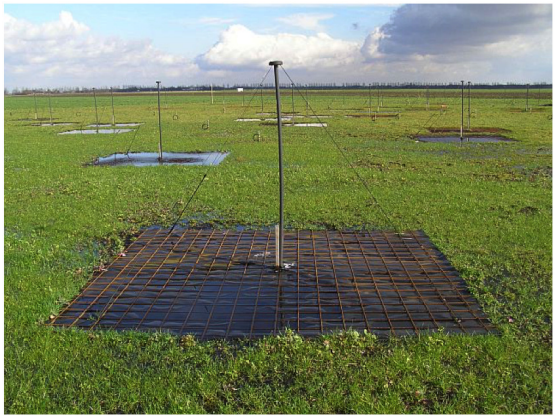


EE 4715 Array Processing

10. Factor Analysis

June 2022

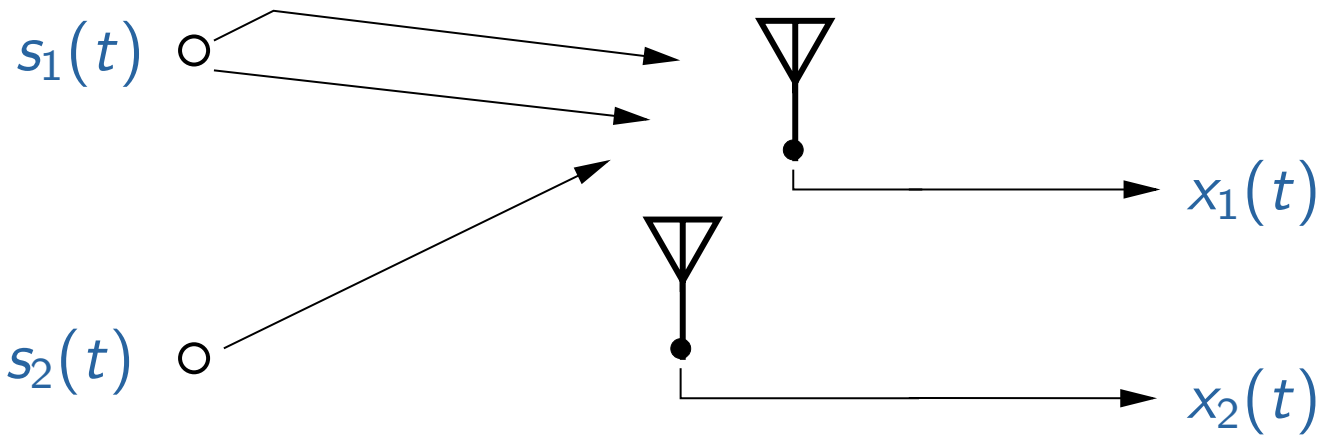
Introduction: Subspace estimation

Classical data model

Consider Q sources $\boldsymbol{s}[n]$, a $P \times Q$ mixing matrix $\boldsymbol{A}$, and white noise $\boldsymbol{n}[n]$:

$$\boldsymbol{x}[n] = \boldsymbol{A}\boldsymbol{s}[n] + \boldsymbol{n}[n]$$

$$\boldsymbol{R} := \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{H}}] = \boldsymbol{A}\boldsymbol{\Sigma}_s\boldsymbol{A}^{\mathrm{H}} + \sigma^2\boldsymbol{I}$$



Introduction: Subspace estimation

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{\Sigma}_s\boldsymbol{A}^{\mathrm{H}} + \sigma^2\boldsymbol{I}$$

Eigenvalue decompositions are used to estimate the column span of $\boldsymbol{A}$:

- **No noise:**

$$\boldsymbol{R} =: \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{H}} = \begin{bmatrix} \boldsymbol{U}_0 & \boldsymbol{U}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_0 & \\ & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_0^{\mathrm{H}} \\ \boldsymbol{U}_1^{\mathrm{H}} \end{bmatrix}$$

- **With additive white noise:**

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{U}_0 & \boldsymbol{U}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_0 + \sigma^2\boldsymbol{I} & \\ & \sigma^2\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_0^{\mathrm{H}} \\ \boldsymbol{U}_1^{\mathrm{H}} \end{bmatrix}$$

In both cases, $\boldsymbol{U}_0$ is a basis for the column span of $\boldsymbol{A}$. This is used in subspace-based techniques, e.g. in MUSIC.

Subspace estimation

- **With spatially colored noise:** $\boldsymbol{R}_n = \boldsymbol{D}$ (some diagonal)

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{\Sigma}_s\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{D}$$

The eigenvalue decomposition does not give the correct subspace. We need to do *prewhitening*: work with $\boldsymbol{R}_n^{-1/2}\boldsymbol{x}[n]$.

$$\begin{aligned}\boldsymbol{R}_n^{-1/2}\boldsymbol{R}\boldsymbol{R}_n^{-1/2} &= (\boldsymbol{R}_n^{-1/2}\boldsymbol{A})\boldsymbol{\Sigma}_s(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{R}_n^{-1/2}) + \boldsymbol{I} \\ &= \begin{bmatrix} \boldsymbol{U}_0 & \boldsymbol{U}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_0 + \boldsymbol{I} & \\ & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_0^{\mathrm{H}} \\ \boldsymbol{U}_1^{\mathrm{H}} \end{bmatrix}\end{aligned}$$

After prewhitening, $\boldsymbol{U}_0$ is a basis for the column span of $\boldsymbol{R}_n^{-1/2}\boldsymbol{A}$.

This can be done if $\boldsymbol{R}_n$ is known (requires calibration). What if it isn't?

Factor Analysis

Data model

We work with the model

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}} + \boldsymbol{D} \qquad \text{[low rank plus diagonal]}$$

The goal in FA is, given $\boldsymbol{R}$, estimate $\boldsymbol{A}$ and $\boldsymbol{D}$.

FA is an old problem (1920?), usually for real factors. It is used in many domains where data has arbitrary scalings.

Uniqueness

- $\boldsymbol{A}$ is unique only up to a rotation $\boldsymbol{Q}$: if $\boldsymbol{A}$ fits the model, then also $\boldsymbol{A}\boldsymbol{Q}$. This is solved by placing constraints on $\boldsymbol{A}$.
  The subspace $\mathrm{ran}(\boldsymbol{A})$ is invariant anyway.

Factor Analysis

If $Q \geq P$ then the problem is not identifiable: we can set $\boldsymbol{D} = 0$. What is the maximum Q (number of sources?)

Counting number of equations and number of unknowns:

- For a complex $\boldsymbol{R}$, we have P^2 real-valued "observations"
- Number of real-valued unknowns in $\boldsymbol{A}$ is $2PQ$, in $\boldsymbol{D}$ is P
- Need Q^2 constraints to make $\boldsymbol{A}$ unique (e.g., constraining $\boldsymbol{A}^{\mathrm{H}}\boldsymbol{D}^{-1}\boldsymbol{A}$ to be diagonal and the first row of $\boldsymbol{A}$ real)

Total degrees of freedom:

$$s = P^2 \; - \; (2PQ + P - Q^2) = (P - Q)^2 - P$$

Identifiability requires $s > 0$ (more equations than unknowns)

$$s > 0 \quad \Leftrightarrow \quad Q < P - \sqrt{P}$$

FA algorithms

Ad hoc algorithm

In practice we do not have $\boldsymbol{R}$, but an estimate $\hat{\boldsymbol{R}}$.

We can formulate Factor Analysis as a covariance fitting problem (least squares):

$$\min_{\boldsymbol{A},\boldsymbol{D}} \|\hat{\boldsymbol{R}} - \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}} - \boldsymbol{D}\|_F^2$$

Solve using Alternating Least Squares: estimate $\boldsymbol{A}$, then estimate $\boldsymbol{D}$, etc.

This converges *very* slowly.

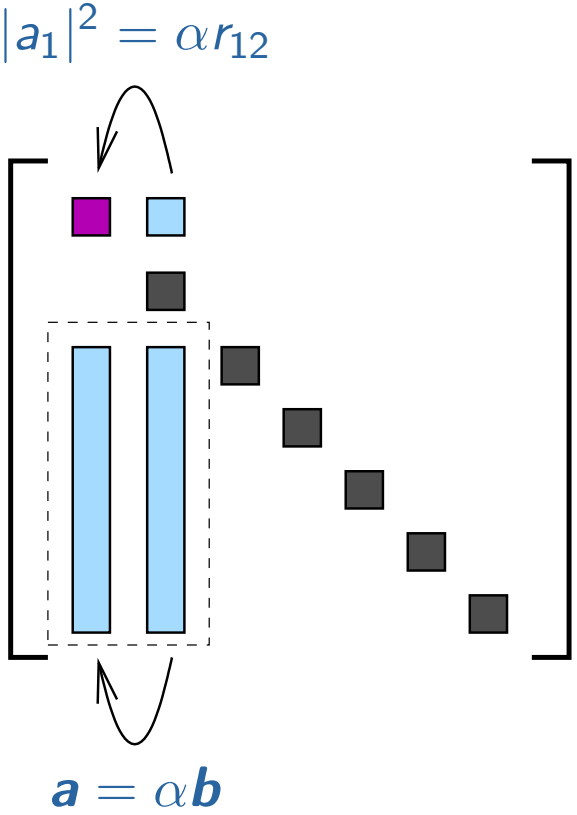
FA algorithms

Closed-form approximate solution

We can also view this as a matrix extension problem: replace the main diagonal of $\hat{\boldsymbol{R}}$ such that it becomes low rank ($= \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}$).

- Rank-1 factor model:

$$\boldsymbol{R}' = \boldsymbol{R} - \boldsymbol{D} = \boldsymbol{a}\boldsymbol{a}^{\mathrm{H}} =$$



Property: each submatrix not involving the main diagonal is rank-1. The ratios can be used to estimate the diagonal entries of $\boldsymbol{a}\boldsymbol{a}^{\mathrm{H}}$.

Intermezzo

Recap: the Kronecker product

For a matrix, $\text{vec}(\cdot)$ denotes the stacking of the columns of a matrix into a vector.

For two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the Kronecker product is defined as

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1N}\boldsymbol{B} \\ \vdots & & \vdots \\ a_{M1}\boldsymbol{B} & \cdots & a_{MN}\boldsymbol{B} \end{bmatrix},$$

Some properties:

$$\begin{aligned} (\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) &= \boldsymbol{AC} \otimes \boldsymbol{BD} \\ \text{vec}(\boldsymbol{ab}^{\mathsf{H}}) &= \boldsymbol{b}^{*} \otimes \boldsymbol{a} \\ \text{vec}(\boldsymbol{ABC}) &= (\boldsymbol{C}^{\mathsf{T}} \otimes \boldsymbol{A})\text{vec}(\boldsymbol{B}) \end{aligned}$$

Intermezzo

The variance of a sample covariance matrix

Consider an i.i.d. zero mean random sequence $\boldsymbol{x}_n$.

Let $\boldsymbol{R} = \mathrm{E}[\boldsymbol{x}_n \boldsymbol{x}_n^{\mathrm{H}}]$. We estimate $\boldsymbol{R}$ from N data samples as

$$\hat{\boldsymbol{R}} = \frac{1}{N} \sum_{n=0}^{N-1} \boldsymbol{x}_n \boldsymbol{x}_n^{\mathrm{H}}$$

Then $\mathrm{E}[\hat{\boldsymbol{R}}] = \boldsymbol{R}$: unbiased. But what is the variance of this estimate?

To define this, consider $\boldsymbol{r} = \mathrm{vec}(\boldsymbol{R})$, $\hat{\boldsymbol{r}} = \mathrm{vec}(\hat{\boldsymbol{R}})$. Then

$$\hat{\boldsymbol{r}} = \boldsymbol{r} + \boldsymbol{e}$$

where $\hat{\boldsymbol{r}}$ is a random variable, $\boldsymbol{r}$ is its expected value (mean), and $\boldsymbol{e}$ zero mean "finite sample noise". We define

$$\mathrm{cov}[\hat{\boldsymbol{R}}] =: \mathrm{cov}[\hat{\boldsymbol{r}}] = \mathrm{E}[\boldsymbol{e}\boldsymbol{e}^{\mathrm{H}}]$$

Intermezzo

The variance of a sample covariance matrix

Recall

$$\hat{\boldsymbol{R}} = \frac{1}{N}\sum \boldsymbol{x}[n]\boldsymbol{x}[n]^{\mathrm{H}} \quad \Rightarrow \quad \hat{\boldsymbol{r}} = \frac{1}{N}\sum \boldsymbol{x}_n^* \otimes \boldsymbol{x}_n$$

Use the fact that $\boldsymbol{x}_i$ is independent of $\boldsymbol{x}_j$ for $i \neq j$ to derive

$$\begin{aligned}
\mathrm{cov}[\hat{\boldsymbol{R}}] &= \mathrm{E}\left[(\frac{1}{N}\sum(\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i) - \mathrm{E}[\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i])(\frac{1}{N}\sum(\boldsymbol{x}_j^* \otimes \boldsymbol{x}_j) - \mathrm{E}[\boldsymbol{x}_j^* \otimes \boldsymbol{x}_j]\right. \\
&= \frac{1}{N^2}\sum\sum \mathrm{E}\left[(\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i - \mathrm{E}[\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i])(\boldsymbol{x}_j^* \otimes \boldsymbol{x}_j - \mathrm{E}[\boldsymbol{x}_j^* \otimes \boldsymbol{x}_j])^{\mathrm{H}}\right] \\
&= \frac{1}{N^2}\sum \mathrm{E}\left[(\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i - \mathrm{E}[\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i])(\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i - \mathrm{E}[\boldsymbol{x}_i^* \otimes \boldsymbol{x}_i])^{\mathrm{H}}\right] \\
&= \frac{1}{N}\left(\mathrm{E}[(\boldsymbol{x}^* \otimes \boldsymbol{x})(\boldsymbol{x}^* \otimes \boldsymbol{x})^{\mathrm{H}}] - \mathrm{E}[\boldsymbol{x}^* \otimes \boldsymbol{x}]\mathrm{E}[\boldsymbol{x}^* \otimes \boldsymbol{x}]^{\mathrm{H}}\right) \\
&= \frac{1}{N}\boldsymbol{C}
\end{aligned}$$

where $\boldsymbol{C} = \mathrm{E}[(\boldsymbol{x}_k^* \otimes \boldsymbol{x}_k)(\boldsymbol{x}_k^* \otimes \boldsymbol{x}_k)^{\mathrm{H}}] - \mathrm{E}[\boldsymbol{x}_k^* \otimes \boldsymbol{x}_k]\mathrm{E}[\boldsymbol{x}_k^* \otimes \boldsymbol{x}_k]^{\mathrm{H}}$

Intermezzo

The variance of a sample covariance matrix

The covariance of $\hat{\boldsymbol{R}}$ involves fourth-order correlations. These can often be described in simpler terms using cumulants.

For the special case where the entries x_i of $\boldsymbol{x}$ are zero-mean and **jointly Gaussian** distributed, it is known that (for arbitrary indices $a, b, c, d = 0, \cdots, M-1$)

$$\mathrm{E}[x_a\, x_b^*\, x_c\, x_d^*] = \mathrm{E}[x_a x_b^*]\mathrm{E}[x_c x_d^*] + \mathrm{E}[x_a x_d^*]\mathrm{E}[x_b^* x_c] + \mathrm{E}[x_a x_c]\mathrm{E}[x_b^* x_d^*]$$

"Proper" (or circularly symmetric) complex variables are such that $\mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathsf{T}}] = 0$. In this case, the last term vanishes.

Thus, for Gaussians, higher-order moments reduce to a function of second-order moments.

Intermezzo

The variance of a sample covariance matrix

Stacking in a matrix with row-index $a + M\,b$ and column-index $c + M\,d$, we can write this expression compactly as

$$\begin{aligned}\mathrm{E}[(\boldsymbol{x}^* \otimes \boldsymbol{x})(\boldsymbol{x}^* \otimes \boldsymbol{x})^{\mathrm{H}}] = \quad & \mathrm{E}[\boldsymbol{x}^* \otimes \boldsymbol{x}]\mathrm{E}[\boldsymbol{x}^* \otimes \boldsymbol{x}]^{\mathrm{H}} + \mathrm{E}[\boldsymbol{x}^*\boldsymbol{x}^{*\mathrm{H}}] \otimes \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{H}}] \\ & + \mathrm{E}[(\boldsymbol{x}^* \otimes 1)(1 \otimes \boldsymbol{x})^{\mathrm{H}}] \odot \mathrm{E}[(1 \otimes \boldsymbol{x})(\boldsymbol{x}^* \otimes 1)^{\mathrm{H}}]\end{aligned}$$

For proper complex variables, the last term vanishes, and thus, for **zero mean proper complex-valued Gaussian random variables**,

$$\mathrm{cov}[\hat{\boldsymbol{R}}] = \frac{1}{N}\,\boldsymbol{R}^* \otimes \boldsymbol{R}$$

FA algorithms

Parametrization

We parametrize $\boldsymbol{R} = \boldsymbol{R}(\theta) = \boldsymbol{A}\boldsymbol{A}^{\mathsf{H}} + \boldsymbol{D}$ using parameters

$$\theta = \begin{bmatrix} \theta_{\boldsymbol{A}} \\ \theta_{\boldsymbol{A}^*} \\ \theta_{\boldsymbol{D}} \end{bmatrix} = \begin{bmatrix} \mathrm{vec}(\boldsymbol{A}) \\ \mathrm{vec}(\boldsymbol{A}^*) \\ \mathrm{diag}(\boldsymbol{D}) \end{bmatrix}$$

These are redundant, but the constraints can be implemented later (after estimating θ).

The parameters are **complex** and we use **Wirtinger calculus** to differentiate. Essentially: treat θ and θ^* as independent.

The Jacobian $\boldsymbol{J}(\theta)$ for $\boldsymbol{R}(\theta)$ is given by

$$\boldsymbol{J} = \frac{\partial \mathrm{vec}(\boldsymbol{R})}{\partial \theta^{\mathsf{T}}} = [\boldsymbol{J}_{\boldsymbol{A}}, \quad \boldsymbol{J}_{\boldsymbol{A}^*}, \quad \boldsymbol{J}_{\boldsymbol{D}}] = [\boldsymbol{A}^* \otimes \boldsymbol{I}, \quad (\boldsymbol{I} \otimes \boldsymbol{A})\boldsymbol{K}, \quad \boldsymbol{I} \circ \boldsymbol{I}]$$

where $\circ$ denotes the Khatri-Rao product (column-wise Kronecker product), and $\mathrm{vec}(\boldsymbol{A}^{\mathsf{T}}) = \boldsymbol{K}\mathrm{vec}(\boldsymbol{A})$.

FA algorithms

Non-Linear Weighted Least Squares

Define $\boldsymbol{r} = \text{vec}(\boldsymbol{R})$, then we can formulate a covariance matching problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; \|\boldsymbol{W}^{1/2}[\hat{\boldsymbol{r}} - \boldsymbol{r}(\boldsymbol{\theta})]\|^2 = \arg\min_{\boldsymbol{\theta}} \; [\hat{\boldsymbol{r}} - \boldsymbol{r}(\boldsymbol{\theta})]^{\text{H}} \boldsymbol{W} [\hat{\boldsymbol{r}} - \boldsymbol{r}(\boldsymbol{\theta})]$$

where $\boldsymbol{W}$ is a weighting matrix. The optimum weight is the inverse of the covariance matrix of $\hat{\boldsymbol{r}}$, for proper Gaussian-distributed data estimated as

$$\boldsymbol{W} = \hat{\boldsymbol{R}}^{-T} \otimes \hat{\boldsymbol{R}}^{-1}$$

NOte that we use $\hat{\boldsymbol{R}}$ instead of $\boldsymbol{R}$. Asymptotically (for large N), this converges to the ML solution.

FA algorithms

Gauss-Newton for solving Non-Linear WLS

The Gauss-Newton technique for solving nonlinear WLS is the iteration

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \mu^{(k)} \boldsymbol{\delta}$$

where the direction of descent $\boldsymbol{\delta}$ is found by solving

$$\boldsymbol{B}(\boldsymbol{\theta}^{(k)})\, \boldsymbol{\delta} = \boldsymbol{g}(\boldsymbol{\theta}^{(k)})$$

$\boldsymbol{g}(\boldsymbol{\theta})$ is the gradient of the cost function, and $\boldsymbol{B}(\boldsymbol{\theta})$ is the Gramian of the Jacobians (approximating the Hessian):

$$\boldsymbol{B}(\boldsymbol{\theta}) = \boldsymbol{J}^{\mathrm{H}}(\boldsymbol{\theta}) \boldsymbol{W} \boldsymbol{J}(\boldsymbol{\theta}), \qquad \boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{J}^{\mathrm{H}}(\boldsymbol{\theta}) \boldsymbol{W} [\hat{\boldsymbol{r}} - \boldsymbol{r}(\boldsymbol{\theta})]$$

FA algorithms

Closed-form solution for direction of descent

$\boldsymbol{B}(\theta)$ can be very large: hard to construct and invert.

A complicated derivation shows how we can solve for $\delta_{\boldsymbol{D}}$ inside δ in closed form. Define

$$\tilde{\boldsymbol{W}} = \hat{\boldsymbol{R}}^{-1} - \hat{\boldsymbol{R}}^{-1}\boldsymbol{A}(\boldsymbol{A}^{\mathsf{H}}\hat{\boldsymbol{R}}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathsf{H}}\hat{\boldsymbol{R}}^{-1} \qquad [\tilde{\boldsymbol{W}}\boldsymbol{A} = 0]$$

$$\tilde{\boldsymbol{B}}_{\boldsymbol{D}} = \boldsymbol{J}_{\boldsymbol{D}}^{\mathsf{H}}\left(\tilde{\boldsymbol{W}}^{\mathsf{T}} \otimes \tilde{\boldsymbol{W}}\right)\boldsymbol{J}_{\boldsymbol{D}} \quad = \tilde{\boldsymbol{W}}^{\mathsf{T}} \odot \tilde{\boldsymbol{W}} \qquad [\boldsymbol{J}_{\boldsymbol{D}} = \boldsymbol{I} \circ \boldsymbol{I}]$$

$$\tilde{\boldsymbol{g}}_{\boldsymbol{D}} = \boldsymbol{J}_{\boldsymbol{D}}^{\mathsf{H}}\left(\tilde{\boldsymbol{W}}^{\mathsf{T}} \otimes \tilde{\boldsymbol{W}}\right)\mathrm{vec}[\hat{\boldsymbol{R}} - \boldsymbol{R}(\theta)]$$

Then the computation of

$$\delta = \begin{bmatrix} \mathrm{vec}(\boldsymbol{\Delta}_{\boldsymbol{A}}) \\ \mathrm{vec}(\boldsymbol{\Delta}_{\boldsymbol{A}^*}) \\ \delta_{\boldsymbol{D}} \end{bmatrix}$$

reduces to the computation of $\delta_{\boldsymbol{D}}$ from $\tilde{\boldsymbol{B}}_{\boldsymbol{D}}\delta_{\boldsymbol{D}} = \tilde{\boldsymbol{g}}_{\boldsymbol{D}}$, while

$$\boldsymbol{\Delta}_{\boldsymbol{A}} = \frac{1}{2}(\boldsymbol{I} + \hat{\boldsymbol{R}}\tilde{\boldsymbol{W}})(\hat{\boldsymbol{R}} - \boldsymbol{R}(\theta) - \mathrm{diag}(\delta_{\boldsymbol{D}}))\hat{\boldsymbol{R}}^{-1}\boldsymbol{A}(\boldsymbol{A}^{\mathsf{H}}\hat{\boldsymbol{R}}^{-1}\boldsymbol{A})^{-1}$$

FA algorithms

Alternating WLS method

If we take step size $\mu = 1$, then the closed-form result simplifies.

$[\cdots]$

The result is the Alternating Weighted Least Squares (AWLS) algorithm (cf the Ad-Hoc algorithm), where $\boldsymbol{W} = \hat{\boldsymbol{R}}^{-1}$:

$$\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{H}} := \boldsymbol{D}_{(k)}^{-1/2}\hat{\boldsymbol{R}}\boldsymbol{D}_{(k)}^{-1/2} \qquad \text{[EVD; prewhitening of } \hat{\boldsymbol{R}}\text{]}$$

$$\boldsymbol{A}_{(k+1)} := \boldsymbol{D}_{(k)}^{1/2}\boldsymbol{U}_0(\boldsymbol{\Lambda}_0 - \boldsymbol{I})^{1/2}$$

$$\tilde{\boldsymbol{W}} := \boldsymbol{W} - \boldsymbol{W}\boldsymbol{A}_{(k+1)}(\boldsymbol{A}_{(k+1)}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{A}_{(k+1)})^{-1}\boldsymbol{A}_{(k+1)}^{\mathsf{H}}\boldsymbol{W}$$

$$\boldsymbol{d}_{(k+1)} := \left[\tilde{\boldsymbol{W}}^{\mathsf{T}} \odot \tilde{\boldsymbol{W}}\right]^{-1} \operatorname{vecdiag}(\tilde{\boldsymbol{W}})$$

$$\boldsymbol{D}_{(k+1)} := \operatorname{diag}(\boldsymbol{d}_{(k+1)})$$

Extensions

Extended FA

Let $\boldsymbol{M}$ with $m_{ij} \in \{0, 1\}$ be a "masking matrix". Then

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{A}^{\mathsf{H}} + \boldsymbol{\Psi}\,, \qquad \boldsymbol{\Psi} = \boldsymbol{M} \odot \boldsymbol{\Psi}$$

This generalizes $\boldsymbol{D}$ to a more general structure.

This can be further generalized to linear models of the form $\mathrm{vec}(\boldsymbol{\Psi}) = \boldsymbol{G}\boldsymbol{\theta}_{\psi}$, where $\boldsymbol{G}$ is a fixed basis (e.g. selected columns of a Fourier matrix to model spatially lowpass noise).

Joint FA

Suppose we have multiple "snapshots" $\boldsymbol{R}_m$, with different $\boldsymbol{A}_m$ but a common noise covariance matrix,

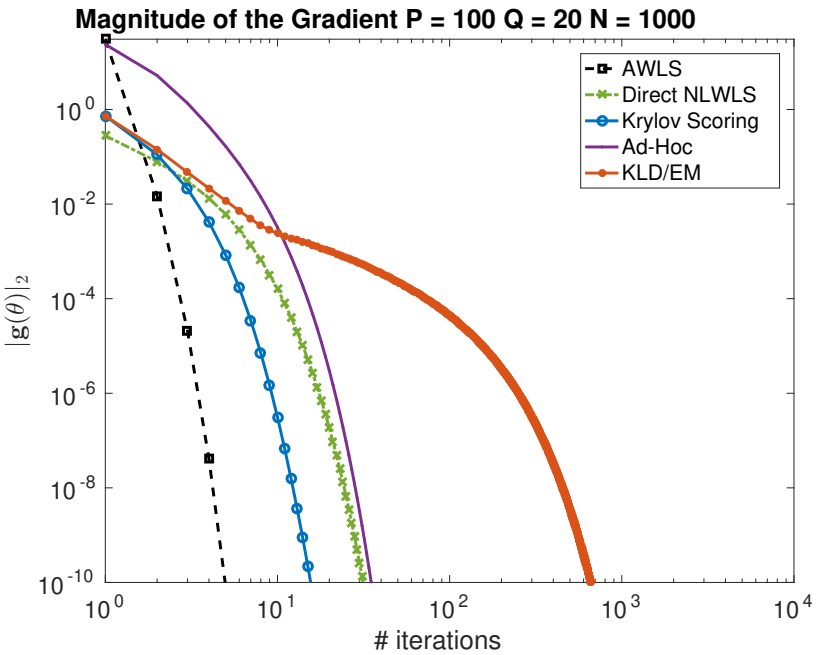
$$\boldsymbol{R}_m = \boldsymbol{A}_m \boldsymbol{A}_m^{\mathsf{H}} + \boldsymbol{D}, \quad m = 1, \ldots, M$$

Simulations

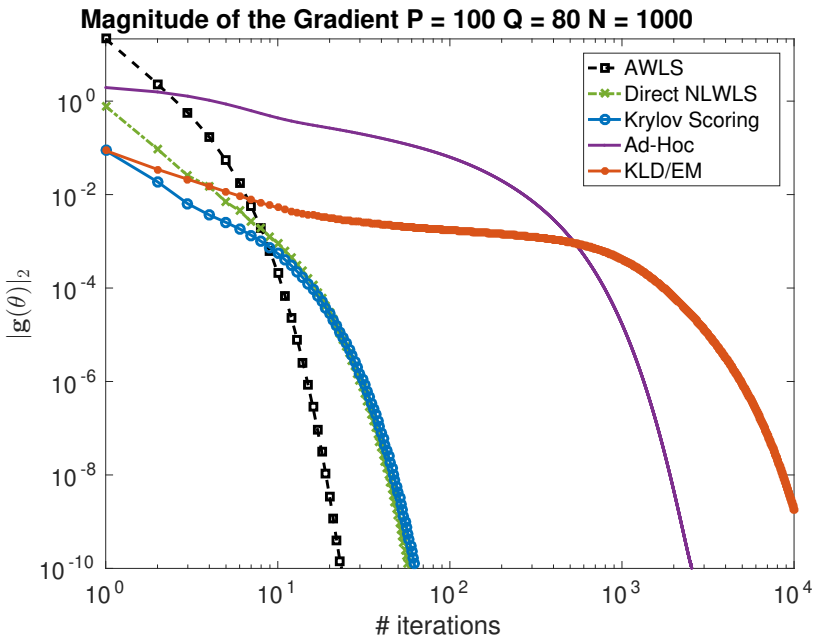
Convergence

Convergence speed for $P = 100$ sensors, $N = 1000$ samples

$Q = 20$ sources:



$Q = 80$ sources:

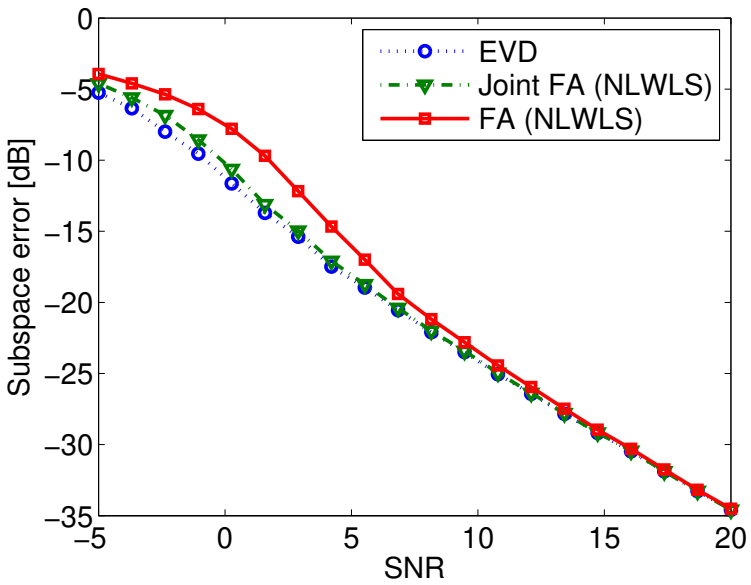


Simulations

Subspace Estimation Performance

Subspace error as function of SNR for $\boldsymbol{\Psi} = \boldsymbol{I}$ (white noise).

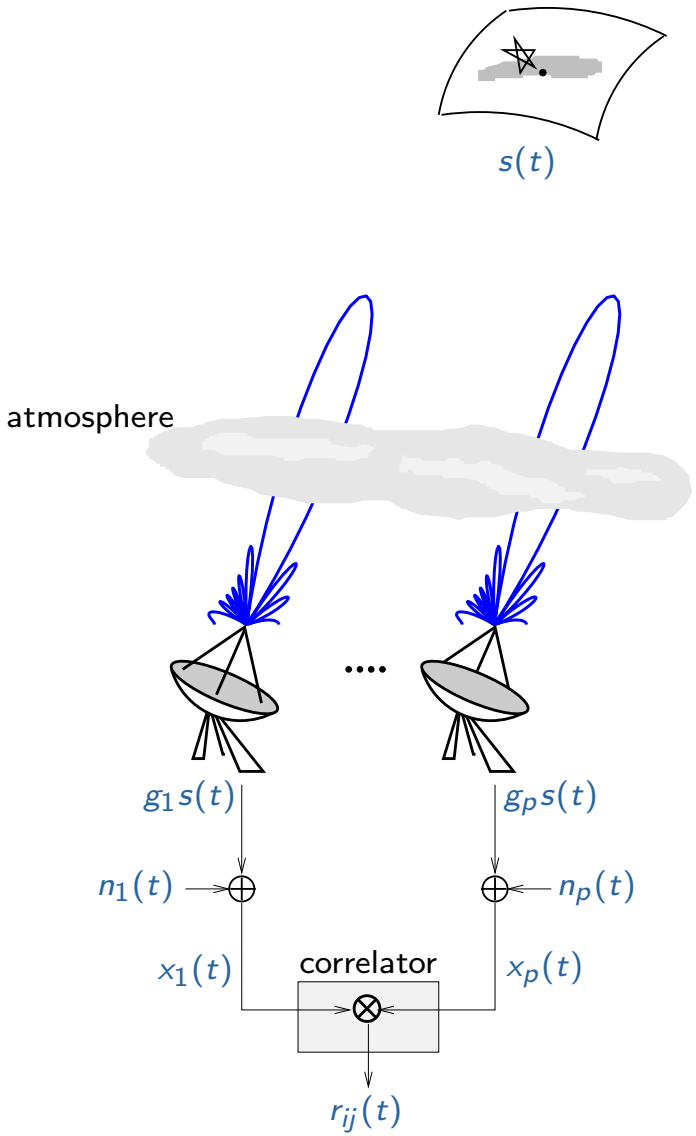
$Q_m = 2$, $P = 5$, $M = 5$, $N = 100$



For low SNR, there is some drop in performance in particular for non-joint processing. For higher SNR, there is no performance penalty.

Applications - 1 (FA)

Gain and noise power calibration at the Westerbork Radio Telescope

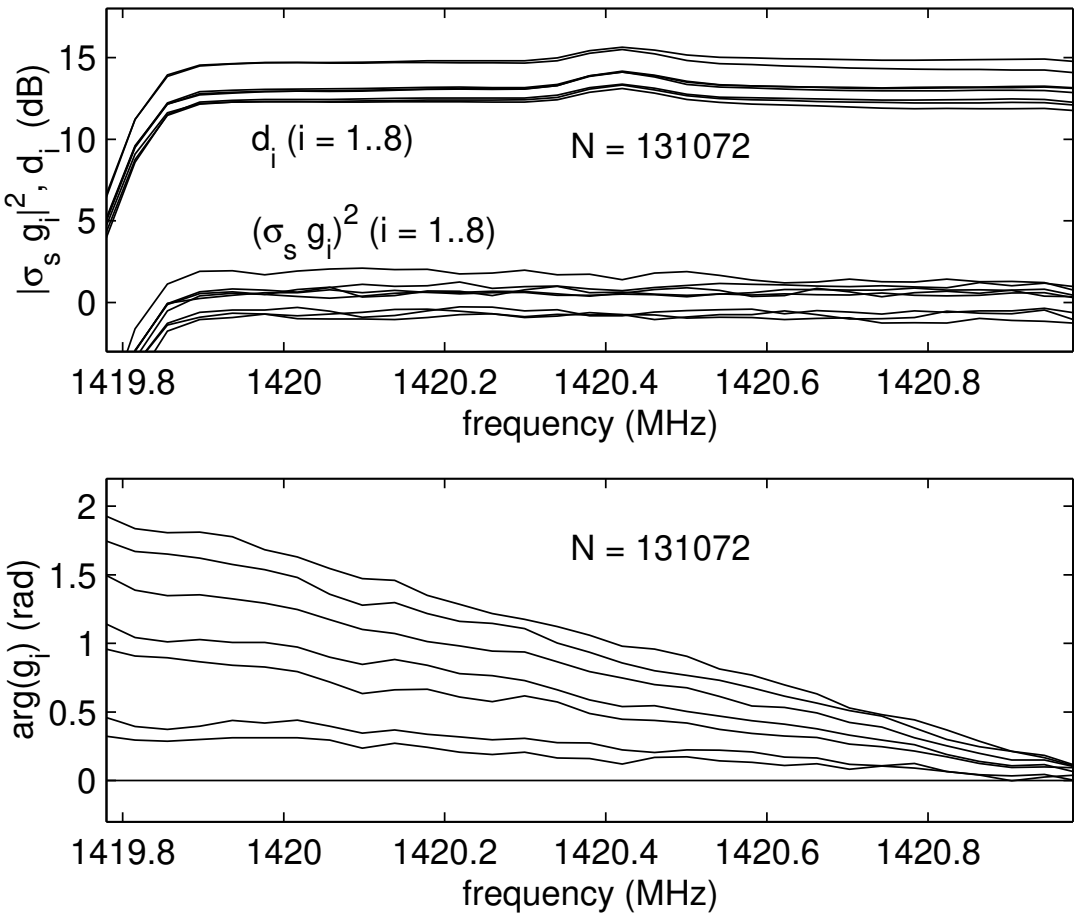


- General calibration strategy: observe a single 'strong' sky source
- Stack the p antenna signals in a vector $\boldsymbol{x}$ and analyze $\boldsymbol{R} = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{H}}]$ with model

$$\boldsymbol{R} = \boldsymbol{g}\sigma_s^2\boldsymbol{g}^{\mathrm{H}} + \boldsymbol{D}$$

- Need to estimate the complex gains g_i and noise powers in $\boldsymbol{D}$.

Applications - 1 (FA)



Estimates of (a) gain magnitude, noise powers, and (b) gain phase, as function of frequency, around $f = 1420$ MHz. Based on observation of the astronomical source 3C48. Frequencies processed independently.

Applications - 2 (JFA)

Narrowband interference subspace estimation at the Westerbork Radio Telescope

For each frequency f, we have a sequence of short-term correlation matrices $\hat{\boldsymbol{R}}_{k,f}$ (averaged to 10 ms) with model

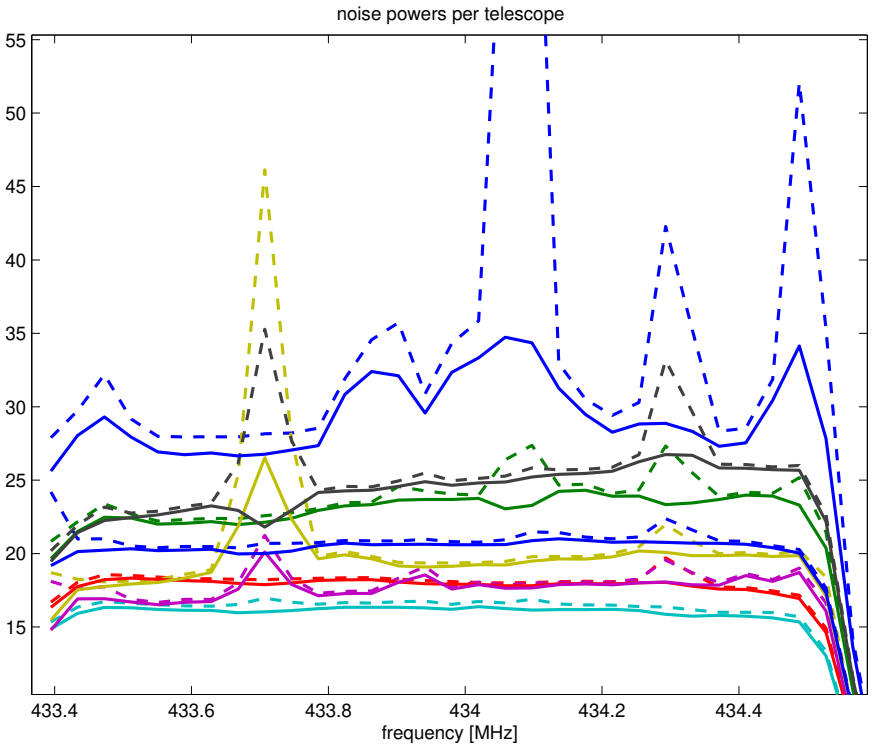
$$\boldsymbol{R}_{k,f} = \boldsymbol{A}_{k,f}\boldsymbol{A}_{k,f}^{\mathsf{H}} + \boldsymbol{D}_f, \qquad k = 1, 2, \cdots, 1000$$

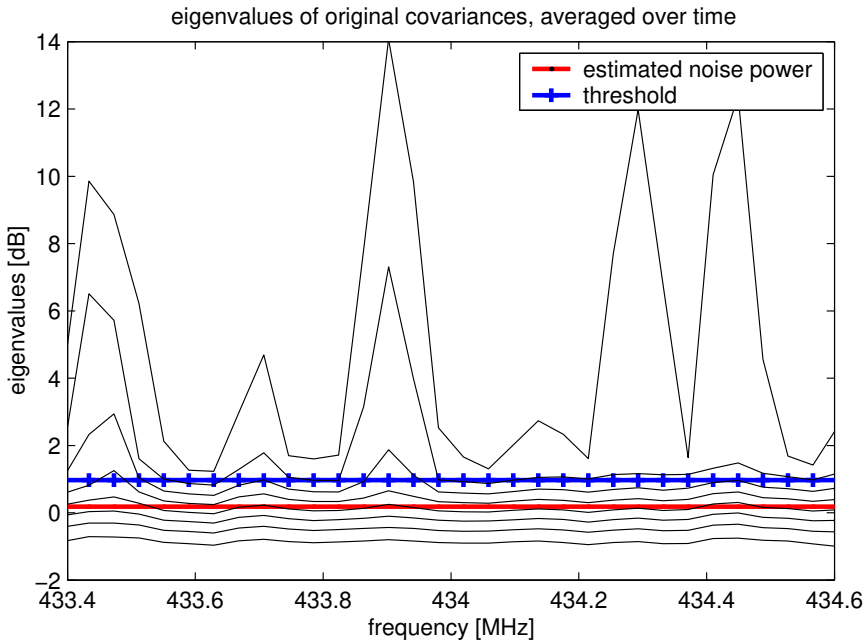
- $\boldsymbol{A}_{k,f}$: interference subspace at time k, frequency f (32 frequency bins)
- $\boldsymbol{D}_f$: noise power estimate at a single frequency – to be calibrated

For each frequency, we use JFA to estimate the factors and noise powers.

Applications - 2 (JFA)

Amateur AM radio broadcast interference at 434 MHz, both continuous and intermittent, recorded at the WSRT pointed at 3C48:

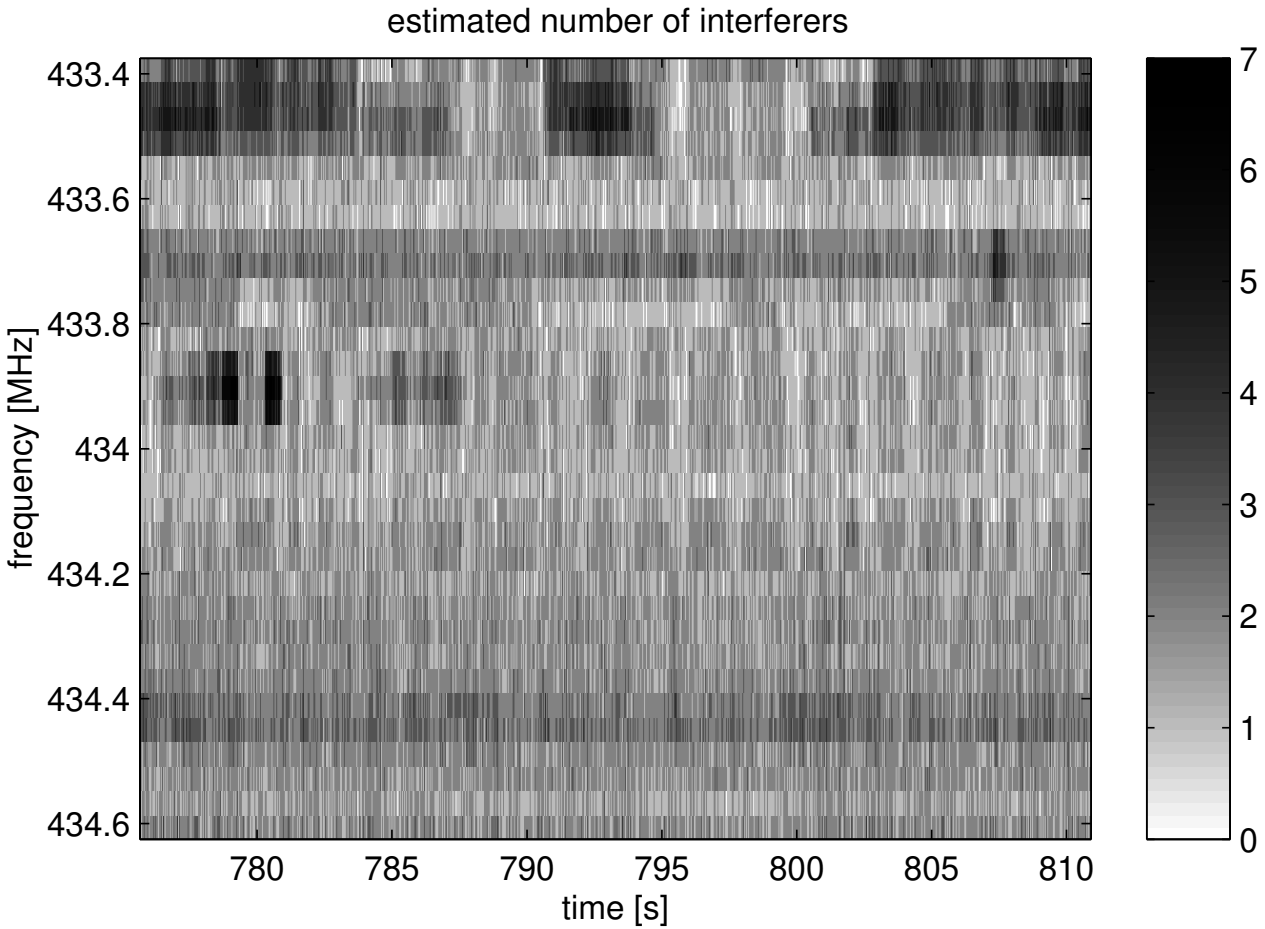




(*a*) estimated noise powers, (*b*) averaged eigenvalues (after whitening)

Applications - 2 (JFA)

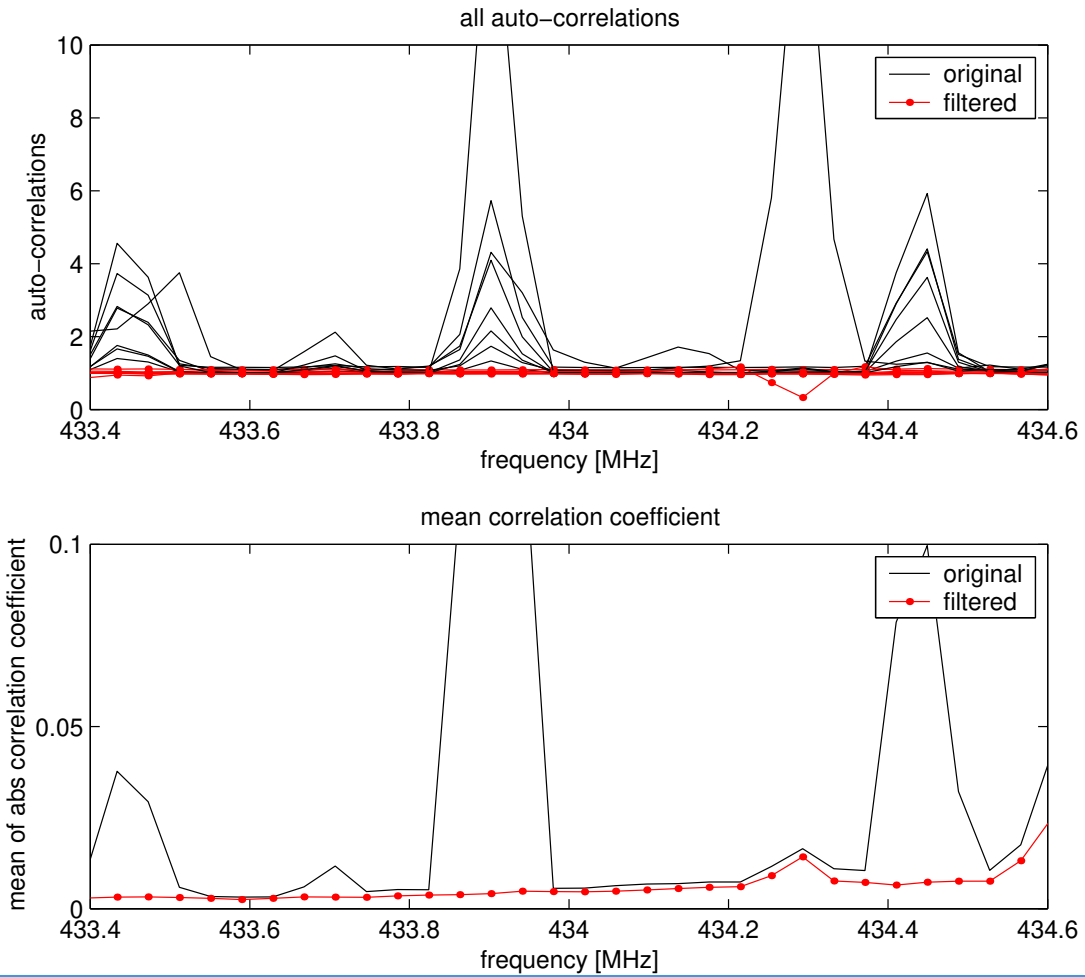
Estimated number of interferers



Applications - 2 (JFA)

Correlation spectra after interference cancellation

The interference subspaces are projected out, and the results are further averaged over time.



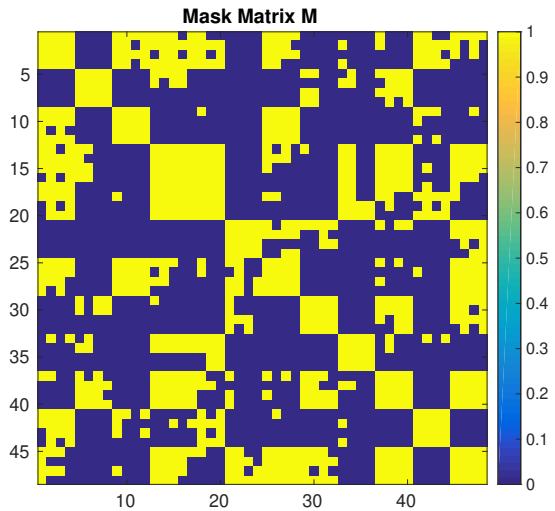
Applications - 3 (EFA)

Suppression of the Milky Way at a LOFAR station

Using a LOFAR LBA station, we observe two strong sources, Cas A and Cyg A, and the Milky Way (cloud-like emission).

The Milky way is modeled by $\boldsymbol{R}_n$ with mask $\boldsymbol{M}$ incorporating all baselines shorter than 4 wavelengths.

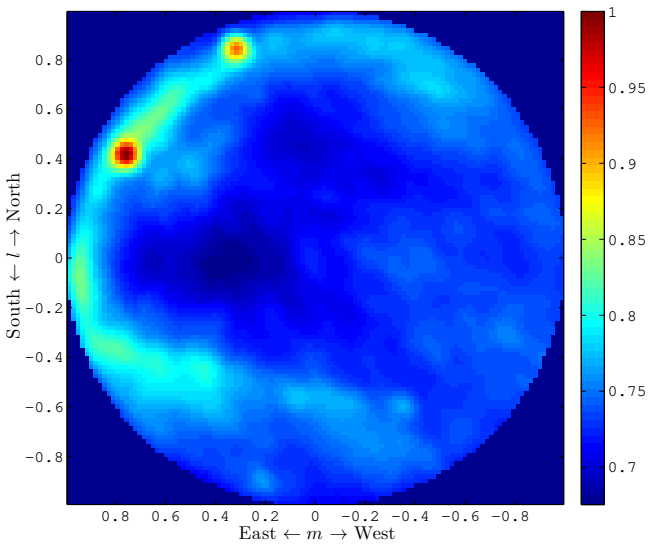
After estimating $\boldsymbol{R}_n$ using Extended Factor Analysis, the residual $\hat{\boldsymbol{R}}_0 = \hat{\boldsymbol{R}} - \boldsymbol{R}_n$ is imaged.



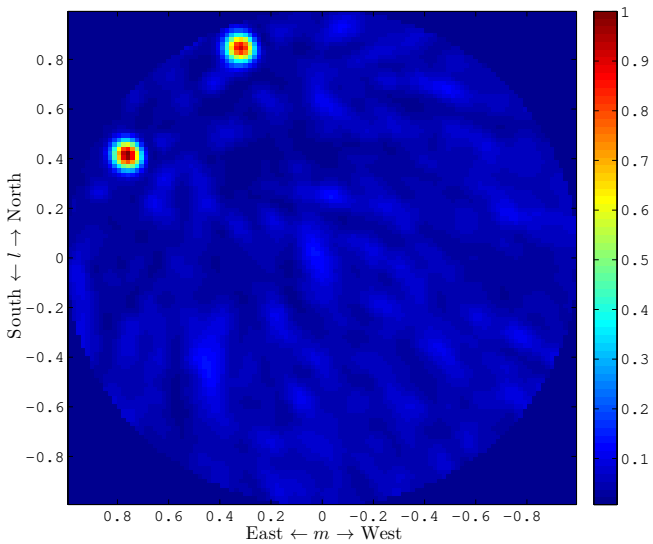
Applications - 3 (EFA)

Total sky image for 1 station using classical DFT beamforming ("dirty image") combining data from 24 156kHz subbands distributed between 45.3 and 67.3 MHz and 10 seconds of integration per channel.

DFT Imaging without FA



DFT Imaging with Extended FA

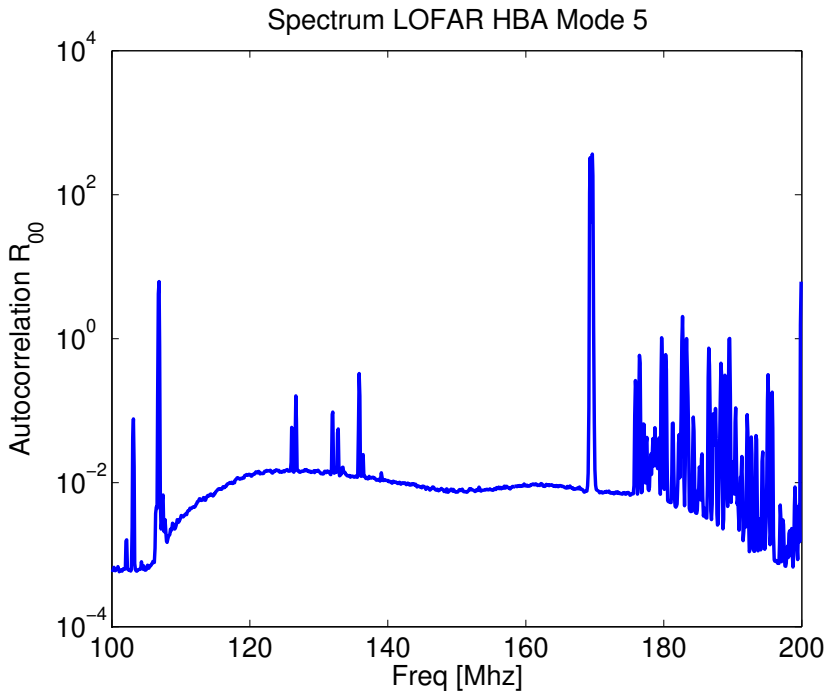


Next, it is possible to calibrate the gains of the individual antennas using the (known) intensities of the point sources.

Applications - 4

Spatial Filtering at a LOFAR station

Data from the LOFAR station RS409 in HBA mode 5 (100-200 MHz), tracking the strong astronomical source Cyg A.



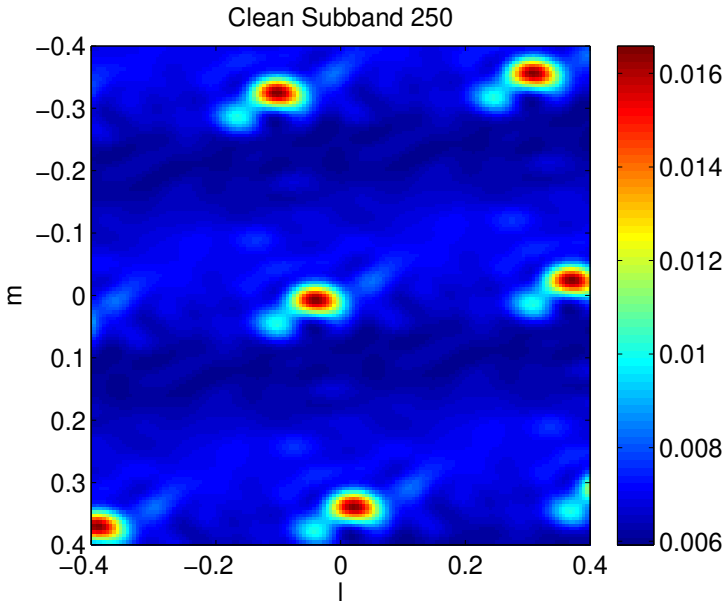
Above 174 MHz, the spectrum is heavily contaminated by wideband DAB transmissions.

Applications - 4

Snapshot image of the sky

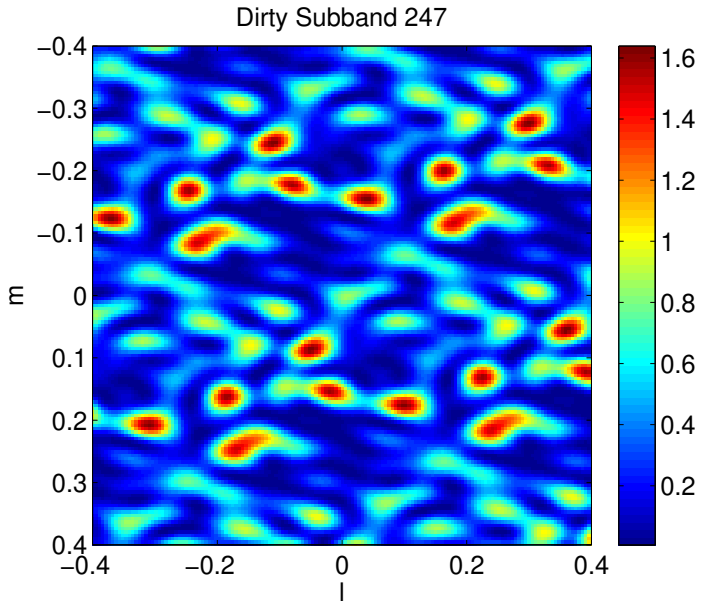
Uncontaminated image

subband 250 at 175.59 MHz



RFI-contaminated data

subband 247 at 175.88 MHz



Applications - 4

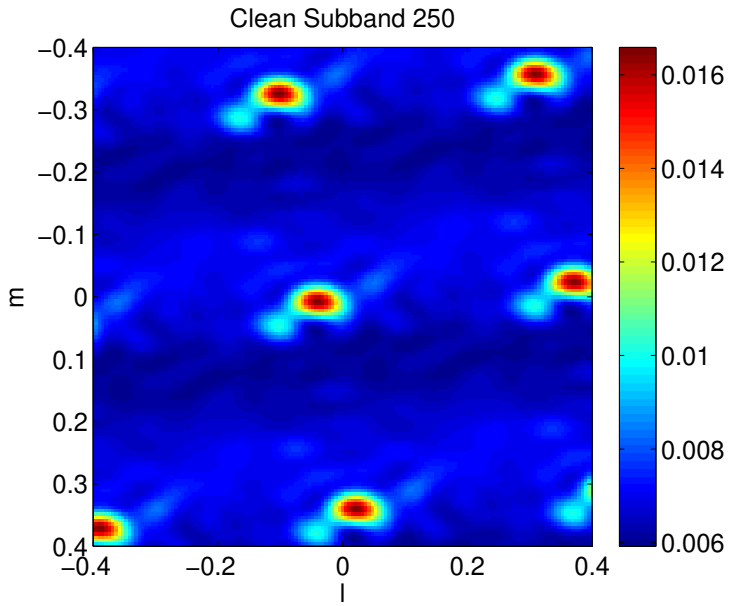
For spatial filtering, we take 6 of the 46 receiving elements as reference array. The covariance model is

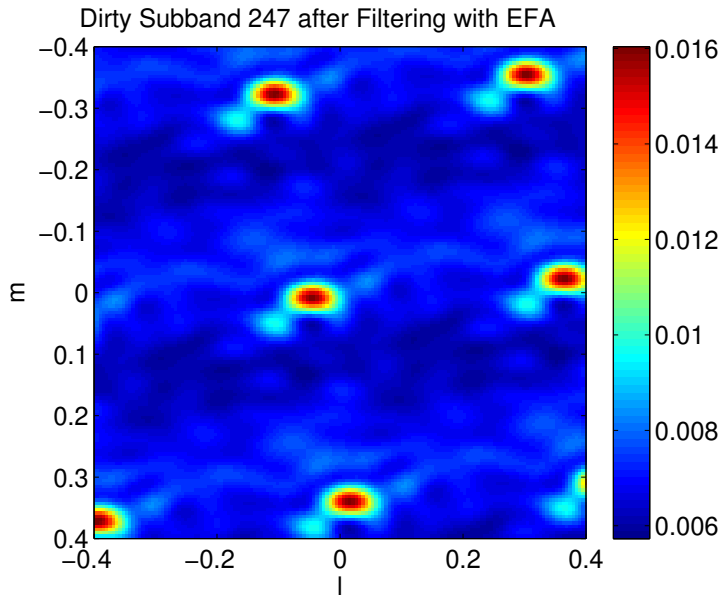
$$\boldsymbol{R}_k = \boldsymbol{A}_k \boldsymbol{A}_k^{\mathsf{H}} + \boldsymbol{\Psi} = \boldsymbol{A}_k \boldsymbol{A}_k^{\mathsf{H}} + \left[\begin{array}{c|c} \boldsymbol{\Psi}_{00} & 0 \\ \hline 0 & \boldsymbol{\Sigma}_1 \end{array}\right], \qquad \boldsymbol{M} = \left[\begin{array}{c|c} 11^{\mathsf{T}} & 0 \\ \hline 0 & \boldsymbol{I} \end{array}\right]$$

$\boldsymbol{\Psi}_{00}$ contains the astronomical covariances of interest. The mask $\boldsymbol{M}$ is used to avoid a subspace model for these. $\boldsymbol{A}_k$ is the subspace of the interference.

Applications - 4

EFA is used to estimate the subspace of the interference. This is then used as spatial filter on the 40 remaining antennas (primary array).





Summary

Factor Analysis is viewed as an extension of the eigenvalue decomposition for nonwhite noise.

We proposed:

- New algorithms based on Gauss-Newton iterations
- Extensions of FA to multiple matrices and more general noise models

The algorithms show reliable and efficient convergence, feasible for moderately large problem sizes ($P = 100$ sensors).

Even if the noise is white, the performance penalty with respect to EVD is minor.