



RHODES UNIVERSITY
Where leaders learn

Addressing flux suppression, radio frequency interference, and
selection of optimal solution intervals during radio interferometric
calibration

A thesis submitted in fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

of


Rhodes University

by

Ulrich Armel Mbou Sob

Supervised by Prof. Oleg M. Smirnov & Dr Hertzog L. Bester

Rhodes Centre for Radio Astronomy Techniques & Technologies

 0000-0001-9710-9368

March, 2020

To my family...

Abstract

The forthcoming Square Kilometre Array is expected to provide answers to some of the most intriguing questions about our Universe. However, as it is already noticeable from MeerKAT and other precursors, the amounts of data produced by these new instruments are significantly challenging to calibrate and image. Calibration of radio interferometric data is usually biased by incomplete sky models and radio frequency interference (RFI) resulting in calibration artefacts that limit the dynamic range and image fidelity of the resulting images. One of the most noticeable of these artefacts is the formation of spurious sources which causes suppression of real emissions. Fortunately, it has been shown that calibration algorithms employing heavy-tailed likelihood functions are less susceptible to this due to their robustness against outliers.

Leveraging on recent developments in the field of complex optimisation, we implement a robust calibration algorithm using a Student's t likelihood function and Wirtinger derivatives. The new algorithm, dubbed the *robust solver*, is incorporated as a subroutine into the newly released calibration software package CubiCal. We perform statistical analysis on the distribution of visibilities and provide an insight into the functioning of the robust solver and describe different scenarios where it will improve calibration. We use simulations to show that the robust solver effectively reduces the amount of flux suppressed from unmodelled sources both in direction independent and direction dependent calibration. Furthermore, the robust solver is shown to successfully mitigate the effects of low-level RFI when applied to a simulated and a real VLA dataset.

Finally, we demonstrate that there are close links between the amount of flux suppressed from sources, the effects of the RFI and the employed solution interval during radio interferometric calibration. Hence, we investigate the effects of solution intervals and the different factors to consider in order to select adequate solution intervals. Furthermore, we propose a practical brute force method for selecting optimal solution intervals. The proposed method is successfully applied to a VLA dataset.

Declaration of Non-Plagiarism

I, **Ulrich Armel Mbou Sob** declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers.

Acknowledgments

I never dare to dream I could do a PhD in my life, but most often, our lives are shaped by those who cross our paths. This note of thanks is the final touch on my thesis and is dedicated to all those who supported me throughout this journey.

First and foremost I thank my supervisor Prof Oleg Smirnov – you are not just a great supervisor but also a kind, supportive and understanding human being from whom I learnt a lot. I thank my co-supervisor Dr Landman Bester, for his patience and willingness to always help. I am also indebted to Dr Jonathan Kenyon, whose PhD paved the way for mine and for allowing me to use most of his codes. I thank Dr Trienko Grobler with whom I wrote the initial research proposal.

I thank Ronel, Zizipo and Verushca for doing all the administrative procedures on our behalf. Finally, on the academic level, I thank the SARAQ, the South African National Research Foundation and Newton Foundation for funding my studies.

I thank all my colleagues from the Department and particularly those from the RATT group. Thank you for all the surprised birthday parties and for accepting to play football with me even those who had not done any physical exercise in a decade.

I thank my family for their moral support and prayers. I also thank all the new friends I made in South Africa. You all made sure I felt at home here in South Africa. Finally, I give praise to the almighty God for all his blessings in my life. . .

Publication List

Publications that form part and/or include research presented in this thesis:

1. Sob, U. M., Bester, H. L., Smirnov, O. M., Kenyon, J. S., & Grobler, T. L., *Radio interferometric calibration using a complex Students t -distribution and Wirtinger derivatives*, Monthly Notices of the Royal Astronomical Society, Volume 491, Issue 1, January 2020, Pages 1026–1042.
2. Sob, U. M., Bester, H. L., & Smirnov, O. M., *Solution Intervals Considered Harmful*, in preparation.

The code for the robust solver is publicly available on the Github repository of CubiCal at:

<https://github.com/ratt-ru/CubiCal/>

Contents

Abstract	ii
Declaration of Non Plagiarism	iii
Acknowledgments	iv
Declaration of Publications	v
Preface	xvi
1 Introduction	1
1.1 Cosmic Signals	2
1.2 Single-dish-Astronomy	5
1.3 Radio Interferometry	6
1.3.1 Visibility function	8
1.3.2 Sky Mapping	9
1.4 Sensitivity	12
1.5 KAT-7, MeerKAT and VLA	13
1.6 Radio sciences	14
2 Radio Interferometric Data Reduction — Calibration	17
2.1 Radio Interferometer Measurement Equation (RIME)	17
2.1.1 Discrete sources formulation	17
2.1.2 Jones Matrices	20
2.1.3 Full-sky RIME & Van Cittert-Zernike theorem	22
2.2 Data Reduction Steps	23

2.2.1	Preprocessing	24
2.2.2	First Generation Calibration (1GC)	25
2.2.3	Second Generation Calibration (2GC)	27
2.2.4	Third Generation Calibration (3GC)	30
2.3	Conventional Calibration Algorithms	31
2.3.1	Schwab & Thompson-D’Addario method	31
2.3.2	Non Linear Least Squares (NLLS) methods	32
2.3.3	StEFCal	35
2.4	Calibration Bottlenecks and Novel Calibration Algorithms	36
2.4.1	Calibration Bottlenecks	36
2.4.2	Novel Calibration Algorithms	40
3	Calibration using a Complex Student-t distribution and Wirtinger derivatives	44
3.1	Calibration as a complex optimisation problem	44
3.1.1	The Wirtinger approach	45
3.1.2	CubiCal overview	47
3.2	Proper Complex Student’s t Calibration	48
3.2.1	Proper CST	48
3.2.2	Robust Calibration	49
3.2.3	Implementation details	51
3.2.4	Computational cost	53
4	Applications of the Robust Solver	56
4.1	Robust solvers and flux suppression	57
4.1.1	Statistical properties of visibilities	57
4.1.2	SNR, model concentration, and flux suppression	62
4.1.3	Flux suppression in DD calibration	66
4.1.3.1	Simulation setup	66
4.1.3.2	Results	67

4.1.3.3	Solution intervals	68
4.1.4	Flux suppression from extended sources	73
4.2	Robust solvers and RFI mitigation	76
4.2.1	Simulating low-level RFI	77
4.2.2	Application to real data	79
4.3	Discussion	84
5	Solution Intervals Considered Harmful	86
5.1	Problem overview	87
5.1.1	Calibration	87
5.1.2	Solution Intervals	88
5.2	Gain Errors	92
5.2.1	Noise in the visibilities	97
5.2.2	Intrinsic variability of the gains	106
5.2.3	Model Completeness & Radio Frequency Interference	108
5.3	Searching for optimal solution intervals	108
5.3.1	Calibration minor cycle	108
5.3.2	Optimal solution interval search algorithm	109
5.4	Application on simulated data	112
5.4.1	Time only simulations	112
5.4.2	Time and frequency	115
5.5	Application to Real Data	117
5.6	Discussion	122
6	General Conclusions	124
Appendix A	Non Linear Least Squares (NLLS) methods	127
Appendix B	Expectation Maximisation for Non-linear Models with Proper CST Noise	131
Appendix C	Boxcars parametrisation of the gains	136

Appendix D Simulation Tools	137
D.1 Simulation pipeline	137
D.2 Generating realistic gains	139
Bibliography	142

List of Figures

1.1	EM windows	3
1.2	Element of solid angle	4
1.3	Primary beam	6
1.4	Two-element interferometer	7
2.1	Data reduction pipeline	24
2.2	Ghost sources in Westerbork Synthesis Radio Telescope (WSRT) observations.	38
4.1	Histogram for visibilities of point sources	58
4.2	Histogram of the visibilities of extended sources	60
4.3	Plot of Average Suppression against Signal to Noise Ratio	65
4.4	DD flux suppression plots for high-SNR	69
4.5	DD flux suppression plots for low-SNR	70
4.6	DD flux suppression against solution interval	72
4.7	Images of the extended source simulation	75
4.8	Extended source suppression	76
4.9	Images of the RFI simulations	80
4.10	“VIDEO” field visibilities before self-calibration	81
4.11	Images of RFI mitigation on the “VIDEO” field	82
4.12	Gain plots for the “VIDEO” field	84
5.1	Boxcar reconstructed gains	91
5.2	MSE of estimated gains and artefacts rms	96
5.3	MSE of estimated gains against solution interval on a log scale	100

5.4	Corrected images at different time intervals	105
5.5	Gain's bias against solution interval	106
5.6	Artefact maps for intrinsic gains variability	107
5.7	GP simulated gains	113
5.8	AIC plots for time only simulations	114
5.9	AIC of "VIDEO" field	118
5.10	Residual images of the simulated "VIDEO" field	119
5.11	"VIDEO" field rms per channel	121
5.12	Residual images of the real"VIDEO" field dataset	122

List of Tables

5.1	Table with simulation setups	93
5.2	Gaussian Process parameters and input rms (simulation \mathbf{v})	94
D.1	Basic MS configuration	138

LIST OF ABBREVIATIONS

- 1GC** First Generation Calibration
- 2GC** Second Generation Calibration
- 3GC** Third Generation Calibration
- AIC** Akaike Information Criterion
- AS** Average Suppression
- CST** Complex Student's t-distribution
- DD** Direction Dependent
- DDE** Direction Dependent Effect
- DI** Direction Independent
- DIE** Direction Independent Effect
- EM** Electromagnetic
- FIM** Fisher Information matrix
- GN** Gauss-Newton
- GP** Gaussian Processes
- KAT-7** Karoo Array Telescope
- LM** Levenberg-Marquardt

MeerKAT (Meer) Karoo Array Telescope

MSE Mean Squared Error

MS Measurement Set

NLLS Non Linear Least Squares

RFI Radio Frequency Interference

RI Radio Interferometry

RIME Radio Interferometer Measurement Equation

rms root mean squared error

SNR Signal-to-Noise-Ratio

ST Student's t-distribution

VLA Karl G. Jansky Very Large Array

LIST OF RECURRENT SYMBOLS AND NOTATIONS

N_a	Number of antennas
N_{bl}	Number of baselines
σ_{rms}	Visibilities' rms
v	Scalar visibilities
\tilde{v}, d	Corrupted visibilities (scalar)
r	Scalar residual visibilities
g	Scalar gains
J	Arbitrary Jones matrix
\mathbf{J}	Jacobian matrix
\mathbf{W}	Weight matrix
\mathbf{G}	Gain matrix
$\check{\mathbf{R}}$	Augmented residual
$\check{\mathbf{D}}$	Augmented data
$\check{\mathbf{V}}$	Augmented model
Σ	Covariance matrix (chapter 3)
$(\cdot)^T$	Transpose
$(\cdot)^*$	Complex conjugate
$(\cdot)^H$	Hermitian transpose
$\check{(\cdot)}$	Augmented vectors/matrices
$\vec{(\cdot)}$	Vector of matrices
$\mathbf{1}$	Matrix with all entries 1
\mathbf{I}	Identity matrix
$\lceil \cdot \rceil$	Ceil function

Preface

Radio astronomy is currently in an exciting era with the advent of new telescopes such as the Square Kilometre Array (SKA) ([Schilizzi et al., 2008](#)), MeerKAT ([Jonas & Team, 2018](#)), the Hydrogen Epoch of Reionisation Array (HERA; [Greenhill & Bernardi \(2012\)](#)) and the Australian Square Kilometre Array Pathfinder (ASKAP) ([Johnston et al., 2008](#)). This new generation of telescopes, notably the SKA which will be the world's largest and most powerful telescope, are expected to have unprecedented sensitivities, resolutions and data rates. These telescopes will be used to study science topics such as the Epoch of Reionisation (EoR) ([McQuinn et al., 2006](#)), Galaxy formation and evolution (see [Colafrancesco et al. \(2015\)](#); [De Blok et al. \(2017\)](#)), Pulsars ([Burnell, 1984](#)) and Transient sources, and for performing numerous deep continuum surveys (see [Jarvis et al. \(2017\)](#)) in an attempt to answer some of the most fundamental questions about the origin of our Universe (see [Weltman et al. \(2020\)](#) for more details).

However, the unprecedented sensitivities, resolutions and data rates promised by these instruments come with a whole world of data processing challenges. Accurate and efficiently processing, as well as storing, this data is one of the most challenging research topics currently in the radio astronomy community. Crucial to every data processing pipeline in radio astronomy is the process of calibration, which refers to the process of removing all corruptions introduced in the data during the observation. Practically, before any imaging and science study can be done, the data needs to be adjusted to be an accurate representation of the sky. Radio interferometric (RI) calibration is the main focus of this document.

In a nutshell, RI calibration is the process of finding the instrumental parameters and sky model, which best fits the observed or measured data. This problem is non-trivial since we only have the observed data. However, we need to simultaneously solve for both the instrumental parameters (propagation effects) and our intended target sky. It is well documented (see [Linfield \(1986\)](#), [Wilkinson et al. \(1988\)](#), [Martí-Vidal & Marcaide \(2008\)](#), [Grobler et al. \(2014\)](#)

[Wijnholds et al. \(2016\)](#)) that inaccurate modelling of propagation effects and sky parameters (incomplete sky models or wrong sky models) during calibration results in spurious sources called artefacts that degrade the quality of the output images and the deduced scientific conclusions. One particularly worrying effect from these artefacts is the flux suppression/overestimation of faint emissions. This problem needs to be well addressed since most interesting astrophysical signals are faint.

Due to the inevitability of having unmodelled sources during calibration, the calibration problem therefore needs to be reformulated adequately to avoid suppressing faint signals. This entails understanding precisely to what extent these faint signals contribute to the calibration solutions. Another unmodelled signal during calibration is radio frequency interference (RFI). Interferometric data is always corrupted by unwanted signals from different sources emitting at radio wavelengths. These unwanted sources include, for example, signals from television and radio stations, flying objects such as helicopters and planes, and even radio signals mistakenly generated by people working at observatories. These effects can generally be mitigated through rigorous flagging, but it is practically impossible to completely get rid of all RFI. Because of the presence of unmodelled sources and RFI, we need to carefully formulate our calibration problem mathematically and adapt existing algorithms in order to mitigate the effects of these outliers.

Traditional calibration methods have mostly employed non linear least squares (NLLS) algorithms such as the Levenberg-Marquardt (LM) and the Gauss-Newton (GN) (see [Madsen et al. \(2004\)](#)) with a few exceptions such as trust-region methods ([Yatawatta, 2013](#)) and quasi-Newton methods ([Yatawatta et al., 2019](#)). Furthermore, because interferometric data is complex, until recently, calibration algorithms proceeded by first splitting the data and the different propagation effects into their real and imaginary components before deriving the update rules for the optimisation. The recent developments in the field of complex optimisation particularly the application of Wirtinger calculus (see [Kreutz-Delgado \(2009\)](#); [Sorber et al. \(2012\)](#)), have made it possible to circumvent the need to split the data, and instead to treat calibration as a complex optimisation problem. [Tasse \(2014a\)](#) and [Smirnov & Tasse \(2015\)](#) have shown that exploiting Wirtinger calculus yields significant algorithmic advantages. Specifically, by careful ordering of the data, the Hessian of the optimisation problem can be adequately approximated by its diagonal, leading to

significant algorithmic speed-ups. In this work, we follow the same approach in formulating the calibration problem as a complex optimisation problem. Furthermore, we extend previous work from [Kazemi & Yatawatta \(2013\)](#) and [Ollier et al. \(2017\)](#) in order to address the problem of flux suppression from faint sources and the effects of RFI during calibration. The last investigated issue we report in this document is the choice of solution intervals during calibration. Data is generally averaged during calibration to improve the signal-to-noise-ratio. However, as we illustrate throughout the thesis, there is a link between the employed solution interval, the amount of flux suppression and the effects of RFI during calibration.

We begin in Chapter 1 with a brief introduction to radio astronomy and interferometry, focussing on the description of the fundamental observables used by radio astronomers for their studies. Next in Chapter 2, we introduce the Radio Interferometer Measurement Equation (RIME), which is fundamental for calibration, then discuss the different generations of calibration, particularly self-calibration, calibration algorithms and briefly introduce calibration artefacts.

In Chapter 3, we describe the formulation of calibration as a complex optimisation problem using Wirtinger calculus. This is followed by the mathematical description of our novel calibration solver dubbed the *robust solver* and its implementation details. In Chapter 4, the robust solver is tested using both simulated and real data sets. Furthermore, we perform a statistical analysis of visibilities in order to fully understand flux suppression and elucidate the different scenarios in which robust calibration is adequate.

Chapter 5 extends the results of Chapter 4 by looking in detail at the effects of solution intervals during calibration. We discuss these effects and describe an approach for selecting optimal solution intervals. It should be noted that this chapter does not transcend the question of selecting solution intervals but rather highlights the necessity of thinking beyond solution intervals. The main conclusions and discussions of the methods presented in this document are the focus of Chapter 6.

Four Appendices accompany the main body of the thesis. Appendix A presents a short review of the NLLS algorithms employed in the thesis. Appendix B describes the full details on the derivation of the robust solver algorithm. Appendix C gives the explicit form of the parametri-

sation used to represent gains as a function of solution intervals. Appendix [D](#) describes specific tools that we repeatedly use for simulations in the thesis. The essential part here is how we generate the gains with which we corrupt the data in simulations and constrain them to have specific statistical properties using Gaussian Processes.

CHAPTER 1

Introduction

The presence of electromagnetic (EM) radiation from celestial objects such as cosmic sources, galaxies, stars and the interstellar medium provide an essential means of probing and studying the Universe. Astronomy, which is the science that studies the Universe, i.e. celestial objects and the different physical phenomena that lead to their formation, has relied on EM radiation to reveal most of our understanding of the Universe to date. From a historical point of view, the science of astronomy has mostly used the visible part of the EM spectrum – called Optical Astronomy. Astronomy now transcends the visible range (wavelength – 740 nm to 380 nm) and employs nearly all of the EM spectrum leading to different fields such as X-ray, Gamma-ray and Radio Astronomy.

Radio astronomy, which studies the Universe through the radio part (i.e. wavelength \approx 15 m to 3 mm) of the EM spectrum, was born in the early 1930s. Karl G. Jansky who was a radio engineer at the Bell laboratories ([Jansky, 1933](#); [Kraus, 1966](#)), while trying to determine the origin of a source of noise in their receivers at 20 MHz, found out that the signal was from the centre of our galaxy, the Milky Way. Jansky did not perform further investigation of this signal as the Bell laboratory felt the signal was too weak to affect their communication. Grote Reber, also a radio engineer, reviewed Jansky's work and suggested the signal could be detected at a high frequency because he thought it was of thermal origin. Using a parabolic dish he built in his backyard ([Reber, 1940](#)), he had several unsuccessful attempts to detect the signal at high frequencies. He was finally able to trace out the signal at 162 MHz, and the emission mechanism remained

mysterious until the much later discovery of synchrotron radiation. Grote Reber is credited with building the first-ever radio telescope. Following these discoveries, the field of radio astronomy started flourishing after the second world war as most of the techniques developed for radar during the war were transferred to radio astronomy.

Moving away from history, it is essential to be able to study celestial objects at different wavelengths. Studying at multiple wavelengths does not only allow accurate identification of objects, but the different spectrum reveals different properties of these objects. Furthermore, most radiation from the EM spectrum is severely absorbed by molecules in the Earth's atmosphere, notably water vapour, H_2O , and O_2 . This absorption results from these molecules having rotation band energies corresponding to the wavelengths from these radiations. For example, H_2O has energy bands at wavelengths 1.35 cm and 1.63 mm while O_2 has an energy band at 5 mm (Wilson et al., 2009). For wavelengths < 1 mm, the absorption is mostly dominated by N_2 and CO_2 molecules. Coincidentally, radio waves are the least absorbed and have the largest window compared to all other parts of the EM spectrum (Condon & Ransom, 2016). Fig. 1.1 is a schematic representation of the different EM windows illustrating the transparency of the radio window for ground-based astronomy. At low frequencies, the atmosphere ceases again to be transparent to EM waves because of free electrons in the atmosphere that scatter EM radiations through the process of Compton scattering. Hence, the transmission of EM waves through the atmosphere is only possible for waves at frequencies higher than a certain frequency called the plasma frequency. This sets a low frequency cutoff of ≈ 4.5 MHz for the radio window for any layer in the atmosphere with the average maximum electron density, N_e , of $\approx 2.5 \times 10^5 \text{cm}^{-3}$ (Wilson et al., 2009).

1.1 Cosmic Signals

In astronomy, we seek to measure EM waves, henceforth called signals from cosmic sources in the sky, using receiving elements called antennas or telescopes. The voltages induced in the antennas by such signals are referred to as cosmic signals. We characterise sources using the amount of EM waves they radiate, or the rate at which we receive them, called intensity. A

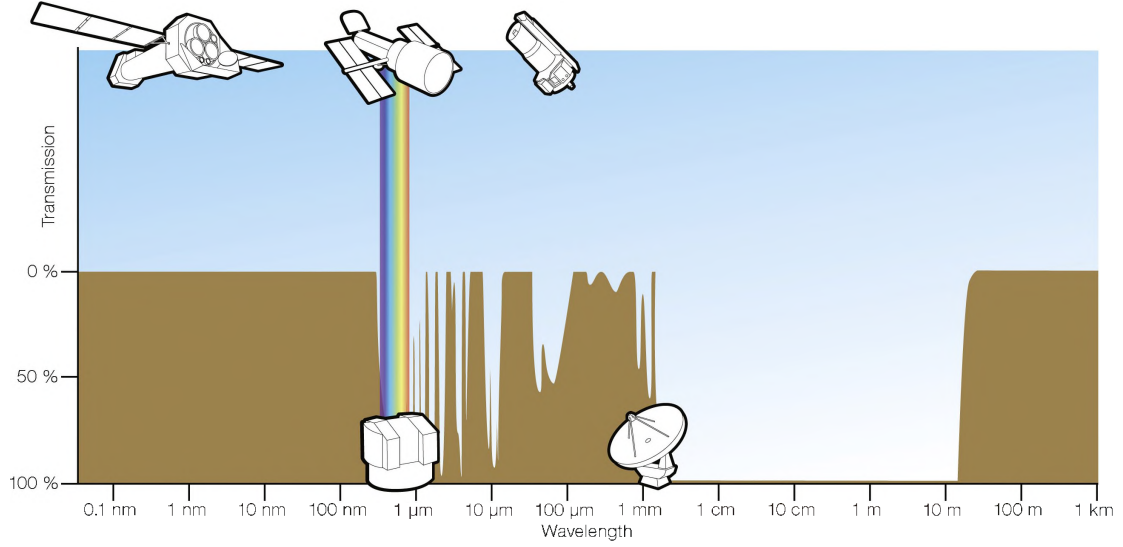


Figure 1.1: Schematic representation of the Earth’s atmosphere opacity to EM radiation. The clear zone lying between $\lambda \approx 1$ mm and $\lambda \approx 30$ m depicts the so-called “radio window” through which celestial radio waves are not absorbed by gas molecules in the Earth’s atmosphere (Condon & Ransom, 2016).

source’s intensity is thus defined as the amount of power emitted per unit angle by the solid angle subtended by the source surface at a specific frequency (Thompson et al., 2017). In radio astronomical imaging, it is more intuitive to think of the signal as emanating from the surface of the celestial sphere. Hence, we generally refer to the source intensity as its surface intensity or surface brightness. A receiving element or telescope will only receive an amount of radiation proportional to its collecting area. Thus the quantity which is generally measured is the specific intensity defined as the intensity per unit area. The specific intensity, I_ν , has units $\text{Wm}^{-2}\text{Hz}^{-1}\text{sr}^{-1}$.

If we consider measuring radiation from a source in direction, s , subtending a solid angle, $d\Omega$, with a bandwidth, $d\nu$, using an instrument with an element of surface area, dA (see Fig. 1.2), then the total power, dP , received is described by

$$dP = I_\nu(s)d\Omega d\nu dA . \quad (1.1)$$

The most common quantity used to describe the amount of radiation from a source is the flux

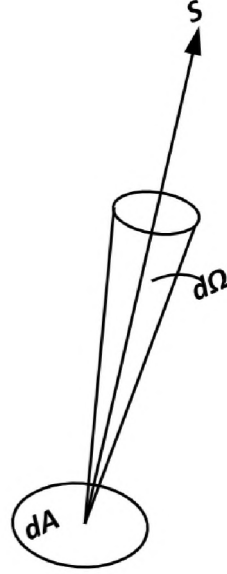


Figure 1.2: Element of solid angle and surface area used for the definition of surface intensity (Thompson et al., 2017).

density, S_ν . We obtain the flux density by integrating the specific intensity over the total solid angle, Ω , subtended by the source,

$$S_\nu = \int_{\Omega} I_\nu(s) \cos \gamma d\Omega , \quad (1.2)$$

where γ is the angle between the element of area, dA , and the direction, s . The flux density has unit $\text{Wm}^{-2}\text{Hz}^{-1}$. Cosmic signals are usually very weak and hence a new unit called the Jansky (Jy) was introduced for such small values where

$$1 \text{ Jy} = 10^{-26} \text{Wm}^{-2}\text{Hz}^{-1} .$$

For thermal radiation from a black body its intensity at frequency, ν , is related to its brightness temperature, T , by Planck's law (Wilson et al., 2009) as

$$I_\nu = \frac{2k_b T \nu^2}{c^2} \left(\frac{h\nu/k_b T}{e^{h\nu/k_b T} - 1} \right) , \quad (1.3)$$

where h is Planck's constant and k_b is the Boltzmann constant. In the regime $h\nu \ll k_b T$, the Rayleigh-Jeans law holds, i.e.

$$I_\nu = \frac{2k_b T \nu^2}{c^2} \Rightarrow T = \frac{I_\nu c^2}{2k_b \nu^2} . \quad (1.4)$$

Eq. (1.4) gives the brightness temperature of the source. Note that, in radio astronomy, even for non-thermal emission, Eq. (1.4) is still used to define the brightness temperature even though the measured quantity does not represent a real physical temperature but gives an indication on the strength of the source.

1.2 Single-dish-Astronomy

As the name implies, in single-dish-astronomy, we study the Universe using voltages from a single dish or an antenna. Antennas are the basic elements used to measure the EM radiation emitted by observed sources. One of the fundamental properties of an antenna is its primary beam or beam pattern. The primary beam of an antenna describes the antenna's sensitivity as a function of the direction of the incoming radiation. Fig. 1.3 is a basic illustration of an antenna's reception pattern. It consists of the main lobe, which is approximately elliptically Gaussian for most antenna types, and numerous side lobes. The primary beam of an antenna is generally characterised by its half power beam width (HPBW), i.e. the angle between the half-power points of the main lobe. The HPBW indicates the antenna's field of view. For an antenna with a circularly symmetric untampered aperture with diameter, D (Thompson et al., 2017), its HPBW is given by

$$\text{HPBW} \approx \frac{1.02\lambda}{D}, \quad (1.5)$$

where λ is the wavelength of the observation. Another important property of an antenna is its resolving power or resolution, i.e. the finest angular scale at which it can distinguish two objects. Mathematically, the resolution, θ , of a single dish telescope can be approximated as

$$\theta \approx \frac{1.22\lambda}{D}. \quad (1.6)$$

Eq. (1.5) and (1.6) tells us that both the HPBW and the resolution are directly proportional to the wavelength and inversely proportional to the aperture's diameter. Furthermore, for single dishes, $\text{HPBW} \approx \theta$. The angular resolution of a single dish is significantly limited in the radio regime because radio waves have very long wavelengths. For example, the Green Bank Tele-

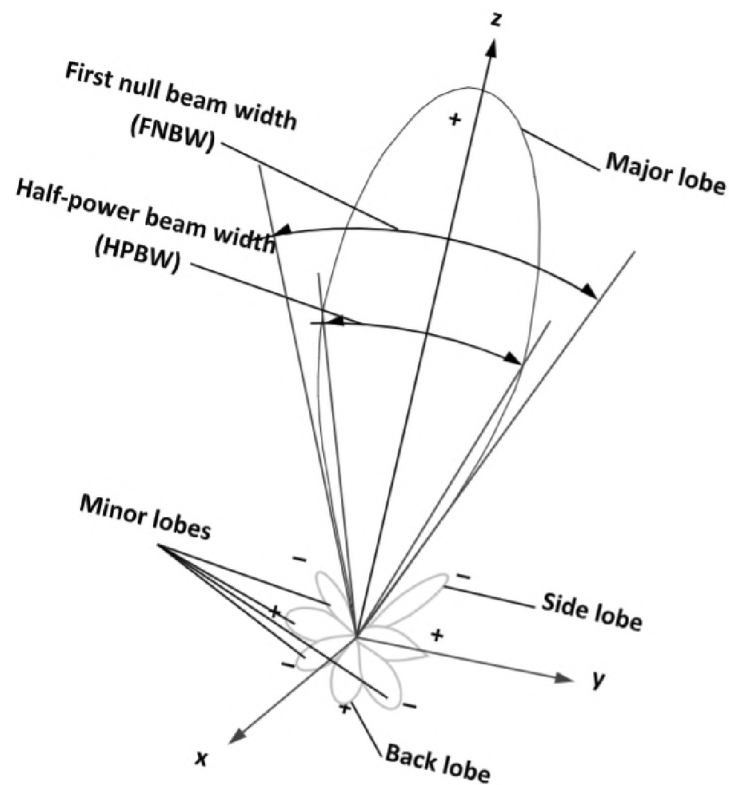


Figure 1.3: Basic illustration of an antenna response pattern or primary beam. We can see the half power beam width (HPBW), which is used to define an antenna’s field of view, and side lobes where the signals are heavily attenuated and back lobes (Taylor et al., 1999).

scope (Prestage et al., 2009) which has a diameter of 100 m will only have a resolution of ≈ 35 arcminutes for an observation with a wavelength of 1 m.

1.3 Radio Interferometry

In order to overcome the resolution limits of single dish telescopes, astronomers use a technique called interferometry. The technique of interferometry in astronomy dates to 1880 with Michelson (Thompson et al., 2017). The first interferometer was the Michelson stellar interferometer. The instrument used constructive and destructive interference of signals from two separated receiving elements to determine the angular width of objects. Fig. 1.4 is a schematic representation

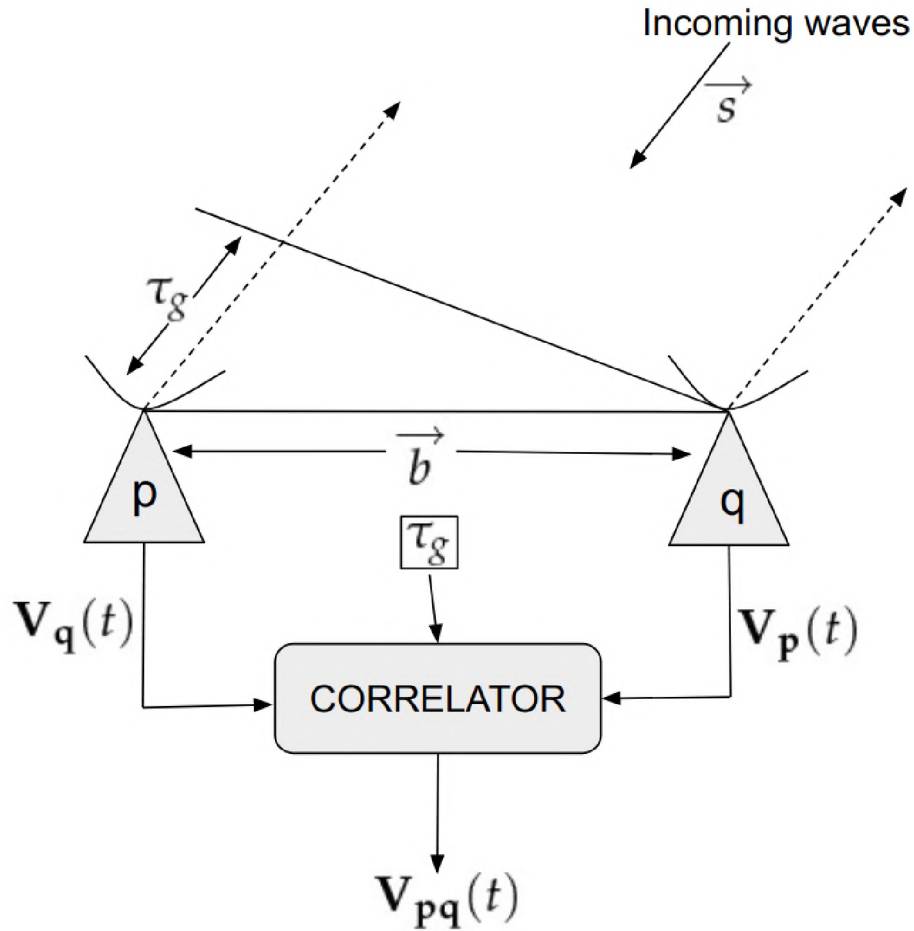


Figure 1.4: Schematic representation of a two-element interferometer. The signal from a source with direction vector \vec{s} is measured by the antenna pair, p and q . The signal is then correlated while inserting a time delay, τ_g , to compensate for the path difference between both antennas in order to obtain the measured visibilities.

of a basic two-element interferometer. Interferometers nowadays consist of multiple such pairs of elements operating as a single unit called arrays. For an interferometer, the resolution, θ , is defined as

$$\theta \approx \frac{\lambda}{B}, \quad (1.7)$$

where B is the longest baseline, i.e. the longest separation between all antenna pairs. Hence, we can achieve high resolutions by constructing arrays with very long baselines.

1.3.1 Visibility function

The measurements recorded by single dishes and interferometers are called visibilities. The term visibility stems from Michelson's definition based on fringe amplitudes (Thompson et al., 2017). For a radio interferometer (see Fig. 1.4), visibilities correspond to the correlation of the voltages measured by both antennas. Consider the two-element interferometer in Fig. 1.4, if a signal travels in the direction \vec{s} to both antennas, p and q , the signal will reach both antennas at different times. Thus, before correlating the voltages from both antennas, a time delay is introduced in the signals to ensure we correlate signals at the same time. The time delay, also known as the geometrical delay, is given by

$$\tau_g = \frac{\vec{b} \cdot \vec{s}}{c}, \quad (1.8)$$

where \vec{b} is the baseline vector connecting both antennas. If the output voltages from both antennas are $\mathbf{V}_p(t)$ and $\mathbf{V}_q(t)$ respectively, then the output from the correlator at time, t , is described as

$$\mathbf{V}_{pq}(t) = \langle \mathbf{V}_p(t) \cdot \mathbf{V}_q(t - \tau_g) \rangle, \quad (1.9)$$

where $\langle \rangle$ denotes averaging.

It is common practice to specify the visibilities as a function of the baseline vectors between antennas. A convenient coordinate system for specifying these is the Cartesian coordinate system, XYZ . We can adopt equatorial coordinates, i.e. hour angle, H , and declination, δ , with the X axis pointing towards $(0^h, 0)$ (the point where the vernal equinox crosses the local meridian), the Y axis towards $(-6^h, 0)$ due east and the Z axis towards the north celestial pole, i.e. $\delta = 0$. If (b_x, b_y, b_z) is the baseline vector between two antennas, then for an observation with reference direction, (H_0, δ_0) , the corresponding baseline coordinates called uvw coordinates are given by (see Thompson et al. (2017) for more details)

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \sin H_0 & \cos H_0 & 0 \\ -\sin \delta_0 \cos H_0 & \sin \delta_0 \cos H_0 & \cos \delta_0 \\ \cos \delta_0 \cos H_0 & -\cos \delta_0 \sin H_0 & \sin \delta_0 \end{bmatrix} \begin{bmatrix} b_x/\lambda \\ b_y/\lambda \\ b_z/\lambda \end{bmatrix}. \quad (1.10)$$

The uvw coordinates are functions of hour angle (i.e. time). These coordinates change throughout an observation as the Earth rotates, making us measure visibilities at different uvw positions. Ideally we want to observe for a very long time so as to sample different portions of the uvw -plane (improve the uv -coverage). The sky is represented using the $(l, m, n = \sqrt{1 - l^2 - m^2})$ frame where the axes are the direction cosines of the u, v and w axes respectively. Using these coordinates systems, the Van Cittert-Zernike theorem states that the measured visibilities and the sky brightness are related by (Thompson et al., 2017)

$$\mathbf{V}(u, v, w) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} A(l, m) I(l, m) e^{-2\pi i (ul + vm + w(\sqrt{1-l^2-m^2}-1))} \frac{dldm}{\sqrt{1-l^2-m^2}}, \quad (1.11)$$

where $\mathbf{V}(u, v, w)$ denotes the measured visibilities at baseline coordinates, uvw , $I(l, m)$ represents the sky brightness distribution and $A(l, m)$ denotes the combined primary beam response of the antennas involved.

1.3.2 Sky Mapping

Interferometers measure visibilities but our science goals rely on recovering the sky brightness, $I(l, m)$, i.e. computing $I(l, m)$ from Eq. (1.11). $I(l, m)$ is generally recovered by approximating Eq. (1.11) using a Fourier transform. The following two specific conditions will make Eq. (1.11) a Fourier transform:

- (i) if we are observing with a coplanar array, then $w = 0$ everywhere and Eq (1.11) reduces to

$$\mathbf{V}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} A(l, m) I(l, m) e^{-2\pi i (ul + vm)} \frac{dldm}{\sqrt{1-l^2-m^2}}, \quad (1.12)$$

which is the two dimensional Fourier transform of $\frac{A(l, m) I(l, m)}{\sqrt{1-l^2-m^2}}$.

- (ii) observing a small area. If our field of view is small such that $(\sqrt{1-l^2-m^2}-1)w \approx -\frac{1}{2}(l^2+m^2)w$ (Thompson et al., 2017) then the w -term in the exponent may as well be neglected leading to the same expression for the visibilities as in Eq. (1.12).

Under any of the above conditions, we write the Van Cittert-Zernike theorem as

$$\mathbf{V}(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathbf{I}(l, m) e^{-2\pi i(ul+vm)} dl dm, \quad (1.13)$$

where $\mathbf{I}(l, m) = \frac{A(l, m)I(l, m)}{\sqrt{1-l^2-m^2}}$. Hence an interferometer measures the Fourier transform of an *apparent sky* which is the true sky attenuated by the antennas' primary beam. The true sky, $I(l, m)$ can be recovered from the image of the apparent sky, $\mathbf{I}(l, m)$ if we have a reasonably good model for the combined primary beam response, $A(l, m)$. Traditionally, the beam is assumed to be constant in time and the same for all antennas for small fields of view. In this case, the factors $A(l, m)$ and $\frac{1}{\sqrt{1-l^2-m^2}}$ can be removed by dividing them from the output image (Smirnov, 2011b). For wide-field observations and arrays with complex beam models, recovering the true sky is not trivial and requires direction dependent calibration (see §2). From Eq. (1.13) the apparent sky brightness is given by

$$\mathbf{I}(l, m) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathbf{V}(u, v) e^{2\pi i(ul+vm)} du dv. \quad (1.14)$$

In practice we do not have a continuous function $\mathbf{V}(u, v)$ for the visibilities since we only sample the (u, v) plane at specific positions. This poses a problem as the recovered sky image is a function of this sampling. Suppose we have a sampling function, $S(u, v)$, which is 1 everywhere we have measurements and 0 otherwise, then the effective reconstructed sky is described as

$$\mathbf{I}^D(l, m) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W(u, v) S(u, v) \mathbf{V}(u, v) e^{2\pi i(ul+vm)} du dv, \quad (1.15)$$

$$= \mathbf{I} * B, \quad (1.16)$$

where $B = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W(u, v) S(u, v) e^{2\pi i(ul+vm)} du dv$ is called the point spread function (PSF), \mathbf{I}^D is the recovered sky image traditionally referred to as the dirty image, and $W(u, v)$ labels the weight applied to each sample visibility defined from its uncertainty. Eq. (1.16) is simply a result of the Fourier convolution theorem. The dirty image is a convolution of the beam attenuated sky,

I , and the PSF (i.e. the instrument response to a point source). I needs to be separated from the dirty image, and this is usually done during imaging using algorithms such as CLEAN (Högbom, 1974) or maximum entropy (Cornwell & Evans, 1985). Note that, for wide-field images, the Fourier relation does not hold. Nevertheless, techniques such as those described by Tasse et al. (2013); Offringa et al. (2014); Tasse et al. (2018) can be used to correct for the effects of the w -term.

Due to the size of datasets which are generally processed in radio interferometry, the most common method for computing the Fourier transform of the visibilities is the Fast Fourier transform (FFT). The FFT requires interpolating the visibilities onto a rectangular uv -grid in a process referred to as *gridding*. The FFT is computationally cheaper than the Direct Fourier transform (DFT). However, gridding is an expensive operation especially for small images, where the DFT may be faster than the combination of gridding and FFT (Taylor et al., 1999).

Another important decision whenever performing imaging is the choice of weighting scheme or weighting function $W(u, v)$ to apply to the visibilities. Two extreme schemes exist namely *natural* and *uniform* weighting. Natural weighting weights each visibility by the inverse of its measured uncertainty as reported by the correlator. Hence natural weighting maximises sensitivity and gives more emphasis to the short baselines (especially for arrays with dense cores) thereby increasing the collecting area of the array. Uniform weighting, on the other hand, tries to assign equal weights to all uv cells. This improves resolution as the long baselines have a similar contribution as the short baselines thereby suppressing side lobes but at the detriment of sensitivity. The weight of each uv cell in uniform weighting is computed as a function of its density, i.e. the number of points in the cell. The problem is finding the perfect trade-off between sensitivity and resolution. An intermediate weighting scheme is the *Briggs* weighting scheme (Briggs, 1995) which has a robust parameter which helps to compromise between natural and uniform weighting. Briggs weighting with the robust parameter close to 2 will correspond to natural weighting while the robust parameter close to -2 will denote uniform weighting. Any intermediate value for the robust parameter will be a mixture of both schemes.

1.4 Sensitivity

The sensitivity of an interferometer is the weakest signal it can detect. Sensitivity plays an essential role in astronomical observations as it sets a limit on the achievable science studies. This section discusses the expected sensitivity of radio interferometers (see [Taylor et al. \(1999\)](#) and references therein for more details). Using the Rayleigh-Jeans approximation to Planck's law, the power P from a signal with brightness temperature T is given by

$$P = k_b T \Delta\nu, \quad (1.17)$$

where k_b is the Boltzmann constant and $\Delta\nu$ is the observing bandwidth. This power is amplified in the antenna's feeds by a gain factor, g^2 , which depends on the type of antenna, to produce an output power,

$$P = g^2 k_b T \Delta\nu. \quad (1.18)$$

The temperature, T , can be split into two components namely: the temperature of the target source, T_a , and the system temperature, T_{sys} , i.e.

$$T = T_a + T_{sys}. \quad (1.19)$$

The system temperature is the power of the system's noise. It consists of contributions from the receiver noise, galactic background, spillover, feed losses and cosmic background. The power from a source with flux density, S , is related to the antenna's collecting area, A , and efficiency, η_a , by

$$P_a = \frac{1}{2} g^2 \eta_a A S \Delta\nu = g^2 k_b T_a \Delta\nu. \quad (1.20)$$

From Eq. (1.20), if we substitute T_{sys} for T_a , we can define a quantity called the system equivalent flux density (SEFD) given by

$$\text{SEFD} = \frac{2k_b T_{sys}}{\eta_a A}. \quad (1.21)$$

The SEFD represents the source's flux density that will produce the same power as the system temperature. The lower the SEFD, the higher the instrument's sensitivity. Any signal weaker

than the SEFD will have a power lower than that from the system temperature. In addition, we can show that the noise rms of an interferometer with antennas having the same properties is

$$\Delta V = \frac{1}{\eta_a} \frac{\text{SEFD}}{\sqrt{2\Delta\nu\delta_t}}, \quad (1.22)$$

where ΔV is the noise rms per baseline often referred to as thermal noise and δ_t is the integration time (see [Taylor et al. \(1999\)](#)).

1.5 KAT-7, MeerKAT and VLA

This section briefly overviews the three telescopes we use in this thesis, namely the Karoo Array Telescope (KAT-7), MeerKAT ([Jonas & Team, 2018](#)) and Jansky Very Large Array (VLA) ([Perley et al., 2011](#)).

KAT-7: KAT-7 is a seven dish telescope located in the Northern Cape Province of South Africa that served as an engineering prototype for the MeerKAT telescope. Its longest baseline is 180 m giving it a resolution of ≈ 3 arcmins at its central frequency of 1.83 GHz. Each dish has a diameter of 12 m giving it a field of view of ≈ 0.8 degrees at 1.83 GHz. KAT-7 operates at a single frequency band with range of 1.2 GHz to 1.95 GHz.

MeerKAT: Following the success of the KAT-7 telescope, the 64 dish MeerKAT telescope was also built in the Northern Cape Province of South Africa. MeerKAT is currently one of the world's most powerful radio telescopes, and it will be absorbed into the future Square Kilometre Array (SKA) ([Schilizzi et al., 2008](#)) phase 1 configuration. MeerKAT has a dense core array configuration which provides it with many short baselines as well as a few long baselines with the longest at 8 km for high-resolution imaging. MeerKAT operates over the frequency range 0.58 GHz to 14.5 GHz. This range is divided into different bands, namely the L-band, the X-band, the S-band, and the UHF-band.

VLA: The VLA is a 28 dish Y-shaped telescope located in New Mexico in the United States of America. Each VLA dish has 25 m diameter. The VLA array operates in a varying number of configurations depending on the specific observation's sensitivity and resolution requirements (configuration A, B, C and D). The A configuration has the longest baseline, i.e. 36 km and thus

provides its highest resolution. VLA operates at frequency range 1.0 GHz to 50 GHz divided over multiple bands namely the L, S, C, U, Ku, K, Ka, and Q bands. Over recent years the VLA has undergone major upgrades from the VLA to Expanded VLA and the current plans for the VLA will be the addition many more antennas to form what will be call the next generation VLA (ngVLA).

Some other current and future general purpose radio interferometric telescopes not mentioned here include, uGMRT in India, LOFAR in Europe, the Australian SKA Pathfinder (ASKAP) in Australia, and the Atacama Large Millimeter Array in Chili.

1.6 Radio sciences

The collaborative efforts to build and upgrade several powerful telescopes [such as the SKA, MeerKAT, the Hydrogen Epoch of Reionisation Array (HERA) ([Greenhill & Bernardi, 2012](#)), the Expanded Very Large Array ([Perley et al., 2011](#))] and the Low-Frequency Array (LOFAR) ([van Haarlem et al., 2013](#))] have radio astronomers at their utmost delight. We conclude this introductory chapter with an overview of some research areas these new instruments will investigate.

Epoch of Reionisation

The Epoch of Reionisation (EoR) is a period in the history of the Universe during which the first luminous sources ionised the neutral interstellar medium (see [McQuinn et al. \(2006\)](#)). Probing this epoch will provide an enormous amount of information about the formation of the Universe and the nature of the first luminous sources – these include objects such as galaxies, stars and quasars. Detecting the redshifted 21-cm signal from Neutral Hydrogen (HI) is believed to be one of the most promising means of probing reionisation, formation and evolution of galaxies as well as dark matter (i.e. an unseen mass that is believed to be the cause of gravitational field holding boundary stars together). EoR projects will be a key science goal for arrays like HERA and LOFAR.

Neutral Hydrogen (HI) and Galaxy formation

The distribution of galaxies in the Universe can be described using the cosmic web. The cosmic web is 3-dimensional network of interconnecting filaments of galaxy clusters and gases separated by voids. It is speculated that these filaments supply gas to galaxies through accretion and act as fuel for continued star formation in galaxies. Studying HI around regions with low column densities will provide a better understanding of galaxy formation and their distributions and will provide evidence or not for cosmic web based cosmological models. MeerKAT, for example, is expected to detect HI signals at redshifts up to 0.4, but using techniques such as *stacking* (see [Healy \(2016\)](#)) it will be possible to push these detections to redshifts up to 1.4. MeerKAT surveys such as LADUMA and MHONGOOSE (see [Booth & Jonas \(2012\)](#)) are examples of surveys that will be carried out in search of HI signals.

Spectral Line studies

Spectral line studies target absorption or emission processes that occur at a specific wavelength as a result of electrons moving across different orbitals or energy states, thereby absorbing or releasing energy in the process. Spectral line studies can be done using different chemical compounds other than HI, such as the Carbon Monoxide (CO) and the Hydroxyl radical (OH) lines. Comparing absorption and emission lines yields information about the temperature and density of the emitting source ([Thompson et al., 2017](#)). With the high spectral resolution of MeerKAT, it will be possible to detect thin absorption lines such as the line splitting caused by the Zeeman effect ([Booth & Jonas, 2012](#)).

Deep Continuum Surveys

Continuum studies, unlike spectral lines, involve multiple frequencies. Continuum studies are not only necessary because of the information provided about the Universe but also because spectral line studies are generally performed only after subtraction of continuum sources from the visibility data. Thus accurate knowledge and modelling of continuum sources are essential

for studies of weak signals like HI, haloes and diffuse emissions which are faint compared to continuum foreground sources (see [Morales et al. \(2006\)](#)). The MeerKAT, and even more the SKA, will have the sensitivity and resolution required for the deepest radio survey ever. These surveys will improve the understanding of known sources and will allow the detection of numerous new sources. An example is the MeerKAT International GHz Tiered Extragalactic Exploration (MIGHTEE) survey ([Jarvis et al., 2017](#)) which will cover an entire field of view of 20 square degrees, with μJy expected sensitivity at frequencies up to the GHz range.

Pulsar and Transients sources

Discovered by Jocelyn Bell and her doctorate supervisor Antony Hewish (who received a Nobel prize), pulsars are highly magnetised neutron stars emitting EM radiation ([Burnell, 1984](#)). Pulsars are only detected when their radiation beam points towards the Earth, hence the name pulsar (i.e. a contraction of “pulse” and “star”). Transient sources produce short-lived burst radiation. Unlike transients, pulsar emissions are continuous or long-lived but can only be detected at specific periods. Studying pulsars and their timing is necessary to understand gravity in binary systems. Transient studies are important, as most often, unknown signals in the data are considered to be radio frequency interference and is flagged. Such signals may be from transient sources or rapidly varying sources. The MeerKAT L-band and high-frequency receiver will be used to explore the Galactic centre and the Galactic plane in order to conduct precision pulsar timing and searches for fast radio transients ([Booth & Jonas, 2012](#)).

Radio Interferometric Data Reduction — Calibration

The process of estimating and removing the different propagation effects present in interferometric data (visibilities), known as *calibration*, is a critical step of any data reduction pipeline. Calibration in radio interferometry, just as in any experimental science, is an essential factor for the quality of the science outputs. In a nutshell, calibration is the process of adapting the parameters of an instrument so that the instrument has the correct response to some a priori known model. This chapter reviews some concepts of radio interferometric data reduction with particular emphasis on the process of calibration. We discuss in §2.1 the mathematical formalism adopted by radio astronomers to describe the measurement process of an interferometer and propagation effects. In §2.2, we describe the different steps involved in a typical data reduction process and how the different steps and techniques have evolved over the years. We present some conventional calibration algorithms in §2.3 and we conclude with a discussion of calibration bottlenecks and some recent calibration frameworks in §2.4.

2.1 Radio Interferometer Measurement Equation (RIME)

2.1.1 Discrete sources formulation

In §1, we defined the visibilities measured by an interferometer as the correlation of the voltages measured by a pair of antennas. From waves and optics, we can represent an electromagnetic signal by its complex vector amplitude, e . If we assume a Cartesian XYZ coordinate system,

with the Z -axis pointing along the direction of propagation then the complex vector amplitude, \mathbf{e} , is defined as

$$\mathbf{e} = \begin{pmatrix} e_x \\ e_y \end{pmatrix}.$$

The choice of coordinates is completely arbitrary as we can easily move from one coordinate system to another through a coordinate transformation. The most common systems adopted in radio astronomy are a linear coordinate system for horizontal and vertical polarisation and circular coordinate system for left and right circular polarisation. If the signal, \mathbf{e} , is assumed to be quasi-monochromatic, then all transformations along its path are linear and can be represented by matrix operators called Jones matrices. Hence the modified voltage, \mathbf{v}_p , measured by an antenna, p , from a source emitting a signal \mathbf{e} is given by

$$\mathbf{v}_p = \mathbf{J}_p \mathbf{e}, \quad (2.1)$$

where \mathbf{J}_p denotes the Jones matrix representing the propagation effects in the direction of antenna p . The visibility matrix for two antennas p and q is thus defined as

$$\mathbf{V}_{pq} = 2 \left\langle (\mathbf{J}_p \mathbf{e}) \cdot (\mathbf{J}_q \mathbf{e})^H \right\rangle \quad (2.2)$$

$$= \mathbf{J}_p \left\langle 2 \begin{pmatrix} e_x \\ e_y \end{pmatrix} \begin{pmatrix} e_x^* & e_y^* \end{pmatrix} \right\rangle \mathbf{J}_q^H \quad (2.3)$$

$$= \mathbf{J}_p \left\langle \begin{pmatrix} 2e_x e_x^* & 2e_x e_y^* \\ 2e_y e_x^* & 2e_y e_y^* \end{pmatrix} \right\rangle \mathbf{J}_q^H \quad (2.4)$$

where $\langle \cdot \rangle$ represents the expectation value, $(\cdot)^H$ represents the conjugate transpose and $(\cdot)^*$ represents the complex conjugate. The validity of Eq. (2.1) relies on the assumptions that we are observing a monochromatic signal and the Jones matrices have infinitesimally small spectral variations. Hence, if we assume the Jones matrices are also constant over sampling times, then we can safely remove them out of the expectation operator to get Eq. (2.3). Note that here the visibilities are defined as a matrix product of a column vector and a conjugate row vector. This is contrary to the 4×4 formalism in which \mathbf{V}_{pq} is a 4×1 vector defined by the following Kronecker

product

$$\mathbf{V}_{pq} = 2 \langle (\mathbf{J}_p \mathbf{e}) \otimes (\mathbf{J}_q \mathbf{e})^* \rangle \quad (2.5)$$

$$= 2 \langle \mathbf{J}_p \otimes \mathbf{J}_q^* \rangle \langle \mathbf{e} \otimes \mathbf{e}^* \rangle. \quad (2.6)$$

The middle term of Eq. (2.4) provides the definition of Stokes parameters in radio interferometry and unifies it with optics (Hamaker et al., 1996). For linearly polarised feeds

$$\begin{pmatrix} 2\langle e_x e_x^* \rangle & 2\langle e_x e_y^* \rangle \\ 2\langle e_y e_x^* \rangle & 2\langle e_y e_y^* \rangle \end{pmatrix} = \begin{pmatrix} I + Q & U + iV \\ U - iV & I - Q \end{pmatrix} = \mathbf{B}, \quad (2.7)$$

where I, Q, U, V are the Stokes parameters and \mathbf{B} is the brightness matrix of the source (Born & Wolf, 1964; Hamaker et al., 1996). The factor of 2 here is introduced as a convention. As discussed in Smirnov (2011a), there are two conflicting conventions, the convention-1/2 and the convention-1. We define the total intensity, Stokes I , as

$$I = \langle e_x e_x^* \rangle + \langle e_y e_y^* \rangle. \quad (2.8)$$

For a 1-Jy unpolarised phase centred source, i.e. unity Stokes parameters ($I = 1, Q = U = V = 0$), $\langle e_x e_x^* \rangle = \langle e_y e_y^* \rangle = 1/2$. The question is how do we define the output of an ideal interferometer? Using convention-1, a unity correlation output of $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ corresponds to unity Stokes. Hence

$$\mathbf{V}_{pq} = 2 \begin{pmatrix} \langle e_x e_x^* \rangle & \langle e_x e_y^* \rangle \\ \langle e_y e_x^* \rangle & \langle e_y e_y^* \rangle \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Substituting \mathbf{B} in Eq. (2.4) the RIME for a single source of emission is given in terms of its brightness as

$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{B} \mathbf{J}_q^H. \quad (2.9)$$

An important aspect of the RIME is that every propagation effect is simply represented by a new Jones term. Hence, if a series of propagation effects corrupt the signal, then the RIME is given by

$$\mathbf{V}_{pq} = \mathbf{J}_{pm} (\dots (\mathbf{J}_{p2} (\mathbf{J}_{p1} \mathbf{B} \mathbf{J}_{q1}^H) \mathbf{J}_{q2}^H) \dots) \mathbf{J}_{qn}^H. \quad (2.10)$$

We refer to Eq. (2.10) as the onion form of the RIME and an important property of the RIME is that the number of propagation effects m and n do not have to be equal for different antennas. The order of the Jones matrices is crucial as it represents the order by which the different Jones matrices corrupt the signal. Jones matrices for effects closer to the source are the closest to the brightness matrix while the effects closer to the antennas are furthest. Furthermore, this order needs to be preserved as matrix multiplication is not in general commutative. Since signals in the sky from different sources are uncorrelated, we modify the RIME as follows to include emissions from different sources

$$\mathbf{V}_{pq} = \sum_{s=1}^{N_s} \mathbf{J}_{sp} \mathbf{B}_s \mathbf{J}_{sq}^H, \quad (2.11)$$

where \mathbf{J}_{sp} represents the propagation effects encountered by the signal from the source s on its path to the antenna p , \mathbf{B}_s is the brightness matrix of the source s and N_s is the number of sources.

2.1.2 Jones Matrices

Having derived the RIME for discrete sources, we now discuss some familiar Jones matrices and their forms. Jones matrices represent the different propagation effects a signal encounters along its path. Propagation effects are generally classified as direction dependent (DD) or direction independent (DI) based on whether they vary with direction in the sky or not. DI effects are constant across the sky and thus the same for all the sources. These are usually effects close to the antennas. DD effects vary across the sky and are effects generally close to the source of emission such as ionospheric and tropospheric effects. Jones matrices generally have different mathematical representations based on the nature of the propagation effect. The following definitions are mostly for linear feeds. For circular feeds, some of the Jones terms, particularly the rotation matrices, take slightly different forms.

Phase matrices: $\mathbf{K} = e^{-i\phi} = \begin{pmatrix} e^{-i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix}$

Such matrices are used to model propagation effects which only modify the phases of visibilities. An example is the phase delay we need to insert into the correlator to correct for the geometric path length difference between the signals measured by the antennas (see Fig. 1.4). The

phase delay is an intrinsic property of any interferometer and it is present even in the case of an ideal uncorrupted observation. From [Thompson et al. \(2017\)](#), for an antenna, p , with coordinates, $\mathbf{u}_p = (u_p, v_p, w_p)$, its phase difference relative to $\mathbf{u}_0 = (0, 0, 0)$ for a signal with direction cosines $l, m, n = \sqrt{1 - l^2 - m^2}$ is given by

$$\phi_p = \frac{2\pi}{\lambda}(u_p l + v_p m + w_p(n - 1)), \quad (2.12)$$

where λ is the signal wavelength.

Another example of phase corruption is the ionospheric phase delay ([Smirnov, 2011b](#)) caused by excess path length due to signal refraction in the atmosphere. Ionospheric phase delay is a function of electron cloud density. The ionospheric phase delay can reach up to 10^4 rad at low frequencies. For a small field of view (fov), the ionospheric corruptions can be considered constant across the fov and treated as DI effects during calibration (see [Lonsdale \(2005\)](#)).

The phase matrix is scalar since it can be parametrised as a scalar multiplied by the identity matrix. Scalar matrices have the same representation in all coordinate systems. Also, scalar matrices commute with every matrix and thus can be placed at any position in the RIME.

Rotation matrices: $\mathbf{R} = \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix}$

Certain propagation effects such as Faraday rotation or parallactic angle rotation of antenna feeds are modelled as matrix rotations. Rotation matrices are not scalar but, in 2D, they commute among themselves.

Antenna instrumental gains: $\mathbf{G} = \begin{pmatrix} g_x & 0 \\ 0 & g_y \end{pmatrix}$

can be represented by a diagonal matrix if a linear coordinate system is assumed with no polarisation. Note that this is not the case for circular feeds. Similarly to rotation matrices, diagonal matrices commute among themselves.

Fully polarised gains: $\mathbf{P} = \begin{pmatrix} g_x & g_{xy} \\ g_{yx} & g_y \end{pmatrix}$.

An example of such a Jones matrix is the feed-error matrix, $\mathbf{D} = \begin{pmatrix} 1 & d_{xy} \\ -d_{yx} & 1 \end{pmatrix}$ which represents the polarisation leakages of the antenna's feed, i.e. the sensitivity of the x feed to the y

polarisation and vice versa.

Pointing errors: All antennas have pointing errors. These generally result from the gravitational load, thermal expansion, wind pressure and errors in the antennas driving mechanism (Smirnov, 2011b). Pointing errors cause DD effects because of misalignment of the antenna's primary beam, \mathbf{E} , i.e. instead of a source at a position (l, m) to have beam gain, $\mathbf{E}(l, m)$, it has a gain $\mathbf{E}(l + \delta l, m + \delta m)$. δl and δm are called pointing offsets and ideally should be included in the RIME and solved for during calibration.

Note that certain corruptions can not be represented using Jones matrices. We refer to such corruptions as *interferometer-based errors*. These errors are generally due to the correlator or mutual coupling between antennas. Interferometer-based errors are modelled per baseline as either an element-wise multiplicative 2×2 matrix term or an additive term (Smirnov, 2011a).

2.1.3 Full-sky RIME & Van Cittert-Zernike theorem

In §2.1.1, we derived the RIME by assuming the sky consists of a discrete set of sources. In reality, the sky brightness is not discrete but continuous. We obtain the full-sky RIME by integrating the entire sky brightness distribution over a unit sphere. Consider the following discrete form for the RIME,

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\sum_{s=1}^{N_s} \mathbf{E}_{sp} \mathbf{K}_{sp} \mathbf{B}_s \mathbf{K}_{sq}^H \mathbf{E}_{sq}^H \right) \mathbf{G}_q^H, \quad (2.13)$$

where \mathbf{G}_p is the DI instrumental gain for antenna p , \mathbf{K}_{sp} denotes the phase delay matrix along the direction of the source, s , to the antenna p and \mathbf{E}_{sp} is the antenna's primary beam in the direction s for antenna p . Its' full-sky representation is given by

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\int_l \int_m \frac{1}{n} \mathbf{E}_p(l, m) \mathbf{K}_p(l, m) \mathbf{B}(l, m) \mathbf{K}_q^H(l, m) \mathbf{E}_q^H(l, m) dl dm \right) \mathbf{G}_q^H, \quad (2.14)$$

where we have replaced the discrete source direction, s , by the (l, m, n) direction cosines sky coordinates. Using Eq. (2.12) for the phase delay matrix, i.e.

$$\mathbf{K}_p(l, m) = e^{-i\phi} = e^{-\frac{2\pi}{\lambda} i(u_p l + v_p m + w_p(n-1))}$$

we get

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\int_l \int_m \frac{1}{n} \mathbf{E}_p(l, m) \mathbf{B}(l, m) \mathbf{E}_q^H(l, m) e^{-2\pi i(u_{pq}l + v_{pq}m + w_{pq}(n-1))} dldm \right) \mathbf{G}_q^H, \quad (2.15)$$

where $\mathbf{u}_{pq} = (\mathbf{u}_p - \mathbf{u}_q)/\lambda$. The dependence of Eq. (2.15) on the w -term can be removed by absorbing it in the beam term \mathbf{E} in order to turn Eq. (2.15) into a 2D Fourier transform. Let $\tilde{\mathbf{E}}_p(l, m) = \mathbf{E}_p(l, m) \mathbf{W}_p(l, m)$ where $\mathbf{W}_p(l, m) = \frac{1}{\sqrt{n}} e^{-2\pi i(w_p(n-1))}$, then we have

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\int_l \int_m \tilde{\mathbf{B}}_{pq}(l, m) e^{-2\pi i(u_{pq}l + v_{pq}m)} dldm \right) \mathbf{G}_q^H, \quad (2.16)$$

where $\tilde{\mathbf{B}}_{pq}(l, m) = \tilde{\mathbf{E}}_p(l, m) \tilde{\mathbf{B}}(l, m) \tilde{\mathbf{E}}_q(l, m)$.

Eq. (2.16) tells us that the measured visibilities as modelled by the RIME is a Fourier transform of the sky corrupted by different propagation effects. An interesting observation here is the presence of the baseline subscripts pq in the modified sky brightness $\tilde{\mathbf{B}}_{pq}(l, m)$. This tells us that each baseline sees the sky differently. Thus DD effects will make the sky appear different to each baseline. Finally, we add noise as astronomical observations are always corrupted by noise to write our final form of the RIME as

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{X}_{pq} \mathbf{G}_q^H + \mathbf{N}_{pq}, \quad (2.17)$$

where $\mathbf{X}_{pq} = \mathcal{F}(\tilde{\mathbf{B}}_{pq}(l, m))$. \mathbf{X}_{pq} is called the sky coherency whenever the only DD effect present is the phase delay, \mathbf{N}_{pq} represents Gaussian noise and \mathcal{F} denotes the Fourier transform operator.

2.2 Data Reduction Steps

Data reduction refers to the sum of all the processes which we perform on the data from the moment the instrument collects it to producing the final images required for the intended science goals. These steps range from data averaging and flagging through calibration to imaging. In this section, we discuss some traditional data reduction steps with special emphasis on calibration and its development over the years. The current trend in the field is to automate these processes as

much as possible using data reduction pipelines. Fig. 2.1 is a schematic illustration of what a typical data reduction pipeline could look like.

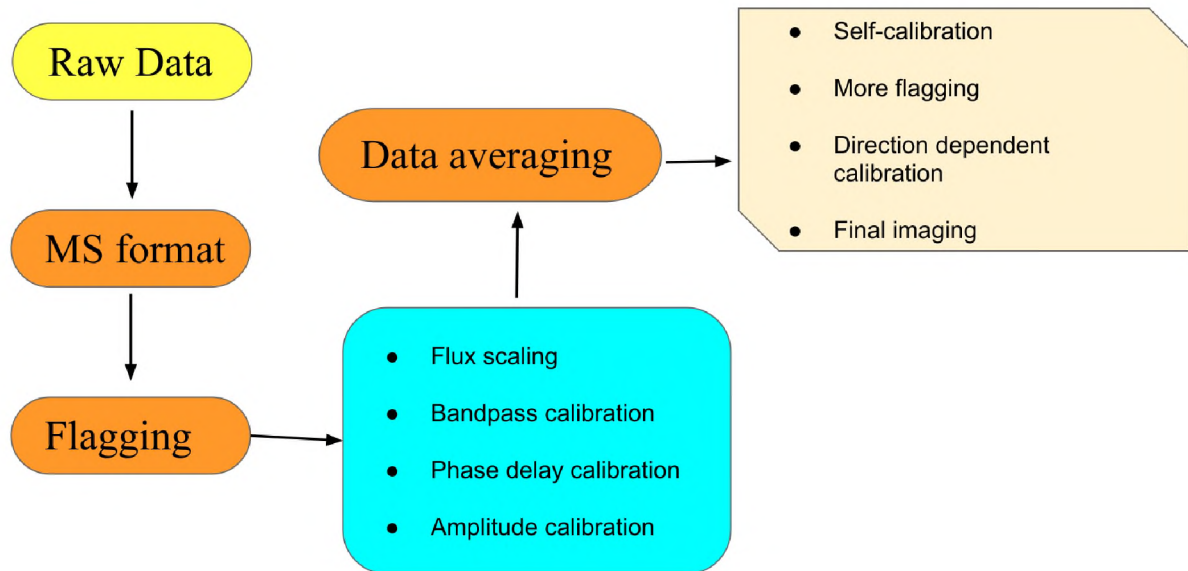


Figure 2.1: Overview of a typical radio interferometric data reduction pipeline showing the different tasks performed at the various stages.

2.2.1 Preprocessing

Preprocessing is the first step after obtaining the data. It involves understanding the data format and transferring it to a preferred data format. This may be, for example, changing from the raw UVFITS format (Chen et al., 2013) to the standard measurement set or CASA table format (van Diepen, 2015).

The next step after exporting to a desired format is usually data editing and flagging. Editing and flagging refer to removing or tagging severely corrupted data. These corruptions, called radio frequency interference (RFI), come in different flavours and correspond to all unwanted signals and even data with significant instrumental errors, such as antenna failures for example, during observations. This process can be done manually by plotting and inspecting the data or

using automated tools such as AOflagger (Offringa et al., 2012).

With the current capacities of the new generation of telescopes, the data rates are too large to be stored at full resolution. Thus after First Generation Calibration (see §2.2.2), the data is generally averaged both for computational and storage reasons. Averaging needs to be performed intelligently to minimise the amount of information loss through time and bandwidth smearing (Taylor et al., 1999), i.e. the loss in the amplitude of the signal or decorrelation due to excessive averaging. The rising trend is to employ baseline-dependent averaging (BDA) techniques (see Atemkeng et al. (2016)). Averaging can also be done after the final calibration in order to avoid smearing caused by rapidly varying corruptions. Wijnholds et al. (2018) present different BDA schemes and frameworks that can be used to average the data and transform it later to its full resolution for calibration. The averaging scheme can also be chosen depending on the science goals. For spectral line or pulsar studies, for example, we try to keep the data at the highest frequency resolution possible. In §5, we will focus on the concept of *solution intervals*, where data is averaged during calibration in order to improve the SNR for the solutions.

2.2.2 First Generation Calibration (1GC)

After preprocessing, we are ready to correct the data. Following Noordam & Smirnov (2010), calibration can be broadly divided into three stages: First Generation Calibration (1GC), Second Generation Calibration (2GC) and Third Generation Calibration (3GC). 1GC was predominantly the only calibration step until the early eighties (Noordam & Smirnov, 2010). 1GC, also known as cross-calibration, involves deriving gain solutions from a well-studied source called the calibrator and applying them to a target field. Hence, when scheduling observations, we first need to identify calibrator sources in the vicinity of the target field. During the observation, the telescopes switch between observing the target and the calibrator field. Different calibrators are used for different propagation effects, but the same calibrator can be used for different effects as well. Below we provide a list of calibration steps as well as the properties of the associated calibrators:

- **Amplitude Calibration:** Amplitude calibration, also known as flux scaling, is required to scale the fluxes of the visibilities to their true values. The calibrator used for amplitude

calibration should be very bright and unresolved in order to obtain a high signal to noise ratio in the shortest time possible to reduce the time spent on the calibrator source. The flux calibrator is usually observed for a short time at the start and the end of observation. When it is not possible to find unresolved calibrators, amplitude calibration can be performed using the system equivalent flux density of all the antennas (Janssen et al., 2019).

- **Phase or any unknown delay calibration:** Different errors cause insertion of wrong delay values during correlation, e.g. incorrect antenna positions in the correlator delay model or additional delays caused by atmospheric effects. Such errors lead to a constant in time linear phase slope as a function of frequency in the data for each baseline. The slope of the linear phase needs to be corrected for in order to avoid decorrelation of the signal during continuum imaging.
- **Bandpass calibration:** This is done to correct for the frequency response of the instrument. An ideal Bandpass calibrator should be bright with a known spectrum and unresolved or have an accurately known model. Bandpass corruptions are expected to change slowly with time, so they are solved using long solution time intervals. Phase delay and bandpass can sometimes be merged into a single step using an appropriate parametrisation.
- **Complex gain calibration:** This calibrator should be as close as possible to the target field so that it suffers approximately the same atmospheric effects. This calibrator is also called the secondary calibrator and is the most observed as it is used to track gain variations with time. The solutions obtained here are interpolated to match the target observation time before being applied.

The list above is not exhaustive, numerous other effects such as polarisation angles, astrometry and pointing errors could be corrected for during calibration. These effects vary for different arrays and fields, and the calibration strategy might slightly vary as well.

2.2.3 Second Generation Calibration (2GC)

Even after performing 1GC, the data still needs to be further calibrated for numerous reasons. Sometimes it is not possible to find good calibrator sources next to the target field for 1GC. Hence the 1GC solutions are usually not applicable. Secondly, the calibrator and target fields are observed in different scans; therefore, rapid gain variations are not captured by the 1GC solutions. Second Generation Calibration (2GC), often referred to as self-calibration, entails using the target field to calibrate itself. During 2GC the type of Jones matrices we generally solve for highly depends on the array and the frequency of the observation. For example, at low frequencies, ionospheric effects are very dominant. 2GC methods, like 1GC, correct for direction independent effects. Hence the measured visibilities at a time, t , for the scalar case (single correlation) can be written as

$$\tilde{\mathbf{v}}_{pq}(t) = \mathbf{g}_p(t)\mathbf{v}_{pq}(t)\mathbf{g}_q^*(t) + \epsilon_{pq}(t), \quad (2.18)$$

where $\tilde{\mathbf{v}}_{pq}$ denotes the measured visibilities, \mathbf{v}_{pq} the true sky visibilities, \mathbf{g}_p is the direction independent gain and ϵ_{pq} is the additive Gaussian noise. 2GC can generally be performed using *closure relations* or a direct optimisation approach.

The first approach that was used when 2GC was invented was the closure relations approach and the method was called *hybrid mapping* (Cornwell & Wilkinson, 1981). Dropping the time dependence and taking the phase of Eq. (2.18) we have

$$\tilde{\phi}_{pq} = \phi_{pq} + \theta_p - \theta_q + \text{noise term}, \quad (2.19)$$

where $\tilde{\mathbf{v}}_{pq} = |\tilde{\mathbf{v}}_{pq}|e^{-i\tilde{\phi}_{pq}}$, $\mathbf{v}_{pq} = |\mathbf{v}_{pq}|e^{-i\phi_{pq}}$, $\mathbf{g}_p = |\mathbf{g}_p|e^{-i\theta_p}$, and $\mathbf{g}_q = |\mathbf{g}_q|e^{-i\theta_q}$. The lower case symbols are used here to indicate scalar calibration where each visibility is a single complex number and not a 2×2 matrix as in the fully polarised case¹. If we consider three antennas, p , q and r say, and compute the sum of their phases in a closed loop, we obtain the following

¹This convention is used throughout all the remaining sections. i.e. if \mathbf{Z} is 2×2 visibility or Jones matrix in the fully polarised case, then its scalar counterpart is z .

expression

$$\tilde{C}_{pqr} = \tilde{\phi}_{pq} + \tilde{\phi}_{qr} + \tilde{\phi}_{rp}, \quad (2.20)$$

$$= (\phi_{pq} + \theta_p - \theta_q) + (\phi_{qr} + \theta_q - \theta_r) + (\phi_{rp} + \theta_r - \theta_p) + \text{noise term}, \quad (2.21)$$

$$= \phi_{pq} + \phi_{qr} + \phi_{rp} + \text{noise term}, \quad (2.22)$$

$$= C_{pqr} + \text{noise term}. \quad (2.23)$$

\tilde{C}_{pqr} is called the closure phase and Eq. (2.23) tells us that for any 3 baselines forming a closed loop, the gains' phases do not contribute to the closure phase. The closure phase of the measured and modelled visibilities only deviate by some additive noise term. A similar closure amplitude exists by considering the amplitudes of 4 baselines in a loop. For 4 antennas, p , q , r and s say, we have

$$\Gamma_{pqrs} = \frac{|\tilde{v}_{pq}| |\tilde{v}_{rs}|}{|\tilde{v}_{pr}| |\tilde{v}_{qs}|}, \quad (2.24)$$

where Γ_{pqrs} is the closure amplitude. Initially closure phases were not very useful but with the advent of fast computers in the eighties, numerous techniques were developed to make radio maps with consistent closure phases and amplitudes (examples are [Readhead & Wilkinson \(1978\)](#) and [Ekers \(1984\)](#)). The key ideas of closure phases revolves around the following procedure ([Taylor et al., 1999](#))

1. Make an initial model of the target field or source.
2. Find all the closed loops baselines and compute their closure phases. Derive the true phases for two antennas using the model closure phase and the phase of the last antenna in the loop from the measured data closure phase.
3. Form a new model from the image of the visibility amplitudes and use that to derive visibility phases.
4. Repeat from step 2 until we are satisfied.

Various variants of this approach exist and later techniques were suggested on how to treat the additive noise term. Nowadays, self-calibration corresponds to an approach completely different

from this approach but closure quantities are still widely used for redundant arrays (i.e. array with a large number of baselines measuring the same information) such as HERA (DeBoer et al., 2017) under the name *redundant self-calibration* (see Marthi & Chengalur (2013) and Grobler et al. (2018)).

The direct optimisation approach which is currently the most widely used 2GC approach, is called self-calibration. It involves finding DI complex gains that minimise the square of residuals between the measured and modelled visibilities,

$$\min_{\mathbf{g}(t)} \sum_{pq, t|p < q} |\tilde{\mathbf{v}}_{pq}(t) - \mathbf{g}_p(t)\mathbf{v}_{pq}(t)\mathbf{g}_q^*(t)|^2. \quad (2.25)$$

This is possible because for an array with N_a antennas, at each time and frequency index we have N_a unknown parameters and $N_{bl} = \frac{N_a(N_a - 1)}{2}$ visibilities. Hence Eq. (2.25) is an overdetermined system of equations and can be solved if we have a good enough model for \mathbf{v}_{pq} . Self-calibration proceeds as follows:

1. Make an image of your 1GC calibrated data.
2. Make a sky model from the 1GC calibrated image by running a source finder such as PyBDSF (Mohan & Rafferty, 2015) or using the model components from your imaging tool.
3. Calibrate the data using the sky model from 2.
4. Compute new corrected visibilities and image them.
5. Check your corrected visibilities and residuals and if satisfied stop or repeat from step 2.

Self-calibration has proven to be quite successful over the years by considerably improving dynamic range² of radio maps (Noordam & Smirnov, 2010).

²Dynamic range, $DR = \frac{\text{Image peak}}{\text{Image rms}}$ is a common metric used to check how good calibration was in radio interferometry. Note that dynamic range can also be defined as the ratio of the image peak to brightest artefact in the image.

2.2.4 Third Generation Calibration (3GC)

Radio interferometry has now transitioned to the 3GC era. Both 1GC and 2GC only correct for DI effects, but with the sensitivities and the wide-field imaging capabilities of the current and future generation telescopes (such as MeerKAT, the SKA and the Expanded Very Large Array (EVLA)), DD effects have become more prominent and need to be properly addressed. A classical DD problem will have its RIME formulated as follows

$$\mathbf{V}_{pq} = \sum_{s=1}^{N_g} \mathbf{J}_{sp} \mathcal{F}(\mathbf{B}_{pqs}) \mathbf{J}_{sq}^H, \quad (2.26)$$

where each variable is defined as before. Because DD effects are different per direction in the sky, gains need to be solved separately for each direction. DD effects can be corrected either in the image domain or in the visibility domain. Solving for DD effects in the visibility domain entails dividing the sky into numerous directions or sources. This sometimes makes the problem intractable as we can not solve for one gain at every sky position. For example, faceting could be used where a wide-field of view is approximated with many small narrow field images for which the DD effects can be assumed to be constant (see [Van Weeren et al. \(2016\)](#) and [Tasse et al. \(2018\)](#)). Two common approaches are generally used for tackling DD effects namely *Physics-based approaches* and *Heuristic approaches*.

Physics-based approaches model the underlying phenomenon causing the DD effect. [Tasse \(2014b\)](#) for example is a calibration scheme which can be used to solve directly for physical quantities such as clock drifts or total electron content (TEC) that appear in the RIME instead of their resulting ionospheric Jones matrices. Such an approach significantly reduces the number of free parameters, thereby improving the conditioning of the problem. Others include modelling and applying the primary beam in the image plane and solving for antenna pointing errors (see for example [Bhatnagar et al. \(2004\)](#)).

Whenever no physics based model exists for the specific DD effect, the most affected sources are identified and complex gains solved for them. One of the first approaches proposed for this was *Peeling* ([Noordam, 2004](#)). Peeling solves for DD effects by correcting the effects towards each source separately in decreasing order of brightness. Every time a source is corrected, it

is removed from data and the process is repeated with the next brightest source. Peeling is a computationally expensive process, and the gain solutions are usually contaminated by 1GC and 2GC errors which are locked in as discussed by [Smirnov \(2011b\)](#). [Smirnov \(2011c\)](#) proposed the *differential gains* approach which reduces gain contamination by solving for the DI and DD effects simultaneously by using the following RIME model

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\sum_{s=1}^{N_s} \Delta \mathbf{x}_{sp} \mathcal{F}(\mathbf{B}_{pqs}) \Delta \mathbf{x}_{sq}^H \right) \mathbf{G}_q^H, \quad (2.27)$$

where $\Delta \mathbf{x}_{sp}$ is the DD differential gain for antenna p in the direction s .

2.3 Conventional Calibration Algorithms

In the preceding three sections, we described calibration without emphasis on the exact mathematical computations or expressions required to compute the gains from the measured and modelled visibilities. Following the RIME, one defines calibration as an optimisation process where we construct a model and try to find the optimal gains which fit the model. We discuss here a few traditional calibration algorithms for 2GC and 3GC.

2.3.1 Schwab & Thompson-D'Addario method

One of the first formulations of calibration as an optimisation problem was made by [Schwab \(1980\)](#). [Schwab \(1980\)](#) formulated the calibration problem using both l_1 and l_2 minimisation. We describe the l_2 minimisation approach here since it is the most widely used approach. [Schwab \(1980\)](#) writes the calibration problem as

$$S = \sum_k \sum_{pq, t_k | p < q} w_{pq}(t_k) |\tilde{\mathbf{v}}_{pq}(t_k) - \mathbf{g}_p(t_k) \mathbf{v}_{pq}(t_k) \mathbf{g}_q^*(t_k)|^2, \quad (2.28)$$

where we seek to find the gains, \mathbf{g} , that minimise S , $w_{pq} = \frac{1}{\sigma_{pq}}$ represents the weights for each visibilities computed from its variance σ_{pq} , $\tilde{\mathbf{v}}_{pq}$ denotes the corrupted visibilities and \mathbf{v}_{pq} are the modelled visibilities. Note the presence of the k subscript on the time index to incorporate *solution intervals* (see §5). One of the key steps to solving this problem was suggested by [Thompson](#)

& d'Addario (1982), i.e. to divide all through by the modelled visibilities, thus rewriting Eq. (2.28) as

$$S = \sum_k \sum_{pq, t_k | p < q} w_{pq}(t_k) |\mathbf{v}_{pq}(t_k)|^2 |\tilde{\mathbf{x}}_{pq}(t_k) - \mathbf{g}_p(t_k) \mathbf{g}_q^*(t_k)|^2, \quad (2.29)$$

where $\tilde{\mathbf{x}}_{pq}(t_k) = \frac{\tilde{\mathbf{v}}_{pq}(t_k)}{\mathbf{v}_{pq}(t_k)}$. Eq. (2.29) is identical to calibrating a single point source. Separating the problem into real and imaginary parts, the solution for the complex gain as quoted from Schwab (1980) is given by

$$\mathbf{g}_p(t_k) = \frac{\sum_{pq, t_k | p < q} w_{pq}(t_k) \tilde{\mathbf{x}}_{pq}(t_k) \mathbf{g}_q(t_k)}{\sum_{pq, t_k | p < q} w_{pq}(t_k) \mathbf{g}_q(t_k)}. \quad (2.30)$$

The gains are computed iteratively with updates given by Eq. (2.30).

2.3.2 Non Linear Least Squares (NLLS) methods

The standard approach by most current calibration packages nowadays is to employ NLLS algorithms such as the Levenberg-Marquardt (LM) and the Gauss-Newton (GN) (see Madsen et al. (2004)). Let's start by modifying the optimisation problem, Eq. (2.29) as follows

$$S = \sum_k \sum_{pq, t_k | p < q} w_{pq}(t_k) |\mathbf{d}_{pq}(t_k) - \mathbf{v}_{pq}(t_k)|^2, \quad (2.31)$$

where \mathbf{d}_{pq} represents the measured visibilities and \mathbf{v}_{pq} denotes the chosen RIME model. If we drop the weights, w_{pq} , and the time indices, t_k , and stack the data and model visibilities into vectors, i.e.

$$\mathbf{d} = [\mathbf{d}_{pq}], \text{ and, } \mathbf{v} = [\mathbf{v}_{pq}], \quad (2.32)$$

where $[\]$ denotes a vector of stacked visibilities for all the baselines, then the optimisation problem can be rewritten as

$$\min_{\mathbf{g}} \|\mathbf{r}\|_F^2 = \min_{\mathbf{g}} \|\mathbf{d} - \mathbf{v}\|_F^2, \quad (2.33)$$

where \mathbf{g} is a vector containing the gains for all the antennas, \mathbf{r} are the stacked residual visibilities and $\|\cdot\|_F$ denotes the Frobenius norm. In order to solve Eq. (2.33) using a NLLS method, we need to compute the Jacobian matrix, \mathbf{J} defined as

$$\mathbf{J}_{ij} = \frac{\partial \mathbf{v}_i}{\partial \mathbf{g}_j}. \quad (2.34)$$

Because the gains are complex it is not possible to compute the derivatives in Eq. (2.34) using conventional calculus³. The problem is solved by splitting the data and all the variables involved in Eq. (2.33) into their real and imaginary parts. In other words, we transform the n complex-valued variables problem to a $2n$ real-valued variables problem. Thus we construct the following augmented vectors $\check{\mathbf{d}}$, $\check{\mathbf{v}}$, $\check{\mathbf{r}}$ and $\check{\mathbf{g}}$ for the data, model, residuals and gains respectively

$$\check{\mathbf{d}} = \begin{bmatrix} \mathbf{d}^R \\ \mathbf{d}^I \end{bmatrix}, \quad \check{\mathbf{v}} = \begin{bmatrix} \mathbf{v}^R \\ \mathbf{v}^I \end{bmatrix}, \quad \check{\mathbf{r}} = \begin{bmatrix} \mathbf{r}^R \\ \mathbf{r}^I \end{bmatrix}, \quad \check{\mathbf{g}} = \begin{bmatrix} \mathbf{g}^R \\ \mathbf{g}^I \end{bmatrix}, \quad (2.35)$$

where the superscripts R and I denotes the real and imaginary components of a complex number respectively. Using these augmented vectors, the optimisation problem is reformulated as

$$\min_{\check{\mathbf{g}}} \|\check{\mathbf{r}}\|_F^2 = \|\check{\mathbf{d}} - \check{\mathbf{v}}\|_F^2, \quad (2.36)$$

where it is now possible to compute the Jacobian matrix, \mathbf{J} where

$$\mathbf{J}_{ij} = \frac{\partial \check{\mathbf{v}}_i}{\partial \check{\mathbf{g}}_j}. \quad (2.37)$$

The GN and LM algorithms solves the problem iteratively using a gradient descent approach with the update step, $\Delta\check{\mathbf{g}}$ given by (see Appendix A for a review on the LM and the GN algorithms)

$$GN; \quad \Delta\check{\mathbf{g}} = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \check{\mathbf{r}}, \quad (2.38)$$

$$LM; \quad \Delta\check{\mathbf{g}} = -(\mathbf{J}^T \mathbf{J} + \mu \text{Diag}(\mathbf{J}^T \mathbf{J}))^{-1} \mathbf{J}^T \check{\mathbf{r}}, \quad (2.39)$$

where μ is the LM damping factor. The gains at the k -iteration, $\check{\mathbf{g}}^{[k]}$ are computed as

$$\check{\mathbf{g}}^{[k]} = \check{\mathbf{g}}^{[k-1]} + \alpha \Delta\check{\mathbf{g}}, \quad (2.40)$$

where α is the learning rate used to control the step size at every iteration. The exact forms of Eq. (2.37) and Eq. (2.40) depends on the RIME model. For DI calibration, for example, we have

$$\mathbf{v}_{pq} = \mathbf{g}_p \mathbf{m}_{pq} \mathbf{g}_q^*, \quad (2.41)$$

³Since all real-valued functions of complex variables are not necessarily holomorphic.

where \mathbf{m}_{pq} are the modelled visibilities. If we let $\mathbf{v}_{pq} = v_{pq}^R + iv_{pq}^I$, $\mathbf{m}_{pq} = m_{pq}^R + im_{pq}^I$, $\mathbf{g}_p = g_p^R + ig_p^I$ and $\mathbf{g}_q = g_q^R + ig_q^I$, then Eq. (2.41) can be rewritten as

$$\mathbf{v}_{pq} = (g_p^R + ig_p^I)(m_{pq}^R + im_{pq}^I)(g_q^R - ig_q^I), \quad (2.42)$$

$$\begin{aligned} &= (g_p^R m_{pq}^R g_q^R - g_p^I m_{pq}^I g_q^R + g_p^R m_{pq}^I g_q^I + g_p^I m_{pq}^R g_q^I) \\ &\quad + i(g_p^R m_{pq}^I g_q^R + g_p^I m_{pq}^R g_q^R - g_p^R m_{pq}^R g_q^I + g_p^I m_{pq}^I g_q^I). \end{aligned} \quad (2.43)$$

We can show that the Jacobian matrix, \mathbf{J} , for this problem will consist of 4 matrix blocks given by

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_R^R & \mathbf{J}_I^R \\ \mathbf{J}_R^I & \mathbf{J}_I^I \end{bmatrix} \quad \text{where} \quad \begin{cases} [\mathbf{J}_R^R]_{ab} = \left[\frac{\partial v_a^R}{\partial g_b^R} \right] \\ [\mathbf{J}_I^R]_{ab} = \left[\frac{\partial v_a^R}{\partial g_b^I} \right] \\ [\mathbf{J}_R^I]_{ab} = \left[\frac{\partial v_a^I}{\partial g_b^R} \right] \\ [\mathbf{J}_I^I]_{ab} = \left[\frac{\partial v_a^I}{\partial g_b^I} \right] \end{cases}. \quad (2.44)$$

Furthermore, using Eq. (2.43), we can give the explicit forms of the different blocks of \mathbf{J} as

$$[\mathbf{J}_R^R]_{ab} = \left[\frac{\partial v_{a \rightarrow pq}^R}{\partial g_b^R} \right] = \begin{cases} m_{pq}^R g_q^R + m_{pq}^I g_q^I & \text{if } b = p, \\ g_p^R m_{pq}^R - g_p^I m_{pq}^I & \text{if } b = q, \end{cases} \quad (2.45)$$

$$[\mathbf{J}_I^R]_{ab} = \left[\frac{\partial v_{a \rightarrow pq}^R}{\partial g_b^I} \right] = \begin{cases} -m_{pq}^I g_q^R + m_{pq}^R g_q^I & \text{if } b = p, \\ g_p^R m_{pq}^I + g_p^I m_{pq}^R & \text{if } b = q. \end{cases} \quad (2.46)$$

$$[\mathbf{J}_R^I]_{ab} = \left[\frac{\partial v_{a \rightarrow pq}^I}{\partial g_b^R} \right] = \begin{cases} m_{pq}^I g_q^R - m_{pq}^R g_q^I & \text{if } b = p, \\ g_p^R m_{pq}^I + g_p^I m_{pq}^R & \text{if } b = q, \end{cases} \quad (2.47)$$

$$[\mathbf{J}_I^I]_{ab} = \left[\frac{\partial v_{a \rightarrow pq}^I}{\partial g_b^I} \right] = \begin{cases} m_{pq}^R g_q^R + m_{pq}^I g_q^I & \text{if } b = p, \\ -g_p^R m_{pq}^R + g_p^I m_{pq}^I & \text{if } b = q, \end{cases} \quad (2.48)$$

where $[\mathbf{J}]_{ab} = 0$ whenever b is neither equal to p nor q . In a similar way, following some rigorous algebra, we can write down the exact analytic expressions for the GN and LM update equations (2.38) and (2.39). Although we used a scalar case for simplicity here, the fully polarised case will be solved in the same way by carefully constructing the augmented vectors, i.e. first vectorising the 2×2 matrices before splitting them into their real and imaginary components. Hence each 2×2 matrix will contribute 8 elements to its corresponding augmented vector.

2.3.3 StEFCal

StEFCal is an alternating direction implicit method for solving DI gains. Mitchell et al. (2008) originally proposed the idea, but it became popular after its re-derivation and implementation by Salvini & Wijnholds (2014). We reformulate the optimisation problem as follows

$$\mathbf{g} = \min_{\mathbf{g}} \|\tilde{\mathbf{V}} - \mathbf{G}\mathbf{V}\mathbf{G}^H\|_F^2 \quad (2.49)$$

where $\tilde{\mathbf{V}}$ is an $N_a \times N_a$ matrix representing the observed or measured visibilities, \mathbf{V} is an $N_a \times N_a$ matrix denoting the model visibilities, \mathbf{G} is an $N_a \times N_a$ diagonal matrix whose elements are the antenna gains, \mathbf{g} .

StEFCal solves for \mathbf{G}^H by assuming \mathbf{G} is known, and likewise \mathbf{G} is computed by fixing \mathbf{G}^H . Since $\Delta = \tilde{\mathbf{V}} - \mathbf{G}\mathbf{V}\mathbf{G}^H$ is hermitian, the two steps are in fact identical and thus during each iteration only one of either \mathbf{G} or \mathbf{G}^H is computed. Therefore at the i -iteration we solve for

$$\mathbf{G}^{[i]} = \min_{\mathbf{G}^{[i]}} \|\tilde{\mathbf{V}} - \mathbf{G}^{[i-1]}\mathbf{V}\mathbf{G}^{[i]}\|_F^2. \quad (2.50)$$

If we define $\mathbf{Z}^{[i]} = \mathbf{G}^{[i]}\mathbf{V}$, then $\Delta = \|\tilde{\mathbf{V}} - \mathbf{Z}\mathbf{G}^H\|_F^2 = \sqrt{\|\sum_{i=1}^p \tilde{\mathbf{V}}_{:,p} - \mathbf{Z}_{:,p}\mathbf{g}_p^*\|_F^2}$

where $\{\cdot\}_{:,p}$ denotes the p^{th} column of the matrix $\{\cdot\}$. Using the normal equation method the gains at the i^{th} iteration are readily given by

$$\mathbf{g}_p^{[i]} = \left\{ \frac{(\mathbf{Z}_{:,p}^{[i-1]})^H \cdot \tilde{\mathbf{V}}_{:,p}}{(\mathbf{Z}_{:,p}^{[i-1]})^H \cdot \mathbf{Z}_{:,p}^{[i-1]}} \right\}^* \quad (2.51)$$

StEFCal provides a considerable computational advantage over most NLLS calibration algorithms which scale as $\mathcal{O}(P^3)$ while StEFCal scales as $\mathcal{O}(P^2)$ where P is the number of free

parameters. There have been extensions to StEFCal to include DD calibration and polarisation (see [Salvini & Wijnholds \(2014\)](#) for more details).

2.4 Calibration Bottlenecks and Novel Calibration Algorithms

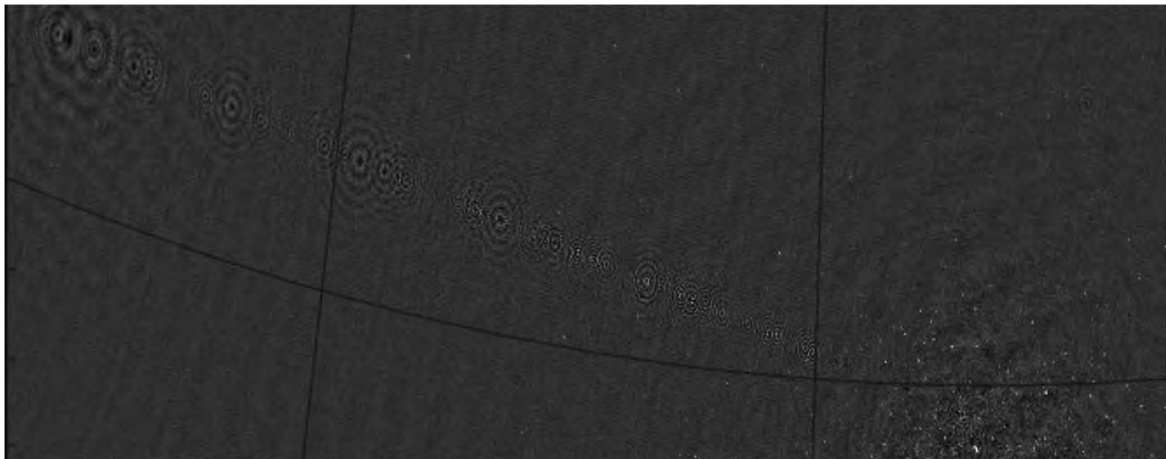
In practice, calibration is not a straightforward task. Several bottlenecks significantly limit the performance of conventional calibration algorithms and cause imperfections in the output images called calibration artefacts. These imperfections generally manifest in the images as spurious source components, deformations in the structures of extended sources and suppression of real emissions ([Linfield, 1986](#); [Wilkinson et al., 1988](#); [Martí-Vidal & Marcaide, 2008](#)). In this section, we describe some of these bottlenecks and a few frameworks which can help improve traditional algorithms.

2.4.1 Calibration Bottlenecks

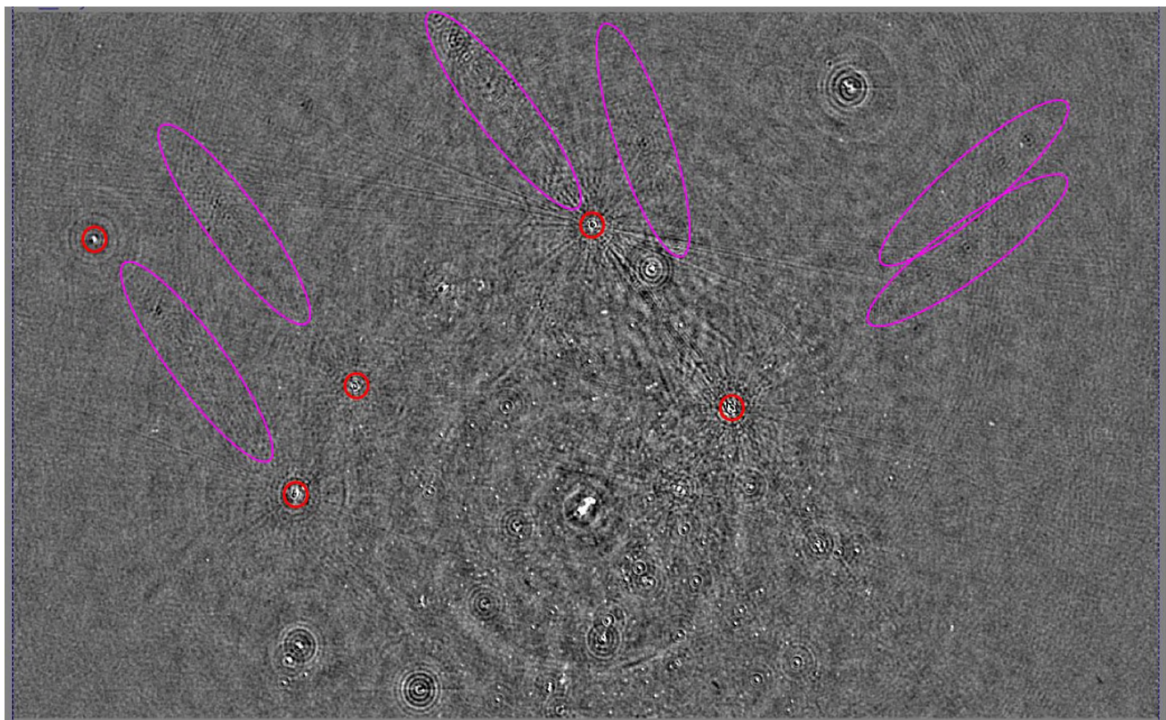
Incomplete sky models

2GC and 3GC calibration necessitate that we build our calibration model from the imperfectly calibrated data. The idea is to start with an initial sky model containing only the bright emissions and to progressively improve it in successive calibration runs by including fainter ones. During this process, most faint emissions are generally not included. The unmodelled sources, as studied by [Grobler et al. \(2014\)](#); [Wijnholds et al. \(2016\)](#); [Grobler et al. \(2016\)](#); [Patil et al. \(2016\)](#) and [Barry et al. \(2016\)](#), will have their fluxes either suppressed or overestimated by the calibration. Flux suppression is a major bottleneck to the calibration process, as many interesting science results from weak signals, and it is practically impossible to have complete sky models. [Fig. 2.2a](#) shows one of the first and most striking sightings of ghost sources in real data. After calibration of a 92-cm Westerbork Synthesis Radio Telescope (WSRT) observation of J1819+3845 in 2004, the output image had numerous negative [point spread function like] artefacts. These artefacts appeared in a regular pattern along the line between the brightest source in the field and a distant bright source (Cygnus A) well away from the target field. Following investigations with the

“Quality Monitoring Committee project” (QMC), [Smirnov \(2011\)](#) observed similar artefacts in different observations and simulations calibrated with incomplete sky models and having large pointing errors. Fig. [2.2b](#) show similar artefacts to those in Fig. [2.2a](#) from the QMC project.



(a) Ghost sources in a 92-nforccm Westerbork Synthesis Radio Telescope (WSRT) observation of J1819+3845. The string of circular patterns are negative ghost sources. These artefacts occurred along the line joining the brightest source in the field (bottom right of the image) and Cygnus A which is 20 deg away from the field (outside the image on the top left) (Grobler et al., 2014).



(b) Ghost sources in a residual image from the Quality Monitoring Committee project. The red circles indicate the positions of model sources which have been subtracted from the data. The blue ellipse indicate regions with the faint ghost sources.

Figure 2.2: Ghost sources in Westerbork Synthesis Radio Telescope (WSRT) observations.

RFI

As discussed earlier, an important component of any data reduction pipeline is flagging, i.e. detection and excision of severely corrupted visibilities. As we will discuss throughout the thesis, RFI comes in numerous forms, ranging from different electronics systems to emissions from contaminating astrophysical sources such as the moon and the sun, or even resulting radio emission accidentally generated by people working at observatories. State-of-the-art RFI flaggers such as [Offringa et al. \(2012\)](#) use the statistics of the visibilities to flag outliers. Flagging is a complex task and usually most ultra-faint RFI evades the flagging tool. Unflagged RFI is a major problem for calibration and imaging as this limits SNR and dynamic range. We expand on this issue in the upcoming chapters.

Inaccurate modelling of the different propagation effects

Wrong models for propagation effects such as antennas' primary beams, antennas' feeds, correlator errors, antenna pointing direction and ionospheric effects will lead to different sorts of calibration artefacts. The primary beam, for example, is particularly challenging to model and usually leads to severe DDEs. Most of the current primary beam models used for calibration are only accurate within their main lobe, thus can not be used to correct for sources outside the main lobe. Furthermore, primary beam effects are coupled with pointing errors (see §2.1.2), making it even more complicated to model. As recently shown by [Iheanetu et al. \(2019\)](#), reflector antennas like VLA's antennas have a standing wave effect which causes small spectral variations in the primary beam. Though these variations appear to be small, they can significantly reduce the dynamic range of images if we do not model them correctly.

In addition to the above bottlenecks, calibration algorithms are also suffering from the high data rates of new telescopes. Astronomers want to have calibration solvers that produce images that are an accurate representation of the sky. However, they also want these solvers to process massive datasets very rapidly, i.e. if possible, perform real-time data reductions, and store only the corrected images. The latter is not always possible because having fast calibration algorithms generally requires us to make different kinds of assumptions on the data and propagation effects

which can lead to various types of artefacts. For example, most of the time when performing DD calibration, we generally have to divide the sky or select specific sources for which we solve the DDEs. Such decisions, if not carefully taken, can have severe effects on the output images. Another such decision is the choice of the length of the solution intervals to use during calibration. As we will describe in §5, while solution intervals are an excellent regularising tool for calibration, inadequate solution interval width can significantly affect the output images from calibration and the indented science studies. While experienced astronomers are usually capable of making sound decisions during calibration, we need to develop robust and efficient calibration solvers with significant automation to facilitate the process of calibration for young astronomers with the capability of handling the massive data rates of new telescopes.

2.4.2 Novel Calibration Algorithms

In the next chapters, we will present our research on how to mitigate against some of these calibration bottlenecks for the robust calibration of radio interferometers. Before moving on to that, we conclude this chapter with a brief discussion of some of the recent algorithms and frameworks which have been suggested by different authors to improve calibration.

Regularized Maximum Likelihood algorithms

In recent years, a few regularized maximum likelihood (RML) algorithms have been applied in radio interferometry using ideas from the field of compressed sensing (Candes et al., 2006). Initial applications were for image deconvolution leading to the development of packages such as PURIFY (Carrillo et al., 2014; Pratley et al., 2017). Because the deconvolution problem is ill-posed, it needs to be regularised in order to be solved. RML methods solves this problem by minimising, for example, the following objective function

$$\min_{\mathbf{x}} \mathbf{h}(\mathbf{x}) + \beta \|\mathbf{x}\|_1, \quad (2.52)$$

where $\mathbf{h}(\mathbf{x})$ is a standard NLLS objective function called the data fidelity term and $\|\mathbf{x}\|_1$ is the regularizer. The regularizer, $\|\mathbf{x}\|_1$, defined using an l_1 -norm is a sparsity promoting prior and

β is a chosen regularization parameter which balances between the data fidelity term and the regularizer. Since regularizers are not always smooth functions, such optimisation problems are solved using forward-backward propagation (Combettes & Wajs, 2005). DD calibration can also be ill-posed, for example, whenever the number of parameters we are solving for exceeds the number of data points, i.e.

$$N_a \frac{N_a - 1}{2} N_t N_\nu < N_d N_a \frac{N_t N_\nu}{\Delta t \Delta \nu},$$

where N_a is the number of antennas, N_d is the number of directions, N_t and N_ν and the number of times and frequencies, respectively, and Δt and $\Delta \nu$ are the chosen time and frequency solution intervals. RML methods have now been extended to jointly solve both the calibration and imaging problem (see Kazemi et al. (2015), Repetti et al. (2017), (Chiarucci & Wijnholds, 2017), ?). This is done by Repetti et al. (2017), for example, by adding an extra term to the objective function, Eq. 2.52, in order to enforce smoothness on the solved DD gains.

Bayesian methods

RML is limited in that it provides a maximum a posteriori (MAP) estimate for a fixed regularisation parameter β . Bayesian methods allow for a full posterior inference as well as the possibility of inferring the parameters of the prior from the marginal likelihood or evidence.

A promising approach that has been suggested for addressing the problem of outliers in the data (i.e. incomplete sky models and unflagged RFI during calibration) is the use of heavy-tailed distributions for the noise during calibration instead of a Gaussian distribution used by most algorithms (see Kazemi & Yatawatta (2013) and Ollier et al. (2017)). These algorithms have shown their superiority compared to conventional least squares methods in terms of reducing the amount of flux suppressed from unmodelled sources. These algorithms are generally implemented using Bayesian methods where we assign prior distributions to the data weights and solve for the gains and parameters of these prior distributions by maximising the Bayesian evidence of the resulting posterior distribution. We will elaborate more on this approach in §4 and §5 when we describe the *robust* solver and its implementation.

BIRO (Lochner et al., 2015) is another example of the application of Bayesian methods in radio interferometry. BIRO relies on the software package Montblanc (Perkins et al., 2015). Montblanc is a tool which uses GPU acceleration to compute the RIME efficiently. By combining Montblanc and Multinest (Buchner, 2016), BIRO makes full parametrised Bayesian inference feasible, but this is still computationally far more expensive than getting MAP solutions. The reason is that we need to perform a discrete model selection which requires sampling and computing the evidence of each model. These algorithms also suffer from the curse of dimensionality, where it is not practical to sample the full posterior distribution.

Arras et al. (2019) recently showed that it is possible to use Information Field Theory (IFT) algorithms for RI calibration and imaging. IFT techniques fall into a class of approximate Bayesian inference, where we use a form of variational inference to approximate the posterior distribution around MAP solutions in order to compute approximate uncertainty bounds. Arras et al. (2019) specifically uses Gaussian Processes with smooth power spectrums as priors.

Consensus or Stochastic optimisation

Stochastic optimisation algorithms are required when the full data cannot fit into memory at once. A typical scenario where this can happen is when using parametric models during calibration. For example, if we are modelling gains using a polynomial function of frequency, we will need to have a large bandwidth of data in memory to be able to compute correct coefficients. Recently, Yatawatta (2015a) and Yatawatta et al. (2017) showed that parametric models could be implemented efficiently for a large dataset using consensus optimisation. In this framework, the data are split into small chunks which can fit into the memory, and the different chunks are calibrated separately using so-called compute agents. Next, the solutions from the different compute agents are combined at a fusion centre where the desired parametric model is enforced.

Another framework suggested for developing scalable RML algorithms is using primary-dual (Komodakis & Pesquet, 2015) methods. This framework splits the optimisation problem into smaller problems, each for every term in the RML objective function. The individual smaller problems are solved in parallel by simultaneously solving for a dual formulation of the original

problem. [Onose et al. \(2016\)](#) present a good review on the application of these methods in radio interferometric algorithms with novel data structures for their implementations.

Calibration using a Complex Student-t distribution and Wirtinger derivatives¹

As discussed in Chapter 2, radio interferometric gain calibration can be biased by incomplete sky models and RFI, resulting in calibration artefacts that limits the dynamic range of the output images and causes suppression in the fluxes of the sources. In this chapter, we present a calibration algorithm based on a Student's t-distribution which leverages the framework of complex optimisation and Wirtinger calculus for efficient and robust interferometric gain calibration. The implemented algorithm is an extension to the algorithm derived in [Kazemi & Yatawatta \(2013\)](#) and it is integrated as an option in the calibration software package, CubiCal ([Kenyon et al., 2018](#)). We begin this chapter by providing a brief introduction to Wirtinger calculus and discuss how to apply it in the context of RI calibration in §3.1. The details and implementation of the new algorithm are presented in §3.2.

3.1 Calibration as a complex optimisation problem

In §2.3.2, we described how calibration can be performed using NLLS methods. In order to apply these methods we had to split the data and the different propagation effects into their real and imaginary components to circumvent taking complex derivatives. Current developments in the field of complex optimisation allow bypassing this data transform (see for example [Kreutz-](#)

¹The work presented in this chapter was originally published as part of [Sob et al. \(2019\)](#)

Delgado (2009) and Sorber et al. (2012)) by using Wirtinger derivatives (Wirtinger, 1927). In this section, we present the Wirtinger formalism used to express calibration as complex optimisation problem, and the calibration suite in which the new solver is implemented. A more extensive description of Wirtinger calculus and its applicability for RI calibration is available in Smirnov & Tasse (2015) and Kenyon et al. (2018).

3.1.1 The Wirtinger approach

Wirtinger calculus relies on treating the complex variable z and its conjugate counterpart z^* as independent variables. Wirtinger derivatives are defined as

$$\frac{\partial}{\partial z} = \frac{1}{2} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right), \quad \frac{\partial}{\partial z^*} = \frac{1}{2} \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right), \quad (3.1)$$

where $z = x + iy$, $\frac{\partial z}{\partial z^*} = 0$ and $\frac{\partial z^*}{\partial z} = 0$.

Consider the following optimisation problem

$$\min_z \|\mathbf{r}(z, z^*)\|_F^2 = \min_z \|\mathbf{d} - \mathbf{v}(z, z^*)\|_F^2, \quad (3.2)$$

where \mathbf{r} , \mathbf{d} , and \mathbf{v} are complex variables and $\|\cdot\|_F$ is the Frobenius norm. This problem is solved by simply extending any NLLS algorithm such as the LM and the GN (see Madsen et al. (2004)) to use Wirtinger derivatives. By treating z and z^* as independent variables, we construct the following augmented vector for the unknown parameters

$$\check{\mathbf{z}} = \begin{bmatrix} z \\ z^* \end{bmatrix}. \quad (3.3)$$

Furthermore, we augment all the functions with their conjugates. Hence, we have the following for residuals, data and model respectively

$$\check{\mathbf{r}} = \begin{bmatrix} \mathbf{r}(\check{\mathbf{z}}) \\ \mathbf{r}^*(\check{\mathbf{z}}) \end{bmatrix}, \quad \check{\mathbf{d}} = \begin{bmatrix} \mathbf{d}(\check{\mathbf{z}}) \\ \mathbf{d}^*(\check{\mathbf{z}}) \end{bmatrix}, \quad \check{\mathbf{v}} = \begin{bmatrix} \mathbf{v}(\check{\mathbf{z}}) \\ \mathbf{v}^*(\check{\mathbf{z}}) \end{bmatrix}. \quad (3.4)$$

Based on these definitions, the full Jacobian matrix, \mathbf{J} , for the problem is defined as

$$\mathbf{J} = \frac{\partial \check{\mathbf{v}}}{\partial \check{\mathbf{z}}} = \begin{bmatrix} \frac{\partial \mathbf{v}}{\partial z} & \frac{\partial \mathbf{v}}{\partial z^*} \\ \frac{\partial \mathbf{v}^*}{\partial z} & \frac{\partial \mathbf{v}^*}{\partial z^*} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{vz} & \mathbf{J}_{vz^*} \\ \mathbf{J}_{v^*z} & \mathbf{J}_{v^*z^*} \end{bmatrix}. \quad (3.5)$$

The terms \mathbf{J}_{vz} , \mathbf{J}_{vz^*} , \mathbf{J}_{v^*z} and $\mathbf{J}_{v^*z^*}$ are called partial and partial conjugate Jacobians. A deeper look shows that the diagonally adjacent terms are element-by-element conjugates of each other. From these definitions, the update steps for the parameters are defined as follows for the GN and LM algorithms

$$GN; \quad \delta\check{\mathbf{z}} = (\mathbf{J}^H \mathbf{J})^{-1} \mathbf{J}^H \check{\boldsymbol{\gamma}}, \quad (3.6)$$

$$LM; \quad \delta\check{\mathbf{z}} = (\mathbf{J}^H \mathbf{J} + \lambda \mathbf{D})^{-1} \mathbf{J}^H \check{\boldsymbol{\gamma}}, \quad (3.7)$$

where λ is the LM damping factor and \mathbf{D} is the diagonalised Hessian matrix, $\mathbf{J}^H \mathbf{J}$.

For the full polarised case all that is required is to vectorise the 2×2 complex matrices and derive the update steps given above. In Appendix B of their paper, [Smirnov & Tasse \(2015\)](#) define an operator calculus which makes the manipulation of 2×2 complex variables more convenient. For 2×2 complex variables, \mathbf{Z} , we define our augmented variables analogously to the scalar case i.e.

$$\check{\mathbf{Z}} = \begin{bmatrix} \vec{\mathbf{Z}} \\ \vec{\mathbf{Z}}^* \end{bmatrix}, \quad \check{\mathbf{R}} = \begin{bmatrix} \vec{\mathbf{R}}(\check{\mathbf{Z}}) \\ \vec{\mathbf{R}}^*(\check{\mathbf{Z}}) \end{bmatrix}, \quad \check{\mathbf{V}} = \begin{bmatrix} \vec{\mathbf{V}}(\check{\mathbf{Z}}) \\ \vec{\mathbf{V}}^*(\check{\mathbf{Z}}) \end{bmatrix}, \quad (3.8)$$

where $\vec{\mathbf{Z}}$ denotes the vector of matrices formed from all the parameters \mathbf{Z} . The quantities $\check{\mathbf{R}}$ and $\check{\mathbf{V}}$ are the augmented residuals and modelled visibilities respectively, expressed as functions of 2×2 complex matrices. The superscript $*$ denotes element-wise complex conjugation. The full Jacobian matrix naturally follows as

$$\mathbf{J} = \frac{\partial \check{\mathbf{V}}}{\partial \check{\mathbf{Z}}} = \begin{bmatrix} \frac{\partial \vec{\mathbf{V}}}{\partial \vec{\mathbf{Z}}} & \frac{\partial \vec{\mathbf{V}}}{\partial \vec{\mathbf{Z}}^*} \\ \frac{\partial \vec{\mathbf{V}}^*}{\partial \vec{\mathbf{Z}}} & \frac{\partial \vec{\mathbf{V}}^*}{\partial \vec{\mathbf{Z}}^*} \end{bmatrix}. \quad (3.9)$$

The derivatives that appear in Equation (3.9) are matrix by matrix derivatives. These can be conveniently dealt with by using the operator calculus introduced by [Smirnov & Tasse \(2015\)](#) which can be consulted for further details. The crucial result is that for any 2×2 matrices \mathbf{A} , \mathbf{B} and \mathbf{C} , we have

$$\frac{\partial(\mathbf{ABC})}{\partial \mathbf{A}} = \mathcal{R}_{\mathbf{C}} \mathcal{R}_{\mathbf{B}}, \quad \frac{\partial(\mathbf{ABC})}{\partial \mathbf{B}} = \mathcal{L}_{\mathbf{A}} \mathcal{R}_{\mathbf{C}}, \quad \frac{\partial(\mathbf{ABC})}{\partial \mathbf{C}} = \mathcal{L}_{\mathbf{A}} \mathcal{L}_{\mathbf{B}}, \quad (3.10)$$

where \mathcal{L}_A and \mathcal{R}_A are matrix operators which act on 2×2 matrices called the left and right multipliers. They are defined such that for any 2×2 matrices \mathbf{A} and \mathbf{B} :

$$\begin{aligned}\mathcal{L}_A \mathbf{B} &= \mathbf{A} \mathbf{B}, \\ \mathcal{R}_A \mathbf{B} &= \mathbf{B} \mathbf{A}.\end{aligned}\tag{3.11}$$

The key point here is that, even for 2×2 complex variables, by carefully vectorising and using Eq. (3.10) and Eq. (3.11), we end up with the following GN and LM update steps

$$GN; \quad \delta \check{\mathbf{Z}} = (\mathbf{J}^H \mathbf{J})^{-1} \mathbf{J}^H \check{\mathbf{R}},\tag{3.12}$$

$$LM; \quad \delta \check{\mathbf{Z}} = (\mathbf{J}^H \mathbf{J} + \lambda \mathbf{D})^{-1} \mathbf{J}^H \check{\mathbf{R}}.\tag{3.13}$$

These are similar to those for the complex scalar case and can be implemented in an analogous way.

3.1.2 CubiCal overview

CubiCal (Kenyon et al., 2018) is a recently developed software package which exploits complex optimisation. We provide a brief discussion of the software package, as our algorithm has been implemented as one of its subroutines.

A common bottleneck when implementing any NLLS algorithm is inverting the linearised approximation of the Hessian matrix, $\mathbf{J}^H \mathbf{J}$, appearing in (3.12) and (3.13). Smirnov & Tasse (2015) showed that, given a particular ordering of the solvable parameters (viz. antennas, directions, and correlations), this matrix is sparse in nature provided the problem is approached using Wirtinger calculus. Consequently, it can be approximated by a diagonal matrix. CubiCal utilises this diagonal approximation to significantly reduce the computational cost of implementing the GN or LM update rules, albeit with slightly less accuracy. The algorithmic trade-off is that we usually require more of these significantly cheaper iterations to reach convergence. Kenyon (2019) shows that this results in significant performance benefits in real-life cases.

CubiCal's modular structure makes implementing additional solvers, such as those presented here, relatively easy. In fact, all currently implemented CubiCal solvers (i.e. phase-only solvers, amplitude and phase solvers, parametrised slope solvers, and so on) could easily be augmented

with Complex Student's t implementations. CubiCal reads visibilities in the conventional Measurement Set data format. Model visibilities can be read from a Measurement Set or computed on-the-fly from a component sky model using the Montblanc package (Perkins et al., 2015). This flexibility allows CubiCal to be incorporated into various 2GC and 3GC schemes with ease.

3.2 Proper Complex Student's t Calibration

This section details the implementation of the iteratively re-weighted complex NLLS solver (henceforth the *robust solver*). In particular, we give the form of the proper complex Student's t-distribution (CST) as well as the update rules used for calibration. A full derivation is provided in Appendix B.

3.2.1 Proper CST

The Student's t-distribution (ST) is well known in the field of optimisation for its robustness in the presence of data containing outliers (see Lange et al. (1989) for example), when compared with a Gaussian distribution. One way of constructing the ST is to visualise it as a mixture of random variables drawn from several Gaussian distributions with different standard deviations. We construct a CST by integrating a proper complex normal distribution over an unknown scale parameter τ for which we prescribe a Gamma prior, i.e.

$$\text{CST}(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Lambda}, v) = \int_0^\infty \text{CN}(\mathbf{y}|\boldsymbol{\mu}, (\tau\boldsymbol{\Lambda})^{-1}) \text{Gam}(\tau|v, v) d\tau, \quad (3.14)$$

$$\begin{aligned} &= \int_0^\infty \frac{\tau^D |\boldsymbol{\Lambda}|}{\pi^D} \exp(-(\mathbf{y} - \boldsymbol{\mu})^H (\tau\boldsymbol{\Lambda})(\mathbf{y} - \boldsymbol{\mu})) \\ &\quad \times \frac{v^v \tau^{v-1} \exp(-v\tau)}{\Gamma(v)} d\tau \end{aligned} \quad (3.15)$$

where $\text{CN}(\mathbf{y}|\boldsymbol{\mu}, (\tau\boldsymbol{\Lambda})^{-1})$ is a proper complex normal distribution with mean $\boldsymbol{\mu} \in \mathbb{C}^D$ and Hermitian precision matrix $(\tau\boldsymbol{\Lambda}) \in \mathbb{C}^{D \times D}$. $\text{Gam}(\tau|v, v)$ is a Gamma distribution in $\tau \in \mathbb{R}^+$ and acts as a prior on the unknown scale parameter that we want to marginalise over. The resulting

distribution takes the form

$$\text{CST}(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Lambda}, v) = \frac{\Gamma(v+D)|\boldsymbol{\Lambda}|}{\Gamma(v)(v\pi)^D} \left(1 + \frac{\Delta^2}{v}\right)^{-v-D}. \quad (3.16)$$

Since this distribution is not a member of the exponential family, working with it directly is usually difficult. The standard way to overcome this difficulty is to utilise the Expectation Maximisation (EM) algorithm (see [Bishop \(2006\)](#) for example). The EM algorithm iterates between estimating the expected value of the latent variables (missing or unavailable data) from their posterior distributions (E-step) and maximising the complete data likelihood function (M-step), which is generally easier to work with compared to the marginalised likelihood. For the full maximum likelihood solution, we need to solve for all the parameters of the CST distribution (viz. the means $\boldsymbol{\mu}$, the precision matrix $\boldsymbol{\Lambda}$ and the number of degrees of freedom) during this step.

For the specific problem of robust regression with a CST, the latent variables correspond to the scale parameter τ in Equation (3.14) whose posterior distribution is a Gamma distribution. As we show in Appendix B, the solution can be obtained using an iteratively re-weighted complex NLLS algorithm in which the weights are computed as the expectation of τ under the Gamma posterior. An important aspect of the algorithm is that it adapts the likelihood used for calibration to the problem at hand by inferring the optimal number of degrees of freedom v . This parameter dictates the shape of the distribution. If the residuals are Gaussian, the inferred v parameter will be large and we essentially recover a Gaussian likelihood. If the residuals contain outliers, the inferred v parameter will be small and data points which could otherwise bias the optimisation procedure get down-weighted and therefore do not significantly affect the calibration solutions. Some further insight into this behaviour is provided below.

3.2.2 Robust Calibration

Consider the following form of the RIME for a field with N_d sources

$$\mathbf{V}_{pqs} = \sum_{d=1}^{N_d} \mathbf{G}_{ps}(d) \mathcal{X}_{pqs}(d) \mathbf{G}_{qs}^H(d) + \epsilon_{pqs}, \quad (3.17)$$

where \mathbf{G}_{ps} is the gain for antenna p , \mathcal{X}_{pqs} is the sky coherency in direction d , s is the corresponding time and frequency index and ϵ_{pqs} is the noise which is assumed to be CST distributed. Thus calibration can be performed as described in Appendix B. Our goal is to find both the model parameters and the expectation values of the latent variables τ_i that minimise the joint log-likelihood function Q given by Eq. (B.19). For the model parameters the solution as described by Eq. (B.20) and Eq. (B.21) is a weighted NLLS where the weights are the expectation values $\mathbb{E}[\tau_i]$ of the latent variables τ_i . Hence given initial values for the weights, \mathbf{W} , the Jones matrices, or gains, can be computed by minimising the following objective function

$$\min_{\mathbf{G}} \|\mathbf{W}(\check{\mathbf{R}}(\mathbf{G}, \mathbf{G}^H))\|_F^2 = \min_{\mathbf{G}} \|\mathbf{W}(\check{\mathbf{D}} - \check{\mathbf{V}}(\mathbf{G}, \mathbf{G}^H))\|_F^2, \quad (3.18)$$

where \mathbf{G} is the gain matrix and $\check{\mathbf{R}}$, $\check{\mathbf{D}}$ and $\check{\mathbf{V}}$ are the augmented residual, data and model vectors respectively. The elements of the \mathbf{W} matrix are updated at each iteration using the expectation values of the latent variables τ_i of the CST. Explicitly following Eq. (B.25), they can be written as

$$\mathbf{w}_{pqs} = \frac{v + n_c}{v + \mathbf{R}_{pqs}^H \boldsymbol{\Sigma}^{-1} \mathbf{R}_{pqs}}, \quad (3.19)$$

where \mathbf{w}_{pqs} represents the weight of the 2×2 visibility matrix between antenna p and q at time and frequency index s , n_c is the number of correlations in our data and $\mathbf{R}_{pqs} = \text{vec}(\check{\mathbf{R}}_{pqs})$ is the residual of the corresponding visibility. Note that \mathbf{R}_{pqs} here is a 4×1 vector and not a 2×2 matrix, as expected from the vectorisation. $\boldsymbol{\Sigma}$ is the covariance matrix of the residual visibilities and it is a 4×4 matrix we generally assume to be diagonal. The number of correlations, n_c , is important because, even though CubiCal assumes a data structure where each visibility is a 2×2 matrix, for scalar calibration or data with single correlations, the cross correlation terms are set to zero. Hence, n_c , which represents the dimension of a single vectorised visibility, is effectively 4 only when all the correlations are present. Note that the $\mathbf{R}_{pqs}^H \boldsymbol{\Sigma}^{-1} \mathbf{R}_{pqs}$ term in the denominator will have an expectation value of n_c if the data are Gaussian distributed with covariance matrix $\boldsymbol{\Sigma}$.

The v -term is computed by solving the following equation (see the derivation of Eq. (B.24))

for more details)

$$-\psi(v) + \log(v) + 1 + \psi(v + n_c) - \log(v + n_c) + \frac{1}{N} \sum_{pq} (\log(\mathbf{w}_{pqs}) - \mathbf{w}_{pqs}) = 0, \quad (3.20)$$

where ψ is called the digamma function (logarithmic derivative of the gamma function) and N is the total number of visibilities. Eq. (3.20) has no closed form solution and has to be solved numerically. We find that, in practice, it is sufficient to restrict v to be an integer and to simply do a grid search between $2 \leq v \leq 50$ since, as already mentioned, the ST is almost indistinguishable from a Gaussian when $v > 30$ or so. Finally, at each iteration, the covariance matrix Σ is computed as follows

$$\Sigma = \frac{1}{N} \sum_{pqs} (\mathbf{R}_{pqs} \mathbf{R}_{pqs}^H \mathbf{w}_{pqs}), \quad (3.21)$$

where N is again the total number of visibilities.

A closer look at Eq. (3.19) can provide some insight into the workings of the robust solver. Clearly, the solver assigns small weights to visibilities with large residuals and large weights to visibilities with small residuals². When the residuals follow a Gaussian distribution with covariance Σ , the v -term is large and all the visibilities end up having approximately equal weights. On the other hand, for visibilities containing outliers, the v -term is small and the outliers can be effectively down-weighted. Finally, suppose that the covariance has been under-estimated (as will be the case if the residuals also contain a realistic unmodelled point source distribution). In this case, the $\mathbf{R}_{pqs}^H \Sigma^{-1} \mathbf{R}_{pqs}$ term in the denominator will be much larger than n_c and these points will be down-weighted, thus discouraging over-fitting.

3.2.3 Implementation details

Algorithm 1 shows the details of the new algorithm which we have dubbed the robust solver. The robust solver implementation was greatly simplified thanks to CubiCal's object-oriented programming approach. CubiCal provides an abstract class interface with preset attributes and

²Note the upper bound on the weights is finite and equal to $\frac{v + n_c}{v}$.

functions which need to be inherited and defined to develop any new solver. In CubiCal terminology, we refer to this as a Gain Machine. The new solver is invoked in CubiCal by setting the solver's option *gain-type* to *robust-2x2*. The expected thermal noise level for the observation is

Algorithm 1 : Robust Solver Algorithm

Require: Data \check{D} , Model \check{V} , Jacobian'func, i_{\max}

Initialisation: $\check{G}_0 \leftarrow \mathbf{1}$, $\mathbf{w}_{pqs} \leftarrow 1$, $v \leftarrow 2$, $i \leftarrow 0$

while (not **converged** or not **stalled** or $i \leq i_{\max}$) **do**

$\mathbf{W} \leftarrow \text{Diag}(\mathbf{w}_{pqs})$ {# Diagonal matrix with weights}

$\mathbf{J} \leftarrow \text{Jacobian'func}(\check{D}, \check{V}, \check{G}_{i-1})$

$\check{R} \leftarrow \check{D} - \check{V}$

$\Sigma \leftarrow \text{Update } \Sigma \text{ using } \check{R}, \mathbf{W} \text{ and Eq. (3.21)}$

$\delta\check{G} \leftarrow (\mathbf{J}^H \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^H \mathbf{W} \check{R}$

$\check{G}_{\text{temp}} \leftarrow \check{G}_{i-1} + \delta\check{G}$

if $i \bmod 2 = 0$ or DD calibration

then

$\check{G}_i \leftarrow \frac{1}{2} (\check{G}_{\text{temp}} + \check{G}_{i-1})$

else

$\check{G}_i \leftarrow \check{G}_{\text{temp}}$

end if

for all baselines **do**

$\mathbf{w}_{pqs} \leftarrow \frac{v + n_c}{v + \mathbf{R}_{pqs}^H \Sigma^{-1} \mathbf{R}_{pqs}}$

end for

$v \leftarrow \text{Compute } v \text{ using Eq. (3.20)}$

$i \leftarrow i + 1$

end while

used to pre-whiten the data. This means that the weights can be initialised to 1 during the first iteration. We treat them as scalar real variables meaning all correlations have the same weight. Furthermore, the weights are assigned per visibility, independently of time, frequency or base-

line. As is customary in radio interferometry, the weights of flagged data are set to zero from the start. The computation of the weights involves the residual covariance matrix, Σ , which is not included in [Kazemi & Yatawatta \(2013\)](#) but just assumed to be \mathbf{I} . We do not make this assumption. Instead, we implement two variants of the algorithm, one with Σ computed using Eq. (3.21), and another where we set Σ to \mathbf{I} . A setting is made available to the user to decide whether or not they want Σ to be computed during every iteration or simply set it to \mathbf{I} . The default behaviour of the solver is to compute Σ as it is more consistent with our derivation of the algorithm (see Appendix B). Furthermore, we also provide an option to fix the number of degrees of freedom at the outset without inferring it using Eq. (3.20).

It has been observed that averaging the gain solutions every second iteration improves the convergence speed of the algorithm (see [Salvini & Wijnholds \(2014\)](#)). [Smirnov & Tasse \(2015\)](#) explain that this averaging corresponds to alternating between the GN and LM algorithms. This is very helpful when calibrating for DD effects as these generally converge slowly. For the CubiCal solver, we average solutions at every iteration for DD calibration, and at even iterations for DI calibration.

3.2.4 Computational cost

The main additional operations performed by the robust solver are the computations of the weights and the numerical solution for the degrees of freedom, v . Assigning the weights relies on computing the residual visibilities and the covariance matrix Σ . The algorithm is implemented such that the residuals are computed only once during every iteration. The residuals computed for the weight updates are stored in memory and reused during the gain updates. For DI calibration, the default solver does not compute residual visibilities at every iteration. This is made possible thanks to an observation from [Tasse \(2014a\)](#). For DI calibration, we have the following RIME form

$$\check{\mathbf{V}} = \check{\mathbf{G}}\check{\mathbf{M}}\check{\mathbf{G}}^H, \quad (3.22)$$

where $\check{\mathbf{M}}$ corresponds to the true or modelled visibilities. [Tasse \(2014a\)](#) states,

$$\check{\mathbf{V}} = \mathbf{J}_L \mathbf{G} = \frac{1}{2} \mathbf{J} \check{\mathbf{G}}, \quad (3.23)$$

where $(\cdot)_L$ denotes the left half of a matrix and $\check{\mathbf{G}} = \begin{bmatrix} \mathbf{G} \\ \mathbf{G}^H \end{bmatrix}$. Substituting Equation (3.23) in Equation (3.12), we have the update rule for DI calibration below

$$\delta \mathbf{G} = (\mathbf{J}^H \mathbf{J})_U^{-1} \mathbf{J}^H (\check{\mathbf{D}} - \mathbf{J}_L \mathbf{G}) \quad (3.24)$$

$$= (\mathbf{J}^H \mathbf{J})_U^{-1} \mathbf{J}^H \check{\mathbf{D}} - \mathbf{G}, \quad (3.25)$$

where $(\cdot)_U$ stands for the upper half of a matrix. This implies that

$$\mathbf{G}_i = \mathbf{G}_{i-1} + \delta \mathbf{G} \quad (3.26)$$

$$= (\mathbf{J}^H \mathbf{J})_U^{-1} \mathbf{J}^H \check{\mathbf{D}}. \quad (3.27)$$

Hence, residuals are not required for updating the gains. In the case of DD calibration Equation (3.23) does not hold, and both solvers have to compute residuals at each iteration. Fortunately, CubiCal employs various levels of parallelism, and we script the most expensive tasks in the Cython programming language (note that, in the latest version, Cython has been replaced with Numba). These dramatically improve the speed for generating the necessary residual visibilities. Additionally, in CubiCal, only the diagonal of the Hessian is computed and the full Jacobian matrix is never loaded into memory but is instead implemented as an operator.

CubiCal uses the below data structure

$$\check{\mathbf{D}} = \begin{bmatrix} 0 & \mathbf{D}_{12} & \mathbf{D}_{13} & \dots & \mathbf{D}_{1N_a} \\ \mathbf{D}_{12}^H & 0 & \mathbf{D}_{23} & \dots & \mathbf{D}_{2N_a} \\ \mathbf{D}_{13}^H & \mathbf{D}_{23}^H & 0 & \dots & \mathbf{D}_{3N_a} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{D}_{1N_a}^H & \mathbf{D}_{2N_a}^H & \mathbf{D}_{3N_a}^H & \dots & 0 \end{bmatrix}, \quad (3.28)$$

where N_a is the number of antennas and each element is a 2×2 complex matrix. Half of the data is just the conjugate transpose of the other half. This implies only half of the data is required to compute the covariance matrix Σ . Similarly, half of the weights are sufficient to solve for v . Another optimisation strategy is to update v only after a specific number of iterations. The number of iterations after which to recompute v is a setting which can be modified by the user.

Moreover, we restrict the search space for v by assuming it is an integer and performing a grid search between 2 and 50. We do this by computing the function at different v integer positions and take the v position with the minimum value as the solution. We avoid using numerical solvers as they may introduce convergence issues or slow the solver since we only need an estimate of this value.

Applications of the Robust Solver¹

Chapter 3 described the implementation of a robust algorithm for radio interferometric calibration. In this chapter, we follow up by demonstrating that the implemented algorithm can mitigate some of the biases introduced by incomplete sky models and radio frequency interference by applying it to both simulated and real data. Our results show significant improvements compared to a conventional least-squares solver which assumes a Gaussian likelihood function. Furthermore, we provide some insight into why the algorithm outperforms the conventional solver and discuss specific scenarios for both DI and DD self-calibration where this is expected to be the case. §4.1 describes different simulations demonstrating how the implemented solver outperforms traditional solvers based on the amount of flux suppressed from unmodelled sources for both DD and DI calibration. We show that the implemented solver improves on the results from [Kazemi & Yatawatta \(2013\)](#) because the residual covariance matrix is estimated from the data and not assumed to be equal to the identity matrix. In §4.2, the algorithm is applied to synthetic and real data from the Karl G. Jansky Very Large Array (VLA) for low-level RFI mitigation.

¹The chapter is based on [Sob et al. \(2019\)](#), hence most of the tests, figures and formulations are drawn from On therein.

4.1 Robust solvers and flux suppression

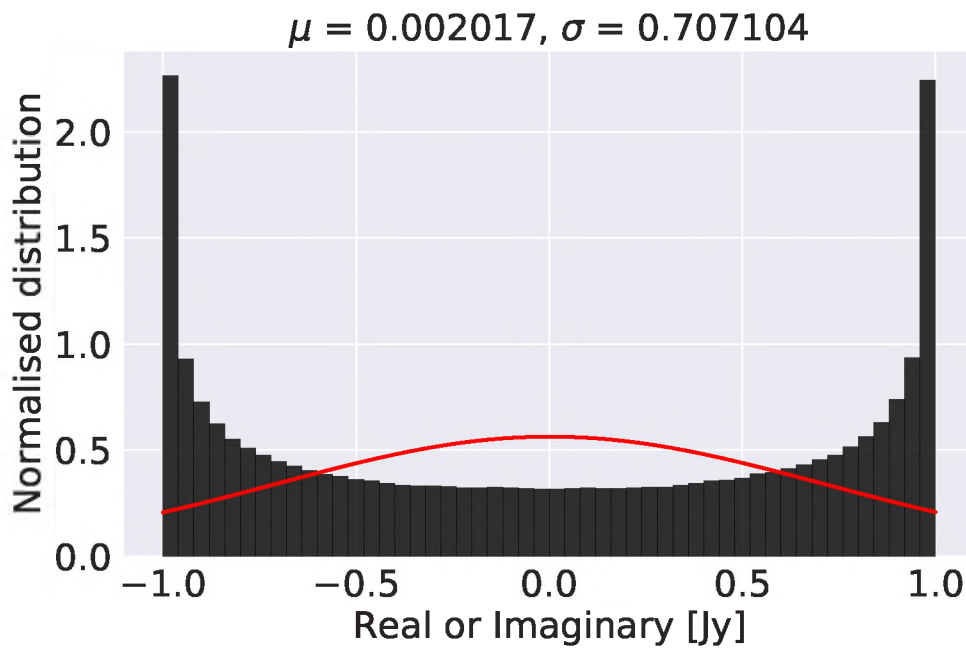
This section uses simulated data to identify some regimes in which the robust solver can be expected to improve the results of calibration. Our main aim is to compare how much of the unmodelled flux gets suppressed during calibration with the different solvers. For brevity, we refer to them as follows:

- “complex solver”: a conventional least-squares solver employing the Wirtinger formulation (identified as “cp” in figure legends).
- “robust solver” with covariance iteratively recomputed (identified as “rb” in figure legends).
- “robust-I solver” for the robust solver with covariance set to \mathbf{I} (identified as “rb-I” in figure legends).

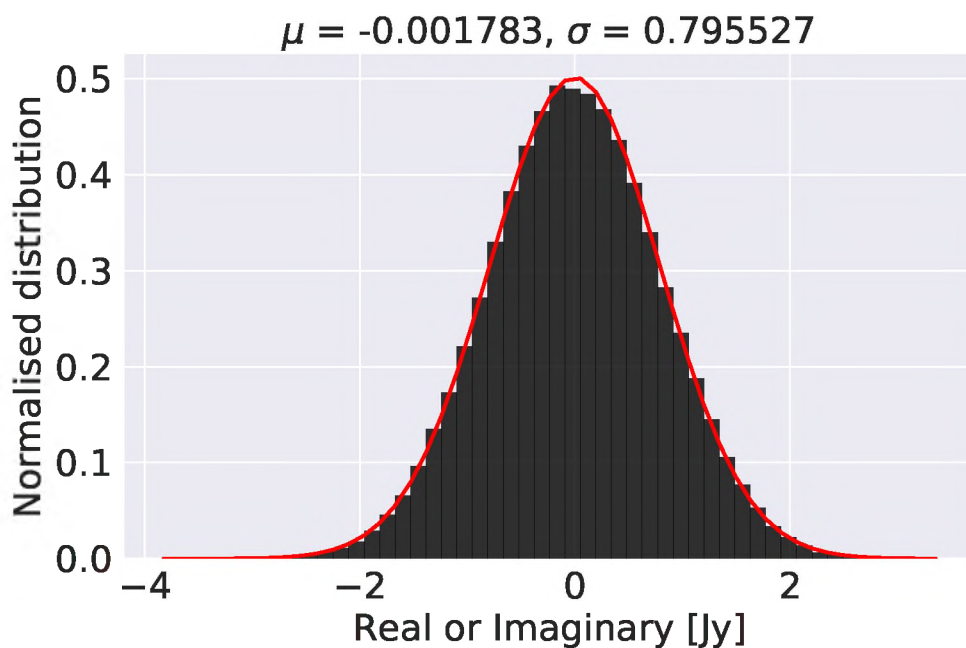
To aid our understanding of when the robust solver can be expected to out-perform the traditional solver, we start with a simple illustration of how unmodelled sources affect the statistics of the residual visibilities. The tools used to perform all the simulations in this chapter and the next are fully described in Appendix D.

4.1.1 Statistical properties of visibilities

Calibration with incomplete sky models implies that the residuals which we attempt to minimise during the optimisation process (calibration) still contain the contribution of numerous unmodelled sources. To understand how this affects the solver (which assumes that the residuals consist of pure noise) we simulate some data and plot a histogram of the real and imaginary parts in Fig. 4.1. If we consider a field consisting of a single 1 Jy source at its centre, the visibilities for this sky have no phase component, and all the visibilities are equal to 1 in this case. The histogram for these visibilities will have two peaks, one at 1 Jy for the real part of the visibilities and the other at 0 Jy for the imaginary component of the visibilities. If we move the source to an offset position from the field centre, the visibilities now have a phase which depends on the offset. Fig.



(a) 1 Jy source at an offset position



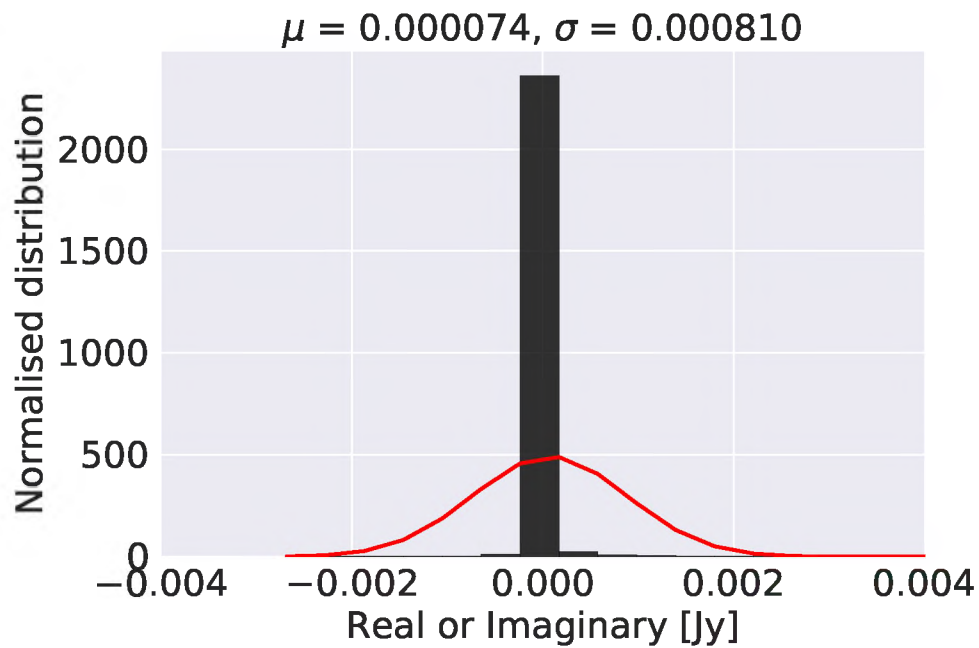
(b) 100 sources

Figure 4.1: (a): Histogram of simulated visibilities of a 1 Jy source at an offset position from the phase centre of the field. (b): Histogram of simulated visibilities for 100 sources drawn from a realistic sky model. The red curve in each plot is the corresponding Gaussian probability function computed using the mean and standard deviation of the visibilities. The mean, μ , and standard deviation, σ , are shown respectively on the figure titles.

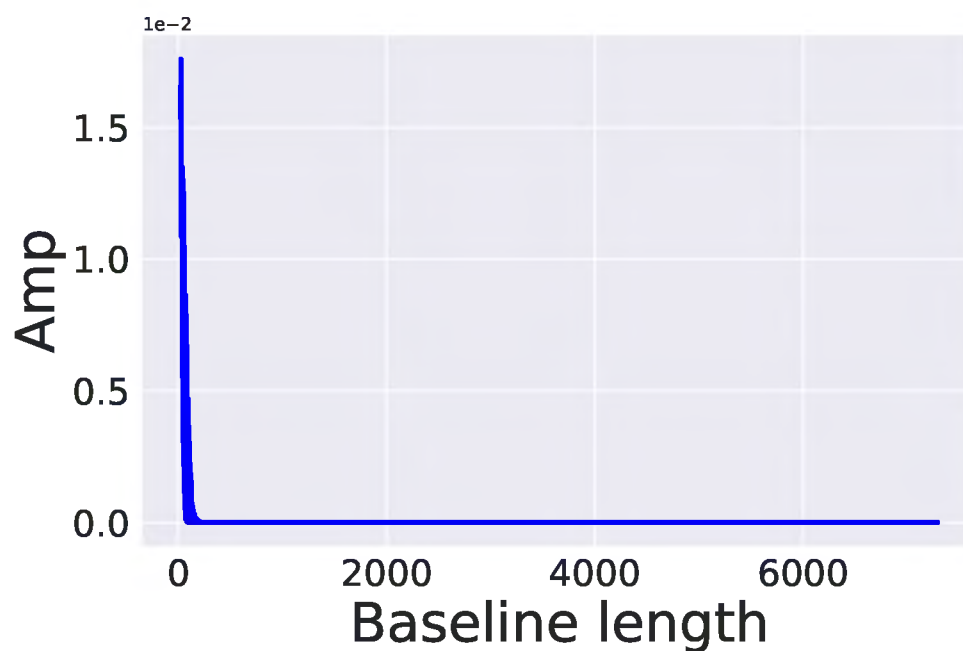
[4.1a](#) is the histogram of simulated visibilities for a 1 Jy source at an offset position from the centre. The distribution in this case also has two peaks, one at 1 Jy and one at -1 Jy. In between these peaks, the distribution is almost uniform.

[Fig. 4.1b](#) shows the histogram of the visibilities for a field consisting of 100 sources having uniform positions and fluxes drawn from a power law distribution (Pareto distribution with $\alpha = 1$) with peak flux set to 1 Jy. Clearly, the distribution of the visibilities approaches a Gaussian. This implies that, given a sufficiently large number of unmodelled point sources with random positions and fluxes (as is usually the case for the fainter sources which do not end up in the model), the distribution of residuals remains Gaussian. Unmodelled point sources therefore tend to simply increase the variance of the residuals above that of the thermal noise contribution. This suggests that unmodelled sources with a flux level below a certain noise-dependent threshold will have almost no effect on the gain solutions.

In contrast to point sources, extended sources are usually challenging to model, both during calibration and imaging. Conventional imaging algorithms such as CLEAN (see [Schwab \(1984\)](#) for example) models the sky as a combination of point sources. The latter makes it very difficult to construct a model for an extended source during calibration. [Fig. 4.2a](#) is a histogram of the visibilities of an extended source. The distribution is centred around a few values. We can model an extended source as a Gaussian with a large radius in the image domain, hence in the uv domain, the visibilities are Gaussian with a tiny radius. In other words, the visibilities of extended sources are concentrated around the short baselines. We illustrate this with a plot of amplitude against the baseline length for the visibilities of an extended source (see [Fig. 4.2b](#)). The extended source here is simulated to be elliptical with minor and major axis 200 and 300 arcseconds respectively. [Fig. 4.2b](#) confirms that the visibilities from extended structures only contribute to the short baselines.



(a)



(b)

Figure 4.2: (a): Histogram of the visibilities of an extended source with size $200'' \times 300''$. The red curve in each plot is the corresponding Gaussian probability function computed using the mean and standard deviation of the visibilities. (b): Plot of the amplitude against baseline length for an extended source.

Since a typical selfcal procedure begins by constructing a sky model from a 1GC-calibrated image, down to a certain flux threshold, the unmodelled source fraction will tend to consist of multiple faint sources, and therefore will follow Fig. 4.1b. The properties of the modelled source fraction will, on the other hand, strongly depend on the spatial distribution of flux across the field. Here, we can identify two contrasting regimes. In a field dominated by a bright source, most of the modelled flux will be concentrated in that source, and the distribution of model visibilities will look like Fig. 4.1a. We'll call this the *concentrated model* regime. In a field with no bright sources, model flux will be spread between multiple fainter sources (we'll call this a *dispersed model* regime), and the visibility distribution will resemble that of Fig. 4.1b.

We can define the effective SNR of the sky model in terms of the visibilities, as

$$\text{SNR} = 10 \log \left(\frac{\langle \mathbf{V} \cdot \mathbf{V}^* \rangle_{\nu,t,pq}}{\langle \mathbf{N}' \cdot \mathbf{N}'^* \rangle_{\nu,t,pq}} \right), \quad (4.1)$$

where $\langle \rangle_{\nu,t,pq}$ denotes averaging over frequency, time and baseline, \mathbf{V} represents the modelled visibilities, and \mathbf{N}' is the effective noise, i.e. the sum of the unmodelled visibilities and the noise. Clearly, SNR is a function of both the total model flux, and the model concentration. For a maximally concentrated model consisting of a single source, $\mathbf{V} \cdot \mathbf{V}^*$ will be equal to the source flux squared. For a disperse model, $\langle \mathbf{V} \cdot \mathbf{V}^* \rangle$ will contain contributions from many interfering fringes. A dispersed model with the same total flux will therefore have much lower SNR. We can then ask whether model concentration, as well as SNR, affects the degree of source suppression.

Conventional intuition for the workings of selfcal is honed in the “classic regime” of high-SNR, concentrated models, typically associated with targeted observations of individual sources. With the advent of blind large-area surveys, we are seeing more and more fields lacking a dominant source: these need to be calibrated in a low-SNR, dispersed model regime. Finally, direction-dependent calibration deals with concentrated models almost by definition, but these can be quite low SNR. The next section shows marked differences in flux suppression across these regimes.

4.1.2 SNR, model concentration, and flux suppression

In this section, we investigate how flux suppression of direction-independent calibration behaves with varying effective SNR and model concentration. We determine under which circumstances we can expect the robust solver to deliver an improvement over the traditional solver. To do this, we simulate a series of observations containing point sources and thermal noise. The point sources are split between a fainter “unmodelled fraction” (i.e. assumed unknown for the purposes of calibration), and a brighter known fraction (i.e. included in the calibration sky model). We calibrate the mock observations using the calibration sky model, and then measure flux suppression at the position of the unmodelled sources. More specifically:

1. For the unmodelled source fraction, we generate a sky model containing 100 random point sources as before, and then rescale the fluxes so that their total flux comes to 1 Jy. We call this the faint sky.
2. For the modelled fraction, we generate a variety of calibration sky models corresponding to different model concentrations and effective SNR levels:
 - (ii-a) We fix the total flux in the modelled fraction at 1 Jy, and vary the number of sources from 1 to 50. This corresponds to diluting the model and decreasing SNR simultaneously.
 - (ii-b) **Concentrated model, varying SNR:** we use a modelled fraction of one source, and scale its flux to achieve different SNR levels.
 - (ii-c) **Dispersed model, varying SNR:** we use a modelled fraction of 50 sources, and scale their fluxes to achieve different SNR levels.
 - (ii-d) **Fixed SNR, varying concentration:** we generate models of 1, 10, 20, 30, 40 and 50 sources. We scale the fluxes of each model to achieve an effective SNR of 10 dB in each case.
3. We combine the faint sky and the calibration sky model for each experiment, and simulate visibilities corresponding to the combined sky model using the MeerKAT ([Jonas & Team](#),

2018) array layout. We simulate a single-channel observation at 1 GHz, with a bandwidth of 1 MHz, a total synthesis time of 2 hours, and an integration time of 10 seconds.

4. We add Gaussian noise with an rms of 10 mJy to the simulated visibilities. This value is approximately 3 times the expected rms using MeerKAT system equivalent flux density (SEFD) at frequencies around 1 GHz. This rms corresponds to an image noise rms of 6 μ Jy/beam using natural weighting. This value will be used in all simulations unless stated otherwise.
5. We perform DI calibration on the data with all three solvers using only the calibration sky model to compute the model visibilities. Since no gains are applied to the visibilities during the simulation, we expect a perfect calibration to return unity gain solutions.
6. We compute the residuals (by applying the gain solutions to the model visibilities and subtracting them from the data) and image these to get a residual image.
7. We deconvolve the resulting images using WSCLEAN (Offringa et al., 2014) in single scale mode with natural weighting to try and recover the faint source distribution.

We are now in a position to study the degree of flux suppression of the unmodelled sources. Since our simulations consist of point sources only, the recovered fluxes are estimated by simply measuring the pixel values at the position of the sources in the respective restored deconvolved images.

To quantify how a reduction in SNR affects source suppression, we have to create a statistic to measure it with. For this purpose we use the average suppression (AS), which is defined as

$$\text{AS} = \frac{1}{N_s} \sum_i^{N_s} \frac{|I_i - \hat{I}_i|}{I_i}, \quad (4.2)$$

where N_s is the number of sources and I_i and \hat{I}_i are the true and recovered flux of the i^{th} source respectively.

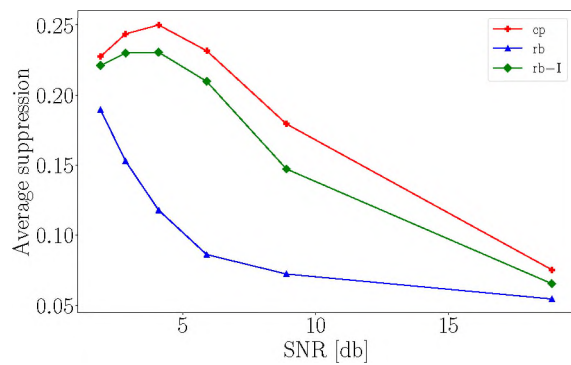
Fig. 4.3 shows the AS as a function of SNR and model diluteness, following scenarios (a) – (d) outlined above. These plots reveal a number of very interesting trends:

- The robust solver (blue) curve always outperforms (in the sense of reducing flux suppression) both the robust-I (green curve) and standard solvers (red curve), in some regimes by a very significant margin. The robust-I solver outperforms the standard solver in almost all cases, but this improvement is not always significant.
- For a concentrated model (Fig. 4.3b), flux suppression increases with decreasing SNR. At high SNR (the “classical regime” of selfcal), the performance of all solvers tends to converge (Figs. 4.3a and 4.3b, right end of the plot), to a value of slightly below 7%.²
- Flux suppression increases significantly (to over 25%!) with model dispersion (Fig. 4.3d), at least with the standard and robust-I solvers.
- For a highly dispersed model (Fig. 4.3c), flux suppression with the standard and robust-I solvers is quite high, and almost independent of SNR. The robust solver offers much better performance in all but the lowest SNR regimes.
- There is an interesting downturn in flux suppression at low SNR in Figs. 4.3a, 4.3c (left end of the plots). We can only speculate as to its ultimate cause. Grobler et al. (2014) showed that flux suppression comes about through a combination of ghost sources (see e.g. Eq. 35 therein), and that the intensity of the ghost response has a complex relationship to modelled/unmodelled flux ratios, even in the simplest, two-source case studied in that work. Perhaps pertinently, Fig. 15 *ibid.* shows a distinct downturn in the ghost response towards low SNR (i.e. higher flux ratios in the figure). We speculate that we are seeing the same mechanism at work here. Furthermore, from continuity considerations, it is obvious that there *must* be a downturn in flux suppression at very low SNR – after all, an empty calibration model cannot suppress flux at all. Since calibration in such a low SNR scenario is pointless, we won’t pursue this puzzle further here.

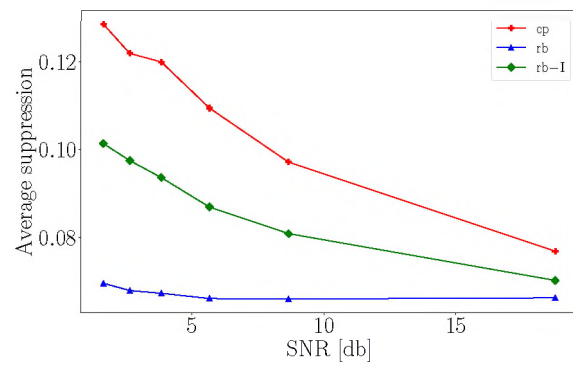
The crucial conclusion of this section is that, in principle, the robust solvers outperform the traditional complex solver in all the cases we have considered (at least as far as flux suppression

²Previous studies (Grobler et al., 2014; Nunhokee, 2015) have shown that flux suppression is highly dependent on array layout and other factors, so the particular value of 7% is only significant to this series of simulations.

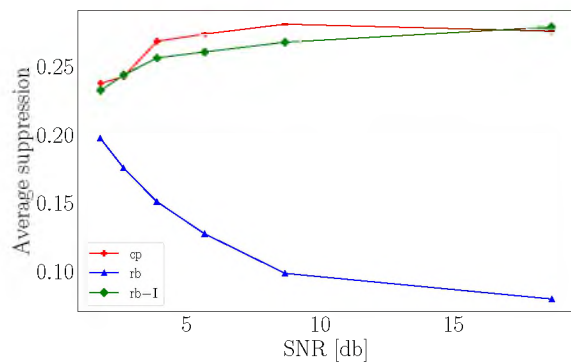
is concerned). The actual degree of improvement is highly dependent on model concentration and SNR. In the extreme regimes, the performance of the solvers appears to converge, so a robust solver may not be worth the extra computational cost. However, as we illustrate in the next section, robust calibration is particularly important for DD calibration, where we are unlikely to operate in a high-SNR regime.



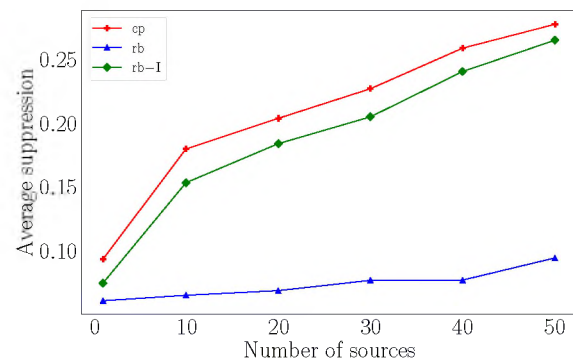
(a) Varying SNR and number of model sources



(b) 1 source model



(c) 50 sources model



(d) Fixed SNR and varying number of model sources

Figure 4.3: The average flux suppression for all the sources in different simulations against the SNR of the data or the number of sources in the model. The red curve is for the complex solver, the blue curve is for the robust solver, and the green curve is the robust-I solver.

4.1.3 Flux suppression in DD calibration

§4.1.2 shows that the robust solver significantly improves calibration in a low SNR regime and with a concentrated model. This is expected to be the case for DD calibration, since direction-dependent model components tend to be both concentrated, and low in SNR. Consequently, in this section, we extend the simulations in §4.1.2 to DD calibration.

4.1.3.1 Simulation setup

We perform two simulations with a similar setup to [Kazemi & Yatawatta \(2013\)](#), illustrating two characteristic regimes of the solvers. The difference between the simulations is the flux level of the sources relative to the thermal noise and the flux level of the unmodelled sources. Henceforth we refer to them as *high-SNR* and *low-SNR*.

The data were simulated using the same setup as before i.e. MeerKAT array configuration with a single frequency channel at 1 GHz with 1 MHz bandwidth, an integration time of 10 seconds and total synthesis time of 2 hours. For the high-SNR simulation, the noise added to the visibilities has an rms of 10 mJy which results in an image noise rms of 6 μ Jy/beam using natural weighting. For the low-SNR simulation, we add noise with an rms of 0.1 mJy (0.06 μ Jy/beam image noise rms). Changing the noise rms may be counter-intuitive, but this is done in order to have high enough SNR for calibration. Additionally, we should note that low-SNR and high-SNR in this context refer to the ratio of the power of the model sources to the power of the unmodelled sources.

For both simulations, we generate sky models containing 100 sources with the positions and fluxes generated as before. In the high-SNR simulation, we scale the fluxes of the sources such that the brightest source has a flux of 20 Jy, while for the low-SNR simulation, we scale the fluxes so that the brightest source has a flux of 0.05 Jy. We choose a peak flux of 20 Jy for the high-SNR regime in order to replicate one of the setups in [Kazemi & Yatawatta \(2013\)](#) where the modelled sources are very bright (i.e. > 5 Jy) and the unmodelled sources are also relatively bright (reaching values even up to 3 or 4 Jy). This simulation is similar to the high-SNR end of [Fig. 4.3b](#). In the low-SNR simulation, we seek to investigate a different regime where fluxes of

model sources are very low and comparable to the faint sky. Here we expect a scenario similar to the low SNR part of Fig. 4.3b.

We assume that the 10 brightest sources are included in the calibration model, and the remaining 90 are unmodelled. We corrupt the 10 brightest sources with DD gains (technically, such gains will affect all sources and not just the brightest ones, but for reasons of computational economy, we restrict DD gains to the modelled sources) and add Gaussian noise to the corrupted visibilities. We apply smoothly varying DD gains generated from a circularly symmetric Gaussian process with a Squared Exponential covariance function with parameters ($\sigma_f = 0.2, l = 300$) for both the amplitude and phase of the gains (see Appendix D).

4.1.3.2 Results

We perform DD-calibration on the corrupted visibilities, with only the 10 brightest sources modelled, using a solution interval of 150 s. Calibration is performed using the complex, robust and robust-I solvers. As in the previous simulation, after calibration, we produce residual images to study the suppression in the fluxes of the unmodelled sources.

We show the results of the high-SNR simulations in Fig. 4.4. Fig. 4.4a shows the recovered flux against the input flux for the different algorithms. We observe that both robust solvers outperform the complex solver, with the robust-I solver producing marginally better results than the robust solver. We show the difference map, i.e. the image recovered by the robust-I solver, minus the image recovered by the complex solver in Fig. 4.4b. The difference image has numerous bright, positive peaks corresponding to the additional flux recovered by the robust-I solver. The low-SNR simulation, by contrast, shows the robust solver produces the best results (see Fig. 4.5). The difference maps in Fig. 4.5b further emphasise this. We also note the negative peaks which occur at the positions of the *modelled* sources in both Fig. 4.4b and Fig. 4.5b. These peaks imply that the complex solver residuals contain *more* flux at the model source positions. Since the modelled sources (with DD-gains applied) have been subtracted from the residual maps, this, in turn, implies that, in the presence of unmodelled sources, the complex solver tends to *underestimate* the modelled sources to a greater extent than the robust solvers.

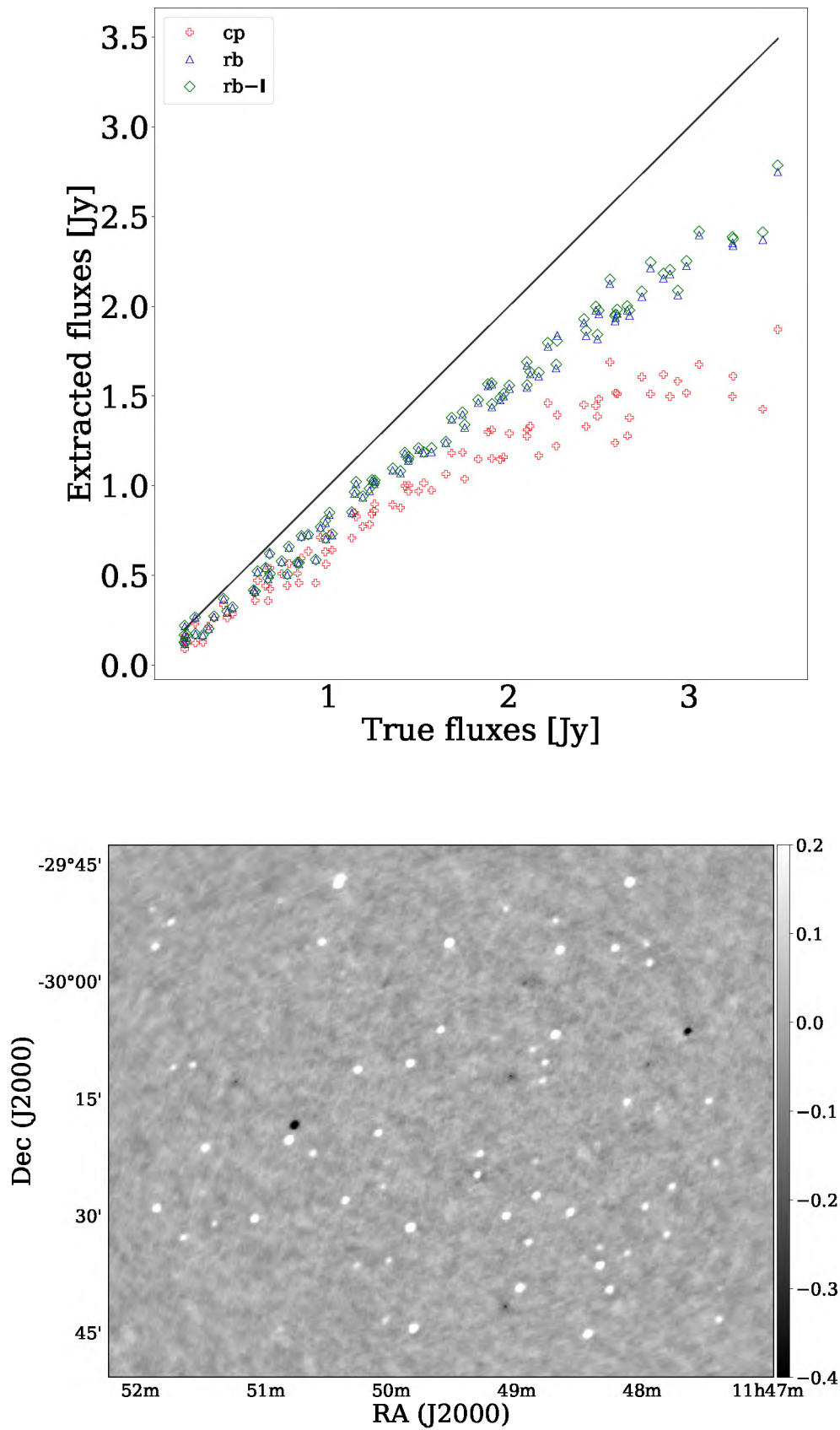
The reason why the robust-**I** solver performs well in the high-SNR simulation compared to the low-SNR simulation is that for the high-SNR simulation, the covariance of the residuals is higher than **I**. Hence, in this simulation, visibilities are adequately weighted. However, for the low-SNR simulation, the variance is over-estimated (true residual covariance is smaller than **I**) effectively assigning equal weights to all visibilities and hence results in a similar performance as the complex solver.

4.1.3.3 Solution intervals

One of the most critical decisions during calibration is the choice of solution intervals. Solution intervals are generally employed to improve the SNR, to make the system of equations overdetermined. Ideally, solution intervals are chosen such that they are shorter than the time and frequency scales of the gains' variability, but long enough to provide significant SNR. For differential gains (or DD calibration), longer solution intervals are thus necessary (since the SNR in per-direction models is lower); somewhat fortuitously, physical intuition suggests that in most regimes, the DD component of the gain (e.g. primary beam rotation) should vary slower in frequency and time relative to the DI component (e.g. atmospheric phase). In order to investigate the effects of solution intervals on gain solutions, we repeat the experiments above while varying the solution intervals.

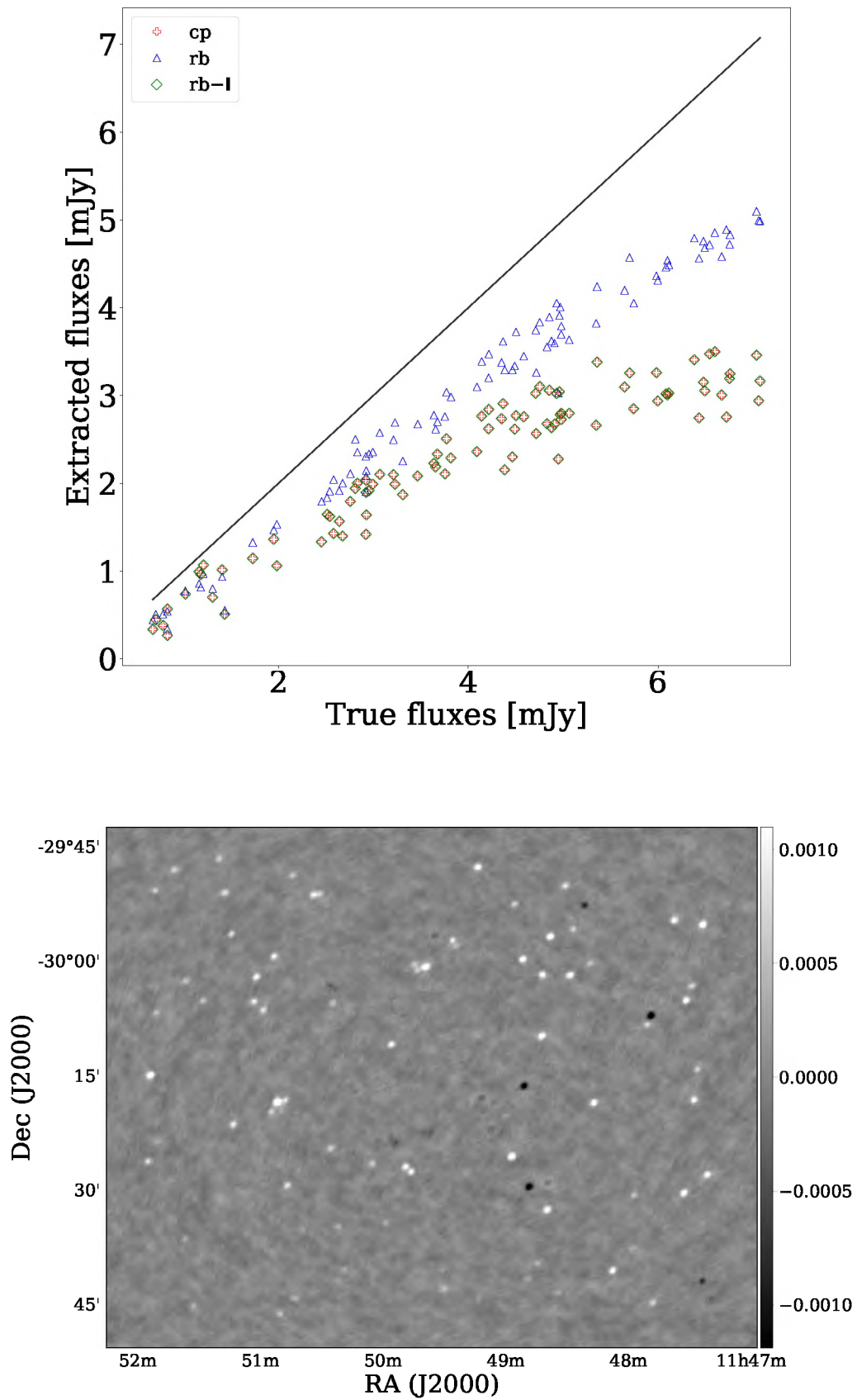
The average suppression (AS) as a function of solution interval is shown in Figs. 4.6a and 4.6b. The figures show that, as the solution interval increases, flux suppression goes down, which is consistent with the results of Nunhokee (2015). At sufficiently large time intervals, all three solvers eventually reach an asymptotic level of flux suppression.

This clearly illustrates the benefits of a robust solver (at least in the sense of lower flux suppression) only kick in in specific regimes. In particular, in the low-SNR regime, if the gains are sufficiently stable for long solution intervals to give acceptable results, the [computationally cheaper] complex solver produces almost equivalent results to the [more expensive] robust solvers. With shorter solution intervals, the robust solvers tend to produce markedly lower flux suppression. We note that gain (in)stability is not the only reason to choose shorter solution in-



(b)

Figure 4.4: (a): Recovered flux against input flux for the high-SNR simulation. (b): Difference map between the corrected residuals of the robust-I solver and the complex solver.

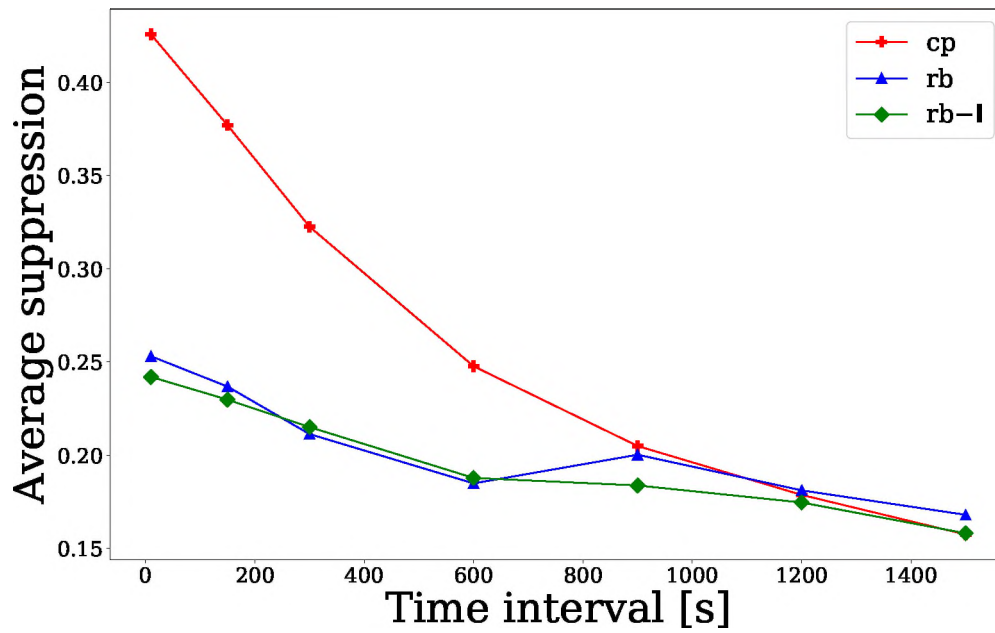


(b)

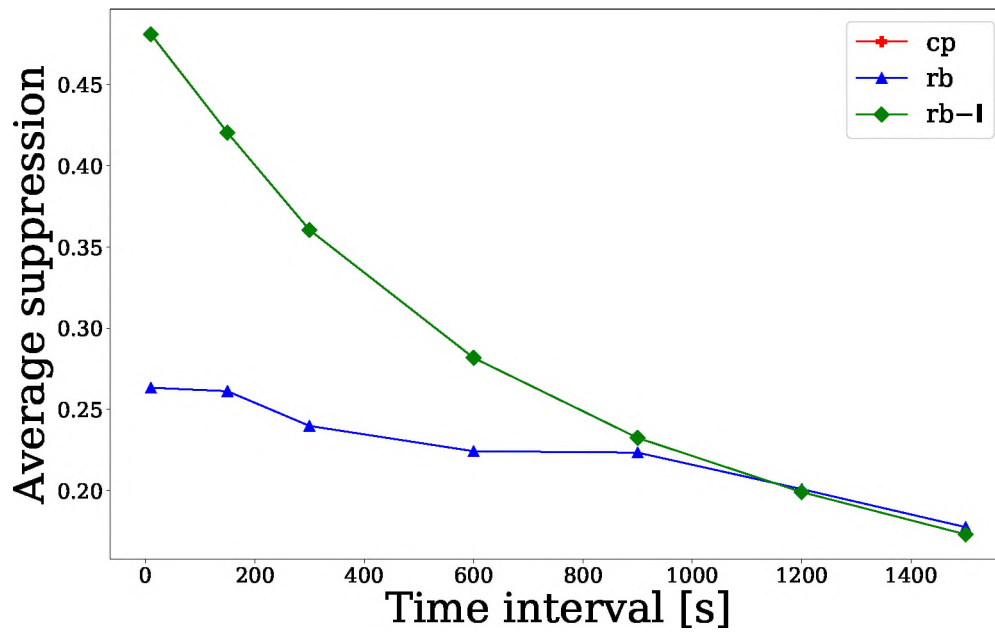
Figure 4.5: (a): Recovered flux against input flux for the low-SNR simulation. (b): Difference map between the corrected residuals of the robust solver and the complex solver.

tervals: there may also be purely operational reasons. In particular, the amount of data produced by new arrays such as MeerKAT (and the future SKA will push this up by orders of magnitude) drives a requirement for data parallelism, while at the same time increasing the memory footprint of existing algorithms. This implies that the data needs to be processed in smaller chunks, thus constraining the size of a practical solution interval for this class of algorithms, and potentially opening a precious niche for robust solvers.³

³For completeness, we should note other approaches to the small-chunk problem, such as consensus optimisation (Yatawatta, 2015b), filtering (Tasse, 2014b) and, recently, stochastic LBFGS (Yatawatta et al., 2019).



(a)



(b)

Figure 4.6: The average flux suppression (AS) across all sources in the simulation. (a): High-SNR regime (b): Low-SNR regime.

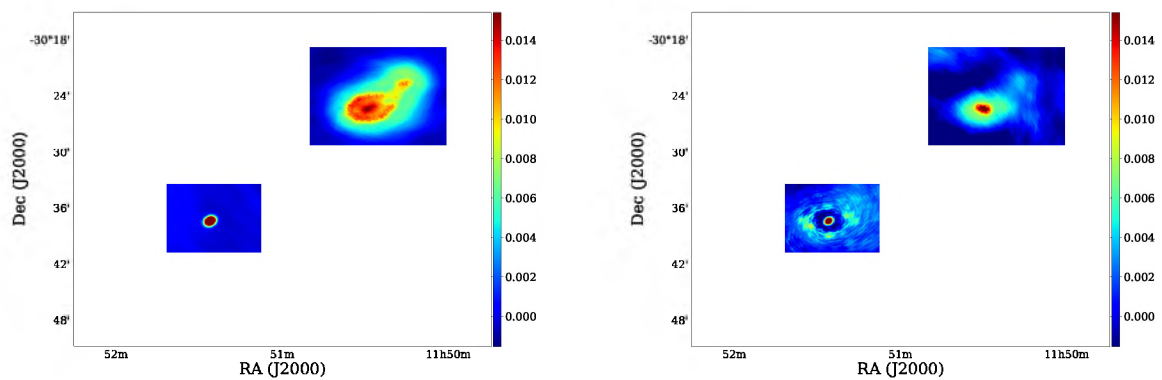
4.1.4 Flux suppression from extended sources

We observed in §4.1.1 that the visibilities from large extended structures only contribute to visibilities of short baselines. A common trick in radio interferometry is to exclude short baselines (uv-cut) during calibration in order to avoid suppressing the emission from extended sources which are complicated to model. By construction, the robust solver will implicitly do this, by giving low weights to the visibilities dominated by the extended emission, hence obviating the need to choose a cutoff baseline length for our calibration. We demonstrate this with the following simulation:

1. Simulate visibilities of a field containing a 0.05 Jy point source at its phase centre and a faint extended emission ($200'' \times 300''$) with peak flux 0.015 Jy at an offset position. We show the image of the simulated field in Figure 4.7a.
2. Calibrate the simulated visibilities using a model consisting of the phase centred point source only with the robust and complex solver. For the robust solver we do not apply any uv-cut. In the case of the complex solver we use baselines length cutoffs of 0 m (i.e not cutoff), 50 m , 80 m, 100 m and 200 m.
3. After calibration subtract the modelled point source and image the residual visibilities. Ideally, this should contain the unmodelled extended source only. The images are shown in Fig. 4.7.

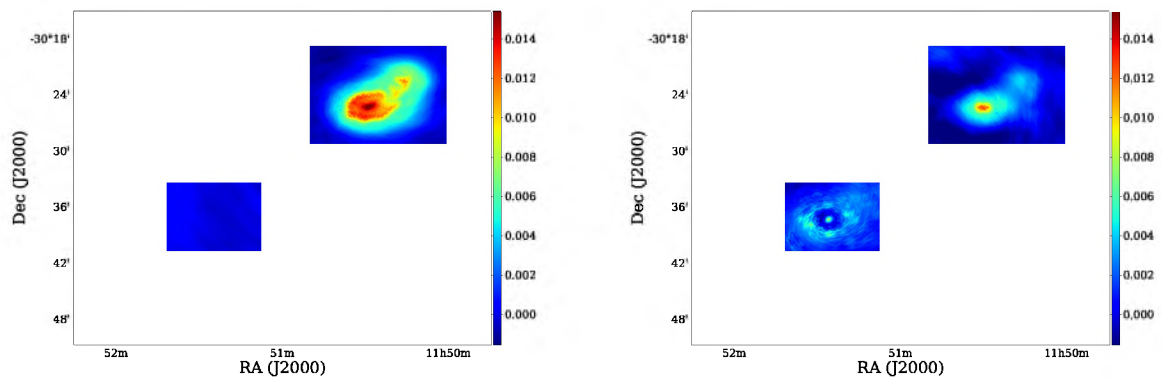
Fig. 4.7c shows that the robust solver performs as expected. The extended source is well recovered even without any baseline cutoff. For the complex solver, we observe a high suppression of the extended emission when no cut-off (see Fig. 4.7b) is applied or when the cut-off is low (see Fig. 4.7d). In these cases we observe, as in the previous simulations, suppression in the flux of the model source due to the unmodelled extended emission. Increasing the cutoff length allows the complex solver to completely recover the extended emission since this is now effectively excluded from the calibration (see Fig. 4.7e and Fig. 4.7f). Hence the robust solver can efficiently be used to preserve the flux of diffuse and extended sources with complicated structures

during calibration. It is also important to note that the uv-cut approach throws away a lot of data, particularly in dense-core arrays like MeerKAT.



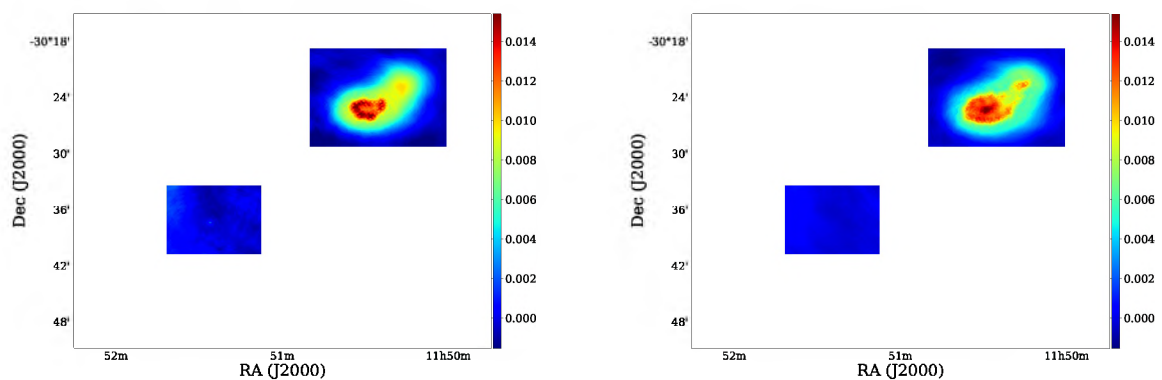
(a) Point and extended source

(b) complex solver



(c) robust solver

(d) complex solver with uv-cut = 50 m



(e) complex solver with uv-cut = 80 m

(f) complex solver with uv-cut = 200 m

Figure 4.7: (a) Image of the simulated field showing both the modelled point source and the unmodelled extended emission. (b), (c), (d), (e) and (f) are residuals after calibration with the modelled point source subtracted showing the recovered extended emission for the robust and the complex solver with the different uv-cut thresholds. All images are plotted with a colormap set to the maximum and minimum of the extended emission, and regions without the point and extended emission have been masked out.

Fig. 4.8 is a plot of the ratio of the total flux of the extended source (i.e. the sum of all non-negative pixels in the region around the source) to its true flux. Fig. 4.8 shows that the optimal cutoff for MeerKAT is ≈ 200 m. For this cutoff length the complex and the robust solver show almost 100 % recovery rate. We also observe a strange drop for a cutoff of 50 m when compared to not applying any cutoff, but we do not have an explanation.

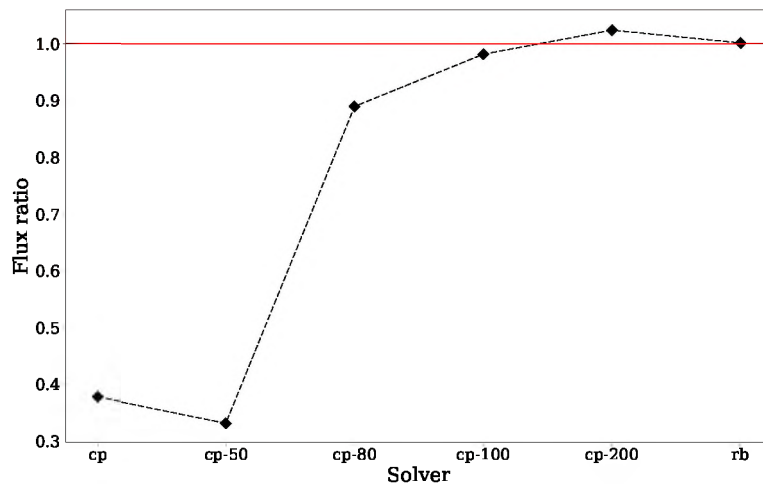


Figure 4.8: Flux ratio of the total recovered flux of the extended source to the true total flux for calibration with the different solvers and uv-cut thresholds. The numbers next to the solver's labels indicates cutoff lengths when uv-cuts are applied. The horizontal red line indicates the flux ratio of 1.

4.2 Robust solvers and RFI mitigation

The robust solver works by iteratively recomputing weights based on how far our modelled visibilities are from the observed visibilities. During calibration, the robust solver will tend to suppress the effect of remaining outliers in data such as those caused by low-level RFI, which is particularly difficult to remove using conventional data flagging. This is conceptually similar to the approach of [Bonnassieux et al. \(2018\)](#), where uncertainties from calibration solutions are used as weights during imaging to reduce the effects of outliers. Note that here, however, the weights from the robust solver shouldn't be used for imaging, as these will tend to suppress the

unmodelled sources. We demonstrate this behaviour in a simulation, and then on real observational data.

The dataset in question is a 1.2 hours 2013 VLA observation of the VIDEO deep field (J2000, RA=02h11m21.09s, Dec=-04d11m13.5s). VIDEO was deliberately chosen as a field relatively free from bright sources (so as to minimise the level of deconvolution and DDE-related artefacts), with the brightest object in the field being only ≈ 0.02 Jy. This particular observation covers a frequency range of 0.9–2.6 GHz, with 16 spectral windows each having 64 channels. The integration time on average is 9 seconds. It employs 28 VLA antennas, with a maximum baseline of 36.4 km. For this experiment, we first transform the measurement set to have a single spectral window by combining all spectral windows. We obtained the data after initial flagging and 1GC calibration using the CASA software (see [Heywood et al. \(submitted\)](#) for more details). We then image the 1GC-corrected data, and extract a component-based sky model using the PyBDSF package ([Mohan & Rafferty, 2015](#)). This sky model is used as a basis for the simulations in this section.

Before testing our solvers on real data, we first discuss the qualitative effects of unflagged RFI on data processing, and present some simulations to illustrate our predictions. For the sake of simplicity, we restrict the discussion and our simulations to stationary terrestrial RFI sources; we note, however, that other types of RFI (e.g. self-RFI, aircraft and satellite RFI) also manifest themselves as outliers in the data (see [Offringa et al. \(2015\)](#) for a few examples).

4.2.1 Simulating low-level RFI

Let's consider a single narrow-band (and, possibly, on/off or time-variable) RFI source. Stationary (terrestrial) RFI sources are fixed with respect to the baselines, and therefore have a nominal fringe rate of zero. Radiation from a stationary RFI source is (as far as the interferometer is concerned, in a given timeslot and frequency channel, and assuming the receiver chain is not saturated by the RFI signal) indistinguishable from a real source at either celestial pole, modulo the primary beam gains, modulo a constant phase offset. Delay tracking in the correlator, being more rapid for longer baselines, consequently imposes a higher fringe rate for such sources on longer

baselines, which attenuates the RFI response on longer baselines due to time and bandwidth averaging.

If we consider only the imaging problem, the net effect of an (unflagged) RFI source is then very similar to that of a bright source at a celestial pole. Images of the target field will be contaminated by a structure that is modulated by the PSF sidelobes of a polar source. For low-level RFI, and a field sufficiently far from a pole, these can be ignored, or even lost in the noise. This is especially true in the case of continuum imaging. One can think of it in terms of *RFI occupancy*: a narrow-band, on/off source contributes to relatively few of the visibilities that go into a Fourier transform (i.e. has low occupancy), therefore its effect on the image is diluted.

The effect on calibration can be far more insidious, particularly if short time/frequency intervals are employed. Within a particular short time/frequency interval, an RFI source can happen to have high occupancy, thus significantly biasing the gain solutions for that interval. In the worst case, the gain solutions are biased low, and applying their inverse then “blows up” some of the corrected visibilities. Let’s consider the following RIME model as an example

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{C}_{pq} \mathbf{G}_q^H + \epsilon_{pq} + \eta_{pq}, \quad (4.3)$$

where \mathbf{V}_{pq} denotes the corrupted visibilities, \mathbf{G}_p represents the gains for antenna p , \mathbf{C}_{pq} is the sky coherency, ϵ_{pq} and η_{pq} are the noise and RFI corruptions respectively. The corrected data after calibration, \mathbf{V}_{pq}^c , is obtained by applying the inverse of the estimated gains, $\hat{\mathbf{G}}$, to the data as follows

$$\mathbf{V}_{pq}^c = \hat{\mathbf{G}}_p^{-1} \mathbf{V}_{pq} \hat{\mathbf{G}}_q^{-H}, \quad (4.4)$$

$$= \hat{\mathbf{G}}_p^{-1} (\mathbf{G}_p \mathbf{C}_{pq} \mathbf{G}_q^H + \epsilon_{pq} + \eta_{pq}) \hat{\mathbf{G}}_q^{-H}. \quad (4.5)$$

Consequently, if the gains of antenna p , for example, are biased low by RFI, the application of their inverses not only amplifies the RFI but also the noise. Since the noise is present on all baselines containing antenna p , the amplified noise now has a high occupancy, and results in strong imaging artefacts.

We now perform a simulation in order to illustrate these arguments. We replicate the “VIDEO” observation by using the MeqTrees package (Noordam & Smirnov, 2010) to simulate the visi-

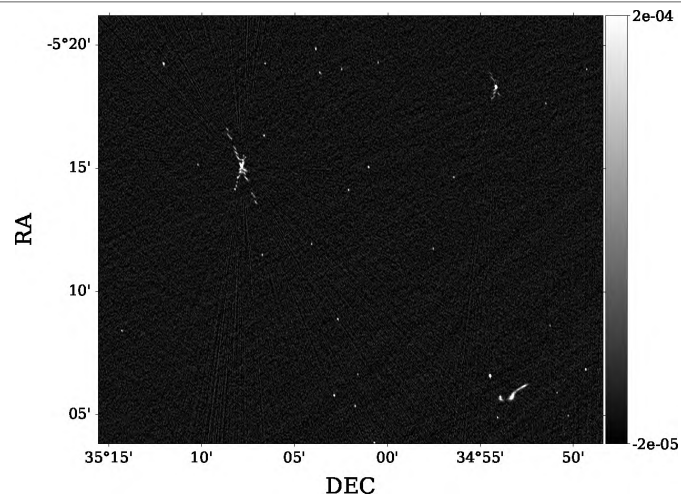
bilities corresponding to the sky model for the “VIDEO” deep field derived above. We then inject a mock low-level RFI source into the data by simulating a 3.5 Jy point source at the South Celestial Pole⁴. We simulate the visibilities corresponding to the RFI source separately, again using MeqTrees, with time and bandwidth smearing enabled, which effectively attenuates power on the longer baselines, as would be expected for a real RFI source. In order to replicate the narrow-band and on-off behaviour of RFI, we inject the simulated RFI visibilities into the simulated sky data at a randomly sampled subset of timeslots and frequency channels. We also add thermal noise at a level of 0.16 Jy (which corresponds to the rms estimated from the real data measurement set). We do not add any other effects to the simulation, as the objective of the experiment is to study the impact of the RFI in isolation.

Having simulated our mock-RFI-contaminated data, we perform full amplitude and phase DI calibration on the data with both the complex and the robust solver using a time interval of 9 seconds and a frequency interval 128 MHz. Fig. 4.9 shows maps of a field-centre patch of the simulated data, as well as the corrected data obtained after calibration with both solvers. In Fig. 4.9a, the RFI source manifests itself as faint linear structure in the background. In Fig. 4.9b, some gain solutions from the complex solver have been biased by RFI, as predicted by the discussion above, resulting in significant image degradation. By contrast, with the robust solver (Fig. 4.9c), no such contamination occurs, as the robust solver effectively excludes the RFI-affected visibilities via its weighting scheme.

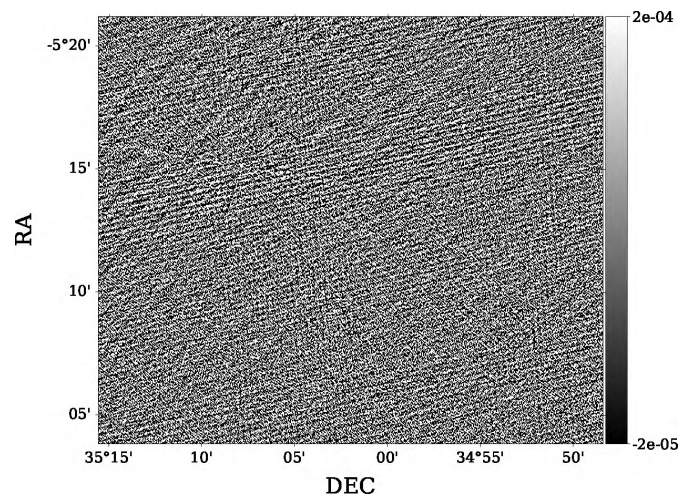
4.2.2 Application to real data

Having demonstrated the success of the robust solver on simulated data we now attempt to calibrate the real VIDEO data set. This data set is an excellent test case because it is a deep field with low SNR and the data is contaminated by low-level RFI which is difficult to remove using conventional flagging. Fig. 4.10 is a waterfall plot of the average visibilities. The bright spots are the visibilities that are corrupted by the low-level RFI. Since the data was already bandpass

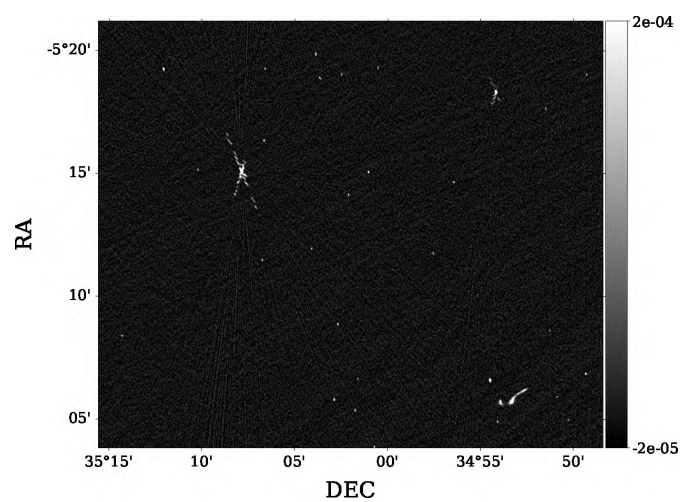
⁴The flux value was deliberately chosen to illustrate the effects above. Note that our simulation does not include primary beam attenuation, so the quoted brightness of the RFI source is unattenuated.



(a) RFI Data



(b) complex solver time-int = 9 secs



(c) robust solver time-int = 9 secs

Figure 4.9: Images of a patch at the centre of the field for simulated RFI-corrupted data, and corrected data after calibration with both solvers. (a) RFI-corrupted data, (b) after calibration using the complex solver with a time interval of 9 secs, (c) after calibration using the robust solver with the same time interval. RFI-induced artefacts are clearly visible in case (b).

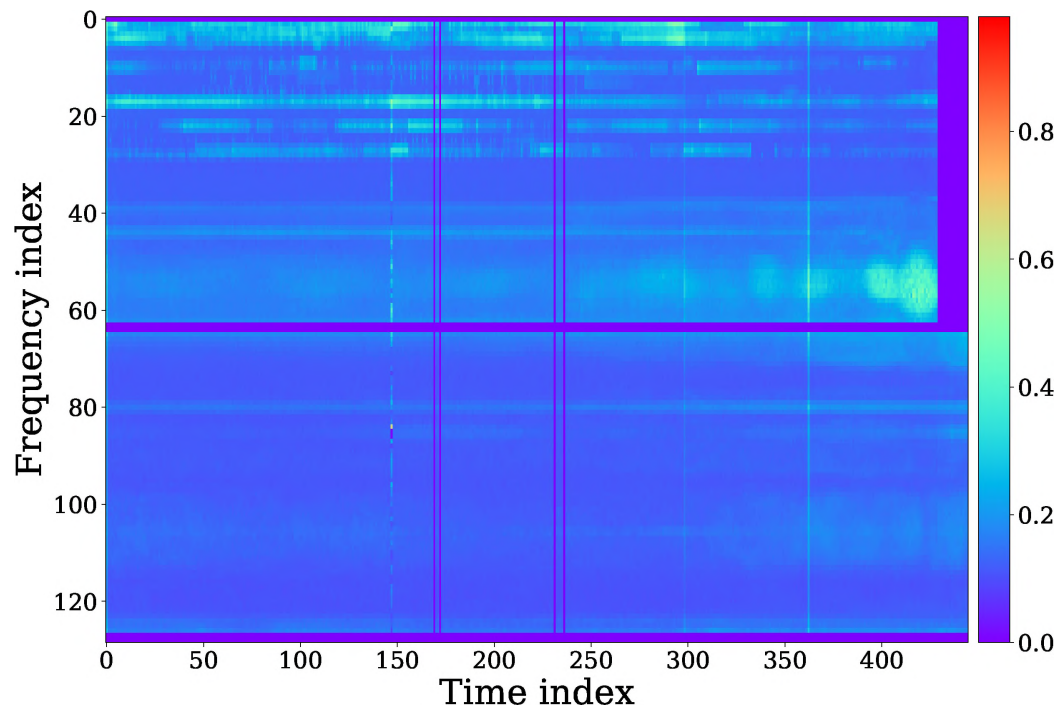


Figure 4.10: A waterfall plot of the visibilities for the VIDEO observation before self-calibration, averaged across all baselines and correlations. This is an image of a chunk of data containing 128 frequency channels. The gains plotted in Fig. 4.12 correspond to the same data chunk. The purple stripes correspond to previously flagged data, and the bright spots correspond to low-level RFI.

calibrated during 1GC, we perform self-calibration with solution intervals of 9 sec and 7.5 min, using both the robust and the complex solver.

Images of the 1GC-corrected and post-2GC data are shown in Fig. 4.11. The strong artefacts present in Fig. 4.11a, which are not visible in the simulated data (see Fig. 4.9a), are caused by the primary beam: as the earth rotates during an observation, the sources move through the beam and produce these artefacts. Fig. 4.11b and 4.11c show that the robust solver removes the beam-related artefacts in the data (see Fig. 4.11a) without introducing more RFI-related artefacts. In contrast, the solutions from the complex solver are similar to the predictions of the RFI simulation. On the other hand, at a time interval of 7.5 mins, both solvers produce good results and the artefacts are effectively removed. A look at the gain plots in Fig. 4.12 provides

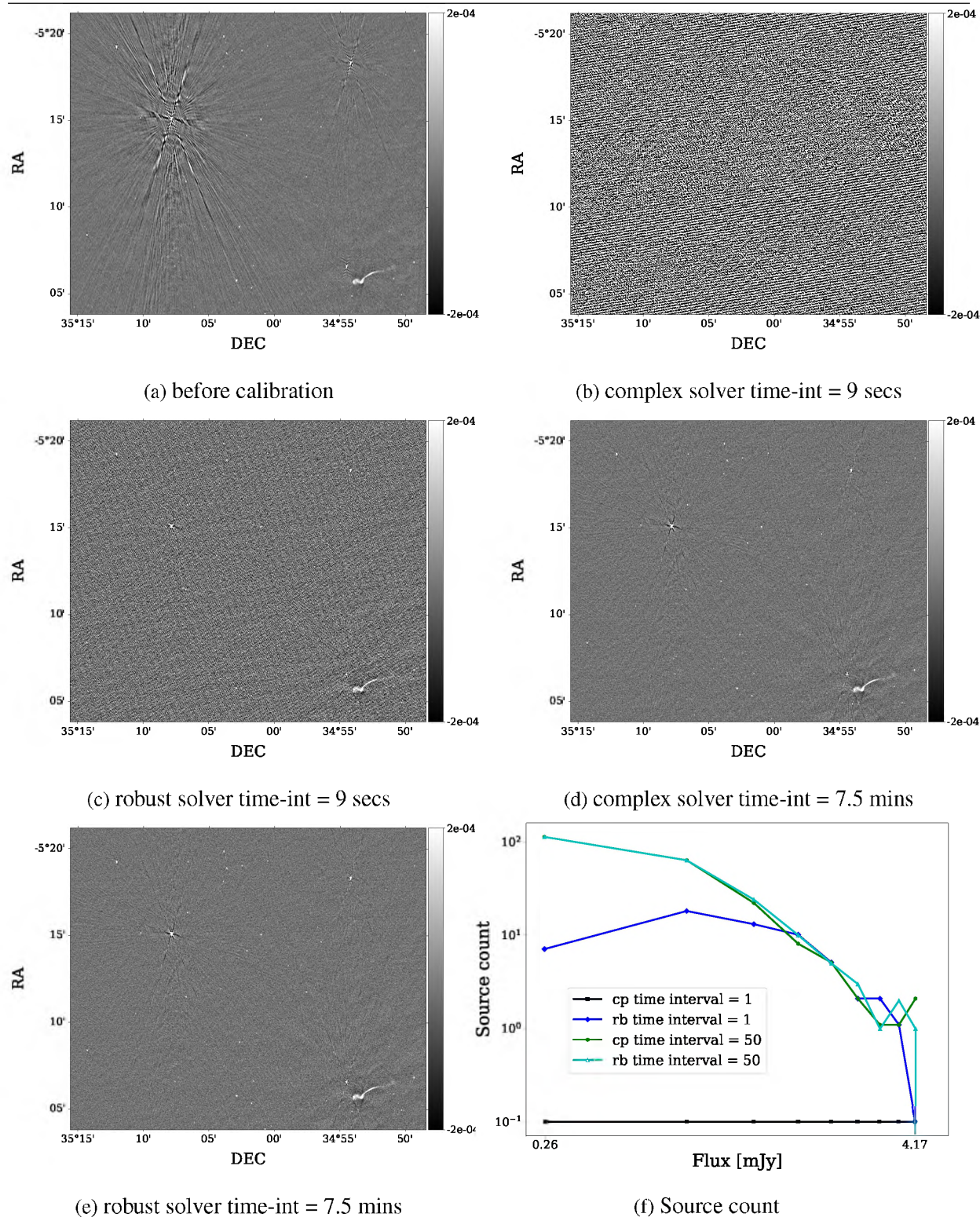


Figure 4.11: An image of the centre of the VIDEO field before and after calibration with different solution intervals: (a) before self-calibration; (b) after calibration using the complex solver with a time interval of 9 secs. The image gets worse because RFI contaminates some gain solutions; (c) after calibration using the robust solver with a time of interval 9 secs. Most artefacts from the uncalibrated image are gone, but the noise level is increased due to the low SNR of the solutions; (d) after calibration using the complex solver with a time interval of 7.5 mins. The RFI-induced artefacts are gone because they have been averaged out by the long solution interval; (e) after calibration using the robust solver with a time interval of 7.5 mins. (f) Source counts, showing no detections for the complex solver-calibrated image with a time interval of 9 secs.

additional insight. At a time interval of 9 secs, the RFI occupancy in some of the time intervals is rather high, leading to biased gain solutions derived from the complex solver. These biased solutions propagate errors into the corrected visibilities, resulting in strong imaging artefacts. The robust solver does not suffer from this effect because very low weights are applied to these visibilities, thus effectively flagging them during the computation of the gains. We observe more noise in Fig. 4.11c: at low SNR, if short time/frequency intervals are used for calibration, the solver fits noise instead of signal and this may result in an increase in the noise level of the corrected data. We elaborate on this subtle trade-off between solution interval width and SNR in §5. Note that as the solution interval increases, the performance of both solvers converges. The RFI contribution is averaged out by the long time interval, so the complex solver is able to perform adequately, as shown by the output gains plot (Fig. 4.12).

Finally, we conclude by presenting Fig. 4.11f, which shows a (log scale) plot of the source counts extracted from the different images. Fig. 4.11f gives us an insight into how such low-level RFI could affect our science. In particular, we don't detect any sources in the calibrated image when we use the complex solver with a time interval of 9 secs. Therefore, low-level RFI needs to be handled properly during calibration, even when it is not immediately obvious in the image domain. This final point is particularly relevant for mJy and μ Jy science targets.

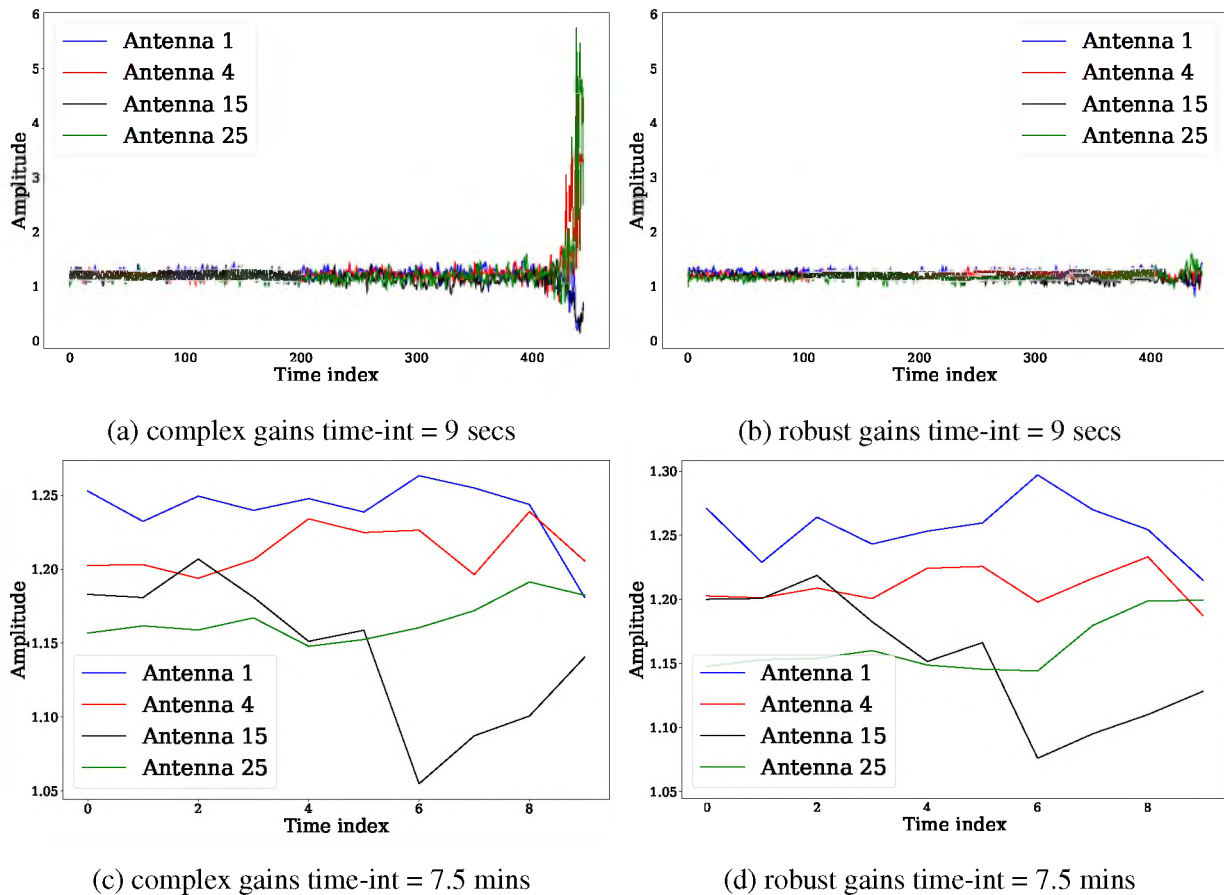


Figure 4.12: The amplitudes of the estimated gains for both solvers with time intervals 9 secs and 7.5 mins. For the complex solver (a) we can see various peaks which are absent for the robust solver (b). These peaks appear exactly at the times where RFIs hugely dominate the visibilities (see Figure 4.10). The robust solver highly attenuates these peaks because of the weighting. (c) and (d) show that with a large time interval of 7.5 mins, the peaks are averaged out for the complex solver.

4.3 Discussion

This chapter and the preceding chapter presented the derivation and implementation of a novel algorithm for RI calibration based on a CST distribution. The different experiments in this chapter attempt to show different scenarios in which using such an algorithm will improve RI

calibration. In this chapter, we mainly focussed on the problem of unmodelled sources and RFI present in visibilities during calibration.

In §4.1.2, we tried to identify different regimes based on the SNR and model concentration where flux suppression is most severe and robust calibration therefore advantageous. The results from this section showed similar performance for the robust solver and the traditional solver in high SNR regimes and for concentrated models. However, the robust solver hugely outperforms the traditional solver in low SNR regimes and for highly dispersed models. This result suggests that the robust solver, which is computationally more expensive, should be considered in low SNR regimes and for highly dispersed models.

The indications from §4.1.2 were confirmed in §4.1.3 and §4.1.4 based on the results of the two DD calibration simulations and the extended source simulation. Hence, a first take away from this chapter is that we can use the robust solver when performing DD calibration on faint sources. The performance of the robust-I solver in the high-SNR DD simulation implies that down-weighting the residuals covariance matrix can improve calibration, this requires further investigation. In §4.2, we showed that the robust solver could be used efficiently to mitigate against unflagged RFI during calibration. Both the RFI and flux suppression result showed the significance of solution intervals during calibration, and we will further investigate this in the next chapter.

Beside optimising, profiling and testing our current implementation, an interesting research from this chapter will be to extend the analysis in §4.1.1. Different probability distributions (heavy-tailed and non-heavy-tailed) could be fitted to corrupted and uncorrupted visibilities using a Bayesian approach. Then comparing the different Bayesian evidence would provide more statistical evidence on what is the best probability distribution to consider for the modelling of errors during calibration.

Solution Intervals Considered Harmful¹

In Chapter 4, we highlighted how the choice of solution interval could improve flux suppression and reduce the effects of RFI during calibration. This interval is usually selected based on expertise using heuristic approaches. Often calibration has to be repeated several times until we find an adequate interval. This chapter presents an extensive study of the effects of solution intervals on calibration and imaging outputs. The main focus of the chapter is to describe different subtle factors which make the selection of an optimal solution interval problematic. We start by first reintroducing the calibration problem and discuss the concept of solution intervals used for gain parametrisation by most calibration packages in §5.1. This parametrisation is further analysed in §5.2 where we discuss the main factors which influence the choice of optimal solution intervals. Because solution intervals are the most widely used regularisation tool during calibration, we propose in §5.3 a practical algorithm for finding the optimal solution interval during calibration. The proposed algorithm is validated using simulations in §5.4 and then applied to VLA data in §5.5 to demonstrate its advantages. We conclude in §5.6 with more discussions on the difficulties of finding an optimal interval in practice and the possibility of doing calibration without solution intervals.

¹The work presented in this chapter will be submitted for publication as [Sob et al. \(in preparation\)](#)

5.1 Problem overview

This section presents an overview of the problem we want to address in this chapter, i.e. how to select the optimal solution interval during calibration? We first review some calibration concepts already discussed in Chapter 2 and Chapter 3 and then provide a concise definition of solution intervals and why they are necessary during calibration.

5.1.1 Calibration

In this chapter, we restrict the discussion to DI calibration where the aim is to solve for the time and frequency dependence of the gains only. In this case, the measurement model, or RIME, has the following simplified representation,

$$\mathbf{V}_{pq} = \mathbf{G}_p \mathbf{C}_{pq} \mathbf{G}_q^H + \epsilon_{pq}, \quad (5.1)$$

where \mathbf{C}_{pq} is the sky coherency term, \mathbf{V}_{pq} denotes the measured visibilities, \mathbf{G}_p is the DI gain of antenna p and we assume that the noise, ϵ_{pq} , follows a circular complex Gaussian distribution. Note that we do not preclude the possibility of absorbing any *known* DDEs into the definition of \mathbf{C}_{pq} . Furthermore, we assume that the noise for the correlations are i.i.d. so that

$$\mathbf{r}_{pq} = \text{vec}(\epsilon_{pq}) = \text{vec}(\mathbf{V}_{pq} - \mathbf{G}_p \mathbf{C}_{pq} \mathbf{G}_q^H) \sim \text{CN}(\mathbf{0}, \Sigma_{pq}) \quad (5.2)$$

where $\mathbf{0}$ is a 4×1 zero vector, Σ_{pq} is a diagonal 4×4 covariance matrix.

The discussion that follows is most relevant after 1GC has been performed and the gains have been transferred to the target field. Hence we assume we have an initial (partially complete) sky model with which to compute \mathbf{C}_{pq} . Since (in the absence of auto-correlations) an array consisting of N_a antennas measures $N_a(N_a - 1)/2$ pairs of visibilities for each observational time and frequency stamp, the DI calibration problem amounts to solving an overdetermined system for N_a complex unknowns for each measured correlation. Following the discussions in Chapter 2 and Chapter 3 we aim to solve the following NLLS problem:

$$\min_{\boldsymbol{\theta}} \sum_{pq|p < q} \|\mathbf{r}_{pq}\|_F^2, \quad (5.3)$$

where the gains are parametrised using $\boldsymbol{\theta}$.

5.1.2 Solution Intervals

Solving for an independent gain at each time and frequency, especially at low SNR, usually results in overfitting as the reconstructed gains will not be smooth functions of time and frequency. Any algorithm that solves for an independent gain at each time and frequency will be fitting noise as well as the signal. The most common way to regularise the calibration problem is to introduce solution intervals over which the gains are assumed to be constant. This corresponds to having multiple measurements for each baseline across the chosen time and frequency intervals. The optimisation problem, Eq. (5.3), is then slightly modified as follows

$$\min_{\boldsymbol{\theta}} \sum_{pq|s|p<q} \|\mathbf{r}_{pq|s}\|_F^2, \quad (5.4)$$

where s is an index for all the time and frequency samples that are part of the same solution interval. If the full domain of the problem is a $N_t \times N_\nu$ time/frequency grid, the solution interval approach parametrises the gains as a sum of boxcar functions, i.e.

$$\mathbf{g}_p(t, \nu, \boldsymbol{\theta}_p) = \boldsymbol{\theta}_{p,ij} \Pi_{t_i, t_i + \Delta_t}(t) \Pi_{\nu_j, \nu_j + \Delta_\nu}(\nu), \quad (5.5)$$

where $\boldsymbol{\theta}_{p,ij}$ is a 4×1 vector containing the value of the gain for antenna p in an interval labelled by ij , Δ_t is the resolution of the coarsened time grid, Δ_ν is the resolution of the coarsened frequency grid and the boxcar function is defined as

$$\Pi_{a,b}(x) = \begin{cases} 1 & \text{if } x \in [a, b), \\ 0 & \text{otherwise.} \end{cases} \quad (5.6)$$

The units that discretise our grid are the integration time, δ_t , and the channel width, δ_ν . Each solution interval therefore contains a multiple of δ_t and δ_ν which we denote by n_t and n_ν ² such that

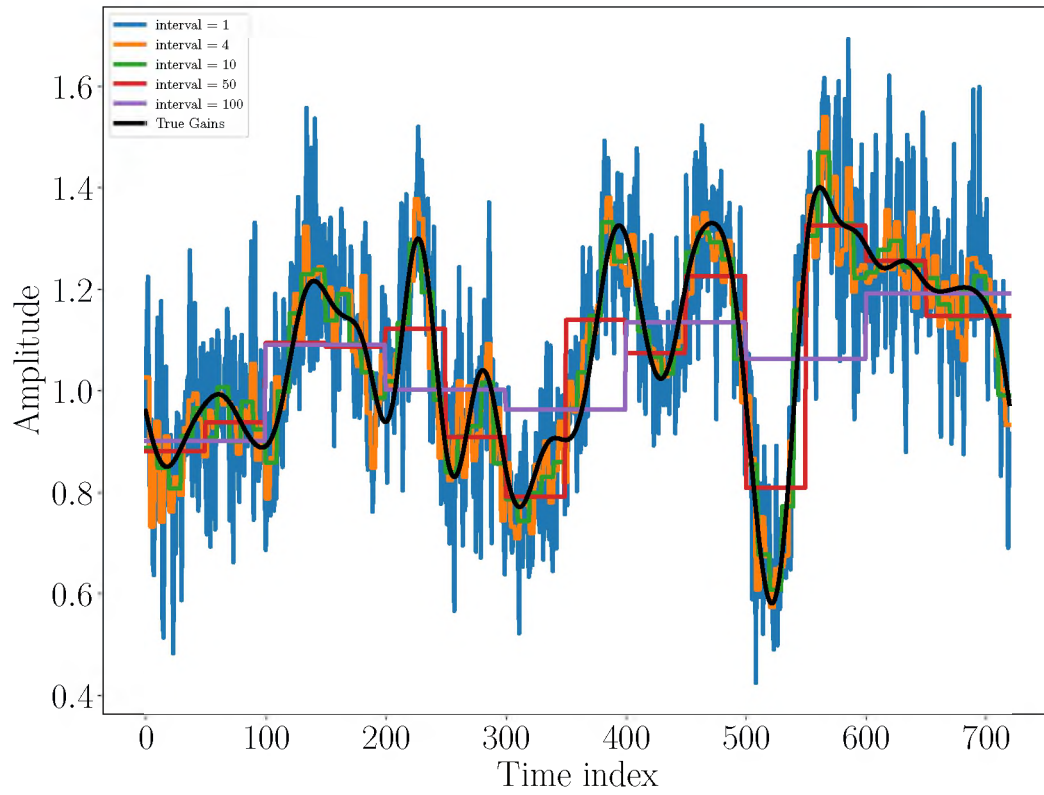
$$n_t = \frac{\Delta_t}{\delta_t} \quad \text{and} \quad n_\nu = \frac{\Delta_\nu}{\delta_\nu}. \quad (5.7)$$

²Throughout the text, when employed without units, the term solution interval refers to the discrete units, n_t and n_ν .

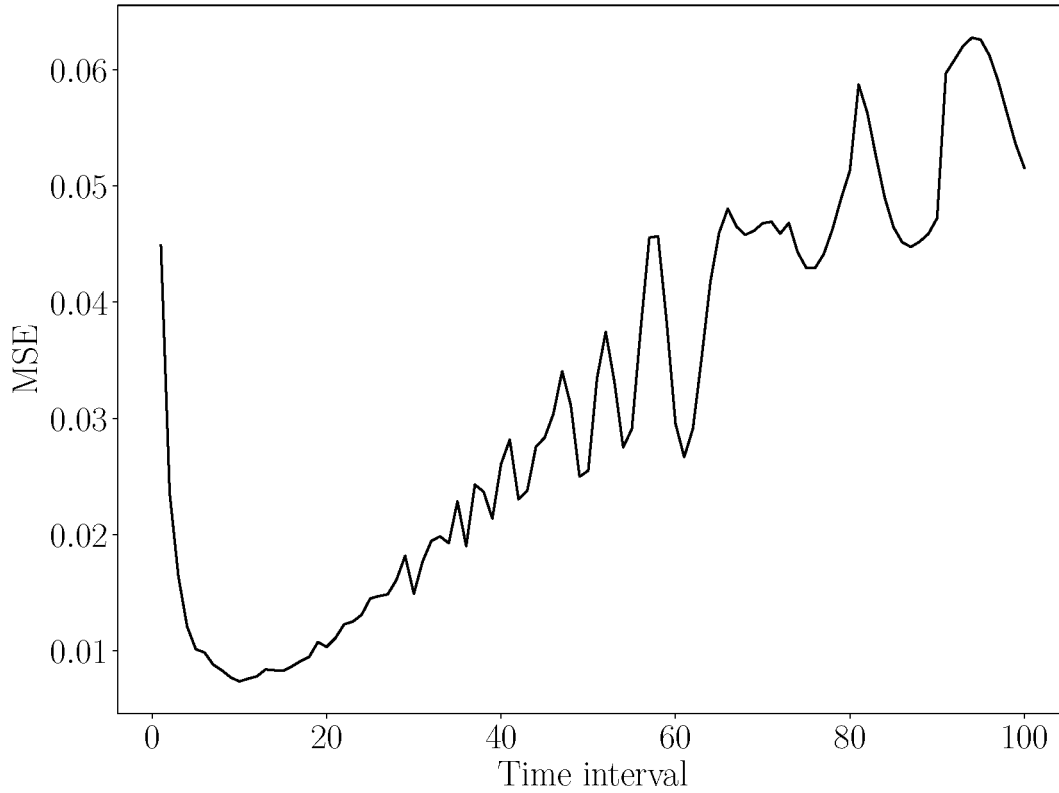
Thus, if the full grid consists of $N = N_t N_\nu$ points, and each solution interval contains $n = n_t n_\nu$ such points, there will be a total of $M_t = \left\lceil \frac{N_t}{n_t} \right\rceil$ time intervals and $M_\nu = \left\lceil \frac{N_\nu}{n_\nu} \right\rceil$ ³ frequency intervals giving a total of $M = M_t M_\nu$ parameters for each gain.

To gain more intuition, let's assume we are calibrating a dataset with a low SNR. If we use a single discrete time interval, $n_t = 1$, we will mostly fit noise, and the estimated gains will be noisy as depicted by the blue curve in Fig. 5.1a. Hence we need to increase the size of the interval to reduce the variance of the gain solutions as depicted by the red, orange and green curves in Fig. 5.1a. Fig. 5.1b shows the mean squared error (MSE) between the true gains and the noisy estimated gains at different intervals. When we increase the interval, we reach a turning point where the MSE in the gains starts increasing because we can no longer keep track of the short scale variations of the gains.

³ $\lceil \cdot \rceil$ is the ceil function defined such that $\lceil x \rceil$ is the smallest integer $\geq x$.



(a) Plot of boxcar reconstructed gains with different intervals illustrating the definition of solution intervals.



(b) Mean squared error (MSE) of the boxcar reconstructed gains shown in Fig. 5.1a at different solution intervals relative to the true gains.

Figure 5.1

The error in the estimated gains during calibration can thus be explained as being the contribution of two error terms: one from the uncertainties in the data, and the second from the solution interval boxcars approximation of the gains. Using the definition of the MSE of a parameter, θ , and its estimate, $\hat{\theta}$, we have

$$\text{MSE} = \mathbb{E}[(\hat{\theta} - \theta)^2], \quad (5.8)$$

$$= \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2] + (\mathbb{E}[\hat{\theta}] - \theta)^2, \quad (5.9)$$

$$= \text{Var}(\hat{\theta}) + \text{Bias}(\hat{\theta}, \theta)^2, \quad (5.10)$$

where \mathbb{E} is the expectation operator with respect to $\hat{\theta}$. The first term of Eq. (5.10) is the variance of the estimated parameter and represents the error in the estimated parameter due to the uncertainties in the measured data. This corresponds to the error in the gains because of the noise in the visibilities, unmodelled sources and RFI for the calibration problem. The second term is called the Bias and it results from the information lost when we approximate the gains as a sum of boxcars when using solution intervals. The focus of this chapter is to provide guidelines and discuss the difficulties on how to correctly identify the turning point of Fig. 5.1b, i.e. how long can we make solution intervals during calibration in order to have the highest SNR possible while still capturing time and frequency variations in the gains.

Note that, despite being suboptimal, the solution interval approach has some advantages over other parametrised approaches. Most notably, the problem is easy to formulate, even in the fully polarised case. Formulated in this way, the problem is also highly distributable. That said, piece-wise constant gains are not physical and finding the optimal solution interval is non-trivial. In principle, it is possible to parametrise gains as smooth functions (e.g. low order polynomials) of time and frequency (see for example Tasse (2014b); Yatawatta (2015a)). As with the solution interval approach, these methods also suffer from a discrete model selection problem. For example, using a polynomial prior, the order of the polynomial (and perhaps the mean and variance of the polynomial coefficients) would have to be specified in advance. Because of the computational advantages offered by the solution interval approach, we will not delve much further into these alternatives.

5.2 Gain Errors

Before moving on to discuss how to find optimal solution intervals, we first elaborate on the different factors that contribute to gain errors during calibration. In this section, we use results from simulations to describe how these factors (i.e. the noise in the visibilities, the intrinsic variability of the gains, the degree of model incompleteness and RFI) affect calibration and how their effects vary with solution intervals. In order to make the presentation simple, we summarise the setups for all the simulations in Table 5.1 (see Appendix D for more details).

Identifier	Setup
i	Sky model: 100 sources (random positions) Peak flux = 0.5 Jy Input rms = 1 Jy (per-visibility) Input gains from a GP ($\sigma_f = 0.2, l = 100$) Single channel MS a) Calibrate with the complete sky model b) Calibrate with an incomplete sky model (50 % of total flux). Peak unmodelled flux = 0.05 Jy
ii	Sky model: 1 source (phase centre) Input rms = 2 Jy (per-visibility) No input gains, i.e. unity gains Array = KAT-7, VLA, VLA-14 and MeerKAT Single channel MS
iii	Sky model: 2 sources (1 at the phase centre and at 0.2 deg offset) Fluxes = 0.05 Jy each Input rms = 0.5 Jy (per-visibility) Array = MeerKAT Single channel MS No input gains, i.e. unity gains a) Calibration with the phase centre source only
iv	Sky model: 1 source (phase centre) Input rms = 0 Jy (no noise added) Input gains from a GP ($\sigma_f = 0.5, l = 200$) Array = MeerKAT
v	Sky model: 100 sources (random positions) Peak flux = 1 Jy Input gains from a GP Input rms and GP parameters in Table 5.2 Single channel MS

Table 5.1: Summary of the parameters for the different simulations. Note that certain simulations are repeated with calibration performed using complete and incomplete sky models. We use a square exponential covariance function for the GP gains. The length scale employed is in units of seconds, for example, $l = 100$ implies the gains vary on scales of 100 secs, i.e. ten units of integration time (10 secs). We use the same parameters for the amplitude and phase of the gains.

Simulation	σ_f	l	σ_{rms}
a	0.3	100	0.2 Jy
b	0.1	200	0.8 Jy
c	0.1	100	2 Jy
d	0.3	100	0.6 Jy

Table 5.2: GP parameters used for the time only simulations (§5.4.1) and the rms of the noise added to the visibilities.

Each simulation is referred to as **Simulation-X** where **X** is the corresponding identifier in Table 5.1. We generate all the gain corruptions using a Gaussian Process (GP). The full details on how we generate these gain corruptions are presented in §D.2. Additionally, all the combinations of GP parameters and input rms used in **Simulation-v** are given in Table 5.2.

Simulation-i provides a general summary of the effects of solution intervals on calibration outputs. Fig. 5.2a and Fig. 5.2b illustrate how the MSE of the estimated gains varies with solution intervals. Increasing the solution interval improves the SNR, by reducing the noise in the visibilities, but when the interval becomes too large, it becomes impossible to track the gain variations. The MSE then increases to an asymptotic value which depends on the gains' intrinsic variability. Another quantity we can look at is the noise contribution from the calibration errors. Hence, we also show in Fig. 5.2a and Fig. 5.2b the rms of the artefact maps, r_{pq}^Δ , which we define as

$$r_{pq}^\Delta = \hat{\epsilon}_{pq} - \mathbf{n}_{pq}, \quad \hat{\epsilon}_{pq} = \mathbf{V}_{pq} - \hat{\mathbf{G}}_p \mathbf{C}_{pq} \hat{\mathbf{G}}_q^H, \quad (5.11)$$

where $\hat{\mathbf{G}}$ denotes the solution to Eq. (5.3), and \mathbf{n}_{pq} is the exact noise realisation added to the simulated visibilities, i.e. we subtract the exact input noise from the uncorrected residual visibilities to isolate the artefacts. The artefact rms follows a similar profile as the error in the gains but increases to larger values at longer intervals. We plot in Fig. 5.2c and 5.2d ratios of the fluxes recovered after calibration to the true fluxes for two sources namely:

src 0 – the brightest source in the field (which has a flux of 0.5 Jy),

src 1 – the brightest unmodelled source in the field (which has a flux of 0.05 Jy) for the incomplete

sky model case.

For the complete sky model, we slightly overestimate the fluxes of the sources at the shortest interval because of the low SNR, but as we move to longer intervals, we start observing source suppression. For the incomplete sky model, the brightest source has a similar flux plot as in the case with a complete sky model, but here, source suppression in its flux is observed earlier and gets worse with increasing solution interval. On the other hand, the unmodelled flux has a more complicated flux profile. Initially, the flux is largely suppressed, the suppression reduces with the solution interval, and then suppression starts increasing again.

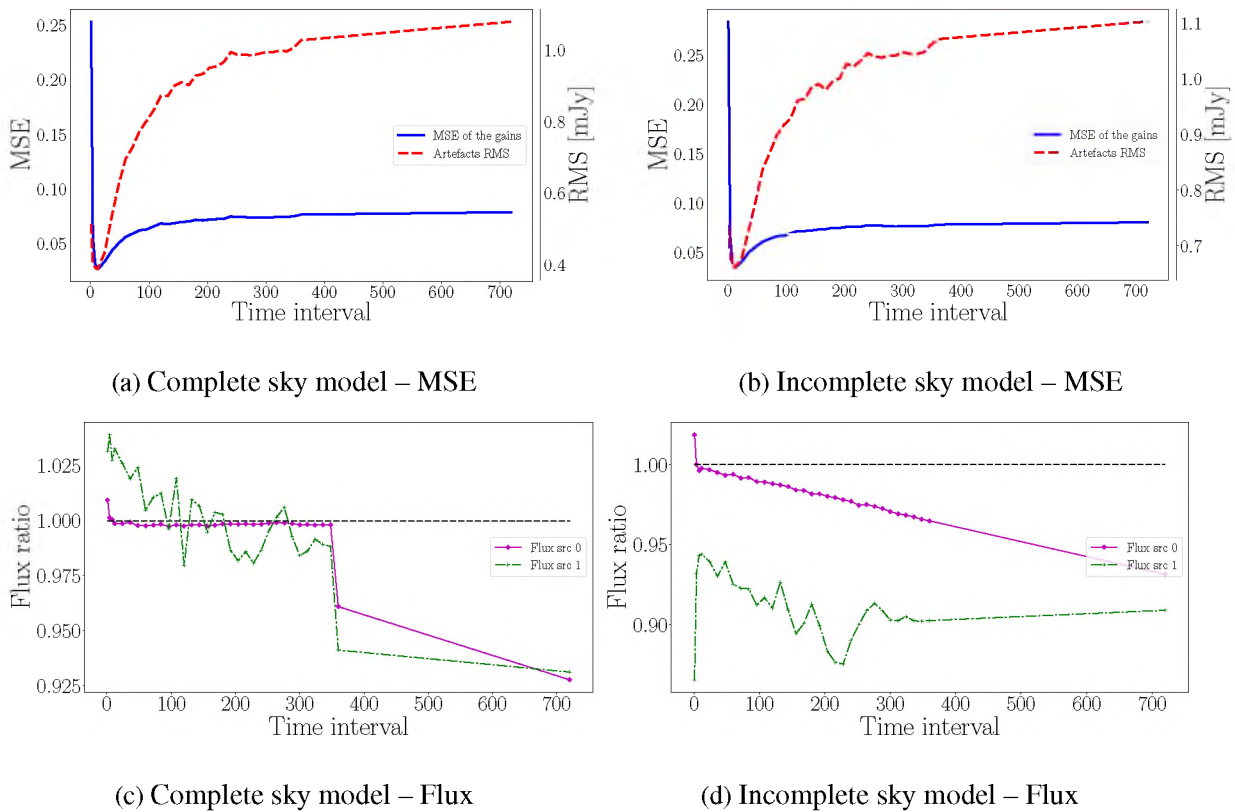


Figure 5.2: (a) and (b) are plots of the MSE of estimated gains (blue solid line) against the solution interval used on the left y-axis and the rms of the artefact images (red dashed line) on the right y-axis. These plots illustrate the trade-off between the error due to noise on the visibilities and the intrinsic variability of the gains as we increase the solution interval. There is a strong correlation between the MSE and the artefact rms. When the sky model is complete, the minimum rms occurs at approximately the same point as the minimum MSE, which is not guaranteed when the sky model is incomplete. (c) and (d) are plots of the ratio of recovered to the actual fluxes for the brightest source (magenta) and a faint source (green) respectively. The black dashed line indicates the flux ratio of 1 for perfect reconstruction.

Isolating and understanding the effects of solution intervals is not a straightforward task, but Fig. 5.2 summarises most of what can happen as a result of solution intervals. From the simulations, long solution intervals which do not capture the variations in the gains could cause a flux suppression of up to $\approx 10\%$ even for modelled sources, while the variance in the gains due to the solution interval at low SNRs could increase the image rms by a factor of up to 3. Martí-Vidal & Marcaide (2008), for example, describes the formation of spurious sources when phase self

calibration is done at low SNR and even derives an approximate expression for the fluxes of the spurious sources as a function of solution interval and number of antennas.

An appropriate definition for the optimal solution interval should then be one which minimises source suppression and overestimation while maximising dynamic range. Based on the above discussion, we define the optimal solution interval as the one which minimises the rms of the artefact image. This metric is only available for simulations. For the rest of the chapter, we will use the MSE as a reference metric, and we will show in §5.3.2 that it is possible to derive a nearly equivalent metric which we can efficiently compute in practice.

5.2.1 Noise in the visibilities

Variance of estimated gains

In this section, we use the Fisher Information matrix (FIM) introduced in the early 1920s by [Fisher et al. \(1920\)](#) to find an expression for the minimum expected variance of the error in gains resulting from the noise in the visibilities during calibration. In simple terms, the FIM can be thought of as the amount information from an unknown parameter than be obtained from a measured data. Given an optimisation problem

$$\min_{\boldsymbol{\theta}} \sum_t |\mathbf{y}_t - \mathbf{f}_t|^2,$$

where \mathbf{y}_t , \mathbf{f}_t and $\boldsymbol{\theta}$ are the measured data, the model and the unknown parameter vector respectively, if we assume a Gaussian likelihood function, then the elements of its FIM, \mathbf{F} , are given by

$$\mathbf{F} = \sum_t \frac{1}{\sigma^2} \frac{\partial \mathbf{f}_t}{\partial \boldsymbol{\theta}_i} \frac{\partial \mathbf{f}_t}{\partial \boldsymbol{\theta}_j}, \quad (5.12)$$

where σ is the standard deviation of the errors in the measured data and t is a time index or data count index (the Gaussian noise approximation only holds for synthesis arrays like MeerKAT and VLA where the noise on each visibility is mutually independent, but this is generally not the case for aperture and phased arrays). From the Cramer-Rao bound (see [Jansen & Claeskens \(2011\)](#)), the inverse of the FIM is a lower bound on the variance of the estimated parameters. In

the simplest case of DI calibration of unpolarised single channel visibilities, the RIME is given by

$$\mathbf{v}_{pqt} = \mathbf{g}_{pt} \mathbf{m}_{pqt} \mathbf{g}_{qt}^* + \epsilon, \quad (5.13)$$

where \mathbf{m}_{pqt} is the coherency matrix, \mathbf{v}_{pqt} the measured visibilities, \mathbf{g}_{pt} is the gain of antenna p at time t . Let us assume a classical dish such as MeerKAT, operating in the frequency range where the noise is dominated by the system contribution rather than the sky ($T_{sys} \gg T_{sky}$). The noise can then be seen as independent Gaussian:

$$\epsilon \sim \text{CN}(0, \sigma_{\text{rms}}^2).$$

For a given solution time interval, n_t , using Eq. (5.12), the diagonal elements of the Fisher matrix for this process are given by

$$F_{pp} = \frac{1}{\sigma_{\text{rms}}^2} \sum_{t=1}^{n_t} \sum_{p \neq q}^{N_a} (\mathbf{m}_{pqt} \mathbf{g}_{qt}^*)^2, \quad (5.14)$$

where N_a is the number of antennas in our array. Assuming that the Hessian matrix is diagonally dominant, as in the preceding chapters, the inverse of the Fisher matrix becomes $F_{ij}^{-1} = \frac{1}{F_{ij}}$ (for all non zero entries). Hence

$$F_{pp}^{-1} = \text{Var}(\hat{\mathbf{g}}_p) = \frac{\sigma_{\text{rms}}^2}{\sum_{t=1}^{n_t} \sum_{p \neq q}^{N_a} (\mathbf{m}_{pqt} \mathbf{g}_{qt}^*)^2}. \quad (5.15)$$

Eq. (5.15) can be simplified if we consider a field with a single source having flux S located at the phase centre with constant gains $1 + 0j$. Using these simplifications, the error in the estimated gains resulting from the noise in the visibilities is given by

$$\text{Var}(\hat{\mathbf{g}}_p) \approx \frac{\sigma_{\text{rms}}^2}{n_t(N_a - 1)S^2}. \quad (5.16)$$

Additionally, for a multichannel observation with solution frequency interval, n_ν , Eq. (5.16) becomes

$$\text{Var}(\hat{\mathbf{g}}_p) \approx \frac{\sigma_{\text{rms}}^2}{n_\nu n_t (N_a - 1) S^2} = \frac{\sigma_{\text{rms}}^2}{n (N_a - 1) S^2}, \quad (5.17)$$

where $n = n_v n_t$. Eq. (5.17) is an already known result in the field of radio interferometry and is stated in Taylor et al. (1999) in the following forms for phase-only and full-complex calibration respectively

$$\text{Var}(\hat{\mathbf{g}}_p) = \frac{\sigma_{\text{rms}}^2}{n(N_a - 2)S^2} ; \text{Var}(\hat{\mathbf{g}}_p) = \frac{\sigma_{\text{rms}}^2}{n(N_a - 3)S^2}.$$

The factor $N_a - 1$ is replaced by the factors $N_a - 2$ and $N_a - 3$ in order to account for the extra degrees of freedom required for phase-only and full-complex calibration respectively. In what follows and in the rest of this chapter, we will use Eq. 5.17 with the factor $N_a - 1$ as an estimated of the expected variance on gains solutions resulting from the noise the visibilities.

We check the validity of Eq (5.17) using **Simulation-ii**. VLA-14 in Table 5.1 denotes a sub-array made using only 14 of the 27 VLA antennas. Fig. 5.3 is a plot of the MSE against solution interval for the different datasets. The plot shows that the estimated MSE are in close agreement with Eq. (5.17). These plots show the importance of having a large number of antennas. The predicted and measured MSE is close for MeerKAT even at the smallest time interval of 1. For the VLA and KAT-7 arrays, a much longer time interval is required to match theoretical expectations.

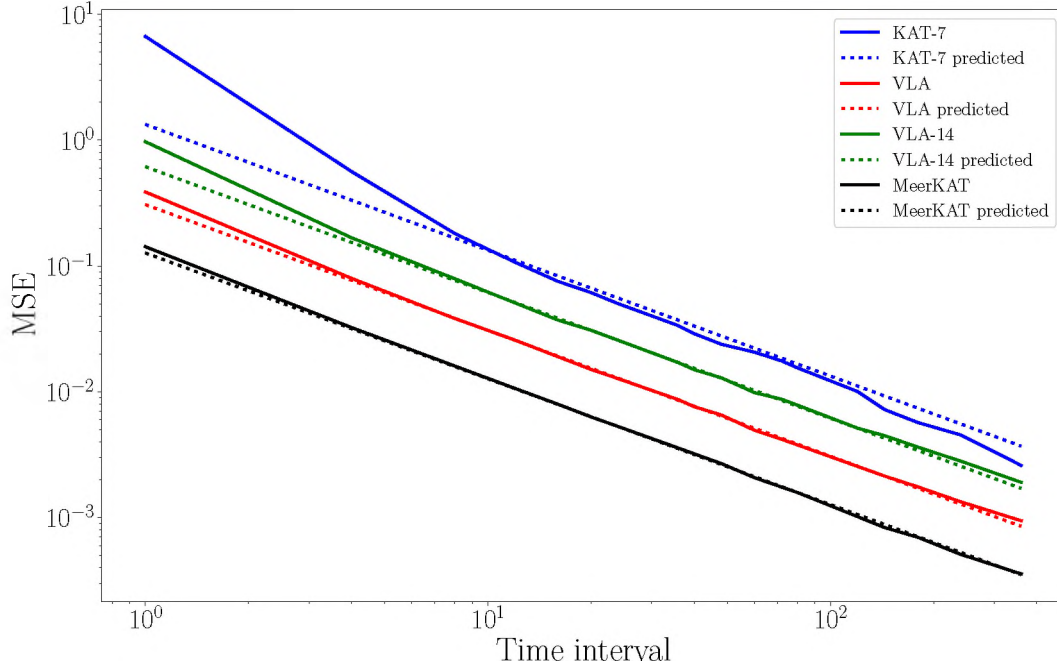


Figure 5.3: MSE of estimated gains against solution interval on a log scale. In this simulation, the input gains are all set to 1. The solid lines are the measured values while the dashed lines are the error values predicted using Eq. (5.17). Eq. (5.17) is a close fit for MeerKAT because of its large number of antennas. For the other arrays, the predicted and measured values only start agreeing after a certain time interval.

Effects of noisy gains on corrected visibilities

Similarly as in §4.2, we can decompose the corrected visibilities into several components,

$$\mathbf{V}_{pq}^c = \hat{\mathbf{G}}_p^{-1} \mathbf{V}_{pq} \hat{\mathbf{G}}_q^{-H} \quad (5.18)$$

$$= \hat{\mathbf{G}}_p^{-1} \mathbf{G}_p \mathbf{C}_{pq} \mathbf{G}_q^H \hat{\mathbf{G}}_q^{-H} + \hat{\mathbf{G}}_p^{-1} \boldsymbol{\epsilon}_{pq} \hat{\mathbf{G}}_q^{-H} \quad (5.19)$$

$$= \hat{\mathbf{G}}_p^{-1} (\mathbf{G}_p (\mathbf{C}_{pq}^m + \mathbf{C}_{pq}^{um}) \mathbf{G}_q^H + \boldsymbol{\epsilon}_{pq}) \hat{\mathbf{G}}_q^{-H} \quad (5.20)$$

$$= \hat{\mathbf{G}}_p^{-1} \mathbf{G}_p \mathbf{C}_{pq}^m \mathbf{G}_q^H \hat{\mathbf{G}}_q^{-H} + \hat{\mathbf{G}}_p^{-1} \mathbf{G}_p \mathbf{C}_{pq}^{um} \mathbf{G}_q^H \hat{\mathbf{G}}_q^{-H} + \hat{\mathbf{G}}_p^{-1} \boldsymbol{\epsilon}_{pq} \hat{\mathbf{G}}_q^{-H}, \quad (5.21)$$

after splitting the coherency, C_{pq} , into a modelled and an unmodelled component C_{pq}^m and C_{pq}^{um} , respectively. Eq (5.21) consists of three terms which we refer to as the *Modelled*⁴, the *Unmodelled*⁵ and the *Noise*⁶ terms respectively. If the calibration is perfect, $\hat{\mathbf{G}}_p = \mathbf{G}_p$ and $\hat{\mathbf{G}}_p^{-1}\mathbf{G}_p = \mathbf{I}$. Thus, the *Modelled* and *Unmodelled* terms correspond to the true sky visibilities, and the *Noise* term is the noise in the visibilities slightly modified by the inverse of the gains. However, at low SNR, the solver fits noise rather than the model, thereby transferring the model flux into the gains. When such gains are applied to the noise, the resultant *Noise* term contains ghost sources which lead to overestimation of sources' fluxes. This problem is not only related to solution intervals but forces us to re-assess our definition of corrected visibilities. How exactly should we apply the gains to the data without modifying the underlying structure of the noise in the data? We performed a two source simulation (**Simulation-iii**) to investigate the contribution of the different terms in Eq. (5.21) to the corrected visibilities at low SNRs.

Fig. 5.4 shows images of the various components of the corrected visibilities after calibration. Alongside these, we have plotted the following quantities called the *distilled modelled* and the *distilled unmodelled* terms.

$$\begin{aligned} \text{distilled modelled} &= \hat{\mathbf{G}}_p^{-1}\mathbf{G}_p\mathbf{C}_{pq}^m\mathbf{G}_q^H\hat{\mathbf{G}}_q^{-H} - \mathbf{C}_{pq}^m, \\ \text{distilled unmodelled} &= \hat{\mathbf{G}}_p^{-1}\mathbf{G}_p\mathbf{C}_{pq}^{um}\mathbf{G}_q^H\hat{\mathbf{G}}_q^{-H} - \mathbf{C}_{pq}^{um}. \end{aligned}$$

Each 2×3 block corresponds to a specific time interval. On the top row, from left to right, we have the images of the corrected visibilities, the *Modelled* and the *Unmodelled* terms, respectively. On the bottom row, from left to right, we have the *Noise*, *distilled modelled* and the *distilled unmodelled* terms, respectively. The blue and black circles indicate the position of the modelled and unmodelled source, respectively. The red circle indicates a position at which we expect a ghost source to form because of the unmodelled source.

When the time interval is 1 (see Fig. 5.4a), the SNR is very low. Hence we fit noise instead of the modelled visibilities. In this case, the flux of the model source is absorbed into the gains.

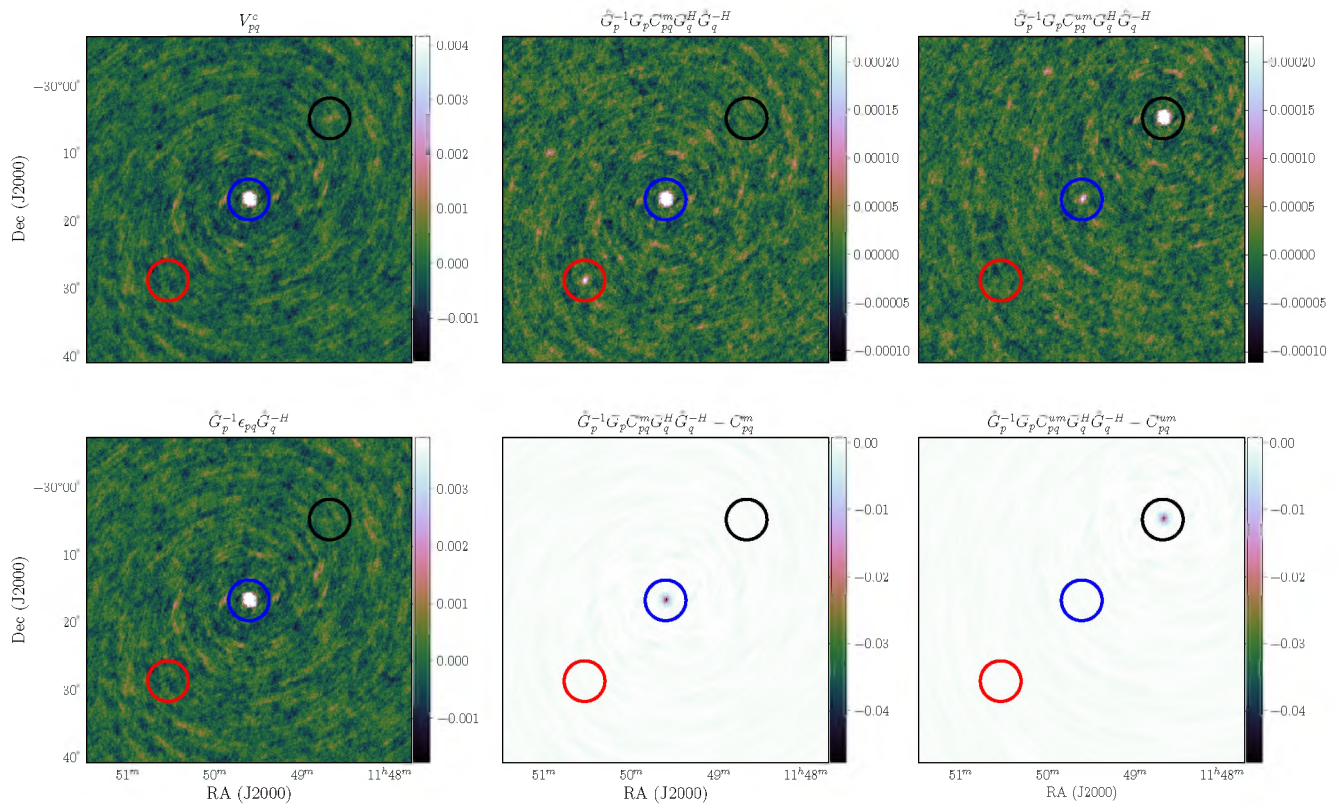
⁴*Modelled* = $\hat{\mathbf{G}}_p^{-1}\mathbf{G}_p\mathbf{C}_{pq}^m\mathbf{G}_q^H\hat{\mathbf{G}}_q^{-H}$

⁵*Unmodelled* = $\hat{\mathbf{G}}_p^{-1}\mathbf{G}_p\mathbf{C}_{pq}^{um}\mathbf{G}_q^H\hat{\mathbf{G}}_q^{-H}$

⁶*Noise* = $\hat{\mathbf{G}}_p^{-1}\epsilon_{pq}\hat{\mathbf{G}}_q^{-H}$

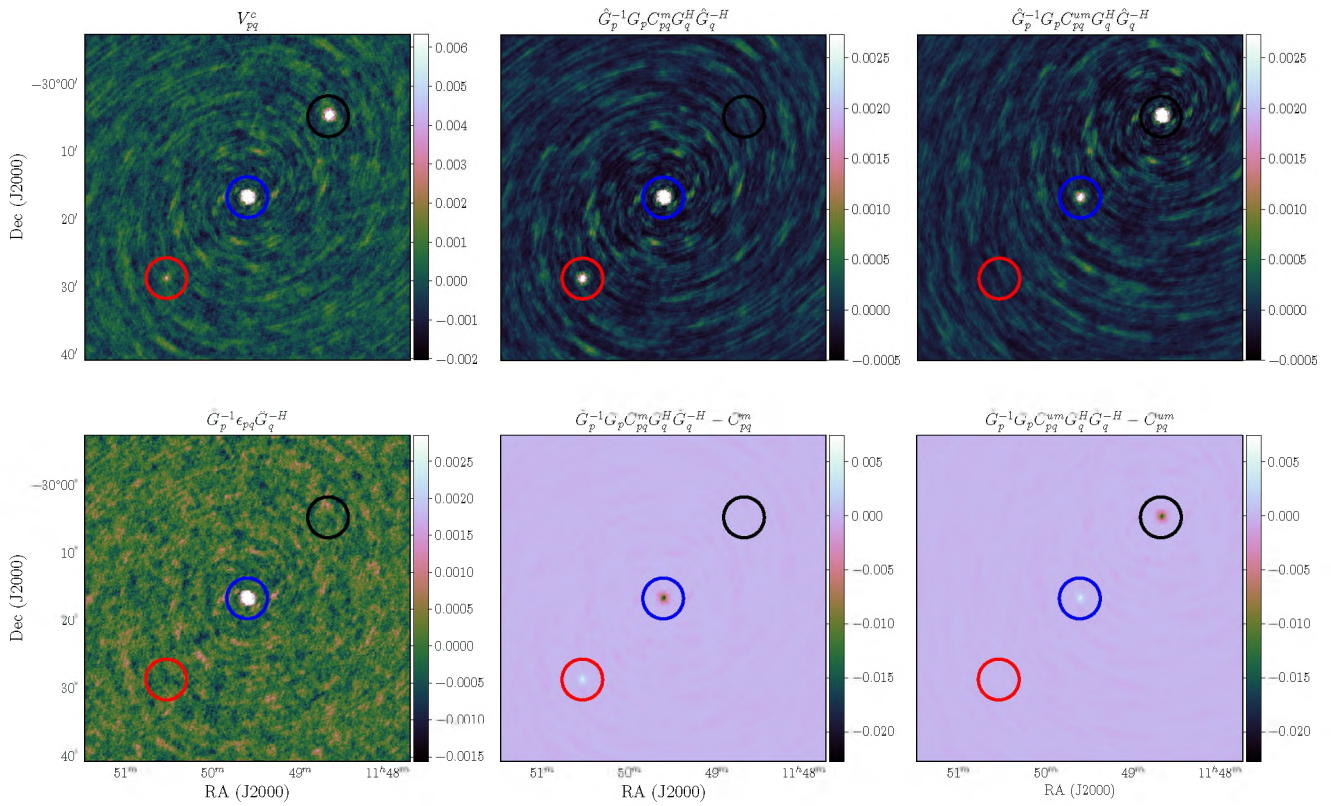
Applying these gains to the data suppresses the model source from the *Modelled* term and its signature instead appears in the *Noise* term. Because of the strong influence of the noise, in this case, the corrected visibilities look very similar to the *Noise* term. Similarly to the *Modelled* term, the unmodelled source appears to be largely suppressed in the *Unmodelled* term. In images of both the *Modelled* and the *Unmodelled* terms, we can see artefacts caused by the presence of the unmodelled source. As discussed in [Grobler et al. \(2014\)](#) and [Grobler et al. \(2016\)](#), unmodelled sources will lead to the formation of a string of ghost sources along the line between the model and the unmodelled source. Because of the short time interval used here, the noise dominates the calibration, and the ghost sources are too faint to see their signatures in the distilled maps.

When the time interval is 8 (see Fig. 5.4b), the SNR is improved. This causes the unmodelled source to have more influence on the calibration, hence the ghost sources are brighter, and we can even see a ghost source in the corrected visibilities. Likewise, the amount of flux transferred to the *Noise* term is significantly reduced, and some of the ghost sources are visible in the distilled maps. Using an interval of 720 (see Fig. 5.4c), both the effects of the noise and the unmodelled source are entirely averaged out. The corresponding images look exactly like what we expect from a perfect calibration. Such a long interval is only possible in such a case because we did not put any gain corruptions into the data to start with.



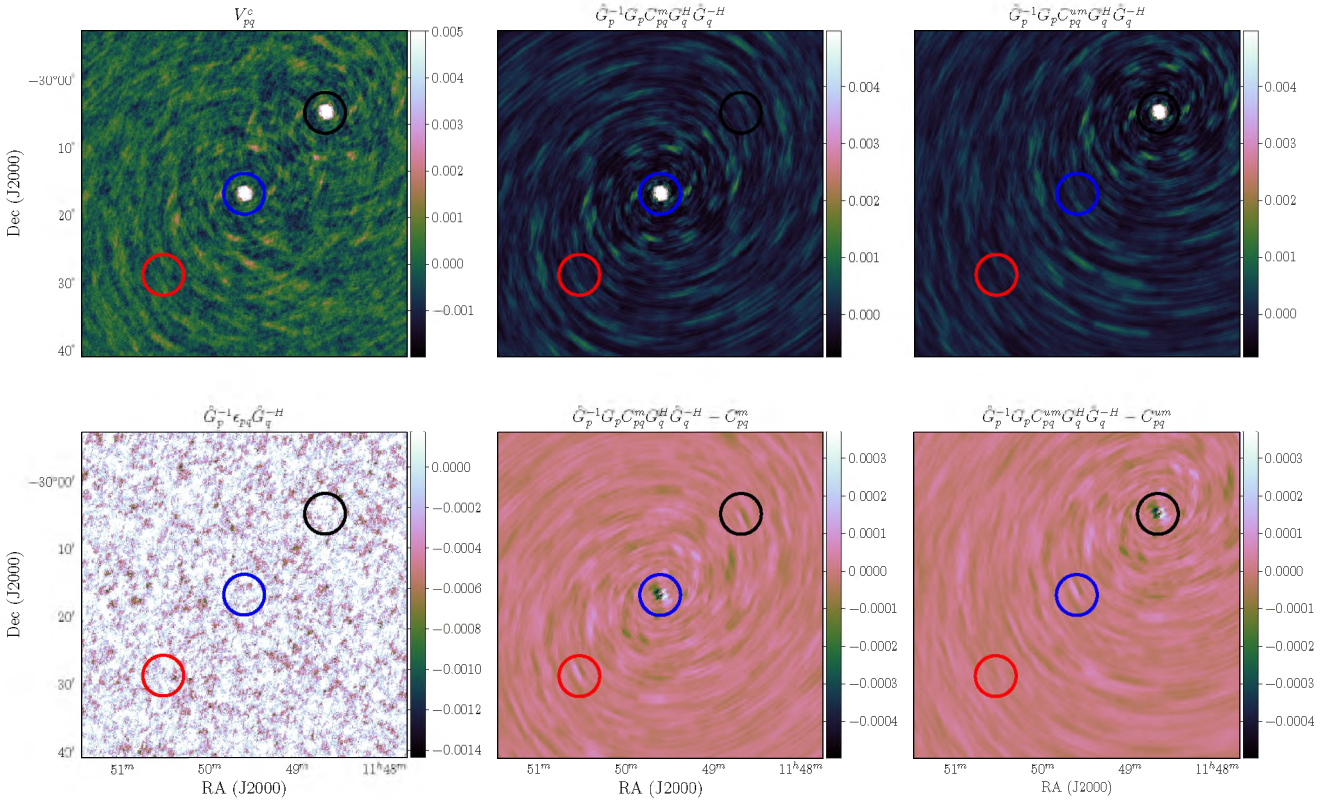
(a) Time interval = 1 (10 secs)

Because of the low SNR in this case, we fit noise instead of modelled visibilities. The modelled source is absorbed into the gains and its signature appears in the blue circle of the *Noise* term. In the *Modelled* and *Unmodelled* terms, we can see ghost sources appearing in the red and blue circles respectively. These ghost sources results from the unmodelled source and causes further suppression in the fluxes of the sources. Due to the large suppression caused by the noise in the *distilled modelled* and the *distilled unmodelled* terms, we can only see negative peaks in the blue and black circles at the position of the sources confirming that their fluxes have been hugely suppressed.



(b) Time interval = 8 (80 secs)

Here the SNR is higher, hence the unmodelled source has a stronger influence on the calibration. This results in brighter ghost sources which we can see in the corrected visibilities (red circle), the *Modelled* term (red circle) and the *Unmodelled* term (blue circle). Also, we can see positive peaks in the red and blue circles of the *distilled modelled* and the *distilled unmodelled* terms respectively confirming the stronger influence of the unmodelled source in this case.



(c) Time interval = 720 (2 hours)

In this case, the effects of the noise and the unmodelled source are ultimately average out because of the very long solution interval. Hence in the corrected visibilities, *Modelled* and *Unmodelled* terms, we see that the model and unmodelled source in the blue and black circles have not been suppressed. The *Noise* term is entirely noise like as we expect, while the *distilled modelled* and the *distilled unmodelled* term images are almost blank except for some remaining PSF structures.

Figure 5.4: Images of the different terms in Eq. (5.21) and the *distilled modelled* and the *distilled unmodelled* terms. Each 2×3 block corresponds to a specific time interval. On the top row, from left to right, we have the images of the corrected visibilities, the *Modelled* and the *Unmodelled* terms respectively. On the bottom row, from left to right, we have the *Noise*, *distilled modelled* and the *distilled unmodelled* terms respectively. The blue and black circles indicate the position of the modelled and unmodelled source, respectively. The red circle indicates a position at which we expect a ghost source to form because of the unmodelled source. (a): 1 (10 secs), (b): 8 (80 secs) and (c): 720 (2 hours) respectively.

5.2.2 Intrinsic variability of the gains

The expressions derived in §5.2.1 rely on the assumption that we have an unbiased estimator, the residuals visibilities or noise is normally distributed and the Hessian is diagonally dominant. This is almost never true because the gains are neither constant in time nor in frequency. This section focuses on the bias term of Eq. (5.10). This term is the error in the estimated gains due to the chosen solution interval. This term is complicated to understand and to model analytically because, in practice, we do not have any information about the variability of the gains. To acquire more insights on the bias term, we plot in Fig. 5.5 a similar MSE curve to Fig. 5.1b for non-noisy gains.

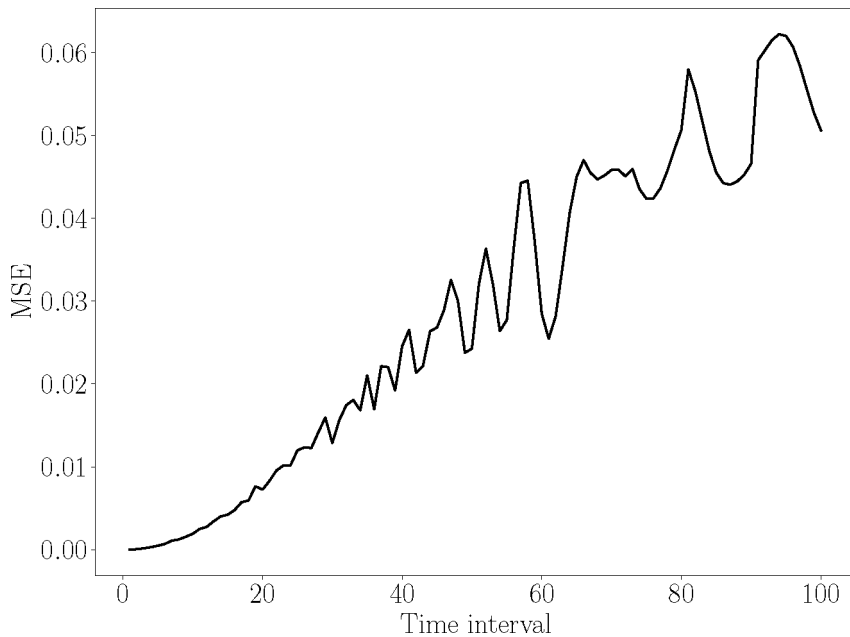


Figure 5.5: This plot shows the error in the gains caused by the averaging as a function of the solution interval. This error term, contrary to that from the noise, increases with increasing solution intervals.

As expected, the error increases with the solution interval. The bumpiness in the curve is as a result of the fact that not all intervals divide the data equally, i.e. all the solution interval blocks do not have the same number of visibilities. This effect increases with solution intervals

as we get more and more unequal divisions with longer intervals. This is an additional overhead which makes it difficult to formulate the solution interval problem as an optimisation problem. In practice, this might even be more complicated, since observations are usually in scans (switching between numerous fields) each having a different number of visibilities and different time stamps.

We use **Simulation-iv** to visualise the effects of the intrinsic variability of the gains on corrected visibilities. We simulate a 1 Jy phase centred source using MeerKAT and corrupt it with rapidly varying gains. We add no noise to the data to ensure a high SNR and avoid hiding the effects of the gain variations in the noise. The results are plotted in Fig. 5.6.

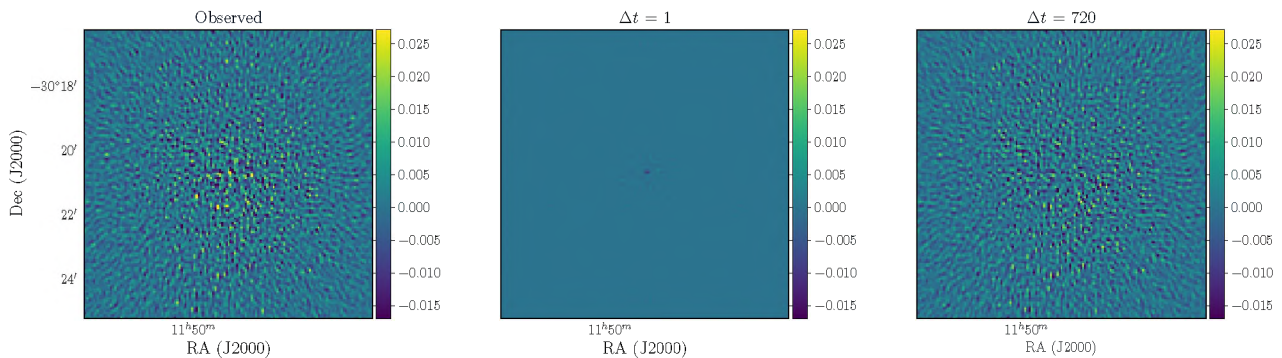


Figure 5.6: Left is an image of the artefacts introduced by the gains (corrupted visibilities minus model visibilities). The middle and right images are the artefacts maps after calibration (corrected visibilities minus model visibilities) with time intervals of 1 and 720 respectively.

The leftmost image shows the artefacts added to the data by the gains. This image is computed by subtracting the model visibilities from the corrupted visibilities. For the middle and the rightmost images, we subtract the model visibilities from the corrected visibilities. These images show how well we have removed the artefacts introduced by the gains. With a time interval of 1, almost all the artefacts introduced by the gains are removed because we have the highest time resolution possible to track all gain variations in time. On the other hand, if a time interval of 720 is used, we have low resolution, and cannot track the gain variations. As seen earlier in Fig. 5.2, this leads to artefacts that increase the noise in the image, particularly around sources. The artefact's noise is not necessarily Gaussian or follows a symmetric i.i.d distribution, and thus will not go down with averaging. Hence, even if we perform multi-frequency observations over a long period, if

the solver does not accurately capture the variations in the gains, the noise in the averaged image will be higher than expected.

5.2.3 Model Completeness & Radio Frequency Interference

In §5.2.1, we saw that the effects of unmodelled sources depend on their brightness relative to the noise in the visibilities. Hence an unmodelled source with flux \ll noise rms will have minimal impact on calibration especially in the case of DI calibration. For DD calibration, given the low number of constraints per free parameter, the faint sources may have a stronger impact. On the other hand, a source relatively bright compared to the noise and the model flux will have a substantial contribution. This is precisely what we observe in §5.2.1 for **Simulation-iii**. Furthermore, we demonstrated in §4.1.1 that the main effect of numerous unmodelled point sources is to make the data seem noisier, i.e. it increases the variance of the residual visibilities, and that their effects could be mitigated using the robust solver.

Another parameter which complicates the task of selecting an optimal solution interval is the presence of RFI in RI data. We saw in §4.2 that if the RFI is local to few time chunks and frequency channels, then using long solution intervals can significantly reduce the effects of the RFI, with the preferred solution being to employ the robust solver which downweights contaminated visibilities.

5.3 Searching for optimal solution intervals

After discussing the concept of solution intervals, and most of its effects in different calibration scenarios, we now present a brute force optimal solution interval search algorithm. Before describing the algorithm in §5.3.2, we first discuss the idea of a *calibration minor cycle* in §5.3.1.

5.3.1 Calibration minor cycle

The process of finding an optimal solution interval for calibration requires repeating calibration numerous times and identifying the interval which produces the best residuals. Given the large

size of RI datasets, this is not a practical approach, and any inference method which requires repeating the calibration and computing residual visibilities will be extremely inefficient. We suggest a framework termed the *calibration minor cycle* in analogy to the major/minor cycles used by deconvolution algorithms such as Schwab (1984). The minor cycle here refers to projecting the problem from the high dimensional visibility space to the low dimensional gain space for computational efficiency. If we solve for the gains using the full time and frequency resolution, then, because of the high number of degrees of freedom, the gain solutions will be very noisy. However, once we have a maximum likelihood estimate of \mathbf{g} , $\hat{\mathbf{g}}$, the equivalence

$$\mathcal{P}(\mathbf{g}|\mathbf{V}) = \frac{\mathcal{P}(\mathbf{V}|\mathbf{g}, \Sigma)\mathcal{P}(\mathbf{g})}{\mathcal{P}(\mathbf{V})} \approx \mathcal{P}(\mathbf{g}|\hat{\mathbf{g}}) = \frac{\mathcal{P}(\hat{\mathbf{g}}|\mathbf{g}, \Sigma_g)\mathcal{P}(\mathbf{g})}{\mathcal{P}(\hat{\mathbf{g}})} \quad (5.22)$$

is approximately true. The reason for this is that $\hat{\mathbf{g}}$ is a stationary point of Eq. (5.3) and Σ_g is the Cramer-Rao bound and plays a similar role as the point spread function during imaging. Hence, following Eq. (5.22), given an initial noisy estimate of the gains, the search for an optimal interval can be done directly in gain space without having the computational cost of manipulating visibilities. Using the solution interval parametrisation below

$$\mathbf{g}_p(\boldsymbol{\theta}_p) = X\boldsymbol{\theta}_p, \quad (5.23)$$

where X^7 is a suitable design matrix, we seek to find

$$\min_{\boldsymbol{\theta}} \chi^2, \quad \text{where} \quad \chi^2 = (\hat{\mathbf{g}} - X\boldsymbol{\theta})^H \Sigma_g^{-1} (\hat{\mathbf{g}} - X\boldsymbol{\theta}), \quad (5.24)$$

i.e. we assume that the gain solutions are normally distributed around their true value (that we wish to approximate with a boxcar model) with noise drawn from $\text{CN}(0, \Sigma_g)$. This formulation is per antenna and correlation allowing one to parallelise over these dimensions. This is important because we have to solve the problem many times in order to find the optimal solution interval.

5.3.2 Optimal solution interval search algorithm

Following up from §5.3.1, our goal is to find solution intervals for which the gains parametrised by Eq. (5.23) minimises the χ^2 in Eq. 5.24. Because a Bayesian evidence approximation will

⁷See the Appendix C for the explicit form of X .

computationally be very expensive, we suggest the following simplistic procedure to find adequate solution intervals for calibration.

The first step in the process is identifying a minimum solution interval for which the maximum likelihood problem gives a per-antenna SNR of approximately 3⁸. Thus, for a field with peak flux, P , observed with an interferometer consisting of N_a antennas, having visibilities with noise σ_{rms} per baseline, we want, from Eq. (5.17), an interval satisfying

$$\sigma_a \leq \frac{P}{3} \quad \text{where} \quad \sigma_a = \frac{\sigma_{\text{rms}}}{\sqrt{N_a - 1}}. \quad (5.25)$$

For a given interval containing $n = n_t n_\nu$ combined time and frequency grid points, we have,

$$\sigma_a = \frac{\sigma_{\text{rms}}}{\sqrt{n(N_a - 1)}} \quad \Rightarrow \quad n \geq \frac{9\sigma_{\text{rms}}^2}{P^2(N_a - 1)}. \quad (5.26)$$

Once the minimum solution interval has been obtained, the next step is to perform calibration with this minimum interval to obtain a maximum likelihood solution, $\hat{\boldsymbol{\theta}}$, and an updated noise covariance, $\hat{\Sigma}_g$. By using a relatively short interval, the maximum likelihood solution is unlikely to be significantly biased and can be used to map the problem into gain space.

To avoid having to specify informative priors, we attempt to identify an easily computable statistical inference criterion for our model selection problem of finding the optimal interval given an initial maximum likelihood estimate. We choose the Akaike Information Criterion (AIC; see Akaike (1974, 1998)) defined as

$$\text{AIC} = 2N_p - \ln \left(\sum_i \frac{\mathbf{r}_i^2}{\sigma_i^2} \right) + \frac{2N_p^2 + 2N_p}{N_g - N_p - 1}, \quad (5.27)$$

where \mathbf{r} is the difference between the estimated gains and reconstructed boxcar gain vector separated into its real and imaginary parts, N_g and N_p are the number of real terms in the gains at full resolution and the boxcar parameter vector respectively, and σ_i is the uncertainty in the gain with index i . The AIC is a model selection criterion that minimises the information loss by connecting

⁸This value being arbitrary and entirely up to the user to specify. The most common value used by calibration packages such as CASA (McMullin et al., 2007) to check valid solutions is 3 (see Brogan et al. (2018) for more discussions on how to chose to this threshold).

the Kullback-Leibler measure (see [Kullback & Leibler \(1951\)](#)) and the maximum likelihood estimation method. The AIC is a robust statistic for this specific model selection problem because it corrects for the low number of degrees of freedom and sample sizes. Precisely, the last term of Eq. (5.27) is a penalty term which corrects for the low number of degrees of freedom. The best model when using the AIC is the one with the minimum AIC. We summarise the proposed search algorithm as follows:

1. Read the observation parameters from a measurement set, i.e. integration time, channel width, number of channels, number of antennas, flags, and scans information.
2. Estimate the noise in the visibilities from the Data and Model or using the System Equivalent Flux Density.
3. Estimate the number of points, n , necessary to reach a certain minimum SNR (for example, 3 or 5).
4. Define the minimum solution interval by first choosing the largest frequency interval, which matches n (i.e. set $n_\nu = n$ and $n_t = 1$). Ideally, anyone performing self-calibration should have already done bandpass calibration. Hence, the variations in frequency are expected to be slow compared to time. We can also do the contrary (i.e. set $n_\nu = 1$ and $n_t = n$) if we expect the specific gain to vary faster in frequency compared to time. A conservative approach will be to use \sqrt{n} for both time and frequency (i.e. set $n_\nu = \sqrt{n}$ and $n_t = \sqrt{n}$). In the case $n >$ the total number of channels possible, $\max(n_\nu)$, then we set $n_\nu = \max(n_\nu)$ and $n_t = \left\lceil \frac{n}{\max(n_\nu)} \right\rceil$, and likewise we will use $n_\nu = \left\lceil \frac{n}{\max(n_t)} \right\rceil$ and $n_t = \max(n_t)$ when we expect more rapid variations in frequency compared to time.
5. Perform calibration with the selected minimum interval and using the estimated gains, search for the optimal interval using the AIC.

5.4 Application on simulated data

In this section, we test the proposed method on simulated datasets. We only perform simulations for DI calibration. For DD calibration, solution intervals play a slightly different role, i.e. long solution intervals are generally employed not only to improve the SNR but more importantly to make the calibration well posed since the number of unknown parameters increases rapidly with the number of directions. §5.4.1 focuses on single frequency simulations, and its main objective is to examine how the AIC performs in just finding optimal time intervals using gains at their full time resolution. §5.4.2 presents a more realistic multi-channel observation based on the “VIDEO” field dataset.

5.4.1 Time only simulations

For all simulations, we used the MeerKAT array with a bandwidth of 1 MHz and a start frequency of 0.9 GHz. All the simulations have 2 hours of synthesis time with 10 seconds integration time (720 timeslots). The parameters for these simulations correspond to **Simulation-v** in Table 5.1. We varied the gain and noise parameters across the simulations in order to vary from low to high SNR regimes as well as from slowly to rapidly varying gains. The corrupted data is then repeatedly calibrated with a range of solution intervals and the optimal solution intervals from the actual gains compared with that obtained using the boxcar constructed gains’ AIC.

Because of the entanglement between the gain variability and the SNR of the data, it is difficult to cleanly separate the presentation here into low vs high SNR and slowly vs rapidly varying gains, but we will mention the specific regimes when describing the results. Table 5.2 shows the parameters of the squared exponential covariance function used for the gain realisations and the rms of the noise added to the visibilities (σ_{rms}). The phases of the input gains for a few antennas are shown in Fig. 5.7. Here we made the amplitudes and phases to have similar variations but in practice the phases vary on shorter time scales compared to the amplitudes.

Fig. 5.8 shows plots of the MSE of the estimated gains and the AIC of the boxcar reconstructed gains at different time intervals based on the solution at time interval 1. These plots depict the following:

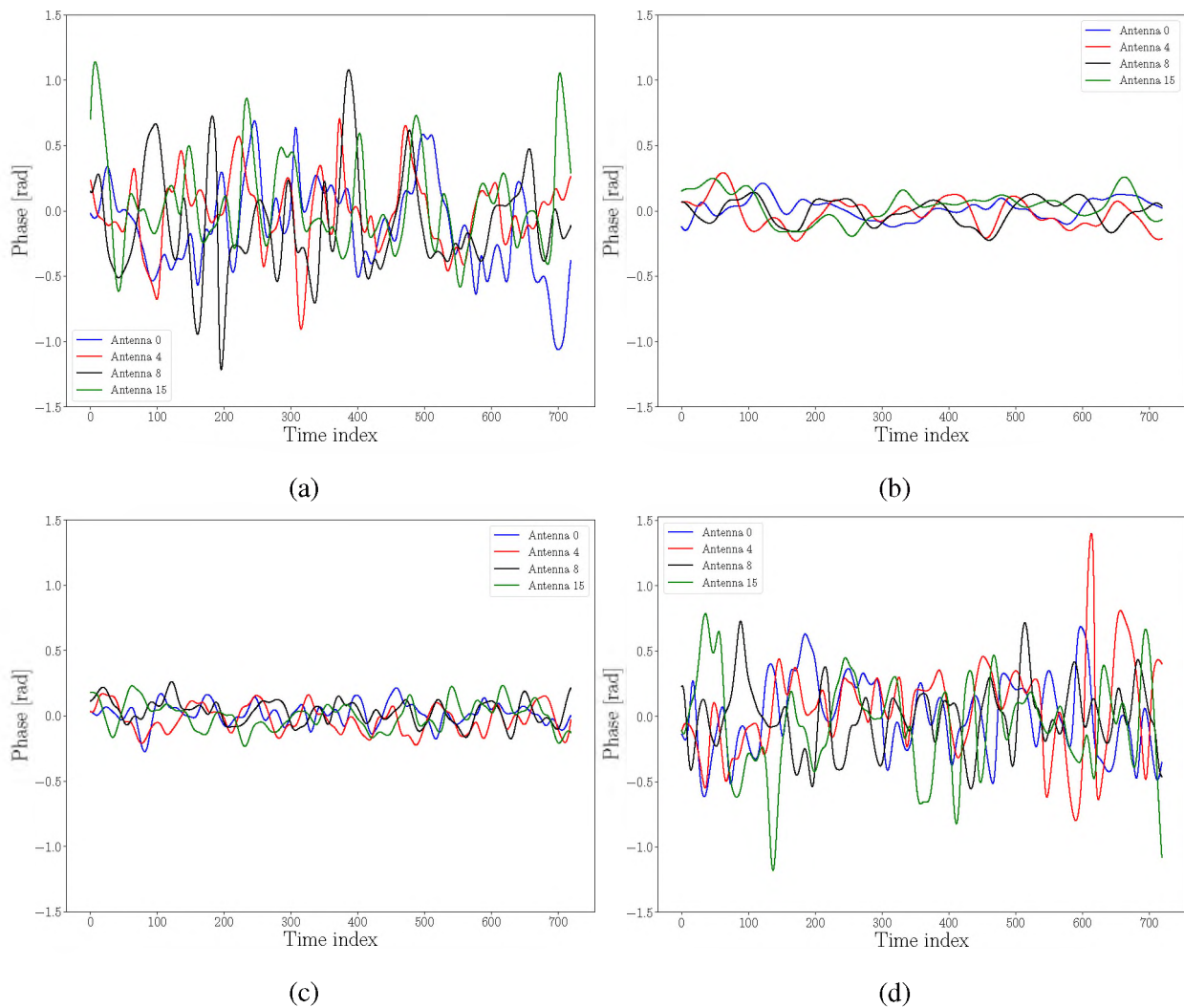


Figure 5.7: Plots of the different gain realisations for a few selected antennas (phase against time index). The colours blue, red, black and green are for antenna 0, 4, 8 and 15 respectively. The GP parameters used to simulated these gains are shown in Table 5.2. All the figures are plotted on the scale to illustrate the differences in variability across the different experiments clearly. (a) and (d) corresponds to rapidly varying gains with high variation. (b) corresponds to slowly varying gains with low variation while in (c) we show relatively rapidly varying gains with low variation.

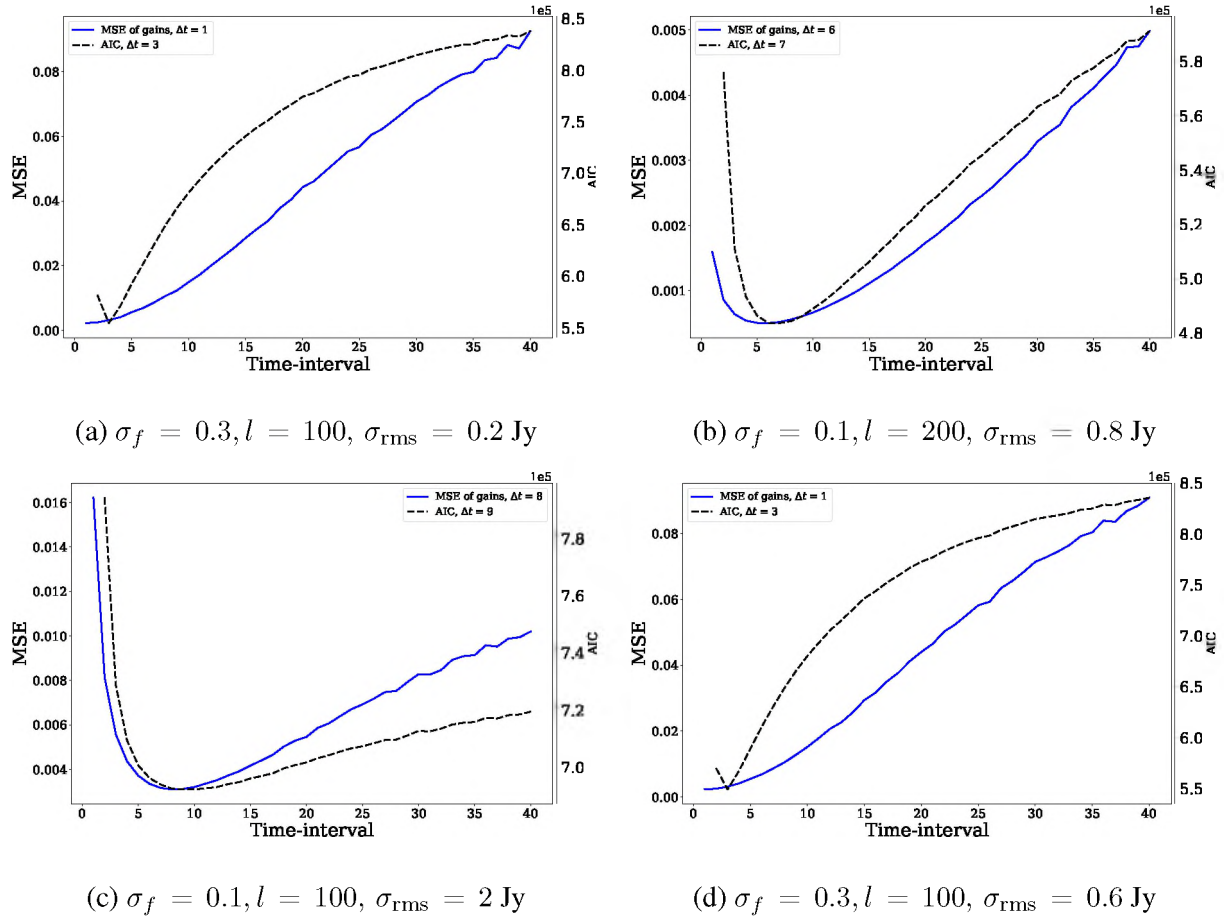


Figure 5.8: Plots of the MSE of the estimated gains at different time intervals (blue) and the AIC of the reconstructed boxcar gains at different intervals based on the gains (black dashed lines). (a) is a high SNR regime with rapidly varying gains. Thus a short time interval is required for calibration, which is well predicted by the AIC. (b) is a case with medium SNR and slow varying gains and thus, a longer time interval can be used for calibration as suggested. (c) is a scenario with a low SNR and rapidly varying gains. The AIC agrees with the minimal MSE for a slightly longer interval. (d) is a simulation with highly varying gains on a short time scale and a medium SNR. The prediction is again very close to the minimal MSE of the gains. Note the shape of AIC curves for (a) and (d). This is as a result of the rapid variability of the gains in these simulations.

1. Fig. 5.8a is a high SNR simulation with rapidly varying gains. Here, the optimal interval is short, and this is well predicted by the AIC.
2. Fig. 5.8b is a medium SNR regime with rapidly but not highly varying gains. A longer time interval can be used, and this is correctly predicted.
3. Fig. 5.8c is a case with a low SNR and slowly varying gains. Because of the low SNR, we need a long time interval in this scenario. The stability of the gains also favours this. The AIC accurately predicts the optimal interval.
4. Fig. 5.8d is a case of extremely high variability (these are shown in Fig. 5.7d). Here, $\sigma_f = 0.3$ suggests high variability in the gains and $l = 100$ (10 units since the integration time in our data is 10 seconds) implies variability at a relatively short time scale. The AIC prediction is close to the minimal MSE of the gains.

Fig. 5.8 confirms that it is possible to define how long we can make our solution intervals using the AIC computed from the gain solutions [at the shortest possible intervals] and their boxcars approximations.

5.4.2 Time and frequency

After demonstrating that it is possible to search for optimal interval using the AIC on single frequency simulations, we now proceed to multi-frequency observations, since most continuum radio observations are made over a large bandwidth consisting of multiple channels. We attempt in this section to replicate the VLA observation of the “VIDEO” field we used in §4.2 and, in the next section, we apply our brute force approach to find the optimal solution interval for the calibration of the real “VIDEO” dataset.

We recall that this observation covers a frequency range of 0.9–2.6 GHz split into 16 spectral windows with 64 channels each. The integration time was ≈ 9 seconds on average. 28 antennas were used for the observation with a maximum baseline of 36.4 km. The target field was observed in 9 different scans with three groups of consecutive scans which can be calibrated as

single chunks. After initial flagging and calibration using CASA (McMullin et al., 2007) (see Heywood et al. (submitted) for more details), we imaged the 1GC-corrected data and extracted a component-based sky model using PyBDSF (Mohan & Rafferty, 2015). The sky model consists of over 200 Gaussian sources with the brightest source of ≈ 0.02 Jy using an island and pixel threshold of 5 and 10σ for the source extraction using PyBDSF. Note that this is an apparent sky model since no primary beam correction has been done.

We then used the MeqTrees software package (Noordam & Smirnov, 2010) to replicate the observation and simulate visibilities corresponding to the “VIDEO” apparent sky model. We corrupt the simulated visibilities with gains drawn from a GP with a Matérn covariance function⁹. The gains are rapidly varying in time compared to frequency. Furthermore, the gains are constructed to have rapidly varying phases and slowly varying amplitudes. We add Gaussian noise with an rms value of 0.16 Jy to the corrupted visibilities. The added rms corresponds to the estimated rms from the real dataset. The real data contains RFI but we do not include RFI here since this was already treated in Chapter 4 using this dataset.

Following the steps described in §5.3.2, we first compute the solution interval that provides a minimum SNR of at least 3. For this computation, we use the simulated noise rms of 0.16 Jy and estimate the Peak flux as the mean of the absolute value of the model visibilities. The latter is used to estimate the peak flux since the sky model consists of numerous sources. The estimated peak flux is 0.029 Jy. Hence, using Eq. (5.26), we get the minimum combined frequency and time interval, n , to be 64 using a per-antenna SNR threshold of 3. Because the gains are slowly varying in frequency compared to time, we set the minimum frequency and time intervals to 64 channels (64 MHz) and 1 (9 secs) respectively. We perform calibration using this interval, and from the estimated gains, we only search for the optimal time interval. Fig. 5.9a shows a plot

⁹The following parameters are used

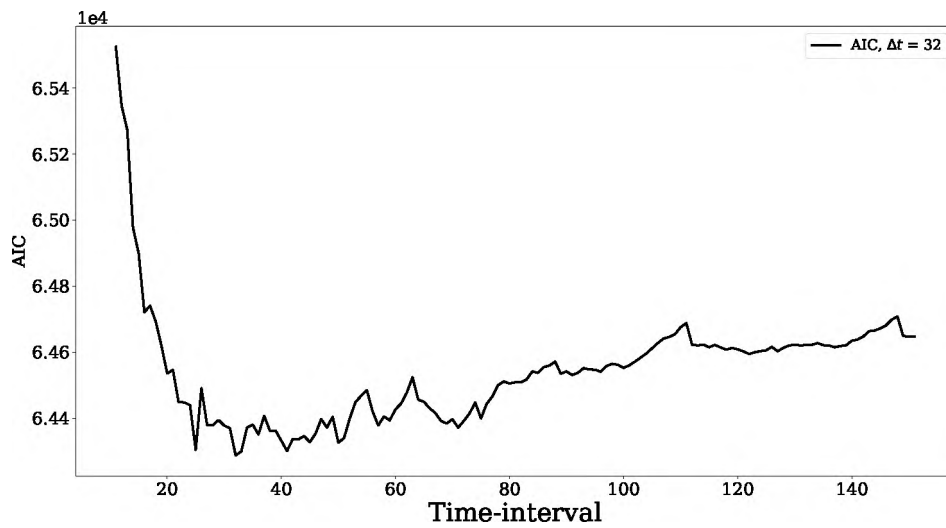
- Amplitude: $(\sigma_f, v, l) = (0.5, 3/2, 0.2)$ for time and $(\sigma_f, v, l) = (0.1, 7/2, 1)$ for frequency.
- Phase: $(\sigma_f, v, l) = (0.5, 3/2, 0.08)$ for time and $(\sigma_f, v, l) = (0.1, 7/2, 1)$ for frequency.

Here the units of the length, l , have been normalised to the range $[0, 1]$. See Appendix D for the definition of the Matérn covariance function.

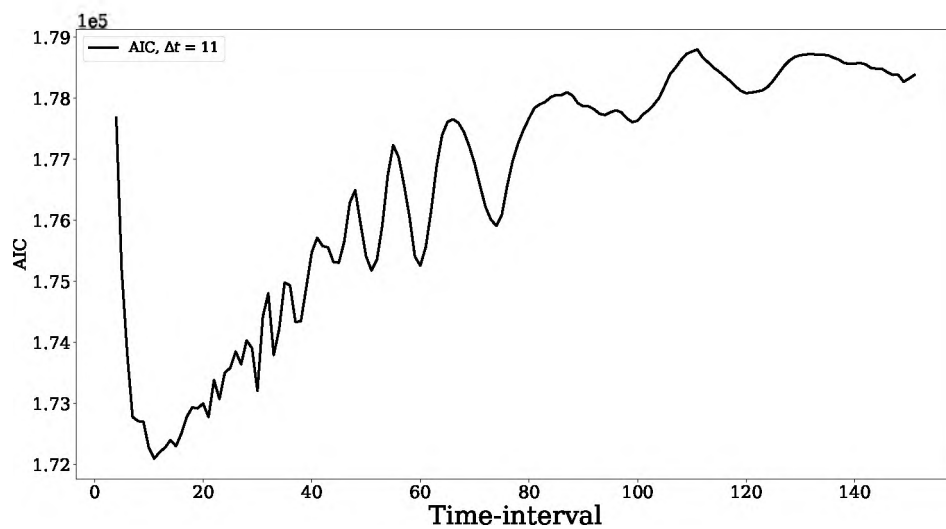
of the AIC obtained using the minimum possible frequency and time interval. From this plot, the suggested optimal time interval is (32, i.e. 4.8 mins). Next we calibrate using the suggested optimal interval. We also calibrate using the longest possible time interval (151, i.e. 23 mins). In both case we keep the frequency interval to the 64 MHz initially computed from the per-antenna SNR. After calibration, we show residual images similar to Fig. 5.6 for a patch at the field centre around a source in the field in Fig. 5.10. Fig. 5.10a shows the artefacts introduced by the gains (i.e. the image of the difference between the corrupted and the model visibilities). Fig. 5.10b is the residual image with the lowest solution time interval of 1; this image appears noisier compared to all the other images because the SNR is not high enough. At the suggested optimal time interval (see Fig. 5.10c), we see that the artefacts have been considerably attenuated and the noise in the image appears much lower. Fig. 5.10d shows the output at the highest time interval, in this image we still have most of the background structures from the artefacts because we cannot track the short time variations of the gains using such a long time interval. These images confirm that solution intervals of 64 MHz in frequency and 4.8 mins in time are the most adequate in this context for this simulated dataset. It is important to note that such a brute force technique only suggests appropriate solution intervals for the given dataset. Depending on the variability of gains and noise in the data, different combinations of time and frequency intervals may be possible for a given dataset. For example, if we have constant gains, high-SNR and no unmodelled sources using long or short solution interval will not produce significantly different results.

5.5 Application to Real Data

Following the successful application of the proposed method to synthetic data, we test its performance on the real “VIDEO” dataset. This section demonstrates some prominent practical limitations in defining optimal solution intervals. Firstly, examining the dataset we see that, it contains flagged visibilities because of the RFI present in the data. Fig. 5.11 shows a plot of estimated rms per scan chunks (i.e. the three groups of consecutive scans) at all frequencies. Fig. 5.11a shows the rms of time chunks at individual frequencies, while Fig. 5.11b is the rms



(a) simulated datasets



(b) real observational dataset

Figure 5.9: Plots of the AIC against time interval for the simulated and real VLA datasets. The optimal solution interval suggested by the per-antenna SNR and the AIC for the simulated dataset is 32 (4.8 mins) and 64 MHz for time and frequency respectively, while that for the real dataset is 11 (1.65 mins) and 128 MHz for time and frequency respectively.

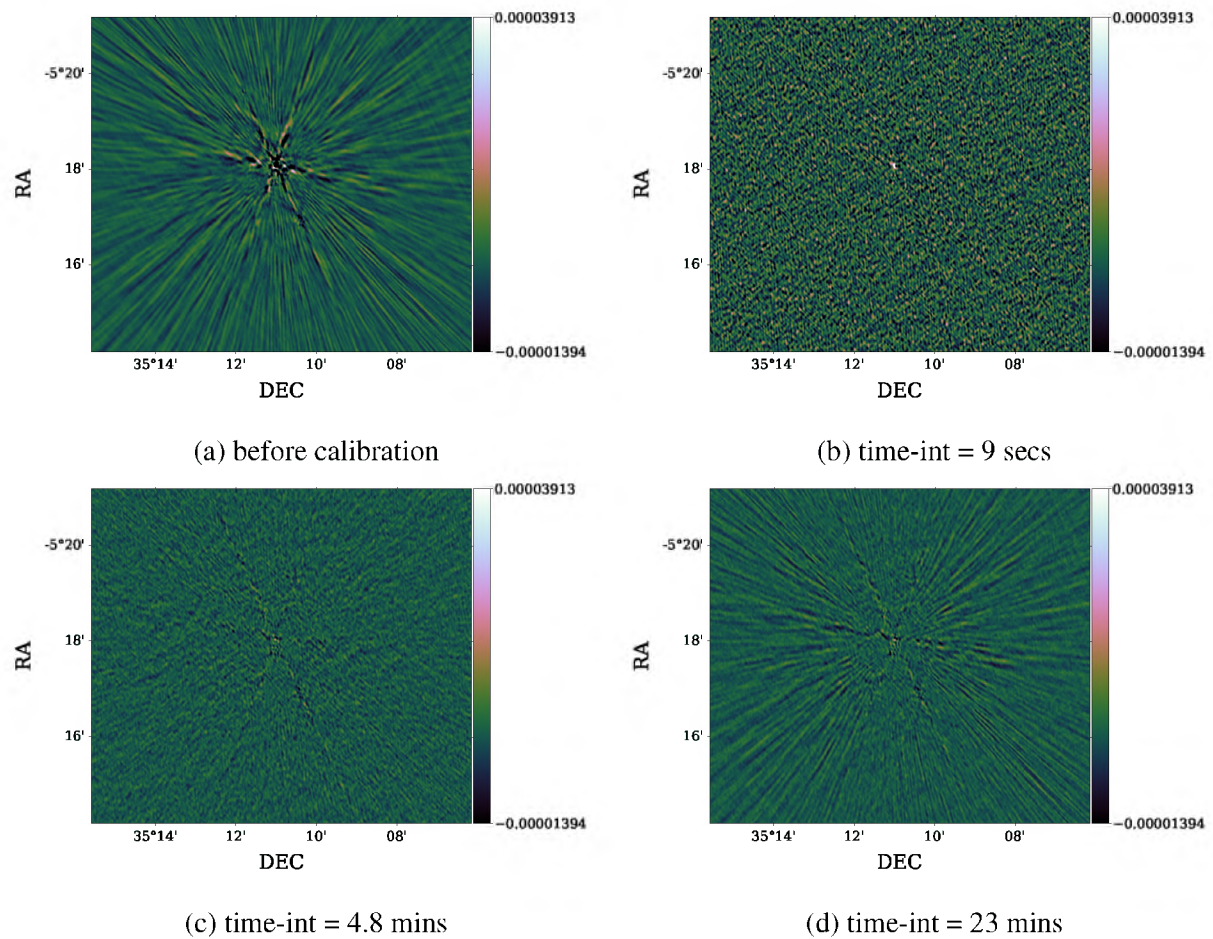
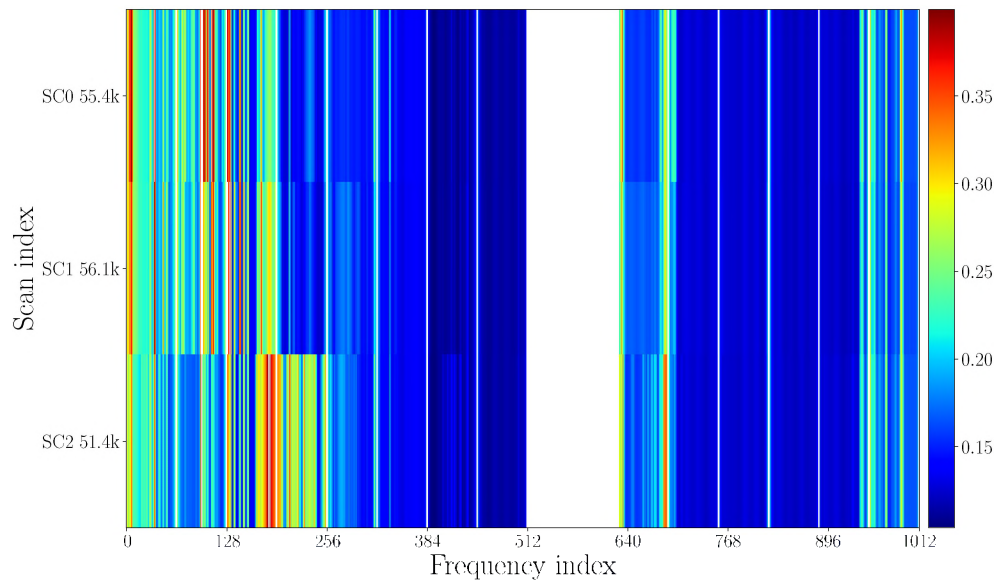


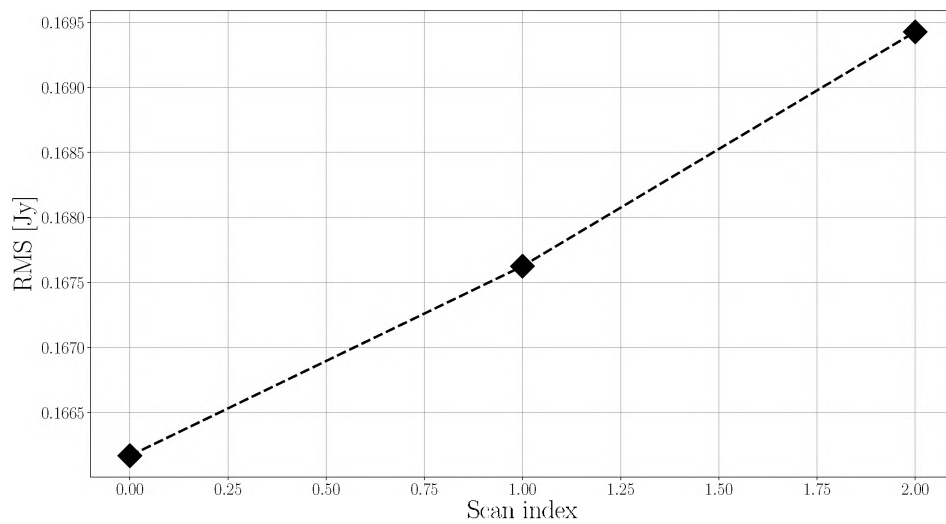
Figure 5.10: Residuals images for patch at the field-centre of the simulated “VIDEO” field. (a) corrupted visibilities minus model visibilities, showing the artefacts introduced by the gains. (b), (c) and (d) are images of corrected visibilities minus model visibilities, at different time solution intervals with a frequency solution interval of 64 MHz.

with all the frequencies in the time chunk combined. Using the rms per scan chunk, we estimated the minimum required interval for each chunk. We used an SNR of 5 because of the flagged data (this reduces the SNR because the valid number of visibilities at each time is now less than what it should be). Similarly as in §5.4.2, because the data is already bandpass calibrated, we set the frequency interval equal to the combined time and frequency interval, n , and only search for optimal time interval. For the three scan chunks, we found the minimum frequency intervals of 112 MHz, 118 MHz and 128 MHz respectively. Hence, we selected the highest frequency interval (128 MHz). Using this interval, we performed amplitude and phase calibration on the data. We then searched for the optimal interval using the estimated gains.

We show the AIC obtained from the estimated gains with the minimum possible interval in Fig. 5.9b. The suggested optimal time interval is 11 (1.65 mins). We then calibrate the data using the suggested time interval, as well as the longest possible time interval. Fig. 5.12 shows images of the same patch at the field-centre around a bright source as in §5.4.2 for the 1GC data, and the corrected data after calibration. Every calibration in this section is performed using the robust solver because of the unflagged RFI still present in the data. The images show similar results as in the simulations. At the lowest time interval (9 secs), even though the artefacts have been removed, the noise is slightly increased. Also, the effects of the RFI are also slightly visible in the map (see Fig. 5.12b). At the optimal interval (1.6 mins), the artefacts are removed, and the noise is lowered (see Fig. 5.12c). On the other hand, for the longest time interval (23 mins), despite some of the artefacts being removed, we can still see some background structures around the source (see Fig. 5.12d).



(a) rms per frequency



(b) rms using all frequencies

Figure 5.11: (a) is the estimated rms for the 3 different scan groups, per channel. The white blocks are flagged visibilities. The numbers on the y-axis next to the scan indices are the total number of visibilities in each scan (k denotes 1000). (b) is the estimated rms for each scan, using visibilities of all frequencies. The average estimated rms is ≈ 0.16 Jy.

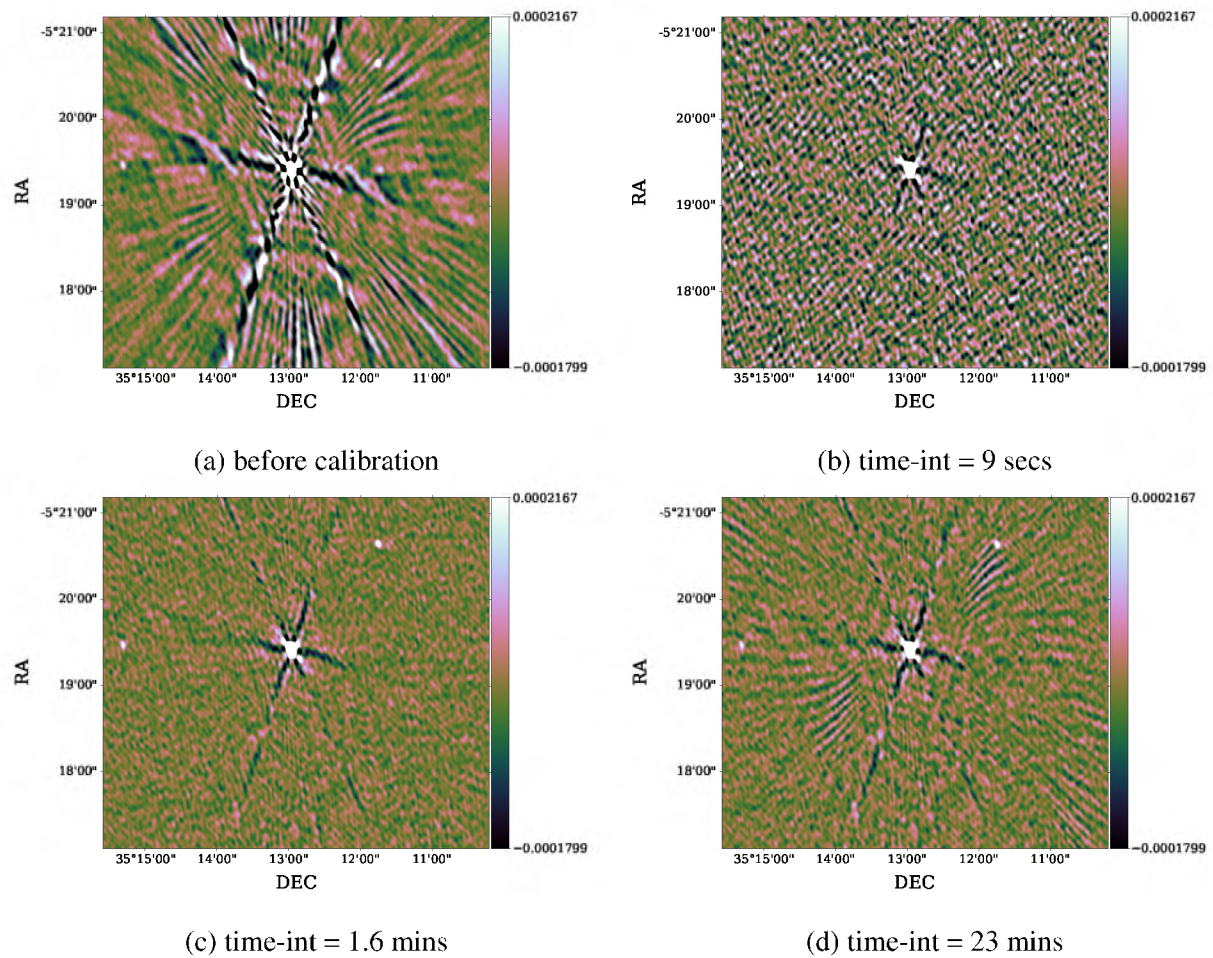


Figure 5.12: Images for patch at the field-centre around a corrupted source in the “VIDEO” field. (a) is an image of the corrupted visibilities before calibration. (b), (c) and (d) are images of the corrected visibilities after calibration with a frequency interval of 128 MHz and a short time interval of 9 secs, the optimal time interval of 1.6 mins and the longest time interval of 23 mins respectively.

5.6 Discussion

We presented an analysis of the various impacts of the choice of solution intervals on calibration and science goals. Furthermore, we proposed a practical statistical approach for choosing adequate solution intervals during calibration, and demonstrated this on simulated and real data.

The analysis presented here focused on DI calibration only, but Eq (5.26) can also be used for DD calibration. For DD calibration, the model is per direction, hence a conservative approach will be to utilise the flux of the faintest source in the model as the peak flux, for example.

In the real data application, when searching for the optimal solution interval, we used the robust solver to mitigate against the RFI still present in the data. This makes the selection of solution intervals during calibration complicated. Generally, RFI is localised to few time chunks and frequency channels, hence can we perhaps use different intervals for different chunks of the same dataset? However, we can only utilize this approach if we know in advance which data chunks are corrupted. Another factor we did not consider here is the spectrum of the sources. Ideally, we should take this into account when choosing solution intervals during calibration. We can also use multiple solution intervals in this case, but again this requires us to know the spectrum of the sources beforehand.

In summary, solution intervals can significantly improve calibration results, but it is not trivial to select the optimal solution interval during calibration. Ultimately, we have to encourage the use of fully parametric and regularised maximum likelihood calibration algorithms.

General Conclusions

In this thesis, we explored robust calibration techniques that mitigate against the effects of unmodelled sources and RFI during calibration. We presented the implementation of a robust calibration algorithm (Chapter 3) which uses the framework of complex optimisation as discussed by Tasse (2014a) and Smirnov & Tasse (2015) and a Student's *t* likelihood function. The implemented robust solver is now a subroutine in the latest release of the radio interferometric calibration suite, CubiCal (Kenyon et al., 2018). We demonstrated that the robust solver effectively reduces the amount of flux suppressed from unmodelled sources during calibration (Chapter 4). Statistical analysis of the visibilities showed that the main effect of unmodelled sources is to increase the perceived variance of the residual visibilities. This increased variance reduces the effective SNR of the data. The robust solver, which employs an interactive weighting scheme, significantly reduces flux suppression by down weighting baselines with high deviations from the estimated data covariance.

Furthermore, we successfully showed that the robust solver (Chapter 4) helps mitigate against the effects of low-level RFI (which can be missed by the flagger) on calibration. By adequately downweighting the RFI-contaminated visibilities, the robust solver prevents them from blowing up the calibration solutions and propagating the RFI into the corrected visibilities. The use of solution intervals as regularising tool for the calibration problem, already mentioned in Chapter 4, was thoroughly investigated in Chapter 5. We showed that there is a link between flux suppression, effects of RFI and solution intervals used during calibration. We discussed the different

factors that influence the choice of solution intervals and proposed a brute force algorithm for finding optimal solution intervals. We successfully applied this algorithm to both simulated and real data.

In summary, the thesis discusses and provides guidelines for robust calibration of radio interferometers, which is essential for the future SKA and the new generation of radio telescopes, with the primary objective to reduce flux suppression and the effects of RFI. We now discuss several ways in which we can extend the material presented in this thesis.

The flux suppression results (particularly Chapter 4, Fig. 4.3) showed some surprising trends. For example, why do we have more flux suppression from traditional solvers when calibrating with dispersed sky models? This strong link between flux suppression and model concentration needs to be properly investigated. This involves revisiting and extending the calibration artefact studies of [Grobler et al. \(2014\)](#) and [Wijnholds et al. \(2016\)](#).

While we briefly discussed extended emission, the simulation in §4.1.4 is relatively simplistic, and only serves as a proof of concept for the applicability of the robust solver to preserve the signal of extended sources and diffuse emission during calibration. Hence, future work will entail performing more realistic simulations for diffuse emission. Furthermore, the robust solver needs to be tested on a variety of different data sets. An interesting example will be polarisation calibration, which generally requires very complicated models. Future research will include extending the analysis in §4.1.1 using a Bayesian approach and different heavy-tailed distributions in order to find the best probability distributions for visibilities. Such an approach will provide more statistical evidence to why using heavy-tailed distributions can improve calibration for data containing outliers.

On the implementation side, proper benchmarking and profiling needs to be performed in order to identify the optimal settings for the robust solver and the different scenarios under which it improves calibration. The fact that the robust-**I** solver performs remarkably well in the high SNR simulation (when the covariance of the residuals is much higher than **I**) suggests that by scaling down the covariance, we can improve the results of the robust solver. This aspect needs further investigation. Because the robust solver is entirely independent of the RIME model, extending it to CubiCal's different specialised solvers should be a straightforward task.

Regarding the question of finding optimal solution intervals, we highlighted some future perspectives in §5.6. While solution intervals are an excellent tool for regularising the calibration problem, solution intervals have two significant limitations. Firstly, because of the different factors such as variability of the gains, RFI in the data and unmodelled sources, it is not an easy task to find the optimal solution interval during calibration. Secondly, because of the enormous data rates of the new radio telescopes, we are seeking more data parallelism with distributed systems for calibration. This approach requires splitting datasets into small chunks which we can calibrate and process in parallel on different computing resources. This approach is a limitation for solution intervals, since solution intervals will be constrained by the size of the small data chunks. Hence, we need to think beyond solution intervals, and implement solvers that do not need solution intervals for regularisation (see [Yatawatta \(2015b\)](#) and [Yatawatta et al. \(2019\)](#) for few examples already in that direction). These approaches, however, are also limited by a continuous model selection problem as opposed to a discrete model selection problem.

Non Linear Least Squares (NLLS) methods

The objective of any optimisation or minimisation problem is to find a real variable $x^* \in \mathbb{R}^n$ such that

$$x^* \equiv \min_x Q(x), \quad (\text{A.1})$$

where $n \geq 1$ and $Q : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth function called the objective function (Nocedal & Wright, 2006). If there exists x^* which satisfies Eq. (A.1), then x^* is called the global minimiser of Q . If x^+ satisfies Eq. (A.1) in a limited domain, then x^+ is called a local minimiser, i.e.

$$x^+ = \min_x Q(x), \quad \forall |x - x^+| \leq \delta, \quad (\text{A.2})$$

where δ defines the region for which x^+ is a local minimiser of Q . An optimisation problem is termed *least squares* if the objective function is defined as the sum of the residual squares between the data and model, i.e.

$$x^* = \min_x F(x), \quad (\text{A.3})$$

where $F(x) = \frac{1}{2} \sum_{i=1}^m |f(x_i) - y_i|^2$, $f(x)$ is the model function, y denotes the measured data to be fitted and m is the total number of data points. Note that the factor $\frac{1}{2}$ is only included for convenience and it has no effect on the solution.

Least squares optimisation problems can be divided into categories, namely: linear and non linear depending on whether the model is a linear function of the parameters or not. Optimisation problems are usually solved iteratively. We start by making an initial guess x_0 for x^* , and at every

k^{th} iteration we update x^* as

$$x_k^* = x_{k-1}^* + h, \quad (\text{A.4})$$

where h is the size of our update step. The process is repeated until x^* converges,

$$\text{i.e. } |x_k^* - x_{k-1}^*| \leq \epsilon \text{ or } |F(x)| \leq \epsilon,$$

where ϵ is a chosen threshold. Different optimisation algorithms exist, but they all follow the above approach, the main difference being how we compute the update step h .

Gauss-Newton (GN) algorithm

Consider the following objective function

$$F(x) = \frac{1}{2} \sum_{i=1}^m |\mathbf{r}_i(x)|^2 = \frac{1}{2} \|\mathbf{r}(x)\|_F^2 = \frac{1}{2} \mathbf{r}(x)^T \mathbf{r}(x), \quad (\text{A.5})$$

where \mathbf{r} is the residual function and $\|\cdot\|_F$ denotes the Frobenius norm. If \mathbf{r} has continuous second order partial derivatives, then its Taylor series approximation is

$$\mathbf{r}(x+h) = \mathbf{r}(x) + \mathbf{J}(x)h + O(h^2), \quad (\text{A.6})$$

where $\mathbf{J} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix of the problem with its elements defined as

$$\mathbf{J}(x)_{ij} = \frac{\partial \mathbf{r}_i}{\partial x_j}(x).$$

The Taylor series of \mathbf{r} can be written in the following form as a function l of h :

$$\mathbf{r}(x+h) \approx l(h) \equiv \mathbf{r}(x) + \mathbf{J}(x)h. \quad (\text{A.7})$$

Substituting Eq. (A.7) into Eq. (A.5) we have

$$F(x+h) \approx L(h) \equiv \frac{1}{2} l(h)^T l(h) \quad (\text{A.8})$$

$$\equiv \frac{1}{2} \mathbf{r}^T \mathbf{r} + h^T \mathbf{J}^T \mathbf{r} + \frac{1}{2} h^T \mathbf{J}^T \mathbf{J} h \quad (\text{A.9})$$

$$\equiv F(x) + h^T \mathbf{J}^T \mathbf{r} + \frac{1}{2} h^T \mathbf{J}^T \mathbf{J} h, \quad (\text{A.10})$$

where $f = f(x)$ and $\mathbf{J} = \mathbf{J}(x)$. The update step for the GN algorithm h is defined such that $L(h)$ is minimised (Madsen et al., 2004), i.e.

$$h_{GN} = \min_x L(h).$$

Taking the first and second derivatives of $L(h)$, we have

$$L'(h) = \mathbf{J}^T \mathbf{r} + \mathbf{J}^T \mathbf{J} h, \quad (\text{A.11})$$

$$L''(h) = \mathbf{J}^T \mathbf{J}. \quad (\text{A.12})$$

If $L'(h_{GN}) = 0$, then h_{GN} is given by

$$h_{GN} = -(\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{r}. \quad (\text{A.13})$$

Note that the presence of the negative sign in Eq. (A.12) depends on the convention used to define the residual function, i.e data - model or model - data. The algorithm is usually implemented with the full update for x^* given as

$$x_k^* = x_{k-1}^* + \alpha h_{GN}, \quad (\text{A.14})$$

where α is called the learning rate and it controls the size of the step we take at every iteration towards the solution.

Levenberg-Marquardt (LM) algorithm

The GN sometimes suffers from slow convergence issues, additionally the matrix $\mathbf{J}^T \mathbf{J}$ is not always full rank, and hence not guaranteed to be positive semi-definite. Hence, h_{GN} may not always be the right step towards the global minimum. Suggested by both Levenberg (1944) and Marquardt (1963), the LM algorithm is a slight modification of the GN algorithm. Here the updated h_{LM} is defined as

$$(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}) h_{LM} = -\mathbf{J}^T \mathbf{r} \quad (\text{A.15})$$

$$h_{LM} = -(\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})^{-1} \mathbf{J}^T \mathbf{r} \quad (\text{A.16})$$

where $\mu > 0$ and it is called the damping factor. Note that different implementations exist. A popular one consists of replacing the term $\mu\mathbf{I}$ by μD , where $D = \text{Diag}(\mathbf{J}^T \mathbf{J})$ is the diagonal matrix constructed using the leading diagonal elements of $\mathbf{J}^T \mathbf{J}$. The damping factor ensures that $\mathbf{J}^T \mathbf{J} + \mu\mathbf{I}$ is positive semi-definite. If $\mu = 0$ or is very small, we simply have the GN algorithm. Whenever μ is very large, the LM algorithm is equivalent to the steepest gradient descent algorithm where $h = F'(x)$. Different variants of this algorithm exist, depending on how we compute the damping factor μ and the learning rate α .

Expectation Maximisation for Non-linear Models with Proper CST Noise

Suppose we have a measurement model given by

$$\mathbf{y} = f(\mathbf{x}) + \epsilon, \quad \text{where } \mathbf{y} \in \mathbb{C}^D, \mathbf{x} \in \mathbb{C}^M, \epsilon \sim \text{CST}(\mathbf{y}|\boldsymbol{\mu} = f(\mathbf{x}), \boldsymbol{\Lambda}, v) \quad (\text{B.1})$$

where $f : \mathbb{C}^M \rightarrow \mathbb{C}^D$ is some non-linear function and we want to find the maximum likelihood (ML) estimate of \mathbf{x} given a set of data \mathbf{y} . The ML solution requires solving for the parameters \mathbf{x} , as well as the parameters defining the CST distribution, i.e. v and $\boldsymbol{\Lambda}$. We will denote these as a single parameter vector, θ . Unfortunately, the CST is not part of the exponential family, and the log-likelihood is generally clumsy to work with. We will now illustrate how the ML solution can be obtained using an iteratively reweighted complex NLLS algorithm.

From §3.2, it is clear that we can view the distribution of each data point \mathbf{y}_i as an infinite mixture of proper complex normal distributions with variance drawn from a Gamma distribution i.e.

$$P(\mathbf{y}_i|\theta) = \int_0^\infty \text{CN}(\mathbf{y}_i|\boldsymbol{\mu}_i = f(\mathbf{x}), (\tau_i \boldsymbol{\Lambda})^{-1}) \text{Gam}(\tau_i|v, v) d\tau_i, \quad (\text{B.2})$$

where we have left the dependence of f implicit for notational simplicity. In this case, assuming

conditional independence of the data, the likelihood given N data points is simply

$$P(\mathbf{y}|\theta) = \prod_{i=1}^N P(\mathbf{y}_i|\theta),$$

$$P(\mathbf{y}|\theta) = \int d\mathbf{z} \prod_{i=1}^N \text{CN}(\mathbf{y}_i|\boldsymbol{\mu}_i, (\tau_i\boldsymbol{\Lambda})^{-1}) \text{Gam}(\tau_i|v, v) \quad (\text{B.3})$$

where we have denoted $\mathbf{z} = [\tau_1, \tau_2, \dots, \tau_N]^T$ as the set of latent variables corresponding to the scale parameter for each data point, and used the fact that all the τ_i are independent to exchange the order of the product and the integral.

The form (B.3) can now be solved using the expectation maximisation (EM) algorithm. For notational convenience, we denote the joint density (i.e. the integrand of (B.3)) by $P(\mathbf{y}, \mathbf{z}|\theta)$. The key idea behind EM is to identify latent variables \mathbf{Z} , governed by a distribution $q(\mathbf{z})$ for example, for which the joint density $P(\mathbf{y}, \mathbf{z}|\theta)$ is easier to evaluate than the marginal in (B.3). The trick is then to decompose the log of the marginal density into two functionals viz.

$$\log P(\mathbf{y}|\theta) = \int d\mathbf{z} q(\mathbf{z}) \log \left(\frac{P(\mathbf{y}, \mathbf{z}|\theta)}{q(\mathbf{z})} \right) - \int d\mathbf{z} q(\mathbf{z}) \log \left(\frac{P(\mathbf{z}|\mathbf{y}, \theta)}{q(\mathbf{z})} \right), \quad (\text{B.4})$$

$$= \mathcal{L}(q(\mathbf{z}), \theta) + \text{KL}(q(\mathbf{z})\|P(\mathbf{z}|\mathbf{y}, \theta)). \quad (\text{B.5})$$

Noting that the last term is the Kullback-Leibler divergence, which satisfies $\text{KL}(q(\mathbf{z})\|P(\mathbf{z}|\mathbf{y}, \theta)) \geq 0$ with equality holding iff $q(\mathbf{z}) = P(\mathbf{z}|\mathbf{y}, \theta)$, we see that $\mathcal{L}(q(\mathbf{z}), \theta)$ is a lower bound on $\log P(\mathbf{y}|\theta)$. This implies that the optimal choice for $q(\mathbf{z})$ is the true posterior distribution $P(\mathbf{z}|\mathbf{y}, \theta)$ at the ML solution of θ since then $\mathcal{L}(q(\mathbf{z}), \theta) = \log P(\mathbf{y}|\theta)$. However, since we do not have this solution, we adopt an iterative procedure which involves setting $q(\mathbf{z}) = P(\mathbf{z}|\mathbf{y}, \theta_k)$ at each step k . Substituting into the expression for $\mathcal{L}(q(\mathbf{z}), \theta)$ gives

$$\mathcal{L}(q(\mathbf{z}), \theta) = \int d\mathbf{z} P(\mathbf{z}|\mathbf{y}, \theta_k) \log P(\mathbf{y}, \mathbf{z}|\theta) \quad (\text{B.6})$$

$$- \int d\mathbf{z} P(\mathbf{z}|\mathbf{y}, \theta_k) \log P(\mathbf{y}, \mathbf{z}|\theta_k),$$

$$= \int d\mathbf{z} P(\mathbf{z}|\mathbf{y}, \theta_k) \log P(\mathbf{y}, \mathbf{z}|\theta) + \text{const.} \quad (\text{B.7})$$

Thus we see that to maximise $\mathcal{L}(q(\mathbf{z}), \theta)$ at θ_k we need to compute the expectation value of $\log P(\mathbf{y}, \mathbf{z}|\theta)$ with respect to the posterior distribution $P(\mathbf{z}|\mathbf{y}, \theta_k)$. This is known as the E-step

and it defines a function which we can subsequently maximise viz.

$$Q(\theta, \theta_k) = \mathbb{E}_{P(\mathbf{z}|\mathbf{y}, \theta_k)} [P(\mathbf{y}, \mathbf{z}|\theta)]. \quad (\text{B.8})$$

To solve the ML problem, we now need to solve each of the following problems in order

$$\nabla_{\mathbf{x}} Q = 0, \quad \nabla_{\mathbf{\Lambda}} Q = 0, \quad \text{and} \quad \nabla_v Q = 0. \quad (\text{B.9})$$

This is known as the M-step and we can iterate between the M-step and the E-step until convergence.

The required joint density (also known as the complete likelihood function) for all N observations Y is given by the integrand of (B.3) i.e.

$$P(\mathbf{y}, \mathbf{z}|\theta) = \prod_{i=1}^N \text{CN}(\mathbf{y}_i | \boldsymbol{\mu}_i(\mathbf{x}), (\tau_i \mathbf{\Lambda})^{-1}) \text{Gam}(\tau_i | v, v). \quad (\text{B.10})$$

The complete log-likelihood function is therefore given by

$$\begin{aligned} \log P(\mathbf{y}, \mathbf{z}|\theta) &\propto \sum_{i=1}^N D \log \tau_i + N \log |\mathbf{\Lambda}| - \sum_{i=1}^N \tau_i \Delta_i^2(\mathbf{x}, \mathbf{\Lambda}) \\ &\quad + Nv \log(v) + (v-1) \sum_{i=1}^N \log(\tau_i) - N \log(\Gamma(v)) \\ &\quad - v \sum_{i=1}^N \tau_i, \end{aligned} \quad (\text{B.11})$$

Next, we need to compute the expectation value of $\log P(\mathbf{y}, \mathbf{z}|\theta)$ w.r.t. $P(\mathbf{z}|\mathbf{y}, \theta_k)$. Using the product rule of probability, we see that

$$P(\mathbf{z}|\mathbf{y}, \theta_k) = \frac{P(B\mathbf{z}, \mathbf{y}|\theta_k)}{P(\mathbf{y}|\theta_k)} \propto P(\mathbf{z}, \mathbf{y}|\theta_k), \quad (\text{B.12})$$

where we have used the fact that all terms independent of \mathbf{z} are irrelevant when computing the expectation values in (B.8). We can therefore evaluate the conditional density up to a normalisation

constant as

$$P(\mathbf{z}|\mathbf{y}, \theta_k) \propto \prod_{i=1}^N \text{CN}(\mathbf{y}_i | \boldsymbol{\mu}_i(\mathbf{x}_k), (\tau_i \boldsymbol{\Lambda}_k)^{-1}) \text{Gam}(\tau_i | v_k, v_k), \quad (\text{B.13})$$

$$\propto \prod_{i=1}^N \tau_i^{D+v_k-1} \exp(-\tau_i(v_k + \Delta_k^2)), \quad (\text{B.14})$$

$$\propto \prod_{i=1}^N \text{Gam}(\tau_i | v_k + D, v_k + \Delta_k^2), \quad (\text{B.15})$$

where we have obtained the parameters of the Gamma distribution by inspection. This is actually very convenient, because the terms for which we need expectation values in (B.8) (i.e. 1 , τ_i and $\log(\tau_i)$) can all be obtained analytically using the well known properties of the Gamma distribution. They are¹

$$\mathbb{E}[1] = 1, \quad (\text{B.16})$$

$$\mathbb{E}[\tau_i] = \frac{v_k + D}{v_k + \Delta^2(\mathbf{x}_k, \boldsymbol{\Lambda}_k)}, \quad (\text{B.17})$$

$$\mathbb{E}[\log(\tau_i)] = \psi(v_k + D) - \log(v_k + \Delta^2(\mathbf{x}_k, \boldsymbol{\Lambda}_k)). \quad (\text{B.18})$$

This implies that using (B.11) we can rewrite (B.8) as

$$\begin{aligned} Q(\theta, \theta_k) &= D \sum_i \mathbb{E}[\log(\tau_i)] + N \log|\boldsymbol{\Lambda}| + \sum_i \Delta_i^2(\mathbf{x}, \boldsymbol{\Lambda}) \mathbb{E}[\tau_i] \\ &\quad + Nv \log(v) + (v-1) \sum_{i=1}^N \mathbb{E}[\log(\tau_i)] - N \log(\Gamma(v)) \\ &\quad - v \sum_{i=1}^N \mathbb{E}[\tau_i]. \end{aligned} \quad (\text{B.19})$$

Note that the dependence on θ_k is implicit in the expressions for the expectation values. To solve the ML problem, we first need to solve

$$\nabla_{\mathbf{x}} Q = 0. \quad (\text{B.20})$$

Since the dependence on \mathbf{x} is confined to the Δ^2 term, this amounts to solving

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmin}} \sum_i (\mathbf{y}_i - f(\mathbf{x}))^H \boldsymbol{\Lambda} (\mathbf{y}_i - f(\mathbf{x})) \mathbb{E}[\tau_i]. \quad (\text{B.21})$$

¹Note specifically that the expression for $\mathbb{E}[\log(\tau_i)]$ differs from the real valued case.

This is just a weighted NLLS problem. However, note that the objective function is a real valued function of complex variables, which is not holomorphic for all choices of f . Thus, we require the machinery of Wirtinger calculus to tackle it. With this solution in hand, the next step is to solve

$$\nabla_{\mathbf{\Lambda}} Q = 0. \quad (\text{B.22})$$

It is not a fact that $\mathbf{\Lambda}$ is diagonal; we assume this to reduce the computational cost of this step. As it stands, the Hermitian symmetry of $\mathbf{\Lambda}$ implies that the diagonal part has to be real valued and we can proceed as normal. The solution is available in closed form and is given by

$$\hat{\mathbf{\Lambda}} = \left(\frac{1}{N} \sum_{i=1}^N (\mathbf{y}_i - f(\hat{\mathbf{x}}))(\mathbf{y}_i - f(\hat{\mathbf{x}}))^H \mathbb{E}[\tau_i] \right)^{-1}. \quad (\text{B.23})$$

Finally, we need to update the value of v by solving

$$\nabla_v Q = 0 = N \log(v) + N + \sum_{i=1}^N \mathbb{E}[\log(\tau_i)] - N\psi(v) - \sum_{i=1}^N \mathbb{E}[\tau_i]. \quad (\text{B.24})$$

This last expression needs to be solved numerically using a root finding algorithm if v is continuous or using grid search if v is assumed to be an integer. Once $\hat{\theta}$ has been obtained, we can re-evaluate the expectation values (B.17) and (B.18) (E-step) and perform another M-step. This process is iterated until convergence, and we therefore refer to it as an iteratively reweighted complex NLLS algorithm.

Radio interferometric gain calibration

Let $\mathbf{d} = [d_{pq}]$ and $\mathbf{v} = [v_{pq}]$ respectively represent the vectorised observed and modelled visibilities. Using Equation (B.17), with D replaced by the number of correlations, n_c , the weights are given by

$$w_{pq} = \frac{v + n_c}{v + (\mathbf{d}_{pq} - \mathbf{v}_{pq})^H \mathbf{\Sigma}^{-1} (\mathbf{d}_{pq} - \mathbf{v}_{pq})}, \quad (\text{B.25})$$

where $\mathbf{\Sigma} = \hat{\mathbf{\Lambda}}^{-1}$. If $\mathbf{\Sigma}$ is assumed to be \mathbf{I} (identity matrix of appropriate shape), the weights become

$$w_{pq} = \frac{v + n_c}{v + \|\mathbf{d}_{pq} - \mathbf{v}_{pq}\|^2}. \quad (\text{B.26})$$

Boxcars parametrisation of the gains

Using a suitable choice of design matrix, Eq. (5.23) can be written as a linear matrix vector equation. For example, for each frequency, we may define

$$R_\nu = \begin{bmatrix} \text{ones}(n_\nu, 1) & 0 & \cdots \\ 0 & \text{ones}(n_\nu, 1) & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{C.1})$$

where $\text{ones}(n, m)$ represents an $n \times m$ matrix of ones. Next, we stack n_t copies of R_ν on top of each other to define the design matrix for a single time and frequency interval i.e.

$$R = \begin{bmatrix} R_\nu \\ R_\nu \\ \vdots \end{bmatrix} \quad (\text{C.2})$$

Finally, for the full design matrix X , we simply stack M_t copies of R into the “diagonal” as follows

$$X = \begin{bmatrix} R & 0 & \cdots \\ 0 & R & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (\text{C.3})$$

Thus, assuming we have a $M_t \times M_\nu$ matrix of parameters, Θ_p say, the gain can be written as

$$\mathbf{g}_p = X\theta_p \quad (\text{C.4})$$

where $\theta_p = \text{vec}(\Theta_p)$ is the vector obtained when stacking rows of Θ_p on top of each other.

Simulation Tools

This appendix presents the general framework and tools which we used to perform the simulations in the thesis. These tools are standalone recipes, which can be extended and used for different radio interferometric simulations. §D.1 describes the different components of our simulation pipeline, while §D.2 describes how we generate realistic propagation effects (gains).

D.1 Simulation pipeline

Each simulation pipeline is built using the following components.

- **Creating Measurement Sets (MS):** Throughout the thesis, we create measurement sets using the SIMMS¹ tool. SIMMS is a Python wrapper around CASA that facilitates the creation of empty measurement sets. These are created by passing the observational settings we want to simulate. We simulate visibilities for different arrays. The most employed arrays are MeerKAT and VLA. We simulated both single frequencies and multi-channel observations. Unless specified otherwise, the basic configuration in Table D.1 was used during simulations.
- **Creating sky models:** The sky models we use for simulations have positions drawn from a uniform distribution. The fluxes of the sources are drawn from a power law. In particular,

¹SIMMS: <https://github.com/SpheMakh/simms>

Integration time	10 secs
Synthesis time	2 hours
Start frequency	1 GHz
Bandwidth	1 MHz
Number of channels	1
Array	MeerKAT

Table D.1: Basic MS configuration

we used the Pareto distribution which has probability density function, $p(x)$, defined as follows

$$p(x) = \frac{am^a}{x^{a+1}}, \quad (\text{D.1})$$

where a is called the shape parameter and m is the scale. We exclusively use a value of $m = 1$ for the scale parameter. In the different simulations, we scaled the sampled fluxes such that the peak flux has different desired values. Thus, if s are the sampled fluxes, then scaled fluxes, denoted s_p , with a peak, p , are given by

$$s_p = p \cdot \frac{s}{\max(s)}. \quad (\text{D.2})$$

Another scaling used in certain simulations is to scale fluxes in a model such that they sum to a certain value. This scaling is defined as

$$s_T = T \cdot \frac{s}{\text{sum}(s)}, \quad (\text{D.3})$$

where s_T are the fluxes scaled such that the total flux in the model is T . Except for §4.1.4 all simulations are comprised entirely of point-like sources.

- **Computing visibilities:** Our simulation framework uses the software packages Montblanc (Perkins et al., 2015) and MeqTrees (Noordam & Smirnov, 2010) to compute the RIME. In our framework, we used Montblanc and MeqTrees only to compute uncorrupted visibilities. Montblanc uses GPU acceleration. Hence it is more efficient when a large number of sources, frequency channels and observation times are involved. MeqTrees, on the other

hand, provides the possibility of including different sorts of possible effects such as smearing, and it is very effective for simulations, since it can also be used for calibration as well.

- Corrupting visibilities: Visibilities are corrupted using functionalities of the CubiCal ([Kenyon et al., 2018](#)) package.
- The measurement set tool, Python-cascore² is used to manipulate the data in the created measurement sets. Using this tool, we can read and write visibilities to arbitrarily created measurement set columns. Additionally, we added noise to the corrupted visibilities using Python-casacore script.

D.2 Generating realistic gains

This section focuses on the effects corrupting the visibilities, i.e gains. We simulate gains as stochastic processes with predefined statistical properties using Gaussian Processes (GP). The material presented is mostly taken from [Rasmussen & Williams \(2006\)](#).

Gaussian Processes (GP)

GP are a family of stochastic processes defined such that every finite subset of random variables drawn from it follows a multivariate normal distribution. A GP can be interpreted as a Gaussian distribution over functions defined on continuous domains called input fields (examples are time and space). Mathematically, a GP is specified by two functions, namely a *mean* and a *covariance* or *kernel* function. For a real process, $f(x)$, on an input field, x , the mean function, $m(x)$, and the covariance function, $k(x, x')$, are given by

$$m(x) = \mathbb{E}[f(x)], \tag{D.4}$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]. \tag{D.5}$$

²Python-casacore: <https://github.com/casacore/python-casacore>

Note that the covariance is a function of two input field locations represented by x and x' here. As we will discuss in §D.2, the covariance function determines the nature of the process and encodes all prior information about the specific process. We then write the GP, $f(x)$, as

$$f(x) \sim \text{N}(m(x), k(x, x')).$$

In order to sample a GP with a specific prior covariance, $k(x, x')$, and a mean function, $m(x')$, we first construct its Gram matrix, K , as

$$K_{ij} = k(x_i^*, x_j^*), \quad (\text{D.6})$$

where x_i^* and x_j^* are the i^{th} and j^{th} desired input field locations. Using the constructed Gram matrix, samples at input field points x_* are simply drawn from the following Normal distribution,

$$f(x^*) \sim \text{N}(m^*, K), \quad (\text{D.7})$$

where $m^* = m(x^*)$ is the mean function evaluated at the desired field points. This can be done using random samples drawn from the standard normal distribution as follows.

If samples, u , are drawn from the distribution, $\text{N}(0, \mathbf{I})$, where 0 is a vector with all entries 0 and \mathbf{I} is the identity matrix, then

$$f(x^*) \equiv m^* + K^{1/2}u, \quad (\text{D.8})$$

has mean m^* and covariance K .

Covariance functions

The covariance function is a crucial component of any GP. It contains the assumptions on the statistical properties of the GP and describes how similar the data points are with respect to each other. GP are an example of semiparametric models where the model has latent parameters called hyper-parameters which parametrises the covariance function in some intuitive way. A function, k , of input pairs x and x' is a valid covariance function if and only if its Gram matrix, K , is positive semidefinite, i.e.

$$\forall v \in \mathbb{R}^n, \quad v^T K v \geq 0.$$

For this thesis, a specific class of covariance functions called *stationary covariance functions* are employed. A stationary covariance function is a function of the separation between input pairs only, $\tau = x - x'$. Stationary covariance functions are invariant under translation. Furthermore if a covariance function is a function of the absolute separation, $|\tau| = |x - x'|$, it is called isotropic. Isotropic covariance functions result in block-wise Toeplitz Gram matrices, which can be transformed into circulant³ Gram matrices when on a regular grid.

The Wiener-Khinchine theorem states that the Fourier transform of a stationary covariance function is its Power Spectrum, if the latter exists. Hence, if a kernel function, $k(\tau)$, has a power spectrum, $S(s)$, then

$$k(\tau) = \int S(s)e^{2\pi is \cdot \tau} ds, \quad S(s) = \int k(\tau)e^{-2\pi is \cdot \tau} d\tau. \quad (\text{D.9})$$

Converting between the power spectrum and the kernel function is an essential tool which allows us to accelerate computations by employing the Fast Fourier Transform (FFT) and the convolution property. Additionally, using a physically motivated spectrum, such as the Kolmogorov energy spectrum, it is possible to simulate processes like turbulence (Frisch & Kolmogorov, 1995).

Squared Exponential Covariance function

The Squared Exponential (SE) covariance function is defined as

$$k(\tau) = k(x - x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) = \sigma_f^2 \exp\left(-\frac{\tau^2}{2l^2}\right). \quad (\text{D.10})$$

The two hyper parameters σ_f and l define the standard deviation of the GP and its characteristic length scale (i.e. the input separation required for the GP to vary significantly) respectively. The SE covariance function is infinitely differentiable and hence samples drawn from processes with

³Circulant matrices are a special kind of Toeplitz matrix where each row vector or column vector is shifted one element of the preceding row or column (Davis, 2013). A discrete Fourier transform diagonalises circulant matrices. Hence, most operations containing them can be quickly performed using a Fast Fourier transform, if the data is on a regular grid.

SE kernel are extremely smooth. Its power spectrum exists and is given by

$$S(s) = \sigma_f^2 (2\pi l^2)^{D/2} \exp(-2\pi^2 l^2 s^2), \quad (\text{D.11})$$

where D represents the dimension of the input field vector.

The Matérn Class of Covariance functions

The extreme smoothness of the SE covariance function makes it an unrealistic assumption for certain physical processes. The Matérn class of covariance functions is less smooth and is defined as follows ([Rasmussen & Williams, 2006](#))

$$k_{\text{matern}}(\tau) = \sigma_f^2 \frac{2^{1-v}}{\Gamma(v)} \left(\frac{\sqrt{2}v\tau}{l} \right)^v K_v \left(\frac{\sqrt{2}v\tau}{l} \right), \quad (\text{D.12})$$

where K_v is the modified Bessel function, and Γ is the Gamma function. σ_f and l play the same rule as in the SE covariance function. The additional parameter v controls the smoothness of the GP. A GP with Matérn covariance is $v - 1$ times differentiable. Extremely smooth processes are generated by using high values for v , while rough processes are generated using small values for v . Note that, as v approaches ∞ , the Matérn covariance turns into the SE covariance. Its generalised power spectrum is given by

$$S(s) = \sigma_f^2 \frac{2^D \pi^{D/2} \Gamma(v + D/2) (2v)^2}{\Gamma(v) l^{2v}} \left(\frac{2v}{l^2} + 4\pi^2 s^2 \right)^{-(v+D/2)}. \quad (\text{D.13})$$

In our simulations we used Matérn covariances with $v = 3/2$ and $v = 5/2$.

Fast sampling

Propagation effects in radio interferometry can be modelled as fields defined on time, frequency and spatial domains. Hence the gain realisations need to be sampled on a multi-dimensional input space. A common bottleneck with GP is the high time and memory complexity required by such algorithms. Sampling gains as described in §D.2 can be hugely hindered by the cost of computing the square root of the Gram matrix, K , and the product $K^{1/2}u$. To accelerate the sampling process, we used methods described in §5 of [Saatçi \(2012\)](#).

Bibliography

- Akaike, H. 1974, in *Selected Papers of Hirotugu Akaike* (Springer), 215–222
- . 1998, in *Selected papers of hirotugu akaike* (Springer), 199–213
- Arras, P., Frank, P., Leike, R., Westermann, R., & Enßlin, T. A. 2019, *Astronomy & Astrophysics*, 627, A134
- Atemkeng, M., Smirnov, O., Tasse, C., Foster, G., & Jonas, J. 2016, *Monthly Notices of the Royal Astronomical Society*, 462, 2542
- Barry, N., Hazelton, B., Sullivan, I., Morales, M., & Pober, J. 2016, *Monthly Notices of the Royal Astronomical Society*, stw1380
- Bhatnagar, S., Cornwell, T., Golap, K., & NRAO, S. 2004, EVLA Memo 84. Solving for the antenna based pointing errors, Tech. rep., Tech. rep., NRAO
- Bishop, C. M. 2006, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Berlin, Heidelberg: Springer-Verlag)
- Bonnassieux, E., Tasse, C., Smirnov, O., & Zarka, P. 2018, *Astronomy & Astrophysics*, 615, A66
- Booth, R. S., & Jonas, J. L. 2012, *African Skies*, 16, 101
- Born, M., & Wolf, E. 1964, *Principles of Optics Electromagnetic Theory of Propagation, Interference and Diffraction of Light* 2nd edition by Max Born, Emil Wolf New York, NY: Pergamon Press, 1964, 1
- Briggs, D. S. 1995, PhD thesis, New Mexico Institute of Mining and Technology

- Brogan, C. L., Hunter, T. R., & Fomalont, E. B. 2018, arXiv preprint arXiv:1805.05266
- Buchner, J. 2016, Astrophysics Source Code Library
- Burnell, J. B. 1984, in *Serendipitous Discoveries in Radio Astronomy*, 160
- Candes, E. J., Romberg, J. K., & Tao, T. 2006, *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59, 1207
- Carrillo, R. E., McEwen, J. D., & Wiaux, Y. 2014, *Monthly Notices of the Royal Astronomical Society*, 439, 3591
- Chen, M., Deng, H., Wang, F., & Ji, K. 2013, in *2013 6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, IEEE, 324–327
- Chiarucci, S., & Wijnholds, S. J. 2017, *Monthly Notices of the Royal Astronomical Society*, 474, 1028
- Colafrancesco, S., Regis, M., Marchegiani, P., et al. 2015, arXiv preprint arXiv:1502.03738
- Combettes, P. L., & Wajs, V. R. 2005, *Multiscale Modeling & Simulation*, 4, 1168
- Condon, J. J., & Ransom, S. M. 2016, *Essential radio astronomy*, Vol. 2 (Princeton University Press)
- Cornwell, T. J., & Evans, K. 1985, *Astronomy and Astrophysics*, 143, 77
- Cornwell, T. J., & Wilkinson, P. N. 1981, *Monthly Notices of the Royal Astronomical Society*, 196, 1067
- Davis, P. J. 2013, *Circulant matrices* (American Mathematical Soc.)
- De Blok, W., Adams, E., Amram, P., et al. 2017, arXiv preprint arXiv:1709.08458
- DeBoer, D. R., Parsons, A. R., Aguirre, J. E., et al. 2017, *Publications of the Astronomical Society of the Pacific*, 129, 045001

- Ekers, R. D. 1984, in *Serendipitous Discoveries in Radio Astronomy*, 154
- Fisher, R. A., et al. 1920
- Frisch, U., & Kolmogorov, A. N. 1995, *Turbulence: the legacy of AN Kolmogorov* (Cambridge university press)
- Greenhill, L., & Bernardi, G. 2012, arXiv preprint arXiv:1201.1700
- Grobler, T., Nunhokee, C., Smirnov, O., Van Zyl, A., & De Bruyn, A. 2014, *Monthly Notices of the Royal Astronomical Society*, 439, 4030
- Grobler, T., Stewart, A., Wijnholds, S., Kenyon, J., & Smirnov, O. 2016, *Monthly Notices of the Royal Astronomical Society*, 461, 2975
- Grobler, T. L., Bernardi, G., Kenyon, J. S., Parsons, A. R., & Smirnov, O. M. 2018, *Monthly Notices of the Royal Astronomical Society*, 476, 2410
- Hamaker, J. P., Bregman, J. D., & Sault, R. J. 1996, *Astronomy & Astrophysics*, 117, 137
- Healy, J. L. 2016, PhD thesis, University of Cape Town
- Heywood, I., Hale, C. L., Jarvis, M., J., et al. submitted, *Monthly Notices of the Royal Astronomical Society*
- Högbom, J. 1974, *Astronomy and Astrophysics Supplement Series*, 15, 417
- Iheanetu, K., Girard, J. N., Smirnov, O., et al. 2019, *Monthly Notices of the Royal Astronomical Society*, 485, 4107
- Jansen, M., & Claeskens, G. 2011, *Cramér–Rao Inequality*, ed. M. Lovric (Berlin, Heidelberg: Springer Berlin Heidelberg), 322–323
- Jansky, K. G. 1933, *Proceedings of the Institute of Radio Engineers*, 21, 1387
- Janssen, M., Goddi, C., van Bemmell, I. M., et al. 2019, arXiv preprint arXiv:1902.01749

- Jarvis, M. J., Taylor, A., Agudo, I., et al. 2017, arXiv preprint arXiv:1709.01901
- Johnston, S., Taylor, R., Bailes, M., et al. 2008, *Experimental astronomy*, 22, 151
- Jonas, J., & Team, M. 2018, *MeerKAT Science: On the Pathway to the SKA (MeerKAT2016)*, eds Taylor P, Camilo F, Leeuw L, Moodley K (International School for Advanced Studies, Trieste, Italy), 277
- Kazemi, S., Hurley, P., Öçal, O., & Cherubini, G. 2015, in 2015 3rd International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa), IEEE, 164–168
- Kazemi, S., & Yatawatta, S. 2013, *Monthly Notices of the Royal Astronomical Society*, 435, 597
- Kenyon, J. 2019, PhD thesis, Rhodes University
- Kenyon, J. S., Smirnov, O. M., Grobler, T. L., & Perkins, S. J. 2018, *Monthly Notices of the Royal Astronomical Society*, 478, 2399
- Komodakis, N., & Pesquet, J.-C. 2015, *IEEE Signal Processing Magazine*, 32, 31
- Kraus, J. D. 1966, *Radio astronomy* (New York: McGraw-Hill, 1966)
- Kreutz-Delgado, K. 2009, ArXiv e-prints, arXiv:0906.4835
- Kullback, S., & Leibler, R. A. 1951, *The annals of mathematical statistics*, 22, 79
- Lange, K. L., Little, R. J., & Taylor, J. M. 1989, *Journal of the American Statistical Association*, 84, 881
- Levenberg, K. 1944, *Quarterly of applied mathematics*, 2, 164
- Linfield, R. 1986, *The Astronomical Journal*, 92, 213
- Lochner, M., Natarajan, I., Zwart, J. T., et al. 2015, *Monthly Notices of the Royal Astronomical Society*, 450, 1308

- Lonsdale, C. 2005, in *From Clark Lake to the Long Wavelength Array: Bill Erickson's Radio Science*, Vol. 345, 399
- Madsen, K., Nielsen, H., & Tingleff, O. 2004, *Informatics and Mathematical Modelling Technical University of Denmark*, 60
- Marquardt, D. W. 1963, *Journal of the society for Industrial and Applied Mathematics*, 11, 431
- Marthi, V. R., & Chengalur, J. 2013, *Monthly Notices of the Royal Astronomical Society*, 437, 524
- Martí-Vidal, I., & Marcaide, J. 2008, *Astronomy & Astrophysics*, 480, 289
- McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in *Astronomical data analysis software and systems XVI*, Vol. 376, 127
- McQuinn, M., Zahn, O., Zaldarriaga, M., Hernquist, L., & Furlanetto, S. R. 2006, *The Astrophysical Journal*, 653, 815
- Mitchell, D. A., Greenhill, L. J., Wayth, R. B., et al. 2008, *IEEE Journal of Selected Topics in Signal Processing*, 2, 707
- Mohan, N., & Rafferty, D. 2015, *PyBDSF: Python Blob Detection and Source Finder*, *Astrophysics Source Code Library*, ascl:1502.007
- Morales, M. F., Bowman, J. D., & Hewitt, J. N. 2006, *The Astrophysical Journal*, 648, 767
- Nocedal, J., & Wright, S. 2006, *Numerical optimization* (Springer Science & Business Media)
- Noordam, J. E. 2004, in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Vol. 5489, *Ground-based Telescopes*, ed. J. Oschmann, Jacobus M., 817–825
- Noordam, J. E., & Smirnov, O. M. 2010, *Astronomy & Astrophysics*, 524, A61
- Nunhokee, C. D. 2015, *Master's thesis*, Rhodes University

- Offringa, A., McKinley, B., Hurley-Walker, N., et al. 2014, *Monthly Notices of the Royal Astronomical Society*, 444, 606
- Offringa, A. R., van de Gronde, J. J., & Roerdink, J. B. T. M. 2012, *Astronomy & Astrophysics*, 539, A95
- Offringa, A. R., Wayth, R. B., Hurley-Walker, N., et al. 2015, *Publications of the Astronomical Society of Australia*, 32, e008
- Ollier, V., Korso, M. N. E., Boyer, R., Larzabal, P., & Pesavento, M. 2017, *IEEE Transactions on Signal Processing*, 65, 5649
- Onose, A., Carrillo, R. E., Repetti, A., et al. 2016, *Monthly Notices of the Royal Astronomical Society*, 462, 4314
- Patil, A. H., Yatawatta, S., Zaroubi, S., et al. 2016, *Monthly Notices of the Royal Astronomical Society*, 463, 4317
- Perkins, S. J., Marais, P. C., Zwart, J. T. L., et al. 2015, *Astronomy and Computing*, 12, 73
- Perley, R., Chandler, C., Butler, B., & Wrobel, J. 2011, *The Astrophysical Journal Letters*, 739, L1
- Pratley, L., McEwen, J. D., d’Avezac, M., et al. 2017, *Monthly Notices of the Royal Astronomical Society*, 473, 1038
- Prestage, R. M., Constantikes, K. T., Hunter, T. R., et al. 2009, *Proceedings of the IEEE*, 97, 1382
- Rasmussen, C. E., & Williams, C. K. I. 2006, *Gaussian Processes for Machine Learning* (Springer)
- Readhead, A., & Wilkinson, P. 1978, *The Astrophysical Journal*, 223, 25
- Reber, G. 1940, *Proceedings of the IRE*, 28, 68

- Repetti, A., Birdi, J., Dabbech, A., & Wiaux, Y. 2017, *Monthly Notices of the Royal Astronomical Society*, 470, 3981
- Saatçi, Y. 2012, PhD thesis, Citeseer
- Salvini, S., & Wijnholds, S. J. 2014, *Astronomy & Astrophysics*, 571, A97
- Schilizzi, R. T., Dewdney, P. E. F., & Lazio, T. J. W. 2008, in *Ground-based and Airborne Telescopes II*, Vol. 7012, 70121I
- Schwab, F. 1984, *The Astronomical Journal*, 89, 1076
- Schwab, F. R. 1980, in *1980 Intl Optical Computing Conf I*, Vol. 231, International Society for Optics and Photonics, 18–26
- Smirnov, O. 2011, in *Pointing Errors And The Westerbork Wobble*. presentation at CALIM2011 conference (Manchester 2011)
- Smirnov, O. M. 2011a, *Astronomy & Astrophysics*, 527, A106
- . 2011b, *Astronomy & Astrophysics*, 527, A107
- . 2011c, *Astronomy & Astrophysics*, 527, A108
- Smirnov, O. M., & Tasse, C. 2015, *Monthly Notices of the Royal Astronomical Society*, 449, 2668
- Sob, U. M., Bester, H. L., Smirnov, O. M., Kenyon, J. S., & Grobler, T. L. 2019, *Monthly Notices of the Royal Astronomical Society*, 491, 1026
- Sob, U. M., Bester, L. H., & Smirnov, O. M. in preparation
- Sorber, L., van Barel, M., & de Lathauwer, L. 2012, *SIAM J. Optim.*, 22, 879
- Tasse, C. 2014a, ArXiv e-prints, arXiv:1410.8706
- . 2014b, *Astronomy & Astrophysics*, 566, A127

- Tasse, C., van der Tol, S., van Zwieten, J., van Diepen, G., & Bhatnagar, S. 2013, *Astronomy & Astrophysics*, 553, A105
- Tasse, C., Hugo, B., Mirmont, M., et al. 2018, *Astronomy & Astrophysics*, 611, A87
- Taylor, G. B., Carilli, C. L., & Perley, R. A., eds. 1999, *Astronomical Society of the Pacific Conference Series*, Vol. 180, *Synthesis Imaging in Radio Astronomy II*
- Thompson, A., & d'Addario, L. 1982, *Radio Science*, 17, 357
- Thompson, A. R., Moran, J. M., & Swenson, Jr., G. W. 2017, *Interferometry and Synthesis in Radio Astronomy*, 3rd Edition (Wiley New York et al.), doi:10.1007/978-3-319-44431-4
- van Diepen, G. 2015, *Astronomy and Computing*, 12, 174
- van Haarlem, M. P., Wise, M. W., Gunst, A., et al. 2013, *Astronomy & astrophysics*, 556, A2
- Van Weeren, R., Williams, W., Hardcastle, M., et al. 2016, *The Astrophysical Journal Supplement Series*, 223, 2
- Weltman, A., Bull, P., Camera, S., et al. 2020, *Publications of the Astronomical Society of Australia*, 37
- Wijnholds, S., Grobler, T., & Smirnov, O. 2016, *Monthly Notices of the Royal Astronomical Society*, 457, 2331
- Wijnholds, S., Willis, A., & Salvini, S. 2018, *Monthly Notices of the Royal Astronomical Society*, 476, 2029
- Wilkinson, P., Conway, J., & Biretta, J. 1988, in *The Impact of VLBI on Astrophysics and Geophysics*, Vol. 129, 509–518
- Wilson, T. L., Rohlf, K., & Hüttemeister, S. 2009, *Tools of Radio Astronomy* (Springer-Verlag), doi:10.1007/978-3-540-85122-6
- Wirtinger, W. 1927, *Mathematische Annalen*, 97, 357

Yatawatta, S. 2013, arXiv e-prints, arXiv:1303.1029

Yatawatta, S. 2015a, *Monthly Notices of the Royal Astronomical Society*, 449, 4506

—. 2015b, *Monthly Notices of the Royal Astronomical Society*, 449, 4506

Yatawatta, S., De Clercq, L., Spreeuw, H., & Diblen, F. 2019, arXiv preprint arXiv:1904.05619

Yatawatta, S., Diblen, F., Spreeuw, H., & Koopmans, L. V. E. 2017, *Monthly Notices of the Royal Astronomical Society*, 475, 708