A Fast Fixed-Point Neural Blind Deconvolution Algorithm

Simone Fiori

Abstract— **The aim of the present Letter is to introduce a new blind deconvolution algorithm based on fixed-point optimization of a 'Bussgang'-type cost function. The cost function relies on approximate Bayesian estimation achieved by an adaptive neuron. The main feature of the presented algorithm is fast convergence that guarantees good deconvolution performances with limited computational demand compared to algorithms of the same class.**

Keywords— **'Bussgang'-type blind deconvolution; Neural Bayesian estimation; Fixed-point iteration.**

I. Introduction

Blind deconvolution [7] aims at recovering a source signal distorted by the linear medium that it propagates within. Its most known engineering applications are the equalization of communication channels [2], the analysis of geophysical measurements [15], the restoration of images from their blurred versions [11] as well as the enhancement of storage and retrieval functions in opto-magnetic memory supports [5].

The most known blind-deconvolution technique is perhaps the 'Bussgang' one, which relies on the iterative Bayesian estimation of the source sequence, where the Bayesian estimator is matched to source statistics and to the model of the filter output signal. Some modified 'Bussgang' algorithms (MBAs), based on neural-type approximate Bayesian estimators, have recently been proposed by the present author in the contributions [8], [9].

The aim of the present Letter is to illustrate a new fixed-point version of the MBAs. Fixed-point algorithms in neural learning theory have recently received considerable attention due to their properties of low computational requirement and fast iteration convergence [4], [12]. The proposed algorithm appears as a batch version of the MBAs and is characterized by good deconvolution performances corresponding to low computational burden and fast adaptation. To the best of our knowledge, a fixed-point algorithm has never been used before in a neural/Bayesian blind deconvolution context.

II. Blind Deconvolution by Neural Bayesian Estimation

The linear system to deconvolve is described by the following input/output model:

$$u_t = \mathbf{h}^T \mathbf{s}_t + \mathcal{N}_t \ , \tag{1}$$

where $\mathbf{s}_t$ is the system's input vector-stream at time t, namely $\mathbf{s}_t \stackrel{\text{def}}{=} [s_t \ s_{t-1} \ s_{t-2} \ \ldots \ s_{t-L_h+1}]^T$, s_t denotes the source signal and $\mathcal{N}_t$ is a zero-mean white measurement disturbance independent of the source signal. The variable L_h denotes the length of the system impulse response $\mathbf{h}$. The following minimal hypotheses about the system and data stream may be considered [10]: The system's impulse response satisfies $\mathbf{h}^T\mathbf{h} = 1$ and its inverse has finite energy; the system is time-invariant or slowly time-varying; the source signal s_t is a stationary, ergodic, independent identically distributed (IID) random process with mean $\mathbb{E}_s[s_t] = 0$ and variance $\mathbb{E}_s[s_t^2] = 1$; also, if $p_s(s)$ denotes the probability density function of the input signal, it is supposed that $p_s(s) = p_s(-s)$ and that it is non-Gaussian.

The author is with the Faculty of Engineering of the Perugia University, Terni Campus, Loc. Pentima bassa, 21, I-05100 Terni (Italy). Email: sfr@unipg.it (S. Fiori).

A filter described by the vector impulse response $\mathbf{w}$ is the inverse of system (1) if $\mathbf{w}$ approximately cancels the effects of $\mathbf{h}$ on the source signal. Denoting with $\mathbf{u}_t$ the vector containing the filter input samples $\mathbf{u}_t \stackrel{\text{def}}{=} [u_t \ u_{t-1} \ u_{t-2} \ \ldots \ u_{t-L_w+1}]^T$, where L_w is the number of tap-weights in $\mathbf{w}$, the output of the filter writes:

$$x_t = \mathbf{w}_t^T \mathbf{u}_t \ . \tag{2}$$

Note that, in general, the deconvolution may only be approximate because of the presence of the additive noise affecting the measure and because a finite-impulse-response (FIR) filter cannot represent the inverse of the FIR system (1) (for more details the Reader is addressed to [10]).

Since $\mathbf{h}$ and s_t are unknown, the optimal filter $\mathbf{w}_\star$ such that $x_t \sim s_t$ has to be *blindly* identified possibly by means of a neural algorithm. From the basic theory of blind deconvolution, it is known that the source signal may be recovered up to arbitrary amplitude scaling and time-delay [10]; in the present setting, however, we supposed, without loss of generality, that the source stream power and the system energy are known, thus the amplitude of the recovered source is controlled by the norm of the weight-vector $\mathbf{w}$.

During neural filter learning, the misadjustment of filter's coefficients makes the filter output differ from the source signal. Both misadjustment and FIR-by-FIR inversion difficulty may be properly taken into account through the following filter output signal model:

$$x_t = c s_{t-\delta} + n_t \ , \tag{3}$$

where n_t is the so-called *deconvolution noise*, δ is a finite delay and c denotes instantaneous amplitude distortion. A suitable representation of n_t is a zero-mean, white, Gaussian random process of variance σ^2, incorrelated with the source signal [10]. The model (3) reveals that the relationship between x_t and $cs_{t-\delta}$ is deterministic but for deconvolution noise, therefore, an appropriate estimator $c\hat{s}_{t-\delta} = B(x_t)$ can be designed according to Bayesian estimation theory. On the basis of the available estimator, in [1] the error criterion:

$$C(\mathbf{w}_t) \stackrel{\text{def}}{=} \frac{1}{2}\mathbb{E}_{n_t}[n_t^2] = \frac{1}{2}\mathbb{E}_{x_t}\left[(x_t - B(x_t))^2\right] \tag{4}$$

had been proposed. The minimization of the above cost function may be achieved by a pseudo-LMS stochastic algorithm, as proposed in [1], by means of a stochastic gradient steepest descent algorithm, as in [6], [8], or any suitable known optimization method.

In the case of uniformly distributed source stream, which is of interest e.g. in telecommunication systems, a suitable approximation of the actual Bayesian estimator $B(x)$ is the neural transfer function [10]:

$$\hat{B}(x) = \kappa \tanh(\lambda x) \ , \tag{5}$$

with κ and λ being properly tuned parameters. In order to select suitable values for these parameters, in [8] we proposed to adapt them through time by means of a gradient-based algorithm applied to $C(\kappa, \lambda, \mathbf{w})$ defined as in (4). The obtained iterative procedure was referred to as modified Bussgang algorithm

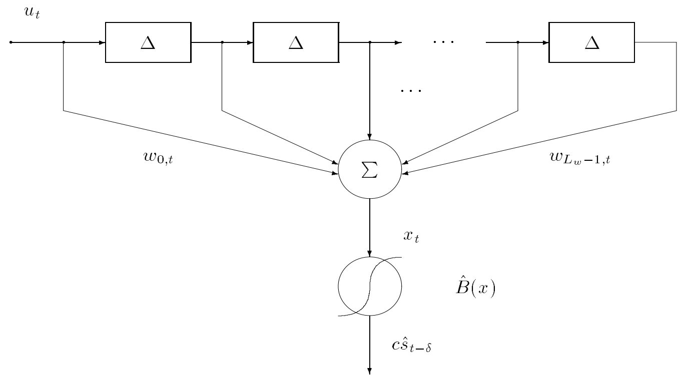

Fig. 1. A neuron with filtering synapses. (Symbol Δ denotes unitary delay, namely, $\Delta u_t = u_{t-1}$, while $w_{i,t}$ denotes the value of the i^{th} filter tap-weight at time t.)

(MBA). In this case, however, the optimization problem is not well-posed, because the minimum of the cost function is attained for $\mathbf{w} = \mathbf{0}$. In order to get rid of this drawback, the automatic gain control (AGC) technique may be employed which aims at keeping constant the energy of the filter impulse response sequence, that means enforcing the constraint $\mathbf{w}^T\mathbf{w} = \omega^2$, with $\omega > 0$ being a fixed constant. It is worth noting that the value of ω does not matter at this stage, because it simply represents a scaling of the recovered source signal that may be easily corrected, if necessary.

In the next section we describe a two-step batch algorithm for cost function (4) minimization based on the pre-optimization of the neural Bayesian estimator and on an efficient fixed-point minimization algorithm. It is worth recalling here that the employed neural structure is closely related to the MLP structures endowed with filtering synapses (a review was given e.g. in [13]), even if its learning paradigm is, by the 'blind' nature of the problem, inherently unsupervised. A sketch of the employed deconvolving neuron is shown in the Figure 1.

III. Fast Fixed-Point Deconvolution Algorithm

In order to derive the proposed new algorithm, we may proceed separately to the optimization of the values of the parameters κ and λ of the neural estimator (5), with respect to the given distribution of the source values, and to the optimization of neural filters' impulse response coefficients. In fact, the optimization of the Bayesian estimator is independent of the particular filter-taps updating algorithm used as well as of the system that warps the source signal. In order to proceed with the derivation of the two above-mentioned steps, we make the technical hypothesis that a noiseless system is dealt with (that is $\mathcal{N}_t = 0$ identically in (1)). The effect of noise on the filter's performance, and thus the robustness of the proposed design method, will be evaluated numerically through experiments in section IV. Also, as mentioned it is assumed that s_t is a white random signal uniformly distributed within $[-\sqrt{3}, +\sqrt{3}]$.

A. Optimization of the neural Bayesian estimator

At theoretic optimality, namely when $x_t = cs_{t-\delta}$, the mean-squared error (4) should attain the minimal possible value. This suggests that a sensible choice for the pair (κ, λ) should guarantee the minimal distortion between the true source signal and its estimate provided by the non-linear neural FIR filter. To this end, we define the residual distortion after deconvolution as $C_R(\kappa, \lambda) \stackrel{\text{def}}{=} C(\kappa, \lambda, \mathbf{w}_\star)$ and study its minimality properties: A pair of values (κ, λ) is looked for which guarantees the best deconvolution performance with respect to the measure $C_R(\kappa, \lambda)$.

Such analysis has been performed numerically: In particular, it was first observed that the minima of the $C_R(\kappa, \lambda)$ lie on a curve $\kappa = \kappa(\lambda) \stackrel{\text{def}}{=} \Lambda/\lambda$, with Λ being a suitable constant (in this case, $\Lambda = 1.185$). The value of λ that guarantees the minimal residual distortion after deconvolution may be found by setting to zero the first derivative of the compound function $C_R(\kappa(\lambda), \lambda)$. A unique non-zero solution exists, that is denoted by $\lambda_\star$, then the corresponding value $\kappa_\star = \kappa(\lambda_\star)$ may be computed, so that the optimization of the Bayesian estimator is completed. It is worth noting that the optimal neural estimator depends only on the probability density function of the source, therefore, as far as the source statistics stays the same, the optimal estimator needs not to be re-calculated.

B. Fixed-point iteration

The cost function $C(\lambda_\star, \kappa_\star, \mathbf{w})$ may be taken as basis for the filter optimization process under the AGC constraint. Namely, we define the Lagrangean:

$$U(\mathbf{w}_t) \stackrel{\text{def}}{=} \frac{1}{2}\mathbb{E}_{x_t}\left[\left(x_t - \hat{B}(x_t)\right)^2\right] + \frac{\mu_t}{2}(\mathbf{w}_t^T\mathbf{w}_t - \omega^2) \ , \tag{6}$$

whose free minimum coincides with the constrained minimum of the cost function (4). In the above expression, μ_t denotes the Lagrange multiplier needed to enforce the AGC constraint. The standard elimination method allows computing the optimal multiplier by solving the scalar equation $\mathbf{w}_t^T \frac{\partial U(\mathbf{w}_t)}{\partial \mathbf{w}_t} = 0$ under the constraint $\mathbf{w}_t^T\mathbf{w}_t = \omega^2$. By defining the scalar quantity $\gamma(x) \stackrel{\text{def}}{=} \frac{1}{2}\frac{d}{dx}(x - \hat{B}(x))^2$, the multiplier elimination equation casts into $\mathbb{E}_{x_t}[x_t\gamma(x_t)] + \mu_t\omega^2 = 0$, so that the gradient of the cost function U with respect to the filter-weight vector may be written as:

$$\frac{\partial U(\mathbf{w}_t)}{\partial \mathbf{w}_t} = \mathbb{E}_{x_t}\left[\gamma(x_t)\mathbf{u}_t\right] - \mathbb{E}_{x_t}\left[\gamma(x_t)x_t\right]\frac{\mathbf{w}_t}{\omega^2} \ . \tag{7}$$

The optimal filter's weight-vector finds by setting the above gradient to zero. Such condition readily leads to the fixed-point iteration:

$$\tilde{\mathbf{w}}_{t+1} = \omega^2 \frac{\mathbb{E}_{u_t}[\gamma(\mathbf{w}_t^T\mathbf{u}_t)\mathbf{u}_t]}{\mathbb{E}_{u_t}[\gamma(\mathbf{w}_t^T\mathbf{u}_t)(\mathbf{w}_t^T\mathbf{u}_t)]} \ , \mathbf{w}_{t+1} = \omega\frac{\tilde{\mathbf{w}}_{t+1}}{\|\tilde{\mathbf{w}}_{t+1}\|} \ , \tag{8}$$

with initial condition $\mathbf{w}_0^T\mathbf{w}_0 = \omega^2$ and with $\|\cdot\|$ denoting the L_2 norm. By plugging the first equation into the second equation, we obtain the (simpler) final expression of the proposed fixed-point algorithm:

$$\mathbf{w}_{t+1} = \omega F(\mathbf{w}_t) \ , \ F(\mathbf{w}_t) \stackrel{\text{def}}{=} \frac{\mathbb{E}_{u_t}[\gamma(\mathbf{w}_t^T\mathbf{u}_t)\mathbf{u}_t]}{\|\mathbb{E}_{u_t}[\gamma(\mathbf{w}_t^T\mathbf{u}_t)\mathbf{u}_t]\|} \ . \tag{9}$$

It deserves to mention that the last expression inherently embodies the necessary AGC.

C. Learning algorithm and implementation details

While the source stream s_t is IID, after passing through the system (1) the samples are no longer temporally statistically independent and, in particular, they possess second-order statistical correlation. Second order correlation is easy to remove by temporal data pre-whitening. This is not a necessary operation but certainly it makes the deconvolution process easier, thus it is worth applying. In particular, temporal pre-whitening may be implemented by computing the covariance matrix of deconvolving filter input vector stream, namely $\mathbf{R}_{\mathbf{uu}} \stackrel{\text{def}}{=} \mathbb{E}_{u_t}[\mathbf{u}_t\mathbf{u}_t^T]$, and by defining the whitened input vector-stream as $\hat{\mathbf{u}}_t \stackrel{\text{def}}{=} \mathbf{R}_{\mathbf{uu}}^{-\frac{1}{2}} \mathbf{u}_t$. It is worth noting that, thanks to the ergodicity of the considered processes, the ensemble averages may be replaced with temporal averages.

The above considerations give rise to the following batch fixed-point algorithm:

- Collect the filter input stream and form the multivariate stream $\mathbf{u}_t$;
- Whiten the multivariate signal $\mathbf{u}_t$ in order to make the input stream have unitary covariance matrix;
- Choose a starting point for the inverse filter impulse response $\mathbf{w}_0$ and compute the inverse filter impulse response $\mathbf{w}_\star$ by the fixed point algorithm (9).

The system deconvolution accuracy may be measured by means of the residual inter-symbol interference (ISI), defined as:

$$ISI_t \stackrel{\text{def}}{=} \frac{\mathbf{T}_t^T \mathbf{T}_t - T_{t,\max}^2}{T_{t,\max}^2} , \qquad (10)$$

where $\mathbf{T}_t$ denotes the convolution between the system's impulse response and the inverse filter's impulse response, and $T_{t,\max}$ denotes the component of $\mathbf{T}_t$ having the maximal absolute value. However, the ISI measure does not account for the additive noise, thus the quality of source estimate should also be measured by the mean-squared estimation error defined as:

$$MSE_t \stackrel{\text{def}}{=} \frac{1}{2} \mathbb{E}_{s_t}[(\xi_t - s_{t-\delta})^2] , \qquad (11)$$

where $\xi_t \stackrel{\text{def}}{=} x_t / \sqrt{\mathbb{E}_{x_t}[x_t^2]}$, in order to adjust the filter output signal power to the source power. It is also worth noting that the total group-delay δ may be estimated as the lag corresponding to the same component of $\mathbf{T}_t$ displaying the maximal absolute value. Note that the ISI and MSE are inherently non-stationary, as they measure the learning progress of the neural filter.

Also, in order to carry out experiments on noisy channels, it is worth defining the signal-to-noise ratio (SNR) as $SNR \stackrel{\text{def}}{=} \mathbb{E}_{s_t}[s_t^2]/\mathbb{E}_{\mathcal{N}_t}[\mathcal{N}_t^2]$. However, the signal power has been assumed equal to 1, thus $SNR_{\text{dB}} = -10 \log_{10}(\mathbb{E}_{\mathcal{N}_t}[\mathcal{N}_t^2])$.

D. *On the convergence of the learning algorithm*

A standard tool for proving the convergence of a fixed-point algorithm is the well-known Banach contraction theorem, ensuring that in a complete metric space a fixed-point algorithm based on a contractive operator surely converges to a steady-state. In this case, the iteration is carried out on a L_w-dimensional hyper-sphere which is a complete metrizable space, while the fixed-point operator $F(\mathbf{w}_t)$ is apparently differentiable, hence Lipschitzian. As a consequence, the scaled fixed-point iteration (9) may be made contractive (at least locally) by selecting a proper value of the damping parameter ω. Then, in the present case the Banach theorem holds and ensures the convergence of the proposed learning rule.

IV. Experimental results

The proposed algorithm is tested to adaptively learn an inverse filter for the sampled (BGR) telephonic channel described in [3] having $L_h = 14$: Its features are illustrated in Figure 2 (the channel impulse response has been normalized so that $\mathbf{h}^T\mathbf{h} = 1$).

In all the following experiments the initial impulse response of the filter, $\mathbf{w}_0$, was assumed as a null sequence, except for the central tap-weight which equals ω. Its length was assumed $L_w = 14$ as the result of validation. Moreover, the value 0.3 was

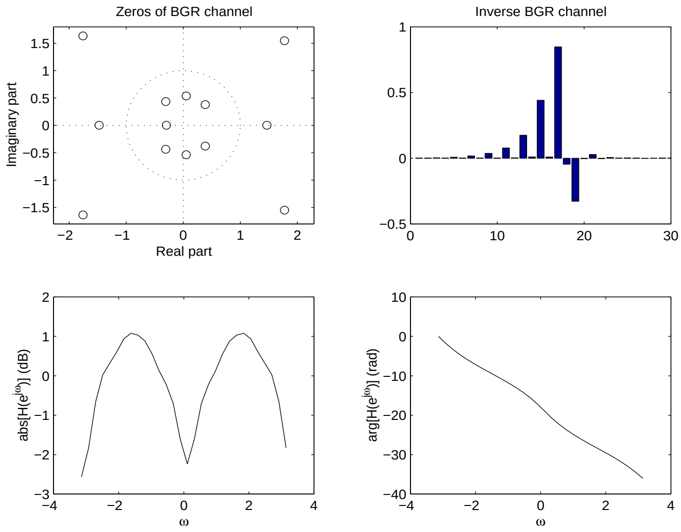

Fig. 2. Sampled telephonic channel. Top: Zero-plot and impulse response bar. Bottom: Amplitude/phase spectra.

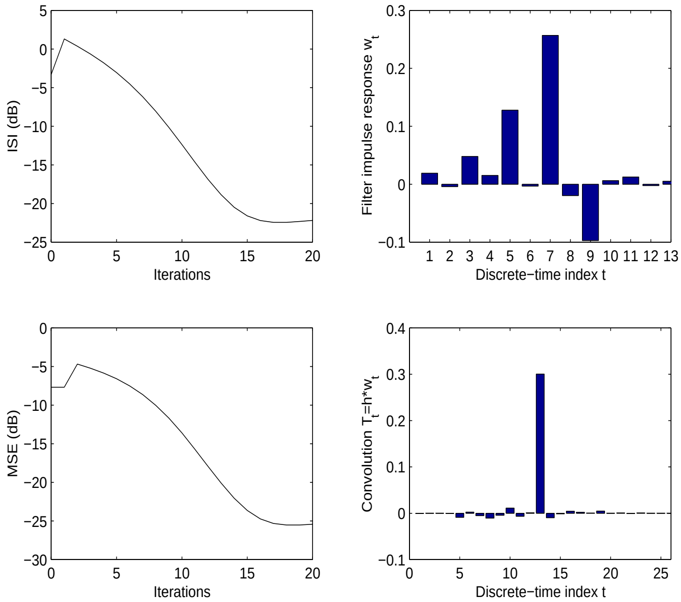

Fig. 3. Test of the fixed-point MBA algorithm over the BGR telephonic channel. Left: ISI and MSE curves vs. fixed-point iteration index. Right: Adapted filter impulse response and its convolution with the BGR channel impulse response bars.

assumed for the constant ω which guarantees the fixed-point iteration algorithm to converge steadily.

The fixed-point algorithm has first been tested with a noiseless BGR telephonic channel, and the results are illustrated in the Figure 3. The final value of the interference residual in this case is about -22.19 dB and the corresponding value of the mean squared error is about -25.41 dB.

In order to measure the performance of the deconvolution algorithm in presence of additive noise, a robustness curve in the SNR-ISI/MSE plane may be estimated through repeated experiments, which represents the interference residual as well as the residual mean squared error after learning corresponding to a given signal-to-noise ratio. It is worth considering, however, that *the Bayesian estimator stays the same that was optimized separately in the zero-noise limit.* The results of fixed-

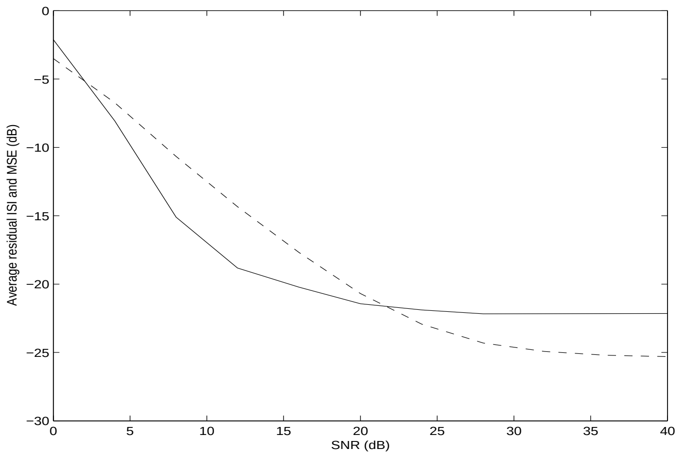

Fig. 4. Robustness curves ISI = ISI(SNR) (solid-line) and MSE = MSE(SNR) (dashed-line) against additive Gaussian noise at different SNR levels (on noisy telephonic channel).

point iterations are illustrated in the Figure 4. It refers to SNR values ranging from 0 dB to 40 dB. In order to take into account the statistical fluctuations of the performances, the obtained results have been averaged over 50 independent experiments corresponding to independent realizations of the noise sequence. The obtained robustness curves ISI=ISI(SNR) and MSE=MSE(SNR) look regular and, as might be expected, they saturate to the noiseless interference residual and mean square error levels for sufficiently high SNR values.

The last experiment concerns the comparison of the proposed algorithm (referred to as FPMBA) with some algorithms of the same class, namely the original 'Bussgang' (referred to as BGA), the 'Bussgang' with natural gradient (referred to as BGANG), the modified 'Bussgang' algorithm with adaptive Bayesian estimator (referred to as MBA) and the modified 'Bussgang' algorithm with natural gradient (referred to as MBANG). These are described in details in [8] and have been optimized separately. Also, as it is well-known that independent component analysis (ICA) algorithms [4] can be used for deconvolution, it is worth comparing the proposed algorithm with an ICA algorithm, in particular with the fixed-point 'FastICA' algorithm (please see [4] and references therein) properly adapted to a single neuron endowed with filtering synapses (referred to as FastICA-BD). To the best of our knowledge, there do not exist previous applications of the 'FastICA' technique to neural blind deconvolution, thus the version developed for comparison is a novelty as well. However, it was noted that the so-called "super-exponential" algorithm [14] contains the kurtosis-based 'FastICA' as a special case: For comparison purposes, such "super-exponential" algorithm (referred to as SEA) was implemented, as well.

The mentioned algorithms are compared in terms of separation performances (final ISI and MSE) as well as in terms of computational complexity. As all the considered algorithms are expressed in terms of complex non-linear adapting equations, the analytical estimation of their complexity is unfeasible, therefore the flops-count and the elapsed-time for every run are retained as measures of the computational burden of each algorithm. These values pertain to a 850MHz-128MB machine.

The results of this comparative analysis are summarized in the Table I. All the seven algorithms ran on the same batch of $1,000$ channel output samples and adapted through 20 sweeps of the same batch (which coincide to 20 iterations of the fixed-point algorithms). The flops count refers to the number of floating point operations required by the implemented code to run, averaged over the total number of samples (in this case $1,000 \times 20$), while the time count refers to the total time required by each algorithm to run on the specified platform.

TABLE I

Results of the comparative analysis of the separation ability and computational complexity exhibited by five Bayesian-estimation-based, an ICA-based deconvolution algorithms, and the "super-exponential" algorithm.

Algorithm	ISI (dB)	MSE (dB)	Flops	Time (sec.s)
BGA	-19.88	-22.99	272	21.37
BGANG	-7.77	-11.41	312	23.73
MBA	-19.34	-22.52	159	3.51
MBANG	-7.82	-11.44	189	6.04
FPMBA	-22.19	-25.41	114	0.11
FastICA-BD	-21.00	-24.07	112	0.16
SEA	-21.65	-24.75	136	0.22

The comparative analysis shows that the best-performing algorithm among the old ones in terms of ISI and MSE is the BGA; this is not surprising because it works on the basis of the exact Bayesian estimator. Nevertheless, the proposed FPMBA - as well as the other fixed-point algorithms - exhibits better numerical performances: The flops count shows that the FPMBA exhibits the best trade-off between deconvolution ability and computational complexity among the considered methods and the time count reinforces the conclusion that the fixed-point iteration offers a reliably fast adaptation rule. For instance, the FPMBA shows the ratio of $0.11/3.51 \approx 1/32$ of time demand with respect to the fastest algorithm among the iterative 'Bussgang' ones.

V. Conclusion

The aim of the present Letter was to illustrate some recent achievements on neural 'Bussgang' deconvolution theory, aimed at ameliorating the performances of the original algorithm by endowing it with a kind of source adaptivity and at reducing its computational complexity.

Experiments on a real-world system showed the good behavior of the proposed algorithm both in the noiseless and in the noisy cases, and a comparison with six existing algorithms of the same class clearly showed the relative merits of the proposed method.

Acknowledgment

I would like to thank the anonymous Reviewers whose interesting comments helped enriching the initial version of the manuscript.

Part of this research was carried out when I was visiting the Neuroscience Research Institute (AIST, Tsukuba – Japan): I would like to thank Dr. Y. Nishimori and the NRI people for the interesting and fruitful discussions on fixed-point algorithms and their convergence. I would also like to thank Dr. A. Hyvärinen (HUT, Helsinki – Finland) for kindly bringing to my attention the "super-exponential" algorithm [14].

References

[1] S. Bellini, *Blind equalization*, Alta Frequenza, Vol. 57, pp. 445 – 450, 1988

[2] A.J. Bell and T.J. Sejnowski, *An information maximisation approach to blind separation and blind deconvolution*, Neural Computation, Vol. 7, No. 6, pp. 1129 – 1159, 1995

[3] A. Benveniste, M. Goursat and G. Ruget, *Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data commu-*

nication, IEEE Trans. on Automatic Control, Vol. AC-25, No. 3, pp. 385 – 399, June 1980

[4] A. Chicocki and S.-i. Amari, *Adaptive Blind Signal and Image Processing*, J. Wiley & Sons, 2002

[5] S. Choi, S. Ong, J. Cho, C. You and D. Hong, *Performances of neural equalizers on partial erasure model*, IEEE Trans. on Magnetics, Vol. 33, No. 5, pp. 2788 – 2790, Sept. 1997

[6] J.B. Destro-Filho, G. Favier and J.M. Travassos-Romano, *New 'Bussgang' methods for blind equalization*, Proc. on International Conference on Acoustics, Speech and Signal Processing, pp. 2269 – 2272, 1997

[7] Z. Ding and Y. Li, *Blind equalization and identification*, Marcel Dekker, New York, 2001

[8] S. Fiori, *A contribution to (neuromorphic) blind deconvolution by flexible approximated Bayesian estimation*, Signal Processing, Vol. 81, No. 10, pp. 2131 – 2153, Sept. 2001

[9] S. Fiori, *Blind deconvolution by simple adaptive activation function neuron*, Neurocomputing, Vol. 48, No. 1-4, pp. 763 – 778, Oct. 2002

[10] S. Haykin, *Adaptive Filtering*, (Chapter 20: Blind Deconvolution), Prentice-Hall, 1991

[11] D. Kundur and D. Hatzinakos, *Blind image deconvolution*, IEEE Signal Processing Magazine, Vol. 13, No. 3, pp. 43 – 64, May 1996

[12] P.A. Regalia and E. Kofidis, *Monotonic convergence of fixed-point algorithms for ICA*, IEEE Trans. on Neural Networks, Vol. 14, No. 4, pp. 943 – 949, July 2003

[13] A.C. Tsoi and A.D. Back, *Discrete time recurrent neural network architectures: A unifying review*, Neurocomputing, Vol. 15, pp. 183 – 223, June 1997

[14] O. Shalvi and E. Weinstein, *Super-exponential methods for blind deconvolution*, IEEE Trans. on Information Theory, Vol. 39, No. 2, pp. 504 – 519, 1993

[15] R.A. Wiggins, *Minimum entropy deconvolution*, Geoexploration, Vol. 16, pp. 21 – 35, 1978