

LABORATOIRE



INFORMATIQUE, SIGNAUX ET SYSTÈMES
DE SOPHIA ANTIPOLIS
UMR 6070

IDENTIFICATION AVEUGLE DE MÉLANGES SOUS-DÉTERMINÉS

P. Comon, X. Luciani, A. Lima Ferrer de Almeida

Equipe SIGNAL

Rapport de recherche
ISRN I3S/RR-2008-09-FR

Mai 2008

Version révisée : Juillet 2008

RÉSUMÉ :

Le problème de l'identification aveugle de mélanges peut être abordé de deux façons. Soit on dispose d'une diversité suffisante pour construire un tenseur de mesures dont les tranches matricielles ne sont pas proportionnelles, auquel cas on se ramène à la décomposition d'un tenseur non symétrique en somme de tenseurs de rang 1. Soit ce n'est pas le cas, et on recourt à des approches statistiques, exploitant l'indépendance des variables aléatoires mélangées. Dans ce dernier cas, les équations de base sont fournies par la seconde fonction caractéristique conjointe, qui se décompose en combinaison linéaire de fonctions d'une seule variable. La résolution d'un tel problème se ramène à la décomposition d'un tenseur comportant des symétries en somme de tenseurs de rang 1. Nous proposons dans ce rapport plusieurs façons de construire ces tenseurs, et plusieurs algorithmes pour réaliser leurs décompositions.

MOTS CLÉS :

Fonction caractéristique, variables aléatoires indépendantes, tenseur, Parafac

ABSTRACT:

The problem of Blind Identification of Mixtures can be addressed in two manners. Either one can exploit some diversity in order to build a data tensor whose matrix slices are not proportional, in which case the problem amounts to decomposing a non symmetric tensor into a sum of rank-one tensors. Or this is not the case, and one resorts to statistical approaches based on the independence of random variables that are mixed. In the latter case, basic equations are given by the second joint characteristic function, which decomposes into a linear combination of functions of a single variable. Solving this problem can be carried out by decomposing a tensor enjoying symmetries into a sum of rank-one tensors. We propose in this report several ways of building such tensors, and various algorithms to compute their decompositions.

KEY WORDS :

Characteristic function, independent random variables, tensor, Parafac

Identification aveugle de mélanges sous-déterminés

P. Comon, X. Luciani, A. De Almeida

ISRN I3S/RR-2008-09-FR

Juillet 2008

1 Décompositions tensorielles

Les décompositions tensorielles apparaissent dans de nombreuses applications, notamment en télécommunications et en analyse de données. Contrairement à l'analyse matricielle, l'analyse tensorielle comporte encore de nombreux problèmes ouverts. Ce rapport n'aborde pas ces problèmes théoriques, dont certains sont évoqués dans [10] [38], mais est uniquement consacré aux aspects pratiques d'implantation numérique.

Avant d'aborder le problème des mélanges sous-déterminés, il est utile de définir les décompositions tensorielles existantes. Certaines d'entre elles ont été développées par nos soins en 2007. Introduisons tout d'abord quelques incontournables notations. Les scalaires seront notés en minuscules (a, b, \dots), les vecteurs en minuscules grasses ($\mathbf{a}, \mathbf{b}, \dots$), les matrices en capitales grasses ($\mathbf{A}, \mathbf{B}, \dots$) et les tenseurs en lettres calligraphiques ($\mathcal{A}, \mathcal{B}, \dots$). Les capitales italiques seront réservées pour les bornes de sommation ($i = 1, 2, \dots, I$). La composante de la ligne i et de la colonne j d'une matrice \mathbf{A} , i.e., $(\mathbf{A})_{ij}$, sera notée a_{ij} . De même nous aurons $(\mathcal{A})_{i_1 i_2 \dots i_N} = a_{i_1 i_2 \dots i_N}$. Les colonnes de \mathbf{A} sont notées $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_J$. Rappelons aussi la définition des produits de Kronecker \otimes et de Khatri-Rao \odot [43] :

$$\mathbf{A} \otimes \mathbf{H} \stackrel{\text{def}}{=} \begin{pmatrix} a_{11}\mathbf{H} & a_{12}\mathbf{H} & \cdots \\ a_{21}\mathbf{H} & a_{22}\mathbf{H} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix},$$
$$\mathbf{A} \odot \mathbf{H} \stackrel{\text{def}}{=} (\mathbf{a}_1 \otimes \mathbf{h}_1 \quad \mathbf{a}_2 \otimes \mathbf{h}_2 \quad \cdots).$$

La j ème colonne de $\mathbf{A} \odot \mathbf{B}$ est le produit $\mathbf{a}_j \otimes \mathbf{b}_j$ des j èmes colonnes de \mathbf{A} et \mathbf{B} . En outre, le produit dit de Hadamard entre deux matrices de mêmes dimensions, noté $\mathbf{A} \square \mathbf{B}$, représente le produit composante par composante dont l'élément générique est $A_{ij}B_{ij}$.

1.1 Généralités

1.1.1 Produit tensoriel

Soient $A_{i..j}$ et $B_{k..l}$ deux tableaux de valeurs, de dimensions quelconques. Le produit tensoriel entre \mathbf{A} et \mathbf{B} est le tableau \mathbf{C} dont les composantes sont :

$$C_{i..j k..l} = A_{i..j} B_{k..l}$$

Si \mathbf{A} et \mathbf{B} ont r_A et r_B indices, respectivement, alors \mathbf{C} a $r_A + r_B$ indices. On notera ce produit tensoriel (parfois appelé produit extérieur)

$$\mathbf{C} = \mathbf{A} \circ \mathbf{B} \tag{1.1}$$

Soit \mathbf{T} un tableau à L indices, de dimensions N_ℓ , $1 \leq \ell \leq L$. Ce tableau admet toujours une décomposition en somme de produits extérieurs de vecteurs :

$$\mathbf{T} = \sum_{r=1}^R \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \dots \circ \mathbf{u}_r^{(L)} \quad (1.2)$$

où $\mathbf{u}_r^{(\ell)}$ est un tableau $N_\ell \times 1$, $\forall r$. Cette écriture n'est pas unique, en particulier si rien n'est défini pour limiter la valeur de R .

Si \mathbf{T} prend ses valeurs dans un corps \mathbb{K} (les réels ou les complexes), les tableaux $\mathbf{u}_r^{(\ell)}$ peuvent être considérés comme des vecteurs de \mathbb{K}^{N_ℓ} . Donc en tant que combinaison linéaire de produits tensoriels, \mathbf{T} peut être considéré comme un *tenseur*. Si on effectue un changement linéaire de système de coordonnées dans l'espace \mathbb{K}^{N_ℓ} défini par une matrice $\mathbf{A}^{(\ell)}$, le tenseur est représenté par un autre tableau, qu'on obtient par transformée multilinéaire $\{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(L)}\}$ (voir le prochain paragraphe). Nous ne ferons donc plus la distinction entre un tenseur et sa représentation par un tableau de coordonnées. Le nombre d'indices nécessaire pour décrire les coordonnées d'un tenseur est l'*ordre* du tenseur. Un tenseur d'ordre L a par conséquent L *dimensions*, N_ℓ . Par exemple, une matrice est un tenseur d'ordre 2.

Le produit tensoriel entre deux tenseurs \mathbf{A} et \mathbf{B} est parfois noté $\mathbf{A} \otimes \mathbf{B}$; nous le noterons $\mathbf{A} \circ \mathbf{B}$ afin d'éviter une confusion. On peut en effet toujours stocker les coordonnées d'un tenseur dans une matrice. La représentation matricielle du produit tensoriel $\mathbf{A} \circ \mathbf{B}$ est alors le *produit de Kronecker* de leurs représentations matricielles, $\mathbf{A} \otimes \mathbf{B}$. Pour des tenseurs d'ordre plus grand que 2, il est donc souhaitable d'utiliser deux notations différentes.

En statistiques, les tenseurs sont souvent symétriques, car ils sont obtenus en prenant la dérivée Lième d'une fonction, par exemple une fonction caractéristique. Les moments et les cumulants l'ordre L d'une variable aléatoire de \mathbb{R}^N sont des exemples de tenseurs symétriques [40]. Dans le cas de variables aléatoires de \mathbb{C}^N , les symétries peuvent être complexes ou hermitiennes. Dans ce cas, il peut être intéressant de faire la distinction entre indices covariants et contravariants [31], car la conjugaison complexe peut être vue comme un passage dans le dual.

1.1.2 Transformation linéaire

Pour simplifier la présentation, on se limitera aux tenseurs d'ordre 3, même s'il a été nécessaire de recourir à des tenseurs d'ordre plus élevé dans nos développements. Le passage de l'ordre 3 à des ordres plus élevés est immédiat, et ne fait que compliquer les notations. Soit \mathbf{T} un tenseur d'ordre 3 de coordonnées T_{ijk} , $1 \leq i \leq I$, $1 \leq j \leq J$, $1 \leq k \leq K$. En d'autres termes, ce tenseur est de dimensions $I \times J \times K$.

Pour nous, un tenseur d'ordre 3 sera simplement un objet défini par ses IJK coordonnées, qui bénéficie de la propriété de *multi-linéarité* par transformation linéaire. Habituellement, cette propriété est présentée pour des changements de base, c'est à dire des transformations inversibles. Mais ceci représente une limitation inutile dans nos développements, et les transformations considérées pourront être définies par des matrices rectangulaires. Plus précisément, soient \mathbf{A} , \mathbf{B} et \mathbf{C} trois matrices de dimensions $I' \times I$, $J' \times J$, et $K' \times K$, respectivement. Alors le tenseur \mathbf{T} est transformé en un tenseur \mathbf{T}' donné par la propriété de multi-linéarité :

$$T'_{ijk} = \sum_{lmn} A_{il} B_{jm} C_{kn} T_{lmn} \quad (1.3)$$

1.1.3 Contraction

La contraction est l'opération qui consiste à sommer sur un des indices dans une expression. Par exemple, pour deux tenseurs \mathbf{A} et \mathbf{B} d'ordre α et β , ayant la dimension k en commun, on peut définir le tenseur $\mathbf{C} = \mathbf{A} \bullet_k \mathbf{B}$

d'ordre $\alpha + \beta - 2$:

$$C_{i_1 \dots i_\alpha j_1 \dots j_\beta} = \sum_{n_k=1}^{N_k} A_{i_1 \dots n_k \dots i_\alpha} B_{j_1 \dots n_k \dots j_\beta}$$

Le symbole \bullet_k représente la contraction selon le *mode* k du tenseur. La contraction permet de définir un produit scalaire entre deux tenseurs de mêmes dimensions. Par exemple, pour deux tenseurs d'ordre 3 et de dimensions $I \times J \times K$, nous avons :

$$\langle \mathbf{A}, \mathbf{B} \rangle \stackrel{\text{def}}{=} \mathbf{A} \underset{1}{\bullet} \underset{2}{\bullet} \underset{3}{\bullet} \mathbf{B}$$

Ce produit induit la norme de Frobenius :

$$\|\mathbf{A}\|^2 = \langle \mathbf{A}, \mathbf{A} \rangle = \sum_{ij..k} |A_{ij..k}|^2$$

cette notation est pratique, mais il faut cependant faire attention aux expressions faisant apparaître plusieurs contractions, comme :

$$\mathbf{A} \underset{1}{\bullet} \mathbf{B} \underset{2}{\bullet} \mathbf{C}$$

En effet, la contraction n'est pas associative. Il faut donc préciser l'ordre des contractions, ou adopter une convention. Une convention couramment admise est la suivante : lorsque un tenseur et plusieurs matrices sont impliquées dans plusieurs contractions, on associe à chaque contraction de mode k le k ème indice du tenseur, mais la contraction portera toujours sur le *deuxième indice* des matrices. Ainsi, la propriété de multi-linéarité (1.3) s'écrit sous forme compacte :

$$\mathbf{T}' = \mathbf{T} \underset{1}{\bullet} \mathbf{A} \underset{2}{\bullet} \mathbf{B} \underset{3}{\bullet} \mathbf{C}$$

Lorsque plusieurs tenseurs d'ordre supérieur à 2 sont impliqués dans une séquence de contractions, il est nécessaire de préciser dans quel ordre ces contractions doivent être exécutées, car il n'existe pas aujourd'hui de convention universellement admise.

1.1.4 Rang tensoriel

La décomposition (1.2) présente un grand intérêt lorsque R est minimal. Cette idée remonte au début du 19ème siècle par F.L.Hitchcock [10], mais a été indépendamment proposée dans des applications numériques par Carroll et Chang [7] et Harshman [26], qu'ils ont baptisée CANDECOMP et PARAFAC, respectivement. Dans ce rapport, nous adopterons l'acronyme CAND.

Etant donné un tenseur d'ordre 3, \mathbf{T} , de dimensions $I \times J \times K$, cette décomposition consiste à écrire (1.2) avec un nombre minimal de termes simples :

$$\mathbf{T} = \sum_{r=1}^{R(\mathbf{T})} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{1.4}$$

Le nombre minimal de termes nécessaires pour avoir égalité est noté $R(\mathbf{T})$, et sera appelé rang du tenseur \mathbf{T} . En d'autres termes, il existe trois matrices de dimension $I \times R$, $J \times R$, et $K \times R$, respectivement, telles que

$$T_{ijk} = \sum_{r=1}^{R(\mathbf{T})} A_{ir} B_{jr} C_{kr} \tag{1.5}$$

La définition aux ordres plus élevés est similaire.

1.1.5 Matrices dépliantes

Les matrices dépliantes sont une façon (parmi d'autres) de ranger les coordonnées d'un tenseur dans une matrice, et bien souvent une manière "compacte" d'écrire des équations sans faire figurer d'indices. On définit L matrices dépliantes associées à un tenseur d'ordre L . Prenons l'exemple du tenseur d'ordre 3 défini en (1.5). A k fixé, nous avons une matrice de taille $I \times J$ que nous pouvons noter $\mathbf{T}_{::k}$. La collection de ces K matrices peut être rangée dans une matrice de dimension $IK \times J$:

$$\mathbf{T}_{KI \times J} = \begin{bmatrix} \mathbf{T}_{::1} \\ \vdots \\ \mathbf{T}_{::k} \\ \vdots \\ \mathbf{T}_{::K} \end{bmatrix}$$

La définition pour les deux autres matrices dépliantes $\mathbf{T}_{IJ \times K}$ et $\mathbf{T}_{JK \times I}$, s'obtient par permutation circulaire des modes ; ces dernières contiennent les blocs de taille $J \times K$ et $K \times I$, respectivement. Avec ces définitions, il est possible de réécrire (1.5) de façon compacte. On trouve dans la littérature la terminologie de "unfolding" ou "flattening" matrices.

1.1.6 Autres modes de rangement

On utilise aussi fréquemment des représentations matricielles différentes des matrices dépliantes, lorsque les tenseurs sont d'ordre supérieur à trois. Ainsi, nous définissons la représentation d'un tenseur \mathcal{T} de dimensions $I \times J \times K \times L$ par une matrice la matrice \mathbf{T} de dimensions $IJ \times KL$ par :

$$\mathbf{T} = \text{mat}(\mathcal{T}) \Leftrightarrow T_{(i-1)J+j, (k-1)L+l} = (\mathcal{T})_{ijkl},$$

de même qu'une matrice $I \times J$ peut être stockée dans un vecteur de taille IJ :

$$\mathbf{m} = \text{vec}(\mathbf{M}) \Leftrightarrow m_{(i-1)J+j} = M_{ij}$$

Pour des matrices carrées, l'opération inverse peut être définie sans ambiguïté : $\mathbf{M} = \text{Unvec}(\mathbf{m})$. Pour des matrices rectangulaires, ou pour des tenseurs d'ordre supérieur à deux, il est nécessaire de spécifier les dimensions du tableau désiré (arguments d'entrée de l'application $\text{Unvec}(\cdot)$).

1.1.7 Propriétés

On peut alors montrer que (1.5) revient à :

$$\mathbf{T}_{KI \times J} = (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T \quad (1.6)$$

Par ailleurs, nous avons entre autres les identités utiles [43] :

$$\text{vec}(\mathbf{A} \mathbf{B} \mathbf{C}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B}) \quad (1.7)$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{A} \mathbf{C} \otimes \mathbf{B} \mathbf{D} \quad (1.8)$$

1.2 Décompositions sous contraintes

Dans le problème d'identification aveugle de mélanges sous déterminés par approches déterministes, il pourra être nécessaire de calculer la décomposition tensorielle suivante (cf. section 2.5.1) :

$$T_{ijk} = \sum_p \sum_q \sum_r A_{ipq} B_{jpr} C_{kpqr} \quad (1.9)$$

Cette décomposition est plus complexe que la CAND définie en (1.4).

1.2.1 Décomposition bloc-contrainte, BLOCFAC

Cette décomposition allie les propriétés de la CAND et Tucker3 [18]. Il s'agit de décomposer un tenseur d'ordre 3 en somme de K blocs structurés. Dans chaque bloc, des contraintes (ou *interactions*) impliquant les colonnes des trois matrices de la décomposition imposent une structure. La structure imposée dans chaque bloc est définie par deux *matrices de contraintes*, et pourrait varier d'un bloc à l'autre, mais nous supposons ici qu'il s'agit de la même structure pour simplifier. Pour un tenseur d'ordre trois $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$, cette décomposition est donnée par :

$$x_{i_1, i_2, i_3} = \sum_{q=1}^K \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} a_{i_1, r_1}^{(q)} b_{i_2, r_2}^{(q)} c_{r_1, r_2, i_3}^{(q)}. \quad (1.10)$$

x_{i_1, i_2, i_3} est décomposé en une somme de K blocs tri-linéaires, chaque bloc étant lui-même une somme de $R_1 R_2$ produits triples. Dans l'expression du tenseur, $a_{i_1, r_1}^{(q)}$ contribue R_2 fois, $b_{i_2, r_2}^{(q)}$ contribue R_1 fois, et $c_{r_1, r_2, i_3}^{(q)}$ contribue $R_1 R_2$ fois. Définissons deux ensembles de matrices :

$$\{\mathbf{A}^{(q)}\} \in \mathbb{C}^{I_1 \times R_1}, \quad \{\mathbf{B}^{(q)}\} \in \mathbb{C}^{I_2 \times R_2}$$

et un ensemble de K tenseurs d'ordre trois $\{\mathcal{C}^{(q)}\} \in \mathbb{C}^{R_1 \times R_2 \times I_3}$ dont les éléments sont décrits respectivement par :

$$a_{i_1, r_1}^{(q)} = [\mathbf{A}^{(q)}]_{i_1, r_1}, \quad b_{i_2, r_2}^{(q)} = [\mathbf{B}^{(q)}]_{i_2, r_2}, \quad c_{r_1, r_2, i_3}^{(q)} = [\mathcal{C}^{(q)}]_{r_1, r_2, i_3}.$$

Définissons aussi un ensemble de K matrices $\{\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(K)}\} \in \mathbb{C}^{I_3 \times R_1 R_2}$:

$$[\mathbf{C}^{(q)}]_{i_3, (r_1-1)R_2+r_2} = c_{r_1, r_2, i_3}^{(q)}, \quad q = 1, \dots, K, \quad (1.11)$$

où la q ième matrice $\mathbf{C}^{(q)}$ est liée au tenseur $\mathcal{C}^{(q)}$ par :

$$\mathbf{C}^{(q)} = [\text{vec}(\mathbf{C}_{\cdot\cdot 1}^{(q)T}) \cdots \text{vec}(\mathbf{C}_{\cdot\cdot I_3}^{(q)T})]^T, \quad (1.12)$$

Ici nous avons noté $\mathbf{C}_{\cdot\cdot i_3}^{(q)} \in \mathbb{C}^{R_1 \times R_2}$ la i_3 ième tranche matricielle de $\mathcal{C}^{(q)}$ selon sa troisième dimension, $i_3 = 1, \dots, I_3$. Les tranches matricielles $\mathbf{X}_{\cdot\cdot i_3} \in \mathbb{C}^{I_1 \times I_2}$ de \mathcal{X} peuvent être exprimées sous une forme de type CAND [21] :

$$\mathbf{X}_{\cdot\cdot i_3} = \sum_{q=1}^K (\mathbf{A}^{(q)} \otimes \mathbf{1}_{R_2}^T) D_{i_3}(\mathbf{C}^{(q)}) (\mathbf{1}_{R_1}^T \otimes \mathbf{B}^{(q)})^T. \quad (1.13)$$

En utilisant la propriété (1.7), il vient que :

$$\mathbf{A}^{(q)} \otimes \mathbf{1}_{R_2}^T = (\mathbf{A}^{(q)} \otimes \mathbf{1})(\mathbf{I}_{R_1} \otimes \mathbf{1}_{R_2}^T) = \mathbf{A}^{(q)} \underbrace{(\mathbf{I}_{R_1} \otimes \mathbf{1}_{R_2}^T)}_{\Psi^{(q)}} = \mathbf{A}^{(q)} \Psi^{(q)}, \quad (1.14)$$

et

$$\mathbf{1}_{R_1}^T \otimes \mathbf{B}^{(q)} = (\mathbf{1} \otimes \mathbf{B}^{(q)})(\mathbf{1}_{R_1}^T \otimes \mathbf{I}_{R_2}) = \mathbf{B}^{(q)} \underbrace{(\mathbf{1}_{R_1}^T \otimes \mathbf{I}_{R_2})}_{\Phi} = \mathbf{B}^{(q)} \Phi, \quad (1.15)$$

où

$$\Psi = \mathbf{I}_{R_1} \otimes \mathbf{1}_{R_2}^T, \quad \Phi = \mathbf{1}_{R_1}^T \otimes \mathbf{I}_{R_2} \quad (1.16)$$

sont les *matrices de contraintes* qui modélisent les combinaisons linéaires entre facteurs de modes différents (interactions). Elles ont respectivement les dimensions $R_1 \times R_1 R_2$ et $R_2 \times R_1 R_2$. Ces définitions permettent de réécrire (1.13) :

$$\mathbf{X}_{\cdot\cdot i_3} = \sum_{q=1}^K \mathbf{A}^{(q)} \Psi D_{i_3}(\mathbf{C}^{(q)}) (\mathbf{B}^{(q)} \Phi)^T. \quad (1.17)$$

Définissons à présent les matrices blocs :

$$\begin{aligned}\mathbf{A} &\stackrel{\text{def}}{=} [\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(K)}] \in \mathbb{C}^{I_1 \times KR_1} \\ \mathbf{B} &\stackrel{\text{def}}{=} [\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(K)}] \in \mathbb{C}^{I_2 \times KR_2} \\ \mathbf{C} &\stackrel{\text{def}}{=} [\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(K)}] \in \mathbb{C}^{I_3 \times KR_1 R_2}.\end{aligned}$$

et les *matrices de contraintes bloc-diagonales* :

$$\begin{aligned}\overline{\Psi} &\stackrel{\text{def}}{=} \text{blockdiag}(\Psi \cdots \Psi) = \mathbf{I}_K \otimes \Psi \quad (KR_1 \times KR_1 R_2) \\ \overline{\Phi} &\stackrel{\text{def}}{=} \text{blockdiag}(\Phi \cdots \Phi) = \mathbf{I}_K \otimes \Phi \quad (KR_2 \times KR_1 R_2).\end{aligned}\tag{1.18}$$

Avec ces définitions nous avons :

$$\begin{aligned}\mathbf{X}_1 &= (\mathbf{C} \odot (\mathbf{A}\overline{\Psi}))(\mathbf{B}\overline{\Phi})^T, \\ \mathbf{X}_2 &= ((\mathbf{A}\overline{\Psi}) \odot (\mathbf{B}\overline{\Phi}))\mathbf{C}^T, \\ \mathbf{X}_3 &= ((\mathbf{B}\overline{\Phi}) \odot \mathbf{C})(\mathbf{A}\overline{\Psi})^T.\end{aligned}\tag{1.19}$$

On notera que BLOCFAC (1.19) peut être interprétée comme un modèle CAND [26, 7] structuré, à travers les matrices $\mathbf{A}\overline{\Psi}$, $\mathbf{B}\overline{\Phi}$ et \mathbf{C} . On remarquera aussi que la structure intra-bloc de notre modèle bloc-contraint ressemble à celle de PARALIND (PARALLEL profiles with LINEar Dependencies) proposé dans [6]. Enfin, notre décomposition bloc-contrainte BLOCFAC peut être vue comme une décomposition en blocs [32, 33, 41]; voir aussi [34].

1.2.2 Décomposition CONFAC

La décomposition (1.9) apparaît comme un cas particulier d'une décomposition plus générale baptisée CONFAC [19] :

$$T_{i_1 i_2 i_3} = \sum_{f=1}^F \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} A_{i_1 r_1} B_{i_2 r_2} C_{i_3 r_3} \Psi_{r_1 f} \Phi_{r_2 f} \Omega_{r_3 f}\tag{1.20}$$

où Ψ , Φ , et Ω sont des matrices de contraintes ne contenant que des éléments de $\{0, 1\}$, et ayant un seul élément non nul par colonne. Une autre façon de voir cette décomposition est de la présenter comme une décomposition de type "Tucker3", c'est à dire faisant intervenir un tenseur cœur \mathbf{G} :

$$T_{i_1 i_2 i_3} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} A_{i_1 r_1} B_{i_2 r_2} C_{i_3 r_3} G_{r_1 r_2 r_3}$$

mais où ce dernier a la structure suivante :

$$G_{r_1 r_2 r_3} = \sum_{f=1}^F \Psi_{r_1 f} \Phi_{r_2 f} \Omega_{r_3 f}$$

1.3 Unicité

Lorsqu'on traite des mélanges sous déterminés, nous avons $F > \min(R_1, R_2, R_3)$ dans (1.20), et même généralement aussi $F \geq \max(R_1, R_2, R_3)$. De même dans la CAND, cela veut simplement dire que $R(\mathbf{T}) > \min(R_1, R_2, R_3)$, et même généralement aussi que $R(\mathbf{T}) \geq \max(R_1, R_2, R_3)$.

Nous savons déjà que la CAND (1.5) n'est pas unique, et qu'elle est définie à une permutation et un facteur d'échelle près pour chacune des matrices, pourvu que les effets de ces indéterminations se compensent. En clair, cela veut dire que si $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ est solution de (1.5), alors $(\mathbf{A}', \mathbf{B}', \mathbf{C}')$ est aussi solution avec $\mathbf{A}' = \mathbf{A} \Delta_A \Pi_A$, $\mathbf{B}' = \mathbf{B} \Delta_B \Pi_B$, $\mathbf{C}' = \mathbf{C} \Delta_C \Pi_C$, et $\Delta_A \Delta_B \Delta_C = \mathbf{I}$, où les matrices Δ sont diagonales et les matrices Π sont des permutations. En l'absence d'autres hypothèse supplémentaires, on ne peut garantir un meilleur résultat d'unicité. On qualifie souvent ce résultat d'unicité de "*unicité essentielle*".

1.3.1 Unicité de la CAND.

Une condition suffisante a été donnée par Kruskal [30], dont la démonstration a été améliorée et généralisée à des ordres plus élevés [45] [47] :

Génériquement, les matrices \mathbf{A} , \mathbf{B} , et \mathbf{C} sont de rang ligne plein (si $N \geq K$), mais aussi toute sous-matrice $N \times N$ sera de rang plein ; c'est le concept de k -rang [27, 30]. On peut alors dire que la décomposition CAND est génériquement essentiellement unique si

$$\min(I, R) + \min(J, R) + \min(K, R) \geq 2R + 2 \quad (1.21)$$

On notera que si le rang excède toutes les dimensions, cette condition suffisante revient à $I+J+K > 2R+2$. Par ailleurs, nous savons que cette condition n'est pas nécessaire à l'unicité essentielle. En effet, notre conjecture est que l'unicité essentielle est garantie avec probabilité 1 lorsque le ratio $IJK/(I+J+K-2)$ est entier [14] [15]. La démonstration de ce résultat est difficile, et fait l'objet d'études en cours. Il a fallu d'ailleurs plusieurs années pour prouver les théorèmes d'unicité dans le cas symétrique.

1.3.2 Unicité de la décomposition BLOCFAC.

Théorème 1 (condition nécessaire d'unicité) : *Considérons l'ensemble des représentations dépliées (1.19) pour BLOCFAC. Admettons que les matrices $\mathbf{A} \in \mathbb{C}^{I_1 \times R_1}$, $\mathbf{B} \in \mathbb{C}^{I_2 \times R_2}$ et $\mathbf{C} \in \mathbb{C}^{I_3 \times R_3}$ sont de rang plein. La condition suivante est alors nécessaire pour assurer l'unicité de la décomposition au sens des moindres carrés :*

$$\min \left(\left\lfloor \frac{I_1 I_2}{R_1 R_2} \right\rfloor, \left\lfloor \frac{I_1 I_3}{R_2} \right\rfloor, \left\lfloor \frac{I_2 I_3}{R_1} \right\rfloor \right) \geq K, \quad (1.22)$$

où $\lfloor x \rfloor$ désigne l'arrondi de x à l'entier le plus proche. Cette condition garantit l'unicité inter-blocs de BLOCFAC.

Ces matrices sont généralement de rang colonne plein [21].

Il existe en effet des matrices bloc-diagonales

$$\begin{aligned} \mathbf{T}_a &= \text{blockdiag}(\mathbf{T}_a^{(1)} \dots \mathbf{T}_a^{(K)}), \\ \mathbf{T}_b &= \text{blockdiag}(\mathbf{T}_b^{(1)} \dots \mathbf{T}_b^{(K)}), \\ \mathbf{T}_c &= \text{blockdiag}(\mathbf{T}_c^{(1)} \dots \mathbf{T}_c^{(K)}), \end{aligned} \quad (1.23)$$

qui satisfont :

$$(\mathbf{T}_a^{(q)} \otimes \mathbf{T}_b^{(q)})^{-1} = \mathbf{T}_c^{(q)\top}, \quad q = 1, \dots, K, \quad (1.24)$$

et telles que $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{T}_a$, $\tilde{\mathbf{B}} = \mathbf{B}\mathbf{T}_b$ et $\tilde{\mathbf{C}} = \mathbf{C}\mathbf{T}_c$ génèrent les mêmes matrices $\{\mathbf{X}_i\}_{i=1,2,3}$.

1.3.3 Unicité de la décomposition CONFAC.

Pour CONFAC, on parlera d'*unicité partielle* [48]. On dira que le triplet $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ est partiellement unique si un sous-ensemble des colonnes de ces matrices est essentiellement unique, alors que les autres colonnes ne le sont pas. Nous avons les résultats suivants :

- Si $\text{rank}\{\Psi\Phi^\top\} = \text{rank}\{\Omega\Psi^\top\} = R_1$, alors \mathbf{A} est essentiellement unique
- Si $\text{rank}\{\Phi\Omega^\top\} = \text{rank}\{\Psi\Phi^\top\} = R_2$, alors \mathbf{B} est essentiellement unique
- Si $\text{rank}\{\Omega\Psi^\top\} = \text{rank}\{\Phi\Omega^\top\} = R_3$, alors \mathbf{C} est essentiellement unique

1.4 Algorithme ALS

1.4.1 Calcul de la CAND

Le principe de l'algorithme des moindres carrés alternés est extrêmement simple. Pour toutes matrices \mathbf{A} et \mathbf{C} fixées, la norme de Frobenius $\|\mathbf{T}_{KI \times J} - (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T\|^2$ est une forme quadratique en \mathbf{B} . Son minimum est atteint en une matrice \mathbf{B} qui peut donc être calculée simplement. Il en est de même pour chacun des modes :

$$\begin{aligned}\mathbf{B}^T &= (\mathbf{C} \odot \mathbf{A})^\dagger \mathbf{T}_{KI \times J} \stackrel{\text{def}}{=} f_B(\mathbf{C}, \mathbf{A}) \\ \mathbf{C}^T &= (\mathbf{A} \odot \mathbf{B})^\dagger \mathbf{T}_{IJ \times K} \stackrel{\text{def}}{=} f_C(\mathbf{A}, \mathbf{B}) \\ \mathbf{A}^T &= (\mathbf{B} \odot \mathbf{C})^\dagger \mathbf{T}_{JK \times I} \stackrel{\text{def}}{=} f_A(\mathbf{B}, \mathbf{C})\end{aligned}\tag{1.25}$$

où \mathbf{M}^\dagger désigne la pseudo-inverse de \mathbf{M} . L'algorithme ALS consiste à calculer tour à tour les matrices \mathbf{A} , \mathbf{B} et \mathbf{C} , en exécutant au cours de l'itération n :

$$\mathbf{B}_{n+1} = f_B(\mathbf{C}_n, \mathbf{A}_n), \quad \mathbf{C}_{n+1} = f_C(\mathbf{A}_n, \mathbf{B}_{n+1}), \quad \mathbf{A}_{n+1} = f_A(\mathbf{B}_{n+1}, \mathbf{C}_{n+1}),$$

Il n'existe pas de preuve de convergence de cet algorithme, et le choix de la valeur initiale a une influence importante. Outre le problème des minima locaux, l'algorithme peut faire du sur place dans certains cas ; c'est pourquoi plusieurs accélérations ont été proposées [5] [42], sans qu'elles ne soient pour l'instant entièrement satisfaisantes.

1.4.2 Calcul de CONFAC

Si les matrices de contraintes sont connues, peu de choses diffèrent pour la décomposition CONFAC (1.20), qui peut être calculée en recourant au même algorithme ALS. La première matrice dépliant s'écrit :

$$\mathbf{T}_{KI \times J} = (\mathbf{C}\mathbf{\Omega} \odot \mathbf{A}\mathbf{\Psi})\mathbf{\Phi}^T\mathbf{B}^T$$

de sorte que l'itération ALS est du type :

$$\mathbf{B}_{n+1}^T = [(\mathbf{C}_n\mathbf{\Omega} \odot \mathbf{A}_n\mathbf{\Psi})\mathbf{\Phi}^T]^\dagger \mathbf{T}_{KI \times J}\tag{1.26}$$

On peut montrer que si

$$IK \geq R_2, \quad IJ \geq R_3, \quad JK \geq R_1\tag{1.27}$$

les matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ seront calculables grâce à l'algorithme ALS si les matrices $\mathbf{G}_1 \stackrel{\text{def}}{=} (\mathbf{\Omega} \odot \mathbf{\Psi})\mathbf{\Phi}^T$, $\mathbf{G}_3 \stackrel{\text{def}}{=} (\mathbf{\Phi} \odot \mathbf{\Omega})\mathbf{\Psi}^T$, et $\mathbf{G}_2 \stackrel{\text{def}}{=} (\mathbf{\Psi} \odot \mathbf{\Phi})\mathbf{\Omega}^T$ sont de rang colonne plein.

On notera que la condition d'unicité essentielle ne coïncide pas avec la condition de calculabilité par l'algorithme ALS. Le même constat peut être fait pour la CAND, ce qui souligne le caractère suffisant mais non nécessaire de la condition (1.27).

2 Identification aveugle de mélanges sous-déterminés

2.1 Introduction

Les algorithmes d'identification aveugle dans le cas sous-déterminé sont de deux types. Soit ils exploitent une diversité permettant le rangement des mesures dans un tenseur de données d'ordre au moins trois suivant un modèle multi-linéaire, dont les tranches matricielles ne sont pas proportionnelles. Ces conditions ne semblant pas être réunies dans cette étude, cette famille de techniques n'a pas été retenue. Soit ils exploitent des statistiques d'ordre supérieur à deux, et pour être plus précis, le tenseur cumulant dans la quasi-totalité des cas. Les algorithmes FOOBI, FOOBI2, et BIOME appartiennent à cette famille. Afin d'être innovant, il nous a

paru intéressant de sortir de cette limitation, et de proposer un outil statistique plus général que les cumulants. L'idée proposée dans [13] était d'utiliser la seconde fonction caractéristique conjointe des observations, et ses dérivées, en des points pouvant être différents de l'origine. Ceci a donné lieu à l'algorithme ALESCAF.

Nous allons donc comparer les quatre algorithmes d'identification aveugle, que sont FOOBI, FOOBI2, ALESCAF, et BIOME, en termes de performances, de complexité numérique, et de nombre maximal de sources identifiables.

Cependant, ALESCAF n'avait été implanté et testé que dans le cas réel et pour deux capteurs [13]. En outre, il n'avait été implanté qu'avec l'aide des moindres carrés alternés (ALS). Afin de pouvoir conduire une comparaison dans des conditions similaires, il a été nécessaire de réécrire entièrement ALESCAF. Nous résumons ci-après les problèmes rencontrés et la démarche suivie, avant de présenter les comparaisons.

Pour simplifier, on supposera donc dans toute cette section que les sources sont stationnaires. Le mélange est quant à lui instantané sous-déterminé, et défini par une matrice \mathbf{H} de dimension $N \times K$, avec $N < K$, de sorte que

$$\mathbf{y}(m) = \mathbf{H}\mathbf{s}(m), \quad 1 \leq m \leq M \quad (2.28)$$

Puisque le mélange est instantané et qu'aucune hypothèse n'est disponible concernant la dépendance éventuelle entre les échantillons successifs (connaissance de la couleur), l'indice m sera considéré comme un simple numéro de réalisation d'une variable aléatoire. Le but est d'identifier la matrice de mélange \mathbf{H} et, si possible, les réalisations successives du vecteur source $\mathbf{s}(m)$, $1 \leq m \leq M$, à partir des seules observations $\mathbf{y}(m)$.

Les hypothèses de travail sont donc les suivantes

- H1. aucune paire de colonnes de la matrice \mathbf{H} ne sont deux à deux colinéaires.
- H2. les sources s_k sont statistiquement indépendantes et non gaussiennes, de distributions (identiques ou non) inconnues
- H3. soit les distributions des sources sont indécomposables, soit le nombre de sources est connu.

Sans l'hypothèse H1, deux sources pourraient être portées par deux vecteurs colinéaires et ne seraient donc pas discernables d'une source unique égale à leur somme (ou n'importe quelle combinaison linéaire), de sorte que le modèle d'observation (2.28) ne serait pas identifiable, quel que soit l'algorithme utilisé. Sous les hypothèses H1 et H2, le mélange est théoriquement identifiable lorsque le nombre de sources K est connu ; ceci fait appel à des résultats anciens de statistiques sur les mélanges linéaires [28] [22]. Cependant, il s'agit de résultats non constructifs, dans le sens où aucun algorithme d'identification n'y était proposé. Nous développons dans cette section plusieurs algorithmes d'identification.

En ce qui concerne l'hypothèse H3, il est utile de préciser qu'une variable aléatoire a une distribution *décomposable* si celle-ci peut s'écrire comme la somme de deux autres variables statistiquement indépendantes. Si une source devait avoir une distribution décomposable, alors il serait nécessaire de connaître le nombre de sources pour que le mélange soit génériquement identifiable. Prenons un exemple. Si le nombre de sources n'est pas connu, on ne pourra jamais faire la distinction entre une source QPSK et deux sources BPSK mélangées avec les coefficients 1 et j , ou bien entre une source PAM4 et deux sources BPSK mélangées avec les coefficients 1 et 2.

2.2 Techniques tensorielles à base de fonction caractéristique

2.2.1 Position du problème dans le corps des complexes

Rappelons brièvement le principe. Le problème de l'identification aveugle de la matrice \mathbf{H} revient à décomposer la fonction caractéristique conjointe $\Psi_y(\mathbf{w})$ en somme de fonctions d'une seule variable :

$$\Psi_y(\mathbf{w}) = \sum_k \psi_k \left(\sum_n H_{nk} w_n \right)$$

Dans [13], la fonction caractéristique était définie pour des variables réelles par $\Psi_y(\mathbf{u}) \stackrel{\text{def}}{=} \log E\{\exp(j\mathbf{u}^T \mathbf{y})\}$. Comme le coefficient j (racine de -1) ne joue aucun rôle, il est plus simple d'utiliser les fonctions génératrices, définies dans le cas réel par $\Psi_y(\mathbf{u}) \stackrel{\text{def}}{=} \log E\{\exp(\mathbf{u}^T \mathbf{y})\}$. Si \mathbf{y} est complexe, il suffit de considérer la distribution conjointe des parties réelle et imaginaire, de sorte qu'on peut adopter la définition suivante de la seconde fonction génératrice (qui génère les cumulants) :

$$\Psi_y(\mathbf{w}) \stackrel{\text{def}}{=} \log E\{\exp(\Re\{\mathbf{w}^H \mathbf{y}\})\}$$

où cette fois $\mathbf{w} \in \mathbb{C}^N$. Comme une fonction réelle de la variable complexe n'est pas holomorphe, on doit plonger \mathbb{C}^N dans \mathbb{R}^{2N} afin de pouvoir calculer les dérivées successives. Pour ce faire, on pose

$$\mathbf{x} = \begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} \text{ et } \mathbf{u} = \begin{bmatrix} \mathbf{u}^R \\ \mathbf{u}^I \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \Re\{\mathbf{w}\} \\ \Im\{\mathbf{w}\} \end{bmatrix}$$

L'écriture équivalente de la fonction génératrice est alors :

$$\Psi_x(\mathbf{u}) \stackrel{\text{def}}{=} \log E\{\exp(\mathbf{u}^T \mathbf{x})\} \quad (2.29)$$

où $\mathbf{u} \in \mathbb{R}^{2N}$. Posons maintenant $\mathbf{H} = \mathbf{A} + j\mathbf{B}$, où \mathbf{A} et \mathbf{B} sont des matrices réelles, et posons aussi $\mathbf{F} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$.

L'équation coeur devient finalement, si les sources sont réelles :

$$\Psi_x(\mathbf{u}^R, \mathbf{u}^I) \stackrel{\text{def}}{=} \Psi_x(\mathbf{u}) = \sum_k \psi_k \left(\sum_n F_{nk} u_n \right) \stackrel{\text{def}}{=} \sum_k \psi_k \left(\sum_n A_{nk} u_n^R + B_{nk} u_n^I \right) \quad (2.30)$$

Formellement, le problème est exactement le même que dans le cas réel, car estimer la matrice réelle \mathbf{F} de dimensions $2N \times K$ revient à estimer la matrice complexe \mathbf{H} de dimensions $N \times K$. Par contre, il faut éviter d'identifier séparément les matrices \mathbf{A} et \mathbf{B} en raison de l'indétermination de permutation-échelle, inhérente au problème. Ce piège est évité en posant le problème de cette façon.

Remarquons que ces écritures sont valables si les sources sont réelles. Lorsque ce n'est pas le cas, la fonction génératrice ψ_k d'une source est une fonction de deux variables, les parties réelle et imaginaire, de sorte que l'équation coeur (2.30) se complique en :

$$\Psi_x(\mathbf{u}^R, \mathbf{u}^I) = \sum_k \psi_k \left(\sum_n A_{nk} u_n^R + B_{nk} u_n^I, \sum_n A_{nk} u_n^I - B_{nk} u_n^R \right) \quad (2.31)$$

Le nombre de dérivées de la fonction Ψ_x reste le même qu'en (2.30), mais l'équation tensorielle qui en résulte diffère, comme nous allons le voir dans la section 2.2.2.

A la lecture de la formule (2.31), on notera que l'approche précédente reste valable pour des mélanges réels ($\mathbf{B} = 0$) de sources complexes. Cela vient du fait que les deux arguments de ψ_k sont identiques. La portée pratique de ce résultat est toutefois limitée.

On notera enfin qu'il n'a pas été nécessaire de préciser l'expression exacte de la fonction génératrice marginale de chaque source, puisqu'elle est supposée inconnue a priori. Si elle était connue, il pourrait être utile d'utiliser sa définition : $\psi_k(v) \stackrel{\text{def}}{=} \log E\{\exp(\Re\{v^* s_k\})\}$.

2.2.2 Transformation du problème en décomposition tensorielle

On suppose maintenant que nous avons affaire à un mélange complexe de sources réelles (ou un mélange réel de sources complexes). Dans [13], il a été expliqué comment résoudre l'équation fonctionnelle dans le corps

des réels. L'idée était de remarquer qu'en dérivant r fois l'équation (2.30), on fait apparaître une équation entre tenseurs d'ordre r paramétrée par \mathbf{u} :

$$\frac{\partial^r \Psi_y(\mathbf{u})}{\partial u_i \partial u_j \dots \partial u_p} = \sum_{k=1}^K F_{ik} F_{jk} \dots F_{pk} \psi_k^{(3)} \left(\sum_q F_{qk} u_q \right)$$

Si on dispose des valeurs de ces dérivées en L points d'une grille, \mathbf{u}_ℓ , $1 \leq \ell \leq L$, alors nous avons une équation tensorielle d'ordre $r + 1$:

$$T_{ij\dots p}(\ell) = \sum_{k=1}^K F_{ik} F_{jk} \dots F_{pk} G_k(\ell) \quad (2.32)$$

où $G_k(\ell) \stackrel{\text{def}}{=} \psi_k(\sum_q F_{qk} u_q(\ell))$ sont les éléments d'une matrice $K \times L$ inconnue, que l'on pourra noter \mathbf{G} . La première difficulté consiste à estimer les quantités $T_{ij\dots p}(\ell)$ à partir des observations.

Les dérivées de la première fonction génératrice, $\Phi_x(\mathbf{u}) \stackrel{\text{def}}{=} E\{\exp(\mathbf{u}^T \mathbf{x})\}$ s'écrivent très simplement, grâce à la commutativité de la dérivation et de l'espérance mathématique :

$$D_{ij\dots p} \Phi \stackrel{\text{def}}{=} \frac{\partial^r \Phi_x}{\partial u_i \partial u_j \dots \partial u_p}(\mathbf{u}) = E\{x_i x_j \dots x_p \exp(\mathbf{u}^T \mathbf{x})\}$$

où D_i désigne l'opérateur de dérivation partielle par rapport à u_i . En remplaçant l'espérance mathématique par une moyenne empirique, nous avons un estimateur consistant des dérivées de Φ_x . Le calcul des dérivées de la seconde fonction génératrice est plus compliqué.

Pour mener ce calcul, on utilise ci-après la notation crochet de McCullagh, dans laquelle les termes se déduisant les uns des autres par permutation ne sont pas répétés [40], mais leur nombre est simplement signalé entre crochets. A titre d'exemple, la somme des trois termes ci-dessous s'écrirait simplement $[3]A_{ij} A_{kl}$:

$$A_{ij} A_{kl} + A_{ik} A_{jl} + A_{il} A_{jk} \stackrel{\text{def}}{=} [3]A_{ij} A_{kl}$$

Avec cette notation, nous donnons l'expression des six premières dérivées ci-dessous.

$$\begin{aligned} D_i \Psi &= \frac{1}{\Phi} D_i \Phi \\ D_{ij} \Psi &= \frac{1}{\Phi} D_{ij} \Phi - \frac{1}{\Phi^2} D_i \Phi D_j \Phi \\ D_{ijk} \Psi &= \frac{1}{\Phi} D_{ijk} \Phi - \frac{1}{\Phi^2} [3]D_{ij} \Phi D_k \Phi + \frac{2}{\Phi^3} D_i \Phi D_j \Phi D_k \Phi \\ D_{ijkl} \Psi &= \frac{1}{\Phi} D_{ijkl} \Phi - \frac{1}{\Phi^2} [4]D_{ijk} \Phi D_l \Phi - \frac{1}{\Phi^2} [3]D_{ij} \Phi D_{kl} \Phi + \frac{2}{\Phi^3} [6]D_{ij} \Phi D_k \Phi D_l \Phi \\ &\quad - \frac{3!}{\Phi^4} D_i \Phi D_j \Phi D_k \Phi D_l \Phi \\ D_{ijklm} \Psi &= \frac{1}{\Phi} D_{ijklm} \Phi - \frac{1}{\Phi^2} [5]D_{ijk} \Phi D_m \Phi - \frac{1}{\Phi^2} [10]D_{ijk} \Phi D_{lm} \Phi + \frac{2}{\Phi^3} [10]D_{ijk} \Phi D_l \Phi D_m \Phi \\ &\quad + \frac{2}{\Phi^3} [15]D_{ij} \Phi D_{kl} \Phi D_m \Phi - \frac{3!}{\Phi^4} [10]D_{ij} \Phi D_k \Phi D_l \Phi D_m \Phi + \frac{4!}{\Phi^5} D_i \Phi D_j \Phi D_k \Phi D_l \Phi D_m \Phi \\ D_{ijklmn} \Psi &= \frac{1}{\Phi} D_{ijklmn} \Phi - \frac{1}{\Phi^2} [6]D_{ijklm} \Phi D_n \Phi - \frac{1}{\Phi^2} [15]D_{ijkl} \Phi D_{mn} \Phi - \frac{1}{\Phi^2} [10]D_{ijk} \Phi D_{lmn} \Phi \\ &\quad + \frac{2}{\Phi^3} [15]D_{ijk} \Phi D_m \Phi D_n \Phi + \frac{2}{\Phi^3} [60]D_{ijk} \Phi D_{lm} \Phi D_n \Phi + \frac{2}{\Phi^3} [15]D_{ij} \Phi D_{kl} \Phi D_{mn} \Phi \\ &\quad - \frac{3!}{\Phi^4} [20]D_{ijk} \Phi D_l \Phi D_m \Phi D_n \Phi - \frac{3!}{\Phi^4} [45]D_{ij} \Phi D_{kl} \Phi D_m \Phi D_n \Phi \\ &\quad + \frac{4!}{\Phi^5} [15]D_{ij} \Phi D_k \Phi D_l \Phi D_m \Phi D_n \Phi - \frac{5!}{\Phi^6} [15]D_i \Phi D_j \Phi D_k \Phi D_l \Phi D_m \Phi D_n \Phi \end{aligned}$$

Nous avons donc 203 termes à l'ordre 6. Non seulement les expressions sont plus complexes pour la seconde fonction génératrice, mais aussi les estimateurs empiriques correspondants sont biaisés. Nous allons néanmoins les utiliser dans ce rapport, et examinerons la correction du biais ultérieurement. Le tenseur des dérivées de Ψ est donc obtenu à partir de celui de Φ , et ce, en n'importe quel point $\mathbf{u}(\ell)$.

Dans le cas de sources complexes, le principe reste le même mais les relations sont sensiblement plus compliquées. Tout se passe comme si on remplaçait la matrice

$$\mathbf{F} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \text{ par la matrice } \mathbf{F} = \begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix}$$

de sorte que le nombre de composantes de la matrice inconnue augmente, mais que le nombre de variables libres reste le même. Nous ne développons pas ici ces relations.

2.2.3 Construction du tenseur à décomposer

Supposons que l'on ait calculé les dérivées de Ψ d'ordre r en L points. Nous avons donc un tenseur d'ordre $r+1$. Plusieurs possibilités s'offrent à nous. Soit nous décomposons le tenseur d'ordre $r+1$, soit nous le rangeons dans un tenseur d'ordre inférieur et nous décomposons ce dernier, ce qui revient à faire une *décomposition conjointe de plusieurs tenseurs* d'ordre inférieur. Si les décompositions sont exactes, alors les deux sont nécessairement équivalentes. En revanche, si le rang est trop grand pour qu'une des deux (ou les deux) décompositions puissent être exacte, alors les décompositions donneront des résultats différents.

On peut alors se demander quel intérêt il y aurait à ranger un tenseur d'ordre $r+1$ dans un tenseur d'ordre inférieur, puisque ce dernier a un rang générique inférieur, bien qu'une de ses dimensions soit supérieure. L'avantage réside dans la possibilité de décomposer conjointement des tenseurs d'ordres différents. Par exemple, les dérivées d'ordre 4 calculées en L points peuvent être rangées dans un tenseur $N \times N \times N \times NL$, et celles d'ordre 3 calculées en L' points dans un tenseur $N \times N \times N \times L'$. Ces deux tenseurs sont du même ordre et partagent la même matrice \mathbf{F} sur les trois premiers modes. Il est donc judicieux de les empiler selon le quatrième mode, et de décomposer un tenseur unique de dimensions $N \times N \times N \times NL+L'$.

Le nombre de points L ainsi que leur valeur ont une influence sur les performances des algorithmes de décomposition que nous avons utilisé. Dans la plupart des cas il semble que 3 ou 4 points de calcul des dérivées suffisent pour assurer la convergence. Cependant, nous avons choisi ici d'utiliser $L = 8$ points en raison de la difficulté de certaines conditions de test (sous détermination ou bruit élevé, faible taille de bloc). Ces L points sont tirés aléatoirement, chacun dans le pavé $[-1;1]^N$. Cette configuration semble donner les meilleurs résultats dans l'ensemble des tests. Néanmoins, une étude systématique de l'influence de ces paramètres devrait être menée afin de déterminer le meilleur choix s'il existe.

2.2.4 Optimisation

Une fois le tenseur construit, regroupant éventuellement des dérivées d'ordres différents, il s'agira de calculer les matrices \mathbf{F} et \mathbf{G} minimisant un critère polynomial du type

$$\varepsilon = \sum_{ij..k} \sum_{\ell} |T_{ij..k}(\ell) - \sum_q F_{iq} F_{jq} \dots F_{kq} G_q(\ell)|^2 \quad (2.33)$$

Cette minimisation peut se faire de nombreuses manières. Posons

$$\mathbf{p} = \begin{bmatrix} \text{vec}\mathbf{F}^T \\ \text{vec}\mathbf{G}^T \end{bmatrix}$$

Pour exécuter cette minimisation, nous avons implanté les algorithmes suivants :

- **ALS asymétrique** (une seule des variables \mathbf{F} apparaissant plusieurs fois dans le critère est mise à jour à chaque itération) :

$$\mathbf{F}(k+1) = \min_{\mathbf{F}} \|\mathbf{T} - \mathbf{F} \bullet_2 \mathbf{F}(k) \bullet_2 \dots \mathbf{F}(k) \bullet_2 \mathbf{G}\|^2$$

ce qui permet une solution algébrique puisque le degré en \mathbf{F} est alors 2.

- **ALS symétrique** Nous pouvons envisager au moins deux façons de forcer la symétrie à chaque itération. Par exemple, pour le tenseur des dérivées d'ordre 4 :

1. Forçage souple : $\mathbf{F}(k+1) = \mathbf{T}_{N \times N^3 L} \cdot \{(\mathbf{F}(k) \odot \mathbf{F}(k-1) \odot \mathbf{F}(k-2) \odot \mathbf{G}(k))^\top\}^{-1}$

2. Forçage dur : $\mathbf{F}(k+1) = \mathbf{T}_{N \times N^3 L} \cdot \{(\mathbf{F}(k) \odot \mathbf{F}(k) \odot \mathbf{F}(k) \odot \mathbf{G}(k))^\top\}^{-1}$

Le forçage dur n'est recommandé qu'au voisinage de la convergence.

Il serait aussi possible de traiter la symétrie sur deux modes à la fois, comme cela est expliqué dans [9], mais notre expérience nous conduit à penser qu'un tel algorithme ne fonctionne en pratique que dans le cas de mélanges sur-déterminés, cadre dans lequel l'idée a été introduite [50]. Dans le cas présent, aucune de ces deux approches n'a permis d'améliorer les résultats de l'ALS asymétrique.

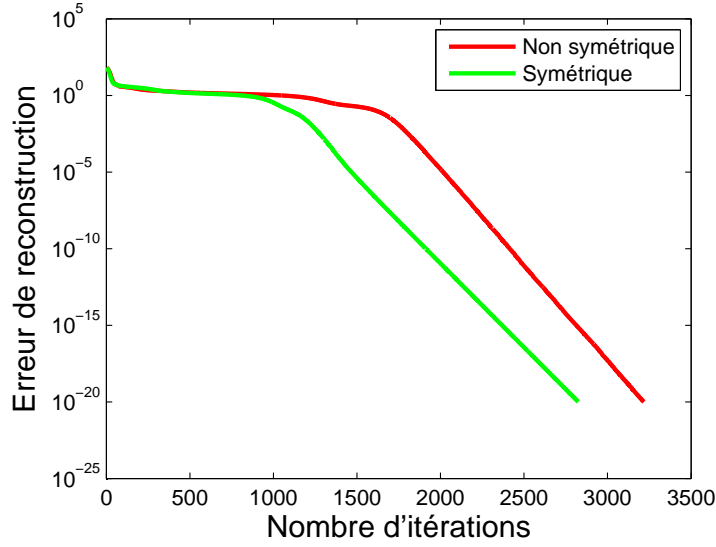


FIG. 1 – Convergences de l'algorithme de descente du gradient symétrique et non symétrique dans le cas d'un tenseur d'ordre 3 présentant une symétrie sur 2 des 3 modes.

- **Descente du gradient** à pas variable systématique, prenant en compte la symétrie éventuelle :

$$\mathbf{p}(k+1) = \mathbf{p}(k) - \mu(k) \mathbf{g}(k)$$

où μ est maintenu constant si $\varepsilon(k) - \varepsilon(k+1) > 0.005\varepsilon(k)$, est augmenté par la règle $\mu(k+1) = 1.1\mu(k)$ si $0 \leq \varepsilon(k) - \varepsilon(k+1) \leq 0.005\varepsilon(k)$, et est diminué par la règle $\mu(k+1) = \mu(k)/2$ si $\varepsilon(k) < \varepsilon(k+1)$. Comme cela a été vérifié par des simulations, cet algorithme n'est guère utilisable dans notre problème, compte-tenu de la présence de bassins très anisotropes et de plateaux. Nous verrons également que cet algorithme simple converge très lentement. Par ailleurs, la version symétrique de la descente du gradient (paramétrisation symétrique du problème) donne des résultats contrastés suivant l'ordre du tenseur comme l'illustrent les figures 1 et 2, représentatives du comportement de l'algorithme symétrique aux ordres 3 et 4 respectivement.

- **Descente de Levenberg-Marquardt**, prenant en compte la symétrie éventuelle :

$$\mathbf{p}(k+1) = \mathbf{p}(k) - [\mathbf{J}(k)^\mathbf{H} \mathbf{J}(k) + \lambda(k) \mathbf{I}]^{-1} \mathbf{g}(k)$$

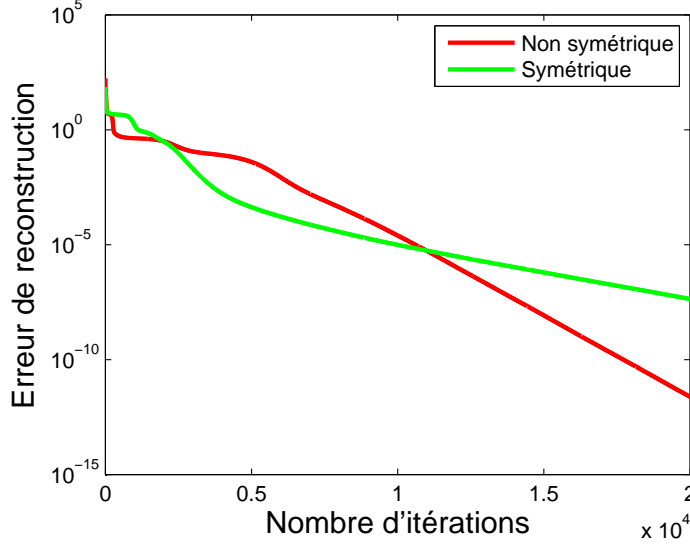


FIG. 2 – Convergences de l’algorithme de descente du gradient symétrique et non symétrique dans le cas d’un tenseur d’ordre 4 présentant une symétrie sur 3 des 4 modes.

où $\mathbf{J}(k)$ désigne le Jacobien et $\lambda(k)$ un facteur de régularisation. Il existe dans la littérature de nombreuses méthodes pour mettre à jour le paramètre de régularisation λ , généralement en fonction de la valeur du gain : $\gamma \stackrel{\text{def}}{=} |\varepsilon(k+1) - \varepsilon(k)| \cdot |\hat{\varepsilon}(k+1) - \varepsilon(k)|^{-1}$, où $\hat{\varepsilon}(k+1)$ désigne la prédiction du second ordre de l’erreur ε . Nous avons utilisé l’approche classique suivante [39] :

- Si $\gamma \geq 0$, alors : mise à jour de \mathbf{p} , $\lambda(k+1) = \lambda(k) \max(\frac{1}{3}, 1 - (2\gamma - 1)^3)$ et $\nu = 2$,
- sinon, $\lambda(k+1) = \nu\lambda(k)$ et $\nu = 2\nu$.

Il est également possible de ne tenir compte du gain que pour la mise à jour de λ , autrement dit, de mettre à jour le vecteur paramètre \mathbf{p} quelle que soit la valeur de γ . Appliqué à la minimisation de (2.33), cette stratégie donne généralement de très bons résultats en terme de nombre d’itérations nécessaires. Cependant, dans certains cas, un mauvais conditionnement peut entraîner la divergence de l’algorithme. Par conséquent nous avons préféré la version précédente, beaucoup plus stable.

Cet algorithme est assez populaire car il présente un bon compromis complexité-performances. Mais on lui préfère généralement le gradient conjugué pour traiter les problèmes mal conditionnés (mauvais conditionnement du hessien en plusieurs zones de la fonction objectif). Par ailleurs, nous avons rencontré avec la version symétrique de Levenberg-Marquardt les mêmes difficultés qu’avec la descente du gradient. Enfin, nous testons actuellement l’application à (2.33) d’autres variantes classiques de la méthode de Newton.

- **Contraintes sur les matrices de la décomposition.** Dans certaines applications les matrices sont cherchées dans un cône, dans une variété paramétrée ou dans un sous-espace connu. C’est le cas notamment des spectres de fluorescence, ou des mélanges convolutifs en télécommunications. Comme pour le forçage de la symétrie de la décomposition, les contraintes peuvent être dures (imposées exactement à chaque itération) ou souples. Prenons l’exemple d’une matrice \mathbf{F} à éléments tous positifs, et désignons par \mathbf{F}_c la matrice calculée sans contrainte à l’itération k , et \mathbf{F}_o celle obtenue en mettant à zéro tous ses éléments négatifs. Alors l’algorithme par contrainte dure consiste à prendre $\mathbf{F}(k+1) = \mathbf{F}_o$, tandis que par contrainte souple on peut prendre $\mathbf{F}(k+1) = \alpha \mathbf{F}_o + (1 - \alpha) \mathbf{F}_c$, pour un réel α de l’intervalle $[0, 1]$. En outre, ce nombre α peut être choisi variable au cours de la convergence, commençant autour de 0 et finissant autour de 1.

Nous donnons ci-dessous les expressions compactes du gradient et du Jacobien du critère, nécessaires pour les algorithmes de descente ci-dessus. Pour plus de lisibilité, on considère le cas d’un tenseur asymétrique d’ordre

3 et de dimensions $I \times J \times K$. Le gradient du critère $\varepsilon = \|\mathbf{T} - \mathbf{A} \bullet_2 \mathbf{B} \bullet_2 \mathbf{C}\|^2$ par rapport à la matrice \mathbf{A} est donné par :

$$\mathbf{g}_A = [\mathbf{I}_A \otimes (\mathbf{C}^H \mathbf{C} \boxminus \mathbf{B}^H \mathbf{B})] \text{vec} \mathbf{A}^T - [\mathbf{I}_A \otimes (\mathbf{C} \odot \mathbf{B})] \text{vec} \mathbf{T}_{KJ \times I}$$

et les autres gradients se déduisent par permutation circulaire :

$$\begin{aligned} \mathbf{g}_B &= [\mathbf{I}_B \otimes (\mathbf{A}^H \mathbf{A} \boxminus \mathbf{C}^H \mathbf{C})] \text{vec} \mathbf{B}^T - [\mathbf{I}_A \otimes (\mathbf{A} \odot \mathbf{C})] \text{vec} \mathbf{T}_{IK \times J} \\ \mathbf{g}_C &= [\mathbf{I}_C \otimes (\mathbf{B}^H \mathbf{B} \boxminus \mathbf{A}^H \mathbf{A})] \text{vec} \mathbf{C}^T - [\mathbf{I}_C \otimes (\mathbf{B} \odot \mathbf{A})] \text{vec} \mathbf{T}_{JI \times K} \end{aligned}$$

Si on pose

$$\mathbf{p} = \begin{bmatrix} \text{vec} \mathbf{A}^T \\ \text{vec} \mathbf{B}^T \\ \text{vec} \mathbf{C}^T \end{bmatrix}, \quad \mathbf{J} = [\mathbf{J}_A, \mathbf{J}_B, \mathbf{J}_C] \quad \text{et} \quad \mathbf{g} = \begin{bmatrix} \mathbf{g}_A \\ \mathbf{g}_B \\ \mathbf{g}_C \end{bmatrix}$$

alors nous avons les formes compactes suivantes pour les matrices jacobiniennes :

$$\begin{aligned} \mathbf{J}_A &= \mathbf{I}_A \otimes (\mathbf{C} \odot \mathbf{B}) \\ \mathbf{J}_B &= \Pi_1 [\mathbf{I}_B \otimes (\mathbf{A} \odot \mathbf{C})] \\ \mathbf{J}_C &= \Pi_2 [\mathbf{I}_C \otimes (\mathbf{B} \odot \mathbf{A})] \end{aligned}$$

où Π_i sont des permutations permettant de faire figurer les éléments dans le bon ordre.

Dans le cas où il existe une symétrie partielle, telle que $\mathbf{A} = \mathbf{B}$, alors les résultats précédents s'appliquent en posant

$$\mathbf{p} = \begin{bmatrix} \text{vec} \mathbf{A}^T \\ \text{vec} \mathbf{C}^T \end{bmatrix}, \quad \mathbf{J} = [\mathbf{J}_A + \mathbf{J}_B, \mathbf{J}_C] \quad \text{et} \quad \mathbf{g} = \begin{bmatrix} \mathbf{g}_A + \mathbf{g}_B \\ \mathbf{g}_C \end{bmatrix}$$

avec $\mathbf{B} = \mathbf{A}$.

La figure 3 compare les trois algorithmes pour un tenseur non symétrique, de dimensions $3 \times 3 \times 3 \times 3 \times 3$ et de rang 5. L'initialisation est la même pour les trois algorithmes. Malgré l'influence de la dimension du tenseur et de l'initialisation sur le comportement des algorithmes, cette figure est représentative des vitesses de convergence.

2.2.5 Problèmes de convergence et pas optimal

L'introduction de méthodes d'optimisation du second ordre, basées sur les dérivées premières de l'erreur, ou de méthodes basées sur le hessien, ne règlent pas le problème de convergence vers des minima locaux.

C'est pourquoi d'autres algorithmes sont à l'étude et seront testés dans un futur proche. L'un d'entre eux, que nous avons baptisé ELS (Enhanced Line Search) consiste à calculer le pas conduisant au *minimum absolu* le long de la direction de recherche.

Les simulations que nous avons effectuées par le passé [25] [51] ont montré que cette technique permet de sortir des minima locaux dans la plupart des cas. Elle se résume simplement comme suit. Soit $\Delta \mathbf{p}(k)$ la direction obtenue dans l'algorithme itératif considéré, quel qu'il soit. Au lieu d'exécuter l'itération $\mathbf{p}(k+1) = \mathbf{p}(k) + \Delta \mathbf{p}(k)$, on exécutera $\mathbf{p}(k+1) = \mathbf{p}(k) + \mu \Delta \mathbf{p}(k)$ après avoir cherché la meilleure valeur du pas μ , telle que le polynôme de degré $2d$ en μ suivant soit minimal :

$$\varepsilon(\mathbf{p}(k) + \mu \Delta \mathbf{p}(k))$$

Cela conduit à chercher toutes les racines d'un polynôme de degré $2d - 1$, puis à sélectionner celle donnant le minimum global de ε . Dans le cas du modèle CAND d'ordre d , il est plus simple de séparer les différentes

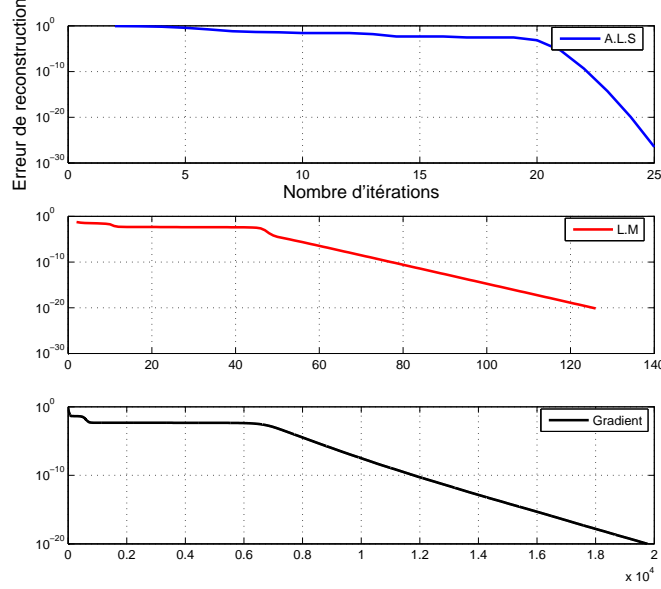


FIG. 3 – Comparaison de la vitesse de convergence des algorithmes ALS, LM et gradient.

composantes du vecteur paramètre. Ainsi, en notant $A_1 \cdots A_d$ les matrices de la décomposition, le polynôme en μ à minimiser est le suivant :

$$\|\mathbf{T} - (\mathbf{A}_1 + \mu\Delta_{\mathbf{A}_1}) \left(\bigodot_{p=2}^d (\mathbf{A}_p + \mu\Delta_{\mathbf{A}_p}) \right)^T\|^2$$

La complexité est dominée par le calcul des coefficients de ce polynôme, et est négligeable sauf si le rang du tenseur est faible devant ses dimensions [42, section A.1]. Cet algorithme a été codé pour des tenseurs d'ordre quelconque et combiné aux trois algorithmes précédents, fournissant chacun une direction de recherche à chaque itération. Il n'est pas forcément nécessaire d'effectuer systématiquement le calcul du pas optimal. Généralement l'optimisation est réalisée périodiquement toutes les n itérations. En pratique, les meilleurs résultats (au sens du compromis vitesse/convergence/coût calcul) ne sont pas toujours atteints pour la période la plus petite ($n = 1$).

- **ALS.** La direction de recherche est simplement déterminée par la différence entre les paramètres estimés lors de deux itérations successives $\Delta\mathbf{p}(k) = \mathbf{p}(k) - \mathbf{p}(k-1)$. Il y a principalement deux possibilités :

1. $\mathbf{B}_{k+1} = f_B(\mathbf{C}_k, \mathbf{A}_k)$, $\mathbf{C}_{k+1} = f_C(\mathbf{A}_k, \mathbf{B}_{k+1})$, $\mathbf{A}_{k+1} = f_A(\mathbf{B}_{k+1}, \mathbf{C}_{k+1})$,
2. $\mathbf{B}_{k+1} = f_B(\mathbf{C}_k, \mathbf{A}_k)$, $\mathbf{C}_{k+1} = f_C(\mathbf{A}_k, \mathbf{B}_k)$, $\mathbf{A}_k = f_A(\mathbf{B}_k, \mathbf{C}_k)$,

avec les définitions de (1.25). La première possibilité est préférable pour l'algorithme ALS (principe de Gauss-Seidel), tandis que l'autre donne la vraie direction de descente au point $\mathbf{p}(k)$. En réalité, la seconde possibilité ne s'avère pas significativement meilleure que la première. Un exemple de résultat est donné figure 4 avec une période de $n = 10$, en employant la première possibilité.

- **Descente du gradient.** La direction de recherche correspond à la direction de descente à chaque itération : $\Delta\mathbf{p}(k) = \mathbf{g}(k)$. Un exemple de résultat est donné figure 4.
- **Levenberg-Marquardt.** La direction de recherche est donnée par : $\Delta\mathbf{p}(k) = [\mathbf{J}(k)^T \mathbf{J}(k) + \lambda(k) \mathbf{I}]^{-1} \mathbf{g}(k)$. Contrairement aux algorithmes précédents, l'application pratique du pas optimal sur la méthode de Levenberg-Marquardt n'a pas donné de résultats toujours concluants. Ce point reste à élucider.

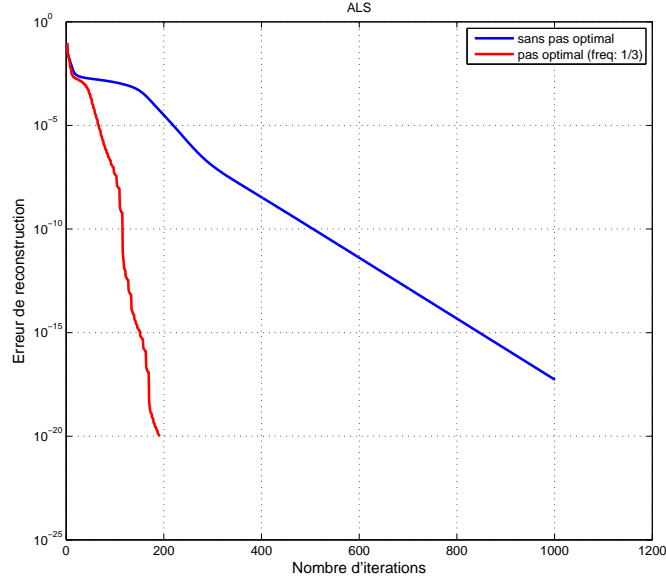


FIG. 4 – Amélioration de la convergence de l’ALS en utilisant le pas optimal.

- **Problème de l’ordre 5.** Nous avons récemment testé l’application du pas optimal (avec ALS) à des tenseurs symétriques ou non, de différents rangs et dont l’ordre variait de 3 à 6. Celui-ci a toujours contribué à une amélioration des algorithmes à l’exception notable des tenseurs d’ordre 5. Ce comportement inattendu reste à expliquer notamment par l’étude des polynômes de degré 10 à minimiser.

2.2.6 Evaluation des performances

L’évaluation de performances d’identification aveugle consiste à comparer l’ensemble des matrices obtenues $[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}]$ avec les matrices utilisées dans la simulation, $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$. La qualité de l’estimation est mesurée par la valeur du NMSE (Normalised Mean Square Error) entre deux matrices correspondantes. La difficulté vient du fait que ces matrices ne sont estimées qu’à une permutation-échelle post-multiplicative près. Contrairement au cas sur-déterminé, on ne peut pas utiliser de critère invariant aux permutations-échelles [8].

Nous proposons deux façons d’effectuer ce calcul. Toutes les deux nécessitent le calcul de la distance entre deux vecteurs à un facteur d’échelle près, ce qui se fait très simplement. On la définit comme suit :

$$\delta(\mathbf{u}, \mathbf{v}) = \left\| \mathbf{u} - \frac{\mathbf{v}^H \mathbf{u}}{\mathbf{v}^H \mathbf{v}} \mathbf{v} \right\|$$

Algorithme optimal. Il s’agit de l’algorithme utilisé dans nos simulations. Pour chaque permutation \mathbf{P} de dimension $K \times K$, on calcule l’erreur d’identification :

$$\Delta(\mathbf{P}) = \sum_{k=1}^K \delta(\mathbf{u}(k), \mathbf{v}(k))$$

où $\mathbf{u}(k)$ et $\mathbf{v}(k)$ désignent respectivement la k ème colonne des matrices

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{bmatrix} \mathbf{P} \quad \text{et} \quad \hat{\mathbf{M}} \stackrel{\text{def}}{=} \begin{bmatrix} \hat{\mathbf{A}} \\ \hat{\mathbf{B}} \\ \hat{\mathbf{C}} \end{bmatrix}$$

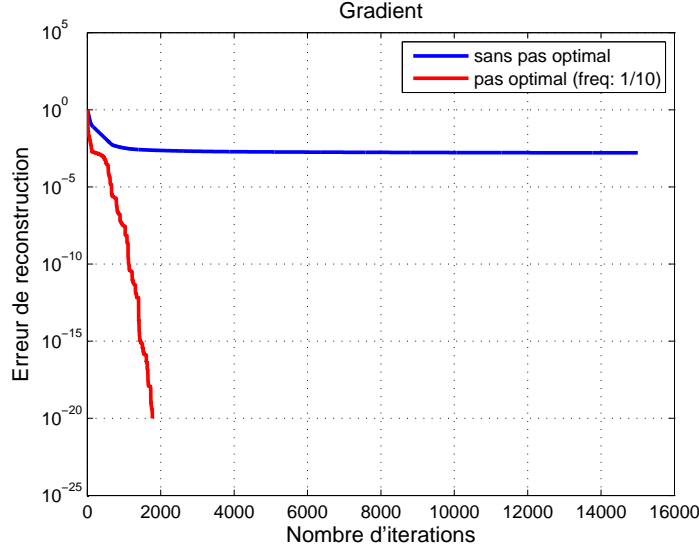


FIG. 5 – Amélioration de la convergence de la descente de gradient en utilisant le pas optimal.

L'erreur d'identification retenue sera l'erreur minimale obtenue sur l'ensemble des permutations. Evidemment, cet algorithme n'est applicable que si la dimension K est très modérée. En pratique, le rang du tenseur ne doit pas excéder 9. C'est le cas des simulations que nous présentons dans ce rapport.

Algorithme glouton. Pour des valeurs de K plus élevées, il est impossible de décrire exhaustivement l'ensemble des permutations. On procède donc de façon sous-optimale. On choisit la colonne $\hat{\mathbf{m}}(1)$ de $\hat{\mathbf{M}}$ de plus grande norme et on calcule sa distance avec la colonne $\mathbf{m}(k_1)$ de \mathbf{M} la plus proche au sens de la distance δ :

$$\Delta_1 = \delta(\mathbf{m}(k_1), \hat{\mathbf{m}}(1))$$

Puis on retire des matrices \mathbf{M} et $\hat{\mathbf{M}}$ les colonnes sélectionnées, et on recommence. On obtient ainsi une distance sous-optimale

$$\Delta = \sum_{k=1}^K \Delta_k$$

après K itérations.

On notera que le critère proposé par R.Bro faisait intervenir une permutation par mode, ce qui non seulement augmentait la complexité calcul, mais aussi fournissait des valeurs optimistes de l'erreur. En outre, dans le cas glouton, l'approximation de la solution optimale était moins bonne car la permutation opérait en dimension plus faible.

2.2.7 HOSVD

Bien que très séduisante pour sa rapidité, la méthode de Levenberg-Marquardt appliquée à la décomposition tensorielle nécessite la résolution de systèmes linéaires dont la taille est égale au produit des dimensions du tenseur. Ceci peut s'avérer réhibitoire dans le cas de tenseurs de grandes dimensions. Une généralisation de la décomposition en valeur singulière (SVD) à des ordres supérieurs à deux permet alors de compresser le tenseur à décomposer.

Dans le cas général, étant donné le tenseur \mathcal{T} d'ordre d et de dimensions $(I_1 \times I_2 \times \dots \times I_d)$, nous cherchons d matrices unitaires $\mathbf{U}^{(1)} (I_1 \times R_1)$, $\mathbf{U}^{(2)} (I_2 \times R_2)$, \dots , $\mathbf{U}^{(d)} (I_d \times R_d)$, de rang R_1, R_2, \dots, R_d , respectivement,

à partir desquelles le tenseur “coeur” \mathcal{S} de dimensions $(R_1 \times R_2 \times \dots \times R_d)$ est obtenu. Ce tenseur s’écrit :

$$\mathcal{S} = \mathcal{T} \underset{1}{\bullet} \mathbf{U}^{(1)T} \underset{2}{\bullet} \mathbf{U}^{(2)T} \dots \underset{d}{\bullet} \mathbf{U}^{(d)T}.$$

En pratique, \mathcal{S} est calculé à partir de d décompositions en valeurs singulières des matrices dépliantes du tenseur selon chaque mode. Dans ce cas, l’algorithme d’estimation (ALS, LM, gradient,...) calcule la décomposition CAND du tenseur réduit \mathcal{S} . Après convergence, nous revenons à l’espace originel en reconstruisant le tenseur \mathcal{T} à partir des matrices de changement de base $\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(d)}$.

L’intérêt de la HOSVD est bien illustré par l’exemple suivant : Nous avons construit un tenseur de taille $50 \times 50 \times 50$ et de rang 8. Dans ce cas, le calcul d’une itération de LM est très coûteux et dépasse rapidement les possibilités de notre machine de test (outre le temps de calcul, la première limitation est causée par un manque de mémoire). Après compression par HOSVD, le tenseur à décomposer n’est plus que de taille au plus $8 \times 8 \times 8$. L’algorithme a alors convergé au bout de 71 itérations, pour une erreur de reconstruction de 2.10^{-25} . Après reconstruction des matrices estimées, le NMSE est de l’ordre de 10^{-13} pour les trois matrices. En outre, l’ajout de bruit ne nuit pas dramatiquement à l’estimation. En effet nous obtenons des NMSE proches de 5.10^{-5} et 5.10^{-3} pour des rapports signal sur bruit valant respectivement $60dB$ et $20dB$.

2.3 Techniques tensorielles basées sur les cumulants

Dans cette section, on utilise les cumulants d’ordre 4 ou 6, c’est à dire les dérivées 4ème ou 6ème de la seconde fonction caractéristique à l’origine. On note le tenseur cumulant d’ordre 4 :

$$\mathcal{C}_{ijkl}^{\mathbf{y}} \stackrel{\text{def}}{=} \text{Cum}\{y_i, y_j^*, y_k^*, y_l\} \in \mathbb{C}^{N \times N \times N \times N}$$

que l’on convient de ranger dans une matrice de “quadricovariance” de taille $N^2 \times N^2$:

$$\mathbf{C}_4^{\mathbf{y}} \stackrel{\text{def}}{=} \text{mat}(\mathcal{C}^{\mathbf{y}})$$

et celui d’ordre 6 :

$$\mathcal{H}_{ijklmn}^{\mathbf{y}} \stackrel{\text{def}}{=} \text{Cum}\{y_i, y_j, y_k^*, y_l^*, y_m^*, y_n\} \in \mathbb{C}^{N \times N \times N \times N \times N \times N}$$

que l’on range dans une matrice d’“hexacovariance” de taille $N^3 \times N^3$:

$$\mathbf{C}_6^{\mathbf{y}} \stackrel{\text{def}}{=} \text{mat}(\mathcal{H}^{\mathbf{y}})$$

Si les observations ne sont pas stationnaires, on utilisera les mêmes estimateurs, ce qui aura pour effet de stationnariser les cumulants [23, 24]. Les cumulants, stationnarisés ou non, sont des grandeurs tensorielles, car ils jouissent de la propriété de multi-linéarité [40]. L’équation (2.28) permet de modéliser ces cumulants, grâce à la propriété de multilinéarité aux ordres pairs [4] :

$$\mathbf{C}_4^{\mathbf{y}} = \mathbf{H}^{\odot 2} \mathbf{\Delta}_4 \mathbf{H}^{\odot 2H} \quad (2.34)$$

$$\mathbf{C}_6^{\mathbf{y}} = \mathbf{H}^{\odot 3} \mathbf{\Delta}_6 \mathbf{H}^{\odot 3H} \quad (2.35)$$

où $\mathbf{H}^{\odot 2} \stackrel{\text{def}}{=} \mathbf{H} \odot \mathbf{H}^*$ et $\mathbf{H}^{\odot 3} \stackrel{\text{def}}{=} \mathbf{H} \odot \mathbf{H} \odot \mathbf{H}^*$, et où $\mathbf{\Delta}_4$ et $\mathbf{\Delta}_6$ sont des matrices diagonales de dimensions $K \times K$ contenant sur leur diagonale les cumulants marginaux des sources d’ordre 4 et 6, respectivement, $\mathcal{C}_{iii}^{\mathbf{s}}$ et $\mathcal{H}_{iiii}^{\mathbf{s}}$, que l’on notera κ_i pour alléger les écritures.

Le problème consiste à estimer la matrice \mathbf{H} à partir de la donnée de la matrice $\mathbf{C}_4^{\mathbf{y}}$ ou $\mathbf{C}_6^{\mathbf{y}}$. Suivant l’idée de [3, 4], nous savons que toute racine carrée de (2.34-2.35) s’écrit

$$(\mathbf{C}_{2q}^{\mathbf{y}})^{1/2} = \mathbf{H}^{\odot q} \mathbf{\Delta}_{2q} \mathbf{V} \quad (2.36)$$

où \mathbf{V} est une matrice unitaire. L’estimation de la matrice \mathbf{V} , puis de $\mathbf{H}^{\odot q}$, puis de \mathbf{H} , par les algorithmes BIOME est l’objet de la section suivante.

2.3.1 Algorithmes BIOME

Les algorithmes BIOME ont été présentés dans [4] pour un ordre pair quelconque supérieur ou égal à 4, mais nous présentons la version à l'ordre 6 (BIRTH), introduite à l'origine dans [3] et améliorée dans [1]. Cela ne restreint pas la généralité et allège la présentation. On notera que, en dépit de ses vertus [2], la version à l'ordre 4 (ICAR) ne permet pas d'identifier des mélanges sous-déterminés. L'ordre 6 est par conséquent l'ordre le plus faible pour lequel les algorithmes de la famille BIOME peuvent être utilisés dans le contexte sous-déterminé qui nous occupe.

L'idée est d'utiliser une invariance existant entre deux blocs consécutifs de $(\mathbf{C}_6^y)^{1/2}$. Remarquons tout d'abord que cette matrice, de taille $N^3 \times K$, peut être découpée en N blocs $\mathbf{\Gamma}_n$ chacun de taille $N^2 \times K$ où :

$$\mathbf{\Gamma}_n = (\mathbf{H} \odot \mathbf{H}^*) \mathbf{\Phi}_n \mathbf{\Delta}_4^{1/2} \mathbf{V}^H$$

et où $\mathbf{\Phi}_n$ désigne la matrice diagonale $K \times K$ contenant la n ème ligne de \mathbf{H} , $1 \leq n \leq N$. Cette relation montre clairement que toutes ces matrices ont théoriquement le même espace singulier à droite. L'algorithme BIRTH repose sur l'observation que la famille de matrices hermitiennes $\mathbf{\Theta}_{m,n} \stackrel{\text{def}}{=} \mathbf{\Gamma}_m^\dagger \mathbf{\Gamma}_n$, où le symbole \dagger désigne la pseudo-inverse, admet les mêmes vecteurs propres. On peut donc obtenir la matrice \mathbf{V} en diagonalisant simultanément (et approximativement dans un contexte bruité) l'ensemble de ces $N(N-1)/2$ matrices $\mathbf{\Theta}_{m,n}$.

Calcul de la matrice de mélange. Une fois la matrice unitaire \mathbf{V} obtenue, un certain nombre de façons de calculer \mathbf{H} sont envisageables. Cinq solutions sont décrites dans [1], la plus simple consistant à n'utiliser que le premier bloc $N \times K$ de $\mathbf{H}^{\odot 3}$, dont le conjugué donne une estimation de \mathbf{H} à un facteur d'échelle près. La plus performante (mais aussi la plus coûteuse) consiste à calculer chacune des K colonnes de \mathbf{H} à partir de chaque colonne \mathbf{b}_n de $\mathbf{H}^{\odot 3}$, $1 \leq n \leq K$, en procédant comme suit. On découpe \mathbf{b}_n en N vecteurs de taille égale $N^2 \times 1$, et pour chacun d'entre eux on forme une matrice \mathbf{B}_n de taille $N \times N$ par une opération Unvec. On cherche ensuite le vecteur propre dominant approximativement en commun des ces N matrices $N \times N$. Entre ces deux extrêmes, une approche de complexité modérée consiste à prendre la moyenne des N sous matrices $N \times N$, ce qui fournit une estimée de la matrice $\mathbf{H}^{\odot 2}$. De la même manière que précédemment, le vecteur propre dominant de chacune des matrices dont la représentation vectorisée est donnée par une colonne de $\mathbf{H}^{\odot 2}$, est une estimation d'une colonne de \mathbf{H} .

En réalité, chaque colonne de la matrice $\mathbf{H}^{\odot 3}$ estimée représente théoriquement, sous la forme d'un vecteur $N^3 \times 1$, un tenseur de rang 1 et de dimensions $N \times N \times N$. L'estimation de chaque colonne de \mathbf{H} se déduit de l'approximation de rang 1 de ce tenseur. Cette approximation peut donc aussi être calculée à l'aide d'un algorithme itératif [29, 37], initialisé à partir d'une des solutions proposées dans [1] ou [36]. Ce serait un raffinement possible.

2.3.2 Algorithmes FOOBI

Deux algorithmes de type FOOBI ont été décrits dans [35]. Nous renvoyons à cette référence pour leur description. Ces algorithmes, proposés tout récemment, permettent d'identifier le mélange lorsque le nombre de sources est proche de la borne ultime (mais encore significativement inférieure) : le rang générique. Comme nous l'avons précisé dans la section 2.3.1, les algorithmes BIOME peuvent identifier un mélange en aveugle en recourant aux cumulants d'ordre 6 ou plus. Les algorithmes FOOBI, bien que conçus uniquement à l'ordre 4, en sont capables.

En contrepartie de cet avantage, l'algorithme FOOBI nécessite une diagonalisation conjointe approximative de plusieurs matrices, qui requiert une procédure itérative. Cette dernière est toutefois maintenant bien connue, et bien que ne bénéficiant pas de preuve théorique de convergence, elle exhibe un bon comportement.

L'algorithme FOOBI2 nécessite quant à lui une "zéro diagonalisation conjointe", c'est à dire la minimisation conjointe des termes diagonaux de plusieurs matrices. Ce problème est résolu à l'aide d'un algorithme itératif

	4-3			5-3			6-3		
	min (1e-4)	max	med (1e-3)	min (1e-3)	max	med (1e-2)	min (1e-3)	max	med (1e-1)
ALESCAF4	0.22	1.04	0.33	0.11	0.89	0.24	0.3	1.43	0.3
LEMACAF4	0.22	1.04	0.27	0.11	1.18	0.16	0.3	1.45	0.2
ALESCAF3	0.26	1.54	0.41	0.14	1.77	0.51	1	1.75	5.38
LEMACAF3	0.26	1.54	0.4	0.14	1.58	0.39	1	1.95	4.91
6-BIOME	0.21	2.08	0.56	0.22	1.93	0.33	0.6	1.98	0.84
FOOBI2	0.33	1.47	0.3	0.52	2.09	0.49	0.21	2.32	7.91

TAB. 1 – Erreur d’estimation minimale, maximale et médiane de la matrice de mélange pour différents niveaux de sous-détermination, déterminées sur 100 expériences.

dont la convergence fait défaut en pratique. On doit donc tirer aléatoirement plusieurs valeurs initiales en espérant que l’une d’entre elles permet d’atteindre le minimum global.

Enfin, notons aussi que la borne sur le nombre maximal de sources identifiables est évidemment bien plus élevée à l’ordre 6 qu’à l’ordre 4.

2.4 Comparaisons des algorithmes pour des sources BPSK

2.4.1 Algorithmes retenus pour les simulations

Les observations précédentes nous ont permis de ne retenir que 4 algorithmes pour tester les performances de la méthode d’identification basée sur la fonction caractéristique :

- ALS sur tenseur de dérivées d’ordre 4 (noté par la suite ALESCAF4)
- ALS à pas optimal sur tenseur de dérivées d’ordre 3 (ALESCAF3)
- LM sur tenseur de dérivées d’ordre 4 (LEMACAF4)
- LM sur tenseur de dérivées d’ordre 3 (LEMACAF3)

Les performances asymptotiques de ces algorithmes sont donc comparées à celle de BIOME et FOOBI dans trois séries de simulations faisant toutes intervenir des sources de type BPSK. Nous nous intéressons exclusivement à l’erreur d’estimation de la matrice de mélange (NMSE). Chaque série correspond à l’étude de l’influence d’un paramètre du problème d’identification : le degré de sous-détermination, la taille du bloc source et le rapport signal sur bruit. Le protocole de test est commun aux trois séries. Un seul de ces paramètres évolue à chaque fois. Cependant il est évident que de nombreux autres paramètres ont une incidence sur les résultats, en particulier, la réalisation des sources, l’initialisation des tenseurs et le choix de la grille.

Afin de limiter l’influence de ces paramètres sur la comparaison des algorithmes, nous avons effectué des tirages de Monte-Carlo. Pour chaque simulation de chaque série, nous avons réalisé 100 expériences différentes pour lesquelles seules variaient, la réalisation des sources, la grille et l’initialisation. Bien entendu, à chaque fois les mêmes 100 conditions expérimentales sont utilisées pour les six algorithmes.

Pour chaque série, le nombre de capteurs est fixé à 3 et le nombre de sources est au minimum 4 afin d’assurer la sous-détermination.

Par ailleurs, nous rappelons que le nombre de points de la grille pour ALESCAF et LEMACAF est fixé à 8 et que ceux-ci sont tirés aléatoirement dans $[-1; 1]^N$, $N = 3$.

2.4.2 Première série, influence de la sous-détermination

Dans cette première étude, seul le nombre de sources varie de 4 à 6. La taille du bloc source est fixée à 10000 échantillons et le bruit est nul. Les résultats des trois simulations (4 sources, 5 sources et 6 sources)

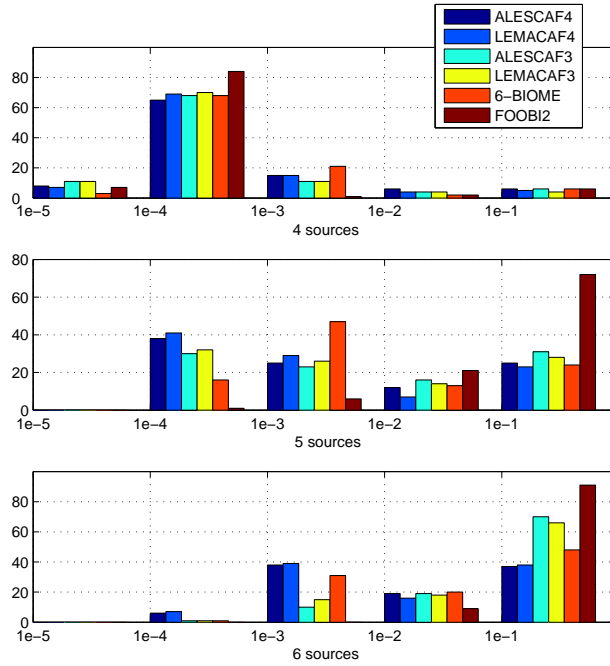


FIG. 6 – Répartition de l’erreur d’estimation de la matrice de mélange en fonction de la sous-détermination et des algorithmes utilisés.

sont représentés sur les histogrammes de la figures 6. Le tableau 1 permet une représentation plus synthétique.

4 sources Quel que soit l’algorithme utilisé, peu d’estimations ont un NMSE inférieur à 10^{-4} ou supérieur à 10^{-2} . La majorité des erreurs étant comprise entre 10^{-4} et 10^{-3} . Dans ce cas, FOOB12 se détache légèrement comme le meilleur algorithme. On notera également que les algorithmes basés sur les dérivées d’ordre 3 font un peu mieux que leurs homologues d’ordre 4. Enfin, peu de différences entre ALS et LM.

5 sources Il n’y a plus aucune valeur inférieure à 10^{-4} . Au-delà, celles-ci sont plus diversement réparties suivant les algorithmes que dans le cas précédent. ALESCAF4 et LEMACAF4 donnent maintenant les meilleurs résultats (40% de valeurs inférieures à 10^{-3}) devant ALESCAF3 et LEMACAF3 (30% de valeurs inférieures à 10^{-3}) puis viennent respectivement 6-BIOME (moins de 20% de valeurs inférieures à 10^{-3}) et enfin FOOB12 dont les résultats ne sont pas bons (70% de valeurs supérieures à 10^{-1}). Enfin, ALS et LM ne se départagent vraiment que sur la valeur médiane en faveur de ce dernier.

6 sources Seuls les algorithmes basés sur la fonction caractéristique produisent encore des estimations inférieures à 10^{-3} mais dans moins de 10% des cas. Ensuite, LEMACAF4 et ALESCAF4 se détachent assez nettement (plus de 45% de valeurs inférieures à 10^{-2}) devant 6-BIOME (30% de valeurs inférieures à 10^{-2}) puis ALESCAF3 et LEMACAF3 (environ 15% de valeurs inférieures à 10^{-2}) et surtout FOOB12 (aucune valeur, puisque la borne sur le nombre maximal de sources est dépassé).

Conclusion Il apparaît assez clairement que ALESCAF4 et LEMACAF4 sont les plus attrayants lorsque la sous-détermination augmente. Au contraire FOOB12 est le plus efficace lorsque la sous-détermination est faible, mais le moins bon dans le cas inverse. Les algorithmes d’ordre 3 ne sont pas non plus utilisables lorsque

la sous-détermination se fait trop importante. Les performances de 6-BIOME ne sont ni les meilleures ni les pires, et restent convenables. Enfin LEMACAF donne globalement de meilleurs résultats que ALESCAF mais la différence est assez faible.

2.4.3 Seconde série, étude de la taille du bloc source

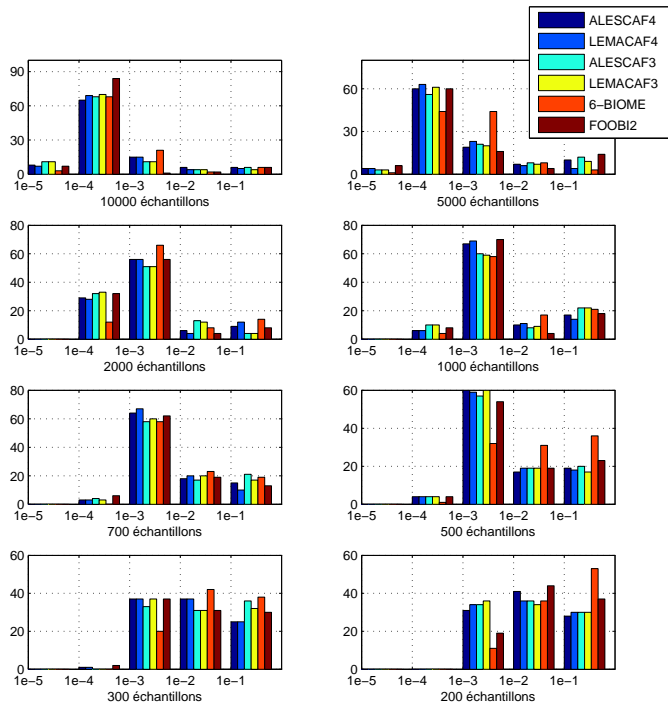


FIG. 7 – Répartition de l’erreur d’estimation de la matrice de mélange en fonction du nombre d’échantillons des sources et des algorithmes utilisés.

Pour cette série de simulations, nous faisons varier le nombre d’échantillons des sources de 200 à 10000. En effet, il s’agit d’un paramètre important dans le cadre des approches statistiques du problème d’identification. Le nombre de sources est fixé à 4 et le bruit est nul. Les tailles de blocs testées sont 10000, 5000, 2000, 1000, 700, 500, 300 et 200. Les résultats de ces huit simulations sont représentés sur les histogrammes de la figure 7. La figure 8 représente l’évolution de la médiane des NMSE obtenus avec chaque algorithme en fonction de la taille des blocs.

Analyse des résultats Il était prévisible de voir les performances de ces algorithmes statistiques se dégrader avec la diminution du nombre d’échantillons des sources, puisque l’estimation des moments statistiques devient alors de moins en moins fiable. D’une manière générale, au dessus de 5000 échantillons, la majorité des valeurs d’erreur est inférieure à 10^{-4} . Entre 5000 et 1000 échantillons, celle-ci se répartie entre 10^{-4} et 10^{-2} . Entre 1000 et 500 échantillons elle est comprise entre 10^{-3} et 10^{-2} . Enfin, entre 500 et 200 échantillons les estimations se répartissent équitablement entre 10^{-3} et 10^{-1} . Ces observations doivent être nuancées en fonction des algorithmes utilisés. Il est assez difficile de dégager une tendance claire. On remarquera toutefois que 6-BIOME donne systématiquement les moins bon résultats. Cela n’est pas réhibitoire lorsque le nombre d’échantillon est supérieur à 700 mais devient très clair en dessous. Les cinq autres algorithmes sont très

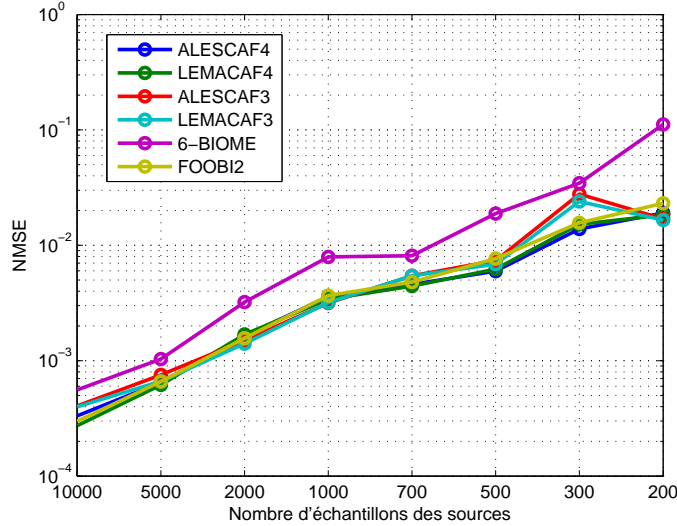


FIG. 8 – valeur médiane de l’erreur d’estimation de la matrice de mélange en fonction du nombre d’échantillons des sources.

difficiles à départager. Seul le dernier cas de figure (200 échantillons) permet de détacher ALESCAF et LEMACAF de FOOB12 puisque ce dernier donne un résultat correct (NMSE inférieur à 10^{-2}) dans près de deux fois moins d’expérience.

2.4.4 Troisième série, étude du niveau de bruit

Enfin, nous faisons varier le rapport signal sur bruit des données. La taille du bloc source est fixée à 10000 échantillons et le nombre de sources à 4. Les RSB testés sont : RSB infini, $80dB$, $50dB$, $20dB$, $5dB$, $0dB$. Les résultats de ces six simulations sont représentés sur les histogrammes de la figure 9. La figure 10 représente l’évolution de la médiane des NMSE obtenus avec chaque algorithme en fonction du niveau de bruit.

Analyse des résultats Jusqu’à $20dB$, une forte majorité (entre 60 et 70%) de valeurs sont comprises entre 10^{-4} et 10^{-3} . Pour $5dB$ cette majorité se déplace entre 10^{-3} et 10^{-2} puis entre 10^{-2} et 10^{-1} pour $0dB$ de rapport signal sur bruit. Les résultats de la comparaison sont globalement très proches, comme en témoigne la figure 10. Jusqu’à $20dB$, FOOB12 donne généralement les meilleurs résultats et 6-BIOMES les moins bons. Entre $20dB$ et $5dB$ ce sont ALESCAF4 LEMACAF4 et 6-BIOMES qui nous apparaissent préférables. Enfin pour $0dB$, très peu de résultats sont acceptables, néanmoins FOOB12 et 6-BIOMES ressortent comme les moins mauvais, suivis de ALESCAF3 et LEMACAF3. Enfin, encore une fois LM converge généralement légèrement mieux que ALS.

2.4.5 Conclusion

Ces différentes comparaisons montrent que ces six algorithmes sont globalement difficiles à départager. Cependant, sur une expérience donnée, les différences d’estimation peuvent être très importantes. Cette dernière observation illustre l’influence des paramètres non testés dans ce rapport. En particulier, il nous sera utile de savoir si pour une expérience donnée, les performances d’ALESCAF et LEMACAF peuvent être améliorées en choisissant au mieux leurs paramètres propres (grille, initialisation) ou au contraire si seule la réalisation des sources est réellement importante.

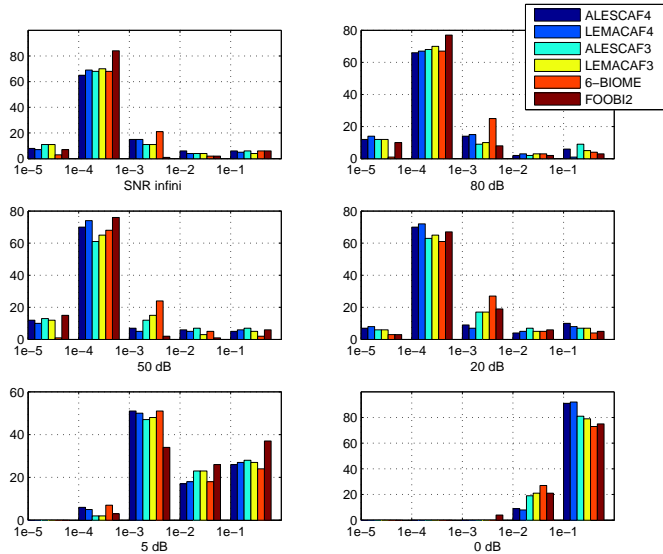


FIG. 9 – Répartition de l’erreur d’estimation de la matrice de mélange en fonction du nombre d’échantillons des sources et des algorithmes utilisés.

Pour l’heure, nous pouvons déjà tirer de ces résultats quelques enseignements importants. La tendance la plus nette concerne la sous détermination, il est clair que plus celle-ci s’accroît, plus il est intéressant d’utiliser la fonction caractéristique et en particulier ALESCAF4 et LEMACAF4. Notons d’ailleurs que d’une manière générale, même si ce n’est pas toujours vrai (en particulier dans les cas simples), l’usage de dérivées d’ordre 4 est préférable à celui de dérivées ordre 3. De même LM est régulièrement meilleur qu’ALS. Cependant, statistiquement la différence est assez faible. Il faudra alors faire le choix entre un algorithme demandant peu d’itérations pour converger et un autre pour lequel celles-ci sont moins coûteuses. Par ailleurs, FOOB12 paraît être la solution la plus sûre dans les cas les moins difficiles. Enfin sur ces essais, 6-BIOME est l’algorithme qui souffre le plus d’un faible nombre d’échantillons. Lorsque celui-ci est suffisant, ses résultats sont proches de ceux d’ALESCAF4 et LEMACAF4.

2.4.6 Mélanges Complexes

Comme nous l’avons montré dans la section 2.2.1, la transformation du problème en décomposition tensorielle pour des observations issues de sources et/ou de mélanges complexes diffère du cas précédent, l’équation coeur n’étant plus la même. En particulier, le problème reste ouvert lorsque sources et mélange sont tous deux complexes. En revanche, les algorithmes ALESCAF et LEMACAF ont été facilement étendus au cas de sources réelles avec mélange complexe (ou l’inverse). Ce cas de figure n’est pas très réaliste mais il s’agit là d’une première étape. Pour cela, il est nécessaire de considérer les valeurs du mélange comme des points de \mathbb{R}^2 et non pas de \mathbb{C} . En outre, ce traitement double artificiellement le nombre de capteurs autorisant ainsi un ratio sources/capteurs plus important. Afin d’illustrer ce propos nous présentons sur la figure 11 une comparaison entre ALESCAF4 et LEMACAF4 pour un mélange complexe de sources BPSK en fonction du niveau de bruit.

Attention, il s’agit de performances ultimes dans le sens où ce sont les vraies valeurs des statistiques qui sont utilisées. Par conséquent, seuls $2^8 = 256$ échantillons sont utilisés pour les 8 sources. Les niveaux de bruit ne sont donc pas comparables avec ceux des figures 9 et 10. Les mauvaises performances d’ALESCAF face à LEMACAF peuvent être dues à un nombre d’itérations trop faible. Pour être plus juste, la comparaison

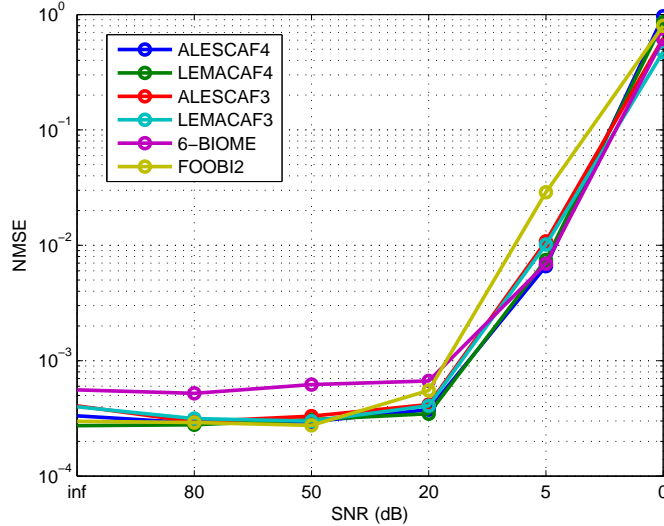


FIG. 10 – valeur médiane de l’erreur d’estimation de la matrice de mélange en fonction du nombre d’échantillons des sources.

devrait se faire à complexité numérique égale. Mais il est probable que dans ce dernier cas, l’écart se creuserait encore plus en faveur de LEMACAF ; ceci sera vérifié prochainement.

2.5 Techniques déterministes

Dans cette section, nous n’allons utiliser aucune statistique sur les données ; les algorithmes sont donc entièrement déterministes. Les tenseurs que nous manipulerons seront construits par rangement direct des données d’une manière appropriée, qui va dépendre des diversités disponibles. Nous allons présenter le cas des mélanges convolutifs, car cela ne complique pas significativement la présentation, dès lors que la synchronisation au rythme symbole est admise pour l’ensemble des utilisateurs simultanément, ce qui sera admis. Le cas d’utilisateurs de périodes symbole différentes fait partie de nos perspectives (c.f. section 2.6.6).

2.5.1 Modélisations du tenseur de données

La représentation en bande de base du signal à temps discret, $x_k(n)$, reçu sur le capteur k après propagation à travers un canal MIMO FIR de longueur maximale I prend la forme suivante :

$$x_k(n) = \sum_{q=1}^K x_k^{(q)}(n) = \sum_{q=1}^K \sum_{i=1}^I h_k^{(q)}(i) s^{(q)}(n - i + 1), \quad (2.37)$$

où $h_k^{(q)}(\cdot)$ est la réponse impulsionnelle du canal reliant le q ième utilisateur au capteur k . La séquence $s^{(q)}(n)$ désigne la suite de symboles transmise par le q ième utilisateur.

Canal multi-trajet. On suppose que le signal émis par chaque utilisateur traverse un canal spéculaire contenant L trajets significatifs. Pour simplifier, on suppose aussi que le nombre L de trajets est le même pour tous. Chaque trajet correspond à un groupe de réflecteurs voisins (figure 12) ; on suppose que l’étalement angulaire des réflecteurs au sein de chaque groupe est faible devant la résolution angulaire de l’antenne de capteurs.

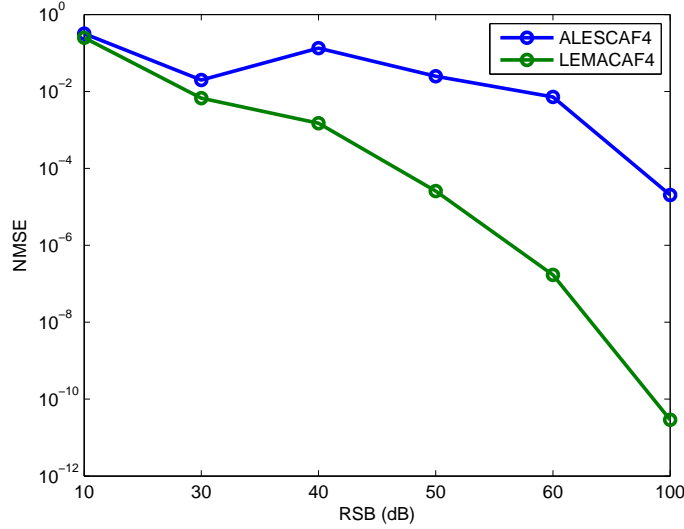


FIG. 11 – Erreur d’estimation de la matrice de mélange pour 8 sources et 2 capteurs, en fonction du rapport signal à bruit (dB). Comparaison entre ALESCAF4 et LEMACAF4 avec une grille de 4 points

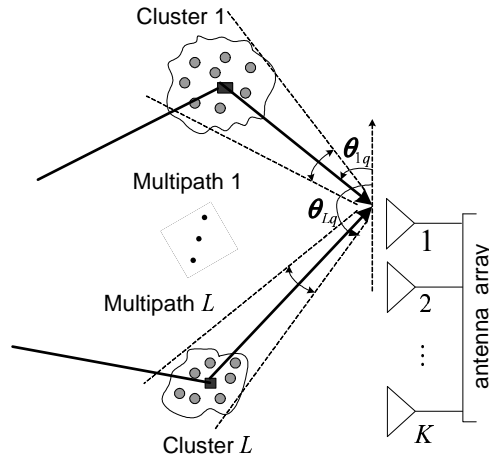


FIG. 12 – Représentation standard d’une propagation à trajets multiples.

Par exemple, ce modèle est valable en transmission montante si la station de base est suffisamment haute, c’est à dire s’il n’y a pas d’obstacles qui obstruent la communication dans son voisinage. le canal de propagation peut lui-même être factorisé [49] :

$$h_k^{(q)}(i) = \sum_{l=1}^L \beta_l^{(q)} a_k^{(q)}(\theta_l^{(q)}) g^{(q)}(i - 1 - \tau_l^{(q)}), \quad i = 1, \dots, I, \quad (2.38)$$

où $\beta_l^{(q)}$ désigne le coefficient de réflexion du trajet l du q ème utilisateur, $a_k(\theta_l^{(q)})$ est la réponse de l’antenne k au trajet l de l’utilisateur q , $\theta_l^{(q)}$ étant la direction d’arrivée correspondante. Le terme $\tau_l^{(q)}$ désigne le retard de propagation (normalisé par la période symbole T), et $g^{(q)}(\cdot)$ représente le filtre d’émission de l’utilisateur q .

L'astuce consiste ensuite à regrouper les termes g et β correspondant à l'émission et à la propagation : ensemble, ils représentent les effets du canal, tandis que le terme a représente la diversité spatiale à la réception [17]. Le modèle de réception peut alors s'écrire, en utilisant (2.37) et (2.38) :

Sur-échantillonnage. En présence d'excès de bande (fréquence symbole inférieure à la largeur de bande), on peut sur-échantillonner le signal reçu d'un facteur P . En posant

$$x_{k,n,p} \stackrel{\text{def}}{=} x_k(n + (p-1)/P), \quad g_{l,i,p}^{(q)} \stackrel{\text{def}}{=} g^{(q)}(i-1 + (p-1)/P - \tau_l^{(q)}),$$

nous pouvons réécrire (2.37) et (2.38) :

$$x_{k,n,p} = \sum_{q=1}^K \sum_{i=1}^I h_{k,i,p}^{(q)} s_{n,i}^{(q)} \quad (2.39)$$

$$h_{k,i,p}^{(q)} = \sum_{l=1}^L \beta_l^{(q)} a_{k,l}^{(q)} g_{l,i,p}^{(q)} \quad (2.40)$$

Si maintenant nous groupons les parties émission et propagation du canal, $\bar{h}_{l,i,p} \stackrel{\text{def}}{=} \beta_l^{(q)} g_{l,i,p}^{(q)}$, nous obtenons les données mesurées sous la forme du tenseur suivant :

$$x_{k,n,p} = \sum_{q=1}^K \sum_{l=1}^L \sum_{i=1}^I a_{k,l}^{(q)} \bar{h}_{l,i,p}^{(q)} s_{n,i}^{(q)}. \quad (2.41)$$

On peut montrer que ce modèle est de type CONFAC, avec $\Psi = \mathbf{I}_K \otimes (\mathbf{I}_L \otimes \mathbf{1}_I^T)$, $\Phi = \mathbf{I}_K \otimes (\mathbf{1}_L^T \otimes \mathbf{I}_I)$, et $\Omega = \mathbf{I}$.

Transmission DS-CDMA. Le modèle tensoriel a été proposé il y a quelques années pour égaliser les transmissions CDMA en champ lointain [46]. La présentation de cette section permet d'étendre cette dernière approche au cas où plusieurs trajets relatifs à un même utilisateur peuvent être reçus en provenance de directions différentes.

On note $c^{(q)}(t)$ la séquence d'étalement de l'utilisateur q , et J le facteur d'étalement, c'est à dire le ratio entre période symbole et période chip. Cette séquence est liée au code d'étalement par

$$c^{(q)}(t) = \sum_{j=1}^J c_j^{(q)} g(t - (j-1)T_c), \quad (2.42)$$

où $g(\cdot)$ désigne toujours le filtre d'émission. Cette fois-ci, il est judicieux de grouper le code d'étalement avec la réponse du canal, en posant :

$$u_{l,i,j}^{(q)} = \sum_{j'=1}^J \bar{h}_{l,i,j-j'}^{(q)} c_{j'}^{(q)} \quad (2.43)$$

On aboutit alors au même type de modèle tensoriel que dans le cas du sur-échantillonnage :

$$x_{k,n,j} = \sum_{q=1}^K \sum_{l=1}^L \sum_{i=1}^I a_{k,l}^{(q)} u_{l,i,j}^{(q)} s_{n,i}^{(q)}. \quad (2.44)$$

et le facteur d'étalement a remplacé le facteur de sur-échantillonnage.

Transmission OFDM. Lors des transmissions OFDM à F sous-porteuses, les symboles à transmettre sont groupés par paquets de F symboles, chacune des porteuses transmettant simultanément un symbole. Notons $s_{n,f}^{(q)}$ le f ème symbole du n ème bloc associé à l'utilisateur q . Un préfixe cyclique est inséré pour éviter l'interférence entre blocs adjacents. Si la longueur du préfixe cyclique est égale (au moins) à la mémoire de la réponse impulsionnelle, on peut alors montrer que le signal reçu s'écrit (après avoir retiré le préfixe cyclique) :

$$\mathbf{X}_{k,n} = \sum_{q=1}^K \mathbf{D}_k^{(q)} \mathbf{s}_n^{(q)}.$$

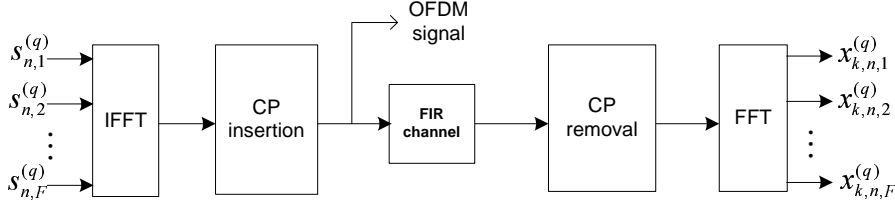


FIG. 13 – Schéma synoptique d'un système OFDM

où les matrices $\mathbf{D}_k^{(q)}$ sont diagonales et contiennent les canaux de propagation $d_{k,f}^{(q)}$ sur leur diagonale :

$$d_{k,f}^{(q)} = \sum_{l=1}^L \beta_l^{(q)} a_{k,l}^{(q)} \underbrace{\sum_{i=1}^I g_{i,l}^{(q)} e^{j2\pi(f-1)(i-1)/F}}_{\bar{g}_{f,l}^{(q)}} = \sum_{l=1}^L \beta_l^{(q)} a_{k,l}^{(q)} \bar{g}_{f,l}^{(q)}.$$

Cette fois-ci, il est judicieux de regrouper le coefficient de réflexion, le filtre d'émission, et le terme de Fourier :

$$\lambda_{f,l}^{(q)} = \beta_l^{(q)} \bar{g}_{f,l}^{(q)}$$

de sorte que le signal reçu peut être représenté par le tenseur suivant :

$$x_{k,n,f} = \sum_{q=1}^K \sum_{l=1}^L a_{k,l}^{(q)} \lambda_{f,l}^{(q)} s_{n,f}^{(q)}. \quad (2.45)$$

Si on pose $\lambda_{l,i,f}^{(q)} = \lambda_{f,l}^{(q)} \delta_{if}$, on peut faire apparaître le même modèle d'observation que dans les deux paragraphes précédents :

$$x_{k,n,f} = \sum_{q=1}^K \sum_{l=1}^L \sum_{i=1}^F a_{k,l}^{(q)} \lambda_{l,i,f}^{(q)} s_{n,i}^{(q)} \quad (2.46)$$

Autres diversités. D'autres modes d'émission-réception (CDMA, OFDM) utilisant des formes de diversité différentes (code, fréquence), et peuvent être abordés par le même problème de décomposition tensorielle [21]. Pour utiliser pleinement cette approche unifiée, il est utile d'utiliser la décomposition bloc-contrainte définie dans la section 1.2. Les différences résident dans (i) la troisième dimension I_3 du tenseur \mathbf{X} (qui n'a pas la même signification physique), (ii) la nature de la deuxième matrice de la décomposition représentant le canal de propagation, (iii) la forme exacte des matrices de contraintes, et (iv) la matrice \mathbf{S} qui n'est pas bloc-Töplitz dans le cas OFDM alors qu'elle l'est dans les autres cas. Plus précisément, nous avons dans tous les cas actuellement connus :

$$\Psi = \mathbf{I}_K \otimes (\mathbf{I}_L \otimes \mathbf{1}_{R_2}^T) \in \mathbb{C}^{K \times L \times L K R_2}, \quad \Phi = \mathbf{I}_K \otimes (\mathbf{1}_L^T \otimes \mathbf{I}_{R_2}) \in \mathbb{C}^{K R_2 \times L K R_2}, \quad (2.47)$$

2.5.2 Comparaison des algorithmes déterministes

Les simulations informatiques seront conduites prochainement pour les tenseurs asymétriques, et feront l'objet d'un rapport séparé.

2.6 Conclusion générale et perspectives

2.6.1 Méthodes basées sur un tenseur de statistiques des observations

En l'absence de diversité suffisante, il n'est pas possible de construire un tenseur de données d'ordre au moins 3 ayant des tranches matricielles qui ne soient pas proportionnelles entre elles. Dans de telles situations, le recours aux statistiques d'ordre supérieur à deux (fonction caractéristique, cumulants) permet de créer un tenseur d'ordre arbitrairement grand, et donc théoriquement d'identifier le mélange d'un nombre arbitraire de sources. Dans ce cas, on remplace la diversité par l'hypothèse d'indépendance statistique des sources. Cette solution risque d'être la seule viable en présence d'un excès de bande modeste et en l'absence d'autres formes de diversité (autres que espace et temps). C'est celle-ci sur laquelle notre attention a été portée dans le présent rapport. Les conditions théoriques d'utilisation (rang maximal, unicité...) commencent à être connues dans le cas de tenseurs complexes symétriques [11], grâce à des simulations informatiques. Des travaux théoriques sont en cours et ont permis de confirmer ces résultats [10]. Les autres cas (complexe à symétries hermitiennes, réel, non symétrique) sont actuellement approfondis [16].

Cependant, deux limitations doivent être soulignées : (i) bien évidemment, le nombre de sources prises en compte dans le mélange identifié est limité par les erreurs d'estimation de ces statistiques, et donc par l'horizon d'observation et par le rapport signal à bruit ; (ii) les techniques basées sur des tenseurs statistiques permettent d'identifier le mélange en aveugle, mais pas d'extraire les sources.

2.6.2 Techniques basés sur la fonction caractéristique

Il conviendrait de se pencher sur le calcul des performances élémentaires (biais et variance au second ordre) des estimateurs des fonctions caractéristiques et de leurs dérivées partielles. Ceci pourra peut-être mettre en évidence l'intérêt d'une part d'adopter une grille de calcul comprenant des points en dehors de l'origine (voire d'optimiser la dite grille), et d'autre part d'utiliser conjointement des dérivées d'ordre différent.

D'un point de vue algorithmique, l'usage de plusieurs ordres simultanément dans ALESCAF ou LEMACAF n'a pas encore été étudié lorsque $N > 2$, mais semble possible. Ceci permettrait de conférer une meilleure robustesse à la solution obtenue avec $N > 2$ capteurs. Par ailleurs, l'utilisation de notre outil de détection du rang 1 au sein d'autres algorithmes itératifs que celui de la section 2.3.2 peut s'avérer attractif.

2.6.3 Techniques basés sur les cumulants

Dans la section 2.3.2, il a été supposé que les tenseurs $\varphi(\mathbf{h}_u \mathbf{h}_{u'}, \mathbf{h}_v \mathbf{h}_{v'})$ sont linéairement indépendants [35]. On peut chercher à se passer de cette hypothèse, ce qui devrait lever la contrainte que K soit nécessairement borné par : $K(K-1) \leq N^2(N-1)^2$. Ceci donnerait naissance à un nouvel algorithme basé sur les cumulants d'ordre 4.

Les algorithmes BIOME peuvent aussi être mis en œuvre simultanément à plusieurs ordres. L'intérêt de concevoir des algorithmes de ce type fonctionnant à la fois avec les cumulants d'ordre 4 et 6 pourra être évalué.

Enfin, il pourrait être intéressant de développer un algorithme de la même veine que les algorithmes FOABI et FOABI-2 pour un tenseur contenant des dérivées de la fonction caractéristique en d'autres points que l'origine. Ceci n'a pas été étudié pour l'instant.

2.6.4 Algorithmes d'optimisation

De nombreux points restent à approfondir dans les prochains mois :

- L'algorithme LEMACAF doit être implanté pour des sources pouvant être complexes, avec des parties réelles et imaginaires pouvant être statistiquement indépendantes ou non.
- Les méthodes vont être comparées aussi en termes de complexité arithmétique. La question qui se pose est de savoir quelles performances peut atteindre chacune des méthodes pour un volume de calcul donné ; le nombre d'itérations ne veut pas dire grand chose lorsque les algorithmes sont différents.
- Il pourrait être intéressant d'étudier l'intérêt de l'algorithme LEMACAF lorsque les dérivées utilisées sont d'ordre réduit, par exemple ordre 2, ou bien ordres 2 et 3 conjointement. L'avantage de la fonction caractéristique est précisément qu'elle contient toutes les informations statistiques au voisinage de l'origine.
- En parallèle, les travaux de recherche se poursuivent pour la mise au point d'un algorithme de calcul de la décomposition des tenseurs symétriques d'ordre et de dimensions arbitraires, en présence d'un nombre de sources le plus grand possible : égal au rang générique lorsque la solution est essentiellement unique, et égal au rang générique moins 1 dans le cas contraire. C'est la borne d'identifiabilité maximale [11] [14] [15]. Cet algorithme s'inspire de celui développé dans [11] dans le cas de 2 capteurs, basé sur un théorème de Sylvester, et étend la procédure à des polynômes homogènes en plus de deux variables [12].

2.6.5 Techniques déterministes

Le point fort des techniques tensorielles déterministes est de permettre l'estimation conjointe du canal et des sources. Compte-tenu de leur potentiel considérable, nous pensons qu'approfondir les contextes opérationnels d'application de ces techniques est intéressant.

Le recours à différentes formes de diversité est crucial pour le développement de nouvelles techniques d'identification/égalisation MIMO aveugles. Ainsi, les diversités exploitables incluent habituellement l'espace, le temps, le code, le sur-échantillonnage en présence d'excès de bande, la polarisation, ou la fréquence. Dans le cadre de la présente étude, il n'est pas évident d'exploiter de telles diversités. On pourrait étudier la possibilité d'exploiter la non stationnarité sur un des modes, si les autres modes restent stationnaires. La motivation centrale est que plus l'ordre du tenseur sera élevé, plus le nombre de sources que l'on pourra extraire sera élevé.

Les conditions suffisantes permettant de garantir l'unicité des décompositions en blocs devront être établies (pour commencer dans le cadre de réflexions en champ lointain, puis dans le cas général). Dans ces deux cas, des algorithmes du type ALS pourront être développés, dans lesquels on met à jour les générateurs de la matrice Töplitz plutôt que la matrice elle-même ; le forçage (souple ou dur) de la structure Töplitz par projection n'est alors pas nécessaire (cf. section 2.2.4).

Par ailleurs, les conditions nécessaires d'unicité de la décomposition CAND à une permutation-échelle près, indépendamment de tout algorithme numérique, seront poursuivies, en particulier dans le cas non symétrique.

2.6.6 Utilisateurs asynchrones

En pratique, supposer que tous les utilisateurs sont synchrones peut être irréaliste. Bien que cette hypothèse n'ait pas d'incidence directe sur l'identification du mélange (à condition que l'excitation du système soit suffisante), cela en a évidemment sur l'estimation des sources. L'approche décrite dans ce rapport peut être modifiée lorsque les périodes symbole des utilisateurs sont connues mais différentes.

On peut en effet construire un tenseur de données adapté à chaque période cyclique. On obtient ainsi autant de tenseurs que de sources pré-détectées. La décomposition de chacun de ces tenseurs nous fournit des estimations de sources, parmi lesquelles il faut détecter celle (ou celles) qui a un sens physique (par exemple en examinant leur caractère discret ou non). Deux pistes peuvent être alors étudiées.

- (a) Soit on conserve l'estimation de la source synchronisée (et du canal afférent) obtenue en décomposant chacun de ces tenseurs

(b) Soit on met en place une procédure de déflation, en commençant par la source ayant été le mieux estimée (en se basant par exemple sur le caractère discret de l'estimation)

Ces deux approches rappellent les techniques de déflation parallèle ou séquentielle proposées dans [52, 44]. Il est probable que des sources ayant des périodes symbole différentes ne soient pas reliées dans un rapport rationnel. Par conséquent, les estimations obtenues pour les sources non synchronisées n'auront aucun sens et seront perçues comme du bruit (de niveau élevé). Il sera nécessaire de valider la procédure à l'aide de simulations informatiques.

2.6.7 Unicité de la décomposition CONFAC et applications à des systèmes multi-antennes

De nouvelles décompositions tensorielles (BLOCFAC et CONFAC) ont été développées et étudiées en termes d'identifiabilité. En particulier, l'exploitation des matrices de contrainte de la décomposition CONFAC permet de concevoir une architecture de transmission et réception multi-antennes généralisées avec diversités spatiale, temporelle et fréquentielle. Il est à noter que les propriétés d'unicité du modèle CONFAC ont été étudiées en vue de proposer un récepteur aveugle [20] [19].

D'un point de vue fondamental, nous envisageons poursuivre l'étude des propriétés d'unicité de la décomposition tensorielle CONFAC et détermination de conditions nécessaires et suffisantes d'unicité globale ou partielle.

Remerciements

Pour FOABI et BIOME, les simulations informatiques présentées ici ont utilisé les codes Matlab gracieusement fournis par L.DeLathauwer et L.Albera, respectivement.

Références

- [1] L. ALBERA, P. COMON, P. CHEVALIER, and A. FERREOL. Blind identification of underdetermined mixtures based on the hexacovariance. In *ICASSP'04*, volume II, pages 29–32, Montreal, May 17–21 2004.
- [2] L. ALBERA, A. FERREOL, P. CHEVALIER, and P. COMON. ICAR, a tool for blind source separation using fourth order statistics only. *IEEE Trans. Sig. Proc.*, 53(10) :3633–3643, October 2005.
- [3] L. ALBERA, A. FERREOL, P. COMON, and P. CHEVALIER. Sixth order blind identification of under-determined mixtures - BIRTH. In *ICA'03*, Japan, April 1-4 2003.
- [4] L. ALBERA, A. FERREOL, P. COMON, and P. CHEVALIER. Blind identification of overcomplete mixtures of sources (BIOME). *Lin. Algebra Appl.*, 391 :1–30, November 2004.
- [5] R. BRO and C. A. ANDERSSON. Improving the speed of multiway algorithms. part ii : Compression. *Chemometrics and Intelligent Laboratory Systems*, 42(1-2) :105–113, 1998.
- [6] R. BRO, R. A. HARSHMAN, and N. D. SIDIROPOULOS. Modeling multi-way data with linearly dependent loadings. Technical Report 176, KVL, 2005.
- [7] J. D. CARROLL and J. J. CHANG. Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition. *Psychometrika*, 35(3) :283–319, September 1970.
- [8] P. COMON. Independent Component Analysis, a new concept ? *Signal Processing, Elsevier*, 36(3) :287–314, April 1994. Special issue on Higher-Order Statistics.
- [9] P. COMON. Canonical tensor decompositions. Research Report RR-2004-17, I3S, June 17 2004.
- [10] P. COMON, G. GOLUB, L-H. LIM, and B. MOURRAIN. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis Appl.*, 2008. to appear.

- [11] P. COMON and B. MOURRAIN. Decomposition of quantics in sums of powers of linear forms. *Signal Processing, Elsevier*, 53(2) :93–107, September 1996. special issue on High-Order Statistics.
- [12] P. COMON, B. MOURRAIN, and E. TSIGARIDAS. Symmetric tensor decomposition. Research Report ISRN I3S/RR-2008-12-FR, I3S, Sophia-Antipolis, France, July 15 2008.
- [13] P. COMON and M. RAJIH. Blind identification of under-determined mixtures based on the characteristic function. *Signal Processing*, 86(9) :2271–2281, September 2006. <http://dx.doi.org/10.1016/j.sigpro.2005.10.007>.
- [14] P. COMON and J. ten BERGE. Generic and typical ranks of three-way arrays. Research Report ISRN I3S/RR-2006-29-FR, I3S, Sophia-Antipolis, France, September 4 2006.
- [15] P. COMON and J. ten BERGE. Generic and typical ranks of three-way arrays. In *Icassp'08*, Las Vegas, March 30 - April 4 2008.
- [16] P. COMON, J. M. F. ten BERGE, L. De LATHAUWER, and J. CASTAING. Generic and typical ranks of multi-way arrays. *Linear Algebra Appl.*, 2008. submitted.
- [17] A. L. F. de ALMEIDA, G. FAVIER, and J. C. M. MOTA. PARAFAC models for wireless communication systems. In *Int. Conf. on Physics in Signal and Image processing (PSIP)*, Toulouse, France, Jan. 31 - Feb. 2 2005.
- [18] A. L. F. de ALMEIDA, G. FAVIER, and J. C. M. MOTA. *The constrained block-PARAFAC decomposition*. Three-way methods in Chemistry and Psychology (TRICAP), Chania, Crete, Greece, June 2006.
- [19] A. L. F. de ALMEIDA, G. FAVIER, and J. C. M. MOTA. A constrained factor decomposition with application to MIMO antenna systems. *accepted for IEEE Transactions on Signal Processing*, 2007.
- [20] A. L. F. de ALMEIDA, G. FAVIER, and J. C. M. MOTA. Constrained tensor modeling approach to blind multiple-antenna CDMA schemes. *accepted for IEEE Transactions on Signal Processing*, 2007.
- [21] A. L. F. de ALMEIDA, G. FAVIER, and J. C. M. MOTA. PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization. *Signal Processing*, 87(2) :337–351, February 2007.
- [22] W. FELLER. *An Introduction to Probability Theory and its Applications*, volume II. Wiley, 1966.
- [23] A. FERREOL and P. CHEVALIER. On the behavior of current second and higher order blind source separation methods for cyclostationary sources. *IEEE Trans. Sig. Proc.*, 48 :1712–1725, June 2000. erratum in vol.50, pp.990, Apr. 2002.
- [24] A. FERREOL, P. CHEVALIER, and L. ALBERA. Second-order blind separation of first- and second-order cyclostationary sources. *IEEE Trans. Sig. Proc.*, 52 :845–861, April 2004.
- [25] O. GRELLIER and P. COMON. Blind separation of discrete sources. *IEEE Signal Processing Letters*, 5(8) :212–214, August 1998.
- [26] R. A. HARSHMAN. Foundations of the Parafac procedure : Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16 :1–84, 1970. <http://publish.uwo.ca/harshman>.
- [27] R. A. HARSHMAN and M. LUNDY. PARAFAC : Parallel factor analysis. *Computational Stat. Data Analysis*, 18 :39–72, 1994.
- [28] A. M. KAGAN, Y. V. LINNIK, and C. R. RAO. *Characterization Problems in Mathematical Statistics*. Probability and Mathematical Statistics. Wiley, New York, 1973.
- [29] E. KOFIDIS and P. REGALIA. One the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM Jour. Matrix Ana. Appl.*, 23 :863–884, 2002.
- [30] J. B. KRUSKAL. Three-way arrays : Rank and uniqueness of trilinear decompositions. *Linear Algebra and Applications*, 18 :95–138, 1977.

- [31] J. L. LACOUME, P. O. AMBLARD, and P. COMON. *Statistiques d'ordre supérieur pour le traitement du signal*. Collection Sciences de l'Ingénieur. Masson, 1997.
- [32] L. De LATHAUWER. The decomposition of a tensor in a sum of rank- (R_1, R_2, R_3) terms. In *Workshop on Tensor Decompositions and Applications*, Marseille, France, 2005.
- [33] L. De LATHAUWER. The decomposition in block terms. In *ThRee-way methods In Chemistry and Psychology (TRICAP 2006)*, Chania, Crete, Greece, June 2006.
- [34] L. De LATHAUWER. Decompositions of a higher-order tensor in block terms — part II : Definitions and uniqueness. *SIAM J. Matrix Anal. Appl.*, 2008. submitted.
- [35] L. De LATHAUWER, J. CASTAING, and J.F. CARDOSO. Fourth-order cumulant-based identification of underdetermined mixtures. *IEEE Trans. Sig. Proc.*, 55(6) :2965–2973, June 2007.
- [36] L. De LATHAUWER, B. de MOOR, and J. VANDEWALLE. A multilinear singular value decomposition. *SIAM Jour. Matrix Ana. Appl.*, 21(4) :1253–1278, April 2000.
- [37] L. De LATHAUWER, B. de MOOR, and J. VANDEWALLE. On the best rank-1 and rank- (R_1, R_2, \dots, R_N) approximation of high-order tensors. *SIAM Jour. Matrix Ana. Appl.*, 21(4) :1324–1342, April 2000.
- [38] L-H. LIM and V. Da SILVA. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis Appl.*, 2008. to appear.
- [39] K. MADSEN, H. B. NIELSEN, and O. TINGLEFF. *Methods for Non-Linear Least Squares Problems*. Informatics and mathematical Modelling. Technical University of Denmark, 2004.
- [40] P. McCULLAGH. *Tensor Methods in Statistics*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1987.
- [41] D. NION and L. De LATHAUWER. A block factor analysis based receiver for blind multi-user access in wireless communications. In *Proc. ICASSP*, Toulouse, France, May 2006.
- [42] M. RAJIH, P. COMON, and R. HARSHMAN. Enhanced line search : A novel method to accelerate PARAFAC. *SIAM Journal on Matrix Analysis Appl.*, 2008. to appear.
- [43] C.R. RAO and S. MITRA. *Generalized Inverse of Matrices and Its Applications*. New York : Wiley, 1971.
- [44] L. ROTA, V. ZARZOSO, and P. COMON. Parallel deflation with alphabet-based criteria for blind source extraction. In *IEEE SSP'05*, Bordeaux, France, July, 17-20 2005.
- [45] N. D. SIDIROPOULOS and R. BRO. On the uniqueness of multilinear decomposition of N-way arrays. *Jour. Chemo.*, 14 :229–239, 2000.
- [46] N. D. SIDIROPOULOS, G. B. GIANNAKIS, and R. BRO. Blind PARAFAC receivers for DS-CDMA systems. *Trans. on Sig. Proc.*, 48(3) :810–823, March 2000.
- [47] A. STEGEMAN and N. D. SIDIROPOULOS. On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition. *Linear Algebra and Appl.*, 420 :540–552, 2007.
- [48] J. M. F. ten BERGE. Partial uniqueness in CANDECOMP/PARAFAC. *Jour. Chemometrics*, 18 :12–16, 2004.
- [49] A.-J. van der VEEN. Algebraic methods for deterministic blind beamforming. *Proceedings of the IEEE*, 86(10) :1987–2008, October 1998.
- [50] A. YEREDOR. Non-orthogonal joint diagonalization in the LS sense with application in blind source separation. *IEEE Trans. Signal Proc.*, 50(7) :1545–1553, 2002.
- [51] V. ZARZOSO and P. COMON. Optimal step-size constant modulus algorithm. *IEEE Trans. Com.*, January 2008. submitted in 2004.
- [52] V. ZARZOSO, L. ROTA, and P. COMON. Deflation parallele avec des contrastes APF pour l'extraction aveugle de sources. In *Gretsi'05*, Louvain La Neuve, Belgique, 5-10 sept 2005.