Stiefel-Manifold Learning by Improved Rigid-Body Theory Applied to ICA

Simone Fiori and Roberto Rossi

Faculty of Engineering, University of Perugia
Polo Didattico e Scientifico del Ternano,
Loc. Pentima bassa, 21, I-05100 Terni (Italy)

E-mail: sfr@unipg.it

Paper submitted to:
International Journal of Neural Systems

Pages: 29, **Figures and Tables**: 11, **References**: 44

Submitted May 6, 2003. Revised September 1, 2003.

Stiefel-Manifold Learning by Improved Rigid-Body Theory Applied to ICA

Simone Fiori and Roberto Rossi

Abstract

In previous contributions we presented a new class of algorithms for orthonormal learning of a linear neural network with p inputs and m outputs, based on the equations describing the dynamics of a massive rigid frame in a submanifold of $\mathbb{R}^p$. While exhibiting interesting features, such as intrinsic numerical stability, strongly binding to the orthonormal submanifolds, and good controllability of the learning dynamics, tested on principal/independent component analysis, the proposed algorithms were not completely satisfactory from a computational-complexity point of view. The main drawback was the need to repeatedly evaluate a matrix exponential map. With the aim to lessen the computational efforts pertaining to these algorithms, we propose here an improvement based on the closed-form Rodriguez formula for the exponential map. Such formula is available in the $p = 3$ and $m = 3$ case, which is discussed with details here. In particular, experimental results on independent component analysis (ICA), carried out with both synthetic and real-world data, help confirming the computational gain due to the proposed improvement.

1 Introduction

During the last years, several contributions appeared in the neural network literature as well as in other research areas regarding neural learning and optimization involving flows on special sets, such as the Riemannian manifolds.

Understanding the underlying geometric structure of a network parameters space is extremely important to designing systems that can effectively navigate the space while learning. Although modern mathematics is needed in the research of neural networks, and there are some very powerful results and techniques in these geometric methods, these are currently scattered in various sources.

Over the last decade or so, driven greatly by the work on information geometry, we are seeing the merging of the fields of statistics and geometry applied to neural network and learning. Such research topics involve Lie-group learning algorithms [24], the natural (Riemannian) gradient techniques [1, 25, 30], learning by weight flows on Stiefel-Grassman manifolds [22, 23, 24], the theories for learning on orthogonal group [10, 22], neuro-computing using Clifford geometric

algebra [4], the numerical aspects of the solution of the matrix-equations on Lie groups arising in neural learning/optimization and related topics [12, 16].

Some specific exemplary applied topics that can be addressed under the mentioned general methodology are: Principal component/subspace analysis [22, 42]; Neural independent component analysis and blind source separation [22, 43]; Information geometry [1]; Geometric Clifford algebra for the generalization of neural networks [4]; Geometrical methods of unsupervised learning for blind signal processing [22, 24]; Conformal and horosphere models for neuro-computing [4]; Tensorial approaches for geometrical neural computation and learning [5]; Eigenvalue and generalized eigenvalue problems, CS decomposition, optimal linear compression, noise reduction and signal representation [15, 18, 36, 41, 42]; Simulation of the physics of bulk materials [17]; Minimal linear system realization from noise-injection measured data and invariant subspace computation [17, 33]; Optimal de-noising by sparse coding shrinkage and local manifold projection [29, 37]; Linear programming and sequential quadratic programming [10, 17]; Optical character recognition by transformation-invariant neural networks [40]; Analysis of geometric constraints on neural activity for natural three-dimensional movement [44]; Electrical networks fault detection [31]; Synthesis of digital filters by improved total least-squares technique [26]; Adaptive image coding [34]; Dynamic texture recognition [38].

As a contribution to this research field, a new learning theory derived from the study of the dynamics of an abstract system of masses, moving in a multidimensional space under an external force field, was presented and studied in details in [23, 24]. The set of equations describing system's dynamics was interpreted as a learning algorithm for neural layers termed *MEC*. Relevant properties of the proposed learning theory were discussed, along with results of computer simulations performed in order to assess its effectiveness in applied fields.

In particular, some applications of the proposed approach were suggested, and cases of orthonormal independent component analysis and principal component analysis were tackled through computer simulations, which showed the MEC algorithm is effective and provides a good trade-off between numerical performances and computational complexity even when compared with closely-related algorithms.

An open question about the mentioned algorithm concerned the computational complexity which arises from the necessity of burdensome matrix computation, such as the repeated evaluation of the exponential map. The aim of the present paper is to investigate, in the simplified case of a 3-input/3-output network, a possible solution to the computational burden problem. The solution

is offered by the Rodriguez closed-form of exponential map in the 3×3 case. The new algorithm proposed here is applied to the analysis of non-destructive evaluation data.

2 Summary of the Theory and Proposed Improvement

In orthonormal learning, the target of the adaptation rule for a neural network is to learn an orthonormal weight-matrix related in some way to the input signal. Since it is a prior knowledge that the final state must belong to the subset of the whole weight-space containing the orthonormal matrices, the evolution of the weight-matrix could be strongly bounded to always belong to the orthonormal manifold.

We solved this strongly-binding problem by adopting as columns of the weight-matrix the position-vectors of some masses of a rigid system: Because of the intrinsic rigidity of the system, the required constraint is always respected.

By recalling that a (dissipative) mechanical system reaches the equilibrium when its own potential energy function (PEF) is at its minimum or local minima, a PEF may be assumed proportional to a cost function to be minimized, or proportional to an objective function to be maximized, both *under the constraint of orthonormality*.

In the following sections we briefly recall the mentioned theory, its principal features and the drawback related to its computational complexity. We then describe the proposed improvement.

2.1 Summary of rigid-body learning theory

Let $\mathcal{S}_m = \{(\mathcal{M}_i, \mathbf{w}_i), (\mathcal{M}_i, -\mathbf{w}_i)\}_{i \in \{1,\ldots,m\}}$ be a *rigid* system of masses, where the m vectors $\mathbf{w}_i \in \mathbb{R}^p$ represent the instantaneous positions of $2m$ masses $\mathcal{M}_i \in \mathbb{R}_0^+$ in a coordinate system. Such masses are positioned at constant (unitary) distances from the origin $\mathcal{O}$ fixed in the space $\mathbb{R}^p$, and over mutually orthogonal immaterial axes. In [23] we assumed the values of the masses $\mathcal{M}_i$ constant at 1. In Figure 1 an exemplary configuration of $\mathcal{S}_m$ for $p = 3$ and $m = 3$ is shown.

Note that by definition the system has been assumed rigid with the axes origin $\mathcal{O}$ fixed in the space, thus the masses are allowed only to instantaneously rotate around this point, while they cannot translate with respect to it.

The masses move in the space $\mathbb{R}^p$ where a point $\mathcal{P}$, endowed with a negligible mass, moves too; its position with respect to $\mathcal{O}$ is described by an independent

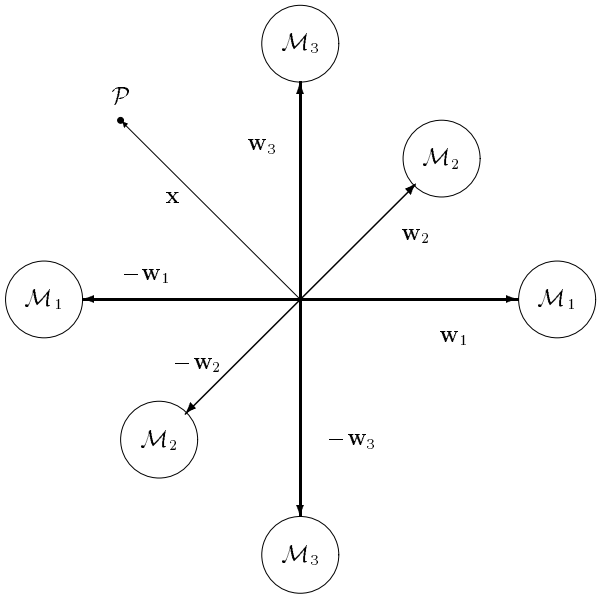

Figure 1: A configuration of $\mathcal{S}_m$ for $p = 3$ and $m = 3$. The $\mathcal{M}_i$'s denote the masses, the $\mathbf{w}_i$'s denote the masses' positions, $\mathcal{P}$ denotes the external point of coordinates $\mathbf{x}$.

vector $\mathbf{x}$ that, in the context of unsupervised learning considered in the paper, represents a random vector field. The point $\mathcal{P}$ exerts a force on each mass and the set of the forces so generated causes the motion of the global system $\mathcal{S}_m$. Furthermore, the masses move in an homogeneous and isotropic fluid endowed with a non-negligible viscosity: The corresponding resistance brakes the motion, makes the system dissipative and stabilizes its dynamics.

The equations describing the motion of such abstract system are summarized in the following proven result.

Theorem 1 *([23].) Let $\mathcal{S}_m \subset \mathbb{R}_0^+ \times \mathbb{R}^p$ be the abstract system described above: Let us denote with $\mathbf{F}$ the $p \times m$ matrix of the* active forces*, with $\mathbf{R}$ the $p \times m$ matrix of the* viscosity resistance*, with $\mathbf{\Omega}$ the $p \times p$* angular speed *matrix and with $\mathbf{W}$ the $p \times m$ matrix of the* instantaneous positions *of the masses. The dynamics of the system obeys the following equations:*

$$\frac{d\mathbf{W}}{dt} = \mathbf{\Omega}\mathbf{W} \ , \tag{1}$$

$$\frac{d\mathbf{\Omega}}{dt} = \frac{1}{4}[(\mathbf{F}+\mathbf{R})\mathbf{W}^T - \mathbf{W}(\mathbf{F}+\mathbf{R})^T] \ , \tag{2}$$

$$\mathbf{R} = -\mu\mathbf{\Omega}\mathbf{W} \ , \tag{3}$$

with μ being a positive parameter termed viscosity coefficient*.* □

The set of equations (1)-(3) may be assumed as a learning rule (briefly referred to as *MEC*) for a neural layer with weight-matrix $\mathbf{W}$. The MEC adaptation algorithm applies to any neural network described by the input-output transference $\mathbf{y} = \mathbf{S}[\mathbf{W}^T\mathbf{x} + \mathbf{w}_0]$, where $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{W}$ is $p \times m$, with $m \leq p$, $\mathbf{w}_0$ is a generic biasing vector in $\mathbb{R}^m$ and $\mathbf{S}[\cdot]$ is an arbitrarily-chosen $m \times m$ diagonal activation operator.

The basic properties of this algorithm may be summarized as follows:

- Let us denote by $\mathbf{so}(p, \mathbb{R})$ the set of skew-symmetric matrices[1]. It is immediate to verify that if $\mathbf{\Omega}(0) \in \mathbf{so}(p, \mathbb{R})$ then equation (2) makes $\dot{\mathbf{\Omega}}(t) \in \mathbf{so}(p, \mathbb{R})$ and thus $\mathbf{\Omega}(t) \in \mathbf{so}(p, \mathbb{R})$ because $\mathbf{so}(p, \mathbb{R})$ is a linear space;

- Let us denote by $\mathrm{St}(p, m, \mathbb{R})$ the set of the real-valued orthonormal $p \times m$ matrices (usually termed Stiefel manifold [9]). Because of the skew-symmetry of $\mathbf{\Omega}(t)$ we see from equation (1) that if $\mathbf{W}(0) \in \mathrm{St}(p, m, \mathbb{R})$ then $\mathbf{W}(t) \in \mathrm{St}(p, m, \mathbb{R})$ for all $t > 0$;

- The equilibrium conditions for the system (1)-(3), i.e. the stationarity conditions for the learning rule, write: $\mathbf{\Omega W} = \mathbf{0}_{p\times m}$, $\mathbf{F W}^T - \mathbf{W F}^T = \mathbf{0}_{p\times m}$, $\mathbf{W} \in \mathrm{St}(p, m, \mathbb{R})$, $\mathbf{\Omega} \in \mathbf{so}(p, \mathbb{R})$, where $\mathbf{0}_{p\times m}$ denotes the null element of $\mathbb{R}^{p\times m}$. It is important to recall that both $\mathbf{W}(t)$ and $\mathbf{\Omega}(t)$ are unknown and that $\mathbf{F}(t)$ is in general a non-linear function of the network's connection weights;

- As a mechanical system, stimulated by a conservative force field, tends to *minimize* its potential energy, the set of learning equations (1)-(3) for a neural network with connection pattern $\mathbf{W}$ may be regarded as a non-conventional (second-order, non-gradient) optimization algorithm.

The MEC learning rule possesses a fixed structure, the only modifiable part is the computation rule of the active forces applied to the masses. Here we suppose that the forcing terms derive from a super-symmetric potential energy function (PEF) U, which yields forces:

$$\mathbf{F} \stackrel{\text{def}}{=} -2\frac{\partial U}{\partial \mathbf{W}} \ . \tag{4}$$

Generally we may suppose U dependent upon $\mathbf{W}$, $\mathbf{w}_0$, and on the statistics of $\mathbf{x}$. In fact, if $\mathbf{x}$ is endowed with a probability density function $p_{\mathbf{x}}(\mathbf{x})$, the PEF computes as:

$$U = \int_{\mathbb{R}} p_{\mathbf{x}}(\mathbf{x}) u(\mathbf{W}, \mathbf{w}_0, \mathbf{x}, \mathbf{y}) d\mathbf{x} \ , \tag{5}$$

[1] The plain definition is $\mathbf{so}(p, \mathbb{R}) \stackrel{\text{def}}{=} \{\mathbf{C} \in \mathbb{R}^{p\times p} | \mathbf{C}^T = -\mathbf{C}\}$

where $u(\cdot,\cdot,\cdot,\cdot)$ represents a network's performance index. Recalling that a (dissipative) mechanical system reaches an equilibrium state when its own potential energy U is at its minimum (or local minima), we can use as PEF any arbitrary smooth function to be optimized. Vector $\mathbf{w}_0$ may be arbitrarily adapted.

If we regard the above learning rule as a minimization algorithm, the following observations might be worth noting:

- The search space is considerably reduced; if fact, the set of matrices belonging to $\mathbb{R}^{p\times m}$, with $p \geq m$, has pm degrees of freedom, while the subset of same-size orthonormal matrices has $pm - m(m+1)/2$ degrees of freedom;
- Non-orthonormal local (sub-optimal) solutions are inherently avoided as they do not belong to the search-space.

To conclude the summary of MEC theory, it is useful to mention that we possess two proven results about the stationary points of the algorithm and on their stability.

Theorem 2 *([24].) Let us consider the dynamical system (1)-(3) where the initial state is chosen so that* $\mathbf{W}(0) \in St(p,m,\mathbb{R})$ *and* $\mathbf{\Omega}(0)$ *is skew-symmetric. Let us also define the matrix function* $\mathbf{F} \stackrel{\text{def}}{=} -\kappa\frac{\partial U}{\partial \mathbf{W}}$*, and denote as* $\mathbf{F}_\star$ *the value of* $\mathbf{F}$ *at* $\mathbf{W}_\star$*. A state* $\mathbf{X}_\star = (\mathbf{\Omega}_\star, \mathbf{W}_\star)$ *is stationary for the system if* $\mathbf{F}_\star^T\mathbf{W}_\star$ *is symmetric and* $\mathbf{\Omega}_\star\mathbf{W}_\star = \mathbf{0}$*. These stationary points are among the extremes of learning criterion* U *over* $St(p,m,\mathbb{R})$*.*

Let us denote by $\mathrm{SO}(p,\mathbb{R})$ the set of real-valued square orthonormal matrices of dimension p (namely, $\mathrm{SO}(p,\mathbb{R}) = \mathrm{St}(p,p,\mathbb{R})$).

Theorem 3 *([24].) Let* U *be a real-valued function of* $\mathbf{W}$*,* $\mathbf{W} \in SO(p,\mathbb{R})$*, bounded from below with a minimum in* $\mathbf{W}_\star$*. Then the equilibrium state* $\mathbf{X}_\star = (\mathbf{0}, \mathbf{W}_\star)$*, is asymptotically (Lyapunov) stable for system (1)-(3) if* $\mu > 0$*, while simple stability holds if* $\mu \geq 0$*.*

In the present paper we consider linear neural networks with dynamics in $\mathrm{SO}(3,\mathbb{R})$.

2.2 Study motivation

In order to implement the MEC algorithm on a discrete-time machine, a discretization of the continuous-time equations (1)-(3) is necessary. In the following we recall from [23] a way to perform discretization which gives good results, i.e.

that allows keeping the orthonormality of the columns of $\mathbf{W}$ with a degree of accuracy as good as one needs.

The expression (2) may be discretized by replacing $d\mathbf{\Omega}/dt$ with $\Delta\mathbf{\Omega}/\theta$, where $1/\theta$ plays the role of a "sampling rate" and symbol $\Delta\mathbf{\Omega}$ denotes discrete-time variation of the discretized matrix-variable $\mathbf{\Omega}$. This corresponds to using a standard (forward or backward) Euler method for solving equation (2): This is allowed because, again, the space of skew-symmetric matrices is linear. In this way $\mathbf{\Omega}$ keeps constant within each open temporal interval $]n\theta, (n+1)\theta]$, and equals $\mathbf{\Omega}_n \stackrel{\text{def}}{=} \mathbf{\Omega}(n\theta)$. Then equation (1) can be solved exactly within each of these intervals, in that it can be rewritten for $n \geq 0$ as:

$$\frac{d\mathbf{W}}{dt} = \mathbf{\Omega}_n \mathbf{W} \text{ for } t \in]n\theta, (n+1)\theta] \ ,$$

with $\mathbf{W}(n\theta)$ being known from calculus for $t \in](n-1)\theta, n\theta]$ when $n \geq 1$, and from the initial condition when $n = 0$. The solution evaluated in $t = (n+1)\theta$ gives:

$$\mathbf{W}((n+1)\theta) = \exp(\mathbf{\Omega}_n \theta)\mathbf{W}(n\theta) \ . \tag{6}$$

The above expression represents a Lie-group Euler integration method that replaces the standard Euler method which does not work on curved spaces such as the Stiefel manifold (see [12] and references therein).

In the general case, the exponential map $\exp(\mathbf{\Omega}_n\theta)$ is not easily computable; we used the definition of the exponential map in terms of a MacLaurin series and employed the approximation $\exp(\mathbf{\Omega}_n\theta) \cong \sum_{k=0}^{r} \frac{\mathbf{\Omega}_n^k}{k!}\theta^k$ with r being a non-negative bounded integer. This way, the following set of learning equations was obtained:

$$\Delta\mathbf{W} = \left[\sum_{k=1}^{r} \frac{\mathbf{\Omega}_n^k}{k!}\theta^k\right]\mathbf{W} \ , \ \mathbf{R} = -\mu\mathbf{\Omega}\mathbf{W} \ , \tag{7}$$

$$\Delta\mathbf{\Omega} = \frac{\theta}{4}[(\mathbf{F}+\mathbf{R})\mathbf{W}^T - \mathbf{W}(\mathbf{F}+\mathbf{R})^T] \ , \tag{8}$$

with $\theta > 0$, $\mu > 0$ and r fixed. There all quantities are intended to be evaluated at the same discrete-time step n.

The following observations deserve attention:

- The constant θ and r arise from the continuous-time equations discretization and their choice determines how much the layer's weight matrix belongs to the manifold of orthonormal matrices. This observation suggests that r should be large enough in order for the truncated expression (7) to account for as many meaningful terms as necessary;

- Conversely, the parameter r should be chosen as small as possible because the computation of terms like $\mathbf{\Omega}_n^k \theta^k / k!$ has a significant impact on the total computational complexity of the algorithm.

The selection of the order r of approximation is thus a non-trivial question. In the paper [23] we concluded that the best values in the examined applied cases were $r = 3$ or 4.

2.3 The Rodriguez formula for exponential map

It is clear that the most computationally burdensome expression in the equation (6) is the matrix exponentiation $\exp(\mathbf{C})$ of skew-symmetric terms $\mathbf{C}$. The approximation (7), however, suffers from possible deviation of $\mathbf{W}$ from orthonormality and of excessive computational burden. We wonder what is a computationally convenient technique for implementing such calculus on a computer.

At least in the simplified case of a 3-input/3-output neural network, the answer comes from the closed-form Rodriguez formula for exponential map computation over the Lie algebra $\mathbf{so}(3, \mathbb{R})$ [12].

Let us illustrate the Rodriguez formula for the 3×3 case, i.e. when $\mathbf{W} \in \mathrm{St}(3, 3, \mathbb{R}) = \mathrm{SO}(3, \mathbb{R})$. For the sake of notation simplicity, let us define $\bar{\mathbf{\Omega}}_n \stackrel{\text{def}}{=} \theta \mathbf{\Omega}_n$ and let us drop down the time-index n.

The skew-symmetric matrix $\bar{\mathbf{\Omega}} \in \mathbf{so}(3, \mathbb{R})$ may be parameterized as follows:

$$\bar{\mathbf{\Omega}} = \begin{bmatrix} 0 & -\bar{\omega}_3 & \bar{\omega}_2 \\ \bar{\omega}_3 & 0 & -\bar{\omega}_1 \\ -\bar{\omega}_2 & \bar{\omega}_1 & 0 \end{bmatrix} , \tag{9}$$

where scalars $\bar{\omega}_1$, $\bar{\omega}_2$ and $\bar{\omega}_3$ are the only three free parameters of the considered skew-symmetric matrix in $\mathbf{so}(3, \mathbb{R})$. Let us define the Rodriguez angle α as:

$$\alpha \stackrel{\text{def}}{=} \sqrt{\bar{\omega}_1^2 + \bar{\omega}_2^2 + \bar{\omega}_3^2} . \tag{10}$$

On the basis of these quantities, we have the exact identity [12]:

$$\exp(\bar{\mathbf{\Omega}}) = \mathbf{I}_3 + \frac{\sin \alpha}{\alpha} \bar{\mathbf{\Omega}} + \frac{1 - \cos \alpha}{\alpha^2} \bar{\mathbf{\Omega}}^2 , \tag{11}$$

where $\mathbf{I}_3$ denotes the 3×3 identity matrix.

The above Rodriguez formula for exponential map computation allows computing the matrix exponentiation required by the MEC algorithm to run without any approximation and with a computational saving that shall be evaluated numerically in the following sections. It allows introducing an improved version of the MEC algorithm, valid for training a 3×3 neural network, that we refer to

as MEC_2 algorithm. It writes:

$$\alpha = \theta\sqrt{\omega_1^2 + \omega_2^2 + \omega_3^2} \ , \tag{12}$$

$$\Delta\mathbf{W} = \left[\mathbf{I}_3 + \frac{\sin\alpha}{\alpha}\theta\boldsymbol{\Omega} + \frac{1-\cos\alpha}{\alpha^2}\theta^2\boldsymbol{\Omega}^2\right]\mathbf{W} \ , \tag{13}$$

$$\mathbf{G} \stackrel{\text{def}}{=} \mathbf{F}\mathbf{W}^T - \mu\boldsymbol{\Omega} \ , \Delta\boldsymbol{\Omega} = \theta(\mathbf{G} - \mathbf{G}^T) \ , \tag{14}$$

where the meaning of the symbols is immediate. Note that we have replaced the definition of $\mathbf{R}$ and the calculus of $(\mathbf{F}+\mathbf{R})\mathbf{W}^T - \mathbf{W}(\mathbf{F}+\mathbf{R})^T$ in (8) with the definition of new matrix $\mathbf{G}$ and the calculus of $\mathbf{G} - \mathbf{G}^T$, and the the constant $\frac{1}{4}$ has been absorbed in the constants; the orthogonal-group property $\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^T = \mathbf{I}_3$ has been exploited, too.

It deserves to note that it could be useful to consider Clifford algebras for the formulation of the Rodriguez formula using bivector algebras: Since matrices have many redundant coefficients, by means of bivectors (e.g. quaternions or dual quaternions), we could be enabled to simplify the learning procedure and perhaps the computation may be even faster. This improvement lies, however, outside the scope of the manuscript.

3 Experimental set-up on independent component analysis

Blind signal separation techniques allow to recover unknown signals by processing their observable mixtures, which are the only available data. In particular, under the hypothesis that the source signals to separate out are statistically independent and are mixed by a linear full-rank operator, the neural independent component analysis (ICA) theory may be employed: It aims at re-mixing the observed mixtures in order to make the resulting signals *as independent as possible* [6, 13, 19, 20, 21]. In practice, a suitable measure of statistical dependence is exploited as an optimization criterion which drives network's learning.

In the following we use the well-known result [13] whereby it is known that an ICA stage can be decomposed into two subsequent stages: A pre-whitening stage and a orthonormal-ICA one, therefore the signal $\mathbf{z} = \mathbf{M}^T\mathbf{s}$ at the sensors can be first standardized and then *orthonormally separated* by a three-layer neural network as depicted in Figure 2. Here we suppose the source signal stream $\mathbf{s} \in \mathbb{R}^m$, observed linear mixture stream $\mathbf{z} \in \mathbb{R}^p$, with $p \geq m$, thus the mixing matrix $\mathbf{M}^T \in \mathbb{R}^{p\times m}$.

In the following experiments, the aim is to separate out independent signals from their linear mixtures. To this aim, the following simple potential energy

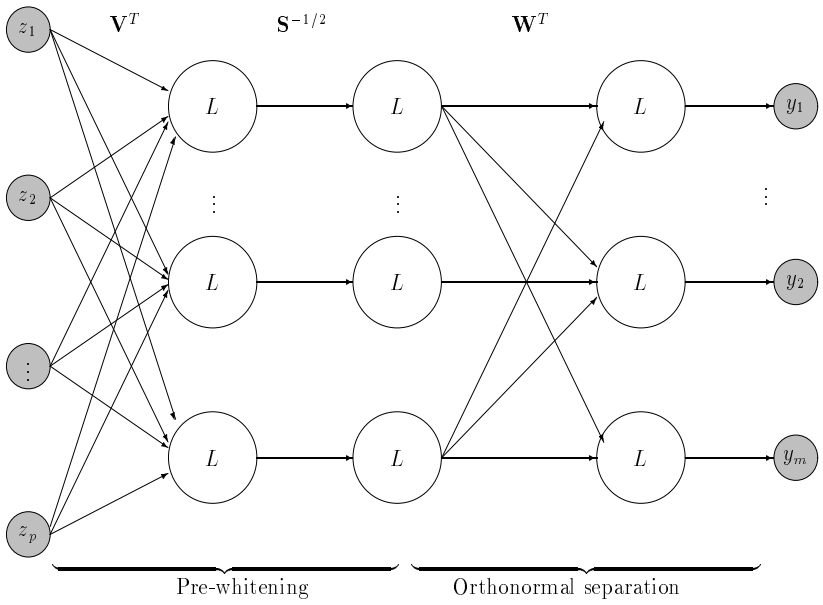

Figure 2: Three-layer neural architecture for blind source separation. The circled 'L' denotes a linear neural unit.

function may be used as optimization criterion [11, 14]:

$$U(\mathbf{W}) = \frac{k_{\mathrm{C}}}{8} \sum_{i=1}^{4} E_{\mathbf{x}}[y_i^4] \ , \tag{15}$$

where k_{C} is a scaling factor. The resulting active force has the expression:

$$\mathbf{F} = -k_{\mathrm{C}} E_{\mathbf{x}}[\mathbf{x}(\mathbf{x}^T\mathbf{W})^3] \ , \tag{16}$$

where the $(\cdot)^3$-exponentiation acts component-wise. In practice, as we are interested in on-line learning, the expectation operator $E_{\mathbf{x}}[\cdot]$ is dropped down so that statistical-mechanical learning is resorted to.

The whitening matrix pair $(\mathbf{S}, \mathbf{V})$ computes as follows: If $\mathbf{C}_{zz}$ denotes the covariance matrix of the multivariate random vector $\mathbf{z}$, then $\mathbf{S}$ contains the eigenvalues and $\mathbf{V}$ contains (as columns) the corresponding eigenvectors of the covariance matrix. The number of eigen-pairs considered needs not to be equal to the dimension of the input signal: If only m eigenvalues are retained then an efficient data-compression is achieved which allows to operate on a reduced-size input vector-stream; in this case the orthonormal-ICA network is square (i.e. $m \times m$).

3.1 Algorithms considered for comparison

In order to compare the performances of the presented algorithm with those exhibited by other algorithms known from the scientific literature, we present simulation results obtained by running the Bell-Sejnowski's algorithm with Amari's natural gradient (referred to as WACY) [2], the Laheld-Cardoso algorithm (referred to as LC) [11], the Oja-Hyvärinen algorithm [27] (referred to as OH), and the algorithm proposed by Nishimori [35] (referred to as NMA). We also consider the old version of MEC algorithm [23] in order to compare its numerical performance and computational burden with the features of the new algorithm.

The mentioned learning rules for linear neural structures allow the networks to perform blind signal separation and were derived on the basis of rather different principles: The Bell-Sejnowski's algorithm derives from a gradient-based optimization of an INFOMAX criterion, based on minimization of mutual information among network's output channels, and has been improved by the use of Amari's natural gradient theory; the Laheld-Cardoso algorithm comes from relative-gradient optimization of standard kurtosis-based criterion and the algorithm presents under the form of serial updating; the Oja-Hyvärinen algorithm derives by an extension to Hebbian learning and achieves the orthonormality of the network weight matrix by the help of a penalty term which penalizes deviation from orthonormality; the Nishimori's algorithm derives from the theory of optimization over a Riemannian manifold and exploits the differential geometrical structure of network's parameters space to define an approximate geodesic learning-path on it. A good summary of most of these algorithms, along with the details of their implementation, was given in [28].

3.2 Performance indices description

Since the overall source-to-output matrix $\mathbf{P} \stackrel{\text{def}}{=} \mathbf{W}^T\mathbf{S}^{-1/2}\mathbf{V}^T\mathbf{M}^T \in \mathbb{R}^{m\times m}$ should become as quasi-diagonal (i.e. such that only one entry per row and column differs from zero) as possible, we might took as convergence measure the general Comon time-index [13]. The Comon index measures the distance between the source-to-output separation matrix and a quasi-identity and is able to measure also degeneracy, that is the case where the same source signals get encoded by two or more neurons. However, in the present context degeneracy is impossible, basically thanks to the fact that learning happens on the orthogonal group which inherently prevent the different neurons from sharing the same source signals (consequently $\det(P) \neq 0$). We may thus employ the reduced

criterion:

$$F_2(t) \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} P_{ij}^2(t) - \sum_{i=1}^{m} \max_k\{P_{ik}^2(t)\}}{\sum_{i=1}^{m} \max_k\{P_{ik}^2(t)\}} \,, \tag{17}$$

that is a proper measure of distance between $\mathbf{P}$ and an unspecified quasi-diagonal matrix at any time.

Let us denote with $\mathcal{T} = [0, T]$ the time-interval that the algorithm runs within[2]. Each run of an algorithm generates a whole curve $\{F_2(t)\}_{t\in\mathcal{T}}$, thus, in order to present results in a compact numeric format, we decided to define four shape-index that try to describe numerically the learning curves:

1. F_2^M: This is defined as the maximum value of the index $F_2(t)$ over $t \in \mathcal{T}$. If the curve $F_2(t)$ is monotonically decreasing, the maximal value should be assumed at $t = 0$; a value $F_2^M > F_2(0)$ denotes the occurrence of an error-spike in the middle of the simulation which, in turn, denotes a difficulty of convergence;

2. F_2^∞: This number is defined as the last value took by the index, that is the final separation index value $F_2(T)$. It measures the separation ability of the learnt network after convergence;

3. F_2^{ave}: This is defined as an average value of $F_2(t)$ during network learning. In order to consider the worst error-values at the beginning of learning, the average is taken only over the first half-period of learning, that is with $t \in [0, T/2]$. This index is designed to measure the convergence speed of the algorithm: The lesser the F_2^{ave} index value is, the faster the algorithm separates out the source signals;

4. F_2^{var}: This is defined as the variance of $F_2(t)$ during network learning. In order to avoid considering large error-values at the beginning of learning, the expectation value is computed over the last fourth-of-period of learning only, that is with $t \in [3T/4, T]$. This index is designed to measure the stability of the algorithm around the separating solution: The lesser the F_2^{var} index value is, the lesser the algorithm oscillates around the separating network configuration.

About parameter F_2^{var}, it is important to specify what follows: In order for the reason which led to the definition of this index to hold, the algorithm that it measures the stability of must have reached convergence within $3T/4$, otherwise the index F_2^{var} loses its meaning. As a matter of fact, when an algorithm is excessively slow in reaching convergence, it is possible that the variance of F_2

[2]In the present context, we deal with discrete-time signals. Therefore, the value of T is conventionally set to 1 (sec).

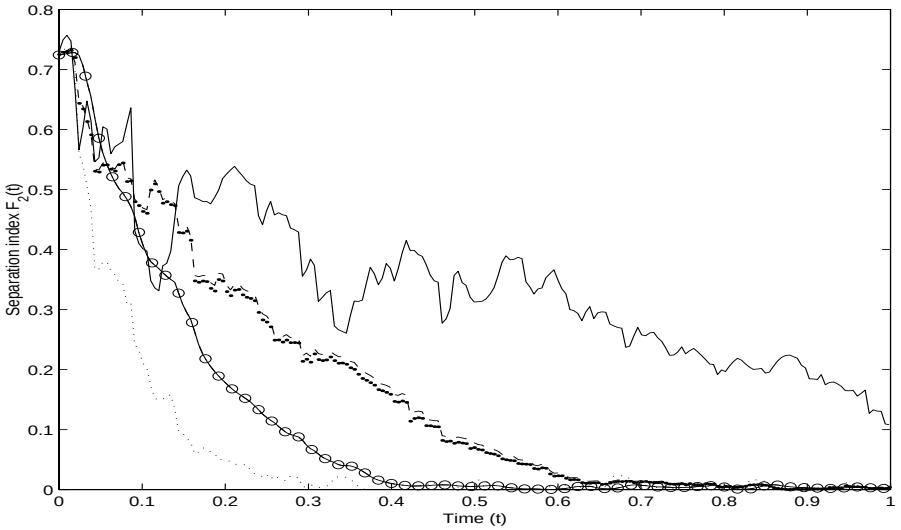

Figure 3: Exemplary behavior of $F_2(t)$ curves for the six considered algorithms. (Solid-line: WACY, Solid-dotted: MEC and Solid-circled: MEC_2 (these two are superimposed in this single trial), Dashed-line: OH and Thick-Dotted-line: NMA (these two are nearly superimposed in this single trial), Dotted-line: LC).

over the last fourth-of-period of learning is not appropriate to this analysis. The problem arises by the fact that the period of analysis is fixed regardless of the algorithm speed; however, this problem may be solved by assuming that the value of index F_2^{var} is meaningful only if the convergence speed, as measured by F_2^{ave}, is sufficiently high.

As a cautionary example of $F_2(t)$ index behavior, let us examine the curves displayed in Figure 3: The LC, MEC and MEC_2 algorithm converge in $0.4T$, thus the stability can be measured in $[0.4T, T]$; on the other hand, NMA and OH are slower and it would be meaningful to measure the stability of the results they achieve in $[0.7T, T]$ only; The WACY algorithm does not converge in this single trial, and it is not meaningful to measure the stability of the result in any subinterval of $[0, T]$.

It is noted that the above shape indices are somewhat arbitrary and could be replaced by other valid indices. For example, instead of employing F_2^{ave}, which is supposed to measure the convergence speed, we could measure the average gradient over the first halve of the learning period.

We also considered two complexity parameters:

1. The average number of floating point operations (flops) per iteration required when running MATLAB© code implementations of the considered

learning algorithms;

2. The total CPU-time that the algorithms take to run on a 500MHz, 512MB machine [3].

It is worth noting that the flops index just takes into account computational-complexity factors, such as the number of multiplication required in order to compute the quantities needed by the algorithms to be implemented, while the CPU-time also accounts for times lost by the machine to e.g. swap arrays in the memory. For this reason we believe they both are worth considering.

In our analysis we also considered an index that takes into account the ability of the network to keep the network weight-matrix orthonormal at any iteration, that is one of the main features of MEC-type learning algorithms class. A measure of the orthonormality of $\mathbf{W}$ was defined as:

$$\gamma(t) \stackrel{\text{def}}{=} \|\mathbf{W}^T(t)\mathbf{W}(t) - \mathbf{I}_m\|_{\text{F}}^2 \ ,$$

where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm. It appears as a valid measure of non-orthonormality.

To end with, it is important to note that the performances index $F_2(t)$ may be defined only when the separation algorithms are run over synthetic data, because they require the knowledge of the mixing operator $\mathbf{M}$ to be computed.

In a real-world test, however, the mixing parameters are of course unknown and the performance index should be replaced with a 'blind' state index that is able to describe the internal state of the network and which allows to easily recognize when the network has found a stable configuration.

In the manuscript [24], we recently suggested a possible internal-state indicator on the basis of rational-kinematics considerations. In particular, as the MEC (as well as MEC_2) network is considered quiet when the kinetic energy of the associated massive body is near zero, we suggested to assume the kinetic energy $K(t)$ as an index of network internal-learning-state. This energy is computed as usual for the discrete-ensembles of point-particles [3] and writes:

$$K(t) \stackrel{\text{def}}{=} \frac{1}{2}\text{tr}[(\mathbf{\Omega W})^T(\mathbf{\Omega W})] = -\frac{1}{2}\text{tr}[\mathbf{W}^T\mathbf{\Omega}^2\mathbf{W}] \ . \tag{18}$$

We in fact used the latter expression, where symbol $\text{tr}[\cdot]$ denotes the trace of the matrix contained within the brackets.

[3] It is worth noting that it is important to specify at least the clock frequency and the total RAM memory of the machine that the experiments are performed on for further comparison purpose. In particular, the memory size is relevant when one deals with large amount of data because large memory segments allow fast storing and swapping of data. On the other hand, it is also worth noting that, unfortunately, different types of CPU may mean different behaviors).

4 Experimental Results

In the following we present experimental results obtained with a 3×3 network on both synthetic data (in order to validate the network learning parameters) and real-world data, in order to ascertain the usefulness of the proposed approach on applied fields.

4.1 Experiments on synthetic data

The first set of experiments have been conducted on synthetic data. In particular, the 3 considered source signals are defined as follows:

$$\mathbf{s} = \begin{bmatrix} \cos(2\pi 50.0t) \\ \sin(2\pi 30.0t + 6\cos(2\pi 6.0t)) \\ \sin(2\pi 9.0t) \end{bmatrix},$$

and the mixing matrix is random with entries randomly drawn from a normal distribution.

These experiments pertain to the algorithm MEC_2 only and have been conducted with $\theta = 0.001$ and with the combinations of values $k_C = 2, 4, 6, 8$ and $\mu = 2, 4, 6, 8$ in order to find the best pairs of these parameters for subsequent experiments, that is to perform algorithm parameter validation.

We ran the MEC_2 algorithm and computed the four shape-indices, whose values have been averaged over twenty independent trials in order to present results as much independent as possible from statistical fluctuations in the performances. In any trial the mixing matrix has been generated independently of the other trials, but the data has been kept the same for any of the 16 combinations of parameters, in order for the different configurations to be tested on the same data-set. The obtained results are reported in Table 1.

The interpretation of these results is particularly meaningful: The behavior of learnt-network performance index F_2^{∞} is not monotonic with the growing of parameters k_C and μ, therefore the validation process over all the 16 possible combinations of the considered values for these parameters was necessary. Also, it is worth noting that the convergence speed, here conventionally measured by the index F_2^{ave}, monotonically increases with the increasing of value k_C for any fixed value of μ: This result may be explained by considering the physical meaning of the value k_C that controls the magnitude of the forcing terms. About the stability index F_2^{var}, it is expected that higher values of μ lead to lower values of F_2^{var} which would evidence enhanced stability of the achieved network configuration; as mentioned, the meaningfulness of the stability index is related to the convergence speed of the algorithm: We note that for the configurations

μ	k_C	$\langle F_2^M \rangle$ (dB)	$\langle F_2^\infty \rangle$ (dB)	$\langle F_2^{ave} \rangle$ (dB)	$\langle F_2^{var} \rangle$ (dB)
2.000	2.000	-2.502	-21.463	-4.683	-45.472
	4.000	-2.497	-24.892	-5.780	-48.372
	6.000	-2.457	-24.574	-6.367	-45.320
	8.000	-2.424	-22.148	-6.529	-42.031
4.000	2.000	-2.524	-19.368	-4.194	-39.945
	4.000	-2.474	-32.873	-5.431	-57.947
	6.000	-2.494	-33.726	-6.298	-60.709
	8.000	-2.479	-32.404	-6.928	-57.299
6.000	2.000	-2.532	-11.433	-3.833	-34.715
	4.000	-2.502	-28.100	-4.981	-51.601
	6.000	-2.481	-34.109	-5.863	-60.885
	8.000	-2.492	**-35.077**	**-6.577**	**-61.372**
8.000	2.000	-2.535	-8.513	-3.587	-34.393
	4.000	-2.516	-19.824	-4.591	-41.253
	6.000	-2.488	-31.129	-5.443	-54.867
	8.000	-2.484	-32.884	-6.144	-60.065

Table 1: Learning parameters validation for MEC_2 algorithm: Indices F_2^M, F_2^∞, F_2^{ave} and F_2^{var}, expressed in dB, averaged over 20 independent trials.

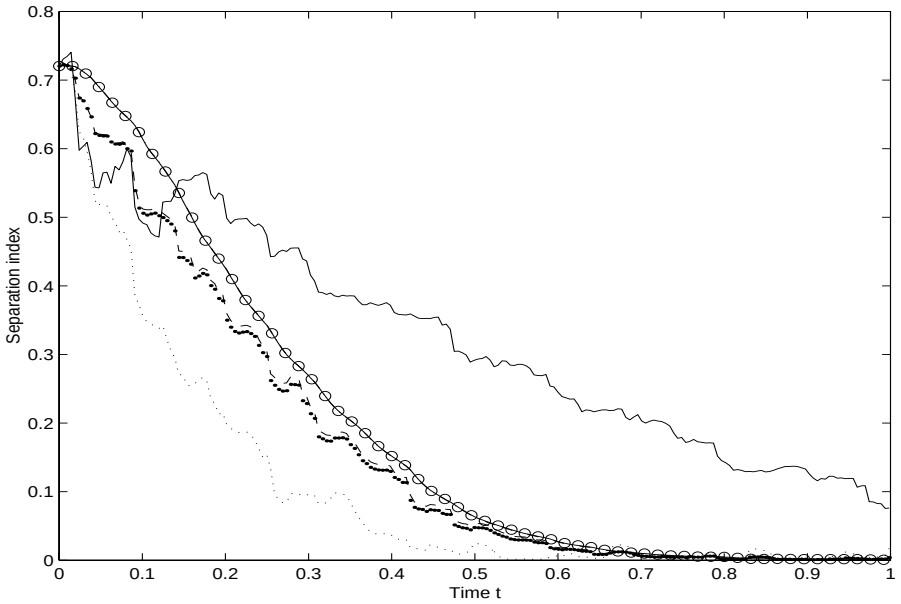

Figure 4: Comparison of the six considered algorithms: Courses of index $F_2(t)$. (Solid-line: WACY, Solid-dotted: MEC, Solid-circled: MEC_2, Dashed-line: OH, Thick-Dotted-line: NMA, Dotted-line: LC).

that values $F_2^{ave} < -6$ dB pertain to, the variance-index values are ordered in a decreasing way with μ increasing, as expected.

As a consequence, it is straightforward to conclude that the best parameter configuration among the ones considered here is $\mu = 6$ and $k_C = 8$, as evidenced in Table 1.

The initial average separation index in the simulations was about -2.5 dB, and the column F_2^M reveals that in all the experiments the situation $F_2^M \gg F_2(0)$ never occurred.

Having found the best learning-values for the considered set of data, we proceeded with parallel simulations in order to compare the behavior of the six considered algorithms. Again it was assumed $\theta = 0.001$ in the MEC and MEC_2 and the learning step-sizes for the remaining four algorithms were assumed equal to θ. The courses of the index $F_2(t)$ for the six algorithms are shown in the Figure 4, averaged over 20 independent trials, while Table 2 summarizes the values of the four shape-indices for the six algorithms, averaged over 20 independent trials. Both Figure 4 and Table 2 clearly evidence the good behavior of algorithm MEC, MEC_2 and NMA on the considered blind source separation problem. In particular, the MEC and MEC_2 algorithms maximize the triplet learnt-network-performance, convergence-speed and converged-solution

Algorithm	$\langle F_2^M \rangle$ (dB)	$\langle F_2^\infty \rangle$ (dB)	$\langle F_2^{ave} \rangle$ (dB)	$\langle F_2^{var} \rangle$ (dB)
WACY	-1.466	-13.817	-3.447	-24.580
MEC	-1.466	-32.622	-4.195	-40.962
LC	-1.466	-17.637	-7.892	-44.738
OH	-1.466	-23.172	-4.757	-37.460
NMA	-1.466	-23.355	-4.837	-38.250
MEC_2	-1.466	-32.764	-4.199	-40.927

Table 2: Comparison of the six considered algorithms: Indices F_2^M, F_2^∞, F_2^{ave} and F_2^{var}, expressed in dB, averaged over 20 independent trials.

stability.

Two additional aspects must be taken into account, however, in the comparison: The ability of keeping the weight-matrix orthonormal and the computational complexity of the algorithms.

The computational burden of the learning rules is monitored by the flops and CPU-time indices, and a summary of the found result is graphically given in Figure 5. The computational saving due to the introduction of the Rodriguez formula into MEC learning equation is apparent and may be numerically estimated to be about 33%. It should be noted that MEC_2 appears to be slower than MEC in CPU-time by about 25%, even though the flops figure is lower. This is not surprising because, as already mentioned, the two indices measure different behaviors of the machine that the algorithms under considerations have been implemented on. The above result shows that, in the experiment, the computer spent time not devoted to computation but to other operations.

The ability to keep the connection-matrix $\mathbf{W}$ orthonormal during learning is measured by the proper index for the OH, WACY, LC and NMA algorithms, as well as for the MEC_2 algorithm and for the MEC algorithm considered in the three versions with $r = 3$, $r = 2$ and $r = 1$. The results of this analysis are reported in Figure 6.

This result shows that the MEC algorithms behave better for large values of the approximation degree r, but the MEC_2, thanks to the exact formulation of exponential map by Rodriguez, keeps the orthonormality of network connection-matrix up to machine precision. NMA algorithm also allows the same good result, but at the expense of a definitely higher computational burden.

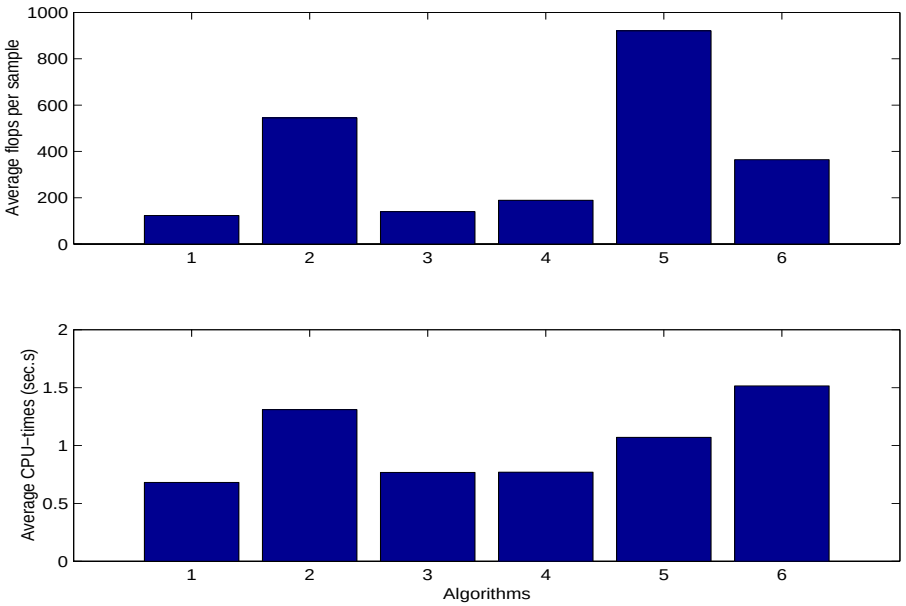

Figure 5: Comparison of the six considered algorithms: Flops per sample and CPU-time (sec.s.) averaged over 20 idependent trials. (1 = WACY, 2 = MEC, 3 = LC, 4 = OH, 5 = NMA and 6 = MEC_2.)

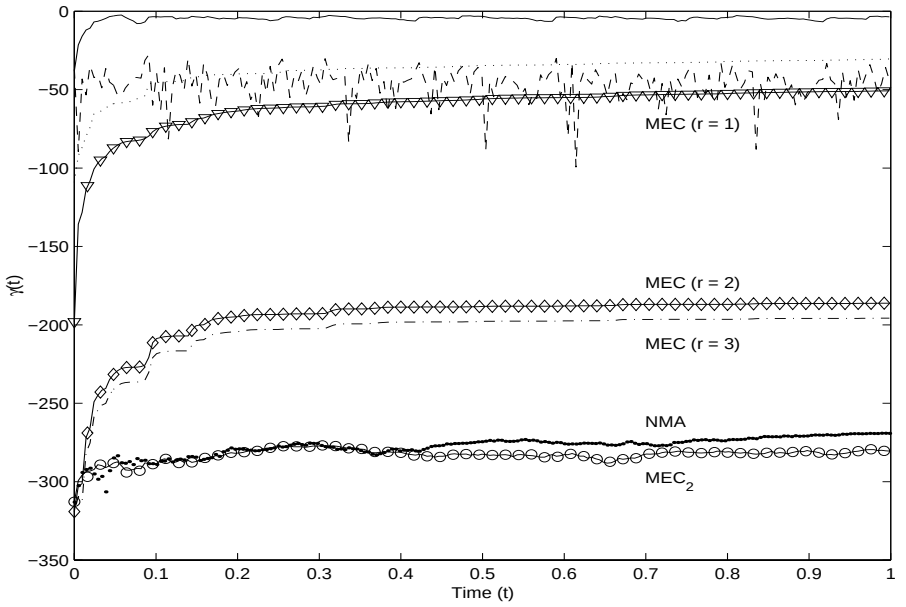

Figure 6: Comparison of the six considered algorithms: Courses of index $\gamma(t)$.

4.2 Experiments on NDE data

The aims of non-destructive evaluation (NDE) are to analyze material properties, to locate and characterize flaws within materials, fabricated parts and assemblies, and to monitor or image the internal as well as the external conditions of mechanisms, without damaging or altering the functionality of tested objects.

Well-known NDE inspection techniques rely on conventional radiography, microfocus radiography, computer-based tomography, flash radiography, eddy-current, x-ray fluorescence, infrared imaging, ultrasonics, fluorescent or dye penetrant, magnetic particle and time-resolved radiography. Applications range from non-destructive evaluation of composite components (such as glass-fiber reinforced systems) to the non-destructive determination of the stiffness of asphalt concrete, from the non-destructive evaluation of wood and timber to the evaluation of steel slabs. The inspection technique to be used depends strongly on the kind of material under test. For instance, eddy-current testing (ECT), based on the phenomenon of parasitic electrical currents, is particularly well suited for non-destructive evaluation of conductive objects such as metallic slabs [7, 8].

The eddy current testing is based on the following physical principle: A time-varying magnetic field is induced in the sample material by using a magnetic coil excited with alternating current. This magnetic field causes an electric current to be generated in conducting materials which, in turn, produce small magnetic fields around the conducting materials. The weaker magnetic fields generally oppose the original field which changes the impedance of the magnetic coil. Thus, by measuring the changes in impedance of the magnetic coil as it traverses the sample, it is possible to identify different characteristics of the sample.

In general, eddy-current-based NDE is used to detect flaws such as cracks, voids and inclusions, to determine material properties (as e.g. alloy type, heat treat and thermal history) and to measure coating thickness and displacements. Examples of applications are the measurement of stress corrosion in aluminum, the measurement of coatings and displacements in the micron range, the discrimination of aluminum alloys on the basis of resistivity, and the measurement of resistivity variations in carbon composite materials (carbon is a conductor in this case).

The spatial resolution achievable with eddy currents is not as great as the resolution of ultrasonics or other techniques which make use of focused transducers. However, it has a very good depth resolution even when measuring

through multiple layers: A sub-millimetric space can be detected quite easily.

In particular, in ECT-NDE, when a defect is present on the surface of the specimen, to prevent the evolution of the damage, it is important to detect, localize and size the crack; however, the eddy-current measurement is corrupted by the skin effect, the lift-off noise and uncorrelated noise [39]. Prior to perform a flaw detection/recognition processing, each measure has thus to be restored, by separately featuring the lift-off signal and the defect signal. The magnitude and the phase of the complex-valued voltage-type ECT signals, acquired on the upper and lower sides of the specimen, have been considered as available measures. In order to extract information from the measured data, a proper signal processing algorithm should be designed.

An ECT-NDE data processing approach is described in the following to remove the effects of the eddy-current sensor drift during the horizontal/vertical scanning of an inspected metallic slab.

4.2.1 Measures description and data model

We analyze a set of experimental ECT-NDE data, provided by the Hungarian Academy of Sciences [32]. They have been acquired by a single 'pancake' exciting coil with FLUXSET sensor (for a detailed explanation of experimental set-up see [32]). The tested specimen consists of a square plate ($8 \times 8 \times 0.125$ cm) of INCONEL material, which presents a rectangular thin crack (about 0.2 mm thick and 9 mm in length), surface or hidden according to the inspection side, located in a region of 2×2 mm width around the plate center. The depth of the defect is about 20% of the plate thickness. The scanned area is a region of 40×40 mm with 0.5 mm spacing along x and y axes; the output voltage is proportional to the absolute value of the y component of the magnetic flux density, and the voltage magnitude and phase have been recorded on a grid of 81×81 measurement points.

Figure 7 represents a real- and imaginary-part pairs among the ECT-NDE signals. Even if the plate has a constant horizontal thickness, the signal has a magnitude related not only to the defect but also to the variable sensor lift-off over the specimen surface, which creates a drift effect on the measurements.

From the experimental set-up, four different measurements are available: The magnitude and the phase of the ECT signal acquired from the side 'A' of plate, and the magnitude and the phase from the side 'B' of the plate. If a defect is present near one of the sides, a measure can be considered as A and the second measure will be B type.

By using a single measurement, the detection of the crack is unfeasible because of two concurring problems:

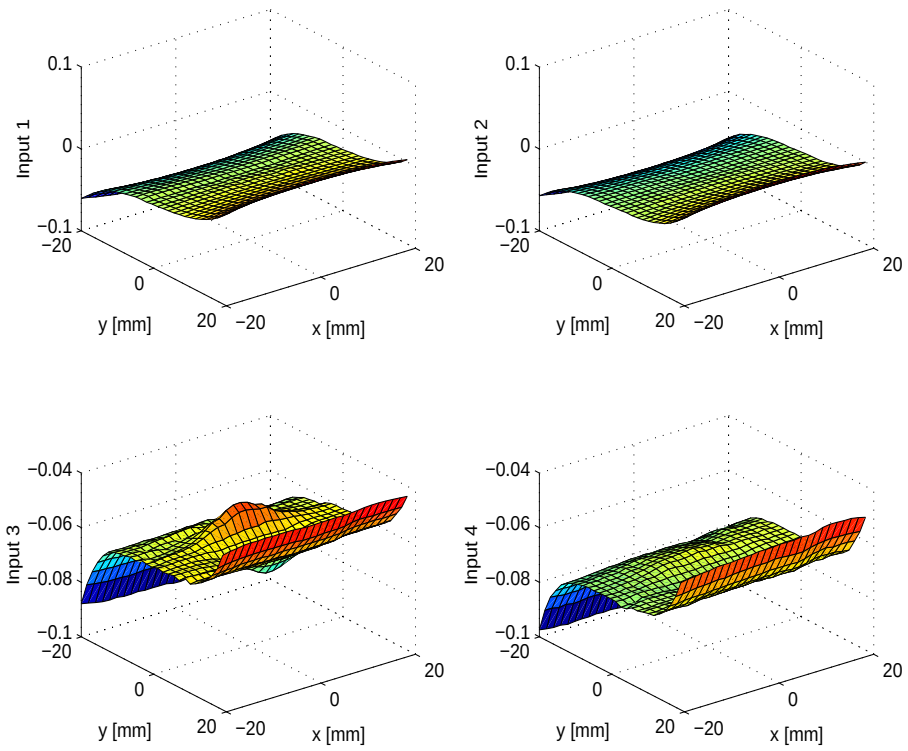

Figure 7: Real- and imaginary-part of the measured ECT-NDE signal (Inputs 1 and 3: measure A; Inputs 2 and 4: measure B.

- When the defect is located near the surface on the same side of the sensor (measure A), although the signal-to-noise ratio is high, it does not suffice to provide the detection/recognition system a suitable knowledge to correctly locate and size the flaw;
- When the defect is located near the surface on the other side of the sensor (measure B), the signal related to the crack is completely buried into background disturbance, due to the skin effect, to the lift-off noise and to the uncorrelated Gaussian noise, as can be readily seen from Figure 7.

Our working hypothesis is that the measured signals are linear mixtures of different sources: The signal related to the defect and the one related to the lift-off noise. This suggests that, on the basis of ICA technique, a way can be envisaged to extract the defect signal. More formally, we hypothesize that there exist two latent signals whose linear superposition with proper (unknown) weights give rise to the observed signals. In our model such latent variables are statistically independent, thus they may be separated through an ICA technique.

As mentioned, we hypothesize a linear model relating the independent signals

with the measures. The measured-data models is thus the one hypothesized in section 3, that is:

$$\mathbf{z} = \mathbf{M}^T \mathbf{s} \ .$$

Our proposal for processing the available data is to suppose that the real and imaginary parts of the involved signals interact in an additive way, thus the network input signal contains 4 scalar sub-signals: The real- and imaginary-part of both the A and B measures.

4.2.2 Data pre-processing

The signal $\mathbf{z}$ gets first whitened by mean-value removal and eigenvalue-decomposition based normalization of the covariance matrix, so that the whitened data are centered and have unitary covariance.

However, in the present context we further need a pre-compression stage, in that the available signals are four, while the presently studied separating network only allows three-dimensional signal processing. Pre-compression may be achieved by eigenvalue decomposition of measures' covariance matrix and by computing projections of 4-dimensional data onto a data-subspace of dimension 3.

The eigenvalues of the 4×4 covariance matrix of signal $\mathbf{z}$ are: 4.89×10^{-7}, 1.81×10^{-6}, 1.76×10^{-5} and 7.81×10^{-4}. The criterion that drove us to the most meaningful choice of signal subspace is the observation that the defect signals clearly appear to be less powerful than the lift-off disturbance to be filtered out, thus we concluded that the optimal compression, in this case, attains by projecting the data-samples over the eigenvectors corresponding to the three smallest eigenvalues.

Also, as the signal shows a non-negligible spatial correlation, the one-unit ICA algorithm is run over $15,000$ samples randomly picked (with replacement) from the set of $81 \times 81 = 6561$ available measures (regardless of their ordering), to simulate stationarity[4].

4.2.3 Experimental results

In this experiment we tested the behavior of MEC_2 algorithm on NDE measured data.

As already mentioned, in the present experiment it is not possible to define an objective separation algorithm performance index, thus we are left with the analysis of the network's kinetic energy $K(t)$ that measures the network internal state-transition speed.

[4]Note that this way of using the signals is data-structure preserving.

Consequently, in order to find a pair of parameters (k_C, μ) that ensures fast convergence and stability of the reached steady-state, we performed several experiments and monitored the course of kinetic energy $K(t)$. The Figure 8 shows the best course found, in terms of convergence speed and steady-state stability, which corresponds to the pair $k_C = -0.2$ and $\mu = 1.1$ ($\theta = 0.001$ as in the previous experiments).

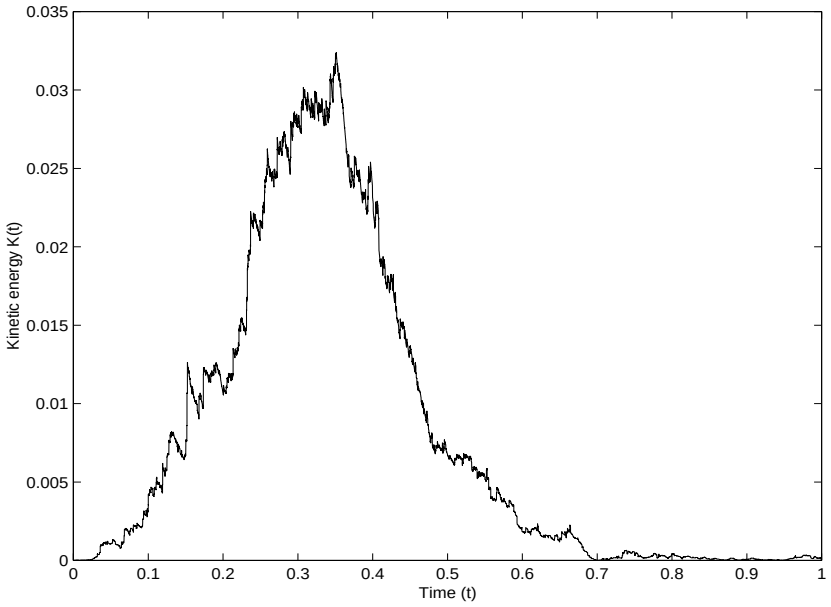

Figure 8: Course of the kinetic energy $K(t)$ for learning-parameter validation on NDE signals.

After the learning-parameters validation step, some simulations were performed in order to monitor the separation-ability behavior of the MEC_2 neural network. Figure 9 illustrates the three extracted components (shown in two different slants). It is readily seen that the components 1 and 3 are pure lift-off noises, while the component 2 is a cleaned surface where the contribution owing to the defect has been consistently emphasized, as expected.

5 Conclusion

In previous papers we presented a new class of learning rules for linear neural network learning based on the equations describing the dynamics of massive rigid bodies of dimension p, whose main drawback was the need to repeatedly evaluate a matrix exponential map. With the aim to lessen the computational burden pertaining to these algorithms, we proposed here an improvement, discussed for

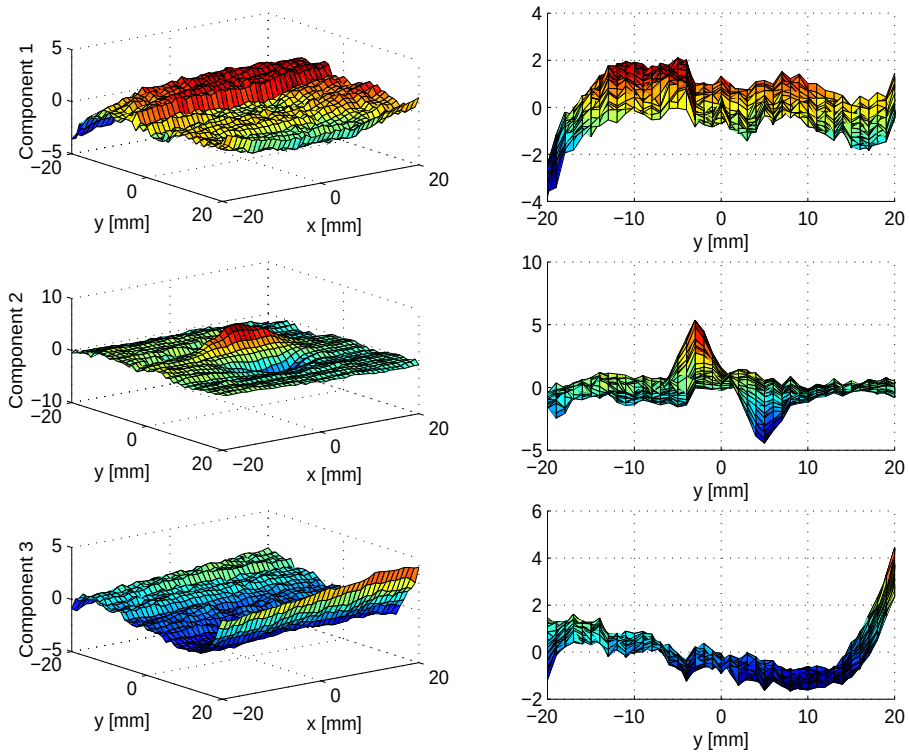

Figure 9: Three extracted components from NDE signals (Left: Component surface; Right: Projection over the horizontal axis).

the case $p = 3$ only, based on the Rodriguez formula for matrix exponential map computation.

The features of the new class of algorithms have been discussed by tackling cases of blind source separation by the independent component analysis and by discussing experimental results carried out with both synthetic signal and real-world non-destructive-evaluation data.

Both separation performance and computational-complexity evaluations and comparison reveal that the proposed theory provide a good trade-off between performances and computational burden.

We plan to extend the present theory to the general case of a $p \times m$ neural network. This would, however, require the extension of the Rodriguez formula to the case of Lie-group learning on the compact Stiefel manifold $\mathrm{St}(p, m, \mathbb{R})$. Such matrix-analysis solution is to be looked for in the field of numerical analysis of matrix differential equations on Lie groups [12].

Acknowledgments

We wish to gratefully thank P. Burrascano (University of Perugia, Italy) for some interesting comments on the content of this report, that helped clarifying some parts of it, and E. Celledoni and the members of the Geometric Integration group at the Center for Advanced Study in Olso, Norway, for the interesting and fruitful discussions with the author SF on Lie-group integration methods and the geometry of the Stiefel manifold. This research work has been carried out when the student RR followed the course of "Modellistica Elettrica dei Materiali" at the Faculty of Engineering of the Perugia University.

References

[1] S.-i. Amari, *Natural Gradient Works Efficiently in Learning*, Neural Computation, Vol. 10, pp. 251 – 276, 1998

[2] S.-I. Amari, A. Cichocki and H.H. Yang, *A new learning algorithm for blind signal separation*, Advances in Neural Information Processing Systems 8, MIT Press, 1993

[3] J. Angeles, *Rational Kinematics*, Springer Tracts in Natural Philosophy, Vol 34, 1989

[4] E. Bayro-Corrochano, *Geometric Neural Computation*, IEEE Trans. on Neural Networks, Vol. 12, No. 5, pp. 968 – 986, Sept. 2001

[5] E. Bayro-Corrochano and G. Sobczyk (Ed.s), *Advances in Geometric Algebra with Applications in Science and Engineering. Automatic theorem proving, computer vision, quantum and neural computing and robotics*, Birkhauser, New York, February 2001

[6] 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

[7] J. Blitz, *Electrical and Magnetic Methods of Non-Destructive Testing*, Chapman and Hall, 2nd Edition, 1997

[8] J.M. Bowler, *Review of Eddy Current Inversion with Application to Non-destructive Evaluation*, Int. Journal of Applied Electromagnetics and Mechanics, Vol. 8, pp. 3 – 16, 1997

[9] G.E. Bredon, *Topology and Geometry*, New-York: Springer-Verlag, 1995

[10] R.W. Brockett, *Dynamical Systems that Sort Lists, Diagonalize Matrices and Solve Linear Programming Problems*, Linear Algebra and Its Applications, Vol. 146, pp. 79 – 91, 1991

[11] J.F. Cardoso and B. Laheld, *Equivariant Adaptive Source Separation*, IEEE Trans. on Signal Processing, Vol. 44, No. 12, pp. 3017 – 3030, Dec. 1996

[12] E. Celledoni and B. Owren, *On the implementation of Lie group methods on the Stiefel manifold*, Preprint Numerics no. 9/2001, Norwegian University of Science and Technology, Trondheim (Norway), 2001

[13] P. Comon, *Independent Component Analysis, A New Concept ?*, Signal Processing, Vol. 36, pp. 287 – 314, 1994

[14] P. Comon and E. Moreau, *Improved Contrast Dedicated to Blind Separation in Communications*, Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3453 – 3456, 1997

[15] S. Costa and S. Fiori, *Image Compression Using Principal Component Neural Networks*, Image and Vision Computing Journal (special issue on "Artificial Neural Network for Image Analysis and Computer Vision"), Vol. 19, No. 9-10, pp. 649 – 668, Aug. 2001

[16] L. Dieci and E. Van Vleck, *Computation of orthonormal factors for fundamental solution matrices*, Numerical Mathematics, Vol. 83, pp. 599 – 620, 1999

[17] A. Edelman, T.A. Arias, and S.T. Smith, *The Geometry of Algorithms with Orthogonality Constraints*, SIAM Journal on Matrix Analysis Applications, Vol. 20, No. 2, pp. 303 – 353, 1998

[18] Y. Ephraim and L. Van Trees, *A Signal Subspace Approach for Speech Enhancement*, IEEE Trans. on Speech and Audio Processing, Vol. 3, No. 4, pp. 251 – 266, July 1995

[19] S. Fiori, *Entropy Optimization by the PFANN Network: Application to Independent Component Analysis*, Network: Computation in Neural Systems, Vol. 10, No. 2, pp. 171 – 186, May 1999

[20] S. Fiori, *Blind Separation of Circularly-Distributed Sources by Neural Extended APEX Algorithm*, Neurocomputing, Vol. 34, No. 1-4, pp. 239 – 252, Aug. 2000

[21] S. Fiori, *Blind Signal Processing by the Adaptive Activation Function Neurons*, Neural Networks, Vol. 13, No. 6, pp. 597 – 611, Aug. 2000

[22] S. Fiori, *A Theory for Learning by Weight Flow on Stiefel-Grassman Manifold*, Neural Computation, Vol. 13, No. 7, pp. 1625 – 1647, July 2001

[23] S. Fiori, *A Theory for Learning Based on Rigid Bodies Dynamics*, IEEE Trans. on Neural Networks, Vol. 13, No. 3, pp. 521 – 531, May 2002

[24] S. Fiori, *Unsupervised Neural Learning on Lie Group*, International Journal of Neural Systems, Vol. 12, No.s 3 & 4, pp. 219 – 246, 2002

[25] A. Fujiwara and S.-i. Amari, *Gradient systems in view of information geometry*, Physica D, Vol. 80, pp. 317 – 327, 1995

[26] K. Gao, M.O. Ahmed, and M.N. Swamy, *A Constrained Anti-Hebbian Learning Algorithm for Total Least-Squares Estimation with Applications to Adaptive FIR and IIR Filtering*, IEEE Trans. on Circuits and Systems II, Vol. 41, No. 11, pp. 718 – 729, Nov. 1994

[27] A. Hyvärinen and E. Oja, *Independent Component Analysis by General Non-Linear Hebbian-Like Rules*, Signal Processing, Vol. 64, No. 3, pp. 301 – 313, 1998

[28] J. Karhunen, *Neural Approaches to Independent Component Analysis and Source Separation*, Fourth European Symposium on Artificial Neural Networks (ESANN'96), pp. 249 – 266, 1996

[29] A. Kern, D. Blank, and R. Stoop, *An optimal noise cleaning by local manifold projection*, Proc. of Second International ICSC Symposium on Neural Computation (NC), pp. 399 – 404, 2000

[30] J. Kivinen and M. Warmuth, *Exponentiated gradient versus gradient descent for linear predictors*, Information and Computation, Vol. 132, pp. 1 – 64, 1997

[31] R.-w. Liu, *Blind Signal Processing: An Introduction*, Proc. of International Symposium on Circuits and Systems (IEEE-ISCAS), Vol. 2, pp. 81 – 84, 1996

[32] Manodet project, *Data pertaining to the 'Manodet Project', hosted by the Hungarian Academy of Sciences*. Mirrored on the web page http://neurolab.ing.unirc.it

[33] B.C. Moore *Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction*, IEEE Trans. on Automatic Control, Vol. AC-26, No. 1, pp. 17 – 31, 1981

[34] H. Niemann and J.-K. Wu, *Neural Network Adaptive Image Coding*, IEEE Trans. on Neural Networks, Vol. 4, No. 4, pp. 615 – 627, July 1993

[35] Y. Nishimori, *Learning Algorithm for ICA by Geodesic Flows on Orthogonal Group*, Proc. of the International Joint Conference on Neural Networks (IJCNN'99), Vol. 2, pp. 1625 – 1647, 1999

[36] E. Oja, *Neural Networks, Principal Components and Subspaces*, Int. Journal of Neural Systems, Vol. 1, pp. 61 – 68, 1989

[37] E. Oja, A. Hyvärinen, and P. Hoyer, *Image Feature Extraction and Denoising by Sparse Coding*, Pattern Analysis and Applications Journal, Vol. 2, Issue 2, pp. 104 – 110, 1999

[38] P. Saisan, G. Doretto, Y.N. Wu, and S. Soatto, *Dynamic texture recognition*, Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 58 – 63, Dec. 2001

[39] G. Simone and F.C. Morabito, *ICA-NN Based Data Fusion Approach in ECT Signal Restoration*, Proc. International Joint Conference on Neural Networks, Vol. V, pp. 59 – 64, Como (Italy), July 2000

[40] D. Sona, A. Sperduti and A. Starita, *Discriminant Pattern Recognition Using Transformation Invariant Neurons*, Neural Computation, Vol. 12, No. 6, pp. 1355 – 1370, June 2000

[41] L. Xu, E. Oja, and C.Y. Suen, *Modified Hebbian learning for curve and surface fitting*, Neural Networks, Vol.5, pp. 393 – 407, 1992

[42] B. Yang, *Projection Approximation Subspace Tracking*, IEEE Transaction on Signal Processing, Vol. 43, No. 1, pp. 1247 – 1252, Jan. 1995

[43] H.H. Yang and S.-i. Amari, *Adaptive online learning algorithms for blind separation: maximum entropy and minimal mutual information*, Neural Computation, Vol. 9, pp. 1457 – 1482, 1997

[44] K. Zhang and T.J. Sejnowski, *A theory of geometric constraints on neural activity for natural three-dimensional movement*, Journal of Neuroscience, Vol. 19, No. 8, pp. 3122 – 3145, 1999