



HAL
open science

Neural networks for array processing: from DOA estimation to blind separation of sources

Gilles Burel, N. Rondel

► **To cite this version:**

Gilles Burel, N. Rondel. Neural networks for array processing: from DOA estimation to blind separation of sources. IEEE Systems Man and Cybernetics Conference - SMC, Oct 1993, Le Touquet, France. pp.601-606, 10.1109/ICSMC.1993.384940 . hal-03223336

HAL Id: hal-03223336

<https://hal.science/hal-03223336>

Submitted on 17 Mar 2023

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.

Copyright

Neural Networks for Array Processing: From DOA estimation to Blind Separation of Sources

Gilles BUREL[†] & Nadine RONDEL^{†‡}

[†]Thomson CSF/LER, Av. Belle Fontaine, 35510 Cesson-Sévigné, France

[‡]SEFT, 18 rue du Dr Zamenhof, 92130 Issy-Les-Moulineaux, France

IEEE Int. Conf. on Systems, Man and Cybernetics, October 17-20th, 1993, Le Touquet, France

Abstract— In many signal processing applications, signals are received on an array of sensors, and the problem consists in estimating the Directions Of Arrival (DOA) of the signals, and/or in estimating the sources. Basically, the techniques proposed for its solution use either information about the geometry of the array, or information about the statistics of the sources.

Efficient neural-based approaches for both kinds of situations are proposed in this paper. When geometrical knowledge is available, the weights and structure of the neural networks are constrained according to the geometry of the array. When statistical information is available, neural networks which optimize a statistical criterion (namely the measure of dependence) are developed. Furthermore, neural networks provide the opportunity to fuse both approaches in a unified framework, and to take profit simultaneously of both kind of information.

Keywords— Array Processing, Blind Separation of Sources, Higher Order Moments, Neural Networks, Backpropagation.

I. INTRODUCTION

The localization or estimation of radiating sources by passive sensor arrays has received considerable attention in the last 30 years because it is one of the central problems in radar, sonar, radio-astronomy, and seismology [7]. Depending on the kind of a priori knowledge, two classes of approaches have been proposed: when the geometry of the array is known, a Maximum Likelihood Estimator (MLE) can be used [5]; when the sources are statistically independent, blind separation methods can be employed [1] [2], without need of geometrical knowledge (e.g. long under-

water antennas which are being deformed by streams).

In this paper, neural based approaches are developed. The formulation of the problem is recalled in section 2, and the principles of neural networks are summarized in section 3. In section 4, it is shown that use of geometrical information can be realized via constraints on the weights of the neural network. The neural approach provides a dramatic reduction of computation time for the same precision than MLE. In section 5, an unsupervised neural network which minimizes a statistical measure of dependence is proposed. This neural network is able to realize blind estimation of the sources, without any geometrical knowledge. Then, in section 6, it is shown that both approaches can be fused in a unified framework, and the interest of taking profit simultaneously of geometrical and statistical information is stressed. Finally, section 7 provides some experimental results.

II. PROBLEM FORMULATION

Consider an array composed of m sensors with arbitrary locations and arbitrary directional characteristics [7], and assume that n narrow-band sources centered around a known frequency ν_0 impinge on the array from locations $\theta_1, \dots, \theta_n$. Let us note:

- $\rho_i(\theta_k)$ the amplitude response of the i^{th} sensor to a wavefront impinging from location θ_k
- $\tau_i(\theta_k)$ the propagation delay between a reference point and the i^{th} sensor for a wavefront impinging from location θ_k
- $b_i(t)$ the complex envelope of the noise at the i^{th} sensor
- $x_k(t)$ the complex envelope of the signal emitted by the k^{th} source and received at the reference point
- $y_i(t)$ the complex envelope of the signal received at the i^{th} sensor

The signal emitted by the k^{th} source and received at the i^{th} sensor is:

$$Y_{ik}(t) = \text{Re} \left\{ x_k(t - \tau_i(\theta_k)) \rho_i(\theta_k) e^{j2\pi\nu_0(t - \tau_i(\theta_k))} \right\}$$

Because of the narrow-band assumption $x_k(t - \tau_i(\theta_k)) \simeq x_k(t)$, hence:

$$y_i(t) = \sum_{k=1}^n \rho_i(\theta_k) e^{-j2\pi\nu_0\tau_i(\theta_k)} x_k(t) + b_i(t)$$

Using matrix notation, we can write:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_1(\theta_1) & \dots & a_1(\theta_n) \\ \vdots & & \vdots \\ a_m(\theta_1) & \dots & a_m(\theta_n) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

or more compactly:

$$y = A(\Theta)x + b$$

The problem is to estimate the sources $x(1), \dots, x(N)$ and/or the directions of arrival Θ , on the basis of a set of snapshots $y(1), \dots, y(N)$. In the following, it will be assumed that the number of sources has been determined (see [6]).

III. PRINCIPLES OF NEURAL NETWORKS

Neural networks have gained popularity among the scientific community during the last decade because of their success as non-linear adaptative systems [3]. Many neural networks models can be described as a non-linear parametric function $s = G_w(e)$, where e is the input vector, s the output vector, and G_w a function parameterized by a vector w . The entries of w are the weights of the network.

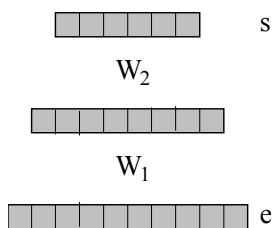


Figure 1: A 3-layer perceptron

Let us consider for instance the well known multilayer perceptron (MLP). The output vector of the 3-layer perceptron depicted on figure 1 is given by:

$$s = f [W_2 f (W_1 e + b_1) + b_2]$$

where f is a function, W_1 and W_2 are matrices, and b_1 and b_2 are bias vectors. The learning algorithm, known as "backpropagation" [4], updates the components of the matrices and of the bias vectors according to the gradient of the mean-square error $e_{MS} = E\{|s - s_{desired}|^2\}$.

IV. TAKING PROFIT OF GEOMETRICAL INFORMATION

The first class of techniques uses information about the geometry of the array. By way of illustration, let us consider the case of a uniform linear array (fig 2). Let us note c the celerity of the wave, and λ the wavelength. The array is composed of n sensors equispaced by $d = \lambda/2$, and the sensors have uniform directional response.

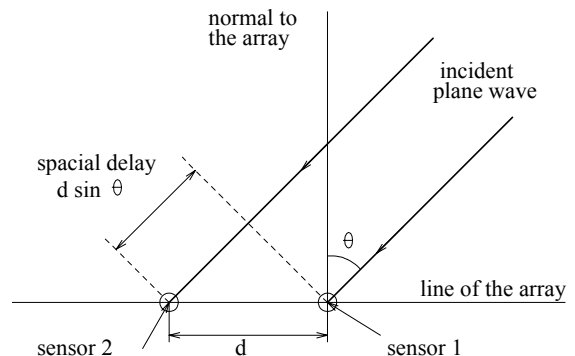


Figure 2: Uniform Linear Array

Taking the first sensor as the reference point, the delay is:

$$\tau_i(\theta_k) = \frac{(1-i)d \sin \theta_k}{c}$$

Hence, $a_i(\theta_k) = e^{j(i-1)\omega_k}$ with $\omega_k = \pi \sin \theta_k$

Usually, the noise is assumed to be a white gaussian noise with zero mean and correlation matrix $E\{b(t)b^H(t)\} = \sigma^2 I$, where H stands for the hermitian transpose, and I stands for the identity matrix. The noise is also assumed to be independent from the sources. The log-likelihood function of the observations is then:

$$L = \text{const} - mN \ln \hat{\sigma} - \frac{1}{2\hat{\sigma}^2} \sum_{t=1}^N \|y(t) - \hat{A}\hat{x}(t)\|^2$$

Maximizing the log-likelihood function is equivalent to minimizing

$$C_{MLE} = \sum_{t=1}^N \|y(t) - \hat{A}\hat{x}(t)\|^2$$

For a given \hat{A} , the best choice for $\hat{A}\hat{x}(t)$ is the orthogonal projection of $y(t)$ on the subspace spanned by the columns

of \hat{A} . Hence:

$$\hat{x}(t) = (\hat{A}^H \hat{A})^{-1} \hat{A}^H y(t)$$

The MLE of Θ is then given by the $\hat{\Theta}$ which minimizes

$$\sum_{t=1}^N \|(I - P_{\hat{A}(\hat{\Theta})})y(t)\|^2$$

where $P_{\hat{A}(\hat{\Theta})}$ is the projection operator. It should be pointed out that estimation of Θ can be performed only if $m \geq n + 1$ (otherwise, $P_{\hat{A}(\hat{\Theta})}$ would be always equal to identity).

Let us consider the MLP depicted on figure 3 (mlp2). It receives on input a snapshot $y(t)$, and it tries to minimize $e_{MS} = \sum_{t=1}^N \|y(t) - \hat{y}(t)\|^2$. The structure of the MLP is such that $\hat{x}(t) = By(t)$ and $\hat{y}(t) = \hat{A}\hat{x}(t)$

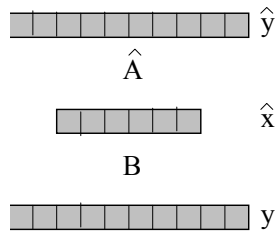


Figure 3: MLP constrained by geometrical information

Matrix \hat{A} is constrained by geometrical information:

$$\hat{A} = \begin{pmatrix} 1 & \dots & 1 \\ e^{j\hat{\omega}_1} & \dots & e^{j\hat{\omega}_n} \\ \vdots & & \vdots \\ e^{j(m-1)\hat{\omega}_1} & \dots & e^{j(m-1)\hat{\omega}_n} \end{pmatrix}$$

The result is an implicit constraint on matrix B , because, for any \hat{A} , the matrix B which minimizes e_{MS} is the pseudo-inverse $B = (\hat{A}^H \hat{A})^{-1} \hat{A}^H$. Hence, it becomes obvious that the neural network minimizes the same criterion as MLE.

The interest of the neural network in comparison with MLE is its speed. MLE requires the computation of C_{MLE} for each possible value of Θ (with some quantification step) in order to find the minimum. Since Θ is multi-dimensional, the number of possible values is huge. The neural network directly goes to the minimum via gradient descent. However, it should be stressed that the neural network might be trapped in a local minimum. To avoid

this problem, a rough estimation of DOA is computed before activation of this MLP.

The rough estimation is provided by another neural network, depicted on figure 4 (mlp1). It receives on input the normalized correlation matrix R^{nor} whose entries are

$$R_{ij}^{nor} = \frac{\text{Re}(R_{ij})}{\sum_{i=1}^m R_{ii}}$$

with $R = \frac{1}{N} \sum_{t=1}^N y(t)y^H(t)$. The objective of the normalization is invariance w.r.t. the energy of the sources. Providing only correlation information to this neural net can be justified by the fact that MLE itself uses only correlation information as shown in appendix A. In this network, the function f is the hyperbolic tangent. This rough estimation MLP must be trained previously on a large number of signals. The rough estimation is then used to initialize the weights of the constrained MLP.

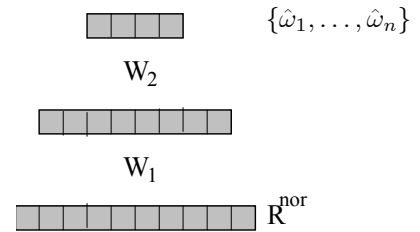


Figure 4: MLP for rough estimation of DOA

Computation time is low, because it only requires one propagation through the rough estimation MLP, and a few learning iterations on the fine estimation MLP. Since the fine estimation MLP starts from an initial state close to the solution, it is not trapped in a local minimum, and it reaches quickly the global minimum.

V. TAKING PROFIT OF STATISTICAL INFORMATION

Without information about the geometry of the array, it is impossible to estimate the directions of arrival. Anyway, the sources still can be estimated if statistical information is available. The methods based on statistical information are usually called "Blind Separation Methods". The information is generally the statistical independence of the sources. Assume that:

$$\forall x_1^R, x_1^I, \dots, x_n^R, x_n^I, \\ p(x_1^R, x_1^I, \dots, x_n^R, x_n^I) = p(x_1^R)p(x_1^I)\dots p(x_n^R)p(x_n^I)$$

where p is the density of probability, and z^R and z^I stand for the real and imaginary part of z .

A neural network which minimizes a measure of the statistical dependence of its outputs $e_D(s)$ has been proposed in [1]. The algorithm is based on the minimization of a quadratic form of high order moments. Assume that the number of sensors is equal to the number of sources (if it is not the case, one may perform a principal component analysis on the snapshots, and keep the n most significant directions). Let us consider a linear neural network ($s = Be$) whose input is $e = y(t)$ and whose output is $s = \hat{x}(t)$. Then, we have:

$$\hat{x}(t) = BAx(t) + Bb(t)$$

If the SNR is high enough, $\hat{x}(t) \simeq BAx(t)$, and a minimum of $e_D(\hat{x})$ is obtained when BA is the product of a diagonal matrix by a permutation matrix (because, in that case, the entries of \hat{x} are statistically independent).

It should be underlined that such a blind separation method works even if the number of sources is equal to the number of sensors, while the MLE requires more sensors than sources. But it does not work with gaussian sources, because in that case, it suffices that BA be pseudo-orthogonal (see appendix B for more details) to provide independence of the entries of \hat{x} . Hence \hat{x} is not an estimation of the sources any more. Another way to understand why blind separation does not work with gaussian variables is to notice that gaussian variables are totally defined by their moments up to order 2. Hence, the higher order moments employed in [1] become useless.

VI. FUSION

Let us go further by noticing that neural networks provide a framework to combine geometrical and statistical information. Consider the network of figure 3 again, and let us replace the mean square error e_{MS} by a mixed error:

$$e_{mixed}(\hat{y}, \hat{x}) = e_{MS}(\hat{y}) + \gamma e_D(\hat{x})$$

One may retort that e_{MS} is useless when $m = n$ because there is always a matrix ($B = \hat{A}^{-1}$) which provides a null mean square error for any \hat{A} . But in fact, the mean square error is still useful because, as stressed previously, it imposes an implicit constraint on matrix B .

The table below shows the conditions with respect to the information taken into account and to the nature of the sources (gaussian with similar variances or non-gaussian). The mixed approach is able to estimate the sources and the DOA for any kind of sources, even if the number of sensors is not greater than the number of sources.

cost function	$e_{MS}(\hat{y})$	$e_D(\hat{x})$	$e_{mixed}(\hat{y}, \hat{x})$
gaussian	$m > n$	does not work	$m \geq n$
non gaussian	$m > n$	$m \geq n$	$m \geq n$

Let us consider for instance the case of gaussian sources when $m = n$. Because the sources are gaussian, minimizing $e_D(\hat{x})$ does not provide enough equations to correctly determine the $2n^2$ real unknowns of matrix B (see appendix B). But, with the addition of $e_{MS}(\hat{y})$, B is implicitly constrained to be close to the pseudo-inverse of \hat{A} . Since \hat{A} contains only n unknowns, the system becomes over-determined.

VII. EXPERIMENTAL RESULTS

Experimental results obtained with two gaussian sources and a SNR of 20dB are presented in this section.

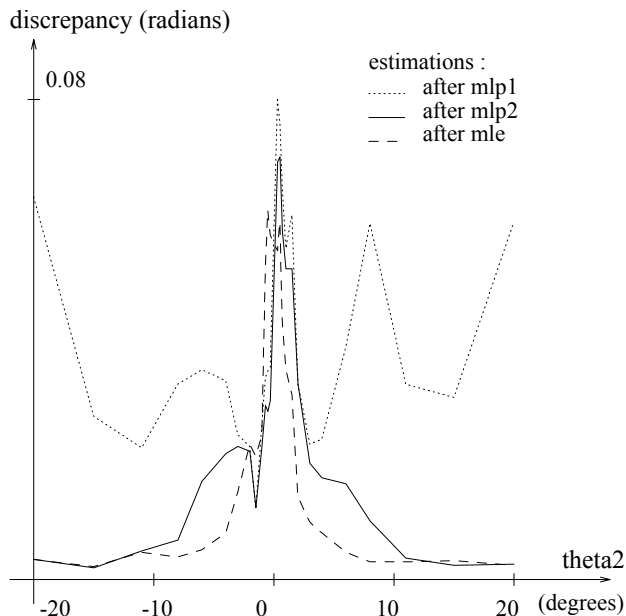


Figure 5: Estimation error with 2 sources and 10 sensors

Figure 5 shows the discrepancy $\sqrt{E\{(\hat{\omega}_2 - \omega_2)^2\}}$ as a function of θ_2 , when the number of sensors is $m = 10$ and the number of snapshots is $N = 20$. The other source is at $\theta_1 = 0^\circ$. As expected the MLE and the fine estimation MLP provide comparable results. Cases where MLE is better can be explained by the fact that the MLP has not performed enough iterations to reach exactly the minimum by gradient descent. When the MLP is better, it may be due to the limit imposed by the quantization of the MLE, or to the inversion of an almost singular matrix (near $\theta_2 = 0^\circ$) in the computation of C_{MLE} .

The rough estimation MLP comprises $m^2 = 100$ inputs, 42 hidden units, and 2 outputs. It has been previously

trained on 3071 examples (various combinations of ω_1 and ω_2). The fine estimation MLP comprises 10 complex inputs, 2 complex hidden units and 10 complex outputs. For MLE, the criterion C_{MLE} is computed for each combination of angles ($-60^\circ \leq \theta_1 \leq 60^\circ$ and $-60^\circ \leq \theta_2 \leq 60^\circ$) with steps of 0.0047rad on ω_1 and ω_2 . Hence C_{MLE} must be evaluated around 1.3×10^6 times for each set of 20 snapshots. On a Sun workstation, the MLE requires 11 hours for each set of 20 snapshots, while the neural networks require only 1.5 second. Figure 6 shows the value of e_{MS} for a particular set of snapshots. The true DOA are $\theta_1 = 0^\circ$ and $\theta_2 = 20^\circ$. As mentioned in section 3, there is no spurious local minimum near the solution.

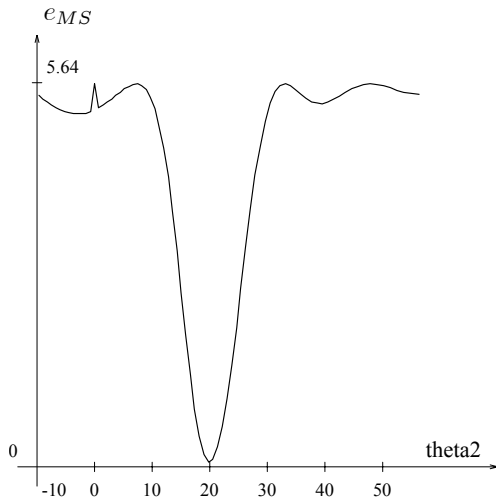


Figure 6: value of the mean square error as a function of the estimated angle, for a particular set of snapshots

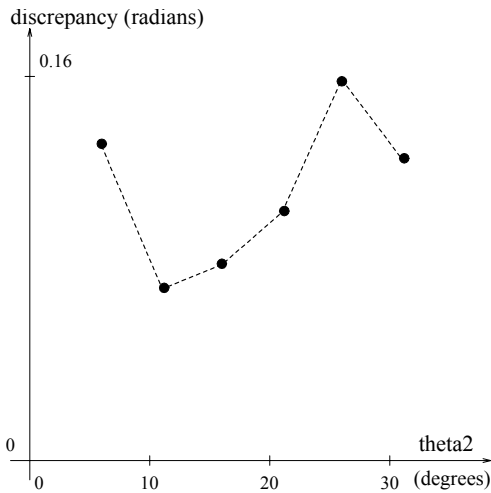


Figure 7: Estimation error with 2 sources and 2 sensors

Figure 7 shows $\sqrt{E\{(\hat{\omega}_2 - \omega_2)^2\}}$ when the number of sensors is $m = 2$ and the number of snapshots is $N = 200$. Since the number of sources is equal to the number of sensors, classical methods such as MLE do not work. Here, a MLP using e_{mixed} (see section 6) has been used. Figure 8 shows the value of e_{mixed} for a particular set of snapshots. It can be seen from this figure that there is no spurious local minimum in the neighbourhood of the solution. The true DOA are $\theta_1 = 0^\circ$ and $\theta_2 = 20^\circ$. As expected, the minimum is close to the true value of θ_2 .

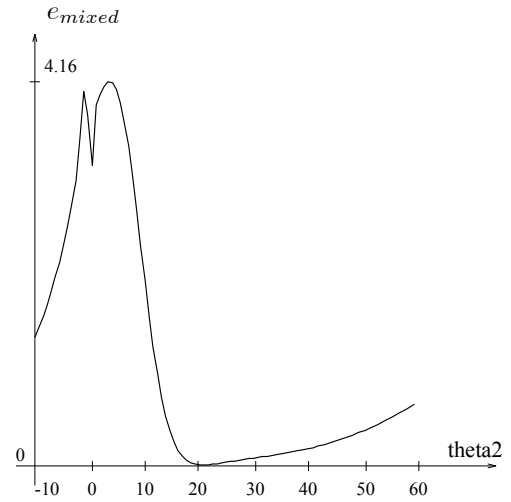


Figure 8: value of the mixed error as a function of the estimated angle, for a particular set of snapshots

VIII. CONCLUSION

Neural approaches for DOA Estimation and Blind Separation of Sources have been developed in this paper. Furthermore, neural networks allow to combine in an efficient way geometrical and statistical a priori information. The result is the possibility to estimate the sources and the DOA, even with gaussian sources and with as many sources as sensors. It has also been shown that (when the MLE works) the neural approach provides almost the same precision than MLE with less computation time. Taking into account that some potentialities of this approach still remain unexploited, we are currently studying the possibility to estimate more directions of arrival than sensors.

REFERENCES

- [1] G. Burel, "Blind Separation of Sources: a Nonlinear Neural Algorithm", Neural Networks, vol. 5, pp. 937-947, 1992
- [2] C. Jutten, J. Héroult, "Blind Separation of Sources, part I: an adaptive algorithm", Signal Processing, vol. 24, n^o1, July 1991

- [3] R.P. Lippmann, "Pattern Classification using Neural Networks", IEEE Communication Magazine, November 1989
- [4] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning internal representations by error backpropagation", Parallel Distributed Processing, D.E. Rumelhart and J.L. McClelland, Chap8, Bradford book - MIT Press - 1986
- [5] P. Stoica, A. Nehorai, "MUSIC, Maximum Likelihood, and Cramer-Rao Bound", IEEE ASSP, vol 37, n°5, May 1989
- [6] M. Wax, T. Kailath, "Detection of signals by information theoretic criteria", IEEE ASSP, vol 33, n°2, pp 387-392, April 1985
- [7] I. Ziskind, A. Wax, "Maximum Likelihood Localization of Multiple Sources by Alternating Projection", IEEE ASSP, vol 36, n°10, October 1988

APPENDIX A:

INTEREST OF THE CORRELATION MATRIX

It is proved below that the MLE uses only information contained in the correlation matrix:

$$\begin{aligned} C_{MLE} &= \sum_{t=1}^N \|(I - P_{\hat{A}})y_t\|^2 \\ &= \sum_{t=1}^N \text{tr} \left\{ (I - P_{\hat{A}})y_t y_t^H (I - P_{\hat{A}})^H \right\} \\ &= \text{tr} \left\{ (I - P_{\hat{A}}) \left[\sum_{t=1}^N y_t y_t^H \right] (I - P_{\hat{A}})^H \right\} \end{aligned}$$

APPENDIX B:

PARTICULARITY OF GAUSSIAN SOURCES

Let us consider a blind separation of sources problem, and let us note x the sources, A the mixture, B the separator, and \hat{x} the output of the separator. The vectors are of dimension n , and the matrices are $n \times n$. By noting $C = BA$, we have:

$$\hat{x} = Cx$$

The ideal would be to obtain a matrix C equal to identity. In fact, we can expect only $C = \Gamma P$, where Γ is a diagonal matrix, and P a permutation matrix. This is not a problem because the sources are estimated modulo a permutation and a dilatation. But, with gaussian sources, we show below that C can be any pseudo-orthogonal matrix, hence the separation may fail. Let us consider the characteristic function:

$$\begin{aligned} \Psi_{\hat{x}}(v) &= \ln E\{e^{jv^H \hat{x}}\} \\ &= \ln E\{e^{jv^H Cx}\} \\ &= \ln E\{e^{j(C^H v)^H x}\} \end{aligned}$$

$$\begin{aligned} &= \Psi_x(C^H v) \\ &= \sum_{j=1}^n \Psi_{x_j}(C_j^H v) \end{aligned}$$

where C_j is the j^{th} column of C . The last equation is due to the independence of the sources. After convergence of the separator, the entries of \hat{x} are independent, hence we can write:

$$\begin{aligned} \Psi_{\hat{x}}(v) &= \sum_{i=1}^n \Psi_{\hat{x}}(\dots, 0, v_i, 0, \dots) \\ &= \sum_{i=1}^n \sum_{j=1}^n \Psi_{x_j}(C_{ij}^* v_i) \end{aligned}$$

Thus, we obtain:

$$\sum_{j=1}^n \sum_{i=1}^n \Psi_{x_j}(C_{ij}^* v_i) = \sum_{j=1}^n \Psi_{x_j} \left(\sum_{i=1}^n C_{ij}^* v_i \right)$$

And using Taylor expansion:

$$\sum_{j=1}^n \sum_{i=1}^n \sum_{k=1}^{\infty} b_{jk} (C_{ij}^* v_i)^k = \sum_{j=1}^n \sum_{k=1}^{\infty} b_{jk} \left(\sum_{i=1}^n C_{ij}^* v_i \right)^k$$

where b_{jk} is proportional to the cumulant of order k of x_j . Since the equation above must be true for any v , we can equal the terms of the same order:

$$\forall k, \quad \sum_{j=1}^n \sum_{i=1}^n b_{jk} (C_{ij}^* v_i)^k = \sum_{j=1}^n b_{jk} \left(\sum_{i=1}^n C_{ij}^* v_i \right)^k$$

Then, by developing the right hand side term, we obtain:

$$\forall k, \quad \sum_{j=1}^n b_{jk} \sum_{\substack{i_1, \dots, i_k \\ (\text{not all equal})}} C_{i_1 j}^* \dots C_{i_k j}^* v_{i_1} \dots v_{i_k} = 0$$

Since the equation above must be true for any v , all the coefficients of the polynomial must be null:

$$\forall k, \quad \forall (i_1, \dots, i_k) (\text{not all equal}), \quad \sum_{j=1}^n b_{jk}^* C_{i_1 j}^* \dots C_{i_k j}^* = 0$$

One can easily verify that any matrix $C = \Gamma P$ is a solution. For gaussian sources, the b_{jk} are null for $k > 2$, hence the equations system reduces to:

$$\forall (i_1, i_2) (\text{not equal}), \quad \sum_{j=1}^n b_{j2}^* C_{i_1 j}^* C_{i_2 j}^* = 0$$

This is some kind of pseudo-orthogonality condition, hence, any pseudo-orthogonal matrix C is a solution, and, consequently, separation of sources may fail. On the contrary, non-gaussian sources have at least one cumulant of order greater than 2 which is non zero. Hence, the number of equations becomes large enough to constraint C as wanted.