BLIND SOURCE SEPARATION IN AN ARRAY OF ISFET SENSORS: II. SIMULATIONS

G. BEDOYA, S. BERMEJO
UNIVERSITAT POLITECNICA DE CATALUNYA (UPC), SPAIN

KEYWORDS: Blind Source Separation, Independent Component Analysis, Sensor Arrays, ISFETS, Unsupervised learning, Neural Networks.

ABSTRACT: In this work we explore the application of Blind Source Separation (BSS) and Independent Component Analysis (ICA) algorithms as signal processing tools in an array of ISFET sensors. BSS is an emerging technique for array processing and data analysis that aims to recover unobserved signals from observed mixtures, i.e. the outputs of the sensors of the array. As [1] shows, the signals from the outputs of ISFET sensors array can be considered as a mixture of independent ion concentration, so BBS techniques are suitable to recover the original concentrations from the observed mixture. Preliminary experiment results show, that a source signal separation of the outputs of an ISFET sensors to extract ion concentration can be achieved using BSS and ICA algorithms. We have employed known signals and theoretical estimations of the sensors output signals to achieve source separation by means of the OOLABSS software and the FastICA algorithm.

INTRODUCTION

In practical applications, sensors can inherently suffer from imperfections (nonlinearly, sensitivity to temperature) and interference from the environment. In order to improve sensor performance we propose a system architecture that includes signal-processing tools with the role of allows a signal enhancement for the extraction of useful information.

In the SEWING project, we will consider an array of ISFET sensors, as the front-end of the data acquisition system which aims to detect several ion concentrations on the water. As [1] shows, the output of this array can be considered as a mixture of several ion concentrations and an additive noise caused by the interference in the sensing process of the multiple ions located in the water sample. Given this mixture corrupted by noise, the central problem is to recover the original signals, i.e. the ion concentrations.

Independent component analysis (ICA) is emerging as a new standard in signal processing and data analysis. ICA has received much attention over the last years in the field of neural computation due to its potential application to the array signals processing, such a speech and natural images. ICA algorithms have been considered to be information-theory-based unsupervised learning rules. Given a set of multidimensional observations, which are assumed to be linear mixtures of unknown independent sources through an unknown mixing source, an ICA algorithm performs a search of the demixing matrix by which observations can be linearly translated to form independent output components. Consequently, ICA has been proposed as a solution to the blind source separation (BSS) problem in which the goal is to recover the waveforms of unobserved signals, from observed mixtures without knowing the mixing coefficients.

The contents of this paper are organised as follows. Section 2 introduces the principles of independent component analysis (ICA) that complements [1]. In section 3, discussion about the use of an on-line approach is given, and the way to obtain the separation using an adaptive linear neural network that performs the linear transformation . Finally, conclusions are given.

PRINCIPLES OF INDEPENDENT COMPONENTS ANALYSIS

Assume that the M unknown mutually independent sources are of zero mean and are denoted by a random vector $s = [s_1, \ldots s_M]$; and the observations are samples from the linear transformation of these independent sources via an unknown matrix, such a:

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

where $\mathbf{x}$ denotes the observed mixed data, $\mathbf{s}=\{s_i\}$. contains the original source signals and $\mathbf{A}$ is an unknown mixing data. The goal of ICA is to estimate the original sources $\mathbf{s}$ given only the observations $\mathbf{x}$, given the realistic assumption of the independence of the sources, in the context of sensing unrelated physical information.

A set of independent random variables have a joint distribution that is the product of all individual marginal distribution [1]. Hence, ICA aims to recover original sources through a set of output components of which the joint distribution is as close as possible to the product of their marginal distributions

$$p(y) = \prod_{i=1}^{M} pi(yi) \qquad (2)$$

where p_i denotes the marginal distribution of the *i*th recovered source or output component y_i. The estimation y of independent sources is the linear transformation of the observations by means of the demixing matrix W as follows:

$$\mathbf{y} = \mathbf{Wx} = \mathbf{WAs}$$

where **A** must be the inverse of **W** (i.e., $\mathbf{W} = \mathbf{A}^{-1}$)in order to recover the originals signals **s**. The dependency among output components **y** can be quantitatively measured by the Kullback–Leibler (KL) divergence[2], which is the expected value of the log ratio of the joint distribution to the product of the marginal distributions. The KL divergence is defined by

$$D(y) = \int p(y)\log\frac{p(y)}{\prod_{i=1}^{M} pi(yi)}\,dy \qquad \textbf{(3)}$$

The way in which Equation (4) can be approximated and then minimised has produced different ICA algorithms. The fundamental restriction of classic ICA is that the independent components must be non-gaussian [3][4].
To use non-gaussianity in ICA estimation, we must have a quantitative measure of non-gaussianity of the random variable **y**. To simplify the analysis we assume that **y** is centred (zero mean) and has a variance equal to one. Then, the classical measure of non-gaussianity is Kurtosis or the fourth order cumulant. The Kurtosis of y is classically defined by

$$\text{Kurt (y)} = E\{y^4\}\text{-}3(E\{y^4\})^2 \qquad \textbf{(4)}$$

ON-LINE ALGORITHMS FOR BSS & ICA

The choice of the ICA algorithm depends on several factors including the type of objective function employed for measuring independence and non-gaussianity, the optimisation method used for achieving a minimum (or a maximum) of this function and the way in which independent components are extracted. One popular class of learning algorithms are based in stochastic on-line optimisation [5]. In the on-line approach, the learning algorithms typically perform a stochastic deepest descent (e.g. gradient descent) over the objective function using only one training sample $\mathbf{x_i}$ to update the demixing matrix **W**. , On the other extreme we have batch learning algorithms that use the whole training set $\{\mathbf{x_i}\}$in order to update **W**. Examples of batch algorithms are the tensor-based methods [2], which are efficient in small dimensions but they cannot be used in larger dimensions, and the FastICA algorithm [6] that is one of the most promising solution for linear ICA due to its simplicity and fast convergence. However we will use here a class of on-line adaptive algorithm with equivariant properties and with self-adaptive learning rate [7][8][9][10][11]. On the other hand, the independent components can be estimated at the same time, as the algorithm proposed in [4] that is based in the natural gradient ascent of likelihood, or extracted one-by-one using a deflation technique like the Fast ICA performs.

In order to introduce the notation needed in the rest of this document, let us assume that $s_i(k)$ are an unknown number of stochastic independent sources linearly mixed by unknown coefficients according to the matrix equation

$$\mathbf{x(k) = As(k)}$$

where $\mathbf{s}(k) = [s_1(k),...,s_n(k)]$ is a vector of unknown original sources, x(k) is the observed vector of mixed sensor signals, k denotes the discrete-time index.

Using an on-line approach, the separation can be performed using an adaptive linear neural network that performs the linear transformation

$$\text{y (k)} = \text{W(k) x(k)}$$

where $\mathbf{W}(k) = w_{ij}$ is a *n* x *m* non-singular matrix of weights w_{ij} updated, according to the following on-line learning rule [7]:

$$\text{W(k+1)} = \text{W(k)} + \eta(k)\widetilde{\text{G}}\text{(k)} \qquad \textbf{(5)}$$

(k = 0,1,2….)

where *k* is the iteration index, *η(k)* > 0 is a learning rate (step size), and

$$\widetilde{G}(k) = \widetilde{G}[W(k), x(k), y(k)] = [\widetilde{g}_{ij}(k)]_{m\,x\,n}$$

is a gradient flow matrix depending on **W**(k), **x** (k) and **y** (k).

In the equation (5), the learning rate can be computed in different ways:

$$\eta(\text{k}) = \Phi(\text{k}) \text{ defined function} \qquad \textbf{(6)}$$

$$\eta\,(\text{k}) = (1 - \rho_1)\,\eta(k-1) + \rho_1\beta\psi\left\lfloor \left\|\hat{G}(k)\right\| \right\rfloor \qquad \textbf{(7)}$$

$$\eta\,(\text{k}) = [1 - \rho_1\,\eta\,(k-1)]\eta\,(k+1) + \rho_1\beta\;\eta\,(k-1)\psi\left[\left\|\hat{G}\,(k)\right\|\right] \qquad \textbf{(8)}$$

where $\hat{\text{G}}$ (k) (an average of $\widetilde{G}$) is an *n* x *m* matrix and can itself be computed or estimated using one of the below equations:

$$\text{G (k)} = (1 - \rho_2)\,\hat{\text{G}}\,(\text{k}-1) + \rho_2\hat{G}(\kappa) \qquad \textbf{(9)}$$

$$\hat{G}(k)=\frac{1}{M}\sum_{i=0}^{M-1}\gamma^{i}\widetilde{G}(k-i) \qquad \textbf{(10)}$$

Equation (5) can be presented in various equivalent forms using a vector or component notation:
Vector form:

$$\mathrm{w}_{\mathrm{i}}\ (\mathrm{k}+1)=\mathrm{w}_{\mathrm{i}}\ (\mathrm{k})+\eta_{\mathrm{i}}(\mathrm{k})\ \widetilde{\mathrm{G}}(\mathrm{k}) \qquad \textbf{(11)}$$

Component form:

$$w_{ij}(k+1)=w_{ij}(k)+\eta_{ij}(k)\widetilde{g}_{ij}(k) \qquad \textbf{(12)}$$

where $w_i = [w_{i1},\dots, w_{in}]^T$ is the i[th] column of **W**, $\widetilde{G}_i$ is the i[th] column of $\widetilde{G}$, $\widetilde{g}_{ij}$ is the ij[th] element of $\widetilde{G}$; η_i and η_{ij} are computed using suitable modifications of equations (7) and (8). According to equations (5) (11) and (12), the weights w_{ij} are modified using a global or local) learning rate. Thus the learning rate can be a scalar (equation 5), a vector (equation 11) or a matrix (equation 12).
It is worth noting that the above description corresponds to a feed-forward layer neural network The algorithm that we use is based on non-linearities (activation functions) corresponding to the original algorithms that were proposed for BSS and ICA.

Initially we have to define the instantaneous gradient $\boldsymbol{\widetilde{G}}$(k), which can take many forms. However, most of the known learning algorithms without pre-whitening can be generalised or represented in the following unified form:

$$\widetilde{\mathrm{G}}(\mathrm{k})=\Big[\Lambda(\mathrm{k})-\underline{\mathrm{y}(\mathrm{k})\mathrm{y}^{\mathrm{T}}(\mathrm{k})}-\underline{\mathrm{f}(\mathrm{y}(\mathrm{k}))\mathrm{g}^{\mathrm{T}}(\mathrm{y}(\mathrm{k}))}+ \qquad \textbf{(13)}$$

$$+\underline{\mathrm{g}(\mathrm{y}(\mathrm{k}))\mathrm{f}^{\mathrm{T}}(\mathrm{y}(\mathrm{k}))}\Big]\begin{cases}\mathrm{I}\\ \mathrm{W}(\mathrm{k})\\ \mathrm{W}^{-\mathrm{T}}(\mathrm{k})\end{cases}$$

where the underlined terms are optional
For the adaptive activation functions, we can select the vector functions f(y) and g(y) between:

$$\mathrm{f}(\mathrm{y})=\alpha\,\mathrm{y}+\mathrm{sign}_{\delta}\big(\mathrm{k}_4(\mathrm{y})\big)\tanh(\beta\,\mathrm{y}) \qquad \textbf{(14)}$$

And

$$\mathrm{g}(\mathrm{y})=\mathrm{y} \qquad \textbf{(15)}$$

where:

$$\mathrm{sign}_{\delta}(\mathrm{x})=\begin{cases}1 & \text{if } \mathrm{x}\ \rangle\ \delta\\ -1 & \text{if } \mathrm{x}\ \langle -\delta\\ 0 & \text{otherwise}\end{cases}$$

and

$$\mathrm{f}_{\mathrm{i}}(\mathrm{y}_{\mathrm{i}})=\begin{cases}\tanh(\beta\mathrm{y}_{\mathrm{i}}) & \text{for } \mathrm{k}_4\ \rangle\ \delta\\ 0 & \text{otherwise}\end{cases} \qquad \textbf{(16)}$$

$$\mathrm{g}_{\mathrm{i}}(\mathrm{y}_{\mathrm{i}})=\begin{cases}\mathrm{y}_{\mathrm{i}} & \text{for } \mathrm{k}_4(\mathrm{y}_{\mathrm{i}})\ \rangle -\delta\\ \tanh(\beta\mathrm{y}_{\mathrm{i}}) & \text{otherwise}\end{cases} \qquad \textbf{(17)}$$

METHODOLOGY FOR THE ISFETs SENSOR MEASUREMENTS

The development of microsensors needs a specific methodology and instrumentation. For the ISFET-based sensors the electrical parameters to be measured are equal to the MOS´s transistors (*V_{th}, I_{OFF}, β, etc.)* but, due to the lack of connection to the electrode of gate, the way of obtaining them is different. The work environment of this sensors (aqueous solution) make necessary that we should control some parameters more than in the MOS transistors (e.g. the drifts of threshold voltage and the gate leakage currents). The sample signals for the simulations were obtained from [12] and [13] based on the effective detection range of the CHEMFETs proposed in [12], i.e., measurements taken inside the range of values included between -4.5 up to-1.5 of $\log a_i$. Where a_i represents the activities of the respective ions in the aqueous solution.

NUMERICAL SIMULATION

In the following simulations, we use the performance measure, denoted as PI (k), proposed by Cichocki and Orsier [9],

$$\mathrm{PI}(\mathrm{k})=\frac{1}{\mathrm{m}}\sum_{\mathrm{i}=1}^{\mathrm{m}}\left\{\sum_{\mathrm{j}=1}^{\mathrm{m}}\frac{|\mathrm{p}_{\mathrm{ij}}|^2}{\max_{\mathrm{q}}|\mathrm{p}_{\mathrm{iq}}|^2}-1\right\}+\frac{1}{\mathrm{m}}\sum_{\mathrm{j}=1}^{\mathrm{m}}\left\{\sum_{\mathrm{i}=1}^{\mathrm{m}}\frac{|\mathrm{p}_{\mathrm{ij}}|^2}{\max_{\mathrm{q}}|\mathrm{p}_{\mathrm{qj}}|^2}-1\right\} \qquad \textbf{(18)}$$

Where p_{ij} denotes the joint element of the i[th] row and the j[th] column of the product of the mixing matrix and the demixing matrix. With this measure, we will test the performance and the stability of the algorithm.

For the first test, in order to demonstrate the performance of the proposed solution in the sensors signals processing system, we will initially use the following sources that have been used by Amari et al. in [8]: s (t) = [sign (cos (2π 155t)), sin (2 π300t + 6cos (2 π60t)), sin (2 π90t), r (t)]. These sources represents signals simultaneously present in the sensor array, where the first four components are modulating the data signals and r (t) is a noise uniformly distributed in the interval [-1,1]. We assume that the four sources are unknown to the algorithms and are mixed by a randomly generated mixing matrix **A**. the source signals (representing the ion concentration) are shown in Fig.1. On the other hand, the mixed signals (representing the sensor signals) are sampled at sampling rate of 1KHz. We feed 10000 samples of the mixed signals (Fig. 2) to the algorithm. For this first test, we have performed two simulations using the OOLABSS software, which can be downloaded from the homepage provided in[11].

In the first simulation, we solve this problem using the self-adaptive learning rate proposed by Cichocki et al. [9] based on the use one of individual learning rates where each of the 16 elements of the demixing matrix has its own learning rate. Notice that the values of the learning rates decrease automatically, and then adapt to the change in the environment.

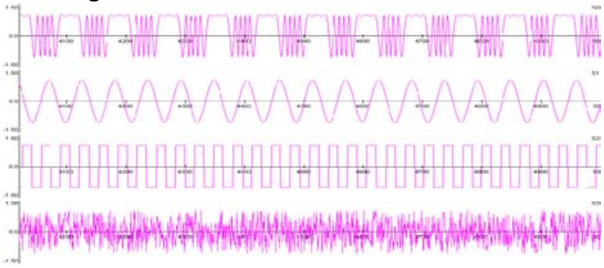

Figure 1. The four source signals in the first test.

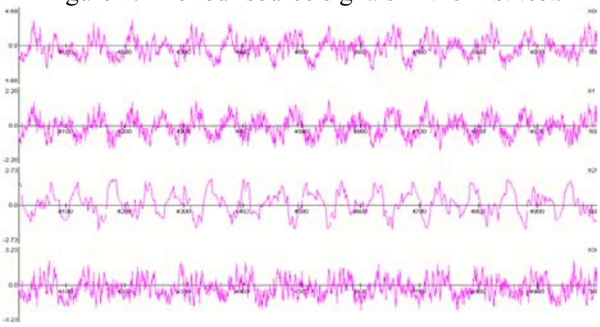

Figure 2. The mixed signals of the source signals in Fig.1 from a randomly generated mixing matrix.

In the second simulation, we solve the mixing problem using the self-adaptive learning rate proposed by Murata et al. [10]. In this case the initial value is not relevant since the learning rate increases and decreases automatically, self-adapting to the environment.

We used a Feed-forward neural network that computes its output according to the Equation (7)

The parameters setting for the simulations includes the neural model and learning rate $\eta(k)$, where three methods are proposed for computing the learning rate.

1. $\eta(k) = \Phi(k)$ where Φ is a defined function; Typical functions are $\Phi(k)$ = constant value. and $\Phi(k) = c_1\exp(-k/c_2)$;
2. $\eta(k)$ According to equ. (7). That is the self-adaptive learning rate proposed in [4].
3. $\eta(k)$ According to equ. (8). That is the self-adaptive learning rate proposed in [5].

As we pointed out before, we can compute the learning rates globally or locally using different update equations:

1. Global for all the neural network (the learning rate is the same $\eta(k)$ for all the synaptic weights w_{ij}) in the form:

$$\mathrm{W}(k+1) = \mathrm{W}(k) + \eta(k)\widetilde{\mathrm{G}}(k) \quad (\mathrm{K} = 1,2,..)$$

2. Global for each neuron, i.e., each neuron has an individual (local) learning rate according to:

$$\eta_i(k) = \Phi(k)$$

$$\eta(k) = (1 - \rho_1)\,\eta_i(k-1) + \rho_1 \beta\, \psi \left[\left\| \hat{G}_{iL}(k), \ldots \hat{G}_{im}(k)^T \right\|\right]$$

$$\eta(k) = \left[(1 - \rho_1\, \eta_i(k-1)\right]\eta_i(k+1) + \rho_1\, \beta\, \eta_1(k+1)\, \psi \left[\left\| \hat{G}_{iL}(k), .. \hat{G}_{im}(k)^T \right\|\right]$$

3. Individual, where each synaptic weight W_{ij} has an local learning rate η_{ij} according to:

$$\eta_{ij}(k) = \Phi(k)$$

$$\eta(k) = (1 - \rho_1)\,\eta_{ij}(k-1) + \rho_1 \beta \psi \left[\left\| \hat{G}_{ij}(k) \right\|\right]$$

$$\eta(k) = \left[(1 - \rho_1\, \eta_{ij}(k-1)\right]\eta_{ij}(k+1) + \rho_1 \beta \eta_{j1}(k+1)\, \psi \left[\left\| \hat{G}_{ij}(k) \right\|\right]$$

In the first simulation we use the Global (scalar) learning rate and the first method for compute the learning rate where used with $\eta(k)$ = 0,0025, while in the second simulation we use the individual learning rate $\eta(k)$ = 0,0025.According to the equation (5) the choosing of the instantaneous gradient (feed-forward case) it was made as:

$$\widetilde{G}(k) = \left[\Lambda(k) - \underline{f(y(k))g^T(y(k))} + W(k)\right]$$

Since online algorithms for BSS & ICA usually perform a stochastic gradient descent based on equation (5), the algorithm computes an average gradient $\hat{\mathrm{G}}(k)$, which is necessary for the self -adaptive learning rates of equations (7) and (8). The computation of the average gradient it was computed according to the equation (9):

$$\hat{\mathrm{G}}(k) = (1 - \rho_2)\,\hat{\mathrm{G}}(k-1) + \rho_2 \hat{G}(\kappa)$$

Figure 3 shows the signals separated by the algorithm. The same experiment was repeated six times and for each repetition the mixing matrix and the source signals were renewed. The performance of the algorithm for the separation of source signals was measured by means of the performance index (Equation (18)). Results are showed in Figure 4. It is clear that for data sets with a number greater than 2000 samples, the performance index is not much better, and then the training time increases. The variance of the performance index is comparable, so we must conclude, that the actual performance is about 0.025 for training sets longer than 2500. The shortest data set is about 2000 or 2500 samples, where PI is over 0.5. The conclusion of this test is that we must take data sets over 2000 samples to get a good separation of the sources using a reasonable training time.

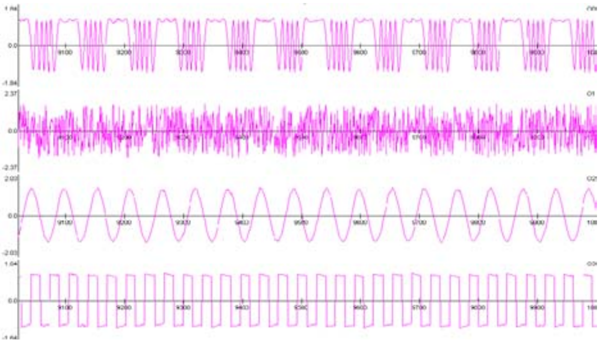

Figure 3. The recovered signals by the algorithm from the mixed signals in figure 2.

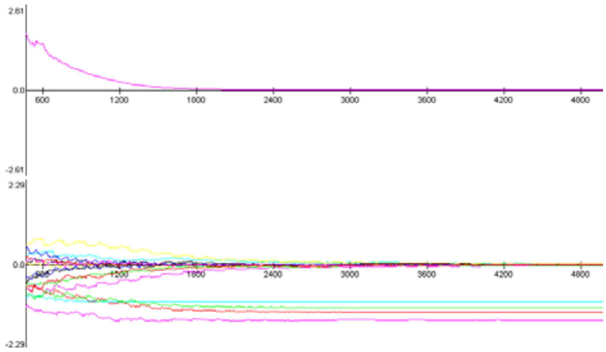

Figure 4. (Above) Performance of the algorithm for the source separation according to Equation 10; (below) Components of the performance matrix.

The second test was executed using the MATLAB codes for the fastICA algorithm that was downloaded from the homepages provided in [6]. We made one single test to evaluate the behaviour of the FastICA algorithm (i.e. the fixed-point algorithm developed by Hyvarinen).

The goal of this experiment was to get some idea about how good the separation can be under the presence of different ions in the sample solution. The experiment consist in simulate the Ca^{2+} ISFET (or ChemFET) sensor array introducing to the algorithm four different signals. The first one represent the Ca^{2+} concentration (based on the CHEMFET Ca^{2+} characteristics), the second one represent the NH_4^+ concentration (based on the CHEMFET Ca^{2+} characteristics), and two additional signals (sin and saw-tooth) that represent disturbing ions. Four sensors compose the array and the goal is to separate and to recover the four different source signals. Since the FastICA is a batch algorithm, we made experiments to show the behaviour when data blocks are overlapping. The input data length is 200, i.e. the four basic signals including two estimated samples signals obtained by means a linear relation. All signals were adjusted to have a zero mean and unit variance. Data were mixed via random mixing matrix, and after execute the FastICA algorithm we obtained the separated signals that are shown on Fig 6. The performance index plot for the FastICA algorithm is shown in figure 7.

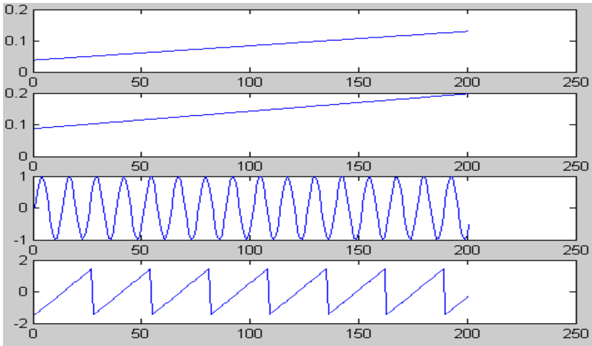

Figure 5. The four source signals in the second test (include two estimated samples signals that represent the response of the ISFET sensor to a particular ion concentration according to [8]).

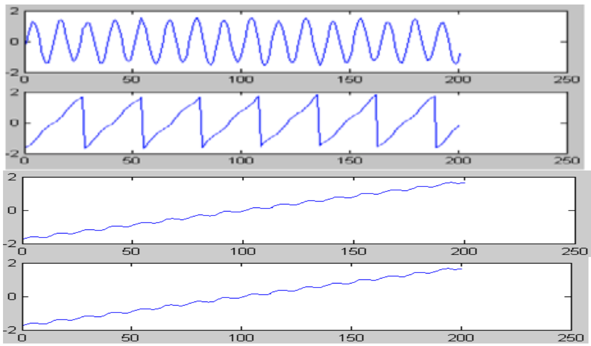

Figure 6. The recovered signals using fastICA.

The CHEMFETs curves were obtained taking samples from the curves presented in [12] between the intervals of values that correspond to the linear region. The samples were generated by means a linear spacing relation. FastICA was applied to data blocks of length from 1 to 200.

The following test explores the dependence of the performance on problem size. The number of sources (N) number is increased one by one from 100 to 10000. Table I list average performance measures based on Equation 18 computed over the different experiments for each of the two algorithms. To evaluate the performance measure we apply to the fastICA algorithm and OOLABSS the same four source signals used in the first test, see fig 1.

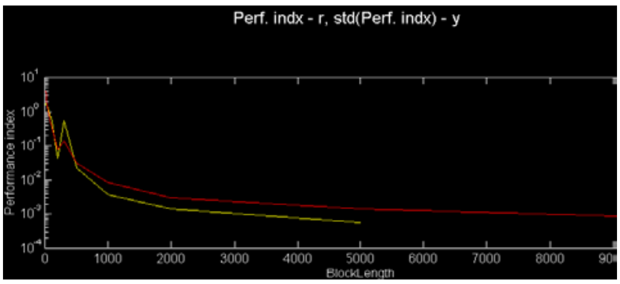

Figure 7. Performance index curve for the separated signals using fastICA.

TABLE I
PERFORMANCE OF THE TWO ALGORITHMS FOR THE TESTS

Mean PI	OOBSSLAB	FASTICA
N = 100	5.23	4,3
N = 200	4.56	3,26
N = 400	1.36	1,28
N = 800	1.02	0,5
N = 1000	0.81	0,19
N = 1500	0.16	0,13
N = 2000	0.1	0,05
N = 3000	0.078	0,039
N = 4000	0.06	0,0267
N = 5000	0.035	0,0203
N = 6000	0.022	0,0178
N = 7000	0.012	0,0134
N = 8000	0.0084	0.0094
N = 9000	0.0063	0.0047
N = 10000	0.0037	0.0026

CONCLUSIONS

ICA has been proposed as a solution to the blind source separation problem in which sensory signals are given those are mixtures of unknown source signals.
Our work was focused on the principles of BSS algorithms and the simulation of the source signals separation in an array of ISFET sensors. BSS and ICA are statistical methods that can be applied to a wide range of applications in neural computing and signal processing like source separation.

As our experiments have shown, a source signal separation of outputs of an ISFET sensor to extract ion concentration can be achieved using BSS and ICA algorithms. We have employed known signals and theoretical estimations of the sensors output signals to achieve source separation by means of the OOLABSS software and the FastICA algorithm. We represent the sensor array input sources by means of four signals, while the mixing matrix emulates the sensor mixed signals that are applied to the algorithms.

The OOLABSS and FastICA performance was compared based on the training set size. It was demonstrated that is not necessary to use more than 2000 samples in order to obtain a good performance of the fastICA algorithm, and we estimate that 500 samples could be a convenient training set size to solve the problem. For the OOLABSS the number of samples has to be equal to 2500 for good estimations.

In spite of the good results achieved in the simulations, some inherent limitations of BSS and ICA must be considered. Firstly, the computation of the demixing matrix has the indeterminacy of a scale factor [2]. Therefore, the waveforms of the source signals are estimated up to a scale factor (see Fig 6), which implies that a calibration step is required for each estimated source after separation. On the other hand, when ICA is used for blind source separation of stochastic processes, time delays may arise due to the different propagation time of signals from the physical sources to the sensors. Both problems would be addressed in a real application of BSS and ICA techniques to the SEWING water monitoring system based on ISFETs.

The standard ICA formulation has several restrictions such as the number of sensors has to be equal or greater than the number of sources. There are not additive noise signals in the sensors and the sources are modelled as random variables. When applying ICA to real-world problems it becomes evident that those restrictions are a problem and although the ICA algorithms give good approximate solutions there is need to develop new algorithms that can relax some of the assumptions.

ACKNOWLEDGEMENTS

This work was supported by the IST Programme No. 2000-28084 (SEWING) of the EU.

THE AUTHORS

Guillermo Bedoya and Dr. Sergio Bermejo are with the Department of Electronic Engineering, Universitat Politècnica de Catalunya, C4 building, Jordi Girona 1-3, 08034 Barcelona.
E-mail: {bedoya, sbermejo}@eel.upc.es

REFERENCES

[1] S. Bermejo. "Blind Source Separation For An Array Of Isfet Sensors: I. Mathematical Foundations"
[2] J.-F.Cardoso, "Blind Signal Separation: Statistical Principles", Proceedings of the IEEE, Vol. 86, No. 10, 1998, pp. 2009-2025
[3] A. Hyvärinen, "Survey on Independent Component Analysis", Neural Computation Surveys, 1999, Available electronically in http://www.cis.hut.fi/_aapo/
[4] A. Hyvärinen, E. Oja, "Independent Component Analysis: A Tutorial", International Joint Conference on Neural Networks, ijcnn99, 1999, Available electronically in: http://www.cis.hut.fi/projects/ica/
[5] A. Benveniste, M. Métivier, P. Priouret, "Adaptive Algorithms and Stochastic Approximations", Berlin: Springer-Verlag, 1990.
[6] A. Hyvärinen and E. Oja, " Fast and robust fixed-point algorithms for independent component analysis" *IEEE trans. Neural Networks,* vol. 10, pp.626-634, may 1999.
[7] B.Orsier and A. Cichocki. "Object-oriented Laboratory for blind separation of sources and independent component analysis (OOLABSS) user manual v1.0" Brain-style information systems group. Technical report, nov. 1997. http://www.open.brain.riken.go.jp/OOLABSS/Welcome.html
[8] S. Amari , A. Cichocki and H.H Yang. " A new learning algorithm for blind source separation," *Advances Neural Inform. Processing Sys.* vol. 8, pp. 757-763, 1996.
[9] A. Cichocki, S. Amari, M Adachi, and W Kasprzak. "Self adaptive neural networks for blind separation of sources". In 1996 *IEEE international Symposium on circuits and systems, ISCAS'96*, volume 2, pages 157-161, Atlanta USA, may 1996. IEEE.
[10] N. Murata, K. Mueller, A. Ziehle, and S-I. Amari. "Adaptive on-line learning in changing environments". In *NIPS-96*, volume 9. MIT Press, 1997.
[11] A. Cichocki, B. Orsier, A.D. Back and S. Amari "On-line adaptive algorithms in nonstationary environments using a modified conjugate gradient approach" in *Proc. Of IEEE Workshop on neural Networks for signal processing,* pages 316 – 325, 1997.
[12] A. Napieralski, W. Wojciak, D. Marcin, M. Szermer, "Library of Chemfet for the eldo environment". Technical report for the System for European Water Monitoring project., Lodz November 2001.
[13] A. Napieralski, W. Wojciak, D. Marcin, M. Szermer, "measurements of pH-ISFET sensors". Technical report for the System for European Water Monitoring project., Lodz November 2001.