International Journal of Neural Systems, Vol. 12, Nos. 3 & 4 (2002) 000–000

UNSUPERVISED NEURAL LEARNING ON LIE GROUP

SIMONE FIORI
Neural Network and Signal Processing Group, Faculty of Engineering,
Perugia University Via Pentima Bassa, 21-05100 Terni, Italy
sfr@unipg.it

Received 12 April 2002
Accepted 25 June 2002

The present paper aims at introducing the concepts and mathematical details of unsupervised neural learning with orthonormality constrains. The neural structures considered are single non-linear layers and the learnable parameters are organized in matrices, as usual, which gives the parameters spaces the geometrical structure of the Euclidean manifold. The constraint of orthonormality for the connection-matrices further restricts the parameters spaces to differential manifolds such as the orthogonal group, the compact Stiefel manifold and its extensions. For these reasons, the instruments for characterizing and studying the behavior of learning equations for these particular networks are provided by the differential geometry of Lie groups. In particular, two sub-classes of the general Lie-group learning theories are studied in detail, dealing with first-order (gradient-based) and second-order (non-gradient-based) learning. Although the considered class of learning theories is very general, in the present paper special attention is paid to unsupervised learning paradigms.

Keywords: Learning with orthonormality constraints; Lie group; differential geometry; multidimensional signal processing; blind signal processing.

1. Introduction

Multidimensional signal processing by neural networks is an emerging research field concerned with advanced multiple signal treatment techniques. Neural computation is considered as a new area in the information processing field, sometimes referred to as soft-computation, which deals with adaptive, parallel and localized (distributed) signal/data processing. The artificial neural networks have been inspired by the biological neural systems and the organization of the structures of the brain, and their usefulness in engineering lies in their ability of self-designing to solve a problem by learning the solution from data.

Neural learning usually takes place in a parameter space which often is endowed with a specific geometrical structure. In recent years, learning on a geometrical structure has attracted considerable interest, and differential geometry and linear algebra have been recognized to play a fundamental role in gaining a deep insight into the behavior of learning systems.[2]

Irrespective of the nature of learning (i.e. supervised or unsupervised), the adaptation of a neural network may often be formally conceived of as an optimization problem: A criterion or objective function describes the task to be performed by the network, and a numerical optimization procedure allows adapting the network's tunable parameters (e.g. connection weights, biases, neurons' internal parameters). This means that neural network learning may be interestingly conceived of as a search or non-linear programming problem in a parameter space, which is usually wide. Any pre-knowledge about the searched optimal solution, that is, the optimal configuration of the selected neural network with respect to the task at hand and some performance metrics, might be advantageously exploited in order to narrow the search space.

In supervised learning, recent research studies have clearly illustrated how it is often beneficial to incorporate additional knowledge in the neural network architecture or learning rules,[15,28,47] while in unsupervised learning these benefits have been investigated less extensively.[2,3,24] Usually, the methods exploited in order to handle these modified neural network tasks are drawn from the classical constrained optimization field and rely on the Lagrange multipliers method, the penalty or barrier techniques, and from classical numerical algebra techniques, such as deflation/renormalization,[19] the Gram-Schmidt orthogonalization procedure or the projection over the orthogonal group.[42,58]

We propose here a quite different perspective, well suited for both supervised and unsupervised learning tasks. The embedding of pre-knowledge in the network learning rule is regarded from the point of view of the geometry of networks parameters spaces: The considered constraints are taken into account by modifying the intimate geometric structure of a network parameter space, which gives rise to classes of learning rules compatible with these constraints, by properly describing the analytical structure of parameters spaces.

Numerous statistical learning problems lead to smooth nonlinear optimization problems over parameter spaces with the structural properties of orthonormal manifolds. This class, up to now, has been insufficiently examined — in spite of its importance — either from the theoretical or methodological point of view. This paper aims at presenting general results about a new class of learning rules for linear as well non-linear neural layers, which allows the weight-matrix, describing the connection-strengths between the inputs and the neurons, to learn in unsupervised frameworks under the constraints of *orthonormality*, namely, when the network parameters can be arranged in vectors of constant lengths and orthogonal to each other. This paper follows our preceding work,[21] devoted to the first analysis of learning rules on Stiefel–Grassman manifold and on a wide bibliographical investigation in order to show the close relationships among existing contributions, and Ref. 23, devoted to a wide numerical comparison of orthonormal neural signal processing techniques in the principal/independent component analysis field. The present paper answers to the necessity of a more general treatment of the learning theories with orthonormal constraints and of a more detailed investigation of specific examples, from which useful hints on the general applicability of the proposed theory emerge.

Prior to proceeding with the detailed discussion of the mentioned concepts, we wish to briefly formalize the problem at hand and to present a brief survey of existing contributions (which have sparsely tackled it).

1.1. *First formalization of learning with orthonormality constraints*

To formally explain the concept of learning with orthonormality constraints, let us define the matrix set:

$$\mathcal{H}_m^{p\times q} \stackrel{\text{def}}{=} \{\mathbf{M} \in \mathcal{R}^{p\times q} | \mathbf{M}^T\mathbf{M} = m^2\mathbf{I}_q\}\,, \qquad (1)$$

where $\mathbf{I}_q$ represents a $q\times q$ identity matrix, and $q \leq p$. By definition m must differ from zero and is allowed to vary through time, i.e. $m = m(t)$, where $m(t)$ is a function differentiable at least once. Here we consider $\mathbf{M}(t)$ as the connection matrix of a neural layer, connecting the p inputs to the q neurons. An exemplary representation of the set $\mathcal{H}_m^{p\times q}$ is given in Fig. 1.

Also, we wish to define a set of learning rules which retain the maximal arbitrariness, but for guaranteeing the connection matrix to always belong to the manifold $\mathcal{H}_m^{p\times q}$. Formally, we can thus define the following class:

$$\mathcal{F} \stackrel{\text{def}}{=} \{\mathcal{L}_U(\mathbf{M}) | \forall t \in \mathcal{T} : \mathbf{M}(t) \in \mathcal{H}_{m(t)}^{p\times q}\}\,, \qquad (2)$$

where $\mathcal{L}_U(\mathbf{M})$ is a generic learning algorithm for a layer with connection pattern $\mathbf{M}$, $\mathcal{T}$ is a time-interval that the algorithm runs within, and U is an objective function whose iterative optimization drives network's learning.

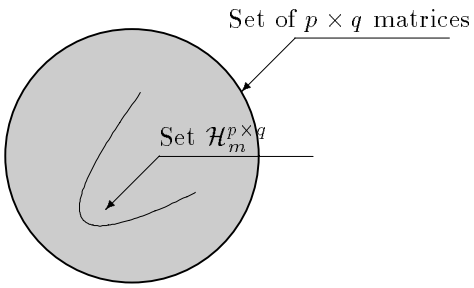

Fig. 1. Exemplary representation of the manifold $\mathcal{H}_m^{p\times q}$.

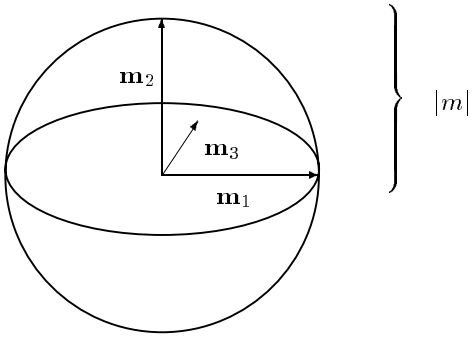

Fig. 2. Exemplary representation of the positions of the column-vectors $\mathbf{m}_i$ of the matrix $\mathbf{M}$ for $p = q = 3$.

Any learning/optimization algorithm within the class $\mathcal{F}$ is characterized by a very fundamental property. Differentiating both members of $\mathbf{M}^T(t)\mathbf{M}(t) = m^2(t)\mathbf{I}_q$ with respect to the time yields:

$$\frac{d(\mathbf{M}^T\mathbf{M})}{dt} = \dot{\mathbf{M}}^T\mathbf{M} + \mathbf{M}^T\dot{\mathbf{M}} = \lambda\mathbf{I}_q\,,\ 2\dot{m}m = \lambda\,. \quad (3)$$

The time-function $\lambda(t)$ partially describes the dynamics of the weight-matrix when traveling on the manifold $\mathcal{H}_m^{p\times q}$. An exemplary picture of the positions of the column-vectors $\mathbf{m}_i$ of the matrix $\mathbf{M}$ is illustrated in Fig. 2 for $p = q = 3$.

These very basic concepts constitute the basis for developing a detailed analysis of learning rules subject to orthonormality constraints: By considering these constraints we will resort to learning rules, expressed as ordinary matrix differential equations in the flow $\mathbf{M}(t)$, that can be studied in the context of Lie-group equations. In the next sections, we shall present general theoretical results and some examples of learning rules in $\mathcal{F}$, namely first-order and second-order learning equations, which prove to be compatible with learning with orthonormal constraints. As the fundamental mathematical theory for developing learning paradigms with the constraints of orthonormality is the one relying on Lie-group formalization, we shall refer to $\mathcal{F}$ as the set of learning theories on Lie group (LLG).

1.2. *Connections of learning with orthonormality constraints to other theories*

Learning with orthonormality constraints naturally occurs in many adaptation problems related to neural networks, signal processing by adaptive nonlinear systems and optimization systems. It is also worth recalling that the singular value decomposition theorem is a widely known mathematical tool that allows recasting any linear problem into a pair of orthonormal problems.[3]

The general learning framework proposed in the following borrows fundamental mathematical concepts derived by the general theoretical contributions by Stiefel[52] about Stiefel manifold (which needs however to be generalized here), Aluffi-Pentini *et al.*[4] about optimization via second-order dynamical systems and Brockett[8] concerning first-order dynamical systems on double-orthogonal group. Also, application-oriented learning theories have been considered, with special reference to principal component analysis and independent component analysis by pre-whitening.

The present paper deals with the theoretical aspects of learning with orthonormality constraints and, in essence, aims at presenting a general framework that encompasses the excellent contributions appearing in the scientific literature in recent years, which have seen many engineering applications ranging from digital signal processing to automatic control, from pattern recognition to numerical solutions of solid-state physics problems for materials.

Interesting applications are the eigenvalue and generalized eigenvalue problems, Rayleigh quotient iteration, CS decomposition, optimal linear compression, noise reduction and signal representation by principal/minor component analysis and principal/minor subspace decomposition/ tracking;[1,14,18,44,57,58] the simulation of bulk materials;[17] minimal linear system realization from noise-injection measured data and invariant subspace computation;[17,39] blind source separation by signal pre-whitening;[10,13,20,22,32,33,46] optimal denoising by sparse coding shrinkage and local manifold projection;[34,45] direction of arrival estimation;[1] best signal basis search;[40] linear programming and sequential quadratic programming;[8,17] optical character recognition by transformation-invariant neural networks;[51] electrical networks fault detection;[36] spectral analysis of unevenly sampled data;[53] the solution of the orthogonal Procrustes problem,[a] which

[a]The orthogonal Procrustes problem is a particular case of the Penrose regression problem, which arises e.g. in multivariate data analysis.[29]

is a minimization problem defined on $\mathcal{H}^{p\times q}$, arising in many fields as in image processing, that has no known analytical solutions when q differs from 1 and p;[17] the holography-like memories (such as the *holophone*;[48,55] the synthesis of digital filters by improved total least-squares technique;[26] the speaker verification by the independent component analysis;[54] the general problem of data analysis and visualization (as e.g. artifact removal from EEG traces and analysis of fMRI data) and data mining;[27,38] the adaptive image coding by learning with respect to experiences and perspectives (LEP),[41] and the development of a theory of geometric constraints on neural activity for natural three-dimensional movement.[59]

1.3. *Recent selected applications*

In addition to the above-recalled connections with other theories, we wish to cite here three selected meaningful recent applications (not necessarily appearing in the neural network literature).

First, let us recall that in independent component analysis (ICA), observed signal models are considered, namely $\mathbf{x}(t) = \mathbf{A}^T\mathbf{s}(t)$, where $\mathbf{s}(t)$ is a vector random field in $\mathcal{R}^q$, $\mathbf{x}(t)$ is a vector random field in $\mathcal{R}^p$, and $\mathbf{A}$ is the mixing matrix in $\mathcal{R}^{q\times p}$. Neural ICA is an emerging signal processing technique that allows us to recover the mixed sources from sensor observations only. Hereafter we consider indeed the orthogonal ICA problem, which refers to the case that the mixture is due to an orthogonal mixing operator; it is important to note that any ICA problem can be brought back to this case by so-called mixture pre-whitening.[3]

According to Ref. 3, depending on the number of observations p and the number of independent sources q, the following classification of the ICA problem is considered:

- Case $p < q$: In this case the ICA problem is termed *over-complete*, because the number of observations is not sufficient to cover the whole set of independent components. In this case, by hypothesis the mixing operator is $\mathbf{A} \in \mathrm{St}(q, p, \mathcal{R})$, thus some maximum-likelihood-based estimation algorithm, with optimization on the Stiefel manifold, can be employed in order to estimate the mixing operator $\mathbf{A}$ as well as the source-signal stream $\mathbf{s}(t)$;[3,9]
- Case $p = q$: This is the standard ICA case, where the number of observations is exactly equal to the number of components. In this case source separation may be achieved both through a model-parameters inference technique and through a neural network, described by $\mathbf{y}(t) = \mathbf{M}^T(t)\mathbf{x}(t)$, with connection matrix $\mathbf{M} \in \mathrm{SO}(p, \mathcal{R})$ (see e.g. Ref. 5);
- Case $p > q$: In this case the ICA problem is termed *under-complete*, because the number of observation exceeds the set of independent components. The connection-matrix of the neural network employed to separate out the independent contributions is $\mathbf{M} \in \mathrm{St}(p, q, \mathcal{R})$, thus neural learning techniques relying on Stiefel-manifold optimization may be employed.[23]

In the cases of under- and over-complete ICA, the proposed representation, in terms of learning rules of the class $\mathcal{F}$, is therefore particularly meaningful. Algorithms for performing independent component analysis in these cases, based on information theory, have been devised and surveyed by Amari in Ref. 3.

Second, let us consider the dynamic texture recognition approach presented in Ref. 50. Dynamic textures are sequences of images that exhibit some form of temporal stationarity. While a vast literature exists about recognition based on geometry and photometry, recognizing scenes based upon their dynamics is still in its infancy: In particular, the problem of recognizing and classifying dynamic textures can be posed in the space of dynamical systems (autoregressive models) where each dynamic texture is uniquely represented.

In Ref. 50, images of stationary processes are represented as the output of a stochastic dynamical model. The model is learned from the data, and recognition is performed in the space of models. Formally, an image is represented as the output $\mathbf{y}(t) \in \mathcal{R}^p$ of the following discrete-time Gauss-Markov ARMA model:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{v}(t), \quad \mathbf{x}(0) = \mathbf{x}_0\,,$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{w}(t)\,,$$

with $\mathbf{x}(t) \in \mathcal{R}^q$ being the state-space vector of the model, $\mathbf{v}(t) \in \mathcal{R}^q$ and $\mathbf{w}(t) \in \mathcal{R}^p$ being two zero-mean Gaussian random vector-streams, and $\mathbf{A} \in \mathcal{R}^{q\times q}$ and $\mathbf{C} \in \mathcal{R}^{p\times q}$ ($p \gg q$) are the state-transition matrix and the output transformation matrix, respectively.

The choice of the model matrices is not unique (there exist infinitely many models that give rise to the same result starting from suitable initial states). Having in mind the necessity of defining a measure of distance between two ARMA models of that kind, which is a fundamental requirement for classification purposes,[50] it can be proven that selecting the realization that makes matrix $\mathbf{C}$ orthonormal gives rise to a really well-suited representation. In fact, by exploiting the geometrical structure of the space of orthonormal $p \times q$ matrices, the authors of Ref. 50 are able to develop a powerful theory of probability distributions on the orthonormal manifold in order to establish a suitable optimal classification scheme based on maximum-likelihood ratios. Of course, even if this is outside the scope of the present paper, a formulation of the solution to this problem within the framework of recurrent neural networks could be interesting.

Third, we wish to briefly survey the applied-physics problem arising in electronic structure computation; in particular, we address the problem of *ab initio* calculation of electronic structures within the local density approximation, which uses only the charge and mass of electrons and atomic nuclei as input.[17] This problem requires for example the study of the behavior of *thousands* of atoms of defects in glasses, complexes of extended crystals and large molecules.

Formally speaking, the mentioned problem consists in finding the smallest eigenvalue E_0 of the Scrödinger equation in the space of the $3N$-dimentional functions $\mathbb{H}$:

$$\mathbb{H}[\psi] = E_0 \psi \,,$$

where $\mathbb{H}$ represents the Hamiltonian operator and N is the number of electrons in the electronic system under analysis, which takes into account the Laplacian operator, the potential function due to the nuclei and inner electrons and the Coulomb interactions.[17] A direct discretization of this equation leads to an infeasible eigenvalue problem.

The fundamental method for solving the eigenvalue problem for the Hamiltonian operator relies on the fact that, in certain situations, the ground-state solution to the above Scrödinger equation coincides to the solution of a minimization problem defined on a (quadratic) energy function over all possible sets of N three-dimensional electronic-orbital functions, under the constraint of orthonormality.

This observation clearly shows the connection of the physical problem of electronic structure computation with the theory of optimization under the constraint of orthonormality. Again, even if this topic falls outside the scope of the present paper, a neural-network formulation of the ground-state calculation problem by electronic-orbital learning could be fruitful.

The mentioned examples clearly illustrate the importance of learning and optimization in the scientific field and motivated the present research work.

Notation: Symbol tr[·] denotes the trace of the matrix contained within, while symbol det[·] denotes the determinant of the matrix contained within. A matrix $\mathbf{A}$ is said to be skew-symmetric if $\mathbf{A}^T = -\mathbf{A}$. A matrix-to-matrix operator $\mathbf{S}[\cdot]$ is termed skew-symmetric if it is of the form $\mathbf{N}^T[\cdot] - \mathbf{N}[\cdot]$, with $\mathbf{N}[\cdot]$ being an arbitrary matrix-to-matrix operator that symmetrizes if an only if the argument is symmetric. Symbol $\dot{\mathbf{x}}$ stands for $d\mathbf{x}/dt$. Also, $\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})]$ denotes mathematical expectation of $f(\mathbf{x})$ with respect to $\mathbf{x}$.

2. General Results on Learning on Lie Group

Let us consider a neural layer described by:

$$\mathbf{y} = \mathbf{G}[\mathbf{M}^T \mathbf{x} + \mathbf{m}_0] \,, \tag{4}$$

where $\mathbf{x}(t) \in \mathcal{R}^p$ is the input stream, $\mathbf{y}(t) \in \mathcal{R}^q$ represents the layer's response, $\mathbf{G}[\cdot]$ is a non-linear diagonal operator and $\mathbf{m}_0$ is a biasing vector arbitrarily adapted. In order to adapt the connection matrix $\mathbf{M}$ so that it keeps within $\mathcal{H}_m^{p\times q}$, we consider two subclasses of learning laws of $\mathcal{F}$: The first-order LLG systems, where only $\dot{\mathbf{M}}$ is involved, and the second-order LLG systems, where $\ddot{\mathbf{M}}$ is considered. Furthermore, in this paper we consider autonomous learning rules only, that means taking $U(\mathbf{M}) = \mathbb{E}_{\mathbf{x}}[u(\mathbf{x}, \mathbf{y}, \mathbf{M})]$; here $u(\cdot, \cdot, \cdot)$ represents a measure of the performance of the neural system with respect to the task it should perform after training, on the input pattern $\mathbf{x}$ when the network is on state $\mathbf{M}$. This formulation accounts for both supervised and unsupervised learning, even if in this paper examples and applications concentrate on the unsupervised learning mode.

2.1. *Some useful notes on differential geometry of Lie groups*

The differential geometry of smooth manifolds provides the necessary notions we need to express the learning equations in a compact and suggestive way; the aim of this section is to recall some useful concepts from differential geometry with examples.[7]

Given the manifold $\mathcal{M}$ and a point $\mathbf{w} \in \mathcal{M}$, we denote by $T_{\mathbf{w}}\mathcal{M}$ the tangent space of $\mathcal{M}$ at $\mathbf{w}$, while $T\mathcal{M} \stackrel{\text{def}}{=} \bigcup_{\mathbf{w}\in\mathcal{M}} T_{\mathbf{w}}\mathcal{M}$ is the tangent bundle of $\mathcal{M}$. A vector field $\mathbf{F}$ on $\mathcal{M}$ is a section of $T\mathcal{M}$, that is, to each point $\mathbf{w} \in \mathcal{M}$ it associates a vector $\mathbf{F}_{\mathbf{w}} \in T_{\mathbf{w}}\mathcal{M}$.

Lie groups are special manifolds that are endowed with an internal binary operation, usually referred to as *group multiplication* which is compatible with the topology; this means that if $a(t)$ and $b(t)$ are two smooth (differentiable infinitely many) one-parameter families of elements of the Lie group G, then also $a(t)b(t)$ and $a^{-1}(t)$ are in G and smooth. The Lie algebra $\mathbf{g}$ is the tangent space to G at the identity. Because of its structure, in a Lie group G the tangent bundle TG can be identified with the product $\mathbf{g}$. Some examples of Lie groups and associate Lie algebras are: The general linear group $\mathrm{GL}(p, \mathcal{R})$ of the invertible $p \times p$ real-valued matrices with $\mathbf{gl}(p, \mathcal{R})$ being the set of $p \times p$ real-valued matrices; the special orthogonal group $\mathrm{SO}(p, \mathcal{R})$ of the orthogonal matrices with unitary determinant, with $\mathbf{so}(p, \mathcal{R})$ being the set of $p \times p$ skew-symmetric matrices; the special linear group $\mathrm{SL}(p, \mathcal{R})$ of invertible matrices with unitary determinant having as Lie algebra $\mathbf{sl}(p, \mathcal{R})$ the set of traceless $p \times p$ matrices.

In the present paper we are interested in matrix groups. The algebra $\mathbf{g}$ of a Lie group G is endowed with an internal operation, termed *commutator* operator or Lie bracket $[\cdot, \cdot] : \mathbf{g} \times \mathbf{g} \to \mathbf{g}$ defined as $[\mathbf{A}, \mathbf{B}] \stackrel{\text{def}}{=} \mathbf{AB} - \mathbf{BA}$; also, there exists an important map from $\mathbf{g}$ to G, the exponential $\exp(\cdot) : \mathbf{g} \to G$ defined as $e^{\mathbf{A}} = \sum_{k=0}^{\infty}(\mathbf{A}^k/k!)$. A particular matrix differential equation described by $\dot{\mathbf{A}}(t) = \mathbf{B}(t)\mathbf{A}(t)$, with $\mathbf{A}(t) \in G$ and $\mathbf{A}(0) = \mathbf{I}$, and with $\mathbf{B}(t) \in \mathbf{g}$ is termed Lie-group equation.

An useful special Lie-group equation arises when $G = \mathrm{SO}(p, \mathcal{R})$ and $\mathbf{g} = \mathbf{so}(p, \mathcal{R})$. Let us consider, as an example, the matrix-field $\mathbf{A}(t) \in \mathrm{SO}(2, \mathcal{R})$ given by:

$$\mathbf{A}(t) = \begin{bmatrix} \cos\theta(t) & \sin\theta(t) \\ -\sin\theta(t) & \cos\theta(t) \end{bmatrix}.$$

It satisfies the Lie-group equation; in fact, entry-by-entry derivation gives:

$$\begin{aligned} \dot{\mathbf{A}}(t) &= \begin{bmatrix} -\sin\theta(t)\dot{\theta}(t) & \cos\theta(t)\dot{\theta}(t) \\ -\cos\theta(t)\dot{\theta}(t) & -\sin\theta(t)\dot{\theta}(t) \end{bmatrix} \\ &= \begin{bmatrix} 0 & \dot{\theta}(t) \\ -\dot{\theta}(t) & 0 \end{bmatrix} \begin{bmatrix} \cos\theta(t) & \sin\theta(t) \\ -\sin\theta(t) & \cos\theta(t) \end{bmatrix} \\ &= \mathbf{B}(t)\mathbf{A}(t)\,. \end{aligned} \tag{5}$$

An important manifold we are interested in is the *Stiefel manifold* characterized by:

$$\mathrm{St}(p,\, q,\, \mathcal{R}) \stackrel{\text{def}}{=} \{\mathbf{A} \in \mathcal{R}^{p\times q} | \mathbf{A}^T\mathbf{A} = \mathbf{I}_q\}\,. \tag{6}$$

The main properties of $\mathrm{St}(p,\, q,\, \mathcal{R})$ are summarized as: (1) $\mathrm{St}(p,\, q, \mathcal{R})$ is a smooth, compact manifold of dimension $pq - (1/2)q(q+1)$; (2) the tangent space is $T_{\mathbf{A}}\mathrm{St}(p,\, q,\, \mathcal{R}) = \{\mathbf{B} \in \mathcal{R}^{q\times p} | \mathbf{A}^T\mathbf{B} + \mathbf{B}^T\mathbf{A} = \mathbf{0}\}$; (3) the normal space $T^{\perp}_{\mathbf{A}}\mathrm{St}(p,\, q,\, \mathcal{R})$ writes $T^{\perp}_{\mathbf{A}}\mathrm{St}(p,\, q,\, \mathcal{R}) = \{\mathbf{A}\boldsymbol{\Lambda} \in \mathcal{R}^{q\times p} | \boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T \in \mathcal{R}^{p\times p}\}$.

It is important to clarify that the defined set $\mathcal{H}^{p\times q}_m$, the Stiefel manifold $\mathrm{St}(p,\, q,\, \mathcal{R})$ and the orthogonal group $\mathrm{SO}(p,\, \mathcal{R})$ are closely related but exhibit some remarkable differences. In particular, they all are submanifolds of the Euclidean manifold $\mathcal{R}^{p\times q}$ and are mutually encapsulated as illustrated in the Fig. 3. In particular, it is worth pointing out that $\mathcal{H}^{p\times q}_m$ coincides with the Stiefel manifold only when $m = \text{costant} = 1$, otherwise it has a more complex structure. For instance, let us suppose $p = 3$, $q = 1$ and $m(t) = 4 + \cos(2\pi t)$: In this case, the considered Stiefel manifold is a unit sphere while $\mathcal{H}^{3\times 1}_{m(t)}$ is a *pulsating* sphere which vibrates with a frequency of one cycle per second; also, let us consider a material particle sliding on the two surfaces: With reference

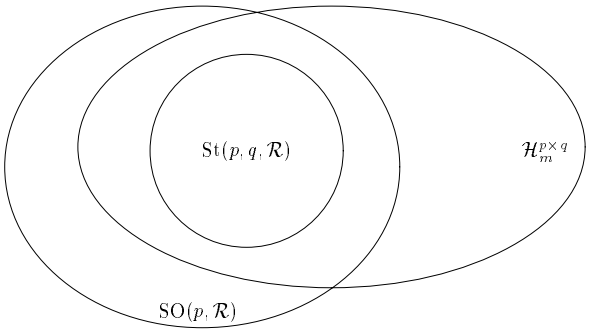

Fig. 3. Mutual encapsulation of manifolds $\mathcal{H}^{p\times q}_m$, $\mathrm{St}(p,\, q,\, \mathcal{R})$ and $\mathrm{SO}(p,\, \mathcal{R})$.

to an observer fixed in the center of the spheres, the particle moving over $\mathrm{St}(3,1,\mathcal{R})$ has velocity tangent to the sphere, while the particle sliding on $\mathcal{H}^{3\times 1}_{m(t)}$ possesses a non-null radial component. This simple example clearly illustrates that learning on the $\mathcal{H}^{p\times q}_{m}$ manifold deserves a generalized analysis with respect to learning on the Stiefel/orthogonal groups.

2.2. *First-order learning on Lie group*

A special subset of LLG algorithms is that of the "gradient based" ones, namely of those algorithms $\mathcal{L}_U(\mathbf{M})$ of the form:

$$\frac{d\mathbf{M}}{dt} = -\frac{\partial U}{\partial \mathbf{M}}\,, \quad \mathbf{M}(0) \in \mathcal{H}^{p\times q}_{m_0}\,, \tag{7}$$

where $\nabla U = \partial U/\partial \mathbf{M}$ is the Jacobian of the criterion function U (or its ordinary gradient) with respect to the variables in $\mathbf{M}$. Clearly, since the dynamics of $\mathbf{M}(t)$ is governed by U, the function $m(t)$ will in general not be free. In particular, in this case it is easy to see that $\lambda_U(t) = -(2/q)\mathrm{tr}[\mathbf{M}^T(t)\nabla U(t)]$.

Not every criterion U gives rise to a first-order LLG: In order to ensure that the associated gradient-based learning equation belongs to the class $\mathcal{F}$, the criterion U has to satisfy an important condition derived from Eq. (3), which clearly depends on the form of the measure $u(\cdot,\cdot,\cdot)$ and on the structure of the network described by operator $\mathbf{G}[\cdot]$.

Theorem 1

Let us parameterize a trajectory $\mathcal{P}$ of learning system (7) by means of the temporal parameter t, i.e. $\mathcal{P} = \{\mathbf{M}(t)|t \in \mathcal{T}\}$. A necessary and sufficient condition in order for the criterion U in Eq. (7) to generate a LLG is that its Jacobian possesses the structure:

$$\frac{\partial U}{\partial \mathbf{M}} = \left(\frac{d\mathbf{\Theta}}{dt} - \frac{d\alpha}{dt}\mathbf{I}_q\right)\mathbf{M}\,, \tag{8}$$

where $\mathbf{\Theta}(t) \in \mathcal{R}^{p\times p}$ is a matrix-stream such that $\mathbf{M}^T\dot{\mathbf{\Theta}}\mathbf{M}$ is skew-symmetric and $\alpha(t) \in \mathcal{R}$ is a scalar-stream. The latter function relates to $\lambda_U(t)$ and $m(t)$ by:

$$\frac{d\alpha}{dt} = \frac{\lambda_U}{2m^2} = \frac{\dot{m}}{m}\,. \tag{9}$$

The structure (8) ensures the pathway $\mathcal{P}$ to be a subset of $\mathcal{H}^{p\times q}_{m}$.

Proof

Necessity. Let us denote by $\mathbf{M}_0 \in \mathrm{St}(p,\,q,\,\mathcal{R})$ the configuration of the system for $t = 0$. The dynamics over $\mathcal{H}$ restricts the set of admissible configurations to matrices $\mathbf{M}(t)$ whose columns have equal norm and are orthogonal to each other. This suggests that $\mathbf{M}(t)$ must be related to $\mathbf{M}_0$ by the relationship:

$$\mathbf{M}(t) = \beta(t)\mathbf{R}(t)\mathbf{M}_0\,, \tag{10}$$

where $\beta(t) > 0$ denotes the stretch of vectors $\mathbf{m}_i(t)$ with respect to $\mathbf{m}_i(0)$ and $\mathbf{R}(t) \in \mathrm{SO}(p,\,\mathcal{R})$ is a rotation matrix such that $\mathbf{R}^{-1}(t)$ aligns $\mathbf{M}(t)$ to $\mathbf{M}_0$. If $\beta(t) > 1$ then the column-vectors at t are longer than the column-vectors at $t = 0$, while $\beta(t) < 1$ corresponds to shorter columns.

By differentiation we find:

$$\dot{\mathbf{M}}(t) = \dot{\beta}(t)\mathbf{R}(t)\mathbf{M}_0 + \beta(t)\dot{\mathbf{R}}(t)\mathbf{M}_0\,. \tag{11}$$

We have seen that $\dot{\mathbf{R}}(t)$ may be conveniently written as $\mathbf{P}(t)\mathbf{R}(t)$, with $\mathbf{P}(t) \in \mathbf{so}(p,\,\mathcal{R})$. By plugging Eq. (10) and the above Lie-group relationship into Eq. (11) we obtain:

$$\dot{\mathbf{M}}(t) = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{M}(t) + \mathbf{P}(t)\mathbf{M}(t)\,.$$

The obtained expression is equivalent to Eq. (8), provided that we define $\dot{\alpha}(t) \stackrel{\mathrm{def}}{=} -\dot{\beta}(t)/\beta(t)$ and $\dot{\mathbf{\Theta}}(t) \stackrel{\mathrm{def}}{=} \mathbf{P}(t)$. Note that $\mathbf{M}^T\mathbf{P}\mathbf{M}$ is skew-symmetric.

Sufficiency. If the Jacobian of criterion U possesses the structure (8), then from Eq. (7) we have:

$$\dot{\mathbf{M}}^T\mathbf{M} = -\mathbf{M}^T\dot{\mathbf{\Theta}}^T\mathbf{M} + \dot{\alpha}\mathbf{M}^T\mathbf{M}\,.$$

By hypothesis we know that $\mathbf{M}^T\dot{\mathbf{\Theta}}\mathbf{M}$ is skew-symmetric, thus $\dot{\mathbf{M}}^T\mathbf{M} + \mathbf{M}^T\dot{\mathbf{M}} = 2\dot{\alpha}m^2\mathbf{I}_p$, that is, the learning rule in question belongs to $\mathcal{F}$.

Scalar velocity. Because of form (7), we have $\mathrm{tr}[\mathbf{M}^T\nabla U] = \mathrm{tr}[\mathbf{M}^T\dot{\mathbf{\Theta}}\mathbf{M}] - \mathrm{tr}[\dot{\alpha}\mathbf{M}^T\mathbf{M}]$. Now, $\mathbf{M}^T\dot{\mathbf{\Theta}}\mathbf{M}$ is skew-symmetric, thus $\mathrm{tr}[\mathbf{M}^T\dot{\mathbf{\Theta}}\mathbf{M}] = 0$, while $\mathrm{tr}[\mathbf{M}^T\mathbf{M}] = qm^2$, thus $\lambda_U = 2\dot{\alpha}m^2$. This fact and the second of the general relationships in Eq. (3) gives expression (9). □

Any algorithm of the form (7) belonging to $\mathcal{F}$ has a criterion U and a function m obeying the above equations. It is not difficult to recognize in the right-hand side of Eq. (8) a term producing a pure rotation

of the connection-matrix $\mathbf{M}$, and a term that implies a stretching of matrix $\mathbf{M}$'s columns. About the condition on matrix-stream $\dot{\mathbf{\Theta}}$, it is worth metioning that, for instance, a stream such that $\mathbf{M}^T\dot{\mathbf{\Theta}}\mathbf{M} = \mathbf{0}$ is valid, because the null matrix is skew-symmetric.

It is interesting to note that the very general formulation above also accounts for common modifications to the straight gradient direction, as the addition of a regularization term: Let us consider, as an example, the expression $\nabla U = \mathbf{D} - \delta\mathbf{M}$, where $\mathbf{D}$ is a gradient direction and $\delta \geq 0$ is the decay constant.

For a neural system, it is important to determine analytically the steady-states of the learning equation that the learning system is equipped with and to study the dynamical properties of the learning algorithm in the vicinity of these states. These results are available for the presented theory: About the equilibrium and stability of the discussed learning system, we can give the following general theorem.

Theorem 2

System (7) with gradient (8) is stationary at time $t = t_*$ if and only if $\dot{\mathbf{\Theta}}(t_\star)\mathbf{M}(t_\star) = \mathbf{0}$ and $\dot{\alpha}(t_\star) = 0$. Also, let us suppose function U to be continuous in $\mathcal{R}^{p\times q}$, and denote as $U_{\mathcal{H}}$ the restriction of U to $\mathcal{H}_m^{p\times q}$; let us suppose further that there exists a region $\mathcal{H}_\star \subset \mathcal{H}_m^{p\times q}$ where $U_{\mathcal{H}}$ is convex. If $\mathbf{M}(0) \in \mathcal{H}_\star$ then system (7) asymptotically converges to the extreme of $U_{\mathcal{H}}$ contained in $\mathcal{H}_\star$.

Proof

By definition, a stationary point is a configuration $\mathbf{M}_\star = \mathbf{M}(t_\star)$ where $\dot{\mathbf{M}}(t_\star) = \mathbf{0}$. In the present context $\dot{\mathbf{M}} = \dot{\alpha}\mathbf{M} - \dot{\mathbf{\Theta}}\mathbf{M}$, hence the stationarity condition is $\dot{\mathbf{\Theta}}\mathbf{M} = \dot{\alpha}\mathbf{M}$; by pre-multiplying both sides by $\mathbf{M}^T$ we have $\mathbf{M}^T\dot{\mathbf{\Theta}}\mathbf{M} = \dot{\alpha}m^2\mathbf{I}_q$; now, the left-hand side of this equation is known to be skew symmetric, while the right-hand side is clearly symmetric, therefore we conclude their common value must be $\mathbf{0}$. This leads to $\dot{\alpha} = 0$ at the equilibrium, which implies $\dot{\mathbf{\Theta}}\mathbf{M} = \mathbf{0}$.

Once the equilibrium configurations have been found, it is necessary to ensure they are stable.

By hypothesis, it is known that there exists a region $\mathcal{H}_\star \subset \mathcal{H}_m^{p\times q}$ where the restriction of U to $\mathcal{H}_m^{p\times q}$ is convex. Formally, it is worth introducing a curvilinear-coordinate ϕ that parameterizes the pathway $\{\mathbf{M}\}$ over the manifold during network's learning phase, so that any learning pathway may be described by an appropriate application $\mathbf{M} = \mathbf{M}(\phi)$ with $\phi \in [\phi_1,\, \phi_2] \subset \mathcal{R}$. The restricted learning criterion is now simply $U_{\mathcal{H}} = U(\phi)$. The curvilinear coordinate is a function of the time, $\phi = \phi(t)$ and parameterization consistency requires it to be a monotonic function, namely $\dot{\phi}(t) > 0$.

By the derivative chain-rule it can be written:

$$\frac{dU}{dt} = \frac{dU}{d\phi}\frac{d\phi}{dt} \text{ which implies}$$

$$\text{sign}\left(\frac{dU}{dt}\right) = \text{sign}\left(\frac{dU}{d\phi}\right).$$

The time-derivative of criterion U may be computed as follows:

$$\frac{dU}{dt} = \text{tr}\left[\left(\frac{\partial U}{\partial \mathbf{M}}\right)^T \frac{d\mathbf{M}}{dt}\right]$$

$$= -\text{tr}\left[\left(\frac{\partial U}{\partial \mathbf{M}}\right)^T \left(\frac{\partial U}{\partial \mathbf{M}}\right)\right] \leq 0\,.$$

The above considerations prove that $U(\phi)$ is a monotonically decreasing function of ϕ. Note that this rule allows ϕ to eventually come arbitrarily close to an extreme of function U without necessarily coinciding with it.[b]

Function analysis ensures that a convex function possesses a minimum inside the region of convexity, thus the learning system actually converges (asymptotically) to the minimum of function U in $\mathcal{H}_\star$. □

It is very important to remark that the theorem just stated does not require the learning criterion U to be bounded nor convex on its whole range $\mathcal{R}^{p\times q}$, as its optimization is carried out over $\mathcal{H}_m^{p\times q}$ only. Nevertheless, it ensures that if there exists a region where the restriction of U to $\mathcal{H}_m^{p\times q}$ is convex, the algorithm surely finds the extreme contained within. Also note that the dual case that $-U_{\mathcal{H}}$ is convex may be considered as well: In this case the theorem holds with the sign reversed in Eq. (7).

An example helps in clarifying the conditions of Theorem 2. Let us consider a network formed by a

[b] Continuous-time systems usually converge in infinite time.

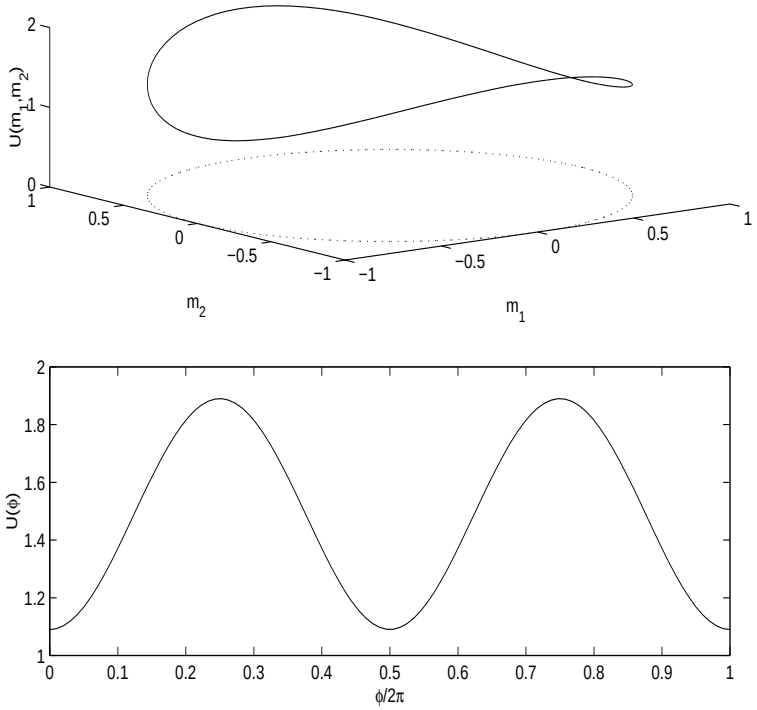

Fig. 4. Top: The dotted-line circle represents the manifold $\mathcal{H}_1^{2\times1}$, and the solid-line curve illustrates the shape of $U(m_1, m_2)$ with $[m_1\ m_2]^T \in \mathcal{H}_1^{2\times1}$. Bottom: The restriction $U(\phi)$ with the trigonometric parameterization.

single neuron with two connection-weights m_1, $m_2 \in \mathcal{R}$. In this case, the parameter space is $\mathcal{H}_m^{2\times1}$, that is a circle of radius $|m|$; for simplicity we assume $|m| =$ constant $= 1$. As a useful parameterization we take $m_1 = \cos\phi$ and $m_2 = \sin\phi$, with $\phi \in [0, 2\pi]$. Let us also suppose the criterion function has the structure $U(m_1, m_2) = 1 + 0.1m_1^2 + 0.9m_2^2 - 0.01(m_1^4 + m_2^4)$: This is not a bounded nor convex function in $\mathcal{R}^{2\times2}$, however, its restriction $U = U(\phi)$ is bounded and convex in a sub-interval of $[0, 2\pi]$ centered around π. A graphical representation of the manifold, of the restriction of U to the manifold and of $U(\phi)$ are given in Fig. 4.

2.3. *Second-order learning on Lie group*

As a special member of $\mathcal{F}$, let us consider the following coupled learning system:

$$\dot{\mathbf{M}} = \sigma\mathbf{B}\mathbf{M}\,, \quad \dot{\mathbf{B}} = \mathbf{S}[\mathbf{H}]\,, \tag{12}$$

with $\mathbf{M} \in \mathcal{R}^{p\times q}$, $\mathbf{B} \in \mathcal{R}^{p\times p}$, $\mathbf{B}^T(0) = -\mathbf{B}(0)$, $\mathbf{M}^T(0)\mathbf{M}(0) = m_0^2\mathbf{I}$, σ is a positive constant and $\mathbf{H}$ is an arbitrarily variable $p \times p$ matrix depending on criterion U, which controls layer's learning by means of the skew-symmetric operator $\mathbf{S}[\cdot]$.

Because of the structure of Eqs. (12), it is straightforward to see that for all $t \geq 0$ the property $\mathbf{B}^T = -\mathbf{B}$ holds true. In fact, from:

$$\dot{\mathbf{B}} = \mathbf{S}[\mathbf{H}]\,,\ \dot{\mathbf{B}}^T = -\mathbf{S}[\mathbf{H}]\,,\ \mathbf{B}(0) + \mathbf{B}^T(0) = \mathbf{0}\,,$$

it follows that $\mathbf{B}(t) + \mathbf{B}^T(t) = 0$. Moreover, the following result holds:

Theorem 3

Let us consider the dynamical system $d\mathbf{M}/dt = \sigma\mathbf{B}\mathbf{M}$ where $\mathbf{B}$ is skew-symmetric and σ is a real-valued scalar; if $\mathbf{M}(0) \in \mathcal{H}_{m_0}^{p\times q}$ for some m_0 real constant, then $\mathbf{M}(t)$ is a LLG. In this case the variable $m(t)$ keeps constant to $m(0) = m_0$, thus $\lambda_U(t) = 0$.

Proof

The considered dynamical system is described by a Lie-group equation, therefore it is known that if $\sigma\mathbf{B} \in \mathbf{so}(p, \mathcal{R})$ then $\mathbf{M}$ remains orthonormal. In particular, in this case $\dot{\mathbf{M}}^T\mathbf{M} + \mathbf{M}^T\dot{\mathbf{M}} = \mathbf{0}$, thus

velocity $\dot{\mathbf{M}} = 0$ and, as a remarkable result, $M(t)/m_0 \in \mathrm{St}(p, q, \mathcal{R})$. □

The learning rule (12) describes a second-order LLG. In fact, upon time-differentiation the above coupled first-order system may be rewritten as an implicit second-order non-linear differential learning rule $\mathcal{L}_U$:

$$\mathbf{M}^T\ddot{\mathbf{M}} - \ddot{\mathbf{M}}^T\mathbf{M} = 2\sigma\mathbf{M}^T\mathbf{S}[\mathbf{H}]\mathbf{M}\,, \qquad (13)$$

where $\ddot{\mathbf{M}} = d^2\mathbf{M}/dt^2$.

About system (12), a general result on its equilibrium can be given.

Theorem 4

System (12) is stationary at time $t = t^\star$ if $\mathbf{B}(t_\star)\mathbf{M}(t_\star) = \mathbf{0}$ and $\mathbf{H}(t^\star)$ symmetrizes.

Proof

In order for a configuration $\mathbf{M}(t)$ to be an equilibrium point at time $t = t_\star$ for system (12) it is necessary and sufficient that $\dot{\mathbf{B}} = \mathbf{0}$ and $\dot{\mathbf{M}} = \mathbf{0}$ at the same time.

The first condition implies $\mathbf{S}[\mathbf{H}] = \mathbf{0}$. By definition this happens when the matrix-argument is symmetric. The second condition implies $\mathbf{BM} = \mathbf{0}$. It is worth noting that this constraint does not necessarily imply matrix $\mathbf{B}$ to be zero. To show this fact, suppose a solution to the equilibrium problem $\mathbf{M}_\star$ is known, then we wonder how many solutions does the linear equation $\mathbf{BM}_\star = \mathbf{0}$ possess. By considering that $\mathbf{B}$ is skew symmetric, it is immediately seen that the above matrix-equation consists of $E = pq$ independent equations in $N = p(p-1)/2$ independent unknowns. In order for the system to possess more than one solution it must hold that $E < N$, that is:

$$2q + 1 < p\,.$$

In conclusion, as long as $2q + 1 \geq p$ the equilibrium happens for $\mathbf{B} = \mathbf{0}$ only. It is interesting to note that $q \geq 1$ by definition, thus a network must have a number of inputs $p \geq 4$ in order to admit dynamical equilibria with $\mathbf{B} \neq \mathbf{0}$. □

This general result allows the determination of the equilibrium points of the second-order LLG family of algorithms, whose stability should be studied case-by-case upon operator $\mathbf{S}[\cdot]$ specification.

3. Contributions to First-Order Learning on Lie Groups

The aim of this section is to present some examples of first order LLG theories drawn from the scientific literature.

3.1. *Xu's principal subspace rule*

Let us consider the following learning theory proposed by L. Xu in Ref. 56:

$$\beta\dot{\mathbf{M}} = \beta\nabla U_{\mathrm{X}} = \mathbf{C}_x\mathbf{M}(\mathbf{M}^T\mathbf{C}_x\mathbf{M})^{-1} - \gamma\mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}, \qquad (14)$$

where $\gamma(\mathbf{M})$ and $\beta(\mathbf{M})$ are properly defined matrix-to-scalar functions arising when evaluating the gradient of an objective function used by Xu for finding the principal subspaces of a multivariate random process $\mathbf{x} \in \mathcal{R}^p$ endowed with the covariance matrix $\mathbf{C}_x$; such an objective function has the structure $U_{\mathrm{X}}(\mathbf{M}) = f(\det(\mathbf{M}^T\mathbf{C}_x\mathbf{M}))/g(\det(\mathbf{M}^T\mathbf{M}))$, where $f(\cdot)$ and $g(\cdot)$ fulfill some minimal regularity requirements.[56]

Provided that $\mathbf{M}(0) \in \mathcal{H}_{m_0}^{p\times q}$, the above system easily proves to generate a LLG, in the sense that $\mathbf{M}(t) \in \mathcal{H}_m^{p\times q}$ where m changes through time. Therefore the gradient in Eq. (14) may be post-multiplied by $\mathbf{M}^T\mathbf{M}$ (which, by definition, is non-singular) obtaining the new system:

$$\dot{\mathbf{M}} = \frac{1}{\beta}[\mathbf{C}_x\mathbf{M}(\mathbf{M}^T\mathbf{C}_x\mathbf{M})^{-1}\mathbf{M}^T - \gamma\mathbf{I}_q]\mathbf{M}\,. \qquad (15)$$

It can be shown that the equation above generates a Lie-group dynamics of the matrix $\mathbf{M}$. It has in fact the structure (8), where:

$$\dot{\boldsymbol{\Theta}} = \frac{1}{\beta}\mathbf{C}_x\mathbf{M}(\mathbf{M}^T\mathbf{C}_x\mathbf{M})^{-1}\mathbf{M}^T - \frac{1}{\beta}\mathbf{I}_p,\ \dot{\alpha} = \frac{-\gamma+1}{\beta}\,.$$

3.2. *Oja's subspace rule*

Oja's subspace rule is a well-known learning algorithm that allows for the extraction of a basis of the subspace spanned by the eigenvectors of a signal's covariance matrix.[44] It arises from the gradient-based

optimization of the criterion:

$$U_{\rm O}(\mathbf{M}) = \frac{1}{2}{\rm tr}[\mathbf{M}^T\mathbf{C}_x\mathbf{M}] + \frac{1}{2}{\rm tr}[(\mathbf{M}^T\mathbf{M} - m_0^2\mathbf{I}_q)\mathbf{L}]\,, \tag{16}$$

where $\mathbf{L}$ is a symmetric matrix[c] containing the Lagrange multipliers for the constraint $\mathbf{M}^T\mathbf{M} = m_0^2\mathbf{I}_q$. The ordinary gradient of Oja's criterion writes:

$$\frac{\partial U_{\rm O}}{\partial \mathbf{M}} = \mathbf{C}_x\mathbf{M} + \mathbf{M}\mathbf{L}\,;$$

the optimal multiplier $\mathbf{L}^{\rm opt}$ may be computed by solving equation $\mathbf{M}^T\nabla U_{\rm O} = \mathbf{0}$, admitting the solution $\mathbf{L}^{\rm opt} = -(1/m_0^2)\mathbf{M}^T\mathbf{C}_x\mathbf{M}$, which leads to the first-order LLG rule:

$$\dot{\mathbf{M}} = \frac{1}{m_0^2}(m_0^2\mathbf{I}_q - \mathbf{M}\mathbf{M}^T)\mathbf{C}_x\mathbf{M}\,. \tag{17}$$

It is straightforward to demonstrate that this truly represents a LLG algorithm. What we aim to show is that it is of the form (8). To this end it is sufficient to note that $m_0^2\mathbf{C}_x\mathbf{M} - \mathbf{M}\mathbf{M}^T\mathbf{C}_x\mathbf{M}$ is equivalent to $\mathbf{C}_x\mathbf{M}\mathbf{M}^T\mathbf{M} - \mathbf{M}\mathbf{M}^T\mathbf{C}_x\mathbf{M}$; thus, Eq. (17) is equivalent to the Lie-group equation:

$$\dot{\mathbf{M}} = \frac{1}{m_0^2}[\mathbf{C}_x,\, \mathbf{M}\mathbf{M}^T]\mathbf{M}\,,$$

which is expressed in terms of the Lie bracket and coincides to the general equation (8), provided that:

$$\dot{\mathbf{\Theta}} = \mathbf{C}_x\mathbf{M}\mathbf{M}^T - \mathbf{M}\mathbf{M}^T\mathbf{C}_x\,, \quad \dot{\alpha} = 0\,.$$

3.3. *Chen–Amari–Lin contribution to principal/minor subspace analysis*

In Ref. 12, a theory for principal/minor subspace analysis (PSA/MSA) was developed on the basis of the Riemannian gradient. We may summarize here the work of Chen *et al.* as follows: First, it is recognized that the parameter space of PSA and MSA is a Stiefel manifold; then, this Riemannian manifold is endowed with the Killing metric, defined by a (non-holonomic) basis of the manifold's tangent space; the metric allows defining a Riemannian gradient on the manifold, thus a gradient-based optimization technique on it; finally, it is shown that the Riemannian-gradient-based maximization/minimization of the criterion $U_{\rm CAL}(\mathbf{M}) \stackrel{\rm def}{=} {\rm tr}[\mathbf{M}\mathbf{M}^T\mathbf{C}_x]$ leads to a pair of dual first-order algorithms of the form (8):

$$\dot{\mathbf{M}} = \pm[\mathbf{C}_x,\, \mathbf{M}\mathbf{M}^T]\mathbf{M}\,, \tag{18}$$

which generate a LLG whose stationary points are the matrices formed by the principal (sign +) or minor (sign −) eigenvectors of the covariance matrix $\mathbf{C}_x$.

It is most interesting to see that $U_{\rm CAL}(\mathbf{M})$ coincides to the unconstrained version of $U_{\rm O}(\mathbf{M})$, and that Oja's rule may be generalized from PSA to MSA by simply reversing the sign. Also, as it holds that:

$$2U_{\rm O}(\mathbf{M}) - 2U_{\rm CAL}(\mathbf{M}) = {\rm tr}[(\mathbf{M}^T\mathbf{M} - m_0^2\mathbf{I}_q)\mathbf{L}]\,,$$

we have as an interesting result that the Lagrangian term on the right-hand side of above equation allows converting of the ordinary gradient into a Riemannian gradient.

3.4. *A case-study on principal component analysis*

In order to make clearer the mentioned concepts, let us discuss the case-study where an adaptive linear network, described by $\mathbf{M} \in \mathcal{R}^{2\times 2}$, is employed to extract the two principal components from a zero-mean random signal with covariance matrix:

$$\mathbf{C}_x = \begin{bmatrix} A & B \\ B & C \end{bmatrix}. \tag{19}$$

To this aim, we may define a criterion $U_{\rm OW}$ as a weighted Oja's criterion, as $U_{\rm OW}(\mathbf{M}) \stackrel{\rm def}{=} {\rm tr}[\mathbf{M}^T\mathbf{C}_x\mathbf{M}\mathbf{W}]$, where $\mathbf{W}$ is a weighting kernel of the form $\mathbf{W} \stackrel{\rm def}{=} {\rm diag}(w_{11},\, w_{22})$, with $w_{11} \neq w_{22}$, which breaks the symmetry of $U_{\rm O}(\mathbf{M})$ and makes it a PCA criterion. The above cost function is clearly a quadratic form: It is unbounded but can be optimized under the constraint of orthonormality.

The natural parameterization for $\mathbf{M}$ in $\mathcal{H}_1^{2\times 2}$ is:

$$\mathbf{M}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}, \tag{20}$$

with $\phi \in [-\pi,\, \pi]$ radians.

The restriction of $U_{\rm OW}$ to $\mathcal{H}_1^{2\times 2}$ writes:

$$\begin{aligned}(U_{\rm OW})_{\mathcal{H}}(\phi) = {}& (w_{11}A + w_{22}C)\cos^2\phi \\ & - 2B(w_{22} - w_{11})\sin\phi\,\cos\phi \\ & + (w_{11}C + w_{22}A)\sin^2\phi\,. \end{aligned} \tag{21}$$

[c]More detailed notes on the matrix $\mathbf{L}$ are given in the Proof of Theorem 5.

Finding the values of the parameter ϕ that optimizes the criterion implies computing the zeros of its first derivative corresponding to negative values of the second derivative; these derivatives read:

$$\begin{aligned}\frac{d(U_{\mathrm{OW}})_{\mathcal{H}}(\phi)}{d\phi} &= -(w_{22}-w_{11})[2B\,\cos(2\phi) \\ &\quad -(A-C)\sin(2\phi)]\,, \\ \frac{d^2(U_{\mathrm{OW}})_{\mathcal{H}}(\phi)}{d\phi^2} &= 2(w_{22}-w_{11})[2B\,\sin(2\phi) \\ &\quad +(A-C)\cos(2\phi)]\,.\end{aligned} \tag{22}$$

It deserves to note that the values of w_{11} and w_{22} do not change the solution, while changing the sign of $w_{11}-w_{22}$ results in a rotation of the configuration of $\mathbf{M}(\phi)$ of $\pi/2$ radians. In order to give a numerical example, let us consider the case that $w_{11} > w_{22}$ and that the covariance values A, B and C are in the relationship $2B = (A-C)\sqrt{3}$. In this way, the optimal value of the network's parameter proves to be $\phi^\star = \pi/6$ rad. The above functions are depicted in the Fig. 5, from which it emerges that there exists at least a sub-interval of $[-\pi, +\pi]$ where $-(U_{\mathrm{OW}})_{\mathcal{H}}$ is convex, thus, according to Theorem 2, we may expect that a gradient-based algorithm will be able to find the extreme contained within.

In order to extract, say, the first principal component, algorithm (18) may be employed with + sign. The results are illustrated in the Fig. 6, where it may be seen that the algorithm correctly finds the optimum $\phi^\star \approx 0.5236$ rad, according to Theorem 2. It is worth noting that by reversing the sign, the minor component could be found as well.

A formal analysis of the behavior of the above learning theory may be carried out. In particular, in this special case we are enabled to write the explicit solution $\phi = \phi(t)$ to the learning equation in the curvilinear-coordinate system.

First, it is necessary to express the time-derivative of the curvilinear coordinate as a function of the restricted criterion $U_{\mathcal{H}}$. This result may be achieved in the following way: From the general first-order learning equation (12) we have $d\mathbf{M}/dt = -(\partial U/\partial \mathbf{M})$; by transposing, post-multiplying by $d\mathbf{M}/d\phi$ and computing the trace of both members we get:

$$\mathrm{tr}\left[\left(\frac{d\mathbf{M}}{dt}\right)^T \frac{d\mathbf{M}}{d\phi}\right] = -\mathrm{tr}\left[\left(\frac{\partial U}{\partial \mathbf{M}}\right)^T \frac{d\mathbf{M}}{d\phi}\right] = -\frac{dU}{d\phi}\,.$$

By substituting the derivative $d\mathbf{M}/dt$ with $(d\mathbf{M}/d\phi)(d\phi/dt)$ we finally get:

$$\frac{d\phi}{dt} = -\mathrm{tr}^{-1}\left[\left(\frac{d\mathbf{M}}{dt}\right)^T \frac{d\mathbf{M}}{d\phi}\right] \frac{dU}{d\phi}\,. \tag{23}$$

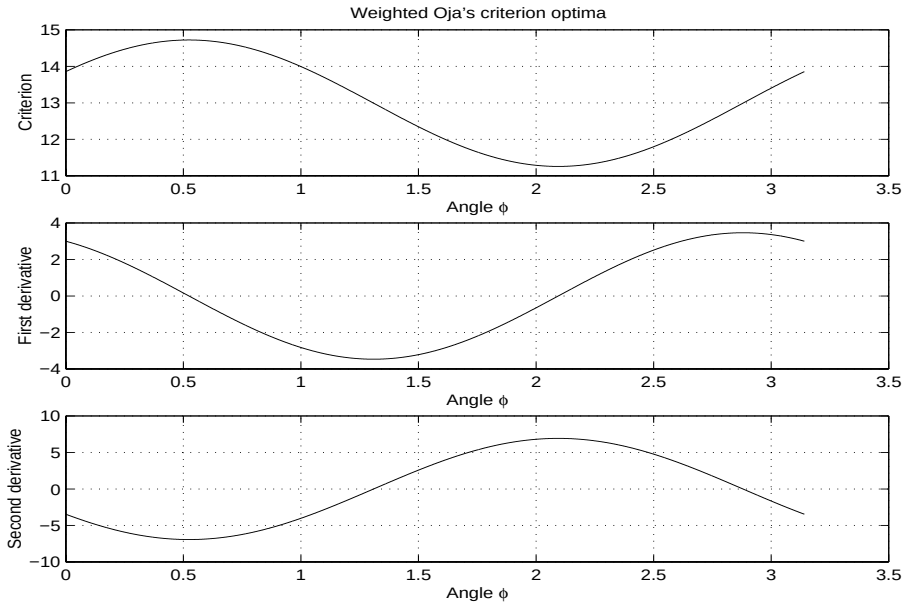

Fig. 5. Weighted Oja's criterion along with its first and second derivative when $2B = (A-C)\sqrt{3}$.

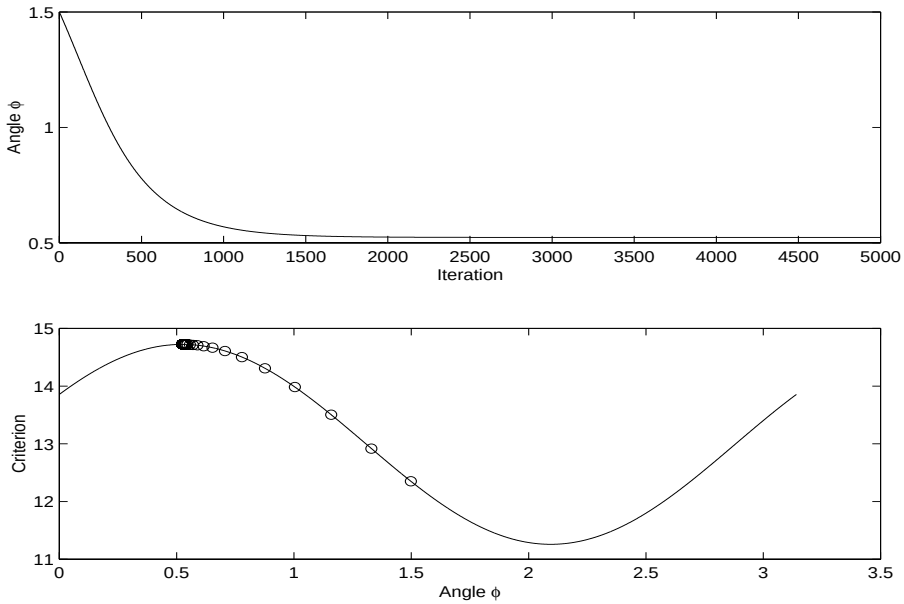

Fig. 6. First principal component extraction. Top: dynamics of the principal angle exhibited by algorithm (18); bottom: dynamics on the criterion (from right to left).

In the present case, from definition (20) we have, for the derivative of network connection matrix with respect to the curvilinear coordinate:

$$\frac{d\mathbf{M}(\phi)}{d\phi} = \begin{bmatrix} -\sin\phi & -\cos\phi \\ \cos\phi & -\sin\phi \end{bmatrix},$$

$$\text{which implies tr}\left[\left(\frac{d\mathbf{M}}{d\phi}\right)^T \frac{d\mathbf{M}}{d\phi}\right] = 2\,.$$

The derivative of the restricted-criterion computed with respect to the curvilinear coordinate has already been computed, and may be conveniently rewritten from Eq. (22), under the mentioned hypothesis that $2B = (A-C)\sqrt{3}$, as:

$$\frac{d(U_{\text{OW}})_{\mathcal{H}}}{d\phi} = 2\gamma_{\text{OW}}\sin(2\phi - \pi/3)\,,$$

$$\gamma_{\text{OW}} \stackrel{\text{def}}{=} \frac{(w_{22}-w_{11})(A-C)}{2\,\cos(\pi/3)}\,.$$

By substitution of the found results into Eq. (23) we find the dynamics description in ϕ:

$$\frac{d\phi}{dt} = -\gamma_{\text{OW}}\sin(2\phi - \pi/3)\,. \tag{24}$$

The above differential equation has separable variables, thus by the help of the general integral:

$$\int_{x_0}^{x_1}\frac{dx}{\sin x} = \log\left[\frac{1-\cos x_1}{\sin x_1}\frac{\sin x_0}{1-\cos x_0}\right],$$

$$x_0,\, x_1 \in]0,\, \pi[\,,$$

it is possible to write the solution to the network learning equation, with initial state ϕ_0, as:

$$\frac{1-\cos(2\phi-\pi/3)}{\sin(2\phi-\pi/3)} = \frac{1-\cos(2\phi_0-\pi/3)}{\sin(2\phi_0-\pi/3)}e^{-2\gamma_{\text{OW}}t}\,. \tag{25}$$

The discussion of the possible asymptotic behavior of the learning equations can be made over two admissible cases:

- $\gamma_{\text{OW}} > 0$: In this case, as $t \to +\infty$ the right-hand side of Eq. (25) tends to vanish, therefore $1-\cos(2\phi-\pi/3) \to 0$ and $\phi \to \phi^\star$;
- $\gamma_{\text{OW}} < 0$: In this case, as $t \to +\infty$ the right-hand side of Eq. (25) tends to infinity, therefore $\sin(2\phi-\pi/3) \to 0$ and again $\phi \to \phi^\star$.

This analysis confirms that the system always converges to the right solution, and shows that the speed that the system travels with on the manifold $\mathcal{H}$ depends on the eigenvalue spread and on the separation $|w_{11}-w_{22}|$.

3.5. *A case-study on kurtosis-based independent component analysis*

Another interesting example arises from the theory of independent component analysis (ICA) by kurtosis optimization. Let us consider the 2×2 ICA problem described by the noiseless mixing model and neural de-mixing model:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{A}^T \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \quad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{M}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad (26)$$

where $s_1(t)$ and $s_2(t)$ represent two zero-mean, non-jointly-Gaussian statistically independent source signals with covariance matrix $\mathbf{C}_s = \mathbf{I}_2$; $\mathbf{A}$ is an orthonormal 2×2 matrix such that the observable signals $x_1(t)$ and $x_2(t)$ have a covariance matrix $\mathbf{C}_x = \mathbf{I}_2$. The aim of ICA is to recover s_1 and s_2 from observations of x_1 and x_2 only (either s_1, s_2, and $\mathbf{M}$ are unknown).

As a cost function to be optimized under the constraint of orthonormality, it may be assumed the weighted sum of fourth-order moments of network's output signals[27]:

$$U_{\mathrm{K}}(\mathbf{M}) \stackrel{\mathrm{def}}{=} w_{11}\mathbb{E}_{\mathbf{x}}[y_1^4] + w_{22}\mathbb{E}_{\mathbf{x}}[y_2^4]\,; \qquad (27)$$

again w_{11} and w_{22} are two different weighting numbers.

Again the natural parameterization for $\mathbf{A}$ and $\mathbf{M}$ is (ϕ, $\phi_A \in [-\pi, \pi]$ rad):

$$\mathbf{A} = \begin{bmatrix} \cos\phi_A & -\sin\phi_A \\ \sin\phi_A & \cos\phi_A \end{bmatrix}, \ \mathbf{M} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}. \qquad (28)$$

Straightforward calculations give, for the restriction of the criterion to $\mathcal{H}_1^{2\times 2}$:

$$\begin{aligned} &(U_{\mathrm{K}})_{\mathcal{H}}(\phi) \\ &= (w_{11}\mathbb{E}_{\mathbf{s}}[s_1^4] + w_{22}\mathbb{E}_{\mathbf{s}}[s_2^4])\cos^4(\phi_A - \phi) \\ &\quad + (w_{11}\mathbb{E}_{\mathbf{s}}[s_2^4] + w_{22}\mathbb{E}_{\mathbf{s}}[s_1^4])\sin^4(\phi_A - \phi) \\ &\quad + 6(w_{11} + w_{22})\cos^2(\phi_A - \phi)\sin^2(\phi_A - \phi)\,. \end{aligned} \qquad (29)$$

The above function is depicted in the Fig. 7 along with its first derivative and second derivative for $\mathbb{E}_{\mathbf{s}}[s_1^4] = 4$, $\mathbb{E}_{\mathbf{s}}[s_1^4] = 5$, $w_{11} = -0.1$, $w_{22} = -1$, $\phi_A = 1$. A local and a global minimum, corresponding to two *equivalent* solutions, clearly appears, thus we may invoke Theorem 2 to ensure convergence to the expected solution.

4. Contributions to Second-Order Learning on Lie Group

The aim of the present section is to present contributions to second-order Lie-group learning theories,

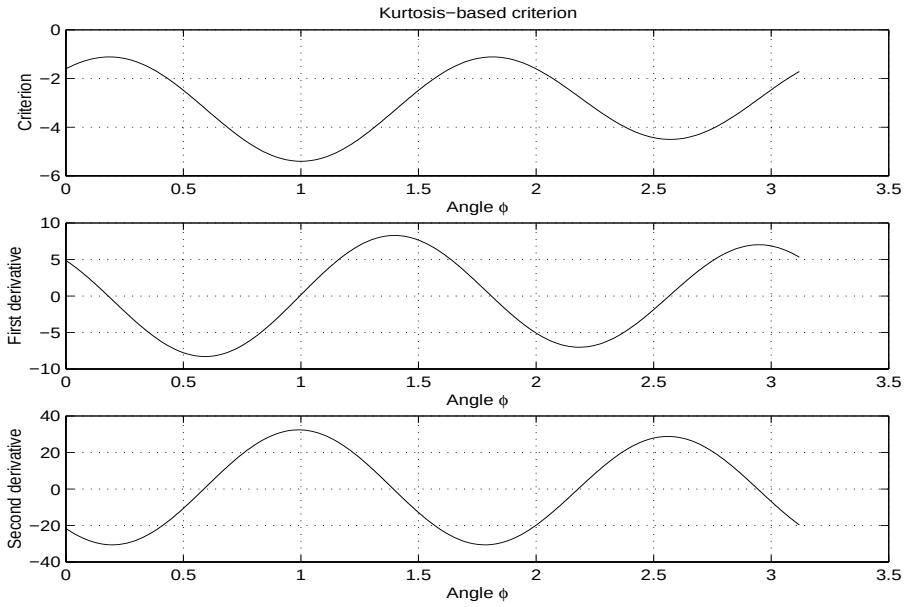

Fig. 7. Kurtosis-based criterion and its first and second derivative.

in terms of a class of learning rules (termed "mechanical" because of the close relationship with rigid-bodies dynamics phenomena) and of proven relationships with existing contributions drawn from the scientific literature.

4.1. *"Mechanical" learning rule*

It is possible to interpret Eq. (12) as equations describing the dynamics of a rigid body moving in an abstract space under a potential energy field, provided that:

$$\mathbf{S}[\mathbf{H}] \stackrel{\text{def}}{=} \mathbf{H} - \mathbf{H}^T,\ \mathbf{H} \stackrel{\text{def}}{=} -\left(\kappa \frac{\partial U}{\partial \mathbf{M}} + \mu \mathbf{B}\mathbf{M}\right)\mathbf{M}^T, \tag{30}$$

where U is a function bounded below to be minimized, under the restriction $\mathbf{M} \in \mathcal{H}_{m_0}^{p\times q}$ and $\mu \geq 0$, $\kappa \geq 0$.[23] A schematic representation of the mechanical system under analysis is illustrated in Fig. 8 for $p = q = 3$.

Prior to illustrating the features of the "mechanical" learning theory, it seems interesting to discuss the role of the learning function U introduced in the equations above. In a rational-mechanics context it plays the role of a *potential energy function* which compactly describes the effects of the external stimuli on the mechanical system. From a neural-network perspective, it is associated with the criterion which drives the network's learning and measures its ability to perform a pre-defined task.

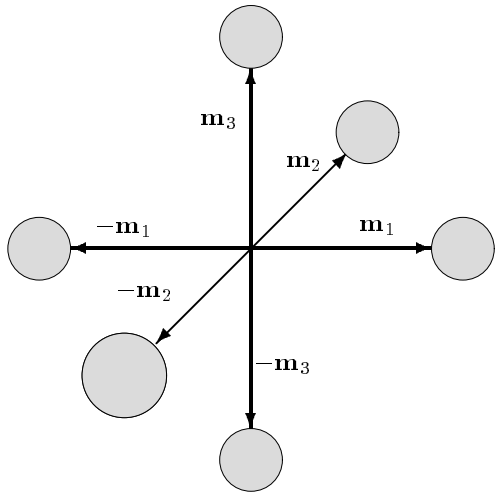

Fig. 8. Exemplary representation of the mechanical system for $p = q = 3$.

Very remarkably, recently the theory of "natural" potentials (such as gravitational and electrostatic ones) has been recognized as a fruitful way to design new learning algorithms for adaptive systems (or fruitfully re-interpret existing algorithms). We wish to cite here two of such contributions.

First, the theory of Coulomb classifiers,[30] where a family of classifiers is introduced based on the physical analogy to an electrostatic system of charged conductors; the class includes the two best-known support-vector machines, the ν–**SVM** and the C–**SVM**; in the electrostatics analogy, a training example corresponds to a charged conductor at a given location in space, the classification function corresponds to the electrostatic potential function, and the training objective function corresponds to the Coulomb energy. Such an electrostatic framework not only provides a novel interpretation of existing algorithms and their interrelationships, but it suggests a variety of new methods for support vector machines including kernels that bridge the gap between polynomial and radial-basis functions, objective functions that do not require positive-definite kernels, regularization techniques that are not cast in terms of violation of margin constraints, and speed-up techniques using either approximated or restricted algorithms.

Second, we wish to cite the theory of force field energy functionals for image feature extraction:[31] In the context of ear biometrics, a novel force field transformation was developed in which the image is treated as an array of Gaussian attractors that behave as the source of a force field. The directional properties of the force field are exploited to automatically locate the extremes of a small number of potential energy wells and associated potential channels, which form the basis of the ear description for automatic ear recognition.

In order to present the formal features of the introduced second-order LLG theory, it is useful to consider system (13) as being represented by the extended state-matrix $\mathbf{X} = (\mathbf{B}, \mathbf{M})$. The following theorem studies the existence of special equilibrium points for the second-order system (12) when the definitions (30) hold.

Theorem 5

Let us consider the dynamical system (12) where matrix $\mathbf{H}$ is assumed as in Eqs. (30), the initial state

is chosen so that $\mathbf{M}(0) \in \mathcal{H}_{m_0}^{p\times q}$ and $\mathbf{B}(0)$ is skew-symmetric. Let us also define the matrix function $\mathbf{F} \stackrel{\text{def}}{=} -\kappa(\partial U/\partial \mathbf{M})$, and denote as $\mathbf{F}_\star$ the value of $\mathbf{F}$ at $\mathbf{M}_\star$. A state $\mathbf{X}_\star = (\mathbf{B}_\star, \mathbf{M}_\star)$ is stationary for the system if $\mathbf{F}_\star^T\mathbf{M}_\star$ is symmetric and $\mathbf{B}_\star\mathbf{M}_\star = \mathbf{0}$. These stationary points are among the extremes of learning criterion U over $\mathcal{H}_{m_0}^{p\times q}$.

Proof

With the definition given, the learning equations write as:

$$\dot{\mathbf{M}} = \mathbf{B}\mathbf{M}\,, \quad \dot{\mathbf{B}} = \mathbf{H} - \mathbf{H}^T\,,$$
$$\mathbf{H} = \mathbf{F}\mathbf{M}^T - \mu(\mathbf{B}\mathbf{M})\mathbf{M}^T\,.$$

From Theorem 4 we know that this system sticks when $\mathbf{B}\mathbf{M} = \mathbf{0}$ and $\mathbf{H}$ is symmetric, which is equivalent to $\mathbf{B}\mathbf{M} = \mathbf{0}$ and $\mathbf{F}\mathbf{M}^T = \mathbf{M}\mathbf{F}^T$. These two equations give rise to a system of non-linear coupled scalar equations (the force-matrix $\mathbf{F}$ is a function of the connection matrix) and cannot be further solved. However, it is worth noting that the second equilibrium equation consists of at most $p(p+1)/2$ scalar identities; its solutions are also solutions of $\mathbf{M}^T(\mathbf{F}\mathbf{M}^T)\mathbf{M} = \mathbf{M}^T(\mathbf{M}\mathbf{F}^T)\mathbf{M}$, that is of $\mathbf{M}^T\mathbf{F} = \mathbf{F}^T\mathbf{M}$; this matrix-equation consists of at most $q(q+1)/2$ independent identities, where $q(q+1)/2 \leq p(p+1)/2$. As a consequence, the last system of constraints is smaller and easier to solve (though it has less equations and may have a number of non-equilibrium solutions).

In order to prove that the set of solutions of the above system contains the extremes of the learning criterion U over the manifold $\mathcal{H}_{m_0}^{p\times q}$, let us characterize the extremal points of $U(\mathbf{M})$ on the manifold $\mathcal{H}_{m_0}^{p\times q}$. This operation may be conveniently performed by the help of Lagrangian function $L(\mathbf{M})$ defined as:

$$L(\mathbf{M}) \stackrel{\text{def}}{=} \kappa U(\mathbf{M}) + \text{tr}((\mathbf{M}^T\mathbf{M} - m_0^2\mathbf{I}_q)\mathbf{L}),\ \mathbf{L}^T = \mathbf{L}\,,$$

where matrix $\mathbf{L}$ contains the Lagrange multipliers ℓ_{ij} that take into account the fact that we are interested in the extremal points of the learning criterion on the Stiefel manifold. In particular, the diagonal entries of the Lagrange matrix weight the deviation of the connection matrix from normality, while the off-diagonal multipliers measure the deviation of network connection matrix from orthogonality; as there is no way to discriminate, in the expression $\text{tr}((\mathbf{M}^T\mathbf{M} - m_0^2\mathbf{I}_q)\mathbf{L})$, between the constraint on $\mathbf{m}_i^T\mathbf{m}_j$ and $\mathbf{m}_j^T\mathbf{m}_i$, the multipliers ℓ_{ij} and ℓ_{ji} are equal, thus $\mathbf{L}$ is symmetric. Now, the extremal points of $U(\mathbf{M})$ on the manifold compute as the free extremal points of $L(\mathbf{M})$ in $\mathcal{R}^{p\times q}$, which are found from the equation:

$$\frac{\partial L}{\partial \mathbf{M}} = \kappa\frac{\partial U}{\partial \mathbf{M}} + 2\mathbf{M}\mathbf{L} = -\mathbf{F} + 2\mathbf{M}\mathbf{L} = 0\,.$$

By pre-multiplying the last equation by $\mathbf{M}^T$, the characterization $\mathbf{M}^T\mathbf{F} = 2\mathbf{L}$ arises. This proves that the product $\mathbf{F}^T\mathbf{M}$ is symmetric at the optimum. □

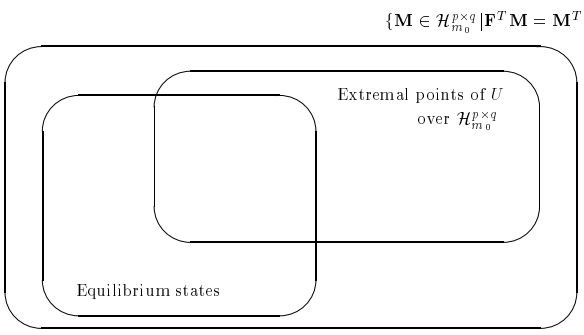

Fig. 9. Relationship between solutions of equilibrium equation, actual equilibrium states and extremal points of learning criterion function for the "mechanical" learning paradigm.

The above theorem shows that the set of states $\{\mathbf{M} \in \mathcal{H}_{m_0}^{p\times q} | \mathbf{F}^T\mathbf{M} = \mathbf{M}^T\mathbf{F}\}$ contains the equilibrium states for the mechanical system and the extremal points of the learning criterion on the manifold. This relationship is illustrated in the Fig. 9.

A fundamental feature of the system (12) + (30) is its asymptotic (Lyapunov) stability.

Theorem 6

Let U be a real-valued function of $\mathbf{M}$, $(1/m_0)\mathbf{M} \in \text{SO}(p, \mathcal{R})$, bounded from below with a minimum in $\mathbf{M}_\star$. Then the equilibrium state $\mathbf{X}_\star = (\mathbf{0}, \mathbf{M}_\star)$, is asymptotically (Lyapunov) stable for system (12) + (30) if $\mu > 0$, while simple stability holds if $\mu \geq 0$.

Proof

The learning equations under analysis may be summarized as:

$$\dot{\mathbf{M}} = \mathbf{B}\mathbf{M},\ \dot{\mathbf{B}} = (\mathbf{F}+\mathbf{P})\mathbf{M}^T - \mathbf{M}(\mathbf{F}+\mathbf{P})^T,\ \mathbf{P} = -\mu\dot{\mathbf{M}}\,.$$

By differentiating the first equation with respect to the time we get:

$$\ddot{\mathbf{M}}=\dot{\mathbf{B}}\mathbf{M}+\mathbf{B}\dot{\mathbf{M}}=m_0^2(\mathbf{F}+\mathbf{P})-\mathbf{M}(\mathbf{F}+\mathbf{P})^T\mathbf{M}+\mathbf{B}^2\mathbf{M}.$$

Let us now define function $K(t) \stackrel{\text{def}}{=} (1/2)\text{tr}(\dot{\mathbf{M}}^T\dot{\mathbf{M}})$. Its time-derivative reads $\dot{K}(t) = \text{tr}(\ddot{\mathbf{M}}^T\dot{\mathbf{M}})$. Then from the above identity, the product $\ddot{\mathbf{M}}^T\dot{\mathbf{M}}$ writes:

$$\begin{aligned}\ddot{\mathbf{M}}^T\dot{\mathbf{M}} &= m_0^2(\mathbf{F}+\mathbf{P})^T\dot{\mathbf{M}} \\ &\quad -\mathbf{M}^T(\mathbf{F}+\mathbf{P})\mathbf{M}^T\dot{\mathbf{M}}-\mathbf{M}^T\mathbf{B}^3\mathbf{M}\,, \\ &= m_0^2\mathbf{F}^T\dot{\mathbf{M}}-m_0^2\mu\dot{\mathbf{M}}^T\dot{\mathbf{M}}-\mathbf{M}^T\mathbf{F}\mathbf{M}^T\mathbf{B}\mathbf{M} \\ &\quad +m_0^2\mu\mathbf{M}\dot{\mathbf{M}}-\mathbf{M}^T\mathbf{B}^3\mathbf{M}\,.\end{aligned}$$

The last equalities have been obtained by the definition of $\mathbf{P}$ and by the knowledge that $\mathbf{M}\mathbf{M}^T = m_0^2\mathbf{I}_p$. It is now worth computing the trace of both sides of the last identity; to this aim, it is useful to remember that $\text{tr}(\mathbf{M}^T\mathbf{A}\mathbf{M}) = \text{tr}(\mathbf{A}\mathbf{M}\mathbf{M}^T) = m_0^2\text{tr}(\mathbf{A})$ for every $\mathbf{A} \in \mathcal{R}^{p\times p}$, and that $\mathbf{B}^3$ is a skew-symmetric matrix, which is traceless, afterwards we find:

$$\begin{aligned}\text{tr}(\ddot{\mathbf{M}}^T\dot{\mathbf{M}}) &= m_0^2\text{tr}(\mathbf{F}^T\dot{\mathbf{M}})-m_0^2\mu\text{tr}(\dot{\mathbf{M}}^T\dot{\mathbf{M}}) \\ &\quad -m_0^2\text{tr}(\mathbf{F}\mathbf{M}^T\mathbf{B})-m_0^2\mu\text{tr}(\dot{\mathbf{M}}^T\dot{\mathbf{M}}) \\ &\quad -\text{tr}(\mathbf{M}^T\mathbf{B}^3\mathbf{M})\,, \\ &= 2m_0^2\text{tr}(\mathbf{F}^T\dot{\mathbf{M}})-2m_0^2\mu\text{tr}(\dot{\mathbf{M}}^T\dot{\mathbf{M}})\,.\end{aligned}$$

We have already proven that $\dot{U}(t) = -\text{tr}(\mathbf{F}^T\dot{\mathbf{M}})$, therefore a relationship between functions K, U and $\dot{K}$ is:

$$\dot{K}(t) = -2m_0^2\dot{U}(t)-4\mu m_0^2K(t)\,.$$

Let us finally define the function:

$$T(t) \stackrel{\text{def}}{=} K(t)+2m_0^2[U(t)-U_\star]\,,$$

By construction it holds $T(t) \geq 0\,\forall\, t$, and because of the last relationship found, it also holds:

$$\dot{T}(t) = -4\mu m_0^2K(t) \leq 0\,.$$

This shows that for $\mu > 0$ there exists a Lyapunov function for the system, $T(t)$, that proves the network equilibrium state $\mathbf{M}_\star$, that the $U_\star$ corresponds to, is asymptotically stable. It is also worth noting that, from general Theorem 4, we know that in the present case the equilibrium holds only for $\mathbf{B} = \mathbf{0}$, this is the only point where the Lyapunov function vanishes. If $\mu = 0$ the function $T(t)$ keeps constant to $T(0)$ and does not represent a valid Lyapunov function for the neural network; however, the network state $\mathbf{M}$ keeps within a compact manifold, thus the neural network is by construction simply stable. □

Note that in general, function $U(\mathbf{M})$ may have more than one minimum (local minima) corresponding to local maxima of $-U(\mathbf{M})$. Also, the choice of $\mathbf{B}(0)$, together with $\mathbf{M}(0)$, affects the behavior of the learning system and may provide additional control of the solution of second-order learning equations.[4]

The above result holds for the case of a "complete" network (having p inputs and $q = p$ neurons). A similar result holds true for the simpler case of a reduced-size network, as stated in the following Theorem.

Theorem 7

Let U be a real-valued function of $\mathbf{M} \in \mathcal{H}_{m_0}^{p\times q}$, bounded from below with a minimum in $\mathbf{M}_\star$. Then the equilibrium state $\mathbf{M}_\star$, is asymptotically (Lyapunov) stable for system (12) + (30) if $\mu > 0$, while simple stability holds if $\mu \geq 0$.

Proof

The proof is identical to the proof of Theorem 6, with the replacement of $\mathbf{M}$ with the extended matrix $[\mathbf{M}\ \mathbf{M}_c]$ such that $(1/m_0)[\mathbf{M}\ \mathbf{M}_c] \in \text{SO}(p, \mathcal{R})$, $\mathbf{F}$ with $[\mathbf{F}\ \mathbf{0}] \in \mathcal{R}^{p\times p}$ and $\mathbf{P}$ with $[\mathbf{P}\ \mathbf{0}] \in \mathcal{R}^{p\times p}$. It is worth noting that in this case, according to Theorem 4, the Lyapunov function does not necessarily vanishes in $\mathbf{B} = \mathbf{0}$ only. □

The proofs of the above theorems for the stability of network learning have been facilitated by the parallelism with rational-kinematics concepts: It is not difficult, for instance, to correlate the meaning of function $K(t)$ with the kinetic energy of mechanical systems, and the term $\mathbf{B}^2\mathbf{M}$ in the expression of acceleration $\ddot{\mathbf{M}}$ with the Coriolis force (which, in fact, has null associated power in the energetic balances).

4.2. *A case-study on variance extremization*

In order to gain qualitative knowledge on this behavior, we propose the following case-study. Let us

consider the studied second-order system with $p = 2$ and $m = 1$. In this case the network is described by $y = \mathbf{m}^T\mathbf{x}$, with the normalized weight-vector $\mathbf{m}/m_0$ belonging to $\mathrm{St}(2, 1, \mathcal{R})$ and the input vector $\mathbf{x}$ belonging to $\mathcal{R}^2$. The matrix $\mathbf{B}$ and connection vector may be thus parameterized as:

$$\mathbf{B} = \begin{bmatrix} 0 & b \\ -b & 0 \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} \sin\phi \\ \cos\phi \end{bmatrix}, \tag{31}$$

with $b \in \mathcal{R}$ and angle $\phi \in [0,\, 2\pi]$.

Let us suppose the input-stream $\mathbf{x}(t)$ possesses bounded covariance matrix $\mathbf{C}_x$ defined again as in (19) and zero mean.

We now wish to investigate the extraction of the first principal component from $\mathbf{x}$, that may be obtained by means of the potential energy function $U(\mathbf{m}) \stackrel{\mathrm{def}}{=} -(\kappa/2)\mathbb{E}_{\mathbf{x}}[y^2]$, with $\kappa > 0$ arbitrary. By definition, the matrix $\mathbf{F}$ in this case reduces to a 2×1 vector $\mathbf{f} = \kappa\mathbf{C}_x\mathbf{m}$. As a consequence, the learning equations for the parameters b and ϕ write as:

$$\begin{aligned} \dot{b} &= \frac{K}{2}(A - C)\sin(2\phi) + \kappa B\,\cos(2\phi) - \mu b\,, \\ \dot{\phi} &= b\,. \end{aligned} \tag{32}$$

Let

$$\begin{aligned} \sin(2\phi_P) &\stackrel{\mathrm{def}}{=} \frac{2B}{\sqrt{(A-C)^2 + 4B^2}}\,, \\ \cos(2\phi_P) &\stackrel{\mathrm{def}}{=} \frac{C - A}{\sqrt{(A-C)^2 + 4B^2}}\,, \\ \bar{\kappa} &\stackrel{\mathrm{def}}{=} \frac{\kappa}{2}\sqrt{(A-C)^2 + 4B^2}\,. \end{aligned} \tag{33}$$

With these definitions, the above system of first-order differential equations may be recast into the following second-order differential equation:

$$\begin{aligned} &\frac{d^2\phi}{dt^2} + \mu\frac{d\phi}{dt} = -\bar{\kappa}\,\sin(2(\phi - \phi_P))\,, \\ &\mu > 0\,, \quad \bar{\kappa} \in \mathcal{R}\,, \quad \phi(0) = \phi_0\,, \quad \dot{\phi}(0) = 0\,. \end{aligned} \tag{34}$$

The equilibrium points of this system are $\dot{\phi}_\star = 0$ and $\phi_\star = \phi_P + (n\pi/2)$ with $n \in \mathcal{Z}$. Note that the new constant $\bar{\kappa}$ inherits the signum of κ and is therefore always positive.

The functions $K(t)$ and $U(t)$ may be expressed in closed form. In particular, because of the chosen parameterization, we easily obtain $K(t) = (1/2)b^2(t)$. Also, for the potential energy function, we have:

$$U(t) = -\frac{\kappa}{2}[A\,\sin^2\phi(t) + C\,\cos^2\phi(t) + B\,\sin(2\phi(t))]\,.$$

By the use of trigonometric identities and the definitions (33), the above function recasts into:

$$U(t) = -\frac{\bar{\kappa}}{2}\left[\frac{A+C}{\sqrt{(A-C)^2+4B^2}} + \cos(2(\phi(t) - \phi_P))\right]\,.$$

It is worth remembering that the matrix with components A, B, C is a covariance tensor, thus it must hold $A \geq 0$ and $C \geq 0$; consequently, the first term in the parentheses in the above equation is non-negative, and the function $U(t)$ has the minimal value for $\cos(2(\phi(t) - \phi_P)) = 1$. The minimal value is:

$$U_\star = -\frac{\bar{\kappa}}{2}\left[\frac{A+C}{\sqrt{(A-C)^2+4B^2}} + 1\right]\,.$$

In the present case, the lifted potential energy function has therefore expression:

$$U(t) - U_\star = -\frac{\bar{\kappa}}{2}[\cos(2(\phi(t) - \phi_P)) - 1] \in [0,\, \bar{\kappa}]\,. \tag{35}$$

For a numerical example, the above differential equation has been solved numerically and the solutions $\phi = \phi(t)$ and $b = b(t)$ have been reported in the Fig. 10. The dynamics of the MEC learning equations for the single neuron considered is closely related to the dynamics of the simple pendulum subject to gravity, as clearly evidenced by the phase-plane plot.

The same graph also shows the kinetic and (lifted) potential energy functions during learning, which may be used to monitor the state of the neuron during the adaptation phase: The kinetic energy starts from zero and tends to zero, the potential energy reaches its minimum (in fact, the difference $U - U_\star$ reaches zero) and the total energy is a monotonically decreasing function of time, as predicted by the theory of Theorem 6.

A closed-form solution of the above differential equation would provide useful insight into the convergence properties of the learning equations. Unfortunately, the transcendental nature of the forcing term in the equation prevents us from finding closed form expressions for $\phi(t)$ and $b(t)$. However, in the hypothesis that the learning dynamics starts sufficiently close to the asymptotic solution $\phi_\star = \phi_P$, we can gain some qualitative indication from the approximated differential equation in the error term

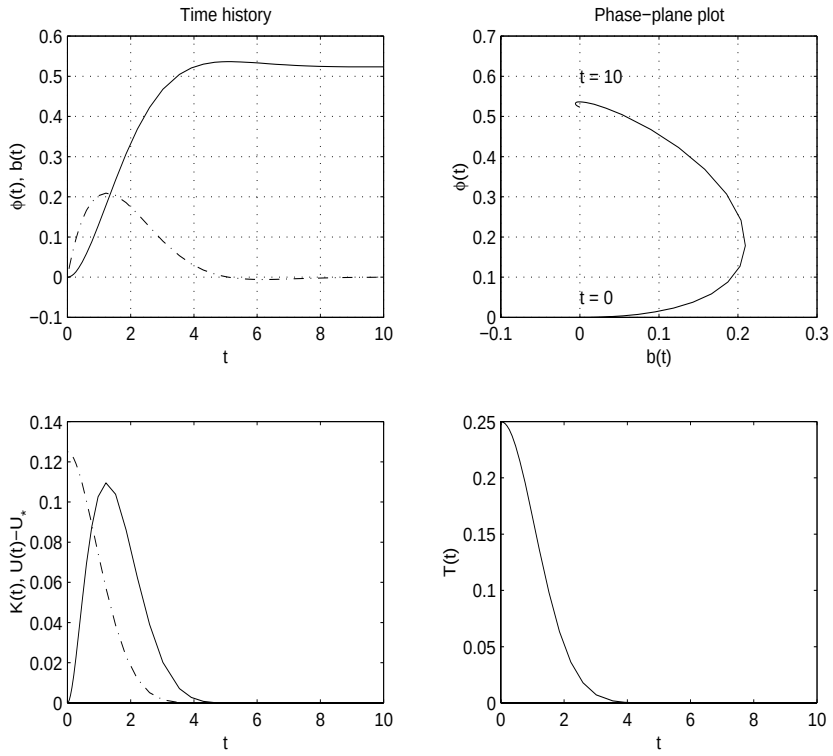

Fig. 10. Top: Example of solution of the differential equation (34). Left: Solutions $\phi = \phi(t)$ (solid-line) and $b = b(t)$ (dotted-line). Right: Phase-plane plot of equation's dynamics. Bottom: Learning state functions. Left: Kinetic energy (solid-line) and lifted potential energy (dotted-line); Right: Total energy.

$\Delta\phi \stackrel{\text{def}}{=} \phi - \phi_P \approx 0$. The approximated law comes from the approximation $\sin x \approx x$ for $x \approx 0$, which gives rise to the following initial-value problem:

$$\begin{aligned} &\frac{d^2(\Delta\phi)}{dt^2} + \mu\frac{d(\Delta\phi)}{dt} = -2\bar{\kappa}\Delta\phi\,, \\ &\Delta\phi(0) = \Delta\phi_0\,, \quad (\dot{\Delta\phi})(0) = 0\,, \end{aligned} \tag{36}$$

whose solution is easily found to be:

$$\Delta\phi(t) = c_1 e^{(-\mu-\sqrt{-8\bar{k}+\mu^2})t/2} + c_2 e^{(-\mu+\sqrt{-8\bar{k}+\mu^2})t/2}\,,$$

with constants c_1 and c_2 determined by the two initial conditions.

It is important to note that this solution is always stable, i.e. convergent to zero: In fact, surely $-8\bar{\kappa} + \mu^2 < \mu^2$, thus the solution contains at least some decaying terms; if, moreover, $-8\bar{\kappa}+\mu^2 < 0$, the solution contains an oscillating term weighted by a decaying exponential, namely:

$$\Delta\phi(t) = ce^{-\mu t/2}\cos(\sqrt{8\bar{\kappa} - \mu^2}t/2 + \psi)\,,$$

with constants c and ψ determined by the initial conditions. This expression, obtained under the hypothesis of small oscillations around the asymptotic solution, allows us to come to the following qualitative observations:

- The braking effect due to constant μ facilitates rapid convergence to the asymptotic solution and avoid oscillations around it, thus relatively high values of this constant should be preferred;
- The magnifying factor κ determines the frequency of oscillation of the variable ϕ around its asymptotic value, thus relatively low values of this constant should be preferred.

The oscillating solution should be avoided and the most favorable case is the one corresponding to a purely damped dynamics. The simulation results shown in Fig. 10 have been obtained by selecting the learning parameters in order to ensure damped solutions.

4.3. *A case-study on kurtosis extremization*

In order to give another example of mechanical-like LLG rule, here we present a detailed analysis of a case-study concerning one-unit learning based on kurtosis criterion.

Let us suppose we have two signals, arranged into the vector stream $\mathbf{s}(t) \in \mathcal{R}^2$, which are linearly mixed by a 2×2 full-rank orthonormal operator denoted as $\mathbf{A}$, which gives two signals arranged in $\mathbf{x}(t)$. About signals $s_i(t)$, we make the following hypotheses: (1) $\mathbb{E}_{\mathbf{s}}[s_i(t)] = 0$ (zero-mean), (2) $\mathbb{E}_{\mathbf{s}}[s_i^2(t)] = 1$ (spherical distributions), (3) $p_{s_i}(s_i) = p_{s_i}(-s_i)$ (symmetry around the mean value), (4) $p_{s_1 s_2}(s_1, s_2) = p_{s_1}(s_2)p_{s_2}(s_2)$ (statistical independence); as a consequence, the kurtosis κ_4 of the signals writes $\kappa_{4,i} = \mathbb{E}_{\mathbf{s}}[s_i^4(t)] - 3$; the last hypothesis about the signals is that $\kappa_{4,1}$ and $\kappa_{4,2}$ are not contemporarily null.

Let us also suppose we have a single linear neuron, described by $y(t) = \mathbf{m}^T(t)\mathbf{x}(t)$ with $\mathbf{m} \in \mathcal{R}^2$, which is trained to extract one of the two signals $s_i(t)$ from the mixture $\mathbf{x}(t)$.

In this section, we wish to study the behavior of simplified kurtosis-based learning criterion $U_{\mathrm{SK}}(\mathbf{m}) \stackrel{\mathrm{def}}{=} (1/4)w\mathbb{E}_{\mathbf{s}}[y^4]$, where w is a weighting term either positive or negative, depending on signal kurtoses signs. The gradient of the criterion thus writes $\nabla U_{\mathrm{SK}}(\mathbf{m}) = w\mathbb{E}_{\mathbf{s}}[y^3\mathbf{x}]$. With the usual parameterization, the gradient, thought of as a function of the angle ϕ, writes:

$$\begin{aligned}&\nabla(U_{\mathrm{SK}})_{\mathcal{H}}(\phi)\\ &= w\begin{bmatrix}\cos\phi_A & -\sin\phi_A\\ \sin\phi_A & \cos\phi_A\end{bmatrix}\begin{bmatrix}\kappa_{4,1}\cos^3(\phi-\phi_A)\\ \kappa_{4,2}\sin^3(\phi-\phi_A)\end{bmatrix}\\ &\quad + 3\begin{bmatrix}\cos\phi\\ \sin\phi\end{bmatrix}. \end{aligned} \tag{37}$$

The forcing term for the neuron is $\mathbf{f}(\phi) = -\nabla(U_{\mathrm{SK}})_{\mathcal{H}}$. Equation (12) in this case writes $\dot{\mathbf{m}} = \sigma\mathbf{B}\mathbf{m}$, where $\mathbf{B}$ in this case has the simple parameterization $\mathbf{B} = \begin{bmatrix}0 & b\\ -b & 0\end{bmatrix}$; the above expression simplifies into $\dot{\phi} = b$, where we have assumed $\sigma = 1$ for the sake of simplicity. Also, the term $\mathbf{p} = -\mu\mathbf{B}\mathbf{m}$ in this case writes:

$$\mathbf{p} = -\mu b\begin{bmatrix}\sin\phi\\ -\cos\phi\end{bmatrix}. \tag{38}$$

The learning term involved in Eqs. (12)–(30) writes $(\mathbf{f}+\mathbf{p})\mathbf{m}^T - \mathbf{m}(\mathbf{f}+\mathbf{p})^T$, whose $(1,2)$th entry accounts for the dynamics of variable b; after lengthy (though straightforward) calculations, we find:

$$\begin{aligned}\dot{b} &= w\kappa_{4,1}\cos^3(\phi-\phi_A)\sin(\phi-\phi_A)\\ &\quad + w\kappa_{4,2}\sin^3(\phi-\phi_A)\cos(\phi-\phi_A) - \mu b\,. \end{aligned} \tag{39}$$

In order to determine the equilibrium points of the mechanical system under kurtosis-based criterion, according to Theorem 4 we need to compute the solutions of equation:

$$\begin{aligned}&\sin(2(\phi-\phi_A))[\kappa_{4,1}+\kappa_{4,2}\\ &+ (\kappa_{4,1}-\kappa_{4,2})\cos(2(\phi-\phi_A))] = 0\,. \end{aligned} \tag{40}$$

Let us denote by $\mathcal{E}_{\phi_A} \stackrel{\mathrm{def}}{=} \{\phi = \phi_A + (\pi/2)n | n \in \mathcal{Z}\}$ the set of expected equilibrium points, corresponding to a correctly separating neuron. The above equations has solutions in $\mathcal{E}_{\phi_A}$, but may also have spurious attractors that satisfy:

$$-\frac{\kappa_{4,1}+\kappa_{4,2}}{\kappa_{4,1}-\kappa_{4,2}} = \cos(2(\phi-\phi_A))\,.$$

It is easy to show that the above equation admits a solution only when $\kappa_{4,1}\kappa_{4,2} < 0$, that is when the signals to be separated out are mixed sub-Gaussian and super-Gaussian. As an example, let us consider the case $\kappa_{4,1} = 1$ and $\kappa_{4,2} = -1$: the spurious equilibrium points are given by $\cos(2(\phi-\phi_A)) = 0$, that is, for instance, $\phi = \phi_A \pm (\pi/4)$. It is worth noting, however, that the existence of spurious equilibria does not necessarily mean that the system actually converges to one of them: The convergence properties depend on the values $\phi(0)$ and $b(0)$, among others.

In order to numerically investigate the behavior of mechanical learning system under kurtosis-based forcing field for $\kappa_{4,1} = 1$ and $\kappa_{4,2} = -1$, we performed three experiments, which refer to $w = -3$, $\mu = 4$ and $\phi_A = \pi/6$.

First, we simulated the learning equation with $b(0)$ randomly picked in $[-0.5, +0.5]$ and six different values of $\phi(0)$; the results are shown in the Fig. 11. The algorithm converges to either $\pi/6 \approx 0.5236$ rad, to $\pi/6 + \pi/2 \approx 2.0944$ rad or $\pi/6 + \pi \approx 3.6652$, depending on the basin of attraction that the initial guess belongs to; in this case no spurious solutions were encountered.

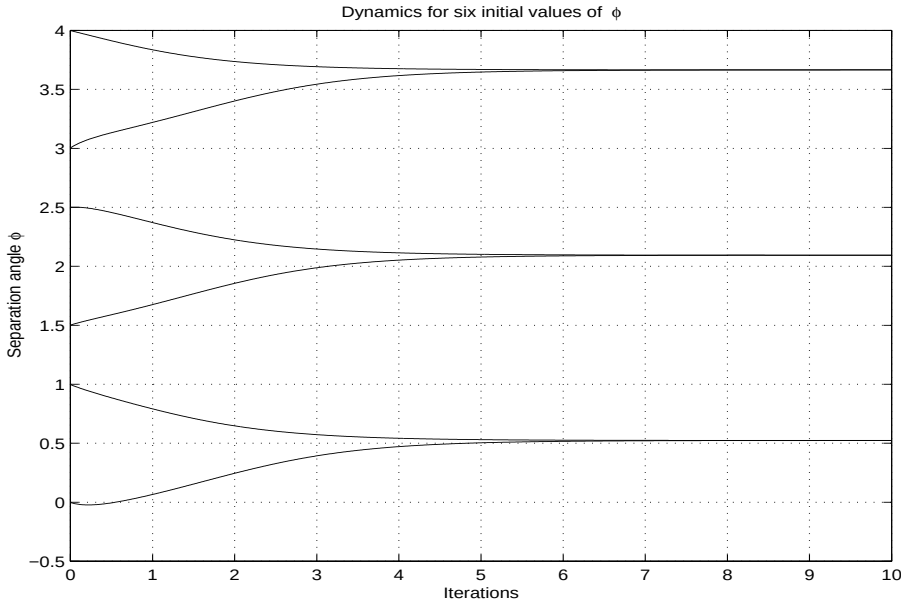

Fig. 11. Trials on six different guesses of initial separating angle.

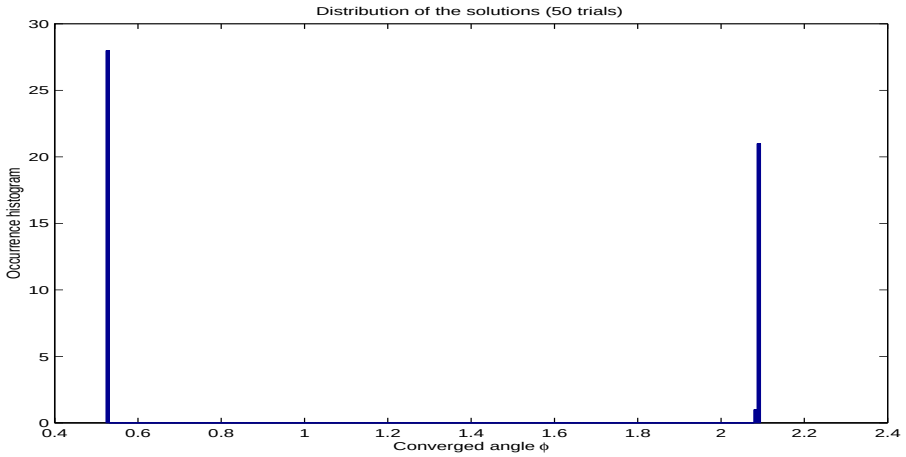

Fig. 12. Distribution of the solutions for 50 randomly chosen initial states.

The second experiment consisted in randomly picking both $b(0)$ (in $[-0.5, +0.5]$) and $\phi(0)$ in $[0, 2.5]$; 50 trials were performed and the states ϕ reached from $t = 0$ to $t = 10$ seconds were collected: In this way we were able to measure the distribution of the solutions $\phi_\star$ in $[0, \pi]$; the result is shown in the Fig. 12. Again the solutions distribute around the equivalently separating angle values $\pi/6$ rad and $\pi/6 + \pi/2$ rad.

In the third experiment, the learning equation was simulated on $b(0) = 0$ and $\phi(0) = \pi/6 + \pi/4$ rad, which from the mathematical analysis is known to be a spurious equilibrium point, and then from the same initial guess but with $b(0)$ randomly picked in $[-0.5, +0.5]$. The results are shown in Fig. 13. It clearly emerges that a non-zero initial speed may allow avoiding spurious equilibrium points, confirming the observation that a proper choice of matrix $\mathbf{B}(0)$ in second-order learning may provide additional control of learning behavior with respect to the first-order one.

As already done for the case of variance optimization, we wish to illustrate now the behavior of the differential equation (39) in relation to the theory

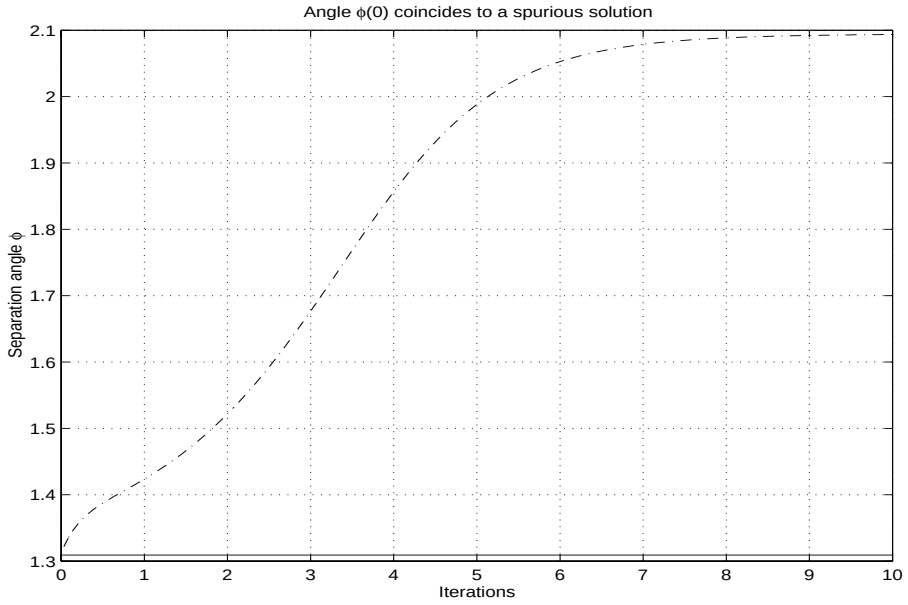

Fig. 13. Example of spurious solution avoidance property of mechanical system. Dynamics for $b(0) = 0$ (solid line) and dynamics for $b(0)$ picked-up randomly (dot-dashed line).

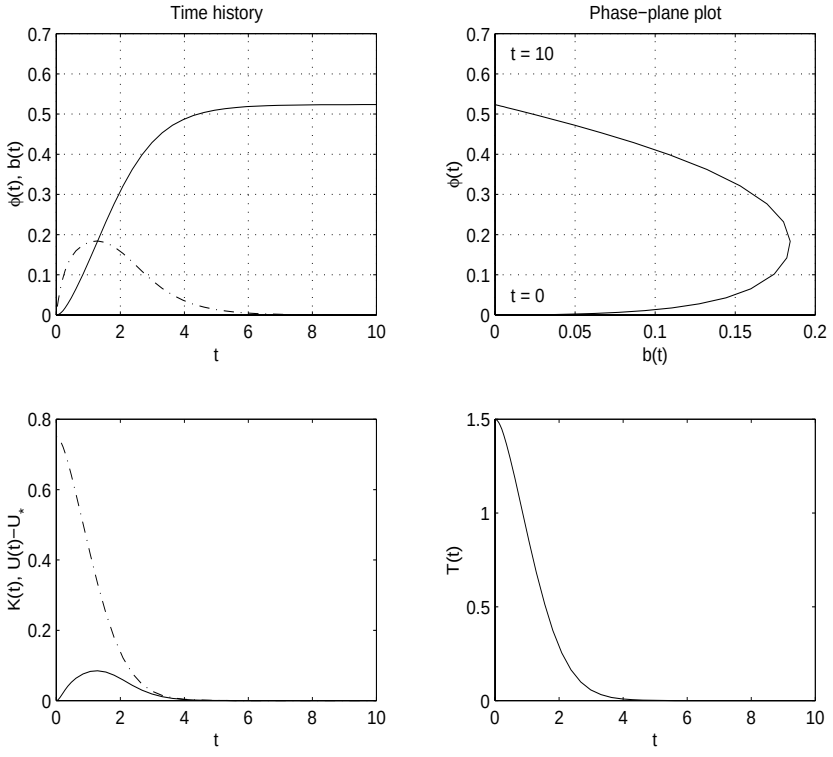

Fig. 14. Top: Example of solution of the differential equation (39). Left: Solutions $\phi = \phi(t)$ (solid-line) and $b = b(t)$ (dotted-line). Right: Phase-plane plot of equation's dynamics. Bottom: Learning state functions. Left: Kinetic energy (solid-line) and lifted potential energy (dotted-line); Right: Total energy.

about potential energy function and kinetic energy function.

The kinetic energy has the expression proportional to $b^2(t)$ already seen in the preceding section, while the potential energy function, in the present case, assumes the expression:

$$\begin{aligned} U(t) = \frac{w}{4}[&(\kappa_{4,1}+3)\cos^4(\phi(t)-\phi_A) \\ &+(\kappa_{4,2}+3)\sin^4(\phi(t)-\phi_A) \\ &+\frac{3}{2}\sin^2(2(\phi(t)-\phi_A))]\,. \end{aligned}$$

Its minimal value is attained for $\phi_\star = \phi_A$, which leads to $U_\star = (w/4)(\kappa_{4,1}+3)$; as a consequence, the lifted potential energy writes:

$$\begin{aligned} U(t)-U_\star = \frac{w}{4}[&(\kappa_{4,1}+3)(\cos^4(\phi(t)-\phi_A)-1) \\ &+(\kappa_{4,2}+3)\sin^4(\phi(t)-\phi_A) \\ &+\frac{3}{2}\sin^2(2(\phi(t)-\phi_A))]\,. \end{aligned} \tag{41}$$

For a numerical example, the learning differential equation (39) has been solved numerically with zero initial speed and zero initial solution, and the traces of $\phi = \phi(t)$ and $b = b(t)$ have been reported in Fig. 14.

The same graph also shows the kinetic and lifted potential energy functions during learning. The kinetic energy starts from a value equal to zero and tends to zero, the potential energy reaches its minimum and again the total energy is a monotonically decreasing function of time, as predicted by the theory.

5. Implementation Issues

The aim of this section is to briefly discuss the important topic of computer-based implementation of the presented learning rules. In fact, in practical computer-based implementations, discrete-time counterparts of the presented general first- and second-order learning equations on Lie group are necessary; in other words, it is necessary to define discrete-time learning algorithms on the basis of the continuous-time learning rules presented in the above sections, *ensuring that the algorithms keep LLG*.

As an interesting result, by properly performing the discretization operation and by suitable approximation, we are able to explain within the "mechanical" learning framework two learning algorithms that recently appeared in the scientific literature, and they can therefore be explained within the general learning framework proposed in the present paper.

Other important questions are worth discussing in the present section, such as the problem related to the practical representation of the quantities required for network learning such as matrices $\mathbf{M}$ and $\mathbf{B}$, and the problem of efficient computation of matrix operations in the learning equations.

5.1. *Discrete-time counterparts of the LLG equations*

The simplest way for determining a discrete-time counterpart of the learning equations described before is to employ the standard sampling method, consisting in determining a sufficiently narrow time-slice where the learning variables are almost stationary, say η, and replacing derivatives $d\mathbf{x}/dt$ with $\Delta\mathbf{x}/\eta$, where $\Delta\mathbf{x} = \mathbf{x}(\eta(t+1)) - \mathbf{x}(\eta t)$ and where $t \in \mathcal{Z}$ now denotes a discrete-time index. Let us see what this implies on systems (7) and (13).

For system (7), it can be simply discretized as:

$$\Delta\mathbf{M} = \eta\nabla U(\mathbf{M})\,, \tag{42}$$

where η plays the role of a learning step-size, whose magnitude controls the speed and the accuracy of the learning steps.

For system (13), it is more easy to consider its equivalent form Eq. (12).

The equation describing the evolution of matrix $\mathbf{B}$ may be simply discretized by sampling, as it remains skew-symmetric:

$$\Delta\mathbf{B} = \eta\mathbf{S}[\mathbf{H}]\,. \tag{43}$$

Now $\mathbf{B}$ is piece-wise constant, thus the differential equation for $\mathbf{M}$ may be exactly solved and gives:

$$\Delta\mathbf{M} = (e^{\eta\sigma\mathbf{B}} - \mathbf{I}_p)\mathbf{M}\,. \tag{44}$$

In opposition to rule (42), which no longer describes a LLG, rule (44) does generate a LLG, as can be easily proven by noting that $(e^{\eta\sigma\mathbf{B}})^T = e^{-\eta\sigma\mathbf{B}}$.

Matrix $e^{\mathbf{X}}$ can be computed either by the truncated series $e^{\mathbf{X}} \approx \sum_{k=0}^{r}\mathbf{X}^k/k!$, with $r \in \mathcal{Z}^+$, or by canonical eigenvalue decomposition[42] $e^{\mathbf{X}} = \mathbf{V}\mathbf{R}\mathbf{V}^T$, with $\mathbf{V}$ being an orthogonal matrix and $\mathbf{R}$ being a block-diagonal matrix with 2×2 skew-symmetric

blocks and null scalar blocks. Clearly, when the matrix exponentiation is approximated, Eq. (44) no longer describes LLG's; however, if η is sufficiently small, the LLG approximation holds with good faith.

The question of time-discretization is intimately related to the question of sequential parameter estimation. Irrespective of their nature, the learning trajectories on the Lie group have been supposed to be driven by an average autonomous criterion, i.e. a smooth function of networks' adjustable parameters only. Namely, we represent the learning criterion as $U(\mathbf{M}) = \mathbb{E}_{\mathbf{x}}[u(\mathbf{x}, \mathbf{y}, \mathbf{M})]$, where $u(\cdot,\cdot,\cdot)$ is a stochastic measure of network's performance. However, in many practical applications the average performance $U(\cdot)$ is unavailable. In this case we may resort to stochastic adaptation also known as *sequential parameter estimation*.

Sequential methods for parameter estimation rely on iterative algorithms to update the values of parameters as new data become available. These methods play an important role in signal processing and pattern recognition for three main reasons: (1) they do not require storage of a complete data set since each datum can be discarded once it has been used, making them very efficient when large volumes of data are to be handled; (2) they can be employed for on-line learning in real-time adaptive systems; (3) in case of operation under non-stationary conditions, i.e. when the process which generates the data is slowly-varying, the parameters values can continuously adapt and can therefore track the behavior of the process.

From a more formal viewpoint, the invoked adapting algorithms may be regarded as procedures for finding the roots of functions which are defined stochastically. To give an example, let us consider two scalar variables u and m, which are correlated; the average of u for each m defines a function $g(m) \stackrel{\text{def}}{=} E[u|m]$. In the hypothesis that several observations of the variable u for a given value of m are available, we have a set of random values whose mean, thought of as a function of m, is usually referred to as *regression function*. A general procedure for finding the roots $m^\star$ of such function was given by Robbins and Monro,[49] which reads:

$$m(t+1) = m(t) + \eta_t u(m(t))\,;$$

under four main conditions[25,49] on u, g, and the sequence of learning stepsizes η_t, it can be proven that the sequence of estimates $m(t)$ converges to one of the roots $m^\star$ with probability one (see also Ref. 37). Such stochastic sequential approximation scheme was extended to the multidimensional case by Blumm.[6] Analogously, in the present paper we derived results for the expected criteria/algorithms, and suppose they hold for their stochastic counterparts, too.

5.2. *Approximated "mechanical" learning equations*

The "mechanical" learning system is very general as it takes into account at any time the forces which act on the point of coordinates $\mathbf{M}$. Useful simplified algorithms may be obtained by relaxing this strict scheme. These may be obtained as approximations of the proposed second-order LLG as may be illustrated by making use of the following informal reasoning.

Let us hypothesize that the mechanical system follows a continuous regular motion within a medium where the viscous effect is negligible, i.e. $\mu \approx 0$. It is thus described by:

$$\dot{\mathbf{M}} = \sigma \mathbf{B}\mathbf{M},\ \dot{\mathbf{B}} = \mathbf{H} - \mathbf{H}^T,\ \mathbf{H} = \mathbf{F}\mathbf{M}^T,\ t \in \mathcal{T}\,, \qquad (45)$$

where the notation reflects the definitions of Theorem 4. Let us divide the time-interval $\mathcal{T}$ into a set of time-intervals $\mathcal{T}_i \stackrel{\text{def}}{=} [t_i^-\ t_i^+]$ such that $\bigcup_i \mathcal{T}_i = \mathcal{T}$ and $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$ for any $i \neq j$, and let us denote the duration of each time-interval as $|\mathcal{T}_i| \stackrel{\text{def}}{=} t_i^+ - t_i^-$.

The average value of $\mathbf{B}$ within $\mathcal{T}_i$ is easily computed as:

$$\begin{aligned}\bar{\mathbf{B}}_i &\stackrel{\text{def}}{=} \frac{\mathbf{B}(t_i^+) - \mathbf{B}(t_i^-)}{|\mathcal{T}_i|} \\ &= \frac{1}{|\mathcal{T}_i|}\int_{t_i^-}^{t_i^+} [\mathbf{H}(\tau) - \mathbf{H}^T(\tau)]d\tau \\ &= \mathbf{H}(\tau_i) - \mathbf{H}^T(\tau_i)\,,\end{aligned}$$

where τ_i is an appropriate value in $\mathcal{T}_i$, and $\mathbf{H}(\tau_i) = \mathbf{F}(\tau_i)\mathbf{M}^T(\tau_i) \stackrel{\text{def}}{=} \mathbf{F}_i\mathbf{M}_i^T$. The "average motion" of the system (45) thus obeys the equation:

$$\dot{\mathbf{M}} = \sigma(\mathbf{F}_i\mathbf{M}_i^T - \mathbf{M}_i\mathbf{F}_i^T)\mathbf{M}\,, \quad t \in \mathcal{T}_i\,. \qquad (46)$$

The above approximated learning rule closely resembles the Douglas–Kung rule[16] which, for the linear neural network (4) and for criterion $U(\mathbf{M})$ recasts

into:

$$\dot{\mathbf{M}} = -((\nabla U)\mathbf{M}^T - \mathbf{M}(\nabla U)^T)\mathbf{M}\,. \qquad (47)$$

Clearly, the more $|\mathcal{T}_i|$ approaches zero, the more system (46) approaches system (47).

Another closely related learning algorithm is the one proposed by Nishimori in Refs. 42 and 43 which may arise from the solution of the continuous-time equation (46) $\dot{\mathbf{M}}(\tau) = \bar{\mathbf{X}}\mathbf{M}(\tau)$ within the interval $\tau \in [\eta t,\, \eta(t+1)[$ and with $\bar{\mathbf{X}} = \mathbf{X}(\eta t)$.

Nishimori's learning equation is closely related to Douglas–Kung rule: Let us show the mutual relationships among the cited algorithms. The rule given in Ref. [42] for a (discrete-time) linear network is:

$$\Delta\mathbf{M} = (e^{-\eta\mathbf{X}} - \mathbf{I}_p)\mathbf{M}\,, \qquad (48)$$

where $\mathbf{X}$ is a skew-symmetric matrix and η is a small positive constant. Matrix $\mathbf{X}$ has the expression:

$$2\mathbf{X} = (\nabla U)\mathbf{M}^T - \mathbf{M}(\nabla U)^T\,. \qquad (49)$$

Note that in the limit $\eta \to 0$ we have $e^{-\eta\mathbf{X}} \to \mathbf{I}_p - \eta\mathbf{X}$, thus Eq. (48) tends to Eq. (47), which therefore results in being a first-order approximation of Eq. (48).

It is worth pointing out that Nishimori's algorithm is the only one, among the learning equations considered in this section, generating exact Lie-group learning under discrete-time operation mode. It is also worth noting that the work presented in Ref. 42 is closely related to the work about "exponentiated gradient" presented in Ref. 35.

5.3. *Efficient representation and computation*

For computer-based implementations it is useful to note that $\mathbf{B}$ is a $p \times p$ matrix with only $p(p-1)/2$ distinct entries in general, that is its lower-triangular part can be obtained from the upper-triangular part and thus need not be stored in memory.

A similar consideration might be carried out for $\mathbf{M}$ which, however, requires a more detailed discussion. The more appropriate framework for discussing this topic is the question about the choice of the representation of the network's variables. Within the paper we made use of two different representations relying on extrinsic variables, namely the entries $m_{ij}(t)$ of the matrix $\mathbf{M}(t)$, and intrinsic variables, namely the curvilinear coordinate or the network angle ϕ. In principle the two representations are equivalent. However, in order to prove theorems and to fix the concept of criteria restriction over manifolds, the intrinsic coordinate systems are more suited and provide a better insight than the extrinsic ones; also, usually the number of intrinsic coordinates, when used for example to parameterize quantities over the manifolds, coincides to the dimension of the manifold itself, which is by definition the smallest-dimension set of free coordinates required to uniquely represent a point on the manifolds. In contrast, we found that in order to represent the involved quantities on a computer, the most advantageous representation relies on extrinsic coordinates, that is, we represent the connection matrix $\mathbf{M}$ as a standard $p \times q$ matrix with pq variables/entries, even if we know these variables are not to be all independent. This practical consideration has been supported by other authors in the numerical matrix analysis field.[17]

The same choice may be justified from another point of view, related to intrinsic parameterization singularities. The theory of (local) Lagrange variables suggests a way to represent the matrix $\mathbf{M}$ with the smallest number of free parameters; however, it can be easily seen that the common parameterizations which require the lowest number of parameters are quite difficult to handle in practical computer implementation also because of coordinate singularities. It is in fact well known that the Lagrangean coordinates systems may in general be defined only locally; this fact suggests the necessity to handle a set of local coordinate systems for the same problem, taking care of singularities in the boundaries separating a local system from each other (interested readers would find a detailed discussion on this topic in Ref. 17).

The last point we wish to briefly discuss here deals with the problem of efficient computation of the matrix-operations required to implement the proposed second-order LLG learning algorithm. In particular, from discrete-time "mechanical" learning equations and its best approximation, given by Nishimori's algorithm, it is seen that the most computationally burdensome expression is the matrix exponentiation $\exp(\mathbf{C})$ of skew-symmetric terms of the type $\mathbf{C} = \mathbf{A}_1\mathbf{A}_2^T - \mathbf{A}_2\mathbf{A}_1^T$. We wonder what is a

computationally convenient technique for implementing such calculus on a computer.

The answer comes from the recent numerical matrix-analysis paper,[11] which presents a computationally advantageous method for performing such calculations. In that paper, it is hypothesized that both $\mathbf{A}_1$ and $\mathbf{A}_2$ belong to $\mathcal{R}^{p\times q}$; integers p and $q \leq p$ may assume arbitrary values, but the method is proven particularly profitable when $2q \ll p$. First, the skew-symmetric matrix $\mathbf{C}$ is regarded as a product of the type $\mathbf{G}_1\mathbf{G}_2^T$, that can be obtained by the previous expression by defining the matrix-pencils $\mathbf{G}_1 \stackrel{\text{def}}{=} [\mathbf{A}_1 - \mathbf{A}_2]$ and $\mathbf{G}_2 \stackrel{\text{def}}{=} [\mathbf{A}_2 \ \mathbf{A}_1]$, where now both $\mathbf{G}_1$ and $\mathbf{G}_2$ belong to $\mathcal{R}^{p\times 2q}$. Then, the authors of Ref. 11 show how to compute $\exp(\mathbf{C})$ on the basis of $\mathbf{G}_2^T\mathbf{G}_1$ which is a considerably smaller $2q\times 2q$ matrix. Using the conclusion of Ref. 11, under the mentioned hypotheses, the complexity of the whole neural-network parameters updating computation is of the order of $\mathcal{O}(pq^2)$ flops.

6. Conclusions

The large amount of specific algorithms for orthonormal learning in neural networks and of experimental results appearing in the literature, concerning topics such as principal/independent component analysis, suggests the importance of a unifying theoretical framework able to explain and encompass the many different contributions.

The aim of this paper was to present some general considerations on learning on Lie group, its usefulness in signal/data processing, and general theoretical results about it, along with a discussion on the latest issues appearing in the scientific literature concerning this topic.

General results on first- and second-order LLG algorithms have been given, and hidden properties of some learning theories known from the literature and relationships between them have been disclosed by recognizing the differential geometry of Lie groups as the natural instrument for studying the properties of learning occurring on a weight-space endowed with a specific Lie-group structure.

Acknowledgments

The present paper was finished after my attendance at the First European Meeting on Independent Component Analysis, held in February 2002 in Vietri sul Mare (Italy), and brings ideas which came from fruitful discussions with other attendees after my presentation of some of the unpublished concepts reported here. Especially, I wish to gratefully thank the organizers of the meeting, Dr. M. Funaro and Prof. M. Marinaro (University of Salerno, Italy), for inviting me to give the talk and the chairman of the session, Prof. E. Oja (Helsinki University of Technology, Finland) and coworkers for the interesting and stimulating inquiries, comments and suggestions; I would like to sincerely thank Dr. E. Celledoni and Prof. B. Owren (Trondheim University of Science and Technology, Norway) for the fruitful discussion on Lie group theory and methods and the useful pointers to papers on the numerical solution of matrix differential equations defined on Lie group.

References

1. S. Affes and Y. Grenier 1995, "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.
2. S.-I. Amari 1998, "Natural gradient works efficiently in learning," *Neural Computation* **10**, 251–276.
3. S.-I. Amari 1999, "Natural gradient learning for over- and under-complete bases in ICA," *Neural Computation* **11**, 1875–1883.
4. C. Aluffi-Pentini, V. Parisi and F. Zirilli 1985, "Global optimization and stochastic differential equations," *J. Optimization Theory and Applications* **47**, 1–16.
5. A. J. Bell and T. J. Sejnowski 1996, "An information maximisation approach to blind separation and blind deconvolution," *Neural Computation* **7**(6), 1129–1159.
6. R. Blumm 1954, "Multidimensional stochastic approximation methods," *Annals of Mathematical Statistics* **25**, 737–744.
7. G. E. Bredon 1995, *Topology and Geometry* (Springer-Verlag, New York).
8. R. W. Brockett 1991, "Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems," *Linear Algebra and Its Applications* **146**, 79–91.
9. J.-F. Cardoso 1998, "Blind signal separation: Statistical Principles," *Proc. IEEE* (*special issue on "Blind Identification and Estimation,"* eds. R.-W. Liu and L. Tong) **90**(8), 2009–2026.
10. J. F. Cardoso and B. Laheld 1996, "Equivariant adaptive source separation," *IEEE Trans. on Signal Processing* **44**(12), 3017–3030.
11. E. Celledoni and B. Owren 2001, "On the implemen-

tation of Lie group methods on the Stiefel manifold," Preprint Numerics No. 9/2001 (Norwegian University of Science and Technology, Trondheim, Norway).
12. T. P. Chen, S. Amari and Q. Lin 1998, "A unified algorithm for principal and minor components extraction," *Neural Networks* **11**(3), 385–390.
13. A. Chicocki, J. Karhunen, W. Kasprzak and R. Vigario 1999, "Neural networks for blind separation with unknown number of sources," *Neurocomputing* **24**, 55–93.
14. S. Costa and S. Fiori 2001, "Image compression using principal component neural networks," *Image and Vision Computing Journal* (*special issue on "Artificial Neural Network for Image Analysis and Computer Vision"*) **19**(9&10), 649–668.
15. Y. le Cun, L. D. Jackel, B. E. Boser, J. S. Denker, H.-P. Graf, I. Guyon, D. Henderson, R. E. Howard and W. Hubbard 1989, "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, pp. 41–46.
16. S. C. Douglas and S.-Y. Kung 1999, "An ordered-rotation Kuicnet algorithm for separating arbitrarily-distributed sources," *Proc. Int. Conf. on Independent Component Analysis* (ICA'99), pp. 81–86.
17. A. Edelman, T. A. Arias and S. T. Smith 1998, "The geometry of algorithms with orthogonality constraints," *SIAM J. on Matrix Analysis Applications* **20**(2), 303–353.
18. Y. Ephraim and L. Van Trees 1995, "A signal subspace approach for speech enhancement," *IEEE Trans. on Speech and Audio Processing* **3**(4), 251–266.
19. S. Fiori and F. Piazza 2000, "A general class of APEX-like PCA neural algorithms," *IEEE Trans. on Circuits and Systems – Part I* **47**(9), 1394–1398.
20. S. Fiori 2000, "Blind signal processing by the adaptive activation function neurons," *Neural Networks* **13**(6), 597–611.
21. S. Fiori 2001, "A theory for learning by weight flow on Stiefel–Grassman manifold," *Neural Computation* **13**(7), 1625–1647.
22. S. Fiori 2002, "Hybrid independent component analysis by adaptive LUT activation function neurons," *Neural Networks* **15**(1), 85–94.
23. S. Fiori 2002, "A theory for learning based on rigid bodies dynamics," *IEEE Trans. on Neural Networks* **13**(3), 521–531.
24. A. Fujiwara and S.-I. Amari 1995, "Gradient systems in view of information geometry," *Physica* **D80**, 317–327.
25. K. Fukunaga 1990, *Introduction to Statistical Pattern Recognition*, 2nd edition (Academic Press, San Diego).
26. K. Gao, M. O. Ahmed and M. N. Swamy 1994, "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 – Part II* **41**(11), 718–729.
27. M. Girolami 2000, *Self-Organizing Neural Networks* (Springer-Verlag).
28. S. Gold, A. Rangarajan and E. Mjolsness 1996, "Learning with preknowledge: Clustering with point and graph matching distance," *Neural Computation* **8**, 787–804.
29. J. C. Gower 1984, "Ordination, multidimensional scaling and allied topics," ed. E. Lloyd, *Handbook of Applicable Mathematics*, Vol. VI (John Wiley & Son).
30. S. Hochreiter and M. C. Mozer, "Coulomb classifiers: Reinterpreting SVMs as electrostatic systems," Technical report CU-CS-921-01, Dept. of Computer Science, University of Colorado.
31. D. J. Hurley, M. S. Nixon and J. N. Carter 2002, "Force field energy functionals for image feature extraction image and vision computing," Vol. 20, Issue 5–6, pp. 311–317.
32. A. Hyvärinen and E. Oja 1998, "Independent component analysis by general non-linear Hebbian-like rules," *Signal Processing* **64**(3), 301–313.
33. J. Karhunen 1996, "Neural approaches to independent component analysis and source separation," *Proc. of ESANN'96*, pp. 249–266.
34. A. Kern, D. Blank and R. Stoop 2000, "An optimal noise cleaning by local manifold projection," *Proc. of 2nd Int. ICSC Symposium on Neural Computation* (NC), pp. 399–404.
35. J. Kivinen and M. Warmuth 1997, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation* **132**, 1–64.
36. R.-W. Liu 1996, "Blind signal processing: an introduction," *Proc. of Int. Symposium on Circuits and Systems* (IEEE-ISCAS) **2**, pp. 81–84.
37. L. Ljung 1977, "Analysis of recursive stochastic algorithms," *IEEE Trans. on Automatic Control* **AC-22**, 551–575.
38. M. J. McKeown, S. Makeig, G. G. Brown, T.-P. Jung, S. S. Kindermann, A. J. Bell and T. J. Sejnowski 1998, "Analysis of fMRI data by blind separation into independent spatial components," *Human Brain Mapping* **6**, 160–188.
39. B. C. Moore 1981, "Principal component analysis in linear systems: controllability, observability and model reduction," *IEEE Trans. on Automatic Control* **AC-26**(1), 17–31.
40. E. Moreau and J. C. Pesquet 1997, "Independence/decorrelation measures with application to optimized orthonormal representations," *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 3425–3428.
41. H. Niemann and J.-K. Wu 1993, "Neural network adaptive image coding," *IEEE Trans. on Neural Networks* **4**(4), 615–627.

42. Y. Nishimori 1999, "Learning algorithm for ICA by Geodesic flows on orthogonal group," *Proc. Int. Joint Conference on Neural Networks* (IJCNN'99) **2**, pp. 1625–1647.
43. Y. Nishimori 2001, "Multiplicative learning algorithm via Geodesic flows," *Proc. Int. Symposium on Nonlinear Theory and Its Applications* (NOLTA'01) **2**, pp. 529–532.
44. E. Oja 1989, "Neural networks, principal components, and subspaces," *Int. J. Neural System* **1**, 61–68.
45. E. Oja, A. Hyvärinen and P. Hoyer 1999, "Image feature extraction and denoising by sparse coding," *Pattern Analysis and Applications Journal* **2**(2), 104–110.
46. A. Paraschiv-Ionescu, C. Jutten and G. Bouvier 1997, "Neural network based processing for smart sensor arrays," *Artificial Neural Networks*, pp. 565–570.
47. S. J. Perantonis and D. A. Karras 1995, "An efficient learning algorithm with momentum acceleration," *Neural Networks* **8**, 237–249.
48. E. Pfaffelhuber 1975, "Correlation memory models — A first approximation in a general learning scheme," *Biological Cybernetics*, **18**, 217–223.
49. H. Robbins and S. Monro 1951, "A stochastic approximation method," *Annals of Mathematical Statistics* **22**, 400–407.
50. P. Saisan, G. Doretto, Y. N. Wu and S. Soatto 2001, "Dynamic texture recognition," *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2**, pp. 58–63.
51. D. Sona, A. Sperduti, and A. Starita 2000, "Discriminant pattern recognition using transformation invariant neurons," *Neural Computation* **12**(6), 1355–1370.
52. E. Stiefel 1935–36, "Richtungsfelder und Fernparallelismus in n-Dimensionalem Manning Faltigkeiten," *Commentarii Math. Helvetici* **8**, 305–353.
53. R. Tagliaferri, A. Ciaramella, L. Milano and F. Barone 1999, "Neural networks for spectral analysis of unevenly sampled data," *Proc. XI Italian Workshop on Neural Networks* (WIRN'99), pp. 226–233.
54. I.-T. Um, J.-J. Wom and M.-H. Kim 2000, "Independent component based Gaussian mixture model for speaker verification," *Proc. of 2nd Int. ICSC Symposium on Neural Computation* (NC), pp. 729–733.
55. D. J. Willshaw and H. L. Longuet-Higgins 1969, "The holopone — Recent developments," *Machine Intelligence*, eds. B. Metzler and D. Michie (Edimburg University Press) **4**, pp. 349–357.
56. L. Xu 1994, "Theories for unsupervised learning: PCA and its nonlinear extension," *Proc. of Int. Joint Conference on Neural Networks*, pp. 1252–1257.
57. L. Xu, E. Oja and C. Y. Suen 1992, "Modified Hebbian learning for curve and surface fitting," *Neural Networks* **5**, 393–407.
58. B. Yang 1995, "Projection approximation subspace tracking," *IEEE Trans. on Signal Processing* **43**, 1247–1252.
59. K. Zhang and T. J. Sejnowski 1999, "A theory of geometric constraints on neural activity for natural three-dimensional movement," *J. Neuroscience* **19**(8), pp. 3122–3145.