

Un algorithme ALS pour la décomposition conjointe de matrices complexes

Tual TRAININI, Eric MOREAU,

Université du Sud Toulon Var, LSEET, UMR CNRS 6017
Av. G. Pompidou, BP 56, 83162, La Valette du Var Cedex, France
trainini@univ-tln.fr, moreau@univ-tln.fr

Résumé – Dans ce papier, nous proposons un algorithme de type moindres carrés alternés pour la décomposition conjointe de deux ensembles de matrices. Le premier ensemble est constitué de matrices hermitiennes et le second de matrices complexes symétriques. Afin d’accélérer la vitesse de convergence, nous développons une approche complémentaire basée sur une mise à jour linéaire à pas optimal. Nous comparons les performances de notre algorithme à une méthode récente proposée dans la littérature grâce à des simulations informatiques.

Abstract – In this paper, an alternate least squares algorithm is proposed to the joint décomposition of two complex matrices sets. The first set corresponds to hermitian matrices while the second one corresponds to complex symmetric matrices. In order to accelerate the convergence speed, an enhanced line search complementary approach is developed. It is based on an optimal step size. Algorithm performances are compared in using computer simulations to récent algorithm.

1 Modèle et Problématique

On considère conjointement deux ensembles de matrices \mathcal{M}_1 et \mathcal{M}_2 . Le premier ensemble \mathcal{M}_1 est constitué de N_1 matrices complexes se décomposant sous la forme :

$$\mathbf{M}_i^{(1)} = \mathbf{A}\mathbf{D}_i^{(1)}\mathbf{A}^H + \mathbf{B}_i^{(1)}, \quad (1)$$

tandis que le deuxième ensemble \mathcal{M}_2 est constitué de N_2 matrices complexes se décomposant sous la forme :

$$\mathbf{M}_i^{(2)} = \mathbf{A}\mathbf{D}_i^{(2)}\mathbf{A}^T + \mathbf{B}_i^{(2)}. \quad (2)$$

$(\cdot)^H$ désigne la transposée-conjuguée de la matrice en argument et $(\cdot)^T$ la transposée de la matrice en argument. \mathbf{A} est une matrice complexe supposée de rang plein en colonne, de norme unité et de dimension $M \times N$. Les matrices $\mathbf{D}_i^{(j)}, \forall i = 1, \dots, N_j$ avec $j = 1, 2$ sont supposées diagonales et les matrices $\mathbf{B}_i^{(j)} \in \mathbb{C}^{M \times M}$ représentent des erreurs de modélisation liées par exemple à une estimation des matrices $\mathbf{M}_i^{(1)}$ et $\mathbf{M}_i^{(2)}$.

Le problème considéré est le problème de l’identification aveugle de la matrice \mathbf{A} et des matrices diagonales $\mathbf{D}_i^{(j)}, \forall i = 1, \dots, N_j$ avec $j = 1, 2$.

Lorsque le seul ensemble \mathcal{M}_1 est considéré, le problème est connu sous le nom de diagonalisation conjointe de matrices. De nombreux algorithmes ont été proposés pour résoudre ce problème, par exemple [1], [2] et [9].

Lorsque les deux ensembles précédents sont considérés en même temps, relativement peu d’algorithmes ont été proposés. Une première solution a été donnée dans [8] et un algorithme de gradient relatif a été proposé dans [7].

Notre approche s’inspire des travaux proposés dans [9] pour une part, et des travaux liés à la décomposition Parafac pour

des tenseurs [6] d’autre part. L’originalité principale provient de l’estimation de la matrice \mathbf{A} prenant en compte en même temps les deux ensembles de matrices \mathcal{M}_1 et \mathcal{M}_2 .

L’application en séparation de sources potentiellement non circulaires, [5], provient par exemple de la prise en compte conjointe de statistiques circulaires (par l’intermédiaire de matrices de corrélation à des retards différents) et de statistiques non circulaires (par l’intermédiaire de matrices de corrélation non conjuguées à des retards différents).

2 Algorithme proposé

L’algorithme proposé résout successivement un ensemble de systèmes linéaires au sens des moindres carrés. Cela est rendu possible par rapport aux décompositions précédentes lorsque l’on fixe toutes les matrices recherchées sauf une et lorsque l’on considère les matrices \mathbf{A}^H et \mathbf{A}^T comme ”indépendantes” de \mathbf{A} . Ce sera le cas de l’algorithme proposé.

2.1 ALS

Nous considérons un algorithme des moindres carrés alternés pour décomposer conjointement les deux ensembles de matrices définis en (1) et (2). L’objectif est d’estimer, de manière récursive, les matrices inconnues $\mathbf{D}_i^{(j)}, \mathbf{A}, \mathbf{A}^H$ et \mathbf{A}^T .

En fixant les valeurs de \mathbf{A}, \mathbf{A}^H et \mathbf{A}^T , on détermine $\mathbf{D}_i^{(j)}$. Une fois celle-ci obtenue, on la combine avec \mathbf{A}^H et \mathbf{A}^T (maintenues à leurs valeurs fixées précédemment) pour estimer \mathbf{A} . Enfin, il ne reste plus qu’à évaluer \mathbf{A}^H et \mathbf{A}^T en utilisant $\mathbf{D}_i^{(j)}$ et \mathbf{A} établies dans les étapes précédentes.

Pour plus de clarté dans les expressions à venir, nous définissons les arrangements de matrices suivants :

$$\mathbf{vdD}^{(j)} \triangleq [\text{vecdiag}(\mathbf{D}_1^{(j)}), \dots, \text{vecdiag}(\mathbf{D}_{N_j}^{(j)})], \quad (3)$$

$$\mathbf{vM}^{(j)} \triangleq [\text{vec}(\mathbf{M}_1^{(j)}), \dots, \text{vec}(\mathbf{M}_{N_j}^{(j)})], \quad (4)$$

$$\mathbf{M}_h \triangleq [\mathbf{M}_1^{(1)}, \dots, \mathbf{M}_{N_1}^{(1)}], \quad (5)$$

$$\mathbf{M}_t \triangleq [\mathbf{M}_1^{(2)}, \dots, \mathbf{M}_{N_2}^{(2)}], \quad (6)$$

$$\mathbf{D}_{\tilde{\mathbf{A}}} \triangleq [\mathbf{D}_1^{(j)} \tilde{\mathbf{A}}, \dots, \mathbf{D}_{N_j}^{(j)} \tilde{\mathbf{A}}]. \quad (7)$$

L'opérateur $\text{vec}(\cdot)$ empile toutes les colonnes d'une matrice dans un vecteur. $\text{vecdiag}(\cdot)$ récupère la diagonale d'une matrice dans un vecteur. Enfin, $\tilde{\mathbf{A}}$ servira à représenter les versions transposées et hermitiennes de \mathbf{A} .

2.1.1 Mise à jour de $\mathbf{D}_i^{(j)}$

En considérant la propriété suivante de l'opérateur $\text{vec}(\cdot)$

$$\text{vec}(\mathbf{X}\mathbf{Y}\mathbf{Z}) = (\mathbf{Z}^T \otimes \mathbf{X})\text{vec}(\mathbf{Y})$$

où \otimes représente le produit de Kronecker, nous pouvons en déduire $\forall i = 1, \dots, N_j$:

$$\begin{aligned} \text{vec}(\mathbf{M}_i^{(j)}) &= (\tilde{\mathbf{A}}^T \otimes \mathbf{A})\text{vec}(\mathbf{D}_i^{(j)}) \\ &= (\tilde{\mathbf{A}}^T \odot \mathbf{A})\text{vecdiag}(\mathbf{D}_i^{(j)}), \end{aligned} \quad (8)$$

où \odot symbolise le produit de Khatri-Rao. L'application de (8) aux ensembles (1) et (2), en utilisant les définitions (3) et (4), conduit à :

$$\mathbf{vM}^{(j)} = (\tilde{\mathbf{A}}^T \odot \mathbf{A})\mathbf{vdD}^{(j)}. \quad (9)$$

En se servant de $\mathbf{vdD}^{(j)} \in \mathbb{C}^{N \times N_j}$ qui contient tous les termes diagonaux de $\mathbf{D}_i^{(j)}$, il vient :

$$\mathbf{D}_i^{(j)} = \text{unvecdiag}\{(\tilde{\mathbf{A}}^T \odot \mathbf{A})^\dagger \mathbf{vM}^{(j)}\}. \quad (10)$$

L'opérateur $\text{unvecdiag}(\mathbf{vdD}^{(j)})$ reconstruit $\mathbf{vdD}^{(j)}$ sous la forme d'un ensemble de matrices diagonales $\mathbb{C}^{N \times N \times N_j}$, et $(\cdot)^\dagger$ représente la pseudo-inverse (Moore-Penrose) d'une matrice.

2.1.2 Mise à jour de \mathbf{A}

En ayant mis à jour les matrices $\mathbf{D}_i^{(j)}$, on peut passer à l'étape suivante de l'algorithme qui consiste à mettre à jour la matrice de mélange \mathbf{A} . Le dépliement horizontal des matrices définies en (5) et (6) permet de se rendre compte que la matrice \mathbf{A} est multipliée à droite par la concaténation de deux matrices introduites en (7) :

$$[\mathbf{M}_h, \mathbf{M}_t] = \mathbf{A}[\mathbf{D}_{A^H}, \mathbf{D}_{A^T}]. \quad (11)$$

On obtient alors directement une estimation de \mathbf{A} :

$$\mathbf{A} = [\mathbf{M}_h, \mathbf{M}_t][\mathbf{D}_{A^H}, \mathbf{D}_{A^T}]^\dagger. \quad (12)$$

2.1.3 Mise à jour de $\tilde{\mathbf{A}}$

La dernière étape de l'algorithme proposé consiste à déterminer les matrices \mathbf{A}^H et \mathbf{A}^T , exprimées dans la suite par $\tilde{\mathbf{A}}$. En dépliant verticalement les deux ensembles de matrices (1) et (2), on obtient la formulation suivante :

$$\begin{bmatrix} \mathbf{M}_1^{(j)} \\ \vdots \\ \mathbf{M}_{N_j}^{(j)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{D}_1^{(j)} \\ \vdots \\ \mathbf{A}\mathbf{D}_{N_j}^{(j)} \end{bmatrix} \tilde{\mathbf{A}}. \quad (13)$$

Finalement, on obtient l'estimation suivante

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}\mathbf{D}_1^{(j)} \\ \vdots \\ \mathbf{A}\mathbf{D}_{N_j}^{(j)} \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{M}_1^{(j)} \\ \vdots \\ \mathbf{M}_{N_j}^{(j)} \end{bmatrix}. \quad (14)$$

Ainsi, l'utilisation de ces trois étapes permet d'estimer itérativement l'ensemble des matrices inconnues mais tout particulièrement la matrice de mélange \mathbf{A} .

2.2 Line search

Un tel algorithme peut avoir des problèmes de vitesse de convergence, par exemple lorsque deux vecteurs colonne de \mathbf{A} sont quasiment colinéaires. Dans ce cas, nous proposons et développons une approche de type "line search" [6]. Le principe consiste à mettre à jour les matrices recherchées de la façon suivante (en ce qui concerne \mathbf{A}) :

$$\check{\mathbf{A}}^k = \check{\mathbf{A}}^{k-2} + \rho(\check{\mathbf{A}}^{k-1} - \check{\mathbf{A}}^{k-2}) = \check{\mathbf{A}}^{k-2} + \rho\mathbf{T}_{\check{\mathbf{A}}}. \quad (15)$$

en sachant bien évidemment que cela reste valable pour les matrices $\mathbf{D}_i^{(j)}$, \mathbf{A}^H et \mathbf{A}^T . Le pas $\rho \in \mathbb{R}^*$ peut être fixé par avance.

2.2.1 Enhanced Line Search (ELS)

La méthode présentée ci-dessus est significativement améliorée en calculant le pas de manière optimale. A cette fin, on considère le critère quadratique suivant :

$$\mathcal{J}_k(\check{\mathbf{A}}, \check{\mathbf{D}}_i^{(j)}) = \sum_{j=1}^2 \beta_j \sum_{i=1}^{N_j} \|\check{\mathbf{A}}^k \check{\mathbf{D}}_i^{(j)k} \check{\mathbf{A}}^k - \mathbf{M}_i^{(j)}\|_F^2, \quad (16)$$

où $\beta_1 = \alpha$, $\beta_2 = 1 - \alpha$, $\alpha \in [0, 1]$, $(\check{\cdot})$ correspond à l'estimée de (\cdot) et $\|\cdot\|_F$ représente la norme de Frobenius.

En rappelant que $\|\mathbf{A}\|_F^2 = \text{tr}\{\mathbf{A}^H \mathbf{A}\}$, (16) peut être écrit sous la forme d'un polynôme de degré 6 en ρ , et naturellement, sa dérivée en fonction de ρ amène un polynôme de degré 5 :

$$\mathcal{J}_k(\check{\mathbf{A}}, \check{\mathbf{D}}_i^{(j)}) = \sum_{l=0}^6 \rho^l a_l, \quad (17)$$

$$\frac{\partial(\mathcal{J}_k(\check{\mathbf{A}}, \check{\mathbf{D}}_i^{(j)}))}{\partial \rho} = \sum_{l=0}^5 (l+1)\rho^l a_{l+1}. \quad (18)$$

Dans un but de concision, les indices $(k-2)$ et (\cdot) sont implicites. On montre que les coefficients a_l prennent les valeurs suivantes :

$$a_l = \sum_{m,n=0}^3 \text{tr}\{\mathbf{Z}_m \mathbf{Z}_n\} \quad m+n=l, \quad (19)$$

$$\begin{cases} \mathbf{Z}_0 = \sum_{j=1}^2 \beta_j \sum_{i=1}^{N_j} \text{tr}\{\mathbf{A} \mathbf{D}_i^{(j)} \tilde{\mathbf{A}}\} \\ \mathbf{Z}_1 = \sum_{j=1}^2 \beta_j \sum_{i=1}^{N_j} \text{tr}\{\mathbf{A} \mathbf{D}_i^{(j)} \mathbf{T}_{\tilde{\mathbf{A}}} + \mathbf{A} \mathbf{T}_{\mathbf{D}_i^{(j)}} \tilde{\mathbf{A}} \\ + \mathbf{T}_{\mathbf{A}} \mathbf{D}_i^{(j)} \tilde{\mathbf{A}}\} \\ \mathbf{Z}_2 = \sum_{j=1}^2 \beta_j \sum_{i=1}^{N_j} \text{tr}\{\mathbf{A} \mathbf{T}_{\mathbf{D}_i^{(j)}} \mathbf{T}_{\tilde{\mathbf{A}}} + \mathbf{T}_{\mathbf{A}} \mathbf{D}_i^{(j)} \mathbf{T}_{\tilde{\mathbf{A}}} \\ + \mathbf{T}_{\mathbf{A}} \mathbf{T}_{\mathbf{D}_i^{(j)}} \tilde{\mathbf{A}}\} \\ \mathbf{Z}_3 = \sum_{j=1}^2 \beta_j \sum_{i=1}^{N_j} \text{tr}\{\mathbf{T}_{\mathbf{A}} \mathbf{T}_{\mathbf{D}_i^{(j)}} \mathbf{T}_{\tilde{\mathbf{A}}}\} \end{cases}$$

La racine de (18) qui minimise (17) sera donc choisie comme pas ρ .

3 Simulations

Les simulations présentées illustrent les performances de l'algorithme proposé. Il sera comparé à une extension aux 2 ensembles considérés de l'algorithme FAJD suggéré dans [8].

La matrice $\mathbf{A} \in \mathbb{C}^{M \times N}$, ainsi que les N sources complexes \mathbf{S} , sont générées aléatoirement, suivant une loi uniforme de moyenne nulle. La variance est unitaire pour \mathbf{A} et égale à σ_s^2 pour les sources. Le bruit \mathbf{B} est créé suivant une loi gaussienne de moyenne nulle et de variance σ_b^2 . A partir de cela, nous définissons le rapport signal sur bruit (en dB) par RSB = $10 \log_{10}(\frac{\sigma_s^2}{\sigma_b^2})$.

On considère un modèle de signal directement sous la forme vectorielle :

$$\mathbf{X} = \mathbf{A} \mathbf{S} + \mathbf{B}. \quad (20)$$

Les k ensembles de matrices définis en (1) et (2) sont créés en utilisant des statistiques cumulantes d'ordre quatre, estimées en utilisant $N_e = 2^{15}$ échantillons :

$$(m_{ij}^{(1)})_k = \text{Cum}(x_i, x_j^*, x_k, x_k^*), \quad (21)$$

$$(m_{ij}^{(2)})_k = \text{Cum}(x_i, x_j, x_k, x_k^*), \quad (22)$$

où $(\cdot)^*$ correspond au conjugué de (\cdot) et $1 < i, j, k < M$. Nous obtenons donc deux ensembles composés de M matrices.

Pour comparer les algorithmes, nous utilisons un indice de performance, présenté dans [3] et défini comme suit :

$$\begin{aligned} I(\mathbf{G}) &= \frac{1}{r(r-1)} \sum_{i=1}^r \left(\sum_{j=1}^r \frac{\|\mathbf{G}_{i,j}\|_F^2}{\max_l \|\mathbf{G}_{i,l}\|_F^2} - 1 \right) \\ &+ \frac{1}{r(r-1)} \sum_{j=1}^r \left(\sum_{i=1}^r \frac{\|\mathbf{G}_{i,j}\|_F^2}{\max_l \|\mathbf{G}_{l,j}\|_F^2} - 1 \right), \quad (23) \end{aligned}$$

où $\mathbf{G} = \mathbf{B} \mathbf{A}$ et $\mathbf{B} = \tilde{\mathbf{A}}^\dagger$ est calculée à partir de l'estimation de \mathbf{A} .

Deux cas seront considérés dans les simulations : le cas carré et le cas rectangulaire. Afin de considérer le cas rectangulaire pour FAJD, une projection sur l'espace "signal" associé aux N plus grandes valeurs propres de la matrice obtenue par concaténation des 2 ensembles (1) et (2) est d'abord effectuée. Ceci nous ramène alors au cas carré. L'initialisation des algorithmes se fera en utilisant le projecteur défini ci-dessus, et dont l'espace correspond donc au rang de la matrice \mathbf{A} .

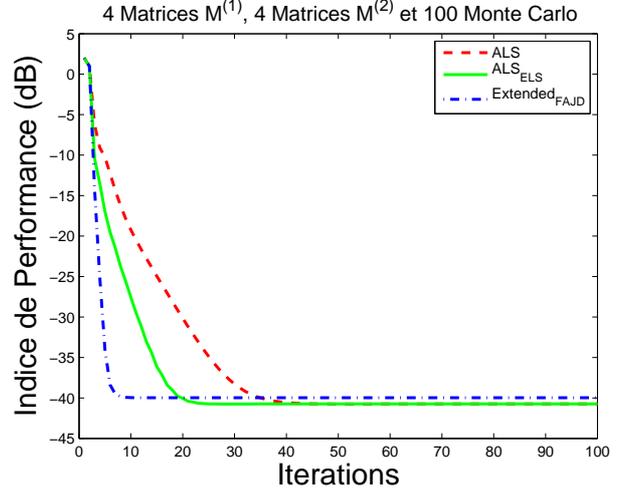


FIGURE 1 – Indice de Performance en fonction du RSB pour le cas carré

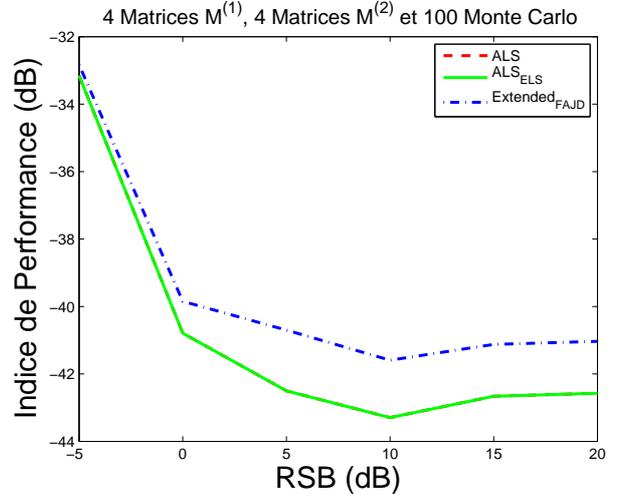


FIGURE 2 – Indice de Performance en fonction du RSB pour le cas rectangulaire

Dans la figure 1, le cas carré est considéré, avec $M = N = 4$ et $N_1 = N_2 = 4$. L'indice de Performance est moyenné sur 100 réalisations indépendantes (Monte Carlo). On remarque que l'algorithme ALS_ELS fournit des résultats similaires à ALS mais surpasse extendedFAJD. Ceci est logique. En effet, le pas optimal ne change pas les performances, mais seulement la vitesse de convergence.

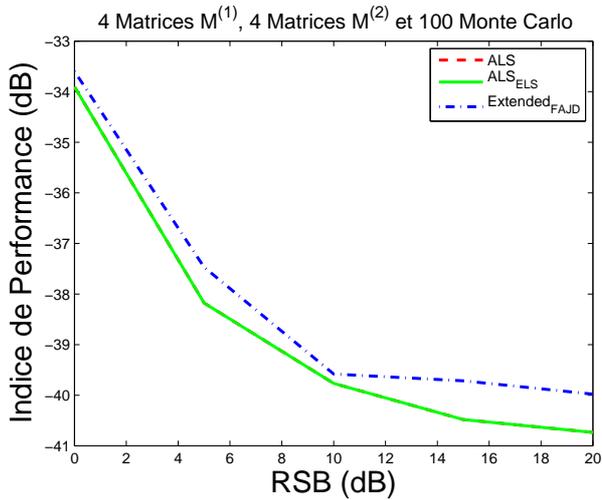


FIGURE 3 – Indice de Performance en fonction du nombre d'itérations, RSB= 20dB, cas carré

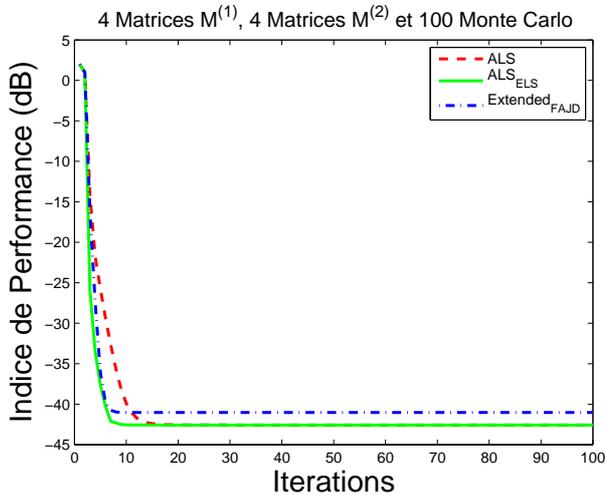


FIGURE 4 – Indice de Performance en fonction du nombre d'itérations, RSB= 20dB, cas rectangulaire

Le cas rectangulaire est présenté dans la figure 2, avec $M = 4$, $N = 2$ et $N_1 = N_2 = 4$, moyenné aussi sur 100 réalisations indépendantes. Les algorithmes proposés ont également de meilleures performances que extendedFAJD.

Les figures 3 et 4 permettent de comparer les vitesses de convergence des algorithmes. Dans le cas carré, extendedFAJD converge plus rapidement. Cependant, le cas rectangulaire montre que l'ALS combiné avec le pas optimal converge plus rapidement que les deux autres algorithmes.

4 Conclusion

Un nouvel algorithme de décomposition conjointe de matrices est proposé dans ce papier. L'utilisation des ensembles de matrices hermitiennes et symétriques en même temps par l'in-

termédiaire d'un algorithme de type ALS couplé à une mise à jour linéaire à pas optimal, permet d'obtenir de bons résultats en terme de performances.

Références

- [1] A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique based on second order statistics," *IEEE Trans. Signal Process.*, Vol. 45, No. 2, pp. 434–444, Février 1997.
- [2] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non Gaussian signal," *IEE Proc.-F*, Vol. 40, pp. 362–370, 1993.
- [3] H. Ghennioui, E.-M. Fadaili, N. Thirion-Moreau, A. Adib and E. Moreau, "A non-unitary joint block diagonalization algorithm for blind separation of convolutive mixtures of sources", *IEEE Signal Process. Letters*, Vol. 14, No. 11, pp. 860–863, Novembre 2007.
- [4] X-L Li and X-D Zhang, "Nonorthogonal Joint Diagonalization Free of Degenerate Solution", *IEEE Trans. on Sig. Pro.*, Vol. 55, No. 5, Mai 2007
- [5] B. Picinbono, "On circularity," *IEEE Trans. Signal Process.*, Vol. 42, No. 12, pp. 3473–3482, Decembre 1994.
- [6] M. Rajih, P. Comon, and R. Harshman, "Enhanced Line Search : A Novel Method to Accelerate PARAFAC", *SIAM Journal on Matrix Analysis Appl.*, Vol.30, No. 3, pp. 1148–1171, Septembre 2008.
- [7] T. Trainini, X.-L. Li, E. Moreau, and T. Adali, "A relative gradient algorithm for joint decompositions of complex matrices", *18th European Signal Processing Conference (EUSIPCO-2010)*, Aalborg, Denmark, pp. 1073–1076, Août 2010.
- [8] W.-J. Zeng, X.-L. Li, X.-D. Zhang and X. Jiang, "An improved signal-selective direction finding algorithm using second-order cyclic statistics", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 2141–2144, 2009.
- [9] A. Yeredor, "Non-Orthogonal Joint Diagonalization in the Least-Squares Sense With Application in Blind Source Separation", *IEEE Trans. on Sig. Pro.*, Vol. 50, No. 7, pp. 1545–1553, Juillet 2002.