Joint Extended Factor Analysis

Ahmad Mouri Sardarabadi, Alle-Jan van der Veen, TU Delft , Delft, The Netherlands

Abstract

Classical Factor Analysis decomposes a covariance matrix into a low-rank and a diagonal part. In this paper we will extend this idea in two ways. We will study the case where we have more than one covariance matrix, and these matrices share the diagonal component. We will also extend Factor Analysis by replacing the diagonal part with a more general data structure. We will solve this problem via non-linear optimization, where we will exploit the Kronecker structure and arrive at a Newton-Krylov based algorithm. We will also provide an algorithm to find the Maximum-Likelihood solution taking advantage of Krylov based solvers.

1 Introduction

Eigenvalue decomposition (EVD) is at the heart of many subspace based signal processing techniques. However, the application of EVD limited to systems with identical components and noise models. In other words we must be able to model the noise covariance matrix as $\sigma^2\mathbf{I}$ where σ^2 is the variance of the noise and $\mathbf{I}$ an identity matrix. If the noise covariance matrix is known, calibration and whitening techniques can be used as pre-processing procedures to make EVD applicable for systems where this assumption does not hold. However a more preferable approach is to develop techniques that can replace EVD for more practical and generic data models.

Using Factor Analysis (FA) [1, 2, 3] for array processing has been suggested by [4] to address the case where the noise is unknown, independent and different for each element in the array i.e. the noise covariance matrix is a diagonal matrix with unknown elements. For cases where the noise covariance matrix is no longer diagonal but has a known structure, Extended FA (EFA) has been suggested [5].

The mentioned techniques work on a single covariance matrix. However in many applications the desired subspace changes rapidly which means that a series of short-term covariance matrices are available. In this paper we will show that applying subspace estimation techniques for each short-term covariance leads to sub-optimal estimates, since the stationarity of the diagonal (or extended structure) is not used. To address this issue we will develop a technique based on EFA that estimates the desired subspaces jointly and is flexible enough to include generic noise models. We will also show that some undesired signals, like interfering sources, can be modeled using EFA.

Estimating the unknown parameters leads to a non-linear optimization problem. We will develop a Gauss-Newton-Krylov based technique that solves the non-linear optimization efficiently in both memory usage and complexity.

The setup of this paper is as follows: In Sec.2 we discuss the data and covariance models, in Sec.3 we derive the Cramér-Rao bound for the estimated parameters, in Sec.4 describes the algorithm we use to estimate the parameters and in Sec.5 we use simulations to evaluate the performance of the proposed method.

2 Data Model and Problem Definition

We assume to have access to a series of sample covariance matrices $\hat{\mathbf{R}}_k$ where $k = 1,...,K$, from K independent "snapshots", each containing N samples. We assume the following model for $\mathscr{E}\{\hat{\mathbf{R}}_k\} = \mathbf{R}_k$:

$$\mathbf{R}_k = \mathbf{A}_k\mathbf{A}_k^H + \mathbf{\Psi} \tag{1}$$

where $\mathbf{A}_k$ is a low rank matrix of size $P \times Q_k$ with $Q_k < P$ for all k, ^{H} is the Hermitian transpose and $\mathbf{\Psi}$ is a positive-definite matrix that is assumed to be stationary. Depending on the application we are interested in $\mathbf{A}_k$ and $\mathbf{\Psi}$ or only one of them. In many applications we are just interested in the column span of $\mathbf{A}_k$.

A common application for $\mathbf{\Psi}$ is to represent the noise covariance matrix of the system. For a system which is calibrated or has identical components, it is common to assume that $\mathbf{\Psi} = \sigma^2\mathbf{I}_P$, where $\mathbf{I}_P$ is a $P \times P$ identity matrix. In this case the column span of $\mathbf{A}_k$ can be estimated using an eigenvalue decomposition of $\mathbf{R}_k$. In other words let the (economical) singular value decomposition (SVD) of $\mathbf{A}_k$ be

$$\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^H$$

where $\mathbf{U}_k$ is a semi-unitary matrix of size $P \times Q_k$ forming an orthonormal basis for the column space of $\mathbf{A}_k$, $\mathbf{V}_k$ is a $Q_k \times Q_k$ unitary matrix forming an orthonormal basis for the row space of $\mathbf{A}_k$ and $\mathbf{\Sigma}_k$ is a $Q_k \times Q_k$ diagonal matrix containing the singular values. Similarly let the eigenvalue

ISBN 978-3-8007-3662-1

decomposition of $\mathbf{R}_k$ be

$$\mathbf{R}_k = \mathbf{Q}_k \boldsymbol{\Lambda}_k \mathbf{Q}_k^H.$$

Then for the case where $\boldsymbol{\Psi} = \sigma^2 \mathbf{I}_P = \sigma^2 \mathbf{Q}_k \mathbf{Q}_k^H$ we have

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{1,k} & \mathbf{Q}_{2,k} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_k & \mathbf{Q}_{2,k} \end{bmatrix}.$$

As mentioned, this makes it possible to find the subspace for $\mathbf{A}_k$, from $\mathbf{R}_k$, using eigenvalue decomposition of $\mathbf{R}_k$. However this technique fails when $\boldsymbol{\Psi}$ takes another model. In this paper we will allow more general models for $\boldsymbol{\Psi}$ and discuss techniques for finding $\mathbf{A}_k$ and $\boldsymbol{\Psi}$ from noisy estimates $\hat{\mathbf{R}}_k$. We will also estimates $\mathbf{A}_k$ and $\boldsymbol{\Psi}$ jointly. The joint covariance matrix of the entire dataset can be found using the independence of the data-matrices to be

$$\begin{aligned}\mathbf{R}_{\text{total}} &= \begin{bmatrix} \mathbf{R}_1 & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{R}_K \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_1\mathbf{A}_1^H & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{A}_K\mathbf{A}_K^H \end{bmatrix} + \mathbf{I}_K \otimes \boldsymbol{\Psi} \qquad (2)\end{aligned}$$

where $\otimes$ is the Kronecker product.

2.1 Joint Factor Analysis

The first generalization that we are going to consider is modeling $\boldsymbol{\Psi} = \mathbf{D}$ where $\mathbf{D}$ is a positive definite unknown diagonal matrix. For a single covariance matrix, this decomposition is known as Factor Analysis (FA).

We are interested in finding $\mathbf{D}$ and $\mathbf{A}_k$. One way to achieve this is by applying FA to each covariance matrix separately to find $\hat{\mathbf{D}}_k$ and $\hat{\mathbf{A}}_k$, and then use $\hat{\mathbf{D}} = 1/K \sum_k \hat{\mathbf{D}}_k$ to estimate $\mathbf{D}$. Then we can use whitening techniques and find a better estimate of the subspace of $\mathbf{A}_k$.

However we propose Joint Factor Analysis (JFA), for estimating $\hat{\mathbf{A}}_k$ and $\hat{\mathbf{D}}$ using the entire dataset. We will demonstrate, using Cramér-Rao bound and simulations, that this approach gives estimates with lower variance and we avoid re-estimating the subspaces using whitening.

2.2 Joint Extended Factor Analysis

Factor Analysis can be extended to a more general model for $\boldsymbol{\Psi}$ where a certain structure is assumed to be known for $\boldsymbol{\Psi}$. Here we consider $\boldsymbol{\Psi}$ of the form

$$\boldsymbol{\Psi} = \mathbf{M} \odot \boldsymbol{\Psi}$$

where $\odot$ is the Hadamard or element-wise multiplication and $\mathbf{M}$ is a symmetric matrix containing only ones and zeros. We call $\mathbf{M}$ a mask matrix. We can model various types of covariance matrices using this approach (for example: block-diagonal matrices, band matrices, sparse matrices, etc.). We assume $\mathbf{M}$ to be known based on the application. Similar to JFA we propose Joint Extended FA (JEFA), where we are interested in estimating $\hat{\boldsymbol{\Psi}}$ and $\hat{\mathbf{A}}_k$ jointly. For this problem the total number of unknowns that need to be estimated is

$$n = 2P \sum_{k=1}^{K} Q_k + \operatorname{tr}(\mathbf{M}^2) \qquad (3)$$

Note that for $\mathbf{M} = \mathbf{I}$ where $\mathbf{I}$ is an identity matrix of appropriate size, this model reduces to diagonal factor analysis.

Before we introduce the algorithm to estimate the desired parameters, we derive the Cramér-Rao bound for the variance of the estimates.

3 Cramér-Rao Bound

Some general results from multivariate statistical analysis are as follows. Suppose that $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator of $\boldsymbol{\theta}_0$. Then the asymptotic distribution of $(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0)$ is $\mathcal{N}(\mathbf{0}, \mathbf{C})$ where $\mathbf{C}$ is the inverse of the Fisher information matrix $\mathbf{F}$. $\mathbf{C}$ is the Cramér-Rao lower bound (CRB) for an unbiased estimator. For normally distributed data with covariance matrix $\mathbf{R}$, the Fisher information matrix is

$$\mathbf{F} = N\mathbf{J}^H (\mathbf{R}^{-T} \otimes \mathbf{R}^{-1}) \mathbf{J}$$

where $\mathbf{J} = \partial \operatorname{vec}(\mathbf{R}) / \partial \boldsymbol{\theta}^T$.

Suppose now that $\mathbf{F}$ is singular, then for identifiability we need to pose additional constraints on $\boldsymbol{\theta}$, say $\mathbf{h}(\boldsymbol{\theta}_0) = \mathbf{0}$, where $\mathbf{h}(\boldsymbol{\theta})$ is a vector of functions. Let the Jacobian of $\mathbf{h}(\boldsymbol{\theta})$ be given by

$$\mathbf{H}(\boldsymbol{\theta}) := \frac{\partial \mathbf{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}.$$

The constrained CRB, $\mathbf{C}$, is then given by [6]

$$\mathbf{C} = \mathbf{U}(\mathbf{U}^H \mathbf{F}(\boldsymbol{\theta}_0) \mathbf{U})^{-1} \mathbf{U}^H \qquad (4)$$

where $\mathbf{U}$ is a semi-unitary matrix, and the columns of $\mathbf{U}$ form an orthonormal basis for the null-space of $\mathbf{H}$ such that $\mathbf{H}(\boldsymbol{\theta}_0)\mathbf{U}(\boldsymbol{\theta}_0) = \mathbf{0}$. The constraints should be chosen such that $\mathbf{U}^H \mathbf{F}(\boldsymbol{\theta}_0) \mathbf{U}$ is invertible.

We will now apply these results to our situation. First we consider only a single snapshot $\mathbf{R}_k$ with model $\mathbf{R}_k = \mathbf{A}_k \mathbf{A}_k + \boldsymbol{\Psi}$, as given by (1). We will parametrize $\operatorname{vec}(\boldsymbol{\Psi})$ as

$$\operatorname{vec}(\boldsymbol{\Psi}) = \mathbf{S}_U \psi + \mathbf{S}_L \bar{\psi} + (\mathbf{I}_P \circ \mathbf{I}_P)\mathbf{d},$$

where $\bar{\cdot}$ is the complex conjugate, $\mathbf{S}_L$ and $\mathbf{S}_U$ are suitable selection matrices based on the structure of $\mathbf{M}$, the entries of the strictly upper-triangular part of $\boldsymbol{\Psi}$ are stacked into the vector ψ, its diagonal entries $\mathbf{d} = \operatorname{vecdiag}(\boldsymbol{\Psi})$. The unknown complex parameters are stacked into a vector $\boldsymbol{\theta}_k$,

$$\boldsymbol{\theta}_k = \begin{bmatrix} \operatorname{vec}(\mathbf{A}_k) \\ \operatorname{vec}(\bar{\mathbf{A}}_k) \\ \psi \\ \bar{\psi} \\ \mathbf{d} \end{bmatrix} \qquad (5)$$

To derive the Fisher information matrix, we partition the corresponding $\mathbf{J}_k = \partial \operatorname{vec}(\mathbf{R}) / \partial \boldsymbol{\theta}_k^T$, to conform with the partitioning of $\boldsymbol{\theta}_k$ such that $\mathbf{J}_k = [\mathbf{J}_{\mathbf{A}_k}, \mathbf{J}_{\bar{\mathbf{A}}_k}, \mathbf{J}_{\psi}, \mathbf{J}_{\bar{\psi}}, \mathbf{J}_{\mathbf{d}}]$. Using Wirtinger derivatives we find

$$\begin{aligned} \mathbf{J}_{\mathbf{A}_k} &= (\bar{\mathbf{A}}_k \otimes \mathbf{I}_P) \\ \mathbf{J}_{\bar{\mathbf{A}}_k} &= (\mathbf{I}_p \otimes \mathbf{A}_k)\mathbf{K}_{P,Q_k} \\ \mathbf{J}_{\psi} &= \mathbf{S}_U \\ \mathbf{J}_{\bar{\psi}} &= \mathbf{S}_L \\ \mathbf{J}_{\mathbf{d}} &= (\mathbf{I}_P \circ \mathbf{I}_P) \end{aligned} \tag{6}$$

where $\mathbf{K}_{P,Q}$ is a permutation matrix such that $\operatorname{vec}(\mathbf{X}^T) = \mathbf{K}_{P,Q}\operatorname{vec}(\mathbf{X})$ for $\mathbf{X}$ a $P \times Q$ matrix. We also used the relation

$$\begin{aligned} \operatorname{vec}(\mathbf{R}_k) &= (\bar{\mathbf{A}}_k \otimes \mathbf{I}_P)\operatorname{vec}(\mathbf{A}_k) + \operatorname{vec}(\boldsymbol{\Psi}) \\ &= (\mathbf{I}_P \otimes \mathbf{A}_k)\mathbf{K}_{P,Q_k}\operatorname{vec}(\bar{\mathbf{A}}_k) + \operatorname{vec}(\boldsymbol{\Psi}). \end{aligned}$$

The unconstrained Fisher information $\mathbf{F}_k$ is singular, because the FA model is invariant with respect to a multiplication of the matrix $\mathbf{A}_k$ with a unitary matrix at the right, including a phase change for each column. For identifiability, we need to pose Q_k^2 constraints on the matrix $\mathbf{A}_k$. Without loss of generality we choose $\mathbf{A}_k^H\mathbf{A}_k$ to be diagonal (which poses $Q_k(Q_k-1)$ real-valued constraints), and $\operatorname{diag}(\mathbf{A}_k^T\mathbf{A}_k)$ to be real (which poses another Q_k constraints).

To write this as a function $\mathbf{h}(\boldsymbol{\theta}_k) = \mathbf{0}$, let $\mathbf{E}_1 = (\mathbf{I}_{Q_k} \circ \mathbf{I}_{Q_k})^T$ and let $\mathbf{E}_2$ be a complementary set of rows such that $[\mathbf{E}_1^T, \mathbf{E}_2^T]^T$ is a permutation matrix. Then $\mathbf{E}_1\operatorname{vec}(\mathbf{A}_k^T\mathbf{A}_k)$ selects the diagonal elements of $\mathbf{A}_k^T\mathbf{A}_k$, and $\mathbf{E}_2\operatorname{vec}(\mathbf{A}_k^H\mathbf{A}_k)$ selects the off-diagonal entries of $\mathbf{A}_k^H\mathbf{A}_k$. The constraint function $\mathbf{h} = [\mathbf{h}_1^T, \mathbf{h}_2^T]^T$ is then

$$\mathbf{h}(\boldsymbol{\theta}_k) = \begin{bmatrix} \mathbf{h}_1(\boldsymbol{\theta}_k) \\ \mathbf{h}_2(\boldsymbol{\theta}_k) \end{bmatrix} = \begin{bmatrix} \mathbf{E}_1\operatorname{vec}(\mathbf{A}_k^T\mathbf{A}_k - \mathbf{A}_k^H\bar{\mathbf{A}}_k) \\ \mathbf{E}_2\operatorname{vec}(\mathbf{A}_k^H\mathbf{A}_k) \end{bmatrix} = \mathbf{0}$$

and its Jacobian $\mathbf{H}(\boldsymbol{\theta}_k)$ is

$$\mathbf{H}(\boldsymbol{\theta}_k) = \begin{bmatrix} \mathbf{H}_{\mathbf{A}_k} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where

$$\mathbf{H}_{\mathbf{A}_k} = \begin{bmatrix} \frac{\partial \mathbf{h}_1}{\partial \operatorname{vec}^T(\mathbf{A}_k)} & \frac{\partial \mathbf{h}_1}{\partial \operatorname{vec}^T(\bar{\mathbf{A}}_k)} \\ \frac{\partial \mathbf{h}_2}{\partial \operatorname{vec}^T(\mathbf{A}_k)} & \frac{\partial \mathbf{h}_2}{\partial \operatorname{vec}^T(\bar{\mathbf{A}}_k)} \end{bmatrix},$$

and the trailing zeros correspond to derivatives of $\mathbf{h}(\boldsymbol{\theta}_k)$ with respect to $\boldsymbol{\Psi}$, $\bar{\boldsymbol{\Psi}}$ and $\mathbf{d}$. The needed derivatives are given by

$$\begin{aligned} \frac{\partial \mathbf{h}_1}{\partial \operatorname{vec}^T(\mathbf{A}_k)} &= \mathbf{E}_1\left[(\mathbf{I}_Q \otimes \mathbf{A}_k^T) + (\mathbf{A}_k^T \otimes \mathbf{I}_Q)\mathbf{K}_{P,Q}\right] \\ \frac{\partial \mathbf{h}_1}{\partial \operatorname{vec}^T(\bar{\mathbf{A}}_k)} &= -\mathbf{E}_1\left[(\mathbf{I}_Q \otimes \mathbf{A}_k^H) + (\mathbf{A}_k^H \otimes \mathbf{I}_Q)\mathbf{K}_{P,Q}\right] \\ \frac{\partial \mathbf{h}_2}{\partial \operatorname{vec}^T(\mathbf{A}_k)} &= \mathbf{E}_2(\mathbf{I}_Q \otimes \mathbf{A}_k^H) \\ \frac{\partial \mathbf{h}_2}{\partial \operatorname{vec}^T(\bar{\mathbf{A}}_k)} &= \mathbf{E}_2(\mathbf{A}_k^T \otimes \mathbf{I}_P)\mathbf{K}_{P,Q} \end{aligned}$$

Using QR or SVD on $\mathbf{H}(\boldsymbol{\theta}_k)$, we can find a basis $\mathbf{U}_k$ for the null-space of $\mathbf{H}(\boldsymbol{\theta}_k)$, and calculate the Constrained CRB $\mathbf{C}_k$ for a single measurement.

The performance of FA on each snapshot separately, follows by assuming that the estimates $\hat{\boldsymbol{\theta}}$ are independent, then for $\hat{\mathbf{D}} = 1/K\sum_k \hat{\mathbf{D}}_k$ the CRB becomes

$$\mathbf{C}_{\mathbf{d}} = 1/K^2 \sum_k \mathbf{C}_{\mathbf{d},k}, \tag{7}$$

where $\mathbf{C}_{\mathbf{d},k}$ is the sub-matrix of $\mathbf{C}_k$ corresponding to $\mathbf{d}$. This bound is higher than the bound we will drive for the entire dataset shortly.

Now that we have the Fisher Information for each snapshot we can use them to find the CRB of the entire dataset. Because the time samples are independent, the joint pdf for the total dataset is given by

$$p(\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_K; \boldsymbol{\theta}) = \prod_k p_k(\mathbf{X}_k; \boldsymbol{\theta}_k) \tag{8}$$

where $\mathbf{X}$ is a $P \times N$ data matrix for each snapshot. Loglikelihood of the entire dataset then becomes

$$l(\boldsymbol{\theta}) = \sum_k l_k(\boldsymbol{\theta}_k). \tag{9}$$

For simpler representation we redefine the unknowns as

$$\boldsymbol{\theta}_{\mathbf{A}_k} = \begin{bmatrix} \operatorname{vec}(\mathbf{A}_k) \\ \operatorname{vec}(\bar{\mathbf{A}}_k) \end{bmatrix}. \tag{10}$$

and

$$\boldsymbol{\theta}_{\boldsymbol{\Psi}} = \begin{bmatrix} \boldsymbol{\Psi} \\ \bar{\boldsymbol{\Psi}} \\ \mathbf{d} \end{bmatrix}. \tag{11}$$

Now we need to find the gradients of the new loglikelihood. First we take derivatives with respects to $\boldsymbol{\theta}_{\mathbf{A}_k}$. If we assume that all $\mathbf{A}_k$ are independent i.e. $\partial l_k(\boldsymbol{\theta}_k)/\partial \mathbf{A}_j = \mathbf{0}$ for $k \neq j$ and we find

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_{\mathbf{A}_k}} = \frac{\partial l_k(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_{\mathbf{A}_k}} \tag{12}$$

which is the same as for estimating the ML separately. However for $\boldsymbol{\Psi}$ we have

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_{\boldsymbol{\Psi}}} = \sum_k \frac{\partial l_k(\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_{\boldsymbol{\Psi}}}. \tag{13}$$

Now we can write the Fisher information for the entire dataset as

$$\mathbf{F}_{total,\mathbf{A}_k\mathbf{A}_k} = \mathbf{F}_{k,\mathbf{A}_k\mathbf{A}_k} \tag{14}$$

$$\mathbf{F}_{total,\mathbf{A}_k\mathbf{A}_j} = \mathbf{0} \qquad k \neq j \tag{15}$$

$$\mathbf{F}_{total,\mathbf{A}_k\boldsymbol{\Psi}} = \mathbf{F}_{k,\mathbf{A}_k\boldsymbol{\Psi}} \tag{16}$$

$$\mathbf{F}_{total,\boldsymbol{\Psi}\boldsymbol{\Psi}} = \sum_k \mathbf{F}_{k,\boldsymbol{\Psi}\boldsymbol{\Psi}} \tag{17}$$

or in matrix form

$$\mathbf{F}_{total} = N \begin{bmatrix} \mathbf{F}_{1,\mathbf{A}_1\mathbf{A}_1} & \mathbf{0} & \cdots & \mathbf{F}_{1,\mathbf{A}_1\boldsymbol{\Psi}} \\ \mathbf{0} & \mathbf{F}_{2,\mathbf{A}_2\mathbf{A}_2} & \cdots & \mathbf{F}_{2,\mathbf{A}_2\boldsymbol{\Psi}} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{F}_{1,\mathbf{A}_1\boldsymbol{\Psi}}^H & \mathbf{F}_{2,\mathbf{A}_2\boldsymbol{\Psi}}^H & \cdots & \sum_i^K \mathbf{F}_{k,\boldsymbol{\Psi}\boldsymbol{\Psi}} \end{bmatrix}. \tag{18}$$

ISBN 978-3-8007-3662-1

where the $\mathbf{F}_k$ can be calculated using the results above as

$$\mathbf{F}_{k,\mathbf{A}_k\mathbf{A}_k} = \begin{bmatrix}\mathbf{J}_{\mathbf{A}_k}^H \\ \mathbf{J}_{\bar{\mathbf{A}}_k}^H\end{bmatrix} \left(\mathbf{R}_k^{-T} \otimes \mathbf{R}_k^{-1}\right) \begin{bmatrix}\mathbf{J}_{\mathbf{A}_k} & \mathbf{J}_{\bar{\mathbf{A}}_k}\end{bmatrix} \tag{19}$$

$$\mathbf{F}_{k,\mathbf{A}_k\boldsymbol{\Psi}} = \begin{bmatrix}\mathbf{J}_{\mathbf{A}_k}^H \\ \mathbf{J}_{\bar{\mathbf{A}}_j}^H\end{bmatrix} \left(\mathbf{R}_k^{-T} \otimes \mathbf{R}_k^{-1}\right) \mathbf{J}_{\boldsymbol{\Psi}} \tag{20}$$

$$\mathbf{F}_{total,\boldsymbol{\Psi}\boldsymbol{\Psi}} = \mathbf{J}_{\boldsymbol{\Psi}}^H \left[\sum_k^K \left(\mathbf{R}_k^{-T} \otimes \mathbf{R}_k^{-1}\right)\right] \mathbf{J}_{\boldsymbol{\Psi}} \tag{21}$$

where

$$\mathbf{J}_{\boldsymbol{\Psi}} = \begin{bmatrix}\mathbf{S}_U & \mathbf{S}_L & (\mathbf{I}_P \circ \mathbf{I}_P)\end{bmatrix}.$$

Because the constraint matrix is only a function of $\boldsymbol{\theta}_{\mathbf{A}_k}$, the constraint matrix for the entire dataset becomes

$$\mathbf{H}_{total} = \left[\begin{array}{ccccccc} \mathbf{H}_{\mathbf{A}_1} & \mathbf{0} & \dots & \mathbf{0} & & & \\ \mathbf{0} & \mathbf{H}_{\mathbf{A}_2} & \dots & \vdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \mathbf{0} & & & \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{H}_{\mathbf{A}_K} & & & \end{array}\right].$$

Because $\mathbf{H}_{total}$ is very sparse, we can use efficient QR decomposition algorithms to find a unitary basis, $\mathbf{U}_{total}$ for its null space efficiently.

The Constrained CRB can now be found, similar as before, using $\mathbf{F}_{total}$ and $\mathbf{U}_{total}$.

4 Algorithm

In this part we discuss some techniques to estimate the unknown parameters. As we will show this leads to a nonlinear optimization problem that we will solve using a variation of a Jacobian-free Newton-Krylov (JFNK) technique [7] and a matrix-free Gauss-Newton-Krylov (MFGNK) [8]. The main idea behind the Newton-Krylov technique is to solve the linear system needed to find the direction of descent using Krylov subspace based solvers. Krylov-subspace based algorithms find the solution to a linear system such as $\mathbf{B}\mathbf{x} = \mathbf{y}$ by repeated calculation of the matrix vector product $\mathbf{B}\mathbf{v}$ where, for a square $\mathbf{B}$, $\mathbf{v}$ is a vector having the same length as the unknown vector $\mathbf{x}$. In many applications, and as we will demonstrate in our case, $\mathbf{B}$ is related to the Jacobians and the multiplications can be performed using these Jacobians. The JFNK and MFGNK techniques avoid storing the Jacobian by using a Taylor expansion to approximate the needed matrix vector products [7, 8]. The Kronecker structure of the Jacobians derived in the previous section allows us to develop a method that also avoids storing the Jacobians but does exact computation of the matrix vector product similar to the work done in [9].

We will discuss Non-linear Weighted Least Squares (NLWLS) and the Maximum Likelihood (ML) for finding $\hat{\boldsymbol{\Psi}}$ and $\hat{\mathbf{A}}_k$.

4.1 Non-linear Weighted Least Squares

We start by vectoring and stacking all the covariance matrices to form a measurement vector

$$\hat{\mathbf{r}} = \begin{bmatrix}\mathrm{vec}^T(\hat{\mathbf{R}}_1) & \dots & \mathrm{vec}^T(\hat{\mathbf{R}}_K)\end{bmatrix}^T, \tag{22}$$

and similarly

$$\mathbf{r}(\boldsymbol{\theta}) = \begin{bmatrix}\mathrm{vec}^T(\mathbf{R}_1(\boldsymbol{\theta})) & \dots & \mathrm{vec}^T(\mathbf{R}_K(\boldsymbol{\theta}))\end{bmatrix}^T, \tag{23}$$

where

$$\boldsymbol{\theta} = \begin{bmatrix}\boldsymbol{\theta}_{\mathbf{A}_1}^T & \dots & \boldsymbol{\theta}_{\mathbf{A}_K}^T & \boldsymbol{\theta}_{\boldsymbol{\Psi}}^T\end{bmatrix}^T. \tag{24}$$

We can estimate the unknown parameters in $\boldsymbol{\theta}$ using NLWLS defined as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|(\mathbf{W}^{T/2} \otimes \mathbf{W}^{1/2})[\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})]\|_2^2 \tag{25}$$

where $\mathbf{W}$ is a weighting matrix. The optimum weighting matrix is the covariance matrix of the entire dataset $\mathbf{W} = \mathbf{R}_{\text{total}}^{-1}$, however because we have only access to $\hat{\mathbf{R}}_k$ we use

$$\mathbf{W} = \begin{bmatrix}\hat{\mathbf{R}}_1^{-1} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \hat{\mathbf{R}}_K^{-1}\end{bmatrix} \tag{26}$$

which will give an asymptotically optimal solution for a Gaussian distributed data matrix [10].

A very common iterative technique for solving nonlinear optimization problems is the Gauss-Newton algorithm, where the Hessian is replaced by the Gramian of the Jacobians [11]. The updates are similar to Newton updates and are given by

$$\hat{\boldsymbol{\theta}}^{j+1} = \hat{\boldsymbol{\theta}}^{j} + \mu_j \boldsymbol{\Delta}\Delta \tag{27}$$

where $\boldsymbol{\Delta}$ is the direction of descent. To find $\boldsymbol{\Delta}$ we need to solve

$$\mathbf{B}(\boldsymbol{\theta})\boldsymbol{\Delta} = \mathbf{g}(\boldsymbol{\theta}) \tag{28}$$

where

$$\mathbf{B}(\boldsymbol{\theta}) = \mathbf{J}^H(\boldsymbol{\theta})(\mathbf{W}^T \otimes \mathbf{W})\mathbf{J}(\boldsymbol{\theta}) \tag{29}$$

and $\mathbf{g}(\boldsymbol{\theta})$ is the gradient of the NLWLS given by

$$\mathbf{g}(\boldsymbol{\theta}) = \mathbf{J}^H(\boldsymbol{\theta})(\mathbf{W}^T \otimes \mathbf{W})[\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})]. \tag{30}$$

We will drop the dependency on $\boldsymbol{\theta}$ from the notation and write only $\mathbf{J}$ and $\mathbf{r}$ because $\boldsymbol{\theta}$ does not change while we are solving for $\boldsymbol{\Delta}$. We will continue the iterations given by (27) until $\|\mathbf{g}(\hat{\boldsymbol{\theta}}^{(j)}\|_2 < \varepsilon$ where $\varepsilon > 0$ depends on the desired accuracy. This concludes the Gauss-Newton algorithm. The key step is solving the linear system in (28). We will discuss a Krylov based method for solving this system to find $\boldsymbol{\Delta}$.

4.2 Krylov-Based Methods

There are various Krylov subspace based solvers, an overview of these solvers can be found for example in [12]. We know from our study of the Fisher information that the solution to the problem is not unique, this means that the Jacobians and hence $\mathbf{B}$ is singular. One possible Krylov solver that is capable of finding a solution for singular matrices is the Minres-QLP algorithm [13] and for this reason we have chosen this solver for our iterative approach.

The most expensive computation during the Minres-QLP iterations is the multiplication of the matrix $\mathbf{B}$ with a vector, i.e operation of the form $\mathbf{u} = \mathbf{B}\mathbf{v}$. We will now show how we can achieve this multiplication without needing to store the Jacobians using the Kronecker structure.

In order to calculate $\mathbf{u} = \mathbf{B}\mathbf{v}$ for $\mathbf{B}$ given in (29), we define an intermediate result $\mathbf{z} = \mathbf{J}\mathbf{v}$. Given the fact that $\mathbf{v}$, $\mathbf{u}$ have the same dimensions as $\boldsymbol{\theta}$ and $\mathbf{z}$ has the same dimensions as $\mathbf{r}$ we are going to partition them in the same manner

$$\mathbf{v} = \begin{bmatrix} \mathrm{vec}(\mathbf{V}_{\mathbf{A}_1}) \\ \mathrm{vec}(\mathbf{V}_{\bar{\mathbf{A}}_1}) \\ \vdots \\ \mathbf{S}_U^T \mathrm{vec}(\mathbf{V}_{\boldsymbol{\Psi}}) \\ \mathbf{S}_L^T \mathrm{vec}(\mathbf{V}_{\boldsymbol{\Psi}}) \\ (\mathbf{I}_P \circ \mathbf{I}_P)^T \mathrm{vec}(\mathbf{V}_{\boldsymbol{\Psi}}) \end{bmatrix}, \ \mathbf{u} = \begin{bmatrix} \mathrm{vec}(\mathbf{U}_{\mathbf{A}_1}) \\ \mathrm{vec}(\mathbf{U}_{\bar{\mathbf{A}}_1}) \\ \vdots \\ \mathbf{S}_U^T \mathrm{vec}(\mathbf{U}_{\boldsymbol{\Psi}}) \\ \mathbf{S}_L^T \mathrm{vec}(\mathbf{U}_{\boldsymbol{\Psi}}) \\ (\mathbf{I}_P \circ \mathbf{I}_P)^T \mathrm{vec}(\mathbf{U}_{\boldsymbol{\Psi}}) \end{bmatrix} \tag{31}$$

and

$$\mathbf{z} = \begin{bmatrix} \mathrm{vec}(\mathbf{Z}_1) \\ \vdots \\ \mathrm{vec}(\mathbf{Z}_K) \end{bmatrix} \tag{32}$$

To find $\mathbf{u}$ we will compute $\mathbf{U}_{\mathbf{A}_k}$, $\mathbf{U}_{\bar{\mathbf{A}}_k}$ and $\mathbf{U}_{\boldsymbol{\Psi}}$. We note that if $\mathbf{V}_{\bar{\mathbf{A}}_k} = \bar{\mathbf{V}}_{\mathbf{A}_k}$ then $\mathbf{U}_{\bar{\mathbf{A}}_k} = \bar{\mathbf{U}}_{\mathbf{A}_k}$ which means that only $\mathbf{U}_{\mathbf{A}_k}$ needs to be calculated.

The Jacobian for the entire dataset is given by

$$\mathbf{J} = \frac{\partial \mathrm{vec}(\mathbf{R})}{\partial \boldsymbol{\theta}^T} = \begin{bmatrix} \mathbf{J}_{\mathbf{A}_1} & \mathbf{J}_{\bar{\mathbf{A}}_1} & \cdots & \mathbf{0} & \mathbf{J}_{\boldsymbol{\Psi}} \\ \mathbf{0} & & \cdots & \mathbf{0} & \mathbf{J}_{\boldsymbol{\Psi}} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \mathbf{J}_{\boldsymbol{\Psi}} \\ \mathbf{0} & \cdots & \mathbf{J}_{\mathbf{A}_K} & \mathbf{J}_{\bar{\mathbf{A}}_K} & \mathbf{J}_{\boldsymbol{\Psi}} \end{bmatrix} \tag{33}$$

and hence using $\mathbf{z} = \mathbf{J}\mathbf{v}$, (32) and (6):

$$\begin{aligned} \mathrm{vec}(\mathbf{Z}_k) &= (\bar{\mathbf{A}}_k \otimes \mathbf{I}_P)\mathrm{vec}(\mathbf{V}_{\mathbf{A}_k}) \\ &\quad + (\mathbf{I}_P \otimes \mathbf{A}_k)\mathbf{K}_{P,Q}\mathrm{vec}(\mathbf{V}_{\bar{\mathbf{A}}_k}) + \mathrm{vec}(\mathbf{V}_{\boldsymbol{\Psi}}) \\ &= \mathrm{vec}\left(\mathbf{V}_{\mathbf{A}_k}\mathbf{A}_k^H + \mathbf{A}_k\mathbf{V}_{\mathbf{A}_k}^H + \mathbf{M} \odot \mathbf{V}_{\boldsymbol{\Psi}}\right) \end{aligned}$$

where we have used $\mathbf{V}_{\bar{\mathbf{A}}_k} = \bar{\mathbf{V}}_{\mathbf{A}_k}$. It follows directly from unvectorizing both sides that

$$\mathbf{Z}_k = \mathbf{V}_{\mathbf{A}_k}\mathbf{A}_k^H + \mathbf{A}_k\mathbf{V}_{\mathbf{A}_k}^H + \mathbf{M} \odot \mathbf{V}_{\boldsymbol{\Psi}}. \tag{34}$$

This means that we can calculate $\mathbf{J}\mathbf{v}$ by only reshaping the vector $\mathbf{v}$ to appropriate matrices and applying (34). The variables $\mathbf{A}_k$ are the current estimates of unknown parameters.

The next matrix vector multiplications that we need is $\mathbf{z}_{\mathbf{W}} = (\mathbf{W}^T \otimes \mathbf{W})\mathbf{z}$. Using the properties of Kronecker products it is straightforward to show that

$$\mathbf{W}^T \otimes \mathbf{W} = \begin{bmatrix} \hat{\mathbf{R}}_1^{-T} \otimes \hat{\mathbf{R}}_1^{-1} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \hat{\mathbf{R}}_K^{-T} \otimes \hat{\mathbf{R}}_K^{-1} \end{bmatrix}, \tag{35}$$

thus $\mathbf{z}_{\mathbf{W}} = (\mathbf{W}^T \otimes \mathbf{W})\mathbf{z}$ can be calculated using

$$\mathbf{Z}_{\mathbf{W}_k} = \hat{\mathbf{R}}_k^{-1}\mathbf{Z}_k\hat{\mathbf{R}}_k^{-1} \tag{36}$$

and

$$\mathbf{z}_{\mathbf{W}} = \begin{bmatrix} \mathrm{vec}(\mathbf{Z}_{\mathbf{W}_1}) \\ \vdots \\ \mathrm{vec}(\mathbf{Z}_{\mathbf{W}_K}) \end{bmatrix}. \tag{37}$$

The final product we need to calculate is $\mathbf{u} = \mathbf{J}^H\mathbf{z}_{\mathbf{W}}$. From the structure of (33), we see that

$$\begin{aligned} \mathrm{vec}(\mathbf{U}_{\mathbf{A}_k}) &= \mathbf{J}_{\mathbf{A}_k}^H \mathrm{vec}(\mathbf{Z}_{\mathbf{W}_k}) \\ \mathrm{vec}(\mathbf{U}_{\bar{\mathbf{A}}_k}) &= \mathbf{J}_{\bar{\mathbf{A}}_k}^H \mathrm{vec}(\mathbf{Z}_{\mathbf{W}_k}). \end{aligned}$$

Unvectorizing both sides and applying (6) we find

$$\begin{aligned} \mathbf{U}_{\mathbf{A}_k} &= \mathbf{Z}_{\mathbf{W}_k}\mathbf{A}_k \\ \mathbf{U}_{\bar{\mathbf{A}}_k} &= \mathbf{Z}_{\mathbf{W}_k}^T\bar{\mathbf{A}}_k. \end{aligned} \tag{38}$$

The remaining term $\mathbf{U}_{\boldsymbol{\Psi}}$ is given by:

$$\mathbf{U}_{\boldsymbol{\Psi}} = \sum_{k=1}^{K} \mathbf{M} \odot \mathbf{Z}_{\mathbf{W}_k}. \tag{39}$$

To summarize, in order to calculate $\mathbf{B}\mathbf{v}$ we reshape $\mathbf{v}$ into $\mathbf{V}_{\mathbf{A}_k}$ and $\mathbf{V}_{\boldsymbol{\Psi}}$ and use (34), (36), (38) and (39) to find the result. The gradient $\mathbf{g}$ can be calculated in a similar manner by using $\mathbf{Z}_k = \hat{\mathbf{R}}_k - \mathbf{R}_k$. The procedure that performs these steps is provided to Minres-QLP which then solves for $\boldsymbol{\Delta}$.

By assuming $\mathbf{V}_{\bar{\mathbf{A}}_k} = \bar{\mathbf{V}}_{\mathbf{A}_k}$ we showed in (34) that $\mathbf{Z}_k$ is Hermitian, and because $\hat{\mathbf{R}}_k^{-1}$ is Hermitian so is $\mathbf{Z}_{\mathbf{W}_k}$. From the properties of Hermitian matrices we have $\mathbf{Z}_{\mathbf{W}_k}^T = \bar{\mathbf{Z}}_{\mathbf{W}_k}$ and thus $\mathbf{U}_{\bar{\mathbf{A}}_k} = \bar{\mathbf{U}}_{\mathbf{A}_k}$. We still need to show that the assumption about $\mathbf{V}_{\bar{\mathbf{A}}_k}$ is valid. It can be shown that Minres-QLP provides $\mathbf{v}$ that have the property $\mathbf{V}_{\bar{\mathbf{A}}_k} = \bar{\mathbf{V}}_{\mathbf{A}_k}$ when solving $\mathbf{B}\boldsymbol{\Delta} = \mathbf{g}$ if $\mathbf{g}$ has this property. Calculating $\mathbf{g}$ is achieved by setting $\mathbf{Z}_k = \hat{\mathbf{R}}_k - \mathbf{R}_k$ and following the procedure above. Because both $\hat{\mathbf{R}}_k$ and $\mathbf{R}_k$ are Hermitian, it follows that the needed property holds for $\mathbf{g}$ and hence for $\mathbf{v}$.

4.3 Maximum Likelihood

An alternative to the NLWLS solved in Sec. 4.1 is to maximize the likelihood function (9). A Hessian-free (similar to Jacobian and matrix free) variation in combination with Krylov solvers have been suggested in [14]. However by using the results of the previous section we will develop a method based on the scoring method, where the Hessian is replaced by the Fisher information matrix [15]. We already have derived the Fisher information for the entire dataset. We will denote this method Scoring-Krylov (SK).

The scoring method as presented by [15] can be summarized as

$$\hat{\boldsymbol{\theta}}^{(j+1)} = \hat{\boldsymbol{\theta}}^{(j)} + \mu_j\boldsymbol{\Delta}$$

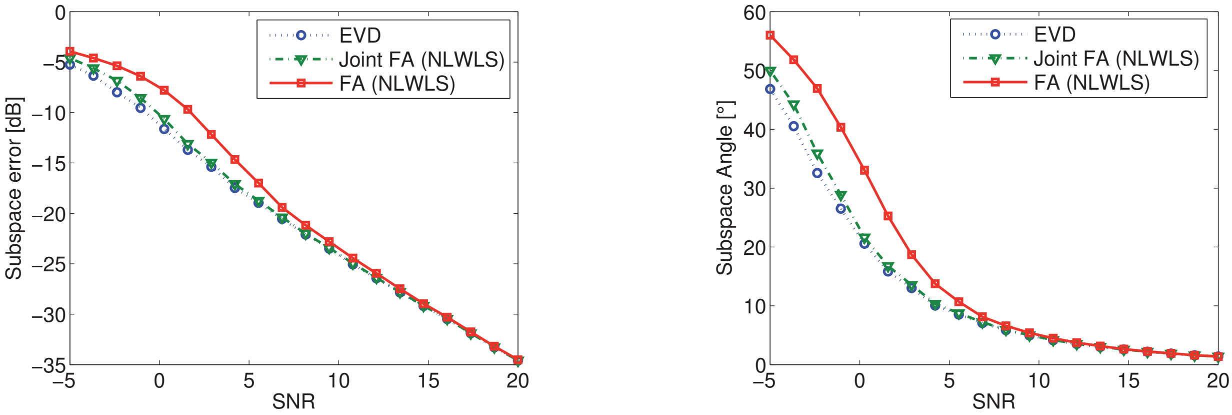
where $\boldsymbol{\Delta}$ is the solution to

$$\mathbf{F}_{total}(\boldsymbol{\theta})\boldsymbol{\Delta} = \nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

Using the same technique as we have done to find the derivatives for NLWLS we find the derivative of the likelihood to be

$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \mathbf{J}^H(\boldsymbol{\theta})(\mathbf{R}^{-T}(\boldsymbol{\theta}) \otimes \mathbf{R}^{-1}(\boldsymbol{\theta}))(\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})).$$

Figure 1



(a) Attentuation as the function of SNR for different techniques.

(b) Angle difference between the estimated subspace and true subspace.

This similarity between the ML and (NL)WLS is well known and is also studied in [10]. Given the definition of the gradient and the Fisher information, we observe that the same techniques used for NLWLS can be applied to ML. We only need to replace the weighting matrix to be the current best estimate of $\mathbf{R}(\boldsymbol{\theta})$. However this approach requires the inversion of the covariance matrices at each iteration. If based on the structure of $\mathbf{M}$, inversion of $\boldsymbol{\Psi}$ is computationally more accommodating then the Woodbury matrix identity can be used to find the inverse with less complexity.

This means that the SK method is similar to the NLWLS with the exception of the weighting matrix. Also in application of Minres-QLP the only step that needs modification is (36), where we replace $\hat{\mathbf{R}}_k$ with $\mathbf{R}_k(\hat{\mathbf{A}}_k^{(j)}, \hat{\boldsymbol{\Psi}}^{(j)})$ (note that j remains the same during the Minres-QLP iterations).

This will allow us to find the ML solution using the advantages of the Krylov based solvers without the need to store the Fisher information matrix or the Jacobians.

5 Simulations

We will evaluate the performance of the proposed model and algorithm using a series of simulations. We will start by studying performance of JFA and then we will simulate a direction of arrival (DOA) estimation scenario using JEFA.

5.1 Subspace Performance

In this section we study the performance of FA and JFA where we take $\boldsymbol{\Psi} = \sigma^2 \mathbf{I}_P$, so we can compare the performance to EVD.

For this simulations we have chosen $Q_k = 2$, $P = 5$, $K = 5$ and $\sigma = 1$. We study the subspace estimation performance for various signal to noise ratios (SNR) ranging from -5 dB to 20 dB. Each sample covariance matrix is generated using $N = 100$ samples and $\mathbf{A}_k$ is generated as a random complex matrix.

Two metrics are used to measure performance of the estimated subspace. We use the estimated subspaces to find a projection matrix into the null-space of $\hat{\mathbf{A}}_k$ which we will denote by $\hat{\mathbf{P}}_k$ and we measure

$$\text{Subspace error} = \frac{\|\hat{\mathbf{P}}_k \mathbf{A}\mathbf{A}^H \hat{\mathbf{P}}_k\|_F}{\|\mathbf{A}\mathbf{A}^H\|_F}.$$

In **Fig.** 1a the result of this simulations is presented. As FA and JFA have to estimate more parameters, we expect a drop in performance compared to EVD. This simulation shows that this occurs for FA at low SNR. JFA exploits the stationarity of the noise component and hence has a quite small performance penalty with respect to EVD.

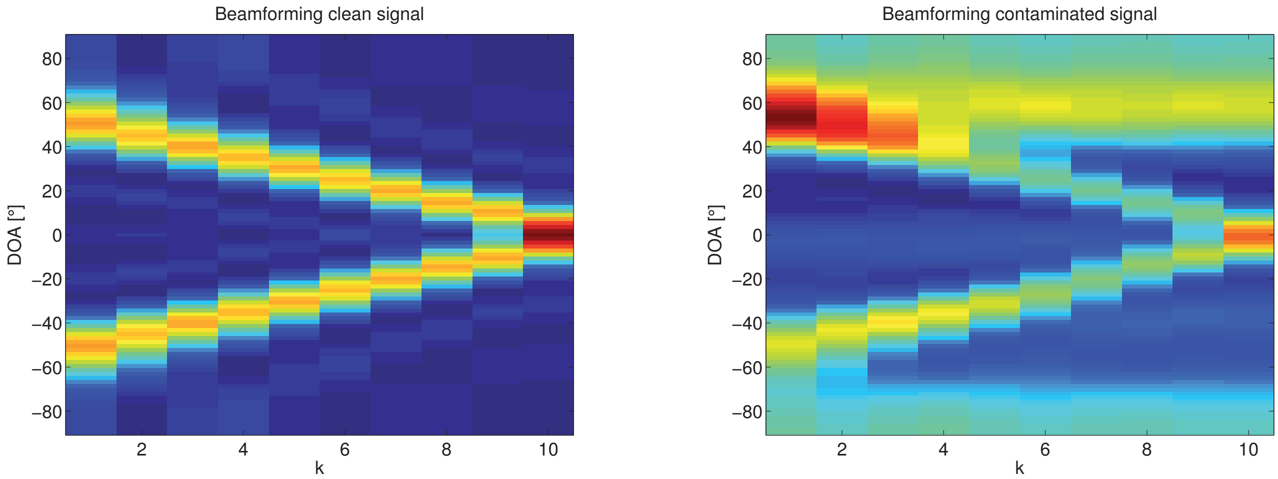
The other metric we use is the angle between two subspaces calculated using MATLAB command *subspace*. This result is shown in **Fig.** 1b. The subspace angle difference between the true and estimated subspaces decreases as SNR increases. JFA follows the performance of EVD with a very small gap.

Because JFA has a more general model, it is applicable in many practical situations and we have shown that applying this technique in classical scenarios where $\boldsymbol{\Psi} = \boldsymbol{\Sigma}^2 \mathbf{I}_P$ does not result in a significant performance loss.

5.2 DOA Estimation using EJFA

In this scenario we use the estimated subspace from EVD, EJFA and EFA as the input to a DOA estimator based on ESPRIT. We simulate $K = 10$ with $Q_k = 2$ targets moving along the tracks T_1 and T_1 as illustrated in **Fig.** 3 between the snapshots. We have a uniform linear array with $P = 7$ receivers that observe the targets, however $P_0 = 5$ of these receivers are contaminated with unknown interfering signals. We will model the interfering signal as a stationary unknown colored noise which leads to a mask matrix de-

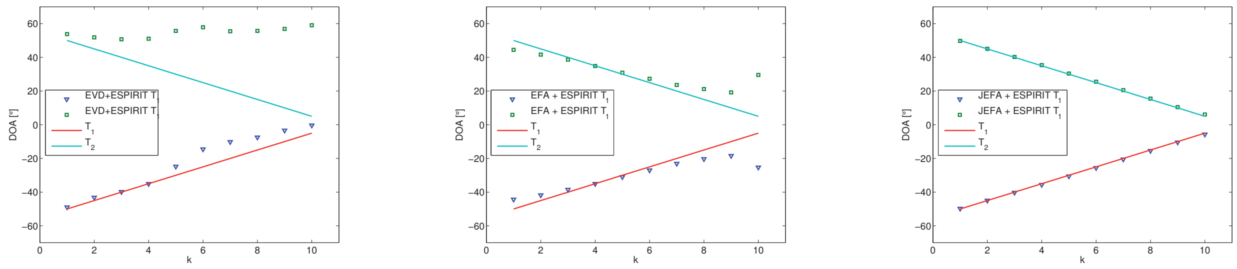
Figure 2



(a) Result of Beamforming on $\mathbf{A}_k\mathbf{A}_k^H$

(b) Result of Beamforming on $\mathbf{R}_k$

Figure 3



(a) DOA results for EVD + ESPRIT

(b) DOA results for EFA + ESPRIT

(c) DOA results for JEFA + ESPRIT

fined by

$$\mathbf{M} = \begin{bmatrix} \mathbf{1}_{P_0}\mathbf{1}_{P_0}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}$$

where $\mathbf{1}_{P_0}$ is a $P_0 \times 1$ vector with all entries equal to unity. The sample covariance matrix for each snapshot is obtained using $N = 100$ samples.

Fig. 2a shows the result of matched filter beamforming on the simulated data when there is no inferefere present and **Fig.** 2b shows the effect of the interfering signals. Because of the limited resolution of the device, the beamformer cannot differentiate the two signals in the last snapshot.

We present the Monte-Carlo (MC) results of ESPRIT for each snapshot based on the subspace estimated by different algorithm in **Fig.** 3.

- Because $\mathbf{\Psi} \neq \sigma^2\mathbf{I}$, EVD is not able to recover the correct subspace and hence as illustrated in **Fig.** 3a the estimated subspace is biased, (note that the bias is different between each snapshot and is a function of both $\mathbf{A}_k$ and $\mathbf{\Psi}$ and the wild behavior shown in this figure does not disappear by increasing the MC runs).
- **Fig.** 3b shows the result obtained by applying the EFA separately on each snapshot followed by ESPRIT. Because the resolution decreases for higher angles (as can be seen in **Fig.** 2a) and because not the entire dataset is used the variance of the DOA estimates is higher for the first few snapshots, also as the targets get closer it is more difficult to differentiate their subspace. Both problems affect the performance of EFA.
- The performance of JEFA is illustrated in **Fig.** 3c. Because both the correct data model has been used and estimation is done over the entire dataset, JEFA is able to recover the subspaces and hence the DOA estimates more accurately.

5.3 Cramér-Rao Bound simulation

In this part we investigate the performance of the proposed algorithm using the Cramér-Rao bound. We use a setup with $P = 5$, $Q_k = 2$, $\mathbf{\Psi} = \mathbf{D}$ with diagonal element ranging from 0.5 to 1.5. Two different approaches have been compared. The first approach is to apply FA separately and then use $\hat{\mathbf{D}} = 1/K\sum_k \hat{\mathbf{D}}_k$. The other approach is to estimate using JFA.

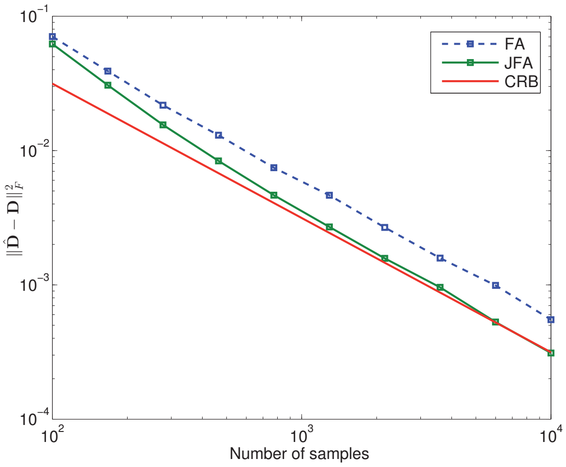


Figure 4 Cramér-Rao Bound

We use

$$\mathscr{E}\{\|\hat{\mathbf{D}}-\mathbf{D}\|_F^2\} = \mathscr{E}\{\mathrm{vec}(\hat{\mathbf{D}}-\mathbf{D})^H \mathrm{vec}(\hat{\mathbf{D}}-\mathbf{D})\}$$
$$= \mathrm{tr}[\mathscr{E}\{\mathrm{vec}(\hat{\mathbf{D}}-\mathbf{D})\mathrm{vec}(\hat{\mathbf{D}}-\mathbf{D})^H\}] \geq \mathrm{tr}(\mathbf{C}_{\boldsymbol{\Psi}})$$

where $\mathbf{C}_{\boldsymbol{\Psi}}$ is the sub-matrix of CRB corresponding to $\boldsymbol{\Psi}$, to measure performance. We estimate $\mathscr{E}\{\|\hat{\mathbf{D}}-\mathbf{D}\|_F^2\}$ using Monte Carlo simulations. **Fig.** 4 shows the result of this simulations. This figure clearly illustrates that the proposed joint estimation reaches the CRB asymptotically and that applying the estimation separately followed by an averaging results in a sub-optimal estimation with higher variance.

6 Conclusion

We have provided a method for jointly estimating the non-stationary low-rank and stationary structured part of a series of covariance matrices by developing efficient algorithms based on Newton-Krylov optimization techniques. An algorithm to find the ML estimates has also been presented.

The Cramér-Rao bound for the entire dataset has been provided and the performance of the algorithm have been illustrated using simulations.

The general structure of the data model should make application of this technique possible in a wide range of signal processing applications.

7 Literature

[1] T. W. Anderson and H. Rubin, "Statistical inference in factor analysis," *In Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, vol. 5, pp. 111 – 150, 1956.

[2] K. G. Jöreskog, "A general approach to confirmatory maximum likelihood factor analysis," *Psychometrika*, vol. 34, no. 2, 1969.

[3] S. Y. Lee, "The Gauss-Newton algorithm for the weighted least squares factor analysis," *Journal of the Royal Statistical Society*, vol. 27, no. 2, June 1978.

[4] A.-J. van der Veen, A. Leshem, and A.-J. Boonstra, "Signal processing for radio astronomical arrays," in *Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2004*, July 2004, pp. 1–10.

[5] A. Mouri Sardarabadi and A. van der Veen, "Subspace estimation using factor analysis," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012 IEEE 7th*, june 2012, pp. 477 –480.

[6] A. K. Jagannatham and B. D. Rao, "Cramer-Rao lower bound for constrained complex parameters," *IEEE Signal Processing Lettets*, vol. 11, no. 11, NOVEMBER 2004.

[7] D. Knoll and D. Keyes, "Jacobian-free Newton–Krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, no. 2, pp. 357 – 397, 2004.

[8] V. Akcelik, G. Biros, and O. Ghattas, "Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation," in *Supercomputing, ACM/IEEE 2002 Conference*, Nov 2002, pp. 41–41.

[9] A. Mouri Sardarabadi and A.-J. van der Veen, "Application of Krylov based methods in calibration for radio astronomy," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014 IEEE 8th*, June 2014, pp. 153–156.

[10] B. Ottersten, P. Stoica, and R. Roy, "Covariance matching estimation techniques for array signal processing applications," *Digital Signal Processing*, vol. 8, no. 3, pp. 185 – 210, 1998.

[11] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. London: Academic Press Inc. [Harcourt Brace Jovanovich Publishers], 1981.

[12] S.-C. T. Choi, "Iterative methods for singular linear equations and least-squares problems," Ph.D. dissertation, Stanford University, 2006.

[13] S. Choi, C. Paige, and M. Saunders, "MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems," *SIAM Journal on Scientific Computing*, vol. 33, no. 4, pp. 1810–1836, 2011. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/100787921

[14] O. Vinyals and D. Povey, "Krylov subspace descent for deep learning," *arXiv preprint arXiv:1111.4259*, 2011.

[15] S. M. Kay, *Fundamentals of Statistical Signal Processing, Estimation theory*. Prentice Hall, 1993, vol. Volume I.