

# A Theory for Learning Based on Rigid Bodies Dynamics

Simone Fiori

**Abstract**—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, is presented. The set of equations describing system's dynamics may be directly interpreted as a learning algorithm for neural layers. Relevant properties of the proposed learning theory are discussed within the paper, along with results of computer simulations performed in order to assess its effectiveness in applied fields.

**Index Terms**—Matrix-flows on Stiefel-manifold, orthonormal signal processing, principal/independent component analysis, rational mechanics, unsupervised neural learning.

## I. INTRODUCTION

IN THE past, researchers involved in artificial neural networks have been attracted by the fashion of physical sciences and have developed many learning paradigms inspired by a variety of thermodynamic, mechanical, biological, and electrical parallelisms. Such works yielded very good and elegant theoretical and practical results, such as the well-known Hopfield's associative memory [45], the Boltzmann machine [34], the Helmholtz machine [14], the simulated annealing [30] optimization technique, the "neural-gas" networks [33], the "information particles" theory and the statistical mechanics of mutual information [38], [44], and the neuro-evolutionary algorithms [10]. By taking advantage of this way of thinking, new approaches to solve some well-known but difficult problems can of course be developed.

In this paper, we present a new learning theory whose derivation is based on the analysis of the dynamics of a rigid body moving in an abstract space: The set of equations describing the dynamics of such a system may be directly interpreted as a learning algorithm for linear as well as nonlinear neural layers. We discuss the existing relationship between the structure of dynamical equations and the tasks that can be performed by means of the associated neural system, with special reference to the neural solution of "orthonormal problems."

The orthonormal problems (ONPs) arise in several contexts, as for instance in optimal linear data compression (principal component/subspace analysis, [16], [24], [37], [39]), whitening [12], blind separation of sources [5], [11], [12], [19], [29], direction of arrival (DOA) estimation [12], [41], in feature transformation learning applied to unsupervised classification [43], in image processing [35] and digital filter design [27]. In an ONP,

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. Hence

- 1) the searching space is considerably reduced; in fact, the set of matrices belonging to  $\mathcal{R}^{p \times m}$ , 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;
- 2) nonorthonormal local (suboptimal) solutions are inherently avoided as they do not belong to the search-space;
- 3) the searching algorithm may be geodesic: The space of the orthonormal matrices is endowed with a specific geometry and a geodesic connecting two points, which is the shortest pathway between them, may be defined. A geodesic algorithm follows the geodesics between any pair of searching steps, thus providing the best search-path.

We solve 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 if it initially was.

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*.

The analysis of the motion of a rigid body is the theoretical basis of this paper. It is developed in Section II, along with some considerations on the relationship between the structure of the forcing field and the tasks that the network can perform. In support of the mentioned theory, computer simulations about the application of the proposed algorithm to principal component analysis (PCA) and blind separation of sources by the independent component analysis (ICA) are presented and discussed in Section III. Section IV concludes the paper.

## II. DYNAMICS OF AN ABSTRACT RIGID BODY IN $\mathcal{R}^p$

Let  $\mathcal{S}_m = \{(m_i, \mathbf{w}_i), (m_i, -\mathbf{w}_i)\}_{i \in \{1, \dots, m\}}$  be a rigid system of masses, where the  $m$  vectors  $\mathbf{w}_i \in \mathcal{R}^p$  represent the instantaneous positions of  $2m$  masses  $m_i \in \mathcal{R}_0^+$  in a coordinate system. Such masses are positioned at constant (unitary) distances from the origin  $\mathcal{O}$  fixed in the space  $\mathcal{R}^p$ ,

Manuscript received October 28, 1999; revised November 27, 2000.

The author is with the Neural Networks and Adaptive System Research Group, Department of Industrial Engineering, University of Perugia, I-60026 Marcellini di Numana, Italy (e-mail: sfr@unipg.it).

Publisher Item Identifier S 1045-9227(02)00358-2.

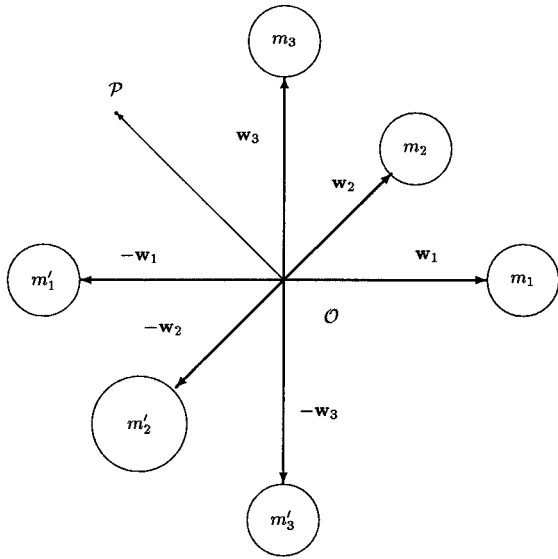


Fig. 1. A configuration of  $S_m$  for  $p = 3$  and  $m = 3$ .

and over mutually orthogonal immaterial axes. In this paper we assume the values of the masses  $m_i$  constant at 1. In Fig. 1 an exemplary configuration of  $S_m$  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.

Masses move in the space  $\mathcal{R}^p$  where a physical point  $\mathcal{P}$ , endowed with a negligible mass, moves too; its position with respect to  $\mathcal{O}$  is described by an independent vector  $\mathbf{x}$ . 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  $S_m$ . Furthermore, masses move in an homogeneous and hysotropic fluid endowed with a nonnegligible viscosity: The corresponding resistance brakes the motion, makes the system dissipative and stabilizes its dynamics.

#### A. Equations Describing the Dynamics of the System

The first part of the work consists in the research of the equations describing the dynamics of the mechanical system. Such a work is summarized in the following result.

*Theorem 1 (Dynamics of the System  $S_m$ ):* Let  $S_m \subset \mathcal{R}_0^+ \times \mathcal{R}^p$  be the physical 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  $\mathbf{B}$  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:

$$\frac{d\mathbf{W}}{dt} = \mathbf{B}\mathbf{W} \quad (1)$$

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

$$\mathbf{R} = -\mu\mathbf{B}\mathbf{W} \quad (3)$$

with  $\mu$  being a positive parameter termed *viscosity coefficient*.

*Proof:* To ensure the rigidity of the system at any time it is sufficient to require that the evolution of each position-vector is described by the equation

$$\frac{d\mathbf{w}_i(t)}{dt} = \mathbf{B}(t)\mathbf{w}_i(t) \quad (4)$$

for  $i \in \{1, \dots, m\}$ , with  $\mathbf{B}^T = -\mathbf{B}$ . To confirm this statement, we need to show that  $d[\mathbf{w}_i^T(t)\mathbf{w}_k(t)]/dt = 0$  at any time, provided that it was  $\mathbf{w}_i^T(0)\mathbf{w}_k(0) = \delta_{ik}$ , where  $\delta_{ik}$  is the Kronecker's "delta," because in this way it will be  $\mathbf{w}_i^T(t)\mathbf{w}_k(t) = \delta_{ik}$  at any time. To this aim, let us evaluate the first derivative of a generic scalar product

$$\frac{d(\mathbf{w}_i^T \mathbf{w}_k)}{dt} = \frac{d\mathbf{w}_i^T}{dt} \mathbf{w}_k + \mathbf{w}_i^T \frac{d\mathbf{w}_k}{dt}.$$

Replacing the derivatives with the terms at the right hand-side of (4) yields

$$\frac{d(\mathbf{w}_i^T \mathbf{w}_k)}{dt} = \mathbf{w}_i^T (\mathbf{B}^T + \mathbf{B}) \mathbf{w}_k = 0$$

because of the skew-symmetry of  $\mathbf{B}$ , as expected. If a matrix  $\mathbf{W}$  is defined such that  $\mathbf{W} \stackrel{\text{def}}{=} [\mathbf{w}_1 \ \mathbf{w}_2 \ \dots \ \mathbf{w}_m]$ , the set of equations (4) can be rewritten in a compact way as in (1).

Now, the instantaneous acceleration  $\mathbf{a}_i$  of each mass can be obtained by the following pair of equations:

$$\begin{aligned} m_i \mathbf{a}_i &= \mathbf{f}_i^{(+)} + \mathbf{g}_i^{(+)} + \mathbf{r}_i^{(+)} \\ -m_i \mathbf{a}_i &= \mathbf{f}_i^{(-)} + \mathbf{g}_i^{(-)} + \mathbf{r}_i^{(-)} \end{aligned}$$

where  $\mathbf{f}_i$  denotes the active force exerted by the point  $\mathcal{P}$  on mass  $m_i$ ,  $\mathbf{g}_i$  denotes the resultant of the rigidity constraints and of the eventual internal forces on the mass  $m_i$  and  $\mathbf{r}_i$  represents the viscosity resistance exerted by the fluid on the same mass. The superscript (+) denotes a force applied on mass  $m_i$ , while the superscript (-) denotes a force applied on the "ghost" of  $m_i$  in  $-\mathbf{w}_i$ .

It was assumed  $m_i = 1$ , thus subtracting hand-by-hand the second equation from the first one yields

$$2\mathbf{a}_i = \mathbf{f}_i + \mathbf{g}_i + \mathbf{r}_i \quad (5)$$

where  $\mathbf{f}_i \stackrel{\text{def}}{=} \mathbf{f}_i^{(+)} - \mathbf{f}_i^{(-)}$ ,  $\mathbf{g}_i \stackrel{\text{def}}{=} \mathbf{g}_i^{(+)} - \mathbf{g}_i^{(-)}$ ,  $\mathbf{r}_i \stackrel{\text{def}}{=} \mathbf{r}_i^{(+)} - \mathbf{r}_i^{(-)}$ . The instantaneous acceleration  $\mathbf{a}_i$  can be expressed in terms of  $\mathbf{B}$  and  $\mathbf{w}_i$  by differentiating each of the equations in (4), obtaining

$$\mathbf{a}_i = \frac{d}{dt} \left[ \frac{d\mathbf{w}_i}{dt} \right] = \left[ \frac{d\mathbf{B}}{dt} + \mathbf{B}^2 \right] \mathbf{w}_i. \quad (6)$$

Plugging (6) into (5) yields

$$2 \left[ \frac{d\mathbf{B}}{dt} + \mathbf{B}^2 \right] \mathbf{w}_i = \mathbf{f}_i + \mathbf{g}_i + \mathbf{r}_i. \quad (7)$$

The set of the equations (7) for  $i \in \{1, \dots, m\}$  can be rewritten in a compact way by introducing the following matrices:

$$\begin{aligned} \mathbf{F} &\stackrel{\text{def}}{=} [\mathbf{f}_1 \ \mathbf{f}_2 \ \dots \ \mathbf{f}_m], & \mathbf{G} &\stackrel{\text{def}}{=} [\mathbf{g}_1 \ \mathbf{g}_2 \ \dots \ \mathbf{g}_m] \\ \mathbf{R} &\stackrel{\text{def}}{=} [\mathbf{r}_1 \ \mathbf{r}_2 \ \dots \ \mathbf{r}_m]. \end{aligned}$$

Then, the matrix expression of (7) writes

$$2 \left[ \frac{d\mathbf{B}}{dt} + \mathbf{B}^2 \right] \mathbf{W} = \mathbf{F} + \mathbf{G} + \mathbf{R}. \quad (8)$$

Let us now look for a solution of the differential equation (8) for  $\dot{\mathbf{B}}$ . To this end, it is useful to introduce an adjoint system  $\tilde{\mathcal{S}}_m$  by completing the base provided by the columns of  $\mathbf{W}$  in order to obtain a full orthonormal base of  $\mathcal{R}^p$ , i.e., a matrix  $\tilde{\mathbf{W}} = [\mathbf{W} | \mathbf{W}_1]$  such that  $\tilde{\mathbf{W}}^T \tilde{\mathbf{W}} = \mathbf{I}_p$ . With respect to  $\mathcal{S}_m$ , the new system  $\tilde{\mathcal{S}}_m$  is described by the following matrices:

$$\tilde{\mathbf{F}} \stackrel{\text{def}}{=} [\mathbf{F} | \mathbf{0}], \quad \tilde{\mathbf{G}} \stackrel{\text{def}}{=} [\mathbf{G} | \mathbf{G}_1], \quad \tilde{\mathbf{R}} \stackrel{\text{def}}{=} [\mathbf{R} | \mathbf{0}]$$

where  $\tilde{\mathbf{F}}$ ,  $\tilde{\mathbf{G}}$  and  $\tilde{\mathbf{R}}$  belong to  $\mathcal{R}^{p \times p}$ . Clearly  $\tilde{\mathcal{S}}_m$  is fully equivalent to  $\mathcal{S}_m$ . The dynamics of  $\tilde{\mathcal{S}}_m$  is governed by

$$2 \left[ \frac{d\tilde{\mathbf{B}}}{dt} + \tilde{\mathbf{B}}^2 \right] \tilde{\mathbf{W}} = \tilde{\mathbf{F}} + \tilde{\mathbf{G}} + \tilde{\mathbf{R}} \quad (9)$$

that is the counterpart of (8) for the augmented system. Now, since  $\tilde{\mathbf{W}}$  is full column-rank, surely (9) admits the solution  $\dot{\tilde{\mathbf{B}}} + \tilde{\mathbf{B}}^2 = \tilde{\mathbf{X}} \tilde{\mathbf{W}}^T$ , with  $\tilde{\mathbf{X}} = [\mathbf{X} | \mathbf{X}_1]$  being a  $p \times p$  unknown matrix to be determined. Substituting in (9) gives

$$2\tilde{\mathbf{X}} \left( \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} \right) = 2\tilde{\mathbf{X}} = \tilde{\mathbf{F}} + \tilde{\mathbf{G}} + \tilde{\mathbf{R}}.$$

The last equality implies that

$$\mathbf{X}\mathbf{W}^T = \frac{1}{2} (\mathbf{F} + \mathbf{G} + \mathbf{R}) \mathbf{W}^T, \quad \mathbf{X}_1 \mathbf{W}_1^T = \frac{1}{2} \mathbf{G}_1 \mathbf{W}_1^T$$

hence

$$\begin{aligned} \dot{\mathbf{B}} + \mathbf{B}^2 &= \tilde{\mathbf{X}} \tilde{\mathbf{W}}^T = \mathbf{X}\mathbf{W}^T + \mathbf{X}_1 \mathbf{W}_1^T \\ &= \frac{1}{2} [(\mathbf{F} + \mathbf{G} + \mathbf{R}) \mathbf{W}^T + \mathbf{G}_1 \mathbf{W}_1^T]. \end{aligned} \quad (10)$$

The skew-symmetry constraint by transposing both members of (10). Subtracting hand-by-hand the transposed equation from (10) yields

$$2 \frac{d\mathbf{B}}{dt} = \frac{1}{2} (\mathbf{F} + \mathbf{R}) \mathbf{W}^T - \frac{1}{2} \mathbf{W} (\mathbf{F} + \mathbf{R})^T + \frac{1}{2} \tilde{\mathbf{G}} \tilde{\mathbf{W}}^T - \frac{1}{2} \tilde{\mathbf{W}} \tilde{\mathbf{G}}^T. \quad (11)$$

Since the rigidity of the system is ensured by the skew-symmetry of the matrix  $\mathbf{B}$  and since the internal forces cannot produce motion, of course it must hold

$$\tilde{\mathbf{G}} \tilde{\mathbf{W}}^T - \tilde{\mathbf{W}} \tilde{\mathbf{G}}^T = \mathbf{0}. \quad (12)$$

It remains to determine the structure of  $\mathbf{R}$ . By definition  $\mathbf{r}_i^{(+)}$  is a viscosity resistance, therefore it can be assumed equal to  $\mathbf{r}_i^{(+)} = -(1/2)\mu(d\mathbf{w}_i/dt)$ , where  $\mu$  is a positive parameter termed *viscosity coefficient*. Then it holds true that

$$\mathbf{r}_i^{(+)} = -\frac{1}{2}\mu\mathbf{B}\mathbf{w}_i, \quad \mathbf{r}_i^{(-)} = +\frac{1}{2}\mu\mathbf{B}\mathbf{w}_i$$

hence  $\mathbf{r}_i = -\mu\mathbf{B}\mathbf{w}_i$ , and the matrix  $\mathbf{R}$  has the expression  $\mathbf{R} = -\mu\mathbf{B}\mathbf{W}$ , as in (3). ■

The main result just proven states that equations (1)–(3) describe a “rigid” evolution of the matrix  $\mathbf{W}$  in  $\mathcal{R}^{p \times m}$ : If  $\mathbf{W}(0)$  is orthonormal then  $\mathbf{W}(t)$  keeps always orthonormal.

## B. MEC Network and Potential Energy Structure

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  $\mathbf{x} \in \mathcal{R}^p$ ,  $\mathbf{W}$  is  $p \times m$ , with  $m < p$ ,  $\mathbf{w}_0$  is a generic biasing vector in  $\mathcal{R}^m$  and  $\mathbf{S}[\cdot]$  is an arbitrarily-chosen  $m \times m$  diagonal activation operator.

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 potential energy function (PEF)  $U$ , which yields forces

$$\mathbf{f}_i^{(+)} = -\frac{\partial U}{\partial \mathbf{w}_i}, \quad \mathbf{f}_i^{(-)} = -\frac{\partial U}{\partial (-\mathbf{w}_i)}$$

hence by definition of matrix  $\mathbf{F}$

$$\mathbf{F} \stackrel{\text{def}}{=} -2 \frac{\partial U}{\partial \mathbf{W}}. \quad (13)$$

Generally we may suppose  $U$  dependent upon  $\mathbf{W}$ ,  $\mathbf{w}_0$ , and on the statistics of  $\mathbf{x}$ ; more formally  $U = E_{\mathbf{x}}[u(\mathbf{W}, \mathbf{w}_0, \mathbf{x}, \mathbf{y})]$ , 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.

When the optimization criterion driving network’s learning has been specified, we can assume  $U \stackrel{\text{def}}{=} J_c$ , with  $J_c$  being a cost function to be minimized, or  $U \stackrel{\text{def}}{=} -J_o$ , where  $J_o$  is an objective function to be maximized, both *under the constraint of orthonormality*. Vector  $\mathbf{w}_0$  may be arbitrarily adapted.

## C. Discussion on the MEC Learning Theory and Computer Implementation Issues

Equations (1) and (2) and their derivation clearly show that the MEC learning model also is a *second-order optimization* method. In other terms, the MEC rule behaves as a nonconventional optimization algorithm. Interesting observations are the following.

- 1) For the same initial point  $\mathbf{W}(0)$  different choices of  $\mathbf{B}(0)$  facilitate additional control of the expected solution, therefore second-order learning provides more degrees of freedom with respect to first-order one [1], [9].
- 2) The well-known backpropagation algorithm with the addition of a *momentum term*, which exhibits good local minima avoidance property, describes, in the continuous-time limit, the dynamics of an inertial particle subject to viscous drag (see, for instance, [6]), whose dynamical equation closely resembles (1)–(3).

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 present a simple 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  $d\mathbf{B}/dt$  with  $\Delta\mathbf{B}/\theta$ , where  $1/\theta$  is the ‘‘sampling rate.’’ In this way  $\mathbf{B}$  keeps constant within each open temporal interval  $]n\theta, (n+1)\theta[$ , and equals  $\mathbf{B}_n \stackrel{\text{def}}{=} \mathbf{B}(n\theta)$ , then equation (1) can be solved exactly within each of these intervals, in that it can be rewritten for  $n \geq 0$  as:

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

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

$$\mathbf{W}((n+1)\theta) = \exp(\mathbf{B}_n \theta) \mathbf{W}(n\theta). \quad (14)$$

By a computational viewpoint, the exponential matrix  $\exp(\mathbf{B}_n \theta)$  is not easily computable; we use the approximation  $\exp(\mathbf{B}_n \theta) \cong \sum_{i=0}^r (\mathbf{B}_n^i / i!) \theta^i$  with  $r$  being a nonnegative bounded integer. This way, the following set of learning equations is obtained:

$$\Delta\mathbf{W} = \left[ \sum_{i=1}^r \frac{\mathbf{B}_n^i}{i!} \theta^i \right] \mathbf{W}, \quad \mathbf{R} = -\mu \mathbf{B} \mathbf{W} \quad (15)$$

$$\Delta\mathbf{B} = \frac{\theta}{4} [(\mathbf{F} + \mathbf{R}) \mathbf{W}^T - \mathbf{W}(\mathbf{F} + \mathbf{R})^T] \quad (16)$$

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

It is important to note that running the MEC learning algorithm requires selecting three parameters ( $\mu$ ,  $\theta$ ,  $r$ ) plus some other constants depending on the particular application dealt with. Some simple considerations may drive us while choosing the values of these parameters. The constant  $\theta$  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. The parameter  $r$  should be chosen as small as possible because the computation of terms like  $\mathbf{B}_n^k \theta^k / k!$  has a significant impact on the total computational complexity of the algorithm.

About parameter  $\mu$ , its effect opposites to rapid body movement by contrasting the forcing field. An interesting consequence is that it contrasts the natural tendency of the second-order system to generate oscillatory trajectories, and generally prevents the system from oscillating around the solutions. This allows keeping constant the learning step-size in order to make the network reactive to nonstationary signals.

#### D. Equilibrium and Convergence Issues

When the learning task for the MEC network is specified, i.e., the driving objective function  $U$  is analytically known, the matrix field  $\mathbf{F}$  can be computed and the algorithm may be used for training the network. From a theoretical viewpoint, it is possible to study the behavior of the learning system by means of two necessary results that allow for computing the equilibrium points of the learning equations and to ensure the system asymptotically converges to one of them.

*Theorem 2 (Equilibrium):* Let us consider the state  $\mathbf{W}_*$  that the angular speed  $\mathbf{B}_*$  and the forcing term  $\mathbf{F}_*$  correspond to. If it

simultaneously happens that  $\mathbf{B}_* = \mathbf{0}$  and  $\mathbf{F}_* \mathbf{W}_*^T - \mathbf{W}_* \mathbf{F}_*^T = \mathbf{0}$  then  $\mathbf{W}_*$  is a steady state for the system (1)–(3).

*Proof:* If the above two conditions are verified then from equations (1)–(3) it follows that  $\dot{\mathbf{W}} = \mathbf{0}$  and  $\dot{\mathbf{B}} = \mathbf{0}$ , thus system sticks. ■

It is interesting to note that the second condition mentioned in the theorem represents the natural requirement that the total mechanical momentum applied to the rigid body vanishes to zero.

Provided that the equilibrium points for the systems are known, the study of their stability may be carried out. In fact, it can be envisaged that as the forcing field is conservative and the system continuously loses its energy during motion owing to the viscosity resistance, the body may stop in an equilibrium configuration.

*Theorem 3 (Convergence):* Let  $U$  be a real-valued, bounded-below function of  $\mathbf{W} \in \mathcal{R}^{p \times q}$ , with a minimum in the orthonormal point  $\mathbf{W}_*$ . Then the equilibrium state  $\mathbf{W}_*$  of the system (1)–(3) is asymptotically stable.

*Proof:* By reasoning on the physical meaning of the quantities involved in the learning equations, it is possible to demonstrate that there exists a Lyapunov function for the system.

In fact, by defining the auxiliary functions  $\mathbf{V}(t) \stackrel{\text{def}}{=} \mathbf{B}(t) \mathbf{W}(t)$  and  $K(t) \stackrel{\text{def}}{=} (1/2) \text{tr}[\mathbf{V}^T(t) \mathbf{V}(t)]$ , energy balance equation [4] at time  $t$  reads

$$K(t) - K(0) = -[U(t) - U(0)] - \mu \int_0^t \text{tr}[\mathbf{V}^T(\theta) \mathbf{V}(\theta)] d\theta. \quad (17)$$

By hypothesis, function  $U$  is bounded below and presents a minimum  $U_* = U(\mathbf{W}_*)$ . Hence, a function  $Y(t)$  defined as

$$Y(t) \stackrel{\text{def}}{=} K(t) + U(t) - U_* \quad (18)$$

in a neighborhood of  $\mathbf{W}_*$  satisfies  $Y(t) \geq 0$ , and the equality holds when  $\mathbf{B} = \mathbf{0}$  and  $U = U_*$ , that is when  $\mathbf{W} = \mathbf{W}_*$ . Furthermore, from equations (18) and (17) function  $Y(t)$  satisfies

$$Y(t) = Y(0) - 2\mu \int_0^t K(\theta) d\theta.$$

Since  $K(t) \geq 0$ , the time-derivative of  $Y(t)$  enjoys the following property:

$$\frac{dY(t)}{dt} = -2\mu K(t) \leq 0 \quad (19)$$

where the equality holds when  $\mathbf{B} = \mathbf{0}$ . Thus  $Y(t)$  is positive definite in  $\mathbf{W}_*$ , and there  $\dot{Y}(t)$  is negative-definite, hence  $\mathbf{W}_*$  is an asymptotically stable steady state. ■

These results show the mechanical parallelism allows us to discover important features of the learning equations.

#### E. Notes on Learning by Geodesic Flow, Riemannian Metrics, and Natural Gradient

The concept of iterative optimization under the constraint of orthonormality has gained more and more theoretical and practical importance during the last years. At present, a number of

papers from the scientific literature covering this topic is available; among other relevant papers, interested readers can find mathematical surveys in [7], [18], [25].

About neural learning, two interesting contributions have recently appeared in the scientific literature concerning neural learning under orthonormality constraints: The Douglas–Kung rule [17], leading to the algorithm here referred to as DKA, and the theory proposed by Nishimori [36], whose algorithm is here referred to as NMA; they both rely on the concept of geodesic. Owing to its nature, the MEC equations also generate a geodesic flow on the manifold of orthonormal matrices [22].

An interesting theoretical question is whether the MEC learning equations represent steepest directions or not. To answer, it is worth recalling that the ordinary gradient of an objective function gives the steepest direction only in the case of a Euclidean space; when the parameter space is endowed with an involved geometrical structure, the ordinary gradient does not necessarily provide a tangent direction; the actual gradient is the Riemannian (or natural) gradient [2], which MEC equations are a particular case of [22].

### III. COMPUTER SIMULATION RESULTS

In the following subsections, applications of the discretized MEC algorithm, expressed as in (15) and (16), are presented in support of the developed learning theory. The first experiments help gaining qualitative knowledge about the effects of the choice of mechanical constants on the dynamical behavior of learning equations. Applications deal with principal component analysis and blind source separation by the independent component analysis.

#### A. Qualitative Behavior of Learning Equations: A Case Study

The choice of “mechanical” learning constants (such as the viscosity coefficient and the magnifying factor for the forcing term) affects the dynamical behavior of the learning algorithm, thus a careful selection of the appropriate values is necessary.

In order to gain qualitative knowledge on this behavior, we propose the following case-study. Let us consider the MEC system with  $p = 2$  and  $m = 1$ . In this case the network is described by  $y = \mathbf{w}^T \mathbf{x}$ , with both the weight-vector  $\mathbf{w}$  and the input vector  $\mathbf{x}$  belonging to  $\mathcal{R}^2$ . The angular-speed matrix and positions vector may be thus parameterized as

$$\mathbf{B} = \begin{bmatrix} 0 & b \\ -b & 0 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \sin \phi \\ \cos \phi \end{bmatrix} \quad (20)$$

with  $b \in \mathcal{R}$  and  $\phi \in [0, 2\pi]$  usually referred to as *principal angle*.

Just to fix the ideas, let us suppose now input-stream  $\mathbf{x}(t)$  follows a Gaussian distribution with the covariance matrix having elements

$$\mathbf{C}_x = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}, \quad (\sigma_{12} = \sigma_{21})$$

and the mean value  $\mathbf{m} = \mathbf{0}$  for simplicity.

We now wish to simulate the extraction of the first principal component from  $\mathbf{x}$ , that may be obtained by means of the PEF

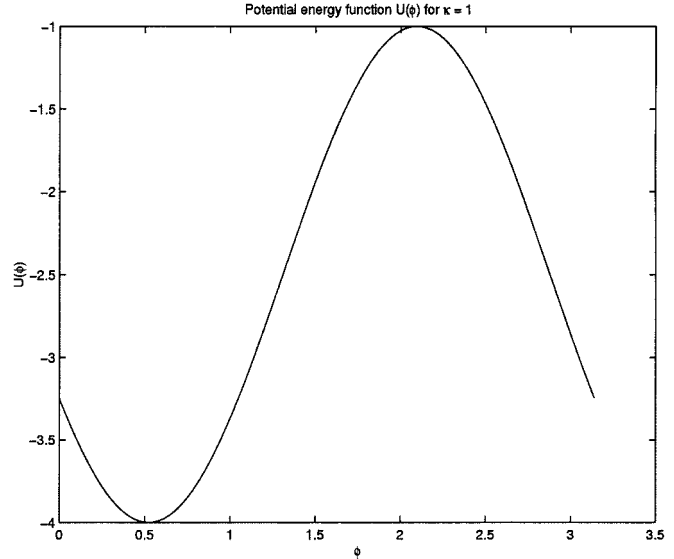


Fig. 2. A case study on first principal component extraction: Shape of potential energy function versus principal angle.

function  $U(\mathbf{w}) \stackrel{\text{def}}{=} -\kappa E_{\mathbf{x}}[y^2]$  [37], with  $\kappa > 0$  arbitrary. By definition, the forcing term in this case writes  $\mathbf{f} = 4\kappa \mathbf{C}_x \mathbf{w}$ . As a consequence, the MEC learning equations for the parameters  $b$  and  $\phi$  write

$$\begin{cases} \dot{b} = \frac{1}{4}[2\kappa(\sigma_{11} - \sigma_{11}) \sin(2\phi) + 4\kappa\sigma_{12} \cos(2\phi) - \mu b] \\ \dot{\phi} = b. \end{cases} \quad (21)$$

For a numerical example, the values of  $\sigma_{11}$ ,  $\sigma_{22}$  and  $\sigma_{12}$  were selected so that covariance matrix eigenvalues equal four and one and principal angle equals  $\phi_P = \pi/6$ .

First, let us verify that Theorem 2 ensures convergence to the expected solution. The equilibrium points of the system should be looked for among the solutions of the equation  $\mathbf{f}\mathbf{w}^T = \mathbf{w}\mathbf{f}^T$ , which in this case recasts into  $\mathbf{C}_x \mathbf{w} = \lambda(\mathbf{w})\mathbf{w}$ , with  $\lambda(\mathbf{w}) \stackrel{\text{def}}{=} \mathbf{w}^T \mathbf{C}_x \mathbf{w}$ ; clearly the solutions are the eigenvectors  $\mathbf{w}$  with eigenvalues  $\lambda(\mathbf{w})$  (irrespective of the value of constant  $\kappa$ ). It is worth noting that the potential energy function easily writes as  $U(\mathbf{w}) = -\kappa\lambda(\mathbf{w})$ , thus as  $\kappa$  is positive and  $U$  is minimized under the constraint that  $\|\mathbf{w}\|^2 = 1$ , the algorithm actually searches for the eigenvector corresponding to the largest eigenvalue. Theorem 3 ensures that the algorithm can find the absolute minimum, provided that  $\mathbf{w}(0)$  belongs to its basin of attraction; as can be noted from Fig. 2, which illustrates the shape of  $U$  as a function of parameter  $\phi$  (i.e., as actually “seen” by the MEC algorithm), in this case the PEF has a only minimum in  $\phi = \phi_P$ , thus Theorem 3 does ensure convergence *at large*.

In order to carry out the simulations, we had  $b(0)$  randomly selected in  $[-0.5, 0.5]$  and  $\phi(0) = 0$  rad.

The Fig. 3 refers to  $\mu = 5$  and three different values of  $\kappa$ , namely  $\kappa = 0.1, 1, 10$ ; it shows that growing the value of the magnifying factor may result in reducing the rise time but, as a counterpart, in growing the overshoot, that is the maximum error with respect to the asymptotic result  $\phi_P$ .

Fig. 4 refers to three values of the viscosity parameter, namely  $\mu = 1, 5, 10$ . According to the theoretical analysis, large values

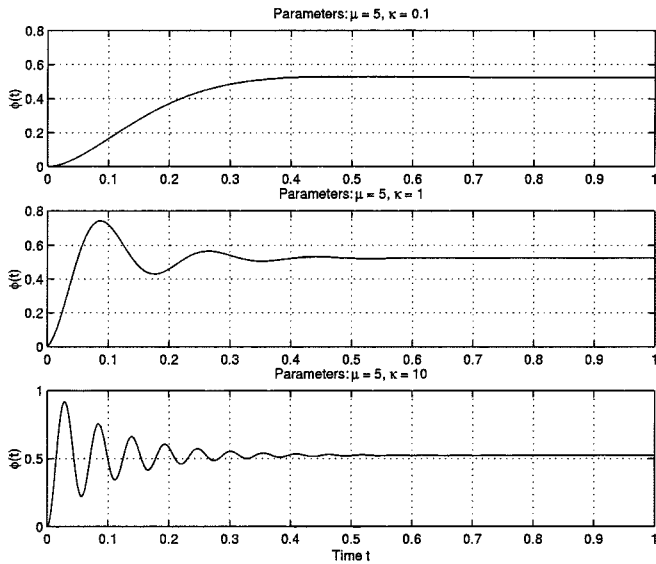


Fig. 3. A case study on first principal component extraction: Comparison of MEC behavior for different values of magnifying factor.

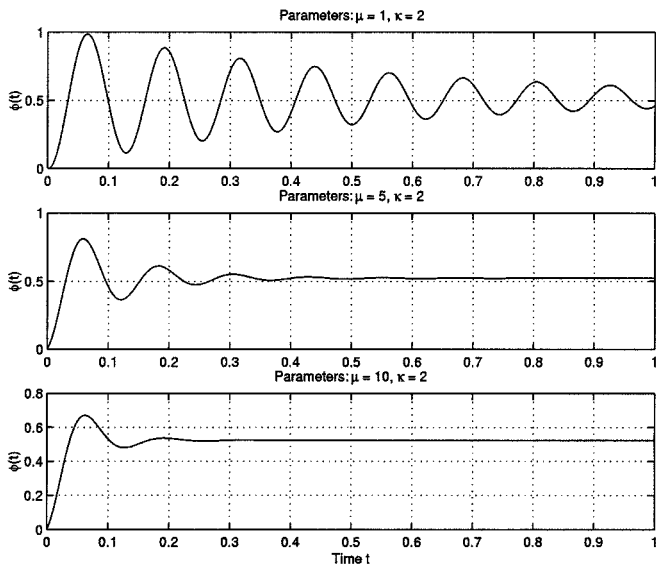


Fig. 4. A case study on first principal component extraction: Comparison of MEC behavior for different values of viscosity parameter.

of  $\mu$  provide better dissipation of system's energy, thus damping effect reduces oscillation magnitude.

### B. Application to Principal Component Analysis (PCA)

Principal component analysis (PCA) is a widely known statistical data processing technique which allows representing a multivariate zero-mean data stream  $\mathbf{x}(t) \in \mathcal{R}^p$  by means of a lower-dimensional data stream  $\mathbf{y}(t) \in \mathcal{R}^m$  obtained from the linear transformation  $\mathbf{y}(t) = \mathbf{W}^T \mathbf{x}(t)$ ; the PCA matrix  $\mathbf{W}$  performs data compression if its columns are the eigenvectors of the input stream covariance matrix corresponding to the  $m$  largest eigenvalues.

From [37] we know that PCA arises from the maximization of the power  $\Pi_{\mathbf{y}}$  of the transformed signal  $\mathbf{y} = \mathbf{W}^T \mathbf{x}$  defined as  $\Pi_{\mathbf{y}}(\mathbf{W}) \stackrel{\text{def}}{=} (1/2) \text{tr}(E_{\mathbf{x}}[\mathbf{y}\mathbf{y}^T])$ , therefore the objective function to be maximized under the constraint of orthonormality is

$J_{\circ} = (1/2) \sum_{i=1}^m a_i E_{\mathbf{x}}[y_i^2]$ , where constants  $a_1, \dots, a_m$  have been introduced to achieve an ordering: If  $a_1 > \dots > a_m$  then the signal  $y_1$  will contain the first principal part of  $\mathbf{x}$ ,  $y_2$  will contain the second one, and so on. Since  $J_{\circ}$  must be maximized, the PEF is assumed such that  $U \propto -J_{\circ}$ . More compactly we write  $U = -(1/2)k_H \text{tr}(E_{\mathbf{x}}[\mathbf{y}\mathbf{y}^T]\mathbf{A})$ , where  $\mathbf{A} = \text{diag}(a_1, \dots, a_m)$  and  $k_H$  is a positive scaling term. By definition of  $\mathbf{F}$ , the instantaneous stochastic estimation (ISE) of the force is found to be

$$\mathbf{F} = 2k_H \mathbf{x}\mathbf{y}^T \mathbf{A} \quad (22)$$

that we refer to as the Hebbianic force (H-Force) because it recalls the well-known Hebbianic adaptation term. Often a simple but effective choice is  $\mathbf{A} = \text{diag}(m, m-1, \dots, 1)$ .

*Experiment 1:* In this experiment we try to extract the first  $m < p$  principal components (PCs) from a signal  $\mathbf{x}$  of dimension  $p$  obtained by mixing the  $p$  components  $s_1, \dots, s_p$  of a random signal  $\mathbf{s}$ . Sub-signals  $s_i$  are random processes whose powers are  $\sigma_i^2$ , ordered in  $\mathbf{s}$  so that  $\sigma_i^2 > \sigma_j^2$  if  $i < j$ . Since  $\mathbf{x}$  is supposed to be obtained from  $\mathbf{s}$  by means of the linear model  $\mathbf{x} = \mathbf{Q}\mathbf{s}$ , with  $\mathbf{Q}$  being a randomly generated orthonormal matrix, the first  $m$  PC's of  $\mathbf{x}$  are, respectively, the first  $m$  columns of  $\mathbf{Q}$ .

In this first exploratory simulation, sources powers were drawn by the exponential law  $\sigma_i^2 = 2^{3-i}$ . Also, the following parameters' values were used:  $\mathbf{A} = \text{diag}(5, \dots, 1)$ ,  $r = 3$ ,  $\eta = 0.05$ ,  $\mu = 4$ ,  $k_H = 0.45$ .

Computer simulations were performed in order to compare performances of three PCA algorithms: Sanger's generalized Hebbian algorithm (GHA, [39]), Kung-Diamantaras' Adaptive Principal component EXtractor (APEX, [16]), and the MEC one with H-Force as forcing term. Both GHA and APEX algorithms require a learning stepsize parameter here denoted as  $\eta$ , while the MEC algorithm involves parameters  $\theta$ ,  $\mu$ ,  $r$ ,  $k_H$ , and  $\mathbf{A}$ . In order to give a fair comparison of the three algorithms, here we assumed  $\theta = \eta$ : This choice was motivated by the observation that  $\theta$  roughly plays the role of "learning rate" in equations (15) and (16). Network's input-output's dimensions are:  $p = 10$  and  $m = 5$ . In order to measure the convergence rate of the algorithms, a performance index  $\varepsilon(\mathbf{W})$  is defined as follows. Saying  $\mathbf{Q}_c$  the matrix whose  $m$  columns coincide with the first  $m$  columns of  $\mathbf{Q}$  and  $\mathbf{D} = \mathbf{W}^T \mathbf{Q}_c$ , the performance index is

$$\varepsilon(\mathbf{W}) \stackrel{\text{def}}{=} \left\| \mathbf{D}^{(2)} - \mathbf{I}_m \right\|_2 \quad (23)$$

in fact if components in  $\mathbf{W}^T \mathbf{Q}_c$  are ordered as in  $\mathbf{s}$ , then matrix  $\mathbf{D}$  should approach the diagonal form  $\text{diag}(\pm 1, \dots, \pm 1)$ . In definition (23) symbol  $\mathbf{D}^{(2)}$  denotes element-wise squaring.

Initial values for the GHA weight-matrix  $\mathbf{W}_1$  and the APEX direct-connections weight-matrix  $\mathbf{W}_2$  were randomly chosen within  $[-0.1, +0.1]$ , the initial values of the lateral-connections weight-matrix for the APEX algorithm was chosen null as well as the initial values of the angular-speed matrix  $\mathbf{B}$ , while the initial values for the MEC weight-matrix  $\mathbf{W}_3$  was  $\text{eye}(10, 5)$  so that  $\mathbf{W}_3^T \mathbf{W}_3 = \mathbf{I}_5$  when simulation starts.

In Fig. 5 performance indexes  $\varepsilon(\mathbf{W}_1)$ ,  $\varepsilon(\mathbf{W}_2)$  and  $\varepsilon(\mathbf{W}_3)$  are depicted. The MEC algorithm with the above parameter values

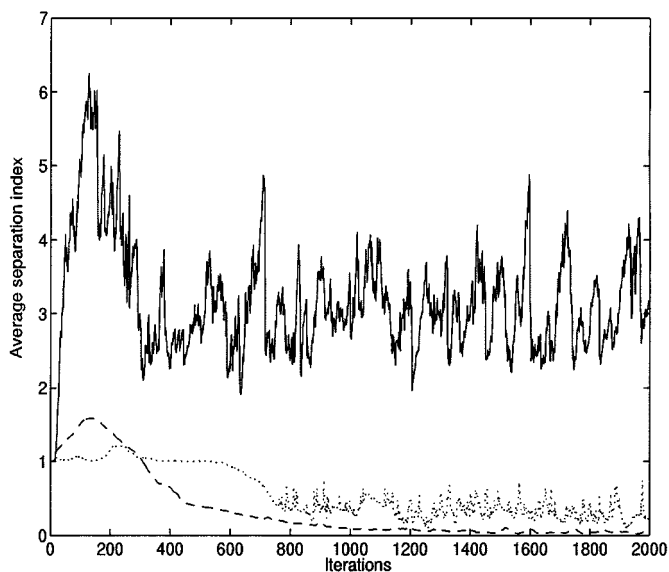


Fig. 5. Performance indexes  $\varepsilon(\mathbf{W})$  plotted versus iteration number. GHA: Dotted line, APEX: Solid line, MEC: Dashed line ( $\mu = 4$ ).

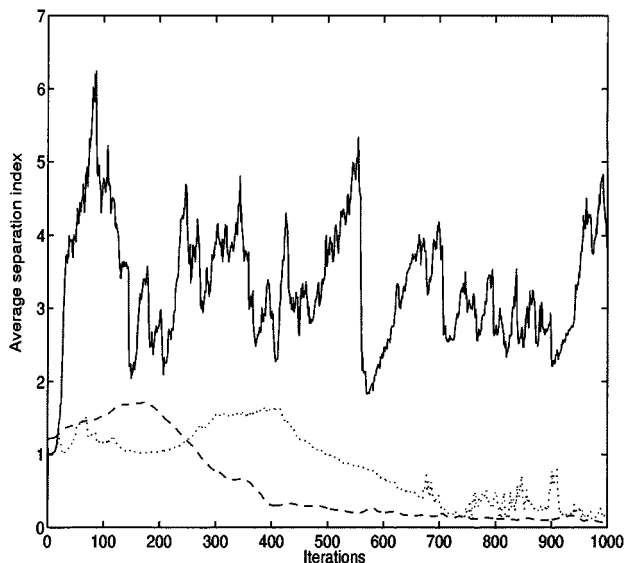


Fig. 6. Performance indexes  $\varepsilon(\mathbf{W})$  plotted versus iteration number. GHA: Dotted line, APEX: Solid line, MEC: Dashed line ( $\mu = 2.5$ ).

clearly exhibits best performances and superior PCs analysis behavior.

*Experiment 2:* In the second simulation, it was  $\mu = 2.5$ , while the other parameter values were unchanged.

Modifying the value of the viscosity coefficient results in speeding up the convergence of the MEC algorithm against the others, as can be seen in the Fig. 6, without altering the steady-state precision.

*Experiment 3:* In what follows we aim at briefly illustrating the performances of the Douglas–Kung algorithm, the Nishimori’s algorithm and the MEC one on a data-size reduction problem.

Results reported in the Fig. 7 concern principal component analysis extraction from a  $256 \times 256$  gray-levels natural image. Comparisons have been performed with Douglas–Kung DKA algorithm and Nishimori’s NMA one. The image was first sub-

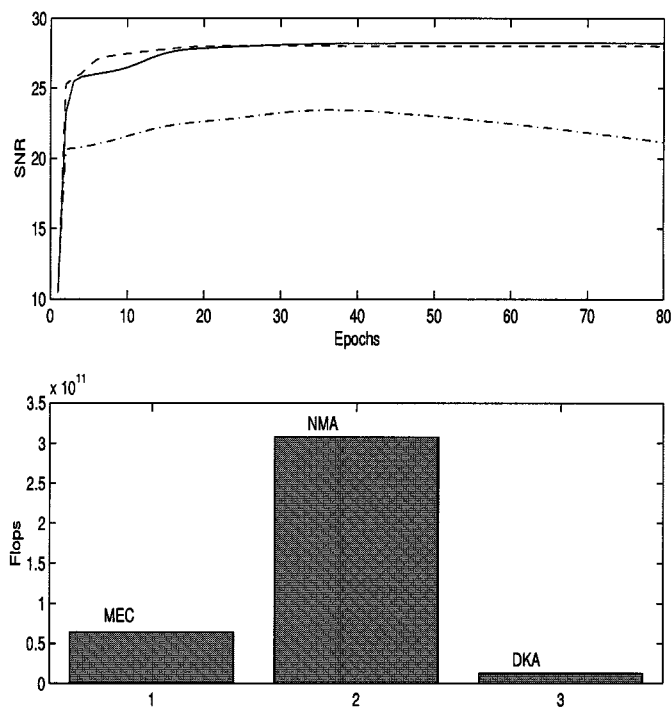


Fig. 7. Performance of Douglas–Kung algorithm, Nishimori’s algorithm and mechanics-like one for principal component extraction from natural image. Top: SNR versus epochs. Bottom: total flops.

jected to standard pre-processing like scaling and windowing-vectorization to give vectors  $\mathbf{x} \in \mathcal{R}^{64}$  as input to the network, which has 8 outputs. The performance index, the signal-to-noise ratio (SNR), defined as in [39] measures the discrepancy between the original and the reconstructed (compressed–decompressed) image, thus the goodness of principal component estimation. It is shown versus the number of learning epochs; each epoch corresponds to a whole image scanning, that is to  $256 \times 256/64 = 1024$  iterations. In these simulations Douglas–Kung algorithm gives no good results, while both MEC and Nishimori’s exhibit similar and satisfactory performances; anyway, it is worth noticing the difference in computational complexity exhibited by the latter two algorithms, which allows concluding that MEC shows best tradeoff between numerical performances and computational efforts.

### C. Application to Blind Signal Separation (BSS)

As another application, the proposed algorithm can be used to solve blind separation problems. 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 be separated out are statistically independent and are mixed by a linear full-rank operator, the independent component analysis (ICA) theory may be employed; it tries to remix the observed mixture in order to make the resulting signals *as independent as possible*. 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 [12] whereby it is known that an ICA stage can be decomposed into two sub-

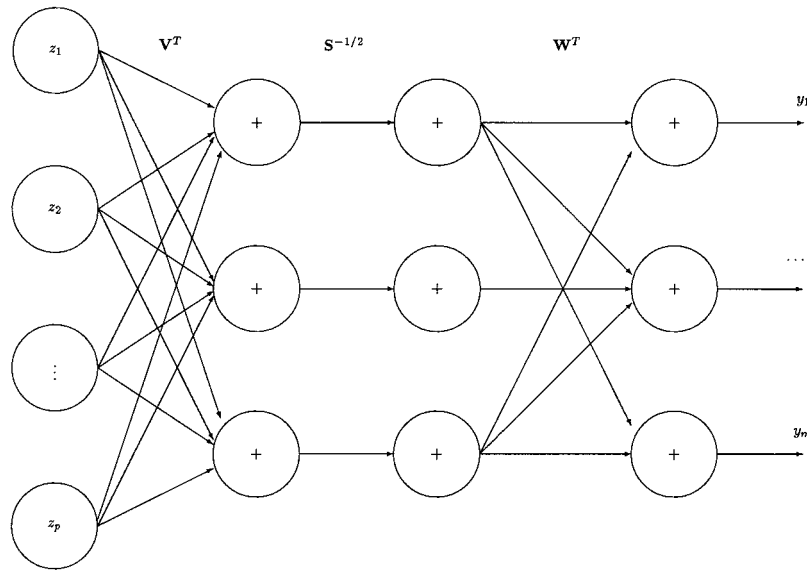


Fig. 8. Three-layer neural architecture for blind source separation.

sequent stages: A whitening stage and a ICA<sup>⊥</sup> one, therefore the signal  $\mathbf{z} = \mathbf{H}^T \mathbf{s}$  at the sensors can be first standardized and then *orthonormally separated* (ICA<sup>⊥</sup>) by a three-layer network as depicted in Fig. 8. Here we suppose source signal stream  $\mathbf{s} \in \mathcal{R}^m$ , observed linear mixture stream  $\mathbf{z} \in \mathcal{R}^p$ , with  $p \geq m$ , thus mixing matrix  $\mathbf{H}^T \in \mathcal{R}^{p \times m}$ .

The first operation can be performed by using a two-stage neural network, described by  $\mathbf{x} = \mathbf{S}^{-1/2} \mathbf{V}^T \mathbf{z}$ , composed by a linear neural compressor with weight-matrix  $\mathbf{V}$  (e.g., a PCA network) followed by a simple neural post-scaler, described by a diagonal weight-matrix  $\mathbf{S}^{-1/2}$  which forces the powers of its outputs to be unitary. Here we use a MEC-net endowed with H-Force forcing term and a low-pass filter described by

$$\Delta \mathbf{S} = \beta [\text{diag}(\mathbf{V}^T \mathbf{z} \mathbf{z}^T \mathbf{V}) - \mathbf{S}] \quad (24)$$

with  $\beta > 0$ , as an estimator of the true optimal scaler  $\mathbf{S}$ . Here  $\mathbf{x} \in \mathcal{R}^m$ , thus  $\mathbf{S}^{-1/2} \mathbf{V}^T \in \mathcal{R}^{m \times m}$ .

After whitening it holds  $E_{\mathbf{x}}[\mathbf{x} \mathbf{x}^T] = \mathbf{I}_p$ , and source signals can be separated from  $\mathbf{x}$  by means of a pure rotation through a linear neural transformation described by  $\mathbf{y} = \mathbf{W}^T \mathbf{x}$ ; this way,  $\mathbf{W}^T \in \mathcal{R}^{m \times m}$ . The orthonormal separation can be performed by a MEC-net with a proper forcing term. As a theoretical example only, let us consider Bell–Sejnowski’s criterion [5], which arises from InfoMax principle (see for instance [2], [5], [19] and references therein). It is defined as

$$J_{\text{BS}}(\mathbf{W}) = \frac{1}{2} E_{\mathbf{x}} \left\{ \ln \left[ |\det(\mathbf{W}^T)| \prod_{i=1}^3 \tanh'(\mathbf{w}_i^T \mathbf{x}) \right] \right\} \quad (25)$$

and should be *minimized* with respect to  $\mathbf{W}$ , therefore we just assume  $U \stackrel{\text{def}}{=} k_{\text{BS}} J_{\text{BS}}$ , where  $k_{\text{BS}}$  is a positive scaling factor. By definition, the ISE of the resulting active force reads

$$\mathbf{F} = -k_{\text{BS}} \mathbf{x} \tanh[\mathbf{x}^T \mathbf{W}] \quad (26)$$

where the  $\tanh[\cdot]$  function acts component-wise. Note that  $U$  assumes a simplified expression with respect to  $J_{\text{BS}}$  since the MECs property  $\det(\mathbf{W}^T \mathbf{W}) = 1 \Rightarrow |\det(\mathbf{W}^T)| = 1$  holds true. It is comparable to the effect introduced by Amari’s natural gradient [2], [3].

Since the overall source-to-output matrix  $\mathbf{P} \stackrel{\text{def}}{=} \mathbf{W}^T \mathbf{S}^{-1/2} \mathbf{V}^T \mathbf{H}^T \in \mathcal{R}^{m \times m}$  should become quasidiagonal (i.e., such that only one entry per row differs from zero), we took as convergence measure the Amari–Cichocki–Yang index [3]

$$n(\mathbf{P}) \stackrel{\text{def}}{=} \sum_{i=1}^m \sum_{j=1}^m \left( \frac{|P_{ij}|}{\max_k |P_{ik}|} + \frac{|P_{ij}|}{\max_k |P_{kj}|} \right) - 2m. \quad (27)$$

As practical examples, in the following five experiments are illustrated through computer simulations.

*Experiment 1* [20]: In the following simulation, the aim is to separate out four independent synthetic signals from four their linear mixtures; all the source signals have negative kurtosis, thus the following simple potential energy function may be used as optimization criterion [8], [13]

$$U(\mathbf{W}) = \frac{k_C}{8} \sum_{i=1}^4 E_{\mathbf{x}}[y_i^4] \quad (28)$$

where  $k_C$  is a positive scaling factor. The resulting ISE of active force has the expression

$$\mathbf{F} = -k_C \mathbf{x} (\mathbf{x}^T \mathbf{W})^3 \quad (29)$$

where the  $(\cdot)^3$ -exponentiation acts component-wise.

Fig. 9 shows the result of the application of the proposed algorithm on the four mixed signals; the learning parameters were  $\mu = 5$ ,  $k_C = 10$ ,  $\theta = 0.001$ ,  $r = 1$ . The first four rows depict the network outputs during the learning phase, while the fifth row shows the separation index.



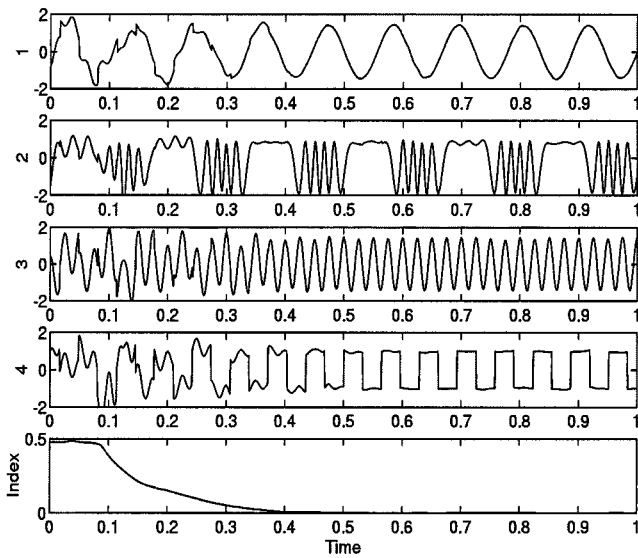


Fig. 9. Four synthetic source signals separation. Outputs of the MEC network and separation index  $n(\mathbf{P})$  versus time.

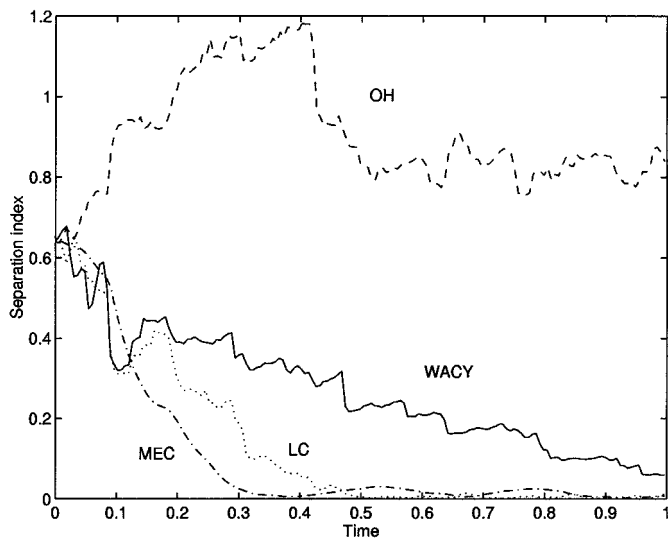


Fig. 10. Separation indexes  $n(\mathbf{P})$  for comparison, four synthetic signals.

*Experiment 2:* 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) [3], the Laheld–Cardoso (referred to as LC) algorithm [8], and the Oja–Hyvärinen algorithm [28] (referred to as OH).

In Fig. 10 separation indexes are shown when the algorithms are used to separate out the four synthetic signals of Experiment 1. In this simulations we used the same values for the parameters as above except that  $k_C = 15$ , and the same learning step size  $\theta$  for all the algorithms.

*Experiment 3:* The following experiment has no direct practical relevance, but was designed to illustrate the deep difference among the “mechanical” *strongly constrained* learning algorithm and the conventional loosely constrained gradient-based ones.

TABLE I  
AN EXAMPLE OF THE RESIDUAL FREEDOM DEGREES (RFD) FOR  $p = m = 5$

$\bar{m}$	RFD	STILLNESS STATUS
1	9	HYPO-STATIC
2	7	HYPO-STATIC
3	4	HYPO-STATIC
4	0	STATIC
5	-5	HYPER-STATIC

As source signals we picked  $100 \times 100$  gray-levels natural images either super-Gaussian and sub-Gaussian. In this case criterion (28) should not be used, since it is valid only when source signals are known to be all super-Gaussian or, upon sign switching, sub-Gaussian. Nevertheless, even in this case the MEC algorithm can be effective when provided with it. To see why, let us consider a “partial learning” case where  $p$  is the number of mixtures,  $m$  is the number of sources, and  $\bar{m}$  is the number of correctly extracted components, i.e., those source signals whose kurtosis’ sign corresponds to the sign chosen within criterion (28); the number RFD of residual degrees of freedom, corresponding to the effective number of network’s parameters still to be learned, is easily recognized to be

$$\text{RFD} = pm - m(m+1)/2 - \bar{m}(\bar{m}+1)/2$$

clearly learning stops when  $\text{RFD} = 0$ . The Table I illustrates an example for  $p = m = 5$ ; the static condition is evidenced against the hypo-static case (i.e., less constraints than degrees of freedom) and the hyper-static case (i.e., more constraints than degrees of freedom). It clearly evidences that when  $\bar{m} = 4$  the system has learned all the expected five independent components.

To experimentally demonstrate the previous considerations, Fig. 11 shows the separation of two sub-Gaussian images and two super-Gaussian images. As the performances may depend on the mixing matrix, the separation indexes are averaged over ten trials.

*Experiment 4:* An interesting experiment concerns blind separation of several source signals. An example of the behavior of MEC, LC, OH and WACY algorithms is illustrated in Fig. 12; it has been obtained by averaging over ten trials the performance indexes on 12 synthetic sub-Gaussian source streams.

As this simulation shows, in this large-scale problem the MEC algorithm is the only one which exhibits a reliable behavior. For this example, we have evaluated numerically the computational complexity related to the four considered algorithms, here expressed as the number of floating point operations (flops) required by each algorithm to run; they are also reported in the Fig. 12. The MEC and OH algorithms require nearly the same computational efforts, which is about three times the complexity exhibited by LC and WACY algorithm: The correct source separation required in these simulations an extra computational cost.

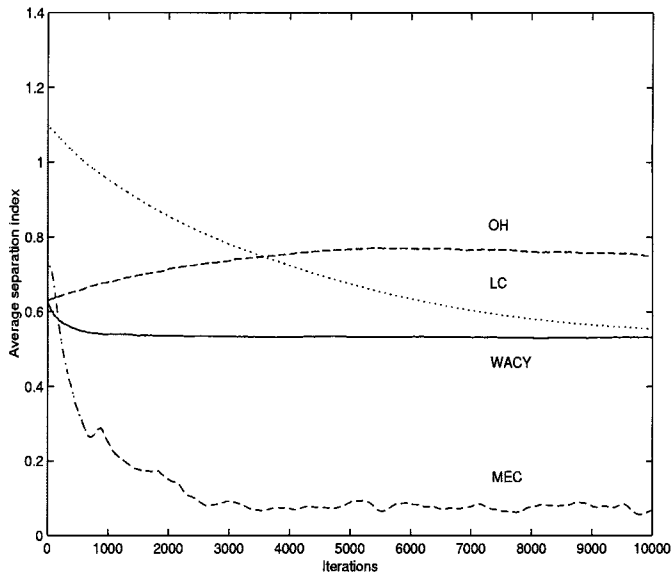


Fig. 11. Separation indexes  $n(\mathbf{P})$  for comparison, four natural images (two super-Gaussian and two sub-Gaussian).

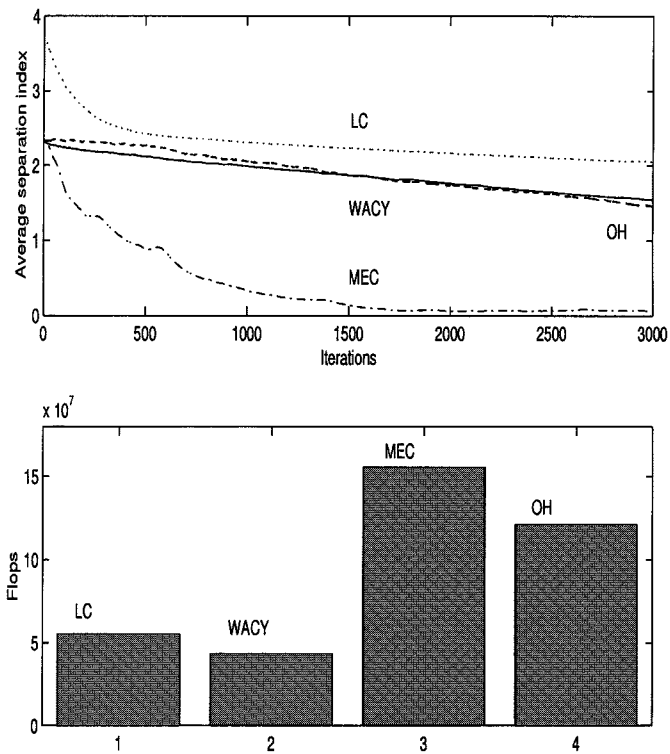


Fig. 12. Top: Separation indexes  $n(\mathbf{P})$  for comparison, 12 synthetic sub-Gaussian source streams. Bottom: Total flops.

*Experiment 5:* The problem of mixed super-Gaussian and Sub-Gaussian source signals, referred to as hybrid (or eterokurtic) blind source separation by the independent component analysis, may be definitely solved in a theoretically satisfactory way (for a recent review, see, e.g., [19], [23]). Fig. 13 refers to the case when four super-Gaussian and sub-Gaussian independent signals are linearly mixed to give four mixtures, which, after prewhitening, are fed to the network. The cost function for network's learning is based upon network's outputs kurtoses,

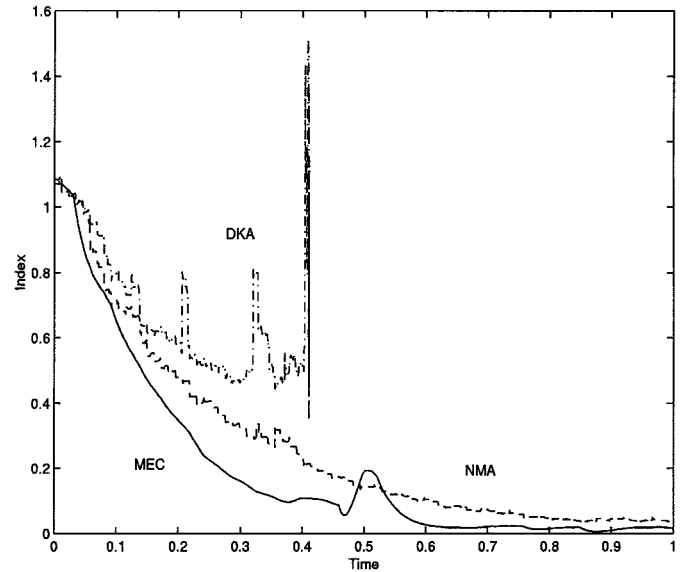


Fig. 13. Performance of Douglas–Kung algorithm, Nishimori's algorithm and MEC one for blind source separation. The index  $n(\mathbf{P})$  versus time is shown.

and an automatic kurtoses sign estimation mechanism is also employed, as in [15], [28], [42].

In fact, as already mentioned in this case PEF function (28) is not valid, as term  $E_x[y_i^4]$  should be maximized or minimized depending on the sources' kurtoses signs. However, as separation is blind, these signs are unknown, thus their should be “guessed” on the basis of source signals estimation provided by the network during learning.

Likely, the reason for which Douglas–Kung algorithm suddenly diverges, is that their rule keeps the weight-matrix orthogonal only in the continuous-time limit, while in the discrete-time implementation it quickly loses the constraint, contrary to Nishimori's algorithm which exactly meets the restriction and to MEC algorithm which allows keeping the constraints respected with the desired accuracy.

The reported results show the MEC may exhibit faster and more accurate convergence when run on both synthetic and real-world signals.

#### IV. CONCLUSION

In this paper a new class of learning rules, based on a mechanical paradigm, was presented. 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 to closely-related geodesic algorithms.

Forthcoming work on this topic will involve the extension of the presented theory both to complex-valued signal processing (e.g., for achieving blind separation of complex-valued “circularly distributed” telecommunications signals—for a recent review see, e.g., [26], [21]) and, most interestingly, the extension to para-unitary filter banks adaptive design for blind separation from convolutional mixtures [31], [32], [40].

## REFERENCES

- [1] C. Aluffi-Pentini, V. Parisi, and F. Zirilli, "Global optimization and stochastic differential equations," *J. Optimization Theory Applicat.*, vol. 47, pp. 1–16, 1985.
- [2] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, pp. 251–276, 1998.
- [3] S.-I. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1993, vol. 8.
- [4] J. Angeles, "Rational kinematics," *Springer Tracts in Natural Philosophy*, vol. 34, 1989.
- [5] A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.
- [6] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
- [7] R. W. Brockett, "Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems," *Linear Algebra Applicat.*, vol. 146, pp. 79–91, 1991.
- [8] J. F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 3017–3030, Dec. 1996.
- [9] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*. New York: Wiley, 1993.
- [10] K. Chellapilla and D. B. Fogel, "Evolution, neural networks, games, and intelligence," *Proc. IEEE*, vol. 87, no. 9, pp. 1471–1496, Sept. 1999.
- [11] A. Cichocki, J. Karhunen, W. Kasprzak, and R. Vigario, "Neural networks for blind separation with unknown number of sources," *Neurocomput.*, vol. 24, pp. 55–93, 1999.
- [12] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.
- [13] P. Comon and E. Moreau, "Improved contrast dedicated to blind separation in communications," in *Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, 1997, pp. 3453–3456.
- [14] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel, "The Helmholtz machine," *Neural Comput.*, vol. 7, pp. 889–904, 1995.
- [15] N. Delfosse and Ph. Loubaton, "Adaptive blind separation of independent sources: A deflation approach," *Signal Processing*, vol. 45, pp. 59–83, 1995.
- [16] K. I. Diamantaras and S.-Y. Kung, *Principal Component Neural Networks: Theory and Applications*. New York: Wiley, 1996.
- [17] S. C. Douglas and S.-Y. Kung, "An ordered-rotation Kuicnet algorithm for separating arbitrarily-distributed sources," in *Proc. Int. Conf. Independent Component Anal. (ICA'99)*, 1999, pp. 81–86.
- [18] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Applicat.*, vol. 20, no. 2, pp. 303–353, 1998.
- [19] S. Fiori, "Entropy optimization by the PFANN network: Application to independent component analysis," *Network: Comput. Neural Syst.*, vol. 10, no. 2, pp. 171–186, May 1999.
- [20] —, "'Mechanical' neural learning applied to blind source separation," *Electron. Lett.*, vol. 35, no. 22, Oct. 1999.
- [21] —, "Blind separation of circularly-distributed sources by neural extended APEX algorithm," *Neurocomput.*, vol. 34, no. 1–4, pp. 239–252, Aug. 2000.
- [22] —, "Neural networks for blind signal processing," Ph.D. dissertation, Dept. Elect. Eng., Univ. Bologna, Bologna, Italy, Mar. 2000.
- [23] —, "Blind signal processing by the adaptive activation function neurons," *Neural Networks*, vol. 13, no. 6, pp. 597–611, Aug. 2000.
- [24] S. Fiori and F. Piazza, "A general class of  $\psi$ -APEX PCA neural algorithms," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 1394–1398, Sept. 2000.
- [25] S. Fiori, "A theory for learning by weight flow on Stiefel–Grassman manifold," *Neural Comput.*, vol. 13, no. 7, pp. 1625–1647, July 2001.
- [26] —, "On blind separation of complex-valued sources by extended Hebbian learning," *IEEE Signal Processing Lett.*, vol. 8, pp. 217–220, Aug. 2001.
- [27] 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. Circuits Syst. II*, vol. 41, pp. 718–729, Nov. 1994.
- [28] A. Hyvärinen and E. Oja, "Independent component analysis by general nonlinear Hebbian-like rules," *Signal Processing*, vol. 64, no. 3, pp. 301–313, 1998.
- [29] J. Karhunen, "Neural approaches to independent component analysis and source separation," in *Proc. 4th Europ. Symp. Artificial Neural Networks (ESANN'96)*, 1996, pp. 249–266.
- [30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
- [31] J. L. Lacomme, "A survey of source separation," in *Proc. Independent Component Anal. (ICA'98)*, 1998, pp. 1–6.
- [32] R. H. Lambert, "Multi-channel blind deconvolution: FIR matrix algebra and separation of multipath mixtures," Ph.D. dissertation, Dept. Elec. Eng., Univ. Southern California, Los Angeles, 1996.
- [33] T. M. Martinez, S. G. Berkovich, and K. J. Schulten, "'Neural-gas' network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks*, vol. 4, pp. 558–569, July 1993.
- [34] J. L. McClelland and D. E. Rumelhart, *Parallel Distributed Processing*. Cambridge, MA: MIT Press, 1986.
- [35] D. M. Monro, B. E. Bassil, and G. J. Dickson, "Orthonormal wavelets with balanced uncertainty," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, 1996, pp. 581–584.
- [36] Y. Nishimori, "Learning algorithm for ICA by geodesic flows on orthogonal group," in *Proc. Int. Joint Conf. Neural Networks*, Washington, DC, July 1999.
- [37] E. Oja, "Neural networks, principal components and subspaces," *Int. J. Neural Syst.*, vol. 1, pp. 61–68, 1989.
- [38] J. Principe, J. Fisher, III, and D. Xu, "Information theoretic learning," in *Unsupervised Adaptive Filtering*, S. Haykin, Ed. New York: Wiley, 2000.
- [39] T. D. Sanger, "Optimal unsupervised learning in a single-layered feed-forward network," *Neural Networks*, vol. 2, pp. 459–473, 1989.
- [40] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomput.*, vol. 22, pp. 21–34, 1998.
- [41] P. Stoica and K. C. Sharma, "Maximum likelihood methods for DOA estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1132–1143, July 1990.
- [42] R. Thawonmas, A. Cichocki, and S.-I. Amari, "A cascade neural network for blind signal extraction without spurious equilibria," *IEICE Trans. Fundamentals*, vol. E81-A, no. 9, pp. 1833–1846, Sept. 1998.
- [43] K. Torkkola and W. M. Campbell, "Mutual information in learning feature transformation," in *Proc. 17th Int. Conf. Machine Learning*: Stanford University, June 29–July 2, 2000.
- [44] R. Urbanczik, "Statistical mechanics of mutual information maximization," *Europhys. Lett.*, vol. 49, pp. 685–690, 2000.
- [45] J. M. Zurada, *Introduction to Neural Artificial Systems*. St. Paul, MN: West, 1992.



**Simone Fiori** was born in Rimini, Italy, in June 1971. He received the Laurea (Dr. Ing.) degree *cum laude* in electronics engineering from the University of Ancona, Ancona, Italy, in July 1996 and the Ph.D. degree in electrical engineering (circuit theory) from the University of Bologna, Bologna, Italy, in March 2000.

In November 2000, he was appointed Assistant Professor of circuit theory at the University of Perugia, Italy. His research interests include unsupervised learning theory for artificial neural networks, linear and nonlinear adaptive discrete-time filter theory, vision and image processing by neural networks, continuous-time and discrete-time circuits for stochastic information processing. He is author of more than 70 refereed journal and conference papers on these topics and is serving as Reviewer for the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, *Neurocomputing*, and the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION.

Dr. Fiori was the recipient of the 2001 "E.R. Caianiello Award" for the best Ph.D. dissertation in the artificial neural network field.