Conjugate Gradient Algorithm for Optimization Under Unitary Matrix Constraint[1]

Traian Abrudan, Jan Eriksson, and Visa Koivunen

Abstract

In this paper we introduce a Riemannian algorithm for minimizing (or maximizing) a real-valued function $\mathcal{J}$ of complex-valued matrix argument $\mathbf{W}$ under the constraint that $\mathbf{W}$ is an $n \times n$ unitary matrix. This type of constrained optimization problem arises in many array and multi-channel signal processing applications.

We propose a conjugate gradient (CG) algorithm on the Lie group of unitary matrices $U(n)$. The algorithm fully exploits the group properties in order to reduce the computational cost. Two novel geodesic search methods exploiting the almost periodic nature of the cost function along geodesics on $U(n)$ are introduced.

We demonstrate the performance of the proposed conjugate gradient algorithm in a blind signal separation application. Computer simulations show that the proposed algorithm outperforms other existing algorithms in terms of convergence speed and computational complexity.

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

1 Introduction

Constrained optimization problems arise in many signal processing applications. In particular, we are addressing the problem of optimization under unitary matrix constraint. Such problems may be found in communications and array signal processing, for example, blind and constrained beamforming, high-resolution direction finding, and generally in all subspace-based

---

methods. 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. See [1, 2], and [3] for recent reviews.

Commonly, optimization under unitary matrix constraint is viewed as a constrained optimization problem on the Euclidean space. Classical gradient algorithms are combined with different techniques for imposing the constraint, for example, orthogonalization, approaches stemming from the Lagrange multipliers method, or some stabilization procedures. If unitarity is enforced by using such techniques, one may experience slow convergence or departures from the unitary constraint as shown in [1].

A constrained optimization problem may be converted into an unconstrained one on a different parameter space determined by the constrained set. The unitary matrix constraint considered in this paper determines a parameter space which is the Lie group of $n \times n$ unitary matrices $U(n)$. This parameter space is a Riemannian manifold [4] and a matrix group [5] under the standard matrix multiplication at the same time. By using modern tools of Riemannian geometry, we take full benefit of the nice geometrical properties of $U(n)$ in order to solve the optimization problem efficiently and satisfy the constraint with high fidelity at the same time.

Pioneering work in Riemannian optimization may be found in [6–9]. The optimization with orthogonality constraints is considered in detail in [10]. Steepest descent (SD), conjugate gradient (CG) and Newton algorithms on Stiefel and Grassman manifolds are derived. A CG algorithm on the Grassmann manifold has also been proposed recently in [11]. A non-Riemannian approach, which is a general framework for optimization, is introduced in [12]. Modified SD and Newton algorithms on Stiefel and Grassman manifolds are derived. SD algorithms operating on orthogonal group are considered recently in [13–15] and on the unitary group in [1,2]. A CG on the special linear group is proposed in [16]. Algorithms in the existing literature [6–8,10,12,17] are, however, more general in the sense that they can be applied on more general manifolds than $U(n)$. For this reason when applied to $U(n)$, they do not take full benefit of the special properties arising from the Lie group structure of the manifold [1].

In this paper we derive a conjugate gradient algorithm operating on the Lie group of unitary matrices $U(n)$. The proposed CG algorithm provides faster convergence compared to the existing SD algorithms [1, 12] at even lower complexity. There are two main contributions in this paper. First, a computationally efficient CG algorithm on the Lie group of unitary matrices $U(n)$ is proposed. The algorithm fully exploits the Lie group features such as simple formulas for the geodesics and tangent vectors.

The second main contribution in this paper is that we propose Riemannian optimization algorithms which exploit the *almost periodic* property of the cost function along geodesics on $U(n)$. Based on this property we derive novel high accuracy *line search methods* [8] that facilitate fast convergence and selection of suitable step size parameter. Many of the existing geometric optimization algorithms do not include practical line search methods [10,13], or if they do, they are too complex when applied to optimization on $U(n)$ [12,14]. In some cases, the line search methods are either valid only for specific cost functions [8], or the resulting search is not highly accurate [1,11,12,14,15]. Because the conjugate gradient algorithm assumes exact search along geodesics, the line search method is crucial for the performance of the resulting algorithm. An accurate line search method exploiting the periodicity of a cost function which appears on the limited case of special orthogonal group $SO(n)$ for $n \leq 3$, is proposed in [15] for non-negative ICA (independent component analysis). The method can also be applied for $n > 3$, but the accuracy decreases, since the periodicity of the cost function is lost.

The proposed high-accuracy line search methods have lower complexity compared to well-known efficient methods [1] such as the Armijo method [18]. To our best knowledge the proposed algorithm is the first ready-to-implement CG algorithm on the Lie group of unitary matrices $U(n)$. It is also valid for the orthogonal group $O(n)$.

This paper is organized as follows. In Section 2 we approach the problem of optimization under unitary constraint by using tools from Riemannian geometry. We show how the geometric properties may be exploited in order to solve the optimization problem in an efficient way. Two novel line search methods are introduced in Section 3. The practical conjugate gradient algorithm for optimization under unitary matrix constraint is given in Section 4. Simulation results and applications are presented in Section 5. Finally, Section 6 concludes the paper.

2 Optimization on the Unitary Group $U(n)$

In this section we show how the problem of optimization under unitary matrix constraint can be solved efficiently and in an elegant manner by using tools of Riemannian geometry. In Subsection 2.1 we review some important properties of the unitary group $U(n)$ which are needed later in our derivation. Few important properties of $U(n)$ that are very beneficial in optimization are pointed out in Subsection 2.2. The difference in behavior of the steepest

descent and conjugate gradient algorithm on Riemannian manifolds in explained in Subsection 2.3. A generic conjugate gradient algorithm on $U(n)$ is proposed in Subsection 2.4.

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 $U(n)$ and show how they can be exploited in optimization algorithms. Consider a real-valued function $\mathcal{J}$ of an $n \times n$ complex matrix $\mathbf{W}$, i.e., $\mathcal{J} : \mathbb{C}^{n \times n} \to \mathbb{R}$. Our goal is to minimize (or maximize) the function $\mathcal{J} = \mathcal{J}(\mathbf{W})$ under the constraint that $\mathbf{W}\mathbf{W}^H = \mathbf{W}^H\mathbf{W} = \mathbf{I}$, i.e., $\mathbf{W}$ is unitary. This constrained optimization problem on $\mathbb{C}^{n \times n}$ may be converted into an unconstrained one on the space determined by the unitary constraint, i.e., the Lie group of unitary matrices. We view our cost function $\mathcal{J}$ as a function defined on $U(n)$. The space $U(n)$ is a *real differential manifold* [4]. Moreover, the unitary matrices are closed under the standard matrix multiplication, i.e., they form a Lie group [5]. The additional properties arising from the Lie group structure may be exploited to reduce the complexity of the optimization algorithms.

2.1.1 Tangent vectors and tangent spaces

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

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

After $U(n)$ is equipped with a Riemannian structure (metric), the Riemannian gradient on $U(n)$ can be defined. The inner product given by

$$\langle \mathbf{X}, \mathbf{Y} \rangle_\mathbf{W} = \frac{1}{2}\Re\{\text{trace}\{\mathbf{X}\mathbf{Y}^H\}\}, \ \mathbf{X}, \mathbf{Y} \in T_\mathbf{W}U(n) \tag{1}$$

induces a bi-invariant metric on $U(n)$. This metric induced from the embedding Euclidean space is also the natural metric on the Lie group, since it equals the negative (scaled) Killing form [19]. The Riemannian gradient gives the steepest ascent direction on $U(n)$ of the function $\mathcal{J}$ at some point $\mathbf{W} \in U(n)$:

$$\tilde{\nabla}\mathcal{J}(\mathbf{W}) \triangleq \mathbf{\Gamma}_\mathbf{w} - \mathbf{W}\mathbf{\Gamma}_\mathbf{w}^H\mathbf{W}, \tag{2}$$

4

where $\mathbf{\Gamma_W} = \frac{d\mathcal{J}}{d\mathbf{W}^*}(\mathbf{W})$ is the gradient of $\mathcal{J}$ on the Euclidean space at a given $\mathbf{W}$, defined in terms of real derivatives with respect to real and imaginary parts of $\mathbf{W}$ [20].

2.1.3 Geodesics and parallel transport on $U(n)$

Geodesics on the Riemannian manifold are curves such that their second derivative lies in the normal space. Locally, they are also the length minimizing curves. Their expression is given by the Lie group exponential map [19, Ch. II, §1]. For $U(n)$ this coincides with the matrix exponential, for which stable and efficient numerical methods exist (see [1] for details). The geodesic emanating from $\mathbf{W}$ in the direction $\tilde{\mathbf{S}} = \mathbf{SW}$ is given by

$$\mathcal{G}_\mathbf{W}(t) = \exp(t\mathbf{S})\mathbf{W}, \quad \mathbf{S} \in \mathfrak{u}(n), \quad t \in \mathbb{R}. \tag{3}$$

The parallelism on a differentiable manifold is defined with respect to an affine connection [4, 19, 21]. Usually the Levi-Civita connection [8, 17] is used on Riemannian manifolds. Now, the parallel transport of a tangent vector $\tilde{\mathbf{X}} = \mathbf{XW} \in T_\mathbf{W}$, $\mathbf{X} \in \mathfrak{u}(n)$, w.r.t. the Riemannian connection along the geodesic (3) is given by [19]

$$\tau\tilde{\mathbf{X}}(t) = \exp(t\mathbf{S}/2)\mathbf{X}_k \exp(-t\mathbf{S}/2)\mathcal{G}_\mathbf{W}(t), \tag{4}$$

where τ denotes the parallel transport. In the important special case of transporting the velocity vector of the geodesic, i.e., $\tilde{\mathbf{X}} = \tilde{\mathbf{S}}$, Eq. (4) simply reduces to the right multiplication by $\mathbf{W}^H\mathcal{G}_\mathbf{W}(t)$:

$$\tau\tilde{\mathbf{S}}(t) = \mathbf{S}\mathcal{G}_\mathbf{W}(t) = \tilde{\mathbf{S}}\mathbf{W}^H\mathcal{G}_\mathbf{W}(t). \tag{5}$$

2.2 Almost periodic cost function along geodesics

In this subsection we present an important property of the unitary group, i.e., the behavior of a smooth cost function along geodesics on $U(n)$. A smooth cost function $\mathcal{J} : U(n) \to \mathbb{R}$ takes predictable values along geodesics on $U(n)$. This is a consequence of the fact that geodesics (3) are given by matrix exponential of skew-Hermitian matrices $\mathbf{S} \in \mathfrak{u}(n)$. Such matrices have purely imaginary eigenvalues of form $\jmath\omega_i, i = 1, \ldots, n$. Therefore, the eigenvalues of the matrix exponential $\exp(t\mathbf{S})$ are complex exponentials of form $e^{\jmath\omega_i t}$. Consequently, the composed cost function $\hat{\mathcal{J}}(t) = \mathcal{J}(\mathcal{G}_\mathbf{W}(t))$ is an *almost periodic function*, and therefore it may be expressed as a sum of periodic functions of t. $\hat{\mathcal{J}}(t)$ is a periodic function only if the frequencies $\omega_i, i = 1, \ldots, n$ are in harmonic relation [22]. This happens for $SO(2)$ and

5

$SO(3)$ as noticed in [15], but not for $U(n)$ with $n > 1$, in general. The derivatives of the cost function are also almost periodic functions. The almost periodic property of the cost function and its derivatives appears in the case of exponential map. This is not the case for other common parametrizations such as the Cayley transform or the Euclidean projection operator [1]. Moreover, this special property appears only on certain manifolds such as the unitary group $U(n)$ and the special orthogonal group $SO(n)$, and it does not appear on Euclidean spaces or on general Riemannian manifolds. The almost-periodic behavior of the cost function along geodesics may be used to perform geodesic search on $U(n)$. This will be shown later in Section 3 where two novel line search methods for selecting a suitable step size parameter are introduced.

2.3   SD vs. CG on Riemannian manifolds

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 Riemannian SD algorithm has the same drawback as its Euclidean counterpart, i.e., it takes ninety degree turns at each iteration [8]. This is illustrated in Figure 1 (top), where the contours of a cost function are plotted on the manifold surface. The steps are taken along geodesics, i.e., the trajectory of the SD algorithm is comprised of geodesic segments connecting successive points $\mathbf{W}_k, \mathbf{W}_{k+1}, \mathbf{W}_{k+2}$ on the manifold. The zig-zag type of trajectory decreases the convergence speed, e.g., if the cost function has the shape of a "long narrow valley". The conjugate gradient algorithm may significantly reduce this drawback Figure 1 (bottom). It exploits the information provided by the current search direction $-\tilde{\mathbf{H}}_k$ at $\mathbf{W}_k$ and the SD direction $-\tilde{\mathbf{G}}_{k+1}$ at the next point $\mathbf{W}_{k+1}$. The new search direction is chosen to be a combination of these two, as shown in Figure 1 (bottom). The difference compared to the Euclidean space is that the current search direction $-\tilde{\mathbf{H}}_k$ and the gradient $\tilde{\mathbf{G}}_{k+1}$ at the next point lie in different tangent spaces, $T_{\mathbf{W}_k}$ and $T_{\mathbf{W}_{k+1}}$, respectively. For this reason they are not directly compatible. In order to combine them properly, the parallel transport of the current search direction $-\tilde{\mathbf{H}}_k$ from $\mathbf{W}_k$ to $\mathbf{W}_{k+1}$ along the corresponding geodesic is utilized. The new search direction at $\mathbf{W}_{k+1}$ (see Figure 1 (bottom)) is

$$-\tilde{\mathbf{H}}_{k+1} = -\tilde{\mathbf{G}}_{k+1} - \gamma_k \tau \tilde{\mathbf{H}}_k, \qquad (6)$$

where $\tau \tilde{\mathbf{H}}_k$ is the parallel transport (5) of the vector $\tilde{\mathbf{H}}_k$ into $T_{\mathbf{W}_{k+1}}$. The weighting factor γ_k is determined such that the directions $\tau \tilde{\mathbf{H}}_k$ and $\tilde{\mathbf{H}}_{k+1}$ are
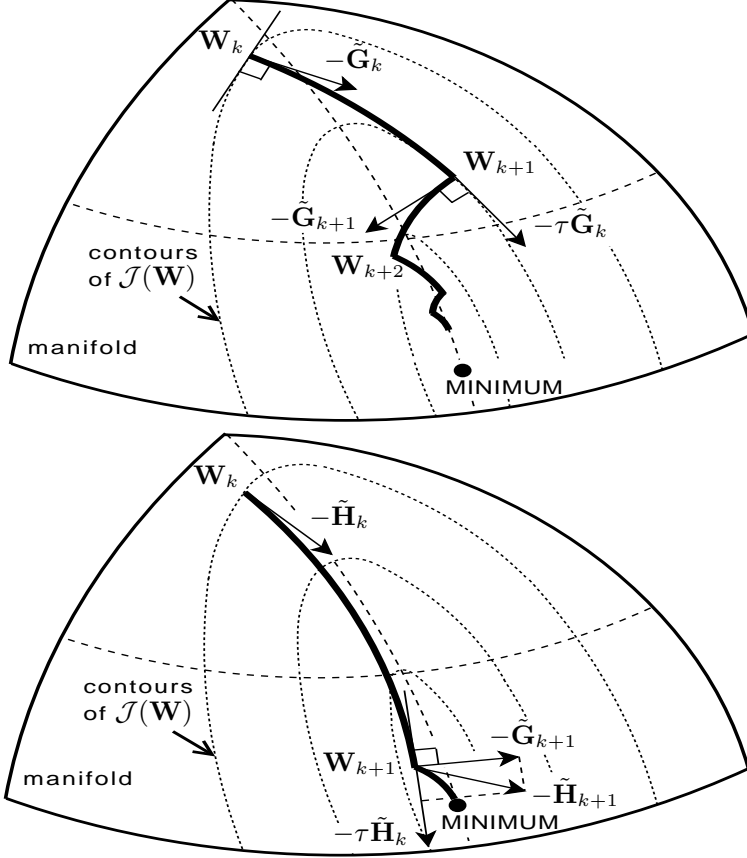
6

Figure 1: SD (top) vs. CG (bottom) on Riemannian manifolds. 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 $\mathbf{W}_{k+1}$. 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 $-\mathbf{H}_k$ at $\mathbf{W}_k$ translated to $\mathbf{W}_{k+1}$, i.e., $\langle -\tilde{\mathbf{G}}_{k+1},\ -\tau\tilde{\mathbf{H}}_k \rangle_{\mathbf{W}_{k+1}} = 0$.

Hessian-conjugate [8, 10], i.e.,

$$\gamma_k = -\frac{\text{Hess } \mathcal{J}(\tau\tilde{\mathbf{H}}_k, \tilde{\mathbf{G}}_{k+1})}{\text{Hess } \mathcal{J}(\tau\tilde{\mathbf{H}}_k, \tau\tilde{\mathbf{H}}_k)}. \tag{7}$$

2.4 Conjugate Gradient Algorithm on $U(n)$

In the exact conjugacy formula (7), only the special case of vector transportation (5) is needed. However, the factor γ_k contains the computationally expensive Hessian, and therefore, as usual, it is approximated. Using the crucial assumption that $\mathbf{W}_{k+1}$ is a minimum point along the geodesic $\boldsymbol{\mathcal{G}}_{\mathbf{W}}(t)$ and the first-order Taylor series approximation of the first differential form of $\mathcal{J}(\mathbf{W}_{k+1})$, the Polak-Ribièrre approximation formula for the factor γ_k is obtained [8, 10]:

$$\gamma_k = \frac{\langle \tilde{\mathbf{G}}_{k+1} - \tau\tilde{\mathbf{G}}_k, \tilde{\mathbf{G}}_{k+1} \rangle_{\mathbf{W}_{k+1}}}{\langle \tilde{\mathbf{G}}_k, \tilde{\mathbf{G}}_k \rangle_{\mathbf{W}_k}}. \tag{8}$$

The parallel transport for $\tilde{\mathbf{G}}_k$ could be obtained from (4), but we choose to approximate it by using the right multiplication by $\mathbf{W}_k^H \mathbf{W}_{k+1}$ leading to the approximate Polak-Ribièrre formula.

$$\gamma_k = \frac{\langle \mathbf{G}_{k+1} - \mathbf{G}_k, \mathbf{G}_{k+1} \rangle_{\mathbf{I}}}{\langle \mathbf{G}_k, \mathbf{G}_k \rangle_{\mathbf{I}}} \tag{9}$$

If $\tilde{\mathbf{H}}_k = \tilde{\mathbf{G}}_k$, then by (5) formulae (8) and (9) become equal. In our simulations (8) and (9) have given identical results, and therefore we propose the computationally simpler formula (9).

Finally, the conjugate gradient step is taken along the geodesic emanating from $\mathbf{W}_k$ in the direction $-\tilde{\mathbf{H}}_k = -\mathbf{H}_k\mathbf{W}_k$, i.e.,

$$\mathbf{W}_{k+1} = \exp(-\mu_k\mathbf{H}_k)\mathbf{W}_k. \tag{10}$$

A line search needs to be performed in order to find the step size μ_k which corresponds to a local minimum along the geodesic.

3 Line Search on $U(n)$. Step size selection

Step size selection plays a crucial role in the overall performance of the CG algorithm. In general, selecting an appropriate step size may be computationally expensive even for the Euclidean gradient algorithms. This is due

to the fact that most of the line search methods [18] require multiple cost function evaluations. On Riemannian manifolds, every cost function evaluation requires expensive computations of the local parametrization (in our case, the exponential map). In [11], a conjugate gradient on the Grassmann manifold is proposed. The line search method is exact only for the convex quadratic cost functions on the Euclidean space. The difficulty of finding a closed-form solutions for a suitable step size is discussed in [11]. A one-dimensional Newton method which uses the first-order Fourier expansion to approximate the cost function along geodesics on $SO(n)$ is proposed in [15]. It requires computing the first and the second-order derivatives of the cost function along geodesics. The method exploits the *periodicity* of the cost function along geodesics on $SO(2)$ and $SO(3)$. For $n > 3$ the accuracy of the approximation decreases, since the periodicity of the cost functions is lost. The method avoids computing the matrix exponential by using the closed-form Rodrigues formula, valid for $SO(2)$ and $SO(3)$ only. In case of $U(2)$ and $U(3)$ and in general for $n > 1$, the Rodrigues formula cannot be applied.

In this section, we propose two novel methods for performing high-accuracy one-dimensional search along geodesics on $U(n)$. They rely on the fact that smooth functions as well as their derivatives are *almost periodic* [22] along geodesics on $U(n)$. The first method is based on a polynomial approximation of the first-order derivative of the cost function along geodesics. The second one is based on an approximation using Discrete Fourier Transform (DFT). We choose to approximate the derivative of the cost function along geodesics and find the corresponding zeros, instead of approximating the cost function itself and finding the local minima as in [15]. Moreover, compared to [15] the proposed method does not require the second-order derivative.

The main goal is to find a step size $\mu_k > 0$ along the geodesic curve

$$\boldsymbol{\mathcal{W}}(\mu) = \exp(-\mu \mathbf{H}_k)\mathbf{W}_k \triangleq \boldsymbol{\mathcal{R}}(\mu)\mathbf{W}_k, \quad \boldsymbol{\mathcal{R}}(\mu) \subset U(n), \qquad (11)$$

which minimizes the composed function

$$\hat{\mathcal{J}}(\mu) \triangleq \mathcal{J}(\boldsymbol{\mathcal{W}}(\mu)). \qquad (12)$$

The direction $-\mathbf{H}_k \in \mathfrak{u}(n)$ in (11) may correspond to a steepest descent, conjugate gradient, or any other gradient-type of method. Consider two successive points on $U(n)$ such that $\mathbf{W}_k = \boldsymbol{\mathcal{W}}(0)$ and $\mathbf{W}_{k+1} = \boldsymbol{\mathcal{W}}(\mu_k)$. Finding the step size $\mu = \mu_k$ that minimizes $\hat{\mathcal{J}}(\mu)$ may be done by computing the first-order derivative $d\hat{\mathcal{J}}/d\mu$ and setting it to zero. By using the chain rule for the composed function $\mathcal{J}(\boldsymbol{\mathcal{W}}(\mu))$, we get

$$\frac{d\hat{\mathcal{J}}}{d\mu}(\mu) = -2\Re\left\{\text{trace}\left\{\frac{\partial \mathcal{J}}{\partial \boldsymbol{\mathcal{W}}^*}(\boldsymbol{\mathcal{R}}(\mu)\mathbf{W}_k)\,\mathbf{W}_k^H \boldsymbol{\mathcal{R}}^H(\mu)\mathbf{H}_k^H\right\}\right\}. \qquad (13)$$

9

Almost periodicity may be exploited in many ways in order to find the zeros of the first-order derivative corresponding to the desired values of the step size. We present two different approaches. The first approach finds only the first zero of the derivative by using a polynomial approximation of the derivative. The second one finds several zeros of the derivative and is based on a Fourier series approximation. Other approaches may also be possible, and they are being investigated.

3.1 Line search on $U(n)$ by using polynomial approximation approach

The goal of the polynomial approximation approach is to find *the first local minimum* of the cost function along a given geodesic. This corresponds to finding the first zero-crossing value of the first-order derivative of the cost function, which is also almost periodic. In this purpose we use a low-order polynomial approximation of the derivative and find its smallest positive root. The approximation range of the derivative is determined from its spectral content. The method provides computational benefits since *only one* evaluation of the matrix exponential is needed. The method is explained in detail below.

The direction $-\tilde{\mathbf{H}}_k$ is a descent direction at $\mathbf{W}_k$, otherwise it will be reset to the negative gradient. Therefore, the first-order derivative $d\hat{\mathcal{J}}/d\mu$ is always negative at the origin (at $\mu = 0$) and the cost function $\hat{\mathcal{J}}(\mu)$ is monotonically decreasing up to the first zero-crossing of $d\hat{\mathcal{J}}/d\mu$. This value corresponds to a local minimum of $\hat{\mathcal{J}}(\mu)$ along geodesic (11) (or seldom to a saddle point). Due to differentiation, the spectrum of $d\hat{\mathcal{J}}/d\mu$ is the high-pass filtered spectrum of $\hat{\mathcal{J}}(\mu)$. The frequency components are determined by the purely imaginary eigenvalues of $-\mathbf{H}_k$ as shown in Subsection 2.2. Therefore, the cost function as well as its derivative possess discrete frequency spectra. For our task at hand, we are not interested in the complete spectrum of $d\hat{\mathcal{J}}/d\mu$, but the main interest lies in the smallest zero-crossing value of the derivative. This is determined by the highest frequency component in the spectrum of $d\hat{\mathcal{J}}/d\mu$ in the following way. In the interval of μ which is equal to one period corresponding to the highest frequency in the spectrum, the function $d\hat{\mathcal{J}}/d\mu$ has at most one complete cycle on that frequency, and less than one on other frequencies. The highest frequency component of $d\hat{\mathcal{J}}/d\mu$ is $q|\omega_{\max}|$, where $\omega_{\max}$ is the eigenvalue of $\mathbf{H}_k$ having the highest magnitude, and q is the order of the cost function. The order q corresponds to the highest degree that t appears on in the Taylor series expansion of $\mathcal{J}(\mathbf{W} + t\mathbf{Z})$ about $t_0 = 0$, and it is assumed to be finite (most of the practical cost functions).
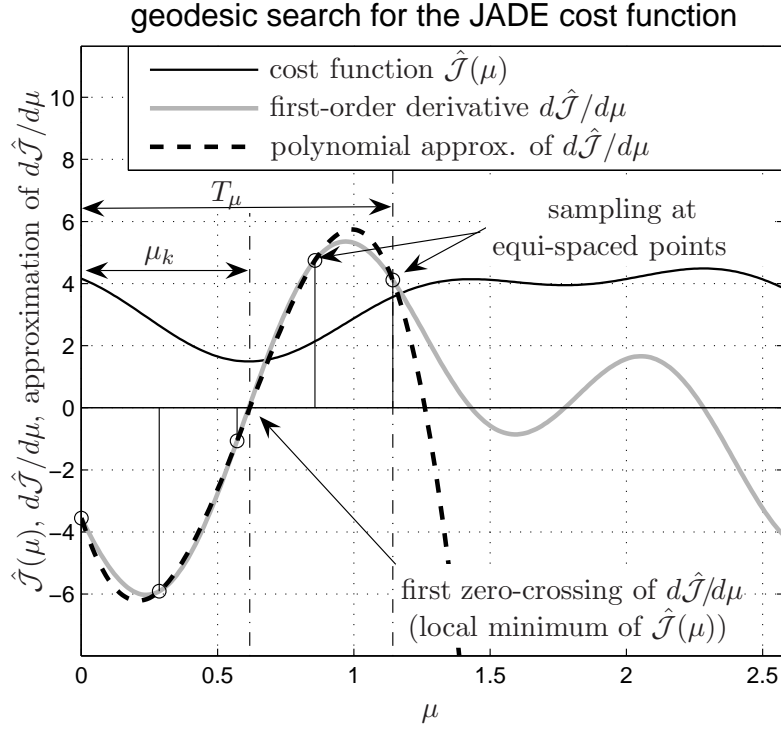
Figure 2: Performing the line search for the JADE [23] cost function. The almost periodic behavior of the function $\hat{\mathcal{J}}(\mu)$ and its first-order derivative $d\hat{\mathcal{J}}/d\mu$ (13) along geodesic $\boldsymbol{\mathcal{W}}(\mu)$ (11). The first zero-crossing of $d\hat{\mathcal{J}}/d\mu$ corresponds to the first local minimum of $\hat{\mathcal{J}}(\mu)$, i.e., the desired step size μ_k. This first zero-crossing value is obtained by using a fourth-order polynomial approximation of $d\hat{\mathcal{J}}/d\mu$ at equi-spaced points within the interval T_μ.

11

Otherwise, a truncated Taylor series may be used. The period corresponding the the highest frequency component is

$$T_\mu = \frac{2\pi}{q|\omega_{\max}|}. \qquad (14)$$

The highest frequency component is amplified the most due to differentiation (high-pass filtering). The other components have less than one cycle within that interval as well as usually smaller amplitudes. Therefore, the first-order derivative $d\hat{\mathcal{J}}/d\mu$ crosses zero at most twice within the interval $[0, T_\mu)$. The presence of the zeros of the derivative are detected as sign changes of the derivative within $[0, T_\mu)$. Since $d\hat{\mathcal{J}}/d\mu$ varies very slowly within the interval $[0, T_\mu)$ due to the almost periodic property of the derivative, a low-order polynomial approximation of the derivative is sufficient to determine the corresponding zero-crossing value. The approximation requires evaluating the cost function at least at P points, where P is the order of the polynomial, resulting into at most P zero-crossings for the approximation of the derivative. In order to reduce complexity, the derivative is evaluated at equi-spaced points $\{0, \frac{T_\mu}{P}, 2\frac{T_\mu}{P}, \ldots, T_\mu\}$. Consequently, only one computation of the matrix exponential $\boldsymbol{\mathcal{R}}(\mu) = \exp(-\mu \mathbf{H}_k)$ is needed at $\mu = T_\mu/P$, and the next $(P-1)$ values are the powers of $\boldsymbol{\mathcal{R}}(\mu)$. The polynomial coefficients may be found by solving a set of linear equations.

In Figure 2 we take as an example the JADE cost function used to perform the joint diagonalization for blind separation in [23]. A practical application of the proposed algorithm to blind separation by optimizing the JADE criterion will be given later in Section 5.2. The cost function $\hat{\mathcal{J}}(\mu)$ is represented by black continuous curve in Figure 2. Its first-order derivative $d\hat{\mathcal{J}}/d\mu$ is represented by the gray continuous curve in Figure 2. The interval T_μ where the derivative needs to be approximated is also shown in Figure 2. In Figure 2, a fourth-order polynomial approximation at equi-spaced points within the interval $[0, T_\mu)$ is used. The approximation is represented by thick dashed line. The steps of the proposed geodesic search algorithm based on polynomial approximation are given in Table 1.

3.2 Line search on $U(n)$ by using a DFT-based approach.

The goal of our second line search method is to find *multiple local minima* of the cost function along a given geodesic and select the best one. The main benefit of this method is that it allows large steps along geodesics. The proposed method requires also *only one* evaluation of the matrix exponential,

1	Given $\mathbf{W}_k \in U(n)$, $-\mathbf{H}_k \in \mathfrak{u}(n)$, compute the eigenvalue of $\mathbf{H}_k$ of highest magnitude $|\omega_{\max}|$
2	Determine the order q of the cost function $\mathcal{J}(\mathbf{W})$ in the coefficients of $\mathbf{W}$, which is the highest degree that t appears in the Taylor expansion of $\mathcal{J}(\mathbf{W} + t\mathbf{Z})$, $\quad t \in \mathbb{R}, \mathbf{Z} \in \mathbb{C}^{n \times n}$
3	Determine the value: $\quad T_\mu = 2\pi/(q|\omega_{\max}|)$
4	Choose the order of the approximating polynomial: $P = 3$, 4, or 5.
5	Evaluate $\mathcal{R}(\mu) = \exp(-\mu\mathbf{H}_k)$ at equi-spaced points $\mu_i \in \{0, T_\mu/P, 2T_\mu/P, \dots, T_\mu\}$ as follows: $\mathbf{R}_0 \triangleq \mathcal{R}(0) = \mathbf{I}$ $\mathbf{R}_1 \triangleq \mathcal{R}\left(\frac{T_\mu}{P}\right) = \exp\left(-\frac{T_\mu}{P}\mathbf{H}_k\right),$ $\mathbf{R}_2 \triangleq \mathcal{R}\left(2\frac{T_\mu}{P}\right) = \mathbf{R}_1\mathbf{R}_1, \dots,$ $\mathbf{R}_P \triangleq \mathcal{R}(T_\mu) = \mathbf{R}_{P-1}\mathbf{R}_1.$
6	By using the computed values of $\mathbf{R}_i$, evaluate the first-order derivative of $\hat{\mathcal{J}}(\mu)$ at μ_i, for $i = 0, \dots, p$: $\quad \hat{\mathcal{J}}'(\mu_i) = -2\Re\{\text{trace}\{\frac{\partial\mathcal{J}}{\partial\boldsymbol{\mathcal{W}}^*}(\mathbf{R}_i\mathbf{W}_k)\ \mathbf{W}_k^H\mathbf{R}_i^H\mathbf{H}_k^H\}\}$
7	Compute the polynomial coefficients $a_0, \dots, a_P$: $\quad a_0 = \hat{\mathcal{J}}'(0)$ and $$\begin{bmatrix} a_1 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} \mu_1 & \mu_1^2 & \dots & \mu_1^P \\ \vdots & \vdots & \dots & \vdots \\ \mu_P & \mu_P^2 & \dots & \mu_P^P \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathcal{J}}'(\mu_1) - a_0 \\ \vdots \\ \hat{\mathcal{J}}'(\mu_P) - a_0 \end{bmatrix}$$
8	Find the smallest real positive root $\rho_{\min}$ of $a_0 + a_1\mu + \dots + a_p\mu^P = 0$. If it exists, then set the step size to $\mu_k = \rho_{\min}$. Otherwise set $\mu_k = 0$.

Table 1: Proposed geodesic search algorithm on $U(n)$ based on polynomial approximation.

but more matrix multiplication operations. The basic idea is to approximate the almost periodic function $d\hat{\mathcal{J}}/d\mu$ (13), by a periodic one, using the classical Discrete Fourier Series (DFT) approach. The method is explained next.

First, the length of the DFT interval T_{DFT} needs to be set. The longer DFT interval is considered, the better approximation is obtained. In practice we have to limit the length of the DFT interval to few periods T_μ (14) corresponding to the highest frequency component (minimum one, maximum depending on how many minima are targeted). Once the DFT interval length is set, the derivative $d\hat{\mathcal{J}}/d\mu$ needs to be sampled at N_{DFT} equi-distant points. According to the Nyquist sampling criterion, $K \geq 2$ samples must be taken within an interval of length T_μ. Therefore, if N_T periods T_μ are considered, the DFT length $N_{\mathrm{DFT}} \geq 2N_T$. Due to the fact that T_μ does not necessarily correspond to any almost period [22] of the derivative, its values at the edges of the DFT interval may differ. In order to avoid approximation mismatches at the edges of the DFT interval, a window function may be applied [24]. The chosen window function must be strictly positive in order to preserve the position of the zeros of the first-order derivative that we are interested in. In our approach we choose a Hann window $h(i)$ [24] and discard the zero-values at the edges. This type of window minimizes the mismatches at the edges of the window. Therefore, instead of approximating the first-order derivative (13), it is more desirable to approximate the windowed derivative $\mathcal{D}(\mu_i) = h(i)\frac{d\hat{\mathcal{J}}}{d\mu}(\mu_i)$, $i = 0, \ldots, N_{\mathrm{DFT}} - 1$ as

$$\mathcal{D}(\mu) \approx \sum_{k=-(N_{\mathrm{DFT}}-1)/2}^{(N_{\mathrm{DFT}}+1)/2} c_k \exp\left(\jmath\frac{2\pi k}{T_{\mathrm{DFT}}}\mu\right), \tag{15}$$

where N_{DFT} is chosen to be an odd number. Again, in order to avoid computing the matrix exponential, the derivative $d\hat{\mathcal{J}}/d\mu$ is evaluated at points $\mu_i \in \{0, T_{\mathrm{DFT}}/N_{\mathrm{DFT}}, \ldots, (N_{\mathrm{DFT}} - 1)T_{\mathrm{DFT}}/N_{\mathrm{DFT}}\}$. After determining the Fourier coefficients c_k, the polynomial corresponding to the Fourier series approximation (15) is set to zero. The roots of the polynomial (15) which are close to the unit circle need to be determined, i.e., $\rho_l = e^{\jmath\omega_l}$, $l \leq 2N_T$. A tolerance δ from the unit circle may be chosen experimentally (e.g., $\delta < 1\%$). The values of μ corresponding to those roots need to be found. Given a descent direction $-\mathbf{H}_k$, the smallest step size value μ_l corresponds to a minimum (or seldom to a saddle point). If no saddle points occur within the DFT window, all the step size values μ_l with l odd, correspond to local minima and the even ones correspond to maxima. Within the interval T_{DFT} there are at most N_T minima, and it is the possible to choose the best one. Therefore, the global minimum within the DFT window can be chosen in order to
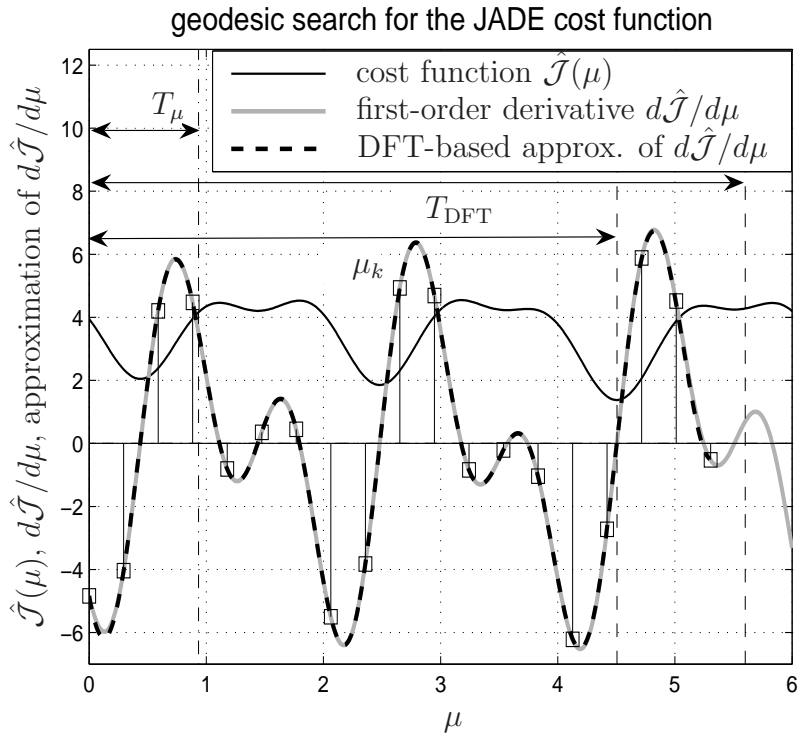
14

Figure 3: Performing the line search for the JADE [23] cost function. The almost periodic behavior of the function $\hat{\mathcal{J}}(\mu)$ and its first-order derivative $d\hat{\mathcal{J}}/d\mu$ (13) along geodesic $\boldsymbol{\mathcal{W}}(\mu)$ (11) may be noticed. The odd zero-crossing values of $d\hat{\mathcal{J}}/d\mu$ correspond to local minima of $\hat{\mathcal{J}}(\mu)$, i.e., to desired values of the step size μ_k. They are obtained by DFT-based approximation of $d\hat{\mathcal{J}}/d\mu$ at equi-spaced points within the interval T_{DFT}.

reduce the cost function as much as possible at every iteration. Finding the best minimum would require evaluation the cost function, therefore computing the matrix exponential for all μ_l with odd l, which is rather expensive. A reasonable solution is in this case to use the information on the sampled values of the cost function. Therefore, the step size is set to the root which is closest to the value that achieves a minimum of the sampled cost function. In Figure 3, we consider the JADE cost function [23], analogously to the example in Figure 2. The steps of the proposed geodesic search algorithm based on discrete Fourier series (DFT) approximation are given in Table 2.

1	Given $\mathbf{W}_k \in U(n)$, $-\mathbf{H}_k \in \mathfrak{u}(n)$, compute the eigenvalue of $\mathbf{H}_k$ of highest magnitude $|\omega_{\max}|$
2	Determine the order q of the cost function $\mathcal{J}(\mathbf{W})$ in the coefficients of $\mathbf{W}$, which is the highest degree that t appears in the Taylor expansion of $\mathcal{J}(\mathbf{W} + t\mathbf{Z}), \quad t \in \mathbb{R}, \mathbf{Z} \in \mathbb{C}^{n \times n}$
3	Determine the value: $\quad T_\mu = 2\pi/(q|\omega_{\max}|)$
4	Choose the sampling factor $K = 3, 4,$ or 5. Select the number of periods T_μ for the approximation, $N_T = 1, 2, \ldots$
5	Determine the length of the DFT interval $T_{\mathrm{DFT}} = N_T T_\mu$ and the DFT length $N_{\mathrm{DFT}} = 2\lfloor K N_T/2 \rfloor + 1$, where $\lfloor \cdot \rfloor$ denotes the integer part.
6	Evaluate the rotation $\mathcal{R}(\mu) = \exp(-\mu \mathbf{H}_k)$ at equi-spaced points $\mu_i \in \{0, T_{\mathrm{DFT}}/N_{\mathrm{DFT}}, \ldots, (N_{\mathrm{DFT}} - 1)T_{\mathrm{DFT}}/N_{\mathrm{DFT}}\}$ as follows: $\mathbf{R}_0 \triangleq \mathcal{R}(0) = \mathbf{I}$ $\mathbf{R}_1 \triangleq \mathcal{R}(T_{\mathrm{DFT}}/N_{\mathrm{DFT}}) = \exp\left(-\frac{T_{\mathrm{DFT}}}{N_{\mathrm{DFT}}}\mathbf{H}_k\right),$ $\mathbf{R}_2 \triangleq \mathcal{R}(2T_{\mathrm{DFT}}/N_{\mathrm{DFT}}) = \mathbf{R}_1 \mathbf{R}_1, \ldots,$ $\mathbf{R}_{N_{\mathrm{DFT}}-1} \triangleq \mathcal{R}((N_{\mathrm{DFT}} - 1)T_{\mathrm{DFT}}/N_{\mathrm{DFT}}) = \mathbf{R}_{N_{\mathrm{DFT}}-2}\mathbf{R}_1.$
7	By using $\mathbf{R}_i$ computed in step **6**, evaluate the first-order derivative of $\hat{\mathcal{J}}(\mu)$ at μ_i, for $i = 0, \ldots, N_{\mathrm{DFT}} - 1$: $\quad \hat{\mathcal{J}}'(\mu_i) = -2\Re\{\mathrm{trace}\{\frac{\partial \mathcal{J}}{\partial \boldsymbol{\mathcal{W}}^*}(\mathbf{R}_i \mathbf{W}_k) \ \mathbf{W}_k^H \mathbf{R}_i^H \mathbf{H}_k^H\}\}$
8	Compute the Hann window: $\quad h(i) = 0.5 - 0.5\cos\left(2\pi\frac{i+1}{N_{\mathrm{DFT}}+1}\right), \ i = 0, \ldots, N_{\mathrm{DFT}} - 1.$
9	Compute the windowed derivative: $\mathcal{D}(\mu_i) = h(i)\hat{\mathcal{J}}'(\mu_i), \ i = 0, \ldots, N_{\mathrm{DFT}} - 1$
10	For $k = -(N_{\mathrm{DFT}} - 1)/2, \ldots, +(N_{\mathrm{DFT}} - 1)/2$ compute the Fourier coefficients: $$c_k = \sum_{i=0}^{N_{\mathrm{DFT}}-1} \mathcal{D}(\mu_i) \exp\left(-\jmath\frac{2\pi i}{N_{\mathrm{DFT}}}k\right).$$
11	Find the roots ρ_l of the approximating Fourier polynomial $$\mathcal{P}(\mu) = \sum_{k=-(N_{\mathrm{DFT}}-1)/2}^{(N_{\mathrm{DFT}}+1)/2} c_k \exp\left(+\jmath\frac{2\pi k}{T_{\mathrm{DFT}}}\mu\right) \approx \mathcal{D}(\mu)$$ close to the unit circle with the radius tolerance δ. If there are not roots of form $\rho_l \approx e^{\jmath \omega_l}$ then set the step size to $\mu_k = 0$ and STOP.
12	If there are roots of form $\rho_l \approx e^{\jmath \omega_l}$ compute the corresponding zero-crossing values of $\mathcal{P}(\mu)$: $\quad \mu_l = [\omega_l T_{\mathrm{DFT}}/(2\pi)]_{\mathrm{modulo}\ T_{\mathrm{DFT}}}.$ Order μ_l in ascending order and pick the odd values $\mu_{2l+1}, l = 0, 1, \ldots$.
13	By using $\mathbf{R}_i$ computed in step **6**, find the value of μ which minimizes the function $\hat{\mathcal{J}}(\mu)$ at μ_i: $\quad \mu_{i\star} = \arg\min_{\mu_i} \mathcal{J}(\mathbf{R}_i \mathbf{W}_k), \ i = 0, \ldots, N_{\mathrm{DFT}} - 1.$ Set the step size to $\mu_k = \arg\min_{\mu_{2l+1}} |\mu_{i\star} - \mu_{2l+1}|.$

Table 2: Proposed geodesic search algorithm on $U(n)$ based on DFT approximation.

3.3 Computational aspects

Both the polynomial approach and the DFT-based approach require several evaluations of the cost function $\hat{\mathcal{J}}(\mu)$ and its first-order derivative $d\hat{\mathcal{J}}/d\mu$ (13) within the corresponding approximation interval. However, they require *only one* computation of the matrix exponential. The desirable property of the matrix exponential that $\exp(-m\mu\mathbf{H}_k) = [\exp(-\mu\mathbf{H}_k)]^m$ is used to evaluate the rotation matrices at equi-spaced points. We also emphasize the fact that for both methods, when evaluating the approximation interval T_μ by using (14), only the largest eigenvalue $|\omega_{\max}|$ of $\mathbf{H}_k$ needs to be computed and not the full eigen-decomposition (nor the corresponding eigen-vector) which is of complexity of $\mathcal{O}(n)$ [8]. The major benefit of the DFT method is that multiple minima are found and the best minimum can be selected at every iteration (which is not necessarily the first local minimum). In conclusion, in terms of complexity both proposed geodesic search methods are more efficient than the Armijo method [18] which requires multiple evaluations of the matrix exponential at every iteration [1] Unlike the method in [15], the proposed method does not require computing any second-order derivatives, which in some cases may involve large matrix dimensions, or they may be non-trivial to calculate (e.g. the JADE criterion (18)).

4 The practical conjugate gradient algorithm on $U(n)$

In this section we propose a practical conjugate gradient algorithm operating on the Lie group of unitary matrices $U(n)$. By combining the generic conjugate gradient algorithm proposed in Subsection 2.4 which uses the approximated Polak-Ribièrre formula with one of the the novel geodesic search algorithms described in Subsection 3.1 and 3.2, we obtain a low complexity conjugate gradient algorithm on the unitary group $U(n)$. The proposed CG-PR algorithm on $U(n)$ is summarized in Table 3.

Remark 1 The line search algorithms in Table 1 and Table 2 and the CG algorithm in Table 3 are designed for *minimizing* a function defined on $U(n)$. They may be easily converted into algorithms for *maximizing* a function on $U(n)$. The rotation matrix $\mathbf{R}_1$ in the step **5** (Table 1) needs to be replaced by $\mathbf{R}_1 = \exp[(+T_\mu/p)\mathbf{H}_k]$ and the sign of the derivative $\hat{\mathcal{J}}'(\mu_i)$ in step **6** needs to be changed, i.e., $\hat{\mathcal{J}}'(\mu_i) = +2\Re\{\text{trace}\{\frac{\partial \mathcal{J}}{\partial \mathcal{W}^*}(\mathbf{R}_i\mathbf{w}_k)\ \mathbf{W}_k^H\mathbf{R}_i^H\mathbf{H}_k^H\}\}$. In Table 2, the same sign changes are needed in steps **6, 7**. Additionally, in step **13**, the value $\mu_{i\star} = \arg\max_{\mu_i} \mathcal{J}(\mathbf{R}_i\mathbf{W}_k)$. Similarly, for the CG algorithm in

1	Initialization: $k = 0$, $\mathbf{W}_k = \mathbf{I}$
2	Compute the Euclidean gradient at $\mathbf{W}_k$. Compute the Riemannian gradient and the search direction at $\mathbf{W}_k$, translated to the group identity: if (k modulo n^2) == 0 $\quad\quad \mathbf{\Gamma}_k = \frac{\partial \mathcal{J}}{\partial \mathbf{W}^*}(\mathbf{W}_k)$ $\quad\quad \mathbf{G}_k = \mathbf{\Gamma}_k \mathbf{W}_k^H - \mathbf{W}_k \mathbf{\Gamma}_k^H$ $\quad\quad \mathbf{H}_k := \mathbf{G}_k$
3	Evaluate $\langle \mathbf{G}_k, \mathbf{G}_k \rangle_{\mathbf{I}} = (1/2)\mathrm{trace}\{\mathbf{G}_k^H \mathbf{G}_k\}$. If it is sufficiently small, then STOP.
4	Given a point $\mathbf{W}_k \in U(n)$ and the tangent direction $-\mathbf{H}_k \in \mathfrak{u}(n)$ determine the step size μ_k along the geodesic emanating from $\mathbf{W}_k$ in the direction of $-\mathbf{H}_k \mathbf{W}_k$, by using the algorithm in Table 1 or the algorithm in Table 2
5	Update: $\mathbf{W}_{k+1} = \exp(-\mu_k \mathbf{H}_k)\mathbf{W}_k$
6	Compute the Euclidean gradient at $\mathbf{W}_{k+1}$. Compute the Riemannian gradient and the search direction at $\mathbf{W}_{k+1}$, translated to the group identity: $\quad\quad \mathbf{\Gamma}_{k+1} = \frac{\partial \mathcal{J}}{\partial \mathbf{W}^*}(\mathbf{W}_{k+1})$ $\quad\quad \mathbf{G}_{k+1} = \mathbf{\Gamma}_{k+1} \mathbf{W}_{k+1}^H - \mathbf{W}_{k+1} \mathbf{\Gamma}_{k+1}^H$ $\quad\quad \gamma_k = \frac{\langle \mathbf{G}_{k+1} - \mathbf{G}_k, \mathbf{G}_{k+1} \rangle_{\mathbf{I}}}{\langle \mathbf{G}_k, \mathbf{G}_k \rangle_{\mathbf{I}}}$ $\quad\quad \mathbf{H}_{k+1} = \mathbf{G}_{k+1} + \gamma_k \mathbf{H}_k$
7	If $\langle \mathbf{H}_{k+1}, \mathbf{G}_{k+1} \rangle_{\mathbf{I}} = \frac{1}{2}\Re\{\mathrm{trace}\{\mathbf{H}_{k+1}^H \mathbf{G}_{k+1}\}\} < 0$, then $\mathbf{H}_{k+1} := \mathbf{G}_{k+1}$
8	$k := k + 1$ and go to step **2**

Table 3: Conjugate gradient algorithm on $U(n)$ using the Polak-Ribièrre formula – CG-PR

Table 3, the update in step **5** would be $\mathbf{W}_{k+1} = \exp(+\mu_k \mathbf{H}_k)\mathbf{W}_k$.

Remark 2   In step **7**, if the CG search direction in not a descent/ascent direction when minimizing/maximizing a function on $U(n)$, it will be reset to the steepest descent/ascent direction. This step remains the same both when minimization or maximization is performed since the inner product $\langle \mathbf{H}_{k+1}, \mathbf{G}_{k+1} \rangle_{\mathbf{I}}$ needs to be positive.

5   Simulation Examples

In this section we apply the proposed Riemannian conjugate gradient algorithm to two different optimization problems on $U(n)$. The first one is the maximization of the Brockett function on $U(n)$, which is a classical example of optimization under orthogonal matrix constraint [9,17]. The second one is the minimization of the JADE cost function [23] which is a practical application of the proposed conjugate gradient algorithm to blind source separation. Other possible signal processing applications are considered in [1–3].

5.1   Diagonalization of a Hermitian Matrix. Maximizing the Brockett criterion on $U(n)$

In this subsection we maximize the Brockett criterion [9,17], which is given as:

$$\mathcal{J}_{\mathrm{B}}(\mathbf{W}) = \mathrm{tr}\{\mathbf{W}^H \boldsymbol{\Sigma} \mathbf{W} \mathbf{N}\}, \text{ subject to } \mathbf{W} \in U(n). \tag{16}$$

The matrix $\boldsymbol{\Sigma}$ is a Hermitian matrix and $\mathbf{N}$ is a diagonal matrix with the diagonal elements $1, \ldots, n$. By maximizing[2] (16), the matrix $\mathbf{W}$ will converge to the eigenvectors of $\boldsymbol{\Sigma}$ and the matrix $\mathbf{D} = \mathbf{W}^H \boldsymbol{\Sigma} \mathbf{W}$ will converge to a diagonal matrix containing the eigenvalues of $\boldsymbol{\Sigma}$ sorted in the ascending order along the diagonal. 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 [2]. This example is chosen for illustrative purposes. The order of the Brockett function is $q = 2$. The Euclidean gradient is given by $\boldsymbol{\Gamma}_{\mathbf{W}} = \boldsymbol{\Sigma} \mathbf{W} \mathbf{N}$. 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

$$\Delta = 10\lg \frac{\mathrm{off}\{\mathbf{W}^H \boldsymbol{\Sigma} \mathbf{W}\}}{\mathrm{diag}\{\mathbf{W}^H \boldsymbol{\Sigma} \mathbf{W}\}}, \quad \Omega = 10\lg \|\mathbf{W}\mathbf{W}^H - \mathbf{I}\|_F^2, \tag{17}$$
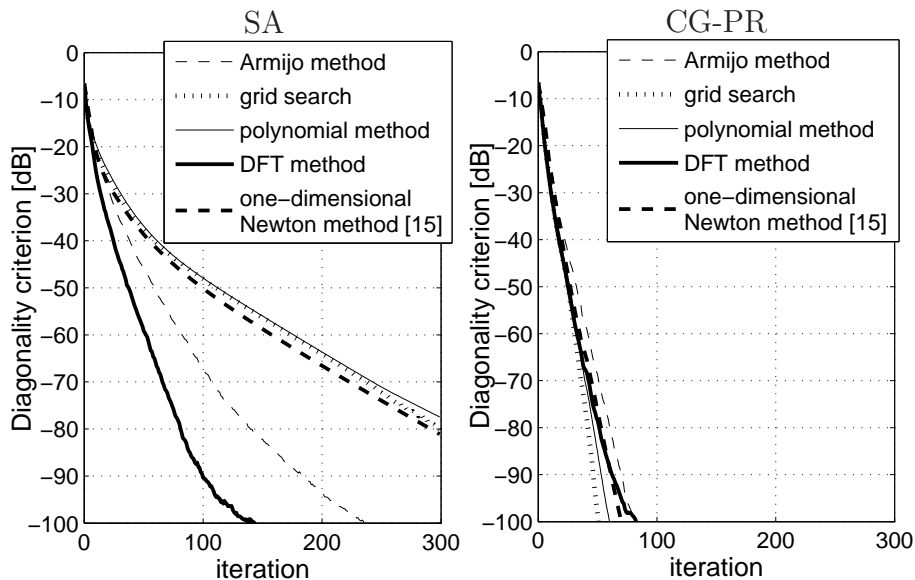
---

[2]See *Remark 1* in Section 4

19

Figure 4: A comparison between the steepest ascent algorithm (SA) on $U(n)$ and the proposed conjugate gradient algorithm on $U(n)$ in Table 3, using Polak-Ribièrre formula (CG-PR). Five line search methods for selecting the step size parameter are considered: the Armijo method [18], the grid search, the line search method in Table 1 which is based on polynomial approximation the line search method in Table 2 which is based on DFT approximation, and the line search method proposed for $SO(n)$ in [15]. The performance measure is the diagonality criterion Δ vs. the iteration step.

where off$\{\cdot\}$ operator computes the sum of the squared magnitudes of the off-diagonal elements of a matrix, and diag$\{\cdot\}$ does the same operation, but for the off-diagonal ones [25]. The diagonality criterion Δ (17) measures the departure of the matrix $\mathbf{W}^H \mathbf{\Sigma} \mathbf{W}$ from the diagonal property in logarithmic scale and it is minimized when the Brockett criterion (16) is maximized. The results are averaged over 100 random realizations of the 6×6 Hermitian matrix $\mathbf{\Sigma}$.

In Figure 4, we compare two different optimization algorithms. The first algorithm is the geodesic steepest ascent on $U(n)$ (SA) obtained from the algorithm in CG-PR Table 3 by setting γ_k to zero at every iteration k. The second algorithm is the CG-PR algorithm in Table 3 (see *Remark 1*, Section 4).

For both algorithms (SA, CG-PR) five different line search methods for selecting the step size parameter are compared. The first one is the Armijo

method [18] used as in [1, 12]. The second is an exhaustive search along geodesics $\mathcal{W}(\mu)$ (11) by using a linear grid $\mu \in [0, 10]$ for the parameter μ_k. A sufficiently large upper limit of the interval has been set experimentally in order to ensure that the interval contains at least one local maximum at every iteration. The lower limit has been set to ensure a reasonable resolution (10^{-4}). The grid search method is very accurate, but extremely expensive and it has been included just for comparison purposes. The third line search method is the polynomial approximation approach proposed in Table 1. The fourth one is the DFT approximation approach in Table 2. The fifth one is the line search method proposed for $SO(n)$ in [15]. It is based on a Newton step which approximates the cost function geodesics by using a first-order Fourier expansion. For the proposed line search methods a polynomial order $P = 5$ has been used (see Table 1). The parameters used in the DFT approach are the sampling factor $K = 3$ and $N_T = 10$ periods T_μ (see Table 2).

It may be noticed in Figure 4 that the CG-PR algorithm outperforms significantly the steepest ascent (SA) algorithm for all line search methods considered here, except the proposed DFT-based approach. The polynomial approximation approach proposed in Table 1 performs equally well as the method in [15], and the grid search method when used with the SA. The proposed DFT-based line search approach in Table 2 outperforms significantly the method in [15] when used with the SA algorithm, and achieves a convergence speed comparable to the one of the CG-PR algorithm. The convergence of SA algorithm with Armijo line search method [18] is better than the proposed polynomial approach and worse than the DFT approach. When used with the CG-PR, all methods achieve similar convergence speed, but their complexities differs. In terms of satisfying the unitary constraint, all algorithms provide good performance. The unitarity criterion (17) is close to the machine precision as also shown in [1].

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

In this subsection we apply the proposed CG-PR algorithm together with the two novel line search methods to a practical application of blind source separation (BSS) of communication signals. A number of $m = 16$ independent signals are separated from their $r = 18$ mixtures based on the statistical properties of the original signals. Four signals from each of the following constellations are transmitted: BPSK, QPSK, 16-QAM and 64-QAM. A total of 5000 snapshots are collected and 100 independent realizations of the $r \times m$

mixture matrix are considered. The signal-to-noise-ratio is SNR= $20dB$. The blind recovery of the desired signals may be done in two stages by using the JADE approach [23]. It can be done up to a phase and a permutation ambiguity, which is inherent to all blind methods. The first stage is the pre-whitening of the received signals based on the subspace decomposition of the received correlation matrix, and it could also be formulated as a maximization of the Brockett function (16) as shown in Section 5.1. The second stage is a unitary rotation operation which needs to be applied to the whitened signals. It is formulated as an optimization under unitary constraint and solved by using the approach proposed in Table 3. The function to be minimized is the joint diagonalization criterion [23]

$$\mathcal{J}_{\text{JADE}}(\mathbf{W}) = \sum_{i=1}^{m} \text{off}\{\mathbf{W}^H \hat{\mathbf{M}}_i \mathbf{W}\} \text{ subject to } \mathbf{W} \in U(m). \qquad (18)$$

The eigenmatrices $\hat{\mathbf{M}}_i$ which are estimated from the fourth order cumulants. The criterion penalizes the departure of all eigen-matrices from the diagonal property [25]. The order of the function (18) is $q = 4$, and the Euclidean gradient of the JADE cost function is given in [1].

In Figure 5-a) we show four of the eighteen received signals, i.e., noisy mixtures of the transmitted signal. Four of the sixteen separated signals are shown in Figure 5-b).

In the the first simulation we study the performance of the proposed Riemannian algorithms in terms of convergence speed considering the *JADE criterion* (18). This JADE criterion (18) is a measure of how well the eigenmatrices $\hat{\mathbf{M}}_i$ are jointly diagonalized.

The whitening stage is the same for both the classical JADE and the Riemannian algorithms. The unitary rotation stage differs. The classical JADE algorithm in [23] performs the approximate joint diagonalization task by using Givens rotations. Three different Riemannian optimization algorithms are considered. The first one is the steepest descent (SD) on $U(m)$ obtained from the CG algorithm in Table 3 by setting γ_k to zero at every iteration k. The line search method in Table 2, which is based on DFT approximation approach is used. The second one is the CG-PR algorithm in Table 3 with the line search method proposed in Table 1 (polynomial approximation approach). The third algorithm is the CG-PR algorithm in Table 3 with the line search method proposed in Table 2 (DFT approximation approach).

In Figure 6 it may noticed that all three Riemannian algorithms outperform the classical Givens rotations approach used in [23]. Again CG algorithm convergences faster compared to the SD algorithm, with both proposed line search methods (Table 1 and Table 2). The parameters used in

a) four of the eighteen received constellation patterns

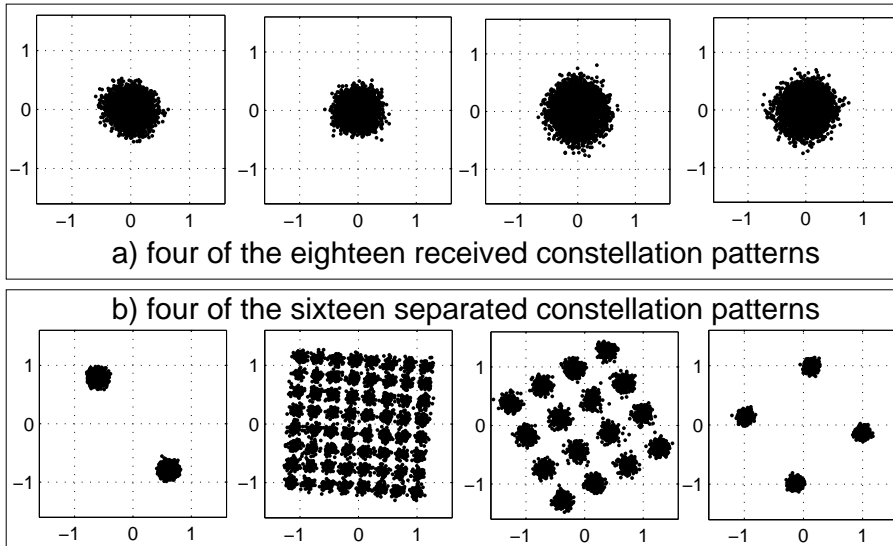b) four of the sixteen separated constellation patterns

Figure 5: The constellation patterns corresponding to a) 4 of the 18 received signals and b) 4 of the 16 recovered signals by the CG-PR algorithm proposed in Table 2 with the novel line search method in Table 1. Since the method is blind there is inherent phase ambiguity, as well as permutation ambiguity.

this simulation for line search methods in Table 1 and 2 are the same as in the previous simulation (Subsection 5.1). All three Riemannian 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 [23] 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 $\mathbf{M}_i$. Therefore, the total complexity of the proposed algorithm 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 very large $m > 10$.

6   Conclusions

In this paper, a Riemannian conjugate gradient algorithm for optimization under unitary matrix constraint is proposed. The algorithm operates on the Lie group of $n \times n$ unitary matrices. In order to reduce the complex-
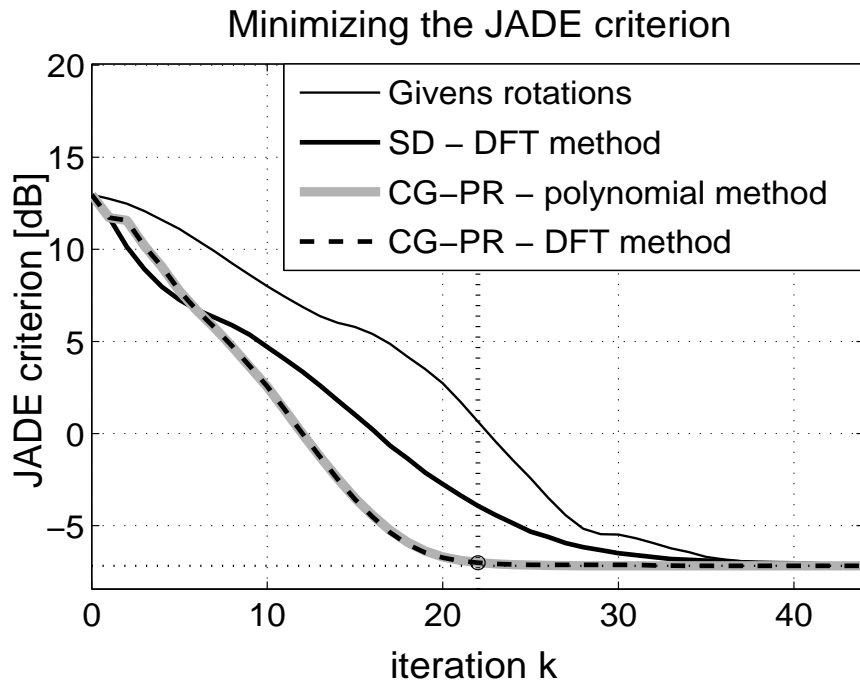
Figure 6: A comparison between the classical JADE algorithm in [23] based on Givens rotations and other three different optimization algorithms: SD on $U(m)$ (in Table 3 by setting $\gamma_k = 0$) + line search method in Table 2 (DFT method), CG-PR algorithm on $U(m)$ in Table 3 + line search method in Table 1 (polynomial method), CG-PR algorithm on $U(m)$ in Table 3 + line search method in Table 2 (DFT method). The performance measures are the JADE criterion (18) vs. the iteration step. All three Riemannian algorithms outperform the classical Givens rotation approach used in [23]. CG-PR converges faster than SD for both proposed line search methods.

ity, it exploits the geometrical properties of $U(n)$ such as simple formulas for the geodesics and the tangent vectors. The almost-periodic behaviour of smooth functions and their derivatives along geodesics on $U(n)$ is shown. Two novel line search methods exploiting this property are introduced. The first one used a low-order polynomial approximation for finding the first local minimum along geodesics on $U(n)$. The second one uses a DFT-based approximation for finding multiple minima along geodesics and selects the best one unlike the Fourier method in [15], which finds only one minimum. Our method models better the spectral content of the almost periodic derivative of the cost function. The two proposed line search methods outperform the Armijo method [18] in terms of computational complexity and provide better performance. The proposed Riemannian CG algorithm not only achieves faster convergence speed compared to the SD algorithms proposed in [1, 12], but also has lower computational complexity. The proposed Riemannian CG algorithm also outperforms the widely used Givens rotations approach used for jointly diagonalizing Hermitian matrices, i.e., in the classical JADE algorithm [23]. 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.

References

[1] T. Abrudan, J. Eriksson, and V. Koivunen, "Steepest descent algorithms for optimization under unitary matrix constraint," *IEEE Transaction on Signal Processing*, vol. 56, pp. 1134–1147, Mar. 2008.

[2] T. Abrudan, J. Eriksson, and V. Koivunen, "Optimization under unitary matrix constraint using approximate matrix exponential," in *Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, 2005.*, pp. 242–246, 28 Oct.–1 Nov. 2005.

[3] J. H. Manton, "On the role of differential geometry in signal processing," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 5, (Philadelphia), pp. 1021–1024, Mar. 2005.

[4] M. P. do Carmo, *Riemannian Geometry*. Mathematics: theory and applications, Birkhauser, 1992.

[5] A. Knapp, *Lie groups beyond an introduction*, vol. 140 of *Progress in mathematics*. Birkhauser, 1996.

[6] D. G. Luenberger, "The gradient projection method along geodesics," *Management Science*, vol. 18, pp. 620–631, 1972.

[7] D. Gabay, "Minimizing a differentiable function over a differential manifold," *Journal of Optimization Theory and Applications*, vol. 37, pp. 177–219, Jun. 1982.

[8] S. T. Smith, *Geometric optimization methods for adaptive filtering*. PhD thesis, Harvard University, Cambridge, MA, May 1993.

[9] R. W. Brockett, "Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems," *Linear Algebra and its Applications*, vol. 146, pp. 79–91, 1991.

[10] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[11] M. Kleinsteuber and K. Hüper, "An intrinsic CG algorithm for computing dominant subspaces," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007*, vol. 4, pp. 1405–1408, Apr. 2007.

[12] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Transactions on Signal Processing*, vol. 50, pp. 635–650, Mar. 2002.

[13] Y. Nishimori and S. Akaho, "Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold," *Neurocomputing*, vol. 67, pp. 106–135, Jun. 2005.

[14] S. Fiori, "Quasi-geodesic neural learning algorithms over the orthogonal group: a tutorial," *Journal of Machine Learning Research*, vol. 1, pp. 1–42, Apr. 2005.

[15] M. D. Plumbley, "Geometrical methods for non-negative ICA: manifolds, Lie groups, toral subalgebras," *Neurocomputing*, vol. 67, pp. 161–197, 2005.

[16] L. Zhang, "Conjugate gradient approach to blind separation of temporally correlated signals," in *IEEE International Conference on Communications, Circuits and Systems, ICCCAS-2004*, vol. 2, (Chengdu, China), pp. 1008–1012, 2004.

[17] S. T. Smith, "Optimization techniques on Riemannian manifolds," *Fields Institute Communications, American Mathematical Society*, vol. 3, pp. 113–136, 1994.

[18] E. Polak, *Optimization: Algorithms and Consistent Approximations.* New York: Springer-Verlag, 1997.

[19] S. Helgason, *Differential geometry, Lie groups and symmetric spaces.* Academic Press, 1978.

[20] S. G. Krantz, *Function theory of several complex variables.* Pacific Grove, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 2nd ed., 1992.

[21] K. Nomizu, "Invariant affine connections on homogeneous spaces," *American Journal of Mathematics*, vol. 76, pp. 33–65, Jan. 1954.

[22] A. Fischer, "Structure of Fourier exponents of almost periodic functions and periodicity of almost periodic functions," *Mathematica Bohemica*, vol. 121, no. 3, pp. 249–262, 1996.

[23] J. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proceedings-F*, vol. 140, no. 6, pp. 362–370, 1993.

[24] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, pp. 51–83, Jan. 1978.

[25] K. Hüper, U. Helmke, and J. B. Moore, "Structure and convergence of conventional Jacobi-type methods minimizing the off-norm function," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 2, (Kobe, Japan), pp. 2124–2129, 11-13 Dec 1996.