



HAL
open science

Diagonalisation conjointe par similitude avec contrainte de non négativité sur les valeurs propres

Rémi André, Xavier Luciani, Eric Moreau

► **To cite this version:**

Rémi André, Xavier Luciani, Eric Moreau. Diagonalisation conjointe par similitude avec contrainte de non négativité sur les valeurs propres. GRETSI 2017, Sep 2017, Juan-les-Pins, France. hal-01862233

HAL Id: hal-01862233

<https://hal.science/hal-01862233>

Submitted on 27 Aug 2018

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

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

Diagonalisation conjointe par similitude avec contrainte de non négativité sur les valeurs propres

Rémi ANDRÉ, Xavier LUCIANI, Eric MOREAU

Laboratoire des Sciences de l'Information et des Systèmes, UMR CNRS 7296
Aix Marseille Université, CNRS, ENSAM, 13397 Marseille et Université de Toulon, 83957 La Garde, France
randre@univ-tln.fr, luciani@univ-tln.fr, moreau@univ-tln.fr

Résumé – Nous proposons un algorithme de diagonalisation conjointe par similitude d'un ensemble de matrices avec contrainte de non négativité sur les valeurs propres. Ce problème est au cœur de récentes méthodes de dé-mélange de signaux multidimensionnels. L'approche proposée garantit la non négativité des valeurs propres estimées et nous montrons à l'aide de simulations numériques que l'utilisation de cette contrainte permet d'améliorer l'estimation des valeurs propres.

Abstract – In this paper we propose an algorithm for the joint eigenvalue decomposition of a matrix set with a non-negativity constraint on the eigenvalues. This kind of problem notably occurs for the separation of multidimensional signals. The proposed approach ensures the non-negativity of the estimated eigenvalues besides, our numerical simulations show that using the non-negativity constraint allows to improve the estimation of the eigenvalues.

1 Introduction

La Décomposition Canonique Polyadique de tenseur (DCP) apparait aujourd'hui comme un outils efficace de dé-mélange de signaux multidimensionnels [1]. Mathématiquement, la DCP consiste en une décomposition multilinéaire d'un tenseur d'ordre quelconque à l'aide de matrices facteurs. Une catégorie d'algorithmes récents de DCP est basée sur une réécriture de la décomposition comme un problème de Diagonalisation Conjointe de matrices par Similitude (DCS) [2–4]. Le problème de DCS est le suivant. Nous disposons d'un jeu $\{M^{(k)}\}$ de K matrices réelles carrées, $M^{(k)}$, représentant des données bruitées. Nous supposons que les matrices $M^{(k)}$ sont approximativement diagonalisables dans la même base de vecteurs propres A . On note $D^{(k)}$ les matrices diagonales des valeurs propres de sorte que nous ayons :

$$M^{(k)} = AD^{(k)}A^{-1} + E^{(k)}, \quad \forall k = 1, \dots, K \quad (1)$$

où les K matrices $E^{(k)}$ représentent le bruit. L'objectif est alors d'estimer la matrice A et les matrices $D^{(k)}$ en diagonalisant conjointement les matrices $M^{(k)}$. Les algorithmes de DCS sont des algorithmes itératifs basés sur une succession de mises à jour multiplicatives de l'estimation de la matrice A . Ils sont inspirés par la méthode de Jacobi et les algorithmes de diagonalisation conjointe par congruence [5, 6]. Ils diffèrent entre eux principalement par la paramétrisation choisie pour estimer A : décomposition polaire dans [3, 7, 8], factorisation LU dans [9] et factorisation de type LU simplifiée dans [10]. Appliqués au calcul de la DCP, ces algorithmes permettent de remonter aux matrices facteurs en utilisant soit la matrice A [2,3] soit les matrices $D^{(k)}$ [4]. Les algorithmes de DCP basés sur la DCS sont

généralement plus rapides que les algorithmes de DCP classiques tels que l'ALS [3]. Dans plusieurs applications, comme la spectroscopie de fluorescence [12], la DCP est appliquée sur des données non négatives. Lors de la réécriture de la DCP non négative en DCS, la propriété de non négativité d'une ou plusieurs matrices facteurs se transmet directement aux matrices $D^{(k)}$. Nous proposons donc ici un algorithme de DCS utilisant cette propriété. La première étape de l'algorithme consiste à transformer le jeu de matrices à diagonaliser en un jeu de matrices semi-définies positives dont la diagonalisation dans la deuxième étape fournira une estimation des matrices $D^{(k)}$, non négative par construction.

2 Algorithme proposé

2.1 Réécriture du problème

Toute matrice carrée inversible pouvant se décomposer comme le produit d'une matrice triangulaire inférieure et d'une matrice orthogonale, il existe L et Q respectivement triangulaire inférieure et orthogonale telles que $A = LQ$. Ainsi le jeu d'équations (1) peut se réécrire comme :

$$M^{(k)} = LQD^{(k)}Q^T L^{-1} + E^{(k)}, \quad \forall k. \quad (2)$$

En posant alors

$$S^{(k)} = L^{-1}M^{(k)}L, \quad \forall k, \quad (3)$$

il vient

$$S^{(k)} = QD^{(k)}Q^T + L^{-1}E^{(k)}L, \quad \forall k. \quad (4)$$

Sans bruit, les matrices $\mathbf{S}^{(k)}$ sont donc symétriques. De plus les éléments diagonaux des matrices $\mathbf{D}^{(k)}$ étant non négatifs par hypothèse, $\{\mathbf{S}^{(k)}\}$ forme un jeu de matrices approximativement semi-définies positives.

L'idée est alors de chercher dans un premier temps une matrice \mathbf{L} réalisant la symétrisation conjointe de $\{\mathbf{M}^{(k)}\}$. Cette première étape de symétrisation nous permet de transformer le problème de DCS en un problème plus simple de diagonalisation conjointe orthogonale résumé par l'équation (4). De plus, nous préservons ainsi la contrainte de non négativité sur les matrices diagonales. La deuxième et dernière étape de l'algorithme consistera donc à résoudre ce problème de diagonalisation conjointe sous contrainte de non négativité des valeurs propres.

2.2 Étape de symétrisation conjointe

2.2.1 Principe

Nous proposons pour calculer $\{\mathbf{S}^{(k)}\}$ un algorithme de balayage par paires. Le principe est d'estimer \mathbf{L} comme un produit de matrices élémentaires. Nous appelons ici matrice élémentaire, toute matrice de structure connue et dont la connaissance de l'ensemble des coefficients ne dépend que d'un seul paramètre (scalaire). Nous utiliserons deux types de matrices élémentaires :

- Les matrices triangulaires inférieures élémentaires, notées $\mathbf{L}^{(ij)}(x)$, ($i > j$) : égales à la matrice identité sauf pour le coefficient $L_{i,j}^{(ij)}$ qui prend la valeur $x \in \mathbb{R}$.
- Les matrices diagonales élémentaires, notées $\mathbf{\Lambda}^{(j)}(a)$: égales à la matrice identité sauf pour le coefficient $\Lambda_{j,j}^{(j)}$ qui prend la valeur $a \in \mathbb{R}$.

La proposition suivante donne alors une décomposition simple des matrices triangulaires inférieures :

Proposition 2.1. *Soit \mathbf{L} une matrice triangulaire inférieure de taille N alors, il existe $N(N-1)/2$ matrices triangulaires inférieures élémentaires $\{\mathbf{L}^{(ij)}(x^{(ij)})\}$ et N matrices diagonales élémentaires $\{\mathbf{\Lambda}^{(j)}(a^{(j)})\}$ telles que :*

$$\mathbf{L} = \prod_{j=1}^N \prod_{i=j+1}^N \mathbf{L}^{(ij)}(x^{(ij)}) \mathbf{\Lambda}^{(j)}(a^{(j)}). \quad (5)$$

Le calcul de chaque matrice élémentaire s'effectuera alors par optimisation successive d'un critère de symétrisation ne dépendant que d'un seul paramètre.

2.2.2 Mise à jour multiplicative de $\mathbf{M}^{(k)}$

On appelle mise à jour multiplicative de $\{\mathbf{M}^{(k)}\}$ par la matrice \mathbf{X} l'ensemble d'opérations suivantes :

$$\mathbf{M}^{(k)} \leftarrow \mathbf{X} \mathbf{M}^{(k)} \mathbf{X}^{-1} \quad \forall k = 1, \dots, K \quad (6)$$

On associe ici à une mise à jour multiplicative le critère de

symétrisation suivant :

$$\begin{cases} C(\mathbf{X}) = \sum_{k=1}^K \sum_{p=1}^{N-1} \sum_{q=p+1}^N (N_{p,q}^{(k)} - N_{q,p}^{(k)})^2 \\ \text{où } N^{(k)} = \mathbf{X} \mathbf{M}^{(k)} \mathbf{X}^{-1}. \end{cases} \quad (7)$$

2.2.3 Construction de $\{\mathbf{S}^{(k)}\}$ et estimation de \mathbf{L}

D'après la proposition 2.1 et l'équation (3), la symétrisation de $\{\mathbf{M}^{(k)}\}$ peut s'effectuer à l'aide de mises à jour multiplicatives successives par les matrices des ensembles $\{\mathbf{L}^{(ij)}(x^{(ij)})\}$ et $\{\mathbf{\Lambda}^{(j)}(a^{(j)})\}$. Plus précisément, à chaque paire d'indices (i, j) , $j = 1, \dots, N-1, i = j+1, \dots, N$ correspond une mise à jour par $\mathbf{L}^{(ij)}(x^{(ij)})$. De plus lorsque $i = N$, la mise à jour par $\mathbf{L}^{(Nj)}(x^{(Nj)})$ est immédiatement suivie par une mise à jour par $\mathbf{\Lambda}^{(j)}(a^{(j)})$ et enfin la dernière mise à jour est celle par $\mathbf{\Lambda}^{(N)}(a^{(N)})$.

Nous voulons que chaque mise à jour rende les matrices $\mathbf{M}^{(k)}$ de plus en plus symétriques. Ainsi, pour chaque mise à jour multiplicative, la matrice de mise à jour est choisie comme celle (ou parmi celles) minimisant le critère de symétrisation associé. Les matrices de mise à jour étant élémentaires, le critère aura pour expression une fonction simple à une variable. Nous détaillons maintenant le calcul effectif des deux types de matrices de mise à jour.

Considérons d'abord une mise à jour par une matrice élémentaire $\mathbf{L}^{(ij)}(x)$, c'est à dire $\mathbf{X} = \mathbf{L}^{(ij)}(x)$ dans (7), soit :

$$\mathbf{N}^{(k)} = \mathbf{L}^{(ij)}(x) \mathbf{M}^{(k)} \left(\mathbf{L}^{(ij)}(x) \right)^{-1}. \quad (8)$$

On vérifie facilement que la mise à jour n'affecte que les lignes i et les colonnes j et donc

$$C(\mathbf{L}^{(ij)}(x)) = \sum_{k=1}^K \left(\sum_{\substack{p \neq i \\ p \neq j}} \left((M_{p,i}^{(k)} - N_{i,p}^{(k)})^2 + (N_{p,j}^{(k)} - M_{j,p}^{(k)})^2 \right) + (N_{i,j}^{(k)} - M_{j,i}^{(k)})^2 \right) + R \quad (9)$$

où R est une constante ne dépendant pas de x et

$$\begin{cases} \forall p \neq j, N_{i,p}^{(k)} = -x M_{j,p}^{(k)} + M_{i,p}^{(k)}, \\ \forall p \neq i, N_{p,j}^{(k)} = x M_{p,i}^{(k)} + M_{p,j}^{(k)}, \\ N_{i,j}^{(k)} = -x^2 M_{j,i}^{(k)} + x(M_{i,i}^{(k)} - M_{j,j}^{(k)}) + M_{i,j}^{(k)}. \end{cases} \quad (10)$$

Ainsi, nous avons :

$$C(\mathbf{L}^{(ij)}(x)) = P(x) = \alpha_4 x^4 + \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0 + R \quad (11)$$

avec :

$$\begin{aligned} \alpha_4 &= \sum_{k=1}^K (M_{j,i}^{(k)})^2; & \alpha_3 &= -2 \sum_{k=1}^K M_{j,i}^{(k)} (M_{i,i}^{(k)} - M_{j,j}^{(k)}); \\ \alpha_2 &= \sum_{k=1}^K \left(-2 M_{j,i}^{(k)} (M_{i,j}^{(k)} - M_{j,i}^{(k)}) + (M_{i,i}^{(k)} - M_{j,j}^{(k)})^2 + \right. \end{aligned} \quad (12)$$

$$\sum_{\substack{p \neq i \\ p \neq j}} \left((M_{j,p}^{(k)})^2 + (M_{p,i}^{(k)})^2 \right); \quad (13)$$

$$\alpha_1 = 2 \sum_{k=1}^K \left((M_{i,i}^{(k)} - M_{j,j}^{(k)}) (M_{i,j}^{(k)} - M_{j,i}^{(k)}) + \sum_{\substack{p \neq i \\ p \neq j}} \left(M_{p,i}^{(k)} (M_{p,j}^{(k)} - M_{j,p}^{(k)}) - M_{j,p}^{(k)} (M_{i,p}^{(k)} - M_{p,i}^{(k)}) \right) \right). \quad (14)$$

Nous choisisons donc $x^{(ij)}$ comme la racine réelle de la dérivée de $P(x)$ minimisant $P(x)$ (les expressions des constantes α_0 et R ne sont donc pas utiles ici).

Intéressons nous maintenant au calcul des matrices $\Lambda^{(j)}(a^{(j)})$. Le principe est similaire. Lors de la mise à jour par la matrice élémentaire $\Lambda^{(j)}(a)$ ($\mathbf{X} = \Lambda^{(j)}(a)$ dans (7)), nous avons cette fois :

$$\mathbf{N}^{(k)} = \Lambda^{(j)}(a) \mathbf{M}^{(k)} \left(\Lambda^{(j)}(a) \right)^{-1}. \quad (15)$$

On montre alors facilement que

$$\mathcal{C}(\Lambda^{(j)}(a)) = \sum_{k=1}^K \sum_{p \neq j} (N_{j,p}^{(k)} - N_{p,j}^{(k)})^2 + R' \quad (16)$$

$$= \sum_{k=1}^K \sum_{p \neq j} \left(\frac{(M_{j,p}^{(k)})^2}{a} - a (M_{p,j}^{(k)})^2 \right)^2 + R' \quad (17)$$

$$= f(a) + R' \quad (18)$$

où R' est une constante ne dépendant pas de a . Et donc :

$$\forall a \neq 0, f'(a) = 0 \Leftrightarrow a^4 = \frac{\sum_{k=1}^K \sum_{p \neq j} (M_{j,p}^{(k)})^2}{\sum_{k=1}^K \sum_{p \neq j} (M_{p,j}^{(k)})^2}. \quad (19)$$

$$f(a) \text{ étant paire, il vient } a^{(j)} = \left(\frac{\sum_{k=1}^K \sum_{p \neq j} (M_{j,p}^{(k)})^2}{\sum_{k=1}^K \sum_{p \neq j} (M_{p,j}^{(k)})^2} \right)^{\frac{1}{4}}.$$

Il est clair que cette stratégie d'optimisation disjointe des matrices élémentaires n'est pas équivalente à l'optimisation conjointe des coefficients de la matrice \mathbf{L} recherchée. Par conséquent, rien ne garantit qu'à l'issu des $N(N+1)/2$ mises à jour, $\{\mathbf{M}^{(k)}\}$ soit symétrisé de manière optimale. C'est pourquoi l'ensemble de ces mises à jour doit être répété jusqu'à satisfaire un critère d'arrêt. Un balayage (ou sweep) complet des paires d'indices (i, j) ne constitue donc qu'une itération de l'algorithme de symétrisation. La convergence de l'algorithme vers une solution satisfaisante est basée sur le fait que la matrice identité fait partie de l'ensemble des matrices triangulaires inférieures élémentaires et de l'ensemble des matrices diagonales élémentaires. Par conséquent, à chaque mise à jour multiplicative et donc, par extension, à chaque itération, le critère de symétrisation ne peut que diminuer ou rester inchangé. Le critère d'arrêt sera donc lié à l'évolution du critère de symétrisation. Ici nous choisisons de prendre comme critère d'arrêt une diminution relative de la valeur du critère de symétrisation \mathcal{C} entre deux itérations successives, inférieure à 10^{-6} et un nombre d'itérations maximal de 150. On note $\hat{\mathbf{L}}$ la matrice triangulaire estimée par cet algorithme de symétrisation et $\hat{\mathbf{S}}^{(k)}$ les matrices symétrisées.

2.3 Étape de diagonalisation conjointe

En présence de bruit, les matrices $\hat{\mathbf{S}}^{(k)}$ estimées à l'étape précédente ne sont qu'approximativement semi-définies positives. Afin de remédier à ce problème, nous construisons le jeu de matrices $\{\hat{\mathbf{S}}_2^{(k)}\}$ en calculant :

$$\hat{\mathbf{S}}_2^{(k)} = (\hat{\mathbf{S}}^{(k)})^\top \hat{\mathbf{S}}^{(k)} \quad \forall k. \quad (20)$$

Les $\hat{\mathbf{S}}_2^{(k)}$ sont exactement semi-définies positives par construction et l'équation (4) est remplacée par :

$$\hat{\mathbf{S}}_2^{(k)} = \mathbf{Q}(\mathbf{D}^{(k)})^2 \mathbf{Q}^\top + \mathbf{E}'^{(k)} \quad \forall k \quad (21)$$

où le terme $\mathbf{E}'^{(k)}$ est lié au bruit. Nous obtenons alors un problème de diagonalisation conjointe orthogonale d'un jeu de matrices semi-définies positives pour lequel Pham a proposé dans [11] un algorithme dont il a démontré la convergence.

Nous proposons d'utiliser ici cet algorithme sur le jeu $\{\hat{\mathbf{S}}_2^{(k)}\}$ afin d'estimer $\{(\mathbf{D}^{(k)})^2\}$. Nous estimons alors $\{\mathbf{D}^{(k)}\}$ sous contrainte de positivité en prenant les racines carrées positives des matrices précédemment estimées.

Ce détour par $\{\hat{\mathbf{S}}_2^{(k)}\}$ permet d'utiliser la contrainte de non négativité pour améliorer l'estimation des valeurs propres (et garantir leur non négativité). En revanche pour l'estimation de la matrice \mathbf{Q} , il est préférable d'utiliser un algorithme de diagonalisation conjointe orthogonale sans contrainte tel que JADE [5] directement sur le jeu $\{\hat{\mathbf{S}}^{(k)}\}$. En notant $\hat{\mathbf{Q}}$ la matrice ainsi estimée, nous estimons finalement la matrice de vecteurs propres comme : $\hat{\mathbf{A}} = \hat{\mathbf{L}}\hat{\mathbf{Q}}$. Nous appelons cet algorithme « Joint Diagonalisation based on Joint Symetrisation » (JDJS).

3 Simulations numériques

Nous comparons ici la méthode proposée avec les algorithmes de DCS JDJM [3] et SH-RT [8]. Un jeu de matrices à diagonaliser est construit de la manière suivante : nous générons aléatoirement un ensemble de K matrices $\mathbf{D}^{(k)}$ diagonales de dimensions N suivant une loi uniforme sur l'intervalle $[0; 1]$, une matrice \mathbf{A} de dimensions N suivant une loi normale, ainsi que les K matrices de bruit $\mathbf{E}^{(k)}$ de dimension N suivant une loi uniforme sur l'intervalle $[0; 1]$.

Afin de comparer les différents algorithmes testés, nous utilisons deux indices de performances :

$$r_A = \frac{\|\mathbf{A} - \hat{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}; \quad r_D = \frac{1}{K} \sum_{k=1}^K \frac{\|\mathbf{d}^{(k)} - \hat{\mathbf{d}}^{(k)}\|_2}{\|\mathbf{d}^{(k)}\|_2}, \quad (22)$$

où $\mathbf{d}^{(k)}$ et $\hat{\mathbf{d}}^{(k)}$ sont respectivement les vecteurs contenant les diagonales des matrices $\mathbf{D}^{(k)}$ vraies et estimées et $\|\cdot\|_F$ désigne la norme de Frobenius. L'indice r_A permet de mesurer la qualité de l'estimation de la matrice de vecteurs propres. L'indice r_D , quant à lui, permet de mesurer la qualité de l'estimation des valeurs propres. Nous comparons les différentes méthodes sur des ensembles de $K = 10$ matrices de dimension

$N = 5$ en faisant varier le Rapport Signal sur Bruit (RSB) de 10 dB à 70 dB par pas de 5 dB. Pour chaque RSB, 100 simulations sont effectuées, à chaque simulation un nouveau jeu de matrices est construit.

L'évolution de la valeur moyenne du critère r_A est représentée sur la figure 1. Il apparaît que les résultats de JDJS pour l'estimation de A sont comparables à ceux de JDJM et meilleurs que ceux de SH-RT quelque soit la valeur du RSB. L'évolution de la valeur moyenne de r_D est donnée figure 2. Nous pouvons constater que JDJS offre ici clairement les meilleures performances pour tous les niveaux de bruit considérés. Ainsi la méthode proposée permet d'améliorer l'estimation moyenne des valeurs propres tout en gardant pour l'estimation des vecteurs propres le même niveau de performance moyenne que l'algorithme le plus précis.

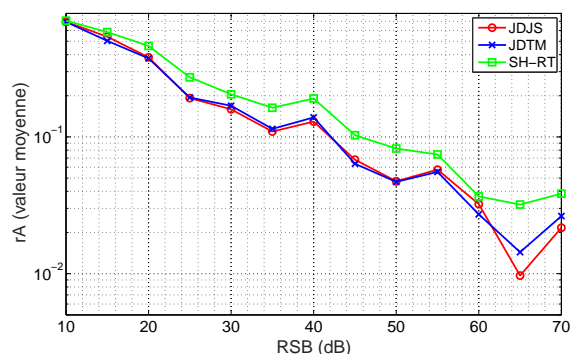


FIGURE 1 – Evolution de la valeur moyenne du critère r_A en fonction du RSB

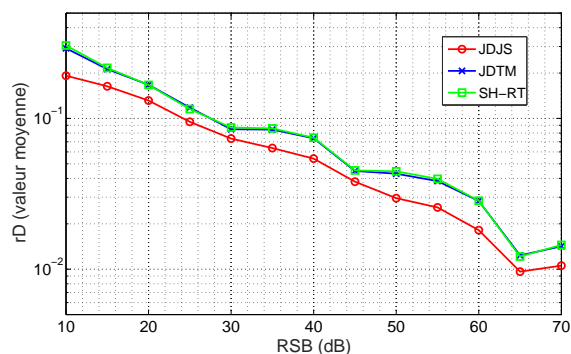


FIGURE 2 – Evolution de la valeur moyenne du critère r_D en fonction du RSB

4 Conclusion et perspective

Nous avons donc proposé un algorithme de diagonalisation conjointe de matrices par similitude pouvant intégrer une contrainte de non négativité sur les valeurs propres. Nos premières simulations numériques montrent que la prise en compte de cette contrainte permet d'améliorer l'estimation des valeurs pro-

pres tout en garantissant leur non négativité. L'étape naturelle suivante sera d'appliquer cet algorithme à la décomposition canonique polyadique de tenseurs non négatifs.

Références

- [1] T.G. Kolda et B.W. Bader, Tensor Decompositions and Applications, SIAM Rev, 2009, p. 455-500.
- [2] F. Roemer et M. Haardt, A closed-form solution for multilinear PARAFAC decompositions, SAM 08, Fifth IEEE Sensor Array and Multichannel Signal Processing Workshop, 2008, p. 487-491.
- [3] X. Luciani et L. Albera, Canonical Polyadic Decomposition based on joint eigenvalue decomposition, Chemometrics and Intelligent Laboratory Systems, 2014, V. 132 (0), p. 152 - 167.
- [4] S. Hajipour, L. Albera, M.B. Shamsollahi et I. Merlet, Canonical polyadic decomposition of complex-valued multi-way arrays based on simultaneous Schur decomposition, ICASSP 2013, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, p. 4178-4182.
- [5] J.F. Cardoso et A. Souloumiac, Blind beamforming for non-Gaussian signals, IEE Proceedings F - Radar and Signal Processing 1993, V. 140 (6), p. 362-370.
- [6] G. Chabriel, M. Kleinstueber, E. Moreau, H. Shen, P. Tichavsky et A. Yeredor, Joint Matrices Decompositions and Blind Source Separation. A Survey of Methods, Identification and Applications, IEEE Signal Processing Magazine 2014, V. 31, (3), p. 34-43.
- [7] R. Iferroudjene, K. Abed-Meraim et A. Belouchrani, A new Jacobi-like method for joint diagonalization of arbitrary non-defective matrices, Applied Mathematics and Computation, 2009, V. 211 (2), p. 363-373.
- [8] T. Fu et X. Gao, Simultaneous Diagonalization with Similarity Transformation for Non-defective Matrices, ICASSP 2006, 2006 IEEE International Conference on Acoustics Speech and Signal Processing, 2006, V. 4, p. 1137-1140.
- [9] X. Luciani et L. Albera, Joint Eigenvalue Decomposition of Non-Defective Matrices Based on the LU Factorization With Application to ICA, IEEE Transactions on Signal Processing, 2015, V. 63 (17), p. 4594-4608.
- [10] R. Andre, T. Trainini, X. Luciani et E. Moreau, A fast algorithm for joint eigenvalue decomposition of real matrices, European Signal Processing Conference (EU-SIPCO'2015), Nice, France, 2015, p. 1326-1330.
- [11] D.T. Pham, Joint Approximate Diagonalization of Positive Definite Hermitian Matrices, SIAM Journal on Matrix Analysis and Applications, 2000, V. 22 (4), p. 1136-1152.
- [12] R. Bro, PARAFAC, Tutorial and applications, Chemom. Intel. Lab. Syst., 1997, V.38, p.149-171, Elsevier.