

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

المدرسة الوطنية المتعددة التقنيات
ECOLE NATIONALE POLYTECHNIQUE.



DEPARTEMENT D'ELECTRONIQUE

THESE
présentée par

Hicham BOUSBIA-SALAH
Ingénieur d'Etat en Electronique

En vue de l'obtention du diplôme de
DOCTORAT D'ETAT
En électronique

Thème

**SEPARATION DE SIGNAUX « SOURCES »
A PARTIR DE LEURS MELANGES CONVOLUTIFS :
TECHNIQUES DE BLOC DIAGONALISATION CONJOINTE
BASEE SUR LES STATISTIQUES D'ORDRE DEUX.**

Soutenue le 27 Mai 2006, devant le jury composé de :

Président :	Mr Ahmed ZERGUERRAS	Professeur (ENP)
Examineurs :	Mr Salah BOURENNANE	Professeur (EGIM-France)
	Mme Amina SERRIR	Maître de conférences (USTHB)
	Mr Mohamed TRABELSI	Maître de conférences (ENP)
Directeur de thèse :	Mr Adel BELOUHRANI	Maître de conférences (ENP)
Codirecteur de thèse :	Mr Karim ABED-MERAIM	Maître de conférences (ENST- France)

ENP, 10 avenue Hacène BADI – EL HARRACH, BP182 – 16200 ALGER

ملخص

يهتم هذا البحث بالفصل العصامي للإشارات و فك الالتفاف باستعمال خليط هاته الإشارات و إحصائياتها ذات الدرجة الثانية. صفة "عصامي" تعبّر عن عدم معرفتنا لأي معلومة عن قنوات الاتصال أو عن الإشارات ما عدى اختلاف إحصائيات هذه الأخيرة ذات الدرجة الثانية. تمّ انجاز تقنيات و برامج لجعل المصفوفات كتلية قطرية. هذه المصفوفات تمثل إحصائيات خليط إشارات مستقرّة و غير مستقرّة. من أجل فك كلي للالتفاف و استرجاع الإشارات الأصلية، اتخذنا طريقة عمل ذات مرحلتين: المرحلة الأولى تكمن في فصل الإشارات بجعل المصفوفات كتلية قطرية، المرحلة الثانية هي عبارة عن تسوية لقناة كلّ إشارة أصلية باستعمال تقنية الفضاء الجزئي مع تقدير درجة القناة. أخيراً، تمّ انجاز واجهة بيانية تمنح للمستعمل إمكانية تشغيل كلّ البرامج و مقارنة التقنية المقترحة في عملنا هذا مع التقنية التقليدية التي تسوي القناة أولاً ثمّ تفصل ما بين الإشارات.

مفاتيح: الفصل العصامي للإشارات؛ فك الالتفاف العصامي؛ إحصائيات ذات الدرجة الثانية؛ كتلية قطرية.

Résumé

Cette étude concerne la séparation aveugle ou autodidacte de sources et leur déconvolution à partir de leurs mélanges convolutifs au second ordre. Le caractère *aveugle ou autodidacte* indique que nous ne disposons pas d'informations déterministes sur les sources et sur leurs canaux de transmission. Des algorithmes et techniques de bloc diagonalisation de matrices représentant les statistiques de ces mélanges, ont été développées pour la séparation de signaux stationnaires et non stationnaires. Pour obtenir une déconvolution complète des mélanges et restituer les signaux sources, nous avons adopté une approche à deux étapes : une séparation avec bloc diagonalisation suivie d'une égalisation avec la méthode sous-espace incorporant l'estimation de l'ordre du canal. Enfin, nous avons développé une plate-forme logicielle permettant d'évaluer les performances des différents algorithmes et de comparer notre approche à celle, plus classique, où l'égalisation précède la séparation des sources.

Mots clés: Séparation aveugle de sources, Déconvolution aveugle, Statistiques d'ordre deux, Bloc-diagonalisation.

Abstract

This study concerns the second order blind separation of sources and their deconvolution given their convolutive mixtures. The blindness indicates that we do not have any deterministic information neither on the sources nor on the transmission channels. We have developed block diagonalization techniques and algorithms of matrices that represent the data statistics in order to separate stationary, as well as, non stationary signals. To complete the deconvolution of the mixtures and recover the sources, we adopted a two-step approach: a separation with a block diagonalization followed by a subspace equalization including a channel order estimation. Finally, we have developed a graphic interface to evaluate the performances of the different algorithms and to compare our approach to a more classical one where equalization is prior to source separation.

Key Words: Blind Source Separation, Blind Deconvolution, Second order statistics, Block-diagonalization.

Remerciements

J'exprime mon immense gratitude à Messieurs Adel BELOUHRANI et Karim ABED-MERAIM pour leur patience et leur suivi intensif tout au long de ce travail de thèse. Qu'ils trouvent ici la preuve de ma reconnaissance.

Je remercie vivement Monsieur Ahmed ZERGUERRAS pour l'intérêt qu'il a bien voulu porter à ce travail en acceptant de présider le jury.

Je souhaite, également, remercier Madame Amina SERRIR et Messieurs Salah BOURENNANE et Mohamed TRABELSI pour avoir accepté de bien vouloir examiner mon travail.

Enfin, ce travail n'aurait jamais vu le jour sans le dévouement et le soutien de ma femme, Assya. Je la remercie pour tous ses sacrifices.

Table des matières

1	Introduction	9
2	Etat de l'art	13
2.1	Bref historique de la séparation aveugle de sources	13
2.2	Le mélange linéaire instantané	14
2.3	Le mélange linéaire convolutif	16
2.3.1	Conditions d'identifiabilité	18
2.3.2	Egalisation suivie d'une séparation	21
2.3.3	Séparation suivie d'une égalisation	29
3	Séparation à un filtre prés	41
3.1	Séparation de signaux stationnaires	41
3.1.1	Position du problème	42
3.1.2	Algorithme proposé :	43
3.1.3	Illustration	45
3.1.4	Le cas bruité :	46
3.2	Séparation de signaux non-stationnaires	49
3.2.1	Position du problème	49
3.2.2	Algorithme proposé	50
3.2.3	Critère de bloc diagonalisation conjointe	51
3.2.4	Illustration	52
4	Bloc diagonalisation conjointe	55
4.1	Position du problème	56
4.2	Approche par les Moindres Carrés	57

4.3	Les algorithmes basés sur la technique de Jacobi	58
4.3.1	Les rotations de Givens	58
4.3.2	Algorithme de BDC exact	60
4.3.3	Algorithme de BDC approximé	63
4.4	BDC via une Diagonalisation Conjointe	64
4.4.1	BDC via DC de matrices positives	64
4.4.2	BDC via DC sous une transformation unitaire commune	67
4.5	Evaluation des performances	69
5	Déconvolution complète du système MIMO	77
5.1	Algorithme proposé	77
5.1.1	Identification SIMO	78
5.1.2	Egalisation par forçage à zéro	81
5.2	Evaluation des performances	82
6	Simulations et évaluation des performances	85
6.1	Séparation de signaux stationnaires utilisant l'algorithme NJBD :	85
6.1.1	Influence du bruit	86
6.1.2	Influence du nombre de matrices d'auto-corrélation	88
6.1.3	Influence du nombre d'échantillons	89
6.2	Séparation de signaux stationnaires utilisant l'algorithme ALSBD :	90
6.2.1	Influence du bruit	90
6.2.2	Influence du nombre de matrices d'auto-corrélation	91
6.2.3	Influence du nombre d'échantillons	92
6.3	Séparation de signaux non-stationnaires utilisant l'algorithme PJBD :	93
6.3.1	Influence du bruit	93
6.3.2	Influence du nombre d'échantillons	95
6.4	Comparaison des algorithmes de séparation utilisant différentes méthodes de BDC	96
6.5	Comparaison des taux de réussites entre la méthode proposée avec différentes méthodes existantes	98
7	Conclusion et perspectives	101

Notations et définitions

Dans ce document, quelque soit le corps de référence :

- les matrices sont représentées par des majuscules en style gras (exemple : \mathbf{M}).
- les vecteurs par des minuscules en style gras (exemple : \mathbf{v}).
- les scalaires par des minuscules en style non gras (exemple : s).
- un élément d'une matrice \mathbf{M} est désigné par m_{ij} .

s^* ou \bar{s} : Nombre complexe conjugué du scalaire s .

\mathbf{v}^* ou $\bar{\mathbf{v}}$: Vecteur conjugué du vecteur \mathbf{v} .

\mathbf{M}^* ou $\bar{\mathbf{M}}$: Matrice conjuguée de la matrice \mathbf{M} .

$\Re(s)$: Partie réelle de s .

$\Im(s)$: Partie imaginaire de s .

\cdot^T : Transposé d'un vecteur ou d'une matrice.

\cdot^H : Transposé conjugué d'un vecteur ou d'une matrice.

$\text{vec}(\mathbf{M})$: Opérateur vecteur colonne permettant
la concaténation de toutes les colonnes de la matrice \mathbf{M} .

$\det(\mathbf{M})$: Déterminant de la matrice \mathbf{M} .

$|\cdot|$: Valeur absolue d'un nombre réel ou le module d'un nombre complexe.

$\|\mathbf{M}\|$: La norme de Frobenius de la matrice \mathbf{M} soit $\|\mathbf{M}\| = (\sum |m_{ij}|^2)^{1/2}$.

$\mathbf{M}^\#$: La matrice pseudo-inverse de \mathbf{M} au sens de Moore-Penrose,
si \mathbf{M} , de taille $[N \times M]$, $N \geq M$, est de rang plein ($\text{rang}(\mathbf{M}) = M$),
la pseudo-inverse de \mathbf{M} est $\mathbf{M}^\# = (\mathbf{M}^H \mathbf{M})^{-1} \mathbf{M}^H$.
Dans le cas où $N \leq M$, $\mathbf{M}^\# = \mathbf{M}^H (\mathbf{M} \mathbf{M}^H)^{-1}$.

\mathbf{I}_M : Matrice identité de dimension $[M \times M]$.

Λ : Matrice diagonale.

- $\text{diag}(\mathbf{M})$: Matrice diagonale formée par les éléments diagonaux de la matrice carrée \mathbf{M} .
- $\text{bdiag}(\mathbf{M})$: Matrice carrée contenant les bloc diagonaux de la matrice carrée \mathbf{M} et zéro ailleurs.
- $\text{deg}(\mathbf{H}_z(z))$: Degré de la matrice polynômiale $\mathbf{H}_z(z)$, degré le plus élevé de tous ses polynômes. Il correspond à l'ordre du canal représenté par $\mathbf{H}_z(z)$.
- $E[\cdot]$: Espérance mathématique.
- $*$: Opérateur de convolution linéaire.
- $\text{Range}(\mathbf{M})$: Espace engendré par les colonnes de la matrice \mathbf{M} .
- $\text{Range}(\mathbf{M})^\perp$: Espace orthogonal à $\text{Range}(\mathbf{M})$.
- $\text{rang}(\mathbf{M})$: Nombre de lignes ou de colonnes de \mathbf{M} linéairement indépendantes.
- Matrice polynômiale unimodulaire : Matrice polynômiale dont le déterminant est constant et dont l'inverse est aussi une matrice polynômiale.
- Matrice de Toeplitz : Matrice dont les éléments $m_{ij} = m_{i-j}$. Exemple :

$$\mathbf{M} = \begin{bmatrix} 1 & 7 & -1 & 2 & 9 \\ 4 & 1 & 7 & -1 & 2 \\ -8 & 4 & 1 & 7 & -1 \end{bmatrix}.$$

Matrice de Sylvestre : Matrice de Toeplitz particulière. Exemple :

$$\mathbf{M} = \begin{bmatrix} 1 & 7 & -1 & 0 & 0 \\ 0 & 1 & 7 & -1 & 0 \\ 0 & 0 & 1 & 7 & -1 \end{bmatrix}.$$

Produit de Kronecker : Soient \mathbf{A} et \mathbf{B} deux matrices de tailles respectives $[P \times Q]$ et $[M \times N]$, le produit de Kronecker entre \mathbf{A} et \mathbf{B} , noté $\mathbf{A} \otimes \mathbf{B}$, est une matrice de taille $[PM \times QN]$ définie par :

$$\mathbf{A} \otimes \mathbf{B} \triangleq \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1q}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{p1}\mathbf{B} & \cdots & a_{pq}\mathbf{B} \end{bmatrix}.$$

Abréviations

- ALSBD : Algorithme de BDC approximé utilisant les moindres carrés
(*Approximate Least Squares Block Diagonalization*).
- BDC : Bloc Diagonalisation Conjointe.
- BGL : Vraisemblance bloc gaussienne (*Block Gaussian Likelihood*).
- BIDS : Identification aveugle de canaux RIF MIMO par décorrélation de sous-canaux
(*Blind Identification of FIR MIMO Channels by Decorrelating Subchannels*).
- DC : Diagonalisation Conjointe.
- KL : Kullback-Leibler.
- LSBD : Algorithme de BDC exact utilisant les moindres carrés
(*Least Squares Block Diagonalization*).
- MC : Moindres Carrés.
- MIMO : Plusieurs entrées, plusieurs sorties (*Multiple Inputs, Multiple Outputs*).
- MP : Faisceau de matrices (*Matrix Pencil*).
- NJBD : Algorithme de BDC utilisant la technique de Jacobi de DC
(*New Joint Block Diagonalization*).
- PJBD : Algorithme de BDC inspiré de la technique de DC de Pham
(*Pham Joint Block Diagonalization*).
- RIF / RII : Réponse Impulsionnelle Finie / Réponse Impulsionnelle Infinie.
- SAS : Séparation Aveugle de Sources.
- SIMO : Une entrée, plusieurs sorties (*Single Input, Multiple Outputs*).
- SNR : Rapport signal sur bruit (*Signal to Noise Ratio*).
- SOBI : Identification aveugle au second ordre (*Second Order Blind Identification*).
- SOMOD : Déconvolution MIMO au second ordre
(*Second Order Multi Output Deconvolution*).

Chapitre 1

Introduction

Cette étude concerne la séparation aveugle de sources décorréées à partir de leurs mélanges convolutifs. Dans cette introduction, il convient de replacer cet énoncé, un tant soit peu académique, dans différents contextes, afin de mettre en lumière à la fois ses liens avec de nombreuses applications et sa pertinence scientifique.

Pour cela, nous commencerons par définir et recenser, brièvement, les domaines utilisateurs des techniques aveugles en traitement du signal. Le caractère *aveugle* indique que nous ne disposons pas d'informations a priori sur les sources et sur leurs canaux de transmission. On rencontre cette situation dans plusieurs domaines où les signaux sources ainsi que le canal de propagation ne sont pas connus ou, parfois, inaccessibles.

La séparation aveugle de sources (SAS), appelée également, séparation autodidacte de sources, consiste à estimer M signaux inconnus (appelées sources ou entrées) d'après la seule connaissance de leurs N mélanges (appelées observations ou sorties). Elle peut être divisée en plusieurs catégories suivant trois degrés de difficulté.

Le premier degré est le nombre d'observations N par rapport au nombre M de sources. Intuitivement, on conçoit aisément que le cas *sur-déterminé* (plus d'observations que de sources $N \geq M$) est plus simple à appréhender et, donc, à résoudre que le cas *sous-déterminé* ($N < M$). Ce dernier cas ne peut être résolu qu'au prix d'une importante information a priori sur les sources.

Un deuxième degré est la nature des sources. La plupart des méthodes de la SAS reposent sur l'hypothèse que les sources sont mutuellement indépendantes, au moins à l'ordre deux. Un autre aspect de la nature des sources est leur caractère stationnaire :

le cas des sources stationnaires est plus " facile " à résoudre que le cas des sources non-stationnaires. Nous verrons, par la suite, qu'il existe des techniques pour appréhender ce dernier cas.

Enfin, le troisième degré de difficulté est la nature des mélanges. Le mélange le plus simple est le mélange linéaire instantané : à chaque instant, les observations sont des combinaisons linéaires des sources au même instant. Plusieurs algorithmes et techniques très performants ont été développés dans ce domaine [1, 2]. Dans la réalité, les mélanges sont souvent convolutifs : à chaque instant, les observations dépendent aussi des sources aux instants précédents.

Dans le domaine du traitement du signal audio, une application classique de mélanges convolutifs est le problème connu sous le nom de " cocktail party " : lorsque plusieurs personnes sont entrain de parler dans une salle, et si ces personnes sont enregistrées par différents microphones, la proximité de ces personnes des microphones ainsi que l'effet de réverbération de la salle produisent au niveau des microphones des mélanges convolutifs. Dans la réalité, ces mélanges sont aussi variant dans le temps. Par exemple, lorsque les personnes se déplacent, les fonctions de transfert entre les émetteurs (les personnes) et les récepteurs (les microphones) changent et produisent, donc, sur les microphones, des mélanges variant dans le temps. De plus, certains microphones peuvent avoir un comportement non linéaire, bien souvent négligeable, par rapport à l'intensité des sources enregistrées. La technique, que nous proposerons dans ce travail, peut être appliquée pour séparer les contributions de chaque locuteur.

Une autre application audio est le " démixage " d'extraits musicaux stéréophoniques ou le rehaussement de la parole. Dans le domaine de la téléphonie mobile, la SAS à partir de mélanges convolutifs peut être appliquée pour séparer les interférences du signal utile. Nous citerons, également, les techniques d'élaboration de prothèses auditives (voir [3] pour un panorama plus exhaustif des applications en audio).

Dans le domaine biomédical, de nombreux travaux existent sur la séparation d'Electro-Encéphalo-Grammes (EEG) [4] et Electro-Cardio-Grammes (ECG) [5] (voir [6] pour un panorama des applications biomédicales).

Dans cette thèse, nous nous intéresserons, uniquement au cas *sur-déterminé* où le nombre

d'observations N est supérieur au nombre M de sources. De plus, toutes les méthodes, que nous développerons, supposent que nos sources sont mutuellement indépendantes à l'ordre deux (ou mutuellement décorréélées), ce qui positionne notre travail dans la SAS au second ordre. De même, les sources que nous séparons sont issues de mélanges convolutifs.

Ce document est organisé comme suit, le chapitre 2 sera consacré à l'état de l'art dans le domaine de la SAS. Nous ferons un tour d'horizon sur les techniques existantes ; nous présenterons quelques exemples d'algorithmes de déconvolution qui pour certains réalisent une égalisation du canal avant une séparation des signaux et pour d'autres une séparation avant une égalisation du canal dont les méthodes que nous proposons font partie.

Dans le chapitre 3, nous détaillerons deux méthodes que nous avons développées : l'une pour séparer des signaux stationnaires et l'autre pour des signaux non-stationnaires. Ces deux méthodes utilisent une bloc diagonalisation conjointe de matrices des statistiques des données. Ces deux méthodes effectuent une séparation à un filtre prés. Ce qui est suffisant pour certaines applications comme nous le verrons plus loin.

Au chapitre 4, nous passerons en revue quatre algorithmes de bloc diagonalisation conjointe : les deux premiers sont basés sur la technique de Jacobi qui minimise, d'une manière itérative, un critère des moindres carrés ; les deux autres utilisent une diagonalisation conjointe pour bloc diagonaliser des matrices à une matrice de permutation prés. Nous verrons comment enlever l'effet de ces permutations. Des simulations permettent de faire une comparaison entre ces différents algorithmes en fonction du bruit, de la taille des matrices, du nombre d'itérations ...

Le chapitre 5 complète la présentation du chapitre 3 en égalisant le filtre manquant et, donc, en réalisant une déconvolution complète du système. Dans le chapitre 6, nous allons présenter des simulations pour évaluer les performances des différents algorithmes de séparation développés. Pour cela nous avons réalisé deux plate-formes graphiques, l'une pour la bloc diagonalisation conjointe et l'autre pour la déconvolution complète du système. Et enfin, nous terminerons par une conclusion générale dans laquelle, nous citerons, quelques perspectives pour la suite de notre travail.

Chapitre 2

Etat de l'art

2.1 Bref historique de la séparation aveugle de sources

La séparation aveugle de sources est un problème relativement récent en traitement du signal. Il a été formulé, pour la première fois, en 1984 par J. Héroult *et al.* [7, 8]; ces derniers ont modélisé un phénomène de mouvement neurobiologique à l'aide d'un algorithme de séparation d'un mélange instantané sans bruit, à base d'un réseau neuro-mimétique. J. L. Lacoume et P. Ruize [9] ont présenté, en 1988, un algorithme qui permet la minimisation d'une fonction objective basée sur les cumulants croisés des sorties d'un filtre inverse. En 1989, J. F. Cardoso [10] propose une méthode algébrique pour extraire des composants indépendants en considérant les cumulants d'ordre quatre et en utilisant l'information d'ordre deux pour rendre unitaire la matrice de mélange puis l'identifier par des techniques de diagonalisation. Plus tard, le même auteur [11, 12], montre qu'il suffit d'utiliser les cumulants d'ordre quatre pour estimer la matrice de mélange.

P. Comon [13], en 1990, développe une nouvelle approche pour appréhender le problème de la SAS en généralisant la théorie de l'analyse en composantes principales à l'analyse en composantes indépendantes. Pour cela, Il utilise une fonction contraste qui permet de maximiser la somme des cumulants d'ordre quatre.

A la même période, M. Gaeta et J. L. Lacoume [14] proposent une technique de maximum de vraisemblance approchée pour résoudre le problème de la séparation.

De même, D. T. Pham *et al.* [15] utilisent la notion du maximum de vraisemblance pour séparer des observations non-bruitées.

En 1993, N. Delfosse [16] propose un algorithme se basant sur le principe de "déflation" : dans un premier temps, l'algorithme sépare une seule source après blanchiment, ce qui permet d'obtenir un filtre ligne "séparant" et donc son complément orthogonal. A la sortie de ce dernier, on récupère un mélange des autres sources auxquelles, de nouveau, on applique un filtre semblable de dimension inférieure.

La même année, B. Laheld [17] utilise la notion du gradient matriciel relatif et propose un certain nombre d'algorithmes adaptatifs. Il est à noter que, en absence du bruit, les performances de ces algorithmes sont indépendantes des différents mélanges. Ce qui représente un avantage par rapport aux autres approches.

Depuis, la SAS a pris un essor grandissant et est devenue un domaine de recherche à part entière. Une multitude d'algorithmes ont vu le jour, certains utilisent les statistiques d'ordre élevé et d'autres, plus simples, d'ordre deux comme celle proposée par A. Belouchrani *et al.* en 1997 [1]. Nous citerons, également, les méthodes qui permettent de séparer des sources non-stationnaires : par exemple, la méthode de A. Belouchrani *et al.* en 1998 qui utilise les signatures temps-fréquences des sources à séparer [18]. Et enfin, plus tard, celle de D. T. Pham et J. F. Cardoso dans [2] en 2000.

Nous allons, dans ce qui suit, présenter le modèle de la SAS, tout d'abord, d'une manière succincte, pour le mélange instantané puis pour le mélange convolutif qui est l'objet de notre travail.

2.2 Le mélange linéaire instantané

Le problème de la séparation de sources pour le mélange linéaire instantané peut être formulé comme suit :

On suppose, qu'à l'instant n , il existe M signaux sources notés $s_j(n)$, pour $j = 1, \dots, M$ dont on observe N mélanges linéaires instantanés $x_i(n)$, $i = 1, \dots, N$, appelés signaux capteurs ou observations. Le modèle s'écrit alors sous la forme matricielle :

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{n}(n) \quad n = 0, \dots, T - 1 \quad (2.1)$$

où le vecteur $\mathbf{x}(n)$ regroupe dans ses composantes les N signaux capteurs, alors que les M signaux sources $s_1(n), \dots, s_M(n)$, supposés statistiquement indépendants, sont contenus dans le vecteur $\mathbf{s}(n)$. Les coefficients des mélanges forment une matrice notée \mathbf{A} de

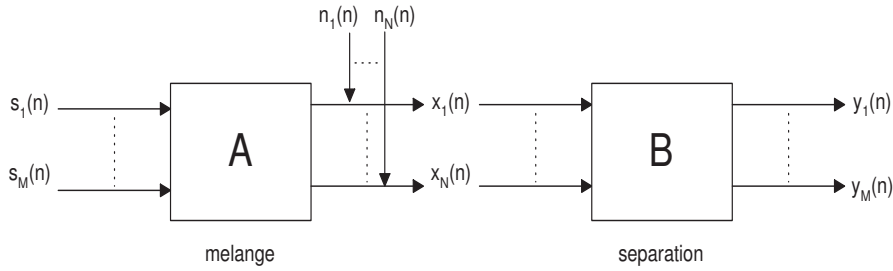


FIG. 2.1 – Mélange instantané.

dimension $[N \times M]$, appelée *matrice de mélange*. $\mathbf{n}(n)$ représente un bruit additif.

Dans un contexte aveugle où l'on n'a aucune information *a priori* sur le mélange, une identification complète de la matrice de mélange \mathbf{A} est impossible. En effet, deux indéterminations importantes surgissent :

- Tout d'abord, on remarque facilement que la relation entrée-sortie est une application *surjective* : le même vecteur des observations $\mathbf{x}(n)$ peut être généré à l'aide d'une infinité de vecteurs $\mathbf{s}(n)$. En effet, l'ordre des signaux est arbitraire car toute permutation appliquée sur les sources et sur les lignes de \mathbf{A} correspondantes donne naissance au même vecteur $\mathbf{x}(n)$. Ce qui revient à dire que l'indice j associé à une certaine source est arbitraire. Les sources ne peuvent, donc, être déterminées qu'à une permutation près.
- L'échange d'un facteur complexe quelconque entre un signal source et le vecteur colonne de la matrice de mélange \mathbf{A} correspondant n'affecte en rien les observations. Ceci est montré par la relation suivante :

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{n}(n) = \sum_{j=1}^M \frac{\mathbf{a}_j}{\alpha_j} \alpha_j s_j(n) + \mathbf{n}(n) \quad (2.2)$$

On dira que les sources ne peuvent, donc, être déterminées qu'à un facteur d'échelle et à une phase près pour les mélanges instantanés.

Certaines hypothèses doivent être prises en compte pour résoudre le problème de la SAS :

- **Hypothèse1** : Les sources sont mutuellement indépendantes si l'on utilise les statistiques d'ordre supérieur ou mutuellement décorrélatées si les statistiques d'ordre deux sont utilisées.

- **Hypothèse2** : On considère que nous sommes dans le cas *sur-déterminé* (plus d'observations que de sources $N \geq M$). La matrice \mathbf{A} est supposée de rang plein et inversible.

• **Hypothèse3** : Les sources sont non gaussiennes sauf au plus une si l'on utilise les statistiques d'ordre supérieur ou ont des spectres différents si les statistiques d'ordre deux sont utilisées.

Finalement, nous appellerons *séparation de source* l'opération qui, à partir des signaux capteurs, nous permet dans le cas idéal (sans bruit) d'obtenir un ensemble de signaux proportionnels aux signaux émis par chacune des sources. De même, la matrice de mélange ne peut être estimée qu'à une constante et à une permutation près de ces colonnes.

Le principe général de la séparation de sources consiste à appliquer des transformations aux observations pour obtenir des signaux *statistiquement indépendants ou décorrélés* qui seront proportionnels aux signaux sources que l'on veut estimer. Ceci se formalise comme suit :

$$\mathbf{y}(n) = \mathbf{B}\mathbf{x}(n) \quad (2.3)$$

où \mathbf{B} est une matrice "séparante" et $\mathbf{y}(n)$ est une estimation des sources aux indéterminations d'échelle et de permutation près. Plusieurs travaux ont été réalisés pour proposer des solutions au problème de la SAS dans le cas d'un mélange linéaire instantané. Pour plus de détails, nous invitons le lecteur à consulter ces quelques publications qui nous paraissent significatives : [1], [2], [10], [13].

2.3 Le mélange linéaire convolutif

Malheureusement, le modèle instantané de la SAS reste restrictif dans un grand nombre de domaines d'applications tels que la télé-conférence. Par exemple, dans le cas de sources enregistrées dans un environnement acoustique réel : le signal capturé par un microphone (l'observation) sera représenté par une convolution du signal source avec un filtre RIF (filtre à réponse impulsionnelle finie) d'ordre élevé, ce qui permet de modéliser l'acoustique de la pièce entre la source et le microphone (le capteur). Intuitivement, afin d'augmenter la précision de ce modèle, des filtres RII (filtre à réponse impulsionnelle infinie) pourraient être utilisés pour modéliser l'acoustique de la pièce. Seulement, les filtres RII ont besoin de condition de phase minimale et se révèlent, donc, moins stables que les filtres RIF. Nous nous limiterons au modèle suivant pour décrire le problème de la SAS à

partir de mélanges linéaires convolutifs :

$$\mathbf{x}(n) = \mathbf{H}(n) * \mathbf{s}(n) + \mathbf{n}(n) \quad (2.4)$$

Le symbole $*$ représente l'opérateur de convolution linéaire. Soit, en développant et en utilisant la transformée en z :

$$\mathbf{x}(n) = \sum_{l=0}^{L-1} \mathbf{H}(l) \mathbf{s}(n-l) + \mathbf{n}(n) \triangleq [\mathbf{H}_z(z)] \mathbf{s}(n) + \mathbf{n}(n) \quad (2.5)$$

où $\forall l \in [0, L-1]$, $\mathbf{H}(l)$ est une matrice de taille $[N \times M]$ et $\mathbf{H}_z(z) = \sum_{l=0}^{L-1} \mathbf{H}(l) z^{-l}$ est une matrice polynômiale en z de dimension $[N \times M]$ représentant la fonction de transfert d'un filtre de mélange causal inconnu avec $N > M$.

$$\mathbf{H}_z(z) \triangleq \begin{bmatrix} h_{11}(z) & \cdots & h_{1M}(z) \\ \vdots & \ddots & \vdots \\ h_{N1}(z) & \cdots & h_{NM}(z) \end{bmatrix} \quad (2.6)$$

$(L-1)$ désigne l'ordre du canal, soit le degré le plus élevé des polynômes des différentes fonctions de transfert scalaires $h_{ij}(z)$ entre un capteur i et une source j .

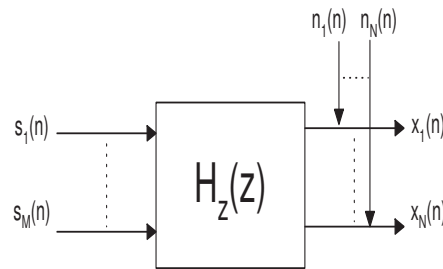


FIG. 2.2 – Mélange convolutif.

La même indétermination concernant la résolution de ce problème à une permutation près subsiste ; en effet, l'indice j associé à une source particulière reste arbitraire . Cependant, l'indétermination de la constante complexe près (à une échelle et à une phase près) des mélanges instantanés devient une indétermination à **un filtre près** dans le cas des mélanges convolutifs.

Les premiers travaux qui ont abordé la SAS à partir de mélanges convolutifs ont été

développés dans le domaine temporel : en 1996, K. Torkkola [19] a modélisé le procédé de séparation par un filtre RIF,

$$\mathbf{y}(n) = \sum_{l=0}^{L-1} \mathbf{G}(l)\mathbf{x}(n-l) \quad (2.7)$$

Il s'agit, donc, d'estimer les coefficients g_{ij} de la matrice "séparante" \mathbf{G} . Pour cela, il choisit de maximiser une information mutuelle [20]. Par contre, T. W. Lee *et al.* [21] ont utilisé des filtres RII pour modéliser la procédure de "démélange", ce qui les oblige à supposer que les filtres qui représentent l'enregistrement acoustique sont à phase minimale et donc de s'éloigner des conditions réelles de travail. D'autres auteurs comme P. Smaragdis [22], ont proposé de travailler exclusivement dans le domaine fréquentiel, ce qui permet de transformer la convolution en multiplication. Le problème devient une estimation de la matrice "séparante" pour chaque trame de fréquences. De même, L. Parra *et al.* [23] ont également travaillé dans le domaine temps-fréquences en utilisant la non-stationnarité pour effectuer la séparation, leur contribution se résume à imposer une contrainte à la longueur du filtre séparateur.

Avant d'illustrer les techniques de la séparation aveugle à partir de mélanges convolutifs avec des exemples détaillés d'algorithmes existants, il est bon de citer les conditions qui permettent à de tels systèmes d'être identifiables.

2.3.1 Conditions d'identifiabilité

Pour bien comprendre les conditions d'identifiabilité des systèmes plusieurs-entrées plusieurs-sorties (MIMO), nous allons donner, auparavant, les conditions d'identifiabilité des systèmes une entrée, plusieurs sorties (SIMO).

Conditions d'identifiabilité des systèmes SIMO :

Un étude détaillée de l'identifiabilité SIMO est présentée dans [24, 25]. Nous citerons, dans ce qui suit les points essentiels. Un système SIMO de N observations est modélisé de la manière suivante :

$$\mathbf{x}(n) = \mathbf{h}(n) * \mathbf{s}(n) + \mathbf{n}(n) \quad (2.8)$$

Soit, en développant et en utilisant la transformée en z :

$$\mathbf{x}(n) = \sum_{l=0}^{L-1} \mathbf{h}(l)\mathbf{s}(n-l) + \mathbf{n}(n) \triangleq [\mathbf{h}_z(z)]\mathbf{s}(n) + \mathbf{n}(n) \quad (2.9)$$

où $\forall l \in [0, L-1]$, $\mathbf{h}(l)$ est un vecteur de taille $[N \times 1]$ et $\mathbf{h}_z(z) = \sum_{l=0}^{L-1} \mathbf{h}(l)z^{-l}$ est un vecteur polynômial en z de dimension $[N \times 1]$ représentant la fonction de transfert d'un filtre de mélange causal inconnu entre les N capteurs et la source.

$$\mathbf{h}_z(z) \triangleq \begin{bmatrix} h_1(z) \\ \vdots \\ h_N(z) \end{bmatrix} \quad (2.10)$$

L'identifiabilité SIMO fait appel aux notions de "zéros" et de "modes" :

1. **Zéros** : z_0 est un zéro du i^{eme} canal du système si :

$$h_i(z_0) = \sum_{l=0}^{L-1} h_i(l)z_0^{-l} = 0$$

Un canal scalaire de degré $(L-1)$ possède $(L-1)$ zéros et ces derniers le caractérisent complètement à un scalaire près.

2. **Modes** : Un mode est une séquence de la forme $m_l = l^k z^l$ où l'entier k représente l'ordre du mode.

La séquence d'entrée finie $(\mathbf{s}(n), n = -L+1, \dots, 0, \dots, T-1)$ de longueur $(L+T-1)$ a p modes si $\mathbf{s}(n)$ peut s'écrire comme une combinaison linéaire de ces p modes de longueur $(L+T-1)$,

$$\mathbf{s}(n) = \sum_{i=1}^p c^{(i)} m_l^{(i)}$$

T étant le nombre d'échantillons observés sur chaque canal.

On peut, à présent, formuler les conditions d'identifiabilité comme suit :

a/- Conditions nécessaires d'identifiabilité :

Le système est identifiable si et seulement si :

- Les N canaux ne partagent pas de zéro commun : $\iff \nexists z_0$ tel que $h_i(z_0) = 0$ pour tout i .
- Le nombre de modes de la séquence d'entrée est supérieur ou égal à $L+1$.

- Le nombre d'échantillons $T \geq L + 1$.

b/- Conditions suffisantes d'identifiabilité :

Le système est identifiable si et seulement si :

- Les N canaux ne partagent pas de zéro commun.
- Le nombre de modes de la séquence d'entrée est supérieur ou égal à $2L + 1$.
- $T \geq 3L - 2$

Interprétation :

Ces conditions traduisent les exigences suivantes :

- Les canaux doivent être suffisamment distincts les uns des autres. Ils ne peuvent pas être identiques.
- La séquence d'entrée doit être suffisamment *complexe*. Elle ne peut être ni nulle, ni constante, ni une simple sinusoïde.
- Il doit y avoir un nombre suffisant d'échantillons T pour identifier le canal. Le nombre d'échantillons ne peut être inférieur à celui des paramètres à estimer.

Conditions d'identifiabilité des systèmes MIMO :

Les conditions de l'identifiabilité MIMO sont détaillées dans [29]. C'est une extension de celles présentées dans le cas SIMO. Nous les résumerons comme suit :

- Le nombre des signaux sources M est strictement inférieur au nombre des signaux capteurs N .
- Les M signaux sources sont supposés statistiquement mutuellement indépendants et ayant une *excitation persistante*¹.
- La matrice polynômiale, $\mathbf{H}_z(z) = \sum_{l=0}^{L-1} \mathbf{H}(l)z^{-l}$, de taille $[N \times M]$, $M < N$, est *irréductible* i.e. $\mathbf{H}_z(z)$ est de rang plein pour tout z . Soit $\text{rang}(\mathbf{H}_z(z)) = M, \forall z$.
- $\mathbf{H}_z(z)$ est à *colonnes réduites*, i.e. si les vecteurs colonnes de $\mathbf{H}_z(z)$ sont de même degré $(L - 1)$, la matrice $\mathbf{H}(L - 1)$, qui contient les coefficients du degré le plus élevé, est de rang plein, i.e. $\text{rang}(\mathbf{H}(L - 1)) = M$.

Dans ce qui suit, nous allons décrire quelques méthodes pour résoudre le problème de la SAS à partir de mélanges convolutifs linéaires MIMO. Une multitude de travaux, utili-

¹En anglais : *persistently exciting*, cela rejoint la deuxième condition dans le cas SIMO

sant différentes approches, existe, nous ne pouvons les citer tous. Nous nous limiterons, dans ce travail, aux méthodes de second ordre et nous les grouperons en deux classes de méthodes ou d'algorithmes en fonction de leur séquençement :

- **1ère classe** : Les algorithmes utilisant une égalisation² suivie d'une séparation.
- **2ème classe** : Les algorithmes utilisant une séparation suivie d'une égalisation.

Nous allons, à présent, décrire chaque classe en l'illustrant d'un ou deux exemples d'algorithmes qui, à nos yeux, paraissent significatifs.

2.3.2 Egalisation suivie d'une séparation

Cette classe regroupe la majorité des méthodes de la SAS à partir de mélanges convolutifs MIMO. En effet et jusqu'à un passé récent, il n'existait dans la littérature que des méthodes, dites, de *déconvolution aveugle* ou encore *identification aveugle* qui, dans leur structure générale, réalisent une égalisation pour retrouver le canal manquant à une matrice constante près, ce qui permet de passer d'un mélange convolutif à un mélange instantané, puis une séparation pour estimer les sources désirées. L'algorithme proposé par D. Godard [26] pour les communications numériques en 1980 dans le cas d'une seule source est considéré comme pionnier dans ce domaine, s'ensuit, alors, un très grand nombre de contributions utilisant aussi bien les statistiques d'ordre deux que celles d'ordres plus élevés. Dans le cas de la déconvolution aveugle d'un système SIMO, plusieurs types de solution existent, l'article de K. Abed-Meraim [27] en propose une revue.

Parmi les méthodes utilisant les statistiques du second ordre, la méthode de *sous-espace* paraît, à nos yeux, la plus représentative de cette classe de méthodes qui commencent par une égalisation puis effectuent une séparation.

Description de la méthode sous-espace :

Cette méthode a été introduite par Moulines *et al.* [28] en 1995 pour le cas SIMO. Elle exploite, comme nous le verrons plus loin, la structure bloc Sylvestre de la matrice de filtrage. L'idée de base est de décomposer une matrice de covariance des données reçues en sous-espaces bruit et signal. Cette méthode a l'avantage d'être simple et relativement

²L'égalisation est prise ici dans son sens le plus large et ne concerne pas uniquement les communications numériques.

peu coûteuse en calculs. Nous allons la présenter dans le cas MIMO.

Reprenons le modèle défini en (2.5),

$$\mathbf{x}(n) = \sum_{l=0}^{L-1} \mathbf{H}(l)\mathbf{s}(n-l) + \mathbf{n}(n) \triangleq [\mathbf{H}_z(z)]\mathbf{s}(n) + \mathbf{n}(n). \quad (2.11)$$

avec :

- $\mathbf{x}(n) = [x_1(n), \dots, x_N(n)]^T$ est un processus de dimension N stationnaire à temps discret représentant les signaux capteurs.
- $\mathbf{s}(n) = [s_1(n), \dots, s_M(n)]^T$ est le processus inconnu de dimension M représentant les signaux sources supposées mutuellement indépendants, i.e., $E[\mathbf{s}(n)\mathbf{s}^H(n)] = \sigma_s^2 \mathbf{I}_M$.
- $\mathbf{n}(n) = [n_1(n), \dots, n_N(n)]^T$ est un bruit additif de dimension N indépendant de la séquence $\{\mathbf{s}(n)\}$, spatialement et temporellement blanc, i.e. $E[\mathbf{n}(n)\mathbf{n}^H(n)] = \sigma_n^2 \mathbf{I}_N$ avec σ_n^2 inconnu.
- $\mathbf{H}_z(z) = \sum_{l=0}^{L-1} \mathbf{H}(l)z^{-l}$ est une matrice polynômiale en z de dimension $[N \times M]$ décrite en 2.3.

Dans le contexte des communications numériques, la séquence d'entrée $\{\mathbf{s}(n)\}$ représente les symboles transmis et la fonction de transfert inconnue $\mathbf{H}_z(z)$ modélise le canal de propagation entre les sources et les capteurs.

Les hypothèses sur ce modèle sont comme suit :

- **H1** : La matrice $\mathbf{H}_z(z)$ est irréductible soit $\text{rang}(\mathbf{H}_z(z)) = M$ pour tout z .
- **H2** : La matrice $\mathbf{H}_z(z)$ est à colonnes réduites.

Ces hypothèses sont vérifiées dès que les divers canaux de propagation sont suffisamment différents les uns des autres. Autrement dit, les capteurs sont "assez" éloignés les uns des autres de manière à ce que leurs canaux soient indépendants.

Il est montré dans [29] que, si les degrés des différents polynômes sont quelconques, la matrice $\mathbf{H}_z(z)$ est identifiable à une matrice $\mathbf{R}_z(z)$ polynômiale unimodulaire bloc triangulaire supérieure prés, de dimension $[M \times M]$, soit,

$$\mathbf{F}_z(z) = \mathbf{H}_z(z)\mathbf{R}_z(z)$$

où,

$$\mathbf{R}_z(z) = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12}(z) & \cdots & \mathbf{R}_{1s}(z) \\ \mathbf{0} & \mathbf{R}_{22} & \cdots & \mathbf{R}_{2s}(z) \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{R}_{ss} \end{bmatrix}$$

Les matrices, $\mathbf{R}_{ij}(z)$, $i, j = 1, \dots, s$, sont des matrices de même ordre. Les matrices \mathbf{R}_{kk} , $k = 1, \dots, s$, sont des matrices constantes et non singulières.

Dans le cas, plus simple, où les degrés les plus élevés de tous les polynômes sont tous égaux à $(L - 1)$, la matrice $\mathbf{R}_z(z)$ est réduite à une matrice constante inversible telle que : $\mathbf{F}_z(z) = \mathbf{H}_z(z)\mathbf{R}$.

Nous allons, par la suite, présenter une méthode d'identification pour le cas où tous les degrés les plus élevés des polynômes sont identiques et égaux à $(L - 1)$, soit l'ordre des différents canaux est égal à $(L - 1)$. Cette identification est réalisée en calculant la matrice de covariance des observations. Nous allons résumer les principales étapes de la méthode sous-espace.

Reprenons notre modèle (2.11) défini pour tout instant n dans $[0, T - 1]$ et soit W un entier désignant la longueur d'une fenêtre temporelle que l'on choisit. On définit, pour chaque capteur i et $\forall n \in [W - 1, T - 1]$, les vecteurs observation et bruit suivants de dimension $[W \times 1]$:

$$\begin{aligned} \mathbf{x}_i(n) &= [x_i(n), \dots, x_i(n - W + 1)]^T \\ \mathbf{n}_i(n) &= [n_i(n), \dots, n_i(n - W + 1)]^T \end{aligned}$$

Construisons, à présent, par concatenation, les vecteurs de dimension $[NW \times 1]$ qui regroupent toutes les observations ainsi que leurs bruits respectifs,

$$\begin{aligned} \mathbf{x}(n) &= [\mathbf{x}_1^T(n), \dots, \mathbf{x}_N^T(n)]^T \\ \mathbf{n}(n) &= [\mathbf{n}_1^T(n), \dots, \mathbf{n}_N^T(n)]^T \end{aligned}$$

On définit, également, pour chaque source j , le vecteur suivant de dimension $[(W + L - 1) \times 1]$:

$$\mathbf{s}_j(n) = [s_j(n), \dots, s_j(n - (W - 1) - (L - 1))]^T$$

De la même manière, construisons par concaténation, le vecteur de dimension $[M(W + L - 1) \times 1]$ qui regroupe tous les signaux sources :

$$\mathbf{s}(n) = [\mathbf{s}_1^T(n), \dots, \mathbf{s}_M^T(n)]^T$$

Entre le capteur i et la source j , on définit la matrice de Sylvestre suivante de dimension $[W \times (W + L - 1)]$:

$$\mathbf{H}_W(\mathbf{h}_{ij}) = \begin{bmatrix} h_{ij}(0) & \cdots & h_{ij}(L-1) & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & h_{ij}(0) & \cdots & h_{ij}(L-1) \end{bmatrix} \quad (2.12)$$

Où \mathbf{h}_{ij} est un vecteur ligne de dimension $[1 \times L]$, tel que $\mathbf{h}_{ij} = [h_{ij}(0) \cdots h_{ij}(L-1)]$. Finalement, on définit la matrice filtre suivante de dimension $[NW \times M(W + L - 1)]$:

$$\mathcal{T}_W(\mathbf{H}) = \begin{bmatrix} \mathbf{H}_W(\mathbf{h}_{11}) & \cdots & \mathbf{H}_W(\mathbf{h}_{1M}) \\ \vdots & \ddots & \vdots \\ \mathbf{H}_W(\mathbf{h}_{N1}) & \cdots & \mathbf{H}_W(\mathbf{h}_{NM}) \end{bmatrix}$$

Avec ces notations, on aura la nouvelle formulation pour $\forall n \in [W - 1, T - 1]$:

$$\mathbf{x}(n) = \mathcal{T}_W(\mathbf{H})\mathbf{s}(n) + \mathbf{n}(n). \quad (2.13)$$

Le paramètre W doit être choisi de telle sorte que le système soit *sur-déterminé*, c'est à dire tel que, $NW \geq M(L + W - 1)$.

Pour construire le sous-espace orthogonal (dit sous-espace bruit), on construit la matrice de covariance $\mathbf{R}_{\mathbf{xx}}^W$ des données reçues :

$$\begin{aligned} \mathbf{R}_{\mathbf{xx}}^W &\triangleq E[\mathbf{x}(n)\mathbf{x}^H(n)] \\ &= \mathcal{T}_W(\mathbf{H}) \mathbf{R}_{\mathbf{ss}}^W \mathcal{T}_W(\mathbf{H})^H + \sigma_n^2 \mathbf{I}_N \end{aligned}$$

avec la matrice de covariance des signaux sources $\mathbf{R}_{\mathbf{ss}}^W \triangleq E[\mathbf{s}(n)\mathbf{s}^H(n)] = \sigma_s^2 \mathbf{I}_M$.

Si $\mathbf{R}_{\mathbf{ss}}^W$ est de rang plein (ce que l'on suppose), le sous-espace signal et le sous-espace bruit peuvent-être calculés par la décomposition en vecteurs et valeurs propres de $\mathbf{R}_{\mathbf{xx}}^W$. Dans ce cas, la variance du bruit σ_n^2 est la plus petite valeur propre de $\mathbf{R}_{\mathbf{xx}}^W$. Le sous-espace associé à σ_n^2 , appelé *sous-espace bruit*, est de dimension $[NW - M(W + L - 1)]$. Le

sous-espace bruit est le complément orthogonal du sous-espace signal $\text{Range}(\mathcal{T}_W(\mathbf{H}))$.

Soit :

$$\begin{aligned} \mathbf{R}_{xx}^W &\triangleq \sigma_s^2 \mathcal{T}_W(\mathbf{H}) \mathcal{T}_W(\mathbf{H})^H + \sigma_n^2 \mathbf{I}_N \\ &= \begin{bmatrix} \mathbf{U}_s & \mathbf{U}_n \end{bmatrix} \begin{bmatrix} \Lambda_s & \mathbf{0} \\ \mathbf{0} & \Lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^H \\ \mathbf{U}_n^H \end{bmatrix} \end{aligned}$$

\mathbf{U}_s est de dimension $[NW \times M(W + L - 1)]$ et \mathbf{U}_n est de dimension $[NW \times (NW - M(W + L - 1))]$, elles représentent des matrices dont les colonnes sont, respectivement, les vecteurs propres des sous-espaces signal et bruit. Λ_s et Λ_n sont, respectivement, des matrices diagonales construites à partir des valeurs propres des sous-espaces signal et bruit.

On définit $\mathbf{\Pi}_W$ la matrice de projection orthogonale sur l'espace $\text{Range}(\mathcal{T}_W(\mathbf{H}))^\perp$, qu'on appelle, le sous-espace bruit. Et, en utilisant la relation d'orthogonalité entre les sous-espaces signal et bruit, on obtient l'équation fondamentale suivante :

$$\mathbf{\Pi}_W \mathcal{T}_W(\mathbf{H}) = \mathbf{0} \quad (2.14)$$

Pour estimer le canal de propagation, on doit trouver la matrice polynômiale $\mathbf{H}_z(z)$ qui satisfait à (2.14). Pratiquement, il s'agit de trouver un vecteur $\hat{\mathbf{h}}$ qui minimise le critère quadratique suivant,

$$\hat{\mathbf{h}} = \arg \min \{ \|\mathbf{\Pi}_W \mathcal{T}_W(\mathbf{F})\|^2 : \|\mathbf{F}\| = 1 \}. \quad (2.15)$$

Le vecteur $\hat{\mathbf{h}}$ de dimension $[1 \times NML]$ est construit, par concatenation et regroupe tous les filtres estimés du système.

Pour minimiser (2.15), procédons comme suit, la matrice \mathbf{U}_n , correspondante au sous-espace bruit, peut être représentée par ses colonnes comme suit : $\mathbf{U}_n = [\mathbf{u}_1 \mathbf{u}_2 \cdots \mathbf{u}_d]$ avec $d = NW - M(W + L - 1)$.

Prenons un vecteur \mathbf{u}_k ($1 \leq k \leq d$) de dimension $[NW \times 1]$:

$$\mathbf{u}_k \triangleq \begin{bmatrix} \mathbf{u}_k^1 \\ \vdots \\ \mathbf{u}_k^N \end{bmatrix}$$

Les \mathbf{u}_k^i ($1 \leq i \leq N$) sont de dimension $[W \times 1]$.

Pour chaque k , la relation (2.14) devient :

$$\mathbf{u}_k^H \mathcal{T}_W(\mathbf{H}) = \mathbf{0} \Leftrightarrow \left[(\mathbf{u}_k^1)^H \cdots (\mathbf{u}_k^N)^H \right] \begin{bmatrix} \mathbf{H}_W(\mathbf{h}_{11}) & \cdots & \mathbf{H}_W(\mathbf{h}_{1M}) \\ \vdots & \ddots & \vdots \\ \mathbf{H}_W(\mathbf{h}_{N1}) & \cdots & \mathbf{H}_W(\mathbf{h}_{NM}) \end{bmatrix} = \mathbf{0}$$

Ce qui nous donne :

$$\left[\sum_{i=1}^N (\mathbf{u}_k^i)^H \mathbf{H}_W(\mathbf{h}_{i1}) \quad \cdots \quad \sum_{i=1}^N (\mathbf{u}_k^i)^H \mathbf{H}_W(\mathbf{h}_{iM}) \right] = \mathbf{0} \quad (2.16)$$

La relation ci-dessus (2.16) est une opération de convolution et à cause de la structure Sylvestre des matrices $\mathbf{H}_W(\mathbf{h}_{ij})$ (voir (2.12)), on peut montrer qu'elle est équivalente à la relation suivante :

$$\left[\sum_{i=1}^N \mathbf{h}_{i1} \mathbf{H}_L((\mathbf{u}_k^i)^H) \quad \cdots \quad \sum_{i=1}^N \mathbf{h}_{iM} \mathbf{H}_L((\mathbf{u}_k^i)^H) \right] = \mathbf{0} \quad (2.17)$$

Chaque somme du terme à gauche de l'équation, correspondante à la source j , peut être développée de cette manière :

$$\sum_{i=1}^N \mathbf{h}_{ij} \mathbf{H}_L((\mathbf{u}_k^i)^H) = \left[\mathbf{h}_{1j} \quad \cdots \quad \mathbf{h}_{Nj} \right] \begin{bmatrix} \mathbf{H}_L((\mathbf{u}_k^1)^H) \\ \vdots \\ \mathbf{H}_L((\mathbf{u}_k^N)^H) \end{bmatrix}$$

Posons à présent, pour plus de simplicité :

$$\mathbf{H}_L^k \triangleq \begin{bmatrix} \mathbf{H}_L((\mathbf{u}_k^1)^H) \\ \vdots \\ \mathbf{H}_L((\mathbf{u}_k^N)^H) \end{bmatrix} \quad \text{et} \quad \mathbf{h}_j = \left[\mathbf{h}_{1j} \quad \cdots \quad \mathbf{h}_{Nj} \right]$$

Finalement, l'équation (2.17) s'écrit :

$$\left[\mathbf{h}_1 \mathbf{H}_L^k \quad \cdots \quad \mathbf{h}_M \mathbf{H}_L^k \right] = \mathbf{0} \quad (2.18)$$

Généralisons pour tous les vecteurs propres du sous-espace bruit,

$$[\mathbf{u}_1 \cdots \mathbf{u}_d]^H \mathcal{T}_W(\mathbf{H}) = \mathbf{0} \Leftrightarrow \begin{bmatrix} \mathbf{u}_1^H \mathcal{T}_W(\mathbf{H}) \\ \vdots \\ \mathbf{u}_d^H \mathcal{T}_W(\mathbf{H}) \end{bmatrix} = \mathbf{0}$$

Soit,

$$\begin{bmatrix} \mathbf{h}_1 \mathbf{H}_L^1 & \cdots & \mathbf{h}_M \mathbf{H}_L^1 \\ \vdots & \ddots & \vdots \\ \mathbf{h}_1 \mathbf{H}_L^d & \cdots & \mathbf{h}_M \mathbf{H}_L^d \end{bmatrix} = \mathbf{0} \quad (2.19)$$

Le vecteur $\mathbf{h} = [\mathbf{h}_1 \cdots \mathbf{h}_M]$ de dimension $[1 \times NML]$ regroupe tous les filtres du système à estimer. La relation (2.19) devient,

$$\begin{bmatrix} \mathbf{h}(\mathbf{I}_M \otimes \mathbf{H}_L^1) \\ \vdots \\ \mathbf{h}(\mathbf{I}_M \otimes \mathbf{H}_L^d) \end{bmatrix} = \mathbf{0}$$

Où \otimes est le produit de Kronecker.

$$\Rightarrow \begin{bmatrix} (\mathbf{I}_M \otimes \mathbf{H}_L^1)^T \\ \vdots \\ (\mathbf{I}_M \otimes \mathbf{H}_L^d)^T \end{bmatrix} \mathbf{h}^T = \mathbf{0}$$

Pour plus de clarté, posons :

$$\mathbf{Q} = \begin{bmatrix} (\mathbf{I}_M \otimes \mathbf{H}_L^1)^T \\ \vdots \\ (\mathbf{I}_M \otimes \mathbf{H}_L^d)^T \end{bmatrix} \quad \text{et} \quad \mathbf{f} = \mathbf{h}^T = \begin{bmatrix} \mathbf{h}_1^T \\ \vdots \\ \mathbf{h}_M^T \end{bmatrix}$$

Pour trouver le vecteur \mathbf{h} , il s'agit de résoudre le système d'équations linéaires suivant :

$$\mathbf{Qf} = \mathbf{0} \quad (2.20)$$

D'autre part, pour pouvoir identifier $\mathbf{H}_z(z)$ à une matrice constante inversible près, il faut que $\mathbf{H}_z(z)$ soit de rang plein pour tout z i.e. irréductible. Il est montré dans [29] que si l'on arrive à garantir le rang plein de la matrice $\mathbf{H}(0) = \lim_{z \rightarrow \infty} \mathbf{H}_z(z)$ (pour rappel $\mathbf{H}_z(z) = \sum_{l=0}^{L-1} \mathbf{H}(l)z^{-l}$), on peut en déduire que $\mathbf{H}_z(z)$ est de rang plein pour tout z .

Pour cela, considérons la matrice $\mathbf{H}(0)$,

$$\mathbf{H}(0) \triangleq \begin{bmatrix} h_{11}(0) & \cdots & h_{1M}(0) \\ \vdots & \ddots & \vdots \\ h_{N1}(0) & \cdots & h_{NM}(0) \end{bmatrix}$$

Imposons la contrainte linéaire suivante qui force $\mathbf{H}(0)$ à être de rang plein :

$$\mathbf{H}(0) \triangleq \begin{bmatrix} 1 & 0 & \cdots & 0 \\ h_{21}(0) & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{M1}(0) & h_{M2}(0) & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1}(0) & \cdots & \cdots & h_{NM}(0) \end{bmatrix} \quad (2.21)$$

Soit, pour tout $j = 1, \dots, M$, $\mathbf{f}_j = \mathbf{h}_j^T$, le vecteur \mathbf{f} s'écrira : $\mathbf{f} = [\mathbf{f}_1, \dots, \mathbf{f}_M]^T$.

A présent, enlevons tous les éléments du vecteur \mathbf{f} correspondants aux valeurs 1 et 0 dans (2.21) et définissons les intervalles suivants :

$$\mathcal{I}_j = \{2 + (j-1)L : jL\} \quad \text{et} \quad \mathcal{D}_j = \{2 + (j-1)L : NL\}$$

Nous obtenons le nouveau vecteur $\tilde{\mathbf{f}} = [\tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_M]^T$ tel que ³,

$$\begin{aligned} \tilde{\mathbf{f}}_1 &= \mathbf{f}_1(\mathcal{D}_1) \\ \tilde{\mathbf{f}}_2 &= \mathbf{f}_2(\mathcal{I}_1, \mathcal{D}_2) \\ &\vdots \\ \tilde{\mathbf{f}}_M &= \mathbf{f}_M(\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_{M-1}, \mathcal{D}_M) \end{aligned}$$

De la même manière, soient :

- $\tilde{\mathbf{Q}}$ une matrice construite à partir de la matrice \mathbf{Q} dont on enlève les colonnes correspondantes aux valeurs égales à 1 des composantes du vecteur \mathbf{f} .
- $\tilde{\mathbf{q}}_1$, un vecteur égal à la somme des colonnes de \mathbf{Q} correspondantes aux valeurs égales à 1 des composantes du vecteur \mathbf{f} .

L'équation (2.20) devient,

$$\tilde{\mathbf{Q}}\tilde{\mathbf{f}} + \tilde{\mathbf{q}}_1 = \mathbf{0}$$

Finalement, il suffit de résoudre l'équation suivante pour estimer tous les filtres désirés,

$$\tilde{\mathbf{f}} = -\tilde{\mathbf{Q}}^\# \tilde{\mathbf{q}}_1$$

³Par convention, la notation $\tilde{\mathbf{f}} = \mathbf{f}(\mathcal{I})$ où \mathcal{I} représente un intervalle donné signifie que le vecteur $\tilde{\mathbf{f}}$ prend toutes les valeurs de \mathbf{f} dans l'intervalle \mathcal{I} .

Où le symbole[#] représente la matrice pseudo-inverse.

A ce niveau, la matrice $\mathbf{H}_z(z)$ est identifiée à une matrice constante inversible prés. Si l'on désire reconstruire nos sources, il faut appliquer un des algorithmes de SAS pour des mélanges instantanés ([1], [2], ...).

2.3.3 Séparation suivie d'une égalisation

Une autre classe d'algorithmes de SAS à partir de mélanges convolutifs est constituée par des algorithmes qui effectuent une séparation des signaux puis une égalisation du canal. Nous allons résumer, dans ce qui suit, deux techniques de second ordre,

- La première présentée par Yingbo Hua *et al.* en 2003 dans [30]. Cette technique est appelée, *Identification aveugle de canaux RIF MIMO par décorrélation de sous-canaux* (BIDS-1)⁴.
- La deuxième technique, présentée par Chor Tin Ma *et al.* en 2000 dans [31], est appelée *faisceau de matrices* (*matrix pencil* en anglais) (MP).

De même, pour un panorama plus large, nous avons développé une méthode de séparation-égalisation appelée *déconvolution MIMO au second ordre* (*second order multi output deconvolution* en anglais) (SOMOD) [32]. Plusieurs aspects de cette technique seront repris dans le chapitre 3.

ALGORITHME BIDS-1 :

L'algorithme BIDS-1 [30] consiste à partitionner les sorties du canal original $\mathbf{H}_z(z)$ en un ensemble de sous-canaux pour lesquels on construit les décorrélateurs $\mathbf{G}_i(z)$ correspondants, comme le montre la figure 2.3.

Modèle proposé et hypothèses :

Considérons, tout d'abord, le système sur-déterminé, M -sources, N -observations, suivant :

$$\mathbf{x}(n) = \sum_{l=0}^{L-1} \mathbf{H}(l)\mathbf{s}(n-l) + \mathbf{n}(n) = [\mathbf{H}_z(z)]\mathbf{s}(n) + \mathbf{n}(n).$$

⁴L'auteur présente, dans la même publication, une autre méthode BIDS-2 qui commence par égaliser le canal puis sépare les signaux.

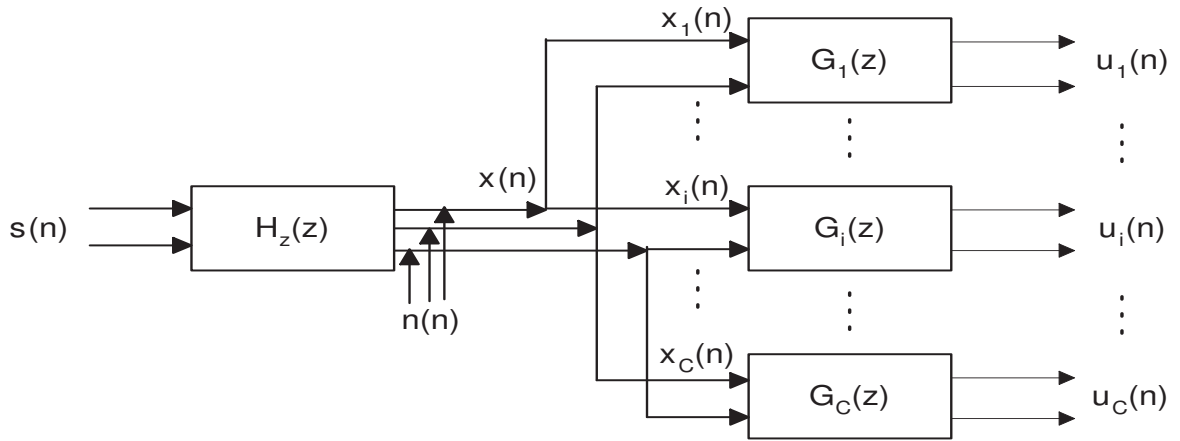


FIG. 2.3 – Illustration de l'algorithme BIDS-1.

La matrice de densité spectrale de puissance des observations sera :

$$\mathbf{S}_{\mathbf{xx}}(z) = \mathbf{H}_z(z)\mathbf{S}_{\mathbf{ss}}(z)\mathbf{H}_z(z^{-1})^H + \mathbf{S}_{\mathbf{nn}}(z) \quad (2.22)$$

où $\mathbf{S}_{\mathbf{ss}}(z)$ et $\mathbf{S}_{\mathbf{nn}}(z)$ sont, respectivement, les matrices de densité spectrale de puissance des sources et du bruit. En supposant que les sources formant le vecteur $\mathbf{s}(n)$ sont mutuellement décorrélées, la matrice $\mathbf{S}_{\mathbf{ss}}(z)$ est diagonale. De même, le vecteur bruit $\mathbf{n}(n)$ est décorrélé par rapport à $\mathbf{s}(n)$.

Soit \mathbf{S}_i , une matrice de sélection de taille $[M \times N]$. L'algorithme consiste à diviser les observations en C sous-groupes de taille M :

$$\mathbf{x}_i(n) \triangleq \mathbf{S}_i \mathbf{x}(n) \quad \text{où } i = 1, \dots, C$$

tel que,

$$C \triangleq \frac{N!}{(N-M)!M!}$$

Pour chaque i , l'algorithme BIDS-1 doit trouver un décorrélateur $\mathbf{G}_i(z)$ à M entrées⁵, tel que,

$$\mathbf{u}_i(n) = [\mathbf{G}_i(z)]\mathbf{x}_i(n). \quad (2.23)$$

La matrice d'auto-corrélation de $\mathbf{u}_i(n)$ est définie par :

$$\mathbf{R}_{\mathbf{u}_i\mathbf{u}_i}(\tau) \triangleq E[\mathbf{u}_i(n)\mathbf{u}_i^H(n-\tau)]$$

⁵Chaque $\mathbf{G}_i(z)$ peut avoir plus d'entrées que de sorties en fonction de la taille de $\mathbf{H}_z(z)$

et la matrice de densité spectrale de puissance sera,

$$\mathbf{S}_{\mathbf{u}_i \mathbf{u}_i}(z) \triangleq \sum_{\tau=-\infty}^{\infty} \mathbf{R}_{\mathbf{u}_i \mathbf{u}_i}(\tau) z^{-\tau}$$

Le décorrélateur $\mathbf{G}_i(z)$ est construit de manière à ce que la matrice de densité spectrale de puissance $\mathbf{S}_{\mathbf{u}_i \mathbf{u}_i}(z)$ soit *diagonale*⁶. De façon générale,

$$\mathbf{G}_i(z) = \sum_{l=0}^{L_G-1} \mathbf{G}_i(l) z^{-l}$$

$\mathbf{u}_i(n)$ peut donc s'écrire :

$$\mathbf{u}_i(n) = \begin{bmatrix} \mathbf{G}_i(0) & \cdots & \mathbf{G}_i(L_G - 1) \end{bmatrix} \begin{bmatrix} \mathbf{x}_i(n) \\ \vdots \\ \mathbf{x}_i(n - L_G + 1) \end{bmatrix}$$

ou d'une manière plus compacte,

$$\mathbf{u}_i(n) = \tilde{\mathbf{G}}_i \tilde{\mathbf{x}}_i(n). \quad (2.24)$$

La matrice d'auto-corrélation de $\mathbf{u}_i(n)$ peut être estimée par :

$$\hat{\mathbf{R}}_{\mathbf{u}_i \mathbf{u}_i}(\tau) = \frac{1}{N_s} \sum_{n=0}^{N_s-1} \mathbf{u}_i(n) \mathbf{u}_i^H(n - \tau)$$

où N_s est le nombre d'échantillons. En utilisant (2.24), on obtient,

$$\hat{\mathbf{R}}_{\mathbf{u}_i \mathbf{u}_i}(\tau) = \tilde{\mathbf{G}}_i \hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i}(\tau) \tilde{\mathbf{G}}_i^H$$

$\hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i}(\tau)$ est la matrice d'auto-corrélation de $\tilde{\mathbf{x}}_i(n)$.

$$\hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i}(\tau) \triangleq \begin{bmatrix} \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau) & \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau + 1) & \cdots & \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau + L_G - 1) \\ \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau - 1) & \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau) & \cdots & \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau + L_G - 2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau - L_G + 1) & \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau - L_G + 2) & \cdots & \hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau) \end{bmatrix}$$

où,

$$\hat{\mathbf{R}}_{\mathbf{x}_i \mathbf{x}_i}(\tau) = \mathbf{S}_i \hat{\mathbf{R}}_{\mathbf{x} \mathbf{x}}(\tau) \mathbf{S}_i^T$$

⁶ $\mathbf{S}_{\mathbf{u}_i \mathbf{u}_j}(z)$ avec $i \neq j$ doit aussi être diagonale car les \mathbf{u}_i et \mathbf{u}_j sont issues, à un filtre près, des mêmes sources $\mathbf{s}(n)$ et $\mathbf{S}_{\mathbf{s} \mathbf{s}}(z)$ est diagonale.

avec,

$$\hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}(\tau) = \frac{1}{N_s} \sum_{n=0}^{N_s-1} \mathbf{x}(n)\mathbf{x}^H(n-\tau)$$

De la même manière que la méthode sous-espace décrite précédemment 2.3.2 : si le bruit $\mathbf{n}(n)$ est spatialement et temporellement blanc, $E[\mathbf{n}(n)\mathbf{n}^H(n)] = \sigma_n^2 \mathbf{I}_N$, la variance du bruit σ_n^2 correspond à la plus petite valeur propre de $\hat{\mathbf{R}}_{\mathbf{x}\mathbf{x}}(0)$.

Procédure de décorrélation :

Pour construire les différents décorrélateurs, une fonction coût, représentant la moyenne des carrés des valeurs en dehors de la diagonale de $\hat{\mathbf{R}}_{\mathbf{u}_i\mathbf{u}_i}(\tau)$, est définie :

$$E_i = \frac{1}{(L_x - 1)M(M - 1)} \sum_{\tau=0}^{L_x-1} \sum_{j \neq k} (\mathbf{g}_{i,k}^H \hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i}(\tau) \mathbf{g}_{i,j})^2$$

avec,

$$\tilde{\mathbf{G}}_i^H = \begin{bmatrix} \mathbf{g}_{i,1} & \mathbf{g}_{i,2} & \cdots & \mathbf{g}_{i,M} \end{bmatrix}$$

E_i n'est pas une fonction quadratique de $\tilde{\mathbf{G}}_i$ mais est quadratique si l'on considère chaque ligne de $\tilde{\mathbf{G}}_i$ séparément. Pour minimiser E_i , l'algorithme, à chaque itération, renouvelle les lignes de $\tilde{\mathbf{G}}_i$ séquentiellement : chaque ligne de $\tilde{\mathbf{G}}_i$ minimise E_i en fixant les autres lignes. Pour éviter les solutions triviales ($\mathbf{g}_{i,k} = \mathbf{0}$, $k = 1, \dots, M$), l'algorithme adopte la contrainte suivante : chaque ligne de $\tilde{\mathbf{G}}_i$ doit aboutir à une valeur constante de la diagonale de $\hat{\mathbf{R}}_{\mathbf{u}_i\mathbf{u}_i}(0)$, i.e.,

$$\mathbf{g}_{i,k}^H \hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i}(0) \mathbf{g}_{i,k} = 1.$$

Cette manière de minimiser par rapport à chaque ligne séparément est un problème quadratique linéaire standard. Cependant, la minimisation globale par rapport à toute la matrice $\tilde{\mathbf{G}}_i$ peut converger vers des minimums locaux. En pratique, plusieurs initialisations de $\tilde{\mathbf{G}}_i$ sont nécessaires pour atteindre un minimum global.

Développons, à présent, la relation (2.23),

$$\mathbf{u}_i(n) = \mathbf{G}_i(z)\mathbf{x}_i(n) = \mathbf{G}_i(z)\mathbf{H}_i(z)\mathbf{x}(n) \quad \text{avec} \quad \mathbf{H}_i(z) = \mathbf{S}_i\mathbf{H}_z(z)$$

Y.Hua montre que si l'on choisit judicieusement le degré de la matrice polynômiale $\mathbf{G}_i(z)$, $\deg(\mathbf{G}_i(z))$, le produit $\mathbf{G}_i(z)\mathbf{H}_i(z)$ est *diagonalisable* par $\mathbf{G}_i(z)$ ⁷.

⁷Une démonstration de ce résultat est détaillée dans [30].

En d'autres termes, en diagonalisant la matrice de densité spectrale de puissance $\mathbf{S}_{\mathbf{u}_i\mathbf{u}_i}(z)$, on peut trouver une matrice $\mathbf{G}_i(z)$ de façon à ce que $\mathbf{G}_i(z)\mathbf{H}_i(z)$ soit diagonale à une matrice de permutation des lignes près \mathbf{P}_i . Le problème est que cette permutation est *différente* pour chaque i , i.e. chaque sous canal.

Il montre aussi que, si la matrice $\mathbf{H}_i(z)$ n'est pas singulière, alors la matrice $\mathbf{G}_i(z)$ correspondante l'est aussi. Dans ses algorithmes, il ne considère que les décorrélateurs dont les matrices $\mathbf{G}_i(z)$ sont non-singulières. Pour cela, on définit pour $i = 1, \dots, C'$, ($N \leq C' \leq C$)⁸,

$$\begin{aligned} \mathbf{G}_i(z)\mathbf{H}_i(z) &\triangleq \mathbf{K}_i(z) = \text{diag} \left[k_{i,1}(z) \quad k_{i,2}(z) \quad \cdots \quad k_{i,M}(z) \right] \\ \mathbf{k}_j(z) &= \left[k_{1,j}(z) \quad k_{2,j}(z) \quad \cdots \quad k_{C',j}(z) \right] \end{aligned}$$

Dans [30], il est démontré que, pour tout $j = 1, \dots, M$, un zéro de $\mathbf{k}_j(z)$ est un zéro de $\mathbf{H}(z)$.

Couplage des sorties des différents décorrélateurs :

Pour retrouver les permutations \mathbf{P}_i ou du moins les rendre égales, considérons la matrice de densité spectrale croisée entre les vecteurs \mathbf{u}_1 et \mathbf{u}_i , et en utilisant (2.22),

$$\mathbf{S}_{\mathbf{u}_1\mathbf{u}_i}(z) = \mathbf{G}_1(z)\mathbf{H}_1(z)\mathbf{S}_{\text{ss}}(z)\mathbf{H}_i(z^{-1})^H\mathbf{G}_i(z^{-1})^H \quad (2.25)$$

Dans ce qui suit, $\mathbf{G}_i(z)$ représente un décorrélateur parfait tel que la matrice $\mathbf{G}_i(z)\mathbf{H}_i(z)$ est parfaitement diagonale. $\mathbf{G}_i(z)$ à une permutation près sera notée :

$$\hat{\mathbf{G}}_i(z) = \mathbf{P}_i\mathbf{G}_i(z) \quad (2.26)$$

L'expression (2.25) devient,

$$\begin{aligned} \mathbf{S}_{\mathbf{u}_1\mathbf{u}_i}(z) &= \mathbf{P}_1^T \hat{\mathbf{G}}_1(z)\mathbf{H}_1(z)\mathbf{S}_{\text{ss}}(z)\mathbf{H}_i(z^{-1})^H \hat{\mathbf{G}}_i(z^{-1})^H \mathbf{P}_i \\ &= \mathbf{P}_1^T \hat{\mathbf{G}}_1(z)\mathbf{S}_1\mathbf{S}_{\text{xx}}(z)\mathbf{S}_i^T \hat{\mathbf{G}}_i(z^{-1})^H \mathbf{P}_i \\ &= \mathbf{P}_1^T \hat{\mathbf{S}}_{\mathbf{u}_1\mathbf{u}_i}(z)\mathbf{P}_i \end{aligned}$$

avec $\hat{\mathbf{S}}_{\mathbf{u}_1\mathbf{u}_i}(z) = \hat{\mathbf{G}}_1(z)\mathbf{S}_1\mathbf{S}_{\text{xx}}(z)\mathbf{S}_i^T \hat{\mathbf{G}}_i(z^{-1})^H$ qui peut être calculé à partir des valeurs des observations. Rappelons que la matrice $\mathbf{S}_{\mathbf{u}_1\mathbf{u}_i}(z)$ est diagonale. Par dualité, on peut

⁸ C' est le nombre total de matrices $\mathbf{G}_i(z)$ non-singulières.

écrire dans le domaine temporel,

$$\mathbf{R}_{\mathbf{u}_1\mathbf{u}_i}(\tau) = \mathbf{P}_1^T \hat{\mathbf{R}}_{\mathbf{u}_1\mathbf{u}_i}(\tau) \mathbf{P}_i \quad (2.27)$$

où $\mathbf{R}_{\mathbf{u}_1\mathbf{u}_i}(\tau)$ est diagonale, et, par définition⁹,

$$\hat{\mathbf{S}}_{\mathbf{u}_1\mathbf{u}_i}(z) = \sum_{\tau=-\infty}^{\infty} \hat{\mathbf{R}}_{\mathbf{u}_1\mathbf{u}_i}(\tau) z^{-\tau}$$

Pour identifier les éléments d'une matrice dont les valeurs sont supérieures à un certain seuil ϵ (les éléments non nuls dans le cas idéal), introduisons l'opérateur η suivant :

$$\eta(x) = \begin{cases} 1 & 0 < \epsilon \leq |x| \\ 0 & \text{sinon} \end{cases}$$

D'une manière générale, pour une matrice \mathbf{A} dont les éléments sont notés a_{ij} ,

$$(\eta(\mathbf{A}))_{ij} = \eta(a_{ij})$$

Il est facile de déduire d'après (2.27) :

$$\eta(\mathbf{R}_{\mathbf{u}_1\mathbf{u}_i}(\tau)) = \mathbf{P}_1^T \eta(\hat{\mathbf{R}}_{\mathbf{u}_1\mathbf{u}_i}(\tau)) \mathbf{P}_i = \mathbf{I}_M$$

Dans le cas idéal (décorrélation parfaite $\epsilon = 0$), en sommant sur tous les τ et en élevant au carré¹⁰,

$$\mathbf{P}_{1,i} = \eta\left(\sum_{\tau} |\hat{\mathbf{R}}_{\mathbf{u}_1\mathbf{u}_i}(\tau)|^2\right) = \mathbf{P}_1 \mathbf{P}_i^T$$

En utilisant (2.26), nous aurons :

$$\mathbf{P}_{1,i} \hat{\mathbf{G}}_i(z) = \mathbf{P}_1 \mathbf{G}_i(z)$$

Finalement, la matrice $\mathbf{P}_{1,i} \hat{\mathbf{G}}_i(z)$ est égale, pour tout i , au décorrélateur idéal $\mathbf{G}_i(z)$ à une matrice de permutation près *indépendante* de i .

Nous sommes, à présent, prêts pour présenter les étapes qui permettent de retrouver, à une constante et à une permutation près, les sources $\mathbf{s}(n)$ de départ.

D'après l'expression (2.23), on définit pour $i = 1, \dots, C'$:

$$\begin{aligned} \mathbf{u}_i(n) &= \begin{bmatrix} u_{i,1}(n) & u_{i,2}(n) & \cdots & u_{i,M}(n) \end{bmatrix}^T \\ \mathbf{v}_j(n) &= \begin{bmatrix} u_{1,j}(n) & u_{2,j}(n) & \cdots & u_{C',j}(n) \end{bmatrix}^T \end{aligned}$$

⁹Les définitions ci-dessus des matrices de corrélation sont valables lorsque le nombre d'échantillons $N_s = \infty$, alors que l'algorithme utilise un nombre fini d'échantillons.

¹⁰ $|\mathbf{A}|^2$ est la matrice \mathbf{A} avec tous ses éléments a_{ij} élevés au carré.

On peut, donc, écrire :

$$\mathbf{v}_j(n) = \mathbf{k}_j(z)s_j(n)$$

L'équation ci-dessus représente un système RIF SIMO. Si $\mathbf{H}_z(z)$ est irréductible ($\mathbf{k}_j(z)$ l'est aussi), nous pouvons appliquer une des méthodes d'identification SIMO (par exemple, la méthode sous-espace [28]) pour reconstituer $s_j(n)$ à partir $\mathbf{v}_j(n)$.

Résumé de l'algorithme BIDS-1

Etape 1 : Calculer $\hat{\mathbf{R}}_{\mathbf{xx}}(\tau) = (1/N_s) \sum_{k=0}^{N_s-1} \mathbf{x}(n)\mathbf{x}^H(n-\tau)$ pour $\tau = 0, 1, \dots, L_x - 1$.

Etape 2 : Enlever l'effet du bruit $\mathbf{n}(n)$ de $\hat{\mathbf{R}}_{\mathbf{xx}}(\tau)$.

Etape 3 : Construire $\hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i}(\tau)$ pour chaque sous-canal i .

Etape 4 : Pour chaque sous-canal i , minimiser l'expression suivante,

$$\frac{1}{(L_x-1)M(M-1)} \sum_{\tau=0}^{L_x-1} \sum_{j \neq k} (\mathbf{g}_{i,k}^H \hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i}(\tau) \mathbf{g}_{i,j})^2$$

sous la contrainte $\mathbf{g}_{i,k}^H \hat{\mathbf{R}}_{\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i}(0) \mathbf{g}_{i,k} = 1$.

Etape 5 : Construire le décorrélateur $\mathbf{G}_i(z)$ à partir de $\tilde{\mathbf{G}}_i^H$ pour chaque i .

Etape 6 : Trouver la matrice de permutation $\mathbf{P}_{1,i}$ pour chaque i .

Etape 7 : Multiplier, à gauche, le vecteur de sortie du décorrélateur

$$\mathbf{u}_i(n) = [\mathbf{G}_i(z)]\mathbf{x}_i(n) \text{ par } \mathbf{P}_{1,i} \text{ pour chaque } i.$$

Etape 8 : Construire le vecteur $\mathbf{v}_j(n) = \begin{bmatrix} u_{1,j}(n) & u_{2,j}(n) & \dots & u_{C',j}(n) \end{bmatrix}^T$ pour chaque source j .

Etape 9 : Appliquer, pour chaque j , une technique SIMO pour estimer $s_j(n)$ en se basant sur le modèle $\mathbf{v}_j(n) = \mathbf{k}_j(z)s_j(n)$.

La figure 2.4 illustre l'algorithme BIDS-1 pour $M = 2$ sources et $N = 3$ capteurs. L'auteur insiste sur un avantage de BIDS-1 par rapport aux autres méthodes comme la méthode sous-espace (vue précédemment) et la méthode de faisceau de matrices que l'on résumera ci-après : la matrice décrivant le canal $\mathbf{H}_z(z)$ n'a pas besoin d'être à colonnes réduites. Cependant, elle reste toujours irréductible (i.e. de rang plein pour tout z). De notre point de vue, cet avantage n'est pas crucial car, en pratique, les divers canaux de propagation sont suffisamment différents les uns des autres et, donc, cette condition est souvent remplie.

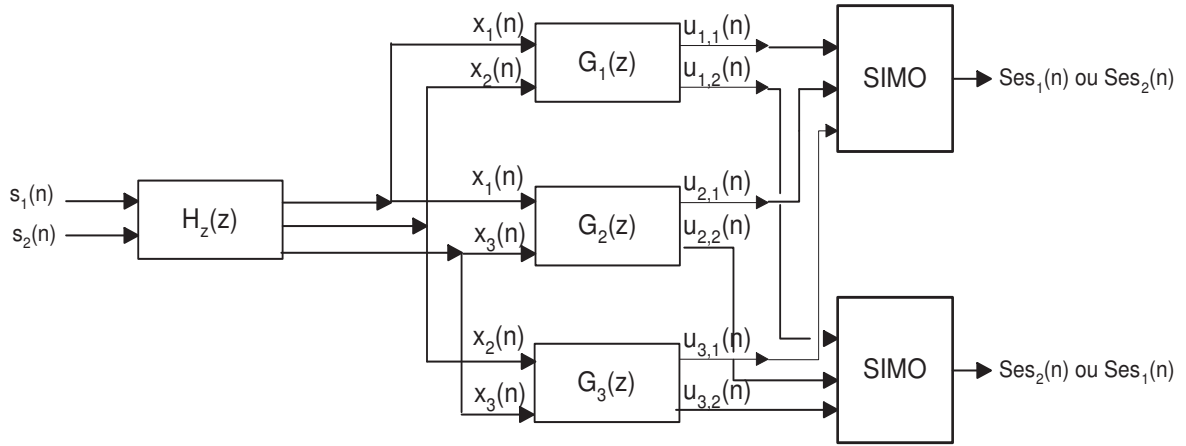


FIG. 2.4 – Illustration de l’algorithme BIDS-1.

ALGORITHME MP :

L’algorithme MP [31] consiste en une décomposition en vecteurs et valeurs propres *généralisés* de matrices d’auto-corrélations des observations à des retards différents.

Reprenons le même modèle présenté dans l’étude de la méthode sous-espace 2.3.2, cette fois-ci, *non-bruité* et adoptons les mêmes hypothèses.

La relation (2.11) peut s’écrire entre un capteur i et une source j ,

$$x_i(n) = \sum_{j=1}^M \sum_{l=0}^{L-1} h_{ij}(l) s_j(n-l) \quad \text{pour } i = 1, \dots, N \quad (2.28)$$

On définit, pour chaque capteur i et $\forall n \in [L' - 1, T - 1]$, le vecteur observation $\mathbf{x}_i(n)$ de dimension $[L' \times 1]$ et pour chaque source j , le vecteur $\mathbf{s}_j(n)$ de dimension $[(L' + L - 1) \times 1]$:

$$\begin{aligned} \mathbf{x}_i(n) &= [x_i(n), \dots, x_i(n - L' + 1)]^T \\ \mathbf{s}_j(n) &= [s_j(n), \dots, s_j(n - (L' - 1) - (L - 1))]^T \end{aligned}$$

Construisons, à présent, par concaténation, les vecteurs $\mathbf{x}(n)$ et $\mathbf{s}(n)$ de tailles respectives $[NL' \times 1]$ et $[M(L' + L - 1) \times 1]$ qui regroupent toutes les observations et tous les signaux sources,

$$\begin{aligned} \mathbf{x}(n) &= [\mathbf{x}_1^T(n), \dots, \mathbf{x}_N^T(n)]^T \\ \mathbf{s}(n) &= [\mathbf{s}_1^T(n), \dots, \mathbf{s}_M^T(n)]^T \end{aligned}$$

On peut formuler, de nouveau, l'équation (2.28) sous la forme matricielle suivante :

$$\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n) \quad (2.29)$$

où

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \cdots & \tilde{\mathbf{H}}_{1M} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{H}}_{N1} & \cdots & \tilde{\mathbf{H}}_{NM} \end{bmatrix} \quad (2.30)$$

avec

$$\tilde{\mathbf{H}}_{ij} = \begin{bmatrix} h_{ij}(0) & \cdots & h_{ij}(L-1) & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & h_{ij}(0) & \cdots & h_{ij}(L-1) \end{bmatrix}$$

Formons, à présent, les matrices d'auto-corrélations à un retard k de $\mathbf{s}(n)$ et $\mathbf{x}(n)$,

$$\mathbf{R}_{\mathbf{ss}}(k) = E[\mathbf{s}(n)\mathbf{s}(n-k)^H]$$

D'après l'hypothèse d'indépendance mutuelle des sources, la matrice $\mathbf{R}_{\mathbf{ss}}(k)$ est bloc diagonale avec des blocs $\mathbf{R}_{\mathbf{s}_j\mathbf{s}_j}(k)$ sur sa diagonale ($j = 1, \dots, M$).

$$\mathbf{R}_{\mathbf{ss}}(k) = \text{diag}\{\mathbf{R}_{\mathbf{s}_j\mathbf{s}_j}(k)\}$$

De même,

$$\begin{aligned} \mathbf{R}_{\mathbf{xx}}(k) &= E[\mathbf{x}(n)\mathbf{x}(n-k)^H] = \tilde{\mathbf{H}} \mathbf{R}_{\mathbf{ss}}(k) \tilde{\mathbf{H}}^H \\ &= \tilde{\mathbf{H}} \text{diag}\{\mathbf{R}_{\mathbf{s}_j\mathbf{s}_j}(k)\} \tilde{\mathbf{H}}^H \end{aligned} \quad (2.31)$$

$\mathbf{R}_{\mathbf{xx}}(k)$ est de taille $[NL' \times NL']$.

Soient $\mathbf{R}_{\mathbf{xx}}(k_1)$ et $\mathbf{R}_{\mathbf{xx}}(k_2)$ les matrices d'auto-corrélation à des retards différents $k_1 \neq k_2$ et considérons le problème des valeurs propres généralisées,

$$\mathbf{R}_{\mathbf{xx}}(k_1)\mathbf{v} = \lambda\mathbf{R}_{\mathbf{xx}}(k_2)\mathbf{v}$$

\mathbf{v} est un vecteur propre généralisé associé à la valeur propre généralisée λ . En combinant avec (2.31), chaque couple (\mathbf{v}, λ) doit satisfaire à la relation suivante ,

$$\tilde{\mathbf{H}} \begin{bmatrix} \mathbf{R}_{\mathbf{s}_1\mathbf{s}_1}(k_1) - \lambda\mathbf{R}_{\mathbf{s}_1\mathbf{s}_1}(k_2) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{R}_{\mathbf{s}_M\mathbf{s}_M}(k_1) - \lambda\mathbf{R}_{\mathbf{s}_M\mathbf{s}_M}(k_2) \end{bmatrix} \tilde{\mathbf{H}}^H \mathbf{v} = \mathbf{0} \quad (2.32)$$

Supposons que l'on ai l valeurs propres généralisées $\lambda_1, \lambda_2, \dots, \lambda_l$ correspondantes à l vecteurs propres généralisés $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_l$.

Il existe une solution triviale à (2.32) qui est $\tilde{\mathbf{H}}^H \mathbf{v}_i = \mathbf{0}$. Dans ce cas, λ_i peut prendre n'importe quelle valeur et il n'y a plus de relation entre \mathbf{v}_i et les signaux sources. Les valeurs propres, répondant à ce cas de figure, sont celles pour lesquelles nous avons l'égalité suivante :

$$\mathbf{v}_i^H \mathbf{R}_{\mathbf{xx}}(0) \mathbf{v}_i = \mathbf{v}_i^H \tilde{\mathbf{H}} \begin{bmatrix} \mathbf{R}_{\mathbf{s}_1 \mathbf{s}_1}(0) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{R}_{\mathbf{s}_M \mathbf{s}_M}(0) \end{bmatrix} \tilde{\mathbf{H}}^H \mathbf{v}_i = \mathbf{0}$$

Parce-que les matrices de covariances des sources sont de rang plein ($\mathbf{R}_{\mathbf{s}_j \mathbf{s}_j}(0)$ pour $j = 1, \dots, M$), on peut conclure que $\tilde{\mathbf{H}}^H \mathbf{v}_i = \mathbf{0}$. Ces valeurs et vecteurs propres triviaux ne sont pas utilisables et seront, donc, éliminés au début de l'algorithme.

Etudions la solution de (2.32) dans deux cas non-triviaux, le premier est lorsque nous avons des valeurs propres *distinctes* non-triviales (le cas le plus favorable) :

Supposons qu'il existe une valeur propre généralisée distincte et non triviale λ_i correspondante au vecteur propre généralisé \mathbf{v}_i . C'est la situation désirée qui permet de séparer un des M signaux sources. Pour montrer cela, écrivons :

$$\tilde{\mathbf{H}}^H \mathbf{v}_i = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}$$

L'équation (2.32) peut être décomposée comme suit :

$$(\mathbf{R}_{\mathbf{s}_j \mathbf{s}_j}(k_1) - \lambda_i \mathbf{R}_{\mathbf{s}_j \mathbf{s}_j}(k_2)) \mathbf{u}_j = \mathbf{0} \quad \text{pour } j = 1, \dots, M$$

L'auteur, C.Tin Ma [31], montre que si λ_i n'est pas une solution triviale, alors elle satisfait,

$$\mathbf{R}_{\mathbf{s}_j \mathbf{s}_j}(k_1) \mathbf{u}_j = \lambda_i \mathbf{R}_{\mathbf{s}_j \mathbf{s}_j}(k_2) \mathbf{u}_j$$

pour un *seul* et *unique* j tel que $\mathbf{u}_j \neq \mathbf{0}$.

A présent, pour extraire les signaux, on utilise le vecteur propre obtenu \mathbf{v}_i :

$$y_i(n) = \mathbf{v}_i^H \mathbf{x}(n) = \mathbf{v}_i^H \tilde{\mathbf{H}} \mathbf{s}(n) = \mathbf{u}_j^H \mathbf{s}_j(n) \quad (2.33)$$

$y_i(n)$ est, donc, une version *filtrée* de la source $s_j(n)$.

Le deuxième cas non-trivial est lorsque nous obtenons des valeurs propres *répétitives* (non-distinctes) : ce cas est purement théorique, il ne peut survenir que lorsque des sources ont des matrices de densité spectrale de puissance identiques, et donc leurs matrices d'auto-corrélation sont égales.

Comme le montre (2.33), le vecteur propre généralisé correspondant à $\mathbf{R}_{s_m s_m}(k_1)$ et $\mathbf{R}_{s_m s_m}(k_2)$ va extraire de multiples versions des signaux sources à des instants différents. Chaque version contient une information distincte émanant d'une source unique. Nous devons, donc, classer les signaux $y_i(n)$, ou, d'une manière équivalente, grouper les vecteurs propres généralisés en fonction des signaux sources qu'ils véhiculent.

Pour cela, utilisons l'hypothèse d'indépendance mutuelle des signaux sources au second ordre. Les $s_j(n)$, ($j = 1, \dots, M$) sont, donc, mutuellement décorrélés. De même, des versions $y_i(n)$ de différents signaux sources sont mutuellement décorrélés. Par contre, des versions différentes (à des instants distincts) de la même source doivent être, en principe, *corrélés*.

Considérons les corrélations croisées entre deux signaux obtenus $y_i(n)$ et $y_j(n - k)$,

$$E[y_i(n)y_j(n - k)] = \mathbf{v}_i^H E[\mathbf{x}(n)\mathbf{x}(n - k)^H] \mathbf{v}_j = \mathbf{v}_i^H \mathbf{R}_{\mathbf{xx}}(k) \mathbf{v}_j$$

Soit la matrice des corrélations croisées \mathbf{R}_y de taille $[l \times l]$ dont les éléments sont :

$$\mathbf{R}_y(i, j) = \max_{-k_0 \leq k \leq k_0} |\mathbf{v}_i^H \mathbf{R}_{\mathbf{xx}}(k) \mathbf{v}_j|$$

où k_0 est assez grand pour couvrir tous les retards possibles des versions filtrées des signaux sources. On peut déterminer, à l'aide de cette matrice, si deux signaux $y_i(n)$ obtenus sont issus de la même source.

L'auteur adopte une approche de réduction de la dimension [33] pour classer tous les vecteurs propres généralisés au sein de M groupes. Au début, les l vecteurs appartiennent à l groupes différents contenant un seul vecteur. A chaque étape, l'algorithme réduit le nombre de groupes de un. Ceci est réalisé par l'union de deux groupes ayant la plus forte corrélation. La corrélation croisée entre deux groupes est définie par la valeur médiane parmi les éléments de la matrice \mathbf{R}_y correspondants à tous les signaux des deux groupes.

Cette procédure est répétée autant de fois jusqu'à ce que tous les l vecteurs propres généralisés soient classés dans M groupes. Comme pour l'algorithme BIDS-1, chaque groupe, à l'aide d'une méthode d'identification SIMO (par exemple, la méthode sous-espace [28]), va permettre l'extraction d'un seul signal parmi les M sources.

Le contexte dans lequel notre travail s'inscrit est la deuxième classe de techniques qui réalisent une séparation, à l'ordre deux, à partir de mélanges convolutifs suivie d'une égalisation pour retrouver les canaux manquants. Alors que toutes les méthodes de cette classe se basent sur une diagonalisation de matrices de corrélation ou de covariances des données dans leur structure générale comme les deux méthodes BIDS-1 et MP, décrites précédemment, la technique que nous avons développé utilise une bloc diagonalisation conjointe de matrices de corrélation à différents retards ou de matrices de covariances. Notre travail est, dans une première étape, l'utilisation et la contribution dans le développement de techniques de bloc diagonalisation conjointe de ces matrices pour retrouver les signaux sources à un filtre près, ce qui permet d'isoler les mélanges, à différents instants, d'une même source. Dans une seconde étape, notre travail consiste à retrouver les différents canaux, ce qui suppose une estimation, au préalable, de leurs ordres. Rappelons que toute cette étude est réalisée dans un contexte *aveugle* où les seules données disponibles sont les mélanges, à différents instants, des signaux sources.

Chapitre 3

Séparation à un filtre prés

Nous allons, à travers ce chapitre, présenter notre contribution dans l'étude de la séparation aveugle MIMO à partir de mélanges convolutifs qui consiste à séparer les différents signaux mélangés à *un filtre prés*. Notre travail s'inscrit, donc, dans la deuxième classe de méthodes introduite en 2.3.3 qui réalisent, d'abord, une séparation puis si, l'application l'exige, une égalisation SIMO. Nous présenterons deux méthodes différentes en fonction des données statistiques disponibles pour le problème considéré. La première utilise des matrices d'auto-corrélation des observations dans le cas de signaux stationnaires [34, 35]. La deuxième utilise les matrices de covariance de données non-stationnaires [36]. Toutes ces statistiques peuvent être traduites, sous certaines conditions et hypothèses, sous la forme de matrices bloc diagonales. Il en résulte que le point commun de ces deux méthodes est leur utilisation de techniques de bloc diagonalisation de matrices de données. Ces techniques feront l'objet d'un vaste panorama, plus en avant, dans le chapitre 4.

3.1 Séparation de signaux stationnaires

Nous avons traité ce problème, en détails, dans [34, 35]. Nous présenterons, ici, un résumé qui illustre un algorithme de bloc diagonalisation de matrices de corrélation à des retards différents.

3.1.1 Position du problème

Considérons le modèle, non-bruité, présenté en 2.3.3 que nous rappelons brièvement pour plus de clarté :

$$x_i(n) = \sum_{j=1}^M \sum_{l=0}^{L-1} h_{ij}(l) s_j(n-l) \text{ for } i = 1, \dots, N$$

Où les $s_j(n)$, $j = 1, \dots, M$, sont les M signaux sources, $x_i(n)$, $i = 1, \dots, N$, sont les N signaux capteurs, h_{ij} est la fonction de transfert entre la j -ème source et le i -ème capteur pendant une durée L .

Les hypothèses sont les mêmes que pour l'algorithme MP, à savoir, la décorrélation mutuelle des sources, et le rang plein de la matrice du canal. De plus, nous supposons que nous sommes en présence de signaux sources stationnaires.

En notation matricielle, notre modèle s'écrit :

$$\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n)$$

avec,

$$\begin{aligned} \mathbf{s}(n) &= [s_1(n), \dots, s_1(n - (L + L' - 1) + 1), \\ &\quad \dots, s_M(n), \dots, s_M(n - (L + L' - 1) + 1)]^T \\ \mathbf{x}(n) &= [x_1(n), \dots, x_1(n - L' + 1), \\ &\quad \dots, x_N(n), \dots, x_N(n - L' + 1)]^T \end{aligned}$$

où,

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \cdots & \tilde{\mathbf{H}}_{1M} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{H}}_{N1} & \cdots & \tilde{\mathbf{H}}_{NM} \end{bmatrix}$$

et

$$\tilde{\mathbf{H}}_{ij} = \begin{bmatrix} h_{ij}(0) & \cdots & h_{ij}(L-1) & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & h_{ij}(0) & \cdots & h_{ij}(L-1) \end{bmatrix}$$

Les matrices $\tilde{\mathbf{H}}$ et $\tilde{\mathbf{H}}_{ij}$ sont de tailles respectives $[NL' \times M(L+L'-1)]$ et $[L' \times (L+L'-1)]$.

La taille de la fenêtre L' est choisie de telle façon que $NL' \geq M(L+L'-1)$.

Comme dans l'algorithme MP, voir 2.3.3, les matrices d'auto-corrélation des observations $\mathbf{R}_{\mathbf{xx}}(k)$ et des sources $\mathbf{R}_{\mathbf{ss}}(k)$ sont reliées par,

$$\mathbf{R}_{\mathbf{xx}}(k) = \tilde{\mathbf{H}} \mathbf{R}_{\mathbf{ss}}(k) \tilde{\mathbf{H}}^H \quad (3.1)$$

$\mathbf{R}_{\mathbf{xx}}(k)$ et $\mathbf{R}_{\mathbf{ss}}(k)$ sont de tailles respectives $[NL' \times NL']$ et $[M(L' + L - 1) \times M(L' + L - 1)]$.

L'indépendance mutuelle des sources nous permet d'affirmer que la matrice $\mathbf{R}_{\mathbf{ss}}(k)$ est bloc diagonale. Nous présentons, dans la section suivante, l'algorithme qui permet de reconstruire les signaux sources à un filtre prés en utilisant la structure bloc diagonale des matrices d'auto-corrélation des sources à différents retards.

3.1.2 Algorithme proposé :

La première étape de notre procédure consiste à blanchir le vecteur des données $\mathbf{x}(n)$ de taille $[NL' \times 1]$. Ceci est réalisé en lui appliquant une *matrice de blanchiment* \mathbf{W} de taille $[M(L' + L - 1) \times NL']$ telle que :

$$\underline{\mathbf{x}}(n) = \mathbf{W}\mathbf{x}(n) = \mathbf{W}\tilde{\mathbf{H}}\mathbf{s}(n)$$

La matrice \mathbf{W} vérifie pour la matrice de covariance des données blanchies $\underline{\mathbf{R}}_{\mathbf{xx}}(0)$,

$$\begin{aligned} \underline{\mathbf{R}}_{\mathbf{xx}}(0) &= E[\underline{\mathbf{x}}(n)\underline{\mathbf{x}}(n)^H] = E[\mathbf{W}\mathbf{x}(n)\mathbf{x}(n)^H\mathbf{W}^H] \\ &= \mathbf{W}\mathbf{R}_{\mathbf{xx}}(0)\mathbf{W}^H = \mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_{\mathbf{ss}}(0)\tilde{\mathbf{H}}^H\mathbf{W}^H \\ &= (\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_{\mathbf{ss}}(0)^{\frac{1}{2}})(\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_{\mathbf{ss}}(0)^{\frac{1}{2}})^H = \mathbf{I}_{M(L'+L-1)} \end{aligned} \quad (3.2)$$

L'équation (3.2) suggère que si \mathbf{W} est une matrice de blanchiment, alors,

$$\mathbf{U} = \mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_{\mathbf{ss}}(0)^{\frac{1}{2}} \quad (3.3)$$

est une matrice unitaire de taille $[M(L' + L - 1) \times M(L' + L - 1)]$ où $\mathbf{R}_{\mathbf{ss}}(0)^{\frac{1}{2}}$ est bloc diagonale¹.

La matrice de blanchiment \mathbf{W} peut être obtenue par la décomposition en vecteurs et valeurs propres de la matrice d'auto-corrélation des données $\mathbf{R}_{\mathbf{xx}}(0)$. Cela a, déjà, été fait dans l'algorithme d'identification aveugle au second ordre (SOBI) dans le cas de

¹La racine carrée d'une matrice bloc diagonale est bloc diagonale.

mélanges instantanés [1].

Considérons, maintenant, les matrices d'auto-corrélation "blanchies" $\underline{\mathbf{R}}_{\mathbf{xx}}(k)$ définies par :

$$\forall k \neq 0, \underline{\mathbf{R}}_{\mathbf{xx}}(k) = \mathbf{W}\mathbf{R}_{\mathbf{xx}}(k)\mathbf{W}^H. \quad (3.4)$$

De (3.1) et (3.3), on obtient la relation "clef" suivante :

$$\begin{aligned} \forall k \neq 0 \underline{\mathbf{R}}_{\mathbf{xx}}(k) &= \mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_{\mathbf{ss}}(k)\tilde{\mathbf{H}}^H\mathbf{W}^H \\ &= \mathbf{U}\mathbf{R}_{\mathbf{ss}}(0)^{-\frac{1}{2}}\mathbf{R}_{\mathbf{ss}}(k)\mathbf{R}_{\mathbf{ss}}(0)^{-\frac{H}{2}}\mathbf{U}^H \\ &= \mathbf{U}\mathbf{D}(k)\mathbf{U}^H \end{aligned} \quad (3.5)$$

où nous posons $\mathbf{D}(k) = \mathbf{R}_{\mathbf{ss}}(0)^{-\frac{1}{2}}\mathbf{R}_{\mathbf{ss}}(k)\mathbf{R}_{\mathbf{ss}}(0)^{-\frac{H}{2}}$.

Puisque la matrice \mathbf{U} est unitaire et la matrice $\mathbf{D}(k)$ est bloc diagonale, la relation (3.5) signifie que toute matrice d'auto-corrélation des observations blanchies est bloc diagonale sur la base des colonnes de la matrice \mathbf{U} .

La matrice unitaire peut être, donc, extraite en bloc diagonalisant une matrice $\underline{\mathbf{R}}_{\mathbf{xx}}(k)$. Seulement, si cette dernière s'avère singulière, nous obtiendrions une indétermination. Pour palier à cette situation et minimiser la probabilité d'une indétermination i.e. rendre plus robuste l'extraction de la matrice \mathbf{U} , il est préférable de bloc diagonaliser **conjointement** un ensemble de K matrices d'auto-corrélation blanchies $\{\underline{\mathbf{R}}_{\mathbf{xx}}(k); k = 1, \dots, K\}$

La bloc diagonalisation conjointe² (BDC) est obtenue par la maximisation sous une transformation unitaire du critère suivant,

$$C(\mathbf{U}) \stackrel{\text{def}}{=} \sum_{k=1}^K \sum_{m=1}^M \sum_{i,j=(L'+L-1)(m-1)+1}^{(L'+L-1)m} |\mathbf{u}_i^H \underline{\mathbf{R}}_{\mathbf{xx}}(k) \mathbf{u}_j|^2 \quad (3.6)$$

pour un ensemble de matrices unitaires $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_{M(L'+L-1)}]$.

Il existe un algorithme très efficace pour la BDC basé sur la méthode de Jacobi dans [37, 38]. Il permet la maximisation du critère (3.6) à travers des rotations successives de Givens. Cet algorithme sera détaillé dans la section 4.3.

Une fois la matrice unitaire \mathbf{U} obtenue, les signaux sources seront reconstitués, à un filtre prés, par :

$$\hat{\mathbf{s}}(n) = \mathbf{U}^H \mathbf{W} \mathbf{x}(n)$$

²Des techniques de bloc diagonalisation conjointe seront présentées dans le chapitre 4.

Suivant le modèle proposé ($\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n)$) et la relation (3.3), les signaux reconstitués vérifient,

$$\hat{\mathbf{s}}(n) = \mathbf{R}_{\mathbf{ss}}^{-\frac{1}{2}}(0)\mathbf{s}(n) \quad (3.7)$$

où, l'on rappelle que, la matrice $\mathbf{R}_{\mathbf{ss}}^{-\frac{1}{2}}(0)$ est bloc diagonale.

Remarques importantes :

1. En pratique, seulement M signaux parmi les $M(L' + L - 1)$ signaux reconstitués sont sélectionnés. Ils correspondent à ceux qui donnent les coefficients de corrélation les plus faibles. Notons, que cette information est déjà contenue dans la procédure de la BDC et, donc, ne demande pas un calcul supplémentaire.
2. Dans nos simulations, les matrices d'auto-corrélation spatio-temporelles $\mathbf{R}_{\mathbf{xx}}(k)$ sont remplacées par leurs moyennes temporelles :

$$\hat{\mathbf{R}}_{\mathbf{xx}}(k) = \frac{1}{N_s} \sum_{n=0}^{N_s-1} \mathbf{x}(n)\mathbf{x}^H(n-k)$$

N_s étant le nombre d'échantillons.

3.1.3 Illustration

Dans cette section, on présente une simulation de notre algorithme appliquée aux signaux de paroles³ mélangés. Ce dernier s'est avéré très efficace pour séparer ces signaux à un filtre près. Le premier signal de parole correspond au mot "école", le second correspond au mot "polytechnique". Les paramètres utilisés sont :

- $M = 2$ sources, $N = 3$ capteurs, $L = 3$ et $L' = 4$.
- Les deux signaux sources sont échantillonnés à 22050Hz.
- La matrice de la fonction de transfert du canal simulé est donnée par,

$$\mathbf{H}_z(z) = \begin{bmatrix} 0.4 + 0.35z^{-1} + 0.15z^{-2} & 0.6 + 0.95z^{-2} \\ 0.9 + 0.8z^{-1} + 0.2z^{-2} & 0.3 + 0.75z^{-1} + 0.5z^{-2} \\ 0.05 + 0.2z^{-2} & 0.2 + 0.45z^{-1} + 0.4z^{-2} \end{bmatrix}$$

³Cet exemple d'application n'a pour but que d'illustrer l'algorithme car les signaux de parole sont, en réalité, non stationnaires. Ce qui prodigue un degré de robustesse à notre algorithme.

La figure 3.1 illustre les signaux de parole originaux, la figure 3.2 leurs mélanges convolutifs et enfin la figure 3.3 les signaux reconstitués par l'algorithme. Nous avons représenté seulement deux signaux parmi les douze reconstitués. Ces deux signaux présentent les coefficients d'auto-corrélation les plus faibles. Ce qui revient à dire que ce sont les signaux les moins semblables.

Dans le cas de cette application, la séparation à un filtre prés s'avère suffisante, surtout si la longueur de ce filtre est relativement réduite. En effet, pour les signaux de parole ou les signaux audio, en général, l'oreille s'accommode parfaitement de l'existence de ce filtre. Autrement dit, l'oreille est pourvue, dans son fonctionnement, de mécanismes qui, à notre connaissance, ne sont pas encore assimilés et qui permettent de corriger les effets de ce filtre indésirable.

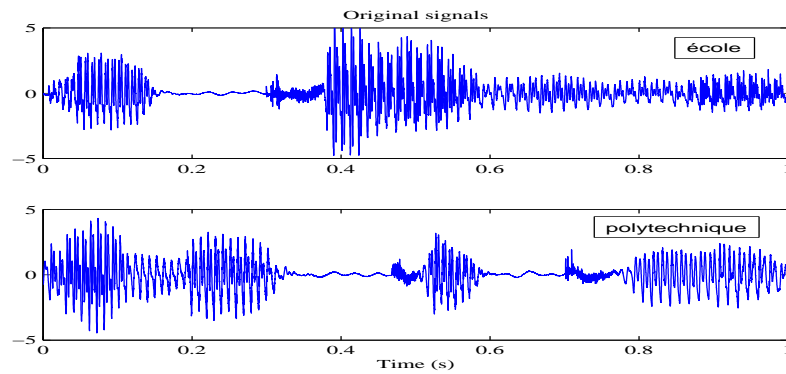


FIG. 3.1 – Signaux sources originaux

3.1.4 Le cas bruité :

Dans un souci de simplicité, tout l'algorithme a été présenté dans le cas non-bruité. Cependant, en pratique, on a affaire, au niveau de chaque capteur i , à un bruit $n_i(n)$ spatialement et temporellement blanc. Le modèle bruité s'écrit alors :

$$x_i(n) = \sum_{j=1}^M \sum_{l=0}^{L-1} h_{ij}(l) s_j(n-l) + n_i(n) \quad \text{for } i = 1, \dots, N$$

En notation matricielle,

$$\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n) + \mathbf{n}(n)$$

avec,

$$\mathbf{n}(n) = [n_1(n), \dots, n_1(n-L'+1), \dots, n_N(n), \dots, n_N(n-L'+1)]^T$$

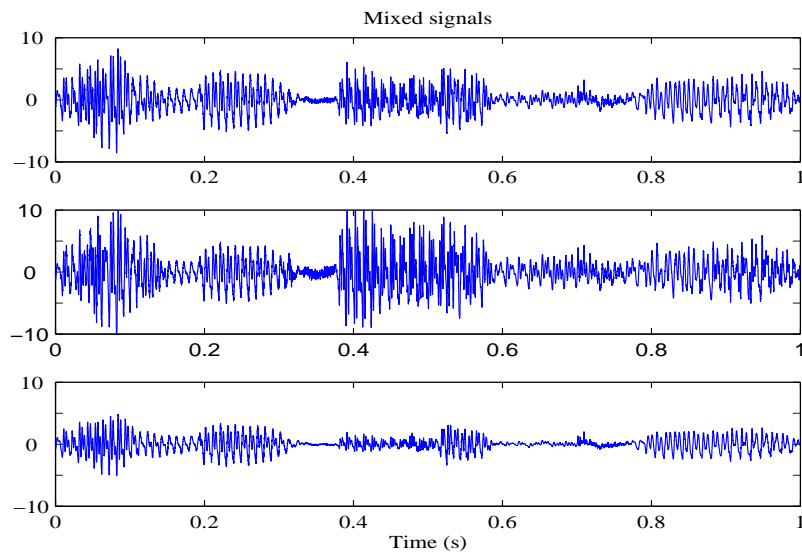


FIG. 3.2 – Signaux de mélange

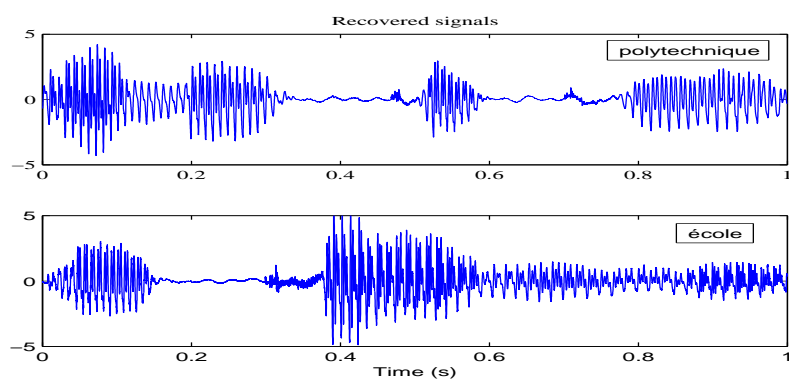


FIG. 3.3 – Deux des douze signaux reconstitués

Les matrices d'auto-corrélation des observations $\mathbf{R}_{xx}(k)$ deviennent,

$$\mathbf{R}_{xx}(k) = \tilde{\mathbf{H}} \mathbf{R}_{ss}(k) \tilde{\mathbf{H}}^H + \mathbf{R}_{nn}(k) \quad (3.8)$$

Où, les $\mathbf{R}_{nn}(k)$ sont les matrices d'auto-corrélation du bruit à des retards différents.

Dans [39], L.Tong *et al.* montrent que la puissance du bruit peut être estimée comme la valeur propre la plus faible de la matrice d'auto-corrélation des observations pour un retard nul. Soit :

$$\mathbf{R}_{xx}(0) = \tilde{\mathbf{H}} \mathbf{R}_{ss}(0) \tilde{\mathbf{H}}^H + \sigma_n^2 \mathbf{I}_{NL'}$$

Pour un retard non nul,

$$\mathbf{R}_{xx}(k) = \tilde{\mathbf{H}} \mathbf{R}_{ss}(k) \tilde{\mathbf{H}}^H + \sigma_n^2 (\mathbf{J}^T)^k$$

Où, la matrice \mathbf{J} de taille $[NL' \times NL']$ s'écrit :

$$\mathbf{J} = \begin{bmatrix} 0 & \cdots & 0 \\ 1 & \ddots & \\ \vdots & \ddots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix}$$

Après cela, les matrices d'auto-corrélation du bruit peuvent être soustraites de celles des observations pour obtenir les "nouvelles" matrices d'auto-corrélation des observations $\mathbf{R}'_{xx}(k)$ auxquelles on applique l'algorithme proposé.

$$\mathbf{R}'_{xx}(k) = \mathbf{R}_{xx}(k) - \sigma_n^2 (\mathbf{J}^T)^k$$

Illustration :

Dans cette section, on présente une simulation de notre algorithme appliquée à la séparation des signaux de paroles de l'exemple présenté en 3.1.3. La figure 3.4 représente l'erreur entre les signaux originaux et les signaux reconstitués. On remarque aisément que lorsqu'on enlève les matrices d'auto-corrélation du bruit de celles des observations, les performances sont sensiblement améliorées surtout pour de faibles valeurs du rapport signal sur bruit.

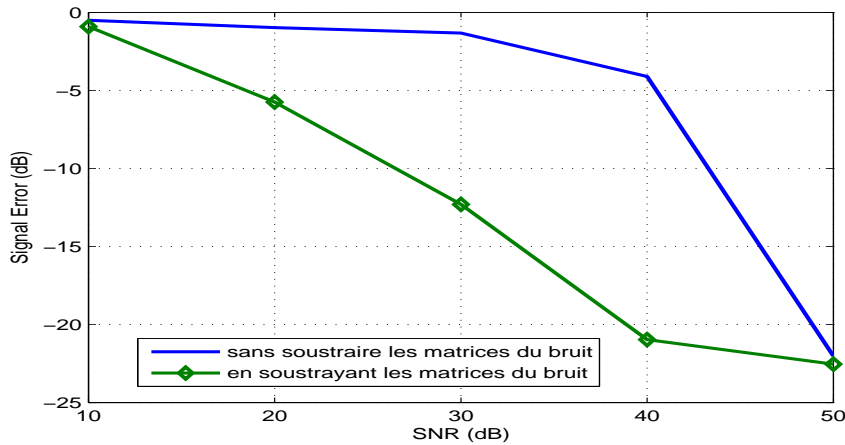


FIG. 3.4 – Effet de la soustraction des matrices d’auto-corrélation du bruit de celles des observations sur les performances de l’algorithme.

3.2 Séparation de signaux non-stationnaires

Pour la séparation de signaux non-stationnaires, il existe une méthode basée sur une technique de vraisemblance bloc gaussienne (*Block Gaussian Likelihood* en anglais)(BGL) présentée par Pham [2, 40] qui consiste en une diagonalisation conjointe de matrices de covariances positives des observations. Notre travail présenté en [36] représente une extension de la technique BGL au cas des mélanges convolutifs. Il est basé sur la bloc diagonalisation conjointe de ces mêmes matrices. La mesure de la "bloc diagonalité" des matrices est directement liée à une fonction objective de vraisemblance et est optimisée sans contrainte orthogonale. Nous allons résumer les principales étapes de ce travail.

3.2.1 Position du problème

Le même modèle présenté dans la section (3.1.1) est repris dans cette technique avec les mêmes hypothèses, sauf que, cette fois ci, chaque signal source est supposé non stationnaire. Cette supposition, quoique moins simple à appréhender, est plus proche de la réalité de la majorité des phénomènes physiques. En effet, elle est moins restrictive que celle qui suppose la stationnarité des signaux sources.

Reprenons l’expression de notre modèle non-bruité,

$$\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n) \quad (3.9)$$

La matrice $\tilde{\mathbf{H}}$ représentant le canal de propagation est de taille $[NL' \times M(L + L' - 1)]$. De la même manière, elle possède plus de lignes que de colonnes ou au moins autant. On suppose, pour cet algorithme, que $\tilde{\mathbf{H}}$ est une matrice carrée, i.e., $NL' = M(L + L' - 1)$, sinon elle peut être rendue carrée en projetant le vecteur des observations $\mathbf{x}(n)$ sur le sous-espace des signaux sources.

3.2.2 Algorithme proposé

Nous sommes, dans ce cas, en présence de signaux non stationnaires durant le temps du phénomène physique considéré $[0, T]$. Nous pouvons, néanmoins tolérer une certaine "stationnarité" des signaux si l'on divise cet intervalle de temps en K sous-intervalles T_1, \dots, T_K de manière ce que la variance des signaux soit considérée comme constante durant un sous-intervalle T_k . La matrice de covariance estimée des observations dans le sous-intervalle T_k est donnée par :

$$\mathbf{R}_{\mathbf{xx}}^k = \frac{1}{N_{T_k}} \sum_{n \in T_k} \mathbf{x}(n)\mathbf{x}(n)^H \quad (3.10)$$

où N_{T_k} est le nombre d'échantillons dans le sous-intervalle T_k .

Suivant notre modèle linéaire (3.9), l'équation ci-dessus peut être formulé comme suit :

$$\mathbf{R}_{\mathbf{xx}}^k = \tilde{\mathbf{H}}\mathbf{R}_{\mathbf{ss}}^k\tilde{\mathbf{H}}^H \quad (3.11)$$

Où les $\mathbf{R}_{\mathbf{ss}}^k$ sont les matrices de covariance estimées des signaux sources. En tenant compte de la décorrélation mutuelle des signaux sources, les $\mathbf{R}_{\mathbf{ss}}^k$ sont des matrices approximativement bloc diagonales avec M blocs diagonaux de taille $[(L + L' - 1) \times (L + L' - 1)]$ chacun, i.e.

$$\mathbf{R}_{\mathbf{ss}}^k \approx \begin{bmatrix} \tilde{\mathbf{R}}_{\mathbf{ss}_1}^k & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}_{\mathbf{ss}_2}^k & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{R}}_{\mathbf{ss}_M}^k \end{bmatrix} \quad (3.12)$$

$\tilde{\mathbf{R}}_{\mathbf{ss}_1}^k, \tilde{\mathbf{R}}_{\mathbf{ss}_2}^k, \dots, \tilde{\mathbf{R}}_{\mathbf{ss}_M}^k$ sont les matrices de covariance "locales" des M sources, k étant l'indice du sous-bloc des données.

Les équations (3.11) et (3.12) signifient que toute matrice de covariance des données est bloc diagonale sur la base des vecteurs colonnes de la matrice $\tilde{\mathbf{H}}$. Cette dernière peut être,

donc, reconstruite en bloc diagonalisant simultanément un ensemble de K matrices de covariance des observations $\tilde{\mathbf{R}}_{\mathbf{xx}}^k$ sur K intervalles, $k = 1, \dots, K$.

3.2.3 Critère de bloc diagonalisation conjointe

Nous allons définir un nouveau critère de minimisation⁴ inspiré de la BGL présenté par Pham [2, 40]. Il fait appel à la notion de divergence ou de distance de Kullback-Liebler entre deux densités normales à moyenne nulle ayant des matrices de covariance respectives \mathbf{R}_a et \mathbf{R}_b :

$$\mathcal{D}(\mathbf{R}_a, \mathbf{R}_b) > 0 \quad (3.13)$$

avec stricte égalité si et seulement si $\mathbf{R}_a = \mathbf{R}_b$. Cette distance est, donc, une mesure légitime de déviation entre deux matrices positives.

En utilisant la relation (3.11), on peut donc déduire une mesure de déviation par rapport à la bloc diagonalisation de la manière suivante :

$$\mathcal{D}(\tilde{\mathbf{H}}^{-1} \tilde{\mathbf{R}}_{\mathbf{xx}}^k \tilde{\mathbf{H}}^{-H}, \tilde{\mathbf{R}}_{\mathbf{ss}}^k) \quad (3.14)$$

Notre algorithme utilise cette mesure de déviation dans un nouveau critère de minimisation qui réalise une bloc diagonalisation approximée d'un ensemble de matrices hermitiennes définies positives \mathbf{M}_k ,

$$\sum_{k=1}^K [\log \det(\text{bdiag}(\mathbf{M}_k)) - \log \det(\mathbf{M}_k)] \quad (3.15)$$

avec,

$$\mathbf{M}_k = \mathbf{B} \tilde{\mathbf{R}}_{\mathbf{xx}}^k \mathbf{B}^H \quad (3.16)$$

sur un ensemble de matrices \mathbf{B} . Avec, $\text{bdiag}(\mathbf{M}_k)$ est la matrice bloc diagonale avec les mêmes bloc diagonaux de taille $[(L + L' - 1) \times (L + L' - 1)]$ que \mathbf{M}_k .

Nous montrons dans la section 4.4.1 du chapitre suivant que pour des matrices définies positives :

$$\det(\mathbf{M}_k) < \det(\text{bdiag}(\mathbf{M}_k)) \quad (3.17)$$

avec une égalité si et seulement si la matrice \mathbf{M}_k est bloc diagonale.

Notre critère (3.15) est bien une mesure de la déviation globale des matrices de la structure

⁴Ce critère est étudié et détaillé dans la section 4.4.1.

bloc diagonale. A la fin, la minimisation de (3.15) nous donne :

$$\mathbf{B} \approx \mathbf{D} \tilde{\mathbf{H}}^{-1} \quad (3.18)$$

\mathbf{D} est une matrice bloc diagonale arbitraire émanant de l'indétermination inhérente du problème de la SAS.

Une fois la matrice \mathbf{B} trouvée, les signaux sources sont reconstitués, à un filtre près par,

$$\hat{\mathbf{s}}(n) = \mathbf{B} \mathbf{x}(n) \quad (3.19)$$

Ces signaux vérifient la relation suivante :

$$\hat{\mathbf{s}}(n) = \mathbf{D} \mathbf{s}(n) \quad (3.20)$$

3.2.4 Illustration

Dans cette section, on présente une simulation de notre algorithme appliqué à des signaux de paroles mélangés. Ce dernier a été très performant pour séparer ces signaux à un filtre près, ce qui est souvent suffisant pour des signaux de parole comme nous l'avons expliqué précédemment en 3.1.3. Les paramètres utilisés sont :

1. $M = 3, N = 2, L = 3$ et $L' = 4$.
2. Les deux signaux de parole sont échantillonnés à une fréquence de 8kHz.
3. La matrice de la fonction de transfert du canal MIMO simulé est :

$$\mathbf{H}_z(z) = \begin{bmatrix} 1 + 0.5z^{-1} + 0.7z^{-2} & 0.1z^{-1} + 0.85z^{-2} \\ 0.8 + 0.7z^{-1} + 0.4z^{-2} & 1 + 0.9z^{-1} \\ 1 + 0.5z^{-1} + 0.3z^{-2} & 0.7 + 0.85z^{-1} + 0.1z^{-2} \end{bmatrix}$$

Les figures 3.5, 3.6 et 3.7 illustrent un déroulement de l'algorithme proposé. Remarquons que, comme dans le cas de l'algorithme utilisant les matrices de corrélation et présenté en (3.1.1), deux signaux sources parmi les douze reconstitués sont représentés. Ces deux signaux sont les moins corrélés entre eux.

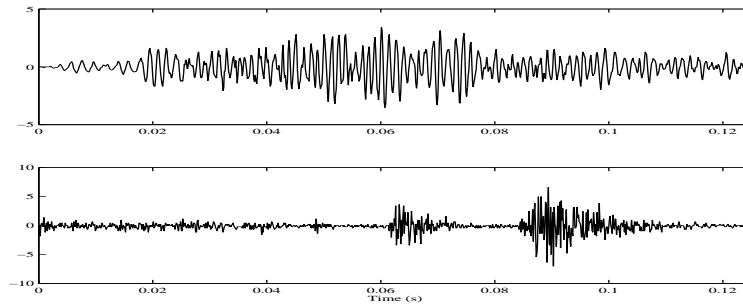


FIG. 3.5 – Signaux de parole originaux.

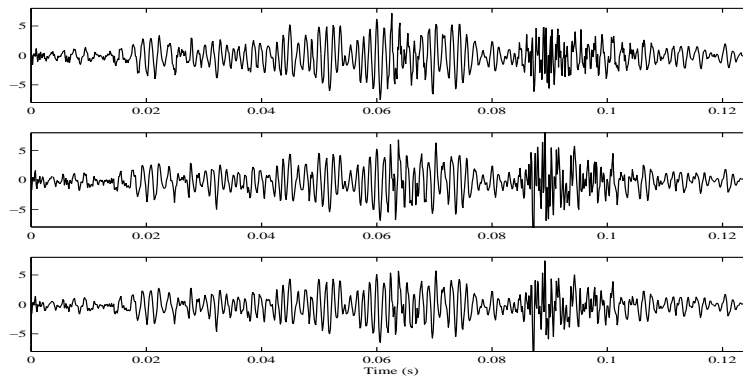


FIG. 3.6 – Signaux de parole mélangés.

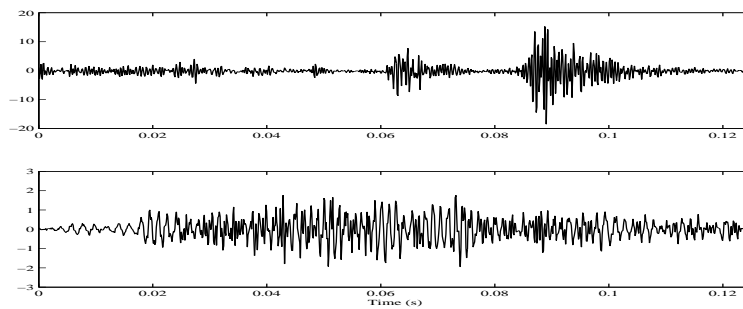


FIG. 3.7 – Deux signaux de parole parmi les douze reconstitués.

Dans la séparation de signaux non-stationnaires à partir de mélanges instantanés, il existe une autre approche basée sur une technique de diagonalisation conjointe de matrices de distributions spatiales temps-fréquence [18, 41, 42]. Ces matrices représentent les signatures temps-fréquence des données, ce qui est une formulation différente des statistiques des données reçues. Il est possible de présenter une extension de notre travail en [42] et, au lieu de diagonaliser conjointement des matrices de distributions spatiales

temps-fréquence dans le cas de mélanges instantanés, on *bloc diagonalise* ces mêmes matrices dans le cas de mélanges convolutifs.

D'autre part, dans le cas de signaux non-stationnaires, et pour l'approche par les covariances (présentée ci-dessus) ainsi que celle utilisant les matrices de distributions spatiales temps-fréquence, les algorithmes n'exigent pas de contrainte orthogonale et donc, aucun blanchiment préalable des données observées n'est nécessaire. Ce qui représente un avantage par rapport à celui présenté en 3.1.

Dans ce chapitre, nous avons présenté la séparation de sources, à un filtre prés, à partir de mélanges convolutifs. En raison du caractère convolutif des mélanges, une bloc diagonalisation conjointe de matrices représentant les statistiques des données s'avère plus appropriée qu'une simple diagonalisation conjointe. Les différentes techniques de bloc diagonalisation conjointe seront abordées, en détail, dans le chapitre suivant 4. Suivant les applications, le filtre manquant peut avoir une influence, plus ou moins, préjudiciable sur le résultat de la séparation. Dans le cas de signaux audibles, ce manque est automatiquement corrigé par l'oreille comme nous l'avons, déjà, expliqué. Par contre, d'autres applications, comme les communications numériques ou le domaine biomédical, ne peuvent s'accommoder de cette lacune. Même pour certaines applications audio comme, écouter de la musique, où on exige une certaine qualité d'audition, ce filtre ne peut être toléré. Dans le chapitre 5, nous présenterons une méthode qui permet de retrouver ce filtre et donc de l'égaliser. En définitive, nous aurons, donc, réalisé une déconvolution complète du système.

Chapitre 4

Bloc diagonalisation conjointe

Tout au long de ce chapitre, nous allons aborder le problème de la bloc diagonalisation conjointe, appelée aussi bloc diagonalisation simultanée, de plusieurs matrices qui se présente dans beaucoup d'applications en traitement du signal. Ce problème a été traité pour la première fois dans [37, 38] pour un ensemble de matrices d'auto-corrélation blanchies. Dans [37], une nouvelle approche de la BDC de matrices singulières a été introduite pour résoudre le problème de l'estimation de la direction d'arrivée. L'utilisation de la BDC pour la séparation de signaux audio à partir de leurs mélanges convolutifs a été initié dans [35, 36] et plus tard, dans [47]. Une bloc diagonalisation a été utilisée pour le contrôle de la puissance et l'élimination des interférences dans les systèmes MIMO dans [48].

Dans ce travail, nous présenterons une approche utilisant les moindres carrés (MC) pour réaliser une BDC. Puis, nous proposerons deux algorithmes itératifs : un "exact" et un "approximatif" pour minimiser le critère des MC. Ces algorithmes sont basés sur la technique de Jacobi qui minimise, d'une manière itérative, une fonction coût par des rotations de Givens. L'avantage des méthodes se basant sur la technique de Jacobi est leur parallélisme, ce qui facilite leur utilisation sur certaines architectures parallèles [49]. Une autre qualité de ces méthodes est leur propriété concernant les erreurs d'arrondi dans le sens où une petite perturbation des éléments des matrices cause, relativement, une petite perturbation de leur valeurs et vecteurs propres [50].

Une autre manière de bloc diagonaliser conjointement un ensemble de matrices, à une matrice de permutation près, est l'utilisation d'une Diagonalisation Conjointe (DC). On

présentera, dans ce chapitre deux approches différentes : la première bloc diagonalise conjointement un ensemble de matrices définies positives sans un blanchiment au préalable. Ce qui est un avantage unique par rapport aux autres méthodes. La deuxième approche utilise un algorithme standard de DC [51] qui permet de faire une BDC à une matrice de permutation près. Nous proposerons, dans ce chapitre, deux algorithmes simples pour enlever, d'une manière aveugle, cette indétermination et achever la BDC. Dans l'algorithme de DC, la transformation unitaire est estimée par un produit de rotations de Givens mais contrairement aux algorithmes de BDC précédents, l'optimisation des angles de rotation donne un critère quadratique au lieu d'un critère quadratique linéaire, ce qui permet de réduire la complexité des calculs.

Enfin, nous présenterons, à l'aide de nombreuses simulations, une étude comparative des différents algorithmes suivant plusieurs paramètres : estimation de l'erreur, taux de convergence, convergence vers des minimums locaux et sensibilité à l'initialisation.

4.1 Position du problème

Soit un ensemble de K matrices carrées, $\mathbf{M}_1, \dots, \mathbf{M}_K$, $\mathbf{M}_k \in \mathbb{C}^{N \times N}$, $k = 1, \dots, K$, qui peuvent être décomposées de la manière suivante :

Pour $k = 1, \dots, K$

$$\mathbf{M}_k = [\mathbf{E}_1, \dots, \mathbf{E}_r] \begin{bmatrix} \mathbf{D}_{k1} & \cdots & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & \cdots & \mathbf{D}_{kr} \end{bmatrix} \begin{bmatrix} \mathbf{E}_1^H \\ \vdots \\ \mathbf{E}_r^H \end{bmatrix} = \mathbf{E} \mathbf{D}_k \mathbf{E}^H \quad (4.1)$$

où $\mathbf{E} = [\mathbf{E}_1, \dots, \mathbf{E}_r]$ est unitaire, \mathbf{E}_j est de taille $[N \times M_j]$ et les \mathbf{D}_{kj} , $j = 1, \dots, r$ sont r matrices carrées de tailles $[M_j \times M_j]$ avec $M_1 + \dots + M_r = N$.

Les matrices \mathbf{M}_k , $k = 1, \dots, K$ sont conjointement bloc diagonalisables sous la transformation unitaire \mathbf{E} , c'est à dire, $\mathbf{E}^H \mathbf{M}_k \mathbf{E}$ sont des matrices bloc diagonales. Remarquons que les matrices \mathbf{M}_k , $k = 1, \dots, K$ ne sont pas, en général, diagonalisables conjointement, sauf pour le cas où $M_1 = \dots = M_r = 1$. De ce fait, la DC est un cas particulier de la BDC lorsque $M_j = 1$, $\forall j$.

Le problème de la BDC consiste à estimer les matrices \mathbf{E} et \mathbf{D}_{kj} , $k = 1, \dots, K$, $j = 1, \dots, r$ connaissant les matrices \mathbf{M}_k , $k = 1, \dots, K$.

Remarquons que la décomposition issue de la BDC n'est pas unique car :

si $\{\mathbf{E}_1, \dots, \mathbf{E}_r, \mathbf{D}_{11}, \dots, \mathbf{D}_{Kr}\}$ est une solution, alors,

$\mathbf{E}'_j = \mathbf{E}_j \mathbf{U}_j$ et $\mathbf{D}'_{kj} = \mathbf{U}_j^H \mathbf{D}_{kj} \mathbf{U}_j$ est une autre solution acceptable, avec les matrices unitaires $\mathbf{U}_j, j = 1, \dots, r$.

En d'autres termes, la matrice \mathbf{E} peut être déterminée, seulement, à une matrice unitaire bloc diagonale près. De plus, n'importe quelle permutation bloc diagonale donnera une solution valide car il n'y a aucun tri ou classement des blocs. Cependant, pour la majorité des applications pratiques, ces indéterminations n'affectent pas le résultat final.

En pratique, les matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ sont données par leurs statistiques estimées avec des erreurs d'estimation dues au bruit et aux effets de la taille finie des échantillons. Ces matrices peuvent être, donc, "approximativement", bloc diagonalisées simultanément. Un algorithme de BDC "viable" doit, donc, fournir une sorte de structure "moyenne" des valeurs et vecteurs propres quand il est appliqué à un ensemble de matrices "presque" bloc diagonalisables conjointement. Une solution optimale basée sur une approche des moindres carrés sera donnée ci-après.¹

4.2 Approche par les Moindres Carrés

La technique MC consiste, ici, à choisir la matrice $[N \times N]$ unitaire \mathbf{E} et les matrices \mathbf{D}_{kj} de tailles $[M_j \times M_j]$ qui minimisent la norme de Frobenius des différences entre les matrices de données \mathbf{M}_k et les matrices exactes données par (4.1), autrement dit :

$$\min_{\mathbf{E}, \mathbf{D}_k} \sum_{k=1}^K \|\mathbf{M}_k - \mathbf{E} \mathbf{D}_k \mathbf{E}^H\|^2 \quad (4.2)$$

Puisque la norme de Frobenius est invariante sous l'effet d'une transformation unitaire, le critère MC en (4.2) est équivalent à :

$$\min_{\mathbf{E}, \mathbf{D}_k} \sum_{k=1}^K \|\mathbf{E}^H \mathbf{M}_k \mathbf{E} - \mathbf{D}_k\|^2 \quad (4.3)$$

La minimisation de (4.3) par rapport à \mathbf{D}_k est obtenue lorsque :

$$\mathbf{D}_k = \text{bdiag}(\mathbf{E}^H \mathbf{M}_k \mathbf{E})$$

¹Une approche similaire a été proposée dans [52] pour la diagonalisation et dans [45] pour la décomposition conjointe Schur.

où $\text{bdiag}(\mathbf{M})$ est une matrice bloc diagonale construite à partir de \mathbf{M} de la manière suivante :

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \cdots & \mathbf{M}_{1r} \\ \vdots & & \vdots \\ \mathbf{M}_{r1} & \cdots & \mathbf{M}_{rr} \end{bmatrix} \implies \text{bdiag}(\mathbf{M}) = \begin{bmatrix} \mathbf{M}_{11} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{rr} \end{bmatrix}. \quad (4.4)$$

où les \mathbf{M}_{jj} sont de tailles $[M_j \times M_j]$, $j = 1, \dots, r$. Encore une fois, utilisons l'invariance de la norme de Frobenius sous l'effet d'une transformation unitaire : la minimisation de (4.3) est équivalente à la maximisation de la partie qui a été éliminée, soit,

$$\max_{\mathbf{E}} \sum_{k=1}^K \|\text{bdiag}(\mathbf{E}^H \mathbf{M}_k \mathbf{E})\|^2 \quad (4.5)$$

En d'autres termes, la BDC de $\{\mathbf{M}_1, \dots, \mathbf{M}_K\}$ consiste en une maximisation, sous la transformation unitaire \mathbf{E} , de la somme des normes des éléments bloc diagonaux. Ceci est équivalent à la minimisation de la somme des normes des éléments en dehors de ses blocs. Le critère (4.5) sera le critère de la BDC par la suite.

4.3 Les algorithmes basés sur la technique de Jacobi

Pour minimiser le critère de la BDC (4.5), nous déterminerons la matrice unitaire \mathbf{E} à partir de produits de rotations de Givens que nous décrirons ci après.

4.3.1 Les rotations de Givens

Dans les algorithmes basés sur Jacobi, une matrice unitaire \mathbf{U} est décomposée en un produit de rotations de Givens élémentaires :

$$\mathbf{U} = \prod_{\text{nb de balayages}} \prod_{1 \leq p < q \leq N} \Theta_{(qp)}$$

où les rotations de Givens élémentaires $\Theta_{(qp)}$ sont définies par des matrices unitaires où tous les éléments diagonaux sont égaux à "1", sauf pour deux éléments égaux à c dans les lignes (et colonnes) p et q . De même, tous les éléments en dehors de la diagonale de $\Theta_{(qp)}$ sont nuls sauf pour deux éléments égaux à s et $-\bar{s}$ respectivement aux positions (p, q) et

(q, p) . \bar{s} représente la valeur conjuguée de s . Les scalaires c et s sont donnés par :

$$\begin{cases} c = \cos \theta \\ s = \sin \theta \exp(i\alpha) \end{cases}$$

avec $i = \sqrt{-1}$. Par la suite, nous décrivons la procédure qui permet de choisir les angles de rotations θ et α à chaque itération de manière à augmenter au maximum la fonction coût (4.5). Pour cela, nous devons définir la transformation orthogonale suivante :

$$\mathbf{M}' = \Theta_{(qp)}^H \mathbf{M} \Theta_{(qp)} \quad (4.6)$$

pour toute matrice $\mathbf{M} \in \mathbb{C}^{N \times N}$. Ces transformations orthogonales modifient uniquement les lignes p et q et les colonnes p and q de \mathbf{M} , tel que :

$$\begin{aligned} \mathbf{M}'(p, j) &= c\mathbf{M}(p, j) - s\mathbf{M}(q, j), \quad j \neq p \\ \mathbf{M}'(j, p) &= c\mathbf{M}(j, p) - \bar{s}\mathbf{M}(j, q), \quad j \neq p \\ \mathbf{M}'(q, j) &= \bar{s}\mathbf{M}(p, j) + c\mathbf{M}(q, j), \quad j \neq q \\ \mathbf{M}'(j, q) &= s\mathbf{M}(j, p) + c\mathbf{M}(j, q), \quad j \neq q \\ \mathbf{M}'(p, p) &= c^2\mathbf{M}(p, p) + |s|^2\mathbf{M}(q, q) - \bar{s}c\mathbf{M}(p, q) - sc\mathbf{M}(q, p) \\ \mathbf{M}'(q, q) &= c^2\mathbf{M}(q, q) + |s|^2\mathbf{M}(p, p) + \bar{s}c\mathbf{M}(q, p) + sc\mathbf{M}(p, q) \\ \mathbf{M}'(p, q) &= cs\mathbf{M}(p, p) + c^2\mathbf{M}(p, q) - s^2\mathbf{M}(q, p) - sc\mathbf{M}(q, q) \\ \mathbf{M}'(q, p) &= \bar{s}c\mathbf{M}(p, p) - \bar{s}^2\mathbf{M}(p, q) + c^2\mathbf{M}(q, p) - c\bar{s}\mathbf{M}(q, q) \end{aligned} \quad (4.7)$$

Remarquons que lorsque p and q balayent le même bloc diagonal, la valeur du critère de la BDC (4.5) reste inchangée, comme cela est établi par le lemme suivant :

Lemme 1 *La norme de la matrice bloc diagonale de \mathbf{M}' est égale à celle de \mathbf{M} , soit :*

$$\|bdiag(\mathbf{M}')\| = \|bdiag(\mathbf{M})\| \quad (4.8)$$

Si p et q balayent le même bloc diagonal, c'est à dire ($1 \leq p, q \leq M_1$, ou $M_1 < p, q \leq M_1 + M_2, \dots$, ou $\sum_{i=1}^{r-1} M_i < p, q \leq N$).

Démonstration : Supposons que $1 \leq p, q \leq M_1$, tel que, uniquement, le premier bloc diagonal de la matrice \mathbf{M} soit modifié². Nous aurons pour $j \neq p$ et $j \neq q$:

$$\begin{aligned} |\mathbf{M}'(p, j)|^2 + |\mathbf{M}'(q, j)|^2 &= |\mathbf{M}(p, j)|^2 + |\mathbf{M}(q, j)|^2 \\ |\mathbf{M}'(j, p)|^2 + |\mathbf{M}'(j, q)|^2 &= |\mathbf{M}(j, p)|^2 + |\mathbf{M}(j, q)|^2 \end{aligned}$$

²Cette supposition peut être généralisée pour les autres bloc diagonaux.

L'équation (4.8) est réduite à l'égalité suivante,

$$\begin{aligned} & |\mathbf{M}'(p, p)|^2 + |\mathbf{M}'(p, q)|^2 + |\mathbf{M}'(q, p)|^2 + |\mathbf{M}'(q, q)|^2 \\ &= |\mathbf{M}(p, p)|^2 + |\mathbf{M}(p, q)|^2 + |\mathbf{M}(q, p)|^2 + |\mathbf{M}(q, q)|^2 \end{aligned}$$

Soit,

$$\mathcal{M}_{pq} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{M}(p, p) & \mathbf{M}(p, q) \\ \mathbf{M}(q, p) & \mathbf{M}(q, q) \end{bmatrix}.$$

Il est facile de voir que :

$$\mathcal{M}'_{pq} = \begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix}^H \mathcal{M}_{pq} \begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix}.$$

Et donc, $\|\mathcal{M}'_{pq}\|^2 = \|\mathcal{M}_{pq}\|^2$. Ce qui correspond à l'égalité précédente. \square

En se basant sur le lemme précédent, nous considérerons dans nos algorithmes de BDC uniquement les indices (p, q) qui balayent deux bloc diagonaux différents. Par exemple, pour $r = 2$, nous choisirons p et q dans le domaine $1 \leq p \leq M_1 < q \leq N$. En d'autres termes, nous choisirons $(p, q) \in \mathcal{I}$ où :

$$\mathcal{I} = \{(p, q) \mid \exists 0 \leq i_p < i_q < r, \sum_{j=1}^{i_p} M_j < p \leq \sum_{j=1}^{i_p+1} M_j \text{ et } \sum_{j=1}^{i_q} M_j < q \leq \sum_{j=1}^{i_q+1} M_j\}. \quad (4.9)$$

4.3.2 Algorithme de BDC exact

La méthode proposée consiste à maximiser, d'une manière itérative, le critère de la BDC (4.5) par des rotations successives de Givens en commençant par $\mathbf{E} = \mathbf{I}$. A la fin de cette procedure itérative, nous aurons :

$$\mathbf{E} = \prod_{\text{nb de balayages}} \prod_{1 \leq p < q \leq N} \Theta_{(qp)}$$

où les $\Theta_{(qp)}$ seront calculées telles que (4.5) est maximum³.

Autrement dit, à chaque itération, les angles (θ, α) seront donnés par :

$$\begin{aligned} (\theta, \alpha) &= \arg \max_{\theta, \alpha} \mathcal{C}(\theta, \alpha) \\ \mathcal{C}(\theta, \alpha) &\stackrel{\text{def}}{=} \sum_{k=1}^K \|\text{bdiag}(\mathbf{M}'_k)\|^2 \end{aligned} \quad (4.10)$$

où \mathbf{M}'_k est définie comme dans (4.6). L'algorithme de la BDC exact peut être résumé comme suit :

$$\begin{aligned} &\mathbf{E} = \mathbf{I} \\ &\text{pour } k = 1, \dots, \text{nb de balayages} \\ &\text{pour } (p, q) \in \mathcal{I} \\ &\quad \Theta_{(qp)} = \arg \max \mathcal{C}(\theta, \alpha) \\ &\quad \mathbf{E} := \mathbf{E}\Theta_{(qp)} \text{ et } \mathbf{M}_k := \Theta_{(qp)}^H \mathbf{M}_k \Theta_{(qp)}, \quad k = 1, \dots, K \end{aligned}$$

Après quelques calculs détaillés dans l'annexe A, la maximisation de $\mathcal{C}(\theta, \alpha)$ est équivalente à la maximisation de la forme quadratique suivante :

$$\max_{\|\mathbf{v}\|=1} (\mathbf{v}^T \mathbf{G} \mathbf{v} + \mathbf{g}^T \mathbf{v}) \quad (4.11)$$

Où,

$$\mathbf{v} = [\cos(2\theta), \sin(2\theta) \cos(\alpha), \sin(2\theta) \sin(\alpha)]^T \quad (4.12)$$

et \mathbf{G} (respectivement \mathbf{g}) est une matrice $[3 \times 3]$ à valeurs réelles (respectivement un vecteur $[3 \times 1]$ à valeurs réelles). Leurs expressions sont données dans l'annexe A.

En utilisant un multiplicateur de Lagrange, la maximisation de l'équation (4.11) implique :

$$2(\mathbf{G} + \lambda \mathbf{I})\mathbf{v} + \mathbf{g} = \mathbf{0} \quad (4.13)$$

et donc,

$$\mathbf{v} = -\frac{1}{2}(\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{g} \quad (4.14)$$

³Le critère est calculé avec les valeurs actuelles des matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ qui sont modifiées à chaque itération en utilisant la transformation orthogonale ci-dessus. Pour simplifier, nous utiliserons les notations $\mathbf{M}_1, \dots, \mathbf{M}_K$ pour désigner les matrices renouvelées.

où λ est un scalaire réel choisi de telle façon que :

$$\|\mathbf{v}\|^2 = 1 \iff \frac{1}{4} \sum_{i=1}^3 \frac{|\mathbf{u}_i^T \mathbf{g}|^2}{(\lambda_i + \lambda)^2} = 1 \quad (4.15)$$

$\{\mathbf{u}_i\}$ et $\{\lambda_i\}$ sont les vecteurs propres et les valeurs propres de la matrice \mathbf{G} .

Ou bien :

$$\mathbf{v} = -\frac{1}{2}(\mathbf{G} - \lambda_i \mathbf{I})^\# \mathbf{g} + c_i \mathbf{u}_i, \quad i = 1, 2, 3 \quad (4.16)$$

Ce qui correspond au cas où λ serait égal à l'opposé d'une valeur propre de \mathbf{G} , soit : $-\lambda_i$, $i = 1, 2, 3$. c_i est une constante réelle choisie de telle manière que $\|\mathbf{v}\| = 1$ et (4.11) est maximum⁴. Elle est donnée par $c_i = \text{sign}(\mathbf{u}_i^T \mathbf{g})(1 - \|(\mathbf{G} - \lambda_i \mathbf{I})^\# \mathbf{g}\|^2/4)^{\frac{1}{2}}$ où $\text{sign}(x)$ indique le signe de x .

Remarque : Dans nos simulations, le cas $\lambda = -\lambda_i$, $i = 1, 2, 3$ n'est pratiquement jamais rencontré en pratique. Ce qui permet de simplifier l'algorithme en résolvant uniquement les équations (4.14) et (4.15).

D'après l'équation (4.15), chercher λ implique de trouver les solutions d'un polynôme de degré 6. En générale, l'équation (4.15) a des racines réelles multiples λ . Ce qui implique plusieurs solutions pour le vecteur \mathbf{v} . Parmi toutes ces solutions possibles, nous garderons celle qui, associée avec le vecteur \mathbf{v} , maximise la valeur de $\mathcal{C}(\theta, \alpha)$, c'est à dire, le vecteur \mathbf{v} qui maximise (4.11).

Dans le cas de données réelles (non complexes), la matrice \mathbf{G} est de dimension $[2 \times 2]$ (voir annexe A). L'équation (4.15) se réduit à un polynôme de degré 4. Ceci peut être obtenu, autrement, en remplaçant dans l'équation (4.11) les rotations de Givens par $t = \tan \theta$.

$$\cos(\theta) = \frac{1}{\sqrt{1+t^2}}, \quad \sin(\theta) = \frac{t}{\sqrt{1+t^2}}$$

Maximiser l'équation (4.11) en fonction de t aboutit aussi à un polynôme de degré 4. Ceci a été utilisé, par exemple, dans [53] pour une décomposition conjointe Schur.

$\mathbf{v} = [\mathbf{v}(1), \mathbf{v}(2), \mathbf{v}(3)]^T$ obtenu, les paramètres des rotations de Givens c et s seront calculés suivant (4.12) de cette manière :

$$c = \sqrt{\frac{\mathbf{v}(1) + 1}{2}}, \quad s = \frac{\mathbf{v}(2) + i\mathbf{v}(3)}{2c} \quad (4.17)$$

Dans ce qui suit et par une "légère" approximation du critère de la BDC, nous présenterons une solution alternative à l'équation (4.11) où il n'y a aucun polynôme à résoudre.

⁴Cette solution n'existe que si $\|(\mathbf{G} - \lambda_i \mathbf{I})^\# \mathbf{g}\|/2 \leq 1$. Remarquons que si c'est le cas, c_i et $-c_i$ satisfont la condition $\|\mathbf{v}\| = 1$. Nous choisirons celle qui maximise (4.11), autrement dit $\text{sign}(c_i) = \text{sign}(\mathbf{u}_i^T \mathbf{g})$.

4.3.3 Algorithme de BDC approximé

Pour simplifier l'algorithme précédent, nous allons approximer $|\mathbf{M}'(p, p)|^2$ (à une constante scalaire indépendante des paramètres de rotation (θ, α) près) par

$$|\mathbf{M}'(p, p)|^2 \approx |[\mathbf{M}\Theta_{(qp)}](p, p)|^2 + |[\Theta_{(qp)}^H \mathbf{M}](p, p)|^2.$$

Cette approximation de $|\mathbf{M}'(p, p)|^2$ est considérée du premier ordre au voisinage du point optimal. Effectivement, en écrivant $\Theta_{(qp)} = \mathbf{I} + \epsilon_{qp}$ (avec $\|\epsilon_{qp}\| \ll 1$ au voisinage du point de convergence), on obtient l'approximation du premier ordre suivante :

$$\begin{aligned} |\mathbf{M}'(p, p)|^2 &\approx |[\mathbf{M} + \epsilon_{qp}^H \mathbf{M} + \mathbf{M}\epsilon_{qp}](p, p)|^2 \\ &\approx |\mathbf{M}(p, p)|^2 + 2\Re\{[\mathbf{M}(p, p)[\epsilon_{qp}^H \mathbf{M}](p, p) + \mathbf{M}(p, p)[\mathbf{M}\epsilon_{qp}](p, p)]\} \\ &\approx |[\mathbf{M}\Theta_{(qp)}](p, p)|^2 + |[\Theta_{(qp)}^H \mathbf{M}](p, p)|^2 - |\mathbf{M}(p, p)|^2 \end{aligned} \quad (4.18)$$

Puisque au premier ordre, nous avons :

$$\begin{aligned} |[\mathbf{M}\Theta_{(qp)}](p, p)|^2 &\approx |\mathbf{M}(p, p)|^2 + 2\Re\{[\mathbf{M}(p, p)[\mathbf{M}\epsilon_{qp}](p, p)]\} \\ |[\Theta_{(qp)}^H \mathbf{M}](p, p)|^2 &\approx |\mathbf{M}(p, p)|^2 + 2\Re\{[\mathbf{M}(p, p)[\epsilon_{qp}^H \mathbf{M}](p, p)]\}. \end{aligned}$$

Le critère de la BDC devient de la forme suivante (voir annexe B) :

$$\mathcal{C}(\theta, \alpha) \approx \tilde{\mathbf{g}}^T \mathbf{v} \quad (4.19)$$

et sa maximisation aboutit aux expressions explicites suivantes :

$$\begin{aligned} \alpha &= \arctan\left(\frac{\Im m(a)}{\Re e(a)}\right), \\ \theta &= \frac{1}{2} \arctan\left(-2 \frac{\Re e(e^{-i\alpha} a)}{b}\right) + \frac{1 - \text{sign}(b)}{2} \frac{\pi}{2} \\ a &= \sum_{k=1}^K \left(\sum_{j \in \mathcal{I}_p} \mathbf{M}_k(p, j) \overline{\mathbf{M}}_k(q, j) + \mathbf{M}_k(j, p) \overline{\mathbf{M}}_k(j, q) \right. \\ &\quad \left. - \sum_{j \in \mathcal{I}_q} \mathbf{M}_k(p, j) \overline{\mathbf{M}}_k(q, j) + \mathbf{M}_k(j, p) \overline{\mathbf{M}}_k(j, q) \right) \\ b &= \sum_{k=1}^K \left(\sum_{j \in \mathcal{I}_p} |\mathbf{M}_k(p, j)|^2 + |\mathbf{M}_k(j, p)|^2 - (|\mathbf{M}_k(q, j)|^2 + |\mathbf{M}_k(j, q)|^2) \right. \\ &\quad \left. - \sum_{j \in \mathcal{I}_q} |\mathbf{M}_k(p, j)|^2 + |\mathbf{M}_k(j, p)|^2 - (|\mathbf{M}_k(q, j)|^2 + |\mathbf{M}_k(j, q)|^2) \right) \end{aligned}$$

où $\Re(a)$ et $\Im(a)$ représentent respectivement les parties réelles et imaginaires de a , et

$$\mathcal{I}_p = \{j \mid \sum_{l=1}^{i_p} M_l < j \leq \sum_{l=1}^{i_p+1} M_l\}, \quad \mathcal{I}_q = \{j \mid \sum_{l=1}^{i_q} M_l < j \leq \sum_{l=1}^{i_q+1} M_l\}$$

i_p et i_q sont définies par (4.9).

4.4 BDC via une Diagonalisation Conjointe

Une autre manière de réaliser, à une matrice de permutation près, la BDC d'un ensemble de matrices est d'utiliser une méthode de DC. Nous présenterons ici deux algorithmes différents : le premier bloc diagonalise ces matrices en utilisant la distance de Kullback-Liebler (KL) entre deux matrices définies positives. Le second est basé sur la méthode de Jacobi, ce qui permet une BDC à l'aide d'une transformée unitaire commune.

4.4.1 BDC via DC de matrices positives

Dans cette section, nous définirons un critère de BDC inspiré du critère de DC de D.Pham présenté dans [54]. Ce dernier est basé sur une méthode de maximum de vraisemblance qui utilise la divergence de Kullback-Leibler entre deux matrices définies positives.

Dans le cas d'une DC, la divergence KL permet d'obtenir le critère suivant [54] :

$$\sum_{k=1}^K n_k [\log \det(\text{diag}(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)) - \log \det(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)] \quad (4.20)$$

Où \mathbf{B} est une matrice "diagonalisante", qui n'est pas nécessairement unitaire, n_k sont des coefficients positifs (poids) et \mathbf{M}_k , $k = 1 \dots K$ sont des matrices définies positives.

Dans le cas de la BDC, cette mesure peut être adaptée de la manière suivante :

$$\sum_{k=1}^K n_k [\log \det(\text{bdiag}(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)) - \log \det(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)] \quad (4.21)$$

Ce critère a été utilisé dans notre travail [36] ainsi que dans [47] pour la séparation aveugle de mélanges convolutifs avec les poids $n_1 = \dots = n_K = 1$. Cependant, à notre connaissance, aucune preuve n'a été fournie pour justifier la validité du critère (4.21) comme un critère qui permet une BDC. Nous donnerons une preuve à travers le lemme suivant :

Lemme 2 *Le critère (4.21) est une fonction à valeurs positives égale à zéro si et seulement si $\mathbf{M}_k = \text{bdiag}(\mathbf{M}_k)$ pour tout $k = 1 \dots K$.*

Démonstration : Il est établi que pour une matrice définie positive \mathbf{M} (voir [54]), nous avons :

$$\det(\mathbf{M}) \leq \det(\text{diag}(\mathbf{M})) \quad (4.22)$$

Avec une égalité si et seulement si \mathbf{M} est diagonale, $\mathbf{M} = \text{diag}(\mathbf{M})$. De plus, pour toute matrice définie positive \mathbf{M} , les bloc diagonaux \mathbf{M}_i , $i = 1 \dots r$, et, en particulier, $\text{bdiag}(\mathbf{M})$ sont définis positifs.

Ecrivons la décomposition en valeurs et vecteurs propres de \mathbf{M}_i i.e. $\mathbf{M}_i = \mathbf{U}_i \Lambda_i \mathbf{U}_i^H$. Où \mathbf{U}_i sont unitaires et Λ_i sont des matrices diagonales positives et soit $\mathbf{U} \stackrel{\text{def}}{=} \text{diag}(\mathbf{U}_1 \dots \mathbf{U}_r)$. Nous aurons donc :

$$\det(\text{bdiag}(\mathbf{M})) = \prod_{i=1}^r \det(\mathbf{M}_i) = \prod_{i=1}^r \det(\Lambda_i)$$

D'un autre côté :

$$\det(\mathbf{M}) = \det(\mathbf{U}\mathbf{M}\mathbf{U}^H) \leq \det(\text{diag}(\mathbf{U}\mathbf{M}\mathbf{U}^H)) = \prod_{i=1}^r \det(\Lambda_i) = \det(\text{bdiag}(\mathbf{M}))$$

L'égalité est vérifiée si et seulement si $\mathbf{U}\mathbf{M}\mathbf{U}^H = \text{diag}(\mathbf{U}\mathbf{M}\mathbf{U}^H)$. A la fin,

$$\mathbf{M} = \mathbf{U}^H \text{diag}(\mathbf{U}\mathbf{M}\mathbf{U}^H) \mathbf{U} = \text{bdiag}(\mathbf{M}). \quad \square$$

Le critère (4.21) est, donc, une mesure de la déviation globale des matrices par rapport à une structure bloc diagonale.

Dans [47], des algorithmes basés sur le calcul du gradient ont été proposés pour minimiser le critère (4.21). Cependant, mis à part leur convergence lente, nous avons remarqué que pour une initialisation aléatoire, ces algorithmes convergent le plus souvent vers un minimum local et donc, ne donnent pas de solutions satisfaisantes.

Dans notre travail [36], un algorithme de DC de Pham a été utilisé pour minimiser le critère (4.21).

En fait, cet algorithme [54] consiste à faire des transformations successives, à chaque fois sur une paire de lignes, \mathbf{b}_p et \mathbf{b}_q , de la matrice \mathbf{B} suivant :

$$\begin{bmatrix} \mathbf{b}_p \\ \mathbf{b}_q \end{bmatrix} \leftarrow \mathbf{T}_{pq} \begin{bmatrix} \mathbf{b}_p \\ \mathbf{b}_q \end{bmatrix} \quad (4.23)$$

Où \mathbf{T}_{pq} est une matrice $[2 \times 2]$ non-singulière, choisie de manière à ce que le critère de DC est diminué suffisamment. Une fois ceci fait, cette procédure est répétée avec une autre paire de lignes jusqu'à terminer les $N(N - 1)/2$ paires de lignes. Le traitement de toutes les paires de lignes est appelé *un balayage*. L'algorithme est une suite de balayages jusqu'à la convergence. La diminution du critère de DC est donnée dans [54] et est égale à :

$$\sum_{k=1}^K n_k \{ 2 \log |\det \mathbf{T}_{pq}| - \log \det \left(\text{diag}(\mathbf{T}_{pq} \begin{bmatrix} (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp} & (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pq} \\ (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qp} & (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qq} \end{bmatrix} \mathbf{T}_{pq}^H) \right) + \log ((\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp} \times (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qq}) \} \quad (4.24)$$

Les indices $_{pq}$ indiquent le $(p, q)^{ieme}$ élément de la matrice considérée.

En adoptant la même stratégie, on a trouvé que la diminution du critère de la BDC est nulle si p et q balayent le même bloc. Dans le cas contraire, elle correspond à :

$$\sum_{k=1}^K n_k \{ 2 \log |\det \mathbf{T}'_{pq}| - \log \det \left(\text{bdiag}(\mathbf{T}'_{pq} \begin{bmatrix} [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{pp} & [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{pq} \\ [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{qp} & [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{qq} \end{bmatrix} \mathbf{T}'_{pq}{}^H) \right) + \log \left(\det [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{pp} \times \det [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{qq} \right) \} \quad (4.25)$$

\mathbf{T}'_{pq} est une matrice qui, si elle est multipliée à droite par la matrice bloc dans (4.25), permet la même transformation des lignes que dans (4.23).

Au lieu de maximiser la diminution dans (4.24), Pham propose de maximiser une borne inférieure de cette diminution, ce qui permet une implementation plus simple.

Maintenant, on peut remarquer que (4.24) est déjà une borne inférieure de (4.25). Ceci justifie l'utilisation de l'algorithme de Pham⁵ pour réaliser une BDC. Comme cela est montré dans [54], à chaque itération, \mathbf{T}_{pq} est calculé comme suit :

$$\mathbf{T}_{pq} = \mathbf{I} - \frac{2}{1 + \sqrt{1 - 4h_{pq}h_{qp}}} \begin{bmatrix} 0 & h_{pq} \\ h_{qp} & 0 \end{bmatrix}. \quad (4.26)$$

Avec les définitions suivantes :

$$\begin{bmatrix} h_{pq} \\ h_{qp} \end{bmatrix} = \begin{bmatrix} w_{pq} & 1 \\ 1 & w_{qp} \end{bmatrix}^{-1} \begin{bmatrix} g_{pq} \\ g_{qp} \end{bmatrix}. \quad (4.27)$$

⁵L'algorithme de Pham est légèrement modifié de manière à considérer, uniquement, les paires d'indices p et q qui balayent deux blocs différents.

et

$$g_{pq} = \sum_{k=1}^K \frac{n_k}{N} \frac{(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pq}}{(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp}}, \quad w_{pq} = \sum_{k=1}^K \frac{n_k}{N} \frac{(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qq}}{(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp}}.$$

Similairement, on a g_{qp} et w_{qp} .

4.4.2 BDC via DC sous une transformation unitaire commune

Dans [1, 51], un algorithme basé sur la méthode de Jacobi a été introduit pour réaliser une DC ⁶ d'un ensemble de matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ sous une transformée unitaire commune \mathbf{E} . La matrice \mathbf{E} est estimée par un produit de rotations de Givens, mais contrairement à l'algorithme de la BDC, l'optimisation des angles de rotation implique un critère quadratique (au lieu d'un critère linéaire quadratique) de la forme $\mathbf{v}^T \tilde{\mathbf{G}} \mathbf{v}$ où $\tilde{\mathbf{G}}$ est une matrice à valeurs réelles de dimension $[3 \times 3]$ (ou $[2 \times 2]$ dans le cas réel) et \mathbf{v} est un vecteur défini par l'équation (4.12). Par conséquent, \mathbf{v} correspond au vecteur propre de norme unitaire associé à la plus petite valeur propre de $\tilde{\mathbf{G}}$.

Nous proposerons, ici, d'utiliser l'algorithme de DC, présenté dans [51] pour effectuer une BDC. Nous pouvons affirmer que l'on peut arriver à une BDC (à une matrice de permutation près) en effectuant une DC d'un ensemble de matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$. Plus précisément, nous présentons le lemme suivant :

Lemme 3 *Soit $\mathbf{M}_1, \dots, \mathbf{M}_K$ un ensemble de K matrices satisfaisant l'équation (4.1). Optimiser le critère de la DC pour $\mathbf{M}_1, \dots, \mathbf{M}_K$ en utilisant l'algorithme dans [51], nous procure un ensemble de matrices bloc diagonales à une matrice de permutation \mathbf{P} près.*

Démonstration : Ce résultat provient de l'observation suivante : Il est montré que optimiser le critère de DC par des rotations successives de Givens revient à résoudre le même problème pour des matrices $[2 \times 2]$, autrement dit, l'optimisation de $\Theta_{(qp)}$ est équivalente à la DC d'un ensemble de matrices $[2 \times 2]$.

$$\mathbf{M}_k^{(qp)} = \begin{bmatrix} \mathbf{M}_k(p, p) & \mathbf{M}_k(p, q) \\ \mathbf{M}_k(q, p) & \mathbf{M}_k(q, q) \end{bmatrix}, \quad k = 1, \dots, K. \quad (4.28)$$

⁶la DC peut être considérée comme une BDC particulière lorsque $m_i = 1, \forall i$.

Cet algorithme itératif s'arrête lorsque à un certain balayage (ou itération), nous aurons ⁷ :

$$\Theta_{(qp)} = \mathbf{I} \quad \forall 1 \leq p < q \leq N \quad (4.29)$$

Ceci signifie que le critère de DC (i.e. $\sum_k \|\mathbf{E}^H \mathbf{M}_k \mathbf{E} - \text{diag}(\mathbf{E}^H \mathbf{M}_k \mathbf{E})\|^2$) ne peut plus être diminué.

A présent, si les matrices $[2 \times 2]$ dans l'équation (4.28) ne sont pas diagonales pour des valeurs de p et q correspondantes à des éléments qui ne se trouvent pas dans des blocs diagonaux (i.e. $(p, q) \in \mathcal{I}$), alors, il existe une matrice unitaire et, donc, une rotation de Givens qui les transformera en matrices diagonales (puisque $\mathbf{M}_1, \dots, \mathbf{M}_K$ satisfont (4.1)). Par conséquent, le critère de DC peut être encore diminué et la condition d'arrêt de l'algorithme (4.29) n'est pas vérifiée dans ce cas. \square

En pratique, effectuer une BDC à une matrice de permutation inconnue prés n'est pas suffisant car les seules permutations qui soient acceptables sont celles qui préservent la structure bloc diagonale (permuter les blocs diagonaux ou permuter les éléments à l'intérieur d'un même bloc sont les seules permutations admissibles). Pour éliminer cette matrice de permutation indésirable, nous proposons les deux solutions suivantes :

- La première solution consiste à décomposer la permutation \mathbf{P} en un produit de permutations élémentaires ⁸ $\mathbf{P}_{(qp)}$. La permutation n'est retenue ou considérée que si elle augmente le critère de la BDC, c'est à dire :

$$\sum_{k=1}^K \|\text{bdiag}(\mathbf{P}_{(qp)}^T \mathbf{M}_k \mathbf{P}_{(qp)})\|^2 > \sum_{k=1}^K \|\text{bdiag}(\mathbf{M}_k)\|^2 \quad (4.30)$$

$\mathbf{P}_{(qp)}$ est construite de manière à ce que tous ses éléments diagonaux sont égaux à "1" sauf pour les deux éléments sur les lignes (et colonnes) p and q qui seront nuls. De façon similaire, tous les éléments non-diagonaux de $\mathbf{P}_{(qp)}$ sont nuls sauf pour les $(p, q)^{ieme}$ et $(q, p)^{ieme}$ éléments qui sont égaux à "1".

Comme cela a été détaillé dans 4.3.2, l'augmentation du critère ci-dessus implique la maximisation de la forme quadratique suivante : $\max_{\|\mathbf{v}\|=1} (\mathbf{v}^T \mathbf{G} \mathbf{v} + \mathbf{g}^T \mathbf{v})$.

Le terme à gauche de (4.30) correspond au cas où $\mathbf{v} = [-1, 0, 0]^T$, alors que le

⁷ $\Theta_{(qp)} = \mathbf{I}$ est équivalent à $s = 0$. En pratique, nous utiliserons un seuil très faible ϵ pour tester si pour tout $1 \leq p < q \leq n$, $|s| < \epsilon$.

⁸ $\mathbf{P}_{(qp)}$ est définie de sorte que pour un vecteur \mathbf{x} donné, $\tilde{\mathbf{x}} = \mathbf{P}_{(qp)} \mathbf{x}$ si et seulement si $\tilde{x}(k) = x(k)$, pour $k \notin \{p, q\}$, $\tilde{x}(p) = x(q)$ et $\tilde{x}(q) = x(p)$.

terme à droite correspond à $\mathbf{v} = [1, 0, 0]^T$. Par conséquent, l'inégalité (4.30) se réduit à :

$$-\mathbf{g}(1) > \mathbf{g}(1) \quad (4.31)$$

Autrement dit $\text{sign}(\mathbf{g}(1)) < 0$, $\mathbf{g}(1)$ étant le premier élément du vecteur \mathbf{g} . Finalement, nous obtenons :

$$\mathbf{P} = \prod_{\text{nb de balayages}} \prod_{1 \leq p < q \leq N} \tilde{\mathbf{P}}_{(qp)}$$

$\tilde{\mathbf{P}}_{(qp)}$ est soit la matrice identité, soit la matrice de permutation $\mathbf{P}_{(qp)}$ en fonction de la décision binaire dans (4.31).

Le processus itératif est arrêté lorsque toutes les matrices $\tilde{\mathbf{P}}_{(qp)}$ sont égales à la matrice identité. Dans nos simulations, nous avons remarqué que un ou, au plus, deux balayages sont suffisants pour arriver à la permutation voulue.

- La deuxième approche consiste à déterminer la permutation à partir des positions des éléments non-nuls de \mathbf{M}_k . D'une manière itérative et en commençant à la première ligne, l'algorithme détecte les positions des éléments non-nuls et groupe leur vecteur colonne et leur vecteur ligne correspondants de manière à former des bloc diagonaux.

Ce qui donne une solution relativement simple lorsque l'ensemble des matrices satisfait, exactement, à l'équation (4.1) (voir le code pseudo-MATLAB en annexe C). Cependant, lorsque les matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ sont "approximativement" bloc diagonalisables, cette approche fait appel à un seuillage pour décider si un élément donné d'une matrice est considéré comme nul ou pas. Dans ce cas, la solution précédente est préférable car elle ne demande aucun seuillage.

4.5 Evaluation des performances

Cette section est destinée à l'analyse des performances et à la validation des différents algorithmes proposés.

Le taux de convergence et l'erreur estimée de tous les algorithmes sont étudiés et comparés à travers de simulations intensives.

Nous comparons, dans cette section, les performances des algorithmes de la BDC sui-

vants : le LSBDD (Least Squares Block Diagonalization) pour l'algorithme exact utilisant les moindres carrés présenté en 4.3.2, le ALSBD (Approximate Least Squares Block Diagonalization) pour le LSBDD approximé présenté en 4.3.3, le NJBD (New Joint Block Diagonalization) pour le nouveau algorithme utilisant la technique de Jacobi de DC et qui a été étudié en 4.4.2 et enfin le PJBD (Pham Joint Block Diagonalization) pour l'algorithme inspiré de Pham et détaillé en 4.4.1.

Dans cette étude comparative, trois critères sont utilisés pour évaluer l'erreur estimée (i.e, la qualité de la BDC), le taux de convergence moyen et la sensibilité à l'initialisation aléatoire (i.e, la convergence aux minimums locaux).

La qualité de la BDC est évaluée par :

$$\rho = \sum_{k=1}^K \frac{\| \mathbf{M}_k - \text{bdiag}(\mathbf{M}_k) \|^2}{\| \text{bdiag}(\mathbf{M}_k) \|^2} \quad (4.32)$$

Pour mesurer le taux de convergence moyen, on utilise le critère suivant :

$$\chi = \frac{1}{N_r} \sum_{r=1}^{N_r} I_r$$

Où N_r est le nombre d'essais Monte-Carlo réussis ('réussis' veut dire que l'algorithme a convergé vers un minimum global comme cela sera expliqué ci-dessous) et I_r est le nombre d'itérations nécessaires pour obtenir une convergence au r^{ieme} essai.

Pour affirmer si un algorithme a convergé, on utilise un seuil, ($\epsilon_1 = 10^{-7}$) et on décidera qu'il y a convergence si pendant un balayage $|\theta_{qp}| \leq \epsilon_1$ pour tous les (q, p) pour les algorithmes se basant sur Jacobi (LSBD, ALSBD et NJBD) et si $|h_{pq}| + |h_{qp}| \leq \epsilon_1$ pour l'algorithme PJBD.

Et enfin, pour évaluer la sensibilité des algorithmes au problème des minimums locaux, on mesure le pourcentage du taux de convergences réussies :

$$\eta = \frac{\text{Nombre de convergences réussies}}{\text{Nombre total d'essais}} \times 100 \quad (4.33)$$

Pour décider, dans le cas sans bruit, si un algorithme a effectué une BDC, on utilise, encore, un autre seuil ϵ_2 ($\epsilon_2 = 10^{-2}$) pour le critère de BDC normalisé, c'est à dire, nous avons une convergence réussie si $\rho \leq \epsilon_2$.

Toutes les statistiques sont calculées après 500 essais Monte-Carlo. A chaque essai, les matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ sont générées d'une manière aléatoire suivant le modèle dans

(4.1), à l'exception de l'algorithme PJBD où les matrices sont générées d'une manière aléatoire de telle manière que la matrice \mathbf{E} est aléatoire et pas spécialement unitaire mais les matrices \mathbf{D}_{ki} ($k = 1, \dots, K$ et $i = 1, \dots, r$) sont aléatoires et définies positives.

Dans la première expérience, nous avons pris $K = 3$ matrices carrées de dimension $N = 4$ ayant des blocs de dimensions $M_1 = M_2 = 2$. La figure 4.1 montre le taux de convergence des quatre méthodes de BDC dans le cas sans bruit. Les graphes représentent l'évolution, en moyenne, du critère normalisé de la BDC (4.32) par rapport au nombre d'itérations (ou balayages). Nous pouvons constater que la méthode NJBD est, de loin, la plus rapide. Elle converge après, généralement, une dizaine d'itérations. Les autres méthodes sont, relativement, lentes mais convergent après un grand nombre d'itérations. Notons que, la méthode ALSBD est la moins rapide. Néanmoins, cet algorithme est le moins coûteux en nombre d'opérations par itération.

La figure 4.2 représente la convergence des différents algorithmes dans le cas bruité. Le critère normalisé de la BDC (4.32) est tracé en fonction du rapport signal-sur-bruit (SNR). Le bruit est intégré de la manière suivante : les matrices exactes \mathbf{M}_i , $i = 1, \dots, K$ sont perturbées par des matrices aléatoires dont les éléments sont gaussiens i.i.d. et de puissance :

$$\sigma^2 = \frac{\sum_{k=1}^K \|\mathbf{M}_k\|^2}{K N^2} \times 10^{-\frac{SNR}{10}}$$

Le SNR est exprimé en dB. Pour un faible SNR, la meilleure performance est obtenue par la méthode PJBD ; alors que, pour des SNR relativement moyens ou élevés, la méthode NJBD est la meilleure. Cependant, à l'exception de NJBD, ces résultats sont biaisés par le fait que le critère (4.32) est calculé, en moyenne, sur tous les essais Monte-Carlo, y compris ceux pour lesquels les algorithmes de BDC convergent vers des minimums locaux. Nous verrons, par la suite, que les méthodes LSB, ALSBD et PJBD sont très sensibles à l'initialisation et, donc, sont sujettes au problème des minimums locaux.

Dans la deuxième expérience, on considère, dans le cas non bruité, $K = 3$ matrices carrées avec des blocs de dimensions $M_i = 2$ pour tout $i = 1, \dots, r$ et un nombre de blocs r variant dans l'intervalle $[1 \dots 5]$.

La figure 4.3 illustre le problème de la convergence indésirable (ou l'absence de convergence) dans le cas d'une initialisation aléatoire des algorithmes. Comme on peut le voir, pour les algorithmes LSB, ALSBD et PJBD, le taux de convergences réussies tend rapi-

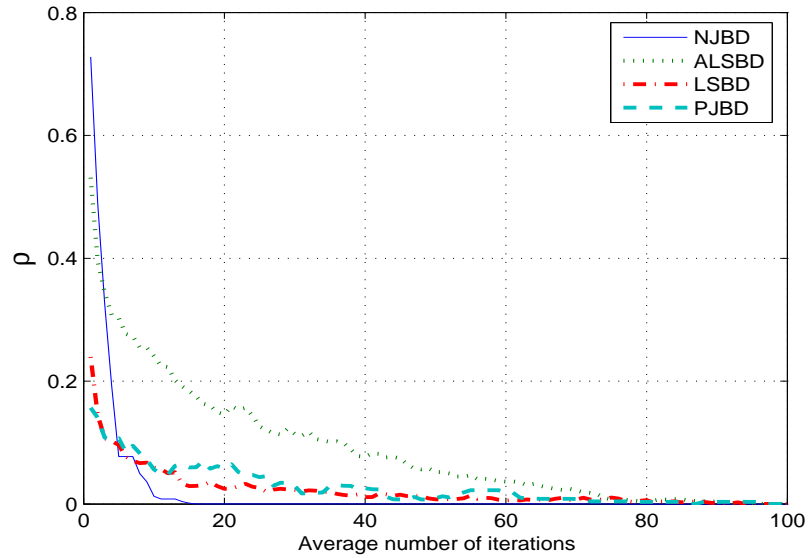


FIG. 4.1 – Le critère de BDC par rapport au nombre d’itérations dans le cas non bruité : $K = 3$, $N = 4$ et $r = 2$.

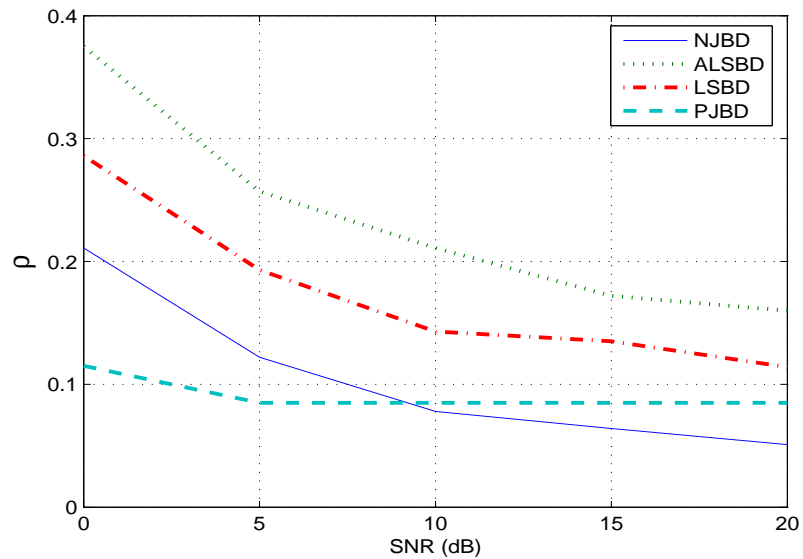


FIG. 4.2 – Le critère de BDC par rapport au SNR : $K = 3$, $N = 4$ et $r = 2$.

dement vers zéro lorsque le nombre de blocs dans une matrice augmente. NJBD est le seul algorithme qui converge, presque, dans tous les cas quelque soit la taille de la matrice.

Sur la figure 4.4, on trace le nombre moyen d'itérations par rapport au nombre de blocs diagonaux élémentaires. Le nombre d'itérations nécessaires pour converger augmente, d'une manière significative, pour les algorithmes LSB et ALSB ; alors qu'il reste stable (presque constant) pour les algorithmes NJBD et PJBD. Cependant, pour PJBD et ALSB, le nombre d'essais réussis diminue rapidement et s'annule lorsque le nombre de blocs est égal à $r = 5$. Le meilleur taux de convergence est obtenu par NJBD qui converge, en moyenne, après une dizaine d'itérations.

Un résultat similaire est obtenu dans notre troisième expérience qui consiste à bloc diago-

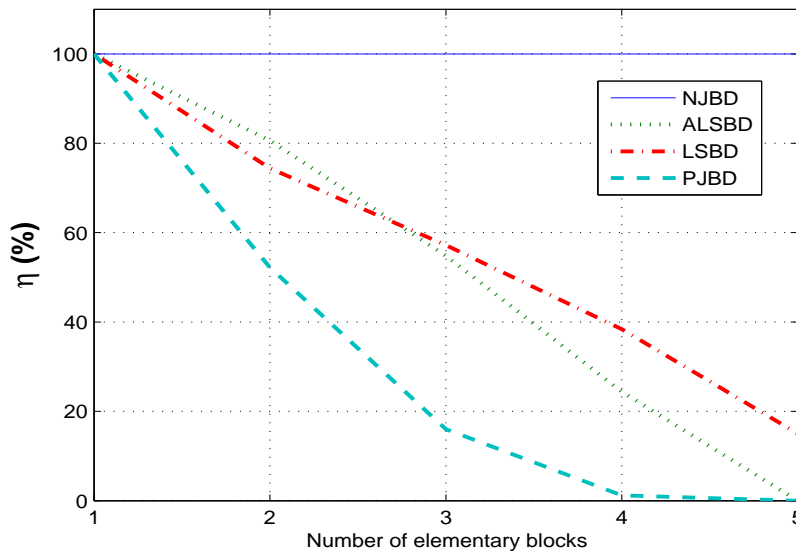


FIG. 4.3 – Pourcentage du taux de BDC réussies par rapport au nombre de blocs : $K = 3$ et $r = 1, \dots, 5$.

naliser conjointement $K = 3$ matrices carrées dont la dimension globale reste constante $N = 12$ mais la dimension des blocs diagonaux varie dans la bande $\{2,3,4,6\}$. La figure 4.5 montre le taux de convergences réussies pour les algorithmes NJBD et LSB par rapport à la taille des blocs. Les algorithmes PJBD et ALSBD ne convergent pas vers la solution désirée et, pour cette raison, ne sont pas représentés sur le graphe. Pour ces algorithmes, lorsque la dimension des matrices à bloc diagonaliser augmente, une initialisation aléatoire aboutit, presque toujours, à la convergence vers un minimum local. D'un

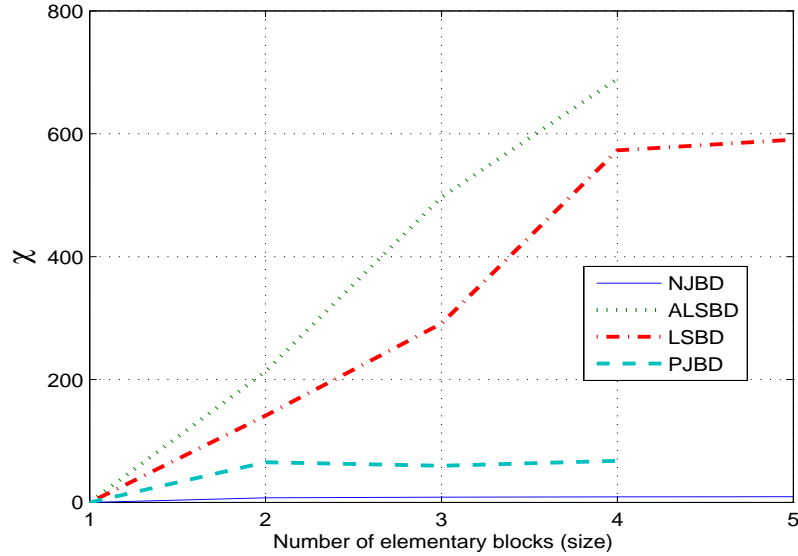


FIG. 4.4 – Taux de convergence moyen par rapport au nombre de blocs : $K = 3$ et $r = 1, \dots, 5$.

autre côté, ce graphe confirme le bon comportement de l’algorithme NJBD qui a un taux de convergences réussies proche de 100%.

En se basant sur tous ces résultats, nous suggérons d’utiliser l’algorithme NJBD pour obtenir une solution initiale au problème de la BDC. Cette solution sera utilisée pour initialiser les trois autres algorithmes, ce qui permet d’affiner la BDC des matrices à bloc diagonaliser. La figure 4.6 compare les performances, en terme du critère de la BDC dans (4.32), des trois algorithmes ALSBD, LSB et PJBD. Comparativement avec les résultats obtenus dans la figure 4.2, nous remarquons que le critère a été diminué et divisé par un facteur de 5 à 10, approximativement. Pour un SNR faible, le meilleur résultat est obtenu par l’algorithme PJBD.

Les figures 4.7 et 4.8 représentent une étude du comportement des méthodes proposées par rapport à la variation du nombre de matrices à bloc diagonaliser conjointement K . Nous avons considéré le problème de la BDC dans le cas non bruité avec la taille des matrices égale à $N = 4$ et la taille des blocs égale à $M_1 = M_2 = 2$. Nous pouvons voir que les différentes méthodes ne sont pas affectées lorsque K varie, à l’exception de PJBD. Comme le montre la figure 4.7, PJBD est moins performante lorsque le nombre de matrices à bloc diagonaliser conjointement est, relativement, petit. Dans ce cas, PJBD

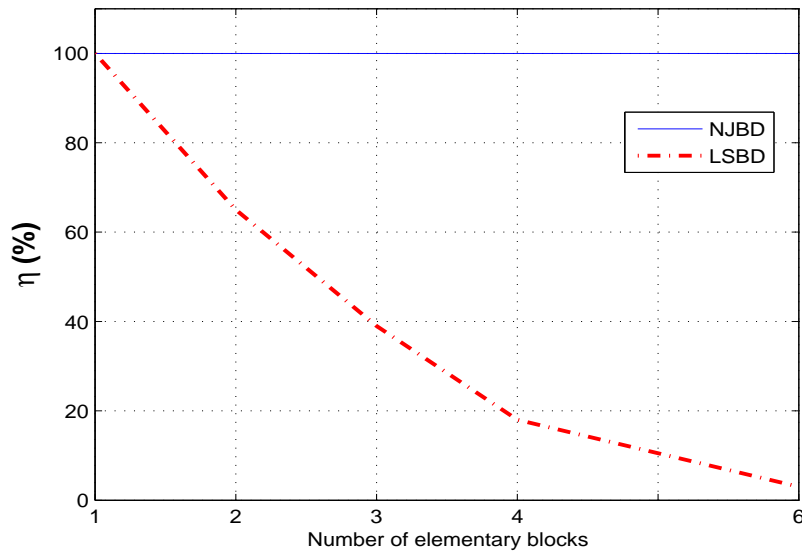


FIG. 4.5 – Pourcentage du taux de BDC réussies par rapport au nombre de blocs : $K = 3$ et $N = 12$.

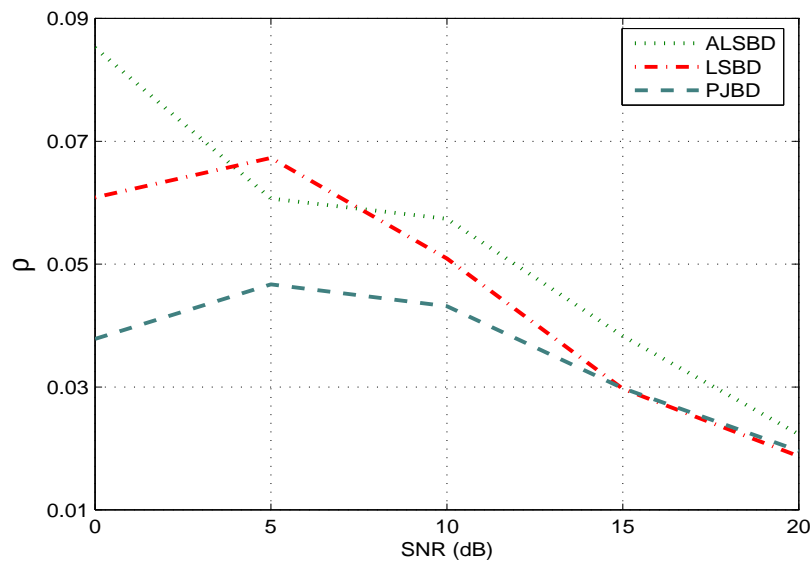


FIG. 4.6 – Le critère de BDC par rapport au SNR avec, au préalable, une initialisation NJBD : $K = 3$, $N = 4$ et $r = 2$.

est plus sensible au problème des minimums locaux.

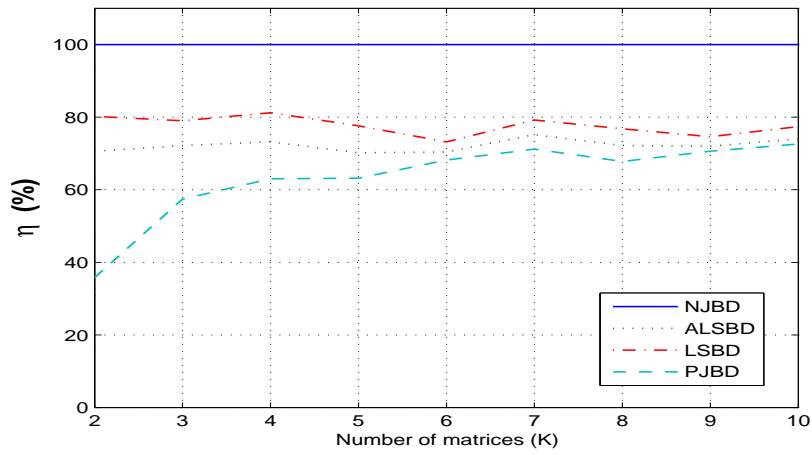


FIG. 4.7 – Pourcentage du taux de convergences réussies par rapport au nombre de matrices K : $N = 4$ et $r = 2$.

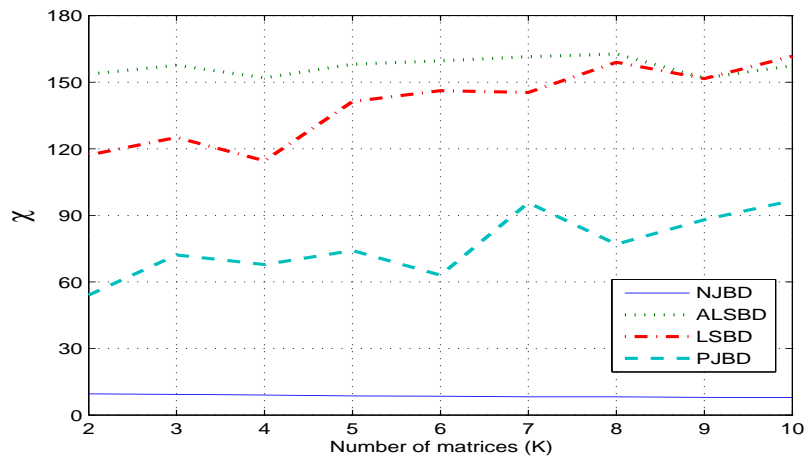


FIG. 4.8 – Taux de convergence moyen par rapport au nombre matrices K : $N = 4$ et $r = 2$

Chapitre 5

Déconvolution complète du système

MIMO

Dans le chapitre 3, nous avons présenté deux techniques de séparation à partir de mélanges convolutifs. Ces techniques sont basées sur la BDC de matrices d'auto-corrélations et de matrices de covariances, respectivement. Ces techniques se sont révélées très efficaces pour les applications pouvant s'accommoder d'un résultat de la séparation à un filtre prés. Néanmoins, pour d'autres applications, ce filtre représente un obstacle à la reconstruction des signaux sources. Dans ce chapitre, nous présenterons une méthode de déconvolution *complète* de notre système MIMO. Autrement dit, la méthode de séparation, décrite dans le chapitre 3, sera suivie de plusieurs égalisations SIMO (une pour chaque source considérée). Ce qui permet de retrouver le canal manquant pour chaque source.

5.1 Algorithme proposé

Reprenons le modèle, non bruité, de séparation de signaux stationnaires décrit en 3.1.1. Rappelons, brièvement, les différentes opérations effectuées sur les données reçues : après une étape de blanchiment qui permet de rendre le mélange unitaire, on obtient les matrices d'auto-corrélation des données blanchies suivantes (le détail est donné en (3.5)) :

$$\forall k \neq 0 \quad \underline{\mathbf{R}}_{\mathbf{xx}}(k) = \mathbf{U}\mathbf{D}(k)\mathbf{U}^H$$

Puisque la matrice \mathbf{U} est unitaire et la matrice $\mathbf{D}(k)$ est bloc diagonale, cela signifie que \mathbf{U} peut être extraite en bloc diagonalisant conjointement un ensemble de K matrices d'auto-

corrélations $\mathbf{R}_{\mathbf{x}\mathbf{x}}(k); k = 1, \dots, K$. Une fois la bloc diagonalisation réalisée, les signaux sources sont reconstitués, à un filtre près, par :

$$\hat{\mathbf{s}}(n) = \mathbf{U}^H \mathbf{W} \mathbf{x}(n) \quad (5.1)$$

En utilisant les relations (2.29) et (3.3), ces signaux vérifient,

$$\hat{\mathbf{s}}(n) = \mathbf{R}_{\mathbf{ss}}^{-\frac{1}{2}}(0) \mathbf{s}(n) = \begin{bmatrix} \mathbf{D}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{D}_M \end{bmatrix} \begin{bmatrix} \mathbf{s}_1(n) \\ \vdots \\ \mathbf{s}_M(n) \end{bmatrix} \quad (5.2)$$

Les matrices \mathbf{D}_i sont de taille $[(L + L') \times (L + L')]$.

Décomposons, à présent, les sources issues de l'équation (5.1) en M blocs distincts,

$$\hat{\mathbf{s}}(n) = \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_M \end{bmatrix} \quad (5.3)$$

A ce niveau, nous obtenons un ensemble de M matrices \mathbf{S}_i contenant chacune les $(L + L')$ signaux représentant des versions filtrées de chaque source. Chacun de ces M ensembles peut être décrit en utilisant un modèle SIMO.

5.1.1 Identification SIMO

Pour identifier le filtre manquant pour chacune des M sources, nous pouvons utiliser plusieurs méthodes d'égalisation SIMO au second ordre, existantes dans la littérature [28, 57]. Seulement, la plupart de ces méthodes supposent connaître parfaitement l'ordre du canal i.e. son degré le plus élevé. Dans notre travail, et compte tenu du caractère *aveugle* de la séparation, cette information n'est pas disponible. En effet, nous ne disposons que des données issus des différents capteurs. Nous utiliserons une technique basée sur la méthode sous-espace canal SIMO développée par A. Gorokhov *et al.* dans [58]. Comme cela a été présenté pour la méthode sous-espace MIMO en 2.3.2, les sous-espaces signal et bruit sont orthogonaux. On obtient donc une relation dans le cas SIMO équivalente à celle du cas MIMO donnée en (2.14),

$$\mathbf{\Pi}_W \mathcal{T}_W(\mathbf{h}) = \mathbf{0} \quad (5.4)$$

où $\mathbf{\Pi}_W$ est la matrice de projection orthogonale sur l'espace $\text{Range}(\mathcal{T}_W(\mathbf{h}))^\perp$, qu'on appelle, le sous-espace bruit. \mathbf{h} représente le vecteur du canal SIMO qu'on désire identifier et W , une fenêtre d'observation des données. Pour identifier le canal, l'algorithme minimise le critère suivant,

$$\hat{\mathbf{h}} = \arg \min \{ \|\mathbf{\Pi}_W \mathcal{T}_W(\mathbf{f})\|^2 : \|\mathbf{f}\| = 1 \} \quad (5.5)$$

$\hat{\mathbf{h}}$ étant le vecteur estimé du canal. Ce critère est quadratique et peut être, donc, réécrit de la manière suivante :

$$\|\mathbf{\Pi}_W \mathcal{T}_W(\mathbf{f})\|^2 = \mathbf{f}^H \hat{\mathbf{Q}}_{L_i} \mathbf{f}$$

Le détail de la construction de la matrice $\hat{\mathbf{Q}}_{L_i}$ est donné en [28]. Pour ne pas alourdir cette section, nous retiendrons que cette matrice a une structure bloc-Toeplitz. L_i est un paramètre, que l'on fera varier car inconnu, et qui représente l'ordre du canal.

Dans un contexte aveugle, le canal \mathbf{h} est reconstruit à une constante scalaire non nulle α près, i.e. $\hat{\mathbf{h}} = \alpha \mathbf{h}$. On peut, donc, utiliser plusieurs contraintes linéaires différentes pour la minimisation du critère (5.5) :

$$\hat{\mathbf{h}}^{(k)} = \arg \min \{ \mathbf{f}^H \hat{\mathbf{Q}}_{L_i} \mathbf{f} : \mathbf{b}_k^H \mathbf{f} = \alpha_k \} \quad 1 \leq k \leq m \quad (5.6)$$

où \mathbf{b}_k est un vecteur colonne et α_k un scalaire non nul. En posant $\gamma_k = \alpha_k / \mathbf{b}_k^H \hat{\mathbf{Q}}_{L_i}^{-1} \mathbf{b}_k$, les m différentes solutions possibles s'écriront :

$$\hat{\mathbf{h}}^{(k)} = \gamma_k \hat{\mathbf{Q}}_{L_i}^{-1} \mathbf{b}_k \quad 1 \leq k \leq m$$

En supposant que l'ordre exact du canal est de $(L - 1)$, le problème qui se pose est le suivant : comment choisir les contraintes linéaires de manière à ce que $\mathbf{b}_k^H \mathbf{h} \neq 0$ soit assuré ? Une solution possible est de partitionner la matrice de projection dans le sous-espace bruit $\mathbf{\Pi}_W$ en W blocs et prendre le premier et le dernier bloc de l'équation $\mathbf{\Pi}_W \mathcal{T}_W(\mathbf{h}) = \mathbf{0}$. Cette équation peut être développée comme suit [28],

$$\begin{bmatrix} \mathbf{\Pi}_W(0) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{\Pi}_W(W-1) & \cdots & \mathbf{\Pi}_W(0) \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{\Pi}_W(W-1) \end{bmatrix} \begin{bmatrix} \mathbf{h}(0) \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{h}(L-1) \end{bmatrix} = \mathbf{0}$$

Les deux blocs considérés correspondent à,

$$\begin{aligned}\mathbf{\Pi}_W(0) \mathbf{h}(0) &= \mathbf{0} \\ \mathbf{\Pi}_W(W-1) \mathbf{h}(L-1) &= \mathbf{0}\end{aligned}$$

Ce qui correspond à minimiser les formes quadratiques suivantes :

$$\begin{aligned}\min_{\|\mathbf{h}(0)\|=1} (\mathbf{h}(0)^H \mathbf{\Pi}_W(0)^H \mathbf{\Pi}_W(0) \mathbf{h}(0)) &= \mathbf{0} \\ \min_{\|\mathbf{h}(L-1)\|=1} (\mathbf{h}(L-1)^H \mathbf{\Pi}_W(W-1)^H \mathbf{\Pi}_W(W-1) \mathbf{h}(L-1)) &= \mathbf{0}\end{aligned}$$

Pour cela, il suffit de chercher les vecteurs propres \mathbf{g}_0 et \mathbf{g}_{W-1} associés, respectivement, à la plus petite valeur de $(\mathbf{\Pi}_W(0)^H \mathbf{\Pi}_W(0))$ et de $(\mathbf{\Pi}_W(W-1)^H \mathbf{\Pi}_W(W-1))$.

\mathbf{g}_0 et \mathbf{g}_{W-1} sont, respectivement, colinéaires avec $\mathbf{h}(0)$ et $\mathbf{h}(L-1)$. Ces deux vecteurs propres vérifient les relations suivantes :

$$\begin{aligned}\mathbf{g}_0^H \mathbf{h}(0) &\neq 0 \\ \mathbf{g}_{W-1}^H \mathbf{h}(L-1) &\neq 0\end{aligned}$$

Les contraintes linéaires seront, donc, données par :

$$\begin{aligned}\hat{\mathbf{b}}_0 &= [\hat{\mathbf{g}}_0^T, 0, \dots, 0]^T \\ \hat{\mathbf{b}}_{W-1} &= [0, \dots, 0, \hat{\mathbf{g}}_{W-1}^T]^T\end{aligned}$$

où $\hat{\mathbf{g}}_0$ et $\hat{\mathbf{g}}_{W-1}$ sont les estimées de \mathbf{g}_0 et \mathbf{g}_{W-1} , respectivement.

A. Gorokhov *et al.* dans [58] proposent une distance qui mesure la proximité de deux estimées du canal \mathbf{h} ,

$$\mathcal{D}_{L_i}(\hat{\mathbf{h}}^{(0)}, \hat{\mathbf{h}}^{(L-1)}) = \frac{|\hat{\mathbf{h}}^{(0)H} \hat{\mathbf{h}}^{(L-1)}|}{\|\hat{\mathbf{h}}^{(0)}\| \|\hat{\mathbf{h}}^{(L-1)}\|}$$

En fait, ce n'est que la corrélation normalisée entre les deux vecteurs $\hat{\mathbf{h}}^{(0)}$ et $\hat{\mathbf{h}}^{(L-1)}$ (le cosinus de l'angle entre ces deux vecteurs). Plus cette distance est grande, plus les deux vecteurs sont colinéaires et donc $\hat{\mathbf{h}}^{(0)}$ et $\hat{\mathbf{h}}^{(L-1)}$ représentent le même canal à une constante non nulle près. Idéalement, lorsque l'algorithme trouve la valeur exacte de l'ordre de notre canal, cette distance devient maximale (égale à 1).

A présent, nous pouvons appliquer cet l'algorithme à chaque ensemble de $(L + L')$ signaux représentant chacune de nos sources. Dans notre cas, l'ordre du canal à trouver est

$(L + L' - 1)$. Il se résume comme suit :

1. Fixer les valeurs minimum (L_{min}) et maximum (L_{max}) possibles de l'ordre du canal. La fenêtre d'observation doit être $W \geq L_{max}$.
2. Calculer la matrice de covariance des données et trouver une base du sous-espace bruit.
3. Calculer la matrice $\hat{\mathbf{Q}}_{L_{max}}$, et les vecteurs $\hat{\mathbf{g}}_0$ et $\hat{\mathbf{g}}_{W-1}$. Commencer par $L_i = L_{min}$, $\hat{\mathbf{b}}_0 = [\hat{\mathbf{g}}_0^T, \mathbf{0}_{1,(L+L')L_i}]^T$ et $\hat{\mathbf{b}}_{W-1} = [\mathbf{0}_{1,(L+L')L_i}, \hat{\mathbf{g}}_{W-1}^T]^T$.
4. Calculer $\hat{\mathbf{h}}^{(0)} = \hat{\mathbf{Q}}_{L_i}^{-1} \hat{\mathbf{b}}_0$ et $\hat{\mathbf{h}}^{(L+L'-1)} = \hat{\mathbf{Q}}_{L_i}^{-1} \hat{\mathbf{b}}_{W-1}$ où $\hat{\mathbf{Q}}_{L_i}$ est le bloc de taille $[(L + L')(L_i + 1) \times (L + L')(L_i + 1)]$ en haut, à gauche de $\hat{\mathbf{Q}}_{L_{max}}$.
5. Calculer la distance $\mathcal{D}_{L_i}(\hat{\mathbf{h}}^{(0)}, \hat{\mathbf{h}}^{(L+L'-1)})$.
6. Si $L_i = L_{max}$, aller à 7. Sinon, $L_i = L_i + 1$, et $\hat{\mathbf{b}}_0 = [\hat{\mathbf{b}}_0^T, \mathbf{0}_{1,(L+L')}]^T$, $\hat{\mathbf{b}}_{W-1} = [\mathbf{0}_{1,(L+L')}, \hat{\mathbf{b}}_{W-1}^T]^T$, et aller à 4.
7. Choisir l'ordre $\hat{L} = \arg \max_{L_i} \mathcal{D}_{L_i}$ et prendre le $\hat{\mathbf{h}}^{(0)}$ correspondant comme vecteur estimé du canal associé à la source considérée.

$\mathbf{0}_{1,K}$ est un vecteur formé de 0 et de taille $[1 \times K]$.

5.1.2 Egalisation par forçage à zéro

Pour égaliser le canal de chaque source $s_i(n)$ $i = 1, \dots, M$, on utilise la méthode du forçage à zéro [59]. On construit à partir du canal estimé $\hat{\mathbf{h}}^{(0)} = [\hat{\mathbf{h}}(0)^T \dots \hat{\mathbf{h}}(L+L'-1)^T]^T$ la matrice de Sylvestre suivante,

$$\hat{\mathbf{H}} = \begin{bmatrix} \hat{\mathbf{h}}(0) & \dots & \hat{\mathbf{h}}(L+L'-1) & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \vdots & \ddots & & & \vdots \\ \vdots & & \hat{\mathbf{h}}(0) & \dots & \hat{\mathbf{h}}(L+L'-1) & & \vdots \\ \vdots & & & \ddots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \hat{\mathbf{h}}(0) & \dots & \hat{\mathbf{h}}(L+L'-1) \end{bmatrix}$$

Ensuite, on calcule la matrice pseudo-inverse de $\hat{\mathbf{H}}$ qu'on appellera $\hat{\mathbf{G}} = \hat{\mathbf{H}}^\#$. Chaque ligne de $\hat{\mathbf{G}}$ est un égaliseur potentiel, on prendra celui qui correspond à une colonne de $\hat{\mathbf{H}}$

où tous les coefficients de $\hat{\mathbf{h}}^{(0)}$ sont représentés. Soit $\mathbf{g}^{(k)}$, cet égaliseur, en utilisant (5.3), la i^{eme} source estimée sera, alors :

$$\hat{\mathbf{s}}_i(n) = \mathbf{g}^{(k)} \times \mathbf{S}_i$$

5.2 Evaluation des performances

Dans cette section, nous présentons un exemple de simulation de notre algorithme (*separation-equalization*) pour montrer son efficacité à séparer des signaux de paroles. Notre algorithme est comparé à un algorithme qui réalise une égalisation suivie d'une séparation (*equalization-separation*). Pour ce dernier, l'égalisation est effectuée à l'aide d'une méthode de type sous-espace canal qui utilise la technique des moindres carrés [29]. Ce qui permet d'obtenir une estimation du canal à une matrice constante près. Cette opération est suivie d'un algorithme de forçage à zéro pour construire un égaliseur. En appliquant cet égaliseur à nos données, on obtient un mélange instantané des sources. Puis, finalement, l'algorithme SOBI [1] nous permet de séparer les signaux sources à un facteur d'échelle et à une permutation près.

Pour comparer ces deux algorithmes, l'erreur normalisée entre les signaux sources originaux et les signaux sources estimés est tracée en fonction du rapport signal-sur-bruit (SNR). Les paramètres de notre simulation sont les suivants :

- Le nombre de sources est $M = 2$, le nombre de capteurs est $N = 3$ et $L' = 5$.
- Les deux signaux de parole sont échantillonnés à 8kHz.
- Pour notre méthode (*separation-equalization*), la BDC a été réalisée à l'aide de l'algorithme NJBD.
- Toutes les statistiques obtenues sont évaluées après 100 réalisations.

Dans la première expérience, représentée sur la figure 5.1, nous avons pris le même ordre pour les canaux des deux sources originales, soit $L = 2$. Les deux algorithmes ont un comportement satisfaisant en fonction du SNR, surtout pour les SNR élevés. Cependant, les performances du deuxième algorithme (*equalization-separation*) sont, légèrement, meilleures que la méthode que nous proposons.

Dans la deuxième expérience, la figure 5.2 montre les mêmes courbes lorsque les ordres des canaux des deux sources sont différents. Nous avons fixé l'ordre du canal de la

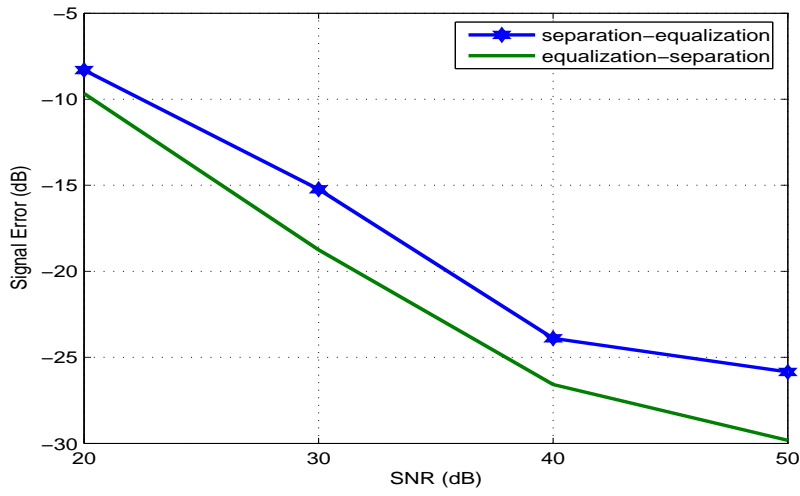


FIG. 5.1 – Comparaison des deux méthodes avec le même ordre du canal pour les deux sources.

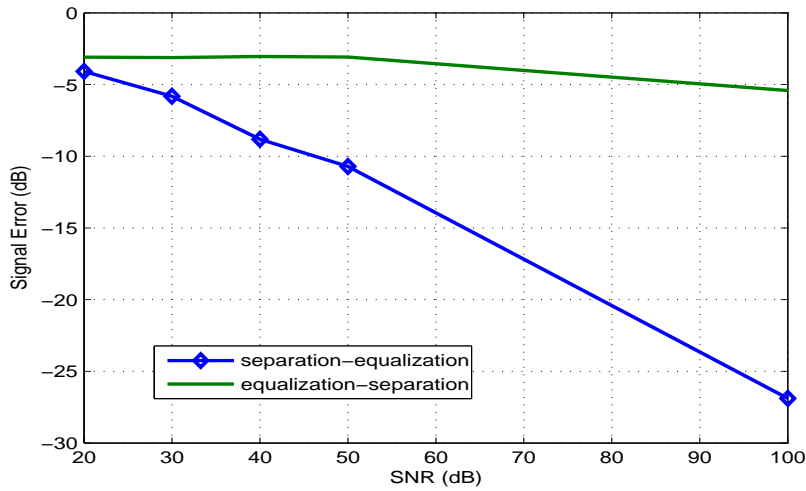


FIG. 5.2 – Comparaison des deux méthodes avec un ordre différent du canal pour les deux sources.

première source à $L_1 = 2$ et celui de la deuxième source à $L_2 = 3$. Cette fois-ci, alors que la méthode classique (*equalization-separation*) est totalement inefficace pour retrouver les sources, notre méthode (*separation-equalization*) a un bien meilleur comportement et on arrive à retrouver les sources avec une erreur acceptable, surtout lorsque le bruit est faible.

Dans ce chapitre, nous avons complété notre technique de séparation, à un filtre près, qui utilise une bloc diagonalisation de matrices d'auto-corrélations blanchies, par une égalisation SIMO pour chaque filtre correspondant à chaque source. Notre technique a un comportement comparable mais néanmoins moins performant que la technique d'égalisation-séparation classique lorsque les ordres des canaux des différentes sources sont identiques. Seulement, lorsque les ordres des canaux des sources sont différents, la méthode que nous proposons s'avère beaucoup plus adéquate et donne un bien meilleur résultat. Ce qui nous permet de dire, que notre technique est plus adaptée dans un contexte aveugle où l'ordre des différents canaux est inconnu. Ce qui lui confère, une relative robustesse par rapport à la surestimation de l'ordre.

Cependant, nous avons remarqué que notre technique est, relativement, sensible au bruit, ainsi qu'à la taille des données. Ce problème est lié à la technique de bloc diagonalisation qui n'est plus performante à partir d'une certaine taille des matrices à bloc diagonaliser et pour des SNR, relativement, bas (voir le chapitre 4).

Chapitre 6

Simulations et évaluation des performances

Dans ce chapitre, nous allons présenter des simulations pour évaluer les performances des algorithmes développés. Pour cela, nous avons opté pour la réalisation de deux plateformes graphiques sous le logiciel *Matlab Version 7.0.1*. La première plateforme graphique permet de tester et d'évaluer les performances des différentes méthodes de bloc diagonalisation conjointe. La deuxième plateforme permet d'évaluer les algorithmes de déconvolution complète du système (séparation-égalisation) et de comparer leurs performances avec une méthode plus classique d'égalisation-séparation. C'est cette dernière plateforme que nous utiliserons, tout le long de ce chapitre. Une description détaillée de ces plateformes est donnée en annexe. Les simulations sont appliquées à deux signaux de paroles. L'erreur quadratique moyenne entre les signaux reconstruits et les signaux d'origine est tracée en fonction du rapport signal sur bruit (SNR) après 100 réalisations de Monte-Carlo. La plage de variation du SNR va de 10 *dB* (signaux avec un bruit fort) à 100 *dB* (signaux avec un bruit négligeable).

6.1 Séparation de signaux stationnaires utilisant l'algorithme NJBD :

Tout le long de cette section, nous étudierons la séparation de signaux stationnaires en utilisant, pour la BDC, l'algorithme NJBD présenté en 4.4.2, à travers un certain

nombre de simulations. Pour cela, nous allons présenter trois groupes de simulations de la méthode proposée en fonction de différents paramètres :

- Influence du bruit.
- Influence du nombre de matrices d’auto-corrélation.
- Influence du nombre d’échantillons décrivant les sources.

6.1.1 Influence du bruit

Notre méthode (séparation-égalisation) est comparée avec la méthode d’égalisation-séparation utilisant la notion de sous-espace présentée en 2.3.2. Pour plus de clarté, les deux couleurs des graphes correspondent à :

vert : L’algorithme ”égalisation-séparation”.

bleu : L’algorithme ”séparation-égalisation”.

L’influence du bruit est étudiée sur deux cas de figure suivant les degrés des canaux des deux signaux de paroles.

Simulation 1.1 :

Lorsque les degrés des canaux sont semblables $L_{S_1} = 2, L_{S_2} = 2$, les variations de l’erreur moyenne est représentée sur la figure 6.1 pour les deux algorithmes.

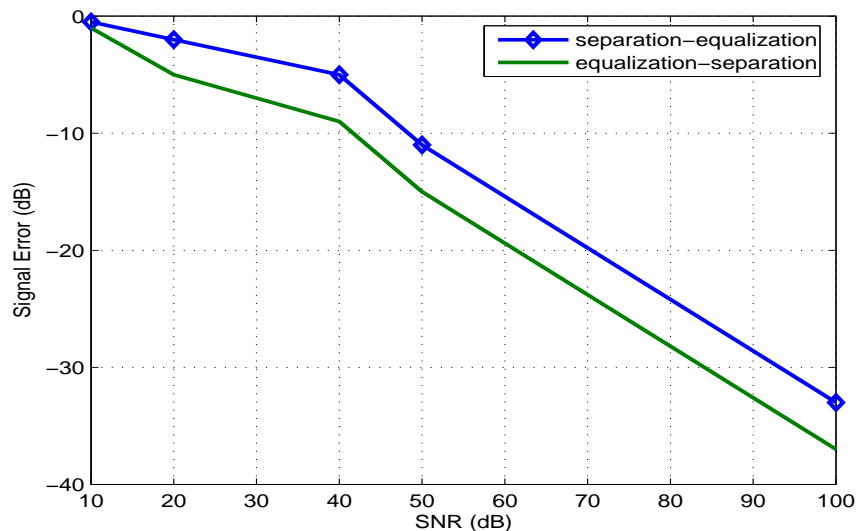


FIG. 6.1 – Comparaison des deux méthodes pour des ordres identiques.

Simulation 1.2 :

Sur la figure 6.2, les degrés des canaux sont différents $L_{S_1} = 2, L_{S_2} = 1$.

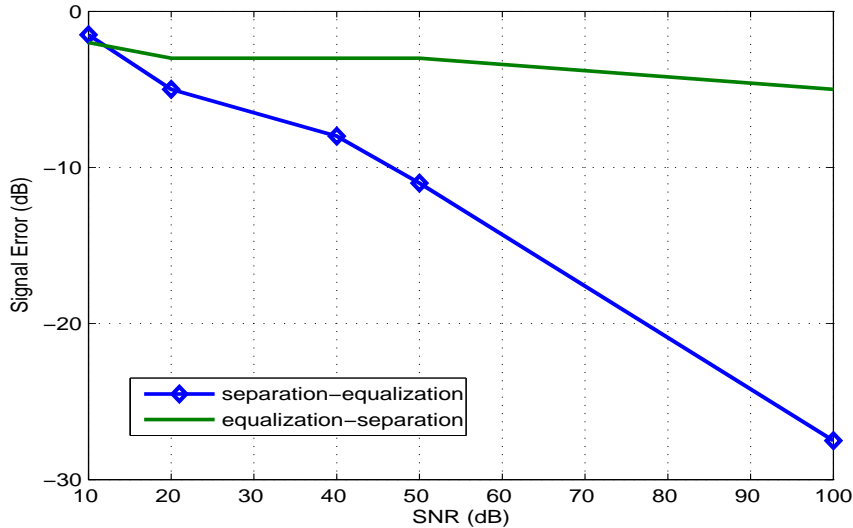


FIG. 6.2 – Comparaison des deux méthodes pour des ordres différents.

Discussion :

Dans la première simulation (figure 6.1), l'algorithme "égalisation-séparation" a un comportement relativement meilleur que notre méthode surtout pour de faibles SNR. Néanmoins les deux approches sont comparables et donnent des résultats semblables pour un fort SNR.

Par contre, dans la deuxième simulation (figure 6.2), l'algorithme "égalisation-séparation" ne converge pas et est, donc, inadéquat lorsque les canaux ont des ordres différents. Alors que, notre méthode (séparation-égalisation) présente de très bons résultats surtout pour de forts SNR. Nous pouvons, donc, présager que la méthode proposée dans ce travail est destinée, plus particulièrement, aux problèmes de séparation lorsque les ordres des différents canaux ne sont pas égaux ou sont inconnus ; ce qui est le cas dans un contexte totalement aveugle.

6.1.2 Influence du nombre de matrices d'auto-corrélation

Cette fois ci, nous étudions l'influence du nombre de matrices d'auto-corrélation des données à bloc diagonaliser sur les performances de notre méthode. Ceci est illustré par la simulation suivante :

Simulation 1.3 :

Ces performances sont traduites en terme d'erreur moyenne en fonction du nombre de matrices à bloc diagonaliser comme le montre la figure 6.3. Notre méthode est appliquée pour un nombre de matrices à bloc diagonaliser différent sur deux expériences :

bleu : Une matrice d'auto-corrélation.

vert : Quatre matrices d'auto-corrélation.

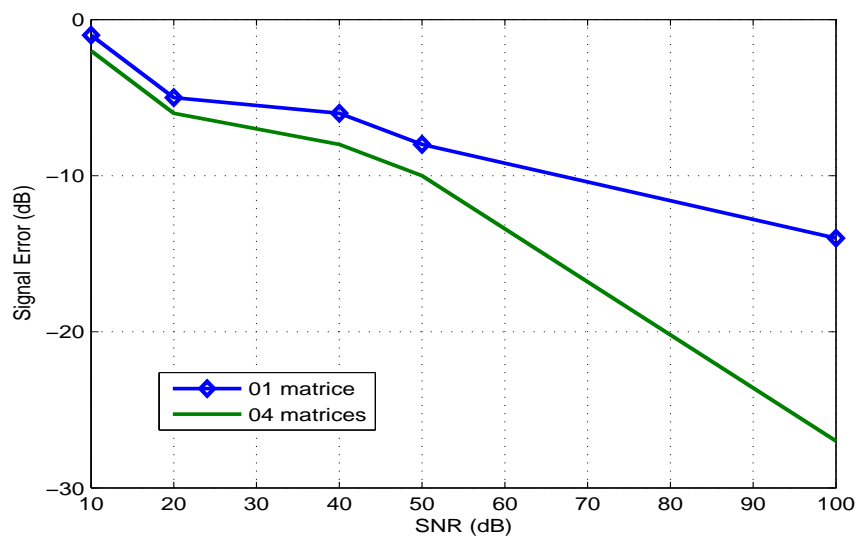


FIG. 6.3 – Comparaison de la méthode proposée avec un nombre de matrices d'auto-corrélation différents.

Discussion :

On voit bien, sur la figure 6.3 et comme cela a été expliqué auparavant en 3.1.2, qu'il est préférable de prendre un grand nombre de matrices d'auto-corrélation pour éviter le cas, éventuel, d'une matrice unique singulière à bloc diagonaliser. En fait, avec une seule

matrice à bloc diagonaliser, l'algorithme a un taux de convergences réussies (rapport entre le nombre de convergences réussies et le nombre total d'essais (voir l'équation (4.33))) beaucoup plus faible.

6.1.3 Influence du nombre d'échantillons

Dans la simulation suivante, le nombre d'échantillons T_s pour chaque source varie à chaque expérience.

Simulation 1.4 :

Sur la figure 6.4, nous avons représenté l'erreur moyenne en fonction du SNR pour trois expériences où les sources sont sous-échantillonnées à chaque expérience.

1. en rouge : $T_s = 16000$ échantillons.
2. en vert : $T_s = 1600$ échantillons.
3. en bleu : $T_s = 160$ échantillons.

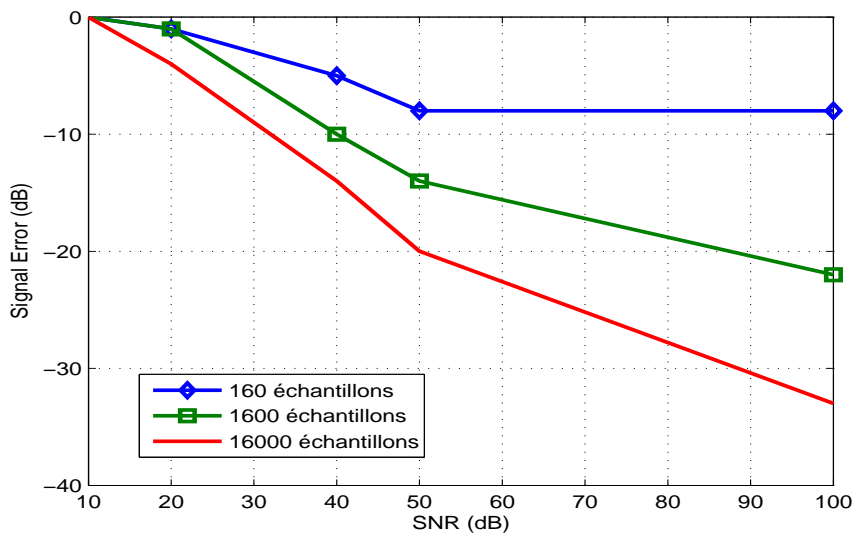


FIG. 6.4 – Erreur moyenne pour un nombre d'échantillons variable.

Discussion :

On remarque, sur la figure 6.4, que plus le nombre d'échantillons est élevé, meilleur est le comportement de notre algorithme en terme d'erreur moyenne.

6.2 Séparation de signaux stationnaires utilisant l'algorithme ALSBD :

Dans cette section, nous effectuons les mêmes simulations pour des sources stationnaires en utilisant pour la BDC l'algorithme ALSBD développé en 4.3.3.

6.2.1 Influence du bruit

De la même manière, notre méthode (séparation-égalisation) est comparée avec la méthode d'égalisation-séparation sous-espace présentée en 2.3.2 :

vert : L'algorithme "égalisation-séparation".

bleu : L'algorithme "séparation-égalisation".

L'influence du bruit est étudiée sur deux cas de figure suivant les degrés des canaux des deux signaux de paroles.

Simulation 2.1 :

Sur la figure 6.5, les degrés des canaux sont semblables $L_{S_1} = 2, L_{S_2} = 2$.

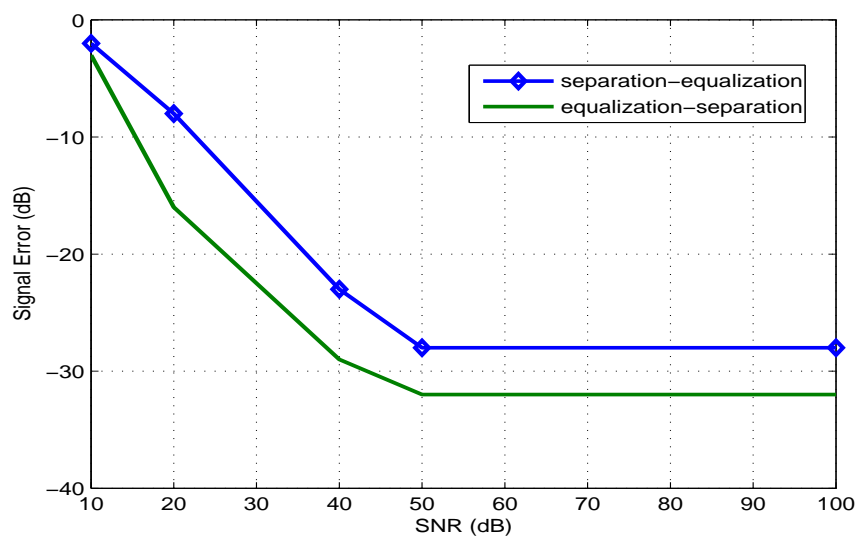


FIG. 6.5 – Comparaison des deux méthodes pour des ordres identiques.

Simulation 2.2 :

Sur la figure 6.6, les degrés des canaux sont différents $L_{S_1} = 2, L_{S_2} = 1$.

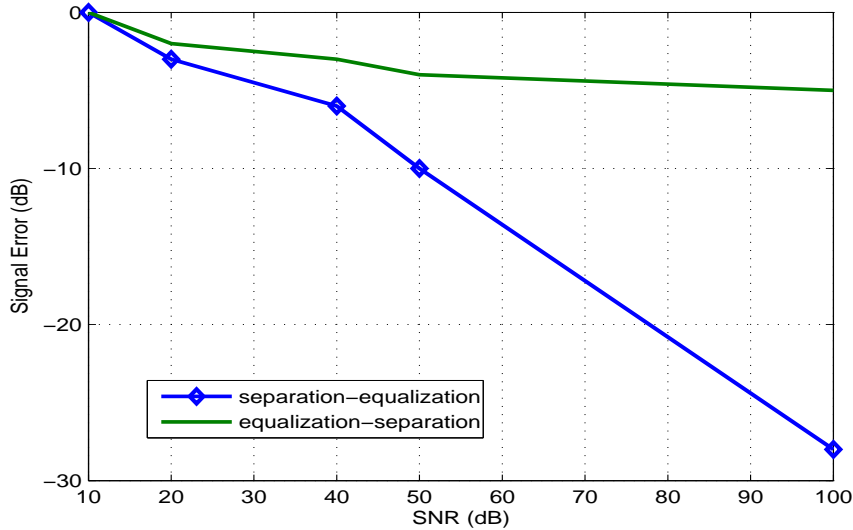


FIG. 6.6 – Comparaison des deux méthodes pour des ordres différents.

Discussion :

Dans la simulation 2.1 (figure 6.5), les deux algorithmes ont un comportement similaire avec une erreur, légèrement, plus faible pour l'algorithme "égalisation-séparation". Par contre, dans la simulation 2.2 (figure 6.6), l'algorithme "égalisation-séparation" ne converge pas lorsque les canaux ont des ordres différents. Alors que, notre méthode présente de très bons résultats surtout pour de forts SNR. Là aussi, la méthode proposée est plus appropriée dans le cas général où les ordres des canaux sont quelconques.

6.2.2 Influence du nombre de matrices d'auto-corrélation

Nous étudions l'influence du nombre de matrices d'auto-corrélation des données à bloc diagonaliser sur les performances de notre méthode. Nous avons effectué 3 simulations avec, respectivement, 1 matrice, 3 matrices et 5 matrices. Le résultat est résumé sur la figure 6.7.

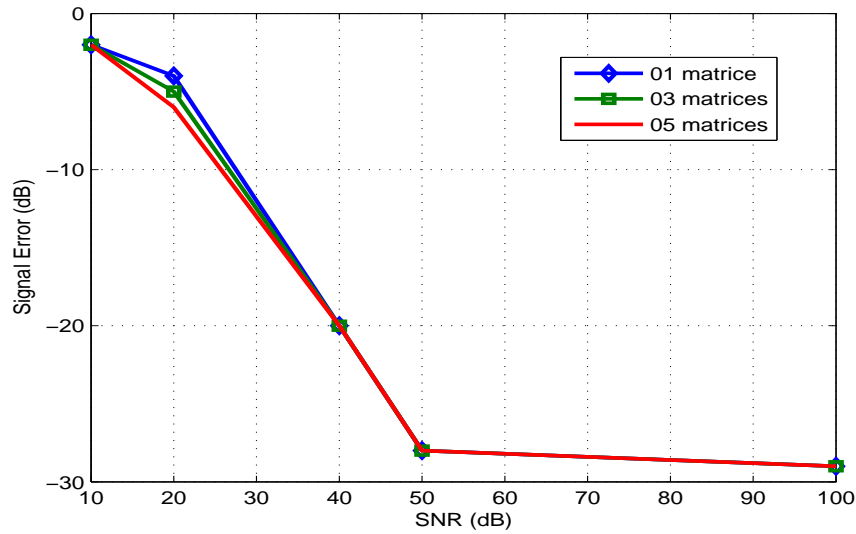


FIG. 6.7 – Comparaison de la méthode proposée avec un nombre de matrices d'auto-corrélation différents.

Simulation 2.3 :

Discussion :

Comme le montre la figure 6.7, le nombre de matrices à bloc diagonaliser n'a aucun effet sur l'erreur moyenne, les trois courbes pour, respectivement, 1 matrice, 3 matrices et 5 matrices, sont, presque, confondues. Un essai d'explication de ce résultat est, à notre avis, que le gain obtenu par l'augmentation du nombre de matrices serait compensé par les erreurs dues à l'approximation faite dans la méthode ALSBD.

6.2.3 Influence du nombre d'échantillons

Dans la simulation suivante, le nombre d'échantillons T_s pour chaque source varie à chaque expérience.

Simulation 2.4 :

Sur la figure 6.8, nous avons représenté l'erreur moyenne en fonction du SNR pour trois expériences où les sources sont échantillonnées différemment à chaque expérience :

1. en rouge : $T_s = 1000$ échantillons.
2. en vert : $T_s = 500$ échantillons.

3. en bleu : $T_s = 100$ échantillons.

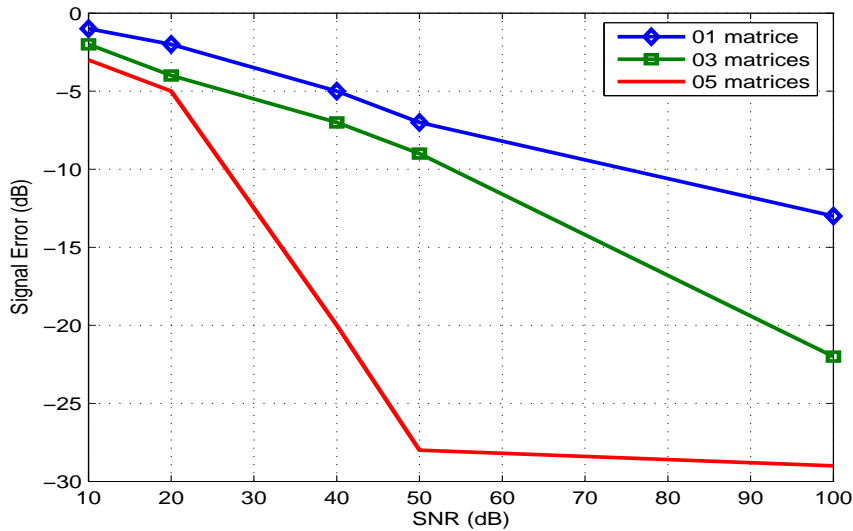


FIG. 6.8 – Erreur moyenne pour un nombre d'échantillons variable.

Discussion :

Sur la figure 6.8, il apparaît que plus le nombre d'échantillons est élevé, meilleure est la séparation.

6.3 Séparation de signaux non-stationnaires utilisant l'algorithme PJBD :

Dans cette section, nous effectuons des simulations similaires aux précédentes pour des sources non-stationnaires en utilisant pour la BDC l'algorithme PJBD présenté en 4.4.1.

6.3.1 Influence du bruit

Notre méthode (séparation-égalisation) est comparée avec la méthode d'égalisation-séparation sous-espace MIMO :

vert : L'algorithme "égalisation-séparation".

bleu : L'algorithme "séparation-égalisation".

Simulation 3.1 :

La figure 6.9 montre la comparaison des deux méthodes lorsque les degrés des canaux sont identiques $L_{S_1} = 2, L_{S_2} = 2$.

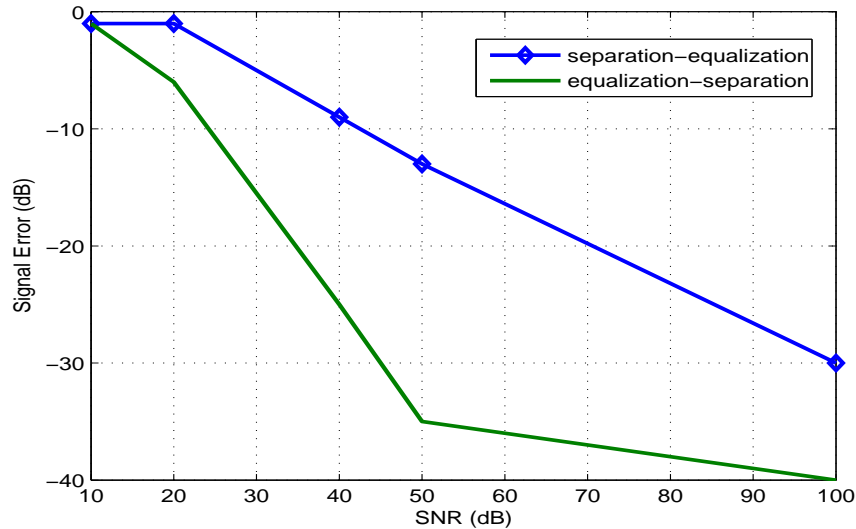


FIG. 6.9 – Comparaison des deux méthodes pour des ordres identiques.

Simulation 3.2 :

Sur la figure 6.10, les degrés des canaux sont différents $L_{S_1} = 2, L_{S_2} = 1$.

Discussion :

Dans la simulation 3.1 (figure 6.9), là aussi, les deux algorithmes ont un comportement similaire que précédemment avec une erreur, légèrement, plus faible pour l'algorithme "égalisation-séparation".

Par contre, dans la simulation 3.2 (figure 6.10), l'algorithme "égalisation-séparation" n'est pas applicable. Alors que, la méthode de séparation-égalisation, présente de très bons résultats surtout pour de forts SNR.

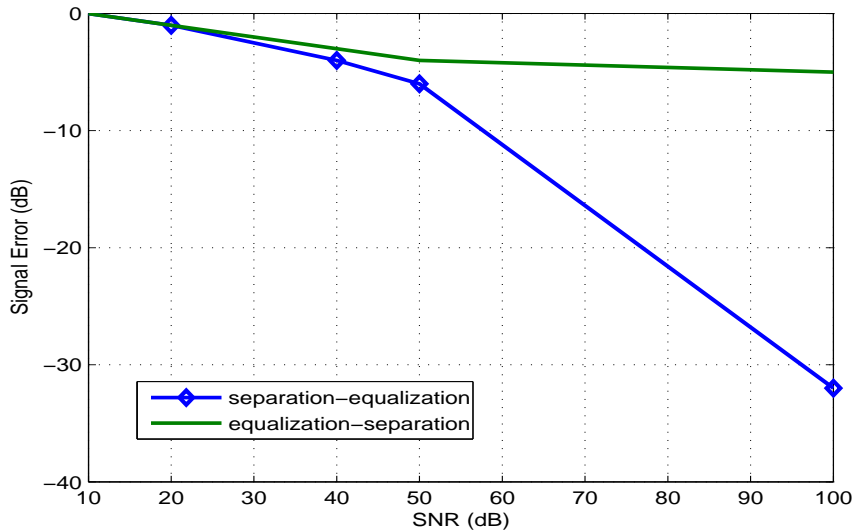


FIG. 6.10 – Comparaison des deux méthodes pour des ordres différents.

6.3.2 Influence du nombre d'échantillons

Dans la simulation suivante, le nombre d'échantillons T_s pour chaque source varie à chacune des deux expériences.

Simulation 3.3 :

Sur la figure 6.11, nous avons représenté l'erreur moyenne en fonction du SNR pour deux expériences :

1. en vert : $T_s = 1000$ échantillons.
2. en bleu : $T_s = 100$ échantillons.

Discussion :

Sur la figure 6.11, il apparaît qu'il est préférable d'avoir un nombre d'échantillons "suffisant" pour réaliser une meilleure séparation.

Dans le cas de la séparation de signaux non-stationnaires utilisant l'algorithme PJBD, nous avons bloc diagonalisé 3 matrices de covariances. Nous avons, aussi, remarqué que si l'on prend 2 matrices de covariances, les résultats sont équivalents à ceux issus de la méthode MP (voir 2.3.3) où l'on utilise les vecteurs et valeurs propres généralisés de

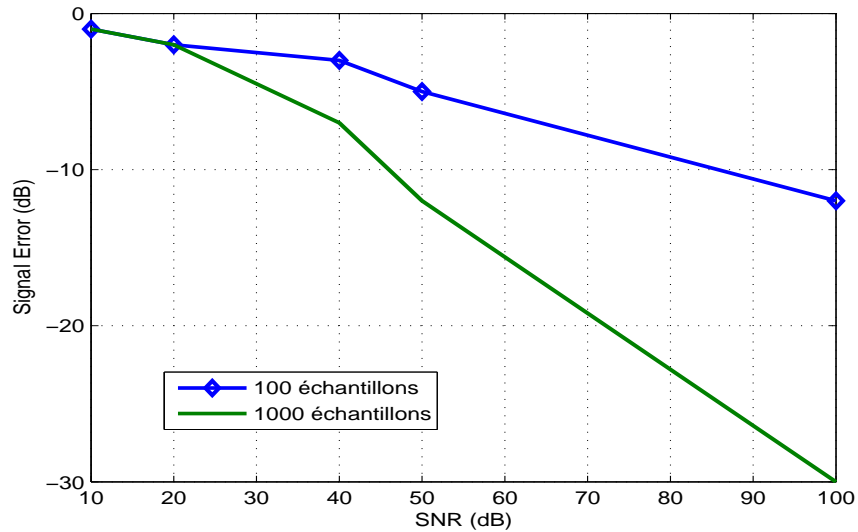


FIG. 6.11 – Erreur moyenne pour un nombre d'échantillons variable.

deux matrices de covariances des données. De plus, à l'issue de l'exécution de l'algorithme, nous avons constaté que les deux matrices de covariances deviennent diagonales, ce qui est un cas particulier de la BDC.

6.4 Comparaison des algorithmes de séparation utilisant différentes méthodes de BDC

Nous allons, dans cette section, faire une comparaison des performances entre les algorithmes de séparation de la méthode proposée utilisant différentes méthodes de BDC.

Simulation 4.1 :

La figure 6.12 représente l'erreur quadratique moyenne en fonction du SNR pour trois méthodes de BDC :

1. en bleu : séparation utilisant NJBD.
2. en vert : séparation utilisant ALSBD.
3. en rouge : séparation utilisant PJBD.

Les paramètres utilisés sont :

- Deux signaux de parole échantillonnés à $22050Hz$.
- Les ordres des canaux pour les deux sources sont égaux $L_{S_1} = 2, L_{S_2} = 2$.
- Le nombre de matrices de corrélations (ou de covariances pour PJBD) est de quatre.
- La fenêtre d'observation a été fixée à $L' = 5$.

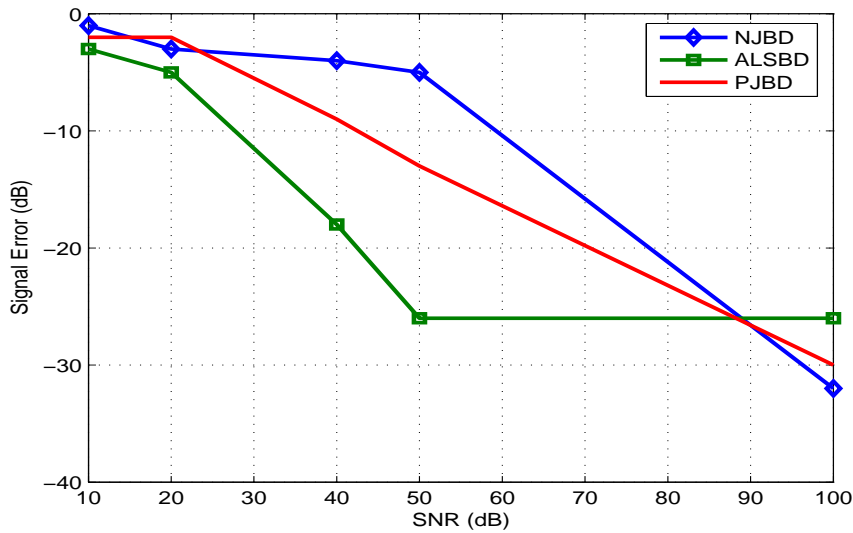


FIG. 6.12 – Comparaison des performances de la méthode proposée pour différents algorithmes de BDC avec des ordres identiques.

Simulation 4.2 :

Dans cette simulation, les mêmes paramètres sont utilisés pour des sources ayant des ordres des canaux différents ($L_{S_1} = 2, L_{S_2} = 1$). La figure 6.13 résume les performances de la séparation pour les trois méthodes de BDC.

Discussion :

Lorsque les ordres sont égaux, la figure 6.12 montre que la méthode ALSBD donne de meilleurs résultats lorsque les signaux sont bruités. Dans le cas où les ordres sont différents (voir la figure 6.13), les performances sont sensiblement équivalentes pour les trois algorithmes de BDC.

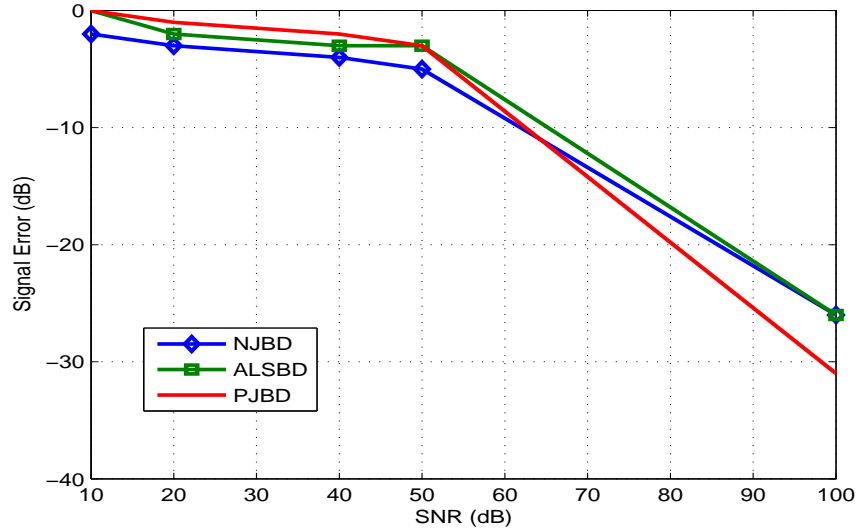


FIG. 6.13 – Comparaison des performances de la méthode proposée pour différents algorithmes de BDC avec des ordres différents.

Nous n'avons pas pu réaliser des simulations de séparation en utilisant l'algorithme LSBD car, cet algorithme est sensible au problème des minima locaux lorsque la taille des blocs des matrices devient, relativement, importante (dans ce cas $L + L' = 7$).

6.5 Comparaison des taux de réussites entre la méthode proposée avec différentes méthodes existantes

Dans le tableau ci-dessous, nous avons regroupé les taux de réussite ou de convergences réussies entre la méthode que nous proposons, en première ligne, la méthode d'égalisation-séparation utilisant la projection en sous-espaces (voir 2.3.2) et la méthode de séparation-égalisation MP résumée en 2.3.3.

Pour cela, nous avons utilisé deux signaux de paroles échantillonnés à 22050 Hz ayant 16000 échantillons chacun dans un contexte sans bruit. Les ordres des canaux sont égaux dans une première expérience ($L_{S_1} = L_{S_2} = 2$) et différents dans une seconde expérience ($L_{S_1} = 2, L_{S_2} = 1$). Les canaux des sources sont fixés d'une manière, complètement, aléatoire à chaque réalisation. Ce qui permet de tester la robustesse de notre méthode vis à vis du changement de canal (contexte mobile par exemple). 1000 réalisations ont été nécessaires pour obtenir ces statistiques. Pour la BDC, nous avons utilisé l'algorithme

ALSBD qui a donné un meilleur taux de réussites compte tenu des tailles des matrices à bloc diagonaliser.

Algorithmes	ordres différents	ordres identiques
Séparation-Egalisation avec BDC	65.5%	77.1%
Egalisation-Séparation avec la méthode sous-espace	4.2%	100%
Séparation-Egalisation MP	41%	100%

Comme le montre le tableau, lorsque les ordres des canaux sont identiques, notre méthode à un taux de réussite, relativement, plus faible que les deux autres méthodes. Néanmoins, dans un contexte plus général où les ordres sont quelconques, les méthodes concurrentes ne peuvent être appliquées et donc ne sont pas robustes vis à vis d'une surestimation de l'ordre.

Chapitre 7

Conclusion et perspectives

Dans ce travail de thèse, nous avons développé des méthodes de séparation aveugle de sources au second ordre à partir de mélanges linéaires convolutifs MIMO. Ces méthodes sont basées sur la bloc diagonalisation conjointe de matrices de statistiques des données. Ces matrices peuvent représenter des auto-corrélations ou des covariances des données reçues. Le principe des méthodes développées se résume en deux étapes : une séparation à un filtre prés qui procure des versions filtrées des sources originales suivie d'une égalisation SIMO qui permet d'estimer chaque source séparément. Les performances des méthodes proposées, du point de vue de l'erreur quadratique et du taux de réussites, sont comparables à ceux de la méthode plus classique d'égalisation suivie d'une séparation ainsi que ceux d'une autre méthode de séparation (MP) lorsque les ordres des canaux relatifs à chaque source sont égaux. Néanmoins, alors que ces méthodes ne sont, en aucun cas, applicables dans le cas où les ordres des canaux relatifs à chaque source sont différents, notre méthode s'est révélée très performante et parfaitement adéquate pour ce cas de figure. Rappelons que ce dernier cas s'inscrit, réellement, dans un contexte aveugle. En effet, supposer que les différents canaux ont le même ordre est un cas très restrictif et s'adapte rarement aux phénomènes physiques. Cependant, nous avons constaté quelques limites à notre méthode dû, essentiellement, aux limites imposées par les techniques de bloc diagonalisation. En effet, lorsque la taille des matrices à bloc diagonaliser est assez grande, la méthode devient moins robuste.

Dans ce contexte, nous avons présenté quatre algorithmes de bloc diagonalisation de matrices différents. Les deux premiers sont basés sur la méthode de Jacobi qui minimise

un critère des moindres carrés, les deux autres effectuent une diagonalisation conjointe pour bloc diagonaliser les matrices à une permutation près. Plusieurs simulations ont été effectuées pour évaluer les performances des algorithmes de séparation ainsi que les algorithmes de bloc diagonalisation à l'aide de deux plate-formes graphiques. Nous nous sommes limités, pour ces simulations, aux exemples de signaux de parole ; néanmoins, la méthode, que nous proposons, s'applique pour tous les signaux mutuellement décorélés. Beaucoup de chemin reste à parcourir pour rendre cette méthode plus robuste. Une suite de ce travail serait d'améliorer la robustesse par rapport à la taille des matrices des données, ce qui revient à améliorer les algorithmes de bloc diagonalisation conjointe. Aussi, il serait judicieux de chercher des solutions pour rendre notre méthode plus robuste par rapport au niveau du bruit. Une autre perspective de ce travail est de bloc diagonaliser les matrices d'auto-corrélation en essayant de forcer la matrice du canal à garder son caractère sylvestre par bloc, ce qui revient à faire une séparation et une égalisation simultanément. Enfin, on peut adopter une méthode de "déflation" pour la séparation : on sépare une première source et on obtient un mélange convolutif des sources restantes et ainsi de suite pour les autres sources.

Toutefois, l'intérêt croissant porté à la SAS pousse à l'optimisme. Et n'oublions pas que la séparation de sources est encore jeune, tout juste majeure...

Bibliographie

- [1] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, and E. Moulines, “Blind source separation using second order statistics.” *IEEE-SP*, vol. 45, no. 2, pp. 434–444, Feb. 1997.
- [2] D. T. Pham, J. F. Cardoso, “Blind separation of instantaneous mixtures of non stationary sources”, *In Proceedings of ICA'2000*, pp. 187-192, Helsinki, Finland, 2000.
- [3] E. Vincent, C. Févotte, and R. Gribonval et al., “A tentative typology of audio source separation tasks”, *In Proc. 4th Symposium on Independent Component Analysis and Blind Source Separation (ICA'03)*, Nara, Japan, Avril 2003.
- [4] R. Vigàrio, and E. Oja ”Independence : A new criterion for the analysis of the electromagnetic fields in the global brain ? *In Neural Networks*, no. 13, pp. 891–907, 2000.
- [5] L. De Lathauwer, B. De Moor, and J. Vandewalle ”Fetal electrocardiogram extraction by blind source subspace separation”, *IEEE Trans on Biomedical Engineering*, vol. 47, no. 5, pp. 567–572, May 2000.
- [6] Y. Deville ”Panorama des applications biomédicales des méthodes de séparation aveugle de sources.” *In Actes 19^e colloque GRETSI sur le Traitement du Signal et des Images*, Paris, Sep. 2003.
- [7] J. Héroult and B. Ans, “Réseaux de neurones à synapses modifiables : décodage de messages sensoriels composites par un apprentissage non supervisé et permanent”, *In C.R. Acad ; Sci.*, pp. 525-528, série III, Paris, 1984.
- [8] J. Héroult, C. Jutten, and B. Ans, “Détection de grandeur primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non

- supervisé”, *In Actes 10eme colloque GRETSI sur le Traitement du Signal et des Images*, pp. 1017-1022, Nice, France, 1985.
- [9] J. L. Lacoume and P. Ruize, “Sources identification : a solution based on cumulants”, *In Proc. IEEE ASSP Workshop V*, Mineapolis, USA, 1988.
- [10] J. F. Cardoso, “Source separation using higher order moments”, *In Proc. ICASSP’89*, pp 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. ICASSP’91*, pp 3109-3112, 1991.
- [12] J. F. Cardoso, “Iterative techniques for blind source separation using only fourth order cumulants”, *In Proc. EUSIPCO*, pp 739-742, 1992.
- [13] P. Comon, “Analyse en composantes indépendantes et identification aveugle”, *Traitement du signal*, Vol. 7, no. 5, pp. 435-450, 1990.
- [14] M. Gaeta, J. L. Lacoume, “Estimateurs du maximum de vraisemblance étendus à la séparation de sources non gaussiennes”, *Traitement du signal*, Vol. 7, no 5, pp. 419-434, 1990.
- [15] D. T. Pham, P. Garat, C. Jutten, “Separation of a mixture of independent sources though a maximum likelihood approach”, *In Proc. EUSIPCO*, pp. 771-774, 1992.
- [16] N. Delfosse, P. Loubaton, “Séparation adaptative de mélange instantané”, *In Actes 14eme colloque GRETSI sur le Traitement du Signal et des Images*, Juan Les Pins, France, 1993.
- [17] B. Laheld and J. F. Cardoso, “Séparation adaptative de sources en aveugle, implémentation complexe sans contraintes”, *In Actes 14eme colloque GRETSI sur le Traitement du Signal et des Images*, Juan Les Pins, France, 1993.
- [18] A. Belouchrani and M. G. Amin, ”Blind source separation based on time-frequency representations.” *IEEE-SP*, vol. 46, no. 11, pp. 2888–2897, Nov. 1998.
- [19] K. Torkkola, “Blind separation of convolved sources based on information maximization”, *in Proc. IEEE Workshop on Neural Networks and Signal Processing*, Kyota, Japan, 1996.

- [20] A. J. Bell, T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", *Neural Comput.*, Vol. 7, pp. 1129-1159, 1995.
- [21] T. W. Lee, A. J. Bell, R. Lambert, "Blind separation of delayed and convolved sources", *In Advances in Neural Information Processing Systems 9.*, pp. 758-764, Cambridge, MA : MIT Press, 1997.
- [22] P. Smaragdis, "Information Theoretic Approaches to Source Separation", *M.Sc. thesis, MIT Media Lab*, Juin 1997.
- [23] L. Parra and C. Spence, "Convulsive blind separation of non-stationary sources", *IEEE Trans. Speech and Audio Processing*, pp. 320-327, Mai 2000.
- [24] P. Loubaton, "Signal et télécoms", *Hermes-Sciences Lavoisier*, pp. 17-54, 2004.
- [25] Y. Hua and M. Wax, "Strict identifiability of multiple FIR channels driven by an unknown arbitrary sequence", *IEEE Trans. on Signal Processing*, Vol. 44, Issue 3, pp. 756-759, March 1996.
- [26] D. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems", *IEEE Trans. on Communications*, Vol. Com-28, no. 11, pp. 1867-1875, Nov. 1980.
- [27] K. Abed-Meraim, W. Qiu and Y. Hua, "Blind system identification", *Proceedings of the IEEE*, Vol. 85, Issue 8, pp. 1310-1322, Aug. 1997.
- [28] E. Moulines, P. Duhamel, J. F. Cardoso and S. Mayrargue, "Subspace Methods for the Blind Identification of Multichannel FIR Filters", *IEEE on Signal Processing*, vol. 43, no. 2, pp. 516-525, February 1995.
- [29] A. Gorokhov and P. Loubaton, "Subspace-based Techniques for Blind Separation of Convulsive Mixtures with Temporally Correlated Sources", *IEEE Trans on Circuits and Systems*, vol. 44, no. 9, pp. 813-820, September 1997.
- [30] Y. Hua, S. An and Y. Xiang "Blind Identification of FIR MIMO Channels by Decorrelating Subchannels", *IEEE Trans on Signal Processing*, vol. 51, no. 5, pp. 1143-1151, May 2003.
- [31] C. Tin Ma, Z. Ding and S. Fong Yau "A Two-Stage Algorithm for MIMO Blind Deconvolution of Nonstationary Colored Signals", *IEEE Trans on Signal Processing*, vol. 48, no. 4, pp. 1187-1192, April 2000.

- [32] H. Bousbia-Salah and A. Belouchrani "A Second Order Multi Output Deconvolution", *Proceedings of the Tenth IEEE Workshop On Statistical Signal and Array Processing*, pp. 306–310, Poconos, August 2000.
- [33] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis", New York, Wiley, pp. 246–248.
- [34] H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, "Blind Separation of Convolutional Mixtures using Joint Block Diagonalization." in *Proc. of the 6th International Symposium on Signal Processing and its Applications.*, Kuala Lumpur, Malaysia, August 2001.
- [35] H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, "Jacobi-like Algorithm for Blind Signal Separation of Convolutional Mixtures.", *Electronics Letters*, vol. 37, no.16, pp. 1049–1050, August 2001.
- [36] H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, "Blind Separation of Non Stationary Sources using Joint Block Diagonalization." *The 11th IEEE Workshop on Statistical Signal Processing*, Singapore, August 2001.
- [37] A. Belouchrani, M. Amin and K. Abed-Meraim, "Direction finding in correlated noise fields based on joint block-diagonalization of spatio-temporal correlation matrices.", *IEEE-SP Letters*, pp. 266–268, Sep. 1997.
- [38] A. Belouchrani, K. Abed-Meraim, and Y. Hua, "Jacobi-like Algorithms for Joint Block Diagonalization : Application to Source Localization", in *Proc. IEEE Int. Work. on Intelligent Sig. Proc. and Comm. Sys.*, Melbourne, 1998.
- [39] L. Tong, G. Xu and Kailath, "A new approach to blind identification and equalization of multipath channels", in *Proc. of the 25th Asilomar Conference*, CA, pp. 856–860, 1991.
- [40] D.T. Pham and J.F. Cardoso, "Blind Separation of Instantaneous Mixtures of Nonstationary Sources", *IEEE Trans on Signal Processing*, vol. 49, no. 9, pp. 1837–1848, September 2001.
- [41] B. Boashash, "Time-frequency Signal Analysis and Processing : A Comprehensive Reference", *Elsevier*, pp. 323–333, 2003.

- [42] A. Bousbia-Salah, A. Belouchrani and H. Bousbia-Salah, "A One Step Time-Frequency Blind Identification", in *Proc. of the 7th International Symposium on Signal Processing and its Applications.*, Paris, France, July 2003.
- [43] Y. Hua and K. Abed-Meraim, "Techniques of eigenvalues estimation and association.", *Digital Signal Processing*, vol. 7, no. 4, pp. 253–259, October 1997.
- [44] M. Haardt, K. Hueper, J.B. Moore, and J.A. Nossek, "Simultaneous Schur decomposition of several matrices to achieve automatic pairing in multidimensional harmonic retrieval problem", *Proc. EUSIPCO*, pp. 531–534, 1996.
- [45] K. Abed-Meraim and Y. Hua, "A Least-Squares Approach to Joint Schur Decomposition", *Proc. ICASSP (Seattle, USA)*, IV, pp. 2541–2544, 1998.
- [46] Alle-Jan van der Veen, A. Paulraj, "An analytical constant modulus algorithm", *IEEE-SP*, pp. 1136–1155, May 1996.
- [47] H. Buchner, R. Aichner, W. Kellermann, "A Generalization of Blind Source Separation Algorithms for Convulsive Mixtures Based On Second-Order Statistics", *IEEE-SAP*, vol.13, no.1, pp. 120–134, Jan. 2005.
- [48] Q.H. Spencer, A.L. Swindlehurst and M. Haardt, "Zero-Forcing Methods for Downlink Spatial Multiplexing in Multiuser MIMO Channels", *IEEE-SP*, vol.52, no.2, pp. 461–471, Feb. 2004.
- [49] P.J. Eberlein, "On one-sided Jacobi methods for parallel computation", *SIAM J. Algebraic Discrete Meth.* vol. 8, pp. 790–796, 1987.
- [50] G. H. Golub and C. F. Van Loan, "Matrix computations", *The Johns Hopkins University Press*, 1989.
- [51] J.F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization.", *SIAM Journal Mat. Anal. Appl.*, vol. 17, no. 1, pp. 161-164, January 1996.
- [52] M. Wax and J. Scheinvald, "A least-squares approach to joint diagonalization", *IEEE-SP Letters*, vol. 4, pp. 52–53, Feb. 1997.
- [53] M. Haardt, "Efficient One-, Two, and Multidimensional High-Resolution Array Signal Processing", *Shaker Verlag, Aachen*, 1997.
- [54] D. Pham, "Joint approximate diagonalization of positive definite Hermitian matrices," *Technical report LMC/IMAG*, Apr. 1999.

- [55] K. Abed-Meraim, Ph. Loubaton and E. Moulines, "A subspace algorithm for certain blind identification problems", *IEEE Tr. on IT.*, vol. 43, pp. 499-511, Mar. 1997.
- [56] A. Belouchrani, K. Abed-Meraim and H. Bousbia-Salah, "Joint Block Diagonalization For Array Signal Processing", *IEEE Tr. on Sig. Proc. (submitted)*, Dec. 2005.
- [57] H. Liu, G. Xu and L. Tong, "A deterministic approach to blind symbol estimation", *Signal Processing Lett.*, vol. 1, pp. 205–207, Dec. 1994.
- [58] A. Gorokhov, M. Kristensson and B. Ottersten, "Robust Blind Second-Order Deconvolution", *IEEE-SP Letters*, vol. 6, no.1, pp. 13-16, Jan.1999.
- [59] J.K. Tugnait, "A multidelay whitening approach to blind identification and equalization of SIMO channels", *IEEE Tr. on Wireless Communications* Vol. 1, Issue 3, pp. 456–467, July 2002.

Annexes

Annexe A : Démonstration de l'équation (4.11)

En utilisant les équations (4.7) et les égalités $c^2 = (\cos(2\theta) + 1)/2$, $|s|^2 = (1 - \cos(2\theta))/2$, et $cs = \sin(2\theta)e^{i\alpha}/2$, nous obtenons : (des expressions similaires seront obtenues si on remplace p par q)

$$|\mathbf{M}'_k(p, j)|^2 - \text{constante} = -\mathbf{x}_{p,kj}^T \mathbf{v} \quad j \neq p \quad (7.1)$$

$$|\mathbf{M}'_k(j, p)|^2 - \text{constante} = -\mathbf{y}_{p,kj}^T \mathbf{v} \quad j \neq p \quad (7.2)$$

$$|\mathbf{M}'_k(p, p)|^2 - \text{constante} = \mathbf{v}^T \mathbf{G}_{p,k} \mathbf{v} + \mathbf{g}_{p,k}^T \mathbf{v} \quad (7.3)$$

Où *constante* représente des termes indépendents de (θ, α) . Alors,

$$\mathbf{G} = \sum_{k=1}^K (\mathbf{G}_{p,k} + \mathbf{G}_{q,k})$$

$$\mathbf{g} = \sum_{k=1}^K \left(\mathbf{g}_{p,k} + \mathbf{g}_{q,k} - \sum_{j \in \mathcal{I}_p, j \neq p} (\mathbf{x}_{p,kj} + \mathbf{y}_{p,kj}) - \sum_{j \in \mathcal{I}_q, j \neq q} (\mathbf{x}_{q,kj} + \mathbf{y}_{q,kj}) \right)$$

$(\mathbf{G}_{p,k}, \mathbf{G}_{q,k}), (\mathbf{g}_{p,k}, \mathbf{g}_{q,k}), (\mathbf{x}_{p,kj}, \mathbf{x}_{q,kj}),$ et $(\mathbf{y}_{p,kj}, \mathbf{y}_{q,kj})$ sont donnés par :

$$\begin{aligned} \mathbf{G}_{p,k} &= \Re e(\mathbf{z}_{p,k} \mathbf{z}_{p,k}^H), & \mathbf{G}_{q,k} &= \Re e(\mathbf{z}_{q,k} \mathbf{z}_{q,k}^H) \\ \mathbf{g}_{p,k} &= \Re e((\overline{\mathbf{M}}_k(p,p) + \overline{\mathbf{M}}_k(q,q)) \mathbf{z}_{p,k}), & \mathbf{g}_{q,k} &= \Re e((\overline{\mathbf{M}}_k(p,p) + \overline{\mathbf{M}}_k(q,q)) \mathbf{z}_{q,k}) \\ \mathbf{z}_{p,k} &= \frac{1}{2} \begin{bmatrix} \mathbf{M}_k(p,p) - \mathbf{M}_k(q,q) \\ -(\mathbf{M}_k(q,p) + \mathbf{M}_k(p,q)) \\ i(\mathbf{M}_k(p,q) - \mathbf{M}_k(q,p)) \end{bmatrix}, & \mathbf{z}_{q,k} &= \frac{1}{2} \begin{bmatrix} \mathbf{M}_k(q,q) - \mathbf{M}_k(p,p) \\ (\mathbf{M}_k(q,p) + \mathbf{M}_k(p,q)) \\ i(\mathbf{M}_k(p,q) - \mathbf{M}_k(q,p)) \end{bmatrix} \\ \mathbf{x}_{p,kj} &= -\mathbf{x}_{q,kj} = \begin{bmatrix} (|\mathbf{M}_k(q,j)|^2 - |\mathbf{M}_k(p,j)|^2)/2 \\ \Re e(\mathbf{M}_k(p,j) \overline{\mathbf{M}}_k(q,j)) \\ \Im m(\mathbf{M}_k(p,j) \overline{\mathbf{M}}_k(q,j)) \end{bmatrix} \\ \mathbf{y}_{p,kj} &= -\mathbf{y}_{q,kj} = \begin{bmatrix} (|\mathbf{M}_k(j,q)|^2 - |\mathbf{M}_k(j,p)|^2)/2 \\ \Re e(\overline{\mathbf{M}}_k(j,p) \mathbf{M}_k(j,q)) \\ \Im m(\overline{\mathbf{M}}_k(j,p) \mathbf{M}_k(j,q)) \end{bmatrix} \end{aligned}$$

Dans le cas où les \mathbf{M}_k sont des matrices à valeurs réelles, le vecteur \mathbf{v} devient $\mathbf{v} = [\cos(2\theta), \sin(2\theta)]^T$ et la matrice \mathbf{G} (resp. le vecteur \mathbf{g}) sera donnée par le bloc $[2 \times 2]$ se trouvant en haut et à gauche (resp. par les deux éléments $[2 \times 1]$ du haut du vecteur).

Annexe B : Démonstration de l'équation (4.19)

En utilisant les équations (7.1) et (7.2), nous aurons

$$\tilde{\mathbf{g}} = - \sum_{k=1}^K \left(\sum_{j \in \mathcal{I}_p} (\mathbf{x}_{p,kj} + \mathbf{y}_{p,kj}) + \sum_{j \in \mathcal{I}_q} (\mathbf{x}_{q,kj} + \mathbf{y}_{q,kj}) \right)$$

Pour maximiser (4.19), on annule ses dérivées partielles, i.e.,

$$\frac{\partial}{\partial \alpha} \mathcal{C}(\theta, \alpha) = \tilde{\mathbf{g}}^T \frac{\partial \mathbf{v}}{\partial \alpha} = 0, \quad \frac{\partial}{\partial \theta} \mathcal{C}(\theta, \alpha) = \tilde{\mathbf{g}}^T \frac{\partial \mathbf{v}}{\partial \theta} = 0 \quad (7.4)$$

Et

$$\frac{\partial \mathbf{v}}{\partial \alpha} = \sin 2\theta \begin{bmatrix} 0 \\ -\sin \alpha \\ \cos \alpha \end{bmatrix}, \quad \frac{\partial \mathbf{v}}{\partial \theta} = 2 \begin{bmatrix} -\sin 2\theta \\ \cos 2\theta \cos \alpha \\ \cos 2\theta \sin \alpha \end{bmatrix} \quad (7.5)$$

En remplaçant (7.5) dans (7.4), nous aurons

$$\begin{aligned} \tan \alpha &= \frac{\Im m(a)}{\Re e(a)} \iff \alpha = \arctan\left(\frac{\Im m(a)}{\Re e(a)}\right) \\ \tan 2\theta &= -2 \frac{\Re e(e^{-i\alpha} a)}{b} \iff \theta = \frac{1}{2} \arctan\left(-2 \frac{\Re e(e^{-i\alpha} a)}{b}\right) + k \frac{\pi}{2} \end{aligned}$$

Où $k \in \{0, 1\}$ est choisi de manière à maximiser (4.19). Après calculs,

$$\tilde{\mathbf{g}}^T \mathbf{v} = \cos 2\theta \times c$$

où $c = (b^2 + 4\Re e(e^{-i\alpha} a)^2)/(2b)$. Donc, (4.19) est maximisé si $\text{sign}(\cos 2\theta) = \text{sign}(c) = \text{sign}(b)$. Puisque $\arctan\left(-2 \frac{\Re e(e^{-i\alpha} a)}{b}\right) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, nous obtenons finalement $k = (1 - \text{sign}(b))/2$.

Annexe C

Nous présentons ci-dessous le pseudo-code qui permet de trouver la permutation manquante pour une seule matrice \mathbf{M} de taille $[n \times n]$ qui est bloc diagonale à une matrice de permutation prés.

```
% valeur du seuil
eps=0.0001 ;
% Selection des elements – non nuls
Mbis=abs(M).* (abs(M)>eps) ;
Mtest=Mbis>0 ;
% Calcul de la permutation
i=0 ; Sp=[1 :n] ;
while i<n,
    ind=find(Mtest(Sp(1), :)>0) ;
    for k=1 :length(ind) ;
        i=i+1 ;
        S1(i)=ind(k) ;
        Ip=find(abs(Sp-ind(k))>0) ;
        Sp=Sp(Ip) ;
    end
end
% Application de la permutation
M=M(S1,S1) ;
```

Annexe D :

Description de la plateforme graphique

Afin d'évaluer les performances des algorithmes développés, nous avons opté pour la réalisation de deux plate-formes graphiques sous le logiciel *Matlab Version 7.0.1*. La première plate-forme graphique permet de tester et d'évaluer les performances des différentes méthodes de bloc diagonalisation conjointe. La deuxième permet d'évaluer les algorithmes de déconvolution complète du système (séparation-égalisation) et de comparer ces performances avec la méthode plus classique d'égalisation-séparation. Ces deux plate-formes offrent la possibilité de tester tous ces algorithmes individuellement ou en groupe. Plusieurs paramètres peuvent être modifiés par l'utilisateur en fonction de l'algorithme considéré. Ces plate-formes graphiques permettent aussi d'ajouter de nouveaux algorithmes dans le futur. Ce qui donne un large choix à l'utilisateur.

Plate-forme graphique pour la bloc diagonalisation conjointe

La figure 7.1 représente la plate-forme pour la BDC qui est divisée en plusieurs parties :

Paramètres : Ce panneau permet d'ajuster les différents paramètres de la BDC souhaitée.

Certains concernent les matrices comme le nombre de matrices à bloc diagonaliser, la taille d'un bloc élémentaire, le nombre de blocs par matrice. Un autre paramètre peut être ajusté comme le nombre de réalisations de Monte Carlo d'un algorithme. On peut aussi choisir l'algorithme de BDC parmi les quatre proposés : NJBD, LSBBD, ALSBD et PJBD. Enfin, si l'utilisateur le désire, il peut sauvegarder le graphe précédent en vue d'une comparaison éventuelle entre des algorithmes différents ou entre différents paramètres d'un même algorithme.

Graphe $Rho=f(SNR)$: Ce graphe, situé dans la partie supérieure de la plate-forme, permet de représenter la qualité de la BDC en fonction du SNR (exprimé en dB). C'est le rapport entre les normes des blocs non-diagonaux sur les blocs diagonaux (voir l'équation (4.32)). Ce qui traduit l'erreur normalisée en fonction du niveau du bruit.

Graphe $Rho=f(itérations)$: Le deuxième graphe situé sur la partie centrale de la plate-forme graphique représente la qualité de la BDC en fonction du nombre d'itérations.

Cela traduit la vitesse de convergence de l'algorithme pour effectuer une BDC. Ce graphe est tracé pour un fort SNR (50dB).

Grappe $\text{Eta}=f(\text{SNR})$: Ce graphe, situé dans la partie inférieure de la plate-forme, permet de représenter le pourcentage du taux de convergences réussies en fonction du SNR (exprimé en dB). C'est le rapport entre le nombre de convergences réussies et le nombre total d'essais (voir l'équation (4.33)).

Bouton RUN : Le rôle de ce bouton est d'exécuter l'algorithme choisi en fonction des paramètres.

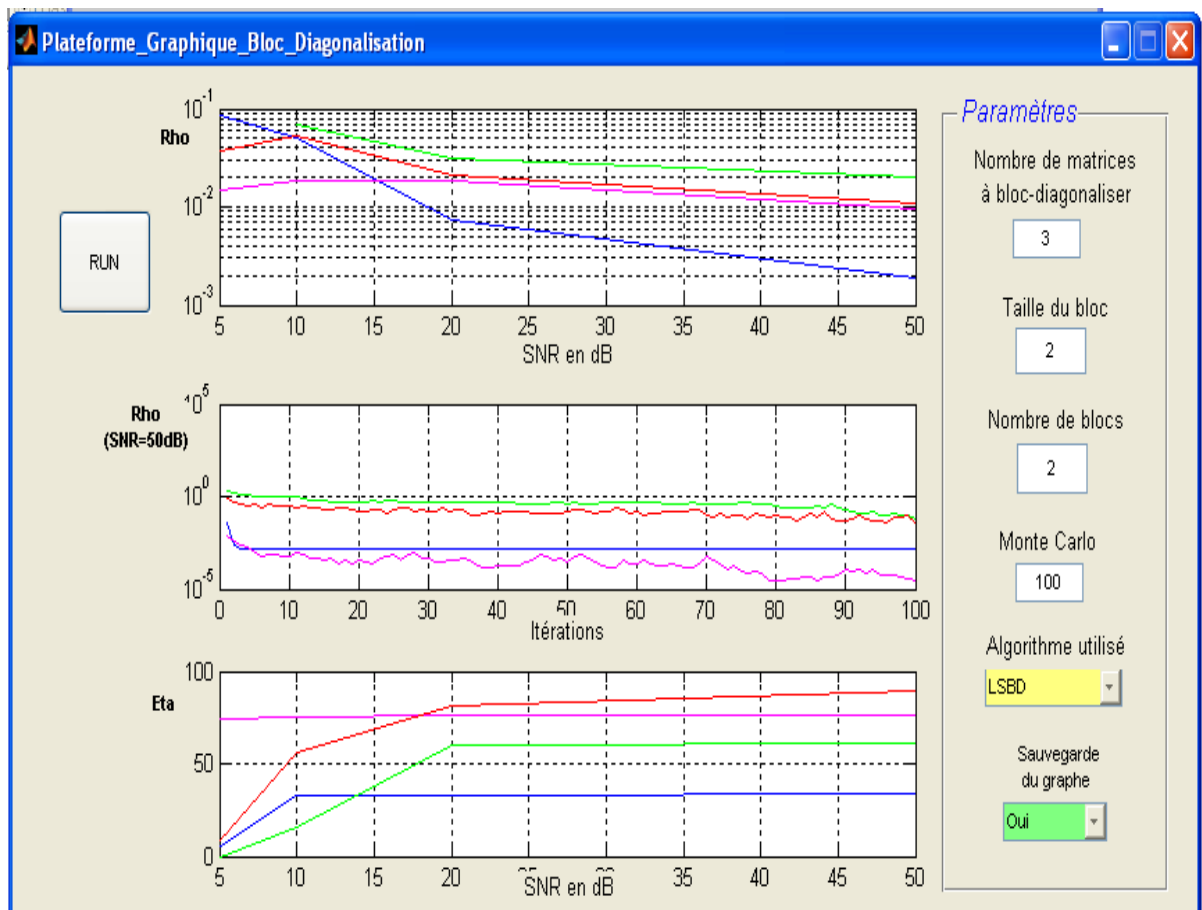


FIG. 7.1 – Plate-forme graphique pour la BDC.

Nous avons inclus dans les deux plate-formes une fenêtre qui permet de suivre le programme d'un algorithme et d'estimer le temps restant à son exécution (figure 7.2).

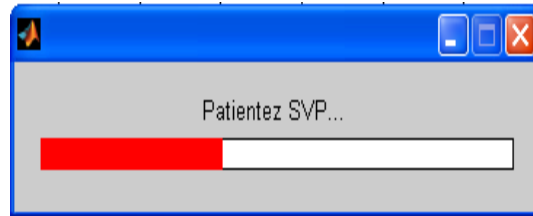


FIG. 7.2 – Fenêtre d'exécution d'un programme.

Plate-forme graphique pour la séparation

La figure 7.3 représente la plate-forme de séparation que l'on peut décrire comme suit :

Paramètres : Ce panneau permet d'ajuster les différents paramètres de l'algorithme de déconvolution complète. Nous avons fixé le nombre de sources à 2. L_{s_1} et L_{s_2} représentent les ordres des canaux des sources s_1 et s_2 , respectivement. Le nombre de capteurs ainsi que le nombre de matrices d'auto-corrélation peuvent être ajustés. De même, nous pouvons choisir l'algorithme utilisé à partir d'un menu : l'égalisation MIMO suivie d'une séparation ou bien une séparation à un filtre prés suivie d'une égalisation SIMO. D'autre part, pour tracer la courbe de l'erreur quadratique moyenne entre les signaux reconstruits et les signaux d'origine, le nombre de réalisations de Monte Carlo d'un algorithme donné peut être fixé. Enfin, si l'utilisateur le désire, il peut sauvegarder le graphe précédent en vue d'une comparaison éventuelle entre les deux algorithmes où entre différents paramètres d'un même algorithme.

Signaux de paroles : Ce panneau, situé dans la partie supérieure droite de la plate-forme, permet de visualiser, dans le cas non-bruité, les deux signaux originaux, trois des signaux de mélange et, enfin, les signaux reconstruits en appuyant sur le bouton **Run**.

Graphe $Signal\ Error=f(SNR)$: Ce graphe, situé sur la partie inférieure de la plate-forme graphique, représente l'erreur quadratique moyenne entre les signaux d'origine et les signaux reconstruits en fonction du rapport Signal sur bruit. Ce graphe est activé par le bouton **Plot Error**.

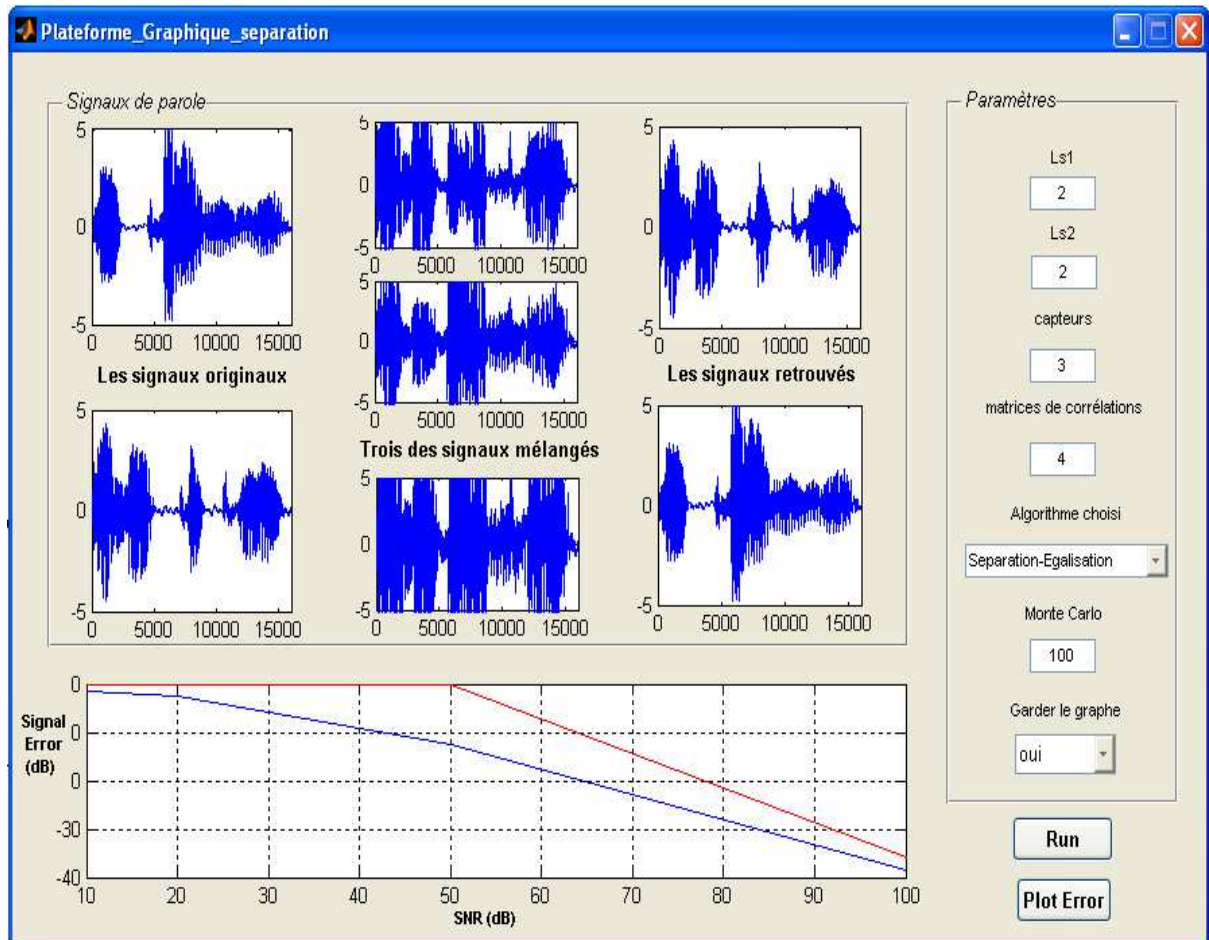


FIG. 7.3 – Plate-forme graphique pour la séparation.

Annexe E :

Publications

1. H. Bousbia-Salah and A. Belouchrani "A Second Order Multi Output Deconvolution", in *Proceedings of the Tenth IEEE Workshop On Statistical Signal and Array Processing*, pp. 306–310, Poconos, August 2000.
2. H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, "Blind Separation of Non Stationary Sources using Joint Block Diagonalization." in *the 11th IEEE Workshop on Statistical Signal Processing*, pp. 448–451, Singapore, August 2001.
3. H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, " Blind Separation of Convulsive Mixtures using Joint Block Diagonalization." in *Proc. of the 6th International Symposium on Signal Processing and its Applications.*, pp. 13–16, Kuala Lumpur, Malaysia, August 2001.
4. H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, "Séparation Aveugle de Mélanges Convolutifs utilisant une Bloc-Diagonalization Conjointe", in *First Polytechnic Symposium on Signal Processing.*, ENP, Algiers, April 2002.
5. H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, "Jacobi-like Algorithm for Blind Signal Separation of Convulsive Mixtures", in *Electronics Letters*, vol. 37, no.16, pp. 1049–1050, August 2001.
6. A. Belouchrani, K. Abed-Meraim and H. Bousbia-Salah " Joint Block Diagonalization For Array Signal Processing", *IEEE Transactions on Signal Processing (submitted)*, Dec. 2005.

A SECOND ORDER MULTI OUTPUT DECONVOLUTION (SOMOD) TECHNIQUE

Hicham Bousbia-Salah and Adel Belouchrani

Electrical Engineering Department,
Ecole Nationale Polytechnique,
P.O. Box 182 El-Harrach, Algiers 16200, Algeria.
(e-mail: hichambs@yahoo.fr, belouchrani@hotmail.com)

ABSTRACT

In this paper, we present an efficient solution to the blind multi-channel deconvolution problem that consists of recovering independent source signals from their convolutive mixtures. In the case of instantaneous mixtures, a robust solution referred to as Second Order Blind Identification (SOBI) has been proposed previously. It is based on the joint diagonalization of spatio-temporal correlation matrices. Herein, we extend this technique to the convolutive mixture case. In contrast to existing deconvolution techniques, this new approach is able to deal with an overestimated source number. The proposed method has been successfully applied to the deconvolution of speech signals.

1. INTRODUCTION

If we consider a set of received signals that are linear convolutive mixtures of decorrelated source signals, the objective of blind deconvolution is to recover the source signals from the set of received signals without any knowledge of the linear mixtures or the Linear Time Invariant (LTI) systems. For instantaneous mixtures, a Second Order Blind Identification (SOBI) algorithm has been presented [1] and showed to be very robust for temporally correlated sources. There are two ways to achieve blind deconvolution. One way is to first identify the channel system from the output mixtures and then to design an equalizer accordingly [2]. The other way consists of directly designing an equalizer from the output mixtures. This approach bypasses the problem of blind system identification and is less costly in computation. Using the second approach, we extend the SOBI technique to the convolutive mixture case. It is based on the joint diagonalization of spatio-temporal corre-

The Authors would like to thank STEP Alger, distributor of Motorola in Algeria, for its support for the presentation of this work.

lation matrices. The proposed method has been successfully applied to the deconvolution of speech signals and showed to be robust with respect to additive noise. Furthermore, this new approach is able to deal with an overestimated source number. In the next section, we will present the data model, the different hypothesis and the identifiability conditions. The proposed algorithm will be described in section 3. And finally, some simulation results are provided in section 4.

2. PROBLEM FORMULATION

2.1. Data Model

Consider a discrete time multiple input multiple output (MIMO) linear time invariant model given by,

$$x_i(n) = \sum_{j=1}^M \sum_{l=0}^{L-1} h_{ij}(l) s_j(n-l) + n_i(n), \text{ for } i = 1, \dots, N \quad (1)$$

where $s_j(n)$, $j = 1, \dots, M$ are the M source signals (model inputs), $x_i(n)$, $i = 1, \dots, N$, are the N sensor signals (model outputs) with $N \geq M$, h_{ij} is the transfer function between the j -th source and the i -th sensor with an overall extend L , and $n_i(n)$, $i = 1, \dots, N$, are additive white noises.

The assumptions made about the data model are as follows:

A1) The source signals $s_j(n)$, $j = 1, \dots, M$, are mutually decorrelated and each source signal is temporally coherent.

A2) The noise processes $n_i(n)$, $i = 1, \dots, N$, are zero-mean stationary processes independent of the source signals.

The purpose of blind multi channel deconvolution is to recover the source signals based only on the sensor signals. This leads to find a set of weights $\{w_{ji}(l)\}$ such

that,

$$\tilde{s}_j(n) = \sum_{i=1}^N \sum_{l=0}^{L-1} w_{ji}(l)x_i(n-l), \quad \text{for } j = 1, \dots, M \quad (2)$$

where $\tilde{s}_j(n)$ are the recovered source signals. We can rewrite equation (2) in the following matrix form,

$$\tilde{\mathbf{s}}(n) = \mathbf{W}\mathbf{x}(n) \quad (3)$$

where

$$\begin{aligned} \tilde{\mathbf{s}}(n) &= [\tilde{s}_1(n), \dots, \tilde{s}_M(n)]^T \\ \mathbf{x}(n) &= [x_1(n), \dots, x_1(n-L+1), \dots, x_N(n-L+1)]^T \\ \mathbf{W} &= \begin{bmatrix} w_{11}(0) & \dots & w_{11}(L-1) & \dots & w_{1N}(L-1) \\ w_{21}(0) & \dots & w_{21}(L-1) & \dots & w_{2N}(L-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{M1}(0) & \dots & w_{M1}(L-1) & \dots & w_{MN}(L-1) \end{bmatrix} \end{aligned} \quad (4)$$

2.2. Identifiability

Using the z-transform, the design of an equalizer $\mathbf{W}(z)$ that recovers the original source signals only from the observations $\mathbf{x}(n)$ can be formulated as follows:

$$\tilde{\mathbf{s}}(n) = \mathbf{W}(z)[\mathbf{H}(z)\mathbf{s}(n) + \mathbf{n}(n)] \quad (5)$$

$$\tilde{\mathbf{s}}(n) = \mathbf{W}(z)\mathbf{H}(z)\mathbf{s}(n) + \mathbf{W}(z)\mathbf{n}(n) \quad (6)$$

Let us write,

$$\mathbf{G}(z) = \mathbf{W}(z)\mathbf{H}(z) \quad (7)$$

As shown in [3], the LTI system represented by its transfer function matrix $\mathbf{G}(z)$ is said to be transparent or *decoupled* if $\mathbf{G}(z)$ has a single nonzero monomial entry in each row and each column.

In other words, an LTI system is transparent if and only if $\mathbf{G}(z)$ can be decomposed into:

$$\mathbf{G}(z) = \mathbf{\Delta}(z)\mathbf{D}\mathbf{P} \quad (8)$$

where $\mathbf{\Delta}(z)$ is a diagonal matrix with diagonal entries:

$$\lambda_{ii} = z^{l_i} \quad (9)$$

where l_i is a non-negative integer, \mathbf{D} is a constant diagonal matrix, and \mathbf{P} a permutation matrix.

Then, a channel system $\mathbf{H}(z)$ is said to be *deconvolvable* if there exists an equalizer $\mathbf{W}(z)$ so that the composite system $\mathbf{G}(z)$ is transparent.

Furthermore, a necessary and sufficient condition for $\mathbf{H}(z)$ to be *deconvolvable* is that the greatest common divisor of all the minors of order M in $\mathbf{H}(z)$ is nonzero monomial (see [4] for details).

3. THE PROPOSED ALGORITHM (SOMOD)

The problem of blind multi channel deconvolution is to find \mathbf{W} an $[M \times NL]$ matrix such that $\tilde{\mathbf{s}}(n) = \mathbf{s}(n)$. We can define the source correlation matrices at time lag k as:

$$\mathbf{R}_s(k) = E[\mathbf{s}(n)\mathbf{s}(n-k)^*] \quad (10)$$

Where $*$ denotes the transpose conjugate of a vector. Under relation (3), the above equation can be put in the following form:

$$\mathbf{R}_s(k) = \mathbf{W}\mathbf{R}_x(k)\mathbf{W}^H \quad (11)$$

where,

$$\mathbf{R}_x(k) = E[\mathbf{x}(n)\mathbf{x}(n-k)^*] \quad (12)$$

are the data correlation matrices at time lag k .

Let us consider the following decomposition of \mathbf{W} ,

$$\mathbf{W} = \mathbf{U}^H\mathbf{B} \quad (13)$$

where \mathbf{U} is an $[M \times M]$ unitary matrix, H denotes the transpose conjugate of a matrix and \mathbf{B} is an $[M \times NL]$ matrix.

Substituting (13) into (11) and assuming, without loss of the generality, that the source signals are of unit variance¹, one can write

$$\mathbf{R}_s(0) = \mathbf{B}\mathbf{R}_x(0)\mathbf{B}^H = \mathbf{I} \quad (14)$$

According to equation (14), \mathbf{B} is nothing than a whitening matrix that can be obtained from an eigen decomposition of $\mathbf{R}_x(0)$.

For time lag k , $k \neq 0$, we have

$$\mathbf{R}_s(k) = \mathbf{U}^H\mathbf{B}\mathbf{R}_x(k)\mathbf{B}^H\mathbf{U} = \mathbf{\Delta}_k \quad (15)$$

where $\mathbf{\Delta}_k$ is a diagonal matrix according to assumption **A1**). By denoting

$$\underline{\mathbf{R}}_x(k) = \mathbf{B}\mathbf{R}_x(k)\mathbf{B}^H \quad (16)$$

where $\{\underline{\mathbf{R}}_x(k), k = 1, \dots, K\}$ is a set of K whitened data correlation matrices at different time lags, we obtain the following key relation

$$\mathbf{\Delta}_k = \mathbf{U}^H\underline{\mathbf{R}}_x(k)\mathbf{U} \quad (17)$$

Since the matrix \mathbf{U} is unitary and $\mathbf{\Delta}_k$ is diagonal, expression (17) shows that any whitened data correlation matrix is diagonal in the basis of the columns of the matrix \mathbf{U} (the eigenvalues of $\underline{\mathbf{R}}_x(k)$ being the diagonal entries of $\mathbf{\Delta}_k$).

¹Because of the well known ambiguity of blind identification.

If, for the time lag k , the diagonal elements of $\mathbf{\Delta}_k$ are all distinct, the missing unitary matrix \mathbf{U} may be ‘uniquely’ (i.e. up to permutation and phase shifts) retrieved by computing the eigen decomposition of $\underline{R}_x(k)$.

Indeterminacy occurs in the case of degenerate eigenvalues. It does not seem possible to *a priori* determine some value for the delay k such that the diagonal entries of $\mathbf{\Delta}_k$ are all distinct. Of course, if the source signals have different spectral shapes, such a kind of eigenvalue degeneracy is unlikely, but it is to be expected that when some eigenvalues of $\underline{R}_x(k)$ comes close to degeneracy, the robustness of determining \mathbf{U} from eigen decomposition of a single whitened data correlation matrix is seriously impaired.

The situation is more favorable when considering simultaneous diagonalization of a set $\{\underline{R}_x(k)\}$ of K whitened data correlation matrices. This set is simultaneously diagonalizable by the unitary matrix \mathbf{U} as in (17).

The matrix \mathbf{U} is unique (to a permutation matrix and phase factors) if, and only if, for any pair (i, j) of sources, there exists a time lag k such that $E[s_i(n)s_i(n-k)^*] \neq E[s_j(n)s_j(n-k)^*]$. Of course, the simultaneous diagonalization holds only for the exact statistics; empirical statistics may only be *approximately* simultaneously diagonalized under the same unitary transform. This calls for the definition of the *approximate* simultaneous diagonalization.

Joint diagonalization: The joint diagonalization (JD) [1] can be explained by first noting that the problem of the diagonalization of a single $n \times n$ normal matrix \mathbf{M} is equivalent to the maximization of the criterion [8]

$$C(\mathbf{M}, \mathbf{V}) \stackrel{\text{def}}{=} \sum_i |\mathbf{v}_i^* \mathbf{M} \mathbf{v}_i|^2 \quad (18)$$

over the set of unitary matrices $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$. Hence, the joint diagonalization of a set $\{\mathbf{M}_k | k = 1..K\}$ of K arbitrary $n \times n$ matrices is defined as the maximization of the following JD criterion:

$$C(\mathbf{V}) \stackrel{\text{def}}{=} \sum_k C(\mathbf{M}_k, \mathbf{V}) = \sum_{k,i} |\mathbf{v}_i^* \mathbf{M}_k \mathbf{v}_i|^2 \quad (19)$$

under the same unitary constraint. An efficient joint approximate diagonalization algorithm exists in [1] and it is a generalization of the Jacobi technique [8] for the exact diagonalization of a single normal matrix.

Finally, the unitary matrix \mathbf{U} in (13) is obtained by the joint diagonalization of the set $\{\underline{R}_x(k)\}$ which

corresponds to the maximization:

$$\hat{\mathbf{U}} = \underset{\mathbf{U}}{\text{Argmax}} \sum_{k=0}^{L-1} \sum_{i=1}^M |\mathbf{u}_i^* \underline{R}_x(k) \mathbf{u}_i|^2 \quad (20)$$

with $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_M]$.

Implementation issue

The eigen decomposition of $\underline{R}_x(0)$ for the determination of matrix \mathbf{B} will provide us with $M \times L$ eigen vectors that span the extended subspace. Only M of them span the original source subspace. Hence, it is impossible, without any knowledge of the original sources to select the M eigen vectors among the $M \times L$ obtained. We propose to use all the $M \times L$ vectors in order to determine \mathbf{B} which will change the dimension of \mathbf{B} from $[M \times NL]$ to $[ML \times NL]$. Then, we maximize the JD criterion using a unitary matrix \mathbf{U}' of dimension $ML \times ML$:

$$\hat{\mathbf{U}}' = \underset{\mathbf{U}'}{\text{Argmax}} \sum_{k=0}^{L-1} \sum_{i=1}^{ML} |\mathbf{u}'_i{}^* \underline{R}_x(k) \mathbf{u}'_i|^2 \quad (21)$$

with $\mathbf{U}' = [\mathbf{u}'_1, \dots, \mathbf{u}'_{ML}]$.

One can easily show that the maximization (21) leads to the maximization (20) and among the $M \times L$ eigen vectors of \mathbf{U}' , M of them correspond to the desired sources. The desired M eigen vectors are selected from the $M \times L$ ones by choosing those which lead to the smallest correlation coefficients of the recovered signals. The proposed approach has shown to be robust with respect to an overestimated source number.

4. SIMULATIONS

Example 1

we consider an array of 2 sensors receiving signals from 2 sources in the presence of white Gaussian noise. The channel length is $L = 4$. The signal to noise ratio (SNR) is set at 40 dB. Figure 1 shows the temporal representation of the original sources, their convolutive mixtures and the recovered signals by the SOMOD algorithm. For the same experiment, Figure 2 shows the Time Frequency representation of the original sources, their convolutive mixtures and the recovered signals by the SOMOD algorithm. The kernel used for the computation of the TFDs is the Choi-Williams kernel [9]. This example is an illustration of the success of the proposed algorithm in separating two sources.

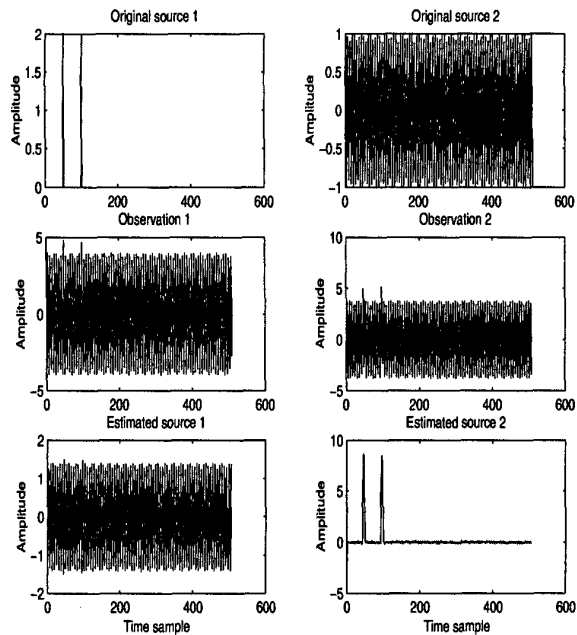


Figure 1: Separation example (Time representation): SNR=40dB.

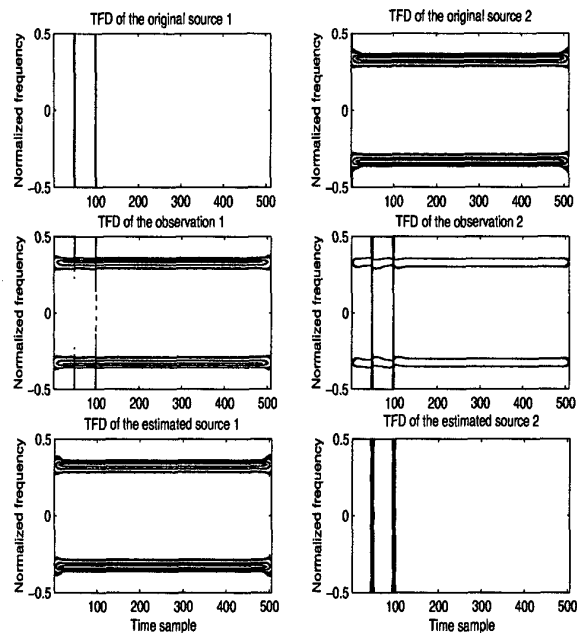


Figure 2: Separation example (Time Frequency representation): SNR=40dB.

Example 2

We present here a simulation to illustrate the effectiveness of our algorithm in deconvolving speech signals. The parameter settings are :

- $M = 2, N = 2$ and $L = 3$.
- The two speech signals are sampled at 16kHz.
- Signal to Noise Ratio (SNR) = 10 dB.
- The transfer function matrix of the simulated multi channel is given by,

$$\mathbf{H}(z) = \begin{bmatrix} -0.40 + 0.82z^{-1} + 1.29z^{-2} & 1.19 - 0.02z^{-1} - 1.60z^{-2} \\ 0.69 + 0.71z^{-1} + 0.67z^{-2} & -1.20 - 0.16z^{-1} + 0.26z^{-2} \end{bmatrix}$$

Figure 2 shows the original speech signals, their convolutive mixtures and the recovered speech signals by the SOMOD algorithm.

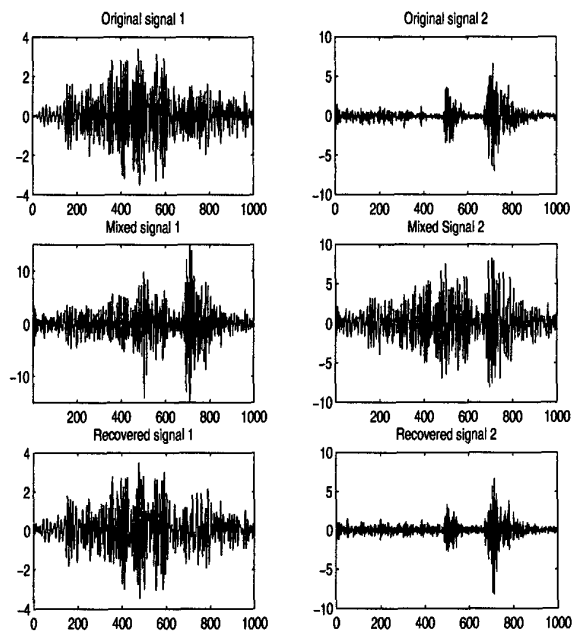


Figure 3: Speech signal separation : SNR=10dB.

5. CONCLUSION

In this contribution, we considered the blind deconvolution of MIMO FIR systems driven by mutually decorrelated source signals. We proposed a solution based on the joint diagonalization of spatio-temporal correlation matrices. This technique has been proposed previously

in the case of instantaneous mixtures [1]. An extension of this method to convolutive mixtures has been presented in this paper. It showed to be robust with respect to additive noise. Moreover, it is able to deal with an overestimated source number; since the method provides an $M \times L$ recovered source subspace instead of the original M source subspace. A source selection criterion has been defined to select the M recovered sources among the $M \times L$ obtained. This method is well suited when applied to the deconvolution of speech signals, which is of great importance in practical applications [7].

6. REFERENCES

- [1] A. Belouchrani and K. Abed Meraim and J.-F. Cardoso and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Trans. on SP*, vol. 45, pp. 434-444, Feb. 1997.
- [2] G. B. Giannakis, Y. Inouye, and J. M. Mendel, "Cumulant-based identification of multichannel moving-average processes," *IEEE Trans. on Automat. Contr.*, vol. 34, pp. 783-787, Jul. 1989.
- [3] Y. Inouye and Ruey-wen Liu, "Criteria for direct blind deconvolution of MIMO FIR systems driven by white source signals," in *Proc. ICAS-SP'99*, Phoenix, Arizona, May 1999.
- [4] J. K. Massey and M. K. Sain, "Inverse of linear sequential circuits," *IEEE Trans. Computers*, pp. 330-337, 1968.
- [5] A. Cichocki, S. Amari and J. Cao, "Blind separation of delayed and convolved signals with self-adaptive learning rate," in *In Proc. Nolta'96*, 1996.
- [6] Kari Torkkola, "Blind separation of convolved sources based on information maximization," in *In IEEE Workshop on Neural Networks for Signal Processing, Kyoto, Japan*, Sept. 1996.
- [7] F. Ehlers and H. G. Schuster, "Blind separation of convolutive mixtures and an application in automatic speech recognition in a noisy environment," *IEEE Trans. on SP*, vol. 45, pp. 2608-2612, Oct. 1997.
- [8] G.H. Golub and C.F. Van Loan, *Matrix computations*. The Johns Hopkins University Press, 1989.
- [9] L. Cohen, *Time-frequency analysis*. Prentice Hall, 1995.

BLIND SEPARATION OF NON STATIONARY SOURCES USING JOINT BLOCK DIAGONALIZATION

Hicham Bousbia-Salah*, Adel Belouchrani*, and Karim Abed-Meraim**

* Electrical Engineering Department,
Ecole Nationale Polytechnique,
P.O. Box 182 El-Harrach, Algiers 16200, Algeria.
E-mail: hichams@yahoo.fr, belouchrani@hotmail.com

** Signal and Image processing Department,
Télécom Paris 46 rue Barrault, 75013 Paris, France.
E-mail: abed@tsi.enst.fr

ABSTRACT

Recovering independent source signals from their convolutive mixtures without any a priori knowledge on their structure represents a great challenge in signal processing. In this paper, we present an efficient solution that is based on the joint block-diagonalization of positive spatio-temporal covariance matrices. In the case of instantaneous mixtures, robust solutions have been proposed previously. Taking advantage of possible non-stationarity of the sources, this new technique uses only second order statistics. The new approach has been successfully applied to the separation of speech signals.

1. INTRODUCTION

If we consider a set of received signals that are convolutive mixtures of independent source signals, the objective of blind separation is to recover the source signals from the set of received signals without any knowledge of the linear mixtures or the Linear Time Invariant (LTI) systems. For instantaneous mixtures, a Second Order Blind Identification (SOBI) algorithm has been presented [1] and showed to be very robust for temporally correlated sources. An analog technique based on Block Gaussian likelihood, presented by Pham [2] uses a joint diagonalization of positive correlation matrices of the received data. An extension of the SOBI technique to the convolutive mixtures has been considered in [3, 4].

When dealing with convolutive mixtures, classical blind separation can be achieved in two ways. One way is to first identify the channel system from the output mixtures and then to design an equalizer accordingly [5]. The other way consists of directly designing an equalizer from the output mixtures. The latter bypasses the problem of blind system identification and is computation less expensive. Herein,

we consider the separation of the source signals up to a scalar filter instead of a full deconvolution. For this purpose, we propose to extend the Block Gaussian likelihood technique [2] to the convolutive mixture case. It is based on the joint block-diagonalization of positive spatio-temporal covariance matrices of the received data. In this contribution, the measure of block-diagonality is directly related to the likelihood objective function and is optimized without any orthogonality constraint which bypasses any prior whitening of the observations. The proposed method has been successfully applied to the separation of speech signals up to a scalar filter. In the next sections, we will present the data model and describe the proposed algorithm. And finally, some simulation results are provided in section 5.

2. DATA MODEL

For simplicity, we shall restrict ourselves to the simplest discrete time multiple input multiple output (MIMO) linear time invariant model given by,

$$x_i(n) = \sum_{j=1}^M \sum_{l=0}^{L-1} h_{ij}(l) s_j(n-l), \quad \text{for } i = 1, \dots, N. \quad (1)$$

where $s_j(n)$, $j = 1, \dots, M$ are the M source signals (model inputs), $x_i(n)$, $i = 1, \dots, N$, are the N sensor signals (model outputs), h_{ij} is the transfer function with an overall duration L between the j -th source and the i -th sensor.

The assumptions made about the data model are as follows:

A1) The source signals $s_j(n)$, $j = 1, \dots, M$, are mutually decorrelated.

A2) Each source signal is non stationary.

A3) The channel matrix \tilde{H} defined in (3) is full column rank.

The purpose of blind source separation is to recover the

source signals based only on the sensor signals. In some applications as in speech processing, the separation of the sources up to a scalar filter is sufficient. In this paper, we consider the problem of the source separation up to a scalar filter instead of the full MIMO deconvolution procedure.

We can rewrite equation (1) in the following matrix form,

$$\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n) \quad (2)$$

where

$$\begin{aligned} \mathbf{s}(n) &= [s_1(n), \dots, s_1(n - (L + L' - 1) + 1), \\ &\quad \dots, s_M(n), \dots, s_M(n - (L + L' - 1) + 1)]^T \\ \mathbf{x}(n) &= [x_1(n), \dots, x_1(n - L' + 1), \\ &\quad \dots, x_N(n), \dots, x_N(n - L' + 1)]^T \end{aligned}$$

Subscript T denotes the transpose of a vector, and:

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \dots & \tilde{\mathbf{H}}_{1M} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{H}}_{N1} & \dots & \tilde{\mathbf{H}}_{NM} \end{bmatrix} \quad (3)$$

with

$$\tilde{\mathbf{H}}_{ij} = \begin{bmatrix} h_{ij}(0) & \dots & h_{ij}(L-1) & \dots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \dots & h_{ij}(0) & \dots & h_{ij}(L-1) \end{bmatrix}$$

Note that $\tilde{\mathbf{H}}$ is a $[NL' \times M(L + L' - 1)]$ matrix and $\tilde{\mathbf{H}}_{ij}$ are $[L' \times (L + L' - 1)]$ matrices. L' is chosen such that $NL' \geq M(L + L' - 1)$.

We assume that $\tilde{\mathbf{H}}$ is a square matrix, i.e., $NL' = M(L + L' - 1)$, if not, it can be made square by projecting the sensors data $\mathbf{x}(n)$ into the sources subspace.

3. THE PROPOSED ALGORITHM

In this section, we extend the Block Gaussian likelihood technique [2] to the convolutive mixture case. It is based on the joint block-diagonalization of positive spatio-temporal covariance matrices of the received data.

The interval $[0, T]$ may be divided into K consecutive sub-intervals T_1, \dots, T_K such that the approximate covariance matrix of the received data in the sub-interval T_k is given by:

$$\tilde{\mathbf{R}}_x(k) = \frac{1}{n_{T_k}} \sum_{n \in T_k} \mathbf{x}(n)\mathbf{x}(n)^* \quad (4)$$

where n_{T_k} is the number of elements (samples) in the sub-interval T_k and subscript $*$ denotes the conjugate transpose of a vector. Implicitly, we assume approximate local stationarity in each data sub-block.

Under the linear model (2), the above equation can be put in the following form:

$$\tilde{\mathbf{R}}_x(k) = \tilde{\mathbf{H}}\tilde{\mathbf{R}}_s(k)\tilde{\mathbf{H}}^H \quad (5)$$

where, $\tilde{\mathbf{R}}_s(k)$ are the approximate covariance matrices of the source signals and subscript H denotes the conjugate transpose of a matrix. Taking advantage of the mutual decorrelation of the source signals, $\tilde{\mathbf{R}}_s(k)$ is approximately block diagonal, with M diagonal blocks of dimension $(L + L' - 1) \times (L + L' - 1)$ each, i.e.

$$\tilde{\mathbf{R}}_s(k) \approx \begin{bmatrix} \tilde{\mathbf{R}}_{s_1}(k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}_{s_2}(k) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{R}}_{s_M}(k) \end{bmatrix} \quad (6)$$

$\tilde{\mathbf{R}}_{s_1}(k), \tilde{\mathbf{R}}_{s_2}(k), \dots, \tilde{\mathbf{R}}_{s_M}(k)$ are the 'local' covariance matrices of the M sources, k being the data sub-block index.

Equations (5) and (6) just mean that any data covariance matrix is block-diagonal in the basis of the column vectors of matrix $\tilde{\mathbf{H}}$, which can be retrieved by computing the joint block-diagonalization of a set of K covariance matrices $\tilde{\mathbf{R}}_x(k), k = 1, \dots, K$.

4. A JOINT BLOCK-DIAGONALIZATION CRITERION

In this section, we derive a joint block-diagonalization criterion inspired from the joint diagonalization criterion of [2]. Using the Kullback-Leiber divergence between two zero mean K -variate normal densities with covariance matrices \mathbf{R}_a and \mathbf{R}_b respectively, the deviation between \mathbf{R}_a and \mathbf{R}_b is defined as:

$$D(\mathbf{R}_a, \mathbf{R}_b) > 0 \quad (7)$$

with equality if and only if $\mathbf{R}_a = \mathbf{R}_b$ and thus is a legitimate measure of deviation between positive definite matrices.

Therefore, a measure of deviation from block-diagonalization could be derived from:

$$D(\tilde{\mathbf{H}}^{-1}\tilde{\mathbf{R}}_x(k)\tilde{\mathbf{H}}^{-H}, \tilde{\mathbf{R}}_s(k)) \quad (8)$$

Following the same steps as in [2], the above measure of deviation is equivalent to,

$$\sum_{k=1}^K [\log \det(\text{bdiag}(\mathbf{M}_k)) - \log \det(\mathbf{M}_k)] \quad (9)$$

with

$$\mathbf{M}_k = \mathbf{B}\tilde{\mathbf{R}}_x(k)\mathbf{B}^H \quad (10)$$

over the set of matrices \mathbf{B} , where $\text{bdiag}(\mathbf{M}_k)$ denotes the block-diagonal matrix with the same diagonal blocks of size

$(L + L' - 1) \times (L + L' - 1)$ as \mathbf{M}_k .

From the generalized Hadamard inequality [6] and for Hermitian positive definite matrices:

$$\det(\mathbf{M}_k) < \det(\text{bdiag}(\mathbf{M}_k)) \quad (11)$$

with equality if and only if \mathbf{M}_k is block-diagonal.

It follows that criterion (9) is a measure of the global deviation of the matrices from block-diagonal structure. Hence, minimization of (9) leads to,

$$\mathbf{B} \approx \mathbf{D}\tilde{\mathbf{H}}^{-1} \quad (12)$$

where the matrix \mathbf{D} is an arbitrary block-diagonal matrix coming from the inherent indeterminacy of the joint block-diagonalization problem.

Once the matrix \mathbf{B} is determined, the recovered signal are obtained up to a filter by,

$$\hat{\mathbf{s}}(n) = \mathbf{B}\mathbf{x}(n) \quad (13)$$

Accordingly, the recovered signals will verify,

$$\hat{\mathbf{s}}(n) = \mathbf{D}\mathbf{s}(n) \quad (14)$$

5. SIMULATIONS

We present here a simulation to illustrate the effectiveness of our algorithm in separating speech signals. The parameter settings are :

1. $M = 3$, $N = 2$, $L = 3$ and $L' = 4$.
2. The two speech signals are sampled at 8kHz.
3. The transfer function matrix of the simulated multi channel is given by,

$$\mathbf{H}(z) = \begin{bmatrix} 1 + 0.5z^{-1} + 0.7z^{-2} & 0.1z^{-1} + 0.85z^{-2} \\ 0.8 + 0.7z^{-1} + 0.4z^{-2} & 1 + 0.9z^{-1} \\ 1 + 0.5z^{-1} + 0.3z^{-2} & 0.7 + 0.85z^{-1} + 0.1z^{-2} \end{bmatrix}$$

Figures 1, 2 and 3 show a sample run of the proposed algorithm. Note that only two speech signals among twelve recovered ones are displayed. These two signals lead to the smallest correlation coefficients.

6. CONCLUSION

In this contribution, we considered the problem of the blind separation of convolutive mixtures of non-stationary source signals. We proposed a solution based on the joint block-diagonalization of positive spatio-temporal covariance matrices. This technique uses only second order statistics and unlike [3, 4] has no orthogonality constraint which bypasses any prior whitening of the data. This method is well suited when applied to the deconvolution of speech signals, which is of great importance in practical applications [7].

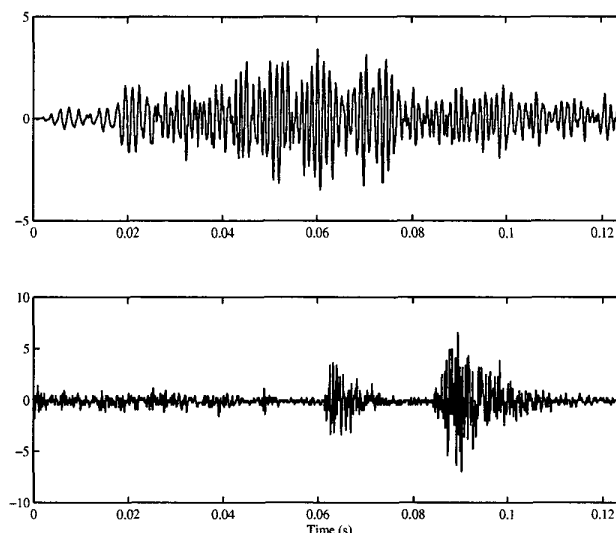


Fig. 1. Original speech signals.

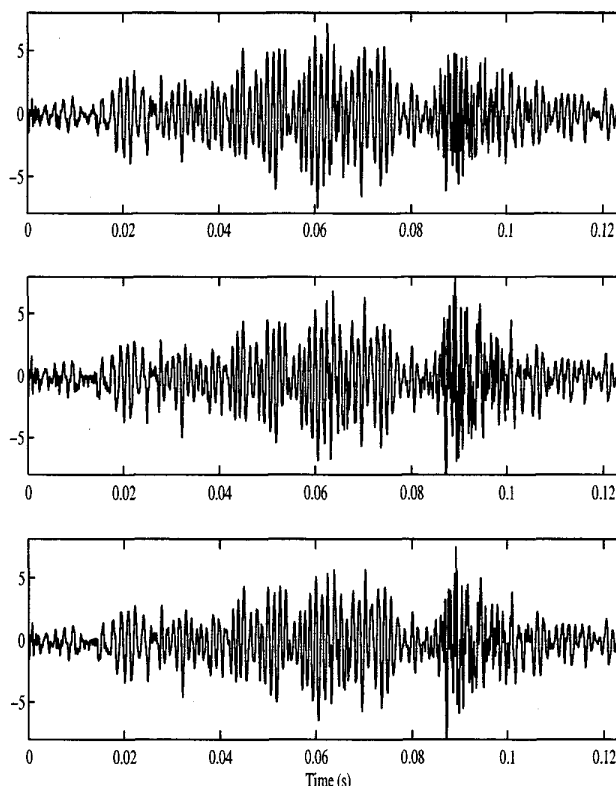


Fig. 2. Mixed speech signals.

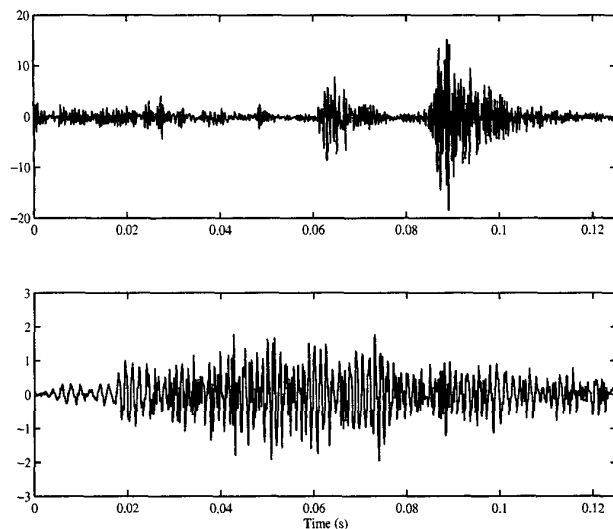


Fig. 3. Two of the twelve recovered speech signals.

7. REFERENCES

- [1] A. Belouchrani and K. Abed Meraim and J-F Cardoso and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Trans. on SP*, vol. 45, pp. 434–444, Feb. 1997.
- [2] D. Pham and J. F. Cardoso, "Blind separation of instantaneous mixtures of non stationary sources," *IEEE Trans. on SP*, vol. 45, pp. 2608–2612, Oct. 1997.
- [3] H. Bousbia-Salah and A. Belouchrani, "A Second Order Multi Output Deconvolution (SOMOD) Technique," *In the tenth IEEE Workshop on Statistical Signal and Array Processing*, pp.306-310, Poconos, Pennsylvania, USA, Aug. 2000.
- [4] H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim, "Blind source separation of convolutive mixtures using joint block diagonalization," *Proc. ISSPA'2001*, Kuala-Lumpur, Malaysia, Aug. 2001.
- [5] G. B. Giannakis, Y. Inouye, and J. M. Mendel, "Cumulant-based identification of multichannel moving-average processes," *IEEE Trans. on Automat. Contr.*, vol. 34, pp. 783–787, Jul. 1989.
- [6] B. Flury and B.E. Neuenschwander, "Principal component models for patterned covariance matrices, with applications to canonical correlation analysis of several sets of variables," *Descriptive multivariate analysis*, Oxford University Press, 1994.
- [7] F. Ehlers and H. G. Schuster, "Blind separation of convolutive mixtures and an application in automatic speech recognition in a noisy environment," *IEEE Trans. on SP*, vol. 45, pp. 2608–2612, Oct. 1997.

BLIND SEPARATION OF CONVOLUTIVE MIXTURES USING JOINT BLOCK DIAGONALIZATION

Hicham Bousbia-Salah *, Adel Belouchrani *, and Karim Abed-Meraim **

(*) Electrical Engineering Department, Ecole Nationale Polytechnique,
10 Avenue Hassan badi, B.P:182 El-Harrach, Algiers 16200, Algeria.

(**) École Nationale Supérieure des Télécommunications,
Depart. TSI, 46 rue Barrault, 75634 Paris Cedex 13, France
e-mail: hichams@yahoo.fr, belouchrani@hotmail.com, abed@tsi.enst.fr

ABSTRACT

In this paper, we present an efficient and new solution to the recovery of independent source signals from their convolutive mixtures without any a priori knowledge on the mixtures. In the case of instantaneous mixtures, a robust solution referred to as Second Order Blind Identification (SOBI) has been proposed previously. Herein, we extend this technique to the convolutive mixture case and we propose an algorithm that performs a joint block diagonalization of spatio-temporal correlation matrices instead of joint diagonalization as in SOBI. The proposed method has been successfully applied to the separation of speech signals.

1. INTRODUCTION

If we consider a set of received signals that are linear convolutive mixtures of decorrelated source signals, the objective of blind separation is to recover the sources from the received signals without any knowledge on the actual linear mixtures.

Many contributions exist in the literature on the blind Multi-Input Multi-Output (MIMO) deconvolution [1]. However, in some applications as in speech processing, the separation of the sources up to a filter is sufficient. In this paper, we consider the problem of the source separation up to a filter instead of the full MIMO deconvolution procedure. Note that if needed a Single Input Multi Output (SIMO) deconvolution/equalization [2] can be applied to the separated sources. For instantaneous mixtures, a Second Order Blind Identification (SOBI) algorithm has been presented [3] and showed to be very effective for the separation of temporally correlated sources. It is based on the joint diagonalization of spatio-temporal correlation matrices. In this contribution, we propose to extend SOBI method to the convolutive mixture case

as in [4]: the proposed algorithm performs a joint block diagonalization of these matrices instead of a joint diagonalization since the diagonal elements in this case are block matrices. The proposed method has been successfully applied to the separation of speech signals.

The paper is organized as follows: In Section 2, we present the data model together with the relevant assumptions. The proposed algorithm is described in Section 3. Simulation results are provided in Section 4 to illustrate the effectiveness of our approach.

2. DATA MODEL

Consider a discrete time MIMO linear time invariant model given by,

$$x_i(n) = \sum_{j=1}^M \sum_{l=0}^{L-1} h_{ij}(l) s_j(n-l) \quad \text{for } i = 1, \dots, N \quad (1)$$

where $s_j(n)$, $j = 1, \dots, M$, are the M source signals (model inputs), $x_i(n)$, $i = 1, \dots, N$, are the N sensor signals (model outputs), h_{ij} is the transfer function between the j -th source and the i -th sensor with an overall extent L .

The following assumptions are made:

A1) The source signals $s_j(n)$, $j = 1, \dots, M$, are mutually uncorrelated and each source signal is temporally coherent.

A2) The channel matrix $\tilde{\mathbf{H}}$ defined in (3) is full column rank.

We can rewrite equation (1) in the following matrix form,

$$\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n) \quad (2)$$

where

$$\begin{aligned} \mathbf{s}(n) &= [s_1^T(n), \dots, s_M^T(n)]^T \\ \mathbf{s}_i(n) &= [s_i(n), \dots, s_i(n - (L + L'))]^T \end{aligned}$$

$$\mathbf{x}(n) = [x_1(n), \dots, x_1(n-L'+1), \dots, x_N(n), \dots, x_N(n-L'+1)]^T$$

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \dots & \tilde{\mathbf{H}}_{1M} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{H}}_{N1} & \dots & \tilde{\mathbf{H}}_{NM} \end{bmatrix} \quad (3)$$

with

$$\tilde{\mathbf{H}}_{ij} = \begin{bmatrix} h_{ij}(0) & \dots & h_{ij}(L-1) & \dots & 0 \\ & \ddots & & \ddots & \\ 0 & \dots & h_{ij}(0) & \dots & h_{ij}(L-1) \end{bmatrix}$$

Note that $\tilde{\mathbf{H}}$ is a $[NL' \times M(L+L')]$ matrix and $\tilde{\mathbf{H}}_{ij}$ are $[L' \times (L+L')]$ matrices. L' is chosen such that $NL' \geq M(L+L')$.

We define the data correlation matrix at time lag k as:

$$\mathbf{R}_x(k) = E[\mathbf{x}(n)\mathbf{x}(n-k)^*] \quad (4)$$

where \mathbf{x}^* denotes the conjugate transpose of vector \mathbf{x} . Under the linear model (2), the above equation can be put in the following form:

$$\mathbf{R}_x(k) = \tilde{\mathbf{H}}\mathbf{R}_s(k)\tilde{\mathbf{H}}^H \quad (5)$$

where $\tilde{\mathbf{H}}^H$ is the conjugate transpose of matrix $\tilde{\mathbf{H}}$ and,

$$\mathbf{R}_s(k) = E[\mathbf{s}(n)\mathbf{s}(n-k)^*] \quad (6)$$

is the source correlation matrix at time lag k . According to assumption **A1**, the $[M(L'+L) \times M(L'+L)]$ matrix $\mathbf{R}_s(k)$ is block diagonal.

Next, we present an algorithm for the recovery of the source signals up to a filter¹ that exploits the block diagonal structure of the source correlation matrices.

3. THE PROPOSED ALGORITHM

The first step of our procedure consists in whitening the $[NL' \times 1]$ data vector $\mathbf{x}(n)$. This is achieved by applying to $\mathbf{x}(n)$ a whitening matrix \mathbf{W} , i.e. a $[M(L'+L) \times NL']$ matrix verifying:

$$E[\mathbf{W}\mathbf{x}(n)\mathbf{x}(n)^*\mathbf{W}^H] = \mathbf{W}\mathbf{R}_x(0)\mathbf{W}^H = (\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}})(\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}})^H = \mathbf{I} \quad (7)$$

Equation (7) shows that if \mathbf{W} is a whitening matrix, then

$$\mathbf{U} = \mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}} \quad (8)$$

is a $[M(L'+L) \times M(L'+L)]$ unitary matrix where $\mathbf{R}_s(0)^{\frac{1}{2}}$ (the Hermitian square root matrix of $\mathbf{R}_s(0)$)

¹We estimate $[\alpha_i(z)]s_i(n)$, $i = 1, \dots, M$ where $\alpha_i(z)$ are FIR scalar filters.

is block diagonal. The whitening matrix \mathbf{W} can be determined from the eigen decomposition of the data correlation $\mathbf{R}_x(0)$ as in [3].

Consider now the spatially whitened correlation matrices $\underline{\mathbf{R}}_x(k)$ defined as

$$\forall k \neq 0, \underline{\mathbf{R}}_x(k) = \mathbf{W}\mathbf{R}_x(k)\mathbf{W}^H. \quad (9)$$

From (5) and (8), we obtain the key relation:

$$\forall k \neq 0 \underline{\mathbf{R}}_x(k) = \mathbf{U}\mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{R}_s(k)\mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{U}^H = \mathbf{U}\mathbf{D}(k)\mathbf{U}^H \quad (10)$$

where we have set $\mathbf{D}(k) = \mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{R}_s(k)\mathbf{R}_s(0)^{-\frac{1}{2}}$ ($\mathbf{R}_s(0)^{-\frac{1}{2}}$ being the inverse square root matrix of $\mathbf{R}_s(0)$).

Since matrix \mathbf{U} is unitary and \mathbf{D}_k is block diagonal, the latter just means that any whitened data correlation matrix is block diagonal in the basis of the columns of matrix \mathbf{U} . The unitary matrix can be retrieved by computing the block diagonalization of some matrix $\underline{\mathbf{R}}_x(k)$. But for a single correlation matrix, indeterminacy can occur. To reduce this indeterminacy and increase the robustness of determining \mathbf{U} , we consider the joint block diagonalization of a set $\{\underline{\mathbf{R}}_x(k); k = 1, \dots, K\}$ of K whitened data correlation matrices.

This joint block-diagonalization (JBD) is achieved by minimizing the Frobenius norm of the off-diagonal blocks of $\underline{\mathbf{R}}_x(k)$, $k = 1, \dots, K$. Since the norm is unchanged under unitary transform, this is equivalent to the maximization of the following criterion (representing the sum of the square norms of the block diagonal matrices),

$$C(\mathbf{U}) \stackrel{\text{def}}{=} \sum_{k=1}^K \sum_{m=1}^M \sum_{i,j=(L'+L)(m-1)+1}^{(L'+L)m} |\mathbf{u}_i^* \underline{\mathbf{R}}(k) \mathbf{u}_j|^2 \quad (11)$$

over the set $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{M(L'+L)}]$ of unitary matrices.

Note that an efficient Jacobi-like algorithm for joint block diagonalization exists in [5, 6]. It is based on the maximization of criterion (11) through successive Givens rotations (see [5, 6] for more details).

Note also that matrix \mathbf{U} is estimated only up to a random block diagonal unitary matrix [6].

Once \mathbf{U} is determined, the recovered signal are obtained up to a filter by

$$\hat{\mathbf{s}}(n) = \mathbf{U}^H \mathbf{W}\mathbf{x}(n) \quad (12)$$

According to (2) and (8), the recovered signals verify,

$$\mathbf{s}(n) = \begin{bmatrix} \mathbf{D}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{D}_M \end{bmatrix} \begin{bmatrix} \mathbf{s}_1(n) \\ \vdots \\ \mathbf{s}_M(n) \end{bmatrix} \quad (13)$$

where \mathbf{D}_i are $[(L+L') \times (L+L')]$ matrices.

Remarks

The following remarks are in order:

1. In practice, only M signals among the $M(L' + L)$ recovered ones are selected. This is done by choosing the signals which lead to the smallest correlation coefficients. Note that this information is a byproduct of the joint block diagonalization procedure and hence this selection needs no additional computations.
2. For simplicity, all the derivations have been presented in the noiseless case. However, we can deal in practice with spatially white noise as shown in [7]. The noise power can be estimated as the least eigenvalue of the zero lag correlation matrix after which the noise correlation matrices can be subtracted from those of the data.
3. In practice, the spatio-temporal correlation matrices $\mathbf{R}_x(k)$ are replaced by their time-averaged estimates $\hat{\mathbf{R}}_x(k)$.
4. The i th block of recovered signal $\hat{s}_i(n)$ can be rewritten as

$$\hat{s}_i(n) = [\mathbf{d}_i(z)]s_i(n)$$

where $\mathbf{d}_i(z) = \sum_{j=1}^{L+L'} \mathbf{d}_{ij}z^{-(j-1)}$ (\mathbf{d}_{ij} being the j th column vector of \mathbf{D}_i) is a polynomial vector of length $L+L'$. In other words, $\hat{s}_i(n)$ represents the output of a SIMO system given by $\mathbf{d}_i(z)$ excited by the input sequence $s_i(n)$. Therefore, if needed, existing SIMO equalization techniques [2] can be used to retrieve $s_i(n)$ up to some delay and scalar constant (instead of a scalar filter).

4. SIMULATION RESULTS

In this section, we present a simulation example to illustrate the effectiveness of our algorithm in separating speech signals. The parameter settings are as follows:

- $M = 2$, $N = 3$, $L = 3$ and $L' = 4$.
- The two speech signals are sampled at 8kHz.
- The transfer function matrix of the simulated multi channel is given by,

$$\mathbf{H}(z) = \begin{bmatrix} 1 & 0.85z^{-1} + 0.1z^{-2} \\ 0.7z^{-1} + 0.4z^{-2} & 0.25 + z^{-1} \\ 1 + 0.5z^{-1} + 0.7z^{-2} & 0.1z^{-1} + 0.85z^{-2} \end{bmatrix}$$

Figure 1 shows the original speech signals, Figure 2 their convolutive mixtures and Figure 3 the recovered

speech signals by the proposed algorithm. Note that only two of the recovered speech signals among the twelve recovered ones are displayed. These two signals correspond to signals with smallest correlation coefficients. This example illustrates the effectiveness of the JBD algorithm in separating speech signals. To access the convergence rate of the JBD algorithm, we define the following convergence criterion:

$$C = \sum_{p \neq q} |\sin(\theta_{pq})|^2 \quad (14)$$

where θ_{qp} are the rotation angles of the Givens rotations at one sweep (see [5] for more details). Figure 4 shows how the convergence criterion decreases with respect to the iteration number. Fast convergence of the JBD algorithm is observed.

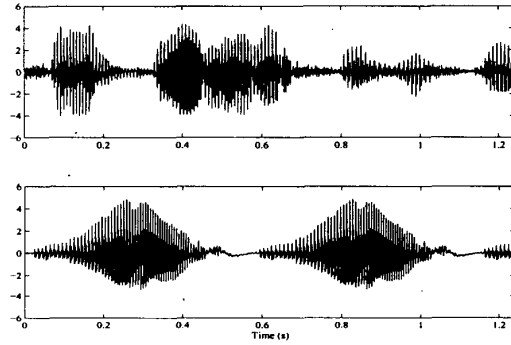


Figure 1: Original signals

5. CONCLUSION

In this contribution, the problem of the blind signal separation of convolutive mixtures is considered. A solution based on the joint block diagonalization of spatio-temporal correlation matrices is proposed. This technique is an extension of SOBI technique that has been proposed previously in the case of instantaneous mixtures [3]. The JBD algorithm shows to be very efficient and presents a fast convergence in the example considered in Section 4. Finally, this method is well suited to separate speech signals and more generally can be integrated as a first step of a blind deconvolution algorithm where SIMO equalization techniques can be used in a second step to perform a complete deconvolution of received signals.

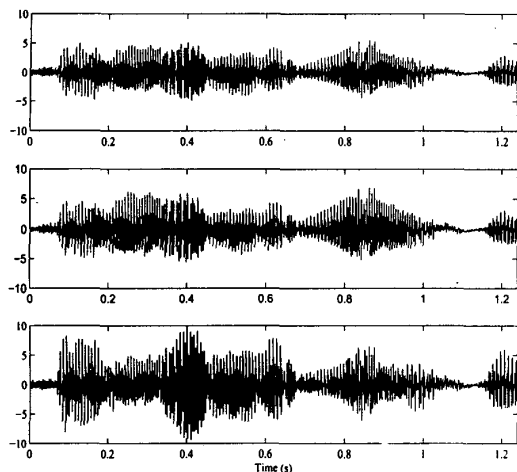


Figure 2: Mixed signals

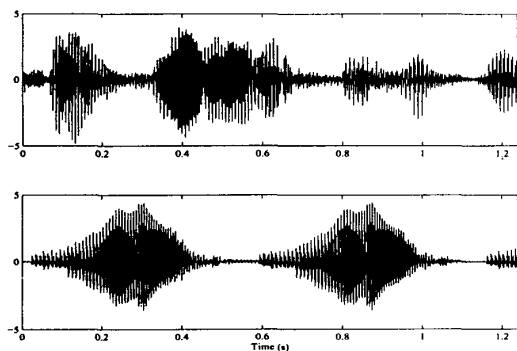


Figure 3: Two of the twelve recovered signals

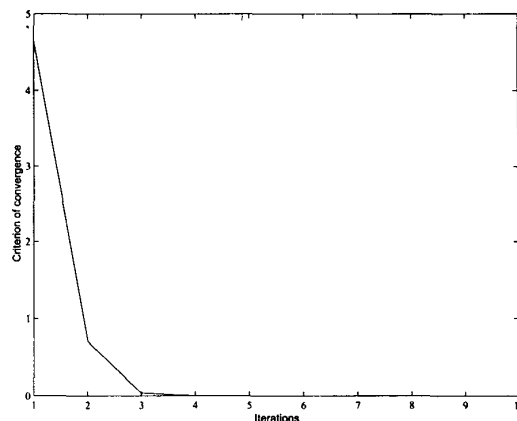


Figure 4: Convergence of the JBD algorithm.

6. REFERENCES

- [1] L. Tong and S. Perreau, "Multichannel blind channel estimation: From subspace to maximum likelihood methods," *Proc. IEEE. on SP*, vol. 86, pp. 1951-1968, Oct. 1998.
- [2] K. Abed-Meraim, W. Qiu, Y. Hua, "Blind System Identification", *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1310-1322, Aug. 1997.
- [3] A. Belouchrani and K. Abed Meraim and J.-F. Cardoso and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Trans. on SP*, vol. 45, pp. 434-444, Feb. 1997.
- [4] H. Bousbia-Salah and A. Belouchrani, "A Second Order Multi Output Deconvolution (SOMOD) Technique," *In the tenth IEEE Workshop on Statistical Signal and Array Processing*, pp. 306-310, Poconos, Pennsylvania, USA, Aug. 2000.
- [5] A. Belouchrani, K. Abed-Meraim, and Y. Hua, "Jacobi-like Algorithms for Joint Block Diagonalization: Application to Source Localization", *in Proc. ISPACS (Melbourne, Australia)*, November 1998.
- [6] A. Belouchrani and K. Abed-Meraim, "Joint Block Diagonalization For Array Signal Processing", *IEEE Tr. on Sig. Proc.* (submitted), Jan. 2001.
- [7] L. Tong, G. Xu and Kailath, "A new approach to blind identification and equalization of multipath channels," *in Proc. of the 25th Asilomar Conference, CA*, pp. 856-860, 1991.

mesa is first defined by wet selective etching of InGaAs ($H_3PO_4:H_2O_2 = 1:1:20$) and InP ($H_3PO_4:HCl = 3:1$). Ti ohmic contact metal of 500 Å is first sputter-deposited, followed by 300 Å Pd and 1000 Å Au. The TLM patterns are defined by the standard lift-off process with seven pairs of patterns with spacings equal to 2, 4, 6, 8, 10, 15 and 20 µm. Fig. 3 shows the TLM pattern pair resistance against pair spacing and the inset of Fig. 3 shows the current-voltage relations for the seven TLM pairs. From the extrapolation of resistance at zero spacing and the slope of Fig. 3 [4], the specific contact resistance is found to be equal to $3.5 \times 10^{-5} \Omega\text{cm}^2$ and the effective sheet resistance is determined to be 357.5 Ω. This is an excellent as-deposited ohmic contact and can be used for many device applications such as photodetector arrays and IR cameras. Because of the high surface hole concentration and the elimination of annealing for ohmic contact formation, this technology can be expected to result in improved product uniformity and yield. Further work is under way to investigate the dependence of the specific contact resistance on ohmic contact annealing temperature, aimed at achieving ultra-low specific contact resistance for high-speed device applications.

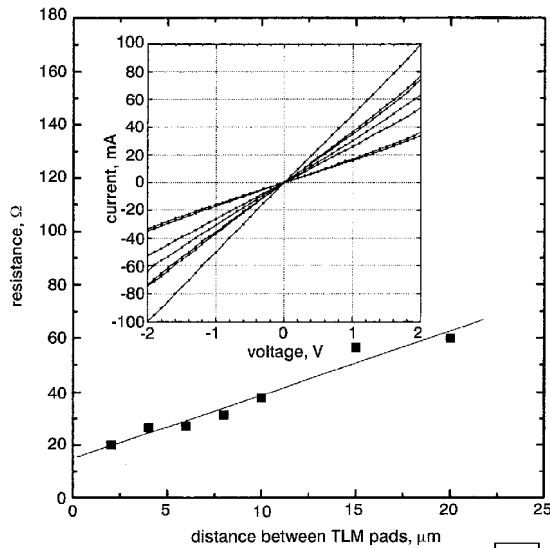


Fig. 3 Resistance against distance for as-deposited Ti/Pd/Au TLM patterns on Zn-diffused InP/InGaAs structure with extra 350 Å InGaAs surface layer

Inset: Measured current-voltage relations for TLM patterns

Conclusion: A simple but effective approach is proposed and demonstrated to achieve as-deposited ohmic contacts based on Zn segregation at heterointerface during Zn diffusion by purposely creating a heterointerface near the surface. Because the surface InGaAs is very thin, it introduces negligible adverse effect for photonic devices such as photodetectors. It is demonstrated that surface hole concentration as high as $7 \times 10^{19} \text{cm}^{-3}$ can be achieved in contrast to the typical range of 2×10^{17} to $2 \times 10^{18} \text{cm}^{-3}$ by using the same Zn diffusion condition. As-deposited ohmic contact with a $3.5 \times 10^{-5} \Omega\text{cm}^2$ specific contact resistance has been demonstrated by using sputtered Ti/Pd/Au multilayers.

© IEE 2001

14 May 2001

Electronics Letters Online No: 20010680

DOI: 10.1049/el:20010680

J.H. Zhao and Y. Li (SiCLAB, ECE Department, School of Engineering, Rutgers University, Piscataway, NJ 08854, USA)

E-mail: jzhao@ece.rutgers.edu

M. Lange, M. Cohen and G.H. Olsen (Sensors Unlimited, Inc., Princeton, NJ 08540-5914, USA)

References

- DILDEY, F., AMANN, M.-C., and TREICHLER, R.: 'Segregation of zinc in InGaAs/InP heterostructures during diffusion: experiment and numerical modeling', *J. Appl. Phys.*, 1990, **29**, pp. 810-812

- BURSTEIN, E.: 'Anomalous optical absorption limit in InSb', *Phys. Rev.*, 1954, **93**, pp. 632-633
- XIANG, N., LAMMASNIEMI, J., KAZANTSEV, A.B., and PESSA, M.: 'Study of light absorption in n-type and p-type GaInAs and the possibility of making 1.55-µm GaInAs/InP Bragg mirrors', *J. Mater. Sci., Mater. Electron.*, 1999, **10**, pp. 255-257
- BERGER, I.H.: 'Models for contacts to planar devices', *Solid-State Electron.*, 1972, **15**, pp. 145-158

Jacobi-like algorithm for blind signal separation of convolutive mixtures

H. Bousbia-Salah, A. Belouchrani and K. Abed-Meraim

A new solution to the blind recovery of independent source signals from their convolutive mixtures is presented. In the case of instantaneous mixtures, a robust solution referred to as second order blind identification (SOBI) has been proposed previously; here, this technique is extended to the convolutive mixture case.

Introduction: Blind signal separation of instantaneous mixtures has received more attention than the blind signal separation of convolutive mixtures. However, more efforts have been concentrated on the blind multi-input multi-output (MIMO) deconvolution. In some applications (e.g. speech application) the separation of the sources up to a filter is sufficient [1]. In this Letter, we consider the problem of the source separation up to a filter instead of the full MIMO deconvolution procedure. Note that, if required, a single-input multi-output (SIMO) deconvolution/equalisation [2] can be applied to the separated sources.

Consider a discrete time MIMO linear time invariant model given by

$$x_i(n) = \sum_{j=1}^M \sum_{l=0}^L h_{ij}(l) s_j(n-l) \quad \text{for } i = 1, \dots, N \quad (1)$$

where $s_j(n)$, $j = 1, \dots, M$, are the M source signals (model inputs), $x_i(n)$, $i = 1, \dots, N$, are the N sensor signals (model outputs), h_{ij} is the transfer function between the j th source and the i th sensor with an overall extent $(L+1)$. The following assumptions are in order:

A1: The source signals $s_j(n)$, $j = 1, \dots, M$, are mutually uncorrelated and each source signal is temporally coherent.

A2: The channel matrix $\tilde{\mathbf{H}}$ defined in eqn. 3 is full column rank.

We can rewrite eqn. 1 in the following matrix form:

$$\mathbf{x}(n) = \tilde{\mathbf{H}}\mathbf{s}(n) \quad (2)$$

where

$$\begin{aligned} \mathbf{s}(n) &= [s_1(n), \dots, s_1(n-(L+L'+1)), \\ &\quad \dots, s_M(n-(L+L'+1))]^T \\ \mathbf{x}(n) &= [x_1(n), \dots, x_1(n-L'+1), \dots, x_N(n-L'+1)]^T \\ \tilde{\mathbf{H}} &= \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \cdots & \tilde{\mathbf{H}}_{1M} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{H}}_{N1} & \cdots & \tilde{\mathbf{H}}_{NM} \end{bmatrix} \quad \text{with} \\ \tilde{\mathbf{H}}_{ij} &= \begin{bmatrix} h_{ij}(0) & \cdots & h_{ij}(L) & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & h_{ij}(0) & \cdots & h_{ij}(L) \end{bmatrix} \end{aligned} \quad (3)$$

Note that $\tilde{\mathbf{H}}$ is a $[NL' \times M(L-L')]$ matrix and $\tilde{\mathbf{H}}_{ij}$ are $[L' \times (L+L')]$ matrices. L' is chosen such that $NL' \geq M(L+L')$. We define the data correlation matrices at time lag k as

$$\mathbf{R}_x(k) = E[\mathbf{x}(n)\mathbf{x}(n-k)^*] \quad (4)$$

Under the linear model (eqn. 2), the above equation can be put in the following form:

$$\mathbf{R}_x(k) = \tilde{\mathbf{H}}\mathbf{R}_s(k)\tilde{\mathbf{H}}^H \quad \text{where } \mathbf{R}_s(k) = E[\mathbf{s}(n)\mathbf{s}(n-k)^*] \quad (5)$$

are the source correlation matrices at time lag k . According to assumption A1, matrix $\mathbf{R}_s(k)$ is block diagonal, with M diagonal blocks each of size $[(L'+L) \times (L'+L)]$.

We now present an algorithm for the recovery of the source signals up to a filter that generalises the second order blind identification (SOBI) algorithm introduced in [3] for the separation of instantaneous mixtures. The new algorithm exploits the block diagonal structure of the source correlation matrices.

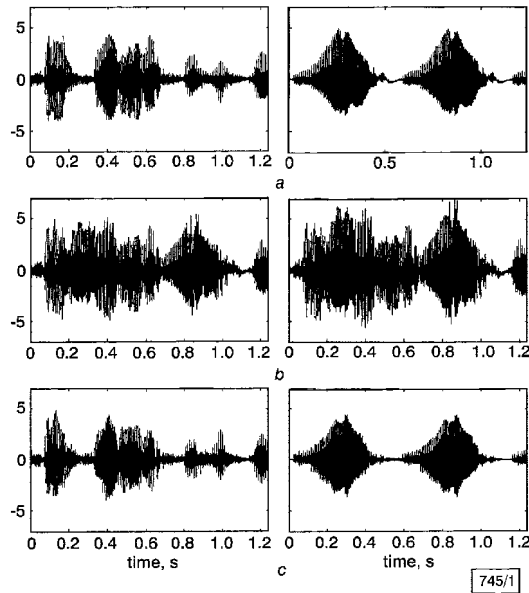


Fig. 1 Speech signal separation

a Two original signals
b Two of the three mixed signals
c Two of the twelve recovered signals

Proposed algorithm: The first step of our procedure consists of whitening the $[NL' \times 1]$ data vector $\mathbf{x}(n)$. This is achieved by applying to $\mathbf{x}(n)$ a whitening matrix \mathbf{W} , i.e. a $[M(L' + L) \times NL']$ matrix verifying:

$$\begin{aligned} E[\mathbf{W}\mathbf{x}(n)\mathbf{x}(n)^* \mathbf{W}^H] &= \mathbf{W}\mathbf{R}_x(0)\mathbf{W}^H \\ &= (\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}})(\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}})^H = \mathbf{I} \end{aligned} \quad (6)$$

Eqn. 6 shows that if \mathbf{W} is a whitening matrix, then

$$\mathbf{U} = \mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}} \quad (7)$$

is a $[M(L' + L) \times M(L' + L)]$ unitary matrix where $\mathbf{R}_s(0)^{\frac{1}{2}}$ is block diagonal. The whitening matrix \mathbf{W} can be determined from the eigen decomposition of the data correlation $\mathbf{R}_x(0)$ as in [3]. Consider now the spatially whitened correlation matrices $\mathbf{R}_x(k)$ defined as

$$\forall k \neq 0 \quad \mathbf{R}_x(k) = \mathbf{W}\mathbf{R}_x(k)\mathbf{W}^H \quad (8)$$

By eqns. 5 and 7, we obtain the key relation

$$\begin{aligned} \forall k \neq 0 \quad \mathbf{R}_x(k) &= \mathbf{U}\mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{R}_s(k)\mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{U}^H \\ &= \mathbf{U}\mathbf{D}(k)\mathbf{U}^H \end{aligned} \quad (9)$$

where we have set $\mathbf{D}(k) = \mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{R}_s(k)\mathbf{R}_s(0)^{-\frac{1}{2}}$.

Since the matrix \mathbf{U} is unitary and \mathbf{D}_k is block diagonal, the latter just means that any whitened data correlation matrix is block diagonal in the basis of the column vectors of matrix \mathbf{U} . The unitary matrix can be retrieved by computing the block diagonalisation of some matrix $\mathbf{R}_x(k)$. But in the case of a singular correlation matrix, indeterminacy can occur. To reduce this indeterminacy and increase the robustness of determining \mathbf{U} , we consider the joint block diagonalisation of a set $\{\mathbf{R}_x(k); k = 1, \dots, K\}$ of K whitened data correlation matrices.

This joint block-diagonalisation (JBD) is achieved by the maximisation under unitary transform of the following criterion:

$$C(\mathbf{U}) \stackrel{\text{def}}{=} \sum_{k=1}^K \sum_{m=1}^M \sum_{i,j=(L'+L)(m-1)+1}^{(L'+L)m} |\mathbf{u}_i^* \mathbf{R}_x(k) \mathbf{u}_j|^2 \quad (10)$$

over the set of unitary matrices $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_{M(L'+L)}]$. Note that an efficient Jacobi-like algorithm for joint block diagonalisation algorithm exists in [4, 5].

Once the unitary matrix \mathbf{U} is determined, the recovered signals are obtained up to a filter by

$$\hat{\mathbf{s}}(n) = \mathbf{U}^H \mathbf{W}\mathbf{x}(n) \quad (11)$$

According to eqns. 2 and 7, the recovered signals verify

$$\hat{\mathbf{s}}(n) = \mathbf{D}\mathbf{R}_s^{-\frac{1}{2}}(0)\mathbf{s}(n) \quad (12)$$

where, we recall that, the matrix $\mathbf{R}_s^{-\frac{1}{2}}(0)$ is block diagonal and \mathbf{D} is a block diagonal unitary matrix coming from the inherent indeterminacy of the JBD problem [5].

Remarks: (1) In practice, only M signals among the $M(L' + L)$ recovered ones are selected. This is performed by choosing the signals which lead to the smallest correlation coefficients. Note that this information is a by-product of the joint block diagonalisation procedure and, hence, this selection needs no additional computations. (2) For simplicity, all the derivations have been presented in the noiseless case. However, we can deal in practice with spatially white noise as shown in [6]. The noise power can be estimated as the least eigenvalue of the zero lag correlation matrix after which the noise correlation matrices can be subtracted from those of the data (for more details see [6]).

Simulation results: In this Section, we present a simulation to illustrate the effectiveness of our algorithm in separating speech signals. The parameter settings are as follows: $M = 2$ (speech signals sampled at 8 kHz), $N = 3$, $L = 2$ and $L' = 4$; the transfer function matrix of the simulated multi channel is given by

$$\mathbf{H}(z) = \begin{bmatrix} 1 & 0.85z^{-1} + 0.1z^{-2} \\ 0.7z^{-1} + 0.4z^{-2} & 0.25 + z^{-1} \\ 1 + 0.5z^{-1} + 0.7z^{-2} & 0.1z^{-1} + 0.85z^{-2} \end{bmatrix}$$

Fig. 1 shows the original speech signals, their convolutive mixture and the recovered signals by the proposed algorithm. This example illustrates the effectiveness of the proposed approach.

© IEE 2001

Electronics Letters Online No: 20010698
DOI: 10.1049/el:20010698

5 February 2001

H. Bousbia-Salah and A. Belouchrani (EEE Department, Ecole Nationale Polytechnique, Algiers, Algeria)

K. Abed-Meraim (Telecom Paris, 46 rue Barrault, 75013 Paris, France)

E-mail: abed@tsi.enst.fr

References

- 1 EHLERS, F., and SCHUSTER, H.G.: 'Blind separation of convolutive mixtures and an application in automatic speech recognition in a noisy environment', *IEEE Trans. Signal Process.*, 1997, pp. 2608–2612
- 2 ABED-MERAIM, K., QIU, W., and HUA, Y.: 'Blind system identification', *Proc. IEEE*, 1997, **85**, (8) pp. 1310–1322
- 3 BELOUCHRANI, A., ABED MERAIM, K., CARDOSO, J.-F., and MOULINES, E.: 'A blind source separation technique using second order statistics', *IEEE Trans. Signal Process.*, 1997, pp. 434–444
- 4 BELOUCHRANI, A., AMIN, M.G., and ABED-MERAIM, K.: 'Direction finding in correlated noise fields based on joint block-diagonalization of spatio-temporal correlation matrices', *IEEE Signal Process. Lett.*, 1997, **4**, (9), pp. 266–268
- 5 BELOUCHRANI, A., ABED-MERAIM, K., and HUA, Y.: 'Jacobi-like algorithms for joint block diagonalization: application to source localization'. Proc. ISAPCS, Australia, 1998
- 6 TONG, L., XU, G., and KAILATH, T.: 'A new approach to blind identification and equalization of multipath channels'. Proc. 25th Asilomar Conf., CA, USA, 1991, pp. 856–860

Programmable processor for on-line computing of inverse Haar transform

O. Martín and J.M. Solana

A processor for the one-dimensional inverse Haar transform (1D-IFHT) programmable for $N = 8$ up to $N = 1024$, with low latency data flow is presented. It enables on-line computing of both the normalised and the non-normalised IFHT to be performed, with very low *nrms*.

Joint Block Diagonalization For Array Signal Processing

Adel Belouchrani*, Karim Abed-Meraim** (contact author)
and Hicham Bousbia-Salah **

* EEE Department, Ecole Nationale Polytechnique, 16200, El Harrach, Algiers, Algeria.

** TSI Department, Telecom Paris, 37-39 rue Dareau, 75014 Paris, France.

e-mail: adel.belouchrani@enp.edu.dz, abed@tsi.enst.fr, hicham.bousbia-salah@enp.edu.dz

Abstract—This paper addresses the problem of joint block diagonalization (JBD) of several matrices that arises in many signal processing applications such as sonar, biomedicine, and communications. A least-squares (LS) approach for computing the JBD is first presented. Based on which, new Jacobi-like algorithms are proposed to minimize the LS criterion. This study is followed by the introduction of a second class of JBD algorithms based on Joint-Diagonalization (JD) procedures. Some comparisons between the different JBD methods are given. An application example of the JBD is presented to solve the problem of blind source separation of convolutive mixtures.

I. INTRODUCTION

Estimating the joint eigenstructure of several matrices is a problem that arises in many multivariate signal processing applications, e.g. joint diagonalization (JD) for source separation [3], joint eigendecomposition for parameter pairing [4], joint block-diagonalization (JBD) for source localization [5] and blind source separation [6], [7], joint Schur decomposition for multidimensional harmonic retrieval [8], [9], [10] and blind system identification [11].

In this paper, we focus on the JBD problem. This problem, to our knowledge, has been first treated in [5], [1] for a set of whitened correlation matrices. In [5], a new JBD approach of singular matrices has been introduced to solve the problem of direction of arrival (DOA) estimation. The blind source separation of convolutive mixtures of speech signals using JBD was first introduced in [6], [12] and later on in [13]. In [14], block diagonalization is used for power control and interference cancelation in multi-input multi-output (MIMO) systems.

Herein, we present a least-squares (LS) approach to JBD. Then, we propose ‘exact’ and ‘approximate’ iterative JBD algorithms to minimize the LS criterion. These algorithms are Jacobi-like techniques that minimize a squared error cost function iteratively by means of Givens rotations. An advantage of the Jacobi methods is their inherent parallelism, which allows efficient implementations on certain parallel architectures [15]. Another virtue of the Jacobi methods is their favorable rounding-error properties, in the sense that small relative perturbation in the matrices entries cause small relative perturbations in the entries of their eigenstructures [16]. Although the two LS algorithms have been, briefly, presented in [1], more details, proofs and new derivations along with computer simulations are given herein. In addition, we explore another way of performing

a JBD of a set of matrices. First, we prove that, although JD and JBD are fundamentally two different problems, the Jacobi-like JD algorithms allow to achieve JBD up to a permutation matrix. Hence, we propose new JBD algorithms using a JD method as a first step and resolving permutation ambiguities as a second step. In this context, we present two different schemes: the first one jointly block-diagonalizes positive definite matrices with no prior whitening, which represents a unique advantage compared to the other methods. The second uses the standard JD algorithm in [17]. We propose, in this paper, two simple algorithms to blindly remove the permutation indeterminacy and complete the JBD. In the JD algorithm, the unitary transform is estimated as a product of Givens rotations but contrary to the previous JBD algorithms, the optimization of the rotation angles leads to a quadratic (instead of linear-quadratic) criterion and, hence has a reduced computational cost.

Our contribution with these four algorithms is a new and efficient solution of the JBD problem which is much more challenging than the JD problem. In fact, although related, these two problems are completely different: while the JD tends to concentrate all the matrices energy into its diagonal entries, the JBD gathers this energy in block diagonal entries. One can classify JD methods as a particular case of JBD methods with a single element in each block.

After algorithms derivations, we present a comparative study, using intensive simulation experiments, of the different algorithms in terms of estimation accuracy, convergence rate, convergence to local minima and sensitivity to the initialization point. Based on this study, we propose a two-step procedure that achieves at best the JBD with respect to the previous criteria.

Finally, we present one application of the JBD in array processing relative to the blind source separation of convolutive mixtures.

The paper is organized as follows: Section II introduces the problem under consideration. In Section III, the LS criterion is derived and in Section IV, two Jacobi-like algorithms (exact and approximate) are introduced to perform the JBD of a set of matrices. In Section V, two different JBD approaches based on JD procedure are presented. Simulation results that include some comparisons between the different JBD methods with respect to the signal to noise ratio (SNR), the size of the matrices to be block-

diagonalized and other parameters along with an application example are presented in section VI. Concluding remarks are given in Section VII.

II. PROBLEM FORMULATION

Consider a set of K matrices, $\mathbf{M}_1, \dots, \mathbf{M}_K$, $\mathbf{M}_k \in \mathbb{C}^{n \times n}$, $k = 1, \dots, K$, that have the following decomposition: For $k = 1, \dots, K$

$$\mathbf{M}_k = [\mathbf{E}_1, \dots, \mathbf{E}_r] \begin{bmatrix} \mathbf{D}_{k1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{D}_{kr} \end{bmatrix} \begin{bmatrix} \mathbf{E}_1^H \\ \vdots \\ \mathbf{E}_r^H \end{bmatrix} = \mathbf{E} \mathbf{D}_k \mathbf{E}^H \quad (1)$$

where $\mathbf{E} = [\mathbf{E}_1, \dots, \mathbf{E}_r]$ is unitary, \mathbf{E}_j is of size $n \times m_j$, \mathbf{D}_{kj} , $j = 1, \dots, r$ are $m_j \times m_j$ square matrices with $m_1 + \dots + m_r = n$ and $\mathbf{D}_k = \text{diag}(\mathbf{D}_{k1}, \dots, \mathbf{D}_{kr})$.

The matrices \mathbf{M}_k , $k = 1, \dots, K$ are said to be jointly block diagonalizable under unitary transform \mathbf{E} , i.e., $\mathbf{E}^H \mathbf{M}_k \mathbf{E}$ are block diagonal matrices for $k = 1, \dots, K$. \mathbf{E}^H denotes the transpose conjugate of \mathbf{E} . Note that, matrices \mathbf{M}_k , $k = 1, \dots, K$ are not, in general, jointly diagonalizable except for the particular case where $m_1 = \dots = m_r = 1$. Hence, JD can be seen as a particular case of JBD when $m_j = 1, \forall j$.

The problem of JBD consists of estimating the matrices \mathbf{E} and \mathbf{D}_{kj} , $k = 1, \dots, K$, $j = 1, \dots, r$ given the matrices \mathbf{M}_k , $k = 1, \dots, K$. Note that the JBD decomposition is not unique since if $\{\mathbf{E}_1, \dots, \mathbf{E}_r, \mathbf{D}_{11}, \dots, \mathbf{D}_{Kr}\}$ is a solution then $\mathbf{E}'_j = \mathbf{E}_j \mathbf{U}_j$ and $\mathbf{D}'_{kj} = \mathbf{U}_j^H \mathbf{D}_{kj} \mathbf{U}_j$ is also another admissible solution, where \mathbf{U}_j , $j = 1, \dots, r$ are unitary matrices. In other words, matrix \mathbf{E} can be determined only up to a block diagonal unitary matrix. Moreover, any diagonal block permutation leads to a valid solution as there is no specific sorting of these blocks. However, in most practical applications these indeterminacies are inherent and do not affect the final result of the considered problem.

In practice, the matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ are given by some sample estimated statistics that are corrupted by estimation errors due to noise and finite sample size effects. Thus, they are only ‘‘approximately’’ simultaneously block diagonalizable. This suggests that a viable JBD algorithm must provide a kind of an ‘‘average eigenstructure’’ when it is applied to a set of nearly joint block diagonalizable matrices. An optimal solution based on a least-squares approach is given next¹.

III. LEAST-SQUARES APPROACH

A least-squares (LS) fitting technique consists here of choosing the $n \times n$ unitary matrix \mathbf{E} and the $m_j \times m_j$ matrices \mathbf{D}_{kj} that minimize the Frobenius norm of the difference between the data matrices \mathbf{M}_k and the true matrices given by (1), i.e.

$$\min_{\mathbf{E}, \mathbf{D}_k} \sum_{k=1}^K \|\mathbf{M}_k - \mathbf{E} \mathbf{D}_k \mathbf{E}^H\|^2 \quad (2)$$

¹A similar approach has been presented in [18] for joint diagonalization and in [9] for joint Schur decomposition.

Since the Frobenius norm is invariant under unitary transformation, the LS criterion in (2) is equivalent to,

$$\min_{\mathbf{E}, \mathbf{D}_k} \sum_{k=1}^K \|\mathbf{E}^H \mathbf{M}_k \mathbf{E} - \mathbf{D}_k\|^2 \quad (3)$$

The minimization of (3) with respect to \mathbf{D}_k is, clearly, attained by,

$$\mathbf{D}_k = \text{bdiag}(\mathbf{E}^H \mathbf{M}_k \mathbf{E})$$

where the notation $\text{bdiag}(\mathbf{M})$ is a block diagonal matrix constructed from \mathbf{M} in the following way:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \cdots & \mathbf{M}_{1r} \\ \vdots & & \vdots \\ \mathbf{M}_{r1} & \cdots & \mathbf{M}_{rr} \end{bmatrix} \implies \text{bdiag}(\mathbf{M}) = \begin{bmatrix} \mathbf{M}_{11} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{rr} \end{bmatrix}. \quad (4)$$

Where \mathbf{M}_{jj} are of size $m_j \times m_j$, $j = 1, \dots, r$. Thus, again, since the Frobenius norm is invariant under unitary transformation, minimization of (3) is equivalent to the maximization of the part that is eliminated namely²,

$$\max_{\mathbf{E}} \sum_{k=1}^K \|\text{bdiag}(\mathbf{E}^H \mathbf{M}_k \mathbf{E})\|^2 \quad (5)$$

In other words, the JBD of $\{\mathbf{M}_1, \dots, \mathbf{M}_K\}$ consists of maximizing under unitary transform \mathbf{E} the sum of the norms of their block diagonal entries. This is equivalent to minimizing the sum of the norms of their block off-diagonal entries. It is clear that the latter is minimum (zero) when all matrices are exactly block diagonal. Criterion (5) is referred to as the JBD criterion in the sequel.

IV. JACOBI-LIKE JBD ALGORITHMS

To minimize the JBD criterion (5), we choose here to compute the unitary matrix \mathbf{E} as products of Givens rotations that we describe next.

A. Givens rotations

In Jacobi-like algorithms, a unitary matrix \mathbf{U} is decomposed into a product of elementary Givens rotations, i.e.,

$$\mathbf{U} = \prod_{\# \text{ sweeps}} \prod_{1 \leq p < q \leq n} \Theta_{(qp)}$$

where $\# \text{ sweeps}$ stands for the number of sweeps, each sweep represents the processing of all $n(n-1)/2$ pairs of indices (p, q) . The elementary Givens rotations $\Theta_{(qp)}$ are defined as unitary matrices where all diagonal elements are 1 except for the two elements equal to c in rows (and columns) p and q . Likewise, all off-diagonal elements of

²Note that in [1], another proof is given, but the one given above is shorter.

$\Theta_{(qp)}$ are 0 except for the two elements s and $-\bar{s}$ at positions (p, q) and (q, p) respectively, where \bar{s} denotes the conjugate of s . The scalar numbers c and s are given by

$$\begin{cases} c &= \cos \theta \\ s &= \sin \theta \exp(i\alpha) \end{cases}$$

where $i = \sqrt{-1}$. In the sequel, we describe a procedure to choose the rotation angles θ and α at a particular iteration such that the cost function (5) is increased to its maximum. To this end, we need to specify the orthogonal transformation

$$\mathbf{M}' = \Theta_{(qp)}^H \mathbf{M} \Theta_{(qp)} \quad (6)$$

for any given matrix $\mathbf{M} \in \mathbb{C}^{n \times n}$. These orthogonal transformations change only the p^{th} and q^{th} rows and p^{th} and q^{th} columns of \mathbf{M} according to:

$$\begin{aligned} \mathbf{M}'(p, j) &= c\mathbf{M}(p, j) - s\mathbf{M}(q, j), & j \neq p \\ \mathbf{M}'(j, p) &= c\mathbf{M}(j, p) - \bar{s}\mathbf{M}(j, q), & j \neq p \\ \mathbf{M}'(q, j) &= \bar{s}\mathbf{M}(p, j) + c\mathbf{M}(q, j), & j \neq q \\ \mathbf{M}'(j, q) &= s\mathbf{M}(j, p) + c\mathbf{M}(j, q), & j \neq q \\ \mathbf{M}'(p, p) &= c^2\mathbf{M}(p, p) + |s|^2\mathbf{M}(q, q) - \bar{s}c\mathbf{M}(p, q) - sc\mathbf{M}(q, p) \\ \mathbf{M}'(q, q) &= c^2\mathbf{M}(q, q) + |s|^2\mathbf{M}(p, p) + \bar{s}c\mathbf{M}(q, p) + sc\mathbf{M}(p, q) \\ \mathbf{M}'(p, q) &= cs\mathbf{M}(p, p) + c^2\mathbf{M}(p, q) - s^2\mathbf{M}(q, p) - sc\mathbf{M}(q, q) \\ \mathbf{M}'(q, p) &= \bar{s}c\mathbf{M}(p, p) - \bar{s}^2\mathbf{M}(p, q) + c^2\mathbf{M}(q, p) - c\bar{s}\mathbf{M}(q, q) \end{aligned} \quad (7)$$

Notice that, when p and q sweep the same diagonal block, the value of the JBD criterion (5) is unchanged as stated in the following lemma:

Lemma 1: The norm of the block-diagonal matrix of \mathbf{M}' , is equal to that of \mathbf{M} , i.e.

$$\|\text{bdiag}(\mathbf{M}')\| = \|\text{bdiag}(\mathbf{M})\| \quad (8)$$

if p and q sweep the same diagonal block, i.e. $(1 \leq p, q \leq m_1)$, or $m_1 < p, q \leq m_1 + m_2, \dots$, or $\sum_{i=1}^{r-1} m_i < p, q \leq n$.

Proof: Without loss of generality, assume $1 \leq p, q \leq m_1$, so that only the first diagonal block of matrix \mathbf{M} is modified. By observing that, for $j \neq p$ and $j \neq q$:

$$\begin{aligned} |\mathbf{M}'(p, j)|^2 + |\mathbf{M}'(q, j)|^2 &= |\mathbf{M}(p, j)|^2 + |\mathbf{M}(q, j)|^2 \\ |\mathbf{M}'(j, p)|^2 + |\mathbf{M}'(j, q)|^2 &= |\mathbf{M}(j, p)|^2 + |\mathbf{M}(j, q)|^2 \end{aligned}$$

Equation (8) reduces to the following equality,

$$\begin{aligned} &|\mathbf{M}'(p, p)|^2 + |\mathbf{M}'(p, q)|^2 + |\mathbf{M}'(q, p)|^2 + |\mathbf{M}'(q, q)|^2 \\ &= |\mathbf{M}(p, p)|^2 + |\mathbf{M}(p, q)|^2 + |\mathbf{M}(q, p)|^2 + |\mathbf{M}(q, q)|^2 \end{aligned}$$

Let,

$$\mathcal{M}_{pq} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{M}(p, p) & \mathbf{M}(p, q) \\ \mathbf{M}(q, p) & \mathbf{M}(q, q) \end{bmatrix}.$$

Then, it is easy to observe that,

$$\mathcal{M}'_{pq} = \begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix}^H \mathcal{M}_{pq} \begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix}.$$

and hence $\|\mathcal{M}'_{pq}\|^2 = \|\mathcal{M}_{pq}\|^2$ which corresponds to the previous equality. \square

Based on the previous lemma, we consider, in our JBD algorithms, only indices (p, q) that sweep two different diagonal blocks, e.g. for $r = 2$ we select p and q in the range $1 \leq p \leq m_1 < q \leq n$. In other words, we choose $(p, q) \in \mathcal{I}$ where,

$$\mathcal{I} = \{(p, q) \mid \exists 0 \leq i_p < i_q < r, \quad (9)$$

$$\sum_{j=1}^{i_p} m_j < p \leq \sum_{j=1}^{i_p+1} m_j \quad \text{and} \quad \sum_{j=1}^{i_q} m_j < q \leq \sum_{j=1}^{i_q+1} m_j\}.$$

B. Exact JBD algorithm

The proposed method consists of maximizing iteratively the JBD criterion (5) by successive Givens rotations, starting from $\mathbf{E} = \mathbf{I}$. At the end of the iterative procedure, we have

$$\mathbf{E} = \prod_{\# \text{ sweeps}} \prod_{1 \leq p < q \leq n} \Theta_{(qp)}$$

where $\Theta_{(qp)}$ are computed such that (5) is maximum³, i.e., at each iteration, the angle parameters (θ, α) are given by:

$$(\theta, \alpha) = \arg \max_{\theta, \alpha} \mathcal{C}(\theta, \alpha)$$

$$\mathcal{C}(\theta, \alpha) \stackrel{\text{def}}{=} \sum_{k=1}^K \|\text{bdiag}(\mathbf{M}'_k)\|^2 \quad (10)$$

where \mathbf{M}'_k is defined as in (6). The exact JBD algorithm can be summarized as follows (using informal notation):

$\mathbf{E} = \mathbf{I}$ for $k = 1, \dots, \# \text{ sweeps}$ for $(p, q) \in \mathcal{I}$ $\Theta_{(qp)} = \arg \max \mathcal{C}(\theta, \alpha)$ $\mathbf{E} := \mathbf{E} \Theta_{(qp)}$ and $\mathbf{M}_k := \Theta_{(qp)}^H \mathbf{M}_k \Theta_{(qp)}$, $k = 1, \dots, K$
--

After some straightforward derivations (see appendix A), the maximization of $\mathcal{C}(\theta, \alpha)$ is shown to be equivalent to the maximization of the quadratic form

$$\max_{\|\mathbf{v}\|=1} (\mathbf{v}^T \mathbf{G} \mathbf{v} + \mathbf{g}^T \mathbf{v}) \quad \text{where,} \quad (11)$$

$$\mathbf{v} = [\cos(2\theta), \sin(2\theta) \cos(\alpha), \sin(2\theta) \sin(\alpha)]^T \quad (12)$$

and \mathbf{G} (resp. \mathbf{g}) is a 3×3 real-valued matrix (resp. a 3×1 real-valued vector) the expressions of which are given in appendix A.

Using a Lagrange multiplier, the maximization of (11) leads to:

$$2(\mathbf{G} + \lambda \mathbf{I})\mathbf{v} + \mathbf{g} = \mathbf{0} \quad \text{and hence,} \quad (13)$$

$$\mathbf{v} = -\frac{1}{2}(\mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{g} \quad (14)$$

where λ is a real scalar chosen in such a way that

$$\|\mathbf{v}\|^2 = 1 \quad \iff \quad \frac{1}{4} \sum_{i=1}^3 \frac{|\mathbf{u}_i^T \mathbf{g}|^2}{(\lambda_i + \lambda)^2} = 1 \quad (15)$$

³Note that the criterion is calculated with the current values of the matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ which are updated at each iteration using the above orthogonal transformation. For notational simplicity, we keep using the generic notation $\mathbf{M}_1, \dots, \mathbf{M}_K$ to denote the updated matrices.

$\{\mathbf{u}_i\}$ and $\{\lambda_i\}$ being the eigenvectors and eigenvalues of \mathbf{G} . Otherwise, the solution of (13) may be of the form:

$$\mathbf{v} = -\frac{1}{2}(\mathbf{G} - \lambda_i \mathbf{I})^\# \mathbf{g} + c_i \mathbf{u}_i, \quad i = 1, 2, 3 \quad (16)$$

which corresponds to the case where λ coincides with the opposite of an eigenvalue of \mathbf{G} , i.e., $-\lambda_i$, $i = 1, 2, 3$. c_i is a real constant chosen such that $\|\mathbf{v}\| = 1$ and (11) is maximum⁴. It is given by $c_i = \text{sign}(\mathbf{u}_i^T \mathbf{g})(1 - \|(\mathbf{G} - \lambda_i \mathbf{I})^\# \mathbf{g}\|^2/4)^{\frac{1}{2}}$ where $\text{sign}(x)$ denotes the sign of x .

Remark: We have observed in our simulations that the case $\lambda = -\lambda_i$, $i = 1, 2, 3$ is almost never met in practice. Hence, the algorithm can be simplified to solving (14) and (15) only. As we can see from (15), finding λ involves a 6th order polynomial rooting. In general, equation (15) has multiple real-valued roots λ leading to multiple solutions for vector \mathbf{v} . Among the set of all possible solutions, we select the one associated with the vector \mathbf{v} that maximizes the value of $\mathcal{C}(\theta, \alpha)$, i.e., the vector \mathbf{v} that maximizes (11). Note that in the real data case, the matrix \mathbf{G} is 2×2 and vector \mathbf{g} is 2×1 (see appendix A). In this case, (15) becomes a 4-th order polynomial equation. The latter can be obtained as well by parameterizing the Givens rotations in terms of $t = \tan \theta$, and replacing in (11),

$$\cos(\theta) = \frac{1}{\sqrt{1+t^2}}, \quad \sin(\theta) = \frac{t}{\sqrt{1+t^2}}$$

Maximizing (11) as a function of t leads to a similar 4-th order polynomial rooting. This has been used, for example, in [19] for joint Schur decomposition. Once $\mathbf{v} = [\mathbf{v}(1), \mathbf{v}(2), \mathbf{v}(3)]^T$ is obtained, the Givens rotation parameters c and s are computed according to (12) as

$$c = \sqrt{\frac{\mathbf{v}(1) + 1}{2}}, \quad s = \frac{\mathbf{v}(2) + i\mathbf{v}(3)}{2c} \quad (17)$$

Next, by means of a slight approximation of the JBD criterion, we present an alternative solution to (11) where no polynomial rooting is required.

C. Approximate JBD algorithm

To simplify the JBD algorithm, we choose here to approximate $|\mathbf{M}'(p, p)|^2$ (up to a scalar constant independent from the rotation parameters (θ, α)) by

$$|\mathbf{M}'(p, p)|^2 \approx |[\mathbf{M}\Theta_{(qp)}](p, p)|^2 + |[\Theta_{(qp)}^H \mathbf{M}](p, p)|^2.$$

This can be shown to be a first order approximation for $|\mathbf{M}'(p, p)|^2$ in the vicinity of the optimal point. Effectively, by writing $\Theta_{(qp)} = \mathbf{I} + \epsilon_{qp}$ (with $\|\epsilon_{qp}\| \ll 1$ in the neighborhood of the convergence point), we obtain as a first order approximation

$$\begin{aligned} |\mathbf{M}'(p, p)|^2 &\approx |[\mathbf{M} + \epsilon_{qp}^H \mathbf{M} + \mathbf{M}\epsilon_{qp}](p, p)|^2 \\ &\approx 2\Re\{\mathbf{M}(p, p)[\epsilon_{qp}^H \mathbf{M}](p, p)\} \end{aligned}$$

⁴This solution exists only if $\|(\mathbf{G} - \lambda_i \mathbf{I})^\# \mathbf{g}\|/2 \leq 1$. Note that, when it exists, both c_i and $-c_i$ satisfy $\|\mathbf{v}\| = 1$. We then choose the one that maximizes (11) which is given by $\text{sign}(c_i) = \text{sign}(\mathbf{u}_i^T \mathbf{g})$.

$$\begin{aligned} &+ \mathbf{M}(p, p)[\mathbf{M}\epsilon_{qp}](p, p) + |\mathbf{M}(p, p)|^2 \\ &\approx |[\mathbf{M}\Theta_{(qp)}](p, p)|^2 \\ &+ |[\Theta_{(qp)}^H \mathbf{M}](p, p)|^2 - |\mathbf{M}(p, p)|^2 \end{aligned} \quad (18)$$

since at the first order we have

$$\begin{aligned} |[\mathbf{M}\Theta_{(qp)}](p, p)|^2 &\approx |\mathbf{M}(p, p)|^2 + 2\Re\{\mathbf{M}(p, p)[\mathbf{M}\epsilon_{qp}](p, p)\} \\ |[\Theta_{(qp)}^H \mathbf{M}](p, p)|^2 &\approx |\mathbf{M}(p, p)|^2 + 2\Re\{\mathbf{M}(p, p)[\epsilon_{qp}^H \mathbf{M}](p, p)\}. \end{aligned}$$

With this approximation, the JBD criterion becomes of the form (see appendix B)

$$\mathcal{C}(\theta, \alpha) \approx \tilde{\mathbf{g}}^T \mathbf{v} \quad (19)$$

and its maximization leads to the explicit expressions:

$$\begin{aligned} \alpha &= \arctan\left(\frac{\Im m(a)}{\Re e(a)}\right), \\ \theta &= \frac{1}{2} \arctan\left(-2 \frac{\Re e(e^{-i\alpha} a)}{b}\right) + \frac{1 - \text{sign}(b)}{2} \frac{\pi}{2} \\ a &= \sum_{k=1}^K \left(\sum_{j \in \mathcal{I}_p} \mathbf{M}_k(p, j) \overline{\mathbf{M}}_k(q, j) + \mathbf{M}_k(j, p) \overline{\mathbf{M}}_k(j, q) \right. \\ &\quad \left. - \sum_{j \in \mathcal{I}_q} \mathbf{M}_k(p, j) \overline{\mathbf{M}}_k(q, j) + \mathbf{M}_k(j, p) \overline{\mathbf{M}}_k(j, q) \right) \\ b &= \sum_{k=1}^K \left(\sum_{j \in \mathcal{I}_p} |\mathbf{M}_k(p, j)|^2 + |\mathbf{M}_k(j, p)|^2 - |\mathbf{M}_k(q, j)|^2 \right. \\ &\quad \left. - |\mathbf{M}_k(j, q)|^2 - \sum_{j \in \mathcal{I}_q} |\mathbf{M}_k(p, j)|^2 + |\mathbf{M}_k(j, p)|^2 \right. \\ &\quad \left. - (|\mathbf{M}_k(q, j)|^2 + |\mathbf{M}_k(j, q)|^2) \right) \end{aligned}$$

where $\Re e(a)$ and $\Im m(a)$ denote the real part and imaginary part of a , respectively and

$$\mathcal{I}_p = \{j | \sum_{l=1}^{i_p} m_l < j \leq \sum_{l=1}^{i_p+1} m_l\}; \quad \mathcal{I}_q = \{j | \sum_{l=1}^{i_q} m_l < j \leq \sum_{l=1}^{i_q+1} m_l\}$$

i_p and i_q are defined as in (9).

V. JBD VIA JOINT DIAGONALIZATION

Another way of performing a JBD of a set of matrices up to a permutation matrix is to use a JD method. We show here two different schemes: the first one block-diagonalizes these matrices using a Kullback-Leibler deviation between two positive definite matrices. The second is a Jacobi-like algorithm that performs a JBD under a common unitary transform.

A. JBD via JD of positive matrices

In this section, we derive a joint block-diagonalization criterion inspired from the joint diagonalization criterion of Pham in [20]. It is based on a maximum likelihood method that uses the Kullback-Leibler (KL) divergence between

two positive definite matrices. In the JD case, the KL divergence leads to the following criterion [20]:

$$\sum_{k=1}^K n_k [\log \det(\text{diag}(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)) - \log \det(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)] \quad (20)$$

Where \mathbf{B} is a diagonalization matrix, not necessarily unitary, n_k are positive weights and \mathbf{M}_k , $k = 1 \dots K$, are positive definite matrices. In the JBD case, the above measure can be adapted as follows:

$$\sum_{k=1}^K n_k [\log \det(\text{bdiag}(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)) - \log \det(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)] \quad (21)$$

This criterion was used in [12], [13] for the blind separation of convolutive mixtures with $n_1 = \dots = n_K = 1$. However, to our knowledge, no proof was provided to justify for the validity of (21) as a JBD criterion. A proof is given here by the following lemma:

Lemma 2: Criterion (21) is a positive valued function equal to zero if and only if $\mathbf{M}_k = \text{bdiag}(\mathbf{M}_k)$ for all $k = 1 \dots K$.

Proof: It is well established (see, for example, [20]) that for any positive definite matrix \mathbf{M} , we have:

$$\det(\mathbf{M}) \leq \det(\text{diag}(\mathbf{M})) \quad (22)$$

With equality if and only if \mathbf{M} is diagonal, i.e. $\mathbf{M} = \text{diag}(\mathbf{M})$. Also, for any positive definite matrix \mathbf{M} , the diagonal blocks \mathbf{M}_i , $i = 1 \dots r$, and, in particular, $\text{bdiag}(\mathbf{M})$ are positive definite. Let us write the eigendecomposition of \mathbf{M}_i i.e. $\mathbf{M}_i = \mathbf{U}_i \Lambda_i \mathbf{U}_i^H$. Where \mathbf{U}_i are unitary and Λ_i are positive diagonal matrices and let $\mathbf{U} \stackrel{\text{def}}{=} \text{diag}(\mathbf{U}_1 \dots \mathbf{U}_r)$. Based on this, we have:

$$\det(\text{bdiag}(\mathbf{M})) = \prod_{i=1}^r \det(\mathbf{M}_i) = \prod_{i=1}^r \det(\Lambda_i)$$

On the other hand,

$$\begin{aligned} \det(\mathbf{M}) &= \det(\mathbf{U}\mathbf{M}\mathbf{U}^H) \\ &\leq \det(\text{diag}(\mathbf{U}\mathbf{M}\mathbf{U}^H)) = \prod_{i=1}^r \det(\Lambda_i) = \det(\text{bdiag}(\mathbf{M})) \end{aligned}$$

The equality holds if and only if $\mathbf{U}\mathbf{M}\mathbf{U}^H = \text{diag}(\mathbf{U}\mathbf{M}\mathbf{U}^H)$. At the end,

$$\mathbf{M} = \mathbf{U}^H \text{diag}(\mathbf{U}\mathbf{M}\mathbf{U}^H) \mathbf{U} = \text{bdiag}(\mathbf{M}). \quad \square$$

It follows that criterion (21) is a measure of the global deviation of the matrices from block-diagonal structure. In [13], gradient-based algorithms were proposed to minimize criterion (21). However, besides their slow convergence, we have observed that for a random initialization point, these algorithms converge most of the time to local minima and hence do not provide satisfactory solutions. In [12], Pham's JD-algorithm was used to minimize criterion (21). Indeed, as shown in [20], Pham's JD algorithm consists in performing successive transformations,

each time on a pair of rows of \mathbf{B} , \mathbf{b}_p and \mathbf{b}_q , according to:

$$\begin{bmatrix} \mathbf{b}_p \\ \mathbf{b}_q \end{bmatrix} \longleftarrow \mathbf{T}_{pq} \begin{bmatrix} \mathbf{b}_p \\ \mathbf{b}_q \end{bmatrix} \quad (23)$$

where \mathbf{T}_{pq} is a 2×2 non singular matrix, chosen such that the JD criterion is sufficiently decreased. Once this is done, the procedure is repeated with another pair of rows (the processing of all the $n(n-1)/2$ pairs of rows is called a sweep). The algorithm consists in repeated sweeps until convergence is achieved. The decrease of the JD criterion is shown in [20] to be equal to:

$$\begin{aligned} &\sum_{k=1}^K n_k \{2 \log |\det \mathbf{T}_{pq}| + \log((\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp}(\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qq}) - \\ &\log \det \text{diag}(\mathbf{T}_{pq} \begin{bmatrix} (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp} & (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pq} \\ (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qp} & (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qq} \end{bmatrix} \mathbf{T}_{pq}^H)\} \quad (24) \end{aligned}$$

Subscripts $_{pq}$ refers to the $(p, q)^{th}$ entry of the considered matrix. By adopting the same strategy, it can be shown that the decrease of the JBD criterion is either zero if p and q sweep the same block or otherwise corresponds to:

$$\begin{aligned} &\sum_{k=1}^K n_k \{2 \log |\det \mathbf{T}'_{pq}| + \log(\det [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{pp} \det [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{qq}) \\ &- \log \det \text{bdiag}(\mathbf{T}'_{pq} \begin{bmatrix} [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{pp} & [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{pq} \\ [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{qp} & [\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{qq} \end{bmatrix} \mathbf{T}'_{pq})\} \quad (25) \end{aligned}$$

\mathbf{T}'_{pq} is the matrix of dimension $(m_p + m_q) \times (m_p + m_q)$ that performs, when multiplied right by the block matrix in (25), the same row transforms as in (23). $[\mathbf{B}\mathbf{M}_k\mathbf{B}^H]_{pq}$ denotes the $(p, q)^{th}$ block matrix of $\mathbf{B}\mathbf{M}_k\mathbf{B}^H$ of size $m_p \times m_q$.

Instead of maximizing the decrease in (24), Pham proposed to maximize a lower bound of it that leads to simpler and tractable implementation. Now, one can observe that (24) is already a lower bound of (25). This justifies the use of Pham's algorithm⁵ in order to achieve the JBD. As shown in [20], at each iteration, \mathbf{T}_{pq} is computed as:

$$\mathbf{T}_{pq} = \mathbf{I} - \frac{2}{1 + \sqrt{1 - 4h_{pq}h_{qp}}} \begin{bmatrix} 0 & h_{pq} \\ h_{qp} & 0 \end{bmatrix}. \quad (26)$$

with the following definitions,

$$\begin{bmatrix} h_{pq} \\ h_{qp} \end{bmatrix} = \begin{bmatrix} w_{pq} & 1 \\ 1 & w_{qp} \end{bmatrix}^{-1} \begin{bmatrix} g_{pq} \\ g_{qp} \end{bmatrix}. \quad (27)$$

$$g_{pq} = \sum_{k=1}^K \frac{n_k (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pq}}{n (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp}}, \quad w_{pq} = \sum_{k=1}^K \frac{n_k (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{qq}}{n (\mathbf{B}\mathbf{M}_k\mathbf{B}^H)_{pp}}.$$

B. JBD via JD under a common unitary transform

In [3], [17], a Jacobi-like algorithm has been introduced to perform the joint diagonalization of a set of matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$, not necessarily positive or Hermitian as in

⁵Pham's algorithm is slightly modified in such a way to consider only pairs of indices p and q that sweep two distinct blocks.

section V-A, under a common unitary transform \mathbf{E} . Matrix \mathbf{E} is estimated as a product of Givens rotations but contrary to the JBD algorithm, the optimization of the rotation angles leads to a quadratic (instead of linear-quadratic) criterion of the form $\mathbf{v}^T \tilde{\mathbf{G}} \mathbf{v}$ where $\tilde{\mathbf{G}}$ is a 3×3 (or 2×2 in the real case) real-valued matrix and \mathbf{v} is the vector defined in (12). Therefore, \mathbf{v} corresponds to the unit-norm least eigenvector of $\tilde{\mathbf{G}}$ that can be computed either explicitly or numerically.

We propose here to use the JD algorithm in [17] to perform the JBD. Indeed, we claim that the JBD can be achieved (up to a permutation matrix) by applying the JD algorithm to the considered set of matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$. More precisely, we have the following lemma:

Lemma 3: Let $\mathbf{M}_1, \dots, \mathbf{M}_K$ be a set of K matrices satisfying (1). Then, optimizing the JD criterion for $\mathbf{M}_1, \dots, \mathbf{M}_K$ using the Jacobi-like algorithm in [17] provides a set of block diagonal matrices up to an unknown permutation matrix \mathbf{P} .

Proof: This result comes from the following observation: It is shown that optimizing the JD criterion by successive Givens rotations leads to solving the same problem for 2×2 matrices: i.e., the optimization of $\Theta_{(qp)}$ is equivalent to the JD of the set of 2×2 matrices

$$\mathbf{M}_k^{(qp)} = \begin{bmatrix} \mathbf{M}_k(p,p) & \mathbf{M}_k(p,q) \\ \mathbf{M}_k(q,p) & \mathbf{M}_k(q,q) \end{bmatrix}, \quad k = 1, \dots, K. \quad (28)$$

The iterative algorithm is stopped when at a given sweep (iteration) we have⁶:

$$\Theta_{(qp)} = \mathbf{I} \quad \forall \quad 1 \leq p < q \leq n \quad (29)$$

which means that the JD criterion (i.e. $\sum_k \|\mathbf{E}^H \mathbf{M}_k \mathbf{E} - \text{diag}(\mathbf{E}^H \mathbf{M}_k \mathbf{E})\|^2$) cannot be further decreased.

Now, if the 2×2 matrices in (28) are non-diagonal for values of p and q corresponding to non-diagonal block entries (i.e. $(p, q) \in \mathcal{I}$), then there exists a unitary matrix and, hence, a Givens rotation that transforms them onto diagonal matrices (because $\mathbf{M}_1, \dots, \mathbf{M}_K$ satisfy (1)). Consequently, the JD criterion can be further decreased and the stopping condition (29) is not verified in this case. \square

Remark: In [2], this result has been presented as a conjecture. The reason is that it is based, implicitly, on the global convergence assumption of the Jacobi-like JD algorithm which is, to our best knowledge, not proved yet. In other words, the lemma does not apply if our algorithm converges to a local minimum.

In practice, performing the JBD up to an unknown permutation matrix is not satisfactory as the ‘admissible’ permutations are only those which preserve the block-diagonal structure (e.g., permuting the diagonal blocks or permuting entries within the same diagonal block is admissible). To remove this undesirable permutation matrix, we propose the two following solutions, as a second stage of the algorithm:

⁶ $\Theta_{(qp)} = \mathbf{I}$ is equivalent to $s = 0$. In practice, we use a small threshold ϵ and test if for all $1 \leq p < q \leq n$, $|s| < \epsilon$.

- In the first one, the permutation \mathbf{P} is decomposed as a product of elementary permutations⁷ $\mathbf{P}_{(qp)}$. The latter is considered, at a given sweep, only if it increases the JBD criterion, i.e.

$$\sum_{k=1}^K \|\text{bdiag}(\mathbf{P}_{(qp)}^T \mathbf{M}_k \mathbf{P}_{(qp)})\|^2 > \sum_{k=1}^K \|\text{bdiag}(\mathbf{M}_k)\|^2 \quad (30)$$

$\mathbf{P}_{(qp)}$ is such that all its diagonal elements are "1" except for the two elements equal to "0" in rows (and columns) p and q . Likewise, all off-diagonal elements of $\mathbf{P}_{(qp)}$ are "0" except for the $(p, q)^{th}$ and the $(q, p)^{th}$ entries that are equal to "1".

As detailed in IV-B, increasing the above criterion leads to the maximization of the quadratic form: $\max_{\|\mathbf{v}\|=1} (\mathbf{v}^T \mathbf{G} \mathbf{v} + \mathbf{g}^T \mathbf{v})$.

The left hand-side of (30) corresponds to the case $\mathbf{v} = [-1, 0, 0]^T$, while the right hand-side corresponds to $\mathbf{v} = [1, 0, 0]^T$. Therefore, the inequality in (30) reduces to:

$$-\mathbf{g}(1) > \mathbf{g}(1) \quad (31)$$

Or equivalently $\text{sign}(\mathbf{g}(1)) < 0$, $\mathbf{g}(1)$ being the first entry of vector \mathbf{g} . Finally, one obtains:

$$\mathbf{P} = \prod_{nb \text{ of sweeps}} \prod_{1 \leq p < q \leq n} \tilde{\mathbf{P}}_{(qp)}$$

$\tilde{\mathbf{P}}_{(qp)}$ being either the identity matrix or the above permutation matrix $\mathbf{P}_{(qp)}$ depending on the binary-decision rule in (31).

We stop the iterative process, when all matrices $\tilde{\mathbf{P}}_{(qp)}$ are equal to the identity. In our simulations, we have observed that one or, at most, two sweeps are sufficient to get the desired permutation.

- The second approach consists in determining the permutation from the positions of the non-zero entries of \mathbf{M}_k . Iteratively and starting from the first row, the algorithm detects the positions of the non-zero entries and group the corresponding column and row vectors in such a way to form the diagonal blocks of the considered matrices.

This leads to a simple solution (see a pseudo-MATLAB code in appendix C) when the considered set of matrices satisfies (1) exactly. This solution has also the advantage of being completely independent of the block sizes. Meaning that, contrary to the other approaches, we do not need to know, *a priori*, the parameters m_1, \dots, m_r . However, when matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ are only approximately JB diagonalizable, this approach requires a non-trivial thresholding to decide whether a given matrix entry is assimilated to zero or not. In that case, the previous solution is preferred as it requires no thresholding.

⁷ $\mathbf{P}_{(qp)}$ is defined in such a way that for a given vector \mathbf{x} , $\tilde{\mathbf{x}} = \mathbf{P}_{(qp)} \mathbf{x}$ iff $\tilde{x}(k) = x(k)$, for $k \notin \{p, q\}$, $\tilde{x}(p) = x(q)$ and $\tilde{x}(q) = x(p)$.

VI. SIMULATION RESULTS

This section is devoted to the performance analysis and validation of the proposed algorithms.

At first, the convergence rate and estimation accuracy of all proposed algorithms are studied and compared through intensive simulation experiments. This is followed by an application example that illustrates the usefulness of the JBD to solve the problem of blind source separation of convolutive mixtures.

A. Performance evaluation

In this subsection, we compare the performance of the considered JBD algorithms, namely the LSBSD for the exact least-squares JBD algorithm, the ALSBD for the Approximate LSBSD, the NJBD for the New JBD algorithm using the Jacobi-like JD method and Pham's JBD (PJBD) algorithm. In this comparative study, three criteria are used to evaluate the estimation accuracy, the average convergence rate and the sensitivity to random initialization (i.e, convergence to local minima) respectively.

The quality of the JBD is evaluated by:

$$\rho = \sum_{k=1}^K \frac{\|\mathbf{M}_k - \text{bdiag}(\mathbf{M}_k)\|^2}{\|\text{bdiag}(\mathbf{M}_k)\|^2} \quad (32)$$

To measure the average convergence rate, we use the criterion,

$$\chi = \frac{1}{N_r} \sum_{r=1}^{N_r} I_r$$

where N_r is the number of successful Monte-Carlo runs ('successful' means that the algorithm has converged to a global minimum as explained below) and I_r is the number of iterations needed for convergence at the r -th run.

To decide whether an algorithm has converged or not, we use a threshold value, (typically $\epsilon_1 = 10^{-7}$) and decide convergence if during a sweep $|\theta_{qp}| \leq \epsilon_1$ for all (q, p) for the Jacobi-like algorithms and if $|h_{pq}| + |h_{qp}| \leq \epsilon_1$ for Pham's algorithm. Finally, to measure the sensitivity to local minima problem, we measure the rate, in percent, of successful convergence as,

$$\eta = \frac{\text{Number of successful convergence}}{\text{Total number of trials}} \times 100$$

To decide, in the noiseless case, whether the algorithm has been successful to achieve the JBD, we use, again, a threshold value ϵ_2 (typically $\epsilon_2 = 10^{-2}$) on the normalized JBD criterion i.e, we have a successful convergence if $\rho \leq \epsilon_2$.

All the statistics are evaluated over 500 Monte-Carlo runs. At each run, the matrices $\mathbf{M}_1, \dots, \mathbf{M}_K$ are generated randomly following the model in (1), except for Pham's algorithm where the matrices are generated randomly in such a way that \mathbf{E} is random (not necessarily unitary) but \mathbf{D}_{ki} , $k = 1, \dots, K$ and $i = 1, \dots, r$ are random positive definite matrices.

In the first experiment, we considered $K = 3$ matrices of size $n = 4$ and with block sizes $m_1 = m_2 = 2$. Figure 1

shows the convergence rate of the four JBD methods in the noiseless case. The plots represent the evolution of the averaged normalized JBD criterion (32) with respect to the number of iterations (sweeps). It can be observed that the NJBD is the fastest one. It converges, generally, for as few as 10 iterations. The other methods are, relatively, slow but converge after a larger number of iterations. Note that, the slowest one is the ALSBD. However, this algorithm has the lowest computational cost per iteration.

Figure 2 shows the performance accuracy for the four considered algorithms. The normalized JBD criterion (32) is plotted against the signal to noise ratio (SNR). The noise is introduced in the following way: the exact matrices \mathbf{M}_i , $i = 1, \dots, K$ are disturbed by random matrices with i.i.d. zero-mean gaussian entries of power,

$$\sigma^2 = \frac{\sum_{k=1}^K \|\mathbf{M}_k\|^2}{K n^2} \times 10^{-\frac{\text{SNR}}{10}}$$

where SNR is expressed in dB. It can be observed that, for low SNR, the best performance is obtained by PJBD, while, for moderate and high SNR, the NJBD is the best. However, except for the NJBD, these results are falsed by the fact that the criterion (32) is averaged over all Monte-Carlo trials including those where the JBD algorithms converge to local minima. As will be shown next, LSBSD, ALSBD and PJBD are very sensitive to the initialization point and thus suffer from ill convergence to local minima.

In the second experiment, we consider, in the noiseless

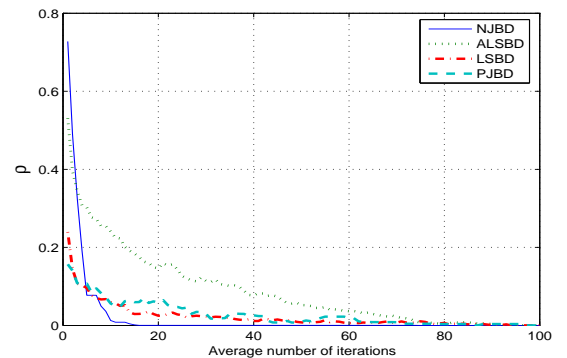


Fig. 1. JBD criterion versus the number of iterations in the noiseless case: $K = 3, n = 4$ and $r = 2$.

case, $K = 3$ matrices with block sizes $m_i = 2$ for all $i = 1, \dots, r$ and a number r of blocks varying in the range $[1 \dots 5]$. Figure 3 illustrates the problem of ill convergence due to random initialization. As can be observed for LSBSD, ALSBD and PJBD, the rate of successful convergence decreases rapidly toward zero when the number of blocks increases. The NJBD is the only algorithm that converges successfully almost all the time regardless of the matrix size. In figure 4, we plot the average number of iterations versus the number of elementary diagonal blocks. It is observed that the number of iterations needed for convergence increases significantly for the LSBSD and ALSBD algorithms, while it remains stable (almost invariant) for

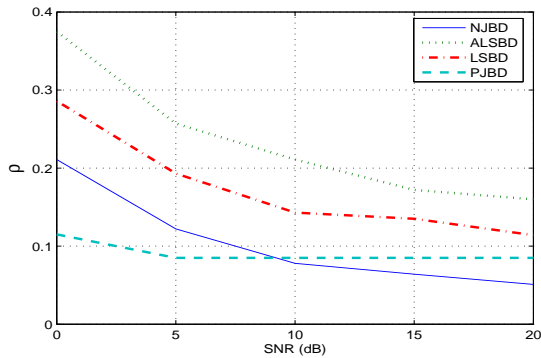


Fig. 2. JBD criterion versus the SNR: $K = 3$, $n = 4$ and $r = 2$.

the NJBD and PJBD. However, for the PJBD and ALSBD, the number of successful trials decreases rapidly and becomes zero when the number of blocks is equal to $r = 5$. The best convergence rate is obtained by the NJBD which converges, in average, after approximately 10 iterations.

A similar result is obtained in our third experiment for

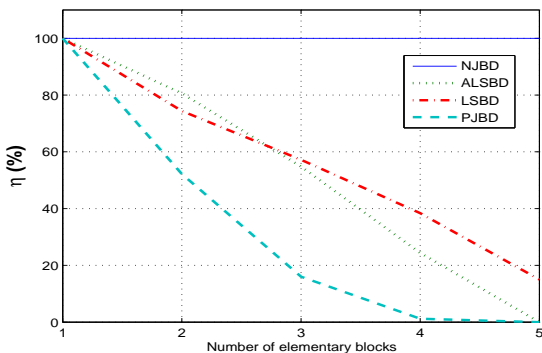


Fig. 3. Percentage of successful JBD rate vs blocks number: $K = 3$ and $r = 1, \dots, 5$.

the JBD of $K = 3$ matrices considered with a matrix size kept constant equal to $n = 12$ but the size of the diagonal blocks varies in the range $\{2, 3, 4, 6\}$. Figure 5 shows the rate of successful convergence for the NJBD and LSB with respect to the block size. Both the PJBD and ALSBD fail to converge to the desired solution and hence are not represented in the plot. For these algorithms, when the dimension of the considered matrices becomes large, a random initialization leads almost always to a local minimum convergence. On the other hand, this plot confirms the good behavior of the NJBD algorithm which still has a rate of successful convergence close to 100%.

Based on the previous results, we suggest to use the NJBD algorithm to get an initial solution of the JBD problem. This solution is used to initialize the other three algorithms that are applied here to refine the JBD of the considered set of matrices. Figure 6 compares the performance of the three algorithms ALSBD, LSB and PJBD initialized by the NJBD in terms of the JBD criterion in (32). Comparatively to the results of figure 2, one can observe that the

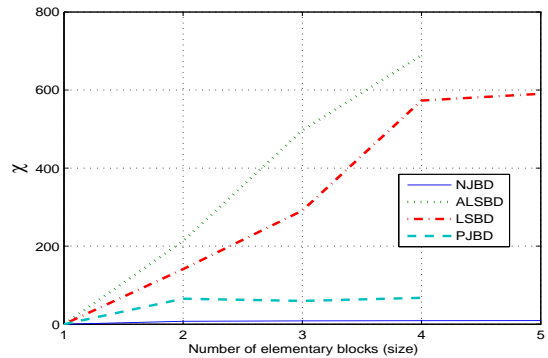


Fig. 4. Convergence rate versus blocks number: $K = 3$ and $r = 1, \dots, 5$.

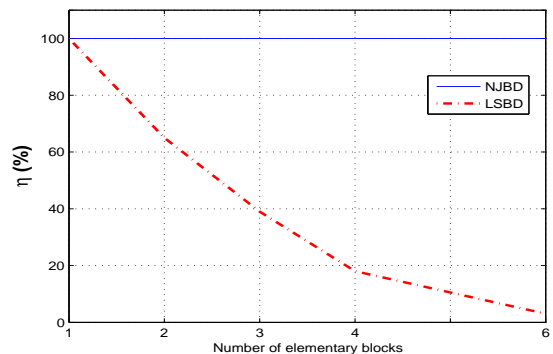


Fig. 5. Percentage of successful JBD rate vs blocks number: $K = 3$ and $n = 12$.

JBD criterion has been decreased by a factor of 5 to 10 approximatively. For low SNRs, the best result is obtained by the PJBD.

A study of the behavior of the proposed methods with respect to the number of matrices K to be jointly block-diagonalized is given in figures 7 and 8. We consider the JBD problem in the noiseless case with matrix size $n = 4$ and block size $m_1 = m_2 = 2$. One can see that the different methods are not affected when K varies except for the PJBD. As shown in figure 7, PJBD is less performing when the number of matrices to be jointly block-diagonalized is, relatively small. In that case, it is more sensitive to local minima problem.

B. Application to blind source separation

To illustrate the usefulness of the JBD method, we provide here an application example to blind source separation. Indeed, the JBD can be used either for the separation of independent groups of dependent (correlated) sources to reduce the overall interference to source ratio [7] or to separate convolutive mixtures [6]. It is this latter that we provide here for illustration by considering a discrete time

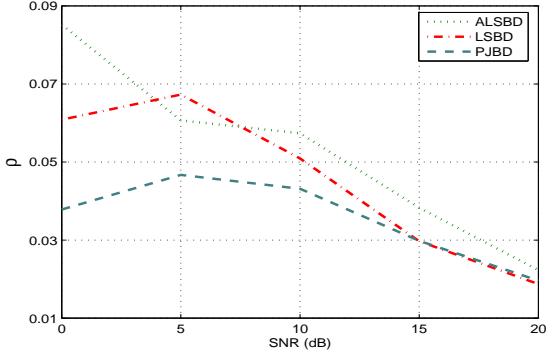


Fig. 6. JBD criterion versus the SNR with a prior NJBD: $K = 3$, $n = 4$ and $r = 2$.

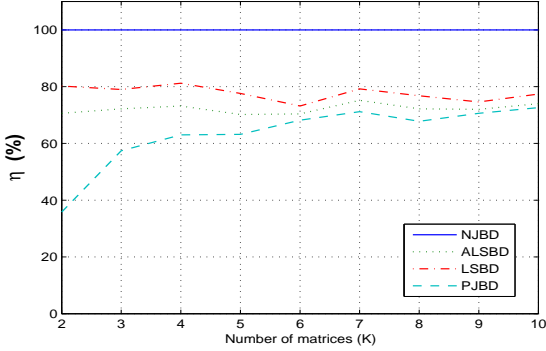


Fig. 7. Percentage of successful JBD rate versus the number of matrices K : $n = 4$ and $r = 2$.

MIMO linear time invariant system given by,

$$x_i(t) = \sum_{j=1}^m \sum_{l=0}^L h_{ij}(l) s_j(t-l) \quad (33)$$

where $s_j(t)$, $j = 1, \dots, m$, are the m source signals, $x_i(t)$, $i = 1, \dots, n$, are the n sensor signals ($n > m$), h_{ij} is the transfer function between the j -th source and the i -th sensor with a maximum overall extent $L + 1$. The source signals $s_j(t)$, $j = 1, \dots, m$, are mutually uncorrelated and each source signal is temporally coherent.

Equation (33) can be rewritten in the following matrix form,

$$\mathbf{x}(t) = \tilde{\mathbf{H}}\mathbf{s}(t) \quad \text{where,} \quad (34)$$

$$\mathbf{s}(t) = [s_1(t), \dots, s_1(t - (L + L') + 1), \dots, s_m(t - (L + L') + 1)]^T$$

$$\mathbf{x}(t) = [x_1(t), \dots, x_1(t - L' + 1), \dots, x_n(t - L' + 1)]^T$$

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \cdots & \tilde{\mathbf{H}}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{H}}_{n1} & \cdots & \tilde{\mathbf{H}}_{nm} \end{bmatrix} \quad \text{with,} \quad (35)$$

$$\tilde{\mathbf{H}}_{ji} = \begin{bmatrix} h_{ji}(0) & \cdots & h_{ji}(L) & \cdots & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & \cdots & h_{ji}(0) & \cdots & h_{ji}(L) \end{bmatrix} \quad (36)$$

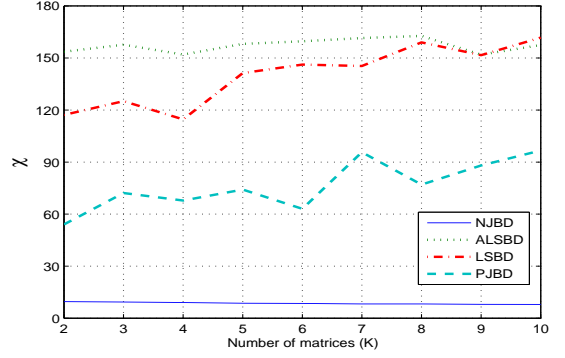


Fig. 8. Convergence rate versus the number of matrices K : $n = 4$ and $r = 2$

Note that $\tilde{\mathbf{H}}$ is a $nL' \times m(L + L')$ matrix (L' being a chosen window length parameter) and $\tilde{\mathbf{H}}_{ji}$ are $L' \times (L + L')$ matrices. $\tilde{\mathbf{H}}$ is assumed to be full column rank which is satisfied under certain diversity condition on the channels [21]. We define the data correlation matrices at time lag τ as:

$$\mathbf{R}_x(\tau) = E[\mathbf{x}(t)\mathbf{x}(t - \tau)^H] \quad (37)$$

Under the linear model (34), the above equation can be put in the following form,

$$\mathbf{R}_x(\tau) = \tilde{\mathbf{H}}\mathbf{R}_s(\tau)\tilde{\mathbf{H}}^H \quad \text{where,} \quad (38)$$

$$\mathbf{R}_s(\tau) = E[\mathbf{s}(t)\mathbf{s}(t - \tau)^H] \quad (39)$$

are the source correlation matrices at time lag τ . Since the source signals are mutually uncorrelated and temporally coherent, the matrix $\mathbf{R}_s(\tau)$ is block diagonal with m diagonal blocks each of size $(L' + L) \times (L' + L)$.

Herein, we show how the JBD can be exploited to separate the sources and recover them up to a filter. First, we start by whitening the data vector $\mathbf{x}(t)$. This is achieved by applying to $\mathbf{x}(t)$ a whitening matrix \mathbf{W} , i.e. a $m(L' + L) \times nL'$ matrix verifying:

$$E[\mathbf{W}\mathbf{x}(t)\mathbf{x}(t)^H\mathbf{W}^H] = \mathbf{W}\mathbf{R}_x(0)\mathbf{W}^H = (\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}})(\mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}})^H = \mathbf{I} \quad (40)$$

Equation (40) shows that if \mathbf{W} is a whitening matrix, then

$$\mathbf{E} = \mathbf{W}\tilde{\mathbf{H}}\mathbf{R}_s(0)^{\frac{1}{2}} \quad (41)$$

is a $m(L' + L) \times m(L' + L)$ unitary matrix where $\mathbf{R}_s(0)^{\frac{1}{2}}$, the Hermitian square root of $\mathbf{R}_s(0)$, is block diagonal. The whitening matrix \mathbf{W} can be determined from the eigen decomposition of the data correlation $\mathbf{R}_x(0)$ as in [3].

Consider now the spatially whitened correlation matrices $\underline{\mathbf{R}}_x(\tau)$ defined as

$$\forall \tau \neq 0 \quad \underline{\mathbf{R}}_x(\tau) = \mathbf{W}\mathbf{R}_x(\tau)\mathbf{W}^H. \quad (42)$$

By (38) and (41), we obtain the key relation:

$$\forall \tau \neq 0 \quad \underline{\mathbf{R}}_x(\tau) = \mathbf{E}\mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{R}_s(\tau)\mathbf{R}_s(0)^{-\frac{1}{2}}\mathbf{E}^H = \mathbf{E}\mathbf{D}(\tau)\mathbf{E}^H \quad (43)$$

where we have set $\mathbf{D}(\tau) = \mathbf{R}_s(0)^{-\frac{1}{2}} \mathbf{R}_s(\tau) \mathbf{R}_s(0)^{-\frac{1}{2}}$.

Since matrix \mathbf{E} is unitary and $\mathbf{D}(\tau)$ is block diagonal, the latter just means that any whitened data correlation matrix for any time lag τ is block diagonal in the basis of the column vectors of matrix \mathbf{E} . Hence, the separating matrix \mathbf{E} can be estimated by the JBD of a combined set of K matrices $\{\underline{\mathbf{R}}_x(\tau_k) | k = 1..K\}$. Once the unitary matrix \mathbf{E} is determined, the recovered signals are obtained up to a filter by

$$\hat{\mathbf{s}}(t) = \mathbf{E}^H \mathbf{W} \mathbf{x}(t) \quad (44)$$

Below, we present a simulation example of blind separation of speech signals using JBD. The parameter settings are as follows:

- $m = 2$, $n = 3$, $L = 2$ and $L' = 4$.
- The two speech signals are sampled at 8kHz.
- The transfer function matrix of the simulated channel is given by:

$$\mathbf{H}(z) = \begin{bmatrix} 1 & 0.85z^{-1} + 0.1z^{-2} \\ 0.7z^{-1} + 0.4z^{-2} & 0.25 + z^{-1} \\ 1 + 0.5z^{-1} + 0.7z^{-2} & 0.1z^{-1} + 0.85z^{-2} \end{bmatrix}$$

- $K = 3$ with $\tau_1 = 1$, $\tau_2 = 2$ and $\tau_3 = 3$

Figure 9 shows the original speech signal, their convolutive mixture and the recovered speech signals by the proposed algorithm. Clearly, the source separation is achieved correctly by the JBD of correlation matrices.

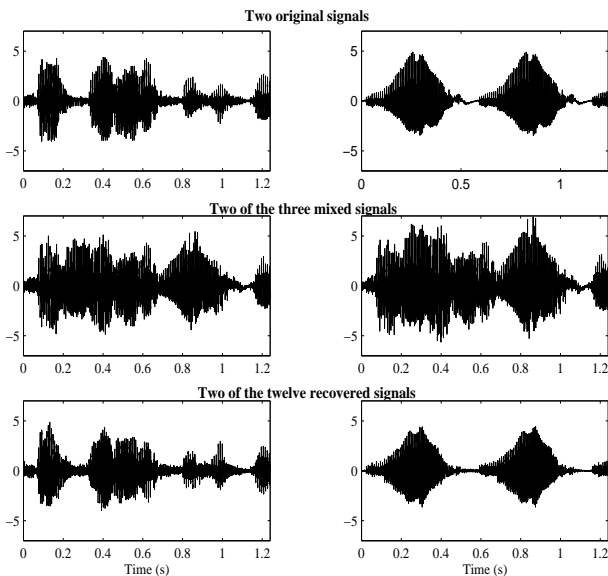


Fig. 9. Speech signal separation.

VII. CONCLUSION

In this paper, we have introduced a least squares approach for JBD. We have proposed new Jacobi-like algorithms (LSBD and ALSBD) that minimize the squared errors cost function iteratively by means of Givens rotations. The main advantages of the new algorithms are their inherent high parallelism, their robustness to noise and rounding errors, and their relative computational simplicity (at

least for the ALSBD). In addition, we have shown that the JBD can be achieved via JD algorithms up to a permutation matrix. We considered two JD methods and proposed solutions to estimate the permutation matrix. We obtained two new JBD algorithms, namely the PJBD algorithm for positive definite matrices and the NJBD. An exhaustive comparison between these four JBD methods has been undertaken. Based on this, we observed that the NJBD method is less sensitive to local minima problem while the other JBD methods are very sensitive to that problem but have better estimation accuracy. As a consequence, we recommend to use the NJBD to initialize one of the other JBD methods depending on the considered context. We have finally presented an application example of the JBD for blind source separation of convolutive mixtures.

APPENDIX A: PROOF OF (11)

Using equations (7) and the equalities $c^2 = (\cos(2\theta) + 1)/2$, $|s|^2 = (1 - \cos(2\theta))/2$, and $cs = \sin(2\theta)e^{i\alpha}/2$, we obtain (similar expressions are obtained when replacing p by q)

$$|\mathbf{M}'_k(p, j)|^2 - constant = -\mathbf{x}_{p,kj}^T \mathbf{v} \quad j \neq p \quad (45)$$

$$|\mathbf{M}'_k(j, p)|^2 - constant = -\mathbf{y}_{p,kj}^T \mathbf{v} \quad j \neq p \quad (46)$$

$$|\mathbf{M}'_k(p, p)|^2 - constant = \mathbf{v}^T \mathbf{G}_{p,k} \mathbf{v} + \mathbf{g}_{p,k}^T \mathbf{v} \quad (47)$$

where *constant* denotes terms independent from (θ, α) .

$$\begin{aligned} \mathbf{G} &= \sum_{k=1}^K (\mathbf{G}_{p,k} + \mathbf{G}_{q,k}) \\ \mathbf{g} &= \sum_{k=1}^K \left(\mathbf{g}_{p,k} + \mathbf{g}_{q,k} - \sum_{j \in \mathcal{I}_p, j \neq p} (\mathbf{x}_{p,kj} + \mathbf{y}_{p,kj}) \right. \\ &\quad \left. - \sum_{j \in \mathcal{I}_q, j \neq q} (\mathbf{x}_{q,kj} + \mathbf{y}_{q,kj}) \right) \end{aligned}$$

$(\mathbf{G}_{p,k}, \mathbf{G}_{q,k})$, $(\mathbf{g}_{p,k}, \mathbf{g}_{q,k})$, $(\mathbf{x}_{p,kj}, \mathbf{x}_{q,kj})$, and $(\mathbf{y}_{p,kj}, \mathbf{y}_{q,kj})$ are given by:

$$\begin{aligned} \mathbf{G}_{p,k} &= \Re(\mathbf{z}_{p,k} \mathbf{z}_{p,k}^H), & \mathbf{G}_{q,k} &= \Re(\mathbf{z}_{q,k} \mathbf{z}_{q,k}^H) \\ \mathbf{g}_{p,k} &= \Re((\overline{\mathbf{M}}_k(p, p) + \overline{\mathbf{M}}_k(q, q)) \mathbf{z}_{p,k}), \\ \mathbf{g}_{q,k} &= \Re((\overline{\mathbf{M}}_k(p, p) + \overline{\mathbf{M}}_k(q, q)) \mathbf{z}_{q,k}) \\ \mathbf{z}_{p,k} &= \frac{1}{2} \begin{bmatrix} \mathbf{M}_k(p, p) - \mathbf{M}_k(q, q) \\ -(\mathbf{M}_k(q, p) + \mathbf{M}_k(p, q)) \\ i(\mathbf{M}_k(p, q) - \mathbf{M}_k(q, p)) \end{bmatrix}, \\ \mathbf{z}_{q,k} &= \frac{1}{2} \begin{bmatrix} \mathbf{M}_k(q, q) - \mathbf{M}_k(p, p) \\ (\mathbf{M}_k(q, p) + \mathbf{M}_k(p, q)) \\ i(\mathbf{M}_k(p, q) - \mathbf{M}_k(q, p)) \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \mathbf{x}_{p,kj} &= -\mathbf{x}_{q,kj} = \begin{bmatrix} (|\mathbf{M}_k(q, j)|^2 - |\mathbf{M}_k(p, j)|^2)/2 \\ \Re(\mathbf{M}_k(p, j) \overline{\mathbf{M}}_k(q, j)) \\ \Im(\mathbf{M}_k(p, j) \overline{\mathbf{M}}_k(q, j)) \end{bmatrix} \\ \mathbf{y}_{p,kj} &= -\mathbf{y}_{q,kj} = \begin{bmatrix} (|\mathbf{M}_k(j, p)|^2 - |\mathbf{M}_k(j, q)|^2)/2 \\ \Re(\overline{\mathbf{M}}_k(j, p) \mathbf{M}_k(j, q)) \\ \Im(\overline{\mathbf{M}}_k(j, p) \mathbf{M}_k(j, q)) \end{bmatrix} \end{aligned}$$

In the case where \mathbf{M}_k are real valued matrices, vector \mathbf{v} becomes $\mathbf{v} = [\cos(2\theta), \sin(2\theta)]^T$ and matrix \mathbf{G} (resp. vector \mathbf{g}) is given by its top left 2×2 submatrix (resp. its top 2×1 subvector).

APPENDIX B: PROOF OF (19)

Using (45) and (46), we have

$$\tilde{\mathbf{g}} = - \sum_{k=1}^K \left(\sum_{j \in \mathcal{I}_p} (\mathbf{x}_{p,kj} + \mathbf{y}_{p,kj}) + \sum_{j \in \mathcal{I}_q} (\mathbf{x}_{q,kj} + \mathbf{y}_{q,kj}) \right)$$

To maximize (19), we express that its partial derivatives are null, i.e.,

$$\frac{\partial}{\partial \alpha} \mathcal{C}(\theta, \alpha) = \tilde{\mathbf{g}}^T \frac{\partial \mathbf{v}}{\partial \alpha} = 0, \quad \frac{\partial}{\partial \theta} \mathcal{C}(\theta, \alpha) = \tilde{\mathbf{g}}^T \frac{\partial \mathbf{v}}{\partial \theta} = 0 \quad (48)$$

$$\frac{\partial \mathbf{v}}{\partial \alpha} = \sin 2\theta \begin{bmatrix} 0 \\ -\sin \alpha \\ \cos \alpha \end{bmatrix}, \quad \frac{\partial \mathbf{v}}{\partial \theta} = 2 \begin{bmatrix} -\sin 2\theta \\ \cos 2\theta \cos \alpha \\ \cos 2\theta \sin \alpha \end{bmatrix} \quad (49)$$

Replacing (49) into (48) leads to

$$\begin{aligned} \tan \alpha &= \frac{\Im m(a)}{\Re e(a)} \iff \alpha = \arctan\left(\frac{\Im m(a)}{\Re e(a)}\right) \\ \tan 2\theta &= -2 \frac{\Re e(e^{-i\alpha} a)}{b} \\ \iff \theta &= \frac{1}{2} \arctan\left(-2 \frac{\Re e(e^{-i\alpha} a)}{b}\right) + k \frac{\pi}{2} \end{aligned}$$

where $k \in \{0, 1\}$ is chosen in such a way to maximize (19). Straightforward derivations lead to

$$\tilde{\mathbf{g}}^T \mathbf{v} = \cos 2\theta \times c$$

where $c = (b^2 + 4\Re e(e^{-i\alpha} a)^2)/(2b)$. Therefore, (19) is maximized if $\text{sign}(\cos 2\theta) = \text{sign}(c) = \text{sign}(b)$. Since $\arctan\left(-2 \frac{\Re e(e^{-i\alpha} a)}{b}\right) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, we finally obtain $k = (1 - \text{sign}(b))/2$.

APPENDIX C

We present below the pseudo-code that computes the permutation for a single $n \times n$ matrix \mathbf{M} that is block-diagonal up to an unknown permutation matrix,

```

% Thresholding value
eps=0.0001;
% Selection of non - zero entries
Mbis=abs(M).*(abs(M)>eps);
Mtest=Mbis>0;
% Calculation of the permutation
i=0; Sp=[1:n];
```

```

while i<n,
    ind=find(Mtest(Sp(1,:),:)>0);
    for k=1:length(ind);
        i=i+1;
        Sl(i)=ind(k);
        Ip=find(abs(Sp-ind(k))>0);
        Sp=Sp(Ip);
    end
end
% Application of the permutation
M=M(Sl,Sl);
```

REFERENCES

- [1] A. Belouchrani, K. Abed-Meraim, Y. Hua, "Jacobi-like Algorithms for Joint Block Diagonalization: Application to Source Localization", in *Proc. IEEE Int. Work. on Intelligent Sig. Proc. and Comm. Sys.*, Melbourne, 1998.
- [2] A. Belouchrani, K. Abed-Meraim, "Algorithms for Joint Block Diagonalization." *Proc. EUSIPCO 2004*, Vienna, Sept.2004.
- [3] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, E. Moulines, "Blind source separation using second order statistics." *IEEE-SP*, vol.45, no.2, pp. 434-444, Feb.1997.
- [4] Y. Hua, K. Abed-Meraim, "Techniques of eigenvalues estimation and association.", *Digital Signal Processing*, vol.7, no.4, Oct.1997.
- [5] A. Belouchrani, M. Amin, K. Abed-Meraim, "Direction finding in correlated noise fields based on joint block-diagonalization of spatio-temporal correlation matrices.", *IEEE-SP Let.*, Sep.1997.
- [6] H. Bousbia-Salah, A. Belouchrani, K. Abed-Meraim, "Jacobi-like Algorithm for Blind Signal Separation of Convolutional Mixtures.", *Electronics Letters*, vol.37, no.16, pp. 1049-1050, Aug.2001.
- [7] F. J. Theis, "Blind signal separation into groups of dependent signals using joint block diagonalization", *IEEE International Symposium on Circuits and Systems*, vol.6, pp. 5878-5881, May 2005.
- [8] M. Haardt, J.A. Nossék, "Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems", *IEEE Trans. Signal Process.*, pp. 161-169, vol.46, no.1, Jan.1998.
- [9] K. Abed-Meraim, Y. Hua, "A Least-Squares Approach to Joint Schur Decomposition", *Proc. ICASSP (Seattle, USA)*, IV, pp. 2541-2544, 1998 (see also book chapter in "Defence Applications of Signal Processing" edited by D. Cochran, W. Moran and L. B. White, Elsevier, pp. 1-14).
- [10] L. De Lathauwer, B. De Moor, J. Vandewalle, "Computation of the Canonical Decomposition by Means of a Simultaneous Generalized Schur Decomposition", *SIAM J. on Mat. Anal. and Appl.*, vol.26, no.2, pp.295-327, 2004.
- [11] Alle-Jan van der Veen, A. Paulraj, "An analytical constant modulus algorithm", *IEEE-SP*, pp. 1136-1155, May 1996.
- [12] H. Bousbia-Salah, A. Belouchrani, K. Abed-Meraim "Blind Separation of Non Stationary Sources using Joint Block Diagonalization." *The 11th IEEE Workshop on SSP*, Singapore, Aug.2001.
- [13] H.Buchner,R.Aichner,W.Kellermann,"A Generalization of Blind Source Separation Algorithms for Convolutional Mixtures Based On Second-Order Statistics", *IEEE-SAP*, vol.13, no.1, Jan.2005.
- [14] Q.H. Spencer, A.L. Swindlehurst, M. Haardt, "Zero-Forcing Methods for Downlink Spatial Multiplexing in Multiuser MIMO Channels", *IEEE-SP*, vol.52, no.2, pp. 461-471, Feb.2004.
- [15] P.J.Eberlein,"On one-sided Jacobi methods for parallel computation", *SIAM J. Alg. Disc. Meth.*, vol.8, pp. 790-796, 1987.
- [16] G. H. Golub, C. F. Van Loan, "Matrix computations", *The Johns Hopkins University Press*, 1989.
- [17] J.F.Cardoso, A.Souloumiac,"Jacobi angles for simultaneous diagonalization.",*SIAM J. Mat. Anal. Appl.*, vol.17, no.1, Jan.1996.
- [18] M. Wax, J. Scheinvald, "A least-squares approach to joint diagonalization", *IEEE-SP Letters*, vol.4, pp. 52-53, Feb.1997.
- [19] M. Haardt, "Efficient One-, Two-, and Multidimensional High-Resolution Array Signal Processing", *Shaker Verlag, Aachen*, 1997.
- [20] D.T. Pham, "Joint Approximate Diagonalization of Positive Definite Matrices", *SIAM Journal Mat. Anal. Appl.*, 22(4), pp. 1136-1152, 2001.
- [21] K. Abed-Meraim, Ph. Loubaton, E. Moulines, "A subspace algorithm for certain blind identification problems", *IEEE Tr. on IT.*, vol.43, pp.499-511, Mar.1997.