*Astronomy & Astrophysics* manuscript no. CLS_final
January 8, 2016


# Radio astronomical image formation using constrained least squares and Krylov subspaces

Ahmad Mouri Sardarabadi[1*], Amir Leshem[2], and Alle-Jan van der Veen[1 ⋆]

[1]

January 8, 2016

**ABSTRACT**

*Aims.* Image formation for radio astronomy can be defined as estimating the spatial intensity distribution of celestial sources throughout the sky, given an array of antennas. One of the challenges with image formation is that the problem becomes ill-posed as the number of pixels becomes large. The introduction of constraints that incorporate a priori knowledge is crucial.

*Methods.* In this paper we show that in addition to non-negativity, the magnitude of each pixel in an image is also bounded from above. Indeed, the classical "dirty image" is an upper bound, but a much tighter upper bound can be formed from the data using array processing techniques. This formulates image formation as a least squares optimization problem with inequality constraints. We propose to solve this constrained least squares problem using active set techniques, and the steps needed to implement it are described. It is shown that the least squares part of the problem can be efficiently implemented with Krylov-subspace-based techniques. We also propose a method for correcting for the possible mismatch between source positions and the pixel grid. This correction improves both the detection of sources and their estimated intensities. The performance of these algorithms is evaluated using simulations.

*Results.* Based on parametric modeling of the astronomical data, a new imaging algorithm based on convex optimization, active sets, and Krylov-subspace-based solvers is presented. The relation between the proposed algorithm and sequential source removing techniques is explained, and it gives a better mathematical framework for analyzing existing algorithms. We show that by using the structure of the algorithm, an efficient implementation that allows massive parallelism and storage reduction is feasible. Simulations are used to compare the new algorithm to classical CLEAN. Results illustrate that for a discrete point model, the proposed algorithm is capable of detecting the correct number of sources and producing highly accurate intensity estimates.

**Key words.** Interferometeres – Numerical Method – Image Processsing

## 1. Introduction

Image formation for radio astronomy can be defined as estimating the spatial intensity distribution of celestial sources over the sky. The measurement equation ("data model") is linear in the source intensities, and the resulting least squares problem has classically been implemented in two steps: formation of a "dirty image", followed by a deconvolution step. In this process, an implicit model assumption is made that the number of sources is discrete, and subsequently the number of sources has been replaced by the number of image pixels (assuming each pixel may contain a source).

The deconvolution step becomes ill-conditioned if the number of pixels is large (Wijnholds & van der Veen 2008). Alternatively, the directions of sources may be estimated along with their intensities, but this is a complex nonlinear problem. Classically, this has been implemented as an iterative subtraction technique, wherein source directions are estimated from the dirty image, and their contribution is subtracted from the data. This mixed approach is the essence of the CLEAN method proposed by Högbom (Högbom 1974), which was subsequently refined and extended in several ways, leading to the widely used approaches described in (Cornwell 2008; Rau et al. 2009; Bhatnager & Cornwell 2004).

The conditioning of the image deconvolution step can be improved by incorporating side information such as non-negativity of the image (Briggs 1995), source model structure beyond simple point sources (e.g., shapelets and wavelets (Reid 2006)), sparsity or $\ell_1$ constraints on the image (Levanda & Leshem 2008; Wiaux et al. 2009), or a combination of both wavelets and sparsity (Carrillo et al. 2012, 2014). Beyond these, some fundamental approaches based on parameter estimation techniques have been proposed, such as the least squares minimum variance imaging (LS-MVI) (Ben-David & Leshem 2008), maximum-likelihood -based techniques (Leshem & van der Veen 2000), and Bayesian-based techniques (H. Junklewitz et al. 2015; Lochner et al. 2015). Computational complexity is a concern that has not been addressed in these approaches.

New radio telescopes such as the Low Frequency Array (LOFAR), the Allen Telescope Array (ATA), Murchison Widefield Array (MWA), and the Long Wavelength Array (LWA) are composed of many stations (each station made

⋆ This research was supported by NWO-TOP 2010, 614.00.005. The research of A. Leshem was supported by the Israeli Science foundation, grant 1240-2009. [1] Department of electrical engineering, Delft University of Technology. [2] Faculty of Engineering, Bar-Ilan University. *Corresponding author, email: a.mourisardarabadi@tudelft.nl

up of multiple antennas that are combined using adaptive beamforming), and the increase in number of antennas and stations continues in the design of the square kilometer array (SKA). These instruments have or will have a significantly increased sensitivity and a larger field of view compared to traditional telescopes, leading to many more sources that need to be considered. They also need to process larger bandwidths to reach this sensitivity. Besides the increased requirements on the performance of imaging, the improved spatial resolution leads to an increasing number of pixels in the image, and the development of computationally efficient techniques is critical.

To benefit from the vast literature related to solving least squares problems, but also to gain from the nonlinear processing offered by standard deconvolution techniques, we propose to reformulate the imaging problem as a parameter-estimation problem described by a weighted least squares optimization problem with several constraints. The first is a non-negativity constraint, which would lead to the non-negative least squares algorithm (NNLS) proposed in (Briggs 1995). But we show that the pixel values are also bounded from above. A coarse upper bound is provided by the classical dirty image, and a much tighter bound is the "minimum variance distortionless response" (MVDR) dirty image that was proposed in the context of radio astronomy in (Leshem & van der Veen 2000).

We propose to solve the resulting constrained least squares problems using an active set approach. This results in a computationally efficient imaging algorithm that is closely related to existing nonlinear sequential source estimation techniques such as CLEAN with the benefit of accelerated convergence thanks to tighter upper bounds on the intensity over the complete image. Because the constraints are enforced over the entire image, this eliminates the inclusion of negative flux sources and other anomalies that appear in some existing sequential techniques.

To reduce the computational complexity further, we show that the data model has a Khatri-Rao structure. This can be exploited to significantly improve the data management and parallelism compared to general implementations of least squares algorithms.

The structure of the paper is as follows. In Sec. 2 we describe the basic data model and the image formation problem in Sec. 3. A constrained least squares problem is formulated, using various intensity constraints that take the form of dirty images. The solution of this problem using active set techniques in Sec. 4 generalizes the classical CLEAN algorithm. In Sec. 5 we discuss the efficient implementation of a key step in the active set solution using Krylov subspaces. We end up with some simulated experiments that demonstrate the advantages of the proposed technique and conclusions regarding future implementation.

### *Notation*

A boldface letter such as $\mathbf{a}$ denotes a column vector, a boldface capital letter such as $\mathbf{A}$ denotes a matrix. Then $a_{ij} = [\mathbf{A}]_{ij}$ corresponds to the entry of $\mathbf{A}$ in the $i$th row and $j$th column, $\mathbf{a}_i = [\mathbf{A}]_i$ is the $i$th column of $\mathbf{A}$, $a_i$ is the $i$th element of the vector $\mathbf{a}$, $\mathbf{I}$ is an identity matrix of appropriate size, and $\mathbf{I}_p$ is a $p \times p$ identity matrix.

The symbol $(\cdot)^T$ is the transpose operator, $(\cdot)^*$ is the complex conjugate operator, $(\cdot)^H$ the Hermitian transpose, $\|\cdot\|_F$ the Frobenius norm of a matrix, $\|.\|$ the two norm of a vector, $\mathcal{E}\{\cdot\}$ the expectation operator ,and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the multivariate complex normal distribution with expected value $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

A tilde, $\tilde{.}$, denotes parameters and related matrices that depend on the "true" direction of the sources. However, in most of the paper, we work with parameters that are discretized on a grid, in which case we drop the tilde. The grid points correspond to the image pixels and do not necessarily coincide with the actual positions of the sources.

A calligraphic capital letter such as $\mathcal{X}$ represents a set of indices, and $\mathbf{a}_{\mathcal{X}}$ is a column vector constructed by stacking the elements of $\mathbf{a}$ that belong to $\mathcal{X}$. The corresponding indices are stored with the vector as well (similar to the storage of matlab "sparse" vectors).

The operator vect$(\cdot)$ stacks the columns of the argument matrix to form a vector, vectdiag$(\cdot)$ stacks the diagonal elements of the argument matrix to form a vector, and diag$(\cdot)$ is a diagonal matrix with its diagonal entries from the argument vector (if the argument is a matrix diag$(\cdot)$ = diag(vectdiag$(\cdot)$)).

Let $\otimes$ denote the Kronecker product, i.e.,

$$\mathbf{A} \otimes \mathbf{B} := \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Furthermore, $\circ$ denotes the Khatri-Rao product (columnwise Kronecker product), i.e.,

$$\mathbf{A} \circ \mathbf{B} := [\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, \cdots],$$

and $\odot$ denotes the Schur-Hadamard (elementwise) product. The following properties are used throughout the paper (for matrices and vectors with compatible dimensions):

$$(\mathbf{B}^T \otimes \mathbf{A})\text{vect}(\mathbf{X}) = \text{vect}(\mathbf{AXB}) \quad (1)$$

$$(\mathbf{B} \otimes \mathbf{A})^H = (\mathbf{B}^H \otimes \mathbf{A}^H) \quad (2)$$

$$(\mathbf{B} \otimes \mathbf{A})^{-1} = (\mathbf{B}^{-1} \otimes \mathbf{A}^{-1}) \quad (3)$$

$$(\mathbf{B}^T \circ \mathbf{A})\mathbf{x} = \text{vect}(\mathbf{A}\text{diag}(\mathbf{x})\mathbf{B}) \quad (4)$$

$$(\mathbf{BC} \otimes \mathbf{AD}) = (\mathbf{B} \otimes \mathbf{A})(\mathbf{C} \otimes \mathbf{D}) \quad (5)$$

$$(\mathbf{BC} \circ \mathbf{AD}) = (\mathbf{B} \otimes \mathbf{A})(\mathbf{C} \circ \mathbf{D}) \quad (6)$$

$$(\mathbf{B}^H\mathbf{C} \odot \mathbf{A}^H\mathbf{D}) = (\mathbf{B} \circ \mathbf{A})^H(\mathbf{C} \circ \mathbf{D}) \quad (7)$$

$$\text{vectdiag}(\mathbf{A}^H\mathbf{XA}) = (\mathbf{A}^* \circ \mathbf{A})^H\text{vect}(\mathbf{X}). \quad (8)$$

## 2. Data model

We consider an instrument where $P$ receivers (stations or antennas) observe the sky. Assuming a discrete point source model, we let $Q$ denote the number of visible sources. The received signals at the antennas are sampled and subsequently split into narrow sub-bands. For simplicity, we consider only a single sub-band in the rest of the paper. Although the sources are considered stationary, the apparent position of the celestial sources will change with time because of the earth's rotation. For this reason the data is split into short blocks or "snapshots" of $N$ samples, where the exact value of $N$ depends on the resolution of the instrument.

We stack the output of the $P$ antennas at a single subband into a vector $\mathbf{y}_k[n]$, where $n = 1, \cdots, N$ denotes the

sample index, and $k = 1, \cdots, K$ denotes the snapshot index. The signals of the $q$th source arrive at the array with slight delays for each antenna that depend on the source direction and the Earth's rotation (the geometric delays), and for sufficiently narrow sub-bands these delays become phase shifts, i.e., multiplications by complex coefficients. The coefficients are later stacked into the so-called array response vector. To describe this vector, we first need to define a coordinate system. Assume a fixed coordinate system based on the right ascension ($\alpha$) and declination ($\delta$) of a source, and define the corresponding direction vector

$$\boldsymbol{\beta} = \begin{bmatrix} \cos(\delta)\cos(\alpha) \\ \cos(\delta)\sin(\alpha) \\ \sin(\delta) \end{bmatrix}.$$

The related earth-bound direction vector $\mathbf{s}$ with coordinates $(l, m, n)$ (taking earth rotation into account) is given by

$$\mathbf{s} = \mathbf{Q}_k(L, B)\boldsymbol{\beta},$$

where $\mathbf{Q}_k(L, B)$ is a $3 \times 3$ rotation matrix that accounts for the earth rotation and depends on the time $k$ and the observer's longitude $L$ and latitude $B$. Because $\mathbf{s}$ has a unit norm, we only need two coordinates $(l, m)$, while the third coordinate can be calculated using $n = \sqrt{1 - l^2 - m^2}$.

For the $q$th source with coordinates $(l_q, m_q)$ at the $k$th snapshot, the direction vector is $\mathbf{s}_q$. Let the vector $\boldsymbol{\xi}_i = [x_i, y_i, z_i]^T$ denote the position of the $i$th receiving element in earth-bound coordinates. At this position, the phase delay (geometric delay) experienced by the $q$ source is given by the inner product of these vectors, and the effect of this delay on the signal is multiplication by a complex coefficient $a_{kqi} := \exp(j\frac{2\pi}{\lambda}\boldsymbol{\xi}_i^T\mathbf{s}_q)$, where $\lambda$ is the wavelength. Stacking the coefficients for $i = 1, \cdots, P$ into a vector $\mathbf{a}_{k,q} = \mathbf{a}_k(\mathbf{s}_q)$, we obtain the array response vector, which thus has model

$$\mathbf{a}_{k,q} = \mathbf{a}_k(\mathbf{s}_q) = \frac{1}{\sqrt{P}} e^{j\frac{2\pi}{\lambda}\boldsymbol{\Xi}^T\mathbf{s}_q} = \frac{1}{\sqrt{P}} e^{j\frac{2\pi}{\lambda}\boldsymbol{\Xi}^T\mathbf{Q}_k(L,B)\boldsymbol{\beta}_q} \quad (9)$$

where $\boldsymbol{\Xi}$ is a $3 \times P$ matrix containing the positions of the $P$ receiving elements. We introduced a scaling by $1/\sqrt{P}$ as a normalization constant such that $\|\mathbf{a}_k(\mathbf{s}_q)\| = 1$. The entries of the array response vector are connected to the Fourier transform coefficients that are familiar in radio astronomy models.

Assuming an array that is otherwise calibrated, the received antenna signals $\mathbf{y}_k[n]$ can be modeled as

$$\mathbf{y}_k[n] = \mathbf{A}_k\mathbf{x}[n] + \mathbf{n}_k[n], \qquad n = 1, \cdots, N \quad (10)$$

where $\mathbf{A}_k$ is a $P \times Q$ matrix whose columns are the array response vectors $[\mathbf{A}_k]_q = \mathbf{a}_{k,q}$, $\mathbf{x}[n]$ is a $Q \times 1$ vector representing the signals from the sky, and $\mathbf{n}_k[n]$ is a $P \times 1$ vector modeling the noise.

From the data, the system estimates covariance matrices of the input vector at each snapshot $k = 1, \cdots, K$, as

$$\hat{\mathbf{R}}_k = \frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_k[n]\mathbf{y}_k[n]^H, \qquad k = 1, \cdots, K\,. \quad (11)$$

Since the received signals and noise are Gaussian, these covariance matrix estimates form sufficient statistics for the imaging problem (Leshem & van der Veen 2000). The covariance matrices are given by

$$\mathbf{R}_k = \mathcal{E}\{\mathbf{y}_k\mathbf{y}_k^H\}, \quad (12)$$

for which the model is

$$\mathbf{R}_k = \mathbf{A}_k\boldsymbol{\Sigma}\mathbf{A}_k^H + \mathbf{R}_{\mathbf{n},k}, \quad (13)$$

where $\boldsymbol{\Sigma} = \mathcal{E}\{\mathbf{x}\mathbf{x}^H\}$ and $\mathbf{R}_{\mathbf{n},k} = \mathcal{E}\{\mathbf{n}_k\mathbf{n}_k^H\}$ are the source and noise covariance matrices, respectively. We have assumed that sky sources are stationary, and if we also assume that they are independent, we can model $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma})$ where

$$\boldsymbol{\sigma} = [\sigma_1 \quad , \ldots, \quad \sigma_Q]^T \quad (14)$$

represents the intensity of the sources. .To connect the covariance data model (13) to language more familiar to radio astronomers, let us take a closer look at the elements of the matrix $\mathbf{R}_k$. Temporarily ignoring the noise covariance matrix $\mathbf{R}_{\mathbf{n},k}$, we note that

$$\begin{aligned}[\mathbf{R}_k]_{ij} &= \frac{1}{P}\sum_{q=1}^{Q} \sigma_q a_{kqi} a_{kqi}^* \\ &= \frac{1}{P}\sum_{q=1}^{Q} \sigma_q e^{j\frac{2\pi}{\lambda}(\boldsymbol{\xi}_i - \boldsymbol{\xi}_j)^T\mathbf{s}_q} \\ &= \frac{1}{P}\sum_{q=1}^{Q} \sigma_q e^{j\frac{2\pi}{\lambda}\left[(x_i-x_j)l_q + (y_i-y_j)m_q + (z_i-z_j)\sqrt{1-l_q^2-m_q^2}\right]}\end{aligned} \quad (15)$$

If we ddefine $\frac{1}{\lambda}[x_i - x_j, y_i - y_j, z_i - z_j]^T = [u_{ij}, v_{ij}, w_{ij}]^T$, then we can write $[\mathbf{R}_k]_{ij} \equiv V(u_{ij}, v_{ij}, w_{ij})$, where $V(u, v, w)$ is the visibility function, and $(u, v, w)$ are the spatial frequencies (Leshem et al. 2000). In other words, the entries of the covariance matrix $\mathbf{R}_k$ are samples of the visibility function at a given frequency and time arranged in a matrix, and (13) represents the measurement equation in matrix form.

We can write this equation in several other ways. By vectorizing both sides of (13) and using the properties of Kronecker products (4), we obtain

$$\mathbf{r}_k = (\mathbf{A}_k^* \circ \mathbf{A}_k)\boldsymbol{\sigma} + \mathbf{r}_{\mathbf{n},k} \quad (16)$$

where $\mathbf{r}_k = \mathrm{vect}(\mathbf{R}_k)$ and $\mathbf{r}_{\mathbf{n},k} = \mathrm{vect}(\mathbf{R}_{\mathbf{n},k})$. After stacking the vectorized covariances for all of the snapshots, we obtain

$$\mathbf{r} = \boldsymbol{\Psi}\boldsymbol{\sigma} + \mathbf{r}_{\mathbf{n}} \quad (17)$$

where

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_K \end{bmatrix}, \quad \boldsymbol{\Psi} = \begin{bmatrix} \mathbf{A}_1^* \circ \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_K^* \circ \mathbf{A}_K \end{bmatrix}, \quad \mathbf{r}_{\mathbf{n}} = \begin{bmatrix} \mathbf{r}_{\mathbf{n},1} \\ \vdots \\ \mathbf{r}_{\mathbf{n},K} \end{bmatrix}. \quad (18)$$

Similarly, we vectorize and stack the sample covariance matrices as

$$\hat{\mathbf{r}}_k = \mathrm{vect}(\hat{\mathbf{R}}_k)\,, \qquad \hat{\mathbf{r}} = \begin{bmatrix} \hat{\mathbf{r}}_1 \\ \vdots \\ \hat{\mathbf{r}}_K \end{bmatrix}. \quad (19)$$

This collects all the available covariance data into a single vector.

Alternatively, we can use the independence between the time samples to write the aggregate data model as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{R}_K \end{bmatrix} = \sum_{q=1}^{Q} \sigma_q (\mathbf{I}_K \circ \mathbf{A}^q)(\mathbf{I}_K \circ \mathbf{A}^q)^H + \mathbf{R}_\mathbf{n}\,, \tag{20}$$

where

$$\mathbf{R}_\mathbf{n} = \begin{bmatrix} \mathbf{R}_{\mathbf{n},1} & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{R}_{\mathbf{n},K} \end{bmatrix}, \tag{21}$$

$$\mathbf{A}^q = [\mathbf{a}_{1,q} \quad \dots \quad \mathbf{a}_{K,q}]\,, \quad q = 1, \cdots, Q. \tag{22}$$

## 3. The imaging problem

Using the data model (17), the imaging problem is to find the intensity, $\boldsymbol{\sigma}$, of the sources, along with their directions represented by the matrices $\mathbf{A}_k$, from given sample covariance matrices $\hat{\mathbf{R}}_k$, $k = 1, \cdots, K$. As the source locations are generally unknown, this is a complicated (nonlinear) direction-of-arrival estimation problem.

The usual approach in radio astronomy is to define a grid for the image and to assume that each pixel (grid location) contains a source. In this case the source locations are known, and estimating the source intensities is a linear problem, but for high-resolution images the number of sources may be very large. The resulting linear estimation problem is often ill-conditioned unless additional constraints are posed.

### 3.1. Gridded imaging model

After defining a grid for the image and assuming that a source exists for each pixel location, let $I$ (rather than $Q$) denote the total number of sources (pixels), $\boldsymbol{\sigma}$ an $I \times 1$ vector containing the source intensities, and $\mathbf{A}_k$ ($k = 1, \cdots, K$) the $P \times I$ array response matrices for these sources. The $\mathbf{A}_k$ are known, and $\boldsymbol{\sigma}$ can be interpreted as a vectorized version of the image to be computed. (To distinguish the gridded source locations and source powers from the "true" sources, we later denote parameters and variables that depend on the $Q$ true sources by a tilde.)

For a given observation $\hat{\mathbf{r}}$, image formation amounts to the estimation of $\boldsymbol{\sigma}$. For a sufficiently fine grid, $\boldsymbol{\sigma}$ approximates the solution of the discrete source model. However, as we discuss later, working in the image domain leads to a gridding-related noise floor. This is solved by fine adaptation of the location of the sources and estimation of the true locations in the visibility domain.

A consequence of using a discrete source model in combination with a sequential source-removing technique such as CLEAN is the modeling of extended structures in the image by many point sources. As we discuss in Sec. 6, this also holds for the algorithms proposed in this paper.

### 3.2. Unconstrained least squares image

If we ignore the term $\mathbf{r}_\mathbf{n}$, then (17) directly leads to least squares (LS) and weighted least squares (WLS) estimates of $\boldsymbol{\sigma}$ (Wijnholds & van der Veen 2008). In particular, solving the imaging problem with LS leads to the minimization problem

$$\min_{\boldsymbol{\sigma}} \frac{1}{2K} \|\hat{\mathbf{r}} - \boldsymbol{\Psi}\boldsymbol{\sigma}\|^2\,, \tag{23}$$

where the normalization factor $2K$ is introduced to simplify the expression for the gradient and does not affect the solution. It is straightforward to show that the solution to this problem is given by any $\boldsymbol{\sigma}$ that satisfies

$$\mathbf{H}_{\mathrm{LS}}\boldsymbol{\sigma} = \hat{\boldsymbol{\sigma}}_{\mathrm{MF}} \tag{24}$$

where we define the "matched filter" (MF, also known as the classical "direct Fourier transform dirty image") as

$$\hat{\boldsymbol{\sigma}}_{\mathrm{MF}} = \frac{1}{K}\boldsymbol{\Psi}^H\hat{\mathbf{r}} = \frac{1}{K}\sum_k \mathrm{vectdiag}(\mathbf{A}_k^H\hat{\mathbf{R}}_k\mathbf{A}_k), \tag{25}$$

and the deconvolution matrix $\mathbf{H}_{\mathrm{LS}}$ as

$$\mathbf{H}_{\mathrm{LS}} = \frac{1}{K}\boldsymbol{\Psi}^H\boldsymbol{\Psi} = \frac{1}{K}\sum_k (\mathbf{A}_k^T\mathbf{A}_k^*) \odot (\mathbf{A}_k^H\mathbf{A}_k), \tag{26}$$

where we have used the definition of $\boldsymbol{\Psi}$ from (18) (with tilde removed) and properties of the Kronecker and Khatri-Rao products. Similarly we can define the WLS minimization as

$$\min_{\boldsymbol{\sigma}} \frac{1}{2K} \|(\hat{\mathbf{R}}^{-T/2} \otimes \hat{\mathbf{R}}^{-1/2})(\hat{\mathbf{r}} - \boldsymbol{\Psi}\boldsymbol{\sigma})\|^2\,, \tag{27}$$

where the weighting assumes Gaussian distributed observations. The weighting improves the statistical properties of the estimates, and $\hat{\mathbf{R}}$ is used instead of $\mathbf{R}$ because it is available and asymptotically gives the same optimal results, i.e., convergence to maximum likelihood estimates (Ottersten et al. 1998). The solution to this optimization is similar to the solution to the LS problem and is given by any $\boldsymbol{\sigma}$ that satisfies

$$\mathbf{H}_{\mathrm{WLS}}\boldsymbol{\sigma} = \hat{\boldsymbol{\sigma}}_{\mathrm{WLS}}, \tag{28}$$

where

$$\hat{\boldsymbol{\sigma}}_{\mathrm{WLS}} = \frac{1}{K}\boldsymbol{\Psi}^H(\hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1})\hat{\mathbf{r}} \tag{29}$$

is the "WLS dirty image" and

$$\mathbf{H}_{\mathrm{WLS}} = \frac{1}{K}\boldsymbol{\Psi}^H(\hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1})\boldsymbol{\Psi} \tag{30}$$

is the associated deconvolution operator.

A connection to beamforming is obtained as follows. The $i$th pixel of the "Matched Filter" dirty image in equation (25) can be written as

$$\hat{\sigma}_{\mathrm{MF},i} = \frac{1}{K}\sum_k \mathbf{a}_{k,i}^H\hat{\mathbf{R}}_k\mathbf{a}_{k,i},$$

and if we replace $\mathbf{a}_{k,i}/\sqrt{K}$ by a more general "beamformer" $\mathbf{w}_{k,i}$, this can be generalized to a more general dirty image

$$\sigma_{\mathbf{w},i} = \sum_k \mathbf{w}_{k,i}^H\hat{\mathbf{R}}_k\mathbf{w}_{k,i}. \tag{31}$$

Here, $\mathbf{w}_{k,i}$ is called a beamformer because we can consider that it acts on the antenna vectors $\mathbf{y}_k[n]$ as $z_{k,i}[n] = \mathbf{w}_{k,i}^H \mathbf{y}_k[n]$, where $z_{k,i}[n]$ is the output of the (direction-dependent) beamformer, and $\sigma_{\mathbf{w},i} = \sum_k \mathcal{E}\{|z_{k,i}|^2\}$ is interpreted as the total output power of the beamformer, summed over all snapshots. We encounter several such beamformers in the rest of the paper. Most of the beamformers discussed in this paper include the weighted visibility vector $(\mathbf{R}^{-T} \otimes \mathbf{R}^{-1})\mathbf{r}$. The relation between this weighting and more traditional weighting techniques, such as natural and robust weighting, is discussed in Appendix A.

### 3.3. Preconditioned weighted least squares image

If $\mathbf{\Psi}$ has full column rank, then $\mathbf{H}_{\mathrm{LS}}$ and $\mathbf{H}_{\mathrm{WLS}}$ are non-singular and a unique solution to LS and WLS exists; for example, the solution to (24) becomes

$$\boldsymbol{\sigma} = \mathbf{H}_{\mathrm{LS}}^{-1} \hat{\boldsymbol{\sigma}}_{\mathrm{MF}} \,. \tag{32}$$

Unfortunately, if the number of pixels is large, then $\mathbf{H}_{\mathrm{LS}}$ and $\mathbf{H}_{\mathrm{WLS}}$ become ill-conditioned or even singular, so that (24) and (28) have an infinite number of solutions (Wijnholds & van der Veen 2008). Generally, we need to improve the conditioning of the deconvolution matrices and to find appropriate regularizations.

One way to improve the conditioning of a matrix is to apply a preconditioner. The most widely used and simplest one is the Jacobi preconditioner (Barrett et al. 1994), which for any matrix $\mathbf{M}$, is given by $[\mathrm{diag}(\mathbf{M})]^{-1}$. Let $\mathbf{D}_{\mathrm{WLS}} = \mathrm{diag}(\mathbf{H}_{\mathrm{WLS}})$, then by applying this preconditioner to $\mathbf{H}_{\mathrm{WLS}}$ we obtain

$$[\mathbf{D}_{\mathrm{WLS}}^{-1}\mathbf{H}_{\mathrm{WLS}}]\boldsymbol{\sigma} = \mathbf{D}_{\mathrm{WLS}}^{-1}\hat{\boldsymbol{\sigma}}_{\mathrm{WLS}} \,. \tag{33}$$

We take a closer look at $\mathbf{D}_{\mathrm{WLS}}^{-1}\hat{\boldsymbol{\sigma}}_{\mathrm{WLS}}$ for the case where $K = 1$. In this case,

$$\begin{aligned}\mathbf{H}_{\mathrm{WLS}} &= (\mathbf{A}_1^* \circ \mathbf{A}_1)^H (\hat{\mathbf{R}}_1^{-T} \otimes \hat{\mathbf{R}}_1^{-1})(\mathbf{A}_1^* \circ \mathbf{A}_1) \\ &= (\mathbf{A}^T \hat{\mathbf{R}}_1^{-T} \mathbf{A}_1^*) \odot (\mathbf{A}_1^H \hat{\mathbf{R}}_1^{-1} \mathbf{A}_1)\end{aligned}$$

and

$$\mathbf{D}_{\mathrm{WLS}}^{-1} = \begin{bmatrix} \frac{1}{(\mathbf{a}_{1,1}^H \hat{\mathbf{R}}_1^{-1} \mathbf{a}_{1,1})^2} & & \\ & \ddots & \\ & & \frac{1}{(\mathbf{a}_{1,I}^H \hat{\mathbf{R}}_1^{-1} \mathbf{a}_{1,I})^2} \end{bmatrix}.$$

This means that

$$\begin{aligned}\mathbf{D}_{\mathrm{WLS}}^{-1}\hat{\boldsymbol{\sigma}}_{\mathrm{WLS}} &= \mathbf{D}_{\mathrm{WLS}}^{-1}(\hat{\mathbf{R}}_1^{-T} \otimes \hat{\mathbf{R}}_1^{-1})(\mathbf{A}_1^* \circ \mathbf{A}_1)^H \hat{\mathbf{r}}_1 \\ &= (\hat{\mathbf{R}}_1^{-T} \mathbf{A}_1^* \mathbf{D}_{\mathrm{WLS}}^{-1/2} \circ \hat{\mathbf{R}}_1^{-1} \mathbf{A}_1 \mathbf{D}_{\mathrm{WLS}}^{-1/2})^H \hat{\mathbf{r}}_1,\end{aligned}$$

which is equivalent to a dirty image that is obtained by applying a beamformer of the form

$$\mathbf{w}_i = \frac{1}{\mathbf{a}_{1,i}^H \hat{\mathbf{R}}_1^{-1} \mathbf{a}_{1,i}} \hat{\mathbf{R}}_1^{-1} \mathbf{a}_{1,i} \tag{34}$$

to both sides of $\hat{\mathbf{R}}_1$ and stacking the results, $\hat{\sigma}_i = \mathbf{w}_i^H \hat{\mathbf{R}}_1 \mathbf{w}_i$, of each pixel into a vector. This beamformer is known in array processing as the minimum variance distortionless response (MVDR) beamformer (Capon 1969), and the corresponding dirty image is called the MVDR dirty image and was introduced in the radio astronomy context in (Leshem & van der Veen 2000). This shows that the preconditioned WLS image (motivated by its connection to the maximum likelihood) is expected to exhibit the features of high-resolution beamforming associated with the MVDR. Examples of such images are shown in Sec. 6.

### 3.4. Bounds on the image

Another approach to improving the conditioning of a problem is to introduce appropriate constraints on the solution. Typically, image formation algorithms exploit external information regarding the image in order to regularize the ill-posed problem. For example, maximum entropy techniques (Frieden 1972; Gull & Daniell 1978) impose a smoothness condition on the image, while the CLEAN algorithm (Högbom 1974) exploits a point source model wherein most of the image is empty, and this has recently been connected to sparse optimization techniques (Wiaux et al. 2009).

A lower bound on the image is almost trivial: each pixel in the image represents the intensity at a certain direction, so is non-negative. This leads to a lower bound $\boldsymbol{\sigma} \geq \mathbf{0}$. Such a non-negativity constraint has been studied, for example, in (Briggs 1995), resulting in a non-negative LS (NNLS) problem

$$\begin{aligned}&\min_{\boldsymbol{\sigma}} \frac{1}{2K}\|\hat{\mathbf{r}} - \mathbf{\Psi}\boldsymbol{\sigma}\|^2 \\ &\text{subject to } \mathbf{0} \leq \boldsymbol{\sigma}\end{aligned} \,. \tag{35}$$

A second constraint follows if we also know an upper bound $\boldsymbol{\gamma}$ such that $\boldsymbol{\sigma} \leq \boldsymbol{\gamma}$, which will bound the pixel intensities from above. We propose several choices for $\boldsymbol{\gamma}$.

By closer inspection of the $i$th pixel of the MF dirty image $\hat{\boldsymbol{\sigma}}_{\mathrm{MF}}$, we note that its expected value is given by

$$\sigma_{\mathrm{MF},i} = \frac{1}{K} \sum_k \mathbf{a}_{k,i}^H \mathbf{R}_k \mathbf{a}_{k,i} \,.$$

Using

$$\mathbf{a}_i = \mathrm{vect}(\mathbf{A}^i) = \begin{bmatrix}\mathbf{a}_{1,i}^T & \dots & \mathbf{a}_{i,K}^T\end{bmatrix}^T, \tag{36}$$

and the normalization $\mathbf{a}_{k,i}^H \mathbf{a}_{k,i} = 1$, we obtain

$$\sigma_{\mathrm{MF},i} = \frac{1}{K}\mathbf{a}_i^H \mathbf{R} \mathbf{a}_i = \sigma_i + \frac{1}{K}\mathbf{a}_i^H \mathbf{R}_r \mathbf{a}_i, \tag{37}$$

where

$$\mathbf{R}_r = \sum_{j \neq i} \sigma_j (\mathbf{I}_K \circ \mathbf{A}^j)(\mathbf{I}_K \circ \mathbf{A}^j)^H + \mathbf{R}_{\mathbf{n}} \tag{38}$$

is the contribution of all other sources and the noise. We note that $\mathbf{R}_r$ is positive-(semi)definite. Thus, (37) implies $\sigma_{\mathrm{MF},i} \geq \sigma_i$ which means that the expected value of the MF dirty image forms an upper bound for the desired image, or

$$\boldsymbol{\sigma} \leq \boldsymbol{\sigma}_{\mathrm{MF}} \,. \tag{39}$$

Using the relation between the MF dirty image and beamformers as discussed in Sec. 3.2, we answer the following question: What is the tightest upper bound for $\sigma_i$ that we can construct using linear beamforming?

This question can be translated into an optimization problem and a closed form solution (Appendix B) exists:

$$\sigma_{\mathrm{opt},i} = \min_k \left( \frac{1}{\mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i}} \right). \tag{40}$$

Here $\sigma_{\mathrm{opt},i}$ is the tightest upper bound, and the beamformer that achieves this bound is called the adaptive selective sidelobe canceller (ASSC) (Levanda & Leshem 2013).

One problem with using this result in practice is that $\sigma_{\mathrm{opt},i}$ depends on a single snapshot. Actual dirty images are based on the sample covariance matrix $\hat{\mathbf{R}}$, *so* they are random variables. If we use a sample covariance matrix $\hat{\mathbf{R}}$ instead of the true covariance matrix $\mathbf{R}$ in (40), the variance of the result can be unacceptably large. An analysis of this problem and various solutions for it are discussed in (Levanda & Leshem 2013).

To reduce the variance we tolerate an increase of the bound with respect to the tightest upper bound, however, we would like our result to be tighter than the MF dirty image. It can be shown that the MVDR dirty image defined as

$$\sigma_{\mathrm{MVDR},i} = \frac{1}{\frac{1}{K}\sum_k \mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i}}\,, \tag{41}$$

satisfies $\sigma_i \le \sigma_{\mathrm{MVDR},i} \le \sigma_{\mathrm{MF},i}$ and produces a very tight bound (see Appendix B for the proof). This leads to the following constraint

$$\boldsymbol{\sigma} \le \boldsymbol{\sigma}_{\mathrm{MVDR}}\,. \tag{42}$$

Interestingly, for $K = 1$ the MVDR dirty image is the same image as we obtained earlier by applying a Jacobi preconditioner to the WLS problem.

### 3.5. Estimation of the upper bound from noisy data

The upper bounds (39) and (42) assume that we know the true covariance matrix $\mathbf{R}$. However, in practice we only measure $\hat{\mathbf{R}}$, which is subject to statistical fluctuations. Choosing a confidence level of six times the standard deviation of the dirty images ensures that the upper bound will hold with a probability of 99.9%.

This leads to an increase in the upper bound by a factor $1 + \alpha$ where $\alpha > 0$ is chosen such that

$$\boldsymbol{\sigma} \le (1+\alpha)\ \hat{\boldsymbol{\sigma}}_{\mathrm{MF}}. \tag{43}$$

Similarly, for the MVDR dirty image, the constraint based on $\hat{\mathbf{R}}$ is

$$\boldsymbol{\sigma} \le (1+\alpha)\ \hat{\boldsymbol{\sigma}}_{\mathrm{MVDR}}, \tag{44}$$

where

$$\hat{\sigma}_{\mathrm{MVDR},i} = \frac{C}{\frac{1}{K}\sum_k \mathbf{a}_{k,i}^H \hat{\mathbf{R}}_k^{-1} \mathbf{a}_{k,i}} \tag{45}$$

is an unbiased estimate of the MVDR dirty image, and

$$C = \frac{N}{N-P} \tag{46}$$

is a bias correction constant. With some algebra the unbiased estimate can be written in vector form as

$$\hat{\boldsymbol{\sigma}}_{\mathrm{MVDR}} = \mathbf{D}^{-1}\boldsymbol{\Psi}^H(\hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1})\hat{\mathbf{r}}, \tag{47}$$

where

$$\mathbf{D} = \frac{1}{KC}\mathrm{diag}^2\left(\mathbf{A}^H\hat{\mathbf{R}}^{-1}\mathbf{A}\right), \tag{48}$$

and

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix}\mathbf{A}_1^T & \dots & \mathbf{A}_K^T\end{bmatrix}^T \\ &= [\mathbf{a}_1 \quad \dots \quad \mathbf{a}_I]\,.\end{aligned} \tag{49}$$

The exact choice of $\alpha$ and $C$ are discussed in Appendix C.

### 3.6. Constrained least squares imaging

Now that we have lower and upper bounds on the image, we can use these as constraints in the LS imaging problem to provide a regularization. The resulting constrained LS (CLS) imaging problem is

$$\begin{aligned}&\min_{\boldsymbol{\sigma}} \frac{1}{2K}\|\hat{\mathbf{r}} - \boldsymbol{\Psi}\boldsymbol{\sigma}\|^2 \\ &\text{s.t. } \mathbf{0} \le \boldsymbol{\sigma} \le \boldsymbol{\gamma}\end{aligned}\,, \tag{50}$$

where $\boldsymbol{\gamma}$ can be chosen either as $\boldsymbol{\gamma} = \boldsymbol{\sigma}_{\mathrm{MF}}$ for the MF dirty image or $\boldsymbol{\gamma} = \boldsymbol{\sigma}_{\mathrm{MVDR}}$ for the MVDR dirty image (or their sample covariance based estimates given by (43) and (44)).

The improvements to the unconstrained LS problem that were discussed in Sec. 3.2 are still applicable. The extension to WLS leads to the cost function

$$f_{\mathrm{WLS}}(\boldsymbol{\sigma}) = \frac{1}{2}\|(\hat{\mathbf{R}}^{-T/2} \otimes \hat{\mathbf{R}}^{-1/2})\,(\hat{\mathbf{r}} - \boldsymbol{\Psi}\boldsymbol{\sigma})\,\|^2\,. \tag{51}$$

The constrained WLS problem is then given by

$$\begin{aligned}&\min_{\boldsymbol{\sigma}} f_{\mathrm{WLS}}(\boldsymbol{\sigma}) \\ &\text{s.t. } \mathbf{0} \le \boldsymbol{\sigma} \le \boldsymbol{\gamma}\,.\end{aligned} \tag{52}$$

We also recommend including a preconditioner that, as shown in Sec.3.3, relates the WLS to the MVDR dirty image. However, because of the inequality constraints, (52) does not have a closed form solution, and it is solved by an iterative algorithm. To have the relation between WLS and MVDR dirty image during the iterations, we introduce a change of variable of the form $\check{\boldsymbol{\sigma}} = \mathbf{D}\boldsymbol{\sigma}$, where $\check{\boldsymbol{\sigma}}$ is the new variable for the preconditioned problem and the diagonal matrix $\mathbf{D}$ is given in (48). The resulting constrained preconditioned WLS (CPWLS) optimization problem is

$$\begin{aligned}\check{\boldsymbol{\sigma}} = \quad &\arg\min_{\check{\boldsymbol{\sigma}}} \frac{1}{2}\|(\hat{\mathbf{R}}^{-T/2} \otimes \hat{\mathbf{R}}^{-1/2})\left(\hat{\mathbf{r}} - \boldsymbol{\Psi}\mathbf{D}^{-1}\check{\boldsymbol{\sigma}}\right)\|^2 \\ &\text{s.t. } \mathbf{0} \le \check{\boldsymbol{\sigma}} \le \mathbf{D}\boldsymbol{\gamma}\end{aligned}\,, \tag{53}$$

and the final image is found by setting $\boldsymbol{\sigma} = \mathbf{D}^{-1}\check{\boldsymbol{\sigma}}$. (Here we use $\mathbf{D}$ as a positive diagonal matrix so that the transformation to an upper bound for $\check{\boldsymbol{\sigma}}$ is correct.) Interestingly, the dirty image that follows from the (unconstrained) WLS part of the problem is given by the MVDR image $\hat{\boldsymbol{\sigma}}_{\mathrm{MVDR}}$ in (47).

## 4. Constrained optimization using an active set method

The constrained imaging formulated in the previous section requires the numerical solution of the optimization problems (50) or (53). The problem is classified as a positive definite quadratic program with simple bounds, this is a special case of a convex optimization problem with linear inequality constraints, and we can follow standard approaches to find a solution (Gill et al. 1981; Boyd & Vandenberghe 2004).

For an unconstrained optimization problem, the gradient of the cost function calculated at the solution must vanish. If we are not yet at the optimum in an iterative process, the gradient is used to update the current solution. For constrained optimization, the constraints are usually added

to the cost function using (unknown) Lagrange multipliers that need to be estimated along with the solution. At the solution, part of the gradient of the cost function is not zero but related to the nonzero Lagrange multipliers. For inequality constraints, the sign of the Lagrange multipliers plays an important role.

As we show here, these characteristics of the solution (based on the gradient and the Lagrange multipliers) can be used to develop an algorithm called the active set method, which is closely related to the sequential source removing techniques such as CLEAN.

In this section, we use the active set method to solve the constrained optimization problem.

### 4.1. Characterization of the optimum

Let $\bar{\boldsymbol{\sigma}}$ be the solution to the optimization problem (50) or (53). An image is called feasible if it satisfies the bounds $\boldsymbol{\sigma} \geq \mathbf{0}$ and $-\boldsymbol{\sigma} \geq -\boldsymbol{\gamma}$. At the optimum, some pixels may satisfy a bound with equality, and these are called the "active" pixels.

We use the following notation. For any feasible image $\boldsymbol{\sigma}$, let

$$\mathcal{L}(\boldsymbol{\sigma}) = \{i \,|\, \sigma_i = 0\} \tag{54}$$
$$\mathcal{U}(\boldsymbol{\sigma}) = \{i \,|\, \sigma_i = \gamma_i\} \tag{55}$$
$$\mathcal{A}(\boldsymbol{\sigma}) = \mathcal{L}(\boldsymbol{\sigma}) \cup \mathcal{U}(\boldsymbol{\sigma}) \tag{56}$$
$$\mathcal{F}(\boldsymbol{\sigma}) = \mathcal{I} \setminus \mathcal{A}(\boldsymbol{\sigma})\,. \tag{57}$$

Here, $\mathcal{I} = \{1, \cdots, I\}$ is the set of all pixel indices; $\mathcal{L}(\boldsymbol{\sigma})$ is the set where the lower bound is active, i.e., the pixel value is 0; $\mathcal{U}(\boldsymbol{\sigma})$ is the set of pixels that attain the upper bound; $\mathcal{A}(\boldsymbol{\sigma})$ is the set of all pixels where one of the constraints is active. These are the active pixels. The free set $\mathcal{F}(\boldsymbol{\sigma})$ is the set of pixels $i$, which have values strictly between 0 and $\gamma_i$. Furthermore, for any vector $\mathbf{v} = [v_i]$, let $\mathbf{v}_{\mathcal{F}}$ correspond to the subvector with indices $i \in \mathcal{F}$, and similarly define $\mathbf{v}_{\mathcal{L}}$ and $\mathbf{v}_{\mathcal{U}}$. We write $\mathbf{v} = \mathbf{v}_{\mathcal{F}} \oplus \mathbf{v}_{\mathcal{L}} \oplus \mathbf{v}_{\mathcal{U}}$.

Let $\bar{\boldsymbol{\sigma}}$ be the optimum, and let $\bar{\mathbf{g}} = \mathbf{g}(\bar{\boldsymbol{\sigma}})$ be the gradient of the cost function at this point. Define the free sets and active sets $\mathcal{F}, \mathcal{L}, \mathcal{U}$ at $\bar{\boldsymbol{\sigma}}$. We can write $\bar{\mathbf{g}} = \bar{\mathbf{g}}_{\mathcal{F}} \oplus \bar{\mathbf{g}}_{\mathcal{L}} \oplus \bar{\mathbf{g}}_{\mathcal{U}}$. Associated with the active pixels of $\bar{\boldsymbol{\sigma}}$ is a vector $\bar{\boldsymbol{\lambda}} = \bar{\boldsymbol{\lambda}}_{\mathcal{L}} \oplus \bar{\boldsymbol{\lambda}}_{\mathcal{U}}$ of Lagrange multipliers. Optimization theory (Gill et al. 1981) tells us that the optimum $\bar{\boldsymbol{\sigma}}$ is characterized by the following conditions:

$$\mathbf{g}_{\mathcal{F}}(\bar{\boldsymbol{\sigma}}) = \mathbf{0} \tag{58}$$
$$\bar{\boldsymbol{\lambda}}_{\mathcal{L}} = \bar{\mathbf{g}}_{\mathcal{L}} \geq \mathbf{0} \tag{59}$$
$$\bar{\boldsymbol{\lambda}}_{\mathcal{U}} = -\bar{\mathbf{g}}_{\mathcal{U}} \geq \mathbf{0}\,. \tag{60}$$

Thus, the part of the gradient corresponding to the free set is zero, but the part of the gradient corresponding to the active pixels is not necessarily zero. Since we have simple bounds, this part becomes equal to the Lagrange multipliers $\bar{\boldsymbol{\lambda}}_{\mathcal{L}}$ and $-\bar{\boldsymbol{\lambda}}_{\mathcal{U}}$ (the negative sign is caused by the condition $-\boldsymbol{\sigma}_{\mathcal{U}} \geq -\boldsymbol{\gamma}_{\mathcal{U}}$). The condition $\boldsymbol{\lambda} \geq \mathbf{0}$ is crucial: a negative Lagrange multiplier would indicate that there is a feasible direction of descent $\mathbf{p}$ for which a small step in that direction, $\bar{\boldsymbol{\sigma}} + \mu\mathbf{p}$, has a lower cost and still satisfies the constraints, thus contradicting optimality of $\bar{\boldsymbol{\sigma}}$ (Gill et al. 1981).

"Active set" algorithms consider that if the true active set at the solution is known, the optimization problem with inequality constraints reduces to an optimization with equality constraints,

$$\mathbf{z} = \arg\min_{\boldsymbol{\sigma}} f(\boldsymbol{\sigma}) \tag{61}$$
$$\text{s.t. } \boldsymbol{\sigma}_{\mathcal{L}} = \mathbf{0}\,,\ \boldsymbol{\sigma}_{\mathcal{U}} = \boldsymbol{\gamma}_{\mathcal{U}}\,.$$

Since we can substitute the values of the active pixels into $\boldsymbol{\sigma}$, the problem becomes a standard unconstrained LS problem with a reduced dimension: only $\bar{\boldsymbol{\sigma}}_{\mathcal{F}}$ needs to be estimated. Specifically, for CLS the unconstrained subproblem is formulated as

$$f(\boldsymbol{\sigma}) = \frac{1}{2K}\|\mathbf{b}_{\text{LS}} - \boldsymbol{\Psi}_{\mathcal{F}}\boldsymbol{\sigma}_{\mathcal{F}}\|^2 \tag{62}$$

where

$$\mathbf{b}_{\text{LS}} = \hat{\mathbf{r}} - \boldsymbol{\Psi}_{\mathcal{U}}\boldsymbol{\sigma}_{\mathcal{U}}. \tag{63}$$

Similarly, for CPWLS we have

$$f(\check{\boldsymbol{\sigma}}) = \frac{1}{2}\left\|\mathbf{b}_{\text{PWLS}} - \left(\hat{\mathbf{R}}^{-T/2} \otimes \hat{\mathbf{R}}^{-1/2}\right)(\boldsymbol{\Psi}\mathbf{D}^{-1})_{\mathcal{F}}\check{\boldsymbol{\sigma}}_{\mathcal{F}}\right\|^2, \tag{64}$$

where

$$\mathbf{b}_{\text{PWLS}} = \left(\hat{\mathbf{R}}^{-T/2} \otimes \hat{\mathbf{R}}^{-1/2}\right)(\hat{\mathbf{r}} - (\boldsymbol{\Psi}\mathbf{D}^{-1})_{\mathcal{U}}\check{\boldsymbol{\sigma}}_{\mathcal{U}}). \tag{65}$$

In both cases, closed form solutions can be found, and we discuss a suitable Krylov-based algorithm for this in Sec. 5.

As a result, the essence of the constrained optimization problem is to find $\mathcal{L}, \mathcal{U}$, and $\mathcal{F}$. In the literature, algorithms for this are called "active set methods", and we propose a suitable algorithm in Sec. 4.3.

### 4.2. Gradients

We first derive expressions for the gradients required for each of the unconstrained subproblems (62) and (64). Generically, a WLS cost function (as function of a real-valued parameter vector $\boldsymbol{\theta}$) has the form

$$f(\boldsymbol{\theta})_{\text{WLS}} = \beta\|\mathbf{G}^{1/2}\mathbf{c}(\boldsymbol{\theta})\|^2 = \beta\mathbf{c}(\boldsymbol{\theta})^H\mathbf{G}\mathbf{c}(\boldsymbol{\theta}) \tag{66}$$

where $\mathbf{G}$ is a Hermitian weighting matrix and $\beta$ is a scalar. The gradient of this function is

$$\mathbf{g}(\boldsymbol{\theta}) = 2\beta\left(\frac{\partial\mathbf{c}}{\partial\boldsymbol{\theta}^T}\right)^H\mathbf{G}\mathbf{c}\,. \tag{67}$$

For LS we have $\boldsymbol{\theta} = \boldsymbol{\sigma}$, $\mathbf{c} = \hat{\mathbf{r}} - \boldsymbol{\Psi}\boldsymbol{\sigma}$, $\beta = \frac{1}{2K}$, and $\mathbf{G} = \mathbf{I}$. This leads to

$$\begin{aligned}\mathbf{g}_{\text{LS}}(\boldsymbol{\sigma}) &= -\frac{1}{K}\boldsymbol{\Psi}^H(\hat{\mathbf{r}} - \boldsymbol{\Psi}\boldsymbol{\sigma}) \\ &= \mathbf{H}_{\text{LS}}\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}_{\text{MF}}.\end{aligned} \tag{68}$$

For PWLS, $\boldsymbol{\theta} = \check{\boldsymbol{\sigma}}$, $\mathbf{c} = \hat{\mathbf{r}} - \boldsymbol{\Psi}\mathbf{D}^{-1}\check{\boldsymbol{\sigma}}$, $\beta = \frac{1}{2}$, and $\mathbf{G} = \hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1}$. Substituting these into (67), we obtain

$$\begin{aligned}\mathbf{g}_{\text{PWLS}}(\check{\boldsymbol{\sigma}}) &= -\mathbf{D}^{-1}\boldsymbol{\Psi}^H(\hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1})(\hat{\mathbf{r}} - \boldsymbol{\Psi}\mathbf{D}^{-1}\check{\boldsymbol{\sigma}}) \\ &= \mathbf{H}_{\text{PWLS}}\check{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}_{\text{MVDR}}\end{aligned} \tag{69}$$

where

$$\mathbf{H}_{\text{PWLS}} = \mathbf{D}^{-1}\boldsymbol{\Psi}^H(\hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1})\boldsymbol{\Psi}\mathbf{D}^{-1}, \tag{70}$$

and we used (47).

An interesting observation is that the gradients can be interpreted as residual images obtained by subtracting the dirty image from a convolved model image. At a later point, this will allow us to relate the active set method to sequential source removing techniques.

### 4.3. Active set methods

In this section, we describe the steps needed to find the sets $\mathcal{L}$, $\mathcal{U}$ and, $\mathcal{F}$, and the solution. We follow the template algorithm proposed in (Gill et al. 1981). The algorithm is an iterative technique where we gradually improve on an image. Let the image at iteration $j$ be denoted by $\boldsymbol{\sigma}^{(j)}$ where $j = 1, 2, \cdots$, and we always ensure this is a feasible solution (satisfies $\mathbf{0} \leq \boldsymbol{\sigma}^{(j)} \leq \boldsymbol{\gamma}$). The corresponding gradient is the vector $\mathbf{g} = \mathbf{g}(\boldsymbol{\sigma}^{(j)})$, and the current estimate of the Lagrange multipliers $\boldsymbol{\lambda}$ is obtained from $\mathbf{g}$ using (59) and (60). The sets $\mathcal{L}$, $\mathcal{U}$, and $\mathcal{F}$ are current estimates that are not yet necessarily equal to the true sets.

If this image is not yet the true solution, it means that one of the conditions in (58)–(60) is violated. If the gradient corresponding to the free set is not yet zero ($\mathbf{g}_\mathcal{F} \neq \mathbf{0}$), then this is remedied by recomputing the image from the essentially unconstrained subproblem (61). It may also happen that some entries of $\boldsymbol{\lambda}$ are negative. This implies that we do not yet have the correct sets $\mathcal{L}, \mathcal{U}$, and $\mathcal{F}$. Suppose $\lambda_i < 0$. The connection of $\lambda_i$ to the gradient indicates that the cost function can be reduced in that dimension without violating any constraints (Gill et al. 1981), at the same time making the pixel no longer active. Thus we remove the $i$th pixel from the active set, add it to the free set, and recompute the image with the new equality constraints using (61). As discussed later, a threshold $\epsilon$ is needed in the test for the negativity of $\lambda_i$, therefore this step is called the "detection problem".

Table 1 summarizes the resulting active set algorithm and describes how the solution $\mathbf{z}$ to the subproblem is used at each iteration. Some efficiency is obtained by not computing the complete gradient $\mathbf{g}$ at every iteration, but only the parts corresponding to $\mathcal{L}$,$\mathcal{U}$, when they are needed. For the part corresponding to $\mathcal{F}$, we use a flag that indicates whether $\mathbf{g}_\mathcal{F}$ is zero or not.

The iterative process is initialized in line 1. This can be done in many ways. As long as the initial image lies within the feasible region ($\mathbf{0} \leq \boldsymbol{\sigma}^{(0)} \leq \boldsymbol{\gamma}$), the algorithm will converge to a constrained solution. We can simply initialize by $\boldsymbol{\sigma}^{(0)} = \mathbf{0}$.

Line 3 is a test for convergence, corresponding to the conditions (58)–(60). The loop is followed while any of the constraints is violated.

If $\mathbf{g}_\mathcal{F}$ is not zero, then the unconstrained subproblem (61) is solved in line 5. If this solution $\mathbf{z}$ satisfies the feasibility constraints, then it is kept, the image is updated accordingly, and the gradient is estimated at the new solution (only $\lambda_{\min} = \min(\boldsymbol{\lambda})$ is needed, along with the corresponding pixel index).

If $\mathbf{z}$ is not feasible, then in lines 12-16 we try to move in the direction of $\mathbf{z}$ as far as possible. The direction of descent is $\mathbf{p} = \mathbf{z} - \boldsymbol{\sigma}_\mathcal{F}^{(j)}$, and the update will be $\boldsymbol{\sigma}_\mathcal{F}^{(j+1)} = \boldsymbol{\sigma}_\mathcal{F}^{(j)} + \mu\mathbf{p}$, where $\mu$ is a non-negative step size. The $i$th pixel will hit a bound if either $\sigma_i^{(j)} + \mu p_i = 0$ or $\sigma_i^{(j)} + \mu p_i = \gamma_i$; i.e., if

$$\mu_i = \max\left(-\frac{\sigma_i^{(j)}}{p_i}, \frac{\gamma_i - \sigma_i^{(j)}}{p_i}\right) \tag{71}$$

($\mu_i$ is non-negative). Then the maximal feasible step size towards a constraint is given by $\mu_{\max} = \min(\mu_i)$ for $i \in \mathcal{F}$. The corresponding pixel index is removed from $\mathcal{F}$ and added to $\mathcal{L}$ or $\mathcal{U}$.

Table 1: Constrained LS imaging using active sets

1: Initialize: set the initial image $\boldsymbol{\sigma}^{(0)} = \mathbf{0}$, $j = 0$, set the free set $\mathcal{F} = \emptyset$, and $\mathcal{L}$,$\mathcal{U}$ accordingly
2: Set the flag *Freegradient-isnotzero* := True
3: **while** *Freegradient-isnotzero* **or** $\lambda_{\min} < 0$ **do**
4:  **if** *Freegradient-isnotzero* **then**
5:  Let $\mathbf{z}$ be the solution of the unconstrained subproblem (61)
6:  **if** $\mathbf{z}$ is feasible **then**
7:  Update the image: $\boldsymbol{\sigma}_\mathcal{F}^{(j+1)} = \mathbf{z}$
8:  Set *Freegradient-isnotzero* := False
9:  Compute the "active" part of the gradient and estimate the Lagrange multipliers
10:  Let $\lambda_{\min}$ be the smallest Lagrange multiplier and $i_{\min}$ the corresponding pixel index
11:  **else**
12:  Compute the direction of descent $\mathbf{p} = \mathbf{z} - \boldsymbol{\sigma}_\mathcal{F}^{(j)}$
13:  Compute the maximum feasible nonnegative step-size $\mu_{\max}$ and let $i$ be the corresponding pixel index that will attain a bound
14:  Update the image: $\boldsymbol{\sigma}_\mathcal{F}^{(j+1)} = \boldsymbol{\sigma}_\mathcal{F}^{(j)} + \mu_{\max}\mathbf{p}$
15:  Add a constraint: move $i$ from the free set $\mathcal{F}$ to $\mathcal{L}$ or $\mathcal{U}$
16:  Set *Freegradient-isnotzero* := True
17:  **end if**
18:  Increase the image index: $j := j + 1$
19:  **else**
20:  Delete a constraint: move $i_{\min}$ from $\mathcal{L}$ or $\mathcal{U}$ to the free set $\mathcal{F}$
21:  Set *Freegradient-isnotzero* := True
22:  **end if**
23: **end while**

If in line 3 the gradient satisfied $\mathbf{g}_\mathcal{F} = \mathbf{0}$ but a Lagrange multiplier is negative, we delete the corresponding constraint and add this pixel index to the free set (line 20). After this, the loop is entered again with the new constraint sets.

If we initialize the algorithm with $\boldsymbol{\sigma}^{(0)} = \mathbf{0}$, then all pixel indices will be in the set $\mathcal{L}$, and the free set is empty. During the first iteration, $\boldsymbol{\sigma}_\mathcal{F}$ remains empty but the gradient is computed (line 9). Equations (68) and (69) show that it will be equal to the negated dirty image. Thus the minimum of the Lagrange multipliers $\lambda_{\min}$ will be the current strongest source in the dirty image, and it will be added to the free set when the loop is entered again. This shows that the method as described above will lead to a sequential source removal technique similar to CLEAN. In particular, the PWLS cost function (69) relates to LS-MVI (Ben-David & Leshem 2008), which applies CLEAN-like steps to the MVDR dirty image.

In line 3, we try to detect whether a pixel should be added to the free set ($\lambda_{\min} < 0$). We note that $\boldsymbol{\lambda}$ follows from the gradient, (68) or (69), which is a random variable. We should avoid the occurrence of a "false alarm", because it will lead to overfitting the noise. Therefore, the test should be replaced by $\lambda_{\min} < -\epsilon$, where $\epsilon > 0$ is a suitable detection threshold. Because the gradients are estimated using dirty images, they share the same statistics (the variance of the other component in (68) and (69) is much smaller). To reach a desired false alarm rate, we propose to choose $\epsilon$ proportional to the standard deviation of the $i$th pixel on

the corresponding dirty image for the given cost function. (How to estimate the standard deviation of the dirty images and the threshold is discussed in Appendix C.) Choosing $\epsilon$ to be *six* times the standard deviation ensures a false alarm of $< 0.1\%$ over the complete image.

The use of this statistic improves the detection and thus the estimates greatly, however the correct detection also depends on the quality of the estimates in the previous iterations. If a strong source is off-grid, the source is usually underestimated, and this leads to a biased estimation of the gradient and the Lagrange multipliers, which in turn leads to including pixels that are not real sources. In the next section we describe one possible solution for this case.

### 4.4. Strong off-grid sources

In this section, we use a tilde to indicate "true" source parameters (as distinguished from the gridded source model); for example, $\tilde{\boldsymbol{\sigma}}$ indicates the vector with the true source intensities, and $\tilde{\boldsymbol{\Sigma}}$ the corresponding diagonal matrix, $\tilde{\mathbf{a}}_{k,q}$ indicates their array response vectors and $\tilde{\mathbf{A}}_k$ the corresponding matrix. The versions without tilde refers to the $I$ gridded sources.

The mismatch between $\boldsymbol{\Psi}$ and the unknown $\tilde{\boldsymbol{\Psi}}$ results in underestimating source intensities, which means that the remaining contribution of that source produces bias and possible artifacts in the image. To achieve high dynamic ranges, we suggest finding a grid correction for the pixels in the free set $\mathcal{F}$.

Let $\mathbf{a}_{k,i}$ have the same model as $\tilde{\mathbf{a}}_{k,q}$ with $\boldsymbol{\beta}_i$ pointing toward the center of the $i$th pixel. When a source is within a pixel but not exactly in the center, we can model this mismatch as

$$\begin{aligned}\tilde{\mathbf{a}}_{k,q} &= \frac{1}{\sqrt{P}} e^{\frac{j2\pi}{\lambda}\boldsymbol{\Xi}^T\mathbf{Q}_k(\boldsymbol{\beta}_i+\boldsymbol{\delta}_i)} \\ &= \mathbf{a}_{k,i} \odot e^{\frac{j2\pi}{\lambda}\boldsymbol{\Xi}^T\mathbf{Q}_k\boldsymbol{\delta}_i}\end{aligned}$$

where $\boldsymbol{\delta}_i = \tilde{\boldsymbol{\beta}}_q - \boldsymbol{\beta}_i$ and $i \in \mathcal{F}$. Because both $\boldsymbol{\beta}_i$ and $\tilde{\boldsymbol{\beta}}_q$ are $3 \times 1$ unit vectors, each only has two degrees of freedom. This means that we can parameterize the unknowns for the grid-correcting problem using coefficients $\delta_{1,i}$ and $\delta_{i,2}$. We assume that when a source is added to the free set, its actual position is very close to the center of the pixel on which it was detected. This means that $\delta_{1,i}$ and $\delta_{i,2}$ are within the pixel's width, denoted by $W$, and height, denoted by $H$. In this case we can replace (61) by a nonlinear constrained optimization,

$$\begin{aligned}&\min_{\boldsymbol{\delta},\boldsymbol{\sigma}} \frac{1}{2}\|\mathbf{b} - \boldsymbol{\Psi}(\boldsymbol{\delta})_{\mathcal{F}}\boldsymbol{\sigma}_{\mathcal{F}}\|_2^2 \\ &\text{s.t.} -W/2 < \delta_{1,i} < W/2 \\ &\quad -H/2 < \delta_{i,2} < H/2 \end{aligned} \tag{72}$$

where $\boldsymbol{\Psi}(\boldsymbol{\delta})_{\mathcal{F}}$ contains only the columns corresponding to the set $\mathcal{F}$, $\boldsymbol{\delta}_j$ is a vector obtained by stacking $\delta_{i,j}$ for $j = 1, 2$, and

$$\mathbf{b} = \hat{\mathbf{r}} - \boldsymbol{\Psi}_{\mathcal{U}}\boldsymbol{\sigma}_{\mathcal{U}}. \tag{73}$$

This problem can also be seen as a direction of arrival (DOA) estimation that is an active research area and beyond the scope of this paper. A good review of DOA mismatch correction for MVDR beamformers can be found in (Chen & Vaidyanathan 2007), and (Gu & Leshem 2012) proposed a correction method that is specifically applicable to the radio astronomical context.

Besides solving (72) instead of (61) in line 5 of the active set method, we also need to update the upper bounds and the standard deviations of the dirty images at the new pixel positions that are used in the other steps (e.g., lines 3, 6, and 13); the rest of the steps remain the same. Because we have a good initial guess to where each source in the free set is, we propose a Newton-based algorithm to do the correction.

### 4.5. Boxed imaging

A common practice in image deconvolution techniques like CLEAN is to use a priori knowledge and to narrow the search area for the sources to a certain region of the image, called CLEAN boxes. Because the contribution of the sources (if any) outside these boxes is assumed to be known, we can subtract them from the data such that we can assume that the intensity outside the boxes is zero.

To include these boxes in the optimization process of the active set algorithm, it is sufficient to make sure that the value of the pixels not belonging to these boxes do not change and remain zero. This is equivalent to replacing $\boldsymbol{\Psi}$ with $\boldsymbol{\Psi}_{\mathcal{B}}$, where $\mathcal{B}$ is the set of indices belonging to the boxes, before we start the optimization process. However, as we explain in the next section, we avoid storing the matrix $\boldsymbol{\Psi}$ in memory by exploiting its Khatri-Rao structure. We address this implementation issue by replacing (57) with

$$\mathcal{F}(\boldsymbol{\sigma}) = (\mathcal{I} \setminus \mathcal{A}(\boldsymbol{\sigma})) \cap \mathcal{B}, \tag{74}$$

which makes certain that the values of the elements outside of the boxes do not change. This has the same effect as removing the columns not belonging to $\mathcal{B}$ from $\boldsymbol{\Psi}$. Of course we have to make sure that these values are initialized to zero. By choosing $\boldsymbol{\sigma}^{(0)} = \mathbf{0}$, this is automatically the case. The only problem with this approach is that the values outside the box remain in the set $\mathcal{L}$ that is used for estimating the Lagrange variables, resulting in expensive calculations that are not needed. This problem is easily solved by calculating the gradient only for the pixels belonging to $\mathcal{B}$. The a priori non-zero values of the pixels (that were not in the boxes and were removed from the data) are added to the solution when the optimization process is finished.

## 5. Implementation using Krylov subspace-based methods

From the active-set methods described in the previous section, we know that we need to solve (62) or (64) at each iteration. In this section we describe how to achieve this efficiently without the need to store the whole convolution matrix in memory.

During the active-set updates, we need to solve linear equations of the form $\mathbf{Mx} = \mathbf{b}$. However, there are cases where we do not have direct access to the elements of the matrix $\mathbf{M}$. This can happen, for example, when $\mathbf{M}$ is too large to fit in memory. There are also cases where $\mathbf{M}$ (or $\mathbf{M}^H$) are implemented as subroutines that produce the result of the matrix vector multiplication $\mathbf{Mv}$ for some input vector $\mathbf{v}$. For example, for $\mathbf{M} = \boldsymbol{\Psi}$ the operation $\boldsymbol{\Psi}^H\mathbf{v}$ generates a dirty image. An equivalent (and maybe optimized)

implementation of such imaging subroutine might already be available to the user. In these scenarios it is necessary or beneficial to be able to solve the linear systems, using only the available matrix vector multiplication or the equivalent operator. A class of iterative solvers that can solve a linear system by only having access to the result of the multiplications with the matrix $\mathbf{M}$ are the Krylov subspace-based methods.

To illustrate the idea behind Krylov subspace-based methods, we assume that $\mathbf{M}$ is a square and non-singular matrix. In this case there exists a unique solution for $\mathbf{x}$ that is given by $\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$. Using the minimum polynomial of a matrix we can write

$$\mathbf{M}^{-1} = \frac{1}{\gamma_0}\sum_{j=0}^{m-1}\gamma_{j+1}\mathbf{M}^j,$$

where for a diagonalizable matrix $\mathbf{M}$, $m$ is the number of distinct eigenvalues (Ipsen & Meyer 1998). Using this polynomial expansion we have for our solution

$$\begin{aligned}\mathbf{x} &= \frac{1}{\gamma_0}\sum_{j=0}^{m-1}\gamma_{j+1}\mathbf{M}^j\mathbf{b} \\ &= \begin{bmatrix}\mathbf{b} & \mathbf{M}\mathbf{b} & \dots & \mathbf{M}^{m-1}\mathbf{b}\end{bmatrix}\boldsymbol{\gamma}\end{aligned}$$

where

$$\boldsymbol{\gamma} = \frac{1}{\gamma_0}[\gamma_1, \dots, \gamma_m]^T,$$

and $\mathcal{K}_m(\mathbf{M}, \mathbf{b}) = [\mathbf{b}, \mathbf{M}\mathbf{b}, \dots, \mathbf{M}^{m-1}\mathbf{b}]$ is called the Krylov subspace of $\mathbf{M}$ and $\mathbf{b}$. Krylov subspace-based methods compute $\mathcal{K}_n(\mathbf{M}, \mathbf{b})$ iteratively, for $n = 1, 2, ..$ and find an approximate for $\mathbf{x}$ by means of a projection on this subspace. Updating the subspace only involves a matrix-vector multiplication of the form $\mathbf{M}\mathbf{v}$.

In cases where $\mathbf{M}$ is singular or where it is not a square matrix, another class of Krylov-based algorithms can be used that is related to bidiagonalization of the matrix $\mathbf{M}$. The rest of this section describes the idea behind the Krylov-based technique LSQR and the way this helps more efficient implementation of a linear solver for our imaging algorithm.

### 5.1. Lanczos algorithm and LSQR

When we are solving CLS or PWLS, we need to solve a problem of the form $\|\mathbf{b} - \mathbf{M}\mathbf{x}\|_2^2$ as the first step in the active-set iterations; for example, in (62) $\mathbf{M} = \boldsymbol{\Psi}_{\mathcal{F}}$. It does not have to be a square matrix, and usually it is ill-conditioned, especially if the number of pixels is large. In general we can find a solution for this problem by first computing the singular value decomposition (SVD) of $\mathbf{M}$ as

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^H, \tag{75}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and $\mathbf{S}$ is a diagonal matrix with positive singular values. Then the solution $\mathbf{x}$ to $\min \|\mathbf{b} - \mathbf{M}\mathbf{x}\|^2$ is found by solving for $\mathbf{y}$ in

$$\mathbf{S}\mathbf{y} = \mathbf{U}^H\mathbf{b}, \tag{76}$$

followed by setting

$$\mathbf{x} = \mathbf{V}\mathbf{y}. \tag{77}$$

Solving the LS problem with this method is expensive in both number of operations and memory usage, especially if the matrices $\mathbf{U}$ and $\mathbf{V}$ are not needed after finding the solution. As we see below, looking at another matrix decomposition helps us to reduce these costs. For the rest of this section we use the notation given by (Paige & Saunders 1982).

The first step in this approach for solving LS problem is to reduce $\mathbf{M}$ to a lower bidiagonal form as follows

$$\mathbf{M} = \mathbf{U}\mathbf{B}\mathbf{V}^H, \tag{78}$$

where $\mathbf{B}$ is a bidiagonal matrix of the form

$$\mathbf{B} = \left[\begin{array}{cccc|c} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \ddots & \ddots & & \\ & & \beta_r & \alpha_r & \\ \hline & & & & \mathbf{0} \end{array}\right], \tag{79}$$

where $r = \text{rank}(\mathbf{M}) = \text{rank}(\mathbf{B})$ and $\mathbf{U}$,$\mathbf{V}$ are unitary matrices (different than in (75)). This representation is not unique, and without loss of generality we could choose $\mathbf{U}$ to satisfy

$$\mathbf{U}^H\mathbf{b} = \beta_1\mathbf{e}_1 \tag{80}$$

where $\beta_1 = \|\mathbf{b}\|_2$ and $\mathbf{e}_1$ is a unit norm vector with its first element equal to one.

Using $\mathbf{B}$, forward substitution gives the LS solution efficiently by solving $\mathbf{y}$ in

$$\mathbf{B}\mathbf{y} = \mathbf{U}^H\mathbf{b} = \beta_1\mathbf{e}_1 \tag{81}$$

followed by

$$\mathbf{x} = \mathbf{V}\mathbf{y}.$$

Using forward substitution we have

$$y_1 = \frac{\beta_1}{\alpha_1} \tag{82}$$

$$\mathbf{x}_1 = \mathbf{v}_1 y_1, \tag{83}$$

followed by the recursion,

$$y_{n+1} = -\frac{\beta_{n+1}}{\alpha_{n+1}}y_n \tag{84}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1}y_{n+1} \tag{85}$$

for $n = 1, \dots, M$ where $M < r$ is the iteration at which $\|\mathbf{M}^H(\mathbf{M}\mathbf{x}_n - \mathbf{b})\|^2$ vanishes within the desired precision. We can combine the bidiagonalization and solving for $\mathbf{x}$ and avoid extra storage needed for saving $\mathbf{B}$, $\mathbf{U}$, and $\mathbf{V}$. One such algorithm is based on a Krylov subspace method called the Lanczos algorithm (Golub & Kahan 1965). We first initialize with

$$\beta_1 = \|\mathbf{b}\|_2 \tag{86}$$

$$\mathbf{u}_1 = \frac{\mathbf{b}}{\beta_1} \tag{87}$$

$$\alpha_1 = \|\mathbf{M}^H\mathbf{u}_1\|_2 \tag{88}$$

$$\mathbf{v}_1 = \frac{\mathbf{M}^H\mathbf{u}_1}{\alpha_1}. \tag{89}$$

The iterations are then given by

$$\begin{aligned}\beta_{n+1} &= \|\mathbf{M}\mathbf{v}_n - \alpha_n \mathbf{u}_n\|_2 \\ \mathbf{u}_{n+1} &= \tfrac{1}{\beta_{n+1}}(\mathbf{M}\mathbf{v}_n - \alpha_n \mathbf{u}_n) \\ \alpha_{n+1} &= \|\mathbf{M}^H \mathbf{u}_{n+1} - \beta_{n+1}\mathbf{v}_n\|_2 \\ \mathbf{v}_{n+1} &= \tfrac{1}{\alpha_{n+1}}(\mathbf{M}^H \mathbf{u}_{n+1} - \beta_{n+1}\mathbf{v}_n)\end{aligned} \tag{90}$$

for $n = 1, 2, \ldots, M$, where $\mathbf{u}_n^H \mathbf{u}_n = \mathbf{v}_n^H \mathbf{v}_n = 1$. This provides us with all the parameters needed to solve the problem.

However, because of finite precision errors, the columns of $\mathbf{U}$ and $\mathbf{V}$ found in this way lose their orthogonality as we proceed. To prevent this error propagation into the final solution $\mathbf{x}$, different algorithms, such as conjugate gradient (CG), MINRES, and LSQR, have been proposed. The exact updates for $\mathbf{x}_n$ and stopping criteria to find $M$ depend on the choice of algorithm used and therefore are not included in the iterations above.

An overview of Krylov subspace-based methods is given by (Choi 2006, pp.91). This study shows that LSQR is a good candidate for solving LS problems when we are dealing with an ill-conditioned and non-square matrix. For this reason we use LSQR to solve Eqs. (62) or (64). Because the remaining steps during the LSQR updates are a few scalar operations and do not have a large impact on the computational complexity of the algorithm, we do not go into the details(see (Paige & Saunders 1982)).

In the next section we discuss how to use the structure in $\mathbf{M}$ to avoid storing the entire matrix in memory and how to parallelize the computations.

### 5.2. Implementation

During the active set iteration we need to solve Eqs. (62) and (64) where the matrix $\mathbf{M}$ in LSQR is replaced by $\mathbf{\Psi}_{\mathcal{F}}$ and $(\mathbf{R}^{-T/2} \otimes \mathbf{R}^{-1/2})(\mathbf{\Psi}\mathbf{D}^{-1})_{\mathcal{F}}$, respectively. Because $\mathbf{\Psi}$ has a Khatri-Rao structure and selecting and scaling a subset of columns does not change this, $\mathbf{\Psi}_{\mathcal{F}}$ and $(\mathbf{\Psi}\mathbf{D}^{-1})_{\mathcal{F}}$ also have a Khatri-Rao structure. Here we show how to use this structure to implement (90) in parallel and with less memory usage.

The only time the matrix $\mathbf{M}$ enters the algorithm is via the matrix-vector multiplications $\mathbf{M}\mathbf{v}_n$ and $\mathbf{M}^H\mathbf{u}_{n+1}$. As an example we use $\mathbf{M} = \mathbf{\Psi}_{\mathcal{F}}$ for solving (62). Let $\mathbf{k}_n = \mathbf{\Psi}_{\mathcal{F}}\mathbf{v}_n$. We partition $\mathbf{k}_n$ as $\mathbf{\Psi}$ into

$$\mathbf{k}_n = \begin{bmatrix}\mathbf{k}_{1,n}^T & \ldots & \mathbf{k}_{K,n}^T\end{bmatrix}^T. \tag{91}$$

Using the definition of $\mathbf{\Psi}$ in (18) , the operation $\mathbf{k}_n = \mathbf{\Psi}_{\mathcal{F}}\mathbf{v}_n$ could also be performed using

$$\mathbf{K}_{k,n} = \sum_{i\in\mathcal{F}} v_{i,n}\mathbf{a}_{k,i}\mathbf{a}_{k,i}^H, \tag{92}$$

and subsequently we set

$$\mathbf{k}_{k,n} = \mathrm{vect}(\mathbf{K}_{k,n}). \tag{93}$$

This process can be highly parallelized because of the independence between the correlation matrices of each time snapshot. The matrix $\mathbf{K}_{k,n}$ can then be used to find the updates in (90).

The operation $\mathbf{M}^H\mathbf{u}$ in (90) is implemented in a similar way. Using the beamforming approach (similar to Sect. 3.4), this operation can also be done in parallel for each pixel and each snapshot. In both cases the calculations can be formulated as correlations and beamforming of parallel data paths, which means that efficient hardware implementations are feasible. Also we can consider traditional LS or WLS solutions as a special case when all the pixels belong to the free set, which means that those algorithms can also be implemented efficiently in hardware in the same way. During the calculations we work with a single beamformer at the time, and the matrix $\mathbf{\Psi}$ need not to be precalculated and stored in memory. This makes it possible to apply image formation algorithms for large images when there is a memory shortage.

The computational complexity of the algorithm is dominated by the transformation between the visibility domain and image domain (correlation and beamforming). The dirty image formation and correlation have a complexity of $O(KP^2I)$. This means that the worst-case complexity of the active set algorithm is $O(TMKP^2I)$ where $T$ is the number of active set iterations and $M$ the maximum number of Krylov iterations. A direct implementation of CLEAN for solving the imaging problem presented in Sect. 3 in a similar way would have a complexity of $O(TKP^2I)$. The proposed algorithm is therefore order $M$ times more complex, essentially because it recalculates the flux for all the pixels in the free set, while CLEAN only estimates the flux of a newly added pixel. Considering that (for a well-posed problem) solving $\mathbf{M}\mathbf{x} = \mathbf{b}$ using LSQR is algebraically equivalent to solving $\mathbf{M}^H\mathbf{M}\mathbf{x} = \mathbf{M}^H\mathbf{b}$ using CG (Fong 2011), we can use the convergence properties of CG (Demmel 1997) to obtain an indication of the required number of Krylov iterations $M$. It is found that $M$ is on the order $O(\sqrt{\mathrm{card}(\mathcal{F})})$ where $\mathrm{card}(\mathcal{F})$ is the cardinality of the free set, which is equal to the number of pixels in the free set.

In practice, many implementations of CLEAN use the FFT instead of a DFT (matched filter) for calculating the dirty image. Extending the proposed method to use similar techniques is possible and will be presented in future works.

## 6. Simulations

The performance of the proposed methods were evaluated using simulations. Because the active-set algorithm adds a single pixel to the free set at each step, it is important to investigate the effect of this procedure on extended sources and noise. For this purpose, in our first simulation set-up we used a high dynamic range simulated image with a strong point source and two weaker extended sources in the first part of the simulations. In a second set up, we made a full sky image using sources from the 3C catalog.

Following the discussion in Sec. 4.2, we defined the residual image for CLS and CLEAN as

$$\boldsymbol{\sigma}_{res} = \mathbf{\Psi}^H(\hat{\mathbf{r}} - \mathbf{\Psi}\boldsymbol{\sigma} - \mathbf{r_n}),$$

and for CPWLS, we used

$$\boldsymbol{\sigma}_{res} = \mathbf{D}^{-1}\mathbf{\Psi}^H(\hat{\mathbf{R}}^{-T} \otimes \hat{\mathbf{R}}^{-1})(\hat{\mathbf{r}} - \mathbf{\Psi}\mathbf{D}^{-1}\check{\boldsymbol{\sigma}} - \mathbf{r_n}),$$

where we assumed we know the noise covariance matrix $\mathbf{R_n}$. We also defined the reconstruction S/N on the image in dB scale as

$$\mathrm{S/N}_r = 20\log_{10}\left(\frac{\|\boldsymbol{\sigma}_{\mathrm{true}}\|}{\|\boldsymbol{\sigma}_{\mathrm{ture}} - \hat{\boldsymbol{\sigma}}\|}\right),$$

where $\boldsymbol{\sigma}_{\text{true}}$ is the true image and $\hat{\boldsymbol{\sigma}}$ is the reconstructed image.

### 6.1. Extended sources

An array of 100 dipoles ($P = 100$) with random distribution was used with the frequency range of 58-90 MHz from which we simulated three equally spaced channels. Each channel has a bandwidth of 195 kHz and was sampled at Nyquist rate. These specifications are consistent with the LOFAR telescope in LBA mode (van Haarlem et al. 2013). LOFAR uses one-second snapshots, and we did the simulation using only two snapshots, i.e., $K = 2$. We used spectrally white sources for the simulated frequency channels, which allowed us to extend the data model to one containing all frequency data by simply stacking the individual $\hat{\mathbf{r}}$ for each frequency into a single vector. Likewise, we stacked the individual $\boldsymbol{\Psi}$ into a single matrix. Since the source intensity vector $\boldsymbol{\sigma}$ is common for all frequencies, the augmented data model has the same structure as before.

The simulated source is a combination of a strong point source with intensity 40 dB and two extended structures with intensities of 0 dB. The extended structures are composed of from seven nearby Gaussian-shaped sources, one in the middle and six on a hexagon around it. (This configuration was selected to generate an easily reproducible example.) Figure 1 shows the simulated image on dB scale. The background noise level that was added is at $-10$ dB, which is also 10 dB below the extended sources. This is equivalent to a dynamic range of 40 dB and a minimum S/N of 10dB.

Figures 2a and 2b show the matched filter and MVDR dirty images, respectively. The first column of Figure 3 shows the final result of the CLEAN, CLS with the MF dirty image as upper bound, CLS with the MVDR dirty image as upper bound, and CPWLS with the MVDR dirty image as upper bound without the residual images. For each image, the extracted point sources were convolved with a Gaussian beam to smoothen the image. We used a Gaussian beam that has the same main beamwidth as the MF dirty image. The second column of Figure 3 shows the corresponding residual images as defined before, and the last column shows a cross section parallel to the $\beta_2$ axis going through the sources at the center of the image.

Remarks are:

- As expected the MVDR dirty image has a much better dynamic range ($\approx 40$ dB) and lower side-lobes compared to the MF dirty image ($\approx 15$ dB dynamic range).
- Due to a better initial dirty image and upper bound, the CPWLS deconvolution gives a better reconstruction of the image.
- The cross sections show the accuracy of the estimated intensities. This shows that not only the shape but also the magnitude of the sources are better estimated using CPWLS.
- Using the MVDR upper bound for CLS improves the estimate, illustrating the positive effect of using a proper upper bound.
- All algorithms manage to recover the intensity of the strong point source with high quality. The $\text{S/N}_r$ for CLS and CLS with MVDR is highest at 62.8 dB then CLEAN and CPWLS with 62.6 and 58.4 dB, respectively. (Only the pixel corresponding to the strong source is used to calculate these $\text{S/N}_r$.)
- CPWLS has the best performance in recovering the extended sources with $\text{S/N}_r$ of 16.5 dB compared to 11.9 and 11.7 dB for CLEAN and CLS respectively. (The pixel corresponding to the strong source was removed for calculating these $\text{S/N}_r$.)
- The residual image for CPWLS is almost two orders of magnitude lower than the residual images for CLEAN and CLS.
- While the residual image of the CLS algorithm appears very similar to the CLEAN reconstruction, CLS can guarantee that these values are inside the chosen confidence interval of six standard deviations of each pixel, while CLEAN does not provide this guarantee.

### 6.2. Full sky with 3C sources

In a second simuation set-up, we constructed an all-sky image with sources from the 3C catalog. The array configuration is the same as before with the same number of channels and snapshots. A background noise level of 0 dB (with respect to 1 Jansky) is added to the sky.

We first checked which sources from the 3C catalog are visible at the simulated date and time. From these we chose 20 sources that represent the magnitude distribution on the sky and produce the highest dynamic range available in this catalog. Table 2 shows the simulated sources with corresponding parameters. The coordinates are the $(l, m)$ coordinates at the first snapshot. Because the sources are not necessarily on the grid points, we combined the active set deconvolution with the grid corrections on the free set as described in Sec. 4.4.

Figure 4a shows the true and estimated positions for the detected sources. Because the detection mechanism was able to detect the correct number of sources, we have included the estimated fluxes in Table 2 for easier comparison. Figure 4b shows the full-sky MF dirty image. Figure 5a shows the final reconstructed image with the residual added to it (with grid corrections applied), and Figure 5c shows the same result for CLEAN.

Remarks:

- The active set algorithm with grid corrections automatically stops after adding the correct number of sources based on the detection mechanism we have incorporated in the active set method.
- Because of the grid correction, no additional sources are added to compensate for incorrect intensity estimates on the grids.
- All 20 sources are visible in the final reconstructed image, and no visible artifacts are added to the image.
- CLEAN also produces a reasonable image with all the sources visible. However, a few hundred point sources have been detected during the CLEAN iteration, most of which are the result of the strong sources that are not on the grid. Some clear artifacts are introduced (as seen in the residual image) that are also the result of the incorrect subtraction of off-grid sources.
- Figure 5b shows that the residual image using active set and grid corrections contains a "halo" around the position of the strong source—the residual image is not flat. In fact, the detection mechanism in the active set algorithm (with a threshold of 6 times the standard deviation) has correctly not considered this halo as a source. The halo is a statistical artifact due to finite samples

Table 2: Simulated sources from 3C catalog

| Names | $l$ | $m$ | Flux (Jy) | Est. flux (Jy) |
|---|---|---|---|---|
| 3C 461 | −0.30485 | 0.19131 | 11000 | 10997.61 |
| 3C 134 | 0.59704 | −0.02604 | 66 | 65.92 |
| 3C 219 | 0.63907 | 0.6598 | 44 | 44.07 |
| 3C 83.1 | 0.28778 | −0.13305 | 28 | 27.97 |
| 3C 75 | 0.30267 | −0.684 | 23 | 23.02 |
| 3C 47 | −0.042882 | −0.51909 | 20 | 19.97 |
| 3C 399.2 | −0.97535 | 0.20927 | 19 | 18.97 |
| 3C 6.1 | −0.070388 | 0.47098 | 16 | 15.99 |
| 3C 105 | 0.57458 | −0.60492 | 15 | 15.10 |
| 3C 158 | 0.9017 | −0.12339 | 14 | 14.01 |
| 3C 231 | 0.28956 | 0.72005 | 13 | 13.02 |
| 3C 303 | −0.1511 | 0.95402 | 12.5 | 12.51 |
| 3C 277.1 | 0.12621 | 0.93253 | 12 | 12.03 |
| 3C 320 | −0.3597 | 0.93295 | 11.5 | 11.62 |
| 3C 280.1 | 0.15171 | 0.98709 | 11 | 10.95 |
| 3C 454.2 | −0.29281 | 0.31322 | 10.5 | 10.48 |
| 3C 458 | −0.61955 | −0.56001 | 10 | 10.01 |
| 3C 223.1 | 0.67364 | 0.68376 | 9.5 | 9.63 |
| 3C 19 | −0.23832 | −0.30028 | 9 | 8.87 |
| 3C 437.1 | −0.83232 | −0.24924 | 5 | 4.99 |

and will be reduced in magnitude by longer observations with a rate proportional to $1/\sqrt{NK^2}$.

– The CLEAN algorithm requires more than 100 sources to model the image. This is mainly because of the the strong off–grid source (Cassiopeia A). This illustrates that while CLEAN is less complex than the proposed method when the number of detected sources are equal, in practice CLEAN might need many more sources to model the same image.

## 7. Conclusions

Based on a parametric model and constraints on the intensities, we have formulated image deconvolution as a weighted least squares optimization problem with inequality constraints. We first showed that the classical (matched filter) dirty image is an upper bound, but a much tighter upper bound is provided by the "MVDR dirty image". The conditioning of the problem can be improved by a preconditioning step, which is also related to the MVDR dirty image.

Second, the constrained least squares problem was solved using an active-set-based method. The relation between the resulting method and sequential source removing techniques such as CLEAN was explained. The theoretical background of the active set methods can be used for deeper insight into how the sequential techniques work. In particular, the active set algorithm uses a detection threshold with a known false alarm, which can be set such that no false sources appear in the image, and we showed that by introducing a grid correcting step into the active set iterations, we can improve both the detection of the sources and the estimation of their intensities.

Third, the Khatri-Rao structure of the data model was used in combination with Krylov based techniques to solve the linear systems involved in the deconvolution process with less storage and complexity. The complexity of the algorithm is higher than that of classical sequential source removing techniques (by a factor proportional to the square root of the detected number of sources), because the detected source intensities are re-estimated by the Krylov subspace technique after each step of the active set iteration. However, the proposed algorithm has a better detection mechanism compared to classical CLEAN, which leads to a lower number of sources to model the image. As a result, the overall complexity is expected to be comparable. We also expect that the performance of the algorithm can be readily improved because the updates by the active set iterations are one-dimensional (one source is added or removed), and this can be exploited to update the Krylov subspaces accordingly, rather than computing them each time from scratch. This is left for future work.

The simulations show that the proposed CPWLS algorithm provides improved spatial structure and improved intensity estimates compared to CLEAN-based deconvolution of the classical dirty image. A particularly attractive aspect is the demonstrated capability of the algorithm to perform automated source detection, which will be of interest for upcoming large surveys.

## Appendix A: Relation between WLS, natural, and robust weighting

Natural weighting is a technique to improve the detection of weak sources by promoting the visibility values that have a better signal-to-noise-ratio (Briggs 1995). This is done by dividing each visibility sample by the variance of noise on that sample (while assuming that the noise on each sample is independent). Considering that the visibility samples are the elements of the covariance matrix $\hat{\mathbf{R}}_k$ we can model the sample visibilities as

$$\hat{\mathbf{r}}_k = \mathbf{r}_k + \boldsymbol{\epsilon}, \tag{A.1}$$

where $\boldsymbol{\epsilon}$ is the complex noise on the samples. As discussed in Appendix C, $\hat{\mathbf{R}}$ has a Wishart distribution and for a large number of samples $N$ we have $\hat{\mathbf{r}}_k \sim \mathcal{N}(\mathbf{r}_k, (\mathbf{R}_k^T \otimes \mathbf{R}_k)/N)$. This means that $\boldsymbol{\epsilon} = \hat{\mathbf{r}}_k - \mathbf{r}_k$ has a complex Gaussian distribution $\mathcal{N}(\mathbf{0}, (\mathbf{R}_k^T \otimes \mathbf{R}_k)/N)$. Because astronomical sources are usually much weaker than the system noise, it is common to use the approximation $\mathbf{R}_k \approx \mathbf{R}_{\mathbf{n},k}$. With this approximation and using the independence of system noise on each receiving element (antenna or station), we can assume that $\mathbf{R}_{\mathbf{n},k}$ is diagonal and that $(\mathbf{R}_k^T \otimes \mathbf{R}_k)/N \approx (\mathbf{R}_{\mathbf{n},k}^T \otimes \mathbf{R}_{\mathbf{n},k})/N$ is also a diagonal approximation of the noise covariance matrix on the visibility samples. With this framework we can write the natural weighting as

$$\hat{\mathbf{r}}_{\text{natural}} = N(\mathbf{R}_{\mathbf{n},k}^{-T} \otimes \mathbf{R}_{\mathbf{n},k}^{-1})\hat{\mathbf{r}}_k. \tag{A.2}$$

This shows that natural weighting is a very reasonable approximation of the weighting used when solving (28) for WLS (except for a factor $N$ that drops out from both sides).

Next, we relate WLS to Robust Weighting (Briggs 1995) by assuming slightly different simplifications. Let us assume that $\mathbf{R}_{\mathbf{n},k} = \sigma_{\mathbf{n}}^2\mathbf{I}$ and let us consider a single source with intensity $\sigma$ then we have for

$$\begin{aligned}\mathbf{R}_k^{-1} &= (\mathbf{R}_{\mathbf{n},k} + \sigma\tilde{\mathbf{a}}_k\tilde{\mathbf{a}}_k^H)^{-1} \\ &= \mathbf{R}_{\mathbf{n},k}^{-1} - \frac{\sigma\mathbf{R}_{\mathbf{n},k}^{-1}\tilde{\mathbf{a}}_k\tilde{\mathbf{a}}_k^H\mathbf{R}_{\mathbf{n},k}^{-1}}{1+\sigma\tilde{\mathbf{a}}_k^H\mathbf{R}_{\mathbf{n},k}^{-1}\tilde{\mathbf{a}}_k} \\ &= \frac{1}{\sigma_{\mathbf{n}}^2}\left(\mathbf{I} - \frac{\tilde{\mathbf{a}}_k\tilde{\mathbf{a}}_k^H}{1+\frac{\sigma_{\mathbf{n}}^2}{\sigma}}\right).\end{aligned} \tag{A.3}$$

Compared to natural weighting, now not only the noise power but also the available signal power is taken into account for the weighting. The term $1/(1+\sigma_{\mathbf{n}}^2/\sigma)$ is the same as the parametric Wiener filter in the Fourier domain as given by (Briggs 1995) which relates Robust Weighting to standard signal processing concepts. However Robust Weighting also takes the visibility sampling of the gridded uv-plane into account when calculating the weights, which is not explained in the derivation above. Hence the exact relation between Robust Weighting and WLS is still missing. This relation is interesting and should be addressed in future works.

## Appendix B: Using beamformers to find upper bounds

In this section we show how to use linear beamformers to define an upper bound for the image. We also show that ASSC gives the tightest bound if we use true covariance matrix $\mathbf{R}$. We also show that the MVDR dirty image is an upper bound but tighter than MF dirty image.

Let us define a beamformer $\mathbf{w}_{\mathrm{MF},i} = \frac{1}{\sqrt{K}}\mathbf{a}_i$, then we observe that each pixel in the MF dirty image is the output of this beamformer:

$$\sigma_{\mathrm{MF},i} = \mathbf{w}_{\mathrm{MF},i}^H \mathbf{R} \mathbf{w}_{\mathrm{MF},i}. \tag{B.1}$$

As indicated in Sec. 3.2, we can extend this concept to a more general beamformer $\mathbf{w}_i$. The output power of this beamformer, in the direction of the $i$th pixel, becomes

$$\sigma_{\mathbf{w},i} = \mathbf{w}_i^H \mathbf{R} \mathbf{w}_i = \sigma_i \mathbf{w}_i^H (\mathbf{I}_K \circ \mathbf{A}^i)(\mathbf{I}_K \circ \mathbf{A}^i)^H \mathbf{w}_i + \mathbf{w}_i^H \mathbf{R}_r \mathbf{w}_i\,. \tag{B.2}$$

If we require that

$$\mathbf{w}_i^H (\mathbf{I}_K \circ \mathbf{A}^i)(\mathbf{I}_K \circ \mathbf{A}^i)^H \mathbf{w}_i = 1 \tag{B.3}$$

we have

$$\sigma_{\mathbf{w},i} = \sigma_i + \mathbf{w}_i^H \mathbf{R}_r \mathbf{w}_i\,. \tag{B.4}$$

As before, the fact that $\mathbf{R}_r$ is positive definite implies that

$$\sigma_i \le \sigma_{\mathbf{w},i}\,. \tag{B.5}$$

We can easily verify that $\mathbf{w}_{\mathrm{MF},i}$ satisfies (B.3) and hence $\sigma_{\mathrm{MF},i}$ is a specific upper bound. We can translate the problem of finding the tightest upper bound to the following optimization question:

$$\begin{aligned} \sigma_{\mathrm{opt},i} = \min_{\mathbf{w}_i} \mathbf{w}_i^H \mathbf{R} \mathbf{w}_i \\ \text{s.t. } \mathbf{w}_i^H (\mathbf{I}_K \circ \mathbf{A}^i)(\mathbf{I}_K \circ \mathbf{A}^i)^H \mathbf{w}_i = 1 \end{aligned} \tag{B.6}$$

where $\sigma_{\mathrm{opt},i}$ would be this tightest upper bound.

To solve this optimization problem we follow standard optimization techniques and define the Lagrangian and take derivatives with respect to $\mathbf{w}$ and the Lagrange multiplier $\mu$. This leads to the following system

$$\mathbf{w} = \mu \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)(\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{w} \tag{B.7}$$

$$1 = \mathbf{w}^H (\mathbf{I}_K \circ \mathbf{A}_i)(\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{w} \tag{B.8}$$

Because $\mathbf{R}$ is full–rank and (B.8) we can model $\mathbf{w}$ as

$$\mathbf{w} = \mu \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)\mathbf{x}. \tag{B.9}$$

Filling back into (B.7) we have

$$\begin{aligned} &\mu \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)\mathbf{x} \\ &= \mu^2 \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)(\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)\mathbf{x} \end{aligned} \tag{B.10}$$

and

$$\begin{aligned} &(\mathbf{I}_K \circ \mathbf{A}_i)\mathbf{x} \\ &= \mu (\mathbf{I}_K \circ \mathbf{A}_i)(\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)\mathbf{x}. \end{aligned} \tag{B.11}$$

Multiplying both sides of this equation by $(\mathbf{I}_K \circ \mathbf{A}_i)^H$ we get

$$\mathbf{x} = \mu (\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)\mathbf{x}. \tag{B.12}$$

Doing the same for (B.8) we have

$$\mu^2 \mathbf{x}^H (\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)(\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)\mathbf{x} = 1. \tag{B.13}$$

Now we use (B.12) and we find

$$\mathbf{x}^H \mathbf{x} = 1, \tag{B.14}$$

which makes finding $\mathbf{x}$ an eigenvalue problem. By taking a closer look at the matrix $(\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i)$ we find that this matrix is diagonal

$$\begin{aligned} &(\mathbf{I}_K \circ \mathbf{A}_i)^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}_i) \\ &= \begin{bmatrix} \mathbf{a}_{1,i}^H \mathbf{R}_1^{-1} \mathbf{a}_{1,i} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{a}_{i,2}^H \mathbf{R}_2^{-1} \mathbf{a}_{i,2} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{a}_{i,K}^H \mathbf{R}_K^{-1} \mathbf{a}_{i,K} \end{bmatrix} \end{aligned} \tag{B.15}$$

and hence $\mathbf{x} = \mathbf{e}_m$ is an elementary vector with all entries equal to zero except for $m$th entry which equals unity. $m$ is the index corresponding to largest eigenvalue, $\lambda_{\max}$, and from (B.12) we have $\mu = 1/\lambda_{\max}$. Filling back for $\mathbf{w}$ we find

$$\mathbf{w}_{i,\mathrm{opt}} = \frac{1}{\mathbf{a}_{i,m} \mathbf{R}_m^{-1} \mathbf{a}_{i,m}} \mathbf{R}^{-1} (\mathbf{e}_m \otimes \mathbf{a}_{i,m}) \tag{B.16}$$

and the output of the beamformer

$$\begin{aligned} \sigma_{opt} &= \mathbf{w}_{i,\mathrm{opt}}^H \mathbf{R} \mathbf{w}_{i,\mathrm{opt}} \\ &= \frac{\mathbf{a}_{i,m}^H \mathbf{R}_m^{-1} \mathbf{a}_{i,m}}{(\mathbf{a}_{i,m}^H \mathbf{R}_m^{-1} \mathbf{a}_{i,m})^2} \\ &= \frac{1}{\mathbf{a}_{i,m}^H \mathbf{R}_m^{-1} \mathbf{a}_{i,m}} \\ &= \min_k \left( \frac{1}{\mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i}} \right). \end{aligned} \tag{B.17}$$

In order to reduce the variance of this solution we suggest to find a beamformer that instead of (B.3) satisfies the slightly different normalization constraint

$$\mathbf{w}_i^H \mathbf{a}_i = \sqrt{K}\,. \tag{B.18}$$

We show that the expected value of the resulting dirty image constitutes a larger upper bound than the ASSC (40), but because the output power of this beamformer depends

on more than one snapshot it has a lower variance than ASSC, so that it is more robust in practice.

With this constraint, the beamforming problem is

$$\mathbf{w}_i = \arg\min_{\mathbf{w}_i} \mathbf{w}_i^H \mathbf{R} \mathbf{w}_i \quad \text{s.t. } \mathbf{w}_i^H \mathbf{a}_i = \sqrt{K} \tag{B.19}$$

which is recognized as the classical minimum variance distortionless response (MVDR) beamforming problem (Capon 1969). Thus, the solution is given in closed form as

$$\mathbf{w}_{\text{MVDR},i} = \frac{\sqrt{K}}{\mathbf{a}_i^H \mathbf{R}^{-1} \mathbf{a}_i} \mathbf{R}^{-1} \mathbf{a}_i. \tag{B.20}$$

To demonstrate that this image is still an upper bound we show that

$$\alpha := \mathbf{w}_i^H (\mathbf{I}_K \circ \mathbf{A}^i)(\mathbf{I}_K \circ \mathbf{A}^i)^H \mathbf{w}_i \geq 1\,. \tag{B.21}$$

Indeed, inserting (B.20) into this inequality gives

$$\begin{aligned} & K \frac{\mathbf{a}_i^H \mathbf{R}^{-1} (\mathbf{I}_K \circ \mathbf{A}^i)(\mathbf{I}_K \circ \mathbf{A}^i)^H \mathbf{R}^{-1} \mathbf{a}_i}{(\mathbf{a}_i^H \mathbf{R}^{-1} \mathbf{a}_i)^2} \\ &= K \frac{\sum_k (\mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i})^2}{\left(\sum_k \mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i}\right)^2} \\ &= K \frac{\mathbf{h}^T \mathbf{h}}{\mathbf{h}^T \mathbf{1}_K \mathbf{1}_K^T \mathbf{h}} \geq K \frac{1}{\lambda_{\max}(\mathbf{1}_K \mathbf{1}_K^T)} = 1, \end{aligned} \tag{B.22}$$

where $\mathbf{h} = (\mathbf{I}_K \circ \mathbf{A}^i)^H \mathbf{R}^{-1} \mathbf{a}_i$ is a $K \times 1$ vector with entries $h_k = \mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i}$ and $\lambda_{\max}(\cdot)$ is the largest eigenvalue of of the argument matrix. Hence, a similar reasoning as in (B.2) gives

$$\sigma_{\text{MVDR},i} = \alpha \sigma_i + \mathbf{w}_{\text{MVDR},i}^H \mathbf{R}_r \mathbf{w}_{\text{MVDR},i} \geq \sigma_i$$

which leads to (42).

Note that $\mathbf{w}_{\text{MF},i}$ also satisfies the constraint in (B.19), i.e. $\mathbf{w}_{\text{MF},i}^H \mathbf{a}_i = \sqrt{K}$, but does not necessary minimize the output power $\mathbf{w}_i^H \mathbf{R} \mathbf{w}_i$, therefore the MVDR dirty image is smaller than the MF dirty image: $\boldsymbol{\sigma}_{\text{MVDR}} \leq \boldsymbol{\sigma}_{\text{MF}}$. Thus it is a tighter upper bound. This relation also holds if $\mathbf{R}$ is replaced by the sample covariance $\hat{\mathbf{R}}$.

## Appendix C: Variance of the dirty image

To find the confidence intervals for the dirty images we need to find estimates for the variance of both matched filter and MVDR dirty images. In our problem the sample covariance matrix is obtained by squaring samples from a Gaussian process. This means that $N\hat{\mathbf{R}} \sim \mathcal{W}_p(\mathbf{R}, N)$ where $\mathcal{W}_p(\mathbf{R}, N)$ is the Wishart distribution function of order $p$ with expected value equal to $\mathbf{R}$ and $N$ degrees of freedom. For any deterministic vector $\boldsymbol{\zeta}$,

$$N \boldsymbol{\zeta}^H \hat{\mathbf{R}} \boldsymbol{\zeta} \sim \boldsymbol{\zeta}^H \mathbf{R} \boldsymbol{\zeta}\ \chi^2(N). \tag{C.1}$$

where $\chi^2(N)$ is the standard $\chi^2$ distribution with $N$ degrees of freedom. In radio astronomical applications $N$ is usually very large and we can approximate this $\chi^2$ distribution with a Gaussian such that $\boldsymbol{\zeta}^H \hat{\mathbf{R}} \boldsymbol{\zeta} \sim \mathcal{N}(\boldsymbol{\zeta}^H \mathbf{R} \boldsymbol{\zeta}, (\boldsymbol{\zeta}^H \mathbf{R} \boldsymbol{\zeta})^2 / N)$. The variance of the matched filter dirty image is given by

$$\text{Var}(\sigma_{\text{MF},i}) = \frac{1}{NK^2} \sum_k (\mathbf{a}_{k,i}^H \mathbf{R} \mathbf{a}_{k,i})^2.$$

Using this result we can find the $x\%$ confidence interval which results in an increase of the upper bound such that

$$\boldsymbol{\sigma} \leq \hat{\boldsymbol{\sigma}}_{\text{MF}} + \alpha \sqrt{\text{Var}(\hat{\boldsymbol{\sigma}}_{\text{MF}})}, \tag{C.2}$$

where $\alpha$ is chosen depending on $x$. Requiring at most a single false detection on the entire image translate into $\alpha \approx 6$.

When we estimate the MVDR dirty image from sample covariance matrices we need to be more careful, mainly because the result is biased and we need to correct for that bias. For each pixel of the MVDR dirty image obtained from sample covariance matrices we have

$$\hat{\sigma}_{\text{MVDR},i} = K g(Z) = \frac{K}{\sum_k \mathbf{a}_{k,i}^H \hat{\mathbf{R}}_k^{-1} \mathbf{a}_{k,i}} \tag{C.3}$$

where $g(Z) = 1/Z$ and $Z = \sum_k \mathbf{a}_{k,i}^H \hat{\mathbf{R}}_k^{-1} \mathbf{a}_{k,i}$. Using a perturbation model $Z = Z_0 + \Delta Z$ and a Taylor approximation we find

$$\begin{aligned} g(Z) &\approx \frac{1}{Z_0} - \frac{1}{Z_0^2} \Delta Z \\ &\approx \frac{1}{Z_0^2} (Z_0 - \Delta Z). \end{aligned} \tag{C.4}$$

Let $Z_0 = \mathcal{E}\{Z\}$ then $\mathcal{E}\{\Delta Z\} = 0$ and $\mathcal{E}\{g(Z)\} \approx 1/Z_0$. We would like this estimate to be unbiased which means that we want

$$\mathcal{E}\{g(Z)\} \approx \frac{1}{\sum_k \mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i}} \tag{C.5}$$

however we have,

$$\begin{aligned} Z_0 &= \sum_k \mathbf{a}_{k,i} \mathcal{E}\{\hat{\mathbf{R}}_k^{-1}\} \mathbf{a}_{k,i} \\ &= \sum_k \mathbf{a}_{k,i}^H \frac{N \mathbf{R}_k^{-1}}{N - p} \mathbf{a}_{k,i} \\ &= \frac{N}{N - P} \sum_k \mathbf{a}_{k,i}^H \mathbf{R}_k^{-1} \mathbf{a}_{k,i} \end{aligned} \tag{C.6}$$

where we have used $\mathcal{E}\{\hat{\mathbf{R}}^{-1}\} = \frac{N}{N-P} \mathbf{R}^{-1}$ (Shaman 1980). So in order to remove this bias we need to scale it by a correction factor

$$C = \frac{N}{N - P} \tag{C.7}$$

and

$$\hat{\sigma}_{\text{MVDR},i} = CK g(Z). \tag{C.8}$$

Now we need to find an estimate for the variance of the MVDR dirty image. Using (C.4) we see that the first order approximation of $\text{Var}(g(Z)) \approx \text{Var}(Z)/Z_0^4$. We find $\text{Var}(Z)$ using the independence of each snapshot so we can write

$$\text{Var}(Z) = \sum_k \text{Var}(\mathbf{a}_{k,i}^H \hat{\mathbf{R}}_k^{-1} \mathbf{a}_{k,i}). \tag{C.9}$$

In order to find $\text{Var}(\mathbf{a}_{k,i}^H \hat{\mathbf{R}}_k^{-1} \mathbf{a}_{k,i})$ we need to use some properties of the complex inverse Wishart distribution. A matrix

has complex inverse Wishart distribution if it's inverse has a complex Wishart distribution (Shaman 1980). Let us define an invertible matrix $\mathbf{B}$ as

$$\mathbf{B} = [\mathbf{a}_{k,i} \quad \mathbf{B}_1], \tag{C.10}$$

then $\mathbf{X} = (\mathbf{B}\hat{\mathbf{R}}^{-1}\mathbf{B}^H)/N$ has an inverse Wishart distribution because $\mathbf{X}^{-1} = N(\mathbf{B}^{-H}\hat{\mathbf{R}}\mathbf{B}^{-1})$ has a Wishart distribution. In this case $\mathbf{X}_{11} = (\mathbf{a}_{k,i}^H\hat{\mathbf{R}}^{-1}\mathbf{a}_{k,i})/N$ also has an inverse Wishart distribution with less degrees of freedom. The covariance of an inverse Wishart matrix is derived in (Shaman 1980), however because we are dealing only with one element, this results simplifies to

$$\mathrm{Var}(N\mathbf{X}_{11}) = \frac{N^2}{(N-P)^2(N-P-1)}(\mathbf{a}_{k,i}^H\mathbf{R}^{-1}\mathbf{a}_{k,i})^2. \tag{C.11}$$

The variance of the unbiased MVDR dirty image is thus given by

$$\begin{aligned}\mathrm{Var}(\hat{\sigma}_{\mathrm{MVDR},i}) &= \mathrm{Var}(CKg(Z)) \\ &\approx \frac{K^2}{(N-P-1)}\frac{\sum_k(\mathbf{a}_{k,i}^H\mathbf{R}_k^{-1}\mathbf{a}_{k,i})^2}{\left(\sum_k \mathbf{a}_{k,i}\mathbf{R}_k^{-1}\mathbf{a}_{k,i}\right)^4}.\end{aligned}$$

Now that we have the variance we can use the same method that we used for MF dirty image to find $\alpha$ and

$$\boldsymbol{\sigma} \leq \hat{\boldsymbol{\sigma}}_{\mathrm{MVDR}} + \alpha\sqrt{\mathrm{Var}(\hat{\boldsymbol{\sigma}}_{\mathrm{MVDR}})}. \tag{C.12}$$

## References


Barrett, R., Berry, M., Chan, T. F., et al. 1994, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition (Philadelphia, PA: SIAM)

Ben-David, C. & Leshem, A. 2008, Selected Topics in Signal Processing, IEEE Journal of, 2, 670

Bhatnager, S. & Cornwell, T. 2004, Astronomy and Astrophysics, 426, 747

Boyd, S. & Vandenberghe, L. 2004, Convex Optimization (Cambridge University Press)

Briggs, D. S. 1995, PhD thesis, The New Mexico Institute of Mining and Technology, Socorro, New Mexico

Capon, J. 1969, Proceedings of the IEEE, 1408

Carrillo, R., McEwen, J., & Wiaux, Y. 2012, Monthly Notices of the Royal Astronomical Society, 426, 1223

Carrillo, R. E., McEwen, J. D., & Wiaux, Y. 2014, Monthly Notices of the Royal Astronomical Society, 439, 3591

Chen, C.-Y. & Vaidyanathan, P. 2007, Signal Processing, IEEE Transactions on, 55, 4139

Choi, S.-C. T. 2006, PhD thesis, Stanford University

Cornwell, T. 2008, Selected Topics in Signal Processing, IEEE Journal of, 2, 793

Demmel, J. W. 1997, Applied Numerical Linear Algebra (SIAM)

Fong, D. C.-l. 2011, PhD thesis, Stanford University

Frieden, B. 1972, Journal of the Optical Society of America, 62, 511

Gill, P. E., Murray, W., & Wright, M. H. 1981, Practical optimization (London: Academic Press Inc. [Harcourt Brace Jovanovich Publishers]), xvi+401

Golub, G. & Kahan, W. 1965, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, 2, 205

Gu, Y. & Leshem, A. 2012, Signal Processing, IEEE Transactions on, 60, 3881

Gull, S. & Daniell, G. 1978, Nature, 272, 686

H. Junklewitz, M.R. Bell, & T. Enßlin. 2015, A&A

Högbom, J. A. 1974, Astron. Astrophys. Suppl, 15, 417

Ipsen, I. C. F. & Meyer, C. D. 1998, The American Mathematical Monthly, 105, pp. 889

Leshem, A. & van der Veen, A. 2000, IEEE Trans. on Information Theory, Special issue on information theoretic imaging, 1730

Leshem, A., van der Veen, A., & Boonstra, A. J. 2000, The Astrophysical Journal Supplements, 355

Levanda, R. & Leshem, A. 2008, Electrical and Electronics Engineers in Israel, 2008. IEEEI 2008. IEEE 25th Convention of, 716

Levanda, R. & Leshem, A. 2013, Signal Processing, IEEE Transactions on, 61, 5063

Lochner, M., Natarajan, I., Zwart, J. T., et al. 2015, Monthly Notices of the Royal Astronomical Society, 450, 1308

Ottersten, B., Stoica, P., & Roy, R. 1998, Digital Signal Processing, 8, 185

Paige, C. C. & Saunders, M. A. 1982, ACM Trans. Math. Softw., 8, 43

Rau, U., Bhatnagar, S., Voronkov, M., & Cornwell, T. 2009, Proceedings of the IEEE, 97, 1472

Reid, R. 2006, Monthly Notices of the Royal Astronomical Society, 367, 1766

Shaman, P. 1980, Journal of Multivariate Analysis, 10, 51

van Haarlem, M. P., Wise, M. W., Gunst, A. W., et al. 2013, A&A, 556, A2

Wiaux, Y., Jacques, L., Puy, G., Scaife, A., & Vandergheynst, P. 2009, Monthly Notices of the Royal Astronomical Society, 395, 1733

Wijnholds, S. & van der Veen, A.-J. 2008, Selected Topics in Signal Processing, IEEE Journal of, 2, 613

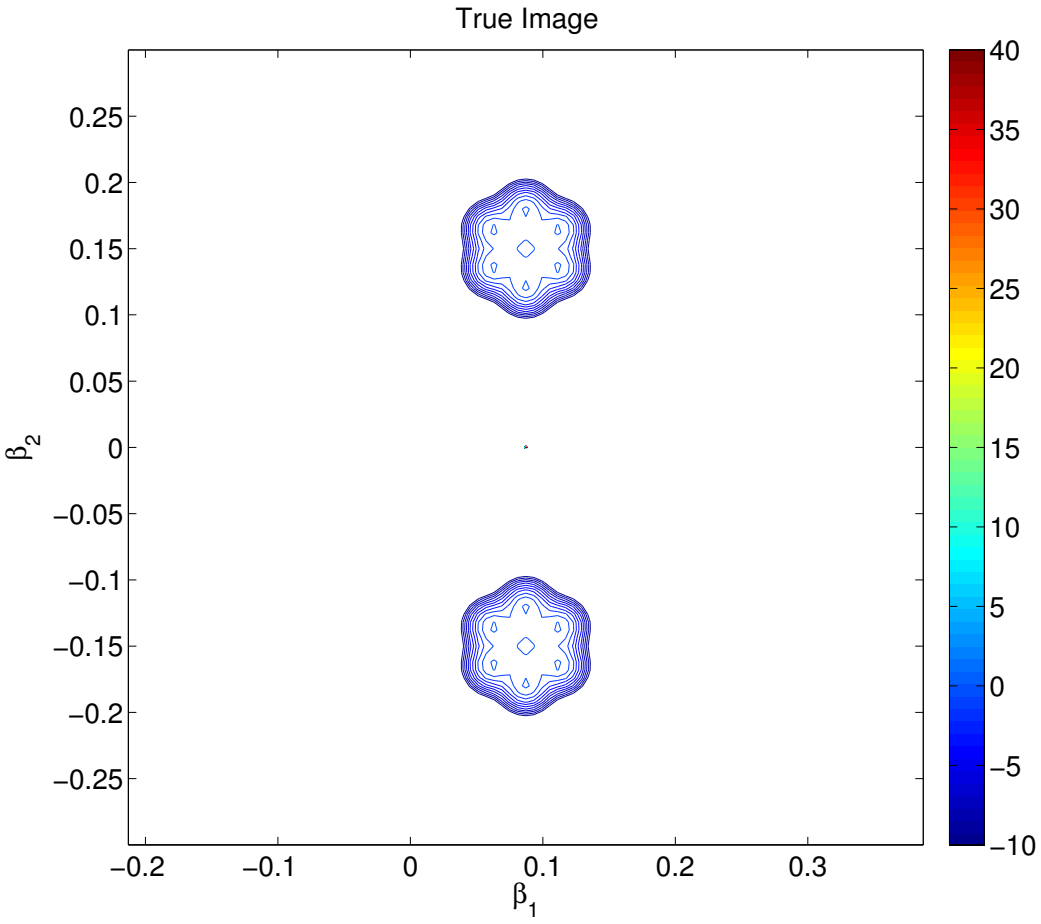
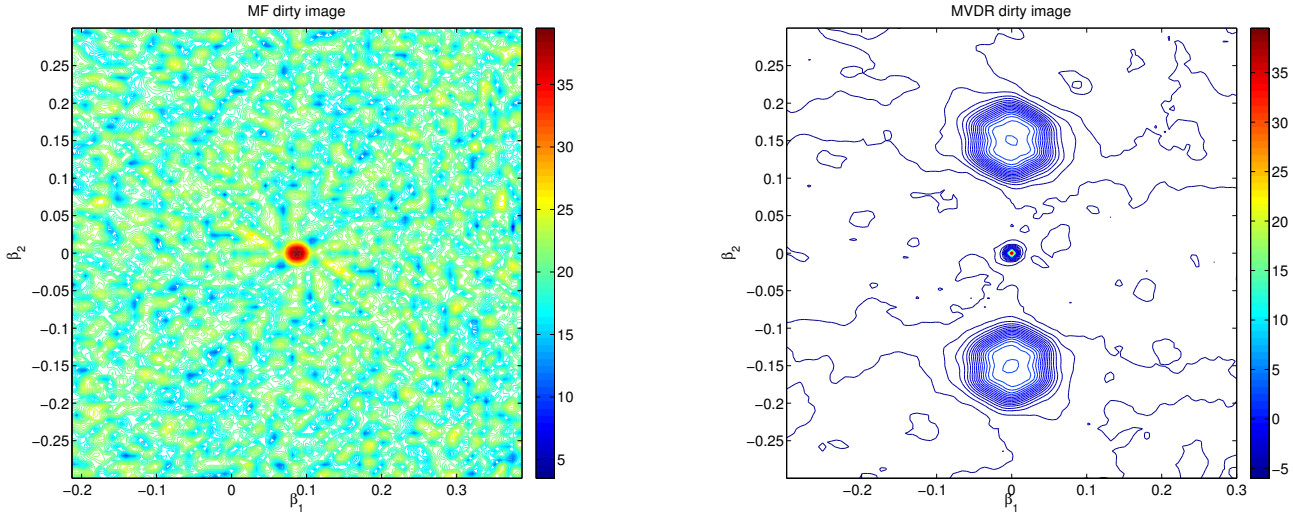


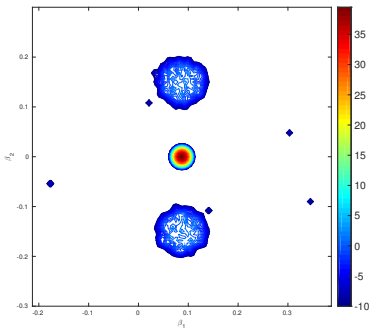

Fig. 1: Contoured true source on dB scale



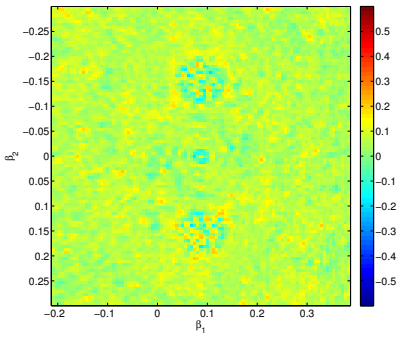

(a) MF dirty image

(b) MVDR dirty image

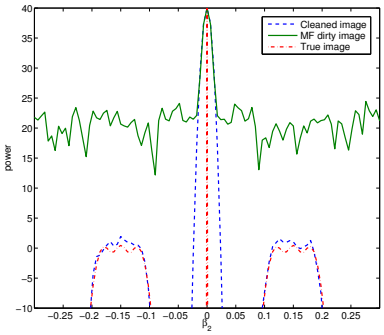
Fig. 2: Contoured dirty images on dB scale



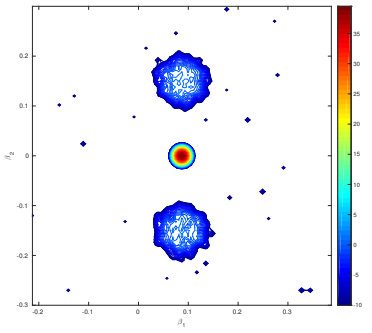
(a) CLEAN Image



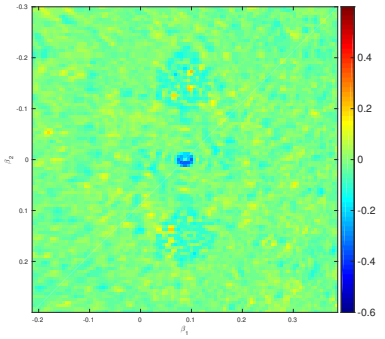
(b) CLEAN Residual image



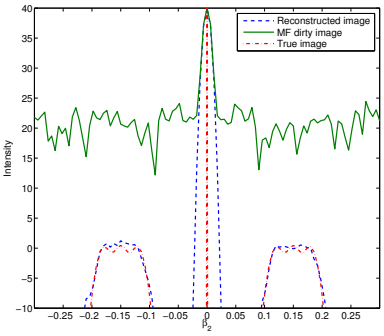

(c) CLEAN cross-section



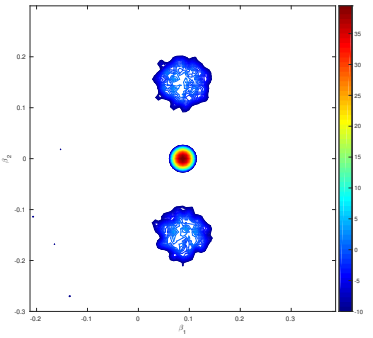
(d) Solution of CLS + MF



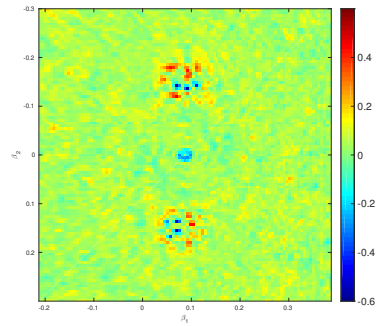
(e) Residual image for CLS + MF



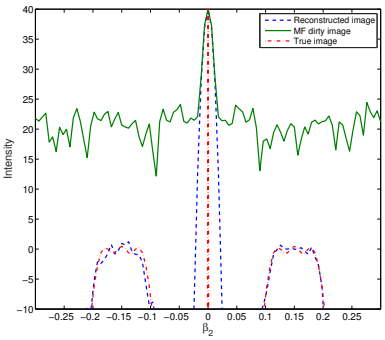

(f) Cross-section for CLS + MF



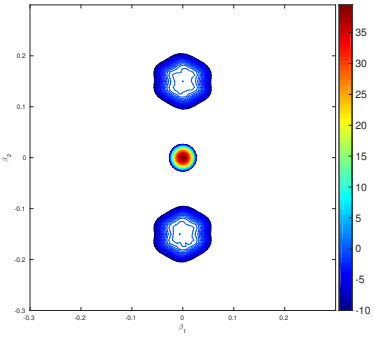
(g) Solution of CLS + MVDR



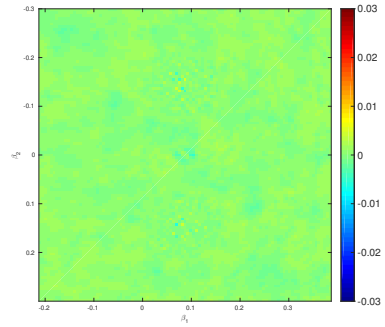
(h) Residual image for CLS + MVDR



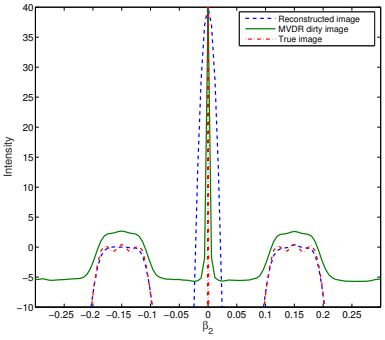

(i) Cross-section for CLS image + MVDR



(j) CPWLS image

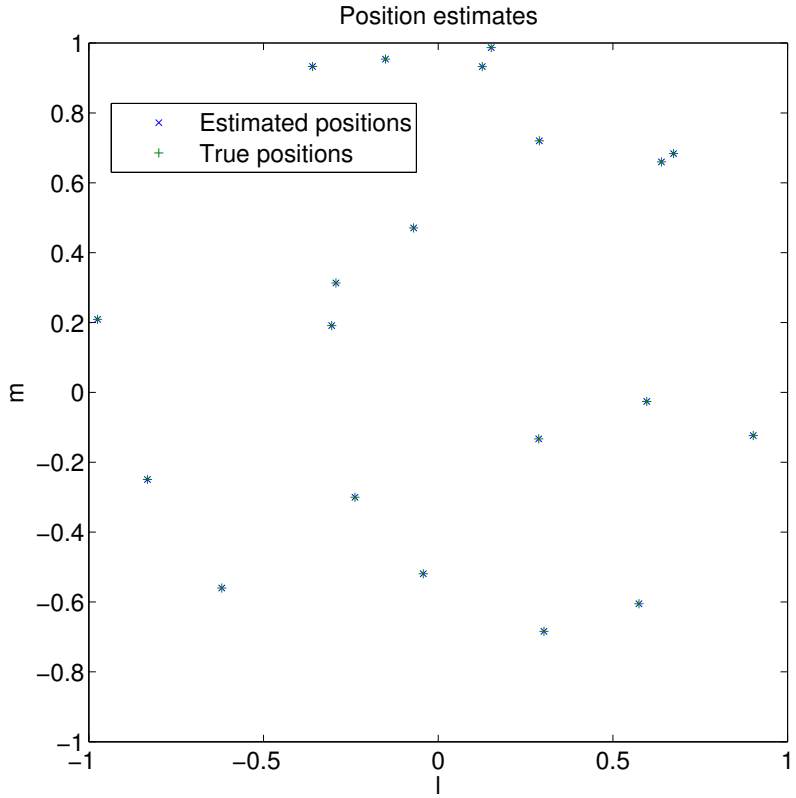


(k) CPWLS residual image

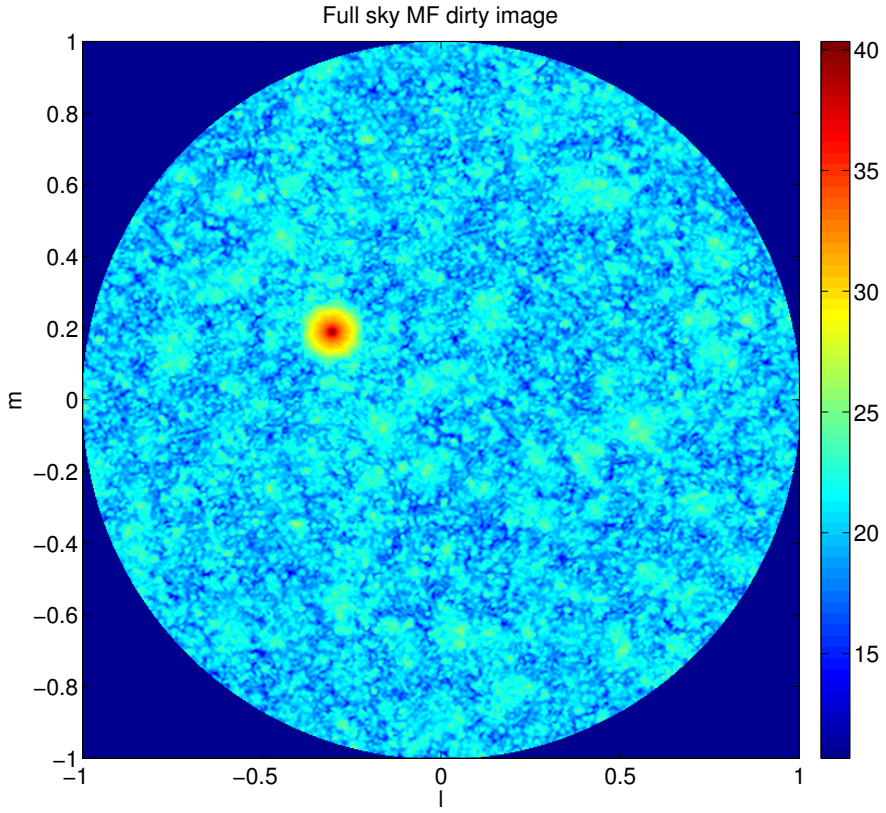



(l) MVDR dirty image and CPWLS cross-section

Fig. 3: Extended source simulations. Units for the first and third columns are in dB. Linear scale is used for residual images (second column).

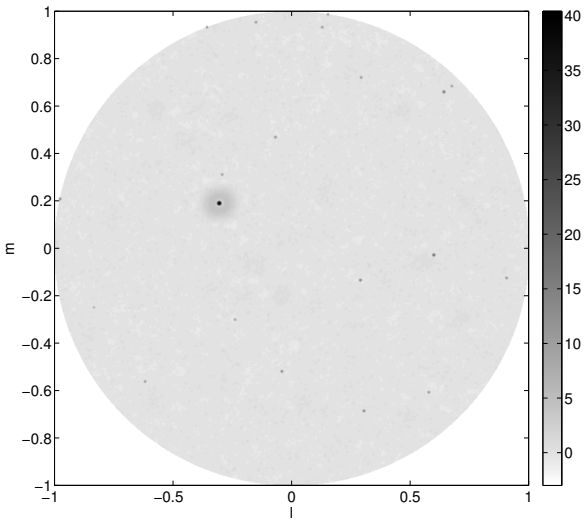



(a) Position estimates

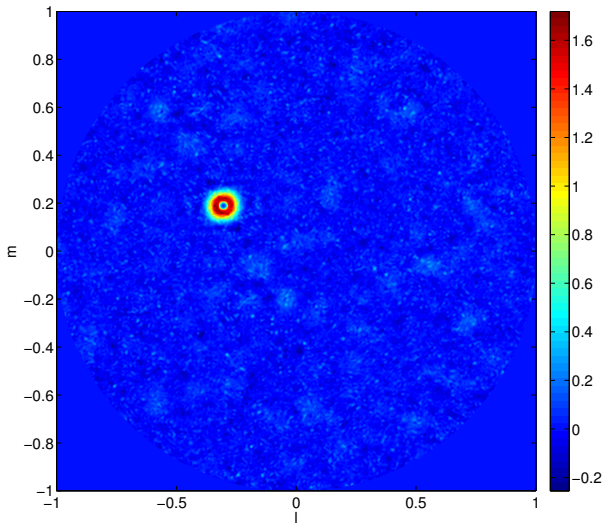



(b) Full sky MF dirty image in dB (with respect to 1 Jy)

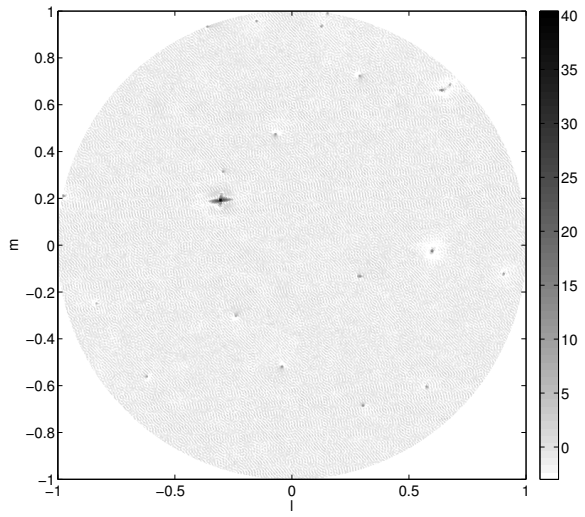
Fig. 4: Point source simulations



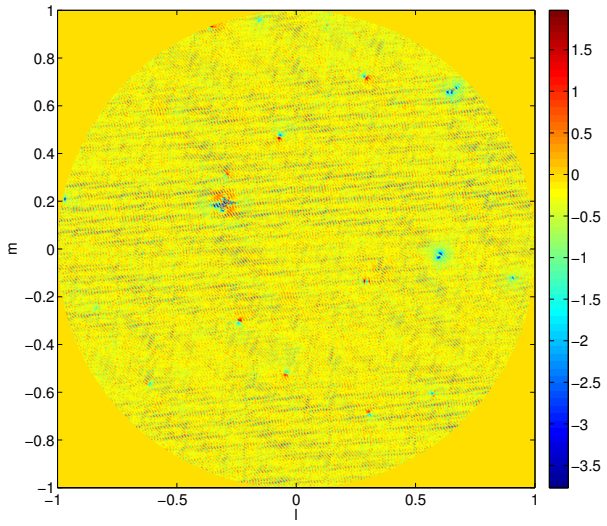

(a) Reconstructed image with grid correction plus residual image




(b) Residual image using active set and grid correction




(c) Reconstructed image with CLEAN plus residual image




(d) Residual image using CLEAN

Fig. 5: Reconstructed images in dB (with respect to 1 Jy) scale and residual images on linear scale