A Small Radio Telescope For The University Observatory Munich

Ein Kleines Radioteleskop Für Die Universitäts-Sternwarte München

Thomas Maletz

A Small Radio Telescope For The University Observatory Munich

Thesis
for the Bachelor of Science in Physics
at the
Ludwig–Maximilians–University (LMU) Munich

Submitted by
Thomas Maletz
born 23. December 1969, Munich

Munich, 24. October 2014

Preface and Acknowledgements

One of the world leading Faculties of Physics belongs to the **L**udwig-**M**aximilians-**U**niversity in Munich.
A part of this faculty is the University Observatory Munich (**U**niversitäts-**S**ternwarte **M**ünchen) [USM].
The USM is a teaching and research institute. The research areas cover cosmology, galaxies, stars, computational astrophysics and many others. In addition the USM is developing and operating its own optical telescope on mount Wendelstein (in the Bavarian Alps) and is building instruments for other facilities, like the ***V**ery **L**arge **T**elescope* of the **E**ropean **S**outhern **O**bservatory [ESO] in Chile.

The USM had the wish for an own ***S**mall **R**adio **T**elescope*, mainly for practical training of students. The training is intended for teaching astrophysics and the usage of a telescope as well as observing objects like our sun, the milky way or other emitters of similar radiation. Such an SRT was developed at the MIT Haystack Observatory [Hay]. A detailed description of their telescope including parts list, mounting manual, ***S**oft**w**are* etc. can be downloaded from their homepage.
The main goal of this thesis is to build a Small Radio Telescope for the USM based on the one from the Haystack Observatory.

The SRT can be seen as a chain of components:

- The Antenna,
- a ***R**adio **F**requency* Receiver,
- a PC with an ***A**nalog to **D**igital **C**onverter* and drivers,
- SW to record signals and to analyse the data and
- a controller to rotate the Antenna.

The challenges were to built the antenna and the receiver and measure their performance as far as possible (details are described later on). Further to install the PC and the ADC, necessary drivers, compile the SW or maybe even develop new tools. Here we encountered several issues like which software platforms and packages are available, compatibility problems and many others (details are mainly described in the appendix).

Another issue was that the USM instrument group has not built RF components so far. Thus almost no equipment to debug or to measure the performance of the SRT was available. Fortunately my company, INTEL (Mobile Communications), supported us within a cooperation.

In the end the ***H**ard**w**are* should be ready and some SW to receive a signal and confirm the HW.
Therefore the C-Program *sample_signal* was written to record signals with the help of the ADC and its driver. Further the MATLAB/Octave-Program *Analyse* was developed to process the measured data and visualize the results. *Analyse* controls *sample_signal* and can be seen as the high level software, with which the measurement can be accomplished. This setup might be used to develop a practical training based on the SRT. An alternative is the Haystack software that has to be compiled for our system.

A further goal of this thesis is to provide a good understanding of this instrument to physicists and students.
On one side engineers should have a certain understanding of the physics, for which they are building the instrument. Furthermore they need to understand the goals and specifications and they have to provide an explanation of the device.
On the other side physicists must have an understanding of the equipment, usage, limitations, the received data and measurement errors.
I wrote the thesis in that spirit and tried to give introductions to Radio-Frequency-Engineering, Digital Signal Processing as well as some Software. Of course all are huge disciplines that cannot be fully covered here and the interested reader must be referred to the literature. There exist a lot of good books on the different topics, many of them are listed in the bibliography. But: Not everything can be found in the books, the SRT covers many areas that are wide spread across the literature and to look up and to understand a certain topic might be difficult for a reader without a certain knowledge. Thus a further motivation was to refer to the books as little as possible, and present everything that is needed for the SRT in an understandable way.
Furthermore in the appendix I give a little overview of Linux (Ubuntu), software tools and the software that is used here inclusive source codes.
All these chapters were also written to serve as look up reference. Either for an advanced user who is looking only for specific details. But also for physicists or students who are not familiar with RF, DSP, Linux, Octave/MATLAB or immediately want to jump to Octave and the Analyse program.

Acknowledgements

First of all I want to thank my employer, INTEL, and especially my former manager Andre Hanke, for giving me the opportunity to work part time and to make my B.Sc. in Physics.
Then I want to thank Prof. Dr. Jochen Weller for recommending me to the instrument group at the USM. Here I want to thank Dr. Ulrich Hopp and Florian Lang for offering me this topic and of course my Professor, Prof. Dr. Ralf Bender.
Florian and Adolf "Ady" Karasz organized and built the parabolic reflector, Marco Häuser bought all parts from the Haystack list. Also thank you so much Ady for all the quick and precise mechanical work on the receiver, the LNA and the Feed. Arno Riffeser and Vanessa

Fahrenschon deal with the Haystack Software that rotates the antenna to the wished object in the sky, and that is used to calibrate the telescope, perform measurements etc. Thank you for supporting me with that and helping me with the antenna measurements. Further Vanessa and Arno will develop a radio astronomy training for students based on the SRT.
I want to thank my mentor Florian for always being so open and supportive, as was the whole team at the USM.
Another big 'Thank You' goes to Prof. Dr. Warren J. Jasper (at the North Carolina State University) for his excellent support and all the time he spent for us: Warren developed the Linux-driver for the ADC and was of immediate, comprehensive and important help concerning problems, providing driver updates and so much more.
I also want to thank KR Electronics and Charles Kiall for their quick and great service and developing the filter for us. Similar I want to thank both, Mrs. Haimerl with Rechinger / Deko Figuren Shop and Mr. Ziemer with Sansys for producing the rods for holding the helix antenna.
We also want to thank INTEL and especially Line Manager Boris Kapfelsperger for the cooperation and to let me use their lab and instruments for measuring.
Further I want to thank Rohde & Schwarz and especially Harald Ripp: They provided us with a Signal Generator and a Spectrum Analyzer for verification at the USM.
Of course I also want to thank the MIT Haystack Observatory for developing the SRT and offering a detailed description, and especially Alan Rogers for all his tips.
And I want to thank my wife Bing: for everything.

Unterhaching, Summer 2014, Thomas Maletz (mailto:thomas.maletz@maletz.de)

Contents

Preface and Acknowledgements . . . iii

Contents . . . vii

List of Figures . . . ix

1 Radio Astronomy . . . 1

2 Introduction to Radio-Frequency Engineering . . . 7
2.1 Basics and Terms . . . 7
2.1.1 Electronics engineering terms . . . 7
2.1.2 Symbols in electrical and electronics engineering . . . 8
2.1.3 Ground . . . 9
2.1.4 Decibel . . . 9
2.2 Time and Frequency Domain . . . 9
2.3 Connectors, Shielding, Cables . . . 10
2.4 Complex description (of AC circuits) . . . 11
2.5 Capacitors and Inductors . . . 18
2.6 Noise . . . 19
2.6.1 Basics . . . 19
2.6.2 Noise Measurements . . . 21
2.7 Filters, Decoupling, Blocking, Bias Tee . . . 23
2.7.1 Filtering . . . 23
2.7.2 Bias Tees . . . 24
2.8 Electromagnetic Waves . . . 25
2.8.1 Transmission Lines . . . 25
2.8.2 Power Splitting / Combining . . . 28
2.9 Scattering-Parameters . . . 29
2.10 Linearity, Dynamic Range . . . 30
2.11 Synthesizer . . . 31
2.12 Mixers and Mixing . . . 33
2.13 Amplifiers . . . 36

3 Small Radio Telescope: Introduction and Overview . . . 39

4 The Antenna . . . 41
4.1 Antenna Basics . . . 42
4.1.1 Field Zones . . . 42

4.1.2 Power and Impedance . . . 42
4.1.3 Radiation Pattern, Directivity, Gain . . . 43
4.1.4 Reciprocity Theorem . . . 44
4.2 The SRT Antenna . . . 44
4.2.1 The Reflector . . . 45
4.2.2 The Feed . . . 45
Assembly and Tuning . . . 48
Matching and Measurement Results . . . 52
Feed and LNA . . . 54
The Antenna . . . 56

5 The Receiver . . . 59
5.1 The Design . . . 59
5.2 Performance Measurements . . . 63
5.2.1 Low Noise Amplifier . . . 64
5.2.2 Receiver without Low Noise Amplifier . . . 67
5.2.3 The Receiver Chain (RX) . . . 69
5.2.4 Conclusions . . . 74
5.2.5 Time Domain Measurements . . . 74
5.3 Customized Filter Performance Plots . . . 77

6 Digital Signal Processing . . . 79
6.1 Introduction . . . 79
6.2 Fourier Transformation . . . 80
6.3 Sampling . . . 89
6.4 DFT and arbitrary signals . . . 91
6.5 Analog Digital Converter . . . 93
6.6 DSP of the SRT . . . 94

7 The complete SRT . . . 99
7.1 Results . . . 99
7.2 Outlook . . . 103

A Software Tools . . . 105
A.1 Operating System . . . 105
A.2 The C Programming Language . . . 106
A.3 Mathematics Tools and Lab Software . . . 106

B Ubuntu and Octave . . . 109

C Ubuntu and the ADC PCI-Card . . . 115

D Software and Source Code . . . 117
D.1 C Program . . . 117
D.2 Octave (MATLAB) . . . 120

Bibliography . . . 125

Deklaration . . . 129

List of Figures

1.1 Electromagnetic Spectrum 1
1.2 21 cm line: Electron Spinflip 2
1.3 Planck Satellite in front of CMB (courtesy ESA). 4
1.4 Effelsberg radio telescope (courtesy [Wiki] and Dr. Schorsch). 5

2.1 Symbols in electrical and electronics engineering 8
2.2 Typical Coaxial Cable [Wiki] 11
2.3 SMA Connectors 11
2.4 N Connectors 12
2.5 BNC Connectors 12
2.6 Kirchhoff's Law 13
2.7 Examples 15
2.8 Abs(Z_{ser}) 15
2.9 Abs(Z_{par}) 16
2.10 Capacitor Model 19
2.11 Inductor Model 19
2.12 Noise Temperature Model 21
2.13 Noise Measurement: Setup 21
2.14 Noise Measurement: Result 22
2.15 Low and High Pass Filter 23
2.16 Band Definitions (Example Low Pass Filter) 24
2.17 Bias Tee 25
2.18 Transmission Line Model 26
2.19 Smith Chart 27
2.20 Two Port 29
2.21 Exponential Function, Diode 30
2.22 3 dB Compression Point 31
2.23 Phase Locked Loop 32
2.24 PLL Phase Noise 33
2.25 Frequency Mixing 34
2.26 Simple Image Canceling Mixer 35
2.27 Diode, Bipolar and Field Effect Transistor 36

3.1 Small Radio Telescope 39

4.1 Radiation Pattern 43
4.2 Helical Antenna with N windings 45
4.3 Feed Conductor 49

4.4 Feed . . . 51
4.5 Feed . . . 51
4.6 Feed S_{11} . . . 52
4.7 Feed with LNA and dish . . . 54
4.8 Feed with LNA and dish . . . 55
4.9 Feed with LNA . . . 55
4.10 The SRT Antenna . . . 56
4.11 The SRT Antenna . . . 56
4.12 The SRT Antenna . . . 57

5.1 Electrical Schematic . . . 59
5.2 Low Noise Amplifier Stage . . . 62
5.3 The Receiver . . . 62
5.4 General Measurement Setup . . . 63
5.5 LNA Measurement Setup . . . 64
5.6 LNA Frequency Response . . . 65
5.7 LNA Compression Point . . . 66
5.8 Receiver without LNA: Measurement Setup . . . 67
5.9 Receiver Frequency Response . . . 68
5.10 Receiver Compression Point . . . 68
5.11 RX Frequency Response . . . 69
5.12 RX Frequency Response 1417.4 MHz . . . 70
5.13 RX Frequency Response 1423.4 MHz . . . 70
5.14 RX Image Rejection . . . 71
5.15 RX frequency response with lower input power . . . 72
5.16 RX Compression Point . . . 72
5.17 RX system noise . . . 73
5.18 RX Time Domain . . . 75
5.19 RX Time Domain with reduced input power . . . 75
5.20 RX Time Domain with increased input power . . . 76
5.21 LPF Frequency Response In Band . . . 77
5.22 LPF Frequency Response Wide Band . . . 77

6.1 Sample and Hold . . . 79
6.2 Complex Circle . . . 82
6.3 Dirac Pulses and their Fourier Transformation . . . 85
6.4 Rectangular Pulse and its Fourier Transformation . . . 85
6.5 Signal with 3 harmonics . . . 86
6.6 Signal with 3 harmonics . . . 87
6.7 Angles of the complex coefficients. . . . 88
6.8 Single Side Band Spectrum of Signal . . . 89
6.9 Analog to digital mapping . . . 89
6.10 Quantization Error . . . 90
6.11 Sampling and Aliasing . . . 91
6.12 DFT of a sinusoidal signal . . . 92
6.13 Hanning Window . . . 93
6.14 Signal Generator: Frequency . . . 94
6.15 Signal Generator: Amplitude . . . 95

6.16 Oscilloscope verifying Signal Generator 95
6.17 Calculated spectrum of test signal 96
6.18 Test signal spectrum in dBm 97

7.1 Power cold sky 100
7.2 Power sun 101
7.3 Sun measured with the Haystack Software 102

B.1 QtOctave 110
B.2 QtOctave Menubar 111
B.3 File Manager: Home 113
B.4 File Manager: Praktikum 113

Chapter 1

Radio Astronomy

Figure 1.1 shows an overview of the electromagnetic spectral range that is explored and used today.

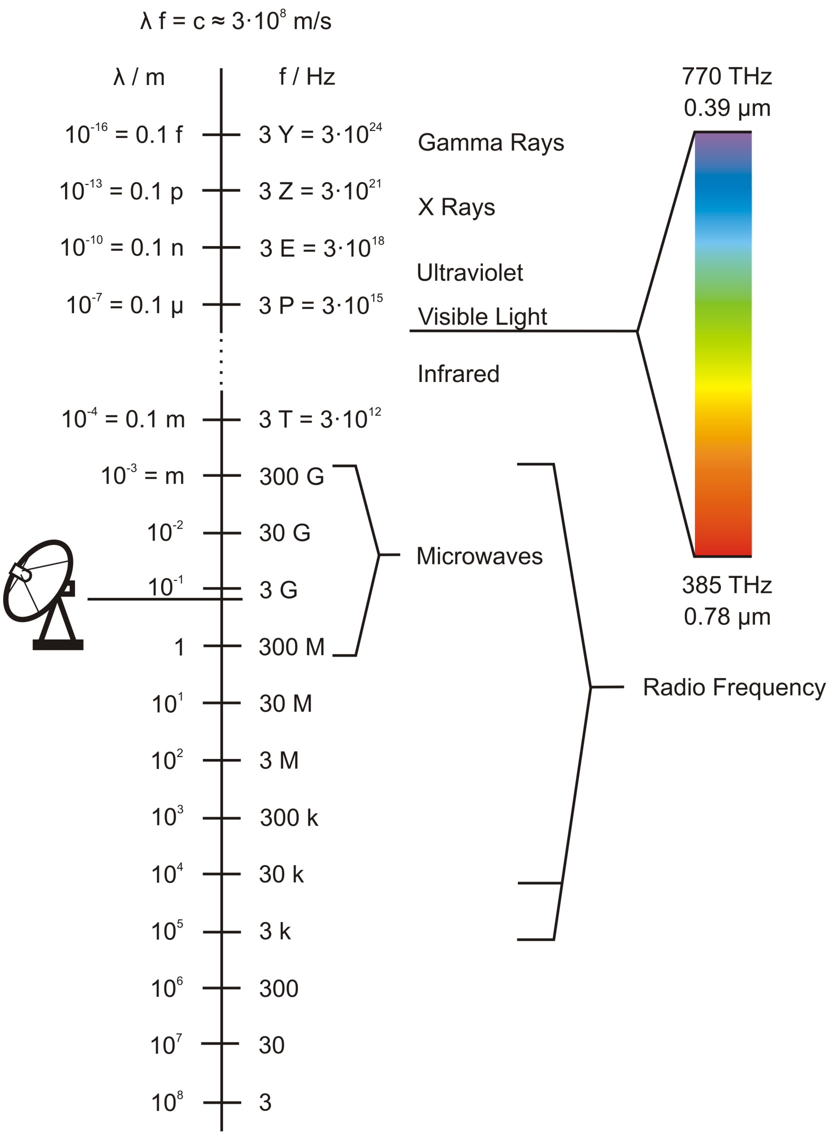

Figure 1.1: Electromagnetic Spectrum

Astronomy started, of course, in the optical spectral range. Already ancient cultures observed the sky to predict seasons and the best time for sowing. Humankind started to use stars for navigation, especially at sea. And we have an inherent curiosity and drive to explore the universe, often also connected to religion and philosophy.
Probably the first time that a telescope was used in astronomy was in the 17. century, when Galileo Galilei used his self-made optical telescope to observe moons, planets and the milky way. Today we not only observe radiation, but even use electromagnetic waves in communication, medicine or devices like the microwave oven.

Radio astronomy started early in the twentieth century, when communication engineers detected unknown noise with their antennas. First investigations by Karl Jansky at the AT&T Bell Labs in 1931 identified the milky way as the source of the radiation.
A similar story happened later on in 1964 to Arno Penzias and Robert Wilson, who also found noise in their system. Robert Henry Dicke et al. identified this noise as **C**osmic **M**icrowave **B**ackground, which is a result of the big bang. Arno Penzias and Robert Wilson got the Nobel Prize in 1978.

Today in radio astronomy all kind of astrophysical objects are observed. Among them are stars, the radiation of charged particles in magnetic fields, **A**ctive **G**alactic **N**uclei, spectrum of molecules, gases, nebulae, the HI-Line or red shifted objects.
Since the universe is expanding the wavelength of radiation is becoming longer as well, while the radiation is traveling through space. This is called *red shift*, a shortening of the wavelength is called *blue shift*. The longer the travel lasts the more the wavelength gets expanded. Thus, emissions could have started at a much higher frequency and energy, but is falling into the RF range today.
The *HI-line* (*I* meaning one), also called *21 cm line*, is emitted by (neutral) hydrogen as illustrated in figure 1.2.

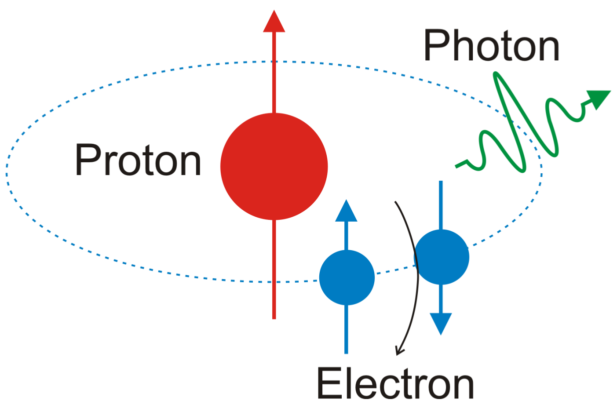

Figure 1.2: 21 cm line: Electron Spinflip

Hydrogen is the simplest atom and consists of only one proton and one electron.
The Schrödinger equation shows that the electron can have different states, where every state means a certain energy level. They are numbered with $n = 1, 2, \ldots$, where one is the lowest energy level.
Proton and electron have both a magnetic moment which is explained by an intrinsic spin S.

The electron in the hydrogen Atom has an additional magnetic moment due to its orbit around the proton, described with an angular momentum L. Note that the electron does not circle around the proton as for example the earth around the sun. Thus *orbit* should be seen here as a more general term.
The spin and the orbit angular momentum interact with each other and sum up to a total angular momentum of the electron $J = L + S$. This is called spin-orbit interaction. The magnetic moments add like the angular momenta. For a fixed energy level n and a fixed angular momentum L the different orientations of the spin S with respect to L results in slightly different energy levels. This interaction slightly splits the energy levels n and creates a so called *fine structure*.
Similar the magnetic moments of electron and proton add to a magnetic momentum of the hydrogen atom. The two magnetic moments can have different orientations that again results in different energy levels. These energy differences are even smaller than due to the spin-orbit interaction and the result is called *hyperfine structure*.
In the $1s_{1/2}$ ground state the electron occupies the lowest energy level $n = 1$, its angular momentum is zero and its magnetic moment is just its spin. The proton and electron spin can be either parallel or anti-parallel, which results in a slightly higher or lower energy level, with a difference of $\Delta E \approx 5.9 \cdot 10^{-6}$ eV. The spin of the electron could flip and the energy is emitted by a photon. Such a spin flip is forbidden for dipole radiation and would require multipole radiation. Therefore an excited electron has an average lifetime of 11 million years [UnBa]. This spin flip is hardly observed in the laboratory but happens more often, if enough hydrogen is available and collisions of atoms support the (de-)excitation.
Energy, frequency and wavelength of a photon and the emitted radiation are related as

$$\Delta E = E_{Photon} = h\nu = \frac{hc}{\lambda}$$

where c is the speed of light, ν and λ are frequency and wavelength of the photon.
The following values have been measured for the HI-line ([Foot] and [CaOs]):

$$\nu = 1420.40575177\,\text{MHz}$$

$$\lambda = 0.211\,\text{m}$$

The spin flip was first predicted by Hendrik Christoffel van de Hulst in 1944.
More than 70% of the atoms in the universe are hydrogen, and most astrophysical objects contain a large amount of hydrogen. Therefore many objects sufficiently emit for being detectable. Furthermore the unique 21 cm line can identify objects as hydrogen.

The ***S****mall* ***R****adio* ***T****elescope* is designed to measure around 1420.4 MHz.

Today both, earthbound telescopes as well as satellites are used in radio astronomy. For example the Planck-Satellite just finished measuring the CMB in the frequency range from 30 GHz to 857 GHz with an angular resolution between 5 and 33 arcmin. The CMB is a black body radiation with a temperature of $T = 2.7255 \pm 0.0005\,\text{K}$, in figure 1.3 visualized with colours.

One of the largest and fully steerable radio telescopes in the world is the 100 m telescope in Effelsberg (Germany), figure 1.4.

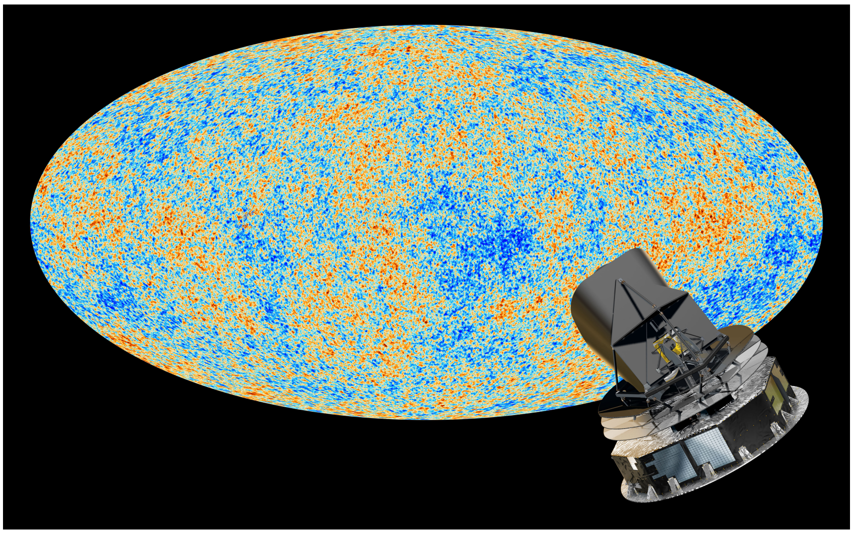

Figure 1.3: Planck Satellite in front of CMB (courtesy ESA).

Many other telescopes were developed, among them are the fixed 305 m Arecibo telescope in Puerto Rico, the James Clerk Maxwell Telescope in Hawaii or the *Very Large Array* in New Mexico that comprises 27 movable antennas.
In the universe large dust clouds often absorb light, but are transparent for radio waves. Thus, with the help of radio astronomy, objects can be observed that are not visible in the optical range.

The atmosphere of the earth significantly attenuates radiation with a wavelength of $\lambda < 5\,\mathrm{mm}$, equivalent to a frequency $f > 60\,\mathrm{GHz}$. Radiation with a wavelength $\lambda > 30\,\mathrm{m}$, that is a frequency $f < 10\,\mathrm{MHz}$, is reflected by the Ionosphere. These are roughly the boundaries for radio astronomy [Schneid].

The angular resolution θ of a telescope depends on its size D as well as the wavelength λ as [CaOs]

$$\sin\theta \propto \frac{\lambda}{D}$$

In radio astronomy the observed wavelengths are long. Thus the angular resolution of one radio telescope is fairly limited.
To achieve higher angular resolutions several telescopes can be combined to an interferometer, where D becomes the distance between them. The different telescopes are measuring the same source. The measured signals have to be added in-phase. Hence the big challenge is to synchronize the telescopes.
Beginning in 1970 the ***V**ery **L**ong **B**aseline **I**nterferometry* was developed. Here radio telescopes can be even distributed around the world. To synchronize them all signals are recorded together with precise timestamps. VLBI achieves angular resolutions in the range of 0.0001 arc seconds [UnBa].

Another important effect is the well known *relativistic Doppler shift*. If a source of an electromagnetic wave with frequency ν_0 moves relative to an observer then the observers sees the radiation with a different frequency ν. Let v be the relative velocity.

Figure 1.4: Effelsberg radio telescope (courtesy [Wiki] and Dr. Schorsch).

Einstein found as relationship [TiMo]

$$\nu = \sqrt{\frac{1 - v/c}{1 + v/c}}\nu_0 \tag{1.1}$$

It was desired to detect a Doppler shift for a relative velocity of $v = \pm 10\,\mathrm{km/s}$. For $\nu_0 = 1420.4\,\mathrm{MHz}$ the detected frequencies are

$$\nu = \sqrt{\frac{1 - 10 \cdot 10^3/3 \cdot 10^8}{1 + 10 \cdot 10^3/3 \cdot 10^8}} \cdot 1420.4\,\mathrm{MHz} \approx 1420.353\,\mathrm{MHz}$$

$$\Delta\nu = \nu - \nu_0 \approx -47\,\mathrm{kHz}$$

and

$$\nu = \sqrt{\frac{1 + 10 \cdot 10^3/3 \cdot 10^8}{1 - 10 \cdot 10^3/3 \cdot 10^8}} \cdot 1420.4\,\mathrm{MHz} \approx 1420.447\,\mathrm{MHz}$$

$$\Delta\nu = \nu - \nu_0 \approx 47\,\text{kHz}$$

The spectral resolution of the receiver is approximately 305 Hz (please see chapter 6 for details). This would be the Doppler shift of objects with a relative velocity $v = \pm 65$ m/s. The receiver can measure signals approximately between 1417.4 MHz and 1423.4 Mhz, in other words in a $\Delta\nu = \pm 3$ MHz band around 1420.4 MHz (please see chapter 5 for details). Thus, according to equation 1.1, objects with a maximal relative velocity $v = \pm 634.3$ km/s can be detected.

Carroll and Ostlie [CaOs], Unsöld and Baschek [UnBa] as well as Schneider [Schneid] describe in their books the kinematics of the milky way. We observe the universe either from earth or with the help of satellites. On the scale of the milky way these positions are identical to the location of the sun. Thus all measurement results are relative to the velocity and position of our star. Therefore it is useful to introduce the ***L****ocal* ***S****tandard of* ***R****est*: the LSR is a point that is today located in the centre of our sun and rotating around the centre of the milky way on a perfect circle. The LSR velocity is $v = 220$ km/s. This is derived by comparison with halo stars while it is assumed that the halo is (in average) not rotating. The absolute velocity of the sun is almost identical to the LSR velocity. Other stars were observed with velocities relative to the LSR between approximately 65 km/s $\ldots - 250$ km/s.

Chapter 2

Introduction to Radio-Frequency Engineering

2.1 Basics and Terms

2.1.1 Electronics engineering terms

AC, DC	Alternating Current, Direct Current (used as indicators)
U, u	DC, AC Voltage
I, i	DC, AC Current
$\underline{x}$	Complex voltage or current
R	(Electrical) Resistance
Z	Complex Resistance, *Impedance*
G, Y	1/R, 1/Z: *Conductance, Admittance*
C	Capacitance
L	Inductance
P	Power
j	$j^2 := -1$
f	Frequency
ω	$2\pi f$
t	Time
$\log_{10}$	lg
$\log_e$	ln
DUT	Device Under Test: The device that is tested
CW	Continuous Wave: sinusoidal signal
RMS	Root Mean Square $\sqrt{1/N \sum_{i=1}^{N} x_i^2}$

2.1.2 Symbols in electrical and electronics engineering

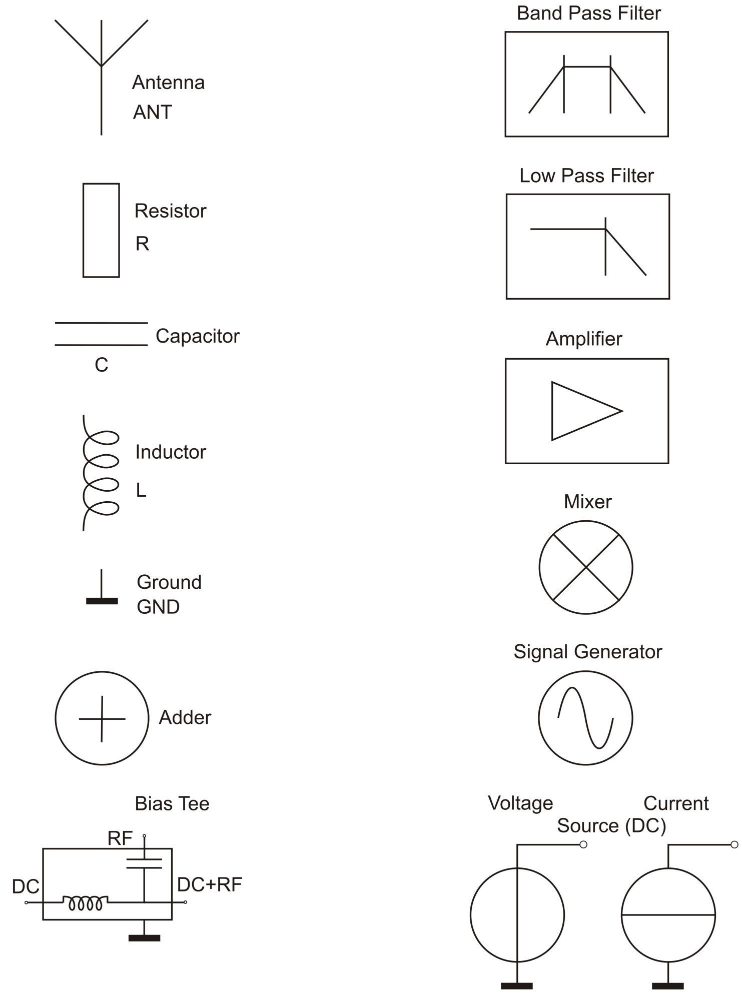

Figure 2.1: Symbols in electrical and electronics engineering

2.1.3 Ground

Usually in electronic circuits a so called ground (*GND*) is introduced. GND is serving as an electric potential, to which all other potentials are referenced to. Potential differences are voltages. Often GND is simply minus. In more complex circuitries GND is zero with plus / minus Voltages relative to it. Most ***P****rinted* ***C****ircuit* ***B****oards* consist of different layers, where one or more are almost pure metal and serve as GND layer – and also shielding. Often GND is connected to earth, further shieldings etc.

2.1.4 Decibel

A very popular unit is Decibel, dB. Basically dB are logarithmed percentages.

We can divide physical values in

- field-like values, for example current, voltage, fields, pressure, ... and
- power-like values, for example electrical power, ...

They depend on each other like power = field^2, for instance $P = UI = U^2/R = I^2R$. We further remember that for the logarithm holds $\log x^2 = 2\,\log x$. This is reflected in the

Definition of Decibel

- For field-like values:
  $x\,\mathrm{dB} = 20 \cdot \log_{10}(\frac{x}{x_{ref}})$
- For power-like values:
  $x\,\mathrm{dB} = 10 \cdot \log_{10}(\frac{x}{x_{ref}})$

where x_{ref} is the reference value; $100 \cdot \frac{x}{x_{ref}}$ would be percentages.
The reference value can be a certain value. Some are very popular and the reference is reflected in the unit name, for example for $x_{ref} = 1\,mW$ the unit is called dBm (e.g. $10\,\mathrm{mW}$ are $10\ \lg(\frac{10\mathrm{mW}}{1\mathrm{mW}}) = 10\,\mathrm{dBm}$).
The reference value can also be of more general nature, example: $10\ \lg 0.5 \approx -3\,\mathrm{dBm}$. Thus, a circuit that has a 3 dB attenuation would attenuate every input power by factor 2 (here the reference is the input power).
Often the power of a given carrier serves as reference. A *carrier* is a signal that is used to carry information. In this case we call the unit dBc.

2.2 Time and Frequency Domain

Often signals are given versus time, for example a sequence of Bits, voice etc. This point of view is called *Time Domain*. But with the help of the Fourier Analysis we can transform it into the *Frequency Domain*.

Many signals in nature radiate or absorb at certain frequencies.

Also in communication all signals only use a limited *band*, for example voice uses approximately 80 Hz - 12 kHz.
Transmission paths like air, cables etc. are linear. This offers the possibility to use different frequency bands in parallel for different communications: the signals can be combined and the receiver can separate them by filtering.

Figure 1.1 shows an overview of the spectral range that we observe or even use for communication today.
The *Radio Frequency* range is used for various applications like radio (below 1 Ghz), mobile phones (around 1 GHz and 2 Ghz), WLAN (approximately 2 Ghz and 5 Ghz), satellite communication/broadcasting and radar (1-40 GHz) and many others.

The band we want to observe is around the 21 cm line, that is around 1420.4 MHz.

Many circuits and transmission paths have a behaviour that depends on the frequency. This is called *frequency response*. An example is a ***L****ow* ***P****ass* ***F****ilter*, which lets through signals with low frequencies but attenuates the high frequency content of a signal.

One of the most important tasks in electronics engineering is to measure in the frequency domain. For this purpose so called *Spectrum Analyzers* were developed. They either measure in the time domain (i.e. record the signal versus time) and calculate the Fourier Transformation; or they measure directly in the frequency domain by measuring the power at certain frequencies, while automatically sweeping the frequency. A Spectrum Analyzer cannot measure the power of one single frequency. The analyzer applies a filter to the signal and can sweep the filter versus frequency [1]. Like a window moving along the frequency axis, for which the whole power is measured. Spectrum Analyzers offer several values for the bandwidth of the filter, termed ***R****esolution* ***B****and* ***W****idth*. For example if we chose a RBW of 100 kHz, and the Spectrum Analyzer measures and shows a power p at frequency f, then actually p is the whole signal content of a 100 kHz band around f!

Also modern *Oscilloscopes* can calculate a Fourier Transformation.

2.3 Connectors, Shielding, Cables

To transmit RF-signals most often coaxial cables are used. *Coax cables* consist of a conductor, the core, surrounded by a metal shielding. The shielding is normally connected to GND.
The current on the core flows in one direction, while the current on the shield is the opposite. Thus the magnetic fields of the two currents add up between core and shield, but cancel each other outside the cable. Similar the electric field goes only between core and shield. Thus waves are guided inside the coax cable, but the cable does not radiate and provides a good decoupling between signal and environment.

[1] Actually the real implementation is that the filter together with amplifiers work at a fixed frequency band. The Spectrum Analyzer mixes the signal (moving it in the frequency domain) so that a certain frequency band falls into the filter. For mixing please see the following chapters.

In the higher microwave range, if the wavelength becomes similar or shorter than the radius of coax cables, better hollow metal waveguides are used.

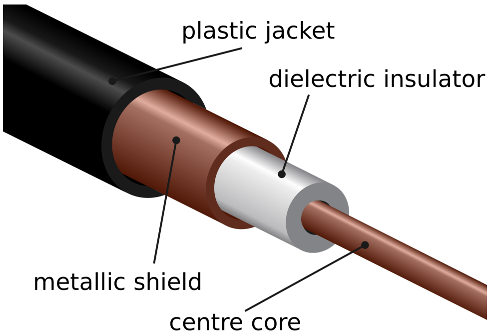

Figure 2.2: Typical Coaxial Cable [Wiki]

The most popular connectors in the GHz frequency range are *SMA*, figure 2.3. Instruments mostly use *N* connectors, figure 2.4. For lower frequencies typically *BNC* connectors are used, figure 2.5.

Figure 2.3: SMA Connectors

The connectors provide the coax principle with a core and a shielding, which serves as mechanical screw (thread) at the same time. The conductor is implemented as sleeve (*female*) or pin (*male*).

In RF shieldings are very important, usually by enclosing circuits in metal cases. Shieldings decouple the circuit from the environment.

2.4 Complex description (of AC circuits)

This chapters gives an introduction to the description of AC circuits with the help of complex calculus and presumes sinusoidal signals and settled states (this is no voltage jumps or similar).

To the foundation of circuit analysis belong Kirchhoff's two famous laws.

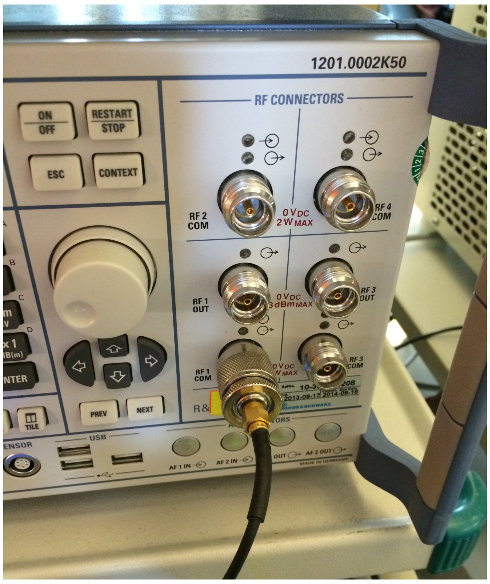

Figure 2.4: N Connectors

Figure 2.5: BNC Connectors

An electronic circuit consists of electronic components, which are connected at *nodes*. Every node has its own potential, the potential difference between two nodes is the voltage U. A loop is a closed path made up by several nodes, for example if a circuit has ten nodes, then a loop would be node 2, node 3, node 6, node 2. This is illustrated in figure 2.6.

The current law says that the sum of all currents in one node is zero; in other words the sum of all currents that flow into the node is equal to the sum of all currents that flow out of the node. The physical meaning is that charges are conserved. In our example we have $I_1 + I_2 = I_3 + I_4$.

The voltage law says that the sum of the voltages along one loop is zero. The physical meaning is that if we walk in a circuit on a circle and sum the potential differences we must get zero, i.e. we must come to the same potential if we come to the same node. It is also very useful to think in the following way: if you go from node A to node B in two different ways you should get the same voltage. In our example $U_1 + U_2 = U_3$.

Now we start with the complex modelling.

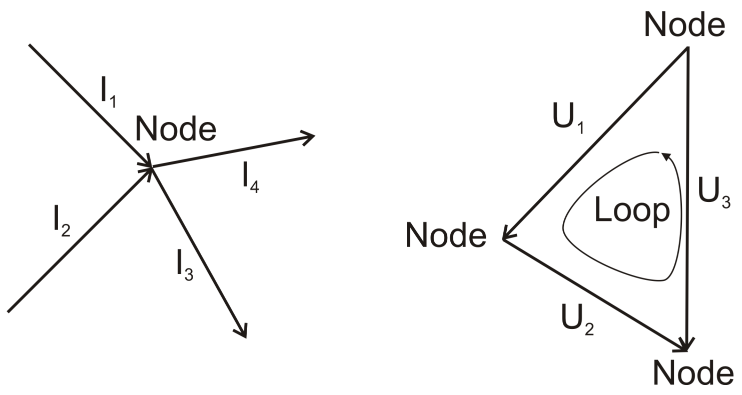

Figure 2.6: Kirchhoff's Law

First we can introduce a complex description of a sinusoidal signal:

$$a(t) = A\sin(\omega t) \rightarrow \underline{a}(t) = A\exp^{j\omega t} \tag{2.1}$$

We also keep in mind

$$A\sin(\omega t) = \frac{1}{2}(A\exp^{j\omega t} - A\exp^{-j\omega t}) \tag{2.2}$$

$$A\cos(\omega t) = \frac{1}{2}(A\exp^{j\omega t} + A\exp^{-j\omega t}) \tag{2.3}$$

Next we model capacitors, inductors and resistors in a similar way. These are the most popular elements, and they can often be used to model other devices.
First we remember the following relationships, which are actually differential equations:

Resistor

$$u = Ri$$

A solution would be $i = I\sin(\omega t)$, $u = RI\sin(\omega t)$.

Capacitor

$$i = C\frac{du}{dt}$$

A solution would be $u = U\sin(\omega t)$, $i = CU\cos(\omega t)$.

Inductor

$$u = L\frac{di}{dt}$$

A solution would be $i = I\sin(\omega t)$, $u = LI\cos(\omega t)$.

We can see for capacitors and inductors that there is a $\pm 90°$ phase shift between voltage and current.
Furthermore we can charge the capacitor / inductor, the energy is stored in the electric / magnetic field, later on we can win back this energy.

Now let's see how we can model them complex:

If we plug in the complex form of the voltage / current according 2.1 into the description of our devices we have

for the Capacitor

$$\underline{i} = C\frac{d}{dt}(U\exp^{j\omega t}) = j\omega UC\exp^{j\omega t} = \underline{I}\exp^{j\omega t}$$

and for the Inductor

$$\underline{u} = L\frac{d}{dt}(I\exp^{j\omega t}) = j\omega IL\exp^{j\omega t} = \underline{U}\exp^{j\omega t}$$

Now we can further calculate the real part or we could have used 2.2 / 2.3 instead from the beginning, since the derivation is a linear operation. It is easy to see that both give the same, correct result.
Furthermore we recognize that we can cancel $\exp^{j\omega t}$ and that we can achieve the following

Complex Models for the impedance $Z = \frac{u}{i}$ for

R	**C**	**L**
$Z = R$	$Z = \frac{1}{j\omega C}$	$Z = j\omega L$

Only if we need to see our signal over time we multiply our results with $\exp^{j\omega t}$ and calculate the real part.

A first and easy to see result of our models together with the Kirchhoff's law is:

If two components are connected serially then the resulting impedance is the sum of the individual impedances.

If two components are connected in parallel then the resulting admittance is the sum of the individual admittances.

Now let's get more familiar with our model and work out the two examples of picture 2.7.

For a) we simply have

$$\begin{aligned} Z &= R + j\omega L + \frac{1}{j\omega C} = \\ &= \frac{j\omega RC - \omega^2 LC + 1}{j\omega C} \end{aligned}$$

We can easily see from the equation, that

- $|Z| \to \infty$ for $\omega \to 0$
- $|Z| \to \infty$ for $\omega \to \infty$

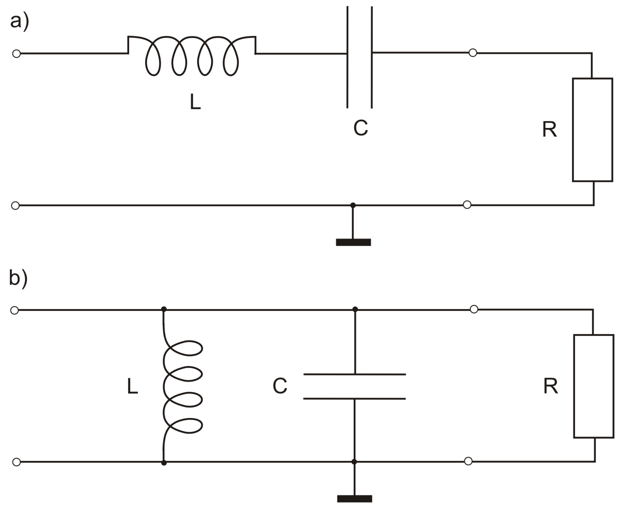

Figure 2.7: Examples

- $|Z|$ is minimum and equals R, if
  $j\omega L + \frac{1}{j\omega C} = 0 \Leftrightarrow j\omega L = \frac{j}{\omega C} \Leftrightarrow \omega^2 = \frac{1}{LC} \Leftrightarrow$
  $f = \frac{1}{2\pi\sqrt{LC}}$

Where $f = \frac{1}{2\pi\sqrt{LC}}$ is the resonant frequency of a series LC circuit, at which the circuit behaves like a short.

Figure 2.8, calculated with MATLAB, shows our results.

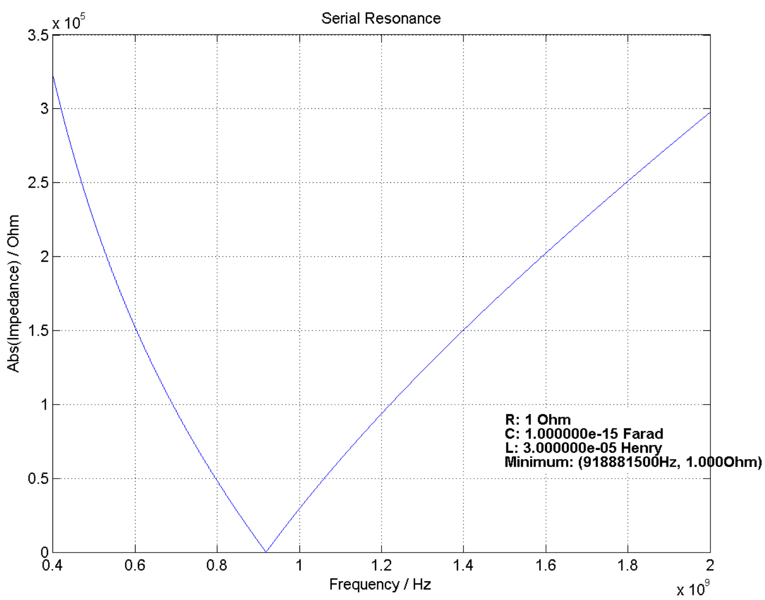

Figure 2.8: Abs(Z_{ser})

Now let's calculate b):

$$Y = G + j\omega C + \frac{1}{j\omega L} = \frac{j\omega GL - \omega^2 LC + 1}{j\omega L}$$

$$Z = \frac{1}{Y} = \frac{j\omega L}{j\omega GL - \omega^2 LC + 1}$$

We can see easily see from the equation, that

- $|Z| \to 0$ for $\omega \to 0$
- $|Z| \to 0$ for $\omega \to \infty$
- $|Z|$ is maximum and equals R, if $G = 1/R$ is minimal:
  $j\omega C + \frac{1}{j\omega L} = 0 \Leftrightarrow j\omega C = \frac{j}{\omega L} \Leftrightarrow \omega^2 = \frac{1}{LC} \Leftrightarrow$
  $f = \frac{1}{2\pi\sqrt{LC}}$

Where $f = \frac{1}{2\pi\sqrt{LC}}$ is the resonant frequency of the parallel LC circuit, at which the circuit behaves like an open.

Figure 2.9, calculated with MATLAB, shows our results.

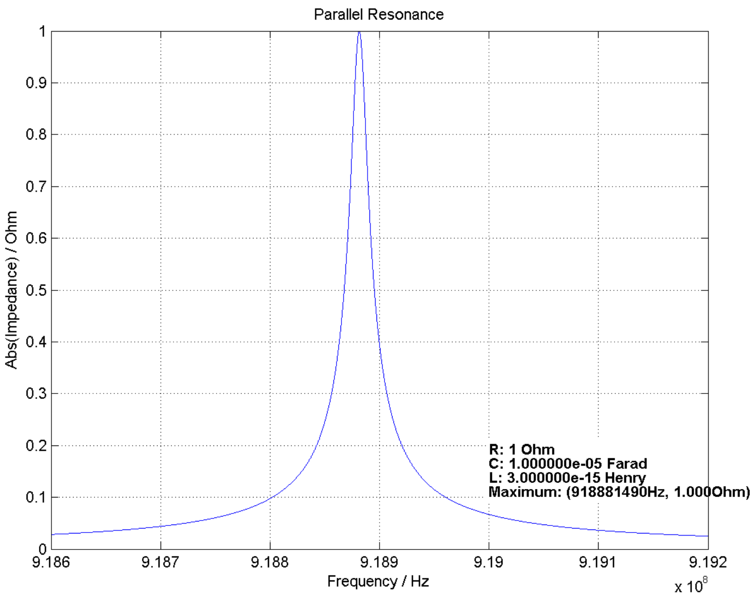

Figure 2.9: Abs(Z_{par})

Calculating Power

We can also use the complex description to calculate the power.
We assume a sinusoidal voltage and current with a relative phase shift ϕ:

$$u(t) = U\sin(\omega t + \phi)$$
$$i(t) = I\sin(\omega t)$$

The power is given by $u(t)i(t)$ and with the well known relationships

$$\sin\alpha\sin\beta = \frac{1}{2}(\cos(\alpha-\beta) - \cos(\alpha+\beta))$$

we achieve

$$p(t) = u(t)i(t) = \frac{1}{2}UI(\cos(\phi) - \cos(2\omega t + \phi)).$$

The average dissipated power is therefore [2]

$$p = \frac{1}{2}UI\cos(\phi).$$

We already saw for the solution of capacitors and inductors that $\phi = \pm 90°$, thus the dissipated power is zero – we can win back the stored energy. For a resistor holds $\phi = 0$.

Now we describe the power with the help of a complex model. Therefore we take the complex values, multiply them with $\exp^{j\omega t}$ and calculate the real part:

$$u(t) = \frac{1}{2}(\underline{U}\exp^{j\omega t} + \underline{U}^*\exp^{-j\omega t})$$
$$i(t) = \frac{1}{2}(\underline{I}\exp^{j\omega t} + \underline{I}^*\exp^{-j\omega t})$$

and

$$\begin{aligned} p(t) &= \frac{1}{4}(\underline{UI}\exp^{j2\omega t} + \underline{UI}^* + \underline{U}^*\underline{I} + \underline{U}^*\underline{I}^*\exp^{-j2\omega t}) \\ &= \frac{1}{2}\mathrm{Re}\{\underline{UI}\exp^{j2\omega t} + \underline{UI}^*\}. \end{aligned}$$

And we achieve like before

$$p = \frac{1}{2}\mathrm{Re}\{\underline{UI}^*\}.$$

It is popular to define *effective* voltages and currents: $U_{eff} := \frac{U}{\sqrt{2}}$, $I_{eff} := \frac{I}{\sqrt{2}}$.
Thus we finally have

$$p = \mathrm{Re}\{\underline{U}_{eff}\underline{I}^*_{eff}\}.$$

For a resistor ϕ is zero and

$$p = \frac{1}{2}\mathrm{Re}\{\underline{UU}^*/R\} = \frac{1}{2}\mathrm{Re}\{R\underline{II}^*\} = \frac{1}{2}\hat{U}^2/R = \frac{1}{2}R\cdot\hat{I}^2 = U_{eff}^2/R = R\cdot I_{eff}^2$$

where $\hat{U}$ and $\hat{I}$ are the amplitudes of the sinusoidal signals. This looks like the equation to calculate the power that the resistor R dissipates, when a DC voltage U is applied, resulting in a DC current I: $P = U\cdot I = U^2/R = R\cdot I^2$. Thus in both cases, for an AC signal with effective values U and I or for a DC signal with U and I, the same power is dissipated.

[2] $\frac{1}{T}\int_0^T \sin(2\omega t + \phi) = 0$

2.5 Capacitors and Inductors

Capacitors and inductors belong together with resistors to the most fundamental elements. Very often other components can be modelled as a LRC combination. For example a semiconductor, which is operating in a linear range, is a simple resistor – maybe together with a capacitor to simulate the storage of charges.

Capacitors

In the previous chapter we already found for the impedance of a capacitor $Z = \frac{1}{j\omega C}$.

We can see that the impedance goes to infinity, if the frequency goes to zero. The physical reason is that if a constant (DC) voltage is applied to a capacitor, the capacitor is being charged (i.e. a current is flowing) until the capacitor reaches the voltages. Then no more current is flowing and the capacitor is like an open.

If the frequency goes to infinity, the impedance goes to zero. If the voltage at a capacitor changes, a current flows to charge capacitor to the same voltage. The current is not infinite high and thus the charging not infinite short. If the voltage changes quickly again it changes, before the capacitor could have been fully charged. The faster the voltage changes the harder the charging can follow. Hence, if the frequency goes to infinity the capacitor behaves like a short.

Inductors

In the previous chapter we already found for the impedance of an inductor $Z = j\omega L$.

Obviously the impedance goes to zero, if the frequency goes to zero: here the inductor is just a wire, i.e. a short.

If the frequency goes to infinity, the impedance becomes infinite as well. The physical reason is that along with the frequency the self-conducting voltage gets stronger, which opposes the change in current.

Realistic Models for Capacitors and Inductors

So far we have talked about ideal devices, but real devices are not ideal and always comprise *parasitic* elements. Parasitic elements are devices (or effects) that we get due to physical effects around the device we want to build. For example two wires next to each other have both resistance and they couple with each other like a capacitor. Another example would be a semiconductor: if the current through it changes the device must be (dis-)charged, just like a capacitor.

A capacitor is basically two (wrapped up) plates. But a real device has also legs that act as resistors and inductors.
A very simple model for a real capacitor is shown in figure 2.10.
Ohmic resistance is neglected here. A capacitor is similar like the series RCL circuit (with

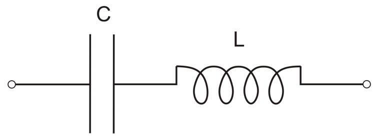

Figure 2.10: Capacitor Model

$R = 0$), which we calculated in the chapter before.
It has a resonant frequency $f = \frac{1}{2\pi\sqrt{LC}}$ where it behaves like a short; for lower or higher frequencies the impedance increases, since either the capacitor or the inductor become more and more to an open.
Remark: Beyond the resonant frequency the capacitor is rather an inductor.

An inductor is basically a wrapped up wire. But every winding couples with the neighbouring ones like a capacitor.
A very simple model for a real inductor is shown in figure 2.11.

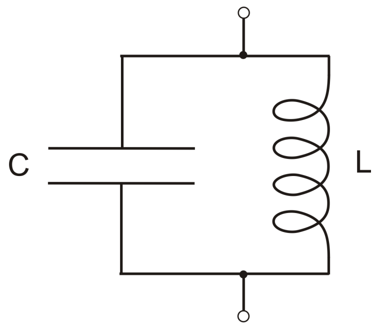

Figure 2.11: Inductor Model

Ohmic resistance is neglected. An inductor is similar like the parallel RCL circuit (with $G = 0$), which we investigated in the previous chapter.
It has a resonant frequency $f = \frac{1}{2\pi\sqrt{LC}}$ where it behaves like an open; for lower or higher frequencies the impedance decreases, since either the capacitor or the inductor become more and more to a short.
Remark: Beyond the resonant frequency the inductor is rather a capacitor.

2.6 Noise

2.6.1 Basics

Unfortunately in electronic engineering we have to deal with noise in our circuits that disturbs the quality and purity of our signal.
There are several different causes for noise, a little overview is given in the following list. Important is also the frequency response of the noise. Many noise sources are considered to contribute at all frequencies equally (at least in a certain band), this is called *white*.

- **Thermal Noise**
  Generated by the Brownian motion of all particles what leads to a randomly varying current. It is considered to be white.

- **Flicker or 1/f Noise**
  Semiconductors consist of several layers and regions like metal, isolators, ... Especially at the surface between semiconductors and isolators bindings can open and close. This leads to a noise that has a contribution versus frequency similar like $1/f$.
  Flicker Noise also shows up in resistors and other elements and the mechanism of it has not been fully understood so far.

- **Shot Noise**
  If a current flows through a potential barrier the granularity of the charge leads to a noise. This was first described by Walter Schottky.

- **Others**
  Further noise causes have been detected so far, like popcorn noise in semiconductors. More details about all noise sources, modelling noise, etc. can be found in [Mue2] and [Lee].

An important model to describe the effects of noise is the so called noise factor F. The noise factor is defined as the ratio between the ***S**ignal to **N**oise **R**atio* at the input (1) of a device and its output (2):

$$F := \frac{SNR_1}{SNR_2} := \frac{P_{S,1}/P_{N,1}}{P_{S,2}/P_{N,2}} =: 1 + F_Z$$

where P is the power and F_Z is the noise that the device adds to the input signal.
It is very popular to give the noise factor in dB, this is called ***N**oise **F**igure*, i.e. $NF = 10 \lg F$ dB.
If we have a chain of devices, each amplifies the signal by a_i, the overall noise factor can be calculated by [MaeSch]

$$F = F_1 + \frac{F_2 - 1}{a_1} + \frac{F_3 - 1}{a_1 \cdot a_2} + \cdots$$

Here we can see that the earlier a device appears in the chain, the more it contributes to the overall noise performance. Therefore weak signals are typically amplified with a chain of amplifiers, where the first device(s) have a very good noise performance, but not necessarily the highest amplification. These are called ***L**ow **N**oise **A**mplifiers*.

Attenuators are made of resistors that convert signals into heat. These resistors produce (thermal) noise. The input signal and noise are attenuated, but the device adds its own noise. If we assume, that the input signal is coming from a device with the same resistance (please see the chapter about electromagnetic waves), then we simply get noise factor = attenuation. This means that attenuators decrease the SNR by their attenuation.
Cables, filters etc. also attenuate the desired signal. Thus, to get the best performance often an LNA is used very early in the receiver chain.

Another important model is the *noise temperature*.
A resistor produces in the frequency range Δf a noise power [MaeSch]

$$P = kT\Delta f$$

where k is the Boltzmann constant.
The term kT is the spectral density of the noise and is assumed to be constant (white) in the band of interest.

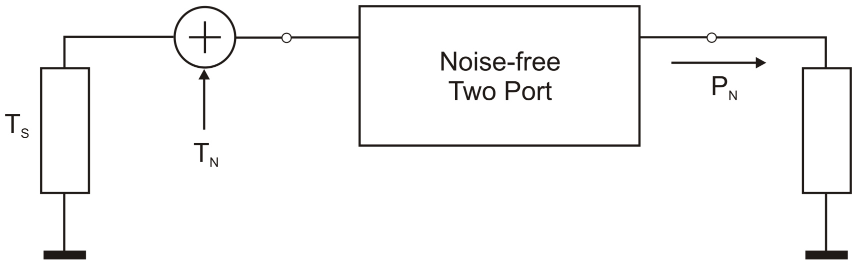

Figure 2.12: Noise Temperature Model

Now we assume a source with noise at T_S. The device shall be noise free, its noise contribution is modelled with the help of an additional noise source T_N at the input.
The relation between noise temperature and noise factor is given by

$$T_N = (F - 1)T_S = F_Z T_S \tag{2.4}$$

For a chain the overall noise temperature can be calculated by [MaeSch]

$$T_N = T_{N,1} + \frac{T_{N,2}}{a_1} + \frac{T_{N,3}}{a_1 \cdot a_2} + \cdots \tag{2.5}$$

If in praxis we measure a signal we can subtract the noise of the system from the measured result. Therefore we first measure the system noise by leaving the system on but make sure that there is no input signal. The power that we measure is the system noise. Then we can apply the signal, measure, and correct our results.

2.6.2 Noise Measurements

The noise temperature or noise figure of a device can be measured with a setup shown in figure 2.13.

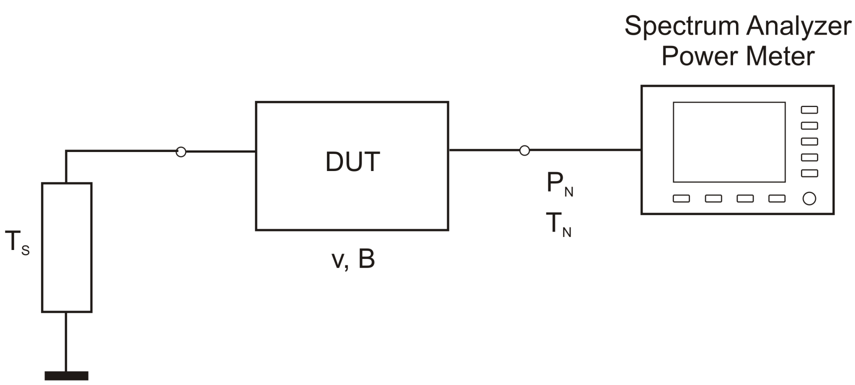

Figure 2.13: Noise Measurement: Setup

A noise source with the same input impedance Z as the DUT is connected to the input of the device. The noise temperature T_S of the source can be varied while the noise power P_N is measured at the output of the DUT. The result is show in figure 2.14.

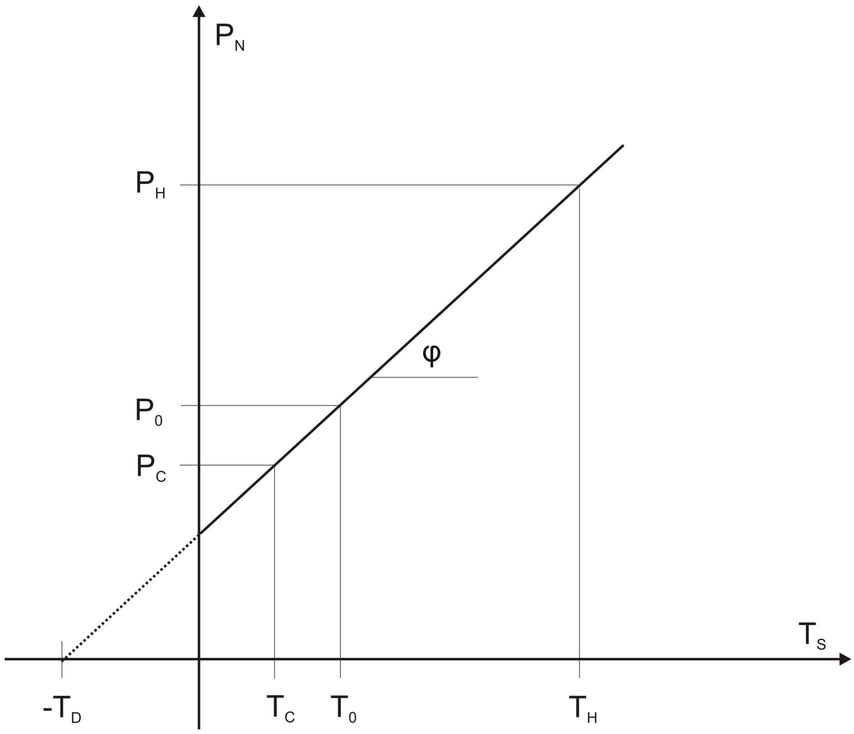

Figure 2.14: Noise Measurement: Result

The device has an amplification v and a bandwidth B.
Thus we have for the noise power

$$P_N = vkBT_S + P \tag{2.6}$$

where k is the Boltzmann constant, P the noise power of the device itself and $\tan\varphi = vkB$.
As shown in figure 2.12 a device can be modelled as a noise free element together with a noise source at its input. Therefore we can write equation 2.6 as

$$P_N = vkB(T_S + T_D) \tag{2.7}$$

where T_D is the noise temperature of the DUT, and $P_N = 0$ for $T_S = -T_D$.

With the help of the graph in figure 2.14 we can work out the noise temperature of the device.
First we measure P_N for two different noise temperatures of the source, called hot and cold. Usually a so called *Y-factor* is introduced:

$$Y = \frac{P_H}{P_C}$$

According to [MaeSch] the graph can be described as

$$T_D = \frac{T_H - YT_C}{Y - 1} \tag{2.8}$$

which gives us the noise temperature of our device.
Furthermore, together with equation 2.4, we have for the noise figure of the DUT

$$F = \frac{(T_H/T_0 - 1) - Y(T_C/T_0 - 1)}{Y - 1} \tag{2.9}$$

where $T_0 = 290\,\mathrm{K}$ is room temperature.

A popular noise source are noise diodes, which deliver different noise powers for different supply voltages.

2.7 Filters, Decoupling, Blocking, Bias Tee

2.7.1 Filtering

Signals have to be filtered all the time.
Transmitters produce in addition to the intended signal other outputs (please see the chapters about amplifiers, synthesizers, ...). These distortions typically appear somewhere else in the frequency domain and can be filtered away. They have to be filtered, if they occupy frequencies that are also used by other transmitters and would overlay them.
Receivers can generally receive much more than the intended signal. Often they are designed to receive a whole frequency band in which the wanted signal just uses a part of it. Thus, everything around the desired signal shall be filtered away.
Digital signals have a so called *quantization noise* that can be filtered (please see the chapter about digital signal processing).
To filter signals different types of filters are available (both, digital and analogue):
L*ow* ***P****ass* ***F****ilters*, ***H****igh* ***P****ass* ***F****ilters*, ***B****and* ***P****ass* ***F****ilters* and many others.

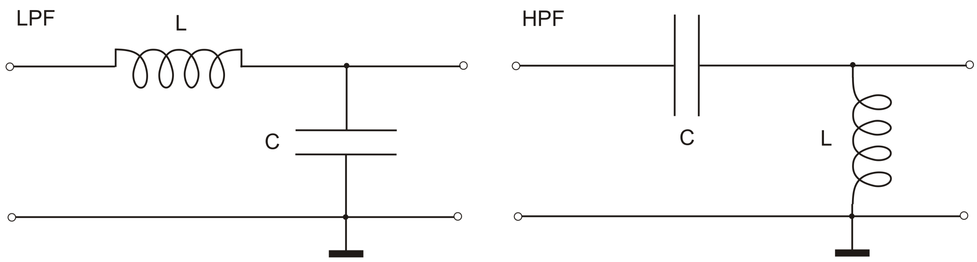

Figure 2.15: Low and High Pass Filter

As example picture 2.15 shows a typical and simple implementation of a LPF and a HPF. Meanwhile we can easily calculate the performance with the help of our complex models. But we can immediately understand the principle: as discussed the impedance of a capacitor decreases with higher frequencies, while the impedance of an inductor increases. Thus the LPF circuit in picture 2.15 is a short for DC. If the frequency is increased the resistance is also increased due to the inductor, and at the same time the LPF better and better shorts the input signal via the capacitor to ground. If capacitor and inductor are exchanged the circuits behaves as HPF. We can also combine them for example to a BPF.

Figure 2.16 shows some definitions for filters (using a LPF as example).

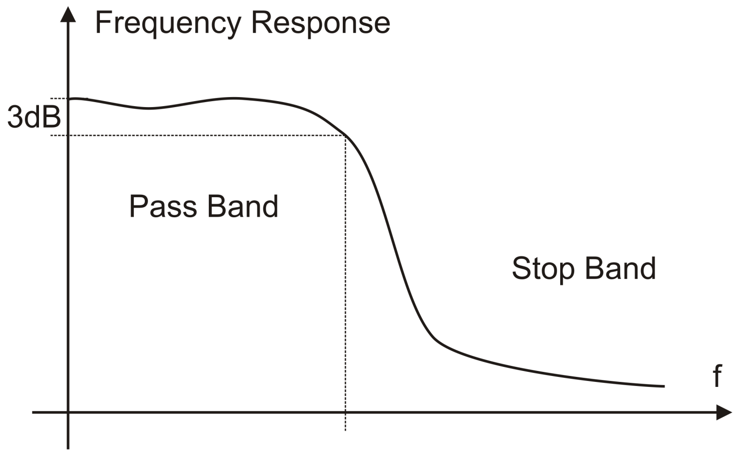

Figure 2.16: Band Definitions (Example Low Pass Filter)

The frequency range that is attenuated is called *stop band*, the low attenuated range *pass band*. To mark the band of interest usually the so called *3 dB corner* is used.

Very often a certain *decoupling* of circuits or *blocking* of certain signals is necessary. For example a wire can act as little antenna. Or it can capacitively or inductively couple to another wire. Thus, it can catch up (or transmit) unwanted signals.
A popular example would be a wire that connects a circuit to its voltage supply. At the same time the wire can catch up the signal of a synthesizer at a certain frequency. This signal can disturb the device, for example an amplifier could modulate the unwanted signal on the wanted.
Another problem is that devices do not continuously draw current. For example a digital circuit draws a peak current every time it is clocked to perform its calculations. This current changes can be distributed over the wire to other devices that are connected to the same supply.

To avoid these effects filters can be implemented on wires.
We saw that capacitors act like a short at their resonant frequency. This frequency can be chosen by the size of the capacitance and the size of the capacitor (which defines the parasitic L in our simple model). Thus, with the help of a capacitor, a signal with a certain frequency can be shorted to GND (in our example the disturbing signal from the synthesizer on the supply line).
In addition to this supply connections should have star topology so that current changes of circuits do not propagate to other devices.
Capacitors at star points of supply lines help to stabilize the voltage.

Furthermore circuits can be covered by metal shieldings to decouple them from their environment.

2.7.2 Bias Tees

Sometimes a DC Voltage (for example a supply voltage) and an RF signal shall be separated or combined. Picture 2.17 shows a simple principle for a so called *Bias Tee*:

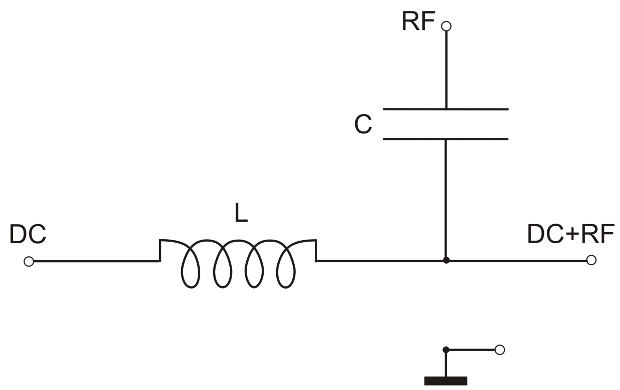

Figure 2.17: Bias Tee

The capacitor blocks the DC and is of low impedance for RF while the inductor has a high impedance for the RF but is a short for DC.

2.8 Electromagnetic Waves

R*adio* ***F****requency* engineering covers a wide range of the electromagnetic spectrum, as shown in picture 1.1.

The first important parameter is the ratio between wavelength and component size. If they are both in the same range or if the component is even larger then the wave propagation is getting important, i.e. components get antennas. Here the geometry plays an important role, like size and place of components etc. If the component size is much smaller than the wavelength those effects are becoming uncritical. But still the strength of parasitic capacitors and inductors including coupling between components will depend on the frequency.
In our case the wavelength is approximately $\frac{3 \cdot 10^8 \frac{\mathrm{m}}{\mathrm{s}}}{1.5\,\mathrm{GHz}} = 20\,\mathrm{cm}$ – much longer than our used components.

2.8.1 Transmission Lines

An infinitesimal short transmission line can be modelled as shown in picture 2.18. Z_L is the impedance of a *load*, which is connected the transmission line.
A real transmission line would be modelled as a chain of these circuits. The general solution of the resulting differential equation is very difficult. But solutions can be found for special cases like sinusoidal signals and zero resistance, see for instance [ZiBr].

If we compare the model with the examples in the previous chapters we can immediately see, that transmission lines behave as low pass filters – what limits the frequency of the transmitted signal.

Furthermore every transmission line has its own, characteristic impedance Z. It does not

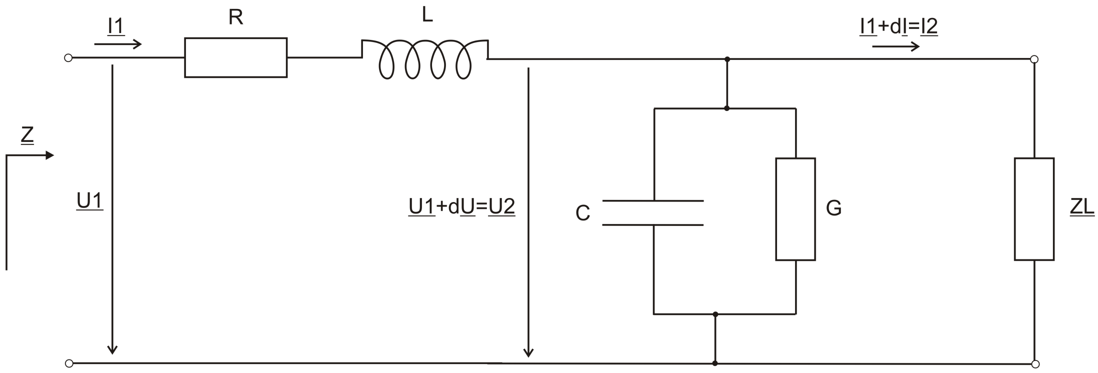

Figure 2.18: Transmission Line Model

depend on the length of the line and is given by

$$Z = \frac{\underline{U}}{\underline{I}} = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

Another important result concerns the reflection of electromagnetic waves. If an electromagnetic wave propagates along a transmission line we have the relationship $Z = \frac{\underline{U}}{\underline{I}}$. If the wave encounters a circuit with a different impedance Z_L, a part of the wave is being reflected.
We can introduce a reflection coefficient that describes (e.g. [ZiBr] or [Web]) the amount of reflection:

$$\Gamma := \frac{\underline{U}_{\text{reflected}}}{\underline{U}_{\text{forward}}} = \frac{Z_L - Z}{Z_L + Z}$$

In communication engineering typically 50 Ohm systems are used, in other areas like video and audio technique also 75 Ohm are very popular. For example [Lee] describes why 50 Ohm and 75 Ohm.

Sometimes circuits shall be combined, which do not have the same impedance. But it is possible to build so called *matching networks*, often a simple LC-combination, that transforms one impedance to another.
Also transmission lines transform the impedance of a load, if the load impedance differs from the characteristic impedance of the transmission line.
If a lossless transmission line has a characteristic impedance Z and is connected at the end to Z_L, then the impedance seen at the beginning of the transmission line can be calculated by [ZiBr]

$$Z_1 = Z \frac{\frac{Z_L}{Z} + j \tan \frac{2\pi l}{\lambda}}{1 + j \frac{Z_L}{Z} \tan \frac{2\pi l}{\lambda}} \tag{2.10}$$

where l is the length of the line and λ the wavelength.

Let's have a look at two important examples, also to get familiar with the situation:

Assume $l = \frac{\lambda}{4}$. Thus we get $\tan \frac{2\pi l}{\lambda} = \tan \frac{\pi}{2} = \infty$ and $Z_1 = \frac{Z^2}{Z_L}$.
Two extreme cases exist for such a quarter wavelength transmission line:

If it is left open, i.e. $Z_L = \infty$, then on the other end of the transmission line is seen a short.
A short, i.e. $Z_L = 0$, would be transformed into an open.

Assume $l = n\frac{\lambda}{2}$ (n natural number). Then we get $\tan\frac{2n\pi l}{\lambda} = \tan n\pi = 0$. Thus the impedance seen at the input of the transmission line is identical to the load.

Phillip H. Smith invented a diagram that is very useful. First we normalize equation 2.10 by dividing by Z and thus we have $\frac{Z_1}{Z} = \frac{R}{Z} + i\frac{X}{Z}$. Now the equation is mapped to the complex plane. The Smith-chart is shown in figure 2.19.

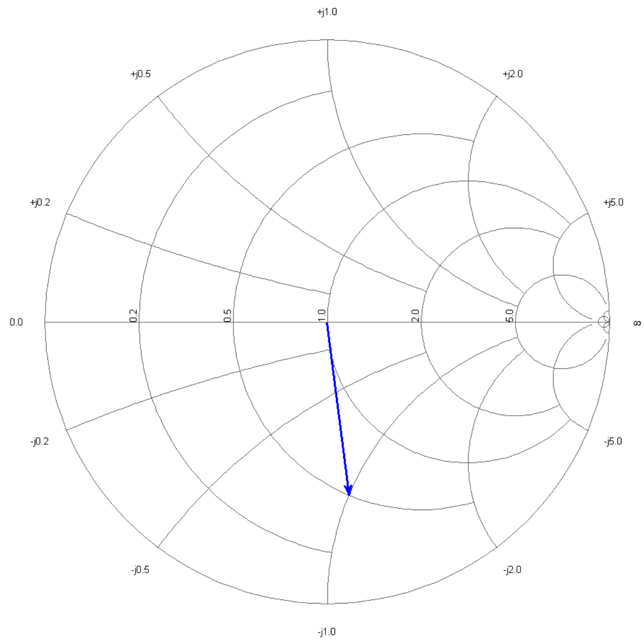

Figure 2.19: Smith Chart

It is not necessary to discuss the mathematical mapping, but we have a look at some examples to get familiar with the chart. We always assume $Z = 50\,\text{Ohm}$.

If we have an impedance of 50 Ohm we get after normalization 1. This is the centre of the Smith chart.
For an impedance of $(25 + i50)$ Ohm we get after normalization $0.5 + i$. This is where the blue vector in the chart points to.
Note the coordinate system of the chart: the real part is given on the horizontal axis, the imaginary part on the outer circle, both are following the plotted circles. The point is located at the crossing of the two circles.
The length of the vector is the reflection coefficient. For the first point it is zero since 50 Ohm means a perfect matching. All points on the outer circle have reflection one: these impedances are mismatched and the wave is fully reflected. This scale in the Smith chart is linear between zero in the centre and one on the outer circle.
The point on the right hand side is infinity, i.e. an open, the point on the left zero, i.e. a short.

Transmission lines transform impedances. This behaviour can be seen in the chart, where all points are moved along a circle, which centre is located at the centre of the chart. The length of the circle, i.e. the angle, is given by the length of the transmission line: a half circle, meaning 180°, would be a transmission line with a length of $\lambda/4$. For example connecting a quarter wavelength cable to a short would transform it along the outer circle by 180° to an open on the right. We already found the same result mathematically with the help of equation 2.10.

The chart does not only show impedance transformations by transmission lines or reflection coefficients of mismatches.
The Smith chart can also help to design matching networks, for example if we want to match a certain impedance Z_L to $Z = 50\,\text{Ohm}$. First we can plot Z_L/Z into the chart. We remember that capacitors and inductors offer complex impedances ($j\omega L$ and $-j/(\omega C)$). Now we can work out, which capacitors and inductors move the point along the coordinate system to the centre. If we found the correct impedances in the chart we have to multiply them with Z (since all values are normalized).

Electromagnetic waves in media

Often transmission lines are implemented with the help of dielectric media, for example PCBs, coax cables and many others. Dielectric media typically have little loss, magnetic susceptibility μ_0 and an electric susceptibility of $\epsilon\epsilon_0$. This influences several parameters of the wave, among others

$$\lambda = \frac{\lambda_0}{\sqrt{\epsilon}}$$

$$v = \frac{c}{\sqrt{\epsilon}}$$

where c and λ_0 are the speed of light and the wavelength in vacuum.

2.8.2 Power Splitting / Combining

Often RF signals shall be combined or split, for example a source shall drive different devices. Let's say we are in 50 Ohm system and all our devices have 50 Ohm as input / output impedance. Now, if for example we simply connect the source to two devices, the source would not see 50 Ohm any more, but two 50 Ohm in parallel, that is 25 Ohm. As we saw this would lead to reflections.
Furthermore it shall be mentioned that to transmit the maximum amount of power from a source to a load, the input impedance of the source should be the conjugate complex of the load impedance [3].

Thus we need some extra effort to match the impedances. These devices are so called *power splitters / combiners*. There are several implementations, most popular is a simple star topology with three resistors. The splitter / combiner has three terminals, each

[3] A source can be modelled as voltage source and a serial impedance, its internal resistance. Now the load is connected which gives two serial impedances. You can now calculate the current ($\underline{u}/(Z_S + Z_L)$), the power in Z_L and look for the maximum.

provides for instance 50 Ohm (if also the other two are terminated with 50 Ohm).

Sometimes it is necessary to attenuate a signal or sink it. Attenuators and terminators can simply be realized with resistors, which dissipate the power and provide at the same time the desired port impedance.

2.9 Scattering-Parameters

As mentioned in the chapter about electromagnetic waves two waves can propagate at the same time forward and backward on one transmission line. The voltage on the line is the sum of U_{forward} and U_{backward}.
The line has a resistance $Z = U/I$.

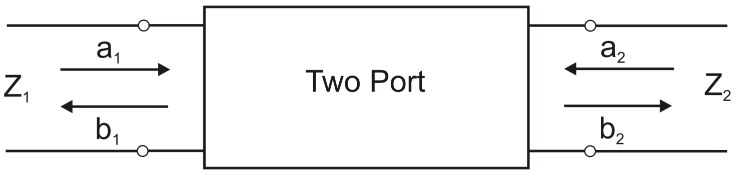

Figure 2.20: Two Port

We define

$$a := \frac{U_{\text{forward}}}{\sqrt{Z}}$$

$$b := \frac{U_{\text{backward}}}{\sqrt{Z}}$$

With this definition we don't have to differentiate between voltages and currents.

Now we define the complex *S-Parameters*:

$$\underline{b}_1 =: S_{11}\underline{a}_1 + S_{12}\underline{a}_2$$

$$\underline{b}_2 =: S_{21}\underline{a}_1 + S_{22}\underline{a}_2$$

If a resistor with impedance Z is connected to the output, then $a_2 = 0$ and S_{11} gives the ratio between forward and reflected signal. At the same time S_{21} gives us the ratio between the forward and the signal at the output, i.e. it shows us how the signal is being transferred. We can look at it also the other way around with Z at the input.

One of the standard tasks in radio frequency engineering is to measure S-Parameters. For this purpose so called *Network Analyzers* are being developed and are offered by several companies. Network Analyzers usually provide two ports, for example with 50 Ohms, which are connected to the input and output of the ***D**evice **U**nder **T**est*.

Further details about S-Parameters and more can be found in [ZiBr] or [Web].

2.10 Linearity, Dynamic Range

Linearity is often a crucial issue.
By nature semiconductors, the most popular devices in electronics engineering, do not behave linear. As simple example the current of a diode follows exponentially the applied voltage, shown in figure 2.21.

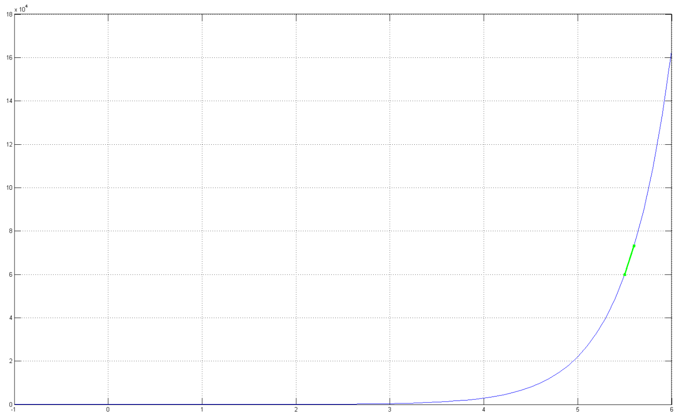

Figure 2.21: Exponential Function, Diode

Often only a part of the voltage (current) range is used, in which the device is as linear as possible, for example the green highlighted area in graph 2.21.

Another example is the transfer function of an amplifier. Mostly amplifiers have a linear range in which the output power follows linear the input power. But of course the amplifier cannot amplify to infinity. Thus, if the input power is increased high enough, the output power cannot follow any more - the amplifier goes into *saturation*.

An important definition, shown in figure 2.22, is the so called *3 dB compression point*: this is the point, where the output power stays 3 dB behind the ideal, linear transfer function.

The non-linear transfer function can be described by a Taylor-series:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + \cdots$$

If we use as test signal $x = \sin(\omega t)$ and remember

$$\cos\alpha\cos\beta = \frac{1}{2}(\cos(\alpha - \beta) + \cos(\alpha + \beta))$$

we can easily see, that the non-linear behaviour creates signals like $\cos(n\omega t)$, where n is a natural number. These are called *harmonics*.
Furthermore, if we use as input a sum of signals like $x = \cos(\omega_1 t) + \cos(\omega_2 t)$, we will get all kind of products like $\cos((n\omega_1 + m\omega_2)t)$, they are termed *intermodulation products*.

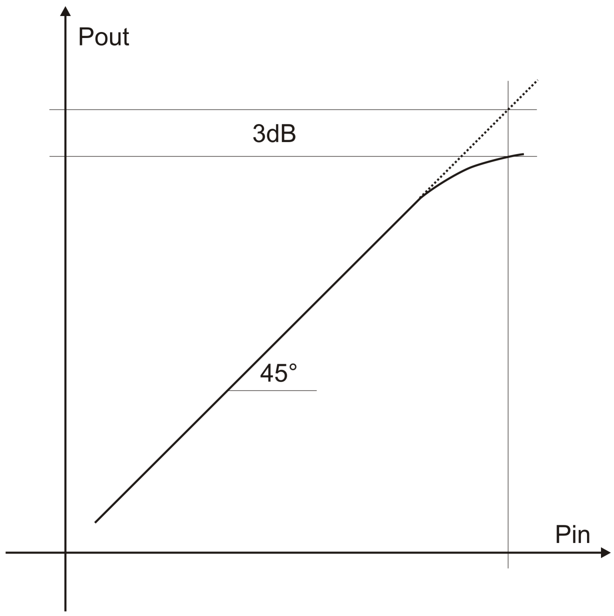

Figure 2.22: 3 dB Compression Point

Due to non-linear behaviour signals can be heavily distorted.

Another popular term is *dynamic range*. The dynamic range is the range of amplitudes, in which signals can be handled: for example where a device is linear, or the range that is covered by a digital circuit (digital circuits have a fixed number of bits that they can work with and thus a fixed range of numbers, which they can cover).

2.11 Synthesizer

A synthesizer is a device, that generates certain frequencies.

A classical synthesizer principle is the ***P**hase **L**ocked **L**oop* as shown in figure 2.23.
The heart of the synthesizer is the ***V**oltage **C**ontrolled **O**scillator*, which generates the output of the synthesizer. The VCO is basically a LC parallel circuit, where the capacitance of the capacitors can be controlled via voltage. These devices are called *varactors*.
We already found that the resonant frequency, at which the circuit is oscillating, is given by

$$f = \frac{1}{2\pi\sqrt{LC}}$$

If for example we have a varactor, for which the capacitance increases with higher voltages, then the frequency would go down by increasing the voltage.

Another element of the PLL is the reference frequency *Ref*. The reference frequency is much lower than the desired output frequency. It is produced by a special device that can deliver a very stable, low noise signal. Very popular are crystal oscillators, which use mechanical resonance.
The output frequency is divided down to the reference frequency.

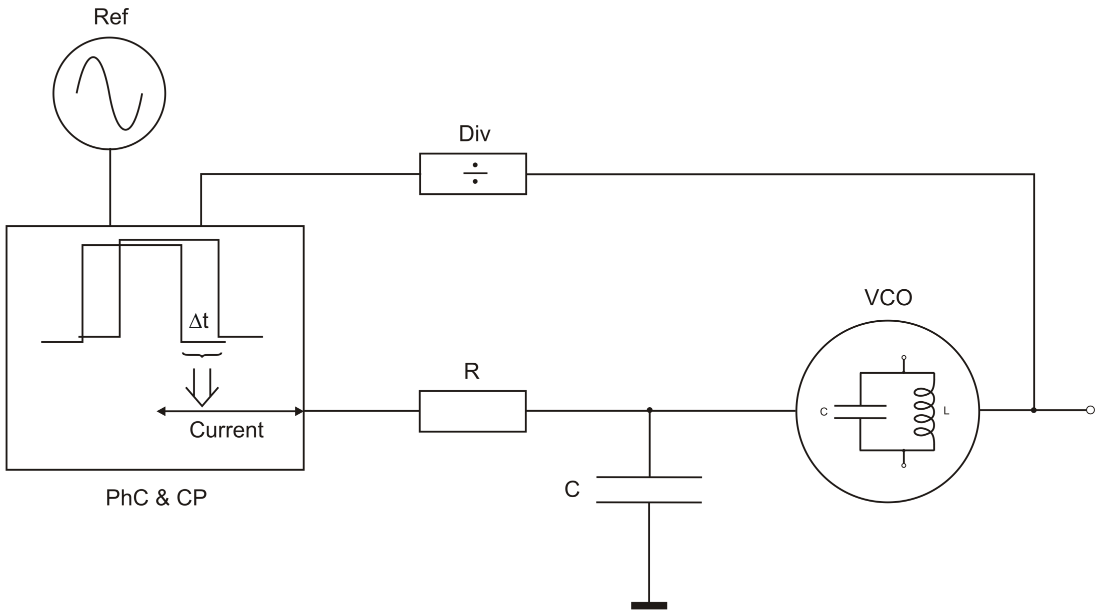

Figure 2.23: Phase Locked Loop

The edges of both signals (in the picture drawn as square) can be compared by a so called ***P**hase **C**omparator*. The phase comparator controls a ***C**harge **P**ump*. The charge pump generates a current that (de-)charges the capacitor, thus the input voltage of the VCO can be controlled.
Let's have a look at an example:
We assume that the edge of the divided output signal comes before the reference clock, meaning the frequency is too high. The PhC & CP detects the phase delay of the reference as shown in the picture. During Δt a positive current is generated to charge the capacitor and increase the voltage of the VCO. If we have a VCO, which decreases the frequency for higher voltages, the frequency goes down a bit. When the frequency is correct, i.e. the phases of reference clock and divided output signal are aligned, Δt as well as the current are zero and the VCO is settled.
If the frequency is too low then Δt would be negative as well as the current; hence the voltage would go down and the frequency up.

The RC circuit behaves as a low pass filter, similar like the LC circuit that we discussed before. It is also called *loop filter*.

A synthesizer cannot produce a perfect signal, the signal always also comprises noise, often simply referred to as *phase noise* [4].
In the frequency domain the output of the synthesizer typically looks like in picture 2.24. As performance criteria is used the ratio between the signal (carrier) power and the noise power per Hertz at certain frequency *offsets* (distances from carrier).

[4] Both, phase and amplitude can contain noise. But the amplitude noise can be converted to phase noise. If phase noise is measured often the measuring device can track the amplitude of the signal to cancel the amplitude variations.

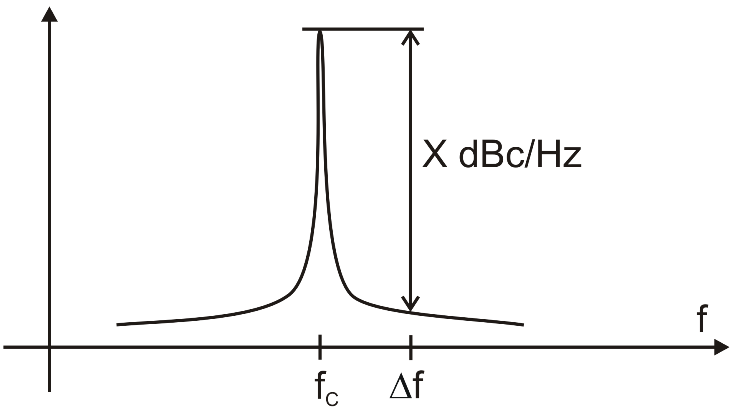

Figure 2.24: PLL Phase Noise

Remarks:

Often a synthesizer shall be able to generate different frequencies: this can be achieved by varying the Divider.

This principle, to generate a highly accurate signal by comparing it to a reference, is very popular, for instance to achieve a precise output power, voltage etc.
It mainly relies on a precise comparison and a good reference, other parts of the loop can be of much lower performance.

There are other possibilities to generate signals, for example *direct digital synthesizer*, where a signal is saved as digital data in a memory and read out at a certain speed to achieve the wanted frequency.

2.12 Mixers and Mixing

Often signals shall be moved from one frequency to another one. For instance if in communication the information is transmitted at several GHz, it can be very difficult or rather impossible to process the information at such a high rate. But a receiver can move the signal to a low frequency where it can be easily handled or digitized.

The technique for moving signals in the frequency domain is based on the well known equation

$$\cos\alpha\cos\beta = \frac{1}{2}(\cos(\alpha-\beta)+\cos(\alpha+\beta))$$

This is called *mixing*. And the device, which multiplies the two signals, is called *mixer*.

Let's assume we have a signal, the so called *wanted*, around frequency $f_W = \alpha$, and want to move it to $f = \Delta f$. The desired destination frequency is termed ***I****ntermediate* ***F****requency*. We only have to multiply it with a sinusoidal signal, often called ***L****ocal* ***O****scillator*, at for instance frequency $f_{LO} = f_W - \Delta f = \beta$.
Mixing also produces a copy of the signal at $f_W + f_{LO}$, which simply can be filtered with a low pass filter.

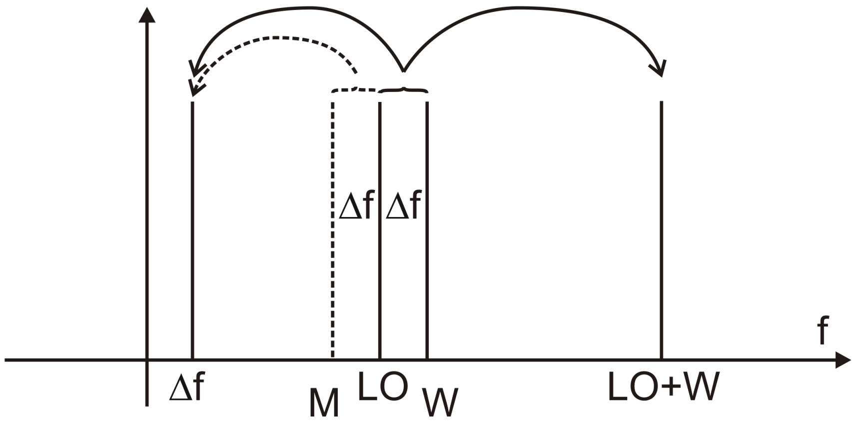

Figure 2.25: Frequency Mixing

Unfortunately also a signal at the so called mirror frequency, in our example $f_M = f_{LO} - \Delta f$, is mixed down to the very same frequency Δf! This unwanted signal is also referred to as ***im**age*.
This effect can be avoided by a clever mixing called *image rejection* or *image canceling* mixing:
Therefore we need also our LO shifted by 90°.
Furthermore we remember some more equations:

$$\cos(\gamma + \pi/2) = -\sin(\gamma) = \sin(-\gamma)$$
$$\sin(\gamma + \pi/2) = \cos(\gamma) = \cos(-\gamma)$$
$$\cos\alpha \sin\beta = \frac{1}{2}(\sin(\beta - \alpha) + \sin(\beta + \alpha))$$

In the following calculations we can leave the $f_{LO} + f$ signals away.

In a first step we multiply our wanted $\cos(f_W)$ with the LO $\cos(f_{LO})$ and the 90° phase shifted LO $\sin(-f_{LO})$ [5]:

$$\mathrm{if}_1 = \frac{1}{2}\cos(f_W - f_{LO})$$
$$\mathrm{if}_2 = \frac{1}{2}\sin(f_W - f_{LO})$$

In a second step we sum if_1 and a 90° phase shifted if_2:

$$\begin{aligned} \mathrm{if} &= \frac{1}{2}\cos(f_W - f_{LO}) + \frac{1}{2}\sin(f_W - f_{LO} + \pi/2) \\ &= \frac{1}{2}\cos(f_W - f_{LO}) + \frac{1}{2}\cos(f_W - f_{LO}) \\ &= \cos(f_W - f_{LO}) \end{aligned}$$

So far not much happened and we would have achieved the same result with the simple multiplication. But now let's see what happens to the image:

[5] Actually all signals have the form $\cos(2\pi f t)$, but the $2\pi t$ is suppressed for clarity.

The important observation is that the image is on the other side of the LO, in our case $f_M < F_{LO}$. But negative frequencies of course do not exist.
Thus we get for the image the same if_1, but

$$\text{if}_2 = \frac{1}{2}\sin(-(f_W - f_{LO})) = -\frac{1}{2}\sin(f_W - f_{LO})$$

Again we sum if_1 and the 90° phase shifted if_2:

$$\begin{aligned}\text{if} &= \frac{1}{2}\cos(f_W - f_{LO}) + (-\frac{1}{2}\sin(f_W - f_{LO} + \pi/2)) \\ &= \frac{1}{2}\cos(f_W - f_{LO}) - \frac{1}{2}\cos(f_W - f_{LO}) \\ &= 0\end{aligned}$$

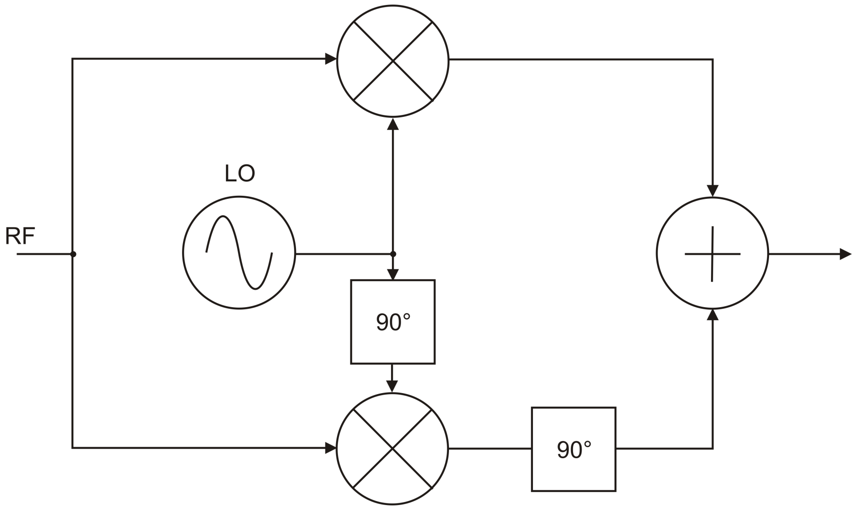

Figure 2.26: Simple Image Canceling Mixer

Remarks:

How much the signal is distorted by the mixing depends on the performance of the mixer and the LO. Here, typical specifications are the linearity of the mixers and the phase noise of the LO.

Mixing a signal up / down is also termed *up / down conversion*.

So far we have spoken about mixing down to a low intermediate frequency. Of course we could mix down to zero. But here several problems can occur:
For example at the mixer the signals can couple from one input to the other one (e.g. LO couples to RF input). In this case they convert themselves down to DC, overlay the desired signal and use up the dynamic range.
In addition to this the result depends also on phase difference between the two multiplied signals.
To avoid these DC problems it is popular to mix down to an intermediate frequency. DC and everything beyond the band of interest can easily be filtered. If the IF is low enough the signal can then be digitally processed.

2.13 Amplifiers

The whole field of amplifiers is much too large to discuss it here in greater details. This is also true for transistors, the heart of every amplifier. However it is possible to a achieve at least a rough insight how transistors and amplifiers work.

Transistors can be split into two types, Bipolar and Field Effect. Their principle is simplified shown in figure 2.27.

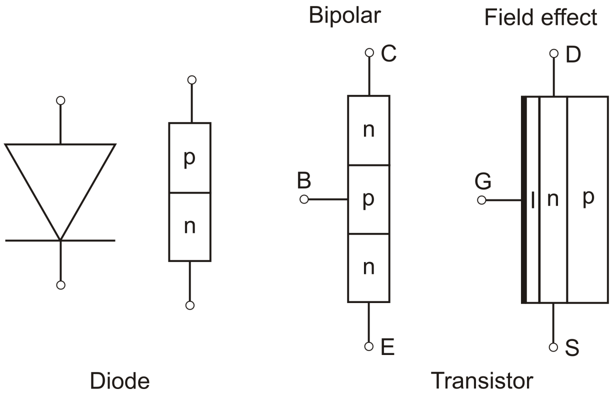

Figure 2.27: Diode, Bipolar and Field Effect Transistor

Both are so called *semiconductors*, mostly composed of silicon, sometimes also germanium or gallium-arsenic. The element silicon belongs to the fourth group in the periodic table and has four valence electrons. Silicone can be doped with other atoms. If a silicone atom is replaced with an atom of the fifth group, like arsenic, only four of its valence electrons are needed for the binding. The fifth electron can easily be released from the atom and can serve for conducting. The atom is termed donor and the doped silicon n-type. Similar an atom of the third group, like gallium, can be introduced. Here one electron is missing what is called hole and can be modelled as a positive charge. The atom is termed acceptor and the doped silicon p-type.

Let us first have a look at the *diode*. Its structure is shown in figure 2.27 as well as its symbol on the left. A diode is composed by an n-type and a p-type region. First we apply a Voltage with the positive potential connected to the n-region and the negative to the p-region. The electrons as well as the positive holes are attracted by the potentials and the diode gets discharged. Thus it can not conduct any longer. Now we apply the voltage with the positive potential connected to the p-region, the negative connected to the n-region. Both potentials repulse either the electrons or the holes. They flow to the opposite side of the diode where they are attracted. Thus a current can flow that depends exponentially on the voltage. The symbol of the diode can be seen as an arrow which points to the direction the current can flow, i.e. from plus to minus.

Next we have a look at the bipolar transistor. It is composed of a structure like the diode, for example here B-E, but with a third region C. Similar like discussed for the diode no current can flow between E and C. But if a voltage is applied between B and E, the B-E part acts like a diode. Only a small current flowing from B to E sufficiently charges the transistor. If now a higher voltage is applied to C a current starts to flow from C to E. This means that a small voltage / current B-E can control a significantly higher voltage / current C-E. They can roughly differ by a factor ten to hundred. The C-E current does not follow the E-B current / voltage linearly, but similar as discussed for the diode in the chapter about linearity, a linear range for small signals can be found, see also figure 2.21. The transistor can also be used as switch by turning on / off E-B and thus C-E. The terminals E-B-C are termed ***E****mitter*, ***B****ase* and ***C****ollector*.

Last we discuss the ***F****ield* ***E****ffect* ***T****ransistor*. They are composed by a conducting channel, its terminals are called ***D****rain* and ***S****ource*. Further FETs comprise a third terminal, called ***G****ate*, parallel to the channel. Channel and Gate are isolated, therefore no current flows through the Gate [6]. If a D-S voltage is applied a high current can flow through the channel. By applying a voltage to the Gate the field attracts or repulses the charges in the channel. Therefore the current in the channel can be modulated. The D-S current is not a linear function of the G voltage, but again a linear range for small signals can be found. By turning the Gate voltage on / off the FET can be used as switch, which is the use case in digital circuits [7].
For bipolar and even more for field effect transistors many variants were developed, for example the bipolar structure can be p-n-p, the FET channel p etc. Thus the direction of currents can be chosen as well as the needed voltages at the three terminals. Furthermore different technologies are used to optimize power consumption, size, noise, cost, heat, speed, linearity or the applicable currents and voltages.

A single transistor can be used in several ways to amplify a current or a voltage. Around the transistor some circuitry is needed above all to supply the transistor and to generate a certain input and output impedance. More advanced amplifiers can use several transistors to achieve a higher amplification and can comprise further circuitry to adjust the amplification, power consumption, current limiting etc.

[6] Actually a current flows to (dis-)charge the gate, thus the gate can be seen as a capacitor. If the frequency increases this current significantly increases along. Please also note that all currents produce heat. Hence, for instance in digital technique this can even be a limiting factor.

[7] With only very few transistors all basic elements that are needed in digital technique can be composed, like (N)AND, (N)OR, NOT etc. or the fastest available memory.

Chapter 3

Small Radio Telescope: Introduction and Overview

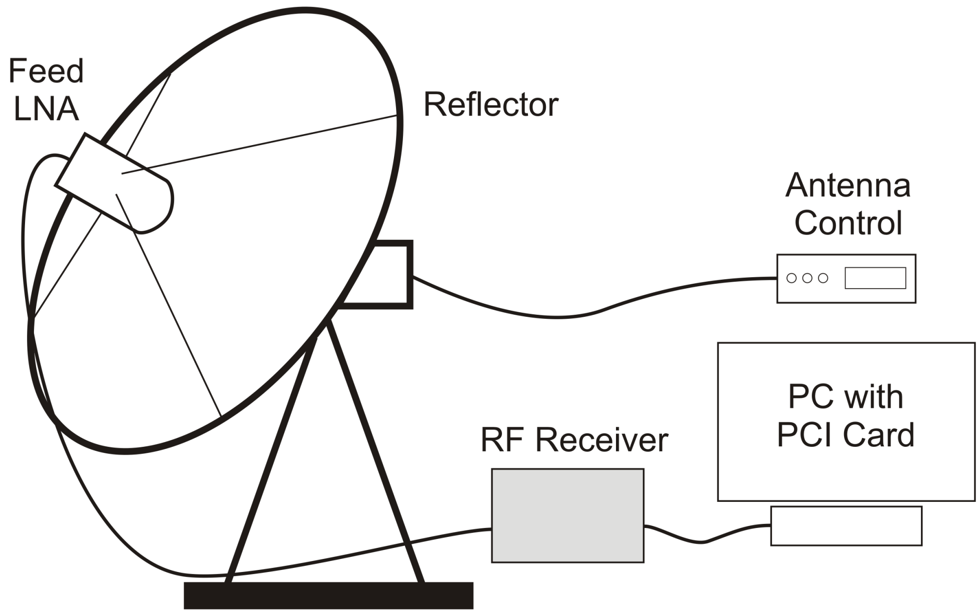

Figure 3.1: Small Radio Telescope

Figure 3.1 shows the entire ***S****mall* ***R****adio* ***T****elescope*.

First a metal *reflector* focuses the electromagnetic radiation. A little antenna, called *feed*, is located at the focal point. Antennas transform electromagnetic radiations into cable-bound waves – and vice versa in case of transmitting. The cables and components of the SRT have 50 Ohm impedance.
A ***L****ow* ***N****oise* ***A****mplifier* is directly connected to the output of the feed. The LNA actually comprises two low noise amplifiers and a band pass filter.
Then the signal is lead with the help of a coaxial cable into the ***R****adio* ***F****requency Receiver*. The receiver is further amplifying and filtering the signal. In addition to this the signal is mixed down to approximately 5 MHz, since signals can be much better handled at low frequencies.
Finally the signal is digitized by an ***A****nalog to* ***D****igital* ***C****onverter*. The ADC is on a PCI-Card, which is located in the PCI bus slot inside the PC. This card is controlled with the

help of a dedicated driver.
A software package is running on the PC to control the ADC via the driver, gather the data, process it and to visualize and save the results. As platform was chosen the Linux distribution Ubuntu.
Prof. Dr. Warren Jasper provided the ADC-driver.

Further two programmes were developed:

- A program called *sample_signal* that was written in C. *sample_signal* uses the ADC via its driver to record the signal. The data is saved in a text file.
- An Octave (MATLAB) program called *Analyse.m*. *Analyse.m* reads in the data, processes it [1] and finally visualizes and saves the results. *Analyse.m* can also call *sample_signal*. Thus with *Analyse* alone measurements can be accomplished.

Introductions to the driver, software tools, Linux, Ubuntu, Octave (MATLAB), *sample_signal* and *Analyse.m* can be found in the appendix.

Further the Antenna can be rotated via an antenna controller to look at a certain astrophysical object.

[1] Details are described in the chapter about digital signal processing.

Chapter 4

The Antenna

Unfortunately the field of antenna theory and design is much too large to discuss it within this thesis. It comprises the Maxwell equations, accelerated charges, relativistic transformation of fields and charges, radiation, propagation of electromagnetic waves, near and far fields of antennas, reflections, impedance matching between antennas and circuits, etc. Furthermore the mathematical modelling of antennas, wave guides or similar is too complex and usually must be solved numerically. There are several software packages on the market for this purpose, but they were not available for this thesis. Note also Sergey Makarov's book *Antenna and EM Modeling with MATLAB*. In addition to this a special laboratory would be needed to measure the antenna. Such laboratories contain so called *anechoic chambers* (meaning echo free). An anechoic chamber is typically a room or hall composed of metal walls, where all interior surfaces are covered with an absorbent material. This decouples the measurement from the environment and prevents disturbing reflections. In addition to the antenna the chamber covers a reference antenna. Both antennas can be rotated around each other to measure the full three-dimensional characteristic of the ***A****ntenna* ***U****nder* ***T****est* [1]. Signal generators and Analyzers are connected to the antennas to generate test signals and measure the performance. Antennas that are too large too fit into a chamber might be verified outside with the help of reference signals.

Antennas can appear in very different shapes and sizes, depending on frequency, bandwidth, power, direction and polarisation of the radiation, costs or available space. Sometimes several antennas are connected, for instance for superposing the radiations to increase power or to use different phases of the radiations to focus in one direction or to generate certain polarisations. Other examples are to triangulate signals, to increase data rate by using several transmissions in parallel or to increase the angular resolution as discussed in chapter 1.

For many type of antennas scientist and engineers were able to work out equations that are now provided in the literature and that can serve as design rules.
Furthermore at least one important measurement can be done also without chamber, that is to verify the input impedance of the antenna, i.e. the reflection coefficient seen from the cable.

[1] Similar as for components that are verified and called ***D****evice* ***U****nder* ***T****est*, we term the antenna that we are verifying ***A****ntenna* ***U****nder* ***T****est*.

This approach is taken in this thesis.

4.1 Antenna Basics

Antennas are used to transform cable bound electromagnetic waves into radiation or vice versa.

4.1.1 Field Zones

Dipoles belong to the most basic and simplest antennas. Infinitesimal dipoles can also be used to model more complex structures, what is one of the numerical methods mentioned before.
The field around an (infinitesimal) dipole antenna can be roughly divided into two regions ([Bala], [KraMar], [Kark]).

- The field closer to the antenna is called *near field* or *Fresnel zone*. It is proportional to $\frac{1}{r}$, where r is the distance to the antenna. The field can have longitudinal and transversal components. Most of the energy is stored and not radiated.

- The field further away from the antenna is called *far field* or *Fraunhofer zone*. This field is radiated from the antenna and is proportional to $\frac{1}{r^2}$, where r is the distance to the antenna. The propagating field has only transversal components. At the antenna fields might start with a complex geometry, the simplest one would be a unit sphere. But in the far field, if the radius is large, it can locally be seen as plane wave.

4.1.2 Power and Impedance

In this section we remember some facts that are derived in the theory of electromagnetic waves and that can be used for the far field.

Electric field E and magnetic field $B = \mu_0 H$ are related as

$$\frac{E}{H} = Z_0 = \sqrt{\frac{\mu_0}{\epsilon_0}} \approx 377\,\mathrm{Ohm} \approx 120\pi\,\mathrm{Ohm}$$

where we can define an impedance Z_0.
The Poynting vector $\vec{S}$ points to the direction of the propagation, its length describes the energy that flows per unit area and time, and it can be calculated as

$$\vec{S} = \frac{1}{\mu_0}(\vec{E} \times \vec{B})$$

Similar as for the electric power $p = ui = u^2/R$ we have for an electromagnetic wave $S = \frac{1}{\mu_0} EB = E^2/Z_0$.

4.1.3 Radiation Pattern, Directivity, Gain

Antennas cannot uniformly receive or radiate in all directions. Such an isotropic radiator is only hypothetical and used as reference. Furthermore as described in the *Reciprocity Theorem* section below, the pattern for receiving and radiating are identical, therefore it is sufficient to look at either one.
An example-pattern is shown in figure 4.1.

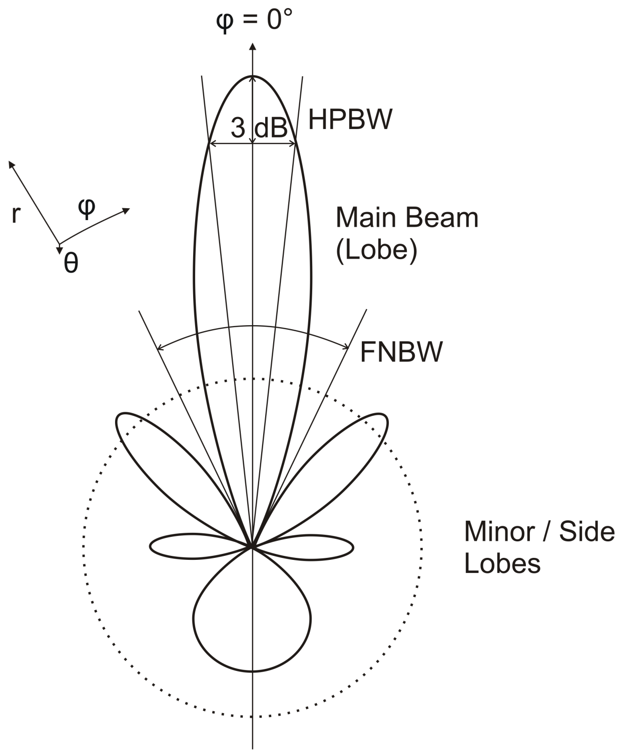

Figure 4.1: Radiation Pattern

There are several possibilities to draw the pattern. Usually spherical coordinates are used, in figure 4.1 r, φ and θ. The graph shows either the field or the power strength. The strength is relative and can be referenced to either the maximum (top of main lobe) or an isotropic radiator (dotted line) that uniformly radiates the same power as the antenna. Thus the pattern of the isotropic radiator is identical to the average of the AUT.

Antennas often focus in one direction and therefore have a *main beam* or *main lobe* and several *minor* or *side lobes*.

Two parameters help to describe the main lobe, both shown in figure 4.1. We reference the power to the maximum and keep in mind that half power is equivalent to 3 dB down (see chapter 2). The ***H****alf* ***P****ower* ***B****eam* ***W****idth* is the width around the maximum of the main lobe up to $P_{max} - 3\,\text{dB}$. The other parameter is the ***F****irst* ***N****ulls* ***B****eam* ***W****idth* or ***B****eam* ***W****idth between* ***F****irst* ***N****ulls*.

The radiation of an isotropic radiator uniformly propagates on a sphere. Thus, if P is the radiated power then the power density of an isotropic radiator is $S_i = \frac{P}{4\pi r^2}$, where $4\pi r^2$ describes the surface the sphere.

The *Directivity* of an antenna is defined as ratio of its maximum power density and the average power density, which is equivalent to the S_i:

$$D := \frac{S_r(\varphi,\, \theta)}{S_i} = 4\pi r^2 \frac{S_r(\varphi,\, \theta)}{P}$$

If a certain power P_S is fed from a source into the antenna only a part is radiated. Some of the power is dissipated in the antenna, i.e. turned into heat. Furthermore the input impedance of the antenna structure is typically significant different from the connected devices. As described in chapter 2 this leads to a partly reflection of the source power. To minimize the reflections an impedance matching between antenna and source has to be introduced. The same is true in the receive case, where the power would be reflected back into the antenna. To match antennas many different methods are being used: transformers, RLC networks, antenna geometry and many others.
We now can define an efficiency factor as ratio between radiated and fed in power

$$\eta = \frac{P}{P_S}$$

where the source power P_S is the sum of radiated, dissipated and reflected power.

Another important characterization is the *Gain*, defined as

$$G := \eta D = 4\pi r^2 \frac{S_r(\varphi,\, \theta)}{P_S}$$

Thus the gain compares the maximal power density of the AUT with an ideal (lossless and reflection free), isotropic radiator.
As discussed in chapter 2 often the reference is reflected in the decibel unit name, here termed dBi (for **i**sotropic).

4.1.4 Reciprocity Theorem

The reciprocity theorem states that all antenna characteristics are the same for receiving or transmitting. This can be very helpful. For example if we use an antenna as transmitter and measure the radiation pattern as shown in figure 4.1. When we now use it as receiver and apply the pattern we can predict how the received signal strength varies while the transmitter is varying the angular position from the antenna point of view. Another, obviously correct example would be, if we work out how a parabolic reflector focuses incoming parallel beams, we can immediately be sure, that it will send out parallel beams, if we put a radiator into the focal point.
The theorem can also be stated in a different form: An antenna A radiates power p_A while a second antenna B is receiving power p_B. If now antenna B will radiate power p_B then antenna A will receive power p_A.

4.2 The SRT Antenna

As shown in figure 3.1 the SRT antenna consists of two parts that are

- a parabolic reflector also called (parabolic) dish to focus the radiation

- a little antenna termed *feed* at the focal point of the reflector

Furthermore the whole antenna can be rotated to point to the astrophysical object that shall be observed. The electronic rotator can be controlled via the PC.

4.2.1 The Reflector

The reflector was bought from RF Hamdesign [RFham] in The Netherlands. It also provides the mechanics that holds the feed at the focal point.
The dish has a diameter $D = 2.4\,\mathrm{m}$, the focus is centred and at a distance of 0.96 m to the vertex. Thus the feed sees the reflector within an angle of $2\arctan(1.2/0.96) \approx 103°$. The reflector consists of an aluminium wire mesh. As discussed in chapter 2 it is important that the device size, here the square hole diameter of the mesh, is smaller than the wavelength. For the SRT we are using a hole diameter of 6 mm compared to the wavelength of approximately 210 mm. Of course the whole dish is significantly larger than the wavelength.
In a professional design it is possible to measure the dish characteristic or estimate it taking into account the size, wavelength, dish shape, accurate surface shape, electric losses, shading due to the feed, etc. This is also needed if an absolute power shall be measured.

4.2.2 The Feed

The Haystack Observatory chose a helical antenna as shown in figure 4.2.

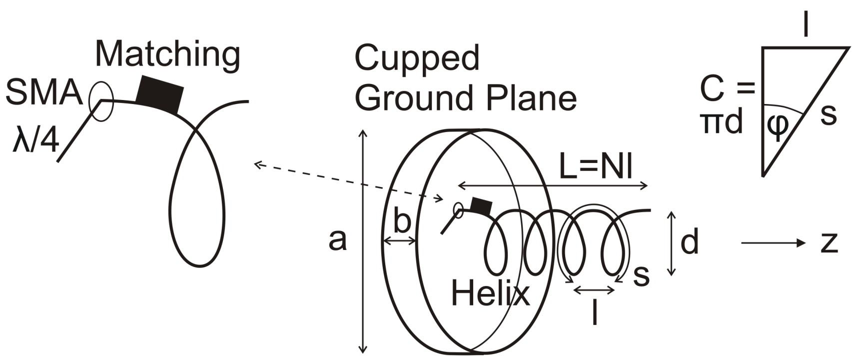

Figure 4.2: Helical Antenna with N windings

We can easily recognize several geometric relationships:

Diameter

$$d = C/\pi$$

Length of one winding

$$l = C\tan\varphi$$

Arc-length of one winding

$$s = \sqrt{C^2 + l^2}$$

Length of helix

$$L = Nl$$

where the width of the conductor was neglected.

Our goal is to observe radiation with a wavelength

$$\lambda \approx 21\,\text{cm}$$

A helical antenna can be designed for two different modes:

- *Normal Mode*
  When operating in normal mode the radiation pattern is maximal perpendicular to the helix (here the z-axis), and minimal along the axis. We do not use this mode here or discuss it any farther.

- *Axial* or *End-fire Mode*
  When operating in axial or end-fire mode the maximum of the radiation pattern is along the axis. Typically the helix is perpendicular mounted on a reflecting, often cupped ground plane. The ground plane prevents side and back slobes.

- Remark: Also mixed modes can be implemented.

Prof. Dr. Kraus describes in his book [KraMar] many more details around the helical antenna including also some of its history and applications. He further states:

> "The monofilar axial-mode helical antenna is very noncritical and one of the easiest of all antennas. Nevertheless, attention to details can maximize its performance."

The helical antenna is best used for elliptical (circular) polarized radiation, where the orientation is given by the orientation of the windings. For linear polarisation the received / transmitted power is halved, i.e. decreased by 3 dB.
Helical antennas are typically quite broad band.

The impedance of our receiver is 50 Ohm. We discussed that antennas usually have very different impedances, which shall be matched to avoid reflections and the loss of power. The Haystack feed uses therefore a little metal plate as shown in figure 4.2.

Unfortunately at the beginning of the thesis we did not have the same materials and with same sizes available as the Haystack design uses.
For the ground plane a cake pan (yes, the bakeware) was used, where we have

- Haystack: $a = 7\,\text{inch} \approx 17.8\,\text{cm}$
  Our cake pan: $a \approx 20\,\text{cm}$

- Haystack: $b = 3\,\text{inch} \approx 7.6\,\text{cm}$
  Our cake pan: $b \approx 7.5\,\text{cm}$

For the helix Haystack uses

- $N = 2$ windings

- Conductor: copper, width 4 mm; wrapped around a foam cylinder (rod)

- $d = 2.5\,\text{inch} \approx 6.35\,\text{cm}$
- $l = 2.997\,\text{cm}$
- $\varphi = \arctan \frac{l}{\pi d} \approx 8.61°$

The Haystack team followed mainly the article by K. M. Keen [Keen]. The antenna in this article is designed for 1590 MHz, that is a wavelength of approximately 18.8 cm. Keen gives the important design parameters a, b and d as functions of the wavelength, and uses two windings. Plugging in our 21 cm into the equations does not exactly result in the Haystack values. Presumably the sizes were further adjusted. An adjustment can be necessary to compensate variations of the conductor width, rod material (see also chapter 2, transmission lines and dielectric media), geometry of ground plane or different frequency.

Kraus and Marhefka [KraMar] give equations to calculate helical antennas. These equations are only valid within certain constraints, which are

$$N \geq 4$$

$$12° \leq \varphi \leq 14°$$

$$0.8 \leq C/\lambda \leq 1.2$$

Within above constraints following relationships are valid:

Half power band width

$$\text{HPBW} = \frac{52\sqrt{\lambda^3}}{C\sqrt{Nl}}$$

Band width between first nulls

$$\text{FNBW} = \frac{115\sqrt{\lambda^3}}{C\sqrt{Nl}}$$

For a helix with peripheral feed [2] input impedance is approximately:

$$R \approx \frac{150}{\sqrt{C/\lambda}}$$

The Geometry of the ground plane is typically given as estimation, rather than exact values, see also [Kark].

Above equations suggests that

- the slightly different cake pan is not critical for the performance

[2] Peripheral feed means that the feeding of the antenna is located underneath the windings – in opposite to an axial feed, where the conductor at the end is lead to the axis of the helix, i.e. the connection is on the helix axis.

- the smaller rod diameter slightly increases the HPBW and the input impedance, which is anyway compensated by matching
- d, l and φ are related as $l/(\pi d) = \tan\varphi$; but the performance seems to depend on d and l where ϕ follows.

Thus for our helix were chosen

- $N = 2$ windings
- Conductor: copper, width 4 mm; wrapped around a foam cylinder (rod)
- $l = 2.997$ cm as Haystack but
- $d = 6$ cm and
- $\varphi = \arctan \frac{l}{\pi d} \approx 9.1°$

But there was a high ambition to reach the best performance and to learn more about helical antennas. Therefore it was decided to build a second antenna closely following the Haystack design.
Finally we found a shop that was selling aluminium pans with the same size as the Haystack one, i.e. 7 inches diameter and 3 inches deep.
Further two shops, Rechinger / Deko Figuren Shop and Sansys, offered to produce rods with a 2.5 inches diameter. The rods were made of Styropor (by BASF), with a density of 30 kg/m^3. According to BASF this material is a very good isolator with a dielectric constant $\epsilon_r < 1.04$. Thus it behaves almost identical to air. Furthermore Styropor is made of Polystyrene just as Styrofoam, which was used by Haystack.

Assembly and Tuning

First two holes were drilled into the ground plane, i.e. the cake pan. The first hole is located in the centre to mount the rod and the PCB between rod and GND with a thin (5 mm), long (100 mm) metal screw. The second hole shall contain the SMA feed-trough. The feed-through is a connector that provides SMA on the back of the ground plane. The opposite end is a pin to which the helix is being soldered. Therefore the hole was simply positioned at a distance of rod-radius plus SMA-connector-radius. For the second antenna the distance between the holes was with 37.7 mm (Haystack desing) slightly higher.
Now a 74 mm long rod was cut off a long foam cylinder, where for the second antenna the Styropor rod was available. A long hole with 5 mm diameter is needed for the screw that fixes the rod to the pan. It is easily possible to drill a hole into the Styropor with a drill used for metal.
Further a 60 mm times 60 mm PCB was sawed out of a blank, thin PCB. And also in the centre of the PCB a 5 mm hole was drilled.
The triangle shown in figure 4.2 was cut out of a piece of paper. It had an angle $\varphi = 9.1°$ and $\varphi = 8.6°$ for the second antenna. The paper triangle was long enough to wrap it two times around the rod. This was used to draw the position of the conductor on the cylinder.
Now the conductor was cut out of a copper tape.
As discussed the Haystack team used a little metal plate, this was soldered to the conductor after gluing the conductor on the rod.

For our antenna the matching became immediately a part of the conductor by cutting the strip as in figure 4.3.

Figure 4.3: Feed Conductor

The Haystack geometry was (with b-d as separated part)

- a = 4 mm
- b = 13 mm
- c = 439 mm
- d = 26.25 mm
- e = 13 mm
- f = 15 mm

The conductor is wrapped around the rod.
The dotted lines mark the bending positions:

- part f is leading from the rod to the feed-through, where it is soldered to the pin of the connector
- part b-d serves as matching

It was expected that the matching must be optimized for our design.
Therefore a conductor was cut out but with longer b, c, d and f.
The backing was removed and the conductor was glued around the rod. It starts at the bottom but almost 1 mm above the triangle which it follows with this distance. After every winding the strip must be 3 mm above the previous. Part f was bended to the connector, thus the starting point of the conductor on the rod was the e-f bending line. The end of the conductor must be after 2 windings plus 15 mm. The matching area was bended as well.
Finally the SMA feed through was mounted, the PCB and the rod were fixed with a screw and the conductor was soldered to the connector.

After the assembly the S_{11} parameter (i.e. the reflection) was measured, please see the following section for details.
At the beginning the reflection was rather high. Now all kind of modifications were tried to get to know the antenna and to optimize for lowest reflections. It turned out that Prof. Kraus was right, the antenna is fairly robust against most parameters.
Rod diameter, size of cake pan, width of copper, removing metal screw or angle of windings were not very critical for the performance. But the performance strongly depends on the

size of the matching plate, its bending angle and its distance to the beginning of the antenna, i.e. length e (f).
As mentioned, b, c, d and f were cut out longer. These sizes could be adjusted by cutting. Even e and f could be modified by bending the inductor and rotating the rod to bring the end of the helix closer to or farther away from the SMA connector.

After all investigations the helix was unsoldered and unwrapped to serve as template for the final one.
All steps above where repeated but now only very little modifications were necessary, like mainly bending the matching.

The copper tape does not stick to the Styropor very strongly. For most parts of the Helix this is fine, since it gets additional support from the wrapping. But the end can get loose. Hence an additional glue was used at this point. This glue was a universal, fluid glue that can be found in many shops, but has to be use-able for Styropor. It turned out that the glue did not influence the performance.

A further problem is to protect the LNA from high currents like electrostatic discharge or even lightning. Even if lightnings don't directly hit the Antenna they can induct a high current. Therefore Haystack suggested a semi-rigid coax cable with the length of $\lambda/4$, which is soldered to the SMA pin. The other end of the cable is soldered to GND. Thus the cable is a short for DC. But as described in chapter 2 the quarter wavelength transmission line transforms the short into an open, therefore the cable does not influence the received signal at a frequency of around 1420.4 GHz.
Unfortunately here several problems occurred.
It turned out to be difficult to solder something to the cake pan. Maybe the cake pan is coated with some protection and aluminium is not easy to solder. Furthermore the solder equipment that is optimized for microscopic devices had difficulties to heat the rather huge ground plane.
Another problem is to find the right length. As discussed in chapter 2, transmission lines, cables are typically filled with a dielectric media to support the conductor. The media shortens the wavelength by a factor $\frac{1}{\sqrt{\epsilon}}$. Therefore the correct length has to be found. Since the protection is applied at a very sensitive point this should be done as precisely as possible. Even if the ϵ of the media is known a pure calculation might not lead to a perfect result. Therefore several trials with different semi rigid cables and lengths might be necessary.
Finally a semi rigid was found that gave a good performance. It was soldered to the helix and the SMA connector; at the other end the conductor was soldered to the shielding and the semi rigid was fixed with the copper tape to the cake pan. Another alternative to soldering to the pan is to drill two holes in the pan on both sides of the semi rigid, put one wire through the holes and loop it around the semi rigid to fix it inclusive a short to GND (pan).

Both antennas showed similar performance, but the second one was slightly better. Thus this one was chosen as main antenna. The first one can serve as fall back solution and for measurements, please see below.
Pictures 4.4 and 4.5 show the final feed.

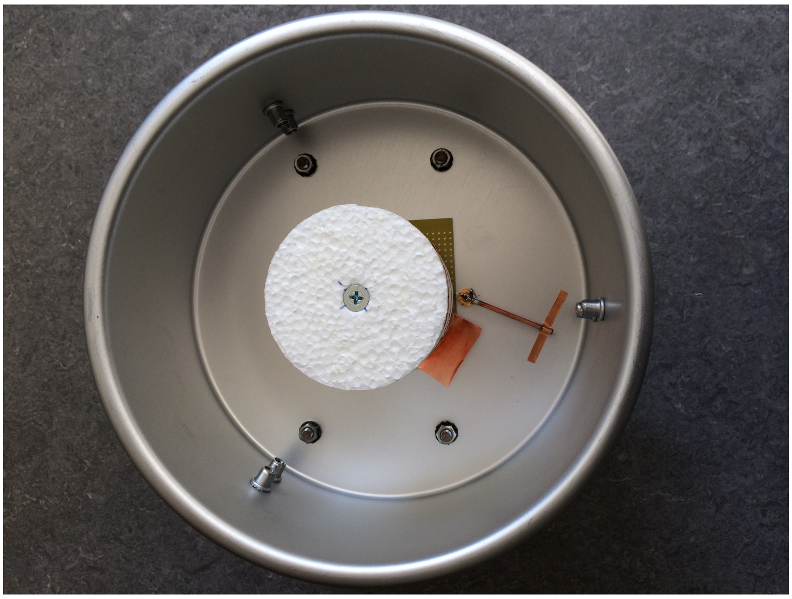

Figure 4.4: Feed

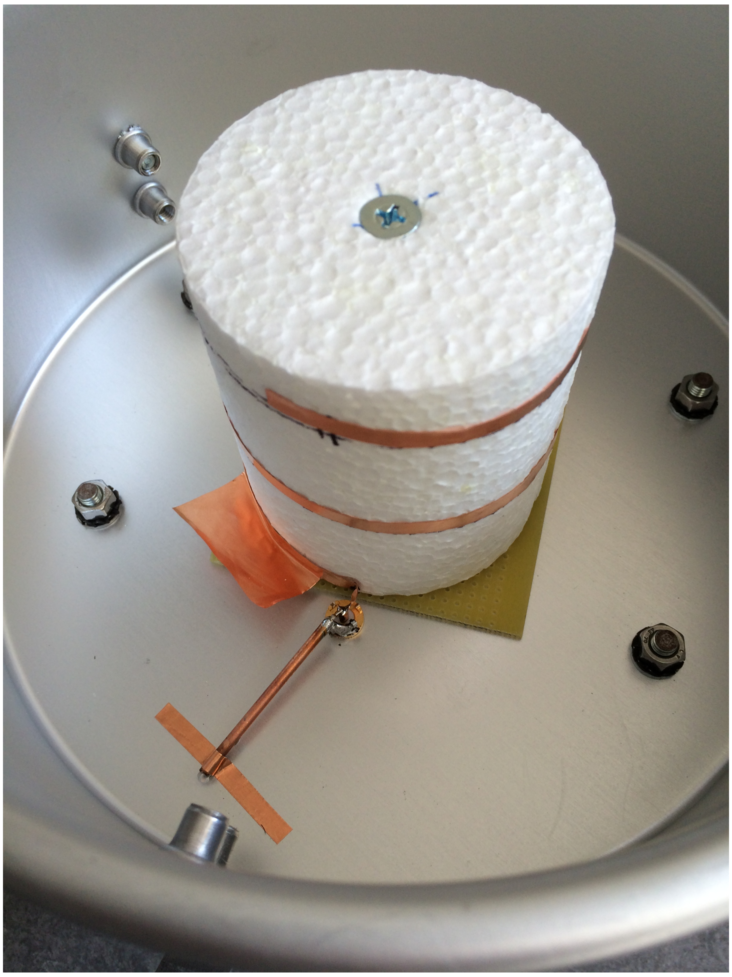

Figure 4.5: Feed

Matching and Measurement Results

Intel labs provided a ***N****etwork* ***A****analyzer* from Rohde & Schwarz as described in chapter 2, scattering parameters. With the NA it was possible to measure the input impedance of the feed as well as the S_{11} parameter, i.e. the reflection in range between 0.5 GHz and 2.5 GHz.
Figure 4.6 shows the final result after optimizing the performance as described in the previous section, inclusive the semi-rigid.

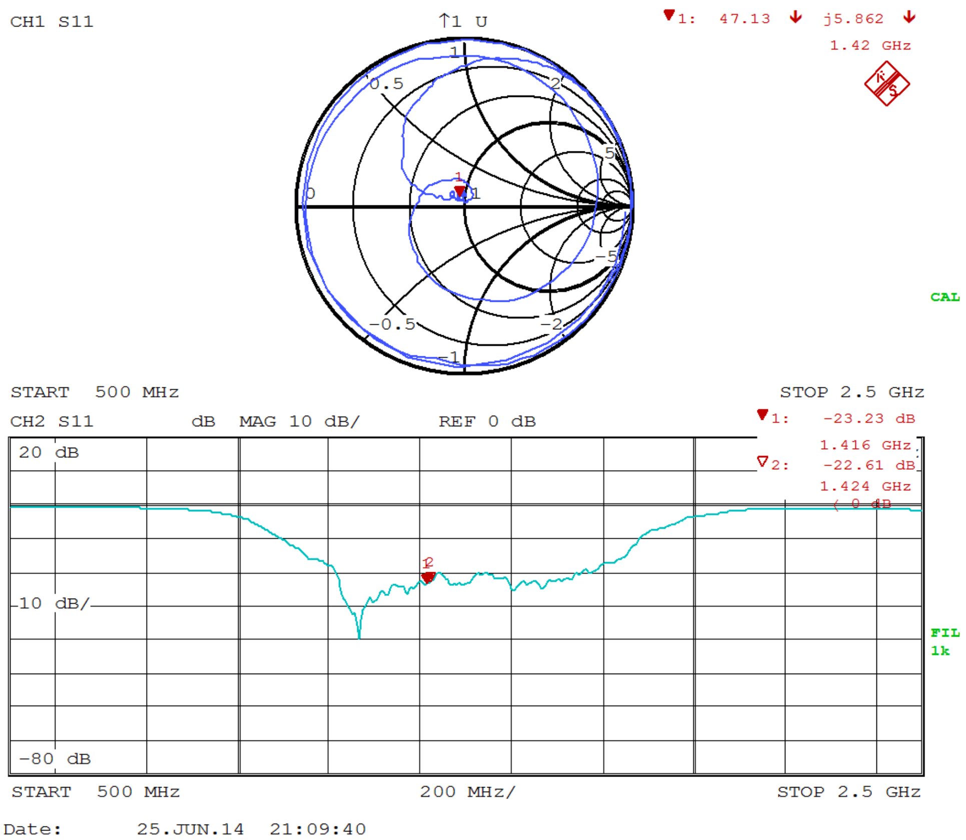

Figure 4.6: Feed S_{11}

The Network Analyzer plots in the top screen the impedance and a Smith chart (see chapter 2 transmission lines). The impedance is $47.1 + i5.9$ Ohm at 1.420 GHz and thus almost in the centre of the chart.
The bottom screen shows the S_{11} parameter versus frequency, which is approximately of remarkable -23 dB between 1416 MHz and 1424 MHz, and less than -20 dB in a broad band. At higher offsets it goes up to zero, which means total reflection, i.e. the antenna is far away from its working point.

The S_{11} parameter was strongly reacting, when the antenna was covered by hand. This indicates that the antenna is well receiving and transmitting.
As discussed an antenna chamber was not available. But meanwhile two similar antennas were built, furthermore Rohde & Schwarz provided an SME02 Signal Generator and an FSW13 Spectrum Analyzer for the USM labs. Together with Arno and Vanessa it was tried to check the transmission performance as well. Therefore both antennas were

mounted at a distance of more than four meters (trying to avoid the near fields), both in a hight of approximately one meter. The antennas were pointing towards each other. The first antenna was connected to the Signal Generator, which provided a CW signal at 14020.4 MHz, and served as transmitter. The second antenna (our main one) was connected to the Spectrum Analyzer and was the receiver, as later in the telescope.
Now the receiving antenna was rotated to the left and the right and showed 3 dB less power at approximately ±30°. Thus the HPBW is approximately 60°. We already saw that the antenna sees the 2.4 m reflector within an angle of 103°. The feed should not cover the edges of the dish to avoid reflections and interference. Thus the 60° HPWB seems to be good match to the 103° dish.
Antenna measurements outside an anechoic chamber are not reliable due to reflections. But since the lab is very unsymmetric, with even windows on one side, but the measurement results were very symmetric, we gained confidence, that the ±30° were close to the truth.

Feed and LNA

The LNA was mounted on the back of the ground plane, similar as suggested from Haystack. This combination was mounted at the parabolic dish with three arms and the help of a simple but smart mechanic that allows to adjust the position and the angle of the feed. The centre of the Helix is located at the focal point, i.e. approximately 96 cm away from the dish. The mechanic was implemented by Ady, so only the best position and orientation of the feed were left to be found. The following pictures show result.

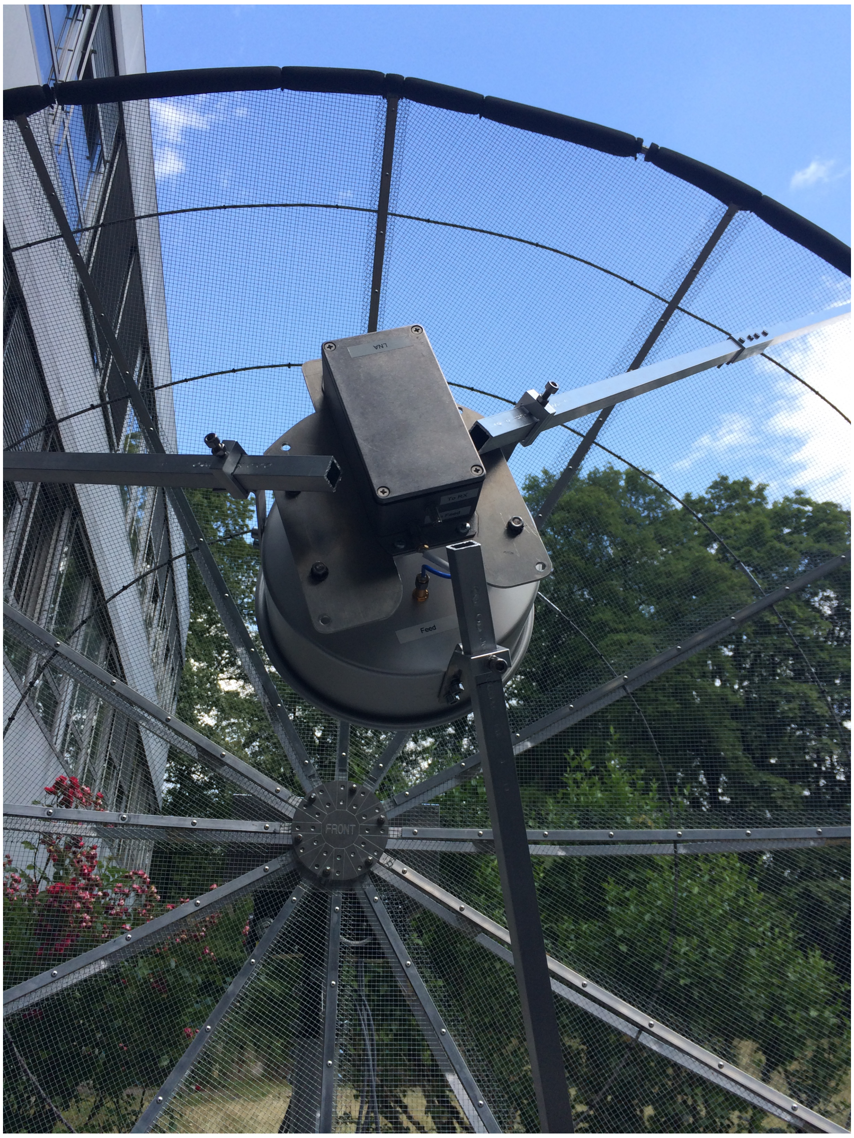

Figure 4.7: Feed with LNA and dish

Figure 4.8: Feed with LNA and dish

Figure 4.9: Feed with LNA

The Antenna

The following pictures show the complete antenna.

Figure 4.10: The SRT Antenna

Figure 4.11: The SRT Antenna

Figure 4.12: The SRT Antenna

Chapter 5

The Receiver

5.1 The Design

The receiver chain is based on the Haystack design, shown in picture 5.1.

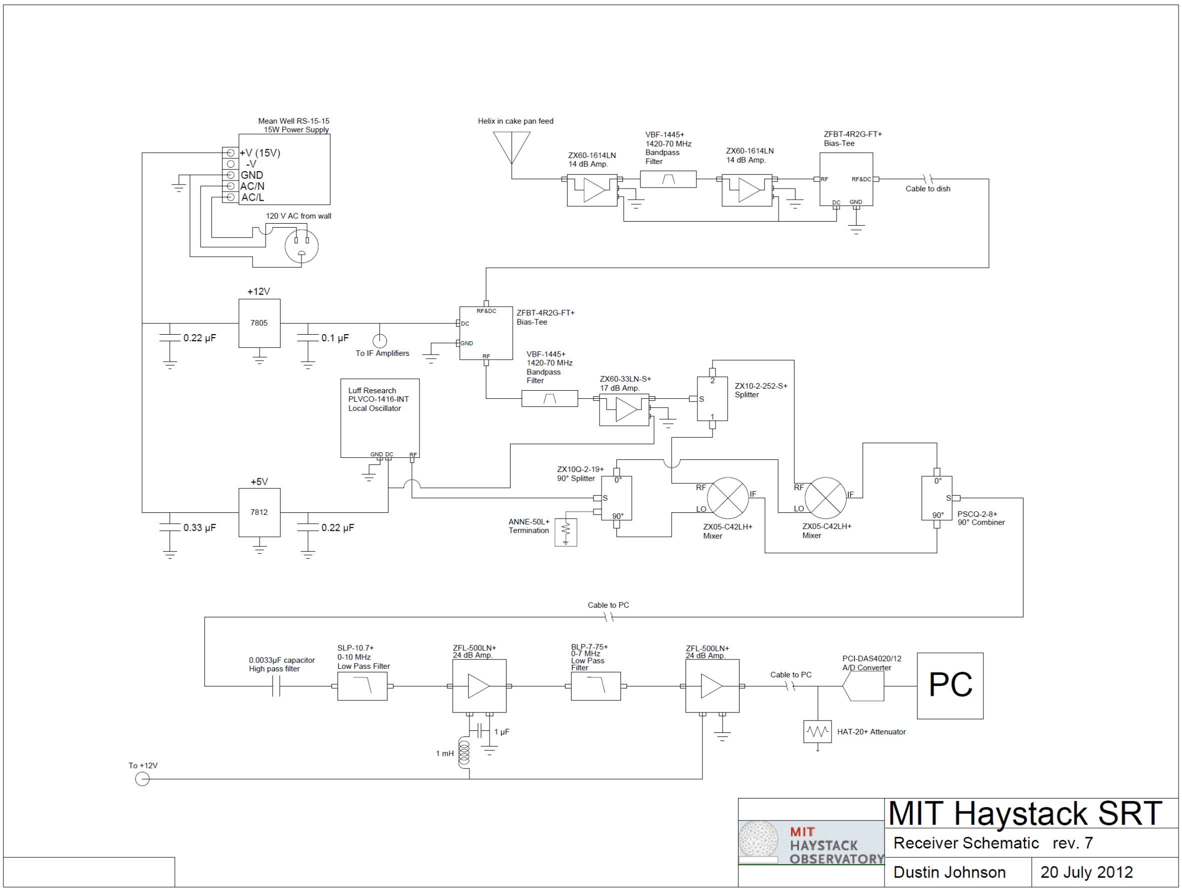

Figure 5.1: Electrical Schematic

We want to observe a frequency band that is located around 1420.4 MHz. This band shall be amplified, filtered and mixed down to 5 Mhz to digitize it and further process it. Therefore the output of the chain is connected to a PC that comprises an ***A**nalog to*

D*igital* ***C****onverter*. On the input side the chain is connected to the antenna. Antenna and PC can be separated by several meters.

The chain is a typical receiver, as described in chapter 2.
To achieve the best (noise) performance a ***L****ow* ***N****oise* ***A****mplifier* shall be implemented early in the chain and as close as possible to the antenna.

Thus the receiver chain starts with an LNA. Then undesired signals, which are also received by the antenna, are filtered with a ***B****and* ***P****ass* ***F****ilter*. Another LNA follows the BPF.
This first stage is combined in a weather proved, metal shielded case and directly mounted on the feed. Also the complete stage is referred to as LNA.

With this construction the LNA is implemented before the long cable, which separates PC and antenna. We chose a low loss cable for outdoor usage. In the picture it is labelled as "Cable to dish".

Two more stages follow:

Next is the down conversion, which is implemented as image canceling mixer.
The synthesizer serves as ***L****ocal* ***O****scillator* with $f_{LO} = 1416\,\text{MHz}$. The power combiners and splitters also partly work as 90° phase shifter.

The last stage amplifies and filters the signal one more time.
First a capacitor serves as simple HPF which is filtering DC that can be generated by the mixer.
Since we mix down our band of interest to 5 Mhz this range shall not be attenuated. But, as described in the chapter about digital signal processing, from 10 MHz on we need good attenuation. Therefore LPFs can be used.
Inexpensive filters with such a frequency response are rare. Hence, the Haystack team chose for one of these low pass filters the BLP-7-75+ from Mini-Circuits. Unfortunately this is a 75 Ohm filter.
Of course a 75 Ohm filter in the 50 Ohm system would lead to reflections – and we tried to improve this.
After an intensive search KR-Electronics [KRF] in the USA made an offer for a customized filter. We took the BLP-7-75 as reference for the specifications, but with 50 Ohm impedance. In addition we chose SMA male connectors, since the two neighbouring amplifiers in the chain have SMA female. The offer was with $145 at a reasonable prize that includes shipping only one week after order.

The amplifiers, synthesizer etc. need power supplies, some 5, some 12 Volt. Therefore a power supply is included which supplies two voltage regulators. They produce the desired 5 V and 12 V and in turn supply the devices. The voltage regulators are stabilized with capacitors, similar as their data sheets suggest. Furthermore at seemingly sensitive points some blocking with capacitors and inductors is implemented.

The LNA does not have its own supply. The supply for the LNA is provided via the RF cables. To combine and separate RF and DC on one cable Bias Tees are used.

These two stages (mixing and amplifying / filtering) together with the power supply are combined in another metal shielded box.

At this point the signal is ready to get digitized. The cable, which leads the signal now to the PC, is in the picture labelled as “Cable to PC”.

The ADC has a high input impedance of 1.5 MOhm. But the system is a 50 Ohm system, thus a 50 Ohm termination is needed – otherwise the signal would be reflected.
This termination is the “HAT-20+ Attenuator”, that serves as sink at which the ADC measures the voltage. The 1.5 MOhm in parallel to 50 Ohm gives an effective termination of $R = \frac{50 \cdot 1.5\,\mathrm{M}}{50 + 1.5\,\mathrm{M}}\,\mathrm{Ohm} \approx 49.998\,\mathrm{Ohm}$, which still leads to a very good performance.

Remarks

Like in the Haystack design the PSCQ-2-8+ 90° combiner together with the 0.0033 µF capacitor were combined and soldered in a dedicated metal case, which provides also BNC connectors.

The Haystack parts list contained three SMA male-male connectors, but actually 4 are needed.

Furthermore capacitors can be made of several materials. For RF Blocking ceramic capacitors are the best choice. Ceramic is also the recommendation in the data sheet for the voltage regulators.

Haystack also offered a detailed mechanical description including hole positions etc. But it is not necessary to follow the plan. In addition problems can occur, like sizes of devices might be different or locally other norms (e.g. for screws) are used. Instead we simply put the electronics together inside the case and mounted it just where it happened to be.

The Mean Well RS-15-15 power supply can also be used with the German system, i.e. 230 V and 50 Hz.

All data sheets can easily be found and downloaded in the internet – except the one of the customized filter. Therefore its performance plots are included at the end of this chapter.

The final receiver chain hardware

Figure 5.2 shows the LNA stage with the two amplifiers and the band pass filter between them on the bottom, on top is the Bias Tee that separates the RF signal and the DC supply for the amplifiers.

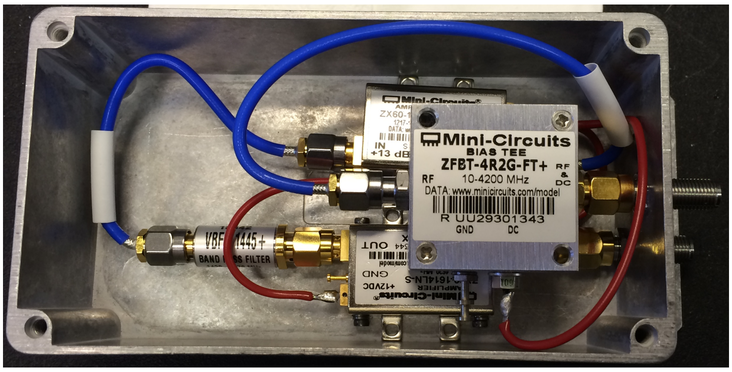

Figure 5.2: Low Noise Amplifier Stage

Figure 5.3 shows the receiver chain. On the upper part is the KR filter visible that replaces the 75 Ohm one.

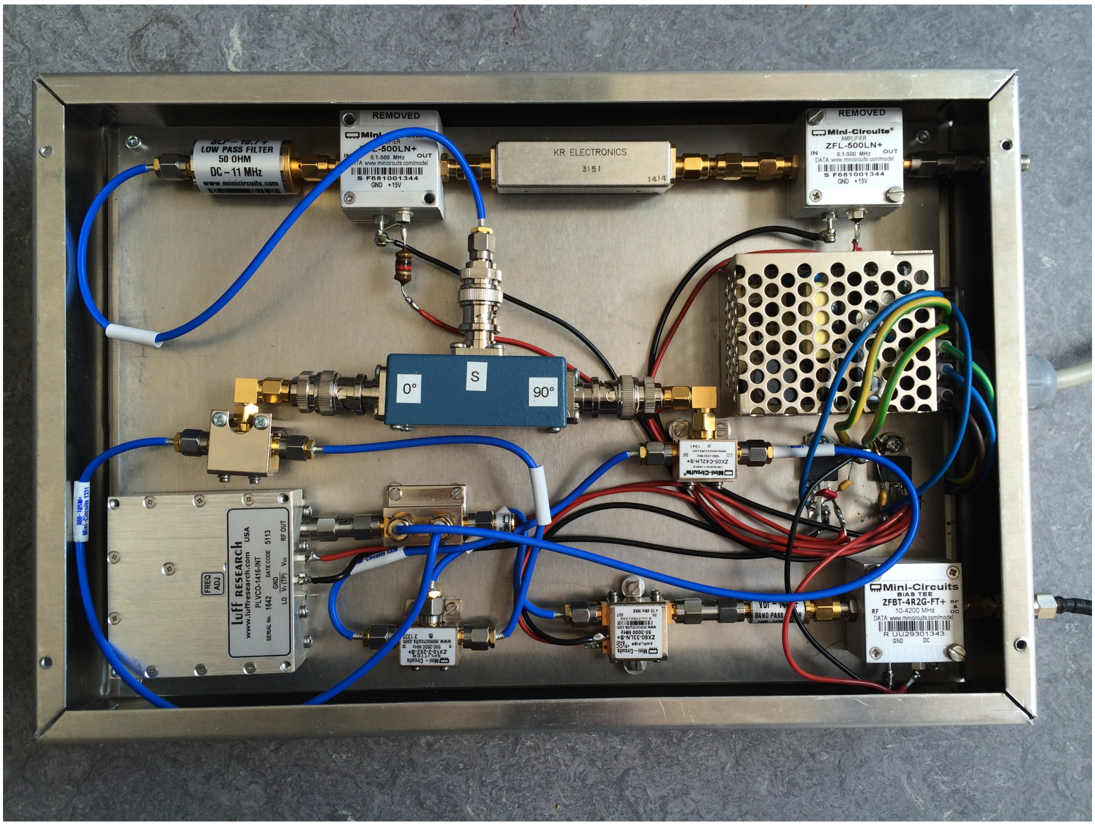

Figure 5.3: The Receiver

5.2 Performance Measurements

The verification was done at the INTEL Labs, where INTEL provided the necessary equipment. This included the Rohde & Schwarz Spectrum Analyzer *FSW*, the Rohde & Schwarz Signal Generator *SMJ100A*, the LeCroy Oscilloscope *wavepro 7200*, the Keithley Digital Multi Meter *2000*, an Agilent power supply and a Mini-Circuits Bias Tee.

Supplies

The output voltages of the power supply, voltage regulators and Bias Tees were verified with the Digital Multi Meter, summary: the two main domains showed 5 V and 12 V as desired.

General Measurement Setup

Figure 5.4 shows the general measurement setup.

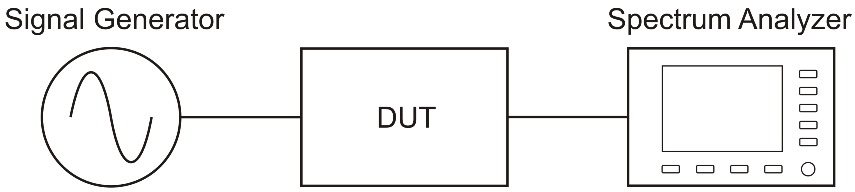

Figure 5.4: General Measurement Setup

The Signal Generator serves as input signal for the ***D****evice* ***U****nder* ***T****est*. It generates a sinusoidal signal, which is termed ***C****ontinuous* ***W****ave*. This signal was swept in frequency and amplitude.

The output of the DUT was measured with the Spectrum Analyzer, the ***R****esolution* ***B****and* ***W****idth* was set to 100 kHz. The SA shows on the horizontal axis the frequency, on the vertical axis the output power with 10 dB/Division. Furthermore the SA offers two different kind of markers that can be set on the trace to read out certain points. One type of markers are the absolute markers called ‘M’, the other type are the delta markers ‘D’ that are referenced to an absolute marker. The marker readouts are shown in the top right corner of the screen and a list with all marker information is provided at the bottom.

To connect the Signal Generator and the Spectrum Analyzer with the DUT cables were used with an attenuation of 0.5 dB. The resulting attenuation of 1 dB was taken into account by setting an RF level offset in the Spectrum Analyzer.

The input signal of the receiver, with a frequency around 1420.4 Mhz, is mixed down with the LO at 1416 Mhz, to 1420.4 Mhz − 1416 = 4.4 Mhz. The LO as well as the measurement equipment (i.e. Signal Generator and Spectrum Analyzer) have both a frequency error. Thus the down converted signal may be measured at slightly different frequencies than 4.4 Mhz.

As discussed in chapter 2 problems can occur when using a down-conversion to zero. The

FSW Spectrum Analyzer also gives out warnings while measuring close to DC. Thus the measurements started with frequencies greater or equal 1 MHz.

For the time domain measurements the Spectrum Analyzer was replaced by the Oscilloscope.

In the following sections all measurement results are shown as screen-shots of the Spectrum Analyzer or the Oscilloscope.

5.2.1 Low Noise Amplifier

Measurement Setup

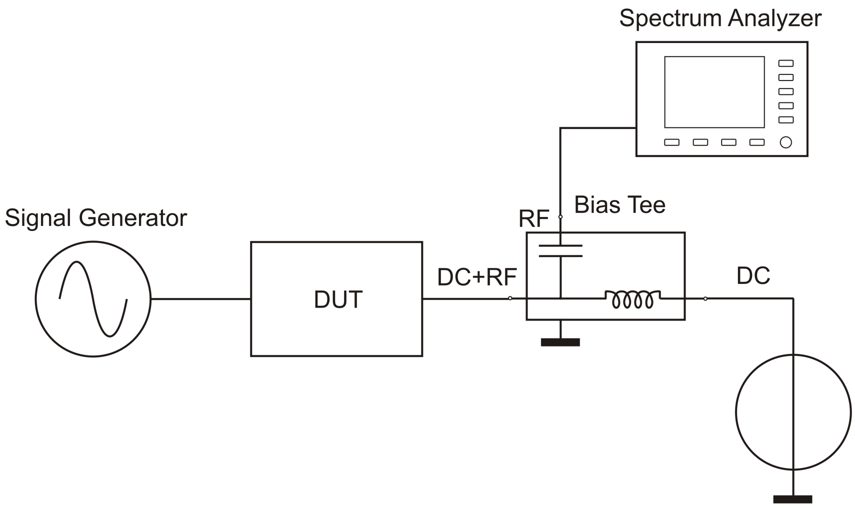

Figure 5.5: LNA Measurement Setup

The LNA comprises two amplifiers, which receive their 12 V supply externally via the cable and a Bias Tee. Therefore the setup was modified to feed in the 12 V with the help of an external supply. An external Bias Tee seperates DC and the measurement device, here the Spectrum Analyzer. Figure 5.5 shows the modified setup.
The Bias Tee had an attenuation of 1 dB around 1420.4 MHz. Thus, to calibrate the setup, the RF level offset of the spectrum analyzer was increased by 1 dB to 2 dB.

The power supply showed a current consumption of the LNA of 80 mA.

Frequency Response

A CW signal with -30 dBm was used for the transfer function. The frequency was swept from 1420.4 MHz − 1 GHz to 1420.4 MHz + 1 GHz.
Figure 5.6 shows the measurement result.
Around the pass band the BPF suppresses unwanted signals.
The Marker at $f = 1420.4\,\text{MHz}$ shows a power of $-5.7\,\text{dBm}$. Hence the amplification is approximately $-6\,\text{dBm} - (-30\,\text{dBm}) = 24\,\text{dB}$.

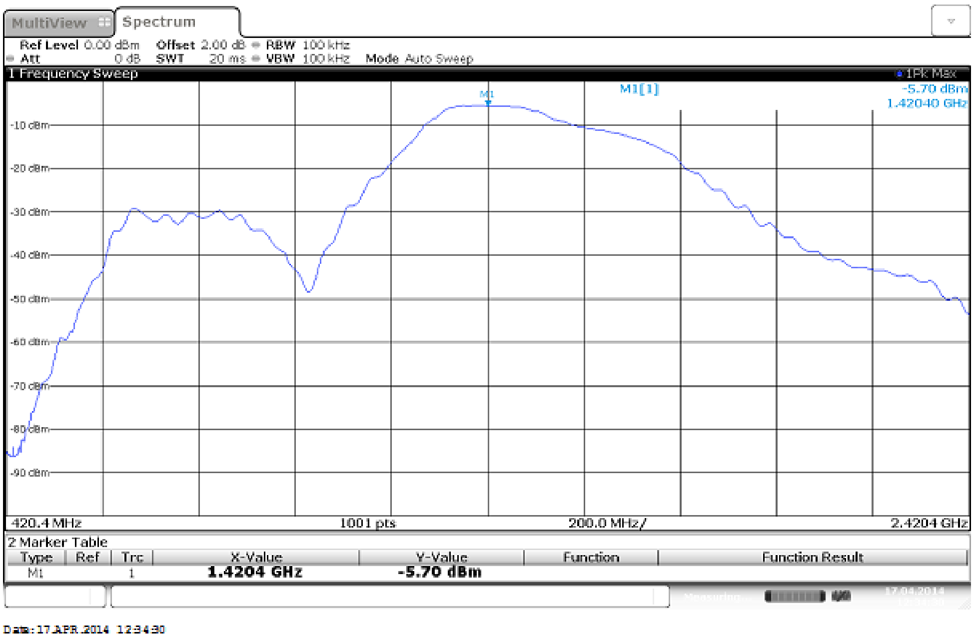

Figure 5.6: LNA Frequency Response

Dynamic Range

To find the 3 dB compression point the power was swept at $f = 1420.4$ MHz.

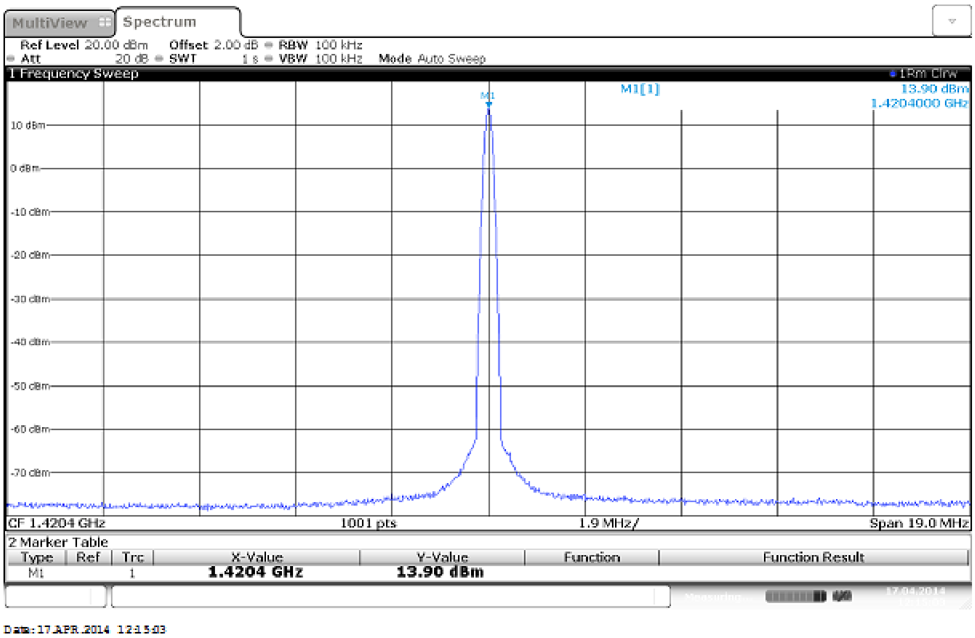

Figure 5.7: LNA Compression Point

When the input power (delivered by the signal generator) reached -7 dBm the output should be $-7\,\text{dBm} + 24\,\text{dB} = 17\,\text{dBm}$. But as showed in figure 5.7 the power was approximately 14 dBm, thus 3 dB too low.

5.2.2 Receiver without Low Noise Amplifier

In this section the receiver chain without the Low Noise Amplifier is simply called *receiver*.

Measurement Setup

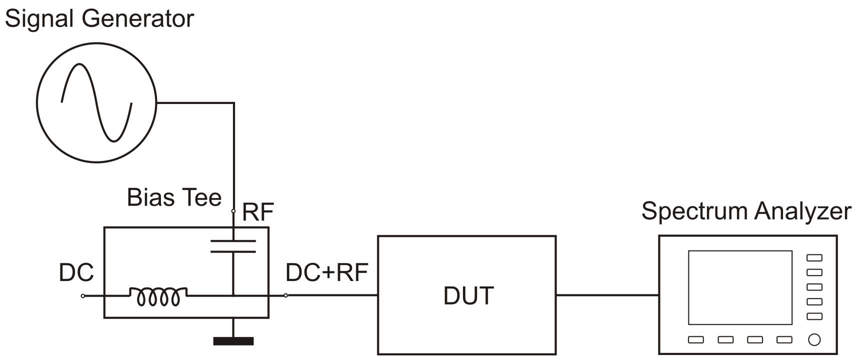

Figure 5.8: Receiver without LNA: Measurement Setup

The receiver delivers at its input the 12 V supply for the LNA. But the Signal Generator must not see a DC at its output. Thus the 12 V have to be blocked. This was achieved with the help of an external Bias Tee, shown in figure 5.8.
The Bias Tee has an attenuation of 1 dB for an RF signal around 1420.4 MHz. To calibrate the setup this time the 1 dB were not taken into account by the RF level offset of the Spectrum Analyzer. To measure the 3 dB compression point more precisely the 1 dB were always subtracted from the power that the Signal Generator delivered.

Frequency Response

First a Signal Generator power of -101 dBm was used, which results in an input power after the Bias Tee of -102 dBm. In the following only the corrected input power (here the -102 dBm) is stated, i.e. the Signal Generator power was always set 1 dB higher.
The frequency of the CW signal was 1420.4 MHz, which is mixed down to 1420.4 MHz $-$ 1416 MHz $\approx$ 4.4 MHz.
The result is shown in figure 5.9. Please also notice the marker readout at 4.412 MHz and 40 dBm.
The frequency response was measured in greater details in combination with the LNA, please see the following sections.
The amplification in the pass band is approximately $-102\,\text{dBm} - (-40\,\text{dBm}) = 62\,\text{dB}$.

Dynamic Range

To find the 3 dB compression point the power was swept at $f = 1420.4\,\text{MHz}$.

When the input power reached $-52\,\text{dBm}$ the output should be $-52\,\text{dBm} + 62\,\text{dB} = 10\,\text{dBm}$. But as showed in figure 5.10 the power was approximately 7 dBm, thus 3 dB too low.

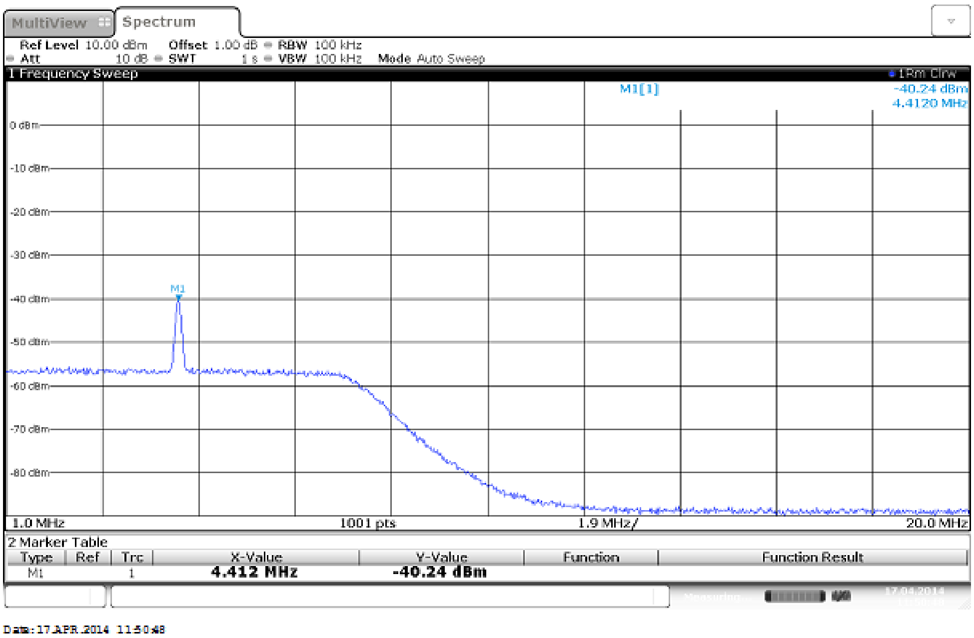

Figure 5.9: Receiver Frequency Response

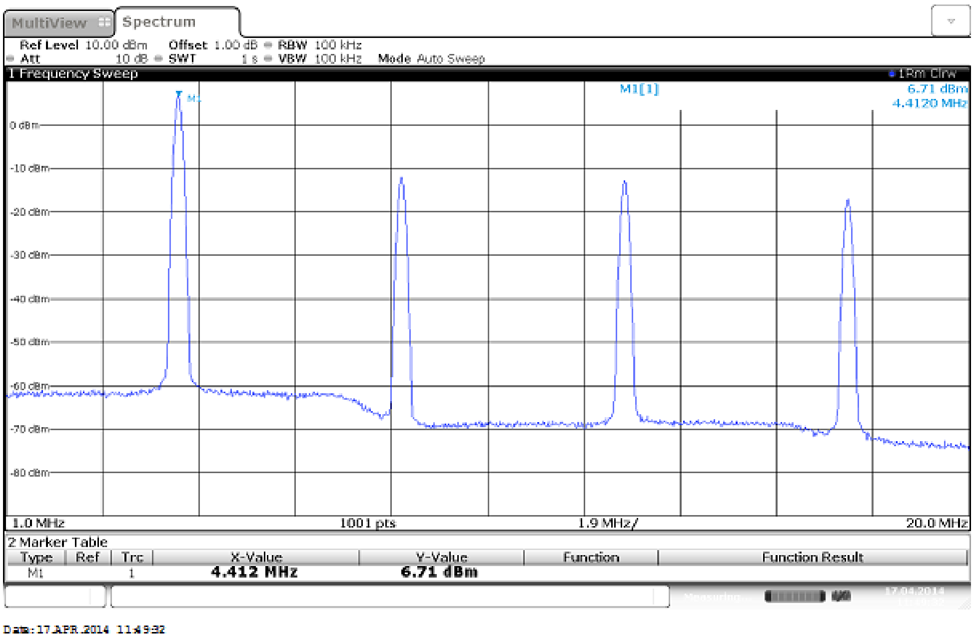

Figure 5.10: Receiver Compression Point

5.2.3 The Receiver Chain (*RX*)

Measurement Setup

In this chapter the LNA was directly connected to the receiver. The general measurement setup of figure 5.4 was used.

Frequency Response

First a Signal Generator power of $-86\,\text{dBm}$ was used.
The frequency of the CW signal was 1420.4 MHz, which is mixed down to $1420.4\,\text{MHz} - 1416\,\text{MHz} \approx 4.4\,\text{MHz}$.
The results are shown in figure 5.11. Please also note the marker readout at 4.412 MHz and $-0.13\,\text{dBm}$.

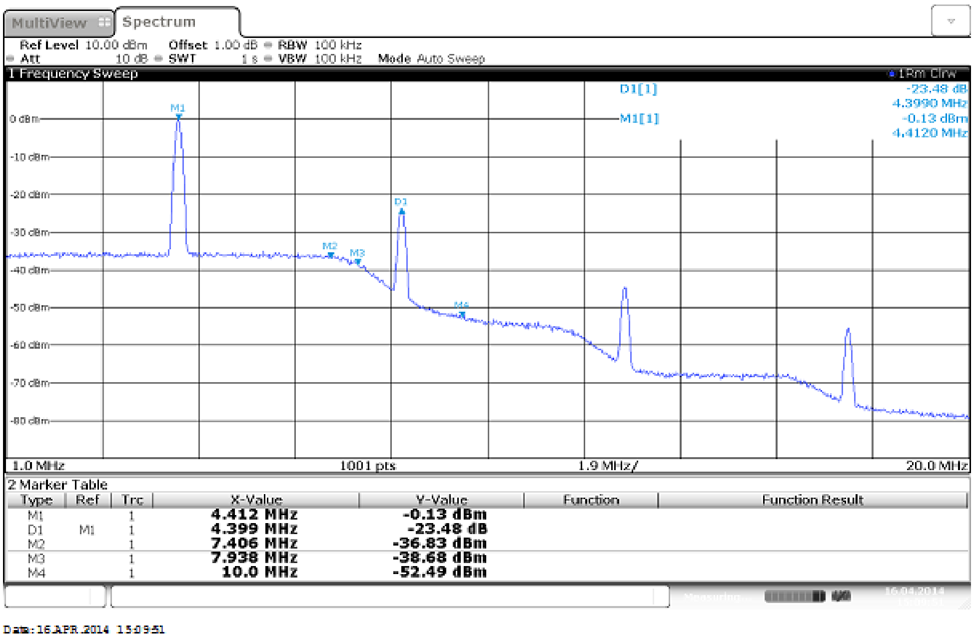

Figure 5.11: RX Frequency Response

The amplification in the pass band is approximately $-86\,\text{dBm} - (0\,\text{dBm}) = 86\,\text{dB}$. This is exactly the sum of the amplifications of LNA and receiver.
The attenuation of the pass band has a ripple of approximately 0.5 dB. In this sense the end of the band is shown with Marker2 at 7.406 MHz. Marker3 shows approximately the 3 dB edge at 7.938 MHz.
Now the frequency was changed from 1420.4 MHz to 1417.4 MHz and 1423.4 Mhz as showed in the figures 5.12 and 5.13.
Thus we can observe a bandwidth of 6 MHz around 1420.4 MHz or in other words: the observable band goes from 1417.4 MHz to 1423.4 Mhz.

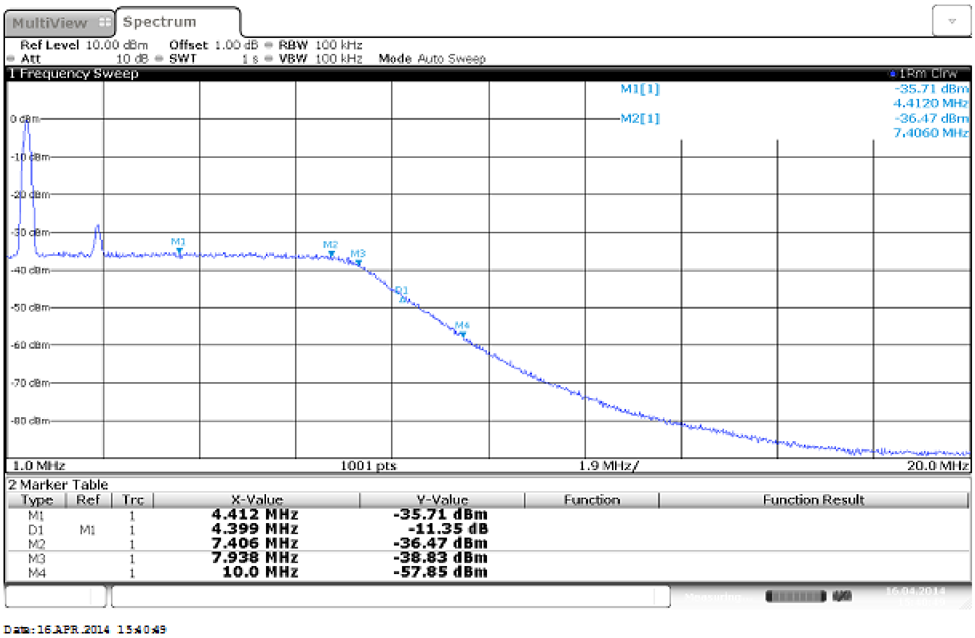

Figure 5.12: RX Frequency Response 1417.4 MHz

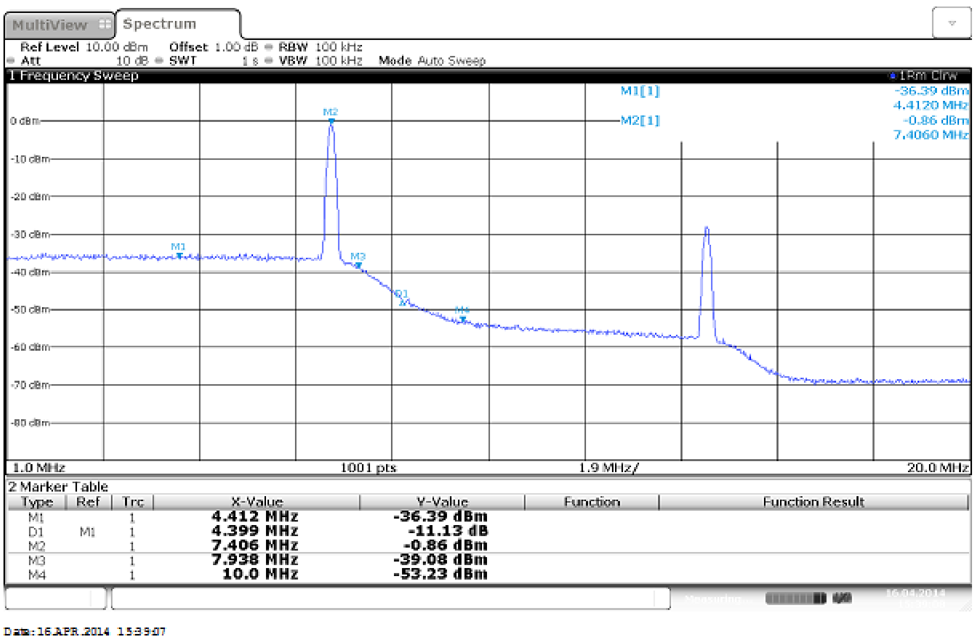

Figure 5.13: RX Frequency Response 1423.4 MHz

Image Canceling

To verify the image rejection the frequency was changed to the image frequency of $1416\,\text{MHz} - (1420.4 - 1416)\,\text{MHz} = 1416\,\text{MHz} - 4.4\,\text{MHz} = 1411.5\,\text{MHz}$.
The Marker in figure 5.14 shows the signal 33 dB lower than in figure 5.11. The image is suppressed with 33 dB compared to the wanted signal.

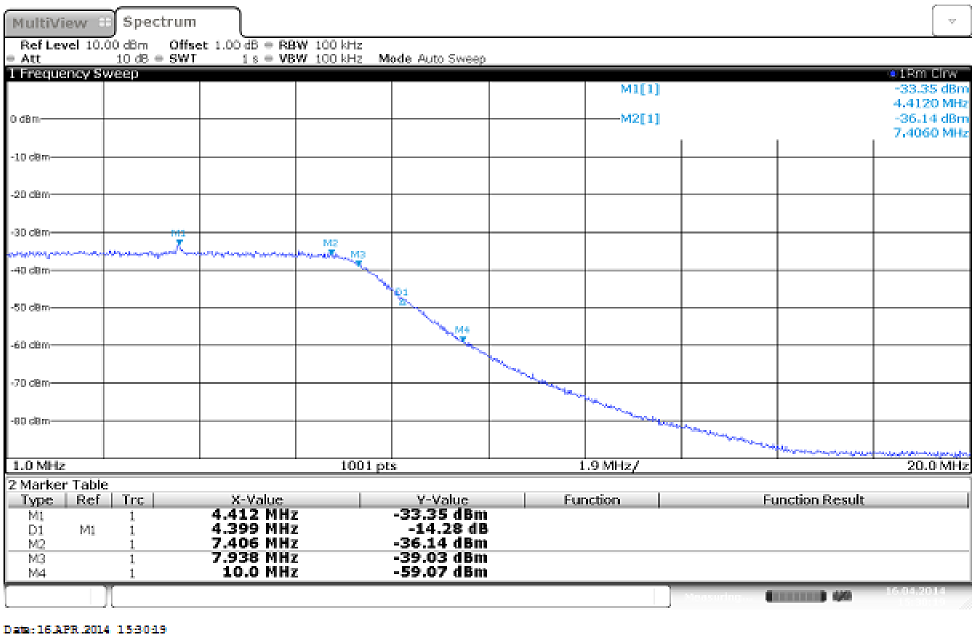

Figure 5.14: RX Image Rejection

Dynamic Range

Figure 5.11 shows also strong harmonics of the signal, please see Marker D1 and also the peaks at higher frequencies. As discussed in the chapter 2 this is typical for a non-linear device.
Indeed the chain starts to saturate at an input power of $-86\,\text{dBm}$.
Decreasing the input power by 1 dB also decreases the output power by 1 dB. A further decreasing of the input power, i.e. further going into the linear range, also improves the harmonic contents, as shown in picture 5.15.
While increasing the input power from -86 dBm onwards the output power does not follow any more. To find the 3 dB compression point the power was swept at $f = 1420.4\,\text{MHz}$.
At $-76\,\text{dBm}$ the output power should be $-76\,\text{dBm} + 86\,\text{dB} = 10\,\text{dBm}$, but is only 7 dBm as shown in figure 5.16.

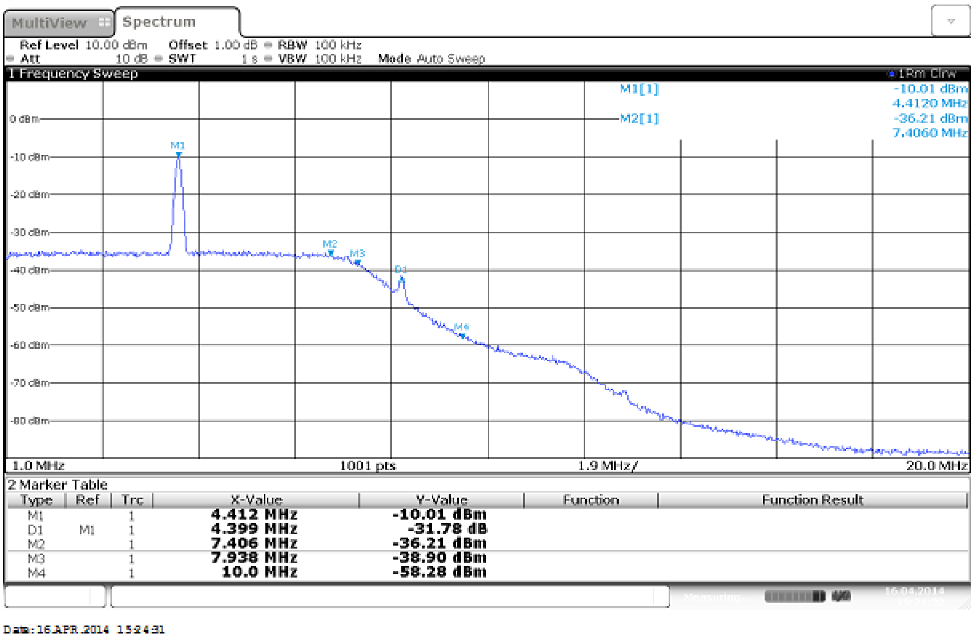

Figure 5.15: RX frequency response with lower input power

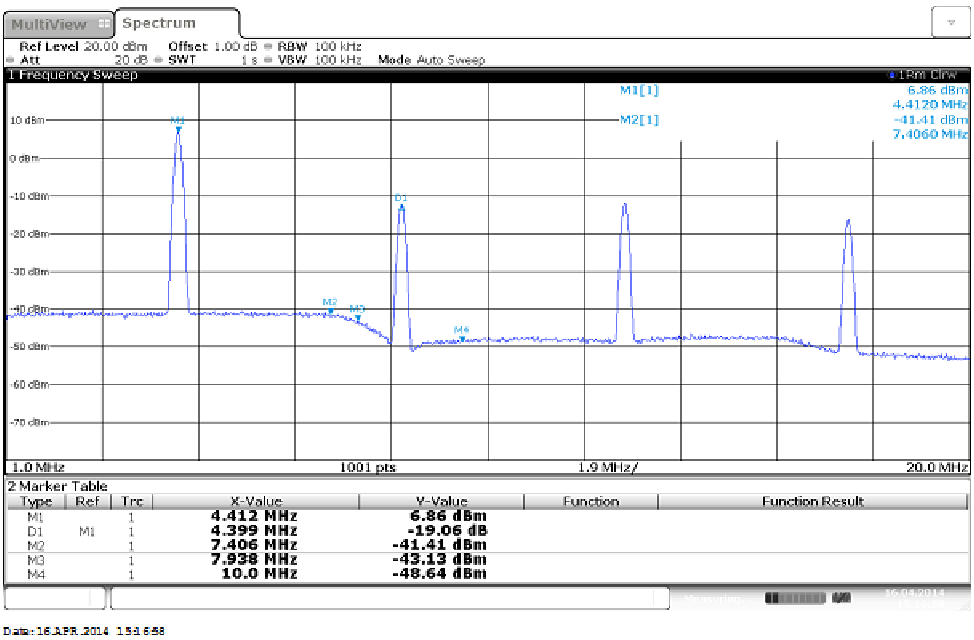

Figure 5.16: RX Compression Point

System Noise

To see the system noise the input power was simply turned off, see figure 5.17.

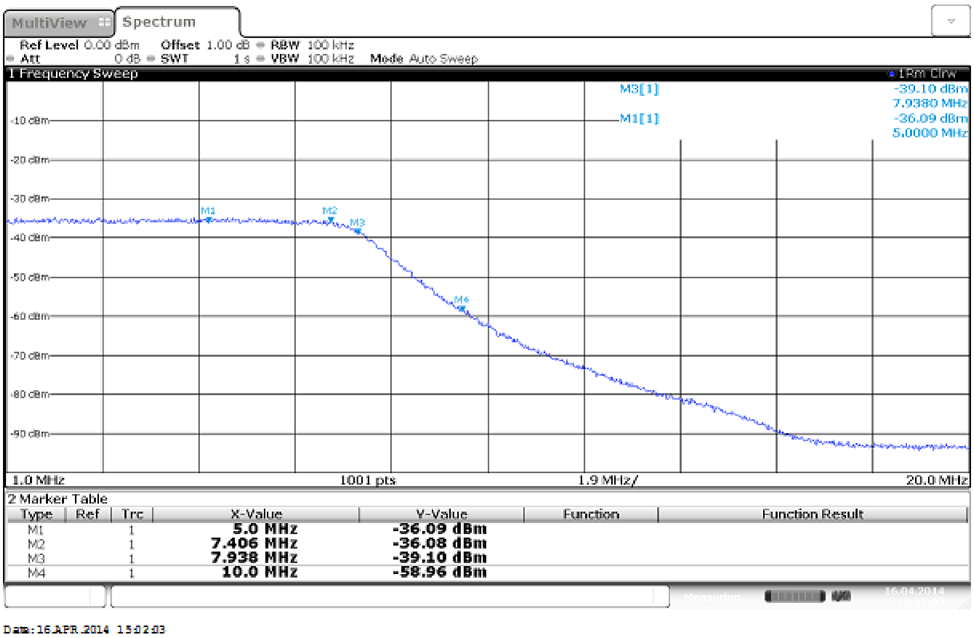

Figure 5.17: RX system noise

The Spectrum Analyzer measures in every (frequency) point the RMS power in a 100 kHz Band, e.g. Marker1 shows −36 dBm. For a rough estimation can be further assumed a width of 8 MHz for the full band. This is also what the ADC will see. Hence the band is eighty times wider than the RBW, in dB $10 \lg 80 = +19\,\mathrm{dB}$. Therefore we get a noise power for the full band of $-36\,\mathrm{dBm} + 19\,\mathrm{dB} = -17\,\mathrm{dBm}$.

5.2.4 Conclusions

The receiver chain can receive signals in a 6 MHz ... 8 MHz band around 1420.4 MHz, i.e. from 1417.4 MHz to 1423.4 Mhz.
The gain is 86 dB.
Signals with a power higher than −86 dBm will drive the chain into saturation. If the signal power drops to lower than −17 dBm − 86 dB = 103 dBm it starts to disappear in the noise floor.
The image is suppressed with 33 dB.
All voltages were as expected.

5.2.5 Time Domain Measurements

To get an impression in the time domain and what the ADC will see the signal was also measured with an Oscilloscope.
The setup is identical to figure 5.4, where the Oscilloscope replaces the Spectrum Analyzer. The input impedance of the Oscilloscope was set to 50 Ohm. In all following measurements the frequency of the input signal was 1420.4 MHz.

Figure 5.18 shows the (output) signal for an input power of −86 dBm, what results in an output power of (please see figure 5.11 Marker readout)

$$-0.13\,\text{dBm} = 10^{-0.13/10}\,\text{mW} \approx 1\,\text{mW}.$$

This would be an effective power at a 50 Ohm resistor of

$$p = \frac{u^2_{\text{effective}}}{R} = \frac{(\hat{u}/\sqrt{2})^2}{R}.$$

Thus the amplitude of the voltage should be

$$\hat{u} = \sqrt{2R \cdot p} = \sqrt{2 \cdot 50\,\text{Ohm} \cdot 0.001\,\text{W}} \approx 316\,\text{mV}.$$

Figure 5.18 shows the measurement with 100 mV per Division. The frequency was calculated by the Oscilloscope as shown in the picture [1]. The signal is a bit noisy but fits to the theoretical expectations.

Figure 5.19 shows the measurement with an input power of -96 dBm and a screen resolution of 50 mV per Division. The signal becomes already very noisy – as estimated before it should be only 7 dB above the noise floor at −103 dBm.

Figure 5.20 shows the measurement with an input power of -76 dBm and a screen resolution of 200 mV per Division. As already assumed from the compression measurements the signal is clearly clipped.

[1] The Oscilloscope calculates the mean of the signal, uses the mean crossings to measure the period time T (dotted lines) and calculates the frequency $f = 1/T$.

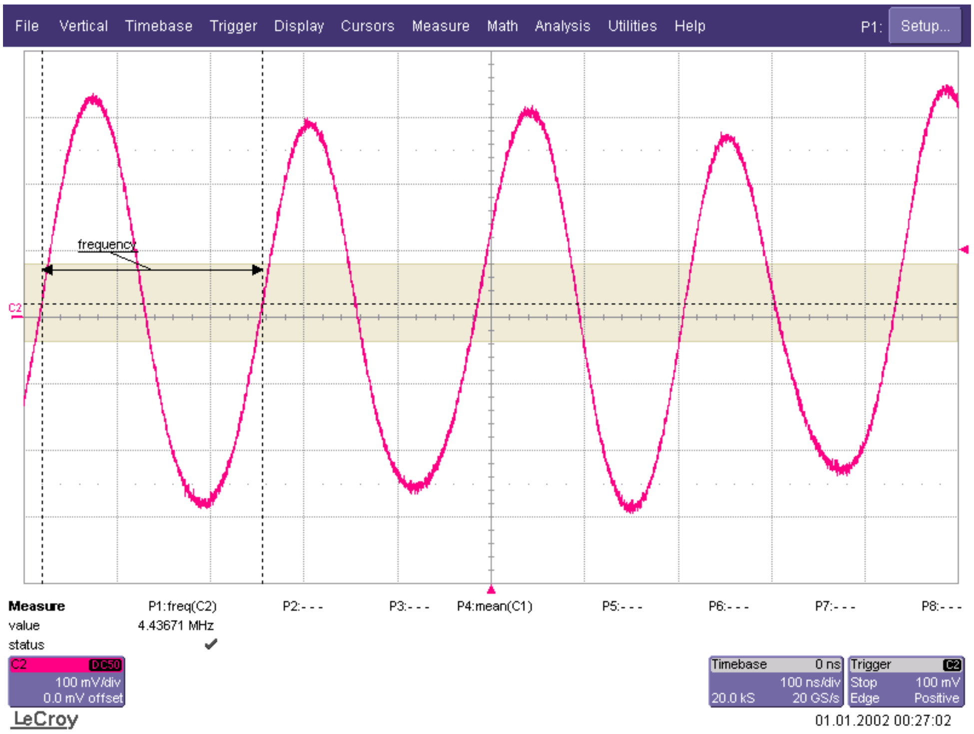

Figure 5.18: RX Time Domain

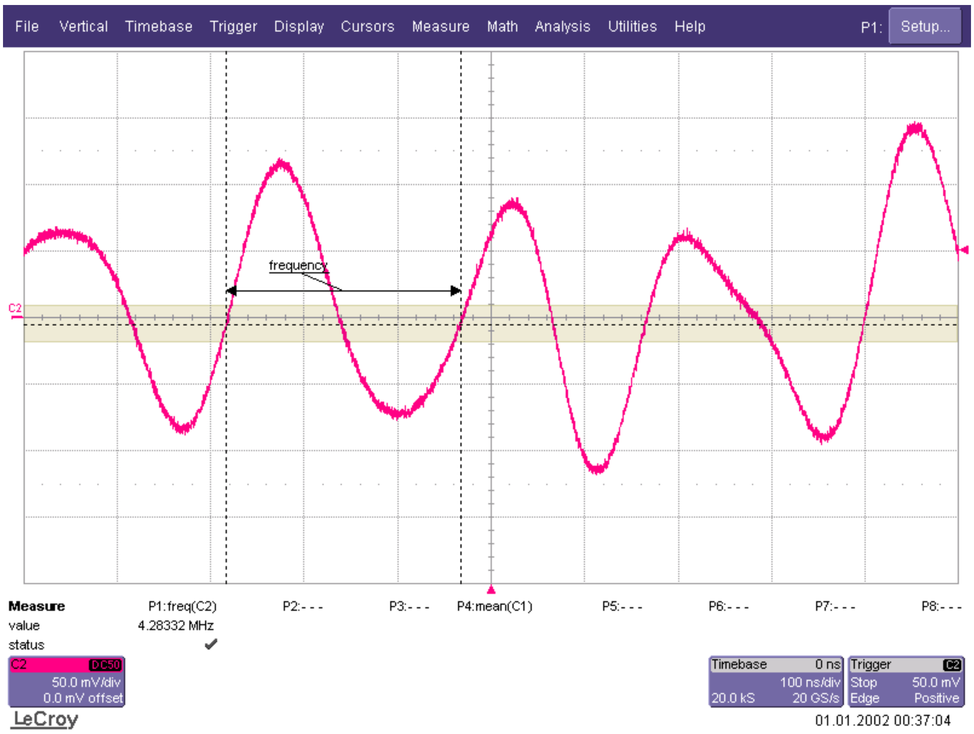

Figure 5.19: RX Time Domain with reduced input power

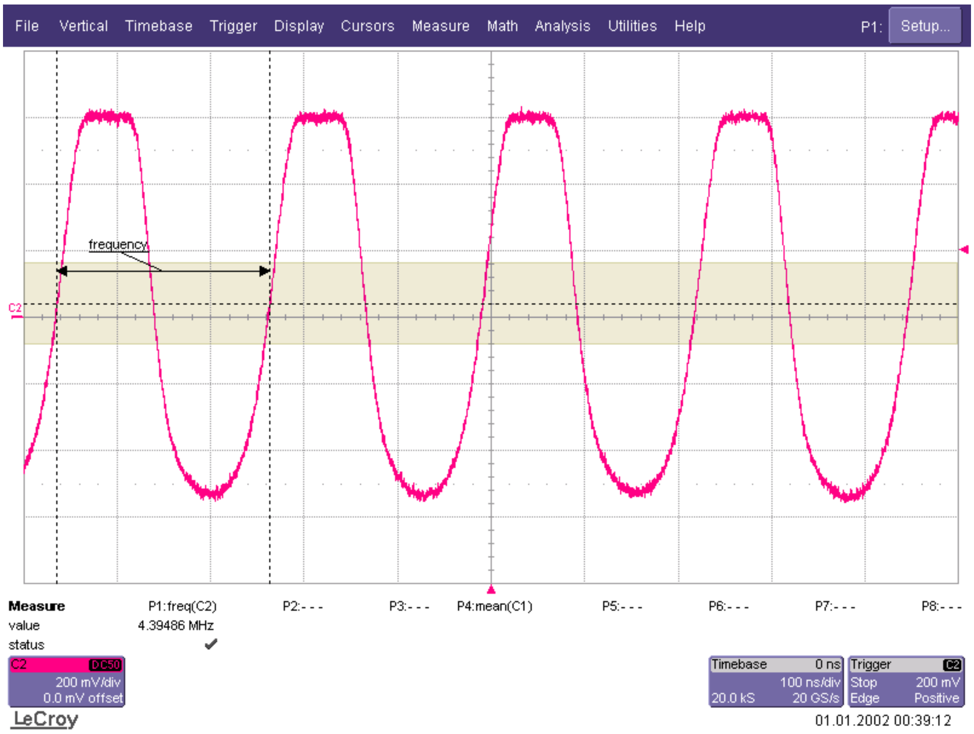

Figure 5.20: RX Time Domain with increased input power

5.3 Customized Filter Performance Plots

Frequency Response In Band

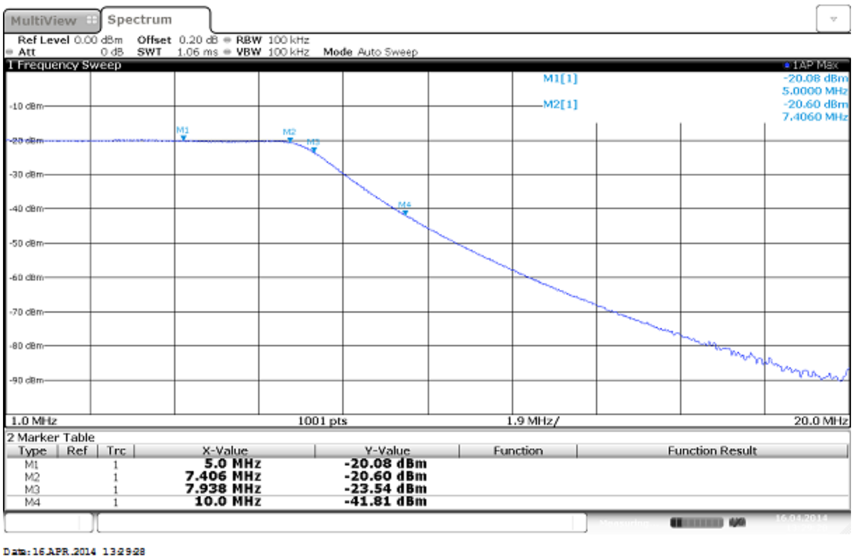

Figure 5.21: LPF Frequency Response In Band

Frequency Response Wide Band

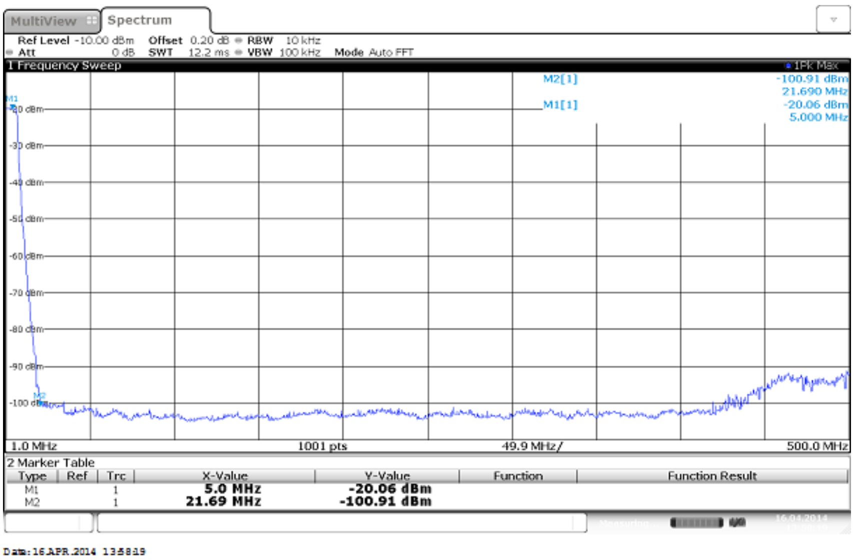

Figure 5.22: LPF Frequency Response Wide Band

Chapter 6

Digital Signal Processing

6.1 Introduction

Most of the signals in our daily life are analog, like voice, light, temperature or speed. Analog means that a signal is continuous. A digital signal comprises only a finite amount of values. Analog signals can be converted into digital signals and vice versa with the help of ***A****nalog to* ***D****igital* ***C****onverters* and ***D****igital to* ***A****nalog* ***C****onverters.*

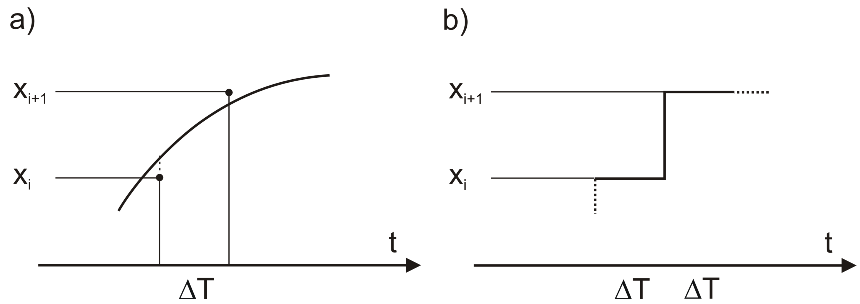

Figure 6.1: Sample and Hold

Figure 6.1 a) shows the process of an ADC, which is also called sampling. The ADC runs at a certain sampling rate $f = \frac{1}{\Delta T}$. The data range and the resolution of the ADC are limited, for example an 8 Bit ADC has $2^8 = 256$ values available. This is a range of $0 \ldots 255$ or $-128 \ldots 127$ with a resolution of $\frac{1}{256-1}$. At every sampling time point the analog signal is measured and mapped to the closest available digital value. The amplitude of the analog signal should match into the available range, for example if the signal would be $A\sin(\omega t)$ in our case the ADC should map $-A \ldots A$ to $0 \ldots 255$. Now the signal is described by $x_i \in \{0, 255\}$.
We could give this signal now to a DAC, which would give out a waveform like in figure 6.1 b). Obviously digitizing always distorts the signal.

Why are we using digital techniques? It not only distorts the signal but also often complicates the situation, for example an analog low pass filter might be much easier to implement than to process it digitally. But there are plenty of advantages:
In communication, if the analog waveform for example is an audio signal on a telephone line, noise will distort the signal and lower the quality. Furthermore the signal is attenuated by the cables and the needed amplifiers add noise, too.

Analog signals can be difficult to store.
Digital numbers can be stored or processed perfectly. For transmission we could sent their binary representation, for example 0 would be 0 Volt, 1 5 Volt. Even if some noise is added the receiver can detect and perfectly recover the signal.
Digital technique might be better suiting the situation, for example if we want to calculate numbers. Furthermore we can easily apply mathematics, for instance to encrypt or compress data.
Another advantage is that parameters, like an amplitude, can be precisely controlled.

Digital data is always stored and processed in binary form. Thus on the low level only Bit arithmetic is needed. One Bit can be inverted ($0 \to 1$ or $1 \to 0$), which also can be seen as *NOT*. Two Bits can be combined as *AND* ($0 \,\&\, 0 = 0$, $0 \,\&\, 1 = 0$, $1 \,\&\, 0 = 0$, $1 \,\&\, 1 = 1$), *OR* ($0 \,|\, 0 = 0$, $0 \,|\, 1 = 1$, $1 \,|\, 0 = 1$, $1 \,|\, 1 = 1$) and many others. Furthermore Bits can be added, multiplied etc. All operations can be broken down to two of the basic functions, for example NOT and AND.
Digital circuits can be divided in synchronous and asynchronous circuits. Asynchronous means that the circuits gets an input and is immediately processing the signal, for example a simple AND that gives out the result as soon as it sees an input. Synchronous circuits are clocked and process in steps, meaning they take over input signals, store, change their values etc. only if they are triggered.
Digital circuit design covers several areas like: designing circuits that give certain outputs for certain inputs, ADCs, timing and synchronizing issues, routing of transmission lines for high speed signals, optimizing for power consumption, area (size of circuit), designing high level functions and many others.

In this thesis we can only discuss the basics of ***D****igital* ***S****ignal* ***P****rocessing*.

6.2 Fourier Transformation

The Fourier Transformation is well known, but some basic ideas are described here as foundation for the following DSP. More details can be found in mathematics or DSP books, for instance [AHKKLS], [MeyVa2], [Goetz], [Gruen], [KamKro], [Mey], [ProMan], [Lyon].

First we remember the orthogonality relationship

$$\frac{1}{T}\int_0^T e^{im\omega t}\overline{e^{in\omega t}}dt = \begin{cases} 1 & n = m \\ 0 & n \neq m \end{cases} \tag{6.1}$$

where $m, n \in \mathbb{Z}$, $T = 1/f$ is the period length with $\omega = 2\pi f$.
Next we assume that a periodic signal $f(t)$ is made up by sinusoidal signals, meaning

$$f(t) = \sum_{k \in \mathbb{Z}} c_k e^{ik\omega t} \tag{6.2}$$

This is called Fourier Series and is a good assumption for all signals we encounter.
Now we can work out the *Fourier Coefficients* by using 6.1

$$c_k = \frac{1}{T}\int_0^T f(t)e^{-ik\omega t}dt \tag{6.3}$$

For a real signal $f(t) \in \mathbb{R}$ holds $c_k = \overline{c_{-k}}$.
The Fourier Coefficients are the complex amplitudes of the *harmonics*, which are the single $e^{ik\omega t}$ components of the signal. The absolute value of the amplitudes $c(\omega)$ versus frequency is called spectrum. We also refer to it as *frequency domain*, where the $f(t)$ point of view is the *time domain*.

For non-periodic signals we can write above relationships as

$$f(t) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \left(e^{i\omega_k t} \int_{-T/2}^{T/2} e^{-i\omega_k t} f(t) dt \right) \Delta\omega$$

with $T \to \infty$, $\Delta\omega = \frac{2\pi}{T} \to 0$ and $\omega_k := k\omega$. Thus we get the general Fourier Transformation

$$\begin{aligned} f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left(e^{i\omega t} \int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt \right) d\omega \\ f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} c(\omega) d\omega \\ c(\omega) &= \int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt \end{aligned}$$

This shows, that the discrete coefficients of periodic signals become a density for a non-periodic signals.

The Fourier Transformation of a signal $x(t)$ is also denoted as $X(f)$.

Of course in DSP we can integrate only over a finite period of time. Thus we will always stay with 6.2 and 6.3. This means that we can handle only periodic signals. Of courses hardly any signals are periodic and we will later on see what this means to DSP.
For now we stay with signals with period length T, $\omega = \frac{2\pi}{T}$. They are sampled with a frequency $f_S = 1/\Delta t$ and $T = N\Delta t$, where N is a natural number. In other words for one period we get N samples with distances Δt.
The period length T doesn't play a role for the coefficients in 6.3, therefore we set it to N. For the discrete points of time we can simply use $n \in \mathbb{N}$.
Together with 6.2 and 6.3 we achieve

$$\boxed{c_k = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{-ik\frac{2\pi}{N}n}} \tag{6.4}$$

and

$$\boxed{f(n) = \sum_{k=0}^{N-1} c_k e^{in\frac{2\pi}{N}k}} \tag{6.5}$$

where $k, n = 0 \ldots N-1$ and N is the number of samples in one period.

Equations 6.4 and 6.5 are called ***D**iscrete **F**ourier **T**ransformation* and ***I**nverse **D**iscrete **F**ourier **T**ransformation*.

Note that if we plug 6.4 into 6.5 we can see that we could move the $\frac{1}{N}$ from 6.4 to 6.5. Or we could even split it or introduce another factor. And indeed we can find different definitions for the Fourier Transformation.

The coefficient-index $k \in \mathbb{N}$ goes from 0 to $N-1$.
For $k = 0$ equation 6.4 is the average of $f(n)$ and hence the coefficient c_0 is the DC component of the function. The coefficients with $k > 0$ are the complex amplitudes for the harmonics.
The complex pointer $e^{-ik\frac{2\pi}{N}}$ in 6.4 runs for $k = 0 \ldots N-1$ on a circle. Figure 6.2 shows an example for $N = 8$.

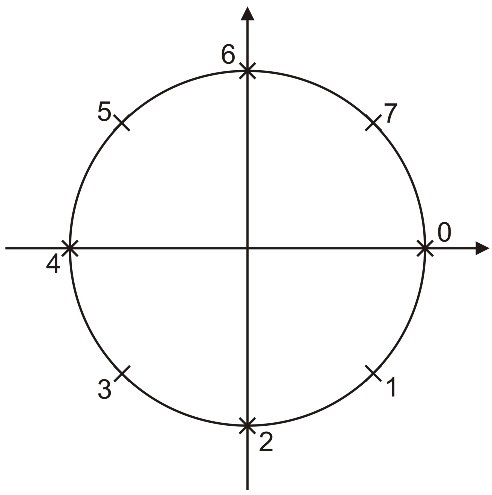

Figure 6.2: Complex Circle

Obviously the pointers are complex conjugate pairs, in our example 1 and 7 , 2 and 6 and so on. We could run on the circle in the opposite direction for the upper half of the indices k. Therefore we replace these k with $k - N$, in our example instead $k = 7,\ 6,\ 5$ we use $k = -1,\ -2,\ -3$ – and get the same pointers.
This proofs that 6.4 / 6.5 is identical to 6.2 / 6.3 for time discrete functions. The coefficients $c_{N-1},\ c_{N-2},\ c_{N-3},\ \ldots$ are identical to $c_{-1},\ c_{-2},\ c_{-3},\ \ldots$.
In the sum in 6.4 the complex pointer pairs $e^{-i(\pm)k\frac{2\pi}{N}}$ are rotated with n and stay complex conjugate during the rotation, since they just rotate in the opposite direction. Furthermore they are weighted with $f(n)$, what could shift them in phase. But for a real signal $f(n) \in \mathbb{R}$ they stay complex conjugate and we get as well $c_1, c_2, c_3, \ldots = \overline{c}_{-1}, \overline{c}_{-2}, \overline{c}_{-3}, \ldots$ or $c_1,\ c_2,\ c_3,\ \ldots = \overline{c}_{N-1},\ \overline{c}_{N-2},\ \overline{c}_{N-3},\ \ldots$.
Finally we put the c_k back into 6.5 to see how $f(n)$ is calculated.
As discussed for 6.4 or shown in figure 6.2 the pointers $e^{in\frac{2\pi}{N}k}$ and $e^{in\frac{2\pi}{N}(N-k)}$ are complex conjugate pairs for every n. Just like the c_k for real signals $f(n) \in \mathbb{R}$. Thus the imaginary parts of $c_k e^{in\frac{2\pi}{N}k}$ and $c_{N-k} e^{in\frac{2\pi}{N}(N-k)}$ cancel while the real parts add up. These are the components, called *harmonics*, which build the real signal. Over time, here n, the harmonics are oscillating. They have amplitude $2|c_k|$ and their phase is defined by the phase of c_k.

Let us have a look at the frequency axis of the DFT. Our signals have period length T ($\omega = \frac{2\pi}{T}$) and are sampled with frequency $f_S = 1/\Delta t$ and N samples per period, therefore $T = N\Delta t$. Furthermore we assume that we use all samples for the DFT [1].
With equation 6.2 and 6.3 it is obvious that the frequency spacing of the harmonics is $f = \frac{1}{T}$. The sampling frequency is $f_S = Nf = \frac{N}{T}$. Therefore we have for the frequency spacing (or resolution)

$$\Delta f = \frac{f_S}{N} \tag{6.6}$$

We saw that for the coefficients c_k, $k = 0 \ldots N-1$, that the upper half of the frequencies are actually identical to negative frequencies with $k = -1, -2, \ldots$. Thus the maximum frequency we actually derive is $f_{max} = \frac{(N-1)}{2}\frac{f_S}{N} \approx \frac{f_S}{2}$. To be precise we have to consider if N is even or odd. If N is even then $c_{N/2}$ does not have a negative counterpart. We can see this also easily with figure 6.2: in the example pointer 4 is -1 and real, while all others are coming in pairs. To get the correct result we have to multiply c_4 by two. Note that c_0, the DC component, is correct. But it is popular to not consider this harmonic and use for the frequencies

$$f = k\frac{f_S}{N},\ k = \begin{cases} 0 \ldots \frac{N-1}{2} & N \text{ odd} \\ 0 \ldots \frac{N}{2} - 1 & N \text{ even} \end{cases} \tag{6.7}$$

By taking a number of samples with $N = 2^m$ ($m \in \mathbb{N}$) the DFT can be further developed to a very efficient algorithm. This is called ***F**ast **F**ourier **T**ransformation*. Most of the time the FFT is used for speed reasons or, when it is processed in Hardware, even for power consumption. But the FFT is nothing else than an efficient DFT and above mentioned books describe how the algorithm works.

DFT and FFT have several properties and rules for calculation, like linearity, differentiation etc. that are described in the literature as well.
We will see that two properties are important later on and shall be mentioned here. According to [MeyVa2] we have:

Shifting

$$f(t+a) = \sum_{k\in\mathbb{Z}} e^{ik\omega a} c_k e^{ik\omega t} \tag{6.8}$$

where $e^{ik\omega a}c_k$ can be seen as new complex coefficient ${c'}_k$.
The coefficients ${c'}_k$ and c_k are just phase shifted and thus are the harmonics of the signal. Looking at the signal at another point of time means looking at its sinusoidal harmonics at another point of time – what is equal to a shift in phase.

Convolution

$$z(t) = \int_{-\infty}^{\infty} x(t-\tau)y(\tau)d\tau =: x(t) * y(t) \tag{6.9}$$

[1] In general this doesn't need to be the case, since we can sample a signal and just take a subset of the samples to perform the DFT

Note that $x(t) * y(t) = y(t) * x(t)$.

The time discrete convolution is analogue

$$z(n) = x(n) * y(n) = \sum_{i=-\infty}^{\infty} x(n-i)y(i) \tag{6.10}$$

It is easy to see that if one of the functions is the Dirac pulse at time T (or n) then

$$x(t) * \delta(t-T) = x(t-T) \tag{6.11}$$

The Fourier Transformation of $x(t) * y(t)$ is simply the product of their individual transforms

$$Z(\omega) = X(\omega)Y(\omega) \tag{6.12}$$

and again for the time discrete form holds analogue

$$Z(n) = X(n)Y(n) \tag{6.13}$$

And for the product in the time domain

$$z(t) = x(t)y(t) \tag{6.14}$$

the Fourier Transformation is the convolution

$$Z(\omega) = \frac{1}{2\pi} X(\omega) * Y(\omega) \tag{6.15}$$

In the literature also many examples are worked out. We will see that one of the most important for DSP is the series of Dirac-Pulses

$$x(t) = \sum_{n=-\infty}^{+\infty} \delta(t-nT) \tag{6.16}$$

which has Fourier Transformation [Gruen]

$$X(f) = \frac{1}{T} \sum_{k=-\infty}^{+\infty} \delta(f - k\frac{1}{T}) \tag{6.17}$$

Both shown in Figure 6.3.

Another very important Function is the rectangular pulse.

$$x(t) = \begin{cases} 1 & |t| \leq T/2 \\ 0 & t > T/2 \end{cases} \tag{6.18}$$

Its Fourier Transformation is

$$X(f) = T\,\mathrm{sinc}(\pi f T) \tag{6.19}$$

where the function sinc (also called si) is defined as

$$\mathrm{sinc}(x) = \begin{cases} \frac{\sin(x)}{x} & x \neq 0 \\ 1 & x = 0 \end{cases} \tag{6.20}$$

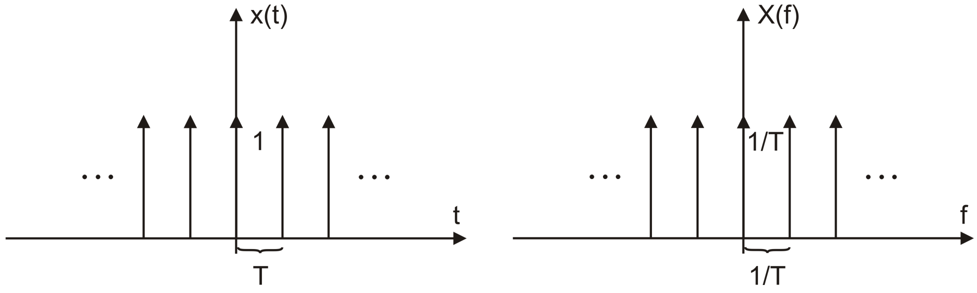

Figure 6.3: Dirac Pulses and their Fourier Transformation

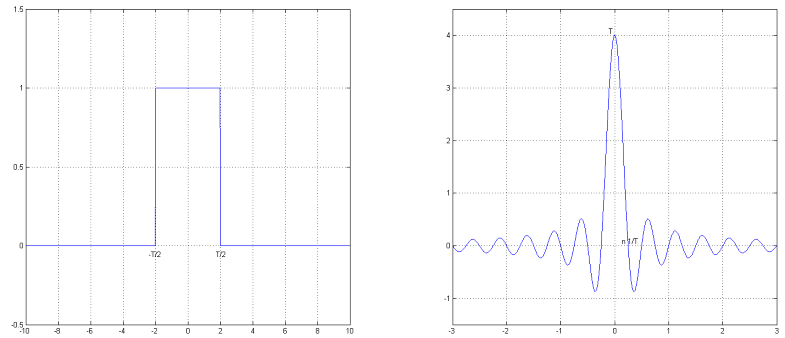

Figure 6.4: Rectangular Pulse and its Fourier Transformation

The rectangular pulse and its Fourier Transformation are shown in figure 6.4.
If the pulse has a width T, its Fourier Transformation has an amplitude of T and zeros at $n\frac{1}{T}$, $n \in \mathbb{Z}$.

Next we look at a concrete example.
In addition to this we start to use MATLAB [2] to calculate the spectrum.

The rectangular signal

$$f(t) = \begin{cases} 1 & 0 \leq t < \pi \\ -1 & \pi \leq t < 2\pi \end{cases} \tag{6.21}$$

with period $T = 2\pi$ can be developed according to [MeyVa2] as

$$f(t) = \frac{4}{\pi}\left(\frac{\sin t}{1} + \frac{\sin 3t}{3} + \frac{\sin 5t}{5} + \cdots\right) \tag{6.22}$$

To make the MATLAB code self explaining it contains explanations as comments (in MATLAB starting with %).
Note that in MATLAB vectors start with index one (not zero as for instance in C).
To create a vector t like $(a,\ a+\Delta,\ a+2\Delta,\ \ldots,\ b)$ MATLAB offers the colon operator: $t = a : \Delta : b$.
First the three harmonics are calculated, summed up and plotted.

[2] For further details on Software Tools please see appendix A.

```
TT = 100; % period length
N = 256;  % number of samples
t = 0:(TT/(N-1)):TT; % time-vector: N points of time from 0 to TT

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

w = 2*pi/TT; % omega
sig1 = 4*sin(w*t)/(pi);     % 1. harmonic
sig2 = 4*sin(3*w*t)/(3*pi); % 3. harmonic
sig3 = 4*sin(5*w*t)/(5*pi); % 5. harmonic

sig = sig1+sig2+sig3; % signal: sum of first 3 harmonics

f1 = figure;          % new figure
plot(t, sig, 'r');    % signal vs. time in red
grid on; hold on;     % grid on; next plots into same figure
plot(t, sig1, 'b'); % 1. harmonic in blue
plot(t, sig2, 'g'); % 3. harmonic in green
plot(t, sig3, 'y'); % 5. harmonic in yellow
```

The first MATLAB plots, figure 6.5, show the three harmonics and their sum.

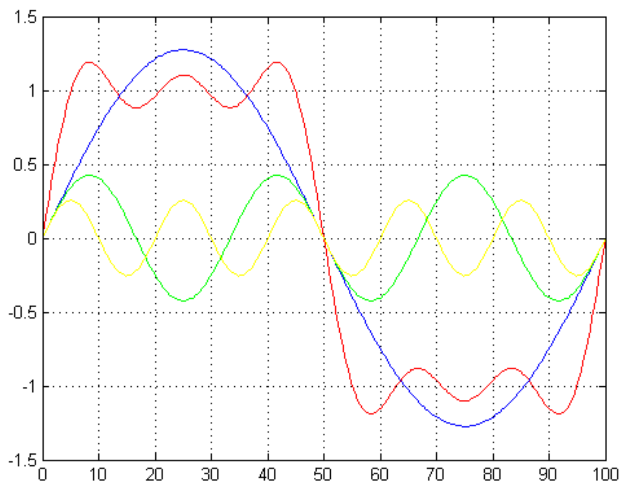

Figure 6.5: Signal with 3 harmonics

Now the FFT is calculated with the help of the `fft(.)` command. MATLAB does not divide by $1/N$ as in 6.4. This is done in the next line.

```
spectrum = fft(sig);
spectrum = spectrum/N; % MATLAB doesn't divide by number of samples
```

The same result could have been achieved manually according to 6.4. Note that MATLAB provides the `zeros(m,n)` command to create an $m \times n$ zero matrix – here to generate spectrum $= (c_k) = (0, \ldots, 0)$.

```
spectrum=zeros(1,N);

for k=0:(N-1)
    for n=0:(N-1)
        spectrum(k+1) = spectrum(k+1) + sig(n+1)*exp(-k*n*2*pi*i/N);
    end
end

spectrum = spectrum/N;
```

Now we want to plot the absolute and the angle of the complex amplitudes.

```
f2 = figure;
plot(abs(spectrum), 'b');
grid on;

f3 = figure;
plot(angle(spectrum), 'g'); % +- pi/2 => c_k = conj.compl.(c_-k), sin
grid on;
```

The spectrum is shown in figure 6.6, the angles in figure 6.7.

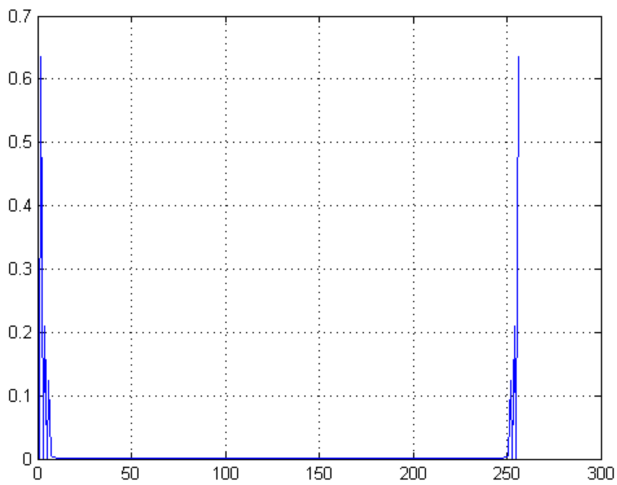

Figure 6.6: Signal with 3 harmonics

The vector *spectrum* contains the complex amplitudes: (spectrum(1), ..., spectrum(N)) = (c_0, ..., c_{N-1}). We saw that for a real signal c_k and the complex conjugate $c_{-k} = c_{N-k}$ are added to the real amplitude of the k-th harmonic ($k > 0$).

- Spectrum 2 and 256 are 0.6362.
  Their sum is $1.2724 \approx \frac{4}{\pi}$, the amplitude of the 1. harmonic.
- Spectrum 4 and 254 are 0.2108.
  Their sum is $0.4216 \approx \frac{4}{3\pi}$, the amplitude of the 3. harmonic.

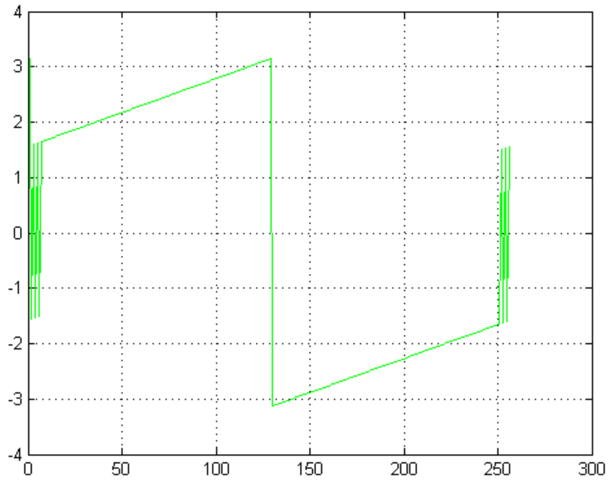

Figure 6.7: Angles of the complex coefficients.

- Spectrum 6 and 252 are 0.1244.
  Their sum is $0.2488 \approx \frac{4}{5\pi}$, the amplitude of the 5. harmonic.

- All other spectrum-components are 0, inclusive spectrum 1 that represents the DC.

The calculated angles of the component pairs ((2, 256), ...) are ±90° and thus complex conjugate. Furthermore ±90° means pure imaginary coefficients.
Now we can plug the complex coefficients back into 6.5.
The pointer $e^{in\frac{2\pi}{N}k}$ is running on a circle, similar as discussed before and shown in figure 6.2, just in the opposite direction.
If we put in the c_k, keeping in mind their pairing, they add up to a real signal − and since the coefficients are pure imaginary our sinus functions are reproduced, which we used to built up the signal.

Of course we want to automate the calculation of the real amplitudes:

```
% calculate single sideband spectrum
spectrumssb=spectrum( 1:(length(spectrum)/2) );
for n=1:((length(spectrum)/2)-1)
spectrumssb(n+1)=spectrumssb(n+1)+conj(spectrum(length(spectrum)+1-n));
end

f4 = figure;
plot(abs(spectrumssb), 'r');
grid on;
```

The last plot, figure 6.8, finally shows the real spectrum, with the amplitudes as discussed before.

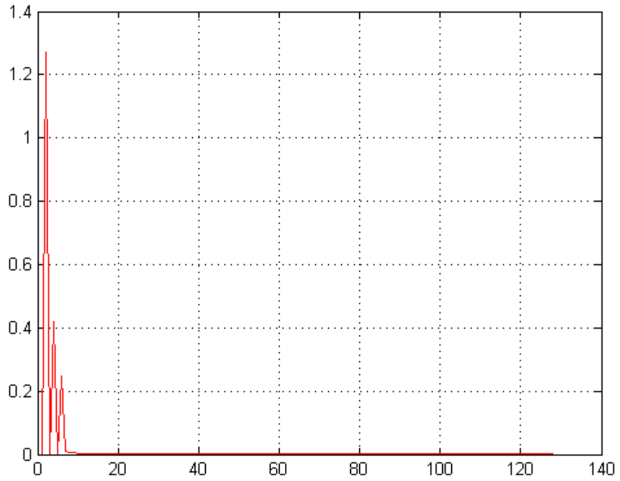

Figure 6.8: Single Side Band Spectrum of Signal

6.3 Sampling

Analog signals are converted to digital signals with the help of ***A****nalog to* ***D****igital* ***C****onverters*. The ADC works with certain number of N Bits. Therefore it has a resolution of $2^N - 1$ and covers a range from for example $0 \dots 2^N - 1$ or $-2^{N-1} \dots 2^{N-1} - 1$. The ADC *samples* the analog signal at a certain rate, the *sampling frequency* f_S. The samples are mapped to digital range of the ADC as shown in figure 6.9.

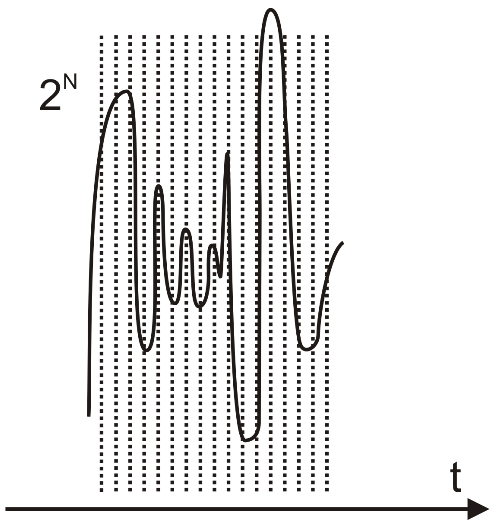

Figure 6.9: Analog to digital mapping

In the optimal case the amplitude range of the analog signal is perfectly mapped to the digital range. But if the analog signal gets too high or low it clips, in other words these parts of the signal getting cut to the maximal or minimal available value. In the opposite case the analog amplitude range stays behind the digital range. In this case the resolution can be improved by a better scaling.

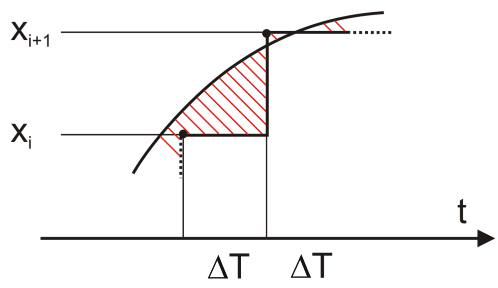

Figure 6.10: Quantization Error

If we give the ADC signal directly to an ***D****igital to* ***A****nalog* ***C****onverter* the original analog signal would turn into a kind of stairway. Figure 6.10 shows the two signals and their difference, in the other words the error due to quantization.
A standard assumption is that the spectrum of the error signal is equally distributed in the frequency domain, in other words *white*. This error is called *quantization noise*.
We can not proof it here mathematically, but is easy to imagine that corners, edges and steep slopes are similar to and thus made of sinusoids with high frequencies. This means a wide band is occupied.
Since signals are often much more limited in their bandwidth than the noise a significant portion of the noise can usually be filtered away.
We can look at it also this way: filtering, i.e. limiting the bandwidth of a signal, flattens edges and corners. Thus the above mentioned stairway gets closer to the original signal.

Next we are going describe sampling mathematically.

Therefore we model the ADC with the help of the Dirac-pulses with distances T, 6.17, 6.16 and 6.3.
The analog input signal is multiplied with the pulses, where every pulse means sampling at this point of time. For the sampling frequency holds $f_S = 1/T$.
As described with 6.15 the Fourier Transformation will be the convolution. Equation 6.3 shows that the Fourier Transform of Dirac pulses are also Dirac pulses at f_S. Further we can apply equation 6.11 in the frequency domain.
The spectrum of the signal will appear around the spectral peaks of the Dirac-Pulses, besides a constant amplitude factor $1/T$.
The spectrum before and after sampling is shown in figure 6.11.

If the spectrum of the original signal is wider than the half sampling frequency, it is going to overlap with itself! This of course destroys the signal and has to be avoided.
Therefore analog signals have to be filtered with a ***L****ow to* ***P****ass* ***F****ilter* that suppresses the spectrum from $f_S/2$ on as good as possible.
On the other hand the sampling frequency has to be chosen high enough so that the analog signal fits (after low pass filtering) into the frequency range up to $f_S/2$.

At this point we can also see why a down conversion of signals is so important. The whole signal, i.e. inclusive its harmonics, has to fall into a range below the half sample frequency. Signals can be transmitted in the microwave range. But most of today's processor work with approximately 1 GHz. Digital circuits are cheaper, easier to build

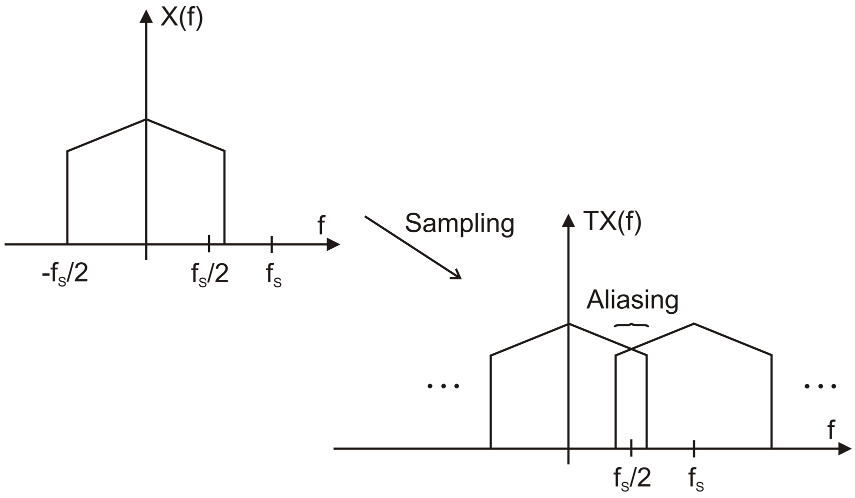

Figure 6.11: Sampling and Aliasing

and to handle and much less power consuming the lower their clock frequency is.

Of course when we start to sample a signal we start at any time without reference to the signal itself. But with equation 6.8 we saw that this means only a phase shift. We can never work out the absolute phase of the signal, but this does not influence the spectrum.

6.4 DFT and arbitrary signals

So far we discussed the DFT for periodic signals. Of course in general signals are not periodic.
We discuss the effects with the help of picture 6.12, inspired by [Goetz]. On the left hand side is shown the time domain and in parallel the frequency domain on the right.
Figure a) shows a sinusoidal signal $x(t)$ with period T_0, its spectrum is a simple Dirac. The signal can be measured (sampled) only during a certain period of time T_1 and in general this is not identical to the signal period: $T_1 \neq T_0$. The length of the sampling can be modelled as window w, shown in figure b), multiplied with the signal. For the spectrum of a rectangular signal we already found 6.19, plotted in 6.4. Hence we come to figure c). But since the DFT assumes periodic signals we actually have figure d). As discussed earlier, in the chapter about Fourier Transformation, periodic signals have discrete spectra, calculated according equations 6.2 and 6.3. We can observe the effect in d). The signal is sampled with a frequency $f_S = 1/T_S$. The number of samples is N, in the example $N = 6$, therefore $T_1 = NT_S$. As discussed sampling can be described by multiplying the signal with Dirac pulses (their spectrum are pulses in the frequency domain). The signal spectrum will appear at the pulses in the frequency domain, shown in figure e).
If $1/T_1 = 1/T_0 = f_0$ then the *windowing* would reproduce the original spectrum. In figure d) (frequency domain) the pulses would appear exactly at $\pm f_0$ – and where the $W(f)$ has zeros.
If the period time of a sampled signal is not equal to the length of the sampling we get two effects.

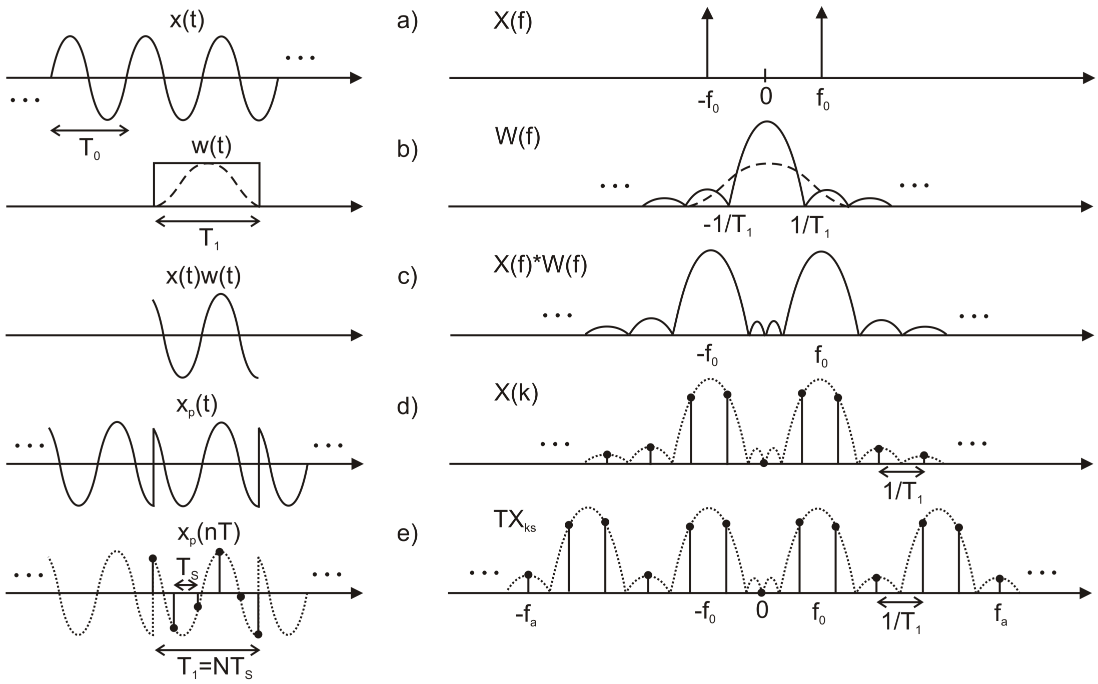

Figure 6.12: DFT of a sinusoidal signal

First *leakage*, that means that the processing creates spectral components where the orignal signal does not have any.

Furthermore we have to consider that the DFT calculates components at certain frequencies, called *(frequency) bins*. But the frequency bins are not identical to the frequency components of the original signal. Thus the spectrum is calculated around the bins, rather than the real components, and the coefficients do not have the correct amplitudes. This is called *picket fence* effect.

Both effects can be improved by using another window than the rectangular one.

Many different windows have been developed and are described in [Goetz], [Gruen], [KamKro], [ProMan] or [Lyon].

A simple and very popular window is the so called *Hanning* or *Hann* window. It is defined as

$$w(n) = \begin{cases} 0.5 - 0.5\cos(2\pi n/N) & n = 0 \dots N-1 \\ 0 & otherwise \end{cases} \tag{6.23}$$

A Hann window is plotted in figure 6.13 (for $N = 64$) and in figure 6.12 b), dashed line.

To modify the rectangular window the samples of the signal simply have to be multiplied with the desired window before the DFT.

The spectral resolution is influenced by the windowing. For the Hann window the width of the main lobe is $8\pi/N$, the spectral resolution is about half of the rectangular window [Gruen].

But the resolution can be improved by *zero padding* [Lyon]. Therefore the vector with the sampled signal is appended by zeros. This does not add any information, of course, but it increases N and therefore the number of frequency bins, where the DFT is calculated.

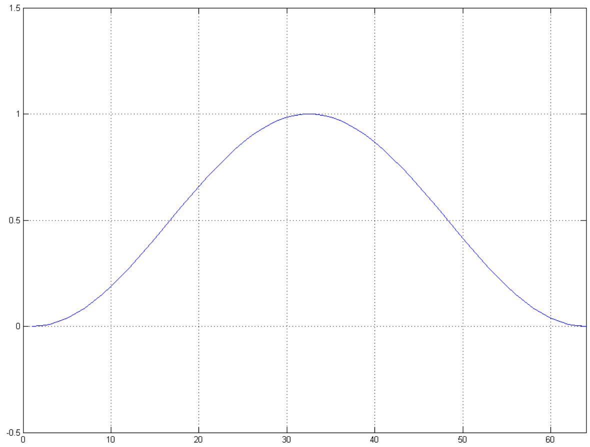

Figure 6.13: Hanning Window

6.5 Analog Digital Converter

Measurement Computing Corp. provides a PCI card for PCs. Among others it comprises an ADC with following specifications:

- 20 M Samples per second ($f_S = 20\,\text{MHz}$)
- 12-bit resolution
- Two input ranges: 1 Volt and 5 Volt (for 5 Volt the input signal is attenuated by 5)
- Connectors: BNC
- Input Impedance 1.5 MOhm

The half sampling frequency, into which the spectrum of the input signal has to fit, is $f_S/2 = 10\,\text{MHz}$. Therefore the RF receiver has to filter the signal, please see chapter 5 for its performance.
It is possible to record $N = 65536$ samples. Thus the DFT is calculated at frequency bins $k \cdot 20\,\text{MHz}/65536 \approx k \cdot 305\,\text{Hz}$. A Hann window halves the spectral resolution, appending 65536 zeros would bring it back to 305 Hz.
The input signal (between $\pm 1\,\text{V}$ is mapped to $0 \ldots 2^{12} - 1 = 4095$. The voltage of a sample can be calculated from the digital value by shifting it around zero (subtracting 2048) and scaling the amplitude from 2048 to 1 V.
If the input amplitude is higher than 1 V then the 5 V range has to be chosen. In th 5 V range the ADC attenuates the input signal by 5 (back to 1 V), thus we have to multiply by five when calculating the amplitude in Volts.
Measurement Computing are providing further details and data sheets at their homepage [McADC].
The UBUNTU driver for the ADC was provided by Prof. Dr. Warren J. Jasper.
A further program called *sample_signal* was developed in C, please see appendix D for details and source code. The program sample_signal triggers the driver to record a signal

with the help of the ADC. The measured samples, which are returned by the driver, are stored on the hard disc drive as text file. It might be useful to set some parameters of the measurement, like the number of samples, the voltage range of the ADC and the polling mode. The driver can either hand over the samples one by one. Another possibility is called ***D**irect **M**emory **A**ccess*. With the help of the DMA technology a computer component can directly access the memory of the PC to transfer data with the highest speed.
The parameters can be handed over to sample_signal, examples for calls

- `sample_signal -range 1 -ns 65536 -soft`
  Voltage range 1, 65536 samples, polling sample by sample
- `sample_signal -range 5 -ns 256 -dma`
  Voltage range 5, 256 samples, polling via DMA

6.6 DSP of the SRT

To test the hardware and lay the foundation for a practical training a software was developed based on Octave/MATLAB. For further details and information about software tools, UBUNTU, Octave, source codes etc. please see the appendix.
The Octave/MATLAB program, called *Analyse.m*, is divided in several section.
First sample_signal is called to record the signal coming from the Antenna-Receiver chain. As discussed the 21 cm line is mixed down to approximately 5 MHz. The signal is sampled with 20 MHz.
To test the ADC, driver and the developed software a reference signal was fed in. Instead of the receiver a signal generator was connected to the ADC. As signal generator served the Agilent 33220A. Please see pictures 6.14 and 6.15.

Figure 6.14: Signal Generator: Frequency

Picture 6.14 shows the frequency setting of 5 MHz, as well as the waveform set to *Sine*.
Picture 6.15 shows the amplitude set to 5 Volts Peak to Peak, as well a zero DC (offset).

Figure 6.15: Signal Generator: Amplitude

The ouput impedance was set to *High Z Load* to match the high input impedance of the ADC.

To verify the output an oscilloscope, the Tektronix 2211, was used, picture 6.16.

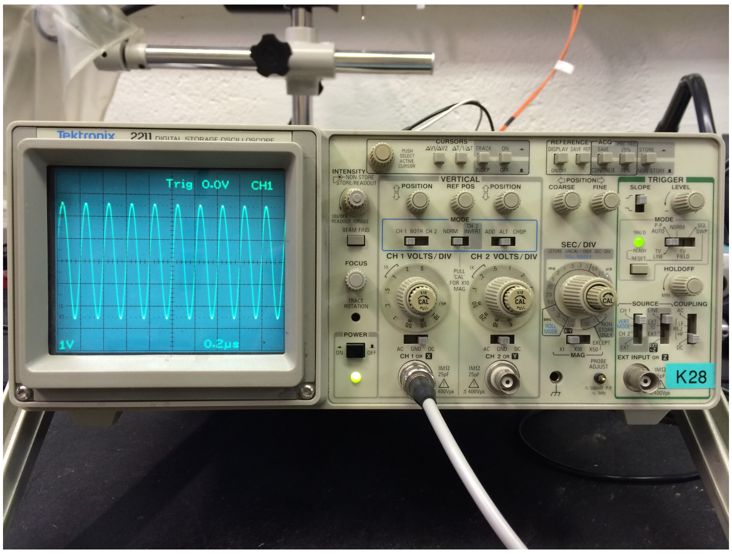

Figure 6.16: Oscilloscope verifying Signal Generator

After recording the signal the data is provided in the text file *sampled_signal.txt*. Analyse reads in the file and converts the digital data to voltages.
Then it creates a first figure in which it plots the samples. In the figure as well as in all following figures titles and labels for the axis are inserted. Then they are saved in different file formats, like PNG, PDF or EMF.
In the next section a Hann window is calculated and multiplied with the signal. The possibility for Zero-Padding is prepared.
Now the spectrum is calculated. After the DFT (FFT) the Fourier coefficient pairs c_k

with $k = (1,\ N-1),\ (1,\ N-1),\ \ldots$ are added. Finally the spectrum is plotted in a new figure. Here also a readout of the maximum, which should be the desired 5 MHz line, is inserted. An example by using the signal generator is in figure 6.17.

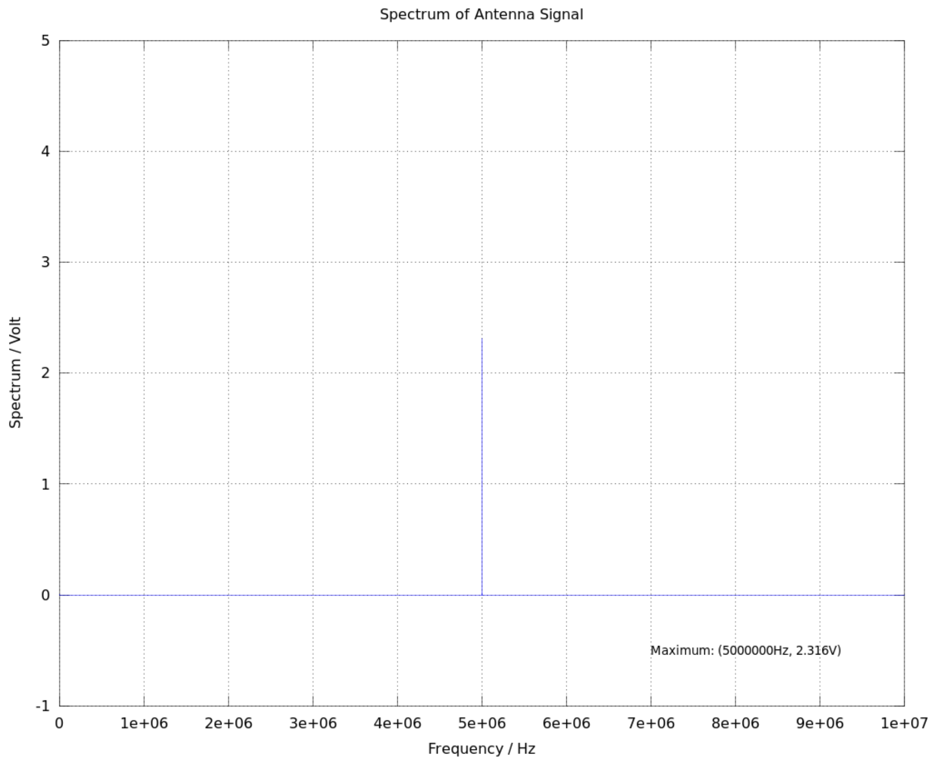

Figure 6.17: Calculated spectrum of test signal

Finally the spectrum is converted to dBm, figure 6.18.

Remark:
The amplitude of the signal was set to 2.5 V at the signal generator. Both, the Oscilloscope as well as the sampled signal show a lower amplitude slightly above 2.3 Volt, see figures 6.16 and 6.17. The reason is that the high input impedances (around 1 MOhm) of Oscilloscope and ADC are connected in parallel, what is leading to a lower impedance $R_{load} = \frac{R_{ADC} \cdot R_{Osc}}{R_{ADC} + R_{Osc}}$ (still in the range of 1 MOhm). On the other side the output impedance (R_{out}) of the generator was set to *high*, what is not further specified in the manual and is the only choice beyond 50 Ohm. A short check proofed that the voltage is correct, if only the Oscilloscope or the ADC is connected. But usually both devices were used in parallel. Thus apparently the signal generator was seeing a slightly lower impedance than expected, what was leading to a slightly lower voltage. (As already discussed a voltage supply can be modelled as an ideal voltage U_{int} in series with an impedance R_{out}. The load impedance is then in series to the supply, thus we have $R = R_{out} + R_{Load}$ connected to U_{int}. For the current we have $I = \frac{U_{int}}{R}$, which is the same for both impedances. We can now calculate the output voltage, i.e. $U_{load} = IR_{load} = \frac{R_{load}}{R_{out}+R_{load}} U_{int}$.)

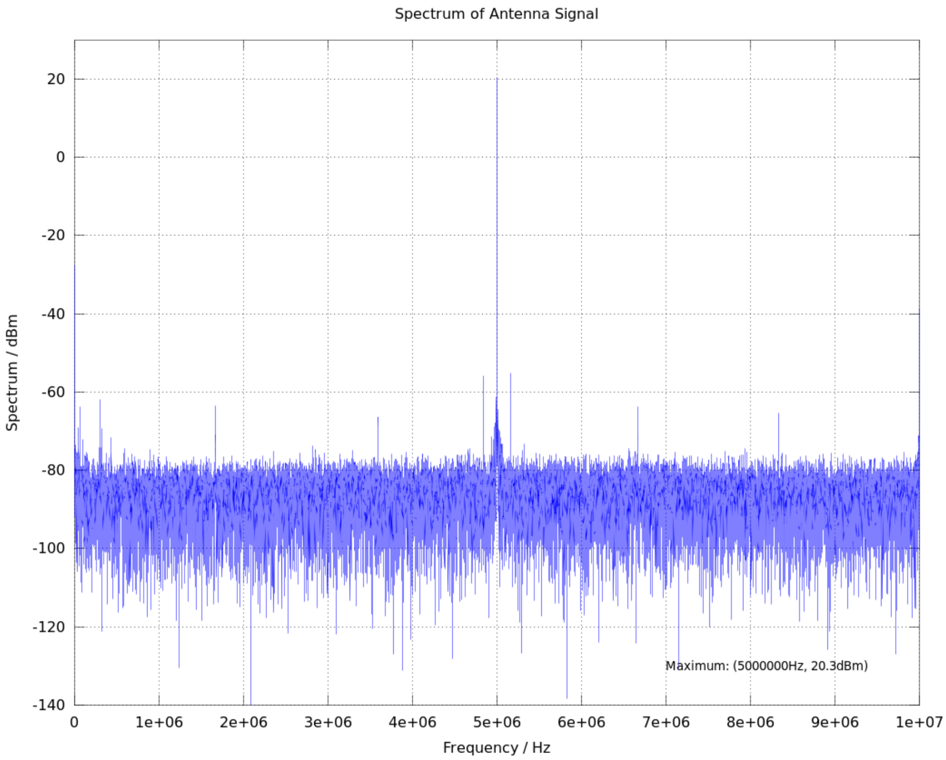

Figure 6.18: Test signal spectrum in dBm

Chapter 7

The complete SRT

7.1 Results

Finally all parts were finished and ready to build the telescope: Parabolic Dish, Feed, LNA, Receiver and PC with Software (inclusive the one from Haystack). Feed, LNA and dish were mounted and connected as shown in the chapter about the antenna. In opposite to the Haystack design Feed and LNA were connected with a short cable, not a fixed SMA connector. This enables to easily separate Feed and LNA, e.g. for performance tests. LNA and receiver were connected with a long coax cable. A shorter cable connects receiver and PC. In parallel the electronics to rotate the dish was connected to the PC.

Furthermore Rohde & Schwarz provided an SME02 Signal Generator and an FSW13 Spectrum Analyzer for the USM labs.

The performance of the SRT was verified step by step.

At the beginning the Signal Generator was connected to the LNA instead of the Feed, while the Spectrum Analyzer was connected to the RF instead of the PC. The Generator was set to 1420 MHz and -86 dB. The measured signal at the SA was about 30 dB lower than in the lab! Since there was no BiasTee available at the USM labs it was not possible to measure LNA and receiver separately. But after some investigations we found that one of the two LNAs in the LNA stage were broken. Instead of amplifying some 15 dB it was attenuating by some 15 dB, its current consumption was approximately 30 mA too low. The broken LNA was removed.
After this the gain of the whole receiver inclusive all cables was measured as 69 dB.

Many parameters like the transmission performance of the Feed, precise flux of the sun etc. were not available. But still it was tried to at least roughly estimate an expectation value for the received radio power of the sun, which served as test object.
According to NASA we can expect from the sun a radio power flux of some $f = 100 \cdot 10^{-22}\,\mathrm{W/(Hz \cdot m^2)}$.
The dish has a radius $r = 1.2\,\mathrm{m}$ and thus an area of $a = \pi r^2 = 4.5\,\mathrm{m}^2$.
Our feed can detect only half of the polarisation, what will be taken into account with a factor 0.5.
Further we have several other losses like: feed beam does not fully cover dish, imperfect

dish, ohmic losses in antenna, feed partly covers dish (shading) etc. All this is combined in an antenna efficiency assumed to be $e = 0.25$.
Our bandwidth is approximately $b = 8\,\text{MHz}$ and the new gain $g = 69\,\text{dB}$, i.e. $g = 10^{6.9} = 7943282$.
Therefore we have for the received power roughly

$$p = f \cdot a \cdot 0.5 \cdot e \cdot b \cdot g \approx 0.36 \cdot 10^{-6}\,\text{W} = 10\log(0.36 \cdot 10^{-3})\,\text{dBm} \approx -34.4\,\text{dBm}$$

Now the antenna (Feed) was connected to the LNA instead of the Signal Generator. The SA measured the power between 1 MHz and 20 MHz (horizontal axis). A marker was set to 5 MHz. Markers can integrate the power of a band around them. This function was used here with a bandwidth of 8 MHz. Marker and band are shown in light blue in the following screen-shots, the band power is shown in the bottom right corner as *Function Result*.

Now the antenna was rotated across the sky until the Spectrum Analyzer showed a minimum.
The result of the cold sky is shown in figure 7.1 with $p = -33.8\,\text{dBm}$.

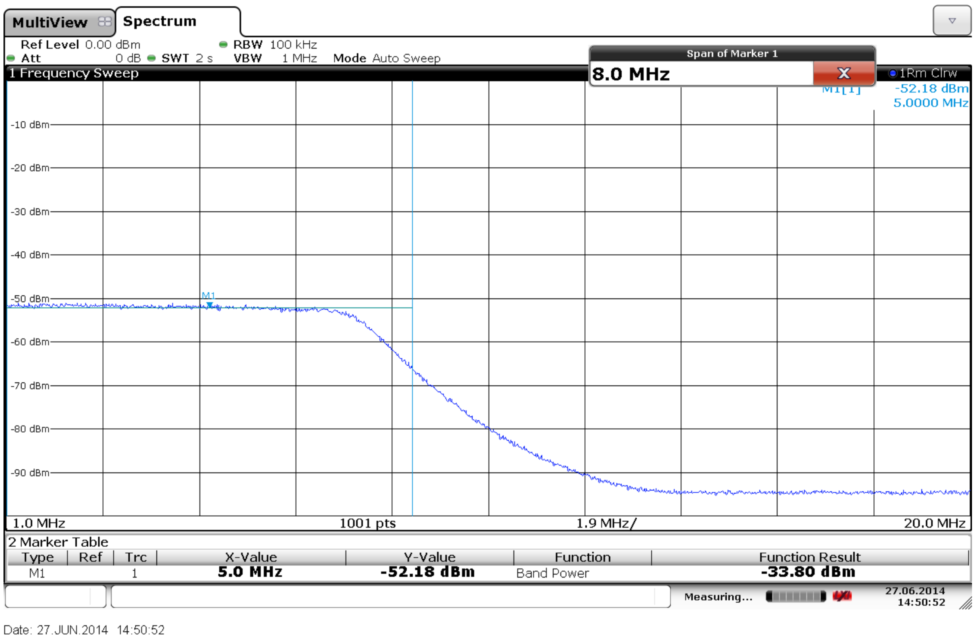

Figure 7.1: Power cold sky

Thus the cold sky plus the noise of our system (which is the dominating contribution) has approximately the same power as was estimated for the sun.

While measuring an object, for example the sun, its power will be added to the power measured for the cold sky.

Since a factor of two is equivalent to 3 dB we can immediately go on with our rough estimation and assume a power while pointing at the sun of approximately $p = -34\,\text{dBm} + 3\,\text{dB} = -31\,\text{dBm}$.

Now the antenna was rotated towards the sun. The measured power is shown in figure 7.2: indeed it almost perfectly meets our expectations with $p = -30.6\,\text{dBm}$.

It was not possible to measure the feed in an antenna chamber, but the results show that the feed seems to work very well.

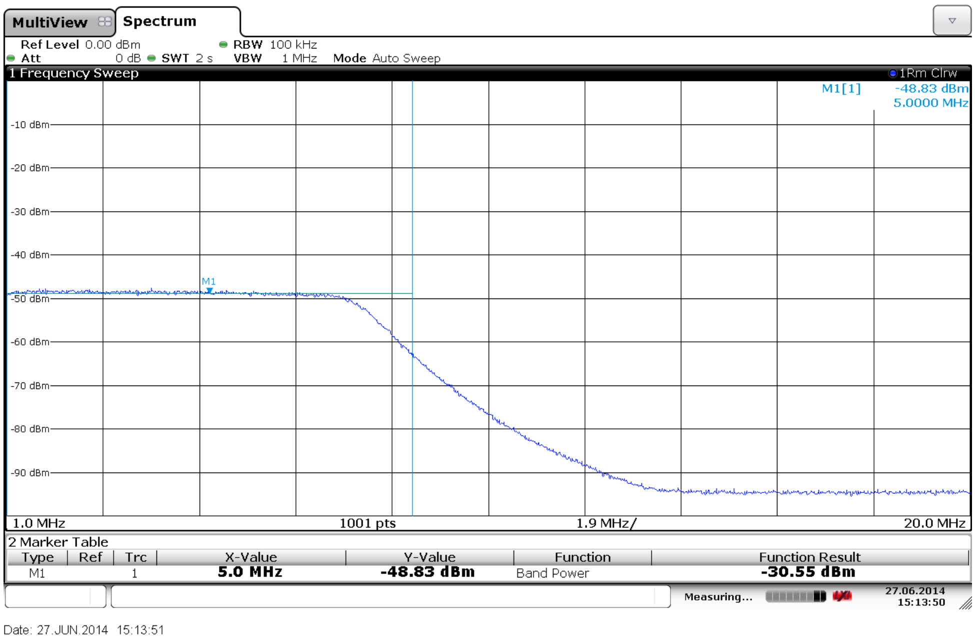

Figure 7.2: Power sun

Finally the Haystack Software was used for measurement. First it measures a reference object for calibration, where again the cold sky was used. Then it measures the object of interest itself. Since the signal of astrophysical objects are usually weaker than the system noise the Software integrates and averages the signal for some time and compares it with the reference. The result of the sun measurement can be seen in figure 7.3. The 21 cm line of the sun can be clearly seen at 1420.4 MHz. The higher peaks are interferers. The Haystack Software is even able to filter so called ***R****adio* ***F****requency* ***I****nterference*. Interferer can be recognized for example if peaks are always coming from the same direction (earthbound) or suddenly change their amplitudes or similar.

Remarks

During the measurements sometimes clouds were covering the sun, which did not influence the received power.

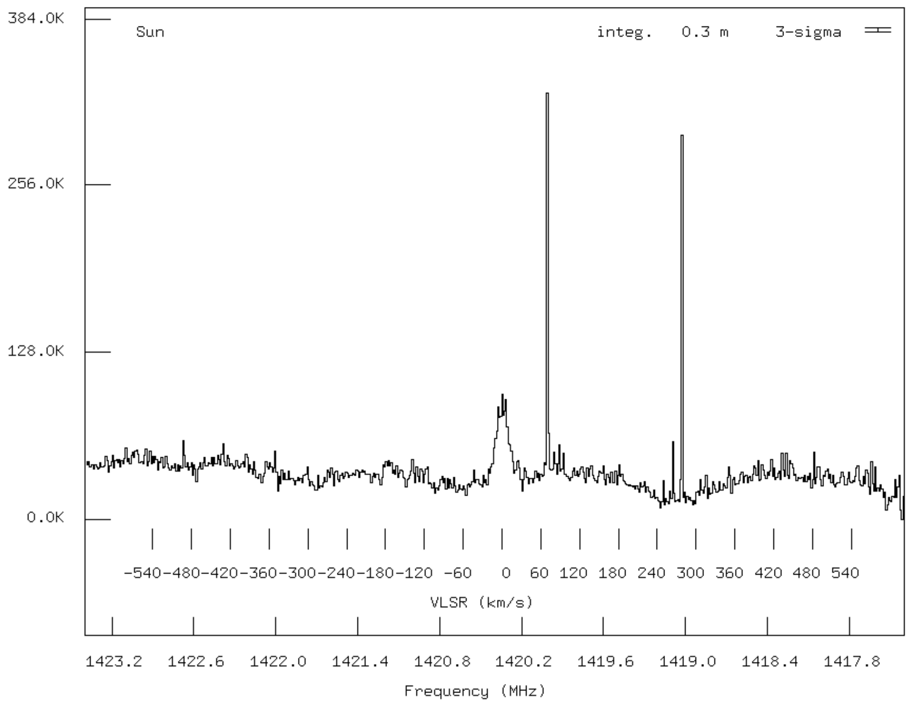

Figure 7.3: Sun measured with the Haystack Software

7.2 Outlook

At this point of time the ten weeks allowed for a thesis are exhausted. The telescope can be considered as finished and shows good performance.

As future tasks beyond this thesis the following items can be considered:

- A practical training for students can be developed based on the telescope.
- The Feed could be covered to further protect it from weather.
- The Antenna can be furthered moved into the park surrounding the USM or even be put on the roof.
- It might be possible to better characterize the antenna with reference signals, measuring the feed in an antenna chamber etc.
- By replacing the broken amplifier instead of just removing it the signal to noise ratio should be improved, but might not be necessary.
- It turned out the power of astrophysical objects is similar to the noise power of the system. Thus the *Analyze* Software is hardly suitable for measurements, since so far the measured signals disappear in the noise floor. It could be further developed, mainly by implementing digital filters, integrating power, taking calibration measurements etc. One advantage would be that MATLAB is available for all platforms. The ADC is delivered with drivers for Windows. Thus the Software would be more platform independent. Other advantages are that MATLAB is easy to understand, feature reach and gives a much better insight of what is going on than a compiled C programme.

Appendix A

Software Tools

A.1 Operating System

Following the Haystack Observatory we decided to use *Linux*. Linux is a free **O**perating **S**ystem. It consists of the kernel, drivers, a ***G**raphical **U**ser **I**nterface* (also called *Desktop (Evironment)*) and Software Packages (like Office etc.). Unfortunately over the years many versions and variants were developed. Those are called *Distributions*. They differ in their look and feel, SW and many other items.
In Unix often a text-based interface is used, for example to start programmes, for administration purposes, to compile SW etc. Therefore a so called *terminal* (*shell*) is provided.
Linux distributions already contain plenty of SW. But further SW can be downloaded as complete packages. Most distributions are coming with their own *package manager* to (de-)install software.
Sometimes SW has to be recompiled in order to run on a distribution and the underlying Hardware. Often programmes are using further packages that have to be found and installed before the program can be compiled.
Many popular SW packages (like Microsoft Office) are not available. But often alternatives exist, many of them are even free.
We chose the *Ubuntu* [Ubunt] distribution, which is one of the most popular today. Another popular one is *SUSE*, but many others are out there, too. Some are not free and are intended for companies and include support etc. The best support for free distributions can be found in the internet.

Some more hints

As said Unix is also coming with a GUI like Microsoft's Windows and can be used with a mouse. But some things in Unix and Windows differ fundamentally:

Text under Windows uses two (hidden) characters for the line end: ***C**arriage **R**eturn* and ***L**ine **F**eed*. Linux is using only the latter one. As result Unix text in a Windows editor as *Notepad* is like one long line, use for instance *Wordpad* to convert (just open and save). On the other hand Windows text under Unix often shows a `^M` at every line end (the unnecessary CR is the 13. ASCII character, M is the 13. character in the alphabet.). Also tools for conversion are available.

Devices under Windows have their own character, like `C:` for the Hard Disc. In Unix HW is accessed as folder in the file system. If for example an USB drive is attached, the drive is *mounted* (made available in the file system) and can then be accessed via a dedicated folder in the existing file system, not via its own device name like `D:`.
Devices have to be unmounted before removing them!

Another important difference concerning files is that in Windows the file-type is coded in the file-extensions (like *.txt* for pure text, *.exe* for executables). Under Unix this information is not encoded in the filename, but the file itself and / or a set of information, which is saved along with every file. This also contains information like who is the owner, who can read or save this file and much more.
But encoded in the filename is the information, whether the file is *hidden*, i.e. by default not listed in the file manager. Hidden files simple start with a dot in their filename.

A.2 The C Programming Language

The drivers for HW are usually written in C. With C the highest performance (speed, memory usage, ...) can be achieved (only an assembler program should outperform C).
Normally a further program has to be written to handle the device with the help of the driver. For the SRT the *sample_signal.c* was developed. It was edited with *gedit*.

Outlook:

The PCI-Card came also with Windows drivers. It might be interesting to rewrite the SW and use Windows instead of Linux.

A.3 Mathematics Tools and Lab Software

For mathematics the probably most popular SW is *Mathematica*® from *Wolfram*. Another very famous tool is *Maple*® from *Maplesoft*. These packages can even perform symbolic calculations, solve integrals or partial differential equations, it is possible to automatize tasks (program) and many more.
Often symbolic calculation is not necessary (to being automatized, or can be looked up in books, ...).
But a SW tool is desired for (numerical) calculations, simulations, ..., which should be easy to use, has an easy syntax etc. This is the strength of *MATLAB*® by MathWorks [MatL].
In addition to this with MATLAB it is easy to talk to bus systems, especially those like *GPIB*, which are used in Labs to connect PCs and instruments. Thus Lab-Hardware can easily be accessed and also tasks like measuring, controlling etc. can be automated. Since MATLAB is very popular lots of further packages are developed around it, for example to design filters and many others.
MATLAB provides also great possibilities for visualizing data.
Sometimes MATLAB (as all the other tools) can be too slow for certain tasks. Those tasks can be implemented for example in C and can then be called from MATLAB.
An alternative to MATLAB is the free *Octave* / *QtOctave* combination. It can handle the same syntax, offers almost the same look and feel, but is not as feature-rich. The Ubuntu

packages are available and used here.
Another popular SW is *LabView*®: it follows a different approach where SW is developed graphically, meaning the structure of the program is drawn. For example a `for` loop is a window that contains the tasks.
The following chapters give some hints about installing the driver for the ADC, how to use Ubuntu and the SW tools.

Appendix B

Ubuntu and Octave

Octave is a mathematics program that works via a command line or programmes, meaning in pure text mode. To offer a modern ***G****raphical* ***U****ser* ***I****nterface* the free package *QtOctave* was developed. QtOctave provides a similar look-and-feel like MATLAB and is used here. It seems that QtOctave is not developed any further, but another GUI for Octave is under development.

Ubuntu offers the so called *launcher* to access programmes, the vertical bar on the left hand side shown in figure B.1.

To start QtOctave click on its icon on the launcher, in figure B.1 the second symbol from top left.
If a program is running it is indicated by white triangles next to the icon.
The GUI consists of several Windows. Unfortunately QtOctave always de-arranges them after start. But they can be arranged easily: On their top are shown the typical buttons for making windows smaller or close them. In addition the relative sizes and positions can be adjusted by moving the grey bars, that separate the windows, with the mouse.
After the start the program Analyse.m should already be loaded in the editor.

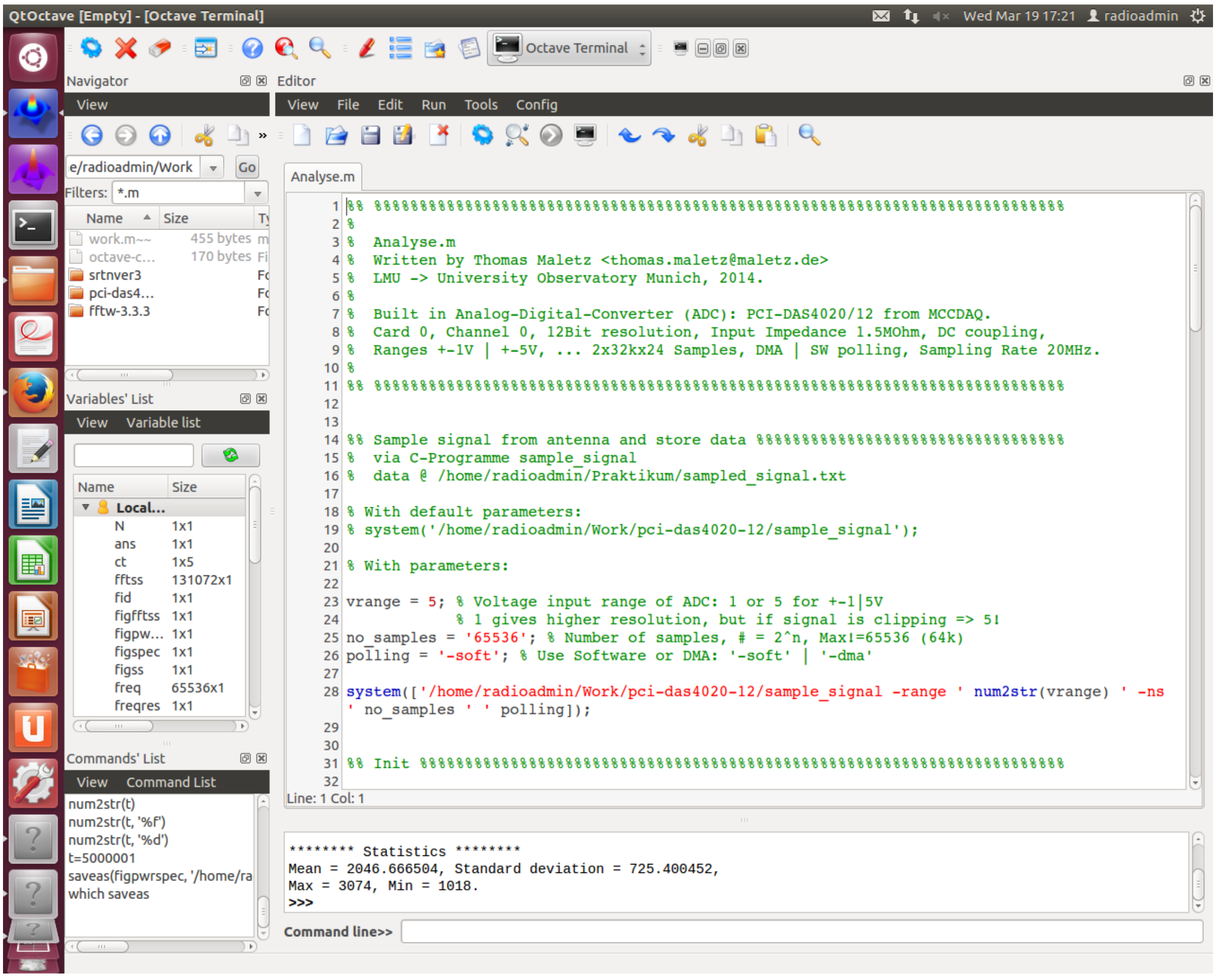

Figure B.1: QtOctave

On the very top of the screen Ubuntu provides a global menu bar. If the mouse is hovering over the bar it shows a menu for the active program, please compare the previous with the next picture B.2.
The typical buttons to resize or close a window appear on the left (compared to Windows, where they are located on the right hand side of the title bar).

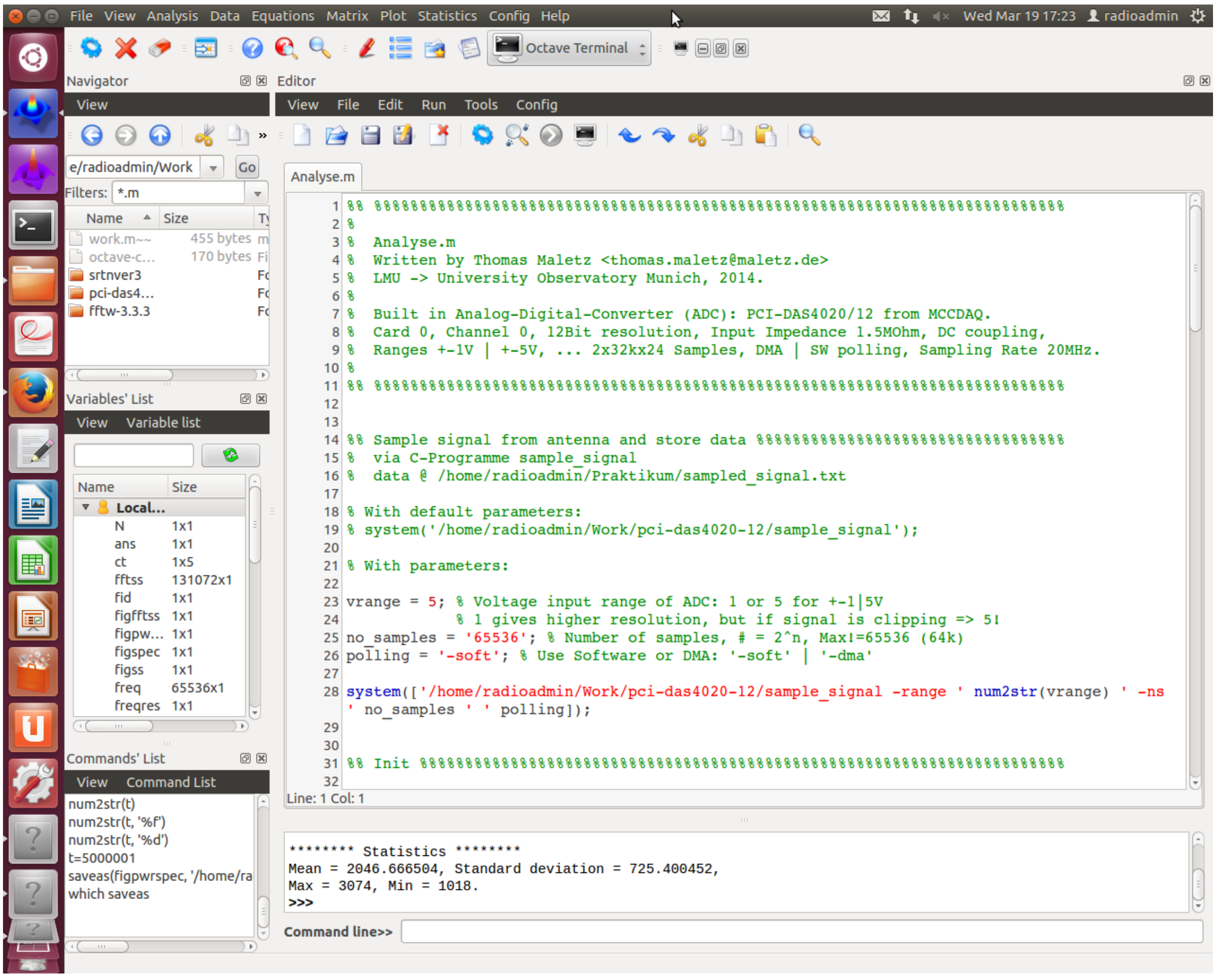

Figure B.2: QtOctave Menubar

Windows overview

- Left Top: File manager.
- Left Middle: List of all variables, including their type, values, ... A double click on a variable opens a new window / table where the content is shown.
- Left Bottom: History of commands.
- Right Bottom Line: Command line.
- Right Bottom Window: Output window, showing results, errors, ...
- Right Top: Editor.

Working with Octave

In the command line single commands can be entered, like `1+1`, followed by Return / Enter.
The result is shown in the window above.
If a new variable is introduced, say `a=0`, the variable appears in the variable list window as well.

If a command ends with a `;` the output is suppressed. For example `a=0` responses with `a=0`, while a `a=0;` does not show a result.
The editor can be used to collect commands and write programmes. It is not possible to give a full introduction to the syntax here, but it is easy to understand and to learn. Explanations can be found in the internet and the given code can be used as reference.

Syntax: Some hints for the start

- Comments start with %.
- Pressing F5 runs the whole program.
- It is possible to mark something [1] and by pressing F9 to execute exactly the marked commands. Several lines can be marked. F9 even works inside comments, example: If in the line `% a=0;` only `a=0;` is marked (without the `%`), F9 will execute `a=0;`.
- Variables can be introduced everywhere in the program. A string (text) is marked with `'`, example `name='LMU';`.
- The command `system(.)` calls an external program on the PC.
- Too break a long line in the editor into two (or more) three dots '...' can be inserted and the line continues in the next line of the editor. During execution the second line is concatenated to the first instead of the dots. In other words the dots and the line break are removed and the lines are being treated as one.
- Analyse.m contains lots of comments to make it easy to work with and self-explaining.

Analyse.m shows results in pictures: Those windows appear on the launcher as icons with `?`.

[1] With *mark* the typical marking inside an editor is meant, for instance: left click with the mouse at the beginning of the desired text, keep the button pressed and go with he mouse pointer to the end of the desired text, release the button. The same can be achieved with the keyboard by keeping the *shift* key pressed while moving the cursor from the beginning to the end of the text that shall be marked.

File Manager

To see and handle files and directories the file manager *nautilus* is available:
It is started via the launcher as well (5. icon from top).
Double click on `Praktikum` icon to go to the folder where Analyse.m stores all data. Please see second picture.

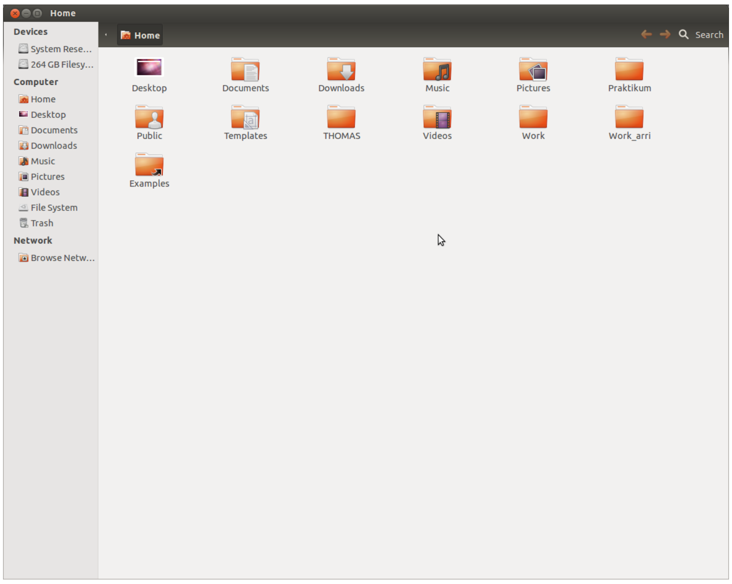

Figure B.3: File Manager: Home

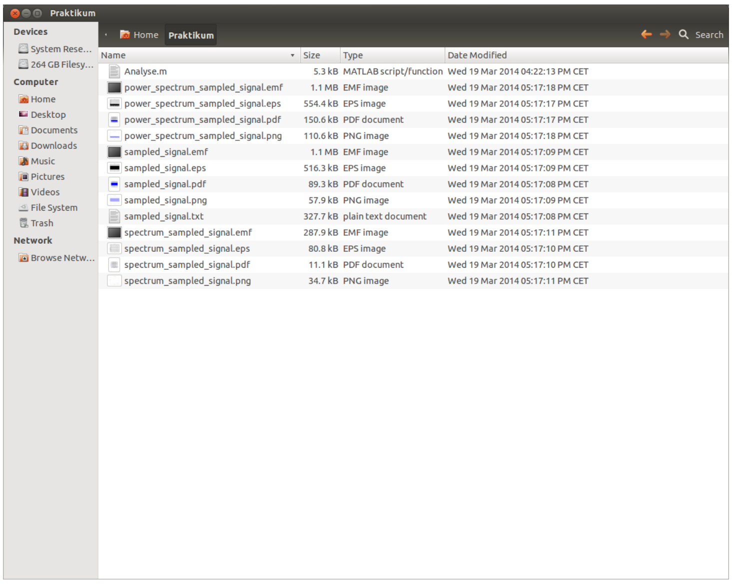

Figure B.4: File Manager: Praktikum

Further available tools

- Terminal (shell), 4. icon from top on the launcher.
- Internet, browser is Firefox, 7. icon from top on the launcher.
- Texteditor *gedit*: 8. icon from top on the launcher.
- Viewer (e.g. PDF): 6. icon from top on the launcher.

Programming in C

For editing C Programmes *gedit* is available.
For compiling often a *Makefile* is used: it gives the compiler directives what to do. Here the Makefile that came with the PCI-card driver was appended with the *sample_signal.c* to compile all. For compilation just start a terminal (shell), go to the directory (command `cd directory`) and type `make`.
To directly compile a program without using a Makefile use the command `gcc file.c -o file`. This is calling the compiler that translates and links *file.c* into the executable *file*.

Appendix C

Ubuntu and the ADC PCI-Card

The driver for the PCI Card was developed by Prof. Dr. Warren Jasper.

The first step is to find a driver (version) that works with the used Linux kernel. In our case the one delivered with the Haystack SW was an older version and not compatible with our system.

Prof. Dr. Jasper offers an ftp site for his drivers:
ftp://lx10.tx.ncsu.edu/pub/Linux/drivers/PCI

The driver is coming in a package that includes several files as C source code, manual with further details etc. In order to use the driver it has to be compiled and installed.

Before compilation two changes in two different files of the package were needed:

In file *a2dc.h* two lines were adjusted to

```
#define ADC_BUFF_PHY_SIZE 0x0180000
#define ADC_BUFF_PAGE_SIZE 0x0180000
```

Furthermore the file *a2dc.c* contained compiler directives that check the kernel version. This had to be adjusted as well.

To compile the software and install the driver the following commands were used.
(Some programmes need administrator rights, for example to write to certain system directories (here installing the driver). You can give a program *admin* rights by starting them via `sudo`.)

```
make
sudo make install
```

Ubuntu comes with a driver package called *Comedi*, that can handle data acquisition hardware, too.
Furthermore Ubuntu has a so called blacklist that lists drivers, which are not allowed to start (while booting).
The following Comedi drivers were blacklisted to ensure they do not interfere with our driver:
In the file `/etc/modprobe.d/blacklist.conf` the following lines were appended

```
blacklist cb_pcidas64
blacklist 8255
blacklist comedi_fc
blacklist comedi
```

In addition to make sure that the systems loads our driver following files were created in dedicated directories

Change to directory:

```
cd /etc/modprobe.d
```

Edit / create (administrator rights needed):

```
sudo gedit das4020_12.conf
```

Enter

```
alias char-major-250 pci-das4020-12
```

and save.

Change to directory:

```
cd /etc
```

Edit (administrator rights needed):

```
sudo gedit modules
```

Append

```
das4020_12
```

and save.

Furthermore the file `61-mcc.rules` was copied to `/etc/udev/rules.d` to enable driver access to programmes without administrator rights.

To install new dependencies of drivers, packages etc. finally the following commands were executed:

```
sudo udevadm control --reload-rules
sudo udevadm trigger
```

After a reboot

```
modinfo das4020_12
```

should ping the driver and should return information.

Appendix D

Software and Source Code

D.1 C Program

The C program *sample_signal.c* was developed to get data from the ADC via the ADC driver.
sample_signal also covers several parameters.
Some of the parameters shall not be changed and are fixed in the source code, for example the sample frequency is set to 20 MHz, which is the available maximum of the ADC.
Other parameters, like the number of data samples, might be useful or even have to be changed for certain tasks. They can be handed over to sample_signal during the call of the program.
For descriptions and examples on the usage of sample_signal please see the Octave code below.
Sample_signal provides the measured data in the file *sampled_signal.txt*.

The sample_signal.c source code can also be downloaded from: http://www.maletz.de/srt/sample_signal.c.

```
/****************************************************************************
 *
 * Written by Thomas Maletz <thomas.maletz@maletz.de>.
 * LMU -> University Observatory Munich, 2014.
 * Partly derived from Warren Jasper's test-das4020.c.
 *
 ****************************************************************************/

/****************************************************************************
 *
 * sample_signal.c
 *
 * This program uses the PCI-DAS4020/12 Analog to Digital Card from MCCDAQ:
 * Card 0, Channel 0, 12Bit resolution, Input Impedance 1.5MOhm, DC coupling,
 * Ranges +-1V | +-5V, ... 2x32kx24 Samples, DMA | SW polling, Sampling Rate 20MHz.
 * Data are stored in /home/radioadmin/Praktikum/sampled_signal.txt.
 *
 ****************************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include "pci-das4020.h"
#include <sys/mman.h>

/****************************************************************************
 *
 *      Global Data, Functions
 *
 ****************************************************************************/
```

```

char resultfile[] = "/home/radioadmin/Praktikum/sampled_signal.txt"; // where to save sampled signal

int ADC_Mode = ADC_DMA_CONVERSION; // ADC_SOFT_CONVERSION | ADC_DMA_CONVERSION;
int Count = 0x0010000; // #samples ("ns"). Default: 65536 samples = 64k = 0x0010000 (hex)
int Gain = BP_5_00V;    // #define BP_5_00V (0xf0) // +-5V for all 4 channels (0xf0 = 240)
                        // #define BP_1_00V (0x00) // +-1V for all 4 channels
int Board = 0;
int Channel = 0;
int NoStop = 0;
int freq_A2D = 20000000; // Max. sample freq = 20MHz

int fdADC, fdADC0; // AD file descriptors

float full_scale = 5.0;

FILE *fid;

void Usage(void)
{
        fprintf(stderr, "\n\
        Usage: \n\
        Options:\n\
           [-range 1|5]     - Voltage input range (1 or 5 for +-1|5V)\n\
           [-ns ##]         - Number of samples to read\n\
           [-soft | -dma]   - Use software triggers or dma and pacer clock triggers\n\
           [-nostop]        - Sample forever (dummy function)\n");
        exit(1);
}
// option nostop just prepared, in case samples not stored in file but read out continuously


void CommandLineArgs(int argc, char **argv)
{
        int i = 1;

        while (i < argc)
        {
                if (strcmp(argv[i], "-range") == 0)
                {
                        i++;
                        if (i == argc)
                                Usage();
                        else
                                if (atoi(argv[i]) == 1)
                                        Gain = BP_1_00V;
                                else if (atoi(argv[i]) == 5)
                                        Gain = BP_5_00V;
                                else
                                        Usage();
                }
                else if (strcmp(argv[i], "-ns") == 0)
                {
                        i++;
                        if (i == argc)
                                Usage();
                        else
                                Count = atoi(argv[i]);
                }
                else if (strcmp(argv[i], "-soft") == 0)
                        ADC_Mode = ADC_SOFT_CONVERSION;
                else if (strcmp(argv[i], "-dma") == 0)
                        ADC_Mode = ADC_DMA_CONVERSION;
                else if (strcmp(argv[i], "-nostop") == 0)
                        NoStop = 0; // just prepared
                else
                        Usage();
                i++;
        } // end while
}

void OpenDevices()
{
        char str[80];
        int adc_chan;
        int *adc_fds[] = {&fdADC0};

        for(adc_chan = 0; adc_chan <= 0; adc_chan++)
        {
                sprintf(str, "/dev/das4020-12/ad%d_%d", Board, adc_chan);
                if ( (*adc_fds[adc_chan] = open(str, ADC_Mode)) < 0 )
                {
                        perror(str);
                        printf("error opening device %s\n", str);
                        exit(2);
                }
        }

        ioctl(fdADC0, ADC_SET_GAINS, BP_5_00V);

        fdADC = *adc_fds[Channel];
}

void ADC()
{
        unsigned short value[1024*64];
        int i;
```

```
	int bytesRead;
	float mean, sd;
	unsigned short max, min;
	unsigned short *readBuff;
	unsigned int toggle = 0x0;

	ioctl(fdADC, ADC_SET_PACER_FREQ, freq_A2D);
	ioctl(fdADC, ADC_GET_HALF_FIFO_SIZE, &i);
	printf("The size of half the FIFO in bytes = %d,\n", i);
	ioctl(fdADC, ADC_GET_DMA_BUF_SIZE, &i);
	printf("The size of the DMA buffer = %d.\n", i);

	if (ADC_Mode == ADC_SOFT_CONVERSION)
	{
		printf("ADC_SOFT_CONVERSION\n");

		while (1)
		{
			toggle ^= 0x1;
			ioctl(fdADC, ADC_PSC_ENB, toggle);
			bytesRead = read(fdADC, value, Count);

			if (bytesRead != Count)
			{
				printf("ADC: Error on read():\n");
				printf("bytesRead = %d, but specified number of samples = %d!\n", bytesRead, Count);
			}

			printf("%d samples read.\n", bytesRead);

			for (i=0; i<Count; i++)
			{
				fprintf(fid, "%d\n", value[i]);
			}

			printf("\n******** Statistics ********\n");

			mean = 0.0;
			// set as starting point max to min-value and min to max-value:
			max = 0;
			min = 4096; // 12 Bits: 2^12 = 4096

			for (i=0; i<Count; i++ )
			{
				mean += (float) value[i];
				if (value[i] > max)
					max = value[i];
				if (value[i] < min)
					min = value[i];
			}
			mean /= (float) Count;

			sd = 0.0;

			for (i=0; i<Count; i++)
			{
				sd += ((float) value[i] - mean)*((float) value[i] - mean);
			}
			sd = sqrt(sd / ((float) Count - 1.0));

			printf("Mean = %f, Standard deviation = %f,\n", mean, sd);
			printf("Max = %d, Min = %d.\n", max, min);

			return;

		} // end while
	} // end if
	else if (ADC_Mode == ADC_DMA_CONVERSION)
	{
		printf("ADC_DMA_CONVERSION\n");

		if( (readBuff= mmap(0, Count*2, PROT_READ, MAP_PRIVATE, fdADC, 0*getpagesize()))
				== (unsigned short *) MAP_FAILED )
		{
			printf("Test Failed: Mmap call failed!\n");
			sleep(3);
			return;
		}
		else
		{
			printf("Test Passed: Succesfully mmaped %d bytes.\n", Count*2);
		}

		/* In the following read calls, the argument value will be ignored */
		/* Since we DMA to stuff over to the address held by readBuff      */

		while(1)
		{
			bytesRead = read(fdADC, value, Count);

			if (bytesRead != Count)
			{
				printf("ADC: Error on read():\n");
				printf("bytesRead= %d, but specified number of samples = %d!", bytesRead,Count);
			}


```

```
                        printf("%d samples read.\n", bytesRead);

                        for (i=0; i<Count; i++)
                        {
                                fprintf(fid, "%d\n", readBuff[i]);
                        }

                        printf("\n******** Statistics ********\n");

                        mean = 0.0;
                        // set as starting point max to min-value and min to max-value:
                        max = 0;
                        min = 4096; // 12 Bits: 2^12 = 4096

                        for (i=0; i<Count; i++)
                        {
                                mean += (float) readBuff[i];
                                if (readBuff[i] > max)
                                        max = readBuff[i];
                                if (readBuff[i] < min)
                                        min = readBuff[i];
                        }
                        mean /= (float) Count;

                        sd = 0.0;

                        for (i=0; i<Count; i++)
                        {
                                sd += ((float) readBuff[i] - mean)*((float) readBuff[i] - mean);
                        }
                        sd = sqrt( sd / ((float) Count - 1.0) );

                        printf("Mean = %f, Standard deviation = %f,\n", mean, sd);
                        printf("Max = %d, Min = %d.\n", max, min);

                        if (munmap(readBuff, 2*Count) == -1)
                        {
                                printf("munmap failure!\n");
                                sleep(3);
                        }
                        return;

                } // end while
        } // end else
}

void ChangeADCGains()
{
        ioctl(fdADC0, ADC_SET_GAINS, Gain);
        switch(Gain)
        {
        case 5:
                full_scale = 5.0;
                break;
        case 1:
                full_scale = 1.0;
                break;
        }
}

int main(int argc, char **argv)
{
        CommandLineArgs(argc, argv);
        OpenDevices();
        ChangeADCGains();
        fid = fopen(resultfile, "w"); // "w" for (over)writing
        ADC();
        fclose(fid);
        close(fdADC0);

        return 0;
}
```

D.2 Octave (MATLAB)

The Octave (MATLAB) program *Analyse.m* was developed to read in the data from sampled_signal.txt, analyse it and visualize and save the results.

All explanations are implemented as comments inside the program, to make the SW self-explaining while working with it.

The Analyse.m source code can also be downloaded from: http://www.maletz.de/srt/Analyse.m.

```
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  Analyse.m
%  Written by Thomas Maletz <thomas.maletz@maletz.de>
%  LMU -> University Observatory Munich, 2014.
%
%  Built in Analog-Digital-Converter (ADC): PCI-DAS4020/12 from MCCDAQ.
%  Card 0, Channel 0, 12Bit resolution, Input Impedance 1.5MOhm, DC coupling,
%  Ranges +-1V | +-5V, ... 2x32kx24 Samples, DMA | SW polling, Sampling Rate 20MHz.
%
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%% Sample signal from antenna and store data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  via C-Programme sample_signal
%  data @ /home/radioadmin/Praktikum/sampled_signal.txt

% With default parameters:
% system('/home/radioadmin/Work/pci-das4020-12/sample_signal');

% With parameters:

vrange = 5; % Voltage input range of ADC: 1 or 5 for +-1|5V
            % 1 gives higher resolution, but if signal is clipping => 5!
no_samples = '65536'; % Number of samples, # = 2^n, Max!=65536 (64k)
polling = '-soft'; % Use Software or DMA: '-soft' | '-dma'

system(['/home/radioadmin/Work/pci-das4020-12/sample_signal ...
-range ' num2str(vrange) ' -ns ' no_samples ' ' polling]);


%% Init %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fclose all; % close figures (e.g. left over from former run)
format long;


%% Read in sampled signal %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fid = fopen('/home/radioadmin/Praktikum/sampled_signal.txt');
sampled_signal = fscanf(fid, '%d');
fclose(fid);

N = length(sampled_signal); % should be like no_samples

% scale to voltage:
% 12Bit resolution, 2^12 = 4096, i.e.
% values 0 ... 4095 = Voltage -Min ... +Max (with 0 @ 2048)
sampled_signal = sampled_signal-2048; % shift from positive to around 0
sampled_signal = vrange*sampled_signal/2048; % scale from 12 Bits to Voltage-Range
```

```
%% Plot data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figss = figure('Position', [0 0 1200 1000]);
plot([1:N], sampled_signal);
axis([1 N -vrange vrange]); % signal in full range
grid on;
title('Sampled Antenna Signal');
xlabel('Samples');
ylabel('Signal / Volt');

print(figss, '/home/radioadmin/Praktikum/sampled_signal.pdf', '-dpdf');
print(figss, '/home/radioadmin/Praktikum/sampled_signal.eps', '-deps');
print(figss, '/home/radioadmin/Praktikum/sampled_signal.png', '-dpng');
print(figss, '/home/radioadmin/Praktikum/sampled_signal.emf', '-demf');


%% Spectrum Analyzer %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Create Hann-Window, insert Zero-Padding:
winh = transpose(0.5 * ( 1 - cos( (2*pi*(0:(N-1))) / (N-1) ) ));
sampled_signal = sampled_signal.*winh; % multiply vectors componentwise
sampled_signal = [sampled_signal; zeros(N,1)]; % double number of calculated freq. bins

fftss=fft(sampled_signal);
% scale result:
% 2 for window, 2 for zero padding and length(fftss) because fft() doesn't do it
fftss=4*fftss/length(fftss);

% calculate single sideband spectrum
spectrum=abs(fftss( 1:(length(fftss)/2) ));
for i=1:((length(fftss)/2)-1)
spectrum(i+1)=spectrum(i+1)+abs(fftss(length(fftss)+1-i));
end

% calculate frequency:
freqres=20e6/(2*N); % resolution = samplingrate/no_samples
freq=0:freqres:(10e6-freqres); % range from DC to Samplingfreq/2 = 10MHz
freq=freq'; % transpose


%% Plot Spectrum %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figspec = figure('Position', [0 0 1200 1000]);
plot(freq, spectrum, "b");
% Spectrum: from DC to fs/2 = 10MHz, in full range:
axis([0 (10e6-freqres) -vrange/5 vrange]);
grid on;
title('Spectrum of Antenna Signal');
```

```
xlabel('Frequency / Hz');
ylabel('Spectrum / Volt');

% Print Max!
[maxyspec maxxspec] = max(spectrum);
maxfspec = freq(maxxspec);
text(7e6, -vrange/10, ['Maximum: (' num2str(maxfspec, '%d') 'Hz, ...
' num2str(maxyspec, '%2.3f') 'V)'], 'fontweight', 'bold');

print(figspec, '/home/radioadmin/Praktikum/spectrum_sampled_signal.pdf', '-dpdf');
print(figspec, '/home/radioadmin/Praktikum/spectrum_sampled_signal.eps', '-deps');
print(figspec, '/home/radioadmin/Praktikum/spectrum_sampled_signal.png', '-dpng');
print(figspec, '/home/radioadmin/Praktikum/spectrum_sampled_signal.emf', '-demf');


%% And in dBm %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Voltage @ 50Ohm => P = UI = U^2/R
power_spectrum = spectrum.*spectrum/50;

% in dB / mW = dBm
power_spectrum = 10*log10(power_spectrum/(1e-3));

figpwrspec = figure('Position', [0 0 1200 1000]);
% semilogx(freq, power_spectrum); % plot with logarithmed x-Axis
plot(freq, power_spectrum);
% Spectrum: from DC to fs/2 = 10MHz, in full range:
xlim([0 (10e6-freqres)]); ylim([-140 30]);
grid on;
title('Spectrum of Antenna Signal');
xlabel('Frequency / Hz');
ylabel('Spectrum / dBm');

% Print Max!
[maxypspec maxxpspec] = max(power_spectrum);
maxfpspec = freq(maxxpspec);
text(7e6, -130, ['Maximum: (' num2str(maxfpspec, '%d') 'Hz, ...
' num2str(maxypspec, '%3.1f') 'dBm)'], 'fontweight', 'bold');

print(figpwrspec, '/home/radioadmin/Praktikum/power_spectrum_sampled_signal.pdf', '-dpdf');
print(figpwrspec, '/home/radioadmin/Praktikum/power_spectrum_sampled_signal.eps', '-deps');
print(figpwrspec, '/home/radioadmin/Praktikum/power_spectrum_sampled_signal.png', '-dpng');
print(figpwrspec, '/home/radioadmin/Praktikum/power_spectrum_sampled_signal.emf', '-demf');
```

Bibliography

[USM] http://www.usm.uni-muenchen.de/

[ESO] http://www.eso.org/

[Hay] http://www.haystack.mit.edu/edu/undergrad/srt/

[KRF] http://krfilters.com/

[MatL] http://www.mathworks.de/

[McADC] http://www.mccdaq.com/

[Card] http://www.mccdaq.com/pci-data-acquisition/PCI-DAS4020-12.aspx

[RFham] http://www.rfhamdesign.com/

[Ubunt] http://www.ubuntu.com/

[Wiki] http://www.wikipedia.org/

[AHKKLS] Tilo Arens, Frank Hettlich, Christian Karpfinger, Ulrich Kockelkorn, Klaus Lichtenegger, Hellmuth Stachel, *Mathematik*, Spektrum, Heidelberg, 1. Auflage 2008.

[Bala] Constantine Balanis*Antenna Theory*, John Wiley & Sons, Hoboken, 3. edition 2005.

[CarHip] Joseph Carr, George Hippisley, *Practical Antenna Handbook*, McGraw-Hill, New York, 5. edition 2012.

[CaOs] Bradley Carroll, Dale Ostlie, *An Introduction to Modern Astrophysics*, Addison-Wesley, Pearson, San Francisco, 2. edition 2007.

[Foot] Christopher Foot, *Atomic Physics*, Oxford University Press, Oxford, Reprint 2011.

[Gian] Douglas Giancoli, *Physik*, Pearson, München, 3. Auflage 2010.

[Goetz] Hermann Götz, *Einführung in die digitale Signalverarbeitung*, Teubner, Stuttgart, Leipzig, 3. Auflage 1998.

[Grein] Walter Greiner, *Klassische Elektrodynamik*, Harri Deutsch, Frankfurt am Main, 6. Auflage 2002.

[Griff] David Griffiths, *Introduction to Electrodynamics*, Pearson, Boston, 4. edition 2013.

[Gruen] Daniel von Grünigen, *Digitale Signalverarbeitung*, Hanser, München, 4. Auflage 2008.

[HaWo] Hermann Haken, Hans Christoph Wolf, *Atom- und Quantenphysik*, Springer, Berlin, Heidelberg, 8. Auflage 2004.

[JoRo] Dustin Johnson, Alan Rogers, *Development of a New Generation Small Radio Telescope*, Haystack Observatory, Final Report, 10. August 2012.

[Kal] Georg Michael Kalvius, *Physik IV*, Oldenbourg, München, 5. Auflage 1999.

[KamKro] Karl-Dirk Kammeyer, Kristian Kroschel, *Digitale Signalverarbeitung*, Springer Vieweg, Wiesbaden, 8. Auflage 2012.

[Kark] Klaus Kark, *Antennen und Strahlungsfelder*, Vieweg+Teubner, Wiesbaden, 4. Auflage 2011.

[Keen] Keen, K.M., *Low profile cavity antenna with small pitch angle helix feed.*, Electronics Letters, 4 March 1993, Vol. 29, No. 5, pp. 501-502.

[KraMar] John Kraus, Ronald Marhefka, *Antennas: For All Applications*, McGraw-Hill, New York, 3. edition 2003.

[KriRot] Alois Krischke, *Rothammels Antennenbuch*, DARC, Baunatal, 13. Auflage 2013.

[Lee] Thomas Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press, Cambridge, Reprint 2000.

[Lyon] Richard Lyons, *Understanding Digital Signal Processing*, Prentice Hall, Pearson, Boston, 3. edition 2011.

[Maka] Sergey Makarov*Antenna and EM Modeling with MATLAB*, John Wiley & Sons, New York, 1. edition 2002.

[MaeSch] Rudolf Mäusl, Erhard Schlagheck, *Meßverfahren in der Nachrichtenübertragungstechnik*, Hüthig, Heidelberg, 2. Auflage 1991.

[Mey] Martin Meyer, *Signalverarbeitung*, Vieweg+Teubner, Wiesbaden, 6. Auflage 2011.

[MeyVa1] Kurt Meyberg, Peter Vachenauer, *Höhere Mathematik 1*, Springer, Berlin, Heidelberg, 6. Auflage 2001.

[MeyVa2] Kurt Meyberg, Peter Vachenauer, *Höhere Mathematik 2*, Springer, Berlin, Heidelberg, 4. Auflage 2003.

[Mue1] Rudolf Müller, *Grundlagen der Halbleiter-Elektronik*, Springer, Berlin, Heidelberg, 7. Auflage 1995.

[Mue2] Rudolf Müller, *Bauelemente der Halbleiter-Elektronik*, Springer, Berlin, Heidelberg, 4. Auflage 1991.

[ProMan] John Proakis, Dimitris Manolakis, *Digital Signal Processing*, Prentice Hall, Pearson, Upper Saddle River, 4. edition 2007.

[Schneid] Peter Schneider, *Einführung in die Extragalaktische Astronomie und Kosmologie*, Springer, Berlin, Heidelberg, Korrigierter Nachdruck 2008.

[Schrue] Elmar Schrüfer, *Elektrische Meßtechnik*, Hanser, München, 5. Auflage 1992.

[TiMo] Paul Tipler, Gene Mosca, *Physik*, Spektrum, Elsevier, München, 2. Auflage 2006.

[UnBa] Albrecht Unsöld, Bodo Baschek, *Der Neue Kosmos*, Springer, Berlin, Heidelberg, Korrigierter Nachdruck 2005.

[Web] Robert Weber, *Introduction to Microwave Circuits*, IEEE Press, New York, 2001.

[ZiBr] Otto Zinke, Heinrich Brunswig, *Hochfrequenztechnik 1*, Springer, Berlin, Heidelberg, 5. Auflage 1995.

Deklaration

Hiermit erkläre ich, dass ich die Arbeit selbstständig verfasst habe, und dass ich keine anderen Quellen und Hilfsmittel, als die angegebenen, benutzt habe.

Unterhaching, 22. Oktober 2014, Thomas Maletz (Matrikelnummer 10705251)