



RHODES UNIVERSITY

A THESIS PRESENTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN
ASTROPHYSICS AT RHODES UNIVERSITY

Advanced Radio Interferometric Simulation and Data Reduction Techniques

by

Sphehshle Makhathini

Supervisor

Prof. Oleg Smirnov
Department of physics and electronics
Rhodes University, South Africa

Co-supervisor

Dr. Ian Heywood
CSIRO Astronomy and Space Science
CSIRO, Australia

June 18, 2017

Abstract

This work shows how legacy and novel radio interferometry software packages and algorithms can be combined to produce high-quality reductions from modern telescopes, as well as end-to-end simulations for upcoming instruments such as the Square Kilometre Array (SKA) and its pathfinders. We first use a **MeqTrees** based simulations framework to quantify how artefacts due to direction-dependent effects accumulate with time, and the consequences of this accumulation when observing the same field multiple times in order to reach the survey depth. Our simulations suggest that a survey like LADUMA (Looking at the Distant Universe with MeerKAT Array), which aims to achieve its survey depth of $16 \mu\text{Jy}/\text{beam}$ in a 72 kHz at 1.42 GHz by observing the same field for 1000 hours, will be able to reach its target depth in the presence of these artefacts.

We also present **stimela**, a system agnostic scripting framework for simulating, processing and imaging radio interferometric data. This framework is then used to write an end-to-end simulation pipeline in order to quantify the resolution and sensitivity of the SKA1-MID telescope (the first phase of the SKA mid-frequency telescope) as a function of frequency, as well as the scale-dependent sensitivity of the telescope. Finally, a **stimela**-based reduction pipeline is used to process data of the field around the source 3C147, taken by the Karl G. Jansky Very Large Array (VLA). The reconstructed image from this reduction has a typical 1σ noise level of $2.87 \mu\text{Jy}/\text{beam}$, and consequently a dynamic range of $8 \times 10^6:1$, given the $22.58 \text{ Jy}/\text{beam}$ flux Density of the source 3C147.

Contents

1	Introduction	7
1.1	The Purpose of this Work	8
1.2	Document Outline	9
2	Radio Interferometry Fundamentals	11
2.1	The Response of an Interferometer	11
2.2	Radio Synthesis Imaging	14
2.2.1	Visibility weighting	17
2.3	Sky Brightness Reconstruction	19
2.4	The Radio Interferometer Measurement Equation	22
2.4.1	Signal propagation	22
2.5	Calibration	25
2.6	Self-calibration	25
2.6.1	Calibration Artefacts	25
3	Forecasting Calibration Limitations of the MeerKAT Telescope	27
3.1	The MeerKAT primary beam	29
3.2	The experiment	29
3.2.1	The strategy	33
3.2.2	Telescope simulations	34
3.2.3	The sky model	34
3.2.4	Data calibration	34
3.2.5	Results and discussions	36
3.3	Conclusions	49
4	Platform Independent Scripting	51
4.1	Radio Astronomy Software	51
4.1.1	Radio Astro PPA (<i>now Kern Suite</i>)	52
4.2	Container Technology	52
4.3	Docker	53
4.3.1	Architecture	54
4.4	Stimela	55
4.4.1	Architecture	56

4.4.2	Python backend	58
4.4.3	User interface	59
4.4.4	Reproducibility	61
4.5	Conclusions	62
5	SKA1-MID Sensitivity and Imaging Performance	63
5.1	The SKA1-MID antenna layout	63
5.1.1	A case for a decreasing the SKA1-MID maximum baseline . .	64
5.2	SKA1-MID performance assessment	66
5.2.1	Resolution and sensitivity	68
5.2.2	SKA1-MID scale-dependent sensitivity	72
5.3	Discussion	74
5.3.1	Telescope design in the modern era	74
5.3.2	Conclusions	80
6	A Platform-Independent Data Reduction Script	81
6.1	Reduction strategy	81
6.1.1	Data products	96
6.2	Discussion	97
6.3	Conclusions	100
7	Conclusions	103
7.1	Future Work	104
	Appendices	107
A	Stimela Example Script	109
A.1	Stimela Data Synthesis and Imaging Recipe	109

List of Figures

2.1	Two-element interferometer	11
2.2	The uv -coverage from a 2 hour VLA (C configuration) observation. The uv -coverage of an interferometer observation is given by the uv -points (blue) as well as the complex conjugates (red).	18
2.3	PSF image (natural weighting) from a 2 hour VLA (C configuration) observation at L-Band.	19
2.4	PSF image (uniform weighting) from a 2 hour VLA (C configuration) observation at L-Band.	20
2.5	Two-element interferometer.	22
3.1	Schematics of the MeerKAT antenna.	28
3.2	MeerKAT antenna layout and uv -coverage	29
3.3	Image plane noise as a function of time for the 3 MeerKAT bands. . .	30
3.4	The MeerKAT primary beam pattern, along with cross-sections along $x = 0$ (top) and $y = 0$ (right).	31
3.5	Amplitudes of the Jones matrix components of the MeerKAT primary beam (voltage beams).	32
3.6	The sky model used in the simulations superimposed on the MeerKAT primary beam. The white points indicate the directions in which direction-dependent solutions were applied.	35
3.7	Corrected residual images from reductions with and without directions dependent calibration.	37
3.8	Distilled residual images from reductions with and without directions dependent calibration.	38
3.9	Accumulation of pointing errors from simulated MeerKAT observations with a pointing accuracy of $15''$	40
3.10	Accumulation of pointing errors from simulated MeerKAT observations with a pointing accuracy of $25''$	41
3.11	Corrected residual images at different stages of the accumulation. These images are from a simulation with a pointing accuracy of $15''$	42
3.12	Corrected residual images at different stages of the accumulation. These images are from a simulation with a pointing accuracy of $25''$	43

3.13	Distilled residual images at different stages of the accumulation. These images are from a simulation with a pointing accuracy of 15".	44
3.14	Distilled residual images at different stages of the accumulation. These are images are from a simulation with a pointing accuracy of 25". . .	45
3.15	Cumulated corrected residual image after a 2056 hour simulated observation with MeerKAT with a pointing accuracy of 15".	45
3.16	Accumulation of pointing errors from simulated MeerKAT observations with a pointing accuracy of 15" in a 72 kHz channel.	46
3.17	Accumulation of pointing errors from simulated MeerKAT observations with a pointing accuracy of 25" in a 72 kHz channel.	47
4.1	Schematic of the difference in architectures between containers and virtual machines.	54
4.2	Dockerfile (Docker image recipe) for the MeqTrees software package. .	55
4.3	Schematic of the sitmela architecture.	56
4.4	Structure of a stimela executor container.	57
4.5	Stimela executor image template for the MeqTrees based calibration task.	57
5.1	The SKA1-MID antenna layout by the SKA organisation (red crosses), which has a maximum baseline of 157 km and our alternative layout (SKA1MID-137), which has a maximum baseline of 137 km.	66
5.2	The SKA1-MID antenna layout by the SKA organisation (SKA1MID; red crosses), which has a maximum baseline of 157 km and our alternative layout (SKA1MID-120; blue dots), which has a maximum baseline of 120 km.	67
5.3	Histograms of the baseline distribution of from the three SKA1-MID antenna layouts under consideration.	67
5.4	<i>uv</i> -Coverage plots from 2-hour simulated observations with the SKA1-MID telescope using the (left to right) SKA1MID, SKA1MID-137, and the SKA1MID-120 antenna layouts.	68
5.5	PSF cross-sections from simulated SKA1-MID observations at 700 MHz based on the three layouts under consideration.	69
5.6	PSF size as function a function of frequency for the three antenna layouts under consideration.	70
5.7	Image plane noise as a function of frequency for the three antenna layouts under consideration.	71
5.8	Illustration of the stimela executor image that applies visibility weights that select specific angular scales.	73
5.9	Customised weights applied to select visibility points corresponding to certain angular scales	74
5.10	Image plane sensitivity as function of angular scale for band one of the SKA1-MID telescope.	75

5.11	Image plane sensitivity as function of angular scale for band two of the SKA1-MID telescope.	76
5.12	Image plane sensitivity as function of angular scale for band three of the SKA1-MID telescope.	77
5.13	Image plane sensitivity as function of angular scale for band four of the SKA1-MID telescope.	78
5.14	Image plane sensitivity as function of angular scale for band five of the SKA1-MID telescope.	79
6.1	Amplitude of the Jones matrix of the VLA primary beam pattern at L-band	83
6.2	DDFacet <code>stimela</code> base image template (Dockerfile).	85
6.3	Corrected residual image after a single calibration step. The data being reduced is a VLA observation of the field around the source 3C147.	86
6.4	Primary beam corrected image of corrected residual data after a single calibration step. The data being reduced is a VLA observation of the field around the source 3C147.	87
6.5	Corrected residual image after 2 calibration steps. The data being reduced is a VLA observation of the field around the source 3C147.	89
6.6	Reconstructed image from a reduction a VLA observation of the field around 3C147 (low band).	90
6.7	Corrected residual image after 3 calibration steps. The data being reduced is a VLA observation of the field around the source 3C147.	91
6.8	Central 13 arcminutes of corrected residual image after 4 calibration steps. The data being reduced is a VLA observation of the field around the source 3C147.	92
6.9	Corrected residual image after 4 calibration steps. The data being reduced is a VLA observation of the field around the source 3C147.	94
6.10	Central 13 arcminutes of corrected residual image after 4 calibration steps. The data being reduced is a VLA observation of the field around the source 3C147.	95
6.11	Reconstructed image from our reduction of a VLA observation of the field around 3C147.	96
6.12	Reconstructed image from a reduction a VLA observation of the field around 3C147.	97
6.13	Central 1 degree of the reconstructed high-resolution image of the field around the source 3C147 at L-band.	99
6.14	Central 1 degree of reconstructed low-resolution image of the field around the source 3C147 at L-band.	100
A.1	contents of the output directory after the example script was executed.	111

List of Tables

3.1	MeerKAT frequency band information and corresponding SEFD values.	28
3.2	Summary of data sizes from the simulations in this chapter.	48
5.1	SKA1-MID frequency band information and corresponding SEFD values (at the time of writing) [3].	64
5.2	SKA1-MID science goals with the strictest angular resolution requirements.	65
5.3	SKA1-MID maximum resolution (λ/B_{max}) at the five band centres. .	72
6.1	Information about the VLA datasets to be reduced.	82
6.2	Top 10 components extracted from our reconstructed high-resolution image.	98
6.3	Top 10 components extracted from our reconstructed low-resolution image.	98

Declaration

I certify that the work presented in this document is my own, that it has not been accepted for the award of any other degree in my name, or any other name, at any other university or tertiary institution. To the best of my knowledge it contains no material that has previously been published, or has been written by any other person, except where explicitly stated, and due reference has been made. I give consent to the Rhodes University to deposit this thesis in the university library, as well as to make it available on the world wide web.

Acknowledgements

I would like to express my gratitude to the following people and institutions, without whom this work would not have been possible.

My supervisors, Oleg Smirnov and Ian Heywood for their unwavering support and guidance. Working under the esteemed guidance of these brilliant scientists has been the privilege of my life.

In the same breath, I would like to thank Ronel Groenewald, the administrator for our research group, for her assistance with administrative and logistical matters, particularly with matters related to both domestic and international travel.

I would also like to thank my family for their support and encouragement throughout my education, in particular my mother Zanele Xaba and grandmother Clara Xaba, who have sacrificed much to ensure that I received a good education.

I would also like to thank the National Research Foundation, and by extension the South African tax payer, who have funded my post-graduate education. Last, but not least, I also extend my gratitude to the South African centre for high performance computing (CHPC) for providing me with much needed computing resources, as well as the SKA organisation for the 23 000 US dollars' worth of computing credits on the Amazon Web Services platform awarded to me through the *SKA/AWS Astro-Compute in the Cloud* initiative.

Nomenclature

Abbreviations

AWS Amazon Web Services

CHPC South African Centre for High Performance Computing

CMB Cosmic Microwave Background

DD Direction Dependent (Telescope or atmospheric effects)

DI Direction Independent (Telescope or atmospheric effects)

FAST Five hundred metre Aperture Spherical Telescope

FFT Fast Fourier Transform

FoV Field of View

FWHM Full Width at Half Maximum

GBT Green Bank Telescope

GCE Google Compute Engine

GMRT Giant Metrewave Radio Telescope

HERA Hydrogen Epoch of Reionization Array

LADUMA Looking at the Distant Universe with MeerKAT Array

LOFAR Low-Frequency Array

MWA Murchison Widefield Array

PSF Point Spread Function

RIME Radio Interferometer Measurement Equation

RMS Root Mean Square

SEFD System Equivalent Flux Density

SKA Square Kilometre Array

SKA1-MID First phase of SKA mid-frequency telescope

VLA Karl G. Jansky Very Long Array

WSRT Westerbork Radio Synthesis Telescope

Constants

i Imaginary unit; $i = \sqrt{-1}$

k_b Boltzmann's constant; $1.38064852 \times 10^{-23} \text{ m}^2 \text{ kg s}^{-2} \text{ K}^{-1}$

π Ratio of a circle's circumference to its diameter; $\pi = 3.14195\dots$

Frequently used symbols

b Baseline vector

G Differential gain Jones matrix

E Jones matrix representing direction-dependent effects

e Electric field

ν Frequency

G Jones matrix representing direction-independent effects

I Sky brightness

J Jones matrix

l, m, n Directional cosines

u, v, w Visibility plane coordinates. Sometimes referred to as simply uv .

s Unit vector pointing towards the direction of a source of radiation

t Time

V Visibility matrix

v Voltage

$d\Omega$ Element of solid angle

\mathbf{X} Sky coherency matrix

Chapter 1

Introduction

Since its accidental inception in the 1930s by Karl G. Jansky [32], the field of radio astronomy has played an important role in advancing our knowledge of our solar system, our Galaxy and the Universe. Prime among these, are the discovery of the 21 cm neutral hydrogen line which has given us great insights into the formation and evolution of galaxies [44], the discovery of pulsars which have proven to be invaluable to our understanding of gravity and stellar evolution [25], and the seminal discovery of the cosmic microwave background (CMB) radiation [49].

In 1937 Grote Reber used a 9 m telescope that he built in his backyard to produce the first map of the radio sky [52], paving the way for a multitude of technological and scientific discoveries. Given the large wavelengths of radio waves however, obtaining high angular resolution requires very large antennas, and accordingly we have seen the 300 m Arecibo telescope in Puerto Rico built in 1963, the 64 m Australian Parkes telescope in 1962, to the 100 m Green Bank telescope (GBT) in the USA which was built in 2000, and in 2016, the Five hundred metre Aperture Spherical Telescope (FAST) was officially launched in China. Building the infrastructure to support these large dishes is extremely difficult, in fact, the two largest of these are only possible due to natural depressions of the landscape on which they are constructed.

To study the fine spatial details of the radio sky we need a more manageable solution, such as interferometry [55]. From the first two-element radio interferometer by Ryle and Vondberg in the 1940s [55], interferometry techniques have played an integral role in the field of radio astronomy since the angular resolution of an interferometer is proportional to the distance that separates the pair of antennas that are the furthest apart (baseline). These radio interferometers, such as the Westerbork Radio Synthesis Telescope (WSRT), the Jansky Very Large Array (VLA) telescope and the Giant Metrewave Radio Telescope (GMRT) have been at the forefront in the field of radio astronomy for decades. This success of radio interferometry has led to the Square Kilometre Array (SKA) project¹, an international initiative to build a radio telescope with a collecting area of a square kilometre. The project will be hosted

¹<https://www.skatelescope.org/ska-organisation>

by South Africa and Australia. With its high sensitivity and angular resolution, the SKA aims to answer fundamental questions about the nature of gravity in extreme conditions [58], the origin of cosmic magnetism [33], as well as detecting and studying the neutral hydrogen signature at the epoch of re-ionization [36], and whether other forms of biologies exist in our Universe [27]. The SKA, along with its precursor instruments, MeerKAT and HERA (Hydrogen Epoch of Reionization Array) in South Africa and ASKAP (Australian SKA Pathfinder) and MWA (Murchison Widefield Array) in Australia, as well as the FAST and the LOFAR (Low-Frequency Array) in Europe, places radio astronomy at the forefront of modern scientific discovery and technological advancement.

It is worth noting that along with the development of the radio interferometry technique and the increased capacity and complexity over the decades, algorithms and software packages that process, visualise and analyse these data have also played a pivotal role in the progress in the field. Radio synthesis imaging and calibration tools such as Högbom’s `clean` algorithm [28], Cornwell’s Maximum Entropy Method [14, MEM] as well as his *self-calibration* algorithm [12], led to end-to-end data reduction packages such as **AIPS** and **CASA**, which have been the prime packages in radio synthesis imaging and calibration. However, as radio instruments have grown more complex so have these packages. Unlike the instruments however, which are built by professional engineers, radio interferometry software is often written by astronomers and not by software engineers, leading to unstable software applications that are often difficult to install. Unfortunately, these issues are not limited to legacy software packages, as even more recent packages that are widely used, such as **MeqTrees**, newer versions of **CASA**, as well a whole host of novel packages have similar shortcomings. Nevertheless, as we will also show in this work, these packages are sophisticated enough that a select few suitably skilled individuals can produce simulations to address many of the imaging and calibration issues that will come with this new generation of telescopes [20, 26, 40], as well as produce quality reductions from modern instruments [21, 26, 42, 61]. However, given the large data rates that are expected with these instruments, it is imperative that these reductions and simulations become more routine; that is, the usage of theses tools (especially in combinations) becomes easier for the general astronomer.

1.1 The Purpose of this Work

In this work we show how legacy and novel radio astronomy software packages, coupled with computing platforms such as **Docker** can be used to develop platform agnostic data synthesis and reduction frameworks that require minimal effort to install. **Docker**² uses container technology to wrap up software applications in light-weight complete and isolated filesystems that contain all the application dependencies (code,

²<https://www.docker.com>

libraries, system tools, etc.). This means that each radio interferometry package can be installed and executed from an isolated environment (equipped with a custom operating system and system settings), thus avoiding many of the issues discussed above. Moreover, this container technology has been adopted by cloud computing platforms such as the Amazon Web Services³ (AWS) and the Google Compute Engine⁴ (GCE), which opens up possibilities for large-scale distributed data reduction and simulation pipelines.

We will also present a system agnostic framework that combines both legacy and novel software packages, and show how it can be used to produce SKA-scale end-to-end simulations as well as quality wide-band and wide-field data processing and imaging of radio data.

1.2 Document Outline

We start with an overview of the fundamentals of radio interferometry synthesis imaging and calibration. The purpose of this overview is to introduce some basic concepts and nomenclature, as well as to set the stage for the work that follows. In Chapter 3 we present a case study of how direction-dependent effects will affect deep MeerKAT surveys. Chapter 4 introduces `stimela`, a framework for flexible and platform-independent radio data synthesis and data reductions. The framework is then used in Chapter 5 to gauge the sensitivity and resolution of the scale-dependent SKA1-MID sensitivity of the SKA1-MID telescope, and in Chapter 6 the same framework is used to reduce a VLA observation of the field around the source 3C147. Finally, we present our concluding remarks in Chapter 7.

³<https://aws.amazon.com>

⁴<https://cloud.google.com/compute>

Chapter 2

Radio Interferometry Fundamentals

In this chapter we discuss the fundamentals of radio interferometric synthesis imaging, modelling and calibration.

2.1 The Response of an Interferometer

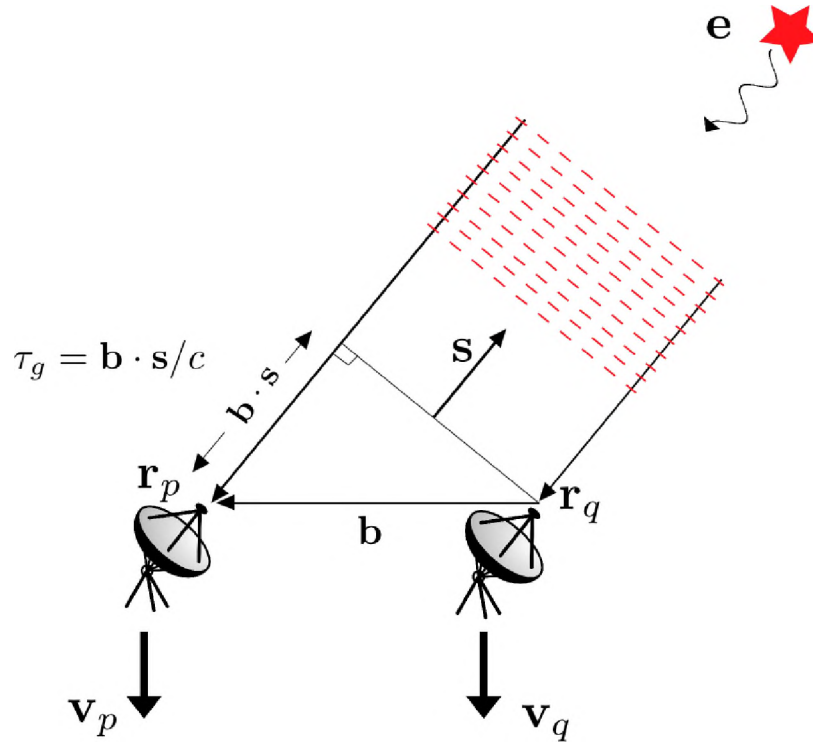


Figure 2.1: Two-element interferometer

The derivations in this section follow [10, 68]. Consider the two-element interferometer in Figure 2.1 which is made up of two identical antennas p and q at positions \mathbf{r}_p and \mathbf{r}_q respectively. A radiation field, \mathbf{e} , emitted from direction \mathbf{R} by a source in the far field is incident on the two antennas separated by a separation vector $\mathbf{r}_p - \mathbf{r}_q = \mathbf{b}$ (baseline vector). Since the source is in the far field, the incident radiation can be considered as a plane wave which arrives at \mathbf{r}_q at time t , and then arrives at \mathbf{r}_p at time $t + \tau_g$. The radiation at antenna p arrives later because it travels an extra path length of $\mathbf{b} \cdot \mathbf{s}$ as shown in the diagram. The resulting delay, often referred to as the geometric delay, is

$$\tau_g = \mathbf{b} \cdot \mathbf{s} / c, \quad (2.1)$$

where $c = 2.998 \times 10^8 \text{ m s}^{-1}$ is the speed of light in vacuum, and $\mathbf{s} = \mathbf{R}/|\mathbf{R}|$ is a unit vector pointing towards the source. The intensity, $I(\mathbf{s}, t, \nu)$, also known as the surface brightness of the source, is simply the flux passing through a unit solid angle, $d\Omega$ at time t and frequency ν , and is measured in Jy, with

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

The interferometer measures the degree of correlation between measurements at \mathbf{r}_p and \mathbf{r}_q of the radiation field, $\mathbf{e}(\mathbf{R}, t, \nu)$. This correlation, also known as the spatial coherence of the source intensity is given by [10]

$$V(\mathbf{r}_p, \mathbf{r}_q, t, \nu) = \int I(\mathbf{s}, t, \nu) e^{-2\pi i \nu \mathbf{s} \cdot (\mathbf{r}_p - \mathbf{r}_q) / c} d\Omega, \quad (2.2)$$

Note that the measurement of the spatial coherence only depends on the difference vector $\mathbf{r}_p - \mathbf{r}_q$. For convenience, but without loss of generality, the polarization properties of the source will not be considered in this section. The voltage responses from the antennas are passed through amplifiers and filters that select the required frequency band of width $\Delta\nu$ centred on ν . The signals are combined in a device called the correlator, which multiplies and averages them. The correlator receives signals $v_p(t)$ and $v_q(t)$, its output is proportional to

$$\langle v_p(t) v_q(t) \rangle,$$

where $\langle \cdot \rangle$ represents a time average. If the signals are quasi-monochromatic then [10]

$$v_p(t) = v_p \cos(2\pi\nu(t - \tau_g)) \quad (2.3)$$

$$v_q(t) = v_q \cos(2\pi\nu t), \quad (2.4)$$

where $2\pi\tau_g$ is a phase term that accounts for the geometric delay. The trigonometric identity

$$\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b), \quad (2.5)$$

leads to

$$\begin{aligned} r(\tau_g) &= v_p(t)v_q(t) \\ &= v_p v_q \cos(2\pi\nu\tau_g). \end{aligned} \quad (2.6)$$

If $A(\mathbf{s})$ is the effective radiation pattern of the antenna in the direction \mathbf{s} (assumed to be the same for each antenna), then the signal power received in bandwidth $\Delta\nu$ from the solid angle element $d\Omega$ is $A(\mathbf{s})I(\mathbf{s})\Delta\nu d\Omega$. The correlator signal per solid angle element is then given by

$$dr = A(\mathbf{s})I(\mathbf{s})\Delta\nu d\Omega \cos(2\pi\nu\tau_g). \quad (2.7)$$

Note that even though the time and frequency dependence has not been explicitly stated in (2.7), A and I are functions of both time and frequency. Substituting (2.1), the signal power over the celestial sphere is

$$r = \int_S A(\mathbf{s})I(\mathbf{s})\Delta\nu \cos(2\pi\nu \frac{\mathbf{b} \cdot \mathbf{s}}{c}) d\Omega, \quad (2.8)$$

where \int_S is a surface integral over the celestial sphere. It is often convenient to specify a position on which the synthesized field of view (FoV) is to be centred. This is the so-called *phase tracking centre*, \mathbf{s}_0 , given by

$$\mathbf{s} = \mathbf{s}_0 + \sigma, \quad (2.9)$$

where, σ is the position w.r.t the phase tracking centre.

Using (2.5), the signal response can now be re-written as

$$\begin{aligned} r &= \Delta\nu \cos\left(\frac{2\pi\nu\mathbf{b} \cdot \mathbf{s}_0}{c}\right) \int_S A(\sigma)I(\sigma) \cos\left(\frac{2\pi\nu\mathbf{b} \cdot \sigma}{c}\right) d\Omega \\ &- \Delta\nu \sin\left(\frac{2\pi\nu\mathbf{b} \cdot \mathbf{s}_0}{c}\right) \int_S A(\sigma)I(\sigma) \sin\left(\frac{2\pi\nu\mathbf{b} \cdot \sigma}{c}\right) d\Omega. \end{aligned} \quad (2.10)$$

Defining the complex visibility, \mathbf{V} , of the interferometer as [10]

$$V \equiv |V|e^{i\phi_V} = \int A'(\sigma)I(\sigma)e^{2\pi i\nu\mathbf{b} \cdot \sigma/c} d\Omega, \quad (2.11)$$

where ϕ_V is the phase of the visibility, $A'(\sigma) \equiv A(\sigma)/A_0$, is the normalised antenna radiation pattern, also known as the *primary beam pattern*. A_0 is the antenna response at the beam centre. Separating the real and imaginary parts of (2.11) yields

$$A_0|V|\cos(\phi_V) = \int_S A(\sigma)I(\sigma)\cos\left(\frac{2\pi\nu\mathbf{b}\cdot\sigma}{c}\right)d\Omega \quad (2.12)$$

$$A_0|V|\sin(\phi_V) = \int_S A(\sigma)I(\sigma)\sin\left(\frac{2\pi\nu\mathbf{b}\cdot\sigma}{c}\right)d\Omega. \quad (2.13)$$

Substituting (2.12) and (2.13) into (2.10) gives

$$r = A_0\Delta\nu|V|\cos\left(\frac{-2\pi\nu\mathbf{b}\cdot\mathbf{s}_0}{c} - \phi_V\right). \quad (2.14)$$

In practice, to make use of (2.14) one has to introduce a more convenient coordinate system. First, let the baseline vector be specified in the u, v, w system, where, u points East, v points North and w points towards the phase tracking centre (normal to the plane that contains u and v), and all are measured in wavelengths. Second, let the sky positions be specified in the l, m, n coordinate system, where l, m, n are direction cosines measured with respect to the u, v, w axes. These coordinate systems are related via

$$\frac{\nu\mathbf{b}\cdot\mathbf{s}}{c} \equiv ul + vm + wn \quad (2.15)$$

$$\frac{\nu\mathbf{b}\cdot\mathbf{s}_0}{c} \equiv w \quad (2.16)$$

$$d\Omega \equiv \frac{dl\,dm}{n} = \frac{dl\,dm}{\sqrt{1-l^2-m^2}}. \quad (2.17)$$

The visibility equation (2.11) then becomes

$$V(u, v, w) = \iint A'(l, m)I(l, m)e^{-2\pi i[ul+vm+w(\sqrt{1-l^2-m^2}-1)]}\frac{dl\,dm}{\sqrt{1-l^2-m^2}}. \quad (2.18)$$

This is a fundamental equation in radio interferometry as it relates the measured sky coherency, $V(u, v, w)$, to the sky brightness $I(\mathbf{s})$. We will show in the next section that this equation reduces to a two-dimensional Fourier transform.

2.2 Radio Synthesis Imaging

Radio synthesis imaging involves inverting (2.18) to get a synthesized image

$$\frac{A'(l, m)I(l, m)}{\sqrt{1-l^2-m^2}} = \iiint V(u, v, w)e^{2\pi i[ul+vm+w(\sqrt{1-l^2-m^2}-1)]}du\,dv\,dw. \quad (2.19)$$

Note that this synthesized image in the lm plane represents a projection of the celestial sphere onto a plane. Equation 2.18 reduces to a Fourier transform relation between the modified sky brightness $A'(l, m)I(l, m)$ and the visibility $V(u, v, w)$ if the term $w(\sqrt{1 - l^2 - m^2} - 1)$, in the exponent, equals zero. Besides the trivial case $l = m = 0$, this is possible if:

1. $|l|$ and $|m|$ are sufficiently small that¹.

$$w(\sqrt{1 - l^2 - m^2} - 1) \simeq -\frac{1}{2}(l^2 + m^2)w \simeq 0. \quad (2.20)$$

In this case (narrow field imaging), the dependence of the visibility on w is negligible.

2. $w = 0$. That is, all baseline vectors lie on a common plane. An example is an interferometer where all the antennas are confined to small enough area such that all baselines lie on the same plane. An example is the KAT-7 telescope² when using very short integrations (snapshots). Another example is Westerbork telescope, which has all its antennas in an East-West line on the Earth's surface, meaning that all the components of the baseline vectors parallel to the Earth's axis are zero, therefore the baseline vectors all lie on the same plane.

In either of the above cases, the visibility can be written as

$$V(u, v) = \int \int A'(l, m)I(l, m)e^{-2\pi i(ul+vm)} dl dm. \quad (2.21)$$

In the $w = 0$ case, the $1/\sqrt{1 - l^2 - m^2}$ dependence is absorbed into $I(l, m)$. Recall that the Fourier transform, $\mathcal{F}\{\}$, of a function $f(x, y)$ is given by

$$\mathcal{F}\{f(x, y)\}(k_1, k_2) = \int \int f(x, y)e^{-2\pi i(xk_1+yk_2)} dx dy$$

Note that (in this case) the visibility is the two-dimensional Fourier transform of the sky brightness. Inverting this equation, the synthesized image is given by

$$A'(l, m)I(l, m) = \int \int V(u, v)e^{2\pi i(ul+vm)} dudv \quad (2.22)$$

$$= \mathcal{F}^{-1}\{V(u, v)\}, \quad (2.23)$$

where $\mathcal{F}^{-1}\{\}$ denotes an inverse Fourier transform. For the remainder of this section, we will assume that the normalized antenna response is unity, that is, the

¹Using the Taylor expansion $\sqrt{1+x} \simeq 1+x/2$, therefore $\sqrt{1-l^2-m^2}-1 \simeq -\frac{1}{2}(l^2+m^2)$

²<https://www.ska.ac.za/science-engineering/kat-7>

modified sky brightness $A'(l, m)I(l, m) = I(l, m)$. To summarise, an interferometer measures the spatial coherence of the sky brightness. For small fields of view, or coplanar baselines, this spatial coherence is given by the visibility equation (2.21), which can be inverted to get an image of the sky brightness via (2.22). Interferometers also have a finite number of antennas, and integration times are finite; during the course of an observation the baselines rotate as the earth rotates, sampling progressively more of the spatial coherence of the sky. Therefore, visibility function is sampled at a finite number of points. It can consequently be written as

$$V'(u, v) = V(u, v)S(u, v) \quad (2.24)$$

$$= \sum_k \delta(u - u_k, v - v_k)V(u_k, v_k), \quad (2.25)$$

$$(2.26)$$

where $S(u, v)$ is the sampling function

$$S(u, v) = \sum_k \delta(u - u_k, v - v_k). \quad (2.27)$$

The synthesized image is then ,

$$I' = \mathcal{F}^{-1}\{V'(u, v)\} \quad (2.28)$$

$$= \mathcal{F}^{-1}\{V(u, v)S(u, v)\}. \quad (2.29)$$

The image represented by (2.29) is the so-called *dirty image*. Using the convolution theorem, (2.29) can be expressed as

$$I' = \mathcal{F}^{-1}\{S(u, v)\} * \mathcal{F}^{-1}\{V(u, v)\}, \quad (2.30)$$

where $*$ denotes a convolution. Note that the Fourier transform of the sampling function is the response of the interferometer to a point source, also known as the *point spread function* (PSF). The PSF, $\mathcal{P} = \mathcal{F}^{-1}\{S(u, v)\}$, is often referred to as the *dirty beam*. Therefore,

$$I' = \mathcal{P}(l, m) * I(l, m). \quad (2.31)$$

Note that the units of I' are flux density per PSF area, also known as Jy/beam.

To summarise, due to the incomplete sampling of the spatial coherence of the sky brightness, the synthesized image from an interferometer is the convolution of the PSF and the sky brightness.

2.2.1 Visibility weighting

To maximise the sensitivity of an interferometer, each visibility point is weighted by the inverse of the variance, σ^2 , of the thermal noise (assumed to be Gaussian) contribution, where σ is the standard deviation of this noise. The weight for k -th visibility point is given by

$$W_k = \frac{1}{\sigma_k^2}. \quad (2.32)$$

This weighting scheme is called *natural weighting* as it gives the natural sensitivity of the interferometer. However, visibility points tend to be distributed more densely near the (u, v) origin, and this non-uniform uv density (see Figure 2.2) leads to a PSF with broad low-level wings, as can be seen in Figure 2.3. These broad wings smear out small scale structure in the image domain, which may be undesirable for science cases that require high angular resolution. To combat this biased visibility density, one could give a higher weight to visibilities further away from the (u, v) origin. This weighting, which seeks to have a uniform visibility density is called *uniform weighting*, and the weights are given by [6]

$$W_k = \frac{\alpha_k}{N_i(k)}, \quad (2.33)$$

where, α_k is some real constant, $N_s(k)$ is the number of visibility points that fall within a symmetric region, of some characteristic width, of the uv -plane centred on the k^{th} visibility point. In practice however, (2.21) is computed via a Fast Fourier Transform (FFT) which requires the visibility points to be placed in a regular grid, i.e., visibility points are placed into regular bins with dimensions that are reciprocals of the target image dimensions. Hence, $N_i(k)$ is commonly defined to be the number of visibility points that fall in a given gridded uv -bin, i . A common choice for α_k is $1/\sigma_k^2$. Since the uv -coverage tends to be more dense on shorter spacings, uniform weighting effectively downweights visibilities closer to the centre, placing more emphasis on visibilities further out (or visibilities corresponding to baselines which are more sensitive to small angular scales). However, the consequence of down-weighting visibilities is a reduced sensitivity, and one has to choose the weighting that best suits their science. Accounting for visibility weights, (2.31) becomes

$$S(u, v) = \sum_k W_k \delta(u - u_k, v - v_k). \quad (2.34)$$

A more general visibility weighting scheme, which minimizes sidelobe power and thermal noise, was proposed by Briggs [5]. Here, the weight for the k -th visibility is given by

$$W_k = \frac{\alpha}{1 + N_i(k)f^2}, \quad (2.35)$$

where,

$$f = \frac{(5^{-r})^2}{\frac{\sum_i N_i(k)}{\sum_k \alpha_k}}. \quad (2.36)$$

$r \in \mathbb{R}$ is the so-called *robustness* parameter. $r = 0$ gives a good balance between resolution and sensitivity, while $r \rightarrow \infty$ reduces to the natural weighting case, and $r \rightarrow -\infty$ reduces to the uniform weighting case. In practice, r is chosen to be between -2 and 2 since these values are sufficiently far away from zero that they effectively lead to uniform and natural weighting, respectively. Recently, a weighting scheme that tries to maximise the sensitivity for a given resolution has been proposed by Yatawatta [72]. At the time of writing we are not aware of any imaging package that uses this weighting scheme.

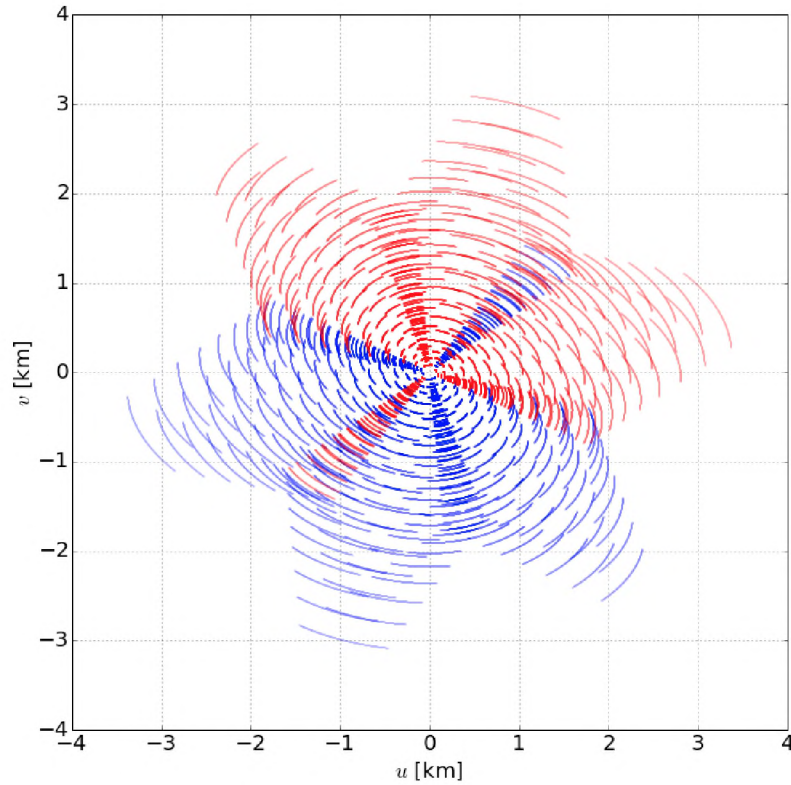


Figure 2.2: The uv -coverage from a 2 hour VLA (C configuration) observation. The uv -coverage of an interferometer observation is given by the uv -points (blue) as well as the complex conjugates (red).

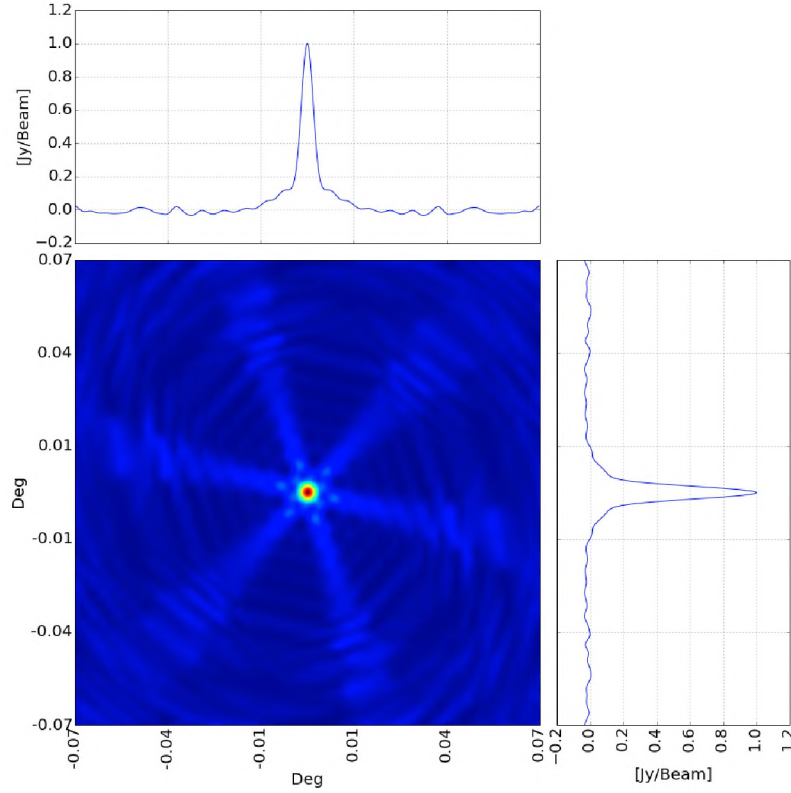


Figure 2.3: PSF image (natural weighting) from a 2 hour VLA (C configuration) observation at L-Band.

2.3 Sky Brightness Reconstruction

As intimated in Section 2.2.1, visibility measurements can be corrupted by a variety of instrumental and propagation effects, which will be discussed in detail in Section 2.4. In this section, we focus on the thermal noise contribution from the antenna receivers. Using the radiometer equation we can estimate the standard deviation of the thermal noise per visibility (real and imaginary) as [10] :

$$\sigma_{\text{vis}} = \frac{\text{SEFD}}{\sqrt{2\Delta\nu\Delta t}}, \quad (2.37)$$

where $\Delta\nu$ (in Hz), Δt (in seconds) are the frequency channel width and integration time, respectively. The SEFD (in Jy) is the so-called *System equivalent flux density*, which can be interpreted as the flux density equivalent of the receiver system

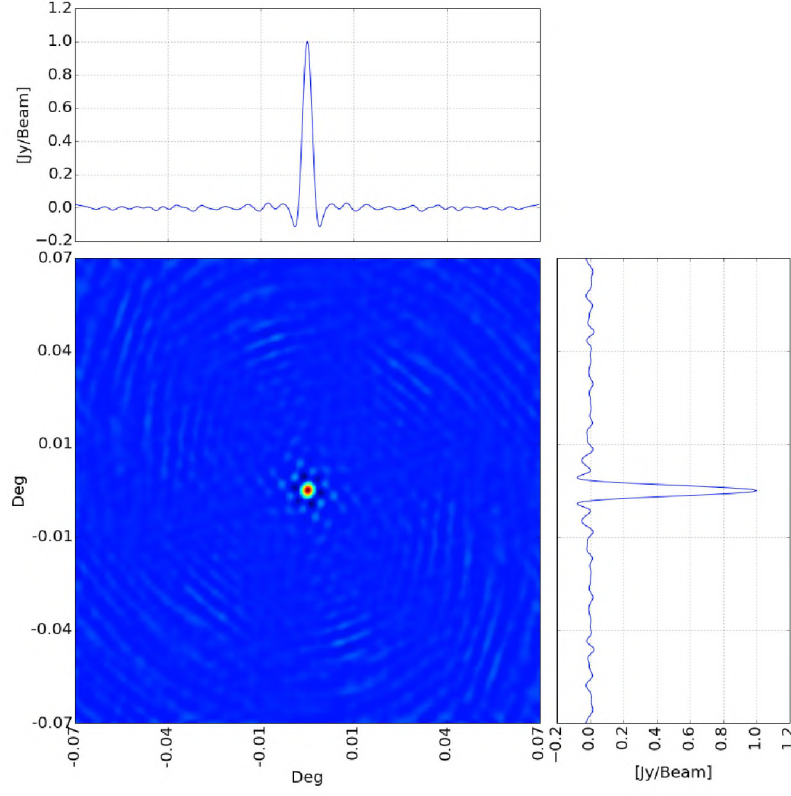


Figure 2.4: PSF image (uniform weighting) from a 2 hour VLA (C configuration) observation at L-Band.

temperature, T_{sys} and is given by

$$\text{SEFD} = \frac{2k_b T_{\text{sys}}}{A_e}, \quad (2.38)$$

where k_b is Boltzmann's constant ($1.3806 \times 10^{-23} \text{ m}^2 \text{ kg s}^{-2} \text{ K}^{-1}$) and A_e is the effective area of the antenna (in m^2). T_{sys} is in Kelvin. Note that σ_{vis} is independent for different visibilities. The “corrupted” visibility then becomes

$$V'(u, v) = V(u, v) + n_{\text{vis}}(u, v), \quad (2.39)$$

with n_{vis} being the thermal noise corruption with standard deviation σ_{vis} and mean of zero. The dirty image then becomes

$$\begin{aligned}
I'' &= \mathcal{F}^{-1}\{S(u, v)\} * (\mathcal{F}^{-1}\{V(u, v) + n(u, v)\}) \\
&= \mathcal{P} * [I(l, m) + \mathcal{F}^{-1}\{n_{\text{vis}}(u, v)\}] \\
&= \mathcal{P} * [I(l, m) + n_{\text{im}}(l, m)] \\
&= \mathcal{P} * I + \mathcal{P} * n_{\text{im}} \\
&= I' + n'_{\text{im}}.
\end{aligned} \tag{2.40}$$

The thermal noise in the image plane, n_{im} is the Fourier transform of n_{vis} . Then considering that the visibility function does not sample the entire uv -plane, the noise in the dirty image is the thermal noise in the image plane convolved with the PSF. That is, the noise in a dirty image is correlated on scales corresponding to the size (full width at half maximum; FWHM) of the PSF.

Since the measured visibilities do not completely sample the spatial coherence of the sky brightness, reconstructing I from (2.40) is an ill-posed inverse problem with an infinite number of solutions. The solution would involve estimating the coherency of the sky brightness at points where measurements were not taken. Stated another way, reconstructing the sky brightness involves deconvolving out the PSF from the dirty image I' (this process will be referred to as *deconvolution* from this point onwards) in the presence of a correlated noise signal. Note that this process cannot reconstruct the sky brightness with complete accuracy, but can produce an approximation of the sky brightness under some assumptions.

The most widely-used deconvolution algorithm is derived from Hogbom's **clean** algorithm [28], which is a matching pursuit algorithm [1] that successively subtracts the PSF from peaks in the dirty image. Implicit in the **clean** algorithm is the assumption that the sky brightness can be approximated by a weighted superposition of delta functions. Another approach is the *maximum entropy* (MEM) algorithm [14], which uses regularization of the dirty image through the optimization of the configurational entropy of the image.

More recently, a multi-scale variant of the *clean* algorithm was introduced [13][*MS-clean*], which unlike the standard *clean* algorithm assumes the sky is composed of sources of various size scales. Moreover, with the introduction of *compressed sensing* (CS) and Bayesian techniques into radio synthesis imaging there has been a set of new deconvolution algorithms, most notably [17, MORESANE], [9, PURIFY] and [22, SASIR] on the CS side, and [35, RESOLVE] on the Bayesian side. It is worth noting that, at the time of writing, the software packages that implement these algorithms are either not publicly available or are not stable enough to be used by a general user, the only exception being MORESANE.

2.4 The Radio Interferometer Measurement Equation

The visibility measurements from an interferometer are corrupted by various propagation and instrumental effects as shown in Figure 2.5. These effects can be separated into two classes: i) direction-independent (DI) effects and ii) direction-dependent (DD) effects. DI effects remain constant across the FoV. Antenna receiver gains are an example of a DI effect. On the other hand, DD effects vary across the FoV. An example of a DD effect is the antenna's primary beam gain.

The radio interferometer measurement equation (RIME) [24, 31] offers a simple yet powerful formalism through which both DI effects and DD effects can be modelled and calibrated out. The derivation of the RIME that follows is largely based on [59, 60].

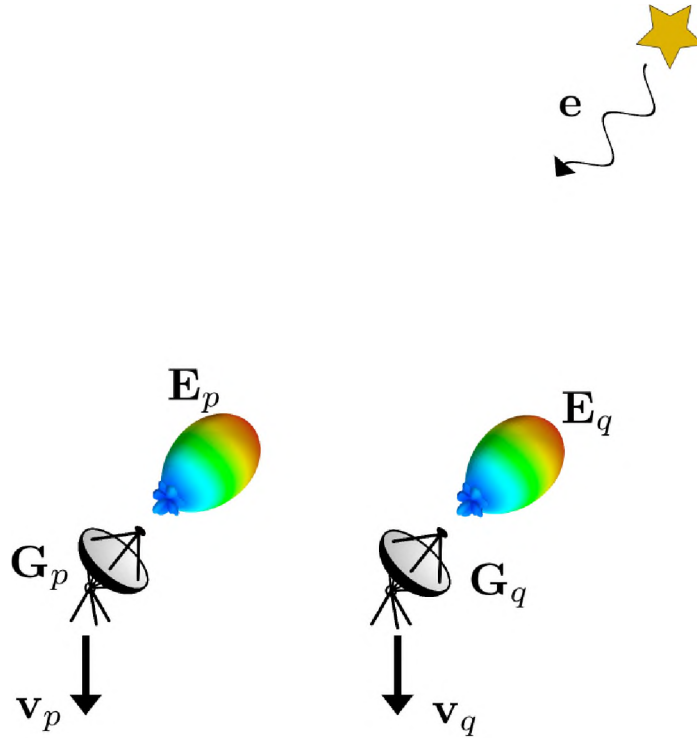


Figure 2.5: Two-element interferometer.

2.4.1 Signal propagation

Consider a quasi-monochromatic point source, which can be described by the complex electric field vector \mathbf{e} . Choosing a Cartesian coordinate system with the electric field propagating along the z axis, the orthogonal components of the field lie along x and

y axes. This electric field can be represented by two complex numbers

$$\mathbf{e} = \begin{pmatrix} e_x \\ e_y \end{pmatrix}. \quad (2.41)$$

All linear transformations of the 2 vector \mathbf{e} may be represented by a matrix multiplication. That is [59],

$$\mathbf{e}' = \mathbf{J}\mathbf{e}, \quad (2.42)$$

where \mathbf{J} is a complex 2×2 Jones matrix [34]. Assuming that all signal corruptions are linear transformations with respect to \mathbf{e} , separate corruptions can be regarded as repeated matrix multiplications, forming a *Jones chain*,

$$\mathbf{e}' = \mathbf{J}_n \mathbf{J}_{n-1} \dots \mathbf{J}_1 \mathbf{e}. \quad (2.43)$$

Note that the order in which the Jones chain is constructed corresponds to the order in which the effects occur. This order is important since matrix multiplication is not commutative in general. Also note that the signal received by the antenna feeds is turned into a complex voltage \mathbf{v} . Further, we assume that each antenna has two feeds a and b (these may be linear dipoles, or left/right circular feeds), and that the feeds are also linear with respect to \mathbf{e} . The relation between \mathbf{e} and \mathbf{v} is yet another matrix multiplication

$$\mathbf{v} = \begin{pmatrix} v_a \\ v_b \end{pmatrix} = \mathbf{J}\mathbf{e}, \quad (2.44)$$

where \mathbf{J} is the cumulative product of all corrupting effects. Equation 2.44 represents the fundamental linear relationship between the voltage, \mathbf{v} , measured by the antenna feeds and the signal vector \mathbf{e} .

The interferometer measures two independent voltage vectors \mathbf{v}_p and \mathbf{v}_q . The correlator then outputs four pairwise correlations

$$\langle v_{pa} v_{qa}^* \rangle, \langle v_{pb} v_{qb}^* \rangle, \langle v_{pa} v_{qb}^* \rangle, \langle v_{pb} v_{qa}^* \rangle, \quad (2.45)$$

where x^* is the complex conjugate of x . These correlations may be represented in the form of a *visibility matrix*

$$\mathbf{V}_{pq} = 2 \begin{pmatrix} \langle v_{pa} v_{qa}^* \rangle & \langle v_{pa} v_{qb}^* \rangle \\ \langle v_{pb} v_{qa}^* \rangle & \langle v_{pb} v_{qb}^* \rangle \end{pmatrix}. \quad (2.46)$$

The origin of the factor of 2 is explained in [59]. The visibility matrix can be re-written as the product of \mathbf{v}_p (as a column vector) and the complex conjugate of \mathbf{v}_q (as a row vector)

$$\mathbf{V}_{pq} = 2 \left\langle \begin{pmatrix} v_{pa} \\ v_{pb} \end{pmatrix} \begin{pmatrix} v_{qa}^* & v_{qb}^* \end{pmatrix} \right\rangle = 2 \langle \mathbf{v}_p \mathbf{v}_q^H \rangle, \quad (2.47)$$

where x^H is the complex conjugate transpose (or Hermitian conjugate) of x . Substituting (2.44) into (2.47) yields

$$\mathbf{V}_{pq} = 2\langle \mathbf{J}_p \mathbf{e} (\mathbf{J}_q \mathbf{e})^H \rangle = 2\langle \mathbf{J}_p (\mathbf{e} \mathbf{e}^H) \mathbf{J}_q^H \rangle. \quad (2.48)$$

Assuming that \mathbf{J}_p and \mathbf{J}_q remain constant over the averaging interval, they can be moved outside the averaging operator:

$$\mathbf{V}_{pq} = 2\mathbf{J}_p \langle \mathbf{e} \mathbf{e}^H \rangle \mathbf{J}_q^H \quad (2.49)$$

$$= 2\mathbf{J}_p \begin{pmatrix} \langle e_x e_x^* \rangle & \langle e_x e_y^* \rangle \\ \langle e_y e_x^* \rangle & \langle e_y e_y^* \rangle \end{pmatrix} \mathbf{J}_q^H. \quad (2.50)$$

The sky *brightness matrix* is defined as

$$\mathbf{B} = 2 \begin{pmatrix} \langle e_x e_x^* \rangle & \langle e_x e_y^* \rangle \\ \langle e_y e_x^* \rangle & \langle e_y e_y^* \rangle \end{pmatrix} = \begin{pmatrix} I + Q & U + iV \\ U - iV & I - Q \end{pmatrix}, \quad (2.51)$$

where, I, Q, U, V are the Stokes parameters [2]. The RIME for a single direction in the FoV

$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{B} \mathbf{J}_q^H. \quad (2.52)$$

For a single point source in direction (l, m) , (2.18) becomes,

$$V(u, v, w) = A'(l, m) I(l, m) e^{-2\pi i [ul + vm + w(\sqrt{1-l^2-m^2}-1)]}. \quad (2.53)$$

Note, that unlike the visibility matrix \mathbf{V}_{pq} which is corrupted by multiplicative linear transformations, the visibility $V(u, v, w)$ in the above equation is free of such corruptions. Recognizing $A'(l, m) I(l, m)$ as the modified sky brightness, and employing the notation developed in this section, the *uncorrupted visibility matrix*, \mathbf{X}_{pq} , can be written as

$$\mathbf{X}_{pq} = \mathbf{B} e^{-2\pi i [u_{pq}l + v_{pq}m + w_{pq}(\sqrt{1-l^2-m^2}-1)]}. \quad (2.54)$$

The “corrupted” RIME then becomes,

$$\mathbf{V}_{pq} = \mathbf{J}_p \mathbf{X}_{pq} \mathbf{J}_q^H. \quad (2.55)$$

The Jones matrix for a direction s (direction towards a source) can be expressed as a chain of matrices accounting for various propagation and instrumental effects

$$\mathbf{J}_{sp} = \mathbf{G}_p \mathbf{E}_{sp}, \quad (2.56)$$

where the Jones matrix \mathbf{G}_p subsumes all DI effects, and \mathbf{E}_{sp} subsumes all DD effects in the direction s . Thus the RIME for a sky containing N point sources is

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\sum_{s=1}^N \mathbf{E}_{sp} \mathbf{X}_{spq} \mathbf{E}_{sq}^H \right) \mathbf{G}_q^H. \quad (2.57)$$

2.5 Calibration

Calibration involves finding a gain matrix \mathbf{J} that minimises

$$\chi^2 = \sum_{pq} \|\mathbf{D}_{pq} - \mathbf{J}_p \mathbf{M}_{pq} \mathbf{J}_q^H\|, \quad (2.58)$$

where \mathbf{D} is the observed visibility data, \mathbf{M} is the model visibility data and $\|\cdot\|$ represents some norm. Once a suitable \mathbf{J} is found, the calibrated data (a.k.a *corrected data*) are then

$$\mathbf{J}_p^{-1} \mathbf{D} (\mathbf{J}_q^H)^{-1}, \quad (2.59)$$

where $()^{-1}$ denotes the inverse of a matrix.

It is often useful to also compute the *corrected residual data*. As will be shown in Chapter 6, the corrected residual data can reveal sources that may have been hidden by uncorrected gain errors (or sidelobes of the sources in the calibration model). Once revealed, the sources can be used to update the model to be used in subsequent calibration step.

$$\mathbf{J}_p^{-1} [\mathbf{D} - \mathbf{J}_p \mathbf{M} \mathbf{J}_q^H] (\mathbf{J}_p^H)^{-1}. \quad (2.60)$$

2.6 Self-calibration

In interferometry, *self-calibration* (or simply selfcal)[16] is a process through which a gain matrix, \mathbf{J} , that minimises (2.58) is obtained in an iterative manner by progressively generating better models \mathbf{M}_i . The process may be described by the following steps:

1. Using an initial sky model \mathbf{M}_0 , find a gain matrix \mathbf{J} that minimises (2.58) in a least squares sense, and compute the corrected data (2.59).
2. Extract an improved model, \mathbf{M}' , from the corrected data.
3. Use the improved model, \mathbf{M}' in another calibration step and find a new gain matrix \mathbf{J}' . Compute the corrected data.
4. Repeat steps 2 and 3 until a satisfactory corrected data set.

2.6.1 Calibration Artefacts

The calibration problem may also be stated as

$$\sum_{pq} \|\mathbf{D}_{pq} - \mathbf{J}_p \mathbf{M}_{pq} \mathbf{J}_q^H\| \leq \varepsilon', \quad (2.61)$$

where ε' represents all unmodelled instrumental and propagation effects, and the thermal noise component, σ , mentioned earlier. That is,

$$\varepsilon' = \varepsilon(\varepsilon, \sigma), \quad (2.62)$$

where ε represents the unmodelled instrumental and propagation effects. Assuming perfect knowledge of the sky and the instrument, the best one can do is calibrate to the thermal noise level, that is $\varepsilon' = \sigma$. In this case, σ can be driven arbitrarily low by observing for longer since the thermal noise scales as the inverse of the square root of the total observation time and frequency band. However, in practice it is impossible to have perfect knowledge of either the sky or the instrument and while ε can be driven to the thermal noise level [42, 61], it can never be totally eliminated. Moreover, as will be discussed in Chapters 3 and 6, driving calibration artefacts to the noise level, achieving the so-called *noise-limited* reduction, is not a straightforward task.

Chapter 3

Forecasting Calibration Limitations of the MeerKAT Telescope

MeerKAT is the South African precursor to the SKA telescope.¹ The 64 13.5 m alt-azimuth mounted MeerKAT antennas with offset Gregorian (OG) optics (see Figure 3.1) will eventually be part of the SKA mid-frequency telescope (SKA-MID). With a sensitivity (A_e/T_{sys}) of 6.5 m²/K and a FoV of 0.95 square degrees at L-band (1-2 GHz) [67], MeerKAT has a survey speed that is five times faster than the VLA which has a similar sensitivity but a FoV of 0.20 square degrees at the same band. The specifications of the MeerKAT receivers are tabulated in Table 3.1, and the antenna configuration, along with a sample uv -coverage is shown in Figure 3.2. Figure 3.3 shows sensitivity estimates for the three MeerKAT bands for different uv -weighting schemes as a function of time. The science case is centred around deep and wide area continuum and line surveys as well as transient searches [67].

The high sensitivity and large FoV of the telescope mean that calibrating and imaging MeerKAT data pose serious challenges. Instrumental artefacts that are below the thermal noise level with less sensitive instruments will now have to be treated with greater care. These artefacts are due to imperfect modelling of the instrument and the sky coherency. In principle, they can be driven down to the thermal noise level using DD calibration techniques such as SAGECal[73], differential gains [60] and SPAM[30]. A good example of how some of these tools can produce noise-limited reductions is the 5 million to 1 dynamic range images from a VLA observation of the field around the source 3C147 at L-band [42]. It must be noted, however, that these reductions were obtained from relatively small datasets (13 hours observation time and 256 MHz bandwidth), and performing these reductions with hundreds of observing hours of MeerKAT data (MeerKAT also has about 5 times more baselines compared to the VLA) is going to be computationally demanding, both in terms of storage and processing. In continuum mode (4096 channels), 100 hours (four correlation products) of data is about 10 Terabytes, this further increases to 85 Terabytes

¹<https://www.skatelescope.org>

in spectral line mode (32768 channels); similar (sometimes higher) data rates are already a reality for the LOFAR telescope. In fact, the visibility data sets that will be generated will be so large that keeping them for more than about six months will be prohibitively expensive [67, private communication]. This means that the data have to be calibrated and stored as images within this time frame (calibration requires visibility data), hence the induced calibration artefacts will be frozen into the resulting images. For deep surveys, this means that the survey depth will be reached by stacking up many images from different stages of the survey lifetime. In this thesis, we consider the survey depth as the root mean square (RMS) of the pixel amplitudes in the residual image (after subtracting the reconstructed sky model) from the survey data.

In this chapter we use simulations to gauge the impact of stacking images from noise-limited data reductions. In particular, we investigate how calibration artefacts accumulate with time (or as many of these images are stacked over time) and the impact of this accumulation on the survey depth. In this thesis, data reduction refers to the process through which one solves for antenna and propagation effects using a technique such as the self-calibration technique described in section 2.6.

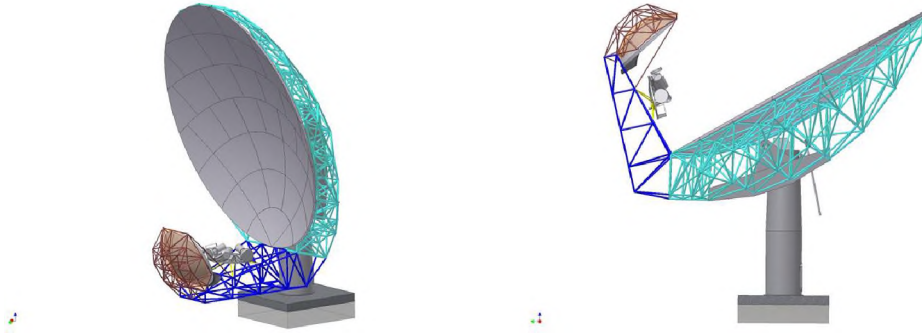


Figure 3.1: Schematics of the MeerKAT antenna. These images are taken from the MeerKAT Telescope website [67]

Table 3.1: MeerKAT frequency band information and corresponding SEFD values. The system temperature of the S-Band receivers has not yet been confirmed, so we use the most conservative estimates; these are marked with an asterisk.

Band	Bandwidth [MHz]	Band Centre [MHz]	SEFD [Jy]	$A_{\text{eff}}/T_{\text{sys}}$ [m ² /K]
1 (UHF)	435	800	830	6.5
2 (L)	810	1400	531	4.8
3 (S)	1750	2600	531*	4.8*

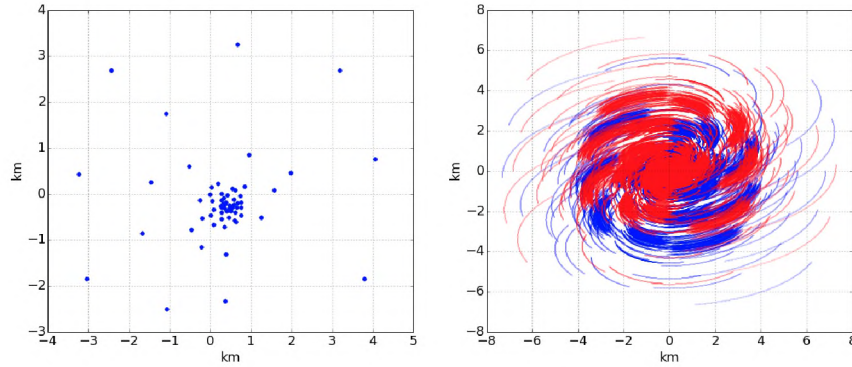


Figure 3.2: MeerKAT antenna layout (left) and a uv-coverage from a 4 hour simulated observation at a declination of -30° (right).

3.1 The MeerKAT primary beam

The full width at half maximum (FWHM) size of the MeerKAT primary beam pattern is about 1.1° at 1400 MHz. The OG optics mean that the primary reflector is unblocked, which leads to an almost circularly symmetric primary beam (up to about the 4th sidelobe) while at the same time lowering the sidelobe levels (compared to a prime focus design). Figure 3.4 shows a simulated MeerKAT primary beam pattern and Figure 3.5 shows the Jones matrix of this primary beam

$$\mathbf{E} = \begin{pmatrix} E_{XX} & E_{XY} \\ E_{YX} & E_{YY} \end{pmatrix}, \quad (3.1)$$

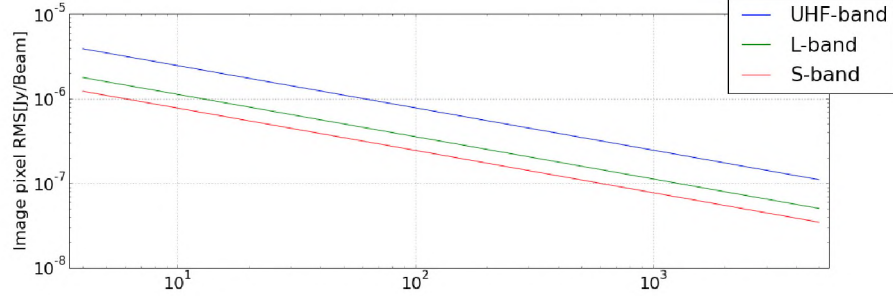
where E_{XX} , E_{XY} , E_{YX} , E_{YY} are the so-called *voltage beam patterns*; XX, YY are the parallel hand correlations, and XY, YX are the cross-correlations. These beam patterns were simulated using a full electromagnetic solver, *Multilevel Fast Multipole Method*, implemented in FEKO². The simulations were carried out by EMSS Antennas,³ a South African company that specialises in the design and electromagnetic simulations of antenna systems.

3.2 The experiment

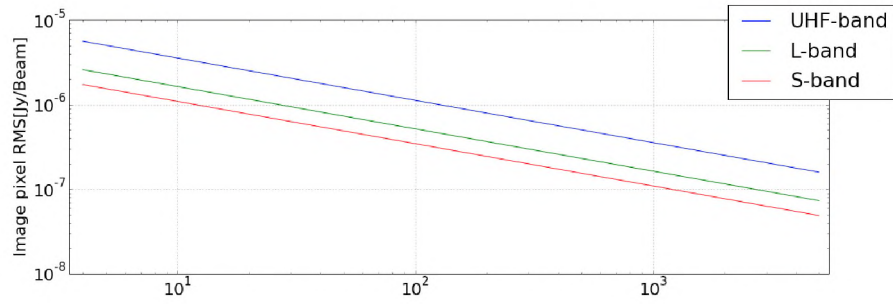
The aim of this experiment is to quantify how calibration artefacts accumulate with time, and how this accumulation affects survey depth. For the MeerKAT telescope the most significant DD effects will be from the primary beam because other sources of DD effects such as ionospheric phase rotation are not significant at the MeerKAT

²<http://www.feko.info>

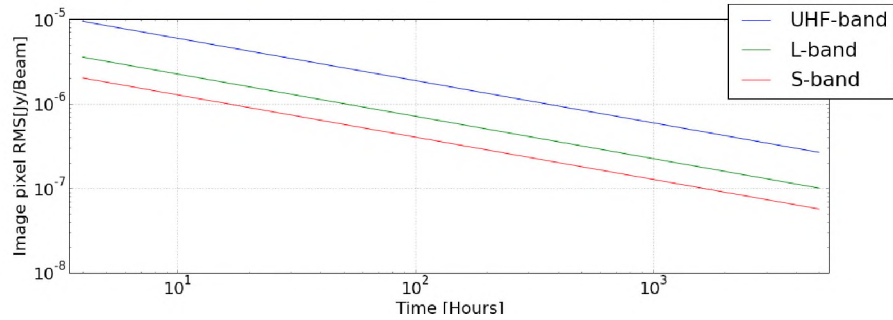
³<http://www.emss.co.za/antennas.php>



(a) Briggs robust=2



(b) Briggs robust=0



(c) Briggs robust=-2

Figure 3.3: Image pixel RMS from MeerKAT noise simulations as a function of observation time for the MeerKAT UHF-band (435 MHz bandwidth; blue), L-band (821 MHz bandwidth; red) and S-band (1750 MHz bandwidth; green). The RMS values are measured from images obtained using Briggs uv -weighting with a robustness parameter of 2 (a), 0 (b) and -2 (c).

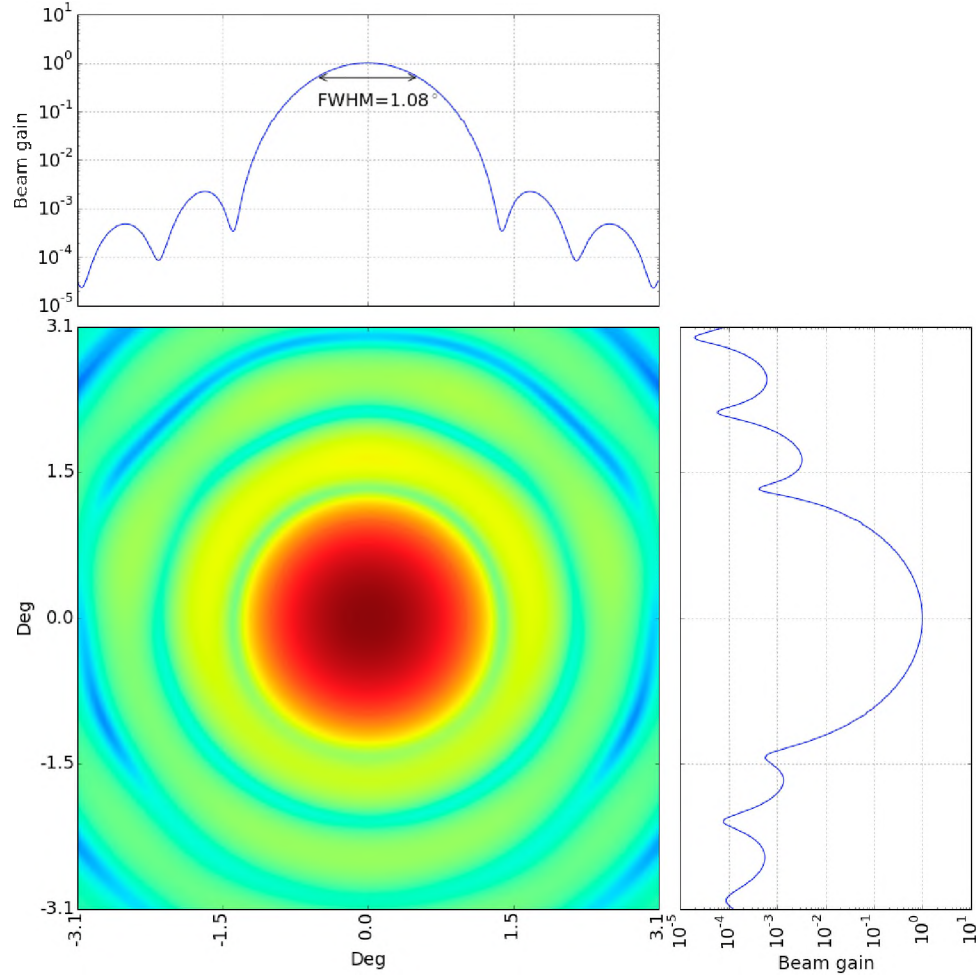


Figure 3.4: The MeerKAT primary beam pattern, along with cross-sections along $x = 0$ (top) and $y = 0$ (right).

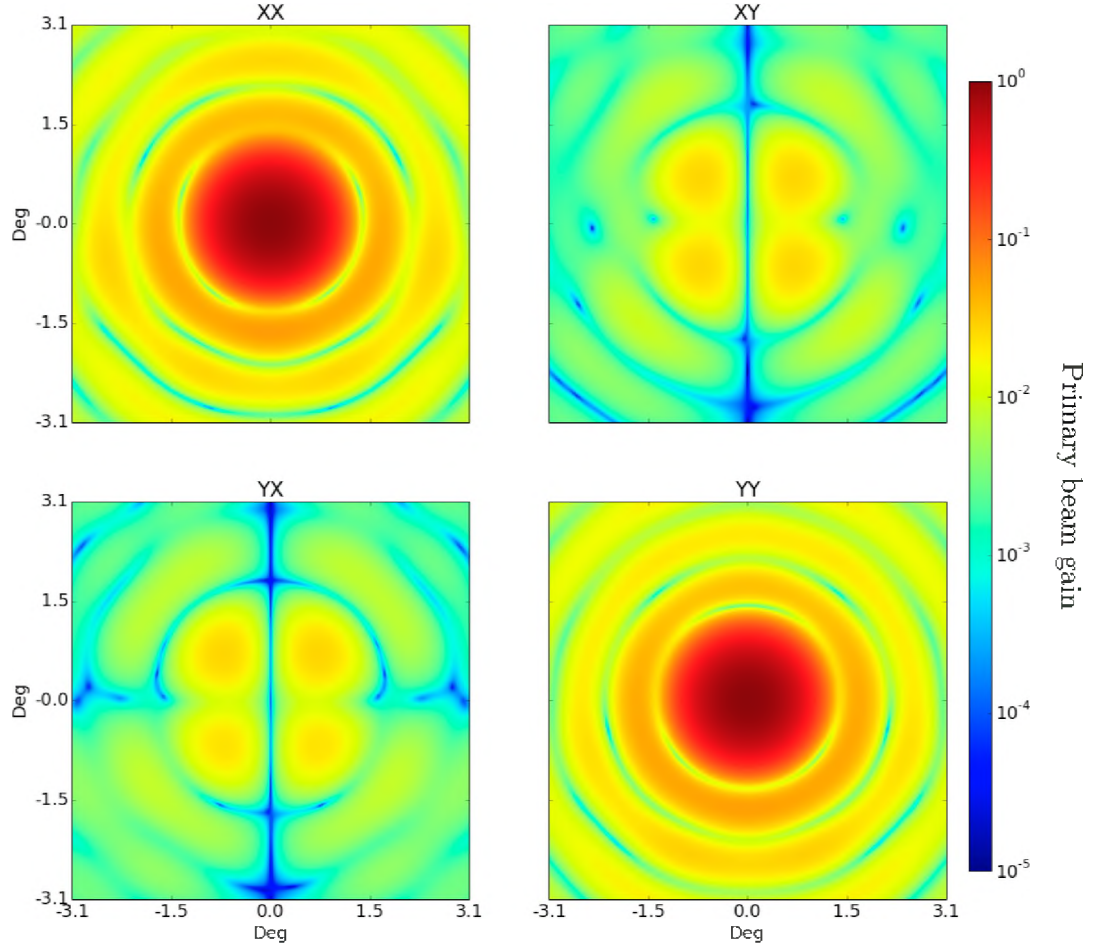


Figure 3.5: Amplitudes of the Jones matrix components of the MeerKAT primary beam (voltage beams).

observing frequencies. Furthermore, of the primary beam related errors we will only focus on antenna pointing errors. These errors are due to antenna pointing offsets which have various causes, including drive mechanics, gravitational loads, thermal expansion, wind pressure, and even control software. These cause the nominal beam shape \mathbf{E}_{pe} to produce a different response

$$\mathbf{E}_{\text{pe}} = \mathbf{E}(l + \delta l_{\text{pe}}, m + \delta m_{\text{pe}}), \quad (3.2)$$

where $\delta l_{\text{pe}}, \delta m_{\text{pe}}$ are the pointing offsets. These offsets are, in general, time-dependent, but will be considered static in this chapter. Other errors, such as primary beam attenuation and parallactic angle rotation (the primary beam of an alt-azimuth mounted antenna rotates with respect to the sky) can be modelled very well with modern calibration tools[42, 61].

Because of insufficient computing resources, we consider a narrow bandwidth of 10 MHz, much smaller than the full 810 MHz, meaning that we will ignore how the accumulation of these artefacts is affected by wide-band effects, e.g, the variation of the primary beam with frequency, and how it is coupled with pointing errors. The estimates derived from these simulations are therefore best-case scenario estimates. Nonetheless, the framework described in this work can easily be extended to larger bandwidths.

3.2.1 The strategy

Below is a summary of our simulation strategy:

1. We simulate MeerKAT visibility data from a sky model (to be discussed later in this section), and add thermal noise. The simulation also includes complex receiver gains (these will also be referred to as G-errors). We also include a primary beam model and antenna pointing errors which we model as static offsets. We call this dataset **DATA**.
2. The same as the above is done, but without the thermal noise contribution. Note that we use the same realisation of G-errors and the pointing errors. We call this dataset **DISTILLED DATA**.
3. We then perform a DI and DD calibration of **DATA** with the same sky model and primary beam used for the simulation. We save the gain solutions, and then save and image the corrected residual data (2.60). These residual data are a combination of the thermal noise and artefacts due to unmodelled instrumental effects.
4. Apply the gains from the above step to **DISTILLED DATA**, and save and image the corrected residual data. Note that this data represents the artefacts due to unmodelled instrumental effects (pointing errors) and the impact of thermal noise on the calibration solutions.

5. Repeat steps 1-4 until the required survey time is reached (1608 hours in this experiment), each time saving the corrected residual image.

For MeerKAT, the expected pointing accuracy in ideal conditions is 15'' and can go up to 25'' in poor conditions[67]. We do this for both low and high pointing error conditions.

3.2.2 Telescope simulations

The CASA[64] based `simms`[39] tool was used to create empty visibility datasets. The standard data structure of visibility data is called a `measurement set`⁴, a hierarchical directory in which the data and metadata are stored as tables and sub-directories. The `MeqTrees`[45] tool was then used to populate the MS with simulated data.

We simulate 8-hour MeerKAT observations at L-band (1-2 GHz) with ten 1 MHz channels and integration time of 5 s at a declination of -30°. The simulations include DI (complex receiver gains) and DD (primary beam) effects as well thermal noise (estimated from the system temperature values in Table 3.1 via the radiometer equation (2.37)). For convenience we assume perfect knowledge of the primary beam, that is, we use the same primary beam model in the simulation and calibration. In reality, the primary beam is not known to a high level of accuracy, however this assumption is good enough for the purposes of this work. The data are imaged with the `wsclean`[47] tool.

3.2.3 The sky model

The sky model used in these simulations is from the SKA Simulated Skies (S³) extragalactic continuum sources[70]. It has 102 sources over a region covering 2 square degrees, and a flux range of 2.42-0.01 Jy. A model with fairly bright sources was chosen because the DD effects around bright sources dominate the contribution made by the spurious emission (henceforth referred to as the artefact budget). The model is shown in Figure 3.6 superimposed on the MeerKAT primary beam pattern.

3.2.4 Data calibration

The calibration is done using the `MeqTrees` package, using the RIME

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\sum_{s=1}^N \Delta \mathbf{E}_{sp} \mathbf{E}_{sp} \mathbf{X}_{spq} \mathbf{E}_{sq}^H \Delta \mathbf{E}_{sq}^H \right) \mathbf{G}_q^H. \quad (3.3)$$

In the above, the Jones matrix \mathbf{E} accounts for all modelled DD effects; these are the primary beam attenuation and parallactic angle rotation in our case, while the Jones matrix \mathbf{G} , dubbed a *differential gain* matrix [61], subsumes all unmodelled

⁴<https://casa.nrao.edu/Release3.3.0/docs/UserMan/UserManse7.html>

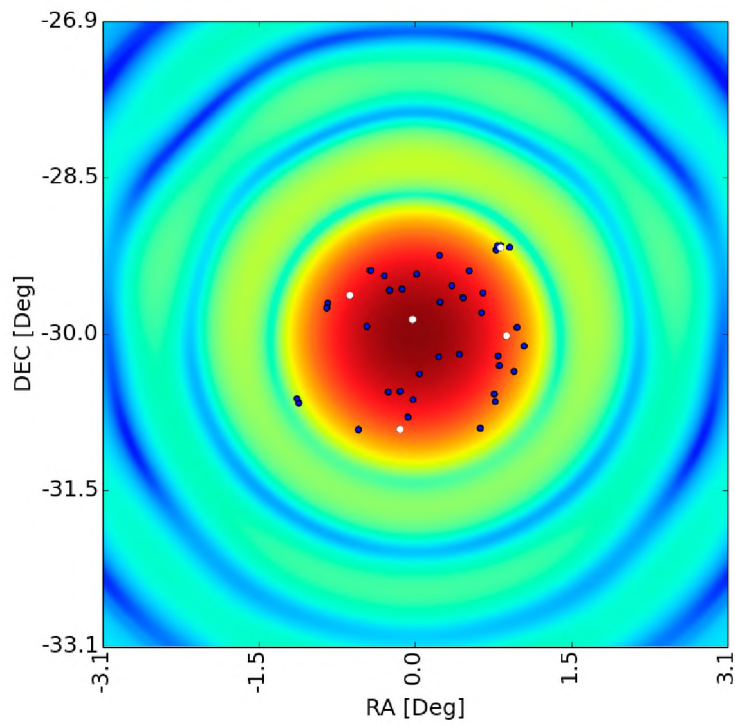


Figure 3.6: The sky model used in the simulations superimposed on the MeerKAT primary beam. The white points indicate the directions in which direction-dependent solutions were applied.

DD effects– the pointing errors in our case. Computing the differential gain is very computationally expensive, and in practice one has to choose sources (or directions) for which to solve for them, meaning that the differential gain is an identity matrix for all but a select set of sources. The rest of the symbols are as defined in Equation 2.57.

3.2.5 Results and discussions

The corrected residual maps in Figure 3.7 show that DD calibration is required for a noise-limited calibration with MeerKAT, even for an 8-hour, 10 MHz synthesis. The map on the left is for a DI-only reduction, and the map on the right is from a reduction that also corrects for DD effects. The DI-only map has spurious emission which mostly emanates from a region at the top right-hand corner – this is the location of a cluster of sources that falls on the edge of the primary beam, the brightest of which is 2.4 Jy (see Figure 3.6) – which is driven down after a DD calibration. DD solutions are applied in 5 directions, as indicated in Figure 3.6. Note however that the DD calibration does not eliminate the spurious emission but merely drives it down below the noise level. Furthermore, once the noise has been “distilled” out, the spurious emission still remains as can be seen in Figure 3.8. It is the accumulation of this spurious emission that we aim to quantify.

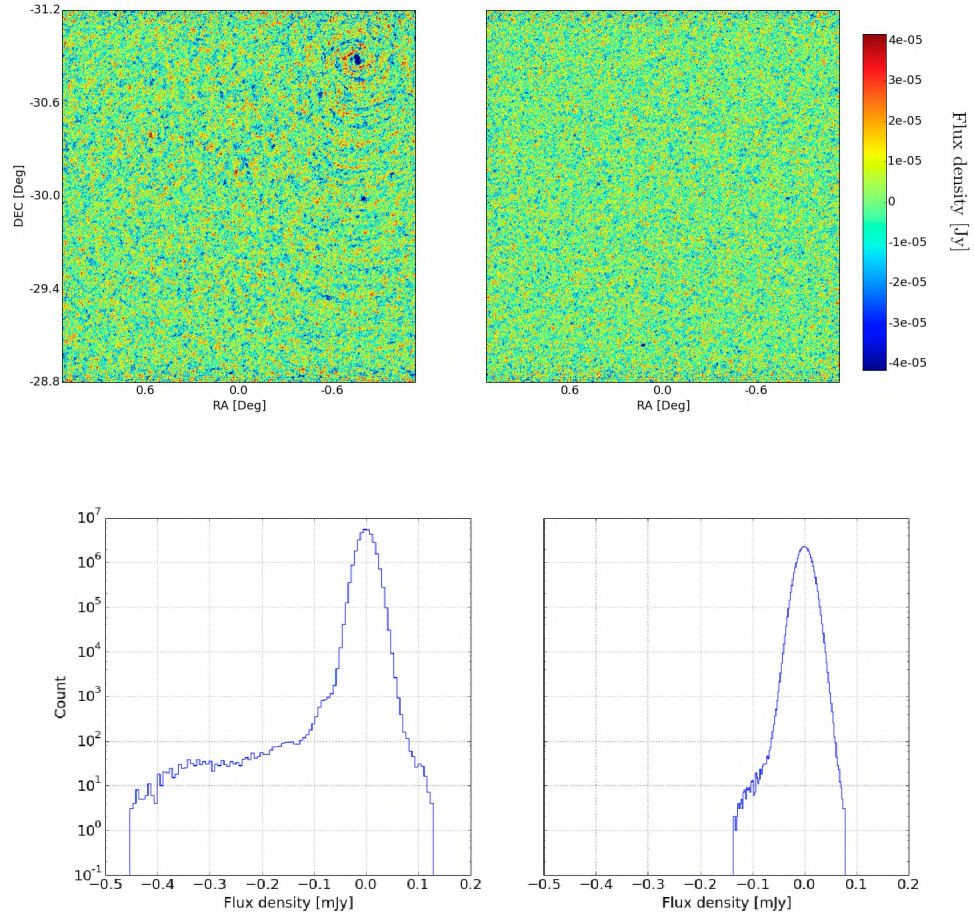


Figure 3.7: Corrected residual maps for an 8-hour scan with MeerKAT. On the top panel, the image on the left is a corrected residual image from a reduction which does not correct for direction dependent gains, while the image on the right corrects for direction dependent effects. The bottom panel shows the corresponding histograms of the signal distribution for both of these images.

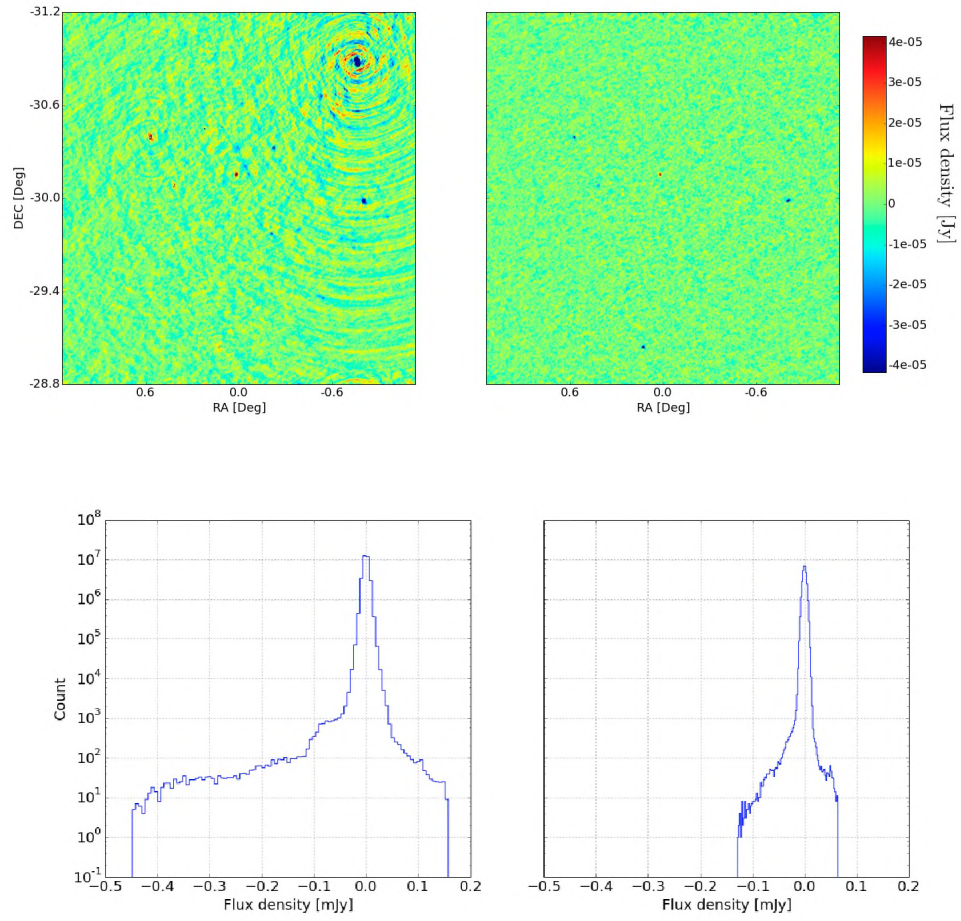


Figure 3.8: Distilled (without thermal noise) residual maps for an 8-hour scan with MeerKAT. On the top panel, the image on the left is a corrected residual image from a reduction which does not correct for direction dependent gains, while the image on the right corrects for direction dependent effects. The bottom panel shows histograms of the signal distribution for both these images.

In Figure 3.9 we show how the spurious emission due to pointing errors accumulates with time for a pointing error of $15''$ (solid lines), as well as the derivatives w.r.t time (right y-axis; dashed lines). The plot shows both the corrected residual from **DATA** (corrected residual; blue) and **DISTILLED DATA** (distilled residual; red), as well as the thermal noise (black). As expected, the noise decays at a constant rate of -0.5 (in log space). The corrected residuals initially decay at constant rate slightly above -0.5 but eventually decay at rate that is increasingly slower, while the distilled residuals start decaying at a rate of about -0.45 and at 1608 hours the pointing error artefacts dominate the image noise contribution. Also by that time the distilled residual is decaying at a rate very close to zero. At this point the depth is about $0.81 \mu\text{Jy}$, and this depth, henceforth referred to as the *artefact floor*, is a limit beyond which observing for longer does not get one deeper. A similar trend can be seen for the case with a pointing accuracy of $25''$ in Figure 3.10, the only notable difference being that the corrected and distilled residuals decay slower and the artefacts dominate the noise contribution at 1368 hours, and the artefact floor is $0.87 \mu\text{J}$.

Figures 3.11 and 3.12 show corrected residual images at different stages of the accumulation for the $15''$ and $25''$ cases respectively, and Figures 3.13 and 3.14 show the distilled residuals at different stages. It is worth noting that some of the bright sources are either over-subtracted or under-subtracted in a given pointing, and though the effect of this imperfect subtraction is very small in a single scan (see Figures 3.7 and 3.8) it becomes more pronounced as many such images are stacked together, as can be seen in Figures 3.11 and 3.15.

The classical confusion limit (the depth at which the telescope cannot resolve the difference between individual sources; see [11, 62] for more details) for the MeerKAT telescope at L-Band is estimated at $2 \mu\text{Jy}$ ⁵, and, given the large bandwidth and high sensitivity, this fundamental limit will be reached within a few hours (see Figure 3.3), meaning that continuum MeerKAT surveys will be limited by classical confusion well before the aforementioned artefact floor. However, for deep spectral line surveys, such as the LADUMA (Looking at the Distant Universe with MeerKAT Array) survey [29], where classical confusion is not an issue (spectral line sources only occupy a limited range of frequencies, and the number of sources that occupy the same frequency range is much smaller compared to sources that emit continuously across the radio spectrum) this artefact floor becomes important. Similar to the strategy described here, in this case the field of interest would be observed many times to achieve the survey depth. Each observation would then be calibrated, the continuum emission subtracted and the data stored as an image cube (the third axis would be the frequency axis). These data cubes would then be combined to form the survey science product.

It is important to ensure that this science data product is not limited by calibration artefacts. Now, considering a typical 72 kHz channel width, the artefact floor is at

⁵Ian Heywood; private communication.

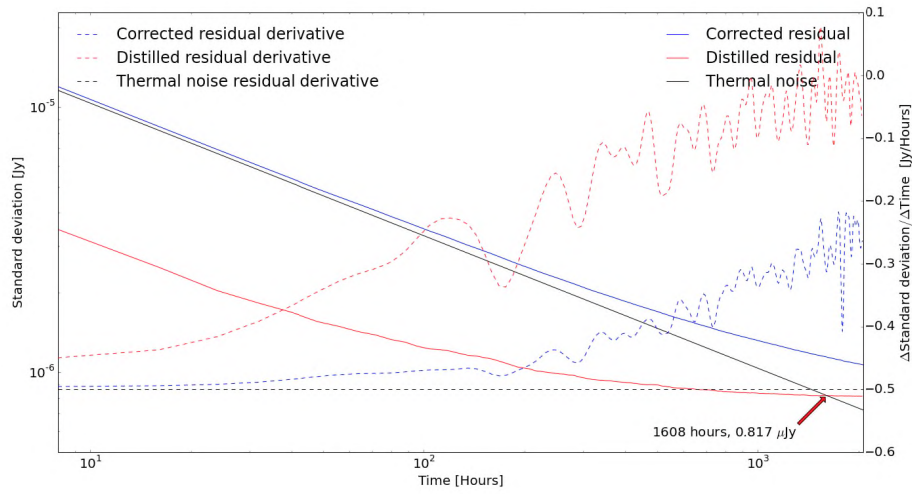


Figure 3.9: Accumulation of spurious emission due to antenna pointing errors with time for a pointing error of $15''$. The blue curve is the accumulation of corrected residual images, the red curve is the corrected residual with the thermal noise “distilled” out (distilled residual), and the black curve is the accumulation of the thermal noise. The extra y-axis (right) shows the derivative (in log space) of the corrected and distilled residual w.r.t time (dashed). The red arrow marks the point at which the pointing error artefacts (distilled residuals) start to dominate the image noise contribution.

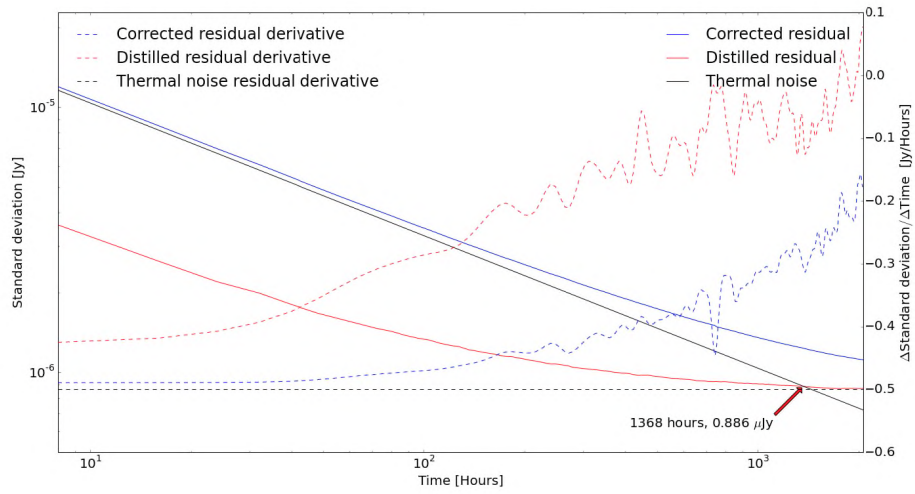


Figure 3.10: Accumulation of spurious emission due to antenna pointing errors with time for a pointing error of $25''$. The blue curve is the accumulation of corrected residual images, the red curve is the corrected residual with the thermal noise “distilled” out (distilled residual), and the black curve is the accumulation of the thermal noise. The extra y-axis (right) shows the derivative (in log space) of the corrected and distilled residual w.r.t time (dashed). The red arrow marks the point at which the pointing error artefacts (distilled residuals) start dominating the image noise contribution.

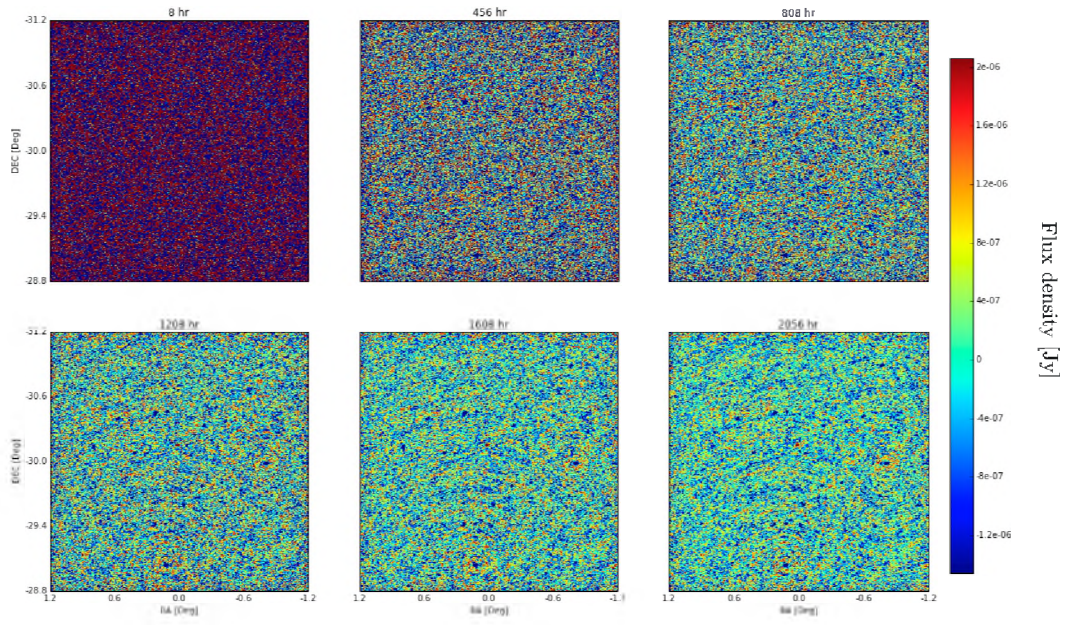


Figure 3.11: Corrected residual images at different stages of the accumulation. These images are from a simulation with a pointing accuracy of $15''$.

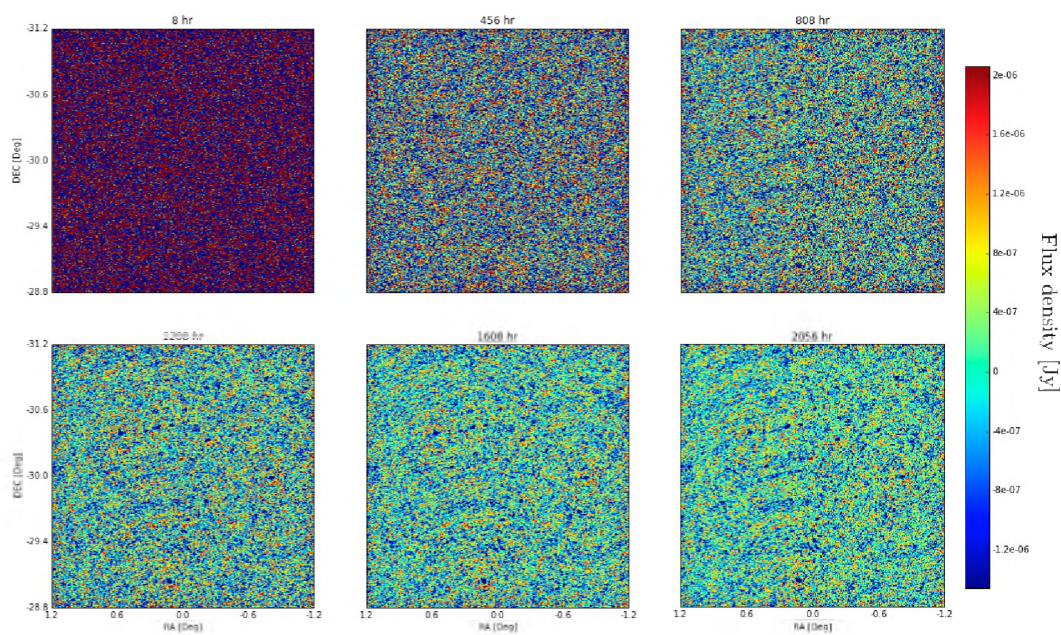


Figure 3.12: Corrected residual images at different stages of the accumulation. These images are from a simulation with a pointing accuracy of $25''$.

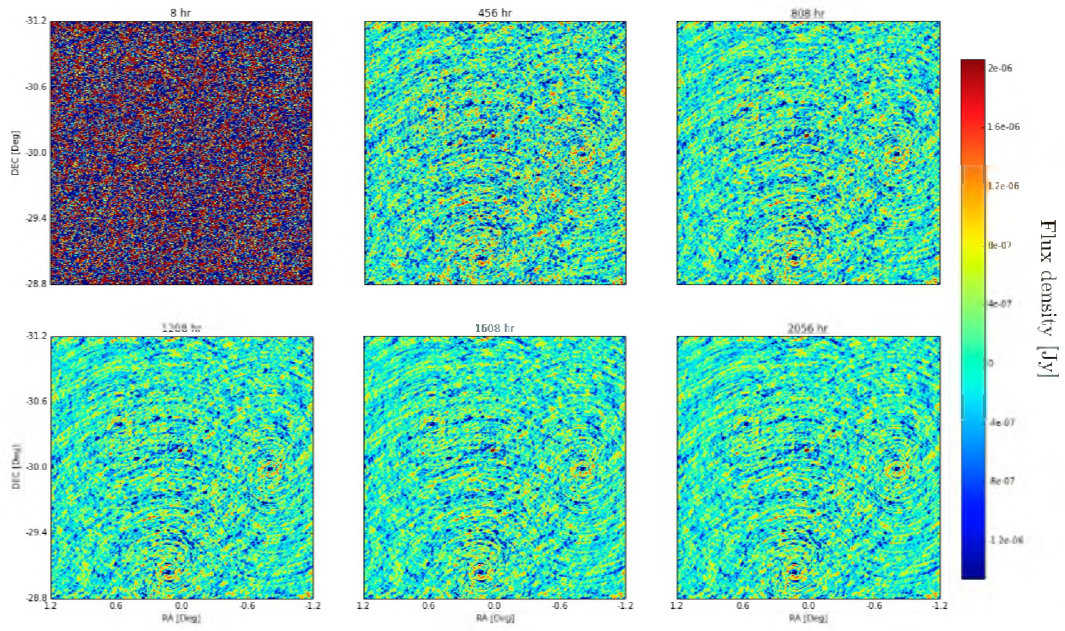


Figure 3.13: Distilled residual images at different stages of the accumulation. These images are from a simulation with a pointing accuracy of $15''$.

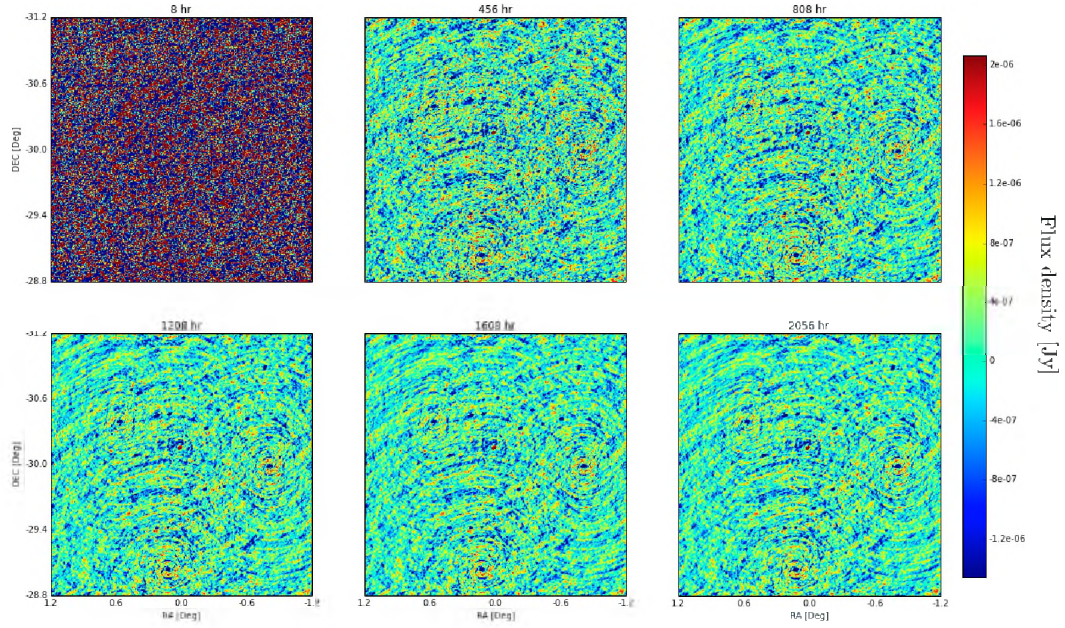


Figure 3.14: Distilled residual images at different stages of the accumulation. These are images are from a simulation with a pointing accuracy of $25''$.

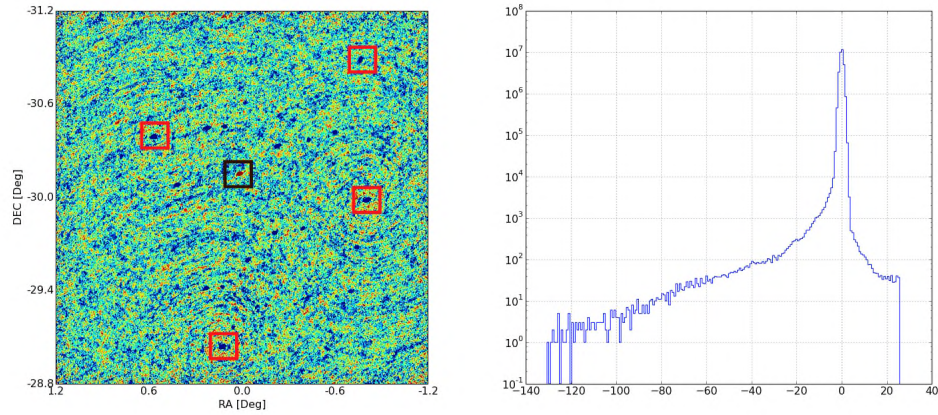


Figure 3.15: Cumulated corrected residual image after 2056 hours for a pointing error of $15''$ (left) along with the histogram (right) of the map. The black boxes mark sources that are under-subtracted, and the red boxes mark sources that are over-subtracted.

about $9.60 \mu\text{Jy}$ for a pointing error of $15''$ as can be seen in Figure 3.16 (derived from Figure 3.9), and after 1748 hours it has not been reached. For a pointing error of $25''$ the artefact floor is at $10.32 \mu\text{Jy}$ as can be seen in Figure 3.17. This is the best case scenario, since the 10 MHz bandwidth we have simulated is not large enough to capture how systematic frequency variations (the primary beam for example) will affect the artefact floor. Also note that the figures show a depth of $15.50 \mu\text{Jy}/\text{beam}$ and $15.89 \mu\text{Jy}/\text{beam}$ being reached after 1000 hours in a 72 kHz channel in the presence of static pointing offsets with an RMS value of $15''$ and $25''$ respectively. This is about a factor 2 deeper the $33 \mu\text{Jy}$ targeted depth⁶. The factor of 2 discrepancy can be accounted for if one considers that the MeerKAT L-Band receiver system temperature used in our simulations is about 2 times better than the value projected when the original survey parameters were calculated.

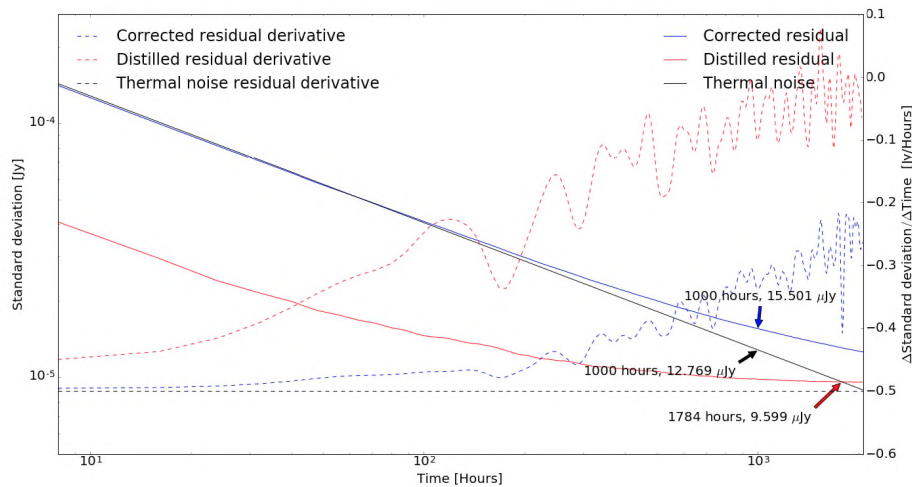


Figure 3.16: Accumulation of spurious emission due to antenna pointing errors with time for a pointing error of $15''$ in a 72 kHz channel. The blue curve is the accumulation of corrected residual images, the red curve is the corrected residual with the thermal noise “distilled” out (distilled residual), and the black curve is the accumulation of the thermal noise. The extra y-axis (right) shows the derivative (in log space) of the corrected and distilled residual w.r.t time (dashed). The red arrow marks the point at which the pointing error artefacts (distilled residuals) start to dominate the image noise contribution. The black arrow marks thermal noise after 1000 hours, a possible milestone for the LADUMA survey at L-Band, while the blue arrow marks the effective noise (thermal noise and calibration artefacts) at the same point.

For each pointing accuracy value, the simulations in this chapter produced about 3.3 Tb of data. That includes, the visibility data, the images and the gain tables (see

⁶<http://www.acgc.uct.ac.za/files/Maddox.pdf>

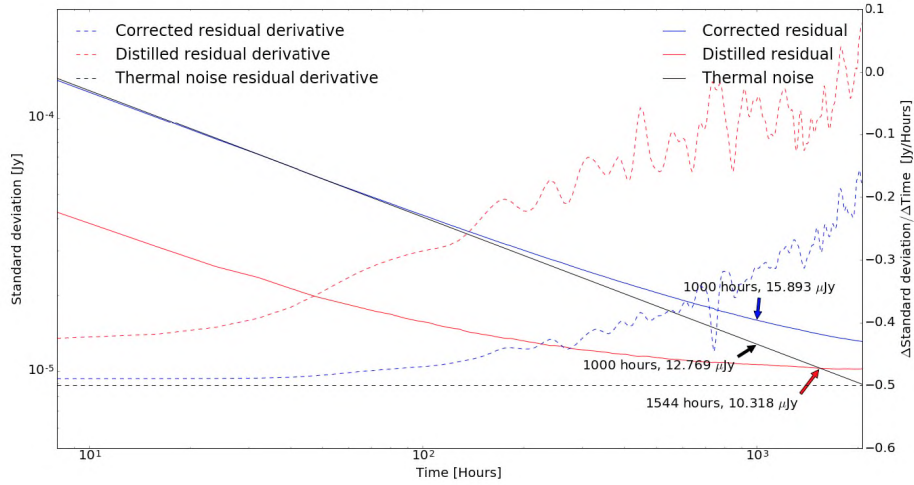


Figure 3.17: Accumulation of spurious emission due to antenna pointing errors with time for a pointing error of $25''$ in a 72 kHz channel. The blue curve is the accumulation of corrected residual images, the red curve is the corrected residual with the thermal noise “distilled” out (distilled residual), and the black curve is the accumulation of the thermal noise. The extra y-axis (right) shows the derivative (in log space) of the corrected and distilled residual w.r.t time (dashed). The red arrow marks the point at which the pointing error artefacts (distilled residuals) start to dominate the image noise contribution. The black arrow marks thermal noise after 1000 hours, a possible milestone for the LADUMA survey at L-Band, while the blue arrow marks the effective noise (thermal noise and calibration artefacts) at the same point.

Table 3.2 for details). Each simulation (one pointing) takes about 4.5 hours to run on an 8-cpu system with 256 Gb of RAM, which amounts to 48.6 days of processing (259 pointings). This is a total of 97.1 days of processing for both values of the pointing accuracy, and requires 6.6 Tb of storage space.

Table 3.2: Summary of the data sizes of the simulated data and output products in this chapter (for one pointing accuracy value). Note that the simulations produce two gain tables, one for the DI calibration, and the other for the DD calibration. The total data size of the simulation is $3375 \text{ Gb} = 3.3 \text{ Tb}$.

	MS	Images	Gain Tables (DI)	Gain Tables (DD)
Size	12 Gb	720 Mb	57 Mb	282 Mb
Quantity	259	259	259	259
Total	2412 Gb	182 Gb	14 Gb	71 Gb

To repeat this experiment for the full band, one could consider splitting the band into 12 sub-bands, each with 64 1 MHz channels. In this case, for each sub-band, with an 8-hour synthesis and a 5 s integration time, the simulated visibility dataset (one pointing) is about 68 Gb, leading to a 816 Gb dataset for each pointing. Processing one sub-band takes about 20 hours (on the system described earlier), meaning that it would take about 10 days to process data from a single pointing on the same system. On the one hand, this suggests that it may not be feasible to repeat this experiment for the entire band. On the other hand, since each pointing is completely independent and the data can be generated in situ, this experiment is ideal for testing the feasibility of cloud computing platforms such as GCE or AWS, which allow for the creation of a large number of virtual machines (VMs). In fact, about a third of the simulations in this chapter were executed in the GCE environment ⁷. We found the cloud platform to be ideal for an experiment of this type, but we note that processing large datasets (that cannot be partitioned into small independent parts) may take a prohibitive amount of time, as the size of the VMs is limited (at the time of writing) to 256 Gb RAM and 32 CPUs. However, it may be advantageous for the radio astronomy community to exploit the flexibility, support and rapid development offered by these services by developing data reduction techniques that can distribute the processing across multiple VMs (or nodes), even across different platforms.

The biggest impediment, however, to having cross-platform and distributed data processing in radio astronomy is the fact that (as will be discussed in detail in the next chapter) radio astronomy software is often unstable and unreliable. For example, in setting up the simulations in this chapter on three different platforms: i) a computing server at Rhodes University; ii) a cluster server at the South African Centre for High Performance Computing (CHPC); iii) and the GCE cloud computing service, we spent about three months installing and building the required software packages and

⁷We acknowledge the SKA South Africa office for making these services available to us

ensuring that they could co-exist on the same platform. It is this laborious software installation process that discourages some in the community from doing simulations like this one, which can be vital in making key scientific as well telescope design decisions. In the next chapter, we will discuss the importance of software packaging (in an attempt to standardise software compilation and installation) and show how it can be used together with container technologies to produce cross-platform and distributed data-processing packages.

3.3 Conclusions

We have shown the impact of pointing errors when stacking up multiple images from noise-limited reductions in order to achieve a deep survey, in particular, how calibration artefacts due to pointing errors accumulate with time when observing the same field multiple times, and how this accumulation can limit the survey depth.

The spurious emission from a single scan can be driven below the thermal noise level, effectively achieving a noise-limited reduction. However, as such images are accumulated, the thermal noise decreases but the spurious emission becomes increasingly significant and eventually dominates the noise budget. The spurious emission has a non-Gaussian distribution, and its accumulation eventually plateaus at some depth, the artefact floor. Stacking up more images (observing for longer) beyond this artefact floor becomes increasingly useless as no extra depth is gained by observing the field for longer. We estimate this artefact floor at $9.95 \mu\text{Jy}$ for a LADUMA type survey, assuming a mean pointing error of $15''$, and $10.32 \mu\text{Jy}$ assuming a mean pointing error $25''$. Nonetheless, our simulations suggest that LADUMA can reach its target depth of about $16 \mu\text{Jy}/\text{beam}$ (correcting for the better than projected system temperature of the MeerKAT L-Band receiver) in a 72 kHz channel in 1000 hours before hitting the artefact floor.

We have also discussed the need for cross-platform and distributed data processing in radio astronomy, a topic that will be discussed in greater detail in the next chapter.

Chapter 4

Platform-Independent Radio Interferometric Scripting

Radio interferometry software packages such as [45, MeqTrees], [64, CASA], [50, Montblanc] and [43, OSKAR] have grown sophisticated enough to allow one to start addressing some of the calibration and imaging challenges that will be faced with SKA size datasets. Furthermore, frameworks such as `pyxis`¹ and `Docker`² have enhanced their interoperability and facilitated their deployment on cluster and cloud computing systems, paving the way for powerful and flexible data reduction and synthesis pipelines. In this chapter we show how these tools can be combined to create flexible and platform-independent radio interferometric data synthesis and reduction pipelines.

4.1 Radio Astronomy Software

The deployment of new radio telescopes such as MWA, ALMA and LOFAR, the preparations for upcoming telescopes such as ASKAP, MeerKAT and the SKA, and the refurbishing of legacy instruments such as the VLA and WSRT have led to major developments in radio astronomy software packages. Much of this can be attributed to new insights enabled by Hamaker’s formulation of the RIME [31], which has led to the development and improvement of data synthesis and calibration packages, most notably `MeqTrees`, `OSKAR` and the LOFAR software packages. At the same time, legacy packages such as `CASA` have also seen major improvements, and have been revamped to take advantage of novel algorithms – for example, multi-scale multi-frequency synthesis (MS-MFS) imaging [51] – and techniques such as parallel computing. A suite of algorithms and software tools has emerged from the introduction of Bayesian and compressive sensing (CS) techniques into radio astronomy. The most notable among

¹<https://github.com/ska-sa/pyxis>

²<https://docker.com>

the Bayesian techniques are **BIRO** [37] and **RESOLVE** [35], and CS based packages include **MORESANE** [17] and **PURIFY** [9].

On the one hand these packages and algorithms show great promise, but on the other hand the packages are generally written by astronomers, not by software engineers. Hence, most of these packages are i) difficult to build and/or to install, ii) have multiple dependencies that often conflict with standard libraries, and iii) unreliable. Unfortunately, this is the case for legacy software packages as well, but more importantly, because of the many dependencies, getting more than one of these software packages to work on the same machine is very difficult, and often prohibitively time-consuming, which is bad for both the community and the developers; for the community, since only a select few who are experts in radio astronomy and computing can do state of the art data reductions, limiting the scientific outputs of telescopes, and for developers since their tools do not get widespread usage.

4.1.1 Radio Astro PPA (*now Kern Suite*)

Recently there has been an effort, led by Gijs Molenaar³, to facilitate easy installation and co-existence of most radio astronomy related software packages. The project is called **Radio Astro**⁴ and provides neatly packaged versions of most legacy and novel radio astronomy packages which are easy to install. They are hosted on the Launchpad⁵ Personal Package Archive (PPA) and can be seamlessly installed on any *Long Term Support* (LTS) Ubuntu distribution. Although **Radio Astro** only supports LTS Ubuntu distributions, it has already had a significant impact in the community. For example, the $5 \times 10^6:1$ dynamic range radio maps obtained from the VLA observation of the 3C147 field [42] were enabled, in part, by the fact that multiple (novel and legacy) packages required to achieve this high dynamic range reduction could be installed on the same machine with relative ease. Moreover, with the emergence of cloud computing services such AWS and GCE, where one can launch VMs with an operating system of their choice, this service will become even more useful in the modern era.

Note that the Radio Astro project has since been re-branded and succeeded by the Kern Suite project⁶.

4.2 Container Technology

Container technology facilitates operating-system-level virtualisation that allows one to build self-contained computing environments (containers) for software applications,

³Rhodes University and University of Amsterdam

⁴<https://github.com/radio-astro>

⁵Launchpad is a web application that allows users to host, develop and maintain software packages online

⁶<http://kernsuite.info>

which can then be executed on a host machine with minimal interaction with the host's system settings and no interaction with the host's applications and libraries. This virtualisation is not new, and was already possible with Linux's `chroot`⁷, and BSD's `jail`⁸ virtualisation tools, but the widespread adoption of this technology was catalysed by the addition of operating-system-level virtualisation to the Linux Kernel in 2008 (version 3.8).

Container technology presents a neat solution to many of the problems that plague radio astronomy software, as applications that are executed in containers do not place specific requirements on the host; the host only needs a Linux kernel, and should be able to run containers. That is, an application can be wrapped up in a complete and isolated environment (with a custom operating system, system settings and libraries) and shipped to users who would then require minimal effort to execute said application. Note that, once configured, containers guarantee that the application always runs on the same environment (up to kernel-level settings). It is also important to note that containers are not virtual machines (VMs), as unlike a VM, which uses a *hypervisor* to emulate a whole operating system (and to some extent the hardware), containers use the host machine's Linux kernel, which not only makes containers much lighter but also gives them some clear advantages:

- Since there is no hardware emulation, the startup and shutdown times are very low (millisecond time scales).
- CPU and memory partitioning is not required, even though CPU and memory resources can be limited for container executions.
- Containers have direct access to low-level hardware, which is particularly useful for graphics processing unit (GPU) acceleration.

On the other hand, VMs have a higher level of isolation because of the hypervisor. Although very unlikely, a compromised container can affect the host and other containers, since the kernel is shared. In addition, containers have a deep level of authorisation as they require administration privileges. That said, this technology is still relatively new, and these issues are likely to be resolved as it evolves. Figure 4.1 shows the difference in architecture of these two technologies.

4.3 Docker

There are various implementations of container technology, such as `RKT`⁹, `Docker`¹⁰ and `LXC`¹¹, with Docker being the most popular. Docker is also the implementation

⁷https://wiki.archlinux.org/index.php/change_root

⁸<http://www.onlamp.com/pub/a/bsd/2003/09/04/jails.html>

⁹<https://github.com/coreos/rkt>

¹⁰<https://www.docker.com/>

¹¹<https://linuxcontainers.org/>

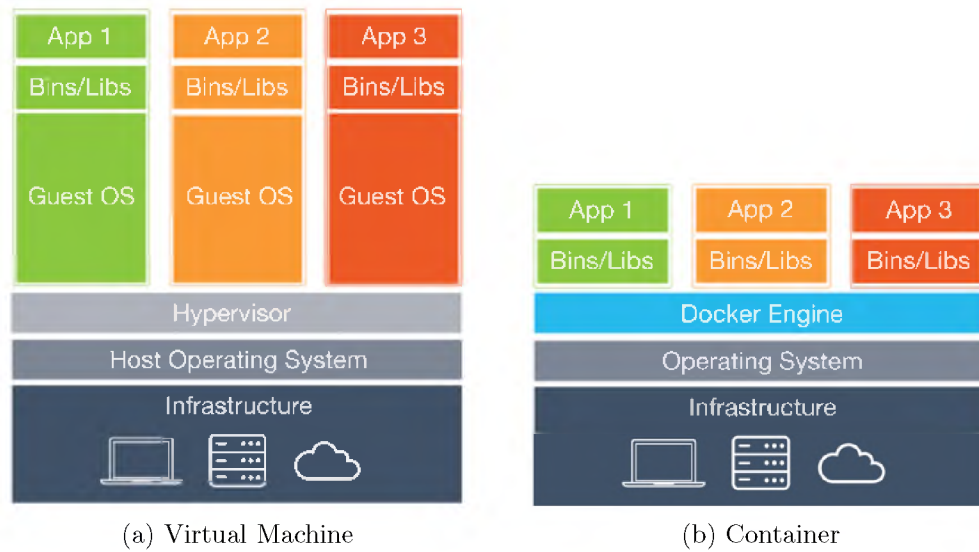


Figure 4.1: Schematic of the architecture of a virtual machine (left) and that of a container (right)[66]

that will be adopted in this work.

Docker offers a wide range of features that facilitate the configuration, deployment and manipulation of containers. It also uses a union file system, which allows files and directories of separate file systems to be transparently overlaid, as so-called *layers*¹², forming a common coherent file system.

4.3.1 Architecture

Docker uses a client-server architecture. The **Docker** client talks to the **Docker** daemon, which builds, runs, and distributes containers.

Docker Containers

As previously stated, a container holds everything that is needed for an application to run. **Docker** containers can be started, stopped, moved, and deleted. Each container is an isolated and secure application platform. Containers are the run component of **Docker**.

Docker Images

Docker images are the build components of **Docker**. These are read-only templates of containers, and containers are created from them. For example, a **Docker** image can contain an Ubuntu operating system with **MeqTrees** installed, then all containers

¹²<https://docs.docker.com/engine/userguide/storagedriver/imagesandcontainers>

```
FROM ubuntu:14.04
MAINTAINER Sphesihle Makhathini <sphemakh@gmail.com>

RUN apt-get update && \
    apt-get install -y \
    software-properties-common \
    python-software-properties && \
    add-apt-repository -ys ppa:radio-astro/main && \
    apt-get clean

RUN apt-get update && \
    apt-get install -y meqtrees

ENV MEQTREES_CATTERY_PATH /usr/lib/python2.7/dist-packages/Cattery

CMD /usr/bin/meqtrees-pipeliner.py
```

Figure 4.2: Dockerfile (Docker image recipe) for the MeqTrees software package.

created from that image can run **MeqTrees**. The images are built through a recipe called a *Dockerfile*¹³; see Figure 4.2 for an example. They can either be built locally, or pulled from the *docker-hub*; an online registry of **Docker** images with various operating systems and applications installed. **Docker** images can be shared and managed through the docker-hub. Alternatively, one can use their own registry to host and manage their **Docker** images. These images can also be versioned, therefore one has control over the architecture on which their packages are run, as well as the versions of the packages and their dependencies.

4.4 Stimela

stimela¹⁴ is a platform-independent radio interferometry scripting framework based on **Docker** and Python. In this framework, radio interferometry related tasks such as imaging, calibration and data synthesis are executed in **Docker** containers. In fact, within this framework the packages that perform these tasks are Python modules. Much like Oleg Smirnov’s **pyxis** package, **stimela** does not do any data processing, synthesis or analysis but merely offers a simple interface to packages that perform these tasks. However, unlike **pyxis** which requires multiple radio astronomy packages to be working on the same system, **stimela** only requires **Docker** and Python. Moreover, because of **Docker**, a **stimela** script runs the same way (in the same isolated environment) regardless of the host machine’s settings. The aim of *stimela* is to provide a user-friendly and modular scripting environment that gives general users easy access to novel radio interferometry calibration, imaging, and synthesis packages.

¹³<https://docs.docker.com/engine/reference/builder>

¹⁴stimela is the IsiZulu word for a train

Furthermore, as we will discuss in section 4.4.4, **stimela** offers the radio astronomy community, in particular modern radio observatories, a solution to problems concerning the reproducibility of scientific data products (discussed in detail in [54]). Note that the work in the rest of this chapter, namely the design of the **stimela** package, is the original work of the author.

4.4.1 Architecture

stimela is centred on two sets of **Docker** images, i) base images, which have the required software tools installed in them, and ii) very light-weight executor images (a.k.a *cab images*) based on the base images, these perform radio interferometry related tasks such as imaging, data synthesis, and calibration. The base images can either be built locally (on the host machine) or pulled from **Docker** hub, and the executor images are built locally. Figure 4.3 shows an overview of the **stimela** architecture described in this section.

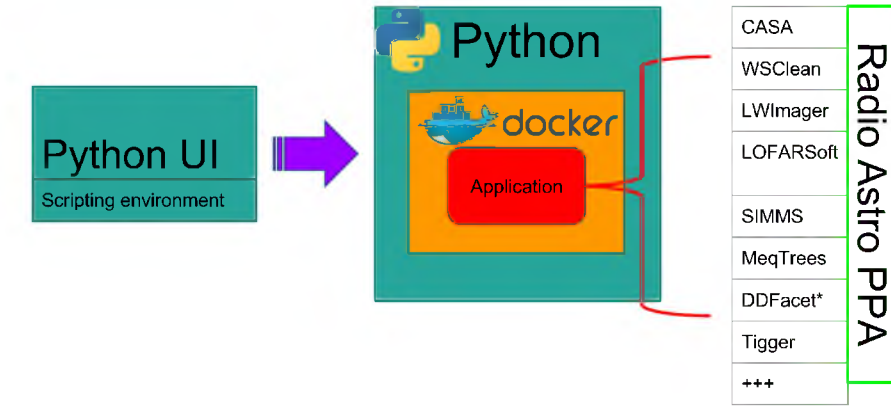


Figure 4.3: Schematic of the stimela architecture.

Base images

Base images are a suite of **Docker** images which have various radio astronomy related packages, including data synthesis and calibration packages such as **CASA** and **MeqTrees**, and imaging packages such as **lwimager** and **wsclean**. The image in Figure 4.2 is an example of a base image. Base images are maintained by the **Radio Astro** project on the **Docker** hub¹⁵, but any **Docker** image can be used as a base image.

¹⁵<https://hub.docker.com/u/radioastro/>

Executor (a.k.a cab) images

These Docker images are generally pre-loaded with Python scripts that perform a specified task (e.g calibrating a visibility dataset). A *stimela cab* image takes some input as well a set of instructions, performs some task, then returns the output. Figure 4.4 shows the work-flow of an executor image, and Figure 4.5 shows the Dockerfile for the *calibrator* image (described below). All executor images must follow this work-flow.



Figure 4.4: Structure of a *stimela* executor container.

```

1 FROM radioastro/meqtrees
2 MAINTAINER <sphemakh@gmail.com>
3
4 RUN mkdir -p /input /output # I/O directories
5 ADD src /code # source code that runs the given task
6
7 WORKDIR /code
8
9 CMD sh run.sh # execute task
  
```

Figure 4.5: Stimela executor image template for the *MeqTrees* based calibration task.

At the time of writing, these are the executor images that are available:

simms: Uses the *CASA* based *simms* tool to create empty visibility datasets (*CASA* Measurement Sets; MS)

autoflagger: Uses *aoflagger* to automatically flag RFI from radio data sets.

flagms: Uses *flag-ms* (Part of the *MeqTrees* package) to manually flag radio frequency interface (RFI) from radio data sets.

simulator: Uses *MeqTrees* to simulate visibilities into an MS

predict: Uses *lwimager* to predict visibilities from a FITS image.

calibrator: Uses **MeqTrees** to run a self-calibration solution

wsclean, casa, lwimager, MORESANE: These are for imaging and deconvolution. MORESANE only deconvolves dirty images.

sourcery: This uses the **sourcery** tool [56] (based on the PyBDSM source-finding tool) to extract sources from radio images.

The user also has the option of defining custom executor images, which can then be seamlessly plugged into a **stimela** script. This will be illustrated in Chapter 6.

4.4.2 Python backend

The **Python** backend comprises two Python modules. The first is a Python wrapper for Docker, and the second is the foundation of the **stimela** scripting environment.

Python wrapper for Docker

stimela interfaces with Docker through this wrapper. This is simply an interface to the **docker-engine**. Note that even though Docker has an official Python interface, **docker-py**¹⁶, it has significant inconsistencies with the **docker-engine** and has some missing features¹⁷, therefore, we decided to write our own docker interface based on the Python **subprocess** module. It can be used to run Docker commands for managing docker images such as:

docker pull: Pulls a Docker image from a remote registry (usually from the Docker hub)

docker build: Builds a Docker image given a Dockerfile

docker rmi: Deletes a Docker image

as well as container related commands such as:

docker create: Creates a Docker container (does not start it)

docker start: Starts an existing container

docker run: Creates and starts a container

docker stop: Stops a running container (terminates process running inside container)

docker rm: Deletes a docker container.

¹⁶<https://docker-py.readthedocs.io/en/stable>

¹⁷<http://blog.bordage.pro/avoid-docker-py>

This module also has functions that, through the **docker-engine**, keeps track of the standard output and standard error streams of containers, as well as the state of a container.

The Recipe module

stimela scripts are driven by the *Recipe* module. This module provisions and manages a set of **stimela** executor containers, and then executes them according to some recipe defined by the user. Each of the containers in the set corresponds to some radio interferometric task. This module has the following functions:

add: Adds a container (task) to a Recipe instance. The user can also define the task I/O flow for the task, as well as specify the computing resources (CPU and RAM) that should be used in executing the task.

run: Runs a set of containers according to a user-defined order – the default is to execute them in the order that they were added.

stop: Stops all running containers of a Recipe instance.

rm: Removes all containers in a Recipe instance.

Note that even though top level (at the level of imaging, calibration, etc.) parallelisation when reducing these data is difficult as the visibility data structure (MS) is generally both the input and output product of the these tasks, this module does allow for tasks to be executed in parallel.

4.4.3 User interface

The user interacts with **stimela** via the command line and through a Python script. The command line interface allows the user to manage **stimela** images and containers, as well as run and keep track of running scripts. The following commands are available:

help With no arguments, the help command prints out a summary of all the functions of all the other commands. To get help on a specific command, the user can give the name of that command as the argument.

pull: Pulls base images

build: Builds executor images

run: Runs a **stimela** script

ps: Lists running scripts

kill: Terminates a running script. Terminating a `stimela` script is not straightforward as it requires stopping and removing all containers as well as killing the Python process.

containers Lists all running executor containers

images Lists executor images.

The Python script, based on the Recipe module, is the primary way in which the user interacts with `stimela`. This is the prescription that defines the tasks (calibration, imaging, RFI flagging, etc.) to be performed on a given visibility data set. In simple terms, a `stimela` script (or recipe) is an ordered collection of executor containers. Each of these containers has to be added to a Recipe instance along with a set of options (Python dictionary; see lines 22-31 of sample code in Listing 4.1) that will be parsed as arguments to the task encapsulated by the container. See Listing 4.1 for a sample script; the full script is documented in Appendix A.

Listing 4.1: Sample `stimela` script. The full script can be found in Appendix A

```

1  # import stimela package
2  import stimela
3
4  # Recipe I/O configuration
5  INPUT = "input" # This folder must exist
6  OUTPUT = "output"
7  MSDIR = "msdir"
8  PREFIX = "stimela-example" # Prefix for output images
9
10 # MS name
11 MS = "meerkat_simulation_example.ms"
12
13 # Use the NVSS skymodel. This is natively available
14 LSM = "nvss1deg.lsm.html"
15
16
17 # Start stimela Recipe instance
18 recipe = stimela.Recipe("Simulation Example", # Recipe name
19                         ms_dir=MSDIR)         # Folder in which to find MSs
20
21 ## 1: Make empty MS
22 simms_dict = {
23     "msname" : MS,
24     "telescope" : "meerkat", # Telescope name
25     "direction" : "J2000,90deg,-45deg", # Phase tracking centre of
                                     observation

```

```

26     "synthesis" : 4,                # Synthesis time of observation
        [in hours]
27     "dtime"    : 5,                # Integration time [in seconds]
28     "freq0"    : "750MHz",         # Start frequency of observation
29     "dfreq"    : "1MHz",           # Channel width
30     "nchan"    : 10                 # Number of channels
31 }
32
33 # Add process to recipe
34 recipe.add("cab/simms",             # Executor image to start container
        from
35     "simms_example",               # Container name
36     simms_dict,                    # Parameters to parse to executor
        container
37     input=INPUT,                   # Input folder
38     output=OUTPUT,                 # Output folder
39     label="Creating MS")           # Process label

```

Then, the script may be run on the command line as follows:

```
$ stimela run <script name>
```

4.4.4 Reproducibility

A cornerstone of the scientific method is the fact that scientific experiments are repeatable and the findings independently reproducible. This basic tenet is becoming progressively difficult to maintain owing to the large amounts of data produced by astronomy observatories and the high complexity of the algorithms and corresponding software applications. As eloquently explained in [54], the process of creating science data products in astronomy can be cumbersome and unreliable for a number of reasons including; i) the reduction scripts (or pipelines) used are so complex that only a few individuals can understand them; ii) these pipelines are often poorly documented; iii) the software applications required to run the pipelines only work in suitably customized (operating system, library versions, system settings) computing environments. **stimela** offers a natural solution to two of these issues; i) the simple scripting environment ensures that anyone with a basic knowledge of Python can understand the scripts; and ii) the developers of these data reduction (or simulation) pipelines can guarantee that the constituent tasks of a given pipeline are always executed on the same computing environment inside the container (customised to the developer's specification) without imposing any further customization on the host system.

Another very useful feature of **stimela** is that it keeps track of all the docker images (along with image versions) used in a given script. That is, a **stimela** script

can be replicated (bit-perfect) on an arbitrary computer system (given adequate resources) by an arbitrary user, given only the log-file from the earlier run. This level of reproducibility and simplicity should lead to robust cross-platform tests of reduction pipelines and data-processing tools. More importantly, it should lead to transparent and repeatable scientific data processing in radio astronomy.

4.5 Conclusions

The **stimela** framework offers a single interface in which to implement end-to-end data synthesis and reduction pipelines. Through this framework, the user has access to standard software packages such as **CASA** and **MeqTrees**, and novel packages such as **wsclean** without having to build or install them. More importantly, one can seamlessly add other packages, and since each step (package) in a **stimela** recipe (pipeline) runs in its own isolated environment, one need not worry about conflicts when incorporating new packages. This makes **stimela** uniquely powerful and flexible. With the increasing sophistication and accessibility of cloud computing facilities such as AWS and GCE, coupled with tools such as **docker-machine**¹⁸, **stimela** opens up interesting possibilities when it comes to running large-scale radio interferometric simulations and reductions (as we show in Chapter 5).

Since **stimela** is based on **Docker**, one requires root privileges to use it. However, given the development of **Docker** up to this point, we expect that this restriction will be lifted soon.

The full documentation of the **stimela** package, as well as tutorials, can be found at <https://github.com/SpheMakh/Stimela/wiki>.

In the next chapter we present a **stimela** based simulations framework aimed at quantifying the scale-dependent sensitivity of the SKA1-MID telescope, and in Chapter 6 we use this framework to reduce VLA data.

¹⁸<https://docs.docker.com/machine>

Chapter 5

SKA1-MID Sensitivity and Imaging Performance

The first phase of the mid-frequency SKA telescope (SKA1-MID) that will be built in the Karoo desert in South Africa will be a 197-dish (including the 64 MeerKAT¹ dishes) telescope with a frequency coverage of 0.35 to 13.8 GHz. Some basic specifications of the telescope are presented in Table 5.1. The science case covers a broad range of science areas including galaxy evolution and cosmology [33], fundamental physics questions such as the precise nature of gravity in extreme gravity conditions and the origin of cosmic magnetism [33, 58], and astrophysics [27]. Some of the science cases are shown in Table 5.2.

Large-scale simulations are an ideal application for radio astronomy computing in a cloud environment, as the necessarily large volumes of data can be rapidly generated and processed in situ, with very little external I/O requirements. For this reason, we carried out these simulations on the AWS cloud computing system. The use of these computer resources was funded by the *SKA/AWS Astro-Compute in the Cloud*² initiative.

In this chapter we use a `stimela` based simulations framework to study the sensitivity and imaging performance of this telescope.

5.1 The SKA1-MID antenna layout

The SKA1-MID antenna layout [18] consists of a dense core of antennas, as well as antennas laid out in three spiral arm-like shapes. The density of the core decreases with radial distance, with the vast majority of the antennas falling within a square kilometre and only a handful antennas at the extremes of the core (about 4 km from the core). The antennas in the spiral arms are laid out in a logarithmic distribu-

¹See the beginning of Chapter 3 for details about MeerKAT

²<https://www.skatelescope.org/ska-aws-astrocompute-call-for-proposals>

Table 5.1: SKA1-MID frequency band information and corresponding SEFD values (at the time of writing) [3].

Band	Band width [MHz]	Band Centre [GHz]	SEFD [Jy]
1	700	0.7	673
2	810	1.4	400
3	1400	2.4	400
4	2380	4.0	441
5	9200	9.2	528

tion. This layout is shown in Figure 5.2, and has a maximum baseline of 157 km. Now, with the majority of the antennas within a square kilometre, SKA1-MID is not only an excellent machine for science goals that require detailed study of transient astrophysical objects or phenomena [58], but it also has enough long baselines to be a very good imaging instrument [20, 40]. However, the baseline distribution remains highly biased towards the shorter baselines (see Figure 5.3), resulting in a naturally weighted synthesized beam that has very broad wings (see Figure 5.5), and therefore makes deconvolution difficult. Also, as shown in Figure 5.7, the sensitivity of the instrument decreases by up to a factor of three when using uniform instead of natural *uv*-weighting. For this reason we have decided to study the sensitivity of the telescope as function of angular scale.

5.1.1 A case for a decreasing the SKA1-MID maximum baseline

The maximum baseline of an interferometer array is an important parameter. From a science point of view, it determines the maximum angular resolution of the array, and therefore the depth (noise level) at which individual sources can be distinguished, the source confusion limit. From a data-processing point of view, the maximum baseline determines the correlator dump times (which must be high enough to avoid time and bandwidth smearing), and the pixelisation of the resulting images (hence the image sizes). From this vantage point, and from an economic perspective, the maximum baseline should be chosen such that it accommodates the science case, while not scaling up the costs and technical complexities.

The theoretical maximum baseline required (5.1) to accommodate the highest angular resolution science case is about 137 km (see Figure 5.2). However, in practice, the resolution of a radio synthesis image depends on the visibility weighting (see section 2.2.1), with uniform weighting usually chosen when measuring the actual angular resolution of an interferometer.

In this chapter, we not only study the resolution and sensitivity of the SKA1-MID antenna layout, but also compare it to two alternative layouts that have shorter maximum baselines compared to the proposed SKA1-MID layout. The layouts under

Table 5.2: SKA1-MID science goals that require the highest angular resolution – sorted in order of required maximum baseline. The full science case can be found in [3]

ID	Goal	ν [GHz]	θ ["]	B_{\max} [km]
SKA1-SCI-11	Measurement of the magnetic field structure and its evolution its relation to gas in a large number of galaxies, active galactic nuclei, galaxy clusters and intergalactic filaments.	0.9-3	0.5	137
SKA1-SCI-8	Resolved studies of the morphology of galaxies, deep fields and lensing clusters	0.9-1.8	0.5	137
SKA1-SCI-16	Mapping of the growth of grains through the centimetre-sized regime both inside and outside the snow-line in cluster of proto-planetary disks at a distance of 100 pc.	12	0.04	129
SKA1-SCI-5	Detect sample of 1000 HI absorption systems with a median redshift $\langle z \rangle \geq 2$.	0.45-0.9	2.0	69
SKA1-SCI-14	Detect a sufficient density of sources to permit sensitive constraints to be placed on the dark energy equation of state using weak gravitational lensing.	1.4-1.7	0.7	63

consideration are:

SKA1MID This is the SKA1-MID antenna layout design proposed by the SKA organisation [4]. The maximum baseline of this array is 157 km.

SKA1MID-137 This is a version of the SKA1MID layout with the dishes in the spiral arms repositioned such that the maximum baseline of the resulting array is 137 km. 137 km is the theoretical maximum baseline required by the highest resolution SKA1MID science case (see Figure 5.2).

SKA1MID-120 This is a version of the SKA1MID layout with the dishes in the spiral arms repositioned such that the maximum baseline of the resulting array is 120 km.

5.2 SKA1-MID performance assessment

In this section we compare the sensitivity and imaging performance of the SKA1-MID layouts described in section 5.1.1

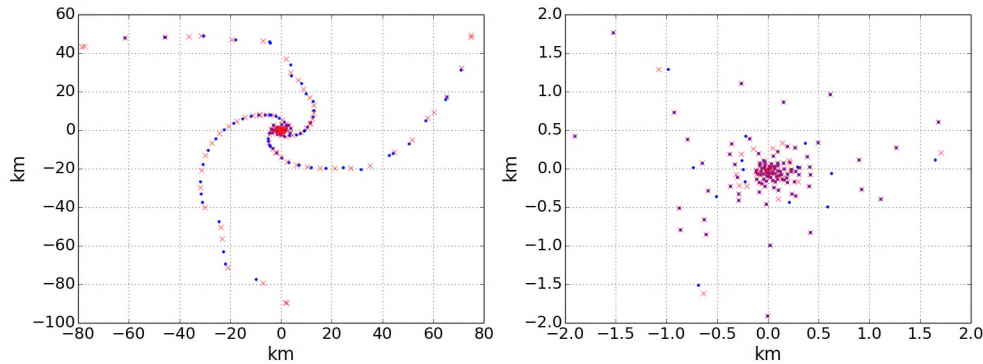


Figure 5.1: The SKA1-MID antenna layout by the SKA organisation (red crosses), which has a maximum baseline of 157 km and our alternative layout (SKA1MID-137), which has a maximum baseline of 137 km.

Telescope simulations and imaging

The `simms` tool was used to simulate 2-hour single-channel observations at the band centres of the SKA1-MID telescope. The observations are at declination -30° with a 60 s integration time. We then inject thermal noise into the simulated visibility data using the `casacore`-based `pyrap` module. The noise per visibility is estimated using the SEFD values in Table 5.1 via the radiometer equation (2.37). The datasets are

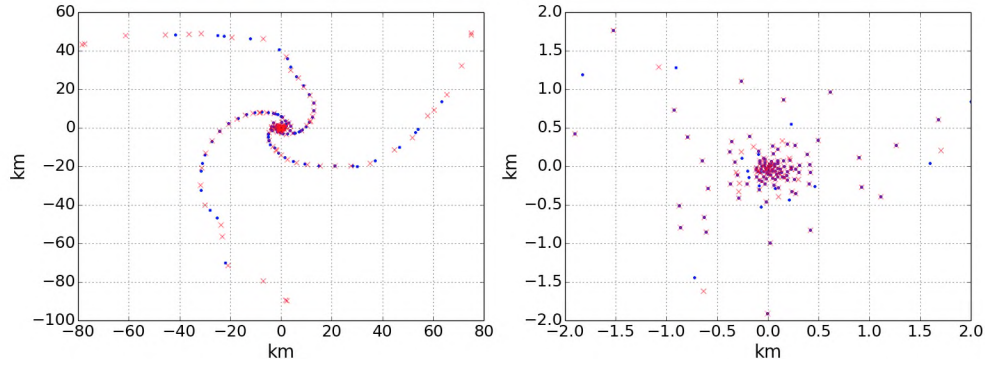


Figure 5.2: The SKA1-MID antenna layout by the SKA organisation (SKA1MID; red crosses), which has a maximum baseline of 157 km and our alternative layout (SKA1MID-120; blue dots), which has a maximum baseline of 120 km.

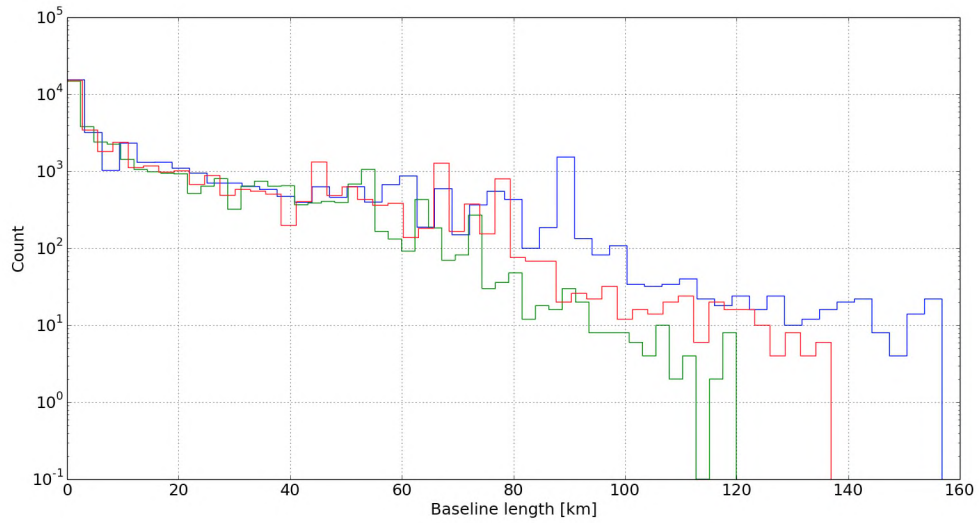


Figure 5.3: Histograms of the baseline distribution of the the proposed SKA1-MID antenna layout (SKA1MID; blue), and our two alternative layouts SKA1MID-137 (green) and SKA1MID-120 (red), which have a maximum baseline length of 137 km and 120 km respectively; compared to the SKA1MID layout which has a maximum baseline of 157 km.

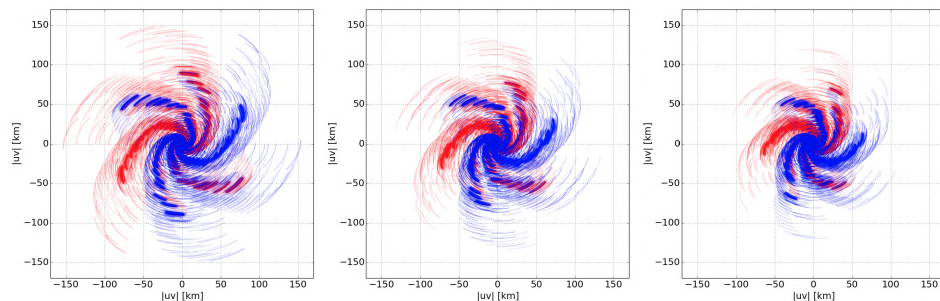


Figure 5.4: uv -Coverage plots from 2-hour simulated observations with the SKA1-MID telescope using the (left to right) SKA1MID, SKA1MID-137, and the SKA1MID-120 antenna layouts.

imaged using the `lwimager` imaging tool (derived from `casarest`) [65]. The simulations in this chapter are all noise-only simulations, hence all sensitivity measurements will be performed on the dirty images, because there is no sky signal to reconstruct.

This simulation pipeline, implemented within `stimela` package, can be found at http://bit.ly/stimela_weights.

5.2.1 Resolution and sensitivity

In this section we refer to the sensitivity as the RMS value of the pixel amplitudes in the dirty images of the noise simulations, and the resolution as the mean value of the FWHM of the PSF cross-sections along the Right Ascension and Declination axes.

In Figure 5.6, sub-figure (a), we see that the resolving performance of the SKA1MID layout is only about 1.08 and 1.48 times better than the SKA1MID-137 and the SKA1MID-120 layouts respectively, when using Briggs weighting with a robustness parameter of 2. However, we note that the highly irregular shape of these PSFs makes these numbers deceptive as the FWHM is not a good indicator of PSF size. The sizes in sub-figure (b) (robustness parameter of 0), which show that the SKA1MID layout is better by a factor of 1.19 and 1.47 compared to the SKA1MID-137 and SKA1MID-120 respectively, also suffer from the problem. The PSFs made using Briggs weighting, using a robustness parameter of -2, sub-figure (c), have much better shapes and the FWHM size is a good indicator of the resolving performance. This figure shows that the resolving performance of the SKA1MID layout is only better by a factor of 1.17 and 1.37 compared to the SKA1MID-137 and the SKA1MID-120 layouts respectively.

Figure 5.7 shows that the noise properties, on all five bands, of our two alternative antenna layouts are very similar to the SKA1MID layout when using Briggs weighting with robustness parameters 2 and 0. Even for a robustness parameter of -2, the SKA1MID layout is only better by a factor of about 1.1 compared to the SKAMID-

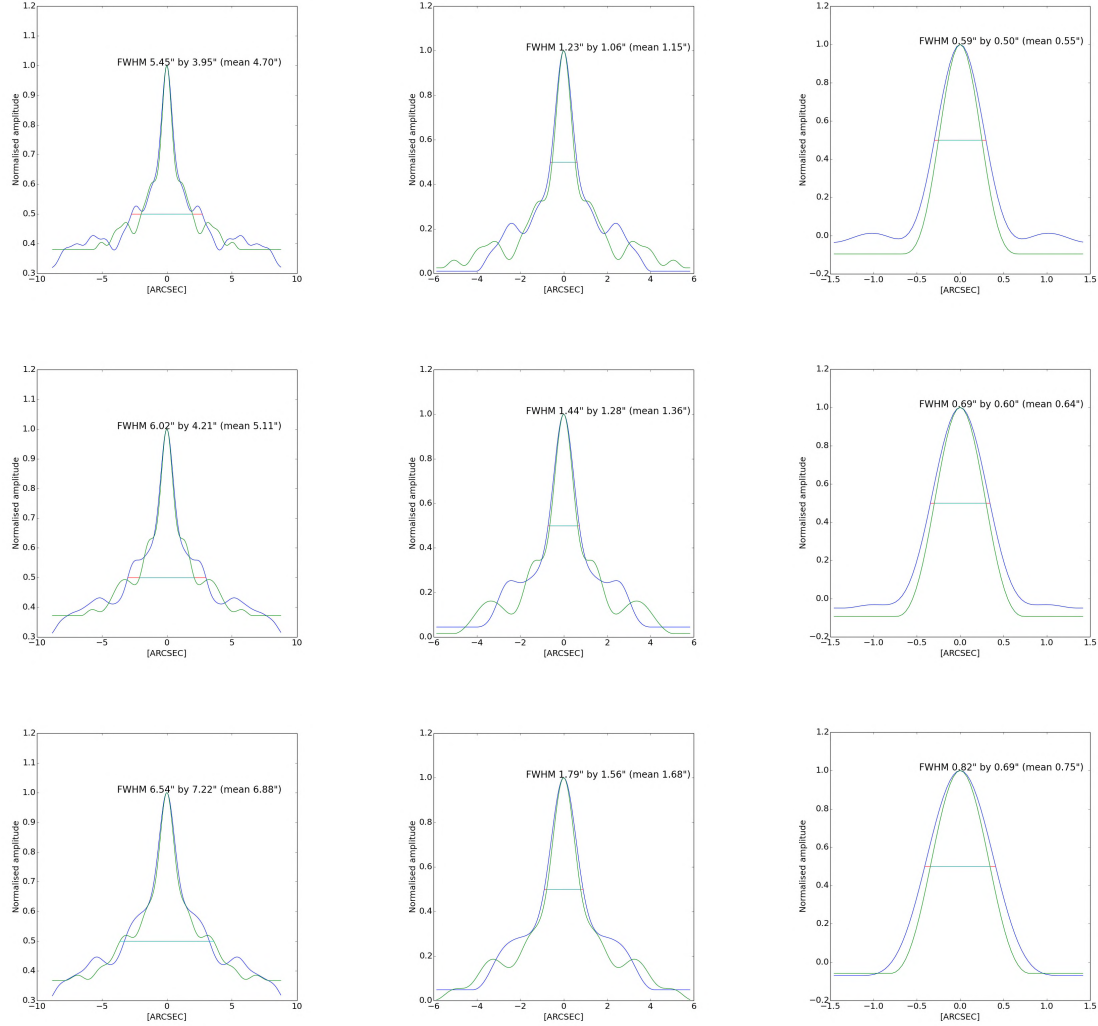


Figure 5.5: PSF cross-sections, along the right ascension (green) and declination (blue) axes, from SKA1MID simulated 2-hour observations at 700 MHz using Briggs weighting with a robustness parameter of 2 (column 1), 0 (column 2), and -2 (column 3), for the SKA1MID, SKA1MID-137, and SKA1MID-120 antenna layouts.

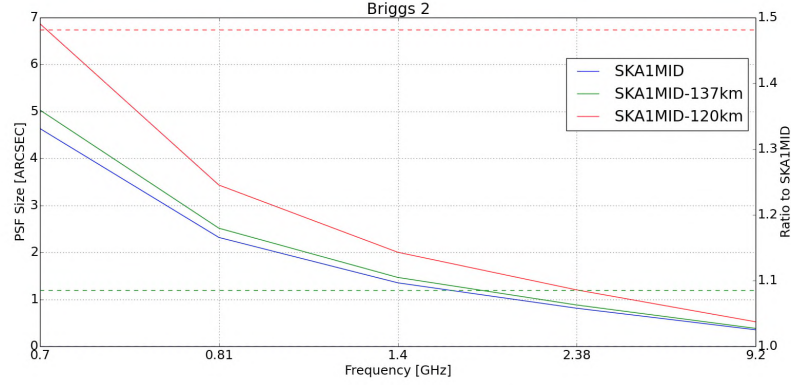
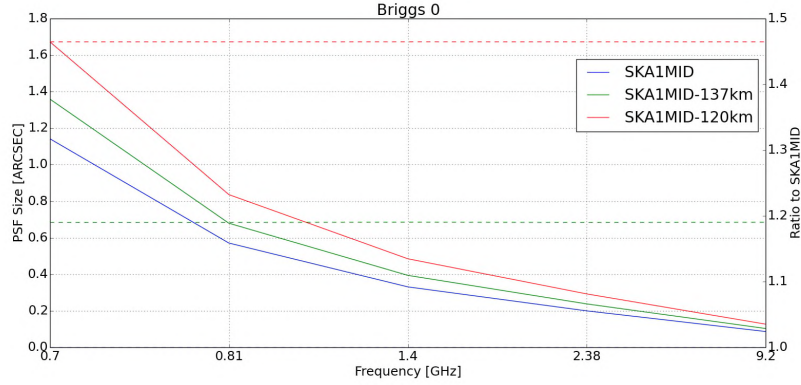
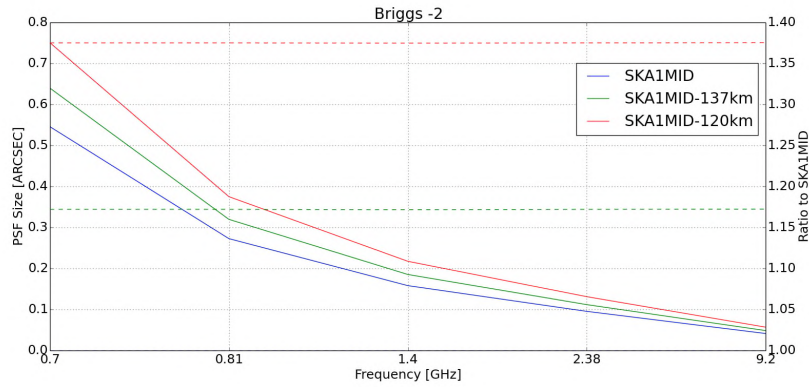
(a) Briggs uv -weighting, robust=2(b) Briggs uv -weighting, robust=0(c) Briggs uv -weighting, robust=-2

Figure 5.6: PSF sizes at the five band centres of the proposed SKA1-MID antenna layout (SKA1MID; blue), which has a maximum baseline of 157 km, ii) a scaled-down version of the SKA1MID antenna layout which has a maximum baseline of 137 km (SKA1MID-137; green), and iii) a scaled version of the SKA1MID layout which has a maximum baseline length of 120 km (SKA1MID-120; red). This is done for Briggs weighting with a robustness parameters of -2, 0 and 2.

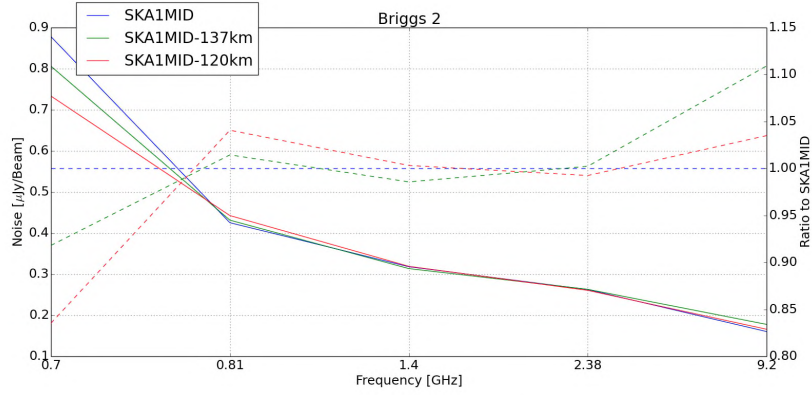
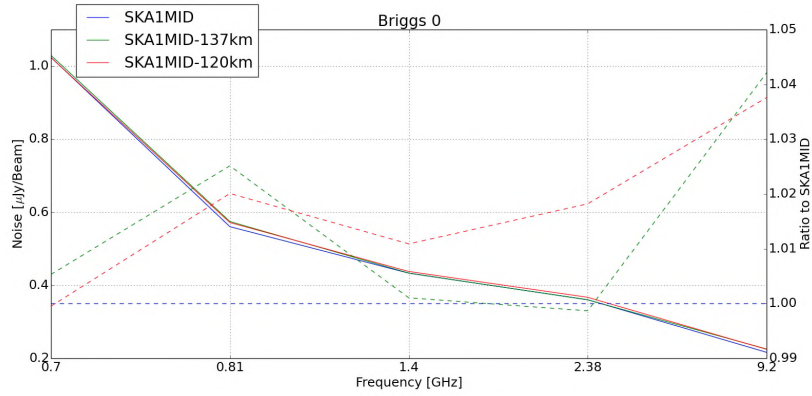
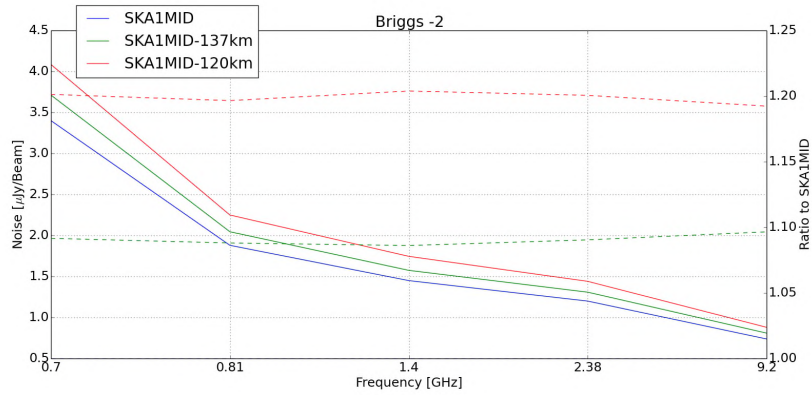
(a) Briggs uv -weighting, robust=2(b) Briggs uv -weighting, robust=0(c) Briggs uv -weighting, robust=-2

Figure 5.7: Image noise properties of the SKA1-MID telescope at the five band centres for the proposed antenna layout (SKA1MID; blue), which has a maximum baseline of 157 km, ii) a scaled-down version of the SKA1MID antenna layout which has a maximum baseline of 137 km (SKA1MID-137; green), and iii) a scaled version of the SKA1MID layout which has a maximum baseline length of 120 km (SKA1MID-120; red). This is done for Briggs weighting with a robustness parameters of -2, 0 and 2.

137 layout and about 1.2 times better than the SKA1MID-120 layout.

5.2.2 SKA1-MID scale-dependent sensitivity

The natural sensitivity at specific angular scales (scale-dependent sensitivity) for an interferometer array can be determined by making noise images using only visibilities corresponding to those angular scales. In practice, only imaging visibilities corresponding to certain angular scales leads to a PSF with high sidelobes since this is equivalent to multiplying the visibility plane with a suitably sized boxcar function. These high sidelobes can be reduced by using the frequency response of the Butterworth function [8], which does not have the discontinuities (sharp corners) that lead to high sidelobes, instead of the boxcar function.

In this section we gauge this scale-dependent sensitivity for the the five SKA1-MID frequency bands.

Scales of interest

The scales of interest are derived from the maximum baseline, $B_{\max} = 157$ km, of the SKA1-MID antenna layout. For each frequency (band centre), we define the maximum resolution element as

$$R_i = \left\{ \frac{\lambda_i}{B_{\max}} \right\}. \quad (5.1)$$

The scales of interest are then $(m_j, m_{j+1})R_i$, where m_j is a positive multiplier. In this case, we choose $m_j = \{1, 3, 9, 27, 81, 243, 729, 3600\}$. In Table 5.3, R_i values are presented for $m_0 = 1$, and Table 5.14a shows our scales of interest in units of R_i .

Table 5.3: SKA1-MID maximum resolution (λ/B_{\max}) at the five band centres.

Frequency [GHz]	R_i ["]
0.7	0.56
1.4	0.28
2.4	0.16
4.0	0.098
9.2	0.043

Once the angular scales of interest have been determined, the visibility data are weighed with a Butterworth function to select visibility points that correspond to those angular scales before imaging. The visibility weights for the scales of interest are shown in Figure 5.9 for Band 1. This weighting is achieved through a Python script that uses the MS manipulation tool, `pyrap`, and the scientific computing tool,

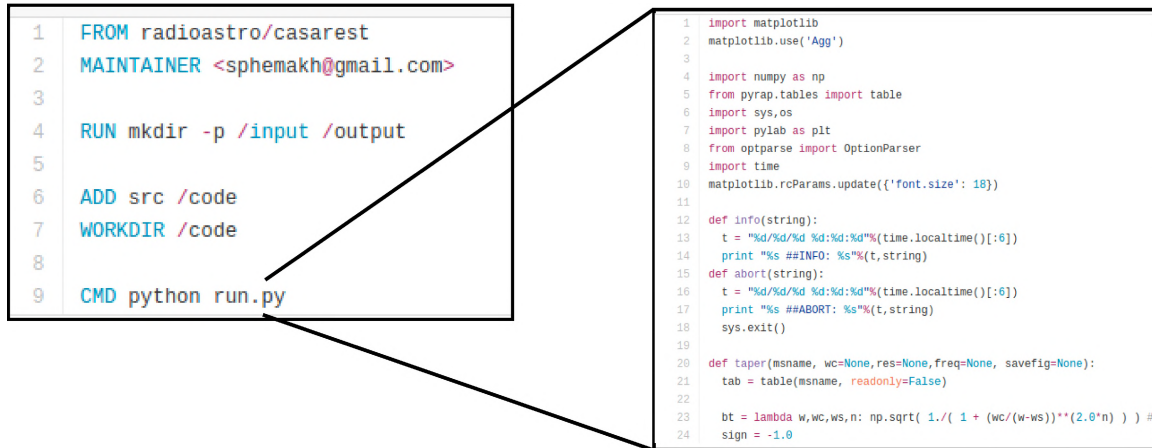


Figure 5.8: Illustration of the **stimela** executor image that applies visibility weights that select specific angular scales.

numpy³. We then create a **stimela** executor image that contains this script, and plug it into the simulation recipe.

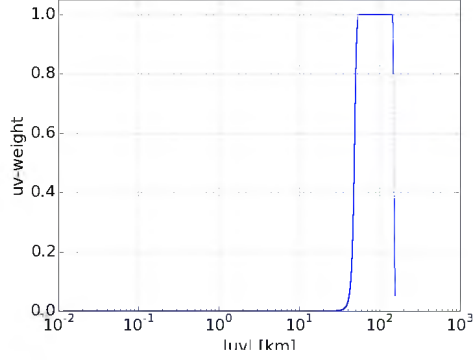
Figures 5.10-5.14 show the scale-dependent sensitivity of the three layouts under consideration. There are a few things to note from these plots: i) The sensitivity across all the bands for all three layouts is similar, except in the highest resolution bin (bin 0), which is not surprising, since the only differences in the layouts are at the longest baselines. ii) In the highest resolution bin there is a significant drop (up to 46%) in sensitivity for the SKA1MID-120 layout, while the SKA1MID-137 layout is only less sensitive by up to 16%. ii) The SKA1-MID telescope is most sensitive to structures that are between 0.76' and 6.84' in Band 1, which corresponds to baselines between 0.22 and 1.9 kilometres. In the same band, the sensitivity is also very good on angular scales between 1.69'' and 5.06'', in fact the SKA1MID-120 layout has the best sensitivity in this resolution. However, there is a significant drop-off in sensitivity on the resolution bin corresponding to the very large scales, 6.48' to 33'', since there are not many baselines corresponding to spacings of 20-100 m.

This shows that there is room to decrease the maximum baseline of the SKA1-MID telescope without compromising the sensitivity or the angular resolution performance of the telescope, even on the highest angular scales. As an extension of this work, we should consider layouts with a smaller maximum baseline, but which also optimise the distribution of antennas along the spiral arms such that the sensitivity on the longest baselines is maximised without severely compromising the sensitivity elsewhere. Another important constraint in this optimisation will be the telescope science goals.

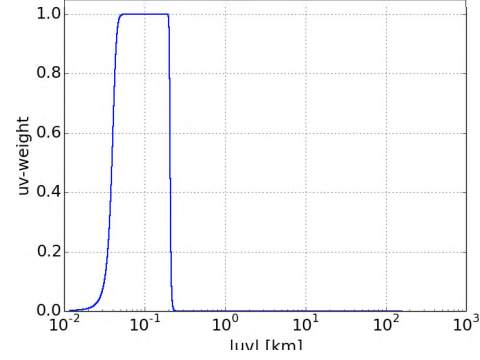
³<http://www.numpy.org>

Angular scales [Arcsec]	0.56-1.69	410.19-2025.64
Resolution bin	0	5

(a) Angular scales



(b) Res. bin 0



(c) Res. bin 5

Figure 5.9: (a) and (b) shows the uv -weights applied to select uv -points at specific angular scales. The corresponding angular scales are shown in Table (a). The uv -points are from a 2-hour simulated observation using the SKA1-MID antenna layout, and the weights are generated from the frequency response of the Butterworth function.

5.3 Discussion

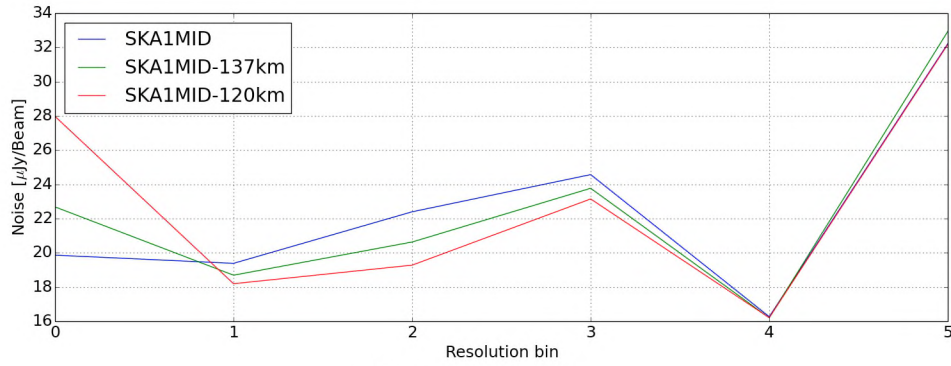
The total size of the data generated (measurement sets, PSF and noise images) was only 102 Gb. Each of the sensitivity and resolution simulations (section 5.2.1) had a run time of about 7 minutes on an AWS virtual machine with 256 Gb RAM and 32 CPUs. The simulations iterate over the three weighting schemes and five frequency bands for each antenna layout, i.e., taking $7 \text{ minutes} \times 3 \times 5 = 105 \text{ minutes}$ for each layout. The scale-dependent sensitivity simulations (section 5.2.2) had a run time of about 5 minutes on the VM. In this case, for each antenna layout, we iterate over five frequency bands and five angular resolution bins, hence taking $5 \text{ minutes} \times 5 \times 5 = 125 \text{ minutes}$ for each antenna layout. This is not very computationally expensive, because effects such time and bandwidth smearing, which require fine time and frequency resolution, can be ignored when making noise-only simulations. This is very encouraging for building exhaustive frameworks to quantify the performance of radio telescope designs.

5.3.1 Telescope design in the modern era

The design of radio interferometers is an important topic in radio astronomy at the moment. But the lack of flexible and reliable frameworks to competently explore the parameter space of cutting edge engineering, software and scientific exploration while

Angular scales [Arcsec]	0.56-1.69	1.69-5.06	5.06-16.32	16.32-45.58	45.58-410.19	410.19-2025.64
Resolution bin	0	1	2	3	4	5

(a) Angular scales

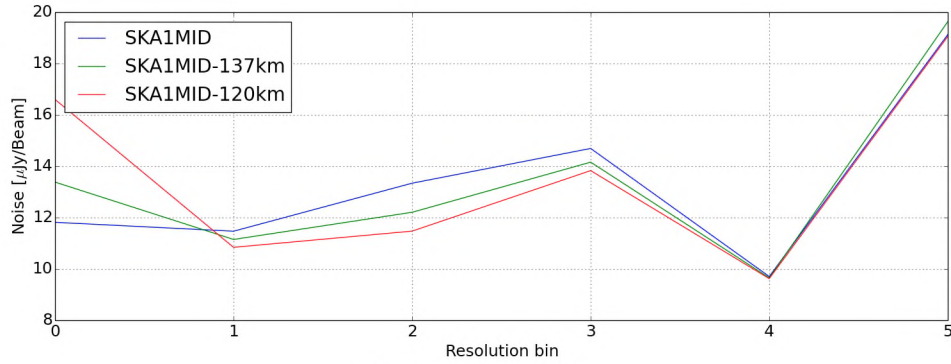


(b) Noise properties

Figure 5.10: The natural sensitivity as a function of angular scale for the SKA1-MID telescope from a 2-hour simulation centred at 700 MHz (Band 1) with a bandwidth of 700 MHz. The simulations were performed using three antenna layouts: i) the SKA1MID layout (blue), which has a maximum baseline of 157 km, ii) a scaled-down version of the SKA1MID antenna layout which has a maximum baseline of 137 km (SKA1MID-137; green), and iii) a scaled version of the SKA1MID layout which has a maximum baseline length of 120 km (SKA1MID-120; red).

Angular scales [Arcsec]	0.28-0.84	0.84-2.53	2.53-8.16	8.16-22.79	22.79-205.10	205.10-1012.82
Resolution bin	0	1	2	3	4	5

(a) Angular scales

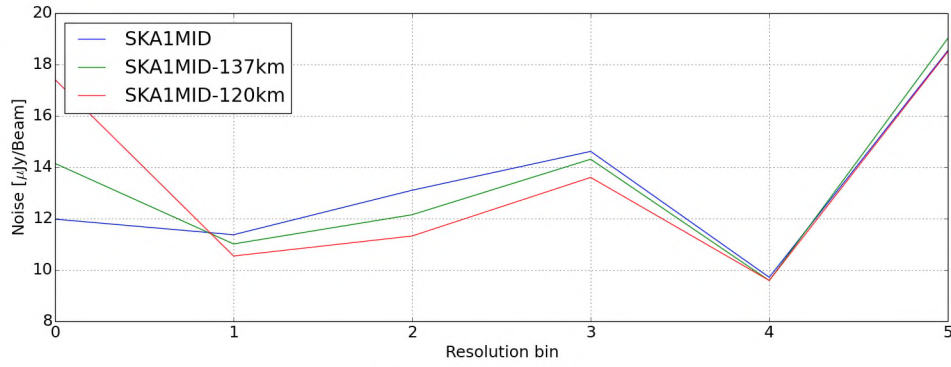


(b) Noise properties

Figure 5.11: The natural sensitivity as a function of angular scale for the SKA1-MID telescope from a 2-hour simulation centred at 1.4 GHz (Band 2) with a bandwidth of 810 MHz. The simulations were performed using three antenna layouts: i) the SKA1MID layout (blue), which has a maximum baseline of 157 km, ii) a scaled-down version of the SKA1MID antenna layout, which has a maximum baseline of 137 km (SKA1MID-137; green), and iii) a scaled version of the SKA1MID layout which has a maximum baseline length of 120 km (SKA1MID-120; red)

Angular scales [Arcsec]	0.16-0.49	0.49-1.48	1.48-4.76	4.76-13.29	13.29-119.64	119.64-590.81
Resolution bin	0	1	2	3	4	5

(a) Angular scales

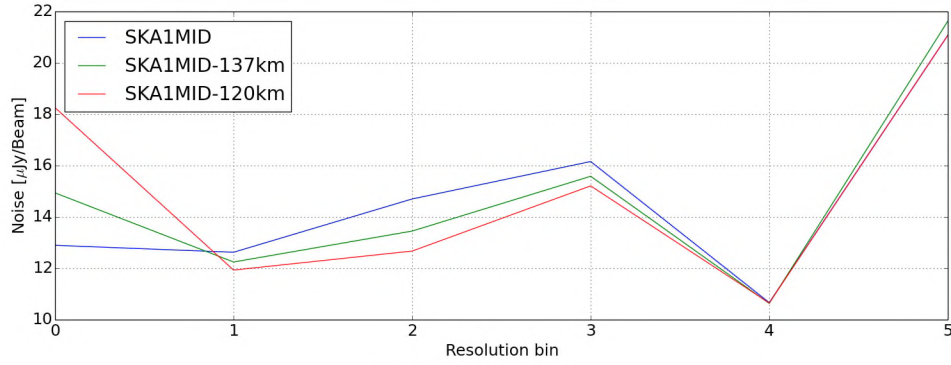


(b) Noise properties

Figure 5.12: The natural sensitivity as a function of angular scale for the SKA1-MID telescope from a 2-hour simulation centred at 2.4 GHz (Band 3) with a bandwidth of 1400 MHz. The simulations were performed using three antenna layouts: i) the SKA1MID layout (blue), which has a maximum baseline of 157 km, ii) a scaled-down version of the SKA1MID antenna layout, which has a maximum baseline of 137 km (SKA1MID-137; green), and iii) a scaled version of the SKA1MID layout which has a maximum baseline length of 120 km (SKA1MID-120; red)

Angular scales [Arcsec]	0.10-0.30	0.30-0.89	0.89-2.86	2.86-7.98	7.98-71.78	71.78-354.49
Resolution bin	0	1	2	3	4	5

(a) Angular scales

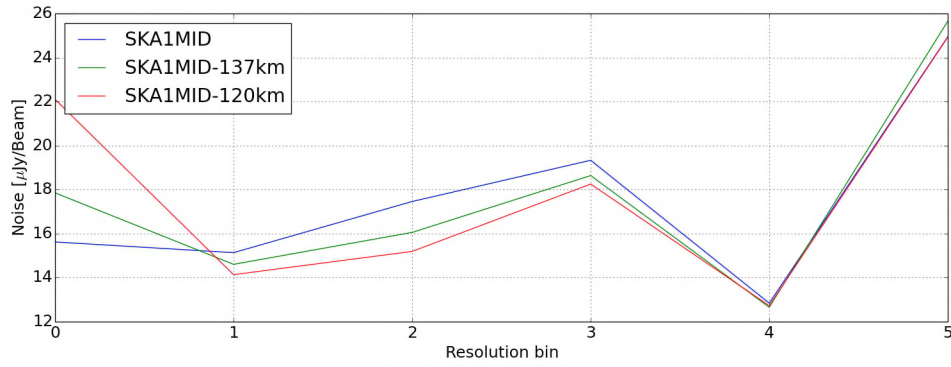


(b) Noise properties

Figure 5.13: The natural sensitivity as a function of angular scale for the SKA1-MID telescope from a 2-hour simulation centred at 4 GHz (Band 4) with a bandwidth of 2800 MHz. The simulations were performed using three antenna layouts: i) the SKA1MID layout (blue), which has a maximum baseline of 157 km, ii) a scaled-down version of the SKA1MID antenna layout, which has a maximum baseline of 137 km (SKA1MID-137; green), and iii) a scaled version of the SKA1MID layout which has a maximum baseline length of 120 km (SKA1MID-120; red)

Angular scales [Arcsec]	0.04-0.13	0.13-0.39	0.39-1.24	1.24-3.47	3.47-31.21	31.21-154.12
Resolution bin	0	1	2	3	4	5

(a) Angular scales



(b) Noise properties

Figure 5.14: The natural sensitivity as a function of angular scale for the SKA1-MID telescope from a 2-hour simulation centred at 9.2 GHz (Band 5) with a bandwidth of 9200 MHz. The simulations were performed using three antenna layouts: i) the SKA1MID layout (blue), which has a maximum baseline of 157 km, ii) a scaled-down version of the SKA1MID antenna layout, which has a maximum baseline of 137 km (SKA1MID-137; green), and iii) a scaled version of the SKA1MID layout which has a maximum baseline length of 120 km (SKA1MID-120; red)

trying to minimize costs means that the design is sometimes left to ad hoc arguments and techniques that cannot be independently verified. As stated at the end of chapter 3, this is largely due to the difficulties involved in creating a software environment that allows for thorough and systematic exploration of various design decisions in a transparent and verifiable way. Software applications such as `stimela`, by easing access to a diverse set of radio interferometry packages that can be combined to produce end-to-end telescope simulations, open the door to more transparent, innovative and efficient telescope designs. For example, one could use `stimela` to design a system that rates and compares the performance of different antenna layouts for an interferometer. The system could then be opened for contributions from the community via an online interface, and the antenna layout submissions rated according to a given set of metrics which could include the impact on science goals, sensitivity, angular resolution, calibratibility, data processing and storage requirements as well as financial costs.

5.3.2 Conclusions

We have used a `stimela` based simulations framework to show that the SKA1-MID antenna layouts with a maximum baseline of 137 km and 120 km (SKA1MID-137 and SKA1MID-120 respectively) have similar resolving and sensitivity performance on each of the five proposed frequency bands compared to the antenna layout proposed by the SKA organisation which has a maximum baseline of 157 km. Even when looking at the sensitivity as a function of angular scale, the loss in sensitivity at the highest angular scales for the SKA1MID-137 layout is about 16% across all the bands. We showed that the SK1MID telescope is most sensitive (natural uv -weighting) to angular scales between $0.76'$ to $6.84'$, but also has very good sensitivity on angular scales between $1.69''$ and $5.06''$ in the first band. Similar behaviour is seen on all the bands.

This work shows how different telescope designs can be compared and quantified in a systematic fashion. In the next chapter, we will show how `stimela` can also be used to calibrate and image a VLA dataset.

Chapter 6

A Platform-Independent Data Reduction Script

In this chapter we use `stimela` to reduce a VLA dataset. The dataset in question is a 21-hour observation of the field around the source 3C147 at L-Band with the VLA telescope in its BnA¹, C and D configurations. 3C147 is a compact steep spectrum Quasar at redshift 0.545 [53, 69]. The data were taken as part of an initiative to observe this field at L-Band by Perley² and Rhodes University. The total bandwidth of the data is 640 MHz³, which has been partitioned into two sub-bands; i) a low (LO) band with a bandwidth of 256 MHz and centred at 1.267GHz, and ii) a high (HI) band which has a bandwidth of 384 MHz and is centred at 1.587 GHz. These two datasets were reduced separately and the resulting reconstructed images combined to form the final reconstructed image from the entire bandwidth.

The field around 3C147 is a good demonstration field for data reduction packages and scripting frameworks, because 3C147 is very bright (about 22.58 Jy at L-Band) and ensures a high signal-to-noise ratio (SNR) for the calibration solutions, while most of the surrounding field sources are much fainter point-like sources. In fact the highest dynamic range (DR) images from both the WSRT [61, $1.6 \times 10^6 : 1$] and VLA [42, $5 \times 10^6 : 1$] telescopes are from observations of this field.

6.1 Reduction strategy

The data have been pre-calibrated RFI, and the bandpass and absolute flux scale have been calibrated) by Perley using the [46, AIPS] software package. Our reduc-

¹BnA is a hybrid configurations in which the antennas on the east and west arms are in the B configuration, but those on the north arm in the extended A configuration in order to enhance our view of sources in the southern sky (<http://www.vla.nrao.edu/genpub/configs/>).

²NRAO and Rhodes University

³The full bandwidth is 1024 MHz, but a large chunk of it is contaminated by RFI. 640 MHz is the portion of the bandwidth that is relatively RFI free.

Table 6.1: Information about the two VLA datasets to be reduced. The frequency information includes both the low (LO) and high (HI) bands.

	VLA BnA	VLA C	VLA D
Maximum baseline	11.1 km	3.4 km	1 km
Frequency (LO, HI)	1.267 GHz, 1.587 GHz	1.267 GHz, 1.587 GHz	1.267 GHz, 1.587 GHz
Angular resolution (LO, HI)	5'', 4''	16'', 13''	54'', 43''
Bandwidth (LO, HI)	256 MHz, 384 MHz	256 MHz, 384 MHz	256 MHz, 384 MHz
Synthesis time	7.64 hr	7.59 hr	5.63 hr
Integration time	5 s	5 s	5 s
Right ascension	05h42m36.138s	05h42m36.138s	05h42m36.138s
Declination	+49d51m07.234s	+49d51m07.234s	+49d51m07.234s

tion also includes a VLA primary beam model generated using Brisken’s `cassbeam` tool [7], which computes complex voltage beam patterns (the Jones matrix of the primary beam; see Figure 6.1) for a Cassegrain antenna using a geometrical ray-tracing simulation.

As previously stated, the data have been partitioned into two sub-bands for each of the three configurations. In the calibration steps, each of the six datasets will be treated separately, but we image all the datasets together in all imaging steps. Our self-calibration reduction strategy follows the *Calico* framework introduced in [61], which can be described as follows:

1. Using an initial sky model \mathbf{M}_0 , find gain matrices \mathbf{J}_i that minimises the difference between the model and the data (2.58) in a least squares sense, and compute the corrected residual data (2.60).
2. Image the corrected residual data from the previous step and extract sources from the image. Then add these sources to the sky model to produce an improved sky model, \mathbf{M}' .
3. Use the improved model, \mathbf{M}' , in another calibration step and find new gain matrices \mathbf{J}'_i . Compute the corrected residual data.
4. Repeat steps 2 and 3 until the required depth has been reached, or no new sources are revealed in the corrected residual data.

Step 1: Initial calibration

The initial calibration model contains only 3C147 itself, which lies at the phase tracking centre. This model was reconstructed from an observation of the same field with the Very Long Baseline Array (VLBA) telescope⁴.

⁴Rick Perley; private communication

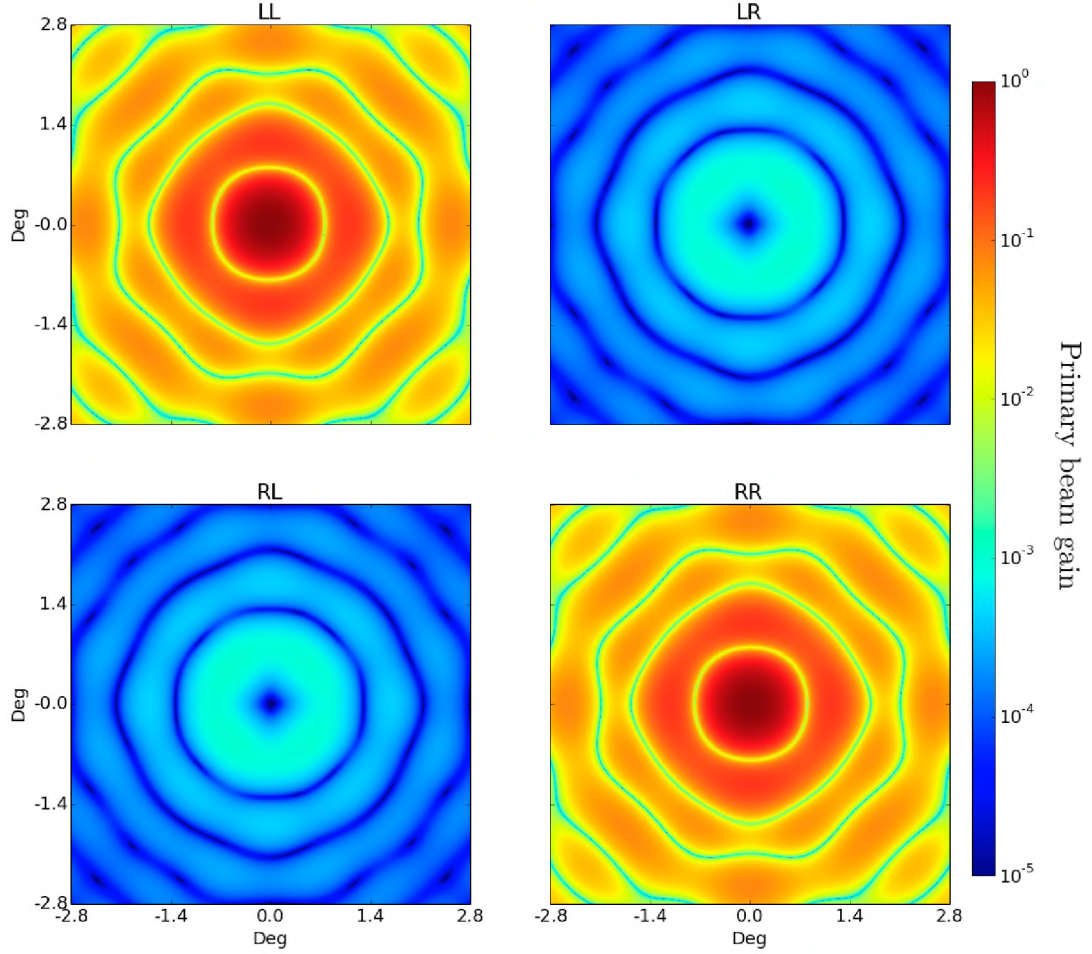


Figure 6.1: Amplitude of the Jones matrix of the VLA primary beam pattern at L-band. The beam pattern was generated using the `cassbeam` tool [7].

The RIME used in this calibration is

$$V_{pq} = \mathbf{G}_p (\mathbf{E}_p X_{pq} \mathbf{E}_q^H) \mathbf{G}_q^H. \quad (6.1)$$

The solution interval for the \mathbf{G} matrix is one solution for every time bin (5 s), for each frequency channel (4 MHz). These short solution intervals are possible because of the high SNR of the observations (due to the 22.58Jy 3C147). Assuming an SEFD value of 420 Jy for the L-band receiver⁵, and given the 5 s integration time and

⁵<https://science.nrao.edu/facilities/vla/docs/manuals/oss/performance/sensitivity>

4 MHz channel width, the radiometer equation (2.37) gives the visibility noise of 66 mJy, hence an SNR of 340.0.

The **E** Jones term is the primary beam model. This calibration is implemented in the **MeqTrees** package. We output the corrected residual data (2.60).

Step 2: RFI flagging

As previously mentioned, the data have been manually flagged for RFI. However, considering that 3C147 is very bright, some low-level RFI, which may have been masked by this source, may show up in corrected residual data from the previous calibration step (with 3C147 subtracted). We therefore use the automatic RFI flagging tool **aoflagger** [48] to flag the data based on the corrected residual data.

Step 3: Updating the sky model

The next step is to extract sources from the corrected residual data generated from the previous step. The data are first imaged and deconvolved using the **DDFacet** tool [63]. **DDFacet** is a new imaging application that uses faceting [15] to correct for widefield imaging effects while also correcting for DD effects. That is, it allows for an “*A*-term” that is non-unity when inverting Equation 2.18; the *A*-term is the primary beam pattern in our case. Note that at the time of writing, **DDFacet** is not publicly available and we have been given special permission by the author to use it for this work.

Incorporating this tool into a **stimela** recipe is a two-step process: i) make a **DDFacet** base image (see Figure 6.2); and ii) make a **DDFacet** executor image. The latter involves writing a Python wrapper for the **DDFacet** binary.

In Figure 6.3 we show the deconvolved image of the corrected residual data from the initial calibration step. The visibility weighting used in the imaging is Briggs weighting with a robustness parameter of zero. The image shows a large-scale stripe-like artefact that goes through 3C147 and a source in the field (marked with a blue circle in sub-figure (a)). This artefact is due to the interaction of the 3C147 source and this bright source, which is not in the sky model used in the initial calibration; similar to the “ghost” phenomenon described in [23]. Some of the sources also have artefacts related to DD effects, the majority of which fall on the first sidelobe of the primary beam. This is despite the primary beam model being included in the imaging, however one should note that **DDFacet** only corrects for first-order effects such as primary beam attenuation and parallactic angle rotation, and that the remaining artefacts are probably due to second-order effects such as antenna pointing errors. The sources (or directions) in question are shown in sub-figure (b), and the beam gain that these sources experience is shown in Figure 6.6. Figure 6.4 shows the primary beam corrected version of the same image.

We then extract and characterize the sources from the primary beam corrected image using the **sourcery** tool [56]. **sourcery** uses statistical analysis techniques, in

```

1 FROM radiocastro/base
2 MAINTAINER <sphemakh@gmail.com>
3
4 RUN apt-get update && apt-get install -y \
5     python-casacore \
6     libfftw3-dev \
7     python-pyephem \
8     python-numexpr \
9     cython \
10    gcc \
11    python-pyfits \
12    python-matplotlib \
13    python-scipy \
14    cmake \
15    casacore-data \
16    python-meqtrees-cattery
17
18 RUN pip install -U SharedArray Polygon2 pyFFTW
19
20 ADD project /project
21
22 RUN cd /project/DDFacet && mkdir cbuild \
23     && cd cbuild && \
24     cmake -DCMAKE_BUILD_TYPE=Release .. && make
25
26 ENV MEQTREE_CATTERY_PATH /usr/lib/python2.7/dist-packages/Cattery
27
28 # Set up environmental variables
29 ENV DDFACET_DIR /project
30 ENV PYTHONPATH $PYTHONPATH:/project
31 ENV LD_LIBRARY_PATH /project/DDFacet/Gridder:$LD_LIBRARY_PATH
32 ENV PATH /project/SkyModel:/project/DDFacet:$PATH
33
34 CMD bash

```

Figure 6.2: DDFacet `stimela` base image template (Dockerfile).

particular the *negative detection* technique described in [57], to distinguish between astrophysical and spurious emission. The extracted sources are then added to the initial sky model, forming the *updated* sky model which will be used in the next calibration step. Note that the ability of the `sourcery` tool to distinguish actual sources from artefacts allows us to create a deep sky model without being concerned about including artefacts in our sky model. Note that we extract sources from an image that has been corrected for the primary beam gain, Figure 6.4, and not from Figure 6.3, which has not been corrected for the primary beam gain. This will be the case for all source-finding steps.

Given the large fractional bandwidth of the data, the spectral behaviour of the sources also has to be modelled. We know from the astrophysical behaviour of synchrotron emission, that the spectral behaviour of the sky brightness can be modelled as a power law,

$$I(\nu) = I_0 \left(\frac{\nu}{\nu_0} \right)^{-\alpha}, \quad (6.2)$$

where I_0 is the sky brightness at some reference frequency ν_0 , and the power law index α is known as the spectral index⁶. The behaviour can then be parametrised via

⁶<http://www.cv.nrao.edu/course/ast534/SynchrotronSpectrum.html>

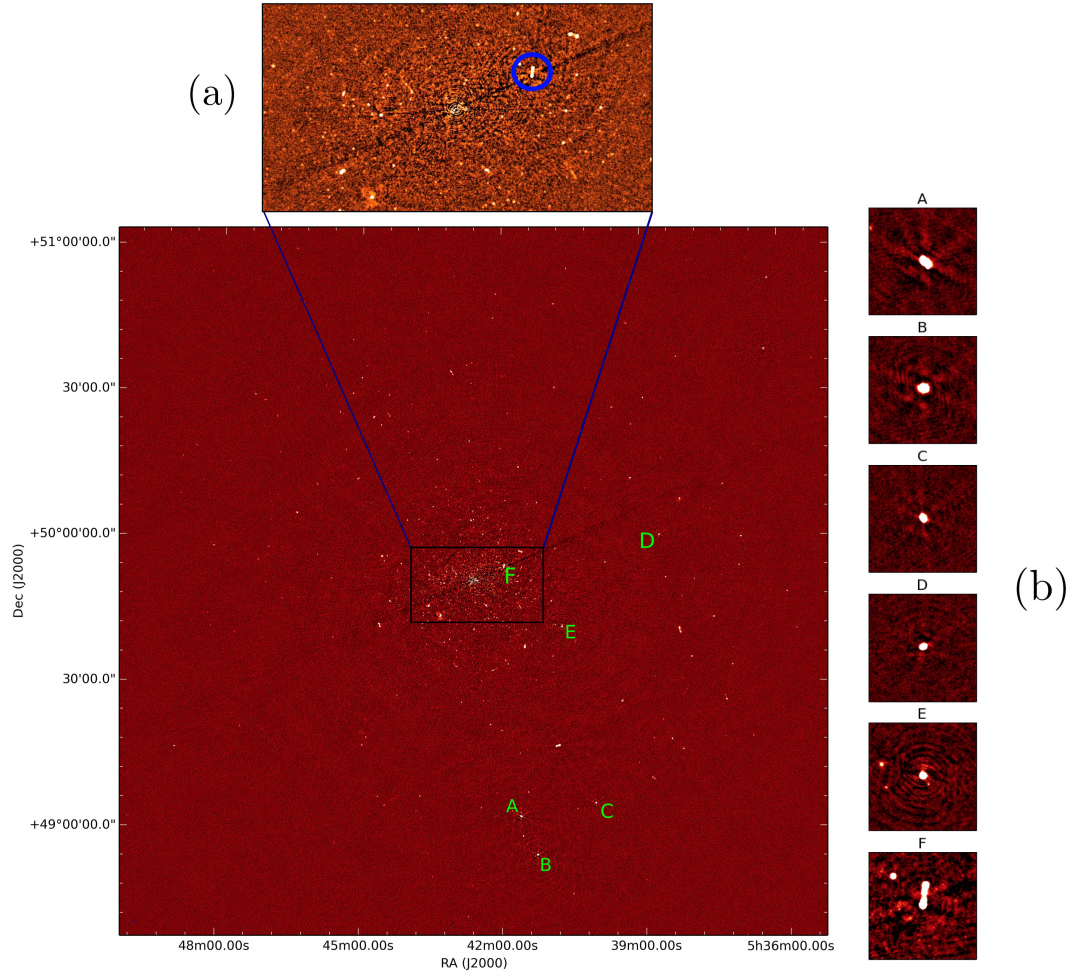


Figure 6.3: Corrected residual image (apparent fluxes) after calibrating with a model of the source 3C147. In sub-figures (a) and (b) we zoom into sources in the image that have artefacts around them; these artefacts are discussed in the text. The image is 2.4 degrees wide, and it has an RMS pixel value of $3.51 \mu\text{Jy}/\text{Beam}$ and an angular resolution of $5.2''$.

this α . Taking the log of both sides on (6.2) gives,

$$\begin{aligned}
 \ln I(\nu) &= \ln \left[I_0 \left(\frac{\nu}{\nu_0} \right)^{-\alpha} \right] \\
 &= \ln I_0 - \alpha \ln \left(\frac{\nu}{\nu_0} \right).
 \end{aligned} \tag{6.3}$$

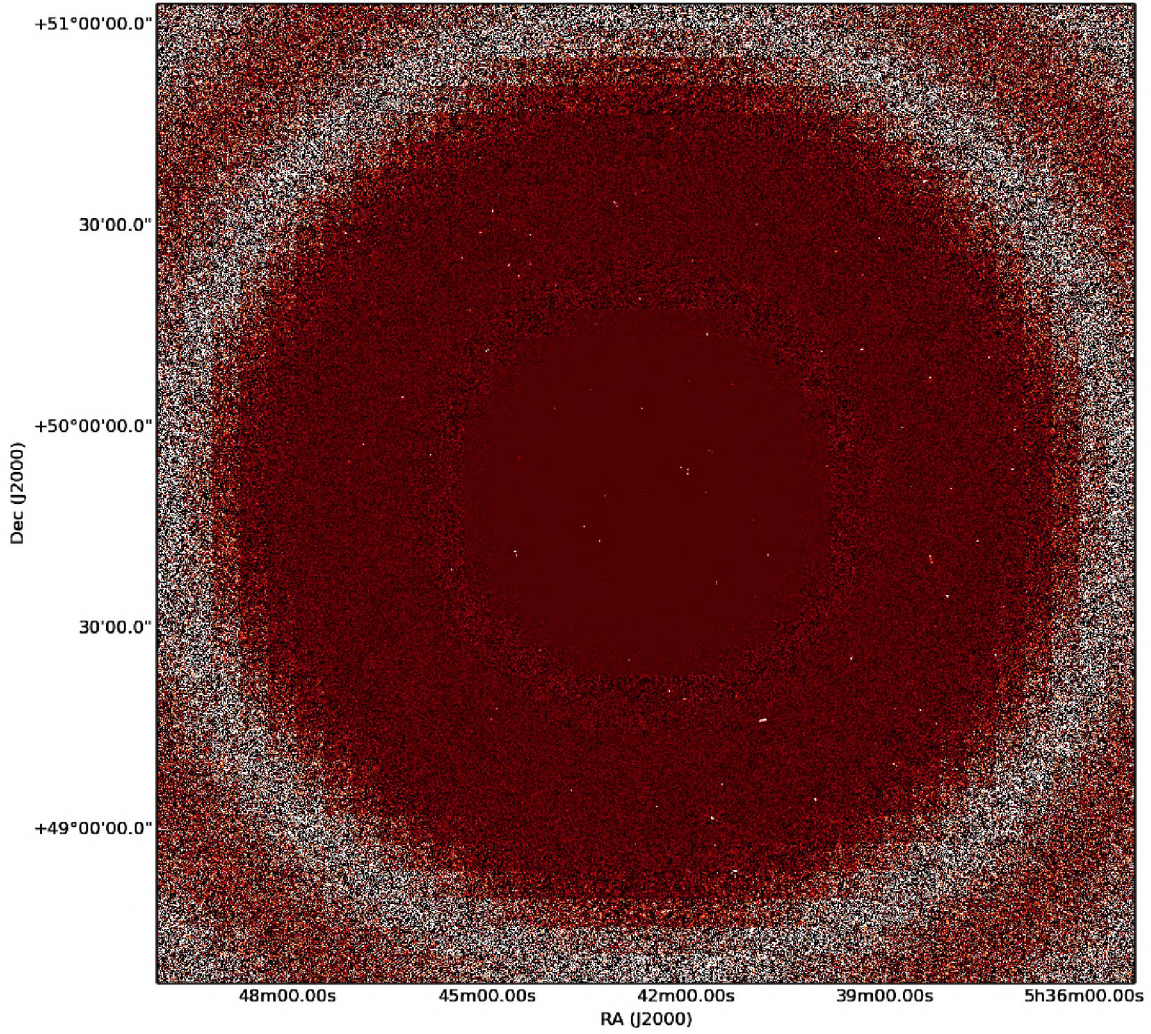


Figure 6.4: Primary beam corrected image of the corrected residual data after calibrating with a model for the source 3C147.

Recognising α and $\ln I_0$ as the slope and intercept of the straight line (6.3), we can use linear regression [71] to estimate α as

$$\alpha = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \quad (6.4)$$

with standard error

$$\sigma_\alpha = \sqrt{\frac{\sum_i (y_i - \bar{y})^2}{N}}, \quad (6.5)$$

where $x = \ln \frac{\nu}{\nu_0}$, $y = \ln I(\nu)$, \bar{x} , \bar{y} are the average values of x and y , and N is the number of samples. Equations (6.4) and (6.5) are then used to create a spectral index map along with a standard error map from multi-channel brightness images from the **DDFacet** imager. The spectral index for each source in the sky model is then the weighted average of the corresponding pixels in the spectral index map, where the weights are $1/\sigma_\alpha$. Again, note that the spectral indices are fitted from a primary beam corrected image cube.

Step 4: Second calibration step

Next, we perform another calibration step, which includes DD solutions for the sources selected as requiring DD calibration in the previous step. The RIME used in this step is

$$\mathbf{V}_{pq} = \mathbf{G}_p \left(\sum_{s=1}^N \Delta \mathbf{E}_{sp} \mathbf{E}_{sp} \mathbf{X}_{spq} \mathbf{E}_{sq}^H \Delta \mathbf{E}_{sq}^H \right) \mathbf{G}_q^H, \quad (6.6)$$

where, $\Delta \mathbf{E}$ is the differential gain term introduced in Section 3.2.4. As in step 1, we output the corrected residual.

Step 5: Updating sky model

The corrected residual data from the previous step is imaged and deconvolved using **DDFacet**. The image is shown in Figure 6.5 and has off-source RMS pixel noise of $3.39 \mu\text{Jy}$ in the image with apparent fluxes. Note that the DD artefacts have been significantly reduced after the inclusion of differential gains in the RIME as can also be seen in Figure 6.6. As we did in step 3, we then extract sources from this image and update the sky model.

Step 6: Third calibration step

We run another calibration step using the updated model from the previous step, and the same RIME as in Step 4 (6.6).

Step 7: Updating sky model

Figure 6.7 shows the corrected residual image (apparent flux) from the previous calibration step. The emission left in the data at this point is what the source finder could not parametrise in the two source-finding steps. Some of this emission, however, is captured in the deconvolution model. The part that is captured in this model is the emission above the deconvolution threshold, which was set to 3σ . We add this deconvolution model to the sky model that will be used in the next calibration step.

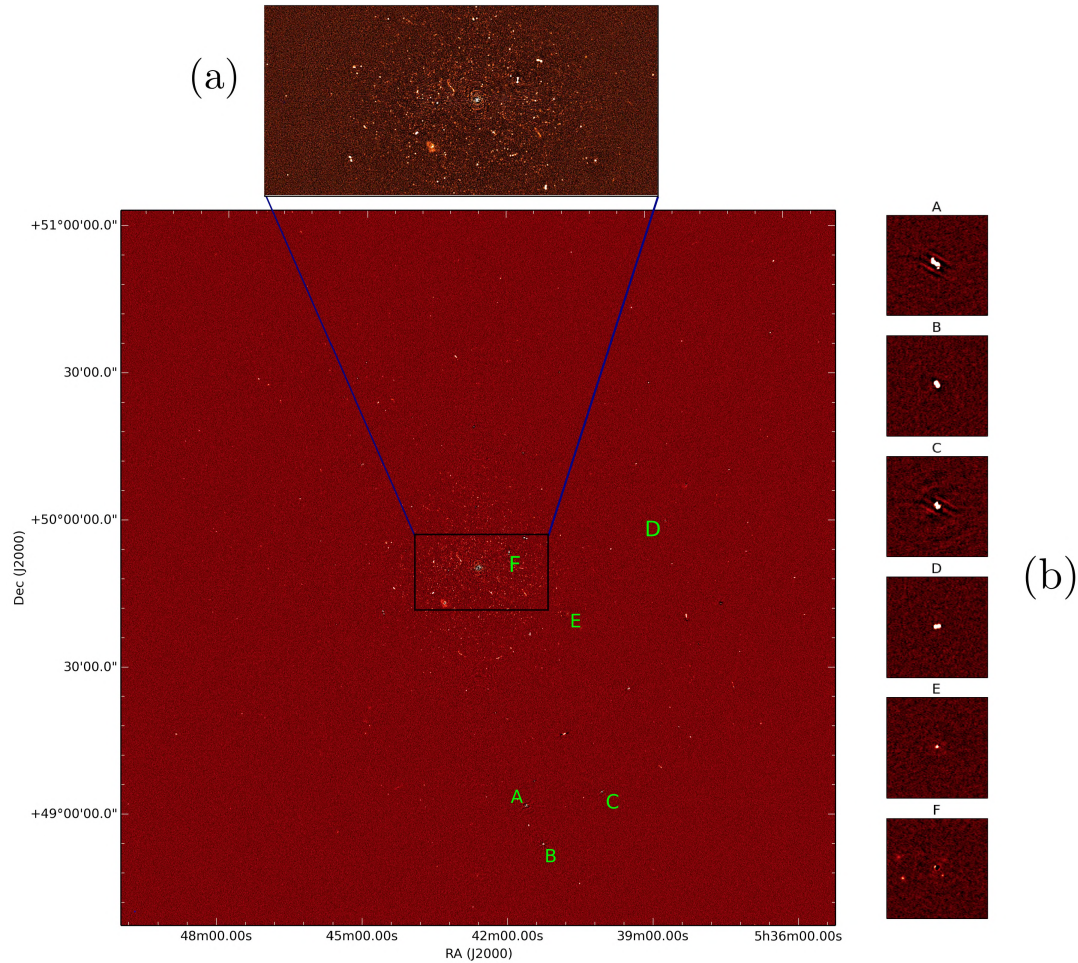


Figure 6.5: Corrected residual image (apparent fluxes) after the second calibration step. Also included is a cut-out of the region around the field centre, as well as a region that has sources that fall in an area of the primary beam that is subject to large variations over the course of the observation. The image is 2.4 degrees wide, and it has an RMS pixel value of $3.39 \mu\text{Jy}/\text{Beam}$ and an angular resolution of $5.2''$.

Step 8: Final calibration step

We can see from Figure 6.8 that the central part of the residual image at this point is limited by spoke-like features as well as spiral-like features that emanate from the centre of the image (also the phase tracking centre of the observations). We believe that these spurious features are due to a combination of two things; i) our model of the 3C147 is not accurate enough to properly subtract the source at this resolution and depth; and ii) closure errors. In the case of the WSRT telescope, closure errors are due to the analogue correlator [61], but while their origin is not fully understood

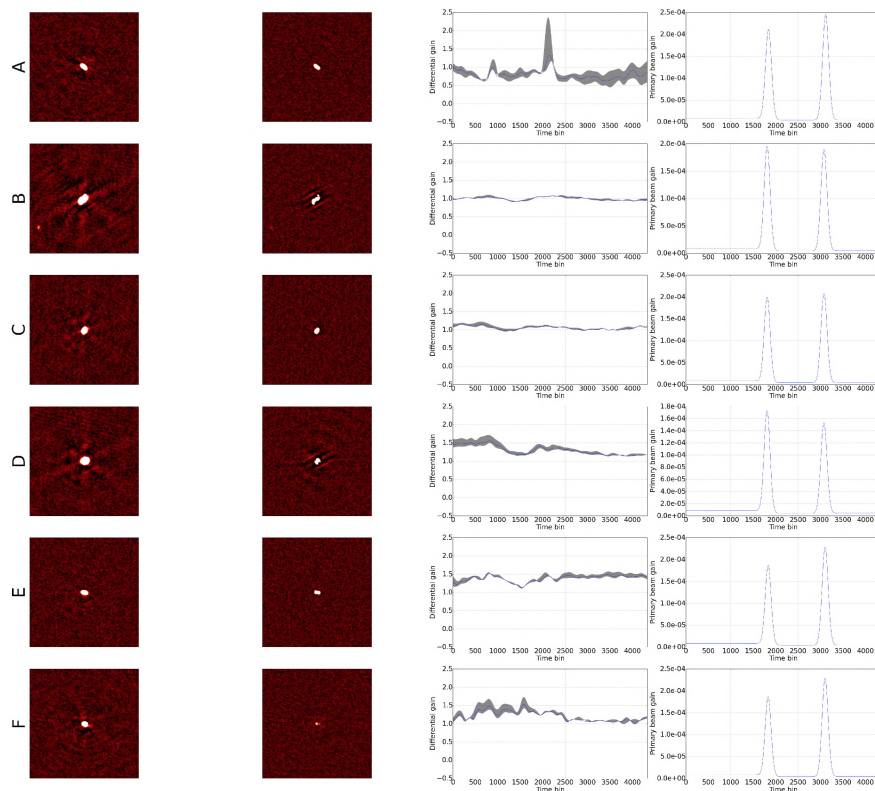


Figure 6.6: Sources that had DD gains applied; differential gains [60] in this case. The first column shows the sources before the gains were applied; the second column the corrected residual after the gains were applied; the third column shows the gain (for antenna 11 of the C-configuration LO data) that was applied as a function of time for the middle frequency channel (blue curve). The grey region shows the gains for the rest of the channels; and the fourth column shows the primary beam gain that the sources experience as a function time (`cassbeam`[7] model of the VLA primary beam). The position of the sources shown here can be located in the field in Figure 6.5.

for the VLA telescope which has a digital correlator, they may be due to phase and averaging errors that have not been accounted for⁷. Nonetheless, as shown in [42] these errors can be corrected for using interferometer based gains (IFR gains). This is a complex per-baseline multiplicative Jones term, M , which can be added to the

⁷Rick Perley, private communication.

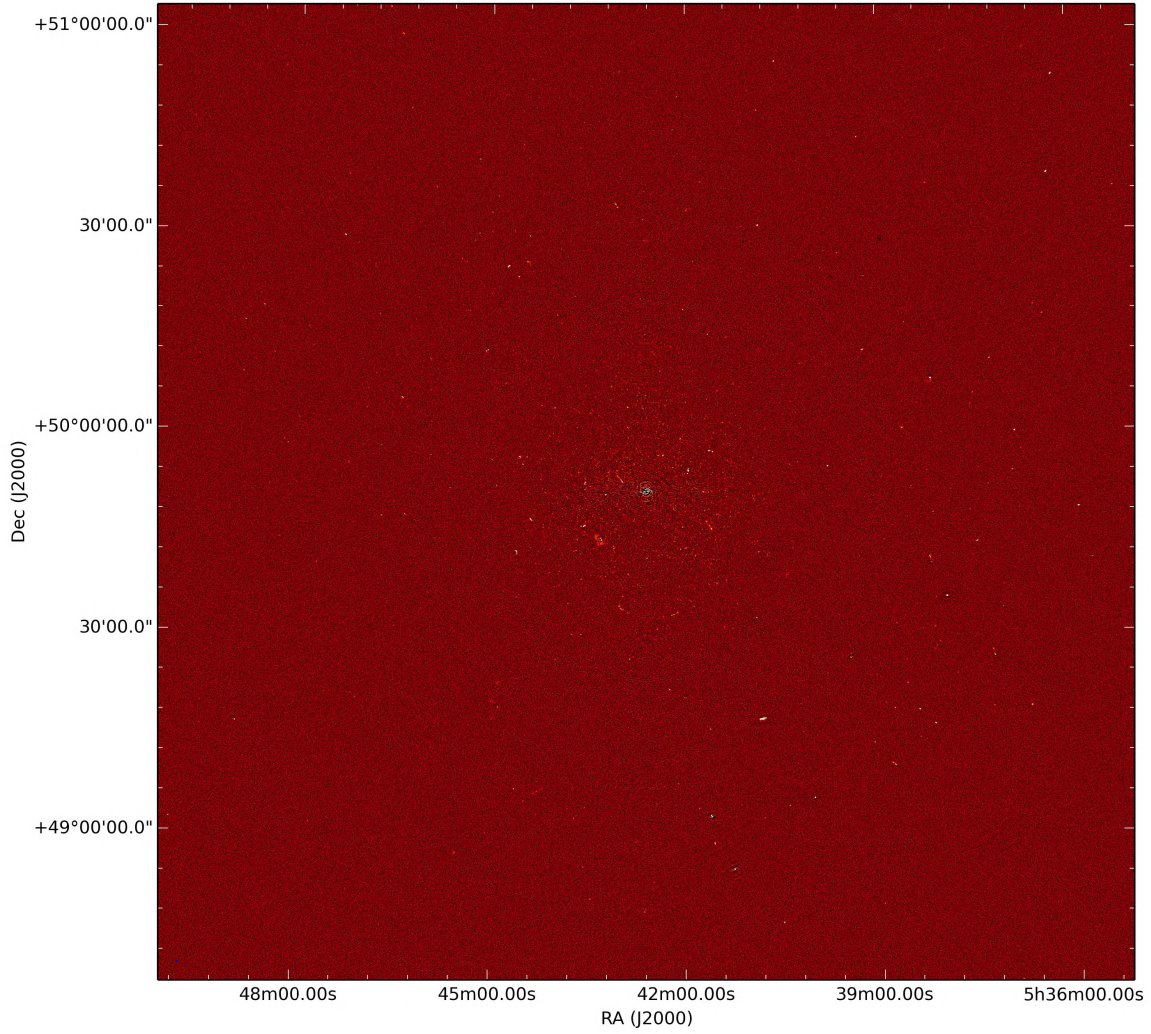


Figure 6.7: Corrected residual image (apparent fluxes) after the third calibration step. The image is 2.4 degrees wide, and it has an RMS pixel value of $3.39 \mu\text{Jy}/\text{Beam}$ and an angular resolution of $5.2''$

RIME as

$$\mathbf{V}_{pq} = M_{pq} \times \mathbf{G}_p \left(\sum_{s=1}^N \Delta \mathbf{E}_{sp} \mathbf{E}_{sp} \mathbf{X}_{spq} \mathbf{E}_{sq}^H \Delta \mathbf{E}_{sq}^H \right) \mathbf{G}_q^H, \quad (6.7)$$

where M is the matrix that accounts for IFR gains, and “ \times ” denotes per element multiplication. The rest of the symbols are as explained in (3.3). Note that IFR gains can absorb all differences between the sky model and the data that are not corrected for by the \mathbf{G} , \mathbf{E} and $\Delta \mathbf{E}$ Jones terms. To preserve the fidelity of the data, it is typical to solve for these gains on large time intervals [61]. Hence, the solution intervals for

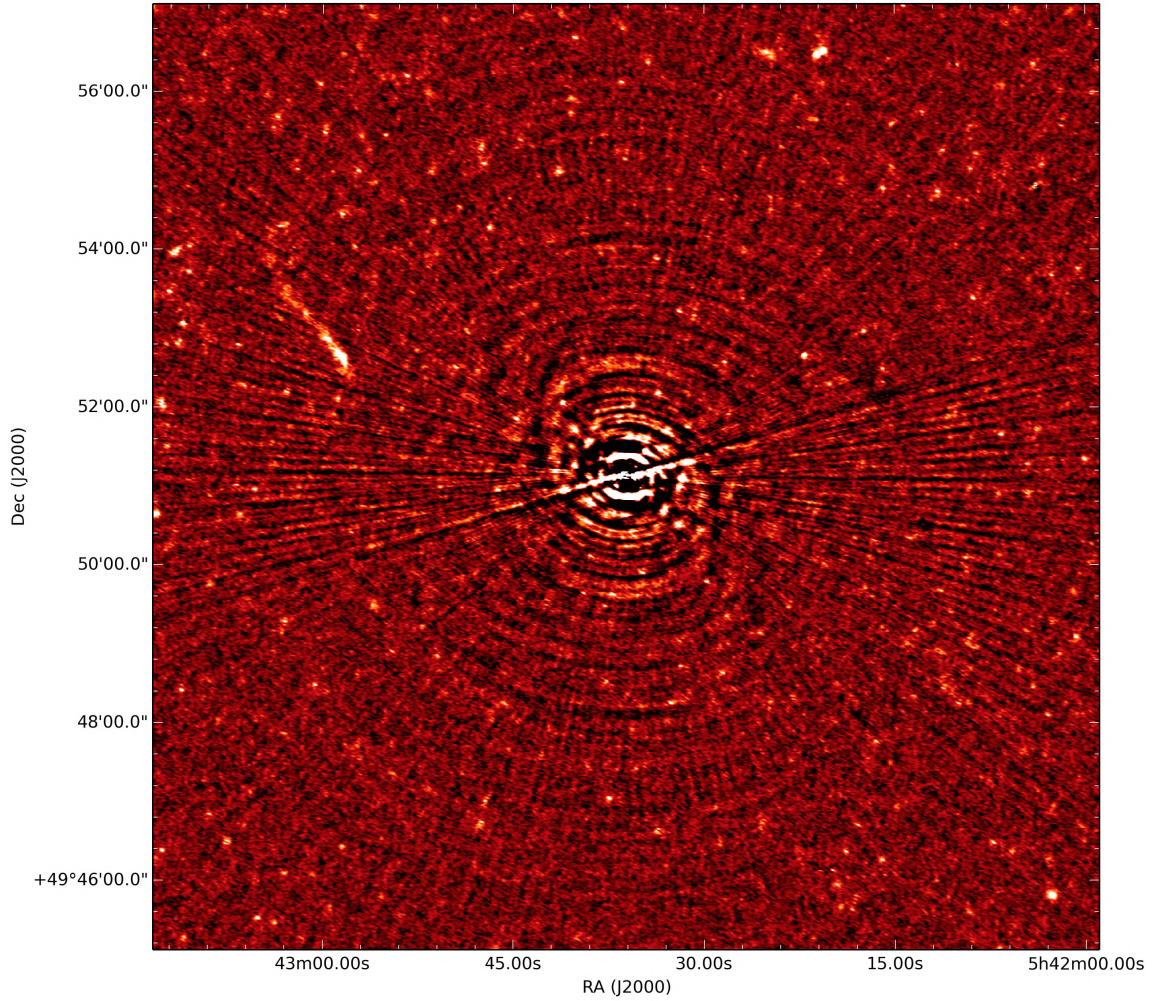


Figure 6.8: Central 13 arcminutes of the corrected residual image after the third calibration step.

the IFR gains is one solution for every frequency bin (the time solution interval is the entire observation), and we use the same solution intervals as in step 4 for the \mathbf{G} and $\Delta\mathbf{E}$ terms.

Step 9: Making final residual images

The image of the corrected residual data from the previous step, Figure 6.10, shows that the artefacts have been significantly reduced after the application of the IFR gains; however, some imperfections still remain. At this point we need a better model of 3C147 in order to subtract the source better. Alternatively, we could improve upon our current model through model-fitting techniques such as the Bayesian Inference for Radio Observations [38, BIRO] or UVMULTIFIT [41]. However, these techniques

(at the time of writing) can only handle a few of sources, and would not be able to handle the 86 components in our 3C147 model and the large volume of data for this observation; at least not within a reasonable time.

We stop our self-cal loop at this point. The final residual from our reduction is shown in Figure 6.9, along with a histogram of the pixels in the images.

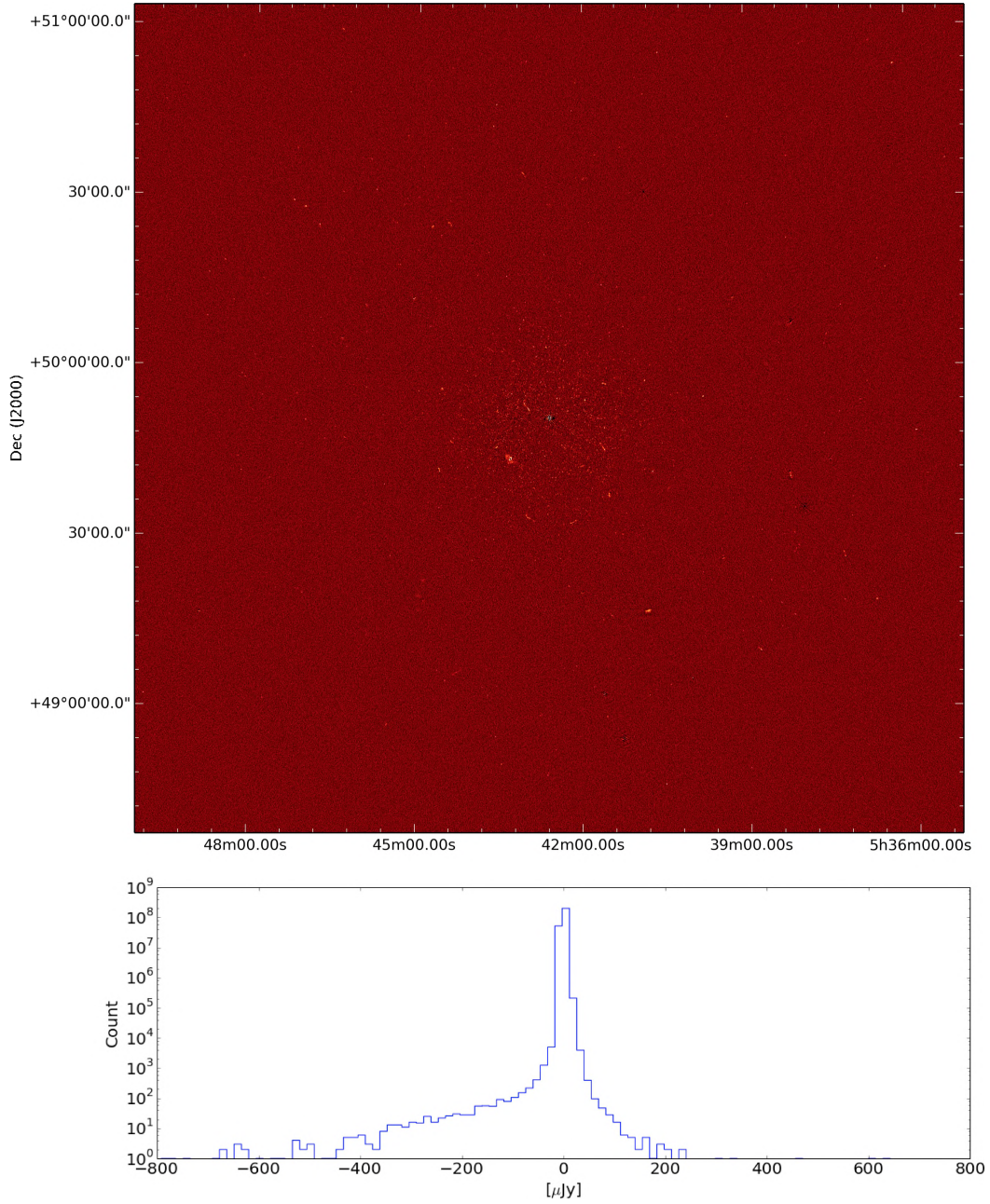


Figure 6.9: Corrected residual image (apparent fluxes) after the fourth calibration step. Also shown is the histogram of the pixels in the image. The RMS value of the pixels in this image is $3.37 \mu\text{Jy}/\text{Beam}$, and has a resolution of $5.2''$.

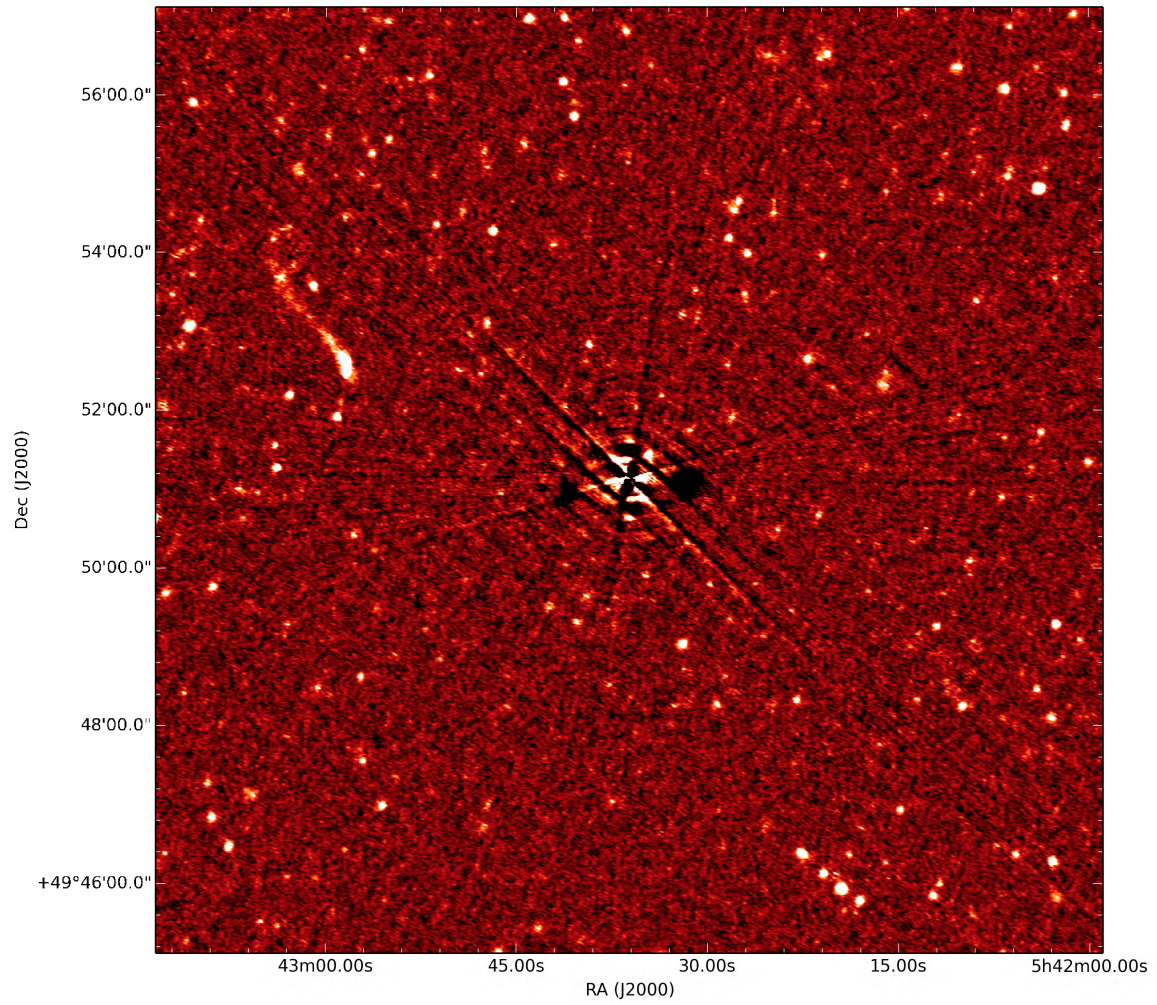


Figure 6.10: Central 6 rminutes of the corrected residual image after the third calibration step.

6.1.1 Data products

The principal products from this reduction are 2.4 degree wide high- and low-resolution images – Briggs weighting with robustness parameters 0 and 0.5 respectively – and corresponding source catalogues for point-like sources in the field. For both cases we provide images with intrinsic (primary beam corrected fluxes) and apparent (primary beam attenuated fluxes) fluxes. The typical 1σ noise level in the high-resolution image (Figure 6.11) is $3.37 \mu\text{Jy}/\text{beam}$ and $2.87 \mu\text{Jy}/\text{beam}$ in the low-resolution image (Figure 6.12), the corresponding the dynamic ranges, computed as the ratio the brightest pixel to the 1σ noise level in the image, are $6.7 \times 10^6 : 1$ and $7.9 \times 10^6 : 1$ respectively. The catalogues from the high- and low-resolution images are presented in Tables 6.2 and 6.3, while Figures 6.13 and 6.14 show the central 1 degree of the high- and low-resolution images respectively. All these data products are publicly available at http://bit.ly/3c147_data_products.

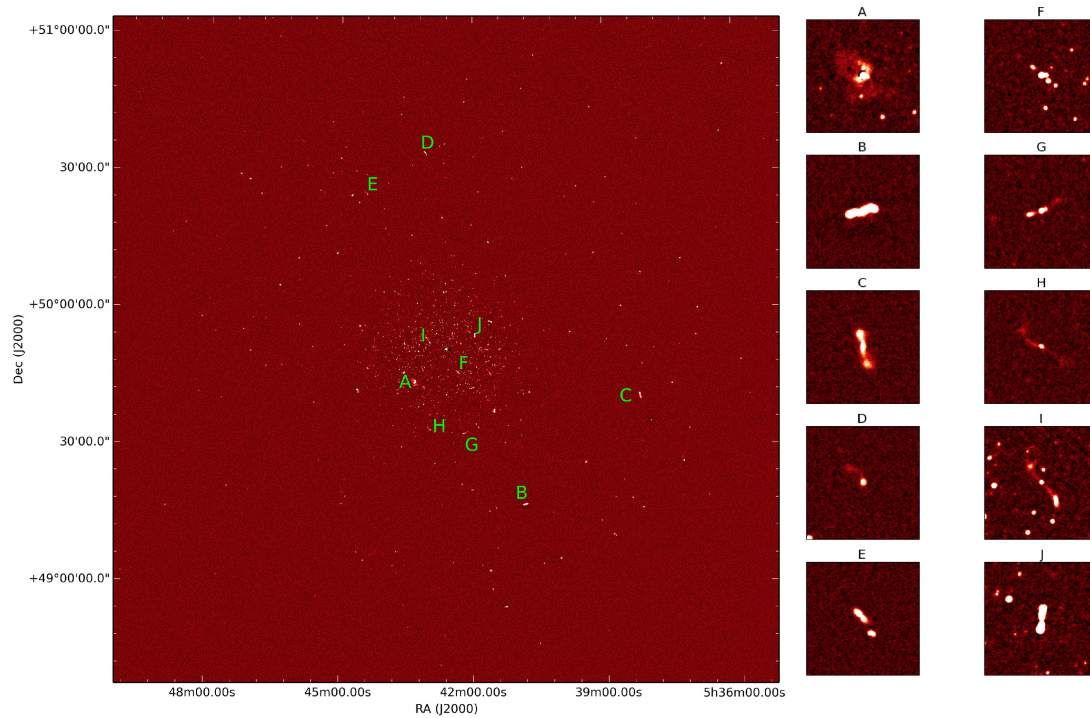


Figure 6.11: Reconstructed image from our reduction of a VLA observation of the field around 3C147. On the right we also show cut-outs of some sources (annotated in the main image) in the field. The image is 2.4 degrees wide and has an effective 1σ noise level (a region of the image without sources) of $3.37 \mu\text{Jy}/\text{beam}$, and an angular resolution of $5.2''$.

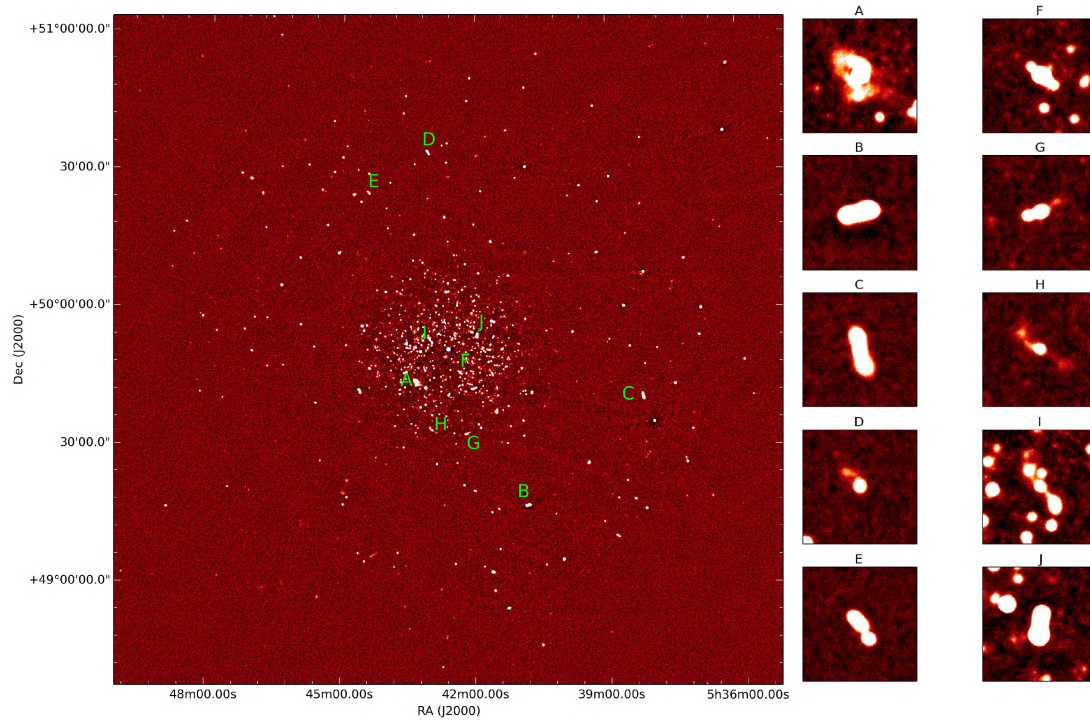


Figure 6.12: Reconstructed image from our reduction of a VLA observation of the field around 3C147. On the right we also show cut-outs of some sources (annotated in the main image) in the field. The image is 2.4 degrees wide and has an effective 1σ noise level (a region of the image without sources) of $2.87 \mu\text{Jy}/\text{beam}$, and an angular resolution of $13.0''$.

6.2 Discussion

The high dynamic range obtained from this reduction, while impressive, is not the most important outcome of this chapter. The most important outcome is the ease at which it was attained, and the fact that the result can be reproduced on almost any computing platform with minimal effort (the platform only needs to be able to run Docker). The scripting framework is also very transparent and configurable, making it easy for others to improve the strategy or adapt it to their datasets. This is particularly exciting when one considers the vast amounts of data that goes unprocessed because many in the community, though competent to reduce the data, do not have the time and/or skills to obtain the required software environments in order to do the processing. More importantly, with instruments such as ASKAP, MeerKAT, and

Table 6.2: Top 10 components extracted from our reconstructed high-resolution (5.2'') image. The parameters presented are: the right ascension (R.A), declination (Dec.) integrated flux density (Stokes I ; S_{Total}), peak flux density (S_{Peak}), major (Maj) and minor (Min) axes as well as a position angle (P.A) from a 2-D Gaussian fit to the source morphology. All the parameters are accompanied by an error estimate $\varepsilon()$ that was reported by the `pybdsm` source finding tool.

ID	R.A. [deg]	δ (R.A.) [arcsec]	Dec. [deg]	δ (Dec.) [arcsec]	S_{Total} [mJy]	$\delta(S_{\text{Total}})$ [mJy]	S_{Peak} [mJy]	S_{Peak} [mJy]	Maj. [arcsec]	δ (Maj) [arcsec]	Min. [arcsec]	δ (Min) [arcsec]	P.A [deg]	δ (P.A) [deg]
0	85.49012	0.01118	49.89855	0.00197	36.18411	0.02430	23.86048	0.00400	28.87814	0.02658	2.20918	0.00289	171.99945	0.04271
1	85.40135	0.00872	49.04353	0.00954	18.62071	0.01718	5.70713	0.00342	14.21155	0.02933	3.46269	0.00813	47.84227	0.12624
2	85.82734	0.00110	49.73167	0.00101	18.61596	0.01843	13.24272	0.00391	7.02293	0.00307	3.46213	0.00173	140.06290	0.04548
3	85.80833	0.00077	49.84377	0.00095	14.50496	0.01146	12.21630	0.00414	3.00037	0.00231	0.59857	0.00171	112.09402	0.09961
4	85.89001	0.00601	49.76652	0.00625	13.86592	0.01406	5.03073	0.00392	12.83335	0.01937	4.07755	0.00645	133.78895	0.10240
5	85.18218	0.00109	49.69490	0.00129	10.39223	0.00993	8.02160	0.00352	3.67307	0.00313	1.93276	0.00245	66.21786	0.15274
6	85.00856	0.00219	49.08883	0.00160	9.76746	0.01081	5.83218	0.00334	6.13688	0.00569	1.45340	0.00291	29.01919	0.08061
7	85.40347	0.31920	49.95320	0.93289	7.33860	0.01558	1.73795	0.00375	38.47764	2.29774	5.97416	0.33358	72.57609	3.55100
8	85.31444	0.00389	48.91186	0.00420	7.11555	0.01351	3.23987	0.00342	7.36660	0.01089	5.18597	0.00796	52.00942	0.26695
9	85.66512	0.00285	50.05987	0.00194	6.40639	0.00987	4.61225	0.00369	5.35442	0.00703	1.44817	0.00406	21.43736	0.13128

Table 6.3: Top 10 components extracted from our reconstructed low-resolution (13.0'') image. The parameters presented are: the right ascension (R.A), declination (Dec.) integrated flux density (Stokes I ; S_{Total}), peak flux density (S_{Peak}), major (Maj) and minor (Min) axes as well as a position angle (P.A) from a 2-D Gaussian fit to the source morphology. All the parameters are accompanied by an error estimate $\varepsilon()$ that was reported by the `pybdsm` source finding tool.

ID	R.A. [deg]	δ (R.A.) [arcsec]	Dec. [deg]	δ (Dec.) [arcsec]	S_{Total} [mJy]	$\delta(S_{\text{Total}})$ [mJy]	S_{Peak} [mJy]	S_{Peak} [mJy]	Maj. [arcsec]	δ (Maj) [arcsec]	Min. [arcsec]	δ (Min) [arcsec]	P.A [deg]	δ (P.A) [deg]
0	84.08442	0.00506	50.64060	0.00543	2996.91493	4.84151	2463.97965	1.98839	8.33526	0.01415	0.00000	0.01026	127.32826	0.22053
1	85.40134	0.00231	49.04354	0.00250	965.12930	0.50795	599.27071	0.17178	14.12982	0.00711	3.21154	0.00372	48.81713	0.04305
2	85.31440	0.00424	48.91177	0.00471	804.28053	1.54439	664.71291	0.45963	6.86726	0.01123	4.80138	0.00983	70.81836	0.43218
3	85.00855	0.00258	49.08885	0.00241	596.69698	0.62843	532.91711	0.21627	6.03145	0.00642	1.40786	0.00530	33.89506	0.18035
4	87.22598	0.06040	49.27558	0.10317	435.83017	5.96086	241.47311	2.47355	12.74071	0.24724	1.05270	0.13462	78.40636	1.69264
5	84.05902	0.03987	50.88467	0.04448	350.80497	2.07329	235.00573	1.34592	13.29438	0.12020	4.52038	0.07306	140.61545	179.74397
6	84.22457	0.00740	49.99786	0.00932	269.56041	0.79833	236.56849	0.32600	0.00000	0.02201	0.00000	0.01733	96.13109	0.49790
7	85.20452	0.02839	49.28628	0.08949	264.87452	0.42289	68.29158	0.15699	44.52920	0.21671	7.39473	0.04375	103.44363	0.25978
8	84.66133	0.00426	50.00814	0.00383	229.47639	0.25275	216.05361	0.14879	4.63715	0.01003	1.13673	0.00900	93.54186	179.76254
9	85.12801	0.07080	48.77748	0.09218	224.01301	3.91483	186.72013	2.38609	8.93196	0.22461	0.00000	0.15642	20.59565	2.61949

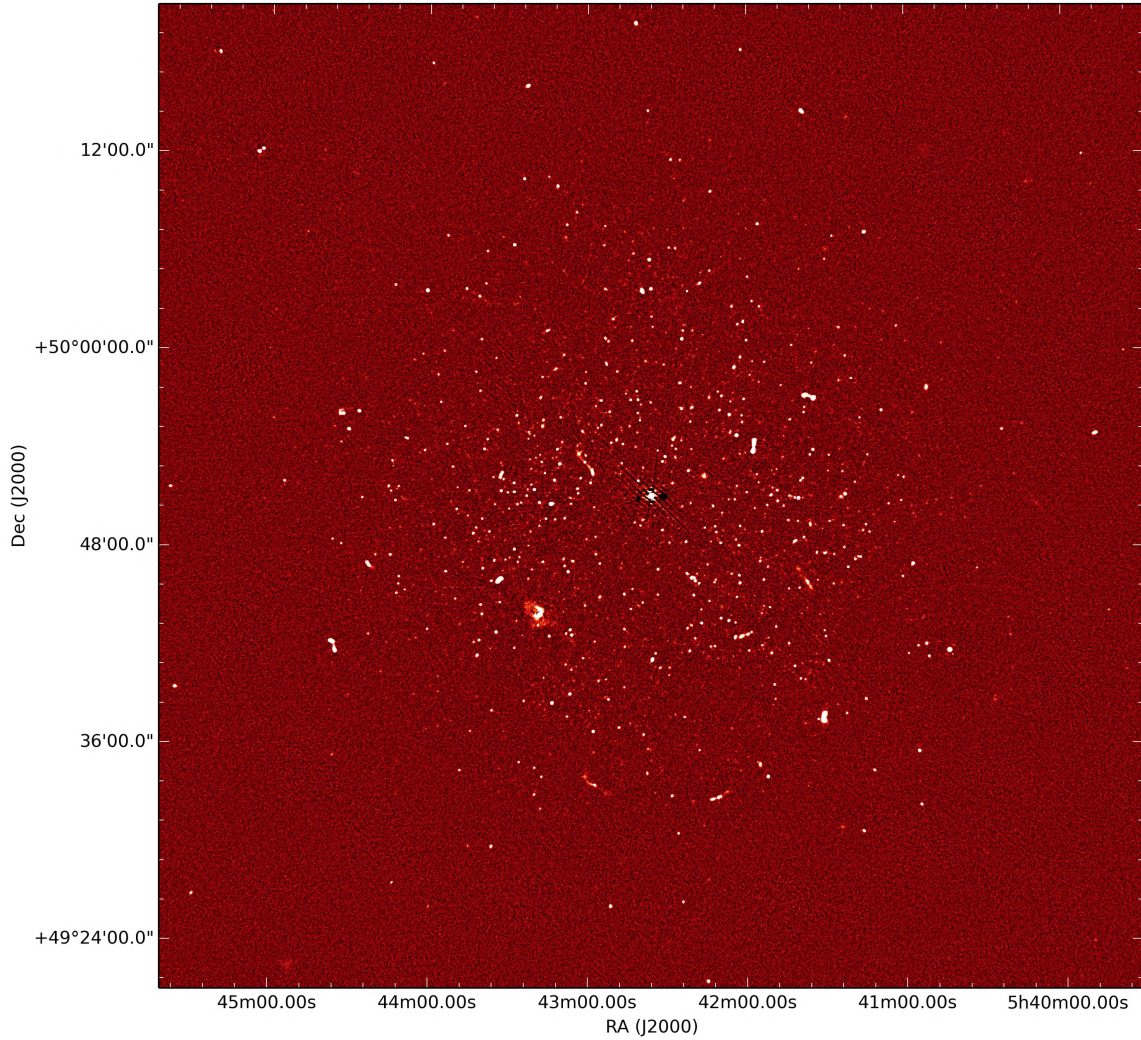


Figure 6.13: Central 1 degree of the reconstructed image from our reduction of a VLA observation of the field around 3C147. The image has an effective 1σ noise level (a region of the image without sources) of $2.37 \mu\text{Jy}/\text{beam}$, and an angular resolution of $5.2''$.

eventually the SKA going online, packages such as `stimela` should play an increasingly important role if these instruments are to fulfil their scientific potential.

Another important feature of this framework, as illustrated with the inclusion of our own spectral fitting module in section 6.1 and the usage of the `DDFacet` imaging tool, is the simplicity with which experimental and novel tools and algorithms can be incorporated and tested. Again, this will be crucial in the development and testing of new algorithms, techniques and pipelines in preparation for the SKA and its precursors.

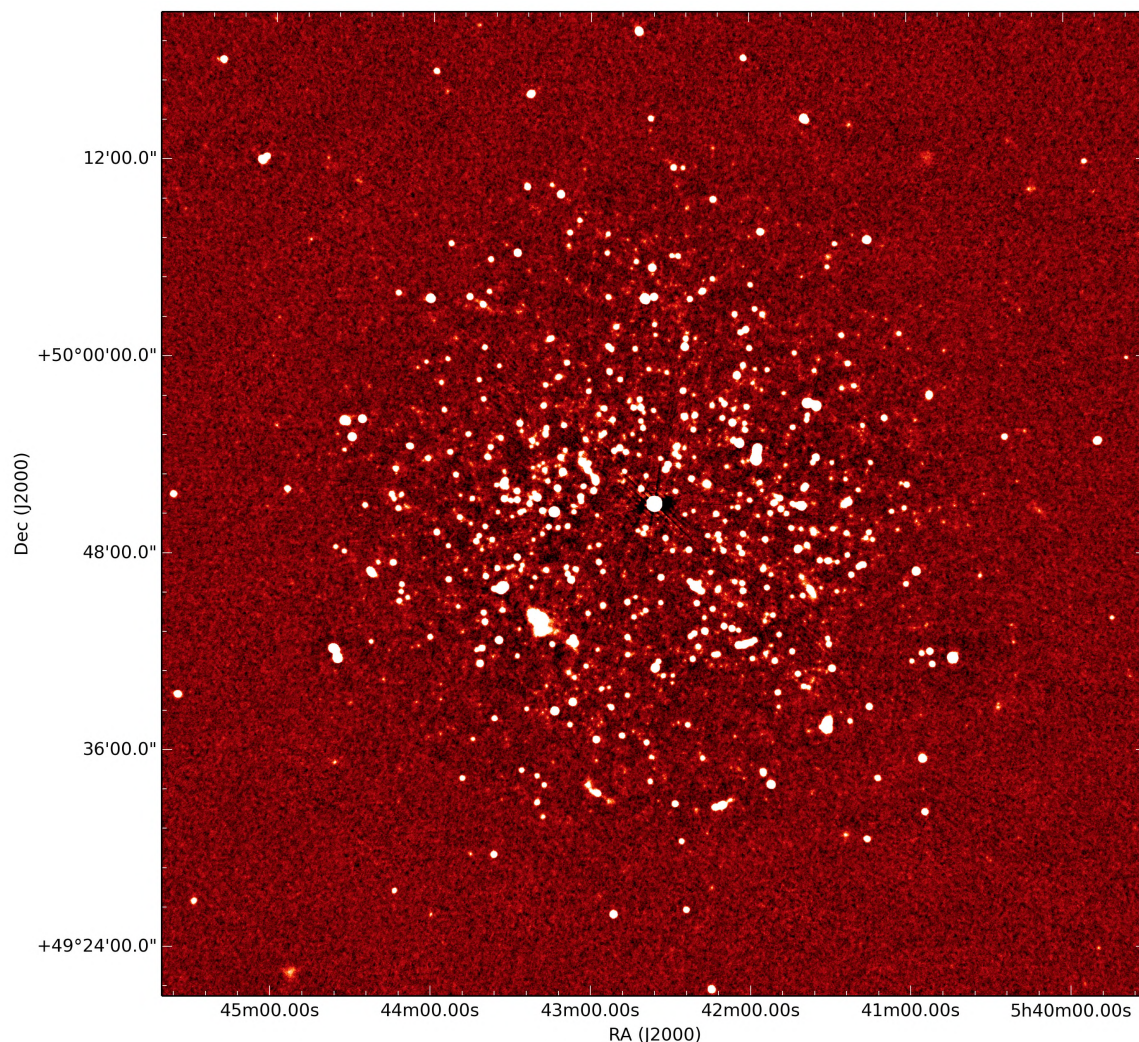


Figure 6.14: Central 1 degree of the reconstructed image from our reduction of a VLA observation of the field around 3C147. The image has an effective 1σ noise level (a region of the image without sources) of $2.87 \mu\text{Jy}/\text{beam}$, and an angular resolution of $13.0''$.

6.3 Conclusions

We have presented a platform-independent reduction of a VLA (BnA, C and D configurations) observation of the field around the source 3C147 at L-Band that includes a thorough treatment of DD effects in both the imaging and calibration steps. The data products from this reduction are high- and low-resolution images along with corresponding source catalogues. The high-resolution image has a noise level of $3.37 \mu\text{Jy}/\text{Beam}$ and a dynamic range of $6.7 \times 10^6 : 1$, while the low-resolution image has a noise level of $2.87 \mu\text{Jy}/\text{Beam}$ and a dynamic range of $7.9 \times 10^6 : 1$. Noting

that the highest published (to the best of our knowledge) dynamic range from a radio image is $5 \times 10^6 : 1$ [42], our low-resolution image has the highest dynamic range to date, and the high-resolution image has the second highest dynamic range to date.

The reduction is largely automated, with the setting of calibration solution intervals, deconvolution thresholds and the identification of sources that require DD solutions being the only parts that require human supervision. The setting of thresholds for the source finder and the discarding of artefacts have been automated via the `sourcery` tool.

Chapter 7

Conclusions

We have presented a simulations framework that can be used to study the impact of DD effects on survey depth when trying to achieve a deep survey by accumulating many images from noise-limited reductions. In the case of antenna pointing errors, for the MeeKAT telescope, we have shown that artefacts induced by pointing errors can be calibrated to the noise level for a given pointing but these artefacts still exist below the noise, and as many such images are accumulated and the noise decreases, the artefacts decrease at a rate that gets progressively slower than that of the noise and eventually dominate the noise budget. Moreover, the accumulation of the artefacts eventually plateaus, at which point observing the field no longer yields additional depth. We estimate that for a LADUMA-like survey, at L-band, this artefact limit is at about $10\ \mu\text{Jy}$ in a 72 kHz channel.

We also presented **stimela**, a system agnostic radio interferometry data synthesis and reduction framework. It is capable of doing end-to-end telescope simulations, and reductions with a full treatment of both DI and DD effects. The **stimela** framework is very flexible, and, as shown in Chapter 6, incorporating new reduction/imaging tools into a **stimela** recipe is very simple. More importantly, this framework gives users access to legacy and novel software packages without them having to install (or build) these packages, which, as discussed in Section 4.1, can be a difficult and time-consuming task.

We then used a **stimela** based simulations framework to study the scale-dependent sensitivity of the SKA1-MID telescope. In the same chapter we showed that two SKA1-MID antenna layouts with maximum baselines of 120 km and 137 km have very similar sensitivity and angular resolution properties compared to the layout proposed by the SKA organisation.

Finally, we use **stimela** to reduce a VLA observation of the field around the source 3C147. The final image from this reduction has a dynamic range of $7.9 \times 10^6:1$, the highest dynamic range in a radio synthesis image to date. However, the central part of the image is limited by improper subtraction of our model of 3C147. A better model of the source is required to improve the reduction beyond this point. Nonetheless, we

have shown the flexibility of the `stimela` package, and how it can simplify the process of reducing radio data. More importantly, the package is an ideal environment for running large-scale and reproducible data reductions using a variety of both legacy and novel software packages. Another important feature of this package is that the user can guarantee that their pipelines will produce the same results if they share them with collaborators.

7.1 Future Work

We will be extending the framework developed in Chapter 3 to gauge the impact of using an incomplete sky model during calibration. As mentioned in Section 2.6.1, calibrating with an incomplete sky model induces artefacts (see [23] for a more detailed discussion), which may also limit survey depth if the survey depth is achieved by accumulating images from noise-limited reduction from various stages of the survey lifetime. This is a much larger problem in terms of computing capacity and storage because the sky models would get very large with increasing observing time, making the simulations and calibrations much harder. However, both the simulations and calibrations can be accelerated by using `montblanc`; a GPU implementation of the RIME.

The `stimela` recipe in Chapter 6 is fully automated, but achieving this has required human intervention along the way in order to tune several key parameters. Examples of these are: (i) determining optimal calibration solution intervals; (ii) determining appropriate deconvolution thresholds and (iii) determining which sources (or directions) require the application of DD solutions. We are planning on developing algorithms that can estimate these parameters. Key to these algorithms will be testing their performance on a broad range of datasets and operating parameters. The results of these can be used to construct heuristics based on the properties of the input data (e.g. position of strong sources relative to primary beam sidelobes, knowledge of PSF sidelobe levels, total expected flux in the pointing, etc.) as well as the performance of various elements of the reduction (e.g. success of automatic artefact rejection algorithms such as `sourcery`). These findings can be converted into code that sets the parameters for future runs of the pipeline in an intelligent fashion.

The current version of `stimela` supports multiprocessing, but a recipe is limited to a single machine (or node). In future we are planning on using tools such as `docker-machine`, which could allow a `stimela` recipe to spawn a cluster of VM instances, equipped with Docker, on cloud platforms such as AWS, GCE, and OpenStack¹, on which to run the recipe in a distributed fashion. The flexibility of these cloud platforms, would allow each task in the recipe to be executed in a VM instance that suits its computing requirements (CPU, RAM or disk space). For example, CPU-intensive tasks would be carried on VM instances that have many CPUs. This is an

¹https://wiki.openstack.org/wiki/Main_Page

obvious next step of this work, as it would lead to computationally and financially efficient radio data processing.

Appendices

Appendix A

Stimela Example Script

A.1 Stimela Data Synthesis and Imaging Recipe

In this section we show how the `stimela` package can be used to write a simulation and imaging pipeline. The aim of the pipeline is to simulate a MeerKAT observation, and then image the simulated data.

The first thing to do in a `stimela` script is to import the `stimela` package.

```
1 | import stimela
```

Next, we define the I/O flow of the pipeline. That is, the input and output directories and files. The pipeline will expect to find all input products in *INPUT* and will dump all the output products to *OUTPUT*. Note that the I/O flow for each task (container) can also be defined/modified when the task is added to the `Recipe` instance.

```
1 | INPUT = "input"
2 | OUTPUT = "output"
3 | PREFIX = "stimela-example"
```

The visibility data that we deal with in radio interferometry is usually stored as a *measurement set* (MS), and this MS is usually both the input and output product for many tasks. Therefore, it is treated differently from the standard input and output directories defined above. Now set the location of the MS file(s), as well as the name of the MS that will be simulated.

```
1 | MSDIR = "msdir"
2 | MSNAME = "meerkat_example_simulation.ms"
```

Next, we set the sky model that will be simulated. In this case, the sky model is catalogue of sources from the NVSS survey. This sky model should be placed in the input folder.

```
1 | SKYMODEL = "nvss1deg.lsm.html"
```

We can now start a new *Recipe* instance.

```
1 | recipe = stimela.Recipe("Stimela simulation example", ms_dir=MSDIR)
```

Note that each *Recipe* is associated with a MS directory, and all MSs that will be simulated or processed in a given recipe should be placed in this directory.

With our *Recipe* instance started, we can proceed to add tasks to the recipe. First, we create a simulated MeerKAT MS using the *simms* executor image.

```
1 | simms_params = {
2 |     "msname" = MSNAME,
3 |     "telescope" = "meerkat",
4 |     "synthesis" = 4 # Synthesis time in seconds
5 |     "intergration" = 5 # integration time in seconds
6 |     "freq0" = "1400MHz" # Start frequency
7 |     "dfreq" = "1MHz",
8 |     "nchan" = 10,
9 | }
10 |
11 | recipe.add("cab/simms", # this is the full name of the executor image
12 |     "create_empty_ms", # container label
13 |     simms_params, # These are the options set above
14 |     input=INPUT,
15 |     output=OUTPUT,
16 |     label="Create empty MS") # Task label. For logging purposes
```

The next step is to generate simulated visibilities from the sky model. We use the *simulator* executor image, which uses *MeqTrees*, to achieve this.

```
1 | simulator_params = {
2 |     "msname" = MSNAME,
3 |     "skymodel" = SKYMODEL,
4 |     "column" = "CORRECTED_DATA",
5 | }
6 |
7 | recipe.add("cab/simulator",
8 |     "simulate_visibilities",
9 |     simulator_params,
10 |     input=INPUT,
11 |     output=OUTPUT,
12 |     label="simulate visibilities into MS")
```

Finally, we image the data. To make things more interesting, let us image the data using Briggs *uv*-weighting with three robustness parameters, -2, 0, 2

```

1 imager_params = {
2     "msname" = MSNAME,
3     "npix" = 2048,
4     "cellsize" = 2 # in arcseconds,
5     "weight" = "briggs"
6     "column" = "CORRECTED_DATA",
7     "clean_iterations" = 3000,
8 }
9
10 for robust in -2,0,2:
11
12     imager_params["robust"] = robust
13     imager_params["prefix"] = "%s-%d"%(prefix, robust)
14
15
16     recipe.add("cab/wsclean",
17         "image_data_robust_%d"%robust,
18         imager_params,
19         input=INPUT,
20         output=OUTPUT,
21         label="Image data. Robust=%d"%robust)

```

With our recipe fully described, we now execute it.

```

1 recipe.start()

```

Also, one can also specify which steps to run in the *Recipe.start()* function via the *steps* argument. For example, to only create an empty MS (step 1), and then generate simulated visibilities (step 2) one would run

```

1 recipe.start(steps=[1, 2])

```

The script can then be executed on command line as

```

1 stimela run <name of script>

```

The output images and the log-file can be found in the output directory, the contents of which are listed in Figure A.1.

```

sphe@nina:/data/sphe/examples/output$ ls -rt
log-meerkat_simulation_example.txt      stimela-example_robust-0-model.fits
stimela-example_robust--2-psf.fits      stimela-example_robust-0-residual.fits
stimela-example_robust--2-dirty.fits    stimela-example_robust-0-image.fits
stimela-example_robust--2-model.fits    stimela-example_robust-2-psf.fits
stimela-example_robust--2-residual.fits stimela-example_robust-2-dirty.fits
stimela-example_robust--2-image.fits    stimela-example_robust-2-model.fits
stimela-example_robust-0-psf.fits       stimela-example_robust-2-residual.fits
stimela-example_robust-0-dirty.fits     stimela-example_robust-2-image.fits

```

Figure A.1: contents of the output directory after the example script was executed.

Bibliography

- [1] F. Bergeaud and S. Mallat. Matching pursuit of images. In *Image Processing, 1995. Proceedings., International Conference on*, volume 1, pages 53–56 vol.1, Oct 1995.
- [2] M. Born and E. Wolf. *Principles of Optics*. Pergamon Press, 1964.
- [3] R. Braun. SKA1 Level 0 Science Requirements. Document Number SKA-TEL.SCI-SKO-SRQ-001, 2014. Available via https://www.skatelescope.org/wp-content/uploads/2014/11/SKA-TEL-SKO-0000007-SCI-REQ-SRQ-Rev01-SKA_Level_0_Science_Requirements.pdf.
- [4] Robert Braun. Ska1 level 0 science requirements. Document Number SKA-TEL.SCI-SKO-SRQ-001 Revision 1, 2014.
- [5] D. S. Briggs. *High Fidelity Deconvolution of Moderately Resolved Sources*. PhD thesis, New Mexico Institute of Mining and Technology, 801 Leroy Pl, Socorro, NM 87801, United States, 7 1995. Available via <http://www.aoc.nrao.edu/dissertations/dbriggs/>.
- [6] D. S. Briggs, F. R. Schwab, and R. A. Sramek. Imaging. In G. B. Taylor, C. L. Carilli, and R. A. Perley, editors, *Synthesis Imaging in Radio Astronomy II*, volume 180 of *Astronomical Society of the Pacific Conference Series*, page 127, 1999.
- [7] W. Bricken. EVLA Memo 58: Using GRASP8 to study the VLA Beam, 2003. Available via <http://www.aoc.nrao.edu/evla/geninfo/memoseries/evlamemo58.pdf>.
- [8] S. Butterworth. On the theory of filter amplifiers. *Experimental Wireless and the Wireless Engineer*, 7:536–541, 1930.
- [9] R. E. Carrillo, J. D. McEwen, and Y. Wiaux. PURIFY: a new approach to radio-interferometric imaging. *Mon. Not. Roy. Astron. Soc.*, 439(4):3591–3604, 2014.

- [10] B. G. Clark. Coherence in Radio Astronomy. In G. B. Taylor, C. L. Carilli, and R. A. Perley, editors, *Synthesis Imaging in Radio Astronomy II*, volume 180 of *Astronomical Society of the Pacific Conference Series*, page 1, 1999.
- [11] J. J. Condon. Confusion and Flux-Density Error Distributions. *The Astrophysical Journal*, 188:279–286, March 1974.
- [12] T. Cornwell and E. B. Fomalont. Self-Calibration. In R. A. Perley, F. R. Schwab, and A. H. Bridle, editors, *Synthesis Imaging in Radio Astronomy*, volume 6 of *Astronomical Society of the Pacific Conference Series*, page 185, 1989.
- [13] T. J. Cornwell. Multiscale clean deconvolution of radio synthesis images. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):793–801, Oct 2008.
- [14] T. J. Cornwell and K. F. Evans. A simple maximum entropy deconvolution algorithm. *Astronomy & Astrophysics*, 143:77–83, February 1985.
- [15] T. J. Cornwell and R. A. Perley. Radio-interferometric imaging of very large fields - The problem of non-coplanar arrays. *Astronomy & Astrophysics*, 261:353–364, July 1992.
- [16] T. J. Cornwell and P. N. Wilkinson. A new method for making maps with unstable radio interferometers. *Monthly Notices of the Royal Astronomical Society*, 196(4):1067–1086, 1981.
- [17] A. Dabbech, C. Ferrari, D. Mary, E. Slezak, and Smirnov O.M. MORESANE: Model reconstruction by synthesis-analysis estimators - deconvolution algorithm for radio interferometric imaging. *A&A*, 2014.
- [18] P. Dewdney. SKA1 System Baseline Design. Document Number SKA-TEL-SKO-DD-001, 2014. Available via https://www.skatelescope.org/wp-content/uploads/2013/05/SKA-TEL-SKO-DD-001-1_BaselineDesign1.pdf.
- [19] FEKO. Feko, 2016.
- [20] C. Ferrari, A. Dabbech, O.M. Smirnov, S. Makhathini, J.S. Kenyon, M. Murgia, F. Govoni, D. Mary, E. Slezak, F. Vazza, A. Bonafede, A. Bruggen, M. Johnston-Hollitt, S. Dehghan, L. Feretti, G. Giovannini, V. Vacca, M. Wise, M. Gitti, M. Arnaud, G. Pratt, K.Z. Adami, and S. Colafrancesco. Non-thermal emission from galaxy clusters: A feasibility study with ska1. In proceedings of “Advancing Astrophysics with the Square Kilometre Array”, PoS(AASKA14)075, 2015.
- [21] T. M. O. Franzen, C. A. Jackson, A. R. Offringa, R. D. Ekers, R. B. Wayth, G. Bernardi, J. D. Bowman, F. Briggs, R. J. Cappallo, A. A. Deshpande, B. M. Gaensler, L. J. Greenhill, B. J. Hazelton, M. Johnston-Hollitt, D. L. Kaplan, C. J. Lonsdale, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, J. Morgan,

- D. Oberoi, S. M. Ord, T. Prabu, N. Seymour, N. Udaya Shankar, K. S. Srivani, R. Subrahmanyam, S. J. Tingay, C. M. Trott, R. L. Webster, A. Williams, and C. L. Williams. The 154 mhz radio sky observed by the murchison widefield array: noise, confusion, and first source count analyses. *Monthly Notices of the Royal Astronomical Society*, 459(3):3314–3325, 2016.
- [22] Garsden, H., Girard, J. N., Starck, J. L., Corbel, S., Tasse, C., Woiselle, A., McKean, J. P., van Amesfoort, A. S., Anderson, J., Avruch, I. M., Beck, R., Bentum, M. J., Best, P., Breitling, F., Broderick, J., Brggen, M., Butcher, H. R., Ciardi, B., de Gasperin, F., de Geus, E., de Vos, M., Duscha, S., Eislffiel, J., Engels, D., Falcke, H., Fallows, R. A., Fender, R., Ferrari, C., Frieswijk, W., Garrett, M. A., Griemeier, J., Gunst, A. W., Hassall, T. E., Heald, G., Hoeft, M., Hrandel, J., van der Horst, A., Juette, E., Karastergiou, A., Kondratiev, V. I., Kramer, M., Kuniyoshi, M., Kuper, G., Mann, G., Markoff, S., McFadden, R., McKay-Bukowski, D., Mulcahy, D. D., Munk, H., Norden, M. J., Orru, E., Paas, H., Pandey-Pommier, M., Pandey, V. N., Pietka, G., Pizzo, R., Polatidis, A. G., Renting, A., Rttgering, H., Rowlinson, A., Schwarz, D., Sluman, J., Smirnov, O., Stappers, B. W., Steinmetz, M., Stewart, A., Swinbank, J., Tagger, M., Tang, Y., Tasse, C., Thoudam, S., Toribio, C., Vermeulen, R., Vocks, C., van Weeren, R. J., Wijnholds, S. J., Wise, M. W., Wucknitz, O., Yatawatta, S., Zarka, P., and Zensus, A. Lofar sparse image reconstruction. *A&A*, 575:A90, 2015.
- [23] T. L. Grobler, C. D. Nunhokee, O. M. Smirnov, A. J. van Zyl, and A. G. de Bruyn. Calibration artefacts in radio interferometry - i. ghost sources in westerbork synthesis radio telescope data. *Monthly Notices of the Royal Astronomical Society*, 439(4):4030–4047, 2014.
- [24] J. P. Hamaker, J. D. Bregman, and R. J. Sault. Understanding radio polarimetry. I. Mathematical foundations. *A&A*, 117:137–147, May 1996.
- [25] A. Hewish, S. J. Bell, J. D. H. Pilkington, P. F. Scott, and R. A. Collins. Observation of a Rapidly Pulsating Radio Source. *Nature*, 217:709–713, February 1968.
- [26] I. Heywood, M. J. Jarvis, A. J. Baker, K. W. Bannister, C. S. Carvalho, M. Hardcastle, M. Hilton, K. Moodley, O. M. Smirnov, D. J. B. Smith, S. V. White, and E. J. Wollack. A deep/wide 12 ghz snapshot survey of sdss stripe 82 using the karl g. jansky very large array in a compact hybrid configuration. *Monthly Notices of the Royal Astronomical Society*, 460(4):4433–4452, 2016.
- [27] M. Hoare. Ska and the cradle of life. In proceedings of “Advancing Astrophysics with the Square Kilometre Array”, PoS(AASKA14)115, 2015.
- [28] J. A. Högbom. Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. *A&A*, 15:417, June 1974.

- [29] B. W. Holwerda, S. L. Blyth, and A. J. Baker. Looking at the distant universe with the MeerKAT Array (LADUMA). *IAU Symp.*, 284:496–499, 2012.
- [30] H. T. Intema. SPAM: Source Peeling and Atmospheric Modeling. Astrophysics Source Code Library, August 2014.
- [31] J. P. Hamaker. Understanding radio polarimetry. *A&A*, 456(1):395–404, 2006.
- [32] C. M. Jansky. The discovery and identification by karl guthe jansky of electromagnetic radiation of extraterrestrial origin in the radio spectrum. *Proceedings of the IRE*, 46(1):13–15, Jan 1958.
- [33] M. Johnston-Hollitt et al. Using ska rotation measures to reveal the mysteries of the magnetised universe. In proceedings of “Advancing Astrophysics with the Square Kilometre Array”, PoS(AASKA14)092, 2015.
- [34] R. Clark Jones. A new calculus for the treatment of optical systems. description and discussion of the calculus. *J. Opt. Soc. Am.*, 31(7):488–493, Jul 1941.
- [35] H. Junklewitz, M.A. Bell, and EnBlin T. RESOLVE: A new algorithm for aperture synthesis imaging of extended emission in radio astronomy. *preprint(arxiv:1311.5282)*, 2013.
- [36] L. Koopmans et al. The cosmic dawn and epoch of reionisation with ska. In proceedings of “Advancing Astrophysics with the Square Kilometre Array”, PoS(AASKA14)001, 2015.
- [37] Michelle Lochner, Iniyan Natarajan, Jonathan T. L. Zwart, Oleg Smirnov, Bruce A. Bassett, Nadeem Oozeer, and Martin Kunz. Bayesian inference for radio observations. *Monthly Notices of the Royal Astronomical Society*, 450(2):1308–1319, 2015.
- [38] Michelle Lochner, Iniyan Natarajan, Jonathan T. L. Zwart, Oleg Smirnov, Bruce A. Bassett, Nadeem Oozeer, and Martin Kunz. Bayesian inference for radio observations. *Monthly Notices of the Royal Astronomical Society*, 450(2):1308–1319, 2015.
- [39] S. Makhathini. simms: Casa based tool for creating empty visibility datasets, 2015. Available via <https://github.com/radio-astro/simms>.
- [40] Sphehile Makhathini, Oleg Smirnov, Matt Jarvis, and Ian Heywood. Morphological classification of radio sources for galaxy evolution and cosmology with ska-mid. In proceedings of “Advancing Astrophysics with the Square Kilometre Array”, PoS(AASKA14)81, 2015.

- [41] I. Marti-Vidal, W. H. T. Vlemmings, S. Muller, and S. Casey. UVMULTIFIT: Fitting astronomical radio interferometric data. Astrophysics Source Code Library, February 2014.
- [42] M. Mitra, S. Makhathini, G. Foster, O. Smirnov, and R. Perley. Incorporation of antenna primary beam patterns in radio-interferometric data reduction to produce wide-field, high-dynamic-range images. In *Electromagnetics in Advanced Applications (ICEAA), 2015 International Conference on*, pages 494–497, Sept 2015.
- [43] B.J. Mort, F. Dulwich, S. Salvini, K.Z. Adami, and M.E. Jones. Oskar: Simulating digital beamforming for the ska aperture array. In *Phased Array Systems and Technology (ARRAY), 2010 IEEE International Symposium on*, pages 690–694, Oct 2010.
- [44] C. A. MULLER and J. H. OORT. Observation of a line in the galactic radio spectrum: The interstellar hydrogen line at 1,420 mc./sec., and an estimate of galactic rotation. *Nature*, 168(4270):357–358, Sep 1951.
- [45] J. E. Noordam and O. M. Smirnov. The meqtrees software system and its use for third-generation calibration of radio interferometers. *A&A*, 524:A61, 2010.
- [46] NRAO. Aips: Astronomical image processing system, 2016. Available via <http://www.aips.nrao.edu/index.shtml>.
- [47] A. R. Offringa, B. McKinley, N. Hurley-Walker, F. H. Briggs, R. B. Wayth, D. L. Kaplan, M. E. Bell, L. Feng, A. R. Neben, J. D. Hughes, J. Rhee, T. Murphy, N. D. R. Bhat, G. Bernardi, J. D. Bowman, R. J. Cappallo, B. E. Corey, A. A. Deshpande, D. Emrich, A. Ewall-Wice, B. M. Gaensler, R. Goeke, L. J. Greenhill, B. J. Hazelton, L. Hindson, M. Johnston-Hollitt, D. C. Jacobs, J. C. Kasper, E. Kratzenberg, E. Lenc, C. J. Lonsdale, M. J. Lynch, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, N. Kudryavtseva, D. Oberoi, S. M. Ord, B. Pindor, P. Procopio, T. Prabu, J. Riding, D. A. Roshi, N. U. Shankar, K. S. Srivani, R. Subrahmanyam, S. J. Tingay, M. Waterson, R. L. Webster, A. R. Whitney, A. Williams, and C. L. Williams. WSCLEAN: an implementation of a fast, generic wide-field imager for radio astronomy. *Monthly Notices of the Royal Astronomical Society*, 444:606–619, October 2014.
- [48] A. R. Offringa, J. J. van de Gronde, and J. B. T. M. Roerdink. A morphological algorithm for improved radio-frequency interference detection. *A&A*, 539, March 2012.
- [49] A. A. Penzias and R. W. Wilson. A Measurement of Excess Antenna Temperature at 4080 Mc/s. *The Astrophysical Journal*, 142:419–421, July 1965.

- [50] S. Perkins, P. Marais, J. Zwart, I. Natarajan, C. Tasse, and O. Smirnov. Montblanc: GPU accelerated Radio Interferometer Measurement Equations in support of Bayesian Inference for Radio Observations. Astrophysics Source Code Library, February 2015.
- [51] U. Rau and T. J. Cornwell. A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry. *Astronomy & Astrophysics*, 532:A71, August 2011.
- [52] G. Reber. Cosmic Static. *The Astrophysical Journal*, 100:279, November 1944.
- [53] A. Rossetti, F. Mantovani, D. Dallacasa, W. Junor, C. J. Salter, and D. J. Saikia. VLBA polarimetric observations of the CSS quasar 3C147. *Astron. Astrophys.*, 504:741, 2009.
- [54] J. E. Ruiz, L. Verdes-Montenegro, S. Sánchez, J. D. Santander-Vela, and J. Garrido. Digital Science: reproducibility and visibility in Astronomy. In J. C. Guirado, L. M. Lara, V. Quilis, and J. Gorgas, editors, *Highlights of Spanish Astrophysics VII*, pages 910–915, May 2013.
- [55] M. Ryle and D. D. Vonberg. Solar Radiation on 175 Mc./s. *Nature*, 158:339–340, September 1946.
- [56] M. L. L. Sebokolodi. Automation of source-artefact classification. Master’s thesis, Rhodes University, Drostdy Rd, Grahamstown, 6139, Western Cape, South Africa, 5 2016.
- [57] P. Serra, R. Jurek, and L. Flöer. Using Negative Detections to Estimate Source-Finder Reliability. *Publications of the Astronomical Society of Australia*, 29:296–300, February 2012.
- [58] L. Shao et al. Testing gravity with pulsars in the ska era. In proceedings of “Advancing Astrophysics with the Square Kilometre Array”, PoS(AASKA14)042, 2015.
- [59] O. M. Smirnov. Revisiting the radio interferometer measurement equation. I. A full-sky Jones formalism. *Astronomy & Astrophysics*, 527:A106, March 2011.
- [60] Smirnov, O. M. Revisiting the radio interferometer measurement equation. II. Calibration and direction-dependent effects. *A&A*, 527:A107, 2011.
- [61] Smirnov, O. M. Revisiting the radio interferometer measurement equation. III. Addressing direction-dependent effects in 21 cm WSRT observations of 3C 147. *A&A*, 527:A108, 2011.

- [62] Tsutomu T. Takeuchi and Takako T. Ishii. A general formulation of the source confusion statistics and application to infrared galaxy surveys. *The Astrophysical Journal*, 604(1):40, 2004.
- [63] C. Tasse. Ddfacet. In Prep., 2016.
- [64] CASA team. About casa, 2004. Available via [//casa.nrao.edu/](http://casa.nrao.edu/).
- [65] Casarest team. Casacore, casarest, 2016. Available via <https://github.com/casacore/casarest>.
- [66] Docker team. What is docker, 2013. Available via <https://www.docker.com/what-docker>.
- [67] MeerKAT Team. Meerkat spec and schedule, 2015. Available via [//public.ska.ac.za/meerkat](http://public.ska.ac.za/meerkat).
- [68] A. R. Thompson, M. J. Moran, and G.W. Swenson Jr. *Interferometry And Sythesis Imaging in Radio Astronomy. 2nd Edition*. Wile-VCH Verlag GmbH & Co. KGaA, 2001.
- [69] P. N. Wilkinson, A. C. S. Readhead, G. H. Purcell, and B. Anderson. Radio structure of 3C147 determined by multi-element very long baseline interferometry. *Nature*, 269:764–768, October 1977.
- [70] R. J. Wilman, L. Miller, M. J. Jarvis, T. Mauch, F. Levrier, F. B. Abdalla, S. Rawlings, H.-R. Klöckner, D. Obreschkow, D. Olteanu, and S. Young. A semi-empirical simulation of the extragalactic radio continuum sky for next generation radio telescopes. *Monthly Notices of the Royal Astronomical Society*, 388:1335–1348, August 2008.
- [71] Xin Yan and Xiao Gang Su. *Linear Regression Analysis: Theory and Computing*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2009.
- [72] S. Yatawatta. Adaptive weighting in radio interferometric imaging. *Monthly Notices of the Royal Astronomical Society*, 444(1):790–796, 2014.
- [73] S. Yatawatta, S. Zaroubi, G. de Bruyn, L. Koopmans, and J. Noordam. Radio Interferometric Calibration Using The SAGE Algorithm. *ArXiv e-prints*, October 2008.