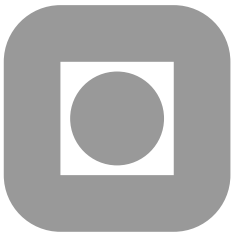
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Neural Learning by Geometric Integration of Reduced 'Rigid-Body' Equations

by
Elena Celledoni and Simone Fiori

PREPRINT
NUMERICS NO. 4/2002



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/numerics/2002/N4-2002.ps
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Neural Learning by Geometric Integration of Reduced 'Rigid-Body' Equations*

Elena Celledoni† Simone Fiori‡

November 18, 2002

Abstract

In previous contributions we presented a new class of algorithms for orthonormal learning of linear neural networks 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 good numerical stability, strongly binding to the orthonormal manifolds, and good controllability of the learning dynamics, the proposed algorithms were not completely satisfactory from a computational-complexity point of view. The main drawbacks were the non-efficient representation of the learning matrix-quantities and the non-efficient integration of the resulting learning differential equations. In this Technical Report a new and efficient representation of the learning equations is proposed, and a possible way to integrate them is suggested. Numerical experiments concerning Principal Subspace Analysis and Independent Component Analysis were carried out with both synthetic and real-world data in order to confirm the effectiveness of the proposed theory.

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 Stiefel manifold).

The analysis of these contributions has raised the idea that geometric concepts (such as the theory of Lie groups) give the fundamental instruments for

*This work is part of the activities of the special year in Geometric Integration at the Center for Advanced Study in Oslo

†Email: Elena.Celledoni@math.ntnu.no, WWW: http://www.math.ntnu.no/~elenac/

‡Email: sfr@unipg.it, WWW: http://www.math.ntnu.no/~elenac/

gaining a deep insight into the mathematical properties of several learning and optimization paradigms.

The interest displayed by the scientific community about this research topic is also testified by several activities such as the organization of the special issue on "Non-Gradient Learning Techniques" of the International Journal of Neural Systems (guest editors A. de Carvalho and S.C. Kremer), the Post-NIPS*2000 workshop on "Geometric and Quantum Methods in Learning", organized by S.-i. Amari, A. Assadi and T. Poggio (Colorado, December 2000), the workshop "Uncertainty in Geometric Computations" held in Sheffield, England, in July 2001, organized by J. Winkler and M. Niranjan, the special session of the IJCNN'02 conference on "Differential & Computational Geometry in Neural Networks" held in Honolulu, Hawaii (USA), in May 2002 and organized by E. Bayro-Corrochano, and the workshop "Information Geometry and its Applications", held in Pescara (Italy), in July 2002, organized by P. Giblisco.

Understanding the underlying geometric structure of a network parameters space is extremely important to designing systems that can effectively navigate the space while learning.

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 networks and learning. Research topics include differential geometrical methods for learning, the Lie group learning algorithms [22], the natural (Riemannian) gradient techniques [2, 24, 30], the numerical aspects of the solution of the matrix-equations on Lie groups and homogeneous spaces [8, 9, 14, 29].

Some specific exemplary applied topics that can be addressed under the mentioned general methodology are: Principal component/subspace analysis [20, 41]; Neural independent component analysis and blind source separation [20, 42]; Information geometry [2]; Geometric Clifford algebra for the generalization of neural networks [3]; Geometrical methods of unsupervised learning for blind signal processing [20, 22]; Eigenvalue and generalized eigenvalue problems, optimal linear compression, noise reduction and signal representation [13, 16, 34, 40, 41]; Simulation of the physics of bulk materials [15]; Minimal linear system realization from noise-injection measured data and invariant subspace computation [15, 32]; Optimal de-noising by sparse coding shrinkage [35]; Direction of arrival estimation [1]; Linear programming and sequential quadratic programming [6, 15]; Optical character recognition by transformation-invariant neural networks [38]; Analysis of geometric constraints on neural activity for natural three-dimensional movement [43]; Electrical networks fault detection

[31]; Synthesis of digital filters by improved total least-squares technique [25]; Speaker verification [39]; Adaptive image coding [33]; Dynamic texture recognition [37].

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 [21, 22]. 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-based experiments 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 performance and computational complexity even when compared to closely-related algorithms.

An open question about the mentioned algorithm concerned the computational complexity which arises from the necessity of matrix computation, such as the repeated evaluation of the exponential map. The aim of the present Technical Report is to investigate a different formulation of the MEC learning equations and to suggest a possible numerical strategy for achieving their solution based on geometric integration.

2 Summary of the MEC 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 drawbacks related to its computational complexity. We then describe the proposed improvement based on an advantageous parameterization of the angular-velocity space.

2.1 Summary of rigid-body learning theory

Let $\mathcal{S}_m = \{(\mu_i, \mathbf{w}_i), (\mu_i, -\mathbf{w}_i)\}_{i\in\{1,\dots,m\}}$ be a *rigid* system of masses, where the m vectors $\mathbf{w}_i \in \mathbb{R}^p$ represent the instantaneous positions of $2m$ masses $\mu_i \in \mathbb{R}_0^+$ 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 [21] we assumed the values of the masses μ_i constant at 1. In Figure 1 an exemplary configuration of $\mathcal{S}_m$ for $p = 3$ and $m = 3$ is illustrated.

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 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 $\mathcal{S}_m$. Furthermore, masses move in a 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 *([21].) Let $\mathcal{S}_m \subset \mathbb{R}_0^+ \times \mathbb{R}^p$ be the physical system described above: Let us denote with $\mathbf{F}$ the $p \times m$ matrix of the* active forces*, with $\mathbf{P}$ 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:*

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

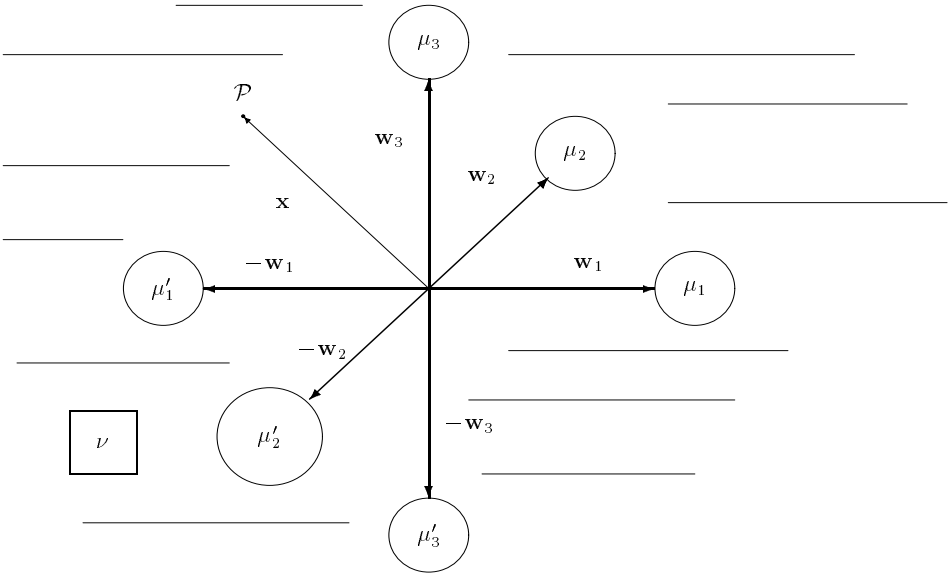


Figure 1: A configuration of $\mathcal{S}_m$ for $p = 3$ and $m = 3$. The μ_i represent the masses, vectors $\mathbf{w}_i$ represent their coordinates and $\mathbf{x}$ is the coordinate-vector of the external point $\mathcal{P}$.

$$\frac{d\mathbf{B}}{dt} = (\mathbf{F} + \mathbf{P})\mathbf{W}^T - \mathbf{W}(\mathbf{F} + \mathbf{P})^T \ , \tag{2}$$

$$\mathbf{P} = -\nu\mathbf{B}\mathbf{W} \ , \tag{3}$$

with ν 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.

Provided that initial conditions $\mathbf{B}(0) = \mathbf{B}_0$ and $\mathbf{W}(0) = \mathbf{W}_0$ are given and the expression of $\mathbf{F}$ as a function of $\mathbf{W}$, the equations (1)-(3) represent an initial-value problem in the matrix-variables $(\mathbf{B}, \mathbf{W})$ whose asymptotic solution $\mathbf{W}_\star$ represents the neural network connection pattern after learning.

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

- Let us denote by $\mathbf{so}(p, \mathbb{R})$ the set of skew-symmetric matrices. It is immediate to verify that if $\mathbf{B}(0) \in \mathbf{so}(p, \mathbb{R})$ then equation (2) provides $\dot{\mathbf{B}}(t) \in \mathbf{so}(p, \mathbb{R})$ and thus $\mathbf{B}(t) \in \mathbf{so}(p, \mathbb{R})$ 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 [5]). Because of the skew-symmetry of $\mathbf{B}(t)$ 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: $\mathbf{BW} = \mathbf{0}_{p\times m}$, $\mathbf{FW}^T - \mathbf{WF}^T = \mathbf{0}_{p\times m}$, $\mathbf{W} \in \mathrm{St}(p, m, \mathbb{R})$, $\mathbf{B} \in \mathbf{so}(p, \mathbb{R})$, where $\mathbf{0}_{p\times q}$ denotes the null element of $\mathbb{R}^{p\times q}$. It is important to recall that both $\mathbf{W}(t)$ and $\mathbf{B}(t)$ 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 potential energy function (PEF) U, which yields force:

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

Generally we may suppose U dependent upon $\mathbf{W}$, $\mathbf{w}_0$, and on the statistics of $\mathbf{x}$; 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. 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 searching 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;

- 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 local search-path.

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 *([22].) 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* $\mathbf{B}(0)$ *is skew-symmetric. Let us also define the matrix function* $\mathbf{F} \stackrel{\text{def}}{=} -\frac{\partial U}{\partial \mathbf{W}}$*, and denote as* $\mathbf{F}_\star$ *the value of* $\mathbf{F}$ *at* $\mathbf{W}_\star$*. A state* $\mathbf{X}_\star = (\mathbf{B}_\star, \mathbf{W}_\star)$ *is stationary for the system if* $\mathbf{F}_\star^T \mathbf{W}_\star$ *is symmetric and* $\mathbf{B}_\star \mathbf{W}_\star = \mathbf{0}$*. 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.

Theorem 3 *([22].) 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* $\mu \geq 0$*.*

2.2 Present study motivation

The discussed equations describing the MEC learning rule are based on two matrix state-variables, namely $\mathbf{B}$ and $\mathbf{W}$, whose dimensions are $p \times p$ and $p \times m$, respectively, where p denotes the number of neural network's inputs and m denotes the number of network's outputs. As a consequence, even if the dimension pm of the network is of reduced size, namely $m \ll p$, the state-matrix $\mathbf{B}$ assumes the largest possible dimension. An extreme example is represented by the one-unit network case, in which in order to train a single neuron ($m = 1$) with many inputs ($p \gg 1$) a full $p \times p$ angular-velocity matrix is required. Also, it is useful noting that $\mathbf{B}$ is a $p \times p$ matrix with only $p(p-1)/2$ distinct entries, because of skew-symmetry.

In order to overcome this representation problem, we propose to recast the

learning equations into the following system of differential equations:

$$\left\{ \begin{array}{rcl} \dot{\mathbf{W}} & = & \mathbf{V} \, , \\ \dot{\mathbf{V}} & = & g(\mathbf{V}, \mathbf{W}) \, , \end{array} \right. \tag{5}$$

where $\mathbf{V} \in \mathbb{R}^{p \times m}$ replaces $\mathbf{B}$ and $g : \mathbb{R}^{p \times n} \times \mathbb{R}^{p \times n} \to \mathbb{R}^{p \times m}$ is describing the dynamics of the considered rigid-body mechanical system.

It is worth noting that the new formulation of the equations is completely advantageous only if they are then integrated numerically in a proper and efficient way. In particular:

- **Preservation of the underlying structure**: The rigid-body dynamics differential equations should be integrated in a way that preserves the rigidity of the system both in order to ensure the quality of the signal processing solution provided by the neural system and to preserve some quantitative features of the learning theory such as intrinsic stability. The last point is well-represented by the very-long integration time question arising in on-line signal processing tasks: The results of extensive numerical simulations presented recently in [23] clearly show that the major part of existing algorithms are unable to tackle long signal processing tasks because the network-state eventually loose the adherence to the invariant manifold their learning theories are equipped with.

- **Efficiency**: It has been observed (see e.g.[22]) that certain classes of learning algorithms involve rectangular matrices whose row/column ratio (p/m) is quite low. This suggests that an integration method that takes into account the structure of matrix-type expressions involved in these learning equations might possess contained computational complexity. In the following we suggest an integration scheme of complexity $\mathcal{O}(pm^2)$.

2.3 New equations for the MEC algorithm

With the proposed representation, the learning dynamics is described by the new pair of state-variables $(\mathbf{V}, \mathbf{W})$ representing a generic point on the tangent bundle of the Stiefel manifold $\mathrm{TSt}(p, m, \mathbb{R})$. In order to derive the new equations for these state-variables we consider the following characterization of second order differential equations on $\mathrm{St}(p, m, \mathbb{R})$.

Theorem 4 *Any second order differential equation on $St(p, m, \mathbb{R})$ can be ex-*

pressed in the form

$$\begin{aligned} \dot{\mathbf{W}} &= \mathbf{V} = (\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}) \cdot \mathbf{W} \\ \dot{\mathbf{V}} &= (\mathbf{L}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{L}^{\mathrm{T}}) \cdot \mathbf{W} + (\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}) \cdot \mathbf{V} \end{aligned} \tag{6}$$

with $\mathbf{G} = \mathbf{V} + \mathbf{W}(-\mathbf{W}^{\mathrm{T}}\mathbf{V}/2 + \mathbf{S})$, $\mathbf{S}$ *arbitrary* $m \times m$ *symmetric matrix, and* $\mathbf{L} = \dot{\mathbf{G}} - \mathbf{G}\mathbf{W}^{\mathrm{T}}\mathbf{G}$.

Proof

It has been proven in [8] that any vector $\mathbf{V}$ tangent to the Stiefel manifold at a point $\mathbf{W}$ can be written in the form

$$\mathbf{V} = (\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}) \cdot \mathbf{W} \tag{7}$$

where $\mathbf{G} = \mathbf{V} + \mathbf{W}(-\mathbf{W}^{\mathrm{T}}\mathbf{V}/2 + \mathbf{S})$ and $\mathbf{S}$ is an arbitrary symmetric $m \times m$ matrix. $\mathbf{S}$ can be chosen arbitrarily because of the skew-symmetry of $\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}$. In fact by substituting the expression for $\mathbf{G}$ in $\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}$ we obtain

$$\mathbf{V}\mathbf{W}^{T} - \frac{\mathbf{W}\mathbf{W}^{T}\mathbf{V}\mathbf{W}^{T}}{2} + \mathbf{W}\mathbf{S}\mathbf{W}^{T} - \mathbf{W}\mathbf{V}^{T} - \frac{\mathbf{W}\mathbf{V}^{T}\mathbf{W}\mathbf{W}^{T}}{2} - \mathbf{W}\mathbf{S}\mathbf{W}^{T} \ ,$$

which is independent of $\mathbf{S}$.

By differentiating (7) with respect to time we obtain,

$$\ddot{\mathbf{W}} = \dot{\mathbf{V}} = (\dot{\mathbf{G}}\mathbf{W}^{\mathrm{T}} + \mathbf{G}\mathbf{V}^{\mathrm{T}} - \mathbf{V}\mathbf{G}^{\mathrm{T}} - \mathbf{W}\dot{\mathbf{G}}^{\mathrm{T}}) \cdot \mathbf{W} + (\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}) \cdot \mathbf{V}. \tag{8}$$

Now by multiplying out $\mathbf{V} = (\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}) \cdot \mathbf{W}$ and using the property $\mathbf{W}^{T}\mathbf{W} = \mathbf{I}_m$, we obtain $\mathbf{V} = \mathbf{G} - \mathbf{W}\mathbf{G}^{\mathrm{T}}\mathbf{W}$, which we substitute in the first term of the right hand side of (8), and we obtain

$$\dot{\mathbf{V}} = ((\dot{\mathbf{G}} - \mathbf{G}\mathbf{W}^{\mathrm{T}}\mathbf{G})\mathbf{W}^{\mathrm{T}} - \mathbf{W}(\dot{\mathbf{G}} - \mathbf{G}\mathbf{W}^{\mathrm{T}}\mathbf{G})^{\mathrm{T}}) \cdot \mathbf{W} + (\mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}) \cdot \mathbf{V},$$

which concludes the proof. □

By comparing the equation (1,2,3) with (6), and recalling that $\dot{\mathbf{W}} = \mathbf{V} = \mathbf{B}\mathbf{W}$ which implies $\dot{\mathbf{V}} = \dot{\mathbf{B}}\mathbf{W} + \mathbf{B}\mathbf{V}$, we recognize that $\mathbf{B} = \mathbf{G}\mathbf{W}^{\mathrm{T}} - \mathbf{W}\mathbf{G}^{\mathrm{T}}$ and $\mathbf{L} = \mathbf{F} + \mathbf{P}$.

The matrix $\mathbf{S}$ plays a role in the computation of $\mathbf{G}$, but not in the evaluation of the rigid-body learning equations. In this paper for simplicity we choose $\mathbf{S} = \mathbf{0}$. It is worth noting that other choices for the matrix $\mathbf{S}$ could be useful e.g. for reasons of numerical stability, [8].

The final expressions for the new MEC learning equations read then:

$$\begin{cases} \dot{\mathbf{W}} &= \mathbf{V} \ , \ \mathbf{P} = -\nu\mathbf{V}, \ \mathbf{G} = \mathbf{G}(\mathbf{V}, \mathbf{W}) \stackrel{\mathrm{def}}{=} \mathbf{V} - \frac{1}{2}\mathbf{W}(\mathbf{W}^{T}\mathbf{V}) \ , \\ \dot{\mathbf{V}} &= \mathbf{F} + \mathbf{P} - \mathbf{W}(\mathbf{F} + \mathbf{P})^{T}\mathbf{W} + (\mathbf{G}\mathbf{W}^{T} - \mathbf{W}\mathbf{G}^{T})\mathbf{V} \ . \end{cases} \tag{9}$$

In order to limit the computational complexity of the above expressions, it is important to compute the matrix products in the right order. For instance, the function $g(\mathbf{V}, \mathbf{W})$ should be computed as follows:

$$g(\mathbf{V}, \mathbf{W}) = \mathbf{F} + \mathbf{P} - \mathbf{W}((\mathbf{F} + \mathbf{P})^T \mathbf{W}) + \mathbf{G}(\mathbf{W}^T \mathbf{V}) - \mathbf{W}(\mathbf{G}^T \mathbf{V}) \ ;$$

in this way, the matrix products involve $p \times m$ and $m \times m$ matrices only, making the complexity burden pertaining to function $g(\cdot, \cdot)$ evaluation of order $\mathcal{O}(pm^2)$.

3 Integration of the Equations

In order to implement the new MEC algorithm on a computer platform, it is necessary to discretize in time the learning equations (9). In order to respect the orthogonality constraints we here use a geometric numerical integrator based on the classical forward Euler method.

3.1 Geometric integration of the learning equations

Geometric Integration (GI) is a recent branch of numerical analysis and computational mathematics. The traditional efforts of numerical analysis and computational mathematics have been to render physical phenomena into algorithms that produce sufficiently precise, affordable and robust numerical approximations. Geometric integration is concerned also with producing numerical approximations preserving the qualitative attributes of the solution to the possible extent. Examples of GI algorithms for differential equations include Lie group integrators, volume and energy preserving integrators, integrators preserving first integrals and Lyapunov functions, Lagrangean and variational integrators, integrators respecting Lie symmetries and integrators preserving contact structures [27].

In the present case, the differential equation for $\mathbf{V}$ may be solved by the standard Euler method, because it belongs to the tangent space to the Stiefel manifold at $\mathbf{W}$, which is a linear space. We integrate the differential equation for $\mathbf{W}$ using the Lie-Euler method which advances the numerical solution by using the left transitive action of $\mathrm{SO}(n)$ on the Stiefel manifold. The action is lifted to the Lie algebra $\mathbf{so}(n)$ using the exponential map. In formulas, we thus get:

$$\left\{ \begin{array}{lcl} \mathbf{V}_{n+1} & = & \mathbf{V}_n + hg(\mathbf{V}_n, \mathbf{W}_n) \ , \\ \mathbf{G}_n & = & \mathbf{V}_n - \frac{1}{2}\mathbf{W}_n(\mathbf{W}_n^T \mathbf{V}_n) \ , \\ \mathbf{W}_{n+1} & = & \exp(h(\mathbf{G}_n \mathbf{W}_n^T - \mathbf{W}_n \mathbf{G}_n^T))\mathbf{W}_n \ , \end{array} \right. \tag{10}$$

where h denotes the time step of the numerical integration, n denotes the discrete-time index, and proper initial conditions are fixed. In the present report we always consider $\mathbf{W}_0 = \mathbf{I}_{p\times m}$ and $\mathbf{V}_0 = \mathbf{0}_{p\times m}$.

3.2 Efficient computation of the matrix exponential

In this section we will discuss the importance of the new formulation of the MEC system for deriving efficient implementations of the method (10).

The computation of the matrix exponential in the equation for $\mathbf{W}_{n+1}$ is a task that should be treated with care in the implementations of (10). Computing $\exp(\mathbf{A}) \stackrel{\text{def}}{=} \sum_{j=0}^{\infty} \mathbf{A}^j/j!$, for a $p \times p$ matrix $\mathbf{A}$, requires typically $\mathcal{O}(p^3)$ flops complexity. The numerical methods for computing the matrix exponential are in fact either based on factorizations of the matrix $\mathbf{A}$, e.g. reduction to triangular or diagonal form [36], or on the use of the powers of $\mathbf{A}$. Among these are for example techniques based on the Taylor expansion, or on the Cayley-Hamilton theorem, e.g. the Putzer algorithm [28] page 506.

Matrix factorizations and powers of matrices require by themselves $\mathcal{O}(p^3)$ flops implying immediately a similar complexity for the mentioned algorithms (see e.g. [26] for an overview).

The complexity is $\mathcal{O}(p^2m)$ if instead we want to compute $\exp(\mathbf{A})\mathbf{X}$ where $\mathbf{X}$ is a $p\times m$ matrix. In this case one could use methods based on the computation of $\mathbf{AX}$ and the successive powers $\mathbf{A}^j\mathbf{X}$ each one involving $\mathcal{O}(p^2m)$ flops.

However, since the first two equations of (10) require $\mathcal{O}(pm^2)$ flops, one would hope to get the same type of complexity for computing $\mathbf{W}_{n+1}$, instead of $\mathcal{O}(p^3)$ or $\mathcal{O}(p^2m)$, especially when p is large and much bigger than m.

At the same time it is very important in our context to obtain approximations $\tilde{\mathbf{X}}$ of $\exp(\mathbf{A})\mathbf{X}$ with the crucial property that $\tilde{\mathbf{X}}$ is an element of the Stiefel manifold. In fact if this requirement is not fulfilled the geometric properties of the method (10) would be compromised.

For this reason the use of approximations of $\exp(\mathbf{A})\mathbf{X}$ based on truncated Taylor expansions is not advisable, because in this case the approximation is not guaranteed to be on the Stiefel manifold.

Since in the method (10) the matrix we want to exponentiate has the special form $\mathbf{A} = \mathbf{G}_n\mathbf{W}_n^T - \mathbf{W}_n\mathbf{G}_n^T = [\mathbf{G}_n, -\mathbf{W}_n][\mathbf{W}_n, \mathbf{G}_n]^T$, the computational costs for $\exp(\mathbf{A})\mathbf{X}$ can be further reduced to $\mathcal{O}(pm^2)$ flops.

In fact, in order to compute $\exp(\mathbf{A})\mathbf{X}$ exactly up to rounding errors one could use the strategy proposed in [10] and proceed as follows: Consider the $2m \times 2m$

matrix defined by $\mathbf{D} \stackrel{\text{def}}{=} [\mathbf{W}_n, \mathbf{G}_n]^T [\mathbf{G}_n, -\mathbf{W}_n]$ and the analytic function $\phi(z) \stackrel{\text{def}}{=} \frac{e^z - 1}{z}$, then it can be proven that:

$$\exp(\mathbf{A})\mathbf{X} = \mathbf{X} + [\mathbf{G}_n, -\mathbf{W}_n]\phi(\mathbf{D})[\mathbf{W}_n, \mathbf{G}_n]^T\mathbf{X} \ . \tag{11}$$

Under the assumption that m is not too large, $\phi(\mathbf{D})$ is easy to compute exactly (up to rounding errors) in $\mathcal{O}(m^3)$ flops. For this purpose we can suggest techniques based on diagonalizing $\mathbf{D}$ or on the use of the Putzer algorithm. The cost of computing $\exp(\mathbf{A})\mathbf{X}$ with this formula is $4pm^2 + pm + \mathcal{O}(m^3)$ flops.

This results in a very convenient algorithm of $\mathcal{O}(pm^2)$ complexity in the case $p \gg m$.

Finally we propose a variant of formula (11) particularly attractive in the case of Stiefel manifolds. We consider a qr-factorization of the $p \times (2m)$ matrix $[\mathbf{W}_n, \mathbf{G}_n]$ since $\mathbf{W}_n$ has orthonormal columns we have

$$[\mathbf{W}_n, \mathbf{G}_n] = [\mathbf{W}_n, \mathbf{W}_n^\perp] \begin{bmatrix} \mathbf{I} & \mathbf{C} \\ \mathbf{O} & \mathbf{R} \end{bmatrix},$$

and $[\mathbf{W}_n, \mathbf{W}_n^\perp]$ has $2m$ orthonormal columns. Analogously

$$[\mathbf{G}_n, -\mathbf{W}_n] = [\mathbf{W}_n, \mathbf{G}_n] \begin{bmatrix} \mathbf{O} & -\mathbf{I} \\ \mathbf{I} & \mathbf{O} \end{bmatrix} = [\mathbf{W}_n, \mathbf{W}_n^\perp] \begin{bmatrix} \mathbf{C} & -\mathbf{I} \\ \mathbf{R} & \mathbf{O} \end{bmatrix}.$$

By putting together the two decomposed factors we obtain the following convenient factorization for $\mathbf{A}$,

$$\mathbf{A} = [\mathbf{W}_n, \mathbf{W}_n^\perp] \begin{bmatrix} \mathbf{C} - \mathbf{C}^{\mathrm{T}} & -\mathbf{R}^{\mathrm{T}} \\ \mathbf{R} & \mathbf{O} \end{bmatrix} [\mathbf{W}_n, \mathbf{W}_n^\perp]^{\mathrm{T}}.$$

Now note that

$$\exp(\mathbf{A}) = [\mathbf{W}_n, \mathbf{W}_n^\perp] \exp\left(\begin{bmatrix} \mathbf{C} - \mathbf{C}^{\mathrm{T}} & -\mathbf{R}^{\mathrm{T}} \\ \mathbf{R} & \mathbf{O} \end{bmatrix} \right) [\mathbf{W}_n, \mathbf{W}_n^\perp]^{\mathrm{T}}$$

and we have reduced the computation of the exponential of the $p \times p$ matrix $\mathbf{A}$ to the computation of the exponential of a $2m \times 2m$ skew-symmetric matrix. The advantage of this new formula lies in the fact that the exponential of a skew-symmetric matrix of moderate size, might be preferable to the computation of the analytic function $\phi(\mathbf{D})$, for a non-normal matrix $\mathbf{D}$. This approach implies however the extra cost of computing a qr-factorization ($\mathcal{O}(pm^2)$ flops).

4 Experimental Set-up on Principal-Subspace and Independent-Component Analysis

The discussed algorithm has been applied to two different problems, namely Principal Subspace Analysis and Independent Component Analysis.

4.1 Principal subspace analysis

Data reduction techniques aim at providing an efficient representation of the data; we consider the research stream which focuses on the compression procedure consisting in mapping the higher dimensional input space into a lower dimensional representation space by means of a linear transformation, as in the Karhunen-Loéve Transform (KLT). The classical approach for evaluating the KLT requires the computation of the input data covariance matrix and then the application of a numerical procedure to extract the eigenvalues and the corresponding eigenvectors; compression is obtained by the use of the only eigenvectors associated with the most significant eigenvalues as a new basis. When large data sets are handled, this approach is not practicable because the dimensions of the covariance matrix become too large to be manipulated. In addition, the whole set of eigenvectors has to be evaluated even though only some of them are used.

In order to overcome these problems, neural-network-based approaches were proposed. Neural principal component analysis (PCA) is a second-order adaptive statistical data processing technique introduced by Oja [34] that helps to remove the second-order correlation among given random processes. In fact, consider the stationary multivariate random process $\mathbf{x}(t) \in \mathbb{R}^p$ and suppose its covariance matrix $\mathbf{\Phi} \stackrel{\text{def}}{=} E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^{\mathrm{T}}]$ exists bounded. If $\mathbf{\Phi}$ is not diagonal, then the components of $\mathbf{x}(t)$ are statistically correlated. This second-order redundancy may be partially (or completely) removed by computing a linear operator $\mathbf{L}$ such that the new random signal defined by $\mathbf{y}(t) \stackrel{\text{def}}{=} \mathbf{L}^{\mathrm{T}}(\mathbf{x}(t) - E[\mathbf{x}]) \in \mathbb{R}^m$ has uncorrelated components, with $m \leq p$ arbitrarily selected. The operator $\mathbf{L}$ is known to be the matrix formed by the eigenvectors of $\mathbf{\Phi}$ corresponding to its largest eigenvalues [34]. The elements of $\mathbf{y}(t)$ are termed *principal components of* $\mathbf{x}(t)$; their importance is proportional to the corresponding eigenvalues $\sigma_i^2 \stackrel{\text{def}}{=} E[y_i^2]$ which are supposed to be arranged in descending order ($\sigma_i^2 \geq \sigma_{i+1}^2$).

The data-stream $\mathbf{y}(t)$ represents a compressed version of data-stream $\mathbf{x}(t)$;

after that the reduced-size data-stream has been processed (i.e. stored, retrieved, transmitted), it needs to be recovered, that is brought back to its original size. However, the principal-component based data reduction technique is not lossless, thus only an approximation $\hat{\mathbf{x}}(t)$ of the original data-stream may be recovered. As $\mathbf{L}$ is an orthonormal operator, an approximation of $\mathbf{x}(t)$ is given by $\hat{\mathbf{x}}(t) = \mathbf{L}\mathbf{y}(t) + E[\mathbf{x}]$; it minimizes the reconstruction error $E[||\mathbf{x} - \hat{\mathbf{x}}||^2]$ which equals $\sum_{k=m+1}^{p} \sigma_k^2$.

A simpler – yet interesting – application is Principal Subspace Analysis (PSA), which focuses on the estimation of an (orthonormal) basis of the subspace spanned by the principal eigenvectors without computing the eigenvectors themselves. The dual case of Minor Subspace Analysis is discussed in details in [23].

To this aim, we may define a criterion U as an Oja's criterion, as $J(\mathbf{W}) \stackrel{\text{def}}{=} k\text{tr}[\mathbf{W}^T \mathbf{\Phi} \mathbf{W}]$, to be maximized under the constraint of orthonormality of the connection matrix $\mathbf{W}$, where $k > 0$ is a scaling factor. In real-world applications the covariance matrix is unknown in advance (and its explicit estimation is to be avoided for computational-burden reasons), thus we may resort to its (rough) instantaneous approximation by replacing $\mathbf{\Phi}$ with $\mathbf{x}\mathbf{x}^T$.

4.2 Independent Component Analysis

Independent component analysis 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* [4, 11, 17, 18, 19]. 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 [11] 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 network as depicted in Figure 2. Here we suppose the source signal stream $\mathbf{s} \in \mathbb{R}^p$, observed linear mixture stream $\mathbf{z} \in \mathbb{R}^p$, thus mixing matrix $\mathbf{M}^T \in \mathbb{R}^{p \times p}$.

In the following experiments, the aim is to separate out m independent

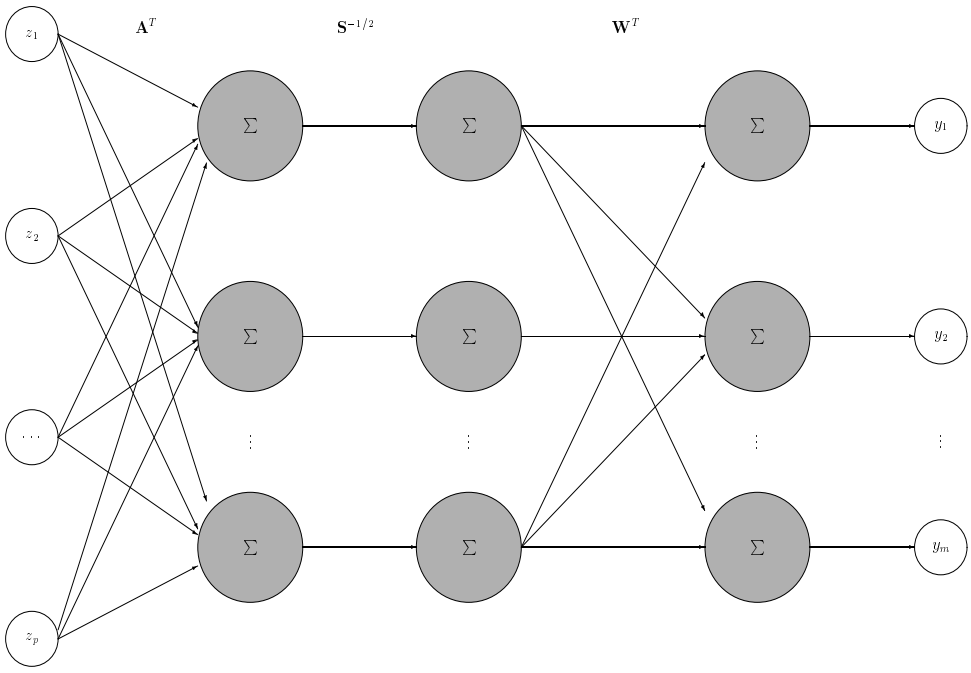


Figure 2: Three-layer neural architecture for blind source separation.

signals from their linear mixtures. To this aim, the following simple potential energy function may be used as optimization criterion [7, 12]:

$$U(\mathbf{W}) = \frac{k}{4} \sum_{i=1}^{m} E_{\mathbf{x}}[y_i^4] \ , \tag{12}$$

where k is a scaling factor. The resulting active force has the expression:

$$\mathbf{F} = -kE_{\mathbf{x}}[\mathbf{x}(\mathbf{x}^T\mathbf{W})^3] \ , \tag{13}$$

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

The whitening matrix pair $(\mathbf{S}, \mathbf{A})$ computes as follows: If $\mathbf{\Phi}_{zz}$ denotes the covariance matrix of the multivariate random vector $\mathbf{z}$, then $\mathbf{S}$ contains the eigenvalues and $\mathbf{A}$ contains (as columns) the corresponding eigenvectors of the covariance matrix. The whitened version of $\mathbf{z}$ is thus $\mathbf{x} = \mathbf{S}^{-1}\mathbf{A}^T\mathbf{z}$.

4.3 Performance indices description

In PSA estimation, the quantity $J(\mathbf{W})$ itself is a valid index of system performance.

In ICA, since the overall source-to-output matrix $\mathbf{K} \stackrel{\text{def}}{=} \mathbf{W}^T\mathbf{S}^{-1/2}\mathbf{A}^T\mathbf{M}^T \in \mathbb{R}^{m\times p}$ should become as quasi-diagonal (i.e. such that only one entry per row

and column differs from zero) as possible, we might take as convergence measure the general Comon time-index [11]. 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, because pre-whitening and orthonormal ICA inherently prevent the different neurons from sharing the same source signals. Consequently, we may employ the reduced criterion:

$$F(t) \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{m}\sum_{j=1}^{p} K_{ij}^2(t) - \sum_{i=1}^{m} \max_k\{K_{ik}^2(t)\}}{\sum_{i=1}^{m} \max_k\{K_{ik}^2(t)\}} \ , \tag{14}$$

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

5 Experimental Results

In order to test the effectiveness of the proposed algorithm, two experiments have been performed on the mentioned PSA and ICA problems.

5.1 Experiment on subspace iteration

In this experiment, a synthetic random process $\mathbf{x}$ with $p = 6$ components has been generated which possesses zero-mean Gaussian statistics with covariance matrix $\mathbf{\Phi} = \frac{1}{2}(\mathbf{H}_6 + \mathbf{H}_6^T)$, where $\mathbf{H}_6$ denotes the sixth-order Hilbert matrix; in this way $\mathbf{\Phi}$ is symmetric and positive definite.

We wish to estimate (an orthonormal basis of) the principal subspace associated to the input signal of dimension $m = 2$ and suppose to have $2,000$ samples of the input signal available.

In order to iterate with the new discretized MEC algorithm we chose parameters values $\nu = 0.5$, $k = 0.5$ and $h = 0.05$. The result of iterations is illustrated in the Figure 3. The obtained numerical results in the approximated-covariance (stochastic) case are in excellent agreement with the expected result.

5.2 Experiment on independent component analysis

In this experiment we considered five gray-scale natural images as source signals. Two of these are leptokurtotic signals (i.e. they have positive kurtoses) and the remaining three are platikurtotic signals (they possess negative kurtosis). The original images as well as their mixtures are shown in the Figure 4. By properly

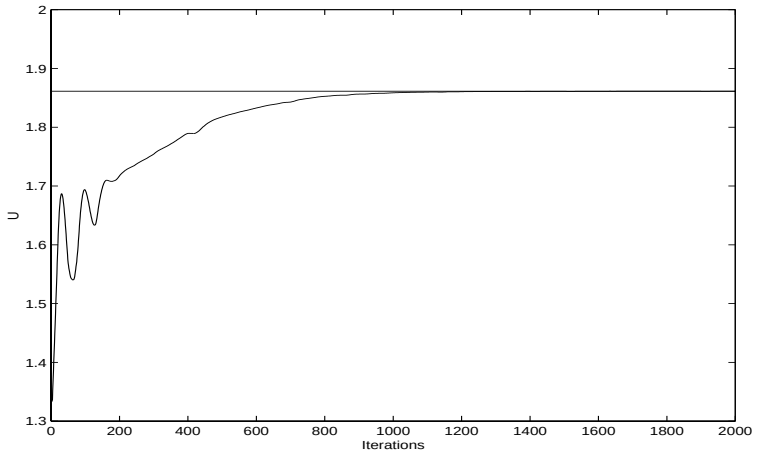


Figure 3: Results of the iteration in the PSA problem. The straight line represents the known exact value of the function $U(\cdot)$ at optimum.

selecting the sign of the constant k it is possible to extract the groups of images from their linear mixtures.

In the present case we chose $p = 5$, $k = -0.06$ and $m = 3$. The images' size is 100×100 pixels thus we have a total of $10,000$ samples that the iteration may be carried over. The other learning parameters were $\nu = 0.6$ and $h = 0.01$.

Figure 5 shows the value of performance index F during iteration. It reaches a substantially low value which reflects the quasi-diagonality of the separation product $\mathbf{K}$ depicted in Figure 6.

The final appearance of the network output signals, shown in the Figure 7, confirms the good level of separation achieved.

6 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 whose main drawback was the inefficient representation of the involved quantities. With the aim to lessen the computational burden pertaining to this algorithm, we proposed here a novel formulation of the learning equations based on an efficient parameterization of the angular-speed tensor.

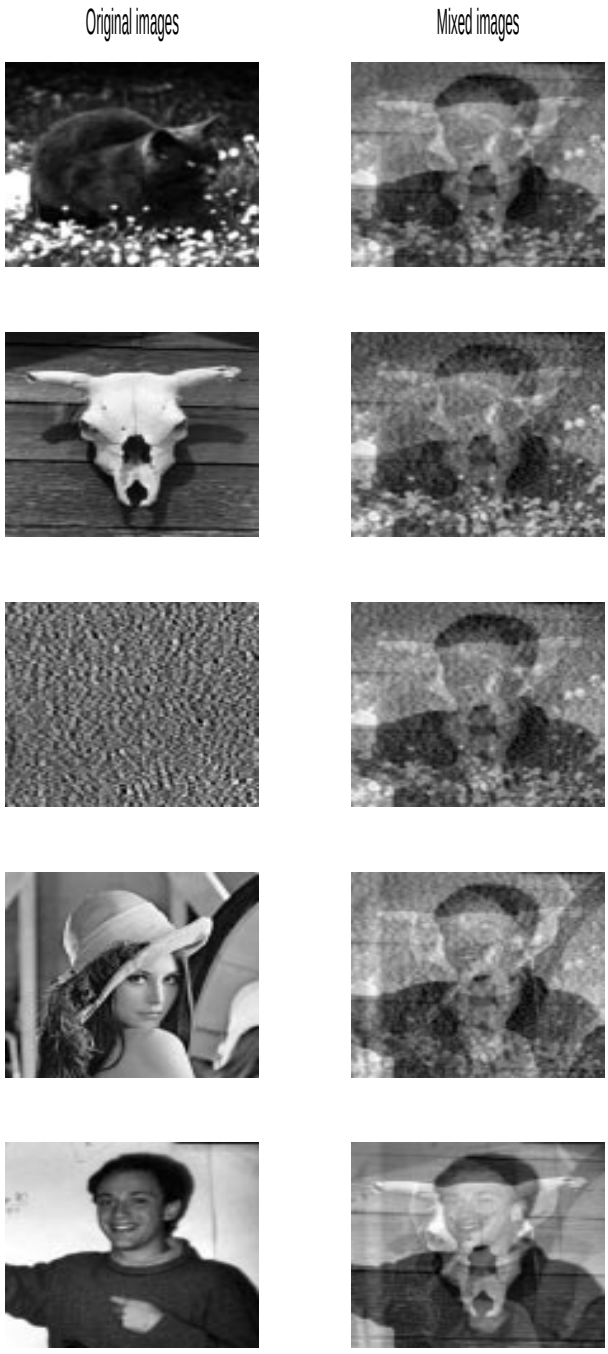


Figure 4: Original images and their mixtures in the ICA problem.

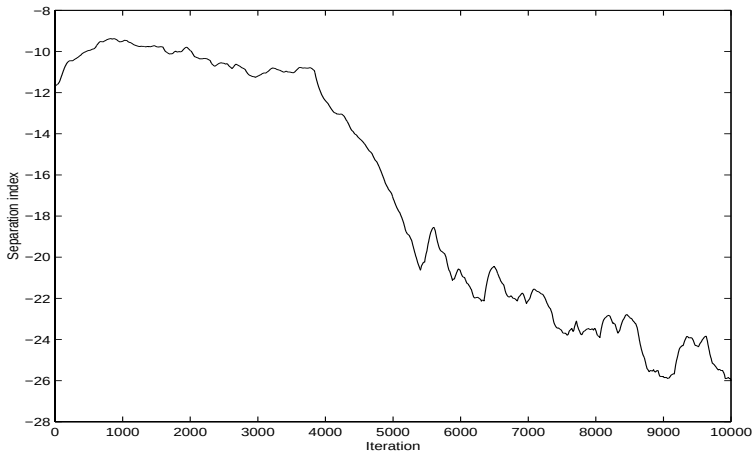


Figure 5: Index F during iteration on the ICA problem.

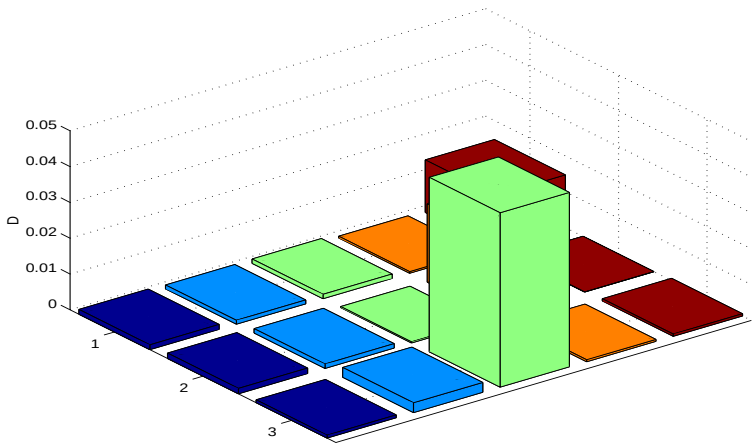


Figure 6: Separation product $\mathbf{K}$ after iteration on the ICA problem.

Figure 7: Three leptokurtotic signals extracted in the ICA problem.

References

[1] S. Affes and Y. Grenier, *A Signal Subspace Tracking Algorithm for Speech Acquisition and Noise Reduction with a Microphone Array*, Proc. of IEEE/IEE Workshop on Signal processing Methods in Multipath Environments, pp. 64 – 73, 1995

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

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

[4] 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

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

[6] 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

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

[8] 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

[9] E. Celledoni and B. Owren, *A class of intrinsic schemes for orthogonal integration*, Technical Report Numerics No. 1/2001, The Norwegian University of Science and Technology, Trondheim, Norway, 2001. To appear in SIAM J. Num. Anal.

[10] E. Celledoni and A. Iserles, *Approximating the exponential form of a Lie algebra to a Lie group*, Math. Comp. 69, pp. 1457 – 1480, 2000

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

[12] 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

[13] 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

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

[15] 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

[16] 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

[17] 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

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

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

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

[21] 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

[22] S. Fiori, *Unsupervised Neural Learning on Lie Group*, International Journal of Neural Systems. Accepted for publication

[23] S. Fiori, *A Minor Subspace Algorithm Based on Neural Stiefel Dynamics*, International Journal of Neural Systems. Accepted for publication

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

[25] 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

[26] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1996

[27] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, Springer series in Computationa Mathematics, Springer, 2002

[28] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Reprinted edition, 1995

[29] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, and A. Zanna: *Lie-group methods*, Acta Numerica, Vol. 9, pp. 215 – 365, 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] 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

[33] 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

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

[35] 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

[36] B. Parlett, *A recurrence among the elements of functions of triangular matrices*, Linear Algebra and its Applications, Vol. 14, pp. 117 – 121, 1976

[37] 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

[38] 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

[39] I.-T. Um, J.-J. Wom, and M.-H. Kim, *Independent component based Gaussian mixture model for speaker verification*, Proc. of Second International ICSC Symposium on Neural Computation (NC), pp. 729 – 733, 2000

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

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

[42] 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

[43] 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