



THÈSE

soutenue le 20/06/2008 pour obtenir

le grade de **Docteur en Sciences** de l'Université de Paris-Est Marne-la-Vallée

Spécialité : Traitement du Signal

École doctorale : Information, Communication, Modélisation et Simulation

par

Jérôme GAUTHIER

## **Analyse de signaux et d'images par bancs de filtres. Applications aux géosciences.**

Composition de la commission d'examen :

Rapporteurs :	Pierre SIOHAN Thierry BLU
Examineurs :	Patrick FLANDRIN Jérôme MARS
Directeur de thèse :	Jean-Christophe PESQUET
Co-Directeur :	Laurent DUVAL



---

# Résumé

---

## Analyse de signaux et d'images par bancs de filtres. Applications aux géosciences.

Afin de réaliser des traitements locaux sur des données de diverses natures (volumes, images ou signaux) contenant des éléments informatifs dans certaines bandes de fréquence, nous nous intéressons dans cette thèse à l'étude de bancs de filtres (BdF). Plus précisément, nous étudions l'existence et la synthèse de BdF de réponses impulsionnelles finies (RIF) inverses d'un BdF d'analyse RIF redondant complexe fixé. Nous proposons en particulier des méthodes testant l'inversibilité de la matrice d'analyse et la construction d'un inverse explicite à l'aide de la formulation polyphase. À partir de ce dernier, nous proposons une paramétrisation réduite de l'ensemble des BdF de synthèse permettant d'optimiser leurs réponses selon différents critères. Cette étude est étendue au cas multidimensionnel notamment par l'utilisation de la notion de résultant.

Ces outils permettant de représenter efficacement certaines informations structurées dans des données, il devient possible de les préserver tout en rejetant d'éventuelles perturbations. Le premier cadre considéré est celui du bruit gaussien. Nous avons utilisé le principe de Stein pour proposer deux méthodes de débruitage : FB-SURELET-E et FB-SURELET-C. Elles sont comparées à des méthodes récentes de débruitage conduisant à de bons résultats en particulier pour des images texturées. Un autre type d'application est ensuite considéré : la séparation des structures orientées. Afin de traiter ce problème, nous avons développé une méthode de filtrage anisotrope. Les algorithmes réalisés sont finalement testés sur des données issues de différents domaines applicatifs (sismique, microscopie, vibrations)

**Mots clefs :** Bancs de filtres suréchantillonnés, redondance, inversion, optimisation, filtrage directionnel, principe de Stein, données sismiques, microscopie électronique en transmission.





---

# Abstract

---

## **Signal and image analysis with filter banks. Applications to geosciences.**

Our main purpose in this PhD thesis is to perform local frequential (or directional) processing in different kind of data (volumes, images or signals). To this end, filter banks (FBs) are studied. More precisely, we first investigate the existence and the construction of synthesis FBs inverse to a given FIR complex analysis FB. Through the study of the polyphase analysis matrix, we are able to propose methods to test the invertibility and to build one inverse FB. Using this inverse, we provide a parametrization of the set of synthesis FB, with which we optimize filter responses with different criteria. The same study is performed in the multidimensional case.

Since FBs provide an efficient representation of structured information in data, it is then possible to preserve them while rejecting unwanted perturbations. By associating Stein's principle and those FBs, we proposed two methods to denoise signals and images corrupted by Gaussian noise. These methods, named FB-SURELET-E and FB-SURELET-C, are compared to recent denoising methods and are found to offer good results, especially for textured images. Another type of application is then investigated : separation of oriented structures. To this end, we have developed an anisotropic filtering method. The different proposed methods are finally applied on images and signals from various fields : seismic images and cubes, transmission electron microscopy (TEM) images of catalysts and vibration signals from car engines.

**Keywords :** Oversampled filter banks, redundancy, inversion, optimization, directionnal filtering, Stein's principle, seismic data, transmission electron microscopy.



---

# Remerciements

---

L'exercice des remerciements constitue à la fois un plaisir, permettant de se replonger avec joie dans les souvenirs heureux de ces dernières années, et une tâche difficile puisqu'il ne faut oublier personne.

En premier lieu je tiens à remercier chaleureusement Jean-Christophe Pesquet de m'avoir accordé sa confiance, guidé et trouvé les mots pour m'encourager et me motiver dans les moments difficiles. Bien entendu, j'y associe Laurent Duval qui m'a fait découvrir le monde du traitement du signal (en m'expliquant par exemple qu'une transformée de Fourier n'était *pas qu'un* isomorphisme d'espaces de Hilbert de  $L^2$  dans  $L^2$  !) et dont le bouillonnement d'idées a été une source constante d'inspiration. J'ai appris énormément à leur contact et ils ont su me communiquer la rigueur, l'envie et la passion nécessaires à l'activité de recherche.

Un merci tout particulier est également adressé à Patrick Flandrin d'avoir accepté de présider mon jury de thèse, à Pierre Siohan et Thierry Blu qui ont eu la lourde tâche de rapporter ce manuscrit et à Jérôme Mars d'avoir participé à mon jury. Je les remercie également pour les remarques et suggestions qui ont conduit à la version finale de ce document.

Une pensée tout particulière va naturellement à ma famille et plus particulièrement à mes parents, ma tante Odile et mon grand père, qui m'ont soutenu tant moralement que logistiquement, en me déchargeant de nombreux soucis d'ordre domestique.

Je tiens aussi à saluer amicalement et avec gratitude certains membres de la communauté scientifique avec lesquels j'ai eu l'occasion d'échanger et d'apprendre, en particulier Toshihisa Tanaka, Laurent Busé, Carlos D'Andrea, Agnes Szanto et Jules Browaeys.

J'adresse aussi mes remerciements aux membres de l'Équipe Signal et Communications du Laboratoire d'Informatique de l'Institut Gaspard Monge ; les permanents : Philippe Loubaton, Antoine Chevreuil, Christophe Vignat, les anciens co-thésards : Marc Castella (merci aussi pour les discussions sur les bases de Gröbner), Pierre Jallon, Julien Dumont, Caroline Chaux (merci en particulier pour le style de document) et Wided Milled, sans oublier bien entendu les bienheureux actuellement en thèse : Nelly Pustelnik, Elena

Florian, Lotfi Chaari, Mounir Kaaniche et Pascal Vallet.

Ayant travaillé pour une grande partie à l'IFP je voudrais remercier mes anciens collaborateurs et amis : Marie-Christine Cacas, Van Bui Tran, Dominique Humeau, Stéphane Ker, Séverine Baillet, Fabrice Guillemain, Benoît Celse, Emmanuel Nguyen, Christoph Pflieger, Josette Bruneau et Martine Ancel.

N'étant par nature pas l'être le plus efficace pour les démarches administratives, je tiens à exprimer ma gratitude à Carole Guénaud, Jocelyne Wiecha, Line Fonfrède, Sylvie Cash, Gabrielle Lethiers et Sylvie Donard pour leur patience et leur aide précieuse.

Je voudrais également saluer mes amis de Paris ou d'ailleurs qui ont su m'aérer un peu l'esprit : Benjamin, Anne Sophie, Sarah et Vincent, Laure et Jonathan, Maxime, Maureen, Nour, Gaëlle, Anna et Sarah, les Pierre (Kerlu et Pir), Gnenhoury et Louis, Matthieu, Sophie et Claude, Karine et Keyang.

Je tiens à saluer les membres de l'équipe TSI de Telecom ParisTech qui m'ont accueilli, m'ont fait me sentir chez moi en leur compagnie et m'ont prodigué de nombreux conseils pour la préparation de ma soutenance : Béatrice Pesquet-Popescu, Thomas, les Marco, Federica, Téodora (pour tout le soutien psychologique) et Aurélia (pour les encouragements et les nombreux conseils).

Enfin, des remerciements personnalisés sont disponibles pour toutes les personnes que j'ai soigneusement évité de nommer ici !

---

# Table des matières

<b>Table des matières</b>	<b>9</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Motivations . . . . .	15
1.2 Démarche . . . . .	16
1.3 Plan de la thèse . . . . .	18
1.4 Publications . . . . .	20
<b>2 Rapide état de l'art</b>	<b>23</b>
2.1 De Fourier aux ondelettes . . . . .	23
2.1.1 Quelques transformées standards . . . . .	23
2.1.2 Transformées par blocs . . . . .	26
2.1.3 Transformées à recouvrement . . . . .	27
2.1.4 Bancs de filtres . . . . .	29
2.1.5 Transformées en ondelettes . . . . .	34
2.2 Redondance et suréchantillonnage . . . . .	36
2.2.1 Trames . . . . .	37
2.2.2 Bancs de filtres suréchantillonnés . . . . .	38
2.3 Transformées multidimensionnelles et directionnelles . . . . .	41
2.3.1 Bancs de filtres multidimensionnels . . . . .	41
2.3.2 Bandelettes . . . . .	42
2.3.3 Curvelets . . . . .	42
2.3.4 Contourlets . . . . .	43
2.4 Conclusion . . . . .	44
<b>3 Inversion de bancs de filtres 1D suréchantillonnés</b>	<b>45</b>
3.1 Formulation polyphase et redondance . . . . .	45
3.1.1 Notations utilisées . . . . .	45
3.1.2 Formulation polyphase . . . . .	46
3.1.3 Banc de filtres de synthèse . . . . .	47
3.2 Inversibilité d'un banc de filtres d'analyse . . . . .	47
3.3 Calcul explicite d'un banc de filtres de synthèse . . . . .	49
3.4 Le cas symétrique . . . . .	51
3.4.1 Une première méthode de construction . . . . .	52
3.4.2 Une seconde méthode de construction . . . . .	53
3.5 Exemples de bancs de filtres . . . . .	55

3.5.1	Transformée étendue à recouvrement . . . . .	55
3.5.2	Transformée modulée complexe redondante à recouvrement . . . . .	56
<b>4</b>	<b>Optimisation de bancs de filtres suréchantillonnés</b>	<b>65</b>
4.1	Réduction de la dimension du problème . . . . .	65
4.1.1	Cas général . . . . .	65
4.1.2	Cas symétrique . . . . .	67
4.2	Solution optimale . . . . .	68
4.2.1	Forme générale de la fonction de coût considérée . . . . .	68
4.2.2	Critère de bonne localisation temporelle . . . . .	69
4.2.3	Critère de bonne localisation fréquentielle . . . . .	70
4.3	Optimisation par descente de gradient . . . . .	71
4.3.1	Algorithme du gradient . . . . .	71
4.3.2	Expression générale du gradient . . . . .	73
4.3.3	Exemples de gradients . . . . .	73
4.4	Solution optimale dans le cas symétrique . . . . .	74
4.4.1	Fonctions de coût dans le cas symétrique . . . . .	74
4.4.2	Exemples de fonctions de coût . . . . .	74
4.4.3	Gradients des fonctions de coût . . . . .	75
4.5	Simulations . . . . .	76
4.5.1	Choix des paramètres des noyaux . . . . .	76
4.5.2	Temps de calculs . . . . .	76
4.5.3	Bancs de filtres optimisés . . . . .	77
4.6	Conclusion . . . . .	86
<b>5</b>	<b>Bancs de filtres multidimensionnels sous-échantillonnés rectangulairement</b>	<b>87</b>
5.1	Notations dans le cadre multidimensionnel . . . . .	87
5.1.1	Objectif . . . . .	87
5.1.2	Notation polyphase . . . . .	88
5.2	Inversibilité et résultants . . . . .	91
5.2.1	Résultats préliminaires . . . . .	91
5.2.2	Description de l'algorithme . . . . .	96
5.3	Méthode utilisant les bases de Gröbner . . . . .	100
5.3.1	Quelques rappels sur les bases de Gröbner . . . . .	101
5.3.2	Application au problème d'inversibilité . . . . .	101
5.3.3	Discussion . . . . .	102
5.4	Calcul d'un inverse . . . . .	103
5.4.1	Méthode de calcul par pseudo-inverse . . . . .	103
5.4.2	Cas symétrique . . . . .	105
5.4.3	Paramétrisation des bancs de filtres de synthèse . . . . .	106
5.4.4	Exemple d'inversion . . . . .	107
5.5	Cas où la redondance est égale au recouvrement . . . . .	107
5.5.1	Expression exacte du banc de filtres de synthèse . . . . .	107
5.5.2	Exemple de fenêtre de reconstruction . . . . .	112
5.5.3	Exemple de banc de filtres . . . . .	113
5.6	Conclusion . . . . .	114

<b>6</b>	<b>Méthodes pour la réjection de bruit</b>	<b>117</b>
6.1	Débruitage basé sur le principe de Stein . . . . .	117
6.1.1	Présentation et fondements théoriques . . . . .	117
6.1.2	Estimation du risque sur les coefficients avant reconstruction . . . . .	119
6.1.3	Estimation du risque après reconstruction . . . . .	122
6.1.4	Cas orthogonal . . . . .	126
6.1.5	Cas des transformées à recouvrement . . . . .	127
6.1.6	Extension en dimension 2 . . . . .	130
6.1.7	Formalisme SURE-LET . . . . .	133
6.1.8	Mise en œuvre du débruitage par SURE-LET . . . . .	136
6.1.9	Simulations . . . . .	140
6.2	Filtrage directionnel dans des images . . . . .	149
6.2.1	Estimation des directions . . . . .	149
6.2.2	Sélection de direction . . . . .	157
6.2.3	Simulations sur une image artificielle . . . . .	158
6.2.4	Estimation des mélanges de direction . . . . .	159
6.2.5	Méthode de filtrage conservant les mélanges . . . . .	160
6.3	Conclusion . . . . .	161
<b>7</b>	<b>Quelques exemples d'applications</b>	<b>163</b>
7.1	Données sismiques . . . . .	163
7.1.1	Présentation : acquisition des données sismiques . . . . .	163
7.1.2	Filtrage directionnel . . . . .	165
7.1.3	Extraction d'onlap/toplap . . . . .	172
7.1.4	Autres applications en sismique . . . . .	178
7.2	Microscopie électronique à transmission . . . . .	182
7.2.1	Présentation et problématique . . . . .	182
7.2.2	Résultats . . . . .	183
7.3	Analyse temps-fréquence de signaux de combustion . . . . .	183
7.3.1	Motivations . . . . .	183
7.3.2	Représentation temps-fréquence . . . . .	185
7.3.3	Résultats . . . . .	186
<b>8</b>	<b>Conclusion</b>	<b>191</b>
8.1	Résumé des travaux réalisés . . . . .	191
8.2	Perspectives . . . . .	193
<b>A</b>	<b>Algorithme de test d'inversibilité</b>	<b>195</b>
	<b>Table des figures</b>	<b>199</b>
	<b>Liste des tableaux</b>	<b>203</b>
	<b>Bibliographie</b>	<b>205</b>





---

# Glossaire

---

$I_N$	Matrice identité de taille $N \times N$
$\mathbf{0}_{M,N}$	Matrice nulle de taille $M \times N$
$\mathbf{J}_N$	Matrice d'échange, ou coidentité, de taille $N \times N$
$(.)^\top$	Opérateur de transposition
$\mathbf{A}^R$	Matrice des parties réelles de la matrice $\mathbf{A}$
$\mathbf{A}^I$	Matrice des parties imaginaires de la matrice $\mathbf{A}$
$\text{diag}(\mathbf{A})$	Vecteur des termes diagonaux de la matrice $\mathbf{A}$
$\text{Diag}(\mathbf{v})$	Matrice diagonale ayant le vecteur $\mathbf{v}$ sur la diagonale principale
$\text{Tr}(\mathbf{A})$	Trace de la matrice $\mathbf{A}$
$\mathbb{N}_L$	$\{0, \dots, L\}$
$\mathbb{N}_L^*$	$\{1, \dots, L\}$
BdF :	Banc de Filtres
RIF :	Réponse Impulsionnelle Finie
RII :	Réponse Impulsionnelle Infinie
TCD :	Transformée en Cosinus Discrète
TFD :	Transformée de Fourier Discrète
TH :	Transformée de Hadamard
$\mathbb{N}$	Espaces des nombres entiers relatifs
$\mathbb{Z}$	Espaces des nombres entiers naturels
$\mathbb{R}$	Espaces des nombres réels
$\mathbb{R}^*$	Espaces des nombres réels non nuls
$\mathbb{R}_+^*$	Espaces des nombres réels positifs non nuls
$\mathbb{C}$	Espaces des nombres complexes
$\mathbb{C}^*$	Espaces des nombres complexes non nuls



---

# Chapitre 1

## Introduction

---

### 1.1 Motivations

En traitement du signal ou des images, des méthodes très variées peuvent être employées suivant le domaine d'application visé. Par exemple en traitement d'images, on peut utiliser des méthodes basées sur des équations aux dérivées partielles pour la restauration d'image aussi bien que des transformées en cosinus pour comprimer l'information (JPEG). Il est cependant agréable de disposer de classes d'outils relativement génériques et aisément adaptables à une gamme de problèmes rencontrés. Ainsi les ondelettes forment une classe de méthodes extrêmement large qui, par le choix de bons noyaux et paramètres, peuvent s'appliquer dans de nombreuses situations. Compression d'images, segmentation, extraction d'informations, déconvolution, restauration, reconstruction : pour ces différents problèmes, il est possible d'utiliser des méthodes basées sur des transformations de type "ondelette" ou "bancs de filtres" avec des caractéristiques différentes, par exemple de redondance, régularité, symétrie ou d'invariance par translation.

Le but de cette thèse a été d'étudier un ensemble de transformations aussi large que possible pouvant fournir des outils flexibles et versatiles s'adaptant à plusieurs problématiques industrielles rencontrées à l'Institut Français du Pétrole, où s'est déroulé la thèse en collaboration avec le Laboratoire d'Informatique de l'Institut Gaspard Monge de l'Université Paris-Est. La première application sur laquelle nous avons travaillé, l'amélioration d'images sismiques, nous a guidé dans le choix de la classe de méthodes à étudier. Les données sismiques présentent, en effet, plusieurs caractéristiques à prendre en considération : forte anisotropie, présence de perturbations structurées et aléatoires et volumes de données importants. Le point de départ de cette thèse a donc été une recherche d'une classe d'outils génériques permettant de travailler sous ces contraintes.

L'anisotropie de ces données nous a conduit à nous intéresser à des transformées complexes qui se révèlent souvent bien adaptées pour isoler et sélectionner des directions locales. La contrainte liée au volume de données nous a amené à travailler avec les transformées à recouvrement qui, par ailleurs, ont déjà fait la preuve de leur utilité pour

la compression [Malvar, 1992; Duval, 2000; de Queiroz, Tran, 2001]. Ce type de représentations linéaires n'agissant que sur un bloc d'échantillons à la fois, elles permettent de travailler sur de gros volumes de données en limitant l'espace mémoire nécessaire et, éventuellement, en effectuant les traitements en parallèle. Enfin, la lecture d'autres travaux [Chaux, 2006] nous avait permis de mesurer l'utilité de la redondance pour la finesse de la détection des directions et donc nous désirions évaluer le potentiel des filtres suréchantillonnés. Dans le cadre d'une utilisation sur des volumes importants de données, il fallait également pouvoir contrôler le facteur de redondance de la transformée pour limiter les problèmes de temps de calcul et d'occupation de la mémoire, sans en compromettre l'efficacité.

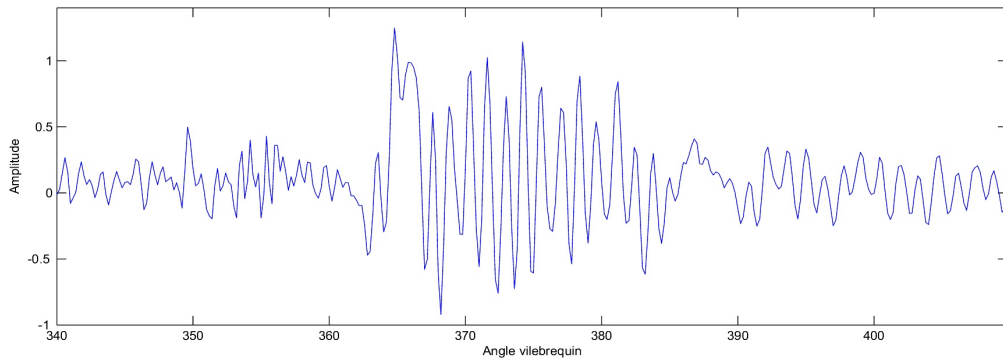
## 1.2 Démarche

L'étude des transformées à recouvrement nous a conduit à travailler sur les bancs de filtres qui sont au cœur de très nombreuses applications en traitement du signal, aussi bien pour les domaines traditionnels d'emploi de ces outils (audio, télécommunications, *etc*) que pour l'implémentation des décompositions en ondelettes, par exemple. Les bancs de filtres ont fait l'objet de nombreuses études depuis le milieu des années 70. L'inversibilité des bancs de filtres d'analyse et la construction de ces structures dans les cas à réponses impulsionnelles finies ou infinies, avec ou sans redondance, ont notamment été étudiées. Notre volonté de générer des outils réemployables dans des cadres applicatifs variés nous a amené à nous intéresser à un problème qui a été relativement peu exploré jusqu'à présent : l'étude de l'inversion de bancs de filtres RIF faiblement redondants. La volonté de laisser l'utilisateur régler cette redondance à souhait a été également l'une de nos motivations.

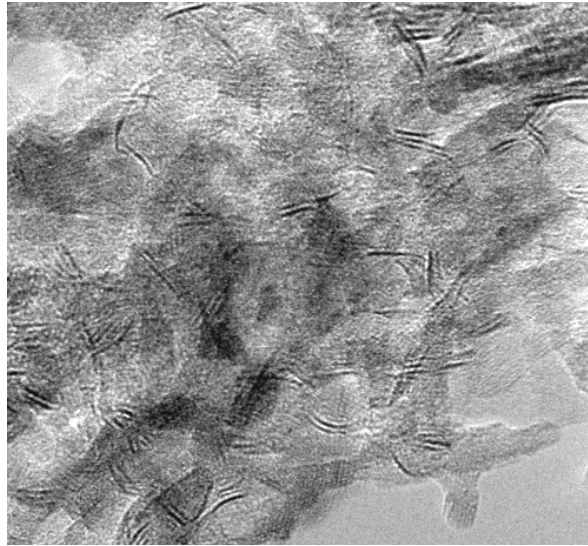
Un autre problème souvent rencontré lors de la construction de couples de bancs de filtres d'analyse/synthèse dans le cas RIF est d'obtenir des filtres avec de bonnes propriétés, telles qu'une bonne sélectivité fréquentielle ou une bonne localisation temporelle. Dans le cadre suréchantillonné, il n'y a en général pas d'inverse unique à un banc de filtres d'analyse inversible donné. Cette non-unicité est un élément que nous avons exploité pour choisir parmi les bancs de filtres de synthèse possibles celui présentant les meilleures caractéristiques. Ce choix va se traduire par une étape d'optimisation de certaines fonctions de coût dans l'ensemble des bancs de filtres de synthèse permettant la reconstruction parfaite.

Après avoir construit des bancs de filtres présentant de bonnes propriétés, il est également important dans les applications de proposer des méthodes de sélection ou de seuillage des coefficients pertinents. En vue de certains problèmes nécessitant un filtrage directionnel, nous avons proposé une méthode de détection et de sélection, automatique ou supervisée, de directions locales. Nous avons également travaillé sur une méthode de débruitage spécifique au cas du bruit blanc gaussien, en utilisant des estimateurs de Stein.

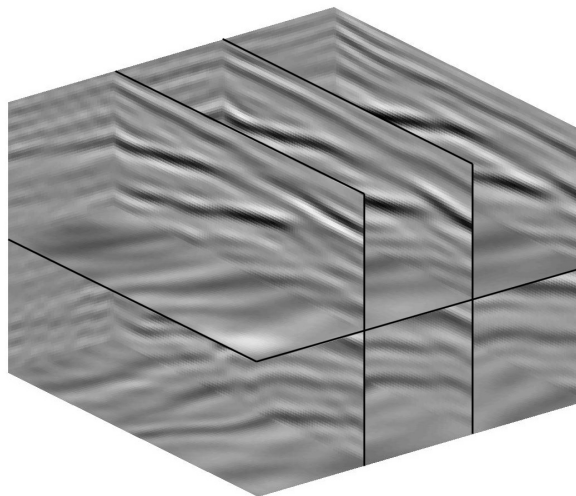
Cette thèse ayant été réalisée dans le cadre d'un contrat CIFRE entre l'Institut Français du Pétrole et l'Université de Paris-Est Marne-la-Vallée, nous avons appliqué les méthodes par banc de filtres à diverses problématiques industrielles de l'IFP. Tout d'abord



(a)



(b)



(c)

FIGURE 1.1 – Différents types de données : (a) Signal de cliquetis d'un moteur, (b) image de microscopie électronique par transmission d'un catalyseur et (c) cube sismique.

nous avons appliqué le filtrage directionnel sur des images sismiques utilisées en exploration pétrolière, puis sur des images de microscopie électronique par transmission de certains catalyseurs qui présentent des structures orientées importantes pour l'utilisateur. Enfin, nous avons employé les bancs de filtres, dans un cadre différent, pour réaliser des analyses temps-fréquences sur des signaux issus de capteurs de cliquetis de moteurs. Des exemples de ces différents types de données sont visibles à la figure 1.1.

## 1.3 Plan de la thèse

### État de l'art

Nous débutons par un rapide tour d'horizon des transformées proches, dans leur forme ou leur finalité, de celles que nous avons étudiées. Après un rappel concernant les transformées par blocs, à recouvrement, par bancs de filtres et en ondelettes, nous nous intéresserons plus particulièrement au cas des transformées redondantes et suréchantillonnées. L'accent sera porté sur les différentes méthodes de construction de bancs de filtres, en particulier dans des cas redondants. Enfin, nous rappellerons les principes de certaines transformées directionnelles récentes pouvant être utilisées pour des applications en filtrage ou en analyse directionnelle.

### Inversibilité et inversion de bancs de filtres 1D suréchantillonnés

Nous nous intéressons ensuite, dans un premier temps, aux transformées à recouvrement unidimensionnelles suréchantillonnées et plus particulièrement à leur inversibilité. L'analyse du problème d'inversion, sous forme polyphase, va nous permettre de proposer un algorithme pour vérifier l'inversibilité d'un banc de filtres d'analyse à réponse impulsionnelle finie donné. Nous allons également présenter une méthode permettant le calcul explicite d'un banc de filtres de synthèse inverse. Ce travail sera complété par l'étude d'un cas intéressant pour les applications impliquant une action ou une sélection des coefficients après décomposition : celui des bancs de filtres de synthèse à reconstruction réelle présentant une symétrie hermitienne. Deux méthodes permettant de générer de tels filtres de synthèse seront exposées. Ce chapitre sera complété par des exemples de construction de bancs de filtres de synthèse à partir de bancs de filtres d'analyse particuliers.

### Optimisation des bancs de filtres

Afin d'améliorer les performances des bancs de filtres de synthèse calculés au chapitre 3, nous allons exploiter la redondance des bancs de filtres d'analyse pour choisir, parmi un ensemble de bancs de filtres de synthèse à reconstruction parfaite, ceux présentant de bonnes caractéristiques, par exemple une bonne sélectivité fréquentielle ou une bonne localisation temps-fréquence. Pour ce faire, nous allons, en premier lieu, réexprimer le système d'équations définissant les filtres de synthèse, présenté dans le chapitre précédent. Cette réécriture va permettre à la fois de réduire l'espace des paramètres à optimiser et de transformer le problème d'optimisation sous contrainte en un problème de

minimisation non contraint. Ensuite, nous allons expliciter la méthode d'optimisation par descente de gradient utilisée, ainsi que les différentes fonctions de coût et leur gradient. Enfin, des exemples d'optimisation seront présentés mettant en relief à la fois l'intérêt en temps de calcul de la réécriture du problème, et l'amélioration apportée aux bancs de filtres de synthèse.

### **Bancs de filtres multidimensionnels**

Une partie des données à traiter étant multidimensionnelle, nous avons travaillé sur quelques aspects concernant les bancs de filtres multidimensionnels sous-échantillonnés rectangulairement. Après avoir présenté les notations employées, le problème de l'inversibilité de bancs de filtres d'analyse multidimensionnels sera étudié. Dans un premier temps, nous tenterons d'étendre la méthode de caractérisation de l'inversibilité, que nous avons proposée au chapitre 3, en utilisant la notion de résultant. La méthode obtenue ne donne pas une condition nécessaire et suffisante d'inversibilité pour la matrice polyphase, mais présente des avantages indéniables en temps de calcul, comme nous l'illustrons en la comparant à la méthode utilisant les bases de Gröbner. Ensuite, nous proposerons une extension au cas multidimensionnel de la méthode unidimensionnelle de construction d'un banc de filtres de synthèse, ainsi que de la paramétrisation de l'ensemble des filtres de synthèse. Enfin nous verrons un cas particulier, où la redondance est égale au recouvrement, de bancs de filtres multidimensionnels, pour lesquels il est possible de donner une expression explicite de l'inverse.

### **Méthodes pour le filtrage**

Dans cette partie, nous allons détailler quelques unes des méthodes que nous avons exploitées pour débruiter des images et des signaux, et utilisant les bancs de filtres présentés dans les chapitres précédents. Tout d'abord, nous avons étudié une méthode statistique utilisant l'estimateur de Stein non biaisé du risque, appliqué dans deux cas différents pour effectuer le débruitage d'images et de signaux perturbés par un bruit blanc gaussien. Nous avons travaillé sur le débruitage par SURE-LET dans le cas où la transformation utilisée est complexe et redondante en considérant d'abord une estimation sur les coefficients après reconstruction, puis une estimation sur les échantillons reconstruits. Ces deux méthodes sont appliquées sur des signaux et des images, obtenant ainsi des résultats proches de ceux produits par l'oracle associé à chaque méthode et comparables ou supérieurs à ceux donnés par d'autres méthodes récentes de débruitage.

Par ailleurs, une méthode plus empirique est proposée pour résoudre un type de problème différent : le filtrage directionnel de données anisotropes. Ce "filtrage", utilisable pour des données 2D et 3D, est finalement amélioré grâce à la prise en compte d'un attribut permettant d'évaluer les mélanges de directions.

### **Applications**

Ce chapitre est consacré à la présentation de quelques applications sur des données réelles impliquant l'utilisation des bancs de filtres construits et optimisés avec les mé-

thodes des chapitres précédents. Tout d'abord, l'acquisition des données sismiques et la problématique de l'extraction d'informations dans ces données sont rappelées. Des exemples d'applications du filtrage directionnel sur des données 2D ou 3D sont présentés ainsi que quelques explications concernant une méthode d'extraction de structures particulières appelées *onlap* et *toplap*. Une autre application de filtrage directionnel pour la séparation de directions est ensuite proposée sur des données issues de microscopie électronique par transmission (MET) de catalyseurs utilisés en raffinage. Enfin, une dernière application est décrite concernant l'analyse temps-fréquence et la reconstruction de signaux vibratoires de moteurs.

## 1.4 Publications

Cette thèse a donné lieu à plusieurs publications (dont certaines encore en préparation).

### Article de revue soumis

- Jérôme Gauthier, Laurent Duval et Jean-Christophe Pesquet, *Inversion and optimization of oversampled complex filter banks*.

### Article de revue en préparation

- Jérôme Gauthier, Laurent Duval et Jean-Christophe Pesquet, *Denoising of images with SURE-LET methods using complex oversampled filter banks*.

### Brevet

- J. Gauthier, M.-C. Cacas et L. Duval, *Méthode d'interprétation stratigraphique d'images sismiques*, 06/10 493.

### Articles de conférences

- Jérôme Gauthier, Laurent Duval et Jean-Christophe Pesquet, *A non separable 2D complex modulated lapped transform and its applications to seismic data filtering*, European Signal Processing Conference (EUSIPCO 2005), Antalya, sept. 2005, 4 pages.
- Jérôme Gauthier, Laurent Duval et Jean-Christophe Pesquet, *Low redundancy oversampled lapped transform and its applications to seismic data filtering*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006), Toulouse, Vol. 2, mai 2006, p. 821–824.
- Jérôme Gauthier et Laurent Duval, *3D seismic data processing with complex non separable multidimensional lapped transforms*, Physics in Signal and Image Processing (PSIP 2007), Mulhouse, jan. 2007, 6 pages.



- Jérôme Gauthier, Laurent Duval et Jean-Christophe Pesquet, *Oversampled inverse complex lapped transform optimization*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007), Honolulu, Hawaii, USA, Vol. 1, avril 2007, p. 549–552.
- Jérôme Gauthier et Laurent Duval, *Short-term spectral analysis and synthesis improvements with oversampled inverse filter banks*, SPIE (Society of Photo-optical Instrumentation Engineers) Conference, San Diego, Ca, USA, août 2007, 12 pages.
- Jérôme Gauthier, Marie-Christine Cacas et Laurent Duval, *Local directional and uncoherent seismic noise filtering with oversampled filter banks*, 69th European Assoc. of Geophysicists and Engineers (EAGE) Conference, Londres, juin 2007, 5 pages.
- Jérôme Gauthier, Laurent Duval et Jean-Christophe Pesquet, *Construction de bancs de filtres faiblement redondants*, (GRETSI 2007), Troyes, France, sept. 2007, 4 pages.

#### Article de conférence soumis

- Jérôme Gauthier, Laurent Duval et Jean-Christophe Pesquet, *Two denoising SURET methods for complex oversampled subband decompositions*, European Signal Processing Conference (EUSIPCO 2008), 5 pages.

#### Autre communication

- Jérôme Gauthier et Laurent Duval, *Inversion de bancs de filtres  $M$ -bandes sur-échantillonnés*, Journée nationale : logiciels pour la modélisation et le calcul scientifique, Paris, nov. 2006, 6 pages.



---

## Chapitre 2

# Rapide état de l'art

---

Ce chapitre présente un tour d'horizon rapide de transformées proches, dans leur forme ou leur objectif, de celles étudiées dans le reste du manuscrit. Nous nous attachons ici à présenter les principes et les idées plutôt que d'entrer dans les détails théoriques de chaque méthode. Après un rappel concernant quelques transformées discrètes classiques [Rao, 1985] comme la transformée de Fourier [Brémaud, 2002], en cosinus ou de Walsh [Beauchamp, 1975], les transformées par blocs, à recouvrement [Malvar, 1992], par bancs de filtres [Vaidyanathan, 1993; Strang, Nguyen, 1996] et en ondelettes [Daubechies, 1992; Flandrin, 1998; Mallat, 1998], nous nous intéresserons plus particulièrement au cas des transformées suréchantillonnées. L'accent sera mis sur les différentes méthodes de construction de bancs de filtres, en particulier dans le cas redondant. Enfin nous rappellerons certaines transformées directionnelles pouvant être utilisées pour des applications de filtrage ou d'analyse directionnelle, telles que les bandelettes, les curvelets et les contourlets.

## 2.1 De Fourier aux ondelettes

### 2.1.1 Quelques transformées standards

Les bases orthogonales discrètes ont fait la preuve de leur utilité depuis des décennies aussi bien pour les mathématiciens que les ingénieurs en proposant des représentations vectorielles efficaces. En particulier, les transformées linéaires sont au centre de nombreuses méthodes en traitement du signal et des images. Elles sont employées par exemple dans l'analyse, la détection, le filtrage, le débruitage ou la compression de données. Le principe est d'utiliser une application linéaire sur tout ou partie des échantillons d'un signal (éventuellement de manière itérative). La nature linéaire de la transformation permet une représentation matricielle qui, nous le verrons, peut également s'appliquer à un banc de filtres.

Les transformations comme la transformée de Fourier discrète (TFD), la transformée en cosinus discrète (TCD) ou la transformée de Hadamard (TH) font partie des outils

fondamentaux dans le traitement du signal. Nous allons maintenant rappeler brièvement le principe et la définition de ces transformations classiques.

### 2.1.1-a) Transformée de Fourier discrète

La transformée de Fourier discrète est devenue depuis les années 1960 et l'apparition de la transformée de Fourier rapide (Fast Fourier Transform - FFT) l'un des outils centraux du traitement du signal. Elle permet entre autres de représenter le spectre d'un signal, autrement dit ses différentes composantes fréquentielles.

On définit la transformée de Fourier à temps discret d'un signal  $(x_n)_{n \in \mathbb{Z}}$  par :

$$\forall \omega \in [0, 1[, \quad \hat{x}[\omega] = \sum_{n=-\infty}^{\infty} x(n) e^{-2i\pi\omega n}. \quad (2.1)$$

Dans le cas où le signal  $x$  est de longueur  $L$ , on définit la transformée de Fourier discrète (TFD) comme l'application de la transformée de Fourier de  $x$  en les fréquences :  $\{0, \frac{1}{L}, \dots, \frac{L-1}{L}\}$ . Cette définition est équivalente à la forme matricielle :

$$\hat{x} = \mathbf{F}x,$$

où :  $\mathbf{F}_{k,n} = e^{-2i\pi \frac{nk}{L}}$  pour tout  $k, n \in \{0, \dots, L-1\}^2$ .

La matrice  $\mathbf{F}$  ainsi définie n'est pas unitaire mais possède un inverse simple :  $\frac{1}{L}\mathbf{F}^*$ . Autrement dit, il est possible, à partir du vecteur transformé  $\hat{x}$  de reconstruire le signal original :

$$\frac{1}{L}\mathbf{F}^*\hat{x} = x.$$

Notons qu'une transformation vérifiant cette propriété est dite à *reconstruction parfaite*. D'un point de vue pratique cette propriété est rarement parfaitement vérifiée en raison des erreurs d'arrondis dans les calculs.

Remarquons, que l'équation (2.1) permet de calculer la transformée de Fourier d'un signal  $x$  pour des fréquences différentes de  $\{0, \frac{1}{L}, \dots, \frac{L-1}{L}\}$ .

De manière plus générale, on peut définir d'autres noyaux de transformations fréquentielles complexes (incluant la TFD classique) faisant intervenir des décalages non entiers sur  $n$  ou  $k$ . Pour la majorité des applications présentées dans cette thèse, nous avons employé un noyau de ce type, qui sera introduit à la section 3.5.2.

### 2.1.1-b) Transformées en cosinus discrètes

Comme nous venons de le voir la TFD est une transformation complexe, autrement dit, elle associe à un signal réel des coefficients complexes, ce qui dans la pratique n'est pas forcément aisé à manipuler. Formellement, en considérant un signal  $x$  réel, la partie réelle de l'équation (2.1) ne fait alors intervenir que des cosinus. L'idée de la transformation que nous allons rappeler maintenant est de décomposer un signal uniquement à l'aide d'une base de cosinus, ce qui permet de ne réaliser que des opérations sur des réels (ce qui est plus commode à manipuler informatiquement).

Soit un signal  $x$  de longueur  $L$ , on définit les transformées en cosinus discrète (TCD) par la formule :

$$\hat{x}[k] = \sum_{n=0}^{L-1} C_{k,n} x(n),$$

où les  $C_{k,n}$  prennent différentes formes, définissant ainsi différent types de TCD :

– TCD de type I :

$$C_{k,n} = \begin{cases} \frac{1}{2} & \text{si } n = 0 \\ \frac{(-1)^k}{2} & \text{si } n = L - 1 \\ \cos\left(\frac{\pi}{L-1}kn\right) & \text{sinon.} \end{cases}$$

– TCD de type II :

$$C_{k,n} = \cos\left(\frac{k\pi}{L}\left(\frac{1}{2} + n\right)\right).$$

– TCD de type III :

$$C_{k,n} = \begin{cases} \frac{1}{2} & \text{si } n = 0 \\ \cos\left(\frac{\pi}{L}\left(\frac{1}{2} + k\right)n\right) & \text{sinon.} \end{cases}$$

– TCD de type IV :

$$C_{k,n} = \cos\left(\frac{\pi}{L}\left(\frac{1}{2} + k\right)\left(\frac{1}{2} + n\right)\right).$$

Pour reconstruire le signal d'origine à partir de la décomposition il faut connaître l'inverse de ces transformations. On peut montrer que l'inverse de la TCD-I est elle-même multipliée par  $\frac{2}{L-1}$  et que, de même, l'inverse de la TCD-IV est la TCD-IV, elle-même, multipliée par  $\frac{2}{L}$ . Les TCD-II et TCD-III sont inverses l'une de l'autre à un facteur  $\frac{2}{L}$  près.

La TCD-II, qui a connu un grand succès (sous forme de transformée par bloc *cf.* section 2.1.2) pour des applications en compression d'images (format JPEG), est souvent appelée simplement TCD. La TCD-III est, de ce fait, souvent appelée TCDI (transformée en cosinus discrète inverse). Une version modifiée de la TCD-IV (sous forme de transformée à recouvrement *cf.* section 2.1.3) est à la base de plusieurs formats de compression audio ("MP3" ou Ogg Vorbis par exemple).

### 2.1.1-c) Transformées de Walsh et Hadamard

Un autre exemple de transformée classique est la transformée de Walsh, qui est souvent associée avec les transformées de Hadamard ou de Paley. Une définition des vecteurs d'une base de Walsh peut être donnée par l'étude des passages par zéro de certaines fonctions sinus et cosinus. Dans [Peyré, 2004], une autre manière de voir la transformée de Walsh est proposée : la transformée de Walsh n'est autre que la réécriture d'une transformée de Fourier sur un groupe abélien  $G = (\mathbb{Z}/2\mathbb{Z})^k$ . La définition complète des fonctions de Walsh est présentée dans le livre [Beauchamp, 1975].

En réordonnant les fonctions de bases définissant la transformée de Walsh, on peut retrouver la Transformée de Hadamard (TH). La matrice de cette transformée se définit récursivement par la formule suivante :

$$\mathbf{H}_0 = (1) \text{ et } \forall m \geq 1, \mathbf{H}_m = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{H}_{m-1} & \mathbf{H}_{m-1} \\ \mathbf{H}_{m-1} & -\mathbf{H}_{m-1} \end{pmatrix}.$$

Au facteur de normalisation près, cette matrice est la matrice d'Hadamard. On peut montrer que cette matrice est symétrique et unitaire :  $\mathbf{H}_m \mathbf{H}_m = \mathbf{I}_{2^m}$ . Pour effectuer la reconstruction, il suffit donc d'appliquer à nouveau la transformation.

Si l'on omet le facteur de normalisation, on voit que cette transformée ne fait intervenir que des facteurs 1 ou  $-1$  ce qui dans les applications peut présenter un avantage. En effet, 1 étant l'élément neutre de la multiplication, les opérations de la transformation s'effectuent uniquement dans l'espace d'origine des données. Cette caractéristique est particulièrement intéressante pour le codage de données entières (par exemple les valeurs des pixels d'une image) puisque les opérations sur les entiers s'exécutent plus rapidement que les opérations sur des types "flottant". Une forme hybride de la TH et de la TCD a été utilisée pour le codec vidéo H.264 (ou MPEG-4 AVC) et pour le format de compression JPEG XR de Microsoft (dans une version modifiée faisant intervenir une transformation à recouvrement).

### 2.1.2 Transformées par blocs

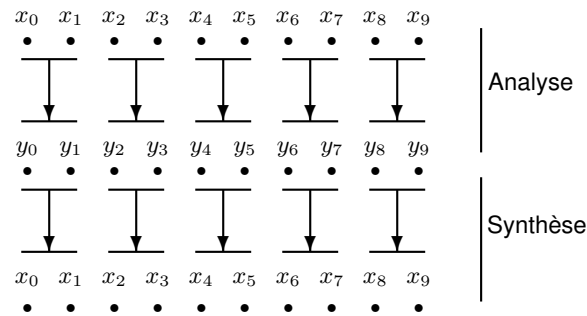


FIGURE 2.1 – Principe d'une transformée par blocs (avec un bloc de taille 2).

Des transformées localisées (ou par bloc) permettent de prendre en compte de lentes non-stationnarités des données réelles, en travaillant sur les données découpées en blocs de longueur fixée. Le principe en est le suivant : au lieu de faire la transformation sur tout le signal, on découpe le signal d'entrée  $x$  en bloc de longueur  $N$  sur lesquels on applique une transformation, comme la DCT ou la transformée de Walsh-Hadamard. Ce principe est illustré par la figure 2.1. Voyons tout de suite un exemple simple de transformée par bloc.

**Exemple 1** La transformée de Haar est décrite par la matrice de Hadamard  $2 \times 2$  suivante :

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Le signal d'entrée  $x(n)$  est découpé en blocs de  $N = 2$  échantillons consécutifs :  $\mathbf{b}_i = (x(2i), x(2i+1))^\top$ , auxquels on fait subir la transformation linéaire représentée par la matrice  $\mathbf{H}$  :

$$\begin{pmatrix} y(2i) \\ y(2i+1) \end{pmatrix} = \mathbf{H} \mathbf{b}_i.$$

*Cette transformée agit donc en calculant des moyennes et différences locales.*

Dans le cas où la transformation appliquée sur chaque bloc est linéaire, et définie par une matrice  $\mathbf{H}_a$  comme dans l'exemple précédent, on peut représenter la transformée par bloc par la matrice infinie suivante :

$$\mathbf{M}_a = \begin{pmatrix} \ddots & & & 0 \\ & \mathbf{H}_a & & \\ & & \mathbf{H}_a & \\ & & & \mathbf{H}_a \\ 0 & & & & \ddots \end{pmatrix}.$$

Pour revenir dans le domaine de départ on agit de façon analogue, en considérant une transformation locale, décrite par la matrice  $\mathbf{H}_s$ , qui agit sur chaque bloc transformé. L'opérateur de reconstruction peut donc être également écrit sous la forme d'une matrice infinie :

$$\mathbf{M}_s = \begin{pmatrix} \ddots & & & 0 \\ & \mathbf{H}_s & & \\ & & \mathbf{H}_s & \\ & & & \mathbf{H}_s \\ 0 & & & & \ddots \end{pmatrix}.$$

Ce formalisme permet de donner une condition de reconstruction parfaite très simple :

$$\mathbf{M}_s \mathbf{M}_a = \mathbf{I}_\infty \Leftrightarrow \mathbf{H}_s \mathbf{H}_a = \mathbf{I}_N.$$

Un avantage de ce type de transformation est son aspect "local". Cela permet de profiter de l'hypothèse que les signaux étudiés peuvent être considérés comme localement stationnaires. De plus, les calculs et l'implémentation s'en trouvent simplifiés (en particulier sur de très long signaux) et permet par exemple des décompositions en cours de réception de données, ou une parallélisation des traitements sur plusieurs machines (dans les cas où de gros volumes de données sont à analyser). Ce type de transformation a été utilisé et popularisé dans le format de compression d'image JPEG. Cependant, traiter indépendamment des blocs de coefficients dans le domaine transformé peut conduire à des artéfacts non voulus tels que l'effet de bloc de la compression JPEG.

Le principe de la compression JPEG est de travailler sur des blocs de  $8 \times 8$  pixels sur lesquels une transformée en Cosinus Discrète (TCD) est appliquée. Les coefficients sont ensuite quantifiés puis encodés. En faisant une quantification trop brutale, pour atteindre un meilleur taux de compression, on voit apparaître un effet de damier sur l'image. Cet effet, illustré par la figure 2.2, est l'un des inconvénients les plus communs des transformées par blocs quand elles sont utilisées pour le filtrage ou la compression, par exemple.

### 2.1.3 Transformées à recouvrement

Afin de réduire l'effet de damier produit par les transformée par blocs un autre type de transformation a été proposé dans les années 1980 [Cassereau, 1985; Malvar, Staelin,

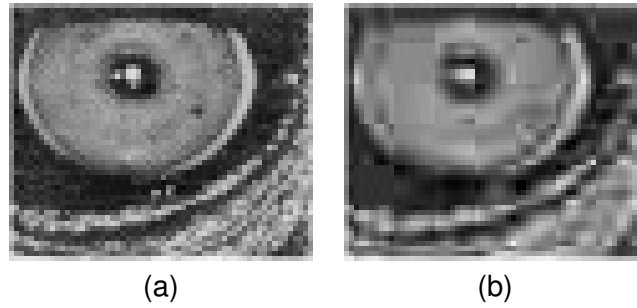


FIGURE 2.2 – Effet de damier dans la compression JPEG : (a) image originale, (b) image fortement compressée avec JPEG.

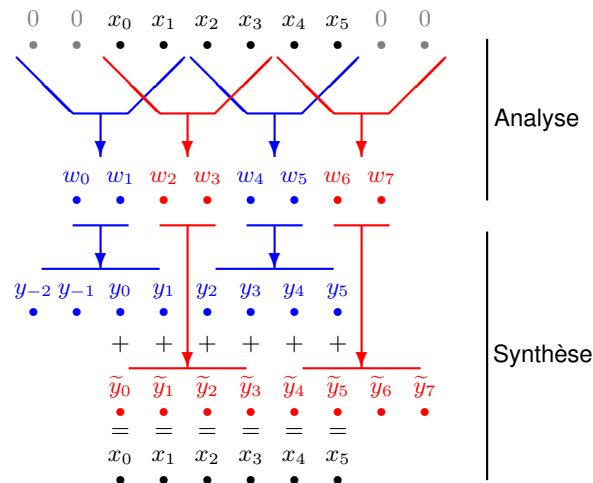


FIGURE 2.3 – Principe d'une transformée à recouvrement avec un recouvrement  $k = 2$  et des blocs élémentaires de tailles  $N = 2$ .

1989] : les transformées à recouvrement (Lapped Transforms). Le principe est, cette fois, de découper le signal d'entrée en paquets de  $k$  blocs de taille  $N$ , soit  $M = kN$  échantillons. Chaque bloc de taille  $N$  appartient ainsi à  $k$  paquets successifs, ce qui revient à dire que deux paquets successifs ont  $(k - 1)N$  échantillons en commun.

Chaque paquet ainsi défini subit alors une transformation que l'on décrit par une matrice  $\mathbf{H}_a$  de taille  $N \times kN$ . On découpe la matrice  $\mathbf{H}_a$  en  $k$  sous-matrices  $(\mathbf{H}_a(j))_{0 \leq j < k}$  de taille  $N \times N$  telles que :

$$\mathbf{H}_a = (\mathbf{H}_a(0), \dots, \mathbf{H}_a(k-1)).$$



On peut alors définir la matrice infinie de la transformation par :

$$\mathbf{M}_a = \begin{pmatrix} \ddots & & & \ddots & & & 0 \\ & \mathbf{H}_a(0) & \mathbf{H}_a(1) & \cdots & \mathbf{H}_a(k-1) & & \\ & & \mathbf{H}_a(0) & \mathbf{H}_a(1) & \cdots & \mathbf{H}_a(k-1) & \\ & & & \mathbf{H}_a(0) & \mathbf{H}_a(1) & \cdots & \mathbf{H}_a(k-1) \\ 0 & & & & \ddots & & \ddots \end{pmatrix}.$$

Le principe de cette transformation est illustré par la partie “analyse” sur la figure 2.3.

Pour retourner dans le domaine d’origine, on applique à chaque bloc transformé de taille  $N$  une transformation décrite par une matrice  $\mathbf{H}_s$  de taille  $kN \times N$ , puis on somme les résultats. Le principe de la reconstruction est illustré par la partie “synthèse” de la figure 2.3 pour  $k = 2$ .

D’un point de vue matriciel, on découpe comme précédemment la matrice  $\mathbf{H}_s$  en  $k$  sous-matrices de taille  $N \times N$  de telle sorte que :

$$\mathbf{H}_s^\top = (\mathbf{H}_s(1-k), \dots, \mathbf{H}_s(0)).$$

On peut alors donner l’expression de la matrice de la transformation inverse :

$$\mathbf{M}_s = \begin{pmatrix} \ddots & & & \ddots & & & 0 \\ & \mathbf{H}_s(0)^\top & \cdots & \mathbf{H}_s(1-k)^\top & & & \\ & & \mathbf{H}_s(0)^\top & \cdots & \mathbf{H}_s(1-k)^\top & & \\ & & & \mathbf{H}_s(0)^\top & \cdots & \mathbf{H}_s(1-k)^\top & \\ 0 & & & & \ddots & & \ddots \end{pmatrix}.$$

On a reconstruction parfaite dès lors que :

$$\begin{cases} \sum_{i=j}^{k-1} \mathbf{H}_s(j-i)^\top \mathbf{H}_a(i) = \delta_j \mathbf{I}_N & \text{si } j \geq 0, \\ \sum_{i=0}^{j+k-1} \mathbf{H}_s(j-i)^\top \mathbf{H}_a(i) = \delta_j \mathbf{I}_N & \text{si } j \leq 0. \end{cases}$$

Les transformées à recouvrement conservent les avantages des transformées par blocs tout en permettant de faire disparaître, ou d’atténuer, les effets de blocs. Elles ont été utilisées dans de nombreuses applications comme l’estimation de mouvements en vidéo [Young, Kingsbury, 1993], la réduction des effets de blocs en audio [Malvar, 1999] et plus récemment dans le nouveau format de compression d’image proposé par Microsoft. Pour une présentation plus détaillée des transformées à recouvrement nous renvoyons à l’ouvrage : [Malvar, 1992].

La notation ci-dessus sous forme de matrice infinie n’est pas commode à manipuler, c’est pourquoi nous avons préféré travailler avec une formulation équivalente par bancs de filtres, dont les transformées par blocs ou à recouvrement sont des cas particuliers, et que nous allons maintenant introduire.

### 2.1.4 Bancs de filtres

#### 2.1.4-a) Principe et exemple

Les bancs de filtres forment une classe d’outils fondamentaux en traitement du signal introduits dès le milieu des années 70 par les travaux sur les *Quadrature Mirror*

*Filter* (QMF) [Croisier *et al.*, 1976]. Le principe de cette transformation est illustré par la figure 2.4 : un signal  $x(n)$  est décomposé par les filtres  $H_0$  et  $H_1$  puis chaque décomposition subit une décimation d'un facteur 2, conduisant aux résultats  $y_0(n)$  et  $y_1(n)$ . Les filtres  $H_0$  et  $H_1$  sont appelés les *filtres d'analyse* et le couple  $[H_0, H_1]$  est appelé le *banc de filtres d'analyse*.

Les coefficients  $y_0(n)$  et  $y_1(n)$  subissent ensuite un suréchantillonnage d'un facteur 2, sont filtrés respectivement par  $\tilde{H}_0$  et  $\tilde{H}_1$  et sont finalement sommés, donnant ainsi le signal reconstruit  $\tilde{x}(n)$ . Le couple  $[\tilde{H}_0, \tilde{H}_1]$  constitue le banc de filtres de synthèse.

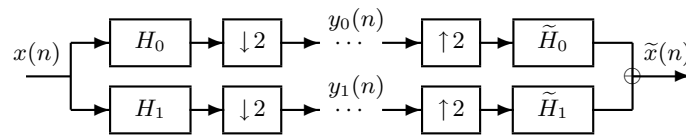


FIGURE 2.4 – Schéma d'un banc de filtres d'analyse/synthèse.

Si les signaux  $\tilde{x}(n)$  et  $x(n)$  sont égaux (éventuellement aux erreurs numériques et à un délai près) on dit que le système est à *reconstruction parfaite*. Cette propriété peut être vérifiée par un couple de bancs de filtres d'analyse/synthèse [Smith, Barnwell, 1984]. La caractérisation la plus commode de cette propriété passe par la représentation polyphase du système qui sera présentée à la section 2.1.4-c).

**Exemple 2** La transformée de Haar, présentée dans l'exemple 1, correspond au filtrage par deux filtres  $H_0$  et  $H_1$  de coefficients :

$$H_0 = \frac{1}{\sqrt{2}}(1, 1)$$

$$H_1 = \frac{1}{\sqrt{2}}(-1, 1).$$

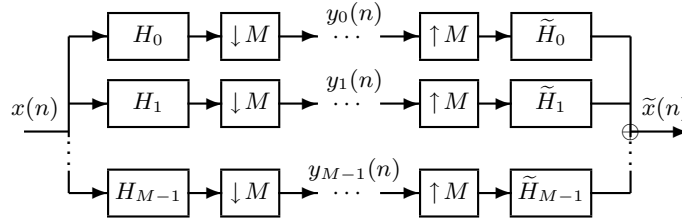
*Le banc de filtres de synthèse* Les réponses impulsionnelles  $\tilde{H}_0$  et  $\tilde{H}_1$  des filtres de synthèse sont alors symétriques de  $H_0$  et  $H_1$  par rapport à l'instant 0.

### 2.1.4-b) Bancs de filtres $M$ -bandes

Le schéma présenté à la section précédente s'étend naturellement au cas où  $M$  filtres, et non plus 2, sont utilisés en analyse et en synthèse. La figure 2.5 illustre un tel banc de filtres  $M$ -bandes.

Ce formalisme permet de matérialiser les transformées linéaires, vues précédemment. Soit une transformée linéaire décrite par une matrice  $\mathbf{H} \in \mathbb{K}^{M \times M}$  (avec en général,  $\mathbb{K} = \mathbb{R}$  ou  $\mathbb{C}$ ), on peut construire son banc de filtres équivalent en considérant  $M$  filtres d'analyse  $(h_i)_{1 \leq i \leq M}$  de réponses impulsionnelles :

$$\forall (i, j) \in \{1, \dots, M\}^2, \quad h_i(j) = H_{i, M+1-j},$$

FIGURE 2.5 – Schéma d'un banc de filtres  $M$ -bandes.

et un facteur de sous-échantillonnage de  $M$ . Soit un vecteur  $\mathbf{v} \in \mathbb{K}^M$  représentant un signal à décomposer. On peut alors écrire le filtrage de  $\mathbf{v}$  par le filtre  $\mathbf{h}_i : \forall k \in \{1, \dots, 2M - 1\}$ ,

$$(\mathbf{h}_i * \mathbf{v})(k) = \sum_{j=\max(1, k+1-M)}^{\min(M, k)} h_i(k+1-j) v_j = \sum_{j=\max(1, k+1-M)}^{\min(M, k)} H_{i, M-k+j} v_j.$$

Après sous-échantillonnage d'un facteur  $M$ , on obtient pour le premier coefficient :

$$(\mathbf{h}_i * \mathbf{v})(M) = \sum_{j=1}^M H_{i,j} v_j = (\mathbf{H}\mathbf{v})_i.$$

Les transformations par blocs comme la TFD, la TCD ou la TH peuvent donc être exprimées sous la forme de bancs de filtres  $M$ -bandes.

Plus généralement, les transformées à recouvrement peuvent être reformulées sous la forme de bancs de filtres RIF. Ce lien sera revu plus en détails à la section 3.1.

La condition de reconstruction parfaite peut également être établie dans le cas  $M$ -bandes. Vaidyanathan a montré [Vaidyanathan, 1987] que l'utilisation d'une représentation polyphase pour des systèmes RIF ou RII permet de donner une condition nécessaire et suffisante pour vérifier cette propriété. Rappelons donc maintenant ce qu'est la notation polyphase et quels sont ses avantages.

### 2.1.4-c) Notation polyphase

La notation polyphase a été introduite au milieu des années 70 par [Bellanger *et al.*, 1976] et permet de simplifier la construction et l'analyse des bancs de filtres. Avant de rappeler la méthode de construction de la matrice polyphase d'un banc de filtres, voyons d'où vient ce concept. Pour ce faire nous allons reprendre une présentation proche de celle proposée dans [Strang, Nguyen, 1996].

Quand un signal subit une décimation d'un facteur 2, cela revient à séparer les échantillons avec un indice pair de ceux avec un indice impair, ou encore, à séparer deux phases distinctes ayant des délais (auss appelé écarts de phase) différents dans le vecteur. Si on considère une décimation d'un facteur  $M$ , le principe est exactement le même mais avec  $M$  phases distinctes.

En considérant le schéma bloc d'un banc de filtres, par exemple 2.5, on voit que le filtrage est classiquement appliqué avant la décimation. Cela signifie donc que le signal

est filtré par les  $M$  filtres et que l'on ne garde ensuite qu'une phase parmi les  $M$  de chaque résultat de filtrage. Cette manière de faire la transformation n'est pas efficace car elle entraîne beaucoup de calculs qui ne seront pas exploités par la suite. La notion de représentation polyphase revient à inverser ce schéma en faisant en premier lieu l'opération de sous-échantillonnage, puis une opération de filtrage. La figure 2.6 illustre ce principe dans le cas de deux filtres.

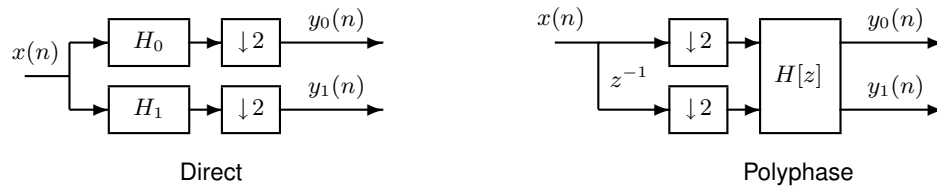


FIGURE 2.6 – Schéma direct et schéma polyphase d'un banc de filtres d'analyse.

Voyons comment écrire la forme polyphase d'un vecteur. Soit un signal  $x(n)$ . On le sépare en deux phases de la manière suivante :

$$\mathbf{x} = (... , x(0), 0, x(2), 0, x(4), ...) + (... , 0, x(1), 0, x(3), 0, ...) = \mathbf{x}_{pair} + \mathbf{x}_{impair}.$$

En considérant la transformée en  $z$  et en séparant les puissances paires et impaires, on peut écrire :

$$\mathbf{x}[z] = (x(0) + x(2)z^{-2} + x(4)z^{-4} + ...) + z^{-1} (x(1) + x(3)z^{-2} + ...),$$

autrement dit :  $\mathbf{x}[z] = \mathbf{x}_{pair}[z^2] + z^{-1}\mathbf{x}_{impair}[z^2]$ . Chaque phase a donc sa propre transformée en  $z$  :

$$\begin{aligned} \mathbf{x}_0[z] &= \sum_k x(2k)z^{-k} \text{ pour } \mathbf{x}_{pair} \\ \mathbf{x}_1[z] &= \sum_k x(2k+1)z^{-k} \text{ pour } \mathbf{x}_{impair} \end{aligned}$$

On peut appliquer exactement le même principe aux filtres, en considérant le vecteur constitué par ses réponses impulsionnelles. Soit  $\mathbf{h}$  un filtre (RIF ou RII), en séparant les réponses d'indice pair et d'indice impair puis en faisant une transformée en  $z$ , on peut écrire :  $\mathbf{h}[z] = \mathbf{h}_0[z^2] + z^{-1}\mathbf{h}_1[z^2]$ .

En remarquant que la transformée en  $z$  d'une convolution se réduit à une multiplication des transformées en  $z$ , la transformée en  $z$  du résultat du filtrage du vecteur  $\mathbf{x}$  par le filtre  $\mathbf{h}$  s'écrit alors :  $\mathbf{h}[z]\mathbf{x}[z]$ . Si le filtrage est suivi d'une étape de décimation, on ne garde que les puissances paires de  $z$  donc :

$$(\mathbf{h}[z]\mathbf{x}[z])_{pair} = \mathbf{h}_0[z]\mathbf{x}_0[z] + z^{-1}\mathbf{h}_1[z]\mathbf{x}_1[z].$$

La matrice polyphase de  $\mathbf{h}$  est alors  $\mathbf{h}_{poly}[z] = (\mathbf{h}_0[z], \mathbf{h}_1[z])$ . On peut remarquer que la longueur du support du filtre n'intervient pas dans la forme de la matrice mais dans la

forme de ses coefficients, en particulier si le filtre est à réponse impulsionnelle finie alors  $\mathbf{h}_0[z]$  et  $\mathbf{h}_1[z]$  sont des polynômes (éventuellement de Laurent).

En considérant maintenant deux filtres d'un banc de filtres notés  $\mathbf{f}$  et  $\mathbf{g}$ , de matrices polyphase  $\mathbf{f}_{poly}[z]$  et  $\mathbf{g}_{poly}[z]$ , la matrice polyphase de ce banc de filtres est :

$$\mathbf{H}[z] = \begin{pmatrix} \mathbf{f}_{poly}[z] \\ \mathbf{g}_{poly}[z] \end{pmatrix} = \begin{pmatrix} \mathbf{f}_0[z] & \mathbf{f}_1[z] \\ \mathbf{g}_0[z] & \mathbf{g}_1[z] \end{pmatrix}.$$

La matrice présentée dans cet exemple est de taille  $2 \times 2$ . D'une manière plus générale, le nombre de lignes de la matrice polyphase est le nombre de filtres considérés dans le banc de filtres et le nombre de colonnes est le nombre de phases, *i.e.* le facteur de sous-échantillonnage.

En suivant une démarche similaire on peut construire la matrice polyphase d'un banc de filtres de synthèse. Dans ce cas, l'opération de suréchantillonnage qui précède le filtrage est remplacée par une étape de filtrage suivie d'un suréchantillonnage. Ce type de matrice polyphase, correspondant à un banc de filtres de synthèse, est parfois appelé *matrice polyphase de type 2* par opposition avec les *matrices polyphase de type 1* qui correspondent à un banc de filtres d'analyse.

La construction des matrices polyphases d'analyse et de synthèse dans le cas  $M$ -bandes sera présentée en détails à la section 3.1.

On a vu que d'un point de vue pratique la représentation polyphase était efficace car elle permet d'éviter des calculs inutiles. D'un point de vue théorique, cette notation présente également certains avantages. En particulier, cette représentation permet de caractériser simplement l'inversibilité d'un banc de filtres comme le montre le théorème suivant.

**Théorème 1** Soient  $\mathbf{H}(z)$  la matrice polyphase d'un banc de filtres d'analyse  $M$ -bandes et  $\tilde{\mathbf{H}}(z)$  celle du banc de filtres de synthèse alors le système est à reconstruction parfaite si et seulement si :

$$\tilde{\mathbf{H}}(z)\mathbf{H}(z) = cz^{-m_0} \begin{pmatrix} \mathbf{0} & \mathbf{I}_{M-r} \\ z^{-1}\mathbf{I}_r & \mathbf{0} \end{pmatrix},$$

avec  $m_0 \in \mathbb{Z}$ ,  $c \in \mathbb{C}^*$  et  $r \in \{0, \dots, M-1\}$ .

Dans le cadre de cette thèse nous avons utilisé une condition de reconstruction parfaite moins générale donnée par le théorème suivant.

**Théorème 2** Avec les mêmes notations, le système est à reconstruction parfaite si :

$$\tilde{\mathbf{H}}(z)\mathbf{H}(z) = cz^{-m_0}\mathbf{I}_M,$$

avec  $m_0 \in \mathbb{Z}$  et  $c \in \mathbb{C}^*$ .

On peut préciser également quelques cas particuliers :

- Si  $\mathbf{H}(z)$  est inversible, on est dans le cas d'une transformation *bi-orthogonale*.
- Si  $\tilde{\mathbf{H}}(z)$  est polynomiale alors on est dans le cas *inversible RIF*.
- Si  $\tilde{\mathbf{H}}(z) = \mathbf{H}^*(z^{-1})$ , on est alors dans le cas d'une transformation *orthogonale*. Dans ce cas on dit que  $\mathbf{H}(z)$  est para-unitaire. L'étude de la condition de para-unitarité est au centre des travaux sur les *Quadrature mirror filter banks*.

### 2.1.5 Transformées en ondelettes

Les transformées en ondelettes telles qu'on les connaît aujourd'hui ont pris corps dans les années 80 (à la suite de travaux notamment en géophysique). Cet outil permet de représenter un signal à différentes résolutions ce qui se révèle utile pour des applications aussi bien en analyse qu'en débruitage ou en compression (des codeurs basés sur les transformations en ondelettes, comme ceux de type JPEG2000, ont été proposés comme alternative à JPEG permettant de réduire les artéfacts produits par l'effet de bloc en codage d'image [Shapiro, 1993]). La théorie des ondelettes dans le cas continu ne sera pas abordée dans ce travail (on pourra consulter à ce sujet : [Flandrin, 1998]), nous nous contenterons donc de présenter ici le principe des ondelettes discrètes. Pour plus d'informations concernant la construction des bases d'ondelettes nous renvoyons aux ouvrages : [Daubechies, 1992; Meyer, 1992; Mallat, 1998].

Notons que l'ouvrage [Heil, Walnut, 2006] regroupe une grande partie des articles fondateurs sur la théorie des ondelettes.

#### 2.1.5-a) Principe et implémentation par banc de filtres

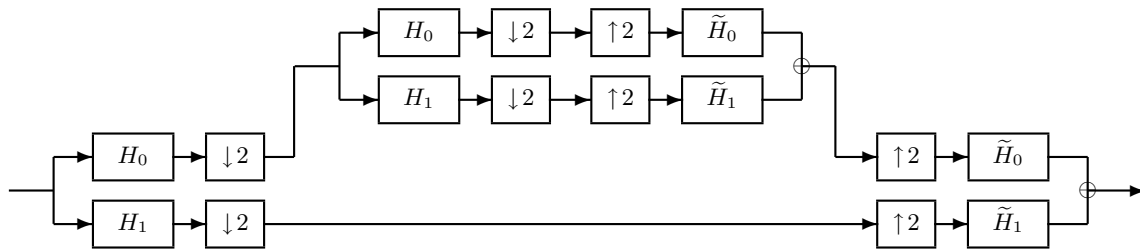


FIGURE 2.7 – Exemple de décomposition/reconstruction par ondelettes.

Une transformée en ondelettes est une analyse multirésolution d'un signal (ou d'une image). Autrement dit, le signal analysé est représenté à différentes échelles permettant de capturer des détails plus ou moins grossiers. Les différentes résolutions peuvent être vues comme des sous-espaces vectoriels  $(V_j)_{j \in \mathbb{Z}} \subset L^2(\mathbb{R})$  emboîtés. L'analyse multirésolution d'un signal  $f$  consiste alors à réaliser des projections de  $f$  sur  $(W_j)_{j \in \mathbb{Z}}$ , les sous-espaces vectoriels complémentaires orthogonaux des  $(V_j)_{j \in \mathbb{Z}}$ .

Dans la pratique, une méthode d'implémentation des décompositions en ondelettes discrètes se fait suivant un schéma de bancs de filtres appliqués en cascade. Ce schéma utilisé dans le cas de filtres RIF est représenté sur la figure 2.7. Il existe également des implémentations dans le domaine fréquentiel [Rioul, Duhamel, 1992] permettant d'utiliser des filtres RII et d'obtenir des gains en temps de calcul.

Le principe est d'appliquer d'abord un banc de filtres  $[H_0, H_1]$  sur les données. Le résultat du filtrage par  $H_0$  (qui sera supposé ici être un filtre passe bas) représente les coefficients d'approximation et le résultat par  $H_1$  (supposé passe haut) représente les coefficients de détail du signal. On applique de nouveau ce banc de filtres sur le résultat

du filtrage par  $H_0$  et cette opération est répétée un nombre fini de fois. À chaque étage de la décomposition en ondelettes on change en fait d'échelle dans la représentation du signal analysé.

Si l'on sait exprimer un banc de filtres inverse de  $[H_0, H_1]$ , le schéma de synthèse se construit simplement de proche en proche, en utilisant une structure en cascade symétrique à celle utilisée pour l'analyse. La figure 2.7 présente également la partie de reconstruction.

Si le banc de filtres inverse s'exprime sous la forme  $[H_0^*, H_1^*]$  alors la transformée en ondelettes associée est dite *orthogonale*. On peut montrer qu'une transformée dyadique réelle en ondelettes ne peut pas être à la fois à support compact, utiliser des filtres symétrique et être orthogonale (à l'exception de la transformée en ondelettes de Haar, présentée dans l'exemple 3). En traitement du signal, et plus particulièrement pour le traitement d'image, les propriétés de compacité du support et de symétrie peuvent être importantes. Afin de conserver ces deux propriétés, il est possible de travailler avec des transformées dites *bi-orthogonales* [Cohen *et al.*, 1992].

Les transformées en ondelettes discrètes sont devenues depuis leur création des outils très populaires et ont été employées pour plusieurs types d'applications, par exemple en débruitage, détection ou déconvolution. Elles sont tout particulièrement utiles pour représenter les singularités d'un signal ou d'une image via une analyse des informations à différentes échelles. En fonction de la longueur des filtres sous-jacents et du niveau de décomposition, les ondelettes discrètes introduisent une certaine quantité de recouvrement entre des échantillons proches, tout en demeurant orthogonales ou presque orthogonales (cas bi-orthogonal).

**Exemple 3** Pour illustrer la transformée en ondelettes, nous allons reprendre le banc de filtres de Haar présenté à l'exemple 2 et considérer le signal suivant :

$$x = (1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1).$$

Le filtrage par  $H_0$  retourne le résultat :  $\mathbf{a}_1 = (\frac{2}{\sqrt{2}}, \frac{2}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}}, \frac{2}{\sqrt{2}})$  et celui par  $H_1$  :  $\mathbf{d}_1 = (0, 0, 0, 0, -\frac{1}{\sqrt{2}}, 0)$ .

On applique à nouveau ces filtres sur  $\mathbf{a}_1$ , le résultat passe bas, on obtient alors pour  $H_0$  :  $\mathbf{a}_2 = (2, 0, \frac{3}{2})$  et pour  $H_1$  :  $\mathbf{d}_2 = (0, 0, -\frac{1}{4})$ .

Finalement, le vecteur transformé de  $x$  sur deux niveaux de décomposition est :  $(\mathbf{a}_2, \mathbf{d}_2, \mathbf{d}_1)$ .

### 2.1.5-b) Transformées en ondelettes $M$ -bandes

Dans la section précédente, nous avons vu comment définir un banc de filtres  $M$ -bandes, la transformée en ondelettes présentée ici peut s'étendre naturellement au cas  $M$ -bandes : on applique au signal le banc de filtres  $[H_0, \dots, H_{M-1}]$  puis on réapplique ce banc de filtres au résultat par le filtre passe bas  $H_0$ .

Ce type de décomposition permet d'obtenir une décomposition plus fine en fréquence. D'un point de vue théorique, elle présente l'intérêt de laisser plus de liberté pour le choix des filtres, permettant par exemple de générer des ondelettes à support compact, orthogonales et utilisant des filtres symétriques [Steffen *et al.*, 1993], ce qui n'est pas possible dans le cas d'ondelettes dyadiques.

### 2.1.5-c) Transformées en paquet d'ondelettes

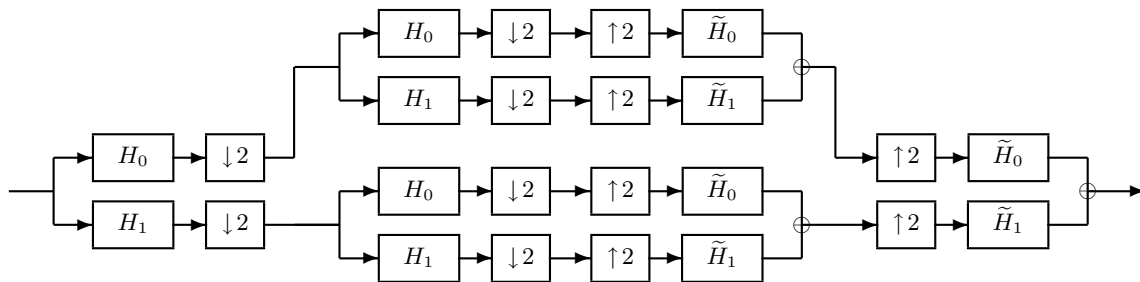


FIGURE 2.8 – Exemple de décomposition/reconstruction par paquet d'ondelettes.

La décomposition en paquet d'ondelettes [Coifman 92] est apparue afin de mieux s'adapter aux caractéristiques temps-fréquence des signaux. Le principe est que les étapes successives de la décomposition ne se font pas uniquement sur les coefficients après le filtrage passe bas mais également sur les coefficients résultant du filtrage passe haut. Ce type de transformation peut être représenté par le schéma de la figure 2.8.

Encore une fois, pour avoir reconstruction parfaite il suffit de connaître un inverse  $[\tilde{H}_0, \tilde{H}_1]$  du banc de filtres d'analyse  $[H_0, H_1]$  et d'appliquer le banc de filtre de synthèse comme indiqué sur le schéma de la figure 2.8.

## 2.2 Redondance et suréchantillonnage

Les transformées rappelées dans ce qui précède étaient considérées dans le cas non-redondant, ou "échantillonnées de manière critique" (aussi appelé décimation critique), c'est-à-dire que le nombre d'échantillons du signal d'origine et le nombre de coefficients après transformation sont essentiellement les mêmes (dans le cas de transformées complexes, comme la TFD sur des données réelles, il faut prendre en compte les symétries éventuelles induites par la transformation). Il a été reconnu très tôt que si l'échantillonnage critique est bien adapté à la compression de données, son intérêt est plus limité pour d'autres applications comme le débruitage ou l'analyse. Ces limites peuvent être liées au signal ou à la transformation.

Du point de vue du signal, des artéfacts peuvent être introduits dans les données traitées, ce qui peut être par exemple le cas lors du débruitage avec des ondelettes discrètes. Les informations les plus fines du signal analysé, qui ne sont pas révélées par les ondelettes discrètes, sont souvent plus visibles en utilisant des ondelettes continues. Ainsi, en considérant l'exemple 3, illustrant la transformée en ondelettes d'un signal simple présentant deux discontinuités, on peut voir que seule la seconde discontinuité est représentée dans les coefficients de détails. En reprenant le même exemple, mais avec une version du signal translatée d'un coefficient, on verrait apparaître dans les détails la première discontinuité, tandis que la seconde disparaîtrait.

Les équivalents en discret des transformées en ondelettes continues ont été découverts à plusieurs reprises sous des noms différents : quasi-continues, stationnaires, à



recouvrement maximal, redondantes, non-décimées, invariante par translation (ou par décalage), sur-complètes, implémentées par l'algorithme "à trous" ou par *cycle spinning*. Ces méthodes sont présentées, par exemple, dans [Coifman, Donoho, 1995; Nason, Silverman, 1995; Pesquet *et al.*, 1996; Fowler, 2005]. Dans le même esprit, d'autres unions de bases offrent une quasi-invariance par translation, c'est le cas de la transformée en arbre dual [Selesnick *et al.*, 2005; Chaux, 2006].

Du point de vue de la transformée, les contraintes imposées lors de la conception des paires de filtres d'ondelettes permettant d'obtenir la reconstruction parfaite et l'orthogonalité réduisent considérablement le nombre d'ondelettes proposant de bonnes propriétés. Les bases orthogonales choisies pour l'étape d'analyse fixent de manière stricte l'étape de synthèse/reconstruction. C'est pourquoi les ondelettes bi-orthogonales ont été proposées. Elles permettent de concilier une décomposition adaptée aux données (par exemple en présentant plusieurs moments nuls) et de bonnes propriétés de reconstruction (par exemple en étant robuste aux perturbations dues à la quantification). Ce type de transformation permet donc de bien prendre en compte l'asymétrie des rôles des phases d'analyse et de synthèse.

**Remarque 2.1** *Souvent les termes redondance et suréchantillonnage sont confondus. Ce ne sont pourtant pas des synonymes. En effet, le suréchantillonnage ne signifie pas qu'une information est répétée plusieurs fois, mais donne simplement une indication quant au nombre d'échantillons après transformation. On peut illustrer cette différence avec l'exemple suivant. Soit le vecteur  $\mathbf{v} = (1, 2, 3)^\top$ , et les matrices :*

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad \mathbf{M}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

On a  $\mathbf{w}_1 = (\mathbf{M}_1 \mathbf{v})^\top = (1, 1, 2, 2, 3, 3)$  et  $\mathbf{w}_2 = (\mathbf{M}_2 \mathbf{v})^\top = (1, 0, 2, 0, 3, 0)$ . Il est clair que dans le vecteur  $\mathbf{w}_2$  l'information du vecteur  $\mathbf{v}$  n'apparaît qu'une fois, i.e. la transformation décrite par  $\mathbf{M}_2$  n'est pas redondante. Dans le vecteur  $\mathbf{w}_1$ , on retrouve deux fois le vecteur  $\mathbf{v}$ , on peut donc dire que la transformation associée à  $\mathbf{M}_1$  est redondante. En revanche, les vecteurs  $\mathbf{w}_1$  et  $\mathbf{w}_2$  comptent 6 échantillons contre 3 dans le vecteur  $\mathbf{v}$ , les deux transformations peuvent être dites suréchantillonnées.

### 2.2.1 Trames

Une possibilité pour s'abstraire de ces restrictions consiste à employer des trames à la place des bases. La notion de trame est apparue dans l'article [Duffin, Schaeffer, 1952] pour étudier certains problèmes liés à l'analyse de séries de Fourier non harmoniques. D'une certaine manière, la notion de trame est une extension du concept de base d'un espace vectoriel.

Cet outil n'a guère été employé en traitement du signal jusqu'au milieu des années 1980, où les trames furent utilisées pour la constructions de transformées en ondelettes discrètes suréchantillonnée [Daubechies *et al.*, 1986]. Il existe de nombreux exemples de

trames d'ondelettes avec de bonnes propriétés, fondées par exemple sur des fonctions *auto-similaires* [Ron, Shen, 1998]. Selesnick a également proposé des ondelettes de plus haute densité basées sur l'ajout d'un filtre non décimé aux paires de filtres d'ondelettes [Selesnick, 2001; Selesnick, 2006].

Rappelons la définition d'une trame. Soit  $\mathcal{H}$  un espace de Hilbert et notons  $\langle \cdot, \cdot \rangle$  son produit. On dit qu'une famille  $(\phi_k)_{k \in I}$ , où  $I \subset \mathbb{N}$  est un ensemble d'indices, est une trame s'il existe un couple  $(\mu, \nu) \in (\mathbb{R}_+^*)^2$  tel que :

$$\forall f \in \mathcal{H}, \quad \mu \|f\|^2 \leq \sum_{k \in I} |\langle f, \phi_k \rangle|^2 \leq \nu \|f\|^2.$$

Les réels  $\mu$  et  $\nu$  sont respectivement appelés bornes inférieure et supérieure de la trame. Si, de plus,  $\mu = \nu$  la trame est alors dite *ajustée*.

On peut montrer que toute famille  $(\phi_k)_{k \in I}$  formant une trame possède au moins une trame duale associée :  $(\tilde{\phi}_k)_{k \in I}$  vérifiant la propriété suivante :

$$\forall f \in \mathcal{H}, \quad \sum_k \langle f, \phi_k \rangle \tilde{\phi}_k = \sum_k \langle f, \tilde{\phi}_k \rangle \phi_k = f.$$

On voit donc que la connaissance d'une trame et d'une trame duale associée peut permettre de décomposer un signal et de le reconstruire parfaitement. En raison de la redondance dans le domaine transformé, plusieurs reconstructions à partir des coefficients sont, en général, possibles à partir des trames duales [Daubechies, 1992]. Les degrés de liberté additionnels sont alors traditionnellement employés pour assurer un certain nombre de moments nuls ou une forme de régularité de Sobolev. Le choix d'une reconstruction n'est pas direct : la forme canonique de la trame duale pour une trame d'ondelettes peut très bien ne pas former un système d'ondelettes [Daubechies, Han, 2002].

Pour plus de détails concernant les trames, on pourra consulter [Daubechies, 1992; Casazza, 2000].

### 2.2.2 Bancs de filtres suréchantillonnés

Des structures en banc de filtres avec un échantillonnage régulier augmentent encore le nombre de degrés de liberté des transformations locales inversibles et permettent d'améliorer les traitements. Les bancs de filtres suréchantillonnés se présentent également sous différents noms : systèmes d'analyse/synthèse généralisés (general Analysis-Synthesis Systems) [Kellermann, 1988], transformée de Fourier discrète à saut, méthode par recouvrement-addition (qui sera présentée dans le chapitre 7) ou transformée de Fourier généralisée, TFD suréchantillonnée ou sous-décimée, bancs de filtres modulés en sinus, cosinus ou complexes, transformées à recouvrement complexes, transformées à recouvrement pseudo bi-orthogonales généralisées, etc. Remarquons d'emblée que les bancs de filtres suréchantillonnés inversibles fournissent des exemples de trames discrètes. Cette association avec les trames a permis par exemple d'étudier la reconstruction parfaite [Bölcskei, Hlawatsch, 1998; Bölcskei *et al.*, 1998; Cvetković, Vetterli, 1998a] et de relier les trames et les matrices polyphase comme nous le rappellerons à la section 2.2.2-b).

En laissant de côté les transformées de Gabor dans le contexte suréchantillonné (pour plus de détails : [Bastiaans, 1982; Feichtinger *et al.*, 1995; Xia, 1996; Gröchenig, Zimmerman, 2001; Durak, Arikan, 2003; Kaiblinger, 2005] ou [Søndergaard, 2007]) et la transformée de Fourier à court-terme [Allen, 1977; Portnoff, 1978; Portnoff, 1980; Crochiere, 1980; Cvetković, 2000], les bancs de filtres suréchantillonnés ont reçu une grande attention aussi bien du point de vue théorique [Bernardini, 2007] qu'applicatif dans les dix dernières années. La flexibilité dans leur conception, leur bonne sélectivité fréquentielle et leur robustesse au bruit, aux distorsions et au repliement spectral les ont rendues très adaptées aux filtrages adaptatifs par sous bande pour des applications en débruitage [Labeau *et al.*, 2005], suppression d'écho [Kellermann, 1988; Dumitrescu *et al.*, 2006], formation de voies (beamforming), quantification [Goyal *et al.*, 2001], construction de codage multi-antennes [Hassibi *et al.*, 2001], noise shaping [Bölcskei, Hlawatsch, 1997], codage à descriptions multiples [Goyal *et al.*, 2001; Bernardini *et al.*, 2006], traitement de signaux audio [Malvar, 1999], égalisation de canal [Pun, Nguyen, 2007; Ihalainen *et al.*, 2007; Johansson, Löwenborg, 2007] ou codage de canal [Weiß *et al.*, 2006].

Deux problèmes principaux doivent être pris en compte lors de l'utilisation de bancs de filtres suréchantillonnés : l'existence d'un inverse pour le banc de filtres d'analyse et la détermination d'un banc de filtres de synthèse optimum (au sens où il offre de bonnes propriétés pour la reconstruction). Le fait d'avoir gagné encore en liberté, par rapport au cas des ondelettes, rend la complexité de la conception encore plus élevée. Différentes méthodes ont été envisagées pour aborder ces questions. Voyons quelques exemples représentatifs de la très riche littérature sur le sujet.

### 2.2.2-a) Utilisation de fenêtres

Plusieurs travaux se sont intéressés au cas de bancs de filtres modulés via l'étude d'une seule ou plusieurs fenêtres utilisées en analyse et/ou synthèse.

Dans [Yui *et al.*, 2004], des bancs de filtres basés sur des TFD fenêtrées par des fenêtres de Hamming, Kaiser ou Dolf-Chebyshev sont analysés et optimisés pour réduire les problèmes de distortion et de repliement spectral. Cvetković et Vetterli [Cvetković, Vetterli, 1998b] se placent également dans un cadre n'utilisant qu'une seule fenêtre et emploient les trames ajustées de Weyl-Heisenberg pour proposer des fenêtres possédant une meilleure sélectivité fréquentielle que dans le cas de l'échantillonnage critique. Un autre exemple de construction de bancs de filtres redondants par étude des fenêtres se trouve dans [Ueng, Scharf, 1996]. Dans cet article, les auteurs proposent de considérer le cas de fenêtres variables et construisent des bancs de filtres de synthèse en utilisant la théorie des trames. Citons également un article récent de Mansour sur des bancs de filtres suréchantillonnés de type TFD dont il optimise la fenêtre de synthèse suivant différents critères [Mansour, 2007].

### 2.2.2-b) Utilisation de la formulation polyphase

Des formulations plus générales, fondées sur la factorisation de la représentation polyphase du banc de filtres suréchantillonné avec des contraintes supplémentaires ont également été proposées. Dans le cas redondant, la matrice polyphase devient rectangulaire rendant l'étude de l'inversibilité plus complexe.

Ainsi dans [von Borries *et al.*, 2001], les auteurs explorent le cas où le facteur  $k'$  de suréchantillonnage du banc de filtres est rationnel. Ils montrent en particulier que dans le cas de BdF modulés réels, il n'est possible d'inverser le banc de filtres que si  $k' \in \mathbb{N}$ . Dans [von Borries, Burrus, 2004], von Borries et Burrus s'intéressent à la construction de matrices polyphases vérifiant la condition de paraunitarité et contraignent les BdF résultant à posséder la propriété de phase linéaire. Encore dans le cadre des BdF à phase linéaire, la reconstruction parfaite avec des bancs de filtres réels (correspondant à des transformées à recouvrement) est étudiée dans [Gan, Ma, 2003] en prenant en compte les symétries des filtres.

Tanaka dans [Tanaka, 2006], s'est intéressé à un cas particulier : la construction d'un banc de filtres de synthèse pour des filtres à RIF de longueur égale au double du facteur de sous-échantillonnage. Il propose une étape d'optimisation matricielle pour obtenir des filtres présentant une bonne atténuation fréquentielle. Riel *et al.* ont proposé dans [Riel *et al.*, 2004] d'utiliser une méthode de type *lifting* pour construire des BdF redondants modulés réels à reconstruction parfaite dans le cas para-unitaire. Ainsi ils obtiennent des filtres présentant, également, une bonne atténuation dans le domaine fréquentiel.

Dans [Bölcskei *et al.*, 1998], les auteurs étudient le lien pouvant exister, pour certains BdF, entre la matrice polyphase et un opérateur de trame correspondant à ce banc de filtres. Ils montrent en particulier que si un BdF suréchantillonné correspond à une trame de  $\ell^2(\mathbb{R})$  alors il est possible de paramétrer les BdF de synthèse permettant la reconstruction parfaite. Une analyse des valeurs propres de la matrice polyphase permet alors d'obtenir les bornes de la trame et ils se servent de ce résultat pour construire une trame de synthèse la plus ajustée possible.

Repartant de cette analyse, Chai *et al.* ont proposé récemment [Chai *et al.*, 2007] une construction basée sur une représentation spatiale des états des bancs de filtres. Cette construction offre des méthodes numériques efficaces pour estimer les bornes et calculer un BdF de synthèse.

Dans ces travaux, la conception peut donc mettre en œuvre différents types d'optimisations, utilisant la régularité des filtres ou des fonctions de coût exploitant la forme des filtres (visant, par exemple, à améliorer l'atténuation dans les sous-bandes ou le gain de codage).

Les travaux évoqués précédemment étudient le cas, classique, où le banc de filtres d'analyse est appliqué avant celui de synthèse. Pour certaines applications (par exemple pour les transmultiplexeurs utilisés dans les communications DSL - *digital subscriber line*) il peut être intéressant de regarder le cas où les rôles des bancs de filtres d'analyse et de synthèse sont inversés. Là encore, l'étude des matrices polyphases permet de construire des bancs de filtres suréchantillonnés et complexes offrant la propriété de reconstruction parfaite [Siclet *et al.*, 2006].

Enfin, d'autres travaux [Kalker *et al.*, 1995; Park *et al.*, 1997; Zhou, Do, 2005] proposent d'employer des outils algébriques comme les bases de Gröbner pour travailler sur les matrices polyphase. Nous verrons plus en détails au chapitre 5 les méthodes utilisant les bases de Gröbner, qui présentent l'avantage d'être directement adaptables au cas multidimensionnel.

### 2.2.2-c) Reconstruction quasi-parfaite

De nombreuses applications en traitement du signal impliquent une étape de traitement des coefficients dans le domaine transformé (par exemple une sélection, un seuillage ou une quantification). Dans ce contexte, imposer une condition de reconstruction parfaite peut sembler inutile, puisque les coefficients sont modifiés pendant le traitement. Ainsi de nombreux travaux visent à relaxer cette contrainte en ne considérant plus qu'une reconstruction *quasi-parfaite* (ou *Near Perfect Reconstruction*) [Rosenbaum, 2007]. Dans [Siohan, Roche, 2000], on voit un exemple de passage d'une reconstruction parfaite à quasi-parfaite dans le cas échantillonné critique. Un avantage évident de ce type de méthodes est qu'elles offrent une plus grande liberté dans la construction des filtres de synthèse (permettant par exemple de choisir des filtres minimisant les problèmes de repliement spectral).

Des constructions ne prenant pas en compte une reconstruction parfaite ont été étudiées par exemple dans : [Harteneck *et al.*, 1999; Wilbur *et al.*, 2003; Dumitrescu *et al.*, 2006; Hermann *et al.*, 2007].

## 2.3 Transformées multidimensionnelles et directionnelles

Comme nous l'avons expliqué dans le chapitre 1, l'un des points de départ de cette thèse était la volonté de traiter des données sismiques bi- ou tri-dimensionnelles. Ces données présentent une forte anisotropie et une analyse directionnelle semble donc indiquée. Dans cette section, nous allons rappeler quelques transformations multidimensionnelles prenant en compte les informations de direction.

### 2.3.1 Bancs de filtres multidimensionnels

En dimension supérieure à 1, on peut traiter les données soit en utilisant des méthodes unidimensionnelles appliquées de manière *séparable* (dans chacune des dimensions) sur des données multidimensionnelles, soit en utilisant directement des transformations multidimensionnelles.

Les bancs de filtres peuvent être construits directement dans le cadre multidimensionnel [Vaidyanathan, 1993]. Dans ce cas, l'échantillonnage des coefficients, qui dans le cas séparable se fait de manière rectangulaire, peut se faire de manière plus libre (par exemple en 2D : quinconce ou hexagonal). Dans le cadre de cette thèse, nous nous sommes limités au cas non-séparable à échantillonnage rectangulaire qui sera abordé au chapitre 5. De nombreux travaux ont porté sur l'étude de l'inversibilité et de la reconstruction parfaite des bancs de filtres non-séparables multidimensionnels [Kovačević, Vetterli, 1992].

Les transformées séparables de par leur construction peuvent donner plus d'importance à certains éléments, favorisant par exemple les directions horizontales et verticales pour une transformation bidimensionnelle ; *a contrario* une transformée non-séparable peut permettre une représentation plus équilibrée et ainsi apporter une directionnalité plus fine. Ainsi en 1992, Bamberger et Smith [Bamberger, Smith, 1992] ont proposé un banc de filtres directionnel 2D décimé de manière critique (avec un échantillonnage en

quinconce) et offrant la propriété de reconstruction parfaite. Ce banc de filtres a ensuite été réutilisé par Do et Vetterli dans la construction des contourlets qui seront brièvement présentées à la section 2.3.4.

### 2.3.2 Bandelettes

La décomposition en bandelettes a été introduite par Le Pennec et Mallat [Le Pennec, 2002]. Elle allie une décomposition en ondelettes et des estimations d'informations à caractère géométrique.

L'estimation de la géométrie se fait en étudiant les contours présents dans une image  $f$ . Un contour est alors vu comme une courbe paramétrique  $\mathcal{C}$  que l'on va caractériser par ses tangentes. Pour ce faire, on recherche les gradients d'importance significative ( $c\grave{a}d$  dont la valeur est supérieure à un certain seuil) dans l'image. Afin d'éviter des problèmes de non différentiabilité, l'image peut être convoluée avec un noyau lissant. Cette étude permet d'approximer les "flots" de l'image en considérant la direction orthogonale au gradient.

Ensuite, une étape de rectification est menée. Elle consiste à déformer la courbe pour la rendre horizontale (respectivement verticale) si elle était plutôt horizontale (respectivement verticale) à l'origine. En notant cette opération  $D$ ,  $Df$  représente alors la version rectifiée de l'image. Enfin, une transformation en ondelettes est appliquée sur  $Df$ . Dans la pratique, il faut dans un premier temps segmenter l'image en régions où la direction du contour n'est jamais verticale (respectivement horizontale).

Les bandelettes de deuxième génération<sup>1</sup> ont ensuite été introduites par Peyré et Mallat [Peyré, 2005]. L'idée est à nouveau de construire une base orthogonale adaptée à une image  $f$ . Cependant, cette fois la recherche de géométrie ne se fait pas sur  $f$  directement mais dans les différentes sous-bandes obtenues après la décomposition en ondelettes de  $f$ . Afin d'améliorer l'étape de détection de la géométrie, les relations inter-échelles peuvent être prises en compte (*via* la construction d'un arbre reliant les coefficients, à chaque niveau de détail, correspondant à la même localisation spatiale).

### 2.3.3 Curvelets

Les curvelets ont d'abord été introduites en 1999 par Donoho et Candès dans un cadre continu dans [Candès, Donoho, 1999a], une approche discrète des curvelets a ensuite été étudiée dans [Donoho, Duncan, 1999]. Les curvelets proposent de décomposer un signal sur une trame fixée, permettant de bien représenter les informations de type géométrique. Cette base permet d'approcher de façon presque optimale des fonctions ayant une régularité géométrique d'ordre  $C^2$ . Contrairement aux bandelettes, les curvelets ne nécessitent pas de phase d'adaptation à l'image ce qui peut présenter un avantage pour certaines applications. Le principe de la transformation peut être résumé en 4 étapes :

- Décomposition en sous-bandes (comme dans le cas d'une transformée en ondelettes classique). Un objet  $f$  est alors représenté par l'approximation  $P_0 f$  et les

1. Un lien vers une toolbox permettant de réaliser une transformée en bandelettes de deuxième génération est disponible sur la page : <http://www.ceremade.dauphine.fr/~peyre/bandelets/>



sous-bandes  $(\Delta_1 f, \Delta_2 f, \dots)$  contenant chacune des détails dont la taille est de l'ordre de  $2^{-2s}$ .

- Partition de chaque sous bande  $\Delta_s f$  en carrés dyadiques, par multiplication par des fenêtres carrées  $w_{Q(s,k_1,k_2)}$  centrées autour de :

$$Q(s, k_1, k_2) = [k_1/2^s; (k_1 + 1)/2^s) \times [k_2/2^s; (k_2 + 1)/2^s).$$

- Renormalisation des carrés dyadiques :  $w_{Q(s,k_1,k_2)} \Delta_s f$ , au sein de chaque sous-bande. Soit l'opérateur :

$$(T_{Q(s,k_1,k_2)} f)(x_1; x_2) = 2^s f(2sx_1 - k_1; 2sx_2 - k_2),$$

la renormalisation se fait en considérant :  $g_{Q(s,k_1,k_2)} = (T_{Q(s,k_1,k_2)})^{-1}(w_{Q(s,k_1,k_2)} \Delta_s f)$ .

- Analyse dans un système Ridgelet orthonormal des carrés dyadiques renormalisés.

L'idée derrière la transformée par curvelets est que, en réalisant un fenêtrage, des segments d'une courbe lisse vont sembler quasiment droits et ainsi peuvent être décomposés de manière efficace par une transformée en Ridgelet. En effet, une propriété intéressante des Ridgelets dans  $\mathbb{R}^2$  est qu'elles sont constantes le long de lignes décrites par  $x_1 \cos(\theta) + x_2 \sin(\theta) = \text{constante}$  et sont des ondelettes le long de la direction orthogonale. Les Ridgelets offrent donc une représentation creuse (*i.e.* sur peu de coefficients) pour des objets lisses et rectilignes. Pour plus de détails concernant la transformée en Ridgelet on pourra consulter [Candès, Donoho, 1999b; Do, 2001].

L'un des problèmes des curvelets a été de passer du cadre continu dans lequel elles ont été définies au domaine discret. Les premières méthodes discrètes comme celle présentée ci-dessus, souffraient d'une redondance élevée pouvant être difficile à gérer pour certaines applications et d'un temps de calcul élevé. Plus récemment, des implémentations<sup>2</sup> rapides et moins redondantes ont été proposées [Candès *et al.*, 2005].

### 2.3.4 Contourlets

La transformée en contourlets<sup>3</sup> a été introduite au milieu des années 2000 par Do et Vetterli [Do, Vetterli, 2005]. Le but est de décomposer une image à l'aide d'une combinaison de filtres multi-échelles et directionnels. Le principe a été de mélanger deux méthodes : les pyramides laplaciennes avec des bancs de filtres directionnels non-séparables. Les bancs de filtres directionnels employés dans cette méthode sont ceux qui avaient été introduits par Bamberger et Smith dans [Bamberger, Smith, 1992].

La méthode de pyramide laplacienne a été introduite en 1983 par Burt et Adelson [Burt, Adelson, 1983]. Elle permet de réaliser une décomposition multi-échelle d'un signal. La décomposition par pyramide laplacienne se fait de la manière suivante : tout d'abord une version passe bas (convolution par un filtre noté  $H$ ) et sous-échantillonnée  $f_b$  de l'image originale  $f$  est générée, puis par suréchantillonnage et utilisation d'un filtre

2. Les codes pour réaliser des transformées en curvelets rapides sont disponibles sur le site : <http://www.curvelet.org/>.

3. Les codes des transformées en contourlets et des pyramides laplaciennes sont disponibles pour Matlab sur le site <http://www.ifp.uiuc.edu/~minhdo/software/>

$G$ , inverse de  $H$ , on construit une version de "prédiction"  $f_p$ , enfin on calcule la différence  $f - f_p$ , ce qui revient à considérer une image passe bande. On passe d'une itération à la suivante en appliquant à nouveau cette méthode sur la version passe bas  $f_b$ .

Du point de vue des trames et des bancs de filtres suréchantillonnés, on peut montrer que l'utilisation de filtres orthogonaux permet de ramener la pyramide laplacienne à une trame ajustée. Dans ce cas, en utilisant l'opérateur correspondant à la trame duale optimale une reconstruction robuste en présence de bruit est possible.

La décomposition par contourlets se fait en combinant ces deux méthodes en appliquant d'abord la pyramide laplacienne puis le banc de filtres directionnel sur chaque sous-bande passe haut. Cette combinaison peut s'écrire sous une forme équivalente via un banc de filtres qui est appelé banc de filtres directionnel pyramidal. La pyramide laplacienne et le banc de filtres directionnel étant des opérations inversibles, il suffit d'appliquer successivement les inverses de ces méthodes pour obtenir la reconstruction parfaite de l'image d'origine.

Notons que, tout comme dans le cas des curvelets, la trame de décomposition est fixée à l'avance et ne nécessite donc pas d'adaptation à chaque image. Enfin, contrairement aux trames de curvelets, ces fonctions de base ont été construites directement dans un domaine discret grâce à l'utilisation de bancs de filtres permettant ainsi une application informatique directe de la méthode.

## 2.4 Conclusion

Depuis le milieu des années 1970, de très nombreuses méthodes mettant en jeu des techniques liées aux bancs de filtres ont été développées. Le tour d'horizon que nous avons proposé dans ce chapitre ne reflète qu'une partie de la vaste littérature sur le sujet. Ainsi, parmi les transformations introduites ces dernières années, on aurait également pu présenter plusieurs autres exemples : les pyramides orientables (*steerable pyramids*) [Freeman, Adelson, 1991], la décomposition en arbre dual ([Kingsbury, 2001] en 2-bandes et [Chaux, 2006] en  $M$ -bandes), les phaselets [Gopinath, 2003] ou encore les grouplets [Mallat, 2008].



---

## Chapitre 3

# Inversion de bancs de filtres 1D suréchantillonnés

---

Dans ce chapitre nous allons nous intéresser aux transformées à recouvrement unidimensionnelles suréchantillonnées et plus particulièrement à leur inversibilité. L'analyse du problème d'inversion, sous forme polyphase, va nous permettre de proposer un algorithme pour vérifier l'inversibilité d'un banc de filtres d'analyse à réponse impulsionnelle finie donné. Nous allons également présenter une méthode permettant le calcul explicite d'un banc de filtres de synthèse inverse. L'étude sera enfin complétée par un cas intéressant pour les applications : celui des bancs de filtres de synthèse à reconstruction réelle présentant une symétrie hermitienne. Deux méthodes permettant de générer de tels filtres de synthèse seront exposées.

### 3.1 Formulation polyphase et redondance

#### 3.1.1 Notations utilisées

Comme nous l'avons vu dans le chapitre 2, les transformées à recouvrement peuvent être vues par le biais d'un formalisme en *banc de filtres*, illustré par la figure 2.5. Le banc de filtres d'analyse est composé de  $M$  filtres à réponses impulsionnelles finies, notées  $(h_i)_{0 \leq i < M}$ . Dans la suite, nous noterons par  $N$  le facteur de décimation, celui-ci sera supposé plus petit que la longueur des filtres. Plus précisément, les filtres seront supposés de longueur  $kN$  avec  $k \in \mathbb{N}^*$ . Autrement dit, on considère une transformation à recouvrement agissant sur des paquets de données de  $k$  blocs de taille  $N$ , chaque paquet ayant  $k - 1$  blocs en commun avec le paquet suivant.

Un signal  $(x(n))_{n \in \mathbb{Z}}$  est décomposé à l'aide d'un tel banc de filtres. Le facteur de décimation étant de  $N$ , la redondance introduite dans la transformation est :  $M/N = k'$ . Dans cette thèse, nous nous sommes intéressés au cas suréchantillonné, qui correspond à  $k' > 1$ . Les  $M$  sorties du banc de filtres d'analyse sont notées  $(y_i(n))_{0 \leq i < M}$ .

Avec ces notations, les sorties du banc de filtres d'analyse s'expriment, pour tout  $i \in \{0, \dots, M-1\}$  et  $n \in \mathbb{Z}$  par :

$$y_i(n) = \sum_p h_i(p)x(Nn-p) = \sum_{\ell} \sum_{j=0}^{N-1} h_i(N\ell+j)x(N(n-\ell)-j). \quad (3.1)$$

### 3.1.2 Formulation polyphase

Le chapitre 2 nous a permis de présenter certains avantages, aussi bien d'un point de vue pratique que théorique, de la formulation polyphase. Voyons maintenant les expressions polyphases avec lesquelles nous allons travailler.

Soient  $\mathbf{H}(\ell) = (h_i(N\ell+j))_{0 \leq i < M, 0 \leq j < N}$ ,  $\ell \in \{0, \dots, k-1\}$  les  $k$  matrices obtenues à partir des réponses impulsionnelles du banc de filtres d'analyse. On définit également le vecteur polyphase du signal à partir du signal d'entrée  $x(n)$  :  $\forall n \in \mathbb{Z}$ ,  $\mathbf{x}(n) = (x(Nn-j))_{0 \leq j < N}$ . L'équation (3.1) peut être réécrite sous une forme plus concise en utilisant un produit de convolution matriciel :

$$\mathbf{y}(n) = (y_0(n), \dots, y_{M-1}(n))^T = \sum_{\ell=0}^{k-1} \mathbf{H}(\ell) \mathbf{x}(n-\ell) = (\mathbf{H} * \mathbf{x})(n). \quad (3.2)$$

En utilisant les notations précédente, l'équation (3.2) peut s'écrire sous la forme :  $\mathbf{y}[z] = \mathbf{H}[z] \mathbf{x}[z]$ , où la matrice de taille  $M \times N$  :

$$\mathbf{H}[z] = \sum_{\ell=0}^{k-1} \mathbf{H}(\ell) z^{-\ell}$$

est la matrice polyphase du banc de filtres d'analyse et  $\mathbf{x}[z]$  (respectivement  $\mathbf{y}[z]$ ) est la transformée en  $z$  de  $(\mathbf{x}(n))_{n \in \mathbb{Z}}$  (respectivement  $(\mathbf{y}(n))_{n \in \mathbb{Z}}$ ).

De la même manière, on peut définir la matrice polyphase du banc de filtre de synthèse :  $\tilde{\mathbf{H}}[z] = \sum_{\ell} \tilde{\mathbf{H}}(\ell) z^{-\ell}$ , qui est telle que :

$$\tilde{\mathbf{x}}[z] = \tilde{\mathbf{H}}[z] \mathbf{y}[z], \quad (3.3)$$

où le vecteur polyphase du signal de sortie du banc de filtres de synthèse  $(\tilde{\mathbf{x}}(n))_{n \in \mathbb{Z}}$  est défini de la même manière que pour le signal d'entrée  $(\mathbf{x}(n))_{n \in \mathbb{Z}}$ .

**Remarque 3.1** *Le fait de supposer les filtres d'analyse RIF causaux ne limite pas la généralité de ce qui va suivre concernant la recherche de l'inversibilité et les calculs explicites d'inverses.*

*En effet si les filtres d'analyse ne sont pas causaux, le facteur de recouvrement s'exprime sous la forme  $k = k_1 + k_2 + 1$  et on a une matrice polyphase de la forme :*

$$\begin{aligned} \mathbf{H}[z] &= \sum_{\ell=-k_1}^{k_2} \mathbf{H}(\ell) z^{-\ell} = z^{k_1} \sum_{\ell=0}^{k_2+k_1} \mathbf{H}(\ell - k_1) z^{-\ell} \\ &= z^{k_1} \sum_{\ell=0}^{k-1} \mathbf{H}'(\ell) z^{-\ell} = z^{k_1} \mathbf{H}'[z], \end{aligned}$$

avec  $\mathbf{H}'(\ell) = \mathbf{H}(\ell - k_1)$ , pour tout  $0 \leq \ell < k$ . La matrice polyphase obtenue est donc, à un décalage près, de la même forme que celle que nous avons considérée. La reconstruction parfaite étant définie à un décalage près, d'après le théorème 2, ce décalage ne pose pas de problème pour la caractérisation de l'inversibilité.

Dans la suite le banc de filtres de synthèse sera considéré non-causal et ce décalage pourra donc y être intégré en posant :

$$\tilde{\mathbf{H}}[z] = z^{-k_1} \tilde{\mathbf{H}}'[z]$$

où  $\tilde{\mathbf{H}}'[z]$  est une matrice polyphase de synthèse inverse de  $\mathbf{H}'[z]$ .

### 3.1.3 Banc de filtres de synthèse

La matrice polyphase  $\tilde{\mathbf{H}}[z] \in \mathbb{C}[z, z^{-1}]^{N \times M}$  étant celle du banc de filtres de synthèse, voyons comment obtenir les expressions des réponses impulsionnelles du banc de filtres de synthèse.

D'une part, on déduit de l'équation (3.3) que :

$$\forall n \in \mathbb{Z} \text{ et } \forall i \in \{0, \dots, N-1\}, \quad \tilde{x}(nN - i) = \sum_{j=0}^{M-1} \sum_{\ell=-\infty}^{\infty} \tilde{H}_{i,j}(n - \ell) y_j(\ell). \quad (3.4)$$

D'autre part, l'expression directe de la reconstruction faisant intervenir les réponses impulsionnelles est :

$$\tilde{x}(m) = \sum_{j=0}^{M-1} \sum_{\ell=-\infty}^{\infty} \tilde{h}_j(m - N\ell) y_j(\ell), \quad \forall m \in \mathbb{Z}.$$

En posant :  $m = nN - i$  où  $i \in \{0, \dots, N-1\}$  et  $n \in \mathbb{Z}$ , ceci conduit à :

$$\tilde{x}(nN - i) = \sum_{j=0}^{M-1} \sum_{\ell=-\infty}^{\infty} \tilde{h}_j(N(n - \ell) - i) y_j(\ell). \quad (3.5)$$

Finalement, par identification des équations (3.4) et (3.5), on déduit que :

$$\tilde{\mathbf{H}}(\ell) = \left( \tilde{h}_j(N\ell - i) \right)_{0 \leq i < N, 0 \leq j < M}, \quad \ell \in \mathbb{Z}. \quad (3.6)$$

Ces expressions restent valides pour tout banc de filtres suréchantillonné.

## 3.2 Inversibilité d'un banc de filtres d'analyse

L'un des avantages de la représentation polyphase des bancs de filtres est d'offrir une caractérisation simple pour l'inversibilité : un banc de filtres d'analyse est inversible si sa matrice polyphase est elle-même inversible (à gauche) [Vaidyanathan, 1993]. La matrice  $\mathbf{H}[z]$  que l'on souhaite étudier appartient à  $\mathbb{C}[z, z^{-1}]^{M \times N}$ , l'anneau des matrices polynomiales de Laurent de dimension  $M \times N$ .

Un point important de ce travail est que le but ici n'est pas de chercher n'importe quel inverse mais un inverse qui soit une matrice polynomiale appartenant à  $\mathbb{C}[z, z^{-1}]^{N \times M}$ . Autrement dit, on veut obtenir un banc de filtres de synthèse qui soit également à réponse impulsionnelle finie.

Pour répondre à cette question, Park et Vetterli [Park *et al.*, 1997] ont proposé une méthode utilisant les bases de Gröbner pour caractériser l'inversibilité des matrices polynomiales. Nous reviendrons sur cette méthode dans le cas multidimensionnel au chapitre 5. Ici, nous allons présenter une autre méthode qui est bien adaptée au cas de bancs de filtres unidimensionnels dans lequel nous nous sommes placés.

Le théorème suivant donne une condition nécessaire et suffisante pour qu'une matrice polynomiale soit inversible à gauche, et constitue un point crucial pour la méthode que nous avons proposée. Une preuve de ce théorème peut être trouvée dans : [Fornasini, Valcher, 1997].

**Théorème 3** Soit  $\mathbf{H}[z] \in \mathbb{C}[z, z^{-1}]^{M \times N}$  une matrice polynomiale avec  $M > N$ . Les conditions suivantes sont alors équivalentes :

1. Les déterminants mineurs maximaux de la matrice  $\mathbf{H}[z]$  sont mutuellement premiers.
2.  $\mathbf{H}[z]$  est inversible à gauche : il existe  $\tilde{\mathbf{H}}[z] \in \mathbb{C}[z, z^{-1}]^{N \times M}$  telle que  $\tilde{\mathbf{H}}[z]\mathbf{H}[z] = \mathbf{I}_N$ .

**Remarque 3.2** Une matrice vérifiant la première condition de ce théorème est appelée en anglais : "coprime".

La première condition de ce théorème est d'un intérêt théorique limité ; en effet calculer tous les déterminants mineurs maximaux et rechercher ensuite si ces polynômes sont premiers mutuellement ou non est difficilement réalisable dans le cadre général de bancs de filtres d'analyse quelconques. Cependant cette condition est intéressante dans la pratique puisqu'elle est directement utilisable pour vérifier si une matrice polyphase est inversible à gauche ou non.

En utilisant la procédure suivante, nous pouvons tester numériquement si la première condition du théorème est satisfaite ou non :

- ① Extraction d'une sous-matrice de taille maximale  $\mathbf{H}_e[z]$  de  $\mathbf{H}[z]$ .
- ② Calcul du polynôme  $\det(\mathbf{H}_e[z])$ , et recherche de l'ensemble  $\mathcal{S}_e$  de ses racines.
- ③ Extraction d'une nouvelle sous-matrice de taille maximale de  $\mathbf{H}[z]$ . Retirer de  $\mathcal{S}_e$  les éléments qui ne sont pas racines du déterminant de cette sous-matrice.
- ④ L'étape ③ est répétée jusqu'à ce que toutes les sous-matrices maximales aient été extraites ou bien que  $\mathcal{S}_e = \emptyset$ .
- ⑤ Si  $\mathcal{S}_e = \emptyset$  alors la matrice polyphase  $\mathbf{H}[z]$  est inversible à gauche ; sinon elle ne l'est pas.

Cette méthode est aisée à programmer et conduit à la vérification d'au plus  $\binom{M}{N}$  polynômes. En pratique, si la matrice polyphase est inversible à gauche le nombre de polynômes à considérer est bien plus petit que  $\binom{M}{N}$ , cette borne étant atteinte uniquement lorsque la matrice n'est pas inversible.

**Exemple 4** Nous allons illustrer cet algorithme à l'aide d'un exemple très simple : on travaille avec le banc de filtres d'analyse défini à la section 3.5.2 avec comme paramètres  $N = 2$ ,  $k = 2$ ,  $k' = 3/2$ . La matrice polyphase  $\mathbf{H}[z]$  appartient donc à  $\mathbb{C}[z, z^{-1}]^{3 \times 2}$ . Afin de vérifier que cette matrice polynomiale est inversible à gauche par une matrice de  $\mathbb{C}[z, z^{-1}]^{2 \times 3}$ , nous considérons les sous-matrices carrées de taille maximale de  $\mathbf{H}[z]$  :

$$\mathbf{H}_\ell[z] = ((\mathbf{H}[z])_{j,k})_{(j,k) \in A_\ell \times \{1,2\}}, \quad \ell \in \{1, 2, 3\}$$

avec  $A_1 = \{1, 2\}$ ,  $A_2 = \{2, 3\}$  et  $A_3 = \{1, 3\}$ . Puisque  $k = 2$ , les éléments de la matrice  $\mathbf{H}[z]$  sont de degré 1 et le degré de  $\det(\mathbf{H}_\ell[z])$  est donc au plus de 2. Le tableau 3.1 présente les deux racines (approchées) de chacun des polynômes  $\det(\mathbf{H}_\ell[z])$ . On peut voir qu'il n'y a pas de racine commune aux trois déterminants simultanément. En conséquence, la matrice  $\mathbf{H}[z]$  est inversible à gauche par une matrice appartenant à  $\mathbb{C}[z, z^{-1}]^{2 \times 3}$ .

	Première racine	Seconde racine
$\det(\mathbf{H}_1[z])$	$0.1045 - 0.9945i$	$-0.9135 - 0.4067i$
$\det(\mathbf{H}_2[z])$	$0.1045 - 0.9945i$	$0.9135 + 0.4067i$
$\det(\mathbf{H}_3[z])$	$0.8090 + 0.5878i$	$0.8090 - 0.5878i$

TABLE 3.1 – Racines des trois déterminants mineurs maximaux de  $\mathbf{H}[z]$ .

### 3.3 Calcul explicite d'un banc de filtres de synthèse

La méthode proposée à la section 3.2, garantit uniquement l'existence d'un inverse à gauche, correspondant à un banc de filtres de synthèse RIF, mais ne fournit pas son expression. Nous allons maintenant voir comment effectuer le calcul explicite d'une matrice polyphase inverse. Remarquons que la matrice  $\mathbf{H}[z]$  que l'on cherche à inverser étant rectangulaire, l'inverse ne sera pas unique en général. Dans cette partie, nous allons supposer que la matrice  $\mathbf{H}[z]$  est inversible.

Le but étant d'obtenir la propriété de reconstruction parfaite, on recherche une matrice  $\tilde{\mathbf{H}}[z]$  appartenant à  $\mathbb{C}[z, z^{-1}]^{N \times M}$  telle que  $\tilde{\mathbf{H}}[z] \mathbf{H}[z] = \mathbf{I}_N$ . La matrice  $\tilde{\mathbf{H}}[z]$  appartenant à  $\mathbb{C}[z, z^{-1}]^{N \times M}$ , il existe des entiers  $(p_1, p_2) \in \mathbb{N}^2$ , inconnus pour le moment, tels que la matrice polyphase du banc de filtres de synthèse s'écrit :  $\tilde{\mathbf{H}}[z] = \sum_{\ell=-p_1}^{p_2} \tilde{\mathbf{H}}(\ell) z^{-\ell}$ . On notera dans la suite :  $p = p_1 + p_2 + 1$ .

En remplaçant, on obtient :

$$\tilde{\mathbf{H}}[z] \mathbf{H}[z] = \sum_{\ell=-p_1}^{p_2} \tilde{\mathbf{H}}(\ell) z^{-\ell} \sum_{\ell=0}^{k-1} \mathbf{H}(\ell) z^{-\ell} = \sum_{\ell=-p_1}^{p_2+k-1} \mathbf{U}(\ell) z^{-\ell},$$

où

$$\mathbf{U}(\ell) = \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \tilde{\mathbf{H}}(s) \mathbf{H}(\ell - s). \quad (3.7)$$

La propriété de reconstruction parfaite est alors équivalente à  $\mathbf{U}(\ell) = \delta_\ell \mathbf{I}_N$ , ce qui conduit au système linéaire suivant :

$$\mathcal{H}\tilde{\mathcal{H}} = \mathcal{U} \quad (3.8)$$

où

$$\begin{aligned} \tilde{\mathcal{H}}^\top &= [\tilde{\mathbf{H}}(-p_1), \dots, \tilde{\mathbf{H}}(p_2)] \in \mathbb{C}^{N \times pM}, \\ \mathcal{U}^\top &= [\mathbf{0}_{N, p_1 N} \ \mathbf{I}_N \ \mathbf{0}_{N, (p_2 + k - 1)N}] \in \mathbb{R}^{N \times (k + p - 1)N}, \end{aligned} \quad (3.9)$$

et

$$\mathcal{H}^\top = \begin{pmatrix} \mathbf{H}(0) & \dots & \mathbf{H}(k-1) & & 0 \\ & \ddots & & \ddots & \\ 0 & & \mathbf{H}(0) & \dots & \mathbf{H}(k-1) \end{pmatrix} \in \mathbb{C}^{pM \times (k + p - 1)N}.$$

Dans la pratique  $p_1$  et  $p_2$  sont inconnus, mais on sait que le système (3.8) peut être résolu pour au moins un couple  $(p_1, p_2)$ . Pour déterminer un tel couple, on essaye donc de résoudre ce système en faisant varier  $p = p_1 + p_2 + 1$ , en partant de  $p = 1$  et en recherchant tous les couples  $(p_1, p_2) \in \mathbb{N}^2$  vérifiant cette égalité. Pour un  $p$  donné, on peut ainsi considérer les couples  $(p_1, p_2)$  appartenant à :  $\{(p-1, 0), (p-2, 1), \dots, (0, p-1)\}$ . La figure 3.1 illustre la manière dont sont parcourus les couples  $(p_1, p_2)$  quand  $p$  augmente. De cette manière, on est assuré que la méthode trouvera une matrice polyphase inverse d'ordre minimum. Ceci implique que les filtres de synthèse ont une longueur minimale. La matrice  $\mathcal{H}$  étant rectangulaire, la résolution du système se fait en utilisant la pseudo-inverse de Moore-Penrose [Penrose, 1955].

En continuant à augmenter la valeur de  $p$ , et donc en recherchant d'autres couples  $(p_1, p_2)$ , on peut ainsi essayer de générer d'autres bancs de filtres de synthèse inverses.

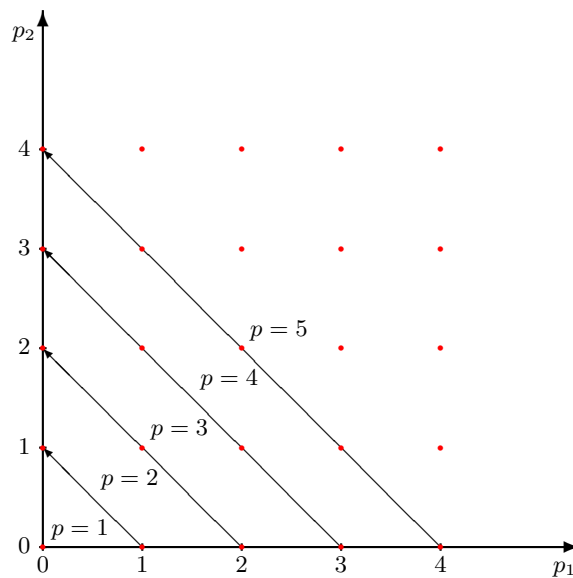


FIGURE 3.1 – Parcours des couples  $(p_1, p_2)$  quand  $p$  augmente.

**Remarque 3.3** Nous avons vu que l'ordre  $p$  étant directement lié à la longueur des filtres de synthèse, et donc, dans la pratique, également au temps de calcul, il serait possible de vouloir fixer une borne supérieure a priori pour  $p$ . La méthode de calcul présentée ci-dessus peut alors permettre de répondre à une question d'existence légèrement différente de celle traitée à la section 3.2 : pour un banc de filtres d'analyse donné, existe-t-il un banc de filtres de synthèse RIF inverse dont les filtres ont une longueur bornée par une constante fixée a priori ?

### 3.4 Le cas symétrique

L'utilisation de la transformée de Fourier sur des signaux réels conduit à une décomposition symétrique pour les parties réelles et anti-symétrique pour les parties imaginaires. Cette propriété est appelée la symétrie hermitienne. Réciproquement, si dans le domaine fréquentiel le signal présente une symétrie hermitienne alors le signal reconstruit sera réel.

Cette propriété est très intéressante pour des applications, par exemple de débruitage, sur des données réelles qui consistent à modifier des coefficients dans le domaine fréquentiel avant de reconstruire. En effet, il est aisé de conserver la symétrie hermitienne des coefficients après une opération de seuillage. Une propriété permettant d'assurer que le signal reconstruit soit réel dès que les coefficients présentent une symétrie hermitienne est donc très utile dans la pratique.

Pour inclure cette propriété dans les filtres de synthèse on doit vérifier que tous coefficients  $(y_i(n))_{0 \leq i < M}$ , tels que  $y_i(n) = 0$  si  $(i, n) \neq (j_f, n_f)$  ou  $(i, n) \neq (M-1-j_f, n_f)$  avec  $j_f \in \{0, \dots, M-1\}$  et  $n_f \in \mathbb{Z}$ , vérifiant :  $y_{j_f}(n_f) = \overline{y_{M-1-j_f}(n_f)}$ , permet de reconstruire un signal réel.

Le signal reconstruit s'écrit dans ce cas :

$$\begin{aligned} \tilde{x}(m) &= \sum_{j=0}^{M-1} \sum_{\ell=-\infty}^{\infty} \tilde{h}_j(m - N\ell) y_j(\ell) \\ &= \tilde{h}_{j_f}(m - n_f N) y_{j_f}(n_f) + \tilde{h}_{M-1-j_f}(m - n_f N) \overline{y_{j_f}(n_f)}. \end{aligned}$$

Une condition nécessaire et suffisante pour que  $\tilde{x}(m) \in \mathbb{R}$  pour tout  $y_{j_f}(n_f) \in \mathbb{C}$ , est que  $\tilde{h}_{j_f}(m - n_f N) = \overline{\tilde{h}_{M-1-j_f}(m - n_f N)}$ . Cette condition doit être vérifiée pour tout couple  $(j_f, n_f)$ . La condition sur les filtres de synthèse est donc que :

$$\forall j \in \{0, \dots, M-1\} \text{ et } \forall n \in \mathbb{Z}, \quad \tilde{h}_j(n) = \overline{\tilde{h}_{M-1-j}(n)}.$$

En utilisant l'équation (3.6) on peut réécrire cette condition sous forme matricielle :

$$\forall \ell \in \{-p_1, \dots, p_2\}, \quad \tilde{\mathbf{H}}(\ell) = \overline{\tilde{\mathbf{H}}(\ell)} \mathbf{J}_M. \quad (3.10)$$

où  $\mathbf{J}_M$  est la matrice d'échange de taille  $M \times M$

$$\mathbf{J}_M = \begin{pmatrix} 0 & & 1 \\ & \ddots & \\ 1 & & 0 \end{pmatrix}.$$

On a supposé ici que les coefficients après transformation d'un signal réel présentent une symétrie hermitienne. Autrement dit, pour tout signal réel  $(x(n))_{n \in \mathbb{Z}}$ , et pour tout couple  $(j_f, n_f) \in \{0, \dots, M-1\} \times \mathbb{Z}$ , on souhaite avoir en sortie du banc de filtres d'analyse :

$$y_{j_f}(n_f) = \overline{y_{M-j_f-1}(n_f)}.$$

On peut réécrire cette condition sous la forme :

$$\sum_m h_{j_f}(m)x(Nn_f - m) = \sum_m \overline{h_{M-j_f-1}(m)}x(Nn_f - m).$$

En considérant des signaux d'entrée  $x$  identiquement nuls sauf en un point, on peut en déduire :

$$h_{j_f}(n) = \overline{h_{M-j_f-1}(n)},$$

ce qui est équivalent à :

$$\mathbf{H}(\ell) = \mathbf{J}_M \overline{\mathbf{H}(\ell)}. \quad (3.11)$$

Nous allons supposer que le banc de filtres d'analyse dans le cas symétrique vérifie cette condition, permettant d'assurer qu'après la décomposition d'un signal réel, les coefficients possèdent la propriété de symétrie hermitienne.

### 3.4.1 Une première méthode de construction

En injectant les relations (3.10) et (3.11) dans l'équation (3.7), on obtient :

$$\begin{aligned} \mathbf{U}(\ell) &= \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \widetilde{\mathbf{H}}(s) \mathbf{H}(\ell - s) \\ &= \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \widetilde{\mathbf{H}}(s) \mathbf{J}_M \overline{\mathbf{H}(\ell - s)} \\ &= \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \overline{\widetilde{\mathbf{H}}(s)} \mathbf{J}_M \mathbf{H}(\ell - s). \end{aligned}$$

Ci-dessus, nous avons utilisé le fait que la matrice  $\mathbf{U}(\ell)$  est réelle et donc invariante par opération de conjugaison.

Ceci implique que si  $\widetilde{\mathcal{H}} = [\widetilde{\mathbf{H}}(-p_1), \dots, \widetilde{\mathbf{H}}(p_2)]^\top$  est solution du système linéaire (3.8) alors, sous l'hypothèse de symétrie hermitienne du banc de filtres d'analyse,  $\widetilde{\mathcal{H}}_2 = [\overline{\widetilde{\mathbf{H}}(-p_1)} \mathbf{J}_M, \dots, \overline{\widetilde{\mathbf{H}}(p_2)} \mathbf{J}_M]^\top$  est aussi une solution du système linéaire. Finalement, on en déduit que la somme :  $\widetilde{\mathcal{H}}_0 = \frac{1}{2}(\widetilde{\mathcal{H}} + \widetilde{\mathcal{H}}_2)$  est également une solution. De plus, elle vérifie par construction la condition (3.10).

Nous avons donc proposé ici une méthode pour construire un banc de filtres de synthèse possédant la propriété de symétrie hermitienne. Cette condition est valable en partant d'un banc de filtres de synthèse quelconque. Cette méthode peut être utile dans



le cas où un banc de filtre de synthèse a déjà été calculé par une autre méthode qui ne garantit pas cette propriété (par exemple par le biais des bases de Gröbner).

Le but du prochain chapitre va être d'ajouter une étape d'optimisation sur les bancs de filtres de synthèse pour améliorer certaines caractéristiques (par exemple la sélectivité fréquentielle des filtres de synthèse ou bien la localisation temps-fréquence). Malheureusement, il n'y a aucune raison en général pour que la caractéristique optimisée reste optimum après la manipulation présentée ici. Nous allons donc voir une autre méthode de construction qui sera plus utile pour l'optimisation.

### 3.4.2 Une seconde méthode de construction

Dans cette section, nous allons supposer que les matrices  $\mathbf{H}(\ell)$  du banc de filtres d'analyse satisfont la condition (3.11). Le but est encore de proposer une méthode pour construire un banc de filtres assurant la reconstruction réelle pour des coefficients respectant la symétrie hermitienne (autrement dit satisfaisant à la condition (3.10)), cependant, maintenant nous n'allons plus supposer qu'un banc de filtres de synthèse a déjà été calculé.

#### 3.4.2-a) Cas où $M$ est pair

Dans un premier temps, on peut réécrire les conditions (3.10) et (3.11) de la manière suivante :

$$(3.10) \Leftrightarrow \forall \ell \in \{-p_1, \dots, p_2\}, \begin{cases} \widetilde{\mathbf{H}}(\ell) = \left( \widetilde{\mathbf{H}}_1(\ell) \widetilde{\mathbf{H}}_2(\ell) \right), \widetilde{\mathbf{H}}_1 \in \mathbb{C}^{N \times M'} \text{ et } \widetilde{\mathbf{H}}_2 \in \mathbb{C}^{N \times M'}, \\ \widetilde{\mathbf{H}}_1(\ell) = \overline{\widetilde{\mathbf{H}}_2(\ell)} \mathbf{J}_{M'} \end{cases}$$

$$(3.11) \Leftrightarrow \forall \ell \in \{0, \dots, k-1\}, \begin{cases} \mathbf{H}(\ell) = \begin{pmatrix} \mathbf{H}_1(\ell) \\ \mathbf{H}_2(\ell) \end{pmatrix}, \mathbf{H}_1 \in \mathbb{C}^{M' \times N} \text{ et } \mathbf{H}_2 \in \mathbb{C}^{M' \times N}, \\ \mathbf{H}_1(\ell) = \mathbf{J}_{M'} \mathbf{H}_2(\ell) \end{cases}$$

où  $M' = M/2$ . En utilisant ces conditions dans l'équation (3.7), on obtient :

$$\begin{aligned} \mathbf{U}(\ell) &= \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \widetilde{\mathbf{H}}(s) \mathbf{H}(\ell-s) \\ &= \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \widetilde{\mathbf{H}}_1(s) \mathbf{H}_1(\ell-s) + \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \overline{\widetilde{\mathbf{H}}_1(s)} \mathbf{J}_{M'}^2 \overline{\mathbf{H}_1(\ell-s)}. \end{aligned}$$

Comme  $\mathbf{J}_{M'}^2 = \mathbf{I}_{M'}$ , l'équation ci-dessus peut être vue comme la somme d'une matrice de nombres complexes et de sa conjuguée, ce qui forme une quantité réelle.

On en déduit que :

$$(3.7) \Leftrightarrow \frac{1}{2} \mathbf{U}(\ell) = \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \left( \widetilde{\mathbf{H}}_1^R(s), -\widetilde{\mathbf{H}}_1^I(s) \right) \begin{pmatrix} \mathbf{H}_1^R(\ell-s) \\ \mathbf{H}_1^I(\ell-s) \end{pmatrix},$$

où  $\mathbf{A}^R$  est la partie réelle pour une matrice  $\mathbf{A}$  et  $\mathbf{A}^I$  sa partie imaginaire.

On définit alors les matrices suivantes :

$$\widetilde{\mathcal{H}}_s^\top = \left[ \widetilde{\mathbf{H}}_1^R(-p_1), -\widetilde{\mathbf{H}}_1^I(-p_1), \dots, \widetilde{\mathbf{H}}_1^R(p_2), -\widetilde{\mathbf{H}}_1^I(p_2) \right] \in \mathbb{R}^{N \times pM}, \quad (3.12)$$

et

$$\mathcal{H}_s^\top = \begin{pmatrix} \mathbf{H}_1^R(0) & \cdots & \mathbf{H}_1^R(k-1) & 0 \\ \mathbf{H}_1^I(0) & \cdots & \mathbf{H}_1^I(k-1) & 0 \\ & \ddots & & \\ 0 & & \mathbf{H}_1^R(0) & \cdots & \mathbf{H}_1^R(k-1) \\ 0 & & \mathbf{H}_1^I(0) & \cdots & \mathbf{H}_1^I(k-1) \end{pmatrix} \in \mathbb{R}^{pM \times (k+p_1+p_2)N}.$$

### 3.4.2-b) Cas où $M$ est impair

De nouveau, on peut réécrire les conditions (3.10) et (3.11), mais d'une manière légèrement différente :

$$(3.10) \Leftrightarrow \forall \ell \in \{-p_1, \dots, p_2\}, \begin{cases} \widetilde{\mathbf{H}}(\ell) = \left( \widetilde{\mathbf{H}}_1(\ell) \mathbf{c}_1(\ell) \widetilde{\mathbf{H}}_2(\ell) \right), \widetilde{\mathbf{H}}_1 \in \mathbb{C}^{N \times M'} \text{ et } \widetilde{\mathbf{H}}_2 \in \mathbb{C}^{N \times M'}, \\ \widetilde{\mathbf{H}}_1(\ell) = \widetilde{\mathbf{H}}_2(\ell) \mathbf{J}_{M'} \text{ et } \mathbf{c}_1(\ell) \in \mathbb{R}^N \end{cases}$$

$$(3.11) \Leftrightarrow \forall \ell \in \{0, \dots, k-1\}, \begin{cases} \mathbf{H}(\ell) = \begin{pmatrix} \mathbf{H}_1(\ell) \\ \mathbf{c}_2(\ell)^\top \\ \mathbf{H}_2(\ell) \end{pmatrix}, \mathbf{H}_1 \in \mathbb{C}^{M' \times N} \text{ et } \mathbf{H}_2 \in \mathbb{C}^{M' \times N}, \\ \mathbf{H}_1(\ell) = \mathbf{J}_{M'} \overline{\mathbf{H}_2(\ell)} \text{ et } \mathbf{c}_2(\ell) \in \mathbb{R}^N \end{cases}$$

où  $M' = (M-1)/2$ . En utilisant ces conditions dans l'équation (3.7), on obtient cette fois :

$$\begin{aligned} \mathbf{U}(\ell) = & \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \widetilde{\mathbf{H}}_1(s) \mathbf{H}_1(\ell-s) + \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \overline{\widetilde{\mathbf{H}}_1(s)} \mathbf{J}_{M'}^2 \overline{\mathbf{H}_1(\ell-s)} \\ & + \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \mathbf{c}_1(s) \mathbf{c}_2(\ell-s)^\top. \end{aligned}$$

On utilise à nouveau le fait que  $\mathbf{J}_{M'}^2 = \mathbf{I}_{M'}$ , et donc les deux premiers termes de l'équation précédente peuvent être vus comme la somme d'une matrice de nombres complexes et de sa conjuguée, qui est réelle. On en déduit que :

$$(3.7) \Leftrightarrow \frac{1}{2} \mathbf{U}(\ell) = \sum_{s=\max(\ell-k+1, -p_1)}^{\min(p_2, \ell)} \left( \widetilde{\mathbf{H}}_1^R(s), \frac{\mathbf{c}_1(s)}{\sqrt{2}}, -\widetilde{\mathbf{H}}_1^I(s) \right) \begin{pmatrix} \mathbf{H}_1^R(\ell-s) \\ \frac{\mathbf{c}_2(\ell-s)^\top}{\sqrt{2}} \\ \mathbf{H}_1^I(\ell-s) \end{pmatrix}.$$

On introduit dans ce cas les matrices :

$$\widetilde{\mathcal{H}}_s^\top = \left[ \widetilde{\mathbf{H}}_1^R(-p_1), \frac{\mathbf{c}_1(-p_1)}{\sqrt{2}}, -\widetilde{\mathbf{H}}_1^I(-p_1), \dots, \widetilde{\mathbf{H}}_1^R(p_2), \frac{\mathbf{c}_1(p_2)}{\sqrt{2}}, -\widetilde{\mathbf{H}}_1^I(p_2) \right] \in \mathbb{R}^{N \times pM}, \quad (3.13)$$

et

$$\mathcal{H}_s^\top = \begin{pmatrix} \mathbf{H}_1^R(0) & \cdots & \mathbf{H}_1^R(k-1) & 0 \\ \frac{\mathbf{c}_2(0)^\top}{\sqrt{2}} & \cdots & \frac{\mathbf{c}_2(k-1)^\top}{\sqrt{2}} & 0 \\ \mathbf{H}_1^I(0) & \cdots & \mathbf{H}_1^I(k-1) & 0 \\ & \ddots & & \\ 0 & & \mathbf{H}_1^R(0) & \cdots & \mathbf{H}_1^R(k-1) \\ 0 & & \frac{\mathbf{c}_2(0)^\top}{\sqrt{2}} & \cdots & \frac{\mathbf{c}_2(k-1)^\top}{\sqrt{2}} \\ 0 & & \mathbf{H}_1^I(0) & \cdots & \mathbf{H}_1^I(k-1) \end{pmatrix} \in \mathbb{R}^{pM \times (k+p_1+p_2)N}.$$

### 3.4.2-c) Conclusion

Dans les deux cas, on se ramène à la résolution d'un système linéaire de la même taille que celui de la section 3.3, la seule différence étant ici qu'on travaille dans le domaine réel. On a ainsi :

$$\mathcal{H}_s \widetilde{\mathcal{H}_s} = \mathcal{U}$$

avec :

$$\mathcal{U}^\top = \frac{1}{2} [\mathbf{0}_{N,p_1N} \quad \mathbf{I}_N \quad \mathbf{0}_{N,(p_2+k-1)N}],$$

De la même manière que dans la section 3.3, on tente de résoudre, par pseudo-inverse, ce système pour différentes valeurs du couple  $(p_1, p_2) \in \mathbb{N}^2$  vérifiant  $p = p_1 + p_2 + 1$  en augmentant  $p$  en partant de  $p = 1$ .

## 3.5 Exemples de bancs de filtres

Dans cette section, nous allons présenter quelques exemples de bancs de filtres qui nous serviront à illustrer les méthodes d'inversion et d'optimisation sur les bancs de filtres, ainsi que pour traiter les applications sur des données réelles au chapitre 7.

### 3.5.1 Transformée étendue à recouvrement

Les transformées orthogonales étendues à recouvrement ont été introduites sous le nom de *GenLOT* (generalized linear-phase lapped orthogonal transform) au milieu des années 1990 [de Queiroz *et al.*, 1996] et constituent une généralisation de la TCD et des transformées orthogonales à recouvrement ou LOT (lapped orthogonal transform).

Pour illustrer la méthode d'inversion, nous avons choisi d'utiliser une GenLOT avec  $M = 16$  filtres de 32 coefficients. Les paramètres que nous avons utilisés pour simuler un banc de filtres redondant sont :  $N = 8$ ,  $k = 4$  et  $k' = 2$ . On peut remarquer que ce banc de filtres est réel et ne vérifie pas la condition de symétrie. La figure 3.2 présente les réponses impulsionnelles et fréquentielles de ce banc de filtres.

Avec ces paramètres le banc de filtre est inversible. En utilisant la méthode décrite à la section 3.3, on obtient  $p_1 = 3$  et  $p_2 = 0$  (donc  $p = 4$ ). Les réponses impulsionnelles et fréquentielles du banc de filtres de synthèse réel obtenu par pseudo-inverse sont montrées à la figure 3.3.

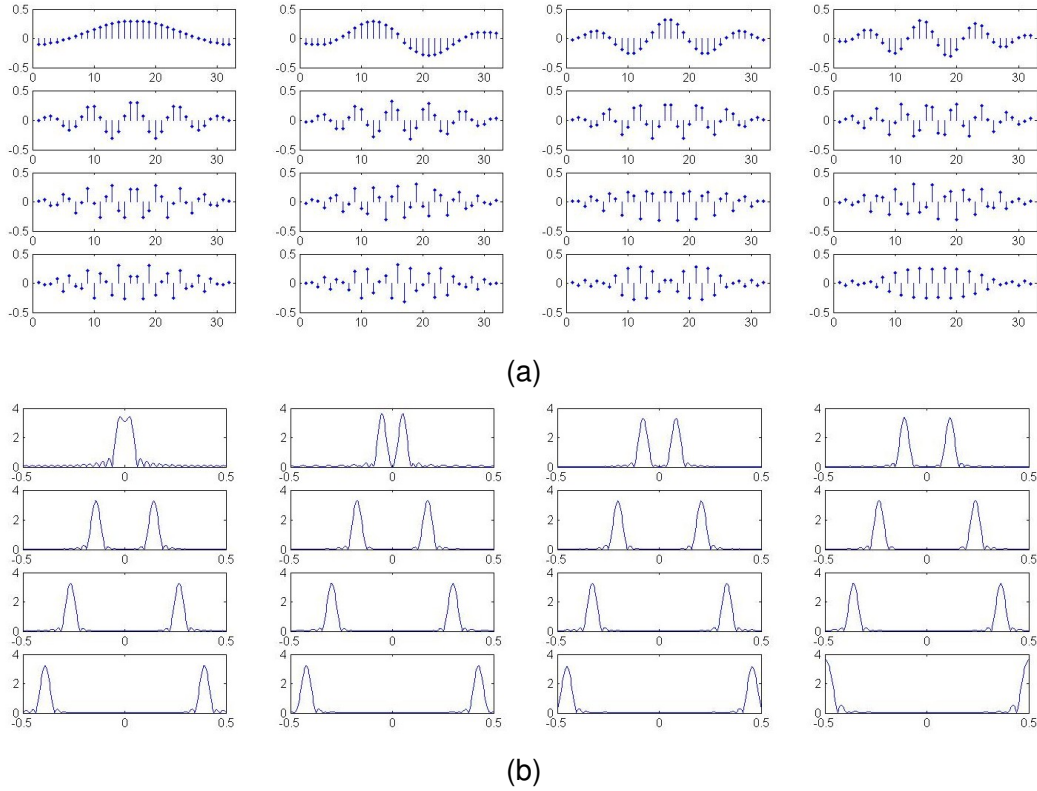


FIGURE 3.2 – (a) Réponses impulsionnelles et (b) réponses fréquentielles d'un banc de filtres d'analyse de type GenLOT.

### 3.5.2 Transformée modulée complexe redondante à recouvrement

Pour illustrer certains résultats théoriques nous avons travaillé avec un banc de filtres d'analyse ayant la forme d'une transformée de Fourier généralisée fenêtrée. Notons également que ce banc de filtres sera utilisés dans plusieurs applications des chapitres suivants. Ce type de banc de filtres a été utilisé dans les travaux de Kingsbury [Young, Kingsbury, 1993] et Malvar [Malvar, 1999] aussi bien pour des applications en vidéo qu'en audio.

Les réponses impulsionnelles de ce banc de filtres sont de la forme :  $h_i(n) = \mathbf{E}(i, n)h_a(n)$ , où

$$\mathbf{E}(i, n) = \frac{1}{\sqrt{k'N}} e^{-j(i - \frac{k'N}{2} + \frac{1}{2})(n - \frac{kN}{2} + \frac{1}{2}) \frac{2\pi}{k'N}},$$

et  $(h_a(n))_{1 \leq n \leq kN}$  est une fenêtre d'analyse non nulle. Plusieurs choix sont possibles pour la fenêtre d'analyse. Nous avons réalisé les applications présentées au chapitre 7 avec la fenêtre :

$$\forall n \in \{1, \dots, kN\}, \quad h_{a1}(n) = \sin\left(\frac{n\pi}{kN+1}\right).$$

Elle avait été utilisée précédemment dans [Malvar, 1999]; un exemple de cette fenêtre est montré sur la figure 3.4(a) (avec  $N = 16$  et  $k = 3$ ).

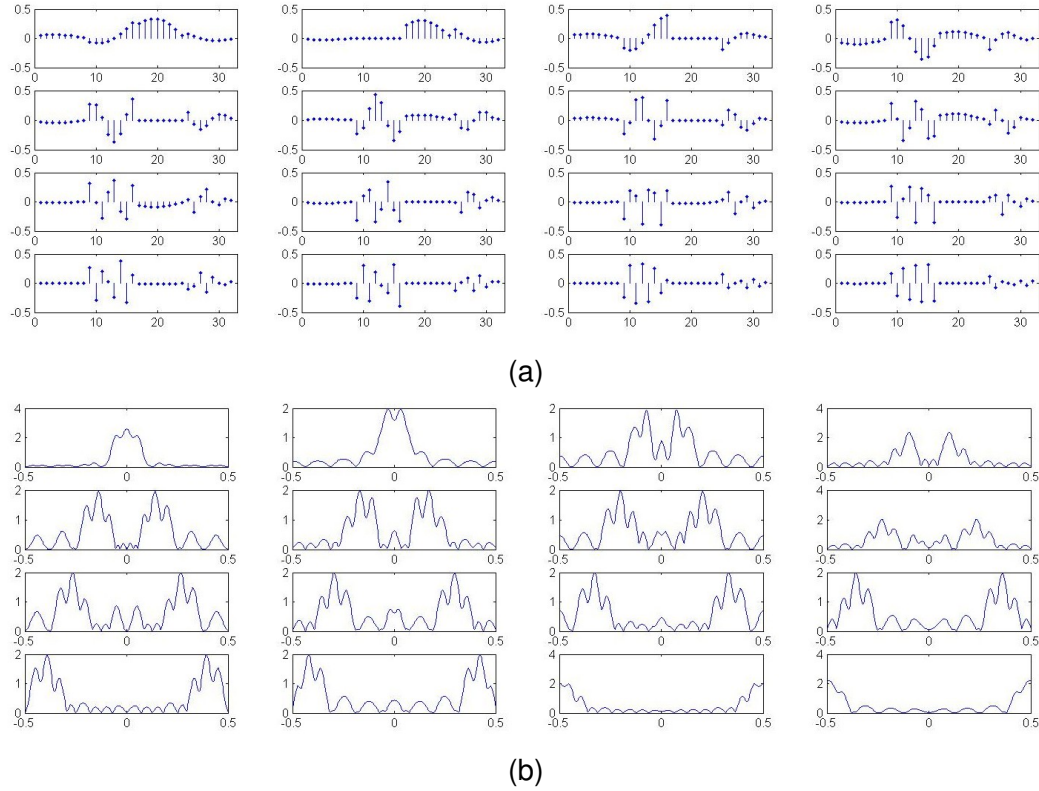


FIGURE 3.3 – (a) Réponses impulsionnelles et (b) réponses fréquentielles du banc de filtres de synthèse obtenu par pseudo-inverse d'un banc de filtres d'analyse de type GenLOT.

Pour illustrer l'inversion et l'optimisation de bancs de filtres nous avons également utilisé une fenêtre d'analyse  $(h_{a_2}(n))_{1 \leq n \leq kN}$  correspondant à un filtre passe bas à phase nulle avec une fréquence de coupure de  $2\pi/kN$ , construit à partir d'une fenêtre de Kaiser. Cette fenêtre est représentée sur la figure 3.4(b) et a été utilisé dans [Mansour, 2007].

On peut voir sur la figure 3.5 les réponses fréquentielles des filtres d'analyse, en utilisant les fenêtres  $h_{a_1}$  et  $h_{a_2}$ , avec  $N = 8$ ,  $k = 3$  et  $k' = 7/4$ . Il est intéressant de noter que ce banc de filtres satisfait à la condition (3.11) ce qui est nécessaire pour la construction de banc de filtres de synthèse à reconstruction réelle à partir de coefficients présentant une symétrie hermitienne.

La méthode de la section 3.2 pour vérifier l'inversibilité a été utilisée sur les bancs de filtres d'analyse présentés précédemment avec comme paramètres :  $N = 8$ ,  $k = 3$  et  $k' = 7/4$ . Le résultat de ce test a été que ces bancs de filtres (avec ces paramètres) sont inversibles.

La méthode de calcul d'un inverse de la section 3.3 (par pseudo-inverse) a d'abord été employée, pour obtenir un premier banc de filtres de synthèse inverse. On a trouvé dans les deux cas les paramètres  $p_1 = 2$  et  $p_2 = 0$ . Les bancs de filtres d'analyse

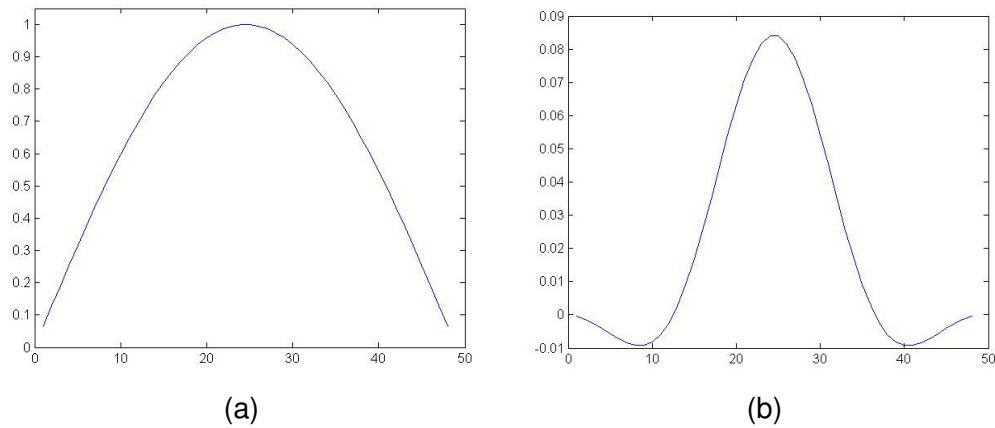
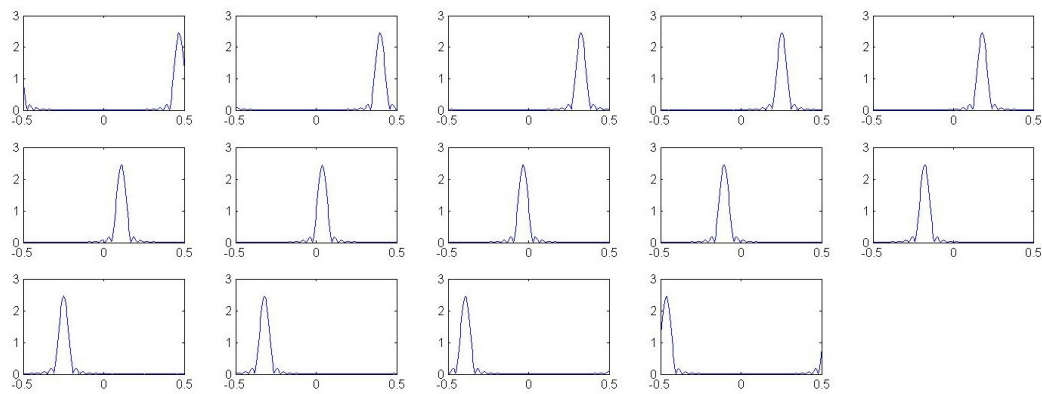


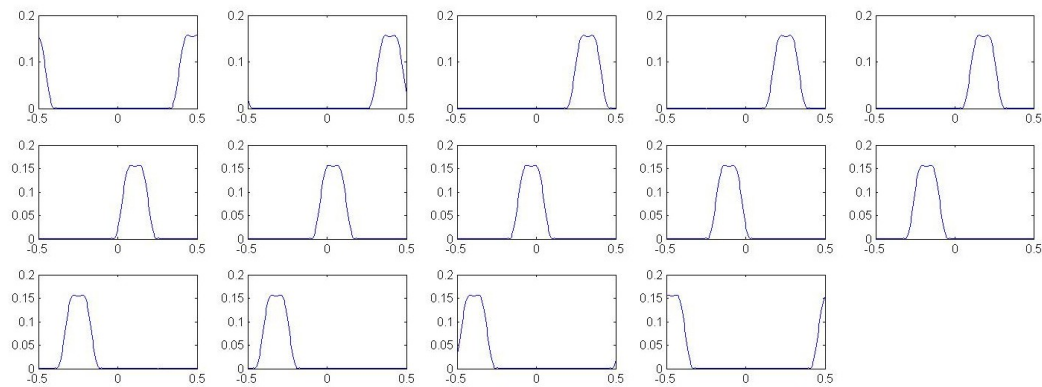
FIGURE 3.4 – Deux exemples de fenêtres d’analyse : (a) fenêtre  $h_{a1}$  et (b) fenêtre  $h_{a2}$ .

vérifiant la condition (3.11), nous avons ensuite utilisé la méthode de construction de filtres de synthèse dans le cas symétrique décrite à la section 3.4.1 en l’appliquant au banc de filtres de synthèse calculé précédemment par pseudo-inverse. Enfin nous avons appliqué la méthode de la section 3.4.2 pour construire directement un autre banc de filtres de synthèse dans le cas symétrique.

Les réponses impulsionnelles et fréquentielles de ces trois bancs de filtres de synthèse sont respectivement montrées sur les figures 3.6 et 3.7 dans le cas où la fenêtre d’analyse est  $h_{a1}$  et sur les figures 3.8 et 3.9 dans le cas où la fenêtre d’analyse est  $h_{a2}$ . Une première remarque est que les bancs de filtres de synthèse obtenus dans le cas symétrique diffèrent suivant la méthode employée. On remarque également, sur la figure 3.6(a), que les filtres inverses présentent une symétrie dans leurs coefficients mais ne sont pas symétriques *entre eux*, contrairement aux versions symétriques des bancs de filtres montrés sur les figures 3.6(b) et (c). On peut enfin voir que ces filtres ne présentent pas nécessairement de bonnes propriétés de sélectivité fréquentielle ou d’atténuation au bord du support (mauvaise localisation temps-fréquence). Nous allons voir dans le chapitre suivant comment améliorer, par optimisation, ces caractéristiques des bancs de filtres.



(a)



(b)

FIGURE 3.5 – Réponses fréquentielles des filtres d'analyse en utilisant (a) la fenêtre  $h_{a1}$  et (b) la fenêtre  $h_{a2}$ .

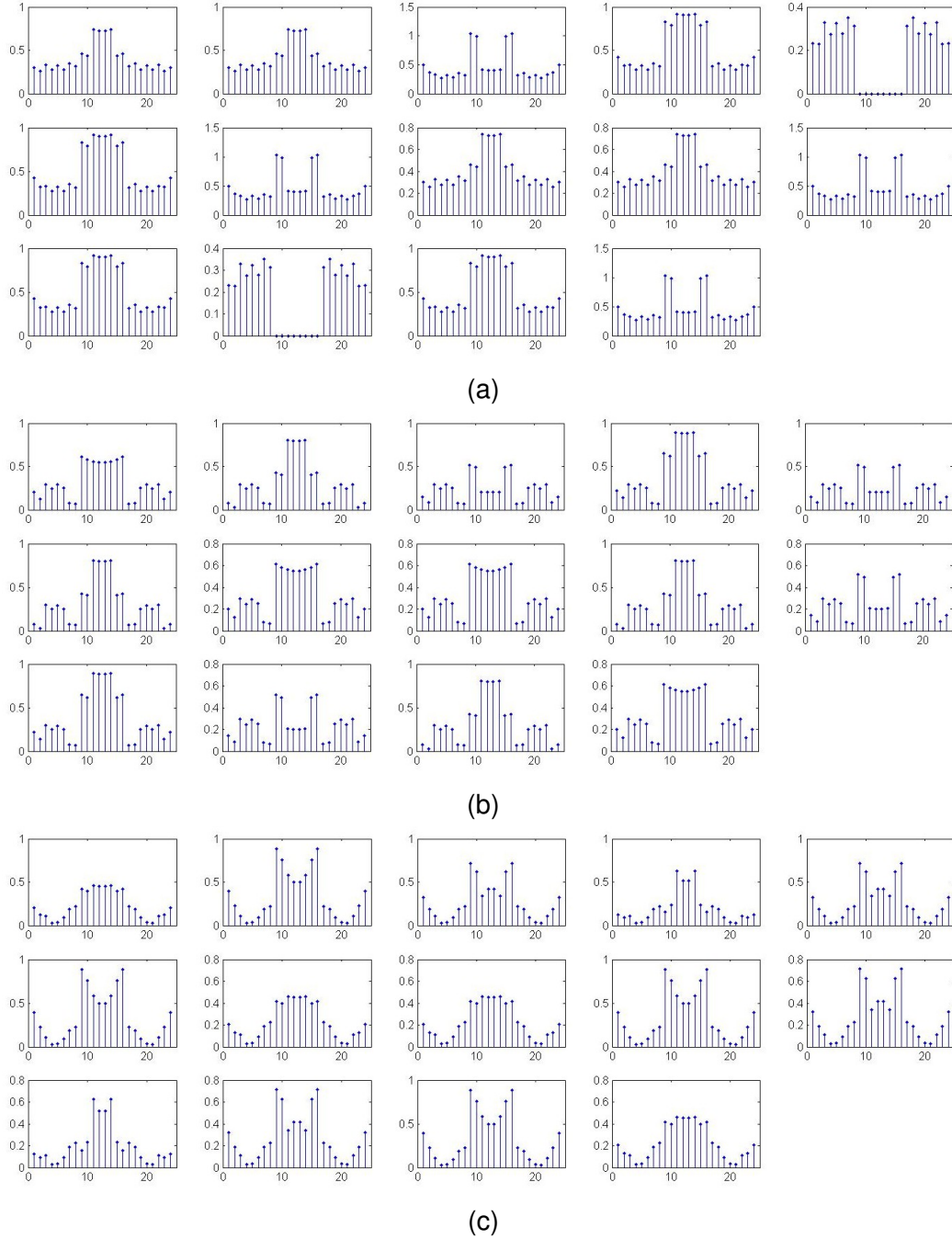


FIGURE 3.6 – Module des réponses impulsionnelles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est  $h_{a_1}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse dans le cas symétrique.



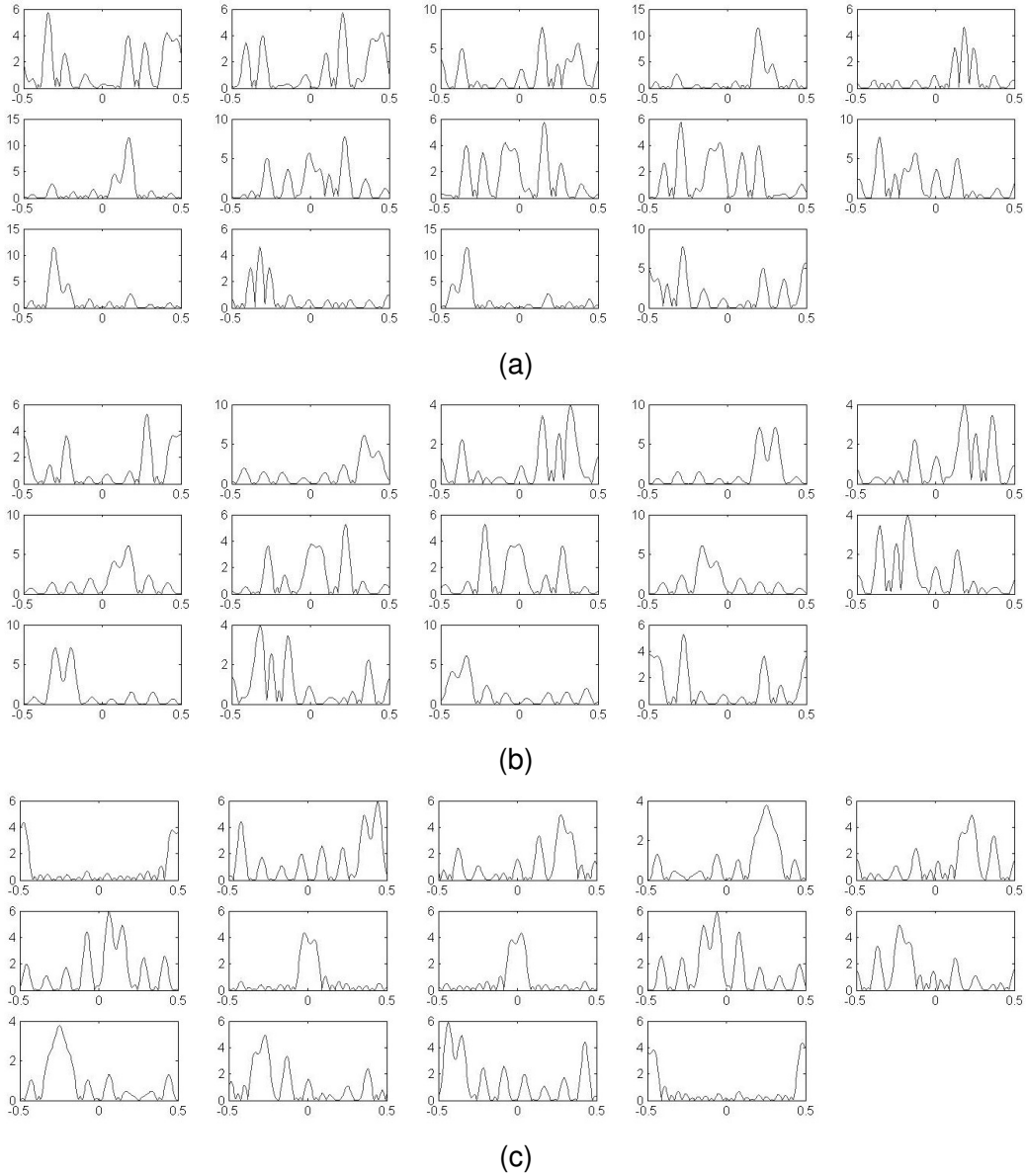


FIGURE 3.7 – Réponses fréquentielles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est  $h_{a1}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse symétrique.

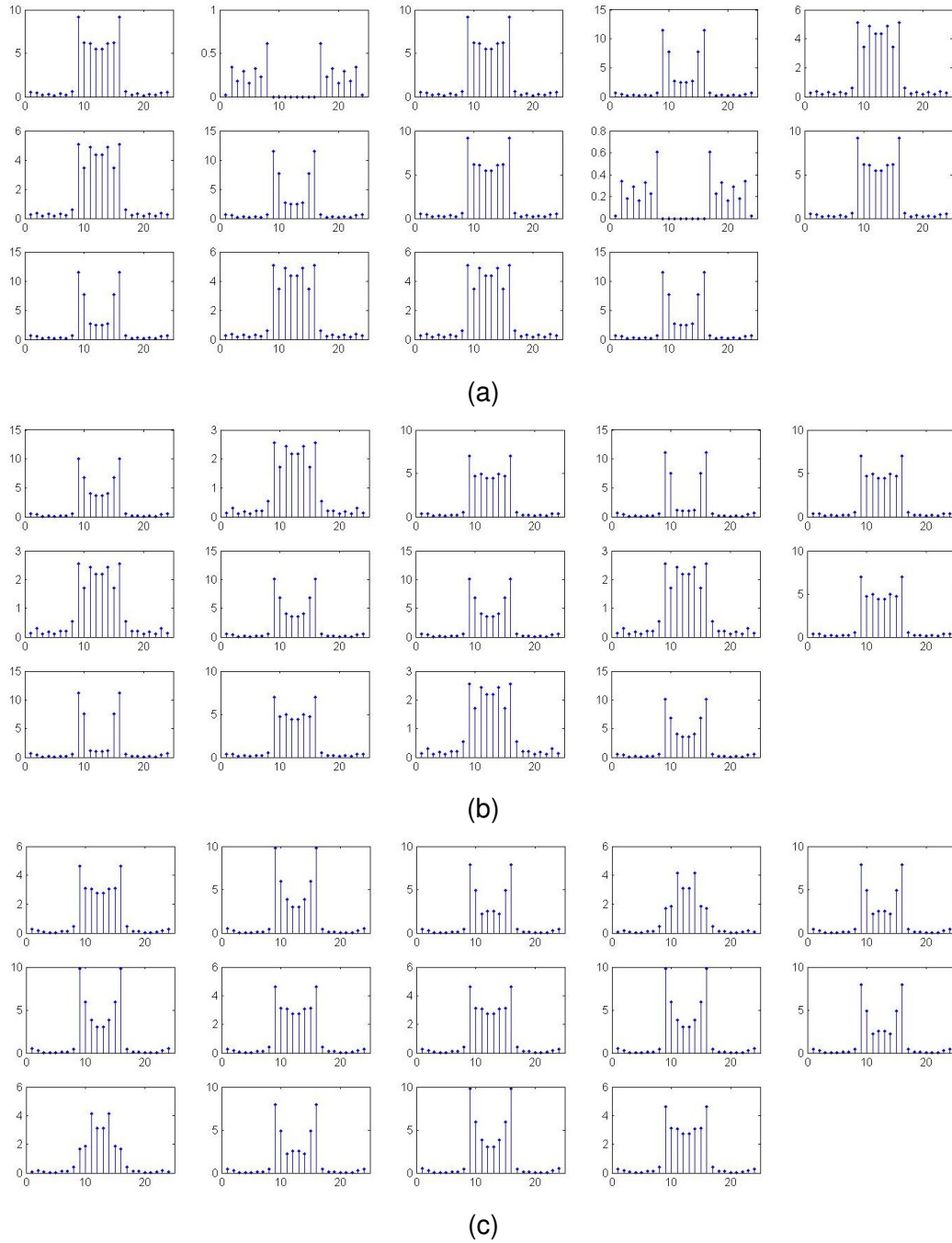


FIGURE 3.8 – Module des réponses impulsionnelles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est  $h_{a_2}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse dans le cas symétrique.

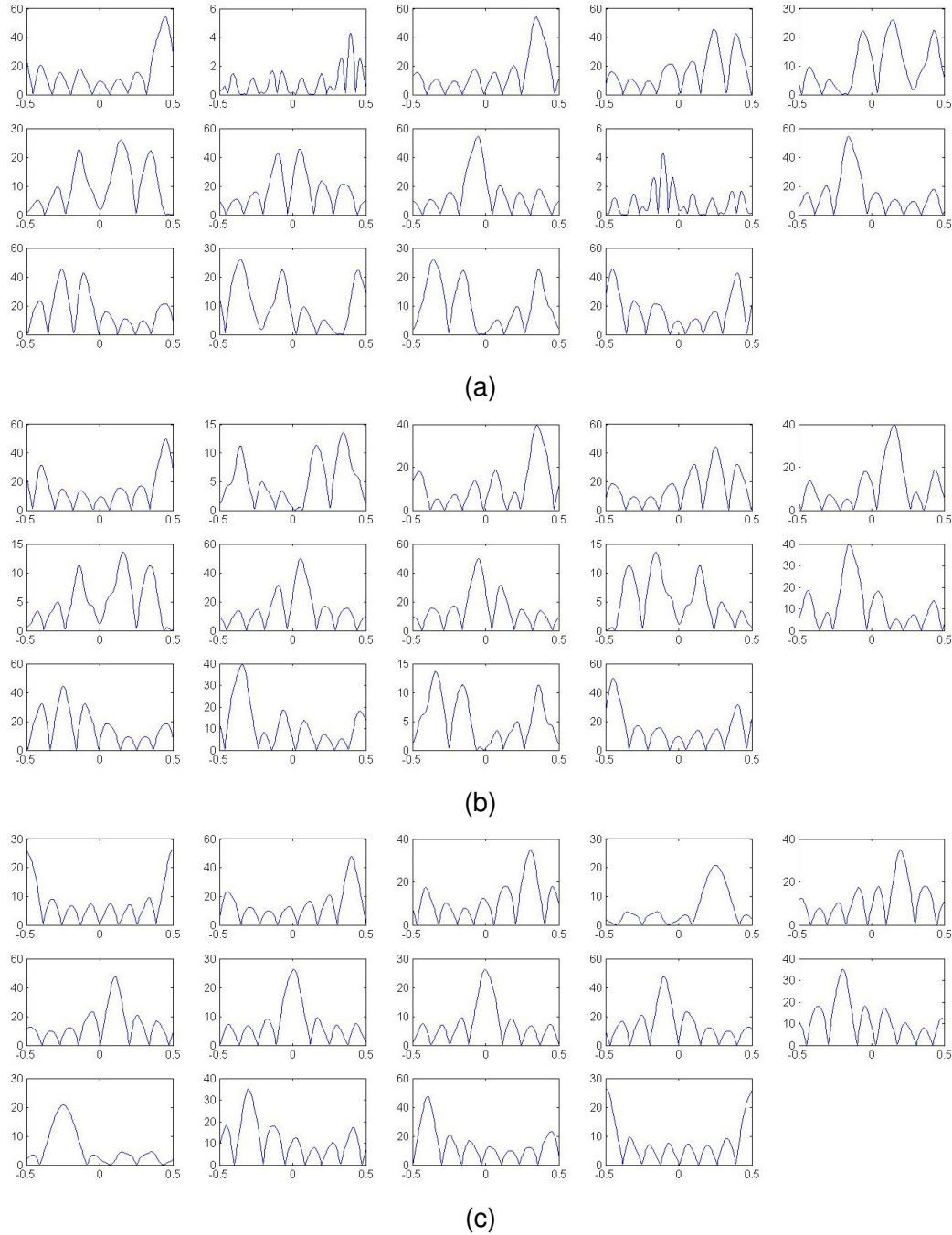


FIGURE 3.9 – Réponses fréquentielles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est  $h_{a2}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse symétrique.



---

## Chapitre 4

# Optimisation de bancs de filtres suréchantillonnés

---

Dans ce chapitre, nous allons exploiter la redondance existant dans un banc de filtres d'analyse suréchantillonné pour choisir parmi l'ensemble des bancs de filtres de synthèse permettant une reconstruction parfaite celui présentant de bonnes caractéristiques, par exemple une bonne sélectivité fréquentielle ou une bonne localisation temps-fréquence. Pour ce faire, nous allons, dans un premier temps, réexprimer le système définissant les filtres de synthèse. Cette réécriture va, de plus, permettre de transformer le problème d'optimisation sous contrainte en un problème de minimisation non contraint. Ensuite nous allons expliciter la méthode d'optimisation par descente de gradient utilisée, ainsi que les différentes fonctions de coût envisagées et leur gradient. Enfin, des exemples d'optimisation seront présentés mettant en relief le gain en temps de calcul de la réécriture du problème.

### 4.1 Réduction de la dimension du problème

#### 4.1.1 Cas général

Avant de s'intéresser à l'optimisation de bancs de filtres de synthèse en elle-même, voyons dans un premier temps comment réécrire de façon plus concise le système linéaire décrivant les bancs de filtres possibles. Dans cette partie, on se place à nouveau dans le cas où le banc de filtres d'analyse est inversible. Les notations utilisées sont les mêmes qu'à la section 3.3.

Soit  $r$  le rang de la matrice  $\mathcal{H} \in \mathbb{C}^{(k+p_1+p_2)N \times pM}$  décrivant le banc de filtres d'analyse ; on suppose ici que  $r < Mp$  (où  $p = p_1 + p_2 + 1$ ). Considérons la décomposition en valeurs singulières [Strang, 1998] de cette matrice :

$$\mathcal{H} = \mathcal{U}_0 \Sigma_0 \mathcal{V}_0^*,$$

où  $\Sigma_0 \in \mathbb{C}^{r \times r}$  est une matrice diagonale inversible,  $\mathcal{U}_0 \in \mathbb{C}^{N(k+p-1) \times r}$  et  $\mathcal{V}_0 \in \mathbb{C}^{Mp \times r}$  sont

des matrices semi-unitaires (i.e.  $\mathcal{U}_0^* \mathcal{U}_0 = \mathbf{I}_r$  et  $\mathcal{V}_0^* \mathcal{V}_0 = \mathbf{I}_r$ ). Ceci signifie qu'il existe des matrices :  $\mathcal{U}_1 \in \mathbb{C}^{N(k+p-1) \times (N(k+p-1)-r)}$  et  $\mathcal{V}_1 \in \mathbb{C}^{Mp \times (Mp-r)}$  telles que  $[\mathcal{U}_0 \ \mathcal{U}_1]$  et  $[\mathcal{V}_0 \ \mathcal{V}_1]$  soient des matrices unitaires.

Le banc de filtres d'analyse étant supposé inversible, on peut alors écrire une solution particulière du système (3.8) comme :

$$\tilde{\mathcal{H}}^0 = \mathcal{H}^\sharp \mathcal{U},$$

où  $\mathcal{H}^\sharp = \mathcal{V}_0 \Sigma_0^{-1} \mathcal{U}_0^*$  est la matrice pseudo-inverse de  $\mathcal{H}$ . L'équation (3.8) est alors équivalente à :

$$\mathcal{U}_0 \Sigma_0 \mathcal{V}_0^* (\tilde{\mathcal{H}} - \tilde{\mathcal{H}}^0) = \mathbf{0}_{N(k+p-1) \times N}.$$

Comme  $\mathcal{U}_0^* \mathcal{U}_0 = \mathbf{I}_r$  et  $\Sigma_0$  est inversible, on obtient :

$$\mathcal{V}_0^* (\tilde{\mathcal{H}} - \tilde{\mathcal{H}}^0) = \mathbf{0}_{r \times N},$$

ce qui est équivalent à dire que les colonnes de  $\tilde{\mathcal{H}} - \tilde{\mathcal{H}}^0$  appartiennent à  $\text{Ker}(\mathcal{V}_0^*)$ , le noyau de l'application linéaire décrite par la matrice  $\mathcal{V}_0^*$ . On peut vérifier, de plus, que  $\text{Ker}(\mathcal{V}_0^*)$  est égal à  $\text{Im}(\mathcal{V}_1)$ . On peut donc écrire :

$$\tilde{\mathcal{H}} = \mathcal{V}_1 \mathcal{C} + \tilde{\mathcal{H}}^0, \quad (4.1)$$

où  $\mathcal{C} \in \mathbb{C}^{(Mp-r) \times N}$ .

La construction du banc de filtres de synthèse est alors réduite au choix de  $\mathcal{C}$ . On peut remarquer qu'en considérant une matrice identiquement nulle :  $\mathcal{C} = \mathbf{0}_{(Mp-r) \times N}$ , on se ramène au résultat obtenu en utilisant la pseudo-inverse.

Nous allons maintenant réécrire cette expression sous une forme équivalente qui sera plus aisée à exploiter par la suite. On définit les matrices  $(\mathbf{V}_j)_{j \in \{0, \dots, M-1\}}$  par : pour tout  $\ell \in \{-p_1, \dots, p_2\}$  et  $n \in \{0, \dots, Mp-r-1\}$ ,

$$(\mathbf{V}_j)_{\ell+p_1, n} = \mathcal{V}_1((\ell+p_1)M + j, n),$$

avec  $\mathcal{V}_1 = [\mathcal{V}_1(s, n)]_{0 \leq s < Mp, 0 \leq n < Mp-r}$ . D'après les équations (4.1) et (3.9), on a pour tout  $\ell \in \{-p_1, \dots, p_2\}$ ,  $i \in \{0, \dots, N-1\}$  et  $j \in \{0, \dots, M-1\}$  :

$$\tilde{H}_{i,j}(\ell) = \sum_{n=0}^{Mp-r-1} (\mathbf{V}_j)_{\ell+p_1, n} \mathcal{C}(n, i) + \tilde{H}_{i,j}^0(\ell),$$

où  $(\tilde{H}_{i,j}(\ell))_{-p_1 \leq \ell \leq p_2}$  est une réponse impulsionnelle du système polyphase équivalent au banc de filtres de synthèse,  $(\tilde{H}_{i,j}^0(\ell))_{-p_1 \leq \ell \leq p_2}$  correspond au système polyphase pseudo-inverse et  $\mathcal{C} = [\mathcal{C}(n, i)]_{0 \leq n < Mp-r, 0 \leq i < N}$ .

Introduisons maintenant les matrices  $\mathbf{H}_j^0$ , pour tout  $j \in \{0, \dots, M-1\}$ , définies de la manière suivante :  $(\mathbf{H}_j^0)_{\ell+p_1, i} = \tilde{H}_{i,j}^0(\ell)$  pour  $\ell \in \{-p_1, \dots, p_2\}$  et  $i \in \{0, \dots, N-1\}$ . On peut finalement écrire :

$$\tilde{H}_{i,j}(\ell) = (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)_{\ell+p_1, i}. \quad (4.2)$$

Cette équation nous sera utile pour simplifier l'écriture du problème d'optimisation des bancs de filtres de synthèse.

**Remarque 4.1** Dans ce chapitre, nous nous plaçons dans le cas complexe pour réaliser les calculs. On peut noter cependant que ces calculs restent valables dans le cas réel, ce qui sera illustré par le premier exemple de la section 4.5.3-a) sur un banc de filtres réel.

### 4.1.2 Cas symétrique

Dans le cas où le banc de filtres de synthèse est construit pour être utilisé sur des coefficients présentant une symétrie hermitienne, on peut également réécrire le problème en utilisant la même méthode que dans la section précédente. On utilise ici les mêmes notations qu'à la section 3.4.2.

Le point important est de remarquer que l'on peut calculer la matrice  $\tilde{\mathcal{H}}$  directement à partir de la matrice  $\tilde{\mathcal{H}}_s$ , qui a été définie à la section 3.4.2 par l'équation (3.12) dans le cas où  $M$  est pair et par (3.13) dans le cas  $M$  impair, de la manière suivante :

$$\tilde{\mathcal{H}} = \mathbf{P}_{\text{rc}} \tilde{\mathcal{H}}_s \quad (4.3)$$

où la matrice  $\mathbf{P}_{\text{rc}}$  vaut :

$$\mathbf{P}_{\text{rc}} = \begin{pmatrix} \mathbf{I}_{M'} & -\mathbf{J}_{M'} & 0 & \cdots & 0 \\ \mathbf{J}_{M'} & \mathbf{I}_{M'} & 0 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & \mathbf{I}_{M'} & -\mathbf{J}_{M'} \\ 0 & \cdots & 0 & \mathbf{J}_{M'} & \mathbf{I}_{M'} \end{pmatrix} \in \mathbb{C}^{pM \times pM},$$

si  $M = 2M'$  (cas pair, traité à la section 3.4.2-a)) et :

$$\mathbf{P}_{\text{rc}} = \begin{pmatrix} \mathbf{I}_{M'} & 0 & -\mathbf{J}_{M'} & 0 & \cdots & 0 \\ 0 & \sqrt{2} & 0 & & & 0 \\ \mathbf{J}_{M'} & 0 & \mathbf{I}_{M'} & 0 & \cdots & 0 \\ & & & \ddots & & \\ 0 & \cdots & 0 & \mathbf{I}_{M'} & 0 & -\mathbf{J}_{M'} \\ 0 & & & 0 & \sqrt{2} & 0 \\ 0 & \cdots & 0 & \mathbf{J}_{M'} & 0 & \mathbf{I}_{M'} \end{pmatrix} \in \mathbb{C}^{pM \times pM},$$

si  $M = 2M' + 1$  (cas impair, traité à la section 3.4.2-b)).

En appliquant à nouveau une décomposition en valeurs singulières sur  $\mathcal{H}_s$ , puis en suivant les mêmes étapes que dans la section 4.1.1, on aboutit à une équation similaire à (4.1) :

$$\tilde{\mathcal{H}}_s = \mathcal{V}_1 \mathcal{C} + \tilde{\mathcal{H}}_s^0. \quad (4.4)$$

Il est intéressant de noter que, de par les propriétés de la décomposition en valeurs singulières d'une matrice réelle, la matrice  $\mathcal{C}$  présente la même dimension que dans le cas général mais que, maintenant, elle est réelle.

En utilisant les équations (4.3) et (4.4), on déduit :

$$\tilde{\mathcal{H}} = \mathbf{P}_{\text{rc}} \mathcal{V}_1 \mathcal{C} + \mathbf{P}_{\text{rc}} \tilde{\mathcal{H}}_s^0.$$

Comme  $\tilde{\mathcal{H}}^0 = \mathbf{P}_{\text{rc}} \tilde{\mathcal{H}}_s^0$ , en posant  $\mathcal{W}_1 = \mathbf{P}_{\text{rc}} \mathcal{V}_1$ , on obtient finalement :

$$\tilde{\mathcal{H}} = \mathcal{W}_1 \mathcal{C} + \tilde{\mathcal{H}}^0. \quad (4.5)$$

On définit les matrices suivantes, pour tout  $\ell \in \{-p_1, \dots, p_2\}$ ,  $j \in \{0, \dots, M-1\}$  et  $n \in \{0, \dots, Mp-r-1\}$  :

$$(\mathbf{W}_j)_{\ell+p_1, n} = \mathcal{W}_1((\ell+p_1)M+j, n).$$

En utilisant l'équation (4.5) de la même manière que ce qui a été fait à la section 4.1.1, on obtient :

$$\tilde{H}_{i,j}(\ell) = (\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0)_{\ell+p_1, i}. \quad (4.6)$$

## 4.2 Solution optimale

### 4.2.1 Forme générale de la fonction de coût considérée

Suivant les caractéristiques que l'on souhaite obtenir pour le banc de filtres de synthèse, plusieurs fonctions de coût peuvent être employées. Nous allons d'abord proposer une forme assez générale de fonction de coût, puis nous donnerons des exemples de fonctions se ramenant à cette forme générale.

On souhaite optimiser les coefficients  $\tilde{h}$  des filtres de synthèse. D'après les résultats de la section 4.1 il est clair que ces coefficients peuvent être représentés dans le cas général en utilisant l'équation (4.2). L'optimisation peut donc être réalisée dans l'espace de dimension réduite (par rapport à celle de l'ensemble des coefficients de  $\tilde{h}$ ) dans lequel varie la matrice  $\mathcal{C}$ . La fonction de coût que nous considérons est alors de la forme :

$$J(\tilde{h}) = \tilde{J}(\mathcal{C}) = \sum_{j=0}^{M-1} \frac{\|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j}^2}{\|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2}.$$

La notation suivante a été ici utilisée :

$$\forall \mathbf{A} \in \mathbb{C}^{N \times p}, \quad \|\mathbf{A}\|_K^2 = \sum_{(i, i', \ell, \ell')} \mathbf{A}_{i, \ell} \overline{\mathbf{A}_{i', \ell'}} K(i, i', \ell, \ell'),$$

où  $K$  est un noyau de taille  $(N \times N \times p \times p)$ . Cette notation est aussi utilisée pour les termes au dénominateur de  $J(\tilde{h})$  faisant intervenir un noyau  $\Lambda$ .

Par ailleurs, on suppose ici que  $\|\mathbf{A}\|_K$  représente une semi-norme sur  $\mathbb{C}^{N \times p}$  et est donc réelle positive ou nulle. Soit la matrice  $\mathbf{K}'$ , définie par :  $\mathbf{K}'_{i+\ell N, i'+\ell' N} = K(i, i', \ell, \ell')$  pour tout  $(\ell, \ell') \in \{0, \dots, p-1\}^2$  et  $(i, i') \in \{0, \dots, N-1\}^2$ . Sans perte de généralité, cette matrice peut être prise hermitienne semi-définie positive, ce qui signifie en particulier :

$$\mathbf{K}'_{i+\ell N, i'+\ell' N} = \overline{\mathbf{K}'_{i'+\ell' N, i+\ell N}} \text{ et donc : } K(i, i', \ell, \ell') = \overline{K(i', i, \ell', \ell)}.$$

On a alors l'équation suivante :

$$\|\mathbf{A}\|_K^2 = \sum_{(i, i', \ell, \ell')} \mathbf{A}_{i, \ell} \overline{\mathbf{A}_{i', \ell'}} K(i, i', \ell, \ell') = \sum_{(i, i', \ell, \ell')} \mathbf{A}_{i, \ell} \overline{\mathbf{A}_{i', \ell'} K(i', i, \ell', \ell)}.$$

Cette relation sera utile pour simplifier certaines expressions dans les sections 4.3.1 et 4.4.3. Notons que  $\mathbf{K}'$  est hermitienne définie positive si et seulement si  $\|\cdot\|_K$  est une norme.



### 4.2.2 Critère de bonne localisation temporelle

Dans un premier temps nous allons considérer une fonction de coût sur les réponses impulsionnelles des filtres de synthèse  $(\tilde{h}_j)_{0 \leq j < M}$ . Le but est ici d'essayer de concentrer ces réponses autour d'indices temporels  $(\bar{m}_j)_{0 \leq j < M}$ . Nous allons voir comment appliquer la réduction de dimension pour simplifier le problème.

La fonction de coût à laquelle on s'intéresse est la suivante :

$$J_t(\tilde{h}) = \sum_{j=0}^{M-1} \frac{\sum_m |m - \bar{m}_j|^\alpha |\tilde{h}_j(m)|^2}{\sum_m |\tilde{h}_j(m)|^2},$$

avec  $\alpha \in \mathbb{R}^*$ .

Remarquons que, si  $\alpha = 2$  et  $\bar{m}_j = \frac{\sum_m m |\tilde{h}_j(m)|^2}{\sum_m |\tilde{h}_j(m)|^2}$ , alors  $J_t(\tilde{h})$  représente la dispersion temporelle classiquement utilisée pour mesurer le pouvoir de localisation temporelle du filtre  $\tilde{h}_j$  [Flandrin, 1998]. Compte tenu de (3.6), on a :

$$\begin{aligned} J_t(\tilde{h}) &= \sum_{j=0}^{M-1} \frac{\sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} |\ell N - i - \bar{m}_j|^\alpha |\tilde{h}_j(\ell N - i)|^2}{\sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} |\tilde{h}_j(\ell N - i)|^2} \\ &= \sum_{j=0}^{M-1} \frac{\sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} |\ell N - i - \bar{m}_j|^\alpha |\tilde{H}_{i,j}(\ell)|^2}{\sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} |\tilde{H}_{i,j}(\ell)|^2}. \end{aligned}$$

On introduit les noyaux  $K_j^t$  et  $\Lambda$  définis par :

$$\begin{aligned} K_j^t(i, i', \ell + p_1, \ell' + p_1) &= |\ell N - i - \bar{m}_j|^\alpha \delta_{i-i'} \delta_{\ell-\ell'}, \\ \Lambda(i, i', \ell + p_1, \ell' + p_1) &= \delta_{i-i'} \delta_{\ell-\ell'}, \end{aligned} \quad (4.7)$$

pour tout  $0 \leq j < M$ ,  $-p_1 \leq \ell, \ell' \leq p_2$  et  $0 \leq i, i' \leq N-1$ .

Les paramètres des noyaux  $K_j^t$  seront précisés à la section 4.5.1.

**Remarque 4.2** On peut remarquer que la forme particulière du noyau  $\Lambda$ , fait que  $\|\cdot\|_\Lambda$  est simplement la norme de Frobenius.

En utilisant l'équation (4.2) on peut écrire :

$$\sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} |\tilde{H}_{i,j}(\ell)|^2 = \|\mathbf{V}_j \mathbf{C} + \mathbf{H}_j^0\|_\Lambda^2$$

et

$$\sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} (\ell N - i - \bar{m}_j)^2 |\tilde{H}_{i,j}(\ell)|^2 = \|\mathbf{V}_j \mathbf{C} + \mathbf{H}_j^0\|_{K_j^t}^2.$$

Finalement, on en déduit que :

$$J_t(\tilde{h}) = \sum_{j=0}^{M-1} \frac{\|\mathbf{V}_j \mathbf{C} + \mathbf{H}_j^0\|_{K_j^t}^2}{\|\mathbf{V}_j \mathbf{C} + \mathbf{H}_j^0\|_\Lambda^2} = \tilde{J}_t(\mathbf{C}).$$

Le problème de minimisation sous contrainte de  $J_t$  a donc été réexprimé sous la forme d'une minimisation non contrainte de la fonction  $\tilde{J}_t$ .

### 4.2.3 Critère de bonne localisation fréquentielle

Nous proposons de refaire l'étude précédente, mais avec une fonction de coût différente :  $J_f(\tilde{h})$ , dont le but est de concentrer les réponses fréquentielles du banc de filtres de synthèse autour de fréquences  $f_j$  :

$$J_f(\tilde{h}) = \sum_{j=0}^{M-1} \frac{\int_{-1/2+f_j}^{1/2+f_j} |\nu - f_j|^\alpha \left| \tilde{h}_j[\nu] \right|^2 d\nu}{\int_{-1/2+f_j}^{1/2+f_j} \left| \tilde{h}_j[\nu] \right|^2 d\nu} \quad (4.8)$$

où  $\alpha \in \mathbb{R}^*$  et  $\tilde{h}_j[\cdot]$  est la réponse fréquentielle du  $j^{\text{ème}}$  filtre de synthèse, définie par :

$$\forall \nu \in [-1/2, 1/2[, \quad \tilde{h}_j[\nu] = \sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} \tilde{H}_{i,j}(\ell) e^{-2i\pi(N\ell-i)\nu}.$$

Si l'on considère le cas  $\alpha = 2$  et on prend  $f_j = \frac{\int_{-1/2}^{1/2} \nu \left| \tilde{h}_j[\nu] \right|^2 d\nu}{\int_{-1/2}^{1/2} \left| \tilde{h}_j[\nu] \right|^2 d\nu}$ , le critère  $J_f(\tilde{h})$  représente une mesure de la dispersion fréquentielle des filtres de synthèse. On définit le noyau :

$$\begin{aligned} K_j^f(i, i', \ell + p_1, \ell' + p_1) &= \int_{-1/2+f_j}^{1/2+f_j} |\nu - f_j|^\alpha e^{-2i\pi(N(\ell-\ell')-(i-i'))\nu} d\nu \\ &= \int_{-1/2}^{1/2} |\nu|^\alpha e^{-2i\pi(N(\ell-\ell')-(i-i'))(\nu+f_j)} d\nu, \end{aligned} \quad (4.9)$$

avec  $(i, i', \ell, \ell') \in \{0, \dots, N-1\}^2 \times \{-p_1, \dots, p_2\}^2$ .

**Exemple 5** Dans la pratique, nous avons travaillé avec un exposant  $\alpha = 2$ . Donnons une expression explicite du noyau dans ce cas. Traitons tout d'abord le cas où  $N(\ell - \ell') - (i - i') = 0$ , autrement dit le cas  $i = i'$  et  $\ell = \ell'$ , on a simplement :

$$K_j^f(i, i, \ell, \ell) = \left[ \frac{\nu^3}{3} \right]_{-1/2}^{1/2} = \frac{1}{12}.$$

Si  $i \neq i'$  et  $\ell \neq \ell'$ , on obtient avec deux intégrations par parties successives :

$$\begin{aligned} K_j^f(i, i', \ell + p_1, \ell' + p_1) &= e^{-2i\pi(N(\ell-\ell')-(i-i'))f_j} \left( \left[ \frac{\nu^2 e^{-2i\pi(N(\ell-\ell')-(i-i'))\nu}}{-2i\pi(N(\ell-\ell')-(i-i'))} \right]_{-1/2}^{1/2} \right. \\ &\quad \left. - \left[ \frac{2\nu e^{-2i\pi(N(\ell-\ell')-(i-i'))\nu}}{(-2i\pi(N(\ell-\ell')-(i-i')))^2} \right]_{-1/2}^{1/2} + \left[ \frac{2e^{-2i\pi(N(\ell-\ell')-(i-i'))\nu}}{(-2i\pi(N(\ell-\ell')-(i-i')))^3} \right]_{-1/2}^{1/2} \right). \end{aligned}$$

$N, \ell, \ell', i$  et  $i'$  étant entiers, on a la relation élémentaire suivante :

$$e^{-i\pi(N(\ell-\ell')-(i-i'))} = (-1)^{N(\ell-\ell')-(i-i')}.$$

À l'aide de cette relation, le calcul conduit finalement au noyau :

$$K_j^f(i, i', \ell + p_1, \ell' + p_1) = \begin{cases} \frac{1}{12} & \text{si } i = i' \text{ et } \ell = \ell' \\ \frac{(-1)^{N(\ell-\ell')-(i-i')} e^{-2i\pi(N(\ell-\ell')-(i-i'))f_j}}{2\pi^2(N(\ell-\ell')-(i-i'))^2} & \text{sinon} \end{cases}$$

avec  $(i, i', \ell, \ell') \in \{0, \dots, N-1\}^2 \times \{-p_1, \dots, p_2\}^2$ .

En utilisant ces notations et l'équation (4.2), on peut écrire :

$$\int_{-1/2+f_j}^{1/2+f_j} |\nu - f_j|^\alpha \left| \tilde{h}_j[\nu] \right|^2 d\nu = \|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j^f}^2.$$

D'après l'identité de Parseval et en utilisant le même noyau  $\Lambda$  que celui défini à la section 4.2.2, on obtient par ailleurs :

$$\int_{-1/2+f_j}^{1/2+f_j} \left| \tilde{h}_j[\nu] \right|^2 d\nu = \sum_{\ell=-p_1}^{p_2} \sum_{i=0}^{N-1} \left| \tilde{H}_{i,j}(\ell) \right|^2 = \|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2.$$

Finalement, en remplaçant dans l'équation (4.8), on déduit que :

$$J_f(\tilde{h}) = \sum_{j=0}^{M-1} \frac{\|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j^f}^2}{\|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2} = \tilde{J}_f(\mathcal{C}).$$

Une nouvelle fois le problème d'optimisation sous contrainte a pu être réécrit comme un problème de minimisation non contraint.

## 4.3 Optimisation par descente de gradient

Après avoir vu comment le problème d'optimisation contraint peut être transformé en un problème non contraint, nous allons fournir plus de détails concernant l'algorithme utilisé pour la minimisation.

### 4.3.1 Algorithme du gradient

Pour réaliser l'optimisation, nous avons choisi d'utiliser une simple descente de gradient avec un pas adapté. Le but ici est de garder le pas  $\mu$  aussi grand que possible quand on est loin d'une solution et de ne le réduire que quand on se rapproche d'un minimum. L'algorithme est alors le suivant :

- ① Initialisation :  $\mathcal{C}_0 = \mathbf{0}$ .

- ②  $\mu = 1$
- ③ Calcul de :  $\mathbf{D}_n = \nabla \tilde{J}(\mathcal{C}_n)$ .
- ④ Tant que  $\tilde{J}(\mathcal{C}_n - \mu \mathbf{D}_n) \geq \tilde{J}(\mathcal{C}_n)$ , calculer :  $\mu \leftarrow \frac{1}{\frac{1}{\mu} + 1}$ .
- ⑤  $\mathcal{C}_{n+1} = \mathcal{C}_n - \mu \mathbf{D}_n$ .
- ⑥ Si  $\|\mathcal{C}_{n+1} - \mathcal{C}_n\| > \epsilon$  alors retour à l'étape ②.

Le pas  $\mu$  utilisé ici reste grand tant que l'algorithme se rapproche d'un minimum local (autrement dit que  $\tilde{J}(\mathcal{C}_{n+1}) < \tilde{J}(\mathcal{C}_n)$ ) et diminue uniquement en cas de problème jusqu'à assurer à nouveau la décroissance du critère. Cette recherche de pas se faisant à chaque itération, cette méthode assure que le pas choisi à chaque étape sera bien adapté. L'initialisation par  $\mathcal{C}_0 = \mathbf{0}$ , signifie que l'on considère comme point de départ pour l'algorithme le banc de filtres de synthèse obtenu directement par pseudo-inverse.

D'autres stratégies pour le choix du pas existent : pas constant, pas optimal, pas satisfaisant les conditions de Wolfe ou d'Armijo [Nocedal, Wright, 1999; Barrault, Le Bris, 1999]. La méthode employée ici présente l'avantage d'être aisée à mettre en œuvre et de s'adapter aux différentes fonctions de coût que nous avons envisagées, tout en conservant une complexité raisonnable.

**Remarque 4.3** Une méthode classique consiste à chercher le pas optimal à chaque itération. Pour ce faire, il faut calculer  $Q(\mu) = \tilde{J}(\mathcal{C}_n - \mu \nabla \tilde{J}(\mathcal{C}_n))$ , puis chercher les zéros de  $Q'(\mu)$  et enfin regarder parmi ces valeurs laquelle minimise  $\tilde{J}(\mathcal{C}_n - \mu \nabla \tilde{J}(\mathcal{C}_n))$ . Regardons dans le cas qui nous intéresse, ce que valent  $Q$  et  $Q'$ . On a :

$$Q(\mu) = \sum_{j=0}^{M-1} \frac{\left\| \mathbf{V}_j(\mathcal{C}_n - \mu \nabla \tilde{J}(\mathcal{C}_n)) + \mathbf{H}_j^0 \right\|_{K_j}^2}{\left\| \mathbf{V}_j(\mathcal{C}_n - \mu \nabla \tilde{J}(\mathcal{C}_n)) + \mathbf{H}_j^0 \right\|_{\Lambda}^2}.$$

Notons :  $\mathbf{A}_j = \mathbf{V}_j \mathcal{C}_n + \mathbf{H}_j^0$  et  $\mathbf{B} = \nabla \tilde{J}(\mathcal{C}_n)$ . On a alors, pour un noyau  $K$  ( $K_j$  ou  $\Lambda$ ) :

$$\|\mathbf{A}_j - \mu \mathbf{B}\|_K^2 = \|\mathbf{A}_j\|_K^2 - \mu \sum_{p,p',q,q'} ((A_j)_{p,q} \overline{B_{p',q'}} + B_{p,q} \overline{(A_j)_{p',q'}}) K(p, p', q, q') + \mu^2 \|\mathbf{B}\|_K^2$$

En utilisant la notation :  $A_j \otimes_K B = \sum_{p,p',q,q'} (A_j)_{p,q} \overline{B_{p',q'}} K(p, p', q, q')$ , et le fait que le noyau  $K$  vérifie :  $K(p, p', q, q') = \overline{K(p', p, q', q)}$ , on peut en déduire que :

$$\|\mathbf{A}_j - \mu \mathbf{B}\|_K^2 = \|\mathbf{A}_j\|_K^2 - 2\mu \operatorname{Re}(\mathbf{A}_j \otimes_K \mathbf{B}) + \mu^2 \|\mathbf{B}\|_K^2.$$

En dérivant  $Q$  par rapport à  $\mu$ , on obtient alors :

$$Q'(\mu) = 2 \sum_{j=0}^{M-1} \frac{(\|\mathbf{B}\|_{K_j}^2 \mu - \operatorname{Re}(\mathbf{A}_j \otimes_{K_j} \mathbf{B})) \|\mathbf{A}_j - \mu \mathbf{B}\|_{\Lambda}^2 - (\|\mathbf{B}\|_{\Lambda}^2 \mu - \operatorname{Re}(\mathbf{A}_j \otimes_{\Lambda} \mathbf{B})) \|\mathbf{A}_j - \mu \mathbf{B}\|_{K_j}^2}{\|\mathbf{A}_j - \mu \mathbf{B}\|_{\Lambda}^4}.$$

En développant le numérateur du  $j^{\text{ème}}$  terme de la somme dans  $Q'(\mu)$ , on obtient :

$$\begin{aligned} N_j = & \mu^2 (\operatorname{Re}(\mathbf{A}_j \otimes_{K_j} \mathbf{B}) \|\mathbf{B}\|_{\Lambda}^2 - \operatorname{Re}(\mathbf{A}_j \otimes_{\Lambda} \mathbf{B}) \|\mathbf{B}\|_{K_j}^2) + \mu (\|\mathbf{B}\|_{K_j}^2 \|\mathbf{A}_j\|_{\Lambda}^2 - \|\mathbf{A}_j\|_{K_j}^2 \|\mathbf{B}\|_{\Lambda}^2) \\ & + \|\mathbf{A}_j\|_{K_j}^2 \operatorname{Re}(\mathbf{A}_j \otimes_{\Lambda} \mathbf{B}) - \|\mathbf{A}_j\|_{\Lambda}^2 \operatorname{Re}(\mathbf{A}_j \otimes_{K_j} \mathbf{B}). \end{aligned}$$

En mettant tous les termes au même dénominateur on obtient alors que le terme de plus haut degré au numérateur (de degré  $2 + 4(M - 1)$ ) est :

$$\sum_j (Re(\mathbf{A}_j \otimes_{K_j} \mathbf{B}) \|\mathbf{B}\|_{\Lambda}^2 - Re(\mathbf{A}_j \otimes_{\Lambda} \mathbf{B}) \|\mathbf{B}\|_{K_j}^2) \|\mathbf{B}\|^{4(M-1)},$$

qui n'a aucune raison d'être nul en général. Cela signifie que pour appliquer la méthode de gradient à pas optimal, il faut résoudre un polynôme de degré  $4M - 2$ . Avec  $N = 16$ ,  $k' = 7/4$ , cela veut dire trouver les zéros d'un polynôme de degré 110 à chaque itération, ce qui n'est pas raisonnable en pratique.

Pour pouvoir utiliser la méthode de descente de gradient exposée ici, il ne manque que l'expression des gradients des différentes fonctions de coût proposées. Nous allons donc les calculer maintenant.

### 4.3.2 Expression générale du gradient

Dans cette partie nous allons étudier le gradient par rapport à la matrice  $\mathcal{C}$  de  $\tilde{J}$ , la forme générale des fonctions de coût. Dans un premier temps nous devons calculer le gradient de  $f(\mathcal{C}) = \|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j}^2$ .

La matrice  $\mathcal{C}$  étant complexe, on a :

$$\begin{aligned} \frac{\partial f}{\partial \mathcal{C}_{m,n}} &= \frac{\partial f}{\partial \mathcal{C}_{m,n}^R} + \imath \frac{\partial f}{\partial \mathcal{C}_{m,n}^I} \\ &= \sum_{i,i',\ell,\ell'} \left( (\mathbf{V}_j)_{i,m} \delta_{\ell-n} (\overline{\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0})_{i',\ell'} + \overline{(\mathbf{V}_j)_{i',m}} \delta_{\ell'-n} (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)_{i,\ell} \right) K_j(i, i', \ell, \ell') \\ &\quad + \imath \left( \imath (\mathbf{V}_j)_{i,m} \delta_{\ell-n} (\overline{\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0})_{i',\ell'} - \imath \overline{(\mathbf{V}_j)_{i',m}} \delta_{\ell'-n} (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)_{i,\ell} \right) K_j(i, i', \ell, \ell') \\ &= 2 \sum_{i,i',\ell} \overline{(\mathbf{V}_j)_{i',m}} (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)_{i,\ell} K_j(i, i', \ell, n). \end{aligned}$$

À partir du résultat du calcul précédent, on déduit :

$$\left( \nabla \tilde{J}(\mathcal{C}) \right)_{m,n} = 2 \sum_{j=0}^{M-1} \frac{\sum_{i,i',\ell} \overline{(\mathbf{V}_j)_{i',m}} (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)_{i,\ell} (\beta_j K_j(i, i', \ell, n) - \alpha_j \Lambda(i, i', \ell, n))}{\beta_j^2}, \quad (4.10)$$

avec  $\alpha_j = \|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j}^2$  et  $\beta_j = \|\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2$ .

### 4.3.3 Exemples de gradients

Un premier exemple est le calcul du gradient de la fonction de coût  $\tilde{J}_t(\mathcal{C})$ . En appliquant le résultat (4.10) sur la fonction de coût  $\tilde{J}_t$ , avec les noyaux  $(K_j^t)_{0 \leq j < M}$ , on a :

$$\begin{aligned} \frac{\partial f}{\partial \mathcal{C}_{m,n}} &= 2 \sum_i \overline{(\mathbf{V}_j)_{i,m}} (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)_{i,n} (\Gamma_j)_{i,n} \\ &= 2 \left( \mathbf{V}_j^* (\Gamma_j \odot (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)) \right)_{m,n}, \end{aligned}$$

avec :  $(\mathbf{\Gamma}_j)_{i,\ell+p_1} = |\ell N - i - \overline{m}_j|^\alpha$  pour tout  $0 \leq i < N$  et  $-p_1 < \ell \leq p_2$ . La notation  $\odot$  désigne ici le produit d'Hadamard (produit matriciel point à point). Finalement on peut écrire :

$$\nabla \tilde{J}_t(\mathcal{C}) = 2 \sum_{j=0}^{M-1} \frac{\mathbf{V}_j^* ((\beta_j \mathbf{\Gamma}_j - \alpha_j \mathbf{1}) \odot (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0))}{\beta_j^2},$$

où  $\mathbf{1}_{i,\ell} = 1$  pour tout  $0 \leq i < N$  et  $0 \leq \ell < p$ .

Le même travail peut être réalisé sur la fonction de coût  $\tilde{J}_f$ , avec les noyaux  $(K_j^f)_{0 \leq j < M}$ . La réécriture du résultat sous forme matricielle n'apportant pas de simplification, le gradient s'écrit alors dans ce cas :

$$\left( \nabla \tilde{J}_f(\mathcal{C}) \right)_{m,n} = 2 \sum_{j=0}^{M-1} \frac{\sum_{i,i',\ell} (\overline{\mathbf{V}}_j)_{i',m} (\mathbf{V}_j \mathcal{C} + \mathbf{H}_j^0)_{i,\ell} (\beta_j K_j^f(i, i', \ell, n) - \alpha_j \delta_{i-i'} \delta_{\ell-n})}{\beta_j^2}.$$

## 4.4 Solution optimale dans le cas symétrique

### 4.4.1 Fonctions de coût dans le cas symétrique

On utilise ici les mêmes notations que quand la section 4.2. Par analogie, les formes des fonctions de coût dans le cas symétrique sont :

$$J_s(\tilde{h}) = \tilde{J}_s(\mathcal{C}) = \sum_{j=0}^{M-1} \frac{\|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j}^2}{\|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2}.$$

De la même manière que dans le cas général, l'équation (4.6) a été utilisée pour transformer le problème d'optimisation contraint sur  $\tilde{h}$  en un problème d'optimisation non contraint sur la matrice  $\mathcal{C}$ .

### 4.4.2 Exemples de fonctions de coût

L'équation (4.6) est très similaire à l'équation (4.2) et on peut procéder de la même manière que dans les sections 4.2.2 et 4.2.3 pour définir les fonctions de coût dans le cas symétrique. On considère donc les fonctions suivantes :

$$\tilde{J}_{ts}(\mathcal{C}) = \sum_{j=0}^{M-1} \frac{\|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j^t}^2}{\|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2},$$

pour concentrer temporellement les réponses impulsionnelles, et :

$$\tilde{J}_{fs}(\mathcal{C}) = \sum_{j=0}^{M-1} \frac{\|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j^f}^2}{\|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2},$$

pour améliorer la sélectivité fréquentielle.

### 4.4.3 Gradients des fonctions de coût

De la même manière que ce qui avait été fait dans la section 4.3.2, on peut calculer le gradient de  $f(\mathcal{C}) = \|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j}^2$  par rapport à la matrice  $\mathcal{C}$ , la seule différence étant que maintenant cette matrice est réelle :

$$\frac{\partial f}{\partial \mathcal{C}_{m,n}} = \sum_{i,i',\ell,\ell'} \left( (\mathbf{W}_j)_{i,m} \delta_{\ell-n} (\overline{\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0})_{i',\ell'} + (\overline{\mathbf{W}_j})_{i',m} \delta_{\ell'-n} (\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0)_{i,\ell} \right) K_j(i, i', \ell, \ell').$$

À partir de cette expression on peut en déduire la forme du gradient de la fonction de coût  $\tilde{J}_s$  :

$$\begin{aligned} (\nabla \tilde{J}_s(\mathcal{C}))_{m,n} &= \sum_{j=0}^{M-1} \left[ \frac{\sum_{i,i',\ell'} (\mathbf{W}_j)_{i,m} (\overline{\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0})_{i',\ell'} (\beta_j K_j(i, i', n, \ell') - \alpha_j \Lambda(i, i', n, \ell'))}{\beta_j^2} \right. \\ &\quad \left. + \frac{\sum_{i,i',\ell} (\overline{\mathbf{W}_j})_{i',m} (\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0)_{i,\ell} (\beta_j K_j(i, i', \ell, n) - \alpha_j \Lambda(i, i', \ell, n))}{\beta_j^2} \right] \\ &= 2\text{Re} \left( \sum_{j=0}^{M-1} \frac{\sum_{i,i',\ell'} (\mathbf{W}_j)_{i,m} (\overline{\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0})_{i',\ell'} (\beta_j K_j(i, i', n, \ell') - \alpha_j \Lambda(i, i', n, \ell'))}{\beta_j^2} \right), \end{aligned} \quad (4.11)$$

avec :  $\alpha_j = \|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{K_j}^2$  et  $\beta_j = \|\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0\|_{\Lambda}^2$ , pour tout  $0 \leq j < M$ . Dans ce calcul, on a utilisé la relation :  $K_j(i, i', \ell, \ell') = \overline{K_j(i', i, \ell', \ell)}$ .

En utilisant l'équation (4.11), le calcul du gradient de  $\tilde{J}_{ts}$  conduit au résultat suivant :

$$\nabla \tilde{J}_{ts}(\mathcal{C}) = 2\text{Re} \left( \sum_{j=0}^{M-1} \frac{\mathbf{W}_j^* ((\beta_j \mathbf{\Gamma}_j - \alpha_j \mathbf{1}) \odot (\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0))}{\beta_j^2} \right),$$

où  $\mathbf{\Gamma}_j$  est défini comme à la section 4.3.3.

Pour la seconde fonction de coût on obtient de même :

$$(\nabla \tilde{J}_{fs}(\mathcal{C}))_{m,n} = 2\text{Re} \left( \sum_{j=0}^{M-1} \frac{\sum_{i,i',\ell'} (\mathbf{W}_j)_{i,m} (\overline{\mathbf{W}_j \mathcal{C} + \mathbf{H}_j^0})_{i',\ell'} (\beta_j K_j^f(i, i', n, \ell') - \alpha_j \delta_{i-i'} \delta_{n-\ell'})}{\beta_j^2} \right).$$

De nouveau, si l'on cherche à la réécrire sous forme matricielle, cette expression ne se simplifie pas.

**Remarque 4.4** Au chapitre 3, nous avons proposé deux méthodes pour obtenir des bancs de filtres à reconstruction parfaite dans le cas symétrique. On voit qu'ici la formulation introduite à la section 3.4.2 nous permet de réaliser l'optimisation directement sur l'ensemble des bancs de filtres de synthèse possédant la propriété de reconstruction réelle dans le cas d'une symétrie hermitienne. Ainsi on est assuré que le résultat de l'optimisation est à la fois symétrique et minimise le critère choisi, ce qui n'aurait pas été le cas si l'on avait appliqué la méthode de la section 3.4.1 au résultat d'optimisation obtenu avec la méthode décrite en 4.2.

## 4.5 Simulations

### 4.5.1 Choix des paramètres des noyaux

Les noyaux introduits précédemment utilisent des paramètres que nous allons expliciter dans cette section.

#### 4.5.1-a) Noyau $K_j^t$

Rappelons que l'expression de  $K_j^t$  est donnée par l'équation (4.7). Les paramètres  $\bar{m}_j$  définissent les positions (temporelles) autour desquelles on veut que la réponse impulsionnelle du  $j^{\text{ème}}$  filtre se concentre. Notre but étant ici de concentrer ces réponses au milieu du support, et ainsi avoir une bonne atténuation aux bords, nous avons choisi le même paramètre pour tous les filtres. Le support des réponses impulsionnelles des filtres de synthèse étant  $\{-p_1N - N + 1, \dots, p_2N\}$ , on a fixé comme valeur de  $\bar{m}_j$  le milieu de ce support :

$$\forall 0 \leq j < M, \quad \bar{m}_j = \bar{m} = \frac{p_2N + 1 - p_1N - N}{2}.$$

L'exposant  $\alpha$  a été pris égal à 2.

#### 4.5.1-b) Noyau $K_j^f$

Rappelons que l'expression de  $K_j^f$  est donnée par l'équation (4.9). Les paramètres  $f_j$  représentent les fréquences réduites autour desquelles on veut concentrer les réponses fréquentielles des filtres de synthèse. Plus précisément, on choisit la fréquence  $f_j$  telle qu'elle soit "au milieu" de la bande passante du filtre d'analyse  $h_j$ . En pratique, on calcule chaque  $f_j$  en recherchant d'abord le maximum  $m_j$  du module de la réponse fréquentielle du filtre d'analyse  $h_j$ , puis en déterminant la bande passante à -3dB (calculée en recherchant les gains supérieurs à  $m_j/\sqrt{2}$ ) et enfin en prenant le milieu du support ainsi déterminé.

L'exposant  $\alpha$  sera également pris égal à 2.

### 4.5.2 Temps de calculs

Dans la section 4.1, nous avons vu comment réduire la dimension du problème pour l'optimisation et, ainsi, transformer le problème de minimisation sous contrainte en un problème non contraint. Pour évaluer l'intérêt de cette réécriture, nous avons réalisé des programmes sous Matlab mettant en œuvre les optimisations contrainte (avec la fonction `fmincon`) et non contrainte (fonction `fminunc`) ainsi que la méthode de descente de gradient proposée à la section 4.3.1. Ces programmes ont été testés avec des bancs de filtres d'analyse du type de ceux que nous avons introduits à la section 3.5.2 avec les paramètres suivant : recouvrement  $k = 3$ , redondance  $k' = 7/4$  et facteur de décimation  $N \in \{4, 8, 16\}$ . Enfin, nous avons utilisé pour ce test la fonction de coût  $J_t$  telle qu'elle a été définie à la section 4.2.2.



Le tableau 4.1 montre les temps de calculs des différentes méthodes en utilisant Matlab 7.2 sur un ordinateur muni d'un processeur Intel Core2 T7400 à 2.16GHz avec 2Go de RAM.

	$N = 4$	$N = 8$	$N = 16$
Optimisation contrainte (avec <code>fmincon</code> )	1.2s	116s	8758s
Optimisation non contrainte (avec <code>fminunc</code> )	0.04s	0.7s	8s
Optimisation non contrainte (descente de gradient)	0.06s	0.6s	7s

TABLE 4.1 – Temps de calcul nécessaire sous Matlab pour l'optimisation de différents bancs de filtres.

Un premier résultat intéressant a été de constater que les 3 méthodes ont convergé vers le même banc de filtres de synthèse. Les temps de calculs avec ces 3 méthodes sont en revanche extrêmement différents : la réduction de dimension apporte un gain énorme puisqu'on passe de plus de deux heures (pour  $N = 16$ ) pour l'optimisation contrainte à quelques secondes pour l'optimisation non contrainte.

On peut remarquer également que la méthode par descente de gradient conduit à un temps de calcul pour l'optimisation équivalent (et parfois même légèrement meilleur, quand par exemple  $N$  est grand) à celui obtenu avec la fonction Matlab `fminunc`. Ce résultat est particulièrement intéressant car il montre que la méthode d'optimisation peut être facilement mise en place, sans perte de performance, quand on ne dispose pas de Matlab, et ce, sans avoir à recoder la fonction `fminunc`<sup>1</sup>. En effet, il suffit dans ce cas de programmer les quelques lignes de code correspondant à l'algorithme de descente de gradient. On peut noter à ce propos qu'il existe déjà des codes disponibles dans plusieurs langages pour ce type d'algorithme [Press *et al.*, 2007].

### 4.5.3 Bancs de filtres optimisés

Dans cette section, nous allons présenter des résultats d'optimisation obtenus sur les bancs de filtres introduits à la section 3.5 et en utilisant les différentes fonctions de coût introduites dans ce chapitre.

#### 4.5.3-a) Résultats dans le cas général

Tout d'abord, nous avons utilisé la méthode d'optimisation sur le banc de filtres réel présenté à la section 3.5.1 avec les paramètres :  $N = 8$ ,  $k = 4$  et  $k' = 2$ . La méthode a été testée avec la fonction de coût  $\tilde{J}_t$  pour améliorer la localisation temps/fréquence des filtres de synthèse. Le résultat est montré à la figure 4.2, où les coefficients obtenus par la méthode par pseudo-inverse sont également rappelés. On peut voir que les coefficients sont bien mieux concentrés au milieu du support des filtres. La figure 4.1 montre les réponses fréquentielles avant et après optimisation. On constate que celles des filtres optimisés présentent une sélectivité équivalente à celles avant optimisation (il n'y a donc

1. Cette fonction assez complexe met en place une méthode de Newton intérieure-réflexive [Coleman, Li, 1994].

pas eu de perte significative en sélectivité fréquentielle à l'issue de l'optimisation).

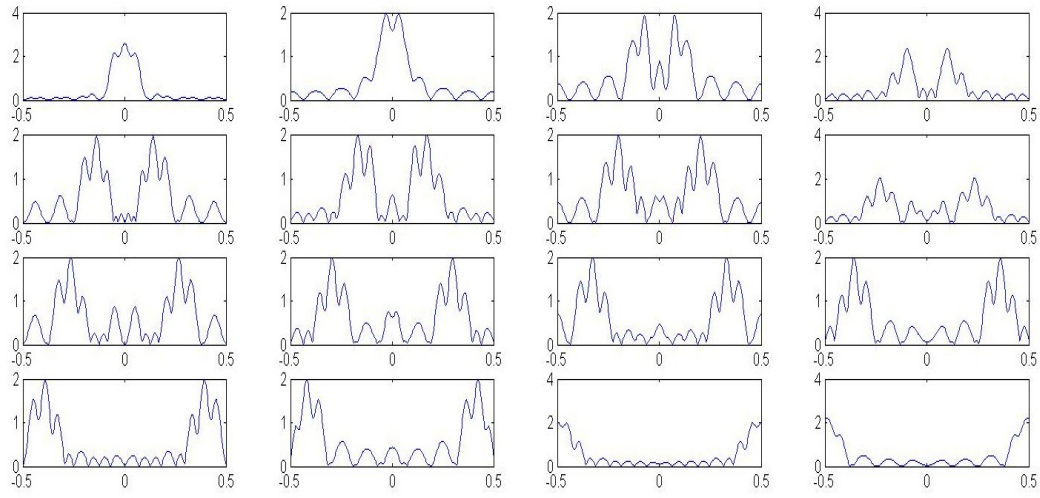
L'optimisation a également été testée sur le banc de filtres de la section 3.5.2 avec la fenêtre  $h_{a1}$ , en utilisant cette fois la fonction de coût  $\tilde{J}_f$ , afin d'améliorer la sélectivité fréquentielle des filtres de synthèse. Les paramètres utilisés ici sont :  $N = 8$ ,  $k = 3$  et  $k' = 7/4$ . On peut voir à la figure 4.3 le résultat obtenu, qui met en évidence que l'optimisation a permis de sélectionner des filtres présentant des réponses fréquentielles bien plus régulières et concentrées autour des fréquences choisies pour chaque filtre, d'où une bien meilleure sélectivité.

La fonction de coût employée pour cet exemple prend en compte tous les filtres de synthèse à la fois. Une question naturelle est alors de savoir si l'optimisation bénéficie à chaque filtre individuellement. Le tableau 4.2, présente la dispersion fréquentielle de chaque filtre avant et après optimisation avec la fonction de coût  $\tilde{J}_f$ . Dans ce cas, on constate que la dispersion fréquentielle de tous les filtres a été améliorée et que la variabilité des valeurs de dispersion a été réduite.

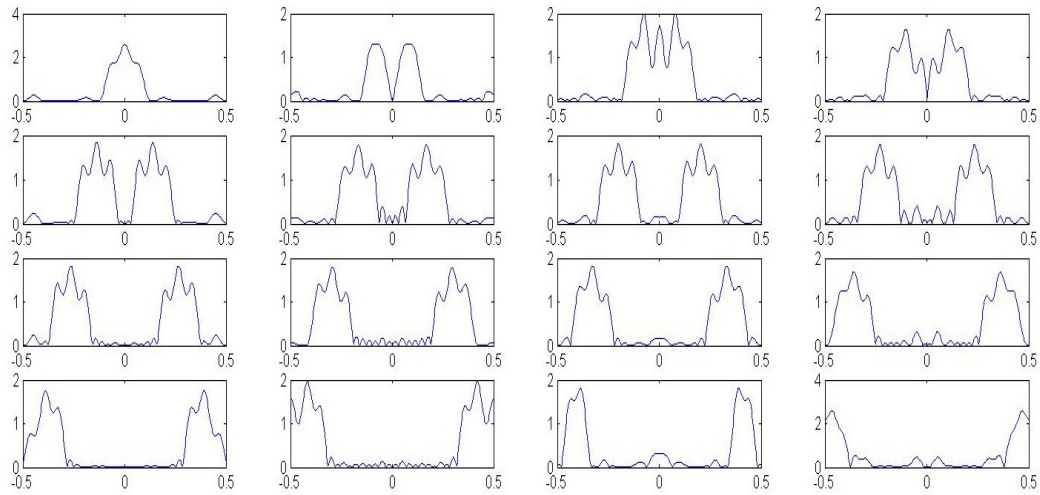
Filtre	Disp. freq.	Filtre	Disp. freq.
$\tilde{h}_0$	0.0290	$\tilde{h}_0^{opt}$	0.0111
$\tilde{h}_1$	0.0851	$\tilde{h}_1^{opt}$	0.0110
$\tilde{h}_2$	0.0569	$\tilde{h}_2^{opt}$	0.0109
$\tilde{h}_3$	0.0606	$\tilde{h}_3^{opt}$	0.0112
$\tilde{h}_4$	0.0658	$\tilde{h}_4^{opt}$	0.0110
$\tilde{h}_5$	0.0596	$\tilde{h}_5^{opt}$	0.0109
$\tilde{h}_6$	0.0363	$\tilde{h}_6^{opt}$	0.0111
$\tilde{h}_7$	0.0156	$\tilde{h}_7^{opt}$	0.0111
$\tilde{h}_8$	0.0411	$\tilde{h}_8^{opt}$	0.0109
$\tilde{h}_9$	0.0231	$\tilde{h}_9^{opt}$	0.0110
$\tilde{h}_{10}$	0.0508	$\tilde{h}_{10}^{opt}$	0.0112
$\tilde{h}_{11}$	0.0499	$\tilde{h}_{11}^{opt}$	0.0110
$\tilde{h}_{12}$	0.0574	$\tilde{h}_{12}^{opt}$	0.0109
$\tilde{h}_{13}$	0.0521	$\tilde{h}_{13}^{opt}$	0.0111
Somme	0.6833		0.1544

TABLE 4.2 – Dispersion fréquentielle des filtres de synthèse avant et après optimisation avec la fonction de coût  $\tilde{J}_f$ .

Le résultat de cette optimisation pour les réponses impulsionnelles est montré à la figure 4.4, où l'on observe aussi que les modules des coefficients des filtres de synthèse après optimisation sont plus concentrés qu'avec le résultat de la pseudo-inverse. En observant l'extension séparable 2D de ces bancs de filtres de synthèse, on voit très bien que les filtres après optimisation sont bien mieux localisés et plus réguliers. Ceci est illustré à la figure 4.5, montrant les parties réelles et imaginaires des coefficients des filtres obtenus par pseudo-inverse et après optimisation. Ce résultat est un cas particulier. Il n'y a, en effet, aucune raison en général pour qu'une optimisation sur les réponses fréquentielles

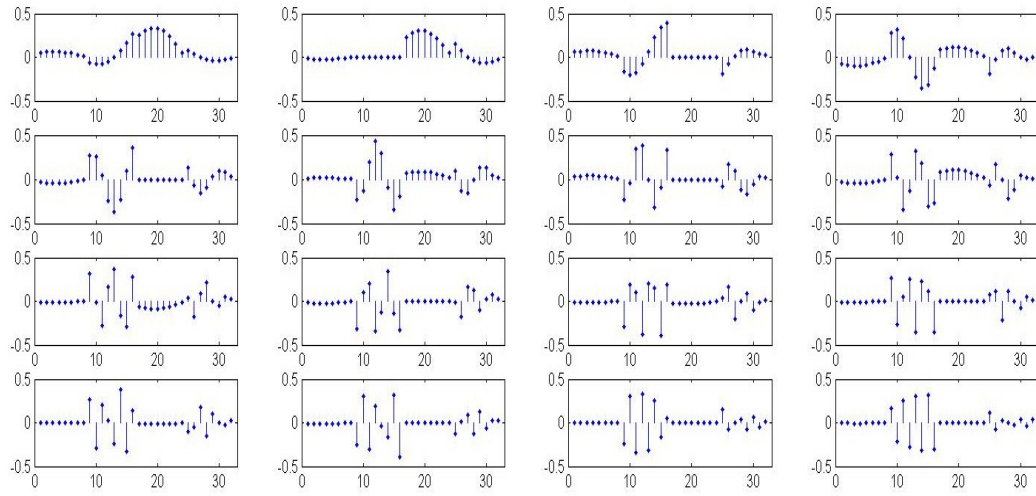


(a)

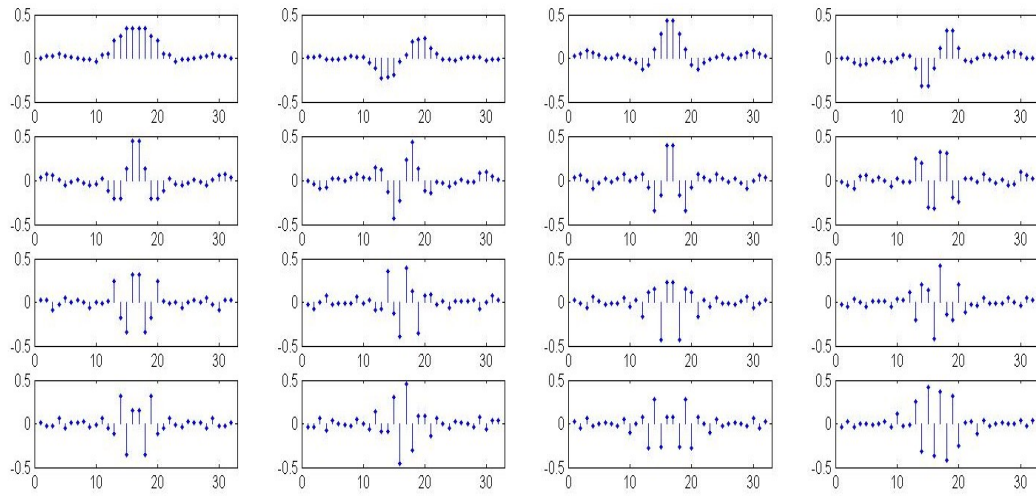


(b)

FIGURE 4.1 – Premier exemple (avec le banc de filtres de la section 3.5.1) : réponses fréquentielles du banc de filtres de synthèse obtenu (a) par pseudo-inverse et (b) et après optimisation avec la fonction  $\tilde{J}_t$ .



(a)



(b)

FIGURE 4.2 – Premier exemple : Réponses impulsionnelles du banc de filtres de synthèse obtenu (a) par pseudo-inverse et (b) et après optimisation avec la fonction  $\tilde{J}_t$ .

entraîne une amélioration de la localisation temporelle des réponses impulsionnelles des filtres.

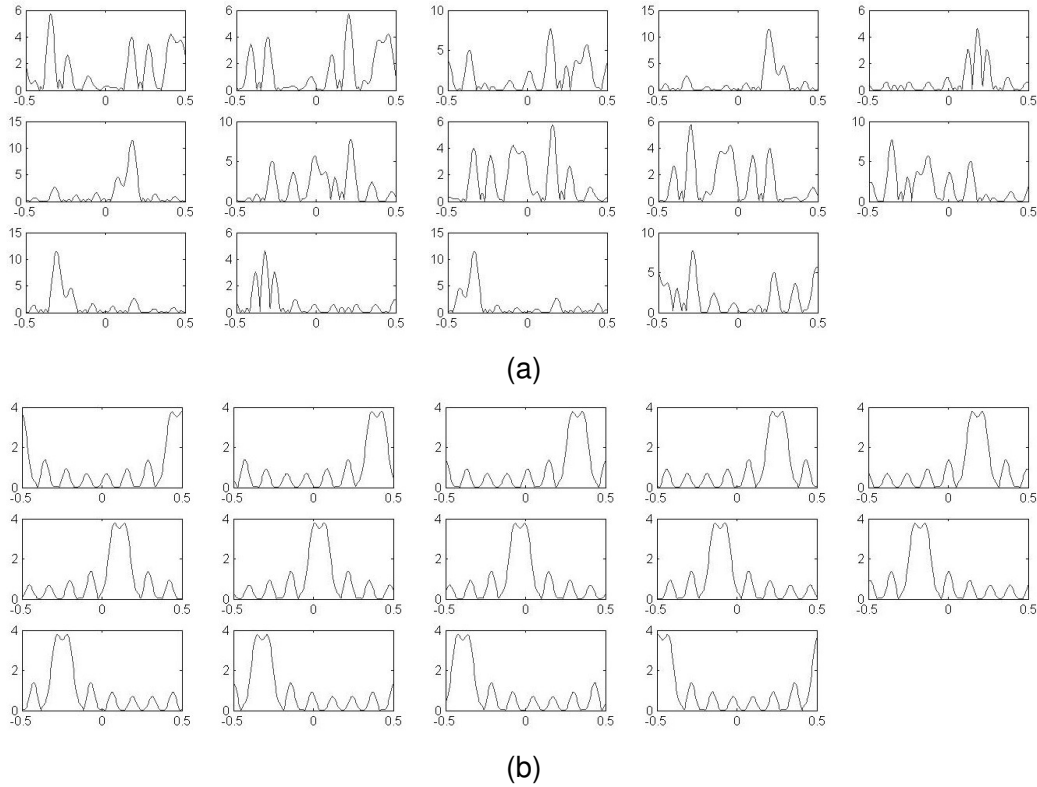


FIGURE 4.3 – Deuxième exemple (banc de filtres de la section 3.5.2 avec  $h_{a1}$ ) : réponses fréquentielles des filtres de synthèse (a) avant optimisation (pseudo-inverse) et (b) après optimisation avec la fonction  $\tilde{J}_f$ .

#### 4.5.3-b) Résultats dans le cas symétrique

Enfin, nous avons testé la procédure d'optimisation dans le cas symétrique sur le banc de filtres de la section 3.5.2 avec la fenêtre  $h_{a2}$ . Nous avons utilisé dans ce cas les fonctions de coût  $\tilde{J}_{ts}$  et  $\tilde{J}_{fs}$ . À nouveau, nous avons utilisé les paramètres :  $N = 8$ ,  $k = 3$  et  $k' = 7/4$ . Les résultats de ces tests sont montrés aux figures 4.6 et 4.7. On constate que les résultats d'optimisation avec les deux fonctions de coût sont différents : avec  $\tilde{J}_{ts}$  les réponses impulsionnelles sont plus concentrées qu'avec  $\tilde{J}_{fs}$  et les réponses fréquentielles présentent une meilleure sélectivité avec  $\tilde{J}_{fs}$  qu'avec  $\tilde{J}_{ts}$ , ce qui est conforme au choix des fonctions de coût.

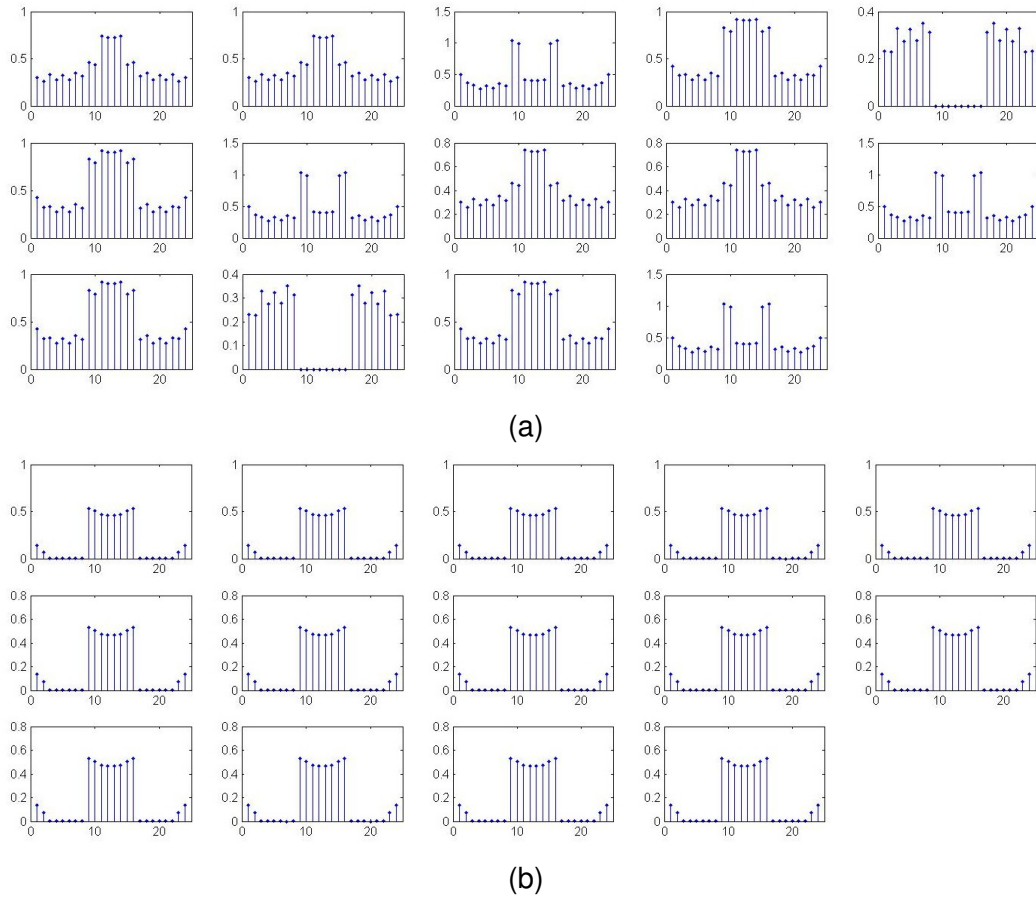


FIGURE 4.4 – Deuxième exemple : module des réponses impulsionnelles des filtres de synthèses (a) avant et (b) après optimisation avec la fonction  $\tilde{\mathcal{J}}_f$ .

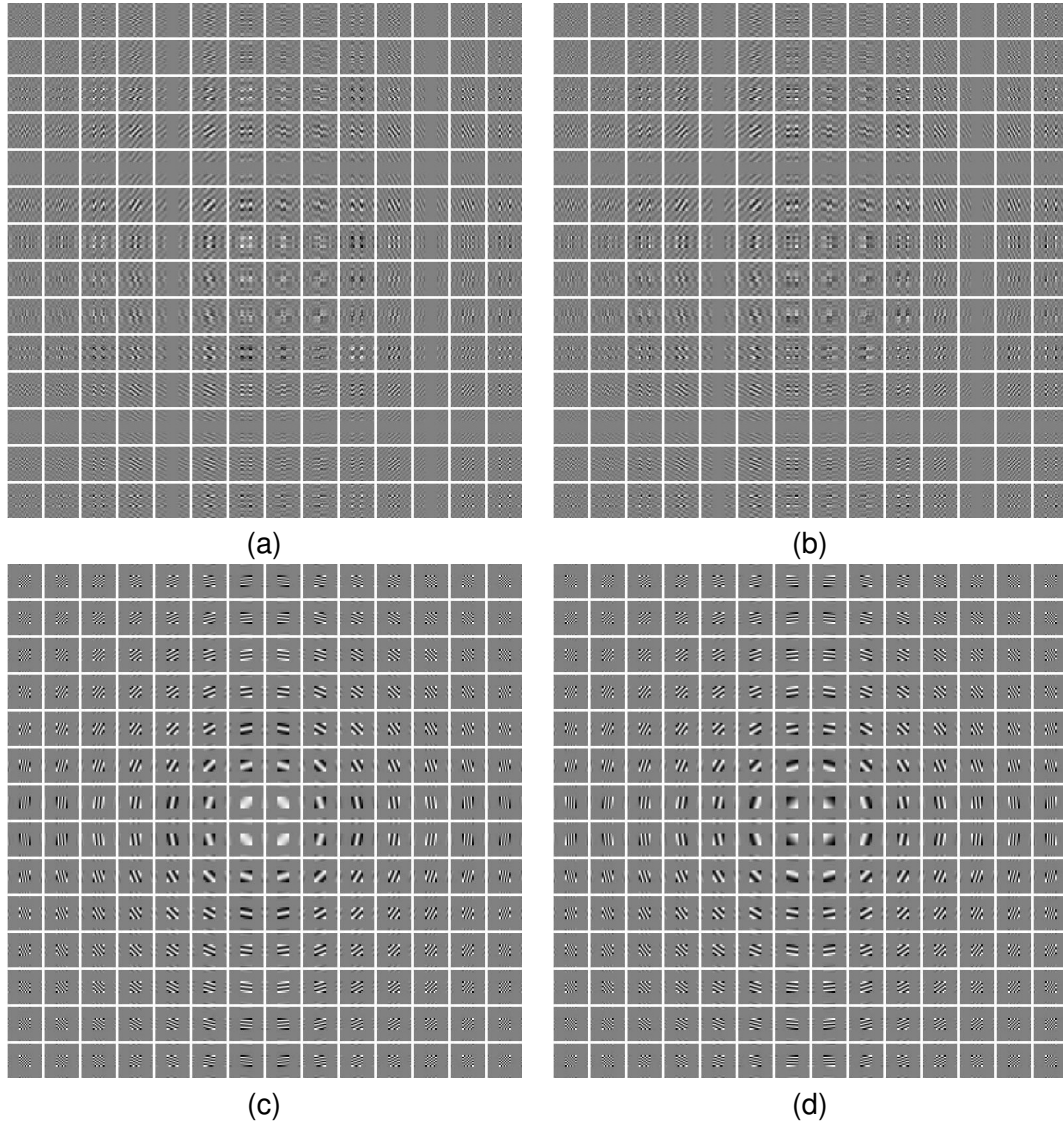


FIGURE 4.5 – Deuxième exemple : parties réelles et imaginaires des réponses impulsionnelles de filtres de synthèses, étendus en 2D de manière séparable, dans les cas pseudo-inverse (a)(b) et optimisé (c)(d).



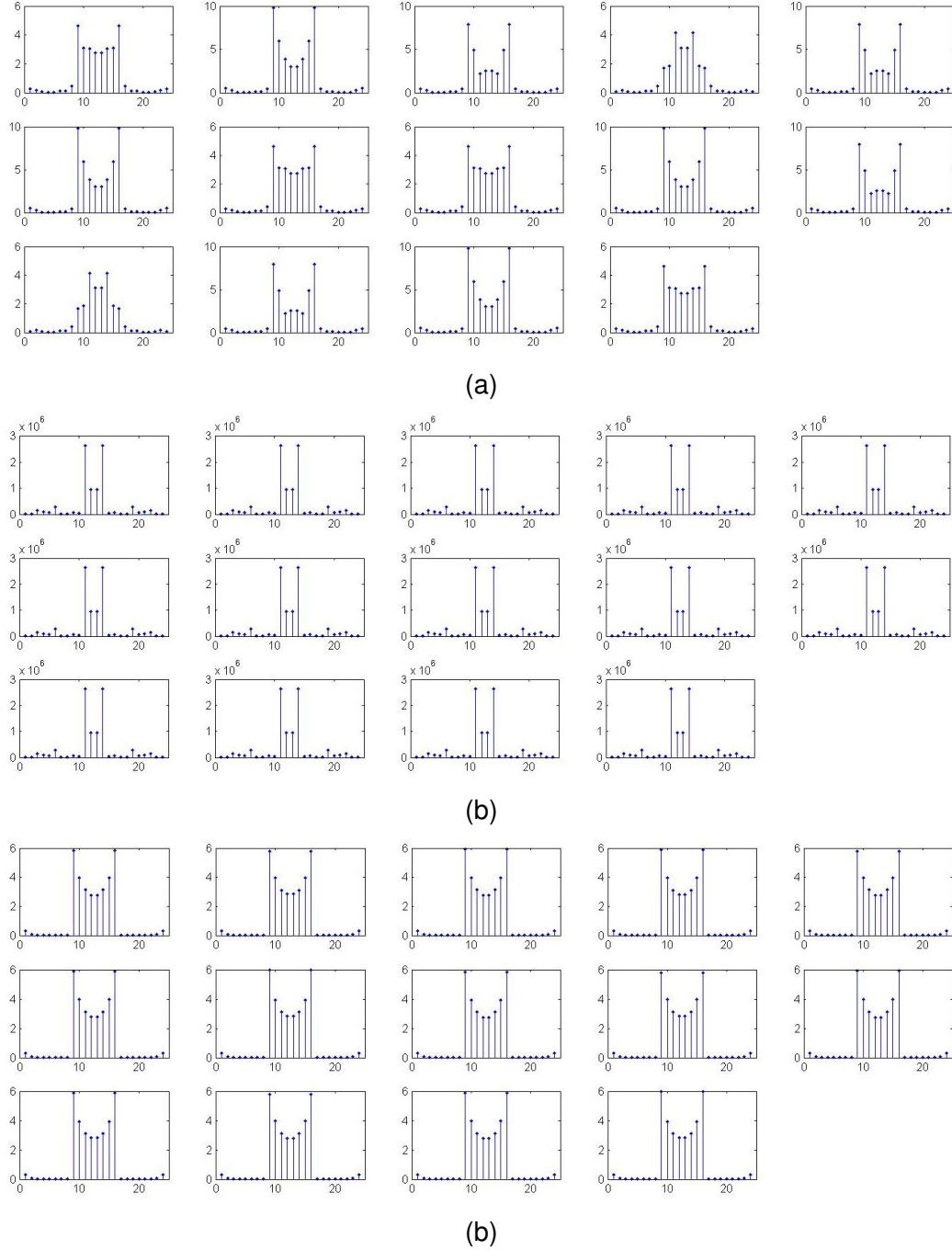
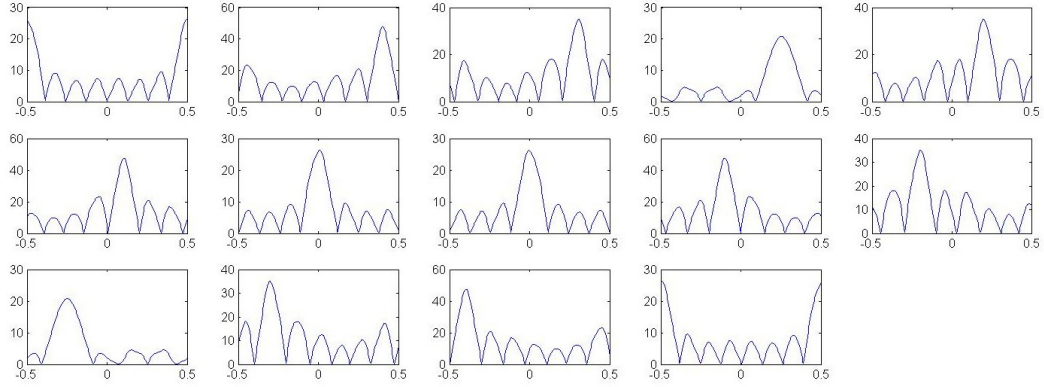
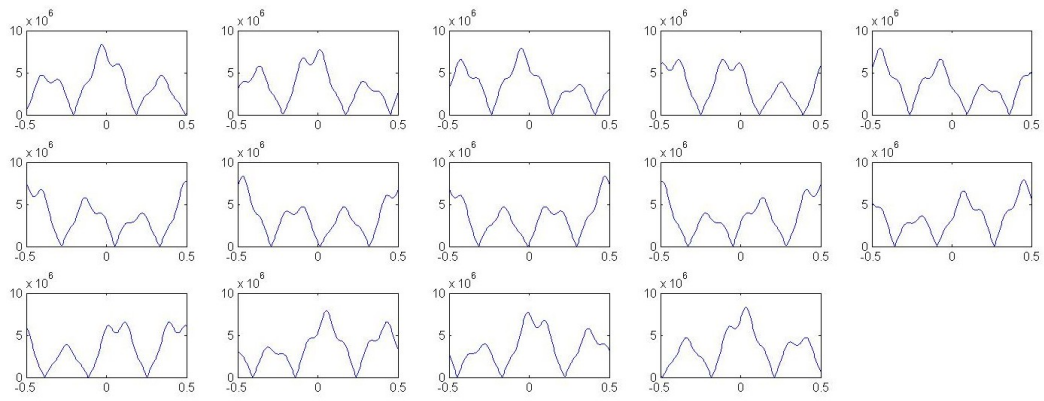


FIGURE 4.6 – Troisième exemple (banc de filtres de la section 3.5.2 avec  $h_{a_2}$ ) : module des réponses impulsionnelles des filtres de synthèses (a) avant et après optimisation (b) avec la fonction de coût  $\widetilde{J}_{ts}$  et (c) avec la fonction de coût  $\widetilde{J}_{fs}$ .

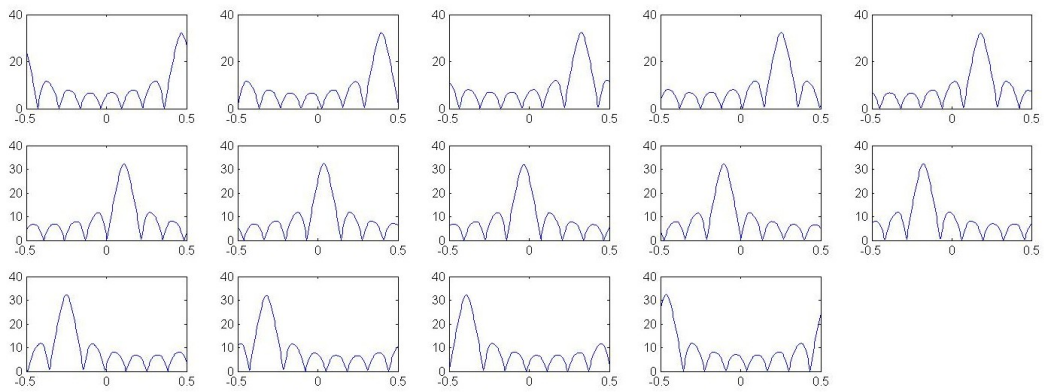




(a)



(b)



(b)

FIGURE 4.7 – Troisième exemple : réponses fréquentielles des filtres de synthèses (a) avant et après optimisation (b) avec la fonction de coût  $\widetilde{J}_{ts}$  et (c) avec la fonction de coût  $\widetilde{J}_{fs}$ .

## 4.6 Conclusion

À l'aide du formalisme introduit au chapitre 3, nous avons pu proposer ici une paramétrisation de l'ensemble des matrices inverses (dans le cas général et le cas symétrique) décrivant les filtres de synthèse. Cette paramétrisation nous a permis de réécrire le problème d'optimisation contraint sous la forme d'un problème de minimisation non contraint, gagnant ainsi en temps de calcul de manière conséquente.

Le tableau 4.3 résume les différentes fonctions de coût qui ont été utilisées dans ce travail pour optimiser les caractéristiques de sélectivité fréquentielle et de localisation temporelle. Les résultats présentés ont montré l'efficacité de ces fonctions pour optimiser ces caractéristiques.

Fonction de coût : $\tilde{J}(\mathcal{C}) = \sum_{j=0}^{M-1} \frac{\ \mathbf{A}_j \mathcal{C} + \mathbf{H}_j^0\ _{K_j}^2}{\ \mathbf{A}_j \mathcal{C} + \mathbf{H}_j^0\ _{\Lambda}^2}$			
$\tilde{J}_t$	$\mathcal{C}$ complexe	$\mathbf{A}_j = \mathbf{V}_j$	$K_j = K_j^t$
$\tilde{J}_f$	$\mathcal{C}$ complexe	$\mathbf{A}_j = \mathbf{V}_j$	$K_j = K_j^f$
$\tilde{J}_{ts}$	$\mathcal{C}$ réelle	$\mathbf{A}_j = \mathbf{W}_j$	$K_j = K_j^t$
$\tilde{J}_{fs}$	$\mathcal{C}$ réelle	$\mathbf{A}_j = \mathbf{W}_j$	$K_j = K_j^f$

TABLE 4.3 – Tableau récapitulatif des différentes fonctions de coût utilisée dans ce travail.

---

## Chapitre 5

# Bancs de filtres multidimensionnels sous-échantillonnés rectangulairement

---

Dans ce chapitre, nous allons étudier quelques aspects du cas multidimensionnel. Après avoir présenté les notations, le problème de l'inversibilité de bancs de filtres d'analyse multidimensionnels sera étudié. Dans un premier temps, nous proposerons une extension de la méthode de caractérisation de l'inversibilité que nous avons proposée au chapitre 3. Cette méthode utilisant des résultants ne donnant pas une condition nécessaire et suffisante à l'existence d'un inverse, nous rappellerons brièvement une autre méthode utilisant les bases de Gröbner et nous comparerons les mérites respectifs de ces deux méthodes. Ensuite nous proposerons une extension de la méthode unidimensionnelle de construction d'un banc de filtres de synthèse. Enfin, nous verrons un cas particulier de bancs de filtres multidimensionnels pour lequel il est possible de donner une expression explicite de l'inverse.

### 5.1 Notations dans le cadre multidimensionnel

#### 5.1.1 Objectif

Une manière classique de traiter des données multidimensionnelles consiste à utiliser des méthodes unidimensionnelles appliquées dans chaque dimension, ce qui formellement revient à utiliser le produit tensoriel des traitements unidimensionnels. Dans le cas linéaire, on obtient ainsi des transformations dites *séparables*. Les transformations séparables sont souvent employées avec succès pour des applications de compression (norme JPEG), de débruitage ou de restauration d'images par exemple. Cependant pour traiter certaines données, par exemple les données de climatologie ou d'écoulement de fluides [Xia, Suter, 1995], il peut être important de considérer des transformées non séparables, offrant des propriétés supplémentaires. Par exemple, un banc de filtres RIF non

séparable peut être orthogonal et à phase linéaire ce qui n'est pas vrai dans le cas séparable (sauf dans le cas de Haar) [Kovačević, Vetterli, 1995]. L'intérêt de transformées non séparables a également été mis en évidence pour traiter des données contenant des informations de type géométrique par des méthodes comme les Curvelets, les Contourlets ou les Bandelettes. L'une des applications privilégiées de cette thèse étant le filtrage directionnel de données anisotropes, il semble également intéressant d'utiliser la plus grande liberté de construction de bancs de filtres d'analyse offerte dans le cas multidimensionnel pour traiter des données pour lesquelles un *a priori* sur la direction dominante serait connu.

Dans les chapitres précédents, nous avons vu comment vérifier qu'un banc de filtres d'analyse unidimensionnel suréchantillonné est inversible, comment en calculer un inverse et éventuellement l'optimiser. Dans ce chapitre nous allons supposer que les bancs de filtres multidimensionnels ne sont pas nécessairement séparables et notre but va être d'examiner si les propriétés obtenues dans le cas unidimensionnel s'étendent en dimension supérieure.

Dans la suite, nous noterons  $L$  la dimension du banc de filtres d'analyse ( $L = 2$  pour une image). Le but, dans un premier temps, est le même qu'au chapitre 3 : caractériser l'inversibilité d'un banc de filtres d'analyse suréchantillonné donné. Par la suite, nous verrons comment obtenir (au moins) un banc de filtres de synthèse.

Le facteur de décimation des filtres dans chaque dimension est encore  $N$ . Ceci correspond à un sous-échantillonnage suivant un réseau rectangulaire (ou plutôt carré), ce qui n'est pas toujours le cas pour des bancs de filtres multidimensionnels. Il existe par exemple des sous-échantillonnages en quinconce ou hexagonaux qui sont couramment employés. On suppose que la redondance dans chaque direction est de  $k'$  et donc la redondance de la transformation complète est  $k'^L$ . Dans la suite, on conserve la notation  $M = k'N$  qui correspond au nombre de sous bandes associées à chaque direction. Les résultats présentés dans la suite s'adaptent au cas où l'on suppose que le facteur de recouvrement, la redondance et le sous-échantillonnage sont différents dans chaque direction. Cependant, la complexité des notations dans ce cadre plus général rendant la lecture plus difficile, nous nous sommes restreints à des valeurs identiques dans chaque direction.

Nous allons introduire les notations polyphases dans le cadre général de transformées de dimension  $L$ . Dans la majeure partie de ce chapitre nous allons cependant nous restreindre à la dimension  $L = 2$  pour des raisons de simplicité d'écriture. À l'exception de l'algorithme de la section 5.2.2, tous les résultats présentés s'étendent naturellement en dimension supérieure.

## 5.1.2 Notation polyphase

### 5.1.2-a) Matrice polyphase du banc de filtres d'analyse

Dans la suite les multi-indices  $(i_1, \dots, i_L)$  seront notés :  $\mathbf{i}$ . Par ailleurs, nous travaillons désormais avec  $L$  variables complexes  $\mathbf{z} = (z_1, \dots, z_L)$  à la place de la variable  $z$ . Nous noterons  $\mathbf{z}^{\mathbf{i}} = z_1^{i_1} \cdots z_L^{i_L}$ ,  $\mathbf{z}$  élevé à la "multi-puissance"  $\mathbf{i}$ .

Soient  $(x(\mathbf{n}))_{\mathbf{n} \in \mathbb{Z}^L}$  le signal multidimensionnel d'entrée et  $(y_{\mathbf{i}}(\mathbf{n}))_{\mathbf{i} \in [0, \dots, M-1]^L, \mathbf{n} \in \mathbb{Z}^L}$  la sortie du banc de filtres d'analyse considéré dont les réponses impulsionnelles sont no-

tées  $h_i(\mathbf{p})$  avec  $\mathbf{i} \in [0, \dots, M-1]^L$  et  $\mathbf{p} \in \mathbb{Z}^L$ . La sortie de chaque branche du banc de filtres s'écrit alors :

$$\begin{aligned} y_i(\mathbf{n}) &= \sum_{\mathbf{p}} h_i(\mathbf{p}) x(Nn_1 - p_1, \dots, Nn_L - p_L) \\ &= \sum_{\ell} \sum_{j_1=0}^{N-1} \cdots \sum_{j_L=0}^{N-1} h_i(N\ell_1 + j_1, \dots, N\ell_L + j_L) \\ &\quad x(N_1(n_1 - \ell_1) - j_1, \dots, N(n_L - \ell_L) - j_L) \end{aligned}$$

En notant :  $x_j(\mathbf{n}) = x(Nn_1 - j_1, \dots, Nn_L - j_L)$  et  $h_{i,j}(\ell) = h_i(N\ell_1 + j_1, \dots, N\ell_L + j_L)$ , on a :

$$y_i(\mathbf{n}) = \sum_{j_1=0}^{N-1} \cdots \sum_{j_L=0}^{N-1} \sum_{\ell} h_{i,j}(\ell) x_j(\mathbf{n} - \ell)$$

En effectuant une transformée en  $\mathbf{z}$  on obtient :

$$y_i[\mathbf{z}] = \sum_{j_1=0}^{N-1} \cdots \sum_{j_L=0}^{N-1} h_{i,j}[\mathbf{z}] x_j[\mathbf{z}]$$

avec :  $h_{i,j}[\mathbf{z}] = \sum_{\ell \in [0, \dots, k-1]^L} h_{i,j}(\ell) \mathbf{z}^{-\ell}$ .

On peut alors vectoriser l'expression ci-dessus pour se ramener à une formulation matricielle, plus proche du cas unidimensionnel. On pose : pour tout  $(i_1, \dots, i_L) \in \{0, \dots, M-1\}^L$  et  $(j_1, \dots, j_L) \in \{0, \dots, N-1\}^L$ ,

$$\begin{aligned} y_{i_1+i_2M+\dots+i_LM^{L-1}}[\mathbf{z}] &= y_i[\mathbf{z}], \\ H_{i_1+i_2M+\dots+i_LM^{L-1}, j_1+j_2N+\dots+j_LN^{L-1}}[\mathbf{z}] &= h_{i,j}[\mathbf{z}] \quad \text{et} \\ x_{j_1+j_2N+\dots+j_LN^{L-1}}[\mathbf{z}] &= x_j[\mathbf{z}]. \end{aligned}$$

On obtient alors :

$$\forall i \in \{0, \dots, M^L - 1\}, \quad y_i[\mathbf{z}] = \sum_{j=0}^{N^L-1} H_{i,j}[\mathbf{z}] x_j[\mathbf{z}].$$

On peut donc réécrire la transformation sous la forme du produit matriciel suivant :  $\mathbf{y}[\mathbf{z}] = \mathbf{H}[\mathbf{z}] \mathbf{x}[\mathbf{z}]$ , où  $\mathbf{x}[\mathbf{z}] = (x_j[\mathbf{z}])_{0 \leq j \leq N^L-1}$  et  $\mathbf{y}[\mathbf{z}] = (y_i[\mathbf{z}])_{0 \leq i \leq M^L-1}$ . La matrice polyphase du banc de filtres d'analyse s'écrit sous la forme :

$$\mathbf{H}[\mathbf{z}] = \sum_{\ell \in \{0, \dots, k-1\}^L} \mathbf{H}(\ell) \mathbf{z}^{-\ell} \in \mathbb{C}[\mathbf{z}, \mathbf{z}^{-1}]^{M^L \times N^L},$$

où :

$$(\mathbf{H}(\ell))_{i_1+\dots+i_LM^{L-1}, j_1+\dots+j_LN^{L-1}} = h_i(N\ell + \mathbf{j}). \quad (5.1)$$

### 5.1.2-b) Matrice polyphase du banc de filtres de synthèse

Étudions de la même manière l'étape de reconstruction. Soit  $\tilde{x}$  la sortie du banc de filtres de synthèse dont on note les réponses impulsionnelles :  $\tilde{h}_{j_1, \dots, j_L}(n_1, \dots, n_L)$ , pour tout  $(n_1, \dots, n_L) \in \mathbb{Z}^L$  et  $(j_1, \dots, j_L) \in \{0, \dots, M-1\}^L$ . Avec ces notations, on peut écrire, pour tout  $\mathbf{m} \in \mathbb{Z}^L$  :

$$\tilde{x}(\mathbf{m}) = \sum_{\mathbf{j} \in \{0, \dots, M-1\}^L} \sum_{\ell \in \mathbb{Z}^L} \tilde{h}_{\mathbf{j}}(\mathbf{m} - N\ell) y_{\mathbf{j}}(\ell).$$

Avec le changement de variable  $m_\ell = n_\ell N - i_\ell$ , on obtient, pour tous  $\mathbf{n} \in \mathbb{Z}^L$  et  $\mathbf{i} \in \{0, \dots, N-1\}^L$  :

$$\tilde{x}(\mathbf{n}N - \mathbf{i}) = \sum_{\mathbf{j} \in \{0, \dots, M-1\}^L} \sum_{\ell \in \mathbb{Z}^L} \tilde{h}_{\mathbf{j}}((\mathbf{n} - \ell)N - \mathbf{i}) y_{\mathbf{j}}(\ell).$$

En posant, pour tout  $\mathbf{i} \in \{0, \dots, N-1\}^L$ ,  $\mathbf{j} \in \{0, \dots, M-1\}^L$  et  $(\mathbf{n}, \ell) \in \mathbb{Z}^L \times \mathbb{Z}^L$  :

$$\tilde{h}_{\mathbf{i}, \mathbf{j}}(\mathbf{n} - \ell) = \tilde{h}_{\mathbf{j}}(N(\mathbf{n} - \ell) - \mathbf{i})$$

et

$$\tilde{x}_{\mathbf{i}}(\mathbf{n}) = \tilde{x}(\mathbf{n}N - \mathbf{i}),$$

l'équation précédente devient :

$$\tilde{x}_{\mathbf{i}}(\mathbf{n}) = \sum_{\mathbf{j} \in \{0, \dots, M-1\}^L} \sum_{\ell \in \mathbb{Z}^L} \tilde{h}_{\mathbf{i}, \mathbf{j}}(\mathbf{n} - \ell) y_{\mathbf{j}}(\ell).$$

En considérant la transformée en  $\mathbf{z}$  de cette expression, on a :

$$\tilde{x}_{\mathbf{i}}[\mathbf{z}] = \sum_{\mathbf{j} \in \{0, \dots, M-1\}^L} \tilde{h}_{\mathbf{i}, \mathbf{j}}[\mathbf{z}] y_{\mathbf{j}}[\mathbf{z}],$$

ce qui après la vectorisation :

$$\begin{aligned} \tilde{H}_{i_1+i_2N+\dots+i_LN^{L-1}, j_1+j_2M+\dots+j_LM^{L-1}}[\mathbf{z}] &= \tilde{h}_{\mathbf{i}, \mathbf{j}}[\mathbf{z}] \quad \text{et} \\ \tilde{x}_{i_1+i_2N+\dots+i_LN^{L-1}}[\mathbf{z}] &= \tilde{x}_{\mathbf{i}}[\mathbf{z}], \end{aligned}$$

peut se réécrire sous la forme du produit matriciel suivant :

$$\tilde{\mathbf{x}}[\mathbf{z}] = \tilde{\mathbf{H}}[\mathbf{z}] \mathbf{y}[\mathbf{z}],$$

où  $\tilde{\mathbf{x}}[\mathbf{z}] = (\tilde{x}_{\mathbf{i}}[\mathbf{z}])_{0 \leq i \leq N^L-1}$ .

L'écriture polyphase nous permet encore une fois d'exprimer facilement la propriété de reconstruction parfaite [Vaidyanathan, 1993] : le but est de trouver une matrice  $\tilde{\mathbf{H}}[\mathbf{z}]$  dans  $\mathbb{C}[\mathbf{z}, \mathbf{z}^{-1}]^{N^L \times M^L}$  telle que  $\tilde{\mathbf{H}}[\mathbf{z}] \mathbf{H}[\mathbf{z}] = \mathbf{I}_{N^L}$ .

Notons que le banc de filtres de synthèse étant contraint à être RIF, sa matrice polyphase s'écrit sous la forme :

$$\begin{aligned} \tilde{\mathbf{H}}[\mathbf{z}] &= \sum_{\ell \in \{-p_1, \dots, q_1\} \times \dots \times \{-p_L, \dots, q_L\}} \tilde{\mathbf{H}}(\ell) \mathbf{z}^{-\ell} \in \mathbb{C}[\mathbf{z}, \mathbf{z}^{-1}]^{N^L \times M^L}, \quad \text{où} \\ (\tilde{\mathbf{H}}(\ell))_{i_1+i_2N+\dots+i_LN^{L-1}, j_1+j_2M+\dots+j_LM^{L-1}} &= \tilde{h}_{\mathbf{j}}(N\ell - \mathbf{i}). \end{aligned} \quad (5.2)$$

## 5.2 Inversibilité et résultants

Dans cette partie, nous allons tenter d'étendre la méthode de caractérisation de l'inversibilité de la section 3.3. La dimension des signaux considérés sera ici :  $L = 2$ . Pour ce faire, nous devons au préalable introduire quelques résultats sur les matrices polynomiales et sur la notion de résultant qui vont nous permettre de construire un algorithme de test d'inversibilité.

### 5.2.1 Résultats préliminaires

Précisons tout d'abord quelques définitions.

**Définition 5.1** Une matrice polynomiale de Laurent  $\mathbf{G} \in \mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]^{p \times m}$  est dite right zero prime (rZP) si  $p \geq m$  et que l'idéal  $\mathcal{I}_{\mathbf{G}}$ , engendré par les mineurs maximaux, est l'anneau  $\mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]$ .

**Définition 5.2** On définit pour tout polynôme non nul  $P \in \mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]$  la variété de Laurent de  $P$  comme étant :

$$\mathcal{V}_L(P) = \{(\alpha, \beta) \in (\mathbb{C}^*)^2 \mid P(\alpha, \beta) = 0\}$$

Si  $\mathcal{I} = \{P_i, \quad i \in \{1, \dots, N\}\}$  est un ensemble de polynômes, sa variété de Laurent sera définie par :

$$\mathcal{V}_L(\mathcal{I}) = \cap_{i=1}^N \mathcal{V}_L(P_i)$$

Si  $\mathbf{G}$  est une matrice polynomiale, on note  $\mathcal{V}_L(\mathbf{G})$  la variété de Laurent de ses mineurs maximaux.

À l'aide de ces définitions on peut alors énoncer le théorème suivant dû à Fornasini et Valcher.

**Théorème 4** Soit  $\mathbf{G} = [g_{i,j}] \in \mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]^{p \times m}$ ,  $p > m$ . Les propositions suivantes sont alors équivalentes :

1.  $\mathbf{G}$  est rZP.
2.  $\mathcal{V}_L(\mathbf{G})$  est vide (ou de manière équivalente,  $\mathbf{G}$  est de rang plein en tout point de  $(\mathbb{C}^*)^2$ ),
3.  $\mathbf{G}$  admet une matrice polynomiale inverse à gauche i.e. il existe une matrice  $\mathbf{H} \in \mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]^{m \times p}$  telle que :

$$\mathbf{H}(\mathbf{z})\mathbf{G}(\mathbf{z}) = \mathbf{I}_m.$$

Preuve : L'équivalence entre les points 1 et 3 a été démontrée dans [Fornasini, Valcher, 1997].

Il suffit de prouver l'équivalence entre les points 1 et 2 pour pouvoir conclure.

Si  $\mathbf{G}$  est rZP, l'idéal engendré par ses mineurs maximaux est l'anneau  $\mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]$  en entier. Supposons que  $\mathcal{V}_L(\mathbf{G})$  soit non vide. Il existe un couple  $(\alpha, \beta) \in (\mathbb{C}^*)^2$  tel que pour tout mineur maximal de  $\mathbf{G}$ , noté  $P_{G_i}$  avec  $i \in \{1, \dots, \binom{p}{m}\}$ , on ait :  $P_{G_i}(\alpha, \beta) = 0$ . Soit  $P \in$

$\mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}] = \mathcal{I}_G$ . Le polynôme  $P$  s'écrit donc comme une combinaison non triviale d'éléments  $a_i$  de  $\mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]$  :

$$P = \sum_{i=1}^{\binom{p}{m}} a_i P_{G_i}.$$

On en déduit :  $P(\alpha, \beta) = 0$ , ce qui est faux (par exemple pour un polynôme  $P$  constant non nul). On a ainsi montré par l'absurde que  $\mathcal{V}_L(G) = \emptyset$ .

Réciproquement, supposons que la variété  $\mathcal{V}_L(G)$  soit vide, c'est-à-dire que les  $(P_{G_i})_i$ , mineurs maximaux de  $G$ , n'ont pas de zéros communs. Montrons que  $\mathcal{I}_G = \mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]$ . L'ensemble des mineurs maximaux étant fini, il existe un couple  $(h_1, h_2) \in (\mathbb{N}^*)^2$  tel que :

$$\forall i \in \left\{1, \dots, \binom{p}{m}\right\}, \quad z_1^{h_1-1} z_2^{h_2-1} P_{G_i}(\mathbf{z}) \in \mathbb{C}[z_1, z_2].$$

Soient les polynômes  $Q_{G_i}$  définis par :

$$\forall i \in \left\{1, \dots, \binom{p}{m}\right\}, \quad Q_{G_i}(\mathbf{z}) = z_1^{h_1} z_2^{h_2} P_{G_i}(\mathbf{z}) \in \mathbb{C}[z_1, z_2].$$

Considérons désormais ces  $\binom{p}{m}$  polynômes. Puisque les polynômes  $(P_{G_i})_i$  n'ont pas de zéros communs, il est clair que leur variété est l'ensemble :

$$\mathcal{V} = \{(\alpha, \beta) \in \mathbb{C}^2, \alpha\beta = 0\}.$$

Rappelons l'énoncé du théorème des zéros de Hilbert (aussi connu sous le nom théorème *Nullstellensatz* de Hilbert) [Fulton, 1969] que nous allons maintenant pouvoir utiliser.

**Théorème 5 (Zéros de Hilbert)** *Soient  $f \in \mathbb{C}[z_1, \dots, z_N]$  et  $\mathcal{I}$  un idéal de  $\mathbb{C}[z_1, \dots, z_N]$ . Si pour tout  $(a_1, \dots, a_N) \in \mathcal{V}(\mathcal{I})$ , on a :  $f(a_1, \dots, a_N) = 0$  alors il existe  $k \in \mathbb{N}^*$  tel que  $f^k \in \mathcal{I}$ .*

On applique alors le théorème des zéros de Hilbert au polynôme  $f(z_1, z_2) = z_1 z_2$ , il existe  $k \in \mathbb{N}^*$  et  $(X_i)_{i=1}^{\binom{p}{m}} \in (\mathbb{C}[z_1, z_2])^{\binom{p}{m}}$  tels que :

$$z_1^k z_2^k = \sum_{i=1}^{\binom{p}{m}} X_i(\mathbf{z}) Q_{G_i}(\mathbf{z}),$$

et donc :

$$1 = \sum_{i=1}^{\binom{p}{m}} Y_i(\mathbf{z}) P_{G_i}(\mathbf{z}),$$

où  $Y_i(z_1, z_2) = z_1^{h_1-k} z_2^{h_2-k} X_i(z_1, z_2) \in \mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]$ . On a montré que  $1 \in \mathcal{I}_G$  et donc  $\mathcal{I}_G = \mathbb{C}[z_1, z_2, z_1^{-1}, z_2^{-1}]$ , ce qui établit le résultat. ■

**Remarque 5.1** *Dans la preuve du théorème précédent, l'équivalence  $1 \Leftrightarrow 2$  n'a pas été publiée à notre connaissance. Elle est due à Maria Elena Valcher dans une communication privée.*

Pour savoir si une matrice polynomiale est inversible à gauche, il suffit donc de vérifier que les zéros communs des déterminants mineurs maximaux ne sont pas dans  $(\mathbb{C}^*)^2$ . Pour ce faire, nous avons pensé utiliser la notion de résultants.



**Définition 5.3** Soient deux polynômes  $P$  et  $Q$  sur un anneau  $\mathcal{A}$  de degré  $n$  et  $m$  respectivement. On note  $P(X) = a_n X^n + a_{n-1} X^{n-1} + \dots + a_1 X + a_0$  et  $Q(X) = b_m X^m + b_{m-1} X^{m-1} + \dots + b_1 X + b_0$  où  $(a_i)_{i=1}^n \in \mathcal{A}^{n+1}$  et  $(b_i)_{i=1}^m \in \mathcal{A}^{m+1}$ . Le résultant de  $P$  et  $Q$  est alors défini comme le déterminant de la matrice de Sylvester de taille  $(m+n) \times (m+n)$  de ces deux polynômes. Autrement dit, le résultant s'écrit sous la forme :

$$\text{Res}(P, Q) = \det \left( \begin{array}{cccccc} a_n & \cdots & a_0 & 0 & & \\ & \ddots & & & \ddots & \\ 0 & & a_n & \cdots & & a_0 \\ b_m & \cdots & b_0 & & & 0 \\ & \ddots & & \ddots & & \\ & & & & \ddots & \\ 0 & & & b_m & \cdots & b_0 \end{array} \right) \left. \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} m \text{ lignes} \\ \\ \\ n \text{ lignes} \end{array}$$

(en supposant dans cette représentation :  $n > m$ ).

Dans la définition précédente, l'anneau  $\mathcal{A}$  peut être pris égal à un anneau de polynômes. Dans ce cas les polynômes  $P$  et  $Q$  sont de deux variables et le résultant devient alors un polynôme d'une variable. Dans la suite, nous allons travailler avec  $\mathcal{A} = \mathbb{C}[z]$ . Nous noterons  $\text{Res}_{z_1}(P, Q)(z_2)$  (respectivement  $\text{Res}_{z_2}(P, Q)(z_1)$ ) le résultant de  $P$  et  $Q$  par rapport à l'anneau  $\mathbb{C}[z_2]$  (respectivement  $\mathbb{C}[z_1]$ ).

**Exemple 6** Soient les polynômes :

$$\begin{aligned} P(z_1, z_2) &= 2z_1^2 z_2^3 + 3z_1^2 z_2^2 - z_1^2 z_2 + 2z_1^2 - 3z_1 z_2^3 + 2z_1 z_2^2 - z_1 z_2 + 6z_1 - 3z_2^3 + 2z_2^2 + 8z_2 - 9, \\ Q(z_1, z_2) &= 6z_1^2 z_2^2 + 6z_1^3 z_2 - 13z_1^2 z_2 + 2z_1^3 - 15z_1 z_2^2 - 5z_1^2 - 9z_2^2 - 3z_2 - 3z_1. \end{aligned}$$

Calculons  $\text{Res}_{z_2}(P, Q)(z_1)$ . Pour ce faire nous allons d'abord réécrire ces polynômes de  $\mathbb{C}[z_1, z_2]$  dans  $\mathbb{C}[z_1][z_2]$  :

$$\begin{aligned} P(z_1, z_2) &= (2z_1^2 - 3z_1 - 3)z_2^3 + (3z_1^2 + 2z_1 + 2)z_2^2 + (-z_1^2 - z_1 + 8)z_2 + (2z_1^2 + 6z_1 - 9), \\ Q(z_1, z_2) &= (6z_1^2 - 15z_1 - 9)z_2^2 + (6z_1^3 - 13z_1^2 - 3)z_2 + (2z_1^3 - 5z_1^2 - 3z_1). \end{aligned}$$

On peut maintenant faire le calcul du résultant :

$$\begin{aligned} \text{Res}_{z_2}(P, Q)(z_1) &= \begin{vmatrix} 2z_1^2 - 3z_1 - 3 & 3z_1^2 + 2z_1 + 2 & -z_1^2 - z_1 + 8 & 2z_1^2 + 6z_1 - 9 & 0 \\ 0 & 2z_1^2 - 3z_1 - 3 & 3z_1^2 + 2z_1 + 2 & -z_1^2 - z_1 + 8 & 2z_1^2 + 6z_1 - 9 \\ 6z_1^2 - 15z_1 - 9 & 6z_1^3 - 13z_1^2 - 3 & 2z_1^3 - 5z_1^2 - 3z_1 & 0 & 0 \\ 0 & 6z_1^2 - 15z_1 - 9 & 6z_1^3 - 13z_1^2 - 3 & 2z_1^3 - 5z_1^2 - 3z_1 & 0 \\ 0 & 0 & 6z_1^2 - 15z_1 - 9 & 6z_1^3 - 13z_1^2 - 3 & 2z_1^3 - 5z_1^2 - 3z_1 \end{vmatrix} \\ &= -1120z_1^{13} + 8880z_1^{12} - 9720z_1^{11} - 99692z_1^{10} + 360620z_1^9 - 394888z_1^8 + 56058z_1^7 \\ &\quad + 62400z_1^6 - 145200z_1^5 + 198774z_1^4 + 340290z_1^3 - 150768z_1^2 - 286740z_1 - 74358. \end{aligned}$$

Le résultat suivant [Collins, 1971] sur les résultants de polynômes de plusieurs variables va nous permettre de faire le lien entre la notion de résultant et la recherche des zéros communs de polynômes.

**Théorème 6** Soit  $\mathbb{F}$  un corps algébriquement clos<sup>1</sup>. Soient  $A(z_1, z_2) = \sum_{i=0}^m A_i(z_1)z_2^i$  et  $B(z_1, z_2) = \sum_{i=0}^n B_i(z_1)z_2^i$  des éléments de  $\mathbb{F}[z_1, z_2]$  de degrés strictement positifs  $m$  et  $n$  en  $z_2$ . Si  $(a_1, a_2)$  est un zéro commun de  $A$  et  $B$  alors  $\text{Res}_{z_2}(A, B)(a_1) = 0$ . Réciproquement : si  $\text{Res}_{z_2}(A, B)(a_1) = 0$  alors au moins l'une des propositions suivantes est vérifiée :

1.  $A_m(a_1) = \dots = A_0(a_1) = 0$ ,
2.  $B_n(a_1) = \dots = B_0(a_1) = 0$ ,
3.  $A_m(a_1) = B_n(a_1) = 0$ ,
4. Il existe  $a_2 \in \mathbb{F}$  tel que  $(a_1, a_2)$  soit un zéro commun de  $A$  et  $B$ .

*Preuve :* Supposons dans un premier temps que  $(a_1, a_2)$  soit un zéro commun de  $A$  et  $B$ . On considère alors le déterminant de la matrice :

$$M = \begin{pmatrix} A_0(a_1) & \dots & A_m(a_1) & 0 \\ & \ddots & & \ddots \\ 0 & A_0(a_1) & \dots & A_m(a_1) \\ B_0(a_1) & \dots & B_n(a_1) & 0 \\ & \ddots & & \ddots \\ & & & \ddots \\ 0 & B_0(a_1) & \dots & B_n(a_1) \end{pmatrix}.$$

En remplaçant la première colonne de cette matrice par la combinaison linéaire des colonnes pondérée par le vecteur :  $(1, a_2, a_2^2, \dots, a_2^{m+n-1})^\top$  (qui est non triviale même si  $a_2 = 0$ ), on obtient une matrice dont la première colonne vaut :

$$\begin{pmatrix} \sum_{i=0}^m A_i(a_1)a_2^i \\ a_2 \sum_{i=0}^m A_i(a_1)a_2^i \\ \vdots \\ a_2^{n-1} \sum_{i=0}^m A_i(a_1)a_2^i \\ \sum_{i=0}^n B_i(a_1)a_2^i \\ a_2 \sum_{i=0}^n B_i(a_1)a_2^i \\ \vdots \\ a_2^{m-1} \sum_{i=0}^n B_i(a_1)a_2^i \end{pmatrix} = \begin{pmatrix} A(a_1, a_2) \\ a_2 A(a_1, a_2) \\ \vdots \\ a_2^{n-1} A(a_1, a_2) \\ B(a_1, a_2) \\ a_2 B(a_1, a_2) \\ \vdots \\ a_2^{m-1} B(a_1, a_2) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Le vecteur ci-dessus est identiquement nul en vertu de l'hypothèse  $A(a_1, a_2) = B(a_1, a_2) = 0$ . Le déterminant de cette matrice est donc nul, d'où  $\text{Res}_{z_2}(A, B)(a_1) = 0$ .

Réciproquement, supposons que  $\text{Res}_{z_2}(A, B)(a_1) = \det(M) = 0$ . En considérant la dernière colonne de  $M$ , on peut distinguer plusieurs cas :

- Premier cas : la colonne est nulle, on a alors :  $A_m(a_1) = B_n(a_1) = 0$ .
- Deuxième cas : les deux éléments  $A_m(a_1)$  et  $B_n(a_1)$  sont non nuls.
- Troisième cas : seulement un de ces deux éléments est nul.

Dans le deuxième cas, le déterminant de  $M$  représente le résultant des polynômes d'une variable  $A(a_1, z_2)$  et  $B(a_1, z_2)$ , qui sont alors respectivement de degré  $m$  et  $n$ , tous deux strictement positifs. On peut désormais appliquer le théorème classique sur les résultants.

1. Un corps  $\mathbb{F}$  est dit algébriquement clos si tout polynôme non constant de  $\mathbb{F}[X]$  possède au moins une racine dans  $\mathbb{F}$ .

**Théorème 7** Soient  $P$  et  $Q$  des polynômes de degré strictement positif, alors  $\text{Res}(P, Q)$  est nul si et seulement si  $P$  et  $Q$  ont un diviseur commun de degré strictement positif.

Ce théorème nous assure donc que les polynômes  $A(a_1, z_2)$  et  $B(a_1, z_2)$  ont au moins un facteur irréductible commun. Le corps  $\mathbb{F}$  étant algébriquement clos cela signifie qu'il existe un  $a_2 \in \mathbb{F}$  tel que  $(z_2 - a_2)$  soit un facteur commun de  $A(a_1, z_2)$  et  $B(a_1, z_2)$ . Dans ce cas on a donc bien prouvé qu'il existe  $a_2 \in \mathbb{F}$  tel que  $(a_1, a_2)$  soit un zéro commun de  $A$  et  $B$ .

Enfin, étudions le troisième cas. Supposons  $B_n(a_1) = 0$  et  $A_m(a_1) \neq 0$  et soit l'ensemble  $\mathcal{E} = \{j \mid B_j(a_1) \neq 0\}$ . Si  $\mathcal{E} = \emptyset$  alors  $B_n(a_1) = \dots = B_0(a_1) = 0$ , sinon on pose  $n_0 = \max(\mathcal{E}) < n$ . En développant le déterminant ci-dessus successivement par rapport aux colonnes  $m+n$  jusqu'à  $m+n_0+1$ , contenant à chaque étape seulement l'élément  $A_m(a_1)$ , on obtient que  $\text{Res}_{z_2}(A, B)(a_1)$  vaut :

$$((-1)^m A_m(a_1))^{n-n_0} \left| \begin{array}{cccccc} A_0(a_1) & \cdots & & A_m(a_1) & & 0 \\ & \ddots & & & \ddots & \\ 0 & & A_0(a_1) & \cdots & & A_m(a_1) \\ B_0(a_1) & \cdots & B_{n_0}(a_1) & & & 0 \\ & \ddots & & \ddots & & \\ & & \ddots & & \ddots & \\ 0 & & & B_0(a_1) & \cdots & B_{n_0}(a_1) \end{array} \right| \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \end{array} \right\} \begin{array}{l} n_0 \text{ lignes} \\ \\ \\ m \text{ lignes} \end{array} \quad (5.3)$$

Si  $n_0 = 0$  (autrement dit seulement  $B_{n_0}(a_1) \neq 0$ ), alors :

$$0 = \text{Res}_{z_2}(A, B)(a_1) = ((-1)^m A_m(a_1))^n \left| \begin{array}{ccc} B_0(a_1) & & 0 \\ & \ddots & \\ 0 & & B_0(a_1) \end{array} \right| = (-1)^{mn} A_m(a_1)^n B_0(a_1)^m.$$

Comme  $A_m(a_1)$  a été supposé non nul cela signifie que  $B_0(a_1) = 0$ , ce qui n'est pas possible (par construction de  $n_0$ ). On en déduit que  $n_0 \geq 1$ .

Le déterminant (5.3) représente alors le résultant des deux polynômes d'une variable  $A(a_1, z_2)$  et  $\tilde{B}(a_1, z_2) = \sum_{i=0}^{n_0} B_i(a_1)z_2^i = B(a_1, z_2)$ , qui sont alors respectivement de degré  $m$  et  $n_0$ , tous deux strictement positifs. En appliquant le théorème 7 et le même raisonnement que dans le deuxième cas, on en déduit qu'il existe  $a_2 \in \mathbb{F}$  tel que  $(a_1, a_2)$  soit un zéro commun de  $A$  et  $B$ .

De la même manière, en supposant que  $A_m(a_1) = 0$  et  $B_n(a_1) \neq 0$ , on obtient que, soit il existe  $a_2 \in \mathbb{F}$  tel que  $(a_1, a_2)$  soit un zéro commun de  $A$  et  $B$ , soit  $A_m(a_1) = \dots = A_0(a_1) = 0$ . ■

**Remarque 5.2** La preuve proposée ici pour la réciproque est basée sur la forme particulière des matrices de Sylvester. Dans l'article [Collins, 1971] la démonstration de la réciproque est moins simple. Elle est fondée sur l'utilisation du résultat : soient  $A$  et  $B$  deux polynômes de degré strictement positif à coefficients dans un anneau commutatif  $\mathcal{A}$ , alors il existe des polynômes  $S$  et  $T$  à coefficients dans  $\mathcal{A}$  tels que  $AS + BT = \text{Res}(A, B)$ ,  $\deg(S) < \deg(B)$  et  $\deg(T) < \deg(A)$ .

Le quatrième cas apparaissant dans la réciproque assure l'existence d'au moins un zéro commun de  $A$  et  $B$ . Il est important de noter que le nombre de zéros commun dans ce cas peut éventuellement être infini s'il existe un facteur commun à  $A$  et  $B$  (s'il n'y a pas de facteur commun alors d'après le théorème de Bezout sur les courbes algébriques [Fulton, 1969], le nombre de zéros commun est donc fini, borné par  $\deg(A)\deg(B)$ ). Si le facteur commun de  $A$  et  $B$  est de la forme  $\prod_j (z_1 - \alpha_j)$  (respectivement  $\prod_j (z_2 - \beta_j)$ ),

alors le résultant de  $A$  et  $B$  en  $z_2$  (respectivement en  $z_1$ ) est le polynôme nul et le résultant en  $z_1$  (respectivement en  $z_2$ ) sera non identiquement nul. On peut alors caractériser les zéros communs sous forme d'hyperplans de  $(\mathbb{C}^*)^2$ .

Si le facteur commun de  $A$  et  $B$  n'est pas d'une des formes précisées ci-dessus, alors les deux résultants en  $z_1$  et  $z_2$  sont identiquement nuls. Dans ce cas, on ne peut pas déterminer explicitement à l'aide des résultants l'ensemble des zéros communs de  $A$  et  $B$ .

**Remarque 5.3** *Il existe également des travaux sur les résultants multivariés pour la recherche d'intersections de courbes algébriques [Macaulay, 1902; Jönsson, Vavasis, 2005]. À notre connaissance, les méthodes existantes présupposent, comme pour le théorème de Bézout sur les courbes algébriques, que les deux polynômes considérés sont premiers entre eux. Cette hypothèse n'a aucune raison d'être vérifiée en général dans le cas qui nous intéresse.*

## 5.2.2 Description de l'algorithme

Des deux résultats précédents, nous allons maintenant déduire un algorithme permettant de caractériser l'inversibilité d'une matrice  $\mathbf{H}[z_1, z_2]$  dans la plupart des cas.

### 5.2.2-a) Principe

D'après le théorème 4, pour caractériser l'inversibilité de la matrice  $\mathbf{H}[z_1, z_2]$  il faut (et il suffit de) vérifier que les déterminants des sous-matrices maximales n'ont pas de racines communes dans  $(\mathbb{C}^*)^2$ . Or, le théorème 6 nous fournit un moyen pratique pour déterminer les racines communes de deux polynômes de deux variables.

Voyons le principe de cette recherche. Soient deux polynômes  $A$  et  $B$  de  $\mathbb{C}[z_1, z_2]$  et de degrés strictement positifs en  $z_1$  et  $z_2$ . Soient les résultants  $C_1(z_1) = \text{Res}_{z_2}(A, B)(z_1)$  et  $C_2(z_2) = \text{Res}_{z_1}(A, B)(z_2)$ . Notons  $S_1$  et  $S_2$  les ensembles des zéros de  $C_1$  et  $C_2$  respectivement. D'après le théorème 6, il est clair que l'ensemble des racines communes à  $A$  et  $B$  est inclus dans l'ensemble :  $S_1 \times S_2$ . Afin de déterminer les racines communes il suffit donc de tester tous les éléments de  $S_1 \times S_2$ .

Pour pouvoir utiliser cette méthode, il faut se placer dans le cadre d'application du théorème. Cela implique de traiter plusieurs cas particuliers : le cas où  $A$  ou  $B$  est "dégénéré", c'est-à-dire qu'il n'est pas de degré strictement positif en les deux variables et le cas où l'un des résultants est identiquement nul.

Le principe général de l'algorithme consiste donc à extraire des sous-matrices maximales, à calculer leur déterminant (formant ainsi des polynômes) puis à chercher les zéros communs des polynômes ainsi obtenus. L'algorithme est divisé en trois étapes :

- d'abord une initialisation des variables qui seront utilisées,
- ensuite la boucle principale dans laquelle les sous-matrices maximales sont extraites et traitées
- et enfin une étape de conclusion.

Voyons maintenant plus en détails la structure du cœur de la méthode : la deuxième étape.

### 5.2.2-b) Structure de la boucle principale

La boucle principale de l'algorithme peut être subdivisée en trois parties. Une première partie permet de vérifier que le déterminant de la matrice maximale extraite, donc le polynôme qui sera considéré à cette itération, n'est pas dégénéré et de traiter les cas particuliers. Si le polynôme n'est pas dégénéré, on peut continuer sinon on passe à l'itération suivante. Si  $\mathcal{S}$ , l'ensemble des zéros communs des polynômes extraits jusqu'à cette itération, n'a pas été réduit ( $\mathcal{S} = (\mathbb{C}^*)^2$ ) et qu'au moins un polynôme non dégénéré a déjà été extrait dans les itérations précédentes, on passe à la deuxième partie, dans laquelle on essaye de réduire l'ensemble  $\mathcal{S}$  à l'aide de calculs de résultants. Enfin, si l'ensemble des zéros communs a déjà été réduit ( $\mathcal{S} \neq (\mathbb{C}^*)^2$ ) alors on passe à la troisième partie permettant de réduire à nouveau l'ensemble  $\mathcal{S}$ . Voyons maintenant plus en détails ces trois parties ; les numéros des points correspondent à la numérotation utilisée dans l'annexe A.

#### Première partie

Le théorème 6 permet, comme nous l'avons vu, de faire le lien entre la notion de résultant et la recherche de zéros communs de deux polynômes de deux variables. Pour que ce théorème soit utilisable, il faut que les polynômes soient non dégénérés, la première partie consiste donc à vérifier que le polynôme considéré  $A_k(z_1, z_2)$  n'est pas dégénéré. Si le polynôme est constant non-nul il est clair que l'ensemble des zéros communs sera vide, en revanche, si la constante est nulle on ne peut rien conclure et il faut poursuivre en extrayant une nouvelle sous-matrice.

Si le polynôme extrait est d'une seule variable, il suffit alors d'analyser les zéros d'un polynôme d'une seule variable ce qui permet de réduire l'ensemble des zéros communs potentiels à une union d'hyperplans de  $(\mathbb{C}^*)^2$ .

#### Deuxième partie

La deuxième partie est la plus complexe. À ce stade du déroulement de l'algorithme, on est assuré que le polynôme  $A_k$  n'est pas dégénéré. Si ce polynôme est le premier polynôme non dégénéré rencontré pendant l'exécution de l'algorithme, on le place dans une file d'attente  $\mathcal{F}$  et on extrait une nouvelle sous-matrice maximale. S'il y a déjà des polynômes dans  $\mathcal{F}$ , alors notons  $A_{k_m}$  l'un de ces polynômes. On calcule les résultants :  $C_1(z_1) = \text{Res}_{z_2}(A_k, A_{k_m})(z_1)$  et  $C_2(z_2) = \text{Res}_{z_1}(A_k, A_{k_m})(z_2)$  de  $A_k$  et  $A_{k_m}$ , il nous faut alors considérer plusieurs cas.

- Si  $C_1$  et  $C_2$  sont des polynômes non nuls (cas (d)) alors, d'après le théorème 6, en étudiant les zéros de ces polynômes on a accès aux zéros communs de  $A_k$  et  $A_{k_m}$ . Cette étape constitue en fait le cas le plus fréquent pour les deux polynômes  $A_k$  et  $A_{k_0}$ .
- Si  $C_1$  ou  $C_2$  est nul (cas (b) et (c)), on peut étudier les racines du résultant non nul et ensuite travailler directement sur les polynômes  $A_k$  et  $A_{k_m}$  qui, appliqués en les racines du résultant, deviennent des polynômes d'une variable.
- Enfin, il reste le cas où  $C_1$  et  $C_2$  sont nuls (cas (a)). Dans ce cas on ne peut rien conclure et il faut, soit tester le polynôme  $A_k$  avec un autre polynôme de  $\mathcal{F}$  (qui n'a pas encore été testé avec  $A_k$ ), soit ajouter  $A_k$  à la file  $\mathcal{F}$  et retourner au début de la boucle principale pour extraire une nouvelle sous-matrice.

Si  $C_1$  ou  $C_2$  est un polynôme non nul, on a donc été en mesure de réduire l'ensemble  $\mathcal{S}$  des zéros communs, on peut alors appliquer la troisième partie avec les polynômes (éventuellement) stockés dans  $\mathcal{F}$  (étape (f)).

### Troisième partie

La troisième partie de l'algorithme (qui a été placée sous forme de sous-fonction dans l'annexe) traite les cas où l'ensemble  $\mathcal{S}$  des zéros communs potentiels a déjà pu être réduit. Le cas où  $\mathcal{S}$  est de cardinal fini (cas 1 de la fonction *reduire* de l'annexe A) revient à tester si les éléments de  $\mathcal{S}$  sont des racines du polynôme  $A_k$ .

Si  $\mathcal{S}$  est une union finie d'hyperplans (cas 2 et 3 de la fonction *reduire*) de  $(\mathbb{C}^*)^2$  (i.e.  $\mathcal{S} = \bigcup_j \{a_j\} \times \mathbb{C}^*$  ou  $\mathcal{S} = \mathbb{C}^* \times \bigcup_j \{b_j\}$ ), on recherche les intersections entre la courbe algébrique définie par  $A_k(z_1, z_2) = 0$  et ces hyperplans, ce qui conduit (par le théorème de Bézout sur les courbes algébriques [Fulton, 1969]) à un nombre fini de points d'intersections si l'ensemble des zéros de  $A_k$  ne contient aucun des hyperplans (autrement dit  $A_k$  est premier avec chaque polynôme  $z_1 - a_j$  ou  $z_2 - b_j$ ) ou à une union finie d'hyperplan et (éventuellement) de points d'intersection.

Une version détaillée de l'algorithme est présentée dans l'annexe A.

### 5.2.2-c) Discussion

L'algorithme ne peut conclure à l'inversibilité ou non de  $\mathbf{H}[z_1, z_2]$ , uniquement dans un cas particulier : celui où l'ensemble  $\mathcal{S}$  ne peut jamais être réduit pendant le déroulement de l'algorithme (i.e.  $\mathcal{S} = (\mathbb{C}^*)^2$  au terme de l'exécution). D'après la discussion sur le théorème 6 à la fin de la section 5.2.1, ce cas de figure ne peut arriver que si tous les polynômes extraits, soit sont dégénérés et nuls, soit présentent deux à deux un facteur commun qui n'est ni de la forme  $\prod_j (z_1 - \alpha_j)$  (cas géré par  $C_1 = 0$  et  $C_2 \neq 0$ ) ni de la forme  $\prod_j (z_2 - \beta_j)$  (cas géré par  $C_1 \neq 0$  et  $C_2 = 0$ ). Notons que si tous les polynômes extraits sont nuls, alors  $\mathcal{F} = \emptyset$  et  $\mathcal{S} = (\mathbb{C}^*)^2$ , il faut donc qu'au moins un polynôme non dégénéré soit extrait.

Au-delà de cette limitation, un autre problème de cet algorithme est son temps de calcul. Dans le pire des cas il faut extraire  $\binom{M^2}{N^2}$  sous-matrices (où  $M$  et  $N$  sont définis comme à la section 5.1.1), calculer leur déterminant, puis selon les cas calculer des résultants (calcul de déterminants de matrices polynomiales d'une seule variable) et enfin faire des recherches de racines de polynômes pouvant avoir des degrés assez élevés.

**Remarque 5.4** Les calculs de déterminants de matrices polynomiales d'une variable peuvent être remplacés par des calculs portant uniquement sur les "séquences" de coefficients de ces polynômes. Soient deux polynômes  $P(X) = \sum_{k=0}^m a_k X^k \in \mathbb{F}[X]$  et  $Q(X) = \sum_{k=0}^n b_k X^k \in \mathbb{F}[X]$ , on définit les séquences de coefficients de ces polynômes par  $S_P = (a_m, \dots, a_0) \in \mathbb{F}^{m+1}$  et  $S_Q = (b_n, \dots, b_0) \in \mathbb{F}^{n+1}$ . La séquence de coefficients du polynôme  $PQ$  est alors donnée par la formule :

$$S_{PQ} = (c_{m+n+1}, \dots, c_0), \quad \text{où} \quad c_k = \sum_{i=\max(0, k-n)}^{\min(m, k)} a_i b_{k-i},$$

autrement dit  $S_{PQ} = S_P * S_Q$ , où  $*$  représente le produit de convolution. On peut ainsi n'effectuer que des opérations "numériques" lors du calcul du déterminant.

Le calcul de déterminants pour des matrices dont les éléments appartiennent à un corps peut se réaliser à l'aide de l'algorithme du pivot de Gauss, conduisant ainsi à un nombre d'opérations de l'ordre de  $\mathcal{O}(n^3)$  pour une matrice de taille  $n \times n$ . Cette méthode, utilisant des divisions, n'est malheureusement pas applicable ici puisque nous nous plaçons dans un anneau. Dans ce cas on peut toujours appliquer la formule de Leibniz, mais sa complexité, de l'ordre de  $n!(n-1)$  (uniquement en comptant les multiplications), la rend inutilisable dans la pratique. Des méthodes de calculs du déterminants ne faisant pas intervenir de divisions ont été introduites depuis les années 1980 [Berkowitz, 1984; Mahajan, Vinay, 1997] et permettent d'obtenir le résultat avec une complexité  $\mathcal{O}(n^4)$ .

Un élément qui manque pour pouvoir conclure dans le cas où les polynômes  $C_1$  et  $C_2$  sont nuls est une forme de division euclidienne sur l'ensemble des polynômes de deux variables. Cette forme de division est introduite dans le cadre de la théorie des bases de Gröbner vers lesquelles nous nous sommes donc tournés et que nous allons brièvement présenter dans la section 5.3.

### 5.2.2-d) Exemples

Donnons quelques exemples de résultats de cet algorithme sur de petites matrices de taille  $9 \times 4$  (i.e.  $M = 3$  et  $N = 2$ ). La matrice :

$$\mathcal{H}[z_1, z_2] = \begin{pmatrix} 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 2z_2 z_1 & -2z_1 & -z_1 + z_2 & -2 - 2z_1 - z_2 + z_2 z_1 \\ -z_2 + z_2 z_1 & -2 + z_2 z_1 & -z_2 & -1 - z_1 - 4z_2 \\ -2z_1 - z_2 z_1 & z_1 + z_2 - z_2 z_1 & -z_2 - z_2 z_1 & 3 - z_2 + z_2 z_1 \\ -1 + z_1 - z_2 z_1 & 1 - 2z_1 & 1 + z_2 - 2z_2 z_1 & 1 + z_1 + z_2 z_1 \\ -z_1 - z_2 + z_2 z_1 & -2 + z_1 & 1 & 1 + z_1 - z_2 \\ 1 + z_1 + z_2 & -z_2 - 3z_2 z_1 & 2z_1 + z_2 + z_2 z_1 & 1 + 2z_1 - z_2 - 2z_2 z_1 \\ z_2 z_1 & -z_1 - z_2 - z_2 z_1 & z_1 + z_2 + z_2 z_1 & z_1 - z_2 + 2z_2 z_1 \end{pmatrix},$$

est le fruit d'un tirage aléatoire de polynômes à coefficients entiers de la forme  $a_0 + a_1 z_1 + a_2 z_2 + a_3 z_1 z_2$ . Après 5 itérations de la boucle principale de l'algorithme (présentée à la section 5.2.2-b)), on obtient que cette matrice est inversible.

On considère ensuite la matrice :

$$\mathcal{F}[z_1, z_2] = \begin{pmatrix} 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 2z_2 z_1 & -2z_1 & -z_1 + z_2 & -2 - 2z_1 - z_2 + z_2 z_1 \\ -z_2 + z_2 z_1 & -2 + z_2 z_1 & -z_2 & -1 - z_1 - 4z_2 \end{pmatrix},$$

construite à partir des 4 premières lignes de la matrice  $\mathcal{H}[z_1, z_2]$ . Le résultat de l'algorithme dans ce cas est qu'on ne peut pas conclure. En regardant de plus près, on constate que tous les déterminants de matrices extraites sont soit nuls soit les polynômes non dégénérés :

$$\begin{aligned} A_{\pm}(z_1, z_2) = \pm & (4 + 20z_2^2 z_1^4 - 34z_2^2 z_1^3 + 4z_1 + 20z_2 + 11z_2 z_1 - 15z_1^2 - 38z_2 z_1^2 + 6z_2^2 \\ & - 60z_2^2 z_1 - 49z_2^2 z_1^2 - 13z_2 z_1^3 - 2z_2^3 - 3z_1^3 - 5z_2^3 z_1 - 39z_2^3 z_1^2 \\ & + 14z_2^3 z_1^3 + 8z_2 z_1^4 + 16z_2^3 z_1^4 - 12z_2^4 z_1^2 + 15z_2^4 z_1^3 - 7z_2^4 z_1 + 3z_2^4 z_1^4). \end{aligned}$$

Les polynômes  $C_1(z_1)$  et  $C_2(z_2)$ , obtenus par des calculs de résultants de la forme  $\text{Res}_{z_i}(A_{\pm}, A_{\pm})(z_{3-i})$  pour  $i \in \{1, 2\}$ , sont toujours nuls. L'ensemble  $\mathcal{S}$  vaut donc toujours  $(\mathbb{C}^*)^2$  à la fin de l'algorithme et l'ensemble  $\mathcal{F}$  est non vide (puisqu'on a trouvé des polynômes non dégénérés).

Enfin, on a construit la matrice :

$$\mathcal{G}[z_1, z_2] = \begin{pmatrix} 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 1 - z_2 & z_2 z_1 & 1 + z_1 + z_2 + z_2 z_1 & 2z_1 + z_2 \\ -2 & -z_1 - z_2 z_1 & -1 - z_1 + z_2 - 2z_2 z_1 & 1 + 3z_1 + z_2 z_1 \\ 2z_2 z_1 & -2z_1 & -z_1 + z_2 & -2 - 2z_1 - z_2 + z_2 z_1 \end{pmatrix},$$

à partir des 3 premières lignes de  $\mathcal{H}[z_1, z_2]$ . Dans ce cas tous les polynômes extraits sont nuls, l'algorithme retourne alors  $\mathcal{S} = (\mathbb{C}^*)^2$  et  $\mathcal{F} = \emptyset$ . La matrice n'est donc pas inversible.

### 5.3 Méthode utilisant les bases de Gröbner

Les bases de Gröbner sont des outils algébriques qui permettent de vérifier l'inversibilité de matrices polynomiales. Les bases de Gröbner furent introduites pour la première



fois en 1965 dans la thèse de B. Buchberger [Buchberger, 1965], étudiant de W. Gröbner. Nous allons maintenant présenter brièvement ces méthodes qui ont été utilisées pour la construction de bases d'ondelettes non séparables [Faugère *et al.*, 1998] et dans le cadre de l'inversion de bancs de filtres multidimensionnels par H. Park et M. Vetterli [Park *et al.*, 1997] et plus récemment par J. Zhou et M. N. Do [Zhou, Do, 2005].

### 5.3.1 Quelques rappels sur les bases de Gröbner

Nous adoptons ici des notations similaires à celles du livre [Cox *et al.*, 1998]. Supposons que l'on ait défini un ordre sur l'ensemble  $\mathcal{T}$  des monômes de  $\mathbb{C}[\mathbf{z}]$ . L'ordre lexicographique des puissances est généralement utilisé. Soient  $\alpha = (\alpha_j)_{1 \leq j \leq L}$  et  $\beta = (\beta_j)_{1 \leq j \leq L}$  deux multi-indices de  $\mathbb{N}^L$ , on définit l'ordre lexicographique par :  $\alpha \succ \beta$  s'il existe un indice  $i \in \{1, \dots, L\}$  tel que pour tout  $1 \leq j < i$ ,  $\alpha_j = \beta_j$  et  $\alpha_i > \beta_i$ . Par exemple, avec  $L = 3$ , on a  $(1, 2, 1) \succ (1, 1, 5)$ .

On peut alors définir un ordre sur les monômes de la manière suivante : soient  $\mathbf{z}^\alpha$  et  $\mathbf{z}^\beta$  deux monômes, on dira que  $\mathbf{z}^\alpha$  domine  $\mathbf{z}^\beta$  si  $\alpha \succ \beta$ . Par exemple, avec  $L = 3$ , on a :  $z_1 z_2^2 z_3$  domine  $z_1 z_2 z_3^5$  (puisque  $(1, 2, 1) \succ (1, 1, 5)$ ).

Soit un polynôme  $P(\mathbf{z}) \in \mathbb{C}[\mathbf{z}]$ , on appelle *monôme dominant* de  $P$  le monôme le plus grand pour l'ordre que l'on s'est donné. On le note  $\text{lt}(P)$ .

Pour un idéal  $I \subset \mathbb{C}[\mathbf{z}]$ , on notera

$$\text{lt}(I) = \{\text{lt}(P) \mid P \in I\},$$

et pour une collection de polynômes :  $P_i \in \mathbb{C}[\mathbf{z}]$ , on note  $\langle P_1, \dots, P_n \rangle$  l'idéal engendré par ces polynômes.

**Définition 5.4** Une famille  $G = \{P_1, \dots, P_n\} \subset I$  est appelée une base de Gröbner de  $I$  si

$$\langle \text{lt}(P_1), \dots, \text{lt}(P_n) \rangle = \text{lt}(I);$$

c'est-à-dire si l'idéal engendré par les  $\text{lt}(P_i)$  coïncide avec  $\text{lt}(I)$ .

Si de plus  $G$  vérifie :

- pour tout  $P \in G$ , le coefficient du monôme  $\text{lt}(P)$  dans  $P$  est 1,
- pour tout  $P \in G$  aucun monôme de  $P$  n'appartient à  $\langle \text{lt}(G - \{P\}) \rangle$ ,

on dit alors que  $G$  est une base de Gröbner réduite.

**Théorème 8** Pour un ordre fixé sur les monômes, il existe une unique base de Gröbner réduite associée à un idéal donné de polynômes.

En général, pour un idéal de polynômes  $\mathcal{I}$  il n'y a pas unicité de la base de Gröbner, le théorème précédent assurant l'unicité de la base de Gröbner réduite, il est donc plus commode de travailler avec une base réduite.

### 5.3.2 Application au problème d'inversibilité

À l'aide de ces définitions, revenons au problème d'inversion. Comme précédemment, nous devons prouver l'existence d'une matrice polynomiale vérifiant :  $\tilde{\mathbf{H}}[\mathbf{z}] \mathbf{H}[\mathbf{z}] = \mathbf{I}$ . Le

théorème suivant [Zhou, Do, 2005] permet de faire le lien entre les bases de Gröbner réduites et l'inversibilité :

**Théorème 9** Soit  $\mathbf{H}[\mathbf{z}] \in \mathbb{C}[\mathbf{z}, \mathbf{z}^{-1}]^{M^2 \times N^2}$  une matrice polynomiale avec  $M > N$ . Les conditions suivantes sont alors équivalentes :

1.  $\mathbf{H}[\mathbf{z}]$  est telle que la base de Gröbner réduite de l'ensemble de ses déterminants mineurs maximaux et du polynôme  $1 - z_1 \dots z_{L+1}$  (où  $z_{L+1}$  est une nouvelle variable) est  $\{1\}$ ,
2.  $\mathbf{H}[\mathbf{z}]$  est inversible à gauche au sens où il existe une matrice polynomiale  $\tilde{\mathbf{H}}[\mathbf{z}] \in \mathbb{C}[\mathbf{z}, \mathbf{z}^{-1}]^{N \times M}$  telle que  $\tilde{\mathbf{H}}[\mathbf{z}] \mathbf{H}[\mathbf{z}] = \mathbf{I}$ .

Pour déterminer l'inversibilité de la matrice  $\mathbf{H}[\mathbf{z}]$ , il suffit donc de calculer les déterminants de toutes ses sous-matrices de taille maximale puis de rechercher la base de Gröbner réduite de cette famille de polynômes.

L'algorithme de Buchberger permet de calculer dans la pratique une base de Gröbner, à partir de laquelle on peut alors calculer la base réduite de Gröbner [Cox *et al.*, 1998]. Des logiciels de calcul formel comme Singular<sup>2</sup> présentent toutes les fonctions nécessaires pour effectuer ces calculs. Des algorithmes comme  $F_5$  proposé par Faugère [Faugère, 2002] permettent de réduire la complexité du calcul d'une base de Gröbner, cependant celle-ci demeure polynomiale.

**Remarque 5.5** L'utilisation de l'algorithme de Buchberger permet non seulement de calculer la base de Gröbner d'une famille de polynômes, mais également une matrice polynomiale de passage transformant les polynômes de la famille d'origine en ceux de la base calculée. Formellement, soit une famille de  $K_2$  polynômes  $(P_k)_{1 \leq k \leq K_2} \in \mathbb{C}[\mathbf{z}]^{K_2}$  et soit  $(B_j)_{1 \leq j \leq K_1}$  la base de Gröbner associée à cette famille, alors il existe des polynômes  $W_{j,k} \in \mathbb{C}[\mathbf{z}]$  pour  $j \in \{1, \dots, K_1\}$  et  $k \in \{1, \dots, K_2\}$ , tels que :

$$\forall j \in \{1, \dots, K_1\}, \quad B_j(\mathbf{z}) = \sum_{k=1}^{K_2} W_{j,k}(\mathbf{z}) P_k(\mathbf{z}).$$

Dans le cas où la base réduite est  $\{1\}$ , il est possible de construire une matrice inverse de  $\mathbf{H}[\mathbf{z}]$  à partir de ces polynômes  $W_{j,k}$ . La vérification de l'inversibilité avec la méthode par bases de Gröbner peut donc être directement couplée au calcul explicite d'un inverse.

### 5.3.3 Discussion

La méthode utilisant les bases de Gröbner présente trois principaux avantages :

- Elle permet de donner une condition nécessaire et suffisante de l'inversibilité d'une matrice polynomiale, et donc évite le risque de ne pas pouvoir conclure, encouru par la méthode par résultant.
- Elle est directement utilisable pour tester l'inversibilité de bancs de filtres de dimension supérieure à deux.
- Elle permet de calculer un inverse explicite.

2. Logiciel libre disponible sur le site : <http://www.singular.uni-kl.de/>

Dans la pratique, cette méthode est également difficile à utiliser pour des matrices de grandes tailles, puisqu'elle implique le calcul des déterminants des  $\binom{M^2}{N^2}$  sous-matrices maximales. L'algorithme par résultant proposé dans la section précédente extrait de manière itérative les sous-matrices et s'arrête dès que l'ensemble des zéros communs potentiels est vide dans  $(\mathbb{C}^*)^2$ . Dans le meilleur des cas, il suffit de calculer les déterminants de deux sous-matrices qui auraient tous leurs zéros communs dans  $\mathbb{C} \times \{0\} \cup \{0\} \times \mathbb{C}$ . En fait, en prenant des matrices polynomiales aléatoires le test d'inversibilité se termine en général après 3 itérations : les deux premières itérations permettant de réduire l'ensemble des zéros communs potentiels à un ensemble fini sur lequel le polynôme généré à la troisième itération n'est pas nul. Avec  $M = 3$  et  $N = 2$ , on a  $\binom{M^2}{N^2} = 126$  déterminants de sous-matrices à calculer avec la méthode par base de Gröbner. Ceci montre bien l'intérêt pratique de la méthode de la section précédente dans le cas où la matrice testée est effectivement inversible.

## 5.4 Calcul d'un inverse

L'algorithme proposé à la section 5.2.2 permet de déterminer si une matrice polyphase  $\mathbf{H}[\mathbf{z}]$  est inversible mais, contrairement à la méthode utilisant les bases de Gröbner, ne fournit pas d'inverse explicite. Nous allons maintenant voir comment calculer une matrice polyphase de synthèse dans le cas où l'on sait que la matrice d'analyse est inversible.

### 5.4.1 Méthode de calcul par pseudo-inverse

Dans cette partie, nous allons supposer que le banc de filtres d'analyse caractérisé par la matrice  $\mathbf{H}[\mathbf{z}]$  est inversible. Le but ici est de construire une matrice  $\tilde{\mathbf{H}}[\mathbf{z}]$  telle que  $\tilde{\mathbf{H}}[\mathbf{z}]\mathbf{H}[\mathbf{z}] = \mathbf{I}_{N^2}$ . En partant de cette expression et en utilisant les équations (5.1) et (5.2), on obtient :

$$\mathbf{I}_{N^2} = \sum_{\ell_1=-p_1}^{k+q_1-1} \sum_{\ell_2=-p_2}^{k+q_2-1} \mathbf{U}(\ell_1, \ell_2) z_1^{-\ell_1} z_2^{-\ell_2},$$

où

$$\mathbf{U}(\ell_1, \ell_2) = \sum_{s_1=\max(\ell_1-k+1, -p_1)}^{\min(q_1, \ell_1)} \sum_{s_2=\max(\ell_2-k+1, -p_2)}^{\min(q_2, \ell_2)} \tilde{\mathbf{H}}(s_1, s_2) \mathbf{H}(\ell_1 - s_1, \ell_2 - s_2). \quad (5.4)$$

La condition de reconstruction parfaite est donc équivalente à  $\mathbf{U}(\ell_1, \ell_2) = \delta_{\ell_1} \delta_{\ell_2} \mathbf{I}_{N^2}$ , pour tous  $\ell_1 \in \{-p_1, \dots, k+q_1-1\}$  et  $\ell_2 \in \{-p_2, \dots, k+q_2-1\}$ .

On peut également réécrire le problème sous la forme d'un système linéaire de  $(p_1 +$

$k + q_1)(p_2 + k + q_2)$  équations matricielles à résoudre :

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \mathbf{U}(-p_1, -p_2) = \tilde{\mathbf{H}}(-p_1, -p_2)\mathbf{H}(0, 0) \\ \vdots \\ \mathbf{U}(-p_1, \ell_2) = \sum_{s_2=\max(\ell_2-k+1, -p_2)}^{\min(q_2, \ell_2)} \tilde{\mathbf{H}}(-p_1, s_2)\mathbf{H}(0, \ell_2 - s_2) \\ \vdots \\ \mathbf{U}(-p_1, k + q_2 - 1) = \tilde{\mathbf{H}}(-p_1, q_2)\mathbf{H}(0, k - 1) \\ \vdots \end{array} \right. \\ \left\{ \begin{array}{l} \mathbf{U}(k + q_1 - 1, -p_2) = \tilde{\mathbf{H}}(q_1, -p_2)\mathbf{H}(k - 1, 0) \\ \vdots \\ \mathbf{U}(k + q_1 - 1, \ell_2) = \sum_{s_2=\max(\ell_2-k+1, -p_2)}^{\min(q_2, \ell_2)} \tilde{\mathbf{H}}(q_1, s_2)\mathbf{H}(k - 1, \ell_2 - s_2) \\ \vdots \\ \mathbf{U}(k + q_1 - 1, k + q_2 - 1) = \tilde{\mathbf{H}}(q_1, q_2)\mathbf{H}(k - 1, k - 1). \end{array} \right. \end{array} \right.$$

Ce système peut, à son tour, se réécrire sous forme matricielle :

$$\mathcal{H}\tilde{\mathcal{H}} = \mathcal{U} \quad (5.5)$$

avec :

$$\begin{aligned} \tilde{\mathcal{H}}^\top &= [\tilde{\mathbf{H}}(-p_1, -p_2), \dots, \tilde{\mathbf{H}}(-p_1, q_2), \dots, \tilde{\mathbf{H}}(q_1, -p_2), \dots, \tilde{\mathbf{H}}(q_1, q_2)] \\ &\in \mathbb{C}^{N^2 \times M^2(p_1+q_1+1)(p_2+q_2+1)}, \\ \mathcal{U}^\top &= [\mathbf{0}_{N^2, ((p_2+q_2+k)p_1+p_2)N^2} \mathbf{I}_{N^2} \mathbf{0}_{N^2, ((p_2+q_2+k)(q_1+k-1)+k+q_2-1)N^2}] \\ &\in \mathbb{C}^{N^2 \times N^2(p_1+q_1+k)(p_2+q_2+k)}, \\ \mathcal{H}^\top &= [\mathbf{B}(i, j)]_{1 \leq i \leq (p_1+q_1+1)(p_2+q_2+1), 1 \leq j \leq (p_1+q_1+k)(p_2+q_2+k)} \\ &\in \mathbb{C}^{M^2(p_1+q_1+1)(p_2+q_2+1) \times N^2(p_1+q_1+k)(p_2+q_2+k)}. \end{aligned}$$

Dans la formulation précédente, les matrices  $\mathbf{B}(i, j)$  sont définies par :

$$\begin{aligned} \forall i \in \{1, \dots, (p_1 + q_1 + 1)(p_2 + q_2 + 1)\} \quad \text{et} \quad \forall j \in \{1, \dots, (p_1 + q_1 + k)(p_2 + q_2 + k)\}, \\ \mathbf{B}(i, j) = \begin{cases} \mathbf{H}(j_1 - i_1, j_2 - i_2), & \text{si } (j_1 - i_1, j_2 - i_2) \in \{0, \dots, k - 1\}^2 \\ \mathbf{0}_{M^2 \times N^2}, & \text{sinon} \end{cases} \end{aligned}$$

où  $i = i_1(p_2 + q_2 + 1) + i_2$  avec  $(i_1, i_2) \in \{0, \dots, p_1 + q_1\} \times \{0, \dots, p_2 + q_2\}$  et  $j = j_1(p_2 + q_2 + k) + j_2$  avec  $(j_1, j_2) \in \{0, \dots, p_1 + q_1 + k - 1\} \times \{0, \dots, p_2 + q_2 + k - 1\}$ .

**Exemple 7** Avec  $k = 2, p_1 = p_2 = 1$  et  $q_1 = q_2 = 0$ , le système ci-dessus s'écrit :

$$\begin{aligned} \tilde{\mathcal{H}}^\top &= [\tilde{\mathbf{H}}(-1, -1), \tilde{\mathbf{H}}(-1, 0), \tilde{\mathbf{H}}(0, -1), \tilde{\mathbf{H}}(0, 0)], \\ \mathcal{U}^\top &= [\mathbf{0}_{N^2, 4N^2} \mathbf{I}_{N^2} \mathbf{0}_{N^2, 4N^2}], \end{aligned}$$

et

$$\mathcal{H}^\top = \begin{pmatrix} H(0,0) & H(0,1) & \mathbf{0} & H(1,0) & H(1,1) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & H(0,0) & H(0,1) & \mathbf{0} & H(1,0) & H(1,1) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & H(0,0) & H(0,1) & \mathbf{0} & H(1,0) & H(1,1) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & H(0,0) & H(0,1) & \mathbf{0} & H(1,0) & H(1,1) \end{pmatrix}.$$

De la même manière que dans le cas unidimensionnel, les valeurs de  $p_1, p_2, q_1$  et  $q_2$  sont inconnues. Pour les déterminer on tente de résoudre le système linéaire 5.5, pour différentes valeurs de ces paramètres. Afin d'obtenir des filtres de supports minimaux, on peut faire varier ces paramètres de la façon suivante : on part de  $\gamma = p_1 + q_1 + p_2 + q_2 = 0$  et on fait augmenter  $\gamma$  en testant pour chaque valeur de  $\gamma$  toutes les combinaisons de paramètres  $(p_1, p_2, q_1, q_2) \in \mathbb{N}^4$  vérifiant l'égalité  $\gamma = p_1 + q_1 + p_2 + q_2$ .

### 5.4.2 Cas symétrique

Comme nous l'avons vu à la section 3.4, il est intéressant, notamment pour des applications impliquant un seuillage de coefficients complexes dans le domaine fréquentiel, d'avoir une méthode permettant d'assurer que la reconstruction des données après traitement soit réelle. Dans le cas unidimensionnel nous nous étions intéressé à la symétrie hermitienne, voyons comment étendre dans le cadre multidimensionnel l'une des constructions que nous avons proposées.

Nous supposons ici que le résultat de la décomposition de données réelles par le banc de filtres d'analyse présente une symétrie hermitienne. Autrement dit, pour toute image réelle  $(x(\mathbf{n}))_{\mathbf{n} \in \mathbb{Z}^2}$ , et pour tout couple  $(\mathbf{j}_f, \mathbf{n}_f) \in \{0, \dots, M-1\}^2 \times \mathbb{Z}^2$ , on souhaite avoir en sortie du banc de filtres d'analyse :

$$y_{\mathbf{j}_f}(\mathbf{n}_f) = \overline{y_{(M-1, M-1) - \mathbf{j}_f}(\mathbf{n}_f)}.$$

Cette condition se réécrit sous la forme :

$$\sum_{\mathbf{m}} h_{\mathbf{j}_f}(\mathbf{m}) x(N\mathbf{n}_f - \mathbf{m}) = \sum_{\mathbf{m}} \overline{h_{(M-1, M-1) - \mathbf{j}_f}(\mathbf{m})} x(N\mathbf{n}_f - \mathbf{m}).$$

Avec une image d'entrée identiquement nulle sauf en un point, on a alors :

$$h_{\mathbf{j}_f}(\mathbf{n}) = \overline{h_{(M-1, M-1) - \mathbf{j}_f}(\mathbf{n})},$$

et en utilisant l'équation (5.1), on obtient finalement la condition :

$$\mathbf{H}(\ell) = \mathbf{J}_{M^2} \overline{\mathbf{H}(\ell)}. \quad (5.6)$$

De la même manière pour assurer que la reconstruction de coefficients présentant une symétrie hermitienne soit réelle, on doit vérifier que tous coefficients  $(y_{\mathbf{i}}(\mathbf{n}))_{0 \leq i_1 < M, 0 \leq i_2 < M}$ , tels que  $y_{\mathbf{i}}(\mathbf{n}) = 0$  si  $(\mathbf{i}, \mathbf{n}) \neq (\mathbf{j}_f, \mathbf{n}_f)$  ou  $(\mathbf{i}, \mathbf{n}) \neq ((M-1, M-1) - \mathbf{j}_f, \mathbf{n}_f)$  avec  $\mathbf{j}_f \in \{0, \dots, M-1\}^2$  et  $\mathbf{n}_f \in \mathbb{Z}^2$ , vérifiant :  $y_{\mathbf{j}_f}(\mathbf{n}_f) = \overline{y_{(M-1, M-1) - \mathbf{j}_f}(\mathbf{n}_f)}$ , permet de reconstruire un signal réel.

L'image reconstruite s'écrit alors :

$$\begin{aligned}\tilde{x}(\mathbf{m}) &= \sum_{\mathbf{j} \in \{0, \dots, M-1\}^2} \sum_{\ell \in \mathbb{Z}^2} \tilde{h}_{\mathbf{j}}(\mathbf{m} - N\ell) y_{\mathbf{j}}(\ell) \\ &= \tilde{h}_{\mathbf{j}_f}(\mathbf{m} - \mathbf{n}_f N) y_{\mathbf{j}_f}(\mathbf{n}_f) + \overline{\tilde{h}_{(M-1, M-1) - \mathbf{j}_f}(\mathbf{m} - \mathbf{n}_f N) y_{\mathbf{j}_f}(\mathbf{n}_f)}.\end{aligned}$$

Une condition nécessaire et suffisante pour que l'image reconstruite  $\tilde{x}(\mathbf{m})$  appartienne à  $\mathbb{R}^2$ , est que  $\tilde{h}_{\mathbf{j}_f}(\mathbf{m} - \mathbf{n}_f N) = \overline{\tilde{h}_{(M-1, M-1) - \mathbf{j}_f}(\mathbf{m} - \mathbf{n}_f N)}$ . En faisant varier les couples  $(\mathbf{j}_f, \mathbf{n}_f)$ , on déduit que la condition sur les filtres de synthèse est que :

$$\forall \mathbf{j} \in \{0, \dots, M-1\}^2 \text{ et } \forall \mathbf{n} \in \mathbb{Z}^2, \quad \tilde{h}_{\mathbf{j}}(\mathbf{n}) = \overline{\tilde{h}_{(M-1, M-1) - \mathbf{j}}(\mathbf{n})}.$$

Grâce à l'équation (5.2), cette condition peut se réécrire :

$$\forall \ell \in \{-p_1, \dots, q_1\} \times \{-p_2, \dots, q_2\}, \quad \tilde{\mathbf{H}}(\ell) = \overline{\mathbf{H}(\ell)} \mathbf{J}_{M^2}. \quad (5.7)$$

En utilisant les conditions (5.7) et (5.6) dans (5.4), on obtient :

$$\begin{aligned}\mathbf{U}(\ell) &= \sum_{s_1 = \max(\ell_1 - k + 1, -p_1)}^{\min(q_1, \ell_1)} \sum_{s_2 = \max(\ell_2 - k + 1, -p_2)}^{\min(q_2, \ell_2)} \tilde{\mathbf{H}}(\mathbf{s}) \mathbf{J}_{M^2} \overline{\mathbf{H}(\ell - \mathbf{s})} \\ &= \sum_{s_1 = \max(\ell_1 - k + 1, -p_1)}^{\min(q_1, \ell_1)} \sum_{s_2 = \max(\ell_2 - k + 1, -p_2)}^{\min(q_2, \ell_2)} \overline{\tilde{\mathbf{H}}(\mathbf{s})} \mathbf{J}_{M^2} \mathbf{H}(\ell - \mathbf{s}).\end{aligned}$$

Ceci implique que si  $\tilde{\mathcal{H}} = \left[ \tilde{\mathbf{H}}(-p_1, -p_2), \dots, \tilde{\mathbf{H}}(-p_1, q_2), \dots, \tilde{\mathbf{H}}(q_1, -p_2), \dots, \tilde{\mathbf{H}}(q_1, q_2) \right]^\top$  est solution du système linéaire (5.5) alors, sous l'hypothèse de symétrie hermitienne du banc de filtres d'analyse, alors la matrice :

$$\tilde{\mathcal{H}}_2 = \left[ \overline{\tilde{\mathbf{H}}(-p_1, -p_2)} \mathbf{J}_{M^2}, \dots, \overline{\tilde{\mathbf{H}}(-p_1, q_2)} \mathbf{J}_{M^2}, \dots, \overline{\tilde{\mathbf{H}}(q_1, -p_2)} \mathbf{J}_{M^2}, \dots, \overline{\tilde{\mathbf{H}}(q_1, q_2)} \mathbf{J}_{M^2} \right]^\top$$

est aussi une solution du système linéaire. Finalement, on en déduit que la somme :  $\tilde{\mathcal{H}}_0 = \frac{1}{2}(\tilde{\mathcal{H}} + \tilde{\mathcal{H}}_2)$  est également une solution. De plus, elle vérifie par construction la condition (5.7).

### 5.4.3 Paramétrisation des bancs de filtres de synthèse

Dans la partie précédente, nous avons vu comment la recherche d'un inverse particulier peut se représenter sous forme d'un système linéaire. Un avantage de cette écriture est que l'on se ramène ainsi à un problème proche (aux dimensions des matrices près) de celui traité à la section 4.1. La méthode pour décrire une paramétrisation de l'ensemble des bancs de filtres de synthèse (pour un choix des paramètres  $(p_1, p_2, q_1, q_2) \in \mathbb{N}^4$ ) dans le cas multidimensionnel est donc similaire à celle proposée dans le cas unidimensionnel. Rappelons brièvement les différentes étapes de cette méthode.

À nouveau, nous allons d'abord considérer la SVD de la matrice  $\mathcal{H}$  définie à la section 5.4.1. Soit  $r$  le rang de cette matrice, la SVD s'écrit :

$$\mathcal{H} = \mathcal{U}_0 \Sigma_0 \mathcal{V}_0^*,$$

où  $\Sigma_0 \in \mathbb{C}^{r \times r}$  est une matrice diagonale inversible,  $\mathcal{U}_0 \in \mathbb{C}^{N^2(p_1+q_1+k)(p_2+q_2+k) \times r}$  et  $\mathcal{V}_0 \in \mathbb{C}^{M^2(p_1+q_1+1)(p_2+q_2+1) \times r}$  sont des matrices semi-unitaires.

On considère alors une solution particulière du système :  $\tilde{\mathcal{H}}^0 = \mathcal{V}_0 \Sigma_0^{-1} \mathcal{U}_0^* \mathcal{U}$ . On peut en déduire que :

$$\mathcal{V}_0^* (\tilde{\mathcal{H}} - \tilde{\mathcal{H}}^0) = \mathbf{0}_{r \times N^2},$$

puis, par un raisonnement identique à celui présenté à la section 4.1 :

$$\tilde{\mathcal{H}} = \mathcal{V}_1 \mathcal{C} + \tilde{\mathcal{H}}^0 \quad (5.8)$$

où  $\mathcal{C} \in \mathbb{C}^{M^2(p_1+q_1+1)(p_2+q_2+1)-r \times N^2}$  et  $\mathcal{V}_1$  est la matrice telle que la matrice bloc  $(\mathcal{V}_0, \mathcal{V}_1)$  soit unitaire.

La matrice  $\mathcal{C}$  nous permet donc de paramétrer l'ensemble des bancs de filtres de synthèse. Cette écriture réduisant le nombre de paramètres à considérer pour définir un banc de filtres de synthèse pourrait, comme dans le cas unidimensionnel, être utile pour une étape ultérieure d'optimisation.

#### 5.4.4 Exemple d'inversion

On considère dans cette section le banc de filtres d'analyse de réponses impulsionnelles :

$$h_{i_1, i_2}(p_1, p_2) = \frac{W(p_1, p_2)}{k'N} e^{\frac{-i2\pi}{k'N} \left( (i_1 - \frac{k'N-1}{2})(p_1 - \frac{kN-1}{2}) + (i_2 - \frac{k'N-1}{2})(p_2 - \frac{kN-1}{2}) \right)},$$

avec  $(i_1, i_2) \in \{0, \dots, k'N - 1\}^2$  et  $(p_1, p_2) \in \{0, \dots, kN - 1\}$ .  $W$  représente une fenêtre appliquée sur les données. Pour cet exemple nous avons utilisé la fenêtre  $W_c$  qui sera explicitée à la section 5.5.2. Enfin, nous avons choisi les paramètres suivant,  $k' = \frac{3}{2}$  pour la redondance,  $k = 3$  pour le recouvrement et  $N = 8$  pour le sous-échantillonnage. La figure 5.1 représente les parties réelle et imaginaire de ces réponses impulsionnelles.

Nous avons ensuite utilisé la méthode de la section précédente pour construire un banc de filtres de synthèse associé. Dans ce cas, il a suffi de choisir  $\gamma = 4$  et les paramètres :  $(p_1; p_2; q_1; q_2) = (2; 1; 0; 1)$ . Les parties réelle et imaginaire des réponses impulsionnelles du banc de filtres de synthèse ainsi obtenu sont présentées à la figure 5.2. On voit que l'énergie des réponses impulsionnelles des filtres de synthèse est très dispersée, une étape d'optimisation pourrait permettre d'améliorer ces filtres.

## 5.5 Cas où la redondance est égale au recouvrement

### 5.5.1 Expression exacte du banc de filtres de synthèse

Dans cette section, nous allons étudier un cas particulier dans lequel on est capable de donner une expression exacte du banc de filtres de synthèse : le cas où la redondance est égale au recouvrement. Les bancs de filtres d'analyse considérés ici sont

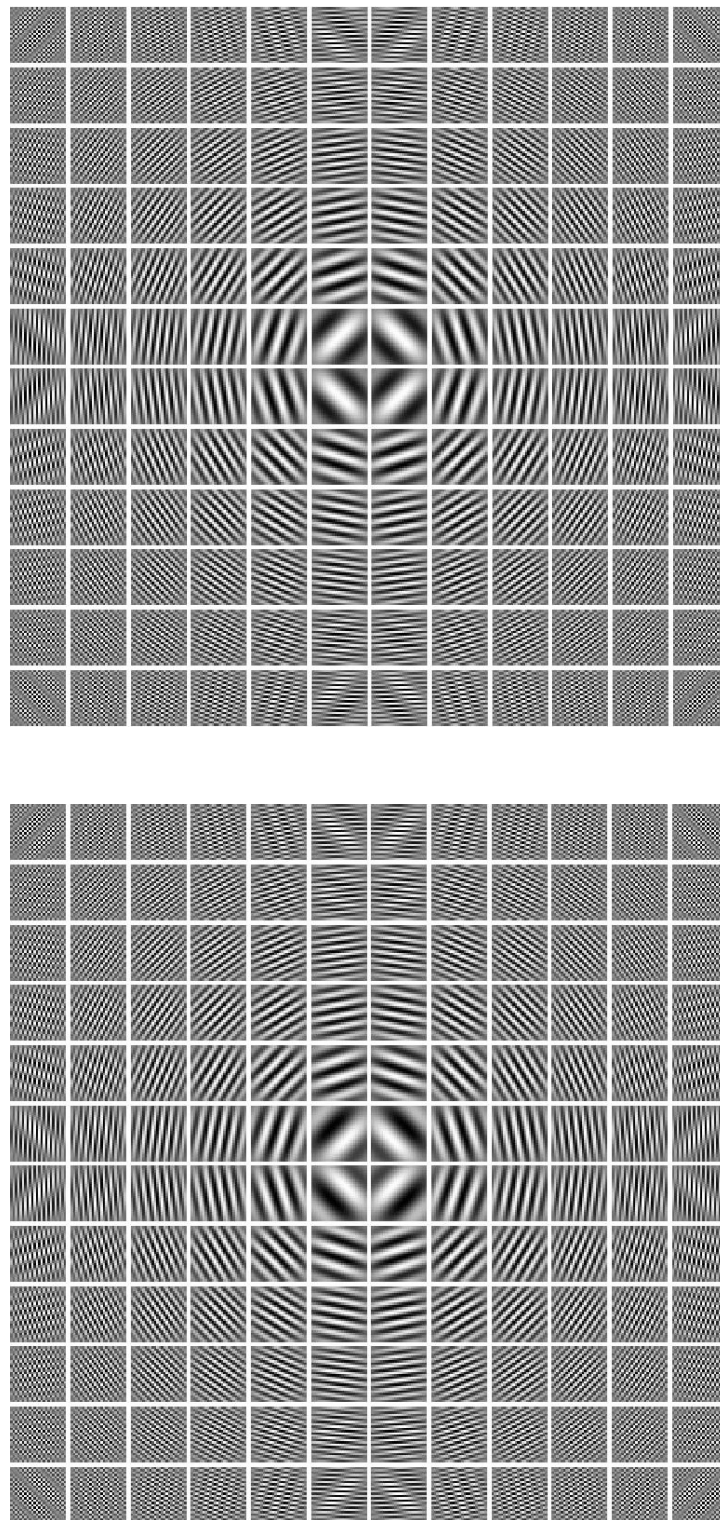


FIGURE 5.1 – Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des  $12 \times 12$  filtres d'analyse.



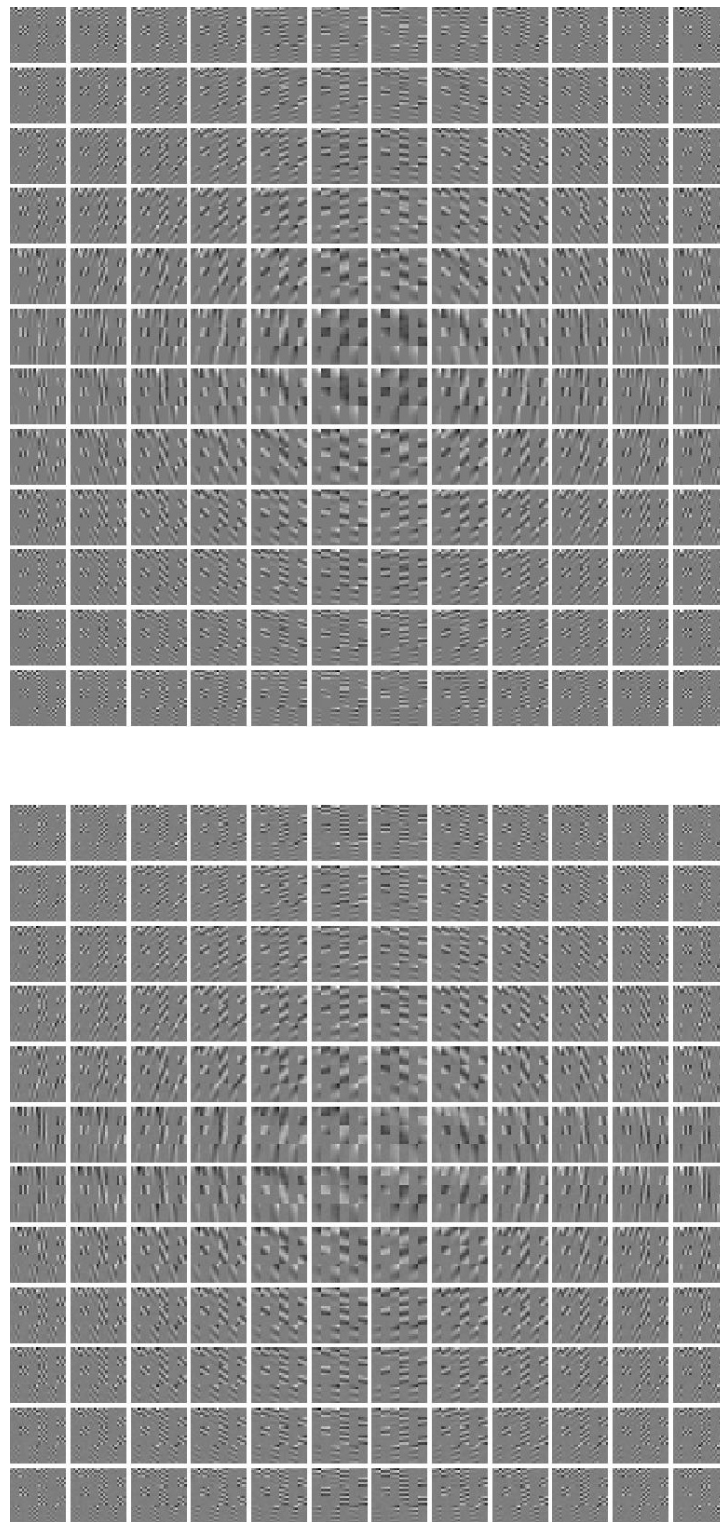


FIGURE 5.2 – Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des  $12 \times 12$  filtres de synthèse.

équivalents localement à l'application d'une fenêtre non-séparable sur les données suivie d'une transformation orthogonale séparable. Cette section est une réécriture avec le formalisme polyphase du résultat présenté dans [Gauthier *et al.*, 2005], qui avait été obtenu à l'aide de matrices infinies.

Nous allons considérer ici que la redondance est fixée, égale au facteur de recouvrement :  $k' = k$ . De plus, nous supposons que les coefficients des filtres d'analyse sont de la forme :  $\forall \mathbf{i} = (i_1, i_2) \in \{1, \dots, M\}^2$ ,  $\mathbf{j} = (j_1, j_2) \in \{1, \dots, N\}^2$ , et  $\ell = (\ell_1, \ell_2) \in \{0, \dots, k-1\}^2$ ,

$$h_{\mathbf{i}}(N\ell + \mathbf{j}) = E_{i_1, \ell_1 N + j_1} E_{i_2, \ell_2 N + j_2} W_{\ell_1 N + j_1, \ell_2 N + j_2},$$

où la matrice unitaire  $\mathbf{E} \in \mathbb{C}^{M \times M}$ , i.e.  $\mathbf{E}^* \mathbf{E} = \mathbf{I}_M$ , représente une transformation orthogonale suivant une direction et  $W \in \mathbb{C}^{M \times M}$  est une fenêtre d'analyse appliquée sur les données. En supposant la fenêtre  $W$  non séparable, la transformation bidimensionnelle considérée n'est pas non plus séparable.

Soient les matrices  $\mathbf{F}(\ell) \in \mathbb{C}^{M^2 \times N^2}$  définies par :  $\forall (i_1, i_2) \in \{1, \dots, M\}^2$ ,  $(j_1, j_2) \in \{1, \dots, N\}^2$ , et  $(\ell_1, \ell_2) \in \{0, \dots, k-1\}^2$ ,

$$\mathbf{F}_{i_1 + (i_2 - 1)M, j_1 + (j_2 - 1)N}(\ell_1, \ell_2) = \mathbf{E}_{i_1, \ell_1 N + j_1} \mathbf{E}_{i_2, \ell_2 N + j_2},$$

et les vecteurs  $\mathbf{w}(\ell) \in \mathbb{C}^{N^2}$  définis par :  $\forall (j_1, j_2) \in \{1, \dots, N\}^2$ ,  $(\ell_1, \ell_2) \in \{0, \dots, k-1\}^2$ ,

$$\mathbf{w}_{j_1 + (j_2 - 1)N}(\ell_1, \ell_2) = W_{\ell_1 N + j_1, \ell_2 N + j_2}.$$

On peut alors écrire la matrice polyphase du banc de filtres d'analyse sous la forme :

$$\mathbf{H}[\mathbf{z}] = \sum_{\ell \in \{0, \dots, k-1\}^2} \mathbf{F}(\ell) \text{Diag}(\mathbf{w}(\ell)) \mathbf{z}^{-\ell},$$

où  $\text{Diag}$  est l'opérateur transformant un vecteur en matrice diagonale.

En utilisant le fait que la matrice  $\mathbf{E}$  est unitaire, on a la propriété suivante sur les matrices  $\mathbf{F}(\ell)$  :

$$\begin{aligned} (\mathbf{F}^*(\ell) \mathbf{F}(\ell'))_{j_1 + (j_2 - 1)N, j'_1 + (j'_2 - 1)N} &= \sum_i \overline{\mathbf{F}_{i_1 + i_2 M, j_1 + j_2 N}(\ell_1, \ell_2)} \mathbf{F}_{i_1 + i_2 M, j'_1 + j'_2 N}(\ell'_1, \ell'_2) \\ &= \sum_{i_1} \overline{\mathbf{E}_{i_1, \ell_1 N + j_1} \mathbf{E}_{i_1, \ell'_1 N + j'_1}} \sum_{i_2} \overline{\mathbf{E}_{i_2, \ell_2 N + j_2} \mathbf{E}_{i_2, \ell'_2 N + j'_2}} \\ &= \delta_{\ell_1, \ell'_1} \delta_{j_1, j'_1} \delta_{\ell_2, \ell'_2} \delta_{j_2, j'_2}, \end{aligned}$$

donc  $\mathbf{F}^*(\ell) \mathbf{F}(\ell') = \delta_{\ell, \ell'} \mathbf{I}_{N^2}$ , pour tous  $\ell$  et  $\ell'$  dans  $\{0, \dots, k-1\}^2$ .

À l'aide de cette relation, étudions  $\mathbf{H}^*[\mathbf{z}^{-1}]\mathbf{H}[\mathbf{z}]$  :

$$\begin{aligned}
 \mathbf{H}^*[\mathbf{z}^{-1}]\mathbf{H}[\mathbf{z}] &= \sum_{\ell \in \{0, \dots, k-1\}^2} \mathbf{H}^*(\ell) \mathbf{z}^\ell \sum_{\ell' \in \{0, \dots, k-1\}^2} \mathbf{H}(\ell') \mathbf{z}^{-\ell'} \\
 &= \sum_{\ell, \ell'} \text{Diag}(\mathbf{w}(\ell))^* \mathbf{F}^*(\ell) \mathbf{F}(\ell') \text{Diag}(\mathbf{w}(\ell')) \mathbf{z}^{\ell - \ell'} \\
 &= \sum_{\ell} \text{Diag}(\overline{\mathbf{w}(\ell)}) \text{Diag}(\mathbf{w}(\ell)) \\
 &= \text{Diag} \left( \sum_{\ell} \left( |\mathbf{w}_1(\ell)|^2, \dots, |\mathbf{w}_{N^2}(\ell)|^2 \right)^\top \right) \\
 &= \mathbf{D}
 \end{aligned}$$

La matrice  $\mathbf{D}$  est donc une matrice diagonale. Pour qu'elle soit inversible, il faut et il suffit que tous ses éléments diagonaux soient non nuls, ce qui est le cas si et seulement si toutes les sommes de coefficients de la fenêtre d'analyse :  $\sum_{\ell} |\mathbf{w}_j(\ell)|^2$ , pour  $j \in \{1, \dots, N^2\}$  sont non nuls.

On peut maintenant donner une expression de la matrice polyphase de synthèse grâce à la pseudo-inverse suivante :

$$\begin{aligned}
 \tilde{\mathbf{H}}[\mathbf{z}] &= (\mathbf{H}^*[\mathbf{z}^{-1}]\mathbf{H}[\mathbf{z}])^{-1} \mathbf{H}^*[\mathbf{z}^{-1}] \\
 &= \mathbf{D}^{-1} \mathbf{H}^*[\mathbf{z}^{-1}] \\
 &= \sum_{\ell \in \{0, \dots, k-1\}^2} \mathbf{D}^{-1} \text{Diag}(\mathbf{w}(\ell))^* \mathbf{F}^*(\ell) \mathbf{z}^\ell
 \end{aligned}$$

La transformation inverse est donc composée de l'utilisation de la matrice conjuguée hermitienne de la matrice  $\mathbf{F}$  employée pour l'analyse et de l'application d'une fenêtre de synthèse s'écrivant sous forme vectorisée :

$$\mathbf{w}_s(\ell) = \text{diag} \left( \mathbf{D}^{-1} \text{Diag}(\overline{\mathbf{w}(\ell)}) \right),$$

où l'opérateur  $\text{diag}$  sert à extraire la diagonale principale d'une matrice.

En repassant sous une écriture bidimensionnelle, la fenêtre de reconstruction s'écrit alors :  $\forall (\ell_1, \ell_2) \in \{0, \dots, k-1\}^2$  et  $(j_1, j_2) \in \{1, \dots, N\}^2$ ,

$$\begin{aligned}
 (W_s)_{\ell_1 N + j_1, \ell_2 N + j_2} &= (\mathbf{w}_s(\ell))_{j_1 + (j_2 - 1)N} \\
 &= \frac{\overline{\mathbf{w}_{j_1 + (j_2 - 1)N}(\ell)}}{\mathbf{D}_{j_1 + (j_2 - 1)N, j_1 + (j_2 - 1)N}} \\
 &= \frac{\overline{W_{\ell_1 N + j_1, \ell_2 N + j_2}}}{\sum_{\ell'_1=0}^{k-1} \sum_{\ell'_2=0}^{k-1} |W_{\ell'_1 N + j_1, \ell'_2 N + j_2}|^2}.
 \end{aligned} \tag{5.9}$$

À l'aide des résultats précédents et de l'équation (5.2), on en déduit que les réponses impulsionnelles du banc de filtres de synthèse s'écrivent :  $\forall (i_1, i_2) \in \{1, \dots, N\}^2$ ,  $(j_1, j_2) \in \{1, \dots, M\}^2$ , et  $(\ell_1, \ell_2) \in \{0, \dots, k-1\}^2$ ,

$$\tilde{h}_{(j_1, j_2)}(-N\ell_1 - i_1, -N\ell_2 - i_2) = \overline{E_{j_1, \ell_1 N + i_1} E_{j_2, \ell_2 N + i_2}} (W_s)_{\ell_1 N + i_1, \ell_2 N + i_2}.$$

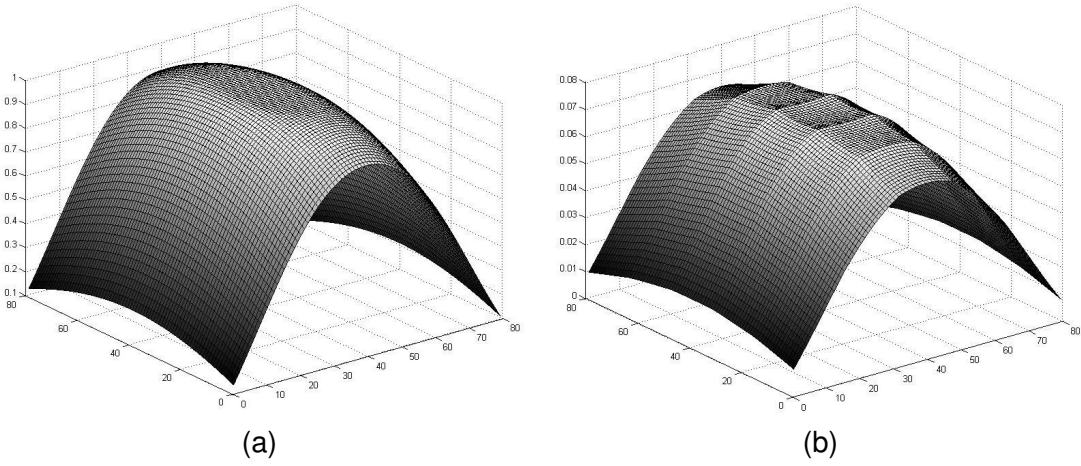


FIGURE 5.3 – Exemple de couple de fenêtres (a) d'analyse ( $W_c$ ) et (b) synthèse non séparables.

**Remarque 5.6** Les calculs dans le cas où la dimension  $L$  du banc de filtre est quelconque, se font exactement de la même manière. Dans ce cas on trouve que la fenêtre de synthèse se calcule par la formule :  $\forall (\ell_1, \dots, \ell_L) \in \{0, \dots, k-1\}^L$  et  $(j_1, \dots, j_L) \in \{1, \dots, N\}^L$ ,

$$(W_s)_{\ell_1 N + j_1, \dots, \ell_L N + j_L} = \frac{\overline{W_{\ell_1 N + j_1, \dots, \ell_L N + j_L}}}{\sum_{\ell'_1=0}^{k-1} \dots \sum_{\ell'_L=0}^{k-1} |W_{\ell'_1 N + j_1, \dots, \ell'_L N + j_L}|^2}. \quad (5.10)$$

En particulier ces calculs restent valables en dimension 1.

### 5.5.2 Exemple de fenêtre de reconstruction

Nous avons appliqué les expressions précédentes avec la fenêtre d'analyse définie par : pour tout  $(n_1, n_2) \in \{1, \dots, kN\}^2$ ,

$$W_c(n_1, n_2) = \cos \left( \left( \sqrt{\left( a \frac{2n_1 - kN}{2kN} \right)^2 + \left( b \frac{2n_2 - kN}{2kN} \right)^2} - R \right) \pi \mathbf{1}_A(n_1, n_2) \right)$$

où  $(a, b, R) \in (\mathbb{R}_+^*)^3$  tels que  $a^2 + b^2 \leq (2R + 1)^2$  et  $\mathbf{1}_A$  est la fonction caractéristique de l'ensemble :

$$A = \left\{ (u, v) \mid (a(2u - kN))^2 + (b(2v - kN))^2 \geq (2kNR)^2 \right\}.$$

Nous avons alors utilisé l'équation (5.9) pour calculer la fenêtre de synthèse associée. Ces deux fenêtres sont représentées à la figure 5.3, avec  $N = 16$ ,  $k = 5$ ,  $a = 7/5$ ,  $b = 4/5$  et  $R = 3/8$ .

La forme de la fenêtre de synthèse suit globalement celle de la fenêtre d'analyse, ce qui implique en particulier que si l'on a une bonne atténuation de la fenêtre d'analyse

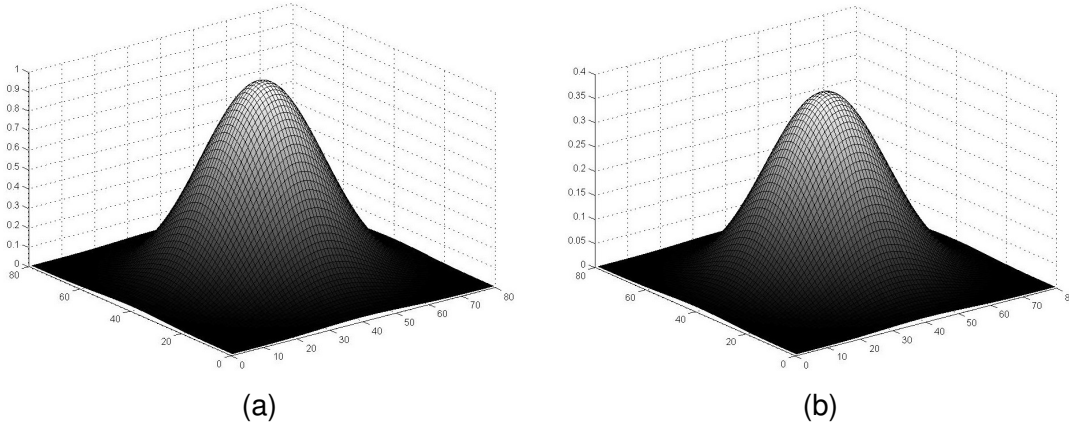


FIGURE 5.4 – Exemple de couple de fenêtres (a) d'analyse ( $W_k$ ) et (b) synthèse non séparables.

aux bords, alors la fenêtre de synthèse présente également un tel comportement. La division dans la formule (5.9) entraîne les “crêtes” qu'on peut observer sur la fenêtre de reconstruction.

Un test a également été réalisé avec une fenêtre 2D de Kaiser non séparable : pour tous  $(n_1, n_2) \in \{1, \dots, kN\}^2$ ,

$$W_k(n_1, n_2) = \frac{I_0 \left( \nu \sqrt{1 - \left( \cos(\alpha) \frac{2n_1 - kN - 1}{2kN} \right)^2 + \left( \sin(\alpha) \frac{2n_2 - kN - 1}{2kN} \right)^2} \right)}{I_0(\nu)}, \quad (5.11)$$

où  $I_0$  est la fonction de Bessel modifiée du premier genre d'ordre 0 [Carrier *et al.*, 1983].

La figure 5.4 présente un exemple de fenêtre de Kaiser 2D avec les paramètres  $N = 16$ ,  $k = 5$ ,  $\alpha = \pi/4$  et  $\nu = 15$ . À nouveau, à l'aide de la formule (5.9) nous avons calculé la fenêtre de synthèse associée à cette fenêtre d'analyse. On voit que dans ce cas il n'y a pas d'effet de “crête” visible, cela est dû au fait que les sommes au dénominateur de l'équation (5.9) présentent des valeurs très proches.

### 5.5.3 Exemple de banc de filtres

Pour illustrer cette méthode de construction de bancs de filtres de synthèse, nous allons utiliser la fenêtre définie à la section 5.5.2 et nous avons choisi la matrice  $\mathbf{E}$  suivante :

$$\mathbf{E}_{n_1, n_2} = \frac{1}{\sqrt{M}} e^{-i(n_1 - \frac{M}{2} - \frac{1}{2})(n_2 - \frac{M}{2} - \frac{1}{2}) \frac{2\pi}{M}},$$

pour tout  $(n_1, n_2) \in \{1, \dots, M\}^2$ . Cette matrice vérifie bien la condition  $\mathbf{E}^* \mathbf{E} = \mathbf{I}_M$ .

Pour cet exemple, considérons les paramètres :  $k = 5$  et  $N = 4$  (donc  $M = 20$ ) pour la matrice  $\mathbf{E}$  et  $a = b = 7/4$  et  $R = 1/8$  pour la fenêtre. Les parties réelles et imaginaires des réponses impulsionnelles des filtres 2D d'analyse et de synthèse (respectivement) sont

montrées sur les figures 5.5 et 5.6. Le banc de filtres de synthèse étant construit directement par conjugaison du banc de filtres d'analyse et utilisation d'une fenêtre, les filtres de synthèse présentent des propriétés de concentration spatiale (et de sélectivité fréquentielle) équivalentes à ceux d'analyse, contrairement aux filtres de synthèse obtenus à la section 5.4.4.

## **5.6 Conclusion**

Dans ce chapitre, nous avons étudié quelques aspects des bancs de filtres multidimensionnels sous-échantillonnés rectangulairement. Comme dans le cas unidimensionnel, nous nous sommes intéressés à la matrice polyphase du banc de filtres d'analyse pour caractériser la propriété de reconstruction parfaite d'un tel système, ce qui nous a conduit à proposer une nouvelle méthode utilisant la notion de résultant pour tester l'inversibilité d'une matrice polynomiale à deux variables. Cette méthode ne permet pas de conclure dans quelques cas particuliers, mais semble plus rapide que la méthode classique utilisant les bases de Gröbner. Nous avons également étendu d'autres résultats que nous avons présentés dans le cas unidimensionnel comme la méthode de calcul explicite d'un inverse ou la paramétrisation des bancs de filtres de synthèse grâce à une SVD. Enfin, nous avons considéré le cas particulier où la redondance est égale au facteur de recouvrement et avons vu que dans ce cas on peut donner une expression explicite de l'inverse.

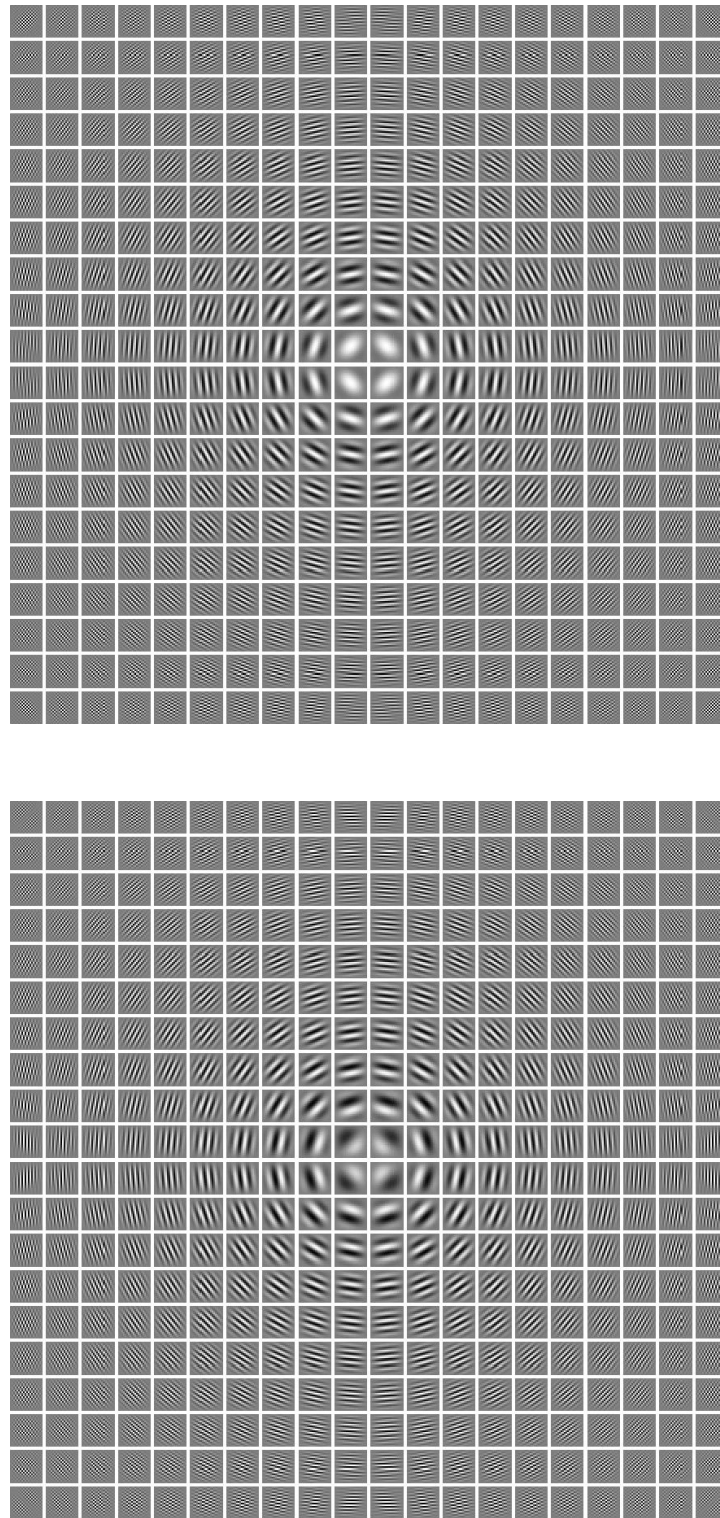


FIGURE 5.5 – Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des filtres d'analyse dans le cas  $k = k'$ .

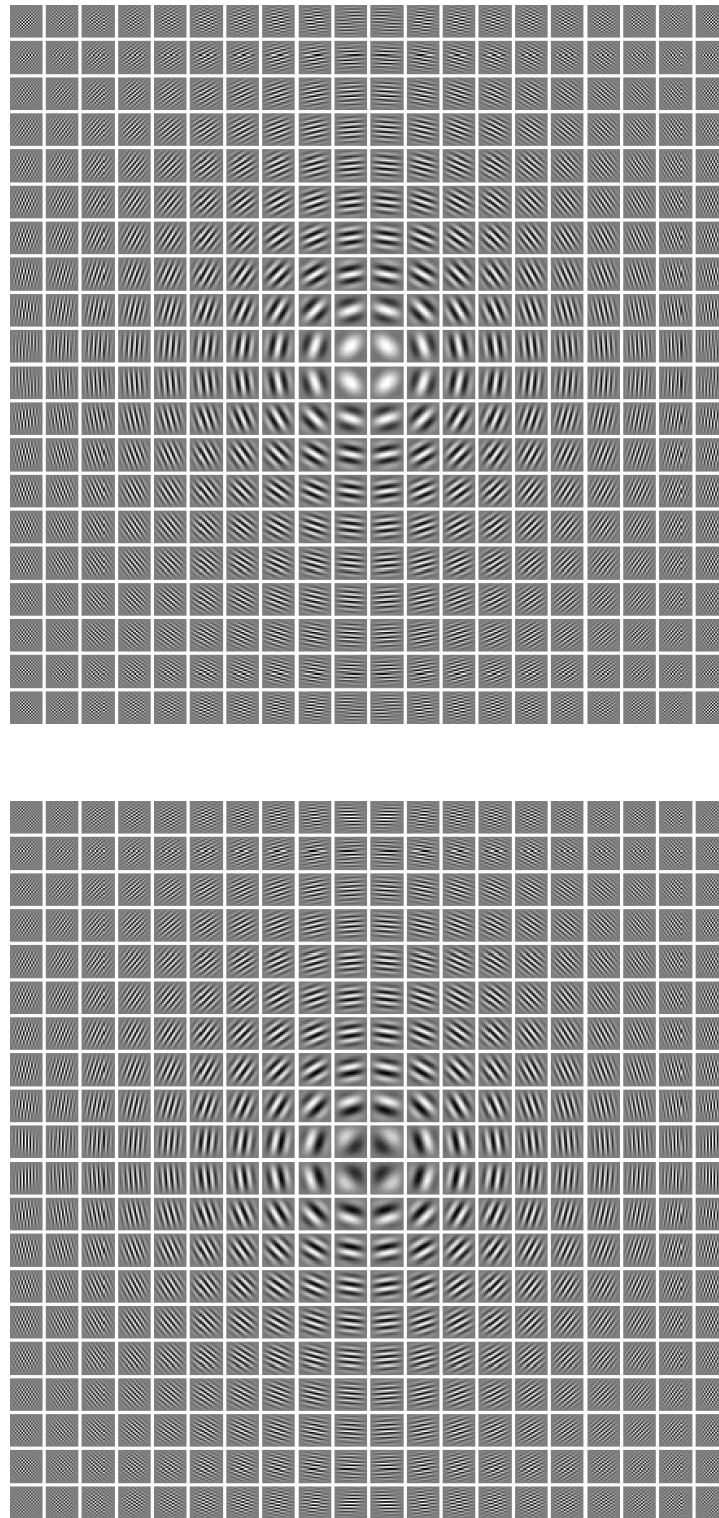


FIGURE 5.6 – Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des filtres de synthèse dans le cas  $k = k'$ .



---

## Chapitre 6

# Méthodes pour la réjection de bruit

---

Dans ce chapitre, nous allons détailler des méthodes statistiques et déterministes que nous avons employées pour débruiter des images et des signaux. Dans un premier temps, l'estimateur de Stein sans biais du risque quadratique est appliqué au débruitage d'images et de signaux perturbés par un bruit blanc gaussien. Nous avons travaillé sur deux méthodes de débruitage de type SURE-LET dans le cas d'une transformation complexe à recouvrement : la première méthode utilisant les coefficients avant reconstruction pour le calcul de l'estimateur et la seconde travaillant directement sur les signaux reconstruits. Nous avons appliqué ces méthodes sur des signaux et des images obtenant ainsi des résultats proches de ceux fournis par l'oracle et légèrement supérieurs à ceux donnés par d'autres méthodes récentes de débruitage. Dans un second temps, une méthode à caractère plus déterministe est proposée pour un type de problème différent : le "filtrage" directionnel de données anisotropes. Le terme filtrage, ici utilisé, est communément employé en sismique mais il peut apparaître impropre puisqu'il désigne une opération d'élimination de bruits dits cohérents. Ce filtrage, utilisable pour des données 2D et 3D, est finalement amélioré grâce à la prise en compte d'un paramètre permettant d'évaluer les mélanges de directions.

### 6.1 Débruitage basé sur le principe de Stein

#### 6.1.1 Présentation et fondements théoriques

En 1961, James et Stein [James, Stein, 1961] ont introduit un estimateur du vecteur des moyennes  $\theta \in \mathbb{R}^d$  d'observations :  $\mathbf{x} \sim \mathcal{N}(\theta, \sigma^2 \mathbf{I}_d)$ . Cet estimateur permet d'obtenir une erreur quadratique moyenne plus petite que l'estimateur du maximum de vraisemblance sur des données de dimension  $d$  supérieure ou égale à 3. Ce résultat a ensuite été généralisé par Block [Bock, 1975] au cas  $\mathcal{N}(\theta, \Gamma)$ , où  $\Gamma \in \mathbb{R}^{d \times d}$  est une matrice symétrique définie positive. Dans ce cas, l'estimateur de James Stein domine celui du maximum de vraisemblance dès que la *dimension effective* :  $\frac{\text{Tr}(\Gamma)}{\lambda_{\max}(\Gamma)}$  (où  $\lambda_{\max}(\Gamma)$  est la plus grande valeur propre de  $\Gamma$  et  $\text{Tr}$  représente l'opérateur de trace d'une matrice) est

plus grande que 2. La formule du risque de l'estimateur de James-Stein fait intervenir le vecteur  $\theta$ , ce qui dans la pratique est un problème puisque  $\theta$  est inconnu. Pour pallier cette difficulté, Stein [Stein, 1981] a alors introduit en 1981 un estimateur sans biais du risque pour des estimateurs s'écrivant sous la forme  $\hat{\theta} = \mathbf{x} + \mathbf{g}(\mathbf{x})$ , où  $\mathbf{g}$  est continue, différentiable presque partout et vérifie :

$$\mathbb{E} \left( \sum_{i=1}^d \left| \frac{\partial g_i}{\partial x_i}(\mathbf{x}) \right| \right) < \infty.$$

Le résultat, prouvé dans le cas d'une matrice de covariance de la forme  $\sigma^2 \mathbf{I}_d$ , peut être étendu aux matrices de covariance symétriques définies positives. L'estimateur sans biais de Stein est maintenant connu sous l'acronyme *SURE* (*Stein's Unbiased Risk Estimate*). Notons que pour plus de détails concernant cet estimateur, on pourra consulter [Richards, 1999].

Par la suite, Donoho et Johnstone l'ont utilisé en débruitage dans le cadre d'un opérateur de seuillage doux sur des coefficients d'ondelettes pour estimer et minimiser l'erreur quadratique moyenne [Donoho, Johnstone, 1995]. Cette méthode est connue aujourd'hui sous le nom de *SureShrink*. Les fonctions de seuillages doux et dur, courantes et ayant fait la preuve de leur efficacité dans des travaux sur le débruitage, ont ensuite été améliorées [Pesquet, Leporini, 1997; Raphan, Simoncelli, 2006] en considérant des estimateurs non linéaires reposant sur des combinaisons linéaires de fonctions élémentaires. Récemment, Luisier et Blu [Luisier, Blu, 2007] ont repris ce principe de paramétrisation linéaire qu'ils ont dénommé LET (*Linear Expansion of Thresholds*) et l'ont généralisé, conduisant ainsi à la méthode SURE-LET. Mentionnons également que des travaux sur les approches SURE ont été menés dans le cadre de débruitage multivarié [Benazza-Benyahia, Pesquet, 2005; Chaux *et al.*, 2008] et de l'estimation ensembliste [Combettes, Pesquet, 2004].

Avant de rappeler le lemme exprimant le principe de Stein, au cœur des résultats présentés dans la suite, introduisons quelques notations. Supposons ici qu'un vecteur  $\mathbf{v}$  de longueur  $d$  est perturbé par un bruit gaussien  $\mathbf{n}$  de moyenne nulle et de matrice de covariance  $\Gamma^{(n)}$ . Soit  $\mathbf{w} = \mathbf{v} + \mathbf{n}$  le signal bruité observé.

**Lemme 1** Soit  $\mathbf{T} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  une fonction vectorielle continue, différentiable presque partout et telle que :

1.  $\mathbb{E} \left( \|\mathbf{T}(\mathbf{w})\|^2 \right) < \infty$ ,
2.  $\mathbb{E} \left( \left\| \frac{\partial \mathbf{T}(\mathbf{w})}{\partial \mathbf{w}^\top} \right\|_F \right) < \infty$ ,
3.  $\forall \mathbf{z} \in \mathbb{R}^d, \lim_{\|\mathbf{t}\| \rightarrow +\infty} \mathbf{T}(\mathbf{t}) \exp \left( -\frac{(\mathbf{t} - \mathbf{z})^\top (\Gamma^{(n)})^{-1} (\mathbf{t} - \mathbf{z})}{2} \right) = \mathbf{0}$ ,

où  $\|\cdot\|_F$  est la norme de Frobenius et  $\mathbb{E}(\cdot)$  représente l'espérance mathématique. On a alors :

$$\mathbb{E}(\mathbf{T}(\mathbf{w})\mathbf{v}^\top) = \mathbb{E}(\mathbf{T}(\mathbf{w})\mathbf{w}^\top) - \mathbb{E} \left( \frac{\partial \mathbf{T}(\mathbf{w})}{\partial \mathbf{w}^\top} \right) \Gamma^{(n)}.$$

Ce lemme est une reformulation du principe de Stein démontré dans [Stein, 1981]. Il permet d'exprimer une espérance faisant intervenir le signal d'origine  $\mathbf{v}$  (supposé inconnu) uniquement en fonction d'espérance faisant intervenir le signal observé  $\mathbf{w}$ .

Une méthode classique pour évaluer la qualité d'un signal reconstruit  $\hat{\mathbf{v}}$  est l'étude du rapport signal sur bruit : SNR (*Signal to Noise Ratio*). Le SNR est défini de la manière suivante :

$$\text{SNR}(\mathbf{v}, \hat{\mathbf{v}}) = 10 \log_{10} \left( \frac{\|\mathbf{v}\|^2}{\|\hat{\mathbf{v}} - \mathbf{v}\|^2} \right).$$

On voit que le dénominateur fait apparaître l'erreur quadratique :  $\|\hat{\mathbf{v}} - \mathbf{v}\|^2$ . En traitement d'images, le PSNR est plus souvent utilisé. De même que pour le SNR, la définition du PSNR fait apparaître l'erreur quadratique  $\|\hat{\mathbf{v}} - \mathbf{v}\|^2$  au dénominateur. Pour obtenir un bon débruitage (SNR ou PSNR élevé) il faut donc minimiser l'erreur quadratique. Cette erreur faisant intervenir le signal d'origine  $\mathbf{v}$ , supposé inconnu, il faut donc pouvoir l'estimer en utilisant uniquement le signal observé. En développant l'expression de l'erreur quadratique moyenne  $E(\|\mathbf{T}(\mathbf{w}) - \mathbf{v}\|^2)$  trois termes apparaissent : le premier ne fait intervenir que le signal observé, le deuxième ne dépend que du signal d'origine et le troisième une combinaison des deux. En supposant que le bruit et le signal d'origine sont indépendants, le terme ne contenant que le signal d'origine peut être remplacé par des termes ne faisant intervenir que l'observation et le bruit. Le lemme 1 permet d'exprimer le troisième terme uniquement en fonction du signal observé. Finalement, l'erreur quadratique moyenne va ainsi pouvoir être estimée. Ce type d'argument est à la base des preuves des théorèmes 10 et 11 qui sont présentés dans les sections suivantes.

Dans cette partie nous allons étudier l'estimateur de Stein dans le cas de transformations complexes redondantes. Plus particulièrement nous allons étudier deux cas : celui où l'estimation se fait sur les coefficients avant reconstruction et celui où l'on travaille sur les échantillons reconstruits. Dans le cas orthogonal ces deux méthodes sont équivalentes (*cf.* section 6.1.4), mais cela n'est plus vrai dans le cas de transformées non orthogonales, en particulier redondantes.

### 6.1.2 Estimation du risque sur les coefficients avant reconstruction

Dans cette section, nous allons nous intéresser à une première forme d'estimation de Stein portant sur les coefficients après transformation. Plus précisément, le but va être ici de choisir, par une approche SURE, les coefficients dans le domaine transformé les plus proches possibles des coefficients obtenus à partir du signal non bruité.

On considère un signal réel  $\mathbf{x}$  de longueur  $L$ , perturbé par un bruit blanc gaussien réel centré et de variance  $\sigma^2$  noté  $\mathbf{b}$  indépendant de  $\mathbf{x}$ . Le signal réel observé est noté  $\mathbf{y} = \mathbf{x} + \mathbf{b}$  (autrement dit :  $y_n = x_n + b_n$ , pour tout  $n \in \{1, \dots, L\}$ ). On suppose que la transformation linéaire de décomposition sur le signal  $\mathbf{x}$  est représentée par la matrice  $\mathbf{D}$ , et que l'opérateur de reconstruction est noté  $\mathbf{R}$ . En notant  $L'$  la longueur du signal après transformation, on a donc des matrices de taille  $L' \times L$  pour  $\mathbf{D}$  et  $L \times L'$  pour  $\mathbf{R}$ . Enfin, on suppose que les matrices  $\mathbf{D}$  et  $\mathbf{R}$  sont complexes et que  $\mathbf{RD} = \mathbf{I}_L$  (cas de la reconstruction parfaite). On note :  $\mathbf{Y} = \mathbf{Dy}$ ,  $\mathbf{X} = \mathbf{Dx}$  et  $\mathbf{B} = \mathbf{Db}$ . On peut alors écrire :

$$\mathbf{Y} = \mathbf{X} + \mathbf{B}.$$

Le but de cette partie va être de construire un estimateur sans biais du risque moyen :  $E \left( \|\mathbf{F}(\mathbf{Y}) - \mathbf{X}\|^2 \right)$  pour une fonction vectorielle  $\mathbf{F} : \mathbb{C}^{L'} \rightarrow \mathbb{C}^{L'}$ , telle que  $\mathbf{F}(\mathbf{Y}) = (f_n(\mathbf{Y}))_{1 \leq n \leq L'}$  estime  $\mathbf{X}$ . Par la suite cet estimateur sera utilisé pour optimiser une fonction  $\mathbf{F}$  minimisant l'erreur quadratique moyenne. Avant d'examiner la forme de l'estimateur, il nous faut d'abord introduire quelques notations.

### 6.1.2-a) Notations

Définissons d'abord les  $L'$  vecteurs de  $\mathbb{R}^2$  représentant chaque coefficient après décomposition, pour tout  $j \in \{1, \dots, L'\}$  :

$$\mathbf{Y}_j = \begin{pmatrix} Y_j^R \\ Y_j^I \end{pmatrix} = \begin{pmatrix} \mathbf{D}_j^R \mathbf{y} \\ \mathbf{D}_j^I \mathbf{y} \end{pmatrix}, \quad \text{où} \quad \mathbf{D} = \begin{pmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_{L'} \end{pmatrix} = \begin{pmatrix} \mathbf{D}_1^R + i\mathbf{D}_1^I \\ \vdots \\ \mathbf{D}_{L'}^R + i\mathbf{D}_{L'}^I \end{pmatrix}.$$

Autrement dit, pour tout  $j \in \{1, \dots, L'\}$ ,  $\mathbf{Y}_j$  représente un vecteur de  $\mathbb{R}^2$ , composé des parties réelle et imaginaire du scalaire  $Y_j \in \mathbb{C}$ . De la même manière, on note  $\mathbf{X}_j$  le vecteur de  $\mathbb{R}^2$  :  $(X_j^R, X_j^I)^\top$ .

Nous allons supposer que l'opérateur de seuillage  $\Theta$  tel que  $\mathbf{F}(\mathbf{y}) = \Theta(\mathbf{D}\mathbf{y})$ , généralement non-linéaire, opère *point par point* sur les coefficients complexes (ou de façon scalaire). Autrement dit, il peut s'écrire :  $\Theta(\mathbf{w}) = (\vartheta_\ell(w_\ell))_{1 \leq \ell \leq L'}$  où le vecteur  $\mathbf{w} = (w_\ell)_{1 \leq \ell \leq L'}$  est un élément de  $\mathbb{C}^{L'}$  et, pour tout  $\ell \in \{1, \dots, L'\}$ ,  $\vartheta_\ell$  est une fonction de  $\mathbb{C}$  dans  $\mathbb{C}$ . Notons que cet opérateur peut éventuellement combiner les parties réelles et imaginaires de ces complexes. Des opérateurs vectoriels plus généraux peuvent également être envisagés [Chaux *et al.*, 2008], mais dans ce travail nous nous sommes concentrés sur les opérateurs scalaires qui, comme nous le verrons plus loin, permettent une recherche simple du minimum de l'erreur d'estimation. Dans la suite nous allons, également, considérer les fonctions  $\theta_\ell : \mathbb{R}^2 \rightarrow \mathbb{C}$ , telles que :  $\vartheta_\ell(w_\ell) = \theta_\ell(w_\ell^R, w_\ell^I)$  pour tout  $\ell \in \{1, \dots, L'\}$ . En suivant la même convention de notation que pour les vecteurs  $\mathbf{Y}_j$ , on peut alors noter pour tout  $j \in \{1, \dots, L'\}$  :

$$\vec{\Theta}_j(\mathbf{Y}_j) = \begin{pmatrix} f_j^R(\mathbf{Y}) \\ f_j^I(\mathbf{Y}) \end{pmatrix} = \begin{pmatrix} \theta_j^R(\mathbf{Y}_j) \\ \theta_j^I(\mathbf{Y}_j) \end{pmatrix}.$$

On peut alors écrire :

$$\begin{aligned} \|\mathbf{F}(\mathbf{Y}) - \mathbf{X}\|_{\mathbb{C}^{L'}}^2 &= \sum_{j=1}^{L'} |f_j(\mathbf{Y}) - \mathbf{X}_j|^2 \\ &= \sum_{j=1}^{L'} (\theta_j^R(\mathbf{Y}_j) - X_j^R)^2 + (\theta_j^I(\mathbf{Y}_j) - X_j^I)^2 \\ &= \sum_{j=1}^{L'} \left\| \vec{\Theta}_j(\mathbf{Y}_j) - \mathbf{X}_j \right\|_{\mathbb{R}^2}^2. \end{aligned} \tag{6.1}$$

Enfin, avec les mêmes notations on définit les  $L'$  vecteurs de bruit :  $\mathbf{B}_j = \begin{pmatrix} \mathbf{D}_j^R \mathbf{b} \\ \mathbf{D}_j^I \mathbf{b} \end{pmatrix}$ .

Chaque vecteur  $\mathbf{B}_j \in \mathbb{R}^2$  représente un bruit réel gaussien centré de matrice de covariance  $\Gamma_j$  définie par :

$$\Gamma_j = \sigma^2 \begin{pmatrix} \mathbf{D}_j^R \mathbf{D}_j^{R\top} & \mathbf{D}_j^R \mathbf{D}_j^{I\top} \\ \mathbf{D}_j^I \mathbf{D}_j^{R\top} & \mathbf{D}_j^I \mathbf{D}_j^{I\top} \end{pmatrix}. \quad (6.2)$$

### 6.1.2-b) Expression de l'estimateur

Notre objectif est de minimiser l'espérance de la quantité  $\|\mathbf{F}(\mathbf{Y}) - \mathbf{X}\|^2$ . Le théorème suivant, utilisant le principe de Stein, permet de construire un estimateur sans biais de  $E(\|\mathbf{F}(\mathbf{Y}) - \mathbf{X}\|^2)$ .

**Théorème 10** *Supposons que, pour tout  $j \in \{1, \dots, L'\}$ , la fonction  $\vec{\Theta}_j$  de  $\mathbb{R}^2$  dans  $\mathbb{R}^2$  vérifie les trois conditions :*

1.  $E\left(\left\|\vec{\Theta}_j(\mathbf{Y}_j)\right\|^2\right) < \infty$ ,
2.  $E\left(\left\|\frac{\partial \vec{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top}\right\|_F\right) < \infty$ ,
3.  $\forall \mathbf{z} \in \mathbb{R}^2, \lim_{\|\mathbf{t}\| \rightarrow +\infty} \vec{\Theta}_j(\mathbf{t}) \exp\left(-\frac{(\mathbf{t} - \mathbf{z})^\top (\Gamma_j)^{-1} (\mathbf{t} - \mathbf{z})}{2}\right) = \mathbf{0}$ ,

où  $\|\cdot\|_F$  est la norme de Frobenius. Alors :

$$\varepsilon_C = \sum_{j=1}^{L'} \left( \left\|\vec{\Theta}_j(\mathbf{Y}_j) - \mathbf{Y}_j\right\|^2 + 2\text{Tr}\left(\frac{\partial \vec{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \Gamma_j\right) - \text{Tr}(\Gamma_j) \right),$$

est un estimateur sans biais de  $E(\|\mathbf{F}(\mathbf{Y}) - \mathbf{X}\|^2)$ .

Preuve : Étudions dans un premier temps les espérances, pour tout  $j \in \{1, \dots, L'\}$  :

$$E\left(\left\|\vec{\Theta}_j(\mathbf{Y}_j) - \mathbf{X}_j\right\|^2\right) = E\left(\left\|\vec{\Theta}_j(\mathbf{Y}_j)\right\|^2\right) + E\left(\left\|\mathbf{X}_j\right\|^2\right) - 2E\left(\left(\vec{\Theta}_j(\mathbf{Y}_j)\right)^\top \mathbf{X}_j\right), \quad (6.3)$$

et voyons plus précisément comment réécrire les termes faisant apparaître les vecteurs  $\mathbf{X}_j$ .

D'une part, le bruit étant supposé à moyenne nulle et indépendant du signal d'origine, on peut écrire :

$$E\left(\left\|\mathbf{Y}_j\right\|^2\right) = E\left(\left\|\mathbf{X}_j\right\|^2\right) + E\left(\left\|\mathbf{B}_j\right\|^2\right)$$

Finalement, on obtient :

$$E\left(\left\|\mathbf{X}_j\right\|^2\right) = E\left(\left\|\mathbf{Y}_j\right\|^2\right) - \text{Tr}(\Gamma_j). \quad (6.4)$$

D'autre part, d'après les hypothèses posées sur les fonction  $\vec{\Theta}_j$  on peut appliquer le lemme 1, exprimant le principe de Stein, ce qui permet d'écrire :

$$E(\vec{\Theta}_j(\mathbf{Y}_j) \mathbf{X}_j^\top) = E(\vec{\Theta}_j(\mathbf{Y}_j) \mathbf{Y}_j^\top) - E\left(\frac{\partial \vec{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top}\right) \Gamma_j.$$

Grâce à ce résultat, on obtient :

$$\begin{aligned} \mathbb{E} \left( \left( \tilde{\Theta}_j(\mathbf{Y}_j) \right)^\top \mathbf{x}_j \right) &= \text{Tr} \left( \mathbb{E}(\tilde{\Theta}_j(\mathbf{Y}_j) \mathbf{x}_j^\top) \right) = \text{Tr} \left( \mathbb{E}(\tilde{\Theta}_j(\mathbf{Y}_j) \mathbf{Y}_j^\top) \right) - \text{Tr} \left( \mathbb{E} \left( \frac{\partial \tilde{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \right) \Gamma_j \right) \\ &= \mathbb{E} \left( \left( \tilde{\Theta}_j(\mathbf{Y}_j) \right)^\top \mathbf{Y}_j \right) - \text{Tr} \left( \mathbb{E} \left( \frac{\partial \tilde{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \right) \Gamma_j \right). \end{aligned} \quad (6.5)$$

En combinant les équations (6.3), (6.4) et (6.5), on peut écrire :

$$\begin{aligned} \mathbb{E} \left( \left\| \tilde{\Theta}_j(\mathbf{Y}_j) - \mathbf{x}_j \right\|^2 \right) &= \mathbb{E} \left( \left\| \tilde{\Theta}_j(\mathbf{Y}_j) \right\|^2 \right) + \mathbb{E} \left( \left\| \mathbf{Y}_j \right\|^2 \right) - \text{Tr}(\Gamma_j) - 2\mathbb{E} \left( \left( \tilde{\Theta}_j(\mathbf{Y}_j) \right)^\top \mathbf{Y}_j \right) \\ &\quad + 2\text{Tr} \left( \mathbb{E} \left( \frac{\partial \tilde{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \right) \Gamma_j \right) \\ &= \mathbb{E} \left( \left\| \tilde{\Theta}_j(\mathbf{Y}_j) - \mathbf{Y}_j \right\|^2 \right) + 2\text{Tr} \left( \mathbb{E} \left( \frac{\partial \tilde{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \right) \Gamma_j \right) - \text{Tr}(\Gamma_j). \end{aligned}$$

On en déduit que, pour tout  $j \in \{1, \dots, L'\}$  :

$$\epsilon_j = \left\| \tilde{\Theta}_j(\mathbf{Y}_j) - \mathbf{Y}_j \right\|^2 + 2\text{Tr} \left( \frac{\partial \tilde{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \Gamma_j \right) - \text{Tr}(\Gamma_j)$$

est un estimateur sans biais de  $\mathbb{E} \left( \left\| \tilde{\Theta}_j(\mathbf{Y}_j) - \mathbf{x}_j \right\|^2 \right)$ .

Finalement, en utilisant ce résultat et l'équation (6.1) il vient que

$$\varepsilon_C = \sum_{j=1}^{L'} \epsilon_j$$

est un estimateur sans biais de  $\mathbb{E} \left( \left\| \mathbf{F}(\mathbf{Y}) - \mathbf{x} \right\|^2 \right)$ . ■

### 6.1.3 Estimation du risque après reconstruction

#### 6.1.3-a) Notations et estimateur

Dans cette partie, on considère toujours un signal réel  $\mathbf{x}$  de longueur  $L$ , perturbé par un bruit réel blanc gaussien centré de variance  $\sigma^2$  noté  $\mathbf{b}$ . Le signal bruité est à nouveau :  $\mathbf{y} = \mathbf{x} + \mathbf{b}$  indépendant de  $\mathbf{x}$ . Notons que, dans le cas du bruit blanc gaussien additif considéré ici, la matrice de covariance du bruit est simplement :  $\mathbf{\Gamma}^{(\mathbf{b})} = \sigma^2 \mathbf{I}_L$ . Notre objectif est à nouveau de construire un estimateur de l'erreur quadratique moyenne (*Mean Square Error* - MSE) :

$$\text{MSE} = \mathbb{E} \left( \sum_{n=1}^L |g_n(\mathbf{y}) - x_n|^2 \right) = \mathbb{E} \left( \left\| \mathbf{G}(\mathbf{y}) - \mathbf{x} \right\|^2 \right),$$

qui, par minimisation, nous permettra de choisir le vecteur  $\mathbf{G}(\mathbf{y})$  estimant  $\mathbf{x}$ .

Le signal d'origine  $\mathbf{x}$  est toujours supposé inconnu, il va donc falloir être capable de trouver une forme équivalente de la MSE ne faisant pas apparaître  $\mathbf{x}$ . Contrairement à

ce qui a été fait dans la partie précédente, on considère maintenant l'effet de l'estimateur directement sur les valeurs du signal et non plus sur les coefficients après décomposition.

Encore une fois, le lemme 1 va nous permettre de démontrer le théorème suivant qui propose un estimateur sans biais de la MSE.

**Théorème 11** *Supposons que  $\mathbf{G} : \mathbb{R}^L \rightarrow \mathbb{R}^L$  soit une fonction vectorielle, continue, différentiable presque partout et telle que :*

1.  $E \left( \|\mathbf{G}(\mathbf{y})\|^2 \right) < \infty$ ,
2.  $E \left( \left\| \frac{\partial \mathbf{G}(\mathbf{y})}{\partial \mathbf{y}^\top} \right\|_F \right) < \infty$ ,
3.  $\forall \mathbf{z} \in \mathbb{R}^L, \quad \lim_{\|\mathbf{t}\| \rightarrow +\infty} \mathbf{G}(\mathbf{t}) \exp \left( -\frac{(\mathbf{t} - \mathbf{z})^\top (\mathbf{t} - \mathbf{z})}{2\sigma^2} \right) = \mathbf{0}$ .

La variable aléatoire :

$$\varepsilon_E = \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2 + 2\sigma^2 \text{div}(\mathbf{G}(\mathbf{y})) - L\sigma^2 \quad (6.6)$$

est alors un estimateur sans biais de la MSE  $E \left( \|\mathbf{G}(\mathbf{y}) - \mathbf{x}\|^2 \right)$ .

La preuve de ce théorème est donnée dans [Blu, Luisier, 2007].

### 6.1.3-b) Expression de la divergence

En regardant le théorème 11, on voit que la construction de l'estimateur fait apparaître un terme de divergence que nous allons réexprimer maintenant sous une forme qui sera plus commode pour la mise en œuvre de cette méthode.

Dans ce qui précède, nous avons vu les hypothèses que  $\mathbf{G}$  doit vérifier pour que l'on puisse écrire l'estimateur. Intéressons-nous maintenant d'un peu plus près à la forme de cette fonction. La fonction  $\mathbf{G}$  est formée de la composition de l'opérateur de décomposition  $\mathbf{D}$ , d'un opérateur de seuillage  $\Theta$  et de la reconstruction  $\mathbf{R}$ , ce qui peut s'écrire :  $\mathbf{G} = \mathbf{R}\Theta\mathbf{D}$ . À nouveau, l'opérateur de seuillage  $\Theta$  opère *point par point* sur les coefficients, autrement dit, il s'écrit :  $\Theta(\mathbf{Y}) = (\vartheta_\ell(Y_\ell))_{1 \leq \ell \leq L'}$ . Dans la suite, nous introduisons l'opérateur de seuillage, équivalent à  $\Theta$  mais agissant sur les parties réelles et imaginaires :

$$\underline{\Theta}(\mathbf{Y}^R, \mathbf{Y}^I) = (\vartheta_\ell(Y_\ell^R, Y_\ell^I))_{1 \leq \ell \leq L'} = (\vartheta_\ell(Y_\ell^R + \imath Y_\ell^I))_{1 \leq \ell \leq L'}.$$

Enfin cet opérateur  $\underline{\Theta}$  de  $(\mathbb{R}^{L'})^2$  dans  $\mathbb{C}^{L'}$  est supposé tel que  $\mathbf{G} = \mathbf{R}\Theta\mathbf{D} = \mathbf{R}\underline{\Theta}(\mathbf{D}^R, \mathbf{D}^I)^\top$  (où  $\mathbf{D}^R$  et  $\mathbf{D}^I$  représentant à nouveau les parties réelle et imaginaire de  $\mathbf{D}$ ) vérifie les conditions du théorème 11 et tel que la reconstruction soit réelle pour des données d'entrées réelles.

**Proposition 6.1** *Sous les mêmes hypothèses que celles du théorème 11, la MSE peut être estimée de manière non biaisée par la variable aléatoire :*

$$\varepsilon_E = \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2 + 2\sigma^2 \left( \text{diag}(\mathbf{D}^R \mathbf{R})^\top \underline{\Theta}^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) + \text{diag}(\mathbf{D}^I \mathbf{R})^\top \underline{\Theta}^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I) \right) - L\sigma^2,$$

où  $\text{diag}$  représente l'opérateur permettant d'extraire la diagonale d'une matrice et

$$\underline{\Theta}^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) = \left( \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} \right)_{1 \leq \ell \leq L'} \quad \text{et} \quad \underline{\Theta}^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I) = \left( \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} \right)_{1 \leq \ell \leq L'}.$$

Preuve : En appliquant le théorème 11, il suffit de montrer :

$$\text{div}(\mathbf{G}(\mathbf{y})) = \text{diag}(\mathbf{D}^R \mathbf{R})^\top \underline{\Theta}^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) + \text{diag}(\mathbf{D}^I \mathbf{R})^\top \underline{\Theta}^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I).$$

On a supposé que :  $\mathbf{G} = \mathbf{R}\Theta\mathbf{D}$ , autrement dit, les composantes de la fonction vectorielle  $\mathbf{G}$  s'écrivent, pour tout  $\mathbf{y} \in \mathbb{R}^L$  et  $n \in \{1, \dots, L\}$  :

$$\begin{aligned} g_n(\mathbf{y}) &= \sum_{\ell=1}^{L'} \mathbf{R}_{n,\ell} \vartheta_\ell(Y_\ell^R + iY_\ell^I) \\ &= \sum_{\ell=1}^{L'} \mathbf{R}_{n,\ell} \theta_\ell(Y_\ell^R, Y_\ell^I). \end{aligned}$$

De plus, on a par définition pour tout  $\ell \in \{1, \dots, L'\}$  :

$$\begin{aligned} (Y_\ell^R, Y_\ell^I) &= ((\mathbf{D}^R \mathbf{y})_\ell, (\mathbf{D}^I \mathbf{y})_\ell) \\ &= \left( \sum_{m=1}^L D_{\ell,m}^R y_m, \sum_{m=1}^L D_{\ell,m}^I y_m \right). \end{aligned}$$

On en déduit que, pour tout  $n \in \{1, \dots, L\}$  :

$$\left( \frac{\partial Y_\ell^R}{\partial y_n}, \frac{\partial Y_\ell^I}{\partial y_n} \right) = (D_{\ell,n}^R, D_{\ell,n}^I).$$

À l'aide de ces relations, exprimons maintenant la divergence :

$$\begin{aligned} \text{div}(\mathbf{G}(\mathbf{y})) &= \sum_{n=1}^L \frac{\partial g_n(\mathbf{y})}{\partial y_n} \\ &= \sum_{n=1}^L \sum_{\ell=1}^{L'} \mathbf{R}_{n,\ell} \frac{\partial \theta_\ell(\mathbf{D}_\ell^R, \mathbf{D}_\ell^I)(\mathbf{y})}{\partial y_n} \\ &= \sum_{n=1}^L \sum_{\ell=1}^{L'} \mathbf{R}_{n,\ell} (\nabla \theta_\ell(Y_\ell^R, Y_\ell^I))^\top \left( \frac{\partial Y_\ell^R}{\partial y_n}, \frac{\partial Y_\ell^I}{\partial y_n} \right)^\top \\ &= \sum_{n=1}^L \sum_{\ell=1}^{L'} \mathbf{R}_{n,\ell} \left( \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R}, \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} \right) (D_{\ell,n}^R, D_{\ell,n}^I)^\top \\ &= \sum_{\ell=1}^{L'} \left( \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} (\mathbf{D}^R \mathbf{R})_{\ell,\ell} + \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} (\mathbf{D}^I \mathbf{R})_{\ell,\ell} \right). \end{aligned}$$

D'où le résultat. ■

**Remarque 6.1** En considérant les matrices  $\mathbf{D}$  et  $\mathbf{R}$  réelles, on retrouve exactement le résultat formulé dans [Blu, Luisier, 2007].



On peut donner une autre forme de la divergence qui sera moins utile dans la pratique, mais qui permet de mieux faire le lien entre les estimateurs  $\varepsilon_E$  et  $\varepsilon_C$  (comme nous le verrons à la section 6.1.4).

**Proposition 6.2** *Sous les mêmes hypothèses que celles du théorème 11, la MSE peut être estimée de manière non biaisée par la variable aléatoire :*

$$\varepsilon_E = \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2 + 2 \sum_{\ell=1}^{L'} \text{Tr} \left( \frac{\partial \vec{\Theta}_\ell(\mathbf{Y}_\ell)}{\partial \mathbf{Y}_\ell^\top} \tilde{\Gamma}_\ell \right) - L\sigma^2,$$

où pour tout  $\ell \in \{1, \dots, L'\}$  on pose :

$$\tilde{\Gamma}_\ell = \sigma^2 \begin{pmatrix} \mathbf{D}_\ell^R \mathbf{R}_\ell^{R\top} & -\mathbf{D}_\ell^R \mathbf{R}_\ell^{I\top} \\ \mathbf{D}_\ell^I \mathbf{R}_\ell^{R\top} & -\mathbf{D}_\ell^I \mathbf{R}_\ell^{I\top} \end{pmatrix}.$$

Preuve : D'après les calculs menés dans la preuve de la proposition 6.1, on a :

$$\text{div}(\mathbf{G}(\mathbf{y})) = \sum_{\ell=1}^{L'} \left( \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} (\mathbf{D}^R \mathbf{R})_{\ell,\ell} + \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} (\mathbf{D}^I \mathbf{R})_{\ell,\ell} \right).$$

Comme  $\theta_\ell(Y_\ell^R, Y_\ell^I) = \theta_\ell^R(Y_\ell^R, Y_\ell^I) + \imath \theta_\ell^I(Y_\ell^R, Y_\ell^I)$ , on peut écrire :

$$\begin{cases} \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} = \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} + \imath \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} \\ \frac{\partial \theta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} = \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} + \imath \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I}. \end{cases}$$

En remplaçant et en développant, on obtient :

$$\begin{aligned} \text{div}(\mathbf{G}(\mathbf{y})) &= \sum_{\ell=1}^{L'} \left( \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} (\mathbf{D}^R \mathbf{R}^R)_{\ell,\ell} - \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} (\mathbf{D}^R \mathbf{R}^I)_{\ell,\ell} \right. \\ &\quad \left. + \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} (\mathbf{D}^I \mathbf{R}^R)_{\ell,\ell} - \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} (\mathbf{D}^I \mathbf{R}^I)_{\ell,\ell} \right) \\ &\quad + \imath \sum_{\ell=1}^{L'} \left( \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} (\mathbf{D}^R \mathbf{R}^R)_{\ell,\ell} + \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} (\mathbf{D}^R \mathbf{R}^I)_{\ell,\ell} \right. \\ &\quad \left. + \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} (\mathbf{D}^I \mathbf{R}^R)_{\ell,\ell} + \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} (\mathbf{D}^I \mathbf{R}^I)_{\ell,\ell} \right). \end{aligned}$$

Dans l'expression ci-dessus, la partie imaginaire est en fait nulle. On peut le voir de deux manières, soit en faisant un calcul similaire à celui proposé dans la preuve de la proposition 6.1, soit en remarquant que la fonction  $\mathbf{G}$  va de  $\mathbb{R}^L$  dans  $\mathbb{R}^L$  et que donc sa divergence est forcément réelle. En factorisant la partie réelle de l'expression précédente, on a :

$$\begin{aligned} \text{div}(\mathbf{G}(\mathbf{y})) &= \sum_{\ell=1}^{L'} \text{Tr} \left[ \begin{pmatrix} \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} & \frac{\partial \theta_\ell^R(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} \\ \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} & \frac{\partial \theta_\ell^I(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} \end{pmatrix} \begin{pmatrix} \mathbf{D}_\ell^R \mathbf{R}_\ell^{R\top} & -\mathbf{D}_\ell^R \mathbf{R}_\ell^{I\top} \\ \mathbf{D}_\ell^I \mathbf{R}_\ell^{R\top} & -\mathbf{D}_\ell^I \mathbf{R}_\ell^{I\top} \end{pmatrix} \right] \\ &= \frac{1}{\sigma^2} \sum_{\ell=1}^{L'} \text{Tr} \left( \frac{\partial \vec{\Theta}_\ell(\mathbf{Y}_\ell)}{\partial \mathbf{Y}_\ell^\top} \tilde{\Gamma}_\ell \right). \end{aligned} \tag{6.7}$$

Finalement on obtient le résultat en appliquant le théorème 11 et en remplaçant la divergence par la relation (6.7). ■

### 6.1.4 Cas orthogonal

Dans les sections précédentes nous avons vu comment construire deux estimateurs du risque, l'un employant les coefficients après décomposition et l'autre les coefficients après reconstruction. En considérant que la transformation est orthogonale, nous allons voir les liens existant entre ces deux estimateurs.

La transformation étant complexe et désormais orthogonale, on a les relations suivantes :  $\mathbf{R} = \mathbf{D}^* \in \mathbb{C}^{L \times L}$  et  $\mathbf{R}\mathbf{D} = \mathbf{D}\mathbf{R} = \mathbf{I}_L$  (en particulier, on a  $L' = L$ ). On suppose toujours que l'opérateur de seuillage  $\Theta$  agit point à point sur les coefficients après décomposition. Sous ces hypothèses et en conservant les notations des sections précédentes, les estimateurs s'écrivent :

$$\begin{cases} \varepsilon_C = \sum_{j=1}^L \left( \left\| \tilde{\Theta}_j(\mathbf{Y}_j) - \mathbf{Y}_j \right\|^2 + 2\text{Tr} \left( \frac{\partial \Theta_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \mathbf{\Gamma}_j \right) - \text{Tr}(\mathbf{\Gamma}_j) \right) \\ \varepsilon_E = \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2 + 2\sigma^2 \left( \text{diag}(\mathbf{D}^R \mathbf{D}^*)^\top \underline{\Theta}^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) + \text{diag}(\mathbf{D}^I \mathbf{D}^*)^\top \underline{\Theta}^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I) \right) - L\sigma^2, \end{cases} \quad (6.8)$$

**Proposition 6.3** *Sous ces hypothèses, les estimateurs  $\varepsilon_C$  et  $\varepsilon_E$  sont égaux.*

*Preuve :* Voyons maintenant comment réécrire certains des termes apparaissant dans ces deux estimateurs.

D'une part, on remarque que l'hypothèse d'orthogonalité permet de mieux connaître les matrices de covariance  $\mathbf{\Gamma}_j$  :

$$\begin{aligned} \mathbf{D}\mathbf{D}^* = \mathbf{I}_L &\Leftrightarrow \left[ \begin{pmatrix} \mathbf{D}_1^R \\ \vdots \\ \mathbf{D}_L^R \end{pmatrix} + \imath \begin{pmatrix} \mathbf{D}_1^I \\ \vdots \\ \mathbf{D}_L^I \end{pmatrix} \right] \left[ (\mathbf{D}_1^{R\top}, \dots, \mathbf{D}_L^{R\top}) - \imath (\mathbf{D}_1^{I\top}, \dots, \mathbf{D}_L^{I\top}) \right] = \mathbf{I}_L \\ &\Rightarrow \begin{cases} \forall j \in \{1, \dots, L\}, & \mathbf{D}_j^R \mathbf{D}_j^{R\top} + \mathbf{D}_j^I \mathbf{D}_j^{I\top} = 1 \\ \forall j \in \{1, \dots, L\}, & \mathbf{D}_j^I \mathbf{D}_j^{R\top} - \mathbf{D}_j^R \mathbf{D}_j^{I\top} = 0. \end{cases} \end{aligned}$$

En combinant la première relation obtenue (notons que la deuxième relation ne nécessite pas l'hypothèse d'orthogonalité) avec l'équation 6.2, on peut calculer la trace de la matrice de covariance :

$$\sum_{j=1}^L \text{Tr}(\mathbf{\Gamma}_j) = \sum_{j=1}^L \sigma^2 \text{Tr} \begin{pmatrix} \mathbf{D}_j^R \mathbf{D}_j^{R\top} & \mathbf{D}_j^R \mathbf{D}_j^{I\top} \\ \mathbf{D}_j^I \mathbf{D}_j^{R\top} & \mathbf{D}_j^I \mathbf{D}_j^{I\top} \end{pmatrix} = \sigma^2 \sum_{j=1}^L (\mathbf{D}_j^R \mathbf{D}_j^{R\top} + \mathbf{D}_j^I \mathbf{D}_j^{I\top}) = L\sigma^2. \quad (6.9)$$

D'autre part, la transformation étant orthogonale elle conserve l'énergie, on a donc :

$$\|\mathbf{F}(\mathbf{Y}) - \mathbf{Y}\|^2 = \|\mathbf{D}^* \mathbf{F}(\mathbf{Y}) - \mathbf{D}^* \mathbf{Y}\|^2$$

Comme  $\mathbf{G}(\mathbf{y}) = \mathbf{D}^* \Theta(\mathbf{D}\mathbf{y}) = \mathbf{D}^* \mathbf{F}(\mathbf{Y})$  et  $\mathbf{D}^* \mathbf{Y} = \mathbf{y}$ , on obtient en combinant l'égalité précédente et l'équation (6.1) :

$$\sum_{j=1}^L \left\| \tilde{\Theta}_j(\mathbf{Y}_j) - \mathbf{Y}_j \right\|^2 = \|\mathbf{F}(\mathbf{Y}) - \mathbf{Y}\|^2 = \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2. \quad (6.10)$$

Enfin, dans le cas orthogonal, on a pour tout  $\ell \in \{1, \dots, L\}$  :

$$\tilde{\mathbf{\Gamma}}_\ell = \sigma^2 \begin{pmatrix} \mathbf{D}_\ell^R \mathbf{R}_\ell^{R\top} & -\mathbf{D}_\ell^R \mathbf{R}_\ell^{I\top} \\ \mathbf{D}_\ell^I \mathbf{R}_\ell^{R\top} & -\mathbf{D}_\ell^I \mathbf{R}_\ell^{I\top} \end{pmatrix} = \sigma^2 \begin{pmatrix} \mathbf{D}_\ell^R \mathbf{D}_\ell^{R\top} & \mathbf{D}_\ell^R \mathbf{D}_\ell^{I\top} \\ \mathbf{D}_\ell^I \mathbf{D}_\ell^{R\top} & \mathbf{D}_\ell^I \mathbf{D}_\ell^{I\top} \end{pmatrix} = \mathbf{\Gamma}_\ell.$$

En combinant les résultats des propositions 6.1 et 6.2, il vient :

$$\begin{aligned} \text{div}(\mathbf{G}(\mathbf{y})) &= \text{diag}(\mathbf{D}^R \mathbf{D}^*)^\top \underline{\Theta}^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) + \text{diag}(\mathbf{D}^I \mathbf{D}^*)^\top \underline{\Theta}^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I) \\ &= \frac{1}{\sigma^2} \sum_{\ell=1}^L \text{Tr} \left( \frac{\partial \tilde{\Theta}_\ell(\mathbf{Y}_\ell)}{\partial \mathbf{Y}_\ell^\top} \mathbf{\Gamma}_\ell \right). \end{aligned} \quad (6.11)$$

Finalement, en remplaçant les résultats (6.9), (6.10) et (6.11) dans l'équation 6.8, on peut écrire :

$$\begin{aligned} \varepsilon_C &= \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2 + 2\sigma^2 \text{div}(\mathbf{G}(\mathbf{y})) - L\sigma^2 \\ &= \varepsilon_E. \end{aligned}$$

■

**Remarque 6.2** L'hypothèse d'orthogonalité est nécessaire pour obtenir ce résultat. Si l'on considère le cas où  $\mathbf{R} = \beta \mathbf{D}^* \in \mathbb{C}^{L \times L}$ ,  $\beta \in \mathbb{R}_+^*$ , on peut cependant établir le résultat suivant :  $\varepsilon_C = \frac{1}{\beta} \varepsilon_E$ .

Cette proposition montre que, dans le cas orthogonal, travailler dans le domaine transformé où dans le domaine reconstruit est équivalent. Dans la suite nous allons employer des transformées redondantes (donc non orthogonales) pour lesquelles ces deux estimateurs ne sont plus équivalents.

## 6.1.5 Cas des transformées à recouvrement

### 6.1.5-a) Construction des matrices de décomposition/reconstruction

Dans cette section, nous allons employer à nouveau les notations du chapitre 3 pour définir les matrices de décomposition  $\mathbf{D}$  et de reconstruction  $\mathbf{R}$  dans le cas de bancs de filtres suréchantillonnés et à recouvrement. En particulier,  $M$  représente ici le nombre de filtres,  $N$  le facteur de décimation,  $k$  le facteur de recouvrement du banc de filtres d'analyse et  $p = p_1 + p_2 + 1$  celui du banc de filtres de synthèse (comme défini à la section 3.3).

Les transformées que nous considérons étant à recouvrement, nous allons étendre les données pour gérer les bords du signal d'entrée. Soit  $n_b \geq k$  le nombre de blocs de taille  $N$  dans le signal étendu. On définit alors la matrice de la transformation sur le signal étendu par :

$$\mathbf{D}_t = \begin{pmatrix} \mathbf{H}(0) & \cdots & \mathbf{H}(k-1) & & 0 \\ & \ddots & & \ddots & \\ 0 & & \mathbf{H}(0) & \cdots & \mathbf{H}(k-1) \end{pmatrix},$$

où la matrice bloc  $[\mathbf{H}(0) \cdots \mathbf{H}(k-1)]$  est répétée  $n_b - k + 1$  fois. Autrement dit la matrice  $\mathbf{D}_t$  appartient à  $\mathbb{C}^{(n_b - k + 1)M \times n_b N}$ .

Supposons de surcroît que  $n_b - k + 1 \geq p$ , on définit alors la matrice de reconstruction :

$$\mathbf{R} = \begin{pmatrix} \tilde{\mathbf{H}}(-p_1) & \cdots & \tilde{\mathbf{H}}(p_2) & & 0 \\ & \ddots & & \ddots & \\ 0 & & \tilde{\mathbf{H}}(-p_1) & \cdots & \tilde{\mathbf{H}}(p_2) \end{pmatrix}.$$

où la matrice bloc  $\begin{bmatrix} \tilde{\mathbf{H}}(-p_1) \cdots \tilde{\mathbf{H}}(p_2) \end{bmatrix}$  est répétée  $n_b - k - p + 2$  fois. La matrice  $\mathbf{R}$  appartient donc à  $\mathbb{C}^{(n_b - k - p + 2)N \times (n_b - k + 1)M}$ .

En utilisant la relation (3.8), on en déduit que :

$$\mathbf{RD}_t = (\mathbf{0}_{(n_b - k - p + 2)N, p_1 N} \mathbf{I}_{(n_b - k - p + 2)N} \mathbf{0}_{(n_b - k - p + 2)N, (p_2 + k - 1)N}).$$

On a choisi dans ce travail d'étendre les données par symétrie. En prenant  $n_b = L/N + k + p - 2$  (en supposant  $L$  multiple de  $N$ ) et la matrice d'extension suivante :

$$\mathbf{Ext} = \begin{pmatrix} \mathbf{J}_{p_1 N} & \mathbf{0}_{p_1 N \times L - (p_1)N} \\ \vdots & \vdots \\ \mathbf{0}_{(p_2 + k - 1)N \times L - (p_2 + k - 1)N} & \mathbf{J}_{(p_2 + k - 1)N} \end{pmatrix} \in \mathbb{C}^{n_b N \times L},$$

où  $\mathbf{J}_d$  représente la matrice anti-identité de taille  $d \times d$  telle que définie à la section 3.4. On définit ainsi une extension symétrique d'un signal de longueur  $L$ , vérifiant :

$$\mathbf{RD}_t \mathbf{Ext} = \mathbf{I}_L.$$

Ceci revient donc à considérer la matrice de décomposition :  $\mathbf{D} = \mathbf{D}_t \mathbf{Ext} \in \mathbb{C}^{(n_b - k + 1)M \times L}$

D'autres types d'extensions, par exemple périodiques, sont envisageables [Brislaw, 1996; Lin, Smith, 2008]; dans ce travail nous nous sommes limités à des extensions symétriques.

### 6.1.5-b) Construction des vecteurs de paramètres $\alpha_I$ et $\alpha_R$

Afin de pouvoir calculer l'estimateur  $\varepsilon_E$  défini dans la proposition 6.1, il faut d'abord calculer les vecteurs :  $\alpha_R = \text{diag}(\mathbf{D}^R \mathbf{R})$  et  $\alpha_I = \text{diag}(\mathbf{D}^I \mathbf{R})$ . On peut remarquer que pour une longueur de signal  $L$  fixée et un banc de filtres d'analyse/synthèse donné, le calcul de ces vecteurs ne doit être réalisé qu'une seule fois. Cependant les tailles des matrices étant liées à la longueur du signal, il est clair que ce calcul peut vite devenir long et fastidieux pour des signaux de grande taille.

#### Méthode générale - Monte Carlo

Une alternative au calcul explicite des matrices  $\mathbf{R}$  et  $\mathbf{D}$  est proposée par Blu et Luisier [Blu, Luisier, 2007]. Elle consiste à utiliser une méthode de Monte Carlo pour obtenir une approximation de  $\alpha_R$  et  $\alpha_I$ . Cette méthode est basée sur le fait que  $E(\text{diag}(\mathbf{DRbb}^T)) = \text{diag}(\mathbf{DR})$  où  $\mathbf{b}$  est un vecteur de loi normale centrée réduite ayant  $L'$  éléments. Il suffit donc de générer un nombre  $I$  de réalisations :  $(\mathbf{b}_i)_{i=1}^I$ , de calculer pour chaque réalisation  $\mathbf{v}_i = \text{diag}(\mathbf{DRb}_i \mathbf{b}_i^T)$  et finalement de considérer qu'une estimation du vecteur de paramètres est donnée par la moyenne des  $I$  vecteurs  $\mathbf{v}_i$ . Cette méthode converge après un nombre raisonnable de tirages (entre 1000 et 3000 tirages pour avoir une bonne approximation), mais reste néanmoins coûteuse en temps de calcul.

### Méthode déterministe

Dans le cas qui nous intéresse ici, les matrices blocs  $\mathbf{R}$  et  $\mathbf{D}$  présentent des structures particulières qui permettent de donner la forme des vecteurs  $\alpha_R$  et  $\alpha_I$ .

Étudions le produit  $\mathbf{D}_t \mathbf{ExtR}$ , on a :

$$\mathbf{ExtR} = \begin{pmatrix} \mathbf{E}_1 \\ \mathbf{R} \\ \mathbf{E}_2 \end{pmatrix},$$

où

$$\mathbf{E}_1 = \begin{pmatrix} 0 & \mathbf{J}_N \tilde{\mathbf{H}}(-p_1) & \cdots & \mathbf{J}_N \tilde{\mathbf{H}}(p_2) & \vdots \\ & \ddots & & \ddots & \vdots \\ \mathbf{J}_N \tilde{\mathbf{H}}(-p_1) & \cdots & \mathbf{J}_N \tilde{\mathbf{H}}(p_2) & 0 & \vdots \end{pmatrix} \begin{matrix} \mathbf{0}_{p_1 N \times (n_b - k - 2p_1 - p_2 + 1)M} \\ \\ \end{matrix}$$

$\in \mathbb{C}^{p_1 N \times (n_b - k + 1)M}$

et

$$\mathbf{E}_2 = \begin{pmatrix} \vdots & 0 & \mathbf{J}_N \tilde{\mathbf{H}}(-p_1) & \cdots & \mathbf{J}_N \tilde{\mathbf{H}}(p_2) \\ \mathbf{0}_{(p_2 + k - 1)N \times (n_b - 2k - p_1 - 2p_2 + 2)M} & \ddots & & \ddots & \vdots \\ \vdots & \mathbf{J}_N \tilde{\mathbf{H}}(-p_1) & \cdots & \mathbf{J}_N \tilde{\mathbf{H}}(p_2) & 0 \end{pmatrix}$$

$\in \mathbb{C}^{(p_2 + k - 1)N \times (n_b - k + 1)M}$

L'objectif étant de calculer les termes diagonaux du produit  $\mathbf{D}_t \mathbf{ExtR}$ , nous nous limitons à l'étude des blocs diagonaux de taille  $M \times M$ . Dans ce cadre, le bloc  $\mathbf{E}_1$  n'agit, dans le calcul des blocs situés sur la diagonale, que sur les  $p_1 M$  premières lignes de  $\mathbf{D}_t$ . De même, la matrice  $\mathbf{E}_2$  n'agit que sur les  $(k + p_2 - 1)M$  dernières lignes de  $\mathbf{D}_t$ . On en déduit que les  $p_1$  premiers blocs diagonaux de taille  $M \times M$  de  $\mathbf{D}_t \mathbf{ExtR}$  et les  $(k + p_2 - 1)$  derniers blocs correspondent à l'action de l'extension symétrique. Les  $(n_b - k + 1) - p_1 - (k + p_2 - 1) = n_b - 2k - p + 3$  blocs diagonaux restant (entre ceux issus de l'extension) sont identiques et s'écrivent :

$$\sum_{\ell=0}^{\min(k-1, p_1)} \mathbf{H}(\ell) \tilde{\mathbf{H}}(-\ell).$$

On en déduit que les vecteurs  $\alpha_R$  et  $\alpha_I$  sont de la forme :

- les  $p_1 M$  premiers points de ces vecteurs, ainsi que les  $(k + p_2 - 1)M$  derniers, correspondent à l'extension du signal par symétrie,
- entre ces deux parties, il y a  $n_b - 2k - p + 3$  blocs identiques de dimension  $M$ .

La longueur du signal  $L$  n'intervient donc dans la structure des vecteurs  $\alpha_R$  et  $\alpha_I$  qu'à travers de  $n_b$ . On peut dès lors facilement construire directement ces vecteurs, pour une longueur de signal quelconque, à partir d'un vecteur de longueur plus courte. Dans la pratique, il suffit de considérer un unique bloc dans la partie centrale, autrement dit la longueur du vecteur que nous allons traiter est :

$$n_b - 2k - p + 3 = 1 \Leftrightarrow L/N + k + p - 2 - 2k - p + 3 = 1 \Leftrightarrow L = kN.$$

La méthode de construction des vecteurs  $\alpha_R$  et  $\alpha_I$  est alors :

1. Construction de  $\mathbf{R}_{kN}$  et  $\mathbf{D}_{kN}$  pour un signal de longueur  $kN$ .
2. Calcul de  $\alpha_{R_{kN}} = \text{diag}(\mathbf{D}_{kN}^R \mathbf{R}_{kN})$  et  $\alpha_{I_{kN}} = \text{diag}(\mathbf{D}_{kN}^I \mathbf{R}_{kN})$ .
3. Calcul de  $\alpha_R$  et  $\alpha_I$  par les formules :

$$\begin{aligned} \alpha_R = & \left( (\alpha_{R_{kN}}(\ell))_{1 \leq \ell \leq p_1 M}, (\alpha_{R_{kN}}(\ell))_{p_1 M+1 \leq \ell \leq p_1 M+M}, \dots \right. \\ & \left. \dots, (\alpha_{R_{kN}}(\ell))_{p_1 M+1 \leq \ell \leq p_1 M+M}, (\alpha_{R_{kN}}(\ell))_{p_1 M+M+1 \leq \ell \leq (2-k)M} \right), \\ \alpha_I = & \left( (\alpha_{I_{kN}}(\ell))_{1 \leq \ell \leq p_1 M}, (\alpha_{I_{kN}}(\ell))_{p_1 M+1 \leq \ell \leq p_1 M+M}, \dots \right. \\ & \left. \dots, (\alpha_{I_{kN}}(\ell))_{p_1 M+1 \leq \ell \leq p_1 M+M}, (\alpha_{I_{kN}}(\ell))_{p_1 M+M+1 \leq \ell \leq (2-k)M} \right) \end{aligned}$$

où le bloc central est répété  $n_b - 2k - p + 3$  fois.

Les matrices  $\mathbf{D}_{kN}$  et  $\mathbf{R}_{kN}$  utilisées ci-dessus, sont les matrices de décomposition et de reconstruction pour un signal de longueur  $kN$  construites avec les formules de la section 6.1.5-a).

### 6.1.6 Extension en dimension 2

Voyons maintenant comment étendre les méthodes d'estimation présentées dans les sections précédentes au cas des images.

Soit  $\mathcal{X}$  une image de taille  $L \times L$  perturbée par un bruit blanc gaussien centré  $\mathcal{B}$  de variance  $\sigma^2$ . On note  $\mathcal{Y} = \mathcal{X} + \mathcal{B}$  l'image bruitée. Notons désormais par  $\mathbf{y}$  la forme vectorisée de l'image  $\mathcal{Y}$  grâce à la relation :  $y_{i_1+(i_2-1)L} = \mathcal{Y}_{i_1,i_2}$  pour tout  $(i_1, i_2) \in \{1, \dots, L\}^2$ . Les transformations que nous avons utilisées dans ce chapitre étant séparables, on peut écrire la décomposition en dimension 2 assez simplement, en reprenant les notations de la dimension 1 :

$$\mathbf{D}\mathcal{Y}\mathbf{D}^\top = \left( \sum_{i_1=1}^L \sum_{i_2=1}^L D_{n_1,i_1} \mathcal{Y}_{i_1,i_2} D_{n_2,i_2} \right)_{1 \leq n_1, n_2 \leq L'},$$

ce qui peut se réécrire sous forme vectorisée, comme un vecteur  $\mathbf{Y}$  de composantes :

$$Y_n = \sum_{i=1}^{L^2} \mathcal{D}_{n,i} y_i,$$

où la matrice de décomposition  $\mathcal{D} \in \mathbb{C}^{L'^2 \times L^2}$  est définie par :

$$\forall (n_1, n_2, i_1, i_2) \in \{1, \dots, L'\}^2 \times \{1, \dots, L\}^2, \quad \mathcal{D}_{n_1+(n_2-1)L', i_1+(i_2-1)L} = D_{n_1,i_1} D_{n_2,i_2}.$$

De même, la matrice de reconstruction  $\mathcal{R} \in \mathbb{C}^{L^2 \times L'^2}$  peut s'exprimer sous la forme :

$$\forall (n_1, n_2, i_1, i_2) \in \{1, \dots, L'\}^2 \times \{1, \dots, L\}^2, \quad \mathcal{R}_{i_1+(i_2-1)L, n_1+(n_2-1)L'} = R_{i_1,n_1} R_{i_2,n_2}.$$

Avec ces notations et en utilisant que  $\mathbf{R}\mathbf{D} = \mathbf{I}_L$ , on a :

$$\begin{aligned} (\mathcal{R}\mathcal{D})_{i_1+(i_2-1)L, j_1+(j_2-1)L} &= (\mathbf{R}\mathbf{D})_{i_1,j_1} (\mathbf{R}\mathbf{D})_{i_2,j_2} \\ &= \delta_{i_1-j_1} \delta_{i_2-j_2} = (\mathbf{I}_{L^2})_{i_1+(i_2-1)L, j_1+(j_2-1)L}. \end{aligned}$$

On vérifie bien que :  $\mathcal{R}\mathcal{D} = \mathbf{I}_{L^2}$ .

### 6.1.6-a) Estimateur avant reconstruction

Comme dans le cas 1D, le but est ici d'estimer les coefficients après la décomposition. Le théorème 10 s'applique naturellement en considérant l'image vectorisée et les matrices  $\mathcal{D}$  et  $\mathcal{R}$  telles qu'elles viennent d'être définies.

D'un point de vue pratique cependant, construire et utiliser les matrices  $\mathcal{D}$  et  $\mathcal{R}$  peut poser des problèmes de mémoire et de temps de calcul. On a vu qu'avec une transformation séparable, la transformée vectorielle d'une image  $\mathcal{Y}$  de taille  $L \times L$  donnée par la formule :  $\mathcal{D}\mathbf{y}$ , s'écrit sous la forme équivalente :  $\mathbf{D}\mathcal{Y}\mathbf{D}^\top$ , où  $\mathbf{D}$  est la matrice de la transformation 1D.

La matrice  $\mathcal{D}$  n'est plus alors utilisée, via ses parties réelle et imaginaire, que dans le calcul des matrices de covariance  $\Gamma_j$  du bruit (définies à la section 6.1.2-a)). La proposition suivante montre comment construire ces matrices de covariance sans avoir à manipuler  $\mathcal{D}^R$  et  $\mathcal{D}^I$ , mais uniquement les parties réelle et imaginaire de la matrice  $\mathbf{D}$  de la transformation 1D.

**Proposition 6.4** *Avec les notations introduites précédemment, les matrices  $(\Gamma_j)_{1 \leq j \leq L'^2}$  de covariance s'écrivent, pour tout couple  $(n_1, n_2) \in \{1, \dots, L'\}^2$  tel que  $j = n_1 + (n_2 - 1)L'$  :*

$$\begin{aligned} (\Gamma_j)_{1,1} &= \sigma^2 \left( (\mathbf{d}_{RR} \mathbf{d}_{RR}^\top)_{n_1, n_2} + (\mathbf{d}_{II} \mathbf{d}_{II}^\top)_{n_1, n_2} - 2(\mathbf{d}_{IR} \mathbf{d}_{IR}^\top)_{n_1, n_2} \right) \\ (\Gamma_j)_{1,2} &= \sigma^2 \left( (\mathbf{d}_{RR} \mathbf{d}_{RI}^\top)_{n_1, n_2} + (\mathbf{d}_{RI} \mathbf{d}_{RR}^\top)_{n_1, n_2} - (\mathbf{d}_{IR} \mathbf{d}_{II}^\top)_{n_1, n_2} - (\mathbf{d}_{II} \mathbf{d}_{IR}^\top)_{n_1, n_2} \right) = (\Gamma_j)_{2,1} \\ (\Gamma_j)_{2,2} &= \sigma^2 \left( (\mathbf{d}_{RR} \mathbf{d}_{II}^\top)_{n_1, n_2} + (\mathbf{d}_{II} \mathbf{d}_{RR}^\top)_{n_1, n_2} + 2(\mathbf{d}_{RI} \mathbf{d}_{IR}^\top)_{n_1, n_2} \right), \end{aligned}$$

où  $\mathbf{d}_{\alpha\beta} = \text{diag} \left( \mathbf{D}^\alpha \mathbf{D}^{\beta\top} \right)$  avec  $(\alpha, \beta) \in \{R, I\}^2$ .

*Preuve :* En écrivant la définition de  $\Gamma_j$  donnée par l'équation (6.2) avec les notations dans le cas bidimensionnel, on a, pour tout  $j \in \{1, \dots, L'^2\}$  :

$$\Gamma_j = \sigma^2 \begin{pmatrix} \mathcal{D}_j^R \mathcal{D}_j^{R\top} & \mathcal{D}_j^R \mathcal{D}_j^{I\top} \\ \mathcal{D}_j^I \mathcal{D}_j^{R\top} & \mathcal{D}_j^I \mathcal{D}_j^{I\top} \end{pmatrix} = \sigma^2 \begin{pmatrix} (\text{diag}(\mathcal{D}^R \mathcal{D}^{R\top}))_j & (\text{diag}(\mathcal{D}^R \mathcal{D}^{I\top}))_j \\ (\text{diag}(\mathcal{D}^I \mathcal{D}^{R\top}))_j & (\text{diag}(\mathcal{D}^I \mathcal{D}^{I\top}))_j \end{pmatrix}.$$

Étudions  $(\Gamma_j)_{1,1} = (\text{diag}(\mathcal{D}^R \mathcal{D}^{R\top}))_j$ . En utilisant la définition de la section 6.1.6 et avec  $(n_1, n_2) \in \{1, \dots, L'\}^2$  tels que  $j = n_1 + (n_2 - 1)L'$ , on peut écrire :

$$\begin{aligned} (\Gamma_j)_{1,1} &= \sigma^2 (\mathcal{D}^R \mathcal{D}^{R\top})_{n_1 + (n_2 - 1)L', n_1 + (n_2 - 1)L'} \\ &= \sigma^2 \sum_{k_1=1}^L \sum_{k_2=1}^L (\text{Re}(D_{n_1, k_1} D_{n_2, k_2}))^2 \\ &= \sigma^2 \left( (\mathcal{D}^R \mathcal{D}^{R\top})_{n_1, n_1} (\mathcal{D}^R \mathcal{D}^{R\top})_{n_2, n_2} + (\mathcal{D}^I \mathcal{D}^{I\top})_{n_1, n_1} (\mathcal{D}^I \mathcal{D}^{I\top})_{n_2, n_2} \right. \\ &\quad \left. - 2(\mathcal{D}^I \mathcal{D}^{R\top})_{n_1, n_1} (\mathcal{D}^I \mathcal{D}^{R\top})_{n_2, n_2} \right). \end{aligned}$$

Dans le calcul précédent, nous avons utilisé la relation :  $(\mathcal{D}^I \mathcal{D}^{R\top})_{n, n} = (\mathcal{D}^R \mathcal{D}^{I\top})_{n, n}$  pour tout  $n \in \{1, \dots, L'\}$ . On obtient finalement :

$$(\Gamma_j)_{1,1} = \sigma^2 \left( (\mathbf{d}_{RR} \mathbf{d}_{RR}^\top)_{n_1, n_2} + (\mathbf{d}_{II} \mathbf{d}_{II}^\top)_{n_1, n_2} - 2(\mathbf{d}_{IR} \mathbf{d}_{IR}^\top)_{n_1, n_2} \right),$$

où les vecteurs  $\mathbf{d}_{\alpha\beta}$  avec  $(\alpha, \beta) \in \{R, I\}^2$  sont définis comme dans l'énoncé.  
De la même manière, on obtient les expressions des éléments  $(\mathbf{\Gamma}_j)_{1,2'}$ ,  $(\mathbf{\Gamma}_j)_{2,1}$  et  $(\mathbf{\Gamma}_j)_{2,2}$ . ■

### 6.1.6-b) Estimateur après reconstruction

Dans ce cas, on peut appliquer à nouveau la proposition 6.1 obtenue dans le cas 1D mais sur l'image vectorisée  $\mathbf{y}$ . On obtient alors que, sous les mêmes hypothèses que celles du théorème 11, l'erreur quadratique moyenne peut être estimée de manière non biaisée par la variable aléatoire :

$$\varepsilon_E = \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2 + 2\sigma^2 \left( \text{diag}(\mathcal{D}^R \mathcal{R})^\top \underline{\Theta}^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) + \text{diag}(\mathcal{D}^I \mathcal{R})^\top \underline{\Theta}^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I) \right) - L^2 \sigma^2,$$

$$\text{où } \underline{\Theta}^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) = \left( \frac{\partial \vartheta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^R} \right)_{1 \leq \ell \leq L'^2} \text{ et } \underline{\Theta}^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I) = \left( \frac{\partial \vartheta_\ell(Y_\ell^R, Y_\ell^I)}{\partial Y_\ell^I} \right)_{1 \leq \ell \leq L'^2}.$$

La proposition suivante montre comment réaliser le calcul des vecteurs de paramètres dans le cas 2D.

**Proposition 6.5** Avec les notations introduites précédemment, les vecteurs  $\text{diag}(\mathcal{D}^R \mathcal{R})$  et  $\text{diag}(\mathcal{D}^I \mathcal{R})$  s'expriment sous la forme suivante :  $\forall (n_1, n_2) \in \{1, \dots, L'\}^2$ ,

$$\begin{cases} (\mathcal{D}^R \mathcal{R})_{n_1+(n_2-1)L', n_1+(n_2-1)L'} = (\mathcal{D}^R \mathbf{R})_{n_1, n_1} (\mathcal{D}^R \mathbf{R})_{n_2, n_2} - (\mathcal{D}^I \mathbf{R})_{n_1, n_1} (\mathcal{D}^I \mathbf{R})_{n_2, n_2} \\ (\mathcal{D}^I \mathcal{R})_{n_1+(n_2-1)L', n_1+(n_2-1)L'} = (\mathcal{D}^I \mathbf{R})_{n_1, n_1} (\mathcal{D}^R \mathbf{R})_{n_2, n_2} + (\mathcal{D}^R \mathbf{R})_{n_1, n_1} (\mathcal{D}^I \mathbf{R})_{n_2, n_2} \end{cases}.$$

*Preuve :* Les expressions des parties réelles et imaginaires de l'opérateur  $\mathcal{D}$  se déduisent de la définition, ainsi pour tout  $(n_1, n_2, i_1, i_2) \in \{1, \dots, L'\}^2 \times \{1, \dots, L\}^2$  :

$$\begin{cases} \mathcal{D}_{n_1+(n_2-1)L', i_1+(i_2-1)L}^R = D_{n_1, i_1}^R D_{n_2, i_2}^R - D_{n_1, i_1}^I D_{n_2, i_2}^I \\ \mathcal{D}_{n_1+(n_2-1)L', i_1+(i_2-1)L}^I = D_{n_1, i_1}^R D_{n_2, i_2}^I + D_{n_1, i_1}^I D_{n_2, i_2}^R \end{cases}.$$

On peut alors écrire pour tout  $(n_1, n_2) \in \{1, \dots, L'\}^2$  :

$$\begin{aligned} (\mathcal{D}^R \mathcal{R})_{n_1+(n_2-1)L', n_1+(n_2-1)L'} &= \sum_{i_1=1}^L \sum_{i_2=1}^L \mathcal{D}_{n_1+(n_2-1)L', i_1+(i_2-1)L}^R \mathcal{R}_{i_1+(i_2-1)L, n_1+(n_2-1)L'} \\ &= \sum_{i_1=1}^L \sum_{i_2=1}^L (D_{n_1, i_1}^R D_{n_2, i_2}^R - D_{n_1, i_1}^I D_{n_2, i_2}^I) R_{i_1, n_1} R_{i_2, n_2} \\ &= \sum_{i_1=1}^L D_{n_1, i_1}^R R_{i_1, n_1} \sum_{i_2=1}^L D_{n_2, i_2}^R R_{i_2, n_2} \\ &\quad - \sum_{i_1=1}^L D_{n_1, i_1}^I R_{i_1, n_1} \sum_{i_2=1}^L D_{n_2, i_2}^I R_{i_2, n_2} \\ &= (\mathcal{D}^R \mathbf{R})_{n_1, n_1} (\mathcal{D}^R \mathbf{R})_{n_2, n_2} - (\mathcal{D}^I \mathbf{R})_{n_1, n_1} (\mathcal{D}^I \mathbf{R})_{n_2, n_2}, \end{aligned}$$

et de la même manière :

$$(\mathcal{D}^I \mathcal{R})_{n_1+(n_2-1)L', n_1+(n_2-1)L'} = (\mathcal{D}^I \mathbf{R})_{n_1, n_1} (\mathcal{D}^R \mathbf{R})_{n_2, n_2} + (\mathcal{D}^R \mathbf{R})_{n_1, n_1} (\mathcal{D}^I \mathbf{R})_{n_2, n_2}.$$



■

Les vecteurs  $\text{diag}(\mathcal{D}^R \mathcal{R})$  et  $\text{diag}(\mathcal{D}^I \mathcal{R})$  s'expriment donc simplement en fonction des produits matriciels des parties réelle et imaginaire de  $\mathbf{D}$  avec  $\mathbf{R}$  ce qui nous épargne le calcul des matrices  $\mathcal{D}$  et  $\mathcal{R}$  (qui serait coûteux en espace mémoire et en temps de calculs).

Cette proposition est valable pour l'extension séparable en 2D de toute transformation 1D complexe. En reprenant les notations de  $\alpha_R$  et  $\alpha_I$  du cas 1D (introduites à la section 6.1.5-b)), le calcul de ces vecteurs s'écrit simplement comme :

$$\begin{cases} (\mathcal{D}^R \mathcal{R})_{n_1+(n_2-1)L', n_1+(n_2-1)L'} = (\alpha_R \alpha_R^\top - \alpha_I \alpha_I^\top)_{n_1, n_2}, \\ (\mathcal{D}^I \mathcal{R})_{n_1+(n_2-1)L', n_1+(n_2-1)L'} = (\alpha_I \alpha_R^\top + \alpha_R \alpha_I^\top)_{n_1, n_2}, \end{cases}$$

où  $(n_1, n_2) \in \{1, \dots, L'\}$ .

### 6.1.7 Formalisme SURE-LET

#### 6.1.7-a) Principe

Nous allons maintenant considérer que la fonction d'estimation  $\mathbf{F}$  (respectivement  $\mathbf{G}$ ) s'écrit comme une combinaison linéaire de  $K$  fonctions élémentaires  $\mathbf{F}_k$  (respectivement  $\mathbf{G}_k$ ) :

$$\mathbf{F}(\mathbf{Y}) = \sum_{k=1}^K a_k \mathbf{F}_k(\mathbf{Y}) \quad \text{ou} \quad \mathbf{G}(\mathbf{y}) = \sum_{k=1}^K a_k \mathbf{G}_k(\mathbf{y}).$$

Cette décomposition de  $\mathbf{F}$  (ou de  $\mathbf{G}$ ), sous la forme d'une combinaison linéaire de fonctions élémentaires faisant chacune intervenir une fonction "seuillage", constitue la partie *Linear Expansion of Thresholds* - LET - de la méthode suivant la terminologie introduite dans [Blu, Luisier, 2007]. L'intérêt de cette formulation est qu'elle permet de formuler le problème de débruitage comme un problème d'optimisation relativement simple à résoudre.

Pour obtenir le meilleur débruitage possible (au sens de l'erreur quadratique moyenne), il suffit de minimiser l'estimateur  $\varepsilon_C$  (respectivement  $\varepsilon_E$ ). Pour ce faire nous allons étudier chaque estimateur en tant que fonction du vecteur de paramètres  $\mathbf{a} = (a_1, \dots, a_K)^\top$ . En dérivant cette fonction par rapport à chaque  $a_k$ , on va pouvoir réécrire le problème sous la forme d'un système linéaire à résoudre.

Voyons maintenant comment s'écrit ce système dans les deux cadres étudiés précédemment.

#### 6.1.7-b) Minimisation de $\varepsilon_C$

Soit  $\mathbf{F}(\mathbf{Y}) = \sum_{k=1}^K a_k \mathbf{F}_k(\mathbf{Y})$ , où les  $K$  fonctions  $\mathbf{F}_k$  vérifient les hypothèse du théorème 10 et  $\mathbf{a} = (a_1, \dots, a_K)^\top \in \mathbb{R}^K$ . En reprenant les notations utilisées précédemment, on peut écrire :

$$\vec{\Theta}_j(\mathbf{Y}_j) = \sum_{k=1}^K a_k \left( \vec{\Theta}_k \right)_j(\mathbf{Y}_j),$$

où chaque fonction de seuillage  $\Theta_k : \mathbb{R}^{L'} \rightarrow \mathbb{R}^{L'}$  agit point à point sur les coefficients et  $(\vec{\Theta}_k)_j : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  est défini comme à la section 6.1.2-a). L'estimateur  $\varepsilon_C$  s'écrit alors :

$$\begin{aligned} \varepsilon_C &= \sum_{j=1}^{L'} \left( \left\| \vec{\Theta}_j(\mathbf{Y}_j) - \mathbf{Y}_j \right\|^2 + 2 \text{Tr} \left( \frac{\partial \vec{\Theta}_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \mathbf{\Gamma}_j \right) - \text{Tr}(\mathbf{\Gamma}_j) \right) \\ &= \sum_{j=1}^{L'} \left( \left\| \sum_{k=1}^K a_k (\vec{\Theta}_k)_j(\mathbf{Y}_j) - \mathbf{Y}_j \right\|^2 + 2 \sum_{k=1}^K a_k \text{Tr} \left( \frac{\partial (\vec{\Theta}_k)_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \mathbf{\Gamma}_j \right) - \text{Tr}(\mathbf{\Gamma}_j) \right) \\ &= J_C(\mathbf{a}). \end{aligned}$$

Le but est de minimiser la fonction convexe quadratique  $J_C(\mathbf{a})$ . En dérivant par rapport à chaque  $a_k$ , on obtient :

$$\begin{aligned} \frac{\partial J_C(\mathbf{a})}{\partial a_k} &= \sum_{j=1}^{L'} \left[ \left( \sum_{\ell=1}^K a_\ell (\vec{\Theta}_\ell)_j(\mathbf{Y}_j) - \mathbf{Y}_j \right)^\top (\vec{\Theta}_k)_j(\mathbf{Y}_j) + 2 \text{Tr} \left( \frac{\partial (\vec{\Theta}_k)_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \mathbf{\Gamma}_j \right) \right. \\ &\quad \left. + (\vec{\Theta}_k)_j(\mathbf{Y}_j)^\top \left( \sum_{\ell=1}^K a_\ell (\vec{\Theta}_\ell)_j(\mathbf{Y}_j) - \mathbf{Y}_j \right) \right] \\ &= 2 \sum_{\ell=1}^K a_\ell \sum_{j=1}^{L'} (\vec{\Theta}_\ell)_j(\mathbf{Y}_j)^\top (\vec{\Theta}_k)_j(\mathbf{Y}_j) \\ &\quad + 2 \sum_{j=1}^{L'} \left( \text{Tr} \left( \frac{\partial (\vec{\Theta}_k)_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \mathbf{\Gamma}_j \right) - \mathbf{Y}_j^\top (\vec{\Theta}_k)_j(\mathbf{Y}_j) \right) \end{aligned}$$

En annulant cette dérivée, on aboutit au système suivant :  $\mathbf{M}\mathbf{a} = \mathbf{c}$  où pour tout couple  $(k, \ell) \in \{1, \dots, K\}^2$  :

$$\begin{aligned} M_{k,\ell} &= \sum_{j=1}^{L'} \left( (\vec{\Theta}_\ell)_j(\mathbf{Y}_j) \right)^\top (\vec{\Theta}_k)_j(\mathbf{Y}_j) \\ c_k &= - \sum_{j=1}^{L'} \text{Tr} \left( \frac{\partial (\vec{\Theta}_k)_j(\mathbf{Y}_j)}{\partial \mathbf{Y}_j^\top} \mathbf{\Gamma}_j \right) + \mathbf{Y}_j^\top (\vec{\Theta}_k)_j(\mathbf{Y}_j). \end{aligned}$$

La recherche d'un minimum de  $\varepsilon_C$  en fonction des  $(a_\ell)_{\ell=1}^K$  est donc équivalente à la résolution d'un système linéaire. Dans le cas où  $\mathbf{F}$  est sur-paramétrée (autrement dit plusieurs jeux de paramètres  $\mathbf{a}$  sont solutions), on peut résoudre le système en considérant une pseudo-inverse de la matrice  $\mathbf{M}$ .

Finalement, le signal est reconstruit par  $\mathbf{R}\mathbf{F}(\mathbf{Y})$  où  $\mathbf{R}$  représente toujours la matrice complexe décrivant l'opération de synthèse.

6.1.7-c) Minimisation de  $\varepsilon_E$ 

De la même manière que précédemment, on va travailler dans cette partie sur une combinaison linéaire de fonctions de seuillage et chercher le vecteur de paramètres permettant de minimiser cette fois l'estimateur  $\varepsilon_E$ . On travaille ici avec la fonction de  $\mathbb{R}^L$  dans  $\mathbb{R}^L$  :

$$\mathbf{G}(\mathbf{y}) = \sum_{k=1}^K a_k \mathbf{G}_k(\mathbf{y}) = \sum_{k=1}^K a_k \mathbf{R} \Theta_k(\mathbf{D}\mathbf{y}),$$

où les  $\Theta_k : \mathbb{C}^{L'} \rightarrow \mathbb{C}^{L'}$  correspondent à des fonctions de seuillage ponctuelles. Avec les mêmes notations qu'à la section 6.1.3-b), on écrit la fonction de seuillage  $\Theta_k : (\mathbb{R}^{L'})^2 \rightarrow \mathbb{C}^{L'}$  définie par :  $\Theta_k(\mathbf{Y}^R, \mathbf{Y}^I) = \Theta_k(\mathbf{Y})$ .

D'après l'équation (6.6) on a :

$$\begin{aligned} \varepsilon_E &= \|\mathbf{G}(\mathbf{y}) - \mathbf{y}\|^2 + 2\sigma^2 \text{div}(\mathbf{G}(\mathbf{y})) - L\sigma^2 \\ &= \left\| \sum_{k=1}^K a_k \mathbf{G}_k(\mathbf{y}) - \mathbf{y} \right\|^2 + 2\sigma^2 \text{div}\left(\sum_{k=1}^K a_k \mathbf{G}_k(\mathbf{y})\right) - L\sigma^2 \\ &= J_E(\mathbf{a}). \end{aligned}$$

Pour débruiter il faut rechercher le vecteur  $\mathbf{a}$  minimisant  $J_E(\mathbf{a})$ . Pour ce faire, étudions la quantité  $\frac{\partial J_E(\mathbf{a})}{\partial a_k}$  :

$$\begin{aligned} \frac{\partial J_E(\mathbf{a})}{\partial a_k} &= \left( \sum_{\ell=1}^K a_\ell \mathbf{G}_\ell(\mathbf{y}) - \mathbf{y} \right)^\top \mathbf{G}_k(\mathbf{y}) + \mathbf{G}_k(\mathbf{y})^\top \left( \sum_{\ell=1}^K a_\ell \mathbf{G}_\ell(\mathbf{y}) - \mathbf{y} \right) + 2\sigma^2 \text{div}(\mathbf{G}_k(\mathbf{y})) \\ &= 2 \sum_{\ell=1}^K a_\ell \mathbf{G}_k(\mathbf{y})^\top \mathbf{G}_\ell(\mathbf{y}) - 2\mathbf{G}^{(k)}(\mathbf{y})^\top \mathbf{y} + 2\sigma^2 \text{div}(\mathbf{G}_k(\mathbf{y})). \end{aligned}$$

On a ainsi :

$$\begin{aligned} \frac{\partial J_E(\mathbf{a})}{\partial a_k} &= 0 \Leftrightarrow \sum_{\ell=1}^K a_\ell \mathbf{G}_k(\mathbf{y})^\top \mathbf{G}_\ell(\mathbf{y}) = \mathbf{G}_k(\mathbf{y})^\top \mathbf{y} - \sigma^2 \text{div}(\mathbf{G}_k(\mathbf{y})) \\ &\Leftrightarrow \tilde{\mathbf{M}} \mathbf{a} = \tilde{\mathbf{c}}, \end{aligned}$$

avec  $\tilde{M}_{k,\ell} = \mathbf{G}_k(\mathbf{y})^\top \mathbf{G}_\ell(\mathbf{y})$  et  $\tilde{c}_k = \mathbf{G}_k(\mathbf{y})^\top \mathbf{y} - \sigma^2 \text{div}(\mathbf{G}_k(\mathbf{y}))$ , pour tous  $(k, \ell) \in \{1, \dots, K\}^2$ . À nouveau, la recherche d'un minimum de  $\varepsilon_E$  en fonction des  $(a_\ell)_{\ell=1}^K$  est équivalente à la résolution d'un système linéaire. De même que dans la section précédente, quand le système est sous-déterminé, on peut le résoudre en considérant la matrice pseudo-inverse de  $\tilde{\mathbf{M}}$ .

On peut remarquer que, d'après la proposition 6.1, on a dans le cas d'une transformation complexe sur des données réelles, pour tout  $k \in \{1, \dots, K\}$  :

$$\text{div}(\mathbf{G}_k(\mathbf{y})) = \text{diag}(\mathbf{D}^R \mathbf{R})^\top \underline{\Theta}_k^{(1)}(\mathbf{Y}^R, \mathbf{Y}^I) + \text{diag}(\mathbf{D}^I \mathbf{R})^\top \underline{\Theta}_k^{(2)}(\mathbf{Y}^R, \mathbf{Y}^I).$$

### 6.1.8 Mise en œuvre du débruitage par SURE-LET

#### 6.1.8-a) Fonctions de seuillage

Il a été précisé précédemment que la fonction de seuillage utilisée dans ce travail est une combinaison linéaire de fonctions de seuillage point à point. Nous avons travaillé principalement ici avec des fonctions à valeurs complexes définies comme suit :

$$\begin{aligned}\tau_1(x, y) &= x + iy, \quad \forall (x, y) \in \mathbb{R}^2 \\ \tau_2(x, y) &= (x + iy) \left( 1 - e^{-\frac{(x^2 + y^2)^{\frac{m_0}{2}}}{(\alpha\sigma)^{m_0}}} \right), \quad \forall (x, y) \in \mathbb{R}^2,\end{aligned}\quad (6.12)$$

où  $m_0$  est un entier supérieur ou égal à 2 et  $\alpha \in \mathbb{R}_+^*$ . La fonction  $\tau_2$  a été choisie de cette forme car elle approche, quand  $m_0$  augmente, un seuillage dur sur des données réelles tout en restant différentiable et en permettant de travailler sur les parties réelle et imaginaire de coefficients complexes. Pour illustrer ce principe, nous avons tracé (sur la figure 6.1) des fonctions  $\tau_2(x, 0)$  pour deux valeurs de  $m_0$  ainsi que la fonction d'un seuillage dur.

Le facteur  $\alpha$  apparaissant au dénominateur dans l'exponentielle est à choisir en fonction de la transformation. Par exemple dans le cas orthogonal, une valeur prise couramment est  $\alpha = 3$ . Avec des transformations redondantes, on peut estimer (par une méthode de type Monte Carlo) des  $\alpha(\sigma)$  optimaux pour un niveau de bruit et une transformation fixés. Dans la pratique, nous avons fixé  $\alpha = \frac{3}{2}$ , qui est adapté à des niveaux de bruits intermédiaire ( $\sigma \in \{20, \dots, 40\}$ ) et des transformations faiblement redondantes ( $k' \in ]1, 3]$ ). Le facteur  $m_0$  a été pris égal à 2 dans la suite.

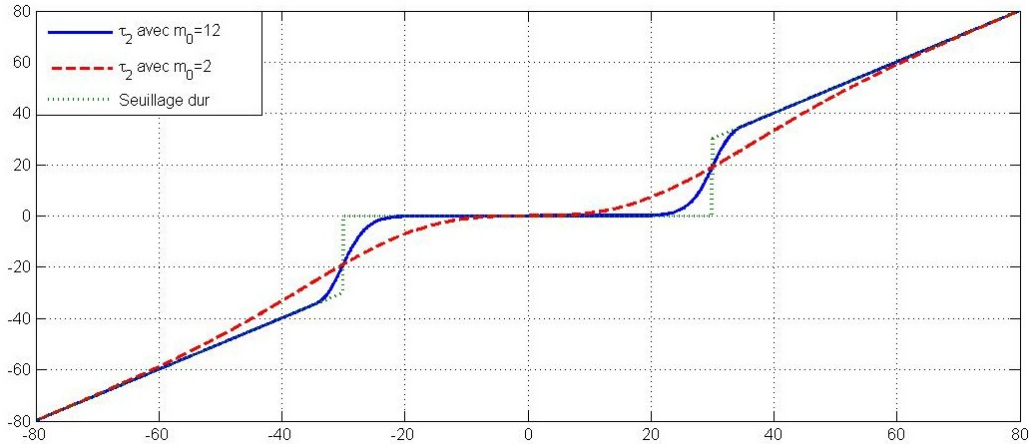


FIGURE 6.1 – Allures de la fonction de seuillage  $x \mapsto \tau_2(x, 0)$  pour  $m_0 = 2$  en rouge (pointillés longs) et  $m_0 = 12$  en bleu et de la fonction de seuillage dur en vert (pointillés courts).

Pour définir les fonctions  $\mathbf{F}_k$  ou  $\mathbf{G}_k$  correctement, il faut également définir les ensembles (disjoints) de coefficients sur lesquels on fait agir les mêmes fonctions de seuillages.

Notons par  $\mathcal{L}_k$  pour tout  $k \in \{1, \dots, K'\}$  les  $K'$  ensembles de coefficients que l'on choisit pour une application donnée. Les fonctions de seuillage  $\Theta^{(k)}$  sont alors construites à partir des fonctions précédentes en utilisant :

$$\vartheta_k(z) = \theta_k(z^R, z^I) = a_{1,k}\tau_1(z^R, z^I) + a_{2,k}\tau_2(z^R, z^I), \quad \forall z \in \mathcal{L}_k.$$

Dans ce cas on a donc  $K = 2K'$  paramètres à évaluer.



FIGURE 6.2 – Modules des coefficients de décomposition, regroupés par sous-bandes, de l'image Cameraman.

Enfin, les transformations complexes utilisées dans ce travail, telles que celles introduites au chapitre 3, sont basées sur un noyau de transformée de Fourier généralisée et sont appliquées à des données réelles. Pour être assuré que les résultats des fonctions  $\mathbf{G}_k$  et que  $\mathbf{RF}_k$  soient également réels, il suffit qu'une même fonction de seuillage agisse sur un coefficient et sur son coefficient symétrique. La seule contrainte sur le

choix des ensembles  $(\mathcal{L}_k)_{1 \leq k \leq K}$  est donc qu'ils respectent la propriété de symétrie. On peut construire de tels ensembles simplement en considérant des unions d'ensembles élémentaires constitués de deux points : un point et son symétrique.

**Exemple 8** La figure 6.2 représente le logarithme du module des coefficients obtenus après décomposition de l'image "Cameraman" par le banc de filtre d'analyse de la section 3.5.2 avec les paramètres  $k = 3$ ,  $k' = 3$  et  $N = 4$ . Les coefficients ont été ici regroupés par sous-bandes pour plus de clarté. On fait agir les fonctions de seuillage sur les sous bandes symétriques (symétrie par rapport au centre de l'image 6.2), ce qui signifie qu'il y a au plus  $(k'N)^2/2$  ensembles de coefficients considérés et pour chacun de ces ensembles on veut estimer 2 paramètres, autrement dit le vecteur  $\mathbf{a}$  est recherché dans  $\mathbb{R}^{(k'N)^2}$ .

On peut choisir plusieurs méthodes pour regrouper les sous-bandes symétriques. En notant  $\mathcal{C}_{i_1, i_2}$  les ensembles de coefficients pour chaque sous-bande  $(i_1, i_2) \in \{0, \dots, M-1\}^2$ . En supposant  $M$  pair, on peut introduire les ensembles de coefficients suivant :

$$\left\{ \begin{array}{l} \forall k \in \{1, \dots, \frac{M}{2}\}, \quad \mathcal{L}_k^{(1)} = \bigcup_{\{(i_1, i_2) \mid \max(|i_1|, |i_2|) = \frac{2k-1}{2}\}} \mathcal{C}_{i_1 + \frac{M-1}{2}, i_2 + \frac{M-1}{2}} \\ \forall k \in \{1, \dots, M\}, \quad \mathcal{L}_k^{(2)} = \bigcup_{\{(i_1, i_2) \mid \max(|i_1|, |i_2|) \text{sgn}(i_1 i_2) = k - \frac{M+1}{2}\}} \mathcal{C}_{i_1 + \frac{M-1}{2}, i_2 + \frac{M-1}{2}} \\ \forall (k_1, k_2) \in \{1, \dots, \frac{M}{2}\}^2, \\ \quad \mathcal{L}_{k_1 + (k_2-1)\frac{M}{2}}^{(3)} = \bigcup_{\{(i_1, i_2) \mid |i_1| = \frac{2k_1-1}{2} \text{ et } |i_2| = \frac{2k_2-1}{2}\}} \mathcal{C}_{i_1 + \frac{M-1}{2}, i_2 + \frac{M-1}{2}} \\ \forall (k_1, k_2) \in \{1, \dots, M\} \times \{1, \dots, \frac{M}{2}\}, \quad \mathcal{L}_{k_1 + (k_2-1)M}^{(4)} = \mathcal{C}_{k_1-1, k_2-1} \cup \mathcal{C}_{M-k_1, M-k_2}. \end{array} \right.$$

**Remarque 6.3** Avec  $M$  impair on peut définir des ensembles équivalents en considérant la sous-bande réelle  $(i_1, i_2) = (\frac{M-1}{2}, \frac{M-1}{2})$ , comme étant sa propre symétrique. Pour les simulations, nous avons travaillé avec  $M$  pair.

La figure 6.3 représente ces différents exemples de choix de sous-bandes pour  $M = k'N = 12$ . On a donc :  $K'_1 = \frac{M}{2} = 6$  en utilisant  $\mathcal{L}^{(1)}$ ,  $K'_2 = M = 12$  avec  $\mathcal{L}^{(2)}$ ,  $K'_3 = (\frac{M}{2})^2 = 36$  avec  $\mathcal{L}^{(3)}$  et  $K'_4 = \frac{M^2}{2} = 72$  avec  $\mathcal{L}^{(4)}$ .

Les ensembles  $\mathcal{L}^{(4)}$  consistent à regrouper deux à deux les sous bandes symétriques, comme dans l'exemple 8. Il s'agit des ensembles minimaux que l'on puisse considérer garantissant une reconstruction réelle.

### 6.1.8-b) Estimation de l'écart type du bruit

Dans ce qui précède, nous avons supposé l'écart-type du bruit blanc gaussien  $\sigma$  connu ce qui, dans la pratique, n'est pas toujours le cas. Il existe des méthodes statistiques permettant de déterminer une valeur approchée de  $\sigma$  [Rousseeuw, Croux, 1993]. Nous proposons ici une méthode d'estimation empirique dont le principal avantage est d'utiliser la décomposition calculée lors du débruitage.

Une méthode couramment employée est de calculer la valeur médiane de certains coefficients correspondant aux hautes fréquences puis de multiplier par un certain facteur

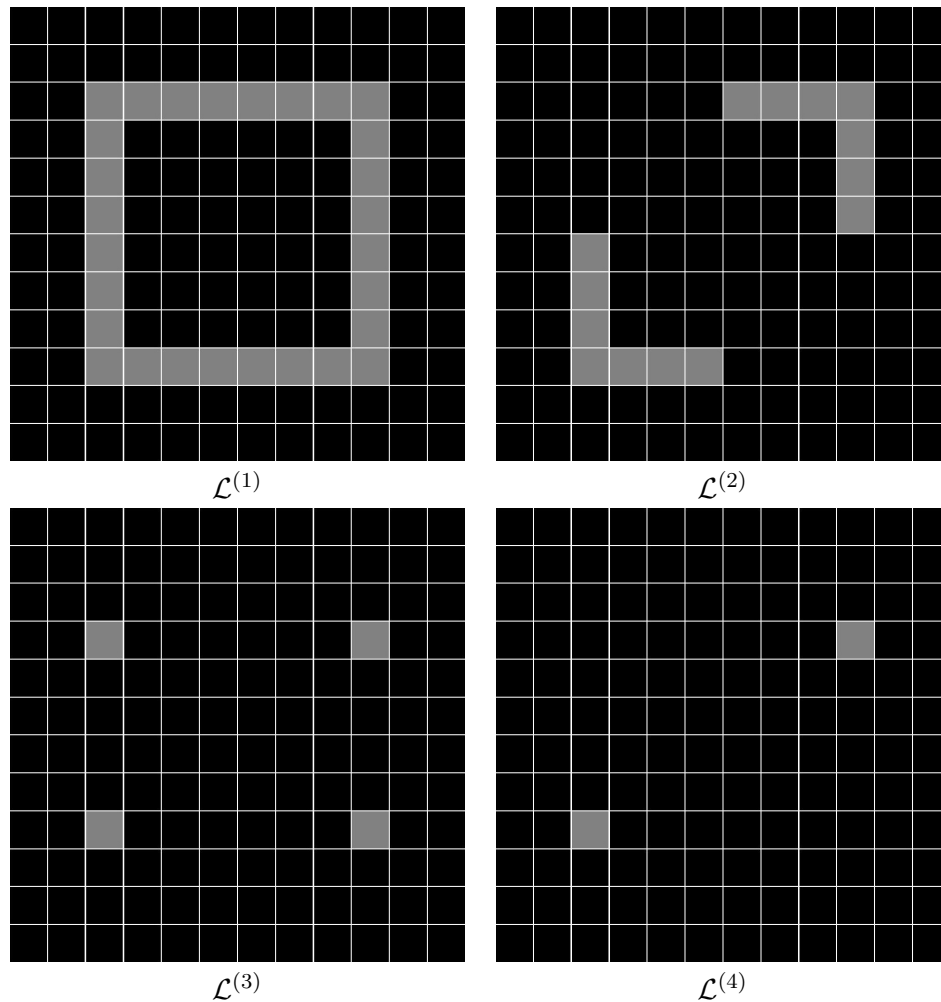


FIGURE 6.3 – Exemples de manières d'apparier des sous-bandes de l'image de la figure 6.2 de manière symétrique. Les sous-bandes représentées en gris sont regroupées.

dépendant de la méthode de décomposition. Par exemple pour l'estimation par ondelettes orthogonales sur une image on peut considérer :  $\frac{\text{médiane}(|H|)}{0.6745}$  où  $H$  représente un ensemble de coefficients de décomposition correspondant aux plus hautes fréquences [Donoho, Johnstone, 1995]. L'utilisation d'un estimateur comme la médiane s'avère robuste dans le cas où les hautes fréquences contiennent une petite proportion de coefficients correspondant au signal parmi ceux représentant le bruit, ce qui est souvent le cas pour des signaux ou des images réels. Nous allons reprendre cette idée pour estimer empiriquement le coefficient multiplicatif, que nous noterons  $\zeta$ , associé à une méthode de décomposition redondante donnée.

La méthode proposée pour évaluer le facteur multiplicatif est la suivante :

1. Calcul de la matrice  $\mathbf{D}$  de décomposition.
2. Pour  $i$  allant de 1 à  $N_{it}$  :
  - Générer un signal de bruit  $b$  pour un  $\sigma$  fixé

- Calculer  $\mathbf{D}b$ , et la médiane  $m$  d'un ensemble de coefficients hautes fréquences
- $v(i) \leftarrow \frac{\sigma}{m}$

3. Facteur multiplicatif empirique :  $\zeta = \text{médiane}(v)$ .

Le nombre d'itérations  $N_{\text{it}}$  n'a, dans la pratique, pas besoin d'être très grand (avec  $N_{\text{it}} = 100$  on a déjà une bonne approximation).

Connaissant le facteur multiplicatif  $\zeta$  associé à une transformation donnée, on peut évaluer l'écart-type  $\sigma$  du bruit de la manière suivante :

1. Effectuer la transformation de l'image à débruiter.
2. Calculer la médiane  $m$  d'un ensemble de coefficients hautes fréquences.
3. L'écart type est alors :  $\sigma_{\text{est}} = \zeta m$ .

**Exemple 9** Nous avons testé cette méthode avec plusieurs bancs de filtres d'analyse, complexes et redondants, définis à la section 3.5.2. Les valeurs obtenues pour différents paramètres apparaissent dans le tableau 6.1. On peut remarquer que les coefficients  $\zeta$  présentent un comportement approximativement linéaire en fonction de  $k'$  pour un  $k$  et un  $N$  donnés.

$k'$	5/4	3/2	7/4	2	9/4	5/2	11/4
$k = 3, N = 8$ ( $k'/\zeta$ )	2,8775 (0,4344)	3,4664 (0,4327)	4,0425 (0,4329)	4,6283 (0,4321)	5,2111 (0,4318)	5,7935 (0,4315)	6,3959 (0,4300)
$k = 3, N = 16$ ( $k'/\zeta$ )	2,9409 (0,4250)	3,5330 (0,4246)	4,1503 (0,4217)	4,7523 (0,4208)	5,3599 (0,4198)	5,9592 (0,4195)	6,5834 (0,4177)
$k = 5, N = 8$ ( $k'/\zeta$ )	2,9268 (0,4271)	3,5330 (0,4246)	4,1199 (0,4248)	4,6837 (0,4270)	5,2905 (0,4253)	5,8809 (0,4251)	6,4892 (0,4238)
$k = 5, N = 16$ ( $k'/\zeta$ )	2,9617 (0,4221)	3,5640 (0,4209)	4,1511 (0,4216)	4,7498 (0,4211)	5,3752 (0,4186)	5,9518 (0,4200)	6,5502 (0,4198)

TABLE 6.1 – Exemple de coefficients  $\zeta$  utilisés pour l'estimation de  $\sigma$ .

### 6.1.9 Simulations

Nous allons évaluer maintenant la qualité du débruitage sur des signaux et des images. Tout d'abord, des résultats de reconstruction de signaux et d'images sont comparés à l'oracle afin de valider les résultats obtenus par minimisation des estimateurs  $\varepsilon_C$  et  $\varepsilon_E$ . Nous présentons ensuite des résultats de débruitage sur des images en utilisant différents bancs de filtres d'analyse/synthèse avec les deux méthodes de type SURE-LET présentées précédemment. La question de l'évaluation de la complexité de ces méthodes est alors abordée, nous verrons que le fait de regrouper certaines sous-bandes permet de réduire les temps de calcul sans trop dégrader les résultats. Enfin, des reconstructions d'images classiques (représentées à la figure 6.4) effectuées par ces méthodes sont comparées entre elles et à plusieurs méthodes classiques de débruitage d'images. Ces comparaisons sont réalisées en terme de PSNR et de qualité visuelle.



Dans la suite, nous appellerons “FB-SURELET-C” la méthode consistant à travailler sur les coefficients après décomposition (liée à l’estimateur  $\varepsilon_C$ ) et “FB-SURELET-E” celle appliquée sur les échantillons après la reconstruction (liée à l’estimateur  $\varepsilon_E$ ).

### 6.1.9-a) Comparaison avec l’oracle

Pour cette première simulation, nous avons travaillé avec différents signaux auxquels un bruit blanc gaussien (d’écart type  $\sigma = 30$ ) a été ajouté. Pour évaluer la qualité des résultats, nous les avons comparés à l’*oracle*, c’est-à-dire la meilleure approximation que l’on puisse obtenir avec les fonctions de filtrage choisies. L’oracle se calcule directement à partir du signal exact que l’on souhaite reconstruire. Si l’on travaille sur les coefficients avant reconstruction la recherche de l’oracle revient à rechercher le vecteur  $\mathbf{a}_{\text{oracle}}$  minimisant la quantité :

$$\phi_C(a_1, \dots, a_K) = \left\| \sum_{k=1}^K a_k \mathbf{F}_k(\mathbf{Y}) - \mathbf{x} \right\|,$$

et si l’on travaille sur les valeurs après reconstruction, on recherche le vecteur  $\mathbf{a}'_{\text{oracle}}$  minimisant :

$$\phi_E(a'_1, \dots, a'_K) = \left\| \sum_{k=1}^K a'_k \mathbf{G}_k(\mathbf{y}) - \mathbf{x} \right\|.$$

Nous avons travaillé ici avec trois signaux différents : le premier est un signal en dent de scie, le deuxième une sinusoïde et le troisième une ligne issue d’une photographie. Nous avons utilisé le banc de filtres d’analyse de la section 3.5.2 avec la fenêtre  $h_{a_1}$  et les paramètres  $k = 3$ ,  $k' = 3/2$  et  $N = 8$ . Pour la reconstruction, nous avons employé le banc de filtres de synthèse obtenu après optimisation avec la fonction  $\widetilde{\mathcal{J}}_{\text{ts}}$ . Enfin, nous avons utilisé les ensembles de coefficients  $\mathcal{L}^{(4)}$  pour l’étape de filtrage.

Le tableau 6.2 présente les résultats médians en SNR (sur les coefficients avant reconstruction), obtenus pour 20 réalisations différentes de bruit, en utilisant la méthode FB-SURELET-C et avec l’oracle (calculé en minimisant  $\phi_C$ ) pour ces trois signaux. On voit que l’écart entre le résultat donné par l’oracle et la reconstruction proposée décroît au fur et à mesure que la longueur du signal augmente.

	Longueur : 1024			Longueur : 2048		
	Signal 1	Signal 2	Signal 3	Signal 1	Signal 2	Signal 3
Signal bruité	13, 9	14, 4	13, 6	13, 9	14, 4	13, 7
FB-SURELET-C	18, 1	20, 5	17, 5	18, 1	20, 6	16, 9
Oracle	18, 5	20, 7	18, 1	18, 3	20, 8	17, 2

TABLE 6.2 – SNR des reconstructions avec FB-SURELET-C et avec l’oracle pour différent signaux perturbés par un bruit blanc gaussien ( $\sigma = 30$ ).

Le tableau 6.3 présente les résultats médians en SNR, obtenus pour 20 réalisations différentes de bruit, en utilisant la méthode FB-SURELET-E et avec l’oracle (calculé en minimisant  $\phi_E$ ). On peut voir que les résultats sont à nouveau très proches de l’oracle, et ce d’autant plus que le nombre de points dans le signal est élevé. La figure 6.5 représente



(a)



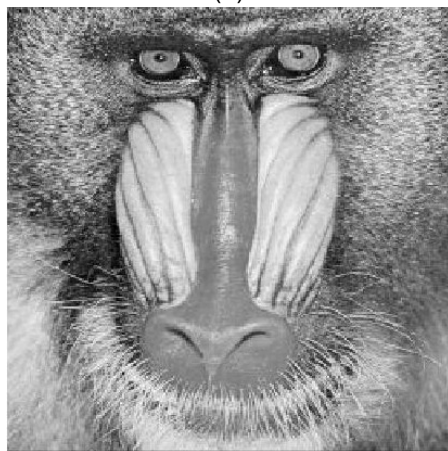
(b)



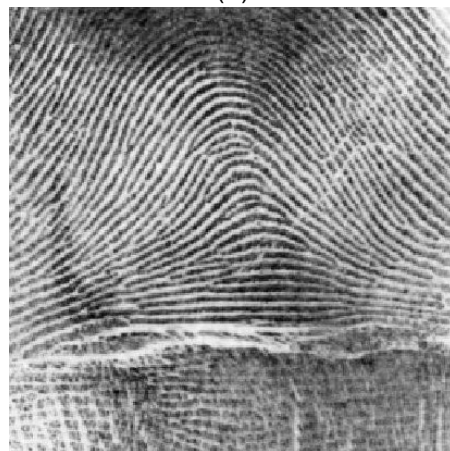
(c)



(d)



(e)



(f)

FIGURE 6.4 – Images utilisées pour les simulations : (a) Lena, (b) Barbara, (c) Boat, (d) Cameraman, (e) Baboon et (f) Fingerprint.

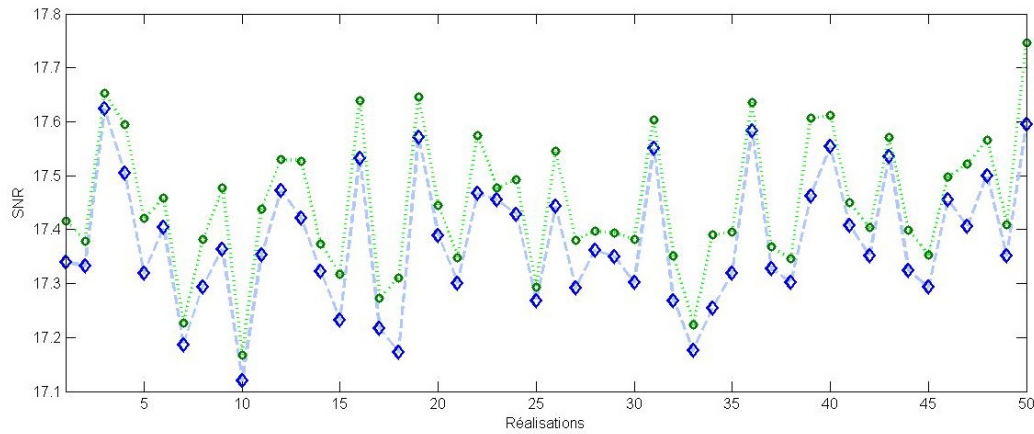


FIGURE 6.5 – SNR après débruitage du signal de la figure 6.6 pour plusieurs réalisations différentes de bruit ( $\sigma = 30$ ) par FB-SURELET-E (diamants bleus) et l'oracle (cercles verts).

les résultats de débruitage pour 50 réalisations de bruit sur le signal issu d'une ligne de photographie de longueur 1024 échantillons, on voit que l'estimateur et l'oracle ont des comportements (croissances et décroissances) similaires.

Enfin, la figure 6.6 présente les résultats obtenus avec la méthode FB-SURELET-E sur le signal issu d'une photographie. On voit que la reconstruction est visuellement très proche de ce que donne l'oracle.

	Longueur : 1024			Longueur : 2048		
	Signal 1	Signal 2	Signal 3	Signal 1	Signal 2	Signal 3
Signal bruité	13, 9	14, 4	13, 6	13, 9	14, 4	13, 7
FB-SURELET-E	18, 4	20, 7	17, 9	18, 2	20, 7	17, 4
Oracle	18, 5	20, 9	18, 1	18, 3	20, 8	17, 4

TABLE 6.3 – SNR des reconstructions avec FB-SURELET-E et avec l'oracle pour différent signaux perturbés par un bruit blanc gaussien ( $\sigma = 30$ ).

De la même manière, nous avons comparé sur des images les résultats des méthodes de débruitage par SURE-LET avec les résultats de l'oracle. Pour ce test, nous avons considéré deux images de Lena de taille  $256 \times 256$  et  $512 \times 512$  ainsi qu'un banc de filtres d'analyse de paramètres :  $N = 4$ ,  $k = k' = 3$  et le banc de filtres de synthèse optimisé (avec la fonction de coût  $\tilde{J}_{ts}$ ) associé. Les résultats sont présentés dans le tableau 6.4 pour un écart type du bruit  $\sigma = 20$ . On constate à nouveau que les méthodes par SURE-LET obtiennent des résultats proches de l'oracle et qu'en faisant croître la taille de l'image considérée, l'écart entre SURELET et l'oracle diminue.

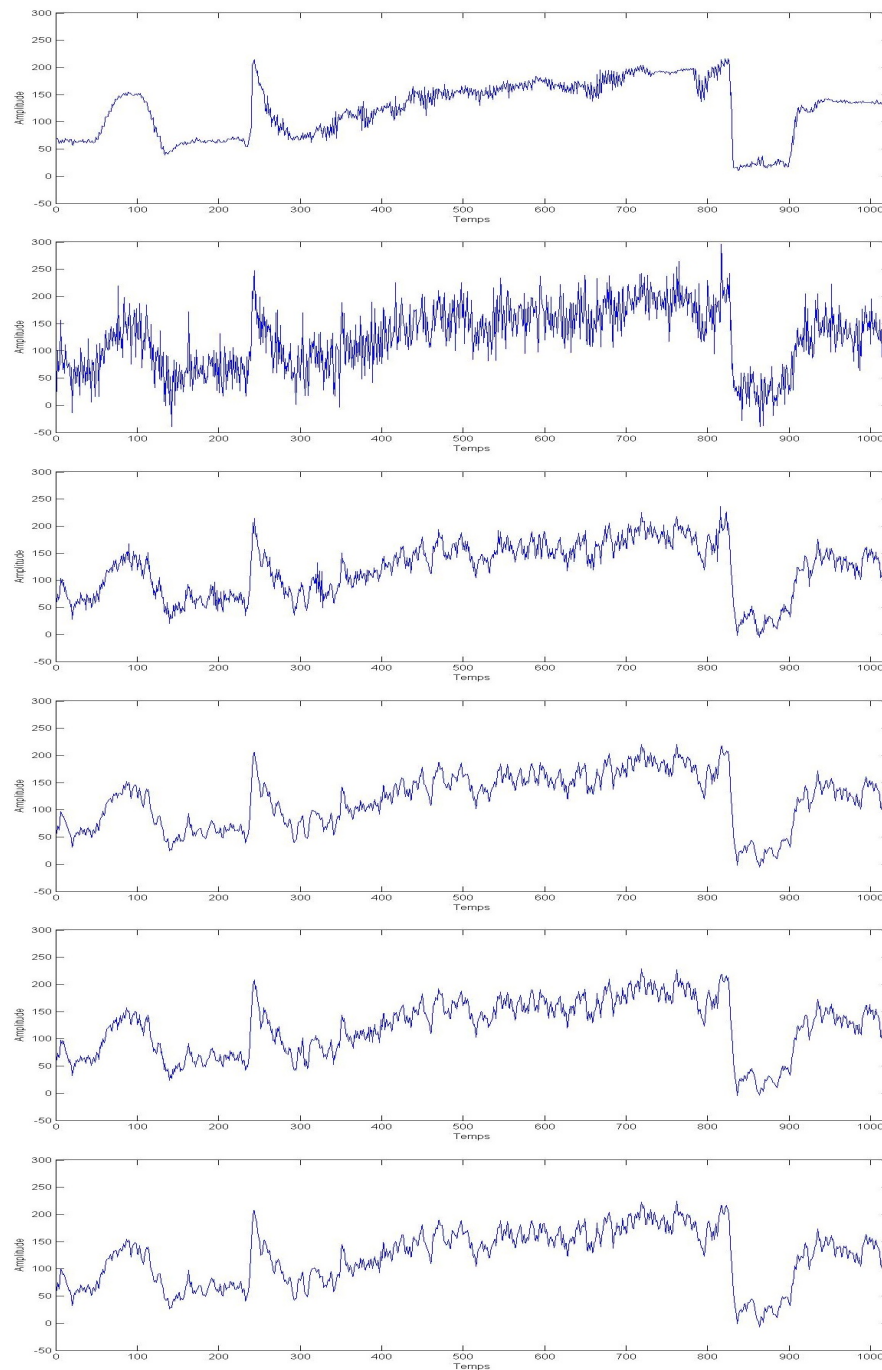


FIGURE 6.6 – Comparaison entre les méthodes proposées et leurs oracles. De haut en bas : signal d'origine, signal bruité (bruit blanc gaussien  $\sigma = 30$ ), signaux débruités par la méthode FB-SURELET-C (17,5dB) puis avec son oracle (18,1dB) et par FB-SURELET-E (18,2dB) puis avec son oracle (18,3dB).

	<i>Lena</i> 256 × 256	<i>Lena</i> 512 × 512
Image bruitée	22, 1	22, 1
FB-SURELET-E	30, 0	31, 3
Oracle-S	30, 2	31, 3
FB-SURELET-C	29, 2	30, 5
Oracle-C	29, 6	30, 7

TABLE 6.4 – Résultats en PSNR de débruitage (pour un bruit  $\sigma = 20$ ) avec FB-SURELET-E et FB-SURELET-C ainsi que leurs oracles respectifs sur l'image Lena.

### 6.1.9-b) Méthodes SURE-LET proposées et banc de filtres

Afin de trouver des couples de bancs de filtres d'analyse/synthèse pouvant donner de bons résultats pour le débruitage, nous avons testé les deux méthodes de débruitage proposées dans ce chapitre avec différents bancs de filtres.

Dans un premier temps, nous avons comparé les résultats de débruitage sur l'image Lena en utilisant le banc de filtres d'analyse de la section 3.5.2 avec les paramètres  $k = 3$ ,  $k' = 7/4$  et  $N = 8$  et les bancs de filtres de synthèse avant et après optimisation (par la méthode du chapitre 4 avec la fonction de coût  $\widetilde{J}_{ts}$ ). Le tableau 6.5 montre les résultats de débruitage obtenus pour différents niveaux de bruit avec FB-SURELET-E et FB-SURELET-C (dans le tableau, la mention *optim.* signifie qu'un banc de filtres optimisé a été utilisé en synthèse). On constate que le fait d'utiliser un banc de filtres optimisé pour la reconstruction améliore les résultats des deux méthodes. Cette amélioration est plus particulièrement marquée dans le cas où l'on travaille avec  $\varepsilon_C$ . Ce phénomène s'explique en partie par le fait que, en minimisant  $\varepsilon_C$ , l'opération de débruitage ne prend pas en compte la reconstruction tandis que la minimisation de  $\varepsilon_E$  fait intervenir l'opérateur  $\mathbf{R}$  et donc "compense" en partie les imperfections dues aux filtres de synthèse.

$\sigma$	10	20	30	40	50	100
PSNR image bruitée	28, 1	22, 1	18, 6	16, 1	14, 2	8, 1
FB-SURELET-E	32, 1	28, 5	26, 6	25, 2	24, 2	21, 1
FB-SURELET-C	31, 5	27, 2	24, 4	22, 4	20, 4	8, 4
FB-SURELET-E (optim.)	32, 5	28, 9	26, 9	25, 6	24, 5	21, 3
FB-SURELET-C (optim.)	32, 1	28, 0	25, 6	23, 6	21, 7	12, 4

TABLE 6.5 – Rapport signal sur bruit (PSNR) en dB après débruitage de l'image Lena par les méthodes FB-SURELET-E et FB-SURELET-C à l'aide d'un banc de filtre de synthèse avant et après optimisation.

Dans un deuxième temps, nous nous sommes intéressés à l'influence que pouvait avoir la redondance sur la qualité du débruitage. Pour ce faire, nous avons appliqué les deux méthodes de débruitage sur l'image Lena en faisant varier cette fois la redondance  $k'$ . Les bancs de filtres d'analyse sont ceux définis à la section 3.5.2 avec les paramètres  $k = 3$  et  $N = 8$  et les bancs de filtres de synthèse sont obtenus après optimisation par

la fonction de coût  $\widetilde{J}_{ts}$ . On peut voir au tableau 6.6 les résultats de débruitage obtenus pour un niveau de bruit  $\sigma = 30$ . On constate que, comme dans l'exemple précédent, la méthode FB-SURELET-E donne de meilleurs résultats que FB-SURELET-C et que quand la redondance augmente l'écart entre les deux méthodes tend à réduire. Dans ce tableau nous avons également inclus la quantité :

$$\xi = \frac{1}{L'^2} \|\mathbf{DR} - \mathbf{I}_{L'}\|_F^2,$$

où  $\|\cdot\|_F$  est la norme de Frobenius. Cette quantité donne une indication concernant "l'écart à l'orthogonalité" (ou plus précisément d'écart à la bi-orthogonalité) de la transformée utilisée. Notons que d'autres "mesures de l'orthogonalité" ont été proposées par Lightstone [Lightstone *et al.*, 1997] ou par Moreau De Saint-Martin et Siohan [MOREAU de Saint-Martin *et al.*, 1999]. On voit que si cet écart réduit, l'écart entre les deux méthodes tend à réduire également. Ce résultat est conforme à l'intuition donnée par la proposition 6.3 qui montre que dans le cas orthogonal les deux estimateurs sont équivalents. On remarque finalement que le fait d'augmenter  $k'$  améliore les résultats de débruitage des deux méthodes.

$k'$	5/4	3/2	7/4	2
FB-SURELET-C	16, 7	24, 3	25, 5	25, 9
FB-SURELET-E	26, 0	26, 7	26, 9	27, 0
Différence	9, 3	2, 4	1, 4	1, 1
$\xi$	$9, 5 \cdot 10^{-3}$	$2, 4 \cdot 10^{-3}$	$1, 9 \cdot 10^{-3}$	$1, 7 \cdot 10^{-3}$
$k'$	9/4	5/2	11/4	3
FB-SURELET-C	26, 0	25, 7	25, 8	27, 1
FB-SURELET-E	27, 2	26, 8	26, 9	27, 4
Différence	1, 2	1, 1	1, 1	0, 3
$\xi$	$1, 5 \cdot 10^{-3}$	$1, 6 \cdot 10^{-3}$	$1, 4 \cdot 10^{-3}$	$0, 86 \cdot 10^{-3}$

TABLE 6.6 – Rapport signal sur bruit (PSNR) en dB après reconstruction par les deux méthodes proposées pour l'image Lena (pour un bruit  $\sigma = 30$ ), avec  $N = 8$ ,  $k = 3$  et  $k'$  variable.

### 6.1.9-c) Temps de calcul

Nous venons de voir que dans tous les cas de figures la méthode FB-SURELET-C donne de moins bons résultats que FB-SURELET-E. Cependant, la première méthode présente l'avantage d'être beaucoup moins coûteuse en calculs. Comme nous l'avons vu, la méthode FB-SURELET-E agit sur les différentes versions de signaux reconstruits. Avec les notations de la section 6.1.7-a), on écrit l'image débruitée  $\mathbf{G}(\mathbf{y})$  sous la forme de la combinaison linéaire  $\sum_{k=1}^K a_k \mathbf{G}_k(\mathbf{y})$ . FB-SURELET-E implique donc le calcul des reconstructions de  $K$  versions de l'image à débruiter, alors qu'en travaillant directement sur les coefficients après décomposition, on ne réalise qu'une seule reconstruction.

Pour illustrer cette différence de complexité, nous avons mesuré les temps de calculs des méthodes FB-SURELET-E et FB-SURELET-C sur une image de taille  $256 \times 256$ . Pour

l'analyse, nous avons utilisé un banc de filtres de paramètres  $k = k' = 3$  et  $N = 4$ , et pour la synthèse le banc de filtres optimisé correspondant. Le débruitage (en utilisant Matlab 7 sur un ordinateur doté d'un processeur Intel Core 2 T7400 à 2.16GHz et 2Go de RAM) avec la méthode FB-SURELET-E a nécessité 55 secondes contre moins de 6 secondes avec FB-SURELET-C.

À la section 6.1.8-a), nous avons vu différents schémas possibles pour regrouper les sous-bandes entre elles. Le nombre de reconstructions à réaliser avec la méthode FB-SURELET-E est lié au nombre de paramètres à estimer, qui dépend lui-même du nombre de groupes de sous-bandes considérés. Il est donc possible de diminuer la complexité de la méthode FB-SURELET-E en réduisant le nombre de groupes de sous-bandes. Le tableau 6.7 montre les temps de calcul et les PSNR médians (pour 20 réalisations) obtenus avec les différents ensembles  $\mathcal{L}^{(j)}$  définis à la section 6.1.8-a), sur l'image Lena de taille  $256 \times 256$  avec  $\sigma = 30$ . On constate que le fait de regrouper certaines sous-bandes entre elles permet de réduire substantiellement le temps de calcul pour la méthode FB-SURELET-E. Le gain pour FB-SURELET-C est moins important, ceci s'explique par le fait que cette méthode ne nécessite qu'une seule reconstruction (qui est l'étape la plus coûteuse en temps de calcul).

On voit également que le résultat en PSNR du débruitage avec ces différents groupes de sous-bandes sont très proches (il faut regarder deux décimales après la virgule pour constater une amélioration graduelle à mesure que  $K$  augmente).

Groupes de sous-bandes	$\mathcal{L}^{(1)}$	$\mathcal{L}^{(2)}$	$\mathcal{L}^{(3)}$	$\mathcal{L}^{(4)}$
Méthode FB-SURELET-C				
Temps de calcul	3s	4s	5s	6s
PSNR	26,96	26,97	27,02	27,04
Méthode FB-SURELET-E				
Temps de calcul	7s	12s	29s	55s
PSNR	28,02	28,03	28,03	28,04

TABLE 6.7 – Temps de calcul et PSNR pour les méthodes FB-SURELET-C et FB-SURELET-E en utilisant différents groupes de sous-bandes.

#### 6.1.9-d) Comparaison avec d'autres méthodes de débruitage

Les méthodes de débruitage présentées ont été testées sur plusieurs images classiques. Pour évaluer la qualité de ces méthodes nous les avons comparées aux résultats obtenus avec le débruitage utilisant les *Curvelets*<sup>1</sup>, la méthode *SureShrink* de Donoho et Johnstone [Donoho, Johnstone, 1995], puis la méthode de *bivariate shrinkage* (BiShrink) de Şendur et Selesnick [Şendur, Selesnick, 2002]<sup>2</sup>, et enfin la méthode de débruitage par SURE-LET avec des ondelettes orthonormales et une étude des dépendances entre échelles (*OWT SURE-LET*) proposée par Luisier et Blu<sup>3</sup> [Luisier, Blu, 2007].

1. Toolbox Curvelab disponible sur le site : [www.curvelet.org/](http://www.curvelet.org/)

2. Toolbox disponible sur le site : <http://taco.poly.edu/WaveletSoftware/>

3. Toolbox disponible sur le site : <http://bigwww.epfl.ch/demo/suredenoising/>



Nous avons également considéré deux autres méthodes de type SURE-LET, similaires à FB-SURELET-C, mais utilisant des ondelettes orthonormales réelles avec deux fonctions non linéaires différentes : celle proposée par Pesquet et Leporini [Pesquet, Leporini, 1997] :

$$\tau_3(x) = \left( \tanh\left(\frac{x + \chi}{\beta}\right) - \tanh\left(\frac{x - \chi}{\beta}\right) \right) x,$$

où  $\chi = \frac{7}{2}$  et  $\beta = \frac{7}{4}$ , conduisant à la méthode notée *SURE-LET (sigmoïde)*, et celle utilisée par Blu et Luisier [Blu, Luisier, 2007] :  $\tau_2$  telle que définie en (6.12) avec les paramètres  $m_0 = 8$  et  $\alpha = 3$ , méthode notée *SURE-LET (exponentielle)*. Les méthodes FB-SURELET-E et FB-SURELET-C ont été testées avec deux couples de bancs de filtres d'analyse/synthèse construits à partir du banc de filtres d'analyse de la section 3.5.2 avec les paramètres  $k = 3$ ,  $k' = 3$  et  $N = 4$  pour le premier (noté  $(3, 3, 4)$ ) et les paramètres  $k = 5$ ,  $k' = 5$  et  $N = 4$  pour le deuxième (noté  $(5, 5, 4)$ ).

Dans les tableaux 6.8 et 6.9, nous avons reporté les résultats médians du PSNR après débruitage, pour 20 réalisations de bruit pour chacune de ces méthodes. Cela permet de constater que la méthode FB-SURELET-E proposée dans ce chapitre donne des résultats en règle générale légèrement meilleurs que ceux obtenus par la méthode OWT SURE-LET, et presque systématiquement meilleurs que ceux obtenus par SureShrink, BiShrink, Curvelet et les autres méthodes SURE-LET. En particulier, on voit que le fait d'utiliser une transformée complexe redondante plutôt qu'une transformation orthogonale réelle (méthodes *SURE-LET (exponentielle)*) permet d'améliorer sensiblement les résultats. On remarque que la méthode FB-SURELET-E semble donner de bons résultats pour de petits niveaux de bruits, en revanche, pour des niveaux de bruits importants les résultats sont de moins bonne qualité. Pour des niveaux faibles de bruit la méthode FB-SURELET-C donne des résultats comparables aux autres méthodes mais semble inadaptée au débruitage d'images fortement perturbées.

Il est intéressant de noter qu'en faisant varier le facteur  $\alpha$  utilisé dans l'équation (6.12), fixé ici à  $\frac{3}{2}$ , on peut adapter la méthode FB-SURELET-E à des niveaux de bruits plus élevés. En prenant  $\alpha = 3$  et en considérant un bruit d'écart type  $\sigma = 100$ , on obtient les résultats de débruitage pour la méthode FB-SURELET-E  $(3, 3, 4)$  présentés dans le tableau 6.10. On voit que les résultats sur les différentes images de test sont équivalents à ceux donnés par les meilleures méthodes pour un niveau de bruit élevé. Le choix des paramètres de la fonction de seuillage non linéaire constitue donc un facteur important pour les résultats. Notons, par ailleurs, que les résultats des méthodes *SURE-LET (sigmoïde)* et *SURE-LET (exponentielle)* sont très proches (la différence en PSNR se faisant souvent deux décimales après la virgule), ceci montre que plusieurs choix pour la fonction non linéaire de seuillage sont possibles. Une perspective pour améliorer les résultats pourrait être de rechercher d'autres types de fonctions de seuillage non linéaire et s'adaptant au niveau de bruit.

On peut aussi remarquer que, suivant les images, le meilleur débruitage avec la méthode FB-SURELET-E ne sera pas obtenu avec le même banc de filtres. Le cas  $(5, 5, 4)$  semble bien adapté pour l'image Fingerprint alors que le cas  $(3, 3, 4)$  donne de meilleurs résultats sur les autres images. On remarque également que le cas  $(3, 3, 4)$  semble mieux adapté pour la méthode FB-SURELET-C avec des niveaux de bruits élevés. Suivant le type d'image, le niveau de bruit et la méthode employée, le choix des filtres présente donc une importance significative.



Une autre indication intéressante est apportée par l'examen visuel des résultats de reconstructions. La figure 6.7 présente ces résultats pour l'image Barbara avec  $\sigma = 30$  et la figure 6.9 les présente pour l'image Cameraman avec  $\sigma = 50$ . La figure 6.8 illustre les différentes reconstructions de détails sur l'image Barbara. On voit que les textures orientées, comme celles présentes sur le foulard, sont très bien reconstruites par FB-SURELET-E. Ceci est dû à la nature des bancs de filtres que nous avons employés dans cette application, qui est bien adaptée à la représentation des structures directionnelles. On peut voir sur la figure 6.9, que la méthode FB-SURELET-E ne semble pas introduire autant d'artéfacts le long des contours, bien marqués, du manteau. En revanche dans le ciel des artéfacts directionnels apparaissent qui sont dûs à la décomposition utilisée. Sur l'image de détail, on voit qu'à nouveau la reconstruction par FB-SURELET-E (3, 3, 4) a permis de bien retrouver certaines informations directionnelles comme les colonnes du bâtiment (dans le fond de l'image).

Il semble donc que cette méthode de débruitage soit bien adaptée à des images présentant des textures orientées et qu'elle donne de moins bons résultats dans des zones de couleur homogène.

## 6.2 Filtrage directionnel dans des images

Nous allons maintenant nous intéresser à un autre type de méthodes d'élimination du bruit que nous avons employé pour certaines applications qui seront présentées au chapitre 7. Dans ces applications des perturbations structurées, par exemples des directions indésirables, sont visibles. Un débruitage utilisant les méthodes statistiques présentées dans la partie précédente, ne permet pas d'éliminer ces bruits structurés et nous allons donc proposer des méthodes plus empiriques pour réaliser ce type de débruitage. Nous allons d'abord expliquer comment estimer simplement des directions de référence dans une image puis nous expliquerons comment ne garder que les orientations les plus énergétiques et enfin comment améliorer cette méthode pour lui donner plus de souplesse dans le cas de mélanges de directions.

### 6.2.1 Estimation des directions

Le but de cette section est de travailler sur des images présentant une forte anisotropie, commençons donc par proposer un modèle simple pour représenter de telles images. Considérons l'image définie par la fonction :

$$\forall (x, y) \in \mathcal{E}, \quad f(x, y) = \sin(ax + by + \nu)$$

avec  $(a, b) \in \mathbb{R}^2$ ,  $\nu \in [0, 2\pi]$  et  $\mathcal{E}$  un domaine de  $\mathbb{R}^2$ . Elle est composée de bandes disposées suivant l'orientation définie par la direction du vecteur  $(-b, a)$ . Autrement dit, pour connaître la direction dominante d'une image il suffit d'être capable d'approximer le vecteur  $(-b, a)$ , ou de manière équivalente le vecteur normal  $(a, b)$ .

En observant le module de la transformée de Fourier de ce type d'image, on voit deux pics symétriques apparaître dans la direction du vecteur  $(a, b)$ . Pour estimer la direction apparente, il suffit donc de rechercher ces maxima et d'en calculer la direction.

$\sigma$	10	20	30	40	50	100
PSNR image bruitée	28,1	22,1	18,6	16,1	14,2	8,1
Lena						
Curvelet	33,0	29,3	27,3	25,9	24,8	21,0
SureShrink	33,0	28,3	26,3	25,2	24,2	21,4
BiShrink	32,8	29,0	26,9	25,5	24,6	21,8
SURE-LET (sigmoïde)	31,9	28,1	26,2	25,0	24,1	21,6
SURE-LET (exponentielle)	32,0	28,1	26,2	25,0	24,1	21,6
FB-SURELET-C (5,5,4)	33,0	29,3	27,2	25,8	24,6	18,5
FB-SURELET-C (3,3,4)	33,0	29,3	27,1	25,5	24,3	20,0
OWT SURE-LET	32,9	29,0	27,0	25,8	24,8	<b>21,9</b>
FB-SURELET-E (5,5,4)	33,6	29,8	27,6	26,2	25,0	21,3
FB-SURELET-E (3,3,4)	<b>33,8</b>	<b>30,0</b>	<b>28,0</b>	<b>26,4</b>	<b>25,3</b>	21,7
Barbara						
Curvelet	31,2	27,2	25,1	23,6	22,6	19,9
SureShrink	30,2	25,9	23,7	22,5	21,7	19,5
BiShrink	30,1	26,6	24,4	23,1	22,2	20,0
SURE-LET (sigmoïde)	30,2	25,8	23,7	22,6	21,8	19,9
SURE-LET (exponentielle)	30,3	25,8	23,7	22,5	21,8	20,0
FB-SURELET-C (5,5,4)	31,6	27,7	25,5	23,9	22,6	16,6
FB-SURELET-C (3,3,4)	31,6	27,6	25,4	23,7	22,3	18,7
OWT SURE-LET	30,6	26,3	24,2	23,0	22,2	20,1
FB-SURELET-E (5,5,4)	32,0	28,2	26,1	24,6	<b>23,6</b>	19,9
FB-SURELET-E (3,3,4)	<b>32,2</b>	<b>28,3</b>	<b>26,2</b>	<b>24,7</b>	<b>23,6</b>	<b>20,3</b>
Boat						
Curvelet	31,7	28,1	26,1	24,9	24,0	<b>21,3</b>
SureShrink	31,2	27,4	25,4	24,3	23,4	20,8
BiShrink	31,8	27,9	25,9	24,7	23,8	<b>21,3</b>
SURE-LET (sigmoïde)	31,3	27,5	25,6	24,4	23,7	21,4
SURE-LET (exponentielle)	31,3	27,5	25,6	24,4	23,6	21,4
FB-SURELET-C (5,5,4)	31,9	28,1	26,1	24,7	23,7	18,3
FB-SURELET-C (3,3,4)	32,0	28,1	26,0	24,6	23,4	20,0
OWT SURE-LET	32,0	28,2	26,3	25,0	24,2	20,6
FB-SURELET-E (5,5,4)	32,3	28,5	26,5	25,1	<b>24,0</b>	20,9
FB-SURELET-E (3,3,4)	<b>32,6</b>	<b>28,7</b>	<b>26,7</b>	<b>25,3</b>	<b>24,3</b>	21,2

TABLE 6.8 – Rapport signal sur bruit (PSNR) en dB après débruitage par les différentes méthodes sur les images Lena, Barbara et Boat.

$\sigma$	10	20	30	40	50	100
PSNR image bruitée	28, 1	22, 1	18, 6	16, 1	14, 2	8, 1
Cameraman						
Curvelet	32, 1	28, 5	26, 5	25, 2	24, 2	21, 0
SureShrink	31, 4	27, 5	25, 4	24, 1	23, 0	20, 2
BiShrink	32, 5	28, 4	26, 3	25, 0	24, 0	21, 2
SURE-LET (sigmoïde)	31, 6	27, 5	25, 5	24, 1	23, 2	20, 7
SURE-LET (exponentielle)	31, 6	27, 5	25, 5	24, 2	23, 2	20, 7
FB-SURELET-C (5,5,4)	32, 2	28, 3	26, 3	24, 8	23, 7	18, 1
FB-SURELET-C (3,3,4)	32, 1	28, 2	26, 2	24, 7	23, 5	19, 5
OWT SURE-LET	32, 4	28, 6	26, 6	25, 2	24, 2	<b>21, 3</b>
FB-SURELET-E (5,5,4)	32, 8	28, 9	26, 9	25, 3	24, 2	20, 6
FB-SURELET-E (3,3,4)	<b>32, 8</b>	<b>29, 0</b>	<b>27, 0</b>	<b>25, 6</b>	<b>24, 4</b>	20, 8
Baboon						
Curvelet	29, 3	25, 1	23, 4	22, 5	21, 8	20, 3
SureShrink	29, 5	25, 2	23, 4	22, 4	21, 7	20, 0
BiShrink	29, 9	25, 5	23, 5	22, 4	21, 8	20, 4
SURE-LET (sigmoïde)	29, 6	25, 5	23, 7	22, 8	22, 2	20, 6
SURE-LET (exponentielle)	29, 6	25, 5	23, 7	22, 8	22, 2	20, 6
FB-SURELET-C (5,5,4)	30, 0	25, 7	23, 7	22, 5	21, 4	15, 4
FB-SURELET-C (3,3,4)	30, 0	25, 7	23, 7	22, 5	21, 7	19, 1
OWT SURE-LET	29, 9	25, 7	23, 9	22, 9	22, 3	<b>20, 6</b>
FB-SURELET-E (5,5,4)	30, 2	26, 2	24, 2	23, 1	22, 3	19, 6
FB-SURELET-E (3,3,4)	<b>30, 3</b>	<b>26, 2</b>	<b>24, 3</b>	<b>23, 2</b>	<b>22, 4</b>	20, 2
Fingerprint						
Curvelet	29, 0	25, 4	23, 6	22, 3	21, 4	18, 5
SureShrink	29, 4	25, 1	22, 9	21, 5	20, 4	17, 1
BiShrink	29, 3	24, 8	22, 5	21, 1	20, 0	17, 2
SURE-LET (sigmoïde)	29, 5	25, 1	22, 9	21, 5	20, 4	17, 7
SURE-LET (exponentielle)	29, 5	25, 1	22, 9	21, 5	20, 4	17, 7
FB-SURELET-C (5,5,4)	30, 1	26, 1	24, 0	22, 7	21, 7	17, 0
FB-SURELET-C (3,3,4)	30, 1	26, 0	23, 8	22, 3	21, 1	17, 2
OWT SURE-LET	29, 7	25, 5	23, 4	21, 9	20, 8	18, 3
FB-SURELET-E (5,5,4)	<b>30, 4</b>	<b>26, 4</b>	<b>24, 5</b>	<b>23, 2</b>	<b>22, 1</b>	<b>19, 0</b>
FB-SURELET-E (3,3,4)	<b>30, 4</b>	<b>26, 4</b>	<b>24, 5</b>	23, 0	21, 9	18, 5

TABLE 6.9 – Rapport signal sur bruit (PSNR) en dB après débruitage par les différentes méthodes sur les images Cameraman, Baboon et Fingerprint.



FIGURE 6.7 – (a) Image originale (Barbara), (b) image bruitée ( $\sigma = 30$ ), (c) image débruitée avec Curvelet, (d) image débruitée avec BiShrink, (e) image débruitée avec OWT SURE-LET et (f) image débruitée avec FB-SURELET-E.

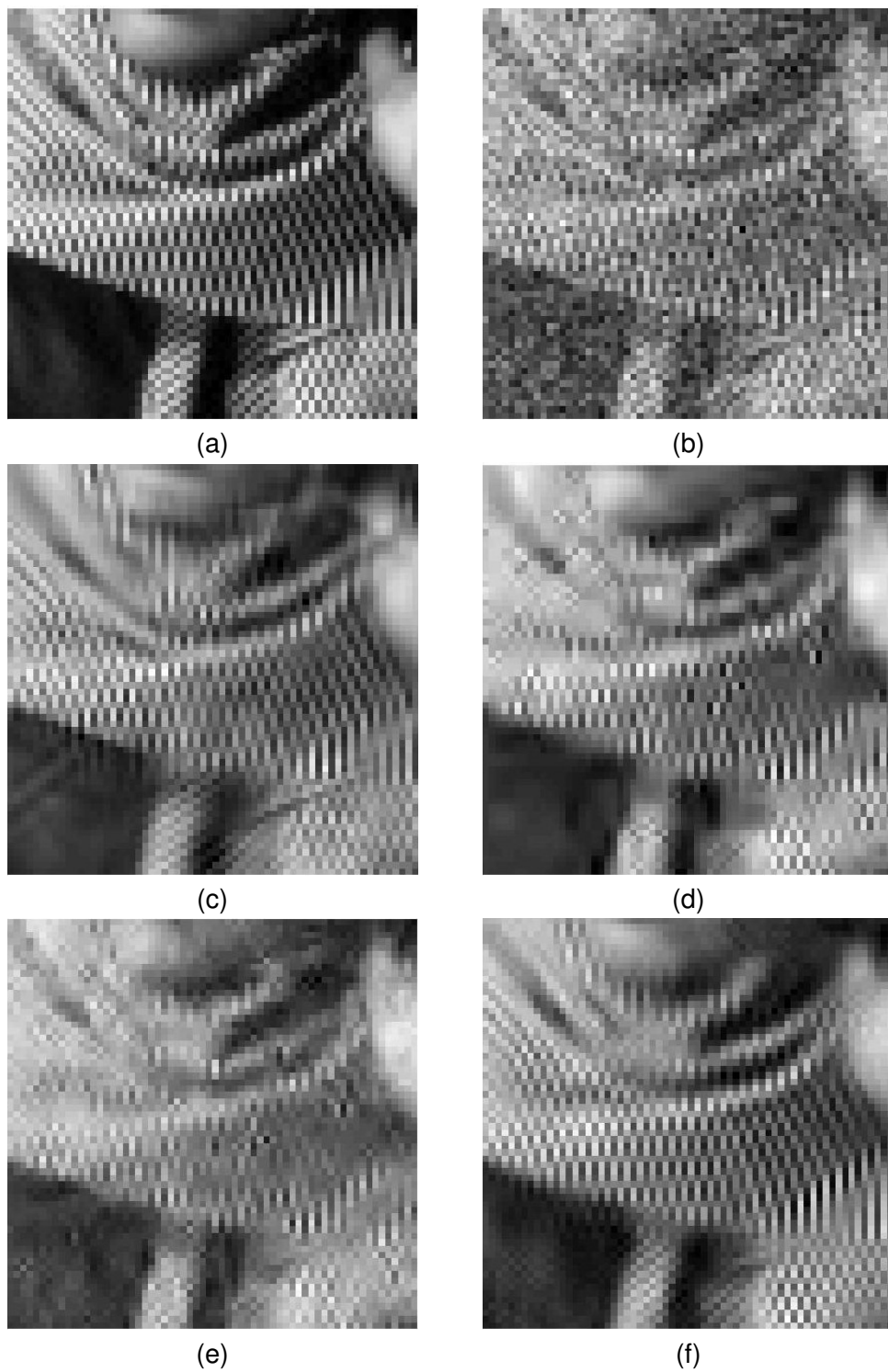


FIGURE 6.8 – Détail de (a) l'image originale (Barbara), (b) l'image bruitée ( $\sigma = 30$ ), (c) l'image débruitée avec Curvelet, (d) l'image débruitée avec BiShrink, (e) l'image débruitée avec OWT SURE-LET et de (f) l'image débruitée avec FB-SURELET-E.

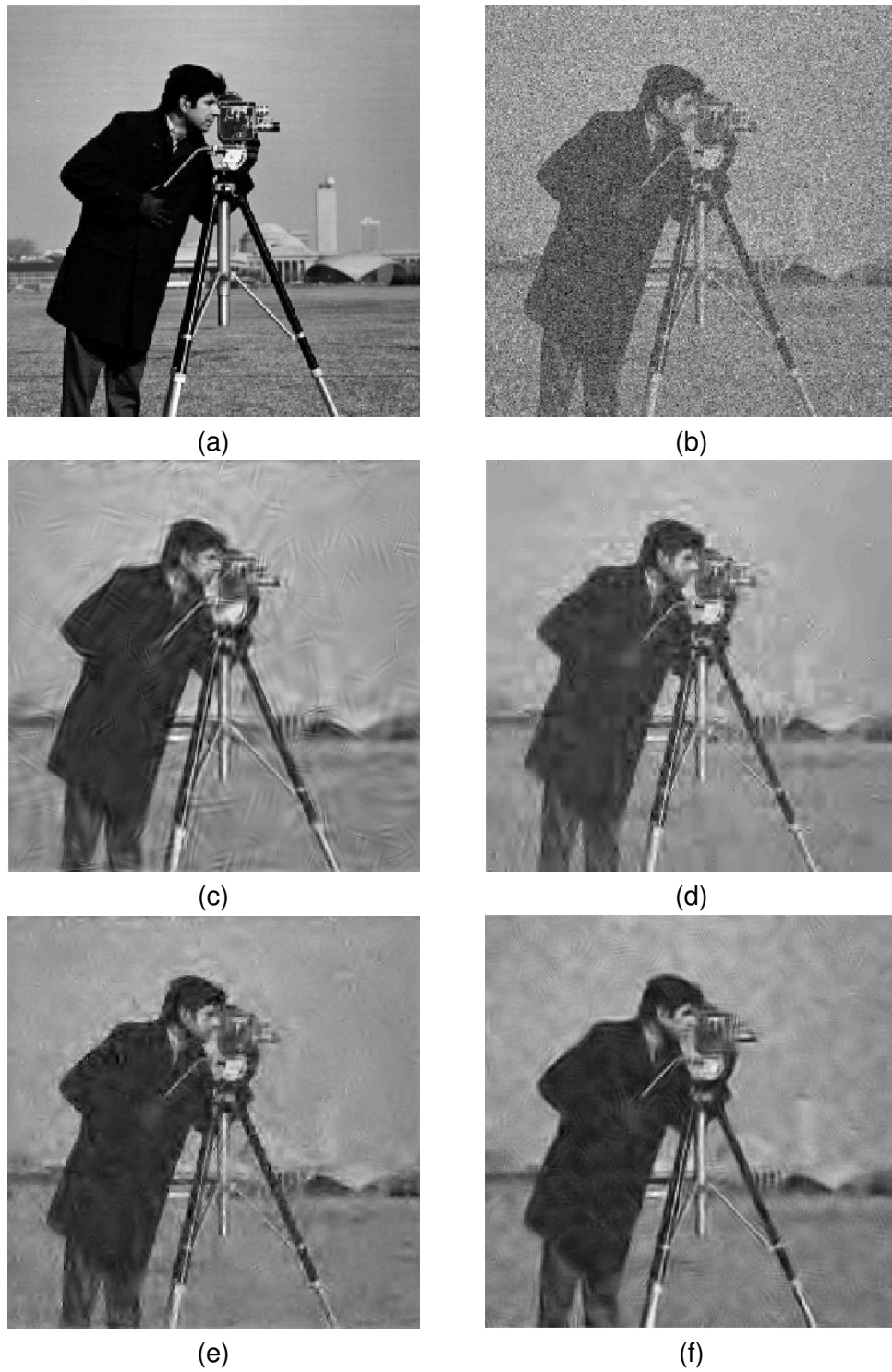


FIGURE 6.9 – (a) Image originale (Cameraman), (b) image bruitée ( $\sigma = 50$ ), (c) image débruitée avec Curvelet, (d) image débruitée avec BiShrink, (e) image débruitée avec OWT SURE-LET et (f) image débruitée avec FB-SURELET-E.

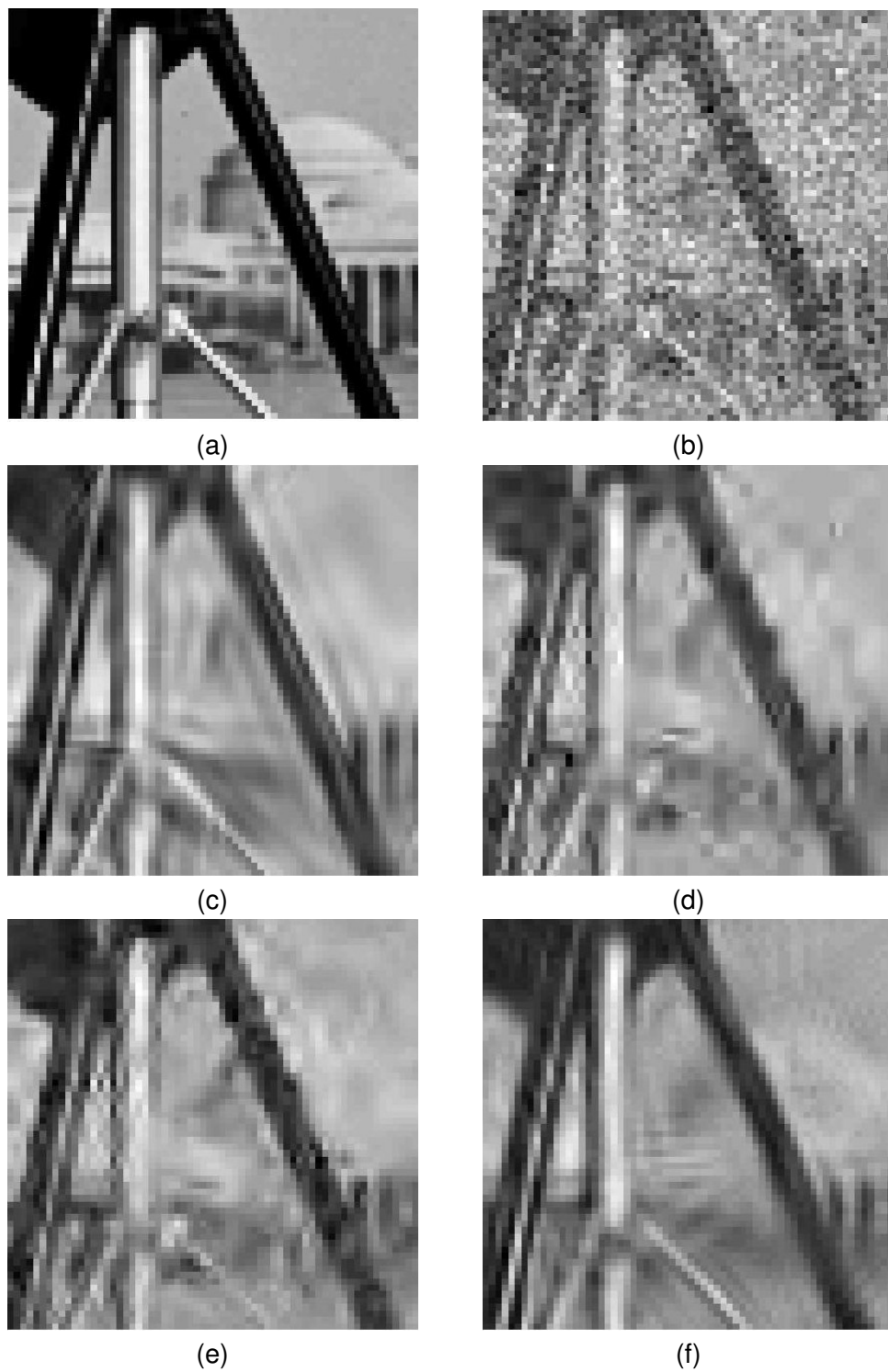


FIGURE 6.10 – Détail de (a) l'image originale (Cameraman), (b) l'image bruitée ( $\sigma = 50$ ), (c) l'image débruitée avec Curvelet, (d) l'image débruitée avec BiShrink, (e) l'image débruitée avec OWT SURE-LET et de (f) l'image débruitée avec FB-SURELET-E.

Image	Lena	Barbara	Boat	Cameraman	Baboon	Fingerprint
FB-SURELET-E (3,3,4)	22, 2	20, 4	21, 5	21, 3	20, 6	18, 5

TABLE 6.10 – Rapport signal sur bruit (PSNR) en dB après débruitage des différentes images de test avec FB-SURELET-E (3,3,4) et  $\alpha = 3$  pour un niveau de bruit  $\sigma = 100$ .

**Remarque 6.4** Nous avons choisi de travailler avec des transformées fréquentielles complexes car elles fournissent, comme nous venons de le voir, un moyen simple pour séparer les directions. Avec une transformation réelle de type TCD le résultat d'une estimation de direction par recherche du maximum dans le domaine fréquentiel peut être ambiguë.

Pour illustrer cette remarque, prenons deux images directionnelles  $f(x, y) = \sin(ax + by + \nu)$  et  $g(x, y) = \sin(ax - by + \nu)$  définies sur le domaine  $[-T, T]^2$ , avec  $T \in \mathbb{R}_+^*$ . Ces images présentent des directions différentes. Considérons des transformées en cosinus continues de ces images. Pour  $f$  on obtient :

$$\begin{aligned} \mathcal{T}(f)(u, v) &= \int_{-T}^T \int_{-T}^T \cos(ux) \cos(vy) \sin(ax + by + \nu) dx dy \\ &= T^2 \sin(\nu) (\text{sinc}(T(u + a)) + \text{sinc}(T(u - a))) (\text{sinc}(T(v + b)) + \text{sinc}(T(b - v))), \end{aligned}$$

où pour tout  $x \in \mathbb{R}$ ,  $\text{sinc}(x) = \frac{\sin(x)}{x}$  est le sinus cardinal (prolongé par continuité en 0).

De même pour  $g$ , on a :

$$\mathcal{T}(g)(u, v) = T^2 \sin(\nu) (\text{sinc}(T(u + a)) + \text{sinc}(T(u - a))) (\text{sinc}(T(v - b)) + \text{sinc}(T(b + v))).$$

On remarque que  $\mathcal{T}(f)(u, v) = \mathcal{T}(g)(u, v)$ . En particulier si le maximum de  $|\mathcal{T}(f)|$  est atteint pour un certain couple  $(u, v)$ , alors  $|\mathcal{T}(g)|$  est aussi maximale en  $(u, v)$ . On voit donc que, dans le cas continu, une transformée en cosinus ne permet pas forcément de séparer deux directions différentes.

Dans le cadre d'une transformée en cosinus discrète on observe un phénomène similaire. La figure 6.11 représente deux images correspondant à des orientations de  $\frac{\pi}{3}$  et  $-\frac{\pi}{3}$  ainsi que les modules de leurs TCD et TFD associées. On voit que la TCD atteint son maximum (en valeur absolue) au même endroit dans les deux cas alors que les pics, visibles avec la TFD, décrivent bien deux directions différentes.

En prenant une combinaison linéaire de fonctions de ce type :  $f_k(x, y) = \sin(a_k x + b_k y + \nu_k)$ , où  $\nu_k \in [0, 2\pi]$  et  $(a_k, b_k) \in \mathbb{R}^2$  avec des vecteurs  $(a_k, b_k)$  non colinéaires, on peut construire une image présentant des mélanges de directions :

$$f(x, y) = \sum_{k=1}^K \alpha_k f_k(x, y), \quad (\alpha_k)_{1 \leq k \leq K} \in \mathbb{R}^K.$$

Dans ce cas, la direction dominante sera celle du vecteur  $(a_{k_0}, b_{k_0})$  associé à la plus grande valeur de  $\alpha_{k_0}$ . Une recherche, dans le domaine fréquentiel, du coefficient de plus grand module permet encore d'isoler la direction dominante. La valeur du module du couple de coefficients de plus grand module sera notée  $\alpha_1$  dans la suite. On pourra dire que  $\alpha_1$  représente le *poids de la direction principale*.



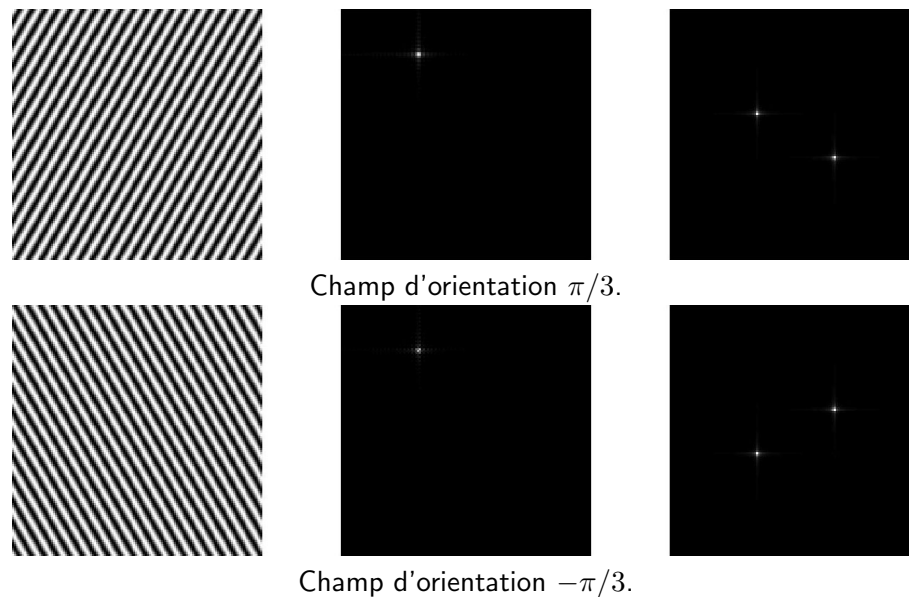


FIGURE 6.11 – Deux images directionnelles avec les transformées en cosinus et de Fourier associées.

Pour caractériser d'autres directions de moindre importance, il suffit de refaire une recherche des coefficients de plus grand module dans l'image transformée, en omettant les coefficients déjà identifiés précédemment ainsi que les coefficients dans des directions proches de celles déjà identifiées. La valeur du module du deuxième couple de coefficients de plus grand module sera notée  $o_2$  dans la suite. On dira que  $o_2$  est le *poids de la direction secondaire*.

### 6.2.2 Sélection de direction

Après avoir vu comment déterminer les différentes directions présentes dans les données, voyons comment utiliser les transformées à recouvrement introduites au chapitre 3 pour sélectionner localement les directions et ainsi réaliser l'opération de filtrage directionnel.

La première étape est, bien entendu, la recherche de la direction à conserver. Dans le cas où l'on veut préserver la direction dominante on peut utiliser la méthode précédente, mais on peut également imaginer d'autres stratégies (par exemple ne conserver que la direction secondaire) suivant le type d'application (un exemple où l'on ne garde pas forcément la direction principale sera décrit à la section 7.1.4-b)). Enfin, cette étape de recherche et d'identification peut être supprimée si l'on dispose d'une information *a priori* sur la direction que l'on souhaite conserver, cela sera par exemple le cas pour l'application de la section 7.2.

La deuxième étape consiste à ne conserver que les coefficients correspondant à la direction d'intérêt et éventuellement des directions proches. On peut imaginer plusieurs méthodes pour réaliser cette opération. La figure 6.12 présente 3 *formes* élémentaires que l'on peut considérer. Le cône permet de bien conserver tout le contenu direction-



FIGURE 6.12 – Trois stratégies possibles pour la sélection de coefficients pour une direction dominante d'angle  $\pi/4$ .

nel mais peut parfois éliminer trop de contenu dans les basses fréquences. Le cylindre conserve bien mieux les basses fréquences mais élimine plus de coefficients haute fréquence dans la direction que l'on souhaite conserver. En combinant ces deux formes on peut conserver à la fois les coefficients de basses fréquences qui ne sont pas dans la direction dominante et les coefficients de hautes fréquences dans les directions proches de la principale.

La troisième étape n'est pas directement liée à la sélection de direction : elle consiste simplement à réaliser un seuillage sur les coefficients restants pour supprimer une partie du bruit (c'est en particulier utile pour le traitement de données réelles, par exemple de données sismiques comme nous le verrons à la section 7.1. Le choix du seuil peut être conditionné par une étude préalable du bruit (comme celle présentée à la section 6.1.8-b) dans le cadre d'un bruit blanc gaussien).

La sélection de direction a lieu dans le domaine de Fourier après chaque transformation locale et peut se résumer en les 3 étapes :

- ① Détermination de la direction principale en recherchant dans l'image après transformation le coefficient avec le plus grand module. On note  $\sigma_1$  cette valeur.
- ② On ne conserve que les coefficients proche de la direction principale.
- ③ Seuillage dur pour retirer les petits coefficients pouvant représenter du bruit. Le seuil peut être calculé comme étant une fraction de  $\sigma_1$ .

La figure 6.13 résume graphiquement les différentes étapes de l'algorithme ci-dessus.

### 6.2.3 Simulations sur une image artificielle

Une image anisotrope artificielle a été créée à partir du modèle de la section 6.2.1. À cette image, une direction parasite ainsi qu'un bruit blanc gaussien ( $\sigma = 40$ ) ont été ajoutés ; le PSNR de l'image bruitée était alors de 10,2 dB. La méthode de filtrage directionnel a alors été utilisée conduisant à une image après reconstruction dans laquelle la direction parasite et la plus grande partie du bruit ont été supprimées. Le PSNR après reconstruction était alors de 21,2 dB. La figure 6.14 présente les images originale, bruitée et obtenue après filtrage directionnel. On peut voir que le but a bien été atteint. En effet la direction parasite a bien été enlevée ainsi que le bruit.

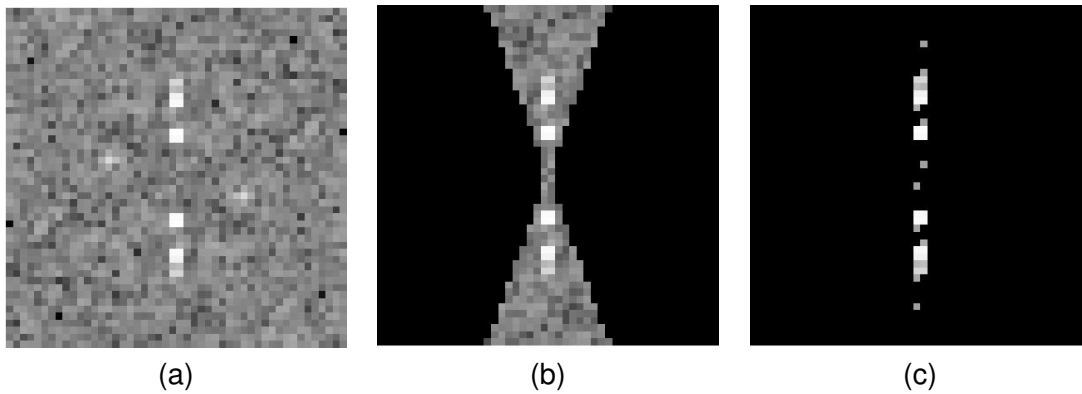


FIGURE 6.13 – Image des (logarithmes des) modules des coefficients (a) après transformation, (b) après sélection de la direction principale et (c) après le seuillage dur.

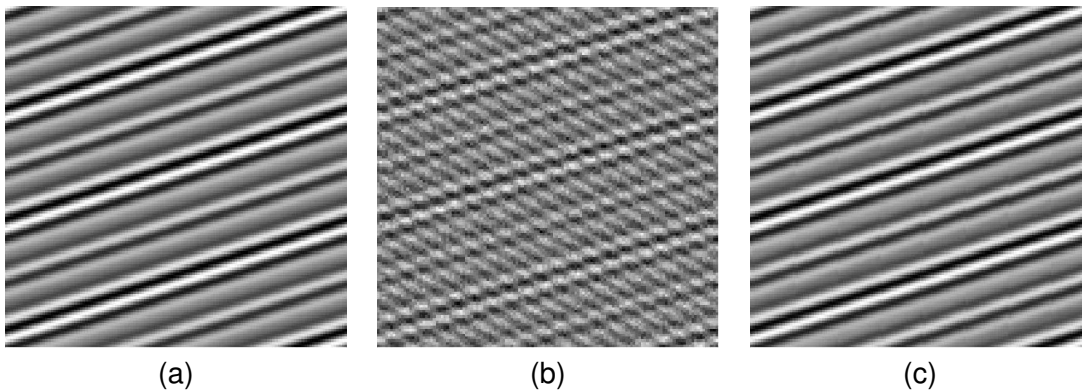


FIGURE 6.14 – (a) Image artificielle originale, (b) image bruitée et (c) résultat du filtrage directionnel.

### 6.2.4 Estimation des mélanges de direction

Une zone d'une image sera dite *désordonnée* si cette zone possède plusieurs orientations d'importance similaire. Avec les notations introduites précédemment, on peut définir un attribut de désordre par  $f(\frac{o_2}{o_1})$  où  $f$  est une fonction de  $[0, 1]$  dans  $[0, 1]$  décroissante. Dans la pratique, nous avons utilisé l'attribut de désordre suivant :

$$D_{\text{att}} = 1 - \frac{o_2}{o_1}.$$

Si deux directions ont des importances similaires il est alors clair que  $o_1$  et  $o_2$  auront des valeurs proches et donc  $D_{\text{att}}$  sera proche de 0 et, *a contrario*, si une direction l'emporte nettement sur les autres et sachant que par construction  $o_2 < o_1$  alors  $D_{\text{att}}$  aura une valeur proche de 1.

D'autres combinaisons des valeurs  $o_1$  et  $o_2$  peuvent être proposées conduisant à d'autres attributs pouvant caractériser les zones les plus chaotiques. Celle-ci présente l'avantage d'être rapide à calculer et aisée à interpréter.

**Remarque 6.5** Cette définition donnée dans le cas 2D s'étend naturellement en dimension quelconque.

### 6.2.5 Méthode de filtrage conservant les mélanges

À la section 6.2.2, nous avons proposé une méthode de sélection permettant de conserver la direction dominante (ou une direction particulière). Comme illustré à la figure 6.15(c), si l'on a deux directions d'importance similaire dans l'image, le filtrage précédent donne des résultats médiocres. Cela est dû au fait que, dans ce cas, l'estimation d'une seule direction dominante est peu fiable. Nous allons maintenant voir comment intégrer l'attribut de mélange de direction  $D_{att}$  pour conserver plusieurs directions dans les cas où l'importance relative de deux directions est proche.

La quantité  $D_{att}$  peut être considérée comme un niveau de confiance dans l'estimation de la direction dominante. Nous allons donc nous en servir pour adapter le filtrage. Pour ce faire, une étape est rajoutée à l'algorithme permettant l'évaluation de  $D_{att}$  et, si cet attribut a une valeur élevée – ce qui signifie qu'il y a deux directions d'importance proche – la stratégie est changée pour conserver les deux directions dominantes.

Le traitement a une fois encore lieu dans le domaine de Fourier après chaque transformation locale et se décompose en 4 étapes :

- ① Détermination de la direction principale en recherchant dans l'image après transformation le coefficient avec le plus grand module. On note  $o_1$  cette valeur.
- ② Recherche de la direction secondaire : on recherche encore une fois le coefficient de module maximum dans l'image après transformation mais, cette fois, en ignorant les coefficients proches de la direction principale. On obtient ainsi la valeur  $o_2$  et la direction secondaire.
- ③ Calcul de  $D_{att}$ . Si  $D_{att}$  est proche de 1 alors on ne conserve que la direction principale, sinon on conserve un mélange des deux directions dominantes.
- ④ Seuillage dur pour retirer les petits coefficients pouvant représenter du bruit. Le seuil peut être calculé comme étant une fraction de  $o_1$ .

**Remarque 6.6** Une nouvelle fois, si l'on est dans un cadre de bruit blanc gaussien additif on peut calculer le seuil pour la quatrième étape à partir de l'estimation de  $\sigma$  présentée à la section 6.1.8-b).

**Remarque 6.7** La troisième étape de cette méthode propose de considérer un mélange des directions principale et secondaire si  $D_{att}$  n'est pas suffisamment proche de 1. Un exemple de mélange peut être de faire une somme pondérée des directions en donnant un poids de 1 pour les coefficients correspondants à la direction principale et  $1 - D_{att}$  pour ceux dans la direction secondaire.

Voyons maintenant le résultat de ce nouveau filtrage sur des données artificielles construites de telle sorte à posséder deux directions d'importance équivalente. On peut voir à la figure 6.15 le résultat des filtrages avec et sans la prise en compte des mélanges de directions. Comme il fallait s'y attendre, en ne conservant que la direction principale on s'expose à un choix de direction dominante arbitraire et donc à des résultats de mauvaise

qualité. Des résultats sur des données réelles utilisant cette méthode seront montrés dans la section 7.1.2-b).

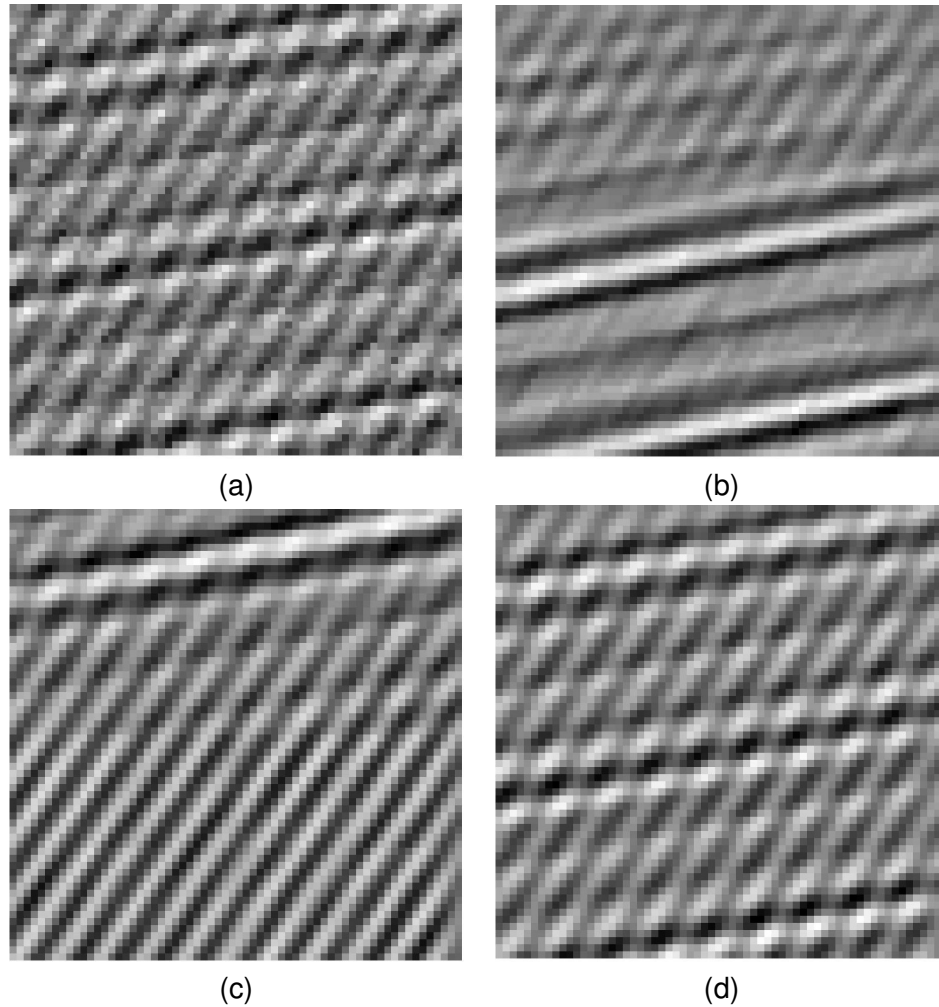


FIGURE 6.15 – (a) Image artificielle bruitée, (c) résultat du filtrage de la section 6.2.2, (d) résultat du filtrage avec l'estimation de mélange et (b) image de différence entre les deux résultats.

## 6.3 Conclusion

Dans ce chapitre, nous avons donc introduit deux méthodes statistiques de débruitage fondées sur l'estimateur de Stein. Ces méthodes s'appliquent à des décompositions complexes et permettent d'obtenir des résultats légèrement meilleurs que ceux que fournissent des méthodes récentes de débruitage. En particulier les bases de décompositions que nous avons utilisées sont bien adaptées pour le traitement d'images présentant des textures orientées.

Nous avons également proposé une méthode déterministe, plus empirique, de filtrage directionnel de données anisotropes, qui a le mérite d'être simple et versatile. La souplesse de cette méthode vient de la liberté dans le choix de la stratégie de filtrage, par exemple : garder uniquement une direction (principale ou non) ou un mélange de direction, et ce même en présence de bruit. Nous allons voir dans le chapitre suivant des exemples d'utilisation de ce filtrage directionnel sur différents types de données réelles.

---

## Chapitre 7

# Quelques exemples d'applications

---

Dans ce chapitre, nous allons présenter plusieurs applications des bancs de filtres construits et optimisés selon les méthodes du chapitre 4. En employant les techniques de sélection de coefficients vues au chapitre 6, des données 1D, 2D et 3D issues de différents problèmes industriels sont traitées. Dans un premier temps l'acquisition des données sismiques et la problématique de l'extraction d'informations sont rappelées. Des exemples d'applications du filtrage directionnel à des données 2D ou 3D sont présentés, accompagnés de quelques explications concernant une méthode d'extraction de structures géologiques particulières, appelées *onlap* et *toplap*. Une autre application de filtrage directionnel pour la séparation supervisée de directions est ensuite décrite mettant en jeu des données de Microscopie Électronique par Transmission (MET) de catalyseurs utilisés en raffinage. Enfin une dernière application est proposée pour l'analyse et la reconstruction temps-fréquence de signaux vibratoires 1D.

### 7.1 Données sismiques

#### 7.1.1 Présentation : acquisition des données sismiques

La sismique réflexion est une technique d'imagerie du sous-sol apparue dans les années 1920 et qui est aujourd'hui largement employée pour l'exploration et la recherche de nouveaux champs pétrolifères. À l'aide d'un système impulsionnel ou vibratoire : canons à air comprimé (en mer), camions vibreurs (sur terre) ou encore avec des explosifs, des ondes sont créées et se propagent dans le sol. Aux interfaces entre différents milieux d'impédances acoustiques distinctes, ou strates, ces ondes se réfléchissent et/ou se réfractent. Une partie peut alors remonter vers la surface où elles sont enregistrées par des capteurs (*hydrophones* en mer ou *géophones* sur terre) disposés en ligne (conduisant à la formation d'une image) ou en quadrillage (formant ainsi un cube de donnée). La figure 7.1 présente le dispositif d'enregistrement ainsi qu'un exemple de résultat "brut" issu d'une ligne de capteurs.

Le but de l'analyse de ces images est bien entendu de découvrir ce que recèle le

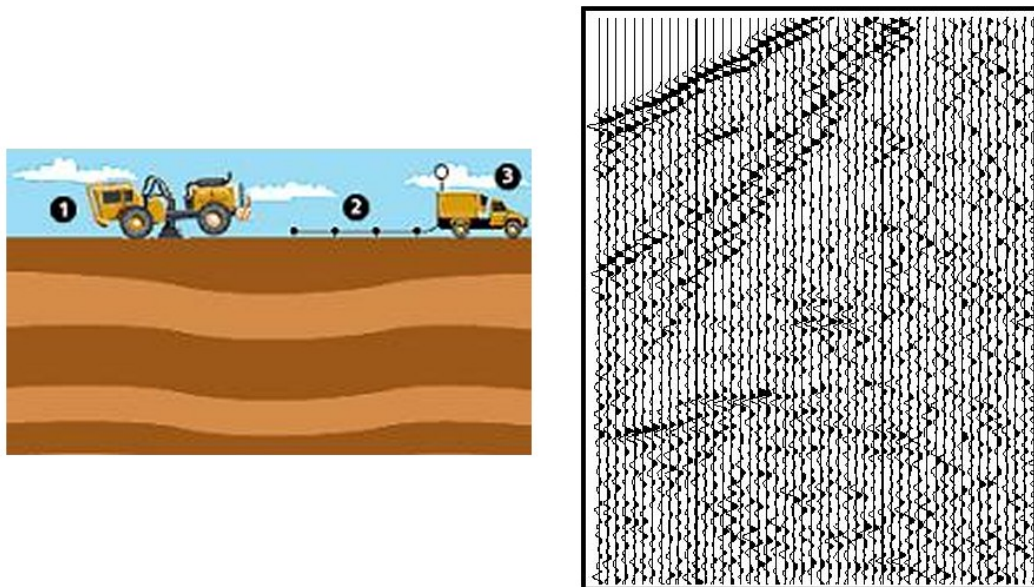


FIGURE 7.1 – Dispositif d'acquisition et exemple de mesure.

sous-sol en observant les changements de milieux qui conduisent aux réflexions visibles. Les images sismiques ainsi obtenues sont souvent très bruitées et difficiles à exploiter telles quelles.

Le bruit (au sens de l'ensemble des perturbations qui affectent le signal) peut provenir de plusieurs sources :

- bruits naturels. Le vent, la houle (pour les mesures en mer) ou un mouvement dans l'écorce terrestre, par exemple un petit séisme à quelques kilomètres, peuvent perturber les mesures.
- bruits humains. L'activité humaine entraîne une production de vibrations parasites à la surface qui sont enregistrées par les capteurs. Ainsi, une route qui passerait à proximité du lieu des mesures peut être cause de perturbations. Les traitements effectués sur les données brutes peuvent provoquer aussi des perturbations.
- bruits issus de la méthode d'acquisition. Par exemple, une onde peut être réfléchie plusieurs fois avant de revenir à la surface, créant ainsi ce qu'on appelle des “multiples” (répétitions du signal).

Un premier travail sur ces images consiste donc à améliorer les données brutes à l'aide d'une série de pré-traitements, incluant une déconvolution, du filtrage [Roueff *et al.*, 2004], *etc.* Dans le cadre de cette application nous nous sommes intéressés essentiellement aux images sismiques pré-traitées telles que celle présentée figure 7.2.

Pour plus de détails sur l'acquisition des images sismiques et leur pré-traitements on peut consulter [Lavergne, 1986; Yilmaz, Doherty, 2001; Mars *et al.*, 2004].

En étudiant localement ces données (par exemple, voir la figure 7.6), il est clair que ce type d'image est fortement anisotrope. Cette propriété est assez naturelle puisque ces images représentent différentes strates du sous sol. On peut voir également que, même après le pré-traitement, ces images sont toujours perturbées par des directions



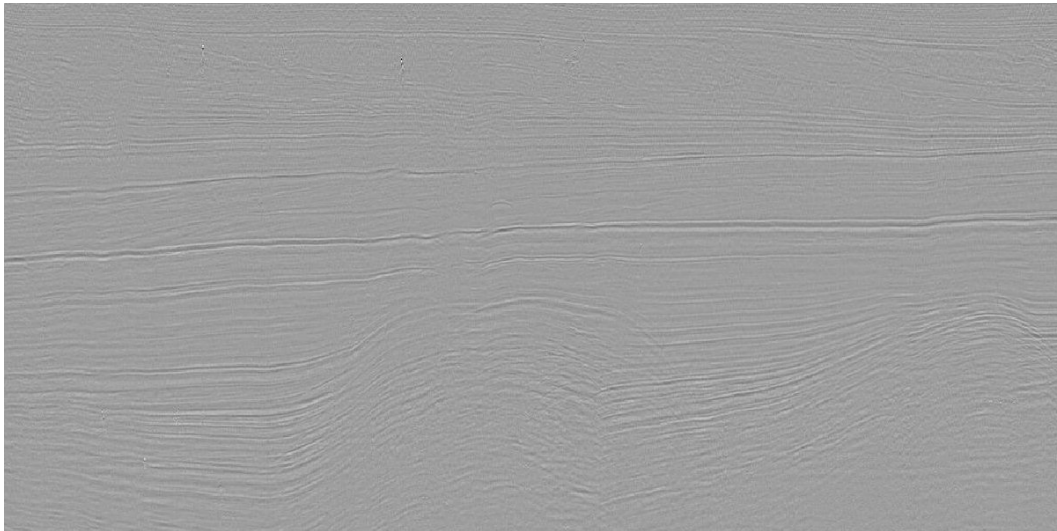


FIGURE 7.2 – Une image sismique après pré-traitement.

indésirables (effet de hachures liées notamment au traitement) et par du bruit aléatoire.

La taille des données est une autre caractéristique importante en sismique ; les cubes peuvent représenter plusieurs giga-octets de données à interpréter. À l'heure actuelle il n'est pas rare que les géologues passent plusieurs mois à analyser un bassin particulier. Automatiser certaines tâches élémentaires de l'interprétation est donc d'un grand intérêt pour les géologues et les compagnies pétrolières. Une tâche en particulier est essentielle, il s'agit de l'extraction automatique des horizons (les réflecteurs sismiques). Malheureusement les bruits évoqués précédemment tendent à perturber les résultats de ces extractions. C'est dans ce contexte difficile que nous avons proposé d'utiliser le filtrage directionnel décrit à la Section 6.2.2.

### 7.1.2 Filtrage directionnel

#### 7.1.2-a) Lien entre données synthétiques et sismiques

Comme nous l'avons vu précédemment, les données sismiques sont localement fortement anisotropes et constituées alternativement de bandes (ou de surfaces en 3D) sombres et lumineuses. Une image définie comme une combinaison linéaire de fonctions du type :  $f_k(x, y) = \sin(a_k x + b_k y)$  (ou  $f_k(x, y, z) = \sin(a_k x + b_k y + c_k z)$  en 3D) peut offrir une approximation locale grossière des données sismiques, comme on peut le voir à la figure 7.3. En particulier si les vecteurs  $(a_k, b_k)$  sont colinéaires, cela revient à ajouter de nouvelles fréquences dans la direction choisie et si des vecteurs ne sont pas colinéaires on ajoute alors des directions parasites aux données. Enfin l'ajout de bruit (gaussien par exemple) aux données permet de créer des données qui sont localement ressemblantes à des données sismiques. On peut voir à la figure 7.4 le résultat de cette modélisation sommaire.

Une modélisation de ce type est identique à celle introduite à la section 6.2.1. Le

filtrage directionnel utilisé sur des données artificielles ainsi générées ayant fait la preuve de son efficacité, voyons maintenant les résultats obtenus sur des données réelles.

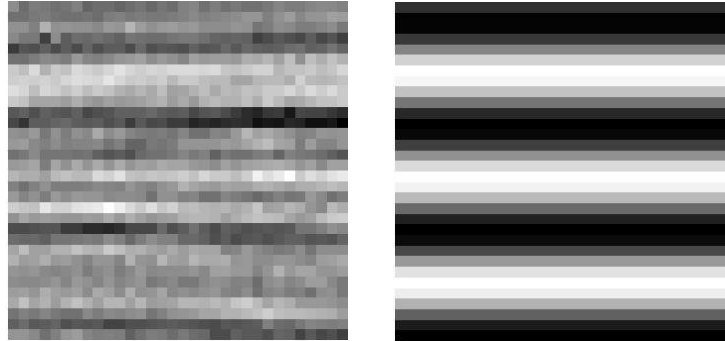


FIGURE 7.3 – Un échantillon d'image sismique et sa modélisation associée.

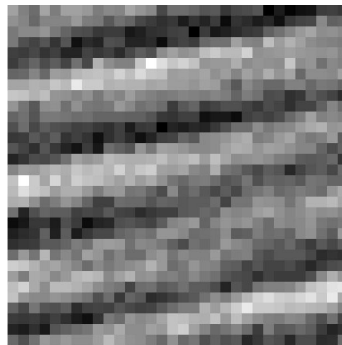


FIGURE 7.4 – Un exemple d'image du modèle perturbée par du bruit.

### 7.1.2-b) Images

**7.1.2-b)-1 Exemples de filtrages.** La figure 7.5 présente un exemple de filtrage sur l'ensemble d'une image sismique. Pour cette application nous avons utilisé le banc de filtres de la section 5.5.3 avec les paramètres  $k = k' = 3$  et  $N = 16$  et une fenêtre de type Kaiser 2D non séparable avec  $\alpha = \pi/4$  et  $\nu = 15$  (cf. équation (5.11)). Pour la sélection de direction nous avons utilisé la méthode décrite à la section 6.2.2, qui consiste à rechercher dans le domaine fréquentiel la direction locale dominante et à la conserver ainsi que les éléments ayant une orientation proche de la direction dominante.

Un premier résultat est que les structures globalement horizontales les plus visibles à l'œil, correspondant aux strates, sont bien conservées par le filtrage directionnel et l'image dans son ensemble semble plus régulière et plus "continue".

Examinons maintenant des détails de ces images. La figure 7.6 montre des zooms sur les données originales et les données filtrées. On y voit bien que localement le filtrage a conservé les structures dominantes et a rendu les données plus lisses en faisant

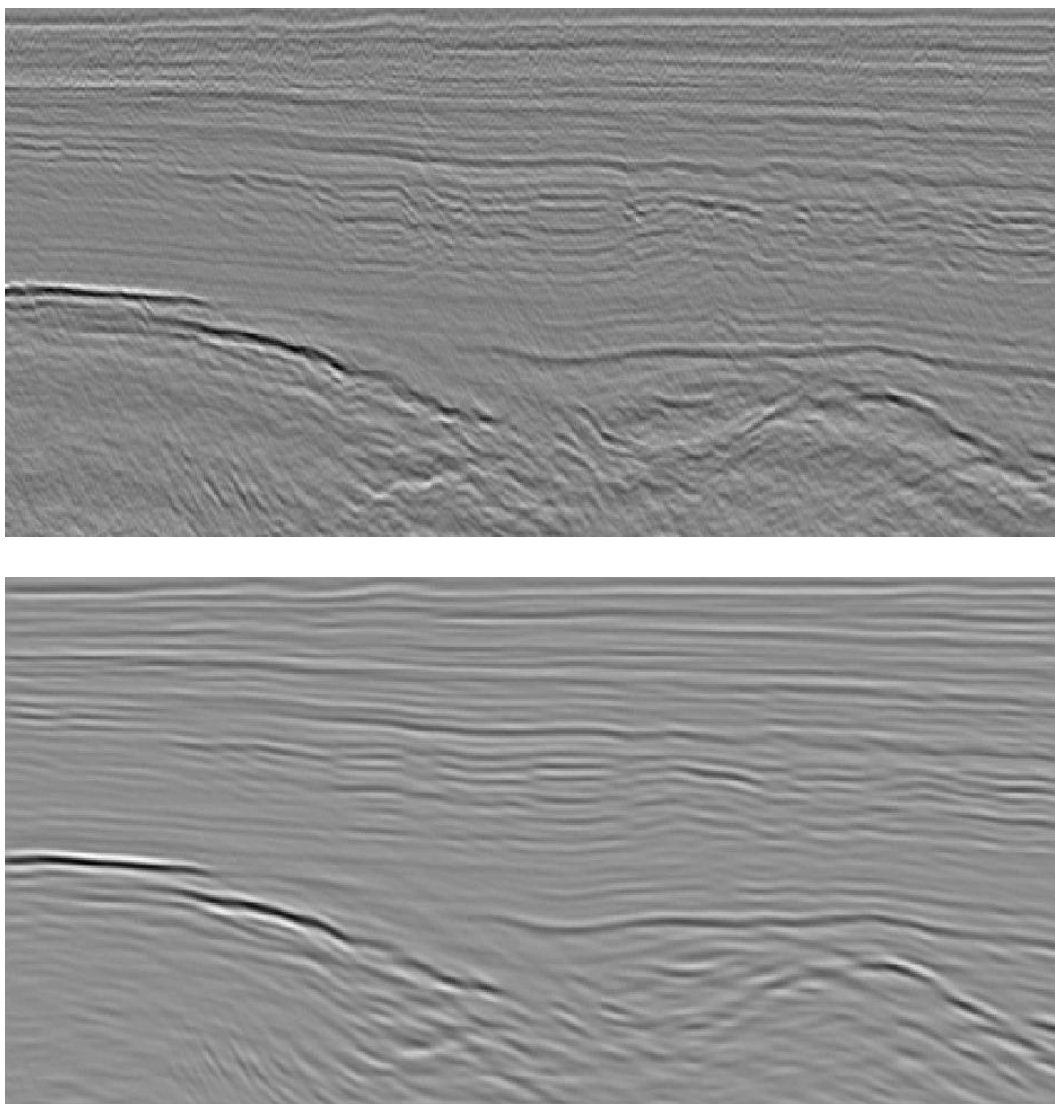


FIGURE 7.5 – Exemple d'une image sismique réelle et d'un résultat de filtrage sur cette image.

disparaître les directions parasites et l'aspect granuleux de l'image.

On peut également observer que ce filtrage a tendance à relier des réflecteurs qui étaient à l'origine légèrement séparés. Cet effet est à l'origine de l'impression de plus grande continuité des réflecteurs dans l'image. Cette caractéristique pose néanmoins un problème particulier pour le traitement des images très riches en failles. Celles-ci risquent en effet d'être estompées comme on peut le voir à la figure 7.7 présentant un filtrage directionnel sur une image fracturée.

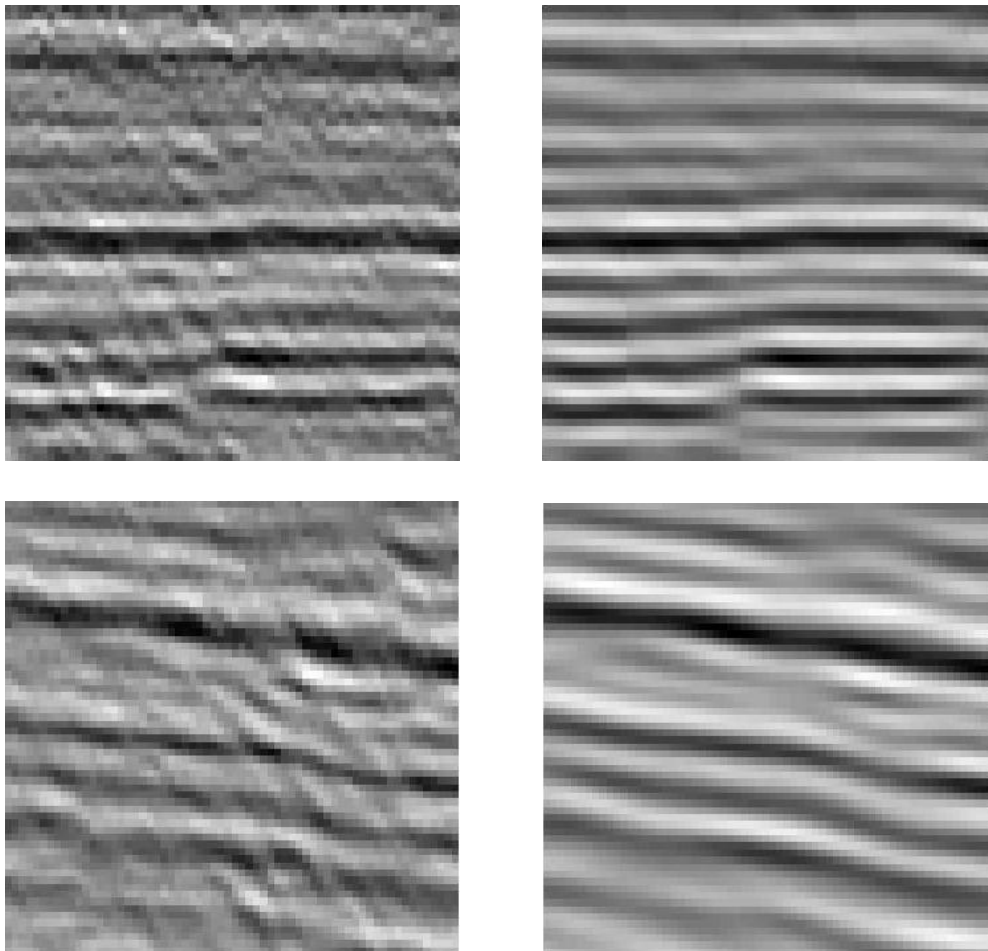
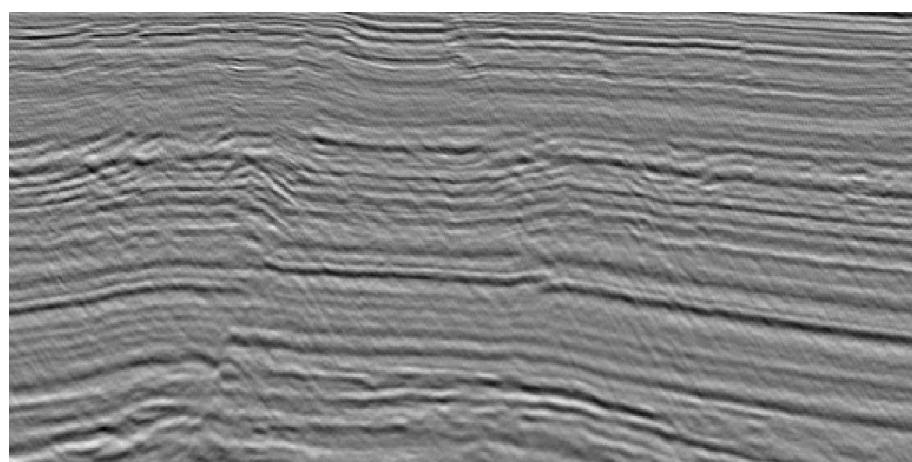


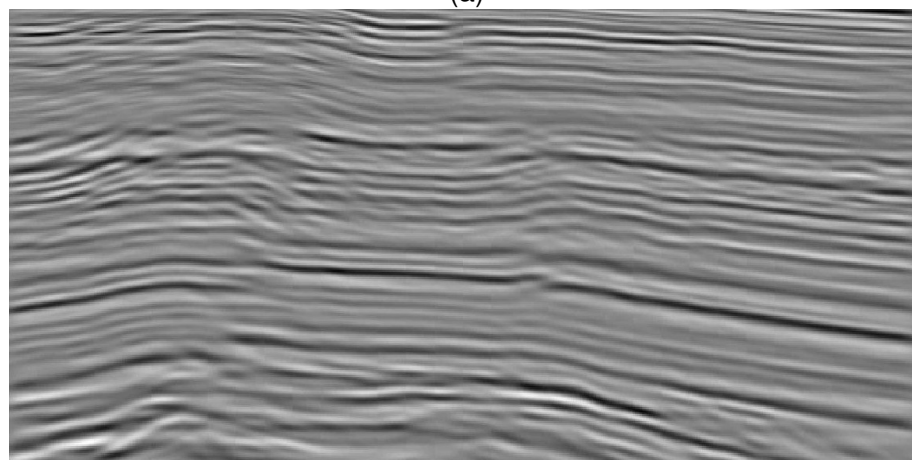
FIGURE 7.6 – Détails d'images sismiques réelles et de résultats du filtrage directionnel.

#### 7.1.2-b)-2 Utilité de l'attribut de désordre.

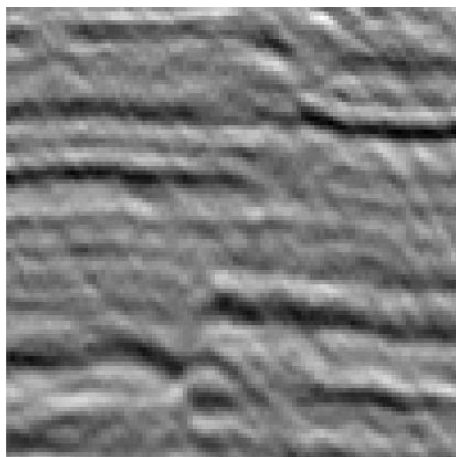
Nous avons introduit à la section 6.2.4, un attribut de désordre  $D_{att}$  permettant de repérer des zones présentant des mélanges de directions d'importance proche. Pour illustrer l'intérêt de  $D_{att}$ , nous avons testé le filtrage directionnel sur un bloc de données



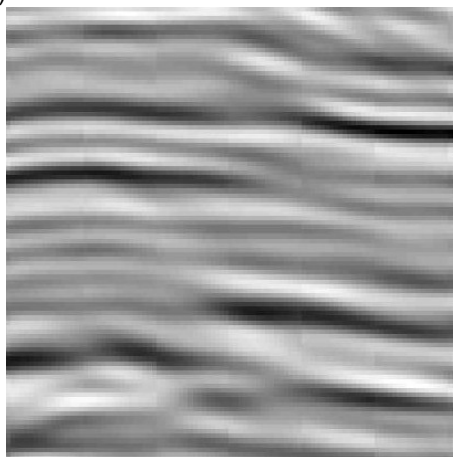
(a)



(b)



(c)



(d)

FIGURE 7.7 – (a) Image sismique fracturée, (b) résultat du filtrage, (c) détail de l'image originale et (d) détail après filtrage.

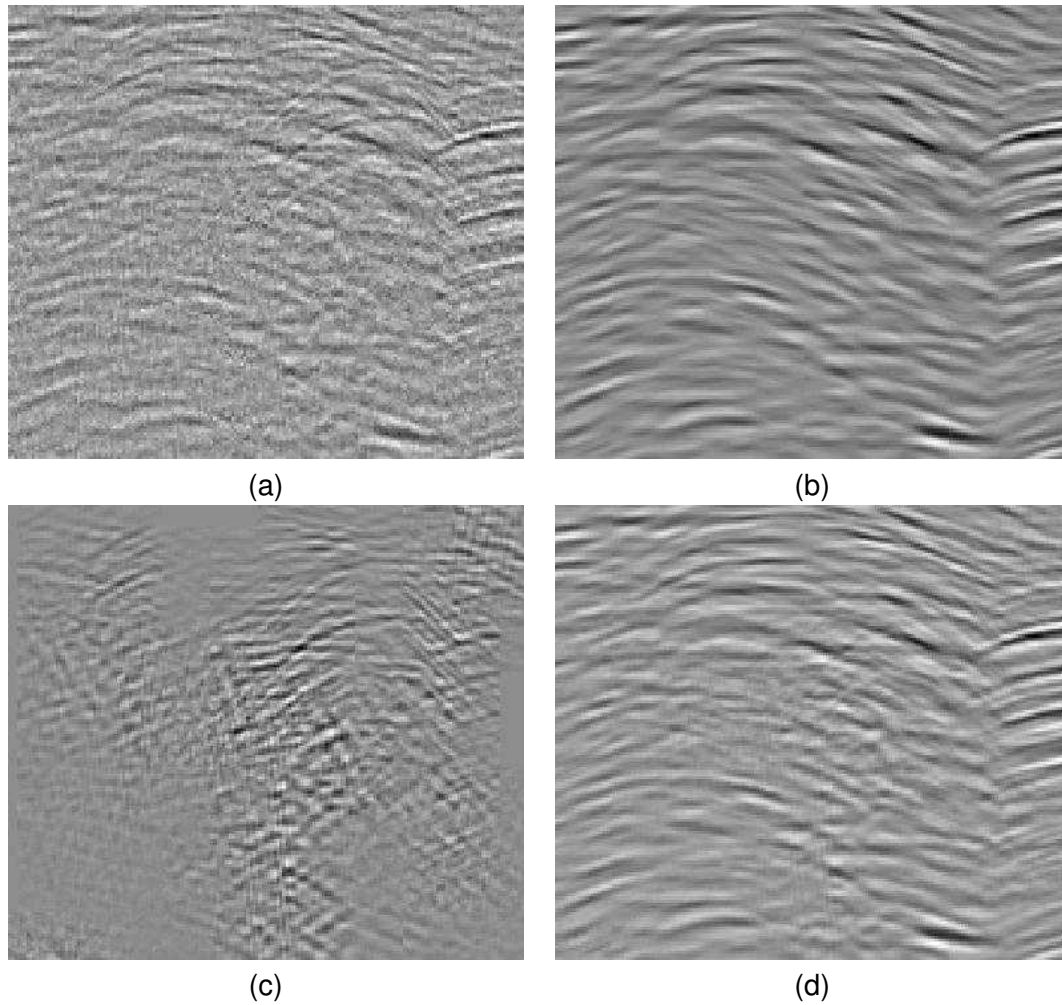


FIGURE 7.8 – Coupe issue : (a) d'un bloc de données sismiques, (b) des données filtrées sans utiliser l'attribut de désordre, (c) de la différence entre les deux filtrages et (d) en utilisant l'attribut.

sismiques présentant des zones complexes, aussi appelées chaotiques, où l'expertise d'un géologue semble nécessaire. En effet, pour ce type de zones où plusieurs directions d'importances similaires sont visibles, sélectionner arbitrairement la direction à conserver a de forte chance de conduire à un résultat erroné. Nous avons travaillé ici avec le banc de filtres d'analyse de la section 3.5.2 de paramètres  $k = 3$ ,  $k' = 7/4$  et  $N = 16$  et la fenêtre d'analyse  $h_{a1}$ . Le banc de filtres de synthèse utilisé était le résultat de l'optimisation à l'aide la fonction  $\widetilde{J}_{fs}$ , servant à optimiser les réponses fréquentielles du banc de filtres de synthèse.

La figure 7.8 présente un échantillon 2D de données sismiques, présentant à la fois des zones avec une direction dominante bien marquée ainsi que des zones plus chaotiques (dues à des mélanges de directions dans les données). Nous avons d'abord employé le filtrage directionnel de la section 6.2.2, qui ne conserve que la direction dominante puis celui de la section 6.2.5, qui préserve un mélange des deux directions

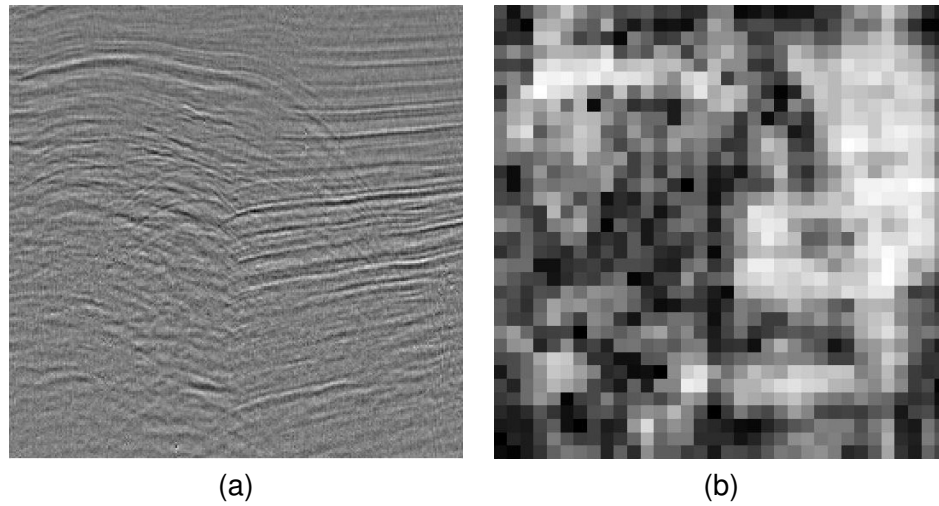


FIGURE 7.9 – (a) Image sismique originale et (b) image des valeurs de l'attribut de désordre  $D_{att}$ .

dominantes.

Les résultats après reconstruction sont très similaires dans les zones les plus régulières, comme on peut le voir sur l'image de différence (par exemple dans la partie en bas à gauche). En revanche, dans les zones chaotiques les données traitées en utilisant l'attribut  $D_{att}$  n'ont été que légèrement modifiées, préservant ainsi les structures dominantes dans plusieurs directions, ce qui n'est pas le cas avec la méthode de la section 6.2.2.

De plus, en mémorisant les zones correspondant à une valeur de  $D_{att}$  élevée, on peut indiquer directement à l'utilisateur des parties problématiques des données où son expertise sera nécessaire pour l'interprétation. La figure 7.9 représente les valeurs prises par  $D_{att}$  pour chaque bloc de taille  $kN \times kN$ . En prenant en compte le recouvrement de la transformation, chaque pixel de l'image d'attribut représente  $N \times N$  pixels de l'image d'origine. Les zones les plus lumineuses indiquent les parties de l'image où une direction est dominante et les zones plus sombres celles où il y a au moins deux directions importantes mixées. On voit en particulier que les zones avec une direction bien marquée (à droite de l'image) présentent une valeur élevée de  $D_{att}$ , alors que la zone du milieu de l'image où deux directions se mélangent conduit à des valeurs faibles de l'attribut.

**Remarque 7.1** *Un autre avantage de marquer les zones avec une valeur élevée de  $D_{att}$  est que cela peut permettre à l'utilisateur d'identifier certaines failles. En effet dans le cas où les directions dominantes de part et d'autre de la faille sont différentes, la valeur de  $D_{att}$  sera faible le long de la faille.*

### 7.1.2-c) Cubes

La technique de filtrage directionnel peut s'étendre naturellement, comme nous l'avons mentionné à la remarque 6.5, à des données tridimensionnelles. Notons que sur ce type de données, il est particulièrement intéressant de pouvoir travailler avec une redondance faible pour les bancs de filtres : en effet, si  $k'$  est la redondance du banc de filtre la re-

dondance de la transformation séparable tridimensionnelle est de  $k'^3$ . Pour cet exemple nous avons travaillé avec le banc de filtres introduit à la section 3.5.2 avec les paramètres :  $k = 3$ ,  $N = 16$  et  $k' = 3/2$ , ce qui fait donc une redondance totale pour la transformation de  $27/8 \approx 3.4$  (à titre de comparaison : avec une redondance  $k' = k = 3$ , comme dans le cas de la transformation de la section 5.5.3, la redondance en 3D est alors de 27). À nouveau, nous avons utilisé la fenêtre d'analyse  $h_{a1}$  et le banc de filtres de synthèse obtenu après optimisation avec la fonction de coût  $\widetilde{J}_{fs}$ .

La figure 7.10 présente une partie d'un cube de données sismiques et le résultat du filtrage directionnel. On peut constater que, comme dans le cas bidimensionnel, les données après filtrage sont beaucoup plus lisses et continues qu'auparavant. Les structures dominantes sont toujours bien conservées tandis que les perturbations directionnelles sont supprimées.

**Remarque 7.2** *Du point de vue de l'implémentation de la méthode de filtrage, on peut remarquer que l'analyse étant essentiellement locale (aux effets de recouvrement près), il est possible de séparer un cube à analyser en plusieurs sous-cubes de moindre taille et se recouvrant partiellement (de moins de  $((k-1)N)^3$  échantillons). Il est également possible de réaliser un découpage sans-recouvrement via des étapes de pré-filtrage et de post-filtrage [Tran et al., 2003]. On peut alors traiter chaque sous-cube indépendamment, ce qui a pour effet de réduire la quantité de mémoire nécessaire. Le fait d'avoir divisé le problème permet également de traiter plusieurs sous-cubes simultanément, à l'aide d'architectures multi-processeurs par exemple, réduisant ainsi le temps de calcul du traitement global.*

### 7.1.3 Extraction d'onlap/toplap

#### 7.1.3-a) Présentation du problème

À partir de l'observation de l'image sismique, l'interprète en stratigraphie sismique dispose de certaines règles, plus ou moins empiriques, pour trouver des indices permettant de reconstituer l'histoire sédimentaire de la zone étudiée, et donc de trouver des zones du sous-sol susceptibles de constituer des pièges pétroliers. Il se base notamment sur l'analyse de la "brillance" et de la longueur des réflecteurs, sur leur position et leur orientation relativement aux réflecteurs voisins (par exemple des réflecteurs parallèles peuvent indiquer un dépôt régulier à grande profondeur d'eau, d'un intérêt pétrolier limité), et sur la manière dont ceux-ci s'interrompent.

Des configurations particulièrement intéressantes sont constituées par des réflecteurs convergents organisés en éventail. Elles traduisent un basculement progressif du terrain au cours du dépôt et elles ont des implications sur la distribution des sédiments, ce qui peut conduire à la formation de pièges pétroliers. On peut distinguer deux configurations de convergence : les *onlaps* et les *toplaps* représentés schématiquement à la figure 7.11. Le but du traitement proposé est d'isoler et d'extraire automatiquement ce type de configurations.

La méthode présentée dans cette partie a été développée en collaboration avec Marie-Christine Cacas du département géologie de l'IFP et a donné lieu à un dépôt de brevet [Gauthier et al., 2006].



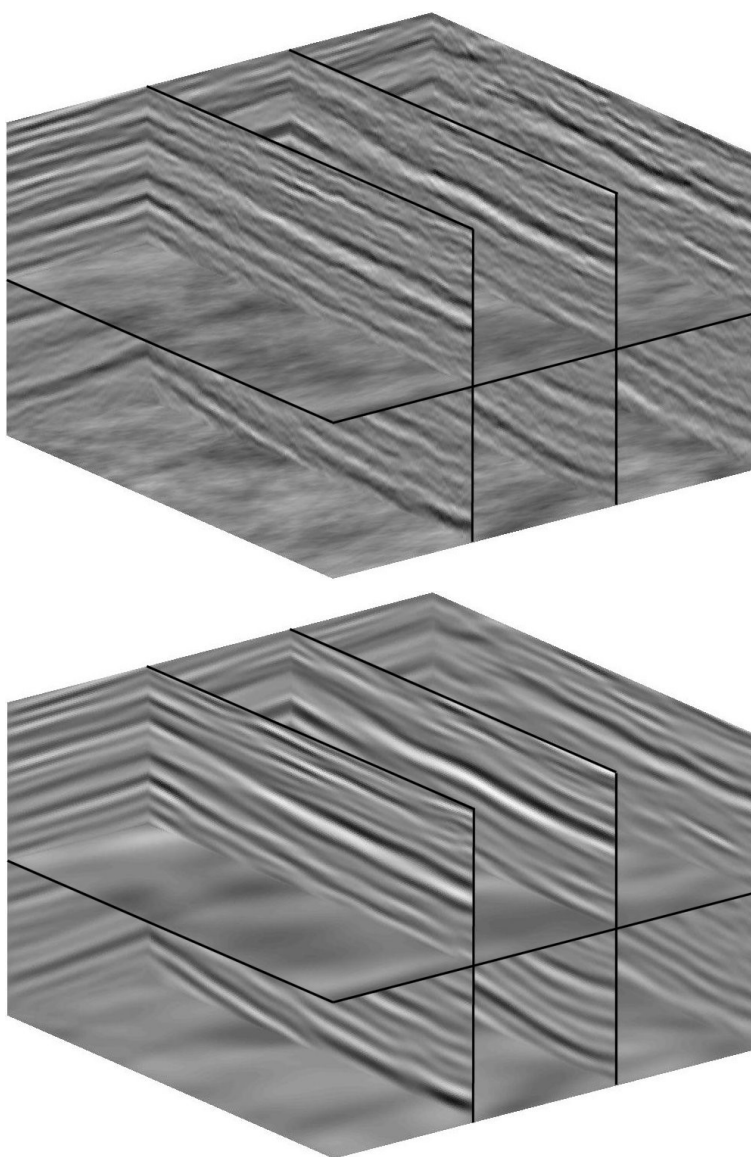


FIGURE 7.10 – Données sismiques 3D avant et après filtrage directionnel.

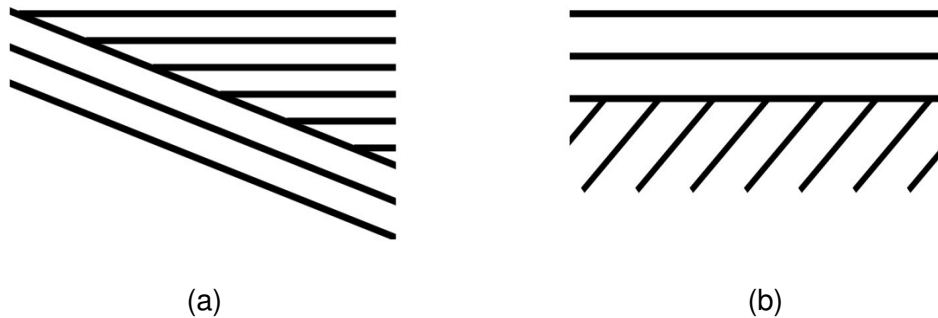


FIGURE 7.11 – Représentation schématique (a) d'un onlap et (b) d'un toplap.

### 7.1.3-b) Algorithme

Afin d'extraire automatiquement les onlaps et toplaps d'une image, la méthode suivante a été proposée.

La première étape consiste à utiliser sur les données à analyser le filtrage directionnel exposé à la section 6.2.2, ceci permet d'obtenir des images plus lisses et plus facile à traiter par la suite.

Ensuite, nous effectuons une étape de recherche et vectorisation des réflecteurs. Pour ce faire, nous avons utilisé la méthode décrite dans le brevet [Cacas, 2005]. Le principe est d'analyser l'image à l'aide d'une fenêtre glissante se présentant sous la forme d'une colonne de  $m$  pixels (dans la pratique on a pris  $m = 5$  pixels). Cette fenêtre verticale est centrée successivement sur chaque pixel de l'image. Si le pixel analysé, *i.e.* le pixel placé au centre de la fenêtre d'analyse, prend la valeur la plus élevée sur la fenêtre d'analyse, il est sélectionné comme faisant partie d'un réflecteur. Enfin, une étude de 8-connexité est menée sur les pixels sélectionnés (les maxima locaux de l'image) pour créer une squelettisation des réflecteurs. Cette méthode a l'avantage de garantir que la squelettisation ne présentera que des réflecteurs connexes d'épaisseur 1 pixel si la recherche de maximum local se fait sur des voisinages de taille au moins  $m = 5$  pixels.

La troisième étape consiste à *redresser* chaque réflecteur repéré après la squelettisation. Le principe est simplement d'aligner les éléments d'un réflecteur, en décalant verticalement chaque colonne de l'image en fonction de l'écart de position entre le point du réflecteur contenu dans cette colonne et le premier pixel du réflecteur. Le fait qu'après la squelettisation un réflecteur donné possède au plus un point dans chaque colonne nous assure que la méthode pour redresser les réflecteurs présentée ici fonctionne bien dans tous les cas.

Puis, on calcule des attributs liés aux réflecteurs redressés ; dans cette application nous avons choisi comme attributs les angles dominants apparents au dessus et en dessous de chaque réflecteur. En effet, les configurations recherchées sont caractérisées par des orientations différentes des réflecteurs situés de part et d'autre d'un réflecteur. Ces calculs d'angles apparents sont réalisés grâce à une méthode identique à celle exposée à la section 6.2.2 pour la détection de l'orientation dominante.

Enfin, en comparant les valeurs d'attributs calculées, on détermine les zones d'onlap/toplap recherchées.

Cette méthode peut être résumée par la procédure suivante :

1. Filtrage directionnel des données sismiques.
2. Recherche et vectorisation des réflecteurs. Ne sont gardés à l'issue de cette étape que les réflecteurs les plus longs. Soit  $\mathcal{S}$  l'ensemble de ces réflecteurs.
3. Un réflecteur de  $\mathcal{S}$  est redressé. Le long de ce réflecteur on calcule des attributs de texture locaux : l'angle dominant au dessus et en dessous du réflecteur redressé.
4. Si tous les vecteurs de  $\mathcal{S}$  n'ont pas été étudiés, retour en 3.
5. À partir des attributs de chaque réflecteur, on détermine les réflecteurs se trouvant probablement le long d'une zone d'onlap/toplap. Pour cela, on calcule la différence entre l'angle dominant au dessus et en dessous de chaque réflecteur et on ne conserve que ceux présentant une grande différence.

**Remarque 7.3** *La méthode de squelettisation, ainsi que les autres opérations utilisées ici, étant réalisables également en 3D, cette méthode est donc également applicable sur des cubes.*

### 7.1.3-c) Exemple de recherche

Voyons maintenant un exemple sur des données réelles. La figure 7.12 représente un exemple d'image sismique avec un toplap ainsi que le résultat du filtrage directionnel et de la squelettisation. Le filtrage directionnel a été effectué par la méthode de la section 6.2.2, avec le banc de filtres d'analyse de la section 5.5.3 avec les paramètres  $k = k' = 3$  et  $N = 16$  et une fenêtre de type Kaiser 2D non séparable avec  $\alpha = \pi/4$  et  $\nu = 15$ . L'ensemble  $\mathcal{S}$  est alors constitué par les plus longs réflecteurs (de longueur supérieure à 48 pixels) visibles en blanc sur cette image.

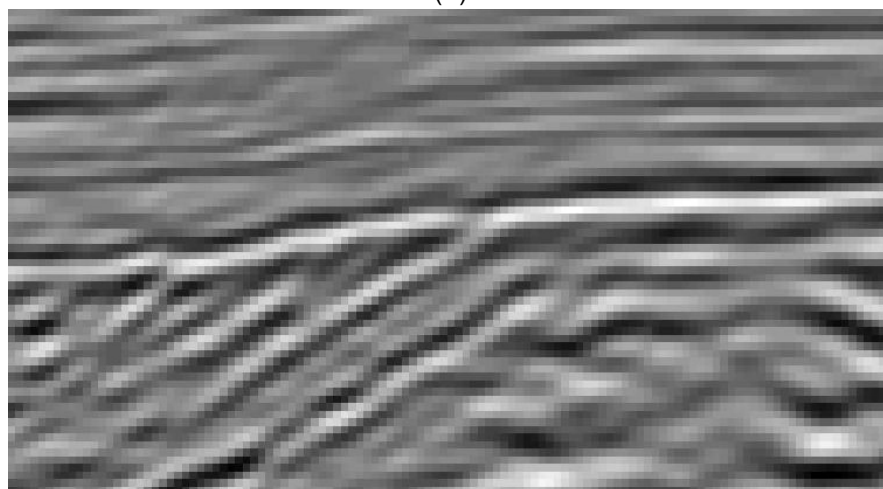
La troisième étape de l'algorithme est illustrée par la figure 7.13, où l'on peut voir un réflecteur sélectionné redressé par la méthode du brevet [Cacas, 2005] avec une taille de voisinage de 5 pixels, ainsi que le calcul de l'attribut d'angle dominant local au dessus de ce réflecteur.

Enfin, on peut voir (figure 7.14) le résultat de la sélection des parties de réflecteurs potentiellement le long d'une configuration de type onlap/toplap. Cette sélection s'est faite simplement en comparant (par différence et seuillage) les angles dominants détectés au dessus et en dessous de chaque réflecteur considéré. Les parties de réflecteurs représentées en blanc marquent les zones où il est probable de rencontrer un onlap ou un toplap.

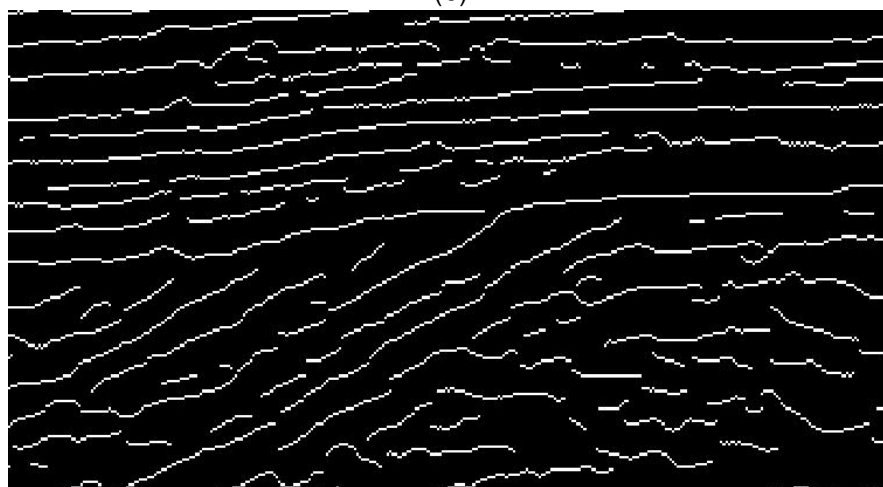
Il ne reste plus alors au géophysicien qu'à valider, ou non, les zones détectées par cette méthode. Cette procédure ne permet pas forcément de repérer exactement toutes les zones d'onlap/toplap. En effet, la phase d'interprétation de l'algorithme, qui se fait par seuillage peut entraîner le marquage de "faux positifs". L'avantage est donc que cette méthode fournit une indication à l'utilisateur des zones à étudier et permet donc de gagner



(a)

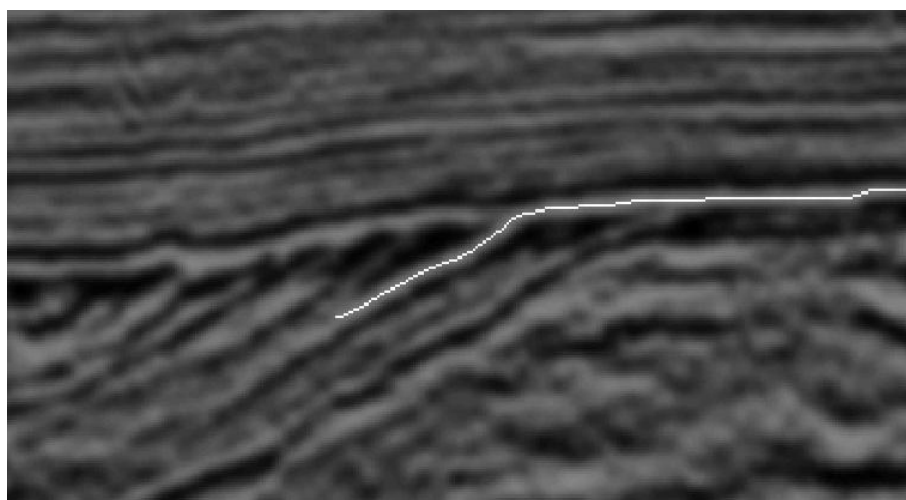


(b)

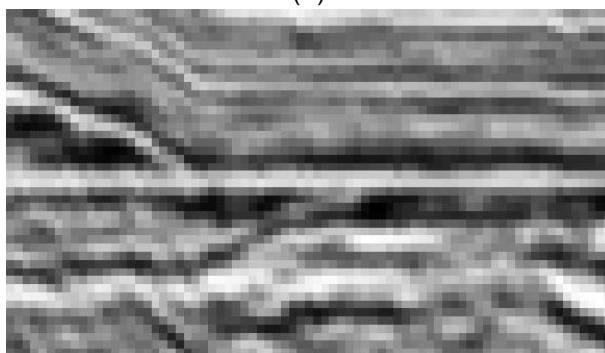


(c)

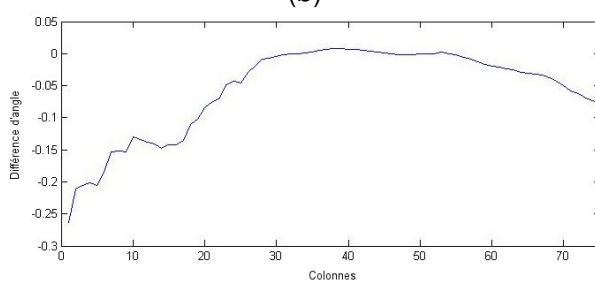
FIGURE 7.12 – (a) Image originale, (b) image après filtrage directionnel et (c) squelettisation.



(a)



(b)



(c)

FIGURE 7.13 – (a) Sélection d'un réflecteur, (b) réflecteur redressé (c) calcul de l'angle dominant au dessus de ce réflecteur.



FIGURE 7.14 – Résultat de la recherche d'onlap/toplap.

du temps dans une analyse de données sismiques.

#### 7.1.4 Autres applications en sismique

##### 7.1.4-a) Classification chronologique

Comme nous avons pu le voir pour la recherche d'onlap/toplap, la finalité du filtrage directionnel n'est pas de se substituer à l'image originale mais de constituer un maillon dans une chaîne de processus permettant une analyse automatique de certaines caractéristiques de l'image. Voyons maintenant une autre application en sismique où ce filtrage est utilisé.

Une information potentiellement importante pour l'analyse des données sismiques, est l'âge relatif des différentes régions apparaissant dans une image. Une méthode pour faire cette classification consiste à extraire les réflecteurs par vectorisation, de manière similaire à ce qui a été fait à la section précédente, puis à les classer en fonction de leurs positions verticales relatives, l'hypothèse étant qu'un réflecteur *A* est antérieur à un réflecteur *B* si le réflecteur *A* recouvre, au moins en partie, un réflecteur *B*. Autrement dit plus un réflecteur est situé en "bas" de l'image, plus il est ancien. On peut partir des réflecteurs les plus récents (qui ne sont recouverts, même en partie, par aucun autre) auxquels on donne l'indice 0, puis on donne l'indice 1 à tous ceux qui ne sont recouverts, partiellement ou complètement, que par des réflecteurs d'indice 0 et ainsi de suite jusqu'à avoir assigné un indice à chaque réflecteur.

On peut également faire la même opération en partant des plus anciens réflecteurs. On obtiendra ainsi une classification légèrement différente (du fait que l'ordre entre réflecteurs n'est pas total). La figure 7.15 illustre le principe de cette classification : les indices en allant du plus ancien au plus récent sont écrits en rouge (à droite) et ceux du plus récent au plus ancien sont en bleu (à gauche). Sur cet exemple, on peut voir que suivant que l'on classe les réflecteurs du plus ancien au plus récent ou du plus récent au plus l'ancien, des réflecteurs considérés comme contemporains (de même indice) dans

un cas ne le sont plus dans l'autre : le réflecteur "1/1" est contemporain de "2/1" si l'on considère l'ordre du plus ancien réflecteur au plus récent et de "1/2" si l'on classe du plus récent au plus ancien.

Il est clair que le résultat de la classification chronologique sera influencé par la longueur des réflecteurs après vectorisation : plus les réflecteurs seront longs, moins il y aura de classes. L'opération de filtrage directionnel faisant ressortir les réflecteurs, ceux-ci, après squelettisation, auront tendance à être plus longs. Nous avons comparé les résultats de classement chronologique avant et après filtrage directionnel. La figure 7.16 représente les résultats de la classification sur l'image de la figure 7.2 avec et sans filtrage. Ceci illustre que l'utilisation d'un filtrage permet d'obtenir un résultat plus fin et plus facilement exploitable. Le fait que les réflecteurs apparaissent plus longs après filtrage implique que les zones d'incertitudes (en noir sur les images) sont plus réduites.

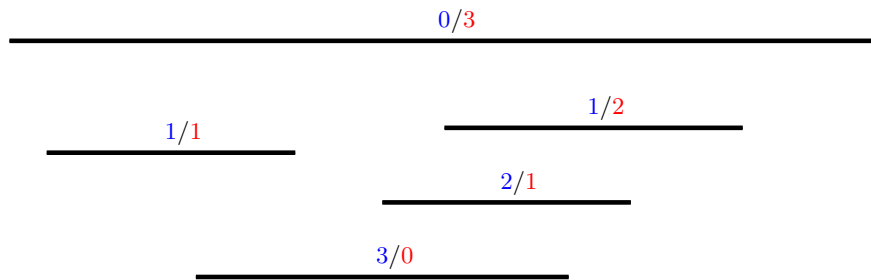


FIGURE 7.15 – Schéma illustrant le principe de la classification chronologique.

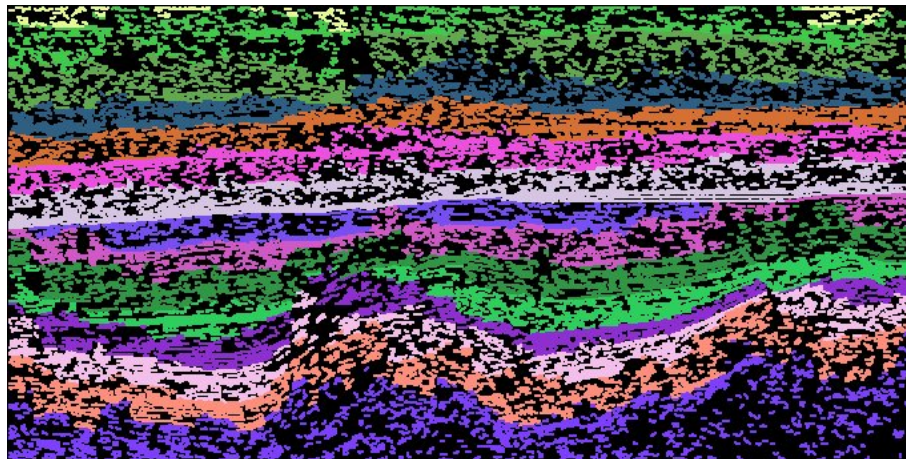
#### 7.1.4-b) Traitement d'image numérisées

Voyons maintenant une dernière application, peut-être un peu anecdotique, du filtrage directionnel appliqué aux données sismiques.

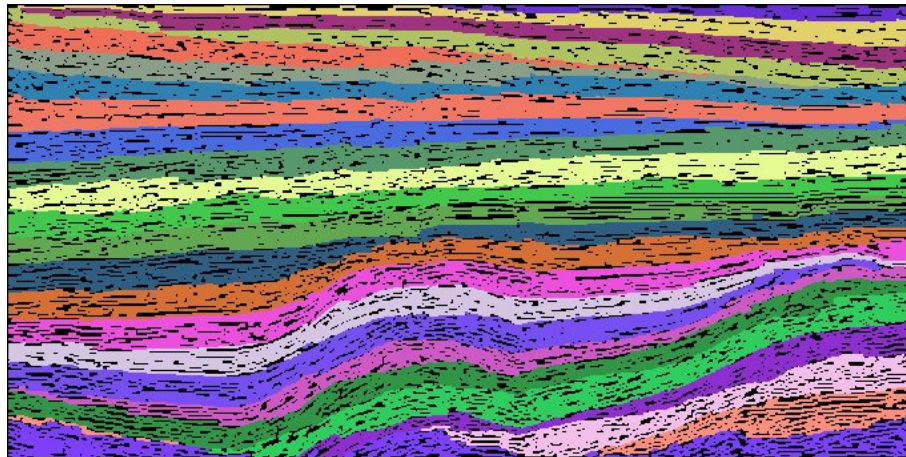
L'enregistrement des données sismiques est, comme nous l'avons vu, une activité apparue dans la première moitié du vingtième siècle. Depuis cette époque de très nombreuses campagnes d'acquisition ont été menées, ce qui implique de grandes quantités de données à conserver. Il arrive que les supports de stockage (par exemple des bandes magnétiques) se dégradent ou bien soient perdus rendant les données inexploitable. Dans ce cas il ne reste parfois plus que les données imprimées. On peut alors tenter de numériser ces images pour pouvoir travailler de nouveau sur les zones qui avaient été sondées.

Un exemple d'image numérisée est présenté à la figure 7.17, où l'on peut voir ressortir des marques blanches verticales correspondant aux endroit où le papier était plié, ainsi que la trame du papier. Afin d'ôter ces marques, nous avons modifié légèrement la méthode de filtrage directionnel présentée à la section 6.2.5 pour pouvoir éliminer la direction verticale (même si elle est localement dominante). Le résultat du filtrage est visible à la figure 7.17, où il est clair que toutes les pliures verticales ont disparues.





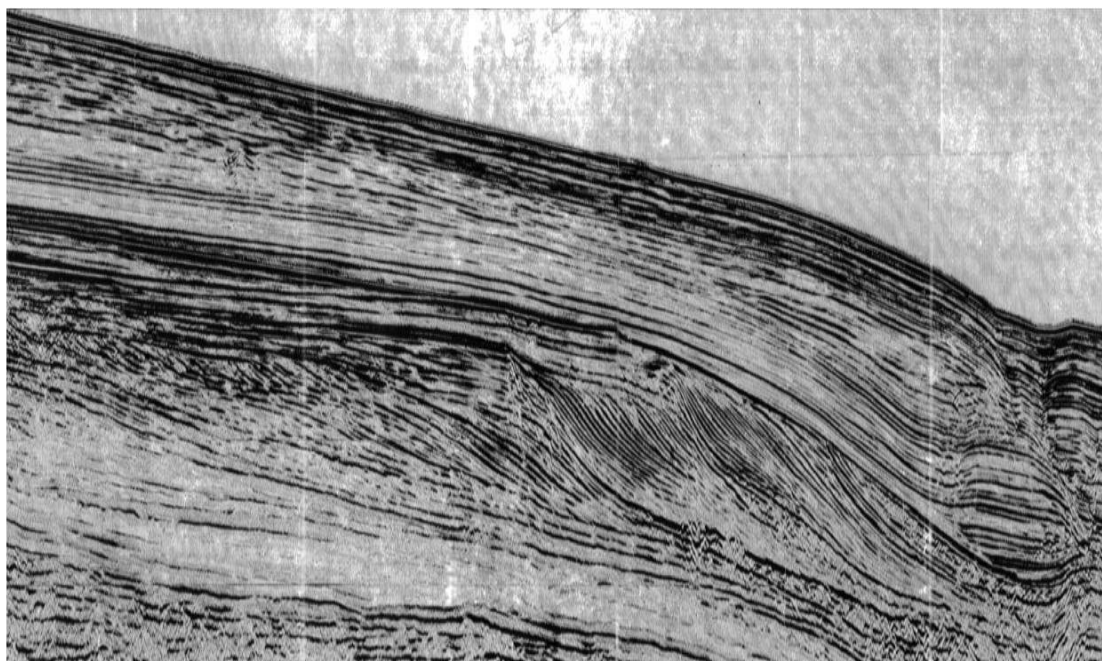
(a)



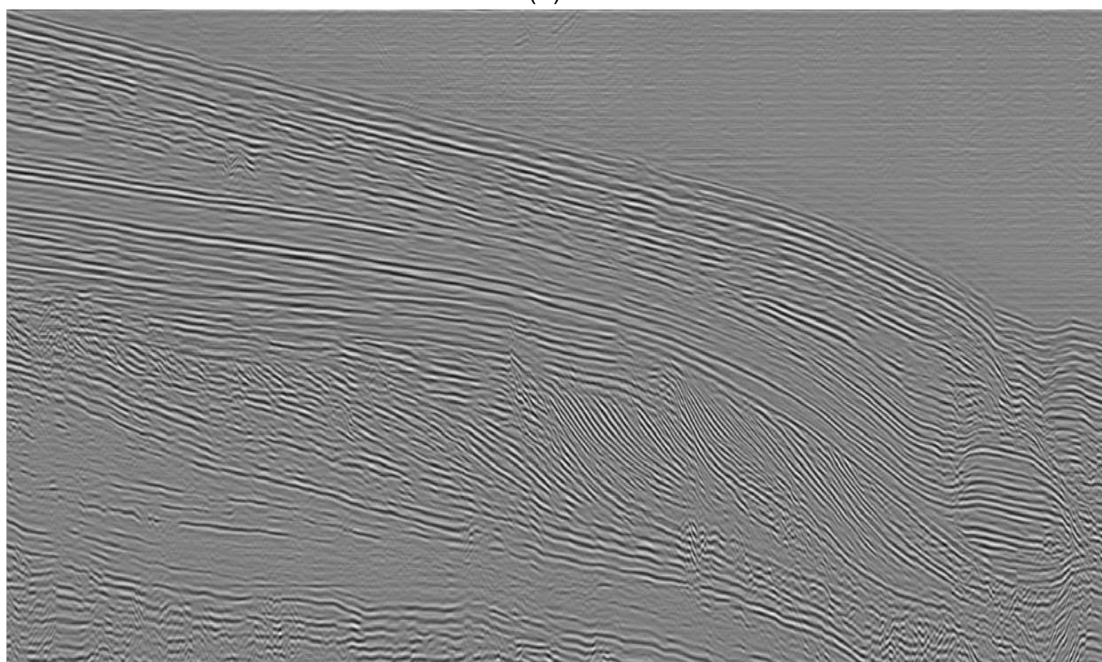
(b)

FIGURE 7.16 – Classification chronologique (a) sans filtrage directionnel préliminaire et (b) après filtrage directionnel.





(a)



(b)

FIGURE 7.17 – (a) Image sismique (sur support papier) numérisée et (b) image filtrée.

## 7.2 Microscopie électronique à transmission

### 7.2.1 Présentation et problématique

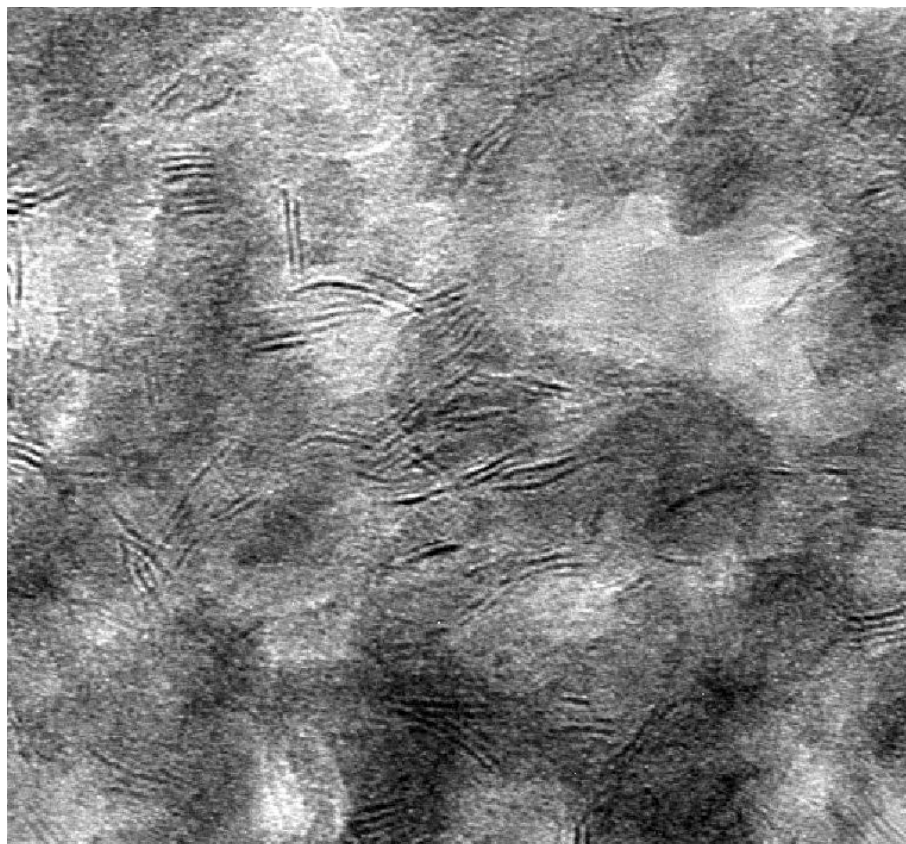


FIGURE 7.18 – Une image de microscopie (MET).

La présence de soufre dans les carburants provoque, lors de leur combustion, la création de molécules de dioxyde de soufre ( $\text{SO}_2$ ) qui sont, en partie, à l'origine des pluies acides. Dans le cadre de la lutte contre la pollution, les normes définissant la teneur en soufre des carburants vendus dans le commerce sont donc de plus en plus sévères, ce qui constitue un défi pour les raffineurs. Une solution à ce problème est l'hydrotraitement qui est un processus chimique visant à éliminer les particules soufrées du pétrole.

Pour analyser et mieux comprendre ce processus catalytique et ainsi pouvoir l'améliorer, il est important d'étudier certaines phases actives impliquées dans le traitement. Plusieurs méthodes sont envisageables pour réaliser ce travail : spectroscopie de photo-électrons X, mesures d'activité ou microscopie électronique à transmission (MET) [Gandubert *et al.*, 2007]. Dans ce travail, nous nous sommes intéressés à cette dernière méthode. Un exemple d'image obtenue par MET est présenté à la figure 7.18. On peut voir que localement il y a des éléments fortement anisotropes appelés *feuillets* ; ces feuillets sont constitués de  $\text{MoS}_2$  (disulfure de molybdène) et sont le siège des réactions chi-

miques. Il est intéressant pour mieux comprendre les réactions de pouvoir compter ces feuillets ainsi que de calculer leur longueur.

Malheureusement, comme on peut le voir à la figure 7.19(a) ces images présentent du bruit et dans certains cas les feuillets se superposent ce qui rend la tâche d'extraction automatique difficile. Il n'est pas rare que les opérateurs doivent isoler et mesurer eux-mêmes les feuillets ce qui peut impliquer des heures de travail.

### 7.2.2 Résultats

Dans cette application, nous allons considérer que l'opérateur isole une zone de l'image contenant des feuillets et qu'il indique une direction privilégiée à extraire. Pour ce faire, nous avons utilisé le filtrage directionnel de la section 6.2.2 avec une direction principale à conserver fixée. Une contrainte pratique étant de minimiser les temps de calcul, nous avons choisi d'utiliser un banc de filtres d'analyse comme celui de la section 3.5.2 avec une *petite* redondance  $k' = 3/2$ , et les paramètres :  $N = 8$  et  $k = 3$ .

Dans cet exemple, une sous-image (de taille  $150 \times 45$ ) de celle présentée sur la figure 7.19 a été choisie présentant deux feuillets se superposant en partie. Ce cas de figure est le plus épineux puisqu'il nécessite de séparer les deux directions. Il s'agit donc d'une application de séparation de sources supervisée à l'aide d'une décomposition temps-fréquence, ce qui peut être rapproché de [Nguyen *et al.*, 2005]. On peut voir à la figure 7.19 les résultats obtenus pour les deux directions données par l'opérateur. Dans les deux cas, il est clair que les éléments directionnels ont été bien séparés : en effet il ne reste dans les images filtrées que les éléments de l'image dans la direction choisie. On peut néanmoins se demander si l'on n'a pas extrait plus d'informations directionnelles que ce que contenait l'image d'origine (ce qui ferait apparaître en des feuillets plus long que ce qu'il est possible de voir dans l'image d'origine). En considérant l'image de différence (d) on peut constater qu'il ne reste plus d'éléments orientés dans l'une ou l'autre des directions choisies par l'utilisateur. Ceci montre que l'on n'a pas extrait d'informations directionnelles qui n'existent pas dans l'image originale.

## 7.3 Analyse temps-fréquence de signaux de combustion

### 7.3.1 Motivations

Un autre type d'application sur laquelle nous avons travaillé est l'étude des signaux vibratoires issus des moteurs d'automobile.

Afin d'obtenir des moteurs plus performants, les constructeurs ont introduit une partie de contrôle sur la combustion. Plusieurs objectifs sont ainsi visés :

- réduction de la consommation. Une combustion bien contrôlée permet d'optimiser la quantité de carburant injectée donc de moins consommer et ainsi de limiter la production de  $\text{CO}_2$ .
- limitation de la production de polluants. Lors de la combustion plusieurs composés peuvent se former, par exemple des oxydes d'azotes ( $\text{NO}_x$ ), du monoxyde de carbone (CO) ou des hydrocarbures. Ces molécules présentent des dangers pour la santé : irritations respiratoires ou oculaires, cancers et leucémies. Contrôler la

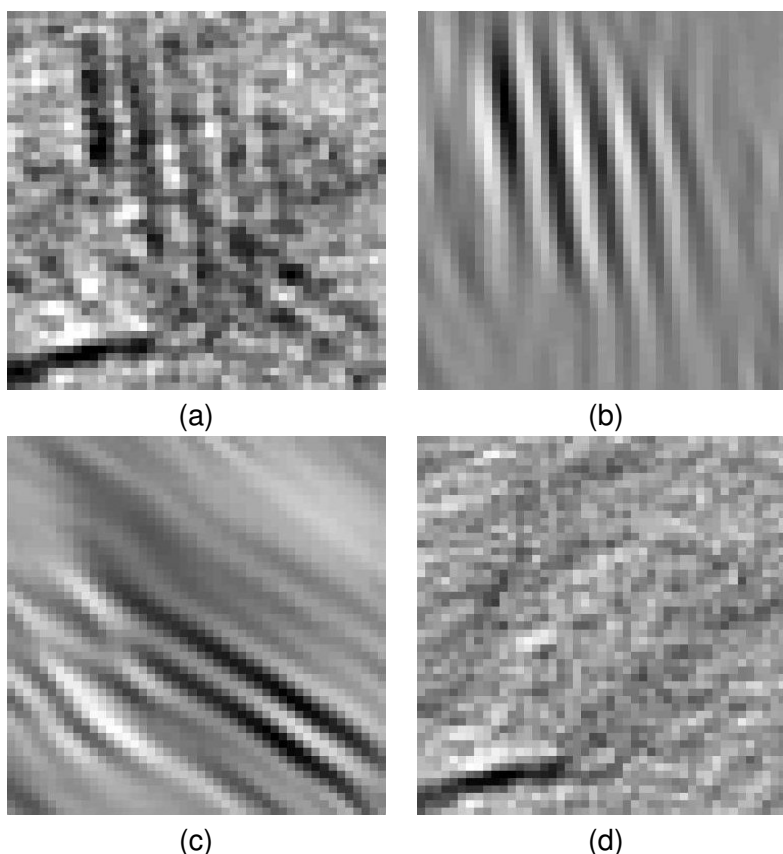


FIGURE 7.19 – (a) Échantillon d'image MET, (b)(c) résultats après filtrage directionnel (d) image de différence.

combustion permet de limiter la formation de ces produits ou au moins de mieux la prévoir et ainsi de pouvoir utiliser d'autres stratégies de traitement après la combustion.

- conserver l'agrément de conduite. Si le conducteur demande un certain couple, il faut que le contrôle adapte la combustion pour le lui fournir.

Une information évidemment importante pour le contrôle de la combustion est la *pression cylindre*. Malheureusement les capteurs de pressions pouvant être utilisés dans les chambres de combustions sont onéreux et ont une durée de vie courte (comparée à la durée de vie d'une voiture), il faut donc trouver une alternative applicable aux véhicules de série pour extraire des informations sur la pression cylindre.

Le gradient de pression dans la chambre de combustion produit des vibrations qui se propagent au travers de la structure du moteur. Ces vibrations, enregistrées par des capteurs de cliquetis (bon marché et existant déjà sur des voitures de série), sont malheureusement corrompus par plusieurs sources de bruits issus du reste du moteur (par exemple les injections). Il est donc intéressant d'essayer d'isoler la part issue de la combustion. Pour ce faire, une analyse temps-fréquence peut s'avérer utile.

### 7.3.2 Représentation temps-fréquence

Un outil classique de représentation temps-fréquence est la transformée de Fourier à court terme (STFT - Short Time Fourier Transform). Rappelons brièvement ce qu'est la STFT.

Soient  $x(t)$  un signal continu et  $h(t)$  une fenêtre d'analyse, tous deux d'énergie finie. La formulation classique de la STFT continue fenêtrée  $F_x(t, \nu; h)$  revient à fenêtrer le signal  $x$  autour d'un moment  $t$  donné puis à calculer la transformée de Fourier du résultat [Flandrin, 1998] :

$$F_x(t, \nu; h) = \int_{-\infty}^{+\infty} x(u) h^*(u - t) e^{-i2\pi\nu u} du.$$

Le signal peut être reconstruit parfaitement à partir de sa STFT à l'aide d'une fenêtre de synthèse  $g(t)$ , en général différente de  $h(t)$ , en appliquant la formule [Daubechies, 1992; Flandrin, 1998] :

$$x(t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F_x(u, \xi; h) g(t - u) e^{i2\pi\xi u} du d\xi,$$

où la fenêtre de synthèse satisfait l'équation :

$$\int_{-\infty}^{+\infty} g(t) h^*(t) dt = 1. \quad (7.1)$$

Si la fenêtre  $h(t)$  est normée, un choix évident pour la fenêtre de synthèse est :  $g(t) = h(t)$ . Suivant l'application, on peut également utiliser la condition (7.1) pour construire des fenêtres présentant de meilleures propriétés [Nuttall, 1981]. Dans la pratique, la STFT peut être discrétisée dans le plan temps-fréquence et utilisée sur un signal discret  $x(k)$ , où  $k \in \mathbb{Z}$ . Dans ce cas, la décomposition revient à découper temporellement le signal à analyser en segments de longueur  $L$  se recouvrant. Si deux segments consécutifs se recouvrent sur  $L - N$  points, alors on peut voir cette opération comme l'application d'une fenêtre glissante de longueur  $L$  se décalant de  $N$  points à chaque étape. Ensuite, chaque segment est fenêtré puis subit une transformée de Fourier discrète. La reconstruction peut alors se faire en considérant la transformée de Fourier inverse de chacun des blocs puis en sommant les résultats avec une pondération appropriée. Cette pondération peut également être vue comme une fenêtre de synthèse. Cette méthode est appelée la synthèse par "recouvrement et sommation" (*overlap-add* en anglais). Une condition suffisante et aisée à utiliser, assurant que le signal puisse être reconstruit parfaitement, est que la fenêtre discrète d'analyse  $h(k)$  ne s'annule pas. Il existe des conditions moins restrictives permettant d'obtenir la reconstruction parfaite [Allen, 1977; Crochiere, 1980], ainsi que des travaux visant à choisir de manière optimale les fenêtres d'analyse/synthèse pour une application donnée, par exemple en traitement de la parole [Portnoff, 1981].

Nous proposons ici d'utiliser les bancs de filtres d'analyse/synthèse RIF redondants optimisés construits au chapitre 4 pour réaliser des représentations temps-fréquence inversibles. Pour ce faire, nous allons considérer le banc de filtres d'analyse défini à la section 3.5.2 avec la fenêtre  $h_{a_1}$ , ainsi que les bancs de filtres de synthèses symétriques avant et après optimisation à l'aide de la fonction de coût  $\widetilde{J}_{ts}$ . La structure de ce banc



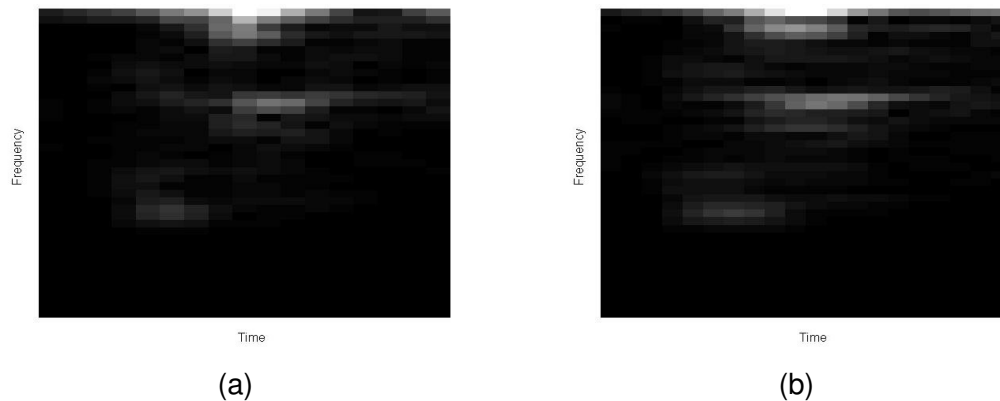


FIGURE 7.20 – Représentation temps-fréquence avec (a) une STFT et (b) une décomposition par banc de filtres.

de filtres d'analyse est très proche de celle d'une transformée de Fourier locale fenêtrée à l'aide d'une fenêtre de Hamming ce qui permet une comparaison plus aisée. Cette structure est également à rapprocher des travaux sur les transformées locales en cosinus (Local Cosine Transforms - LCT) [Aharoni *et al.*, 1993; Malvar, 1992].

L'une des finalités de ce type de méthode étant de traiter des signaux en temps réel, nous avons choisi de limiter le nombre de fréquences utilisées lors de la décomposition. Des exemples de représentations temps-fréquence obtenues avec la STFT et en utilisant le banc de filtres d'analyse de la section 3.5.2 avec la fenêtre  $h_{a_1}$  sont visibles à la figure 7.20.

### 7.3.3 Résultats

Nous allons maintenant présenter des résultats de reconstruction de signaux après sélection dans le plan temps-fréquence de certains coefficients.

**7.3.3-)-1 Données synthétiques.** Dans un premier temps, nous avons construit un signal synthétique à partir d'une sinusoïde fenêtrée à laquelle a été ajoutée d'autre signaux sinusoïdaux (avec des fréquences et des amplitudes différentes) ainsi que du bruit blanc gaussien. Les signaux sinusoïdaux parasites peuvent représenter des événements ayant lieu dans le moteur (des injections par exemple) et le bruit aléatoire représente le bruit de fond. Dans le plan temps-fréquence, les coefficients correspondant au signal d'origine ont été conservés pour les décompositions utilisant la STFT et le banc de filtres d'analyse choisi précédemment. Pour ce faire, nous avons supposé connue la bande de fréquence dans laquelle se situe le signal d'intérêt, et avons simplement choisi de conserver les coefficients dont l'amplitude dépassait un seuil.

La taille des sauts de la STFT et le nombre de fréquences ont été choisis pour correspondre à ceux de la décomposition par bancs de filtres. Plus précisément, en notant  $N$  le facteur de sous-échantillonnage du banc de filtres et  $k'$  son facteur de redondance alors la taille des sauts de la STFT est aussi égale à  $N$  et le nombre de fréquences est  $k'N/2$ .

Finalement, l'opération de reconstruction a été appliquée en utilisant la STFT inverse et les bancs de filtres symétriques de synthèse avant et après optimisation.

Le rapport signal sur bruit (SNR) a été calculé et les résultats obtenus sont fournis dans le tableau 7.1. Un premier constat est que la méthode par bancs de filtres optimisés (BFSO) atteint un SNR de 14.4 dB et donne donc un meilleur résultat que la STFT qui a un SNR de 7.9 dB. Un autre élément intéressant à remarquer est que, une fois encore, l'utilisation d'un banc de filtres optimisé améliore les résultats par rapport à la version obtenue par pseudo-inverse (BFSPI) : dans cette application le gain est de 2 dB. Le signal d'origine, le signal bruité ainsi que les résultats de reconstruction des différentes méthodes sont visibles à la figure 7.21.

Cette application sur des données synthétiques a permis de mettre en évidence l'intérêt de la reconstruction par banc de filtres optimisés. Voyons maintenant les résultats obtenus sur des données réelles.

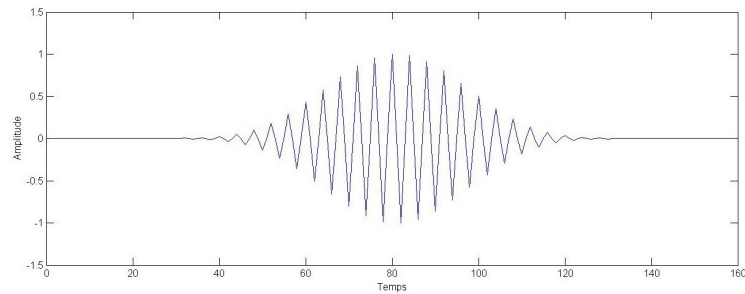
	Données bruitées	STFT	BFSPI	BFSO
SNR	4.45 dB	7.9 dB	12.2 dB	14.4 dB

TABLE 7.1 – Rapport signal sur bruit (SNR) en dB après reconstruction par les différentes méthodes.

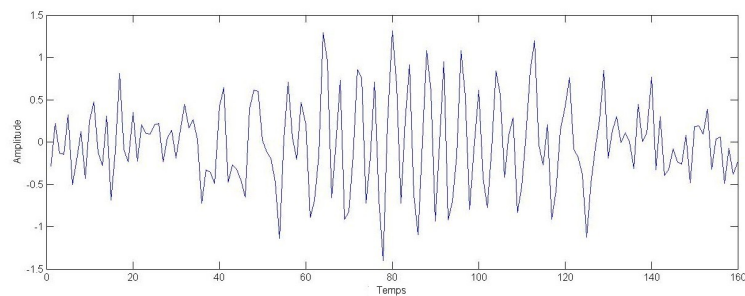
**7.3.3-)-2 Données réelles.** Pour cette application, nous avons travaillé sur le signal de cliquetis correspondant au gradient de pression représenté à la figure 7.22. Le but est ici de réussir à partir du signal de cliquetis à obtenir une information sur le début de la combustion.

Pour ce faire, la stratégie adoptée a été de ne garder que les coefficients de la représentation temps-fréquence les plus grands (en module) situés dans les bandes de basses fréquences ainsi que leurs voisins immédiats. En effet, on sait que c'est dans cette bande de fréquence qu'on a le plus de chance d'observer la détonation ayant eu lieu dans la chambre de combustion. Nous avons utilisé un banc de filtres d'analyse avec les mêmes paramètres que dans l'application précédente et le banc de filtres optimisé avec la fonction de coût  $\tilde{J}_{ts}$ .

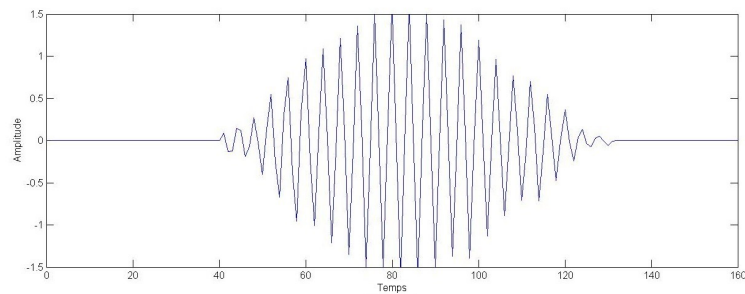
Les résultats de reconstruction montrent que les deux méthodes sont efficaces pour isoler le pic du gradient de pression, ce qui valide notre stratégie de sélection. Visuellement, on peut également constater que la reconstruction par banc de filtres optimisés est mieux localisée qu'en utilisant la reconstruction par STFT. La localisation du début de la combustion sera donc plus précise avec la reconstruction par banc de filtres optimisés.



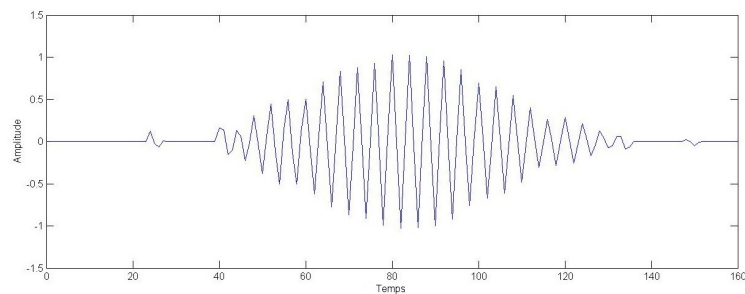
(a)



(b)



(c)



(d)

FIGURE 7.21 – (a) Signal d'origine, (b) signal bruité, (c) signal reconstruit avec la STFT, (d) signal reconstruit avec un banc de filtres optimisé.



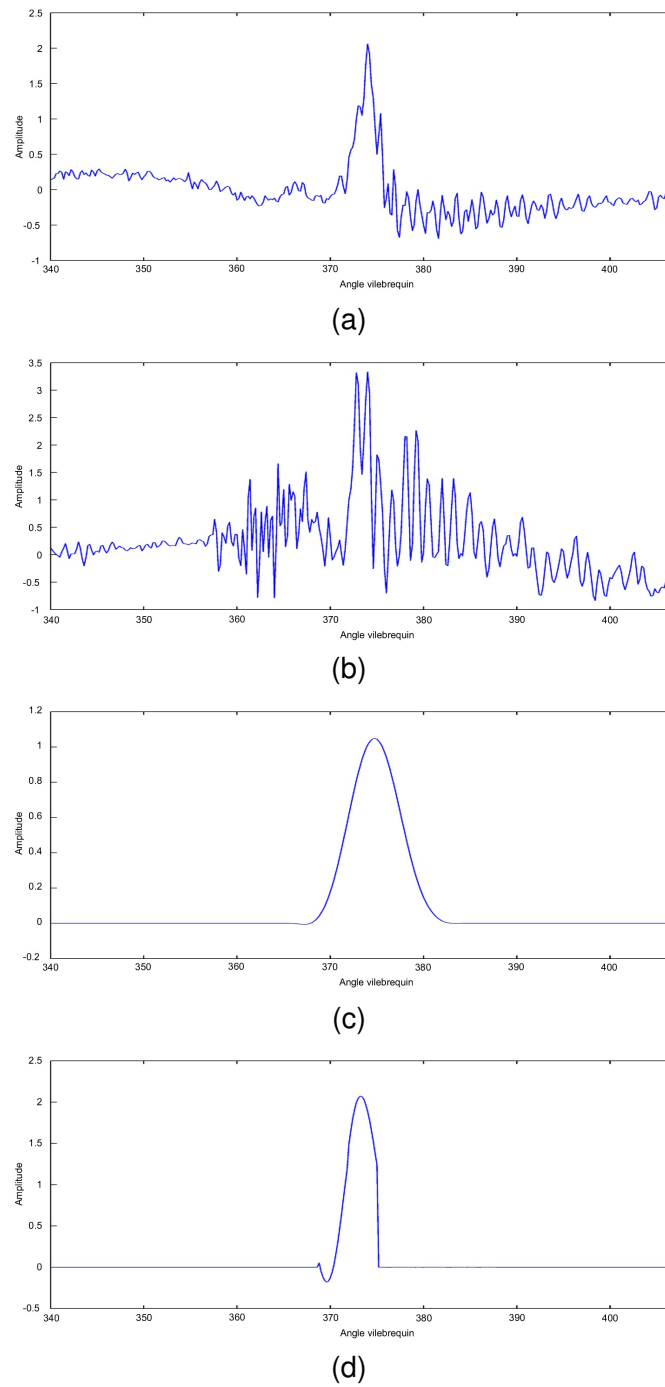


FIGURE 7.22 – (a) Signal du gradient de pression, (b) signal de cliquetis, (c) signal reconstruit avec la STFT et (d) signal reconstruit avec un banc de filtres de synthèse optimisé.



---

## Chapitre 8

# Conclusion

---

Les travaux menés dans cette thèse ont conduit à différents apports tant au niveau théorique que pratique. Nous commencerons par résumer les principales contributions de ce travail, puis nous verrons quelques perspectives ouvertes par les résultats présentés.

### 8.1 Résumé des travaux réalisés

#### Construction de filtres

Dans un premier temps, nous nous sommes intéressés à l'existence et à la construction de bancs de filtres de synthèse RIF, inverses d'un banc de filtres d'analyse RIF suréchantillonné complexe donné. L'étude de la matrice polyphase d'un tel banc de filtres d'analyse nous a permis de proposer une première méthode testant l'inversibilité d'une matrice polynomiale. Cette méthode est complétée par une seconde procédure mettant seulement en œuvre des résolutions de systèmes linéaires et permettant de construire *une* matrice polynomiale inverse particulière d'ordre minimal, et donc un banc de filtres de synthèse à reconstruction parfaite dont le support est de longueur minimale.

Travailler dans un contexte redondant fait que le problème d'inversion étudié ne possède pas une solution unique. Cette observation nous a ensuite amené à rechercher parmi l'ensemble des bancs de filtres de synthèse des versions offrant des propriétés désirables (bonne sélectivité fréquentielle ou bonne localisation temporelle). En étudiant le système linéaire utilisé pour construire un inverse particulier, nous avons pu déduire une paramétrisation de l'ensemble des bancs de filtres de synthèse à reconstruction parfaite ayant une longueur de support donné. Enfin, en minimisant certaines fonctions de coût adaptées au type de propriété recherchée pour les filtres de synthèse, nous avons pu proposer, *via* un algorithme de gradient, une méthode de sélection de banc de filtres de synthèse, rapide et simple à mettre en œuvre. Les bancs de filtres optimisés ont été testés pour de nombreuses applications (débruitage, filtrage directionnel) et fournissent de meilleures performances que les filtres non optimisés.

Les résultats obtenus dans le cas unidimensionnel ont ensuite été étudiés dans le

cas multidimensionnel. Cette étude nous a conduit à proposer une méthode employant la notion de résultant pour tester l'inversibilité d'un banc de filtres sous-échantillonnés rectangulairement. La méthode obtenue a été comparée à celle utilisant les bases de Gröbner. Si la méthode proposée ne permet pas de conclure dans quelques cas (peu probables en pratique), elle présente néanmoins des avantages en terme de complexité. Nous avons enfin été en mesure d'étendre la plupart des propriétés unidimensionnelles : construction d'un inverse particulier, inversion dans le cas symétrique et paramétrisation de l'ensemble des bancs de filtres de synthèse.

### Méthodes pour le débruitage

Nous avons étudié l'estimateur sans biais du risque de Stein dans le cadre de transformations redondantes et complexes. En particulier, nous nous sommes intéressés à deux estimateurs agissant, l'un, sur les coefficients après décomposition et, l'autre, sur les échantillons après reconstruction. Ces deux estimateurs, égaux dans le cas orthogonal, sont utilisés avec le formalisme SURE-LET et des bancs de filtres d'analyse/synthèse redondants complexes et conduisent aux méthodes de débruitage FB-SURELET-C et FB-SURELET-E. Ces méthodes donnent de bons résultats en débruitage au sens où elles approchent les résultats obtenus avec l'oracle et où elles se comparent favorablement aux méthodes récentes de débruitage que nous avons testées.

Une autre méthode de réjection de bruit a ensuite été proposée pour réaliser un filtrage anisotrope d'images ou de cube de données. Ce filtrage permet de préserver des structures directionnelles d'intérêt (soit par détection automatique de la direction dominante locale, soit en utilisant un *a priori* fourni par l'utilisateur), en éliminant les structures parasites ainsi qu'une partie du bruit aléatoire.

### Applications à des données industrielles

Le filtrage directionnel a d'abord été employé sur des images et des cubes de données sismiques. Les résultats montrent que les structures principales des données sont bien conservées. L'utilisation d'un attribut de désordre permet également de ne pas trop filtrer dans les zones chaotiques où plusieurs structures d'importance similaire sont mélangées. Cette méthode a été intégrée dans des chaînes de traitements (incluant des étapes de filtrage, de squelettisation ou de recherche de directions) et permettant, par exemple, d'estimer les âges relatifs des différentes strates du sous-sol ou de reconnaître automatiquement des structures d'intérêt pour les géologues. Cette dernière application a donné lieu au dépôt d'un brevet.

Le filtrage directionnel a également été employé sur des données issues de microscopie électronique à transmission afin d'isoler des structures anisotropes importantes pour l'analyse de réactions de catalyse.

Enfin, nous avons utilisé les bancs de filtres d'analyse/synthèse que nous avons construits, pour réaliser des analyses temps-fréquences de signaux vibratoires issus de moteurs et isoler après reconstruction certaines parties du signal intéressantes pour le contrôle de combustion du moteur.

## 8.2 Perspectives

### Optimisation de bancs de filtres

Les fonctions de coût proposées dans cette thèse constituent de bons exemples de critères intéressants pour optimiser les filtres de synthèse. D'autres fonctions de coût peuvent également être envisagées. Une première idée serait de combiner (par exemple de manière linéaire) les fonctions utilisées dans ce travail afin d'optimiser conjointement différentes caractéristiques des filtres. Une autre possibilité consisterait à choisir des fonctions de coût telles que le banc de filtres de synthèse optimisé résultant présente de bonnes propriétés pour une application donnée, par exemple pour le débruitage par SURELET.

Une autre piste à considérer pourrait être de se placer dans un cadre de reconstruction quasi-parfaite. Les applications que nous avons considérées dans cette thèse impliquent des modifications des coefficients transformés. Une reconstruction quasi-parfaite permettrait donc de réaliser les mêmes opérations tout en augmentant les degrés de liberté dans la construction des filtres et donc, potentiellement, leur qualité après optimisation.

Enfin, il serait également intéressant de tester les bancs de filtres suréchantillonnés optimisés que nous avons construits dans le cadre de bancs de filtres itérés [Blu, 1996].

### Bancs de filtres multidimensionnels

Au chapitre 5, nous avons vu comment obtenir une paramétrisation des bancs de filtres de synthèse à reconstruction parfaite. On dispose donc de tous les éléments pour mettre en place une étape d'optimisation multidimensionnelle, en suivant une démarche similaire au cas unidimensionnel. Pour ce faire, il faudrait également introduire des fonctions de coût adaptées au cas multidimensionnel.

Il serait également intéressant de reprendre les résultats présentés et de les étendre à des échantillonnages, redondances et recouvrements différents suivant les directions.

Enfin, lors de la discussion sur la complexité de l'algorithme multidimensionnel de test d'inversibilité, nous avons évoqué la possibilité de travailler uniquement de manière numérique, en particulier en appliquant des méthodes de calculs de déterminants sans divisions [Berkowitz, 1984; Mahajan, Vinay, 1997]. Il serait donc intéressant de mettre en œuvre ces méthodes afin d'obtenir une comparaison plus précise des temps de calcul de l'algorithme proposé et de la méthode par base de Gröbner.

### Méthodes SURE-LET

Lors de la comparaison des méthodes FB-SURELET-E et FB-SURELET-C, nous avons constaté que plus on s'écarte d'une transformation orthogonale, plus les résultats des deux méthodes diffèrent. Un moyen de faire en sorte que les résultats soient plus proches (ce qui serait intéressant car la méthode FB-SURELET-C s'exécute plus rapidement que FB-SURELET-E) serait de sélectionner, via l'étape d'optimisation du banc de filtres de synthèse, des couples d'analyse/synthèse les plus orthogonaux possibles. Pour ce faire, on pourrait utiliser une fonction de coût basée sur l'estimation de l'écart à "l'orthogonalité"

que nous avons utilisée au chapitre 6 ou bien d'autres mesures comme celles présentées dans [Lightstone *et al.*, 1997; MOREAU de Saint-Martin *et al.*, 1999].

Il serait également intéressant de faire une étude plus complète des résultats de débruitage par les méthodes proposées, avec différents "types" de bancs de filtres. Ceux qui ont été utilisés dans ce travail sont bien adaptés aux structures orientées, mais peuvent faire apparaître des artéfacts dans les zones homogènes. L'utilisation de bancs de filtres permettant de bien représenter à la fois les zones texturées et homogènes, éventuellement de manière adaptative [de Queiroz, Rao, 1993], pourrait permettre d'améliorer les résultats.

Des améliorations concernant la méthode SURE-LET elle-même peuvent être envisagées. En particulier, la recherche de nouvelles fonctions non linéaires pour le seuillage et une analyse des caractéristiques importantes que doivent vérifier ces fonctions, pourraient permettre de proposer des fonctions mieux adaptées.

## Applications

L'utilisation de transformées à recouvrement pour de gros volumes de données permet de traiter en parallèle plusieurs sous-volumes se recouvrant en partie et ainsi d'obtenir un traitement plus efficace en temps de calculs sur des architectures utilisant plusieurs processeurs. Une idée pour éviter d'avoir à considérer numériquement des sous-volumes se recouvrant serait d'ajouter un pré-traitement, consistant à *replier* les données à traiter à la manière de [Tran *et al.*, 2000], aux méthodes que nous avons proposées.

La méthode de filtrage directionnel proposée au chapitre 6 pourrait également être améliorée. Il serait en particulier intéressant, pour des applications de filtrage sur des données devant présenter des structures directionnelles continues, de prendre en compte lors de l'estimation de directions, celles dominantes des "blocs" proches de celui considéré.

Un autre aspect intéressant serait d'essayer de diversifier un peu plus les domaines d'application des bancs de filtres que nous avons construits. Une première possibilité serait d'utiliser les bancs de filtres optimisés pour la réalisation de codes à descriptions multiples (MDC - *Multiple Description Coding*) à la manière de [Petrison *et al.*, 2005]. Des articles récents ont également permis de mettre en lumière certains liens entre la théorie des bancs de filtres et le codage réseau. Ainsi dans [Jaggi *et al.*, 2004], les auteurs introduisent les *filter-bank network codes* permettant de généraliser le codage réseau linéaire et dans [Marinkovic, Guillemot, 2008], des codes par bancs de filtres sur-échantillonnés sont proposés pour réaliser du codage source-canal conjoint. Une application intéressante pourrait, là encore, être de combiner les bancs de filtres optimisés que nous avons construits, avec ces méthodes.

L'application à l'analyse temps-fréquence que nous avons proposée au chapitre 7 présente l'avantage d'être très simple. On pourrait néanmoins essayer de la raffiner, par exemple, par l'utilisation de bancs de filtres mieux adaptés au type de données étudiées ou présentant une meilleure localisation et par des méthodes de sélection des informations à conserver plus évoluées. Il pourrait également être intéressant de comparer les résultats obtenus avec des méthodes utilisant des représentations différentes, comme une représentation temps-octave de type Capon [Martin *et al.*, 1995]. Une dernière application que l'on pourrait évoquer serait de faire de la séparation de sources à l'aide des bancs de filtres optimisés construits dans cette thèse [Nguyen *et al.*, 2005].

---

## Annexe A

# Algorithme de test d'inversibilité

---

Nous allons présenter dans cette annexe une version plus détaillée et plus proche du pseudo-code de l'algorithme décrit à la section 5.2.2, permettant de déterminer dans un grand nombre de cas si un banc de filtres d'analyse multidimensionnel est inversible ou non.

Donnons, tout d'abord, quelques précisions sur les notations utilisées dans cet algorithme. L'ensemble  $\mathcal{F}$  sert à stocker les indices des polynômes non dégénérés déjà extraits, l'entier  $k$  sert d'indice pour identifier les sous-matrices maximales successivement extraites (et donc les polynômes  $A_k$ ),  $m$  sert d'indice pour parcourir les polynômes  $A_{k_m}$  non dégénérés qui ont déjà été extraits et stockés dans  $\mathcal{F}$ ,  $d$  est une variable booléenne (un drapeau) permettant de savoir si un polynôme non dégénéré a déjà été extrait dans la boucle principale et  $\mathcal{S}$  représente l'ensemble des zéros communs potentiels. Enfin, la notation  $\{A_k(\mathcal{S}) = 0\}$  représente l'ensemble :  $\{(a, b) \in \mathcal{S} \mid A_k(a, b) = 0\}$ .

Soit une matrice  $\mathbf{H}[z_1, z_2] \in \mathbb{C}[\mathbf{z}, \mathbf{z}^{-1}]^{N^2 \times M^2}$ , la procédure principale est alors la suivante.

1. *Initialisation :*

$\mathcal{F} \leftarrow \emptyset$ ,  $d \leftarrow 0$  et  $\mathcal{S} \leftarrow (\mathbb{C}^*)^2$

2. *Boucle principale :*

Pour  $k$  allant de 1 à  $\binom{M^2}{N^2}$ , faire :

- Si  $\mathcal{S} = \emptyset$ , aller en 3.
- Extraire une nouvelle sous-matrice maximale  $\mathbf{H}_{e_k}[z_1, z_2]$  de  $\mathbf{H}[z_1, z_2]$ .  
Calculer  $A_k(z_1, z_2) = \det(\mathbf{H}_{e_k}[z_1, z_2])$ .
- Si le polynôme  $A_k$  est constant nul, retour en 2.
- Si le polynôme  $A_k$  est constant non nul,  $\mathcal{S} \leftarrow \emptyset$ , retour en 2.
- Si  $\deg_{z_1}(A_k) = 0$ , soient  $(b_i)_{i \in I}$  les racines de  $A(z_1, z_2)$  en  $z_2$ .  
On a alors :  $\mathcal{S} \leftarrow \mathcal{S} \cap (\mathbb{C}^* \times \{b_i\}_{i \in I})$ .  
Retour en 2.
- Si  $\deg_{z_2}(A_k) = 0$ , soient  $(a_i)_{i \in I}$  les racines de  $A(z_1, z_2)$  en  $z_1$ .  
On a alors :  $\mathcal{S} \leftarrow \mathcal{S} \cap (\{a_i\}_{i \in I} \times \mathbb{C}^*)$ .  
Retour en 2.

- Si  $d = 0$ , alors  $\mathcal{F} \leftarrow \{k\}$ ,  $d \leftarrow 1$  et retour en 2.
  - Si  $\mathcal{S} = (\mathbb{C}^*)^2$  :
    - (a) Pour  $m$  allant de 1 à  $|\mathcal{F}|$ , faire :
      - Prendre  $k_m$  égal au  $m^{\text{ème}}$  élément de  $\mathcal{F}$ .
      - Calculer  $C_1(z_1) \leftarrow \text{Res}_{z_2}(A_{k_m}, A_k)(z_1)$  et  $C_2(z_2) \leftarrow \text{Res}_{z_1}(A_{k_m}, A_k)(z_2)$ .
      - Si  $C_1$  et  $C_2$  sont des polynômes nuls, alors :
        - Si  $m = |\mathcal{F}|$  alors  $\mathcal{F} \leftarrow \mathcal{F} \cup \{k\}$  et retour en 2.
        - sinon aller en (b).
    - (b) Si  $C_1$  est nul et  $C_2$  non nul, rechercher les zéros non nuls  $(b_j)_{j \in J}$  de  $C_2$ .  
 Pour  $j \in J$  rechercher les racines  $(a_{k,j,i})_{i \in I_{j,1}}$  de  $A_k(z_1, b_j)$  et  $(a_{k_m,j,i})_{i \in I_{j,2}}$  de  $A_{k_m}(z_1, b_j)$ . On a alors :
 
$$\mathcal{S} \leftarrow \left( \bigcup_{j \in J} (\{(a_{k,j,i})_{i \in I_{j,1}}\} \cap \{(a_{k_m,j,i})_{i \in I_{j,2}}\}) \times \{b_j\} \right).$$
    - (c) Si  $C_1$  est non nul et  $C_2$  nul, rechercher les zéros non nuls  $(a_i)_{i \in I}$  de  $C_1$ .  
 Pour  $i \in I$  rechercher les racines  $(b_{k,i,j})_{j \in J_{i,1}}$  de  $A_k(a_i, z_2)$  et  $(b_{k_m,i,j})_{j \in J_{i,2}}$  de  $A_{k_m}(a_i, z_2)$ . On a alors :
 
$$\mathcal{S} \leftarrow \left( \bigcup_{i \in I} \{a_i\} \times (\{(b_{k,i,j})_{j \in J_{i,1}}\} \cap \{(b_{k_m,i,j})_{j \in J_{i,2}}\}) \right).$$
    - (d) Si  $C_1$  et  $C_2$  sont non nuls, rechercher les zéros non nuls  $(a_i)_{i \in I}$  de  $C_1$  et  $(b_j)_{j \in J}$  de  $C_2$ , avec  $I$  et  $J$  finis (éventuellement vides).  
 Rechercher les couples  $(a_i, b_j)_{(i,j) \in I_2 \times J_2}$  annulant conjointement  $A_k$  et  $A_{k_m}$ , où  $I_2 \subset I$  et  $J_2 \subset J$ .  
 On a alors :  $\mathcal{S} \leftarrow \{(a_i, b_j)\}_{(i,j) \in I_2 \times J_2}$ .
    - (e)  $\mathcal{F} \leftarrow \mathcal{F} - \{k_m\}$
    - (f) Pour  $m$  allant de 1 à  $|\mathcal{F}|$ , faire :
      - Prendre  $k_m$  égal au  $m^{\text{ème}}$  élément de  $\mathcal{F}$ .
      - $\mathcal{S} \leftarrow \text{reduire}(\mathcal{S}, A_{k_m})$ .
      - $\mathcal{F} \leftarrow \mathcal{F} - \{k_m\}$
      - Si  $\mathcal{S} = \emptyset$ , aller en 3.
  - sinon,  $\mathcal{S} \leftarrow \text{reduire}(\mathcal{S}, A_k)$ .
3. **Conclusion :**  
 Si  $\mathcal{S} = \emptyset$  alors  $\mathbf{H}[z_1, z_2]$  est inversible à gauche.  
 Si  $\mathcal{S} \neq \emptyset$  et  $\mathcal{F} = \emptyset$  alors  $\mathbf{H}[z_1, z_2]$  n'est pas inversible à gauche.  
 Si  $\mathcal{S} \neq \emptyset$  et  $\mathcal{F} \neq \emptyset$ , alors on ne peut pas conclure.

Dans l'algorithme ci-dessus nous avons employé la sous-fonction *reduire* suivante :

Fonction  $\mathcal{S} \leftarrow \text{reduire}(\mathcal{S}, A)$

1. Si  $\mathcal{S}$  est de la forme :  $\{(a_i, b_i)_{i \in I}\}$  avec  $I$  fini, alors  $\mathcal{S} \leftarrow \{A(\mathcal{S}) = 0\}$ .
2. Si  $\mathcal{S}$  est de la forme :  $\{(a_i)_{i \in I}\} \times \mathbb{C}^*$  avec  $I$  fini, alors pour tout  $i \in I$ , soient le polynôme :  $Q_i(z_2) \leftarrow A(a_i, z_2)$  et ses racines :  $(b_j)_{j \in J_i}$ .  
 On a alors :  $\mathcal{S} \leftarrow \mathcal{S} \cap \left( \bigcup_{i \in I} (\{(a_i)_{i \in I}\} \times \{(b_j)_{j \in J_i}\}) \right)$ .



3. Si  $\mathcal{S}$  est de la forme :  $\mathbb{C}^* \times \{(b_j)_{j \in J}\}$  avec  $J$  fini, alors pour tout  $j \in J$ , soient le polynôme :  $P_j(z_1) \leftarrow A(z_1, b_j)$  et ses racines :  $(a_i)_{i \in I_j}$ .  
On a alors :  $\mathcal{S} \leftarrow \mathcal{S} \cap \left( \bigcup_{j \in J} (\{(a_i)_{i \in I_j}\} \times \{(b_j)_{j \in J}\}) \right)$ .

**Remarque A.1** Certains ensembles  $J_i$  de l'étape 2. de la fonction *reduire* peuvent éventuellement être infinis (cas où le polynôme  $Q_i$  est identiquement nul). La même remarque est valable à l'étape 3. pour les ensembles  $I_j$ .

**Remarque A.2** Afin de matérialiser les différentes parties de la boucle principale, nous avons tracé dans la marge des traits verticaux correspondant à des parties introduites dans la description de l'algorithme de la section 5.2.2-b). Le trait rouge (un seul trait) représente la première partie et les deux traits bleus la deuxième partie. La troisième partie de l'algorithme a été isolée dans la sous-fonction *reduire*.

**Remarque A.3** On utilise l'ensemble  $\mathcal{F}$  afin de garder une trace des polynômes non dégénérés déjà extraits. Deux choix pour le stockage de cette information sont alors possibles. Une première possibilité consiste à garder en mémoire seulement l'indice  $k$  du polynôme  $A_k$  et à recalculer le déterminant de la sous matrice  $\mathbf{H}_{e_k}[z_1, z_2]$  à chaque fois que l'on en a besoin. La deuxième possibilité est de stocker directement le polynôme extrait. La première présente l'avantage d'être peu coûteuse en espace mémoire nécessaire mais devient vite lourde en calculs, tandis que la seconde est gourmande en espace mémoire, mais économe en temps de calcul.



---

# Table des figures

1.1	Différents types de données : (a) Signal de cliquetis d'un moteur, (b) image de microscopie électronique par transmission d'un catalyseur et (c) cube sismique. . . . .	17
2.1	Principe d'une transformée par blocs (avec un bloc de taille 2). . . . .	26
2.2	Effet de damier dans la compression JPEG : (a) image originale, (b) image fortement compressée avec JPEG. . . . .	28
2.3	Principe d'une transformée à recouvrement avec un recouvrement $k = 2$ et des blocs élémentaires de tailles $N = 2$ . . . . .	28
2.4	Schéma d'un banc de filtres d'analyse/synthèse. . . . .	30
2.5	Schéma d'un banc de filtres $M$ -bandes. . . . .	31
2.6	Schéma direct et schéma polyphase d'un banc de filtres d'analyse. . . . .	32
2.7	Exemple de décomposition/reconstruction par ondelettes. . . . .	34
2.8	Exemple de décomposition/reconstruction par paquet d'ondelettes. . . . .	36
3.1	Parcours des couples $(p_1, p_2)$ quand $p$ augmente. . . . .	50
3.2	(a) Réponses impulsionnelles et (b) réponses fréquentielles d'un banc de filtres d'analyse de type GenLOT. . . . .	56
3.3	(a) Réponses impulsionnelles et (b) réponses fréquentielles du banc de filtres de synthèse obtenu par pseudo-inverse d'un banc de filtres d'analyse de type GenLOT. . . . .	57
3.4	Deux exemples de fenêtres d'analyse : (a) fenêtre $h_{a_1}$ et (b) fenêtre $h_{a_2}$ . . . . .	58
3.5	Réponses fréquentielles des filtres d'analyse en utilisant (a) la fenêtre $h_{a_1}$ et (b) la fenêtre $h_{a_2}$ . . . . .	59
3.6	Module des réponses impulsionnelles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est $h_{a_1}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse dans le cas symétrique. . . . .	60
3.7	Réponses fréquentielles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est $h_{a_1}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse symétrique. . . . .	61
3.8	Module des réponses impulsionnelles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est $h_{a_2}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse dans le cas symétrique. . . . .	62
3.9	Réponses fréquentielles de différents bancs de filtres de synthèse (cas où la fenêtre d'analyse est $h_{a_2}$ ) : (a) par pseudo-inverse, (b) après symétrisation et (c) par pseudo-inverse symétrique. . . . .	63

4.1	Premier exemple (avec le banc de filtres de la section 3.5.1) : réponses fréquentielles du banc de filtres de synthèse obtenu (a) par pseudo-inverse et (b) et après optimisation avec la fonction $\tilde{J}_t$ . . . . .	79
4.2	Premier exemple : Réponses impulsionnelles du banc de filtres de synthèse obtenu (a) par pseudo-inverse et (b) et après optimisation avec la fonction $\tilde{J}_t$ . . . . .	80
4.3	Deuxième exemple (banc de filtres de la section 3.5.2 avec $h_{a1}$ ) : réponses fréquentielles des filtres de synthèse (a) avant optimisation (pseudo-inverse) et (b) après optimisation avec la fonction $\tilde{J}_f$ . . . . .	81
4.4	Deuxième exemple : module des réponses impulsionnelles des filtres de synthèses (a) avant et (b) après optimisation avec la fonction $\tilde{J}_f$ . . . . .	82
4.5	Deuxième exemple : parties réelles et imaginaires des réponses impulsionnelles de filtres de synthèses, étendus en 2D de manière séparable, dans les cas pseudo-inverse (a)(b) et optimisé (c)(d). . . . .	83
4.6	Troisième exemple (banc de filtres de la section 3.5.2 avec $h_{a2}$ ) : module des réponses impulsionnelles des filtres de synthèses (a) avant et après optimisation (b) avec la fonction de coût $\tilde{J}_{ts}$ et (c) avec la fonction de coût $\tilde{J}_{fs}$ . . . . .	84
4.7	Troisième exemple : réponses fréquentielles des filtres de synthèses (a) avant et après optimisation (b) avec la fonction de coût $\tilde{J}_{ts}$ et (c) avec la fonction de coût $\tilde{J}_{fs}$ . . . . .	85
5.1	Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des $12 \times 12$ filtres d'analyse. . . . .	108
5.2	Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des $12 \times 12$ filtres de synthèse. . . . .	109
5.3	Exemple de couple de fenêtres (a) d'analyse ( $W_c$ ) et (b) synthèse non séparables. . . . .	112
5.4	Exemple de couple de fenêtres (a) d'analyse ( $W_k$ ) et (b) synthèse non séparables. . . . .	113
5.5	Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des filtres d'analyse dans le cas $k = k'$ . . . . .	115
5.6	Représentation spatiale des parties réelles et imaginaires des réponses impulsionnelles des filtres de synthèse dans le cas $k = k'$ . . . . .	116
6.1	Allures de la fonction de seuillage $x \mapsto \tau_2(x, 0)$ pour $m_0 = 2$ en rouge (pointillés longs) et $m_0 = 12$ en bleu et de la fonction de seuillage dur en vert (pointillés courts). . . . .	136
6.2	Modules des coefficients de décomposition, regroupés par sous-bandes, de l'image Cameraman. . . . .	137
6.3	Exemples de manières d'apparier des sous-bandes de l'image de la figure 6.2 de manière symétrique. Les sous-bandes représentées en gris sont regroupées. . . . .	139
6.4	Images utilisées pour les simulations : (a) Lena, (b) Barbara, (c) Boat, (d) Cameraman, (e) Baboon et (f) Fingerprint. . . . .	142

6.5	SNR après débruitage du signal de la figure 6.6 pour plusieurs réalisations différentes de bruit ( $\sigma = 30$ ) par FB-SURELET-E (diamants bleus) et l'oracle (cercles verts). . . . .	143
6.6	Comparaison entre les méthodes proposées et leurs oracles. De haut en bas : signal d'origine, signal bruité (bruit blanc gaussien $\sigma = 30$ ), signaux débruités par la méthode FB-SURELET-C (17,5dB) puis avec son oracle (18,1dB) et par FB-SURELET-E (18,2dB) puis avec son oracle (18,3dB). . . .	144
6.7	(a) Image originale (Barbara), (b) image bruitée ( $\sigma = 30$ ), (c) image débruitée avec Curvelet, (d) image débruitée avec BiShrink, (e) image débruitée avec OWT SURE-LET et (f) image débruitée avec FB-SURELET-E. . . . .	152
6.8	Détail de (a) l'image originale (Barbara), (b) l'image bruitée ( $\sigma = 30$ ), (c) l'image débruitée avec Curvelet, (d) l'image débruitée avec BiShrink, (e) l'image débruitée avec OWT SURE-LET et de (f) l'image débruitée avec FB-SURELET-E. . . . .	153
6.9	(a) Image originale (Cameraman), (b) image bruitée ( $\sigma = 50$ ), (c) image débruitée avec Curvelet, (d) image débruitée avec BiShrink, (e) image débruitée avec OWT SURE-LET et (f) image débruitée avec FB-SURELET-E. . . . .	154
6.10	Détail de (a) l'image originale (Cameraman), (b) l'image bruitée ( $\sigma = 50$ ), (c) l'image débruitée avec Curvelet, (d) l'image débruitée avec BiShrink, (e) l'image débruitée avec OWT SURE-LET et de (f) l'image débruitée avec FB-SURELET-E. . . . .	155
6.11	Deux images directionnelles avec les transformées en cosinus et de Fourier associées. . . . .	157
6.12	Trois stratégies possibles pour la sélection de coefficients pour une direction dominante d'angle $\pi/4$ . . . . .	158
6.13	Image des (logarithmes des) modules des coefficients (a) après transformation, (b) après sélection de la direction principale et (c) après le seuillage dur. . . . .	159
6.14	(a) Image artificielle originale, (b) image bruitée et (c) résultat du filtrage directionnel. . . . .	159
6.15	(a) Image artificielle bruitée, (c) résultat du filtrage de la section 6.2.2, (d) résultat du filtrage avec l'estimation de mélange et (b) image de différence entre les deux résultats. . . . .	161
7.1	Dispositif d'acquisition et exemple de mesure. . . . .	164
7.2	Une image sismique après pré-traitement. . . . .	165
7.3	Un échantillon d'image sismique et sa modélisation associée. . . . .	166
7.4	Un exemple d'image du modèle perturbée par du bruit. . . . .	166
7.5	Exemple d'une image sismique réelle et d'un résultat de filtrage sur cette image. . . . .	167
7.6	Détails d'images sismiques réelles et de résultats du filtrage directionnel. . . . .	168
7.7	(a) Image sismique fracturée, (b) résultat du filtrage, (c) détail de l'image originale et (d) détail après filtrage. . . . .	169
7.8	Coupe issue : (a) d'un bloc de données sismiques, (b) des données filtrées sans utiliser l'attribut de désordre, (c) de la différence entre les deux filtrages et (d) en utilisant l'attribut. . . . .	170

7.9 (a) Image sismique originale et (b) image des valeurs de l'attribut de désordre $D_{att}$ .	171
7.10 Données sismiques 3D avant et après filtrage directionnel.	173
7.11 Représentation schématique (a) d'un onlap et (b) d'un toplap.	174
7.12 (a) Image originale, (b) image après filtrage directionnel et (c) squelettisation.	176
7.13 (a) Sélection d'un réflecteur, (b) réflecteur redressé (c) calcul de l'angle dominant au dessus de ce réflecteur.	177
7.14 Résultat de la recherche d'onlap/toplap.	178
7.15 Schéma illustrant le principe de la classification chronologique.	179
7.16 Classification chronologique (a) sans filtrage directionnel préliminaire et (b) après filtrage directionnel.	180
7.17 (a) Image sismique (sur support papier) numérisée et (b) image filtrée.	181
7.18 Une image de microscopie (MET).	182
7.19 (a) Échantillon d'image MET, (b)(c) résultats après filtrage directionnel (d) image de différence.	184
7.20 Représentation temps-fréquence avec (a) une STFT et (b) une décomposition par banc de filtres.	186
7.21 (a) Signal d'origine, (b) signal bruité, (c) signal reconstruit avec la STFT, (d) signal reconstruit avec un banc de filtres optimisé.	188
7.22 (a) Signal du gradient de pression, (b) signal de cliquetis, (c) signal reconstruit avec la STFT et (d) signal reconstruit avec un banc de filtres de synthèse optimisé.	189

---

# Liste des tableaux

3.1	Racines des trois déterminants mineurs maximaux de $\mathbf{H}[z]$ . . . . .	49
4.1	Temps de calcul nécessaire sous Matlab pour l'optimisation de différents bancs de filtres. . . . .	77
4.2	Dispersion fréquentielle des filtres de synthèse avant et après optimisation avec la fonction de coût $\tilde{\mathcal{J}}_f$ . . . . .	78
4.3	Tableau récapitulatif des différentes fonctions de coût utilisée dans ce travail.	86
6.1	Exemple de coefficients $\zeta$ utilisés pour l'estimation de $\sigma$ . . . . .	140
6.2	SNR des reconstructions avec FB-SURELET-C et avec l'oracle pour différent signaux perturbés par un bruit blanc gaussien ( $\sigma = 30$ ). . . . .	141
6.3	SNR des reconstructions avec FB-SURELET-E et avec l'oracle pour différent signaux perturbés par un bruit blanc gaussien ( $\sigma = 30$ ). . . . .	143
6.4	Résultats en PSNR de débruitage (pour un bruit $\sigma = 20$ ) avec FB-SURELET-E et FB-SURELET-C ainsi que leurs oracles respectifs sur l'image Lena. . . . .	145
6.5	Rapport signal sur bruit (PSNR) en dB après débruitage de l'image Lena par les méthodes FB-SURELET-E et FB-SURELET-C à l'aide d'un banc de filtre de synthèse avant et après optimisation. . . . .	145
6.6	Rapport signal sur bruit (PSNR) en dB après reconstruction par les deux méthodes proposées pour l'image Lena (pour un bruit $\sigma = 30$ ), avec $N = 8$ , $k = 3$ et $k'$ variable. . . . .	146
6.7	Temps de calcul et PSNR pour les méthodes FB-SURELET-C et FB-SURELET-E en utilisant différents groupes de sous-bandes. . . . .	147
6.8	Rapport signal sur bruit (PSNR) en dB après débruitage par les différentes méthodes sur les images Lena, Barbara et Boat. . . . .	150
6.9	Rapport signal sur bruit (PSNR) en dB après débruitage par les différentes méthodes sur les images Cameraman, Baboon et Fingerprint. . . . .	151
6.10	Rapport signal sur bruit (PSNR) en dB après débruitage des différentes images de test avec FB-SURELET-E (3,3,4) et $\alpha = 3$ pour un niveau de bruit $\sigma = 100$ . . . . .	156
7.1	Rapport signal sur bruit (SNR) en dB après reconstruction par les différentes méthodes. . . . .	187





---

# Bibliographie

- AHARONI, G., AVERBUCH, A., COIFMAN, R., ISRAELI, M. (1993). Local cosine transform – a method for the reduction of the blocking effect in JPEG. *Journal of Mathematical Imaging and Vision*, 3(1):7–38. 186
- ALLEN, J. (1977). Short-term spectral analysis, synthesis, and modification by discrete Fourier transform. *IEEE Trans. on Acous., Speech and Signal Proc.*, 25(3):235–238. 39, 185
- BAMBERGER, R. H., SMITH, M. J. T. (1992). A filter bank for the directional decomposition of images : theory and design. *IEEE Trans. on Signal Proc.*, 40(4):882–893. 41, 43
- BARRAULT, M., LE BRIS, C. (1999). Optimisation numérique et différentiation automatique pour un problème industriel. Rapport technique, Rapports de Recherche du CERMICS, Champs-sur-Marne. 72
- BASTIAANS, M. J. (1982). Gabor's signal expansion and degrees of freedom of a signal. *Optica Acta*, 29(9):1223–1229. 39
- BEAUCHAMP, K. G. (1975). *Walsh functions and their applications*. Academic Press, London, UK. 23, 25
- BELLANGER, M., BONNEROT, G., COUDREUSE, M. (1976). Digital filtering by polyphase network: application to sample-rate alteration and filter banks. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 24, pages 109–114. 31
- BENAZZA-BENYAHIA, A., PESQUET, J.-C. (2005). Building robust wavelet estimators for multicomponent images using Stein's principle. *IEEE Trans. on Image Proc.*, 14(11): 1814–1830. 118
- BERKOWITZ, S. J. (1984). On computing the determinant in small parallel time using a small number of processors. *Inf. Process. Lett.*, 18(3):147–150. 99, 193
- BERNARDINI, R. (2007). Time-variant unit-norm tight frames by means of modulated filter banks. *IEEE Trans. on Signal Proc.*, 55(8):4320–4324. 39
- BERNARDINI, R., DURIGON, M., RINALDO, R., VITALI, A., ZONTONE, P. (2006). Frame-based multiple-description video coding with extended orthogonal filter banks. *EUR-ASIP J. Appl. Sign. Proc.*, 2006:17. 39
- BLU, T. (1996). *Bancs de filtres itérés en fraction d'octave. Application au codage de son*. Thèse de doctorat, École nationale supérieure des télécommunications. 193
- BLU, T., LUISIER, F. (2007). The SURE-LET approach to image denoising. *IEEE Trans. on Image Proc.*, 16(11):2778–2786. 123, 124, 128, 133, 148

- BOCK, M. E. (1975). Minimax estimators of the mean of a multivariate distribution. *Annals of Statistics*, 3(1):209–218. 117
- BÖLCSKEI, H., HLAWATSCH, F. (1997). Oversampled filter banks : Optimal noise shaping, design freedom, and noise analysis. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 3, pages 2453–2456, Munich, Germany. 39
- BÖLCSKEI, H., HLAWATSCH, F. (1998). Oversampled cosine modulated filter banks with perfect reconstruction. *IEEE Trans. on CAS II*, 45(8):1057–1071. 38
- BÖLCSKEI, H., HLAWATSCH, F., FEICHTINGER, H. G. (1998). Frame-theoretic analysis of oversampled filter banks. *IEEE Trans. on Signal Proc.*, 46(12):3256–3268. 38, 40
- BRÉMAUD, P. (2002). *Mathematical principles of signal processing : Fourier and wavelet analysis*. Springer-Verlag, New York, USA. 23
- BRISLAWN, C. (1996). Classification of nonexpansive symmetric extension transforms for multirate filter banks. *Applied and Computational Harmonic Analysis*, 3(4):337–357. 128
- BUCHBERGER, B. (1965). *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. Thèse de doctorat, Mathematical Institute, University of Innsbruck, Austria,. 101
- BURT, P. J., ADELSON, E. H. (1983). The Laplacian pyramid as a compact image code. *IEEE Trans. on Commun.*, 31(4):532–540. 43
- CACAS, M. (2005). Méthode d'interprétation sédimentologique par estimation de différents scénarios chronologiques de mise en place des couches sédimentaires. Brevet français, numéro nat. d'enregistrement : 05/04 464. 174, 175
- CANDÈS, E., DONOHO, D. (1999a). Curvelets - a surprisingly effective nonadaptive representation for objects with edges. *In Curves and Surfaces*. L. L. Schumaker et al. (eds), Vanderbilt University Press, Nashville, TN. 42
- CANDÈS, E., DONOHO, D. L. (1999b). Ridgelets : a key to higher-dimensional intermittency ? *Phil. Trans. R. Soc. Lond. A*, 357:2495–2509. 43
- CANDÈS, E. J., DEMANET, L., DONOHO, D. L., YING, L. (2005). Fast discrete curvelet transforms. Rapport technique, Stanford University. 43
- CARRIER, F., KROOK, M., PEARSON, C. (1983). *Functions of a Complex Variable: Theory and Technique*. Hod Books. 113
- CASAZZA, P. G. (2000). The art of frame theory. *Taiwanese J. of Math.*, 15(4):129–201. 38
- CASSEREAU, P. M. (1985). *A new class of optimal unitary transforms for image processing*. Thèse de doctorat, Massachusetts Institute of Technology. 28
- CHAI, L., ZHANG, J., ZHANG, C., MOSCA, E. (2007). Frame-theory-based analysis and design of oversampled filter banks : Direct computational method. *IEEE Trans. on Signal Proc.*, 55(2):507–519. 40
- CHAUX, C. (2006). *Analyse en ondelettes M-bandes en arbre dual ; application à la restauration d'images*. Thèse de doctorat, Université de Marne-la-Vallée, France. 16, 37, 44

- CHAUX, C., DUVAL, L., BENAZZA-BENYAHIA, A., PESQUET, J.-C. (2008). A nonlinear Stein based estimator for multichannel image denoising. *IEEE Trans. on Signal Proc.* To be published. 118, 120
- COHEN, A., DAUBECHIES, I., FEAUVEAU, J.-C. (1992). Biorthogonal bases of compactly supported wavelets. *Comm. ACM*, 45:485–560. 35
- COIFMAN, R., DONOHO, D. (1995). Translation-invariant de-noising. In ANTONIADIS, A., OPPENHEIM, G., éditeurs : *Wavelets and Statistics*, vol. 103 *Lecture Notes in Statistics*, pages 125–150. Springer, New York, NY, USA. 37
- COLEMAN, T. F., LI, Y. (1994). On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. *Mathematical programming*, 67(2): 189–224. 77
- COLLINS, G. (1971). The calculation of multivariate polynomial resultants. *Journal of the ACM*, 18(4):515–532. 93, 95
- COMBETTES, P., PESQUET, J.-C. (2004). Constraint construction in convex set theoretic signal recovery via Stein's principle. In *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 2, pages 813–816, Montréal, Canada. 118
- COX, D., LITTLE, J., O'SHEA, D. (1998). *Using Algebraic Geometry*. Springer-Verlag New York. 101, 102
- CROCHIERE, R. E. (1980). A weighted overlap-add method of short-time Fourier analysis/synthesis. *IEEE Trans. on Acous., Speech and Signal Proc.*, 28(1):99–102. 39, 185
- CROISIER, A., ESTEBAN, D., C., G. (1976). Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques. In *Inter. Conf. on Inform. Science and Systems*, Patras, Greece. 30
- ŞENDUR, L., SELESNICK, I. W. (2002). Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Trans. on Signal Proc.*, 50(11): 2744–2756. 147
- CVETKOVIĆ, Z. (2000). On discrete short-time Fourier analysis. *IEEE Trans. on Signal Proc.*, 48(9):2628–2640. 39
- CVETKOVIĆ, Z., VETTERLI, M. (1998a). Overampled filter banks. *IEEE Trans. on Signal Proc.*, 46(5):1245–1255. 38
- CVETKOVIĆ, Z., VETTERLI, M. (1998b). Tight Weyl-Heisenberg frames in  $\ell^2(\mathbb{R})$ . *IEEE Trans. on Signal Proc.*, 46(5):1256–1259. 39
- DAUBECHIES, I. (1992). *Ten Lectures on Wavelets*. CBMS-NSF, SIAM Lecture Series, Philadelphia, PA, USA. 23, 34, 38, 185
- DAUBECHIES, I., GROSSMANN, A., MEYER, Y. (1986). Painless nonorthogonal expansions. *J. Math. Phys.*, 27:1271–1283. 37
- DAUBECHIES, I., HAN, B. (2002). The canonical dual frame of a wavelet frame. *Appl. and Comp. Harm. Analysis*, 12(3):269–285. 38
- de QUEIROZ, R., NGUYEN, T., RAO, K. (1996). The GenLOT: generalized linear-phase lapped orthogonal transform. *IEEE Trans. on Signal Proc.*, 44(3):497–507. 55

- de QUEIROZ, R. L., RAO, K. R. (1993). Adaptive extended lapped transforms. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 3, pages 217–220, Minneapolis, MN, USA. 194
- de QUEIROZ, R. L., TRAN, T. D. (2001). Lapped transforms for image compression. *In* RAO, K. R., YIP, P., éditeurs : *The Handbook on Transforms and Data Compression*. CRC Press. 16
- DO, M., VETTERLI, M. (2005). The contourlet transform : an efficient directional multi-resolution image representation. *IEEE Trans. on Image Proc.*, 14(12):2091–2106. 43
- DO, M. N. (2001). *Directional Multiresolution Image Representations*. Thèse de doctorat, Swiss Federal Institute of Technology Lausanne. 43
- DONOHOO, D. L., DUNCAN, M. R. (1999). Digital curvelet transform : Strategy, implementation, experiments. Rapport technique, Stanford University. 42
- DONOHOO, D. L., JOHNSTONE, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *J. American Statist. Ass.*, 90:1200–1224. 118, 139, 147
- DUFFIN, R., SCHAEFFER, A. (1952). A class of non-harmonic Fourier series. *Trans. Amer. Math. Soc.*, 72:341–366. 37
- DUMITRESCU, B., BREGOVIĆ, R., SARAMÄKI, T. (2006). Simplified design of low-delay oversampled NPR GDFT filterbanks. *EURASIP J. Appl. Sign. Proc.*, 2006:11. 39, 41
- DURAK, L., ARIKAN, O. (2003). Short-time Fourier transform : two fundamental properties and an optimal implementation. *IEEE Trans. on Signal Proc.*, 51(5):1231–1242. 39
- DUVAL, L. (2000). *Compression de données sismiques : bancs de filtres et transformées étendues, synthèse et adaptation*. Thèse de doctorat, Université Paris-Sud, Orsay. 16
- FAUGÈRE, J. (2002). A new efficient algorithm for computing Gröbner bases without reduction to zero ( $f_5$ ). *In* MORA, T., éditeur : *Proc. of International Symposium on Symbolic and Algebraic Computation*, pages 75–83. ACM Press. 102
- FAUGÈRE, J.-C., MOREAU de SAINT-MARTIN, F., , ROUILLIER, F. (1998). Design of regular nonseparable bidimensional wavelets using Gröbner basis techniques. *IEEE Trans. on Signal Proc.*, 46(4):845–856. 101
- FEICHTINGER, H.-G., KOZEK, W., STROHMER, T. (1995). Reconstruction of signals from irregular samples of its short-time Fourier transform. *In* LAINE, A. F., UNSER, M. A., WICKERHAUSER, M. V., éditeurs : *Proc. SPIE, Wavelets Applications in Signal and Image Processing*, vol. 2569, pages 140–150, San Diego, CA, USA. 39
- FLANDRIN, P. (1998). *Time-frequency and time-scale analysis*. Academic Press, San Diego, USA. 23, 34, 69, 185
- FORNASINI, E., VALCHER, M. E. (1997). nD polynomial matrices with applications to multidimensional signal analysis. *Multidimensional Systems and Signal Processing*, 8:387–408. 48, 91
- FOWLER, J. E. (2005). The redundant discrete wavelet transform and additive noise. *Signal Processing Letters*, 12(9):629–632. 37

- FREEMAN, W. T., ADELSON, E. H. (1991). The design and use of steerable filters. *IEEE Trans. on Pattern Anal. and Match. Int.*, 13(9):891–906. 44
- FULTON, W. (1969). *Algebraic Curves*. Addison Wesley. 92, 95, 98
- GAN, L., MA, K.-K. (2003). Oversampled linear-phase perfect reconstruction filterbanks : Theory, lattice structure and parameterization. *IEEE Trans. on Signal Proc.*, 51(3): 744–759. 40
- GANDUBERT, A. D., LEGENS, C., GUILLAUME, D., REBOURS, D. S., PAYEN, E. (2007). X-ray photoelectron spectroscopy surface quantification of sulfided CoMoP catalysts relation between activity and promoted sites part i : Influence of the Co/Mo ratio. *Oil & Gas Science and Technology - Revue de l'IFP*, 62(1):79–89. 182
- GAUTHIER, J., DUVAL, L., CACAS, M. (2006). Méthode d'interprétation stratigraphique d'images sismiques. Brevet français, numéro nat. d'enregistrement : 06/10 493. 172
- GAUTHIER, J., DUVAL, L., PESQUET, J. C. (2005). A non separable 2D complex modulated lapped transform and its applications to seismic data filtering. *In Proc. Eur. Sig. and Image Proc. Conference*. Antalya, Turkey. 110
- GOPINATH, R. A. (2003). The phaselet transform — an integral redundancy nearly shift-invariant wavelet transform. *IEEE Trans. on Signal Proc.*, 51(7):1792–1805. 44
- GOYAL, V. K., KOVACEVIC, J., KELNER, J. A. (2001). Quantized frame expansions with erasures. *Appl. and Comp. Harm. Analysis*, 10(3):203–233. 39
- GRÖCHENIG, K., ZIMMERMAN, G. (2001). Hardy's theorem and the short-time Fourier transform of Schwartz functions. *J. London Math. Soc.*, 2(63):205–214. 39
- HARTENECK, M., WEISS, S., STEWART, R. (1999). Design of near perfect reconstruction oversampled filter banks for subband adaptive filters. *IEEE Trans. on CAS II*, 46(8): 1081–1086. 41
- HASSIBI, B., HOCHWALD, B., SHOKROLLAHI, A., SWELDENS, W. (2001). Representation theory for high-rate multiple-antenna code design. *IEEE Trans. on Inform. Theory*, 47(6):2355–2367. 39
- HEIL, C., WALNUT, D. F. (2006). *Fundamental papers in wavelet theory*. Princeton University Press. 34
- HERMANN, D., CHAU, E., DONY, R. D., AREIBI, S. M. (2007). Window based prototype filter design for highly oversampled filter banks in audio applications. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 2, pages 405–408, Hawaii, HI, USA. 41
- IHALAINEN, T., STITZ, T. H., RINNE, M., RENFORS, M. (2007). Channel equalization in filter bank based multicarrier modulation for wireless communications. *EURASIP J. Appl. Sign. Proc.*, 2007:18. 39
- JAGGI, S., EFFROS, M., HO, T. C., MEDARD, M. (2004). On linear network coding. *In Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*. 194
- JAMES, W., STEIN, C. (1961). Estimation with quadratic loss. *Proc. Fourth Berkeley Symp. Math. Statist. Prob.*, 1:311–319. 117

- JOHANSSON, H., LÖWENBORG, P. (2007). Flexible frequency-band reallocation networks using variable oversampled complex-modulated filter banks. *EURASIP J. Appl. Sign. Proc.*, 2007:15. 39
- JÖNSSON, G., VAVASIS, S. (2005). Accurate solution of polynomial equations using Macaulay resultant matrices. *Mathematics of Computation*, (74):221–262. 96
- KAIBLINGER, N. (2005). Approximation of the Fourier transform and the dual Gabor window. *J. Fourier Analysis and Appl.*, 11(1):25–42. 39
- KALKER, T., PARK, H., VETTERLI, M. (1995). Groebner basis techniques in multidimensional multirate systems. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 4, pages 2121–2124, Detroit, MI, USA. 40
- KELLERMANN, W. (1988). Analysis and design of multirate systems for cancellation of acoustical echoes. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, New York, NY, USA. 38, 39
- KINGSBURY, N. G. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Appl. and Comp. Harm. Analysis*, 10(3):234–253. 44
- KOVAČEVIĆ, J., VETTERLI, M. (1992). Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for  $\mathbb{R}^n$ . *IEEE Trans. on Inform. Theory*, 38(2):533–555. 41
- KOVAČEVIĆ, J., VETTERLI, M. (1995). Nonseparable two- and three-dimensional wavelets. *IEEE Trans. on Signal Proc.*, 43(5):1269–1273. 88
- LABEAU, F., CHIANG, J.-C., KIEFFER, M., DUHAMEL, P., VANDENDORPE, L., MACQ, B. (2005). Oversampled filter banks as error correcting codes : Theory and impulse noise correction. *IEEE Trans. on Signal Proc.*, 53(12):4619–4630. 39
- LAVERGNE, M. (1986). *Méthodes sismiques*. Technip. 164
- LE PENNEC, E. (2002). *Bandelettes et représentation géométrique des images*. Thèse de doctorat, École Polytechnique, France. 42
- LIGHTSTONE, M., MAJANI, E., MITRA, S. K. (1997). Low bit-rate design considerations for wavelet-based image coding. *Multidimensional Systems and Signal Processing*, 8(1-2):111–128. 146, 194
- LIN, J., SMITH, M. (2008). New perspectives and improvements on the symmetric extension filter bank for subband/wavelet image compression. *IEEE Trans. on Signal Proc.*, 17(2):177–189. 128
- LUISIER, F., BLU, T. (2007). SURE-LET multichannel image denoising : interscale orthonormal wavelet thresholding. *IEEE Trans. on Image Proc.* 118, 147
- MACAULAY, F. (1902). On some formulæ in elimination. *In Proc. London Math. Soc.*, vol. 35, pages 3–27. 96
- MAHAJAN, M., VINAY, V. (1997). A combinatorial algorithm for the determinant. *In Proc. of the 8th annual ACM-SIAM Symp. On Discrete Algorithms (SODA)*, pages 730–738. Society for Industrial and Applied Mathematics. 99, 193
- MALLAT, S. (1998). *A wavelet tour of signal processing*. Academic Press, San Diego, CA, USA. 23, 34

- MALLAT, S. (2008). Geometrical grouplets. *Appl. and Comp. Harm. Analysis*. To appear. 44
- MALVAR, H. S. (1992). *Signal processing with Lapped Transforms*. Artech House, Norward, MA, USA. 16, 23, 29, 186
- MALVAR, H. S. (1999). A modulated complex lapped transform and its applications to audio processing. In *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 3, pages 1421–1424, Phoenix, AZ, USA. 29, 39, 56
- MALVAR, H. S., STAELIN, D. H. (1989). The LOT : Transform coding without blocking effects. *IEEE Trans. on Acous., Speech and Signal Proc.*, 37(4):553–559. 28
- MANSOUR, M. F. (2007). On the optimization of oversampled DFT filter banks. *Signal Processing Letters*, 14(6):389–392. 39, 57
- MARINKOVIC, S., GUILLEMOT, C. (2008). Joint source-channel coding by means of an oversampled filter bank code. *EURASIP J. Appl. Sign. Proc.*, 2008(1):1–12. 194
- MARS, J., LACOUME, J. L., MARI, J. L., GLANGEAUD, F. (2004). *Traitement du signal pour géologues et géophysiciens, Vol 3 : Techniques avancées*. Editions technip. 164
- MARTIN, N., MARS, J., MARTIN, J., CHORIER, C. (1995). A Capon's time-octave representation application in room acoustics. *IEEE Trans. on Signal Proc.*, 43(8):1842–1854. 194
- MEYER, Y. (1992). *Ondelettes et algorithmes concurrents*. Paris, France. 34
- MOREAU de SAINT-MARTIN, F., SIOHAN, P., COHEN, A. (1999). Biorthogonal filterbanks and energy preservation property in image compression. *IEEE Trans. on Image Proc.*, 8(2):168–178. 146, 194
- NASON, G. P., SILVERMAN, B. W. (1995). *The stationary wavelet transform and some statistical applications*, vol. 103 *Lecture Notes in Statistics*, pages 281–300. Springer Verlag, New York, NY, USA. 37
- NGUYEN, L.-T., BELOUCHRANI, A., ABED-MERAIM, K., BOASHASH, B. (2005). Separating more sources than sensors using time-frequency distributions. *EURASIP J. Appl. Sign. Proc.*, 2005(17):2828–2847. 183, 194
- NOCEDAL, J., WRIGHT, S. J. (1999). *Numerical optimization*. Springer Verlag, New York. 72
- NUTTALL, A. H. (1981). Some windows with very good sidelobe behavior. *IEEE Trans. on Acous., Speech and Signal Proc.*, 29(1):84–91. 185
- PARK, H., KALKER, T., VETTERLI, M. (1997). Gröbner bases and multidimensional FIR multirate systems. *Multidimensional Systems and Signal Processing*, 8:11–30. 40, 48, 101
- PENROSE, R. (1955). A generalized inverse for matrices. *Proc. Camb. Philos. Soc.*, 51:406–413. 50
- PESQUET, J.-C., KRIM, H., CARFANTAN, H. (1996). Time-invariant orthogonal wavelet representations. *IEEE Trans. on Signal Proc.*, 44(8):1964–1970. 37
- PESQUET, J.-C., LEPORINI, D. (1997). A new wavelet estimator for image denoising. In *IEE Sixth International Conference on Image Processing and its Applications*, vol. 1, pages 249–253. 118, 148

- PETRISOR, T., TILLIER, C., PESQUET-POPESCU, B., PESQUET, J.-C. (2005). Comparison of redundant wavelet schemes for multiple description coding of video sequences. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 5. 194
- PEYRÉ, G. (2004). *L'algèbre discrète de la transformée de Fourier*. Ellipses. 25
- PEYRÉ, G. (2005). *Géométrie multi-échelles pour les images et les textures*. Thèse de doctorat, École Polytechnique. 42
- PORTNOFF, M. R. (1978). *Time scale modification of speech based on short-time Fourier analysis*. Thèse de doctorat, Mass. Inst. Tech., Cambridge, MA, USA. To be checked (title...). 39
- PORTNOFF, M. R. (1980). Time-frequency representations of digital signals and systems based on short time Fourier analysis. *IEEE Trans. on Acous., Speech and Signal Proc.*, 28:55–69. 39
- PORTNOFF, M. R. (1981). Time scale modification of speech based on short-time Fourier analysis. *IEEE Trans. on Acous., Speech and Signal Proc.*, 29(3):374–390. 185
- PRESS, W., TEUKOLSKY, S., VETTERLING, W., FLANNERY, B. (2007). *Numerical recipes : The Art of Scientific Computing*. Cambridge University Press, 3rd édition. 77
- PUN, K. S. C., NGUYEN, T. Q. (2007). Design of oversampled filter bank based widely linear OQPSK equalizer. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 3, pages 861–864, Honolulu, HI, USA. 39
- RAO, K. R. (1985). *Discrete Transforms and Their Applications*. Krieger Pub Co. 23
- RAPHAN, M., SIMONCELLI, E. P. (2006). Learning to be Bayesian without supervision. *In Proc. Neural Information Processing Systems*, Vancouver, BC, Canada. Published as Advances in Neural Information Processing Systems, eds. B. Schölkopf, J. Platt and T. Hofmann, vol. 19, May 2007. 118
- RICHARDS, J. A. (1999). An introduction to James-Stein estimation. Preprint. 118
- RIEL, B., SHPAK, D. J., ANTONIOU, A. (2004). Lifting-based design and implementation of oversampled paraunitary modulated filter banks. *In Proc. Mid. Symp. Circ. and Syst.*, vol. 2, pages 97–100, Hiroshima, Japan. 40
- RIOUL, O., DUHAMEL, P. (1992). Fast algorithms for discrete and continuous wavelet transforms. *IEEE Trans. on Inform. Theory*, 38(2):569–586. 34
- RON, A., SHEN, Z. (1998). *Advances in wavelets*, chap. Construction of compactly supported affine frames in  $L_2(\mathbb{R}^d)$ , pages 27–49. Springer-Verlag. 38
- ROSENBAUM, L. (2007). *On low-complexity frequency selective digital filters and filter banks*. Thèse de doctorat, Linköping University. 41
- ROUEFF, A., CHANUSSOT, J., MARS, J. I., NGUYEN, M. Q. (2004). Unsupervised separation of seismic waves using the watershed algorithm on time-scale images. *Geophysical Prospecting*, 52(4):287–300. 164
- ROUSSEEUW, P. J., CROUX, C. (1993). Alternatives to the median absolute deviation. *J. American Statist. Ass.*, 88:1273–1283. 138
- SELESNICK, I. W. (2001). The double-density dual-tree discrete wavelet transform. *IEEE Trans. on Signal Proc.* 38



- SELESNICK, I. W. (2006). A higher density discrete wavelet transform. *IEEE Trans. on Signal Proc.*, 54(8):3039–3048. 38
- SELESNICK, I. W., BARANIUK, R. G., KINGSBURY, N. G. (2005). The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151. 37
- SHAPIRO, J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Proc.*, 41:3445–3462. 34
- SICLET, C., SIOHAN, P., PINCHON, D. (2006). Perfect reconstruction conditions and design of oversampled DFT-modulated transmultiplexers. *EURASIP J. Appl. Sign. Proc.*, 2006:14. 40
- SIOHAN, P., ROCHE, C. (2000). Cosine-modulated filterbanks based on extended gaussian functions. *IEEE Trans. on Signal Proc.*, 48(11):3052–3061. 41
- SMITH, M., BARNWELL, T. (1984). A procedure for designing exact reconstruction filter banks for tree structured subband coders. In *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 9, pages 421–424. 30
- SØNDERGAARD, P. L. (2007). *Finite Discrete Gabor Analysis*. Thèse de doctorat, Technical University of Denmark, Department of Mathematics. 39
- STEFFEN, P., HELLER, P. N., GOPINATH, R. A., BURRUS, C. S. (1993). Theory of regular  $M$ -band wavelet bases. *IEEE Trans. on Signal Proc.*, 41(12):3497–3511. 35
- STEIN, C. (1981). Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9(6):1135–1151. 118, 119
- STRANG, G. (1998). *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2nd édition. 65
- STRANG, G., NGUYEN, T. (1996). *Wavelets and filter banks*. Wellesley-Cambridge Press. 23, 31
- TANAKA, T. (2006). A direct design of oversampled perfect reconstruction FIR filter banks of 50%-overlapping filters. *IEEE Trans. on Signal Proc.*, 54(8):3011–3022. 40
- TRAN, T., LIANG, J., TU, C. (2003). Lapped transform via time-domain pre- and post-filtering. *IEEE Trans. on Signal Proc.*, 51(6):1557–1571. 172
- TRAN, T. D., de QUEIROZ, R. L., NGUYEN, T. Q. (2000). Linear phase perfect reconstruction filter bank : lattice structure, design, and application in image coding. *IEEE Trans. on Signal Proc.*, 48:133–147. 194
- UENG, N.-T., SCHARF, L. L. (1996). Frames and orthonormal bases for variable windowed Fourier transforms. In *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 5, pages 2594–2597, Atlanta, GA, USA. 39
- VAIDYANATHAN, P. P. (1987). Theory and design of  $m$ -channel maximally decimated quadrature mirror filters with arbitrary  $m$ , having the perfect-reconstruction property. *IEEE Trans. on Acous., Speech and Signal Proc.*, 35(4):476–492. 31
- VAIDYANATHAN, P. P. (1993). *Multirate systems and filter banks*. Prentice Hall. 23, 41, 47, 90
- von BORRIES, R., BURRUS, C. S. (2004). Linear phase oversampled filter banks. In *Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 2, pages 961–964, Montreal, Canada. 40

- von BORRIES, R., de QUEIROZ, R., BURRUS, C. S. (2001). On filter banks with rational oversampling. *In Proc. Int. Conf. on Acoust., Speech and Sig. Proc.*, vol. 6, pages 3657–3660, Salt Lake City, UT, USA. 40
- WEISS, S., REDIF, S., COOPER, T., LIU, C., BAXTER, P. D., McWHIRTER, J. G. (2006). Paraunitary oversampled filter bank design for channel coding. *EURASIP J. Appl. Sign. Proc.*, 2006:10. 39
- WILBUR, M. R., DAVIDSON, T., REILLY, J. P. (2003). Efficient design of oversampled NPR GDFT filter banks. Rapport technique, McMaster University. 41
- XIA, X.-G. (1996). On characterization of the optimal biorthogonal window functions for Gabor transforms. *IEEE Trans. on Signal Proc.*, 44(1):133–136. 39
- XIA, X. G., SUTER, B. W. (1995). A family of two-dimensional nonseparable Malvar wavelets. *Appl. and Comp. Harm. Analysis*, 2:243–256. 87
- YILMAZ, Ö., DOHERTY, S. (2001). *Seismic data analysis : processing, inversion, and interpretation of seismic data*. Society of Exploration Geophysicists. 164
- YOUNG, R. W., KINGSBURY, N. G. (1993). Frequency-domain motion estimation using a complex lapped transform. *IEEE Trans. on Image Proc.*, 2(1):2–17. 29, 56
- YUI, K. F. C., GRBIĆ, N., NORDHOLM, S., TEO, K. L. (2004). Multicriteria design of oversampled uniform DFT filter banks. *Signal Processing Letters*, 11(6):541–544. 39
- ZHOU, J., DO, M. N. (2005). Multidimensional oversampled filter banks. *In PAPADAKIS, M., LAINE, A. F., UNSER, M. A., éditeurs : Proc. SPIE, Wavelets Applications in Signal and Image Processing*, vol. 5914, pages 591424.1–591424.12, San Diego, CA, USA. 40, 101, 102