Efficient Calibration of Radio Interferometers Using Block LDU Decomposition

Ahmad Mouri Sardarabadi[1], Alle-Jan van der Veen[2] and Léon V. E. Koopmans[1]

***Abstract*—Having an accurate calibration method is crucial for any scientific research done by a radio telescope. The next generation radio telescopes such as the Square Kilometre Array (SKA) will have a large number of receivers which will produce exabytes of data per day. In this paper we propose new direction-dependent and independent calibration algorithms that, while requiring much less storage during calibration, converge very fast. The calibration problem can be formulated as a non-linear least square optimization problem. We show that combining a block-LDU decomposition with Gauss-Newton iterations produces systems of equations with convergent matrices. This allows significant reduction in complexity per iteration and very fast converging algorithms. We also discuss extensions to direction-dependent calibration. The proposed algorithms are evaluated using simulations.**

***Index Terms*—Calibration, Radio Astronomy, Non-Linear Optimization, Covariance Matching**

I. Introduction

One of the key challenges for current and future radio–telescopes, such as LOFAR (LOw Frequency ARray) [4] and SKA (Square Kilometre Array) [10], is the accurate calibration of the instrument with reasonable computational complexity. Modern radio telescopes consist of many receivers which can be large dishes or sub-arrays beamformed into a single element. The calibration problem for radio interferometers has already been addressed by several authors [1], [6]–[8], [12]. During a calibration cycle we use our current knowledge of the radio sources (fore example known from previous observations), to find the gains of the receivers. However, because a typical interferometer has a direction-dependent behavior, we need to solve these gains for different directions [9], [13]. In this paper we assume to have access to an accurate model for the sources and we are interested in developing computationally efficient algorithms that scale well with multi-channel observations.

Based on the resolution of the instrument, in order to avoid source smearing, the observations are divided into small snapshots (order of seconds). However, in order to study very weak sources we need to observe for a very long time (e.g. hundreds of hours). This, combined with a large number of channels (several hundreds), produces a substantial volume of data that needs to be processed. Also, because calibration is a non-linear and non-convex problem, iterative and alternating approaches usually form the basis for a practical solution [12], [13].

In this paper we use the Khatri–Rao structure of the matrices involved in data model to develop a computationally efficient direction independent gain calibration algorithm. We then use this method as a building block for a direction dependent calibration algorithm. Additionally, for very large problems we propose a conjugate gradient based algorithm and use simulation to evaluate the performance of these methods.

II. Data Model

In this section we introduce the covariance model for the data. We assume to have access to P (single polarization or unpolarized) receivers which are exposed to N_s (compact/point) sources. We assume that sources can be grouped into Q clusters which are affected by the same direction-dependent gain similar to the model presented in [6]. We stack the voltage output of each receiver in a vector denoted by $\mathbf{y}$ and assume that narrow-band assumptions hold. This allows us to model the sampled output of the array as

$$\mathbf{y}[n] = \sum_{q=1}^{Q} \mathbf{G}_q \mathbf{s}_q[n] + \mathbf{n}[n]$$

where $\mathbf{s}_q[n]$ represents the total signal from the qth cluster which includes the array response, $\mathbf{G}_q = \mathrm{diag}(\mathbf{g}_q)$ is the common gain for the qth cluster and $\mathbf{n}[n]$ is the noise of each receiver. The covariance matrix for this model is given by $\mathcal{E}\{\mathbf{y}\mathbf{y}^H\}$. However, we assume that some of the elements of this matrix are contaminated and/or are removed. We use a masking matrix $\mathbf{M}$ containing zeros and ones to capture this missing data in the model. We also assume that the gains are stable over several "snapshots" in both time and frequency. We assume to have K frequency channels with T snapshots each. Including the masking matrix we get the following covariance model for each snapshot

$$\mathbf{R}_{t,k} = \mathbf{M} \odot \mathcal{E}\{\mathbf{y}_{t,k}\mathbf{y}_{t,k}^H\} = \mathbf{M} \odot \sum_{q=1}^{Q} \mathbf{G}_q \boldsymbol{\Sigma}_{q,t,k} \mathbf{G}_q^H, \quad (1)$$

where $t = 1, \ldots, T$, $k = 1, \ldots, K$, $\odot$ is the element-wise or Hadamard product, ^{H} is the Hermitian transpose and $\boldsymbol{\Sigma}_{q,t,k} = \mathcal{E}\{\mathbf{s}_{q,t,k}\mathbf{s}_{q,t,k}^H\}$ is the covariance of the qth cluster or the "predicted sky-model", which is assumed to be known. We also assume that $\mathbf{R}_\mathbf{n} = \mathcal{E}\{\mathbf{n}\mathbf{n}^H\}$ is diagonal and is always removed as a result of applying the mask matrix, $\mathbf{M}$.

During the measurements, a noisy estimate of $\mathbf{R}_{t,k}$ is made using the output of the receivers. This estimate is denoted as a sample covariance matrix or sampled visibilities and is given

1 A. Mouri Sardarabadi (ammsa@astro.rug.nl) and L.V.E. Koopmans are affiliated with Kapteyn Astronomical Institute, University of Groningen, The Netherlands. 2 Alle-Jan van der Veen is affiliated with Delft University of Technology, Delft, The Netherlands

by

$$\hat{\mathbf{R}}_{t,k} = \mathbf{M} \odot \sum_{n=1}^{N} \mathbf{y}_{t,k}[n]\mathbf{y}_{t,k}[n]^H, \tag{2}$$

where N is the number of (voltage) samples in a single snapshot.

For the rest of this paper we stack the covariance (visibility) model and the data, respectively, into vectors

$$\mathbf{r}(\boldsymbol{\theta}) = \begin{bmatrix} \text{vect}(\mathbf{R}_{1,1}) \\ \text{vect}(\mathbf{R}_{2,1}) \\ \vdots \\ \text{vect}(\mathbf{R}_{T,K}) \end{bmatrix}, \hat{\mathbf{r}} = \begin{bmatrix} \text{vect}(\hat{\mathbf{R}}_{1,1}) \\ \text{vect}(\hat{\mathbf{R}}_{2,1}) \\ \vdots \\ \text{vect}(\hat{\mathbf{R}}_{T,K}) \end{bmatrix} \tag{3}$$

where vect(.) produces a vector from the argument matrix by stacking its columns and

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_1^H & \cdots & \mathbf{g}_Q^T & \mathbf{g}_Q^H \end{bmatrix}^T$$

is the "augmented" vector of variables. The term augmented means that a complex variable and its conjugate are used as separate variables instead of the real and imaginary part of the complex variables.

Using this data model we want to estimate the gains for each direction.

III. Direction Independent Algorithm

In this section we discuss the case where $Q = 1$. In this case the entire available sky-model is used and as a result the gain solutions are assumed direction independent. This case forms the basis for the direction-dependent calibration, which is discussed in the next section.

We use the least squares cost function to find an estimate for the gains:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})\|_2^2 \tag{4}$$

where $\|.\|_2$ is the l_2 norm of a vector. Because of the non-linear and non-convex nature of this problem we use a Newton-based iterative method known as the Gauss-Newton algorithm. The updates for this algorithm are given by

$$\hat{\boldsymbol{\theta}}^{(i+1)} = \hat{\boldsymbol{\theta}}^{(i)} + \mu^{(i)}\boldsymbol{\delta} \tag{5}$$

where the GN direction of descent $\boldsymbol{\delta}$ is given by the solution of [5]

$$\mathbf{J}^H\mathbf{J}\boldsymbol{\delta} = \mathbf{J}^H\left[\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})\right] \tag{6}$$

where

$$\mathbf{J} = \frac{\partial\mathbf{r}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^T} = \begin{bmatrix} \mathbf{J}_{1,1}^T & \cdots & \mathbf{J}_{T,K}^T \end{bmatrix}^T, \tag{7}$$

$$\mathbf{J}_{t,k} = \mathbf{P}\begin{bmatrix} \mathbf{G}^*\boldsymbol{\Sigma}_{t,k}^T \circ \mathbf{I}_P & \mathbf{I}_P \circ \mathbf{G}\boldsymbol{\Sigma}_{t,k} \end{bmatrix}, \tag{8}$$

with $\circ$ the Khatri-Rao product, * the complex conjugate and $\mathbf{P} = \text{diag}(\text{vect}(\mathbf{M}))$ a projection matrix corresponding to the mask matrix $\mathbf{M}$. There exists a phase ambiguity for the solutions, i.e. if $\mathbf{g}$ is a solution so is $\mathbf{g}' = e^{i\phi}\mathbf{g}$ for any real ϕ. We call the problem identifiable if $\text{rank}(\mathbf{J}) = 2P - 1$ where the deficiency by 1 is the result of the phase ambiguity. In this case a basis for the null space of $\mathbf{J}$ is given by

$$\mathbf{z} = [\mathbf{g}^T, -\mathbf{g}^H]^T. \tag{9}$$

Because $\mathbf{r}(\boldsymbol{\theta}) = 1/2\mathbf{J}\boldsymbol{\theta}$ we have $\mathbf{J}^H\mathbf{J}(\boldsymbol{\delta} + 1/2\boldsymbol{\theta}) = \mathbf{J}^H\hat{\mathbf{r}}$ which combined with the fact that $\boldsymbol{\theta}^H\mathbf{z} = \mathbf{0}$ and hence $\boldsymbol{\theta}$ is in the row space of $\mathbf{J}$, leads to

$$\hat{\boldsymbol{\theta}}^{(i+1)} = \left(1 - \frac{\mu^{(i)}}{2}\right)\hat{\boldsymbol{\theta}}^{(i)} + \mu^{(i)}\tilde{\boldsymbol{\delta}} \tag{10}$$

which is equivalent to (5) for $\tilde{\boldsymbol{\delta}}$ satisfying

$$\mathbf{J}^H\mathbf{J}\tilde{\boldsymbol{\delta}} = \mathbf{J}^H\hat{\mathbf{r}}. \tag{11}$$

With this change of variables for the direction of descent, we remove the necessity to update the model, $\mathbf{r}(\boldsymbol{\theta})$. However, since $\mathbf{J}$ depends on $\boldsymbol{\theta}$, this is only beneficial if we can calculate operations involving $\mathbf{J}$ and $\mathbf{J}^H$ sufficiently fast. Calculating the models $\boldsymbol{\Sigma}_{t,k}$ which are needed for calculating $\mathbf{J}$ is very expensive and we would like to pre-calculate these matrices only once. However, because TK is large, storing all of these model matrices should also be avoided. The rest of this section focuses on solving (11), while avoiding storage of the sky-models $\boldsymbol{\Sigma}_{t,k}$.

For square matrices $\mathbf{A}$ and $\mathbf{B}$ we have $\mathbf{I} \circ (\mathbf{A} \odot \mathbf{B}) = \text{diag}(\text{vect}(\mathbf{B}))(\mathbf{I} \circ \mathbf{A})$ and $(\mathbf{B}^T \odot \mathbf{A}) \circ \mathbf{I} = \text{diag}(\text{vect}(\mathbf{B}))(\mathbf{A} \circ \mathbf{I})$. Using these relations we have

$$\mathbf{J}_{t,k} = \begin{bmatrix} \mathbf{G}^*(\mathbf{M} \odot \boldsymbol{\Sigma}_{t,k})^T \circ \mathbf{I}_P & \mathbf{I}_P \circ \mathbf{G}(\mathbf{M} \odot \boldsymbol{\Sigma}_{t,k}) \end{bmatrix}.$$

Combining these results with $\mathbf{J}^H\mathbf{J} = \sum_k\sum_t \mathbf{J}_{t,k}^H\mathbf{J}_{t,k}$ and $\mathbf{J}^H\hat{\mathbf{r}} = \sum_k\sum_t \mathbf{J}_{t,k}^H\hat{\mathbf{r}}_{t,k}$ we have

$$\mathbf{J}^H\mathbf{J} = \begin{bmatrix} \text{diag}\left[\mathbf{H}\ (\mathbf{g} \odot \mathbf{g}^*)\right] & \mathbf{G}\mathbf{H}\mathbf{G} \\ \mathbf{G}^*\mathbf{H}\mathbf{G}^* & \text{diag}\left[\mathbf{H}\ (\mathbf{g} \odot \mathbf{g}^*)\right] \end{bmatrix} \tag{12}$$

and

$$\mathbf{J}^H\hat{\mathbf{r}} = \begin{bmatrix} \mathbf{E}\mathbf{g} \\ \mathbf{E}^*\mathbf{g}^* \end{bmatrix} \tag{13}$$

where

$$\mathbf{H} = \mathbf{M} \odot \sum_k\sum_t \boldsymbol{\Sigma}_{t,k}^T \odot \boldsymbol{\Sigma}_{t,k}, \tag{14}$$

$$\mathbf{E} = \mathbf{M} \odot \sum_k\sum_t \boldsymbol{\Sigma}_{t,k}^T \odot \hat{\mathbf{R}}_{t,k}. \tag{15}$$

We only need to calculate the real symmetric matrix $\mathbf{H}$ and the Hermitian matrix $\mathbf{E}$ once in order to solve $\tilde{\boldsymbol{\delta}}$ and $\hat{\boldsymbol{\theta}}$. This means that we can discard $\boldsymbol{\Sigma}_{t,k}$ during the calculation of $\mathbf{H}$ and $\mathbf{E}$. This allows for a dramatic reduction of the required storage and also I/O overhead during the calibration.

The remaining problem is the actual solution of (11) which we address now. We would like to point out that this system of equations is normal and consistent. This allows for the solution to be obtained from

$$\tilde{\boldsymbol{\delta}} = \mathbf{X}\mathbf{J}^H\hat{\mathbf{r}},$$

where $\mathbf{X}$ is any generalized inverse of $\mathbf{J}^H\mathbf{J}$ (i.e. $\mathbf{J}^H\mathbf{J}\mathbf{X}\mathbf{J}^H\mathbf{J} = \mathbf{J}^H\mathbf{J}$). However, not all $\tilde{\boldsymbol{\delta}}$ found in this way will have the augmented form $[\mathbf{y}^T, \mathbf{y}^H]^H$, which is required for a valid direction of descent. We use the following lemma to find a simple solution for this problem.

Lemma 1. *Let* $\mathbf{K}$ *be a permutation matrix of the form*

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_M \\ \mathbf{I}_M & \mathbf{0} \end{bmatrix},$$

and $\mathbf{A}$ *be any square matrix of size* $2M \times 2M$ *such that*

ISBN 978-90-827970-1-5 © EURASIP 2018

$\mathbf{A}^* = \mathbf{KAK}$. *Let* $\mathbf{A}^g$ *be a generalized inverse of* $\mathbf{A}$ *(i.e.* $\mathbf{AA}^g\mathbf{A} = \mathbf{A}$*) then* $\mathbf{X} = \frac{1}{2}(\mathbf{A}^g + \mathbf{K}(\mathbf{A}^g)^*\mathbf{K})$ *is also a generalized inverse of* $\mathbf{A}$.

Proof. The proof is a simple verification:

$$\begin{aligned}\mathbf{AXA} &= \tfrac{1}{2}(\mathbf{AA}^g\mathbf{A}) + \tfrac{1}{2}(\mathbf{AK}(\mathbf{A}^g)^*\mathbf{KA}) \\ &= \tfrac{1}{2}\mathbf{A} + \tfrac{1}{2}(\mathbf{KKAK}(\mathbf{A}^g)^*\mathbf{KAKK}) \\ &= \tfrac{1}{2}\mathbf{A} + \tfrac{1}{2}(\mathbf{K}(\mathbf{AA}^g\mathbf{A})^*\mathbf{K}) = \mathbf{A},\end{aligned}$$

where we used $\mathbf{KK} = \mathbf{I}$. □

It is trivial to verify that $\mathbf{KJ}^H\mathbf{JK} = (\mathbf{J}^H\mathbf{J})^*$ and $\mathbf{KJ}^H\hat{\mathbf{r}} = (\mathbf{J}^H\hat{\mathbf{r}})^*$. This allows us to show that for any generalized inverse solution $\tilde{\boldsymbol{\delta}}_1 = (\mathbf{J}^H\mathbf{J})^g\mathbf{J}^H\hat{\mathbf{r}}$,

$$\tilde{\boldsymbol{\delta}} = \mathbf{XJ}^H\hat{\mathbf{r}} = \frac{1}{2}(\tilde{\boldsymbol{\delta}}_1 + \mathbf{K}\tilde{\boldsymbol{\delta}}_1^*) \tag{16}$$

is a solution to the system of equation with the correct format.

Based on this discussion, it is always possible to transform any solution to the correct (augmented) format. This gives us more flexibility in choosing our solver. For the matrix $\mathbf{J}^H\mathbf{J}$ we will show that using a block LDU decomposition will lead to solving a system of equations which involves a convergent matrix which has a stable and fast iterative solution [3].

In order to simplify the notation we introduce the following definitions: $\tilde{\mathbf{g}} \equiv \mathbf{g}^* \odot \mathbf{g}$, $\mathbf{D} \equiv \mathrm{diag}(\mathbf{H}\tilde{\mathbf{g}})$, $\mathbf{b} \equiv \mathbf{D}^{-1/2}\mathbf{Eg}$ and $\mathbf{C} \equiv \mathbf{D}^{-1/2}\mathbf{G}^*\mathbf{HG}^*\mathbf{D}^{-1/2}$. With these definitions the block-LDU decomposition of $\mathbf{J}^H\mathbf{J} = \mathbf{L}\tilde{\mathbf{D}}\mathbf{L}^H$ is given by

$$\mathbf{L} = \begin{bmatrix}\mathbf{D}^{1/2} & \mathbf{0} \\ \mathbf{D}^{1/2}\mathbf{C} & \mathbf{D}^{1/2}\end{bmatrix} \quad \text{and } \tilde{\mathbf{D}} = \begin{bmatrix}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \mathbf{CC}^H\end{bmatrix}.$$

Applying forward-backward substitution we find the following expression for $\tilde{\boldsymbol{\delta}}_1$ in (16):

$$\tilde{\boldsymbol{\delta}}_1 = \begin{bmatrix}\mathbf{D}^{-1/2}(\mathbf{b} - \mathbf{C}^H\tilde{\boldsymbol{\delta}}_{1,2}) \\ \mathbf{D}^{-1/2}\tilde{\boldsymbol{\delta}}_{1,2}\end{bmatrix}, \tag{17}$$

where $\boldsymbol{\delta}_{1,2}$ is the solution to the following system of equations

$$(\mathbf{I} - \mathbf{CC}^H)\tilde{\boldsymbol{\delta}}_{1,2} = \mathbf{b}^* - \mathbf{Cb}. \tag{18}$$

Remembering that $\mathbf{J}^H\mathbf{J}$ is rank-deficient by one and the fact that $\mathbf{L}$ is positive definite, we know that $\mathbf{I} - \mathbf{CC}^H$ is also rank deficient by one and positive semidefinite. We already discussed that $\mathbf{z}$ given by (9) is a basis for the null space of $\mathbf{J}^H\mathbf{J}$. This means that $\mathbf{L}^H\mathbf{z}$ is a basis for the null space of $\tilde{\mathbf{D}}$ and hence

$$\tilde{\mathbf{z}} = \frac{1}{\sqrt{\mathbf{g}^H\mathbf{Dg}}}\mathbf{D}^{1/2}\mathbf{g}^*$$

is a unit-norm basis for the null space of $\mathbf{I} - \mathbf{CC}^H$. Because the system of equations in (11) is consistent, so is (18) and

$$\begin{aligned}\tilde{\boldsymbol{\delta}}_{1,2} &= (\mathbf{I} - \mathbf{CC}^H)^\dagger(\mathbf{b}^* - \mathbf{Cb}) \\ &= (\mathbf{I} - \mathbf{CC}^H + \mathbf{zz}^H)^{-1}(\mathbf{b}^* - \mathbf{Cb}).\end{aligned}$$

Note that $\mathbf{I} - \mathbf{CC}^H + \tilde{\mathbf{z}}\tilde{\mathbf{z}}^H$ is positive definite with $\lambda_{\max} = 1$, which means that the spectral radius of $\rho(\mathbf{zz}^H - \mathbf{CC}^H) < 1$ and hence this matrix is a convergent matrix. For convergent matrices we know [3] that

$$\tilde{\boldsymbol{\delta}}_{1,2}^{(j+1)} = \mathbf{b}^* - \mathbf{Cb} - (\mathbf{zz}^H - \mathbf{CC}^H)\tilde{\boldsymbol{\delta}}_{1,2}^{(j)} \tag{19}$$

will converge to a solution of (18).

To summarize, in order to find a solution to (11), first we need to calculate $\mathbf{D}$, $\mathbf{C}$, $\tilde{\mathbf{z}}$ and then use (19), (17) and (16). The complexity of these operations are P divisions and $O(P^2)$ operations needed for the matrix vector multiplications. This means that we will benefit from the fast convergence of the GN algorithm, while having the same complexity as slower converging alternating algorithms.

The only unsolved issue is the optimal step-size $\mu^{(i)}$ which, as we show in Appendix A, requires solving for the roots of a third order polynomial with real coefficients, for which closed-form solutions exists.

IV. Extension to Direction Dependent Calibration

Now that several key ideas have been derived for the direction independent scenario, we extend to the direction-dependent case. Again we use the least squares cost function to find an estimate for the gains

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{r} - \sum_{q=1}^{Q}\mathbf{r}_q(\boldsymbol{\theta}_q)\|_2^2$$

where

$$\mathbf{r}_q(\boldsymbol{\theta}_q) = \begin{bmatrix}\mathrm{vect}(\mathbf{G}_q\boldsymbol{\Sigma}_{q,1,1}\mathbf{G}_q) \\ \vdots \\ \mathrm{vect}(\mathbf{G}_q\boldsymbol{\Sigma}_{q,T,K}\mathbf{G}_q)\end{bmatrix}.$$

Using this cost function we discuss two different approaches for solving this problem. The first one is based on the repeated application of the method developed for the direction independent scenario which we will denote as "Block Gauss-Newton" (BGN) and the second approach which is based on the Conjugate Gradient (CG) method.

A. Block Gauss-Newton

We can extend the matrices $\mathbf{H}$ and $\mathbf{E}$ defined by (14) and (15) to the direction dependent case as

$$\mathbf{E}_q = \sum_t\sum_k \boldsymbol{\Sigma}_{q,t,k}^T \odot \hat{\mathbf{R}}_{t,k} \tag{20}$$

$$\mathbf{H}_{q_1,q_2} = \sum_t\sum_k \boldsymbol{\Sigma}_{q_1,t,k}^T \odot \boldsymbol{\Sigma}_{q_2,t,k} \tag{21}$$

where q, q_1 and q_2 take values $1,\ldots,Q$, $\mathbf{H}_{q_1,q_2}$ is Hermitian and $\mathbf{H}_{q_1,q_2} = \mathbf{H}_{q_2,q_1}^T$. The use of these matrices is beneficial only if $KT/Q > 1$. If this condition does not hold, storing $\hat{\mathbf{R}}_{t,k}$ and $\boldsymbol{\Sigma}_{q,t,k}$ will be more efficient than generating $\mathbf{E}_q$ and $\mathbf{H}_{q_1,q_2}$. We assume that this condition holds for a practical calibration scenario.

Using $\mathbf{E}_q$ and $\mathbf{H}_{q_1,q_2}$, the gradient for the qth direction can be written as

$$\gamma_q = \mathbf{J}_q^H(\hat{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})) = \begin{bmatrix}\mathbf{E}_q\mathbf{g}_q - \sum_{q_2}\mathbf{G}_{q_2}\mathbf{H}_{q,q_2}(\mathbf{g}_q \odot \mathbf{g}_{q_2}^*) \\ \mathbf{E}_q^T\mathbf{g}_q^* - \sum_{q_2}\mathbf{G}_{q_2}^*\mathbf{H}_{q,q_2}^T(\mathbf{g}_q^* \odot \mathbf{g}_{q_2})\end{bmatrix}. \tag{22}$$

If we change the summation above such that $q_2 \neq q$, then the direction of descent can be found by applying the method discussed in the previous section separately for each direction in a parallelized fashion. Because the updates are done separately for each direction, we are not limited to a single iteration and we can update each solution several time before updating the gradients. This approach is very similar to the ADMM [2]. However, the calibration problem is not convex and the convergence of BGN is local.

B. Conjugate Gradient

For the next generation radio telescopes, such as the SKA, the number of stations and directions will increase dramatically. In these cases where the problem becomes very large even the modified Gauss-Newton method used in previous section could become prohibitive. Simple classical methods such as Conjugate Gradient (CG) become attractive in these scenarios. The CG has very nice convergence properties if the exact optimal step-size is used [5]. If $KT/Q > 1$ we can use the matrices $\mathbf{E}_q$ and $\mathbf{H}_{q_1,q_2}$ to find the optimal step-size (see Appendix A). This, in combination with the Polak-Ribière method [5] will produce a relatively fast converging CG method for the direction-dependent calibration.

For this algorithm we use the previous direction of descent and the gradient given by (22) to a new direction of descent. The updates for the direction of descent are

$$\boldsymbol{\delta}^{(j)} = \boldsymbol{\gamma}^{(j)} + \lambda \boldsymbol{\delta}^{(j-1)}$$

where $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^T, \ldots, \boldsymbol{\gamma}_Q^T]$ and λ is given by Polak-Ribière ratio

$$\lambda = \frac{\Re\{(\boldsymbol{\gamma}^{(j)} - \boldsymbol{\gamma}^{(j-1)})^H \boldsymbol{\gamma}^{(j)}\}}{{\boldsymbol{\gamma}^{(j-1)}}^H \boldsymbol{\gamma}^{(j-1)}},$$

where $\Re\{.\}$ is the real part of the argument. Using simulations we show that this algorithm is computationally competitive with other methods.

V. Simulations

A. Direction Independent Calibration

In this section we use simulations to evaluate the performance of the proposed direction independent calibration technique. We simulate sample covariance data (visibilities) using the array configuration of the LOFAR radio telescope consisting only of the Dutch stations [11] with $P = 62$. For the sky model we use the North Celestial Pole (NCP)[1]. We use 5000 strongest component (point sources) in this field to generate both the data and construct the predicted sky model (i.e. $\mathbf{\Sigma}_{t,k}$). We divide a typical LOFAR channel with 195.3 kHz bandwidth into $K = 3$ sub channels of $\approx$ 65kHz around the central frequency of 150 MHz. For each channel we generate $T = 600$ snapshots, each with an integration time of 1 second, which translates into $N = 2 \times 65 \times 10^3$ samples with Nyquist sampling. We repeat this for a total of 9 observations which are separated by 1 hour from each other. This is done in order to have enough rotation of the Earth to synthesize an image. Fig. 1 show an MVDR dirty image of the simulated field using 10 snapshot from each hour. Table I summarizes the computation on an Intel 7i-6700K CPU with 16GB of RAM. As we see, generating the predicted model $\mathbf{\Sigma}_{t,k}$ is the most expensive part of the problem which cannot be avoided and is common among all currently available calibration models which use a sky model. By using $\mathbf{H}$ and $\mathbf{E}$ for the direction independent calibration we reduce the storage during the calibration by a factor of $TK = 1800$ and as is shown in fig. 2 and Table I, the algorithm converges very fast both in number of iterations and in computing time.

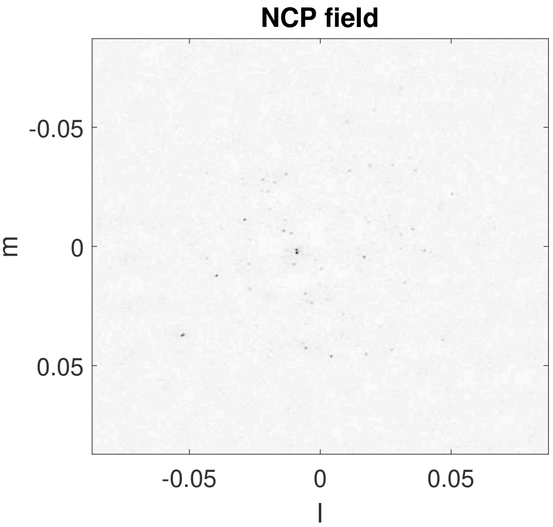

Fig. 1: The MVDR dirty image of the simulated NCP field.

TABLE I: Computation time DIC

Generating $\mathbf{\Sigma}_{t,k}$	Calculating $\mathbf{H}$ and $\mathbf{E}$	optimization
25s	0.1s	0.006s

B. Direction Dependent Calibration

For the direction dependent calibration we use again $P = 62$ receivers with 20 randomly generated sources per direction. We use a single snapshot with $N = 10^4$ samples, which is moderate for radio astronomical observations. We then use the algorithm discussed in Sec. IV to find the gains.

Fig. 3 shows the convergence speed of both algorithms based on the gradient. As expected the Block-Gauss-Newton

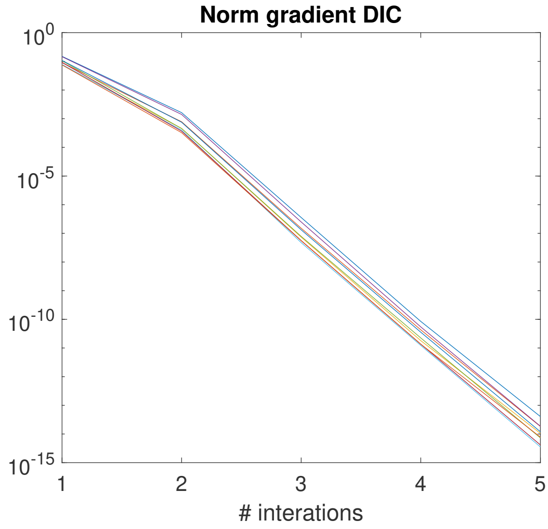

Fig. 2: Convergence of the direction independent calibration for 9 calibration runs

[1] We would like to thank Sarod Yatawatta for this sky model.

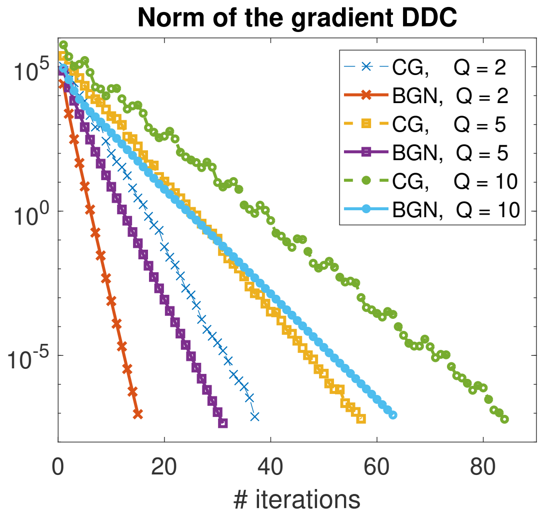

Fig. 3: Convergence of the direction-dependent calibration

(BGN) converges faster than the CG method. However, based on several repetition of our simulations, we have observed that the total computation time of the CG method, especially for larger Q, is much lower. For example in the case where $Q = 10$ BGN method takes $\approx$ 7s while CG take ≈ 0.3s. This fast convergence of CG is mainly because of the exact step-size calculations.

VI. Conclusion

In this paper we have proposed new calibration algorithms for both direction dependent and independent calibration for radio interferometric array. We have shown that the optimal step-size can be calculated in a closed form fashion and does not require (expensive) line-search methods or approximations. All of the proposed algorithms converge reasonably fast and have very small storage requirements.

There are several issues that are not addressed in this paper, including polarization, ionospheric effects and frequency dependency of the gains. The latter places additional restriction on the gains and hence will only improve the proposed algorithm without much anticipated modifications. It can be shown that if the sky model consist of un-polarized sources, the polarized direction independent gain calibration can be formulated as an un-polarized direction-dependent calibration with two directions. These extensions will be addressed in future works.

References

[1] A.-J. Boonstra and A.-J. van der Veen, "Gain calibration methods for radio telescope arrays," *IEEE Tr. Signal Processing*, vol. 51, no. 1, pp. 25–38, Jan. 2003. [Online]. Available: ftp://cas.et.tudelft.nl/pub/allejan-docs/gaincal.pdf

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[3] J. Carl D. Meyer and R. J. Plemmons, "Convergent powers of a matrix with applications to iterative methods for singular linear systems," *SIAM Journal on Numerical Analysis*, vol. 14, no. 4, pp. 699–705, 1977. [Online]. Available: https://doi.org/10.1137/0714047

[4] P. E. Dewdney, P. J. Hall, R. T. Schilizzi, and T. J. L. Lazio, "The Square Kilometre Array," *Proceedings of the IEEE*, vol. 97, no. 8, pp. 1482–1496, 2009.

[5] P. Frandsen, K. Jonasson, H. Nielsen, and O. Tingleff, *Unconstrained Optimization, 3rd edition*, 2004.

[6] S. Kazemi, S. Yatawatta, and S. Zaroubi, "Radio interferometric calibration via ordered-subsets algorithms: OS–LS and OS–SAGE calibrations," *Monthly Notices of the Royal Astronomical Society*, vol. 434, no. 4, pp. 3130–3141, 2013.

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

[8] O. Smirnov and C. Tasse, "Radio interferometric gain calibration as a complex optimization problem," *Monthly Notices of the Royal Astronomical Society*, vol. 449, no. 3, pp. 2668–2684, 2015.

[9] S. van der Tol, B. Jeffs, and A.-J. van der Veen, "Self-Calibration for the LOFAR Radio Astronomical Array," *Signal Processing, IEEE Transactions on*, vol. 55, no. 9, pp. 4497–4510, Sept 2007.

[10] A.-J. van der Veen, A. Leshem, and A.-J. Boonstra, "Array signal processing for radio astronomy," in *The Square Kilometre Array: An Engineering Perspective*, P. Hall, Ed. Dordrecht: Springer, 2005, pp. 231–249, iSBN 1-4020-3797-x. Reprinted from Experimental Astronomy, 17(1-3),2004. [Online]. Available: http://ens.ewi.tudelft.nl/pubs/aj04ska.pdf

[11] M. P. van Haarlem, M. W. Wise, A. W. Gunst *et al.*, "LOFAR: The LOw-Frequency ARray," *Astronomy & Applications*, vol. 556, p. A2, 2013. [Online]. Available: http://dx.doi.org/10.1051/0004-6361/201220873

[12] S. Wijnholds and A.-J. van der Veen, "Multisource self-calibration for sensor arrays," *Signal Processing, IEEE Transactions on*, vol. 57, no. 9, pp. 3512–3522, Sept 2009.

[13] S. Yatawatta, "Distributed radio interferometric calibration," *Monthly Notices of the Royal Astronomical Society*, vol. 449, no. 4, pp. 4506–4514, 2015.

Appendix A
Step Size

The least square costs function in Sec. IV can be written as

$$\begin{aligned} f(\boldsymbol{\theta}) &= \hat{\mathbf{r}}^H\hat{\mathbf{r}} - 2\sum_q \mathbf{g}_q^H\mathbf{E}_q\mathbf{g}_q + \sum_q \tilde{\mathbf{g}}_q^H\mathbf{H}_{q,q}\tilde{\mathbf{g}}_q^H \\ &+2\sum_{q_1}\sum_{q2>q_1} \tilde{\mathbf{g}}_{q_1,q_2}^H\mathbf{H}_{q_1,q_2}\tilde{\mathbf{g}}_{q_1,q_2} \end{aligned}$$

where $\tilde{\mathbf{g}}_q = \mathbf{g}_q \odot \mathbf{g}_q^*$ and $\tilde{\mathbf{g}}_{q_1,q_2} = \mathbf{g}_{q_1} \odot \mathbf{g}_{q_2}^*$. Let the direction of descent be $\boldsymbol{\delta}$ and its sub-vector for qth direction be $\boldsymbol{\delta}_q$, we are interested in

$$\mu_{\text{opt}} = \arg\min_{\mu} f(\boldsymbol{\theta} + \mu\boldsymbol{\delta}).$$

Because of the quadratic relations in $\tilde{\mathbf{g}}$ we know that the cost function is a fourth order polynomial in μ with real coefficients, which means that finding optimal μ requires solving the roots of a third order polynomial. The gradient of the cost function with respect to μ is given by

$$f'(\mu) = 4c_1\mu^3 + 3c_2\mu^2 + 2c_3\mu + c_1$$

where $c_i = a_i + b_i$ with

$$\begin{aligned} a_4 &= \sum_q 2\tilde{\mathbf{y}}_q^H\mathbf{H}_{q,q}\tilde{\mathbf{g}}_q - 4\Re\{\boldsymbol{\delta}_q^H\mathbf{E}_q\mathbf{g}_q\}, \\ a_3 &= \sum_q 2\tilde{\mathbf{g}}_q^H\mathbf{H}_{q,q}\tilde{\mathbf{x}}_q + \tilde{\mathbf{y}}_q^H\mathbf{H}_{q,q}\tilde{\mathbf{y}}_q - 2\Re\{\boldsymbol{\delta}_q^H\mathbf{E}_q\boldsymbol{\delta}_q\}, \\ a_2 &= \sum_q 2\tilde{\mathbf{y}}_q^H\mathbf{H}_{q,q}\tilde{\mathbf{x}}_q, \\ a_1 &= \sum_q \tilde{\mathbf{x}}_q^H\mathbf{H}_{q,q}\tilde{\mathbf{x}}, \\ b_4 &= \sum_q\sum_{p>q} 4\Re\{\tilde{\mathbf{y}}_{q,p}^H\mathbf{H}_{q,p}\tilde{\mathbf{g}}_{q,p}\}, \\ b_3 &= \sum_q\sum_{p>q} 2\Re\{\tilde{\mathbf{y}}_{q,p}^H\mathbf{H}_{q,p}\tilde{\mathbf{y}}_{q,p} + 2\tilde{\mathbf{g}}_{q,p}^H\mathbf{H}_{q,p}\tilde{\mathbf{x}}_{q,p}\}, \\ b_2 &= \sum_q\sum_{p>q} 4\Re\{\tilde{\mathbf{y}}_{q,p}^H\mathbf{H}_{q,p}\tilde{\mathbf{x}}q,p\}, \\ b_1 &= \sum_q\sum_{p>q} 2\Re\{\tilde{\mathbf{x}}_{q,p}^H\mathbf{H}_{q,p}\tilde{\mathbf{x}}_{q,p}\}, \\ \tilde{\mathbf{y}}_{q,p} &= \mathbf{g}_q \odot \boldsymbol{\delta}_p^* + \boldsymbol{\delta}_q \odot \mathbf{g}_p^*, \\ \tilde{\mathbf{x}}_{q,p} &= \boldsymbol{\delta}_q \odot \boldsymbol{\delta}_p^*, \end{aligned}$$

$\tilde{\mathbf{y}}_q = \tilde{\mathbf{y}}_{q,q}$ and $\tilde{\mathbf{x}}_q = \tilde{\mathbf{x}}_{q,q}$.

ISBN 978-90-827970-1-5 © EURASIP 2018