Nonlinear Blind Source Separation for Chemical Sensor Arrays Based on a Polynomial Representation

Rafael Assato Ando[*†], Christian Jutten[*], Bertrand Rivet[*], Romis Attux[†] and Leonardo Tomazeli Duarte[‡]

[*]GIPSA-Lab, Grenoble INP, CNRS, Grenoble, France
[†]School of Electrical and Computer Engineering, University of Campinas (UNICAMP), Brazil
[‡]School of Applied Sciences, University of Campinas (UNICAMP), Brazil

Abstract—In this paper we propose an extension of a blind source separation algorithm that can be used to process the data obtained by an array of ion-selective electrodes to measure the ionic activity of different ions in an aqueous solution. While a previous algorithm used a polynomial approximation of the mixing model and mutual information as means of estimating the mixture coefficients, it only worked for a constrained configuration of two sources with the same ionic valence. Our proposed method is able to generalize it to any number of sources and any type of ions, and is therefore able to solve the problem for any configuration. Simulations show good results for the analyzed application.

I. Introduction

The blind source separation (BSS) problem is an important cornerstone in signal processing theory [1], and its applications include audio signal processing, telecommunications, image processing, brain-computer interface design, analysis of seismic data, among others. Essentially, the BSS problem consists of estimating a set of signals, also called sources, using measurements which are effectively a mixture of them, possibly including some *a priori* information on the nature of the desired signals and mixing process.

The standard approach for solving the BSS problem is based on the hypotheses that the mixing process is linear and that the source signals are statistically independent. In that case, successful algorithms such as *independent component analysis* (ICA) [1] have been proposed and extensively studied over the last years. However, for certain applications the linear model is insufficient, and a nonlinear mixing model might be required. Among the nonlinear BSS mixing models that have been studied so far, it is important to mention the *post-nonlinear* (PNL) [2], [3], in which a linear mixture is further distorted by a nonlinear function before the measurement, as well as the *linear quadratic* (LQ) model, which represents the mixtures as a second-order polynomial on the sources.

In this work, we shall focus on the problem of source separation for ion-selective electrode (ISE) arrays [4], [5]. ISEs are simple devices which are used for measuring the ionic

activity (essentially the effective concentration of an ion) in an aqueous solution. An ISE consists of a sensitive membrane for which the electrochemical potential varies according to the concentration of a specific ion. A well-known example of ISE is the glass electrode used for measuring the pH value [6].

The problem of the ISEs, however, is that the membranes are not perfectly selective. When trying to estimate the activity of a certain ion, the measurement will include not only the activity from the target ion, but also an interference from different ions which are also present in the same chemical solution. Because the measurements are a mixture of the activities, a BSS paradigm can be used to separate the signals from each ion by using an array of multiple ISEs.

The problem of estimating the ionic activities based on ISE arrays has already been studied in the past, with the use of a Bayesian source separation method [5], sparsity-based methods [7] and a PNL model and with a polynomial representation as well, but with limiting constraints which made it work only in specific circumstances [4]. In this paper, we shall propose a more general version of the polynomial algorithm, which should be able to account for all possible configurations, at the expense of additional complexity.

In the paper, we will initially present, in Section II, a description of the BSS problem statement and how it relates to measurements with chemical sensor arrays. In Section III, we will briefly present the previous algorithm for the constrained version of the problem, followed by an explanation of the proposed method for its general version. We shall then present experimental results in Section IV, and a conclusion and future plans shall finally be presented in Section V.

II. Problem Statement

A. Blind Source Separation

For the problem of BSS, let us consider a set of N sources $\mathbf{s}(k) = [s_1(k), s_2(k), ..., s_N(k)]^T$ and M mixtures $\mathbf{x}(k) = \mathcal{F}[\mathbf{s}(k)] = [x_1(k), x_2(k), ..., x_M(k)]^T$, where n is the temporal index and $\mathcal{F}[.]$ is the mixing function, not necessarily linear. While the number of sources and mixtures do not have to be necessarily equal, in our analysis we shall consider the determined BSS problem, in which $N = M$. The goal of the

source separation problem is to obtain the sources $\mathbf{s}(k)$ given the mixtures $\mathbf{x}(k)$, and possibly some *a priori* information on the properties of the sources (e.g. independence, sparsity, non-negativity, etc.) and/or on the mixing process (e.g. linearity, PNL model, etc.). The problem is said to be *non-blind* if the mixing function $\mathcal{F}[.]$ is known in advance, and it is otherwise said to be *blind*. If the function $\mathcal{F}[.]$ is linear, the mixing function can be represented by a matrix:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) \tag{1}$$

where $\mathbf{A}$ is the $N \times M$ mixing matrix. For the linear case, there are several classical methods in the literature that can solve the BSS problem under different priors for the sources. A very well-known class of methods is the *independent component analysis* (ICA) [1].

For the nonlinear case, however, no general solution exists, and the problem has to be analyzed on a case-by-case basis [1], depending on the nature of the mixing function. Among the commonly studied nonlinear models are the post nonlinear (PNL) [3] and polynomial models [8].

B. Chemical Sensor Arrays

For the problem of chemical sensor arrays, let us consider an aqueous solution in which we have N different ions and M ion selective electrodes (ISE) to measure their activities s_i, which are our sources. However, due to the interference problem, the values measured are actually a nonlinear mixture of the activities of each ion. According to the Nicolsky-Eisenman (NE) equation [9], this interference can be approximately modeled by the equation below:

$$y_i(k) = e_i + \frac{RT}{z_i F} \ln \left(s_i(k) + \sum_{j=1, j \neq i}^{N} a_{ij} s_j^{z_i/z_j}(k) \right) \tag{2}$$

where y_i are the observed values (our mixtures), e_i is the standard electrode potential (a scalar constant), R is the universal gas constant, T is the temperature, F is the Faraday constant, z_i is the valence of the ith ion, and a_{ij} are the selectivity coefficients. The $\frac{RT}{z_i F}$ slope is called the *Nernst slope*, and at room temperature of 25°C, it is approximately 26mV, or 59mV if the logarithm is converted to base 10 (which is the standard practice in chemical applications).

III. METHODS AND ALGORITHM

A. Previous Algorithm

In the simplest version of the problem, we have the case of two sources and two mixtures, in which both ions have the same valences, in which case the model can be written as:

$$\mathbf{y} = \mathbf{e} + \mathbf{d} \circ \log(\mathbf{A}\mathbf{s}) \tag{3}$$

where $\mathbf{A}$ is a 2×2 matrix with $a_{11} = a_{22} = 1$, $\mathbf{d}$ is the vector containing the estimates of the slope for each ISE, and the operator $\circ$ denotes the Hadamard (entrywise) matrix product[1]. While the slopes represented by $\mathbf{d}$ can be

[1]The logarithm operation is also performed entrywise.

theoretically calculated directly by (2), the empirical nature of the NE equation cannot always accurately model the data, and better results can be obtained by giving the model an additional degree of freedom, allowing it to vary the slope.

In our previous work, we had estimated the $\mathbf{d}$ slope values using theoretical values and defined the modified mixture set $\mathbf{x}$, with

$$x_i = 10^{\frac{y_i - e_i}{d_i}}. \tag{4}$$

If the slope values were exact (and the mixture were perfectly modeled by (2), which is not the case), it can be shown that $\mathbf{x} = \mathbf{A}\mathbf{s}$, which is a linear mixture of the two sources which can then be solved by classical linear BSS algorithms. This results in the classical PNL approach in which the difficult part is to estimate the parameters that correctly cancel the nonlinearity - in this case the slopes $\mathbf{d}$. However, in our previous work we had simply used the theoretical slope value, and together with the fact that (2) only approximately models the mixture, we can only claim that the modified set $\mathbf{x}$ is an approximately linear mixture of the sources. The idea, then, was to represent $\mathbf{x}$ as a second degree polynomial, in which the quadratic terms can correct small nonlinearities in the modified mixture. The mixture was represented then as

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{s}^{\circ 2} + \mathbf{c}s_1 s_2 \tag{5}$$

where $a_{11} = a_{22} = 1$ and $\mathbf{s}^{\circ n} = [s_1^n, s_2^n]^T$ represents the entrywise power operator. This mixing model encompasses the class known as linear-quadratic models, for which one can find several methods to deal with [8]. Since the mixing model is not as simple as the linear case, inverting it is no longer trivial, and a recurrent network seems indicated to solve it even in the non-blind case [8].

This algorithm, however, only works for the simplest case in which we had two mixtures of two sources with the same valence.

B. General case

A more general case could contain any number of sources and mixtures, and the ions could have different valences. The valences of the ions are known in advance. We shall still restrain ourselves to the case in which the number of sources and mixtures are equal - that is, the determined case.

It is important to start by noticing that when the valences are different, the mixing model is no longer PNL, even if the NE equation (2) were able to perfectly model the mixture. This is because, in (2), we are actually mixing $s_j^{z_i/z_j}$, with z_i and z_j possibly different. A useful change of variable can be made by defining the *pseudo-source*:

$$p_i(k) = (s_i(k))^{1/z_i} \tag{6}$$

Then (2) becomes

$$y_i(k) = e_i + \frac{RT}{z_i F} \ln \left(\sum_{j=1}^{N} a_{ij} p_j^{z_i}(k) \right). \tag{7}$$

The associated modified mixture, as defined in (4), becomes then polynomial in the pseudo-sources $\mathbf{p}$ with degree $k = max(\mathbf{z})/gcd(\mathbf{z})$, where $gcd(\mathbf{z})$ is the greatest common denominator of all the valences of z_i. Since we want to add an additional degree of freedom in order to capture the inaccuracies of the NE-model, we can therefore model the modified mixture set as a polynomial of degree $k + 1$.

Since the modified model is nonlinear, even in the blind case the solution is not trivial, even more so in the general case in which we can have multiple sources and different valences. Therefore, an analytical solution is generally impossible to obtain, we resort to a recurring network to achieve the solution numerically.

C. Recurring Network

Using the definition of our pseudo-sources vector $\mathbf{p}$, we can rewrite the modified mixture vector $\mathbf{x}$, with each component given by (4) as

$$\mathbf{x} = \mathcal{F}[\mathbf{p}] \qquad (8)$$

where $\mathcal{F}[.]$ is a polynomial of the appropriate degree, including all possible cross-multiplied terms of the pseudo-sources p_i. For example, for the case of 3 sources with valences $+1$, $+2$ and $+3$, our polynomial would have degree 4, and could be written as:

$$\mathbf{x} = \sum_{i=1}^{4} \left(\mathbf{A_i} \mathbf{p}^{\circ i} \right) + \mathbf{B} \mathbf{p_{cross}} \qquad (9)$$

where $\mathbf{A_i}$ are square matrices, $\mathbf{p_{cross}}$ contains all the cross multiplied terms of the pseudo-sources and $\mathbf{B}$ is a coefficient matrix of appropriate dimension. While it can be said that the modified mixture set given by (4) should not contain cross multiplied terms, this would only be true if both the estimates of the Nernst slopes and the NE equation (2) were perfectly correct. Since they are in fact only empirical approximations, the additional degrees of freedom provided by the cross multiplied terms, as well as the higher order of the polynomial, can be used to correct inaccuracies in the model.

If the coefficients are known (i.e. in the non-blind case), we have to solve the nonlinear equation system:

$$\mathcal{G}[\mathbf{p}] = \mathcal{F}[\mathbf{p}] - \mathbf{x} = 0. \qquad (10)$$

Applying the Newton-Rhapson root-finding algorithm to (10), we obtain the following recurrent network:

$$\mathbf{q_{t+1}} = \mathbf{q_t} - \mathbf{J}_{\mathcal{G}}^{-1} \left(\mathbf{q_t} \right) \mathcal{G}[\mathbf{q_t}] \qquad (11)$$

where $\mathbf{J}_{\mathcal{G}}$ is the Jacobian of $\mathcal{G}$. Since $\mathcal{G}$ is a polynomial, this is easy to calculate. As usual, to lower the computational cost of the operation, it is better to solve the linear equation system $J_{\mathcal{G}}(\mathbf{q}_t)\Delta\mathbf{q}_t = -\mathcal{G}[\mathbf{q_t}]$ instead of inverting the matrix. When the network converges, $\mathbf{q}$ would eventually converge to the pseudo-sources $\mathbf{p}$.

However, in the problem we are trying to solve, the mixing coefficients are unknown, since the problem is blind. Thus, before being able to use this recurrent network we need a way to estimate the polynomial coefficients first.

D. Estimation of the Polynomial Coefficients

In order to estimate the coefficients of the mixture, we assume that the sources are independent, which automatically implies that our pseudo-sources are also independent. It is important to notice that source independence is a common hypothesis in many BSS problems, and is a good assumption in our application as well.

In order to maximize the independence of the estimated sources, in the first algorithm, for 2 sources, we used the mutual information as a cost function for estimating the parameters, with good results. While different generalizations of the concept of mutual information for more than two sources exists, we can use the *total correlation* for our problem [10]. The total correlation can be defined as the *Kullback-Leibler divergence* between the joint distribution and the independent distribution, which can be simplified to:

$$C(X_1, X_2, ..., X_n) = \left(\sum_{i=1}^{n} H(X_i) \right) - H(X_1, X_2, ..., X_n) \qquad (12)$$

where $C(X_1, X_2, ..., X_n)$ is the total correlation, $H(X_i) = -p_i(x_i) \log (p_i(x_i))$ is the information entropy of X_i and

$$H(X_1, X_2, ..., X_n) = \\ - \sum_{x_1} ... \sum_{x_n} p(x_1, ..., x_n) \log (p(x_1, ..., x_n)) \qquad (13)$$

is the joint entropy of the variables. It is important to notice that since the total correlation is a *Kullback-Leibler divergence*, we have $C \geq 0$ with equality if, and only if, the random variables are statistically independent. Therefore, when trying to obtain polynomial coefficients that produces independent source estimates, the total correlation can be used as a cost function, and we can use the gradient method to minimize it.

A detailed derivation of the differential of the mutual information (or, for the case of more than two sources, the total correlation) can be found in [11]. It can be calculated approximated by

$$\frac{\partial C}{\partial \mathbf{w}} = E \left\{ \frac{\partial \mathbf{q}}{\partial \mathbf{w}}^T \beta_{\mathbf{q}}(\mathbf{q}) \right\} \qquad (14)$$

where $\mathbf{w}$ is the vector containing all the polynomial coefficients, and

$$\beta_{q_i}(\mathbf{q}) = \left(-\frac{\partial \log p(\mathbf{q})}{\partial q_i} \right) - \left(-\frac{d \log p(q_i)}{d q_i} \right) \qquad (15)$$

is what can essentially be interpreted as "gradient" of C with respect to the source estimates $\mathbf{q}$ [11].

By defining the *score function* as the opposite of the log-derivative of a density function, i.e.,

$$\phi(x) = -\frac{d}{dx}\ln p_x(x) \tag{16}$$

it can be seen that $\beta_{q_i}(\mathbf{q})$ is the difference between the ith component of the joint score function of $\mathbf{q}$ and the marginal score function of q_i. It is interesting to note that the score functions, and, by extension, $\beta_{\mathbf{q}}(\mathbf{q})$ as defined in (15), can be accurately computed by efficient methods as described, for instance, in [12], which leads to an efficient computation of our cost function.

As a result, the iteration to find the parameters can be written as:

$$\mathbf{w_{k+1}} = \mathbf{w_k} - \mu\frac{\partial C}{\partial \mathbf{w_k}} \tag{17}$$

where μ denotes the learning rate[2].

Note that the expression for $\frac{\partial \mathbf{q}}{\partial \mathbf{w}}$ would depend on the mixing model. For our problem, we can start from the modified mixing model (8) and apply implicit derivatives, replacing $\mathbf{p}$ (our desired result) by $\mathbf{q}$ (our estimates). For instance, for the example with 3 sources with valences $+1$, $+2$ and $+3$ shown in (9), we would have:

$$
\begin{aligned}
\mathbf{0} &= \frac{\partial}{\partial \mathbf{w}}\left(\sum_{i=1}^{4}\left(\mathbf{A_i}\mathbf{q}^{\circ i}\right) + \mathbf{B}\mathbf{q_{cross}}\right) \\
&= \frac{\partial}{\partial \mathbf{w}}\left(\sum_{i=1}^{4}\left(\mathbf{A_i}\mathbf{q}^{\circ i}\right) + \mathbf{B}\mathbf{q_{cross}}\right)\bigg|_{\mathbf{A_i},\mathbf{B}\text{ constant}} \\
&\quad + \frac{\partial}{\partial \mathbf{w}}\left(\sum_{i=1}^{4}\left(\mathbf{A_i}\mathbf{q}^{\circ i}\right) + \mathbf{B}\mathbf{q_{cross}}\right)\bigg|_{\mathbf{q},\mathbf{q_{cross}}\text{ constant}}
\end{aligned}
\tag{18}
$$

where the product rule is used. The first expression, with the constant matrices, is $\mathbf{J}_{\mathcal{G}}\frac{\partial \mathbf{q}}{\partial \mathbf{w}}$. For the second expression, with the constant vectors, notice that our coefficient vector $\mathbf{w}$ contains all the variables from matrices $\mathbf{A_1}, ..., \mathbf{A_4}, \mathbf{B}$. Let us define $\mathbf{w}$ so that the elements are ordered in such a way that all variables from $\mathbf{A_1}$ appear first, followed by the elements of $\mathbf{A_2}$, and so on until $\mathbf{B}$, and on each matrix the elements are extracted row-by-row.

The second expression, then, would be a matrix $\mathbf{K} = (k_{ij})$ with $k_{ij} \neq 0$ if, and only if, the variable represented by column j appears in the equation for mixture i, and in that case, k_{ij} is the coefficient of that variable. This can be concisely written with the use of the *Kronecker product*. The Kronecker product of two matrices of arbitrary sizes, denoted by $\mathbf{A} \otimes \mathbf{B}$, is the block matrix defined by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}. \tag{19}$$

The expression with constant vectors from (18) can, therefore, be written as:

---

[2]Smaller learning rates improve the algorithm's robustness at the expense of longer convergence times. In our simulations we used $\mu \in [0.001; 0.01]$.

$$
\begin{aligned}
\mathbf{K} &= \begin{bmatrix} I_3 \otimes \mathbf{q}^T & \cdots & I_3 \otimes \left(\mathbf{q}^{\circ 4}\right)^T & I_3 \otimes \mathbf{q}^T_{cross} \end{bmatrix} \\
&= -\mathbf{J}_{\mathcal{G}}\frac{\partial \mathbf{q}}{\partial \mathbf{w}}
\end{aligned}
\tag{20}
$$

where I_3 is the identity matrix of order 3 (the number of sources in our example) and, with the Kronecker product, the expression of $\mathbf{K}$ is written as block matrix. Using (20), we can finally isolate $\frac{\partial \mathbf{q}}{\partial \mathbf{w}}$, as necessary for the iterations of the gradient method proposed.

E. Algorithm

The algorithm can then be summarized as follows:
1) From the mixtures $\mathbf{y}$, obtain the modified mixture set $\mathbf{x}$ as defined by (4).
2) Start with random parameters $\mathbf{w}$.
3) Estimate the pseudo sources $\mathbf{p}$ using the recurrent network (11) with the weights $\mathbf{w}$.
4) Update $\mathbf{w}$ using the gradient of the total correlation of the estimates, calculated via equations (15) to (20)
5) If stopping criterion is not met, return to step 3.
6) After convergence, recover the sources $\mathbf{s}$ from the pseudo-sources $\mathbf{p}$, using equation (6).

It is important to notice that while this paper shows the example of three sources with valences $+1$, $+2$ and $+3$, the same algorithm works with any number of sources and any valences. The number of sources and valences, however, must be known in advance since they are required for calculating the Jacobian $\mathbf{J}_{\mathcal{G}}[.]$, which is used both on steps 3 and 4.

IV. EXPERIMENTAL RESULTS

In our simulations, we deviate from the constrained (2 sources with equal valences) model by either using 3 sources, using sources with different valences, or both. None of the 3 cases presented could have been solved by the previous algorithm, whereas the proposed method was able to obtain solutions in all cases. We present here the results for the worst case scenario, in which both variations occur – a good representation of the more general case in which we have more than 2 sources and different valences. Our simulation had a sample size of $K = 400$. In Fig. 1, we can see the results of the simulation.

We can see in the first row of the figure three independent and uniformly distributed sources in the interval $[0, 1]$, but with the samples of the first one being sorted in ascending in order to be easier to visualize, but without affecting the independence of the sources. The sources are then mixed according to Eq. (2), as shown in the second row.

The estimates obtained by the algorithm can be seen in the third line, and at the end we can see a plot of the estimates and the sources. Ideally, it should be noted that we want to obtain a straight line, of any inclination, since it only indicates a different scalar factor. We can see that for the three sources, we obtain what can be roughly approximated by a straight line, indicating that the result is good. In order to measure
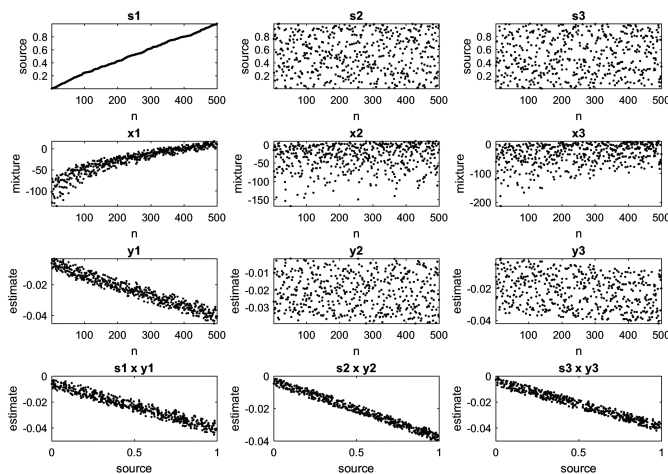
Fig. 1. Plots of sources and estimates

how well the estimates represent the sources, we can use the *signal to interference* ratio, which can be defined, in decibels, as:

$$\text{SIR} = 10 \log_{10} \frac{E\{\mathbf{s}^2\}}{E\{(\mathbf{s} - \mathbf{y})^2\}} \quad (21)$$

where $\mathbf{s}$ is the source vector, $\mathbf{y}$ is the estimate vector, and the higher the SIR value the more accurate the estimate is.

For the simulation here, the SIR values obtained were 20.0 dB, 26.6 dB and 21.4 dB, which are typically considered as good values for the application analyzed.

It should also be noticed that because of the random nature of the algorithm, both in the initial point for the recurrent network, and the initial set of parameters, the algorithm does not always converge. When the algorithm fails to converge, we can simply rerun it with a different set of initial conditions until an estimate is found. This effect, however, is not common enough not to be an issue – in fact, on a Monte Carlo simulation with 200 trials, in which we keep the same sources, mixing model, but randomized initialization points and weights, 181 trials succeeded, for a convergence rate of 90.5

V. CONCLUSION

In this paper, we have generalized an algorithm proposed in [4] which was able to show promising results when applied to the measurement of ionic concentrations using chemical sensors, but was limited only to two sources that had to be ions of the same valence. Using similar ideas, we were able to generalize the algorithm to any number of sources with possibly different valences. In fact, the algorithm is general enough that it is possible to use the same approach for different applications as well, with more general mixing models and source configurations.

By allowing the valences to be different, the resulting mixture can no longer be classified as *post non-linear* (PNL) and, as a result, the method previously found in the literature [13] cannot be compared to our proposed method. Nonetheless, while not as good as the simpler case of 2 sources with equal valence, our simulations indicate that the separation can be performed with accuracy, obtaining SIR between 20 and 27dB, which are typically good for this sort of application.

In the future, we would like to further analyze the theoretical stability of the problem in order to improve the selection of initial conditions for the algorithm. For the current version of the algorithm, the conditions are randomly selected and can sometimes lead to a non-convergence of the algorithm, in which case we restart the method until it succeeds. While this has not been an issue for our current application (possibly due to the fact that the mixing coefficients are typically small), a more detailed theoretical analysis would be important to avoid problems with more general mixing models, and possibly improve the accuracy and convergence speed of our proposed method.

REFERENCES

[1] P. Comon and C. Jutten, "Handbook of blind source separation," *Elsevier*, 2010.

[2] S. Achard, D.-T. Pham, and C. Jutten, "Criteria based on mutual information minimization for blind source separation in post nonlinear mixtures," *Signal processing*, vol. 85, no. 5, pp. 965–974, 2005.

[3] L. T. Duarte, R. Suyama, R. Attux, F. Von Zuben, and J. M. Romano, "Blind source separation of post-nonlinear mixtures using evolutionary computation and order statistics," in *Independent Component Analysis and Blind Signal Separation*, pp. 66–73. Springer, 2006.

[4] R. A. Ando, L. T. Duarte, C. Jutten, and R. Attux, "A blind source separation method for chemical sensor arrays based on a second order mixing model," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 933–937.

[5] L. T. Duarte, C. Jutten, and S. Moussaoui, "A Bayesian nonlinear source separation method for smart ion-selective electrode arrays," *Sensors Journal, IEEE*, vol. 9, no. 12, pp. 1763–1771, 2009.

[6] K. N. Mikhelson, *Ion-selective electrodes*, vol. 81 of *Lecture Notes in Chemistry*, Springer, 2013.

[7] L. Tomazeli Duarte, R. Suyama, R. Attux, J.M. Travassos Romano, and C. Jutten, "A sparsity-based method for blind compensation of a memoryless nonlinear distortion: Application to ion-selective electrodes," *Sensors Journal, IEEE*, vol. 15, no. 4, pp. 2054–2061, April 2015.

[8] Y. Deville and S. Hosseini, "Recurrent networks for separating extractable-target nonlinear mixtures. part i: Non-blind configurations," *Signal Processing*, vol. 89, no. 4, pp. 378–393, 2009.

[9] E. Bakker, E. Pretsch, and P. Bühlmann, "Selectivity of potentiometric ion sensors," *Analytical Chemistry*, vol. 72, no. 6, pp. 1127–1133, 2000.

[10] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM Journal of research and development*, vol. 4, no. 1, pp. 66–82, 1960.

[11] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Differential of the mutual information," *IEEE Signal Processing Letters*, vol. 11, no. 1, pp. 48–51, 2004.

[12] D.-T. Pham, "Fast algorithms for mutual information based independent component analysis," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2690–2700, Oct. 2004.

[13] L. T. Duarte, J. M. Romano, C. Jutten, K. Chumbimuni-Torres, and L. Kubota, "Application of blind source separation methods to ion-selective electrode arrays in flow-injection analysis," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2228–2229, 2014.