

Blind compensation of polynomial mixtures of Gaussian signals with application in nonlinear blind source separation

Bahram Ehsandoust, Bertrand Rivet, Massoud Babaie-Zadeh, Christian Jutten

▶ **To cite this version:**

Bahram Ehsandoust, Bertrand Rivet, Massoud Babaie-Zadeh, Christian Jutten. Blind compensation of polynomial mixtures of Gaussian signals with application in nonlinear blind source separation. ICASSP 2017 - IEEE International Conference on Acoustics, Speech and Signal Processing, Mar 2017, New Orleans, United States. pp.4681 - 4685, 10.1109/ICASSP.2017.7953044 . hal-01552278

HAL Id: hal-01552278
https://hal.science/hal-01552278

Submitted on 1 Jul 2017

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.

BLIND COMPENSATION OF POLYNOMIAL MIXTURES OF GAUSSIAN SIGNALS WITH APPLICATION IN NONLINEAR BLIND SOURCE SEPARATION

Bahram Ehsandoust[⋆†] *Bertrand Rivet*[⋆] *Massoud Babaie-Zadeh*[†] *Christian Jutten*[⋆]

[⋆] University of Grenoble Alpes, CNRS, GIPSA-lab, Grenoble, France
{bahram.ehsandoust, bertrand.rivet, christian.jutten}@gipsa-lab.grenoble-inp.fr
[†] Sharif University of Technology, Electrical Engineering Department, Tehran, Iran
ehsandoust@ee.sharif.edu, mbzadeh@yahoo.com

ABSTRACT

In this paper, a proof is provided to show that Gaussian signals will lose their Gaussianity if they are passed through a polynomial of an order greater than 1. This can help in blind compensation of polynomial nonlinearities on Gaussian sources by forcing the output to follow a Gaussian distribution (the term "blind" refers to lack of any prior information about the nonlinear function). It may have many applications in different fields of nonlinear signal processing for removing the nonlinearity. Particularly, in nonlinear blind source separation, it can be used as a pre-processing step to transform the problem to a linear one, which is already well studied in the literature. This idea is proposed, proved, and finally verified by a simple simulation as a proof of concept in this paper.

Index Terms— Gaussian Processes, Nonlinear Distortion, Polynomial Mappings, Blind Source Separation, Nonlinear Signals Processing

1. INTRODUCTION

In signal processing applications, it is usual to have a number of signals measured by some sensors, while each of them might be a mixture of a number of source signals. Even though this problem is relatively easy to solve when the mixture is linear (more specifically, the system is LTI), it becomes mostly too difficult for nonlinear functions. Thus it is always wanted to transform the nonlinear system to a linear one in order that it can be processes by already established signal processing methods for linear mixtures.

On the other hand, in many application of signal processing, signals are modeled as stochastic processes. In this sense, Gaussian random variables and Gaussian Processes (GPs) are of the most interesting models because of their simplicity, generality and nice characteristics. GPs [1] can be used to track nonlinear communication channel or probabilistic channel equalization [2], to classification [1] or to perform linear source separation [3, 4].

This work is partly funded by the European project 2012-ERC-Adv-320684 CHESS and the Center for International Scientific Studies and Collaboration (CISSC) of Iran.

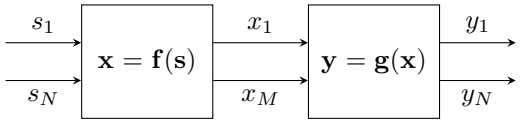


Fig. 1. BSS problem basic model

In this work, the goal is to blindly transform a nonlinear polynomial system to a linear one under the assumption that the sources are normally distributed. As a result of this work, problems in any domain of signals processing which satisfy the mentioned assumptions, can be pre-processes in order to be transformed to linear ones and then processes linearly.

Application to Nonlinear Blind Source Separation (BSS): In BSS, we observe a number of signals each of which is a mixture of some unknown sources (see Fig. 1) as

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) \tag{1}$$

where $\mathbf{x}(t)$ and $\mathbf{s}(t)$ are the $M \times 1$ observation and the $N \times 1$ source vectors respectively. The goal is to blindly estimate (separate) the sources, i.e. finding a separating function $\mathbf{g}$ such that $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) = \mathbf{g} \circ \mathbf{f}(\mathbf{s}(t)) = \hat{\mathbf{s}}(t)$ is as close as possible to the source vector $\mathbf{s}(t)$, up to classical indeterminacies (order recovery, nonlinear distortions).

While this problem has been well studied for linear mixtures and many algorithms have been already proposed in this regard [5, 6, 7, 8, 9, 10], there is not much progress in general nonlinear cases. Consequently, except a few works on nonlinear BSS in general case [11, 12], studies on this issue were focused on specific applications with restricted mixing models [13, 14, 15, 16, 17, 18]. This is why it is thought that in general, roughly speaking, linear mixtures are separable while nonlinear ones are not.

In this sense, the result of this work is important and remarkable in the BSS domain. In this paper, we will show that when $\mathbf{f}(\cdot)$ is an (unknown) invertible polynomial nonlinear

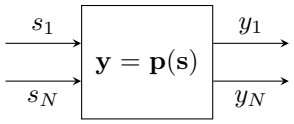


Fig. 2. Polynomial mixture

mapping of the sources, the Gaussianity property is sufficient for inverting the nonlinear mapping and reducing the problem to a source separation problem with linear mixtures.

It is worth noting that there are numerous practical applications of BSS where the sources are modeled by GPs, e.g. [19, 3, 4, 20]. In these applications, utilizing the proposed method leads to a set of signals which are linear mixtures of mutually independent sources. In order to reconstruct the sources after linearizing the mixture, one can use either conventional linear BSS techniques which use the temporal correlation of the sources (like SOBI [9]) or recent specific methods for GPs (like [21]).

The rest of the paper is organized as follows. The theoretical basis including the main proposed theorem comes in Section 2. In Section 3 a simple algorithm is proposed based on the theorem, which is supported by simulations in Section 4. Finally, some future works are suggested in Section 5.

2. THEORETICAL BASIS

As follows, it is proved that polynomials distort the Gaussianity characteristic of their inputs, except linear ones.

Definition An N-dimensional mapping $\mathbf{p} : \mathbb{R}^N \to \mathbb{R}^N$, $\mathbf{p}(\mathbf{s}) = (p_1(\mathbf{s}), \ldots, p_N(\mathbf{s}))^T$ (where $\mathbf{s}$ is an $N \times 1$ vector of variables and $\cdot^T$ denotes matrix transpose) is called an N-dimensional *polynomial mapping* if each p_i is a polynomial (of order O_i of N variables $s_1, \ldots, s_N$).

Theorem 1. *Let N sources $s_1, \cdots, s_N$ be jointly normally distributed and mixed via an invertible polynomial mapping $\mathbf{p} : \mathbb{R}^N \to \mathbb{R}^N$ providing N outputs $y_1, \cdots, y_N$ (Fig. 2).*

If the outputs $y_1, \ldots, y_N$ also follow a joint Gaussian distribution, the polynomial $\mathbf{p}$ is limited to be linear as $\mathbf{y} = \mathbf{p}(\mathbf{s}) = \mathbf{A}\mathbf{s} + \mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are an $N \times N$ matrix and an $N \times 1$ vector of constant numbers respectively.

In other words, the theorem 1 says that the only polynomial which preserves the Gaussianity is the linear one. It is worth noting that the reverse is a well known result: a linear mixture of Gaussian processes (random variables) leads to Gaussian processes (random variables).

Proof. Let us assume $(s_1, \ldots, s_N)$ has a mean vector $\boldsymbol{\mu}_\mathbf{s} = \mathbf{E}[\mathbf{s}]$ and a covariance matrix $\mathbf{K}_\mathbf{s} = \mathbf{E}[(\mathbf{s} - \boldsymbol{\mu}_\mathbf{s})(\mathbf{s} - \boldsymbol{\mu}_\mathbf{s})^T]$ (where $\mathbf{E}$ represents the expected value). Similarly, the output vector $(y_1, \ldots, y_N)$ has a mean vector $\boldsymbol{\mu}_\mathbf{y} = \mathbf{E}[\mathbf{y}]$ and a covariance matrix $\mathbf{K}_\mathbf{y} = \mathbf{E}[(\mathbf{y} - \boldsymbol{\mu}_\mathbf{y})(\mathbf{y} - \boldsymbol{\mu}_\mathbf{y})^T]$. Considering the Gaussianity, the probability density function (pdf) of the vectors $\mathbf{s}$ and $\mathbf{y}$, denoted by $\rho_\mathbf{S}(\mathbf{s})$ and $\rho_\mathbf{Y}(\mathbf{y})$ respectively, can be expressed as

$$\rho_\mathbf{S}(\mathbf{s}) = \frac{1}{\sqrt{(2\pi)^N |\mathbf{K}_\mathbf{s}|}} \, e^{\frac{-1}{2}(\mathbf{s}-\boldsymbol{\mu}_\mathbf{s})^T \mathbf{K}_\mathbf{s}^{-1} (\mathbf{s}-\boldsymbol{\mu}_\mathbf{s})} \quad (2)$$

$$\rho_\mathbf{Y}(\mathbf{y}) = \frac{1}{\sqrt{(2\pi)^N |\mathbf{K}_\mathbf{y}|}} \, e^{\frac{-1}{2}(\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})^T \mathbf{K}_\mathbf{y}^{-1} (\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})} \quad (3)$$

where $|\mathbf{K}_\mathbf{s}|$ and $|\mathbf{K}_\mathbf{y}|$ are the determinants of $\mathbf{K}_\mathbf{s}$ and $\mathbf{K}_\mathbf{y}$ respectively. On the other hand, we know that when $\mathbf{y} = \mathbf{p}(\mathbf{s})$ and $\mathbf{p}$ is an invertible function, the pdf of $\mathbf{y}$ follows

$$\rho_\mathbf{Y}(\mathbf{y}) = \frac{\rho_\mathbf{S}(\mathbf{s})}{|\mathbf{J}_\mathbf{p}|} \quad (4)$$

where $|\mathbf{J}_\mathbf{p}|$ is the determinant of the Jacobian of $\mathbf{p}$.

By definition, it is easy to see that all elements of the Jacobian of a polynomial mapping are polynomials. As a consequence, the determinant of the Jacobian is the absolute value of a polynomial. Let $q(\mathbf{s})$ be a polynomial such that

$$\det(\mathbf{J}_\mathbf{p}) = |q(\mathbf{s})|. \quad (5)$$

Thus (4) can be rewritten as

$$|q(\mathbf{s})| \times p_\mathbf{Y}(\mathbf{y}) = p_\mathbf{S}(\mathbf{s}) \quad (6)$$

$$\Rightarrow \ln|q(\mathbf{s})| + \ln p_\mathbf{Y}(\mathbf{y}) = \ln p_\mathbf{S}(\mathbf{s}). \quad (7)$$

Using (2) and (3) to explicit (7) leads to

$$\ln|q(\mathbf{s})| - \frac{1}{2}\ln|\mathbf{K}_\mathbf{y}| - \frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})^T \mathbf{K}_\mathbf{y}^{-1}(\mathbf{y}-\boldsymbol{\mu}_\mathbf{y}) = - \frac{1}{2}\ln|\mathbf{K}_\mathbf{s}| - \frac{1}{2}(\mathbf{s}-\boldsymbol{\mu}_\mathbf{s})^T \mathbf{K}_\mathbf{s}^{-1}(\mathbf{s}-\boldsymbol{\mu}_\mathbf{s}), \quad (8)$$

that is

$$c + (\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})^T \mathbf{K}_\mathbf{y}^{-1}(\mathbf{y}-\boldsymbol{\mu}_\mathbf{y}) = (\mathbf{s}-\boldsymbol{\mu}_\mathbf{s})^T \mathbf{K}_\mathbf{s}^{-1}(\mathbf{s}-\boldsymbol{\mu}_\mathbf{s}) + 2\ln|q(\mathbf{s})| \quad (9)$$

where $c = \ln(|\mathbf{K}_\mathbf{y}|/|\mathbf{K}_\mathbf{s}|) = \ln|\mathbf{K}_\mathbf{y}\mathbf{K}_\mathbf{s}^{-1}|$ is a constant independent of $\mathbf{s}$ and $\mathbf{y}$. Now it should be proved that since (9) holds for all $\mathbf{s} \in \mathbb{R}^N$, $\mathbf{y}$ must be a linear function of $\mathbf{s}$.

In particular, (9) holds for any vector $\mathbf{s}$ which lies on the line where all entries of the vector take the same value, i.e. $\mathbf{s} = (s, \ldots, s)^T_{N\times 1} = s\mathbf{1}_{N\times 1}$. In this case, $\mathbf{y} = \mathbf{p}(\mathbf{s}) = \tilde{\mathbf{p}}(s)$ and $q(\mathbf{s})$ become single variable polynomials of s as follows

$$\mathbf{s} = s\mathbf{1}_{N\times 1} \Rightarrow \mathbf{y} = \tilde{\mathbf{p}}(s) = [\,\tilde{p}_1(s), \ldots, \tilde{p}_N(s)\,]^T \quad (10)$$

and $q(\mathbf{s}) = \tilde{q}(s)$ where

$$\forall 1 \le k \le N \quad \tilde{p}_k(s) = p_k(\mathbf{s})\big|_{\mathbf{s}=s\mathbf{1}_{N\times 1}} = \sum_{j=0}^{d_k} a_{kj} s^j \quad (11)$$

$$\tilde{q}(s) = q(\mathbf{s})\big|_{\mathbf{s}=s\mathbf{1}_{N\times 1}} = \sum_{i=0}^{d_\mathbf{J}} b_i s^i. \quad (12)$$

Replacing (11) and (12) in (9) results in

$$c + (\tilde{\mathbf{p}}(s) - \boldsymbol{\mu}_{\mathbf{y}})^T \mathbf{K}_{\mathbf{y}}^{-1} (\tilde{\mathbf{p}}(s) - \boldsymbol{\mu}_{\mathbf{y}}) = (s\mathbf{1}_{N\times 1} - \boldsymbol{\mu}_{\mathbf{s}})^T \mathbf{K}_{\mathbf{s}}^{-1} (s\mathbf{1}_{N\times 1} - \boldsymbol{\mu}_{\mathbf{s}}) + 2\ln|\tilde{q}(s)| = \alpha s^2 + \beta s + \gamma + 2\ln|\tilde{q}(s)| \quad (13)$$

where $\alpha = \mathbf{1}_{N\times 1}^T \mathbf{K}_{\mathbf{s}}^{-1} \mathbf{1}_{N\times 1}$, $\beta = -2\mathbf{1}_{N\times 1}^T \mathbf{K}_{\mathbf{s}}^{-1} \boldsymbol{\mu}_{\mathbf{s}}$ and $\gamma = \boldsymbol{\mu}_{\mathbf{s}}^T \mathbf{K}_{\mathbf{s}}^{-1} \boldsymbol{\mu}_{\mathbf{s}}$ are constant scalars.

Particularly, it is interesting to study the equality (13) when s tends to infinity. From (11) it can be seen that for large s, the right side behaves as αs^2, so the left side should also behave as a second order polynomial. In other words, all monomials in $\tilde{\mathbf{p}}(s)$, hence $\mathbf{p}(\mathbf{s})$, has a degree at most 1 and $\mathbf{p}(\mathbf{s})$ is limited to be linear. □

Corollary 1. *In the model of Fig. 1, assuming* $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ *to be an invertible polynomial, and given Gaussian Processes as the sources, if we find a polynomial mapping* $\mathbf{g}(\mathbf{x})$ *such that the outputs* $y_1(t), y_2(t), \ldots, y_N(t)$ *are Gaussian Processes, the whole function* $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ *will be a linear mixture, i.e.* $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) = \mathbf{h}(\mathbf{s}(t)) = \mathbf{A}\mathbf{s}(t)$.

It should be noted that although GPs are special cases of the main theorem, they are very useful and flexible in modeling many practical signals (as introduced in Section 1).

3. PROPOSED ALGORITHM

Although, based on Section 2, theorem 1 holds for any invertible polynomial mapping, it is interesting to study it particularly when its inverse is also a polynomial. In this case, it is necessary and sufficient for linearizaing the mixture to estimate a polynomial $\mathbf{g}$ such that $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t))$ is a vector with Gaussian distribution (because in this case, the combination $\mathbf{h} = \mathbf{g} \circ \mathbf{f}$ will be a polynomial and will satisfy the assumption of the theorem 1). Consequently, one can propose an algorithm which takes a cost function of "non-Gaussianity" and minimizes it with respect to the polynomial $\mathbf{g}$.

Here we assume a parametric model for the polynomial and then the optimization is done with respect to the parameters of our model. The parametric model of an L^{th} order polynomial of N signals is chosen as

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_N(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta_1}^T \\ \boldsymbol{\theta_2}^T \\ \vdots \\ \boldsymbol{\theta_N}^T \end{bmatrix} \mathbf{k}(\mathbf{x}) = \boldsymbol{\Theta}_{N\times P} \mathbf{k}_{P\times 1}(\mathbf{x}) \quad (14)$$

where $\boldsymbol{\theta_i}$ for $i = 1, \ldots, N$ is a column vector of the parameters (constant scalars), $\mathbf{k}(\mathbf{x})$ is the column vector containing all monomials with degree less than or equal to L and $P = \binom{N+L}{L} = \frac{(N+L)!}{N!L!}$ is the number of the parameters of each entry $g_i(\cdot)$ which is equal to the number of monomials with degree at most L.

For any $1 \le i \le N$, the entropy of y_i is defined as

$$\mathrm{H}(y_i) = -\mathbf{E}\{\ln \rho_{Y_i}(y_i)\} \quad (15)$$

where $\rho_{Y_i}(y_i)$ is the pdf of the i^{th} output signal y_i. Consequently, the neg-entropy [22] is calculated as

$$\boldsymbol{\mathcal{J}}(\mathbf{y}) = \mathbf{H}(\tilde{\mathbf{y}}) - \mathbf{H}(\mathbf{y}) \quad (16)$$

where $\tilde{\mathbf{y}}$ is a Gaussian random variable with the same covariance matrix as $\mathbf{y}$'s.

It can be easily shown that among all distributions with a given mean and variance, Gaussian pdf is the one with the highest entropy. Thus, neg-entropy is always non-negative and invariant by any linear invertible transformation, and vanishes iff the signal is Gaussian. Therefore, as well as some previous works on BSS (e.g. [23, 10]), we also use neg-entropy as such a measure of Gaussianity. It should be emphasized that in this work, neg-entropy is the cost function that is *minimized*, because we need to recover the Gaussianity of the sources. While in classical BSS methods, it is maximized in order to retrieve non-Gaussianity.

Thus the algorithm should optimize

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \ \|\boldsymbol{\mathcal{J}}(\boldsymbol{\Theta}\mathbf{k}(\mathbf{x}))\|^2, \quad (17)$$

considering the fact that each entry of $\boldsymbol{\mathcal{J}}(\boldsymbol{\Theta}\mathbf{k}(\mathbf{x}))$ depends only on one row of $\boldsymbol{\Theta}$, hence minimizing all the entries of $\boldsymbol{\mathcal{J}}(\boldsymbol{\Theta}\mathbf{k}(\mathbf{x}))$ will be equivalent to minimize its norm.

Since this cost function is not convex or even close to convex at all and it has too many local minimas, the minimization should done by a probabilistic method. For example, comparing the results of several runs of simulated annealing [24] algorithm with different random initialization is suggested in order to make sure that the global minimum is found.

4. SIMULATION RESULTS

The main theorem proposed in this work is supported by a simple 2-by-2 simulated example as follows. The two sources s_1 and s_2 are randomly chosen as $\mathcal{N}(0,1)$ and are mixed through a 2-dimensional polynomial mapping as

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \to \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} s_1 + (s_1 + s_2)^3 \\ s_2 - (s_1 + s_2)^3 \end{bmatrix} \quad (18)$$

to make the two observations.

From the scatter plot of the sources and the observations, which is depicted in Fig. 3, it is obvious that the observations do not follow a Gaussian distribution.

In cases when the order of the inverse polynomial is not known, one can start from a linear polynomial, and gradually increase the order until getting a low enough cost function. In this experiment, given a cubic model, we are looking for the parameters $\boldsymbol{\theta_1}^T = [\theta_{10}, \ldots, \theta_{19}]$ in

$$y_1 = \theta_{10}x_1^3 + \theta_{11}x_1^2x_2 + \theta_{12}x_1^2 + \theta_{13}x_1x_2^2 + \theta_{14}x_1x_2 + \theta_{15}x_1 + \theta_{16}x_2^3 + \theta_{17}x_2^2 + \theta_{18}x_2 + \theta_{19} \quad (19)$$

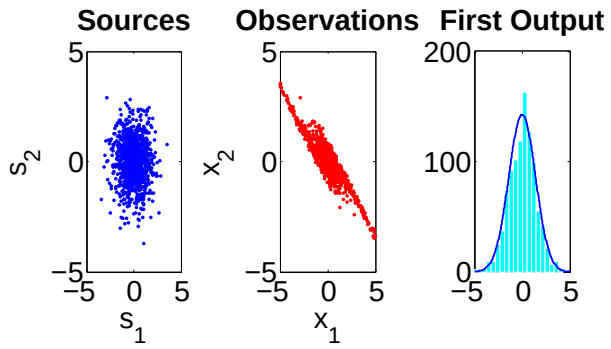


Fig. 3. The scatter plot of the sources and the observations of (18) for 1000 samples and how the histogram of the first output fits a Gaussian function. The neg-entropy for s_1, s_2, x_1, x_2 and y_1 are calculated 0.0524, 0.0476, 0.8664, 1.1073 and 0.0535 respectively.

such that y_1 follows a Gaussian distribution. To this end, as proposed in the previous section, the neg-entropy (16) of y_1 should be minimized with respect to the parameters $\boldsymbol{\theta_1}$ which leads to a linear mixture of s_1 and s_2.

Simulations show that the 10-dimensional minimization of $\boldsymbol{\theta_1}$ is quite difficult mainly because of 1) too many local minimas and non-convexity and 2) the high dimension of the space and the computational cost of the minimization. However, it can be seen that any linear mixture of s_1 and s_2 is a global minimum of the neg-entropy.

Particularly, it is interesting to see the behavior of the cost function (16) around $\boldsymbol{\theta_1} = [0,0,0,0,0,1,0,0,1,0]$, where $y_1 = x_1 + x_2 = s_1 + s_2$ is expected to be a global minimum.

Fig. 4 illustrates the partial variation of the neg-entropy with respect to any of entries of $\boldsymbol{\theta_1}$ around its optimal value $[0,0,0,0,0,1,0,0,1,0]$. It is evident that the neg-entropy tends to zero (global minimum) for the optimal value, and taking distance makes it increase rapidly. It should also be noted that changing θ_{19} does not affect the linearity of the mixture, hence, does not change the neg-entropy.

As it can be seen from Fig. 5, although the global minimum is in the origin, there are too many other local minimums that may trap the minimizing algorithm. This figure also shows that the value of the neg-entropy is minimized with respect to the coefficients of x_1 and x_2 (while not changing the other parameters) as long as we stay on the line $\theta_{15} = \theta_{18}$ where the two coefficients are equal. This can also be mathematically seen that at any point of the line $\theta_{15} = \theta_{18}$, y_1 is a linear mixture of s_1 and s_2, hence follows a Gaussian pdf.

5. DISCUSSION AND FUTURE WORKS

In this work, it is proved that the only invertible polynomial which preserves the Gaussianity is a linear function. This result can be used in many different applications dealing with polynomial nonlinearities. As an example, in nonlinear BSS

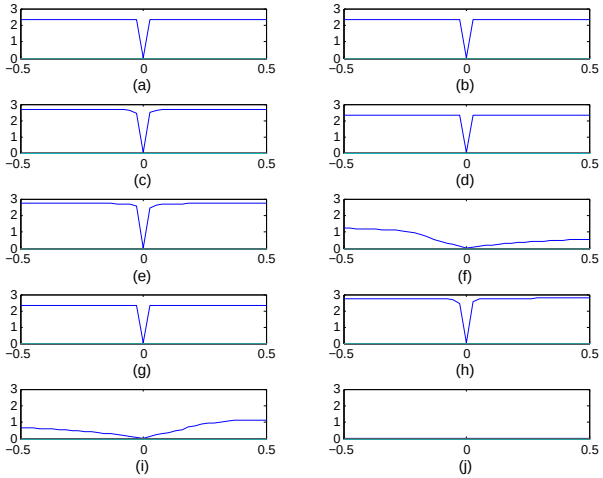


Fig. 4. The neg-entropy of y_1 in (19) with respect to the entries of $\boldsymbol{\theta_1}$ centered around their optimal value $[0,0,0,0,0,1,0,0,1,0]$ (from θ_{10} to θ_{19} in figures (a) to (j) respectively). Plotting with respect to each entry, the other parameters are kept constant.

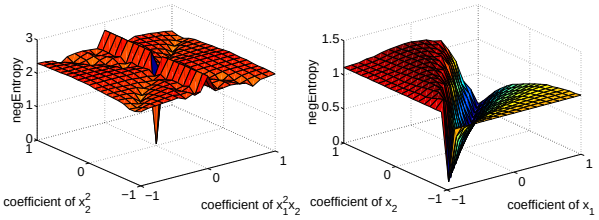


Fig. 5. The value of the neg-entropy of y_1 in (19) with respect to the coefficients of x_1 and x_2 while the other parameters are kept constant and equal to their optimal value in $[0,0,0,0,0,1,0,0,1,0]$.

problem, one may propose a two-step separating scheme, where at the first step the mixture is transformed to linear based on the result of this work, and the second step is a linear BSS method which can separate Gaussian signals based on non-stationnarity [25] or correlation [9]. However, this is a preliminary result and can be extended and generalized in both theoretic and algorithmic aspects.

In practice, nonlinear mixture are not exactly polynomials, but they may be approximated by polynomials. Thus it is interesting to see how a similar result can be achieved in those cases when the equations are not exact.

As mentioned earlier, the minimization of the neg-entropy is a too difficult task to do because of the local minimas and the computational cost. Thus, trying to find a mathematical expression of the minimizer or providing a minimization technique capable of escaping from local minimas is a significant subject for future. Moreover, instead of minimizing the neg-entropy, one may suggest to take an approximation of it as a cost function to be minimized (similar to [10]).

6. REFERENCES

[1] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[2] F. Pérez-Cruz, S. Van Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaria, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *Signal Processing Magazine*, vol. 30, no. 4, pp. 40–50, 2013.

[3] B. Rivet, M. Niknazar, and C. Jutten, "Nonparametric modelling of ecg: applications to denoising and to single sensor fetal ecg extraction," in *Int. Conf. on Latent Variable Analysis and Signal Separation*. Springer, 2012, pp. 470–477.

[4] S. Noorzadeh, B. Rivet, and P. Guméry, "An application of gaussian processes on ocular artifact removal from eeg," in *37th Int. Conf. of EMBC*. IEEE, 2015, pp. 554–557.

[5] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[6] T. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comutation*, vol. 7, no. 6, pp. 1004–1034, 1995.

[7] J.-F. Cardoso and A. Souloumiac, "An efficient technique for blind separation of complex sources," in *Proc. IEEE Signal Processing Workshop on Higher-Order Statistics*, USA (CA), June 1993, pp. 275–279.

[8] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. on SP*, vol. 44, no. 12, pp. 3017–3030, December 1996.

[9] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE transaction on Signal Processing*, vol. 45, no. 2, pp. 434–444, February 1997.

[10] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[11] D. N. Levin, "Performing nonlinear blind source separation with signal invariants," *Signal Processing, IEEE Transactions on*, vol. 58, no. 4, pp. 2131–2140, 2010.

[12] B. Ehsandoust, M. Babaie-Zadeh, and C. Jutten, "Blind source separation in nonlinear mixture for colored sources using signal derivatives," in *Latent Variable Analysis and Signal Separation*, pp. 193–200. Springer, 2015.

[13] B. Ehsandoust, B. Rivet, C. Jutten, and M. Babaie-Zadeh, "Nonlinear blind source separation for sparse sources," in *24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1583–1587.

[14] A. Taleb and C. Jutten, "Source separation in post nonlinear mixtures," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2807–2820, 1999.

[15] M. Babaie-Zadeh, *On blind source separation in convolutive and nonlinear mixtures*, Ph.D. thesis, INP Grenoble, 2002.

[16] F. Merrikh-Bayat, M. Babaie-Zadeh, and C. Jutten, "Linear-quadratic blind source separating structure for removing show-through in scanned documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 4, pp. 319–333, 2011.

[17] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Networks*, vol. 12, pp. 429–439, 1999.

[18] A. M. Kagan, Y. V. Linnik, and C. R. Rao, "Extension of Darmois-Skitovich theorem to functions of random variables satisfying an addition theorem," *Communications in statistics*, vol. 1, no. 5, pp. 471–474, 1973.

[19] A. Liutkus, R. Badeau, and G. Richard, "Gaussian processes for underdetermined source separation," *Trans. on SP*, vol. 59, no. 7, pp. 3155–3167, 2011.

[20] S. Noorzadeh, B. Rivet, and P. Guméry, "A multi-modal approach using a non-parametric model to extract fetal ecg," in *ICASSP*. IEEE, 2015, pp. 832–836.

[21] S. Noorzadeh, M. Niknazar, B. Rivet, J. Fontecave-Jallon, P. Guméry, and C. Jutten, "Modeling quasi-periodic signals by a non-parametric model: Application on fetal ecg extraction," in *36th Int. Conf. of EMBC*. IEEE, 2014, pp. 1889–1892.

[22] A. Hyvärinen, "Survey on independent component analysis," 1999.

[23] M. Girolami and C. Fyfe, "Negentropy and kurtosis as projection pursuit indices provide generalised ica algorithms," *Advances in Neural Information Processing Systems*, vol. 9, 1996.

[24] C. Hwang, "Simulated annealing: Theory and applications," *Acta Applicandae Mathematicae*, vol. 12, no. 1, pp. 108–111, 1988.

[25] D. T. Pham, "Blind separation of instantaneous mixture of sources based on order statistics," *IEEE Transactions on Signal Processing*, vol. 48, pp. 363–375, 2000.