Evaluation and Demonstration of THEP as a Radio Astronomical Observing Facility

Stefan J. Wijnholds
Supervisors: Prof. Dr. A. Ger de Bruyn and Ir. Jaap D. Bregman

May 2, 2003

Abstract

The Thousand Element Array (THEA) is a phased array system which was developed as a demonstrator in the framework of the Square Kilometer Array (SKA) at ASTRON. After a thorough description of the THEA system, results from a number of evaluation and demonstration measurements using four tiles having a total of 256 receiving elements will be presented. These measurements highlighted aspects such as array beam forming, multifrequency multibeaming, calibration on a celestial source, detection of galactic neutral hydrogen and solar fringes. Based on these measurements a first assessment of the quality of the system was made. Furthermore the results have shown that the THEA system is suitable for use as an educational tool.

Preface

In May 2002 I started looking for a graduation project toward instrumental astronomy. This turned out to be the perfect time to become involved in THEA, since Bas van der Tol was still working on his internship on THEA at ASTRON. This allowed me to get acquainted with the THEA system rather quickly, thus identifying the work which could be done in the framework of my graduation project. The actual work was done during the period from September 2002 until April 2003. This report concludes my thesis work.

The foundation ASTRON exists to promote research in astronomy in the Netherlands. Its strategy to do this is to provide front-line observing facilities for Dutch astronomers across a broad range of frequencies and techniques. To this end, ASTRON builds and operates the radio telescope at Westerbork, one of the largest and most sensitive radio telescopes in the world. ASTRON also develops advanced instrumentation for studying the content, structure and evolution of the Universe, and for exploring exotic physical phenomena that cannot be studied in the laboratory. Its ambition for the future is to help develop the technologies necessary for a new generation of sensitive radio telescopes like LOFAR and SKA, capable of detecting objects over the entire past history of the Cosmos.

At this place I would like to acknowledge the support of my work by a number of people. The information supplied by Bas van der Tol during the introductory period has greatly accelerated the process of getting familiar with the THEA system. This in turn helped me making a suitable plan for my thesis work. During measurements Yde Koopmans and Klaas Dijkstra were always ready to solve problems with the frontend while support on the backend was given by Andre Gunst and Gijs Schoonderbeek. Initial help on the GUI's and the Reduction, Acquisition and Processing unit was provided by Harm-Jan Pepping who also worked on the software for the new backend together with Arie Doorduin, Gijs Schoonderbeek, Andreas Honneffer and Dion Kant. The latter and Sieds Damstra also solved a number of other computer problems.

I had a number of helpful discussions on physical issues with Michel Arts, Robert Braun, Dion Kant, Jan-Geralt bij de Vaate and Bert Woestenburg. Finally I would like to thank my supervisors Jaap Bregman and Ger de Bruyn.

Stefan J. Wijnholds,
Dwingeloo, April 21, 2003

Contents

Preface **3**

1 Introduction **7**

2 Theory **9**

2.1 Introduction . . . 9

2.2 Coordinate transformations . . . 10

2.2.1 Conversion between (θ, ϕ) and (u, v) . . . 10

2.2.2 Conversion between (u, v) and (az, el) . . . 10

2.2.3 Conversion between (u, v) and (α, δ) . . . 11

2.2.4 Conversion between (α, δ) and (l, b) . . . 12

2.3 Phased array principles . . . 12

3 Description of the THEA base system **15**

3.1 Introduction . . . 15

3.2 The THEA setup . . . 15

3.2.1 Overview . . . 15

3.2.2 Frontend . . . 16

3.2.3 The HSL . . . 19

3.2.4 The ADBF . . . 19

3.2.5 The cache storage and selection board . . . 21

3.2.6 The RAP unit . . . 21

3.3 Control software . . . 23

4 The microprocessor correlator **25**

4.1 Introduction . . . 25

4.2 Description of the MPC . . . 26

4.2.1 Overview of the new THEA system . . . 26

4.2.2 Ongoing activities . . . 28

4.2.3 Future improvements . . . 28

4.3 Comparison with the old backend . . . 29

5 Experiments **31**

5.1 Introduction . . . 31

5.2 Counting bits and bytes . . . 31

5.3 Correlator noise . . . 32

5.4 Tile calibration . . . 34

5.5 Vector modulator responses 37
5.6 Measurement of the sky background continuum 39
5.7 Array beam Forming 41
5.8 Multifrequency multibeaming 43
5.9 Reduction of LDS data 43
5.10 Imaging of galactic neutral hydrogen with a singe tile 46
5.11 Imaging of galactic neutral hydrogen with four tiles 49
5.12 Solar fringes 50

6 Summary and conclusions 53

7 Further work 55
7.1 Deterministic nulling and beam forming 55
7.2 Outdoor calibration in the presence of multiple sources 56
7.3 Correct beam forming over the full 20MHz band 56
7.4 RFI mitigation at the input 57
7.5 Synthesis imaging 58
7.6 Polarization experiments 58
7.7 Drift 59
7.8 Shadowing 59

Bibliography 62

A Abbreviations 63

B Matlab sources 65
B.1 initTile.m 65
B.2 AppendBeams.m 65
B.3 uvscan.m 66
B.4 uvread.m 68
B.5 setbeamf.m 69
B.6 startCorrelator.m 69
B.7 uvtoxy.m 70
B.8 xytouv.m 71

C C++ sources 73
C.1 daq.h 73
C.2 correlator.h 74
C.3 mpc.cc 76

Chapter 1

Introduction

The international radio astronomy community is currently making detailed plans for the development of a new radio telescope: the Square Kilometer Array (SKA). This instrument will be a hundred times more sensitive than telescopes currently in use. ASTRON is developing one of the options for this new synthesis telescope, using antenna stations with phased array technology consisting of over one million receiving elements with a mixed RF/digital adaptive beam former. The Thousand Element Array (THEA) is one of the demonstrator systems that was build during the SKA development program and is officially continued as the THEA Experimental Platform (THEP), but is usually still named THEA.

THEP is an out-door phased array system with 256 broadband receiving antenna elements. Beam steering is done by a combination of RF and digital beam forming. THEA is able to make multiple beams over the sky simultaneously. In addition, adaptive beam forming may be applied to suppress EMI sources. THEA will also be used for educational purposes and for simple experiments concerning the phased array technique.

The development of THEA has focused on getting the phased array measurement concept to work. This approach has led to an impracticable measurement system without proper documentation. The aim of my thesis work was threefold:

1. further integration to a single measurement system
2. writing proper user-oriented documentation
3. conduction of a number of evaluation and demonstration measurements

A number of steps were identified to achieve these goals. First a number of evaluation measurements would be done. This would lead to a reasonable level of understanding of the current THEA system and would force to take the first steps toward further integration of the THEA system. It would also provide the information needed to write the user documentation.

During the second stage the integration of the THEA system to a single measurement system would be completed and a new backend based on standard microprocessor chips instead of the specialized chips (FPGA's and DSP's), which are used in the current backend, would be developed.

Finally the available hardware would be used for a number of demonstration measurements. It will be clear that these stages have in practice overlapped. The outline of this thesis therefore doesn't reflect this time line. In the next chapter some theoretical background on the THEA coordinate systems and the phased array concept will be provided. In chapters 3 and 4 the THEA system in its current state will be described. The evaluation and demonstration

measurements are described in chapter 5. After the summary and conclusions this thesis will be concluded by a number of suggestions for further work.
In this thesis I will frequently use and introduce abbreviations. For the convenience of the reader a list of abbreviations has been added in appendix A

Chapter 2

Theory

2.1 Introduction

THEP is a phased array system consisting of four tiles with 64 antenna elements and a size of about 1m^2 each. An image of the THEP system is shown in figure 2.1. The receiver elements pick up signals from every direction. By adjusting the complex gains of the individual elements such that the signals coming from one specific direction add coherently, the phased array antenna can be aimed in a specific direction. This process is called beam steering.

In this chapter the theoretical background of this process will be presented. In order to do so a new coordinate system will be defined which is more convenient than the standard azimuth-elevation coordinate system. In the next section this coordinate system will be introduced and related to the conventional azimuth-elevation coordinate system as well as to the equatorial coordinate system.

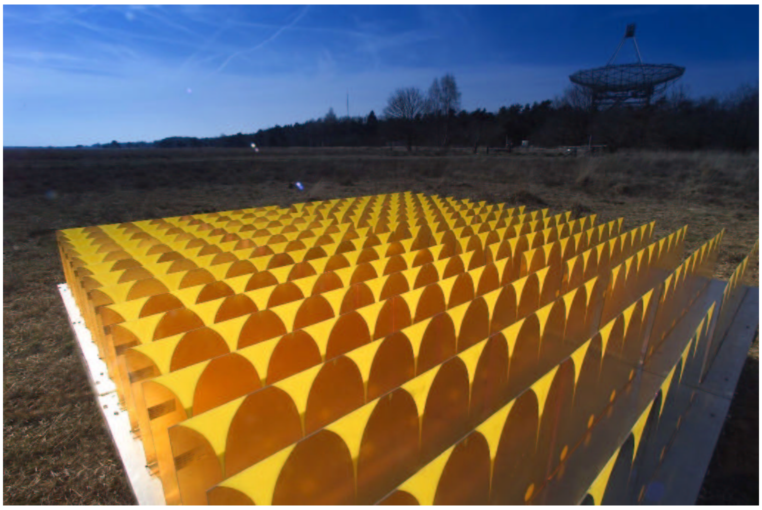


Figure 2.1: Photograph of four THEA tiles in front of the Leiden-Dwingeloo telescope.

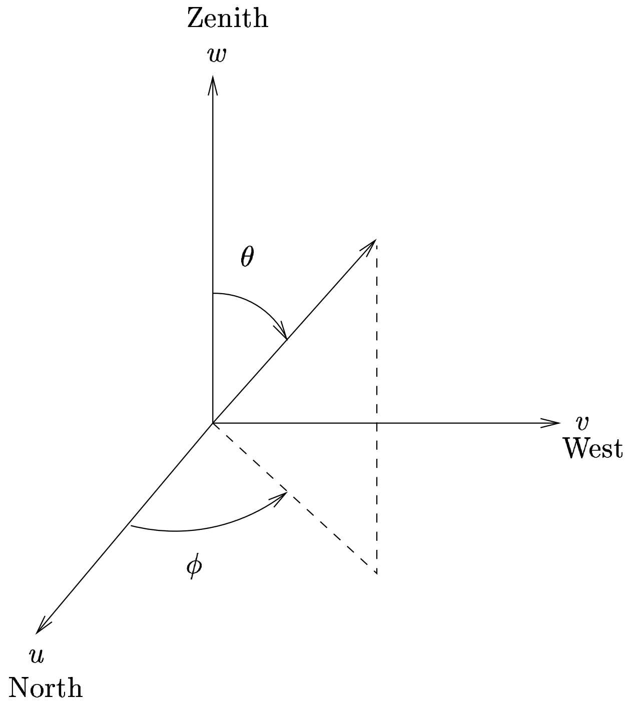


Figure 2.2: Definition of the coordinates of the THEA horizon system

2.2 Coordinate transformations

2.2.1 Conversion between (θ, ϕ) and (u, v)

Coordinates in the horizon system of a THEA tile can either be specified in terms of a zenith angle θ and an azimuthal angle ϕ or in terms of a Cartesian coordinate system (u, v, w) [1]. The relation of these coordinate systems to each other and to the quarters of the compass are shown in figure 2.2.

From this figure it is easy to see that

$$u = \sin\theta\cos\phi \quad (2.1)$$

$$v = \sin\theta\sin\phi \quad (2.2)$$

and that

$$\theta = \sin^{-1}(\sqrt{u^2+v^2}) \quad (2.3)$$

$$\phi = \tan^{-1}\left(\frac{v}{u}\right) \quad (2.4)$$

2.2.2 Conversion between (u, v) and (az, el)

By comparing the definition of θ and ϕ in relation to the quarters of the compass (figure 2.2) with the definition of the horizon coordinates azimuth az and elevation el [2], one will find that

$$\begin{aligned} az &= -\phi && (2.5)\\ el &= 90^\circ - \theta && (2.6) \end{aligned}$$

With these definitions the conversion from (az, el) to (u, v) can be established,

$$\begin{aligned} u &= \cos el \cos az && (2.7)\\ v &= -\cos el \sin az && (2.8) \end{aligned}$$

as well as the corresponding inverse conversion

$$\begin{aligned} az &= -\tan^{-1}\left(\frac{v}{u}\right) && (2.9)\\ el &= \cos^{-1}(\sqrt{u^2+v^2}) && (2.10) \end{aligned}$$

2.2.3 Conversion between (u, v) and (α, δ)

Now that the horizon system of the THEA tiles is clearly defined, we can relate it to the equatorial coordinate system expressed in terms of right ascension α or hour angle HA and declination δ. This can be accomplished by means of standard formulas for conversion between equatorial coordinates and azimuth and elevation [2].

$$\begin{aligned} \sin\delta &= \sin B\cos\theta + \cos B\sin\theta\cos\phi && (2.11)\\ \cos HA &= \frac{\cos\theta - \sin B\sin\delta}{\cos B\cos\delta} && (2.12) \end{aligned}$$

where B is the geographical latitude of the observer.
By combining (2.11) and (2.12) with the definitions for u and v we find

$$\begin{aligned} \sin\delta &= \sqrt{1-u^2-v^2}\sin B + u\cos B && (2.13)\\ \cos HA &= \frac{\sqrt{1-u^2-v^2} - \sin B\sin\delta}{\cos B\cos\delta} && (2.14) \end{aligned}$$

The hour angle of an object can easily be obtained from its right ascension α and the Local Sidereal Time LST:

$$HA = LST - \alpha \qquad (2.15)$$

In this case the Local Sidereal Time needs to be converted to radians or degrees, whichever is used to express the right ascension and the hour angle.

2.2.4 Conversion between (α, δ) and (l, b)

The equatorial coordinate system can in turn be transformed to the galactic coordinate system expressed in terms of galactic longitude l and galactic latitude b. This conversion from equatorial coordinates to galactic coordinates (1950) is given by [2]

$$\begin{aligned} b &= \sin^{-1}(\cos\delta\cos(27.4^\circ)\cos(\alpha - 192.25^\circ) + \sin\delta\sin(27.4^\circ)) & (2.16)\\ l &= \tan^{-1}\left(\frac{\cos\delta\sin(\alpha - 192.25^\circ)}{\cos\delta\sin(27.4^\circ)\cos(\alpha - 192.25^\circ) - \sin\delta\cos(27.4^\circ)}\right) + 33^\circ & (2.17)\end{aligned}$$

while its inverse is given by

$$\begin{aligned} \alpha &= 192.25^\circ + \tan^{-1}\left(\frac{\cos b\sin(33^\circ - l)}{\sin b\cos(27.4^\circ) - \sin(27.4^\circ)\cos b\cos(33^\circ - l)}\right) & (2.18)\\ \delta &= \sin^{-1}(\sin b\sin(27.4^\circ) + \cos b\cos(27.4^\circ)\cos(33^\circ - l)) & (2.19)\end{aligned}$$

2.3 Phased array principles

The signal detected by a receiving element at position $\vec{x}$ can be described as a superposition of plane waves, i.e.

$$s(\vec{x}) = \int_{-\infty}^{\infty} A(\vec{k})e^{-i\vec{k}\cdot\vec{x}}d\vec{k} \tag{2.20}$$

where $A(\vec{k})$ is the plane wave spectrum and the minus sign results from the fact that the element is receiving and not transmitting. Since $\vec{k}$ determines the direction in which the wavefront is propagating, equation (2.20) can also be interpreted as a superposition of signals arriving from different directions $\vec{u}$ on the sky or more explicitly stated

$$s(\vec{x}) = \int_{-\infty}^{\infty} A(|\vec{k}|\vec{u})e^{-i|\vec{k}|\vec{u}\cdot\vec{x}}|\vec{k}|d\vec{u} \tag{2.21}$$

If this signal is probed for every position, the contribution of any arbitrary component may be found by projection on the mutually orthogonal components describing the plane wave spectrum, which can also be interpreted as the response of the phased array when steered to $\vec{u}_0$, $S(\vec{u}_0)$.

$$\begin{aligned} S(\vec{u}_0) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} s(\vec{x})e^{i|\vec{k}|\vec{u}_0\cdot\vec{x}}d\vec{x} & (2.22)\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(|\vec{k}|\vec{u})e^{-i|\vec{k}|(\vec{u}-\vec{u}_0)\cdot\vec{x}}|\vec{k}|d\vec{u}d\vec{x} \\ &= \int_{-\infty}^{\infty} |\vec{k}|A(|\vec{k}|\vec{u})\delta(\vec{u}-\vec{u}_0)d\vec{u} \\ &= |\vec{k}|A(|\vec{k}|\vec{u}_0)\end{aligned}$$

since $\delta(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}d\omega$ [3].

In practice a phased array will sample the signal at a number of specific locations. In the THEA case the elements are even placed on a regular grid. The model may therefore be improved by defining a regular but still infinite array. This can be done by specifying the grid spacing $\Delta\vec{x}$ of the elements. Thus equation (2.22) changes to

$$\begin{aligned} S(\vec{u}_0) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} s(\vec{x})e^{i|\vec{k}|\vec{u}_0\cdot\vec{x}}\delta(\vec{x}-\vec{n}\cdot\Delta\vec{x})d\vec{x} \qquad (2.23)\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(|\vec{k}|\vec{u})e^{-i|\vec{k}|(\vec{u}-\vec{u}_0)\cdot\vec{x}}|\vec{k}|\delta(\vec{x}-\vec{n}\cdot\Delta\vec{x})d\vec{u}d\vec{x}\\ &= \frac{1}{2\pi}\sum_{\vec{n}}\int_{-\infty}^{\infty} |\vec{k}|A(|\vec{k}|\vec{u})e^{-i|\vec{k}|(\vec{u}-\vec{u}_0)\cdot(\vec{n}\cdot\Delta\vec{x})}d\vec{u}\\ &= \frac{1}{\Delta\vec{x}}\sum_{\vec{n}}\int_{-\infty}^{\infty} A(|\vec{k}|\vec{u})\delta\left((\vec{u}_0-\vec{u})-\frac{2\pi\vec{n}}{|\vec{k}|\Delta\vec{x}}\right)d\vec{u}\\ &= \frac{1}{\Delta\vec{x}}\sum_{\vec{n}} A\left(|\vec{k}|(\vec{u}_0+\frac{2\pi\vec{n}}{|\vec{k}|\Delta\vec{x}})\right) \qquad (2.24) \end{aligned}$$

where we have used the fact that $\sum_{n=-\infty}^{\infty}\delta(t+nT)=\frac{1}{T}\sum_{n=-\infty}^{\infty}e^{2\pi int/T}$ [3]. Equation (2.24) shows that the regularity of the array produces a repetition of the plane wave spectrum at intervals of $\Delta\vec{u}=\frac{2\pi}{|\vec{k}|\Delta\vec{x}}=\frac{\lambda}{\Delta\vec{x}}$. These repetitions are called grating lobes. Since the physically relevant range of $\vec{u}$ is limited to values between -1 and 1, this shows that grating lobes may only occur when the wavelength is smaller than two antenna spacings.
In practice the receiving element at $\vec{x}$ will have a complex gain $G(\vec{x})$. Therefore equation (2.23) can be improved further to

$$\begin{aligned} S(\vec{u}_0) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} G(\vec{x})s(\vec{x})e^{i|\vec{k}|\vec{u}_0\cdot\vec{x}}\delta(\vec{x}-\vec{n}\cdot\Delta\vec{x})d\vec{x}\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(\vec{x})A(|\vec{k}|\vec{u})e^{-i|\vec{k}|(\vec{u}-\vec{u}_0)\cdot\vec{x}}|\vec{k}|\delta(\vec{x}-\vec{n}\cdot\Delta\vec{x})d\vec{u}d\vec{x}\\ &= \frac{1}{2\pi}\sum_{\vec{n}}\int_{-\infty}^{\infty} G(\vec{n}\cdot\Delta\vec{x})|\vec{k}|A(|\vec{k}|\vec{u})e^{-i|\vec{k}|(\vec{u}-\vec{u}_0)\cdot(\vec{n}\cdot\Delta\vec{x})}d\vec{u}\\ &\approx \frac{1}{2\pi}\sum_{j}\int_{-\infty}^{\infty} G(\vec{x}_j)|\vec{k}|A(|\vec{k}|\vec{u})e^{-i|\vec{k}|(\vec{u}-\vec{u}_0)\cdot\vec{x}_j}d\vec{u} \qquad (2.25) \end{aligned}$$

where the latter step introduces the limitations of a finite array.
If there is a point source at $\vec{u}_s$, $A(|\vec{k}|\vec{u})=A\delta(\vec{u}-\vec{u}_s)$ where A is a constant related to the brightness of the source. The response of the phased array to this point source $S(\vec{u}_0;\vec{u}_s)$ is thus given by

$$\begin{aligned} S(\vec{u}_0;\vec{u}_s) &= \frac{|\vec{k}|A}{2\pi}\sum_{j}\int_{-\infty}^{\infty} G(\vec{x}_j)\delta(\vec{u}-\vec{u}_s)e^{-i|\vec{k}|(\vec{u}-\vec{u}_0)\cdot\vec{x}_j}d\vec{u}\\ &= \frac{|\vec{k}|A}{2\pi}\sum_{j} G(\vec{x}_j)e^{-i|\vec{k}|(\vec{u}_s-\vec{u}_0)\cdot\vec{x}_j} \qquad (2.26) \end{aligned}$$

The response of the receiving system can be measured for a number of positions $\vec{u}_n$ on the sky, which can be summarized as the vector $\mathrm{U}_0 = (\vec{u}_1, \ldots, \vec{u}_n, \ldots, \vec{u}_N)$. The result can be described by a vector $\mathrm{S}(\mathrm{U}_0; \vec{u}_s)$ which can be evaluated using matrix calculations:

$$\begin{aligned}
\mathrm{S}(\mathrm{U}_0; \vec{u}_s) &= \begin{bmatrix} S(\vec{u}_1; \vec{u}_s) \\ \vdots \\ S(\vec{u}_N; \vec{u}_s) \end{bmatrix} \\
&= \begin{bmatrix} e^{i|\vec{k}|\vec{x}_1 \cdot (\vec{u}_1 - \vec{u}_s)} & \ldots & e^{i|\vec{k}|\vec{x}_J \cdot (\vec{u}_1 - \vec{u}_s)} \\ \vdots & \ddots & \vdots \\ e^{i|\vec{k}|\vec{x}_1 \cdot (\vec{u}_N - \vec{u}_s)} & \ldots & e^{i|\vec{k}|\vec{x}_J \cdot (\vec{u}_N - \vec{u}_s)} \end{bmatrix} \cdot \begin{bmatrix} G(\vec{x}_1) \\ \vdots \\ G(\vec{x}_J) \end{bmatrix} \\
&= \mathrm{C}(\mathrm{U}_0, \mathrm{X}; \vec{u}_s) \cdot \mathrm{G}(\mathrm{X}) \qquad (2.27)
\end{aligned}$$

where $\mathrm{X} = (\vec{x}_1, \ldots, \vec{x}_j, \ldots, \vec{x}_J)^T$ is the vector of the positions of the individual elements. This shows that a holographic measurement on a point source may be used to find the complex gains of the individual elements by inverting equation (2.27) giving

$$\mathrm{G}(\mathrm{X}) = \mathrm{C}^{-1}(\mathrm{U}_0, \mathrm{X}; \vec{u}_s) \cdot \mathrm{S}(\mathrm{U}_0; \vec{u}_s) \qquad (2.28)$$

In general U_0 will not have the same length as X, i.e. the number of measured points will not be equal to the number of antenna elements. In this case $\mathrm{C}(\mathrm{U}_0, \mathrm{X}; \vec{u}_s)$ will not be square and the pseudo-inverse should be used. The pseudo-inverse of a matrix A is defined as $\mathrm{pinv}(\mathrm{A}) = \mathrm{A}^H \cdot (\mathrm{A} \cdot \mathrm{A}^H)^{-1}$ where A^H denotes the Hermite transpose of matrix A which means the transpose of the conjugate of A. This is actually the right pseudo inverse, because $\mathrm{A} \cdot \mathrm{pinv}(\mathrm{A}) = \mathrm{I}$. It takes however only a simple transpose operation to use the same matrix as left pseudo-inverse. The use of the pseudo-inverse is effectively a least-squares regression technique [4].

Chapter 3

Description of the THEA base system

3.1 Introduction

The aim of this chapter is to provide the reader with some in depth knowledge on the THEA setup, which is needed to appreciate the problems pointed out in following chapters. The building blocks of the THEA system described in this chapter were ready for use at the outset. However, during some additional integration of the system necessary to conduct the demonstration and evaluation measurements it turned out that some parts of the software needed to be modified and some additional code needed to be generated to support these measurements. These changes and additions are also included in this description.

3.2 The THEA setup

3.2.1 Overview

A schematic overview of the THEA system is presented in figure 3.1. The THEA system consists of four tiles. Each tile is an array of about one square meter having 64 Vivaldi elements arranged on a rectangular grid of 8 rows and 8 columns. Within each tile beam forming is done on RF level before the normal and quadrature components are converted to a 12 bit signal. The resulting complex samples are sent over an optical point-to-point connection, the high speed link (HSL), to the backend. The THEA tile can be controlled directly from a terminal connection, but there is also a more convenient Matlab interface.

The backend consists of three stages. The adaptive digital beam former (ADBF) can be used to multiply the incoming signals with a complex weighting factor, to add signals together and to multiplex the incoming signals. The result from these operations is written to 16 output channels which form the input signals for the selection board. The ADBF can be controlled from a Matlab GUI.

The selection board is used to select one of the 16 channels at its input. Only one signal can be selected due to bandwidth limitations at the input of the Reduction, Acquisition and Processing (RAP) unit. The selection board can be controlled using a Matlab GUI.

The RAP unit can calculate the complex Fourier transform of the incoming signal and integrate the spectral power density over time. When the incoming signal is actually the result

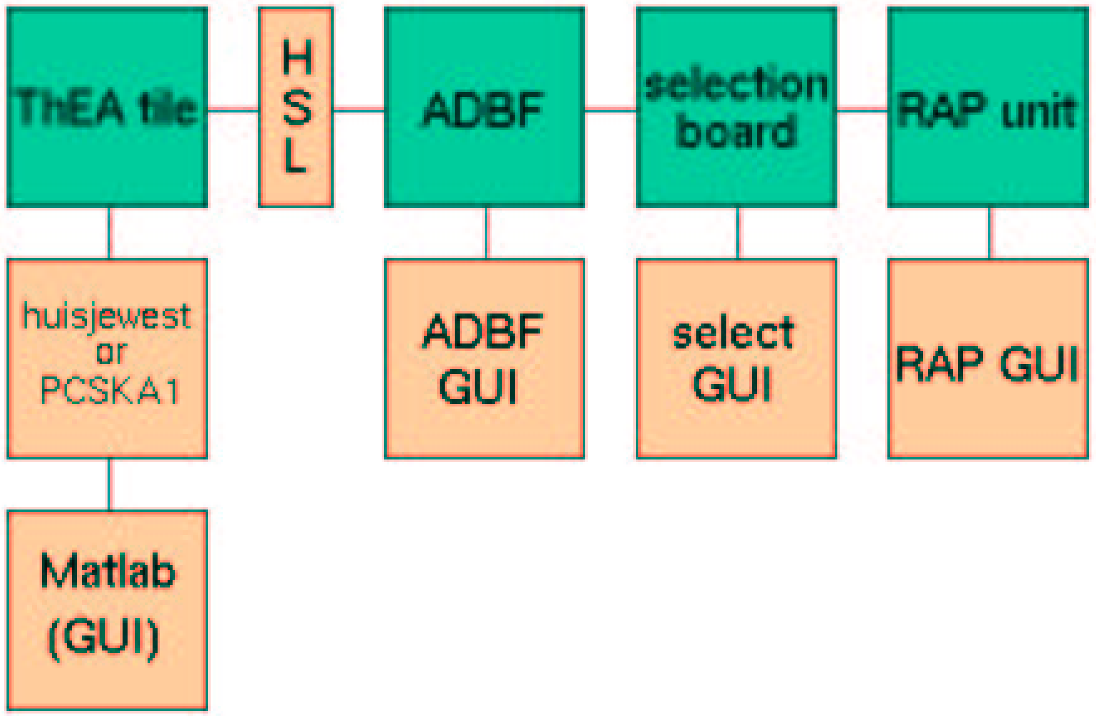


Figure 3.1: Overview of the THEA system

of a multiplexing operation on two signals, the RAP unit can demultiplex these signals and calculate both self spectra or the complex cross spectrum of these two signals. These results can also be integrated over time. The final result will be written to a user specified location. The RAP can be controlled from a Matlab GUI. This GUI however offers only a limited number of measurement options and doesn't provide access to some of the parameters controlling the reduction process. Matlab scripts have been developed to overcome these restrictions.

In the next subsections the aforementioned parts of the system will be described in more detail. The discussion on the software involved in controlling these subsystems will be postponed to the next section.

3.2.2 Frontend

A detailed diagram of a THEA tile is shown in figure 3.2. The 64 Vivaldi elements were produced on prints integrating four elements each. These basic building blocks were added together to get a rectangular matrix of 8 rows and 8 columns of Vivaldi elements. Each Vivaldi element is integrated with its capacitive RF receiver, one low noise amplifier (LNA) stage and two vector modulators. The vector modulators effectively behave like a complex gain which can be set to any arbitrary phase, but a maximum amplitude of unity. Each element has two vector modulators to accommodate dual beaming.

The outputs of the vector modulators are added using a hierarchical scheme, i.e. first the output of pairs of neighboring elements are added, then the resulting signals are added on each print and so on. This is done in the column and row beam formers (CBF and RBF respectively). This whole scheme is implemented separately for each of the two beams as

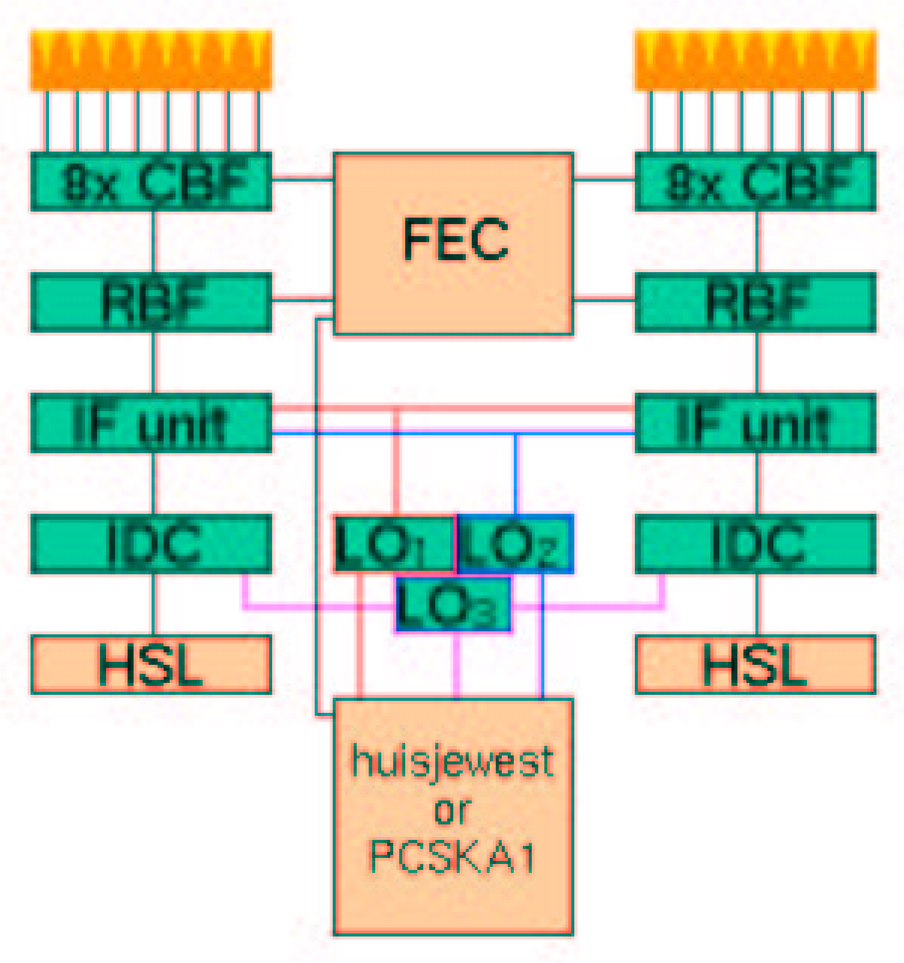


Figure 3.2: Detailed schematic diagram of the THEA frontend

reflected in figure 3.2.

The output signal of the RBF is a superposition of RF signals at all frequencies within the bandwidth of the Vivaldi elements. In principle one can analyze this signal by feeding it to a spectrum analyzer. In fact, this is the way in which measurements were done before the backend became operational and even thereafter a 10dB coupler was placed directly after the RBF in one of the tiles to feed a small part of the power of the signal to a spectrum analyzer as tool to check the operation of the backend and for demonstration purposes. If the backend is to be used, a signal with a bandwidth of 20MHz centered around a specified center frequency f_c should be selected. This is done in a three stage mixing process in the intermediate frequency (IF) unit and the IF to digital converter (IDC).

The IF unit consists of two mixing stages which should produce an output signal which is suitable as input signal for the IDC. This means that the desired frequency band should be shifted to the range 200 to 220MHz. The frequency of the first local oscillator (LO) f_1 can be varied from 3280 to 4380MHz, while the frequency of the second LO f_2 is fixed at 2470MHz. Since the center frequency at the output of the IF unit is specified by the needs of the IDC to be 210MHz, the setting of the first LO can be calculated by

$$f_1 = f_c + f_2 + 210\text{MHz} \tag{3.1}$$

It is also possible to connect the input for the first LO of the two IF units to different LO synthesizers, thereby allowing to tune the beams of the THEA tile to different frequencies, i.e. THEA is able to perform multifrequency multibeam measurements.

In the IDC the measurement band is shifted to the frequency range from 0 to 20MHz by mixing with the signal from a third LO which is tuned to 200MHz. The true and quarter wave (or I and Q) components of this signal are 12 bit sampled at a rate of 40MHz. The IDC also has a digital bandpass filter which removes signals with a frequency higher than 20MHz, i.e. signals lying outside the bandwidth of the THEA system. The resulting 24bit signal is fed to the input of the HSL transmitter at a rate of 40MHz.

The THEA tile is controlled from a computer which communicates with the frontend controller (FEC) over an RS422 connection. The FEC offers functionality to set the vector modulators, to turn the LNA's on and off and to check the settings of the tile and its temperature [5]. It also offers a command interface to the IF unit. Furthermore it is capable of storing 1400 beams in its memory. This allows the user to calculate all the beams needed during a measurement in advance.

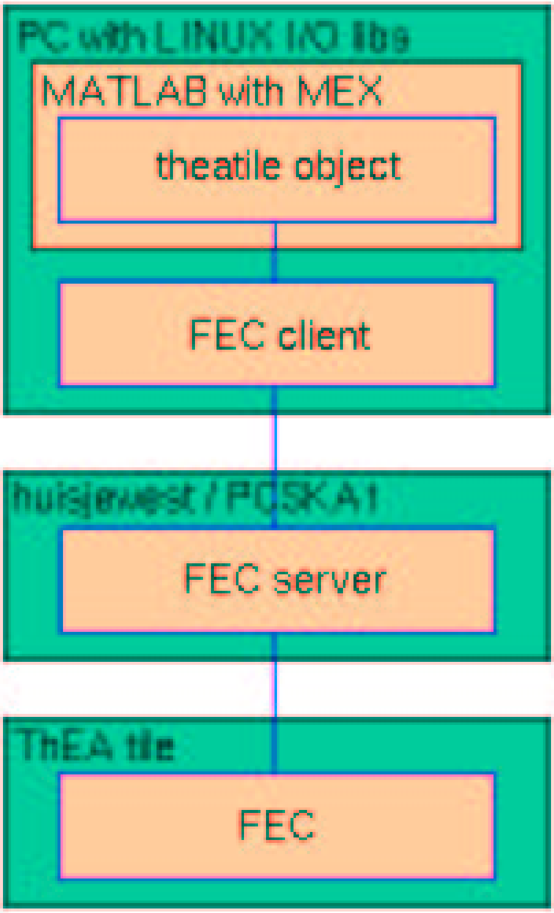


Figure 3.3: Overview of the control structure of the THEA frontend

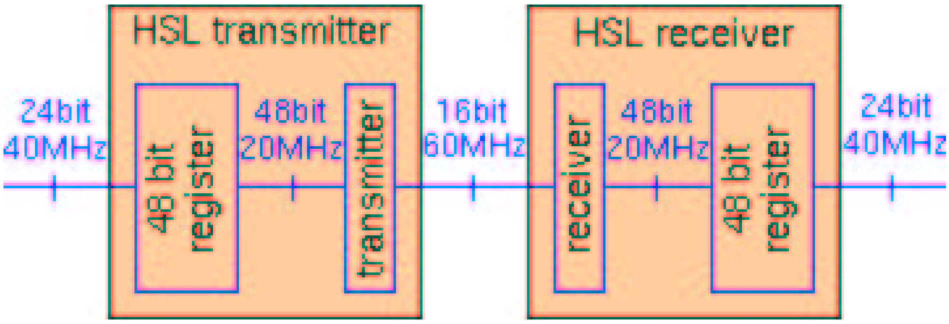


Figure 3.4: The operation of the HSL

At the most basic level the user has access to the FEC by opening a terminal on the RS422 connection. At a higher and more convenient level there is a home written FEC server available which offers an interface to the FEC at a somewhat higher level using a FEC client. This client is in turn implemented as a MEX routine which is accessible from the Matlab command prompt. This feature allows the user to integrate the control of the THEA tile in Matlab functions. A schematic overview of this structure is shown in figure 3.3.

The LO synthesizers are controlled over a GPIB interface from a small command line program, which allows the user to set the power level and the frequency and to switch the carrier on and off. There are no facilities available to integrate this program into the Matlab environment. The only way to use this program remotely is to open a shell to the computer connected to the LO's.

3.2.3 The HSL

An overview of the HSL is shown in figure 3.4. The samples from the frontend are written to a 48bit register, thereby converting the original 24bit signal at 40MHz to a 48bit signal at 20MHz. This signal is transmitted over an optical fiber as a 16bit 60MHz signal.

On the receiver side of the HSL the inverse procedure is followed to regain the original 24bit 40MHz signal. Here the 16bit 60MHz signals are written to an 48bit register thereby effectively creating a 48bit signal at 20MHz. The data in this register are written at 40MHz to the ADBF board, thereby restoring the original 24bit signal.

3.2.4 The ADBF

The 24bit stream from the HSL is mapped on a 32bit stream in such a way that the original two times 12bit signal has now become a two times 16bit signal, where the original 12 bits are stored in the least significant bits and the four most significant bits are set to zero.

An overview of the ADBF board [6][7] is shown in figure 3.5. The structure shown in this figure is implemented twice on the ADBF. These systems are completely independent each having four input channels. These input channels can either be linked to the HSL receiver units or to a test signal, which is simply a memory buffer containing a user specified data set. The input snapshot buffer is used to store snapshots of the incoming data for inspection by the user. This buffer contains a copy of a snapshot of the data at a specific time and is therefore not interfering with the measurement process.

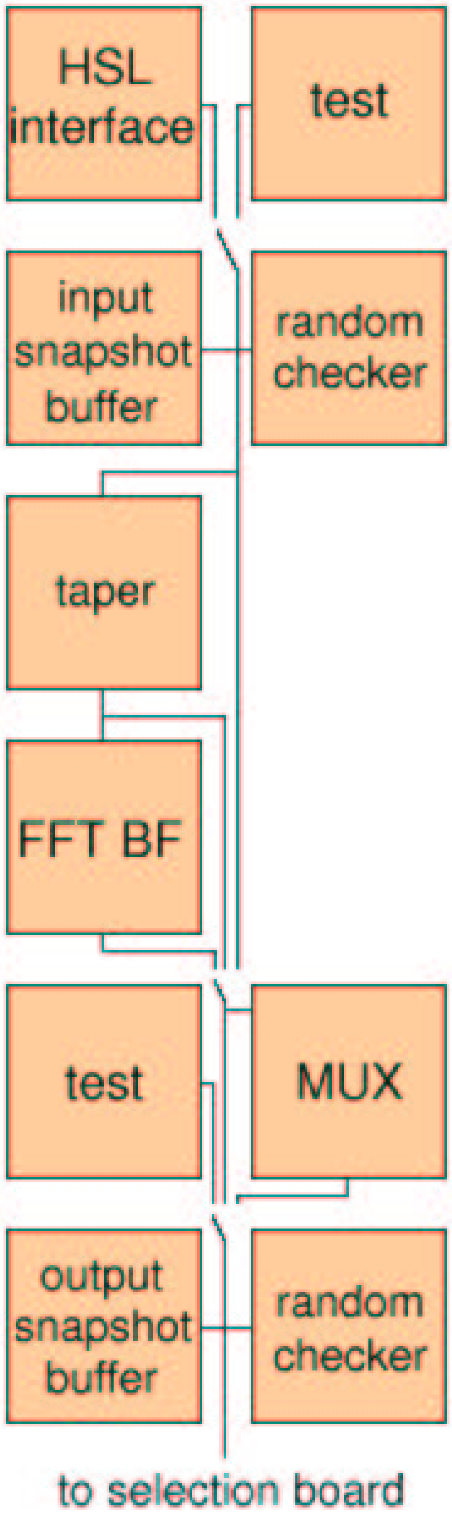


Figure 3.5: Internal structure of the adaptive digital beam former

The data then arrives at the beam forming part of the ADBF. The taper is implemented as a spatial taper, i.e. different signals can be given different complex weights. The word “spatial” in the previous sentence therefore refers to the spatial configuration of the tiles in use. This idea is also reflected in the implementation of the fast Fourier transform (FFT) beam former (BF), where a two dimensional Fourier transform is implemented where the four input signals are assumed to come from a 2x2 array of THEA tiles. In practice these beam forming features were only used to attenuate incoming signals having such a high amplitude that these signals may cause overflows in the RAP unit and to add two or more signals together.

The switch before the multiplexer (MUX) block represents that a decision has to be made whether the original input signals or the weighted input signals or the outputs of the FFT BF are passed to the next processing step. The next processing step is the MUX, which has four inputs and eight outputs. The multiplexer is capable of multiplexing any two of the four input signals together to form one single output signal. Note that this operation is permitted because the system is working at 40MHz using complex samples, while only 20MHz of complex samples are needed to satisfy the Nyquist criterion. The combinations of signals which are written to the outputs depend on the operation mode of the ADBF specified by the user.

Instead of the output signals from the multiplexer, the user may specify a specific test output or pass the input signals through without multiplexing. In either case the signals can be verified at the output using the output snapshot buffer. Again the taking of snapshots doesn’t interfere with the measurement process. The resulting sixteen signals are all passed to the cache storage and selection board.

3.2.5 The cache storage and selection board

The cache storage and selection board [8][9] has 16 inputs each having a bandwidth of 32bit at 40MHz but only a single output with a bandwidth of 32bit at 40MHz. Therefore the original purpose of the selection board was to act as a memory buffer capable of holding a few seconds of data and then writing the user specified buffer to the RAP unit. During system tests on the memory board it turned out that the memory could not be read correctly, so now the only function of the cache storage and selection board is to link one user specified channel to its output. During this operation the signal can be down sampled from 40MHz to 20MHz on-the-fly.

3.2.6 The RAP unit

The Reduction, Acquisition and Processing (RAP) unit [10] is the last stage of the THEA backend. The RAP itself consists of two stages. The first stage is a PMP8 board with 8 DSP’s which reduce the data stream to a rate which can be sustained by the PCI-bus in the RAP PC. The PC performs further integrations and finally saves the data. The schedule in figure 3.6 shows a more detailed overview of the RAP unit.

The incoming 32bit signal at 40MHz can either be the signal from one single beam at 20MHz, the signal from one single beam at 40MHz or the result of the multiplexing operation on two signals coming from two beams sampled at 20MHz giving a total data rate of 40MHz.

The RAP unit has no way to detect which case is valid. It is therefore the responsibility of the user to make the right selections. The first selection is whether the RAP unit has to deal with a single signal (black path) or with two multiplexed signals (blue path).

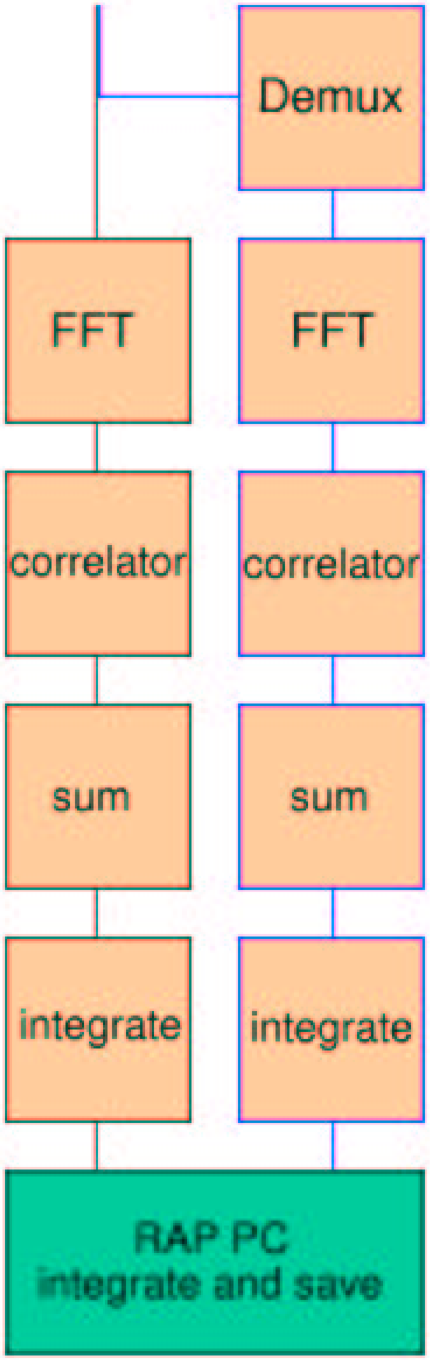


Figure 3.6: Diagram showing the functional blocks of the RAP unit.

Suppose that it has to deal with a single signal. In that case the RAP unit starts by reading 4 blocks of 1024 complex samples. The size of the blocks is more or less fixed. It can be changed in a configuration file, but that feature has never been tested. The four blocks of data will be Fourier transformed independently. Since no scaling is done during the FFT,

this means that the FFT has to add 1024 numbers. Since addition of two 12bit numbers may produce a 13bit number as result, addition of 1024 or 2^{10} 12bit numbers may produce a 22 bit result. Since the FFT is performed in a 16bit register this may cause some unwanted results due to overflow. Therefore the data is shifted over a user specified number of bits before the FFT is performed.
The autocorrelation is calculated by simply calculating the real part, since the imaginary part of an autocorrelation is always zero. Before this operation is performed the data is first converted from 16bit to 32bit. After this operation the data will be using the full dynamic range of a 32bit integer, assuming that the user has adjusted the bitshift at the input critically. Since it is assumed that the most significant bit is set after the correlation, there is another shift directly after the correlation. Immediately after this shift the four blocks which are processed in this run are added together before they are added to the final result in the integrator.
After the DSP integrations the data is transferred over a PCI-bus to the RAP PC. This computer performs additional integrations in floating point, which prevents integer overflows even by long integration times and offers a dynamic range not depending on signal level, i.e. on the number of counts. When the signal is integrated long enough the data will be saved to the hard disk.
If two multiplexed signals are fed into the RAP unit the processing scheme is slightly different. The DSP will still read four blocks of 1024 samples. These 4096 samples are demultiplexed to two times two blocks of 1024 samples, which are treated separately in the first stages. These blocks are Fourier transformed and then correlated. This correlation step can produce either both autocorrelations or the crosscorrelation of the two signals. In the first case the data is stored in complex format where the real part represents the results obtained from the first signal and the imaginary part represents those obtained from the second signal. In the latter case two blocks are complex multiplied giving complex results. Note that in either case this produces output blocks of 1024 complex samples.
These blocks are then summed and integrated further, before leaving the DSP board. The actions on the RAP PC are then the same as in the case of a single signal, i.e. further integration and storage of the data.

3.3 Control software

In this section a number of limitations of the THEA software and thereby implied changes will be discussed. As the reader may deduce from the description above, a normal observation run consists of a number of stages. First the LO synthesizers have to be set, then the ADBF has to be configured using the ADBF GUI and one signal selected using the select GUI. Thereafter the RAP unit can be configured and finally one has to steer the tile to a specific location before the measurement can be started.
All these steps are necessary to do the right measurements, but all steps needed to be done on different computers. The ADBF, selection board and RAP unit are controlled from separate computers running with Windows, while the LO synthesizers and the tiles themselves are controlled from a Linux environment. Especially the fact that for each single measurement point the RAP had to be stopped and restarted was very inconvenient, because it consumed some time, it had to be done by hand and the system tended to become unstable after a while. For these reasons it was desirable to arrange things such that the correlator needed to

be started only once for every measurement cycle while the frontend was tracking a source or scanning the sky in an automated way.

The main problem here was that the data acquisition and processing and the control of the frontend are managed by two completely independent processes. This problem was solved by implementing the data acquisition and processing in a loop which pauses shortly between runs to allow the process doing the overall control to check that the measurement has finished and to prepare the frontend for the next measurement. The overall control was implemented as a Matlab script, the data acquisition and processing loop as an addition to the original C code. The Matlab script can run on a computer using either Windows or Linux.

In the Matlab environment a THEA tile object was available which offered a high level interface to the FEC client and thereby to the FEC. However this Matlab object did not support the multifrequency capabilities of the THEA system. Support for multiple frequencies was implemented by replacing the single frequency stored in the THEA tile object by one frequency for each beam. This also required adapting the existing methods to the new data segment, which meant going through all available methods. In the same run obsolete debugging and testing code was removed from the THEA tile object.

The THEA tile object also didn't offer the flexibility needed to allow a number of THEA tiles to work together as an array. The latter wasn't a shortcoming of the THEA tile object itself, because the ADBF was meant to deal with this problem. It turned out however that it was harder to implement this feature using the ADBF then by using the frontend. The phase settings needed to steer the beam were calculated with respect to a zero phase setting right in the center of the tile. The aforementioned array functionality was implemented by changing this phase origin from constant to free parameter. An array of THEA tiles can now be constructed by setting this free parameter to the same absolute position in space.

An important spin-off from this work was that the system could be controlled independent from the RAP GUI by using Matlab scripts and functions. This already offered far more flexibility, for example to experiment with different tracking algorithms, but it still uses the same hardware, which has its own limitations. These limitations will be discussed in following chapters.

Chapter 4

The microprocessor correlator

4.1 Introduction

The THEA system as described in the previous chapter is inconvenient at some points. The user needs to have a thorough knowledge of the setup to appreciate all the system parameters and is restricted to the beam combinations which are offered by the system. Furthermore the system opposes further development due to considerations of effectiveness of further investment of man power and money, while there are still a number of features not implemented that would offer some new and exciting possibilities.

Based on the experience with THEA and the developments in computer technology, a decision was made for the development of a PCI-card interface to the HSL in the framework of the LOFAR project leaving all the processing to a powerful (cluster of) computer(s). Each of these PCI-cards would have two HSL receivers on it and was therefore called Twin Input Module (TIM). It was decided to adopt these new cards as they were developed for the LOFAR project for the THEA system. It was expected that this would be advantageous from different points of view:

1. The hardware complexity of the THEA system would be greatly reduced, since the TIM's link the HSL's directly to the PCI-bus of a PC, while the cards in the current backend send their data to the next card by a direct data connection while being controlled from different computers.
2. Because all the signal processing would be done on one or more CPU's, the software managing the data flow could simply be written in a common programming language like C. This programming environment offers far more flexibility than the firmware environment of DSP's and FPGA's allowing to implement a full correlator with only a small investment in man power.
3. The enhanced flexibility offered on the technical level may ultimately result in enhanced flexibility for the user in picking the combination of input beams to be studied as well as in setting up very specific experiments. The latter can only be achieved for a system which has a clear software interface at C level allowing new developers to implement the software required for those very specific experiments.
4. Since the LOFAR system is planned to run on a Linux cluster, device drivers and software interfaces for the TIM's would be developed specifically for Linux. This would allow to run the complete THEA system in a Linux environment allowing to use all I/O flexibility offered by this operating system.

5. This would also migrate the data acquisition process from the Windows platform to the Linux platform thus allowing to build a system running completely under Linux. This would solve the communication problems between the Linux and Windows part of the THEA cluster.

6. As the LOFAR project develops, new and better versions of these interface cards will become available. Therefore adoption of the new interface card would allow maintenance of THEA at minimum cost. This is an important point, since THEA is meant to be used as an educational tool for the next three to five years. Furthermore an up-to-date THEA system may also be used for simple experiments aiding the LOFAR and SKA efforts.

In this chapter a description will be given of the microprocessor correlator (MPC) as the new system was soon called. Thereafter the performance of the MPC will be compared with the performance of the original THEA system.

4.2 Description of the MPC

4.2.1 Overview of the new THEA system

An overview of the THEA system when the MPC is used is shown in figure 4.1. It is apparent that the MPC greatly simplifies the setup compared with the old situation which was presented in figure 3.1. It is even possible to reduce the system to a single computer by combining the MPC and the tasks currently done by “huisjewest” or “PCSKA1” on a single computer.

A more detailed scheme is presented in 4.2. The data is coming from the HSL at a rate of 40MHz. The 24bit output signal from the HSL is mapped hardwired such that it can be read from the TIM as a 32bit word representing two 16bit fixed point numbers, which four most significant bits are set to zero. On the TIM board a memory buffer of 2GB is available to collect the data.

Data can either be collected from both HSL’s or one single HSL. In the latter case the memory buffer can hold up to 13.4s of data, which reduces to 6.7s when data is collected from both HSL’s. It is not possible to read data from the TIM and write data to it at the same time. One therefore has to wait until all data has been taken or the memory buffer is filled before one can start reading the data.

During the data acquisition the data is transfered over the PCI-bus, converted from 12bit to 16bit by sign extension and finally converted to a floating point number in preparation for the FFT stage. When data is collected from more than one beam, the data is sorted in blocks consisting of a number of samples equal to the FFT size per beam such that the traces from all beams are synchronized. Therefore this stage is represented as “DAQ & sync”.

Once the data is transformed to the right format and synchronized, the rest of the data processing is straightforward. First each block of data S_{ni} is complex Fourier transformed to produce the corresponding spectra s_{ni}, where the subscript n denotes the channel number and the subscript i denotes the block number. Then the correlation matrix

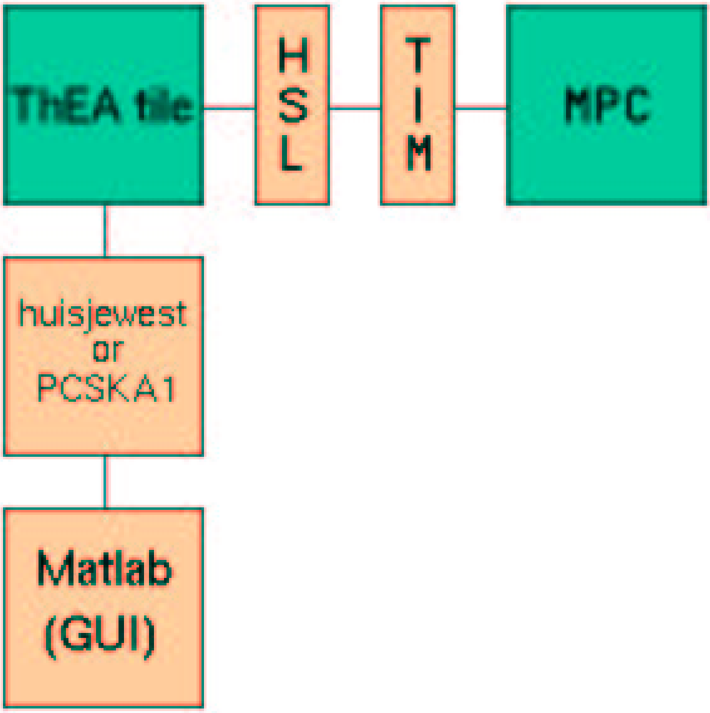


Figure 4.1: Overview of the THEA system when the MPC is used, see figure 3.1 for comparison with the old backend.

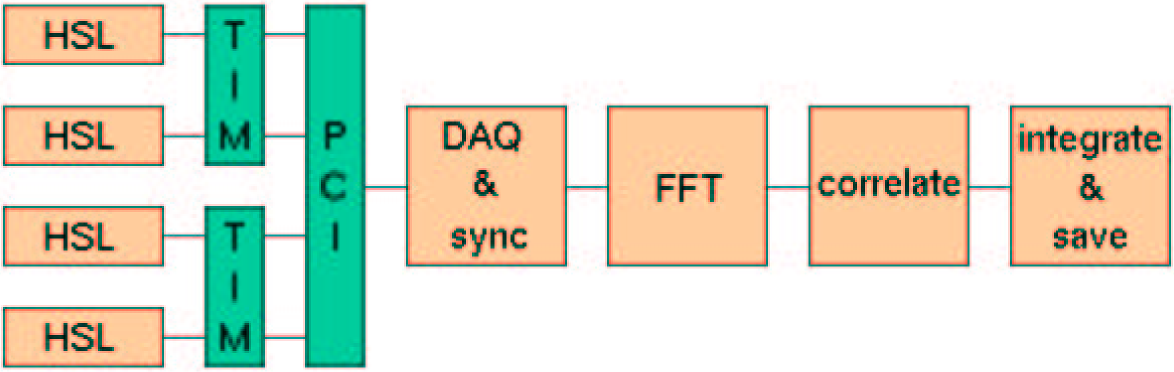


Figure 4.2: Detailed overview of the processing scheme of the MPC.

$$\mathrm{R}_i = \begin{bmatrix} \mathrm{s}_{1i}\overline{\mathrm{s}_{1i}} & \cdots & \mathrm{Re}(\mathrm{s}_{1i}\overline{\mathrm{s}_{ni}}) & \cdots & \mathrm{Re}(\mathrm{s}_{1i}\overline{\mathrm{s}_{Ni}}) \\ \vdots & \ddots & \vdots & & \vdots \\ \mathrm{Im}(\mathrm{s}_{1i}\overline{\mathrm{s}_{ni}}) & \cdots & \mathrm{s}_{ni}\overline{\mathrm{s}_{ni}} & \cdots & \mathrm{Re}(\mathrm{s}_{ni}\overline{\mathrm{s}_{Ni}}) \\ \vdots & & \vdots & \ddots & \vdots \\ \mathrm{Im}(\mathrm{s}_{1i}\overline{\mathrm{s}_{Ni}}) & \cdots & \mathrm{Im}(\mathrm{s}_{ni}\overline{\mathrm{s}_{Ni}}) & \cdots & \mathrm{s}_{Ni}\overline{\mathrm{s}_{Ni}} \end{bmatrix} \tag{4.1}$$

is calculated where $\mathrm{s}_{n_1 i}\overline{\mathrm{s}_{n_2 i}}$ should be read as the pointwise multiplication of the spectrum of block i from channel n_1 with the complex conjugate of the corresponding spectrum of channel n_2. Finally the individual correlation matrices are averaged to obtain the final integrated result, i.e.

$$\mathrm{R}_{integated} = \frac{1}{N_{blocks}} \sum_{i=1}^{N_{blocks}} \mathrm{R}_i \tag{4.2}$$

This average is saved to disk as final result from the correlation process. This result is thus a three dimensional matrix containing a full correlation matrix for all frequency points.

4.2.2 Ongoing activities

In order to use more than two input beams the TIM's involved need to be synchronized. This requires an additional device to ensure that synchronization pulse arrives in synchronization with the 40MHz clock. Experiments have shown that if a random synchronization pulse is used there is a chance of about 50% that both TIM's react to this pulse on two different clock pulses, thus introducing a shift of one sample. Both the synchronization device and the software are ready for use, but the combination is not tested yet.

It has been mentioned earlier that the data transfer rate from the TIM to the computer is limited by the fact that the TIM device driver doesn't support data transfer under DMA control. It is expected that this feature will be available shortly. This is expected to increase the data transfer rate to about 60MB/s, while reducing the load due to data transfer of the CPU, thereby leaving more CPU cycles for the data processing.

Even under DMA control, the data transfer rate is not enough to transfer the data at the same rate as the data comes in, which would require a bandwidth of 320MB/s when both HSL's are used at 40MHz. This would restrict the duty cycle to about 18%. This will become less restrictive if less data needs to be transferred. It is possible to replace the 20MHz filter in the IDC by a 10MHz filter. If a 10MHz bandwidth is used, the Nyquist criterion is already satisfied when 10Msample/s are collected since complex samples are used. This can be realized by a factor four downsampling at the input side of the TIM, thus reducing the input data rate from 320MB/s to 80MB/s, which would allow a duty cycle of 43% assuming a data transfer rate of 60MB/s under DMA control. At this stage downsampling at the input of the TIM can be set to 40MHz (no downsampling), 20MHz (factor 2 downsampling) and 10MHz. The latter option can not be used yet, because the 10MHz filter is not yet implemented on the IDC.

Although a higher data rate over the PCI-bus offers the possibility of an increased duty cycle, the duty cycle will in practice only be increased if the CPU's are able to process the data at the same rate as the data acquisition process. That is, if data is taken during 1s and it takes 2.5s to read this data from the TIM, then the CPU should be able to process the data in 3.5s if the processing is not to slow down the measurement process. Thus at this moment the data transfer rate from the TIM's is the limiting factor, but as soon as the DMA control is successfully implemented, the processing power of the CPU's will be the limiting factor if not used efficiently.

4.2.3 Future improvements

One of the main reasons for choosing the TIM's as alternative to the old backend was the prospect of future developments on the TIM's in the framework of the LOFAR project. A

number of improvements will be mentioned here.
First of all the TIM's may be adapted to 64bit 66MHz PCI-slots, allowing even larger transfer rates. As will be clear at this stage this is only a useful option when this data rate can also be handled by the CPU's. Fortunately this problem also needs to be solved toward the LOFAR project allowing THEA to adopt this solution without much effort.
Another straightforward improvement is memory extension. As memory becomes cheaper it will be possible to upgrade the TIM's by replacing the memory modules. This will allow longer exposures to be taken at once. Finally analog inputs are considered for the LOFAR project, which would open the possibility to connect the tiles to the backend directly on RF level. This option should however be considered carefully, because this would result in larger gain losses due to signal transport and may therefore reduce the sensitivity of the system.

4.3 Comparison with the old backend

The MPC offers some interesting features which aren't available when using the old backend as demonstrated in the previous section. It also demonstrated the ease with which new features can be developed; the MPC was developed using only a fraction of the man power that was invested in the old backend. The flexibility in use and the low development costs are certainly an advantage of this new setup.
A major disadvantage of the MPC at this time is its very poor duty cycle. This is caused by the fact that the maximum read-out speed of the TIM is only 4MB/s, while it has to accept at least 80MB/s for each active HSL. The 4-channel correlator using two TIM's was structured to work in multi-threaded mode whenever possible to ensure effective use of both available processors and achieved a duty cycle of 1.9% at an input data rate of 20MHz. The old backend would require 8 measurements to be conducted to arrive at the corresponding result with the disadvantage of not having all correlations based on the same input signals. Since the old backend can process the signal in real time giving it a duty cycle of 100% for a single measurement, it has a duty cycle of 12.5% when compared to the MPC on equal grounds. This is still much better than the 1.9% of the MPC, but DMA control will soon become available. With the current implementation this is expected to increase the MPC duty cycle to about 10%, making its performance comparable to the performance of the old backend. This figure will only increase further if either the code or the hardware is optimized further. The main conclusion that can be drawn from this discussion is that the MPC will win at the end when the full correlation matrix is required. If only a single autocorrelation is needed, the old backend will have the best duty cycle.

Chapter 5

Experiments

5.1 Introduction

In this chapter results from a number of demonstration and evaluation measurements will be presented. The demonstration measurements served to show that it was possible to perform measurements which were expected to work, such as multifrequency multibeaming and array beam forming. In the evaluation measurements specific aspects of the system are examined more closely to explore the limitations of the system.

The aim of these measurements was twofold. First of all the proper operation of the system needed to be established. Feedback on this part was used to gain experience with THEA as measurement system such that appropriate software could be developed to facilitate the measurement process. The sources of the basic measurement tools are included in appendix B.

The second aim was to obtain a level of understanding on the operation of the system which needed to be sufficient to explain the measurement results. This was realized by a number of noise measurements on the IDC, outdoor calibration measurements and outdoor measurements on the behavior of the vector modulators.

The demonstration and evaluation phase was concluded by a number of measurements on astronomical sources, namely imaging on the galactic neutral hydrogen and fringe measurements on the sun. The results from imaging on galactic neutral hydrogen were compared to the results from the Leiden-Dwingeloo Survey (LDS) [11][12]. This comparison also served as an independent way to establish the noise temperature of the THEA frontend.

Note that all sky images in this paper have the sign of v flipped to obtain the image one would see when looking at the sky. This means that East is on the left hand side of the sky images.

5.2 Counting bits and bytes

Before getting to the discussion on the measurements, a few remarks will be made on the backend. As mentioned in the previous chapter, at two stages in the reduction process a number of bits is thrown away. It was pointed out that this was necessary to prevent integer overflows. This section will focus on the consequences of these bitshifts.

The first bitshift occurs directly before the Fourier transformation. At this stage it is absolutely necessary to prevent integer overflows, because an integer overflow in the radix4 algorithm will cause multiple channels to be corrupted. Therefore a trade off needs to be made between a good number of signal bits and a reasonable number of bits head space to accommodate sudden strong interference signals. Since the signal to be measured can in principle be accommodated in a single bit, one can in principle reduce the signal to a single bit thereby keeping five bits head space.

Measurement experience shows that there are normally two really strong signals excluding the measured signal, namely interference from the IDC and the DC component. The interference from the IDC can be split into two components, a white noise component which simply adds to the system temperature, and a peaked component which is a high order harmonic of the 40MHz clock signal. In principle this peak is just a CW signal, but is broadened due to time jitter occurring in the IDC hardware. This is especially troublesome when making a crosscorrelation between two tiles in which case two independent IDC's are involved which are two independent sources of time jitter. The result is a peak with a Gaussian profile which basis may be detected over a range of more than 5MHz.

Fortunately these two interference signals are detected constantly, which allows to obtain a value for the bitshift just before the FFT such that the number of signal bits is maximized while still leaving enough head room. This choice has proven to work satisfactorily and was checked by comparing the results with results obtained after applying a higher bitshift.

The second bitshift is implemented just before the integration stages. In principle save settings can be calculated by assuming that all bits at the output of the FFT stage are significant. In this case the number of integrations on the DSP determines the bitshift. In most cases this bitshift may be set to a lower value, such that the DC component causes an integer overflow but the IDC peak doesn't. In this case the integer overflow in the DC channel is not troublesome, because it only affects the DC channel which is normally not used as signal channel. Overflow in the channels of the IDC peak should however be avoided, because it might be interesting to filter it from the data by fitting a profile to it.

5.3 Correlator noise

The level at which the correlator software starts to produce numerical artefacts is an important limitation of the system, because this effectively limits the maximum usable integration time. The correlator was tested by connecting two IDC's which were not build into a tile, to two noise generators producing white noise. The signal level was adjusted such that the IDC's produced an output signal amplitude of about 1000 units superimposed on the zero offset of the AD-converters which amounted to about 40 units. On the ADBF each signal was attenuated by a factor 128, thus effectively producing a 4 bit signal (an amplitude of 8 units and a sign). This signal was fed into the RAP unit and integrated during 1, 4, 16, 60 and 240 seconds. The results from these measurements are shown in figure 5.1.

The left hand plot in the figure shows the autocorrelation results of the two IDC's for different integration times. The amount of noise in the spectrum is about halved in every step as expected since the noise goes down with the square root of the integration time. The deviations of the band pass from a white noise spectrum are due to the digital filter applied to the signal in the IDC to prevent aliasing of frequency components outside the 20MHz band.

The right hand plot of figure 5.1 shows the crosscorrelation results for different integration times. Again the amount of noise is decreasing as expected. With increasing integration time and thereby noise reduction a number of unexpected features becomes visible. This can be

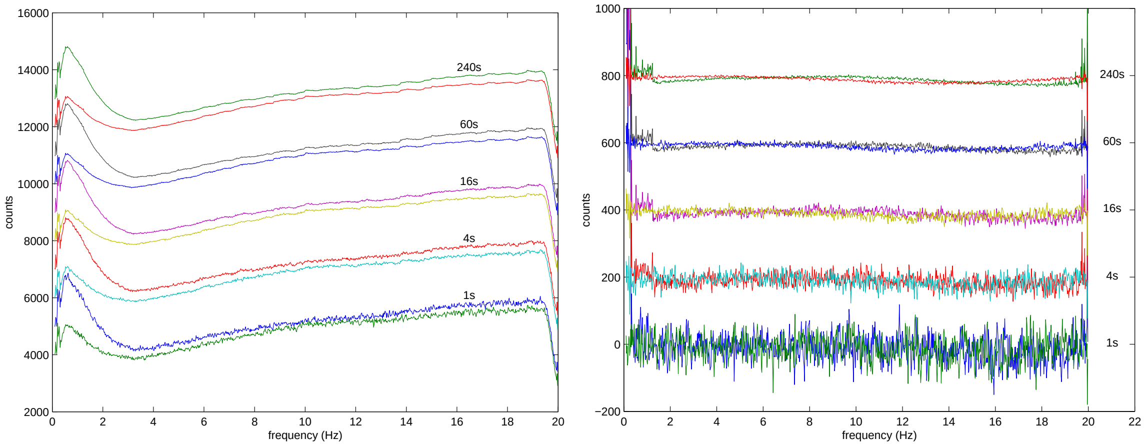


Figure 5.1: Plots showing the autocorrelation results of two IDC's connected to a noise generator (left) and the real and imaginary parts of the corresponding crosscorrelation (right) over the full bandwidth of the digital filter. For readability the autocorrelation results are shifted in steps of 2000 counts, while the crosscorrelation results are shifted in steps 200 counts. Each pair of curves is marked with the integration time of the measurement.

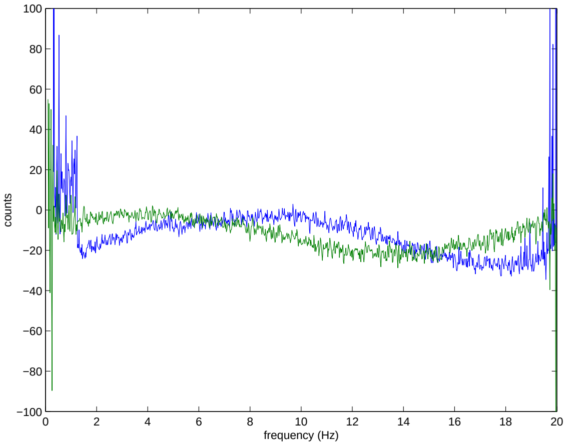


Figure 5.2: Plot showing the real and imaginary part of the crosscorrelation between two IDC's after 240s integration, i.e. this plot zooms on to the upper pair of curves from figure 5.1.

seen more clearly in figure 5.2 which zooms in on the 240s results.

From this plot it becomes apparent that the correlation shows a sinusoidal fluctuation over the band indicating a phase slope of 360 degrees over the 20MHz band. This can be explained by a time delay of two samples between the signals from the two IDC's. A closer look at this phase slope normally reveals that the slope isn't exactly a multiple of 180 degrees indicating that there are also delay differences in the cables between the circuit boards. The first delay component can be reduced completely by fine tuning the measurement system, the latter can largely be calibrated out. The plot also shows that the correlated signal amplitude of about 10 units is only 0.2% of the total power. This is sufficiently low to use the correlation system for further tile evaluation.

A problem which can not be sorted out can be seen near the DC component of every crosscorrelation curve except after 1s integration time. All these plots show that the first 1MHz of the 20MHz band is useless due to a very noisy signal which could not be explained. It turned out to pollute more and more frequency channels when either the input signal level was increased or the integration time was enlarged suggesting that it is due to some kind of saturation effect. Although the exact origin of this effect is not clear, it can be concluded that it doesn't effect the central frequency channels for integration times up to a few minutes, i.e. the effect is smaller than 0.1% of the system noise.

Another interesting point that was noted during the autocorrelation measurements was that the peak due to the IDC clock which is normally present in the spectra was now absent. From this observation it can be concluded that this signal is picked up on RF level. Further support for this assumption was gained from the fact that the center frequency of the IDC peak behaved as harmonic of the 40MHz IDC clock.

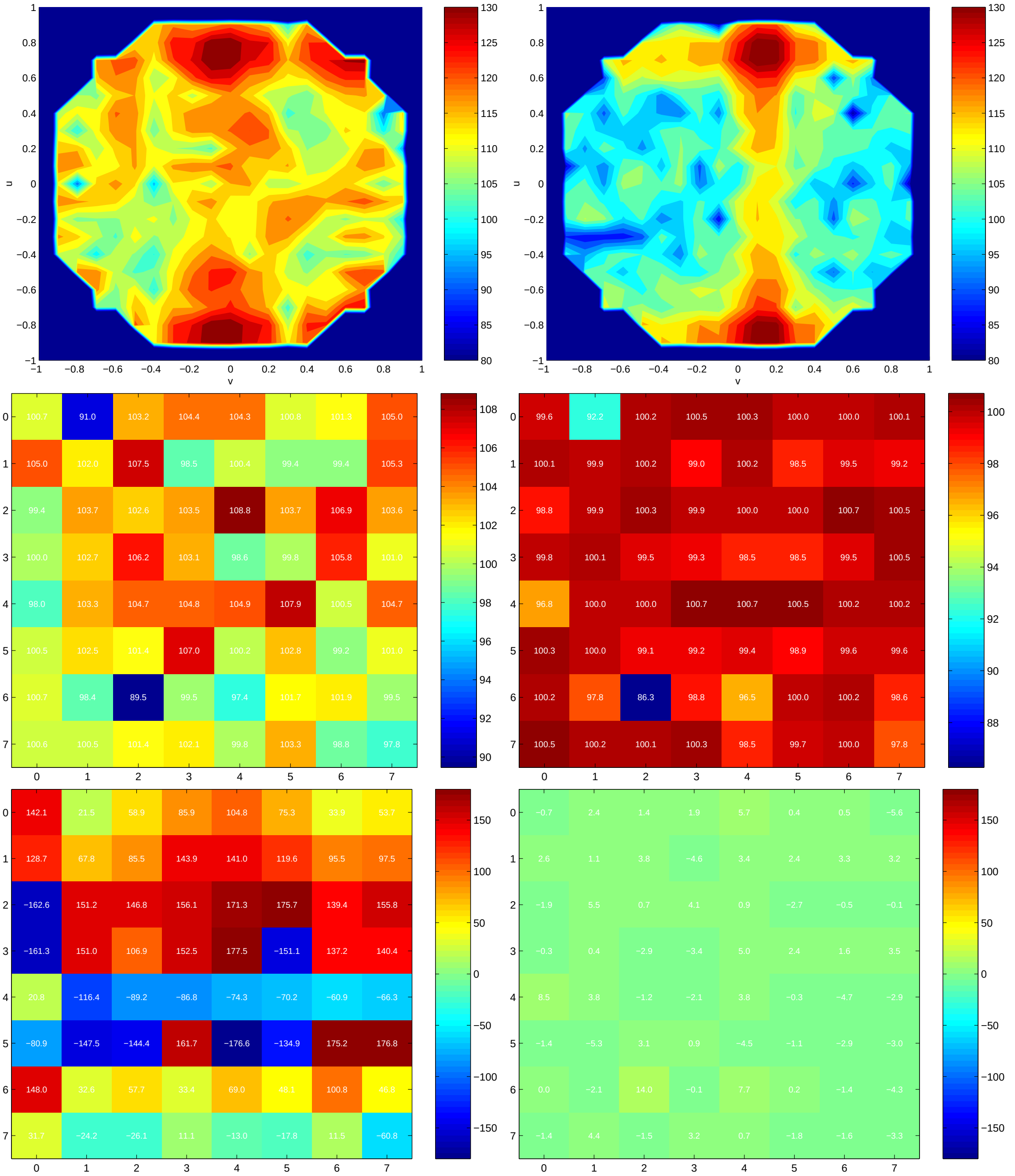
5.4 Tile calibration

The usability of a phased array antenna strongly depends on the beam forming capabilities of the system. In the beam forming process the desired gain settings of the individual antenna elements with respect to one another are calculated. In order to actually set the complex gains of the antenna elements one needs to know the phase and amplitude response of the individual elements for at least one specific setting. The key point in the tile calibration is that there is no such known setting because there are intrinsic variations between the individual elements and the electronic path lengths to them. The purpose of the tile calibration is thus to determine the intrinsic complex gains of the antenna elements.

These gains can be determined by applying equation (2.28) to data from a scan consisting of at least a number of points equal to the number of antenna elements. In this case the gain factors needed to steer the beam $\mathrm{C}(\mathrm{U}_0, \mathrm{X}; \vec{u}_s)$ can be calculated. If the number of points is larger than the number of elements to be calculated the inverse of $\mathrm{C}(\mathrm{U}_0, \mathrm{X}; \vec{u}_s)$ should be replaced by its pseudo-inverse. This effectively produces a least squares fit of the antenna gains to the signal distribution model $\mathrm{S}(\mathrm{U}_0; \vec{u}_s)$. If the calibration measurements are done using a single point source, the signal is simply a delta distribution with its peak at the position of the source.

The application of the theory outlined in the previous paragraph either requires a source to be moved with respect to the THEA tile or the beam of a THEA tile to be sweeped through the field of the source. The first technique is commonly used to calibrate antenna systems. Various forms of the second method were used for THEA [13][14][15]. During these measurements the output signal of the beam former was observed relative to the reference signal with a vector voltmeter or spectrum analyzer in the antenna room at ASTRON. The results which will be presented here were obtained by a third method using Afristar as celestial source.

This holographic method was implemented in a Matlab script, which would take one beam as reference beam fixed on the calibration source while sweeping a second beam over the sky. The phase and amplitude information of the second beam was extracted by crosscorrelating the signal with the reference beam. After the accumulated data was read, a correction was calculated. It turned out that the optimal result was obtained when this procedure was repeated once or twice more, using the correction calculated in the previous iteration as



	0	1	2	3	4	5	6	7
0	100.7	91.0	103.2	104.4	104.3	100.8	101.3	105.0
1	105.0	102.0	107.5	98.5	100.4	99.4	99.4	105.3
2	99.4	103.7	102.6	103.5	108.8	103.7	106.9	103.6
3	100.0	102.7	106.2	103.1	98.6	99.8	105.8	101.0
4	98.0	103.3	104.7	104.8	104.9	107.9	100.5	104.7
5	100.5	102.5	101.4	107.0	100.2	102.8	99.2	101.0
6	100.7	98.4	89.5	99.5	97.4	101.7	101.9	99.5
7	100.6	100.5	101.4	102.1	99.8	103.3	98.8	97.8

	0	1	2	3	4	5	6	7
0	99.6	92.2	100.2	100.5	100.3	100.0	100.0	100.1
1	100.1	99.9	100.2	99.0	100.2	98.5	99.5	99.2
2	98.8	99.9	100.3	99.9	100.0	100.0	100.7	100.5
3	99.8	100.1	99.5	99.3	98.5	98.5	99.5	100.5
4	96.8	100.0	100.0	100.7	100.7	100.5	100.2	100.2
5	100.3	100.0	99.1	99.2	99.4	98.9	99.6	99.6
6	100.2	97.8	86.3	98.8	96.5	100.0	100.2	98.6
7	100.5	100.2	100.1	100.3	98.5	99.7	100.0	97.8

	0	1	2	3	4	5	6	7
0	142.1	21.5	58.9	85.9	104.8	75.3	33.9	53.7
1	128.7	67.8	85.5	143.9	141.0	119.6	95.5	97.5
2	−162.6	151.2	146.8	156.1	171.3	175.7	139.4	155.8
3	−161.3	151.0	106.9	152.5	177.5	−151.1	137.2	140.4
4	20.8	−116.4	−89.2	−86.8	−74.3	−70.2	−60.9	−66.3
5	−80.9	−147.5	−144.4	161.7	−176.6	−134.9	175.2	176.8
6	148.0	32.6	57.7	33.4	69.0	48.1	100.8	46.8
7	31.7	−24.2	−26.1	11.1	−13.0	−17.8	11.5	−60.8

	0	1	2	3	4	5	6	7
0	−0.7	2.4	1.4	1.9	5.7	0.4	0.5	−5.6
1	2.6	1.1	3.8	−4.6	3.4	2.4	3.3	3.2
2	−1.9	5.5	0.7	4.1	0.9	−2.7	−0.5	−0.1
3	−0.3	0.4	−2.9	−3.4	5.0	2.4	1.6	3.5
4	8.5	3.8	−1.2	−2.1	3.8	−0.3	−4.7	−2.9
5	−1.4	−5.3	3.1	0.9	−4.5	−1.1	−2.9	−3.0
6	0.0	−2.1	14.0	−0.1	7.7	0.2	−1.4	−4.3
7	−1.4	4.4	−1.5	3.2	0.7	−1.8	−1.6	−3.3

Figure 5.3: Plots showing results from calibration measurements for a single beam. The images on the left hand side show the measurement results before calibration, the top image presenting the measured intensity on a logarithmic scale, the center image the gains of the individual antenna elements on a logarithmic scale and the bottom image the phase of the individual antenna elements when the tile is pointed to the zenith in degrees. The right hand images present the corresponding results after calibration.

starting point for the next. The results obtained after three iterations are shown in figure 5.3.

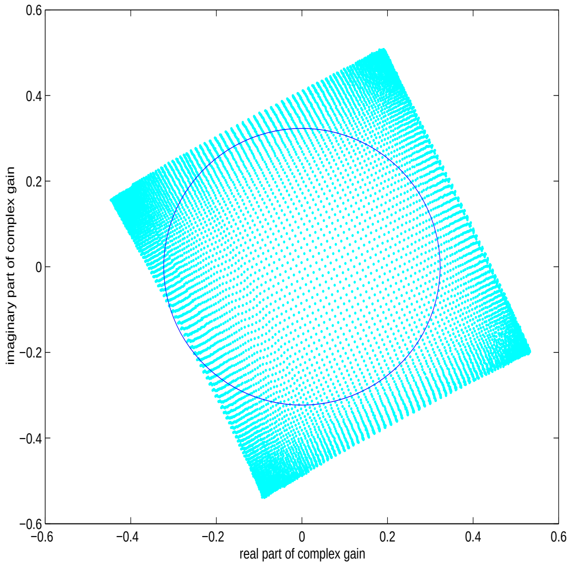


Figure 5.4: Plot showing the available gain settings of the vector modulators at 1480MHz. Also shown is the amplitude range in which all phases are covered by the vector modulator settings.

The upper left hand plot shows the amplitude of the crosscorrelation expressed as $20\log_{10}$(counts) before calibration, while the upper right plot shows the amplitude of the crosscorrelation after calibration on the same scale. This clearly shows that the position of the source is changed to the position where it should be according to the model and that a larger part of the sensitivity of the beam is concentrated in the main beam. These observations are reflected in the phase and amplitude pattern of the antenna elements of the tile which are shown in the other four plots.

The change in position is mainly due to the flattened phase profile. The structure in the phase profile before calibration can for a large part be explained by path length differences on the circuit boards. The change in the shape of the beam pattern is due to the fact that the intrinsic gains are corrected to equal amplitude and equal phase in the zenith direction.

The phase and amplitude errors after calibration are mainly caused by the vector modulators. These devices can only be set to certain discrete values. These complex values are plotted in figure 5.4. During the initialization the largest circle fitting within this range of values is determined and defined as unity circle. This circle is also plotted in the figure. During the sky scan of the first iteration of the calibration only the phase of the elements is modified. The amplitude is fixed on the unity circle. It will be clear from the plot that it is highly undesirable to attenuate all elements to the signal level of the weakest element, because there are only a few points near the origin of the complex plane, which would cause large phase and amplitude errors. Therefore the maximum attenuation during an iteration is limited to a user specified fraction (in this case two-third) of the signal level of the strongest element. After three iterations this could lead to a total attenuation of 1.5^3 or 3.375, but in practice this will only be slightly larger than 2.5 due to measurement errors causing some elements to be less attenuated than projected.

The standard deviation in amplitude and phase after calibration are 0.9dB and 3.2 degrees respectively leaving out the two elements, which are obviously not corresponding correctly. The phase error agrees quite well with the maximum attainable accuracy of the vector modulators. The corresponding amplitude error was expected to be 0.5dB. The difference between the amplitude error as expected from the phase errors and the true amplitude errors are due to systematic deviations produced by the vector modulators. These deviations will be addressed in the next section.

5.5 Vector modulator responses

In the previous section it was demonstrated that the maximum attainable accuracy in the calibration procedure is limited by the discrete settings of the vector modulators. The discreteness of the vector modulator settings is an intrinsic property of the vector modulators so further accuracy can only be achieved by replacing the vector modulators by more accurate electronics. However accuracy is one thing, correctness is something different. The look-up table used to control the vector modulator was determined by measurements on a single vector modulator outside the electronics, i.e. in an environment which electronically differs from an operational THEA tile.

Therefore a measurement was set up to validate the vector modulator settings. In this measurement one of the vector modulators was commanded to phase toggle over the unity circle described in the previous subsection, while the phase and amplitude of the output were measured. The phase agreed to the commanded position within the deviations one would expect from the vector modulator table, while the amplitude varied systematically. The unity circle was changed into an ellipse. Since this measurement was done on a single vector modulator, it could still be a problem with that specific vector modulator. Therefore a new measurement scheme was proposed in which the behavior of all vector modulators of a single beam could be studied at once.

In this scheme the tile was first calibrated using the iterative procedure presented in the previous section. Then a series of scans were made on Afristar where all vector modulators were given an additional phase offset of 30 degrees. Doing this the relative phases of the elements with respect to each other should remain equal, but its absolute value should change 30 degrees compared to the previous scan. One series consisted of 13 scans starting with a zero degrees offset and ending with a 360 degrees offset. This procedure was repeated 10 times to reduce the measurement errors and the effects of drift due to environmental changes, mainly temperature.

The results from these measurements are shown in figure 5.5. Each plot shows the amplitude for eight elements against the phase offset. Most elements show two types of variation. First there is a sinusoidal variation with one period over 360 degrees. This can be explained by a zero offset of the center of the iso-amplitude circle or ellipse from the origin of the complex plane. In this case the deviations from the desired iso-amplitude circle are positive on one side and negative on the other. This is illustrated in figure 5.6.

Also shown in this figure is the other type of deviation, ellipticity. In this case the amplitude is first smaller than expected, than becomes larger than expected, then gets smaller again and finally becomes larger a second time. This behavior is reflected by the sinusoidal amplitude variations having two periods over the full 360 degrees range. The average amplitude of these variations is however quite small. A Fourier analysis of the data produced an average value of 0.1dB for the first component and 0.2dB for the second. Although these deviations are

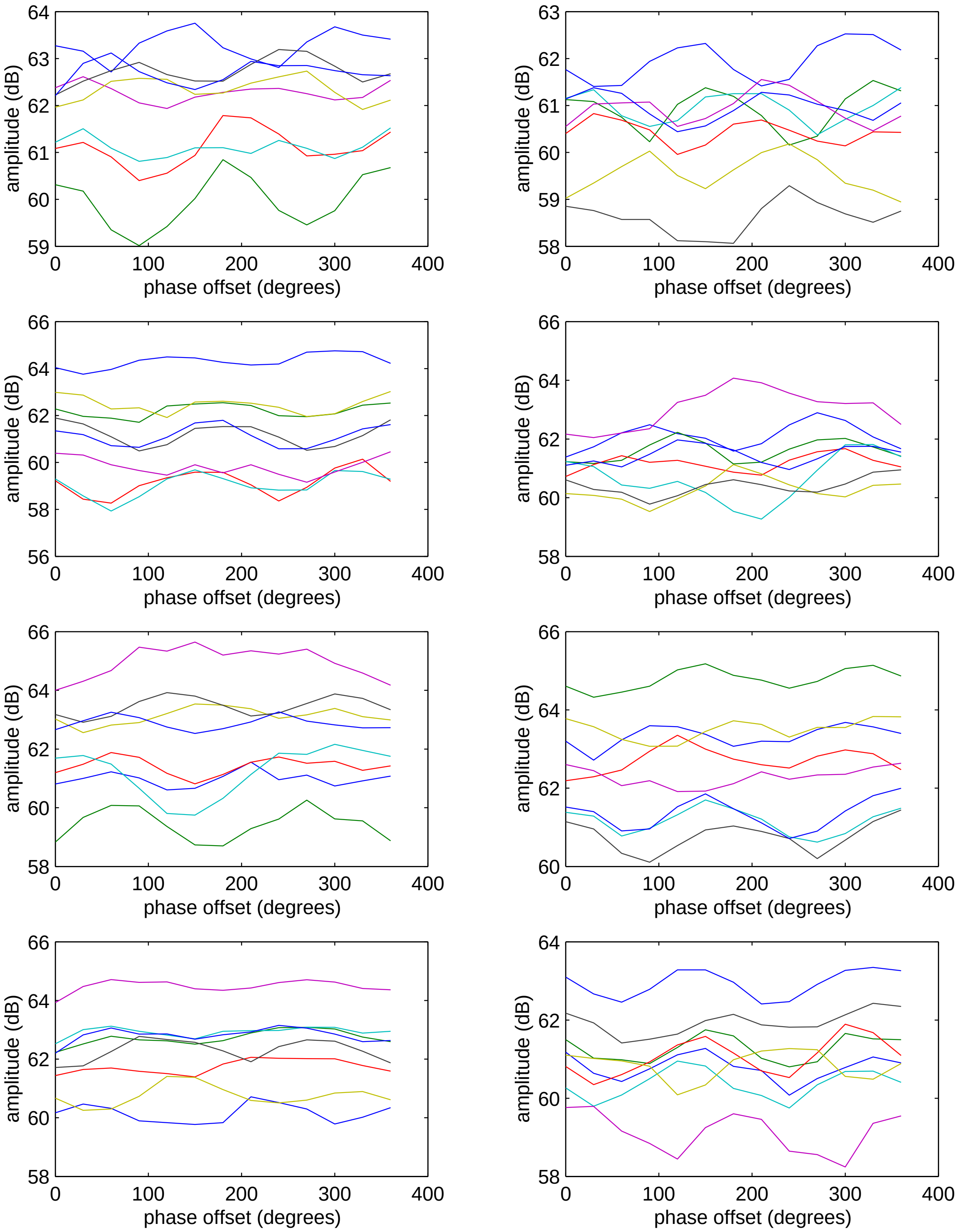


Figure 5.5: Plots showing the amplitude variation of the elements for specific phase offsets. Each plot shows these fluctuations for the eight antenna elements in a single row.

clearly detectable as demonstrated by this measurement, they are smaller than the accuracy of the vector modulators discussed earlier. However, since this is a systematic effect, it adds

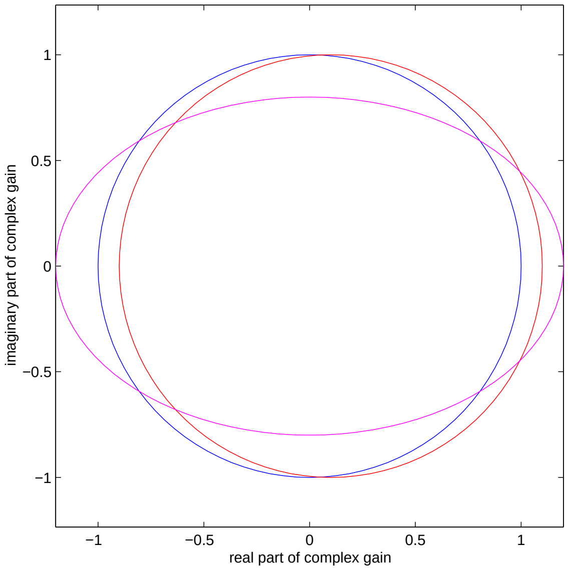


Figure 5.6: Plot showing possible deviations from the expected unity circle (blue): zero offset (red) and ellipticity (magenta).

directly to the amplitude errors found in the calibration. This explains the difference between expected and measured amplitude errors pointed out in the previous section.
On the other hand the extent of this effect for the individual elements is too small to really affect the beam forming capabilities of an individual THEA tile. In this context the reader is reminded to the fact that a THEA tile produces a main beam that extends over 10 degrees, even at the highest frequencies for which the system is specified.

5.6 Measurement of the sky background continuum

Continuum sources are harder to detect than sources with a line spectrum due to the fact that the signal power of the continuum source is merely added to the noise floor. A common technique to distinguish a continuum source from the noise floor is beam switching. This technique assumes that if the telescope is aimed at another, empty, point on the sky, the signal from the source will no longer be detected but the noise will, i.e it assumes that the measured noise is constant over the sky.
For a standard single dish telescope this assumption normally holds, but if THEA is beam switched, the electronics of the tiles are set to different operating points which may cause changes in the noise floor. Furthermore it is good to have an idea about the continuum background caused by the environment of the telescope. Therefore a series of sky images with a single tile was made probing a dark sky, i.e. during a moonless night, at frequencies between 1000MHz and 1500MHz in steps of 20MHz. The results for 1000, 1100, 1200, 1300, 1400 and 1500MHz are shown in figure 5.7.
The sky images show that in general the noise floor rises towards the edges of the field. This is caused by pickup from the ground and from nearby trees which form a continuum background of about 300K at lower elevations. This point is illustrated by the simulation results shown in figure 5.8.

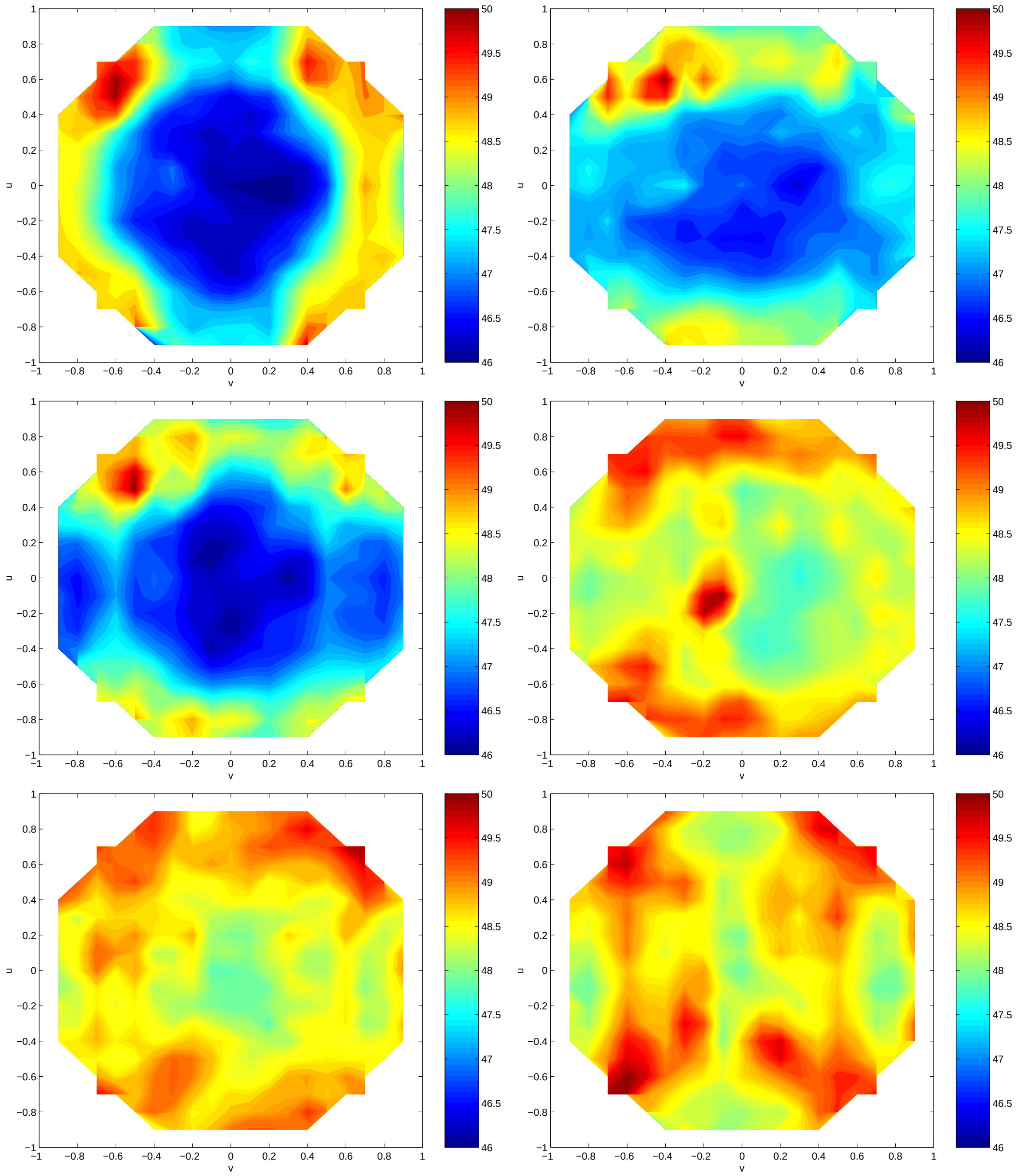


Figure 5.7: Plots showing the fluctuations in the band of the autocorrelation spectrum over a dark sky. All plots are normalized such that the maximum in the plot corresponds to 100000 correlator units or 50dB. From left the right and from top to bottom the plots show the results at 1000, 1100, 1200, 1300, 1400 and 1500MHz. The plotted intensity is calculated as the median over the full 20MHz bands centered at the aforementioned frequencies.

In these simulations a sky model was created having only a continuum background of 300K in every direction up to 20 degrees elevation. This model was weighted with a cosine distribution

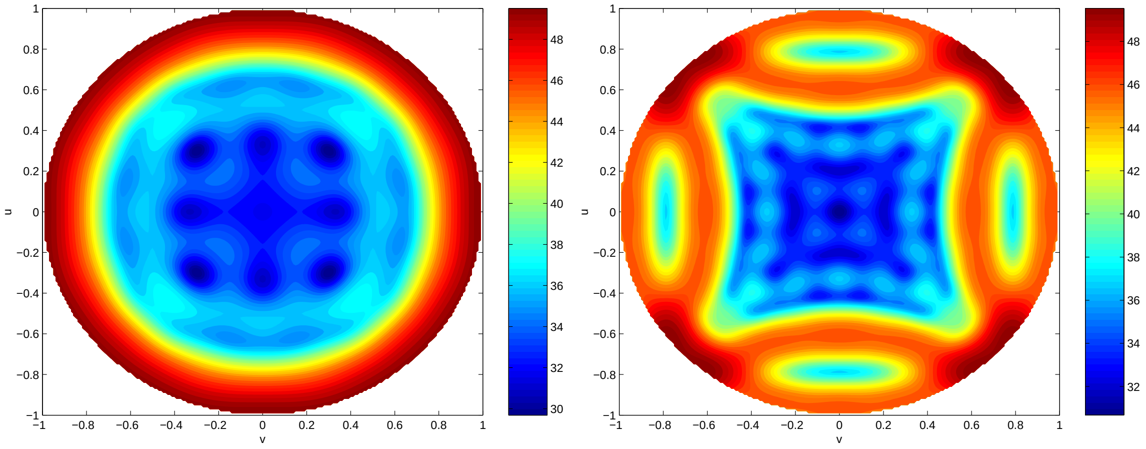


Figure 5.8: Plots showing the predicted fluctuations in the noise floor due to nearby trees rising up to 20 degrees elevation around the tiles at 1000MHz (left) and 1500MHz(right).

to simulate the sensitivity pattern of the individual elements, which become less sensitive toward lower elevations. The resulting model was convolved with the beam pattern calculated based on equation (2.27) and scaled to have its maximum at 50dB for comparison with the measurement results.

Comparison of the plots in figure 5.8 with the corresponding plots in figure 5.7 shows that this model predicts the overall structure quite well. The variations predicted by the model are larger than the measured variations due to the fact that the model assumes an ideal THEA tile without phase and amplitude errors in the gain settings of the individual elements. The presence of phase and amplitude errors decreases the depth of the nulls in the beam pattern and makes the side lobes more diffuse, thus causing lower RFI reduction levels. The measurement as well as the simulation results show that the pick-up from the environment at low elevations starts to leak toward the zenith as soon as grating lobes are present, i.e. as soon as the wavelength becomes shorter than two element spacings.

Basically all measurement detect some additional signal to the North-East. This is probably either the Leiden-Dwingeloo Telescope or the ASTRON building. The point-like source detected closer to the zenith are not accounted for at the moment and do not always reproduce when the THEA system is restarted.

5.7 Array beam Forming

The array beam forming feature was tested by placing the 4 tiles in a 2-by-2 configuration directly next to each other aligned with the quarters of the compass. Calibration at array level was done by pointing the array at Afristar taking the geometrical delay into account by measuring the signal of the individual tiles in autocorrelation and in crosscorrelation between three of the tiles and the fourth. This information was used to find weights for the signals of the individual tiles such that these signals would have the same amplitude and phase.

A regular grid was defined on the (u, v)-plane with a spacing of 0.05 (2.9 degrees near the zenith). With this setup three consecutive scans were made. The results are shown in figure 5.9. For the first scan the weights as found in the aforementioned array calibration

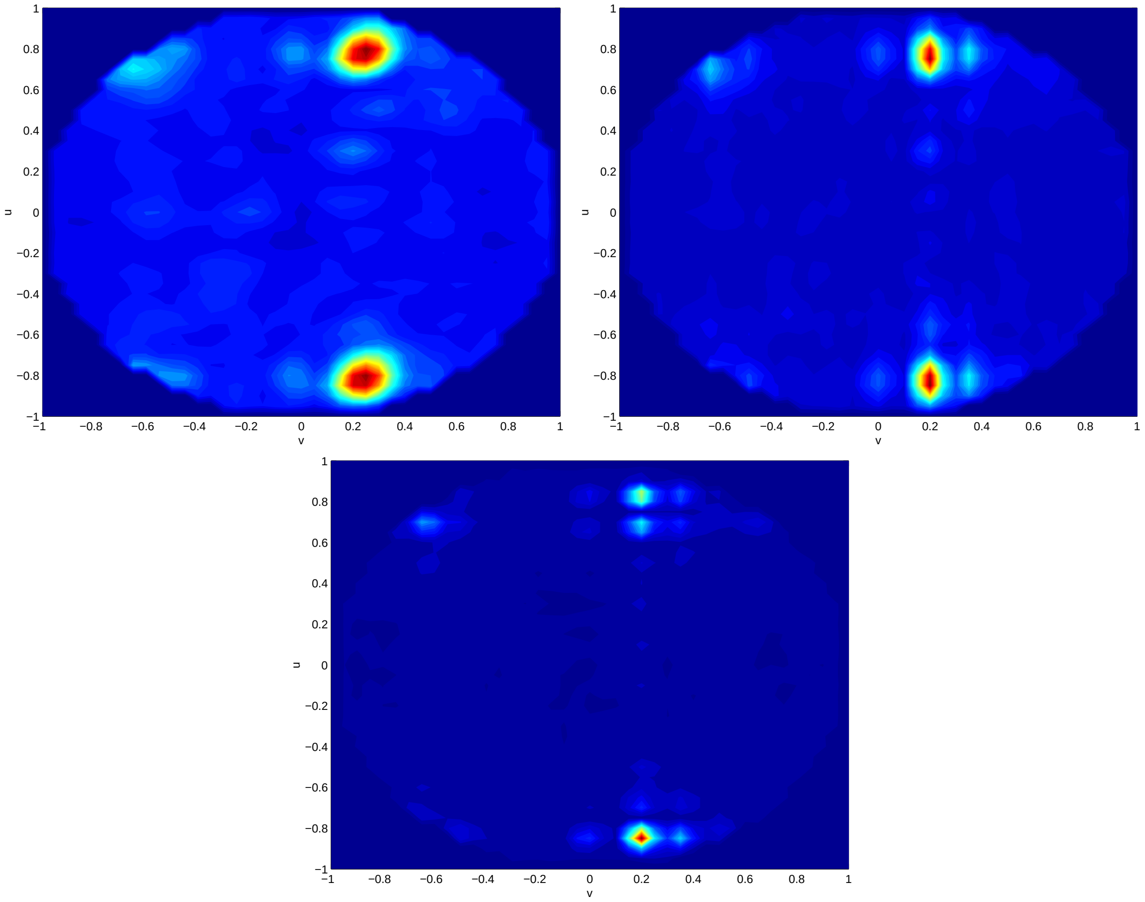


Figure 5.9: Contour plots showing the intensity pattern measured in autocorrelation of an array of 1 tile (top left), 2 tiles on an East-West baseline (top right) and a dense square array of 4 tiles (bottom) on Afristar. All scans were made on the same (u, v)-grid and all results are plotted without baseline subtraction.

scheme were used, resulting in a beam with a half-power beam width (HPBW) of about 6 degrees. During the second scans the weights for tiles 1 and 4 were set to zero, thus effectively measuring with tiles 2 and 3 only. These tiles were placed on an East-West baseline, thus producing an elliptical beam having a HPBW of about 6 degrees along the East-West line and about 12 degrees along the North-South line. Finally the weight of tile 2 was also set to zero, thus effectively measuring with a single tile. This produced a round beam with a HPBW of 12 degrees.

The noise reduction capabilities of array beam forming are also clearly visible in these plots. Since the noise of the individual tiles is in principle independent, the noise adds incoherently while the signal adds coherently, thus improving the S/N ratio by a factor two when four tiles are used.

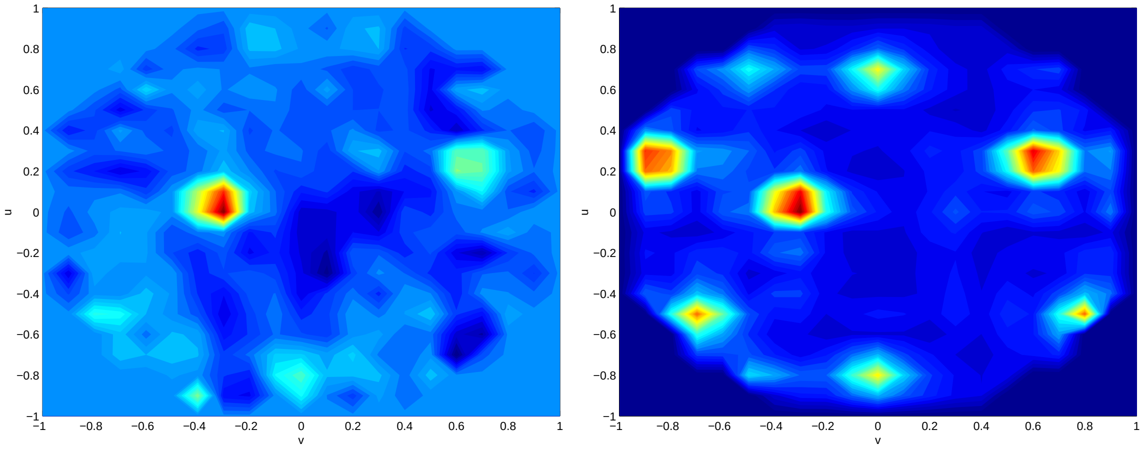


Figure 5.10: Intensity contour plots showing the distribution of GPS satellites over the sky in Dwingeloo on October 16, 2002 at 4:00 AM measured at frequencies of 1227.6MHz (left) and 1575.42MHz (right) respectively. The distinction between the true satellites and the grating lobes can be made by comparing the two images.

5.8 Multifrequency multibeaming

One of the features added to the THEA system was the ability to measure with two beams of the same tile at different frequencies. The GPS satellites were found to be ideal sources to test this feature. This was done by producing a series of sky scans over night on a rectangular (u, v)-grid with a step size of 0.1. This produced one full sky scan every 6 minutes with 1 second integration on each individual point. One of these images is shown in figure 5.10.

At the time of the measurement the GPS satellites transmitted signals at 1575.42MHz and at 1227.6MHz. The first frequency was used in civilian as well as military applications and was not encoded. The latter was used in military applications only and was encoded by redistributing the power over a band of 4MHz around the 1227.6MHz center frequency such that at each individual frequency the signal was to weak to be useful unless you had the key to concentrate all signal power in this band. This also explains why the 1227.6MHz image looks more noisy than the 1575.42MHz image.

This experiment also illustrates that measurements at two distinct frequencies allow to distinguish between the real sources and the grating lobes, because the separation between a source and its grating lobes changes as illustrated in figure 5.11. These images show that the source itself is found at the same spot, but its grating lobes aren't, as expected from equation (2.24), which shows that the position of the grating lobes depends on wavelength. A closer look at the images in figure 5.10 shows that the spot at $(u, v) = (0.25, -0.85)$ is a grating lobe of the satellite at $(u, v) = (0.25, 0.60)$, The other grating lobes visible in the 1575.42MHz image are at $(u, v) = (-0.50, 0.80)$, $(u, v) = (0.70, 0.00)$ and $(u, v) = (0.70, -0.50)$.

5.9 Reduction of LDS data

The Leiden-Dwingeloo Survey has provided an all sky map of galactic neutral hydrogen visible in Dwingeloo [11] [12]. The Leiden-Dwingeloo telescope is a 25m single dish telescope having a HPBW at 1420MHz of 0.6°. The data is collected on a 0.5° grid or at 60% of the spatial

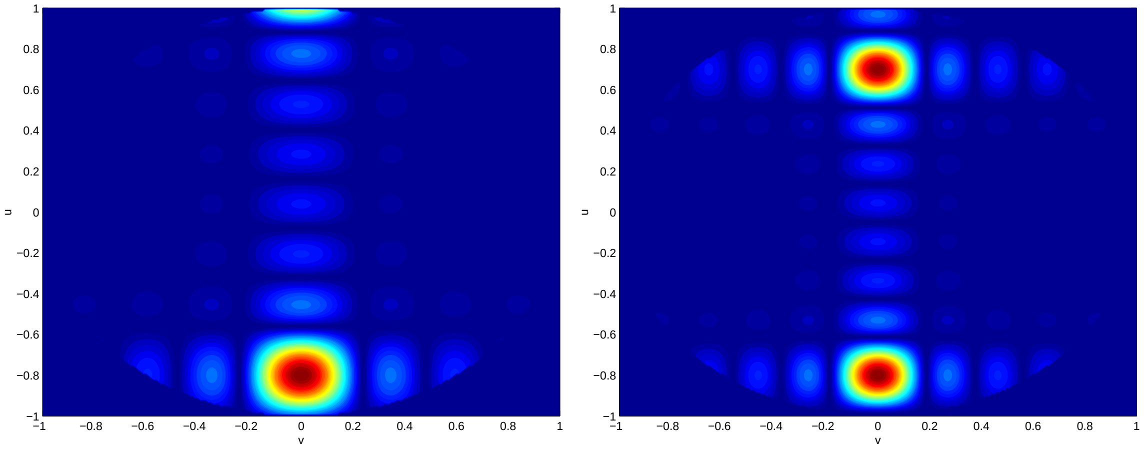


Figure 5.11: Intensity contour plots showing the beam pattern in voltage of an ideal THEA tile at 1227.6MHz (left) and 1575.42MHz (right) where the tile is aimed at $(u, v) = (-0.8, 0)$ or due South at about 37 degrees elevation. These plots clearly show the differences in the positions of the grating lobe and in the sizes of the main lobe.

Nyquist criterion to reduce the number of measurements. The RMS noise of the resulting map is 0.07K.

The LDS data cube is expressed in galactic longitude and latitude and the velocity with respect to the local standard of rest (LSR), while the THEA data cubes are expressed in u, v and the velocity along the line-of-sight (LOS). Furthermore the spatial as well as spectral resolution of the two instruments differ. Therefore the data needs to be expressed with respect to the same quantities in order to compare the results from THEA measurements to the results from the LDS. In this section the reduction steps applied to the LDS data will be described.

1. The spectral resolution of the LDS data is 1.03km/s, while THEA operates with a spectral resolution of 4.22km/s at 1420MHz. Therefore the LDS data was divided in consecutive series of 4 frequency planes and averaged over the frequency planes within each series. This step also reduced the number of data points by a factor 4 thus enabling to keep all data in memory during further processing. The final result was a data cube of 720 × 361 × 212 points.

2. It was concluded that the first and last frequency planes only contained some small high velocity clouds which would not be detectable by THEA. The relevant frequency planes were all found between the 70^{th} and the 150^{th} plane. Therefore the amount of data was reduced further by selecting only these planes for further processing thus reducing the data cube to 720 × 361 × 81 points. A total HI map based on this cube is shown in figure 5.12.

3. The coordinate transformation between galactic coordinates and the (u, v)-plane poses the problem that a regular grid in one coordinate system is mapped on a distorted grid in the other. Since the (u, v)-plane solves the convolution problem caused by the variable beam shape of a phased array telescope, it was decided to do the convolution in the (u, v)-plane. Since a regular grid greatly simplifies the convolution operation, a regular (u, v) grid with a spacing of 0.01 was defined and mapped on the (l, b)-plane. A

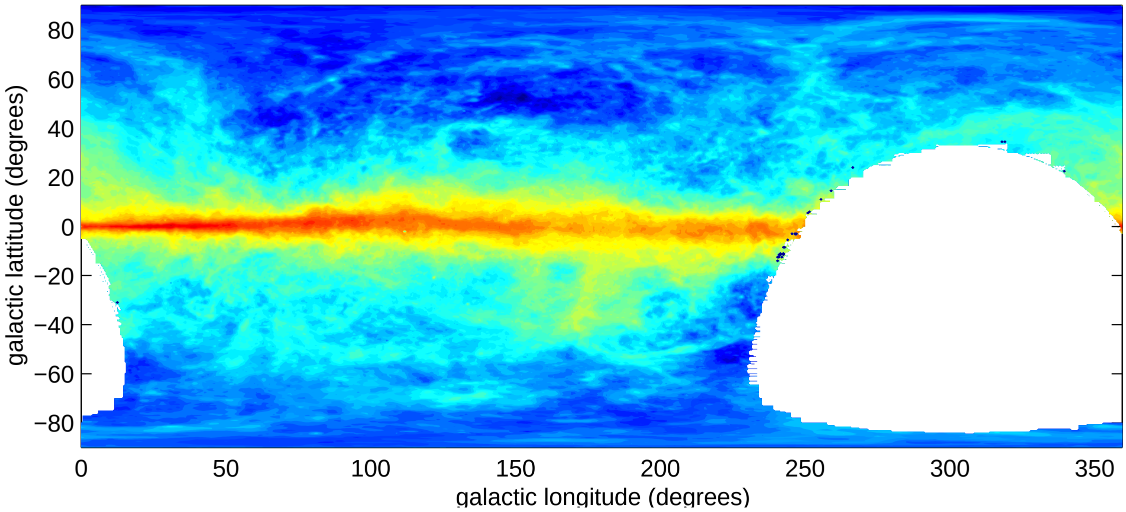


Figure 5.12: Total HI map based on the LDS data after selection of the relevant frequency channels. The intensities are plotted on a logarithmic scale in order to see the filamentary structure irrespective of the local intensity. The white areas are regions which can not be seen from the Dwingeloo.

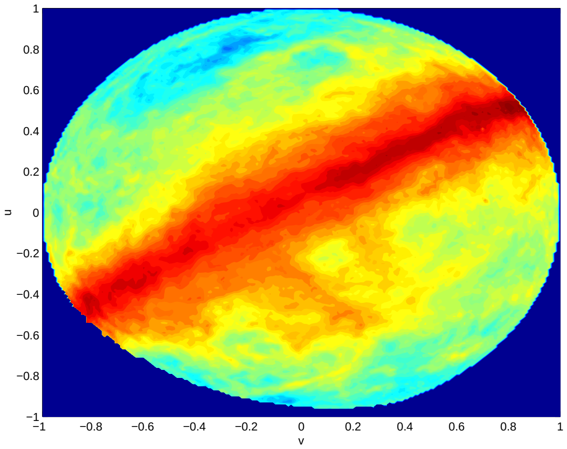


Figure 5.13: Total HI map based on the LDS data after projection on the (u, v) plane on April 4, 2003, 9:46.26 (GMT). The intensities are plotted on a logarithmic scale in order to see the filamentary structure irrespective of the local intensity.

bilinear interpolation was used to obtain the intensity values at the desired positions. This produces a 0.57° spacing in the (l, b)-plane in the zenith direction. This step results in a datacube of $201 \times 201 \times 81$ points, which is padded with zeroes on positions where $\sqrt{u^2 + v^2} > 1$. An example of a total HI map derived from such a data cube is provided in figure 5.13.

4. The points in the resulting (u, v, v_{LSR}) data cube were weighted with $\cos(\frac{1}{2}\pi\sqrt{u^2 + v^2})$ to account for the sensitivity pattern of the individual receiving elements.

5. The beam pattern was calculated on a grid with $-2 \leq u, v \leq 2$ and a spacing of 0.01

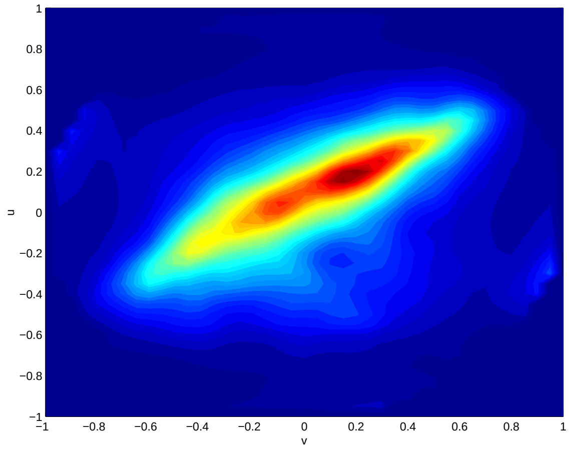


Figure 5.14: Total HI map of figure 5.13 after convolution with the beam of a dense array of 4 THEA tiles on the (u, v) plane on April 4, 2003, 9:46.26 (GMT). The intensities are plotted on a linear scale for easier comparison with the THEA data later on.

using equation (2.27) assuming an ideal THEA tile with omnidirectional receiving elements aimed at the zenith. This assumption can be made since the sensitivity pattern of the receiving elements is already taken into account in the previous step. The seemingly very large grid is needed to account for the full side lobe and grating lobe structure for every point on the map. Note that this equation produces a voltage beam. Since the measurements are done in autocorrelation this needs to be corrected to a power beam by multiplying the calculated complex beam pattern with its complex conjugate.

6. The frequency planes of the (u, v, v_{LSR}) data cube were convolved with the beam pattern calculated in the previous step using the standard Matlab routine for two dimensional convolution of two matrices. Weights were applies to the resulting images such that the intensity at every point was effectively the beam pattern weighted average of the original image with the beam pattern centered around that point.

7. The final result was obtained by downsampling the convolved cube by keeping only the spectra at (u, v) coordinates at which the actual measurement were done. The result of these steps when applied to the image in figure 5.13 is shown in figure 5.14.

5.10 Imaging of galactic neutral hydrogen with a singe tile

On May 23, 2002, the galactic neutral hydrogen was detected for the first time using a spectrum analyzer. It was concluded that the THEA backend should be capable of detecting the galactic neutral hydrogen easily since it offers a far more powerful way to integrate the incoming signal. This was confirmed by the aforementioned study on correlator noise, which didn't reveal any problems which would make HI detection impossible.

A scan with a single tile was made on November 25, 2002, on a regular (u, v)-grid with a spacing of 0.1, which corresponds to 5.7 degrees near the zenith, thus satisfying the Nyquist criterion, since a single tile has a HPBW of about 12 degrees at 1420MHz. The integration

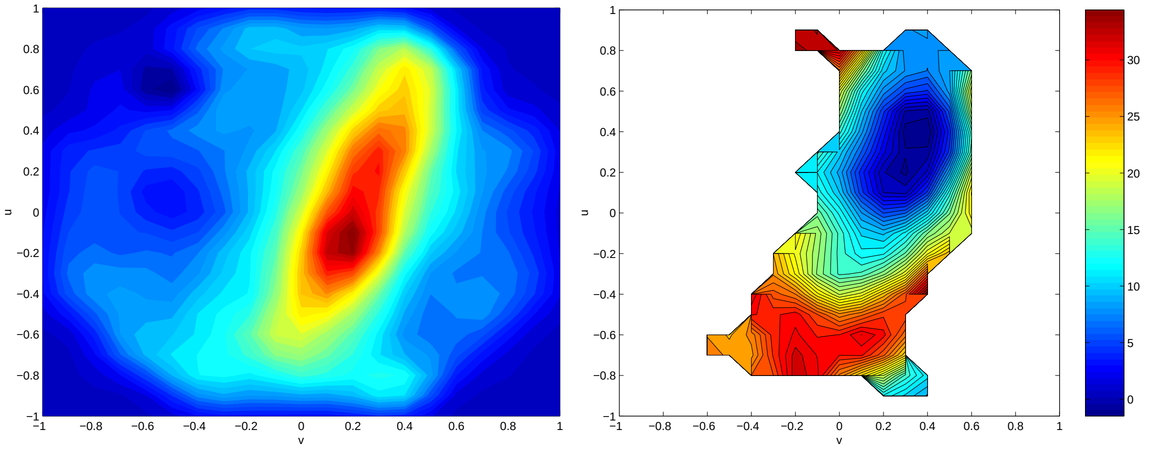


Figure 5.15: Total HI map and velocity field of galactic neutral hydrogen based on measurements with a single tile. The data was taken on November 25, 2002, 14:45.57 (GMT).

time of the individual points in the scan was 1s, allowing a full sky scan to be made in 5 minutes.

RFI was removed from the spectra using an automated procedure which compared the value in every frequency bin with the median of the 10 surrounding frequency bins. If the deviation was larger than 20% of the noise floor, the value was set to the median of the surrounding bins. This provided an effective way to remove all spurs from the spectra while keeping the smoother and possibly meaningful features.

After baseline subtraction using a linear interpolation over the frequency band between 1420.1 and 1420.7MHz, the individual frequency planes were convolved with a Hanning taper resulting in an effective HPBW of 12.7 degrees in the images. This datacube was used to produce a total HI map by simply adding all values along the frequency axis and a velocity field by calculating the first moment. This produced a velocity field which was hardly informative due to all meaningless values on points containing too little signal. Therefore only points with a 3σ detection in the intensity plot were kept. The resulting plots are shown in figure 5.15.

The velocity profiles from the THEA datacube were compared to the corresponding profiles from the LDS datacube after reduction as described in the previous section. Two of these profiles are shown in figure 5.16. These profiles were actually taken from the 4-tile measurement described in the next section and were chosen as illustration because of the structure in the profiles. The scaling factor needed to match the amplitude of the THEA profile to the amplitude of the LDS profile was used to determine the noise temperature of the THEA tiles used to do the HI measurements. This analysis suggested that tile 4 has a noise temperature of 181K and tile 3 has a noise temperature of 220K. Tile 4 has a lower noise temperature due to an additional gain of 3dB. This ensures that the noise budget of tile 4 is really dominated by the noise of the antenna elements and the first amplifier while the RF circuitry contributes significantly to the noise level of tile 3.

This point was clearly demonstrated by the first measurements done with the MPC. Since the MPC is able to determine the crosscorrelation between two beams and both corresponding autocorrelations based on the same data, it allows to normalize the crosscorrelation. In figure 5.17 the normalized crosscorrelations between the A and B beams of tiles 3 and 4 are shown. The A and B beams of the same tile share the antenna elements and the amplifier chain.

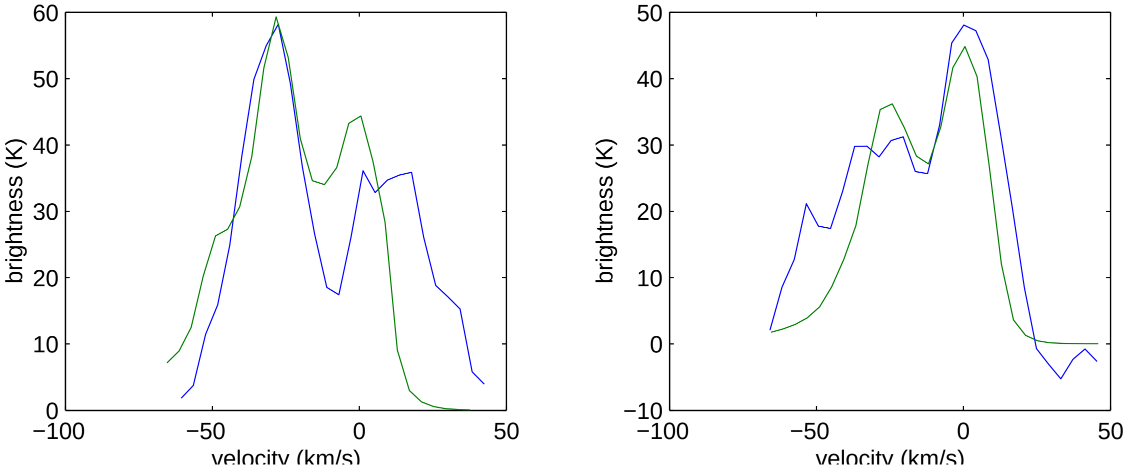


Figure 5.16: Plots showing a number of velocity profiles obtained from THEA (blue curves) and LDS data (green curves). The THEA data are scaled to match the intensities of the LDS data. This scaling factor was used to determine the number of counts per Kelvin brightness temperature.

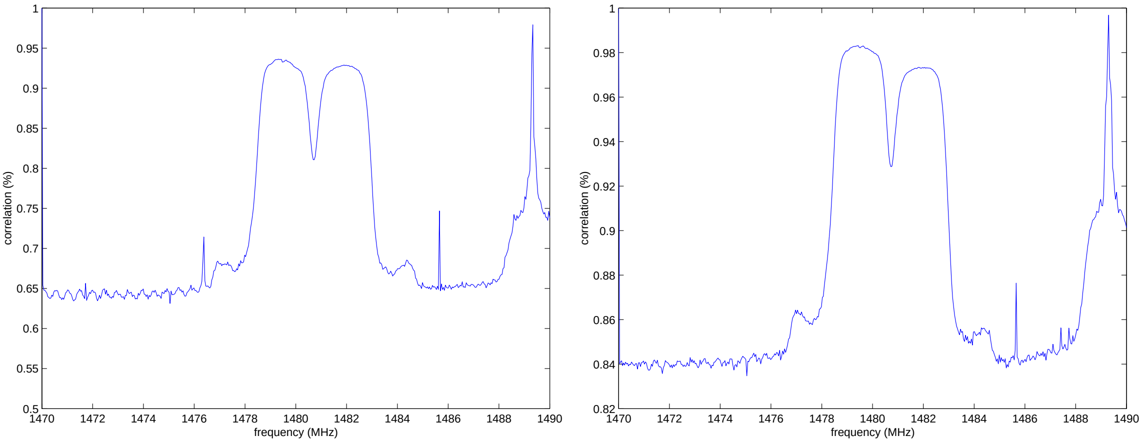


Figure 5.17: Normalized crosscorrelations between the A and B beams of tile 3 (left) and tile 4 (right) on Afristar.

Therefore a considerable part of the noise power in the two beams is correlated. Figure 5.17 shows that for tile 3 65% of the power is correlated and that for tile 4 this is even 84%.

This shows that about 150K of the noise (83% of 181K and 68% of 220K) is accounted for by the antenna elements and the first amplifier stage. The rest, 17% for tile 4 and 32% for tile 3, is produced in the second amplifier and the RF circuitry. The factor 2 between these contributions for tiles 3 and 4 match nicely with the 3dB additional gain of the latter.

These results show that the noise temperature of the system can be improved by inserting a higher gain. This method also has its disadvantages. The most important of these is that the tile becomes far more sensitive to RFI.

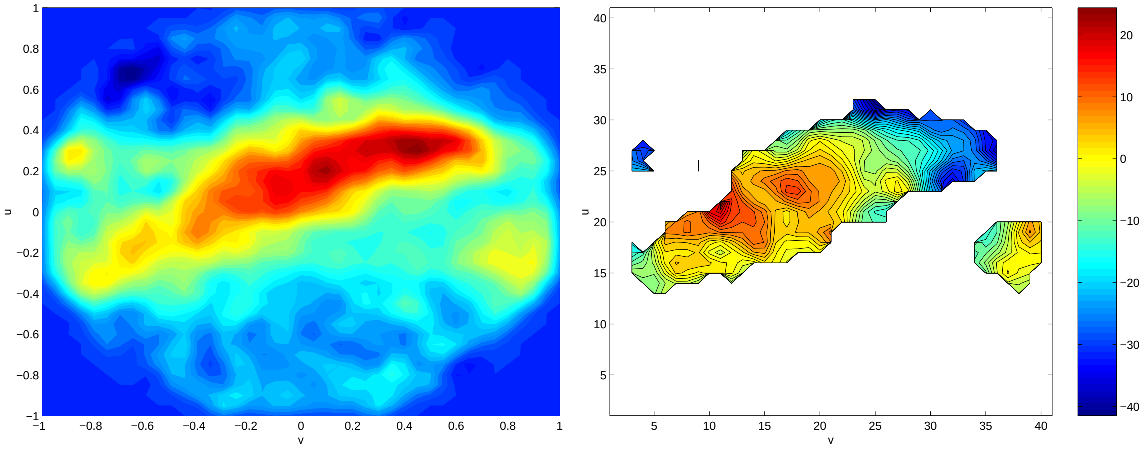


Figure 5.18: Total HI map and velocity field of galactic neutral hydrogen based on measurements with an array of 4 tiles. The data was taken on April 4, 2003, 9:46.26 (GMT). The expected result based on LDS-data is shown in 5.14.

5.11 Imaging of galactic neutral hydrogen with four tiles

Imaging of galactic neutral hydrogen was also done using an array of 4 tiles in a 2-by-2 configuration with tiles placed directly next to one another. The non-geometrical phase delays due to slight differences in cable lengths were calibrated out on Afristar as described in section 5.7. After the calibration the frequency was switched from 1480MHz to 1420MHz without changing other parameters. This trick works as long as the cable lengths differ by at most a few centimeters.

A scan was made on a regular (u, v)-grid with a spacing of 0.05 between points. The integration time was again set to 1s, but due to the much larger number of points, the scan time increased to 23 minutes. This may produce a small distortion in the final image due to sky rotation. This effect will however hardly be visible since neighboring points are still measured within 1 minute.

The data was reduced following the same procedure as outlined in the previous section. The results are shown in figure 5.18 and clearly show the improvement in resolution and S/N due to the use 4 tiles. The corresponding intensity plot based on LDS data is shown in figure 5.14.

Comparison of the velocity profiles of THEA and LDS data resulted in a noise temperature of 186K for the array of 4 tiles. Based on the assumption that the noise temperature of tiles 1 and 2 were the same as the noise temperature of tile 3, the expected noise temperature was expected to be 210K, which is the RMS value of the noise temperatures of the individual tiles. Although the errors on the measured noise temperatures are about 10K, the difference between the expected value and the measured value is considerably larger than the errors. This indicates that either tile 3 is not representative for tiles 1 and 2 or that the calculation leading to the expected value is not applicable, for example due to changes in the effective area per tile. Either possibility requires further investigation.

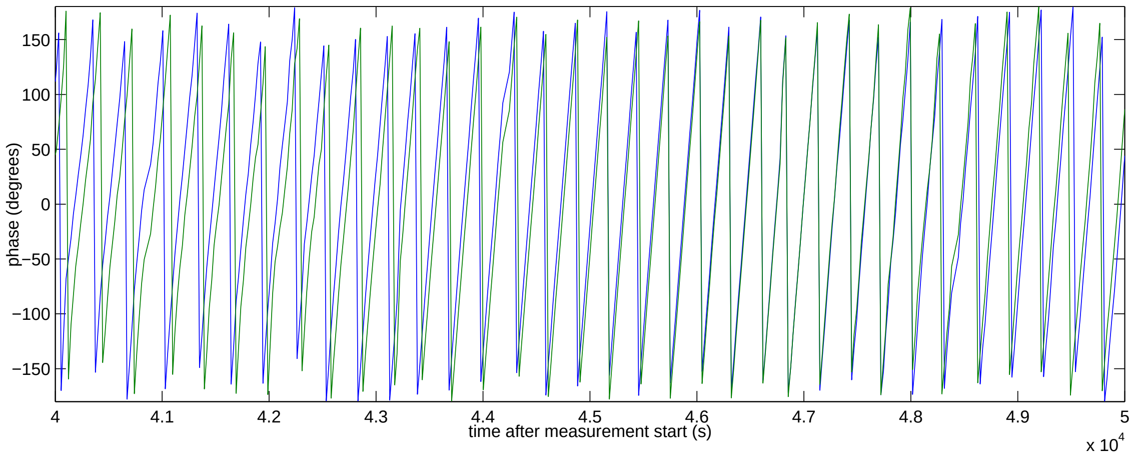


Figure 5.19: Comparison of the measured fringe phase at 1432.5MHz (blue) and 1447.5MHz (green) after subtraction of the correlator offset. This comparison clearly shows the different rates at which the fringe phase varies at the two frequencies.

5.12 Solar fringes

This year's practical exercise in radio astronomy for students from the University of Groningen has focused on measuring solar fringes. The results which will be presented below were measured at 1440MHz on March 14 using two tiles placed 10m apart on an East-West baseline. The tiles were aimed at the meridian passage of the sun. Measurement points have been taken on 30s intervals with 10s integration per measurement. The total measurement lasted 5.5 hours.

The first plots of the real and imaginary part of the correlation results clearly showed a correlator offset. Unfortunately this offset also showed drift effects. Therefore a combination of a linear and a sinusoidal drift term were fitted to the offset and the result subtracted from the data. The results are shown in figure 5.20. The presence of the correlator offset points to pickup of common signals. Since the tiles were placed 10m apart, it is unlikely that they pickup leakage from each others electronics. Therefore the correlator offset is probably caused by LO sharing. This would also provide a reasonable explanation for the drift seen in the correlator offset.

These results were obtained by taking the median of all 1024 point in the 20MHz band centered around 1440MHz. Theoretically the phase of the fringes should change faster at higher frequencies. Since a large number of fringe periods have been measured, it should be possible to detect this effect within the 20MHz band. Therefore the median was taken over the first 256 channels centered around 1432.5MHz and the last 256 channels centered around 1447.5MHz. The resulting fringe phases are shown in figure 5.19.

Since the center frequencies of the fringes in figure 5.19 are 15MHz or about 1% of the center frequency of the full 20MHz band apart, one would expect an additional phase change of 90% in about 25 fringe periods. This agrees very well with the relative phase changes as seen in the figure.

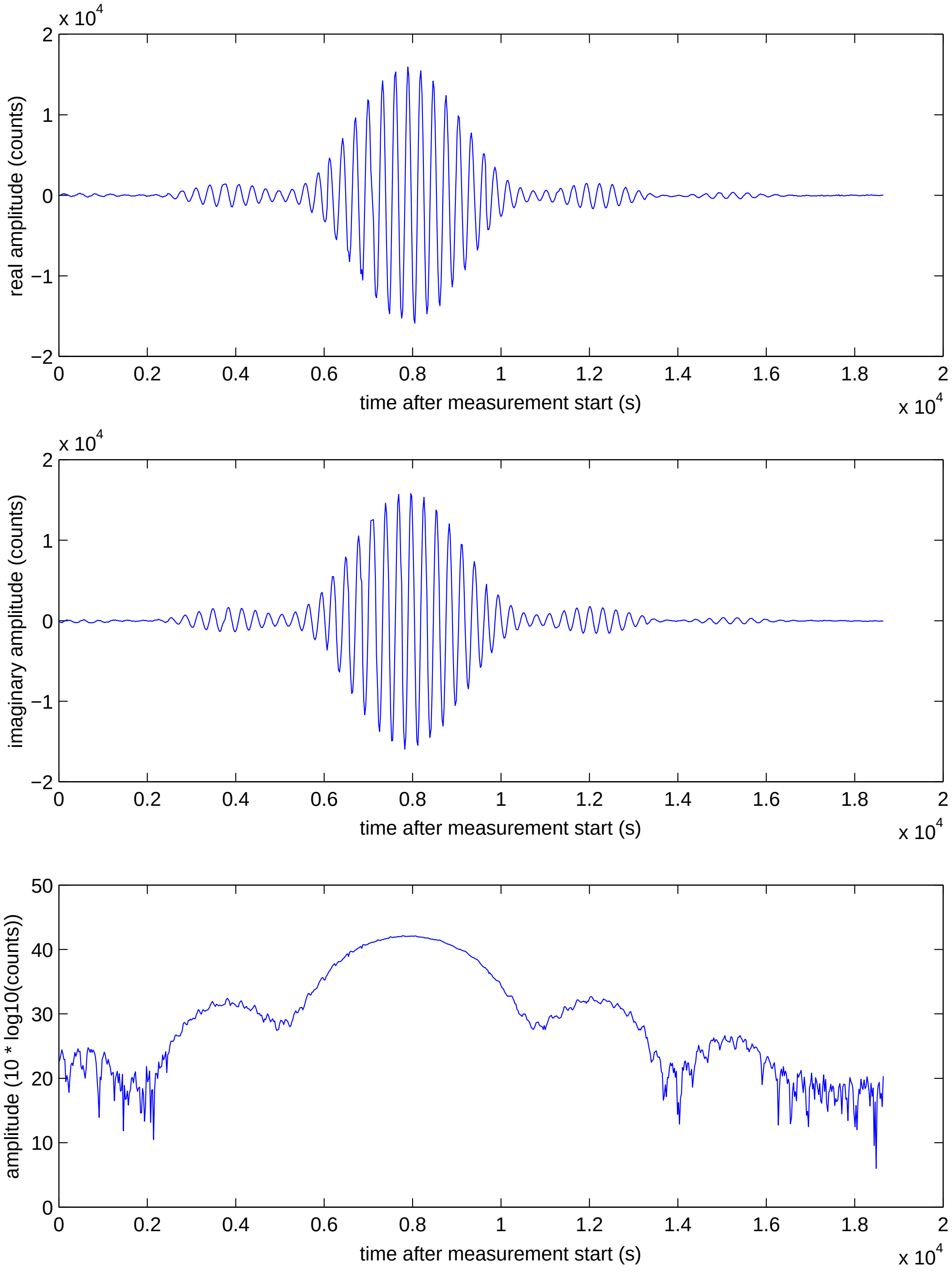


Figure 5.20: Results from solar fringe measurements. The upper plot shows the real part of the signal, the middle plot the imaginary part and the lower plot the total amplitude on a logarithmic scale. All plots present the result after subtraction of the correlator offset.

Chapter 6

Summary and conclusions

As mentioned in the introduction the aim of my thesis work at ASTRON was threefold:

1. further integration of THEA to a single measurement system
2. writing of proper user-oriented documentation
3. conduction of a number of evaluation and demonstration measurements

The necessity of further system integration resulted directly from the way THEA was developed. The development of THEA was split into development of different parts of the THEA system. The result was a system which was able to process data, but couldn't be controlled in a way to do all tests we wanted to do. These problems were solved by changing the data processing from a triggered process to a continuing process. This allowed the construction of a Matlab function which simply synchronized itself with the data acquisition process and changed beams at the right moments. The final result was a system which could be controlled using Matlab scripts. Furthermore control from Matlab allowed to make modifications to the software in a convenient way, resulting in additional features such as multifrequency multibeaming and array beam forming.

Since a detailed knowledge of the system was required for the development and use of these scripts, it provided me with the knowledge needed to produce the corresponding documentation. This documentation was written in HTML-format, which turned out to be a good decision due to the convenience offered by links. This help system was thoroughly tested during the practical exercise in radio astronomy of a group of students from the University of Groningen, which showed that the help system provides all information needed to get new users started.

The operation of the system was evaluated and demonstrated by a considerable number of experiments described in chapter 5. The evaluation of the system started with a number of noise measurements in the correlator. It was shown that correlation between two white noise sources produced a signal amplitude of 0.2% of the total signal power which was good enough for the measurements presented in this thesis. Furthermore these measurements showed that some kind of saturation effect pollutes the result when integration times longer than a few minutes are used.

A holographic measurement technique was used for outdoor calibration on a celestial source. The results showed that this calibration technique is able to attain the maximum possible accuracy, which was not limited by measurement errors but by the accuracy of the vector modulators. The inaccuracies of the vector modulators limited the accuracy in the gain settings to 0.9dB and 3.2 degrees. The amplitude errors are large relative to the phase errors due to systematic deviations in the vector modulator settings. These deviations are probably due to the fact that the vector modulators were calibrated outside the THEA tiles.

Imaging of the sky background continuum over the band from 1000MHz to 1600MHz showed spatial and spectral structure in the noise background of the THEA tiles. The general features of this structure could be accounted for by assuming pickup from surrounding trees. It also showed a number of point like continuum sources which could not all be explained. It seems however that there is at least some pickup from either the ASTRON building or the Leiden-Dwingeloo telescope.

Next the array beam forming ability of THEA was demonstrated by a consecutive series of scans at 1480MHz (Afristar) using 1, 2 and 4 tiles. The multifrequency multibeaming ability was demonstrated by imaging the GPS satellites at the L1 and L2 frequencies simultaneously.

Imaging of galactic neutral hydrogen was done using arrays of 1 and 4 tiles. The results were compared with the results from the Leiden-Dwingeloo Survey and were found to be in reasonable agreement. The noise temperature of the THEA system was derived from these measurements by fitting the LDS velocity profiles to the THEA velocity profiles. It was concluded that the noise temperature of tiles 3 and 4 were 220K and 181K respectively. It was shown that this difference is in agreement with the modifications made to tile 4. The noise temperature of the array turned out to be 186K, which is lower than the expected noise temperature of 210K. No fitting explanation for this difference has been found yet.

Finally interferometric fringes from the sun were successfully detected using two tiles on an East-West baseline. In principle this shows that future practical exercises may focus on synthesis imaging, but this effort may be hampered by a serious correlator offset. The correlator offset is probably caused by LO sharing.

All in all it may be concluded that the THEA system is now suitable for astronomical observations as part of practical exercises. This also shows that the phased array concept may be an interesting option for future radio telescopes.

Chapter 7

Further work

7.1 Deterministic nulling and beam forming

A possibly very interesting feature of a phased array antenna is its ability to change its beam shape by assigning different weights to different antenna elements. This feature may be used to reduce the side lobe level as compared to the main lobe, but deterministic nulling seems far more interesting.

Deterministic nulling allows to specify the locations of a limited number of zeroes in the beam pattern. By placing this zeroes in those directions where there are strong sources of interfering signals, deterministic nulling can be used as an RFI reduction technique. Theoretically this works very well since the beam pattern can be calculated and modified very accurately, but in practice there are some additional difficulties as illustrated in figure 7.1.

The left hand image shows the beam pattern of an ideal tile on a dB scale, while the right

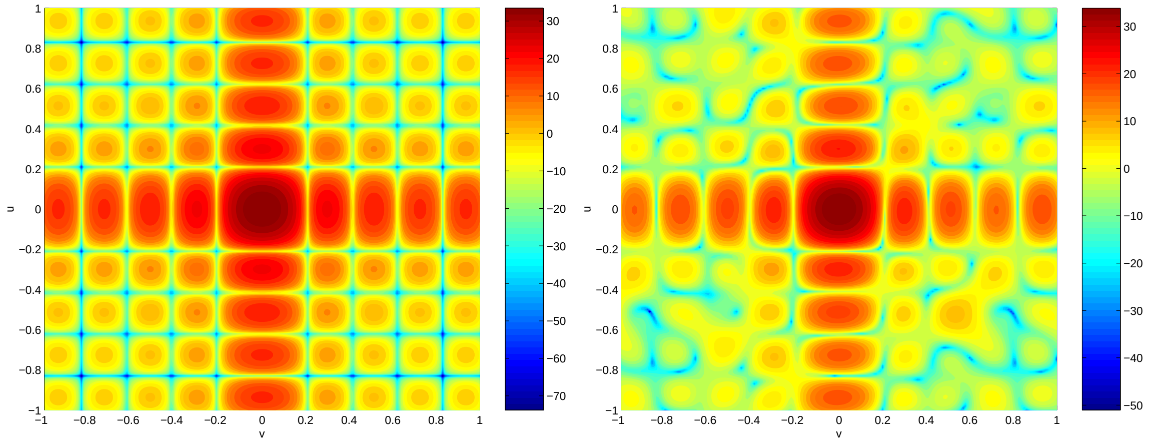


Figure 7.1: The left hand plot shows the ideal beam pattern for a THEA tile at 21.106cm. The beam pattern shown in the right hand plot results from a tile with normally distributed amplitude and phase errors with standard deviations of 0.5dB and 4.3° respectively. Comparison of the two plots shows that the general pattern isn't really affected, but that the deep zeroes in the perfect beam pattern have become more diffuse and are no longer at well determined positions.

hand image shows the beam pattern of the same tile after adding normally distributed random noise to the real and imaginary parts of the antenna gains of the individual elements such that the standard deviation in phase and amplitude were 4.3° and 0.5dB respectively. These numbers were chosen, since they closely resemble the accuracy which has been achieved in the THEA tile calibration. This plot demonstrates that the zeroes have become more diffuse and have changed their position somewhat.

The diffuseness of the zeroes is not necessarily a problem from a deterministic nulling point of view as long as the zeroes are deep enough to effectively null a specific RFI source. The change in position may pose a problem because this change depends on the randomly distributed gain errors and can therefore not be predicted. This may hamper an effort to place one of these zeroes at a specific position. This problem may be solved by the use of genetic algorithms, but at this stage it isn't possible to determine whether this will work and if it works, whether the THEA tile can be controlled accurately enough to produce the specified beam pattern.

7.2 Outdoor calibration in the presence of multiple sources

The calibration procedure used for the measurements in this report are based on calibration measurements using a single source. This calibration scheme was tested outdoors on Afristar, since Afristar is the only visible source at that frequency. In general other sources may be present. From a theoretical point of view the theory presented in this report can easily be extended to deal with multiple sources or even signal distributions over the sky. A practical implementation of this extended version of the theory still needs to be developed.

The main problem here is that the signal should be measured as a complex quantity, i.e. it should be measured in amplitude as well as phase. The phase of the signal can only be measured in comparison with a reference signal. So the solution of the problem probably lies in the answer to the question what should be used for reference signal.

It seemed natural to use one of the detectable sources as reference. This idea was tested on the GPS-satellites, which resulted in an image which clearly showed the reference satellite, but also showed the other satellites being picked up by the side lobes of the reference beam. This result shows that the signals from different GPS satellites are coherent but attenuated by the side lobe level of the reference beam, since otherwise all satellites would have been clearly visible. Although this is an interesting conclusion, the resulting image was not suitable for calibration purposes.

From this result it is clear that the theory needs to be extended a little further to explicitly include the reference beam, the effects of this beam and furthermore the effects of different parts of the sky having different degrees of coherency. Once these aspects have been worked out, this theory can be implemented on a computer to put it on a test using GPS satellites and at a later stage neutral hydrogen, although I doubt that the latter signal is strong enough to produce a suitable signal-to-noise ratio for calibration measurements.

7.3 Correct beam forming over the full 20MHz band

The beam settings of the THEA system are calculated based on the center frequency of the 20MHz measurement band. The phase settings of the antenna elements should however be different for signals at different frequencies. Unfortunately this feature can not be implemented in the THEA frontend since the vector modulators can only be set to one specific

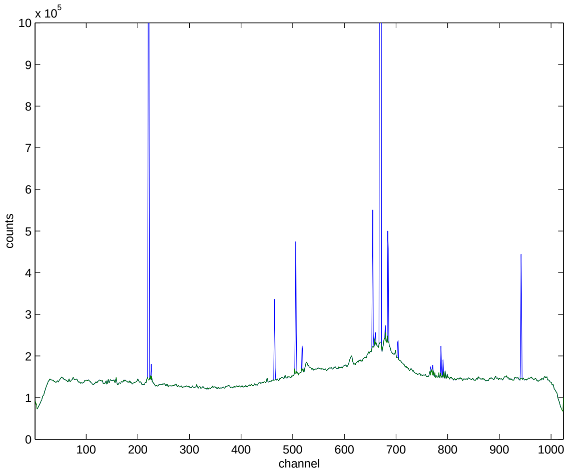


Figure 7.2: Plot showing results from the current RFI removal scheme. The raw data is presented in blue. The RFI removal procedure searches for peaks and changes its value to the median of the 20 surrounding points. The figure also demonstrates the importance of RFI-removal if astronomical sources are to be detected.

value.
Since a single THEA tile is quite small, it has a very large beam and the deviations due to this effect over a 20MHz band around typical measurement frequencies like 1480MHz are quite small, especially near the zenith. If the beam is steered towards lower elevations this effect will aggravate. At about 70 degrees from zenith, the positional error over the 20MHz band is of the order of 1 degree. This gets even worse at even lower elevations, but these elevations are irrelevant since THEA is not meant to be used at those elevations.
It would be interesting to know how this problem affects the beam forming of an array of THEA tiles, since this will produce a smaller beam, which aggravates the problem in a relative sense. If it turns out that only the amplitude is affected, it may be possible to correct the problem afterwards by applying different weights to different frequency bins. When using the MPC, this scheme may be applied to correct this problem at an early stage thereby still allowing correct beam forming over the entire array.

7.4 RFI mitigation at the input

RFI mitigation is an important issue as is illustrated by figure 7.2, in which a specific spectrum from the datacube produced for the single tile HI measurement in this report is shown before and after RFI removal. The RFI was removed by searching for peaks and changing the value of the peak to the median value of the 11-point interval centered around it.
Although this is a very simple algorithm, it works quite well as long as the RFI signals don't affect the shape of the noise floor itself. As can also be seen in the figure this is not necessarily true. This problem may be solved by fitting a specific function to the RFI peaks or the pass band.
One may however also try to filter RFI at the input. This idea can be tested on the MPC. One may, for example, try to remove peaks from the FFT result before the spectra are correlated and integrated. This will require some additional computational effort and thus slow down

the MPC, but it may allow to demonstrate the effectiveness of this idea, which may thus prove interesting for LOFAR and/or SKA. Another point in the data processing where one may try to remove RFI is after the correlation stage, since the crosscorrelation is in principle an effective way to reduce RFI which is not picked up by both beams. The crosscorrelation results may therefore be used to clean the autocorrelation results.

7.5 Synthesis imaging

The successful detection of solar fringes indicates that it should be possible to do synthesis imaging with THEA. A key concept in synthesis imaging is the concept of visibilities. The theoretical framework presented in chapter 2 provides a good starting point to describe the visibilities in terms of the signals measured by the two tiles involved. Although some theoretical work still needs to be done to arrive at a number of directly applicable equations, it is expected that this is quite straightforward if one has some basic knowledge of synthesis imaging.

The correlator offset will probably pose a greater challenge. In order to do meaningful synthesis measurements, the data needs to be corrected for this correlator offset. There are several ways to do this, but all methods are based on some kind of calibration before and after the actual measurement. In order to apply such method one needs to have a clear understanding about the behavior of the correlator offset to set a limit on the number of calibrations to be done during a given time interval.

In this thesis it is suggested that the main contribution to the correlator offset comes from LO sharing. If this proves to be true one may want to reduce the correlator offset by using different LO's for different tiles to reduce the amount of common signals, which is fed into the tiles.

Once the practical and theoretical problems are solved, a lot of new possibilities will become available. Due to the longer baselines, which can be used in synthesis imaging, one may resolve the galactic neutral hydrogen to a level allowing to detect point like radio sources such as Cas A and Cyg A. One may even try to detect absorption features in front of these sources. Another probably easier experiment can be aimed at resolving the sun.

7.6 Polarization experiments

One of the features of THEA, which is completely neglected in this thesis, is the polarization of the system. The Vivaldi elements should behave like polarized receivers. This makes it interesting to fiddle around with the orientation of the tiles, for example to study the symmetry in the sensitivity pattern of the individual elements. If this pattern differs between the direction of polarization and the direction perpendicular to the polarization, the intensity of an unpolarized source will seem different if the tile is rotated 90 degrees.

If the sensitivity pattern is hardly effected by this rotation, one may even want to measure unpolarized sources with an array of THEA tiles having different orientations since the grating lobe pattern will rotate together with the tile. This measurement setup can thus be used to reduce the power in the grating lobes while the power in the main lobe stays the same.

It will be clear that for experiments involving polarization it is extremely important to use a coordinate system, which is able to deal with the relative mutual orientations of the THEA tiles. This coordinate system is not defined yet and will require a new update of the THEA control software to get it working.

7.7 Drift

Most results presented in this paper are based on short measurements. Since the THEA system is now ready for longer and more complicated measurements, stability of the system becomes a more important issue. It has for example already been mentioned that the correlator offset, which needs to be removed for synthesis imaging, shows drift and when using the 4 THEA tiles as an array, it is interesting to know how long the phase offsets between the tiles can be regarded as constant.

These examples show that a thorough knowledge of drift phenomena in the system and their time scales is important. Therefore a number of stability tests need to be done providing data on those drift phenomena, which are important to the practical measurements.

7.8 Shadowing

One of the disadvantages of a phased array telescope is that its performance degrades rapidly toward lower elevations. It has therefore been proposed to tilt a number of tiles in the future LOFAR and SKA projects thereby selecting an astronomically more interesting part of the sky. This may cause some tiles to be standing in the shadow of other tiles. The exact shape of this shadow is yet unknown, i.e. it is not known whether it is just the same as the geometrical shadow or larger or smaller than the geometrical shadow.

Since there have been people thinking about this problem for some time, a tilt unit has been made for one of the tiles. In order to study tilt effects it might be convenient to tilt at least two tiles. This can be realized simply by putting some stones under one side of the tile. Once the tiles are tilted the beam of the shadowed tile can be determined by the calibration procedure outlined in this thesis. The first experiment can be done using Afristar, but if even lower elevations are needed, an artificial signal can be generated.

The most sensitive test will probably be one in which the tilted tile is calibrated on the used source first, then shadowed and then calibrated again. The difference found in the gain factors will provide information on the signals received by the individual elements of the shadowed tile.

Bibliography

[1] A. Bart Smolders. System Specification THEA. Technical Report THEA_specs_sys, NFRA, March 1999.

[2] Peter Duffet-Smith. *Practical Astronomy with Your Calculator.* Cambridge University Press, 1979.

[3] Athanasios Papoulis. *Signal Analysis.* McGraw-Hill Book Company, 1977.

[4] Philip N. Sabes. Linear Algebraic Equations, SVD, and the Pseudo-inverse, October 2001. found at *http://www.keck.ucsf.edu/˜sabes/doc/SVDnotes.pdf.*

[5] Arie Doorduin. THEA Front-end Processor. Technical Report Description-001 26640-1-5-3-R1, ASTRON, January 2002.

[6] Gijs W. Schoonderbeek. DBF, a Beam Former Implementation. Technical Report Description-001 26640-2-1-1-R1, ASTRON, January 2002.

[7] Gijs W. Schoonderbeek. DBF Test Document. Report, ASTRON, October 2001.

[8] Sylvain Alliot. Overview of the Selection / Cache Storage Board. Technical Report Description-002 26640-2-2-1-R1, ASTRON, November 2001.

[9] Sylvain Alliot. Technical Description THEA Selection / Cache Storage Board. Technical Report Description-003 26640-2-2-1-R1, ASTRON, November 2001.

[10] Harm Jan Pepping. Assembling and Testing of a Data Acquisition System, August 2001. Thesis work to pass at the Windesheim University.

[11] Dap Hartmann. *The Leiden/Dwingeloo Survey of Galactic Neutral Hydrogen.* PhD thesis, Leiden University, October 1994.

[12] Dap Hartmann and Burton W.B. *Atlas of Galactic Neutral Hydrogen.* Cambridge University Press, 1997.

[13] Grant A Hampson and A. Bart Smolders. A fast and accurate scheme for calibration of active phased array antennas. In *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting*, July 1999.

[14] Michel Arts. Description and Evaluation of the THEA Calibration Scheme. Technical report, ASTRON, December 2002.

[15] Sebastiaan van der Tol. Calibration of THEA, January 2003. Draft version of report on a research project at ASTRON in the framework of the author's master course at the Delft University of Technology.

Appendix A

Abbreviations

ADBF Adaptive Digital Beam former
ASTRON Astronomisch Onderzoek Nederland (NFRA)
A/D Analog/Digital
B byte
BF Beam Former
CBF Column Beam Former
CPU Central Processing Unit
CVS Concurrent Versions System
CW Continuous Wave
DAQ Data Acquisition
DC Direct Current
DMA Direct Memory Access
DOA Direction of Arrival
DSP Digital Signal Processor
EMI Electro-Magnetic Interference
FEC Frontend Controller
FSB Front Side Bus
FFT Fast Fourier Transform
FPGA Field Programmable Gated Array
GPIB General Purpose Interface Bus
GPS Global Positioning System
GUI Graphical User Interface
HPBW Half-Power Beam Width
HSL High Speed Link
HTML Hypertext Mark-Up Language
IDC IF to Digital Converter
IF Intermediate Frequency
I/O Input/Output
LDS Leiden-Dwingeloo Survey
LNA Low Noise Amplifier
LO Local Oscillator
LOFAR Low Frequency Array
LOS line-of-sight
LSR Local Standard of Rest

MEX Matlab Executable

MUX multiplexer

MPC Microprocessor Correlator

NFRA Netherlands Foundation for Research in Astronomy

PCI Peripheral Component Interconnect

PMP8 8 Parallel Multi-Processor board

RAM Random Access Memory

RAP Reduction, Acquisition and Processing

RBF Row Beam Former

RF Radio Frequency

RFI Radio Frequency Interference

RS422 Recommended Standard 422

SKA Square Kilometer Array

THEA Thousand Element Array

THEP THEA Experimental Platform

TIM Twin Input Module

UDP Unreliable Datagram Protocol

Appendix B

Matlab sources

B.1 initTile.m

```
function T = initTile(fA, fB, tile)

% T = initTile(fA, fB, tile)
%
% This function initializes a ThEA tile object and returns the result
%
% parameters:
% fA    frequency of the A beam. This parameter is used for loading calibration data
%       (if available) and for calculating the beam settins. It will not tune the
%       LO synthesizers to the required frequencies.
% fB    frequency of the B beam. See also the remarks for fA.
% tile  tile number. Note that tile 1 physical has tile number 0 digitally, tile 2 has
%       tile number 1, etc
%
% return value:
% T     the resulting ThEA tile object
%
% SJW, 2002

vmtable = '/home/thea/calibration/vmtableKlaas';
host    = 'huisjewest.thea.nfra.nl';
caldir  = '/home/thea/calibration/caltest';

T = theaTile('vmtable',    vmtable, ...
             'debugmode',  'on', ...
             'host',       host, ...
             'tile',       tile, ...
             'caldir',     caldir);
T = theaSetFrequency(T, fA, fB);
T = theaClearBeamTable(T);
```

B.2 AppendBeams.m

```
function [beamidx, nbeams] = AppendBeams(T, uA, vA, uB, vB, correctA, correctB)
```

```
% [beamidx, nbeams] = AppendBeams(T, uA, vA, uB, vB, correctA, correctB)
%
% This function calculates beams based on the (u, v) coordinates specified by the
% user and appends these beams to the beam table of the desired ThEA tile.
%
% parameters:
% T          ThEA tile object specifying th tile to which the beams should be written
% uA         1-by-dimu or dimu-by-1 vector containing the u-coordinates for the A beam
% vA         1-by-dimv or dimv-by-1 vector containing the v-coordinates for the A beam
% uB         vector containing the u-coordinates for the B beam, should have same size as uA
% vB         vector containing the v-coordinates for the B beam, should have same size as vA
% correctA   8-by-8 matrix containing additional complex corrections which should be
%            applied to the antenna elements while steering the A beam. If no modifications
%            are required, use ones(8).
% correctB   ibid. for B beam
%
% Note on beam calculations: uA and vA are used to calculate points on a regular
% rectangular grid. Only points which lie within the unity circle in the uv-plane are taken
% into account. The same procedure is used for uB and vB. If either the coordinates for the
% A beam or the coordinates for the B beam are outside the unity circle the entire point
% (four coordinates) is skipped.
%
% return values:
% beamidx    dimu-by-dimv matrix containing the indices of the corresponding beam
s
%            in the beam table of the tiles
% nbeams     the number of beam that have actually been added to the beam table
%
% SJW, 2002

dimu = max(size(uA));
dimv = max(size(vA));

nbeams = 0;
for uidx = 1:dimu
  for vidx = 1:dimv
    if ((sqrt(uA(uidx)^2 + vA(vidx)^2) < 1) & (sqrt(uB(uidx)^2 + vB(vidx)^2) < 1));
      phiA = atan2(-vA(vidx), uA(uidx));
      thetaA = asin(sqrt(uA(uidx)^2 + vA(vidx)^2));
      phiB = atan2(-vB(vidx), uB(uidx));
      thetaB = asin(sqrt(uB(uidx)^2 + vB(vidx)^2));
      [T, beamidx(uidx, vidx)] = FECAppendBeam(T, thetaA, phiA, thetaB, phiB,
                                         3.5, 3.5, correctA, correctB);
      nbeams = nbeams + 1;
    end
  end
end
```

B.3 uvscan.m

```
function directory = uvscan(T, uA, vA, uB, vB, beamidx, nbeams, nscans)
```

```
% directory = uvscan(T, uA, vA, uB, vB, beamidx, nbeams, nscans)
%
% This function performs the actual sky scan based on the results from the
% AppendBeams function and assuming that the RAP correlator has been started and is
% writing its spectrum file to /home/thea/studentenpracticum/data/spectrum.dat on
% kant.nfra.nl.
%
% parameters:
% T         ThEA tile object which should make the scan
% uA        1-by-dimu or dimu-by-1 vector containing u-coordinates for beam A
% vA        1-by-dimv or dimv-by-1 vector containing v-coordinates for beam A
% uB        1-by-dimu or dimu-by-1 vector containing u-coordinates for beam B
% vB        1-by-dimv or dimv-by-1 vector containing v-coordinates for beam B
% beamidx   dimu-by-dimv matrix generated by the AppendBeams function contaiing the
%           indices of the beams loaded in the ThEA tile
% nbeams    the number of beams which have been added to the beam table of the ThEA
%           tile by the function AppendBeams
% nscans    the number of scans to make
%
% The coordinate vectors are used according to the same convention as used in th
e
% function AppendBeams.
%
% return value:
% directory full path name of the first directory to which data has been written.
%           Note that the directory names of the directories created by this
%           function contain a time stamp of the starttime of the scan.
%
% SJW, 2002

dimu = max(size(uA));
dimv = max(size(vA));

for n = 1:nscans
  timestamp = datestr(now);
  timestamp = strrep(timestamp, ' ', '_');
  timestamp = strrep(timestamp, ':', '');
  if (n == 1)
    directory = sprintf('/home/thea/studentenpracticum/data/scan%s', timestamp);
  end
  cmd = sprintf('mkdir /home/thea/studentenpracticum/data/scan%s', timestamp);
  dos(cmd);
    for idx_u = 1:dimu
    for idx_v = 1:dimv
      if ((sqrt(uA(idx_u)^2+vA(idx_v)^2)<1) & (sqrt(uB(idx_u)^2+vB(idx_v)^2)<1));
        FECSelectBeam(T, beamidx(idx_u, idx_v));
        info = dir('/home/thea/sjw/studentenpracticum/spectrum');
        olddate = info.date;
        info = dir('/home/thea/studentenpracticum/data/spectrum');
        newdate = info.date;
        while strcmp(olddate, newdate)
          pause(0.1); %reduce the number of interrupts...
          info = dir('/home/thea/studentenpracticum/data/spectrum');
          newdate = info.date;
        end
        while info.bytes < 10000
          info = dir('/home/thea/studentenpracticum/data/spectrum');
        end
        cmd = sprintf('cp /home/thea/studentenpracticum/data/spectrum'
                ' /home/thea/studentenpracticum/data/scan%s/idxu%didxv%d.dat &',
                timestamp, idx_u, idx_v);
        dos(cmd);
        nbeams = nbeams - 1;
        disp(sprintf('%d', nbeams));
      end
    end
  end
end
```

B.4 uvread.m

```
function [real, imag] = uvread(directory, uA, vA, uB, vB)

% [real, imag] = uvread(directory, uA, vA, uB, vB)
%
% This function reads the data as produced by the function uvscan from file.
%
% parameters:
% directory     directory to which the data has been written by the function uvscan without ending
% uA            1-by-dimu or dimu-by-1 vector containing u-coordinates for the A beam
% vA            1-by-dimv or dimv-by-1 vector containing v-coordinates for the A beam
% uB            1-by-dimu or dimu-by-1 vector containing u-coordinates for the B beam
% vB            1-by-dimv or dimv-by-1 vector containing v-coordinates for the B beam
%
% The coordinate vectors are used according to the same convention as used in the
% function AppendBeams.
%
% return values:
% real  dimu-by-dimv-by-1024 (assuming the RAp is set to perform a 1024 point FFT)
%       matrix containing the real parts of the correlator results. If the
%       corresponding (u,v) point lies outside the unity circle, the matrix is padded
%       with zeros.
% imag  dimu-by-dimv-by 1024 matrix containing the imaginary parts of the correlator
%       result. If the corresponding (u,v) point lies outside the unity circle,the
%       matrix is padded with zeros.
%
% SJW, 2002

dimu = max(size(uA));
dimv = max(size(vA));

for idx_u = 1:dimu
  for idx_v = 1:dimv
    if ((sqrt(uA(idx_u)^2 + vA(idx_v)^2) < 1) & (sqrt(uB(idx_u)^2 + vB(idx_v)^2) < 1));
      file = sprintf('%s/idxu%didxv%d.dat', directory, idx_u, idx_v);
      spectrum1 = dlmread(file, ' ', [0 0 1023 0]);
      spectrum2 = dlmread(file, ' ', [0 1 1023 1]);
      real(idx_u, idx_v, :) = spectrum1';
      imag(idx_u, idx_v, :) = spectrum2';
    else
      real(idx_u, idx_v, :) = zeros(1, 1024);
      imag(idx_u, idx_v, :) = zeros(1, 1024);
    end
  end
end
```

B.5 setbeamf.m

```
function setbeamf(T, fA, fB, phiA, thetaA, phiB, thetaB)

% setbeamf(tile, fA, fB, phiA, thetaA, phiB, thetaB)
%
% This function writes a specific beam to the desired ThEA tile object and selects
% this beam.
%
% parameters:
% T         ThEA tile object to which the beam should be written
% fA        frequency of the A beam. This parameter is used for loading calibration data
%           (if available) and for calculating the beam settings. It will not tune the LO
%           synthesizers to the required frequencies.
% fB        ibid. for B beam
% phiA      azimuthal angle for A beam in degrees
% thetaA    zenith angle for A beam in degrees
% phiB      azimuthal angle for B beam in degrees
% thetaB    zenith angle for B beam in degrees
%
% SJW, 2002

T = theaSetFrequency(T, fA, fB);
[T, BeamIndex] = FECAppendBeam(T, thetaA*pi/180, phiA*pi/180, thetaB*pi/180,
                               phiB*pi/180, 3.5, 3.5, ones(8), ones(8));
FECSelectBeam(T, BeamIndex);
```

B.6 startCorrelator.m

```
function startCorrelator(cfgfile)

% startCorrelator(cfgfile)
%
% This function starts the RAP correlator. This function should only be called from
% Matlab on the RAP PC.
%
% parameter:
% cfgfile   name of the configuration file (*.pmp8) which should be used as commandline
%           parameter for the RAP correlator
%
```

```
% SJW, 2002

cmd = sprintf('f:\\sjw\\pccorrelfoat\\Debug\\pccorrelfoat.exe %s &', cfgfile);
dos(cmd);
```

B.7 uvtoxy.m

```
function Gainxy = uvtoxy(u, v, signal, x, y, u0, v0, lambda)

% Gainxy = uvtoxy(u, v, signal, x, y, u0, v0, lambda)
%
% This function calculates the complex gains of the antenne elements in broad side
% based on measured or simulated data on a (u,v) grid from a single source.
%
% parameters:
% u         1-by-dimu or dimu-by-1 vector containing the u-coordinates
% v         1-by-dimv or dimv-by-1 vector containing the v-coordinates
% signal    dimu-by-dimv matrix where each entry contains the complex valued signal
%           at the corresponding (u,v) point resulting from the presence of the
%           source.
% x         1-by-dimx or dimx-by-1 vector containing the x-positions of the antenna
%           elements
% y         1-by-dimy or dimy-by-1 vector containing the y-positions of the antenna
%           elements
% u0        u-coordinate of the source
% v0        v-coordinate of the source
% lambda    measurement wavelength
%
% Note on the coordinate vectors (x, y, u, v): this function uses the same
% convention as the function AppendBeams, i.e. the x- and y-vectors define a
% regular rectangular grid in the xy-plane while the u- and v-vectors define a
% regular rectangular grid in the uv-plane.
%
% return value:
% Gainxy    the complex gains of the antenna elements in broad side direction
%
% SJW, 2002

dimu = max(size(u));
dimv = max(size(v));
dimx = max(size(x));
dimy = max(size(y));

idx_s = 1;
for idx_u = 1:dimu
  for idx_v = 1:dimv
    if (sqrt(u(idx_u)^2 + v(idx_v)^2) < 1);
      S(idx_s) = signal(idx_u, idx_v);
      idx_g = 1;
      for idx_x = 1:dimx
        for idx_y = 1:dimy
          exponential(idx_s, idx_g) =
            exp(2 * pi * i * (x(idx_x) * (v(idx_v) - v0) + y(idx_y) * (u(idx_u) - u0)) / lambda);
          idx_g = idx_g + 1;
        end
      end
      idx_s = idx_s + 1;
    end
  end
end

G = pinv(exponential) * S';

idx_g = 1;
for idx_x = 1:dimx
  for idx_y = 1:dimy
    Gainxy(idx_x, idx_y) = G(idx_g);
    idx_g = idx_g + 1;
  end
end
```

B.8 xytouv.m

```
function Signal = xytouv(x, y, Gain, u, v, lambda)

% Signal = xytouv(x, y, Gain, u, v, lambda)
%
% This function calculates the signal on the specified (u,v) grid in based on the
% specified complex gains in the broad side direction.
%
% parameters:
% x         1-by-dimx or dimx-by-1 vector containing the x-coordinates of the
%           antenna elements
% y         1-by-dimy or dimy-by-1 vector containing the y-coordinates of the
%           antenna elements
% Gain      dimx-by-dimy matrix where each entry represents the complex valued
%           gain of the corresponding antenna element in broad side
% u         1-by-dimu or dimu-by-1 vector containing the u-coordinates of the grid
%           on which the beam pattern should be calculated
% v         1-by-dimv or dimv-by-1 vector containing the v-coordinates of the grid
%           on which the beam pattern should be calculated
% lambda    measurement wavelength
%
% Note on the coordinate vectors (x, y, u, v): this function uses the same
% convention as the function AppendBeams, i.e. the x- and y-vectors define a
% regular rectangular grid in the xy-plane while the u- and v-vectors define a
% regular rectangular grid in the uv-plane.
%
% return value:
% Signal    the complex signal as measured from a source in broad side, i.e.
%           the beam of a tile with antenna gains as specified on the specified
%           (u, v) grid.
%
% SJW, 2002
```

```
dimu = max(size(u));
dimv = max(size(v));
dimx = max(size(x));
dimy = max(size(y));

idx_g = 1;
for idx_x = 1:dimx;
  for idx_y = 1:dimy;
    G(idx_g) = Gain(idx_x, idx_y);
    idx_s = 1;
    for idx_u = 1:dimu
      for idx_v = 1:dimv
        exponential(idx_s, idx_g) =
          exp(2 * pi * i * (x(idx_x) * v(idx_v) + y(idx_y) * u(idx_u)) / lambda);
        idx_s = idx_s + 1;
      end
    end
    idx_g = idx_g + 1;
  end
end

S = exponential * G’;

idx_s = 1;
for idx_u = 1:dimu
  for idx_v = 1:dimv
    Signal(idx_u, idx_v) = S(idx_s);
    idx_s = idx_s + 1;
  end
end
```

Appendix C

C++ sources

C.1 daq.h

```
// wrapper class for easy use of TIM boards
// SJW, 2003

#ifndef _INCLUDED_DAQ_H_
#define _INCLUDED_DAQ_H_

#ifndef _INCLUDED_FFTW_H_
#include "fftw.h"
#define _INCLUDED_FFTW_H_
#endif

// timu.h should be included last due to multiple definitions in system header
// files
#ifndef _INCLUDED_TIMU_H_
#include "timu.h"
#define _INCLUDED_TIMU_H_
#endif

namespace sjw
{
    class Daq: private TIM
    {
        // data members
        unsigned long
            d_blocksize,    // size blocks in readonly mode,
                            // size of FFT otherwise
            d_nblocks,      // number of block to be read per read operation
            d_startaddr,    // internal address administration
            d_currentaddr,
            d_stopaddr,
            d_offset;       // offset of startaddress for second trace when
                            // reading in dual input mode
        bool
            d_dual;         // dual or single input mode?
        fftw_plan d_p;
        fftw_complex *d_inbuf;
        public:
            // constructor
            // parameters: tim device, block size, number of block to be read
            //             per read operation, mode and clock register settings
            Daq(char *dev, unsigned long blocksize, unsigned long nblocks,
                unsigned modereg, unsigned clockreg);

            // destructor: clean up the mess
            ~Daq() {fftw_free(d_inbuf);}

            // data acquisition operations
            // start(): starts acquisition, includes waiting on sync pulse
            //          if necessary
            void start()
            {
                unsigned int dummy = 0;
                write(d_startaddr, reinterpret_cast<void *>(&dummy), 4);
            }
            void wait()  // waits for TIM board to finish
            {
                char buf[64];
                read(d_startaddr & 0x7fffffff, buf, 64);
            }
            // read: returns number of blocks succesfully read
            // parameters:
            //    outbuf: user specified location in memory where the result
            //            should be stored. It is the responsibility of the
            //            caller to ensure that it is large enough
            //    readonly: if this boolean value is set the data will not
            //              be Fourier transformed. Not tested.
            unsigned long acquire(fftw_complex *outbuf, bool readonly);
            bool finished()
            {
                return d_currentaddr == d_stopaddr;
            }
    };
}
#endif
```

C.2 correlator.h

```
// implementation of a software correlator as object
// optimized for a system of two TIM's and two CPU's
// SJW, 2003

#ifndef _INCLUDED_CORRELATOR_H_
#define _INCLUDED_CORRELATOR_H_

#ifndef _INCLUDED_DAQ_H_
#include "daq.h"
#define _INCLUDED_DAQ_H_
```

```
#endif

#define FFTSIZE    1024L
#define MAXBLOCKS  512L

namespace sjw
{
    class Correlator
    {
        private:
            // struct providing information to a thread for reading data
            struct threaddata
            {
                Daq *tim;             // Daq object to use
                fftw_complex *outbuf; // place to store result
            };

            // struct providing information to a thread for correlation
            struct correldata
            {
                Correlator *corobj; // Correlator object containing data
                int *range;         // range of frequency channels to handle
            };

            int
                d_nchannel,         // number of correlator input channels
                d_bandwidth;        // bandwidth settings for Daq objects
            Daq **d_tim;            // available Daq objects themselves
            threaddata **d_daq;     // ibid. wrapped in a thread suitable form
            fftw_complex *d_outbuf; // output of FFT

            // final result, nchannel x nchannel x FFTSIZE
            // output as single trace of spectra corresponding to elements
            // of the correlation matrix running from left to right and from
            // top to bottom, so 11, 12, 13, 14, 21, 22, etc for nchannel = 4
            double *d_result;

        public:
            // constructor
            // parameters: number of correlator channels, bandwidth of the
            //             incoming data (for Daq object) and setting for the
            //             four msb's for the mode register (see TIM reference)
            Correlator(int nchannel, int bandwidth, unsigned blocklength);
            // destructor: clean up he mess
            ˜Correlator();

            // run the correlator
            double *run();

            double *getResult()
            {
                return d_result;
            }
        private:
            // copy constructor: prevents pointer troubles
            Correlator(Correlator const &other); // NI

            // work horses
            void correlate(unsigned nblocks, int lowidx, int highidx);
            static void *readthread(void *info);
            static void *correlthread(void *info);
    };
}
#endif
```

C.3 mpc.cc

```
// main program for THEA software correlator (MPC: MicroProcessor Correlator)
// SJW, 2003

#ifndef _INCLUDED_CORRELATOR_H_
#include "correlator.h"
#define _INCLUDED_CORRELATOR_H_
#endif

#ifndef _INCLUDED_FSTREAM_
#include <fstream>
#define _INCLUDED_FSTREAM_
#endif

#ifndef _INCLUDED_ITERATOR_
#include <iterator>
#define _INCLUDED_ITERATOR_
#endif

#ifndef _INCLUDED_STDLIB_H_
#include <stdlib.h>
#define _INCLUDED_STDLIB_H_
#endif

#ifndef _INCLUDED_IOSTREAM_
#include <iostream>
#define _INCLUDED_IOSTREAM_
#endif

#ifndef _INLUDED_STRING_
#include <string>
#define _INCLUDED_STRING
#endif

using namespace std;

int main(int argc, char **argv)
{
    // the correlator object may throw string exceptions...
    try
    {
        // check number of input arguments. If too small, display usage
        if (argc < 5)
            throw string("Too few arguments\n"
                         "usage: mpc <nchannel> <bandwidth> <datasize> "
                         "<outfile>\n"
                         "with: <nchannel>: number of antenna's\n"
                         "      <bandwidth>: bandwidth in MHz (20 or 40)\n"
                         "      <datasize>: amount of data in MB, specified "
                         "as in TIM manual\n"
                         "      <outfile>: file in which the result should be "
                         "written");
        cout << "Initializing..." << endl;
        sjw::Correlator correlator(atoi(argv[1]), atoi(argv[2]),
                                   atoi(argv[3]));
        cout << "starting measurement" << endl;
        correlator.run();
        cout << "Requesting result..." << endl;
        // correlator object manages memory
        double *result = correlator.getResult();
        // write result to file, keeping format from correlator object
        ofstream outfile(argv[4]);
        copy(result, result + atoi(argv[1]) * atoi(argv[1]) * FFTSIZE,
             ostream_iterator<double>(outfile, "\n"));
    }
    // catch exceptions generated in the process, display the problem and abort
    catch(string msg)
    {
        cout << msg << endl;
        exit(1);
    }
    return 0;
}
```