A GRADIENT BASED ALGORITHM FOR BLIND INVERSION OF WIENER SYSTEM USING MULTI-VARIATE SCORE FUNCTIONS

Massoud BABAIE-ZADEH[1,2], *Jordi SOLÉ-CASALS*[1,3], *Christian JUTTEN*[1]

[1]Institut National Polytechnique de Grenoble (INPG), Laboratoire des images et des signaux (LIS), Grenoble, France

[2]Electrical engineering department, Sharif University of Technology, Teheran, Iran

[3]Signal Processing Group, Universitat de Vic, Sagrada Família 7, 08500, Vic, Spain

mbzadeh@yahoo.com, jordi.sole@uvic.es, Christian.Jutten@inpg.fr

ABSTRACT

A Wiener system is a linear time-invariant filter, followed by an invertible nonlinear distortion. Assuming that the input signal is an independent and identically distributed (iid) sequence, we propose an algorithm for estimating the input signal only by observing the output of the Wiener system. The algorithm is based on minimizing the mutual information of the output samples, by means of a steepest descent gradient approach.

1. INTRODUCTION

When linear models fail, nonlinear models are powerful tools for modeling practical situations. Many researches have been done in the identification and/or the inversion of nonlinear systems. These works usually assume that both the input and the output of the distortion are available, and are based on higher-order input/output cross-correlation [1] or on the application of the Bussgang and Prices theorems [2, 3] for nonlinear systems with Gaussian inputs. However, in a real world situation, one often does not have access to the distortion input. In this case, the blind identification of the nonlinearity becomes the only way to solve the problem. This paper is concerned by a particular class of nonlinear systems, composed by a linear subsystem followed by a memoryless nonlinear distortion (see Fig. 1). This class of nonlinear systems, also known as Wiener systems, is a nice and mathematically attracting model, but also an actual model used in various areas, such as biology [4], industry [5], sociology and psychology (see also [6] and the references therein). Despite its interest, at our knowledge, it only exist two completely blind procedures for inverting such systems [7, 8]. As in these two mentioned procedures, the basic idea of the method is based on source separation techniques. It consists in changing the spatial independence of the outputs - required for inverting nonlinear mixtures - into a time independence of the output - required for inverting the filtered observation, *i.e.* the blind inversion of Wiener system.

This work has been partially funded by the European project Blind Source Separation and applications (BLISS, IST 1999-13077), by the French project Statistiques Avancées et Signal (SASI), by the Direcció General de Recerca de la Generalitat de Catalunya under a grant for Integrated Actions ACI2001, and by the Universitat de Vic under the grant R0912.

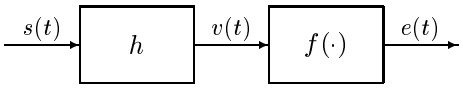

Fig. 1. A Wiener system consists of a linear filter followed by a nonlinear distortion.

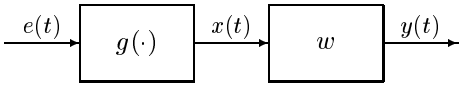

Fig. 2. A Hammerstein system consists of a nonlinear distortion followed by a linear filter.

2. PRELIMINARY ISSUES

2.1. Mutual Information

For designing a system which generates an output with independent samples, we need a criterion for measuring the independence of different samples. A convenient independence measure is mutual information of y_i's, denoted by $I(\mathbf{y})$:

$$I(\mathbf{y}) = \int_{\mathbf{y}} p_{\mathbf{y}}(\mathbf{y}) \ln \frac{p_{\mathbf{y}}(\mathbf{y})}{\prod_{i=1}^{N} p_{y_i}(y_i)} \mathbf{dy} \qquad (1)$$

It is well-known that this quantity is always non-negative, and vanishes if and only if the y_i's are independent. Consequently, the parameters of the inverse

system (the function g and the coefficients of the inverse filter w, see Fig. 2) can be found based on minimization of the mutual information of the output samples.

To do this minimization, knowing an expression for the "gradient" of the mutual information is helpful. Such an expression, which has been already proposed [9], requires multivariate score functions.

2.2. Multivariate Score Functions

In this section, we recall the definitions of multivariate score functions [10]. For the scalar case, we know the following definition from the statistics literature:

Definition 1 (Score Function) *The score function of a scalar random variable y is the opposite of the log derivative of its density:*

$$\psi_y(y) = -\frac{d}{dy}\ln p_y(y) = -\frac{p'_y(y)}{p_y(y)} \qquad (2)$$

where $p_y(y)$ denotes the probability density function (PDF) of y.

Similarly, for a random vector $\mathbf{y} = (y_1, \ldots, y_N)^T$ we define two different kinds of score functions. Let $p_{\mathbf{y}}(\mathbf{y})$ and $p_{y_i}(y_i)$ denote the joint and marginal PDFs, respectively.

Definition 2 (MSF) *The marginal score function (MSF) of $\mathbf{y}$ is the vector whose the i-th component is the score function of the i-th random variable, i.e. :*

$$\boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) = (\psi_1(y_1), \ldots, \psi_N(y_N))^T \qquad (3)$$

where:

$$\psi_i(y_i) = -\frac{d}{dy_i}\ln p_{y_i}(y_i) = -\frac{p'_{y_i}(y_i)}{p_{y_i}(y_i)}. \qquad (4)$$

Definition 3 (JSF) *The joint score function (JSF) of $\mathbf{y}$ is the gradient of $-\ln p_{\mathbf{y}}(\mathbf{y})$, i.e. :*

$$\boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) = (\varphi_1(\mathbf{y}), \ldots, \varphi_N(\mathbf{y}))^T \qquad (5)$$

where:

$$\varphi_i(\mathbf{y}) = -\frac{\partial}{\partial y_i}\ln p_{\mathbf{y}}(\mathbf{y}) = -\frac{\frac{\partial}{\partial y_i}p_{\mathbf{y}}(\mathbf{y})}{p_{\mathbf{y}}(\mathbf{y})} \qquad (6)$$

The difference of these two score functions contains information about the independence of the components of $\mathbf{y}$, and hence it is convenient to define it.

Definition 4 (SFD) *The score function difference (SFD) of $\mathbf{y}$ is the difference between its MSF and JSF:*

$$\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y}) = \boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) - \boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) \qquad (7)$$

The following theorem relates the independence of the components of a random vector $\mathbf{y}$ to its SFD [11].

Theorem 1 *The components of the random vector $\mathbf{y}$ are independent, if and only if, its SFD is zero,* i.e.*:*

$$\boldsymbol{\varphi}_{\mathbf{y}}(\mathbf{y}) = \boldsymbol{\psi}_{\mathbf{y}}(\mathbf{y}) \qquad (8)$$

2.3. "Gradient" of mutual information

The variations of mutual information resulting from a small deviation in its argument (the "differential" of mutual information), is given by the following theorem [9]:

Theorem 2 *Let $\boldsymbol{\Delta}$ be a 'small' random vector, with the same dimension than the random vector $\mathbf{y}$. Then:*

$$I(\mathbf{y}+\boldsymbol{\Delta}) - I(\mathbf{y}) = E\left\{\boldsymbol{\Delta}^T\boldsymbol{\beta}_{\mathbf{y}}(\mathbf{y})\right\} + o(\boldsymbol{\Delta}) \qquad (9)$$

where $o(\boldsymbol{\Delta})$ denotes terms in $\boldsymbol{\Delta}$ of order higher than 1.

Note that for any multivariate differentiable function $f(\mathbf{y})$, we have:

$$f(\mathbf{y}+\boldsymbol{\Delta}) - f(\mathbf{y}) = \boldsymbol{\Delta}^T\nabla f(\mathbf{y}) + o(\boldsymbol{\Delta}) \qquad (10)$$

A comparison between (9) and (10) shows that SFD can be called the *stochastic gradient* of the mutual information.

3. INVERSION CRITERION

From the previous section, the general idea for determining the inverse system is to take the mutual information of the output samples as the inversion criterion, and then to use a gradient based algorithm for minimizing it. This gradient algorithm is based on the "gradient" of mutual information as proposed by (9).

However, using $I(y(0), y(1), y(2), \ldots)$ as independence criterion, is computationally too expensive. This is due to the fact that SFD (which is the gradient of mutual information) requires the estimation of multivariate densities, and the computational load of this estimation increases when the number of samples $y(i)$ increases. Practically, we approximate the independence of the whole sample sequence with independence of sample pairs:

$$J = \sum_{m=1}^{p} I(y(n), y(n-m)) \qquad (11)$$

where p denotes the degree of the separating filter w.

This criterion (11), although requiring only the estimation of bivariate PDFs, is still expensive. For implementing it, we use a stochastic manner, that is, we take $I(y(n), y(n-m))$ as the inversion criterion but at each iteration we use a different random m between 1 and p. With this trick, on the average, we are minimizing the criterion (11) but with much less computation (note that the information in different terms of (11) are not totally independent, and hence it can be intuitively seen that this trick does not highly affect the number of required iterations for convergence). The same idea is used in [8] for blind inverting Wiener systems, and a similar method is used in [11] for blind separating convolutive mixtures.

4. ESTIMATING EQUATIONS

The gradient based approach for minimizing the criterion (11) is:

$$w_k \leftarrow w_k - \mu_1 \frac{\partial J}{\partial w_k} \quad k = 0, \ldots, p \tag{12}$$

$$g \leftarrow g - \mu_2 \frac{\partial J}{\partial g} \tag{13}$$

where w_k are the coefficients of $W(z)$, the filter associated to w in the discrete time representation). Note that $\frac{\partial J}{\partial g}$ is itself a function. For using the above approach, the gradients of $I(y(n), y(n-m))$ with respect to w_k and g must be calculated. For obtaining these gradients, the idea is to let a small deviation in the desired parameter, and then to compute the influence of this deviation in $I(y(n), y(n-m))$ by using (9), which gives the desired gradient.

Before proceeding, let define a notation. For any signal $x(n)$, by $\mathbf{x}^{(m)}(n)$ we denote the vector signal:

$$\mathbf{x}^{(m)}(n) \triangleq \begin{pmatrix} x(n) \\ x(n-m) \end{pmatrix} \tag{14}$$

With this notation, we can now derive the gradients of $I(\mathbf{y}^{(m)}(n))$ with respect to w_k's and g.

4.1. Gradient with respect to w_k

For calculating the gradient of $I(\mathbf{y}^{(m)}(n))$ with respect to w_k, assume a small deviation ϵ in this parameter:

$$\hat{w}_l = \begin{cases} w_l + \epsilon, & l = k \\ w_l, & l \neq k \end{cases} \quad , l = 1, \ldots, p \tag{15}$$

Then the new output will be:

$$\hat{y}(n) = \sum_{l=0}^{p} \hat{w}_l x(n-l) = y(n) + \epsilon\, x(n-k) \tag{16}$$

Consequently:

$$\hat{\mathbf{y}}^{(m)}(n) = \mathbf{y}^{(m)}(n) + \epsilon\, \mathbf{x}^{(m)}(n-k) \tag{17}$$

Hence from (9) we have (up to first order terms):

$$\begin{aligned} &I\left(\hat{\mathbf{y}}^{(m)}(n)\right) - I\left(\mathbf{y}^{(m)}(n)\right) = \\ &= E\left\{\boldsymbol{\beta}_m^\star(n)^T \epsilon\, \mathbf{x}^{(m)}(n-k)\right\} \end{aligned} \tag{18}$$

where $\boldsymbol{\beta}_m^\star(n)$ stands for the SFD of $\mathbf{y}^{(m)}(n)$. This equality can be simplified as:

$$\begin{aligned} \Delta I &= \epsilon\, E\left\{\boldsymbol{\beta}_m^\star(n)^T \mathbf{x}^{(m)}(n-k)\right\} \\ &= \epsilon\, E\left\{\beta_{m,1}^\star(n) x(n-k)\right\} \\ &\quad + \epsilon\, E\left\{\beta_{m,2}^\star(n) x(n-k-m)\right\} \\ &= \epsilon\, E\left\{\beta_{m,1}^\star(n) x(n-k)\right\} \\ &\quad + \epsilon\, E\left\{\beta_{m,2}^\star(n+m) x(n-k)\right\} \\ &= \epsilon\, E\left\{\beta_m(n) x(n-k)\right\} \end{aligned} \tag{19}$$

where:

$$\beta_m(n) \triangleq \beta_{m,1}^\star(n) + \beta_{m,2}^\star(n+m) \tag{20}$$

Note that in the simplification (19) the signals are assumed to be stationary. Finally, from (19) we deduce:

$$\frac{\partial}{\partial w_k} I\left(\mathbf{y}^{(m)}(n)\right) = E\left\{\beta_m(n)\, x(n-k)\right\} \tag{21}$$

In other words, the derivatives of $I(y(n), y(n-m))$ with respect to the coefficients of the filter $W(z)$ are obtained by the cross-correlation coefficients between $x(n)$ and $\beta_m(n)$. The procedure of estimation of $\beta_m(n)$ from $x(n)$ can be depicted as:

$$x(n) \rightarrow \begin{bmatrix} x(n) \\ x(n) \end{bmatrix} \xrightarrow{\text{Shift}} \begin{bmatrix} x(n) \\ x(n-m) \end{bmatrix} \xrightarrow{\text{SFD}}$$

$$\begin{bmatrix} \beta_{m,1}^\star(n) \\ \beta_{m,2}^\star(n) \end{bmatrix} \xrightarrow{\text{Shift back}} \begin{bmatrix} \beta_{m,1}^\star(n) \\ \beta_{m,2}^\star(n+m) \end{bmatrix} \xrightarrow{\text{Sum}} \beta_m(n)$$

4.2. Gradient with respect to g

Here, the calculation of the 'gradient' of $I(\mathbf{y}^{(m)}(n))$ with respect to the 'function' g is considered. This 'gradient' is itself a function.

To obtain this gradient, let a small deviation in the function g of the form:

$$\hat{g} = g + \epsilon \circ g \tag{22}$$

where $\epsilon(\cdot)$ is a 'small' function. The above equation is equivalent to:

$$\hat{x} = x + \epsilon(x) \tag{23}$$

By defining $\delta \triangleq \epsilon(x)$, the output of $W(z)$ becomes:

$$\hat{y}(n) = y(n) + \big[W(z)\big]\delta(n) \tag{24}$$

Let $\eta(n) \triangleq \big[W(z)\big]\delta(n)$. Then, from the above equation:

$$\hat{\mathbf{y}}^{(m)}(n) = \mathbf{y}^{(m)}(n) + \boldsymbol{\eta}^{(m)}(n) \tag{25}$$

Now, by using (9) we have (up to first order terms):

$$\begin{aligned} \Delta I &= E\left\{\boldsymbol{\beta}_m^\star(n)^T \boldsymbol{\eta}^{(m)}(n)\right\} \\ &= E\left\{\beta_m(n)\, \eta(n)\right\} \end{aligned} \tag{26}$$

where a simplification similar to (19) is used for writing the second equality. Now, we write the above equation as:

$$\begin{aligned} \Delta I &= E\left\{\beta_m(n)\, \eta(n)\right\} \\ &= E\left\{\beta_m(n) \sum_{k=0}^{p} w_k \delta(n-k)\right\} \\ &= \sum_{k=0}^{p} w_k E\left\{\beta_m(n)\delta(n-k)\right\} \\ &= \sum_{k=0}^{p} w_k E\left\{\beta_m(n+k)\delta(n)\right\} \\ &= E\left\{\delta(n)\alpha(n)\right\} \end{aligned} \tag{27}$$

where:

$$\alpha(n) \triangleq \sum_{k=0}^{p} w_k \beta_m(n+k) = \left[W\left(\frac{1}{z}\right)\right] \beta_m(n) \quad (28)$$

Consequently:

$$\begin{aligned} \Delta I &= E\{\delta\alpha\} \\ &= E_x\{E\{\delta\alpha \mid x\}\} \\ &= E_x\{E\{\epsilon(x)\alpha \mid x\}\} \\ &= E_x\{\epsilon(x)E\{\alpha \mid x\}\} \\ &= \int_{-\infty}^{+\infty} \epsilon(t)E\{\alpha \mid x = t\}\, p_x(t)\, dt \end{aligned} \quad (29)$$

The above equation shows that the (relative [12]) 'gradient' of $I(y(n), y(n-m))$ with respect to the function g via the weighting function $p_x(x)$ is the function:

$$(\nabla_g I)(\cdot) = E\{\alpha \mid x = \cdot\} \quad (30)$$

which is the regression function [13] from x to α. In other words, (29) shows that taking $\epsilon(\cdot)$ equal to the opposite of the above function, insures a reduction in $I(y(n), y(n-m))$, *i.e.* $\Delta I < 0$.

In our simulations, (30) is estimated using smoothing splines [14], *i.e.* $(\nabla_g I)(\cdot)$ is the cubic spline which fits on the (x, α) data points. In MATLAB's spline toolbox, it is computed with the '`csaps`' function.

5. THE ALGORITHM

Having calculated the gradients of the inversion criterion with respect to the parameters of the inverting system (w_k's and g), the final algorithm is nothing but applying a steepest descent gradient algorithm on these parameters.

However, some other points must be taken into account. First, it must be noted that there are mean and scale indeterminacies in $x(n)$. Consequently, for removing their effects, at each iteration, the mean of $x(n)$ is removed and its energy is normalized. Another indeterminacy is the energy of $y(n)$, and it is removed by normalizing its energy at each iteration.

Another important issue is the initialization of the algorithm. As the starting point of the algorithm, we use the approach proposed in [15]. The idea of this approach is as follows:

1. Because of the central limit theorem [13], the output $v(t)$ of the filter h is a weighted sum of iid samples tends to have a Gaussian distribution, which is then distorted by the nonlinear function f. As a starting estimate of g, we use the nonlinear function which enforces x to be similar to v, *i.e.* a Gaussian. It can be seen [15] that this is achieved with:

$$g = \Phi^{-1} \circ F_e \quad (31)$$

where F_e is the Cumulative Density Function (CDF) of e and Φ^{-1} is the CDF of a (zero mean and unit variance) Gaussian distribution.

2. Since we are looking for an output with independent samples, as an initial estimate for the filter $W(z)$ a filter which creates output with *decorrelated* samples may be used. Such a filter is given by the Linear Predictor Coefficients (LPC) of the x sequence. In MATLAB it can be obtained by the `lpc` function.

The final inversion algorithm is given in Fig. 3. In this figure, p denotes the order of the inverse filter.

- Initialization: $g = \Phi^{-1} \circ F_e$, $x = g(e)$, $W(z) = lpc(x, p)$, $y(n) = [W(z)]x(n)$.
- Loop:
  1. Choose a random $1 \leq m \leq p$.
  2. Estimate the SFD of $\mathbf{y}^{(m)}$ and $\beta_m(n)$ (see (14) and (20)).
  3. Estimate $\frac{\partial I}{\partial w_k}$ and $(\nabla_g I)(\cdot)$ from (21) and (30), respectively.
  4. Let:
$$w_k \leftarrow w_k - \mu_1 \frac{\partial J}{\partial w_k} \quad , k = 0, \ldots, p$$
$$x \leftarrow x - \mu_2 (\nabla_g I)(x)$$
  5. Remove the mean of x and normalize its energy.
  6. Let $y(n) = [W(z)]\, x(n)$.
  7. Normalize the energy of y: (a) Let $\sigma_y = E\{y\}$, (b) $y \leftarrow y/\sigma_y$, (c) $w_k \leftarrow w_k/\sigma_y, k = 0, \ldots, p$.
- Repeat until convergence

Fig. 3. Gradient based algorithm for blind inversion of Wiener systems.

6. EXPERIMENTAL RESULTS

Here, for checking the efficacy of the proposed algorithm, we present an experimental result using uniform random sources and a saturating nonlinear distortion.

In this experiment, the source signal is a uniform random source with zero mean and unit variance. The filter h is the low-pass filter $H(z) = 1 + 0.5z^{-1} - 0.2z^{-2}$, and the nonlinear distortion function is $f(x) = \tanh(3x)$. Then, the algorithm of Fig. 3 is used to obtain the inverse system. The parameters of the algorithm are: $T = 1000$ (number of observed samples), $p = 15$ (order of then inverse filter), $\mu_1 = \mu_2 = 0.2$ (step sizes). For estimating the SFD, a method proposed by D.-T. Pham for estimating conditional score functions is used [16].

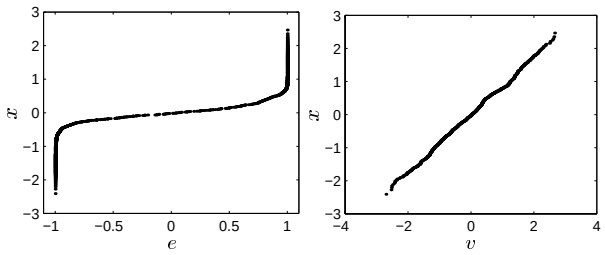

Fig. 4. Left) x versus e, Right) x versus v.

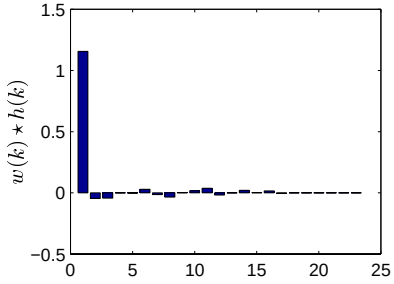

Fig. 5. The coefficients of the global filter $w(k) \star h(k)$.

As performance criterion, we have used the output Signal to Noise Ratio (SNR), defined by:

$$\text{SNR (in dB)} = 10 \log_{10} \frac{E\left\{s^2\right\}}{E\left\{(y-s)^2\right\}} \qquad (32)$$

The averaged output SNR obtained in this experiment is between 18 and 20 dB .

The distribution of x samples versus e and v samples is shown in Fig. 4. These distributions show the estimated g and the compensated function $g \circ f$. Then the efficacy of the algorithm in compensating the nonlinear distortion is demonstrate.

Finally, Figure 5 shows the coefficients of the filter $W(z)H(z)$. The result is almost a Dirac function: it then indicates that the inverse of the linear part is well estimated.

7. REFERENCES

[1] S.A. Bellings and S.Y. Fakhouri, "Identification of a class of nonlinear systems using correlation analysis," *Proc. IEEE*, vol. 66, pp. 691–697, 1978.

[2] E.D. Boer, "Cross-correlation function of a bandpass nonlinear network," *Proc. IEEE*, vol. 64, pp. 1443–1444, 1976.

[3] Jacoviti G., A. Neri, and R. Cusani, "Methods for estimating the autocorrelation function of complex stationary process," *IEEE Trans. ASSP*, 1987.

[4] I.W. Hunter, "Frog muscle fibber dynamic stiffness determined using nonlinear system identification techniques," *Biophys. J.*, pp. 49–81, 1985.

[5] R. Bars, I. Bzi, B. Pilipar, and B. Ojhelyi, "Nonlinear and long range control of a distillation pilot plant. in identification and syst. parameter estimation," in *EUSIPCO*, Budapest, 1990, pp. 848–853.

[6] I.W. Hunter and M.J. Korenberg Korenberg, "The identification of nonlinear biological systems: Wiener and hammerstein cascade models," *Biol Cybern.*, pp. 135–144, 1985.

[7] A. Taleb, J. Solé-Casals, and C. Jutten, "Quasi-nonparametric blind inversion of Wiener systems," *IEEE Trans. on Signal Processing*, vol. 49, no. 5, pp. 917–924, 2001.

[8] M. Babaie-Zadeh, J. Solé-Casals, and C. Jutten, "Blind inversion of wiener system using a minimization-projection (MP) approach," in *ICA2003*, Nara, Japan, April 2003, pp. 681–686.

[9] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Differential of mutual information function," *IEEE Signal Processing Letters*, 2003, accepted.

[10] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Blind separating Convolutive Post-Nonlinear mixtures," in *ICA2001*, San Diego, California, December 2001, pp. 138–143.

[11] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Separating convolutive mixtures by mutual information minimization," in *Proceedings of IWANN'2001*, Granada, Spain, Juin 2001, pp. 834–842.

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

[13] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1991.

[14] R. L. Eubank, *Spline smoothing and nonparanetric regression*, Dekker, 1988.

[15] J. Solé-Casals, M. Babaie-Zadeh, C. Jutten, and D.-T. Pham, "Improving algorithm speed in post nonlinear mixtures and wiener systems inversion," in *ICA2003*, Nara, Japan, April 2003, pp. 639–644.

[16] D. T. Pham, "Fas algorithm for estimating mutual information, entropies ans score functions," in *Proceedings of ICA2003*, Nara, Japan, April 2003, pp. 17–22.