}[\mathbf{H}_m(z)]\ ,\ \text{Rang}[\mathbf{G}_m(z)]\ ] \tag{2.139}$$

Donc, si le rang de $\mathbf{H}_m(z)$ est plus petit que N, il est impossible que $\mathbf{T}(z)$ soit une matrice diagonale, c'est-à-dire que la diaphonie disparaisse.

La nécessité de la partie b) se démontre exactement comme dans le théorème T2.1 et repose sur la direction univoque d'un vecteur normal à N-1 lignes de la matrice $\mathbf{H}_m(z)$ (puisque les lignes de cette matrice sont, selon a), linéairement indépendantes).

Le fait que la condition b) est suffisante se vérifie aisément, puisque si $\mathbf{g}(z)$ est choisi ainsi, alors et en vertu de (2.66):

$$\begin{aligned} \mathbf{T}(z) &= 1/N \cdot \Delta_m(z) \cdot \mathbf{I} && N \text{ impair} \\ &= 1/N \cdot z^{-N/2} \cdot \Delta_m(z) \cdot \mathbf{I} && N \text{ pair} \end{aligned} \tag{2.140}$$

Une multiplication de $\mathbf{g}(z)$ par une matrice diagonale $\mathbf{L}(z^N)$ de fonctions rationnelles en z^{-N} équivaut à prémultiplier $[\mathbf{G}_m(z)]^T$ par $\mathbf{L}(z^N)$, donc $\mathbf{T}(z)$ devient:

$$\mathbf{T}(z) = \mathbf{L}(z^N) \cdot 1/N \cdot \Delta_m(z) \cdot \mathbf{I} \tag{2.141}$$

si N est impair, le tout fois $z^{-N/2}$ si N est pair. Donc, $\mathbf{T}(z)$ en (2.141) reste une matrice diagonale, et la diaphonie est supprimée.

Si nous considérons la matrice de filtres polyphases, nous voyons que la "normalisation" des lignes de $[\mathbf{G}_p(z)]^T$ (voir la remarque R2.14), importante dans le cas des bancs d'analyse, n'est pas nécessaire dans le cas présent, puisque $\mathbf{T}(z)$ peut également s'écrire comme (voir (2.68)):

$$\mathbf{T}(z) = [\mathbf{G}_p(z)]^T \cdot \mathbf{J} \cdot \mathbf{H}_p(z) \tag{2.142}$$

d'où l'on voit que $\mathbf{T}(z)$ peut être une matrice diagonale sans avoir des éléments identiques sur la diagonale. Ceci signifie simplement que les signaux sont reconstruits sans diaphonie mais subissent des altérations différentes selon les canaux. Par contre, on ne peut pas ajouter un filtre quelconque commun à tous les filtres de sortie, comme il était possible de le faire dans le cas du banc d'analyse avec reconstruction, car dès lors les sorties ne sont plus en phase avec le sous-échantillonnage, ce qui conduit à l'apparition de diaphonie entre les canaux.

Cette remarque est importante, car elle montre que les effets d'un canal non-idéal peuvent détruire la propriété d'annulation intrinsèque de la diaphonie.

Des corollaires similaires à Co2.1a-c sont liés au théorème ci-dessus. Nous les donnons sans démonstration ci-après:

Corollaire Co2.2a: Si le suréchantillonnage est surcritique (N'<N), alors la reconstruction sans diaphonie est impossible.

Corollaire Co2.2b: Si le suréchantillonnage est sous-critique (N'>N), il est nécessaire et suffisant que la matrice $\mathbf{H}_m(z)$ (où $\mathbf{H}_p(z)$) de dimension N' x N ait un rang égal au nombre N de signaux afin de reconstruire le signal sans diaphonie.

Corollaire Co2.2c: Pour obtenir une reconstruction parfaite dans le cas d'un suréchantillonnage critique, il est suffisant que les zéros du déterminant soient à l'intérieur du cercle unité. Pour N=2 ou pour des bancs de filtres modulés, cette condition est également nécessaire.

Pour une reconstruction passe-tout, le lecteur intéressé peut se référer à la remarque R2.13.

Résumons brièvement les résultats de cette section. Afin de reconstruire sans diaphonie les signaux d'entrée d'un banc de filtres de synthèse suréchantillonné, il est nécessaire et suffisant que la matrice $\mathbf{H}_m(z)$ ou $\mathbf{H}_p(z)$ ait un rang égal au nombre de signaux d'entrée. Pour une reconstruction parfaite, la condition de phase minimale du déterminant de la matrice de filtres est suffisante. De surcroît, cette condition est nécessaire pour N=2 et pour les bancs de filtres modulés.

2.3.4 Relations entre les bancs d'analyse et de synthèse avec reconstruction

Comme nous l'avons vu, la reconstruction dans le cas de bancs d'analyse ou de synthèse est gouvernée par des propriétés similaires des matrices de filtres correspondantes. Il en résulte que, sous certaines conditions que nous allons dériver, la solution utilisée dans un cas peut être utilisée pour l'autre. Commençons par une remarque préliminaire:

Remarque R2.15: Il est possible d'interchanger sans autre deux à deux les filtres d'entrée et de sortie aussi bien dans un banc d'analyse que dans un banc de synthèse avec reconstruction, et ceci sans perdre la propriété d'annulation des

repliements spectraux ou de la diaphonie, ni éventuellement la propriété de reconstruction parfaite.

Ceci se vérifie aisément, puisque, dans le cas des bancs d'analyse avec reconstruction, on a (voir (2.114)):

$$\mathbf{H}_m(z) \cdot [\mathbf{G}_m(z)]^T = \mathrm{Diag}[\, S(z)\ S(Wz)\ \ldots .\ S(W^{N-1}z)\,] \tag{2.143}$$

Si on transpose (2.143), on obtient:

$$\mathbf{G}_m(z) \cdot [\mathbf{H}_m(z)]^T = \mathrm{Diag}[\, S(z)\ S(Wz)\ \ldots .\ S(W^{N-1}z)\,] \tag{2.144}$$

ce qui signifie que les filtres de sortie se trouvent maintenant à l'entrée, mais ceci sans modifier les propriétés globales concernant la transmission de l'entrée à la sortie du système. Dans le cas des bancs de synthèse avec reconstruction, on trouve similairement:

$$\begin{aligned} [\, [\mathbf{G}_m(z)]^T \cdot \mathbf{H}_m(z)\,]^T &= [\mathbf{H}_m(z)\,]^T \cdot \mathbf{G}_m(z) \\ &= \mathrm{Diag}[\, S_0(z)\ S_1(z)\ \ldots .\ S_{N-1}(z)\,] \end{aligned} \tag{2.145}$$

Donc, la propriété de suppression de la diaphonie ou de la reconstruction parfaite n'est pas affectée si l'on échange les filtres de sortie avec ceux de l'entrée.

Considérons maintenant le théorème suivant, qui définit l'équivalence entre les bancs d'analyse et les bancs de synthèse avec reconstruction:

Théorème T2.3: Prenons une paire de matrices de filtres $\{\mathbf{H}_m(z),\mathbf{G}_m(z)\}$ correspondant à des bancs d'analyse et de synthèse (ou réciproquement). Alors, il est possible d'utiliser ces filtres indifféremment dans un banc d'analyse ou de synthèse avec reconstruction, et ceci tout en conservant la propriété d'annulation des repliements spectraux ou de suppression de la diaphonie, si et seulement si:

$$\mathbf{H}_m(z) \cdot [\mathbf{G}_m(z)]^T = S(z^N) \cdot \mathbf{I} \tag{2.146}$$

La preuve que la condition (2.146) est suffisante est immédiate, car on vérifie alors que:

$$\mathbf{G}_m(z) \cdot [\mathbf{H}_m(z)]^T = \mathbf{S}(z^N) \cdot \mathbf{I} \tag{2.147}$$

La relation (2.146) est nécessaire, car nous savons que le produit diagonal $\mathbf{H}_m(z) \cdot [\mathbf{G}_m(z)]^T$ est commutatif si et seulement si les éléments diagonaux sont tous égaux (voir aussi (2.118)). Quant à la forme particulière de cette matrice, observons que, avec (2.114), on a:

$$\mathbf{H}_m(z) \cdot [\mathbf{G}_m(z)]^T = \text{Diag}[\, S(z)\; S(Wz) \;\ldots.\; S(W^{N-1}z)\,] \tag{2.148}$$

D'autre part, avec (2.71), on a également:

$$\begin{aligned} [\,[\mathbf{G}_m(z)]^T \cdot \mathbf{H}_m(z)\,]^T &= [\mathbf{H}_m(z)\,]^T \cdot \mathbf{G}_m(z) \\ &= \text{Diag}[\, S_0(z^N)\; S_1(z^N) \;\ldots.\; S_{N-1}(z^N)\,] \end{aligned} \tag{2.149}$$

Avec (2.148) et (2.149), on obtient:

$$S_i(z^N) = S_j(z^N) = S(W^k z) = S(W^l z) = S(z^N) \tag{2.150}$$

Illustrons le théorème T2.3 par un diagramme montrant les différentes classes de solutions associées à un banc de filtres d'analyse ou de synthèse. Les filtres d'entrée définissent un vecteur $\mathbf{h}(z)$. A partir de la matrice des filtres modulés $\mathbf{H}_m(z)$ associée à $\mathbf{h}(z)$, on calcule le vecteur des filtres de sortie $\mathbf{g}(z)$ en évaluant la transposée de la première ligne de la matrice des cofacteurs de $\mathbf{H}_m(z)$ (multipliée par $z^{-N/2}$ si N est pair). Dès lors, $\mathbf{H}_m(z) \cdot [\mathbf{G}_m(z)]^T$ remplit la condition donnée par (2.146).

L'ensemble des solutions éliminant les repliements spectraux dans les bancs d'analyse avec reconstruction est donné par:

$$\{\, \mathbf{g}_a(z)\,\} = S(z) \cdot \mathbf{g}(z) \tag{2.151}$$

où S(z) est une fonction arbitraire de z^{-1}. L'ensemble des solutions éliminant la diaphonie dans les bancs de synthèse avec reconstruction est donné par:

$$\{\, \mathbf{g}_s(z)\,\} = \mathbf{S}(z^N) \cdot \mathbf{g}(z) \tag{2.152}$$

où $\mathbf{S}(z^N)$ est une matrice diagonale de fonctions arbitraires en z^{-N}. L'intersection de ces deux ensembles correspond à (2.150). Ceci est illustré dans la figure 2.9:

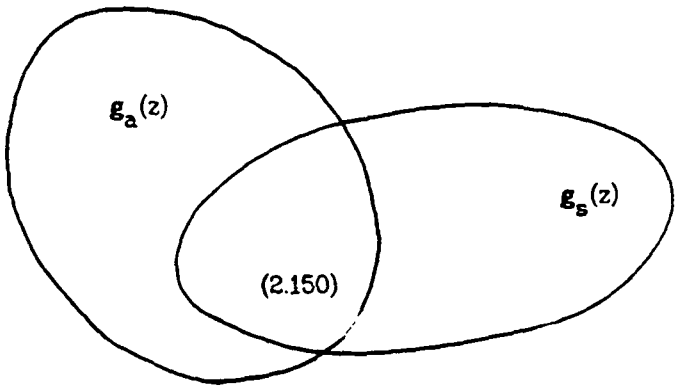

Figure 2.9: Ensemble des solutions pour les bancs d'analyse et de synthèse, étant donné un banc de filtres d'entrée.

Une situation similaire existe pour la reconstruction parfaite.

En résumé, ce paragraphe illustre l'équivalence qui existe entre les bancs d'analyse et de synthèse avec reconstruction. Il montre que ceux-ci sont essentiellement équivalents pour un banc de filtres d'entrée donné. Dans le cas des bancs de filtres d'analyse, la solution pour les filtres de sortie peut être modifiée par la mise en cascade d'un filtre quelconque commun sans pour autant perdre la propriété d'annulation des repliements spectraux. Pour les bancs de filtres de synthèse, la mise en cascade de différents filtres en z^N ne change pas la suppression de la diaphonie. Ces modifications possibles sont les seules différences entre les solutions pour les bancs d'analyse et les bancs de synthèse.

2.4 Principaux résultats du chapitre

Après avoir défini les notions fondamentales de banc de filtres d'analyse (divisant un signal en N signaux par filtrage) et de banc de filtres de synthèse (réunissant N signaux en un seul après filtrage), nous avons posé le problème des bancs de filtres avec reconstruction. Ceux-ci ont en général un débit constant (nombre d'échantillons par unité de temps) à chaque étape du traitement, ce qui donne lieu à un sous- et suréchantillonnage critique des bancs de filtres utilisés.

Afin de traiter efficacement le problème posé, nous avons introduit un formalisme matriciel utilisant des matrices de filtres, et ceci dans deux plans de représentation complémentaires: le plan de modulation et le plan polyphase. Les matrices de filtres dans le plan de modulation ont simultanément été introduites par d'autres chercheurs [ram84,smi85], par contre, la représentation dans le plan polyphase est une généralisation originale du concept polyphase introduit par Bellanger [bel74] dans le cadre des bancs de filtres modulés. Le cadre formel ainsi défini a permis de poser clairement le problème de la reconstruction parfaite et de la reconstruction sans repliement spectral ou diaphonie.

Nous avons ensuite dérivé des résultats généraux concernant les bancs de filtres, résultats qui sont à notre connaissance originaux dans l'ensemble. Les matrices de filtres ont d'abord été analysées en détail (déterminant, cofacteurs, inverse) puis trois théorèmes sur la reconstruction sans repliement spectral, sans diaphonie et sur l'équivalence des deux ont été démontrés.

Références bibliographiques du chapitre 2:

[bel74] M.G.Bellanger, and J.L.Daguet, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT", IEEE Trans. on Communications, Vol. COM-22, No.9, pp. 1199-1204, Sept. 1974.

[bel76] M.G.Bellanger. G.Bonnerot and M.Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample-Rate Alteration and Filter Banks", IEEE Trans. on Acoust., Speech, Signal Processing, Vol. ASSP-24, No.2, pp. 109-114, April 1976.

[bel81] M.G.Bellanger, **Traitement Numérique du Signal**, Masson, Paris, 1981.

[com78] Special Issue on TDM-FDM Conversion, IEEE Trans. on Communications, Vol. COM-26, No.5, May 1978.

[com82] Special Issue on transmultiplexers, IEEE Trans. on Communications, Vol. COM-30, No.7, July 1982.

[crc76] R.E.Crochiere, S.A.Webber, and J.L.Flanagan, "Digital Coding of Speech in Sub-bands", Bell System Technical Journal, Vol.55, No.8, Oct.1976, pp.1069-1085.

[crc83] R.E.Crochiere, and L.R.Rabiner, **Multirate Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1983.

[cro76] A.Croisier, D.Esteban, and C.Galand, "Perfect Channel Splitting by Use of Interpolation, Decimation, Tree Decomposition Techniques", Int. Conf. on Information Sciences/Systems, Patras, pp. 443-446, Aug. 1976.

[est77] D.Esteban, and C.Galand, "Application of Quadrature Mirror Filters to Split-Band Coding", Int. Conf. on ASSP, pp.191-195, Hartford, May 1977.

[gal83] C.Galand, "Codage en Sous-Bandes: Théorie et Application à la Compression Numérique du Signal de Parole", Thèse d'Etat, Université de Nice, 1983.

[gal84] C.R.Galand, and H.J.Nussbaumer, "New Quadrature Mirror Filter Structures", IEEE Trans. on Acoust., Speech and Signal Proc., Vol.32, No.3, June 1984, pp.522-531.

[kai80] T.Kailath, **Linear Systems**, Prentice-Hall, Englewood Cliffs, 1980.

[kun80] M.Kunt, **Traitement Numérique des Signaux**, Editions Georgi, 1980.

[opp75] A.V.Oppenheim, and R.W.Schafer, **Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1975.

[rab75] L.R.Rabiner, and B.Gold, **Theory and Application of Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1975.

[ram84] T.A.Ramstad, "Analysis/synthesis filterbanks with critical sampling", Intl. Conf. on DSP, Florence, Sept.1984, pp. 130-134.

[sha73] R.W.Shafer, and L.R.Rabiner, "A Digital Signal Processing Approach to Interpolation", Proc. IEEE, Vol.61, No.6, pp. 692-702, June 1973.

[smi85] M.J.T.Smith, T.P.Barnwell, "A Unifying Framework for Analysis/Synthesis Systems Based on Maximally Decimated Filter Banks", Proc. IEEE ICASSP-85, pp. 521-524, Tampa, March 1985.

[str80] G.Strang, **Linear Algebra and Its Applications**, Academic Press, New York, 1980.

[vet84a] M.Vetterli, "Multi-Dimensional Sub-Band Coding: Some Theory and Algorithms", Signal Processing, Vol. 6, No.2, pp. 97-112, Feb. 1984.

[vet85c] M.Vetterli, "Splitting a Signal into Subsampled Channels Allowing Perfect Reconstruction", Proc. of the IASTED Conf. on Applied Signal Processing and Digital Filtering, Paris, June 1985.

[vet86b] M.Vetterli, "Filter Banks Allowing Perfect Reconstruction", à paraître dans Signal Processing, Avril 1986.

3 Synthèse de bancs de filtres

Ce chapitre explore d'une part la synthèse de filtres en vue d'une utilisation dans des bancs de filtres, et d'autre part, il s'intéresse à la synthèse de bancs de filtres pour des problèmes réels (codage en sous-bandes, transmultiplexeurs).

Dans ce qui suit, nous n'avons pas à proprement parler conçu des filtres prototypes qu'un utilisateur potentiel pourrait simplement copier pour son application propre. Deux raisons sont à la base de ce fait. D'abord, le but du travail présenté n'était pas la conception de filtres. Il s'avère d'ailleurs que ce problème, dans le cadre des bancs de filtres, est relativement complexe, et nécessitera un travail d'une envergure telle qu'il ne nous était pas possible de l'envisager. Ensuite, et au vu des résultats nouveaux obtenus, (par ex. sur la reconstruction parfaite) il n'est plus tout à fait clair ce que serait un "bon filtre". Il semble que seule l'expérience avec ces nouveaux bancs de filtres montrera quels sont les critères dont il faut tenir compte pour la conception de filtres pour un banc de filtres. Notons à ce propos l'importance que prennent les composantes polyphases d'un filtre, ainsi que les compromis possibles entre de "bons filtres" et une bonne fonction de transfert totale (entrée-sortie). Notons qu'il n'a pas été possible, faute de place et de temps, de mettre tous les résultats du chapitre 2 en pratique.

Le chapitre s'organise comme suit. Une première section traite le cas des bancs de deux filtres. Pour le cas RIF, on donne deux méthodes de conception analytiques permettant la reconstruction parfaite, ainsi qu'une méthode d'optimisation simultanée. Les filtres RII et l'application des résultats aux transmultiplexeurs viennent ensuite, et enfin, l'effet de canaux non-idéaux est considéré. La deuxième section est consacrée aux bancs de N filtres généraux, où l'on donne en particulier une méthode pour dériver des bancs d'analyse et de synthèse RIF permettant la reconstruction parfaite. La section suivante analyse les bancs de N filtres modulés ainsi que le cas important des bancs pseudo-QMF. La quatrième section présente un filtre QMF complexe [nus83a] avec reconstruction parfaite, et la cinquième section présente la généralisation multidimensionnelle de l'analyse en sous-bandes.

En ce qui concerne la nouveauté des résultats présentés, notons que les méthodes de synthèse analytique pour bancs RIF avec reconstruction parfaite (sections 3.1.1c, 3.1.1d et 3.2.1) sont originales [vet86b], ainsi que le filtre QMF complexe parfait (section 3.4) et le cas QMF bidimensionnel [vet84a]. Les bancs pseudo-QMF (section 3.3.2) présentés proviennent de travaux auxquels nous avons pris part [nus84a,nus84b]. Finalement, nombres de remarques qui apparaissent en cours de chapitre (par ex. à la section 3.1.4 [vet86c]) sont également originales à notre connaissance.

3.1 Bancs de deux filtres

Le cas particulier où N=2 est analysé ci-dessous. D'une part, ce cas est important pour les applications en raison de sa simplicité et du fait qu'il est à la base d'une décomposition en bandes à largeur relative égale (d'où son importance en traitement de la parole [gal83]). D'autre part et d'un point de vue plus théorique, il est possible de trouver des solutions analytiques à certains problèmes (alors que cela ne sera plus le cas pour N>2). Nous traiterons tour à tour le cas à réponse impulsionnelle finie (RIF) puis le cas à réponse impulsionnelle infinie (RII).

3.1.1 Filtres à réponse impulsionnelle finie

Les filtres non-récursifs (RIF) possèdent trois caractéristiques désirables: ils sont toujours stables, leur conditionnement numérique est bon, et ils peuvent réaliser une fonction de transfert ayant une phase exactement linéaire. Le prix à payer est une complexité de calcul plus élevée par rapport à un filtre récursif (RII) réalisant une fonction de transfert comparable. Dans le cadre des bancs de filtres avec reconstruction, les filtres non-récursifs ont une quatrième propriété désirable que nous décrivons par la remarque suivante:

Remarque R3.1: Seuls des filtres RIF peuvent réaliser une reconstruction parfaite n'impliquant pas une annulation implicite de pôles par des zéros, et ceci pour N arbitraire.

L'annulation implicite d'un pôle par un zéro apparaît lorsqu'une simplification pôle/zéro est obtenue entre deux filtres mis en cascade mais physiquement bien séparés. Si mathématiquement, cette simplification est aisée, numériquement elle est souvent mal conditionnée [kai80,gol80] et donc si possible à éviter. Concernant la remarque R3.1, il est évident que si tous les filtres impliqués sont RIF, il n'y a pas de simplification implicite possible. La preuve que les filtres RIF sont également nécessaires repose sur le fait que la fonction de transfert totale ne doit pas avoir de pôles (ailleurs qu'en z=0 bien sûr), et nous l'avons donnée en annexe a) de [vet86b].

Ci-dessous, nous allons montrer comment il est possible, dans un système d'analyse à deux canaux, de reconstruire le signal sans repliement spectral, et éventuellement parfaitement, ceci en n'utilisant que des filtres RIF tant pour l'analyse que pour la synthèse. En vertu du théorème T2.3 sur l'équivalence des systèmes d'analyse et de synthèse, des relations similaires existent pour les systèmes de synthèse avec reconstruction, et elles seront également évoquées à la suite des résultats sur les bancs d'analyse.

Rappelons que la sortie d'un système comme celui de la figure 2.3 est égal, lorsque N=2 et en vertu de (2.57), à:

$$\hat{X}(z) = 1/2 \; [\; G_0(z) \; G_1(z) \;] \cdot \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \cdot \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix} \tag{3.1}$$

Afin d'éviter tout repliement spectral, il est nécessaire et suffisant de choisir (selon le théorème T2.1) $\mathbf{g}(z)$ comme égal à la première ligne de $\mathbf{C}_m(z)$ (qui est la matrice des cofacteurs de $\mathbf{H}_m(z)$), donc:

$$\mathbf{g}(z) = [\; H_1(-z) \; -H_0(-z) \;]^T \tag{3.2}$$

Notons que de ce fait les filtres de synthèse sont RIF. Le signal reconstruit devient:

$$\hat{X}(z) = 1/2 \cdot \Delta_m(z) \cdot X(z) \tag{3.3}$$

où:

$$\Delta_m(z) = H_0(z) \cdot H_1(-z) - H_0(-z) \cdot H_1(z) \tag{3.4}$$

En utilisant la représentation polyphase, le relation (3.4) peut s'écrire, en vertu de (2.36,2.59) et du fait que les filtres sont RIF, comme:

$$\Delta_m(z) = 2 \cdot z^{-1} \cdot [H_{00}(z^2) \cdot H_{11}(z^2) - H_{01}(z^2) \cdot H_{10}(z^2) \;] \tag{3.5}$$

Evidemment, si (3.4,3.5) se réduisent à un monôme (un polynôme possédant un seul coefficient différent de zéro), on obtient une reconstruction parfaite. Nous obtenons la condition suivante, par ailleurs valable pour N arbitraire:

Condition C3.1: Afin qu'une reconstruction parfaite soit possible avec un système n'utilisant que des filtres RIF, il est nécessaire et suffisant que le déterminant de la matrice de filtres soit un monôme.

Si seule l'annihilation des repliements spectraux est désirée, alors la remarque suivante peut être faite, valable également pour N arbitraire:

Remarque R3.2: Une reconstruction sans repliements spectraux et n'utilisant que des filtres RIF est toujours possible, indépendamment des filtres d'analyse choisis, puisque la matrice des cofacteurs (qui définit les filtres de synthèse) correspond forcément à des filtres RIF.

Nous avons donc établi les conditions de reconstruction sans repliement spectral (relation (3.2) et remarque ci-dessus) ainsi que de reconstruction parfaite (condition C3.1) pour des systèmes RIF. Avant de poursuivre avec nos développements propres, considérons brièvement les résultats existants sur l'annulation du repliement spectral et la reconstruction idéale.

a) Filtres miroirs en quadrature classiques

Introduit dans [cro76], les filtres miroirs en quadrature (que nous appellerons QMF en accord avec la littérature [gal84]) donnent une solution simple et efficace au problème de l'annulation du repliement spectral. En posant:

$$H_0(z) = H(z)$$
$$H_1(z) = H(-z) \qquad (3.6)$$

et en choisissant, selon (3.2), $\mathbf{g}(z)$ égal à:

$$\mathbf{g}(z) = [\ H(z)\ -H(-z)\ ]^T \qquad (3.7)$$

on obtient bien la suppression du repliement spectral, ce qui est le résultat à la base de tous les développements réalisés autour des QMF [gal83,gal84]. Notons toutefois qu'en (3.6,3.7), nous n'avons pas requis des filtres RIF, ni des filtres à phase linéaire. Nous appellerons donc le système (3.6,3.7) un système QMF au sens large, puisque habituellement, seul un filtre RIF à phase linéaire est considéré [gal84].

Pour obtenir une reconstruction parfaite, seul un filtre H(z) RIF de longueur 2 est possible (une longueur qui est en général insuffisante pour des systèmes réels). La preuve s'établit comme suit. Le déterminant $\Delta_m(z)$ dans (3.5) devient, en raison de (3.6):

$$\Delta_m(z) = 4 \cdot z^{-1} \cdot H_p(z^2) \cdot H_i(z^2) \qquad (3.8)$$

où $H_p(z^2)$ et $H_i(z^2)$ correspondent aux coefficients d'indices pairs et impairs de H(z). Le déterminant $\Delta_m(z)$ dans (3.8) ne peut être un monôme que si $H_p(z^2)$ et $H_i(z^2)$ sont tous deux des monômes, donc H(z) doit être exactement de longueur 2.

En dépit du fait que les filtres QMF de longueur supérieure à 2 ne peuvent pas réaliser une reconstruction parfaite, de très bonnes approximations peuvent être réalisées. Rappelons brièvement ce problème d'approximation dans le cas des QMF classiques. Considérons un filtre H(z) symétrique (donc à phase linéaire) et de longueur paire. L'évaluation de la transformée en z sur le cercle unité permet d'écrire la réponse fréquentielle du filtre comme suit [rab75]:

$$H(e^{j\omega}) = e^{-j\omega(M-1)/2} \cdot H_a(\omega) \tag{3.9}$$

où ω est la fréquence normalisée, M la longueur du filtre (un nombre pair arbitraire) et $H_a(\omega)$ une fonction réelle de ω exprimant la norme de la fonction de transfert (la phase étant linéaire et donnée par le phaseur complexe de norme 1 qui prémultiplie $H_a(\omega)$ dans (3.9)). Puisque nous avons la relation suivante:

$$H(-z)\,|_{z=e^{j\omega}} = H(e^{j(\omega+\pi)}) \tag{3.10}$$

nous pouvons récrire (3.4) en utilisant (3.9) comme:

$$\begin{aligned} \Delta_m(e^{j\omega}) &= e^{-j\omega(M-1)} \cdot H_a^2(\omega) - e^{-j(\omega+\pi)(M-1)} \cdot H_a^2(\omega) \\ &= e^{-j\omega(M-1)} \cdot [\, H_a^2(\omega) + H_a^2(\omega+\pi) \,] \end{aligned} \tag{3.11}$$

Donc, la fonction de transfert du système total correspond à un filtre RIF symétrique de longueur 2M-1 (la phase correspond à un filtre d'une telle longueur, mais la longueur effective du filtre est de 2M-3 puisque tous les coefficients à indices pairs sont nuls, donc en particulier le premier et le dernier coefficient). La norme $H_t(\omega)$ de cette fonction de transfert, donnée par

$$H_t(\omega) = H_a^2(\omega) + H_a^2(\omega+\pi) \tag{3.12}$$

est parfaitement plane sur l'axe des fréquences (à part le cas où M=2) si $[H_a(\omega)]^2$ correspond à un filtre passe-bas idéal de demi-bande (en particulier, $H_a(\pi/2)$ doit valoir $1/\sqrt{2}$ si le filtre est réel). Le cas où M est impair donne lieu à un problème d'approximation similaire [gal83,gal84] (l'atténuation de $1/\sqrt{2}$ à $\pi/2$ n'est alors pas possible et doit être corrigée à la sortie). Notons simplement qu'il est alors nécessaire de mettre un délai dans une des branches d'analyse, ainsi qu'un délai dans l'autre branche de synthèse. Le signe moins dans la relation (3.4) devient alors un signe plus, mais l'analyse reste globalement inchangée. Le cas du système avec délai est traité dans l'annexe A3.1 à laquelle nous renvoyons pour plus de détails.

Pour conclure ce bref appercu des QMF classiques, notons que dans ce cas, un seul filtre (en l'occurrence H(z) de (3.6)) doit être synthetisé, mais sous la contrainte que la relation (3.12) doit exprimer une transmission aussi idéale que possible. Ce problème a été considéré dans [joh80] où des résultats d'optimisation sont proposés.

b) Solution à phase maximum/minimum

Un cas particulier de reconstruction parfaite à été proposé dans [smi84] et nous le

décrivons succinctement ci-après. Prenons un filtre H(z), RIF de longueur paire M et dont la fonction d'autocorrélation, définie comme suit,

$$ACF[H(z)] = H(z)\cdot H(z^{-1}) = \sum_{n=-M+1}^{M-1} a_n \cdot z^{-n} \tag{3.13}$$

a la propriété suivante:

$$\begin{aligned} a_0 &= 1/2 \\ a_{2i} &= 0 \quad i\neq 0 \\ a_{2i+1} &\text{ arbitraire} \end{aligned} \tag{3.14}$$

Choisissons maintenant les filtres d'analyse comme indiqué ci-dessous:

$$\begin{aligned} H_0(z) &= H(z) \\ H_1(z) &= -z^{-M+1}\cdot H(-z^{-1}) \end{aligned} \tag{3.15}$$

Dès lors, le déterminant de la matrice de filtres devient, en vertu de (3.4, 3.13 et 3.14), égal à:

$$\begin{aligned} \Delta_m(z) &= z^{-M+1}\ [\ H(z)\cdot H(z^{-1}) + H(-z)\cdot H(-z^{-1})\ ] \\ &= z^{-M+1} \end{aligned} \tag{3.16}$$

puisque la somme entre parenthèse fait disparaître tous les termes à indices impairs de la fonction d'autocorrélation et double ceux d'indices pairs. Il suffit maintenant de choisir les filtres de synthèse selon (3.2), et l'on obtient bien un système avec reconstruction parfaite. Notons, pour la petite histoire, que si l'idée de base se trouve bien dans [smi84], la dérivation qui s'y trouve est fausse! (en particulier des délais sont introduits, et il est aisé de vérifier qu'alors la reconstruction parfaite n'est pas obtenue). En fait, avec la méthode introduite dans [smi84] et basée sur l'autocorrélation d'un filtre prototype, seul un système sans délai et utilisant des filtres de longueur paire peut accomplir la reconstruction parfaite. En annexe A3.1, il est démontré qu'un système avec délai et ayant des filtres de même longueur ne peut pas réaliser une reconstruction parfaite. De plus, si la longueur des filtres est impaire, on peut vérifier que tous les éléments du déterminant apparaissent deux fois, et qu'il ne peut donc pas se réduire à un monôme.

Le problème de conception se réduit donc à trouver un bon filtre passe-bas dont la fonction d'autocorrélation satisfasse la contrainte donnée par (3.14). Dans [smi84], il

est montré que ce résultat peut être obtenu avec une méthode utilisée pour la conception de filtres à phase minimale [her70]. Notons toutefois que l'on obtient alors un banc d'analyse comportant un filtre passe-bas à phase minimale et un filtre passe-haut à phase maximale. Nous sommes donc bien loin d'obtenir des signaux de canaux en phase comme c'est le cas avec des filtres d'analyse à phase linéaire (QMF classiques). De surcroît, le délai total entrée-sortie est du même ordre que dans le cas à phase linéaire.

Nous ne nous étendrons pas plus sur ces méthodes de conception, puisqu'elles sont connues, mais dérivons ci-dessous deux autres méthodes originales et générales.

c) Méthode de la factorisation

La reconstruction parfaite exige que le déterminant soit un monôme (condition C3.1). Prenons donc un polynôme P(z) de degré M-1 ayant la propriété:

$$P(z) - P(-z) = 2 \cdot p_{2k+1} \cdot z^{-2k-1} \qquad 0 \leq k \leq \lfloor (M-2)/2 \rfloor \tag{3.17}$$

Ce polynôme possède M-1 zéros que l'on peut séparer en deux groupes de M_0-1 et M_1-1 zéros chacun. Ces deux groupes définissent donc deux polynômes $P_0(z)$ et $P_1(z)$, de degré M_0-1 et M_1-1 respectivement, et la relation suivante lie P(z), $P_0(z)$ et $P_1(z)$:

$$P(z) = P_0(z) \cdot P_1(z): \tag{3.18}$$

De par cette construction, il est évident que si les filtres d'analyse sont choisis avec $H_0(z)=P_0(z)$ et $H_1(z)=P_1(-z)$, le déterminant de la matrice de filtres équivalente est égal à:

$$\begin{aligned} \Delta_m(z) &= H_0(z) \cdot H_1(-z) - H_0(-z) \cdot H_1(z) \\ &= P(z) - P(-z) \\ &= 2 \cdot p_{2k+1} \cdot z^{-2k-1} \end{aligned} \tag{3.19}$$

En choisissant les filtres de synthèse conformément à (3.2), donc égaux à $G_0(z)=P_1(z)$ et $G_1(z)=-P_0(-z)$, le signal reconstruit sera bien équivalent au signal original, multiplié par une constante (p_{2k+1}) et avec un délai de z^{-2k-1}.

Considérons plus en détail la factorisation exprimée par la relation (3.18). Nous remarquons que le filtre produit P(z) résulte de la convolution du filtre $H_0(z)$ par le filtre $H_1(z)$ modulé par $(-1)^n$ (donc déplacé de π sur l'axe des fréquences normalisées). Si le filtre produit est par exemple un filtre passe-bas et que la

factorisation en (3.18) est faite de façon à obtenir deux polynômes $P_0(z)$ et $P_1(z)$ correspondant à des filtres passe-bas, alors on obtient bien un système d'analyse comportant un filtre passe-bas d'une part, et passe-haut d'autre part. Avant de poursuivre, dérivons un exemple simple qui illustre cette méthode et certains des problèmes qui lui sont liés.

Exemple E3.1: Prenons la réponse impulsionnelle d'un filtre passe-bas idéal de demi-bande, donnée par sa transformée en z:

$$H_{id}(z) = \sum_{n=-\infty}^{\infty} \frac{\sin(n\pi/2)}{(n\pi/2)} \cdot z^{-n} \tag{3.20}$$

Tronquons cette réponse impulsionnelle à la partie comprise entre -2k-1 et 2k+1, multiplions par une fenêtre de pondération w(n) et appliquons un délai de z^{-2k-1} afin de la rendre causale. Le résultat de ces opérations, P(z), est égal à:

$$P(z) = \sum_{n=0}^{4k+2} w(n-2k-1) \cdot \frac{\sin((n-2k-1)\pi/2)}{((n-2k-1)\pi/2)} \cdot z^{-n} \tag{3.21}$$

Ceci est un filtre passe-bas de longueur 4K+3 et à phase linéaire (si w(n) est symétrique, ce qui est le cas en général). De surcroît, il vérifie bien la propriété exprimée par (3.17).

Maintenant, on peut chercher les 4k+2 zéros de P(z), puis les séparer en deux groupes (par exemple de 2k+1 chacun) afin d'obtenir les facteurs $P_0(z)$ et $P_1(z)$ comme en (3.18). Un partage possible consiste à choisir $P_0(z)$ comme la composante à phase minimale et $P_1(z)$ celle à phase maximale de P(z) (le cas de zéros sur le cercle unité sera traité plus loin). Un autre partage possible consiste à conserver la propriété de phase linéaire aux deux facteurs $P_0(z)$ et $P_1(z)$ (les détails seront également donnés plus loin). La figure 3.1 illustre le procédé décrit ci-dessus pour k=3. Les parties a) et b) montrent l'amplitude de la réponse fréquentielle du filtre idéal tronqué avant et après pondération par la fenêtre w(n) (Hamming). En c) et d), on voit l'amplitude des réponses fréquentielles de $P_0(z)$ et de $P_1(z)$ (qui sont à phase minimale et maximale) et enfin en e) et f), un choix de $P_0(z)$ et de $P_1(z)$ à phase linéaire est montré. Nous remarquons que les fonctions d'amplitude en c) et d) correspondent bien à de bons filtres demi-bandes mais avec une phase évidemment non-linéaire. Par contre, si la phase des filtres en e) et f) est parfaitement linéaire, leurs fonctions d'amplitude sont loin d'être bonnes (même si le produit des deux donne la fonction d'amplitude en b)).

Ce petit exemple a le mérite d'illustrer les problèmes liés à la méthode de la factorisation et qui sont résumés dans la remarque suivante:

Remarque R3.3: Il est relativement difficile de concevoir un filtre produit P(z) en vue d'obtenir certaines caractéristiques pour ses facteurs $P_0(z)$ et $P_1(z)$. Deux raisons contribuent à cet état de fait:

i) Si l'on prend deux filtres passe-bas ayant de bonnes caractéristiques, leur produit aura de bonnes caractéristiques. La réciproque, par contre, n'est pas vraie.

ii) La partition des zéros de P(z) est un problème combinatoire, et le nombre de

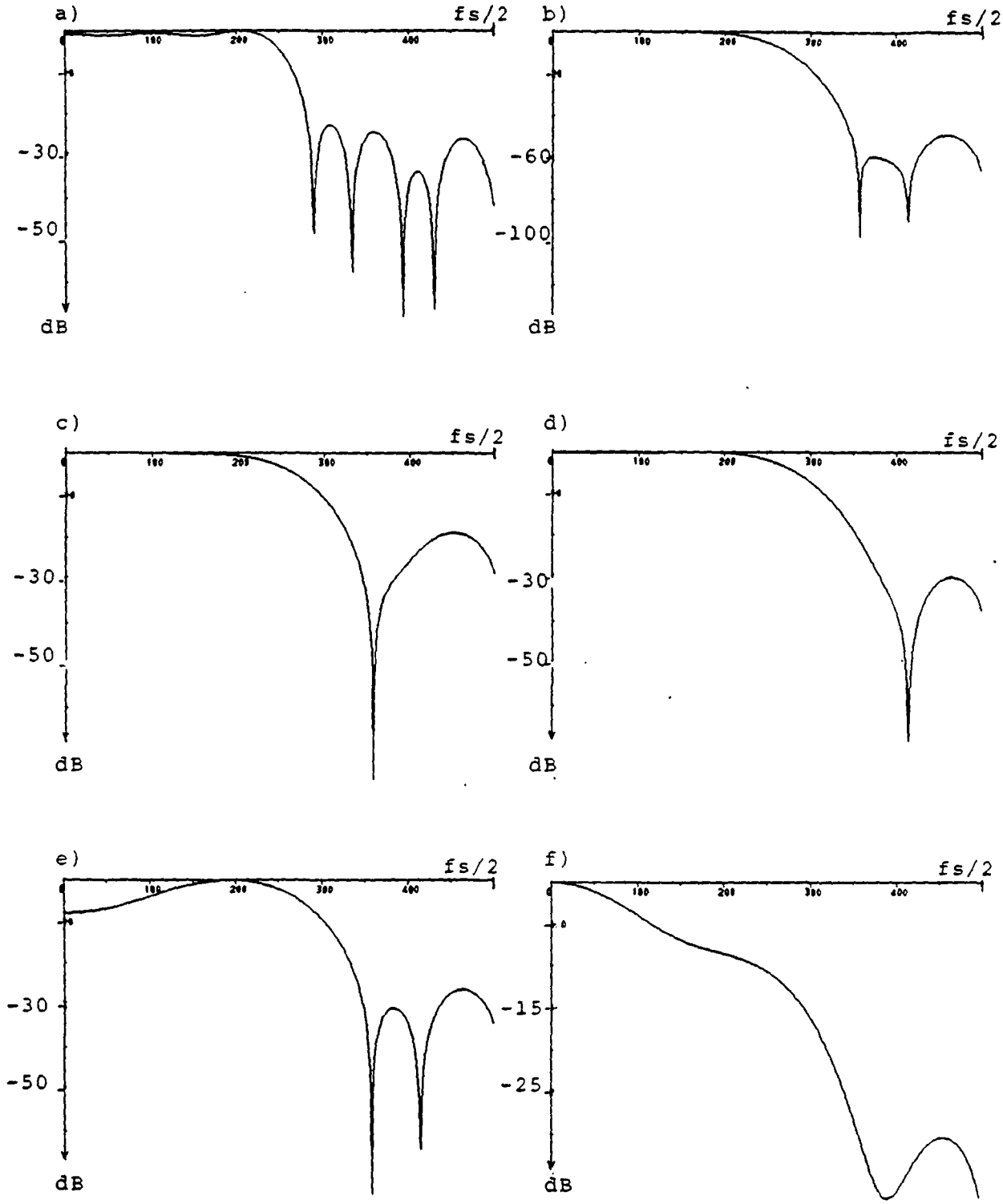

Figure 3.1: Exemple de filtres obtenus avec la méthode de la factorisation (exemple E3.1)

a) norme de la fonction de transfert du filtre produit idéal et tronqué

b) norme de la fonction de transfert du filtre produit après pondération par une fenêtre de Hamming

c) norme de la fonction de transfert de $P_0(z)$ à phase minimale

d) norme de la fonction de transfert de $P_1(z)$ à phase maximale

e) norme de la fonction de transfert de $P_0(z)$ à phase linéaire

f) norme de la fonction de transfert de $P_1(z)$ à phase linéaire

solutions croît exponentiellement avec le nombre de zéros.

Cette remarque, loin de discréditer la méthode de la factorisation, vise à bien placer les problèmes à résoudre. En outre, deux cas particuliers échappent aux limitations esquissées ci-dessus:

- si $P_0(z)$ et $P_1(z)$ sont les composantes à phase minimale et maximale de P(z), alors il y a une certaine réciprocité entre la qualité des fonctions d'amplitude des facteurs et de leur produit.
- si le filtre produit est d'ordre élevé, alors la densité des zéros est en général telle qu'il suffit de répartir ces derniers de façon équitable (pour chaque paire de zéros voisins, attribuer l'un à $P_0(z)$ et l'autre à $P_1(z)$) afin d'obtenir deux filtres de bonne qualité.

Notons que ces deux cas ne sont pas les seuls présentant un intérêt. Les problèmes liés aux solutions à phase maximale/minimale ont été discutés au point b) de cette section, et si l'ordre des filtres est élevé, on peut tout aussi bien utiliser des QMF classiques qui permettent une reconstruction quasi-parfaite.

Après cette discussion qualitative de la méthode de la factorisation, nous allons dériver quelques résultats plus formels. D'abord, montrons que la nature combinatoire du problème de la partition des zéros en deux groupes mène rapidement à un nombre de solutions gigantesque.

La partition des M-1 zéros de P(z) en deux ensembles disjoints et dont la réunion forme l'ensemble des zéros de P(z) a le nombre suivant de solutions différentes:

$$n_0 = 2^{M-2} - 1 \tag{3.22}$$

Il est raisonnable de penser que l'on cherche avant tout à obtenir deux filtres facteurs de même longueur. Donc en admettant M impair, nous voulons diviser les M-1 zéros en deux ensembles disjoints de (M-1)/2 éléments chacun. Dans ce cas, il y a le nombre suivant de solutions (en partant de la formule pour le nombre de combinaisons [olz74]):

$$n_1 = \frac{(M-1)!}{2\cdot[((M-1)/2)!]^2} \tag{3.23}$$

Ce nombre reste à croissance exponentielle, puisqu'en appliquant la formule de Stirling [ang72] pour les composantes factorielles en (3.23), on obtient l'approximation suivante:

$$n_1 \approx \frac{2^{(M-1)}}{\sqrt{2\pi(M-1)}} \tag{3.24}$$

Remarquons que les zéros apparaissent souvent par groupes (voir plus bas), et que ces groupes ne seront pas scindés en général et ceci afin de conserver certaines propriétés aux polynômes $P_0(z)$ et $P_1(z)$ (par exemple d'être réels ou symétriques). Dès lors, le nombre d'éléments intervenant dans la partition est fortement réduit, typiquement d'un facteur 2 à 4. Néanmoins, le nombre de solutions reste grand. Prenons en exemple la partition d'un polynôme de degré 30 afin d'obtenir deux filtres d'analyse. Alors, (3.22) indique plus de 10^9 solutions différentes, et (3.23) environ $7 \cdot 10^7$ partitions en deux groupes d'égale grandeur. Si nous contraignons les polynômes à être réels, la plupart des zéros se retrouvent par paire. Dans ce cas et en admettant 16 groupes, (3.23) donne toujours plus de six mille solutions. Finalement, si nous désirons des filtres symétriques, la plupart des zéros seront quadruples. Prenons l'exemple de 8 groupes, ce qui limite à 35 le nombre de solutions possibles à examiner. Ceci reste un nombre non-négligeable, alors que les filtres ne sont pas encore d'ordre très élevé (longueur 16).

Considérons maintenant les groupes de zéros possibles. Dans ce qui suit, nous admettrons toujours que $P(z)$, $P_0(z)$ et $P_1(z)$ sont des filtres passe-bas réels, typiquement de demi-bande (les autres cas sont obtenus par modulation). Rappelons que les zéros d'un polynôme réel appartiennent à deux classes distinctes:

R1: $\{(z-a_r)\}$, zéros isolés, placés sur l'axe des réels

R2: $\{(z-a_c),(z-a_c^*)\}$, zéros conjugués par paire, situés ailleurs que sur l'axe des réels.

Les zéros d'un polynôme correspondant à un filtre symétrique réel ont la propriété (en plus de celles de polynômes réels) que leurs inverses sont également des zéros du polynôme. Il en résulte que les zéros peuvent être regroupés en 4 classes distinctes:

RS1: $\{(z-1),(z+1)\}$, zéros isolés situés à z=1 ou z=-1. Notons que si le filtre est passe-bas, le cas z=1 est exclu. Si de plus, sa longueur est paire, il en résulte automatiquement un zéro à z=-1.

RS2: $\{(z-a_r),(z-1/a_r)\}$, zéros par paire, situés sur l'axe des réels ailleurs qu'en z=1 ou z=-1.

RS3: $\{(z-a_1),(z-a_1^*)\}$, zéros conjugués par paire, situé sur le cercle unité ailleurs qu'en z=1 ou z=-1.

RS4: $\{(z-a_1),(z-a_1^*),(z-1/a_r),(z-1/a_r^*)\}$, zéros conjugués par quadruple, ailleurs que sur l'axe des réels ou le cercle unité.

Ces groupes doivent être respectés lors de la factorisation: si l'on désire que $P_0(z)$ et $P_1(z)$ soient des filtres réels (P(z) étant réel), il faut garder les paires correspondant à la classe R2 intactes, et de même, si l'on désire obtenir des filtres symétriques (P(z) étant symétrique), il faut maintenir les paires et quadruples correspondant à RS2, RS3 et RS4.

Une configuration qui facilite la partition est évidemment le zéro double. D'ailleurs, dans [smi84], on utilise une méthode de doublement de zéros sur le cercle unité [her70] avant de factoriser P(z). Par contre, nous savons que P(z) ne peut être un carré parfait (tous les zéros doubles), car ce cas peut être ramené au QMF classique, pour lequel nous avons montré qu'une reconstruction parfaite n'était pas possible (pour des filtres d'une longueur supérieure à 2). Une stratégie de répartition simple est la suivante: en avançant le long du cercle unité, on répartit les zéros rencontrés à tour de rôle à $P_0(z)$ et à $P_1(z)$, tout en respectant l'intégrité des groupes de zéros (pour assurer des filtres réels, voire symétriques). Outre la partition à phase minimale/maximale de [smi84], on peut tenter d'approximer une phase linéaire en alternant l'attribution de zéros à l'intérieur du cercle unité avec celle de zéros sis à l'extérieur. Quoique également proposée dans [smi84], cette méthode est non satisfaisante, et nous allons montrer comment réaliser une partition qui permet une phase exactement linéaire. Une remarque préliminaire définit l'ensemble des solutions possibles:

Remarque R3.4: Un système d'analyse avec reconstruction parfaite et n'utilisant que des filtres à phase linéaire existe si et seulement si:
- la longueur des filtres, M, est paire
- $H_0(z)$ est symétrique et $H_1(z)$ antisymétrique ou réciproquement

La preuve de cette affirmation est donnée en annexe A3.2. Le résultat immédiat est que par conséquent $P_0(z)$ et $P_1(z)$ correspondent tous deux à des filtres symétriques (en vertu de (a3.3)) et ont donc chacun un zéro en z=-1 (puisqu'ils sont de longueur paire).

Afin de dériver deux filtres à phase linéaire, prenons d'abord un polynôme P(z) correspondant à un filtre symétrique (en vertu de (a3.3) puisque $P_0(z)$ et $P_1(z)$ doivent être symétriques). La longueur du filtre produit est impaire et peut s'écrire, puisque M est pair, comme:

$$M_t = 2M-1 = 4M'-1, \qquad M' = 1,2,... \tag{3.25}$$

Maintenant, afin que P(z) puisse être mis en facteurs $P_0(z)$ et $P_1(z)$ ayant les caractéristiques désirées, il est nécessaire (outre le fait que ses zéros doivent être groupés selon les classes RS2-RS4) que P(z) possède un double zéro en z=-1 (un pour chacun des facteurs). Ceci n'est pas forcément le cas (car la longueur du filtre correspondant à P(z) est impaire) mais peut facilement être obtenu en modifiant P(z) de la façon suivante:

$$P'(z) = P(z) - P(-1) \tag{3.26}$$

Evidemment, P'(-1)=0, mais est-ce bien un zéro double?. L'argument suivant le montre aisément: P'(z) a un nombre pair de zéros et ceux-ci apparaissent par groupes de deux ou quatre, sauf en z=1 ou z=-1. Comme un zéro en z=1 est exclu (car c'est un filtre passe-bas), le nombre de zéros en z=-1 est forcément un nombre pair. Donc, P'(z) peut s'écrire sous la forme:

$$P'(z) = (z+1)\, P'_0(z) \cdot (z+1)\, P'_1(z) \tag{3.28}$$

où $P'_0(z)$ et $P'_1(z)$ sont choisis en maintenant l'intégrité des groupes RS2 à RS4. Il est clair que:

$$\begin{aligned} P_0(z) &= (z+1)\, P'_0(z) \\ P_1(z) &= (z+1)\, P'_1(z) \end{aligned} \tag{3.27}$$

sont les transformées en z de filtres réels et symétriques. Notons qu'il n'est pas toujours possible de réaliser la factorisation en (3.27) de façon à ce que $P_0(z)$ et $P_1(z)$ soient de même degrés (par exemple si M_t=7 et si P'(z) a, outre le zéro double en z=-1, un groupe de zéros du type RS4). A nouveau, remarquons que cette partition peut s'avérer délicate, en particulier si le degré de P'(z) est relativement faible, et nous rappelons l'exemple de la figure 3.1 e) et f) afin d'illustrer ce fait.

Pour conclure cette section sur la méthode de la factorisation, énumérons les résultats principaux qui ont été obtenus. Etant donné un polynôme de départ P(z) respectant la condition en (3.17), n'importe quelle factorisation du type en (3.18) permet une reconstruction parfaite. Mais ceci ne concerne évidemment que la fonction de transfert totale. Si, de surcroît, on désire satisfaire certaines contraintes sur la caractéristique des filtres d'analyse (donc de synthèse), la répartition des zéros de P(z) devient plus délicate, d'une part en raison du nombre de solutions à examiner (lié à la nature combinatoire du problème) et d'autre part à cause du fait

que les caractéristiques du filtre produit n'apparaissent pas forcément dans ses facteurs. Finalement, on a montré quelle était la classe des solutions à phase linéaire possibles (remarque R3.4) et comment transformer, puis factoriser P(z) afin d'obtenir deux filtres d'analyse à phase linéaire et permettant la reconstruction parfaite.

d) Méthode du calcul d'un filtre complémentaire

Une méthode fort différente de celle de la factorisation consiste à choisir d'abord un premier filtre, $H_0(z)$, puis à calculer l'autre filtre, $H_1(z)$, de telle façon à ce que la relation (3.17) soit respectée. Cette méthode peut être utile dans le cas où $H_0(z)$ est donné et ne peut par conséquent pas être choisi (par exemple si $H_0(z)$ est dérivé du prédicteur linéaire optimal comme proposé pour un dispositif adaptatif de codage en sous-bandes dans [vet86b,sch85]).

Comme seuls les termes d'indices impairs du produit $H_0(z)\cdot H_1(-z)$ apparaissent dans le déterminant, seuls ceux-ci doivent être évalués dans le produit de convolution. Considérons $H_0(z)$ comme donné, et les coefficients de $H_1(-z)$ comme les inconnues à déterminer afin de réduire le déterminant à un monôme. En notation matricielle, nous avons:

$$\begin{bmatrix} h_{0,1} & h_{0,0} & 0 & 0 & \cdots & \cdots \\ h_{0,3} & h_{0,2} & h_{0,1} & h_{0,0} & 0 & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & & & \\ \cdots & \cdots & 0 & h_{0,M-1} & h_{0,M-2} & \end{bmatrix} \cdot \begin{bmatrix} h_{1,0} \\ -h_{1,1} \\ h_{1,2} \\ \vdots \\ (-1)^{M-1} h_{1,M-1} \end{bmatrix} = 1/2 \begin{bmatrix} \Delta_1 \\ \Delta_3 \\ \Delta_5 \\ \vdots \\ \Delta_{2M-3} \end{bmatrix} \tag{3.29}$$

où Δ_i sont les coefficients du déterminant et $h_{j,i}$ ceux du filtre j.

La matrice en (3.29) est une matrice de convolution apériodique, mais où manquent les lignes paires. Notons qu'elle a M colonnes, correspondant aux M inconnues du filtre $H_1(-z)$, mais uniquement M-1 lignes, correspondant aux M-1 coefficients non-nuls du déterminant. Ce degré de liberté permet de choisir un des coefficients $h_{1,i}$. La valeur i doit être telle que la matrice réduite en (3.29) (la matrice sans la colonne i) est si possible non-singulière (un tel choix n'est d'ailleurs pas toujours possible). Pour nos discussions, choisissons $h_{1,0}=1$. De surcroît, nous pouvons également choisir le déterminant voulu, par exemple $\Delta_m(z)=z^{-2k-1}$. Ceci donne lieu au système de M-1 équations à M-1 inconnues suivant:

$$\begin{bmatrix} h_{0,0} & 0 & 0 & \cdots & \\ h_{0,2} & h_{0,1} & h_{0,0} & \cdots & \\ & \ddots & \ddots & \ddots & \\ & & & & \\ \cdots & 0 & h_{0,M-1} & h_{0,M-2} & \end{bmatrix} \cdot \begin{bmatrix} -h_{1,1} \\ h_{1,2} \\ -h_{1,3} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = 1/2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} h_{0,1} \\ h_{0,3} \\ h_{0,5} \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad (3.30)$$

où nous avons éliminé la première inconnue (en posant $h_{1,0}=1$) et passé par conséquent la première ligne de la matrice en (3.29) à droite de l'égalité en (3.30). La résolution de ce système d'équations (si une solution existe, ce qui est en général le cas pour des filtres usuels) donne les coefficients du filtre complémentaire, et on aura bien:

$$\Delta_m(z) = H_0(z) \cdot H_1(-z) - H_0(-z) \cdot H_1(z) = 2 \cdot z^{-2k-1} \qquad (3.31)$$

donc une reconstruction parfaite.

Notons que le choix arbitraire fait ci-dessus peut s'avérer délicat. En particulier, le rapport entre la valeur de $h_{1,i}$ et celle du déterminant est difficile à choisir. Même si les filtres résultants permettent une reconstruction parfaite, le filtre complémentaire peut avoir un comportement exotique. Une échappatoire simple à ce dilemme consiste à imposer une contrainte supplémentaire au filtre complémentaire, donc à poser une équation de plus, ce qui mène alors à M équations pour les M inconnues du filtre à déterminer. Très souvent, on a des informations a priori sur le filtre $H_0(z)$, donc implicitement aussi sur $H_1(z)$. Par exemple, si l'on sait que $H_0(z)$ est un filtre passe-bas, on pourra admettre tacitement que $H_1(z)$ sera un filtre passe-haut, et on peut donc imposer que $z=1$ soit un zéro de $H_1(z)$. Ceci est obtenu en posant comme équation supplémentaire que la somme de $h_{1,i}$ doit être égale à zéro. L'unique problème est que cette équation supplémentaire doit être linéairement indépendante des M-1 équations déjà posées, sinon le système devient insoluble. Dans ce dernier cas (et si les M-1 équations de départ sont bien linéairement indépendantes), il suffit de perturber légèrement l'équation supplémentaire (qui est arbitraire) afin de la rendre indépendante et d'obtenir ainsi un système de dimension M fois M non-singulier.

Un cas intéressant apparaît si l'on impose que $H_0(z)$ et $H_1(-z)$ soient des filtres symétriques de même longueur paire M. Dans ce cas, si $H_0(z)$ est choisi, il faut déterminer les M/2 coefficients de $H_1(-z)$ nécessaires à la définition du filtre complémentaire. En incluant toutes les symétries dans la relation (3.29), il en résulte un système de M/2 équations. Ce système est présenté pour le cas M=8 ci-dessous:

$$[\mathbf{A} + \mathbf{B}] \cdot [h_{1,0} \ldots h_{1,3}]^T = [0\ 0\ 0\ 1]^T \tag{3.32a}$$

$$\mathbf{A} = \begin{bmatrix} h_{0,1} & h_{0,0} & 0 & 0 \\ h_{0,3} & h_{0,2} & h_{0,1} & h_{0,0} \\ h_{0,2} & h_{0,3} & h_{0,3} & h_{0,2} \\ h_{0,0} & h_{0,1} & h_{0,2} & h_{0,3} \end{bmatrix} \tag{3.32b}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & h_{0,0} & h_{0,1} \\ h_{0,0} & h_{0,1} & h_{0,2} & h_{0,3} \end{bmatrix} \tag{3.32c}$$

Si la matrice **A+B** est non-singulière, il suffit de l'inverser afin de trouver le filtre complémentaire. Le système peut être insoluble, comme par exemple dans le cas du filtre moyenneur (tous les coefficients égaux à 1/M). Alors, les matrices en (3.29) et en (3.32) sont singulières donc non-inversibles. Mais ce problème n'apparaît pas en général, comme le montre l'exemple ci-dessous:

Exemple E3.2: En partant d'un filtre passe-bas QMF classique tiré de [joh80] et de longueur 32 (dont la fonction d'amplitude est montrée à la figure 3.2a) nous allons voir quel filtre complémentaire est nécessaire afin d'obtenir une reconstruction parfaite. Comme le filtre de départ est symétrique, nous utilisons la relation (3.32). Le filtre complémentaire est décevant, comme on peut le vérifier dans la figure 3.2b. Même s'il permet, d'un point de vue global, d'obtenir une fonction de transfert idéale, sa caractéristique d'amplitude propre est trop mauvaise pour la plupart des applications.

Cet exemple montre les limitations de cette méthode: un filtre peut être choisi à volonté, la fonction de transfert totale est idéale, mais par contre, le filtre

complémentaire est défini par un système d'équations et n'a pas forcément les caractéristiques désirées.

Toutefois, si le filtre complémentaire n'est pas trop critique, cette méthode a certains avantages. Par exemple, il est possible de réaliser des fonctions de transfert totales ayant un retard minimum. Si dans la relation (3.30) on choisi le déterminant égal à $2 \cdot z^{-1}$ (donc Δ_1=2, et tous les autres Δ_i égaux à zéro), alors la fonction de transfert totale devient, et ceci indépendamment de la longueur des filtres RIF utilisés, égale à (en vertu de (3.3)):

$$\hat{X}(z) = z^{-1} \cdot X(z) \tag{3.33}$$

Ce résultat illustre, pour le cas N=2, la remarqe R2.11 qui établit à N-1 le délai minimum d'un système à N canaux.

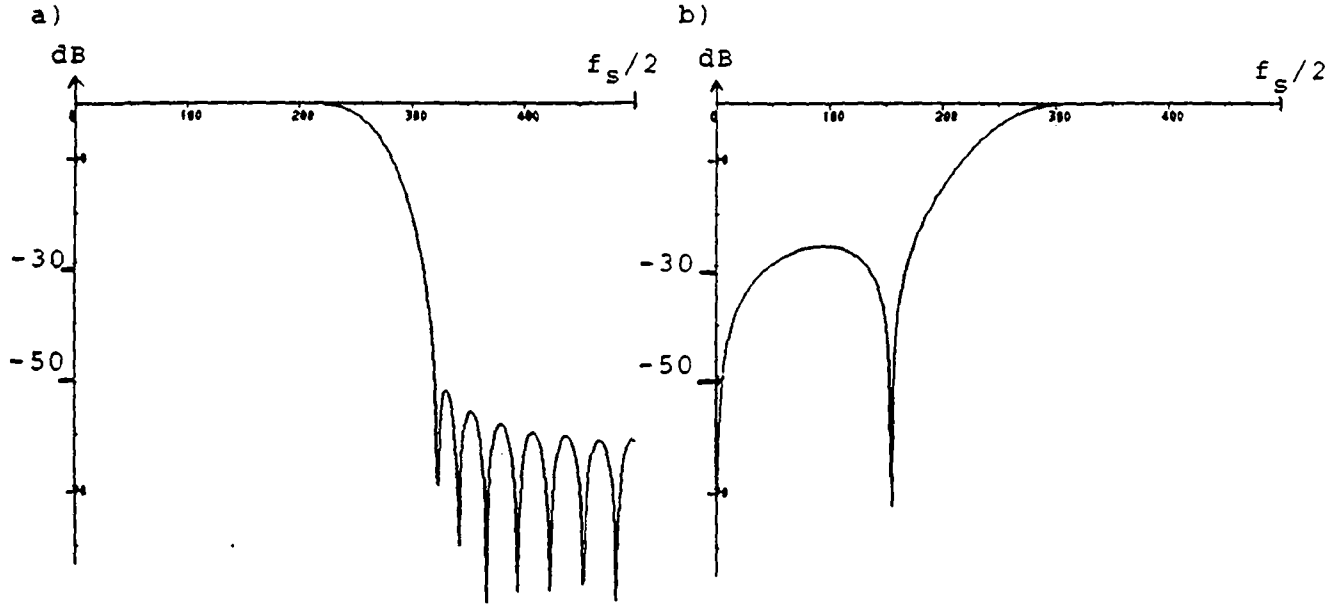

Figure 3.2: Exemple de filtres obtenus avec la méthode du filtre complémentaire (exemple E3.2) pour une longueur 32
a) norme de la fonction de transfert du filtre de départ $H_0(z)$
b) norme de la fonction de transfert du filtre calculé $H_1(z)$

e) Méthode directe

La dernière méthode que nous présentons est peut être la plus simple et la plus souple. Elle consiste à optimiser simultanément $H_0(z)$, $H_1(z)$ et $\Delta_m(z)$ par rapport aux contraintes que l'on impose. Typiquement, on désire un filtre passe-bas $H_0(z)$, un filtre passe-haut $H_1(z)$ et un déterminant équivalent à un filtre passe-tout ayant une phase aussi linéaire que possible (donc idéalement un délai). Dans la figure 3.3, on montre un gabarit idéal par rapport auquel sera calculé une fonction d'erreur. Par la suite, nous allons admettre que $H_0(z)$ et $H_1(z)$ sont des filtres à phase linéaire, donc $\Delta_m(z)$ également. De ce fait, seule la fonction d'amplitude de ces filtres sera considérée.

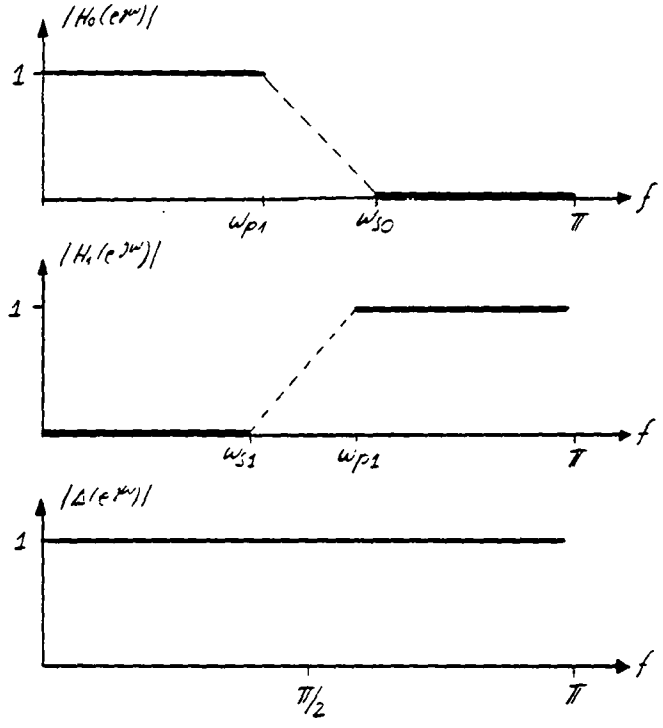

Figure 3.3: Gabarit idéal pour la norme de $H_0(z)$, $H_1(z)$ et $\Delta(z)$

On peut donc évaluer une fonction d'erreur (ou de coût), qui dans sa forme la plus générale peut s'écrire comme:

$$C[H_0(z),H_1(z)] = c_{p0} \cdot {}_0\int^{\omega_{p0}} [1-H_0(e^{j\omega})]^2 \, d\omega + c_{s0} \cdot {}_{\omega_{s0}}\int^{\pi} [H_0(e^{j\omega})]^2 \, d\omega$$
$$+ c_{s1} \cdot {}_0\int^{\omega_{s1}} [H_1(e^{j\omega})]^2 \, d\omega + c_{p1} \cdot {}_{\omega_{p1}}\int^{\pi} [1-H_1(e^{j\omega})]^2 \, d\omega$$
$$+ c_{pt} \cdot {}_0\int^{\pi} [1-\Delta_m(e^{j\omega})]^2 \, d\omega \qquad (3.34)$$

où les indices "p" indiquent les bandes passantes et "s" les bandes "stop", ainsi que "0", "1" les filtres 0 et 1 et "t" la transmission totale. En général, les bandes de transition des deux filtres sont les mêmes, ainsi que la pondération des bandes passantes ou atténuantes. La relation (3.34) se réduit alors à:

$$C[H_0(z),H_1(z)] = c_p \cdot {}_0\int^{\omega_p} ([1-H_0(e^{j\omega})]^2 + [1-H_1(e^{j\omega+\pi})]^2)d\omega$$
$$+ c_s \cdot {}_{\omega_s}\int^{\pi} ([H_0(e^{j\omega})]^2 + [H_1(e^{j\omega+\pi})]^2)d\omega$$
$$+ c_t \cdot {}_0\int^{\pi} [1-\Delta_m(e^{j\omega})]^2 \, d\omega \qquad (3.35)$$

La constante c_p gouverne la qualité de la bande passante et c_s celle de la bande d'atténuation des filtres $H_0(z)$ et $H_1(z)$, tandis que c_t pondère l'importance donnée à la reconstruction du signal à la sortie.

Dans le cas des reconstructions parfaites présentées auparavant, la troisième intégrale dans (3.35) serait évidemment toujours nulle. Dans le cas d'une recherche par optimisation numérique, on pourra tendre vers une reconstruction parfaite en choisissant simplement c_t beaucoup plus grand que c_p et c_s.

Notons également que si $H_1(z)$ est choisi égal à $H_0(-z)$, alors il est suffisant de considérer l'atténuation dans la bande $[\omega_s,\pi]$ pour $H_0(e^{j\omega})$ ainsi que la qualité de la reconstruction totale. Ceci est suffisant, car la qualité de la bande passante apparaît implicitement dans la reconstruction totale (les filtres étant à phase linéaire). Ce cas a été considéré dans la méthode de conception de [joh80] où l'optimisation des filtres a été effectuée en conséquence. Dans notre cas, les différents coûts ne sont d'ailleurs pas non plus entièrement indépendants les uns des autres. Ci-dessous, un exemple de conception basé sur la relation (3.35) est donné.

Exemple E3.3: *Deux filtres à phase linéaire $H_0(z)$ et $H_1(z)$, non liés par une relation de modulation ($H_0(z) \neq H_1(-z)$ a priori), sont conçus en optimisant la fonction coût donnée par la relation (3.35). On a choisi $M=8$, $c_p=1$, $c_s=10$ et $c_t=1000$. A l'aide d'une optimisation stochastique, on a obtenu les filtres et le déterminant montrés dans la figure 3.4. Pour comparaison, on y voit également le filtre de longueur 8 proposé dans [joh80], qui a d'ailleurs été pris comme filtre initial pour l'optimisation. Les filtres résultats sont légèrement moins bons, mais la fonction de transfert totale a pu être améliorée (le minimum est ramené de 0.122 dB à 0.079 db).*

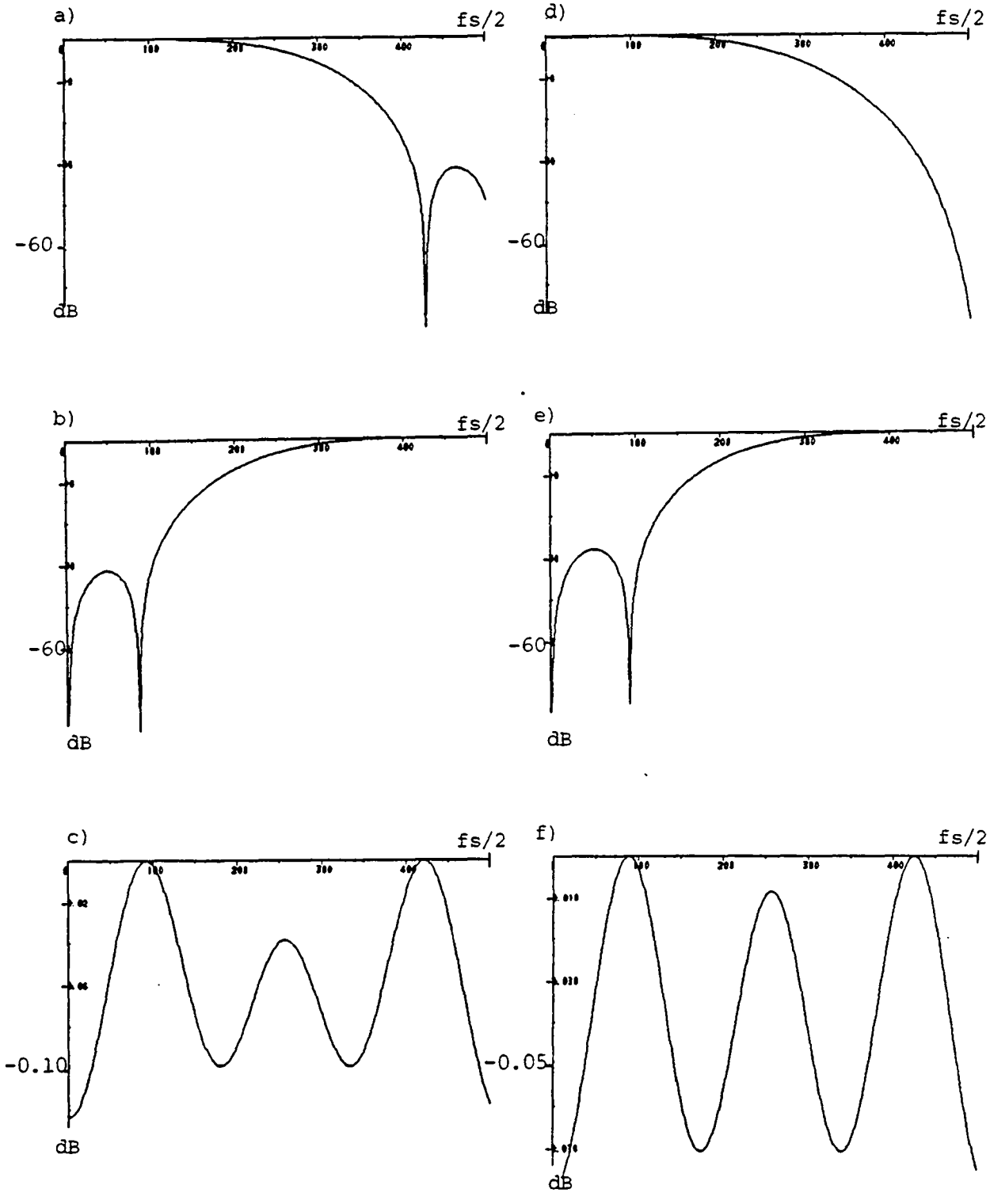

Figure 3.4: Résultat de l'optimisation directe (exemple E3.3)

a) norme du passe-bas initial de longueur 8 de [joh80]

b) norme du passe-haut initial obtenu par modulation

c) norme du déterminant initial

d) norme du passe-bas après optimisation

e) norme du passe-haut après optimisation

f) norme du déterminant après optimisation

Une possibilité intéressante de la méthode directe est que l'on peut chercher des filtres produisant un délai inférieur à celui obtenu avec des filtres à phase linéaire. Le délai entrée-sortie devient alors un des paramètres de l'optimisation, comme le montre l'exemple ci-dessous:

Exemple E3.4: Un filtre passe-bas $H_0(z)$ de longueur M=12 (le filtre passe-haut sera choisi égal à $H_0(-z)$) est optimisé de façon à ce que le codeur QMF correspondant n'ait qu'un délai de 7 échantillons (et non pas de 11 comme cela serait le cas avec des filtres à phase linéaire). Le résultat est donc comparé avec le filtre de longueur 8 de [joh80], puisque celui-ci produit le même délai. Les parties a), b) et c) de la figure 3.5 montrent les résultats obtenus avec le filtre de [joh80], et les parties d), e) et f) proviennent de l'optimisation. Le filtre passe-bas est légèrement détérioré, mais la fonction de transfert totale est par contre nettement meilleure (l'atténuation maximum passe de 0.122 dB à seulement 0.022 dB).

Cet exemple conclut la discussion de la méthode directe. Celle-ci paraît moins élégante que les autres méthodes proposées plus haut, puisqu'en lieu et place de permettre une solution analytique, elle nécessite une optimisation numérique. Néanmoins, elle ne présente pas les désavantages intrinsèques aux méthodes discutées auparavant, et permet l'obtention de filtres de qualité au prix d'un certain effort d'optimisation.

Notons qu'il existe également une méthode récursive pour obtenir des filtres RIF permettant une reconstruction parfaite. Prenons deux matrices de filtres $\mathbf{H}_{p1}(z)$ et $\mathbf{H}_{p2}(z)$ dont les déterminants sont des monômes. Ces matrices de filtres peuvent s'écrire, selon (2.36), comme le produit de matrices de délais $\mathbf{I}_d(z)$ et de matrices de composantes polyphases $\mathbf{N}_{p1}(z^N)$ et $\mathbf{N}_{p2}(z^N)$. Evidemment, les déterminants des matrices de composantes polyphases sont également des monômes. Une nouvelle matrice de filtres ayant également un déterminant sous forme de monôme peut être dérivée de la façon suivante:

$$\mathbf{H}_{p1}(z) = \mathbf{I}_d(z) \cdot \mathbf{N}_{p1}(z^N) \cdot \mathbf{N}_{p2}(z^N) \qquad (3.36)$$

Cette méthode n'a pas été explorée en détail, car il n'est pas clair comment les propriétés des facteurs se répercutent sur la matrice produit. Elle pourrait néanmoins être utile pour les bancs de filtres sur des corps finis [vet86b], puisqu'alors les propriétés des filtres eux-mêmes sont peu importantes, le problème étant plutôt de générer ceux-ci.

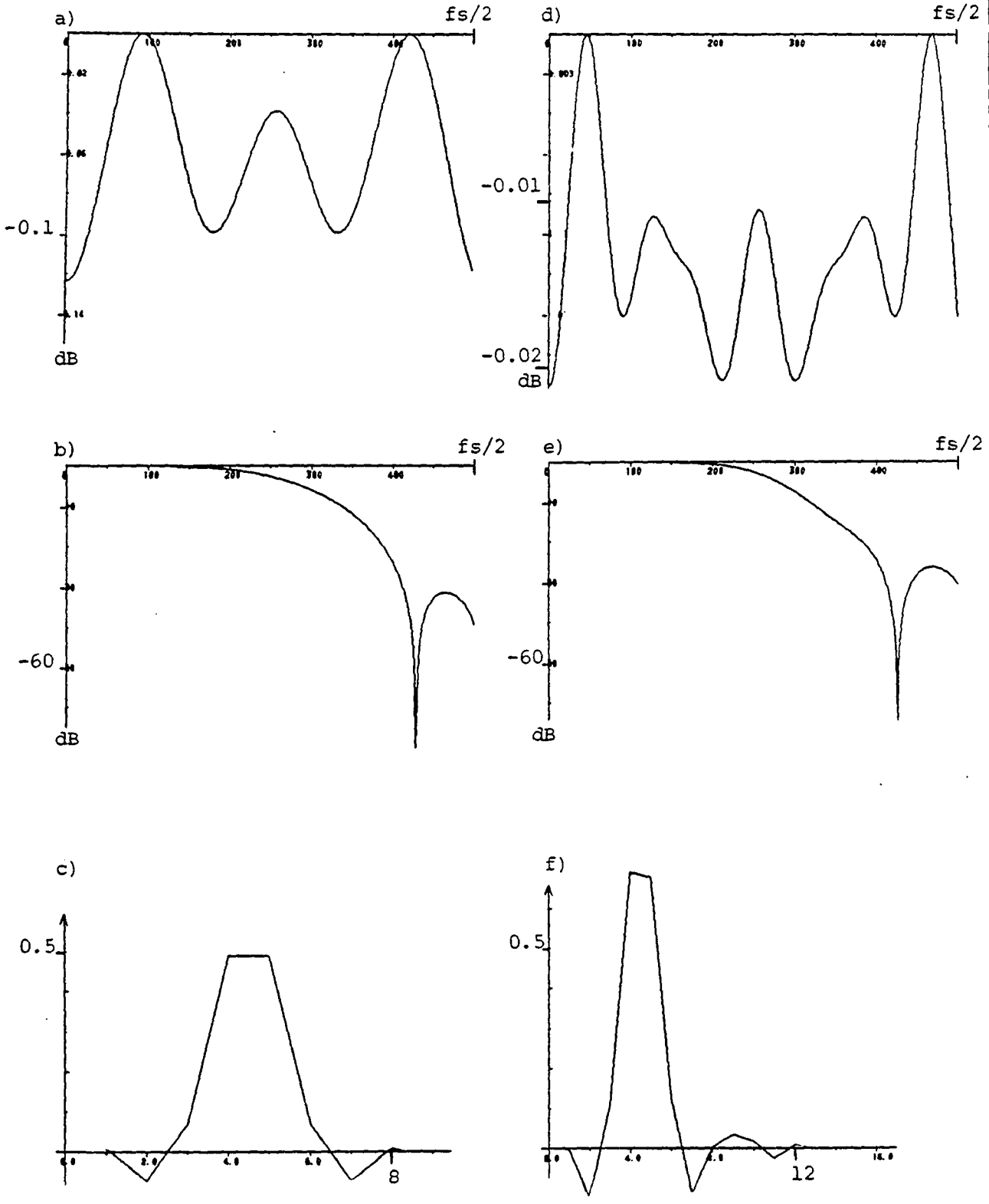

Figure 3.5: Résultat de l'optimisation directe (exemple E3.4)

a) norme du déterminant initial

b) norme du passe-bas initial de longueur 8 de [joh80]

c) réponse impulsionnelle du passe-bas initial

d) norme du déterminant optimisé

e) norme du passe-bas optimisé (longueur 12)

f) réponse impulsionnelle du passe-bas optimisé (qui n'est plus à phase linéaire)

Passons en revue les résultats principaux de cette section sur les bancs de deux filtres RIF. D'abord, les avantages de bancs de filtres RIF ont été montrés (entre autres, l'absence d'annulation implicite de pôles par des zéros), puis la condition nécessaire pour une reconstruction parfaite a été dérivée (déterminant égal à un monôme). Après une revue de deux méthodes connues (les QMF classiques et la solution à phase minimale/maximale), on a introduit trois méthodes de conception générales: la méthode de la factorisation (séparation d'un polynôme, équivalent au déterminant, en deux polynômes correspondant aux filtres d'analyse), la méthode du filtre complémentaire (calcul d'un filtre complémentaire afin d'obtenir un déterminant égal à un monôme) et enfin la méthode directe (optimisation simultanée de $H_0(z)$, de $H_1(z)$ et du déterminant correspondant). Les résultats sont résumés dans la table 3.1 qui donne un aperçu des différentes méthodes, de leur domaines d'application ainsi que de leurs avantages et inconvénients respectifs.

3.1.2 Filtres à réponse impulsionnelle infinie

L'avantage majeur des filtres à réponse impulsionnelle infinie (RII) est une complexité de calcul réduite par rapport à des filtres RIF ayant des caractéristiques fréquentielles similaires. Les désavantages sont leur comportement numérique, le risque d'instabilité ainsi que leur comportement de phase non-linéaire. Dans le cadre des bancs de filtres avec reconstruction, ils ont un désavantage supplémentaire qui est de donner lieu à des annulations implicites de pôles par des zéros (en tous cas si une reconstruction parfaite est désirée, voir à ce propos la remarque 3.1).

Prenons un banc d'analyse comportant deux filtres stables RII, $H_0(z)=N_0(z)/D_0(z)$ et $H_1(z)=N_1(z)/D_1(z)$. Les trois remarques suivantes définissent les différentes conditions de reconstruction.

Remarque R3.5: Une reconstruction sans repliement spectral est toujours possible, puisque la matrice des cofacteurs correspond à des filtres stables en vertu de la remarque R2.9.

Cette remarque est d'ailleurs valable pour N arbitraire. Lorsque les filtres de synthèse sont choisis conformément à (3.2), le signal reconstruit devient égal à:

$$\hat{X}(z) = 1/2 \cdot \Delta_m(z) \cdot X(z) \qquad (3.37a)$$

avec:

Table 3.1: Conception de filtres RIF pour le cas N=2. Le repliement spectral est toujours annulé

Methode	$H_0(z)$	$H_1(z)$	long.M	phase	determinant	recon. parfai	Optimisation	Remarques
Directe								
QMF classique	$H(z)$	$H(-z)$	paire impaire	lin.	$H^2(z)-H^2(-z)$	M=2	'H(z)', ' (z)'	-demande un filtre correcteur dans le cas M impaire
Générale	$H_0(z)$	$H_1(z)$	paire/ impaire	lin./ quelc.	$H_0(z)H_1(-z)$ $-H_0(-z)H_1(z)$	possib.	'$H_0(z)$', '$H_1(z)$ et ' (z)' et	-souplesse: chaque caractéristique peut être pondérée (bande passante, atténuante et reconstruction)
Factorisation								
phase min/max	$H(z)$	$z^{-M+1}H(z^{-1})$	paire	min/max	z^{-M+1}	oui	autocorrélation H(z)	-factorisation d'un polynôme sym. avec double zéros sur le cerc. un.
Générale	$H_0(z)$	$H_1(z)$	quelc.	quelc.	z^{-2k-1}	oui	filtre produit et	-difficulté d'optimiser le filtre
			paire	lin.	z^{-M+1}	oui	répartition des zéros	produit en fonction des facteurs
			quelc.	min.	z^{-1}	oui	filt. prod. donnant un délai minimal	-nombre expon. de solutions -délai minimal entrée/sortie (z^{-1})
Filtre complémentaire								
	$H_0(z)$	$H_1(z)$	quelc.	quelc.	z^{-2k-1}	oui	$H_0(z)$ est arbitraire	-si $H_0(z)$ peut être arbit. bon, on
			paire	lin.	z^{-M+1}	oui	$H_1(z)$ est obtenu par	n'a que peu d'influence sur la
			quelc.	min.	z^{-1}	oui	résol. d'un syst.d'eq.	qualité de $H_1(z)$
Méthode récursive	$H_0'(z)$& $H_0''(z)$	$H_1'(z)$& $H_1''(z)$	quelc.	quelc.	$z^{-2k_1-1}\cdot z^{-2k_2-1}$	oui	optimisation des fact. mais pas du produit	applicable à la génération de filt. sur des corps finis.

$$\Delta_m(z) = \frac{N_0(z)\cdot N_1(-z)\cdot D_0(-z)\cdot D_1(z) - N_0(-z)\cdot N_1(z)\cdot D_0(z)\cdot D_1(-z)}{D_0(z)\cdot D_0(-z)\cdot D_1(z)\cdot D_1(-z)} \qquad (3.37b)$$

qui est une fonction impaire de z^{-1} (le numérateur n'a que des puissances impaires de z^{-1}, et le dénominateur que des puissances paires de z^{-1}). Si $H_0(z)=N(z)/D(z)$ et $H_1(z)=H_0(-z)$, la relation (3.37) devient:

$$\Delta_m(z) = \frac{N^2(z)\cdot D^2(-z) - N^2(-z)\cdot D^2(z)}{D^2(z)\cdot D^2(-z)} \qquad (3.38)$$

Ces types de filtres ont été proposés dans [mil85]. Si les dénominateurs des filtres sont des fonctions de z^{-2} plutôt que des polynômes généraux, (3.37) devient:

$$\Delta_m(z) = \frac{N_0(z)\cdot N_1(-z) - N_0(-z)\cdot N_1(z)}{D_0(z^2)\cdot D_1(z^2)} \qquad (3.39)$$

De tels filtres sont utilisés dans [ram80]. Finalement, si les deux propriétés ci-dessus sont réunies, c'est-à-dire que $H_0(z)=N(z)/D(z^2)$ et que $H_1(z)=H_0(-z)$, il en résulte que:

$$\Delta_m(z) = \frac{N^2(z) - N^2(-z)}{D^2(z)} \qquad (3.40)$$

Bien que les dénominateurs en z^{-2} aient été proposés, ils ne sont à notre avis que de peu d'utilité (nous parlons d'un dénominateur en z^{-2} lorsque le dénominateur d'un filtre, après toutes les simplifications possibles avec le numérateur, reste une fonction en z^{-2}). La raison en est qu'alors, si p_i est un pôle du filtre, $-p_i$ est également un pôle du filtre, et cette paire de pôles ne contribue donc pas à séparer un signal en composantes haute et basse fréquence. De ce fait, les cas les plus intéressants sont le cas général (3.37) et le cas QMF-RII (3.38). La seconde remarque concerne la reconstruction passe-tout:

Remarque R3.6: Une reconstruction passe-tout est possible si et seulement si le déterminant n'a pas de zéros sur le cercle unité.

Il est évident que cette condition est suffisante, car après avoir choisi les filtres de synthèse selon (3.2), on place à la sortie du système un filtre de compensation ayant les propriétés suivantes:

- un zéro en $z=q_i$ pour chaque pôle en $z=q_i$ du déterminant
- un pôle en $z=p_i$ pour chaque zéro du déterminant qui se trouve à l'intérieur du

cercle unité en $z=p_i$

- un pôle en $z=[p_i^*]^{-1}$ pour chaque zéro du déterminant qui se trouve à l'extérieur du cercle unité en $z=p_i$.

Ainsi, les filtres de synthèse sont stables, et la reconstruction est passe-tout. La condition est également nécessaire, puisqu'un zéro du déterminant sur le cercle unité ne peut être annulé de façon stable (voir également la condition c2.2.

La dernière remarque concerne la reconstruction parfaite:

Remarque R3.7: Une reconstruction parfaite n'est possible que si tous les zéros du déterminant se trouvent strictement à l'intérieur du cercle unité.

Dans ce cas, il suffit de placer un filtre de compensation égal à $1/\Delta_m(z)$ à la sortie. La reconstruction est alors stable (tous les zéros du déterminant sont à l'intérieur du cercle unité) et parfaite. Par ailleurs, la nécessité a été prouvée dans la démonstration qui précède la condition C2.2, dans la section 2.3.1c).

Dans le cas général donné par (3.37), il semble difficile de dériver une méthode de synthèse analytique comme il a été possible de le faire dans le cas de filtres RIF. Bien sûr, on peut choisir deux filtres d'analyse puis vérifier que le déterminant satisfait bien aux propriétés requises pour une reconstruction parfaite ou passe-tout, mais une telle méthode par essais successifs n'est pas très satisfaisante.

Un cas plus simple est celui où les filtres sont modulés et où le déterminant est donc donné par la relation (3.38). Notons d'abord que la matrice polyphase $\mathbf{H}_p(z)$ correspondant à un tel système s'écrit de la façon suivante (selon (2.36)):

$$\mathbf{H}_p(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \cdot \begin{bmatrix} P_0(z^2) & P_0(z^2) \\ P_1(z^2) & -P_1(z^2) \end{bmatrix} \quad 1/(D(z)\cdot D(-z)) \tag{3.41a}$$

où:

$$P_0(z^2) = N(z)\cdot D(-z) + N(-z)\cdot D(z) \tag{3.41b}$$

$$P_1(z^2) = z\cdot [N(z)\cdot D(-z) - N(-z)\cdot D(z)] \tag{3.41c}$$

Evidemment, la matrice $\mathbf{H}_p(z)$ est diagonalisable par une post-multiplication avec la

matrice de Fourier, et il est aisé de vérifier que le déterminant de $\Delta_m(z)$ est égal à:

$$\Delta_m(z) = \frac{z^{-1} \cdot P_0(z^2) \cdot P_1(z^2)}{[D(z) \cdot D(-z)]^2} \qquad (3.42)$$

L'équivalence avec (3.38) est immédiate. Il est dès lors suffisant de considérer les zéros de $P_0(z)$ et $P_1(z)$ ($P_0(z^2)$ et $P_1(z^2)$ ont les mêmes zéros que $P_0(z)$ et $P_1(z)$, ainsi ceux-ci multipliés par -1). Ceci réduit de moitié le degré des polynômes dont il faut chercher les zéros. Si aucun de ces zéros ne se trouve sur le cercle unité, alors on peut réaliser une reconstruction passe-tout selon la remarque R3.6. Si, de surcroît, ils se trouvent tous à l'intérieur du cercle unité, une reconstruction parfaite est possible (remarque R3.7). Ce résultat est lié à la condition C2.3, qui, rappelons-le, établit que l'inverse d'une matrice de filtres modulés (ce qui est le cas ici) n'est stable que si les zéros des composantes polyphases sont strictement à l'intérieur du cercle unité.

Si un banc d'analyse est composé de filtres RIF, mais que l'on tolère des filtres RII pour la synthèse, on peut améliorer la qualité de la reconstruction. Toutefois, dans le cas de filtres à phase linéaire, la remarque suivante indique qu'une reconstruction parfaite ne peut être obtenue qu'avec des filtres RIF donnant lieu à un déterminant qui est un monôme.

Remarque R3.8: Dans le cas d'un banc d'analyse constitué de filtres à phase linéaire et ayant tous deux une longueur soit paire soit impaire, une condition nécessaire et suffisante pour la reconstruction parfaite est que le déterminant soit un monôme.

La preuve se base sur l'annexe A3.2, où l'on montre que le déterminant a toujours une symétrie (relation (a3.11) et (a3.14)) lorsque les filtres d'analyse sont à phase linéaire. A part dans le cas où $\Delta_m(z)$ est un monôme, le déterminant aura toujours des zéros ailleurs que strictement à l'intérieur du cercle unité, et une reconstruction parfaite devient impossible en vertu de la remarque R3.7. Notons que si les filtres ne sont pas de même longueur, mais sont tous deux pairs ou impairs, la remarque R3.8 est également valable.

Si les filtres ne sont pas de même longueur, mais l'un de longueur paire et l'autre de longueur impaire, le déterminant n'est plus symétrique et la remarque R3.8 ne s'applique pas. Ceci vient du fait que $H_0(z) \cdot H_1(-z)$ est alors un filtre de longueur paire avec symétrie, mais que $H_0(-z) \cdot H_1(z)$ possède donc la symétrie complémentaire (en vertu de (a3.7)). De ce fait, le dernier terme de (a3.8) s'annule, et le déterminant n'est plus symétrique.

Notons toutefois qu'il est rare d'avoir deux filtres à phase linéaire mais de longueurs différentes (leurs sorties ne seront d'ailleurs pas en phase, mais décalées d'un demi délai). Finalement, la remarque R3.8 démontre également que les QMF classiques ne permettent jamais une reconstruction parfaite, puisque dans ce cas, le déterminant ne peut être un monôme. Ce résultat avait également été montré, mais de façon différente, dans [bar82].

Pour conclure cette section sur les filtres RII dans le cadre des systèmes à deux canaux, rappelons brièvement les résultats les plus importants. D'abord, tout comme dans le cas RIF, la reconstruction sans repliement spectral ne présente aucun problème. En choisissant les filtres de synthèse en fonction de la matrice des cofacteurs (voir (3.2)), on est assuré d'avoir des filtres de synthèse stables et d'annuler parfaitement le repliement spectral. La fonction de transfert totale est alors donnée par le déterminant. Soit ce dernier se rapproche d'un délai idéal, soit il peut être compensé voire annulé par un filtre de sortie. Ce dernier cas est parfaitement possible si tous les zéros du déterminant sont à l'intérieur du cercle unité. Si des zéros se trouvent également à l'extérieur (mais pas sur le cercle unité), on peut obtenir une reconstruction passe-tout. Par contre, si le déterminant possède des zéros sur le cercle unité, il est impossible de reconstruire le signal original, même en acceptant une distorsion de phase. Finalement, nous avons montré que la reconstruction parfaite dans le cas d'une analyse avec des filtres à phase linéaire exige un déterminant égal à un monôme, excluant donc par là les QMF classiques.

3.1.3 Application aux filtres de synthèse avec reconstruction

La différence fondamentale entre les systèmes d'analyse avec reconstruction (du genre codeurs en sous-bandes) et les systèmes de synthèse avec reconstruction (du type transmultiplexeurs TDM-FDM) est que dans le premier cas les opérations de sous- puis de suréchantillonnage sont automatiquement en phase, alors que dans le second cas il faut expressément chercher une mise en phase. D'un point de vue purement formel, cette différence avait été explorée dans la section 2.3.4. Nous allons simplement illustrer ces résultats dans le cadre des bancs de deux filtres, qui est plus simple et plus parlant que le cas général. Prenons deux filtres RIF $H_0(z)$ et $H_1(z)$ pour le banc de synthèse qui se trouve à l'entrée du système (voir la figure 2.4). Il faut alors choisir les filtres du banc d'analyse se trouvant à la sortie du système comme (selon le théorème T2.2):

$$G_0(z) = z^{-1} \cdot H_1(-z) \tag{3.43a}$$

$$G_0(z) = -z^{-1} \cdot H_0(-z) \tag{3.43b}$$

Notons bien le délai de z^{-1}, conformément au théorème T2.2. Si ce délai n'est pas introduit, il faudrait décaler le sous-échantillonnage en conséquence (d'un délai correspondant à la fréquence d'échantillonnage du canal). En fait, la définition de l'instant d'échantillonnage est arbitraire (ici, nous admettons qu'un sous-échantillonnage d'un facteur 2 garde tous les échantillons ayant un indice pair), mais l'essentiel est que la définition choisie soit appliquée de façon cohérente. Voyons brièvement l'effet obtenu si la simultanéité n'est pas respectée. Selon la figure 2.4 et les relations (2.11-14) et (3.43), on obtient par exemple:

$$X'_0(z) = z^{-1} \cdot H_1(-z) \cdot [\, H_0(z) \cdot X_0(z^2) + H_1(z) \cdot X_1(z^2) \,] \tag{3.44}$$

Si l'on échantillonne en phase, on trouve:

$$\begin{aligned} \hat{X}_0(z) &= 1/2 \cdot [\, X'_0(z^{1/2}) + X'_0(-z^{1/2}) \,] \\ &= 1/2 \,[\, z^{-1} \cdot H_1(-z) \cdot H_0(z) - z^{-1} \cdot H_1(z) \cdot H_0(-z) \,] \cdot X_0(z) \\ &\quad + 1/2 \,[\, z^{-1} \cdot H_1(-z) \cdot H_1(z) - z^{-1} \cdot H_1(z) \cdot H_1(-z) \,] \cdot X_1(z) \\ &= 1/2 \cdot z^{-1} \cdot [\, H_1(-z) \cdot H_0(z) - H_1(z) \cdot H_0(-z) \,] \cdot X_0(z) \end{aligned} \tag{3.45}$$

Il y donc bien annulation parfaite de la diaphonie. Par contre, si l'on garde les échantillons d'indices impairs, il résulte que:

$$\begin{aligned} \hat{X}_0(z) &= 1/2 \cdot [\, X'_0(z^{1/2}) - X'_0(-z^{1/2}) \,] \\ &= 1/2 \,[\, z^{-1} \cdot H_1(-z) \cdot H_0(z) + z^{-1} \cdot H_1(z) \cdot H_0(-z) \,] \cdot X_0(z) \\ &\quad + z^{-1} \cdot H_1(-z) \cdot H_1(z) \cdot X_1(z) \end{aligned} \tag{3.46}$$

La diaphonie n'est donc pas nulle. Notons que si $H_1(z)$ est un filtre de demi-bande parfait, le produit $H_1(z) \cdot H_0(-z)$ est nul, et la diaphonie disparaît comme prévu. Néanmoins, la relation (3.46) indique bien la sensibilité du système à la perte de phase, et nous reviendrons sur ce problème par la suite. Par ailleurs, une bonne

conception de filtres pour de tels systèmes devrait tenir compte de (3.45) et de (3.46), voire même d'un échantillonnage plus fin, et ceci afin de diminuer la sensibilité à la perte de phase.

3.1.4 Quelques remarques sur le cas de canaux non-idéaux

Jusqu'à présent, et ceci tant dans le cas des bancs d'analyse que dans celui des bancs de synthèse avec reconstruction, nous avons admis que les canaux de transmission étaient idéaux. Bien que cette abstraction soit utile pour l'analyse, elle n'est évidemment pas conforme à la réalité. D'une part, les bancs d'analyse avec reconstruction sont utilisés dans les codeurs en sous-bandes où les signaux dans les canaux sont soumis à un traitement non-linéaire (quantification grossière). D'autre part, les bancs de synthèse avec reconstruction servent au transmultiplexage, une application où ni le canal ni la récupération de phase ne sont parfaits.

Quoique le sujet des canaux non-idéaux soit fort important, il est trop vaste pour en permettre un traitement détaillé dans le cadre de ce travail. Dans cette section, nous n'avons aucunement la prétention de résoudre ce problème, au contraire, nous essayerons d'en éclairer certaines parties afin de démontrer l'importance du sujet. Pour ce faire, nous utiliserons quelques exemples, accompagnés de simulations et d'un traitement mathématique relativement sommaire.

Trois cas sont traités ci-dessous. D'abord, une analyse préliminaire de l'effet de quantification dans les canaux de codeurs en sous-bandes est donnée, avec comme résultat une approximation du niveau minimum de repliement spectral apparaissant dans la sortie. Ensuite, la perte de phase dans le cas des transmultiplexeurs est illustrée par simulation. Finalement, l'effet d'un canal non-idéal est étudié. Notons que l'effet de la perte de phase et celui du canal non-idéal ne sont pas orthogonaux, mais qu'ils sont traités de façon séparée pour plus de simplicité.

a) Effet de la quantification dans le codage en sous-bandes

Nous allons utiliser un modèle très simplifié afin de montrer que le repliement spectral n'est plus parfaitement supprimé dès que les canaux sont quantifiés (cas N=2). Supposons que le canal 0 soit parfait (aucune quantification), mais que le canal 1 utilise une quantification avec un pas égal à q. Admettons que le signal d'entrée soit une sinusoïde de fréquence f_0 ($<\pi/2$). Le repliement spectral produit également une sinusoïde modulée à la fréquence $f_1=\pi-f_0$. Si l'atténuation du filtre $H_1(z)$ à la fréquence f_0 est telle que le signal a une amplitude qui est de

l'ordre du pas de quantification, le signal disparaîtra. Comme il contient la sinusoïde modulée qui est nécessaire à la suppression cohérente du repliement spectral à la sortie, il en résulte un repliement spectral non-nul dont l'ordre de grandeur est de celui du pas de quantification:

$$\hat{X}(z) = 1/2 \; [\; H_1(-z)\, H_0(z)\, X(z) + H_1(-z)\, H_0(-z)\, X(-z) \;] \qquad (3.47)$$

Dans (3.47), on a simplement admis que le canal 1 dans la relation (3.1) était annulé en raison de la quantification. Notons que si les filtres $H_0(z)$ et $H_1(z)$ sont très raides, la fréquence f_0 (qui correspond à une atténuation proche du pas de quantification) sera proche de π, donc la sinusoïde modulée également, et l'effet sera négligeable.

Bien sûr, tout signal qui disparaît dans un des canaux (en raison de la quantification) va apparaître en version repliée dans la sortie, et ceci selon (3.47). Nous ne pousserons pas plus loin dans cette direction, car comme la quantification est un phénomène non-linéaire, son analyse devient rapidement complexe. Nous notons simplement qu'un repliement spectral de l'ordre de la quantification ne peut être évité.

b) Perte de phase dans les transmultiplexeurs

A la section 3.1.3, il a été montré que si le sous-échantillonnage à la réception était effectué en gardant les échantillons d'indices impairs (et non pas pair), il en résultait de la diaphonie (alors que celle-ci était nulle dans l'autre cas). Il est donc nécessaire d'effectuer le sous-échantillonnage au récepteur en phase avec le suréchantillonnage de l'émetteur. Comme ceci n'est pas toujours possible, nous avons simulé l'effet de la perte de phase. Pour ce faire, un système à deux canaux et utilisant des filtres demi-bandes de longueur 31 en configuration QMF habituelle (voir (3.6)) a été implanté. Les filtres étant de longueur impaire, il a fallu mettre un retard à l'entrée et à la sortie du canal 0 et 1 respectivement. Un canal de transmission analogique a été simulé en utilisant une fréquence d'échantillonnage 100 fois plus élevée (avec un filtre passe-bas adéquat). L'effet de la perte de phase est visualisé dans la figure 3.6, où on voit le rapport (énergie de diaphonie)/(énergie principale) pour une erreur de phase allant de 0 à une période d'échantillonnage du signal de sortie. Pour une erreur de phase de 0 ainsi que d'une période, la diaphonie est bien nulle, alors qu'elle est maximale pour une erreur de 0.5 (dans le cas où tous les filtres sont à phase linéaire).

La figure 3.7 montre les réponses impulsionnelles de l'entrée 0 aux sorties 0 et 1

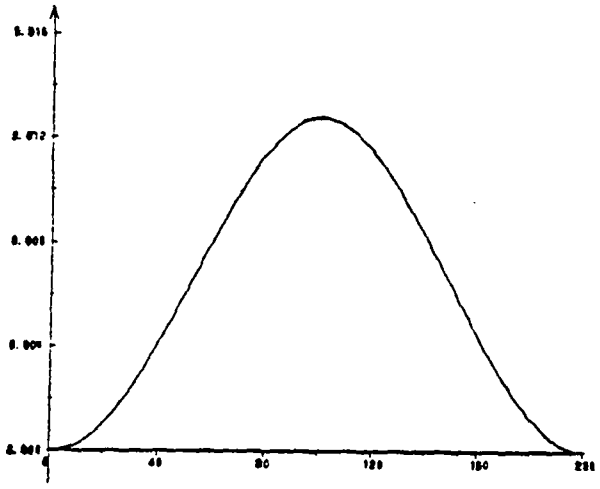

Figure 3.6: Rapport (énergie de diaphonie)/(énergie principale) pour une erreur de phase allant de 0 à une période d'échantillonnage du signal de sortie (le niveau de la diaphonie se situe vers -18.93 dB dans le cas le plus défavorable).

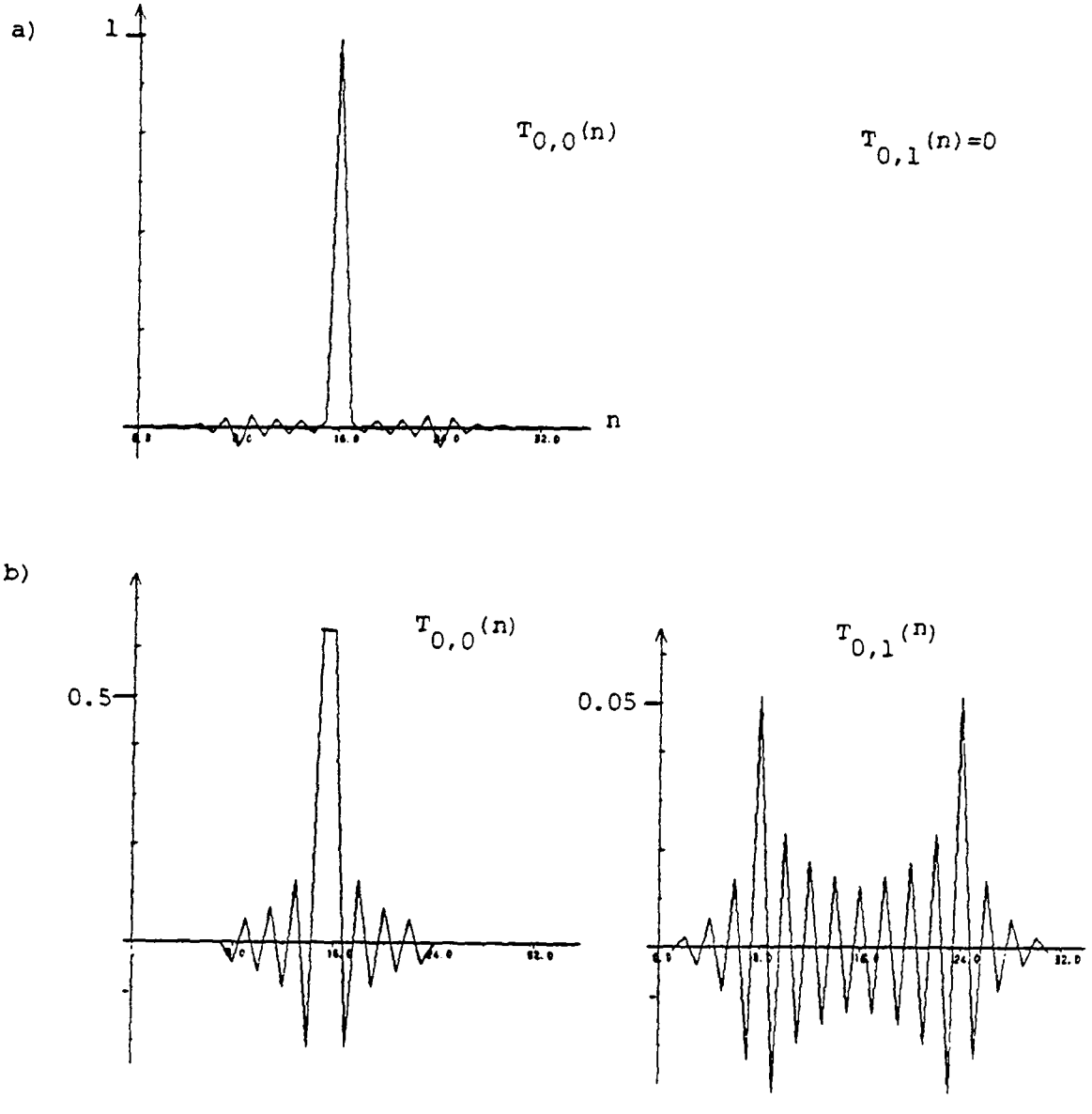

Figure 3.7: Réponse impulsionnelle de l'entrée 0 aux sorties 0 et 1
a) erreur de phase = 0 (il n'y a pas de diaphonie)
b) erreur de phase = 0.5 période d'échantillonnage

dans le cas où la phase est parfaitement récupérée (partie a, la réponse de 0 à 1 est nulle puisque la diaphonie est parfaitement supprimée) et dans le cas le plus défavorable (partie b, où l'erreur de phase est égale à une demi-période d'échantillonnage du signal de sortie).

Il ressort des figures 3.6 et 3.7 que le système est sensible à la perte de phase, mais que la diaphonie apparaît progressivement avec l'augmentation de l'erreur de phase. Notons que si le canal de transmission possède des distorsions de phase, il est relativement aisé de les égaliser (voir par exemple [cox86]).

c) Canaux non-idéaux dans les transmultiplexeurs

Un canal de transmission ne sera évidemment jamais idéal, même après égalisation. Il est possible de vérifier que si le canal a une fonction de transfert donnée par A(z), alors le vecteur des signaux de sortie $\hat{\mathbf{x}}(z^N)$ peut être écrit, en fonction du vecteur des signaux d'entrée $\mathbf{x}(z^N)$ d'un transmultiplexeur, comme [vet86c]:

$$\hat{\mathbf{x}}(z^N) = 1/N\ [\mathbf{G}_m(z)]^T \cdot \mathbf{A}(z) \cdot \mathbf{H}_m(z) \cdot \mathbf{x}(z^N) \tag{3.48}$$

avec:

$$\mathbf{A}(z) = \mathrm{diag}[A(z)\ A(Wz)\ .\ .\ A(W^{N-1}z)] \tag{3.49}$$

Prenons à nouveau le cas simple de N=2, et le choix habituel des filtres QMF (selon (3.6)). La matrice de transmission (voir (2.66)) est alors égale à:

$$\mathbf{T}(z) = 1/2 \begin{bmatrix} A(-z)H^2(-z) - A(z)H^2(z) & [A(-z) - A(z)]H(z)H(-z) \\ [A(z) - A(-z)]H(z)H(-z) & A(-z)H^2(-z) - A(z)H^2(z) \end{bmatrix} \tag{3.50}$$

Celle-ci n'est pas diagonale, à moins que A(z) ne soit une fonction de z^2 (peu probable pour un canal réel) ou que le filtre H(z) soit un filtre demi-bande idéal (auquel cas $H(z) \cdot H(-z) = 0$). Notons toutefois que souvent un canal réel est quasi idéal sur des bandes adjacentes, et que ceci est suffisant pour permettre la suppression de la diaphonie principale (voir à ce propos l'approche pseudo-QMF discutée à la section 3.3.2).

En conclusion à cette section, notons que 3 problèmes majeurs peuvent détruire les propriétés désirables des bancs de filtres étudiés auparavant. La quantification des canaux dans les codeurs en sous-bandes fait apparaître des repliements spectraux, et la perte de phase ou les non-idéalités du média de transmission dans les transmultiplexeurs créent de la diaphonie.

3.2 Bancs de N filtres généraux

Cette section explore le cas de filtres généraux pour des bancs de dimension N plus grande que 2. A la suite du cas RIF, où on montre comment obtenir une reconstruction parfaite avec des filtres d'analyse et de synthèse RIF, on aborde brièvement le cas RII.

Notons tout de suite que le cas de N supérieur à 2 est nettement plus difficile à traiter, et que certaines solutions analytiques que nous avons pu dériver à la section 3.1 n'ont pas encore d'équivalents pour N plus grand que 2. Remarquons que le cas particulier de bancs obtenus par cascade de bancs de dimension 2 et ayant par conséquent une dimension égale à une puissance de 2 n'est pas considéré ici, puisqu'il s'agit d'une généralisation évidente des bancs de deux filtres.

3.2.1 Filtres à réponse impulsionnelle finie

Nous considérons ici des bancs d'analyse et de synthèse constitués de filtres RIF uniquement. Rappelons que la remarque R3.1 et la condition C3.1 (établissant la reconstruction parfaite RIF) ainsi que la remarque R3.2 (selon laquelle l'annulation RIF des repliements spectraux est toujours possible) sont applicables pour N supérieur à 2 également. Notons toutefois que les filtres de synthèse obtenus à partir de la matrice des cofacteurs sont en général beaucoup plus longs que les filtres d'analyse, puisque, si l'on dénote par M_a et M_s la longueur des filtres d'analyse et de synthèse respectivement, on remarque que M_s possède la borne supérieure suivante:

$$M_s \leq (N-1)\cdot M_a \tag{3.51}$$

La complexité des bancs de filtres de synthèse n'est donc comparable à celle des bancs d'analyse que si N=2.

Deux des méthodes présentées à la section 3.1 sont aisément généralisables à N>2: la méthode du filtre complémentaire (section 3.1.1d) et la méthode directe (section 3.1.1e). Comme la seconde est une généralisation immédiate, nous n'allons présenter que la première.

Rappelons qu'une reconstruction parfaite à l'aide uniquement de filtres RIF requiert que le déterminant de la matrice soit égal à un monôme (condition C3.1). Si les

filtres d'analyse sont tous de longueur égale M_a, le déterminant aura au plus le nombre suivant de termes non-nuls (voir (2.82-84)):

$$l = M_a - N + 1 \tag{3.52}$$

Afin que le déterminant soit égal à un délai, il est donc nécessaire de satisfaire l équations. Comme il y a au départ $N \cdot M_a$ inconnues, de surcroît multipliées entre elles, nous proposons une méthode simple pour réduire le nombre d'inconnues à M_a et les équations à un système linéaire. La méthode consiste à choisir a priori N-1 lignes ou colonnes de la matrice $\mathbf{N}_p(z^N)$ en (2.36), puis de résoudre les l équations pour les M_a inconnues restantes.

Considérons le cas où N-1 colonnes de $\mathbf{N}_p(z^N)$ sont choisies a priori, ce qui revient à choisir N-1 filtres d'analyse. Le déterminant de la matrice de filtres est alors une fonction linéaire des M_a coefficients du dernier filtre [vet86b]. Comme seules l équations sont à satisfaire afin de réduire ce déterminant à un monôme, on peut soit ajouter N-1 équations supplémentaires (en imposant des conditions raisonnables au dernier filtre), ou choisir arbitrairement N-1 coefficients du dernier filtre. Evidemment, le système d'équations peut ne pas avoir de solutions dans certains cas pathologiques, similairement à ce qui a été discuté pour le cas N=2 à la section 3.1.1d.

Illustrons cette méthode par un exemple simple d'un banc d'analyse avec N=3 et M_a=7. Choisissons a priori les deux filtres suivants:

$$H_0(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} \tag{3.53a}$$

$$H_1(z) = 1 - z^{-1} + z^{-2} - z^{-3} + z^{-4} - z^{-5} + z^{-6} \tag{3.53b}$$

Choisissons d'abord le filtre $H_2(z)$ avec deux coefficients a priori comme:

$$H_2(z) = 1 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + h_4 z^{-4} + h_5 z^{-5} + z^{-6} \tag{3.53c}$$

Le déterminant de la matrice de filtres correspondante est égal à:

$$\Delta_m(z) = 2/3 \left((h_4-1)z^{-15} + (h_1-h_3)z^{-12} + (h_3-h_5)z^{-6} + (1-h_2)z^{-3} \right) \tag{3.54a}$$

En posant h_1=1, h_2=-1, h_3=1, h_4=1, et h_5=1, (3.54a) se réduit à:

$$\Delta_m(z) = 4/3\ z^{-3} \tag{3.54b}$$

A partir de la matrice des cofacteurs, nous obtenons les filtres de synthèse suivants (où les délais communs superflus ont été simplifiés):

$$G_0(z) = 1/6\ (\ 1 + z^{-1} - z^{-4} - z^{-5} + z^{-7} + z^{-8} - z^{-10}\ ) \tag{3.55a}$$

$$G_1(z) = 1/6\ (\ -z^{-1} + z^{-2} - z^{-4} + z^{-5} - z^{-7}\ ) \tag{3.55b}$$

$$G_2(z) = 1/6\ (\ -1 + z^{-2} - z^{-8} + z^{-10}\ ) \tag{3.55c}$$

Le signal reconstruit à la sortie d'un codeur en sous-bandes et utilisant les filtres donnés en (3.53) et (3.55) pour l'analyse et la synthèse est égal à:

$$\hat{X}(z) = z^{-2}\ X(z) \tag{3.56}$$

Une reconstruction parfaite a donc été obtenue, et ceci avec des filtres RIF uniquement ainsi qu'un délai minimal de z^{-2}. Remarquons toutefois que les filtres utilisés sont assez exotiques (en particulier $H_2(z)$, qui est dicté par la résolution d'un système d'équations), comme nous pouvons le constater dans la figure 3.8.

Plutôt que de choisir deux coefficients de $H_2(z)$, nous pouvons également poser deux équations supplémentaires pour celui-ci, comme par exemple $H_2(1)=0$ et $H_2(-1)=0$ (ce qui est raisonnable puisque $H_0(z)$ est un passe-bas et $H_1(z)$ un passe-haut). En lieu et place de (3.53c), on pose donc $H_2(z)$ égal à:

$$H_2(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + h_4 z^{-4} + h_5 z^{-5} + h_6 z^{-6} \tag{3.53c'}$$

Dans ce cas, le déterminant est égal à:

$$\Delta_p(z) = 2\ ((h_0-h_2)z^{-3}+(h_3-h_5)z^{-6}+(h_6-h_0)z^{-9}+(h_1-h_3)z^{-12}+(h_4-h_6)z^{-15}) \tag{3.57a}$$

Le réduction du déterminant à un délai conduit à 5 équations pour les h_i, i=0..6. Les conditions $H_2(1)=0$ et $H_2(-1)=0$ donnent encore les deux équations supplémentaires nécessaires pour une solution unique (en général) au problème. En choisissant $h_0=-1$, $h_1=0$, $h_2=-1$, $h_3=0$, $h_4=1$, $h_5=0$ et $h_6=1$, la relation (3.57a) devient:

$$\Delta_p(z) = 4 \cdot z^{-9} \tag{3.57b}$$

Notons que tous les filtres d'analyse sont à phase linéaire. De la matrice des cofacteurs, on tire les filtres de synthèse suivants (les délais inutiles ont été supprimés):

$$G_0(z) = 1/4\ (-1 + z^{-2} + 2z^{-3} - 2z^{-5} + 2z^{-7} + z^{-8} - z^{-10}) \tag{3.58a}$$

$$G_1(z) = 1/4\ (-1 + z^{-2} - 2z^{-3} + 2z^{-5} - 2z^{-7} + z^{-8} - z^{-10}) \tag{3.58b}$$

$$G_2(z) = 1/4\ (-2 + 2z^{-2} - 2z^{-8} + 2z^{-10}) \tag{3.58c}$$

Les filtres de synthèse sont également à phase linéaire. Le système ainsi défini produit le signal reconstruit suivant:

$$\hat{X}(z) = z^{-8}\ X(z) \tag{3.59}$$

La caractéristique fréquentielle des différents filtres est donnée dans la figure 3.9.

En résumé, nous avons montré comment réduire le choix des $N \cdot M_a$ coefficients d'un banc d'analyse RIF à un problème linéaire de dimension M_a, et ceci en vue d'obtenir une reconstruction parfaite avec un banc de synthèse RIF également.

3.2.2 Filtres à réponse impulsionnelle infinie

Tout comme dans le cas de N égal à 2, il est toujours possible d'éliminer le repliement spectral. Ceci est obtenu en dérivant les filtres de synthèse à partir de la matrice des cofacteurs, qui est toujours une matrice de filtres stables en vertu de la remarque R2.9.

Afin d'obtenir une reconstruction parfaite, il faut éliminer l'effet du déterminant en plaçant un filtre égal à $1/\Delta(z)$ à la sortie. Le fait d'avoir un déterminant ayant tous ses zéros à l'intérieur du cercle unité est dès lors une condition suffisante pour garantir une reconstruction parfaite. Par contre, il n'est pas démontré que cette condition soit nécessaire dans le cas général (ceci a été démontré pour N=2 et pour les filtres modulés). Néanmoins, la conjecture selon laquelle cette condition est également nécessaire semble vraisemblable. La même réflexion s'applique à la reconstruction passe-tout, qui elle est liée à l'absence de zéros du déterminant sur le

cercle unité..

Nous ne poussons pas plus loin la discussion ci-dessus, car les bancs de N filtres RII généraux (N>2, non-modulés) sont non seulement difficiles à analyser, mais également peu répandus dans la pratique.

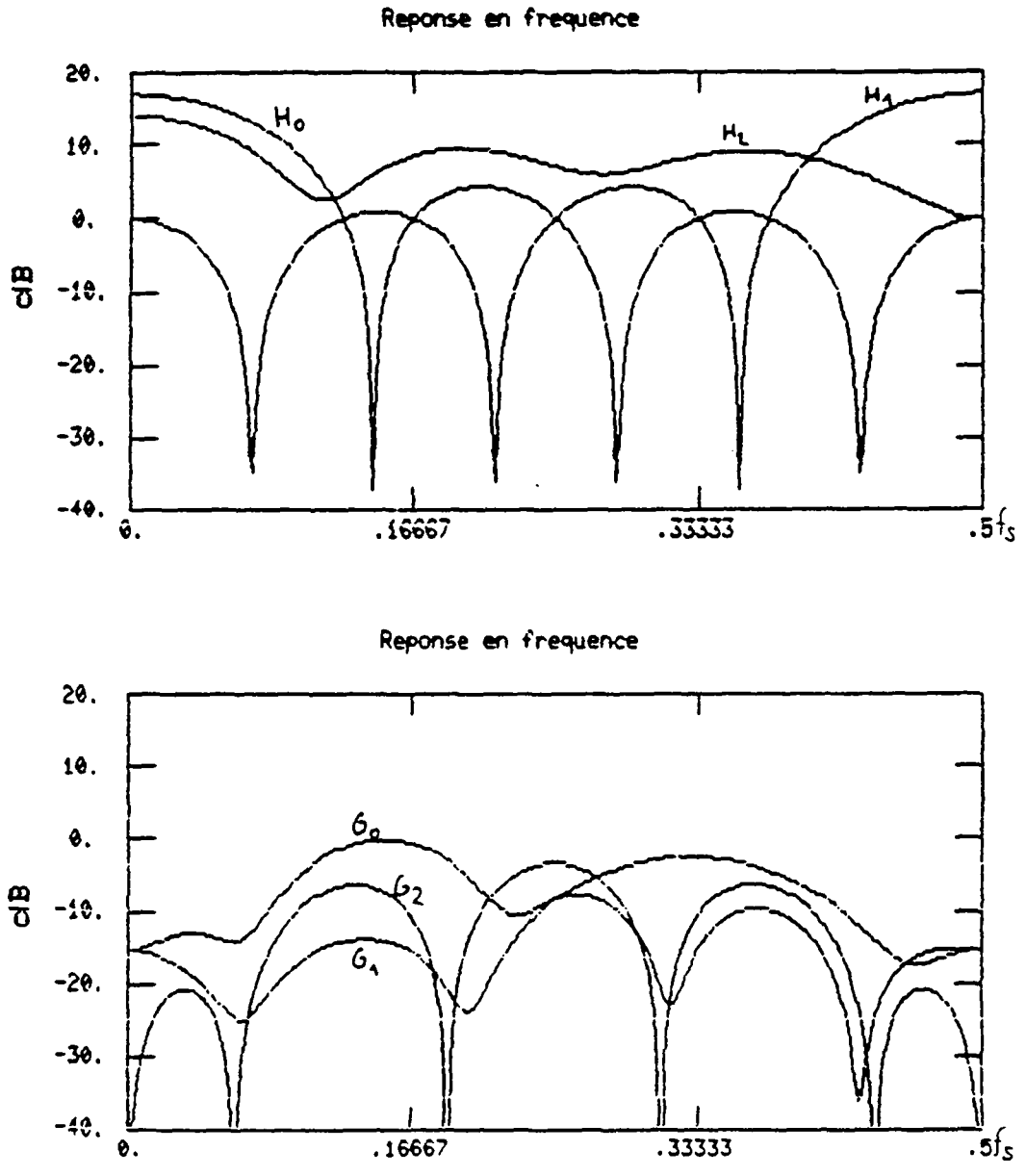

Figure 3.8: Amplitude de la réponse fréquentielle des filtres pour N=3, M_a=7, lors du choix arbitraire des deux premiers filtres ainsi que de deux coefficients du dernier filtre dans le banc d'analyse.

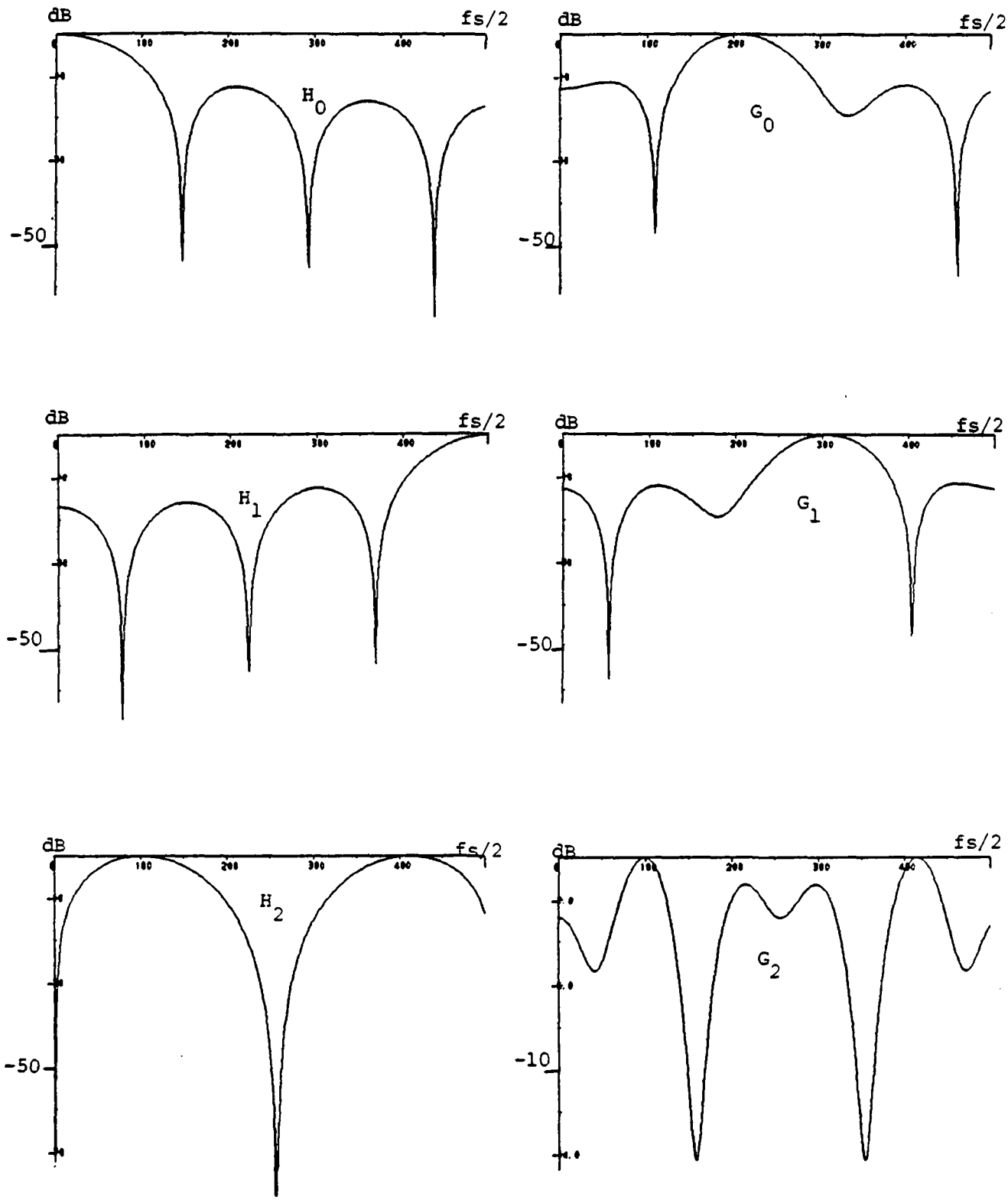

Figure 3.9: Amplitude de la réponse fréquentielle des filtres pour N=3, M_a=7, lors du choix arbitraire des deux premiers filtres et de deux contraintes supplémentaires pour le dernier filtre dans le banc d'analyse.

3.3 Bancs de N filtres modulés

D'une part, et ceci depuis les travaux de Bellanger [bel74], on sait l'importance des bancs de filtres modulés pour les transmultiplexeurs en raison de leur complexité de calcul réduite. D'autre part, Nussbaumer [nus81] a montré l'intérêt de ces bancs de filtres pour le codage en sous-bandes en introduisant le concept de bancs pseudo-QMF, qui allient une propriété d'annulation du repliement spectral avec une implantation efficace. Nous dériverons d'abord quelques résultats sur les bancs de filtres modulés généraux et indiquerons quelques problèmes qui leur sont liés. Ensuite, nous analyserons le cas important des bancs pseudo-QMF. Entre autres, ceux-ci donnent lieu à un banc de synthèse ayant une complexité comparable au banc d'analyse, et ils ne posent de surcroît pas de problème de stabilité.

3.3.1 Filtres modulés généraux

Dans un banc de filtres modulés, le i-ième filtre d'un banc de filtres modulés peut s'écrire comme (voir (2.97)).

$$H_i(z) = H_\Omega(W^i z) \tag{3.60}$$

où $H_\Omega(z)=N_\Omega(z)/D_\Omega(z)$ est un filtre prototype (typiquement, un passe-bas avec une fréquence de coupure égale à $2\pi/N$). Alors, la matrice de filtres modulés peut s'écrire comme (voir (2.100)):

$$\mathbf{H_m}(z) = 1/D(z^N) \cdot \mathbf{J} \cdot \mathbf{F}^{-1} \cdot \mathbf{L}(z) \cdot \mathbf{F} \tag{3.61a}$$

où

$$\mathbf{L}(z) = \mathrm{diag}[\, N_{\Omega 0}(z^N),\ z^{-N+1}N_{\Omega N-1}(z^N) \ldots z^{-1}N_{\Omega 1}(z^N) \tag{3.61b}$$

Pour les définitions des diverses matrices, nous renvoyons le lecteur aux relations (2.44), (2.48) et (2.97-102). Les $N_{\Omega i}(z^N)$ sont les composantes polyphases du filtre prototype (après transformation afin que le dénominateur soit de la forme $D(z^N)$). Rappelons qu'il est nécessaire que ces composantes polyphases soient à phase minimale pour que l'inverse de $\mathbf{H_m}(z)$ soit une matrice de filtres stables (voir condition C2.3). Cette condition est très restrictive. Sa signification physique pour le filtre prototype total n'est pas claire car les zéros du filtre prototype n'ont en général pas de relation directe avec ceux des composantes polyphases. Une exception apparaît si:

$$N_\Omega(z) = N'_\Omega(z) \cdot N''_\Omega(z^N) \tag{3.62}$$

où $N'_\Omega(z)$ est un filtre RIF de longueur N et $N''_\Omega(z^N)$ un filtre n'ayant que chaque N-ième coefficient différent de zéro. Appelons $n'_{\Omega i}$ le i-ième coefficient du filtre $N'_\Omega(z)$. Alors la i-ième composante polyphase de $N_\Omega(z)$ est égale à:

$$N_{\Omega i}(z^N) = n'_{\Omega i} \cdot N''_\Omega(z^N) \tag{3.63}$$

Donc, si $N''_\Omega(z)$ est à phase minimale, l'inverse de $\mathbf{H}_m(z)$ sera bien stable. Evidemment, des filtres de ce type ne sont pas de très bonne qualité (au sens traditionnel). Pourtant, il n'est pas clair si le comportement fréquentiel des composantes polyphases n'est pas plus important que celui du filtre complet. Cette façon de voir est spécifique aux bancs de filtres, et pose donc des contraintes nouvelles à la conception de filtres pour bancs de filtres.

Ce point de vue est déjà apparu lors de la conception de filtres pour transmultiplexeurs, où les composantes polyphases sont interprétées comme des passe-touts avec un comportement de phase en escalier [bel76]. Notons qu'il y a dès lors contradiction entre la contrainte de composantes polyphases passe-touts (pour que le filtre prototype soit un bon passe-bas) et celle de composantes polyphases ayant un numérateur à phase minimale (pour la stabilité de l'inverse).

Remarquons que si la synthèse est faite à l'aide de l'inverse de $\mathbf{L}(z)$, elle nécessite une complexité de calcul comparable à celle de l'analyse. Notons aussi qu'une reconstruction parfaite avec des filtres RIF n'est possible que si les filtres d'analyse sont de longueur N. Ceci se démontre exactement comme dans le cas N=2 (voir (3.8)).

Une échappatoire à ce problème des composantes polyphases à phase minimale consiste à réaliser l'annulation des repliements spectraux sans chercher la reconstruction parfaite. Si la stabilité est alors garantie, la complexité de calcul requise pour la synthèse est par contre grandement accrue (en raison de (3.51)). Notons toutefois que le banc de synthèse est également un banc de filtres modulés, donc qu'il peut être implanté efficacement. La façon la plus simple de dériver un tel banc de synthèse est la suivante: choisissons, à la place du vrai inverse de $\mathbf{H}_m(z)$, la matrice de synthèse suivante:

$$\mathbf{G}_m(z) = D(z^N) \cdot \mathbf{F}^{-1} \cdot \mathbf{L}'(z) \cdot \mathbf{F} \cdot \mathbf{J} \tag{3.64}$$

où $\mathbf{L}'(z)$ est la matrice des cofacteurs de $\mathbf{L}(z)$. Notons que $\mathbf{G}_m(z)$ est bien une matrice de filtres modulés. Dès lors, le produit $\mathbf{H}_m(z)\cdot \mathbf{G}_m(z)$ devient:

$$\mathbf{H}_m(z)\cdot \mathbf{G}_m(z) = 1/D(z^N)\cdot \mathbf{J}\cdot \mathbf{F}^{-1}\cdot \mathbf{L}(z)\cdot \mathbf{F}\cdot D(z^N)\cdot \mathbf{F}^{-1}\cdot \mathbf{L}'(z)\cdot \mathbf{F}\cdot \mathbf{J}$$

$$= \mathbf{J} \cdot \mathbf{F}^{-1} \cdot \mathbf{L}(z) \cdot \mathbf{L}'(z) \cdot \mathbf{F} \cdot \mathbf{J} \tag{3.65}$$

Mais le produit $\mathbf{L}(z)\cdot \mathbf{L}'(z)$ est simplement égal à $\Delta_1(z)\cdot \mathbf{I}$ (voir (2.109)), et (3.65) devient:

$$\mathbf{H}_m(z)\cdot \mathbf{G}_m(z) = \Delta_1(z) \cdot \mathbf{I} \tag{3.66}$$

La transmission de l'entrée à la sortie est donc égale au produit de toutes les composantes polyphases (fois un délai et une constante), ce qui montre bien la nécessité d'avoir des composantes polyphases qui soient des passe-touts.

Notons que nous avons estimé ci-dessus que les canaux étaient parfaits et nous n'avons posé aucune contrainte sur les filtres prototypes utilisés. Pourtant, nous avons vu précédemment (section 3.1.4) que si les canaux étaient non-idéaux, il était nécessaire de travailler avec des filtres d'analyse et de synthèse qui soient de bonnes approximations de filtres passe-bandes idéaux. Dans les solutions ci-dessus, il s'avère que les filtres de synthèse ne sont pas forcément de bons passe-bandes, même si tel est le cas pour les filtres d'analyse. Ceci vient du fait que certaines composantes hors-bande sont accentuées afin de permettre la suppression cohérente du repliement spectral d'une autre bande. Si dans le cas idéal ceci ne représente aucun problème, il n'en est pas de même dans des cas pratiques. Un exemple de filtre de synthèse avec mauvaise atténuation hors-bande est donné dans [smi85]. Nous insistons donc sur les limitations des solutions purement "mathématiques" car elles peuvent s'avérer décevantes en pratique.

En résumé, nous avons montré les conditions de reconstruction parfaite et de recontruction sans repliements spectraux dans le cas de bancs de filtres modulés, et ceci a mis en lumière l'importance centrale des composantes polyphases du filtre prototype. Rappelons les limitations majeures: les bancs de synthèse sont difficiles à stabiliser (reconstruction idéale), ils peuvent devenir beaucoup plus complexes (reconstruction sans repliement spectral) et les filtres de synthèse ne sont pas

forcément de bons passe-bandes.

3.3.2 Filtres pseudo-QMF

Une solution intéressante aux problèmes rencontrés plus haut est apportée par les filtres pseudo-QMF [nus81,rot83,nus84a,nus84b,chu85,mas85]. Ils ont les propriétés suivantes:

- annulation des repliements spectraux majeurs
- caractéristique passe-bande des filtres de synthèse
- analyse et synthèse stable en tous cas
- complexité de calcul de la synthèse comparable à celle de l'analyse

Nous appelons repliement spectral majeur celui qui apparaît dans les bandes adjacentes à la bande passante d'un filtre de synthèse. Dans la suite, il sera toujours admis que les autres repliements spectraux sont automatiquement supprimés en raison de la bonne atténuation hors bande des filtres de synthèse. Admettons que tant le banc de filtres d'analyse que celui de synthèse soit obtenu par modulation comme en (3.60). De surcroît, le produit de filtres correspondant à des bandes non-adjacentes est nul:

$$H(W_i z) \cdot G(W_j z) = 0 \qquad (i-j) \text{ Mod } N > 1 \tag{3.67}$$

Dès lors, il est suffisant de satisfaire les trois équations suivantes:

$$\begin{bmatrix} H(z) & H(Wz) & \cdots & H(W^{N-1}z) \\ H(Wz) & H(W^2z) & \cdots & H(z) \\ H(W^{N-1}z) & H(z) & \cdots & H(W^{N-2}z) \end{bmatrix} \begin{bmatrix} G(z) \\ G(Wz) \\ \vdots \\ G(W^{N-1}z) \end{bmatrix} = \begin{bmatrix} T(z) \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{3.68}$$

Toutes les autres équations, par exemple dans (2.129), sont automatiquement égales à 0, et ceci en vertu de (3.67).

Comme indiqué précédemment, toute une série d'approches pseudo-QMF ont été proposées. Plutôt que de toutes les passer en revue, nous allons en présenter une en particulier tout en utilisant notre formalisme matriciel pour l'analyser. Notons que celle que nous présentons est tirée de [nus84a] (qui est similaire à celle de [nus81,nus84b]), et remarquons qu'elle ressemble à celle de [rot83], qui elle est

incorrecte (...). Une approche similaire a également été développée dans [mas85] avec certaines améliorations.

Pour des raisons pratiques, les filtres sont réels et déduits à partir d'un passe-bas prototype RIF $H_\Omega(z)$ ayant une fréquence de coupure égale à $2\pi/4N$. La fonction de modulation est sinusoïdale, et la figure 3.10 montre les réponses fréquentielles du filtre prototype et du banc de filtres. Notons qu'à cause de la symétrie de la réponse fréquentielle des filtres réels, il n'est plus suffisant de tenir compte uniquement des trois équations qui apparaissent dans le cas complexe (3.68).

Les filtres d'analyse sont choisis de la façon suivante:

$$h_k(n) = h_\Omega \cdot \cos(2\pi(2k+1)(2n-N)/8N) \tag{3.69a}$$

$$H_k(z) = 1/2[W^{N(2k+1)/8} \cdot H_\Omega(W^{(2k+1)/4}z) + W^{-N(2k+1)/8} \cdot H_\Omega(W^{-(2k+1)/4}z)] \tag{3.69b}$$

Notons que si la modulation de base est évidente (vu la configuration des filtres dans la figure 3.10), il n'en est pas de même pour les facteurs de phase. Ceux-ci se justifieront par la suite, car ils permettront les annulations désirées. Similairement, les filtres de synthèse sont égaux à:

$$g_k(n) = h_\Omega \cdot \cos(2\pi(2k+1)(2n+N)/8N) \tag{3.70a}$$

$$G_k(z) = 1/2[W^{-N(2k+1)/8} \cdot H_\Omega(W^{(2k+1)/4}z) + W^{N(2k+1)/8} \cdot H_\Omega(W^{-(2k+1)/4}z)] \tag{3.70b}$$

Introduisons le vecteur $\mathbf{h}(z)$ des filtres d'analyse. Celui-ci peut être écrit comme:

$$\mathbf{h}(z) = 1/2\ \mathbf{D} \cdot [\ H_\Omega(W^{1/4}z),\ H_\Omega(W^{3/4}z) \ldots H_\Omega(W^{(2N-1)/4}z)\ ]^T$$

$$+ 1/2\ \mathbf{D}^* \cdot [\ H_\Omega(W^{-1/4}z),\ H_\Omega(W^{-3/4}z) \ldots H_\Omega(W^{-(2N-1)/4}z)\ ]^T \tag{3.71a}$$

$$\mathbf{D} = \mathrm{diag}[\ e^{-\pi/4},\ e^{-3\pi/4}, \ldots e^{-(2N-1)\pi/4}\ ] \tag{3.71b}$$

En introduisant le vecteur suivant:

$$\mathbf{h}_\Omega(z) = [H_\Omega(W^{1/4}z),\ H_\Omega(W^{3/4}z) \ldots H_\Omega(W^{(2N-1)/4}z)]^T \tag{3.72}$$

on peut récrire (3.71a) plus succinctement comme:

$$\mathbf{h}(z) = 1/2 \cdot \mathbf{D} \cdot \mathbf{h}_\Omega(z) + 1/2 \cdot \mathbf{D}^* \cdot [\mathbf{h}_\Omega(z^*)]^* \tag{3.73}$$

Similairement, le vecteur des filtres de synthèse s'écrit:

$$\mathbf{g}(z) = 1/2 \cdot \mathbf{D}^* \cdot \mathbf{h}_\Omega(z) + 1/2 \cdot \mathbf{D} \cdot [\mathbf{h}_\Omega(z^*)]^* \tag{3.74}$$

La matrice des filtres modulés est maintenant obtenue à partir de $\mathbf{h}(z)$ comme suit:

$$\mathbf{H}_m(z) = [\mathbf{h}(z), \mathbf{h}(Wz) \ldots \mathbf{h}(W^{N-1}z)]^T \tag{3.75}$$

Selon nos prémisses, le filtre prototype $H_\Omega(z)$ est suffisamment sélectif pour que

$$H_\Omega(W^i z) \cdot H_\Omega(W^j z) = 0 \qquad i \neq j \tag{3.76}$$

En utilisant cette propriété ainsi que les définitions des filtres (3.73) et (3.74), il est possible de vérifier que [nus84a]:

$$\mathbf{H}_m(z) \cdot \mathbf{g}(z) = [\ T(z),\ 0,\ 0 \ldots 0\ ]^T \tag{3.77}$$

En fait, on peut vérifier que pour les équations autres que la première dans (3.77), seuls 4 termes non-nuls (composés de produits du filtre prototype modulé) apparaissent. Les phaseurs dans (3.69) et (3.70) ont donc été choisis afin que ces termes s'annulent de façon cohérente. Remarquons que toutes les autres approches pseudo-QMF reposent sur l'annulation de ces 4 termes, avec quelques modifications dans la façon de le faire. Notons que le système pseudo-QMF (en particulier comme transmultiplexeur) peut être sensible aux distorsions de phase, et il peut être nécessaire d'égaliser la phase d'un canal réel (ce qui est relativement simple [cox86]).

La transmission T(z) de l'entrée à la sortie vaut:

$$T(z) = [\ \mathbf{h}_\Omega(z)\ ]^T \cdot \mathbf{h}_\Omega(z) + [\ \mathbf{h}_\Omega(z^*)\ ]^{*T}\ [\ \mathbf{h}_\Omega(z^*)\ ]^* \tag{3.78}$$

L'évaluation de T(z) sur le cercle unité (en admettant que la longueur M du filtre prototype est impaire) donne lieu à une norme similaire à celle obtenue avec les QMF classiques (voir (3.12)) mais avec 2N termes (2 par filtre puisque le filtre prototype est modulé en cosinus). Un exemple de banc de filtre pseudo-QMF de dimension 8 avec un filtre prototype à phase linéaire de longueur 65 est donné dans la figure 3.11. La réponse totale est égale à 1 +- 0.2dB sur tout l'axe des fréquences

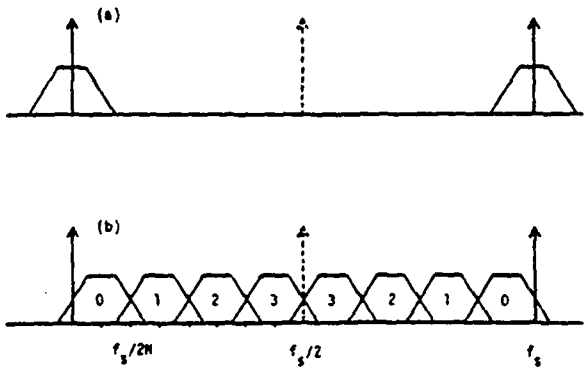

Figure 3.10: Module de la fonction de transfert
a) filtre prototype passe-bas
b) banc de filtres obtenu par modulation

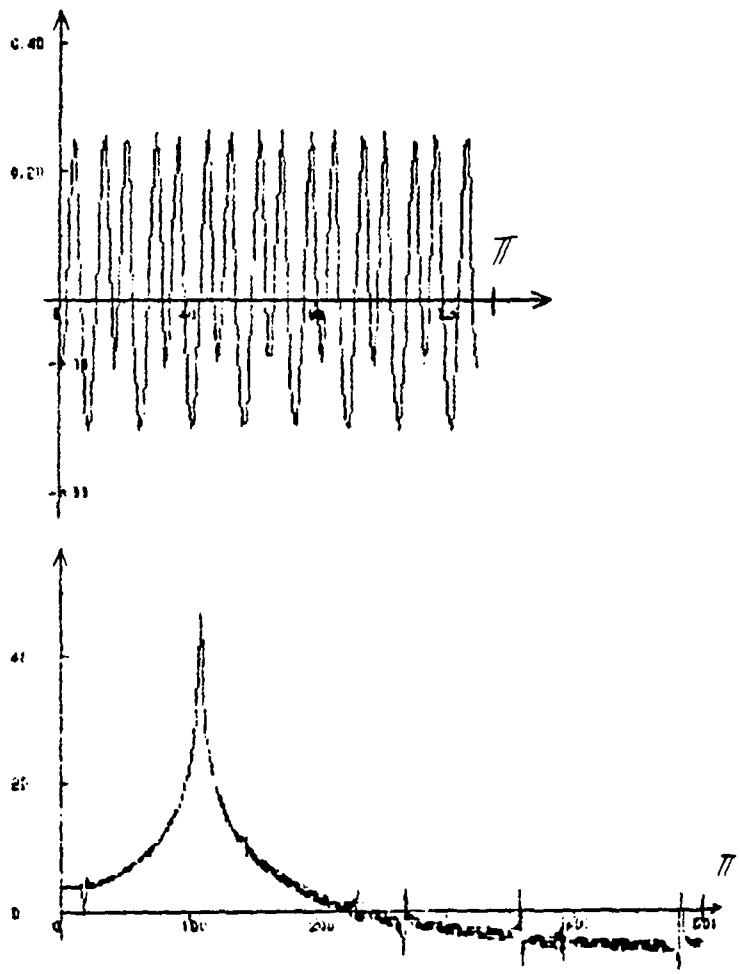

Figure 3.11 Banc pseudo-QMF de dimension 8 et avec un filtre prototype de longueur 65
a) réponse fréquentielle entrée-sortie
b) test de la suppression du repliement spectral

et le repliement spectral majeur est atténué au-delà de 40dB.

Comme nous le verrons au chapitre suivant, ces bancs de filtres pseudo-QMF ont de surcroît une implantation très efficace. Il est également à noter que les bancs de filtres pseudo-QMF ont en général un délai moindre que les bancs QMF classiques, ce qui les rend mieux adaptés au codage en temps réel (voir à ce propos l'utilisation de bancs de filtres pseudo-QMF pour le brouillage de la parole dans [cox86])

En conclusion à cette section sur les bancs de filtres modulés, rappelons qu'avec les bancs pseudo-QMF, nous avons présenté une solution pratique aux problèmes rencontrés avec les bancs de filtres modulés généraux (stabilité, complexité et qualité des filtres de synthèse). L'utilisation croissante de filtres pseudo-QMF en codage de la parole [mas85,cox86] est d'ailleurs une preuve de leur utilité pratique.

3.4 Filtres QMF complexe

Il est souvent désirable d'obtenir la représentation analytique d'un signal réel. Afin de ne pas augmenter le nombre d'échantillons par unité de temps et en vertu du théorème de Shannon, le signal analytique complexe est sous-échantillonné d'un facteur 2. Dès lors, la propriété de reconstruction, en particulier sans repliements spectraux, est importante. Une solution à ce problème a été proposée par Nussbaumer avec l'introduction des filtres QMF complexes [nus83a,gal84]. Notons que de tels filtres QMF complexes (ou CQMF) sont souvent utilisés à la suite d'un banc de N filtres réels (un arbre de filtres QMF typiquement) et permettent ainsi un codage efficace de la parole [gal86]. Ci-dessous, et à l'aide du formalisme matriciel des bancs de filtres, nous indiquons la solution générale du problème des filtres QMF complexes et nous dérivons entre autre les conditions de reconstruction parfaite.

Dans un système CQMF, le canal 0 est obtenu en modulant le signal d'entrée par cosinus de $n\pi/2$, puis en filtrant par $H_0(z)$ (qui est un passe-bas de demi-bande) et finalement en sous-échantillonnant par un facteur 2. Le canal 1 est obtenu similairement, mais en modulant par sinus de $n\pi/2$ et en filtrant par $H_1(z)$. La reconstruction est obtenue en inversant ces opérations, mais en intervertissant la modulation par cosinus avec celle par sinus. Le système complet est montré dans la figure 3.12.

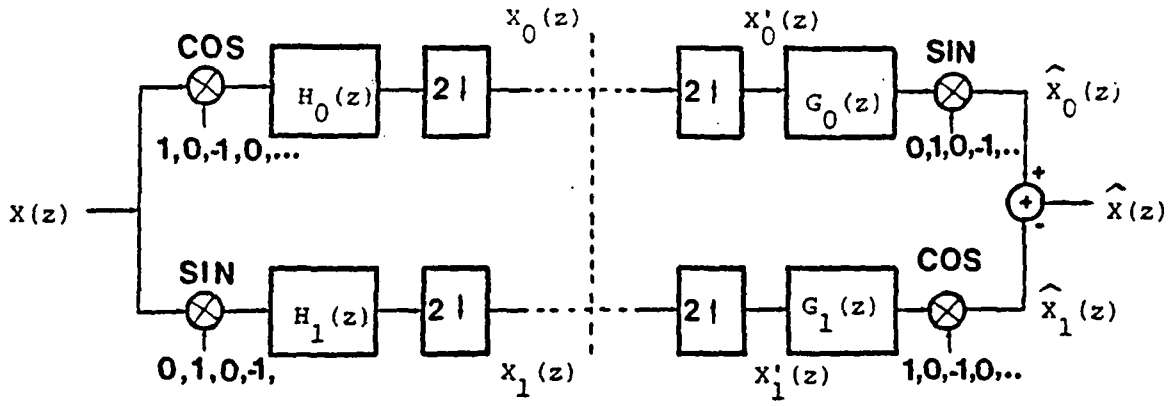

Figure 3.12: Système de dérivation du signal analytique avec reconstruction du signal original, et ceci à l'aide de filtres QMF complexes

Le système proposé dans [nus83a,gal84] utilise $H_0(z)=H(z)$, $H_1(z)=H(z)$, $G_0(z)=H(z)$ et $G_1(z)=-H(z)$. Pour notre part, nous allons garder des filtres généraux dans la

dérivation ci-dessous. Rappelons qu'un signal s(n) multiplié par $\cos(n\pi/2)$ devient un signal s'(n) et que leurs transformées en z sont liées de la façon suivante:

$$S'(z) = 1/2 \; (\; S(jz) + S(-jz) \;) \tag{3.79}$$

et ceci en vertu du théorème de modulation de la transformée en z [kun80]. De même, si la modulation se fait par $\sin(n\pi/2)$ pour obtenir un signal s''(n), nous avons que:

$$S''(z) = 1/2j \; (\; S(jz) - S(-jz) \;) \tag{3.80}$$

Donc, le signal $X'_0(z)$ dans le canal 0, après sous- et sur-échantillonnage, vaut:

$$X'_0(z) = 1/4 \; (\; X(jz) + X(-jz) \;) \cdot (\; H_0(z) + H_0(-z) \;) \tag{3.81}$$

Similairement, le signal $X'_1(z)$ devient:

$$X'_1(z) = 1/4j \; (\; X(jz) - X(-jz) \;) \cdot (\; H_1(z) - H_1(-z) \;) \tag{3.82}$$

La première composante du signal reconstruit, $\hat{X}_0(z)$, devient:

$$\hat{X}_0(z) = 1/2j \; (\; X'_0(jz) \cdot G_0(jz) - X'_0(-jz) \cdot G_0(-jz) \;) \tag{3.83}$$

c'est-à-dire, en remplaçant (3.81) dans (3.83):

$$\hat{X}_0(z) = 1/8j \; (X(z) + X(-z)) \cdot (H_0(jz) + H_0(-jz)) \cdot (G_0(jz) - G_0(-jz)) \tag{3.84}$$

Similairement, la seconde composante, $\hat{X}_1(z)$, vaut:

$$\hat{X}_1(z) = 1/8j \; (X(-z) - X(z)) \cdot (H_1(jz) - H_1(-jz)) \cdot (G_1(jz) + G_1(-jz)) \tag{3.85}$$

La sortie du système, $\hat{X}(z)$, est égale à la somme de $\hat{X}_0(z)$ et de $\hat{X}_1(z)$, et sera par conséquent une fonction de X(z) et de sa version modulée X(-z). Nous pouvons donc écrire $\hat{X}(z)$ comme:

$$\hat{X}(z) = F_0(z) \cdot X(z) + F_1(z) \cdot X(-z) \tag{3.86}$$

où $F_0(z)$ et $F_1(z)$ sont les filtres donnés par:

$$\begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} = 1/8j \begin{bmatrix} H_0(jz)+H_0(-jz) & H_1(-jz)-H_1(jz) \\ H_0(jz)+H_0(-jz) & H_1(jz)-H_1(-jz) \end{bmatrix} \cdot \begin{bmatrix} G_0(jz)-G_0(-jz) \\ G_1(jz)+G_1(-jz) \end{bmatrix} \quad (3.87)$$

Notons que le choix suivant de $G_0(z)$ et de $G_1(z)$ est **nécessaire et suffisant** pour supprimer tout repliement spectral dans la sortie du système (une constante commune aux deux filtres est toujours possible):

$$\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \begin{bmatrix} H_1(z) \\ -H_0(z) \end{bmatrix} \quad (3.88)$$

Ce vecteur est simplement la première colonne de la matrice des cofacteurs de la matrice en (3.87). Avec ce choix, la relation (3.87) devient:

$$\begin{bmatrix} F_0(z) \\ F_1(z) \end{bmatrix} = 1/4j \begin{bmatrix} (H_0(jz)+H_0(-jz)) \cdot (H_1(jz)-H_1(-jz)) \\ 0 \end{bmatrix} \quad (3.89)$$

Par exemple, avec les choix $H_0(z)=H(z)$ et $H_1(z)=H(z)$ (et les $G_i(z)$ selon (3.88)) [nus83a,gal84], on a bien le signal reconstruit suivant:

$$\hat{X}(z) = 1/4j \left(H^2(jz) - H^2(-jz) \right) \cdot X(z) \quad (3.90)$$

Dans [nus83a,gal84], on montre comment choisir H(z) de façon à ce que $\hat{X}(z)$ soit une bonne approximation de X(z) (le filtre H(z) doit être RIF, symétrique et de longueur paire). Le problème d'approximation est identique à celui qui apparaît dans les filtres QMF classiques (voir la section 3.1.1a), et la reconstruction ne peut être parfaite que si la longueur du filtre est égale à 2. Si on admet une dissymétrie dans le système, c'est-à-dire si $H_0(z) \neq H_1(z)$, il est alors possible d'obtenir une reconstruction parfaite avec des filtres RIF de longueur quelconque, et ceci en exigeant que, dans (3.89):

$$H_0(jz) + H_0(-jz) = \alpha_0 \cdot z^{-2k_0} \quad (3.91a)$$

$$H_1(jz) - H_1(-jz) = j \cdot \alpha_1 \cdot z^{-2k_1-1} \quad (3.91b)$$

Alors, en vertu de (3.86) et de (3.89), le signal reconstruit devient:

$$\hat{X}(z) = 1/4 \cdot \alpha_0 \cdot \alpha_1 \cdot z^{-2\,k_0-2k_1-1}\, X(z) \qquad (3.92)$$

Evidemment, on peut annuler l'effet d'une reconstruction imparfaite par un filtre de sortie qui réaliserait soit une reconstruction passe-tout, soit une reconstruction parfaite. Notons toutefois que la reconstruction parfaite est impossible dans le cas donné par (3.91), car la fonction de transfert à annuler est à phase linéaire (ce résultat est lié à la condition C2.2).

En résumé, nous avons montré dans cette section comment appliquer le formalisme matriciel à l'analyse des filtres QMF complexes (qui permettent de dériver le signal analytique, sous-échantillonné d'un facteur 2). Les deux résultats intéressants qui découlent de cette approche sont les suivants:

- on montre la classe générale des filtres qui réalisent la suppression des repliements spectraux (ceux-ci peuvent être RIF ou RII)
- on montre quelles conditions doivent remplir des filtres d'analyse (RIF en particulier) pour permettre une reconstruction parfaite.

3.5 Extension au cas bidimensionnel

Les performances excellentes des bancs de filtres pour le codage de la parole [gal83] nous ont amenés à proposer cette approche pour le codage d'image [vet84a]. Depuis lors, plusieurs tentatives de codage d'image en sous-bandes ont été réalisées [brd85,woo85], et ceci avec des performances relativement bonnes.

Les avantages du codage d'image en sous-bandes sont :
- simplicité d'implantation grâce à une complexité de calcul réduite
- absence d'effet de bord (comme c'est le cas dans le codage par transformée)
- possibilité de transmission progressive de l'image.

Les autres avantages (maîtrise du repliement spectral, reconstruction parfaite en l'absence de codage) sont les mêmes que pour le codage en sous-bandes mono-dimensionnel.

3.5.1 Cas non-séparable

En utilisant une fonction de sous-échantillonage non-séparable et réduisant le nombre d'échantillons d'un facteur 2, on obtient un système à 2 canaux utilisant des filtres bidimensionnels non-séparables. Un tel système est montré dans la figure 3.13.

Le processus de sous- et sur-échantillonage équivaut à une modulation par la fonction suivante:

$$f(n_1,n_2) = 1/2 \; (\; 1 + e^{j\pi(n_1+n_2)}) \tag{3.93}$$

et par conséquent, un signal avec transformée en z $S(z_1,z_2)$ sous- puis sur-échantillonné résulte en un signal avec transformée en z $S'(z_1,z_2)$ égal à:

$$S'(z_1,z_2) = 1/2 \; (\; S(z_1,z_2) + S(-z_1,-z_2) \;) \tag{3.94}$$

Il est aisé de vérifier (en utilisant (3.94)) que la sortie du système de la figure 3.13 est égale à:

$$\hat{X}(z_1,z_2) = 1/2 \; (\; H^2(z_1,z_2) - H^2(z_1,z_2) \;) \cdot X(z_1,z_2) \tag{3.95}$$

Donc, le repliement spectral est parfaitement supprimé. Afin que la reconstruction du signal soit bonne, le filtre $H(z_1,z_2)$ doit remplir une condition similaire à celle qui apparaît dans les QMF classiques (voir(3.12)):

$$H^2(e^{j\omega_1}, e^{j\omega_2}) - H^2(e^{j(\omega_1+\pi)}, e^{j(\omega_2+\pi)}) = 1 \qquad (3.96)$$

En utilisant des filtres à phase linéaire, il faut soit que la longueur selon une dimension soit paire et impaire selon l'autre, soit ajouter des délais si les dimensions sont toutes deux paires ou impaires. Un exemple de filtres est donné dans la figure 3.14.

3.5.2 Cas séparable

Si on utilise une fonction de sous-échantillonage séparable et réduisant le nombre d'échantillons d'un facteur 4 (en ne gardant que les échantillons ayant les deux indices pairs), on obtient un système tel celui de la figure 3.15. Dans ce cas, une condition suffisante pour la suppression du repliement spectral est la séparabilité du filtre prototype. Un exemple de filtres est donné dans la figure 3.16.

Comme les filtres sont séparables, on peut réaliser une division QMF selon un axe d'abord, (filtrage puis sous-échantillonage) puis selon l'autre. Ce sont des systèmes de ce type qui ont été implantés dans [brd85,woo85], et ceci avec des résultats prometteurs.

3.5.3 Implantation

Tout comme dans le cas de l'ouïe, il semble que la vision ait une sensibilité qui décroît avec l'augmentation des fréquences spatiales (voir à ce propos l'article de Hirsch et Hytton [hir81] sur la sensibilité fréquentielle). De ce fait, il semble raisonnable de faire une analyse plus fine dans les fréquences basses que dans les fréquences élevées. Ceci revient à réitérer l'analyse QMF sur l'image passe-bas uniquement. Un exemple de cette approche est donné dans la figure 3.17.

Similairement au cas mono-dimensionnel, l'analyse QMF est efficace du point de vue de la complexité de calcul. De surcroît, on peut généraliser les bancs pseudo-QMF pour le cas de l'analyse d'image [vet84a], ces bancs font alors appel à des transformations bidimensionnelles et sont très efficaces. Le tableau 3.2 indique sommairement le nombre de multiplications nécessaires pour l'implantation d'une analyse QMF en 16 canaux.

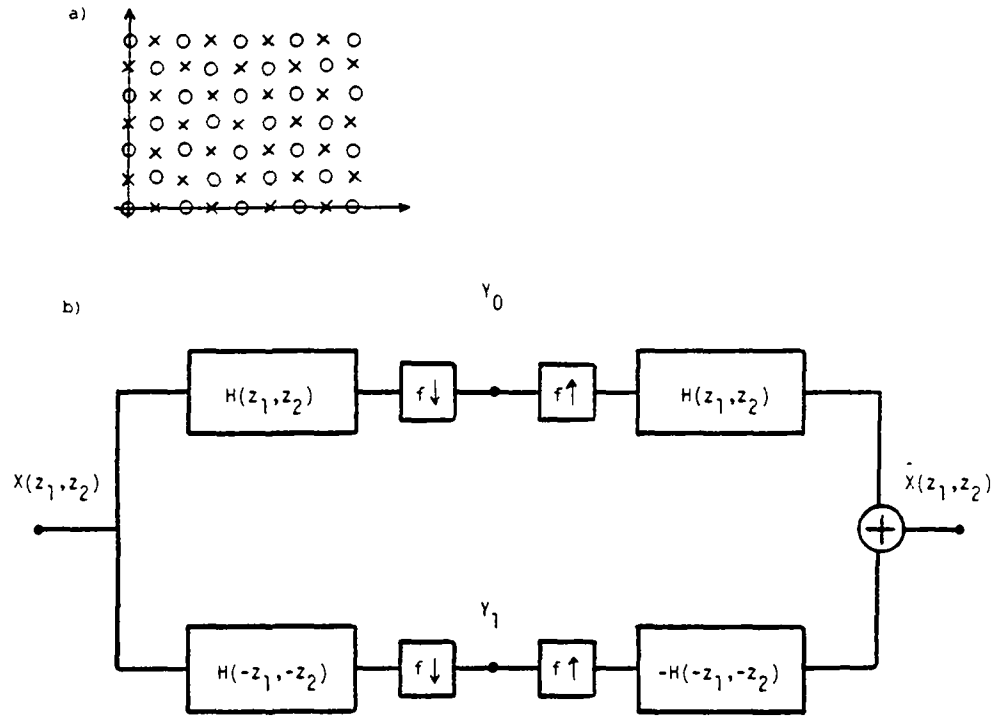

Figure 3.13: Système de codage en sous-bandes bidimensionnel non-séparable
a) fonction de sous-échantillonage, où O indique les échantillons retenus et X les échantillons abandonnés
b) configuration du système

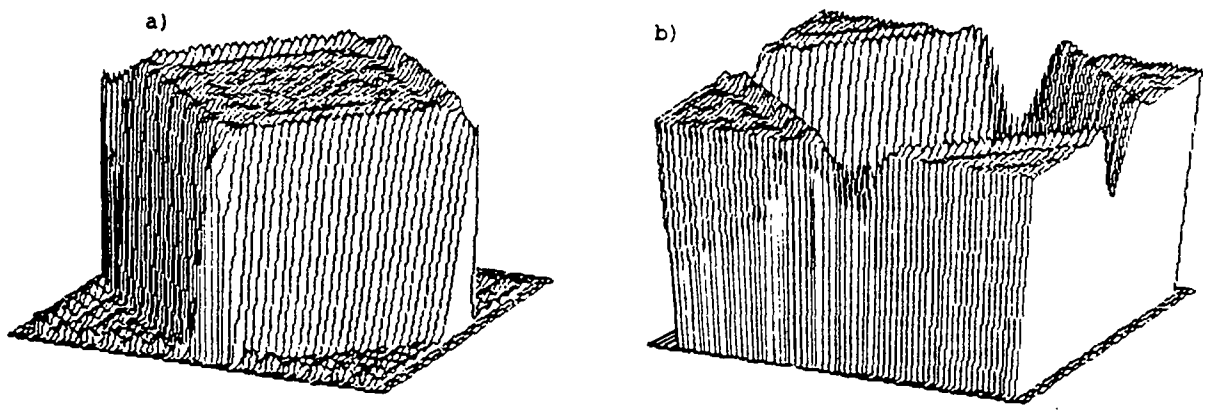

Figure 3.14: Réponse fréquentielle de filtres non-séparables
a) filtre passe-bas
b) filtre miroir

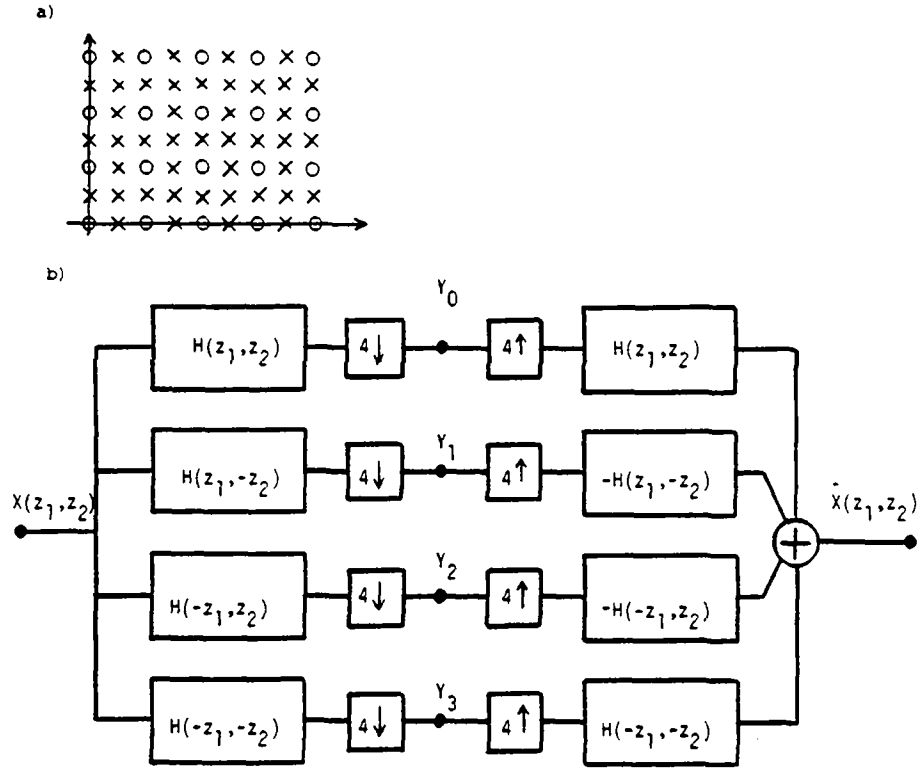

Figure 3.15: Système de codage en sous-bandes bidimensionnel séparable
a) fonction de sous-échantillonage, où 0 indique les échantillons retenus et X les échantillons abandonnés
b) configuration du système

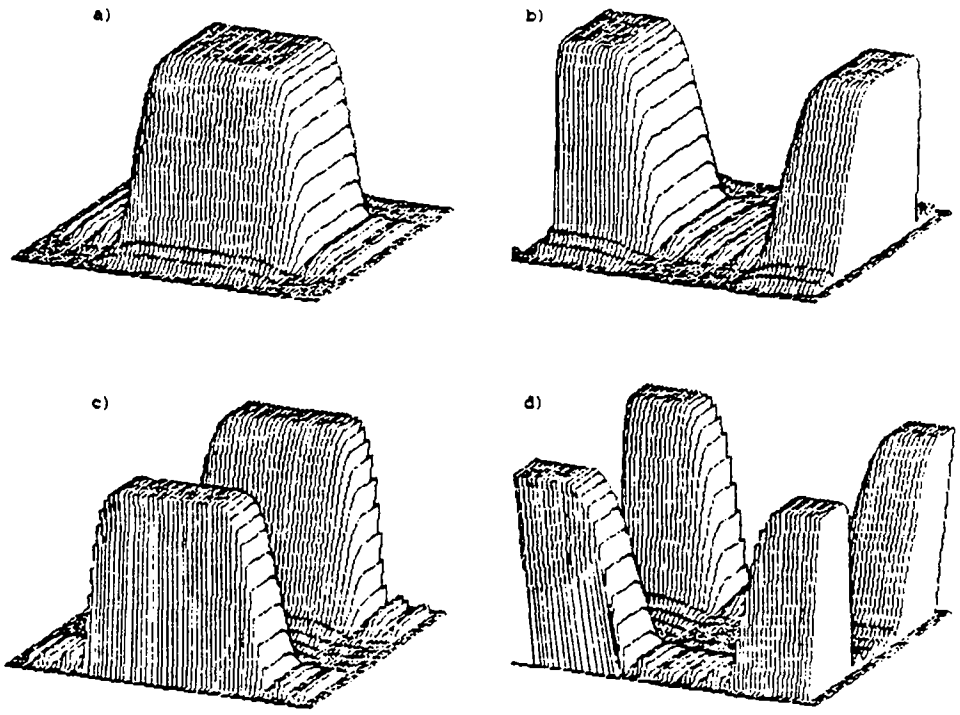

Figure 3.16: Amplitude de la réponse fréquentielle des filtres séparables
a) passe-bas $H(z_1,z_2)$
b) passe-bas/passe-haut $H(-z_1,z_2)$
c) passe-haut/passe-bas $H(z_1,-z_2)$
d) passe-haut $H(-z_1,-z_2)$

Table 3.2: Nombre de multiplications requises pour l'analyse d'une image 256 x 256 en 16 sous-bandes avec différents types de filtres

Type de filtres	nb. de mult ($\cdot 10^6$)
16 filtres non-séparables généraux (25x25)	41
4 étages de filtres QMF non-sép. (17x17, 13x13, 9x9 et 7x7)	9.6
16 filtres séparables généraux (25x25)	3.3
2 étages de filtres QMF séparables (16x16 et 8x8)	1.57
16 filtres pseudo-QMF séparable et de longueur 25	0.68

Les gains en charge de calcul sont importants, particulièrement dans le cas QMF séparable et le cas pseudo-QMF.

Notons quelques remarques en conclusion à cette section sur la généralisation bidimensionnelle de l'analyse en sous-bandes. Quoique les premiers résultats pratiques [brd85,woo85] soient encourageants, il est prématuré de porter un jugement définitif sur cette approche. Entre autre, il est difficile de comparer ces premiers essais avec les meilleurs résultats de méthodes qui ont été améliorées par des années de recherches (comme par exemple le codage par transformation). Ce qui est certain, c'est que cette méthode est très attrayante du point de vue de son implantation. Elle nécessite une complexité de calcul restreinte et utilise une architecture simple.

a)

b)

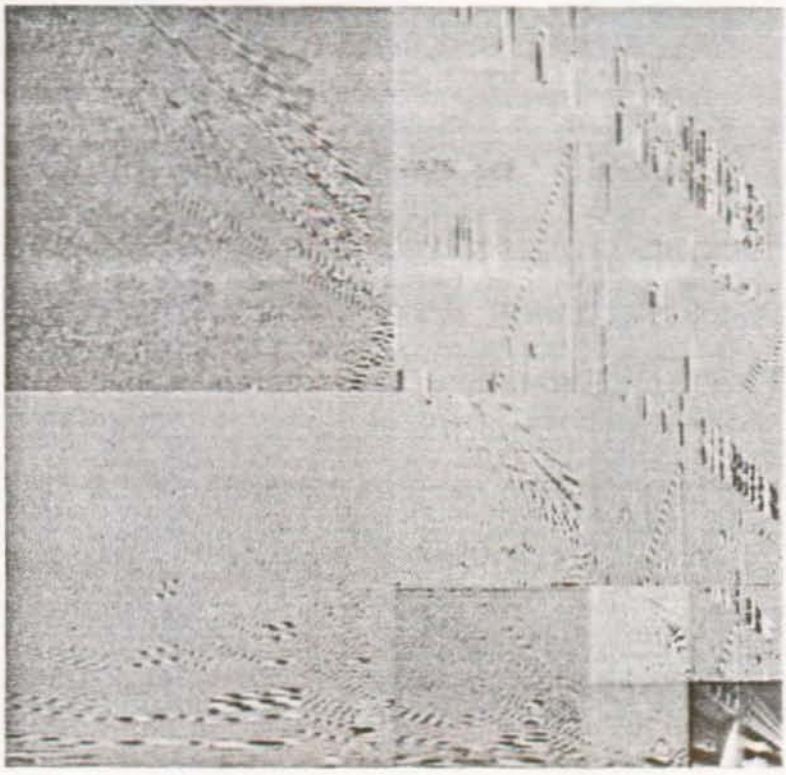

Figure 3.17: Analyse QMF d'une image par réitération successive sur l'image passe-bas

a) original

b) sortie du banc QMF

3.6 Principaux résultats du chapitre

Une première partie a exploré les bancs de deux filtres, en particulier le cas RIF. Deux méthodes analytiques (celle de la factorisation et celle du filtre complémentaire) ont été dérivées afin d'obtenir deux filtres RIF réalisant la reconstruction parfaite. Une méthode d'optimisation simultanée (des deux filtres et du déterminant) a également été suggérée. Les problèmes associés à ces différentes méthodes ont été indiqués. Le cas RII et l'application au transmultiplexage a ensuite été considéré, ainsi que l'effet de canaux non-idéaux.

Ensuite, les bancs de N filtres généraux ont été analysés et une méthode permettant la reconstruction parfaite à l'aide de filtres d'analyse et de synthèse RIF a été donnée.

Le cas important des bancs de N filtres modulés a été ensuite considéré. Les problèmes qui leur sont liés (stabilité, complexité et qualité des filtres de synthèse) ont été évoqués, et le concept de bancs pseudo-QMF a été présenté afin de pallier à ces problèmes.

Finalement, la généralisation des filtres QMF au cas complexe (calcul du signal analytique) et au cas bidimensionnel a fait l'objet des deux sections finales.

En conclusion à ce chapitre, nous pensons d'une part qu'un bon nombre de méthodes de calcul de bancs de filtres et d'applications intéressantes de ceux-ci ont été données, mais d'autre part, il nous semble évident que le sujet est loin d'être clos. A notre avis, la conception de filtres pour l'utilisation dans des bancs de filtres pose des contraintes nouvelles qui ne sauraient être satisfaites pour les méthodes traditionnelles de conception de filtres isolés.

Références bibliographiques du chapitre 3:

[ang72] A.Angot, "Compléments de Mathématiques", Masson, Paris, 1972.

[bar82] T.P.Barnwell, "Subband Coder Design Incorporating Recursive QMF and Optimum DPCM Coders", IEEE Trans. on ASSP, Vol.30, No.5, Oct.1982, pp.751-765.

[bel74] M.G.Bellanger, and J.L.Daguet, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT", IEEE Trans. on Communications, Vol. COM-22, No.9, pp. 1199-1204, Sept. 1974.

[bel76] M.G.Bellanger, G.Bonnerot and M.Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample-Rate Alteration and Filter Banks", IEEE Trans. on Acoust., Speech, Signal Processing, Vol. ASSP-24, No.2, pp. 109-114, April 1976.

[brd85] A.Brandt, "Sub-Band Coding of Videoconference Signals Using Quadrature Mirror Filters", Proc. of the IASTED Conf. on Applied Signal Processing and Digital Filtering, Paris, June 1985.

[chu85] P.L.Chu, "Quadrature Mirror Filter Design for an Arbitrary Number of Equal Bandwidth Channels", IEEE Trans. Acoust. Speech Signal Process., Vol. ASSP-33, No.1, pp.203-218, Feb. 1985.

[cox86] R.V.Cox, et al, "The Analog Voice Privacy System", Proc. ICASSP-86, Tokyo, Japan, May 1986.

[cro76] A.Croisier, D.Esteban, and C.Galand, "Perfect Channel Splitting by Use of Interpolation, Decimation, Tree Decomposition Techniques", Int. Conf. on Information Sciences/Systems, Patras, pp. 443-446, Aug. 1976.

[gal83] C.R.Galand, "Codage en Sous-Bandes: Théorie et Application à la Compression Numérique du Signal de Parole", Thèse d'Etat, Université de Nice, 1983.

[gal84] C.R.Galand, and H.J.Nussbaumer, "New Quadrature Mirror Filter Structures", IEEE Trans. on Acoust., Speech and Signal Proc., Vol.32, No.3, June 1984, pp.522-531.

[gal86] C.R.Galand, H.J.Nussbaumer, and J.B.Perini, "Magnitude-Phase Coding of Base-Band Speech Signals", Proc. 1986 Int. IEEE Conf. on ASSP, Tokyo, May 1986.

[gol80] B.G.Goldberg, "A Continuous Recursive DFT Analyser - The Discrete Coherent Memory Filter", IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-28, No. 6, pp.760-762, Dec. 1980.

[her70] O.Herrmannn, and W.Schuessler, "Design of Non-Recursive Digital Filters with Minimum Phase", Electronics Letters, Vol.6, No.11, pp.329-330, May 1970.

[hir82] J.Hirsch, and R.Hylton, "Limits of Spacial-frequency Discrimination as evidence of Neural Interpolation", J. Opt. Soc. Am., Vol. 72, No. 10, Oct. 1982, pp. 1367-1374.

[joh80] J.D.Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks", Int. Conf. On ASSP, ICASSP 80, pp.291-294, Denver, 1980.

[kai80] T.Kailath, **Linear Systems**, Prentice-Hall, Englewood Cliffs, 1980.

[kun80] M.Kunt, **Traitement Numérique des Signaux**, Editions Georgi, St.Saphorin, 1980.

[mas85] J.Masson, and Z.Picel, "Flexible Design of Computationaly Efficient Nearly Perfect QMF Filter Banks", Proc. 1985 IEEE Conf. on ASSP, Tampa, March 1985.

[mil85] P.C.Millar, "Recursive Quadrature Mirror Filters .- Criteria, Specification and Design Method", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-33, No.2, pp.413-420, April 1985.

[nus81] H.J.Nussbaumer, "Pseudo QMF Filter Bank", IBM Technical Disclosure Bulletin, Vol.24, No.6, pp 3081-3087, Nov.1981.

[nus83a] H.J.Nussbaumer, "Complex Quadrature Mirror Filters", Proc. 1983 Int. IEEE Conf. on ASSP, Boston, March 1983.

[nus84a] H.J.Nussbaumer, and M.Vetterli, "Computationally Efficient QMF Filter Banks", Proc. 1984 Int. IEEE Conf. on ASSP, San Diego, March 1984.

[nus84b] H.J.Nussbaumer, and M.Vetterli, "Pseudo Quadrature Mirror Filters", Proc. of Int. Conf. on Digital Signal Processing, Florence, Sept. 1984.

[olz74] A.Olza, et al, "Tables Numériques et Formulaires", Editions SPES, Lausanne, 1974.

[rab75] L.R.Rabiner, and B.Gold, **Theory and Application of Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1975.

[ram80] T.A.Ramstad, and O.Foss, "Sub-band Coder Design Using Recursive Quadrature Mirror Filters", in **Signal Processing: Theories and Applications**, EUSIPCO 1980, North-Holland, Amsterdam.

[rot83] J.H.Rothweiler, "Polyphase Quadrature Filters - A New Subband Coding Technique", *Proc.* 1983 Int. IEEE *Conf.* on ASSP, pp 1280-1283, Boston, March 1983.

[sch85] F.Schmitt, "Simulation d'un Codeur QMF avec Prédiction Linéaire", Travail de Diplôme, LIT-EPFL, 1985.

[smi84] M.J.T.Smith, T.P.Barnwell, "A procedure for designing exact reconstruction filterbanks for tree structured sub-band coders", Proc. IEEE ICASSP-84, San Diego, March 1984.

[smi85] M.J.T.Smith, T.P.Barnwell, "A Unifying Framework for Analysis/Synthesis Systems Based on Maximally Decimated Filter Banks", Proc. IEEE ICASSP-85, pp. 521-524, Tampa, March 1985.

[vet84a] M.Vetterli, "Multi-Dimensional Sub-Band Coding: Some Theory and Algorithms", *Signal Processing*, Vol. 6, No.2, pp. 97-112, Feb. 1984.

[vet86b] M.Vetterli, "Filter Banks Allowing Perfect Reconstruction", à paraître dans Signal Processing, Avril 1986.

[vet86c] M.Vetterli, "Perfect Transmultiplexers", Proc. 1986 Int. IEEE Conf. on ASSP, Tokyo, May 1986.

[woo85] J.W.Woods, and S.D.O'Neil, "Sub-Band Coding of Images", soumis aux IEEE Trans. on ASSP, June 1985.

4 Complexité de calcul de bancs de filtres

A priori, on pourrait penser que les bancs de filtres nécessitent une charge de calcul considérable, étant donné qu'ils sont constitués par N filtres distincts. Pourtant, un exemple simple infirme déjà cette hypothèse: un seul filtre RIF de longueur M requiert M multiplications et M-1 additions, mais ceci est également la complexité requise par un banc de N filtres RIF de longueur M et sous-échantillonné d'un facteur N. Si, de surcroît, les filtres sont liés les uns aux autres par une relation de modulation, la situation sera encore plus favorable pour le banc de filtres, comme nous le verrons plus loin. En général, nous pouvons remarquer que la plupart des bancs de filtres ont une complexité égale ou inférieure à celle d'un seul filtre sans sous-échantillonnage.

L'exemple ci-dessus illustre les deux caractéristiques dont nous pouvons tirer parti afin de réduire la charge de calcul des bancs de filtres: le sous-échantillonnage et la modulation. Notons que seuls des bancs d'analyse sont considérés par la suite, car les bancs de synthèse leur sont liés par une relation de dualité et ont donc le même complexité. Nous utiliserons ici souvent la représentation polyphase du signal d'entrée, de façon à séparer le signal en composantes indépendantes, ce qui clarifie les questions de charge de calcul. Remarquons enfin que la complexité de calcul des bancs de filtres a été un sujet de recherche constant, tant dans le contexte des codeurs en sous-bandes [nus81,gal84] que dans celui des transmultiplexeurs [bel74,bel82].

Dans ce chapitre, nous considérerons d'abord les arbres de filtres, en particulier dans le cas QMF. Nous présenterons les techniques classiques (dans le domaine temporel), puis des méthodes par transformée qui tirent parti des symétries et des réductions de fréquence d'échantillonnage. Ces méthodes efficaces sont, à notre connaissance, originales, quoique handicapées par le retard inhérent à de telles approches. Ensuite, nous explorerons le cas des bancs de N filtres, particulièrement si ceux-ci sont modulés. Après une revue de la méthode classique (par réseau polyphase et FFT [bel74]) et la méthode par transformée de Nussbaumer [nus83], nous proposerons une méthode différente pour calculer le banc de filtres modulés dans le domaine transformé, et qui donne lieu à une complexité de calcul similaire.

Remarquons que si dans la suite, une référence est faite au nombre d'opérations requis par une transformation de Fourier ou en cosinus, nous utiliserons par avance les résultats du chapitre 5 sur les transformées rapides.

. 4.1 Bancs de filtres en arbre

Les bancs de filtres en arbre, surtout du type QMF, sont fort répandus [gal84]. Quoique de structure complexe, ils sont relativement efficaces en ce qui concerne la charge de calcul. Nous considérons ci-dessous la complexité de ces bancs de filtres, en particulier dans le cas où les bancs élémentaires sont modulés et de dimension 2. Notons que le cas particulier d'arbres de filtres où la longueur des filtres est égale à la dimension du banc a été considéré dans [vet83].

Nous considérerons d'abord ci-dessous l'évaluation des blocs élémentaires de deux filtres, ensuite leur mise en cascade, et finalement l'évaluation du banc de filtres complet dans le domaine de Fourier.

4.1.1 Blocs élémentaires

Dans le cas QMF, un bloc élémentaire est constitué de deux filtres liés par une modulation par $(-1)^n$. Appelons $H_{ij}(z)$ le filtre qui se trouve au i-ième étage et à la j-ième branche d'une décomposition en arbre. Nous pouvons donc écrire:

$$H_{i(2j+1)}(z) = H_{i(2j)}(-z) \tag{4.1}$$

mais également:

$$H_{i(2j)}(z) = H_{i(2k)}(z) \tag{4.2a}$$

$$H_{i(2j+1)}(z) = H_{i(2k+1)}(z) \tag{4.2b}$$

La relation (4.2) vient du fait que, dans un étage particulier, on utilise toujours les mêmes filtres passe-hauts et passe-bas (ce qui n'est pas forcément le cas d'un étage à l'autre). Dans la suite, nous admettrons que les filtres sont RIF (le cas RII pouvant être traité en appliquant la décomposition polyphase de la section 2.2.2).

Un bloc élémentaire de deux filtres est évalué en effectuant la somme et la différence des deux composantes polyphases tirées du filtre prototype [gal84] (figure 4.1). Si la longueur du filtre prototype est égale à M_i, il faudra donc le nombre suivant d'opérations pour chaque nouvel échantillon d'entrée:

$$M_i/2 \text{ multiplications et } M_i/2 \text{ additions} \tag{4.3}$$

Si M_i est impair, (4.3) est simplement la moyenne sur deux échantillons d'entrée successifs. Notons que si le filtre est symétrique et de longueur paire, il n'est pas possible de réduire le nombre d'opérations en (4.3). La raison en est qu'alors les

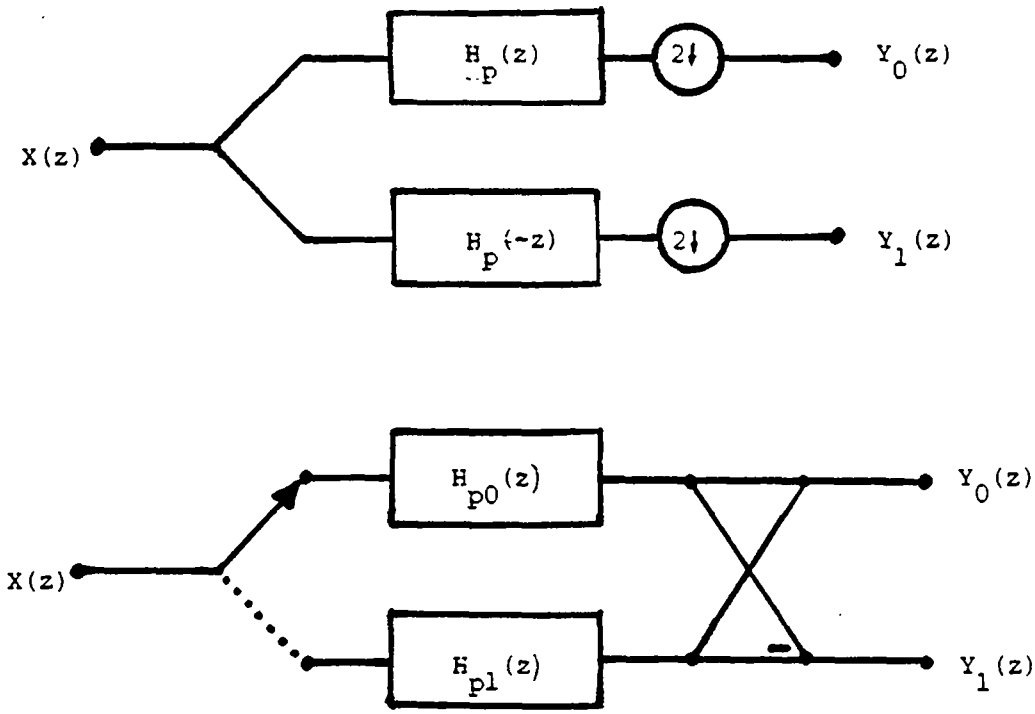

Figure 4.1 Evaluation d'un bloc élémentaire QMF
a) système initial
b) évaluation à l'aide des composantes polyphases et d'une somme/différence

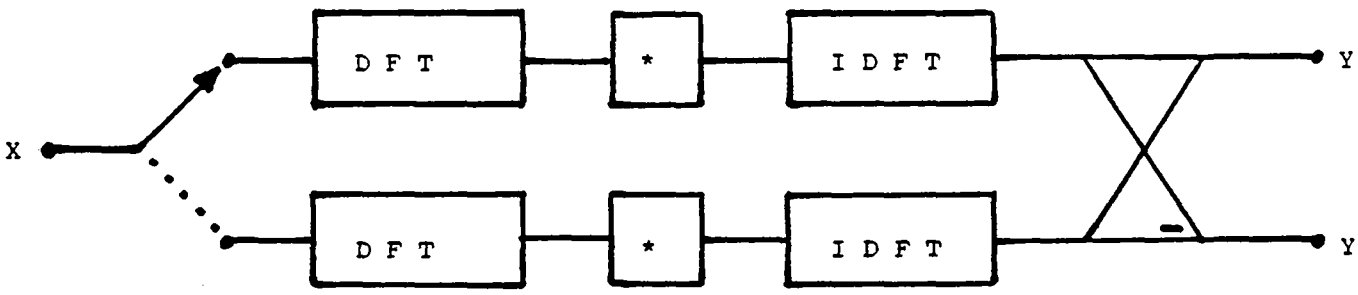

Figure 4.3 Evaluation du bloc élémentaire de la figure 4.1 par transformation

composantes polyphases ne sont pas symétriques, et quoique les mêmes coefficients apparaissent dans les deux composantes, les signaux d'entrée sont indépendants de sorte qu'aucun gain n'est possible.

Par contre, si la longueur est impaire, les composantes polyphases sont symétriques, et un gain d'un facteur 2 est possible sur le nombre de multiplications [gal84]. Si, de surcroît, le filtre est demi-bande, il aura un seul coefficient impair non-nul, divisant donc encore par 2 le nombre de multiplications et d'additions. L'inconvénient des filtres QMF de longueur impaire vient du fait qu'ils nécessitent un filtre correcteur à la sortie [gal84].

Dans le cas à phase minimum/maximum (voir [smi84,gal85] et la section 3.1.1b), un filtre passe-bas (à phase minimale et de longueur paire) est inversé dans le temps et modulé par $(-1)^n$ pour produire le filtre passe-haut (voir la relation 3.15). Il est possible de vérifier [gal85] qu'alors les opérations peuvent être groupées de façon à faire apparaître des rotations, réduisant donc le nombre de multiplications de 25% (au prix de 25% d'additions supplémentaires). Notons que la réduction de 50% (possible si on inverse un filtre dans le temps) n'est pas applicable ici, et ceci en raison du sous-échantillonnage à la sortie, et du fait que la longueur est paire.

Le tableau 4.1 résume la complexité pour l'évaluation de blocs élémentaires (2 filtres sous-échantillonnés d'un facteur 2) constitués de différents filtres. Notons que la synthèse requiert la même complexité.

Tableau 4.1 *Complexité de calcul pour l'évaluation de blocs élémentaires (2 filtres sous-échantillonnés d'un facteur 2) et ceci pour un nouvel échantillon d'entrée*

Filtre RIF	longueur	$H_0(z)$	$H_1(z)$	mults.	adds.
quelconque	quelc.	$H_0(z)$	$H_1(z)$	M	M
phase linéaire	quelc.	$H_0(z)$	$H_1(z)$	M/2	M
modulé quelc.	quelc.	H(z)	H(-z)	M/2	M/2
mod. phase lin.	pair	H(z)	H(-z)	M/2	M/2
" " "	impair	H(z)	H(-z)	M/4	M/2
" demi bande*	impair	H(z)	H(-z)	M/8	M/4
phase min/max	pair	H(z)	$z^{-M+1}H(z^{-1})$	3M/4	5M/4

* la complexité du filtre correcteur n'a pas été prise en considération

4.1.2 Mise en cascade

Un banc de N filtres, où N est une puissance de M (typiquement 2), peut être obtenu par mise en cascade de bancs de dimension M. Nous considérerons ici plus particulièrement le cas QMF. Même si la complexité de calcul ne peut être directement réduite lorsque l'arbre de filtres est considéré dans son ensemble, il est possible d'envisager des structures de calcul différentes, réunissant par exemple une partie des calculs dans une transformation rapide.

Deux modifications peuvent changer la structure d'un arbre de filtres: déplacement des sommes/différences et déplacement des sous-échantillonnages, et ceci en direction de la sortie. Il est aisé de vérifier que la somme/différence dans la figure 4.1 peut être reportée plus loin, à condition que la fonction de transfert qui suit la sortie 0 soit égale à celle qui suit la sortie 1. Ceci est bien le cas dans les bancs QMF en vertu de (4.2). Le déplacement du sous-échantillonnage vers la sortie est encore plus immédiat, étant donné qu'il n'est soumis à aucune contrainte et qu'il suffit de modifier les fonctions de transfert de façon adéquate (une fonction en z placée après un sous-échantillonnage de N équivaut à la même fonction, mais en z^N et placée avant).

En appliquant ces diverses transformations, on obtient une variété d'architectures qui va de la structure la plus arborescente possible (sommes/différences et sous-échantillonnage répartis) à la structure la plus parallèle possible (sommes/différences réunies, sous-échantillonnage final de N). Notons qu'à moins de tronquer les filtres parallèles [gal84], toutes ces structures ont une complexité de calcul identique. Remarquons toutefois que la réunion des sommes/différences à la sortie donne lieu à une transformation de Walsh-Hadamard (WHT) de dimension N [bea75], ce qui peut être avantageux lors d'une implantation. La figure 4.2 indique quelques possibilités pour le cas N=4. Notons que dans le cas 4.2e, les filtres sont très longs et nécessitent une charge de calcul plus élevée que dans les autres cas. Il est alors possible de tronquer les réponses impulsionnelles [est77,gal84], à condition d'accepter une atténuation non-infinie du repliement spectral. Dans ce dernier cas, cette approche permet de réduire tant la complexité structurelle qu'arithmétique des bancs QMF.

Dans la figure 4.2, on a admis que les bancs de deux filtres étaient modulés (relation (4.1)) et que les branches succédant à ces deux filtres étaient identiques (relation (4.2)). Appelons M_i la longueur des filtres à l'étage i. Alors, et avec (4.3), un banc de dimension $N=2^p$ utilise, par échantillon d'entrée:

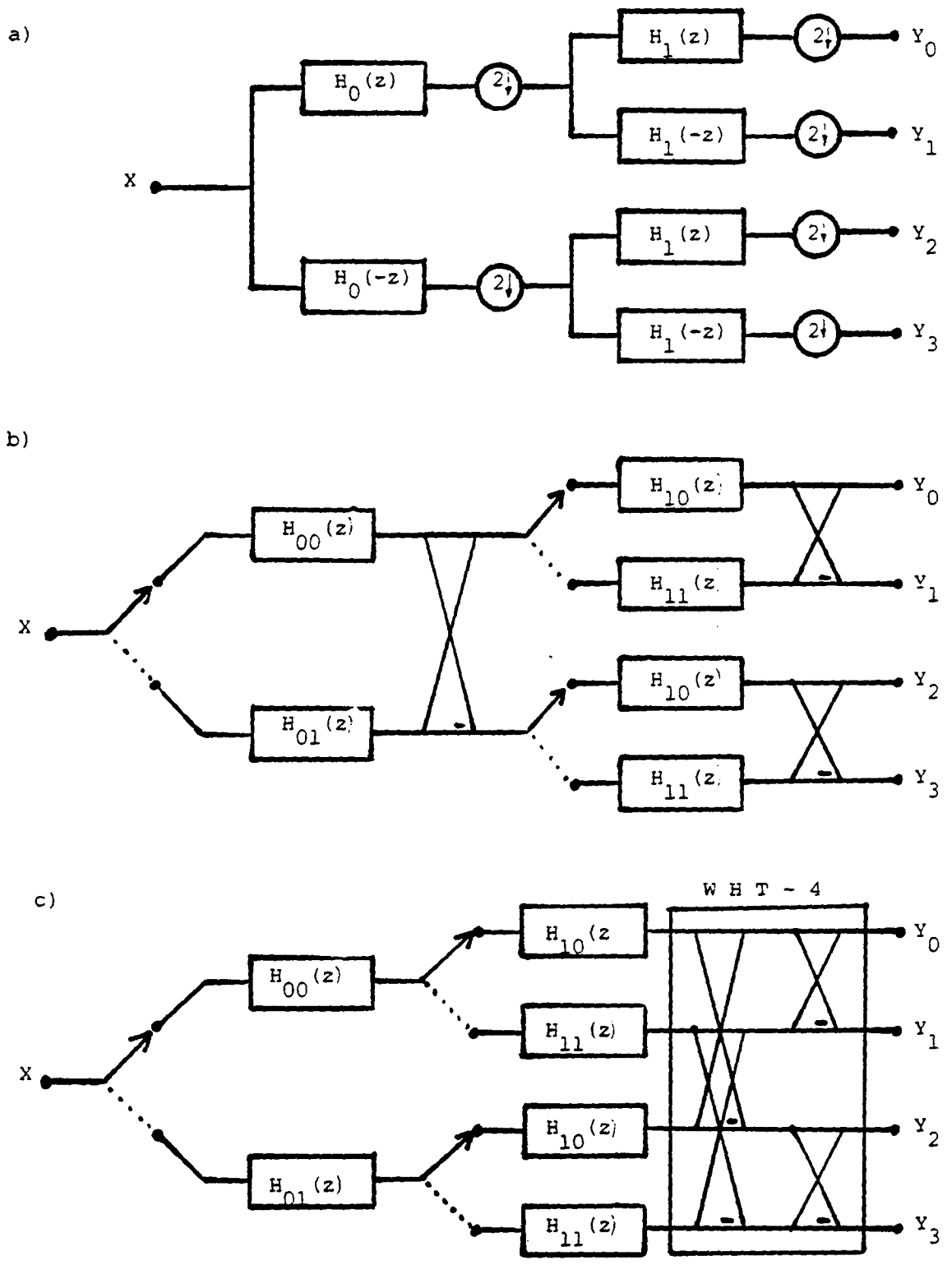

Figure 4.2 Diverses architectures possibles pour le calcul d'un arbre QMF, N=4.
a) arbre d'origine
b) décomposition des filtres en composantes polyphases, sous-échantillonnage et sommes/différences répartis
c) sous-échantillonnage réparti, sommes/différences réunies en une WHT de longueur 4

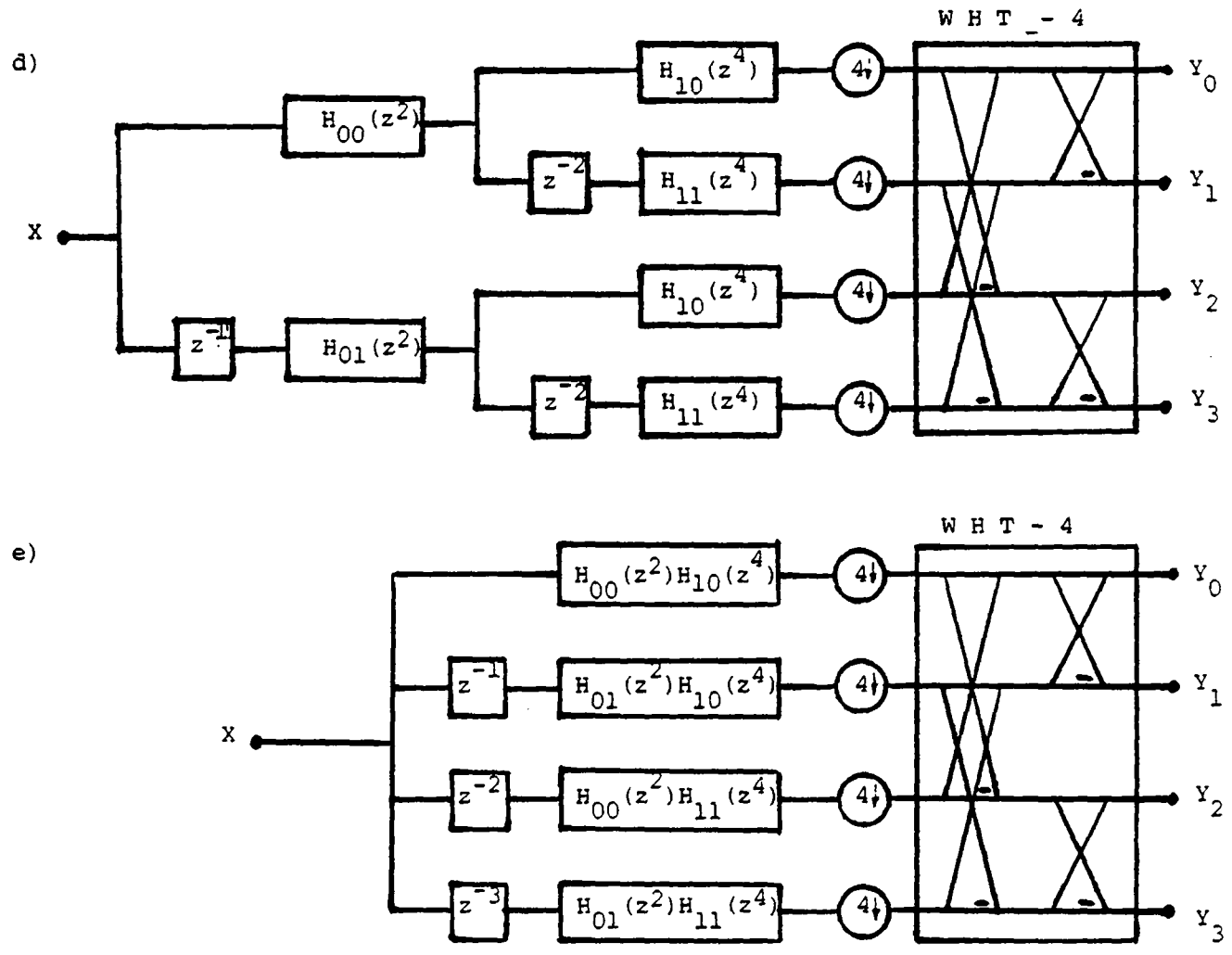

Figure 4.2 Diverses architectures possibles pour le calcul d'un arbre QMF, N=4, suite.

d) Sous-échantillonnage et sommes/différences réunis, calcul des filtres en arbre

e) Sous-échantillonnage et sommes/différences réunis, calcul des filtres en parallèle

$$1/2 \cdot \sum_{i=0}^{p-1} M_i \quad \text{multiplications et additions} \tag{4.4}$$

Souvent, la longueur des filtres est divisée par 2 à chaque étage [gal84]. Dans ce cas, (4.4) devient:

$$M_0 \cdot (1 - 2^{-P}) \quad \text{multiplications et additions} \tag{4.5}$$

Ceci équivaut à environ M_0 opérations par échantillon d'entrée. Si les filtres ont des caractéristiques supplémentaires (symétrique et de longueur paire par exemple), alors il est possible de reporter les gains de la table 4.1 aux bancs constitués de tels filtres. Notons toutefois que le cas du banc QMF classique (filtres à phase linéaire et de longueur paire) requiert la complexité donnée par (4.5).

4.1.3 Evaluation dans le domaine de Fourier

Tout comme dans le cas de l'évaluation directe (figure 4.2), il est possible d'adapter un grand nombre d'architectures différentes pour le calcul par transformation de Fourier. Nous ne considérerons ci-dessous que deux cas particuliers différents, le premier calculant les convolutions de façon répartie (correspondant à la figure 4.2d et intéressant d'un point de vue de complexité de calcul) et le second évaluant les filtres en parallèle (figure 4.2e et ayant une structure simple).

Pour la transformation de Fourier, nous utiliserons les complexités de calcul qui seront dérivées au chapitre 5. Notons que la convolution par un filtre réel nécessite 3M/2-1 multiplications et 3M/2-3 additions dans le domaine transformé (où M est la longueur de la convolution circulaire). Par ailleurs, une transformation de Walsh-Hadamard de dimension N requiert $N\text{Log}_2(N)$ additions [bea75].

Dans la suite, nous admettrons que les filtres sont réels et RIF (les filtres RII peuvent être approximés par une réponse impulsionnelle tronquée mais suffisamment longue), et que le signal d'entrée est également réel et de longueur finie. Nous négligerons donc les additions supplémentaires éventuelles et peu nombreuses que requiert par exemple la méthode de "overlap and add" [nus82] si le signal d'entrée est infiniment long (l'augmentation est de moins d'une addition par échantillon). Il est entendu que la convolution apériodique est réalisée par une convolution circulaire de longueur suffisante (longueur $\geq M_s+M_i-1$, où M_s est la longueur du signal et M_i la longueur du filtre). Notons que lors de la mise en cascade, cette relation doit être respectée à toutes les étapes du calcul.

Appelons M_f la longueur de la transformation de Fourier. Alors, il est nécessaire et suffisant que:

$$M_f \geq M_s + \sum_{i=0}^{p-1} (2^i M_i - 2^i) \qquad (4.6)$$

où p est le nombre d'étages de l'arbre de filtres. Si (4.6) est respecté, on peut sans autre évaluer la convolution apériodique comme une convolution circulaire dans le domaine de Fourier.

Le calcul d'un bloc élémentaire tel que celui de la figure 4.1 est illustré dans la figure 4.3. Notons qu'il est avantageux de calculer séparément les deux filtres correspondant aux composantes polyphases $H_{p0}(z^2)$ et $H_{p1}(z^2)$.

Considérons maintenant l'évaluation d'un arbre de filtres où les sommes/différences ont été réunies à la sortie, mais où la décimation est répartie (voir la figure 4.2d). Dans ce cas, la mise en cascade fait suivre une IDFT par une DFT, mais avec, entre les deux, un distributeur qui sépare les échantillons d'indice pair et impair. Nous allons donc chercher à simplifier ces deux transformations. Appelons x(n) et y(k) l'entrée et la sortie de la DFT inverse de longueur M (M pair). Alors, les échantillons d'indice pair peuvent être écrits comme:

$$y(2k) = \sum_{n=0}^{M-1} x(n)\, W_M^{-2nk} = \sum_{n=0}^{M/2-1} (x(n)+x(n+M/2))\, W_{M/2}^{-nk} \qquad (4.7)$$

où W_M est égal à $e^{-j(2\pi/M)}$. Les échantillons pairs sont donc obtenus à l'aide de M/2 additions (M/2 additions complexes mais dont la moitié est superflue) ainsi que d'une IDFT de longueur M/2. Cette dernière disparaît en choisissant la transformée subséquente de même longueur. Les échantillons d'indice impair sont obtenus similairement:

$$y(2k+1) = \sum_{n=0}^{M-1} x(n)\, W_M^{-n(2k+1)} = \sum_{n=0}^{M/2-1} W_M^{-n} \cdot (x(n)-x(n+M/2))\, W_{M/2}^{-nk} \qquad (4.8)$$

c'est-à-dire à l'aide de 3M/4 multiplications (M/4 multiplications complexes), 5M/4 additions ainsi qu'une IDFT de longueur M/2. Notons toutefois que ces multiplications peuvent être inclues dans la convolution qui suit immédiatement après, et il ne reste donc que M/2 additions à effectuer.

Rappelons qu'à tous les étages, il faut évaluer les différentes convolutions, et qu'à la sortie, le résultat final est obtenu à l'aide d'une WHT de dimension N (pour chaque ensemble de N échantillons de sortie). Admettons maintenant que les filtres du i-ième étage soient de longueur $K \cdot 2^{p-i-1}$, c'est-à-dire que leur longueur soit divisée par 2 à chaque étape (ce qui est raisonnable en pratique [gal83,gal84]. Le filtre qui est équivalent à la cascade des p filtres apparaissant dans une branche de l'arbre est alors de longueur:

$$L = p \cdot K \cdot 2^{p-1} - 2^p + 2 \qquad (4.9)$$

Maintenant, il suffit de choisir la longueur du signal, M_s, de telle façon à ce que $M_f = L + M_s - 1$ soit une puissance de 2.

Récapitulons brièvement les opérations nécessaires:

- à l'entrée, il faut calculer deux DFT de longueur $M_f/2$, une pour les échantillons d'indice pair et l'autre pour ceux d'indice impair
- à chaque étage, il faut calculer $3M/2$ multiplications et additions pour évaluer les différentes convolutions. Notons que les filtres deviennent plus courts, mais que leur nombre augmente.
- après chaque étage, sauf après le dernier, il faut séparer les échantillons d'indice pair et impair (voir (4.8-9)). Ceci requiert à chaque fois M_f additions.
- à la sortie, il faut calculer 2^p IDFT de longueur $M_f/(2^p)$ et portant sur des séquences hermitiennes.
- finalement, il faut évaluer une WHT de dimension $N=2^p$ pour chaque ensemble de N échantillons de sortie, c'est-à-dire $M_f/(2^p)$ transformations, ou:

$$M_f/(2^p) \cdot 2^p \cdot p = p \cdot M_f \text{ additions} \qquad (4.10)$$

En tout, et ceci pour $M_s = M_f - L + 1$ échantillons d'entrée, il faut donc:

$$2\ \mathrm{DFT}(M_f/2) + 2^p\ \mathrm{DFT}(M_f/(2^p)) \qquad (4.11a)$$

$$3 \cdot p \cdot M_f/2 \quad \text{multiplications} \qquad (4.11b)$$

$$5 \cdot p \cdot M_f/2 \quad \text{additions} \qquad (4.11c)$$

Notons que si les filtres sont à phase linéaire mais de longueur paire, aucun gain n'est possible. Si la longueur est impaire, un gain est possible, mais seulement lors de l'évaluation des composantes polyphases correspondant aux indices pairs (pour ceux d'indices impairs, il faut encore effectuer la multiplication correspondant à (4.8)). Les gains sont de l'ordre de $M_f/4$ multiplications et de $3M_f/4$ additions par étage.

Prenons un exemple simple. Il faut diviser un signal en 16 signaux, et ceci avec des filtres de longueur 128, 64, 32 et 16 pour les étages 0, 1, 2 et 3 respectivement. Avec la méthode directe discutée à la section précédente et selon la relation (4.5), il faut (par échantillon d'entrée):

120 multiplications et 120 additions (4.12)

La longueur du filtre équivalent, selon (4.9), est égale à 498. Choisissons de calculer une convolution circulaire de longueur 1024, donc de prendre 527 échantillons d'entrée suivis de 497 zéros. Avec (4.12) et la complexité pour la transformation de Fourier du chapitre 5 (voir les relations (5.29) et (5.42) ou le tableau 5.7) il résulte de (4.11) que 10788 multiplications 28232 additions sont nécessaires, c'est-à-dire, par échantillon d'entrée:

20.45 multiplications et 53.57 additions (4.13)

Il y a donc un gain d'un facteur 6 en ce qui concerne les multiplications et de 2.25 pour les additions. La figure 4.4 illustre schématiquement les opérations à effectuer dans l'exemple ci-dessus.

Considérons maintenant une approche parallèle et introduisons la convention suivante: le filtre total, de l'entrée à la sortie i, $F_i(z)$, est obtenu en utilisant à l'étage k, le filtre suivant:
- passe-bas si $\lfloor i/(2^{p-k-1}) \rfloor$ est pair
- passe-haut si $\lfloor i/(2^{p-k-1}) \rfloor$ est impair

Ceci correspond bien aux relations (4.1-2) et à l'exemple de la figure 4.2a.

Considérons maintenant $F_i(z)$ et $F_{i+N/2}(z)$, où $N=2^p$ et i=0..N/2-1. En fait, il est aisé de vérifier que $F_i(z)$ peut s'écrire:

$$F_i(z) = H(z) \cdot F_i'(z^2) \qquad i = 0..N/2-1 \tag{4.14}$$

et similairement, $F_{i+N/2}(z)$ est égal à:

$$F_{i+N/2}(z) = H(-z) \cdot F_i'(z^2) \qquad i = 0..N/2-1 \tag{4.15}$$

où $F_i'(z^2)$ est une fonction de z^2 uniquement, et ceci en raison du sous-échantillonnage par un facteur 2 qui suit le premier filtre de l'arbre. En utilisant (4.14) et (4.15), on obtient:

$$F_{i+N/2}(z) = F_i(-z) \tag{4.16}$$

Cette propriété a déjà été remarquée dans [gal83,gal84]. Nous allons donc évaluer le banc de N filtres comme N filtres de longueur moitié, puis obtenir les signaux de sortie par des sommes et différences. De surcroît, les filtres réduits sont sous-échantillonnés d'un facteur $N/2=2^{p-1}$ à la sortie.

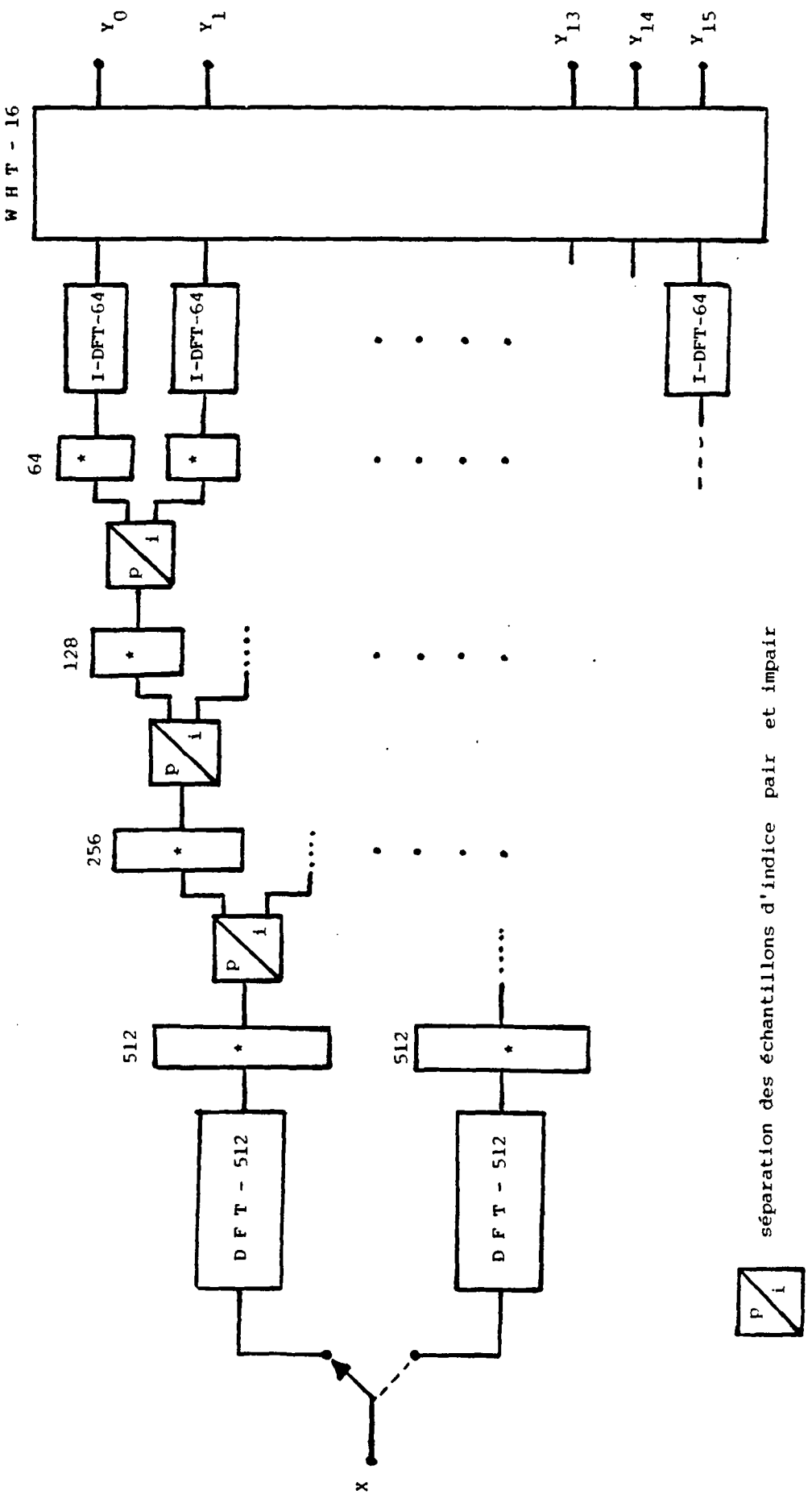

Figure 4.4 Evaluation d'un arbre de filtres QMF dans le domaine transformé

Si le banc de filtres est évalué par transformation de Fourier, nous pouvons tirer parti de ce sous-échantillonnage de sortie. Admettons que dans une transformation de Fourier inverse de longueur M=K· N, on ne désire que chaque N-ième échantillon. Alors, la DFT inverse peut s'écrire comme:

$$x(nN) = \sum_{m=0}^{M-1} X(m) \cdot W_M^{-nmN} = \sum_{m=0}^{K-1} \left(\sum_{k=0}^{N-1} X(m+kK) \right) \cdot W_K^{-nm} \qquad (4.17)$$

La relation (4.17) équivaut à une DFT inverse de longueur k=M/N ainsi que K(N-1) additions.

En évaluant le banc de filtres en parallèle et à l'aide de transformations, il faut donc (où $N=2^p$):

- prendre deux DFT de longueur $M_f/2$ (sur les échantillons d'indice pair et impair)
- calculer 2^p convolutions de longueur $M_f/2$ dans le domaine *transformé*, ce qui revient à $2^p \cdot 3M_f/4$ multiplications et additions
- calculer 2^p DFT inverses de longueur $M_f/2$ mais dont la sortie est sous-échantillonnée d'un facteur 2^{p-1}, c'est-à-dire 2^p DFT inverses de longueur $M_f/2$ et $M_f(2^{p-1}-1)$ additions
- à la sortie, calculer les sommes et différences afin d'obtenir les sorties des filtres $F_i(z)$ et $F_{i+N/2}(z)$, c'est-à-dire M_f additions

En tout, nous obtenons la charge de calcul suivante:

$$2\ \mathrm{DFT}(M_f/2) + 2^p\ \mathrm{DFT}(M_f/(2^p)) \qquad (4.18a)$$

$$3 \cdot 2^{p-2} \cdot M_f/2 \text{ multiplications} \qquad (4.18b)$$

$$5 \cdot 2^{p-2} \cdot M_f/2 \text{ additions} \qquad (4.18c)$$

En comparant (4.11) et (4.18), nous remarquons qu'à part les DFT qui sont les mêmes, nous avons un ordre de grandeur de $p \cdot M_f$ en (4.11) mais de $2^p \cdot M_f$ en (4.18). Ceci vient du fait que dans l'approche répartie, nous avons évalué environ un filtre par étage (et ceci au prix d'une structure complexe), alors que dans l'approche parallèle, environ $N=2^p$ filtres sont calculés.

Reprenons l'exemple qui suit la relation (4.11), mais cette fois-ci en utilisant l'approche parallèle. Outre les DFT, nous avons maintenant 12288 multiplications et 20480 additions correspondant à (4.18b) et (4.18c) respectivement. Ceci donne lieu à la charge de calcul suivante par échantillon d'entrée:

32.13 multiplications et 73.00 additions (4.19)

Quoique ce chiffre soit supérieur à celui donné en (4.13), il reste intéressant en comparaison avec celui de (4.12) pour la méthode directe.

Dans cette discussion des implantations par transformation de Fourier, nous avons omis de parler de leur limitation majeure: le délai associé avec ces méthodes est relativement élevé, puisqu'il faut attendre un grand nombre d'échantillons avant de commencer les calculs (en tous cas si on veut tirer parti de l'efficacité de ces méthodes). Dans les applications en temps réel, ce délai peut présenter une contre-indication majeure pour les méthodes par transformée, mais il n'est par contre d'aucune conséquence dans des applications en temps différé.

Finalement, notons que l'amélioration obtenue avec les méthodes par transformée est moins grande dans le cas des arbres de filtres que ce que l'on pouvait attendre au vu des améliorations obtenues dans le filtrage classique (pour un seul filtre sans sous-échantillonnage, le facteur de gain serait de l'ordre de 25 dans l'exemple que nous avons considéré). Intrinsèquement, l'évaluation d'un banc de filtres par arbre de filtres est donc efficace, même sans utiliser les méthodes par transformée. Ceci est dû au fait que dans un arbre de filtres, une partie des calculs est effectuée simultanément pour plusieurs filtres (mise en commun de zéros).

4.2 Bancs de N filtres

Après une rapide prise en considération du cas de filtres généraux, nous concentrerons notre attention sur les bancs de filtres modulés. Dans ce cas, 3 approches sont explorées plus en détail: la méthode classique séparant le banc de filtres en filtres réduits polyphases suivi d'une FFT [bel74], la méthode proposée par Nussbaumer [nus83] évaluant les filtres polyphases par transformation, et finalement une méthode réalisant tous les calculs dans le domaine Fourier.

4.2.1 Filtres généraux

Ce cas est similaire à l'évaluation en parallèle d'un arbre de filtre (sections 4.1.2 et 4.1.3). Admettons que les sorties soient sous-échantillonnées par un facteur N. Si les filtres sont RIF, la complexité du banc est de l'ordre de la complexité d'un seul filtre mais sans sous-échantillonnage. Si les filtres sont RII, nous remplaçons chaque filtre par un filtre équivalent:

$$H(z) = \frac{N(z)}{D(z)} = \frac{N(z)\cdot D'(z)}{D(z^N)} \qquad (4.20a,b)$$

où

$$D'(z) = \frac{D(z^N)}{D(z)} \qquad (4.20c)$$

Dès lors, le dénominateur en z^N peut être évalué à la fréquence d'échantillonnage réduite. Une évaluation directe de H(z) comme en (4.20a) (en appelant M_N et M_D l'ordre du numérateur et du dénominateur respectivement) requiert de l'ordre de:

$$M_N + M_D \text{ multiplications et additions} \qquad (4.21a)$$

Une évaluation de H(z) en cascade comme en (4.20b) permet de sous-échantillonner le numérateur, (qui est de l'ordre de $M_N+(N-1)\cdot M_D-N+1$) et de calculer le dénominateur après le sous-échantillonnage. Il en résulte un ordre de:

$$M_N/N + M_D \text{ multiplications et additions} \qquad (4.21b)$$

Si le degré du numérateur est nettement plus élevé que celui du dénominateur, (4.21b) peut correspondre à un gain élevé, et même dans le cas où $M_N=M_D$, le gain est de l'ordre de 25% à 50% (pour N allant de 2 à un nombre très grand).

Nous verrons plus loin que la mise en cascade est très intéressante si le dénominateur peut être partagé entre plusieurs filtres. Dans le cas d'un seul filtre, des considérations sur le conditionnement numérique ou le nombre de délais peuvent motiver d'autres architectures [crc83]. Si le filtre est un passe-bande suffisamment

sélectif, il est également possible (si N n'est pas un nombre premier) de réaliser le sous-échantillonnage par étapes, donc également le filtrage, et ceci peut conduire à des gains considérables. Pourtant, ces techniques ne sont pas spécifiques aux bancs de filtres, et nous ne les détaillerons donc pas ici.

4.2.2 Filtres modulés

Afin de simplifier le présentation et ceci dans le but de clarifier les comparaisons entre différentes méthodes, nous admettons dans la suite que les filtres sont complexes et modulés de la façon suivante (voir la relation (2.97)):

$$H_i(z) = H_p(W^i z) \tag{4.22}$$

Similairement, les signaux sont supposés également complexes. Bien évidemment, la plupart des applications font appel à des filtres réels (par exemple les bancs pseudo-QMF) et les résultats peuvent êtres adaptés au cas réel à l'aide de modifications adaptées.

a) Evaluation directe par filtres polyphases et FFT

Cette méthode, introduite par Bellanger [bel74], est aujourd'hui classique. Nous la rappellerons ici simplement pour mémoire. Admettons que le filtre prototype ait un dénominateur de degré M_D-1 et un numérateur de degré M_N-1. Il peut être représenté, avec (4.20), par:

$$H_p(z) = \frac{N'(z)}{D(z^N)} \tag{4.23}$$

où N'(z) est de degré M_N-1+(N-1)·(M_D-1). Comme le dénominateur est maintenant fonction de z^N, il est commun à tous les filtres $H_i(z)$. Appelons $N_i(z^N)$ la i-ième composante polyphase de N'(z):

$$N'(z) = \sum_{i=0}^{N-1} z^{-i} N_i(z^N) \tag{4.24}$$

Avec (4.22-24), il s'ensuit que le vecteur de filtres $\mathbf{h}(z)=[H_0(z), H_1(z), \ldots H_{N-1}(z)]^T$ est égal à:

$$\mathbf{h}(z) = 1/D(z^N) \cdot \mathbf{F} \cdot [N_0(z^N), z^{-1} \cdot N_1(z^N), \ldots, z^{-N+1} \cdot N_{N-1}(z^N)]^T \tag{4.25}$$

Remarquons que, même en l'absence d'un algorithme efficace pour la transformée rapide, la formulation (4.25) réduit le calcul de N filtres à celui de N filtres réduits (correspondant environ à un seul filtre) et d'une multiplication matricielle scalaire de dimension NxN.

Considérons le cas où les sorties du banc sont sous-échantillonnées d'un facteur N. Pour chaque nouvel échantillon, nous devons évaluer la sortie de $1/D(z^N)$ (donc M_D-1 opérations) ainsi qu'un des filtres polyphases, et après N échantillons, il faut calculer une DFT de longueur N. Il faut donc en moyenne, pour chaque échantillon d'entrée (où il est admis qu'un filtre avec degré p et q au numérateur et au dénominateur requiert p+q+1 multiplications et p+q additions):

$$M_N/N + (2-1/N)\cdot(M_D-1) \text{ multiplications complexes} \tag{4.26a}$$

$$M_N/N + (2-1/N)\cdot(M_D-1) - 1 \text{ additions complexes} \tag{4.26b}$$

$$1/N \text{ DFT de longueur } N \tag{4.26c}$$

Le cas RIF correspond à M_D=1. Dans le cas de bancs de filtres réels, la fonction de modulation est en général sinusoïdale (voir section 3.3.2), et la décomposition ci-dessus donne lieu, mis à part les filtres polyphases, à une transformation du type cosinus (voir par exemple [nar79]). Des exemples concrets peuvent être trouvés dans [nus84a,nus84b,mas85], et nous avons également considéré le cas bidimensionnel dans [vet84a].

b) Evaluation des filtres polyphases par FFT

La charge de calcul due aux filtres polyphases peut être dominante, surtout si le filtre prototype est très long par rapport au nombre de canaux. Nussbaumer [nus83] a suggéré une évaluation des filtres polyphases dans le domaine de Fourier, avec l'avantage que les transformées inverses peuvent être combinées avec la tranformation de Fourier qui suit les filtres polyphases. Ceci donne lieu à une transformation bidimensionnelle qui peut être efficacement calculée à l'aide de tranformations polynomiales [nus82].

Nous esquissons l'idée générale ci-après, et ceci, pour plus de clarté, à l'aide de bancs complexes uniquement. Plus de détails peuvent être trouvés dans [nus83], en particulier la spécialisation au cas de bancs réels. Admettons que les filtres polyphases soient évalués à l'aide d'une convolution circulaire de longueur K (K suffisant pour permettre la méthode de "overlap and add"). Si cette convolution est calculée par transformation de Fourier, le calcul du banc de filtre en (4.25) se fera de la façon suivante:

- N DFT de longueur K en entrée
- N convolutions de longueur K dans le domaine tranformé (à l'aide d'une multiplication complexe par point)
- N DFT inverses de longueur K produisant N·K résultats du banc de filtres polyphases
- K DFT de longueur N, correspondant à la DFT qui suit le banc polyphase dans

(4.25).

Evidemment, les N IDFT de longueur K suivies des K DFT de longueur N correspondent à une DFT bidimensionnelle de longueur KxN.

Prenons le cas où tous les filtres polyphases ont la même longueur M_p. Alors, pour $M_s=N(K-M_p+1)$ échantillons d'entrée complexes, la méthode présentée requiert (les filtres sont également complexes):

N DFT de longueur K en entrée (4.27a)

$N \cdot K$ multiplications complexes (4.27b)

une DFT de dimension KxN en sortie (4.27c)

En dépit de cette complexité de calcul attrayante, cette méthode présente l'inconvénient d'introduire un délai relativement long, puisqu'il faut attendre M_s échantillons avant même de pouvoir commencer les calculs.

c) Evaluation complète dans le domaine de Fourier

Nous allons considérer le cas où on commence par prendre une transformation de Fourier du signal d'entrée, puis on calcule les N convolutions dans le domaine transformé avant de prendre N transformations inverses pour retrouver les sorties du banc de filtres. En raison de la nature modulée du banc de filtres, nous allons montrer que les convolutions dans le domaine tranformé peuvent être évaluées efficacement.

Quoique cet algorithme n'améliore pas la complexité de calcul par rapport à la méthode précédente, il permet un éclairage différent du problème. Le fait qu'il donne un nombre d'opérations similaire montre la cohérence des deux approches.

Admettons que la DFT d'entrée soit de longueur $M=K \cdot N$. Appelons $Y_i(k)$ le i-ième signal après filtrage et sous-échantillonnage dans le domaine tranformé. En (4.17), nous avons vu que le sous-échantillonnage dans le domaine transformé correspond à une somme de N termes espacés de K. Donc, $Y_i(k)$ peut s'écrire comme:

$$Y_i(k) = \sum_{l=0}^{N-1} X(k+l \cdot K) \cdot H_i(k+l \cdot K) \qquad k=0..K-1 \tag{4.28}$$

Les indices entre parenthèses (par exemple $k+l \cdot K$) sont toujours pris modulo M dans la suite. Maintenant, notons que, en vertu de (4.22):

$$H_i(k+l\cdot K) = H_p(k+(l-i)\cdot K) \tag{4.29}$$

Donc, $[Y_0(k), Y_1(k), \ldots Y_{N-1}(k)]$ peuvent être obtenus par une convolution circulaire de N points à partir de $[X(k), X(k+K), \ldots X(k+(N-1)K]$. De ce fait, nous pouvons remplacer les N filtres suivis de sous-échantillonnage par K convolutions circulaires de longueur N. Celles-ci peuvent être évaluées par tranformation, conduisant ainsi à $N\cdot K$ multiplications complexes et à K DFT de longueur N avant et après ces multiplications. Notons que nous pouvons utiliser des DFT ou des DFT inverses selon notre gré, puisqu'une simple permutation des échantillons par la matrice J (voir la relation (2.48)) permet de passer de l'une à l'autre.

Admettons maintenant que la DFT d'entrée de longueur $M=K\cdot N$ soit calculée en deux étages, le premier de N DFT de longueur K et le second de K DFT de longueur N, avec entre les deux les multiplications par les facteurs de phases ("twiddle factors") adaptés [nus82]. Dès lors, les K DFT de longueur N disparaissent en raison des DFT de longueur N qui suivent immédiatement, et les multiplications par les facteurs de phases peuvent être inclues avec les multiplications complexes nécessaires au filtrage. En tout et pour tout, il faut donc calculer:

N DFT de longueur K en entrée (4.30a)

M multiplications complexes (4.30b)

une DFT de dimension NxK en sortie (4.30c)

Ceci correspond bien à la même complexité que celle donnée en (4.27). Notons toutefois que maintenant, le nombre d'échantillons d'entrée est égal à $M-M_f+1$ où M_f est la longueur du filtre prototype. Ce nombre est plus petit que celui correspondant à (4.27), et cela de (N-1) lorsque M_f est un multiple de N.

La méthode est illustrée schématiquement dans la figure 4.5 pour le cas de N=4 et M=32.

Quoique fort similaires, les deux méthodes décrites ne sont pas identiques, puisque la première fait appel à une DFT de dimension (KxN) et la seconde à une de dimension (NxK). D'un point de vue de calcul, ces deux DFT sont identiques, mais l'interprétation ne l'est pas. D'autre part, le nombre d'échantillons d'entrée pour une dimension de DFT donnée n'est pas le même (il est légèrement plus élevé dans la première méthode). Pour illustrer le potentiel de ces méthodes relativement complexes, nous prenons un exemple simple: évaluons par blocs de 256 échantillons d'entrée un banc de 16 filtres RIF de longueur 128 (et dont les sorties sont

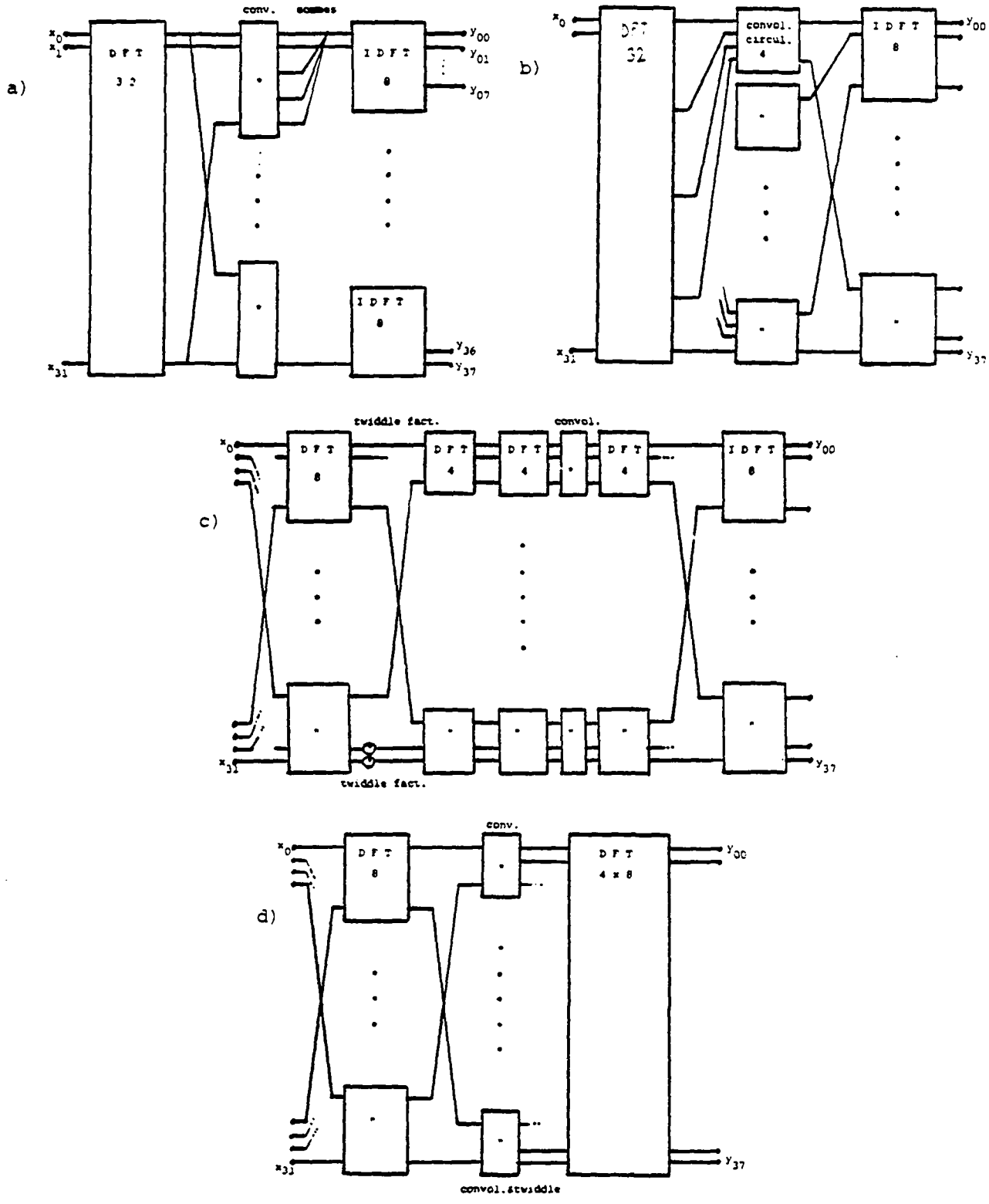

Figure 4.5: Banc de 4 filtres sous-échantillonnés d'un facteur 4 et évalué par tranformation de Fourier de longueur 32

a) DFT 32, puis 4 convolutions dans le domaine transformé et 4 IDFT de 8

b) Evaluation des convolutions dans le domaine transformé par convolutions circulaires de longueur 4

c) Evaluation de la DFT de longueur 32 par 4 DFT de 8, des "twiddle factors" puis 8 DFT de 4. Evaluation des convolutions circulaires par transformée

d) Après toutes les simplifications, il reste 4 DFT de 8 à l'entrée, 32 multiplications complexes (comprenant la convolutions et les "twiddle factors") et une DFT de 4x8 en sortie

sous-échantillonnées par un facteur 16 également). Pour des signaux ainsi que des filtres complexes, on obtient les nombres d'opérations suivants:

Méthode a) 25.25 multiplications et 47.25 additions par échantillon d'entrée
Méthode b) 10.55 multiplications et 53.22 additions par échantillon d'entrée
Méthode c) 11.78 multiplications et 59.41 additions par échantillon d'entrée

Le nombre de multiplications est donc réduit d'un facteur supérieur à 2, tout en gardant un nombre comparable d'additions.

4.3 Principaux résultats du chapitre

La complexité de calcul liée aux bancs de filtres a été considérée. Deux caractéristiques importantes des bancs de filtres ont été mises à contribution afin de diminuer la complexité de leur implantation: a) le fait que les sorties soient sous-échantillonnées et b) le fait que les filtres d'un banc soient liés par modulation.

Une première partie a été consacrée aux arbres de filtres, en particulier aux arbres QMF. Après un rappel des méthodes classiques d'évaluation dans le domaine temporel, nous avons proposé des méthodes par convolution dans le domaine de Fourier, mais en tenant compte des symétries et sous-échantillonnages propres aux bancs QMF.

La deuxième partie du chapitre a présenté la complexité des bancs de N filtres, en particulier dans le cas où ceux-ci sont modulés. Outre la méthode classique des bancs polyphases suivis d'une FFT [bel74], on a développé deux méthodes par transformée, l'une due à Nussbaumer [nus83] et l'autre originale à notre connaissance. Ces deux méthodes donnent lieu à des complexités de calcul presque identiques et sensiblement réduites par rapport à l'approche classique.

En conclusion, nous avons montré que les bancs de filtres peuvent en général être implantés très efficacement, car la complexité de calcul est bien moindre que celle de N filtres généraux sans sous-échantillonnage. L'importance des transformations rapides (Fourier, cosinus) est également apparue clairement, et c'est la raison pour laquelle nous leur consacrons le chapitre suivant.

Références bibliographiques du chapitre 4:

[bea75] K.G.Beauchamp, **Walsh Functions and Their Applications**, Academic Press, London, 1975.

[bel74] M.G.Bellanger, and J.L.Daguet, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT", IEEE Trans. on Communications, Vol. COM-22, No.9, pp. 1199-1204, Sept. 1974.

[bel82] M.G.Bellanger, "On Computational Complexity in Digital Transmultiplexer Filters", IEEE Trans. on Communications, Vol.30, No.7, July 1982, pp.1461-1465.

[crc83] R.E.Crochiere, and L.R.Rabiner, **Multirate Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1983.

[est77b] D.Esteban, and C.Galand, "Direct Approach to Quasi Perfect Decomposition of Speech in Sub-bands", Int. Congress on Acoustics, Madrid, July 1977, pp.852.

[gal83] C.Galand, "Codage en Sous-Bandes: Théorie et Application à la Compression Numérique du Signal de Parole", Thèse d'Etat, Université de Nice, 1983.

[gal84] C.R.Galand, and H.J.Nussbaumer, "New Quadrature Mirror Filter Structures", IEEE Trans. on Acoust., Speech and Signal Proc., Vol.32, No.3, June 1984, pp.522-531.

[gal85] C.R.Galand, and H.J.Nussbaumer, "Quadrature Mirror Filters with Perfect Reconstruction and Reduced Computational Complexity" Proc. 1985 Int. IEEE Conf. on ASSP, pp.525-529, Tampa, March 1985.

[mas85] J.Masson, and Z.Picel, "Flexible Design of Computationaly Efficient Nearly Perfect QMF Filter Banks", Proc. 1985 IEEE Conf. on ASSP, Tampa, March 1985.

[nar79] M.J.Narasimha, and A.M.Peterson, "Design of a 24-Channel Transmultiplexer", IEEE Trans. on ASSP, Vol. ASSP-27, No.6, pp. 752-762, Dec.1979.

[nus81] H.J.Nussbaumer, "Pseudo QMF Filter Bank", IBM Technical Disclosure Bulletin, Vol.24, No.6, pp 3081-3087, Nov.1981.

[nus82] H.J.Nussbaumer, **Fast Fourier Transform and Convolution Algorithms**, Springer, Berlin, 1982.

[nus83c] H.J.Nussbaumer, "Polynomial Transform Implementation of Digital Filter Banks", IEEE Trans. on ASSP, Vol.31, No.3, June 1983, pp.616-622.

[nus84a] H.J.Nussbaumer, and M.Vetterli, "Computationally Efficient QMF Filter Banks", Proc. 1984 Int. IEEE Conf. on ASSP, San Diego, March 1984.

[nus84b] H.J.Nussbaumer, and M.Vetterli, "Pseudo Quadrature Mirror Filters", Proc. of Int. Conf. on Digital Signal Processing, Florence, Sept. 1984.

[smi84] M.J.T.Smith, T.P.Barnwell, "A procedure for designing exact reconstruction filterbanks for tree structured sub-band coders", Proc. IEEE ICASSP-84, San Diego, March 1984.

[vet83] M.Vetterli, "Tree Structures for Orthogonal Transforms and Application to the Hadamard Transform", Signal Processing, Vol.5 , No.6, pp. 473-484, Nov.1983.

[vet84a] M.Vetterli, "Multi-Dimensional Sub-Band Coding: Some Theory and Algorithms", Signal Processing, Vol. 6, No.2, pp. 97-112, Feb. 1984.

5 Transformées rapides

". . . diviser chacune des difficultés à examiner en autant de parcelles qu'il se pourrait et qu'il serait requis pour les mieux résoudre."

R. Descartes, "Le discours de la méthode"

Les algorithmes rapides pour les transformations du genre Fourier jouent un rôle capital dans le traitement numérique du signal. La redécouverte de l'algorithme de transformation de Fourier rapide (FFT) en 1965 [coo65] a d'ailleurs notablement contribué au développement du traitement numérique des signaux, puisqu'avec cet algorithme il devenait possible d'évaluer efficacement les spectres de signaux échantillonnés, mais également le filtrage numérique grâce à la propriété de convolution de la transformée de Fourier.

Dans les chapitres précédents, en particulier le chapitre 4 sur la complexité de calcul liée aux bancs de filtres, il a été montré que les transformations du type Fourier permettent également de réduire sensiblement l'effort de calcul nécessaire à l'implantation de certains types de bancs de filtres (où les différents filtres sont obtenus par modulation à partir d'un seul filtre prototype). Comme ces types de bancs de filtres représentent une part importante de ceux utilisés en pratique, entre autres justement en raison de leur mise en oeuvre plus efficace, il semble approprié d'étudier plus en détails les transformations utilisées dans ces bancs de filtres modulés.

L'origine des résultats nouveaux présentés dans ce chapitre a été l'étude de l'implantation efficace des bancs de filtres pseudo-QMF (voir la section 3.3 et les références [nus81,nus84a,nus84b]). Mais comme d'une part, il a été possible de généraliser les résultats obtenus à plusieurs autres problèmes, et que certaines des transformations étudiées ont également une application immédiate dans le codage (un domaine relativement éloigné des bancs de filtres), nous avons jugé bon de consacrer un chapitre séparé et relativement indépendant à la présentation de ces développements.

Remarquons que la plupart des résultats concerne d'abord des transformations de séquences dont la longueur est une puissance de 2. Ces transformations ont en général une complexité multiplicative plus élevée que celles qui portent sur des séquences de longueur comparable mais égale à un produit de facteurs premiers entre eux. Elles ont néanmoins plusieurs avantages qui les rendent très attrayantes:
i) un nombre total d'opérations faible (comparable avec celui des meilleures versions de l'algorithme de Winograd)
ii) une structure relativement simple
iii) un temps de passage sur processeur en général inférieur à celui d'algorithmes très complexes (Winograd) en raison de la structure plus simple
iv) une longueur adaptée aux contingences d'implantations réelles (dans lesquelles l'espace d'adressage ou le nombre de registres sont presque exclusivement une puissance de 2).

En plus de ces avantages communs à toutes les transformations de longueur en puissance de 2, il sera possible de généraliser l'algorithme qui est proposé dans la suite à toute une série d'autres problèmes, démontrant ainsi la souplesse de l'approche proposée.

Le chapitre se divise de la façon suivante. Une première section présente l'algorihme de base (appelé FFCT pour "Fast Fourier-Cosine Transform"), évalue sa complexité de calcul pratique et théorique, et le compare à d'autres algorithmes connus. La seconde partie s'applique à généraliser cet algorithme à plusieurs autres cas, entre autres pour transformer des signaux comportant certaines symétries, pour la convolution circulaire de signaux réels, et finalement pour le cas bidimensionnel. Ensuite, une section est consacrée à une implantation matérielle qui a pu être réalisée et une dernière section présente l'implantation logicielle de l'algorithme. Tant pour l'implantation matérielle que logicielle, nous avons utilisé une méthodologie et des outils de développement en partie originaux, et la présentation de cette approche devrait donner une certaine généralité à ces exemples particuliers d'implantation.

Notons que notre présentation est relativement succincte, car les résultats présentés sont déjà documentés par toute une série de publications détaillées [vet84b,vet85a,vet85b,vet86a,vet85d,vet85e,duh86].

5.1 L'algorithme de FFCT

L'algorithme de base met en relation deux transformations distinctes, celle de Fourier et celle en cosinus. Le fait que la transformation en cosinus peut s'évaluer à l'aide

d'une transformation de Fourier (de même dimension) est un résultat connu, et une méthode efficace pour le faire a été proposée dans [nar78]. La relation duale établissant que la transformation de Fourier peut être évaluée à l'aide de transformations en cosinus (de plus petites dimensions) est un résultat original à la base de l'algorithme de FFCT.

Outre le fait que ce résultat permet d'obtenir un algorithme très efficace pour le calcul de transformée de Fourier ou en cosinus de séquences dont la longueur est une puissance de 2, il établit une relation étroite entre ces deux transformations. Ceci est d'importance tant d'un point de vue théorique que pratique. La mise en relation de deux problèmes distincts permet de faire profiter l'un des avantages acquis pour la solution de l'autre, et réciproquement. Ensuite, la décomposition proposée permettra de démontrer certains résultats sur la complexité théorique des transformations considérées. Finalement, des résultats d'implantation de l'une des transformations peuvent être utilisées pour l'implantation de l'autre.

Avant d'entrer dans les détails de l'algorithme de FFCT, remarquons qu'en parallèle avec nos propres développements, un autre algorithme a été proposé (algorithme du "split radix" [duh84a]) qui, quoique non-isomorphe, utilise également une division asymétrique du problème original en problèmes de dimension moitié et un quart. Nous verrons par la suite l'importance de cette division asymétrique qui fait l'originalité de ces algorithmes, puisque les algorithmes classiques se basent toujours sur une division symétrique (par exemple en deux ou quatre sous-problèmes de même dimension chacun).

5.1.1 Dérivation de l'algorithme

Introduisons les définitions suivantes, où x(n) est le n-ième élément d'un vecteur réel **x** de longueur N (N étant un multiple de 4 voire une puissance de 2). Nous utilisons la convention suivante: nom_de_transformation(k,N,**x**) désigne le k-ième élément de sortie de la transformée du vecteur **x** de longueur N.

i) Transformation de Fourier discrète ("discrete Fourier transform", DFT):

$$DFT(k,N,\mathbf{x}) := \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi nk/N} \qquad k = 0 \ldots N-1, \quad j = \sqrt{-1} \qquad (5.1)$$

ii) Transformation en cosinus discrète ("discrete cosine transform", DCT):

$$DCT(k,N,\mathbf{x}) := \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{2\pi(2n+1)k}{4N}\right) \qquad k = 0 \ldots N-1 \tag{5.2}$$

iii) Transformation de Fourier-cosinus ("discrete cosine Fourier transform", cos-DFT):

$$\text{cos-DFT}(k,N,\mathbf{x}) := \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{2\pi nk}{N}\right) \qquad k = 0 \ldots N-1 \tag{5.3}$$

iv) Transformation de Fourier-sinus ("discrete sine Fourier transform", sin-DFT):

$$\text{sin-DFT}(k,N,\mathbf{x}) := \sum_{n=0}^{N-1} x(n) \cdot \sin\left(\frac{2\pi nk}{N}\right) \qquad k = 0 \ldots N-1 \tag{5.4}$$

Remarquons que la transformation de Fourier-cosinus (5.3) et la transformation de Fourier-sinus (5.4) sont respectivement la partie réelle et la partie imaginaire (avec un changement de signe) de la transformation de Fourier (5.1) lorsque $\mathbf{x}$ est réel:

$$DFT(k,N,\mathbf{x}) = \text{cos-DFT}(k,N,\mathbf{x}) - j \cdot \text{sin-DFT}(k,N,\mathbf{x}) \tag{5.5}$$

Remarquons également les symétries suivantes:

$$\text{cos-DFT}(N-k,N,\mathbf{x}) = \text{cos-DFT}(k,N,\mathbf{x}) \tag{5.6}$$

et

$$\text{sin-DFT}(N-k,N,\mathbf{x}) = - \text{sin-DFT}(k,N,\mathbf{x}) \tag{5.7}$$

ainsi que le fait suivant:

$$\text{sin-DFT}(0,N,\mathbf{x}) = 0 \tag{5.8a}$$

$$\text{sin-DFT}(N/2,N,\mathbf{x}) = 0 \tag{5.8b}$$

Des relations (5.5-5.8) se déduisent, si $\mathbf{x}$ est un vecteur réel, les propriétés bien connues de la transformée de Fourier d'un signal réel:

$$[\ DFT(N-k,N,x)\ ] = [\ DFT(k,N,x)\ ]^{*} \tag{5.9a}$$

$$Im[\ DFT(0,N,x)\ ] = 0 \tag{5.9b}$$

$$Im[\ DFT(N/2,N,x)\ ] = 0 \tag{5.9c}$$

Notons aussi que dans la définition de la transformation en cosinus, nous avons omis le facteur de normalisation $1/\sqrt{2}$ pour k=0 [ahm74], et ceci afin de simplifier la présentation qui va suivre. Quoique habituellement les transformations définies par les relations (5.1-4) ne sont évaluées que pour des valeurs de k comprises entre 0 et N-1, nous verrons dans la suite que des valeurs n'appartenant pas à cet ensemble peuvent être utiles dans le cas de la DCT. Les relations suivantes se vérifient aisément:

$$DCT(N,N,x) = 0 \tag{5.10a}$$

$$DCT(-k,N,x) = DCT(k,N,x) \tag{5.10b}$$

$$DCT(2N-k,N,x) = -\ DCT(k,N,x) \tag{5.10c}$$

Afin de dériver l'algorithme de FFCT, nous allons considérer séparément l'évaluation de la partie réelle et de la partie imaginaire de la transformation de Fourier (ce qui est évident si x est réel, et nécessite quelques additions de sortie si x est complexe comme nous le verrons plus loin). Ensuite, contrairement à ce qui est réalisé dans les algorithmes classiques (algorithmes à base 2 ou 4 par exemple), nous n'allons pas diviser le problème de départ en sous-problèmes de grandeurs égales mais plutôt en sous-problèmes de grandeurs adaptées.

Commençons par dériver une telle partition pour le calcul de la partie réelle de la transformation de Fourier. En utilisant le fait que la fonction cosinus est une fonction paire et périodique, nous pouvons récrire (5.3) comme:

$$\begin{aligned} \text{cos-DFT}(k,N,x) = & \sum_{n=0}^{N/2-1} x(2n) \cdot \cos\left(\frac{2\pi nk}{N/2}\right) + \\ & \sum_{n=0}^{N/4-1} (x(2n+1)+x(N-2n-1)) \cdot \cos\left(\frac{2\pi(2n+1)k}{4 \cdot N/4}\right) \end{aligned} \tag{5.11}$$

Remarquons que la première somme en (5.11) est simplement une transformation de

Fourier-cosinus de longueur N/2, et que la seconde somme est une transformation en cosinus de longueur N/4. Donc, (5.11) peut être reformulé de la façon suivante, où les relations (5.6) et (5.10c) sont mises à contribution lorsque nécessaire:

$$\text{cos-DFT}(k,N,x) = \text{cos-DFT}(k,N/2,x_1) + \text{DCT}(k,N/4,x_2) \qquad k=0\,..\,N-1 \qquad (5.12a)$$

$$\text{avec } x_1(n) = x(2n) \qquad k=0\,..\,N/2-1 \qquad (5.12b)$$

$$x_2(n) = x(2n+1) + x(N-2n-1) \qquad k=0\,..\,N/4-1 \qquad (5.12c)$$

Une transformation de Fourier-cosinus de dimension N a donc été été réduite à une transformation de Fourier-cosinus de longueur N/2 et une transformation en cosinus de longueur N/4, et ceci au prix de N/4 additions d'entrée et de N/2 additions de sortie (cos-DFT(k,N,x) doit bien être évalué pour k=0 jusqu'à N/2, mais k=N/4 ne requiert pas d'addition de sortie en vertu de (5.10a)). Evidemment, si N est une puissance de 2, nous pouvons réitérer la réduction de la partie cos-DFT dans (5.12a), et ceci jusqu'à obtenir une transformation triviale (N=2). Le cas de la DCT sera traité ultérieurement.

Penchons-nous maintenant sur le calcul de la partie imaginaire de la transformation de Fourier. En utilisant le fait que la fonction sinus est périodique mais impaire, nous pouvons organiser l'évaluation de (5.4) de la façon suivante:

$$\text{sin-DFT}(k,N,x) = \sum_{n=0}^{N/2-1} x(2n) \cdot \sin\left(\frac{2\pi nk}{N/2}\right) + \sum_{n=0}^{N/4-1} (x(2n+1)-x(N-2n-1)) \cdot \sin\left(\frac{2\pi(2n+1)k}{4\cdot N/4}\right) \qquad (5.13)$$

Utilisons l'identité trigonométrique suivante:

$$\sin\left(\frac{2\pi(2n+1)k}{N}\right) = (-1)^n \cdot \cos\left(\frac{2\pi(2n+1)(N/4-k)}{N}\right) \qquad (5.14)$$

et dès lors, la relation (5.13) peut s'écrire comme, (à l'aide de (5.10b) ou (5.7) si nécessaire):

$$\text{sin-DFT}(k,N,x) = \text{sin-DFT}(k,N/2,x_1) + \text{DCT}(N/4-k,N/4,x_3) \qquad k=0\,..\,N-1 \qquad (5.15a)$$

$$\text{avec } x_3(n) = (-1)^n \cdot (x(2n+1) - x(N-2n-1)) \qquad k=0\,..\,N/4-1 \qquad (5.15b)$$

On constate donc que l'évaluation de la partie imaginaire de la transformation de Fourier-sinus de dimension N a été réduite à une transformation de Fourier-sinus de longueur N/2 et une transformation en cosinus de longueur N/4, et ceci au prix de N/4 additions d'entrée et N/2-2 additions de sortie (puisqu'il faut évaluer sin-DFT(k,N,x) pour k=1 à N/2-1 en vertu de (5.7-8) et qu'il n'y a pas d'additions de sortie pour k=N/4 selon (5.8a)).

En combinant (5.12) et (5.15), il est clair qu'une transformation de Fourier sur une séquence réelle de longueur N peut se calculer à partir de:
- une transformation de Fourier portant sur une séquence de longueur N/2
- deux transformations en cosinus portant sur des séquences de longueur N/4 obtenues à partir de sommes et différences

Ceci est obtenu au prix de 3N/2-2 additions seulement. Le procédé décrit ci-dessus est illustré dans la figure 5.1 pour le cas N=8.

Il reste donc à trouver une méthode efficace pour évaluer les transformations en cosinus apparaissant dans (5.12a) et (5.15a). Commençons par permuter les N échantillons d'entrée de la transformée en cosinus. En introduisant la séquence auxiliaire x_4, obtenue à partir de x de la façon suivante [nar78]:

$$x_4(n) = x(2n)$$

$$x_4(N-n-1) = x(2n+1) \qquad n = 0\,..\,N/2-1 \qquad (5.16)$$

il est possible de vérifier que (5.2) devient:

$$\text{DCT}(k,N,x) = \sum_{n=0}^{N-1} x_4(n) \cdot \cos\left(\frac{2\pi(4n+1)k}{4N}\right) \qquad k = 0\,...\,N-1 \qquad (5.17)$$

Par simple manipulation trigonométrique, la relation (5.17) s'exprime en fonction de cos-DFT et de sin-DFT de la façon suivante:

$$DCT(k,N,x) = \cos(\frac{2\pi k}{4N}) \cdot \text{cos-DFT}(k,N,x_4) - \sin(\frac{2\pi k}{4N}) \cdot \text{sin-DFT}(k,N,x_4) \qquad (5.18)$$

$$k = 0 \ldots N-1$$

En tirant parti des symétries intrinsèques aux fonctions sinus et cosinus, il est possible d'évaluer simultanément DCT(k,N,x) et DCT(N-k,N,x) à partir de cos-DFT(k,N,x_4) et de sin-DFT(k,N,x_4), et ceci à l'aide du produit matriciel suivant:

$$\begin{bmatrix} DCT(k,N,x) \\ DCT(N-k,N,x) \end{bmatrix} = \begin{bmatrix} \cos(\frac{2\pi k}{4N}) & -\sin(\frac{2\pi k}{4N}) \\ \sin(\frac{2\pi k}{4N}) & \cos(\frac{2\pi k}{4N}) \end{bmatrix} \cdot \begin{bmatrix} \text{cos-DFT}(k,N,x_4) \\ \text{sin-DFT}(k,N,x_4) \end{bmatrix} \qquad (5.19a)$$

$$k = 0 \ldots N/2-1$$

$$DCT(N/2,N,x) = \cos(\pi/4) \cdot \text{cos-DFT}(N/2,N,x_4) \qquad (5.19b)$$

Remarquons que la matrice en (19.a) est une matrice de rotation et non pas une matrice générale. Nous pouvons donc appliquer l'algorithme de multiplication complexe [nus82] qui ne requiert que 3 multiplications et 3 additions réelles (en lieu et place de 4 et 2 respectivement). Cet algorithme est donné ci-dessous, avec les notations correspondant à la relation (5.19a):

$$c_1 := \cos(\frac{2\pi k}{4N}) + \sin(\frac{2\pi k}{4N})$$

$$c_2 := \sin(\frac{2\pi k}{4N}) - \cos(\frac{2\pi k}{4N})$$

$$s_1 = \text{cos-DFT}(k,N,x_4) + \text{sin-DFT}(k,N,x_4)$$

$$m_1 = s_1 \cdot \cos(\frac{2\pi k}{4N})$$

$$m_2 = \text{sin-DFT}(k,N,x_4) \cdot c_1$$

$$m_3 = \text{cos-DFT}(k,N,x_4) \cdot c_2$$

$$DCT(k,N,x) = m_1 - m_2$$

$$DCT(N-k,N,x) = m_1 + m_3 \qquad (5.20)$$

Il est à noter que les valeurs c_1 et c_2 peuvent être précalculées. L'évaluation de (5.19a-b) pour k allant de 0 à N/2 et étant donné les valeurs de cos-DFT et sin-DFT requiert en tout (3N/2)-2 multiplications et (3N/2)-3 additions. Nous venons donc de montrer comment évaluer une DCT d'une séquence réelle à l'aide d'une DFT de même longueur et d'opérations auxiliaires. Cette organisation des calculs est schématiquement illustrée par la figure 5.2 pour N=8.

Ainsi, l'algorithme de FFCT réduit la transformation de Fourier en DFT de plus en plus petites d'une part, et en DCT d'autre part. Ces dernières sont ramenées à l'aide de permutations à des DFT suivies de rotations. Il en résulte qu'une DFT ou une DCT de dimension arbitraire peut être récursivement réduite à des transformations de plus en plus petites, et ceci jusqu'à ce qu'il ne reste que des transformations triviales (de longueur 2 par exemple). La réduction du problème à l'entrée se fait à l'aide d'additions uniquement, et la recombinaison du résultat à la sortie est obtenue à l'aide d'additions et de rotations. La figure 5.3 illustre la méthode sur l'exemple d'une transformation de Fourier de longueur 32.

Finalement, notons que même si l'algorithme de FFCT est d'abord adapté aux transformations sur des séquences réelles, il peut être immédiatement appliqué au cas de séquences complexes également. Par exemple, si la transformée de Fourier d'une séquence complexe est désirée, on calculera séparément la transformée de sa partie réelle et de sa partie imaginaire, et ceci avec l'algorithme de DFT décrit ci-dessus, puis on recombinera le résultat final à l'aide de 2N-4 additions auxiliaires (étant donné qu'il faut deux additions par point de fréquence, sauf pour k=0 et N/2 où le résultat est déjà correct).

5.1.2 Complexité de calcul pratique

La dérivation de l'algorithme de FFCT nécessite une longueur de transformation égale à un multiple de 4. Pourtant, le potentiel de cet algorithme s'exprime vraiment si cette longueur est une puissance de 2, car alors la décomposition proposée peut être réitérée sur les petites transformations, et nous allons évaluer la charge de calcul qui résulte dans ce cas. $O_m[\text{transformation}(N,x_t)]$ et $O_a[[\text{transformation}(N,x_t)]$ représentent respectivement le nombre de multiplications et d'additions utilisées pour

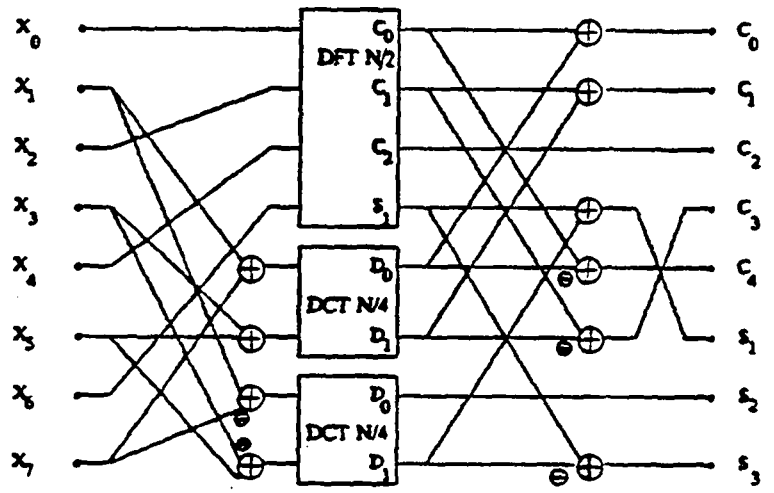

Figure 5.1: Evaluation d'une DFT de longueur N avec une DFT de N/2 et de deux DCT de N/4 (cas N=8).

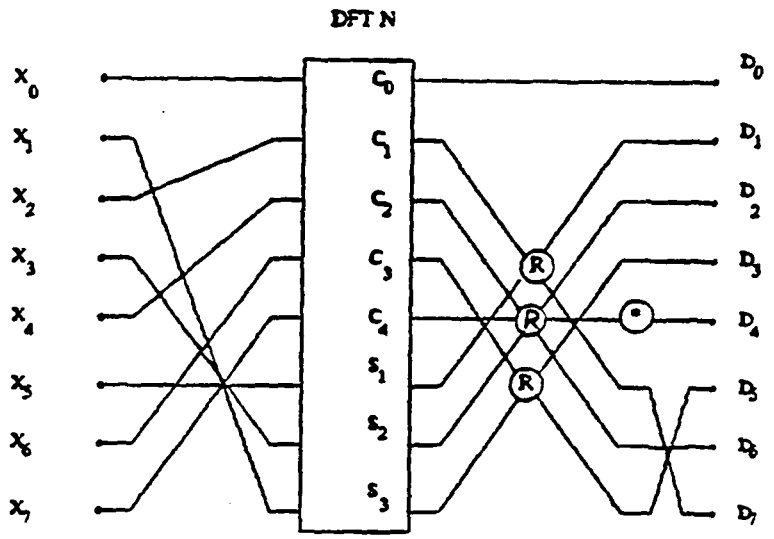

Figure 5.2: Evaluation d'une DCT de longueur N à l'aide d'une permutation, d'une DFT de N et de rotations de sortie pour le cas N=8 (*: multiplication par $1/\sqrt{2}$, R: rotation).

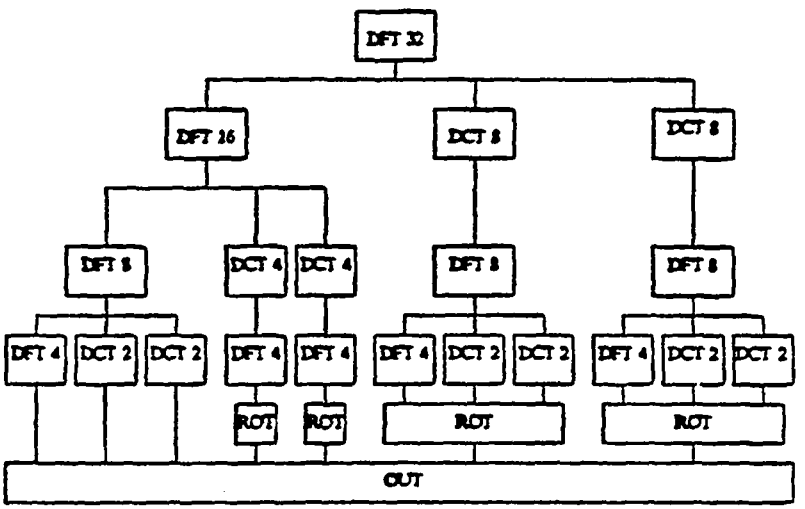

Figure 5.3: Evaluation d'une DFT de longueur 32 par des DFT et DCT de longueurs réduites.

la transformation donnée portant sur un vecteur de longueur N et de type t (r pour réel et c pour complexe par exemple). La complexité de calcul est alors donnée par les relations récursives suivantes.

Pour une transformation de Fourier-cosinus sur une séquence réelle de longueur N, il résulte de (5.12) que:

$$O_m[\text{cos-DFT}(N,x_r)] = O_m[\text{cos-DFT}(N/2,x_r)] + O_m[\text{DCT}(N/4,x_r)] \tag{5.21a}$$

$$O_a[\text{cos-DFT}(N,x_r)] = O_a[\text{cos-DFT}(N/2,x_r)] + O_a[\text{DCT}(N/4,x_r)] + (3N/4) \tag{5.21b}$$

De façon similaire, pour la transformation de Fourier-sinus sur une séquence réelle de longueur N, il résulte de (5.15) que:

$$O_m[\text{sin-DFT}(N,x_r)] = O_m[\text{sin-DFT}(N/2,x_r)] + O_m[\text{DCT}(N/4,x_r)] \tag{5.22a}$$

$$O_a[\text{sin-DFT}(N,x_r)] = O_a[\text{sin-DFT}(N/2,x_r)] + O_a[\text{DCT}(N/4,x_r)] + (3N/4)-2 \tag{5.22b}$$

De (5.5,5.12,5.15), on dérive la complexité suivante pour la transformation de Fourier sur une séquence réelle de longueur N:

$$\begin{aligned} O_m[\text{DFT}(N,x_r)] &= O_m[\text{cos-DFT}(N/2,x_r)] + O_m[\text{sin-DFT}(N/2,x_r)] \\ &= O_m[\text{DFT}(N/2,x_r)] + 2 \cdot O_m[\text{DCT}(N/4,x_r)] \end{aligned} \tag{5.23a}$$

$$\begin{aligned} O_a[\text{DFT}(N,x_r)] &= O_a[\text{cos-DFT}(N/2,x_r)] + O_a[\text{sin-DFT}(N/2,x_r)] \\ &= O_a[\text{DFT}(N/2,x_r)] + 2 \cdot O_a[\text{DCT}(N/4,x_r)] + (3N/2)-2 \end{aligned} \tag{5.23b}$$

Pour la transformation de Fourier sur une séquence complexe, la complexité est:

$$O_m[\text{DFT}(N,x_c)] = 2 \cdot O_m[\text{DFT}(N,x_r)] \tag{5.24a}$$

$$O_a[\text{DFT}(N,x_c)] = 2 \cdot O_a[\text{DFT}(N,x_r)] + 2N-4 \tag{5.24b}$$

Finalement, la transformation en cosinus sur une séquence réelle de longueur N nécessite le nombre suivant d'opérations (selon (5.19)):

$$O_m[DCT(N,x_r)] = O_m[DFT(N,x_r)] + (3N/2)-2 \tag{5.25a}$$

$$O_a[DCT(N,x_r)] = O_a[DFT(N,x_r)] + (3N/2)-3 \tag{5.25b}$$

Avec les relations (5.21-25), il est possible d'évaluer récursivement le nombre d'opérations requises pour une transformation de dimension arbitraire. Pour ce faire, il est nécessaire de connaître les conditions initiales des récursions ci-dessus. Pour N=1, il n'y a pas d'opérations à effectuer, et pour N=2, le nombre d'opérations est donné ci-dessous:

$$O_m[\text{cos-DFT}(2,x_r)] = O_m[\text{sin-DFT}(2,x_r)] = 0$$

$$O_m[DCT(2,x_r)] = 1$$

$$O_a[\text{cos-DFT}(2,x_r)] = O_a[DCT(2,x_r)] = 2$$

$$O_a[\text{sin-DFT}(2,x_r)] = 0 \tag{5.26}$$

Les solutions des relations (5.21-25), étant donné (5.26), sont égales à:

i) Transformation de Fourier-cosinus sur une séquence réelle de longueur N:

$$O_m[\text{cos-DFT}(N,x_r)] = N/4 \cdot (\text{Log}_2(N) - 3) + 1 \tag{5.27a}$$

$$O_a[\text{cos-DFT}(N,x_r)] = N/4 \cdot (3 \cdot \text{Log}_2(N) - 5) + \text{Log}_2(N) + 2 \tag{5.27b}$$

ii) Transformation de Fourier-sinus sur une séquence réelle de longueur N:

$$O_m[\text{sin-DFT}(N,x_r)] = N/4 \cdot (\text{Log}_2(N) - 3) + 1 \tag{5.28a}$$

$$O_a[\text{sin-DFT}(N,x_r)] = N/4 \cdot (3 \cdot \text{Log}_2(N) - 5) - \text{Log}_2(N) + 2 \tag{5.28b}$$

iii) Transformation de Fourier sur une séquence réelle de longueur N:

$$O_m[DFT(N,x_r)] = N/2 \cdot (Log_2(N) - 3) + 2 \quad (5.29a)$$

$$O_a[DFT(N,x_r)] = N/2 \cdot (3 \cdot Log_2(N) - 5) + 4 \quad (5.29b)$$

iv) Transformation de Fourier sur une séquence complexe de longueur N:

$$O_m[DFT(N,x_c)] = N \cdot (Log_2(N) - 3) + 4 \quad (5.30a)$$

$$O_a[DFT(N,x_c)] = 3N \cdot (Log_2(N) - 1) + 4 \quad (5.30b)$$

v) Transformation en cosinus sur une séquence réelle de longueur N:

$$O_m[DCT(N,x_r)] = N/2 \cdot Log_2(N) \quad (5.31a)$$

$$O_a[DCT(N,x_r)] = N/2 \cdot (3 \cdot Log_2(N) - 2) + 1 \quad (5.31b)$$

5.1.3 Comparaison avec d'autres algorithmes

Depuis le développement initial de la FFCT, plusieurs faits nouveaux sont apparus. D'abord, il s'avère qu'une contribution peu connue [yav68] proposait déjà en 1968 un algorithme de DFT réelle et complexe qui utilise exactement le même nombre d'opérations que l'algorithme de FFCT (additions et multiplications). La publication en question est très rarement référencée, et à notre connaissance, l'algorithme n'a jamais été implanté (dire que sa dérivation est complexe fait figure d'euphémismes au vu de la publication en question). Ensuite, simultanément avec l'algorithme de FFCT apparaissaient deux algorithmes tout aussi performants, celui du "split radix" [duh84a] et celui de "recursive cyclotomic factorisation" [mar84]. Il s'avère d'ailleurs que ces deux derniers algorithmes sont isomorphes, et nous ne parlerons dans la suite plus que de celui du "split radix". Notons tout de suite que l'algorithme du "split radix" et celui de FFCT atteignent le même nombre d'opérations dans tous les cas de figures examinés [duh85a], mais qu'ils ne sont pas pour autant isomorphes (ce qui se vérifie au moyen d'un simple contre-exemple).

Pour notre comparaison, nous allons prendre les algorithmes qui étaient usuels au moment de l'introduction de l'algorithme de FFCT et ceci afin de montrer les progrès

obtenus. Notre comparaison portera sur la transformation de Fourier de séquences réelles et complexes, ainsi que sur la transformation en cosinus. Nous choisirons dans chaque cas l'algorithme classique le plus utilisé et nous renvoyons à [vet84b] pour une comparaison plus exhaustive.

Dans le cas d'une transformation de Fourier portant sur une séquence complexe, il y a évidemment une pléthore d'algorithmes possibles. Parmi ceux-ci , l'algorithme de Rader-Brenner [rad76] a été choisi car il utilise un minimum de multiplications. La comparaison est faite dans le tableau 5.1, pour des longueurs allant de 8 à 2048.

Si la séquence à transformer est réelle , deux méthodes différentes sont couramment utilisées. La première consiste à transformer simultanément deux séquences réelles, puis à départager les résultats à l'aide d'additions auxiliaires. Cette méthode a deux désavantages dans les applications pratiques: elle augmente le délai (ce qui est gênant dans les applications en temps réel), et elle utilise deux fois plus d'espace mémoire pour les données, une augmentation difficilement acceptable dans le contexte des processeurs de traitement du signal. La seconde méthode consiste à réduire le problème à celui d'une DFT complexe de longueur moitié, plus quelques opérations supplémentaires (en fait, de l'ordre de N/2 multiplications complexes à la sortie). En raison des désavantages intrinsèques à la première des deux méthodes, la comparaison faite dans le tableau 5.2 utilise la seconde comme algorithme classique de référence. La DFT complexe est à nouveau calculée à l'aide de l'algorithme de Rader-Brenner.

Dans le cas de la transformée en cosinus, l'algorithme le plus utilisé est certainement celui de Chen et al [che77], bien qu'il utilise plus de multiplications qu'un algorithme indirect passant par une FFT et qu'il soit passablement complexe. C'est donc par rapport à cet algorithme que nous comparons dans le tableau 5.3 les résultats de la FFCT pour le calcul de la DCT.

L'évaluation des tableaux 5.1-3 montre que l'algorithme de FFCT diminue le nombre d'additions tout en obtenant le même nombre de multiplications que l'algorithme de Rader-Brenner pour la DFT complexe, qu'il diminue tant le nombre d'additions que de multiplications pour la DFT réelle et enfin, qu'il réduit le nombre de multiplications pour la DCT tout en gardant un nombre comparable d'additions.

Quoique ces résultats soient intéressants puisque les réductions obtenues sont non négligeables, il n'est pas encore clair que ces gains puissent être effectivement traduits en temps de passage réduit sur un processeur, voire en surface de silicium moindre lors de l'implantation VLSI. Ces points fondamentaux seront discutés plus en

Tableau 5.1: Comparaison entre le nombre d'opérations utilisées par l'algorithme de Rader-Brenner [rad76] et celui de FFCT pour le calcul de la transformation de Fourier complexe.

Longueur	FFT de Rader-Brenner		FFCT	
N	multiplications	additions	multiplications	additions
8	4	52	4	52
16	20	148	20	148
32	68	424	68	388
64	196	1 104	196	964
128	516	2 720	516	2 308
256	1 284	6 464	1 284	5 380
512	3 076	14 976	3 076	12 292
1 024	7 172	34 048	7 172	27 652
2 048	16 388	76 288	16 388	61 444

Tableau 5.2: Comparaison entre le nombre d'opérations utilisées par l'algorithme de [bri74] (passant par une FFT de Rader-Brenner de longueur $N/2$) et celui de FFCT pour le calcul de la transformation de Fourier réelle.

Longueur	FFT		FFCT	
N	multiplications	additions	multiplications	additions
8	10	42	2	20
16	26	122	10	60
32	66	290	34	164
64	162	710	98	420
128	386	1 678	258	1 028
256	898	3 870	642	2 436
512	2 050	8 766	1 538	5 636
1 024	4 610	19 582	3 586	12 804
2 048	10 242	43 262	8 194	28 676

Tableau 5.3: Comparaison entre le nombre d'opérations utilisées par l'algorithme de Chen et al [che77] et celui de FFCT pour le calcul de la transformation en cosinus réelle.

Longueur	DCT de Chen		FFCT	
N	multiplications	additions	multiplications	additions
8	16	26	12	29
16	44	74	32	81
32	116	194	80	209
64	292	482	192	513
128	708	1 154	448	1 217
256	1 668	2 690	1 024	2 817
512	3 844	6 146	2 304	6 401
1 024	8 708	13 826	5 120	14 337
2 048	19 460	30 722	11 264	31 745

détail dans les sections concernant les implantations.

Au vu de ces résultats, nous constatons qu'en plus d'utiliser un nombre minimum de multiplications, l'algorithme de FFCT requiert aussi le nombre minimum d'opérations (additions plus multiplications). Il est dès lors intéressant de le comparer avec les meilleurs algorithmes existants pour les transformations dont la longueur peut être factorisée en un produit de facteurs premiers entre eux, c'est-à-dire l'algorithme de Winograd (WFTA) et l'algorithme des facteurs premiers (PFA). Afin de réaliser cette comparaison, nous allons normaliser les résultats des différents algorithmes par rapport à la longueur sur laquelle ils portent, c'est-à-dire que nous parlerons du nombre d'opérations par point transformé. Cette comparaison apparaît dans les figures 5.4, 5.5, et 5.6 pour le nombre de multiplications, additions et la somme des deux respectivement, et ceci pour des longueurs allant de 16 jusqu'à 2520. Les nombres d'opérations pour la WFTA et la PFA proviennent de [nus82].

De ces figures ressort le net avantage de la WFTA en ce qui concerne le nombre de multiplications. La PFA par contre n'améliore que peu le nombre de multiplications par rapport à la FFCT. Pour le nombre d'additions, la situation est inversée, puisque la FFCT en utilise le moins, suivie de la PFA puis de la WFTA. Finalement, en ce qui concerne le nombre total d'opérations, il apparaît dans la figure 5.6 que les 3 algorithmes sont très proches les uns des autres, et ceci pour toutes les longueurs considérées. Notons que nous avons pris le meilleur algorithme de WFTA pour cette comparaison, c'est-à-dire celui qui réduit le nombre d'additions de 10 à 15% grâce à la méthode dite du "split nesting" [nus82], donc un algorithme à structure très complexe. L'algorithme de WFTA habituel aurait une performance moindre dans les figures 5.5 et 5.6 (le nombre de multiplications étant identique). Remarquons également que la comparaison ci-dessus est faite pour le calcul de la DFT complexe, ce qui désavantage l'algorithme de FFCT puisque nous avons vu auparavant qu'il donnait les résultats les plus intéressants dans le cas de transformations réelles.

En conclusion à ces diverses comparaisons, il est possible de dire que d'un point de vue de complexité arithmétique, l'algorithme de FFCT (et celui du "split-radix") apparaît comme le meilleur des algorithmes connus pour des tranformations de longueur $N=2^M$, et qu'il se compare également favorablement avec les meilleurs algorithmes pour des séquences de longueurs décomposables en facteurs premiers entre eux (WFTA,PFA).

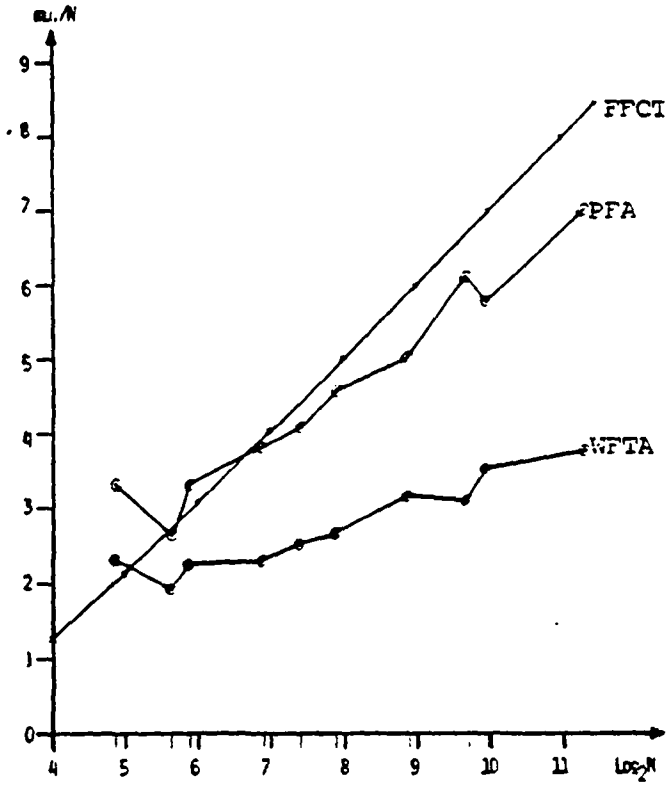

Figure 5.4: Comparaison du nombre de multiplications par point transformé entre les algorithmes de PFA, WFTA et FFCT pour la transformation de Fourier complexe pour N=16..2520.

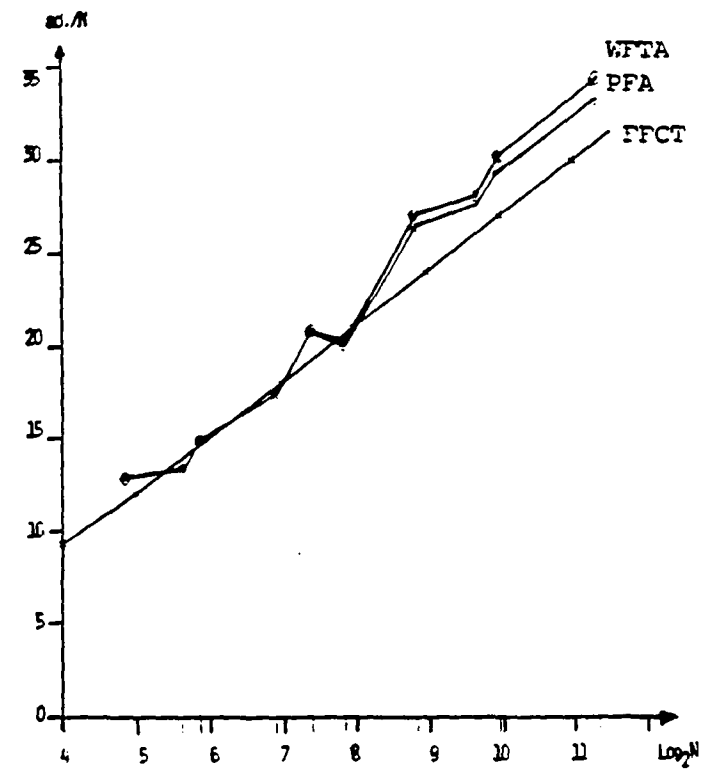

Figure 5.5: Comparaison du nombre d'additions par point transformé entre les algorithmes de PFA, WFTA et FFCT pour la transformation de Fourier complexe pour N=16..2520.

5.1.4 Complexité de calcul théorique

Les relations (5.27) à (5.31) donnent des complexités de calcul de l'ordre de $N \cdot Log_2(N)$, tant pour le nombre de multiplications que d'additions, ce qui est habituel pour ce type d'algorithme. Pourtant, les travaux sur la complexité de calcul théorique ont montré que la complexité multiplicative de la transformée de Fourier est linéaire, et l'algorithme de Winograd en est l'illustration la plus connue. Pour des transformations de longueurs égales à des puissances de 2, une des démonstrations prouvant la complexité linéaire [duh84b] utilise également une division du problème en un sous-problème de dimension moitié et de deux sous-problèmes de dimension un quart, tout comme l'algorithme du "split radix" et l'algorithme de FFCT. Nous allons montrer ci-dessous que l'algorithme de FFCT peut être utilisé pour développer un algorithme de transformation de Fourier ayant une complexité multiplicative linéaire et donnant lieu en fait à un nombre minimum de multiplications.

Pour ce faire, nous allons utiliser sans démonstration un résultat dû à Duhamel [duh85b] qui démontre qu'une transformation en cosinus de longueur $N=2^M$ peut être évaluée à l'aide d'une convolution cyclique et d'additions auxiliaires. De plus, une des multiplications apparaissant dans le calcul de la convolution cyclique est triviale [duh85b], et, étant donné la complexité multiplicative de la convolution cyclique [nus82], on trouve aisément qu'une transformée en cosinus de longueur N requiert au moins le nombre de multiplications suivant:

$$O_m[DCT(N,x_r)] = 2N - Log_2(N) - 2 \qquad (5.32)$$

En vertu de la relation (5.23), l'algorithme de FFCT permet d'obtenir une DFT de longueur N avec une DFT de longueur N/2 et de deux DCT de longueur N/4, et ceci au prix d'additions seulement. Une DFT de longueur N est donc obtenue à partir de deux DCT de N/4, puis deux DCT de longueur N/8 et ainsi de suite jusqu'à deux DCT de longueur 2 qui correspondent au calcul d'une DFT de longueur 8. Le nombre de multiplications nécessaire est égal à (selon (5.32)):

$$O_m[DFT(N,x_r)] = 2 \cdot \sum_{i=1}^{Log_2(N)-2} (2^{i+1} - i - 2) \qquad (5.33)$$

où le terme entre parenthèse correspond à la complexité d'une DCT de 2^i. La solution analytique de la relation (5.33) est égale à:

$$O_m[DFT(N,x_r)] = 2N - (Log_2(N))^2 - Log_2(N) - 2 \qquad (5.34)$$

Une transformation de Fourier portant sur une séquence réelle de longueur N nécessite donc moins de 2N multiplication générale. Notons que nous n'avons pas démontré ci-dessus que le nombre de multiplications donné par (5.34) était minimum. Pour ce faire, il faudrait encore montrer que la séparation du problème réalisée par l'algorithme de FFCT est optimale, ainsi que le calcul de la DCT par convolution. En fait, Heidemann [hei85b] a montré que le nombre de multiplications en (5.32) est minimum pour l'évaluation de la DCT. De surcroît, notons que la division du problème dans l'algorithme de FFCT est celle utilisée dans [duh84b] pour dériver un algorithme optimal, et finalement, le nombre de multiplications en (5.34) est bien la moitié de celui donné dans [hei85a] pour un algorithme de DFT complexe optimisé en nombre de multiplications. La conjecture que le nombre de multiplications donné par (5.34) est minimum est donc très vraisemblable.

Le tableau 5.4 compare les nombres de multiplications donnés par (5.32) et (5.34) avec les nombres effectivement utilisés dans l'algorithme récursif de FFCT pour la DFT et la DCT ((5.29) et (5.31)). Il ressort de ce tableau que l'algorithme de FFCT n'est optimal en nombre de multiplications que jusqu'à N=16 pour la DFT et seulement jusqu'à N=4 pour la DCT. Notons toutefois que si l'on tient également compte du nombre d'additions, les gains réalisés sur le nombre de multiplications s'évaporent. Par exemple, l'algorithme de DCT pour N=8 proposé par Heidemann [hei85b] utilise bien une multiplication en moins, mais par contre 29 additions en plus (dont 15 peuvent être considérées comme des décalages). Pour la DFT de longueur 32, la situation est similaire: 32 multiplications et 222 additions (dont 30 décalages) pour l'algorithme optimisé en multiplications contre 34 multiplications et 164 additions pour l'algorithme de FFCT. L'algorithme de FFCT est donc avantageux en nombre total d'opérations, même s'il n'est pas optimal en nombre de multiplications.

En outre de l'intérêt théorique que présentent ces résultats sur la complexité multiplicative, il en résulte des implications pratiques intéressantes. L'équivalence entre DCT et convolution, par exemple, peut être utilisée pour dériver des architectures originales basée sur des transformées en nombres entiers (pour l'évaluation de la convolution) et conduisant à un nombre de multiplicateurs égal au nombre de points de la transformée [duh85b].

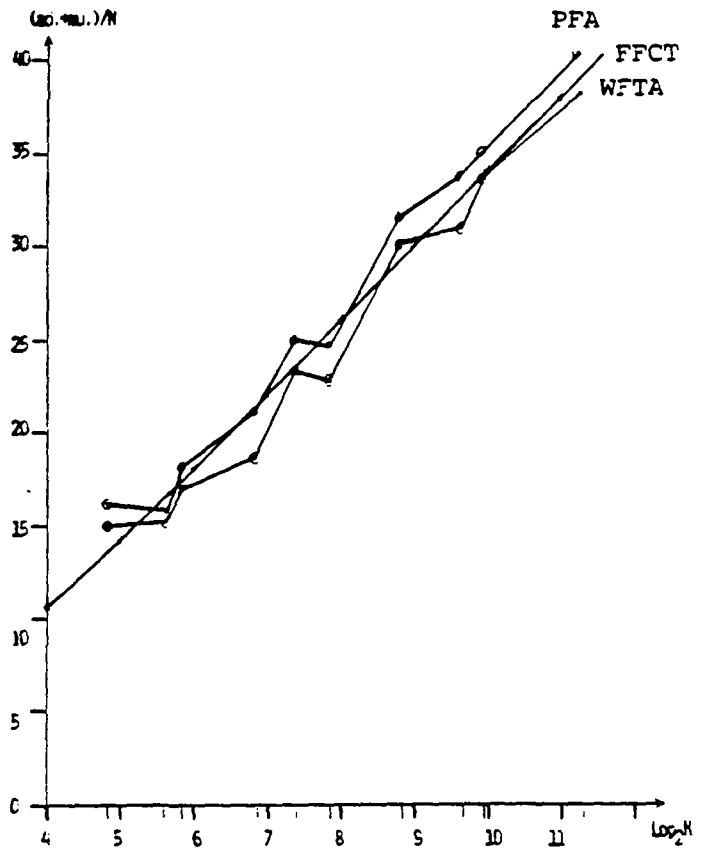

Figure 5.6: Comparaison du nombre d'opérations (additions plus multiplications) par point transformé entre les algorithmes de PFA, WFTA et FFCT pour la transformation de Fourier complexe pour N=16..2520.

Tableau 5.4: Comparaison du nombre minimum de multiplications avec le nombre utilisé par l'algorithme de FFCT.

	DFT		DCT	
N	μ_{min}	μ_{ffct}	μ_{min}	μ_{ffct}
2	0	0	1	1
4	0	0	4	4
8	2	2	11	12
16	10	10	26	32
32	32	34	57	80
64	84	98	120	192
128	198	258	247	448
256	438	642	502	1024
512	932	1538	1012	2304
1024	1936	3586	2035	5120

5.2 Généralisations

Nous avons vu dans la section précédente que l'algorithme de FFCT se compare de façon intéressante avec d'autres algorithmes. Nous allons maintenant généraliser l'approche sous-jacente à cet algorithme à toute une série de problèmes, et ceci tout en maintenant un avantage par rapport à d'autres méthodes en ce qui concerne la complexité de calcul requise. Il est entendu dans la suite que les séquences considérées sont en général d'une longueur égale à une puissance de 2.

5.2.1 Séquences avec symétries

Le calcul de la transformation de Fourier d'une séquence ayant une symétrie est une opération relativement fréquente, par exemple lors de l'évaluation de la fonction d'autocorrélation ou de la convolution par transformée de séquences réelles. En fait, l'algorithme de FFCT s'applique immédiatement dans ces différents cas [vet85d].

a) Séquences symétriques

Une séquence symétrique possède la propriété suivante:

$$x_s(n) = x_s(N-n) \qquad n = 1 \ldots N/2-1$$

$$x_s(0),\ x_s(N/2) \qquad \text{arbitraires} \tag{5.35}$$

La transformée de Fourier d'une telle séquence possède une partie imaginaire nulle. Selon (5.5) et (5.12), il résulte que:

$$DFT(k,N,x_s) = \text{cos-DFT}(k,N,x_s)$$

$$= \text{cos-DFT}(k,N/2,x_{s1}) + 2\cdot DCT(k,N/4,x_{s2}) \qquad k=0 \ldots N-1 \tag{5.36a}$$

avec $$x_{s1}(n) = x_s(2n) \qquad k=0 \ldots N/2-1 \tag{5.36b}$$

$$x_{s2}(n) = x_s(2n+1) \qquad k=0 \ldots N/4-1 \tag{5.35c}$$

Remarquons que x_{s1} est encore une séquence symétrique, donc il est possible d'appliquer la méthode récursivement. Il en résulte la complexité de calcul suivante, où le facteur 2 a été négligé comme étant un simple décalage:

$$O_m[DFT(N,x_{rs})] = N/4 \cdot (Log_2(N) - 3) + 1 \quad (5.37a)$$

$$O_a[DFT(N,x_{rs})] = N/4 \cdot (3Log_2(N) - 7) + Log_2(N) + 3 \quad (5.37b)$$

où l'indice rs indique le fait que le signal est réel et symétrique.

b) Séquences antisymétriques

Dans ce cas, la séquence possède la propriété suivante:

$$x_a(n) = -x_a(N-n) \qquad n= 1 \ldots N/2-1$$

$$x_a(0) = x_a(N/2) = 0 \quad (5.38)$$

La transformée de Fourier de x_a est purement imaginaire, et selon (5.5) et (5.15), égale à:

$$DFT(k,N,x_a) = -j \cdot \text{sin-DFT}(k,N,x_a)$$

$$= -j \cdot (\text{sin-DFT}(k,N/2,x_{a1}) + 2 \cdot DCT(k,N/4,x_{a2})) \quad k=0 \ldots N-1 \quad (5.39a)$$

avec $$x_{a1}(n) = (-1)^n \cdot x_a(2n) \quad k=0 \ldots N/2-1 \quad (5.39b)$$

$$x_{a2}(n) = x_a(2n+1) \quad k=0 \ldots N/4-1 \quad (5.39c)$$

x_{a1} correspond à nouveau à une séquence antisymétrique, et l'algorithme décrit dans (5.39) peut donc être réitéré. La complexité, à nouveau sans tenir compte du facteur 2 précédent la DCT, est alors:

$$O_m[DFT(N,x_{ra})] = N/4 \cdot (Log_2(N) - 3) + 1 \quad (5.40a)$$

$$O_a[DFT(N,x_{ra})] = N/4 \cdot (3Log_2(N) - 7) - Log_2(N) + 3 \quad (5.40b)$$

Remarquons que (5.39a) est égal à (5.37a) (tout comme (5.27a) était identique à (5.28a)).

c) Séquences hermitiennes

Nous disons d'une séquence qu'elle est hermitienne si elle possède la propriété suivante:

$$x_h(n) = [\ x_h(N-n)\ ]^* \qquad n= 1\ ..\ N/2-1$$

$$x_h(0),\ x_h(N/2) \qquad \text{purement réels} \tag{5.41}$$

De telles séquences sont importantes, puisqu'elles sont les représentations fréquentielles de signaux réels (voir (5.9)). Chaque fois qu'il sera nécessaire de retrouver un signal dans le domaine temporel à partir de sa représentation fréquentielle (par exemple après un filtrage effectué dans le domaine de Fourier), il faudra prendre une transformation de Fourier inverse d'une séquence hermitienne. Notons que la transformation de Fourier inverse ne se distingue de la transformation de Fourier que par un changement de signe dans la relation (5.5).

En fait, la relation (5.41) implique que la partie réelle de x_h est symétrique et satisfait donc à (5.35) et que la partie imaginaire est antisymétrique (voir (5.38)). La transformation de Fourier d'une séquence hermitienne se réduit donc à une cos-DFT de la partie réelle, à une sin-DFT de la partie complexe et enfin à une recombinaison du résultat final à l'aide de N-2 additions (puisque sin-DFT possède N-2 sorties différentes de zéro selon (5.8)). La complexité de calcul qui résulte de ces diverses opérations (toujours en négligeant les mises à l'échelle par un facteur 2) est la suivante:

$$O_m[DFT(N,x_h)] = N/2 \cdot (Log_2(N) - 3) + 2 \tag{5.42a}$$

$$O_a[DFT(N,x_h)] = N/2 \cdot (3Log_2(N) - 5) + 4 \tag{5.42b}$$

Il est intéressant de noter que ceci est exactement la même complexité que celle requise par la transformation de Fourier d'une séquence réelle. Ce fait n'est évidemment pas étonnant, car on pourrait également obtenir un algorithme de DFT d'une séquence hermitienne par simple inversion du diagramme de fluence de la DFT d'un signal réel. Dans ce cas, la complexité reste la même (à l'exception de facteurs d'échelle égaux à 2).

5.2.2 Convolution et autocorrélation

A l'aide de l'algorithme de FFCT et de ses versions adaptées aux signaux comportant des symétries, nous allons considérer l'évaluation de convolutions et d'autocorrélations par transformée. Nous n'indiquerons que les complexités de calcul des convolutions et autocorrélations cycliques, étant entendu que pour la même opération mais apériodique, il faudra ajouter N-1 zéros à une séquence de longueur N avant de pouvoir effectuer le calcul par transformée [nus82]. Pour la convolution, il faut d'abord calculer la transformée de Fourier du signal réel (il est admis que la transformée du filtre est pré-calculée). Dans le domaine transformé, tant le filtre que le signal ont une symétrie hermitienne (au sens de (5.38)), tout comme leur produit point à point. Il est aisé de voir que le résultat de la convolution dans le domaine transformé nécessite, en raison de la symétrie hermitienne de tous les vecteurs impliqués, 2 multiplications réelles (pour les points de fréquence k=0 et N/2) et N/2-1 multiplications complexes (pour k=1..N/2-1), donc en tout:

$$3N/2-1 \text{ multiplications et } 3N/2-3 \text{ additions} \qquad (5.43)$$

Finalement, il faut calculer une transformation de Fourier inverse afin d'obtenir le résultat de la convolution dans le domaine temporel. La charge de calcul totale devient, selon (5.29), (5.42) et (5.43):

Convolution d'une séquence réelle de longueur N:

$$O_m[\text{conv}(N,x_r)] = N\cdot \text{Log}_2(N) - 3N/2 + 3 \qquad (5.44a)$$

$$O_a[\text{conv}(N,x_r)] = 3N\cdot \text{Log}_2(N) - 7N/2 + 5 \qquad (5.44b)$$

Notons que cette méthode requiert environ 50% de moins d'additions que celle proposée dans [mca72]. Dans la suite, nous la comparerons également avec la convolution utilisant la transformée de Hartley (voir section 5.2.3).

Lors du calcul de l'autocorrélation d'un signal, nous pouvons profiter dans une mesure encore plus large que dans le cas de la convolution du fait que l'algorithme de FFCT peut pleinement tirer parti des symétries qui existent dans les signaux auxquels il est appliqué. A nouveau, on commence par calculer la transformée de Fourier du signal dont on cherche la fonction d'autocorrélation. Dans le domaine transformé, on doit évaluer $[X(k)]\cdot[X(k)]^*$ [opp75], ce qui nécessite 2 multiplications réelles et 1 addition par point (sauf pour k=0 et N/2 où 1 multiplication réelle est

suffisante), donc en tout:

N multiplications et N/2-2 additions (5.45)

Le résultat de ce calcul d'autocorrélation dans le domaine transformé est une séquence réelle et symétrique (puisque X(k) est une séquence hermitienne). La transformée inverse se réduit donc à une DFT portant sur une séquence symétrique et dont la complexité est donnée par la relation (5.37). En tout, la charge de calcul suivante est utilisée:

Autocorrélation circulaire d'une séquence réelle de longueur N:

$$O_m[auto(N,x_r)] = 3N/4 \cdot Log_2(N) - 5N/4 + 3 \qquad (5.46a)$$

$$O_a[auto(N,x_r)] = 9N/4 \cdot Log_2(N) - 15N/4 + Log_2(N) + 5 \qquad (5.46b)$$

Cet algorithme se compare bien avec d'autres méthodes proposées pour le calcul de l'autocorrélation, et une telle comparaison est effectuée dans le tableau 5.5 avec l'algorithme de [red80] qui est basé sur des transformées rectangulaires.

5.2.3 Comparaison avec la transformation de Hartley

Cette section compare la méthode de convolution utilisant la transformée de Hartley avec celle basée sur l'algorithme de FFCT. Ensuite, une variation originale est proposée qui tente d'allier les avantages respectifs des deux méthodes.

Hartley a proposé il y a plus de quarante ans [har42] une transformation analogique similaire à la transformation de Fourier. En plus de posséder la propriété de convolution, cette transformation a l'avantage d'être sa propre inverse et d'être purement réelle. Restée néanmoins relativement obscure, cette transformation a connu un regain d'intérêt à la suite de la publication de sa version discrète [bra83] puis d'un algorithme rapide pour la calculer [bra84]. La définition de la transformation de Hartley est la suivante [bra83]:

$$DHT(k,N,x) = \sum_{n=0}^{N-1} (\cos(\frac{2\pi nk}{N}) + \sin(\frac{2\pi nk}{N})) \cdot x(n)$$

$$= \text{cos-DFT}(k,N,x) + \text{sin-DFT}(k,N,x) \qquad k=0\,..\,N-1 \qquad (5.47)$$

Si le vecteur **x** est réel, sa transformée de Hartley est donc purement réelle. L'inverse de la transformation de Hartley est simplement:

$$IDHT(k,N,\mathbf{x}) = 1/N \cdot \sum_{n=0}^{N-1} (\cos(\frac{2\pi nk}{N}) + \sin(\frac{2\pi nk}{N})) \cdot x(n)$$

$$= 1/N \cdot (\text{cos-DFT}(k,N,\mathbf{x}) + \text{sin-DFT}(k,N,\mathbf{x})) \qquad k=0\ ..\ N-1 \qquad (5.48)$$

L'inverse est donc égale à la transformation elle-même, à l'exception d'un facteur d'échelle 1/N. La complexité de calcul nécessaire à l'évaluation de (5.47) ou (5.48) est égale à la somme des complexités requises pour la cos-DFT et la sin-DFT, plus N-2 additions de sortie (il n'y pas d'addition de sortie pour k=0 et N/2 en vertu de (5.8)), donc en tout à:

$$O_m[DHT(N,\mathbf{x}_r)] = N/2 \cdot (\text{Log}_2(N) - 3) + 2 \qquad (5.49a)$$

$$O_a[DHT(N,\mathbf{x}_r)] = N/2 \cdot (3\text{Log}_2(N) - 3) + 2 \qquad (5.49b)$$

Notons que dans [duh86], nous proposons une méthode qui inclut les additions supplémentaires (sauf 2) dans la transformation de Fourier (par un "nesting" approprié). Dans ce cas, que nous ne détaillons pas ici, les complexités de la DFT et de la DHT sont donc les mêmes, à l'exception de deux additions de plus pour cette dernière. La convolution dans le domaine transformé est donnée par la relation suivante [bra83]:

$$Y(k) = X(k) \cdot H_e(k) + X(N-k) \cdot H_o(k) \qquad (5.50a)$$

où: $$H_e(k) = \text{cos-DFT}(k,N,\mathbf{h}) \qquad (5.50b)$$

$$H_o(k) = \text{sin-DFT}(k,N,\mathbf{h}) \qquad (5.50c)$$

A priori, la convolution dans le domaine de Hartley nécessite de l'ordre de deux multiplications par point (et c'est ce qui apparaît dans la littérature [bra83]), mais nous allons montrer que tout comme dans le cas de Fourier, 3/2 multiplications sont suffisantes. Notons que selon (5.6-7), $H_e(k)=H_e(N-k)$ et $H_o(k)=-H_e(N-k)$. Y(k) et Y(N-k) peuvent donc être évalués simultanément, et ceci de la façon suivante:

$$\begin{bmatrix} Y(k) \\ Y(N-k) \end{bmatrix} = \begin{bmatrix} H_e(k) & H_o(k) \\ -H_o(k) & H_e(k) \end{bmatrix} \cdot \begin{bmatrix} X(k) \\ X(N-k) \end{bmatrix} \qquad k=1\ ..\ N/2-1 \qquad (5.51a)$$

$$Y(0) = X(0) \cdot H_e(0) \tag{5.51b}$$

$$Y(N/2) = X(N/2) \cdot H_e(N/2) \tag{5.51c}$$

La matrice en (5.51a) est une matrice de rotation, donc le produit matrice-vecteur nécessite 3 multiplications et 3 additions (voir (5.29-20)). La convolution dans le domaine de Hartley se fait donc à l'aide de:

$$3N/2 - 1 \text{ multiplications} \quad \text{et} \quad 3N/2\text{-}3 \text{ additions} \tag{5.52}$$

Ceci est exactement la même charge de calcul que celle utilisée pour la convolution dans le domaine de Fourier (voir (5.43)). Notons que même si la transformation de Hartley est une transformation réelle, la convolution dans le domaine transformé n'utilise pas une multiplication point par point, mais une opération qui est fort similaire à une multiplication complexe.

Afin d'obtenir le résultat de la convolution dans le domaine temporel, il faut encore prendre la DHT inverse du vecteur **Y**. La charge de calcul totale est donc:

$$O_m[conv_{dht}(N,x_r)] = N \cdot Log_2(N) - 3N/2 + 3 \tag{5.53a}$$

$$O_a[conv_{dht}(N,x_r)] = 3N \cdot 3Log_2(N) - 3N/2 + 1 \tag{5.53b}$$

La convolution utilisant la transformation de Hartley utilise donc 2N-4 additions de plus que la convolution par Fourier dans l'approche détaillée ci-dessus, ou 4 additions supplémentaires dans celle proposée dans [duh86].

En dépit de cette légère augmentation de la charge de calcul, l'approche par transformation de Hartley possède une certaine élégance due à l'identité de l'inverse avec la transformation elle-même. Ci-dessous, nous allons décrire une nouvelle méthode que nous avons proposée dans [duh86]. Elle utilise une transformation de Fourier réelle tant pour passer dans le domaine fréquentiel que pour en revenir. La méthode consiste simplement à confondre les sommes et différences à la sortie de la DHT avec les rotations de (5.51a). Notons d'abord que X(k) et X(N-k) dans (5.51a) sont obtenus à partir de cos-DFT(k,N,x_r) et sin-DFT(k,N,x_r) de la façon suivante (en utilisant (5.6-7)):

$$\begin{bmatrix} X(k) \\ X(N-k) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \text{cos-DFT}(k,N,x_r) \\ \text{sin-DFT}(k,N,x_r) \end{bmatrix} \qquad (5.54)$$

$$k=1 \ldots N/2-1$$

En combinant (5.54) avec (5.51a), on obtient le résultat suivant pour la convolution dans le domaine de Hartley:

$$\begin{bmatrix} Y(k) \\ Y(N-k) \end{bmatrix} = \begin{bmatrix} H_e(k)+H_o(k) & H_e(k)-H_o(k) \\ H_e(k)-H_o(k) & -H_e(k)-H_o(k) \end{bmatrix} \cdot \begin{bmatrix} \text{cos-DFT}(k,N,x_r) \\ \text{sin-DFT}(k,N,x_r) \end{bmatrix} \qquad (5.55)$$

$$k=1 \ldots N/2-1$$

Pour k=0 et N/2, les relations (5.51b-c) restent valables. La matrice dans (5.55) est à nouveau une matrice de rotation, et cette méthode d'évaluation de la convolution dans le domaine transformé utilise donc exactement le même nombre d'opérations que celui donné par (5.43) ou (5.52). Pour revenir dans le domaine temporel, nous appliquons une transformation de Fourier au vecteur **Y**, puis le vecteur **y** des échantillons du résultat est obtenu comme:

$$\begin{bmatrix} y(n) \\ y(N-k) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \text{cos-DFT}(k,N,\mathbf{Y}) \\ \text{sin-DFT}(k,N,\mathbf{Y}) \end{bmatrix} \qquad (5.56)$$

$$k=1 \ldots N/2-1$$

c'est-à-dire à l'aide de N-2 additions à partir de la transformation de Fourier de **Y**. La figure 5.7 illustre schématiquement cette méthode de convolution circulaire pour N=8. Au total, nous obtenons la charge de calcul suivante:

$$O_m[\text{conv}_{\text{new}}(N,x_r)] = N \cdot \text{Log}_2(N) - 3N/2 + 3 \qquad (5.57a)$$

$$O_a[\text{conv}_{\text{new}}(N,x_r)] = 3N \cdot 3\text{Log}_2(N) - 5N/2 + 3 \qquad (5.57b)$$

Remarquons que le nombre de multiplications est identique pour les 3 méthodes (relations (5.44a), (5.53a) et (5.57a) mais que le nombre d'additions diffère légèrement.

Il est minimum pour la méthode utilisant une transformation de Fourier puis une transformation de Fourier inverse sur une séquence hermitienne (5.44b). La méthode que nous venons d'introduire et qui est basée sur deux transformations de Fourier réelles utilise N-2 additions supplémentaires à la sortie (5.57b), et finalement, la méthode basée sur la transformation de Hartley requiert 2N-4 additions supplémentaires selon (5.53b) ou 4 additions supplémentaires selon la méthode décrite dans [duh86]. Ces différents algorithmes illustrent bien le compromis possible entre complexité structurelle et complexité arithmétique.

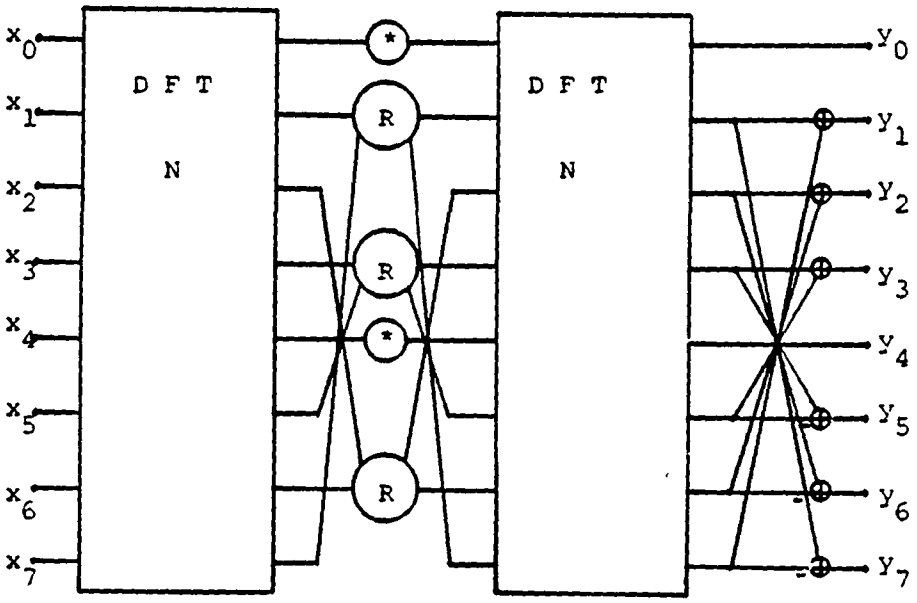

Figure 5.7: Convolution circulaire utilisant 2 transformations de Fourier réelle et des additions de sortie.

5.2.4 Transformations impaires

Les transformations impaires sont importantes pour le calcul de transformations de Fourier multidimensionnelles par transformées polynomiales [nus82]. Elles apparaissent également dans certains bancs de filtres [bon76,mas85], lorsque les filtres modulés doivent être réels.

La transformation de Fourier impaire ("odd DFT") est définie comme suit:

$$DFT_o(k,N,x) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi(2k+1)n}{2N}} \qquad k=0\,..\,N-1 \qquad (5.58)$$

L'algorithme de FFCT peut être appliqué aisément à cette transformation. Il est possible de définir une cos-DFT, une sin-DFT et une DCT impaire, puis d'appliquer la

division par 2 et 4 tout comme dans le cas de l'algorithme de FFCT habituel. Quelques détails doivent être adaptés, et nous référons le lecteur intéressé aux publications [vet85a,vet85b] où cet algorithme est développé. Nous donnons simplement la complexité qui résulte de cette approche:

$$O_m[DFT_o(N,x_r)] = N/2 \cdot (Log_2(N) - 1) \tag{5.59a}$$

$$O_a[DFT_o(N,x_r)] = 3N/2 \cdot (Log_2(N) - 1) \tag{5.59b}$$

Concernant cet algorithme de DFT impaire portant sur des séquences réelles, il n'y a pas à notre connaissance d'algorithme comparable. Pourtant, comme son application immédiate est le calcul de transformées multidimensionnelles, il semble important d'avoir à disposition un algorithme réel qui économise la place mémoire ainsi qu'un grand nombre d'additions.

Si le vecteur x est complexe, il est possible de calculer les transformées des parties réelles et imaginaires séparément, puis de recombiner le résultat à l'aide de 2N additions. Notons que le nombre de multiplications est alors égal à celui obtenu avec l'algorithme de Rader-Brenner pour la DFT impaire [nus82], mais que le nombre d'additions est substantiellement diminué (par exemple pour le cas N=1024, l'algorithme de Rader-Brenner utilise 30% d'additions de plus que l'approche FFCT).

5.2.5 Transformations bidimensionnelles

L'importance des transformées multidimensionnelles, en particulier en traitement d'image, est considérable. Comme les charges de calcul et les espaces mémoires requis sont gigantesques, l'optimisation de ces paramètres est de grande importance. Encore plus que dans le cas monodimensionnel, la plupart des signaux à traiter sont soit réels ou hermitiens, et la contrainte d'espace mémoire restreint pousse à l'utilisation d'algorithmes adaptés à ces cas particuliers.

Les algorithmes de transformation de Fourier bidimensionnelle les plus efficaces sont ceux basés sur les transformées polynomiales [nus78,nus79,nus82]. Dans la suite, nous adapterons ces approches connues pour leur efficacité au cas de transformations de Fourier et en cosinus portant sur des signaux réels, et ceci en améliorant l'utilisation de l'espace mémoire ainsi que le nombre d'additions utilisées. Notons qu'une approche directe des transformations bidimensionnelles par la méthode de division par 2 et 4 (qui est à la base de l'algorithme de FFCT et du "split radix") permet bien d'obtenir des résultats meilleurs qu'avec les algorithmes traditionnels

(méthode ligne-colonne ou "vector radix"), mais utilise plus de multiplications que les méthodes par transformées polynomiales (en tout cas pour N suffisamment grand). C'est la raison pour laquelle seules ces dernières seront considérées par la suite. Posons les définitions suivantes:

Transformation de Fourier bidimensionnelle:

$$DFT(k_1,k_2,NxN,x) = \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} x(n_1,n_2)\cdot e^{-j\frac{2\pi(n_1k_1+n_2k_2)}{N}} \qquad (5.60)$$

$$k_1,\ k_2 = 0..N-1$$

Transformation en cosinus bidimensionnelle:

$$DFT(k_1,k_2,NxN,x) = \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} x(n_1,n_2)\cdot \cos\left(\frac{2\pi(n_1+1)k_1}{N}\right)\cdot$$

$$\cdot \cos\left(\frac{2\pi(n_2+1)k_2}{N}\right) \qquad k_1,\ k_2 = 0..N-1 \qquad (5.61)$$

Notre propos n'est pas de décrire l'algorithme de DFT bidimensionnelle par transformée polynomiale, celui-ci ayant fait l'objet de nombreuses publications. Nous allons plutôt appliquer l'algorithme de FFCT à certaines parties de la méthode par transformée polynomiale (par exemple aux transformées impaires qui apparaissent en cours de calcul), et ceci afin d'améliorer certaines performances (typiquement, le nombre d'additions et l;'utilisation de la mémoire pour le cas réel). Si le signal $x(n_1,n_2)$ est réel, les transformées polynomiales (qui n'utilisent que des additions), ont une réduction de complexité d'exactement 50%. De surcroît, toutes les transformations impaires portent sur des séquences réelles, et nous pouvons donc utiliser l'algorithme efficace qui a été développé à la section précédente. L'algorithme de DFT bidimensionnelle qui résulte de cette approche est décrit en détail dans [vet85a,vet85b], et nous y renvoyons le lecteur intéressé pour plus de précisions. Notons simplement la complexité qui en résulte:

$$O_m[DFT(NxN,x_r)] = N^2/2\cdot Log_2(N) - 7N^2/6 + 8/3 \qquad (5.62a)$$

$$O_a[DFT(NxN,x_r)] = 5N^2/2\cdot Log_2(N) - 13N^2/6 + 56/3 \qquad (5.62b)$$

Ces résultats, en particulier le nombre d'additions, sont très intéressants. De surcroît, peu d'algorithmes efficaces pour la transformation de signaux multidimensionnels réels ont été proposés, alors qu'il s'agit d'une des applications les plus importantes. Si cet algorithme est utilisé pour calculer des transformées de signaux complexes, il donnera

lieu à un nombre de multiplications identique à celui donné dans [nus82], mais diminuera sensiblement le nombre d'additions (par exemple, une réduction de plus de 50% pour le cas 1024x1024).

En ce qui concerne la transformation en cosinus, il est possible de l'évaluer à l'aide d'une transformation de Fourier et de rotations et ceci en permutant les échantillons d'entrée de façon similaire au cas monodimensionnel. Comme la charge de calcul utilisée par les rotations de sortie est relativement importante voire dominante pour les petites transformées (comme 8x8 et 16x16, des cas importants en codage d'image [jai81]), nous avons été amenés à optimiser cette partie de l'algorithme. Il s'en suit que 4 points du résultats de la DCT peuvent être évalués simultanément à partir de 4 points de la DFT réelle du signal d'entrée permuté, et ceci au prix de deux rotations et d'additions supplémentaires (voir [vet85a,vet85b]). En tout, le calcul d'une DCT à partir d'une DFT utilise:

$$3N^2/2 - 2N \text{ multiplications et } 5N^2/2 - 6N + 2 \text{ additions} \tag{5.63}$$

c'est-à-dire bien 1.5 multiplications par point transformé (tout comme dans le cas monodimensionnel) mais par contre plus d'additions (2.5 par point). La complexité d'une DCT de dimension NxN sur un signal réel est donc, selon (5.62) et (5.63):

$$O_m[DCT(NxN,x_r)] = N^2/2 \cdot \text{Log}_2(N) + N^2/3 - 2N + 8/3 \tag{5.64a}$$

$$O_a[DCT(NxN,x_r)] = 5N^2/2 \cdot \text{Log}_2(N) + N^2/3 - 6N + 62/3 \tag{5.64b}$$

Ce résultat est nettement meilleur que tous les algorithmes de DCT bidimensionnelle comparables. Les autres algorithmes utilisant des transformées polynomiales [nus82,nas83,pei84] sont relativement efficaces mais pas adaptés aux signaux réels, et la complexité obtenue avec l'algorithme de [che77] en approche ligne-colonne est comparée avec la méthode que nous proposons dans le tableau 5.6. Les avantages de ce dernier, qui allie les avantages des transformées polynomiales avec ceux de la FFCT, sont évidents pour les multiplications.

5.2.6 Survol des résultats obtenus avec l'algorithme de FFCT

Avant de passer aux applications, nous tenterons ici de faire le point sur les résultats obtenus avec l'algorithme de FFCT. Pour ce faire, nous avons réuni toutes les complexités correspondant aux différents problèmes dans le tableau 5.7.

Remarquons d'abord que la complexité multiplicative suit exactement la complexité intrinsèque des problèmes. Par exemple, si M est le nombre de multiplications utilisées pour une DFT complexe, alors la DFT réelle utilise bien M/2, et la DFT réelle-symétrique M/4 multiplications. Notons que ce résultat n'est pas toujours obtenu avec d'autres algorithmes. Par exemple, dans le cas du calcul d'une DFT réelle à partir d'une DFT complexe de longueur moitié (voir [bri74] et la section 5.1.3), la complexité multiplicative n'est pas réduite de 50%, et l'algorithme dans [sie75] pour des séquences réelles-symétriques ne divise pas le nombre de multiplications par 4. Ces réductions ne sont pas vérifiées exactement pour les additions. La raison en est que les transformations entre différents problèmes sont souvent faites à l'aide d'additions, et ceci afin d'éviter une augmentation de la charge multiplicative.

Une comparaison des résultats donnés dans la table 5.7 avec d'autres algorithmes permet de remarquer que:

- la complexité multiplicative de l'algorithme de FFCT est minimale (parfois égale à d'autres algorithmes pour certains problèmes) parmi les algorithmes ayant un nombre restreint d'additions (nous excluons donc les algorithmes à complexité multiplicative linéaire donné à la section 5.1.4 et qui nécessitent un nombre exorbitant d'additions)

- la complexité additive est en général meilleure que celle d'autres algorithmes, ou en tous cas comparable
- le nombre total d'opérations est le minimum connu pour tous les problèmes considérés.

Tableau 5.5: Comparaison entre l'algorithme de [red80] et celui basé sur l'algorithme de FFCT pour le calcul de l'autocorrélation circulaire d'une séquence réelle de longueur N.

	[red80]		FFCT	
N	mults.	adds.	mults.	adds.
4	8	24	4	10
8	20	58	11	32
16	52	138	31	93
32	132	322	83	250
64	324	738	211	635
128	772	1666	515	1548
256	1796	3712	1219	3661

Tableau 5.6 Comparaison de l'algorithme de [che77] avec l'algorithme de transformée polynomiale/FFCT pour l'évaluation d'une transformation en cosinus de dimension NxN.

	[che77]		FFCT	
N	μ	α	μ	α
8	256	416	104	474
16	1408	2368	568	2570
32	7424	12416	2840	12970
64	37376	61696	13528	62442
128	181248	295424	62552	291434
256	854016	1377280	283480	1331054

Tableau 5.7 Complexité requise pour résoudre certains problèmes avec l'algorithme de FFCT ($N=2^m$).

problème	multiplications	additions	formule
cos-DFT(N,x_r)	$N/4\cdot(\log_2(N)-3)+1$	$N/4\cdot(3\log_2(N)-5)+\log_2(N)+2$	(5.27)
sin-DFT(N,x_r)	$N/4\cdot(\log_2(N)-3)+1$	$N/4\cdot(3\log_2(N)-5)-\log_2(N)+2$	(5.28)
DFT(N,x_r)	$N/2\cdot(\log_2(N)-3)+2$	$N/4\cdot(3\log_2(N)-5)+4$	(5.29)
DFT(N,x_c)	$N\cdot(\log_2(N)-3)+4$	$N\cdot(3\log_2(N)-1)+4$	(5.30)
DCT(N,x_r)	$N/2\cdot\log_2(N)$	$N/2\cdot(3\log_2(N)-2)+1$	(5.31)
DFT(N,x_{rs})	$N/4\cdot(\log_2(N)-3)+1$	$N/4\cdot(3\log_2(N)-7)+\log_2(N)+3$	(5.37)
DFT(N,x_{ra})	$N/4\cdot(\log_2(N)-3)+1$	$N/4\cdot(3\log_2(N)-7)-\log_2(N)+3$	(5.40)
DFT(N,x_h)	$N/2\cdot(\log_2(N)-3)+2$	$N/4\cdot(3\log_2(N)-5)+4$	(5.42)
conv(N,x_r)	$N\cdot\log_2(N)-3N/2+3$	$3N\cdot\log_2(N)-7N/2+5$	(5.44)
auto(N,x_r)	$3N/4\cdot\log_2(N)-5N/4+3$	$9N/4\cdot\log_2(N)-15N/4+\log_2(N)+5$	(5.46)
DHT(N,x_r)	$N/2\cdot(\log_2(N)-3)+2$	$N/2\cdot(3\log_2(N)-3)+2$	(5.49)
DFT_o(N,x_r)	$N/2\cdot(\log_2(N)-1)$	$3N/2\cdot(\log_2(N)-1)$	(5.59)
DFT(NxN,x_r)	$N^2/2\cdot\log_2(N)-7N^2/6+8/3$	$5N^2/2\cdot\log_2(N)-13N^2/6+56/3$	(5.62)
DCT(NxN,x_r)	$N^2/2\cdot\log_2(N)+N^2/3-2N+8/3$	$5N^2/2\cdot\log_2(N)+N^2/3-6N+62/3$	(5.64)

5.3 Implantation matérielle

Les transformations du type Fourier revêtent une telle importance en pratique qu'il s'avère parfois nécessaire de réaliser un matériel, voire un processeur spécialisé intégré, dédié à cette application. Un exemple de ce type est donné dans cette section.

Le codage de signaux vidéo [jai80] (ayant une fréquence d'échantillonnage de l'ordre de 10 Mhz) en temps réel nécessite soit la mise en parallèle d'un nombre énorme de processeurs commerciaux, soit la réalisation d'un processeur dédié au codage. Dans la méthode de codage d'image par transformée, l'essentiel de la complexité arithmétique nécessaire provient de l'évaluation de transformations en cosinus sur des petites portions de l'image (de la dimension 8x8 ou 16x16). Il est dès lors avantageux de concevoir un circuit hautement spécialisé dont l'unique tâche est de calculer une transformation en cosinus monodimensionnelle de longueur 8 (ou 16), mais ceci à une vitesse de l'ordre de 800 ns par transformation effectuée. Notons qu'un circuit de ce type peut ensuite également être utilisé pour le calcul d'une transformation de Fourier de longueur quatre fois plus grande en utilisant l'algorithme de FFCT.

Un tel circuit a pu être conçu durant un séjour de trois mois aux Bell Laboratories [vet86a]. Comportant quelques 35000 transistors et visant une technologie de CMOS 2,5μ, ce circuit est un exemple de conception à la carte ("full custom design") devenu possible pour des néophytes grâce à la disponibilité d'outils de conception assistée par ordinateurs (CAO) puissants.

Loin de vouloir faire le tour des problèmes d'implantations matérielles de transformées rapides, la section ci-dessous vise à montrer que:
a) des algorithmes complexes (du genre FFCT) peuvent être implantés en silicium, et ceci avec profit. Habituellement, seuls des algorithmes simples (comme la FFT en base 2 ou 4) ont été réalisés en matériel.
b) une méthodologie propre et cohérente, englobant toutes les étapes allant de la formulation de l'algorithme en virgule flottante jusqu'aux plans de masques et au test, est nécessaire. Une telle méthodologie a dû être développée en parallèle avec la conception du circuit [lig86].
c) la conception de circuits doit être ouverte à des personnes autres qu'aux seuls spécialistes de conception de circuits, et une CAO puissante doit donc être disponible.

L'organisation de la section est la suivante. D'abord, le problème concret et ses contraintes sont présentés. Le développement du circuit, de la formulation initiale de l'algorithme de FFCT jusqu'au plan symbolique, est ensuite expliqué. Finalement, la méthodologie que nous préconisons pour un développement cohérent (et déjà utilisée en partie pour ce circuit) est décrite.

5.3.1 Présentation du problème

Le codage d'image vidéo en temps réel par transformation est utilisé dans des applications pratiques telles que la vidéo-conférence [jai80]. Le débit d'information est de l'ordre d'un échantillon de 8 bits par 100ns. La transformation utilisée est en général une transformation en cosinus bidimensionnelle (voir la relation (5.61)) appliquée sur des blocs de dimension 8x8 ou 16x16. On estime que la corrélation est grande à l'intérieur de régions aussi petites de l'image, d'où l'intérêt d'appliquer une transformation décorrélante avant le codage. Dans le domaine transformé, les valeurs sont ensuite quantifiées de façon adaptative (typiquement, peu de bits pour les coefficients correspondants aux fréquences élevées de la DCT).

Pour l'évaluation de la DCT bidimensionnelle, la méthode choisie est celle appliquant l'algorithme de FFCT monodimensionnel sur les lignes, puis sur les colonnes des données (étant donné que la transformation est séparable). Notons qu'il faut normaliser le coefficient 0 par $1/\sqrt{2}$ dans cette application de l'algorithme de FFCT. Cette méthode de calcul de la DCT bidimensionnelle est moins efficace que l'algorithme direct basé sur les transformées polynomiales présenté à la section 5.2.5 (pour 8x8, la méthode ligne colonne utilise 208 multiplications et 367 additions, contre 104 et 474 respectivement pour l'algorithme direct), mais elle se prête nettement mieux à une implantation matérielle. De surcroît, un circuit de DCT monodimensionnel peut également servir à d'autres fins (codage de la parole ou calcul de la DFT par exemple).

Avec un débit d'un échantillon par 100ns mais en tenant compte du fait qu'il faut calculer 2N transformations dans la méthode ligne/colonne, il s'avère qu'une DCT de 8 points doit être évaluée toutes les 400 ns. Si un seul circuit de DCT est utilisé, non seulement la charge de calcul par unité de temps est énorme (>3μ/100ns) mais le débit d'entrée (1 échantillon par 50ns) devient critique. Il est donc préférable d'utiliser une cascade de deux circuits, l'un calculant les DCT sur les lignes et l'autre sur les colonnes de la matrice 8x8. Un tel système est montré dans la figure 5.8.

Le circuit à concevoir devra donc calculer une DCT de 8 points, c'est-à-dire 13 multiplications (à cause de la normalisation du coefficient 0), 29 additions et plusieurs permutations, tout ceci en 800ns. La précision des calculs devra être d'au moins 8 bits (vraisemblablement plus en raison des propagations d'erreurs). A l'époque de la conception de ce circuit et dans la technologie considérée, une multiplication de 12 fois 12 bits prenait environ 150ns.

Avec les remarques ci-dessus, les contraintes à respecter sont posées, et la section suivante montre comment elles ont été prises en considération pour la conception du circuit.

5.3.2 Conception du circuit de DCT

Notons d'abord que l'algorithme de FFCT a été choisi car il utilise, pour N=8, environ 25% de multiplications en moins que celui de Chen et al [che77]. Comme la surface d'un multiplicateur augmente avec le carré du nombre de bits (alors qu'elle varie linéairement pour un additionneur), l'algorithme de FFCT a potentiellement un gain de plus de 20% en surface par rapport à celui de Chen.

a) Transformation de l'algorithme pour l'implantation en silicium

La réduction de la complexité de calcul dans l'algorithme de FFCT est obtenue au prix d'une structure plus complexe. Celle-ci à évidemment un prix en intégration à très large échelle. Nous allons donc d'abord tenter de réduire quelque peu la complexité structurelle, et ceci en augmentant légèrement la charge de calcul. La figure 5.9 montre le diagramme de fluence de la DCT de 8 points avant (a) et après (b) transformation. Dans la partie b), les opérations similaires ont été groupées, et les deux multiplications par $1/\sqrt{2}$ à la sortie 0 et 4 ont été imbriquées avec la somme/différence les précédant, créant ainsi une rotation. Ceci augmente la complexité arithmétique d'une multiplication et d'une addition, mais permet d'avoir quatre rotations à la sortie.

b) Choix de l'arithmétique

Si le choix de la représentation en complément à 2 est évident, il n'est pas aussi simple de choisir le type d'arithmétique. L'arithmétique sérielle [hwa79,mea80] a remporté un certain succès lors de l'implantation d'algorithmes de traitement du signal, surtout en raison de sa simplicité. Elle a néanmoins deux désavantages: le traitement des erreurs d'arrondi est difficile, et le temps de calcul est élevé (ce qui

nécessite la mise en parallèle de plusieurs unités si la charge de calcul par unité de temps est élevée). L'arithmétique parallèle est également relativement simple et permet une charge de calcul élevée. Tant que le nombre de bits n'est pas trop grand, les retards dus à la propagation du report restent tolérables, et ceci même sans utiliser un "carry lookahead" [hwa79]. Finalement, l'arithmétique parallèle-cascade se débarrasse totalement du problème du report, mais elle est relativement complexe à concevoir. Dans la figure 5.10, un additionneur de 4 bits est schématiquement esquissé pour les trois types d'arithmétiques.

Afin de comparer différents types d'arithmétique, il est intéressant de leur associer une fonction coût. Celle-ci est du type surface A fois temps T (ou $A \cdot T^2$ si le délai est un facteur important). Dans le tableau 5.8, l'ordre de grandeur de $A \cdot T$ est donné pour un additionneur réalisé avec différents types d'arithmétique.

Tableau 5.8: Ordre de grandeur de $A \cdot T$

problème	surface	temps	$A \cdot T$
multiplication MxM bits			
- sérielle	$O[M \cdot S_a]$	$O[M \cdot T_a]$	$O[M^2 \cdot S_a \cdot T_a]$
- parallèle	$O[M^2 \cdot S_a]$	$>>O[T_a]$	$>>O[M^2 \cdot S_a \cdot T_a]$
- par.-cascade	$O[M^2 \cdot S_a]$	$O[T_a]$	$O[M^2 \cdot S_a \cdot T_a]$
addition M bits			
- sérielle	$O[S_a]$	$O[M \cdot T_a]$	$O[M \cdot S_a \cdot T_a]$
- parallèle	$O[M \cdot S_a]$	$>O[T_a]$	$>O[M \cdot S_a \cdot T_a]$
- par.-cascade	$O[M \cdot S_a]$	$O[T_a]$	$O[M \cdot S_a \cdot T_a]$

S_a est la surface d'un additionneur 1 bits et T_a le temps d'une telle addition.

L'augmentation du temps dans le cas de l'arithmétique parallèle est dû au report et n'est pas indiqué plus précisément (il dépend de la manière de générer, puis de propager ce report). Il ressort du tableau 5.8 que l'arithmétique sérielle et parallèle-cascade ont exactement la même performance. Pour le circuit en question, la charge de calcul élevée ne permettait pas l'utilisation de l'arithmétique sérielle. Pour un nombre de bits suffisamment petit, la détérioration de la performance de l'arithmétique parallèle ne porte pas trop à conséquence. Par contre, elle est beaucoup plus facile à concevoir que l'arithmétique parallèle-cascade, et elle a donc été choisie en raison des contraintes de temps de conception.

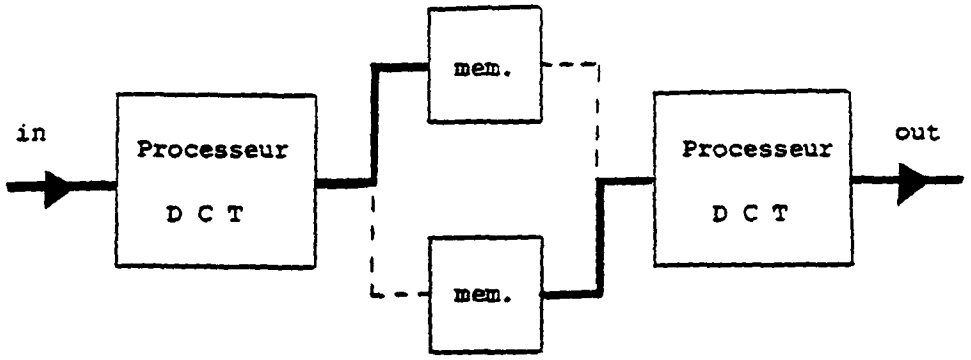

Figure 5.8: Mise en cascade de deux circuits de transformation en cosinus pour l'évaluation d'une transformation bidimensionnelle

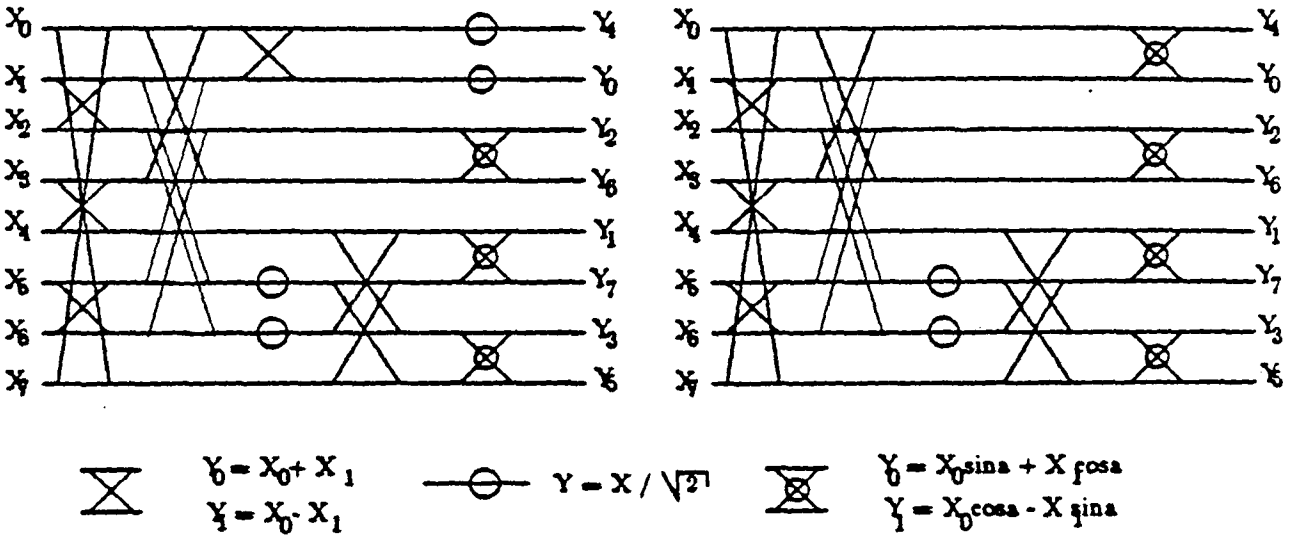

Figure 5.9: Transformation du diagramme de fluence de la DCT de 8 points
a) avant transformation
b) après transformation

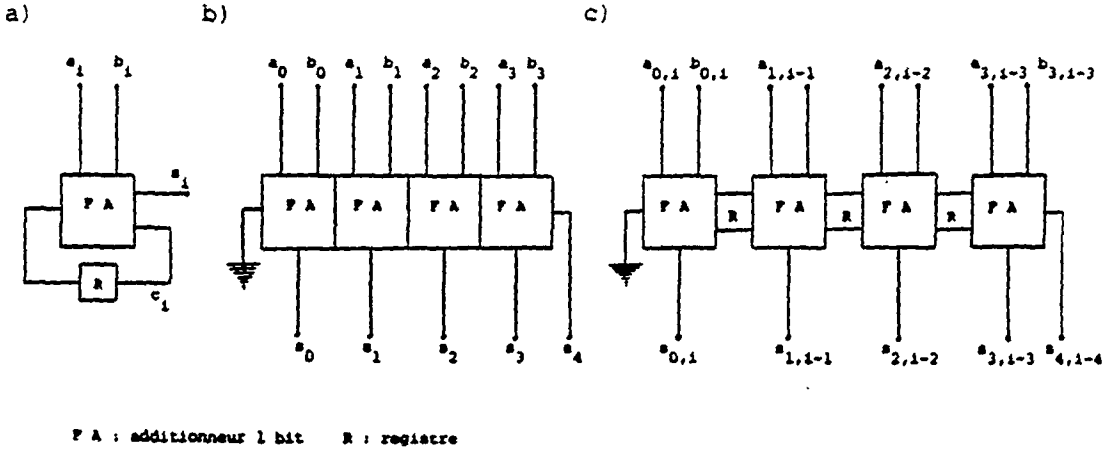

Figure 5.10: Exemple d'un additionneur 4 bits
a) arithmétique sérielle
b) arithmétique parallèle
c) arithmétique parallèle-cascade

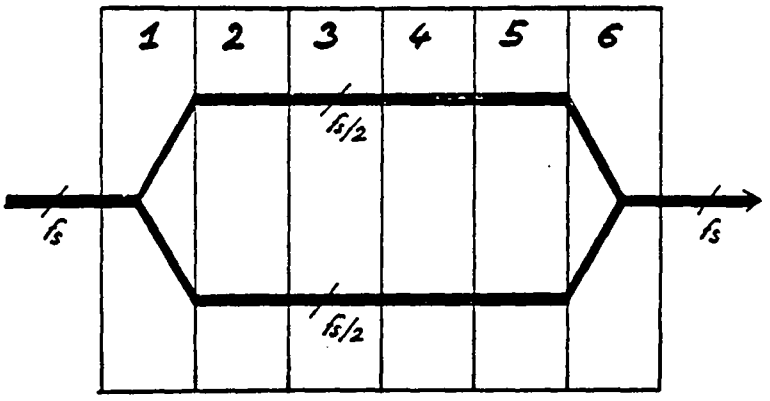

Figure 5.11: Flots de données dans le cicuit

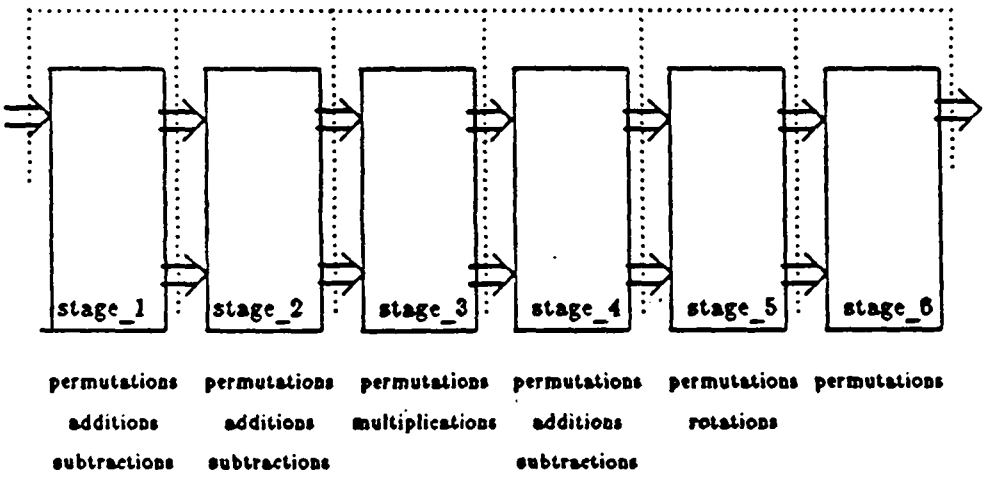

Figure 5.12: Architecture globale du circuit

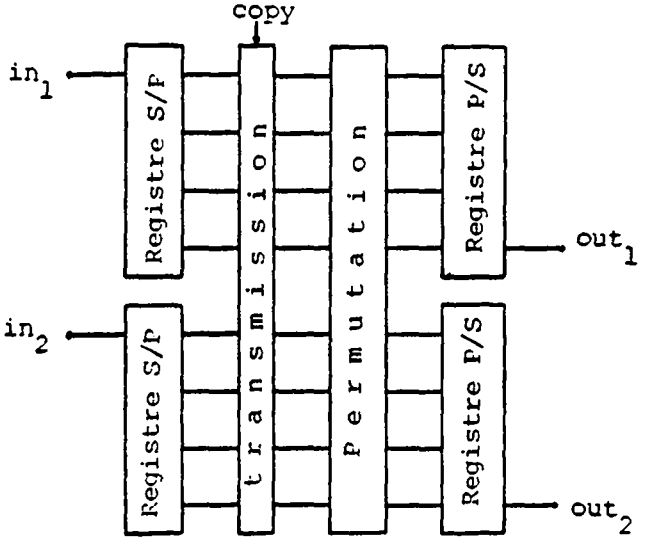

Figure 5.13: Registres à décalage et permutation placés avant l'unité arithmétique

c) Choix architecturaux

Etant donné les contraintes de vitesse et le choix de l'arithmétique, l'architecture se trouve relativement bien définie. En considérant un temps de multiplication de l'ordre de 150 ns et le fait que l'étape de sortie à elle seule requiert une rotation toutes les 200 ns, il est évident que les différentes étapes devront être mise en cascade (pipeline) et séparées par des mémoires tampons (buffers).

Une remarque importante doit être faite: à part deux multiplications isolées (par $1/\sqrt{2}$), toutes les opérations portent sur deux valeurs d'entrée et produisent deux valeurs de sortie. En fait, il s'agit toujours de rotations élémentaires (somme/différence est une rotation de 45 degrés, fois un facteur d'échelle de $\sqrt{2}$). Cette propriété fondamentale est liée au fait que toute transformation unitaire peut être décomposée en rotations élémentaires (appelées rotation de Givens). La conséquence architecturale immédiate de cette propriété est qu'il est avantageux d'avoir deux flots de données en parallèle et sur lesquels on applique les différentes rotations. D'une part, l'unité de calcul traite immédiatement les deux opérandes (aucun stockage intermédiaire n'est nécessaire), et d'autre part, la vitesse d'exécution de l'unité de calcul peut être réduite de moitié (puisque deux données sont traitées simultanément). Ceci est illustré schématiquement dans la figure 5.11, où les étages de calcul correspondent à ceux indiqués dans la figure 5.9b.

Puisque les différents étages sont mis en "pipeline", il est nécessaire de les séparer par des séries de registres. Ceux-ci peuvent être combinés avec les permutations nécessaires dans les différents étages. Il en résulte l'architecture globale suivante:

- 6 étages en "pipeline". Le résultat d'un étage est copié, toutes les 800 ns, dans l'étage suivant
- entrée et sortie du circuit: M bits en parallèle et à la fréquence f_s (10Mhz)
- transmission entre étages: 2 fois M bits en parallèle et à la fréquence $f_s/2$

Cette architecture est illustrée dans la figure 5.12. Notons que chaque étage possède une unité arithmétique adaptée au type de calcul qu'il doit effectuer (rotation, somme/différence, multiplication par $1/\sqrt{2}$). Une largeur de mot de 12 bits a été choisie pour le flot de donnée, ce nombre ayant été jugé suffisant (les images sont codées à 8 bits) et la surface de silicium à disposition ne permettant guère plus.

d) Blocs élémentaires

Les registres et les permutations sont réalisés de la façon suivante. Un registre à décalage à entrée série et sortie parallèle mémorise les bits des échantillons

successifs au fur et à mesure de leur arrivée. A la fin du cycle principal (800 ns), tous les bits sont copiés, à travers un réseau de permutation, dans un registre à décalage à entrée parallèle et sortie série. Celui-ci alimente ensuite l'unité arithmétique. Une section d'un bit de ce système est esquissé dans la figure 5.13.

Concernant les unités arithmétiques, notons que des additionneurs minimaux ont été utilisés. Ceux-ci produisent un report inversé, et les entrées doivent être ajustées en conséquence (par inversion des bits pairs ou impairs). Lors de la mise en cascade de plusieurs additionneurs de ce type, il suffit d'ajuster l'entrée du premier et la sortie du dernier. Une section d'un bit du schéma symbolique de l'étage 2 (comportant une somme et différence) est montrée dans la figure 5.14.

Notons que même si la complexité arithmétique de l'étage 2 (6 additions) est inférieure à celle de l'étage 1 (8 additions), il requiert néanmoins plus de surface. Ceci provient du fait qu'il faut du contrôle supplémentaire et des portes de transmission afin d'éviter l'opération de somme et différence une fois sur quatre. Ceci est un bon exemple de la différence entre la notion de complexité en intégration et la notion de complexité arithmétique.

Pour la multiplication, un générateur a été écrit par A.Ligtenberg [lig86], ce qui a permis de générer automatiquement des multiplicateurs en complément à deux de taille arbitraire. Malheureusement, et par manque de temps, il n'a pas été possible de créer un multiplicateur dédié à la multiplication par $1/\sqrt{2}$. Notons qu'un gain de 50% dans la taille de celui-ci eut été possible (en ne gardant que les colonnes du multiplicateur qui correspondent à des 1 dans la représentation binaire de $1/\sqrt{2}$).

La conception de l'unité de rotation est décrite en détail dans [vet86a] et nous notons que, vu la complexité de cette unité, les opérations ont encore une fois été mises en cascade, créant donc un "pipeline" supplémentaire pour cette opération. Remarquons que les constantes pour les différentes rotations ont été représentées en 10 bits. Une simulation a montré que la multiplication de données par des constantes ayant 80% du nombre de bits de ces données est suffisamment précise. Le schéma de cette unité est montré dans la figure 5.15.

Sans entrer dans les détails, il est à noter que les problèmes de nombre de bits significatifs sont cruciaux, surtout lorsque le nombre total de bits est faible. D'une part, le dépassement de capacité doit être évité à tout prix, et d'autre part, un nombre trop élevé de bits de garde nuit à la précision. Dans le cas de l'unité de rotation, il a été possible de gagner deux bits de précision (sans ajouter de matériel supplémentaire) simplement grâce à une analyse détaillée. D'abord, les deux bits de

poids le plus fort du résultat d'une multiplication en complément à deux sont toujours égaux (sauf dans le cas où les opérandes sont tous deux maximum négatif, cas que nous pouvons exclure pour les constantes qui nous intéressent). Le bit de poids maximum peut donc être négligé. Ensuite, la rotation est une opération unitaire (conservant les longueurs), et il est possible de vérifier que l'addition de sortie dans la figure 5.15 ne produit jamais de dépassement de capacité. Comme la DCT est également une transformation unitaire, donc une rotation multidimensionnelle, elle conserve également les longueurs, permettant donc de gagner un bit de précision. En tout, 3 bits de précision supplémentaires peuvent être obtenus, c'est à dire un gain de 25% dans le cas qui nous intéresse.

Un mot encore sur le contrôle des différentes fonctions. Tous les signaux de contrôles sont générés à partir des deux horloges φ_1 et φ_2 [mea80], et les signaux complémentaires sont obtenus avec de la logique complémentaire afin d'éviter tout retard inutile. Finalement, la phase de transmission φ_1 est choisie plus courte que la phase d'évaluation φ_2, cette dernière étant plus critique.

e) Test

Le test d'un circuit de la complexité de celui que nous décrivons s'avère difficile, et la "testabilité" d'une conception est un des problèmes majeurs de l'intégration à très grande échelle [gut84]. De ce fait, des fonctions de test ont été inclues sur le circuit, et ceci en utilisant les registres à décalage déjà existants pour gèrer les vecteurs de test. Cette méthode n'est pas nouvelle, mais la configuration de notre circuit s'y prête particulièrement bien. Tous les registres à décalage d'un étage donné peuvent, sur commande, être mis en cascade, et chargés ou lus depuis l'extérieur (et ceci à une vitesse réduite si nécessaire). Ainsi, il est possible de tester individuellement chaque étage, ce qui est par exemple avantageux pour déterminer la vitesse limite de fonctionnement de chacun d'eux.

f) Réalisation du circuit

Grâce à l'utilisation d'un sytème de CAO puissant, MULGA [wes81], le plan symbolique du circuit a pu être réalisé en 3 mois (y compris l'apprentissage du système de CAO). Une vue d'ensemble du circuit symbolique est présentée dans la figure 5.16. On reconnaît aisément l'étage 3 avec son multiplicateur, ainsi que l'étage 5 de rotation.

Certains problèmes sont apparus lors de la compaction finale (transformation du schéma symbolique en plan de masques). Ces problèmes étaient essentiellement dus à l'impossibilité, avec MULGA, de faire un assemblage au niveau des plans de masques

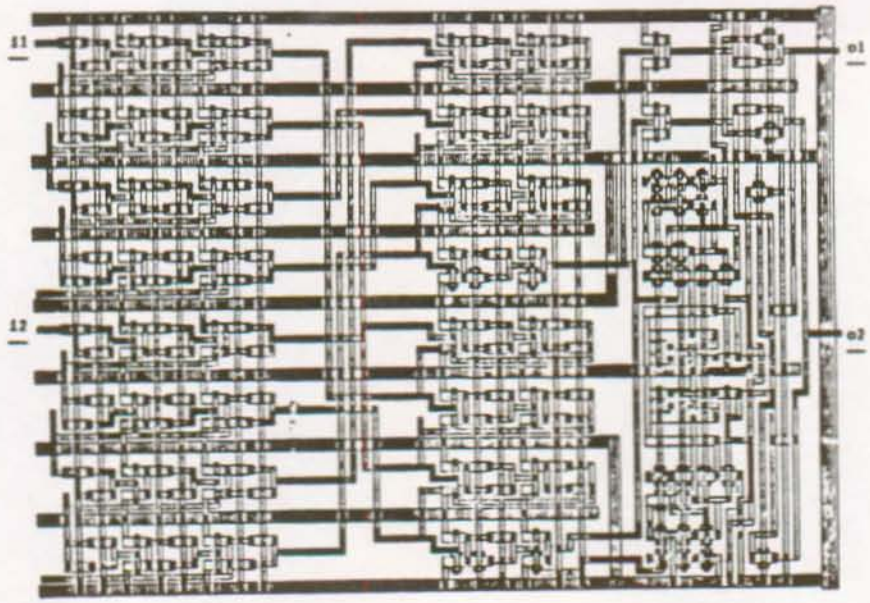

Figure 5.14: Section de 1 bit de l'étage 2, comportant de gauche à droite, les registres d'entrée, la permutation, les registres alimentant l'unité arithmétique, et finalement un bit d'addition/soustraction

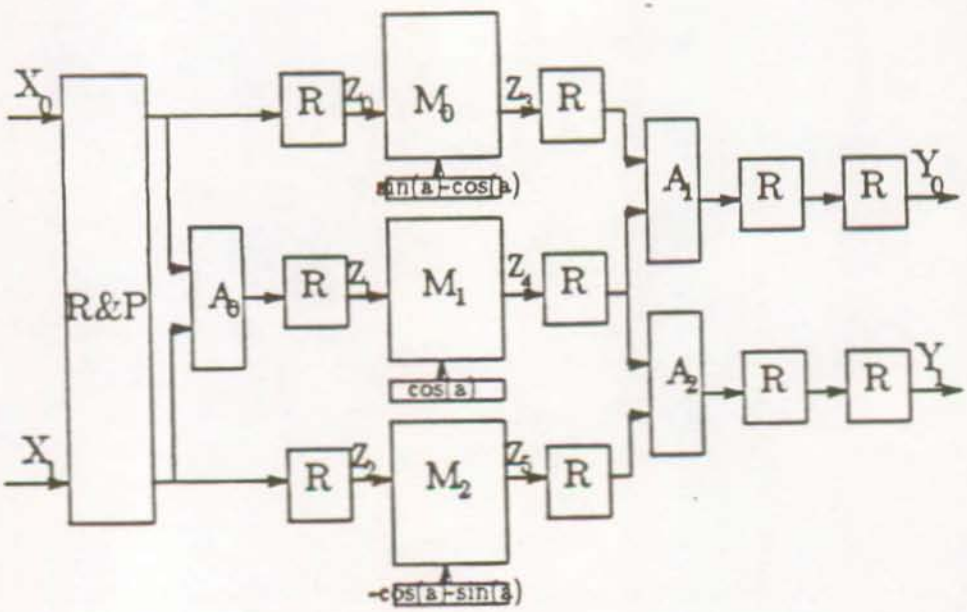

Figure 5.15: Schéma de l'unité de rotation, les sous-étappes étant mises en "pipeline"

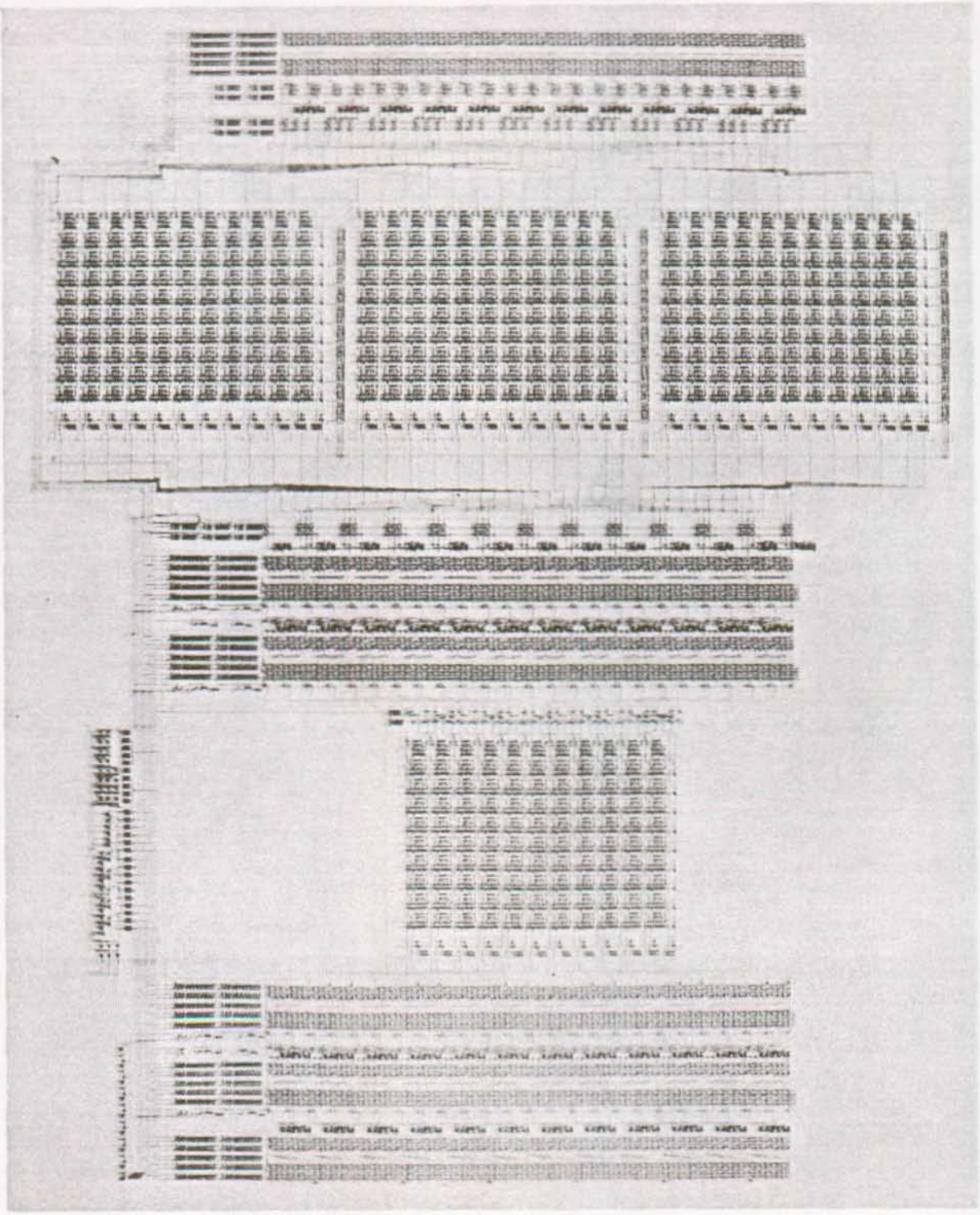

Figure 5.16: Schéma symbolique du circuit complet de 35000 transistors

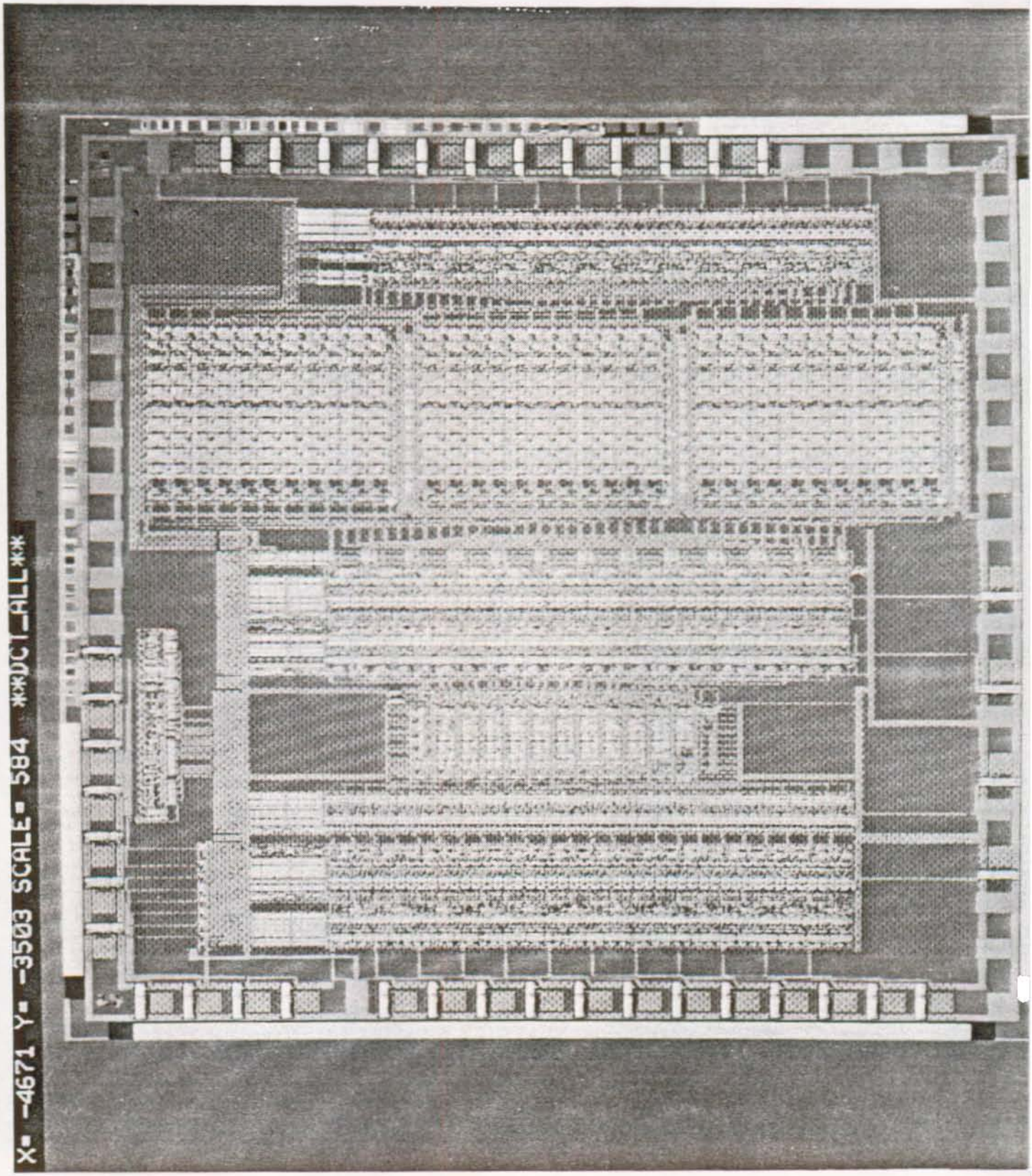

Figure 5.17: Plan de masques du circuit qui a été mis en fabrication

(la compaction globale étant absurde et devenant d'ailleurs impossible vu la taille du circuit). Après amélioration, le circuit a finalement été mis en fabrication et la figure 5.17 montre le plan de masques. Les résultats de ce circuit permettront de mesurer la vitesse d'exécution réelle.

Malgré ces problèmes, cette expérience a été fort profitable. D'une part, elle a démontré la possibilité d'implanter rapidement un algorithme complexe. Vu les contraintes, il est vraisemblable que l'algorithme habituel de Chen n'aurait pas pu être implanté, en tous cas pas en utilisant la même précision de calcul. D'autre part, cette expérience a parfaitement démontré le coût de la communication (permutation), du stockage intermédiaire (registres) et de l'irrégularité (inhibition d'opérations à certaines étapes du calcul), trois facteurs de coût qui sont relativement élevés dans l'algorithme de FFCT. Mais le coût des multiplications est encore plus élevé, et c'est la raison pour laquelle l'algorithme de FFCT reste avantageux dans l'application que nous avons explorée.

Plutôt que de voir dans ce circuit un quelconque résultat définitif, nous préférons l'interpréter comme une étape en direction de l'implantation matérielle efficace d'algorithmes complexes.

5.3.3 Une méthodologie d'implantation en matériel d'algorithmes de traitement du signal

Lors de la réalisation du circuit décrit ci-dessus, il s'avèra rapidement que la gestion de la complexité de conception devenait impossible, à moins de possèder des outils logiciels spécialement conçus à cet effet. Clarifions d'abord le terme de conception. Il englobe toutes les étapes du développement, depuis l'idée initiale (dans notre exemple, l'algorithme de FFCT en formulation mathématique) jusqu'au test du circuit physique qui revient de la fabrication. Lorsque nous parlons de la complexité de cette conception, nous entendons par là:

- l'exploration (à toutes les étapes) de l'espace des possibilités ouvertes
- la réalisation de l'étape suivante, et ceci de façon aussi automatisée que possible
- la vérification (aussi totale que possible) de l'avancement de la conception obtenu par cette nouvelle étape, et la comparaison avec le modèle de référence

L'approche que nous préconisons est bien un développement par étapes, de haut en bas et à l'aide d'améliorations successives (en anglais, top down design by stepwise refinement). Ceci n'a rien d'extraordinaire en soi, si ce n'est que les développements de circuits dans le laboratoire où nous travaillions ne correspondaient en rien à

cette philosophie!. La raison est en partie historique: la conception de circuits a toujours été un domaine réservé à quelques rares spécialistes, et ce n'est que très récemment qu'il a été possible à des "théoriciens" de réaliser leurs idées sous forme de puces. Par le passé, un développement passait par toute une série de personnes avant d'atteindre le silicium, et les trois points mentionnés plus haut ne pouvaient donc être satisfaits.

Identifions d'abord les étapes principales du développement d'un circuit de traitement du signal. Les sept niveaux principaux sont:

1) Niveau algorithmique: ce niveau décrit simplement l'algorithme à l'aide d'arithmétique en virgule flottante. Il permet de vérifier l'algorithme, et servira de modèle de référence par la suite
2) Niveau de précision finie: on associe à chaque variable un nombre de bits de précision. Une exploration des effets de la précision finie est faite
3) Niveau espace/temps: le compromis vitesse d'exécution/espace de silicium requis est exploré, en particulier à l'aide des différents types d'arithmétique (voir sec 5.3.2b))
4) Niveau matériel: les choix de précision et de type d'arithmétique ayant été faits, il est possible de réaliser un modèle très exact du circuit, et ceci à l'aide de primitives élémentaires (additionneurs 1 bit, registres).
5) Niveau du circuit symbolique
6) Niveau du plan de masques
7) Circuit physique

Pour passer d'un niveau à celui immédiatement en dessous, il faut appliquer ce que nous avons appelé des transformations. Celles-ci devraient être aussi automatisées que possible. Notons que les trois derniers niveaux sont disponibles (5 et 6 dans un système de conception au niveau symbolique, et 7 dans la fabrication), ainsi que les transformations entre ces niveaux.

L'implantation logicielle de cette méthodologie [lig86], permettant la représentation cohérente et compatible des différents niveaux ainsi que des transformations en partie automatisée, a commencé parallèlement à la réalisation du circuit de DCT et se poursuit toujours. Elle vise a décrire les niveaux 1 à 4 de façon unifiée (en fait, à l'aide d'un seul langage de haut niveau, C) et en utilisant des primitives différentes selon les niveaux (par exemple un multiplicateur en précision finie pour le niveau 2, et un multiplicateur décrit par des registres et des additionneurs 1 bits pour le niveau 4). Certaines parties des transformations sont automatiques (par exemple, le générateur de schéma symbolique de multiplicateurs évoqué à la section 5.3.2d)), et une grande part des efforts actuels vise à augmenter la partie automatisée des transformations.

L'avantage majeur d'une description des différents niveaux dans un langage unique est la compatibilité ainsi obtenue, permettant par exemple de mélanger des parties d'un circuit n'étant pas au même stade du développement. De surcroît, des interfaces avec les niveaux existants (5,6 et 7) permettent de tester automatiquement et à partir du niveau 4 le schéma symbolique, les masques ainsi que le circuit terminé. Cette possibilité d'effectuer des tests automatiquement est d'une aide inestimable dans le déverminage du circuit (à tous les niveaux). La méthodologie de vérification dans MOVAL est illustrée dans la figure 5.18

L'avantage certain de cette méthodologie est d'obtenir, à toutes les étapes du développement, une image cohérente de l'état de la conception. Evidemment, cette méthodologie requiert une discipline (par exemple, ne pas modifier un niveau inférieur sans refléter les changements aux niveaux supérieurs), tout comme le demande le développement de logiciel [dij76]. Cette similarité entre le développement de matériel et de logiciel (qui deviendra encore plus apparente dans la section suivante) nous conduit à parler de "conception structurée de matériel". A notre avis, c'est la seule approche qui permettra de maîtriser la complexité toujours plus grande de la conception de circuits en intégration à très large échelle.

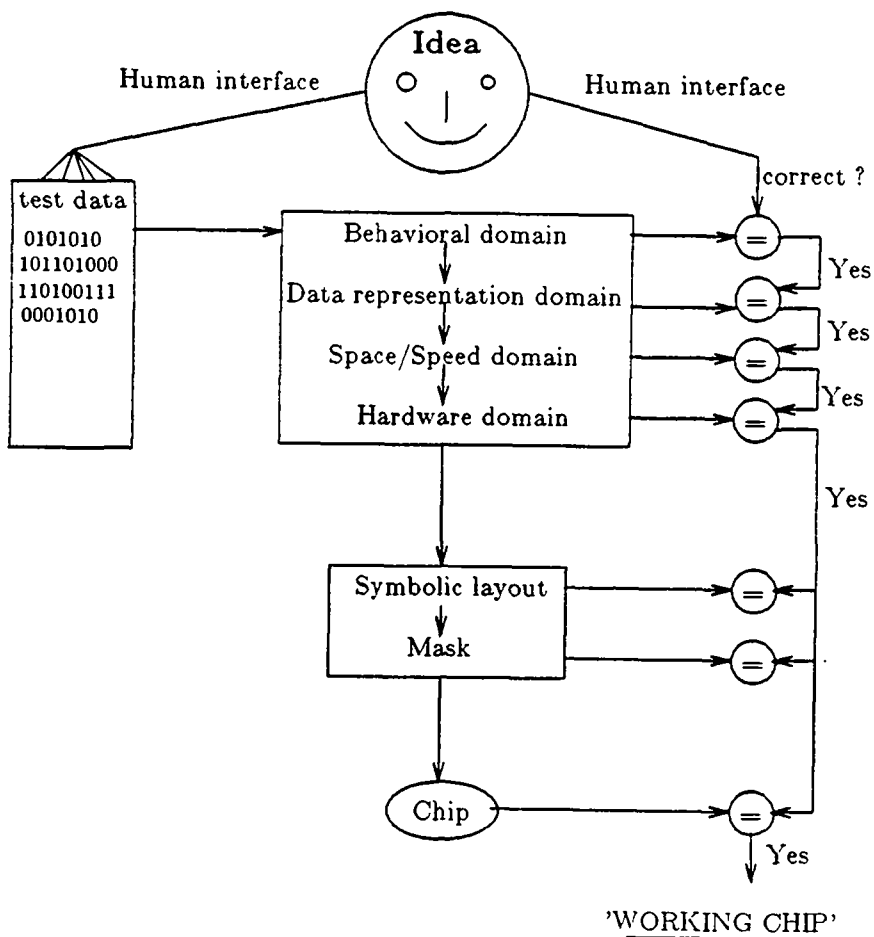

Figure 5.18: Méthodologie de vérification dans MOVAL

5.4 Implantation logicielle

Cette section présente l'implantation logicielle de l'algorithme de FFCT. D'abord, le problème de l'implantation efficace d'algorithmes rapides est présenté, entre autres en discutant la notion de complexité dans ce contexte. Ensuite, l'implantation spécifique de l'algorithme de FFCT en code linéaire sur un processeur de traitement du signal [vet85e] est décrite. Finalement, une méthodologie de développement, similaire à celle présentée pour l'implantation matérielle, est proposée pour l'implantation efficace d'algorithmes de traitement du signal sur des processeurs spécialisés.

5.4.1 Complexité d'un algorithme dans le contexte d'une implantation logicielle

Autant le débat sur la complexité de calcul d'algorithmes rapides a toujours été clairement défini (nombres d'opérations arithmétiques), autant celui concernant leur implantation efficace sur des processeurs réels est resté vague.

Pourtant, même si Winograd ne s'intéresse avant tout qu'au nombre de multiplications par des constantes réelles apparaissant dans un algorithme donné [win80] (les multiplications par des constantes rationnelles ne comptent guère d'un point de vue purement théorique), l'utilisateur s'intéresse uniquement à la vitesse d'exécution de cet algorithme, avec une précision de calcul donnée, sur son ordinateur. Les points de vue du théoricien et du praticien sont donc quelque peu divergents en ce domaine (...).

Le débat sur les implantations a vraiment pris son envol lorsqu'il s'est avéré que l'algorithme de Winograd, quoique théoriquement nettement supérieur à tous les autres algorithmes de FFT (puisqu'il requiert une complexité de l'ordre de O[N] alors que les autres sont en O[NLogN]), était néanmoins plus lent à l'exécution que la FFT en base 4 par exemple!.

Cette constatation a conduit à considérer le nombre de multiplications, d'additions et de transferts (mémoire à registre) [mor78,naw79]. En particulier, il faut remarquer que les temps d'exécution de ces différentes opérations sont très proches les uns des autres sur les processeurs modernes. Notons que du point de vue de la somme des multiplications et des additions, l'algorithme de Winograd est pratiquement équivalent aux bons algorithmes de FFT. Mais en ce qui concerne les transferts, la structure complexe de la WFTA la désavantage nettement par rapport à des algorithmes réguliers du genre FFT base 4. Pourtant, le compte exact des transferts pour un

algorithme donné est une tâche difficile, car il dépend non seulement du nombre de registres, mais également du jeu d'instructions du processeur visé.

Finalement, si l'algorithme est programmé en utilisant un code compact, il faut tenir compte de la complexité liée au calcul de l'adressage. Ce calcul est simple dans le cas FFT, mais relativement complexe dans celui de la WFTA.

Pour mieux cerner le problème de l'efficacité de l'implantation de l'algorithme de FFCT, nous avons choisi de ne considérer que sa version en code linéaire [mor77]. Dans ce cas, toutes les adresses sont pré-calculées, et la complexité d'adressage disparaît. Le prix de cette méthode est une longueur du code proportionnelle au nombre total d'opérations.

5.4.2 Code pour processeur de traitement du signal

Nous avons vu précédemment que l'algorithme de FFCT permettait de résoudre toute une série de problèmes (tableau 5.7) plus efficacement que ne le font les algorithmes classiques de FFT, mais ceci au prix d'une complexité structurelle plus grande. Cette dernière se traduit généralement par un effort de programmation plus élevé, sauf si l'on fait appel à des méthodes de génération automatique de code [mor77]. C'est ce que nous avons été amené à faire, car il était hors de question d'écrire directement du code assembleur pour le processeur de signal visé (TMS 32010, [tx83]). L'annexe A5.1 donne du code linéaire pour des transformations de Fourier et en cosinus portant sur des séquences réelles de petites dimensions et utilisant un nombre minimum d'opérations arithmétiques.

La seule version de l'algorithme qui a dû être écrite à la main est une version récursive en langage Pascal. Cette version se déduit immédiatement de la formulation récursive de l'algorithme de FFCT (relations (5.12), (5.15) et (5.19)).

Comme la machine cible travaille en virgule fixe, il faut simplement ajouter aux différentes opérations des facteurs d'échelle adéquats. Notons qu'une DFT ou une DCT de dimension N peut produire des résultats ayant une amplitude N fois plus grande que les valeurs d'entrée. Rappelons qu'une DFT de longueur N se construit à partir d'une DFT de longueur N/2 et deux DCT de longueur N/4. De surcroît, il y a des sommes/différences avant les DCT ainsi qu'à la sortie pour dériver le résultat final (voir la figure 5.2).

La technique de mise à l'échelle répartie que nous avons appliquée divise par N le

résultat d'une opération portant sur N échantillons. En raison de la récursivité, il est aisé de montrer que si cette règle est appliquée aux petites transformations (2 et 4), elle se vérifie automatiquement pour les transformations plus grandes. Notons simplement que les valeurs de sortie cos-DFT(N/2,N,x) et sin-DFT(N/2,N,x) de la DFT de longueur N doivent être divisées par 2 (car il n'y pas d'opérations de sortie sur ces valeurs, contrairement aux autres, voir la figure 5.2). Cette mise à l'échelle répartie garantit un maximum de précision à toutes les étapes du calcul, tout en garantissant l'impossibilité d'un dépassement de capacité.

Le code Pascal comporte toute l'information définissant l'implantation en virgule fixe de l'algorithme (opérations et mises à l'échelle). Il peut donc être utilisé afin de simuler, sur un grand ordinateur, le comportement numérique de l'implantation en cours de développement.

Le programme récursif Pascal peut maintenant être utilisé pour générer du code linéaire, toujours en Pascal. Celui-ci peut soit être composé d'opérations arithmétiques uniquement (additions, multiplications), soit d'opérations élémentaires (rotations, sommes/différences, multiplications par $1/\sqrt{2}$) ou encore comporter des blocs de petites transformations (de longueur 4 ou 8 par exemple).

La dernière étape consiste à traduire ces opérations élémentaires (ou blocs) en langage assembleur. Ceci ce fait à l'aide d'un programme de traduction [kar85] qui peut faire appel à des séquences de code assembleur optimisé [vet85e].

Rappelons en bref que toutes les étapes, hormis l'écriture du code récursif initial et le choix des facteurs d'échelle, sont effectuées automatiquement. De surcroît, il suffit de vérifier le code récursif initial (comportant déjà les mises à l'échelle) pour garantir un code final correct, et ceci en raison de la construction automatique de ce dernier (on admet que les opérations élémentaires sont codées correctement, ce qui se vérifie aisément).

Outre le fait que le développement du code assembleur se trouve grandement simplifié par cette méthode, le temps d'exécution obtenu est fort intéressant comme le montre le tableau 5.9.

Tableau 5.9: Comparaison entre les temps de calcul donnés par le constructeur (TI) et l'implantation de l'algorithme de FFCT

	FFT réelle			FFT complexe		
N	TI (μs.)	FFCT (μs.)	gain (%)	TI (μs.)	FFCT (μs.)	gain (%)
16	62.6	33.4	46.6	108.8	78.4	28
32	163.4	93.6	42.7	291.2	208.4	28
64	400.2	243	39.3	737.6	526.4	29
128	955.4	601.6	37	-	-	-

Les temps donnés dans le tableau 5.9 ont été obtenus après optimisation du code des DFT de longueurs 4 et 8, de certaines multiplications isolées et d'additions de sortie (voir [vet85e] pour plus de détails). Néanmoins, cet effort d'optimisation relativement grand n'a permis qu'une amélioration minime (4-6% selon les longueurs) par rapport aux temps obtenus par translation directe des opérations élémentaires. Ceci tend à montrer que le code généré automatiquement n'est pas très loin d'être optimum.

5.4.3 Une méthodologie d'implantation d'algorithmes sur processeurs spécialisés

La méthode d'implantation que nous avons utilisée pour développer un logiciel de FFT sur TMS 320 est fort similaire à celle qui était à la base du développement du circuit de DCT décrit dans la section 5.3. Nous en rappelons les grandes lignes afin d'en faire une synthèse.

Le développement du logiciel pour processeur spécialisé passe par plusieurs niveaux distincts qui sont:

1) Niveau algorithmique: il s'agit d'une formulation de l'algorithme à l'aide d'arithmétique en virgule flottante et en langage de haut niveau (utilisant par exemple la récursivité)
2) Niveau de précision finie: on attribue des précisions aux différentes variables, et les facteurs d'échelle sont choisis (la récursivité est toujours permise). Des simulations peuvent être faites.
3) Niveau représentation du code: ce niveau est une image en langage de haut niveau du code assembleur qui va être généré. Seules apparaissent des primitives que l'on sait traduire en assembleur. Ce code peut être linéaire ou bouclé, mais ne

comporte plus de structures complexes (du genre récursivité).

4) Niveau assembleur: code compatible avec le simulateur de la machine cible

5) Niveau code exécutable

Notons que les niveaux 4 et 5 sont déjà mis à disposition par le constructeur. Tout comme dans le cas du matériel, il s'agit de réaliser les transformations entre les niveaux, et ceci de façon aussi automatique que possible. La transformation entre les niveaux 4 et 5 est également fournie par le constructeur, et les transformations entre les niveaux 2 et 3 ainsi que 3 et 4 sont relativement simples comme nous l'avons remarqué précédemment (le problème est plutôt de coder efficacement les primitives importantes). Le passage le plus délicat est celui de 1 à 2, puisqu'il implique des prises de décisions et des compromis sur les précisions à attribuer. Notons par ailleurs que les niveaux 1 et 2 sont absolument identiques à ceux qui sont apparus dans notre analyse du développement de matériel.

Si les transformations de 2 à 4 sont réalisées correctement, les résultats de cette méthode simple et structurée sont excellents, comme l'a montré l'exemple d'implantation de l'algorithme de FFCT.

Puisque les niveaux 1 à 3 sont écrits dans un même langage, ils sont parfaitement compatibles. Ceci permet de mélanger des représentations diverses, et de tester exhaustivement les différentes étapes. De surcroît, le niveau 3 permet (à l'aide d'un interface idoine) de vérifier les résultats de simulation en assembleur (niveau 4) ou du processeur lui-même (niveau 5).

En conclusion, nous pensons que cette méthode structurée permet d'implanter avec profit des algorithmes très complexes sur des processeurs spécialisés. Non seulement la perte de performance est minime, mais surtout, la méthodologie ouvre la porte à l'implantation d'algorithmes dont la complexité ne permet pas d'envisager l'écriture d'une version en assembleur directement.

5.5 Principaux résultats du chapitre

Un algorithme nouveau de transformation de Fourier et en cosinus pour des longueurs en puissances de 2 a été introduit. Il utilise une division asymétrique (N/2 et N/4) du problème initial de dimension N. Cette méthode a ensuite été généralisée à toute une série de problèmes (séquences avec symétries, convolution réelle, transformations impaires). Il est à noter que pour tous les problèmes considérés, l'approche proposée conduit au nombre minimum connu d'opérations.

Une réalisation matérielle de l'algorithme a été faite. Il s'agit de la conception d'un circuit dédié réalisant une transformation en cosinus (8 points) pour le codage vidéo en temps réel. Dans ce contexte, une méthodologie de conception pour circuits intégrés de traitement du signal est brièvement développée.

Finalement, l'implantation logicielle (sur processeur de signal TMS 320) ainsi qu'une méthode simple et structurée pour transformer des algorithmes complexes en code machine efficace sont décrites.

Références bibliographiques du chapitre 5:

[ahm74] N.Ahmed, T.Natarajan, and K.R. Rao, "Discrete Cosine Transform", IEEE Trans. on Computers, Vol. C-23, pp.88-93, Jan. 1974.

[bon76] G.Bonnerot and M.G.Bellanger, "Odd-time Odd-frequency Discrete Fourier Transform for Symmetric Real-Valued Series", Proc. IEEE, Vol.64, 1976, pp.392-393.

[bra83] R.N.Bracewell, "Discrete Hartley Transform", J. Opt. Soc. Am., Vol.73, No.12, Dec.1983, pp.1832-1835.

[bra84] R.N.Bracewell, "The Fast Hartley Transform", Proceedings of the IEEE, Vol. 22, No.8, Aug.84, pp.1010-1018.

[bri74] E.O.Brigham, **The Fast Fourier Transform**, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1974.

[che77] W-H.Chen, C.H.Smith, and S.C.Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Trans. on Communications, Vol. COM-25, pp.1004-1009, Sept. 1977.

[coo65] J.W.Cooley, and J.W.Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Math. of Comput., Vol.19, pp.297-301, April 1965.

[dij76] E.W.Dijkstra, "**A discipline of programming**", Prentice-Hall, 1976.

[duh84a] P.Duhamel, and H. Hollmann, "Split-Radix FFT Algorithms", Electronics Letters, Vol.20, No.1, 5th Jan. 1984.

[duh84b] P.Duhamel, and H. Hollmann, "Existence of a 2^n FFT Algorithm with a Number of Multiplications lower than 2^{n+1}", Electronics Letters, Vol.20, No.17, pp.690-692, Aug. 1984.

[duh85a] P.Duhamel, and H. Hollmann, "Implementation of 'Split-Radix' FFT Algorithms for Complex, Real and Real-Symmetric Data", Proceedings of the IEEE Intl. Conf. on ASSP,Tampa, March 1985.

[duh85b] P.Duhamel, "*Procédé et Dispositif de Transformée en Cosinus Utilisant des* Convolueurs Rapides", Projet de demande de brevet, CNET, 1985.

[duh86] P.Duhamel, and M.Vetterli, "Cyclic Convolution of Real Sequences: Hartley versus Fourier and new Schemes", à paraître dans Proc. of the 1986 IEEE Intl. Conf. on ASSP, Tokyo, April 1986.

[gut84] F.Guterl, "Testing", IEEE Spectrum, Sept.1984, pp.40-46.

[har42] R.V.L.Hartley, "A More Symmetrical Fourier Analysis Applied to Transmission Problems", Proc. IRE 30, pp.144-150, 1942.

[hei85a] M.T.Heideman, and C.S.Burrus, "Multiply/Add Tradeoffs in Length-2^n FFT Algorithms", Proc. IEEE Conf. on ASSP, Tampa, Fl, 1985, pp.780-783.

[hei85b] M.T.Heideman, Personal communication, April 1985.

[hwa79] K.Hwang, "**Computer Arithmetic, Principles, Architecture, and Design**", J.Wiley & Sons, 1979.

[kar85] M.Kardan, "Implantation d'une FFT rapide sur le TMS 320", Projet de semestre, Laboratoire d'Informatique Technique, EPFL,1985.

[lig86] A.Ligtenberg, M.Vetterli, and J.H.O'Neill, "MOVAL: A Framework for Turning Digital Signal Processing Algorithms into Custom Chips", à paraître dans Signal Processing, June 1986.

[mca72] G.K.McAuliffe, "Fourier Digital Filter or Equalizer and Method of Operation Therefore", US Patent No. 3 679 882, July 25, 1972.

[mar84] J.B.Martens, "Recursive Cyclotomic Factorisation - A New Algorithm for calculating the Discrete Fourier Transform", IEEE Trans. on ASSP, Vol. ASSP-32, N0.4, Aug.1984.

[mas85] J.Masson, and Z.Picel, "Flexible Design of Computationaly Efficient Nearly Perfect QMF Filter Banks", Proc. 1985 IEEE Conf. on ASSP, Tampa, March 1985.

[mea80] C. Mead, and L. Conway, **Introduction to VLSI**, Addison-Wesley, 1980.

[mor77] L.R.Morris, "Automatic Generation of Time Efficient Digital Signal Processing Software", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-25, pp.74-78, Feb. 1977.

[mor78] L.R.Morris, "A Comparative Study of Time Efficient FFT and WFTA Programs for General Purpose Computers", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-26, pp.141-150, April 1978.

[nar78] M.J.Narasimha, and A.M.Peterson, "On the Computation of the Discrete Cosine Transform", IEEE Trans. on Communications, Vol. COM-26, pp.934-936, June 1978.

[nas83] N. Nasrabadi, and R. King, "Computationally Efficient Discrete Cosine Transform Algorithm", Elec. Letters, 6th Jan. 1983, Vol.19, No. 1, pp. 24-25.

[naw84] H.Nawab, J.H.McClellan, "Bounds on the Minimum Number of Data Transfers in WFTA and FFT Programs", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-27, pp.394-398, Aug. 1979.

[nus78] H.J. Nussbaumer, and P. Quandalle, "Computation of Convolutions and Discrete Fourier Transforms by Polynomial Transforms", IBM J. Res. Dev., Vol. 22, pp. 134-144, 1978.

[nus79] H.J. Nussbaumer, and P. Quandalle, "Fast Computation of Discrete Fourier Transforms Using Polynomial Transforms", IEEE Trans. on ASSP, Vol. ASSP-27, pp. 169-181, 1979.

[nus81] H.J.Nussbaumer, "Pseudo QMF Filter Bank", IBM Technical Disclosure Bulletin, Vol.24, No.6, pp 3081-3087, Nov.1981.

[nus82] H.J.Nussbaumer, **Fast Fourier Transform and Convolution Algorithms**, Springer, Berlin, 1982.

[nus84a] H.J.Nussbaumer, and M.Vetterli, "Computationally Efficient QMF Filter Banks", Proc. 1984 Int. IEEE Conf. on ASSP, San Diego, March 1984.

[nus84b] H.J.Nussbaumer, and M.Vetterli, "Pseudo Quadrature Mirror Filters", Proc. of Int. Conf. on Digital Signal Processing, Florence, Sept. 1984.

[opp75] A.V.Oppenheim, and R.W.Schafer, **Digital Signal Processing**, Prentice-Hall, Englewood Cliffs, 1975.

[pei84] S-C. Pei, and E-F. Huang, "Improved 2D Discrete Cosine Transforms Using Generalized Polynomial Transforms and DFT's", Proc. IEEE Int. Conf. on Communications, Amsterdam, 1984, pp. 242-244.

[rad76] C.M.Rader, and N.M.Brenner, "A New Principle for Fast Fourier Transformation", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-24, pp.264-265, June 1976.

[red80] N.S. Reddy, and V.U. Reddy, "High-Speed Computation of Autocorrelation Using Rectangular Transforms", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-28, No. 4, pp.481-483, Aug. 1980.

[sie75] A.E. Siegman, "How to Compute Two Complex Even Fourier Transforms with One Transform Step", Proceedings of the IEEE, March 1975, pp. 544.

[tx83] Texas Instruments, "TMS 32010 User's guide", Texas Inst. Co., Houston, 1983.

[vet84b] M.Vetterli, and H.J.Nussbaumer, "Simple FFT and DCT Algorithms with Reduced Number of Operations", Signal Processing, Vol.6, No.4, pp.267-278, Aug. 1984.

[vet85a] M.Vetterli, "Fast 2-D Discrete Cosine Transform", Proc. of the 1985 IEEE Intl. Conf. on ASSP, Tampa, March 1985.

[vet85b] M.Vetterli, and H.J.Nussbaumer, "Algorithmes de Transformée de Fourier et Cosinus Mono et Bidimensionnels", Annales des Telecommunications, Tome 40, Sept.- Oct. 1985, No. 9-10, pp.466-476.

[vet86a] M.Vetterli, and A.Ligtenberg, "A Discrete Fourier-Cosine Transform Chip", IEEE Trans. on Communications, Special Issue on VLSI in Telecommunications. January 1986.

[vet85d] M.Vetterli, "FFT's of Signals with Symmetries and Applications" Proc. of MELECON-85, Madrid, Oct. 1985.

[vet85e] M.Vetterli, E.Debourse, and M.Kardan, "Fast Fourier Transforms on the TMS 320", Proc. of the Journées d'Electronique 1985, EPFL, Lausanne, Oct. 1985, pp.193-204.

[wes81] N.Weste, "MULGA - An Interactive Symbolic Layout System for the Design of Integrated Circuits", Bell System Technical Journal, Vol.60, No.6, July-August 1981, pp.823-857.

[win80] S.Winograd, "Arithmetic Complexity of Computation", Society for Industrial and Applied Mathematics, 1980.

[Yav68] R.Yavne, "An Economical Method for Calculating the Discrete Fourier Transform", AFIPS Proc., Vol.33, pp.115-125, 1968 Fall Joint Computer Conf., Washington.

6 Conclusions

Une première partie se propose de survoler les résultats apportés par le travail présenté, alors qu'une seconde partie indique les développements ultérieurs possibles.

6.1 Principaux résultats

Rappelons que le premier chapitre a présenté, outre l'introduction au travail, un historique des travaux réalisés sur les bancs de filtres numériques et les transformations rapides utilisées en traitement numérique du signal. Cet historique visait à placer ce travail dans le contexte d'autres recherches dans le domaine.

Le second chapitre est consacré à l'analyse des bancs de filtres numériques. Il introduit un formalisme matriciel comme outil d'analyse puissant dans le cadre du traitement du signal par banc de filtres. Quoique également proposé par d'autres auteurs, ce formalisme a été étendu (entre autres, par la généralisation de la notion de banc de filtres polyphases), et utilisé par la suite pour tous nos développements dans le cadre de ce travail. A l'aide de cet outil, nous avons analysé les bancs de filtres utilisés en codage en sous-bandes (division puis reconstruction du signal) et ceux utilisés dans les transmultiplexeurs (multiplexage, puis séparation des signaux). Toute une série de propriétés, dont certaines sont fondamentales, ont pu être démontrées, en particulier sur la reconstruction parfaite, sur la reconstruction sans repliements spectraux (dans le codage en sous-bandes) et sur la reconstruction sans diaphonie (dans les transmultiplexeurs). La dualité entre codeurs en sous-bandes et transmultiplexeurs a été montrée, permettant ainsi d'unifier deux domaines d'application des bancs de filtres qui avaient été traités de façon séparée jusqu'ici. L'intérêt du formalisme matriciel réside dans le fait que des questions qui étaient jusqu'ici restées ouvertes (car trop complexes) se réduisent maintenant à des propriétés simples de matrices de filtres (comme par exemple la forme du déterminant de la matrice de filtres).

Le troisième chapitre, intitulé "synthèse de bancs de filtres", applique les résultats dérivés dans le chapitre précédent à la conception de bancs de filtres. On considère d'abord la conception de filtres pour l'utilisation dans un banc de filtres, un problème soumis à des contraintes différentes de celles qui régissent la conception classique de filtres. Pour le cas de bancs de filtres RIF, on propose des méthodes analytiques pour calculer des filtres d'analyse et de synthèse RIF qui permettent une reconstruction parfaite, ainsi qu'une méthode d'optimisation souple. Ensuite, on considère différents bancs de filtres, en particulier dans le cas où les filtres sont obtenus par modulation à partir d'un seul filtre prototype. Le cas des filtres pseudo-QMF (extension du concept QMF au cas N>2) et celui des filtres QMF

complexes (avec reconstruction parfaite) sont présentés, ainsi que l'extension au cas bidimensionnel.

Le chapitre 4 traite de la complexité de calcul liée aux bancs de filtres. L'évaluation d'arbres de filtres est d'abord présentée, tant dans le domaine temporel que dans le domaine fréquentiel (où certaines améliorations sont obtenues). La charge de calcul des bancs de N filtres est ensuite considérée, et ceci en tirant parti du sous-échantillonnage et de la modulation qui caractérisent la plupart de ces bancs de N filtres. A nouveau, on compare les méthodes dans le domaine temporel à celles dans le domaine fréquentiel (qui s'avèrent très efficaces).

Le dernier chapitre est consacré aux transformations rapides dont on a vu l'importance tant dans les méthodes de convolution par transformée que dans l'évaluation des bancs de filtres modulés. Un nouvel algorithme de transformation de Fourier et en cosinus, appelé algorithme de FFCT, est d'abord présenté. Cet algorithme est ensuite généralisé à toute une série de problèmes (pour lesquels il donne un nombre minimum d'opérations), puis deux exemples d'implantations, la première matérielle (conception d'un circuit VLSI réalisant une transformation en cosinus à débit vidéo) et la seconde logicielle (code efficace de transformée pour processeurs de signaux), sont données. Les méthodologies générales qui ont été utilisées afin de réaliser ces implantations particulières sont également décrites. Quoique les résultats de ce chapitre aient été utilisés lors de la discussion de la complexité de calcul au chapitre 4, il n'en reste pas moins vrai qu'ils dépassent le cadre des bancs de filtres et c'est la raison pour laquelle ils ont été présentés de façon relativement autonome.

6.2 Développements ultérieurs

Evidemment, ce travail n'épuise absolument pas le sujet des bancs de filtres, et nous pensons au contraire qu'il ouvre un bon nombre de possibilités nouvelles. Notre propos aura plutôt été de poser un contexte formel clair et de dériver des résultats de base que de réaliser des conceptions ou des implantations de bancs de filtres réels. De ce point de vue, nous pensons que notre analyse des bancs de filtres (avec le formalisme matriciel qui a été introduit et les résultats fondamentaux qui ont été dérivés) présente une base adéquate pour des travaux futurs.

La synthèse de bancs de filtres, avec entre autres, la conception de filtres dans ce contexte, reste beaucoup plus ouverte. Car si nous avons montré que l'espace des possibilités était plus grand que celui exploré jusqu'alors (par exemple, le compromis possible entre qualité des filtres et qualité de la reconstruction), il nous semble que seules des applications réelles indiqueront les choix à prendre. La conception de filtres RII, qui n'ont pas été implantés dans le cadre de ce travail, semble potentiellement intéressante. Enfin, une exploration plus poussée des effets de

quantification (dans les codeurs en sous-bandes) et de transmission non-idéale (dans les transmultiplexeurs) ainsi que des implications qui en résultent pour la conception de filtres serait également utile.

En ce qui concerne la complexité de calcul des bancs de filtres, nous avons considéré les cas généraux importants, et ceci sans entrer dans trop de détails relatifs à des cas concrets. Une étude détaillée des cas concrets est évidemment nécessaire, mais spécifique aux applications particulières, et c'est une des raisons pour lesquelles nous l'avons laissée ouverte. Un travail plus important consisterait à mettre en relation les algorithmes efficaces pour banc de filtres avec les méthodes traditionnelles de filtrage (par transformée et "overlap and add" par exemple). Finalement, on pourrait dériver des complexités théoriques pour le filtrage par banc de filtres.

Enfin, dans le cadre des transformées rapides présentées au dernier chapitre, la transformation des gains théoriques (sur le nombre d'opérations) en gains pratiques (réduction de la surface de silicium ou du temps d'exécution sur un processeur réel) est certainement un des sujets les plus importants et ouverts en traitement du signal. De ce fait, les méthodologies d'implantation présentées brièvement dans ce travail mériteraient des considérations plus approfondies.

Avec ce travail, nous espérons avoir contribué à la clarification du sujet des bancs de filtres, et de motiver ainsi des travaux futurs dans le domaine.

Annexes

Annexe A2.1: Diagonalisation de matrices de Toeplitz et de Hankel circulantes

Dans le paragraphe 2.3, nous avons vu l'importance des matrices circulantes du type Toeplitz et Hankel. Quoique le résultat de cette annexe soit évident puisqu'il se base sur la propriété de convolution de la transformée de Fourier, nous le dérivons ci-après car il est moins familier dans un contexte matriciel.

Rappelons qu'une matrice de Toeplitz a des éléments identiques le long de ses diverses diagonales, alors qu'une matrice de Hankel a des éléments identiques le long des antidiagonales (ce qui la rend évidemment symétrique).

Maintenant, si les matrices sont de surcroît circulantes, la différence entre une matrice de Toeplitz et de Hankel ayant une première ligne identique se réduit au sens dans lequel les éléments se déplacent d'une ligne à l'autre: dans le cas Toeplitz, ce sens est de gauche à droite, alors qu'il est de droite à gauche dans le cas Hankel. Dès lors, ces deux matrices sont liées par une permutation des lignes donnée par la matrice J définie en (2.48). La pré-multiplication par J s'interprète comme une permutation de la ligne i avec la ligne N-i (la ligne 0 restant inchangée), la post-multiplication ayant le même effet sur les colonnes. Appelons $\mathbf{T}$ une matrice de Toeplitz circulante avec les éléments $[a_0 a_1 \ldots a_{N-1}]$ sur la première ligne et H une matrice de Hankel avec cette même ligne initiale. Alors:

$$\mathbf{T} = \mathbf{J} \cdot \mathbf{H} \qquad \text{et} \qquad \mathbf{H} = \mathbf{J} \cdot \mathbf{T} \tag{A2.1}$$

Maintenant, une matrice de Toeplitz circulante correspond à une matrice de convolution circulaire. A cause de la propriété de convolution de la transformée de Fourier, nous pouvons dès lors diagonaliser $\mathbf{T}$ à l'aide de la matrice de Fourier $\mathbf{F}$ définie en (2.44). Il en résulte une matrice diagonale $\mathbf{D}$ donnée par:

$$\mathbf{D} = \mathbf{F} \cdot \mathbf{T} \cdot \mathbf{F}^{-1} \tag{A2.2}$$

où:

$$D = \text{Diag}[\ F \cdot [a_0 a_{N-1} \ldots a_1]^T\ ] \tag{A2.3}$$

Remarquons l'inversion des élements a_i avec a_{N-i}. La démonstration de la factorisation en (A2.2) se fait le plus aisément par la propriété de convolution. La convolution circulaire d'un vecteur x par une séquence $[a_0 a_{N-1} \ldots a_1]$ est représenté par:

$$y = T \cdot x \tag{A2.4}$$

où T correspond à la matrice de Toeplitz définie au-dessus de (A2.1). La relation (A2.4) correspond à la convolution dans le domaine temporel, alors que la convolution dans le domaine fréquentiel est donnée par:

$$F \cdot y = F \cdot T \cdot F^{-1} \cdot F \cdot x \tag{A2.5}$$

Il est bien connu que cette convolution fréquentielle est donnée par une multiplication point à point, donc que

$$F \cdot T \cdot F^{-1} = D \tag{A2.6}$$

ou bien:

$$T = F^{-1} \cdot D \cdot F \tag{A2.7}$$

La première ligne de la matrice T correspond à la réponse impulsionnelle inversée dans le temps du filtre. Puisque la convolution dans le domaine fréquentiel est une multiplication point à point des transformées de Fourier de x et du filtre, les éléments de la matrice diagonale D sont obtenus par transformée de Fourier de la séquence:

$$[a_0 a_{N-1} \ldots a_1] = [a_0 a_1 \ldots a_{N-1}] \cdot J \tag{A2.8}$$

où l'inversion dans le temps est exprimée par la permutation des éléments a_i avec a_{N-i}. En fait, en vertu de (2.56) et (A2.8), on peut également écrire la matrice D avec:

$$D = N \cdot \text{Diag}[\ F^{-1} \cdot [a_0 a_1 \ldots a_{N-1}]^T\ ] \tag{A2.9}$$

Des relations ci-dessus suivent les relations (2.100) à (2.104).

Annexe A2.2: Nécessité de filtres polyphases à phase minimale pour la stabilité de l'inverse d'une matrice de filtres modulés

La condition C2.3 pose que dans le cas des bancs de filtres modulés, il est nécessaire et suffisant que tous les zéros du déterminant se trouvent à l'intérieur du cercle unité afin que l'inverse de la matrice de filtres soit stable. Selon la relation (2.109), ceci équivaut à dire que les zéros des filtres polyphases doivent tous être à l'intérieur du cercle unité. Si le fait que cette condition soit suffisante est évident, il n'est pas aussi immédiat de voir qu'elle est également nécessaire. C'est ce qui est démontré ci-dessous. Comme les filtres de synthèse sont également modulés, il est suffisant d'analyser le filtre de base, $G_\Omega(z)$, donné d'après (2.107) comme:

$$G_\Omega(z) = D(z^N)\cdot [\ z^{-N}/N_{\Omega 0}(z^N) + z^{-N+1}/N_{\Omega 1}(z^N) + \ldots + z^{-1}/N_{\Omega N-1}(z^N)\ ] \qquad (A2.10)$$

En fait, et ceci afin de simplifier les notations, nous remarquons qu'il est suffisant d'analyser les pôles de la fonction suivante:

$$f(x) = 1/N_{\Omega N-1}(x^N) + x^{-1}/N_{\Omega N-2}(x^N) + \ldots + x^{-N+1}/N_{\Omega 0}(x^N) \qquad (A2.11)$$

A part un pôle en z=0, f(x) possède tous les pôles de $G_\Omega(z)$. Dans la suite, nous admettrons qu'au moins un des polynômes $N_{\Omega i}(x^N)$ n'est pas un monôme (autrement, il s'agit d'un cas trivial où la taille des filtres d'analyse est égale au nombre de canaux, un cas où la stabilité ne présente d'ailleurs pas de problèmes). Nous allons montrer ci-dessous que les zéros des $N_{\Omega i}(x^N)$ correspondent bien aux pôles de f(x). Nous admettrons qu'il n'y a pas de pôles ou de zéros à l'origine, car ce cas correspond à des déplacements dans le temps uniquement. D'abord, considérons le cas où tous les zéros des $N_{\Omega i}(x^N)$ sont différents (donc également simples). Récrivons $N_{\Omega i}(x^N)$ de la façon suivante:

$$N_{\Omega i}(x^N) = \prod_{j=0}^{k_i-1} \cdot \prod_{l=0}^{N-1} (W^l p_{ij} x^{-1} - 1) \qquad (A2.12)$$

où W est la N-ième racine de l'unité et k_i la longueur (divisée par N) du i-ième filtre polyphase. Une fonction rationnelle dont le degré du numérateur est strictement plus petit que celui du dénominateur possède une décomposition en fonctions élémentaires qui est unique. f(x) peut donc s'écrire de la façon suivante:

$$f(x) = \sum_{i=0}^{N-1} \sum_{j=0}^{k_i-1} \sum_{l=0}^{N-1} \frac{\alpha_{ijl}}{(W^l p_{ij} x^{-1} - 1)} \qquad \text{(A2.13)}$$

La première somme correspond aux N filtres polyphases, la seconde à la longueur de chacun de ceux-ci et la dernière aux N pôles disposés sur un cercle de rayon $|p_{ij}|$. Puisque tous les zéros sont différents, il ne peut pas y avoir d'annulation entre deux termes distincts, et il suffit donc de montrer que tous les α_{ijl} sont différents de zéros afin de prouver que les $W^l p_{ij}$ sont tous des pôles de f(x). On peut vérifier que l'expression pour les α_{ijl} est donnée par la relation suivante:

$$\alpha_{ijl} = \frac{[W^l \cdot p_{ij}]^i}{\prod_{\substack{m=0 \\ m\neq l}}^{N-1} \left([W^l p_{ij}]^{-1} \cdot W^m p_{ij} - 1 \right) \cdot \prod_{\substack{n=0 \\ n\neq j}}^{k_i-1} \left([p_{ij}]^{-N} \cdot [p_{il}]^N - 1 \right)} \qquad \text{(A2.14)}$$

Ayant admis que les p_{ij} sont tous différents de zéros et non-égaux entre eux, il s'ensuit de (A2.14) que tous les α_{ijl} sont différents de zéros et bornés. Dans ce premier cas, donc, tous les $W^l p_{ij}$ correspondent bien à des pôles de f(x).

Considérons maintenant le cas où les polynômes $N_{\Omega i}(x^N)$ peuvent avoir des zéros multiples, mais pas de zéros en commun entre eux. Dès lors, si un des zéros possède la multiplicité P, il faut ajouter dans (A2.13) P-1 termes avec le dénominateur correspondant au pôle multiple élevé aux puissances 2 jusqu'à P. En fait, rien n'est changé fondamentalement, et il est par exemple aisé de vérifier que la constante correspondant au dénominateur élevé à la puissance P est toujours différente de zéro, donc que le pôle multiple de $N_{\Omega i}(x^N)$ est aussi un pôle multiple de f(x).

En fait, le cas intéressant apparaît lorsque plusieurs $N_{\Omega i}(x^N)$ ont des zéros en commun, car ils pourraient alors s'annihiler entre eux par sommation. Que ceci ne peut pas être le cas sera montré ultérieurement, mais nous commençons par exclure un cas particulier. Admettons que tous les $N_{\Omega i}(x^N)$ sont identiques et égaux à $N_\Omega(x^N)$. f(x) peut donc s'écrire comme suit:

$$f(x) = \frac{1 + x^{-1} + \ldots + x^{-N+1}}{N_\Omega(x^N)} \qquad \text{(A2.15)}$$

Les N-1 zéros de f(x) sont les racines N-ième de l'unité différentes de x=1. Si $N_\Omega(x^N)$ a un pôle instable, il a forcément N pôles instables, donc f(x) ne peut en aucun cas être stable.

Considérons maintenant le cas d'un zéro commun, p_c, entre $N_{\Omega m}(x^N)$ et $N_{\Omega n}(x^N)$. Prenons les deux termes de f(x) qui contiennent ce zéro afin de montrer qu'il ne se simplifiera pas (les autres termes n'entrant pas en jeu dans une simplification éventuelle):

$$\frac{x^{-m}}{(p_c^N x^{-N} - 1)\cdot N'_{\Omega m}(x^N)} + \frac{x^{-n}}{(p_c^N x^{-N} - 1)\cdot N'_{\Omega n}(x^N)} \qquad \text{(A2.16)}$$

où $N'_{\Omega m}(x^N)$ et $N'_{\Omega n}(x^N)$ sont les polynômes après division par le facteur correspondant à p_c. Le numérateur n(x) de l'expression en (A2.16) est égal à:

$$n(x) = x^{-m} \cdot N'_{\Omega n}(x^N) + x^{-n} \cdot N'_{\Omega m}(x^N) \qquad \text{(A2.17)}$$

et afin que les pôles en $W^l p_c$ (l=0..N-1) se simplifient, il faut que les N équations suivantes soient vérifiées:

$$n(W^l \cdot p_c) = 0 \qquad l=0..N-1 \qquad \text{(A2.18a)}$$

$$W^{-lm} \cdot N'_{\Omega n}(p_c^N) + W^{-ln} N'_{\Omega m}(p_c^N) = 0 \qquad l=0..N-1 \qquad \text{(A2.18b)}$$

Etant donné que m≠n et en raison de l'orthogonalité des racines de l'unité, il est aisé de vérifier que par combinaison linéaire adéquate des N équations en (A2.18b), on obtient les deux équations équivalentes suivantes:

$$N'_{\Omega m}(p_c^N) = 0 \qquad \text{(A2.19a)}$$

$$N'_{\Omega n}(p_c^N) = 0 \qquad \text{(A2.19b)}$$

Les $W^l p_c$ doivent donc également être des zéros des polynômes réduits, et par récursion, il s'ensuit que $N_{\Omega m}(p_c^N)$ et $N_{\Omega n}(p_c^N)$ doivent être égaux afin que les équations dans (A2.18) restent vérifiées de proche en proche. Or, nous avons vu précédemment qu'alors (voir (A2.15)), il reste en tous cas un pôle qui ne peut être simplifié. Si celui-ci est instable, f(x) est instable. Evidemment, la démonstration reste valable si p_c est commun à plus de deux polynômes. Dans ce cas, il y a simplement un nombre correspondant d'équations dans (A2.18-39). Il a donc été prouvé qu'un zéro de $N_{\Omega i}(x^N)$ est forcément un pôle de f(x). Ceci démontre qu'il est nécessaire que les inverses des filtres polyphases soient des filtres stables afin d'obtenir une reconstruction parfaite dans le cas de bancs de filtres modulés.

Annexe A2.3: Matrices de filtres orthogonales

Le calcul de l'inverse d'une matrice classique qui ne comporte que des éléments scalaires est en général une tâche ardue, sauf si la matrice est orthogonale, puisqu'alors la matrice inverse est simplement la transposée. Dans le cas du calcul de l'inverse d'une matrice de filtres, la situation est encore plus difficile: en plus de la difficulté de calculer l'inverse, il faut s'attendre en général à ce que les filtres correspondant à la matrice inverse soient beaucoup plus complexes que les filtres de la matrice de départ. Prenons un exemple simple: la matrice à inverser correspond à N filtres RIF de longueur M. Le déterminant correspond alors à un filtre RIF ayant une longueur de l'ordre de $N \cdot M$, et les cofacteurs à des filtres RIF de l'ordre de $(N-1) \cdot M$. Même si certains termes s'annulent, il n'en reste pas moins qu'en général les filtres correspondant à la matrice inverse sont nettement plus complexes (si N>2).

Pourtant, tout comme dans le cas de matrices classiques, si la matrice de filtres est orthonormale, la matrice inverse peut aisément être calculée par simple transposition. De surcroît, les filtres correspondants ont alors la même complexité pour l'inverse que pour la matrice de départ. Définissons d'abord, de la même façon que pour des élements scalaires, l'orthogonalité de deux vecteurs:

Définition D2.20: Deux vecteurs de fonctions rationnelles en z^{-1} sont orthogonaux si leur produit scalaire est identiquement nul.

Notons le terme "identiquement nul", car le produit scalaire peut être zéro pour certaines valeurs de z sans pour autant que les vecteurs soient orthogonaux. Remarquons, afin d'éviter toute confusion, que ces vecteurs peuvent correspondre à deux filtres, mais que ces filtres n'ont rien à voir avec ce qui est habituellement appelé "filtres orthogonaux" dans la littérature. Définissons également une matrice orthogonale:

Définition D2.21: Une matrice de filtres $\mathbf{H}(z)$ est orthogonale si

$$\mathbf{H}(z) \cdot \mathbf{H}^T(z) = \alpha(z) \cdot \mathbf{I} \qquad \text{(A2.20a)}$$

On a alors immédiatement:

$$\mathbf{H}^T(z) \cdot \mathbf{H}(z) = \alpha(z) \cdot \mathbf{I} \qquad \text{(A2.20b)}$$

Des relations (A2.20a-b), il ressort que les lignes et les colonnes de H(z) sont des vecteurs orthogonaux entre eux ayant tous la même norme $\sqrt{\alpha}(z)$ (où norme signifie racine carrée du produit scalaire du vecteur par lui-même). Le facteur $\sqrt{\alpha}(z)$ est toléré dans cette définition, car une normalisation n'est pas toujours possible (par exemple pour des questions de stabilité).

Notons que le terme orthogonal est utilisé d'une façon moins étroite que ce qui est habituel en algèbre linéaire puisque nous tolérons des facteurs d'échelles. Ceci est lié au fait que la normalisation peut être problèmatique pour des questions de stabilité.

D'une façon similaire, une matrice avec lignes ou colonnes orthogonales peut être définie de la manière suivante:

Définition D2.22: Une matrice de filtres avec des lignes orthogonales se définit telle que:

$$H(z) \cdot H^T(z) = D'(z) \cdot I \qquad \text{(A2.21a)}$$

et une matrice avec des colonnes orthogonales telle que:

$$H^T(z) \cdot H(z) = D''(z) \cdot I \qquad \text{(A2.21b)}$$

où $D'(z)$ et $D''(z)$ sont des matrices diagonales.

La relation (A2.21a) est équivalente à (A2.21b) si et seulement si $D'(z)=D''(z)=\alpha(z)\cdot I$. Notons que cette équivalence est similaire à celle présentée à la suite de la relation (2.118).

Nous avons vu à la section précédente des filtres qui conduisent à des matrices orthogonales (2.122-2.124). Comme ceux-ci ne présentent que peu d'intérêt pratique, nous allons chercher à en concevoir de plus adaptés aux problèmes réels. Pour ce faire, nous allons considérer la matrice de filtres polyphases $H_p(z)$. En considérant (2.36) et (2.90), il est clair que l'inverse de $D_p(z^N)$ et de $I_d(z^N)$ ne présente pas de problème, puisque ces matrices sont déjà diagonales. De ce fait, l'essence du problème se trouve dans l'inversion de $N_p(z^N)$. Nous allons donc tenter de trouver des matrices de $N_p(z^N)$ possédant la propriété d'orthogonalité. Remarquons que $N_p(z^N)$ est une matrice générale de fonctions rationnelles en z^{-1} (et n'a donc pas une structure restrictive comme $H_p(z)$ ou $H_m(z)$).

Commençons par considérer des cas simples. Il est d'abord possible de générer des matrices orthogonales en z^{-1} à l'aide d'une matrice orthogonale de scalaires et d'un filtre polyphase prototype, et ceci de la façon suivante:

$$\mathbf{N}_p(z^N) = H_{0,0}(z^N) \cdot \mathbf{M} \tag{A2.22}$$

où:

$$\mathbf{M} \cdot \mathbf{M}^T = \mathbf{I} \tag{A2.23}$$

Notons que si la matrice **M** est complexe, alors la transposée est hermitienne. Avec la relation (A2.22), la matrice $H_p(z)$ devient:

$$H_p(z) = H_{0,0}(z^N) \cdot \mathbf{I}_d(z) \cdot \mathbf{M} \tag{A2.24}$$

Avec (2.111), définissons $G_p(z)$ par:

$$[\ G_p(z)\ ]^T = H_{0,0}(z^N) \cdot \mathbf{M}^T \cdot \mathbf{I}'_d(z) \cdot \mathbf{J} \tag{A2.25}$$

où $I'_d(z)$ est égal à $z^{-N} \cdot [I_d(z)]^{-1}$. Dans ces conditions, (2.115) devient:

$$H_p(z) \cdot [\ G_p(z)\ ]^T = = z^{-N} \cdot [H_{0,0}(z^N)]^2 \cdot \mathbf{J} \tag{A2.26}$$

De même, (2.117) devient:

$$[\ G_p(z)\ ]^T \cdot \mathbf{J} \cdot H_p(z) = = z^{-N} \cdot [H_{0,0}(z^N)]^2 \cdot \mathbf{I} \tag{A2.27}$$

Comme la formulation de $\mathbf{N}_p(z^N)$ par (A2.22) laisse peu de degrés de liberté, nous pouvons faire un choix moins contraignant avec des lignes ou des colonnes orthogonales. Introduisons la matrice diagonale suivante:

$$H_d(z^N) = \text{Diag}[\ H_{0,0}(z^N), H_{0,1}(z^N), \ldots H_{0,N-1}(z^N)\ ] \tag{A2.28}$$

Si $H_p(z)$ et $G_p(z)$ sont définis par:

$$H_p(z) = \mathbf{I}_d(z) \cdot H_d(z^N) \cdot \mathbf{M} \tag{A2.29a}$$

$$[\ G_p(z)\ ]^T = \mathbf{M}^T \cdot H_d(z^N) \cdot \mathbf{I}'_d(z) \cdot \mathbf{J} \tag{A2.29b}$$

alors, la relation (2.115) devient:

$$H_p(z) \cdot [\ G_p(z)\ ]^T = = z^{-N} \cdot [H_d(z^N)]^2 \cdot J \qquad \text{(A2.30)}$$

ce qui revient à dire que $H_p(z)$ possède des lignes orthogonales. D'une façon similaire, si:

$$H_p(z) = I_d(z) \cdot M \cdot H_d(z^N) \qquad \text{(A2.31a)}$$

$$[\ G_p(z)\ ]^T = H_d(z^N) \cdot M \cdot I'_d(z) \cdot J \qquad \text{(A2.31b)}$$

la relation (2.117) devient:

$$H_p(z) \cdot [\ G_p(z)\ ]^T = = z^{-N} \cdot [H_d(z^N)]^2 \cdot I \qquad \text{(A2.32)}$$

c'est-à-dire que $H_p(z)$ possède des colonnes orthogonales.

En résumé, $H_p(z)$ défini par (A2.24) est une matrice de filtres orthogonale, alors que les $H_p(z)$ définis par (A2.29a) et (A2.31a) sont des matrices de filtres avec lignes ou colonnes orthogonales respectivement.

Si les matrices introduites ci-dessus ont bien la propriété d'orthogonalité, elles restent néanmoins assez limitées dans leurs applications du fait des restrictions qu'elles exigent. Nous allons donc chercher des solutions plus générales. Pour N=2, il est aisé de définir une classe plus générale de matrices orthogonales.

Prenons $N_p(z^2)$ égal à:

$$N_p(z^2) = \begin{matrix} H_{0,0}(z^2) & H_{0,1}(z^2) \\ H_{1,0}(z^2) & H_{1,1}(z^2) \end{matrix} \qquad \text{(A2.33)}$$

$N_p(z^2)$ est en quelque sorte une rotation élémentaire, par analogie avec le cas de matrices de scalaires où $H_{0,0}(z^2)$ est le cosinus d'un angle α et $H_{0,1}(z^2)$ est le sinus correspondant. Dès lors, on peut vérifier que:

$$N_p(z^2) \cdot [\ N_p(z^2)\ ]^T = [\ N_p(z^2)\ ]^T \cdot N_p(z^2)$$

$$= [\ H^2_{0,0}(z^2) + H^2_{0,1}(z^2)\ ] \cdot I \qquad \text{(A2.34)}$$

Donc, choisissons:

$$H_p(z) = I_d(z) \cdot N_p(z^2) \tag{A2.35a}$$

$$[\ G_p(z)\ ]^T = [\ N_p(z^2)\ ]^T \cdot I_d(z) \tag{A2.35b}$$

où nous avons tenu compte, dans (A2.32b), que J=I pour N=2. Il en résulte que:

$$H_p(z) \cdot [\ G_p(z)\ ]^T = [\ G_p(z)\ ]^T \cdot H_p(z)$$

$$= z^{-2} \cdot [\ H^2_{0,0}(z^2) + H^2_{0,1}(z^2)\ ] \cdot I \tag{A2.36}$$

Outre la matrice de "rotation" donnée en (A2.33), la matrice de "symétrie":

$$N_p(z^2) = \begin{matrix} 0 & \beta(z^2) \\ \beta(z^2) & 0 \end{matrix} \tag{A2.37}$$

où $\beta(z)$ est arbitraire, est également orthogonale au sens de la définition D2.21.

Les deux types de matrices données par (A2.33) et (A2.37) ainsi que leurs combinaisons couvrent le spectre des matrices orthogonales possibles pour N=2.

Pour N>2, les matrices orthogonales génériques sont plus complexes à dériver, et nous ne donnerons donc pas de classe générale comme nous l'avons fait ci-dessus. Notons qu'il est relativement aisé de trouver des matrices avec des lignes (ou des colonnes) orthogonales, c'est-à-dire ayant la propriété:

$$N_p(z^N) \cdot [\ N_p(z^N)\ ]^T = \mathrm{Diag}[\ \alpha_0(z^N), \alpha_1(z^N), \ldots\ \alpha_{N-1}(z^N)\ ] \tag{A2.38}$$

Cependant, la normalisation qui permet d'obtenir en même temps des colonnes orthogonales est difficile. En général, elle passe par la recherche du plus petit commun multiple de $\{\alpha_0(z^N), \alpha_1(z^N), \ldots \alpha_{N-1}(z^N)\}$, que nous appellerons $\alpha(z^N)$. Ensuite, il est nécessaire et suffisant de multiplier la ligne i de $N_p(z^N)$ par le facteur $\alpha(z^N)/\alpha_i(z^N)$ afin d'obtenir une matrice orthogonale $N'_p(z^N)$, qui satisfait alors:

$$\mathbf{N}'_p(z^N) \cdot [\, \mathbf{N}'_p(z^N) \,]^T = [\, \mathbf{N}'_p(z^N) \,]^T \cdot \mathbf{N}'_p(z^N)$$

$$= \alpha(z^N) \cdot \mathbf{I} \qquad \text{(A2.39)}$$

Remarquons que cette normalisation diffère sensiblement de ce qui est habituel avec des matrices à éléments scalaires. Dans ce cas, on divise simplement le vecteur $\mathbf{v}$ (qui représente la ligne i transposée) par la racine carrée de $\mathbf{v}^T \cdot \mathbf{v}$. Ceci n'est en général pas possible dans le cas de matrices de fonctions rationnelles en z^{-1}, puisqu'alors il faudrait que $\mathbf{v}^T(z) \cdot \mathbf{v}(z)$ soit un carré parfait et ait tous ses zéros à l'intérieur du cercle unité afin de pouvoir normaliser $\mathbf{v}(z)$. Il est clair que ceci est un cas plutôt rare (mais possible). En fait, la normalisation dans le cas de fonctions rationnelles de z^{-1} ressemble fort à ce que l'on serait amené à faire pour dériver des matrices orthogonales sur des corps autres que R ou C, par exemple sur Q ou Z.

Pour conclure cette section sur les espaces orthogonaux, nous allons introduire une classe de matrices qui ne sont pas elles-mêmes orthogonales, mais font appel à ces matrices pour leur factorisation.

Définition D2.23: Une matrice $\mathbf{H}(z)$ factorisable en matrices orthogonales et diagonales est une matrice de la forme:

$$\mathbf{H}(z) = \mathbf{M}(z) \cdot \mathbf{D}(z) \cdot \mathbf{M}^T(z) \qquad \text{(A2.40)}$$

où $\mathbf{M}(z)$ est une matrice orthogonale au sens de la définition D2.21 et $\mathbf{D}(z)$ est une matrice diagonale.

$\mathbf{M}(z)$ est la matrice des vecteurs propres de $\mathbf{H}(z)$. Elle peut être constituée d'éléments scalaires uniquement, comme dans le cas lors de la factorisation de $\mathbf{J} \cdot \mathbf{H}_m(z)$ en (2.100) (où la transposée est hermitienne puisque la matrice $\mathbf{F}$ est complexe). L'avantage de la formulation (A2.40) est que l'inversion de $\mathbf{H}(z)$ se réduit à celle de $\mathbf{D}(z)$ uniquement. De surcroît, il est possible de tirer parti de la structure particulière de $\mathbf{H}(z)$ afin de réduire la complexité de calcul requise par le banc de filtres correspondant (en particulier lorsque $\mathbf{M}(z)$ est scalaire). Finalement, la notion de vecteur propre permet d'interprèter plus aisément certains phénomènes liés au traitement du signal par des bancs de filtres.

Annexe A3.1: Cas de bancs d'analyse avec canaux retardés, N=2

Admettons que le canal 1 soit retardé d'un échantillon à l'entrée (délai de z^{-1}) et que le canal 0 soit retardé d'un échantillon à la sortie. Alors, $\hat{X}_0(z)$ et $\hat{X}_1(z)$ sont égaux à:

$$\hat{X}_0(z) = 1/2\ [H_0(z)\ X(z) + H_0(-z)\ X(-z)]\ G_0(z)\ z^{-1} \qquad (a3.1)$$

$$\hat{X}_1(z) = 1/2\ [z^{-1}\ H_1(z)\ X(z) - z^{-1}\ H_1(-z)\ X(-z)]\ G_1(z) \qquad (a3.2)$$

Donc, $M_0(z)$ et $M_1(z)$ sont égaux à (voir aussi (2.23-25)):

$$\begin{matrix} M_0(z) \\ \\ M_1(z) \end{matrix} = 1/2 \begin{matrix} H_0(z) & z^{-1}\ H_1(z) & z^{-1}\ G_0(z) \\ \\ H_0(-z) & -z^{-1}\ H_1(-z) & G_1(z) \end{matrix} \qquad (a3.3)$$

et le déterminant devient:

$$\Delta_m(z) = 1/4\ [H_0(z)\ H_1(-z) + H_0(-z)\ H_1(z)]\ z^{-1} \qquad (a3.4)$$

$\Delta_m(z)$ est une fonction ne comportant que des puissances paires de z^{-1} si $H_0(z)$ et $H_1(z)$ des filtres RIF.

Considérons maintenant $H_0(z)\ H_1(-z)$. Si ce polynôme à des coefficients d'indices impairs arbitraires, mais un seul coefficient d'indice pair non-nul, alors la reconstruction parfaite ne pose aucun problème (en utilisant comme filtres de reconstruction les filtres RIF donnés par la relation (3.2)). Ceci est tout à fait équivalent à la condition exprimée en (3.17), simplement avec un délai de z^{-1} supplémentaire. En raison de cette similarité, le cas ci-dessus ne sera pas considéré spécialement par la suite.

Notons que si les deux filtres d'analyse sont de même longueur, alors une reconstruction parfaite n'est pas possible. Dans ce cas $\Delta_1 = h_{00} \cdot h_{10}$ et $\Delta_{2M-1} = h_{0M-1} \cdot h_{1M-1} \cdot (-1)^{M-1}$. Evidemment, à moins que l'un des quatre facteurs (h_{00}, h_{10}, h_{0M-1} ou h_{1M-1}) soit égal à zéro, les deux termes Δ_1 et Δ_{2M-1} sont différents de zéro, donc $\Delta_m(z)$ ne peut être un monôme. Notons toutefois que Δ_1 et Δ_{2M-1} sont en général très petits.

Annexe A3.2: Solution à phase linéaire

Nous montrons ci-dessous dans quelles conditions une solution avec reconstruction parfaite existe pour des filtres à phase linéaire. Ces solutions sont utiles, car il est souvent désirable d'avoir des signaux de canaux qui soient en phase l'un par rapport à l'autre. Un filtre à phase linéaire est un filtre RIF dont la réponse impulsionnelle est soit symétrique ou antisymétrique [rab75]. Introduisons la fonction sym[H(z)] définie comme suit:

$$\begin{aligned} \text{sym}[H(z)] &= 1 \text{ si } H(z) \text{ a une réponse impulsionnelle symétrique} \\ &= -1 \text{ si } H(z) \text{ a une réponse impulsionnelle antisymétrique} \\ &= 0 \text{ si } H(z) \text{ a une réponse impulsionnelle sans symétrie aucune} \end{aligned} \tag{a3.5}$$

Cette fonction a les propriétés suivantes:

$$\text{sym}[H(z)G(z)] = \text{sym}[H(z)]\ \text{sym}[G(z)] \tag{a3.6}$$

$$\begin{aligned} \text{sym}[H(-z)] &= \text{sym}[H(z)] \text{ si } H(z) \text{ est de longueur impaire} \\ &= -\text{sym}[H(z)] \text{ si } H(z) \text{ est de longueur paire} \end{aligned} \tag{a3.7}$$

Lorsque deux filtres de même longueur sont additionnés, la symétrie est conservée si tous deux possèdent la même symétrie, mais disparaît autrement (symétrie différente ou absence de symétrie). Ce fait est explicité par la relation suivante:

$$\text{sym}[H(z)+G(z)] = \text{sym}[H(z)]\ \text{sym}[G(z)]\ 1/2\ (\text{sym}[H(z)]+\text{sym}[G(z)]) \tag{a3.8}$$

Si les longueurs sont différentes, la symétrie n'est en général pas préservée. A l'aide de cette fonction, nous allons analyser le déterminant donné en (3.4) dans le cas où $H_0(z)$ et $H_1(z)$ sont tous deux des filtres à phase linéaire de longueur M. Notons que nous ne considérons pas des filtres ayant des longueurs différentes, puisqu'alors la propriété de signaux de canaux en phase est de toute façon perdue. Considérons le produit $H_0(z) \cdot H_1(-z)$. Si M est impair et à cause de (a3.6-7), nous avons:

$$\text{sym}[H_0(z)H_1(-z)] = \text{sym}[H_0(z)]\ \text{sym}[H_1(z)] \tag{a3.9}$$

Puisque le filtre produit est de longueur impaire également, il s'ensuit que:

$$\text{sym}[H_0(z)H_1(-z)] = \text{sym}[H_0(-z)H_1(z)] \tag{a3.10}$$

Donc que:

$$\text{sym}[\Delta_m(z)] = \text{sym}[H_0(z)]\ \text{sym}[H_1(z)] \tag{a3.11}$$

Le centre de symétrie de Δ_m correspond au coefficient avec l'indice M-1, c'est-à-dire un nombre pair. Mais puisque $\Delta_m(z)$ est une fonction impaire de z, ce coefficient est forcément zéro. Tous les coefficients non-nuls de $\Delta_m(z)$ apparaissent doubles en raison de la symétrie, donc $\Delta_m(z)$ ne pourra jamais être un monôme.

Si M est un nombre pair, nous avons la relation suivante:

$$\text{sym}[H_0(z)H_1(-z)] = (-1)\ \text{sym}[H_0(z)]\ \text{sym}[H_1(z)] \tag{a3.12}$$

Puisque le filtre produit est à nouveau de longueur impaire, il s'ensuit que:

$$\text{sym}[H_0(z)H_1(-z)] = \text{sym}[H_0(-z)H_1(z)] \tag{a3.13}$$

ou bien que:

$$\begin{aligned}\text{sym}[\Delta(z)] &= \text{sym}[H_0(z)H_1(-z)] \\ &= (-1)\ \text{sym}[H_0(z)]\ \text{sym}[H_1(z)]\end{aligned} \tag{a3.14}$$

Le centre de symétrie est le coefficient avec l'indice M-1, un nombre impair. Tous les coefficients de $\Delta_m(z)$ à part celui-là vont apparaître deux fois. Donc, afin que le déterminant se réduise à un monôme, il est nécessaire et suffisant que le coefficient d'indice M-1 soit différent de zéro, tous les autres étant nuls. Il en résulte que $\Delta_m(z)$ doit être symétrique (quoique réduit à un seul coefficient). Ceci entraîne que $H_0(z)$ et $H_1(-z)$ ont la même symétrie, et, puisque M est pair et en vertu de (a3.7), que les deux filtres $H_0(z)$ et $H_0(z)$ sont de symétrie différente. Typiquement, le filtre $H_0(z)$ sera passe-bas et symétrique, alors que $H_1(z)$ sera passe-haut et antisymétrique.

En conclusion, l'ensemble des solutions à reconstruction parfaite et n'utilisant que des filtres RIF à phase linéaire a les caractéristiques suivantes:
- la longueur des filtres doit être un nombre pair
- les deux filtres doivent avoir une symétrie différente

Annexe A5.1: Code linéaire pour petites transformées

Cette annexe donne le code linéaire PASCAL pour des transformées de Fourier et en cosinus de longueur 4 et 8 (d'autres longueurs sont à disposition). Pour une transformation de longueur N, on utilise N+1 places mémoires. L'ordre des échantillons de sortie est indiqué à la fin. Ce code a été généré automatiquement à partir d'une version PASCAL récursive de l'algorithme de FFCT, et il a été utilisé pour générer le code assembleur pour TMS 320.

```
procedure dft4;
var   x0,      x1,      x2,      x3,      x4        : real;
begin
(* 1     a1     *)    x4      :=  x0     +    x2    ;
(* 2     a2     *)    x2      :=  x0     -    x2    ;
(* 3     a3     *)    x0      :=  x1     +    x3    ;
(* 4     a4     *)    x3      :=  x1     -    x3    ;
(* 5     a5     *)    x1      :=  x4     +    x0    ;
(* 6     a6     *)    x0      :=  x4     -    x0    ;
(* ordering:     1     2     0     3    *)
end;

procedure dct4;
const PI    = 3.141592654;
      c1d4  = cos(PI/4);
      c1d8 = cos(PI*1/8);
      cps1d8 = cos(PI*1/8) + sin(PI*1/8);
      smc1d8 = sin(PI*1/8) - cos(PI*1/8);
var  x0,      x1,      x2,      x3,      x4        : real;
begin
(* 1     a1     *)    x4      :=  x0     +    x3    ;
(* 2     a2     *)    x3      :=  x0     -    x3    ;
(* 3     a3     *)    x0      :=  x2     +    x1    ;
(* 4     a4     *)    x1      :=  x2     -    x1    ;
(* 5     a5     *)    x2      :=  x4     +    x0    ;
(* 6     a6     *)    x0      :=  x4     -    x0    ;
(* 7     m1     *)    x0      :=  x0     *    c1d4;
(* 8     a7     *)    x4      :=  x3     +    x1    ;
(* 9     m2     *)    x4      :=  x4     *    c1d8;
(* 10    m3     *)    x1      :=  x1     *    cps1d8;
(* 11    a8     *)    x1      :=  x4     -    x1    ;
(* 12    m4     *)    x3      :=  x3     *    smc1d8;
(* 13    a9     *)    x3      :=  x4     +    x3    ;
(* ordering:     2     1     0     3    *)
end;

procedure dft8;
const PI    = 3.141592654;
      c1d4  = cos(PI/4);
var  x0    , x1    , x2    , x3    , x4    , x5    , x6    , x7  ,  x8  : real
begin
(* 1     a1     *)    x8      :=  x0     +    x4    ;
(* 2     a2     *)    x4      :=  x0     -    x4    ;
(* 3     a3     *)    x0      :=  x2     +    x6    ;
(* 4     a4     *)    x6      :=  x2     -    x6    ;
(* 5     a5     *)    x2      :=  x8     +    x0    ;
(* 6     a6     *)    x0      :=  x8     -    x0    ;
```

```
(* 7     a7     *)    x8      :=  x1     +    x7    ;
(* 8     a8     *)    x7      :=  x1     -    x7    ;
(* 9     a9     *)    x1      :=  x5     -    x3    ;
(* 10    a10    *)    x3      :=  x5     +    x3    ;
(* 11    a11    *)    x5      :=  x8     +    x3    ;
(* 12    a12    *)    x3      :=  x8     -    x3    ;
(* 13    m1     *)    x3      :=  x3     *    c1d4;
(* 14    a13    *)    x8      :=  x7     +    x1    ;
(* 15    a14    *)    x1      :=  x7     -    x1    ;
(* 16    m2     *)    x1      :=  x1     *    c1d4;
(* 17    a15    *)    x7      :=  x2     +    x5    ;
(* 18    a16    *)    x5      :=  x2     -    x5    ;
(* 19    a17    *)    x2      :=  x4     +    x3    ;
(* 20    a18    *)    x3      :=  x4     -    x3    ;
(* 21    a19    *)    x4      :=  x1     +    x6    ;
(* 22    a20    *)    x6      :=  x1     -    x6    ;
(* ordering:     7     2     0     3     5     4     8     6     *)
end;

procedure dct8;
const PI     = 3.141592654;
      c1d4   = cos(PI/4);
      c1d16  = cos(PI*1/16);
      cps1d16 = cos(PI*1/16) + sin(PI*1/16);
      smc1d16 = sin(PI*1/16) - cos(PI*1/16);
      c2d16  = cos(PI*2/16);
      cps2d16 = cos(PI*2/16) + sin(PI*2/16);
      smc2d16 = sin(PI*2/16) - cos(PI*2/16);
      c3d16  = cos(PI*3/16);
      cps3d16 = cos(PI*3/16) + sin(PI*3/16);
      smc3d16 = sin(PI*3/16) - cos(PI*3/16);
var x0    , x1    , x2    , x3    , x4    , x5    , x6    , x7  ,  x8  : real;
begin
(* 1     a1     *)    x8      :=  x0     +    x7    ;
(* 2     a2     *)    x7      :=  x0     -    x7    ;
(* 3     a3     *)    x0      :=  x4     +    x3    ;
(* 4     a4     *)    x3      :=  x4     -    x3    ;
(* 5     a5     *)    x4      :=  x8     +    x0    ;
(* 6     a6     *)    x0      :=  x8     -    x0    ;
(* 7     a7     *)    x8      :=  x2     +    x1    ;
(* 8     a8     *)    x1      :=  x2     -    x1    ;
(* 9     a9     *)    x2      :=  x5     -    x6    ;
(* 10    a10    *)    x6      :=  x5     +    x6    ;
(* 11    a11    *)    x5      :=  x8     +    x6    ;
(* 12    a12    *)    x6      :=  x8     -    x6    ;
(* 13    m1     *)    x6      :=  x6     *    c1d4;
(* 14    a13    *)    x8      :=  x1     +    x2    ;
(* 15    a14    *)    x2      :=  x1     -    x2    ;
(* 16    m2     *)    x2      :=  x2     *    c1d4;
(* 17    a15    *)    x1      :=  x4     +    x5    ;
(* 18    a16    *)    x5      :=  x4     -    x5    ;
(* 19    a17    *)    x4      :=  x7     +    x6    ;
(* 20    a18    *)    x6      :=  x7     -    x6    ;
(* 21    a19    *)    x7      :=  x2     +    x3    ;
(* 22    a20    *)    x3      :=  x2     -    x3    ;
(* 23    m3     *)    x5      :=  x5     *    c1d4;
(* 24    a21    *)    x2      :=  x4     +    x7    ;
(* 25    m4     *)    x2      :=  x2     *    c1d16;
(* 26    m5     *)    x7      :=  x7     *    cps1d16;
(* 27    a22    *)    x7      :=  x2     -    x7    ;
```

```
(* 28    m6     *)    x4      :=  x4     *    smc1d16;
(* 29    a23    *)    x4      :=  x2     +    x4    ;
(* 30    a24    *)    x2      :=  x0     +    x8    ;
(* 31    m7     *)    x2      :=  x2     *    c2d16;
(* 32    m8     *)    x8      :=  x8     *    cps2d16;
(* 33    a25    *)    x8      :=  x2     -    x8    ;
(* 34    m9     *)    x0      :=  x0     *    smc2d16;
(* 35    a26    *)    x0      :=  x2     +    x0    ;
(* 36    a27    *)    x2      :=  x6     +    x3    ;
(* 37    m10    *)    x2      :=  x2     *    c3d16;
(* 38    m11    *)    x3      :=  x3     *    cps3d16;
(* 39    a28    *)    x3      :=  x2     -    x3    ;
(* 40    m12    *)    x6      :=  x6     *    smc3d16;
(* 41    a29    *)    x6      :=  x2     +    x6    ;
(* ordering:     1     7     8     3     5     6     0     4     *)
end;
```

Curriculum Vitae

4.10.1957	Naissance à Soleure, Suisse.
1964-1972	Classes primaires et secondaires dans le canton de Neuchâtel
1973-1976	Etudes au Gymnase Cantonal de Neuchâtel
1976	Obtention du baccalauréat scientifique (mention très bien) et de la maturité fédérale type C
1976-1981	Etudes à l'Ecole Polytechnique Fédérale de Zürich, dans la section d'électricité
1981	Obtention du diplôme d'ingénieur électricien de l'EPFZ
1981	Ingénieur en génie logiciel à Siemens-Albis, Zürich
1981-1982	Etudes à Stanford University, dans la section d'électricité
1982	Assistant de recherche au département d'informatique de Stanford
1982	Obtention du "Master of Science" de Stanford University
1983-1986	Assistant scientifique au Laboratoire d'Informatique Technique de l'EPFL. Travaux de recherche sur les transformées rapides et les bancs de filtres numériques