International Journal of Neural Systems, Vol. 13, No. 2 (2003) 1–16

World Scientific
www.worldscientific.com

SINGULAR VALUE DECOMPOSITION LEARNING ON DOUBLE STIEFEL MANIFOLD

SIMONE FIORI
Faculty of Engineering, Perugia University,
Loc. Pentima bassa, 21, I-05100 Terni (Italy)
sfr@unipg.it

Received 25 October 2002
Revised 5 March 2003
Accepted 5 March 2003

The aim of this paper is to present a unifying view of four SVD-neural-computation techniques found in the scientific literature and to present some theoretical results on their behavior. The considered SVD neural algorithms are shown to arise as Riemannian-gradient flows on double Stiefel manifold and their geometric and dynamical properties are investigated with the help of differential geometry.

Keywords: Singular value decomposition; orthogonal matrix group; Stiefel manifold; differential geometry; Lyapunov stability.

1. Introduction

The computation of the singular value decomposition (SVD) of a non-square matrix, also referred to as Autonne–Eckart–Young decomposition[18,30] plays a central role in several signal/data automatic processing. Originally developed in numerical algebra to provide quantitative information about the structure of linear systems of equations, it has found widespread applications e.g. in signal processing,[7–8,17,31] pattern recognition and classification,[26] automatic control,[24,30] digital circuit design, time-series prediction,[28] image processing[6,21,25] and connectionism.[2]

Recently, some efforts have been devoted to SVD computation by neural networks in the neural community;[5,27,33] the related learning theories emerge as interesting extensions of the well-known neural principal component/subspace analysis techniques,[26] long investigated during the last 15 years. Also, recently a new light has been shed on adaptive second-order (as well as higher-order) statistical decomposition theories by researchers interested in unsupervised learning by non-gradient techniques: For instance, in Ref. 1 a new technique was introduced to enhance the learning capabilities of linear and MLP-type neural networks by Riemannian gradient, in Refs. 4 and 13 a theoretical derivation/analysis of new principal/minor subspace rules has been carried out; also, in Refs. 9 and 12 a large class of learning rules for MLP-type neural networks, based on first/second-order non-gradient dynamics and Lie-group flows, has been introduced and discussed by the present Author as a theoretical framework for explaining many learning paradigms appeared on the scientific literature, while papers[10,11] were devoted to a particular algorithm of this class, based on the rational kinematics of rigid bodies and its applications to real- and complex-valued signal processing.

The aim of this paper is to present some theoretical notes on parallel SVD computation by unsupervised non-gradient neural learning, with special reference to learning theories involving weight-flows on double Stiefel manifold. Parallel techniques are considered in opposition to sequential ones that employ the deflation method,

implemented by laterally connected neural architectures, to discard previously computed vectors from original data.[5,8] In particular, we recall from the scientific literature four neural SVD learning theories appeared independently; then, as novel contribution to this field, we present:

- A unifying view of the mentioned theories, showing the main relationships among them;
- A stability analysis based on Lyapunov criterion, aimed at ensuring the non-divergence of the differential equations governing the learning phases of the SVD neural networks trained via the considered methods;
- A computer-based analysis of the learning differential equations carried out in order to assess their numerical properties.

Through the paper we use the following notation. Symbol $I_{m,n}$ denotes the pseudo-identity matrix of size $m \times n$ and $I_m = I_{m,n}$. Symbol X' denotes the transposition of the matrix X while X^* denotes Hermitian-transposition; symbol $\mathrm{tr}(X)$ denotes the trace of the square matrix X, i.e. the sum of its in-diagonal entries; the trace operator enjoys the following properties: $\mathrm{tr}(X') = \mathrm{tr}(X)$, $\mathrm{tr}(ABC) = \mathrm{tr}(CAB) = \mathrm{tr}(BCA)$. We also define the two matrix operators $\{X,Y\} \stackrel{\mathrm{def}}{=} X'Y - Y'X$ and $[X,Y] \stackrel{\mathrm{def}}{=} X'Y + Y'X$. The following matrix set (termed Stiefel manifold) is also useful to our expository purposes: $\mathrm{St}(m,n,\mathbb{K}) \stackrel{\mathrm{def}}{=} \{X \in \mathbb{K}^{m\times n} | X^*X = I_n\}$ with $m-1$, $n-1 \in \mathbb{N}$; the field $\mathbb{K}$ may be either $\mathbb{R}$ or $\mathbb{C}$; when $m = n$ the manifold coincides with the orthogonal group $O(m,\mathbb{K}) \stackrel{\mathrm{def}}{=} \{X \in \mathbb{K}^{m\times m} | X^*X = I_m\}$. We refer to the product $O(m,\mathbb{K}) \times O(n,\mathbb{K})$ as double orthogonal group and to the product $\mathrm{St}(m,p,\mathbb{K}) \times \mathrm{St}(n,p,\mathbb{K})$ as double Stiefel manifold (some definitions and notes on these geometrical entities are available in the appendix A.1). Also, the Frobenius norm of a matrix $X \in \mathbb{R}^{n\times n}$ is defined as $\|X\|_{\mathrm{F}} \stackrel{\mathrm{def}}{=} \sqrt{\mathrm{tr}(X^*X)}$.

2. Four Parallel SVD Learning Algorithms: A Unifying View

Denoting as $Z \in \mathbb{C}^{m\times n}$ the matrix whose SVD is to be computed and as $r \leq \min\{m,n\}$ the rank of Z, the singular value decomposition writes $Z = UDV^*$, where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices and D is a pseudo-diagonal matrix that has all-zero values except for the first r diagonal entries, termed singular values. It is easily checked that the columns of U coincide with the eigenvectors of ZZ^* while V contains the eigenvectors of Z^*Z with the same eigenvalues.

Here we consider four *parallel* SVD learning algorithms, which allow to simultaneously compute the SVD vectors. The considered neural algorithms have been developed by Weingessel and Hornik[32] and by Helmke and Moore.[18] These algorithms are utilized to train in an unsupervised way a three-layer neural network with the classical 'butterfly' topology (see e.g. Refs. 8, 32 and 33): The first layer has connection matrix A, the second one has connection-matrix B and the middle (hidden) layer provides network's output. Properly learnt, the network is able to perform the mentioned signal/data processing tasks, such as noise filtering.[8]

The aim of this section is to analytically show that the algorithms proposed by Weingessel–Hornik and Helmke–Moore are equivalent to some extent. Also, it is showed that when proper initial conditions are chosen the associated learning trajectories lie on the double Stiefel manifold.

2.1. *The WH2, WH3 and WH4 neural SVD-subspace dynamical systems*

In Ref. 32 some new learning equations have been introduced by Weingessel and Hornik in order to compute the SVD-subspace of a given matrix. Here we investigate on three of them, expressed as continuous-time differential equations. The derivations presented below make use of the matrix differential calculus: A source reference for this is Ref. 23.

Let us denote as $A(t) \in \mathbb{R}^{m\times p}$ the network-connection matrix-stream that should learn p left singular vectors and as $B(t) \in \mathbb{R}^{n\times p}$ the estimator for p right singular vectors of the SVD of matrix $Z \in \mathbb{R}^{m\times n}$, with $p \leq r \leq \min\{m,n\}$, where r denotes again the rank of matrix Z. The algorithm WH2[32] reads:

$$\begin{cases} \dot{A} = ZB - AB'Z'A\,, & A(0) = A_0\,, \\ \dot{B} = Z'A - BA'ZB\,, & B(0) = B_0\,. \end{cases} \tag{1}$$

It has been derived by extending Brockett's work on isospectral flow systems[3] from single to double orthogonal group; the initial state A_0, B_0 of the

dynamical equations may be freely chosen. Here we consider the particular choice $A_0 \in \mathrm{St}(m,p,\mathbb{R})$ and $B_0 \in \mathrm{St}(n,p,\mathbb{R})$, as for instance $A_0 = I_{m,p}$ and $B_0 = I_{n,p}$.

Theorem 1

If the initial states of the WH2 system belong to the Stiefel manifold, then the whole dynamics is double-Stiefel.

Proof

We wish to prove that if $A_0 \in \mathrm{St}(m,p,\mathbb{R})$ and $B_0 \in \mathrm{St}(n,p,\mathbb{R})$, then $A(t) \in \mathrm{St}(m,p,\mathbb{R})$ and $B(t) \in \mathrm{St}(n,p,\mathbb{R})$ for all $t \geq 0$.

To show this for matrix A, it is sufficient to prove that the trajectory emanating from any point such that $A'A = I_p$ has differential $d(A'A) = 0$. Since $d(A'A) = (dA)'A + A'(dA) = (\dot{A}'A + A'\dot{A})dt = [\dot{A}, A]dt$, this may be proven by computing the brackets $[\dot{A}, A]$:

$$\begin{aligned}[\dot{A}, A] &= B'Z'A - A'ZBA'A + A'ZB - A'AB'Z'A \\ &= (I_p - A'A)B'Z'A + A'ZB(I_p - A'A) \\ &= [I_p - A'A, B'Z'A] = [0, B'Z'A] = 0,\end{aligned}$$

from which $d(A'A) = 0$. In a similar way it can be shown that $B_0 \in \mathrm{St}(n,p,\mathbb{R})$ implies $[dB, B] = 0$, that ensures $B(t) \in \mathrm{St}(n,p,\mathbb{R})$ for $t \geq 0$. □

The stationary points of the WH2 algorithm, when the state matrices keep within the double Stiefel manifold, may be easily characterized. In fact, we can state the following result:

Theorem 2

The steady states of WH2 learning system can be written as $A = U_pK$ and $B = V_pK$, where K is arbitrary in $O(p,\mathbb{R})$ and U_p and V_p denote the submatrices whose columns are p right and left singular vectors of the matrix Z, respectively.

Proof

From the WH2 learning equations we find that the steady states satisfy:

$$ZB = AB'Z'A \quad \text{and} \quad Z'A = BA'ZB\,. \tag{2}$$

At equilibrium, the product $A'ZB$ must be symmetric. To prove this, it is sufficient to use the first of conditions (2):

$$\begin{aligned}S \stackrel{\text{def}}{=} A'ZB &= A'(AB'Z'A) = (A'A)B'Z'A \\ &= B'Z'A = S'\,.\end{aligned}$$

Now, as A belongs to the Stiefel manifold $\mathrm{St}(m,p,\mathbb{R})$ at any time and has thus rank p, the equilibrium solution may be parameterized as $A = U_pK_a$; the same holds for $B = V_pK_b$, where K_a and K_b are matrices in $O(p,\mathbb{R})$. This ensures A and B span the SVD-subspace of Z.

On the basis of this parameterization, the product $A'ZB$ writes $K_a'(U_p'ZV_p)K_b$, where, by definition, $U_p'ZV_p = D_1$, the diagonal matrix of p singular values. On the other hand, $S = K_a'D_1K_b$ must be symmetric and this may hold only if $K_a = K_b = K$. □

This shows that the WH2 algorithm does not actually compute the true SVD, but a SVD-subspace of dimension p.

The WH4 learning system introduced in Ref. 32 reads:

$$\begin{cases}\dot{A} = ZB - \dfrac{1}{2}A(A'ZB + B'Z'A)\,, \quad A(0) = A_0\,, \\ \dot{B} = Z'A - \dfrac{1}{2}B(A'ZB + B'Z'A)\,, \quad B(0) = B_0\,,\end{cases} \tag{3}$$

which readily rewrites:

$$\begin{cases}\dot{A} = ZB - \dfrac{1}{2}A[A, ZB]\,, \quad A(0) = A_0\,, \\ \dot{B} = Z'A - \dfrac{1}{2}B[A, ZB]\,, \quad B(0) = B_0\,.\end{cases}$$

Theorem 3

Under the hypotheses $A_0 \in \mathrm{St}(m,p,\mathbb{R})$ and $B_0 \in \mathrm{St}(n,p,\mathbb{R})$ the learning equations WH4 keep $A(t)$ and $B(t)$ within the Stiefel manifold.

Proof

In order to demonstrate the claim, let us compute $[\dot{A}, A]$ and $[\dot{B}, B]$:

$$\begin{aligned}2[\dot{A}, A] &= [I_p - A'A, [A, ZB]]\,, \\ 2[\dot{B}, B] &= [I_p - B'B, [A, ZB]]\,.\end{aligned}$$

If $A'A = B'B = I_p$ then it follows from the above expression that $[dA, A] = 0$ and $[dB, B] = 0$, thus

$A(t) \in \mathrm{St}(m,p,\mathbb{R})$ and $B(t) \in \mathrm{St}(n,p,\mathbb{R})$ for any t. This proves the claim. □

The WH3 learning system was derived as an extension of well-known Oja's subspace rule.[26] The algorithm WH3 reads:

$$\begin{cases} \dot{A} = ZB - A(A'ZB + B'Z'A)\,, \quad A(0) = A_0\,, \\ \dot{B} = Z'A - B(A'ZB + B'Z'A)\,, \quad B(0) = B_0\,. \end{cases} \tag{4}$$

The dynamical properties of system WH3 follow as a trivial corollary of Theorem 3:

Corollary 1

Under the hypotheses $A_0/\sqrt{2} \in \mathrm{St}(m,p,\mathbb{R})$ and $B_0/\sqrt{2} \in \mathrm{St}(n,p,\mathbb{R})$ the learning equations WH3 keep $A(t)/\sqrt{2}$ and $B(t)/\sqrt{2}$ within the Stiefel manifold. Moreover, WH3 is diffeomorphic to WH4.

Proof

By defining auxiliary state-matrices $A_x \stackrel{\mathrm{def}}{=} \sqrt{2}A$ and $B_x \stackrel{\mathrm{def}}{=} \sqrt{2}B$, the system (4) turns out to be identical to (3). □

The structure of the stationary points of the WH3-4 algorithms is similar to the structure of the equilibria of WH2 system. This is proven in the following result:

Theorem 4

The steady states of WH3 and WH4 learning systems write $A = U_p K$ and $B = V_p K$, where K is arbitrary in $\mathrm{O}(p,\mathbb{R})$ and U_p and V_p denote the sub-matrices whose columns are p right and left singular vectors of the matrix Z, respectively.

Proof

The WH3 and WH4 learning equations may be given a unified expression in the following way:

$$\begin{cases} \dot{A} = ZB - \dfrac{1}{\nu}A(A'ZB + B'Z'A)\,, \quad A(0) = A_0\,, \\ \dot{B} = Z'A - \dfrac{1}{\nu}B(A'ZB + B'Z'A)\,, \quad B(0) = B_0 \\ A'A = B'B = \dfrac{\nu}{2}I_p\,, \end{cases} \tag{5}$$

where $\nu = 1$ for the WH3 and $\nu = 2$ for the WH4.

The steady states satisfy:

$$\begin{aligned} \nu ZB &= AA'ZB + AB'Z'A \\ \text{and} \qquad \nu Z'A &= BA'ZB + BB'Z'A\,. \end{aligned} \tag{6}$$

At equilibrium, the product $S \stackrel{\mathrm{def}}{=} A'ZB$ must be symmetric. To prove this, it suffices to use the first of conditions (6):

$$\begin{aligned} S = A'(ZB) &= \frac{1}{\nu}(A'A)(A'ZB + B'Z'A) \\ &= \frac{1}{2}(S + S')\,. \end{aligned}$$

This shows that $2S = S + S'$ and thus that $S = S'$.

The conclusion now follows from the same argument of Theorem 2. □

2.2. *The HM neural SVD dynamical system*

The HM dynamics arises from the maximization of a specific metric-criterion $\Phi_W : O(m,\mathbb{C}) \times O(n,\mathbb{C}) \to \mathbb{R}$ defined as:

$$\Phi_W(A,B) \stackrel{\mathrm{def}}{=} 2\ \mathrm{Re\ tr}(WA^*ZB)\,, \tag{7}$$

where $W \in \mathbb{R}^{n\times m}$ is a weighting matrix and $Z \in \mathbb{C}^{m\times n}$ is the matrix whose (complex-valued) SVD is looked for, in the hypothesis that $m \geq n$. The dynamical system, derived as a Riemannian gradient flow (see appendix A.2) on $O(m,\mathbb{C}) \times O(n,\mathbb{C})$, reads:

$$\begin{cases} \dot{A} = A(W^*B^*Z^*A - A^*ZBW)\,, \quad A(0) = A_0\,, \\ \dot{B} = B(WA^*ZB - B^*Z^*AW^*)\,, \quad B(0) = B_0\,. \end{cases} \tag{8}$$

By construction it holds $A(t) \in O(m,\mathbb{C})$ as well as $B(t) \in O(n,\mathbb{C})$.

In the particular case that $W = -I_{n,m}$ and the involved quantities are real-valued, system (8) recasts into:

$$\begin{cases} \dot{A} = ZBI_{n,m} - AI_{m,n}B'Z'A\,, \\ \dot{B} = Z'AI_{m,n} - B(AI_{m,n})'ZB\,. \end{cases} \tag{9}$$

Such simplified system is equivalent to WH2 when $p = n$; in order to prove such statement, it is first worth noting from the second equation of the system (9) that the last $m - n$ columns of A do not influence the dynamics of B; thus, it is

worth defining the reduced-size matrix $A_n \stackrel{\text{def}}{=} AI_{m,n}$ and to note that $\dot{A}_n = \dot{A}I_{m,n} = ZBI_{n,m}I_{m,n} - (AI_{m,n})B'Z'(AI_{m,n})$; thanks to the hypothesis $m \geq n$ it is directly verified that $I_{n,m}I_{m,n} = I_n$, therefore the system (9) recasts into:

$$\begin{cases} \dot{A}_n = ZB - A_nB'Z'A_n\,, \\ \dot{B} = Z'A_n - BA_n'ZB\,, \end{cases}$$

whereby the equivalence with the algorithm WH2 when $p = n$.

The above analysis shows that the Weingessel–Hornik SVD learning equations may be regarded as special cases of Helmke–Moore system; in particular, this is an indirect proof that the choice of the elements of the weighting kernel W as a pseudo-identity makes the SVD algorithm a SVD-subspace rule. A consequence of these findings is that the properties of the above mentioned learning rules may be given a unified investigation, that is the subject of the following section.

As a useful side-note, it is worth citing the opportunity to modify the HM system (8) when the ratio m/n is much larger than 1: In this case, from a numerical point of view, it is convenient to compute the *thin* SVD of a matrix Z instead of the regular SVD.[15] In the hypothesis that $Z \in \mathbb{C}^{m\times n}$ and n coincides with the rank of Z, the thin SVD of Z is defined as the triple (U_n, D_n, V) such that $U_n \in \mathrm{St}(m,n,\mathbb{C})$, $D_n \in \mathbb{R}^{n\times n}$ diagonal, $V \in O(n,\mathbb{C})$ and $Z = U_nD_nV^*$.

In this case, the HM system (8) easily recasts into a more compact form as:

$$\begin{cases} H = A^*ZB \in \mathbb{C}^{n\times n}, \\ \dot{A} = A(W^*H - HW), \quad A(0) = A_0 \in \mathrm{St}(m,n,\mathbb{C}), \\ \dot{B} = B(WH - H^*W^*), \quad B(0) = B_0 \in O(n,\mathbb{C}), \end{cases} \tag{10}$$

with $W \in \mathbb{R}^{n\times n}$ diagonal. In this case, the neural-network state-matrix A evolves on the Stiefel manifold of dimension $m \times n$, thus its numerical representation is more advantageous under the considered hypothesis $n \ll m$.

3. Theoretical Considerations

This section is dedicated to the statement and proof of some theoretical results about the behavior of Weingessel–Hornik and Helmke–Moore learning systems.

3.1. *Derivation of WH equations*

As mentioned, the WH2 equations arise as a special case of HM equations, therefore the derivation of WH2 equations are implicitly considered in the Sec. 3.2.

Weingessel and Hornik derived the WH3 learning rule from the Oja's principal subspace equation,[26] that, for a $(m+n)\times p$ network with connection matrix M, writes:

$$\dot{M} = (I_{m+n} - MM')CM\,, \tag{11}$$

where C is a $(m+n)\times(m+n)$ covariance matrix. By relating the $(m \times n)$ covariance Z that a SVD subspace is sought for with C and by effecting a proper block-decomposition of state-matrix M into A and B, the WH3 learning rule is easily obtained from Oja's subspace rule, as shown in the following result.

Theorem 5

(Ref. 33.) Let us define $C = \begin{bmatrix} 0_m & Z \\ Z' & 0_n \end{bmatrix}$ and $M = \begin{bmatrix} A \\ B \end{bmatrix}$. Then Eq. (11) is equivalent to system (4).

Proof

By replacing the expressions for C and M into Eq. (11) we obtain:

$$\begin{aligned} \begin{bmatrix} \dot{A} \\ \dot{B} \end{bmatrix} &= \left(\begin{bmatrix} I_m & 0_{m,n} \\ 0_{n,m} & I_n \end{bmatrix} - \begin{bmatrix} A \\ B \end{bmatrix} \begin{bmatrix} A' & B' \end{bmatrix} \right) \begin{bmatrix} 0_m & Z \\ Z' & 0_n \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \\ &= \begin{bmatrix} I_m - AA' & -AB' \\ -BA' & I_n - BB' \end{bmatrix} \begin{bmatrix} ZB \\ Z'A \end{bmatrix} \\ &= \begin{bmatrix} (I_m - AA')ZB - AB'Z'A \\ -BA'ZB + (I_n - BB')Z'A \end{bmatrix}\,. \end{aligned} \tag{12}$$

By separating the two differential equations and by properly regrouping the terms on the right-hand sides, the WH3 learning system reported in this paper is readily obtained. □

It is interesting to observe that the WH3 system inherits the known and noticeable properties of Oja's subspace equations, such as the Riccati structure of the projector MM'. Namely, by defining:

$$P \stackrel{\text{def}}{=} \begin{bmatrix} AA' & AB' \\ BA' & BB' \end{bmatrix}, \tag{13}$$

it is easy to show that P satisfies the differential equation:

$$\dot{P} = CP + PC - 2PCP\,, \tag{14}$$

that is a special kind of Riccati differential equation.[29]

Another interesting observation is that Oja's criterion, that leads to the associated subspace rule, induces a criterion on the pair (A, B) that is a special case of the HM criterion. To show this implication, it is worth recalling the following:

Lemma 1

(Ref. 26.) Oja's subspace rule (11) arises from the optimization of the criterion $\mathrm{tr}[M'CM]$ under the constraint $M \in \mathrm{St}(m+n, p, \mathbb{R})$.

Having recalled this basic fact, we can state the mentioned equivalence result:

Theorem 6

Oja's criterion for the block-pair (A, B) is identical to HM criterion for the real-valued case when $W = -I_{n,m}$.

Proof

By invoking again the block-partition of Theorem 5, we have:

$$\begin{aligned} \mathrm{tr}(M'CM) &= \mathrm{tr}\left([A' \quad B'] \begin{bmatrix} 0_m & Z \\ Z' & 0_n \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right) \\ &= \mathrm{tr}(A'ZB + B'Z'A)\,. \end{aligned} \tag{15}$$

It follows that $\mathrm{tr}(M'CM) = 2\,\mathrm{tr}(A'ZB)$, which proves the claim. □

3.2. *Derivation of HM equations on Stiefel manifold with Killing metric*

The rationale of the Helmke–Moore criterion Φ derives from the basic observation that the aim of SVD is to diagonalize $A'ZB$, that in a signal processing perspective means minimizing the values of covariance among the signals that Z is the cross-covariance matrix of. Fixed thus an arbitrary diagonal matrix H_0, this result may be achieved by minimizing, under proper constraints, the "non-diagonality" measure $\|A'ZB - H_0\|_{\mathrm{F}}^2$. However, the following identity holds:[18]

$$\begin{aligned} \|A'ZB - H_0\|_{\mathrm{F}}^2 = \mathrm{tr}(ZZ') + \|H_0\|_{\mathrm{F}}^2 \\ - \Phi_{H_0}(A, B)\,, \end{aligned}$$

thus minimizing the non-diagonality measure is equivalent to maximizing the function (7). It is understood that the optimization process should be performed over $O(m, \mathbb{R}) \times O(n, \mathbb{R})$. In the real-valued case under consideration here, also the well-known (weighted) Rayleigh quotient (RQ) may be invoked, which is defined as:

$$R_W(A, B) \stackrel{\text{def}}{=} \frac{\mathrm{tr}(A'ZBW)}{\|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}}}\,. \tag{16}$$

As long as A and B belong to the orthogonal group, that implies $\|A\|_{\mathrm{F}}^2 = m$ and $\|B\|_{\mathrm{F}}^2 = n$, the identity $2\sqrt{mn}R_W(A, B) = \Phi_W(A, B)$ holds.

In any case, the quantity $\mathrm{tr}(A'ZBW)$ is a starting point for developing a suitable SVD learning theory, generating HM-type and WH2-type differential equation systems.

In order to derive the Riemannian-gradient flows HM or WH2, both in Refs. 32 and 18 the technique proposed by Brockett[3] was used, which involves the first-order expansion of the dynamics of $A(t)$ and $B(t)$ in series of skew-symmetric matrices. Here we aim at re-deriving the HM equations (for the real-valued orthogonal group) in a different way, following the more straightforward Riemannian-gradient approach suggested by Amari,[1,4] based on the geometry of the Stiefel manifolds.

Theorem 7

Let us define $H \stackrel{\text{def}}{=} A'ZB$. The gradient-based maximization of the objective function $\Phi_r(H) \stackrel{\text{def}}{=}$

$\mathrm{tr}(WH)$, where $A \in O(m,\mathbb{R})$, $B \in O(n,\mathbb{R})$, $W \in \mathbb{R}^{n\times m}$ and $Z \in \mathbb{R}^{m\times n}$, with spaces endowed with the Killing metric, gives rise to the following dynamical system:

$$\begin{cases} \dot{A} = -ZBW + AW'B'Z'A\,, \\ \dot{B} = -Z'AW' + BWA'ZB\,. \end{cases} \tag{17}$$

Proof

The learning criterion function is $\Phi_r = \mathrm{tr}(WA'ZB)$. A perturbation (dA, dB) of neural network state (A, B) causes a change $d\Phi_r$. In particular, up to first order:

$$\begin{aligned} \Phi_r(H+dH) &= \mathrm{tr}(W(A+dA)'Z(B+dB)) \\ &= \Phi_r(H) + \mathrm{tr}(WdA'ZB) \\ &\quad + \mathrm{tr}(WA'ZdB)\,, \end{aligned}$$

therefore, by exploiting the properties of the elements of the orthogonal groups and of trace operator, we have:

$$\begin{aligned} d\Phi_r(H) &= \mathrm{tr}(WdA'(AA')ZB) + \mathrm{tr}(WA'Z(BB')dB) \\ &= \mathrm{tr}(WdA'AH) + \mathrm{tr}(WHB'dB) \\ &= -\mathrm{tr}(HWA'dA) + \mathrm{tr}(WHB'dB)\,. \end{aligned} \tag{18}$$

It is now useful to introduce the differentials $dX \stackrel{\mathrm{def}}{=} A'dA$ and $dY \stackrel{\mathrm{def}}{=} B'dB$, which form a basis of the tangent space to $O(m,\mathbb{R})$ at A and to $O(n,\mathbb{R})$ at B. The tangent spaces are linear spaces and, moreover, they are the sets of proper-size skew-symmetric matrices; in fact, we have $dX' = -dX$ and $dY' = -dY$. In view of optimization, these properties must be preserved. A way to preserve the structure of the tangent space is to note that:

$$\begin{aligned} d\Phi_r(H) &= -\mathrm{tr}(HWdX) + \mathrm{tr}(WHdY) \\ &= -\mathrm{tr}(dX'W'H') + \mathrm{tr}(dY'H'W') \\ &= \mathrm{tr}(dXW'H') - \mathrm{tr}(dYH'W') \\ &= \mathrm{tr}(W'H'dX) - \mathrm{tr}(H'W'dY)\,. \end{aligned} \tag{19}$$

By summing hand-by-hand Eqs. (18) and (19) we ultimately obtain:

$$2d\Phi_r(H) = \mathrm{tr}(\{W,H'\}dX) + \mathrm{tr}(\{W',H\}dY)\,. \tag{20}$$

As our aim is to find directions dA and dB that point toward the maximum of function Φ_r, we now search for the steepest ascent directions ΔX and ΔY, that, by definition, are the variations which maximize the change $\Delta\Phi_r$ under finite-step-length constraints, namely $\|\Delta X\|^2 = \varepsilon_x^2 > 0$ and $\|\Delta Y\|^2 = \varepsilon_y^2 > 0$. To this aim, we need to specify the norm $\|\cdot\|$; in this case we use the standard Euclidean metric on the tangent space, that is the Killing metric (see appendix A.3), with which the constraints rewrite $\mathrm{tr}(\Delta X'\Delta X) = \varepsilon_x^2$ and $\mathrm{tr}(\Delta Y'\Delta Y) = \varepsilon_y^2$. In order to enforce the mentioned constraints, the standard Lagrange multipliers method may be employed, that consists in the definition of the Lagrangean function:

$$\begin{aligned} \mathcal{L} \stackrel{\mathrm{def}}{=} &\mathrm{tr}(\{W,H'\}\Delta X) + \mathrm{tr}(\{W',H\}\Delta Y) \\ &+ \lambda_x(\mathrm{tr}(\Delta X'\Delta X) - \varepsilon_x^2) \\ &+ \lambda_y(\mathrm{tr}(\Delta Y'\Delta Y) - \varepsilon_y^2)\,, \end{aligned}$$

whose free extremes may be looked for. They find by:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\Delta\mathcal{X}} &= \{W,H'\} + 2\lambda_x\Delta X = 0\,, \\ \frac{\partial\mathcal{L}}{\partial\Delta\mathcal{Y}} &= \{W',H\} + 2\lambda_y\Delta Y = 0\,. \end{aligned}$$

Therefore the steepest ascent variations express as $dX \propto \{W,H'\}$ and $dY \propto \{W',H\}$. Coming back to the original variables in the orthogonal groups we have $\dot{A} = A\{W,H'\}$ and $\dot{B} = B\{W',H\}$, which proves the claim. □

The shown learning system coincides to HM system in the real-valued case and also explains the WH2 learning theory.

3.3. *Stability analysis of HM equation on orthogonal group via Lyapunov function*

One of the main theoretical advantages of the learning systems on Stiefel manifolds is their inherent stability,[9] due to the compactness of these sets. In the present case, the convergence of the HM system may be proven by showing that $\Phi_W(A,B)$ is a Lyapunov-type function for the HM system.

More formally, let us denote by $\Phi_{\min}$ the minimal value of the function in $O(m,\mathbb{R})\times O(n,\mathbb{R})$; note that it exists since Φ_W is a continuous function defined on a compact manifold.

Theorem 8

Let us define the (lifted-criterion) time-function:

$$\Psi(t) \stackrel{\text{def}}{=} \text{tr}(A'(t)ZB(t)W) - \Phi_{\min}\,. \qquad (21)$$

It is a Lyapunov function for the system (17).

Proof

By construction $\Psi(t) \geq 0$. Also, by letting $H(t) \stackrel{\text{def}}{=} A'(t)ZB(t)$ and by using Eqs. (17) it is found:

$$\begin{aligned}\dot{\Psi} &= \text{tr}(\dot{A}'ZBW + A'Z\dot{B}W) \\ &= -\text{tr}(W'H'HW + WHH'W' \\ &\quad - 2HWHW)\,. \end{aligned} \qquad (22)$$

Some mathematical work shows that the following identities hold true:

$$\begin{aligned} -\|\{W, H'\}\|_{\text{F}}^2 &= \text{tr}(-2W'H'WH + 2HWHW)\,, \\ -\|\{W', H\}\|_{\text{F}}^2 &= \text{tr}(2WHH'W' - 2WHWH)\,. \end{aligned}$$

In virtue of the these results, we may rewrite the right-hand side of expression (22) as follows:

$$\begin{aligned}\dot{\Psi}(t) = -\frac{1}{2}(&\|\{W, H'(t)\}\|_{\text{F}}^2 \\ &+ \|\{W', H(t)\}\|_{\text{F}}^2) \leq 0\,. \end{aligned} \qquad (23)$$

Such inequality proves the claim. □

The structure of the steady-state solutions of the HM system has been studied in details by Helmke and Moore and has been reported in Ref. 18. The convergence property of the HM system towards such steady-states now follows immediately.

Corollary 2

The HM learning system (17) converges asymptotically.

Proof

The existence of a Lyapunov function for the dynamical system under analysis, proven by Theorem 8, guarantees the asymptotic stability of its equilibria.[16] □

4. Numerical Experiments

Some numerical experiments are described and commented in the following sections. They help assessing the qualitative behavior of the discussed learning equations.

It is worth noting that the discrete-time implementations used to numerically solve the ODEs associated to the learning systems introduce some deviations with respect to the theoretical findings: Whereas the continuous-time versions of the learning algorithms leave the double Stiefel-manifold and orthogonal-group invariant, this is not necessarily true for their discrete-time counterparts. These aspects deserve a separated treatment and are addressed in the last section.

4.1. *Numerical experiments on SVD-subspace extraction by WH2-3-4 algorithms*

We performed some experiments with the Weingessel–Hornik algorithms, in order to numerically evaluate their behavior.

Let us denote again with Z the $m \times n$ matrix whose SVD-subspace of dimension p is looked for and let us denote with (U, D, V) the matrices of left-singular vectors, singular values and right-singular vectors. The extraction of a p-dimensional SVD subspace implies that the columns of matrices A and B in the WH algorithms should span, after convergence, the same subspace spanned by the first p columns of U and V, respectively, in the hypothesis that the singular values are decreasingly ordered. Let us denote by U_p and V_p the sub-matrices of U and V containing their first p columns: A proper measure of the SVD-subspace extraction ability of the network is the SVD-subspace disparity error pair, defined in Ref. 32 as:

$$\begin{aligned} \varepsilon(A) &\stackrel{\text{def}}{=} \frac{\|U_pU_p'A - A\|_2}{\|A\|_2}\,, \\ \varepsilon(B) &\stackrel{\text{def}}{=} \frac{\|V_pV_p'B - B\|_2}{\|B\|_2}\,. \end{aligned} \qquad (24)$$

As a numeric problem we considered the case $m = 8$, $n = 6$ and $p = 3$. The numerical results have been obtained by randomly picking a matrix Z with normal Gaussian entries and by computing the disparity errors $\varepsilon(A)$ and $\varepsilon(B)$ at each iteration, for a total of $8,000$ iterations. This experiment is repeated on 100 independent trials: The average learning curves are presented. Also, if we denote with

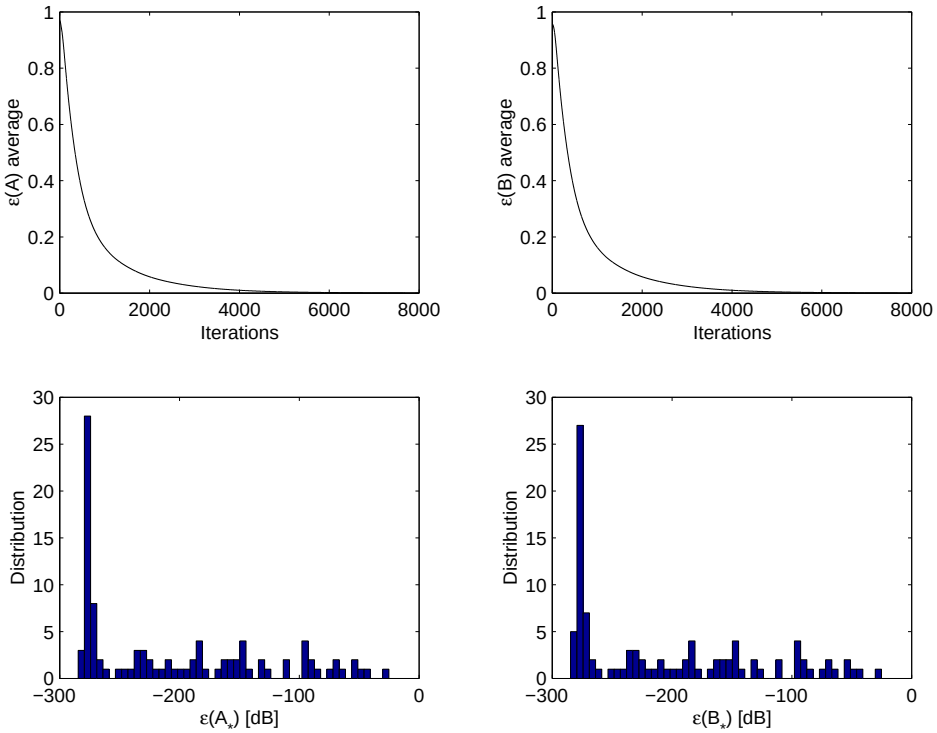

Fig. 1. Values of average disparity errors and estimation of disparity errors distribution after learning for the WH2 algorithm.

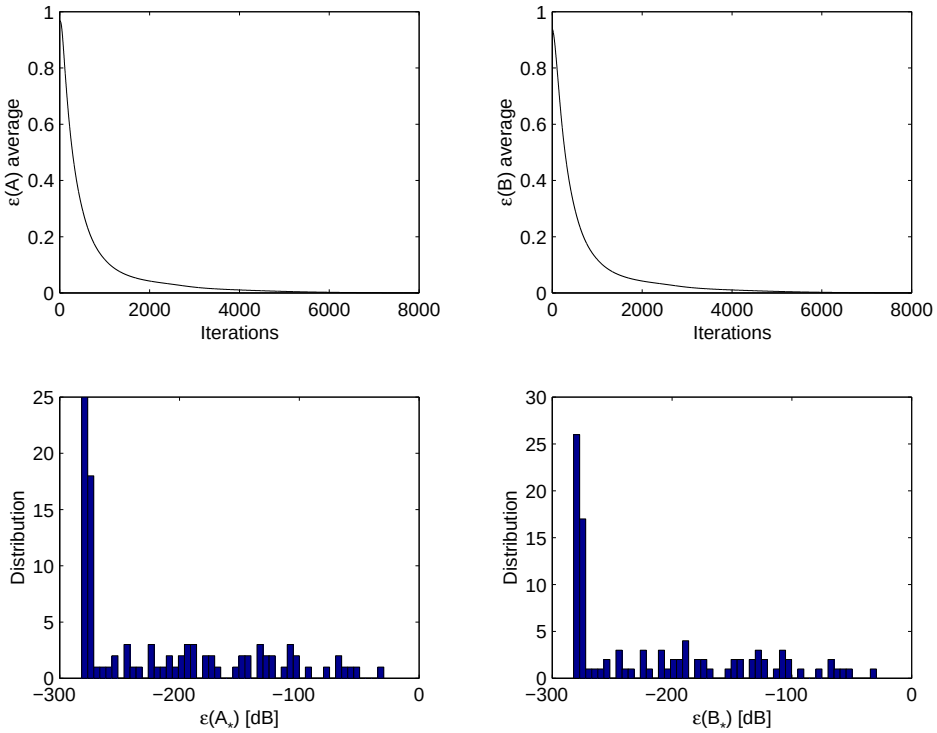

Fig. 2. Values of average disparity errors and estimation of disparity errors after learning for the WH3 algorithm.

$\varepsilon(A_\star)$ and $\varepsilon(B_\star)$ the final values of the errors at the end of the iterations (which measure the learnt-network performance), the statistical distributions of these quantities over the 100 trials is estimated.

For the WH2 algorithm the initial values of the connection-matrices were $A_0 = I_{m,p}$ and $B_0 = I_{n,p}$ and the value of the learning stepsize was $\eta = 0.005$. The results of the numerical experiments are illustrated in the Fig. 1. The average behavior of the algorithm is very good. No instabilities were observed and in the largest part of trials the value of the error at the end of learning is very low: The 97% of the trials gave an error less than -50 dB both for the U_p-subspace and the V_p-subspace. We also checked the orthonormality of the solutions $A_\star$ and $B_\star$ and found that they are orthonormal, with respect to the measures $\|A_\star' A_\star - I_p\|_{\mathrm{F}}$ and $\|B_\star' B_\star - I_p\|_{\mathrm{F}}$, up to order 10^{-15}.

For the WH3 algorithm the initial values of the connection-matrices were $A_0 = I_{m,p}/\sqrt{2}$ and $B_0 = I_{n,p}/\sqrt{2}$ and the value of the learning stepsize was $\eta = 0.005$. The results of the numerical experiments are shown in the Fig. 2. Again, the average behavior of the algorithm is very good and no instabilities were observed. It is interesting to inspect the numerical results for a single-case trial. We consider the random matrix (only four decimal digits):

$$Z = \begin{bmatrix} -1.4283 & -0.8074 & 1.1283 & 1.0507 & 0.6424 & 2.1022 \\ -1.6218 & 0.5095 & -0.1493 & 0.3916 & 0.8243 & 0.5004 \\ 1.4791 & -0.1503 & 0.3535 & 0.7533 & -1.6819 & 1.1130 \\ -0.9191 & 0.3345 & -0.3409 & 2.8772 & -0.5035 & -1.3955 \\ 0.4135 & -1.4989 & -0.0959 & 0.1144 & -0.1325 & -0.6744 \\ -0.4591 & 0.4383 & -2.0674 & 1.0988 & 0.4373 & -1.1486 \\ -1.4654 & 0.9846 & -0.1393 & -2.9265 & -0.3399 & 0.8313 \\ -0.5261 & 0.2351 & -0.5403 & 0.1352 & 1.1526 & -0.4636 \end{bmatrix};$$

for this covariance matrix the algorithm has computed the following left singular vectors (only four decimal digits):

$$A_\star = \begin{bmatrix} 0.4385 & 0.4101 & 0.1641 \\ 0.0668 & 0.3479 & -0.1297 \\ 0.2208 & -0.2714 & 0.2575 \\ 0.2818 & -0.0320 & -0.3900 \\ 0.0350 & -0.1621 & -0.0019 \\ -0.0955 & -0.0248 & -0.4195 \\ -0.3997 & 0.3052 & 0.1590 \\ -0.0695 & 0.1261 & -0.1906 \end{bmatrix}$$

and right singular vectors (only four decimal digits):

$$B_\star = \begin{bmatrix} -0.0196 & -0.5579 & 0.2052 \\ -0.1994 & 0.1046 & -0.1263 \\ 0.2354 & 0.0735 & 0.3370 \\ 0.5947 & -0.0994 & -0.3394 \\ -0.0388 & 0.2870 & -0.1789 \\ 0.2219 & 0.2832 & 0.4256 \end{bmatrix}.$$

In this experiment the subspace disparity errors were about -270 dB.

For the WH4 algorithm the initial values of the connection-matrices were $A_0 = I_{m,p}$ and $B_0 = I_{n,p}$ and the value of the learning stepsize was $\eta = 0.006$. The results of the numerical experiments are illustrated in the Fig. 2. The results are as expected. We again checked the orthonormality of the solutions $A_\star$ and $B_\star$ and found that they are orthonormal up to order 10^{-15}. We also checked for the "symmetrization" property of the matrix-product $A'ZB$ and found that, at the end of learning, it was (only four decimal digits):

$$A_\star' Z B_\star = \begin{bmatrix} 3.1045 & 0.3523 & -0.0450 \\ 0.3523 & 3.2290 & -0.0278 \\ -0.0450 & -0.0278 & 3.0668 \end{bmatrix}.$$

This result comes from a single trial: As expected from the theory (see Theorem 4), the form $A'ZB$ is symmetrical but not diagonal, confirming the analytical finding that the pair (A, B) is just a base of the SVD subspace, not a proper singular-value decomposition.

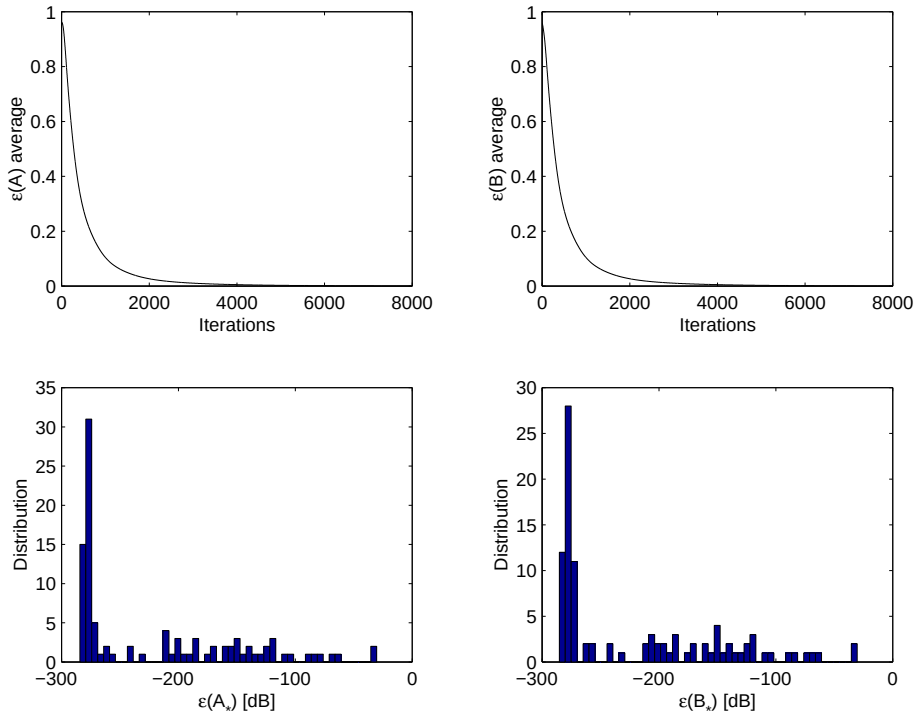

Fig. 3. Values of average disparity errors and estimation of disparity errors after learning for the WH4 algorithm.

4.2. *Numerical experiments on SVD extraction by HM algorithm*

We performed some experiments with the Helmke–Moore algorithm, in order to evaluate numerically its behavior.

Again Z denotes the $m \times n$ matrix whose SVD is sought for and (U, D, V) denote the matrices of left-singular vectors, singular values and right-singular vectors. As in the theoretical sections, we consider $m \geq n$. As indicators of the behavior of the algorithms, we consider the following measures.

First, if A_n, B_n, U_n and V_n denote the submatrices formed by the first n columns of the SVD and network matrices, it is known that the columns of A_n should tend to the columns of U_n, while the columns of B_n should tend to the columns of B_n, ordered in the same way but with a possible sign switch for every column; therefore, a proper measure of (A, B) convergence is:

$$\begin{aligned} \varepsilon(A_p) &\stackrel{\text{def}}{=} \| |U_p| - |A_p| \|_{\text{F}} \,, \\ \varepsilon(B_p) &\stackrel{\text{def}}{=} \| |V_p| - |B_p| \|_{\text{F}} \,, \end{aligned} \tag{25}$$

where $|X|$ stands for component-wise absolute-value extraction.

Second, it is interesting to inspect the value of the criterion function $\Phi(A, B) = 2\,\text{tr}(WA'ZB)$ during learning and to compare its asymptotic value with the optimum $\Phi_\star = 2\,\text{tr}(WU'ZB)$.

Third, we known that the HM learning principle is defined in order to diagonalize the product-matrix $A'ZB$, therefore an interesting error measure is the norm of the off-diagonal part of that matrix; namely we may define a corresponding index as:

$$\delta(A, B) \stackrel{\text{def}}{=} \|\text{offdiag}(WA'ZB)\|_{\text{F}} \,, \tag{26}$$

with clear meaning of the symbols.

Fourth, it is also extremely interesting to measure the deviation from orthonormality of the connection matrices. This may be achieved by the help of the indices:

$$\begin{aligned} n(A) &\stackrel{\text{def}}{=} \|A'A - I_m\|_{\text{F}} \,, \\ n(B) &\stackrel{\text{def}}{=} \|B'B - I_n\|_{\text{F}} \,. \end{aligned} \tag{27}$$

As a numeric problem, we considered the case $m = 8$, $n = 3$. The numerical results have been obtained by randomly generating a matrix Z by computing the above-described indices at each iteration,

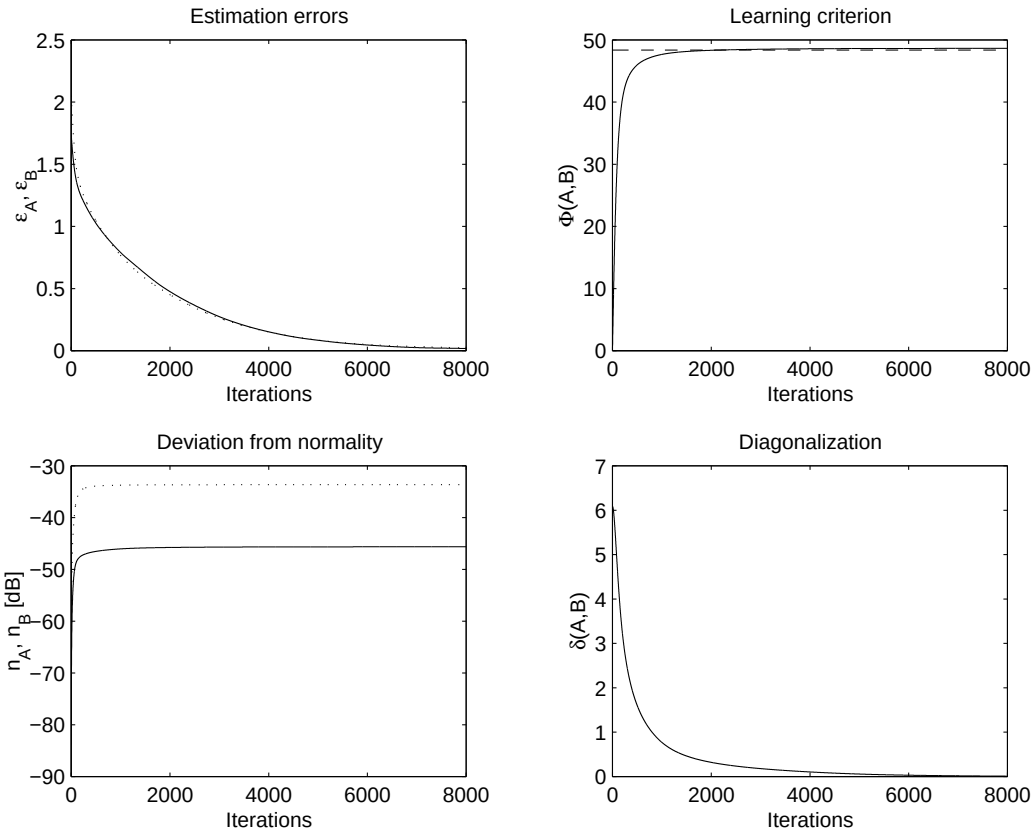

Fig. 4. Values of performance indices for the HM algorithm averaged over 100 independent trials. (Solid: Index of matrix A; Dashed: Index of matrix B).

for a total of $8,000$ iterations. This experiment was repeated on 100 independent trials in order to show average learning curves. It is important to note that, in order to compare the values of the function Φ over different trials, both the initial states and the singular values of Z must keep constant. So we first generated randomly a diagonal matrix D with n non-null entries on the diagonal that keep constant over the whole trial set, and then for each trial a pair of orthogonal matrices (U, V) of proper size were randomly generated; then Z computes as UDV'.

The results of the experiments are illustrated in the Fig. 4. They pertain to the following parameter-values: $\eta = -0.001$, $A_0 = I_m$ and $B_0 = I_n$; also, the top-left diagonal part of the weighting matrix W is $\mathrm{diag}(3, 2, 1)$. As it is readily seen, the numerical results are very good.

As a single-trial result, it is interesting to inspect the singular-value estimation ability of the algorithm: In an experiment it was $\sigma_1 = 5.2506$, $\sigma_2 = 3.5342$ and $\sigma_3 = 1.6557$, while the diagonal part of $A'ZB$ is $\mathrm{diag}(5.2941, 3.5446, 1.6587)$; this shows that the estimation ability of the HM algorithm is quite good.

The last experiment concerns numerical analysis of stability: In this case, the matrix Z suddenly changes in the middle of iteration (but for the singular values which keep constant). The results of 100 independent trials are illustrated in the Fig. 5. They pertain to the same set of parameter-values of the preceding experiment. When the matrix Z changes the performance indices present a peak but they return rapidly to the satisfactory asymptotic values.

4.3. *Discussion on learning equations implementation*

In this section we try to get the numerical simulation results into the right picture by expanding briefly the longer discussion on this topic already appeared in the recent contributions.[12,13]

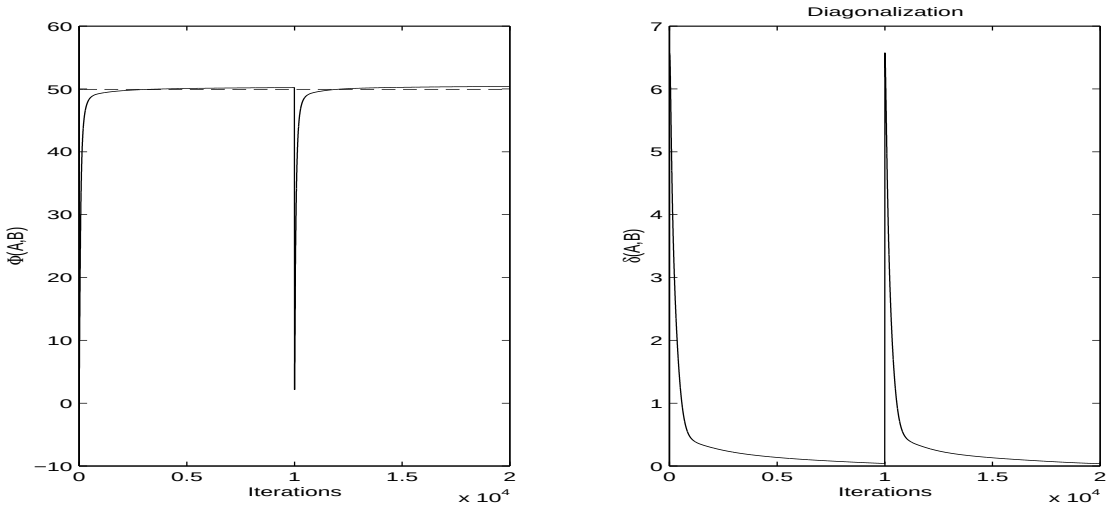

Fig. 5. Values of performance indices for the HM algorithm averaged over 100 independent trials when the right and left singular vectors change during iteration. (Solid: Index of matrix A; Dashed: Index of matrix B).

In general, there are several possibilities of implementing a learning algorithm. In this paper, the results are obtained for the continuous-time case but the simulations are performed for the off-line discrete-time version, related to using a simple Euler approximation for solving the associate ODEs. This opens room to at least one question: How does the used integration algorithm affect the relevance of the obtained numerical results?

We believe a simple yet convincing answer comes from the consideration about the integration time: We classify a learning process into short-integration-time learning, that requires few adaptation steps to get a satisfactory connection pattern, and long-integration-time learning, that involves the solution of the differential system for a long time-interval to obtain a satisfactory result or to tackle a non-stationary signal processing problem, for instance.

The quality of the solution of the continuous-time learning equations may be heavily affected by the selected integration scheme only in long-integration-time learning processes: In this case it is not normally possible to select small learning stepsizes because this would cause an excessive computational burden, thus the learning equations are sampled with relatively high step-sizes; in this case, however, the Euler method may fail in finding accurate state-space trajectories or in fulfilling the constrains (such as orthonormality, in the present case) and more complicated integration techniques should be used, such as the ones relying on the Lie–Euler method or on second-order methods such as the Lie–Runge–Kutta technique.[12]

In the present case, we clearly dealt with short-integration-time learning processess that do not need such complicated integration schemes to be adopted, as the small values of the chosen learning stepsizes ensure good convergence in a reasonably small number of iterations. This claim is confirmed e.g., by the values of the orthogonality measure reported for every simulation which shows that the degree of adherence to the invariant is excellent (up to order 10^{-15}). This consideration suggests that the results of the simulations for the discrete-time case are a legitimate mean for illustrating the theoretical results obtained for the ODEs.

5. Conclusions

The aim of this paper was to present a unifying view of closely-related parallel SVD-computation algorithms by neural networks learning on double Stiefel manifold. After showing a new derivation of HM theory based on Riemannian-gradient on double-orthogonal group endowed with the Killing metric, some important properties of the algorithms have been investigated. Particularly, a suitable Lyapunov stability criterion has been constructed to prove asymptotic convergence.

Some numerical experiments also helped assessing the qualitative behavior of the discussed learning equations.

Acknowledgments

I am definitely indebted with the anonymous Reviewer whose careful comments and detailed suggestions greatly helped to improve the quality and the clarity of the manuscript.

A. Appendix

The aim of the present appendix is to recall some definitions from differential geometry and to justify some notational choices within the paper.

A.1. *Double Stiefel-manifold and orthogonal-group*

Let us recall the definition of group and argument on the group-structure of the Cartesian product of two orthogonal groups.

Definition 1

(Ref. 19.) A *group* is a structure $(G, \diamond)$ formed by a set G and an operation $\diamond$ that associates to every pair $(a, b) \in G \times G$ a unique element $a \diamond b$. The operation satisfies three axioms: (1-Associativity) $(a \diamond b) \diamond c = a \diamond (b \diamond c)$, (2-Existence of a neutral element) there exists some $e \in G$ such that $e \diamond a = a$ for all $a \in G$ and (3-Existence of the inverse) for all $a \in G$ there exists an element $a^{-1} \in G$ such that $a^{-1} \diamond a = e$.

Within the paper the matrix-set $O(n, \mathbb{K})$ has been invoked frequently. It is easy to show that it is a group. In fact, by identifying the operation $\diamond$ with the standard matrix product, it is easily verified that if $a, b, c \in O(n, \mathbb{K})$ then $(ab)c = a(bc)$, $e = I_n$ and there exists an element a^{-1} such that $a^{-1}a = I_n$.

Let us now consider the Cartesian product $G = O(n, \mathbb{K}) \times O(m, \mathbb{K})$ and let us show that it is actually a group (of dimension $m+n$). It is worth considering the representation $a = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$, with $A_1^* A_1 = I_n$ and $A_2^* A_2 = I_m$ and again the identification of $\diamond$ with the matrix product. Then, the associativity property holds, it is readily verified that $a^* a = I_{n+m}$ and $a^{-1} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix}$. This proves the claim.

Let us also recall the definition of smooth manifold and argument on the manifold-structure of the Cartesian product of two of such geometric entities.

Definition 2

(Ref. 19.) Let M be a topological space. A chart of M is a triple (U, ϕ, n) consisting of an open subset U of M, an homeomorphism ϕ of U onto $\mathbb{R}^n$ and the dimension n of the chart.

Two charts (U, ϕ, n) and (V, ψ, m) of M are termed C^∞ compatible charts if either $U \cap V$ is empty or $\phi(U \cap V)$ and $\psi(U \cap V)$ are empty sets and $\psi \circ \phi^{-1}$ as well as $\phi \circ \psi^{-1}$ are C^∞ maps.

A C^∞ atlas of M is a set $\mathcal{A} = \{(U_i, \phi_i, n_i) | i \in \mathcal{I}\}$ of C^∞ compatible charts such that M is completely covered by the union of the U_i's. An atlas $\mathcal{A}$ is maximal if every chart of M which is C^∞ compatible with every chart of $\mathcal{A}$ also belongs to $\mathcal{A}$.

A *smooth manifold* M is a topological Hausdorff space endowed with a countable basis and equipped with a maximal C^∞ atlas. If all the coordinate charts of M have the same dimension n then the manifold is said to have dimension n.

On the basis of the above-recalled definitions, it is possible to show that the Cartesian product of two smooth manifolds is a manifold itself. The argument follows from the observation that if M and N are two smooth manifold, then any two charts (U, ϕ, n) and (V, ψ, m) of M and N define a chart $(U \times V, \phi \times \psi, n + m)$ of $M \times N$. Therefore, $M \times N$ has the structure of a smooth manifold of dimension $m + n$ [19]. The compact Stiefel manifold is a smooth manifold, thus the product space $\mathrm{St}(m, n, \mathbb{K}) \times \mathrm{St}(p, q, \mathbb{K})$ is a manifold itself.

A.2. *Riemannian gradient flow*

As mentioned, the HM system may be derived as a gradient-flow on a Riemannian manifold. In order to clarify the relationship with the presented theory, it is useful to recall the definition of Riemannian gradient flow.

Definition 3

(Ref. 19.) Let M be a smooth manifold. A Riemannian metric on M is a family of non-degenerate

inner products, which are functions of the point on the manifold, defined on the tangent space to each point on the manifold, such that it depends smoothly on the point. When a Riemannian metric is specified, M is termed Riemannian manifold.

Let Φ: $M \to \mathbb{R}$ be a smooth function defined on a Riemannian manifold M. The gradient vector field $\operatorname{grad}\Phi$ of the function with respect to the selected metric is uniquely characterized by two conditions, referred to as tangency and compatibility.

On the basis of these definitions, it is possible to recall the concept of Riemannian gradient flow on a Riemannian manifold M as $\dot{x}(t) = \operatorname{grad}\Phi(x(t))$, with $x(t) \in M$.

With reference to the HM algorithm, it is worth noting that it has been studied over the double orthogonal group. However, the orthogonal group is one of the classical Lie groups, which possess the noticeable property to be also manifolds. Therefore, with the arguments of the above appendix, the double orthogonal group is also a manifold. This gives the connection between the considered criterion function and the gradient-flow structure of HM equations.

A.3. *Killing form and metric*

In order to clarify the concepts underlying the invoked Killing metric, it is useful to first recall some notation from Lie algebra theory.

Definition 4

(Refs. 14, 19, 20.) Let us denote with $\mathbf{so}(n, \mathbb{R})$ the set of real-valued skew-symmetric matrices of dimension n.

Given a Lie algebra $\mathbf{g}$ and two elements of the algebra X and Y, the adjoint operator associated to X as a function of Y is defined as $\operatorname{ad}_X(Y) = XY - YX$. By definition, the adjoint operator is skew-symmetric, thus it belongs to a Lie-algebra $\mathbf{so}$.

The Killing form is an inner-product on a finite-dimensional Lie algebra defined by $K(X, Y) = \operatorname{tr}[\operatorname{ad}_X \operatorname{ad}_Y]$.

It is known that an inner-product defines a metric, thus the Killing form defines the Killing metric $K(X, X)$ on a Lie algebra. In the paper we used the standard Euclidean metric in the Lie algebra $\mathbf{so}(n, \mathbb{R})$, namely $\operatorname{tr}[X'X]$, because it is a linear space. This expression has the same structure of a Killing form because the elements of $\mathbf{so}$ are skew-symmetric matrices, thus we may identify the Euclidean metric on $\mathbf{so}$ with the Killing metric on the same space (up to an inessential constant).

References

1. S.-I. Amari 1998, "Natural gradient works efficiently in learning," *Neural Computation* **10**, 251–276.
2. H. Bourlard and Y. Kamp 1988, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics* **59**, 291–294.
3. R. W. Brockett 1991, "Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems," *Linear Algebra and Its Applications* **146**, 79–91.
4. T.-P. Chen, S.-I. Amari and Q. Lin 1998, "A unified algorithm for principal and minor component extraction," *Neural Networks* **11**, 385–390.
5. A. Cichocki and R. Unbehauen 1992, "Neural networks for computing eigenvalues and eigenvectors," *Biological Cybernetics* **68**, 155–164.
6. 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.
7. E. F. Deprette (ed.) 1988, *SVD and Signal Processing* (Amsterdam, Elsevier Science).
8. K. I. Diamantaras and S.-Y. Kung 1994, "Cross-correlation neural network models," *IEEE Trans. on Signal Processing* **42**(11), 3218–3223.
9. S. Fiori 2001, "A theory for learning by weight flow on Stiefel-Grassman manifold," *Neural Computation* **13**(7), 1625–1647.
10. S. Fiori 2002, "A theory for learning based on rigid bodies dynamics," *IEEE Trans. on Neural Networks* **13**(3), 521–531.
11. S. Fiori 2002, "Complex-weighted one-unit 'Rigid-Bodies' learning rule for independent component analysis," *Neural Processing Letters* **15**(3), 275–282.
12. S. Fiori 2002, "Unsupervised neural learning on Lie group," *International Journal of Neural Systems* **12**(3 & 4), 219–246.
13. S. Fiori, "A minor subspace algorithm based on neural stiefel dynamics," *International Journal of Neural Systems* **12**(5), 339–350.
14. W. Fulton and J. Harris 1991, *Representation Theory* (New York: Springer-Verlag)
15. G. H. Goulb and C. F. van Loan 1996, *Matrix Computations* (The John Hopkins University Press, third edition).
16. W. Hahn 1963, *Theory and Application of Lyapunov's Direct Method* (Englewood Cliffs, New Jersey: Prentice-Hall)

17. S. Haykin 1991, *Adaptive Filter Theory* (Prentice-Hall).
18. U. Helmke and J. B. Moore 1992, "Singular value decomposition via gradient and self-equivalent flows," *Linear Algebra and its Applications* **169**, 223–248.
19. U. Helmke and J. B. Moore 1993, *Optimization and Dynamical Systems* (Springer-Verlag, Berlin)
20. N. Jacobson 1979, *Lie Algebras* (New York: Dover).
21. A. K. Jain 1989, *Fundamentals of Digital Image Processing* (Englewood Cliffs, NJ: Prentice-Hall)
22. W.-S. Lu, H.-P. Wang and A. Antoniou 1990, "Design of two-dimensional FIR digital filters by using the singular value decomposition," *IEEE Trans. on Circuits and Systems* **CAS-37**, 35–46.
23. J. R. Magnus and H. Neudecker 1988, *Matrix Differential Calculus With Applications in Statistics and Econometrics* (Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons).
24. 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.
25. O. Nestares and R. Navarro 2001, "Probabilistic estimation of optical flow in multiple band-pass directional channels," *Image and Vision Computing Journal* **19**(6), 339–351.
26. E. Oja, "Neural networks, principal components and subspaces," *International Journal of Neural System* **1**, 61–68.
27. T. D. Sanger 1994, "Two iterative algorithms for computing the singular value decomposition from input/output samples," in J. D. Cowan, G. Tesauro and J. Alspector (eds), Morgan-Kauffman Publishers Inc., *Advances in Neural Processing Systems* **6**, 1441–151.
28. M. Salmeron, J. Ortega, C. G. Puntonet and A. Prieto 2001, "Improved RAN sequential prediction using orthogonal techniques," *Neurocomputing* **41**(1–4), 153–172.
29. T. Sasagawa 1982, "On the finite escape phenomena for matrix Riccati equations," *IEEE Trans. on Automatic Control* **AC-27**(4), 977–979.
30. S. T. Smith 1991, "Dynamic system that perform the SVD," *Systems and Control Letters* **15**, 319–327.
31. R. Vaccaro (ed.) 1991, *SVD and Signal Processing II: Algorithms, Analysis and Applications* (Amsterdam, Elsevier Science)
32. A. Weingessel and K. Hornik 1997, "SVD algorithms: APEX-like versus subspace methods," *Neural Processing Letters* **5**, 177–184.
33. A. Weingessel 1999, *An Analysis of Learning Algorithms in PCA and SVD Neural Networks* (Ph.D. Dissertation, Technical University of Wien, Austria).