



**HAL**  
open science

# Décompositions conjointes de matrices complexes : application à la séparation de sources

Tual Trainini

► **To cite this version:**

Tual Trainini. Décompositions conjointes de matrices complexes : application à la séparation de sources. Autre. Université de Toulon, 2012. Français. NNT : 2012TOUL0010 . tel-00798019

**HAL Id: tel-00798019**

**<https://theses.hal.science/tel-00798019>**

Submitted on 7 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ECOLE DOCTORALE SCIENCES FONDAMENTALES APPLIQUEES

Laboratoire des Sciences de l'Information et des Systèmes (LSIS)  
Equipe Signal & Images (SIIM)

THESE présentée par :  
**Tual Trainini**

pour obtenir le grade de Docteur en Sciences pour l'Ingénieur

**DECOMPOSITION CONJOINTE DE MATRICES COMPLEXES  
APPLICATION A LA SEPARATION DE SOURCES**

Thèse dirigée par :

**M. MOREAU Eric**      Directeur de thèse  
**M. LUCE Hubert**      Co-directeur de thèse

Jury :

**M. BOURENNANE Salah**  
**M. FAVIER Gérard**  
**M. MARS Jérôme**  
**M. WEBER Rodolphe**



## REMERCIEMENTS

“ Dieu a créé l'homme, et ensuite pour le remercier  
l'homme a créé Dieu.”  
Le tour du chat en 365 jours, P. Geluck

En tout premier lieu, je tiens à remercier M. Eric Moreau qui, en tant que directeur de thèse, s'est toujours montré disponible et de bon conseil, tout en supportant avec calme et maîtrise mes mécontentements réguliers !

Merci à H. Luce d'avoir co-encadré cette thèse, et qui m'aura fait toucher du doigt les difficultés inhérentes aux données réelles.

MM. J. Mars et G. Favier ont accepté d'être les rapporteurs de cette thèse, et je les en remercie. Leurs remarques pertinentes ont permis d'éclaircir certains points de ce manuscrit.

Merci également à M. S. Bourennane d'avoir accepté de faire partie du jury.

Je souhaite également remercier R. Weber, avec qui j'ai pu travailler sur une application concrète et très intéressante pour les méthodes développées. Je garde un excellent souvenir de mes semaines passées à Nançay, et je pense arriver à apprécier désormais la Sologne, même en automne ! J'espère que cette collaboration perdurera, et que cela permettra à d'autres de découvrir un domaine fort intéressant. Merci également à Gregory pour tout ce qu'il a pu faire sur les simulations, et aux responsables de l'institut néerlandais de radioastronomie ASTRON pour m'avoir autorisé l'utilisation des données.

Merci aux membres de l'aile Télécom d'avoir toujours eu une attitude sympathique à mon égard, et de m'avoir réservé un accueil chaleureux. Certaines discussions m'ont permis d'avoir plus de recul sur les algorithmes développés, des conseils judicieux sur l'enseignement etc.

Merci à William Webb Ellis d'avoir inventé un sport afin de m'oxygéner le week-end. Merci aux copains du sifflet Fred, Hervé, Franck, Rice et tous les autres, Papa Coach pour son soutien sans faille et qui m'aura amené de jeune Padawan à Jedi, sans oublier Cha dont les élucubrations trouveront toujours grâce à mes yeux. Un merci particulier à tous les supporters aux quatre coins de la France, qui auront su, par leurs mots fleuris, me donner une surmotivation pour aller travailler le lundi matin !

Merci aux copains et en particulier, pêle mêle, FloFlo, Captain, la famille "M", Seb "le chef" et Fanny, Pti Flo, Vincent, Geooooorges la loutre, Moman, PJ, Polgas, Gnognotte, Touillette et Bidule.

Je ne saurai trouver les mots pour exprimer ce que je ressens vis-à-vis de ma famille, de mes parents, de mon frère, de ma soeur et de ma compagne, eux qui auront réussi pendant toutes ses années à me supporter au quotidien (et ils y arrivent encore, ce n'est pas une mince affaire), et qui je l'espère seront toujours à mes côtés. J'espère que vous serez toujours fiers de ce que j'entreprends.

# Table des matières

<b>Acronymes</b>	<b>x</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Cadre de la thèse . . . . .	3
1.2 Objectifs et contenu du document . . . . .	5
1.3 Notations, définitions et outils . . . . .	6
1.3.1 Notations classiques . . . . .	6
1.3.2 Opération sur les matrices . . . . .	7
<b>2 Présentation du problème</b>	<b>10</b>
2.1 Diagonalisation conjointe . . . . .	10
2.1.1 Décompositions matricielles . . . . .	10
2.1.2 Diagonalisation conjointe approchée . . . . .	12
2.1.3 Au sujet de l'identifiabilité . . . . .	13
2.2 Lien avec la séparation de sources . . . . .	13
2.3 Description de quelques algorithmes . . . . .	17
<b>3 Algorithmes de Gradient</b>	<b>20</b>
3.1 Description générale . . . . .	20
3.2 Calcul du gradient . . . . .	22
3.2.1 Cas hermitien . . . . .	22
3.2.2 Cas symétrique . . . . .	23
3.2.3 Pas optimal . . . . .	23
3.3 Gradient et approximations du critère . . . . .	23
3.3.1 Approximations sur le critère . . . . .	24
3.3.2 Cas hermitien . . . . .	25
3.3.3 Cas symétrique . . . . .	26
3.3.4 Pas optimal . . . . .	27

3.4	Description des algorithmes . . . . .	27
3.5	Simulations . . . . .	28
3.5.1	Cas hermitien . . . . .	29
3.5.2	Cas symétrique . . . . .	29
3.5.3	Combinaison des deux ensembles . . . . .	31
3.5.4	Complexité de calcul . . . . .	36
3.6	Conclusions sur les méthodes de type gradient . . . . .	36
<b>4</b>	<b>Algorithmes de mise à jour analytique de la matrice</b>	<b>38</b>
4.1	Algorithme de mise à jour par couple de termes . . . . .	38
4.1.1	Description générale . . . . .	38
4.1.2	Calcul de la matrice de mise à jour $\mathbf{Z}$ . . . . .	39
4.1.2.1	Cas hermitien . . . . .	39
4.1.2.2	Cas symétrique . . . . .	41
4.1.2.3	Combinaison des deux ensembles . . . . .	43
4.1.3	Description des algorithmes . . . . .	44
4.2	Algorithmes de moindres carrés alternés . . . . .	44
4.2.1	Description générale . . . . .	44
4.2.2	Mise à jour des matrices diagonales $\mathbf{D}_i$ . . . . .	45
4.2.3	Mise à jour de la matrice de mélange $\mathbf{A}$ . . . . .	45
4.2.4	Mise à jour de $\mathbf{C}$ . . . . .	45
4.2.5	Recherche linéaire optimisée . . . . .	46
4.2.6	Description des algorithmes . . . . .	47
4.2.7	Remarque . . . . .	48
4.3	Simulations . . . . .	48
4.3.1	Cas hermitien . . . . .	49
4.3.2	Cas symétrique . . . . .	49
4.3.3	Combinaison des deux ensembles . . . . .	53
4.3.4	Remarque sur les temps de calculs . . . . .	53
4.4	Conclusions sur les méthodes de recherche analytiques . . . . .	55
<b>5</b>	<b>Application à la séparation de sources de télécommunications</b>	<b>57</b>
5.1	Séparation de sources . . . . .	57
5.2	Hypothèses de simulation . . . . .	58
5.3	Cas hermitien . . . . .	59
5.4	Cas symétrique . . . . .	62
5.5	Combinaison des deux ensembles . . . . .	64

5.6	Conclusions sur l'application aux sources de télécommunications numériques . . .	67
<b>6</b>	<b>Application à la radioastronomie</b>	<b>68</b>
6.1	Présentation . . . . .	68
6.2	Données synthétiques . . . . .	70
6.2.1	Modèle . . . . .	70
6.2.2	Simulations informatiques . . . . .	71
6.2.2.1	Comparaison des vitesses de convergence . . . . .	73
6.2.2.2	Influence de la puissance des interférences . . . . .	75
6.2.2.3	Influence du nombre d'échantillons . . . . .	75
6.2.2.4	Influence du nombre de pollueurs . . . . .	77
6.2.2.5	Autres résultats . . . . .	77
6.3	Données expérimentales . . . . .	79
6.3.1	Justification empirique de l'utilisation d'une matrice aléatoire comme point initial . . . . .	80
6.3.2	Justification empirique de l'utilisation des cumulants . . . . .	82
6.3.3	Traitements pour le scénario "land mobile" . . . . .	82
6.3.4	Traitements pour le scénario "pager" . . . . .	85
<b>7</b>	<b>Conclusions et Perspectives</b>	<b>88</b>
7.1	Conclusions . . . . .	88
7.2	Perspectives . . . . .	88
<b>A</b>	<b>Développement des calculs des algorithmes de gradient</b>	<b>90</b>
A.1	Calcul du gradient . . . . .	90
A.1.1	Cas hermitien . . . . .	90
A.1.2	Cas symétrique . . . . .	92
A.2	Calcul du gradient pour un critère simplifié . . . . .	93
A.2.1	Cas hermitien . . . . .	93
A.2.2	Cas symétrique . . . . .	93
<b>B</b>	<b>Développement des calculs pour les coefficients des pas optimaux</b>	<b>95</b>
B.1	Calcul du pas optimal pour le critère initial . . . . .	95
B.2	Calcul du pas optimal dans le cas d'un critère simplifié . . . . .	96
B.3	Calcul des coefficients pour l'ELS . . . . .	97

# Liste des Algorithmes

1	UWEDGE . . . . .	17
2	ACDC . . . . .	18
3	FAJD . . . . .	19
4	Schéma Global pour les algorithmes de gradient . . . . .	22
5	Pas Optimal . . . . .	23
6	H"Diagonalization of complEx SEts using a Relative gradienT" (DESERT) . . . . .	27
7	SDESERT . . . . .	27
8	"NON Orthogonal Decomposition of compLEx Sets" (NOODLES) . . . . .	44
9	"Enhanced Line Search" (ELS) . . . . .	47
10	"Alternating Least Square" (ALS) . . . . .	47
11	ALS <sub>ELS</sub> . . . . .	47

# Table des figures

2.1	Schéma de la séparation de sources . . . . .	14
2.2	Schéma dans le cas bruité . . . . .	14
2.3	Schéma dans le cas d'un blanchiment . . . . .	15
3.1	Vitesse de convergence des algorithmes HDESERT. . . . .	30
3.2	Vitesse de convergence des algorithmes HDESERT. . . . .	30
3.3	Vitesse de convergence des algorithmes HDESERT. Comparaison en fonction du critère. . . . .	31
3.4	Vitesse de convergence des algorithmes HDESERT. Comparaison en fonction du critère. . . . .	32
3.5	Vitesse de convergence des algorithmes SDESERT. . . . .	32
3.6	Vitesse de convergence des algorithmes SDESERT. . . . .	33
3.7	Vitesse de convergence des algorithmes DESERT. . . . .	33
3.8	Performance des algorithmes DESERT. . . . .	34
3.9	Vitesse de convergence des algorithmes DESERT. . . . .	34
3.10	Performance des algorithmes DESERT. . . . .	35
3.11	Vitesse de convergence des algorithmes DESERT. Comparaison en fonction du critère. . . . .	35
3.12	Vitesse de convergence des algorithmes DESERT. Comparaison en fonction du critère. . . . .	36
4.1	Performance des algorithmes ALS et NOODLES . . . . .	49
4.2	Vitesse de convergence des algorithmes ALS et NOODLES . . . . .	50
4.3	Performance des algorithmes ALS et NOODLES . . . . .	50
4.4	Vitesse de convergence des algorithmes ALS et NOODLES . . . . .	51
4.5	Performance des algorithmes ALS . . . . .	51
4.6	Performance des algorithmes ALS et NOODLES . . . . .	52
4.7	Vitesse de convergence des algorithmes ALS et NOODLES . . . . .	52
4.8	Performance des algorithmes ALS et NOODLES . . . . .	53

4.9	Vitesse de convergence des algorithmes ALS et NOODLES . . . . .	54
4.10	Performance des algorithmes ALS et NOODLES . . . . .	54
4.11	Vitesse de convergence des algorithmes ALS et NOODLES . . . . .	55
5.1	Vitesse de convergence des algorithmes . . . . .	60
5.2	Performance des algorithmes en fonction du Rapport Signal sur Bruit (RSB) . . .	60
5.3	Vitesse de convergence des algorithmes . . . . .	61
5.4	Performance des algorithmes en fonction du RSB . . . . .	61
5.5	Vitesse de convergence des algorithmes . . . . .	62
5.6	Performance des algorithmes en fonction du RSB . . . . .	63
5.7	Vitesse de convergence des algorithmes . . . . .	63
5.8	Performance des algorithmes en fonction du RSB . . . . .	64
5.9	Vitesse de convergence des algorithmes . . . . .	65
5.10	Performance des algorithmes en fonction du RSB . . . . .	65
5.11	Vitesse de convergence des algorithmes . . . . .	66
5.12	Performance des algorithmes en fonction du RSB . . . . .	66
6.1	Superterp, le coeur du coeur de "LOw Frequency ARray" (LOFAR) (6 stations "High Band Antennas" (HBA) et "Low Band Antennas" (LBA) dans 300m de diamètre), près de Exloo aux Pays-Bas . . . . .	69
6.2	Vue du ciel sans perturbateurs . . . . .	72
6.3	Vue du ciel avec les interférences . . . . .	72
6.4	Vitesse de convergence des algorithmes . . . . .	74
6.5	Vitesse de convergence des algorithmes . . . . .	74
6.6	Performance des algorithmes en fonction du Rapport Interférence sur Bruit (RIB)	75
6.7	Performance des algorithmes en fonction du nombre d'échantillons . . . . .	76
6.8	Performance des algorithmes en fonction du nombre d'échantillons . . . . .	76
6.9	Performance des algorithmes en fonction du nombre de pollueurs . . . . .	77
6.10	Performance des algorithmes en fonction du nombre de pollueurs . . . . .	78
6.11	Vue du ciel après projection . . . . .	78
6.12	Vue du ciel après soustraction . . . . .	79
6.13	Vue du ciel sur les antennes impaires. Scénario "land mobile" . . . . .	80
6.14	(a) : Décomposition en Valeurs Propres (DVP) avec Vecteur Propre (VecP) associé à plus grande Valeur Propre (ValP), (b) : ALS avec VecP associé à plus grande ValP, (c) : DVP avec VecP associé à 2nd plus grande ValP, (d) : ALS avec VecP associé à 2nd plus grande ValP . . . . .	81

6.15	Influence de l'estimation de la matrice de mélange, en fonction des statistiques utilisées. (a) : Ordre 2, (b) : Ordre 4 . . . . .	82
6.16	dim(Sous-Espace Interférences (SEI)) = 1 pour (a) : DVP et (b) : ALS . . . . .	83
6.17	dim(SEI) = 2 pour (a) : DVP et (b) : ALS . . . . .	83
6.18	Vue du ciel sur les antennes paires. Scénario "land mobile" . . . . .	84
6.19	dim(SEI) = 1 pour (a) : DVP et (b) : ALS . . . . .	84
6.20	dim(SEI) = 2 pour (a) : DVP et (b) : ALS . . . . .	85
6.21	Vue du ciel sur les antennes paires. Scénario "pager" . . . . .	85
6.22	dim(SEI) = 1 pour (a) : DVP et (b) : ALS . . . . .	86
6.23	dim(SEI) = 3 pour (a) : DVP et (b) : ALS . . . . .	86
6.24	dim(SEI) = 4 pour (a) : DVP et (b) : ALS . . . . .	87
6.25	dim(SEI) = 4 pour (a) : DVP et (b) : ALS . . . . .	87
6.26	dim(SEI) = 5 pour (a) : DVP et (b) : ALS . . . . .	87

# Liste des tableaux

1.1	Méthodes de diagonalisation conjointe . . . . .	5
3.1	Paramètres de simulations des algorithmes DESERT . . . . .	29
3.2	Temps d'exécution des algorithmes DESERT . . . . .	37
4.1	Paramètres de simulations des algorithmes ALS et NOODLES . . . . .	48
4.2	Temps d'exécution des algorithmes ALS et NOODLES . . . . .	55
5.1	Paramètres de simulations de signaux de télécommunications . . . . .	59
6.1	Paramètres de simulations pour les données simulés LOFAR . . . . .	73

# Acronymes

**ACDC** "Alternating Columns - Diagonal Centers"

**ACI** Analyse en Composantes Indépendantes

**ALS** "Alternating Least Square"

**BPSK** "Binary Phase Shift Keying"

**DESERT** "Diagonalization of complEx SEts using a Relative gradient"

**DIEM** "DIagonalization using Equivalent Matrices"

**DOMUNG** "Diagonalization Of Matrices Using Natural Gradient"

**DVP** Décomposition en Valeurs Propres

**ELS** "Enhanced Line Search"

**FAJD** "Fast Approximate Joint Diagonalization"

**FFDIAG** "Fast Frobenius DIAGonalization"

**HBA** "High Band Antennas"

**JADE** "Joint Approximate Diagonalization of Eigenmatrices"

**LBA** "Low Band Antennas"

**LOFAR** "LOw Frequency ARray"

**MC** Monte Carlo

**QPSK** "Quaternary Phase Shift Keying"

**RFI** "Radio frequency interference"

**RIB** Rapport Interférence sur Bruit

**RSB** Rapport Signal sur Bruit

**NOODLES** "NON Orthogonal Decomposition of complEx Sets"

**SEI** Sous-Espace Interférences

**SOBI** "Second Order Blind Identification"

**TBB** "Transient Buffer Board"

**U-WEDGE** "Uniformly Weighted Exhaustive Diagonalization with Gauss iterations"

**ValP** Valeur Propre

**VecP** Vecteur Propre

# Chapitre 1

## Introduction

### 1.1 Cadre de la thèse

Dans de nombreux algorithmes de traitement du signal, on calcule à partir des données une ou des matrices dont leur décomposition permet d'estimer les paramètres utiles au problème envisagé. C'est le cas par exemple en traitement d'antennes où, à partir de la décomposition en éléments propres de la matrice de covariance des données, on peut estimer le nombre de sources actives, l'espace signal associé et les différents angles d'arrivée. C'est aussi le cas en séparation de sources où, par exemple, on peut estimer une matrice de séparation à partir de la décomposition en éléments propres généralisée de deux matrices de corrélation des données, calculées à deux instants différents.

De nombreux types de matrices peuvent être envisagés, et ont été utilisées ces dernières années. On a, par exemple, des matrices liées :

- à des statistiques à l'ordre deux [1, 6, 20, 37],
- à des statistiques d'ordre supérieur [3, 10, 11, 12, 13, 36, 47, 55, 60],
- à une décomposition spectrale des données [27],
- à des décompositions temps-fréquences des données [7, 32].

Une autre type de solution pour la séparation de sources a consisté à utiliser un schéma de type déflation, où les sources sont extraites une à une du mélange [28, 30, 31].

Les matrices précédentes peuvent être réelles ou complexes. Dans quasiment tous les cas, ces matrices possèdent naturellement une propriété de symétrie. Dans le cas réel, elles sont très souvent symétriques et dans le cas complexe, elles sont très souvent hermitiennes. Plus récemment, un intérêt croissant est apparu concernant des matrices complexes symétriques. Par exemple en télécommunications numériques, de telles matrices peuvent être envisagées lorsque les signaux considérés sont non circulaires [48, 50, 56, 64]. Mais il existe d'autres domaines d'applications où cela est le cas [38, 67, 71].

En ce qui concerne les décompositions envisagées, elles dépendent bien évidemment du type de matrices et du problème considérés. Les plus classiques sont la décomposition en éléments propres et la décomposition en valeurs singulières. Cependant, la prise en compte simultanée de plusieurs matrices peut avoir un intérêt important (voire primordial) comme en séparation de sources. La première décomposition de ce type est la décomposition en éléments propres généralisée. Cette dernière décomposition ne prend en compte que deux matrices. Une généralisation en a été faite relativement récemment pour un nombre quelconque de matrices (en l'occurrence hermitiennes) sous la dénomination "diagonalisation conjointe d'un ensemble de matrices" dans [18]. La matrice de transformation recherchée était alors unitaire. Cela a donné lieu aux deux algorithmes de séparation de sources "Joint Approximate Diagonalization of Eigenmatrices" (JADE) [18] (généralisé à n'importe quel ordre supérieur à 3 dans [54]) et "Second Order Blind Identification" (SOBI) [6]. Cependant, la recherche d'une matrice unitaire nécessite au préalable une étape de blanchiment des données [46]. Cette étape n'étant pas sans conséquences au niveau des performances d'estimation [14]. L'idée a alors été de développer des algorithmes de diagonalisation conjointe sans contrainte unitaire.

Un certain nombre d'algorithmes de ce type ont été publiés depuis, par exemple [19, 21, 29, 63, 73]. En général, il s'agit d'optimiser un certain critère lié à la décomposition. Il existe deux grands types de critère. Le premier type correspond à un critère lié directement à une erreur de décomposition (ou erreur de modèle) qui sera appelé critère direct et le second type correspond à un critère lié à l'inverse de la transformation qui sera appelé critère inverse.

Des nouveaux problèmes sont apparus récemment, et ont nécessité une généralisation de la diagonalisation conjointe au cas où les matrices sont bloc-diagonales [39, 40].

On peut aussi considérer les ensembles de matrices comme des tenseurs, et faire les traitements directement sur ceux-ci. La méthode originelle s'appelle PARAFAC [41] et d'autres solutions ont été apportées par la suite [34, 35, 52, 53, 66].

La séparation de sources trouve des applications dans de nombreux problèmes tels que le biomédical [42, 44, 49], la spectroscopie [59], l'astronomie [27], la sismique [62, 72] et les communications numériques [20, 23, 37].

Le tableau 1.1 rappelle brièvement un ensemble d'algorithmes de la littérature qui traitent de la diagonalisation conjointe d'ensemble de matrices. Certaines de ces méthodes seront détaillées plus longuement dans la suite, et nous servirons de comparaison par rapport à celles que nous avons développé. On cite principalement "Alternating Columns - Diagonal Centers" (ACDC)[75], "Uniformly Weighted Exhaustive Diagonalization with Gauss iterations" (U-WEDGE)[70], "Fast Approximate Joint Diagonalization" (FAJD)[51], JADE[18], SOBI[6], "Diagonalization using Equivalent Matrices" (DIEM)[22], Joint-Diagonalization J-Di [68] et "Diagonalization Of Matrices Using Natural Gradient" (DOMUNG)[76].

Nom de l'algo.	Critère	Corps	Matrice transfo.	Ensemble
ACDC	direct	complexe	rectangulaire	hermitien/symétrique
U-WEDGE	inverse	complexe	carrée	hermitien(/symétrique?)
FAJD	inverse	complexe	carrée	hermitien
JADE	inverse	complexe	carrée	hermitien
SOBI	inverse	complexe	carrée	hermitien
DIEM	direct	complexe	carrée	hermitien
J-Di	direct	réel	carrée	symétrique
NOJD [2]	inverse	réel	carré	symétrique
DOMUNG	inverse	réel	carrée	symétrique

TABLE 1.1 – Méthodes de diagonalisation conjointe

## 1.2 Objectifs et contenu du document

L'objectif principal de thèse consiste à développer des algorithmes de diagonalisation conjointe de deux ensembles de matrices complexes. Le premier ensemble est constitué de matrices hermitiennes et le second ensemble de matrice complexe symétrique. L'objectif est ainsi de prendre en compte conjointement toute l'information disponible au niveau des statistiques des données. Ces algorithmes seront appliqués dans un premier temps au problème de la séparation de sources de type télécommunications numériques et dans un second temps au problème de la réjection d'interférences liées à des signaux de télécommunications numériques dans un cadre de radioastronomie.

Dans le chapitre 3, les algorithmes de type gradient que nous avons développé seront présentés. Nous proposerons des méthodes novatrices, ou des extensions de méthodes déjà existantes. Par exemple, nous verrons l'extension de DOMUNG développé dans [76] au cas complexe. De plus, nous proposerons des solutions qui nous permettront de nous affranchir du pré-traitement des données consistant à blanchir celles-ci afin, entre autre, d'estimer des matrices de mélange rectangulaires. De plus, nous proposerons des solutions qui nécessitent moins de ressources (temps de calcul ou mémoire). Des simulations permettront de valider les algorithmes développés.

Le chapitre 4 développera deux méthodes de type algébrique. Il s'agit de calculer analytiquement, en fonction des données en notre possession, la matrice de mélange ou la matrice de séparation. Ces solutions s'appuieront sur des résultats présentés dans [9], [25] ou [65]. Nous étendrons toujours ces algorithmes au cas complexe, en considérant des ensembles hermitiens, symétriques, et en combinant ces deux types d'ensemble. Enfin, nous présenterons des simulations numériques qui permettront de mettre en évidence les performances de nos algorithmes, comparés à des algorithmes classiques.

Dans les chapitres précédents, le bon comportement des méthodes développées n'a été illustré que dans le cadre de la diagonalisation conjointe. Le chapitre 5 présentera les performances des

algorithmes développés dans le cas de la séparation de sources, où les signaux sources sont des signaux typiques de télécommunications numériques.

Enfin, le chapitre 6 présentera l'application des méthodes développées au domaine radioastronomique. Le contexte est le suivant : les observations que font les astronomes peuvent être perturbées par des interférences issues du milieu terrestre, comme par exemple les signaux de radiocommunications. Notre objectif sera alors d'estimer au mieux ces perturbations afin de les éliminer des signaux reçus, et donc de permettre aux utilisateurs d'avoir l'observation la plus "propre" possible de ce qui se passe en dehors de la Terre. Dans un premier temps, nous vérifierons le bon comportement de nos algorithmes avec des données synthétiques simulant le type d'observations que l'on a classiquement. Puis, nous les appliquerons sur des données réelles.

## 1.3 Notations, définitions et outils

### 1.3.1 Notations classiques

On notera  $\mathbb{R}$  l'ensemble des nombres réels et  $\mathbb{C}$  l'ensemble des nombres complexes.

Dans ce qui suivra, les scalaires seront notés en minuscule  $m$ , les vecteurs en minuscule gras  $\mathbf{m}$ , les matrices en majuscule grasse  $\mathbf{M}$  et  $\hat{\mathbf{M}}$  est une estimée de  $\mathbf{M}$ .

Soit  $\mathbf{M} \in \mathbb{C}^{M \times N}$  une matrice complexe. Les majuscules simples seront utilisées pour déterminer des quantités (nombre de matrices dans un ensemble, taille des matrices etc.) :  $M, N, K$ . Le  $i$ ème élément d'un vecteur sera noté  $m_i$ , et l'élément ayant pour indice  $(i, j)$  dans une matrice  $\mathbf{M}$  sera noté  $M_{i,j}$ .

On notera  $\mathbf{M}^T$  sa transposée,  $\mathbf{M}^*$  sa conjuguée et  $\mathbf{M}^H$  sa transposée conjuguée.  $\mathbf{M}^{-1}$  est l'inverse d'une matrice,  $\mathbf{M}^\dagger$  la pseudo-inverse.

Le module d'un nombre complexe  $m$  est noté  $|m|$ .

$\Re(m)$  est la partie réelle de  $m$  et  $\Im(m)$  sa partie imaginaire.

**Produits matriciels** Soient  $\mathbf{M}$  et  $\mathbf{P}$  des matrices de dimension respectives  $M \times N$  et  $P \times Q$ . Le produit de Kronecker de  $\mathbf{M}$  et  $\mathbf{P}$ , noté  $\mathbf{M} \otimes \mathbf{P}$  et de dimension  $MP \times NQ$ , correspond à :

$$\mathbf{M} \otimes \mathbf{P} = \begin{pmatrix} m_{1,1}\mathbf{P} & \cdots & m_{1,n}\mathbf{P} \\ \vdots & \ddots & \vdots \\ m_{m,1}\mathbf{P} & \cdots & m_{m,n}\mathbf{P} \end{pmatrix} \quad (1.1)$$

Soient  $\mathbf{M}$  et  $\mathbf{P}$  des matrices de dimension respectives  $M \times N$  et  $P \times N$ . Le produit de Khatri-Rao de  $\mathbf{M}$  et  $\mathbf{P}$ , noté  $\mathbf{M} \odot \mathbf{P}$  et de dimension  $MP \times N$  est un produit de Kronecker par colonnes

défini par :

$$\mathbf{M} \odot \mathbf{P} = (\mathbf{M}_1 \otimes \mathbf{P}_1, \dots, \mathbf{M}_n \otimes \mathbf{P}_n) \quad (1.2)$$

### 1.3.2 Opération sur les matrices

$\text{Diag}\{\mathbf{M}\}$  annule les termes hors-diagonaux de  $\mathbf{M}$  :

$$\text{Diag}\{\mathbf{M}\} = \begin{pmatrix} m_{1,1} & 0 & \dots & 0 \\ 0 & m_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & m_{m,m} \end{pmatrix}$$

$\text{ZDiag}\{\mathbf{M}\}$  est l'opérateur annulant les termes diagonaux de  $\mathbf{M}$  :

$$\text{ZDiag}\{\mathbf{M}\} = \mathbf{M} - \text{Diag}\{\mathbf{M}\}$$

$\text{diag}\{\mathbf{M}\}$  est un vecteur contenant les termes diagonaux de  $\mathbf{M}$  et  $\text{Diag}\{\mathbf{m}\}$  construit une matrice diagonale à partir des éléments de  $\mathbf{m}$ .

$\text{tr}\{\mathbf{M}\}$  représente la trace de la matrice  $\mathbf{M}$ .

Soient  $\mathbf{M}$  et  $\mathbf{P}$  deux matrices carrées. On a :

$$\text{tr}\{\mathbf{MZdiag}\{\mathbf{P}\}\} = \text{tr}\{\text{Zdiag}\{\mathbf{M}\}\mathbf{P}\} \quad (1.3)$$

Soient  $\mathbf{M}$ ,  $\mathbf{N}$  et  $\mathbf{P}$  trois matrices carrées. On a :

$$\text{tr}\{\mathbf{MNP}\} = \text{tr}\{\mathbf{PMN}\} = \text{tr}\{\mathbf{NPM}\} \quad (1.4)$$

$\text{vec}\{\mathbf{M}\}$  correspond à l'empilement de toutes les colonnes de  $\mathbf{M}$  dans un vecteur. Une des propriétés de cet opérateur est :

$$\text{vec}\{\mathbf{MNP}\} = (\mathbf{P}^T \otimes \mathbf{M}) \text{vec}\{\mathbf{N}\} \quad (1.5)$$

De plus, si la matrice  $\mathbf{N}$  est diagonale, on a

$$\text{vec}\{\mathbf{MNP}\} = (\mathbf{P}^T \odot \mathbf{M}) \text{diag}\{\mathbf{N}\} \quad (1.6)$$

**Norme de Frobenius** La norme de Frobenius  $\|\mathbf{M}\|$  d'une matrice  $\mathbf{M}$  est définie comme suit :

$$\|\mathbf{M}\| = (\text{tr} \{\mathbf{M}^H \mathbf{M}\})^{\frac{1}{2}} = \left( \sum_{i,j} |M_{i,j}|^2 \right)^{\frac{1}{2}} \quad (1.7)$$

**Pseudo-inverse** Dans le cas où  $\mathbf{M}$  est une matrice complexe rectangulaire, on utilisera la pseudo-inverse de Moore-Penrose, notée  $\mathbf{M}^\dagger$  et vérifiant :

$$\begin{aligned} \mathbf{M} \mathbf{M}^\dagger \mathbf{M} &= \mathbf{M} \\ \mathbf{M}^\dagger \mathbf{M} \mathbf{M}^\dagger &= \mathbf{M}^\dagger \\ (\mathbf{M} \mathbf{M}^\dagger)^H &= \mathbf{M} \mathbf{M}^\dagger \\ (\mathbf{M}^\dagger \mathbf{M})^H &= \mathbf{M}^\dagger \mathbf{M} \end{aligned}$$

Si le rang de  $\mathbf{M}$  est égal à son nombre de lignes, on a :

$$\mathbf{M}^\dagger = \mathbf{M}^H (\mathbf{M} \mathbf{M}^H)^{-1}$$

De même, si le rang de  $\mathbf{M}$  est égal à son nombre de colonnes, on a :

$$\mathbf{M}^\dagger = (\mathbf{M}^H \mathbf{M})^{-1} \mathbf{M}^H$$

**Dérivation**  $\nabla$  sera utilisé pour le gradient complexe et nous introduisons deux définitions de gradient :

$$\nabla \mathcal{J}(\mathbf{M}) = \frac{\partial \mathcal{J}(\mathbf{M})}{\partial \mathbf{M}^*} = \frac{\partial \mathcal{J}(\mathbf{M})}{\partial \Re(\mathbf{M})} - j \frac{\partial \mathcal{J}(\mathbf{M})}{\partial \Im(\mathbf{M})} \quad (1.8)$$

qui représente le gradient naturel, et :

$$\nabla_r \mathcal{J}(\mathbf{M}) = \nabla \mathcal{J}(\mathbf{M}) \mathbf{M}^H = \frac{\partial \mathcal{J}(\mathbf{M})}{\partial \mathbf{M}^*} \mathbf{M}^H \quad (1.9)$$

qui est le gradient relatif, introduit dans [15, 17] (qui peut être vu comme un cas particulier de [4]).

La propriété suivante, liant la dérivation de matrices et l'opérateur trace, nous sera utile pour le calcul des différents gradients :

$$\frac{\partial \text{tr} \{\mathbf{Z}^H \mathbf{M}\}}{\partial \mathbf{Z}^*} = \mathbf{M} \quad (1.10)$$

## Contributions de l'auteur

### Colloques internationaux avec actes et comité de lecture

[ci4] T. Trainini and E. Moreau, "Variations Around Gradient Like Algorithms for Joint Diagonalization of Hermitian Matrices," in Proc. European Signal Processing Conference (EUSIPCO'2012), Bucharest, Romania, August 2012.

[ci3] G. Hellbourg, T. Trainini, R. Weber, E. Moreau, C. Capdessus and A.J. Boonstra, "RFI Subspace Estimation Techniques for New Generation Radio Telescopes," in Proc. European Signal Processing Conference (EUSIPCO'2012), Bucharest, Romania, August 2012.

[ci2] T. Trainini and E. Moreau, "A Least Squares Algorithm for Global Joint Decomposition of Complex Matrices Sets", in Proc. IEEE Fourth International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP'2011), San Juan, Puerto Rico, December 2011.

[ci1] T. Trainini, X.-L. Li, E. Moreau, and T. Adali, "A Relative Gradient Algorithm for Joint Decompositions of Complex Matrices", in Proc. European Signal Processing Conference (EUSIPCO'2010), Aalborg, Denmark, August 2010.

### Colloques nationaux avec actes et comité de lecture

[cn1] T. Trainini et E. Moreau, "Un algorithme ALS pour la décomposition conjointe de matrices complexes", dans Proc. 23ième colloque GRETSI sur le traitement du signal et des images, Bordeaux, France, Septembre 2011.

## Chapitre 2

# Présentation du problème

### 2.1 Diagonalisation conjointe

Dans ce paragraphe nous décrivons le problème de la diagonalisation conjointe d'ensembles de matrices complexes ainsi que les critères associés. Nous allons considérer deux ensembles de matrices complexes. Le premier concernera des matrices hermitiennes tandis que le second concernera des matrices complexes symétriques.

#### 2.1.1 Décompositions matricielles

Plus précisément, on considère un premier ensemble  $\mathcal{M}_h$  de  $N_h \in \mathbb{N}$  matrices carrées de dimension  $N_o \times N_o$  écrit de la manière suivante

$$\mathcal{M}_h = \{\mathbf{M}_i^{(h)} \in \mathbb{C}^{N_o \times N_o}, i = 1, \dots, N_h\}$$

Pour chacune des matrices de cet ensemble on suppose qu'il existe une matrice  $\mathbf{A}$  de dimension  $N_o \times N_s$ , de telle manière qu'idéalement on ait

$$\mathbf{M}_i^{(h)} = \mathbf{A} \mathbf{D}_i^{(h)} \mathbf{A}^H \quad (2.1)$$

où les matrices  $\mathbf{D}_i^{(h)}$  de dimension  $N_s \times N_s$ , sont toutes diagonales. Dans (2.1),  $(\cdot)^H$  est l'opérateur de transposition conjuguée de la matrice en argument. Ainsi, lorsque les matrices diagonales  $\mathbf{D}_i^{(h)}$  sont réelles, les matrices  $\mathbf{M}_i^{(h)}$  sont alors hermitiennes. Si les matrices  $\mathbf{D}_i^{(h)}$  sont complexes, on fera l'opération

$$\mathbf{M}_i^{(h)} \leftarrow \frac{1}{2} \left( \mathbf{M}_i^{(h)} + \mathbf{M}_i^{(h)H} \right)$$

afin de les rendre hermitiennes.

On suppose dans toute la suite que la matrice  $\mathbf{A}$  est de rang plein en colonne. L'ensemble de

telle matrice sera noté

$$\mathcal{A} = \{\mathbf{A} \in \mathbb{C}^{N_o \times N_s} \mid \mathbf{A} \text{ rang plein en colonne}\} \quad (2.2)$$

Ainsi, il existe une pseudo inverse de  $\mathbf{A}$  notée  $\mathbf{B}$ , i.e.

$$\mathbf{B} = \mathbf{A}^\dagger \quad (2.3)$$

de telle manière que l'on ait

$$\mathbf{B}\mathbf{A} = \mathbf{I} \quad (2.4)$$

où  $\mathbf{I}$  est la matrice identité. L'ensemble de telle matrice  $\mathbf{B}$  sera noté

$$\mathcal{B} = \left\{ \mathbf{B} \in \mathbb{C}^{N_s \times N_o} \mid \exists \text{ une matrice de rang plein } \mathbf{A} \text{ telle que } \mathbf{B} = \mathbf{A}^\dagger \right\} \quad (2.5)$$

Ainsi en utilisant (2.1), on a directement

$$\mathbf{B}\mathbf{M}_i^{(h)}\mathbf{B}^H = \mathbf{D}_i^{(h)} \quad (2.6)$$

On considère aussi un deuxième ensemble  $\mathcal{M}_t$  de  $N_t \in \mathbb{N}$  matrices carrées de dimension  $N_o \times N_o$  écrit de la manière suivante

$$\mathcal{M}_t = \{\mathbf{M}_i^{(t)} \in \mathbb{C}^{N_o \times N_o}, i = 1, \dots, N_t\}$$

On supposera que chacune de ces matrices peut se décomposer de la manière suivante

$$\mathbf{M}_i^{(t)} = \mathbf{A}\mathbf{D}_i^{(t)}\mathbf{A}^T \quad (2.7)$$

où la matrice  $\mathbf{A}$  est identique à celle ayant été utilisée dans la décomposition (2.1) et où es matrices  $\mathbf{D}_i^{(t)}$  de dimension  $N_s \times N_s$ , sont toutes diagonales. Dans (2.7),  $(\cdot)^T$  est l'opérateur de transposition de la matrice en argument. Ainsi toutes les matrices  $\mathbf{D}_i^{(t)}$  sont complexes symétriques. A partir de (2.7) et en utilisant (2.4), on a directement

$$\mathbf{B}\mathbf{M}_i^{(t)}\mathbf{B}^H = \mathbf{D}_i^{(t)} \quad (2.8)$$

A chaque fois que cela ne portera pas à confusion et afin de simplifier les notations, dans toute la suite on utilisera la formulation commune suivante pour les deux types précédents

$$\mathbf{M}_i = \mathbf{A}\mathbf{D}_i\mathbf{A}^\dagger \quad (2.9)$$

où l'opérateur  $(\cdot)^\ddagger$  correspond donc soit à une simple transposition soit à une transposition conjuguée. On aura donc aussi

$$\mathbf{B}\mathbf{M}_i\mathbf{B}^\ddagger = \mathbf{D}_i \quad (2.10)$$

### 2.1.2 Diagonalisation conjointe approchée

Le problème de diagonalisation conjointe considéré consiste, à partir d'un (resp. des) ensemble(s)  $\mathcal{M}_h$  ou (resp. et)  $\mathcal{M}_t$ , en l'estimation de la matrice  $\mathbf{A}$  pour le problème direct ou de la matrice  $\mathbf{B}$  pour le problème inverse. En pratique, les matrices dans les deux ensembles  $\mathcal{M}_h$  et  $\mathcal{M}_t$  sont estimées par exemple à partir d'estimateurs empiriques de statistiques d'ordre deux ou d'ordre supérieur à deux en utilisant un nombre fini de donnée. Ainsi les décompositions précédentes ne sont pas exactes mais "seulement" approchées. C'est la raison pour laquelle on parle alors de diagonalisation conjointe approchée. Dans ce cadre, on aborde le problème par l'intermédiaire d'un critère mesurant cette diagonalisation approchée dont l'optimisation permettra de trouver la matrice recherchée. Il existe de nombreux critères possibles associés au problème direct ou au problème inverse. Concernant le problème direct, le critère peut être le plus classique est le suivant

$$\mathcal{D}(\mathbf{A}, \{\mathbf{D}_i\}) = \sum_{i=1}^N \|\mathbf{M}_i - \mathbf{A}\mathbf{D}_i\mathbf{A}^\ddagger\|^2 \quad (2.11)$$

qui correspond à un critère quadratique d'ajustement de modèle de décomposition. Concernant le problème inverse, le critère le plus classique est

$$\mathcal{J}(\mathbf{B}) = \sum_{i=1}^N \|\mathbf{Z}\text{Diag}\{\mathbf{B}\mathbf{M}_i\mathbf{B}^\ddagger\}\|^2 \quad (2.12)$$

où l'opérateur  $\mathbf{Z}\text{Diag}\{\cdot\}$  mets à zéro l'ensemble des éléments diagonaux de la matrice en argument, *i.e.*  $(\mathbf{Z}\text{Diag}\{\mathbf{A}\})_{i,j} = (1 - \delta_{i,j})A_{i,j}$  où  $\delta_{i,j} = 1$  si  $i = j$  et 0 sinon. Ce critère étant nul pour une matrice diagonale, il s'agit d'une mesure quadratique de la "diagonalité" d'une matrice. Dans le seul cas hermitien, les deux critères précédents seront notés  $\mathcal{D}_h$  et  $\mathcal{J}_h$  et dans le seul cas symétrique, ils seront notés  $\mathcal{D}_s$  et  $\mathcal{J}_s$ . Lorsque que l'on voudra prendre en compte de manière simultanée les deux cas de matrices, on le fera par l'intermédiaire d'une simple combinaison linéaire des deux critères de la manière suivante. Pour le critère direct on posera

$$\mathcal{D}_\lambda = (1 - \lambda)\mathcal{D}_h + \lambda\mathcal{D}_s \quad (2.13)$$

où  $\lambda \in [0, 1]$  et pour le critère inverse on posera

$$\mathcal{J}_\lambda = (1 - \lambda)\mathcal{J}_h + \lambda\mathcal{J}_s \quad (2.14)$$

Les critères étant définis, le but est maintenant d'estimer les paramètres optimaux liés à leur optimisation. Pour le critère direct, l'objectif est d'estimer une matrice  $\widehat{\mathbf{A}}$  et des matrices diagonales  $\widehat{\mathbf{D}}_i$  telles que

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{D}}_i) = \arg \min_{\substack{\mathbf{A} \in \mathcal{A} \\ \mathbf{D}_i \in \mathcal{D}_{i_a}}} \mathcal{D}(\mathbf{A}, \{\mathbf{D}_i\}) \quad (2.15)$$

où  $\mathcal{A}$  est l'ensemble défini en (2.2) et  $\mathcal{D}_{i_a}$  est l'ensemble des matrices diagonales. Pour le critère inverse, l'objectif est d'estimer une matrice  $\widehat{\mathbf{B}}$  telle que

$$\widehat{\mathbf{B}} = \arg \min_{\mathbf{B} \in \mathcal{B}} \mathcal{J}(\mathbf{B}) \quad (2.16)$$

où  $\mathcal{B}$  est l'ensemble défini en (2.5).

### 2.1.3 Au sujet de l'identifiabilité

Il est d'abord important de remarquer qu'il ne sera pas possible, à partir des seuls critères précédents, d'identifier exactement  $\mathbf{A}$  ou  $\mathbf{B}$ . Cela est lié à l'aspect "aveugle" de l'identification. Dans ce cadre, on se contente, mais c'est très classique et largement suffisant en pratique, d'estimer  $\mathbf{A}$  ou  $\mathbf{B}$  au produit d'une matrice diagonale inversible près et d'une matrice de permutation près.

## 2.2 Lien avec la séparation de sources

Dans ce paragraphe nous faisons un bref rappel du problème de la séparation de sources dans un cadre de mélange linéaire et instantané. Puis nous présentons son lien avec la diagonalisation conjointe de matrices. En séparation de sources, on suppose disposer d'un certain nombre d'observations résultant du mélange d'un certain nombre de signaux d'intérêt appelés sources. Dans le cas le plus simple le modèle de mélange est le suivant [42]

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \quad (2.17)$$

$$\mathbf{y}(t) = \mathbf{x}(t) + \mathbf{b}(t) \quad (2.18)$$

où  $\mathbf{A}$  est la matrice de mélange de dimension  $N_o \times N_s$ ,  $\mathbf{s}(t)$  le vecteur  $N_s \times 1$  des signaux sources,  $\mathbf{b}(t)$  le vecteur  $N_o \times 1$  des bruits additifs,  $\mathbf{x}(t)$  le vecteur  $N_o \times 1$  des observations sans bruit et  $\mathbf{y}(t)$  le vecteur  $N_o \times 1$  des observations bruitées. Par rapport à ce modèle, il est classique d'imposer un certain nombre d'hypothèses. En l'occurrence, on supposera toujours les signaux sources centrés, de puissance unité et statistiquement indépendants. En ce qui concerne le mélange, dans le cas sur-déterminé, *i.e.* lorsque  $N_o \geq N_s$ , la matrice  $\mathbf{A}$  sera supposé de rang plein en colonne et dans le cas sous-déterminé, *i.e.* lorsque  $N_o < N_s$ , la matrice de mélange sera supposée de rang plein

avec des vecteurs colonnes deux à deux distincts. Dans le cas sur-déterminé, le but est alors d'estimer une matrice de séparation  $\mathbf{B}$  de telle manière que le vecteur

$$\mathbf{z}(t) = \mathbf{B}\mathbf{x}(t) \quad (2.19)$$

correspond aux signaux sources que l'on recherche. Dans ce cadre la matrice de séparation  $\mathbf{B}$  est recherchée en imposant des composantes indépendantes aux vecteurs  $\mathbf{z}(t)$ . On parle alors d'analyse en composantes indépendantes. Il s'agit donc de restaurer la propriété d'indépendance statistique perdue par le mélange. Cela sera le cas [24] lorsque l'on aura trouvé une matrice de séparation  $\mathbf{B}$  de telle façon que

$$\mathbf{B}\mathbf{A} = \mathbf{D}\mathbf{P} \quad (2.20)$$

où  $\mathbf{D}$  est une matrice diagonale inversible et où  $\mathbf{P}$  est une matrice de permutation.

Le problème peut alors se schématiser de la façon suivante :

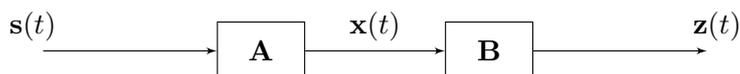


FIGURE 2.1 – Schéma de la séparation de sources

Cependant, il est rare de disposer de données totalement non bruitées. Ainsi, notre problème sera d'estimer cette matrice de séparation  $\mathbf{B}$  à partir des observations bruitées (2.18), et donc la schématisation est la suivante :

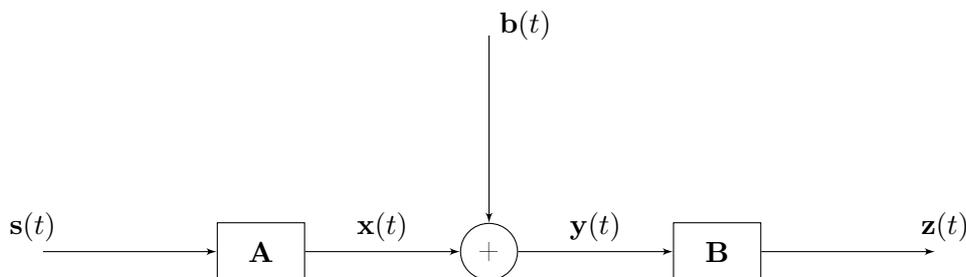


FIGURE 2.2 – Schéma dans le cas bruité

Dans le but de restaurer cette propriété d'indépendance statistique, une première étape de décorrélation peut être envisagée. Cette première étape associée à une normalisation en puissance des signaux est souvent appelée blanchiment des observations. Dans le cas d'observations bruitées, elle peut avoir un intérêt supplémentaire de réduction de bruit par projection sur l'espace signal lorsque que le nombre d'observations  $N_o$  est strictement plus grand que le nombre de sources  $N_s$ . Cela peut aussi avoir l'intérêt, dans le cas sur-déterminé, de ramener le problème à un problème carré plutôt qu'initialement rectangulaire.

Dans ce cas, la matrice de mélange  $\mathbf{B}$  se décompose en un produit matriciel entre une matrice de blanchiment  $\mathbf{W}$  et une matrice unitaire  $\mathbf{U}$  :

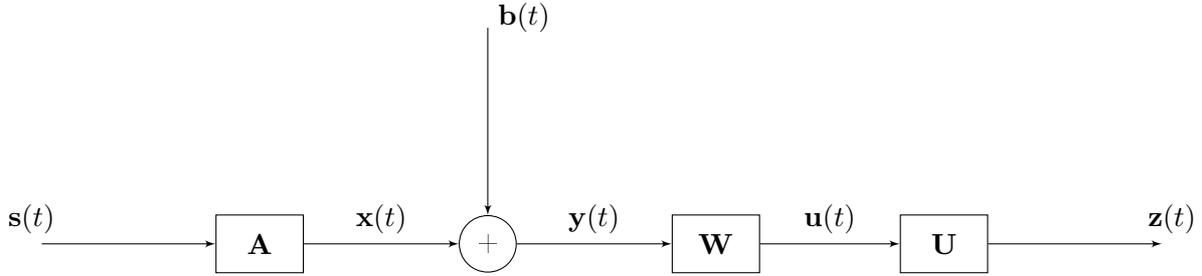


FIGURE 2.3 – Schéma dans le cas d'un blanchiment

De très nombreuses méthodes ont été proposées afin de résoudre ce problème de séparation de sources. Nous allons uniquement dans le cadre de la thèse nous intéresser aux méthodes dites de diagonalisations conjointes de matrices qui, quasiment toutes, tirent leur origine dans les travaux initiaux de Cardoso et Soughoumiac [18]. Dans ce cadre, en général, des statistiques sont estimés et “rangés” dans des matrices et des décompositions algébriques conjointes de ces matrices permettent l'estimation d'une matrice de séparation. On peut considérer les statistiques d'ordre deux par l'intermédiaire de la matrice de corrélation classique des observations

$$\mathbf{R}_x(t, \tau) = \mathbb{E} \{ \mathbf{x}(t) \mathbf{x}(t - \tau)^H \} \quad (2.21)$$

et/ou de la seconde matrice de corrélation des observations

$$\mathbf{R}'_x(t, \tau) = \mathbb{E} \{ \mathbf{x}(t) \mathbf{x}(t - \tau)^T \} \quad (2.22)$$

dans le cas où les signaux sources sont non circulaires. Dans le cas sans bruit, en remplaçant  $\mathbf{x}(t)$  par son expression (2.17), on obtient directement

$$\mathbf{R}_x(t, \tau) = \mathbf{A} \mathbf{R}_s(t, \tau) \mathbf{A}^H \quad (2.23)$$

$$\mathbf{R}'_x(t, \tau) = \mathbf{A} \mathbf{R}'_s(t, \tau) \mathbf{A}^T \quad (2.24)$$

où les deux matrices de corrélation des sources  $\mathbf{R}_s(t, \tau)$  et  $\mathbf{R}'_s(t, \tau)$  sont diagonales car les sources sont supposées statistiquement indépendantes.

On peut aussi considérer des statistiques d'ordre supérieur comme par exemple les statistiques

d'ordre quatre suivantes

$$C_{x,a,b,c,d} = \text{Cum} \{x_a, x_b^*, x_c, x_d^*\} \quad (2.25)$$

$$C'_{x,a,b,c,d} = \text{Cum} \{x_a, x_b, x_c, x_d^*\} \quad (2.26)$$

$$C''_{x,a,b,c,d} = \text{Cum} \{x_a, x_b, x_c, x_d\} \quad (2.27)$$

avec  $1 \leq a, b, c, d \leq N$ .

Il doit être précisé que les deux derniers types de statistique ne peuvent concerner que des signaux non circulaires. A partir de ces statistiques d'ordre quatre, il est possible de construire des matrices de différentes façons. La manière certainement la plus simple consiste à considérer

$$(\mathbf{M}_{x,c,d})_{a,b} = C_{x,a,b,c,d} \quad (2.28)$$

$$(\mathbf{M}'_{x,c,d})_{a,b} = C'_{x,a,b,c,d} \quad (2.29)$$

$$(\mathbf{M}''_{x,c,d})_{a,b} = C''_{x,a,b,c,d} \quad (2.30)$$

Dans le cas sans bruit, en remplaçant  $\mathbf{x}(t)$  par son expression (2.17), on obtient directement

$$\mathbf{M}_{x,c,d} = \mathbf{A}\mathbf{M}_{s,x,c,d}\mathbf{A}^H \quad (2.31)$$

$$\mathbf{M}'_{x,c,d} = \mathbf{A}\mathbf{M}'_{s,x,c,d}\mathbf{A}^T \quad (2.32)$$

$$\mathbf{M}''_{x,c,d} = \mathbf{A}\mathbf{M}''_{s,x,c,d}\mathbf{A}^T \quad (2.33)$$

où les matrices

$$\mathbf{M}_{s,x,c,d} = \text{Cum} \{s_a, s_b^*, x_c, x_d^*\} \quad (2.34)$$

$$\mathbf{M}'_{s,x,c,d} = \text{Cum} \{s_a, s_b, x_c, x_d^*\} \quad (2.35)$$

$$\mathbf{M}''_{s,x,c,d} = \text{Cum} \{s_a, s_b, x_c, x_d\} \quad (2.36)$$

sont diagonales car les sources sont supposées statistiquement indépendantes.

Ainsi, à partir des statistiques d'ordre deux et/ou des statistiques d'ordre quatre, il est possible de construire des ensembles de matrices se diagonalisant sous deux factorisations particulières. Dans cette thèse, nous ne nous préoccupons que du cas où les deux derniers indices sont identiques, *i.e.*  $c = d$ . Les modèles que nous utiliserons alors sont :

$$\mathbf{M}_x = \mathbf{A}\mathbf{M}_{s,x,c,c}\mathbf{A}^H \quad (2.37)$$

$$\mathbf{M}'_x = \mathbf{A}\mathbf{M}'_{s,x,c,c}\mathbf{A}^T \quad (2.38)$$

$$\mathbf{M}''_x = \mathbf{A}\mathbf{M}''_{s,x,c,c}\mathbf{A}^T \quad (2.39)$$

On considèrera donc des matrices hermitiennes de la forme

$$\mathbf{M}_i = \mathbf{A}\mathbf{D}_i\mathbf{A}^H \quad i = 1, \dots, K_h \quad (2.40)$$

avec  $\mathbf{D}_i$  des matrices diagonales. Ce modèle décrit de manière générique les décompositions considérées en (2.23) et (2.37).

On considèrera aussi des matrices complexes symétriques de la forme

$$\mathbf{M}_i = \mathbf{A}\mathbf{D}_i\mathbf{A}^T \quad i = 1, \dots, K_s \quad (2.41)$$

Ce modèle décrit de manière générique les décompositions considérées en (2.24), (2.38) et (2.39).

## 2.3 Description de quelques algorithmes

Dans ce paragraphe nous décrivons trois algorithmes de diagonalisation conjointe de matrices qui nous serviront de comparaison lors des simulations informatiques.

Le schéma global pour l'algorithme U-WEDGE décrit dans [70] est développé dans le pseudocode Algorithme 1 ci-dessous. Pour cette méthode, les ensembles sont réels.

### Algorithme 1: UWEDGE

<p>Critère : <math>\mathcal{J}(\mathbf{B}, \mathbf{A}) = \sum_{i=1}^N \ \mathbf{B}\mathbf{A}_i\mathbf{B}^T - \mathbf{A}\text{Diag}(\mathbf{B}\mathbf{M}_i\mathbf{B}^T)\mathbf{A}^T\ ^2</math></p> <p><b>for</b> <math>k = 1, 2, \dots</math> jusqu'à convergence <b>do</b></p> <p>  Calculer <math>\mathbf{R}_{s_i} = \mathbf{B}\mathbf{M}_i\mathbf{B}^T</math> pour <math>i = 1, \dots, N</math></p> <p>  Construire <math>\mathbf{R} = [\text{diag}(\mathbf{R}_{s_1}), \dots, \text{diag}(\mathbf{R}_{s_N})]</math> et <math>\mathbf{G} = \mathbf{R}\mathbf{R}^T</math></p> <p>  <b>for</b> <math>l = 1, \dots, N_s</math>, (où <math>N_s</math> est le nombre de colonnes) <b>do</b></p> <p>    Calculer <math>\mathbf{C}(l) = \sum_{i=1}^N \mathbf{R}_{s_i}(l) \oslash \mathbf{R}(i)</math>, avec <math>\oslash</math> le produit terme à terme.</p> <p>  <b>end for</b></p> <p>  Calculer <math>\mathbf{A} = \mathbf{I} + \frac{\mathbf{C}\oslash\mathbf{G} - \text{Diag}(\mathbf{G})\mathbf{C}^T}{(\mathbf{G}\oslash\mathbf{G}^T - \text{diag}(\mathbf{G})\text{diag}(\mathbf{G})^T) + \mathbf{I}}</math></p> <p>  <math>\mathbf{B} \leftarrow \mathbf{A}^{-1}\mathbf{B}</math></p> <p>  Normaliser les lignes de <math>\mathbf{B}</math></p> <p><b>end for</b></p>
---

Le schéma global pour l'algorithme ACDC décrit dans [75] est développé dans le pseudocode Algorithme 2 ci-dessous. Il prend en compte l'utilisation de matrices hermitiennes ou symétriques.

<b>Algorithme 2: ACDC</b>
<p>Critère : <math>\mathcal{C}_{\mathcal{L}\mathcal{S}}(\mathbf{B}, \Lambda_i) = \sum_{i=1}^N \ \mathbf{M}_i - \mathbf{B}\Lambda_i\mathbf{B}^\dagger\ ^2</math></p> <p><b>for</b> <math>k = 1, 2, \dots</math> jusqu'à convergence <b>do</b></p> <p style="padding-left: 20px;"><b>if</b> AC Phase <b>then</b></p> <p style="padding-left: 40px;">Calculer <math>\mathbf{P} = \sum_{i=1}^N w_i \lambda_i^l \left[ \mathbf{M}_i - \sum_{\substack{n=1 \\ n \neq l}}^N \lambda_{n,i} \mathbf{b}_n \mathbf{b}_n^\dagger \right]^{(H^\dagger)}</math></p> <p style="padding-left: 40px;">Sélectionner la plus grande valeur propre <math>\mu</math>, ainsi que le vecteur <math>\beta</math> associé, de <math>\mathbf{P}</math> (si <math>\dagger = H</math>) ou <math>\mathbf{Q} = \begin{bmatrix} \text{Real}(\mathbf{P}) &amp; -\text{Imag}(\mathbf{P}) \\ -\text{Imag}(\mathbf{P}) &amp; -\text{Real}(\mathbf{P}) \end{bmatrix}</math> (si <math>\dagger = T</math>)</p> <p style="padding-left: 40px;"><b>if</b> <math>\mu = 0</math> <b>then</b></p> <p style="padding-left: 60px;"><math>\mathbf{b}_l = 0</math></p> <p style="padding-left: 40px;"><b>else</b></p> <p style="padding-left: 60px;"><b>if</b> <math>\dagger = H</math> <b>then</b></p> <p style="padding-left: 80px;"><math>\mathbf{b}_l = \frac{\beta \sqrt{\mu}}{\sqrt{\sum_{i=1}^N w_i (\lambda_i^l)^2}}</math></p> <p style="padding-left: 60px;"><b>else</b></p> <p style="padding-left: 80px;"><math>\mathbf{b}_l = \frac{(\beta(1) + j\beta(2))\sqrt{\mu}}{\sqrt{\sum_{i=1}^N w_i  \lambda_i^l ^2}}</math></p> <p style="padding-left: 60px;"><b>end if</b></p> <p style="padding-left: 40px;"><b>end if</b></p> <p style="padding-left: 20px;"><b>else if</b> DC Phase <b>then</b></p> <p style="padding-left: 40px;">Calculer <math>\mathbf{G} = \left[ (\mathbf{B}^H \mathbf{B})^{(T^\dagger)} \odot (\mathbf{B}^H \mathbf{B}) \right]^{-1}</math></p> <p style="padding-left: 40px;"><b>for</b> <math>i = 1, \dots, N</math> <b>do</b></p> <p style="padding-left: 60px;"><math>\Lambda_i = \text{diag}(\mathbf{G} \text{diag}(\mathbf{B}^H \mathbf{M}_i \mathbf{B}^{(H^\dagger)}))</math></p> <p style="padding-left: 40px;"><b>end for</b></p> <p style="padding-left: 20px;"><b>end if</b></p> <p><b>end for</b></p>

Le schéma global pour l'algorithme FAJD décrit dans [51] est développé dans le pseudocode Algorithme 3 ci-dessous. Pour cet algorithme, les ensembles sont hermitiens.

**Algorithme 3: FAJD**

Critère :  $\mathcal{J}(\mathbf{W}) = \sum_{i=1}^N \alpha_i \sum_{m=1}^{N_s} \sum_{\substack{n=1 \\ n \neq m}}^{N_s} |[\mathbf{W}^H \mathbf{M}_i \mathbf{W}]_{ij}|^2 - \beta \log |\det(\mathbf{W})|$

$\alpha_i, \beta$  sont des coefficients de pondération positifs.

**for**  $k = 1, 2, \dots$  jusqu'à convergence **do**

**for**  $m = 1, \dots, N_s$  **do**

Calculer  $\mathbf{Q}_i = \sum_{\substack{n=1 \\ n \neq m}}^{N_s} \alpha_i [\mathbf{M}_i \mathbf{w}_n \mathbf{w}_n^H \mathbf{M}_i^H + \mathbf{M}_i^H \mathbf{w}_n \mathbf{w}_n^H \mathbf{M}_i]$

Trouver  $1 \leq s \leq N_s$  et construire  $\mathbf{E}_{s, N_s} \mathbf{W} \mathbf{E}_{n, N_s} = \begin{bmatrix} \mathbf{G} & \mathbf{u} \\ \mathbf{g}^H & v \end{bmatrix}$ , avec

$$[\mathbf{E}_{s,t}]_{m,n} = \begin{cases} 1 & \text{si } m = s, n = t \text{ ou } m = t, n = s \\ 1 & \text{si } m = n \text{ et } m \neq s, t \\ 0 & \text{sinon} \end{cases}$$

de telle sorte que  $\mathbf{G}$  soit inversible

Calculer  $\mathbf{d}_s = ([-\mathbf{g}^H \mathbf{G}^{-1}, 1] \mathbf{E}_{s, N_s})^H$

Remplacer la  $m$ ème colonne de  $\mathbf{W}$  avec  $\mathbf{w}_m^{\text{opt}} = \frac{\sqrt{\beta} \mathbf{Q}_i^{-1} \mathbf{d}_s}{\sqrt{2 \mathbf{d}_s^H \mathbf{Q}_i^{-1} \mathbf{d}_s}}$

**end for**

**end for**

Ces algorithmes fonctionnent dans des cadres différents (cf Table 1.1), en fonction de :

- la dimension de la matrice (carrée, rectangulaire),
- des variables (réelles, complexes),
- des ensembles (hermitiens, symétriques).

## Chapitre 3

# Algorithmes de Gradient

Dans ce qui va suivre, nous présenterons un ensemble de méthodes itératives qui utilisent les fonctions de coût présentées au paragraphe 2.1.2 et des mises à jour basées sur le gradient. Cela a déjà été considéré, en particulier dans [43, 76, 26, 45, 69]. Nous allons généraliser certaines approches dans les deux cas complexes, mais aussi proposer des simplifications.

### 3.1 Description générale

Dans ce chapitre nous allons développer un ensemble d'algorithmes de type gradient afin d'estimer la matrice de séparation  $\mathbf{B}$  par minimisation du critère  $\mathcal{J}(\mathbf{B})$  dont on rappelle l'expression

$$\mathcal{J}(\mathbf{B}) = \sum_{i=1}^N \|\mathbf{Z} \text{Diag}\{\mathbf{B} \mathbf{M}_i \mathbf{B}^\dagger\}\|^2 \quad (3.1)$$

Le principe des algorithmes de type gradient est relativement simple, il s'agit, à partir d'une matrice initiale, de construire de manière itérative une estimée de  $\mathbf{B}$  minimisant à chaque itération le critère choisi. En ce qui concerne l'estimation d'une matrice, il existe deux grands types de mise à jour. Le premier type est à une mise à jour additive correspondant à

$$\mathbf{B} \leftarrow \mathbf{B} + \mathbf{Z}_a$$

où la matrice  $\mathbf{Z}_a$  est le terme de mise à jour dépendant du gradient du critère. Ce premier type est très classique. En effet, en utilisant la définition du gradient (1.8) à  $\mathcal{J}(\mathbf{B})$ , on pose pour un algorithme de gradient classique

$$\mathbf{Z}_a = -\mu \nabla \mathcal{J}(\mathbf{B})$$

où  $\mu$  est une petite constante positive appelé le pas de mise à jour. Le deuxième type est à une

mise à jour multiplicative correspondant à

$$\mathbf{B} \leftarrow (\mathbf{I} + \mathbf{Z})\mathbf{B} \quad (3.2)$$

où la matrice  $\mathbf{Z}$  est le terme de mise à jour dépendant également du gradient du critère.

Ce deuxième type est moins classique car il ne peut correspondre qu'au cas matriciel. La notion de gradient relatif définie en (1.9), et appliquée à  $\mathcal{J}(\mathbf{B})$ , permet de considérer le terme de mise à jour suivant

$$\mathbf{Z} = -\mu \nabla_r \mathcal{J}(\mathbf{B}) = \mu \mathbf{F} \quad (3.3)$$

De plus, il a été montré dans [16], qu'il existait un lien entre le gradient en  $\mathbf{B}$  et celui en  $\mathbf{Z}$

$$\nabla_r \mathcal{J}(\mathbf{B}) = \frac{\partial \mathcal{J}((\mathbf{I} + \mathbf{Z})\mathbf{B})}{\partial \mathbf{Z}^*}$$

Il se trouve que, dans notre cas, la dépendance du critère par rapport à la matrice recherchée  $\mathbf{B}$  peut être rendue implicite, ce qui aura un grand avantage dans la suite. Il suffit pour cela de considérer une mise à jour en parallèle de l'ensemble considéré de matrices.

A partir de l'ensemble initial et en posant

$$\mathbf{T}_i = \mathbf{B}\mathbf{M}_i\mathbf{B}^\dagger \quad (3.4)$$

en utilisant la mise à jour (3.2), le critère  $\mathcal{J}(\mathbf{B})$  peut se réécrire

$$\mathcal{J}(\mathbf{Z}) = \sum_{i=1}^N \|\mathbf{Z} \text{Diag}\{(\mathbf{I} + \mathbf{Z})\mathbf{T}_i(\mathbf{I} + \mathbf{Z})^\dagger\}\|^2 \quad (3.5)$$

tout en considérant donc une mise à jour en parallèle des matrices  $\mathbf{T}_i$  en (3.4) de la manière suivante

$$\mathbf{T}_i \leftarrow (\mathbf{I} + \mathbf{Z})\mathbf{T}_i(\mathbf{I} + \mathbf{Z})^\dagger \quad (3.6)$$

Une mise à jour multiplicative possède de nombreux avantages dans le cadre de notre problème.

Tout d'abord elle permet (dans une certaine mesure, *i.e.* avec un pas suffisamment petit) de garder le caractère inversible de la matrice  $\mathbf{B}$  au cours des itérations. Il s'agit d'un algorithme de type gradient relatif, et comme montré dans [76], cela permet d'avoir des performances indépendantes de  $\mathbf{B}$ . En effet l'algorithme global ne dépend que de  $\mathbf{Z}$ .

Ainsi dans toute la suite, cette seule mise à jour multiplicative sera considérée.

Le schéma global pour l'algorithme de gradient décrit dans cette partie est développé dans le pseudocode Algorithme 4 ci-dessous :

L'algorithme précédent dépend d'un paramètre supplémentaire qui est le pas de mise à jour

**Algorithme 4:** Schéma Global pour les algorithmes de gradient

Initialisation : Ensemble de matrices $\mathbf{M}_i$ , matrices initiales $\mathbf{B}^{(0)}$ <b>for</b> $k = 1, 2, \dots$ jusqu'à convergence <b>do</b> Mise à jour des matrices $\mathbf{T}_i$ suivant (3.6) Calcul du gradient (classique ou relatif) de (3.1) Mise à jour de la matrices $\mathbf{Z}$ avec (3.3) puis de $\mathbf{B}$ avec (3.2) <b>end for</b>
---

$\mu$ , cf. (3.3). Dans le cas le plus simple, ce pas est fixé à une certaine valeur définie par l'utilisateur. Cette stratégie très simple est largement sous optimale. Une idée courante consiste alors à considérer ce pas comme un paramètre supplémentaire dans le problème d'optimisation et donc à le calculer à chaque itération par minimisation du critère considéré.

Dans notre cas en reportant l'expression de  $\mathbf{Z}$  en (3.3) dans (3.5), il est alors clair que  $\mathcal{J}$  est une fonction quartique de  $\mu$  et que sa minimisation ne posera pas de problème. C'est ce que nous ferons dans la suite.

## 3.2 Calcul du gradient

Dans ce paragraphe nous allons calculer le gradient du critère dans le cas hermitien puis dans le cas symétrique. Nous allons aussi déterminer les coefficients du polynôme de degré quatre pour le calcul du pas optimal.

### 3.2.1 Cas hermitien

On considère le critère suivant correspondant au seul cas hermitien

$$\mathcal{J}_h(\mathbf{Z}) = \sum_{i=1}^N \|\mathbf{Z} \text{Diag}\{(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^H\}\|^2 \quad (3.7)$$

Dans ce qui suivra, seuls les résultats seront rappelés. Tous les développements des calculs sont effectués dans l'Annexe A.1.1.

Le gradient de ce critère est donné par la formule suivante :

$$\begin{aligned} \frac{\partial \mathcal{J}_h(\mathbf{Z})}{\partial \mathbf{Z}^*} &= \sum_{i=1}^N \mathbf{Z} \text{Diag}\{(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^H\} (\mathbf{I} + \mathbf{Z}) \mathbf{T}_i^H \\ &\quad + \mathbf{Z} \text{Diag}\{(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^H\}^H (\mathbf{I} + \mathbf{Z}) \mathbf{T}_i \end{aligned} \quad (3.8)$$

### 3.2.2 Cas symétrique

On considère le critère suivant correspondant au seul cas symétrique

$$\mathcal{J}_s(\mathbf{Z}) = \sum_{i=1}^N \|\mathbf{Z} \text{Diag}\{(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^T\}\|^2 \quad (3.9)$$

Le calcul du gradient de (3.9), effectué dans l'Annexe A.1.2, donne :

$$\frac{\partial \mathcal{J}_s(\mathbf{Z})}{\partial \mathbf{Z}^*} = 2 \sum_{i=1}^N \mathbf{Z} \text{Diag}\{(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^T\} (\mathbf{I} + \mathbf{Z})^* \mathbf{T}_i^* \quad (3.10)$$

### 3.2.3 Pas optimal

La recherche du pas optimal peut se faire de manière globale. En remplaçant  $\mathbf{Z}$ , exprimé comme en (3.3), dans (3.5), on obtient le polynôme suivant :

$$\mathcal{J}(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \{a_{i,0} + \mu a_{i,1} + \mu^2 a_{i,2} + \mu^3 a_{i,3} + \mu^4 a_{i,4}\} \quad (3.11)$$

dont les calculs sont développés dans l'Annexe B.1 , et dont les coefficients  $a_{i,j} \forall j = 0, \dots, 4$  y sont donnés.

Le schéma global pour la détermination du pas optimal est détaillé dans l'Algorithme 5 :

<b>Algorithme 5:</b> Pas Optimal
Init : coefficients $a_{i,j}, j = 0, \dots, deg$ avec $deg$ le degré du polynôme Détermination des racines $\mu$ de la dérivée du polynôme (3.11) $\mu_{opt} = \underset{\mu}{\text{argmin}} \{\mathcal{J}(\mathbf{Z})\}$

## 3.3 Gradient et approximations du critère

Dans un but de réduction de la complexité des calculs, nous allons maintenant développer plusieurs approximations du critère initial permettant un calcul de gradient simplifié. Ces approximations sont basées sur des idées développées dans les références [76, 77] qui ont conduit aux algorithmes DOMUNG et "Fast Frobenius DIAGonalization" (FFDIAG) .

### 3.3.1 Approximations sur le critère

Principalement deux hypothèses simplificatrices peuvent être considérées. La première concerne le terme de mise à jour  $\mathbf{Z}$  dans (3.2). En fait l'algorithme itératif est construit de telle façon que ce terme de mise à jour tend à s'annuler afin d'obtenir un point stationnaire. Ainsi on peut supposer raisonnablement que, relativement rapidement, on ait  $\|\mathbf{Z}\| \ll 1$ . Cela a deux conséquences.

La première conséquence concerne la relation de mise à jour en (3.2) où  $\mathbf{Z}$  n'intervient que par l'intermédiaire de la somme  $\mathbf{I} + \mathbf{Z}$ . Ainsi si l'on suppose que la norme de  $\mathbf{Z}$  est très petite alors les éléments diagonaux de  $\mathbf{Z}$  peuvent être considérés très petits devant 1 correspondant aux éléments diagonaux de la matrice identité  $\mathbf{I}$ .

Ainsi, à la place de (3.2), on peut considérer la nouvelle mise suivante

$$\mathbf{B} \leftarrow (\mathbf{I} + \mathbf{Z}\text{Diag}\{\mathbf{Z}\})\mathbf{B} \quad (3.12)$$

où les termes diagonaux de  $\mathbf{Z}$  sont négligés devant 1.

La deuxième conséquence concerne directement le critère  $\mathcal{J}(\mathbf{Z})$  en (3.5). En effet, en supposant  $\|\mathbf{Z}\| \ll 1$ , alors au premier ordre on a

$$(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^\dagger \approx \mathbf{T}_i + \mathbf{Z}\mathbf{T}_i + \mathbf{T}_i\mathbf{Z}^\dagger \quad (3.13)$$

conduisant à une approximation  $\mathcal{J}_{a_1}(\mathbf{Z})$  du critère  $\mathcal{J}(\mathbf{Z})$  donnée par

$$\mathcal{J}_{a_1}(\mathbf{Z}) = \sum_{i=1}^N \|\mathbf{Z}\text{Diag}\{\mathbf{T}_i + \mathbf{Z}\mathbf{T}_i + \mathbf{T}_i\mathbf{Z}^\dagger\}\|^2 \quad (3.14)$$

Une seconde hypothèse simplificatrice peut être considérée concernant cette fois les matrices  $\mathbf{T}_i$  mises à jour au cours des itérations en (3.6). En effet l'algorithme est aussi construit de telle façon que ces matrices tendent à devenir des matrices diagonales au cours des itérations.

Ainsi on peut encore raisonnablement supposer que, relativement rapidement, on ait

$$\|\mathbf{Z}\text{Diag}\{\mathbf{T}_i\}\| \ll 1$$

Cela a la conséquence suivante. En notant

$$\mathbf{T}_i^D = \text{Diag}\{\mathbf{T}_i\} \quad \text{et} \quad \mathbf{T}_i^Z = \mathbf{Z}\text{Diag}\{\mathbf{T}_i\} \quad (3.15)$$

où  $\text{Diag}\{\cdot\}$  est l'opérateur matriciel renvoyant une matrice diagonale à partir de la diagonale de la matrice en argument.

Ainsi le membre de droite de (3.13) peut encore être approché par

$$\mathbf{T}_i + \mathbf{Z}\mathbf{T}_i + \mathbf{T}_i\mathbf{Z}^\dagger \approx \mathbf{T}_i^D + \mathbf{T}_i^Z + \mathbf{Z}\mathbf{T}_i^D + \mathbf{T}_i^D\mathbf{Z}^\dagger \quad (3.16)$$

conduisant à une deuxième approximation  $\mathcal{J}_{a_2}(\mathbf{Z})$  du critère  $\mathcal{J}(\mathbf{Z})$  déduite de (3.14) et donnée par

$$\mathcal{J}_{a_2}(\mathbf{Z}) = \sum_{i=1}^N \|\text{ZDiag}\{\mathbf{T}_i^Z + \mathbf{Z}\mathbf{T}_i^D + \mathbf{T}_i^D\mathbf{Z}^\dagger\}\|^2 \quad (3.17)$$

Comme les deux approximations précédentes ont la même expression, on peut les réécrire sous la forme générique suivante

$$\mathcal{J}_a(\mathbf{Z}) = \sum_{i=1}^N \|\text{ZDiag}\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^\dagger\}\|^2 \quad (3.18)$$

où  $\mathbf{T}_i^{(1)} = \mathbf{T}_i^{(2)} = \mathbf{T}_i$  quand  $\mathcal{J}_{a_1}(\mathbf{Z})$  est considéré et où  $\mathbf{T}_i^{(1)} = \mathbf{T}_i^Z$  et  $\mathbf{T}_i^{(2)} = \mathbf{T}_i^D$  quand  $\mathcal{J}_{a_2}(\mathbf{Z})$  est considéré.

Une dernière approximation peut être envisagée. En effet en utilisant le fait que  $\|\text{ZDiag}\{\cdot\}\|^2 = \text{tr}\{(\cdot)^H \text{ZDiag}\{\cdot\}\}$ , où  $\text{tr}\{\cdot\}$  est la trace de la matrice en argument, on a

$$\mathcal{J}_a(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^\dagger \right)^H \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^\dagger \right\} \right\} \quad (3.19)$$

et en considérant de nouveau que  $\|\mathbf{Z}\| \ll 1$ , une approximation supplémentaire  $\mathcal{J}_b(\mathbf{Z})$  de  $\mathcal{J}_a(\mathbf{Z})$  est donnée par

$$\mathcal{J}_b(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^\dagger \right)^H \mathbf{T}_i^Z + \mathbf{T}_i^{(1)H} \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^\dagger \right\} \right\} \quad (3.20)$$

car on a toujours  $\text{ZDiag}\{\mathbf{T}_i^{(1)}\} = \mathbf{T}_i^Z$ .

De manière explicite, lorsque que l'on posera  $\mathbf{T}_i^{(1)} = \mathbf{T}_i^{(2)} = \mathbf{T}_i$  le critère précédent sera noté  $\mathcal{J}_{b_1}(\mathbf{Z})$  et lorsque l'on posera  $\mathbf{T}_i^{(1)} = \mathbf{T}_i^Z$  et  $\mathbf{T}_i^{(2)} = \mathbf{T}_i^D$  il sera noté  $\mathcal{J}_{b_2}(\mathbf{Z})$ .

### 3.3.2 Cas hermitien

On considère les deux critères  $\mathcal{J}_a(\mathbf{Z})$  et  $\mathcal{J}_b(\mathbf{Z})$  du paragraphe précédent dans seul cas hermitien. On les notera alors  $\mathcal{J}_{a,h}(\mathbf{Z})$  et  $\mathcal{J}_{b,h}(\mathbf{Z})$  et ont les expressions

$$\mathcal{J}_{a,h}(\mathbf{Z}) = \sum_{i=1}^N \|\text{ZDiag}\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H\}\|^2 \quad (3.21)$$

et

$$\mathcal{J}_{b,h}(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \right)^H \mathbf{T}_i^Z + \mathbf{T}_i^{(1)H} \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \right\} \right\} \quad (3.22)$$

Le développement du calcul du gradient de (3.21) est effectué en Annexe A.2.1 et donne :

$$\begin{aligned} \frac{\partial \mathcal{J}_{a,h}(\mathbf{Z})}{\partial \mathbf{Z}^*} &= \sum_{i=1}^N \left[ \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\}^H \mathbf{T}_i^{(2)} + \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\} \mathbf{T}_i^{(2)H} \right. \\ &\quad + \text{ZDiag} \left\{ \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \right\} \mathbf{T}_i^{(2)H} \\ &\quad \left. + \text{ZDiag} \left\{ \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \right\}^H \mathbf{T}_i^{(2)} \right] \end{aligned} \quad (3.23)$$

tandis que l'on obtient le gradient suivant pour (3.22) (cf explication en Annexe A.2.1) :

$$\frac{\partial \mathcal{J}_{b,h}(\mathbf{Z})}{\partial \mathbf{Z}^*} = \sum_{i=1}^N \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\}^H \mathbf{T}_i^{(2)} + \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\} \mathbf{T}_i^{(2)H} \quad (3.24)$$

### 3.3.3 Cas symétrique

On considère les deux critères  $\mathcal{J}_a(\mathbf{Z})$  et  $\mathcal{J}_b(\mathbf{Z})$  du paragraphe précédent dans seul cas symétrique. On les notera alors  $\mathcal{J}_{a,s}(\mathbf{Z})$  et  $\mathcal{J}_{b,s}(\mathbf{Z})$  et ont les expressions

$$\mathcal{J}_{a,s}(\mathbf{Z}) = \sum_{i=1}^N \left\| \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T \right\} \right\|^2 \quad (3.25)$$

et

$$\mathcal{J}_{b,s}(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T \right)^H \mathbf{T}_i^Z + \mathbf{T}_i^{(1)H} \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T \right\} \right\} \quad (3.26)$$

Le calcul des dérivées des critères  $\mathcal{J}_{a,s}(\mathbf{Z})$  et  $\mathcal{J}_{b,s}(\mathbf{Z})$  par rapport à  $\mathbf{Z}^*$  est effectué en Annexe A.2.2, et donne :

$$\begin{aligned} \frac{\partial \mathcal{J}_{a,s}(\mathbf{Z})}{\partial \mathbf{Z}^*} &= \sum_{i=1}^N \left\{ \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\} \mathbf{T}_i^{(2)H} + \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\}^T \mathbf{T}_i^{(2)*} \right. \\ &\quad + \text{ZDiag} \left\{ \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T \right\} \mathbf{T}_i^{(2)H} \\ &\quad \left. + \text{ZDiag} \left\{ \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T \right\}^T \mathbf{T}_i^{(2)*} \right\} \end{aligned} \quad (3.27)$$

$$\frac{\partial \mathcal{J}_{b,s}(\mathbf{Z})}{\partial \mathbf{Z}^*} = \sum_{i=1}^N \left\{ \mathbf{Z} \text{Diag} \left\{ \mathbf{T}_i^{(1)} \right\} \mathbf{T}_i^{(2)H} + \mathbf{Z} \text{Diag} \left\{ \mathbf{T}_i^{(1)} \right\}^T \mathbf{T}_i^{(2)*} \right\} \quad (3.28)$$

### 3.3.4 Pas optimal

En introduisant (3.3) dans (3.18), on obtient un polynôme de degré 2 :

$$\mathcal{J}_a(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \{ b_{i,0} + \mu b_{i,1} + \mu^2 b_{i,2} \} \quad (3.29)$$

Et lorsqu'on remplace  $\mathbf{Z}$  par la formule (3.3) dans (3.20), on obtient alors un polynôme de degré 1 :

$$\mathcal{J}_b(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \{ b_{i,0} + \mu b_{i,1} \} \quad (3.30)$$

et dont les coefficients sont développés dans l'Annexe (B.2).

## 3.4 Description des algorithmes

Les pseudocodes ci-dessous décrivent les algorithmes issus des précédents calculs et auront pour dénomination DESERT, pour "Diagonalization of complEx SEts using a Relative gradient". Le cas hermitien est décrit dans l'Algorithme 6 tandis que le cas symétrique est décrit dans l'Algorithme 7.

### Algorithme 6: HDESERT

Initialisation : Ensemble de matrices  $\mathbf{M}_i$ , matrices initiales  $\mathbf{B}^{(0)}$   
**for**  $k = 1, 2, \dots$  jusqu'à convergence **do**  
  Mise à jour des matrices  $\mathbf{T}_i$  suivant (3.6)  
  Calcul du gradient (3.8), (3.23) ou (3.24)  
  Mise à jour de la matrices  $\mathbf{Z}$  avec (3.3) puis de  $\mathbf{B}$  avec (3.2)  
**end for**

### Algorithme 7: SDESERT

Initialisation : Ensemble de matrices  $\mathbf{M}_i$ , matrices initiales  $\mathbf{B}^{(0)}$   
**for**  $k = 1, 2, \dots$  jusqu'à convergence **do**  
  Mise à jour des matrices  $\mathbf{T}_i$  suivant (3.6)  
  Calcul du gradient (3.10), (3.27) ou (3.28)  
  Mise à jour de la matrices  $\mathbf{Z}$  avec (3.3) puis de  $\mathbf{B}$  avec (3.2)  
**end for**

La vitesse de convergence de ces deux algorithmes peut être améliorées en calculant de manière optimale le pas (cf l'Algorithme 5) avec les polynômes (3.11), (3.29) ou (3.30).

### 3.5 Simulations

Dans cette partie, nous allons présenter le comportement des algorithmes développés. Les comparaisons avec les 3 algorithmes de référence seront faites ultérieurement. La matrice de mélange est complexe de dimension  $N_o \times N_s$ . Celle-ci est générée aléatoirement en suivant une loi uniforme entre -2 et 2, autrement notée  $\mathcal{U}(-2, 2)$ . Les matrices diagonales sont construites ainsi : la partie réelle suit la loi  $\mathcal{U}(-1, 2)$  et la partie imaginaire  $\mathcal{U}(-2, 1)$ . Pour tester la robustesse de ce type de méthodes, nous rajoutons du bruit  $\mathbf{B}$ , dont la partie réelle et imaginaire suivent une loi gaussienne de moyenne nulle et de variance  $\sigma_b^2$ , de la manière suivante :

$$\mathbf{M}_i = \mathbf{A} \mathbf{D}_i \mathbf{A}^\dagger + \mathbf{B}$$

Enfin, nous considérons des ensembles composés de  $N$  matrices.

Dans les figures ci-dessous, nous affichons les résultats obtenus avec les déclinaisons des algorithmes 6 et 7, en fonction du gradient et du pas optimal considérés, mais aussi en fonction du critère lorsque l'influence des approximations est illustré. La dénomination standard des méthodes est la suivante :

$$\text{XDESERT}_i \text{OS}_j \mathcal{J}_l$$

avec :

- X représentant le fait d'utiliser un ensemble hermitien (H) ou symétrique (S).
- $i$  correspond au degré d'approximation.  $i = 1$  signifie que l'on utilise le gradient exact (critère  $\mathcal{J}$ ),  $i = 2$  le gradient avec l'approximation d'ordre un en  $\mathbf{Z}$  (critère  $\mathcal{J}_a$ ) et  $i = 3$  pour l'approximation d'ordre deux (critère  $\mathcal{J}_b$ ).
- $j$  permet de déterminer quel polynôme est utilisé pour déterminer le pas optimal.  $j = 1$  représente le polynôme de degré quatre,  $j = 2$  celui de degré deux.
- $l$  indique quel type de critère est utilisé ;  $l = 1$  pour  $\mathcal{J}_{a1}$  ou  $\mathcal{J}_{b1}$  et  $l = 2$  pour  $\mathcal{J}_{a2}$  ou  $\mathcal{J}_{b2}$ .

Les résultats ont été moyennés sur 100 réalisations de Monte Carlo. De plus, la définition en diagonalisation conjointe du RSB est la suivante :

$$\text{RSB}_{\text{dB}} = 10 \log_{10} \frac{1}{\sigma_b^2} \quad (3.31)$$

Enfin, pour évaluer l'estimation de la matrice recherchée, nous utiliserons l'indice de perfor-

mance suivant [57, 58] :

$$I(\mathbf{G}) = \frac{1}{N_s(N_s - 1)} \left\{ \sum_{i=1}^{N_s} \left( \sum_{j=1}^{N_s} \frac{|G_{i,j}|^2}{\max_l |G_{i,l}|^2} - 1 \right) + \sum_{j=1}^{N_s} \left( \sum_{i=1}^{N_s} \frac{|G_{i,j}|^2}{\max_l |G_{l,j}|^2} - 1 \right) \right\} \quad (3.32)$$

avec  $\mathbf{G} = \mathbf{BA}$ ,  $\mathbf{A}$  étant la matrice recherchée et  $\mathbf{B}$  est l'inverse de  $\mathbf{A}$  (ou la pseudo-inverse, en fonction de la dimension).

Ens.	Nbr de mat.	RSB (dB)	Nbr it	$N_o$	$N_s$	Figure
Herm.	10	20	1 :150	4	4	3.1
Herm.	10	30	1 :200	5	3	3.2
Herm.	10	10	1 :250	4	4	3.3
Herm.	10	10	1 :100	5	3	3.4
Sym.	10	20	1 :350	4	4	3.5
Sym.	10	30	1 :350	5	3	3.6
Herm. et Sym.	10 + 10	30	1 :250	4	4	3.7
Herm. et Sym.	10 + 10	-10 :30	350	4	4	3.8
Herm. et Sym.	10 + 10	30	1 :300	5	3	3.9
Herm. et Sym.	10 + 10	-10 :30	350	5	3	3.10
Herm. et Sym.	10 + 10	10	1 :150	4	4	3.11
Herm. et Sym.	10 + 10	10	1 :150	5	3	3.12

TABLE 3.1 – Paramètres de simulations des algorithmes DESERT

### 3.5.1 Cas hermitien

À la vue des Figure 3.1 et Figure 3.2, on remarque que lorsqu'on utilise l'approximation à l'ordre deux, la vitesse de convergence est améliorée sans altérer les performances. On peut aussi déduire que la recherche du pas optimal implique des compromis. En effet, l'utilisation d'un polynôme de degré deux est moins complexe à mettre en oeuvre (et donc moins gourmand en ressources informatiques), mais mettra beaucoup plus de temps à converger.

Les deux résultats présentés en Figure 3.3 et Figure 3.4 permettent d'arriver à la même conclusion, en mettant en lumière l'influence du choix du critère : l'utilisation du critère (3.17) améliore la vitesse de convergence de ce type d'algorithme.

### 3.5.2 Cas symétrique

Les Figures 3.5 et 3.6 confirment les résultats entrevus dans le cas hermitien, et prouvent que ce type de méthode permet d'estimer correctement la matrice de séparation, en considérant un ensemble composé uniquement de matrices symétriques. Cependant, on peut noter au niveau

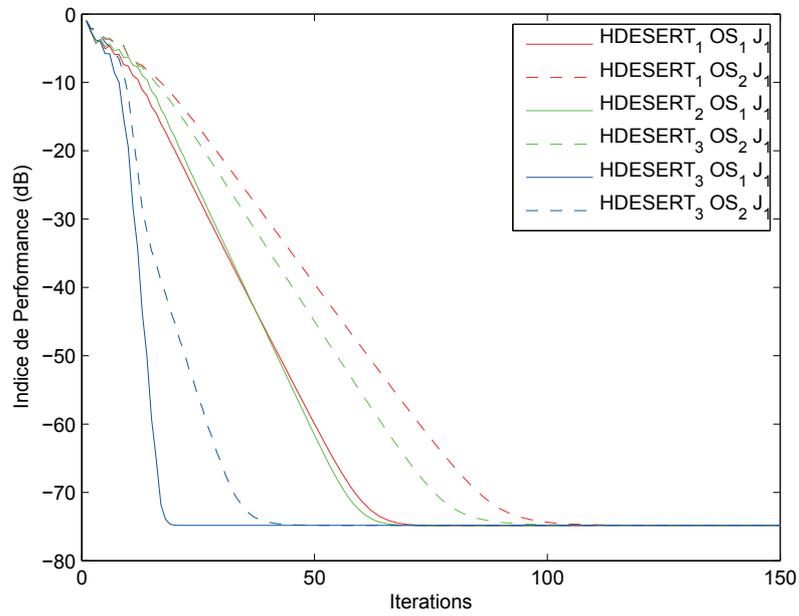


FIGURE 3.1 – Vitesse de convergence des algorithmes HDESERT.

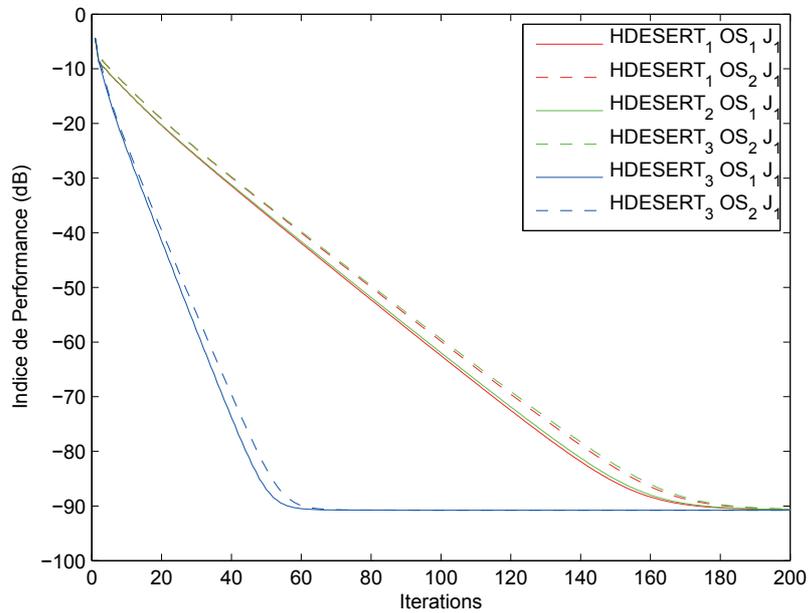


FIGURE 3.2 – Vitesse de convergence des algorithmes HDESERT.

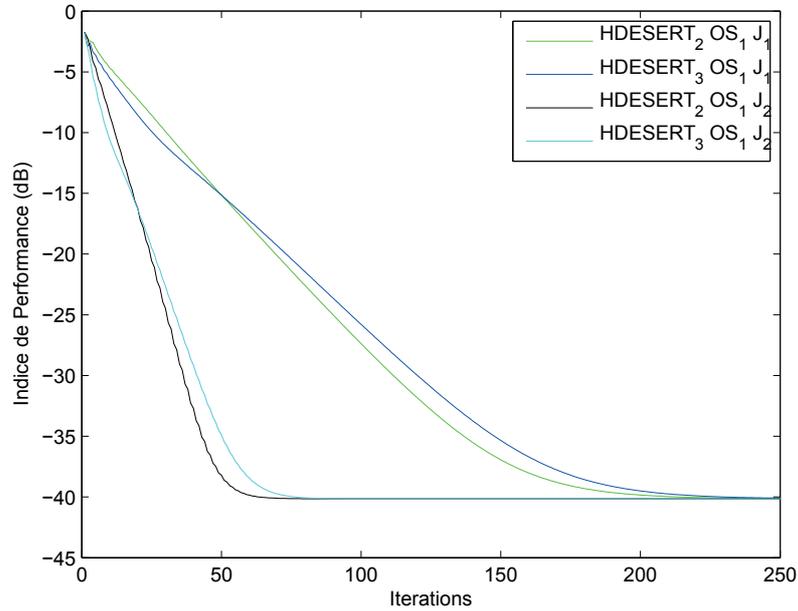


FIGURE 3.3 – Vitesse de convergence des algorithmes HDESERT. Comparaison en fonction du critère.

de la Figure 3.5 que si on utilise le polynôme à l'ordre deux dans la recherche du pas optimal dans le cas carré, on améliore significativement la vitesse de convergence. Ceci n'est alors plus forcément vrai dans le cas où la matrice de mélange est rectangulaire.

### 3.5.3 Combinaison des deux ensembles

La Figure 3.7 montre le comportement des algorithmes DESERT par rapport à HDESERT. On remarque clairement que l'utilisation de deux ensembles de matrices améliore significativement les performances d'estimation de la matrice de mélange, par rapport à la prise en compte d'un seul ensemble. La Figure 3.8 apporte une information supplémentaire ; ces performances sont meilleures, et ce quelque soit le niveau de bruit considéré. Il semble également que l'utilisation combinée de deux ensembles améliore la vitesse de convergence.

La recherche d'une matrice de mélange rectangulaire est intéressante en plusieurs points, et son illustration se fait grâce à la Figure 3.9 et la Figure 3.10. En effet, on remarque qu'on améliore non seulement les performances mais aussi la vitesse de convergence en combinant les deux ensembles de matrices, ce qui vient conforter les observations réalisées dans le cas carré.

La Figure 3.11 et la Figure 3.12 permettent d'illustrer l'influence du critère approximé considéré. On remarque alors que si on utilise le critère  $\mathcal{J}_{b2}(\mathbf{Z})$ , on arrive à améliorer la vitesse de

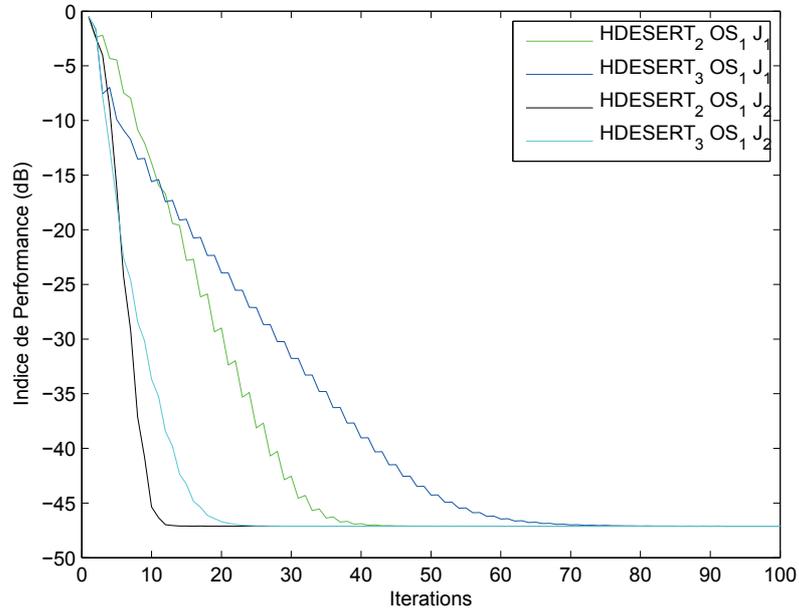


FIGURE 3.4 – Vitesse de convergence des algorithmes HDESERT. Comparaison en fonction du critère.

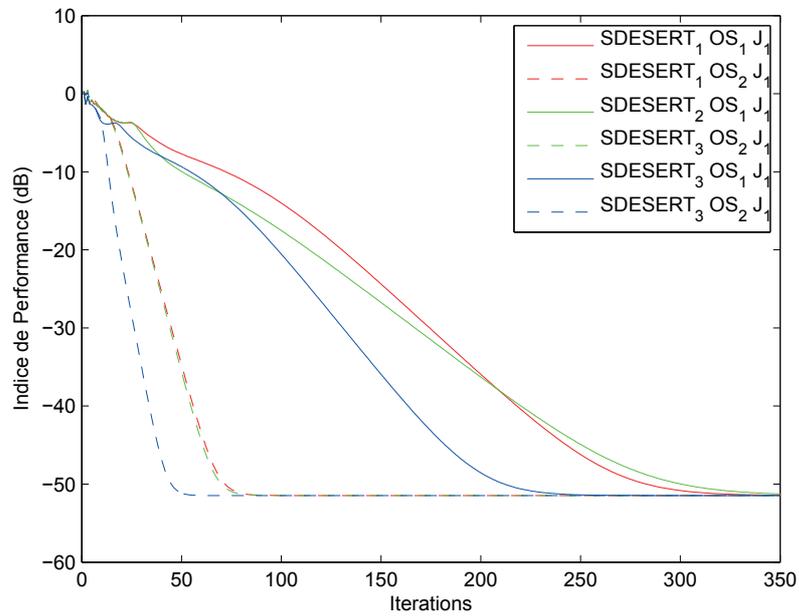


FIGURE 3.5 – Vitesse de convergence des algorithmes SDESERT.

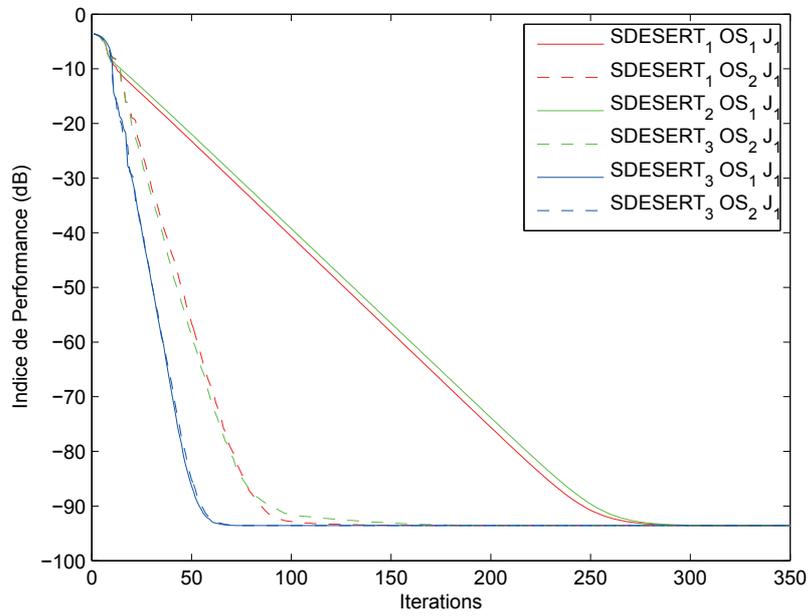


FIGURE 3.6 – Vitesse de convergence des algorithmes SDESERT.

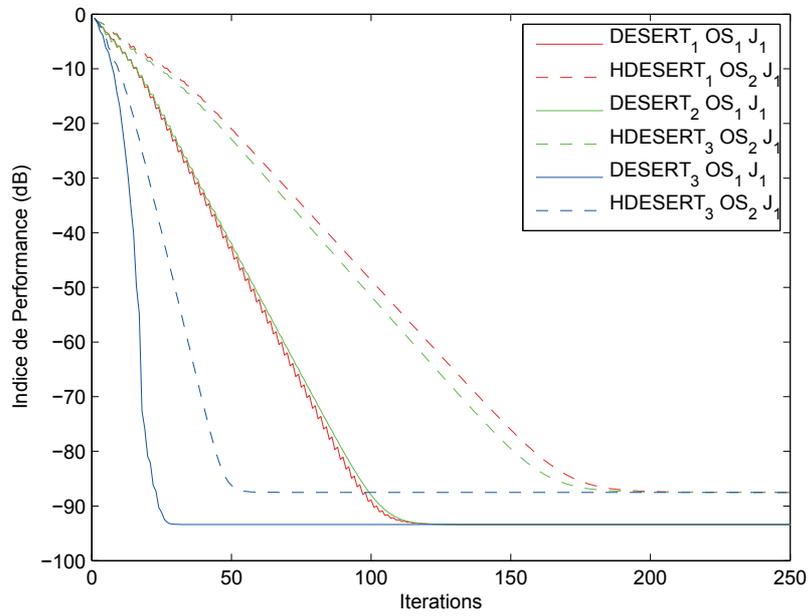


FIGURE 3.7 – Vitesse de convergence des algorithmes DESERT.

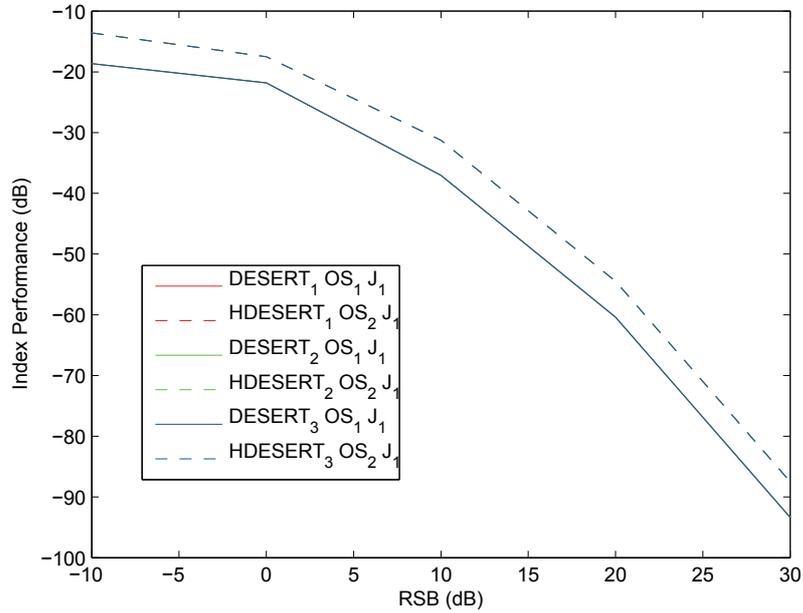


FIGURE 3.8 – Performance des algorithmes DESERT.

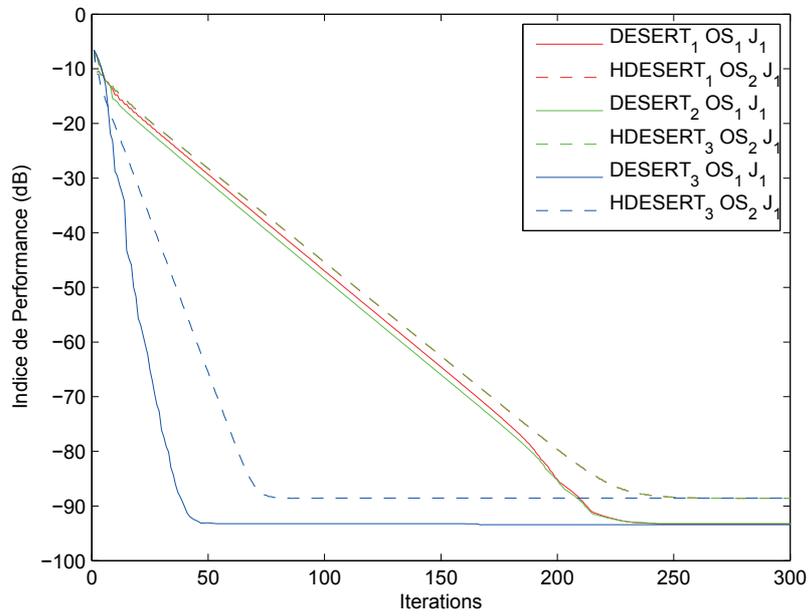


FIGURE 3.9 – Vitesse de convergence des algorithmes DESERT.

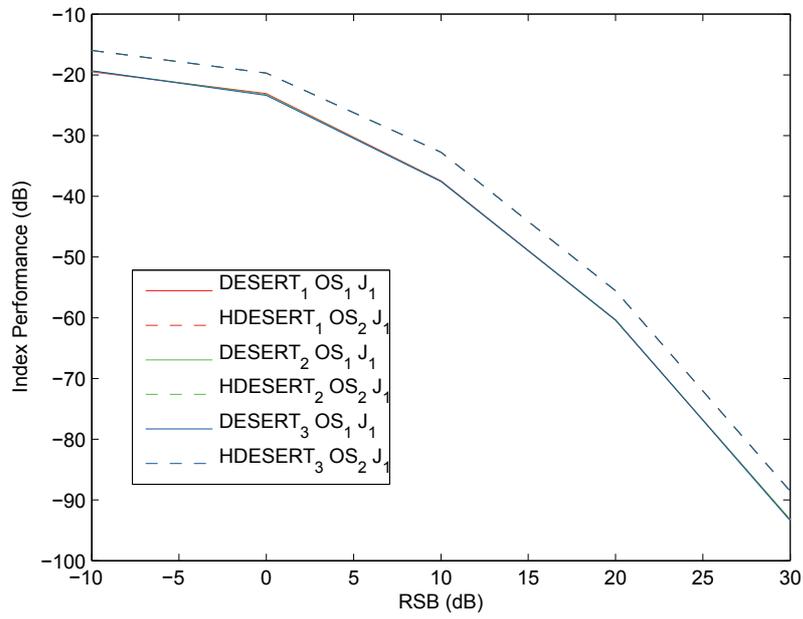


FIGURE 3.10 – Performance des algorithmes DESERT.

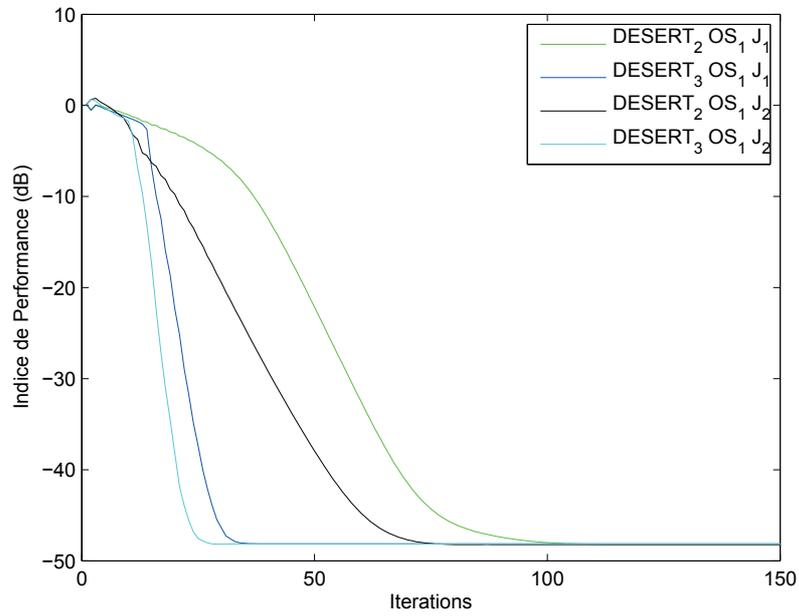


FIGURE 3.11 – Vitesse de convergence des algorithmes DESERT. Comparaison en fonction du critère.

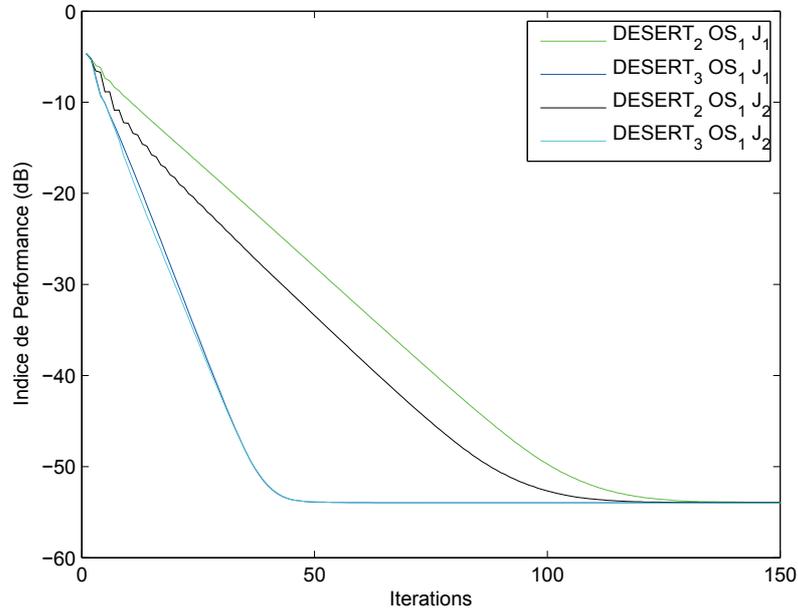


FIGURE 3.12 – Vitesse de convergence des algorithmes DESERT. Comparaison en fonction du critère.

convergence. Ceci est vraiment significatif dans le cas carré.

### 3.5.4 Complexité de calcul

Le tableau 3.2 présente les résultats que l'on obtient en termes de temps d'exécution, dans le cas où la matrice de mélange est rectangulaire. Cela permet de voir l'influence des différentes approximations réalisées. On remarque clairement que le fait d'utiliser un polynôme d'ordre deux pour la recherche du pas optimal (au lieu de celui d'ordre quatre) améliore significativement le temps de calcul. On peut aussi remarquer que, bien que les méthodes utilisant l'approximation la plus forte sur le critère convergent le plus rapidement, elles sont aussi plus chronophages. Ainsi, on peut donc en conclure que le choix de la méthode est directement influencé par les contraintes imposées par l'utilisateur, à savoir utiliser une méthode qui converge rapidement, ou une méthode qui n'utilise que peu de ressources informatiques.

## 3.6 Conclusions sur les méthodes de type gradient

Dans ce paragraphe, nous avons illustré les résultats obtenus avec les algorithmes de type gradient que l'on a développé. On montre alors que ceux-ci permettent d'estimer correctement

Méthode	cas herm. (ms)	cas sym.. (ms)	cas herm. + sym. (ms)
DESERT <sub>1</sub> OS <sub>1</sub> J <sub>1</sub>	2.54	2.30	4.18
DESERT <sub>1</sub> OS <sub>2</sub> J <sub>1</sub>	1.58	1.58	2.96
DESERT <sub>2</sub> OS <sub>1</sub> J <sub>1</sub>	1.91	1.97	3.79
DESERT <sub>2</sub> OS <sub>1</sub> J <sub>2</sub>	1.81	2.05	3.86
DESERT <sub>2</sub> OS <sub>2</sub> J <sub>1</sub>	1.30	1.48	2.74
DESERT <sub>2</sub> OS <sub>2</sub> J <sub>2</sub>	1.43	1.54	2.90
DESERT <sub>3</sub> OS <sub>1</sub> J <sub>1</sub>	1.66	1.94	3.68
DESERT <sub>3</sub> OS <sub>1</sub> J <sub>2</sub>	1.81	1.97	3.75
DESERT <sub>3</sub> OS <sub>2</sub> J <sub>1</sub>	1.32	1.40	2.62
DESERT <sub>3</sub> OS <sub>2</sub> J <sub>2</sub>	1.34	1.51	2.78

TABLE 3.2 – Temps d'exécution des algorithmes DESERT

la matrice de séparation, et ce malgré la présence de bruit. De plus, nos méthodes permettent de considérer le cas déterminé ou sur-déterminé, dans la mesure où le nombre d'observations est systématiquement supérieur ou égal au nombre de sources, que l'on suppose connu.

La prise en compte d'une diversité statistique permet également d'améliorer de manière significative les résultats obtenus classiquement, c'est-à-dire lorsqu'on ne considère que le seul ensemble de matrices hermitiennes. Enfin, Les algorithmes que nous avons développés permettent aux utilisateurs de choisir différents niveaux de complexité. En effet, nous montrons que les approximations ne dégradent pas la vitesse de convergence, tout en maintenant les mêmes valeurs de performances.

## Chapitre 4

# Algorithmes de mise à jour analytique de la matrice

Dans ce chapitre, deux algorithmes vont être présentés. L'un d'eux utilisera le critère direct, tandis que l'autre sera basé sur le critère inverse. Tous les deux auront cependant une caractéristique commune, en l'occurrence celle de déterminer analytiquement la matrice recherchée.

### 4.1 Algorithme de mise à jour par couple de termes

#### 4.1.1 Description générale

Dans ce paragraphe, nous présentons une nouvelle approche permettant de minimiser algébriquement le critère  $\mathcal{J}_{a_2}$  en (3.17)

$$\mathcal{J}_{a_2}(\mathbf{Z}) = \sum_{i=1}^N \|\mathbf{Z} \text{Diag}\{\mathbf{T}_i^Z + \mathbf{Z} \mathbf{T}_i^D + \mathbf{T}_i^D \mathbf{Z}^\dagger\}\|^2 \quad (4.1)$$

Nous développons maintenant l'équation précédente. On obtient, en notant

$$(\mathbf{T}_i^Z)_{mn} = T_{i,mn}^Z$$

$$(\mathbf{Z})_{mn} = Z_{mn}$$

$$(\mathbf{T}_i^D)_{nn} = T_{i,n}^D$$

$$\begin{aligned}
\mathcal{J}_{a_2}(\mathbf{Z}) &= \sum_{i=1}^N \sum_{\substack{m,n=1 \\ m \neq n}}^{N_s} |Z_{mn}T_{i,n}^D + Z_{nm}^\dagger T_{i,m}^D + T_{i,mn}^Z|^2 \\
&= \sum_{\substack{m,n=1 \\ m < n}}^{N_s} \sum_{i=1}^N |Z_{mn}T_{i,n}^D + Z_{nm}^\dagger T_{i,m}^D + T_{i,mn}^Z|^2 + |Z_{nm}T_{i,m}^D + Z_{mn}^\dagger T_{i,n}^D + T_{i,nm}^Z|^2 \\
&= \sum_{\substack{m,n=1 \\ m < n}}^{N_s} \mathcal{I}(Z_{mn}, Z_{nm})
\end{aligned} \tag{4.2}$$

avec

$$\mathcal{I}(Z_{mn}, Z_{nm}) = \sum_{i=1}^N |Z_{mn}T_{i,n}^D + Z_{nm}^\dagger T_{i,m}^D + T_{i,mn}^Z|^2 + |Z_{nm}T_{i,m}^D + Z_{mn}^\dagger T_{i,n}^D + T_{i,nm}^Z|^2 \tag{4.3}$$

Il apparaît alors clairement que chaque terme de la double somme (4.2) ne dépend que de  $Z_{mn}$  et  $Z_{nm}$ . La minimisation de  $\mathcal{J}_{a_2}(\mathbf{Z})$  correspond alors à la minimisation de chacun de ces termes. Ainsi, le problème initialement de dimension  $N_s(N_s - 1)$  (et qui concorde avec le nombre d'inconnues de  $\mathbf{Z}$ ) est réduit à  $N_s(N_s - 1)/2$  problèmes de dimension deux, où chaque couple  $(Z_{mn}, Z_{nm})$  doit être estimé conjointement.

Ainsi, dans la suite, on ne considèrera plus que le problème de dimension deux qui correspond à la minimisation de  $\mathcal{I}(Z_{mn}, Z_{nm})$  en (4.3).

#### 4.1.2 Calcul de la matrice de mise à jour $\mathbf{Z}$

En partant des résultats obtenus dans la partie ci-dessus, les éléments non-diagonaux de  $\mathbf{Z}$  sont estimés par paire. Il s'agit donc de calculer  $(Z_{mn}, Z_{nm})$  pour chaque valeur de  $m$  et  $n$  (avec  $m < n$ ) minimisant  $\mathcal{I}$  dans (4.3).

Nous allons généraliser l'approche de [77] dans les deux cas complexes. Le fait de considérer des variables complexes entraîne certaines spécificités (cas de variables conjuguées ou non), et nous devons ainsi distinguer le cas hermitien et le cas symétrique. Nous allons d'abord considérer le cas hermitien, puis ensuite le cas symétrique.

##### 4.1.2.1 Cas hermitien

Revenons tout d'abord au mélange hermitien. On considère ainsi le critère (4.3) en remplaçant l'opérateur  $\dagger$  par  $H$  pour des matrices. Pour un scalaire, cela correspond à l'opérateur de conjugaison, noté  $(\cdot)^*$  dans la suite.

Dans ce cas, le critère du problème de dimension deux, noté  $\mathcal{I}^{(h)}$ ; s'écrit de la manière

suivante :

$$\mathcal{I}^{(h)}(Z_{mn}, Z_{nm}) = \sum_{i=1}^N |Z_{mn}T_{i,n}^D + Z_{nm}^*T_{i,m}^D + T_{i,mn}^Z|^2 + |Z_{nm}T_{i,m}^D + Z_{mn}^*T_{i,n}^D + T_{i,nm}^Z|^2 \quad (4.4)$$

Calculons alors la dérivée de (4.4), qui sera utile pour la minimisation, par rapport à  $Z_{mn}^*$  puis  $Z_{nm}^*$  :

$$\begin{aligned} \frac{\partial \mathcal{I}^{(h)}}{\partial Z_{mn}^*} &= \sum_{i=1}^N (Z_{mn}T_{i,n}^D + Z_{nm}^*T_{i,m}^D + T_{i,mn}^Z)T_{i,n}^{D*} + (Z_{nm}^*T_{i,m}^{D*} + Z_{mn}T_{i,n}^{D*} + T_{i,nm}^Z)T_{i,n}^D \\ &= H_{11}Z_{mn} + H_{12}Z_{nm}^* - e_1^{(h)} \end{aligned} \quad (4.5)$$

avec

$$H_{11} = 2 \sum_{i=1}^N |T_{i,n}^D|^2 \quad (4.6)$$

$$H_{12} = \sum_{i=1}^N T_{i,n}^{D*}T_{i,m}^D + T_{i,n}^DT_{i,m}^{D*} \quad (4.7)$$

$$e_1^{(h)} = - \sum_{i=1}^N T_{i,mn}^ZT_{i,n}^{D*} + T_{i,nm}^ZT_{i,n}^D \quad (4.8)$$

La deuxième dérivée donne :

$$\begin{aligned} \frac{\partial \mathcal{I}^{(h)}}{\partial Z_{nm}^*} &= \sum_{i=1}^N (Z_{nm}T_{i,m}^D + Z_{mn}^*T_{i,n}^D + T_{i,nm}^Z)T_{i,m}^{D*} + (Z_{mn}^*T_{i,n}^{D*} + Z_{nm}T_{i,m}^{D*} + T_{i,mn}^Z)T_{i,m}^D \\ &= H_{21}Z_{mn}^* + H_{22}Z_{nm} - e_2^{(h)*} \end{aligned} \quad (4.9)$$

avec

$$H_{21} = \sum_{i=1}^N T_{i,n}^{D*}T_{i,m}^D + T_{i,n}^DT_{i,m}^{D*} \quad (4.10)$$

$$H_{22} = 2 \sum_{i=1}^N |T_{i,m}^D|^2 \quad (4.11)$$

$$e_2^{(h)*} = - \sum_{i=1}^N T_{i,nm}^ZT_{i,m}^{D*} + T_{i,mn}^ZT_{i,m}^D \quad (4.12)$$

En regardant les deux dérivées calculées en (4.5) et (4.9), les variables sont conjuguées. On va

donc considérer dans la suite le conjugué de la seconde expression afin de résoudre le système :

$$\begin{cases} \frac{\partial \mathcal{I}^{(h)}}{\partial Z_{mn}^*} = 0 \\ \frac{\partial \mathcal{I}^{(h)}}{\partial Z_{nm}} = \left( \frac{\partial \mathcal{I}^{(h)}}{\partial Z_{nm}^*} \right)^* = 0 \end{cases} \quad (4.13)$$

qui peut être réécrit plus simplement :

$$\mathbf{H} \mathbf{z}_{m,n}^{(h)} = \mathbf{e}^{(h)} \quad (4.14)$$

avec

$$\mathbf{H} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}, \quad \mathbf{e}^{(h)} = \begin{pmatrix} e_1^{(h)} \\ e_2^{(h)} \end{pmatrix} \quad (4.15)$$

et

$$\mathbf{z}_{m,n}^{(h)} = \begin{pmatrix} Z_{mn} \\ Z_{nm}^* \end{pmatrix} \quad (4.16)$$

Lorsque la matrice  $\mathbf{H}$  est inversible, les solutions du système ci-dessus peuvent être calculées analytiquement :

$$\mathbf{z}_{m,n}^{(h)} = \mathbf{H}^{-1} \mathbf{e}^{(h)} \quad (4.17)$$

#### 4.1.2.2 Cas symétrique

Considérons maintenant le second cas où les matrices ne sont plus hermitiennes mais symétriques. L'opérateur  $\ddagger$  correspond alors au transposé dans le cas matriciel, mais qui n'a pas d'influence lorsque des scalaires sont considérés. Dans ce cas, le critère est désormais noté  $\mathcal{I}^{(t)}$  et s'écrit :

$$\mathcal{I}^{(t)}(Z_{mn}, Z_{nm}) = 2 \sum_{i=1}^N |Z_{mn} T_{i,n}^D + Z_{nm} T_{i,m}^D + T_{i,mn}^Z|^2 \quad (4.18)$$

Le calcul de la dérivée de  $\mathcal{I}^{(t)}$  par rapport à  $Z_{mn}^*$  et  $Z_{nm}^*$  donne :

$$\begin{aligned} \frac{\partial \mathcal{I}^{(t)}}{\partial Z_{mn}^*} &= 2 \sum_{i=1}^N (Z_{mn} T_{i,n}^D + Z_{nm} T_{i,m}^D + T_{i,mn}^Z) T_{i,n}^{D*} \\ &= 2(S_{11} Z_{mn} + S_{12} Z_{nm} - e_1^{(t)}) \end{aligned} \quad (4.19)$$

avec

$$S_{11} = \sum_{i=1}^N |T_{i,n}^D|^2 \quad (4.20)$$

$$S_{12} = \sum_{i=1}^N T_{i,n}^{D*} T_{i,m}^D \quad (4.21)$$

$$e_1^{(t)} = - \sum_{i=1}^N T_{i,mn}^Z T_{i,n}^D \quad (4.22)$$

Le calcul de la seconde dérivée donne :

$$\begin{aligned} \frac{\partial \mathcal{I}^{(t)}}{\partial Z_{nm}^*} &= 2 \sum_{i=1}^N (Z_{mn} T_{i,n}^D + Z_{nm} T_{i,m}^D + T_{i,mn}^Z) T_{i,m}^{D*} \\ &= 2(S_{21} Z_{mn} + S_{22} Z_{nm} - e_2^{(t)}) \end{aligned} \quad (4.23)$$

avec

$$S_{21} = \sum_{i=1}^N T_{i,n}^D T_{i,m}^{D*} \quad (4.24)$$

$$S_{22} = \sum_{i=1}^N |T_{i,m}^D|^2 \quad (4.25)$$

$$e_2^{(t)} = - \sum_{i=1}^N T_{i,mn}^Z T_{i,m}^{D*} \quad (4.26)$$

Le système à résoudre

$$\begin{cases} \frac{\partial \mathcal{I}^{(t)}}{\partial Z_{mn}^*} = 0 \\ \frac{\partial \mathcal{I}^{(t)}}{\partial Z_{nm}^*} = 0 \end{cases} \quad (4.27)$$

peut être simplement écrit comme :

$$\mathbf{S} \mathbf{z}_{m,n}^{(t)} = \mathbf{e}^{(t)} \quad (4.28)$$

avec

$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}, \quad \mathbf{e}^{(t)} = \begin{pmatrix} e_1^{(t)} \\ e_2^{(t)} \end{pmatrix} \quad (4.29)$$

et

$$\mathbf{z}_{m,n}^{(t)} = \begin{pmatrix} Z_{mn} \\ Z_{nm} \end{pmatrix} \quad (4.30)$$

Quand la matrice  $\mathbf{S}$  est inversible, les deux solutions du système ci-dessus sont obtenus directement

$$\mathbf{z}_{m,n}^{(t)} = \mathbf{S}^{-1} \mathbf{e}^{(t)} \quad (4.31)$$

#### 4.1.2.3 Combinaison des deux ensembles

Malheureusement, les deux systèmes (4.14) et (4.28) ne peuvent être combinés directement, puisque les deux vecteurs caractérisant les inconnus ne sont pas les mêmes. Pour cela, nous proposons de considérer les variables  $Z_{nm}$  et  $Z_{nm}^*$  comme indépendantes. Ainsi, nous introduisons le nouveau vecteur, composé des 3 inconnues  $Z_{mn}$  (similaire aux deux systèmes),  $Z_{nm}$  et  $Z_{nm}^*$

$$\mathbf{z}_{m,n} = \begin{pmatrix} Z_{mn} \\ Z_{nm}^* \\ Z_{nm} \end{pmatrix} \quad (4.32)$$

En définissant les deux nouvelles matrices suivantes

$$\mathbf{C} = \begin{pmatrix} H_{11} & H_{12} & 0 \\ H_{21} & H_{22} & 0 \\ S_{11} & 0 & S_{12} \\ S_{21} & 0 & S_{22} \end{pmatrix}, \quad \mathbf{e} = \begin{pmatrix} e_1^{(h)} \\ e_2^{(h)} \\ e_1^{(t)} \\ e_2^{(t)} \end{pmatrix} \quad (4.33)$$

nous pouvons combiner les deux systèmes développés en (4.14) et (4.28) :

$$\mathbf{C} \mathbf{z}_{m,n} = \mathbf{e} \quad (4.34)$$

qui sera résolu de la manière suivante :

$$\mathbf{z}_{m,n} = \mathbf{C}^\dagger \mathbf{e} \quad (4.35)$$

### 4.1.3 Description des algorithmes

L'algorithme explicité ci-dessus, dénommé NOODLES pour "NON Orthogonal Decomposition of complex Sets", est décrit dans le pseudocode Algorithme 8 :

Algorithme 8: NOODLES
Initialisation : Ensemble de matrices $\mathbf{M}_i$ , matrices initiales $\mathbf{B}^{(0)}$ <b>for</b> $k = 1, 2, \dots$ jusqu'à convergence <b>do</b> Mise à jour des matrices $\mathbf{T}_i$ suivant (3.6) Mise à jour de la matrices $\mathbf{Z}$ suivant (4.17) dans le cas hermitien, (4.31) dans le cas symétrique et (4.35) lors de la combinaison des deux ensembles <b>end for</b>

## 4.2 Algorithmes de moindres carrés alternés

### 4.2.1 Description générale

Dans cette partie, nous n'allons plus nous intéresser à l'estimation de la matrice de séparation comme cela a été fait précédemment. En effet, nous allons estimer directement la matrice de mélange, par minimisation du critère  $\mathcal{D}(\mathbf{A})$  suivant :

$$\mathcal{D}(\mathbf{A}, \{\mathbf{D}_i\}) = \sum_{i=1}^N \|\mathbf{M}_i - \mathbf{A}\mathbf{D}_i\mathbf{A}^\dagger\|^2 \quad (4.36)$$

Un point important de l'algorithme consiste à considérer que  $\mathbf{A}^\dagger$  est indépendant de  $\mathbf{A}$ . On notera cette matrice  $\mathbf{C}$ . On réécrit alors  $\mathcal{D}$  en (4.36) de la façon suivante :

$$\mathcal{D}(\mathbf{A}, \mathbf{C}, \{\mathbf{D}_i\}) = \sum_{i=1}^N \|\mathbf{M}_i - \mathbf{A}\mathbf{D}_i\mathbf{C}\|^2 \quad (4.37)$$

Le principe de la méthode qui va être développée est relativement simple, rendant ainsi son utilisation pratique dans plusieurs situations. On va s'inspirer de ce qui est fait en décomposition tensorielle [9, 61, 25]. Cependant, il peut arriver que ce type d'algorithme ait des problèmes de convergence. Une solution sera aussi proposée et utilisera une recherche linéaire. L'origine de ces travaux est accessible dans ces références [65, 25].

Elle consiste donc à estimer chacune des matrices en fixant les autres :

1. On fixe  $\mathbf{A}$  et  $\mathbf{C}$ , et on calcule alors  $\mathbf{D}_i$  pour tout  $i$ ,
2. On utilise l'estimation de  $\mathbf{D}_i$ , tout en gardant  $\mathbf{C}$  fixe pour calculer alors  $\mathbf{A}$ ,
3. En ayant estimé  $\mathbf{D}_i$  et  $\mathbf{A}$  que l'on fixera, on calcule finalement  $\mathbf{C}$ .

### 4.2.2 Mise à jour des matrices diagonales $\mathbf{D}_i$

Comme expliqué ci-dessus dans le schéma global de l'algorithme, nous allons tout d'abord estimer les matrices diagonales. Pour cela, on fixe les valeurs de  $\mathbf{A}$  et  $\mathbf{C}$ . En considérant (1.6) à l'ensemble des matrices  $\mathbf{M}_i$ , on a :

$$[\text{vec}\{\mathbf{M}_1\}, \dots, \text{vec}\{\mathbf{M}_N\}] = (\mathbf{C}^T \odot \mathbf{A}) [\text{diag}\{\mathbf{D}_1\}, \dots, \text{diag}\{\mathbf{D}_N\}] \quad (4.38)$$

On en déduit que :

$$[\mathbf{D}_1, \dots, \mathbf{D}_N] = \text{undia}\left\{(\mathbf{C}^T \otimes \mathbf{A})^\dagger [\text{vec}\{\mathbf{M}_1\}, \dots, \text{vec}\{\mathbf{M}_N\}]\right\} \quad (4.39)$$

où  $\text{undia}\{(\cdot)\}$  construit une matrice diagonale à partir du vecteur  $(\cdot)$ , de telle manière que

$$\text{undia}\{\text{diag}\{(\cdot)\}\} = (\cdot).$$

### 4.2.3 Mise à jour de la matrice de mélange $\mathbf{A}$

Comme nous disposons d'une première mise à jour des matrices diagonales, nous allons utiliser ce résultat pour déterminer une estimée de la matrice de mélange  $\mathbf{A}$ . En conservant  $\mathbf{C}$  comme dans la première partie, et en concaténant horizontalement les matrices  $\mathbf{M}_i$ , il vient :

$$[\mathbf{M}_1, \dots, \mathbf{M}_N] = \mathbf{A} [\mathbf{D}_1\mathbf{C}, \dots, \mathbf{D}_N\mathbf{C}] \quad (4.40)$$

Il apparaît alors clairement que la matrice de mélange est accessible avec une multiplication par pseudo-inverse :

$$\mathbf{A} = [\mathbf{M}_1, \dots, \mathbf{M}_N] [\mathbf{D}_1\mathbf{C}, \dots, \mathbf{D}_N\mathbf{C}]^\dagger \quad (4.41)$$

### 4.2.4 Mise à jour de $\mathbf{C}$

Comme nous disposons désormais d'une estimation des matrices diagonales et de la matrice de mélange, il ne nous reste plus qu'à chercher la dernière matrice encore inconnue. En concaténant verticalement les matrices de l'ensemble, on observe que :

$$\begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_N \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{D}_1 \\ \vdots \\ \mathbf{A}\mathbf{D}_N \end{bmatrix} \mathbf{C} \quad (4.42)$$

On remarque alors que  $\mathbf{C}$  peut alors être aisément déterminée :

$$\mathbf{C} = \begin{bmatrix} \mathbf{A}\mathbf{D}_1 \\ \vdots \\ \mathbf{A}\mathbf{D}_N \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_N \end{bmatrix} \quad (4.43)$$

Ainsi, nous avons réussi en trois étapes à déterminer toutes les inconnues de notre problème par résolution de systèmes linéaires au sens des moindres carrés. Ce processus est alors répété de manière itérative, afin d'améliorer les estimations. Cependant, ce type de méthode peut avoir un inconvénient important. La convergence peut être très lente. Ainsi, nous proposons, comme dans tous les algorithmes que nous avons présentés précédemment, d'effectuer une mise à jour en faisant une recherche linéaire optimisée (ELS).

#### 4.2.5 Recherche linéaire optimisée

Reprenons la fonction de coût définie en (4.37), et réécrivons là à l'itération  $n$  :

$$\mathcal{D}^{(n)}(\mathbf{A}, \mathbf{C}, \{\mathbf{D}_i\}) = \sum_{i=1}^N \|\mathbf{M}_i - \mathbf{A}^{(n)}\mathbf{D}_i^{(n)}\mathbf{C}^{(n)}\|^2 \quad (4.44)$$

Une mise à jour linéaire des matrices à l'itération  $n$  utilise les estimations de ces matrices aux instants précédents  $n - 2$  et  $n - 1$ , et s'écrit de la manière suivante :

$$\begin{aligned} \mathbf{A}^{(n)} &= \mathbf{A}^{(n-2)} + \mu \left( \mathbf{A}^{(n-1)} - \mathbf{A}^{(n-2)} \right) = \mathbf{A}^{(n-2)} + \mu \boldsymbol{\Theta}_A \\ \mathbf{C}^{(n)} &= \mathbf{C}^{(n-2)} + \mu \left( \mathbf{C}^{(n-1)} - \mathbf{C}^{(n-2)} \right) = \mathbf{C}^{(n-2)} + \mu \boldsymbol{\Theta}_C \\ \mathbf{D}_i^{(n)} &= \mathbf{D}_i^{(n-2)} + \mu \left( \mathbf{D}_i^{(n-1)} - \mathbf{D}_i^{(n-2)} \right) = \mathbf{D}_i^{(n-2)} + \mu \boldsymbol{\Theta}_{D_i} \end{aligned} \quad (4.45)$$

où  $\mu \in \mathbb{R}^{+*}$  représente le pas.

Le développement de  $\mathbf{A}^{(n)}\mathbf{D}_i^{(n)}\mathbf{C}^{(n)}$  dans (4.44), en utilisant les mises à jour introduites dans (4.45), permet d'obtenir un polynôme de degré six :

$$\mathcal{D}^{(n)}(\mathbf{A}, \mathbf{C}, \{\mathbf{D}_i\}) = \sum_{i=1}^N e_{i,0} + \mu e_{i,1} + \mu^2 e_{i,2} + \mu^3 e_{i,3} + \mu^4 e_{i,4} + \mu^5 e_{i,5} + \mu^6 e_{i,6} \quad (4.46)$$

où les coefficients sont calculés dans l'Annexe B.3.

La dérivée de (4.46) par rapport à  $\mu$  nous donne un polynôme de degré cinq :

$$\frac{\partial \mathcal{D}^{(n)}(\mathbf{A}, \mathbf{C}, \{\mathbf{D}_i\})}{\partial \mu} = \sum_{i=1}^N e_{i,1} + 2\mu e_{i,2} + 3\mu^2 e_{i,3} + 4\mu^3 e_{i,4} + 5\mu^4 e_{i,5} + 6\mu^5 e_{i,6} \quad (4.47)$$

Ainsi, la racine de (4.47) qui minimise (4.46) correspond à  $\mu_{opt}$ .

L'algorithme ELS est développé dans le pseudocode Algorithme 9.

<b>Algorithme 9: ELS</b>
Init. : coefficients $e_{i,j}$ , $j = 0, \dots, 6$ Détermination des racines $\mu$ du polynôme (4.47) (dérivée du polynôme (4.46)) $\mu_{opt} = \underset{\mu}{\operatorname{argmin}} \{ \mathcal{D}(\mathbf{A}, \mathbf{C}, \{\mathbf{D}_i\}) \}$

#### 4.2.6 Description des algorithmes

L'algorithme ALS est développé dans le pseudocode Algorithme 10.

<b>Algorithme 10: ALS</b>
Initialisation : Ensemble de matrices $\mathbf{M}_i$ , matrices initiales $\mathbf{A}^{(0)}$ et $\mathbf{D}_i^{(0)}$ et $\mathbf{C}^{(0)}$ <b>for</b> $k = 1, 2, \dots$ jusqu'à convergence <b>do</b> Mise à jour de $\mathbf{D}_i^{(k)}$ en utilisant $\mathbf{A}^{(k-1)}$ et $\mathbf{C}^{(k-1)}$ suivant (4.39) Mise à jour de $\mathbf{A}^{(k)}$ en utilisant $\mathbf{D}_i^{(k)}$ et $\mathbf{C}^{(k-1)}$ suivant (4.41) Mise à jour de $\mathbf{C}^{(k)}$ en utilisant $\mathbf{D}_i^{(k)}$ et $\mathbf{A}^{(k)}$ suivant (4.43) ou $\mathbf{C}^{(k)} = \mathbf{A}^{\ddagger(k)}$ <b>end for</b>

La vitesse de convergence peut être améliorée en utilisant l'Algorithme 9. L'algorithme résultant sera dénommé  $\text{ALS}_{\text{ELS}}$  dans la suite, et est résumé dans le pseudocode Algorithme 11.

<b>Algorithme 11: <math>\text{ALS}_{\text{ELS}}</math></b>
Initialisation : Ensemble de matrices $\mathbf{M}_i$ , matrices initiales $\mathbf{A}^{(0)}$ et $\mathbf{D}_i^{(0)}$ et $\mathbf{C}^{(0)}$ $k = 1, 2$ : ALS <b>for</b> $k = 3, \dots$ jusqu'à convergence <b>do</b> ELS ALS <b>end for</b>

### 4.2.7 Remarque

Dans le cas hermitien, il est clair que  $\mathbf{C} = \mathbf{A}^H$ , et dans le cas symétrique,  $\mathbf{C} = \mathbf{A}^T$ . Cependant, nous pouvons envisager plusieurs possibilités :

- L'estimation de  $\mathbf{C}$  se fait indépendamment de  $\mathbf{A}$ . Ceci est illustré dans les simulations ci-dessous par l'algorithme ALS,
- L'estimation de  $\mathbf{C}$  est directement obtenue à partir de  $\mathbf{A}$ . L'illustration sera alors faite via l'algorithme ALS<sub>A</sub>.

Ces méthodes, en fonction de l'application à traiter, peuvent sensiblement varier.

## 4.3 Simulations

Dans ce paragraphe, nous allons présenter les résultats que nous obtenons avec nos différents algorithmes, dans un cadre de diagonalisation conjointe. Les paramètres utilisés pour la construction de chaque figure sont décrits dans le Tableau 4.1 avec :

- Ens. : Type d'ensemble utilisé (hermitien, symétrique),
- Nbr de mat. : Nombre de matrices,
- RSB (dB) : Rapport Signal sur Bruit en dB,
- Nbr it : Nombre d'itérations retenues pour les algorithmes,
- $N_o$  : Nombre d'observations,
- $N_s$  : Nombre de sources,
- Figure : Numéro de la figure

et les résultats sont moyennés sur 100 réalisations de Monte Carlo.

Ens.	Nbr de mat.	RSB (dB)	Nbr it	$N_o$	$N_s$	Figure
Herm.	10	-10 :30	250	5	3	4.1
Herm.	10	30	1 :50	5	3	4.2
Herm.	10	-10 :30	250	4	4	4.3
Herm.	10	30	1 :100	4	4	4.4
Herm.	10	-10 :20	50	3	8	4.5
Sym.	10	-10 :30	250	5	3	4.6
Sym.	10	30	1 :50	5	3	4.7
Sym.	10	-10 :30	250	4	4	4.8
Sym.	10	30	1 :250	4	4	4.9
Herm. + Sym.	10	-10 :30	250	5	3	4.10
Herm. + Sym.	10	30	1 :100	5	3	4.11

TABLE 4.1 – Paramètres de simulations des algorithmes ALS et NOODLES

### 4.3.1 Cas hermitien

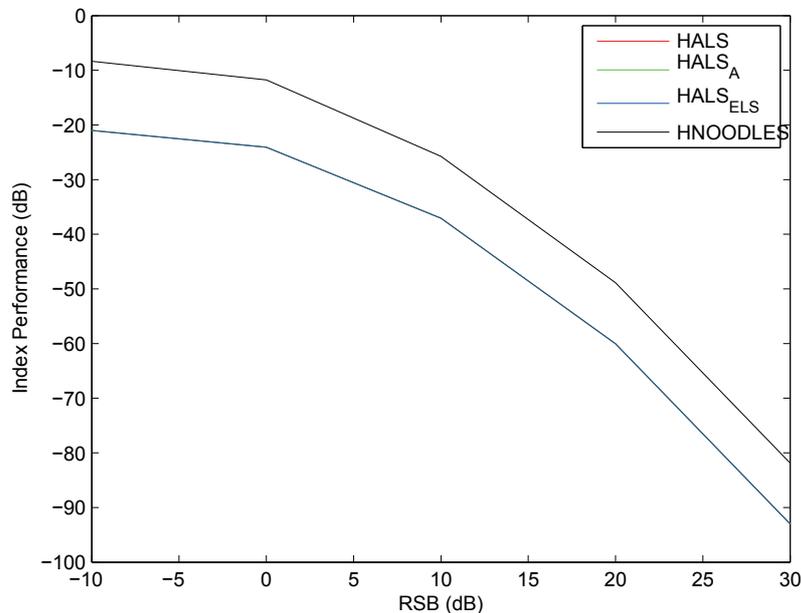


FIGURE 4.1 – Performance des algorithmes ALS et NOODLES

La Figure 4.1 fournit comme information principale que les méthodes basées sur l'ALS donnent les mêmes performances, qui sont légèrement meilleures que celle obtenue en utilisant NOODLES, dans le cas où l'on recherche une matrice de mélange rectangulaire. Les vitesses de convergence des 4 algorithmes sont illustrées dans la Figure 4.2. On remarque alors une différence par rapport au cas non bruité carré : l'ALS<sub>A</sub> converge moins vite que l'ALS.

La Figure 4.3 et la Figure 4.4 viennent confirmer que les remarques faites dans le cas rectangulaire sont valables également dans le cas carré, que ce soit en termes de vitesse de convergence, ou en termes de performances.

La Figure 4.5 présente les résultats obtenus dans le cas sous-déterminé, c'est-à-dire dans le cas où le nombre de sources  $N_s$  est supérieur au nombre d'observations  $N_o$ . Nous remarquons que nous arrivons avec l'algorithme ALS à obtenir des résultats satisfaisants.

### 4.3.2 Cas symétrique

Quant à la Figure 4.6, elle nous permet d'affirmer qu'en utilisant des ensembles hermitiens ou symétriques, nous obtenons le même niveau de performances en terme d'estimation de la matrice de mélange (ou de séparation pour NOODLES). La Figure 4.7 conforte les résultats que nous

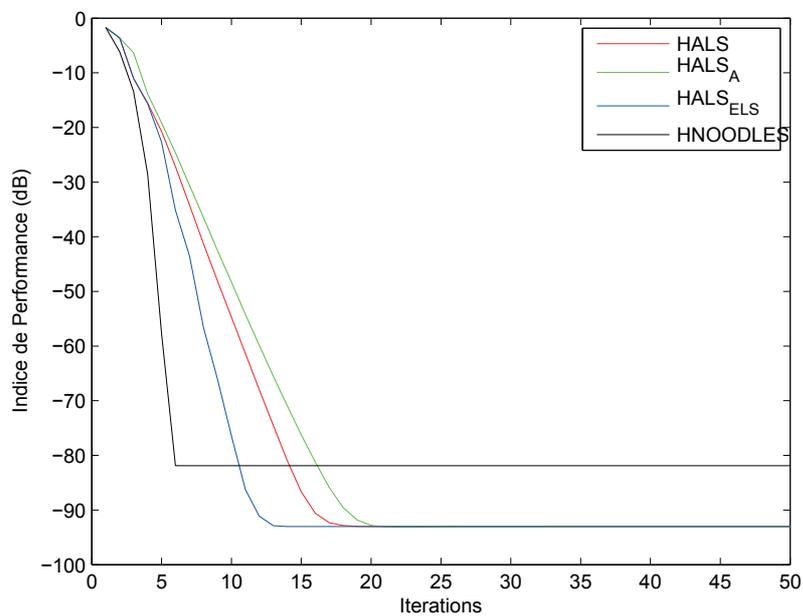


FIGURE 4.2 – Vitesse de convergence des algorithmes ALS et NODLES

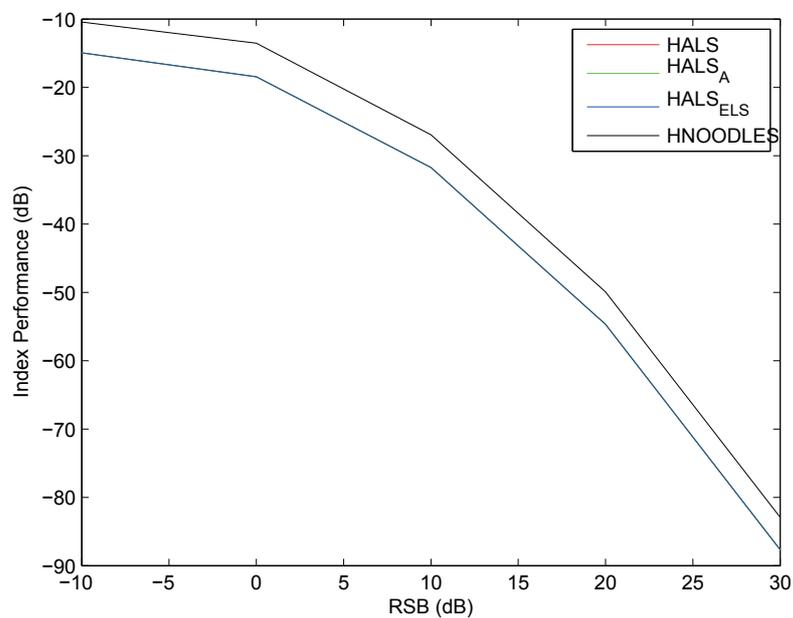


FIGURE 4.3 – Performance des algorithmes ALS et NODLES

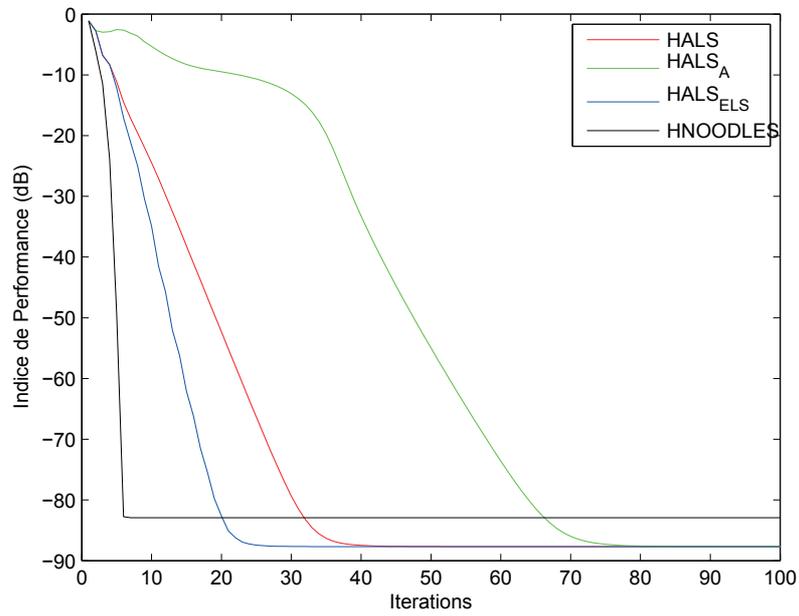


FIGURE 4.4 – Vitesse de convergence des algorithmes ALS et NODLES

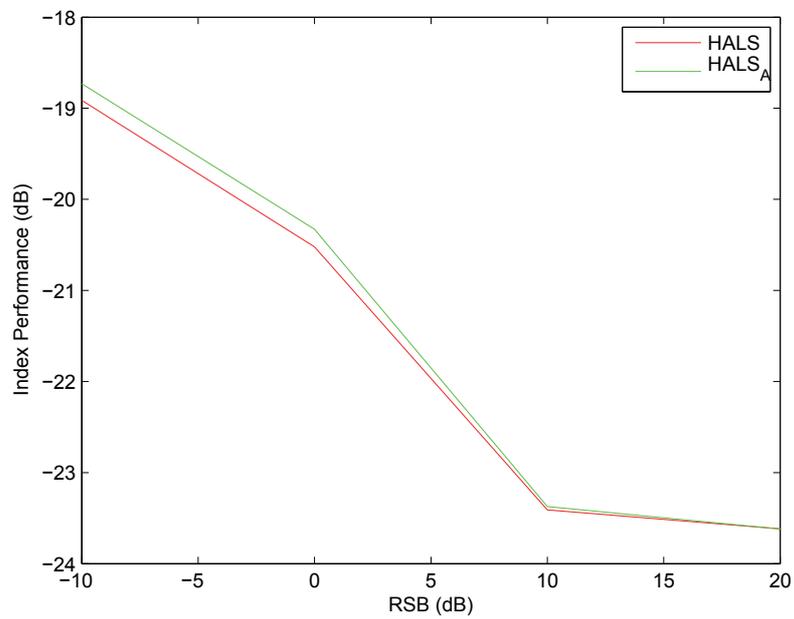


FIGURE 4.5 – Performance des algorithmes ALS

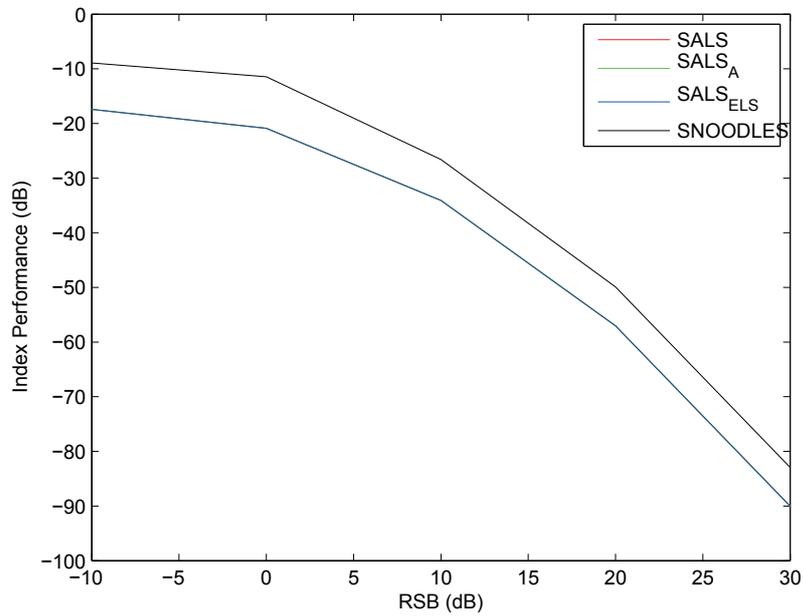


FIGURE 4.6 – Performance des algorithmes ALS et NOODLES

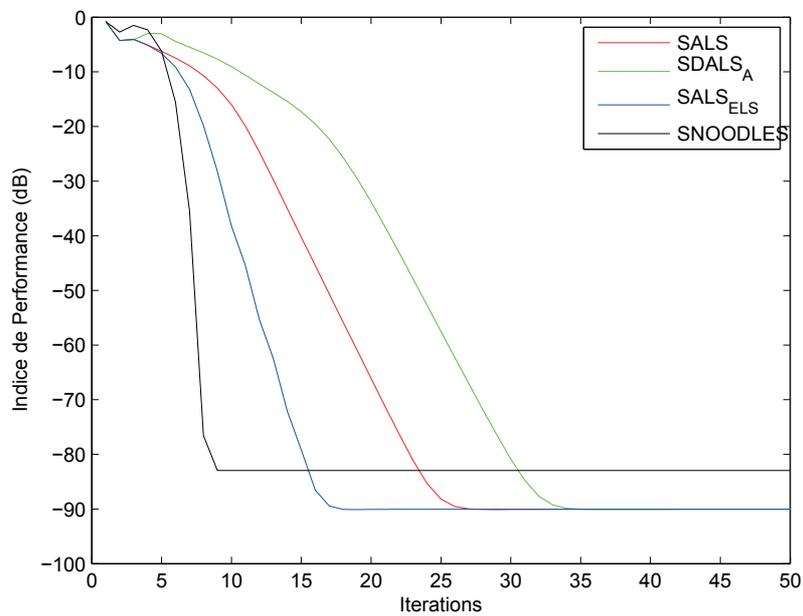


FIGURE 4.7 – Vitesse de convergence des algorithmes ALS et NOODLES

avons obtenus dans le cas hermitien, en termes de vitesse de convergence. NOODLES surclasse toujours les méthodes basées sur l'ALS.

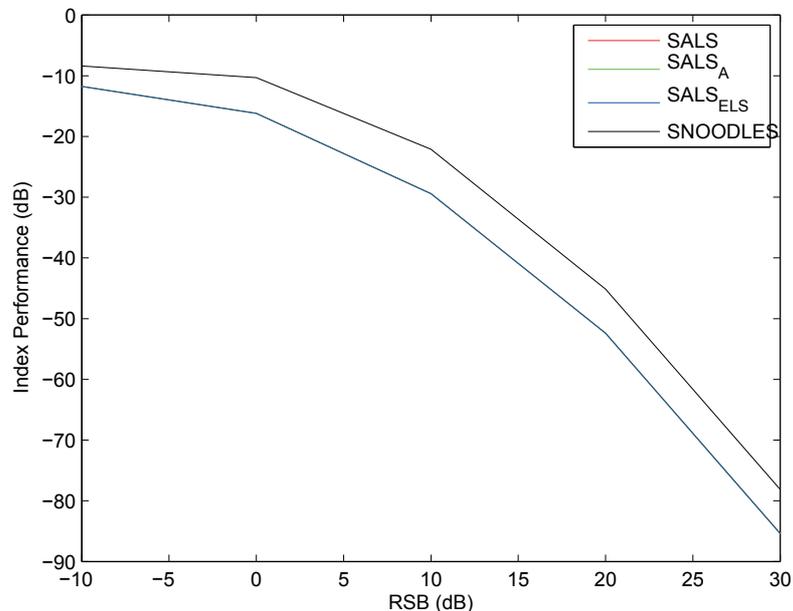


FIGURE 4.8 – Performance des algorithmes ALS et NOODLES

Les résultats obtenus dans le cas carré sont illustrés par les Figure 4.8 et 4.9 et nous amènent à tirer les mêmes conclusions.

### 4.3.3 Combinaison des deux ensembles

Dans cette partie, nous illustrons l'intérêt d'utiliser conjointement deux ensembles simultanément, et ce, en comparant les résultats obtenus lorsque un seul ensemble (en l'occurrence hermitien) est considéré.

Dans un cas bruité, et avec une matrice de mélange rectangulaire, non seulement ce type de combinaison améliore les performances d'estimation (Figure 4.10) mais aussi la vitesse de convergence (Figure 4.11).

### 4.3.4 Remarque sur les temps de calculs

Dans ce paragraphe, nous montrons l'influence des paramètres pris en compte dans les algorithmes sur les temps de calcul. Le tableau 4.2 résume ceci. On remarque alors aisément que l'ELS est la méthode qui nécessite le plus de ressources informatiques. Cependant, comme elle améliore la vitesse de convergence par rapport aux méthodes ALS et ALS<sub>A</sub>, cela permet de

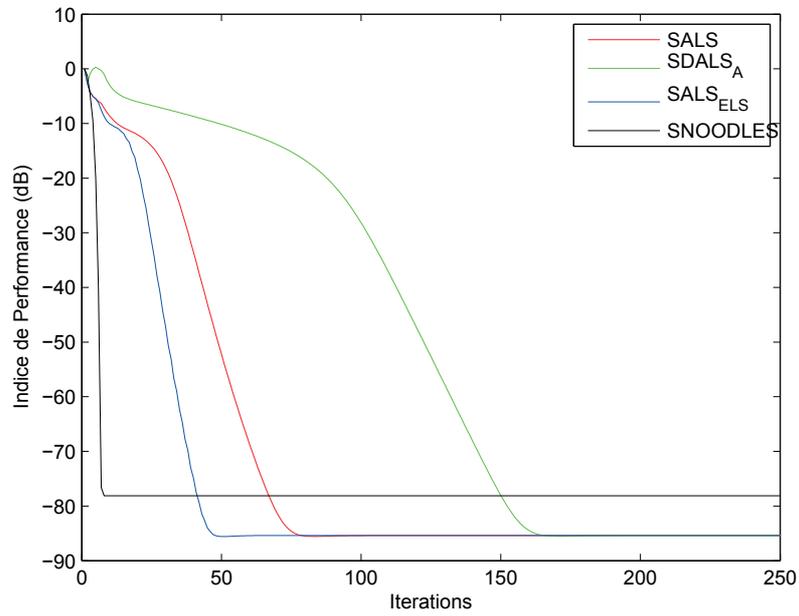


FIGURE 4.9 – Vitesse de convergence des algorithmes ALS et NODDLES

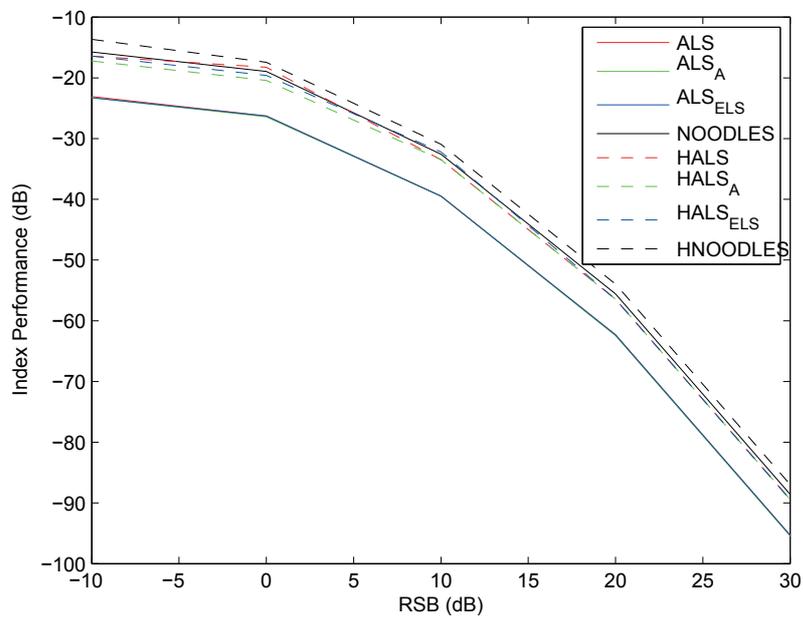


FIGURE 4.10 – Performance des algorithmes ALS et NODDLES

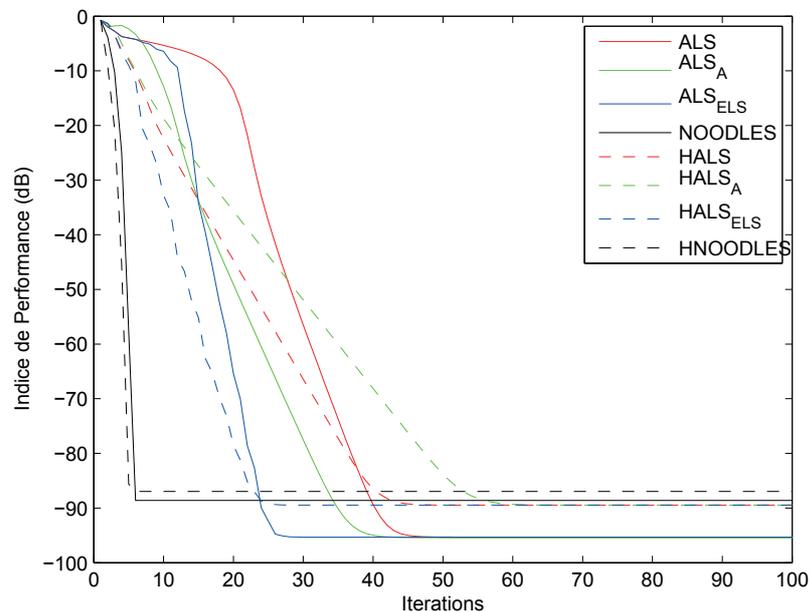


FIGURE 4.11 – Vitesse de convergence des algorithmes ALS et NOODLES

contrebalancer la remarque. De plus, comme cela a déjà été évoqué précédemment, l'ELS est ici effectué à chaque itération. Or, il peut être tout à fait envisagé, pour réduire le temps de calcul, de n'effectuer ceci que toutes les  $n$  itérations. Enfin, on note que NOODLES est la solution la moins coûteuse. En effet, le temps d'exécution est très proche de l'ALS, mais on a également noté que cet algorithme converge très rapidement.

Méthode	cas herm. (ms)	cas sym. (ms)	cas herm. + sym. (ms)
ALS	0.83	0.80	1.27
ALS <sub>A</sub>	0.56	0.55	0.83
ALS <sub>ELS</sub>	2.51	2.52	4.51
NOODLES	0.88	0.87	2.06

TABLE 4.2 – Temps d'exécution des algorithmes ALS et NOODLES

#### 4.4 Conclusions sur les méthodes de recherche analytiques

Aux vues des résultats que nous avons obtenu, nous pouvons dire que l'ALS est une bonne solution à l'estimation de la matrice de mélange. En effet, sa mise en oeuvre est extrêmement simple et ne nécessite que peu de ressources. Cependant, sa vitesse de convergence peut être longue. C'est pour cela que l'ELS a été utilisé afin de réduire ceci, au détriment du temps de

calcul. Il est aussi utile de noter que parfois, l'utilisation de l'algorithme  $ALS_A$  peut être une alternative intéressante. En effet, on améliore les temps de calcul, tout en gardant une vitesse de convergence acceptable.

De plus, nous notons que l'algorithme NOODLES est celui qui offre le meilleur compromis temps de calcul/vitesse de convergence, bien qu'il ne soit pas aussi évident à mettre en oeuvre que l'ALS, et qu'il fournisse des performances légèrement moins bonnes.

Enfin, la même remarque que celle énoncée dans le chapitre précédent peut être faite : la prise en compte d'une diversité statistique permet d'améliorer les performances d'estimation de nos algorithmes.

## Chapitre 5

# Application à la séparation de sources de télécommunications

### 5.1 Séparation de sources

Comme énoncé précédemment, le problème de la séparation de sources peut se résoudre de plusieurs manières. Le principe le plus couramment utilisé est l'Analyse en Composantes Indépendantes (ACI), dont l'objectif est de rendre indépendantes les données collectées après transformation linéaire.

Le principe précédent se transpose de manière algébrique sur des problèmes de diagonalisation conjointe de matrices. C'est ce que nous allons faire dans la suite, en utilisant les algorithmes développés dans les parties précédentes, et les comparer à des méthodes proposées dans la littérature.

Nous reprenons le modèle défini en (2.18) :

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}(t) \quad (5.1)$$

Nous allons utiliser les définitions des cumulants vues en (2.25) et (2.26). Le cumulants présente plusieurs avantages. Tout d'abord, comme les sources et le bruit sont statistiquement indépendants, on peut écrire le cumulants de  $\mathbf{y}$  de la manière suivante :

$$\begin{aligned} C_y^{a,b,c,d} &= \text{Cum} \{y_a, y_b^*, y_c, y_d^*\} \\ &= \text{Cum} \{x_a, x_b^*, x_c, x_d^*\} + \text{Cum} \{b_a, b_b^*, b_c, b_d^*\} \\ &= C_x^{a,b,c,d} + C_b^{a,b,c,d} \end{aligned} \quad (5.2)$$

De plus, comme nous considérons dans cette partie que le bruit est issu d'un processus

gaussien, son cumulatif sera nul. On obtient alors :

$$C_y^{a,b,c,d} = C_x^{a,b,c,d} \quad (5.3)$$

## 5.2 Hypothèses de simulation

Dans ce qui va suivre, nous allons mettre en place notre cadre de travail. Nous allons aussi énoncer toutes les hypothèses qui vont nous être utiles.

- Toutes les variables sont complexes,
- Nous considérons  $M$  capteurs, et  $N$  sources, avec  $N \leq M$ ,
- La matrice de mélange est générée aléatoirement, suivant une loi uniforme  $\mathcal{U}(\sqrt{-3}, \sqrt{3})$  et où chaque vecteur  $a_n$  est ensuite normé,
- Le bruit est gaussien, de moyenne nulle et de variance  $\sigma_b$ .

Concernant les sources  $\mathbf{s}$ , nous allons utiliser des "Quaternary Phase Shift Keying" (QPSK). Cependant, ces signaux sont circulaires, *i.e.*  $\mathbf{E}\{\mathbf{s}\mathbf{s}^T\} = 0$ . Afin d'utiliser des ensembles symétriques, nous rendons ces sources non circulaires, à chaque instant  $t$ , en utilisant la formulation suivante :

$$\mathbf{s}_c = \text{real}(\mathbf{s})\cos(\theta) + j\text{imag}(\mathbf{s})\sin(\theta) \quad \theta \in \left]0, \frac{\pi}{4}\right[ \cup \left]\frac{\pi}{4}, \frac{\pi}{2}\right[ \quad (5.4)$$

Dans ce qui suivra,  $\theta$  sera fixé à  $\frac{\pi}{8}$ .

Afin de valider le bon fonctionnement de nos algorithmes, nous allons les comparer aux trois méthodes rappelées dans les Algorithmes 1, 2 et 3. Cependant, certaines de ces méthodes (U-WEDGE et FAJD) ne peuvent traiter directement le cas  $N < M$ . Ainsi, nous effectuerons une étape préalable classique, consistant à projeter les données sur le sous-espace signal, obtenu en considérant les vecteurs propres associés aux  $N$  plus grandes valeurs propres de la décomposition en valeurs propres de la matrice de covariance.

Dans les simulations qui suivront, nous définirons le RSB comme étant le ratio entre la puissance des sources  $\sigma_s^2$  et la puissance du bruit  $\sigma_b^2$ , dont la mise en équation donne :

$$\text{RSB}_{\text{dB}} = 10 \log_{10} \frac{\sigma_s^2}{\sigma_b^2} \quad (5.5)$$

Les paramètres sont décrits dans le Tableau 5.1 avec :

- Nbr de pts : Nombre de points pour l'estimation des cumulants,
- Ens. : Type d'ensemble considéré,
- RSB(dB) : Rapport Signal sur Bruit en dB,
- Nit : Nombre d'itérations,
- $N_o$  : Nombre d'observations,

- $N_s$  : Nombre de sources,
- Figure : Numéro de la figure

Les résultats de simulation sont moyennés sur 100 réalisations de Monte Carlo (MC). Le nombre de matrices est égal au nombre d'observations, puisque l'ensemble est directement déduit de l'estimation des cumulants d'ordre quatre.

Nbr de pts	Ens.	RSB(dB)	Nit	$N_o$	$N_s$	Figure
$2^{10}$	Herm.	10	1 :100	3	2	5.1
$2^{10}$	Herm.	-10 :20	150	3	2	5.2
$2^{10}$	Herm.	10	1 :250	4	4	5.3
$2^{10}$	Herm.	-10 :20	150	4	4	5.4
$2^{10}$	Sym.	10	1 :120	3	2	5.5
$2^{10}$	Sym.	-10 :20	150	3	2	5.6
$2^{10}$	Sym.	10	1 :180	4	4	5.7
$2^{10}$	Sym.	-10 :20	150	4	4	5.8
$2^{12}$	Herm. + Sym.	20	1 :200	5	2	5.9
$2^{12}$	Herm. + Sym.	-10 :20	200	5	2	5.10
$2^{11}$	Herm. + Sym.	30	1 :100	5	3	5.11
$2^{11}$	Herm. + Sym.	-10 :30	100	5	3	5.12

TABLE 5.1 – Paramètres de simulations de signaux de télécommunications

### 5.3 Cas hermitien

La Figure 5.1 montre la vitesse de convergence des différents algorithmes considérés. Nous remarquons alors que NOODLES converge légèrement plus rapidement que U-WEDGE, et bien plus vite que, respectivement, FAJD et DESERT. En ce qui concerne les méthodes utilisant le critère direct, nous remarquons que l'ALS converge plus rapidement que ACDC.

La Figure 5.2 illustre les performances des méthodes en fonction du RSB. On remarque que l'on obtient sensiblement les mêmes résultats pour les algorithmes FAJD et U-WEDGE mais qui sont meilleurs que NOODLES et DESERT pour des niveaux de bruits élevés ( $RSB < 10$  dB). Puis les performances deviennent égales au-delà de 10dB. L'ALS et ACDC fournissent quant à eux la même estimation de la matrice de mélange, et ce quelque soit le bruit.

La Figure 5.3 illustre les résultats que l'on observe dans le cas carré. Les remarques faites dans le cas rectangulaire restent vraies : NOODLES et U-WEDGE convergent sensiblement aussi rapidement, et bien plus vite que, respectivement, FAJD et DESERT. L'ALS converge toujours plus vite que ACDC.

La Figure 5.4 présente l'influence du bruit sur les performances d'estimation dans le cas carré. On remarque que les algorithmes utilisant le critère inverse fournissent sensiblement les mêmes

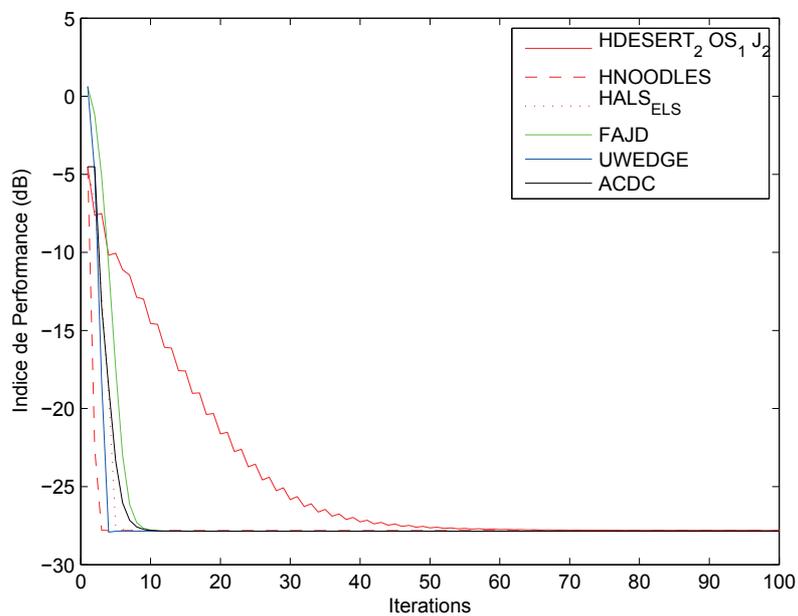


FIGURE 5.1 – Vitesse de convergence des algorithmes

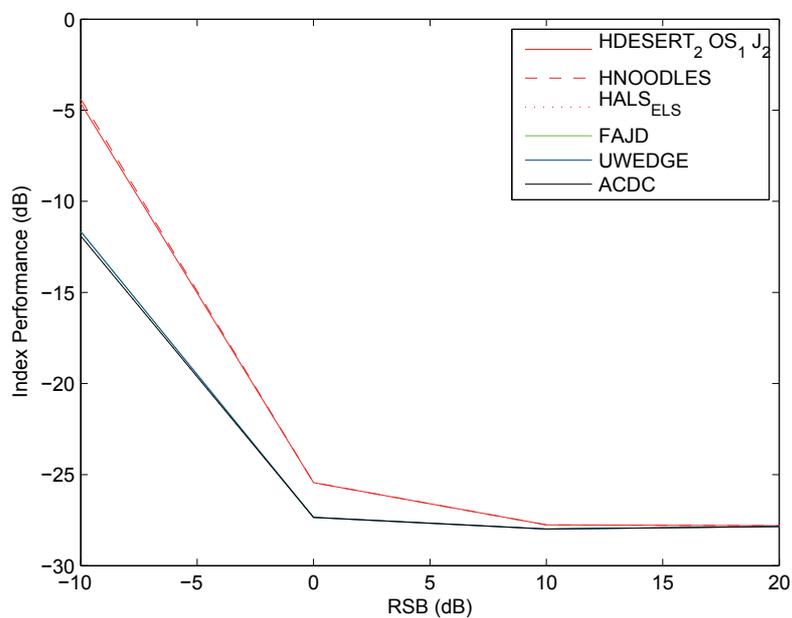


FIGURE 5.2 – Performance des algorithmes en fonction du RSB

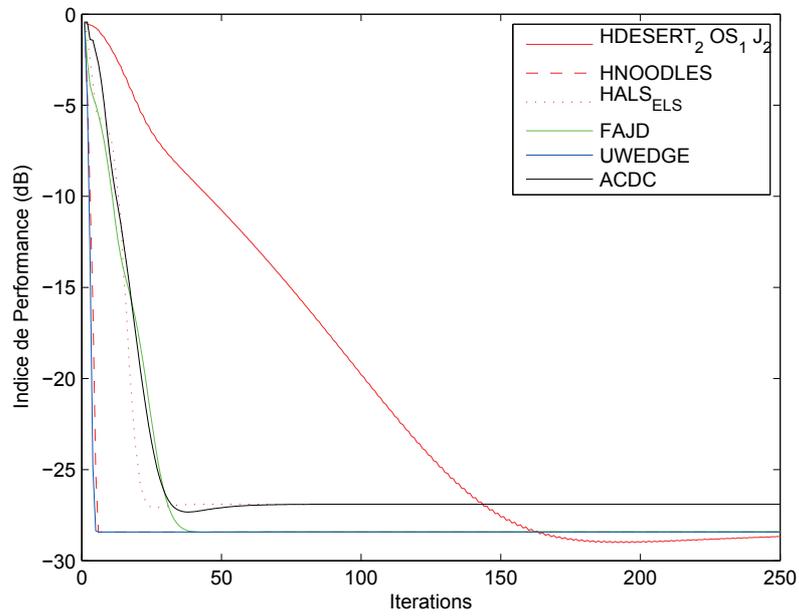


FIGURE 5.3 – Vitesse de convergence des algorithmes

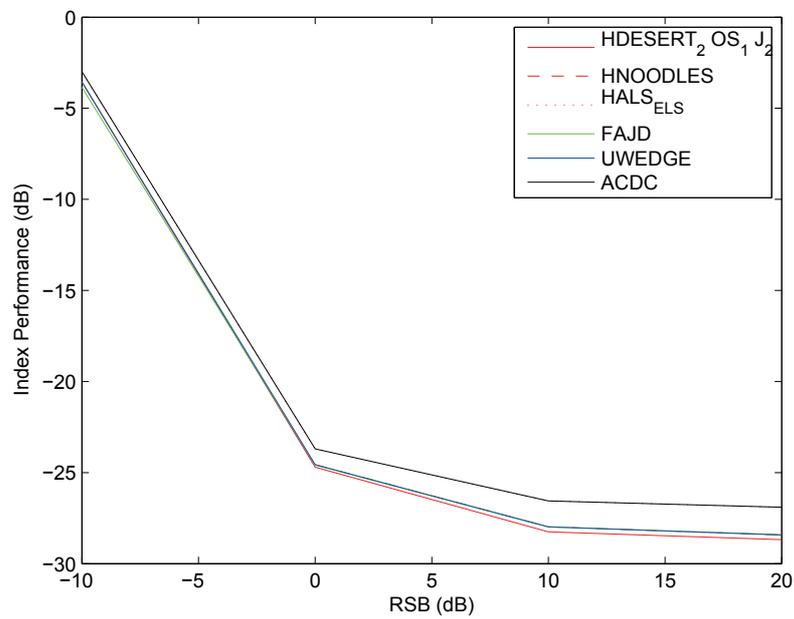


FIGURE 5.4 – Performance des algorithmes en fonction du RSB

résultats, qui sont meilleurs que ceux obtenus avec les méthodes utilisant le critère direct.

## 5.4 Cas symétrique

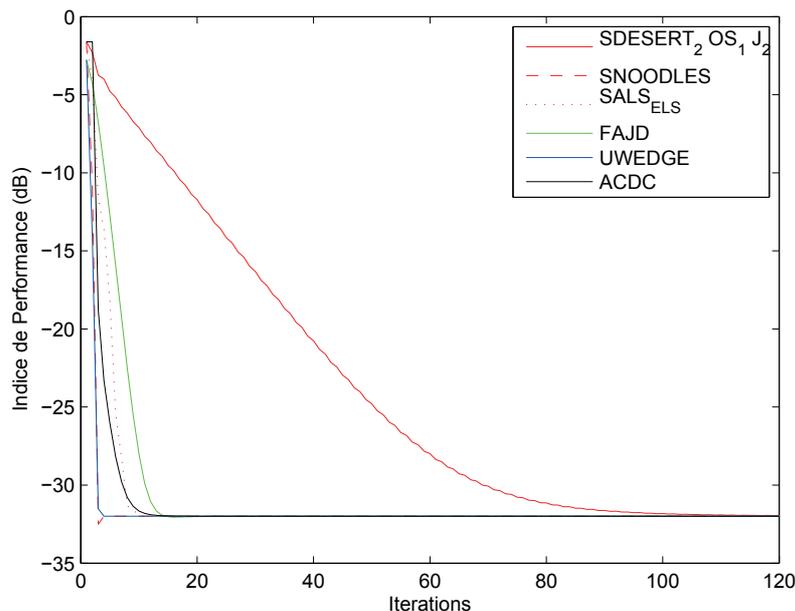


FIGURE 5.5 – Vitesse de convergence des algorithmes

La Figure 5.5 montre la vitesse de convergence des différents algorithmes considérés, dans le cas où la matrice de mélange est rectangulaire. Les remarques faites dans le paragraphe précédent, où les méthodes utilisent un ensemble composé de matrices hermitiennes, peuvent être maintenues.

La Figure 5.6 illustre les performances que l'on obtient dans le cas où les ensembles considérés sont symétriques. On remarque que les méthodes DESERT et NOODLES fournissent de moins bons résultats que les autres méthodes lorsque l'on est en présence de bruits ayant une forte variance.

Les résultats illustrant la vitesse de convergence des algorithmes est illustré dans la Figure 5.7. On remarque ici que NOODLES est bien plus rapide que les autres méthodes, surtout U-WEDGE qui, dans les simulations précédentes, convergeait sensiblement aussi rapidement. L'influence du bruit sur les ensembles symétriques, dans le cas où la matrice de mélange est carrée, est présentée dans la Figure 5.8. On note que toutes les méthodes fournissent sensiblement les mêmes résultats.

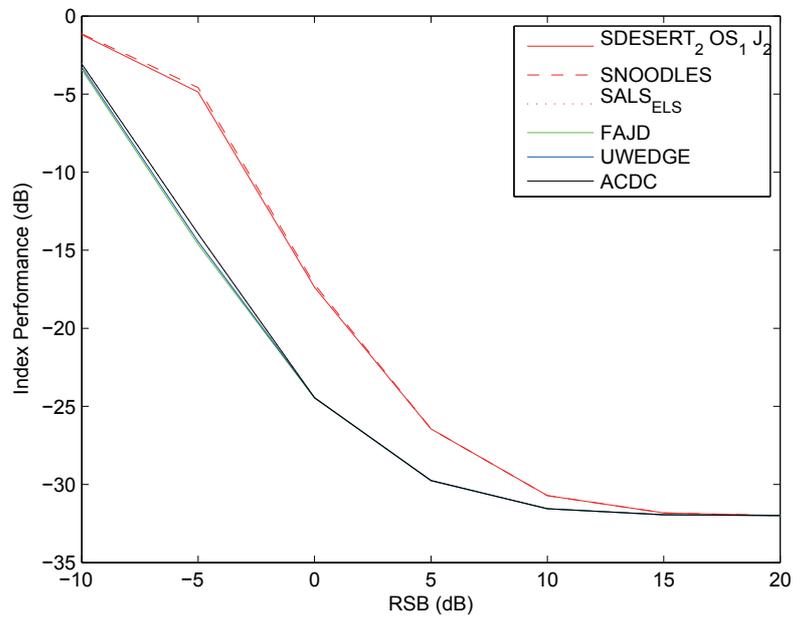


FIGURE 5.6 – Performance des algorithmes en fonction du RSB

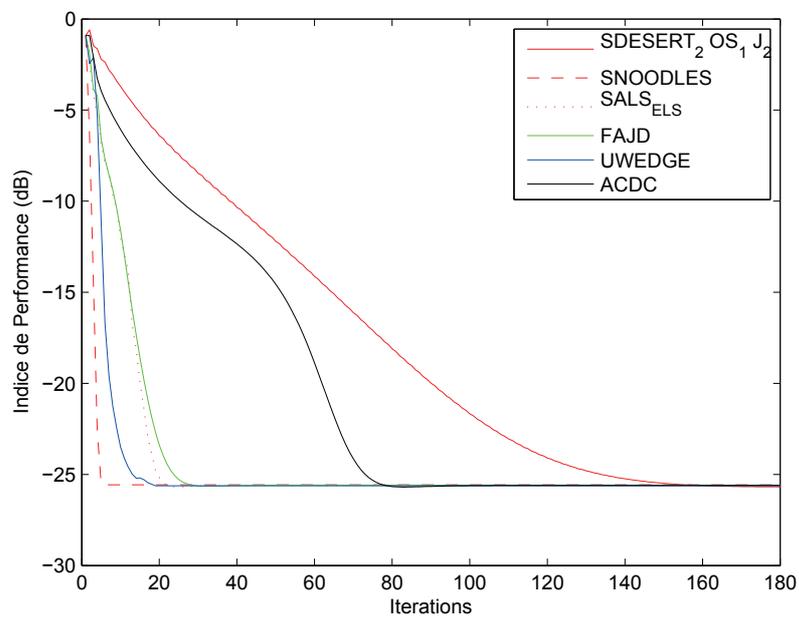


FIGURE 5.7 – Vitesse de convergence des algorithmes

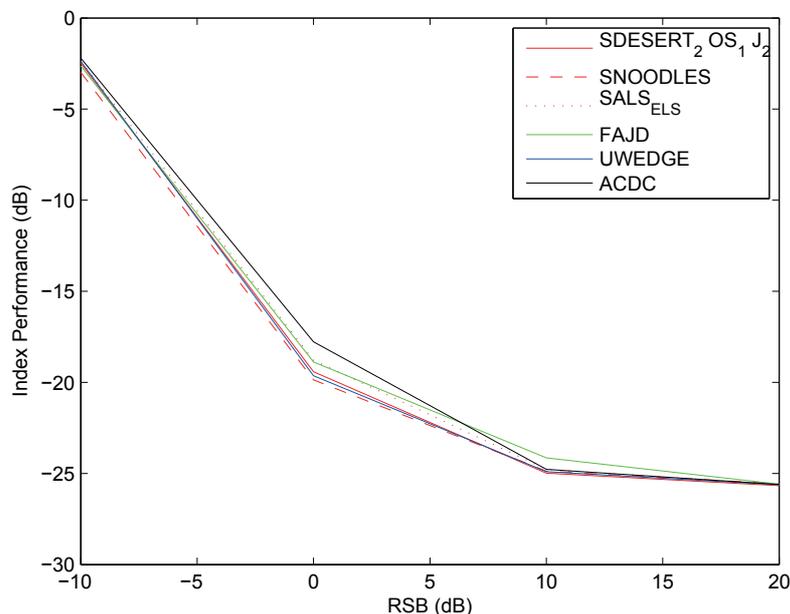


FIGURE 5.8 – Performance des algorithmes en fonction du RSB

## 5.5 Combinaison des deux ensembles

Comme nous l'avons évoqué précédemment, nous allons illustrer, dans un premier temps, l'influence de la prise en compte des ensembles hermitiens et symétriques en même temps, en comparant nos algorithmes DESERT, NOODLES et ALS avec leurs versions n'utilisant que l'ensemble hermitien.

La Figure 5.9 nous montre l'influence de la combinaison des deux ensembles sur la vitesse de convergence. On remarque que cela n'a pas une grande influence sur NOODLES, contrairement aux méthodes ALS et DESERT.

La Figure 5.10 illustre le comportement de ces méthodes en fonction du RSB. On remarque une amélioration des performances lorsqu'on combine les deux ensembles. Ceci nous amène à penser que ce type d'utilisation est plus robuste aux perturbations, comparé aux méthodes n'utilisant qu'un seul ensemble.

La Figure 5.11 compare la vitesse de convergence des méthodes DESERT, NOODLES, ALS et FAJD. Nous remarquons que NOODLES surclasse les autres méthodes, et que l'ALS et FAJD convergent aussi rapidement. De plus, on se rend compte que les algorithmes basés sur l'estimation du gradient mettent beaucoup plus de temps à converger en présence de perturbations, et sont donc bien plus sensibles que les autres méthodes illustrées.

L'évaluation de la robustesse des méthodes à différents niveaux de puissance de perturbations

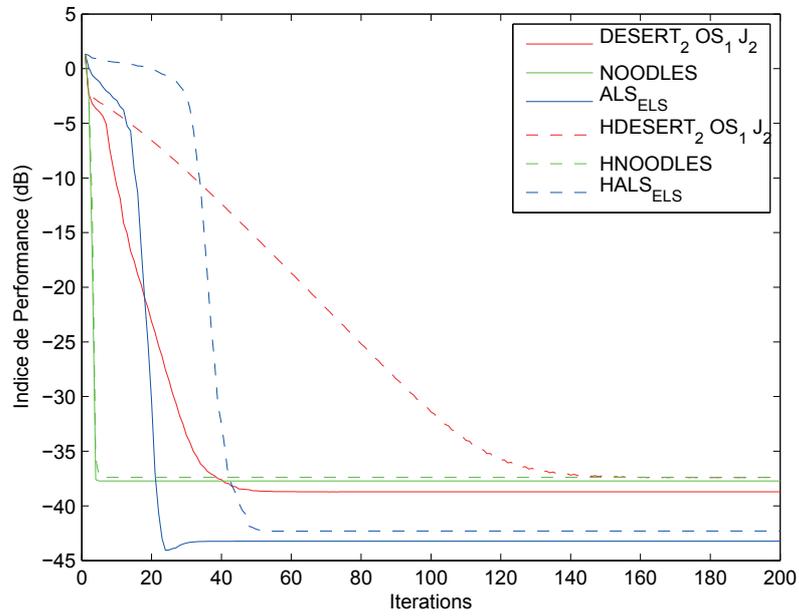


FIGURE 5.9 – Vitesse de convergence des algorithmes

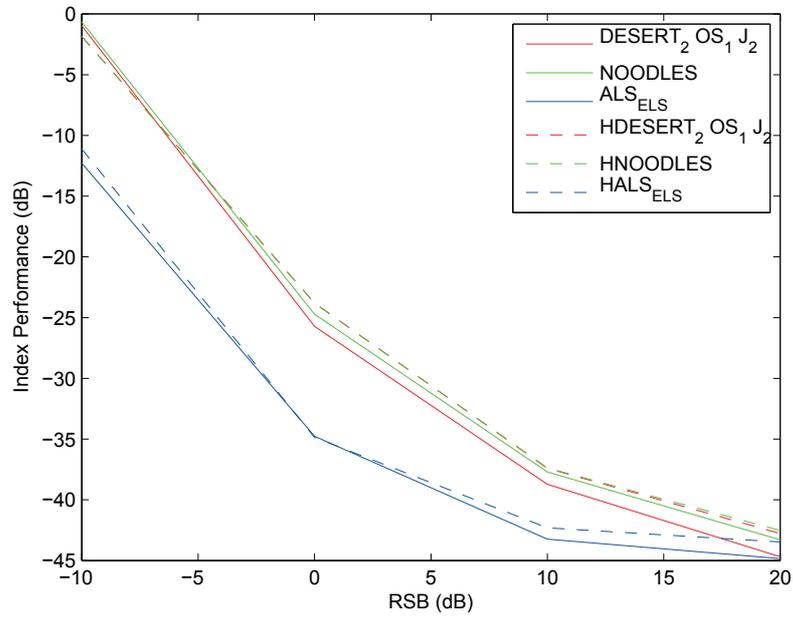


FIGURE 5.10 – Performance des algorithmes en fonction du RSB

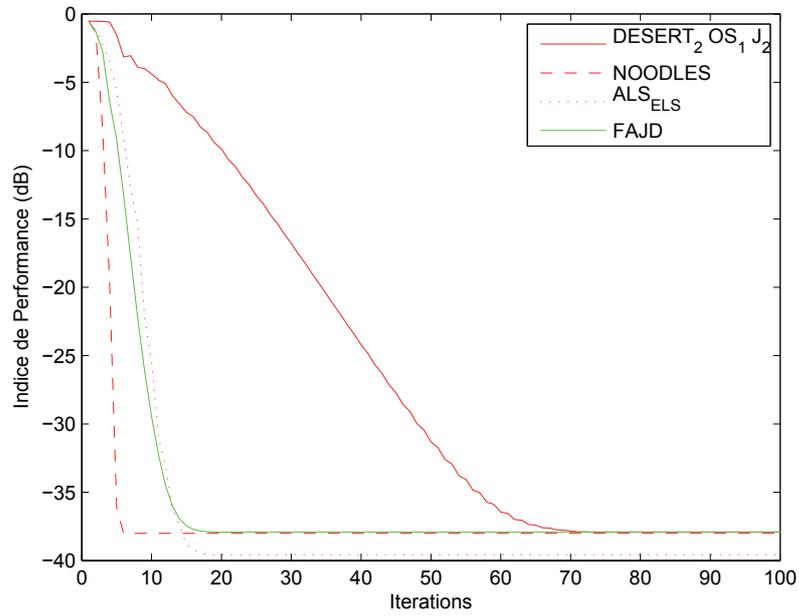


FIGURE 5.11 – Vitesse de convergence des algorithmes

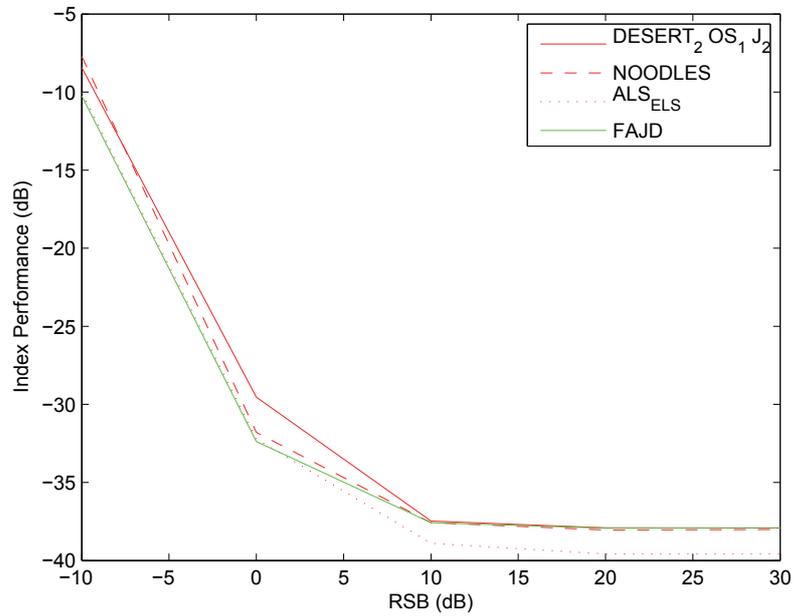


FIGURE 5.12 – Performance des algorithmes en fonction du RSB

## 5.6. CONCLUSIONS SUR L'APPLICATION AUX SOURCES DE TÉLÉCOMMUNICATIONS NUMÉRIQUE

est présenté dans la Figure 5.12. Pour de faibles niveaux de RSB ( $< 0\text{dB}$ ), FAJD et l'ALS fournissent des résultats similaires, et meilleurs que ceux obtenus avec DESERT et NOODLES. Au delà de  $10\text{dB}$ , on remarque que L'ALS donne une bien meilleure estimation de la matrice de mélange que les trois autres méthodes utilisées pour la comparaison, qui donnent, quant à elles, des résultats identiques.

### 5.6 Conclusions sur l'application aux sources de télécommunications numériques

Les résultats obtenus dans ce chapitre permettent d'illustrer le fait que les méthodes développées en diagonalisation conjointe peuvent être utilisables afin de résoudre des problèmes de séparation de sources. Nous remarquons que l'ALS fournit des résultats intéressants, que ce soit en termes de performances ou de vitesse de convergence. De plus, on observe cet algorithme, ainsi que DESERT et NOODLES, sont très intéressants lorsque la matrice recherchée est rectangulaire, puisqu'ils permettent de s'affranchir de l'étape de blanchiment tout en affichant des performances similaires.

## Chapitre 6

# Application à la radioastronomie<sup>1</sup>

### 6.1 Présentation

Dans ce qui va suivre, nous allons considérer des données issues du radiotélescope LOFAR, développé par la fondation astronomique néerlandaise ASTRON. Outre sa dimension, ce radiotélescope se distingue par une approche tout numérique et par une hiérarchisation astucieuse de son architecture :

1. les antennes 'basse fréquence' ou LBA (30MHz à 80 MHz) sont de simples dipôles en V. Réparties aléatoirement par paquet de 96 dans un disque de 80m de diamètre, elles forment une station. Ces 96 antennes sont échantillonnées puis phasées numériquement entre elles pour fournir deux faisceaux et deux polarisations par station. Il y a actuellement 41 stations LBA dont une grosse partie est concentrée dans un disque de 3km de diamètre (cf. figure 6.1), une autre partie est disponible dans un anneau qui s'étend sur 80km et quelques stations sont disponibles à des distances de plusieurs centaines de kilomètres, comme par exemple la station LOFARFR606 installée sur le site de la station de radioastronomie de Nançay dans le Cher.
2. les antennes 'haute fréquence' ou HBA (110 MHz à 250 MHz) sont d'abord sommées analogiquement par groupe de seize afin d'augmenter la sensibilité. Le signal obtenu ainsi que celui de 95 (ou 48 selon les configurations) autres paquets sont numérisés puis phasés pour former des stations de manière similaire à la description faite pour le réseau LBA. Actuellement 65 stations HBA sont connectées.
3. l'ensemble de ces stations sont connectées via des fibres optiques vers un calculateur à très haute capacité qui va combiner tous ces signaux en temps réel afin de fournir les différents

---

1. En collaboration avec G. Hellbourg et R. Weber de l'Observatoire de Paris (station de Radioastronomie de Nançay), et de l'institut de radioastronomie néerlandais ASTRON, dans le cadre du PPF ISSO

modes d'observation du radiotélescope LOFAR. Le seul mode qui nous intéresse (Direct Storage) sera détaillé, les autres (Interferometric et Beam-Formed) sont décrits sur ce site [5].

4. Le mode Direct Storage est décomposé en deux sous-modes : le "Transient Buffer Board" (TBB) et le Single Station Observations. Le premier nous concerne tout particulièrement, puisque les données traitées dans la suite proviennent de ce procédé. Il permet de sauvegarder 1.3 s des 96 signaux issus des antennes d'une station HBA ou LBA. La fréquence d'acquisition est de 200MHz.



FIGURE 6.1 – Superterp, le coeur du coeur de LOFAR près de Exloo aux Pays-Bas <sup>a</sup>

---

a. Toutes les photos et les figures sont utilisées avec l'aimable autorisation de ASTRON

Grâce à ses capacités d'imagerie multi-fréquence et multi-faisceaux, l'objectif scientifique de LOFAR est d'observer l'univers à différentes échelles afin de détecter des signatures caractéristiques. Les phénomènes potentiellement détectables, grâce à ce type de matériel, sont la réionisation de l'hydrogène neutre (caractéristique de la création des premières étoiles et galaxies, et donc de l'Univers), les galaxies massives, les pulsars, l'éjection de masse coronale du Soleil et la cartographie des vents solaires, les rayons cosmiques et les champs magnétiques, les irrégularités dans la ionosphère et éventuellement d'autres détectations non prévues ni envisagées.

Les sources cosmiques sont étudiées à travers les ondes radios qu'elles émettent. Cependant, ces émissions naturelles partagent les mêmes bandes de fréquence que les émetteurs de télécommunications développés pour diverses applications (Télévision Numérique Terrestre, Radio FM, GSM, GPS ,etc). Bien que quelques bandes de fréquence soient réservées (en théorie) exclusivement à la radioastronomie, les besoins scientifiques imposent d'utiliser tout le spectre

électromagnétique. Il faut donc développer des techniques de traitement du signal [33, 74, 8] qui permettent aux radioastronomes d'obtenir des observations de qualité même en présence d'émissions d'origine anthropique. Dans le milieu de la radioastronomie, on parle de "Radio frequency interference" (RFI).

Notre objectif est alors le suivant : Déterminer au mieux la matrice de mélange de ces interférences, afin d'éliminer (ou du moins réduire) leur influence sur nos données. Ainsi, cela permettra, en principe, d'avoir une bien meilleure observation des sources cosmiques, et donc de déterminer si ce que l'on a en notre possession est déjà parfaitement connu, ou alors si ce sont de nouvelles signatures à considérer plus précisément.

Pour évaluer l'intérêt des méthodes proposées nous allons utiliser dans un premier temps des données synthétiques issues d'un simulateur de station LOFAR puis dans un deuxième temps des données réelles obtenues via les TBB.

## 6.2 Données synthétiques

### 6.2.1 Modèle

On s'intéresse à l'analyse de signaux reçus sur un réseau de  $M$  antennes phasées. Le signal reçu est un mélange composé de  $N_s$  sources cosmiques et de  $N_r$  interférences. De plus, ces observations sont perturbées par du bruit additif. Nous considérons alors le signal reçu sur chaque capteur, à chaque instant  $t$ , comme un mélange linéaire instantané. Cela s'écrit de la manière suivante :

$$x_m(t) = \sum_{i=1}^{N_s} (a_s)_{mi} (s_s)_i(t) + \sum_{k=1}^{N_r} (a_r)_{mk} (s_r)_k(t) + n_m(t) \quad (6.1)$$

Nous supposons dans la suite que nous sommes dans une situation où les signaux sont à bande étroite. Cette relation peut alors s'écrire sous forme matricielle :

$$\mathbf{x}(t) = \mathbf{A}_r \mathbf{s}_r(t) + \mathbf{A}_s \mathbf{s}_s(t) + \mathbf{n}(t) \quad (6.2)$$

avec  $\mathbf{A}_r$  la matrice de mélange contenant les signatures spatiales des interférences dont les éléments sont les coefficients  $(a_r)_{mk}$ ,  $\mathbf{s}_r$  contient le signal des interférences à chaque instant,  $\mathbf{A}_s$  la matrice de mélange caractérisant la signature spatiale des sources cosmiques dont les éléments sont les coefficients  $(a_s)_{mi}$ ,  $\mathbf{s}_s$  contient le signal émis par celles-ci, et  $\mathbf{n}$  représente un bruit additif venant des capteurs et de l'électronique. Tout ceci nous permet donc d'obtenir les observations au niveau du radiotélescope  $\mathbf{x}$  à chaque instant  $t$ .

On s'intéresse à l'élimination de ces interférences. Comme ce sont, dans un premier temps,

ces signaux qui doivent être éliminés, nous allons réécrire le modèle (6.2) sous la forme :

$$\mathbf{x}(t) = \mathbf{A}_r \mathbf{s}_r(t) + \mathbf{b}(t) \quad (6.3)$$

Nous faisons donc une nouvelle interprétation. Nos signaux d'intérêt sont les signaux de télécommunication, et tout le reste n'est considéré que comme des perturbations. Les sources cosmiques sont des signaux gaussiens, tout comme le bruit. Quant aux signaux de télécommunications, ils sont dans cette simulation de type "Binary Phase Shift Keying" (BPSK), mis en forme en utilisant un filtre gaussien, et ont pour porteuse une exponentielle complexe.

Nos méthodes de développement utilisant conjointement des matrices hermitiennes et symétriques ne seront pas utilisées dans cette partie. En effet, nous nous sommes seulement concentrés sur l'utilisation d'un ensemble de matrices, et non plus d'une seule matrice, comme cela est fait classiquement dans ce domaine, afin d'en étudier les bénéfices. Nous n'utiliserons alors, à l'ordre deux, que les matrices de corrélation pour différentes valeurs de  $\tau$ .

Traditionnellement, le traitement des interférences, lorsque des méthodes de type séparation de sources ou traitement d'antennes sont utilisées, est effectué en utilisant simplement la matrice de covariance des observations. L'objectif principal est donc d'utiliser un ensemble de matrices plus grand que ce qui se fait usuellement dans cette application, afin d'estimer de manière plus précise la matrice de mélange des interférences et donc de les éliminer au mieux des observations que l'on recueille.

Nous utilisons alors un simulateur développé au sein d'ASTRON par G. Hellbourg. Il permet de prendre en compte plusieurs paramètres comme le nombre d'échantillons, le nombre d'interférence, la puissance des interférences, le type d'antenne et l'espacement intercapteurs. Nous ne présenterons des résultats qui ne traiteront que de l'influence des trois premiers paramètres.

Dans les simulations qui suivront, nous définirons le RIB comme étant le ratio entre la puissance des interférences  $\sigma_i^2$  et la puissance du bruit  $\sigma_b^2$ , dont la mise en équation donne :

$$\text{RIB}_{\text{dB}} = 10 \log_{10} \frac{\sigma_i^2}{\sigma_b^2} \quad (6.4)$$

### 6.2.2 Simulations informatiques

Dans cette partie, nous présentons les résultats que nous obtenons avec le simulateur, et comparons les méthodes DESERT, NOODLES et ALS.

La Figure 6.2 illustre une simulation de vue du ciel dans le cas où aucune perturbation n'est visible, et où trois sources cosmiques d'intérêt sont clairement observables. Ce type de figure est créé à partir de la matrice de corrélation à  $\tau = 0$ .

La Figure 6.3, quant à elle, illustre la même vue mais en présence d'interférences terrestres.

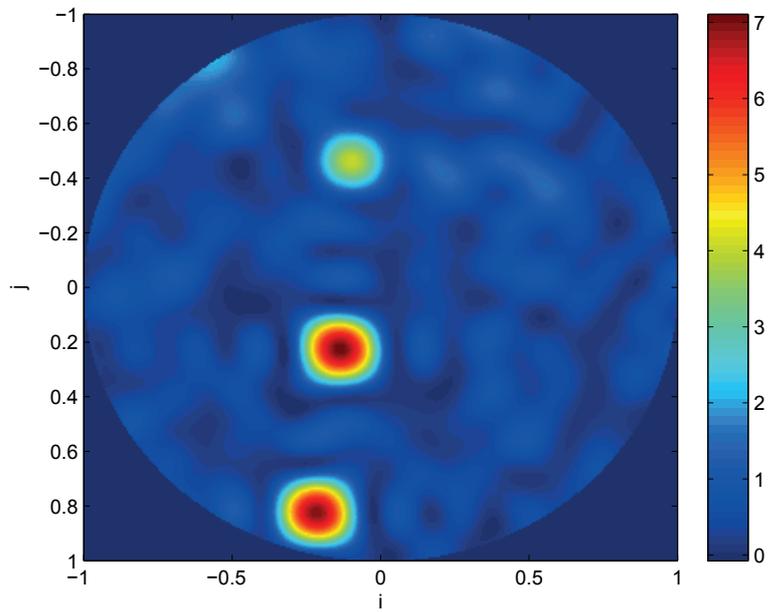


FIGURE 6.2 – Vue du ciel sans perturbateurs

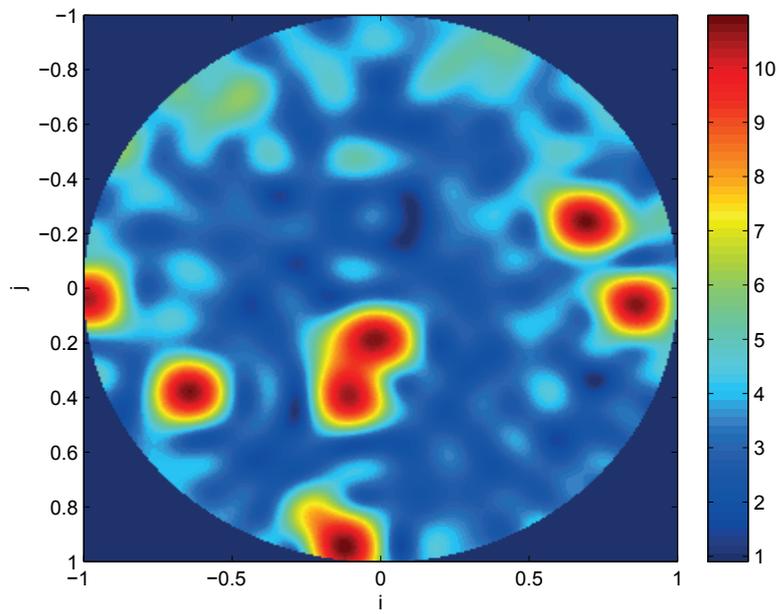


FIGURE 6.3 – Vue du ciel avec les interférences

En l'occurrence, nous en observons sept, dont trois proches des sources. Ce cas est d'autant plus intéressant que les perturbations sont proches des sources, angulairement parlant. Notre objectif est donc d'estimer au mieux la matrice de mélange des interférences, afin de les éliminer des observations, et ainsi avoir une bonne connaissance des sources cosmiques.

L'élimination peut se faire de plusieurs manières. Soit nous projetons les données sur le sous-espace orthogonal au sous-espace lié au interférences, soit nous soustrayons leur influence sur la matrice de corrélation pour  $\tau = 0$ .

Les paramètres sont décrits dans le Tableau 6.1 avec :

- Nbr de pts : Nombre de points,
- Nbr de mat. : Nombre de matrices (égal au nombre de décalages  $\tau$  considérés),
- RIB (dB) : Rapport Interférences sur Bruit en dB,
- Nbr it : Nombre d'itérations retenues pour les algorithmes,
- Nbr poll. : Nombre de pollueurs (interférences),
- MC : Nombre de réalisations de MC,
- Figure : Numéro de la figure

Nbr de pts	Nbr de mat.	RIB (dB)	Nbr it	Nbr poll.	MC	Figure
$2^{12}$	10	10	1 :150	7	100	6.4
$2^{14}$	10	20	1 :150	10	100	6.5
$2^{14}$	10	0 :20	150	10	100	6.6
$2^7 :2^{14}$	10	0	150	8	100	6.7
$2^7 :2^{14}$	10	20	150	10	100	6.8
$2^{14}$	10	10	150	2 :10	100	6.9
$2^{14}$	10	20	150	2 :10	100	6.10

TABLE 6.1 – Paramètres de simulations pour les données simulés LOFAR

De plus, le réseau sera systématiquement de type LBA, composé de 48 antennes, réparties régulièrement de  $d = \frac{1}{2}\lambda$ , et les vues du ciel seront construites de telle sorte que le niveau de couleur représentera la puissance des sources éventuelles.

### 6.2.2.1 Comparaison des vitesses de convergence

La Figure 6.4 et la Figure 6.5 présentent les vitesses de convergence de nos trois méthodes (la courbe rouge, présentant la performance d'estimation obtenue par Décomposition en Valeurs Singulières, n'est pas itérative, et c'est pour cela que la pente est nulle). On remarque que NOODLES converge respectivement plus rapidement que l'ALS (pourtant amélioré avec l'ELS) et DESERT .

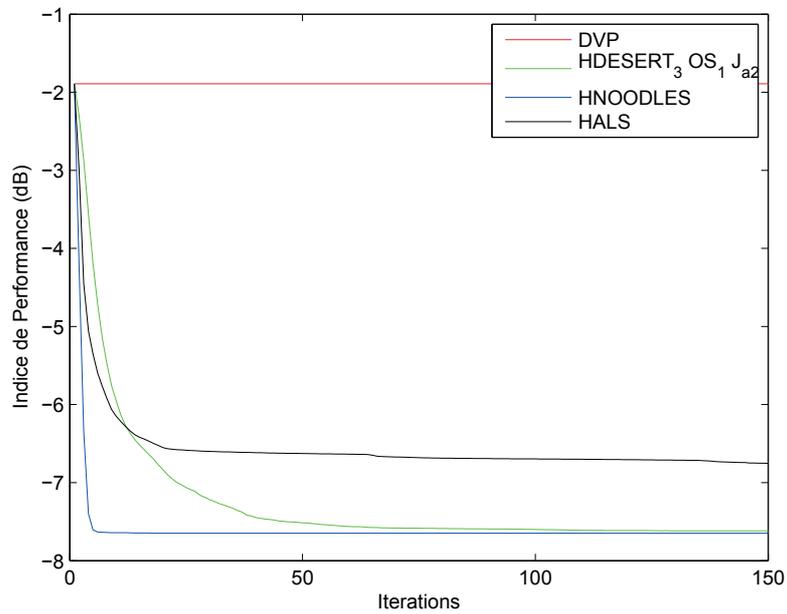


FIGURE 6.4 – Vitesse de convergence des algorithmes

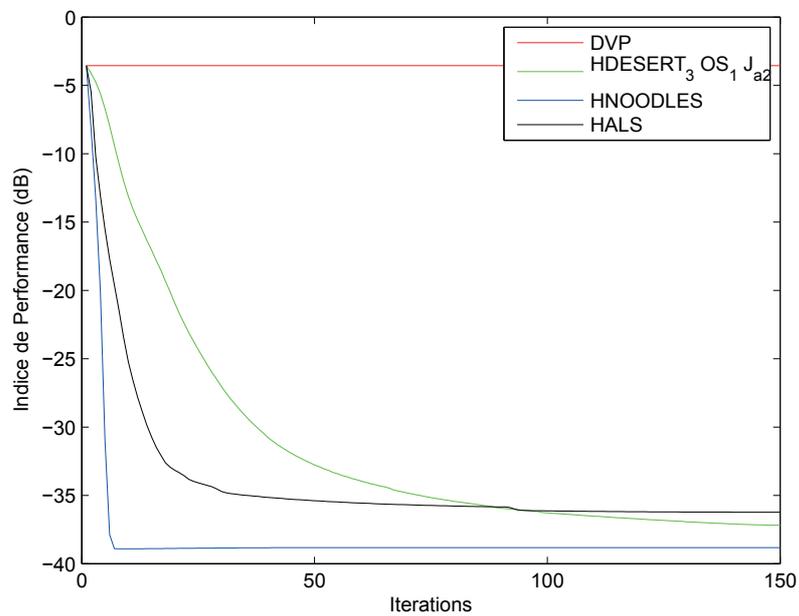


FIGURE 6.5 – Vitesse de convergence des algorithmes

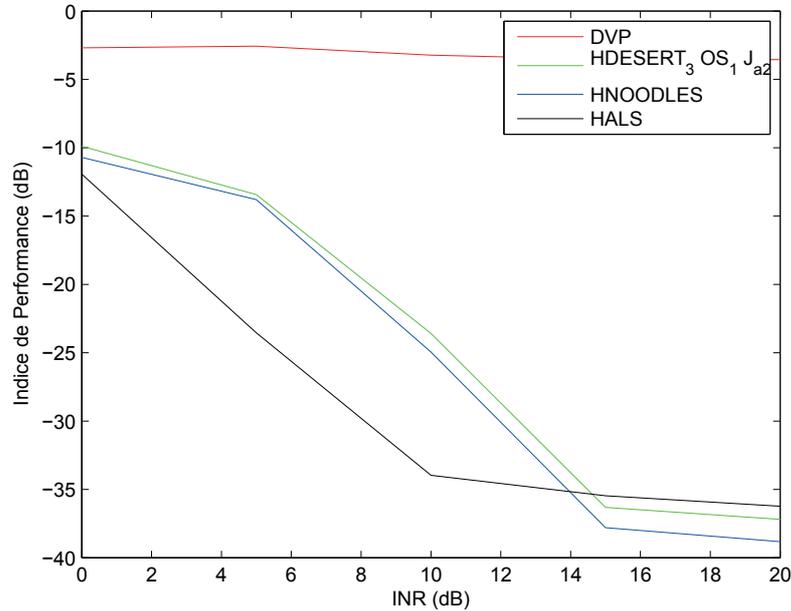


FIGURE 6.6 – Performance des algorithmes en fonction du RIB

### 6.2.2.2 Influence de la puissance des interférences

Les résultats illustrés dans la Figure 6.6 attestent du comportement de nos trois méthodes. On remarque que pour de faibles RIB, l'ALS fournit des résultats plus satisfaisants que les autres méthodes. Ceci est essentiellement dû au fait que cette méthode nécessite des hypothèses de travail moins fortes que DESERT et NOODLES (surtout en ce qui concerne le fait d'avoir des matrices strictement diagonales). De plus, on remarque que DESERT fournit, dans ce scénario, de moins bons résultats en termes de performances, mais il met également plus de temps à converger.

### 6.2.2.3 Influence du nombre d'échantillons

La Figure 6.7 et la Figure 6.8 donnent l'influence du nombre d'échantillons sur les performances d'estimations. On remarque que, évidemment, plus le nombre de points est important, plus l'estimation est précise. Cependant, on remarque également qu'en fonction des autres paramètres (en l'occurrence le nombre de pollueurs ainsi que le RIB), les performances obtenues sont différentes, l'ALS étant plus efficace que NOODLES et DESERT dans un cas où les contraintes sont plus fortes (cf Figure 6.7).

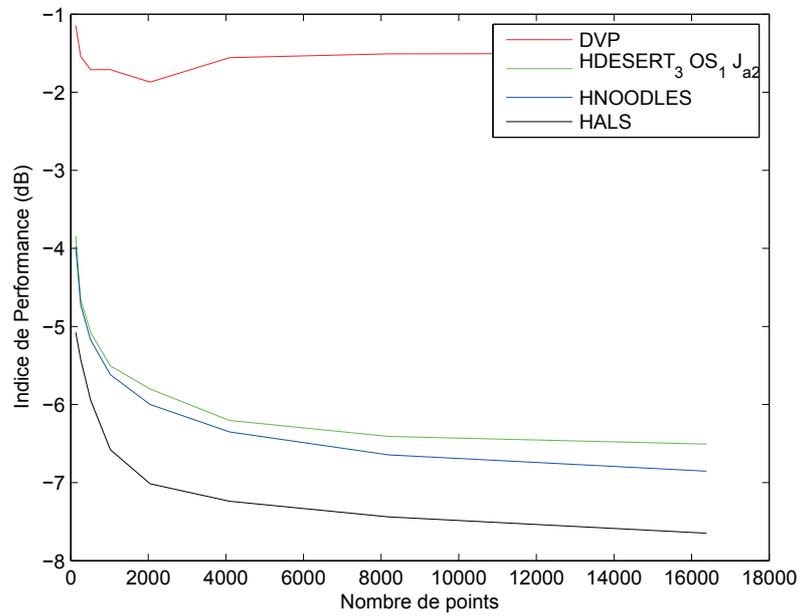


FIGURE 6.7 – Performance des algorithmes en fonction du nombre d'échantillons

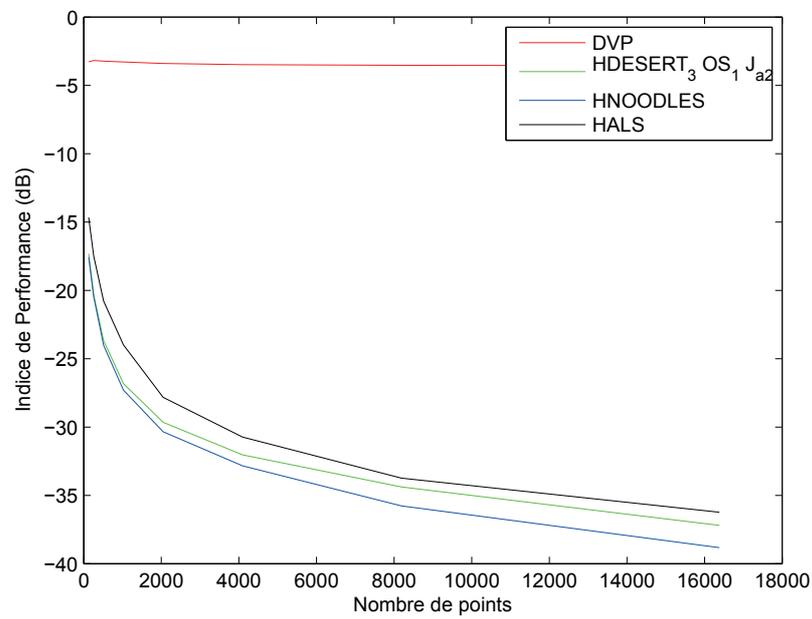


FIGURE 6.8 – Performance des algorithmes en fonction du nombre d'échantillons

### 6.2.2.4 Influence du nombre de pollueurs

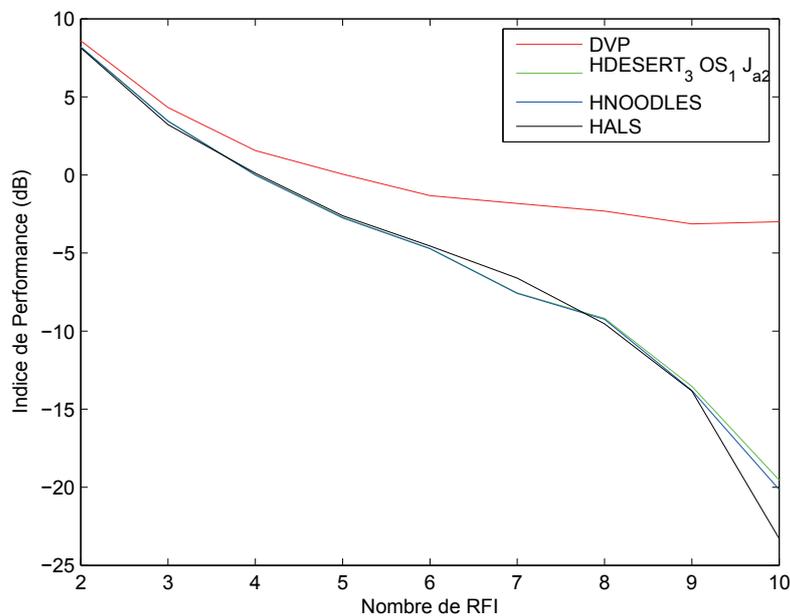


FIGURE 6.9 – Performance des algorithmes en fonction du nombre de pollueurs

La Figure 6.9 et la Figure 6.10 permettent de voir l'influence de la dimension du sous-espace interférences sur les performances. Il est à noter que cette dimension est strictement égale au nombre de pollueurs effectivement présents dans le mélange. Plusieurs remarques peuvent être établies. Tout d'abord, plus le nombre de pollueurs est grand, plus l'estimation est bonne. Il est quand même bon de noter que ceci reste vrai tant que la dimension de celui-ci reste inférieure à la dimension du sous-espace bruit. On notera également une légère différence entre les deux situations. En effet, on remarque une amélioration de la performance donnée par l'ALS pour un RIB de 10dB en présence de dix pollueurs. Sinon, nous obtenons, le reste du temps, sensiblement les mêmes performances entre les trois méthodes

### 6.2.2.5 Autres résultats

La Figure 6.11 présente les résultats obtenus après projection sur le sous-espace orthogonal au sous-espace interférences. Le fait intéressant à noter est que bien que les interférences soient "proches" des sources cosmiques, celles-ci n'ont pas été éliminées de l'image.

La Figure 6.12 illustre ce que l'on obtient après soustraction de la contribution des pollueurs. On remarque plusieurs différences entre projection et soustraction. Tout d'abord, la projection élimine très précisément dans une direction, et cette précision diminue au fur et à mesure. On

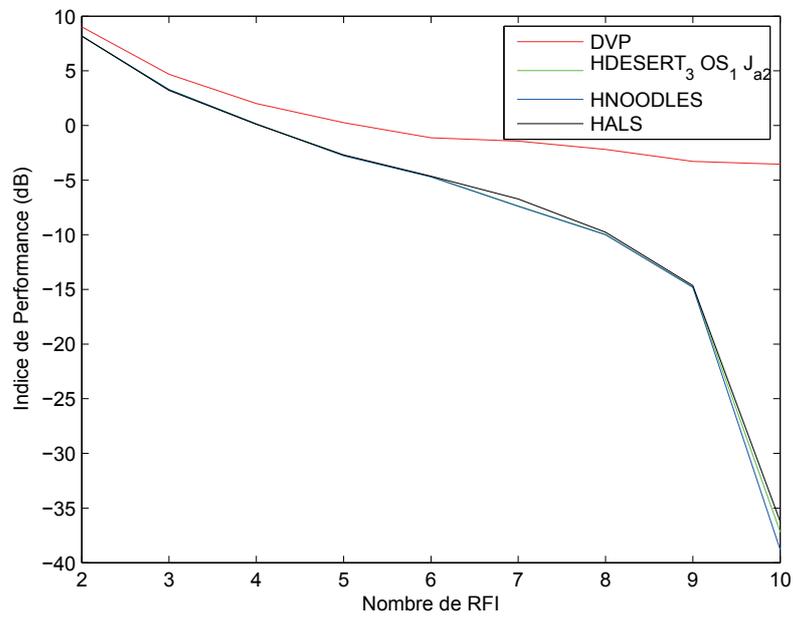


FIGURE 6.10 – Performance des algorithmes en fonction du nombre de pollueurs

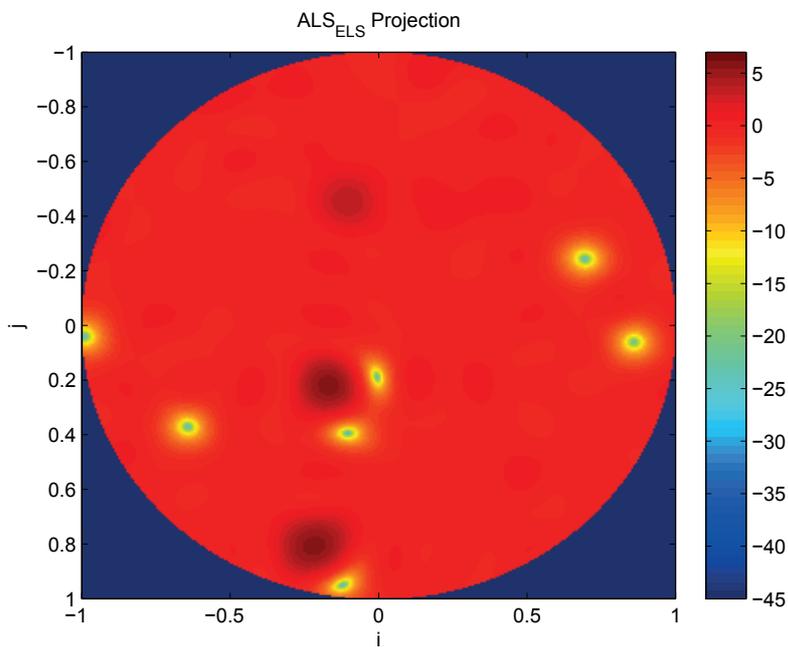


FIGURE 6.11 – Vue du ciel après projection

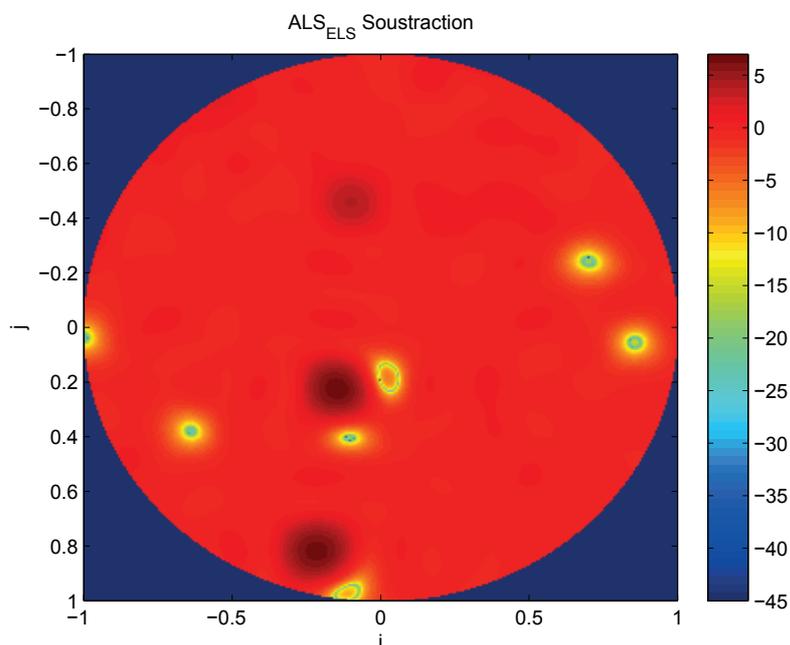


FIGURE 6.12 – Vue du ciel après soustraction

l'observe clairement dans la Figure 6.11, où on a un "trou" ponctuel (de couleur bleue) à la position de l'interférence, et la suppression décroît rapidement. Dans le cas de la soustraction, on ne remarque pas de suppression en un point comme pour la projection mais la suppression est plus homogène sur toute la zone d'influence de la perturbation.

### 6.3 Données expérimentales

Dans la suite, nous allons voir si les méthodes que nous avons présentées ci-dessus donnent toujours des résultats satisfaisants sur des données expérimentales. La campagne de mesure s'est déroulée en février 2010, et les signaux ont été reçus sur 48 LBA.

Comme le signal est étalé sur une grande plage fréquentielle, et que les perturbations sont nombreuses, nous allons nous focaliser sur deux bandes de fréquence. La première sera centrée sur 55MHz, fréquence utilisée par un réseau mobile terrestre ("land mobile" en anglais), la seconde sur 27.3MHz, correspondant à un "pager". Ensuite, nous allons séparer le réseau d'antennes en deux. Les antennes "paires" et les antennes "impaires", qui correspondent à deux polarisations différentes d'un seul et même réseau.

### 6.3.1 Justification empirique de l'utilisation d'une matrice aléatoire comme point initial

Avant d'illustrer tous les résultats que nous avons pu obtenir sur les données, nous allons expliquer les raisons qui nous ont poussé à utiliser une matrice aléatoire comme point initial.

Considérons tout d'abord la bande de fréquence "land mobile". Sur les antennes impaires, on obtient la vue du ciel illustrée en Figure 6.13.

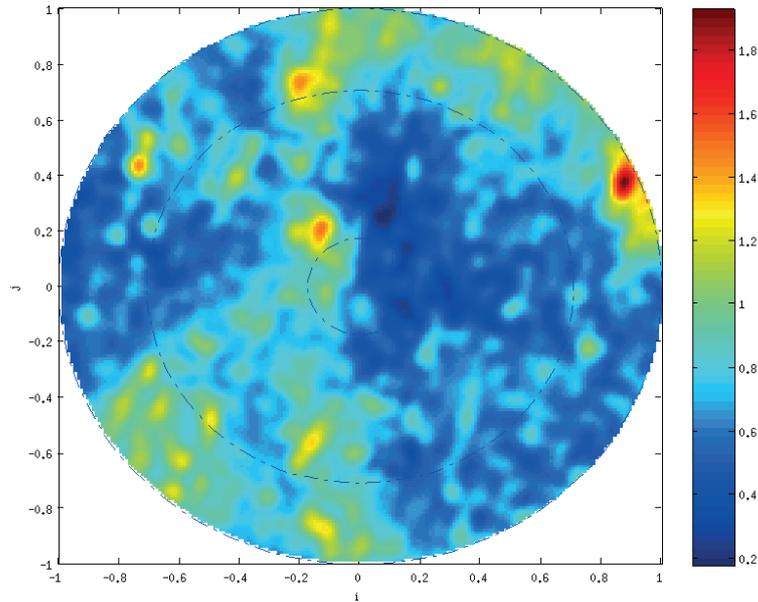


FIGURE 6.13 – Vue du ciel sur les antennes impaires. Scénario "land mobile"

On observe alors plusieurs sources (les coordonnées correspondent à  $(i,j)$  dans l'image) :

- Deux perturbations en  $(-0.7,0.4)$  et  $(0.9,0.4)$ , respectivement dénommées PA et PB dans la suite,
- Cassiopée A en  $(-0.2,0.7)$  dénommée CasA
- Cygnus A en  $(-0.2,0.2)$ , appelée CygA
- La voie lactée autour de  $(-0.1,0.8)$ , abrégée en MW.

Tout d'abord, nous allons essayer de voir le comportement de l'algorithme ALS dans le cas où l'on considère qu'il n'y a qu'une seule interférence. Nous utilisons comme point de départ à la méthode le vecteur propre associé à la plus grande valeur propre, afin de comparer avec la méthode usuelle. Puis, nous prenons le vecteur propre associé à la deuxième plus grande valeur propre. Les résultats sont visibles en Figure 6.14.

En comparant les résultats obtenus, on peut remarquer qu'en prenant un seul vecteur propre,

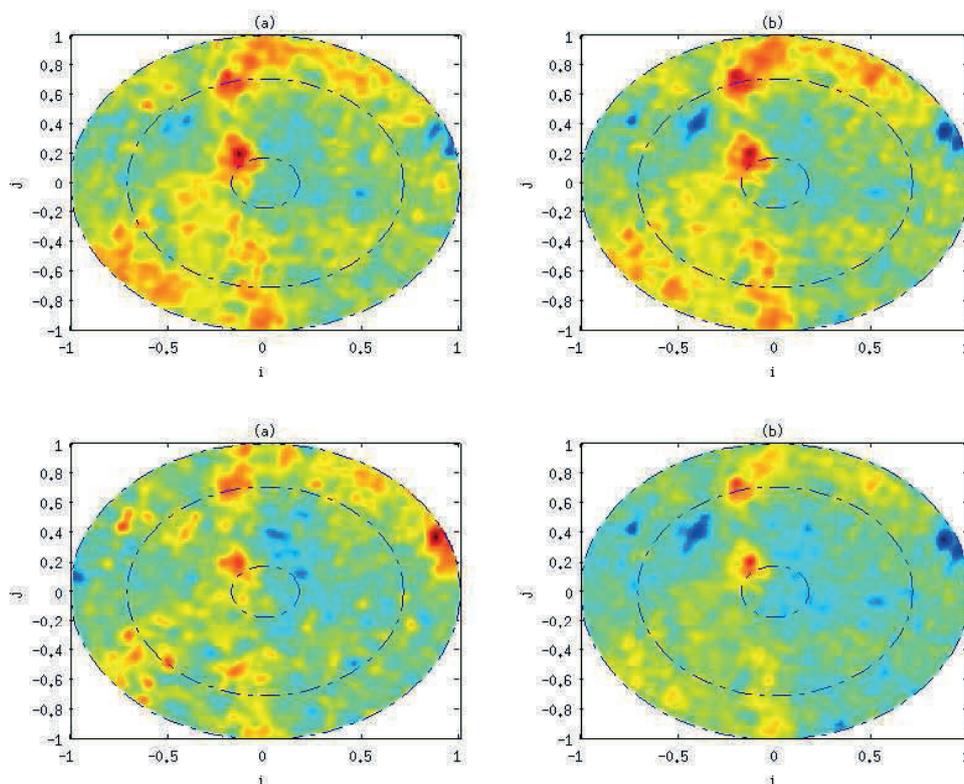


FIGURE 6.14 – (a) : DVP avec VecP associé à plus grande ValP, (b) : ALS avec VecP associé à plus grande ValP, (c) : DVP avec VecP associé à 2nd plus grande ValP, (d) : ALS avec VecP associé à 2nd plus grande ValP

qui de plus est associé à la plus grande valeur propre, l'ALS apporte une légère amélioration, en terme de finesse. En revanche, lorsqu'on prend le vecteur propre associé à la deuxième plus grande valeur propre, on remarque que l'ALS, malgré tout, évolue afin d'éliminer l'interférence la plus puissante. Cela présente un avantage certain : quelque soit l'entrée considérée pour initialiser l'ALS, la méthode semble toujours converger vers une solution acceptable, en l'occurrence l'élimination de la perturbation la plus puissante.

De plus, afin d'éviter de tomber dans un minimum local, nous n'initialiserons pas la méthode ALS avec les vecteurs propres associés aux plus grandes valeurs propres dans la suite, mais avec une matrice proche de l'identité. De plus, comme nous ne nous intéressons qu'à la matrice de mélange, nous ne présenterons que des résultats utilisant la méthode ALS, puisque les algorithmes DESERT et NOODLES ne concerne principalement que la matrice de séparation.

### 6.3.2 Justification empirique de l'utilisation des cumulants

Nous allons montrer ici pourquoi nous utilisons des ensembles créés à partir de statistiques d'ordre supérieurs afin d'estimer la matrice de mélange. Comme cela a été développé dans le chapitre 5, les matrices de corrélation ainsi que les cumulants d'ordre quatre ont la même structure, c'est-à-dire la matrice de mélange  $\mathbf{A}_r$  multipliée à droite par une matrice (celle-ci est par contre différente si on considère des statistiques à l'ordre deux ou quatre). La Figure 6.15 présente les différences que l'on obtient sur la vue du ciel. Sur cet exemple, on observe que PB est éliminée plus efficacement, mais on observe aussi à côté de cette interférence qu'une autre interférence est détectée, et elle est aussi mieux éliminée. PA est également éliminée plus finement. Ainsi, dans la suite, nous n'estimerons la matrice de mélange des interférences qu'à partir d'ensembles constitués à partir de cumulants d'ordre quatre.

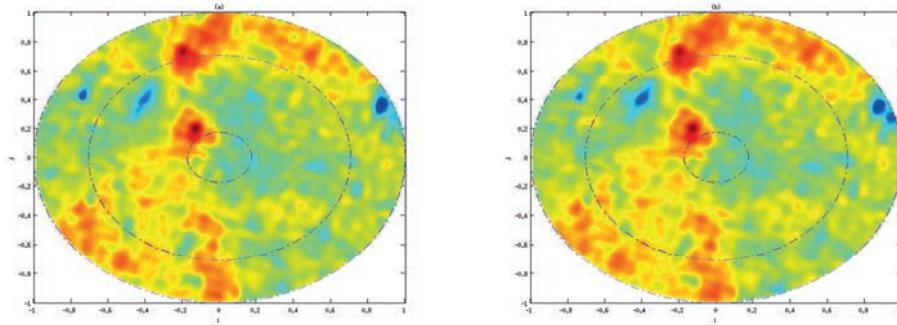
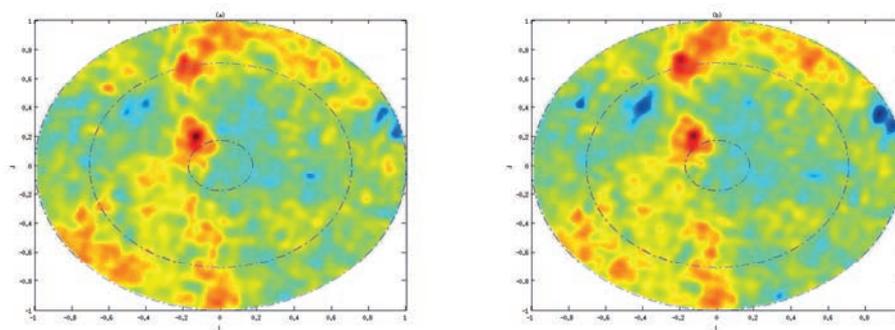
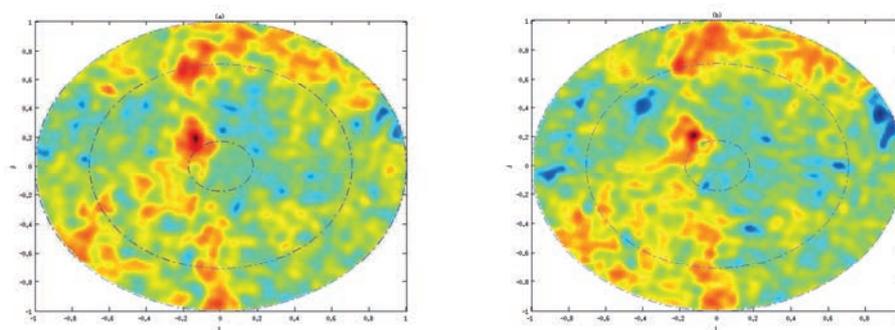


FIGURE 6.15 – Influence de l'estimation de la matrice de mélange, en fonction des statistiques utilisées. (a) : Ordre 2, (b) : Ordre 4

### 6.3.3 Traitements pour le scénario "land mobile"

Nous allons ici présenter les résultats obtenus avec la méthode ALS sur les signaux issus de la bande de fréquence autour de la perturbation correspondant au réseau terrestre mobile. Nous comparerons nos résultats avec ce qui se fait classiquement ; la projection des données sur le sous-espace issu de la décomposition en valeurs propres de la matrice considérée. Observons maintenant l'influence de la dimension du sous-espace interférence sur les données à l'aide de la Figure 6.16 et de la Figure 6.17, dans le cas où on considère les antennes uniquement impaires.

On remarque que l'élimination de PA et PB est bien meilleure en utilisant l'ALS pour un sous-espace interférence de dimension un. De plus, on remarque que la puissance de CasA est conservée avec l'ALS, tandis qu'avec la méthode basée sur la décomposition en valeurs propres, la présence de cette source d'intérêt est réduite. La considération d'un sous-espace interférences de dimension deux ramène à des résultats plus proches, entre les deux méthodes. En effet, les

FIGURE 6.16 –  $\dim(\text{SEI}) = 1$  pour (a) : DVP et (b) : ALSFIGURE 6.17 –  $\dim(\text{SEI}) = 2$  pour (a) : DVP et (b) : ALS

deux ont une incidence sur CasA et éliminent correctement PB. La différence se fait sur PA et les deux autres sources cosmiques.

Regardons maintenant la vue du ciel que l'on obtient dans le cas où on considère seulement les antennes paires (Figure 6.18)

Voyons également l'influence des traitements sur les données, illustrée sur la Figure 6.19 et la Figure 6.20. On observe :

- une interférence PA de coordonnées  $(-0.7, 0.4)$ ,
- une interférence PB de coordonnées  $(0.9, 0.4)$ ,
- une interférence PC de coordonnées  $(-0.4, 0.9)$
- les trois sources cosmiques CasA, CygA et MW toujours à la même place.

On remarque sur la Figure 6.19 et sur la Figure 6.20 que les choses diffèrent par rapport à ce que l'on a pu observer sur les antennes impaires. Tout d'abord, en considérant un sous-espace interférence de dimension un, la méthode utilisant la décomposition en valeurs propres n'élimine absolument pas les interférences, contrairement à la méthode ALS qui en réduit considérablement

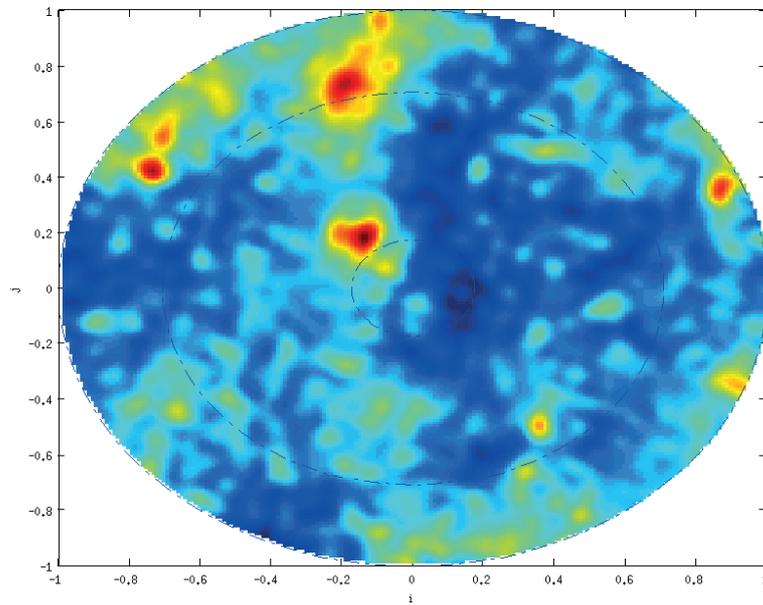
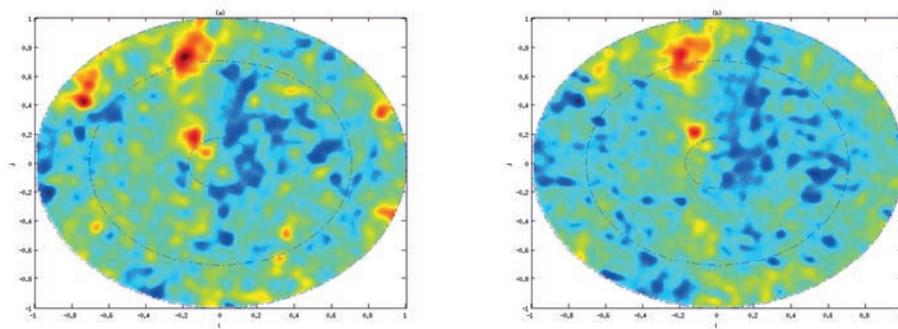
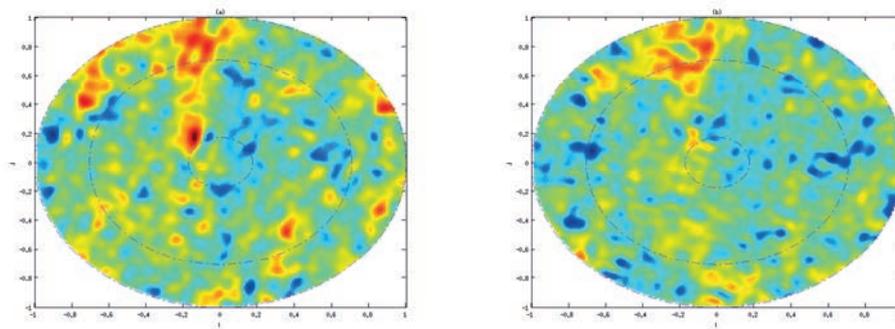


FIGURE 6.18 – Vue du ciel sur les antennes paires. Scénario "land mobile"

FIGURE 6.19 –  $\dim(\text{SEI}) = 1$  pour (a) : DVP et (b) : ALS

l'influence. De plus, en considérant un sous-espace interférences de dimension deux, on remarque que l'ALS élimine de manière plus précise les interférences, mais vient également réduire l'influence des sources cosmiques. La décomposition en valeurs propres, dans ce cas précis, n'apporte pas forcément de meilleurs résultats que si on avait considéré un sous-espace interférences de dimension un.

FIGURE 6.20 –  $\dim(\text{SEI}) = 2$  pour (a) : DVP et (b) : ALS

### 6.3.4 Traitements pour le scénario "pager"

Nous allons maintenant considérer une autre plage fréquentielle, dans laquelle se situe une interférence prédominante, dans le cas où ne nous prenons en compte que les antennes paires.

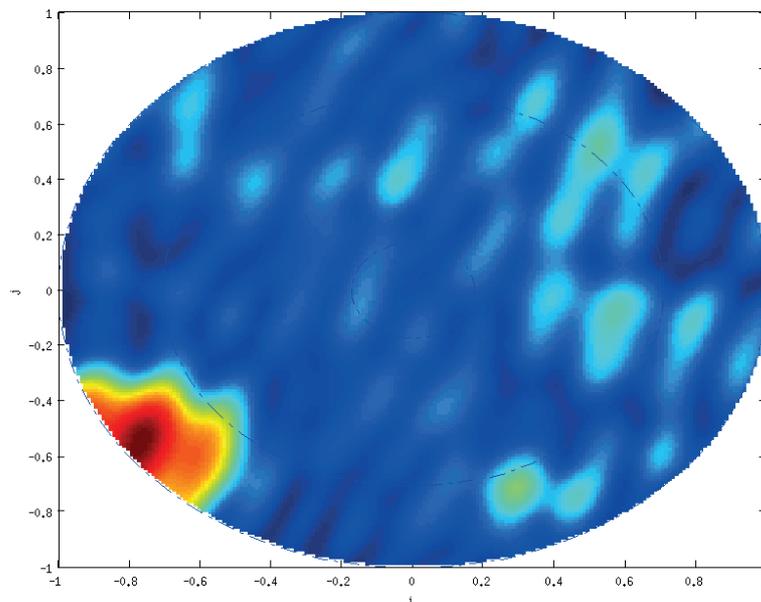


FIGURE 6.21 – Vue du ciel sur les antennes paires. Scénario "pager"

Contrairement au cas précédent, on remarque que l'interférence est bien plus puissante, ce qui fait que nous n'observons pas clairement CasA, CygA et MW. Lorsque nous commençons le traitement, le fait de prendre un sous-espace de dimension un pour l'ALS n'apporte aucune

amélioration par rapport à la méthode utilisant la décomposition en valeurs propres, tout comme un sous-espace interférences de dimension deux. La Figure 6.22 illustre ce fait.

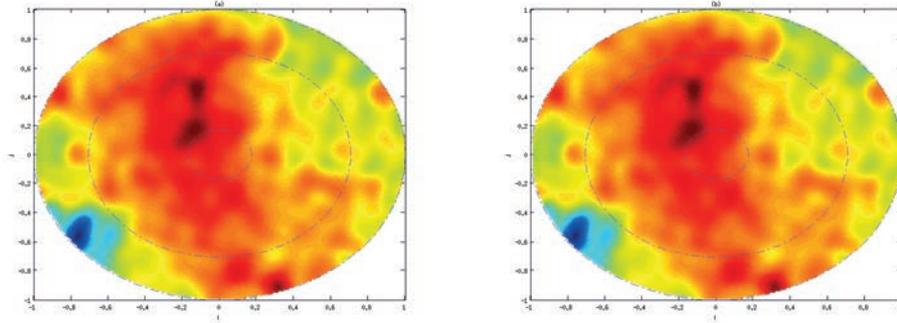


FIGURE 6.22 –  $\dim(\text{SEI}) = 1$  pour (a) : DVP et (b) : ALS

Néanmoins, la Figure 6.23 nous fournit un résultat intéressant, en considérant un sous-espace interférences de dimension trois : la méthode basée sur la décomposition en valeurs propres élimine en partie une des trois sources cosmiques (en l'occurrence CygA), contrairement à l'algorithme ALS. Ceci est également vrai pour un sous-espace de dimension quatre (Figure 6.24).

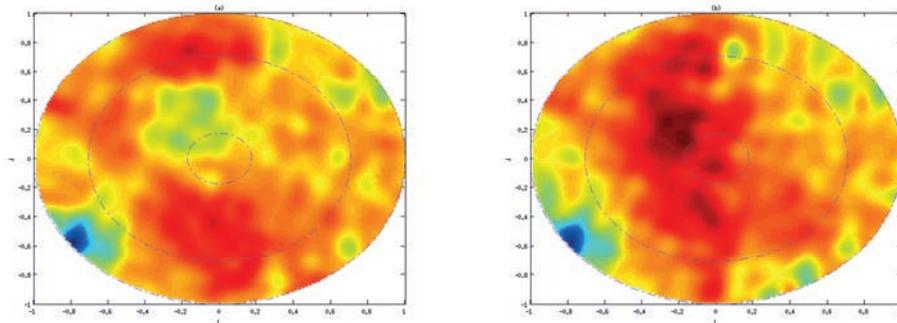
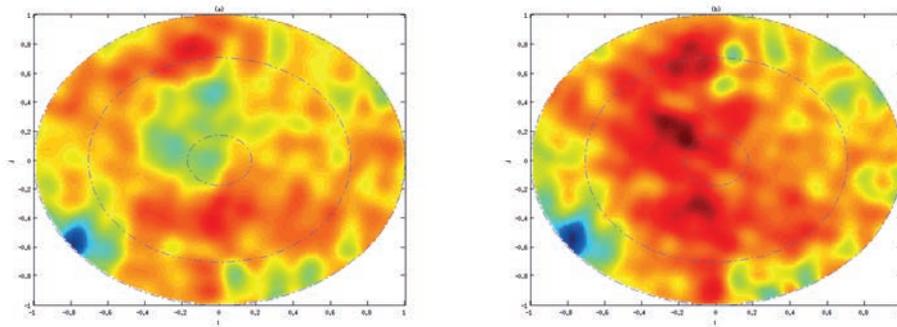
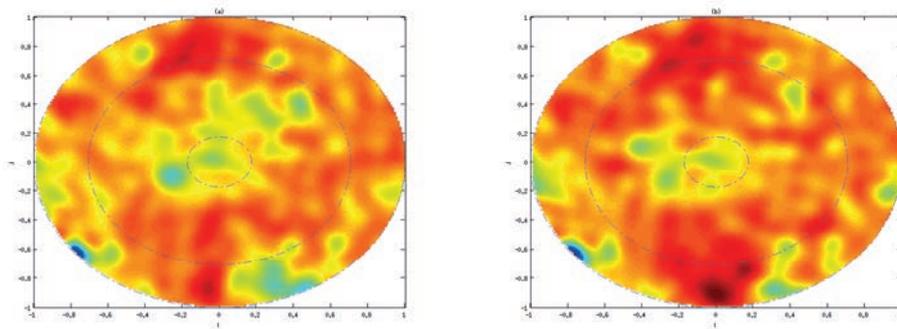
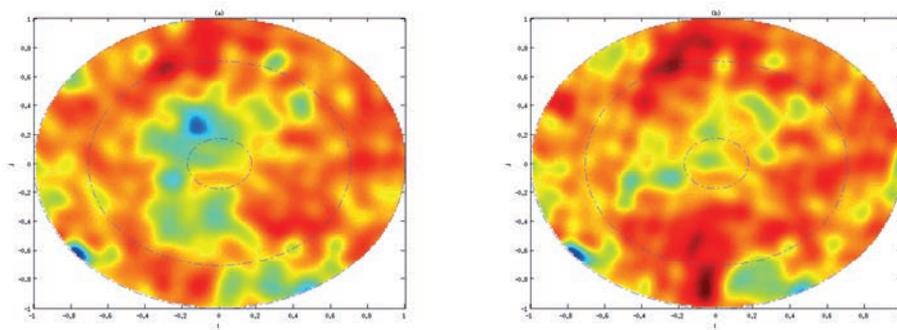


FIGURE 6.23 –  $\dim(\text{SEI}) = 3$  pour (a) : DVP et (b) : ALS

Concernant les antennes impaires, les résultats ne sont pas satisfaisants. En effet, pour un sous-espace interférence de dimension un, deux ou trois, nous obtenons des résultats similaires. Puis pour un sous-espace de dimension quatre, l'influence de CygA est réduite (Figure 6.25) en utilisant les deux méthodes, puis est complètement supprimée en prenant un sous-espace de dimension cinq (Figure 6.26). Le seul avantage, dans ce cas là, que l'on peut voir pour la méthode ALS, est de ne pas altérer significativement CasA et MW.

FIGURE 6.24 –  $\dim(\text{SEI}) = 4$  pour (a) : DVP et (b) : ALSFIGURE 6.25 –  $\dim(\text{SEI}) = 4$  pour (a) : DVP et (b) : ALSFIGURE 6.26 –  $\dim(\text{SEI}) = 5$  pour (a) : DVP et (b) : ALS

# Chapitre 7

## Conclusions et Perspectives

### 7.1 Conclusions

Les travaux menés s'inscrivent dans un cadre de traitement du signal multidimensionnel. Dans un tel cadre, pour des signaux complexes, l'information statistique peut alors facilement se représenter sous une forme matricielle. On rencontre alors principalement deux types de matrices : les matrices hermitiennes et les matrices symétriques complexes conduisant respectivement à deux types de décompositions permettant de les diagonaliser. Le but principal des travaux a été de proposer des algorithmes de diagonalisation de matrices complexes pouvant s'adapter au deux types de matrices conjointement. Cela nous a permis entre autre de généraliser des approches existantes tout en en proposant de nouvelles.

Nous avons à la fois considéré des algorithmes de type gradient et des algorithmes algébriques, à chaque fois dans un souci de simplicité algorithmique. En ce qui concerne les algorithmes de type gradient, nous les avons placé dans un cadre commun en explicitant différents niveaux d'approximation du critère quadratique classique de mesure de distance à une matrice diagonale. En ce qui concerne les algorithmes algébriques, nous nous sommes principalement inspirés de ce qui ce fait dans le cadre des décompositions tensorielles.

Ces algorithmes de diagonalisation conjointe trouvent une application directe en séparation de sources lorsque celles ci sont statistiquement indépendantes. Cette application nous a permis de tester les performances des algorithmes proposés et de considérer une application en radioastronomie où les signaux de radiocommunication terrestre sont vus comme des interférences.

### 7.2 Perspectives

Les développements présentés dans ce manuscrit ont été fait dans le cadre de la diagonalisation conjointe de matrices. Il serait judicieux de voir si cela peut s'appliquer dans le cas où

les statistiques ne seraient plus représentées sous forme matricielle, mais sous forme tensorielle directement. En ce qui concerne l'application à la séparation de sources, nous ne nous sommes intéressés qu'à des situations où les mélanges sont linéaires et instantanés. Il serait aussi intéressant d'aborder le cas de mélanges convolutifs.

En ce qui concerne la décomposition des matrices, le seul type que nous avons considéré concerne les matrices diagonales. L'ensemble des algorithmes présentés peuvent se généraliser assez facilement à des décompositions de type bloc diagonales.

L'application principale considérée a été la séparation de sources. Cependant, le domaine du traitement d'antennes nécessite aussi des décompositions matricielles. Il serait aussi certainement intéressant d'étudier de manière systématique l'apport des algorithmes de diagonalisation conjointe, comme ceux développés dans cette thèse, à la problématique du traitement d'antennes. L'application à la radioastronomie présente des résultats dans un contexte très simplifié. Il serait intéressant d'étudier la réjection des interférences par soustraction plutôt que par projection. Il serait aussi intéressant d'aborder les signaux de radiocommunication terrestre par une modélisation considérant les aspects circulaires ou non et les aspects cyclostationnaires par l'intermédiaire de statistiques d'ordre supérieur à deux.

## Annexe A

# Développement des calculs des algorithmes de gradient

### A.1 Calcul du gradient

#### A.1.1 Cas hermitien

Le gradient de (3.7) formulé en (3.8) s'obtient comme suit. Réécrivons tout d'abord le critère (3.7) avec (1.7) :

$$\mathcal{J}_h(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( (\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^H \right)^H \mathbf{Z} \text{Diag} \{ (\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^H \} \right\}$$

Développons le produit matriciel suivant :

$$(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^H = \mathbf{T}_i + \mathbf{Z} \mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H + \mathbf{Z} \mathbf{T}_i \mathbf{Z}^H$$

On calcule maintenant les dérivées partielles de tous les termes, et on ne conserve ensuite que les termes composés de  $\mathbf{Z}^*$  et  $\mathbf{Z}^H$ , les autres dérivées partielles n'intervenant pas dans le calcul du

gradient :

$$\begin{aligned}
& \partial \text{tr}\{(\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\}\} \\
& \partial \text{tr}\{(\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}\} \\
& \partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i)^H \text{ZDiag}\{\mathbf{T}_i + \mathbf{Z} \mathbf{T}_i\}\} \\
& \partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{T}_i + \mathbf{Z} \mathbf{T}_i\}\} \\
& \partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i)^H \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\}\} \\
& \partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i)^H \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}\} \\
& \partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\}\} \\
& \partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}\}
\end{aligned}$$

En appliquant la propriété (1.3) au premier des termes ci-dessus, puis la propriété (1.4), on a :

$$\begin{aligned}
\text{tr}\{(\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H)^H \partial \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\}\} &= \text{tr}\{\text{ZDiag}\{\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H\}^H \mathbf{T}_i \partial \mathbf{Z}^H\} \\
&= \text{tr}\{\partial \mathbf{Z}^H \text{ZDiag}\{\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H\}^H \mathbf{T}_i\}
\end{aligned}$$

Enfin, la définition (1.10) relative à la dérivation donne :

$$\frac{\partial \text{tr}\{\mathbf{Z}^H \text{ZDiag}\{\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H\}^H \mathbf{T}_i\}}{\partial \mathbf{Z}^*} = \text{ZDiag}\{\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H\}^H \mathbf{T}_i$$

En appliquant ce schéma à tous les termes à dériver, on arrive à :

$$\begin{aligned}
\frac{\partial \text{tr}\{(\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{T}_i + \mathbf{T}_i \mathbf{Z}^H\}^H \mathbf{Z} \mathbf{T}_i \\
\frac{\partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i)^H \text{ZDiag}\{\mathbf{T}_i + \mathbf{Z} \mathbf{T}_i\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{T}_i + \mathbf{Z} \mathbf{T}_i\} \mathbf{T}_i^H \\
\frac{\partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{T}_i + \mathbf{Z} \mathbf{T}_i\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{T}_i + \mathbf{Z} \mathbf{T}_i\} \mathbf{Z} \mathbf{T}_i^H \\
\frac{\partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i)^H \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\} \mathbf{T}_i^H + \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i\}^H \mathbf{T}_i \\
\frac{\partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i)^H \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\} \mathbf{T}_i^H + \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i\}^H \mathbf{Z} \mathbf{T}_i \\
\frac{\partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{T}_i \mathbf{Z}^H\} \mathbf{Z} \mathbf{T}_i^H + \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}^H \mathbf{T}_i \\
\frac{\partial \text{tr}\{(\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H)^H \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\} \mathbf{Z} \mathbf{T}_i^H + \text{ZDiag}\{\mathbf{Z} \mathbf{T}_i \mathbf{Z}^H\}^H \mathbf{Z} \mathbf{T}_i
\end{aligned}$$

et on retrouve alors tous les termes qui composent le gradient du critère (3.7).

### A.1.2 Cas symétrique

Le gradient de (3.9) formulé en (3.10) s'obtient comme suit. Réécrivons tout d'abord le critère (3.9) avec (1.7) :

$$\mathcal{J}_t(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( (\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^T \right)^H \text{ZDiag}\{(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^T\} \right\}$$

Développons le produit matriciel suivant :

$$(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^T = \mathbf{T}_i + \mathbf{Z}\mathbf{T}_i + \mathbf{T}_i\mathbf{Z}^T + \mathbf{Z}\mathbf{T}_i\mathbf{Z}^T$$

Dans la suite, on posera  $(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^T = \mathbf{S}_i$  par souci de clarté. On ne conserve ensuite que les termes composés de  $\mathbf{Z}^*$  et  $\mathbf{Z}^H$ , les autres dérivées partielles n'intervenant pas dans le calcul du gradient :

$$\begin{aligned} & \partial \text{tr}\{(\mathbf{Z}\mathbf{T}_i)^H \text{ZDiag}\{\mathbf{S}_i\}\} \\ & \partial \text{tr}\{(\mathbf{T}_i\mathbf{Z}^T)^H \text{ZDiag}\{\mathbf{S}_i\}\} \\ & \partial \text{tr}\{(\mathbf{Z}\mathbf{T}_i\mathbf{Z}^T)^H \text{ZDiag}\{\mathbf{S}_i\}\} \end{aligned}$$

En appliquant les propriétés (1.3), (1.4) et (1.10) aux termes ci-dessus, on a :

$$\begin{aligned} \frac{\partial \text{tr}\{(\mathbf{Z}\mathbf{T}_i)^H \text{ZDiag}\{\mathbf{S}_i\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{S}_i\} \mathbf{T}_i^H \\ \frac{\partial \text{tr}\{(\mathbf{T}_i\mathbf{Z}^T)^H \text{ZDiag}\{\mathbf{S}_i\}\}}{\partial \mathbf{Z}^*} &= (\mathbf{T}_i^H \text{ZDiag}\{\mathbf{S}_i\})^T \\ \frac{\partial \text{tr}\{(\mathbf{Z}\mathbf{T}_i\mathbf{Z}^T)^H \text{ZDiag}\{\mathbf{S}_i\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{S}_i\} \mathbf{Z}^* \mathbf{T}_i^H + \text{ZDiag}\{\mathbf{S}_i\}^T \mathbf{Z}^* \mathbf{T}_i^* \end{aligned}$$

De plus, les matrices  $\mathbf{T}_i$  étant symétriques, on a  $\mathbf{T}_i^H = \mathbf{T}_i^*$  et les matrices  $\mathbf{S}_i$  sont aussi symétriques. On obtient finalement :

$$\begin{aligned} \frac{\partial \text{tr}\{(\mathbf{Z}\mathbf{T}_i)^H \text{ZDiag}\{\mathbf{S}_i\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{S}_i\} \mathbf{T}_i^* \\ \frac{\partial \text{tr}\{(\mathbf{T}_i\mathbf{Z}^T)^H \text{ZDiag}\{\mathbf{S}_i\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{S}_i\} \mathbf{T}_i^* \\ \frac{\partial \text{tr}\{(\mathbf{Z}\mathbf{T}_i\mathbf{Z}^T)^H \text{ZDiag}\{\mathbf{S}_i\}\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\{\mathbf{S}_i\} \mathbf{Z}^* \mathbf{T}_i^* + \text{ZDiag}\{\mathbf{S}_i\} \mathbf{Z}^* \mathbf{T}_i^* \end{aligned}$$

On retrouve alors tous les termes qui composent le gradient du critère (3.9).

## A.2 Calcul du gradient pour un critère simplifié

### A.2.1 Cas hermitien

Nous développons ici le calcul du gradient du critère (3.21), dont le résultat est donné en (3.23). En appliquant (1.7) à cette fonction de coût, on a :

$$\mathcal{J}_{a,h}(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \} \right\}$$

On conserve ensuite tous les termes ayant une dérivé partielle en  $\mathbf{Z}^*$  et  $\mathbf{Z}^H$  :

$$\begin{aligned} & \partial \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(2)}\mathbf{Z}^H \} \right\} \\ & \partial \text{tr} \left\{ \left( \mathbf{Z}\mathbf{T}_i^{(2)} \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} \} \right\} \\ & \partial \text{tr} \left\{ \left( \mathbf{Z}\mathbf{T}_i^{(2)} \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(2)}\mathbf{Z}^H \} \right\} \end{aligned}$$

En appliquant les propriétés (1.3), (1.4) et (1.10) aux termes ci-dessus, on a :

$$\begin{aligned} \frac{\partial \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(2)}\mathbf{Z}^H \} \right\}}{\partial \mathbf{Z}^*} &= \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(1)} + \mathbf{T}_i^{(2)}\mathbf{Z}^H \}^H \mathbf{T}_i^{(2)} \\ \frac{\partial \text{tr} \left\{ \left( \mathbf{Z}\mathbf{T}_i^{(2)} \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} \} \right\}}{\partial \mathbf{Z}^*} &= \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} \} \left( \mathbf{T}_i^{(2)} \right)^H \\ \frac{\partial \text{tr} \left\{ \left( \mathbf{Z}\mathbf{T}_i^{(2)} \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(2)}\mathbf{Z}^H \} \right\}}{\partial \mathbf{Z}^*} &= \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(2)}\mathbf{Z}^H \} \left( \mathbf{T}_i^{(2)} \right)^H + \mathbf{Z} \text{Diag} \{ \mathbf{Z}\mathbf{T}_i^{(2)} \}^H \mathbf{T}_i^{(2)} \end{aligned} \tag{A.1}$$

On retrouve alors tous les termes qui composent le gradient du critère (3.21). Pour avoir la formulation du gradient de la fonction de coût (3.22), il suffit de ne pas considérer les termes qui comportent un  $\mathbf{Z}$  dans (A.1).

### A.2.2 Cas symétrique

Le calcul du gradient de (3.25) et de (3.26) est développé dans cette partie. Reprenons tout d'abord le critère (3.25) et réécrivons le en utilisant (1.7) :

$$\mathcal{J}_{a,s}(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T \right)^H \mathbf{Z} \text{Diag} \{ \mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T \} \right\}$$

Seuls les termes ayant une dérivé partielle en  $\mathbf{Z}^*$  et  $\mathbf{Z}^H$  sont conservés :

$$\begin{aligned} & \partial \text{tr}\left\{\left(\mathbf{Z}\mathbf{T}_i^{(2)}\right)^H \text{ZDiag}\left\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T\right\}\right\} \\ & \partial \text{tr}\left\{\left(\mathbf{T}_i^{(2)}\mathbf{Z}^T\right)^H \text{ZDiag}\left\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T\right\}\right\} \end{aligned}$$

L'applications des propriétés (1.3), (1.4) et (1.10) aux termes ci-dessus donne :

$$\begin{aligned} \frac{\partial \text{tr}\left\{\left(\mathbf{Z}\mathbf{T}_i^{(2)}\right)^H \text{ZDiag}\left\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T\right\}\right\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\left\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T\right\} \left(\mathbf{T}_i^{(2)}\right)^H \\ \frac{\partial \text{tr}\left\{\left(\mathbf{T}_i^{(2)}\mathbf{Z}^T\right)^H \text{ZDiag}\left\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T\right\}\right\}}{\partial \mathbf{Z}^*} &= \text{ZDiag}\left\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^T\right\}^T \left(\mathbf{T}_i^{(2)}\right)^* \end{aligned}$$

Ces termes sont bien ceux qui composent le gradient de (3.25). De plus, en ne considérant que les termes indépendants de  $\mathbf{Z}$ , on retrouve les éléments qui permettent de calculer le gradient du critère (3.26).

## Annexe B

# Développement des calculs pour les coefficients des pas optimaux

### B.1 Calcul du pas optimal pour le critère initial

Reprenons le critère défini en (3.5) :

$$\mathcal{J}(\mathbf{Z}) = \sum_{i=1}^N \|\mathbf{Z} \text{Diag}\{(\mathbf{I} + \mathbf{Z}) \mathbf{T}_i (\mathbf{I} + \mathbf{Z})^\dagger\}\|^2 \quad (\text{B.1})$$

et utilisons la mise à jour de  $\mathbf{Z}$  énoncée en (3.3)

$$\mathbf{Z} = -\mu \nabla_r(\mathbf{B}) = \mu \mathbf{F} \quad (\text{B.2})$$

On introduit alors (B.2) dans (B.1), et on obtient :

$$\mathcal{J}(\mathbf{Z}) = \sum_{i=1}^N \|\mathbf{Z} \text{Diag}\{(\mathbf{I} + \mu \mathbf{F}) \mathbf{T}_i (\mathbf{I} + \mu \mathbf{F})^\dagger\}\|^2 \quad (\text{B.3})$$

Utilisons alors la définition de la norme de Frobenius (1.7), ainsi que (1.3) à l'équation ci-dessus. Il vient :

$$\mathcal{J}(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( (\mathbf{I} + \mu \mathbf{F}) \mathbf{T}_i (\mathbf{I} + \mu \mathbf{F})^\dagger \right)^H \mathbf{Z} \text{Diag}\{(\mathbf{I} + \mu \mathbf{F}) \mathbf{T}_i (\mathbf{I} + \mu \mathbf{F})^\dagger\} \right\} \quad (\text{B.4})$$

Le développement de  $(\mathbf{I} + \mu\mathbf{F}) \mathbf{T}_i (\mathbf{I} + \mu\mathbf{F})^\dagger$  nous donne un polynôme d'ordre deux en  $\mu$  :

$$(\mathbf{I} + \mu\mathbf{F}) \mathbf{T}_i (\mathbf{I} + \mu\mathbf{F})^\dagger = \mathbf{T}_i + \mu \left( \mathbf{F}\mathbf{T}_i + \mathbf{T}_i\mathbf{F}^\dagger \right) + \mu^2 \mathbf{F}\mathbf{T}_i\mathbf{F}^\dagger \quad (\text{B.5})$$

Il ne nous reste plus qu'à développer le produit en argument de la fonction trace, et nous obtenons alors le polynôme de degré quatre en  $\mu$  (3.11), dont les coefficients sont les suivants :

$$\begin{aligned} a_{i,0} &= \mathbf{T}_i^H \text{ZDiag}\{\mathbf{T}_i\} \\ a_{i,1} &= \mathbf{T}_i^H \text{ZDiag}\{\mathbf{F}\mathbf{T}_i + \mathbf{T}_i\mathbf{F}^\dagger\} + \left( \mathbf{F}\mathbf{T}_i + \mathbf{T}_i\mathbf{F}^\dagger \right)^H \text{ZDiag}\{\mathbf{T}_i\} \\ a_{i,2} &= \mathbf{T}_i^H \text{ZDiag}\{\mathbf{F}\mathbf{T}_i\mathbf{F}^\dagger\} + \left( \mathbf{F}\mathbf{T}_i\mathbf{F}^\dagger \right)^H \text{ZDiag}\{\mathbf{T}_i\} + \left( \mathbf{F}\mathbf{T}_i + \mathbf{T}_i\mathbf{F}^\dagger \right)^H \text{ZDiag}\{\mathbf{F}\mathbf{T}_i + \mathbf{T}_i\mathbf{F}^\dagger\} \\ a_{i,3} &= \left( \mathbf{F}\mathbf{T}_i\mathbf{F}^\dagger \right)^H \text{ZDiag}\{\mathbf{F}\mathbf{T}_i + \mathbf{T}_i\mathbf{F}^\dagger\} + \left( \mathbf{F}\mathbf{T}_i + \mathbf{T}_i\mathbf{F}^\dagger \right)^H \text{ZDiag}\{\mathbf{F}\mathbf{T}_i\mathbf{F}^\dagger\} \\ a_{i,4} &= \left( \mathbf{F}\mathbf{T}_i\mathbf{F}^\dagger \right)^H \text{ZDiag}\{\mathbf{F}\mathbf{T}_i\mathbf{F}^\dagger\} \end{aligned}$$

## B.2 Calcul du pas optimal dans le cas d'un critère simplifié

Une fois qu'une grande partie des approximations ont été faites sur les matrices qui composent le critère (3.5), on se retrouve, dans un premier temps, avec la fonction de coût vue en (3.18).

$$\mathcal{J}_a(\mathbf{Z}) = \sum_{i=1}^N \|\text{ZDiag}\{\mathbf{T}_i^{(1)} + \mathbf{Z}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{Z}^\dagger\}\|^2 \quad (\text{B.6})$$

Il ne reste alors qu'à utiliser le principe vu pour le gradient exact. On remplace  $\mathbf{Z}$  par (B.2) dans (B.6), et on obtient :

$$\mathcal{J}_a(\mathbf{Z}) = \sum_{i=1}^N \|\text{ZDiag}\{\mathbf{T}_i^{(1)} + \mu\mathbf{F}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mu\mathbf{F}^\dagger\}\|^2 \quad (\text{B.7})$$

On utilise alors la définition (1.7), ainsi que (1.3), dans l'équation ci-dessus, et on arrive à :

$$\mathcal{J}_a(\mathbf{Z}) = \sum_{i=1}^N \text{tr} \left\{ \left( \mathbf{T}_i^{(1)} + \mu\mathbf{F}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mu\mathbf{F}^\dagger \right)^H \text{ZDiag}\{\mathbf{T}_i^{(1)} + \mu\mathbf{F}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mu\mathbf{F}^\dagger\} \right\} \quad (\text{B.8})$$

Le développement du produit en argument de la fonction trace nous donne alors le polynôme

de degré deux en  $\mu$  (3.29), dont les coefficients sont détaillés ci-dessous :

$$\begin{aligned} b_{i,0} &= \mathbf{T}_i^{(1)H} \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\} \\ b_{i,1} &= \mathbf{T}_i^{(1)H} \text{ZDiag} \left\{ \mathbf{F}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{F}^\dagger \right\} + \left( \mathbf{F}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{F}^\dagger \right)^H \text{ZDiag} \left\{ \mathbf{T}_i^{(1)} \right\} \\ b_{i,2} &= \left( \mathbf{F}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{F}^\dagger \right)^H \text{ZDiag} \left\{ \mathbf{F}\mathbf{T}_i^{(2)} + \mathbf{T}_i^{(2)}\mathbf{F}^\dagger \right\} \end{aligned}$$

En ce qui concerne le critère (3.20), on remarque vite que la différence provient de l'approximation en  $\mathbf{Z}$  dans le développement du produit en argument de la fonction trace dans (B.8). Ainsi donc, on se retrouve avec seulement un polynôme de degré un en  $\mu$ , exprimée en (3.30), et dont les coefficients sont ceux ne contenant pas de double produit en  $\mathbf{Z}$ , soit  $b_{i,0}$  et  $b_{i,1}$ .

### B.3 Calcul des coefficients pour l'ELS

Reprenons le critère défini en (4.44) :

$$\mathcal{C}^{(n)}(\mathbf{A}, \mathbf{D}_i) = \sum_{i=1}^N \|\mathbf{A}^{(n)} \mathbf{D}_i^{(n)} \mathbf{C}^{(n)} - \mathbf{M}_i\|^2 \quad (\text{B.9})$$

Comme cela a été énoncé précédemment, nous pouvons mettre à jour les matrices qui nous intéressent de la manière suivante :

$$\begin{aligned} \mathbf{A}^{(n)} &= \mathbf{A}^{(n-2)} + \mu \left( \mathbf{A}^{(n_1)} - \mathbf{A}^{(n-2)} \right) = \mathbf{A}^{(n-2)} + \mu \boldsymbol{\Theta}_A \\ \mathbf{C}^{(n)} &= \mathbf{C}^{(n-2)} + \mu \left( \mathbf{C}^{(n_1)} - \mathbf{C}^{(n-2)} \right) = \mathbf{C}^{(n-2)} + \mu \boldsymbol{\Theta}_C \\ \mathbf{D}_i^{(n)} &= \mathbf{D}_i^{(n-2)} + \mu \left( \mathbf{D}_i^{(n_1)} - \mathbf{D}_i^{(n-2)} \right) = \mathbf{D}_i^{(n-2)} + \mu \boldsymbol{\Theta}_{D_i} \end{aligned} \quad (\text{B.10})$$

dont les coefficients à l'itération  $n$  estimés à partir des coefficients à l'itération  $n-1$  et  $n-2$ . Si nous remplaçons  $\mathbf{A}^{(n)}$ ,  $\mathbf{C}^{(n)}$  et  $\mathbf{D}_i^{(n)}$  dans (B.9), on obtient :

$$\mathcal{C}^{(n)}(\mathbf{A}, \mathbf{D}_i) = \sum_{i=1}^N \left\| \left( \mathbf{A}^{(n-2)} + \mu \boldsymbol{\Theta}_A \right) \left( \mathbf{D}_i^{(n-2)} + \mu \boldsymbol{\Theta}_{D_i} \right) \left( \mathbf{C}^{(n-2)} + \mu \boldsymbol{\Theta}_C \right) - \mathbf{M}_i \right\|^2 \quad (\text{B.11})$$

Le développement du produit matriciel donne un polynôme de degré trois :

$$\begin{aligned}
 \mathbf{A}^{(n)} \mathbf{D}_i^{(n)} \mathbf{C}^{(n)} - \mathbf{M}_i &= \mathbf{A}^{(n-2)} \mathbf{D}_i^{(n-2)} \mathbf{C}^{(n-2)} - \mathbf{M}_i \\
 &+ \mu \left( \mathbf{A}^{(n-2)} \mathbf{D}_i^{(n-2)} \Theta_C + (\mathbf{A}^{(n-2)} \Theta_{D_i} \mathbf{C}^{(n-2)} + \Theta_A \mathbf{D}_i^{(n-2)} \mathbf{C}^{(n-2)}) \right) \\
 &+ \mu^2 \left( \mathbf{A}^{(n-2)} \Theta_{D_i} \Theta_C + \Theta_A \mathbf{D}_i^{(n-2)} \Theta_C + \Theta_A \Theta_{D_i} \mathbf{C}^{(n-2)} \right) \\
 &+ \mu^3 (\Theta_A \Theta_{D_i} \Theta_C) \\
 &= \mathbf{Z}_{i,0} + \mu \mathbf{Z}_{i,1} + \mu^2 \mathbf{Z}_{i,2} + \mu^3 \mathbf{Z}_{i,3}
 \end{aligned} \tag{B.12}$$

En utilisant la propriété (1.3), le critère  $\mathcal{C}^{(n)}(\mathbf{A}, \mathbf{D}_i)$  s'écrit alors :

$$\mathcal{C}^{(n)}(\mathbf{A}, \mathbf{D}_i) = \sum_{i=1}^N \text{tr} \left\{ (\mathbf{Z}_{i,0} + \mu \mathbf{Z}_{i,1} + \mu^2 \mathbf{Z}_{i,2} + \mu^3 \mathbf{Z}_{i,3})^H (\mathbf{Z}_{i,0} + \mu \mathbf{Z}_{i,1} + \mu^2 \mathbf{Z}_{i,2} + \mu^3 \mathbf{Z}_{i,3}) \right\} \tag{B.13}$$

qui peut également s'écrire sous la forme d'un polynôme de degré six :

$$\mathcal{C}^{(n)}(\mathbf{A}, \mathbf{D}_i) = \sum_{i=1}^N e_{i,0} + \mu e_{i,1} + \mu^2 e_{i,2} + \mu^3 e_{i,3} + \mu^4 e_{i,4} + \mu^5 e_{i,5} + \mu^6 e_{i,6} \tag{B.14}$$

et dont les coefficients  $e_{i,j} \forall j = 0, \dots, 6$  sont :

$$\begin{aligned}
 e_{i,0} &= \text{tr} \{ \mathbf{Z}_{i,0}^H \mathbf{Z}_{i,0} \} \\
 e_{i,1} &= \text{tr} \{ \mathbf{Z}_{i,1}^H \mathbf{Z}_{i,0} + \mathbf{Z}_{i,0}^H \mathbf{Z}_{i,1} \} \\
 e_{i,2} &= \text{tr} \{ \mathbf{Z}_{i,2}^H \mathbf{Z}_{i,0} + \mathbf{Z}_{i,0}^H \mathbf{Z}_{i,2} + \mathbf{Z}_{i,1}^H \mathbf{Z}_{i,1} \} \\
 e_{i,3} &= \text{tr} \{ \mathbf{Z}_{i,3}^H \mathbf{Z}_{i,0} + \mathbf{Z}_{i,0}^H \mathbf{Z}_{i,3} + \mathbf{Z}_{i,2}^H \mathbf{Z}_{i,1} + \mathbf{Z}_{i,1}^H \mathbf{Z}_{i,2} \} \\
 e_{i,4} &= \text{tr} \{ \mathbf{Z}_{i,3}^H \mathbf{Z}_{i,1} + \mathbf{Z}_{i,1}^H \mathbf{Z}_{i,3} + \mathbf{Z}_{i,2}^H \mathbf{Z}_{i,2} \} \\
 e_{i,5} &= \text{tr} \{ \mathbf{Z}_{i,6}^H \mathbf{Z}_{i,5} + \mathbf{Z}_{i,5}^H \mathbf{Z}_{i,6} \} \\
 e_{i,6} &= \text{tr} \{ \mathbf{Z}_{i,6}^H \mathbf{Z}_{i,6} \}
 \end{aligned}$$

# Bibliographie

- [1] K. Abed-Meraim, A. Belouchrani, J.-F. Cardoso, and E. Moulines. Asymptotic performance of second order blind source separation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 277–280, 1994.
- [2] B. Afsari. Simple lu and qr base non-orthogonal matrix joint diagonalization. In *Proc. Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA)*, 2006.
- [3] L. Albera, A. Ferreol, and P. Chevalier. Sixth order blind identification of underdetermined mixtures (birth) of sources. In *Proc. ICA*, Nara, Japan, April 2003.
- [4] S.-I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2) :251–276, Feb. 1998.
- [5] Astron. Modes d’observation. <http://www.astron.nl/radio-observatory/astromers/users/technical-information/major-observing-modes/major-observing-mode>.
- [6] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2) :434 – 444, 1997.
- [7] A. Belouchrani and M.G. Amin. Blind source separation based on time-frequency signal representations. *IEEE Transactions on Signal Processing*, 46 :2888–2897, 1998.
- [8] A.-J. Boonstra, R. Weber, and P. Colom. Rfi mitigation strategies for phased-array ska concepts. In *Proc. of the SKADS Conference*, pages 211–214, Belgium, 2009.
- [9] R. Bro. *Multi-way Analysis in the Food Industry : Models, Algorithms, and Applications*. PhD thesis, University of Amsterdam, 1998.
- [10] J.-F. Cardoso. Source separation using higher order moments. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2109–2112, 1989.
- [11] J.-F. Cardoso. Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3109–3112, 1991.

- [12] J.-F. Cardoso. Fourth-order cumulant structure forcing. application to blind array processing. In *Proc. IEEE SP Workshop*, pages 136 – 139, 1992.
- [13] J.-F. Cardoso. Iterative techniques for blind source separation using only fourth order cumulants. In *Proc. European Signal Processing Conf. (EUSIPCO)*, pages 739–742, 1992.
- [14] J.-F. Cardoso. On the performance of orthogonal source separation algorithms. In *Proc. European Signal Processing Conf. (EUSIPCO)*, pages 776–779, 1994.
- [15] J.-F. Cardoso. The invariant approach to source separation. In *Proc. NOLTA*, pages 55–60, 1995.
- [16] J.-F. Cardoso. Blind signal separation : statistical principles. *Proc. of the IEEE*, 90(8) :2009–2026, Oct. 1998. special issue on blind identification and estimation.
- [17] J.-F. Cardoso. Learning in manifolds : the case of source separation. In *Proc. SSAP*, 1998.
- [18] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non gaussian signal. *IEE Proceedings-F*, 140 :362 – 370, 1993.
- [19] J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal of Matrix Analysis and Applications*, 17(1) :161–164, Jan. 1996.
- [20] M. Castella, P. Bianchi, A. Chevreuil, and J.-C. Pesquet. A blind source separation framework for detecting cpm sources mixed by a convolutive mimo filter. *Signal Processing*, 86 :1950–1967, 2006.
- [21] G. Chabriel, J. Barrere, N. Thirion-Moreau, and E. Moreau. Algebraic joint zero-diagonalization and blind sources separation. *IEEE Transactions on Signal Processing*, 56(3) :980–989, March 2008.
- [22] G. Chabriel and J. Barrère. A direct algorithm for nonorthogonal approximate joint diagonalization. *IEEE Transactions on Signal Processing*, 60(1) :39–47, January 2012.
- [23] P. Chevalier, L. Albera, P. Comon, and A. Ferreol. Comparative performance analysis of eight blind source separation methods on radiocommunications signals. In *Proc. International Joint Conference on Neural Networks*, Budapest, Hungary, July 2004.
- [24] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36 :287 – 314, 1994.
- [25] P. Comon, X. Luciani, and L.F. Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23 :393 – 405, August 2009.
- [26] Y.H. Dai. A new gradient method with an optimal stepsize property. *Computational Optimization and Applications*, 33 :73 – 88, 2006.
- [27] J. Delabrouille, J.-F. Cardoso, and G. Patanchon. Multi-detector multi-component spectral matching and applications for cmb data analysis. *Monthly Notices of the Royal Astronomical Society*, 346 :1089–1102, 2003.

- [28] N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources : a deflation approach. *Signal Processing*, 45 :59–83, 1995.
- [29] A.-J. Van der Veen. Joint diagonalization via subspace fitting techniques. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 2773 – 2776, 2001.
- [30] R. Dubroca, C. De Luigi, M. Castella, and E. Moreau. A general algebraic algorithm for blind extraction of one source in a mimo convolutive mixture. *IEEE Transactions on Signal Processing*, 58(5) :2484–2493, May 2010.
- [31] R. Dubroca, C. De Luigi, and E. Moreau. Adapted deflation approach for referenced contrast optimization in blind mimo convolutive source separation. In *Proc. ICA*, pages 243–250, 2009.
- [32] E. M. Fadaili, N. Thirion-Moreau, and E. Moreau. Nonorthogonal joint diagonalization/zero diagonalization for source separation based on time-frequency distributions. *IEEE Transactions on Signal Processing*, 55(5) :1673 – 1687, May 2007.
- [33] R. Feliachi, R. Weber, and A.-J. Boonstra. Cyclic spatial filtering in radio astronomy : application to lofar data. In *Proc. European Signal Processing Conf. (EUSIPCO)*, Glasgow, Aug. 2009.
- [34] C. A. R. Fernandes, G. Favier, and J. Mota. Parafac-based channel estimation and data recovery in nonlinear mimo spread spectrum communication systems. *Signal Processing*, 91(2) :311–322, 2011.
- [35] C. E. R. Fernandes, G. Favier, and J. Mota. Blind channel identification algorithms based on the parafac decomposition of cumulant tensors : the single and multiuser cases. *Signal Processing*, 88(6) :1382–1401, 2008.
- [36] A. Ferreol, L. Albera, and P. Chevalier. Fourth order blind identification of underdetermined mixtures of sources (fobium). *IEEE Transactions on Signal Processing*, 53(5) :1640–1653, May 2005.
- [37] A. Ferreol, P. Chevalier, and L. Albera. Second-order blind separation of first and second-order cyclo-stationary sources. application to am, fsk, cpfsk, and deterministic sources. *IEEE Transactions on Signal Processing*, 52 :845–861, 2004.
- [38] M. Gailitis. Behaviour of the cross section near threshold of a new reaction in the case of a coulomb attraction field. *Soviet Physics - JETP*, 17 :1328, 1963.
- [39] H. Ghennioui, E. M. Fadaili, N. Thirion-Moreau, A. Adib, and E. Moreau. A non-unitary joint block diagonalization algorithm for blind separation of convolutive mixtures of sources. *IEEE Signal Processing Letters*, 15(1) :860 – 863, January 2008.

- [40] H. Ghennioui, N. Thirion-Moreau, E. Moreau, and D. Aboutajdine. Gradient-based joint block diagonalization algorithms : Application to blind separation of fir convolutive mixtures. *Signal Processing*, 90(6) :1836–1849, June 2010.
- [41] R.A. Harshman. Foundations of the parafac procedure : Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16 :1–84, 1970. University Microfilms, Ann Arbor, Michigan, No. 10,085.
- [42] J. Héroult, C. Jutten, and B. Ans. Detection de grandeurs primitives dans un message composite par une architecture de calcul neuromimetique en apprentissage non supervise. In *Proc. Xe colloque GRETSI*, volume 2, pages 1017–1022, Nice, France, May 1985.
- [43] M. Joho and H. Mathis. Joint diagonalization of correlation matrices by using gradient methods with application to blind signal separation. In *Proc. Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 273 – 277, August 2002.
- [44] C. Jutten and J. Héroult. Blind separation of sources : an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24 :1–10, 1991.
- [45] C. Labat and J. Idier. Convergence of conjugate gradient methods with a closed-form stepsize formula. *J. Optim. Theory Appl.*, 136 :43 – 60, 2008.
- [46] B. Laheld and J.-F. Cardoso. Adaptive source separation without pre-whitening. In *Proc. European Signal Processing Conf. (EUSIPCO)*, pages 183–186, Edinburgh, Scotland, Sept. 1994.
- [47] L. De Lathauwer, J. Castaing, and J.-F. Cardoso. Fourth-order cumulant based blind identification of underdetermined mixtures. *IEEE Transactions on Signal Processing*, 55(6) :2965–2973, June 2007.
- [48] L. De Lathauwer and B. De Moor. On the blind separation of non-circular sources. In *Proc. European Signal Processing Conf. (EUSIPCO)*, pages 99–102, 2002.
- [49] L. De Lathauwer, B. De Moor, and J. Vandewalle. Fetal electrocardiogram extraction by blind source subspace separation. *IEEE Trans. Biomedical Engineering*, 47(5) :567–572, May 2000.
- [50] X.-L. Li and T. Adali. Blind separation of noncircular correlated sources using gaussian entropy rate. *IEEE Transactions on Signal Processing*, 59(6) :2969–2975, June 2011.
- [51] X.-L. Li and X.-D. Zhang. Nonorthogonal joint diagonalization free of degenerate solution. *IEEE Transactions on Signal Processing*, 55(5) :1803 – 1814, May 2007.
- [52] A.-S. Montcuquet, L. Herve, J.-M. Dinten, and J. Mars. Non-negative matrix factorization : a blind sources separation method applied to optical fluorescence spectroscopy and multiplexing. In *Proc. of Biomedical Optics BIOS congress*, 2010.

- [53] A.-S. Montcuquet, L. Herve, F. Navarro, J.-M. Dinten, and J. Mars. Regularized non negative matrix factorization for autofluorescence removal in fluorescence optical imaging. In *Proc. of IEEE Workshop on Statistical Signal Processing (SSP2011)*, 2011.
- [54] E. Moreau. A generalization of joint-diagonalization criteria for source separation. *IEEE Transactions on Signal Processing*, 49(3) :530 – 541, March 2001.
- [55] E. Moreau. Comments on "cubica : Independent component analysis by simultaneous third and fourth order cumulant diagonalization". *IEEE Transactions on Signal Processing*, 54(12) :4826–4828, December 2006.
- [56] E. Moreau. Joint diagonalization of third order complex symmetric tensors and application to blind separation of non circular sources. In *Proc. Forty First Asilomar Conference on Signals, Systems and Computers (Asilomar'2007)*, Pacific Grove, California, USA,, November 2007.
- [57] E. Moreau and O. Macchi. A one stage self-adaptive algorithm for source separation. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 49–52, April 1994.
- [58] E. Moreau and O. Macchi. High order contrasts for self-adaptive source separation. *International Journal of Adaptive Control and Signal Processing*, 10(1) :19–46, January 1996.
- [59] S. Moussaoui, D. Brie, C. Carteret, and A. Mohammad-Djafari. Application of bayesian non-negative source separation to mixture analysis in spectroscopy. In *Proc. Int. Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, 2004.
- [60] D. Muti and S. Bourennane. Multiway filtering based on fourth-order cumulants. *EURASIP Journal on Applied Signal Processing*, 7 :1147–1158, 2005.
- [61] D. Nion and L. De Lathauwer. An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma. *Elsevier Signal Processing*, 88(3) :749–755, March 2008.
- [62] C. Paulus and J.I. Mars. New multicomponent filters for geophysical data processing. *IEEE Geosciences and Remote sensing*, 44(8) :2260–2270, 2006.
- [63] D. T. Pham. Joint approximate diagonalization of positive definite hermitian matrices. *SIAM J. Matrix Anal. Appl.*, 22(4) :1136 – 1152, 2001.
- [64] B. Picinbono. On circularity. *IEEE Transactions on Signal Processing*, 42(12) :3473 – 3482, December 1994.
- [65] M. Rajih, P. Comon, and R. Harshman. Enhanced line search : A novel method to accelerate parafac. *SIAM J. Matrix Anal. Appl.*, 30(3) :1128 – 1147, 2008.
- [66] N. Renard and S. Bourennane. Improvement of target detection methods by multiway filtering. *IEEE Trans. on Geoscience and Remote Sensing*, 46(8) :2407–2417, August 2008.

- [67] M.J. Seaton. Diagonalisation of complex symmetric matrices using a modified jacobi method. *The Computer Journal*, 12(2) :156–157, 1969.
- [68] A. Souloumiac. Nonorthogonal joint diagonalization by combining givens and hyperbolic rotations. *IEEE Transactions on Signal Processing*, 57(6) :2222–2231, June 2009.
- [69] M. Sørensen, S. Icart, P. Comon, and L. Deneire. Gradient based approximate joint diagonalization by orthogonal transforms. In *Proc. European Signal Processing Conf. (EUSIPCO)*, 2008.
- [70] P. Tichavsky and A. Yeredor. Fast approximate joint diagonalization incorporating weight matrices. *IEEE Transactions on Signal Processing*, 57(3) :878 – 891, March 2009.
- [71] E. Trefftz. Overlapping metastable levels in dielectronic recombination. *Zeits. f. Astrophys.*, 65 :299, 1967.
- [72] V. Vrabie, J.I. Mars, and J.-L. Lacoume. Singular value decomposition by means of independent component analysis. *Signal Processing*, 84(3) :645–652, 2004.
- [73] A. J. Weiss and B. Friedlander. Array processing using joint diagonalization. *Signal Processing*, 50 :205 – 222, 1996.
- [74] S. Wijnholds and A.-J. Van der Veen. Multisource self-calibration for sensor arrays. *IEEE Transactions on Signal Processing*, 57(9) :3512–3522, Sept. 2009.
- [75] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7) :1545 – 1553, July 2002.
- [76] A. Yeredor, A. Ziehe, and K.-R. Müller. Approximate joint diagonalization using a natural gradient approach. In *Proc. Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA)*, volume 3195, pages 89 – 96, September 2004.
- [77] A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5 :777 – 800, 2004.