Application of Krylov Based Methods in Calibration for Radio Astronomy

Ahmad Mouri Sardarabadi†
Delft University of Technology

Alle-Jan van der Veen
Delft University of Technology

***Abstract*—As the number of antennas in the modern radio–telescopes increases, the computational complexity of the calibration algorithms becomes more and more important. In this paper we use the Khatri–Rao structure of the covariance data model used for such calibrations and combine it with Krylov subspace based methods to achieve accurate calibration results with low complexity, very small memory usage and fast convergence properties. We also demonstrate the proposed method on experimental data measured by the LOFAR radio–telescope.**

***Index Terms*—Krylov subspace, calibration, radio astronomy, minresQLP, optimization**

I. Introduction

One of the challenges with current and future radio–telescopes, like LOFAR (LOw Frequency ARray) [1] and SKA (Square Kilometer Array), is the accurate calibration of the instrument with reasonable computational complexity. These modern radio–telescopes consist of an array of antennas which measures the signals coming from radio–sources in the sky [2]. Some of these sources are well known from other studies and can serve as calibration sources. However because the array might have direction dependent behavior [3] we will only assume that the spatial coordinates of these sources are known. We are interested in estimating the complex gain of each antenna element along with the source and noise powers. Further we assume that the position of each antenna is known but we do not require any particular geometry for the array.

The model, as will be presented in the next section, is non–linear and the number of unknowns grows with the number of elements in the array and calibration sources. This means that for very large arrays we are dealing with a complex optimization problem. The authors in [2] and [3] overcome this problem by splitting the unknowns into groups for which a closed formed solution could be found. Then by applying alternating optimization algorithms such as alternating least squares (ALS) or alternating weighted least squares (WALS), a monotonic convergence to the solution could be achieved. Also in a more general case where the polarization is also taken into account, like the case studied by [4], in order to reduce the computation costs the unknowns are split into smaller sub–sets and each set is updated in an alternating fashion.

In this paper we will show that the matrices used during the estimation process possess a strong Khatri–Rao structure which, combined with Krylov subspace based methods, like minresQLP [5], can be exploited to reduce the computational costs and achieve accurate results with low complexity and fast convergence rate, without using an alternating approach. Another advantage of the proposed method is a tremendous reduction in the memory usage which could be desired in some applications.

† This research was supported by NWO-TOP 2010, 614.00.005.

II. Data Model and Problem Definition

An array of p elements with known locations, $\mathbf{z}_m, m = 1, \ldots, p$, is exposed to q calibration sources with known spatial coordinates $\mathbf{k}_n, n = 1, \ldots, q$. The elements of the steering matrix, $\mathbf{A}$ consist of the geometric phase delays,

$$a_{m,n} = \frac{1}{\sqrt{p}} e^{j \frac{2\pi}{\lambda} \mathbf{z}_m^T \mathbf{k}_n}, \tag{1}$$

where λ is the wavelength, ^{T} is the transpose operator and we have assumed that narrowband assumption holds such that the delays translate into phase changes. Now we will stack the received signal from each antenna into a vector called $\mathbf{x}(t)$ and obtain the following the data model

$$\mathbf{x}(t) = \mathbf{G}\mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \tag{2}$$

where $\mathbf{G} = \text{diag}(\mathbf{g})$ is a diagonal matrix modeling the gain of each element, $\mathbf{s}$ is a $q \times 1$ vector representing the signal from calibration sources and $\mathbf{n}$ is a $p \times 1$ vector representing the noise in the system. This is a commonly used model for array calibration [3].

We assume that the noise and the sources are independent Gaussian processes which allows us to write the model for the covariance matrix of the array vector as

$$\mathbf{R} = \mathcal{E}\{\mathbf{x}\mathbf{x}^H\} = \mathbf{G}\mathbf{A}\mathbf{R}_s\mathbf{A}^H\mathbf{G}^H + \mathbf{R}_n, \tag{3}$$

where ^{H} is the Hermitian transpose, $\mathbf{R}_s = \text{diag}(\boldsymbol{\sigma})$ is the covariance matrix of the sources, $\boldsymbol{\sigma}$ is a $q \times 1$ vector which represent the power of each calibration source and $\mathbf{R}_n$ is the covariance matrix of the system and the sky noise.

Using the above model, we will now define the calibration problem. We want to estimate $\mathbf{g}$, $\boldsymbol{\sigma}$ and the unknown parameters in $\mathbf{R}_n$, when we have available to us a sample covariance matrix measured from N samples that is defined as

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_k \mathbf{x}_k \mathbf{x}_k^H. \tag{4}$$

This formulation of the problem leads to covariance matching techniques like the one described in [6]. The results of these studies are used in the next section.

978-1-4799-1481-4/14/$31.00 ©2014 IEEE

III. Algorithm

It is desirable to find statistically efficient estimates for the calibration problem. The Maximum Likelihood methods (ML) are in this case very popular. For ML the log–likelihood cost function, after taking N samples, could be given by

$$f_{ML}(\boldsymbol{\theta}) = N\left[-\log|\pi^p| + \log|\mathbf{R}^{-1}(\boldsymbol{\theta})| - \mathrm{tr}(\mathbf{R}^{-1}(\boldsymbol{\theta})\hat{\mathbf{R}})\right]. \tag{5}$$

where $\boldsymbol{\theta}$ is a vector containing all the unknowns which we will define shortly.

However when a large number of samples is available and a suitable weightings is applied optimal results could also be found using WLS [6]. In the case of WLS we have

$$f_{WLS}(\boldsymbol{\theta}) = \|\mathbf{W}^{1/2}(\hat{\mathbf{R}} - \mathbf{R}(\boldsymbol{\theta}))\mathbf{W}^{1/2}\|_F^2. \tag{6}$$

The WLS reduces to LS if $\mathbf{W} = \mathbf{I}_p$. To have a solution that approaches ML asymptotically we choose, $\mathbf{W} = \hat{\mathbf{R}}^{-1}$ [6]. One method for finding a $\boldsymbol{\theta}$ that minimizes/maximizes these functions is the descent algorithm where the solution is updated as

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \mu\boldsymbol{\Delta}^{(i)}, \tag{7}$$

where $\boldsymbol{\Delta}$ is the direction of descend and μ is the step size. At each iteration we need to find the direction of descend using the Jacobian of our cost functions. This means finding $\boldsymbol{\Delta}$ by solving the following system at each iteration

$$\mathbf{J}^H(\mathbf{W}^* \otimes \mathbf{W})\mathbf{J}\boldsymbol{\Delta} = \mathbf{J}^H(\mathbf{W}^* \otimes \mathbf{W})\,\mathrm{vect}\left[\hat{\mathbf{R}} - \mathbf{R}(\boldsymbol{\theta})\right], \tag{8}$$

where * is the complex conjugate, $\otimes$ is the Kronecker–product, $\mathbf{J} = \frac{\partial \mathrm{vect}(\mathbf{R})}{\partial \boldsymbol{\theta}^T}$ is the Jacobian and $\mathrm{vect}(\mathbf{X})$ stacks the columns of $\mathbf{X}$ into a single vector. For WLS this approach is equivalent to Gauss–Newton and in the case of ML where the weighting matrix, $\mathbf{W}^{(i)} = \mathbf{R}^{-1}(\boldsymbol{\theta}^{(i)})$, this approach is equivalent to the scoring method where

$$\mathbf{F} = \mathbf{J}^H\left(\mathbf{R}^{-T} \otimes \mathbf{R}^{-1}\right)\mathbf{J}, \tag{9}$$

is the Fisher information matrix [7].

In our application we can define $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \begin{bmatrix}\mathbf{g}^T & \mathbf{g}^H & \boldsymbol{\sigma}^T & \boldsymbol{\sigma}_n^T\end{bmatrix}^T, \tag{10}$$

where $\boldsymbol{\sigma}_n = \mathbf{S}^H\mathrm{vect}(\mathbf{R}_n)$ and $\mathbf{S}$ is a selection matrix. If $\mathbf{R}_n$ is assumed to be a diagonal matrix, then $\mathbf{S} = (\mathbf{I}_p \circ \mathbf{I}_p)$ and $\boldsymbol{\sigma}_n = \mathrm{vectdiag}(\mathbf{R}_n)$ where $\circ$ defines the Khatri–Rao product and $\mathrm{vectdiag}(\mathbf{X})$ stacks the diagonal elements of $\mathbf{X}$ into a vector.

When we partition $\boldsymbol{\theta}$ in this way we can also partition the Jacobian as

$$\mathbf{J} = \left[\mathbf{J}_g, \mathbf{J}_{g^*}, \mathbf{J}_{\boldsymbol{\sigma}}, \mathbf{J}_{\boldsymbol{\sigma}_n}\right], \tag{11}$$

where

$$\begin{aligned}\mathbf{J}_g &= \frac{\partial \mathrm{vect}(\mathbf{R})}{\partial \mathbf{g}^T} = \left(\mathbf{G}^*\mathbf{A}^*\mathbf{R}_s\mathbf{A}^T \otimes \mathbf{I}_p\right)(\mathbf{I}_p \circ \mathbf{I}_p)\\ &= \mathbf{G}^*\mathbf{A}^*\mathbf{R}_s\mathbf{A}^T \circ \mathbf{I}_p, \qquad (12)\\ \mathbf{J}_{g^*} &= \frac{\partial \mathrm{vect}(\mathbf{R})}{\partial \mathbf{g}^H} = \left(\mathbf{I}_p \otimes \mathbf{G}\mathbf{A}\mathbf{R}_s\mathbf{A}^H\right)(\mathbf{I}_p \circ \mathbf{I}_p)\\ &= \mathbf{I}_p \circ \mathbf{G}\mathbf{A}\mathbf{R}_s\mathbf{A}^H, \qquad (13)\\ \mathbf{J}_{\boldsymbol{\sigma}} &= \frac{\partial \mathrm{vect}(\mathbf{R})}{\partial \boldsymbol{\sigma}^T} = (\mathbf{G}^*\mathbf{A}^* \otimes \mathbf{G}\mathbf{A})(\mathbf{I}_p \circ \mathbf{I}_p)\\ &= \mathbf{G}^*\mathbf{A}^* \circ \mathbf{G}\mathbf{A}, \qquad (14)\\ \mathbf{J}_{\boldsymbol{\sigma}_n} &= \frac{\partial \mathrm{vect}(\mathbf{R})}{\partial \boldsymbol{\sigma}_n^T} = \mathbf{S}. \qquad (15)\end{aligned}$$

In order to solve (8) we define the matrix

$$\mathbf{H} = \mathbf{J}^H(\mathbf{W}^* \otimes \mathbf{W})\mathbf{J}, \tag{16}$$

and the gradient vector

$$\mathbf{b} = \mathbf{J}^H(\mathbf{W}^* \otimes \mathbf{W})\,\mathrm{vect}\left[\hat{\mathbf{R}} - \mathbf{R}(\boldsymbol{\theta})\right], \tag{17}$$

such that at each iteration we need to solve $\mathbf{H}\boldsymbol{\Delta} = \mathbf{b}$. The dimensions of $\mathbf{H}$ depend on the number of unknowns

$$n = 2p + q + \|\mathrm{vect}(\mathbf{S})\|_1 \tag{18}$$

and for a large array it could become a problem to store it in the memory. Also because we cannot assume any sparse, circular or Toeplitz structure in $\mathbf{H}$, except that it is Hermitian, solving this problem with a direct method has a cubic complexity which must be repeated at each iteration. However if we use a solver based on the Krylov subspace method we can use the Khatri–Rao structure of the Jacobian matrices. This will reduce the complexity and memory usage which is important for very large arrays like SKA.

Krylov Subspace Based Methods

The Krylov subspace based methods solve $\mathbf{H}\boldsymbol{\Delta} = \mathbf{b}$ by using matrix–vector products of the form $\mathbf{H}\mathbf{v}$ repeatedly. If this operation can be performed in an efficient way, then application of these methods are preferred to other methods. Especially because we can define a procedure that performs the matrix–vector product, the matrix $\mathbf{H}$ does not need to be stored in memory. This makes Krylov subspace based methods very suitable for situations where $\mathbf{H}$ is very large. We will now show how this matrix vector product can be performed in an efficient way. We have chosen minresQLP because it is capable of handling singular matrices which adds robustness during the iterations [5].

We will split the operation of $\mathbf{H}\mathbf{v}$ into three steps. First we will calculate two intermediate results

$$\mathbf{c} = \mathbf{J}\mathbf{v}, \tag{19}$$

and $\mathbf{c}_\mathbf{W} = (\mathbf{W}^* \otimes \mathbf{W})\mathbf{c}$. Using these intermediate results we then calculate the final result

$$\mathbf{H}\mathbf{v} = \mathbf{J}^H\mathbf{c}_\mathbf{W}. \tag{20}$$

Now we will show how each of these steps use the Khatri–Rao structure and can be done efficiently. In order to calculate $\mathbf{c}$ we partition $\mathbf{v}$, in the same way we have partitioned $\boldsymbol{\theta}$, as

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_{\mathbf{g}}^T & \mathbf{v}_{\mathbf{g}^*}^T & \mathbf{v}_{\boldsymbol{\sigma}}^T & \mathbf{v}_{\boldsymbol{\sigma}_n}^T \end{bmatrix}^T, \tag{21}$$

then we have

$$\mathbf{c} = \mathbf{J}\mathbf{v} = \mathbf{J}_{\mathbf{g}}\mathbf{v}_{\mathbf{g}} + \mathbf{J}_{\mathbf{g}^*}\mathbf{v}_{\mathbf{g}^*} + \mathbf{J}_{\boldsymbol{\sigma}}\mathbf{v}_{\boldsymbol{\sigma}} + \mathbf{J}_{\boldsymbol{\sigma}_n}\mathbf{v}_{\boldsymbol{\sigma}_n}. \tag{22}$$

If we unvectorize both sides and substitute the definition of each Jacobian we find

$$\begin{aligned} \mathbf{C} &= \text{unvect}(\mathbf{c}) \\ &= \text{diag}(\mathbf{v}_{\mathbf{g}})\mathbf{A}\mathbf{R}_s\mathbf{A}^H\mathbf{G}^H + \mathbf{G}\mathbf{A}\mathbf{R}_s\mathbf{A}^H\text{diag}(\mathbf{v}_{\mathbf{g}^*}) \\ &+ \mathbf{G}\mathbf{A}\mathbf{V}_{\boldsymbol{\sigma}}\mathbf{A}^H\mathbf{G}^H + \mathbf{V}_{\boldsymbol{\sigma}_n} \\ &= \left(\mathbf{v}_{\mathbf{g}}\mathbf{g}^H + \mathbf{g}\mathbf{v}_{\mathbf{g}^*}^T\right) \odot \mathbf{R}_0 + \mathbf{G}\mathbf{A}\mathbf{V}_{\boldsymbol{\sigma}}\mathbf{A}^H\mathbf{G}^H + \mathbf{V}_{\boldsymbol{\sigma}_n}, \end{aligned} \tag{23}$$

where $\odot$ is the Hadamard or element–wise multiplication, $\mathbf{R}_0 = \mathbf{A}\mathbf{R}_s\mathbf{A}^H$, $\mathbf{V}_{\boldsymbol{\sigma}} = \text{diag}(\mathbf{v}_{\boldsymbol{\sigma}})$ and $\mathbf{V}_{\boldsymbol{\sigma}_n} = \text{unvect}(\mathbf{S}\mathbf{v}_{\boldsymbol{\sigma}_n})$. If $\mathbf{R}_n$ is a diagonal matrix then $\mathbf{V}_{\boldsymbol{\sigma}_n} = \text{diag}(\mathbf{v}_{\boldsymbol{\sigma}_n})$.

Because $\mathbf{G}$ and all of the $\mathbf{V}$ matrices are diagonal, the computational complexity for calculating $\mathbf{C}$ is very low. Also because $\mathbf{C}$ has the same dimensions as a covariance matrix calculating $\mathbf{C}_{\mathbf{W}} = \text{unvect}(\mathbf{c}_{\mathbf{W}})$ becomes simply

$$\mathbf{C}_{\mathbf{W}} = \mathbf{W}\mathbf{C}\mathbf{W}^H. \tag{24}$$

Finally we need to calculate

$$\mathbf{H}\mathbf{v} = \mathbf{J}^H\mathbf{c}_{\mathbf{W}} = \begin{bmatrix} [(\mathbf{R}_0^T\mathbf{G}) \odot \mathbf{C}_{\mathbf{W}}]\mathbf{1} \\ [\mathbf{C}_{\mathbf{W}}^T \odot (\mathbf{R}_0\mathbf{G}^H)]\mathbf{1} \\ [(\mathbf{C}_{\mathbf{W}}\mathbf{G}\mathbf{A})^T \odot (\mathbf{A}^H\mathbf{G}^H)]\mathbf{1} \\ \mathbf{S}^H\mathbf{u}_{\mathbf{W}} \end{bmatrix}. \tag{25}$$

The first three operations consist of an element–wise multiplication and summation of the columns of each row, which are computationally cheap operations. Only $\mathbf{R}_0$ and $\mathbf{G}\mathbf{A}$ need to be calculated and saved. The third term can also be calculated efficiently if it is viewed as a beamforming operation done on $\mathbf{C}_{\mathbf{W}}$ with columns of $\mathbf{G}\mathbf{A}$ as the beamformer vectors. The last operation is just a selection operation and if $\mathbf{R}_n$ is diagonal it is equal to vectdiag($\mathbf{C}_{\mathbf{W}}$). If we replace $\mathbf{C}$ by $\hat{\mathbf{R}} - \mathbf{R}$ in (24), the same procedure can be used to calculate $\mathbf{b}$.

In conclusion, to calculate $\mathbf{H}\mathbf{v}$ we perform (23), (24) and (25). The procedure that does these operations is given to the minresQLP along with the gradient $\mathbf{b}$ to produce $\boldsymbol{\Delta}$. Computationally this means that per iteration we have a $O(p^2q)$ complexity while the WALS has a complexity of $O(p^3 + p^2q)$ [3] (for both cases we assume $\mathbf{W}$ to be available beforehand). For ML, $\mathbf{R}$ needs to be inverted at each iteration which increases the complexity, however if $\mathbf{R}_n$ is diagonal the inversion can be reduced from $O(p^3)$ to $O(q^3)$ using the Woodbury matrix identity [8].

IV. Experimental Data

We have used a measurement set from the LOFAR radio–telescope[1] to test our method. The sample covariance matrix of $p = 273$ dipoles is available to us from a single channel with a central frequency of 58.98 MHz and a bandwidth of 195 KHz which is sampled at the Nyquist rate. The integration time for this covariance matrix is 1 second.

The proposed method is used to calibrated this array for three different cases. In each case we assume to know the position of one, two and three sources such that $q = 1, 2$ and 3. For each case we look at the norm for the residual $\mathbf{E}$ defined as

$$||\mathbf{E}||_F = ||\hat{\mathbf{R}} - \mathbf{R}(\hat{\boldsymbol{\theta}}))||_F,$$

and the norm of the gradient defined by (17). These results are illustrated in Figure 1a and 1b. From norm of the gradient we know that in all three cases the algorithm converges after a few iterations. However, for $q = 1$ the final residual is much higher than the other two and it takes the algorithm much longer to converge. In order to explain this we use the calibration results to make a full sky image (Figure 1c). In this image we observe that two strong point sources , Cygnus A and Cassiopeia A, are visible to the array and also there is a strong background radiation from the Milky–Way (going from top to the bottom of the image). In order to obtain better calibration results we use the fact that the extended sources like the Milky–Way mostly affect the shorter baselines [9]. Based on this knowledge we have chosen the selection matrix $\mathbf{S}$ to include the baselines smaller than 25 times the wavelength into $\mathbf{R}_n$ as noise. In this case the total number of unknowns, n, is 23393. The proposed method has a complexity of $O(p^2q)$ while a naive and direct approach to the Gauss–Newton algorithm would have had a complexity of $O(n^3)$ which would be extremely expensive.

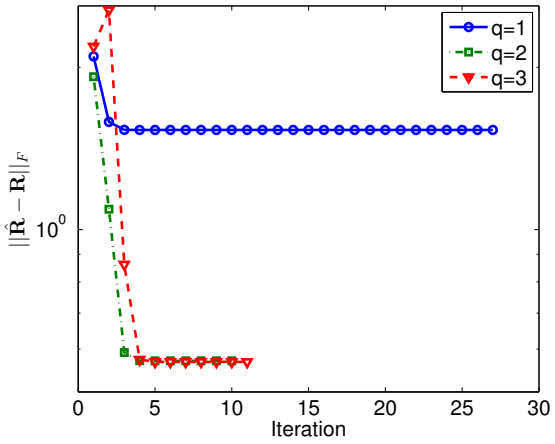
By adding Cassiopeia A and the Milky–Way to the model we have achieved a much better result for $q = 2$. In order to verify this we also made Figure 1d by imaging the residual, $\mathbf{E}$, which shows how smaller sources (three orders of magnitude lower than the strong sources) can now be detected from this residual.

Finally, for the case $q = 3$ we have added the next brightest source to the model, in this case we also converge and have a smaller residual. However because the sources are much weaker the difference is harder to visualize.

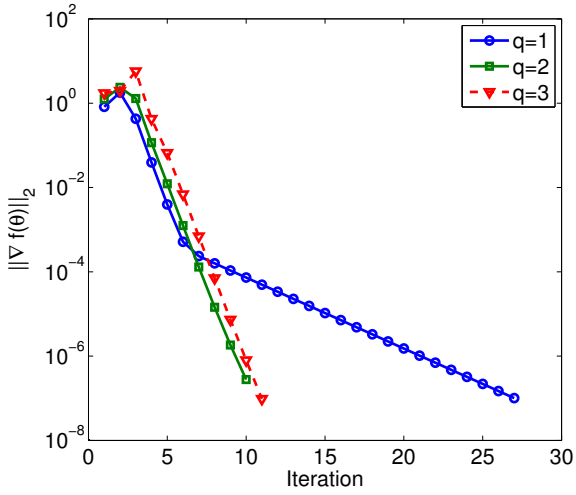
V. Conclusion

We have shown that the covariance data model for calibrating a radio–telescope has a strong Khatri–Rao structure. We have shown that this structure could be used to perform fast matrix–vector computations which is the building block of the Krylov subspace based methods. Finally we have used the proposed method to calibrate real measurement set from the

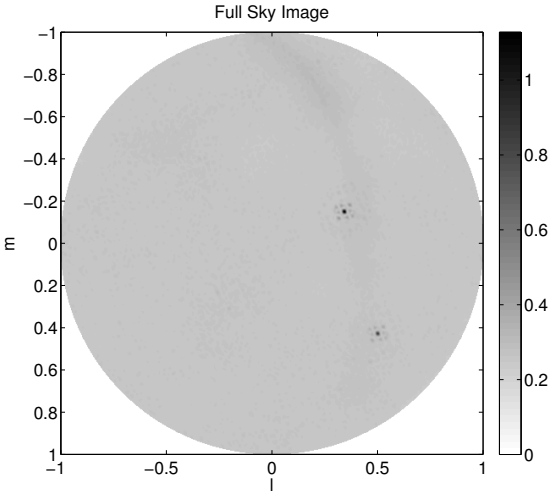
[1]This data is the courtesy of ASTRON (http://www.astron.nl) and is provided to the authors as part of a collaboration within the NWO TOP project. We also would like to acknowledge the help of Stefan Wijnholds and Peeyush Prasad for obtaining this data.



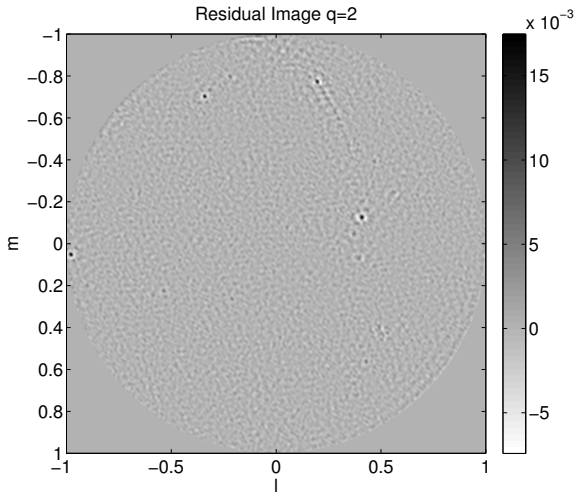
(a) Residual norm for $q = 1, 2$ and 3



(b) Convergence of the gradient to zero



(c) Full Sky Image after gain correction



(d) Full Sky Image after removal of Cas A and Cyg A

Fig. 1: Results from Calibrating LOFAR Data

LOFAR radio–telescope with low computational complexity. A thorough comparison of proposed method with similar algorithms is lacking and will be addressed in future work.

References

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

[2] S. Wijnholds, S. van der Tol, R. Nijboer, and A.-J. van der Veen, "Calibration challenges for future radio telescopes," *Signal Processing Magazine, IEEE*, vol. 27, no. 1, pp. 30–42, Jan 2010.

[3] 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.

[4] 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.

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

[6] B. Ottersten, P. Stoica, and R. Roy, "Covariance matching estimation techniques for array signal processing applications," *Digital Signal Processing*, vol. 8, no. 3, pp. 185 – 210, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200498903165

[7] M. R. Osborne, "Fisher's method of scoring," *International Statistical Review / Revue Internationale de Statistique*, vol. 60, no. 1, pp. pp. 99–117, 1992. [Online]. Available: http://www.jstor.org/stable/1403504

[8] M. A. Woodbury, *Inverting Modified Matrices*, ser. Statistical Research Group Memorandum Reports. Princeton, NJ: Princeton University, 1950, no. 42.

[9] S. J. Wijnholds, "Fish-eye observing with phased array radio telescope," Ph.D. dissertation, Delft University of Technology, 2010.