

Traian Abrudan, Jan Eriksson, and Visa Koivunen. 2008. Efficient Riemannian algorithms for optimization under unitary matrix constraint. In: Proceedings of the 33rd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008). Las Vegas, NV, USA. 31 March - 4 April 2008, pages 2353-2356.

© 2008 IEEE

Reprinted with permission.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Helsinki University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

EFFICIENT RIEMANNIAN ALGORITHMS FOR OPTIMIZATION UNDER UNITARY MATRIX CONSTRAINT

Traian Abrudan, Jan Eriksson, Visa Koivunen

Helsinki University of Technology, Department of Electrical and Communications Engineering,
SMARAD CoE, Signal Processing Laboratory, FIN-02015 HUT, Finland
(e-mail: {tabrudan, jamaer, visa}@wooster.hut.fi)

ABSTRACT

In this paper we propose practical algorithms for optimization under unitary matrix constraint. This type of constrained optimization is needed in many signal processing applications. Steepest descent and conjugate gradient algorithms on the Lie group of unitary matrices are introduced. They exploit the Lie group properties in order to reduce the computational cost. Simulation examples on signal separation in MIMO systems demonstrate the fast convergence and the ability to satisfy the constraint with high fidelity.

Index Terms— Optimization, unitary matrix constraint, array processing, subspace estimation, source separation

1. INTRODUCTION

Many signal processing applications require optimizing a certain criterion w.r.t a complex-valued matrix, under the constraint that the matrix has orthonormal columns. Such problems arise in communications and array signal processing, for example, high-resolution direction finding, blind and constrained beamforming, and generally all methods where subspace estimation or tracking is needed. Another important class of applications is source separation and Independent Component Analysis (ICA). This type of optimization problems occur also in Multiple-Input Multiple-Output (MIMO) communication systems. For a recent review, see [1].

Commonly, the problem of optimization under unitary matrix constraint is solved on the Euclidean space by using classical gradient algorithms. In order to maintain the constraint satisfied, additional orthogonalization, or some stabilization procedures need to be applied after every iteration. Consequently, such algorithms experience slow convergence or deviations from the constraint [1]. The initial constraint optimization problem may be converted into an unconstrained one, on an appropriate differential manifold [2,3]. In the case of unitary matrix constraint, the appropriate parameter space is the Lie group of $n \times n$ unitary matrices $U(n)$. The nice geometrical properties of $U(n)$ may be exploited in order to solve the optimization problem efficiently and satisfy the constraint with high fidelity. Riemannian geometry based algorithms for optimization with orthogonality constraints are considered in [4]. In [5] a non-Riemannian approach is introduced. Optimization algorithms operating on the unitary group are considered in [1, 6]. Algorithms in the existing literature [2–5] are, however, more general in the sense that they can be applied on more general manifolds than $U(n)$. On the other hand when applied to $U(n)$, they do not take benefit of the special properties arising from the Lie group structure of the manifold.

This work was supported in part by the Academy of Finland and GETA Graduate School.

In this paper, efficient steepest descent (SD) and conjugate gradient (CG) algorithms operating on the Lie group of unitary matrices $U(n)$ are proposed. They move towards the optimum along geodesics, which on a Riemannian manifold correspond to the straight line on the Euclidean space. The main contribution in this paper is that we take full benefit of the geometric properties of the Lie group, such as simple formulas for geodesics and parallel transport, as well as special matrix structures. Therefore, the resulting optimization algorithms are computationally efficient.

This paper is organized as follows. In Section 2 we propose practical steepest descent and conjugate gradient algorithm for optimization under unitary matrix constraint. Simulation results are presented in Section 3. Finally, Section 4 concludes the paper.

2. RIEMANNIAN OPTIMIZATION ALGORITHMS ON THE UNITARY GROUP

In this section we propose steepest descent (SD) and conjugate gradient (CG) algorithms operating on the Lie group of unitary matrices $U(n)$. The goal is to minimize (or maximize) the real-valued cost function \mathcal{J} of $n \times n$ complex matrix argument \mathbf{W} , under unitary matrix constraint, i.e. $\mathbf{W}\mathbf{W}^H = \mathbf{W}^H\mathbf{W} = \mathbf{I}$, where \mathbf{I} is the $n \times n$ identity matrix. The constrained optimization problem on the Euclidean space $\mathbb{C}^{n \times n}$ may be formulated as an unconstrained one, on a different parameter space determined by the constraint. The unitary constraint defines the Lie group of unitary matrices $U(n)$, which is a differential manifold and a multiplicative matrix group at the same time. By exploiting the additional group properties of the manifold of unitary matrices a reduction in complexity is achieved.

2.1. Some key geometrical features of $U(n)$

This subsection describes briefly some Riemannian geometry concepts related to the Lie group of unitary matrices $U(n)$. We also show how the properties of $U(n)$ may be exploited in order to reduce the complexity of the optimization algorithms.

2.1.1. Tangent vectors and tangent spaces

The tangent space is a n^2 -dimensional real vector space attached to every point $\mathbf{W} \in U(n)$, and it may be identified with the matrix space $T_{\mathbf{W}}U(n) \triangleq \{\mathbf{X} \in \mathbb{C}^{n \times n} | \mathbf{X}^H\mathbf{W} + \mathbf{W}^H\mathbf{X} = \mathbf{0}\}$. The tangent space at the group identity \mathbf{I} is the *real Lie algebra* of skew-Hermitian matrices $\mathfrak{u}(n) \triangleq T_{\mathbf{I}}U(n) = \{\mathbf{S} \in \mathbb{C}^{n \times n} | \mathbf{S} = -\mathbf{S}^H\}$.

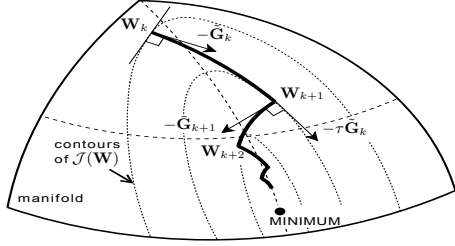


Fig. 1. The SD algorithm takes ninety-degree turns at every iteration, i.e., $\langle -\tilde{\mathbf{G}}_{k+1}, -\tau\tilde{\mathbf{G}}_k \rangle_{\mathbf{W}_{k+1}} = 0$, where τ denotes the parallelism w.r.t. the geodesic connecting \mathbf{W}_k and \mathbf{W}_{k+1} .

2.1.2. Riemannian metric and gradient on $U(n)$

The gradient vector can only be defined after endowing $U(n)$ with a Riemannian metric. The inner product given by $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}} = \frac{1}{2} \Re \{ \text{trace} \{ \mathbf{X} \mathbf{Y}^H \} \}$, $\mathbf{X}, \mathbf{Y} \in T_{\mathbf{W}} U(n)$ induces a bi-invariant metric on $U(n)$. Therefore, the right (and also the left) translation preserves the inner product, i.e., $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{W}} = \langle \mathbf{X} \mathbf{V}, \mathbf{Y} \mathbf{V} \rangle_{\mathbf{W} \mathbf{V}}$, $\forall \mathbf{V} \in U(n)$. This property is called isometry, and it is very useful for performing translations of the tangent vectors (gradients and search directions) from a tangent space to another. The Riemannian gradient at a point $\mathbf{W} \in U(n)$ is:

$$\tilde{\nabla} \mathcal{J}(\mathbf{W}) \triangleq \Gamma_{\mathbf{W}} - \mathbf{W} \Gamma_{\mathbf{W}}^H \mathbf{W}, \quad (1)$$

where $\Gamma_{\mathbf{W}} = \frac{\partial \mathcal{J}}{\partial \mathbf{W}^*}(\mathbf{W})$ is the gradient of \mathcal{J} on the Euclidean space at a given \mathbf{W} [1].

2.1.3. Geodesics and parallel translation on $U(n)$

Intuitively, geodesics on a Riemannian manifold are the locally length minimizing paths. On $U(n)$ they have simple expressions described by the exponential map. The fact that the right translation is an isometry enables simple parallel transport of the tangent vectors along geodesics, via matrix multiplication. When performing the geodesic optimization on $U(n)$, due to computational reasons it is more convenient to translate all tangent vectors into $\mathfrak{u}(n)$ whose elements correspond to skew-Hermitian matrices. Because an isometry maps geodesics into geodesics, the right multiplication also allows translating geodesics from one point to another. The geodesic emanating from the identity element of $U(n)$ in the direction of $\mathbf{S} \in \mathfrak{u}(n)$ is given by $\mathcal{G}_{\mathbf{I}}(t) = \exp(t\mathbf{S})$. Using the right translation, a geodesic emanating from an arbitrary $\mathbf{W} \in U(n)$ in the direction $\mathbf{S} \mathbf{W}$ is given by $\mathcal{G}_{\mathbf{W}}(t) = \exp(t\mathbf{S})\mathbf{W}$, $\mathbf{S} \mathbf{W} \in T_{\mathbf{W}} U(n)$, $t \in \mathbb{R}$. Consequently, the tangent direction $\mathbf{S} \in \mathfrak{u}(n)$ is transported along the geodesic to \mathbf{W} and the resulting tangent vector is $\mathbf{S} \mathbf{W} \in T_{\mathbf{W}} U(n)$. Conversely, if $\mathbf{X} \in T_{\mathbf{W}} U(n)$, then $\mathbf{X} \mathbf{W}^H \in \mathfrak{u}(n)$.

2.2. Steepest descent algorithm on $U(n)$

The unitary optimization can be solved in an iterative manner, by using a steepest descent algorithm along geodesics on $U(n)$. The corresponding rotational update at iteration k is given by:

$$\mathbf{W}_{k+1} = \exp(-\mu_k \mathbf{G}_k) \mathbf{W}_k, \quad k = 0, 1, \dots \quad (2)$$

where $\mathbf{G}_k \triangleq \tilde{\nabla} \mathcal{J}(\mathbf{W}) \mathbf{W}^H = \Gamma_{\mathbf{W}} \mathbf{W}_k^H - \mathbf{W}_k \Gamma_{\mathbf{W}}^H \in \mathfrak{u}(n)$ is the Riemannian gradient of $\mathcal{J}(\mathbf{W})$ at \mathbf{W}_k translated to the group identity,

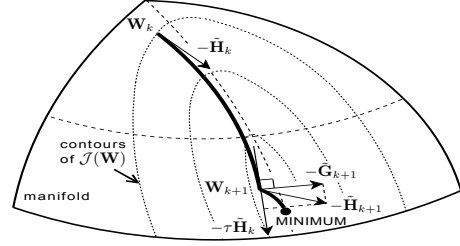


Fig. 2. The CG takes a search direction $-\tilde{\mathbf{H}}_{k+1}$ at \mathbf{W}_{k+1} which is a combination of the new SD direction $-\tilde{\mathbf{G}}_{k+1}$ at \mathbf{W}_{k+1} and the current search direction $-\tilde{\mathbf{H}}_k$ translated to \mathbf{W}_{k+1} along the geodesic connecting \mathbf{W}_k and \mathbf{W}_{k+1} . The new Riemannian steepest descent direction $-\tilde{\mathbf{G}}_{k+1}$ at \mathbf{W}_{k+1} will be orthogonal to the current search direction $-\tilde{\mathbf{H}}_k$ at \mathbf{W}_k translated to \mathbf{W}_{k+1} , i.e., $\langle -\tilde{\mathbf{G}}_{k+1}, -\tau\tilde{\mathbf{H}}_k \rangle_{\mathbf{W}_{k+1}} = 0$.

1	Initialization: $k = 0, \mathbf{W}_k = \mathbf{I}$
2	Compute the Riemannian gradient direction \mathbf{G}_k : $\Gamma_k = \frac{\partial \mathcal{J}}{\partial \mathbf{W}^*}(\mathbf{W}_k)$, $\mathbf{G}_k = \Gamma_k \mathbf{W}_k^H - \mathbf{W}_k \Gamma_k^H$
3	Evaluate $\langle \mathbf{G}_k, \mathbf{G}_k \rangle_{\mathbf{I}} = (1/2) \text{trace} \{ \mathbf{G}_k^H \mathbf{G}_k \}$. If it is sufficiently small, then stop
4	Determine $\mu_k = \arg \min_{\mu} \mathcal{J}(\exp(-\mu \mathbf{G}_k) \mathbf{W}_k)$
5	Update: $\mathbf{W}_{k+1} = \exp(-\mu_k \mathbf{G}_k) \mathbf{W}_k$
6	$k := k + 1$ and go to step 2

Table 1. Steepest descent (SD) algorithm along geodesics on $U(n)$

and $\Gamma_{\mathbf{W}} = \frac{\partial \mathcal{J}}{\partial \mathbf{W}^*}(\mathbf{W}_k)$ is the Euclidean gradient at \mathbf{W}_k . The notation $\exp(\cdot)$ stands for the matrix exponential. The rotational update (2) maintains \mathbf{W}_{k+1} unitary at each iteration. The step size $\mu_k > 0$ controls the convergence speed and needs to be computed at each iteration. In [1], Armijo step size rule [7] is efficiently used. The step size evolves in a dyadic basis. Therefore, when doubling the step size, only a matrix squaring is needed instead of computing a new matrix exponential. In this way the complexity is reduced approximately by half [1], compared to the SD in [5] using also the Armijo rule. Other approaches from the Euclidean space [7] may also be adapted. The proposed SD algorithm is summarized in Table 1.

2.3. Conjugate gradient algorithm on $U(n)$

The Conjugate Gradient (CG) algorithm provides typically faster convergence compared to the Steepest Descent (SD) algorithm not only on the Euclidean space, but also on Riemannian manifolds. This is due to the fact that the Riemannian SD algorithm suffers from the same deficiency as its Euclidean counterpart, i.e., it takes ninety degree turns at each iteration [3]. This fact is illustrated in the left plot of Figure 1 by plotting the cost function level sets on the manifold surface. The conjugate gradient algorithm may significantly reduce this drawback. Moreover, CG provides an inexpensive alternative to Newton algorithm. The new search direction is chosen to be a combination of the current search direction at \mathbf{W}_k and the gradient at the next point \mathbf{W}_{k+1} , as illustrated in Figure 2. The difference compared to the Euclidean space is that the two vectors lie in different tangent spaces. For this reason they are not directly compatible. The fact that $U(n)$ is a Lie group enables simple parallel translation of tangent vectors from a tangent space to another. It is desirable to translate all the tangent directions (steepest descent and search directions) to the same tangent space. Due to compu-

1	Initialization: $k = 0, \mathbf{W}_k = \mathbf{I}$
2	Compute the Riemannian gradient direction \mathbf{G}_k and the search direction \mathbf{H}_k : if $(k \text{ modulo } n^2) == 0$ $\mathbf{\Gamma}_k = \frac{\partial \mathcal{J}}{\partial \mathbf{W}^*}(\mathbf{W}_k)$ $\mathbf{G}_k = \mathbf{\Gamma}_k \mathbf{W}_k^H - \mathbf{W}_k \mathbf{\Gamma}_k^H$ $\mathbf{H}_k := \mathbf{G}_k$ else $\mathbf{G}_k := \mathbf{G}_{k+1}$ $\mathbf{H}_k := \mathbf{H}_{k+1}$
3	Evaluate $\langle \mathbf{G}_k, \mathbf{G}_k \rangle_{\mathbf{I}} = (1/2)\text{trace}\{\mathbf{G}_k^H \mathbf{G}_k\}$. If it is sufficiently small, then stop
4	Determine $\mu_k = \arg \min_{\mu} \mathcal{J}(\exp(-\mu \mathbf{H}_k) \mathbf{W}_k)$
5	Update: $\mathbf{W}_{k+1} = \exp(-\mu_k \mathbf{H}_k) \mathbf{W}_k$
6	Compute the Riemannian gradient direction \mathbf{G}_{k+1} and the search direction \mathbf{H}_{k+1} : $\mathbf{\Gamma}_{k+1} = \frac{\partial \mathcal{J}}{\partial \mathbf{W}^*}(\mathbf{W}_{k+1})$ $\mathbf{G}_{k+1} = \mathbf{\Gamma}_{k+1} \mathbf{W}_{k+1}^H - \mathbf{W}_{k+1} \mathbf{\Gamma}_{k+1}^H$ $\gamma_k = \frac{\langle \mathbf{G}_{k+1} - \mathbf{G}_k, \mathbf{G}_{k+1} \rangle_{\mathbf{I}}}{\langle \mathbf{G}_k, \mathbf{G}_k \rangle_{\mathbf{I}}}$ $\mathbf{H}_{k+1} = \mathbf{G}_{k+1} + \gamma_k \mathbf{H}_k$
7	$k := k + 1$ and go to step 2

Table 2. Conjugate gradient algorithm along geodesics on $U(n)$ using the Polak-Ribière formula (CG-PR)

tational reasons, the tangent space at the group identity element is preferred [6]. Then, all the tangent vectors belong to the Lie algebra $\mathfrak{u}(n)$ and they are represented by skew-Hermitian matrices. The computation of the exponential of skew-Hermitian matrices and its approximations have been thoroughly studied in the literature, see [1]. The new search direction translated into $\mathfrak{u}(n)$ is

$$\mathbf{H}_{k+1} = \mathbf{G}_{k+1} + \gamma_k \mathbf{H}_k, \quad \mathbf{H}_k, \mathbf{H}_{k+1}, \mathbf{G}_{k+1} \in \mathfrak{u}(n) \quad (3)$$

where \mathbf{H}_k is the old search direction at \mathbf{W}_k , translated into $\mathfrak{u}(n)$. The weighting factor γ_k may be determined for example, by using the Polak-Ribière formula $\gamma_k = \langle \mathbf{G}_{k+1} - \mathbf{G}_k, \mathbf{G}_{k+1} \rangle_{\mathbf{I}} / \langle \mathbf{G}_k, \mathbf{G}_k \rangle_{\mathbf{I}}$ [3]. The conjugate gradient step is taken along the geodesic emanating from \mathbf{W}_k in the direction $-\mathbf{H}_k = -\mathbf{H}_k \mathbf{W}_k$, i.e.,

$$\mathbf{W}_{k+1} = \exp(-\mu_k \mathbf{H}_k) \mathbf{W}_k, \quad k = 0, 1, \dots \quad (4)$$

Analogous to the Euclidean CG, it is desirable to reset the search direction $-\mathbf{H}_k$ to the gradient direction $-\mathbf{G}_k$ after each n^2 steps, which is the dimension of $U(n)$. This may enhance the convergence speed. The proposed CG algorithm on $U(n)$ using the Polak-Ribière formula is summarized in Table 2.

Remarks: The SD algorithm in Table 1 is designed to minimize a cost function. It may be converted into a steepest ascent (SA) algorithm for solving maximization problems, by changing the update step 5 into $\mathbf{W}_{k+1} = \exp(+\mu_k \mathbf{H}_k) \mathbf{W}_k$. The same change needs to be applied to the CG-PR in Table 2, step 5, in order to solve maximization problems. Additionally, the step 4 in Table 2 needs to be replaced by $\mu_k = \arg \max_{\mu} \mathcal{J}(\exp(+\mu \mathbf{H}_k) \mathbf{W}_k)$.

3. SIMULATION RESULTS AND APPLICATIONS

In this section we test the proposed Riemannian algorithms on two different optimization problems on $U(n)$. The first one is a classical test function for optimization under orthogonal matrix constraint [3]. The second one is the JADE cost function [8] which is a practical application of the proposed algorithms to blind source separation.

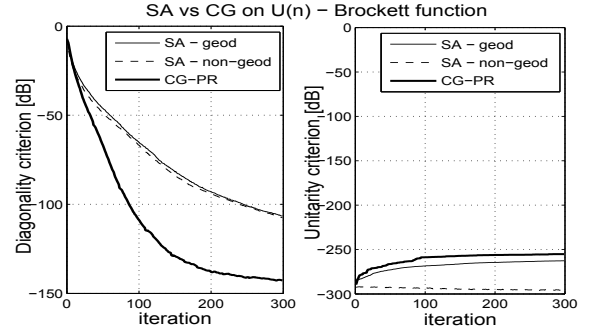


Fig. 3. Comparison between different SA (steepest ascent) and CG algorithms: the geodesic SA obtained for the SD in Table 1, the non-geodesic SA as in [5] and the CG-PR algorithm given in Table 2, which uses the Polak-Ribière formula (CG-PR). The geodesic and the non-geodesic SA algorithms perform the same, but the geodesic SA algorithm has lower complexity. The CG provides faster convergence compared to the geodesic SA at comparable complexity.

3.1. Diagonalization of a Hermitian Matrix

The diagonalization of a Hermitian matrix Σ can be achieved by maximizing the Brockett criterion [3]

$$\mathcal{J}_B(\mathbf{W}) = \text{tr}\{\mathbf{W}^H \Sigma \mathbf{W} \mathbf{N}\}, \quad \text{subject to } \mathbf{W} \in U(n). \quad (5)$$

The matrix \mathbf{W} converges to the eigenvectors of Σ sorted according to the ascending order of the eigenvalues, provided that \mathbf{N} is a diagonal matrix with the diagonal elements $1, \dots, n$. This type of optimization problem arises in many signal processing applications such as blind source separation, subspace estimation, high resolution direction finding as well as in communications applications. This example of computing the eigenvectors of a Hermitian matrix (such as a covariance matrix) is chosen for illustrative purposes since it is well known by most of the readers. A more practical application is the blind source separation problem is considered in Subsection 3.2. The Euclidean gradient of the Brockett function is $\mathbf{\Gamma}_{\mathbf{W}} = \Sigma \mathbf{W} \mathbf{N}$. The performance is studied in terms of convergence speed considering a *diagonality criterion*, Δ , and in terms of deviation from the unitary constraint using a *unitarity criterion* Ω , defined as

$$\Delta = 10 \lg \frac{\text{off}\{\mathbf{W}^H \Sigma \mathbf{W}\}}{\text{diag}\{\mathbf{W}^H \Sigma \mathbf{W}\}}, \quad \Omega = 10 \lg \|\mathbf{W} \mathbf{W}^H - \mathbf{I}\|_F^2, \quad (6)$$

where $\text{off}\{\cdot\}$ operator computes the sum of the squared magnitudes of the off-diagonal elements of a matrix, and $\text{diag}\{\cdot\}$ does the same operation, but for the off-diagonal ones. The diagonality criterion measures the departure of the matrix $\mathbf{W}^H \Sigma \mathbf{W}$ from the diagonal property, in logarithmic scale. The unitarity criterion is the squared Frobenius norm of the deviation from the unitarity property in a logarithmic scale. The results are averaged over 100 random realizations of the 6×6 Hermitian matrix Σ . In order to maximize the criterion (5), the SD in Table 1 and CG-PR in Table 2 need minor modifications (see *Remarks* at the end of Subsection 2.3). In Figure 3, we compare three algorithms. The first one is the geodesic steepest ascent (SA) obtained from the SD in Table 1. The second algorithm is the non-geodesic SA obtained from SD in [5]. The third one is the CG algorithm in Table 2 which uses the Polak-Ribière formula (CG-PR). The step size for all three algorithms is chosen by using the Armijo rule [7] as in [1]. In the left plot of Figure 3 we observe that the geodesic and the non-geodesic SA algorithms perform the

same, but the geodesic SA algorithm has lower complexity, especially when Armijo step is used [1]. The CG-PR algorithm outperforms both the geodesic and the non-geodesic SA algorithms, with comparable computational complexity. In terms of satisfying the unitary constraint, all algorithms provide good performance as it is shown in the right subplot of Figure 3.

3.2. Joint Approximate Diagonalization of a set of Hermitian Matrices. Minimizing the JADE criterion on $U(m)$

In this subsection we test the proposed SD and CG algorithms in a practical application of blind source separation (BSS) of communication signals by using a joint diagonalization approach [8]. We show that the proposed algorithms outperform the classical JADE algorithm [8]. A number of $m = 20$ independent 16-QAM signals are separated blindly from their $r = 20$ mixtures. A total of $N = 15000$ snapshots are collected and the results are averaged over 100 independent realizations of the $r \times m$ mixture matrix. The signal-to-noise-ratio is $20dB$. The blind recovery of the desired signals is based on statistical properties and it is done in two stages. The first one is a whitening operation, which can be done by diagonalizing the sample covariance matrix as in Subsection 3.1. The second stage is the joint diagonalization of a set of eigenmatrices $\hat{\mathbf{M}}_i$ which are estimated from the fourth order cumulants of the whitened signals. In [8], this is done by using Givens rotations. In this paper we find the unitary rotation by minimizing the JADE criterion [8] which penalizes the deviation of eigen-matrices from the diagonal property.

$$\mathcal{J}_{\text{JADE}}(\mathbf{W}) = \sum_{i=1}^m \text{off}\{\mathbf{W}^H \hat{\mathbf{M}}_i \mathbf{W}\} \quad \text{subject to } \mathbf{W} \in U(n). \quad (7)$$

The gradient of the JADE cost function on the Euclidean space is $\Gamma_{\mathbf{W}} = 2 \sum_{i=1}^m \hat{\mathbf{M}}_i \mathbf{W} [\mathbf{W}^H \hat{\mathbf{M}}_i \mathbf{W} - \mathbf{I} \odot (\mathbf{W}^H \hat{\mathbf{M}}_i \mathbf{W})]$, where \odot denotes the elementwise matrix multiplication.

In Figure 4 we compare the classical JADE algorithm to the proposed SD and CG algorithms on $U(m)$. The performance measure for the optimization problem is the JADE criterion (7). The performance index used for the entire blind separation problem is the Amari distance. The geodesic SD and the CG-PR algorithm have similar convergence speed and they outperform the classical JADE algorithm [8]. The non-geodesic SD [5] (not shown in Figure 4) performs the same as the geodesic SD. The geodesic algorithms on $U(m)$ take benefit of the Lie group properties of $U(m)$ in order to reduce complexity. The proposed SD and CG-PR algorithms have complexity of $\mathcal{O}(m^3)$ per iteration and only few iterations are required to achieve convergence. Moreover, the number of iterations needed to achieve convergence stays almost constant when increasing m . The Givens rotation approach in [8] has a total complexity of $\mathcal{O}(m^4)$, since it updates not only the unitary rotation matrix, but also the full set of eigen-matrices $\hat{\mathbf{M}}_i$. Therefore, the total complexity of the proposed algorithms is lower, especially when the number of signals m is very large. The proposed algorithms converge faster at similar computational cost/per iteration. Therefore, they are suitable for blind separation applications, especially when the number of signals to be separated is large.

4. CONCLUSIONS

In this paper, Riemannian steepest descent and conjugate gradient algorithms for optimization under unitary matrix constraint were proposed. They operate on the Lie group of $n \times n$ unitary matrices and it exploits the geometrical properties of $U(n)$, such as simple geodesic

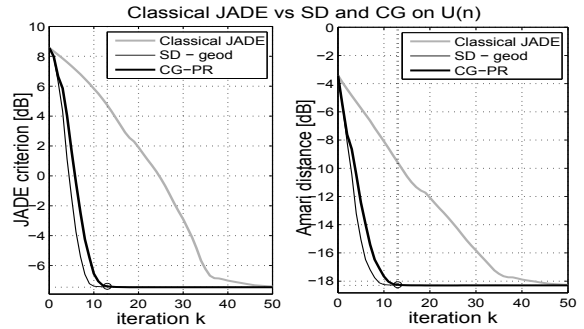


Fig. 4. The classical JADE algorithm [8] vs. SD and CG algorithms on $U(20)$: the geodesic SD in Table 1 and the CG-PR in Table 2. The geodesic SD and the CG-PR algorithm perform similarly in terms of JADE criterion and Amari distance. They outperform the classical JADE algorithm [8], especially when the number of signal sources is large. In this application the CG did not improve convergence speed of the SD.

formulas and parallel transport in order to reduce the complexity. For this reason their complexity is lower than the non-geodesic SD in [5] at the same convergence speed. The algorithms provide a reliable solution to the joint diagonalization problem for blind separation and outperforms the widely used Givens rotations approach, i.e., in the classical JADE algorithm [8]. It may be applied, for example, to smart antenna algorithms, wireless communications, biomedical measurements, signal separation, subspace estimation and tracking tasks where unitary matrices play an important role in general.

5. REFERENCES

- [1] T. Abrudan, J. Eriksson, and V. Koivunen, "Steepest descent algorithms for optimization under unitary matrix constraint," *to appear in IEEE Transactions on Signal Processing*.
- [2] D. Gabay, "Minimizing a differentiable function over a differential manifold," *Journal of Optimization Theory and Applications*, vol. 37, pp. 177–219, June 1982.
- [3] S. T. Smith, *Geometric optimization methods for adaptive filtering*. PhD thesis, Harvard University, Cambridge, MA, May 1993.
- [4] A. Edelman, T. Arias, and S. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.
- [5] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Transactions on Signal Processing*, vol. 50, pp. 635–650, Mar. 2002.
- [6] T. Abrudan, J. Eriksson, and V. Koivunen, "Conjugate gradient algorithm for optimization under unitary matrix constraint," *submitted to IEEE Transactions on Signal Processing*.
- [7] E. Polak, *Optimization: Algorithms and Consistent Approximations*. New York: Springer-Verlag, 1997.
- [8] J. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proceedings-F*, vol. 140, no. 6, pp. 362–370, 1993.