# SIMULTANEOUS TRACKING OF THE BEST BASIS IN REDUCED-RANK WIENER FILTER

*Toshihisa Tanaka*

Tokyo Univ. of Agriculture & Technology
184-8588 Japan. tanakat@cc.tuat.ac.jp

*Simone Fiori*

Università Politecnica delle Marche
I-60131 Ancona, Italy. fiori@deit.univpm.it

**ABSTRACT**

A new on-line learning algorithm that yields a reduced-rank Wiener filter (RRWF) is proposed. The RRWF is defined as the matrix of prescribed rank that provides the best least-squares approximation of a given signal. This implies that an RRWF determines only a subspace, but is not endowed with information of basis functions or axes for the subspace. In other words, even if we want to reduce the rank of the estimated RRWF, we should learn another RRWF of "more reduced rank" again. Our goal in this paper is therefore to establish a learning rule that simultaneously tracks basis functions yielding a matrix that gives an RRWF. To this end, we reformulate the optimization problem of RRWFs, which will be solved by a gradient-based algorithm derived within the framework of differential geometry. Numerical examples are illustrated to support the proposals in the paper.

## 1. INTRODUCTION

The well-known Wiener filter (WF) is known as an operator that gives the optimal approximation of the observation to the original signal in the sense of mean square error [1]. Let $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ be the original and the observed signals, respectively. Generally, $\boldsymbol{x}$ is unknown, but the correlation matrix of $\boldsymbol{y}$, denoted as $\boldsymbol{R}_{\boldsymbol{yy}} \in \mathbb{R}^{m\times m}$, and the cross-correlation matrix between $\boldsymbol{x}$ and $\boldsymbol{y}$, denoted as $\boldsymbol{R}_{\boldsymbol{xy}} \in \mathbb{R}^{n\times m}$, are known or can be estimated. In the paper, we also make use of $\boldsymbol{R}_{\boldsymbol{yx}} = \boldsymbol{R}_{\boldsymbol{xy}}^T$. The WF is given as a solution of the approximation problem to minimize the following cost function:

$$J[\boldsymbol{P}] = \frac{1}{2}E\|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{y}\|^2 = \frac{1}{2}\mathrm{tr}[\boldsymbol{R}_{\boldsymbol{xx}} - 2\boldsymbol{P}\boldsymbol{R}_{\boldsymbol{yx}} + \boldsymbol{P}\boldsymbol{R}_{\boldsymbol{yy}}\boldsymbol{P}^T], \tag{1}$$

with $\boldsymbol{P} \in \mathbb{R}^{n\times m}$. We can explicitly solve this problem and the WF is given by the matrix:

$$\boldsymbol{P}_{\mathrm{WF}} = \arg\min_{\boldsymbol{P}\in\mathbb{R}^{n\times m}} J[\boldsymbol{P}] = \boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{R}_{\boldsymbol{yy}}^{-1}. \tag{2}$$

The estimated signal by the WF is given by $\boldsymbol{P}_{\mathrm{WF}}\boldsymbol{y}$.

An extension of the WF may be obtained by fixing the rank of $\boldsymbol{P}$, denoted by $r$, which is smaller than $\min(m, n)$. When $J[\boldsymbol{P}]$ is minimized under the rank constraint , $\mathrm{rank}(\boldsymbol{P}) = r$, the minimizer of the optimization problem given by

$$\boldsymbol{P}_{\mathrm{RRWF}}^{(r)} = \arg\min_{\mathrm{rank}(\boldsymbol{P})=r} J[\boldsymbol{P}] \tag{3}$$

is termed a *reduced-rank Wiener filter* (RRWF) of rank $r$ [2, 3]. The RRWF can be regarded as a realization of the reduced-rank regression model [2, 3], which describes the input-output relation in mobile communication systems with sensor array and of linear signal/image restoration. We can solve this minimization problem and obtain the solution in closed-form as [3]:

$$\boldsymbol{P}_{\mathrm{RRWF}}^{(r)} = \boldsymbol{U}_r\boldsymbol{U}_r^T\boldsymbol{P}_{\mathrm{WF}}, \tag{4}$$

where $\boldsymbol{U}_r$ is the matrix whose columns are the first $r$ columns of $\bar{\boldsymbol{U}}$ in $\mathbb{R}^n$, which is the unitary eigenmatrix associated with the eigenvalue decomposition of $\boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{R}_{\boldsymbol{yy}}^{-1}\boldsymbol{R}_{\boldsymbol{yx}}$.

For applications to adaptive systems such as communication systems, we need an adaptive learning algorithm. Hua *et al*.[3] proposed to represent an RRWF denoted by $\boldsymbol{P}^{(r)}$ (for sake of simplicity and convenience, an RRWF is written as $\boldsymbol{P}^{(r)}$, instead of $\boldsymbol{P}_{\mathrm{RRWF}}^{(r)}$, hereafter.) as the product of two matrices, $\boldsymbol{P}^{(r)} = \boldsymbol{A}\boldsymbol{B}^T$, where $\boldsymbol{A} \in \mathbb{R}^{n\times r}$ and $\boldsymbol{B} \in \mathbb{R}^{m\times r}$, and to alternatively calculate $\boldsymbol{A}$ and $\boldsymbol{B}$. The idea in [3] is to change the original minimization problem to the following one:

$$(\boldsymbol{A}, \boldsymbol{B})^{(r)} = \arg\min_{\boldsymbol{A}\in\mathbb{R}^{n\times r}, \boldsymbol{B}\in\mathbb{R}^{m\times r}} J[\boldsymbol{A}, \boldsymbol{B}], \tag{5}$$

where

$$J[\boldsymbol{A}, \boldsymbol{B}] = \frac{1}{2}\mathrm{tr}[\boldsymbol{R}_{\boldsymbol{xx}} - 2\boldsymbol{A}\boldsymbol{B}^T\boldsymbol{R}_{\boldsymbol{yx}} + \boldsymbol{A}\boldsymbol{B}^T\boldsymbol{R}_{\boldsymbol{yy}}\boldsymbol{B}\boldsymbol{A}^T]. \tag{6}$$

Then, $\frac{\partial J}{\partial \boldsymbol{A}} = 0$ and $\frac{\partial J}{\partial \boldsymbol{B}} = 0$ are alternatively solved. However, this is a batch algorithm, not a fully on-line algorithm. Moreover, $\boldsymbol{A}$ and $\boldsymbol{B}$ are not uniquely determined because $\boldsymbol{P}^{(r)} = \boldsymbol{A}\boldsymbol{C}\boldsymbol{C}^{-1}\boldsymbol{B}^T$ with any invertible $\boldsymbol{C} \in \mathbb{R}^{r\times r}$. This property is crucial especially for on-line estimation, since in wireless communication systems, the rank can dynamically change and if it varies, then we should learn the RRWF of the new rank again.

The objective of this paper is to newly establish a learning algorithm for RRWFs in such a way that even if the rank of an RRWF changes or is more reduced, it is unnecessary to reset the current estimation of the RRWF and re-learning another RRWF of new rank. To this end, we consider an RRWF as an operator that gives the best approximation to the original signal. Specifically, we describe an estimated signal by the RRWF as the superposition of orthonormal basis functions, which should be simultaneously tracked on-line. A similar idea is the canonical component estimation [3]. The algorithm extracts a rank-1 matrix whose sum is an RRWF; however the extraction is one-by-one manner, not in a simultaneous way, and doesn't focus on the basis functions. In this paper, then, we propose a new cost function for obtaining an RRWF including information of the orthonormal basis for the range of the RRWF by introducing the unitary constratint on one matrix to be optimized. To do this, we derive the gradient-based algorithm by using the theory of differential geometry [4, 5]. In the end of the paper, experimental results are shown to support the theoretical derivation of the learning algorithm in this paper.

Toshihisa Tanaka is also with the Laboratory for Brain Signal Processing, RIKEN Brain Science Institute.

## 2. PROBLEM FORMULATION AND COST FUNCTION

In this section, we formulate the problem to be solved in this paper. Recall the closed-form solution described as in (4), which gives another aspect of the RRWF. It implies that the estimated signal by the RRWF may be rewritten by the superposition of orthonormal basis functions, that is,

$$\boldsymbol{P}^{(r)}\boldsymbol{y} = \boldsymbol{U}_r\boldsymbol{U}_r^T\boldsymbol{P}_{\mathrm{WF}}\boldsymbol{y} = \bar{\boldsymbol{U}}\boldsymbol{\Gamma}_r\bar{\boldsymbol{U}}^T\boldsymbol{P}_{\mathrm{WF}}\boldsymbol{y} = \sum_{i=1}^{r}\langle\boldsymbol{u}_i, \boldsymbol{P}_{\mathrm{WF}}\boldsymbol{y}\rangle\boldsymbol{u}_i, \tag{7}$$

where $\boldsymbol{\Gamma}_r$ be a diagonal matrix whose diagonal elements are unity for the first $r$ diagonals and are 0 for the remaining diagonals, and $\boldsymbol{u}_i$ is the $i$th column of $\bar{\boldsymbol{U}}$. This observation more clearly tells us that the problem should be to find a basis, not a subspace, giving the best approximation to the original signal, as illustrated in Fig. 1. That is, we have to track $\boldsymbol{U}_r$ (or $\bar{\boldsymbol{U}}$) and to compute the RRWF of any rank equal to or less than $r$.

Let us start with the factorization of $\boldsymbol{P}^{(r)}$ by the product of $\boldsymbol{U}$ and $\boldsymbol{V}$, say, $\boldsymbol{P}^{(r)} = \boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U}$ is a column-orthogonal matrix of size $n \times r$ and $\boldsymbol{V}$ is a matrix of size $m \times r$. This implies that matrix $\boldsymbol{U}$ belongs to the *Stiefel manifold* defined as:

$$St(n, r) = \{\boldsymbol{U} \in \mathbb{R}^{n\times r} | \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r\}. \tag{8}$$

Motivated by the closed-form solution as in (4), we introduce a double-optimization problem to find out $\boldsymbol{U}$ and $\boldsymbol{V}$, which are given by

$$(\boldsymbol{U}, \boldsymbol{V})^{(r)} = \arg\min_{\boldsymbol{U}\in St(n,r), \boldsymbol{V}\in\mathbb{R}^{m\times r}} J[\boldsymbol{U}, \boldsymbol{V}], \tag{9}$$

where

$$\begin{aligned} J[\boldsymbol{U}, \boldsymbol{V}] &= \frac{1}{2}\mathrm{tr}[\boldsymbol{R}_{xx} - 2\boldsymbol{U}\boldsymbol{V}^T\boldsymbol{R}_{yx} + \boldsymbol{U}\boldsymbol{V}^T\boldsymbol{R}_{yy}\boldsymbol{V}\boldsymbol{U}^T] \\ &= \frac{1}{2}\left(\mathrm{tr}[\boldsymbol{R}_{xx}] - \mathrm{tr}[2\boldsymbol{V}^T\boldsymbol{R}_{yx}\boldsymbol{U} - \boldsymbol{V}^T\boldsymbol{R}_{yy}\boldsymbol{V}]\right). \end{aligned} \tag{10}$$

We have just imposed more strict constraint – orthonormality – on $\boldsymbol{A}$ in problem (5). However, the optimization problem no longer determines the best basis functions. By introducing an arbitrary orthogonal matrix of size $r\times r$, $\tilde{\boldsymbol{U}}$, we can observe that the RRWF of rank $r$ has another expression given by $\boldsymbol{P}^{(r)}_{\mathrm{RRWF}} = \boldsymbol{U}_r\tilde{\boldsymbol{U}}^T\tilde{\boldsymbol{U}}\boldsymbol{U}_r^T\boldsymbol{X}_{\mathrm{WF}}$, which also minimize the cost function. This observation seems similar to one in the relation between the principal/minor subspace analysis (PSA/MSA) and the principal/minor component analysis (PCA/MCA) [6, 7].

A key idea in the paper to solve the problem of ambiguity is to use the following "weighted" cost function:

$$\hat{J}[\boldsymbol{U}, \boldsymbol{V}] = -\frac{1}{2}\mathrm{tr}[\boldsymbol{D}(2\boldsymbol{V}^T\boldsymbol{R}_{yx}\boldsymbol{U} - \boldsymbol{V}^T\boldsymbol{R}_{yy}\boldsymbol{V})], \tag{11}$$

where $\boldsymbol{D}$ is a $r \times r$ diagonal matrix of positive elements ordered in descending order. It should be noted that we have removed the first term in (10) since it will vanish by the differentiation. As we will see, the weighted sum of the diagonal elements in the argument of $\hat{J}$ can avoid the ambiguity. This cost function is minimized with preserving that $\boldsymbol{U}$ is orthogonal. The derivation is accomplished by applying techniques of differential geometry [4, 5]. In the next section, we show mathematical preliminaries and how to obtain the learning rule, and then the gradients for $\boldsymbol{U}$ and $\boldsymbol{V}$ are derived.

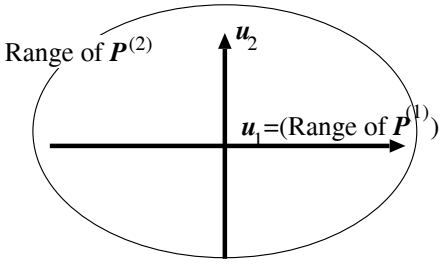



**Fig. 1**. Simple illustration for our problem: if we know $\{\boldsymbol{u}_i\}_{i=1}^r$, then we get $\{\boldsymbol{P}^{(i)}\}_{i=1}^r$. On the other hand, even though we know $\boldsymbol{P}^{(r)}$, we cannot get $\boldsymbol{P}^{(i)}$ for $i = 1, \ldots, r-1$.

## 3. DIFFERENTIAL GEOMETRICAL DERIVATION OF LEARNING ALGORITHMS

We derive in this section ordinary differential equations (ODEs) providing gradient flows to solve the optimization problem. We deal with the optimization problem in the framework of differential geometry which is the calculus of manifolds. The key concept is to formalize the constraints that the sought-for filter should satisfy and to define a proper set of filters that satisfy such constraints, which form a smooth manifold. If the filter-adaptation problem is formulated – as it is often the case – as an optimization problem, then optimization may be effected directly on the defined smooth manifold. We briefly review this concept in the rest of this section.

### 3.1. Review of Gradient-Based Optimization on Manifolds

#### *3.1.1. Riemannian Gradient*

Let $T_\xi\mathcal{M}$ be the tangent space to manifold $\mathcal{M}$ in point $\xi \in \mathcal{M}$ and let $(\mathcal{M}, g^{\mathcal{M}})$ be a Riemannian manifold, where $g^{\mathcal{M}}_\xi : T_\xi\mathcal{M} \times T_\xi\mathcal{M} \to \mathbb{R}$ denotes a bilinear scalar product that turns $\mathcal{M}$ into a metric space. In particular, the Euclidean scalar product denoted by $g^e : T_\xi\mathcal{M} \times T_\xi\mathcal{M} \to \mathbb{R}$ is necessary to define the gradient on a Riemannian manifold. The gradient, $\mathrm{grad}^{\mathcal{M}}_\xi f$, of a differentiable function, $f : \mathcal{M} \to \mathbb{R}$, in $\xi$ on $(\mathcal{M}, g^{\mathcal{M}})$ is defined by the following two conditions [5]: 1) $\mathrm{grad}^{\mathcal{M}}_\xi f \in T_\xi\mathcal{M}$ (tangency condition) and 2) $g^{\mathcal{M}}_\xi\left(\mathrm{grad}^{\mathcal{M}}_\xi f, \boldsymbol{v}\right) = g^e\left(\dfrac{\partial f}{\partial \xi}, \boldsymbol{v}\right)$ for all $\boldsymbol{v} \in T_\xi\mathcal{M}$ (compatibility condition), where $\dfrac{\partial f}{\partial \xi}$ is the standard gradient (or Jacobian) of $f$.

By gradient-based differential equation for the constrained optimization of function $f : \mathcal{M} \to \mathbb{R}$ on $\mathcal{M}$, where the smooth manifold, $\mathcal{M}$, fully describes the constraints, it is meant:

$$\dot{\xi}(t) = \pm\mathrm{grad}^{\mathcal{M}}_\xi f(\xi(t)), \;\; \xi(0) = \xi_0 \in \mathcal{M}, \tag{12}$$

where the positive sign denotes maximization and the negative sign denotes minimization of $f$.

#### *3.1.2. Integration on a Geodesic*

In order to design an effective adapting algorithm, it is necessary to develop a suitable numerical integration method for numerically solving the differential equation (12). In the case that the manifold $\mathcal{M}$ is a flat space, like, e.g., $\mathbb{R}^n$, several integration methods are available from the literature, like, e.g., the Euler method and more sophisticate methods like the ones belonging to the class of Runge-Kutta integration schemes. However, in many cases of interest

the manifold $\mathcal{M}$ has the structure of a *curved space*: This makes it necessary to develop customized integration methods valid for curved spaces. A popular one is based on the concept of *geodesic*. Geodesics on curved spaces replace the concept of "straight lines" on flat spaces. Let us denote by $\gamma^{\mathcal{M}}_{\xi,\boldsymbol{v}}(t)$ a geodesic on the manifold departing from the point $\xi$ in the direction $\boldsymbol{v}$ at point $t \in [0,1]$. The boundary conditions write $\gamma^{\mathcal{M}}_{\xi,\boldsymbol{v}}(0) = \xi$ and $\dfrac{d}{dt}\gamma^{\mathcal{M}}_{\xi,\boldsymbol{v}}(0) = \boldsymbol{v}$. As a geodesic is a curve completely belonging to the manifold $\mathcal{M}$, it can be used to move within the manifold from a starting point to an arrival point along a prescribed (tangent) direction. That is, it may be used to solve a differential equation of the kind $\dot{\xi}(t) = \boldsymbol{g}(t)$, like equation (12), in the following way: $\xi(n+1) = \gamma^{\mathcal{M}}_{\xi(n),\boldsymbol{g}(n)}(\mu(n))$, where $\xi(n)$ denotes the obtained approximation of the exact flow $\xi(t)$, $\boldsymbol{g}(n)$ denotes the value of $\boldsymbol{g}(\xi(n))$ and $\mu(n)$ plays the role of the familiar adapting stepsize schedule. Of course, $\xi(0) = \xi_0$.

### 3.2. Derivation of Riemannian Gradient for $U$ and $V$

By using the framework of differential geometry reviewed in the previous subsection, we solve the optimization problem numerically by joint gradient-based algorithms given as

$$\dot{\boldsymbol{U}} = -\mathrm{grad}^{St(n,r)}_{\boldsymbol{U}} \hat{J}[\boldsymbol{U},\boldsymbol{V}],\ \dot{\boldsymbol{V}} = -\mathrm{grad}^{\mathbb{R}^{m\times r}}_{\boldsymbol{V}} \hat{J}[\boldsymbol{U},\boldsymbol{V}]. \tag{13}$$

We now derive gradients for $\boldsymbol{U}$ and $\boldsymbol{V}$. By simple differentiation, Jacobians with respect to $\boldsymbol{U}$ and $\boldsymbol{V}$ are obtained as

$$\frac{\partial \hat{J}}{\partial \boldsymbol{U}} = -\boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{V}\boldsymbol{D},\ \frac{\partial \hat{J}}{\partial \boldsymbol{V}} = -(\boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U} - \boldsymbol{R}_{\boldsymbol{yy}}\boldsymbol{V})\boldsymbol{D}. \tag{14}$$

We may consider for $St(n,r)$ the well-known canonical metrics [4] defined as

$$g^{St(n,r)}_{\boldsymbol{U}}(\boldsymbol{H}_1,\boldsymbol{H}_2) = \mathrm{tr}\left[\boldsymbol{H}_1^T\left(\boldsymbol{I}_r - \frac{1}{2}\boldsymbol{U}\boldsymbol{U}^T\right)\boldsymbol{H}_2\right], \tag{15}$$

$\forall \boldsymbol{H}_1, \boldsymbol{H}_2 \in T_{\boldsymbol{U}}St(n,r)$. The Riemannian gradient on the Stiefel manifold is derived via the tangency and compatibility conditions, as described in [4]. The tangent space to the Stiefel manifold in a given point, $\boldsymbol{U} \in St(n,r)$, is $T_{\boldsymbol{U}}St(n,r) = \{\boldsymbol{H} \in \mathbb{R}^{n\times r} | \boldsymbol{H}^T\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{H} = \boldsymbol{0}_r\}$. Then, we obtain the gradient as

$$\mathrm{grad}^{St(n,r)}_{\boldsymbol{U}} \hat{J} = \frac{\partial \hat{J}}{\partial \boldsymbol{U}} - \boldsymbol{U}\left(\frac{\partial \hat{J}}{\partial \boldsymbol{U}}\right)^T \boldsymbol{U} = -\boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{V}\boldsymbol{D} + \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T\boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U}. \tag{16}$$

From (13), we obtain the ODE for $\boldsymbol{U}$ as:

$$\dot{\boldsymbol{U}} = \boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{V}\boldsymbol{D} - \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T\boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U}, \tag{17}$$

To the end of geodesic-based numerical integration, we also recall the expression of geodesic on the Stiefel manifold when this is equipped with canonical metrics [5]:

$$\gamma^{St(n,r)}_{\boldsymbol{U},-\mathrm{grad}^{St(n,r)}_{\boldsymbol{U}}\hat{J}}(t) = \left(\exp\left[-t\left(\left(\frac{\partial \hat{J}}{\partial \boldsymbol{U}}\right)\boldsymbol{U}^T - \boldsymbol{U}\left(\frac{\partial \hat{J}}{\partial \boldsymbol{U}}\right)^T\right)\right]\right)\boldsymbol{U}. \tag{18}$$

For $\mathbb{R}^{m\times r}$, we introduce the following weighted Euclidean metric:

$$g^{\hat{\boldsymbol{V}}}_{\boldsymbol{V}}(\boldsymbol{H}_1,\boldsymbol{H}_2) = \mathrm{tr}[\boldsymbol{H}_1^T\boldsymbol{D}\boldsymbol{H}_2]. \tag{19}$$

Since the tangent space of the Euclidean space is itself, we can easily derive the gradient in $\mathbb{R}^{m\times r}$ with respect to $g^{\hat{\boldsymbol{V}}}_{\boldsymbol{V}}$ such that it meets the tangency and the compatibility conditions. Therefore, the gradient is given by

$$\mathrm{grad}^{\mathbb{R}^{m\times r}}_{\boldsymbol{V}} \hat{J} = \frac{\partial \hat{J}}{\partial \boldsymbol{V}}\boldsymbol{D}^{-1} = -(\boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U} - \boldsymbol{R}_{\boldsymbol{yy}}\boldsymbol{V}). \tag{20}$$

It follows then from (13) that we obtain the following ODE for $\boldsymbol{V}$:

$$\dot{\boldsymbol{V}} = \boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U} - \boldsymbol{R}_{\boldsymbol{yy}}\boldsymbol{V}. \tag{21}$$

Note that since $\boldsymbol{V}$ is not on a curved space, we can use the classical Euler-type integration to numerically solve equation (21).

### 3.3. Behavior of the Joint Learning Algorithm

A simple analysis for the behavior of $\boldsymbol{U}$ and $\boldsymbol{V}$ is here given. A key observation is the following fact, which can be shown by solving the non-homogeneous linear ODE given as in (21).

**Proposition 1** Assume that the dynamics of $\boldsymbol{V}$ is given by (21). Then, $\boldsymbol{V}(t) \to \boldsymbol{R}^{-1}_{\boldsymbol{yy}}\boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U}$, as $t \to \infty$.

Proof is omitted here. A standard technique to solve non-homogeneous linear ODEs is applicable as proof. Also, considering the fact that this ODE is a direct deduction from the differentiation of $\hat{J}$, we may understand the proposition. If $\boldsymbol{V} = \boldsymbol{R}^{-1}_{\boldsymbol{yy}}\boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{U}$ in (17), then it holds that

$$\dot{\boldsymbol{U}} = \boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{R}^{-1}_{\boldsymbol{yy}}\boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U}\boldsymbol{D} - \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{R}^{-1}_{\boldsymbol{yy}}\boldsymbol{R}_{\boldsymbol{yx}}\boldsymbol{U}, \tag{22}$$

which is a quite interesting consequence, since this ODE is exactly the same as the learning algorithm for principal components of signals with the correlation matrix given by $\boldsymbol{R}_{\boldsymbol{xy}}\boldsymbol{R}^{-1}_{\boldsymbol{yy}}\boldsymbol{R}_{\boldsymbol{yx}}$, proposed by Xu [8] and analyzed in several papers [6, 7]. The following features have been theoretically proven [6, 7]:

1. Stability on the Stiefel manifold;
2. Stability in the point where principal components are extracted.

The first property guarantees that $\boldsymbol{U}$ always varies satisfying $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r$, and a perturbation from the manifold will decay to zero. The second one assures that $\boldsymbol{U}$ is stable if and only if $\boldsymbol{U} = \boldsymbol{U}_r$, which meets our objective.

## 4. NUMERICAL EXAMPLES

The ODEs given as in (17) and (21) are numerically integrated by the following joint update rules:

$$\boldsymbol{U}(k+1) = \hat{\gamma}^{St(n,r)}_{\boldsymbol{U},-\mathrm{grad}^{St(n,r)}_{\boldsymbol{U}}\hat{J}}(\mu_1(k)), \tag{23}$$

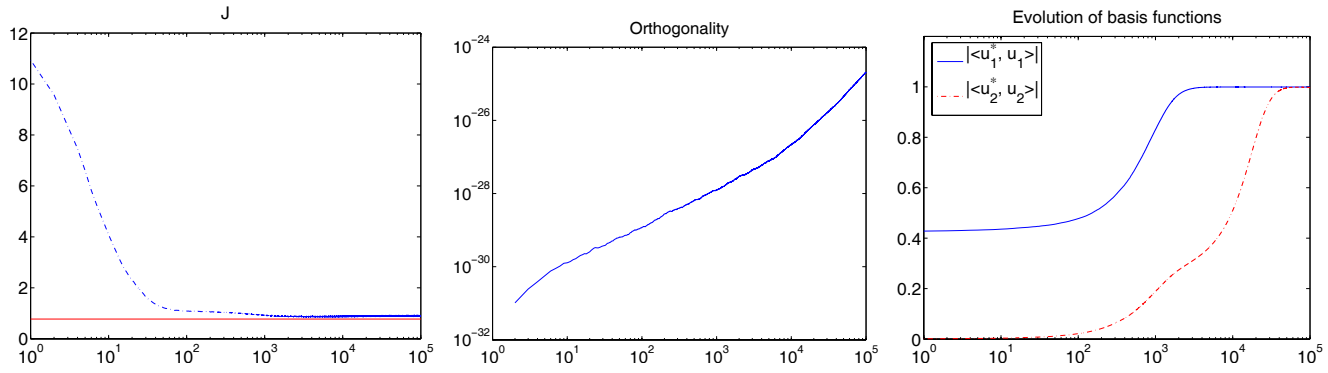
$$\boldsymbol{V}(k+1) = \boldsymbol{V}(k) + \mu_2(k)\Delta\boldsymbol{V}(k), \tag{24}$$

where

$$\hat{\gamma}^{St(n,r)}_{\boldsymbol{U},-\mathrm{grad}^{St(n,r)}_{\boldsymbol{U}}\hat{J}}(t) = e^{-t\left[\boldsymbol{U}(k)\boldsymbol{D}\boldsymbol{V}^T(k)\boldsymbol{y}(k)\boldsymbol{x}^T(k) - \boldsymbol{x}(k)\boldsymbol{y}^T(k)\boldsymbol{V}(k)\boldsymbol{D}\boldsymbol{U}^T(k)\right]}\boldsymbol{U}(k) \tag{25}$$

$$\Delta\boldsymbol{V}(k) = \boldsymbol{y}(k)\boldsymbol{x}^T(k)\boldsymbol{U}(k) - \boldsymbol{y}(k)\boldsymbol{y}^T(k)\boldsymbol{V}(k), \tag{26}$$

and $k$ is an integer index denoting a learning iteration counter. Here, the expectation operator has been removed to employ online learning, which is the reason why the notation for geodesic denoted by $\hat{\gamma}^{St(n,r)}_{\boldsymbol{U},-\mathrm{grad}^{St(n,r)}_{\boldsymbol{U}}\hat{J}}(t)$ is used in the above update rule instead of $\gamma^{St(n,r)}_{\boldsymbol{U},-\mathrm{grad}^{St(n,r)}_{\boldsymbol{U}}\hat{J}}(t)$.




(a) Criterion $J$ versus the iteration number.

(b) $\eta_U$ versus the iteration number.

(c) $|\langle \boldsymbol{u}_i^*, \boldsymbol{u}_i(k)\rangle|$ for $i = 1, 2$ versus the iteration number.

**Fig. 2**. Behavior of the proposed learning algorithm.

In order to evaluate the performance of the the learning algorithms, we introduce the following criteria:

$$J(k) = J[\boldsymbol{U}(k), \boldsymbol{V}(k)], \quad (27)$$

$$\eta_U(k) = \|\boldsymbol{U}^T(k)\boldsymbol{U}(k) - \boldsymbol{I}_r\|_{\mathrm{F}}^2, \quad (28)$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius matrix norm. Moreover, we evaluate $|\langle \boldsymbol{u}_i^*, \boldsymbol{u}_i(k)\rangle|$, where $\boldsymbol{u}_i^*$ and $\boldsymbol{u}_i(k)$ are the $i$th columns of $\boldsymbol{U}_r$ and $\boldsymbol{U}(k)$, respectively.

In the test, it is set that $m = n = 4$, and $r = 3$. Stochastic vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ were generated by the model: $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}$, where we assumed that

$$\boldsymbol{R}_{xx} = \begin{bmatrix} .9 & .4 & .7 & .3 \\ .4 & .3 & .5 & .4 \\ .7 & .5 & 1.0 & .6 \\ .3 & .4 & .6 & .9 \end{bmatrix}, \boldsymbol{A} = \frac{1}{5}\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 1 & -2 & -3 & -4 \\ 0 & 1 & -2 & 3 \end{bmatrix},$$

and $\boldsymbol{\epsilon}$ is a noise, of variance $\sigma^2 = 4$, incorrelated with the signal $\boldsymbol{x}$. In this way, it turns out that $\boldsymbol{R}_{xy} = \boldsymbol{R}_{xx}\boldsymbol{A}^T$. The learning rates in the numerical experiments are fixed to $\mu_1(k) = 0.01$ and $\mu_2(k) = 0.001$ for all $k$. We used the MATLAB for this simulation[1].

Independent tests were performed 100 times and the ensemble average is plotted in each figure. Figure 2 shows the behavior of the proposed learning rule. The straight line in Fig. 2(a) corresponds to the optimal $J[\boldsymbol{P}^{(r)}]$. As seen in these figures, $\boldsymbol{U}(k)\boldsymbol{V}^T(k)$ quickly converges to an RRWF of rank $r$ by preserving the orthogonality of the columns of matrix $\boldsymbol{U}(k)$, as can be seen in Fig. 2(b), where $\eta_U$ is consistently zero. On the other hand, it can be seen in Fig. 2(c) that the convergence of $\boldsymbol{U}(k)$ is relatively slow; however, it definitely approaches $\boldsymbol{U}_r$, which implies that an RRWF of any rank less than $r$ can be obtained by eliminating right columns of $\boldsymbol{U}(k)$. It follows from this observation that we succeeded in obtaining the "best" orthonormal basis for the subspace (the whole space in this experiment) spanned by the RRWF.

Although we used in this paper the geodesic-based integration, we can also apply the classical Euler method for the numerical integration of ODEs, which may be regarded as a first-order approximation of the geodesic-based one. We confirmed that this Euler integration is also stable and gets converged, though orthogonality $\nu_U$ is relatevely larger.

[1]The `expm` was used to calculate the matrix exponential in (25).

## 5. CONCLUSION

This paper has presented a novel on-line learning algorithm that can estimate not only an RRWF of rank $r$, but also find out a basis for the $r$-dimensional subspace spanned by the RRWF. Therefore, one can obtain rank-1 to rank-$r$ RRWFs just from the estimated matrices. We have theoretically analyzed the behavior of the proposed learning algorithms and have shown numerical examples. Future work might focus on applications to communication systems as well as signal restoration. Moreover, the development of learning algorithms for reduced-rank general filters would be an open problem.

## 6. REFERENCES


[1] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. New York, NY: Prentice Hall, 2002.

[2] P. Stoica and M. Viberg, "Reduced-rank linear regression," in *Proc. 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing (SSAP '96)*, pp. 542–545, 1996.

[3] Y. Hua, M. Nikpour, and P. Stoica, "Optimal reduced-rank estimation and filtering," *IEEE Trans. Signal Processing*, vol. 49, pp. 457–469, Mar. 2001.

[4] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. and App.*, vol. 20, no. 2, pp. 303–353, 1998.

[5] S. Fiori, "Formulation and integration of learning differential equations on the Stiefel manifold," *IEEE Trans. Neural Networks*, vol. 16, no. 6, pp. 1697–1701, 2005.

[6] T. Chen and S. Amari, "Unified stabilization approach to principal and minor components extraction algorithms," *Neural Networks*, vol. 14, no. 10, pp. 1377–1387, 2001.

[7] T. Tanaka, "Generalized weighted rules for principal components tracking," *IEEE Trans. Signal Processing*, vol. 53, pp. 1243–1253, Apr. 2005.

[8] L. Xu, "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks*, vol. 6, pp. 627–648, 1993.