



**Hochschule
Koblenz**
University of Applied Sciences

Fachbereich Elektro- und Informationstechnik

**Max-Planck-Institut
für Radioastronomie**



Masterarbeit

im Masterstudiengang Systemtechnik

Entwurf und Implementierung eines Linux-basierten Kontrollsystems für ein 4-Meter-Radioteleskop

**von Sebastian Schweikert
Matrikel-Nr.: 521875**

Hochschulbetreuer: Prof. Dr. Gärtner

Institutsbetreuer: Dr. Keller, Dr. Hafok

Institut: Max-Planck-Institut für Radioastronomie

Eingereicht am: 12. September 2018

Eidesstattliche Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel, insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-Rom).

Remagen, den 12. September 2018

Sebastian Schweikert

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während des Verlaufs meiner Masterarbeit betreut und unterstützt haben.

Ein besonderer Dank gilt meinen beiden Institutsbetreuern Dr. Reinhard Keller und Dr. Heiko Hafok sowie Dr. Alex Kraus, welche mir bei Fragen stets zur Seite standen und hilfreiche Anregungen aufzeigten.

Ebenfalls möchte ich mich bei der Systemgruppe sowie allen Kollegen des Max-Planck-Instituts in Effelsberg bedanken, die mir großes Interesse und Hilfsbereitschaft entgegen brachten.

Meiner Freundin Eva Britz möchte ich ebenfalls danken, da sie mir als Lektorin beiseite stand.

Herzlich bedanke ich mich bei Prof. Dr. Uwe Gärtner, welcher stets großes Interesse an meiner Arbeit gezeigt und mir die Möglichkeit geboten hat, in diesem interessanten Themengebiet meine Masterarbeit zu verfassen.

Inhaltsverzeichnis

Erklärung	i
Abkürzungsverzeichnis	v
Tabellenverzeichnis	vi
Abbildungsverzeichnis	vii
Listingsverzeichnis	x
Formelverzeichnis	xi
1 Einleitung	1
2 Überblick	3
2.1 Entwicklungsstand des Projekts vor Beginn dieser Arbeit	3
2.2 Aufgabenstellung	5
2.3 Radioastronomischer Empfänger	6
2.4 Auflösungsvermögen eines 4-Meter-Parabolspiegels im L-Band	7
2.5 Kalibration radioastronomischer Empfangssysteme	8
2.6 Koordinatensysteme	11
2.6.1 Horizontales Koordinatensystem	13
2.6.2 Äquatoriales Koordinatensystem	14
2.6.3 Galaktisches Koordinatensystem	15
2.6.4 Präzession	16
2.7 Sanson-Flamsteed (Sinusoidal) Projektion	17
3 Entwurf und Implementierung des Kontrollsystems	18
3.1 Aufbau des Kontrollsystems	18
3.2 Entwicklung eines skriptbasierten User Interface in Python	20
3.3 Kontinuum-Backend (Pocket-Backend) und Spektrometer	22
3.4 Entwicklung eines Rohdatenformats	27
3.5 Kontrollprogramme	29
3.5.1 4mcontrol.py	31
3.5.2 scriptcontrol.py	32
3.5.3 regler.c	38

3.5.4	datawriter.py und fftsdawriter.py	39
3.6	Regeleinheiten	43
4	Messdatenaufnahme	46
4.1	Erprobung des Kontinuum-Empfangssystems	46
4.1.1	Cross-Scan-Messung über Elevationsdrehachse	46
4.1.2	Cross-Scan-Messung über Azimutdrehachse	47
4.1.3	Empirische Evaluation der Halbwertsbreite in Elevation und Azimut	49
4.2	Erprobung des Spektrometers	54
4.2.1	Spektralmessung des Supernova-Überrests Cassiopeia A	54
4.2.2	Spektrallinie des atomaren Wasserstoffs (Spinflip-Übergang)	55
5	Entwicklung eines Pointing-Modells	57
5.1	Betrachtung der mechanischen und radioastronomischen Teleskopcharakteristik	57
5.2	Pointing-Modell Elevation	60
5.3	Pointing-Modell Azimut	62
5.4	Evaluation des Pointing-Modells	64
6	Radioastronomische Messungen	66
7	Ergebnis	70
7.1	Fazit	70
7.2	Ausblick	73
	Literaturverzeichnis	74
A	Anhang	76
A.1	Funktionen des User Interfaces	76
A.2	Programm Interfaces	79
A.3	Radioastronomische Messungen	82

Abkürzungsverzeichnis

A/D	analog/digital
APEX	Atacama Pathfinder Experiment
ASCII	American Standard Code for Information Interchange
Bash	Bourne-again shell
CAD	computer-aided design
CD	Compact Disc
cts	counts
EEEI	Electrical and Electronics Engineers Institute
FFTS	Fast Fourier Transform Spektrometer
FIFO	First In – First Out
FWHM	Full Width at Half Maximum
GPS	Global Positioning System
HF	Hochfrequenz
ICRS	International Celestial Reference System
IP	Internet Protocol
IRIG	Inter Range Instrumentation Group Timecode
JD	Julianisches Datum
JSON	JavaScript Object Notation
LNA	Low Noise Amplifier
LOFAR	Low Frequency Array
LTS	Long Term Support
MJD	Modifiziertes julianisches Datum
NHN	Normalhöhennull
Numpy	Numerical Python
OMT	Orthomode Transducer
SCPI	Standard Commands for Programmable Instruments
SOFIA	Stratospheric Observatory For Infrared Astronomy
TCP	Transmission Control Protocol
TDT	Terrestrische Dynamische Zeit
UT	Universal Time
ZF	Zwischenfrequenz

Tabellenverzeichnis

1	Horizontales Koordinatensystem	13
2	Äquatoriales Koordinatensystem	14
3	Galaktisches Koordinatensystem	15
4	Stellt den Datenaufbau der übertragenen TCP/IP-Daten der Backends dar .	26
5	Parameter des gemessenen Objekts Taurus A (Krebsnebel)	67
6	Parameter der gemessenen Objekte	68
7	JSON-Keys, des User Interface	79
8	JSON-Keys, welche über den Multicast als Messparameter verteilt werden .	80
9	JSON-Keys, welche über den Multicast aus dem Programm regler.c übertragen werden	80
10	Parameter, welche von scriptcontrol.py an das Programm regler.c übertragen werden.	80
11	JSON-Keys des Rohdatenformats	81
12	Messwerte einer Sonnenmessung im Azimut	82
13	Messwerte einer Sonnenmessung in Elevation	83
14	Messwerte einer Mondmessung im Azimut	84
15	Messwerte einer Mondmessung in Elevation	85
16	Messwerte der Supernova Cassiopeia A	86
17	Messwerte Cygnus A	87

Abbildungsverzeichnis

1	CAD-Darstellung des 4-Meter-Radioteleskops	3
2	Bild des 4-Meter-Radioteleskops	3
3	Blockschaltbild: Mess- und Steueraufbau	4
4	Blockschaltbild des L-Band Empfängers des 4-Meter-Radioteleskops, dargestellte Schalterstellung 0, bei der Frontend Drive Unit	6
5	Darstellung eines Punktes in Polarkoordinaten.	11
6	Himmelskugel über einem Beobachter-Standort auf der nördlichen Erdhälfte. Großkreis durch Zenit und Südpunkt: Meridian	11
7	Lage von Ekliptik und Äquator	12
8	Darstellung eines Punktes in horizontalen Koordinaten.	13
9	Darstellung eines Punktes in äquatorialen Koordinaten	14
10	Präzessionsbewegung der Erde	16
11	Screenshot aus dem Programm Stellarium. Dargestellt ist der Kosmos des Standort Effelsberg im azimutalen Gradnetz.	17
12	Sphärisches Koordinatensystem mit Ausschnitt der Himmelskugel	17
13	Interface des Kontrollprogramms für das 4-Meter-Radioteleskop. Dargestellt sind der User Input, welche Daten an das Teleskop übermittelt werden und die Visualisierung der Mess- und der Teleskopparameter.	19
14	Skriptbasiertes User Interface. Gezeigt ist: Konfiguration der Backends, Azimut- und Elevationsparametrisierung sowie die Übertragung an das Kontrollprogramm.	21
15	Messdatenaufnahme des Empfangssystems am 100-Meter-Radioteleskop	23
16	Oberes Blockschaltbild zeigt Backend- und Empfängersynchronisation, Spektrometer als Master, Kontinuumbackend und Empfänger als Slave ausgeführt. Unteres Blockschaltbild zeigt, wie die HF-Daten über einen Powersplitter an die Backends übertragen werden.	24
17	Übersicht über das Kontrollprogramm. Gelb markierte Blöcke sind die Unterprogramme, in welchen sich der Entwurf und die Implementierung des Kontrollsystems befinden.	30
18	Ausschnitt aus Abb.17 zeigt die In- und Outputs des Programms 4mcontrol.py	31
19	Ausschnitt aus Abb.17 zeigt die In- und Outputs des Programms scriptcontrol.py	32

20	Aufbau des Pythonprogramms scriptcontrol.py; die vier fett markierten Funktionen sind Threads, die eine parallele Programmabarbeitung ermöglichen.	33
21	Beispiel-Parametereingabe mittels des User Interface für einen Cross-Scan über ein Messobjekt (Sonne)	36
22	Interface zwischen Kontrollsystem und Regeleinheiten: Azimut und Elevation	38
23	Einblick in das Programm regler.c	38
24	Blockschaltbild: Parameterübernahme aus Multicast, Backendkommunikation, Speichern der Rohdaten in MJD und Visualisierung der Messungen . .	40
25	Blick in den Programmteil datawriter.py; die sieben Funktionen sind einzelne Threads, welche kombiniert das Backend konfigurieren, Messdaten aufnehmen, diese als Rohdatenformat speichern und visualisieren.	40
26	Einblick in das Kontrollprogramm. Grün markierte Blöcke sind die Regeleinheiten. Diese stellen das Interface zwischen Software und Hardware her.	43
27	Ausschnitt des Kontrollprogramms. Dieses visualisiert die Kommunikation zwischen den Regeleinheiten und den Motoren.	44
28	Cross-Scan in Elevation über die Sonne; dargestellt ist die Empfangsleistung über den Winkelbereich der Messung, zudem die Teleskopbewegung während der Messung.	46
29	Cross-Scan in Azimut über die Sonne; dargestellt ist die Empfangsleistung über den Winkelbereich der Messung, zudem die Teleskopbewegung während der Messung.	48
30	Halbwertsbreite der Elevations- (blau) und der Azimutscans (grün) bei Messungen der Sonne. Gemessen wurde über den gesamten Sichtbereich der Quelle. Halbwertsbreite über Azimutwinkel (oberes Schaubild) und über Elevationswinkel (unteres Schaubild) aufgetragen.	49
31	Cross-Scan über die Sonne in Azimut, einmal mit Polarisation 1 (oberes Bild) und Polarisation 2 (unteres Bild); Messung in Grundstellung (blau) und mit 90° gedrehtem Empfangssystem (rot).	51
32	Cross-Scan über die Sonne in Elevation, einmal mit Polarisation 1 (oberes Bild) und Polarisation 2 (unteres Bild); Messung in Grundstellung (blau) und mit 90° gedrehtem Empfangssystem (rot).	52
33	Cross-Scan über die Sonne in Elevation (rot) und in Azimut (blau) mit rücktransformierter Sinusoidal-Projektion.	53
34	FFTS-Messung des Supernova-Überrests Cassiopeia A in der Milchstraße, Frequenzbereich von 1100 MHz bis 1500 MHz dargestellt.	55

35	Spektralmessung der Objektes Cassiopeia A innerhalb der Milchstraße (grün) und aus dieser hinaus (blau).	56
36	Abbildung links: radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne über Elevations- und Azimutwinkel. Resultierend aus verschiedenen Elevations-Cross-Scans. Abbildung rechts: mechanische Teleskopcharakteristik, relative Abweichungen der approximierten Funktion zu den Messreihen der Elevationwinkel 0°, 45°, 80° über den Azimutwinkel. .	58
37	Abbildung links: radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, verschiedener Elevations-Cross-Scans, über Elevations- und Azimutwinkel, mit der durch Winkelaufnehmer bestimmten Interpolation. Abbildung rechts: radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, verschiedener Elevations-Cross-Scans, über Elevations- und Azimutwinkel, ohne der durch Winkelaufnehmer bestimmten Interpolation.	59
38	Radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, verschiedener Elevations-Cross-Scans, über Elevations- und Azimutwinkel, ohne der durch Winkelaufnehmer bestimmten Interpolation.	60
39	Pointing-Offset verschiedener Elevations-Cross-Scans mit interpolierter Geradengleichung.	61
40	Radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, mehrere Azimut-Cross-Scans, über Elevations- und Azimutwinkel.	62
41	Horizont am Standort des 4-Meter-Radioteleskops, Elevations- über Azimutwinkel, rot markiert ist der Winkelbereich in dem die Interpolation aus dem Pointing-Offset definiert ist.	64
42	Sonnenstand über Bad Neuenahr-Ahrweiler im Laufe eines Jahres.	65
43	Taurus A (Krebsnebel)	67
44	Starke Radioquellen: Strahlungsfluss in Abhängigkeit von der Frequenz der Strahlung. Rot markiert ist der gemessene Strahlungsfluss mit dem 4-Meter-Radioteleskop.	69
45	Cross-Scan der Sonne in Azimut. Vor überqueren der Quelle wurde die Rauschdiode ein- und ausgeschaltet.	82
46	Elevations-Cross-Scan der Sonne	83
47	Azimut-Cross-Scan des Mondes	84
48	Elevations-Cross-Scan des Mondes	85
49	Elevations-Cross-Scan der Supernova Cassiopeia A	86
50	Elevations-Cross-Scan der Supernova Cassiopeia A	87

Listings

1	Beispiel zum Starten des Spektral- und Kontinuum-Backends mittels SCPI; Programm spiegelt die Parametereingabe und erzeugt ein Echo mit Zeitmarke	25
2	Backendkonfiguration über das User Interface	25
3	Beispiel zum Auslesen des Rohdatenformats, einer FFTS-Messreihe	28
4	Minimalbeispiel zum Transformieren des äquatorialen Koordinatensystems in das horizontale Koordinatensystem	34
5	Ausschnitt aus dem Programmcode Regler.c; Kommandos der Bosch Rex- roth Bibliothek	39

Formelverzeichnis

1	Auflösungsvermögen Radioteleskop	7
2	Auflösungsvermögen Radioteleskop	7
3	Auflösungsvermögen Radioteleskop	7
4	radiated transmitter power (Empfängerleistung)	8
5	Boltzmannkonstante	8
6	radiated transmitter power (Empfängerleistung)	9
7	radiated transmitter power (Empfängerleistung)	9
8	Teleskopempfindlichkeit	9
9	wirksame Antennenfläche	9
10	Teleskopempfindlichkeit	9
11	Flux density (power) (Leistungsdichte)	9
12	Teleskopempfindlichkeit	9
13	Teleskopempfindlichkeit	10
14	Flux density (power) (Leistungsdichte)	10
15	Flux density (power) $\lambda > 6$ cm (Leistungsdichte)	10
16	Auflösungsvermögen Radioteleskop	17
17	Berechnung Azimutwinkel aus User Interface	21
18	Berechnung Elevationswinkel aus User Interface	21
19	Annahme eines Elevationswinkels für die Sonne	36
20	Annahme eines Elevationswinkels für die Sonne	36
21	Elevationswinkel mit Parameteroffset	36
22	Berechnung der relativen Bewegung des Teleskops in Azimut	36
23	Berechnung der relativen Bewegung des Teleskops in Azimut	36
24	Berechnung der relativen Bewegung des Teleskops in Azimut	36
25	Berechnung der relativen Bewegung des Teleskops in Azimut	37
26	Rotationsgeschwindigkeit mit Vorhaltezeit bzw. Nachhaltezeit	37
27	Vorhalte- bzw. Nachhaltezeit	37
28	Rotationsgeschwindigkeit mit Vorhalte- bzw. Nachhaltezeit	37
29	Azimutparameter	44
30	Elevationsparameter	44
31	resultierende Funktion aus Interpolation der Messwerte aus Abbildung 38	61
32	Stärke der Rauschdiode	66
33	Empfindlichkeit des 4-Meter-Teleskops	66
34	Temperatur der Rauschdiode in K/counts	66

35	Systemtemperatur	66
36	Antennentemperatur	66
37	Flux density (power) (Leistungsdichte)	66

1 Einleitung

Neben dem 100-Meter-Radioteleskop und dem Langwellenteleskop LOFAR¹ betreibt das Max-Planck-Institut für Radioastronomie in Effelsberg einen vier Meter Parabolspiegel, der ursprünglich in der Nachrichtentechnik zur Kommunikation mit geostationären Satelliten diente. Dieser soll nach den Anforderungen eines radioastronomischen Messsystems modifiziert werden. Vorgegangene Arbeiten widmeten sich der Thematik des Umbaus eines statischen Paraboloids in ein dynamisches Antennensystem. Dieses bietet die Möglichkeit, den Spiegel in sphärischen Koordinaten, in den Winkeln Azimut und Elevation, auszurichten. Beide Achsen wurden mit Antriebseinheiten ergänzt und ermöglichen durch ein Interface eine Parametrisierung in der Hochsprache C unter dem Linux Betriebssystem Ubuntu. Als Empfangssystem dient ein Empfänger, welcher auf die 21 cm HI-Linie des atomaren Wasserstoffs optimiert ist. Die vorliegende Abhandlung knüpft an die vorgegangenen Arbeiten an und thematisiert den Entwurf und die Implementierung eines Linux-basierten Kontrollsystems für ein 4-Meter-Radioteleskop. Zunächst wird die Implementierung eines User Interfaces diskutiert und wie astronomische Positionierungen sowie Messmodi zu kommandieren sind. Eine skriptbasierte Umsetzung soll dem Benutzer die Möglichkeit bieten, unter Zuhilfenahme definierter Funktionen das Kontrollprogramm zu parametrisieren. Die Thematisierung der Positionierung des Teleskops wirft folgende Fragestellung auf: Wie ist es möglich, eine astronomische Parametervorgabe bzw. einen Messmodus in die lokalen Azimut- und Elevationskoordinaten zu transformieren, anschließend das Teleskop auszurichten und radioastronomische Messungen zum richtigen Zeitpunkt zu starten bzw. zu stoppen. Zudem muss das Teleskop mit der benötigten Genauigkeit ausgerichtet werden und aufgrund der Erdrotation mit einer konstanten Rotationsgeschwindigkeit astronomische Positionen nachführen bzw. soll das Teleskop, beim Abtasten, über das Objekt gefahren werden. Es wird die Implementierung und Datenverarbeitung zweier Backends thematisiert. Zum einen das Kontinuum-Backend, welches über eine Total-Power-Erfassung Intensitätsmessungen durch Aufintegration der Leistung im Bandpass durchführt und zwei Backendsections mit horizontaler und vertikaler Polarisation zur Verfügung stellt. Zum anderen soll für Beobachtungen im Frequenzbereich die Total-Power-Erfassung durch ein Spektrometer in das System eingebunden werden, welches wie das Kontinuum-Backend horizontale und vertikale Polarisationen verarbeitet. Aus einer radioastronomischen Beobachtung resultieren zwei unterschiedliche Datenströme: Die Teleskopparameter, wie beispielsweise Position und Rotationsgeschwindigkeit des Systems sowie die Messdaten der beiden Backends. Aus diesem Grund ist die Entwicklung eines Rohdatenformats erforderlich, welches die Datenströme zu-

¹<http://www.lofar.org/>

sammenführt, in einem definierten Format speichert und ein möglichst effektives Auslesen ermöglicht. Mit diesem kann der Beobachter die aktuellen Messungen verfolgen und evaluieren oder zu einem späteren Zeitpunkt auswerten. Infolge der Implementierung des Kontrollprogramms und Einbindung der Backends werden die ersten radioastronomischen Messungen thematisiert. Resultierend daraus können das Empfangssystem, die Positioniergenauigkeit und das exakte Starten und Stoppen der Backends überprüft werden. Daher besteht die Möglichkeit, die radioastronomische Teleskopcharakteristik zu bestimmen und diese mit der mechanischen Teleskopcharakteristik zu vergleichen. Beobachtungen im Kontinuum sollen ermitteln, welche Objekte mit der Antenne im L-Band sichtbar sind sowie mit Messungen im Frequenzbereich der Versuch unternommen werden, ob mit dem 4-Meter-Radioteleskop atomarer Wasserstoff, über den Spin-Flip-Übergang, in der Milchstraße nachweisbar ist.

2 Überblick

2.1 Entwicklungsstand des Projekts vor Beginn dieser Arbeit

Der mechanische Aufbau des Teleskops ist weitestgehend abgeschlossen. In Abbildung 1 ist das Teleskop als CAD-Modell sowie ein Bild des Teleskops auf dem Institutsgelände, siehe Abbildung 2, dargestellt. Der Azimutantrieb ist über einen Zahnkranz mit dem Teleskop verbunden. Die beiden Zahnräder sind mit einer Evolventenverzahnung ausgeführt und bieten somit ein möglichst geringes Zahnflankenspiel sowie resultierend daraus eine hohe Positioniergenauigkeit. In der Elevationsachse wird ein Aktor verwendet, welcher über einen Kugelgewindetrieb translatorische Bewegungen ausführt. Das Ein- bzw. Ausfahren der Zylinderstange bewirkt ein Neigen der Elevationsachse von 0° bis 90° . Die translatorische Bewegung des Elevationsaktors wurde in der Projektarbeit, *Einbau, Justage und Inbetriebnahme eines L-Band Empfangssystems in ein 4-Meter-Radioteleskop*,² analysiert und in eine rotatorische Bewegung transformiert.



Abbildung 1: CAD-Darstellung des 4-Meter-Radioteleskops³



Abbildung 2: Bild des 4-Meter-Radioteleskops

²vgl. Schweikert [2017]

³[Schweikert 2017, S. 3]

Außerdem wurde in der vorangegangenen Arbeit am Primärfokus des Radioteleskops ein L-Band Empfangssystem eingebunden. Der Fokus wurde bestimmt und das Teleskop in Betrieb genommen. Der Poweroutput des Empfängers ist mit seiner gesamten Leistung an einen analogen Schreiber angeschlossen. Ein Taumeln der Antenne in Abhängigkeit des Azimutdrehbereichs zur Elevationskipppachse wurde kompensiert. Über eine grafische Oberfläche können dem Teleskop Positionen in Azimut und Elevation übermittelt werden. Darüber hinaus besteht die Möglichkeit, radioastronomische Objekte zu tracken.

Die Ausgangssignale des Empfangssystems werden analog übertragen. Eine Darstellung des Mess- und Steueraufbaus zeigt Abbildung 3. Während dieser Arbeit wird das Band mit der Empfangsfrequenz von 1200 MHz bis 1500 MHz verwendet. Es geschieht keine Frequenzumsetzung, somit ist die Empfangsfrequenz gleich der ZF-Frequenz. Diese geht mit der gesamten Bandbreite, in diesem Fall 300 MHz, auf einen Leistungsteiler. Im Laufe dieser Arbeit werden an diesem ein Kontinuum-Backend sowie ein FFT-Spektrometer angeschlossen, siehe Kapitel 3.3.

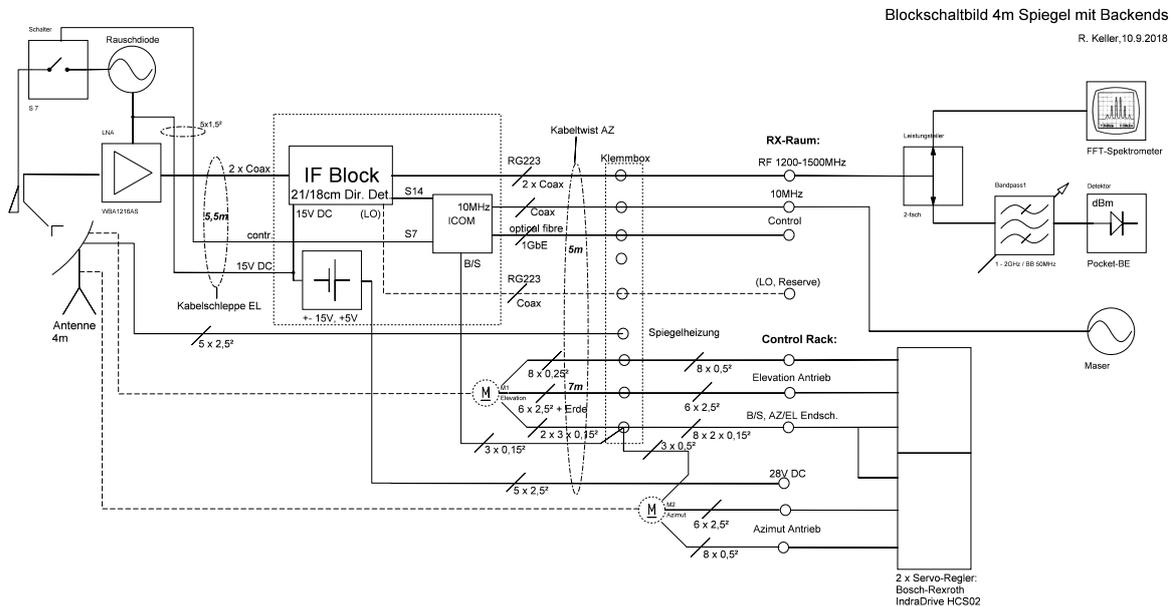


Abbildung 3: Blockschaltbild: Mess- und Steueraufbau

2.2 Aufgabenstellung

Im Rahmen dieser Arbeit wird ein Kontrollsystem für ein 4-Meter-Radioteleskop entworfen und entwickelt, welches an die vorangegangenen Arbeiten, *Automatisierung eines 4-Meter-Radioteleskops im Azimut, basierend auf dem Linux Betriebssystem Ubuntu*⁴ sowie *Einbau, Justage und Inbetriebnahme eines L-Band Empfängers in ein 4-Meter-Radioteleskop*⁵ anschließt.

Die Aufgabenstellung wurde in vier Themengebiete unterteilt, welche in Kombination radioastronomische Messungen mit dem 4-Meter-Teleskop ermöglichen.

Entwicklung eines skriptbasierten User Interface in Python:

Das User Interface soll so implementiert sein, dass der Benutzer über Kommandobefehle das Teleskop parametrisieren kann. Darunter fallen Positionsvorgaben in verschiedenen Koordinatensystemen, das Festlegen radioastronomischer Messmodi sowie die Konfiguration der Backends. Zudem soll der Beobachter die Teleskopparameter auslesen und überwachen können.

Implementierung radioastronomischer Messmodi:

Die Einbindung radioastronomischer Messmodi in das Kontrollsystem soll dem Benutzer die Möglichkeit bieten, dem Teleskop Objekte bzw. Positionen in astronomischen Koordinaten vorzugeben. Das Teleskop muss die Koordinaten in das lokale Koordinatensystem transformieren und die gewünschte Position anfahren, überfahren oder verfolgen. Zudem muss die Messung zum richtigen Zeitpunkt gestartet bzw. gestoppt werden.

Entwicklung eines Rohdatenformats:

Nach einer Messung müssen die empfangenen Daten sowie die Teleskopparameter gespeichert werden. Für diesen Zweck soll ein Rohdatenformat entwickelt werden, in welchem die Daten zu speichern sind und welches die Möglichkeit bietet, diese möglichst einfach auszulesen und verarbeiten zu können.

Messdatenvisualisierung:

Damit der Beobachter die durchgeführte Messung verifizieren kann, soll die letzte aufgenommene Messreihe, welche im Rohdatenformat gespeichert ist, visualisiert werden.

⁴Schweikert [2016]

⁵Schweikert [2017]

2.3 Radioastronomischer Empfänger

Das Empfängersystem ist eine Kopie des 21/18 cm Empfängers des 100-Meter-Radioteleskops. In Verwendung ist ein vereinfachtes Primärfokushorn. Als OMT⁶ wird ein Kreuzdipol, welcher auf 21 cm optimiert ist, verwendet. Die beiden LNAs⁷ sind ungekühlt und befinden sich hinter dem OMT im Fokus. Die weitere Signalverarbeitung ist am Standfuß des Teleskops implementiert. Das Empfangssystem wird wie bei dem 100-Meter-Radioteleskop mit einer Einheit geschaltet und überwacht, welche u.a. die Bandumschaltung in der ZF durchführt. Die ZF ist im Empfängerraum an einem Steuerrack angeschlossen. Es kann dieselbe Infrastruktur wie bei den Empfangssystemen des 100-Meter-Radioteleskops verwendet werden.⁸

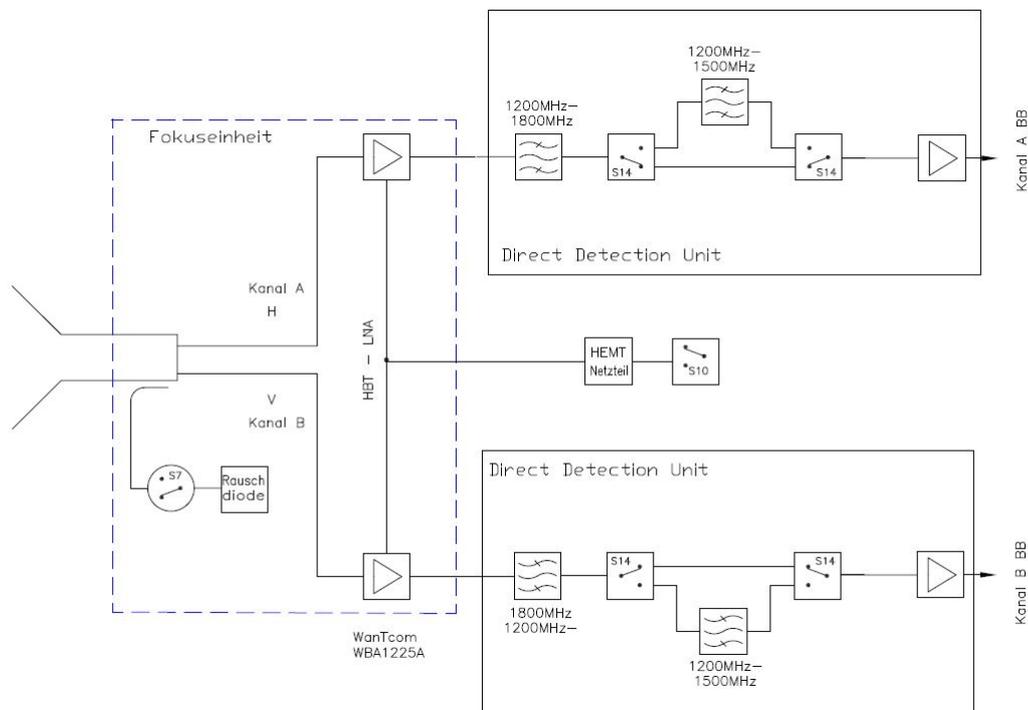


Abbildung 4: Blockschaltbild des L-Band Empfängers des 4-Meter-Radioteleskops, dargestellte Schalterstellung 0, bei der Frontend Drive Unit⁹

⁶Orthomode Transducer → trennt horizontale von der vertikalen Polarisation

⁷Low Noise Amplifier

⁸vgl. [Schweikert 2017, S.4]

⁹[Schweikert 2017, S. 4]

2.4 Auflösungsvermögen eines 4-Meter-Parabolspiegels im L-Band

Zur Charakterisierung eines Radioteleskops spielt das Auflösungsvermögen $\Delta\Phi$ eine große Rolle. Für Radioteleskope gelten die identischen physikalischen Gesetze, wenn es um das Auflösungsvermögen entfernter Objekte geht, wie bei der optischen Astronomie. Das Auflösungsvermögen eines Radioteleskops wird definiert als kleinster Winkelabstand zweier Objekte, die man gerade noch voneinander trennen kann. Für eine Parabolantenne, kreisförmige Apertur, kann das erste Beugungsminimum mit folgender Formel beschrieben werden:

$$\sin(\Delta\Phi) = 1,22 \cdot \frac{\lambda}{d} \quad (1)$$

Der Paraboloid hat einen Durchmesser von $d = 4$ m. Zudem werden Messungen im L-Band mit einer Frequenz von 1400 MHz durchgeführt. Dies entspricht einer Wellenlänge von $\lambda = 21,41$ cm.¹⁰ Mit den Randbedingungen $d \gg \lambda$ erhält man als Abschätzung (in Grad):¹¹

$$\Delta\Phi \approx 1,22 \cdot \frac{\lambda}{d} \cdot \frac{360^\circ}{2\pi} \quad (2)$$

$$\Delta\Phi \approx 1,22 \cdot \frac{0,2141 \text{ m}}{4 \text{ m}} \cdot \frac{360^\circ}{2\pi} = 3,74^\circ \quad (3)$$

Beim Messen einer punktförmigen Quelle mit einem Parabolspiegel wird die Intensitätsverteilung einer Strahlungsquelle gemessen. An die Verteilungsfunktion wird in der Radioastronomie eine Gaußsche Glockenfunktion approximiert. Die Formel 3 beschreibt die erwartete Halbwertsbreite der angenäherten Gaußkurve, resultierend aus einer radioastronomischen Beobachtung.

¹⁰ $\lambda = \frac{c}{f}$; $c = 3 \cdot 10^8$ m/s [Kraus 1988, S.11]

¹¹vgl. [Unsöld u. Baschek 2005, S.130]

2.5 Kalibration radioastronomischer Empfangssysteme

Das Empfangssystem am 4-Meter-Radioteleskop empfängt aus dem Kosmos eine Rauschleistung bzw. beim Überfahren einer Radioquelle eine Leistungserhöhung. Aufgrund des Zusammenhangs:¹²

$$P = k \cdot T_A \cdot \Delta\nu \quad (4)$$

$$k = 1,38 \cdot 10^{-23} \text{JK}^{-1} \quad (5)$$

werden die Leistungen in der Radioastronomie üblicherweise als Temperatur angegeben. T_A entspricht einer Antennenrauschtemperatur. Der Parameter $\Delta\nu$ ist die Bandbreite des Empfangssystems.

Bei dem hier verwendeten Empfänger wird, wie bei allen Empfangssystemen am 100-Meter-Radioteleskop, am Empfängereingang über einen Richtkoppler ein von einer Halbleiter-Rauschdiode erzeugtes Rauschsignal zur Kalibration der Messungen eingespeist. Es kann synchron zum Messzyklus geschaltet werden und addiert auf das empfangene Signal ein breitbandiges, in seiner Größe bekanntes, Rauschsignal. Die Größe des Kalibrationssignals wird normalerweise bei Inbetriebnahme eines Empfängers durch eine Eichmessung mit kaltem und warmem Absorber oder durch Vergleich mit einer bekannten Radioquelle bestimmt. Es wird davon ausgegangen, dass die Amplitude der Kalibration eines Empfängers bekannt und über längere Zeit konstant ist. Für den hier verwendeten Empfänger ist diese Rauschtemperatur $T_{\text{cal}} = 20\text{K}$. Die Amplitude des gemessenen radioastronomischen Signals kann nun durch Vergleich mit der Amplitude des Kalibrationssignals bestimmt werden. Dazu wird das Kalibrationssignal periodisch an- und abgeschaltet, um daraus kontinuierlich die Amplitude des gemessenen Signals in Relation zur bekannten Amplitude des Kalibrationssignals zu bestimmen. Gleichzeitig werden damit Schwankungen in der Verstärkung des Empfangssystems eliminiert, solange diese langsamer sind als der Schaltzyklus der Kalibration. Die so gewonnene Empfangsleistung kann in Form einer Antennentemperatur (in Kelvin) dargestellt werden.¹³

Die empfangene Leistung des Radioteleskops hängt zudem von der Größe des Parabolspiegels sowie der Bandbreite des Empfängers ab. Je größer das Teleskop bzw. je breiter das Frequenzband, umso stärker ist das empfangene Signal. Ein Objekt der Flussdichte S erhöht

¹²Formel und Boltzmann Konstante vgl. [Kraus 1988, S.785]

¹³Keller [2007]

die empfangene Leistung um:¹⁴

$$P = \frac{1}{2} \cdot S \cdot A_e \cdot \Delta\nu \quad (6)$$

S = power density per unit bandwidth, $\text{Wm}^{-2}\text{Hz}^{-1}$ (Leistungsdichte pro Bandbreite)

A_e = effective aperture, m^2 (Antennen Apertur)

$\Delta\nu$ = bandwidth, Hz (Bandbreite)

Faktor: $\frac{1}{2} \rightarrow$ Dipol nimmt von natürlicher, unpolarisierter Strahlung nur eine Polarisationsrichtung auf.

Aufgrund von unterschiedlichen Messinstrumenten muss die Helligkeit des zu beobachtenden Objekts für andere Astronomen vergleichbar gemacht werden. Dies geschieht mit der Umrechnung in die Flussdichte und wird bestimmt mit der Division der gemessenen Antennentemperatur T_A durch die Empfindlichkeit des Teleskops Γ . Die Empfindlichkeit des 100-Meter-Radioteleskops beträgt bei 21 cm ca. 1,5 K/Jy. Das 4-Meter-Teleskop liegt etwa bei 0,0036 K/Jy (angenommene Aperturreffizienz (Antennenwirkungsgrad) $\eta = 0,8$).¹⁵ In der Radioastronomie wird die Flussdichte in Jansky (Jy) angegeben. Die Umrechnung ist wie folgt:¹⁶

$$(4) = (6)$$

$$k \cdot T_A \cdot \Delta\nu = \frac{1}{2} \cdot S \cdot A_e \cdot \Delta\nu \quad (7)$$

$$\Gamma = \frac{T_A}{S} = \frac{A_e}{2 \cdot k} = \eta \cdot \frac{A_{\text{geom}}}{2 \cdot k} \quad (8)$$

$$A_{\text{geom}} = \pi \cdot \left(\frac{D}{2}\right)^2 \quad (9)$$

$$\Gamma = \frac{\pi}{8} \cdot \frac{\eta}{k} \cdot D^2 \quad (10)$$

$$1\text{Jy} = 10^{-26}\text{Wm}^{-2}\text{Hz}^{-1} = 10^{-26}\text{kg/s}^2 \quad (11)$$

$$\Gamma = \frac{\pi}{8} \cdot \frac{0,8}{1,38 \cdot 10^{-23}\text{JK}^{-1}} \cdot (4\text{m})^2 = 3,64 \cdot 10^{23} \text{Ks}^2/\text{kg} \quad (12)$$

¹⁴Formel 6 vgl. [Kraus 1988, S.775]

¹⁵diese Aussage bezieht sich auf einen Erfahrungsaustausch mit Dr. Kraus; Max-Planck-Institut für Radioastronomie

¹⁶vgl. [Kraus 1988, constants and conversions]

$$\Gamma = 3,64 \cdot 10^{-3} \text{ K/Jy} \quad (13)$$

Außerdem müssen weitere Effekte korrigiert werden: Die Dämpfung der Atmosphäre sowie der Verlust an Empfindlichkeit durch Verformung des Paraboloids beim Kippen der Elevationsachse. Der Verlust der Empfindlichkeit durch Verformung wird durch eine Parabel in Elevation ($G = a_0 + a_1 \cdot EL + a_2 \cdot EL^2$) dargestellt.¹⁷ Die Parameter a_i werden durch eine Messung bestimmt. Die Dämpfung in der Atmosphäre (Opazität) ist über den Term $e^{-\tau/\sin(EL)}$ definiert.¹⁸

$$S = \frac{e^{\tau/\sin(EL)}}{G(EL)} \cdot \frac{T_{\text{cal}}}{\Gamma} \cdot (\text{Empfangsleistung}) \quad (14)$$

Bei großen Wellenlängen ($\lambda > 6 \text{ cm}$) können beide Effekte vernachlässigt werden.¹⁹ Aufgrund des verwendeten L-Band Empfangssystems, mit einer Wellenlänge von $\lambda = 21,41 \text{ cm}$, werden folgende Annahmen getroffen:

$$\begin{aligned} a_0 &= 1 \\ a_1 &= a_2 = 0 \\ \tau &\ll 1 \end{aligned}$$

$$S = \frac{T_{\text{cal}}}{\Gamma} \cdot (\text{Empfangsleistung}) \quad (15)$$

Da gewöhnlich weder die Stärke der Rauschdiode, noch die Empfindlichkeit des Teleskops exakt bekannt sind, wird das Verhältnis T_{cal}/Γ durch Kontinuums-Messungen auf Eichquellen, deren Flussdichte bekannt sind, überprüft.

¹⁷im Folgenden werden die beiden Positionswinkel in Formeln als AZ (Azimutwinkel) und EL (Elevationswinkel) abgekürzt.

¹⁸Strahlungstransport: $\tau = \int_0^L \kappa(l') dl'$ mit Absorptionskoeffizient κ einer Materieschicht der Dicke L [Kraus 1997, S. 19]

¹⁹vgl. Kraus [2018]

2.6 Koordinatensysteme

Zum Beschreiben eines Körpers im Raum wird ein Koordinatensystem benötigt, welches durch einen Nullpunkt, eine Bezugsrichtung und eine Bezugsebene definiert ist. Die Radioastronomie verwendet sphärische Koordinaten (Polarkoordinaten). Diese werden durch die zwei Winkel β und λ sowie die Entfernung des Punktes r dargestellt, vgl. als Beispiel Abbildung 5. Die Entfernung spielt zum Ausrichten des Teleskops keine Rolle. In der Radioastronomie sind verschiedene Koordinatensysteme definiert, welche wie folgt aufgebaut sind: Es wird eine Bezugsebene gewählt, die die Himmelskugel in einem Großkreis, siehe Abbildung 6, schneidet. Auf dieser wird ein Nullpunkt und eine Bezugsrichtung festgelegt, von welchem aus der Winkel λ (vgl. Abbildung 5) definiert ist. Die senkrecht auf dem Grundkreis (beispielsweise geozentrischer Horizont, Abbildung 6) stehenden Großkreise schneiden die Pole des Systems. Auf dem Großkreis, der durch die Pole und durch den betrachteten Stern geht, wird der Winkelabstand vom Grundkreis (β vgl. Abbildung 5) festgelegt.

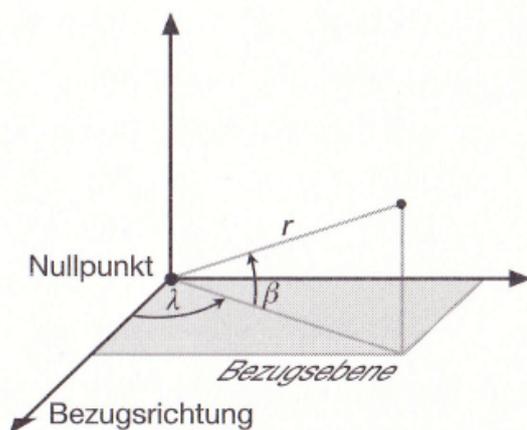


Abbildung 5: Darstellung eines Punktes in Polarkoordinaten.²⁰

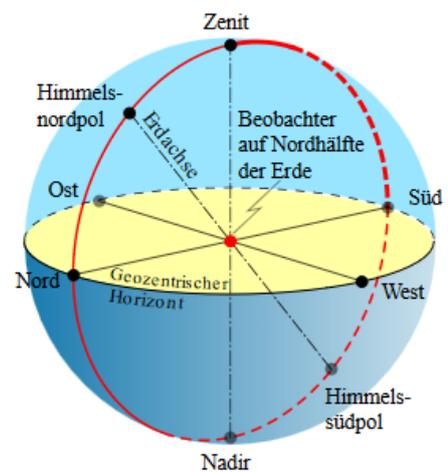


Abbildung 6: Himmelskugel über einem Beobachter-Standort auf der nördlichen Erdhälfte. Großkreis durch Zenit und Südpunkt: Meridian²¹

²¹[Montenbruck 2001, S.1]

²¹Wikipedia [2016]

In Abbildung 7 ist die Ekliptik dargestellt. In dieser Ebene, welche den Mittelpunkt der Sonne schneidet, bewegt sich die Erde um die Sonne. Legt man eine Ebene durch den Erdmittelpunkt, welche senkrecht auf der Erdachse steht, schneidet diese die Erdoberfläche im Erdäquator. Der Frühlingspunkt definiert den Stand der Sonne zum Frühlingsanfang der Nordhalbkugel (Herbstanfang der Südhalbkugel). Der zweite Schnittpunkt des Himmelsäquators mit der Ekliptik wird Herbstpunkt genannt. Der Äquator und die Ekliptik stehen über eine längere Zeitdauer nicht fest im Raum. Diese Bewegung wird Präzession genannt, welche das Kapitel 2.6.4 thematisiert.²²

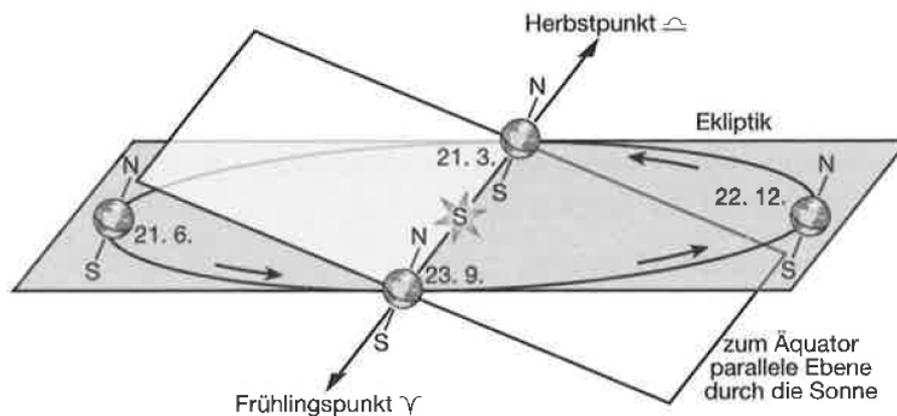


Abbildung 7: Lage von Ekliptik und Äquator²³

Im Folgenden wird auf die verschiedenen Koordinatensysteme, mit welchen sich im Laufe dieser Arbeit beschäftigt wurde, eingegangen.

²²vgl.[Montenbruck 2001, S.3f.]

²³[Montenbruck 2001, S.4]

2.6.1 Horizontales Koordinatensystem

Das horizontale Koordinatensystem bezieht sich auf den Standort des Beobachters. Dieses teilt sich in zwei Winkel auf: den Azimutwinkel, der sich nach den Himmelsrichtungen ausrichtet und den Elevationswinkel, welcher den Höhenwinkel angibt. Diese Koordinaten dienen als lokale Parameter, welche dem Teleskop zur Positionierung vorgegeben werden. Daher müssen alle folgenden Koordinatensysteme in dieses transformiert werden.

Ursprung:	Beobachter auf der Erdoberfläche am Ort der geographischen Breite
Bezugsebene:	Horizont
Bezugsrichtung:	Norden
Koordinaten:	<p>EL: Elevation: Höhe über dem Horizont, gemessen von -90° (Nadir) bis $+90^\circ$ (Zenit)</p> <p>AZ: Azimut: Winkel zwischen dem Großkreis durch Zenit und Himmelspol und dem Großkreis durch den Zenit und des beobachteten Objekts. Es wird von Norden im Uhrzeigersinn von 0° bis 360° gemessen.</p> <p><i>Vorsicht:</i> Es gibt auch Zählungen, die im Süden beginnen. In dieser Arbeit beginnt die Zählung im Norden!</p>

Tabelle 1: Horizontales Koordinatensystem²⁴

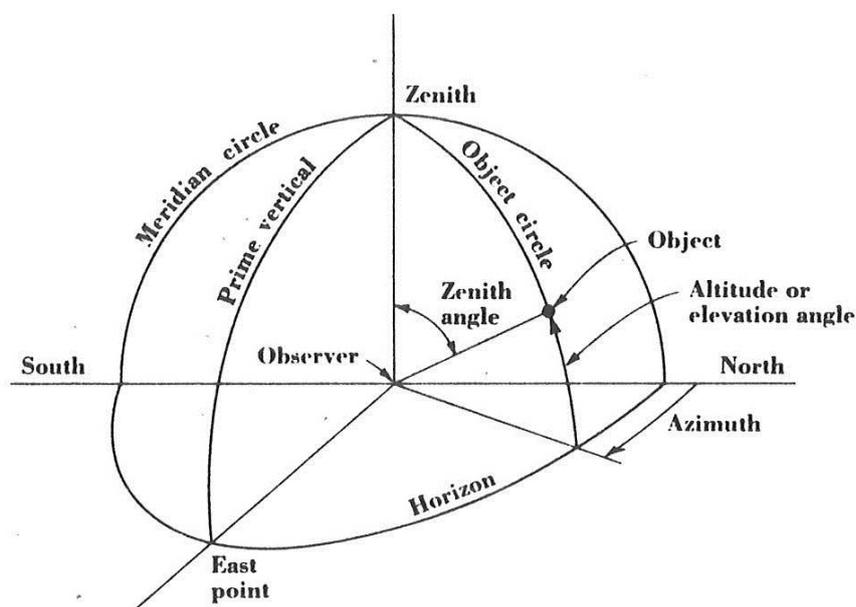


Abbildung 8: Darstellung eines Punktes in horizontalen Koordinaten.²⁵

²⁴vgl. [Kraus 1986, Kapitel 2-8]

²⁵[Kraus 1986, Kapitel 2-22]

2.6.2 Äquatoriales Koordinatensystem

Dieses Koordinatensystem benutzt den Erdmittelpunkt als Ursprung, mit Bezug auf den Himmelsäquator. Daraus resultieren die allgemein üblichen Koordinaten Rektaszension und Deklination.

Ursprung:	Erdmittelpunkt
Bezugsebene:	Erdäquator
Bezugsrichtung:	Frühlingspunkt eines bestimmten Äquinoktiums (in dieser Arbeit J2000.0 bzw. B1950.0) ²⁶
Pole:	Die positive z-Achse weist in Richtung des Nordpols der Erde. Punkte mit $\delta > 0$ ($z > 0$) liegen nördlich des Äquators Δ : Entfernung von der Erde δ : Deklination; Winkel zwischen der Linie Erde-Planet und dem Himmelsäquator, gemessen von Süden (-90°) nach Norden ($+90^\circ$).
Koordinaten:	α : Rektaszension; Winkel zwischen dem Frühlingspunkt und der Projektion der Linie Erde-Planet auf den Äquator, gemessen von 0^h bis 24^h in Richtung der Bewegung der Erde um die Sonne.

Tabelle 2: Äquatoriales Koordinatensystem²⁷

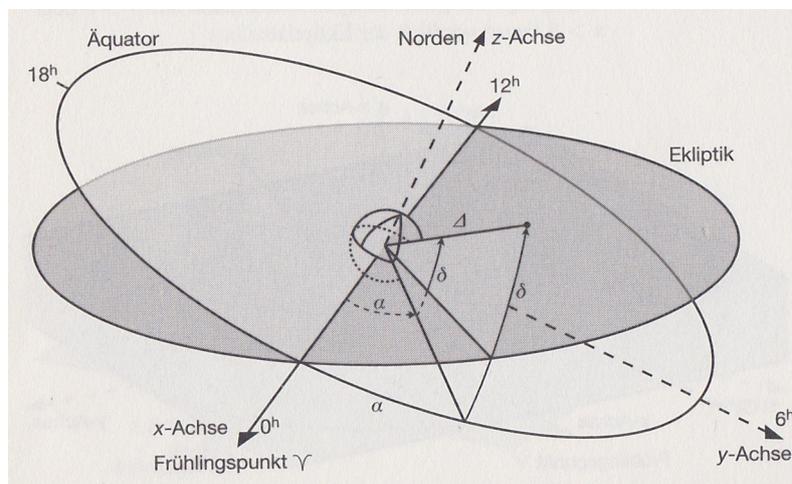


Abbildung 9: Darstellung eines Punktes in äquatorialen Koordinaten²⁸

²⁷siehe Kapitel 2.6.4

²⁷vgl. [Kraus 1986, Kapitel 2-9]

²⁸[Montenbruck 2001, S.8]

2.6.3 Galaktisches Koordinatensystem

Ursprung:	Zentrum der Milchstraße
Bezugsebene:	galaktische Ebene (Ebene der Milchstraße)
Bezugsrichtung:	Frühlingspunkt eines bestimmten Äquinoktiums
Pole:	Die Richtung der positiven z-Achse wird als galaktischer Nordpol bezeichnet. Alle Punkte $b > 0$ liegen nördlich der Milchstraßenebene, alle mit $b < 0$ südlich.
Koordinaten:	b: galaktische Breite; Winkel zwischen der der galaktischen Ebene und dem Objekt
	l: galaktische Länge; Bezug zur Richtung Sonne → galaktisches Zentrum

Tabelle 3: Galaktisches Koordinatensystem²⁹

²⁹vgl. [Kraus 1986, Kapitel 2-11]

2.6.4 Präzession

Als Präzession wird diejenige Richtungsänderung bezeichnet, welche die Rotationsachse eines rotierenden Körpers, in diesem Fall die Erde, ausführt. Eine Periode dauert etwa 25700 Jahre, in der die Achse der Erdbahn einen Kegel mit dem Öffnungswinkel von $23^{\circ}27'$ beschreibt. Vor dem Hintergrund, dass die Präzession die Lage des Koordinatensystems relativ zu den Sternen verschiebt, muss zu den Sternpositionen angegeben werden, auf welches Äquinoktium (Lage des Frühlingspunktes) sich bezogen wird. In dieser Arbeit werden die Fundamentalsysteme julianisches Jahrhundert, J2000.0 (1. Januar 2000, 12:00 TDT) und die Besselsche Epoche B1950.0 (31. Dezember 1949 22:09 UT) verwendet.³⁰

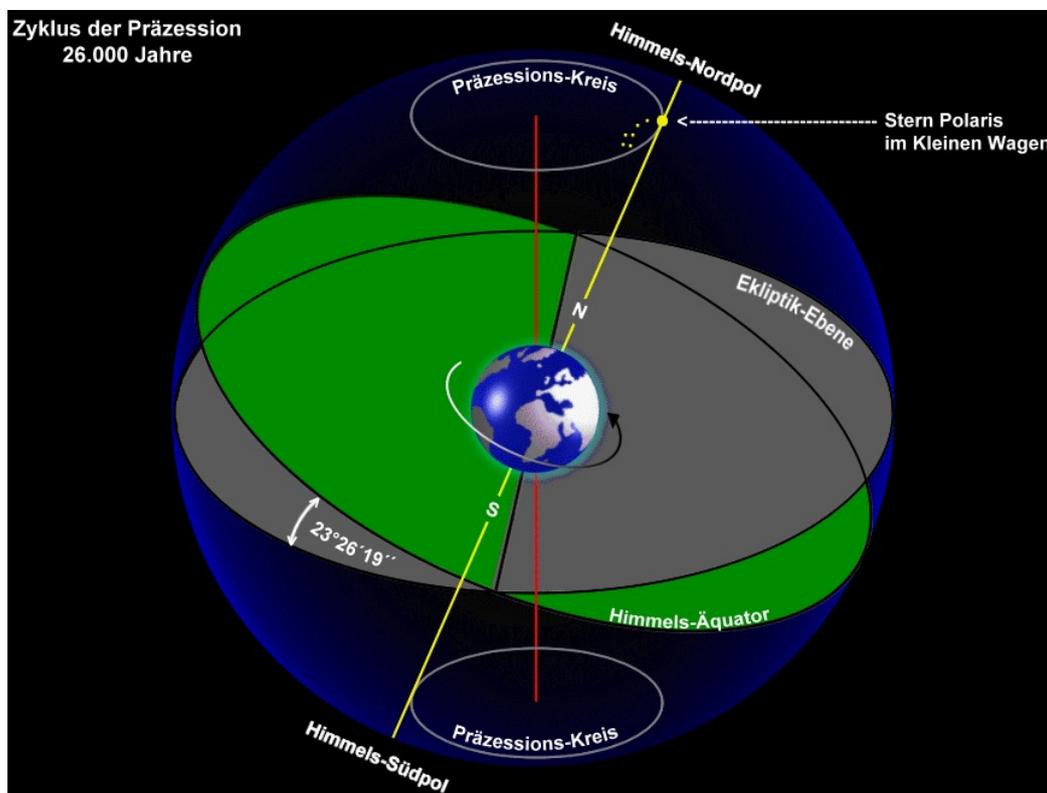


Abbildung 10: Präzessionsbewegung der Erde³¹

³⁰vgl. [Unsöld u. Baschek 2005, S.14]

³¹Sternwarte-Singen [2018]

2.7 Sanson-Flamsteed (Sinusoidal) Projektion

Die Sinusoidal- bzw. Sanson-Flamsteed Projektion ist ein flächentreuer Kartennetzentwurf. In Abbildung 11 ist der sichtbare Kosmos am Standort in Effelsberg dargestellt. Über den Sichtbereich ist das azimutale Gradnetz gelegt. Dieses stellt die sphärischen Koordinaten dar, in welchen sich das Teleskop bewegt. Werden dem Radioteleskop absolute Winkel vorgegeben, ist das Koordinatensystem definiert. Wird jedoch eine relative Bewegung über den Azimutwinkel durchgeführt, ist dieser elevationsabhängig. Befindet sich beispielsweise das Objekt im Zenit, ist die scheinbare Größe dieses über die gesamten 360° im Azimut sichtbar. Je geringer der Elevationswinkel ausfällt, umso geringer die Abhängigkeit der Azimutachse, welches die folgende Funktion beschreibt:

$$\Delta AZ_{\text{projektion}} = \frac{\Delta AZ}{\cos(EL)} \quad (16)$$

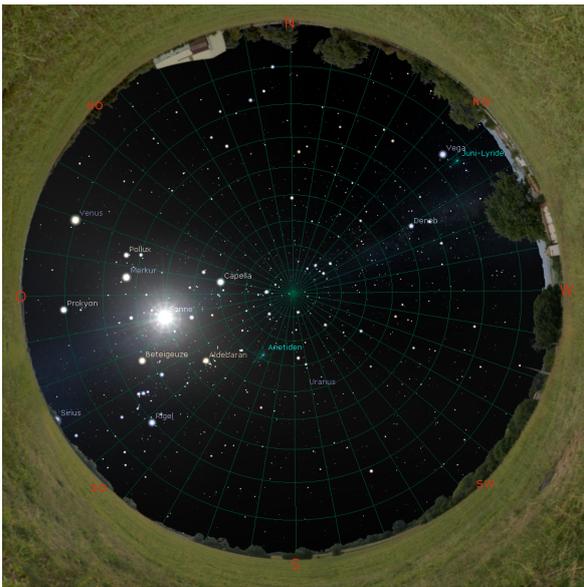


Abbildung 11: Screenshot aus dem Programm Stellarium. Dargestellt ist der Kosmos des Standort Effelsberg im azimutalen Gradnetz³²

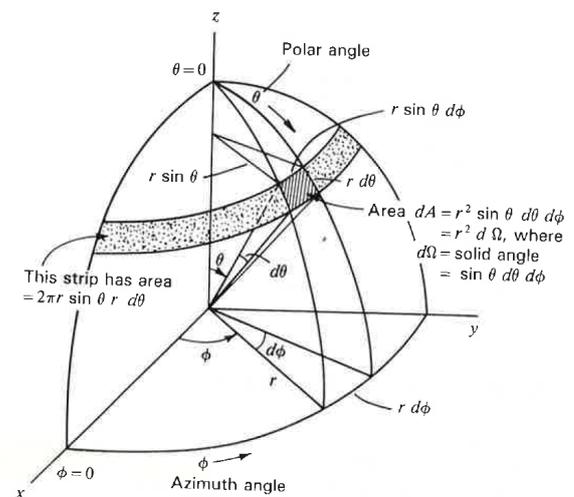


Abbildung 12: Sphärisches Koordinatensystem mit Ausschnitt der Himmelskugel³³

³²Chéreau [2017]

³³[Kraus 1988, S.24]

3 Entwurf und Implementierung des Kontrollsystems

3.1 Aufbau des Kontrollsystems

Abbildung 13 visualisiert in einem Blockschaltbild das Steuersystem des 4-Meter-Radioteleskops, welches unter dem Linux Betriebssystem Ubuntu 16.04 LTS läuft. Es wurde ein Benutzerinterface entworfen, welches dem Beobachter die Möglichkeit bietet, dem Programm Parameter bzw. Messaufträge zu übermitteln. Hierzu werden die implementierten Backends, Kontinuumbackend und Spektrometer,³⁴ je nach Auftrag konfiguriert und die Messung zum gewünschten Zeitpunkt gestartet bzw. gestoppt. Im Anschluss daran erfolgt eine Speicherung der Daten in einem Rohdatenformat sowie eine Darstellung dieser in einem Diagramm. Mithilfe des Intercoms besteht die Möglichkeit, die Rauschdiode zu schalten oder Bandumschaltungen zu kommandieren.³⁵ Das Intercom (ICOM) ist ein TCP/IP-basiertes System zur Empfänger- und ZF-Hardwaresteuerung. Die vorgegebenen astronomischen Koordinaten werden in das lokale Azimutalkoordinatensystem transformiert. Dies geschieht vor dem Hintergrund, dass die beiden standort- und zeitabhängigen Winkel Azimut und Elevation die Bezugspunkte darstellen, die die koordinatenbezogene Ausrichtung des Radioteleskops ermöglichen. Nach der Positionierung beginnt der Start der astronomischen Messung, welche nach einer vorgegebenen Zeit endet. Das Kontrollprogramm richtet das System aus und ermöglicht mittels einer Regelung die Positionsgenauigkeit. Das Steuerungsprogramm übermittelt den Hardwarekomponenten³⁶ unter anderem Azimut- und Elevationskoordinaten, die über eine bidirektionale Verbindung mit den Motoren des Teleskops kommunizieren. Der Bewegungssinn der Azimutalachse ist rotatorisch. Aus diesem Grund ist eine Übertragung der gewünschten Positionen im Winkelmaß möglich. Die Elevationsachse benötigt eine translatorische Parametervorgabe, welche aus den vorgegebenen Winkelkoordinaten errechnet werden muss. Es folgt eine Ausrichtung des Teleskops, die Encoder des Systems nehmen die Bewegung auf und übermitteln diese zurück an das Steuerungsprogramm. Diese Parameter werden in einer grafischen Oberfläche visualisiert.

³⁴mit den beiden implementierten Backends (Kontinuum- und Spektralbackend) wird sich Kapitel 3.3 vertiefend befassen.

³⁵vgl. Kapitel 2.3 Blockschaltbild 4

³⁶vgl. Kapitel 3.6

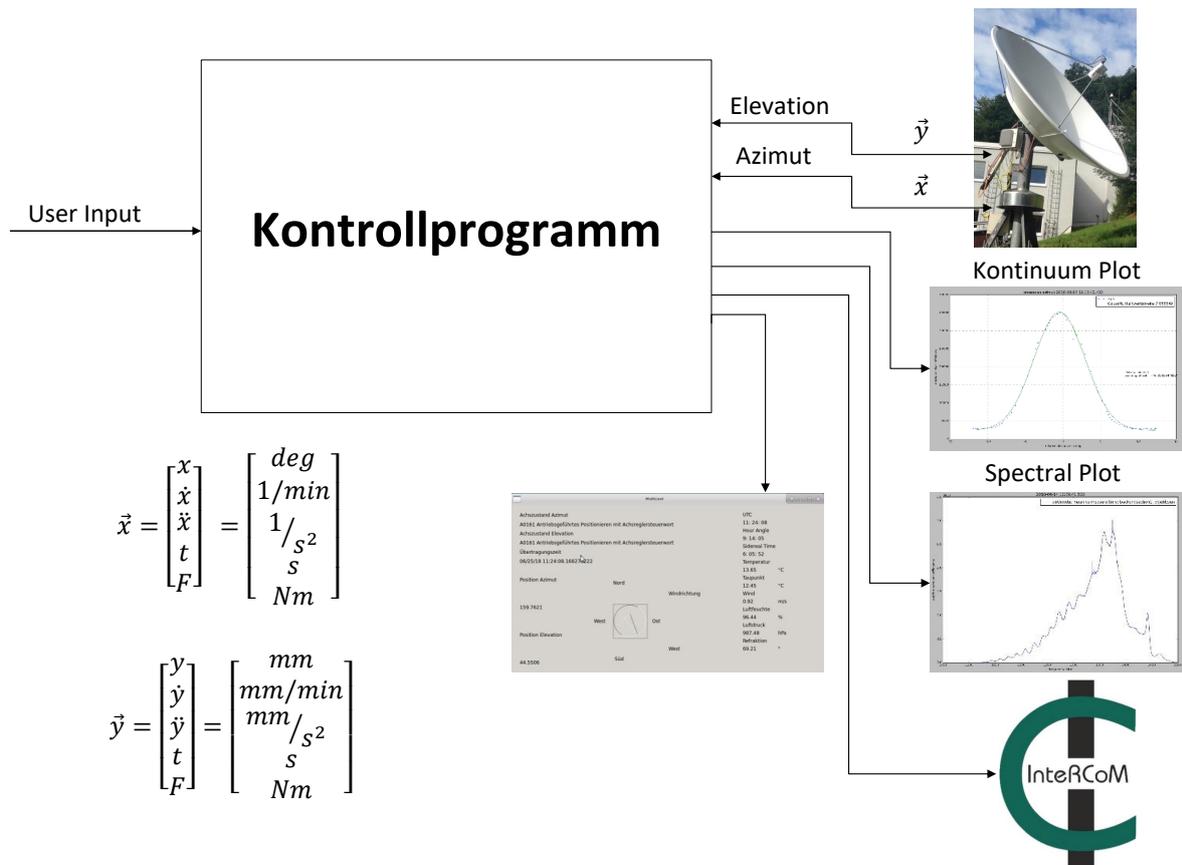


Abbildung 13: Interface des Kontrollprogramms für das 4-Meter-Radioteleskop. Dargestellt sind der User Input, welche Daten an das Teleskop übermittelt werden und die Visualisierung der Mess- und der Teleskopparameter.

3.2 Entwicklung eines skriptbasierten User Interface in Python

Das Interface der Teleskopparametrisierung ist in der Hochsprache Python skriptbasiert implementiert. Die Ausführung erfolgt in IPython, welches ein Kommandozeileninterpreter zur interaktiven Arbeit mit Python ist und speziell zum Entwickeln und Ausführen von Python-Programmen entworfen wurde. Das Benutzer-Interface ist wie in Abbildung 14 kommandobasiert entwickelt.³⁷ Die Ausführung in IPython bietet die Möglichkeit, vordefinierte Funktionen zu verwenden und in kleine Programme einzubinden. Beispielsweise kann die Parametrisierung eines Messauftrags in eine While-Schleife eingebunden werden und mittels Zeitangaben ein Objekt über einen längeren Zeitraum beobachten bzw. astronomische Messaufträge mehrmals wiederholen.

Das Programm ist in die vier Hauptfunktionen *send()*, *set*, *get* und *help()* aufgeteilt. *Set*-Funktionen setzen Teleskopparameter, wie beispielsweise in Abbildung 14 veranschaulicht, Rotationsgeschwindigkeiten oder Positionen. Komplementär dazu bieten die *get*-Funktionen die Möglichkeit, den Zustand des Radioteleskops zu überwachen. Zum Beispiel wird mit der Funktion *get_actual_torque()* das anliegende Drehmoment der Elevations- und Azimutachse ausgegeben. Die *help()*-Funktion stellt dem Benutzer ein Manual zur Verfügung, in welchem alle parametrisierbaren Teleskopfunktionen aufgelistet sind. Das Kommando *send()* stellt die Schnittstelle zwischen dem Steuerungsprogramm und dem User Interface her. Zunächst parametrisiert der Beobachter das Teleskop, mit Ausführen des Befehls *send()* werden die Konfigurationen in das Datenformat JSON exportiert und an das Steuerungsprogramm übertragen. Diese Serialisierung in lesbarer Textform dient dem Zweck des Datenaustauschs zwischen Anwendungen. Das Format besteht aus einem Bezeichner und zugehörigen Parametern, welche auch in einem Array ausgeführt sein können. Mithilfe des Bezeichners (Key) kann der Empfänger gezielt auf die benötigten Daten zugreifen.³⁸

Zudem bietet das User Interface die Möglichkeit der Backendparametrisierung. Da im folgenden Kapitel 3.3 die Backends thematisiert werden, wird hier lediglich die Möglichkeit einer Backendkonfiguration aufgezeigt. Eine Beispielkonfiguration ist in Abbildung 14 dargestellt. Die Funktion *fitwriter()* konfiguriert das Spektral- sowie die Funktion *pbackend()* das Kontinuumbackend.

³⁷Im Anhang A.1 sind alle parametrierbaren Funktion mit jeweiliger Definition hinterlegt.

³⁸Die JSON-Keys aller Schnittstellen sind im Anhang A.2 mit den jeweiligen Definitionen hinterlegt.

```

In [1]: fitwriter('BAND1:CMDNUMSPECCHAN 8192', 'BAND2:CMDNUMSPECCHAN 8192', 'CMDUSEDSECTIONS 1 1', 'CMDMODE INT', 'CMDSYNCTIME 200000', 'CMDBLANKTIME 2', 'CMDNUMPHASES 1', 'CONFIGURE')

In [2]: pbebackend('cmdusedchannels 1 1', 'cmdnumphases 1', 'cmdmode external', 'configure')

In [3]: set_position_azel(10, el = (45, 'd'), az = (160, 'd'), offsetx=(-20, 'min'), offsety = (400, 'sec'))
Azimut:
160 d -20 min
Elevation:
45 d 400 sec
integrationtime: 10

In [4]: set_status('true')

In [5]: send()

JSON-String
=====
{"decl": "", "pbebackend": "('cmdusedchannels 1 1', 'cmdnumphases 1', 'cmdmode external', 'configure')", "VLAM": "0", "offx": "-20", "SOLL-posel": "45", "unitoffx": "min", "loc_LAT": "50.5251", "SOLL-posaz": "160", "BET": "0", "VBET": "0", "unitcrosslengthx": "d", "unitcrosslengthy": "d", "loc_height": "344", "unitaz": "d", "unitel": "d", "system": "az/el", "loc_LONG": "6.88195", "source": "", "ra": "", "coordsys": "J2000.0", "unitoffy": "sec", "integrationtime": "10", "crosslengthel": "0", "fitswriter": "('BAND1:CMDNUMSPECCHAN 8192', 'BAND2:CMDNUMSPECCHAN 8192', 'CMDUSEDSECTIONS 1 1', 'CMDMODE INT', 'CMDSYNCTIME 200000', 'CMDBLANKTIME 2', 'CMDNUMPHASES 1', 'CONFIGURE')", "unitra": "h", "galach": "", "crosslengthaz": "0", "galacl": "", "velposaz": "Schleichgang", "velposel": "Schleichgang", "LAM": "0", "command": "true", "unitdec": "d", "offy": "400"}

In [6]: get_position()
('Position Azimut in deg:', '159.518884182')
('Position Elevation in deg:', '45.10237245')

```

Abbildung 14: Skriptbasiertes User Interface. Gezeigt ist: Konfiguration der Backends, Azimut- und Elevationsparametrisierung sowie die Übertragung an das Kontrollprogramm.

Abbildung 14 veranschaulicht in einem Minimalbeispiel das Benutzer-Interface und präsentiert einen Einblick in dessen Funktionsweise. Die dargestellte Funktion *set_position_azel()* setzt die Polarkoordinaten Azimut 160° und Elevation 45° im horizontalen Koordinatensystem. Additiv werden die Offsetparameter in der X-Achse, Azimutdrehachse, und in der Y-Achse, Elevationsdrehachse, vorgegeben. Um aussagekräftige Parametrisierungen zu gewährleisten, sind die Positionsparameter als Tupel mit der Zahl und der zugehörigen Einheit definiert:

$$AZ = 160^\circ - 20' = 159,67^\circ \quad (17)$$

$$EL = 45^\circ + 400'' = 45,11^\circ \quad (18)$$

Aus der Benutzereingabe ergibt sich der zu positionierende Azimutwinkel von $159,67^\circ$ sowie der Elevationswinkel von $45,11^\circ$.

Die Zahl 10 gibt die Dauer der Messung in Sekunden an. Ist das Teleskop ausgerichtet, wird die Messung begonnen und nach 10 Sekunden beendet. Die aktuelle Position des Teleskops wird mit der Funktion *get_position()* ausgelesen und dargestellt.

3.3 Kontinuum-Backend (Pocket-Backend) und Spektrometer

Das Radioteleskop ist mit zwei Messwerterfassungssystemen, Kontinuum-Backend und Spektrometer, ausgeführt, welche dem Zweck der Messdatenverarbeitung und Digitalisierung dienen.³⁹ Beide Systeme entstammen einer Entwicklung des Max-Planck-Instituts für Radioastronomie. Baugleiche Hardware findet beispielsweise am SOFIA, APEX sowie 100-Meter-Radioteleskop Einsatz. Das Kontinuum-Backend wird Pocket-Backend genannt, welches über einen Total-Power Detektor Empfangsleistung, durch Aufintegration der Leistung über den Bandpass, aufzeichnet und diese A/D wandelt. Das Spektrometer wandelt die Messwerte zunächst A/D und transformiert diese über eine FFT in den Frequenzbereich. Darauf folgt eine Integration der Frequenzspektren, um die digitale Bandbreite zu reduzieren.

Zur Erläuterung der Messdatenaufnahme wird Bezug auf das 100-Meter-Radioteleskop genommen, siehe Abbildung 15. Die Beschreibung der Signalanteile erfolgt durch die Parameter Sync, Blank, Integrationszeit und die Anzahl der Phasen. Bei Messungen mit dem 100-Meter-Radioteleskop werden vier Phasen verarbeitet. Jedes Schalten des Blanksignals bewirkt ein Inkrementieren der Phase bis zur positiven Taktflanke des Syncsignals, welches einen Zyklus definiert. Die positive Taktflanke des Blanksignals signalisiert das Starten der Messdatenaufnahme der nächsten Phase und das Beenden der vorherigen. Damit ein eindeutiges Schalten der elektronischen Bauteile gewährleistet ist, werden die während des positiven Blanksignals aufgenommenen Signalanteile abgeschnitten. Es folgt eine Integration der Messwerte und die Ausgabe dieser in einem Dump.⁴⁰ Die verschiedenen Phasen dienen dazu, abwechselnd mit und ohne zugeschalteter Rauschdiode zu messen. Diese bewirkt eine Verstärkung der Empfangsleistung mit bekannter Stärke und dient der Kalibration des Signals. Das Zu- und Abschalten der Rauschdiode hat den Vorteil, dass die Empfangsleistung direkt in das Verhältnis zu der bekannten Größe der Rauschdiode gesetzt werden kann und infolgedessen die Möglichkeit der Signalkalibration besteht.

³⁹Das Manual der beiden Backends ist auf der beigefügten CD hinterlegt.

⁴⁰Abtastwerte für die jeweilige Polarisation, Kontinuum-Backend gibt einen Messwert pro Integrationszeit aus; das Spektrometer 8192 Spektralkanäle pro Integrationszeit.

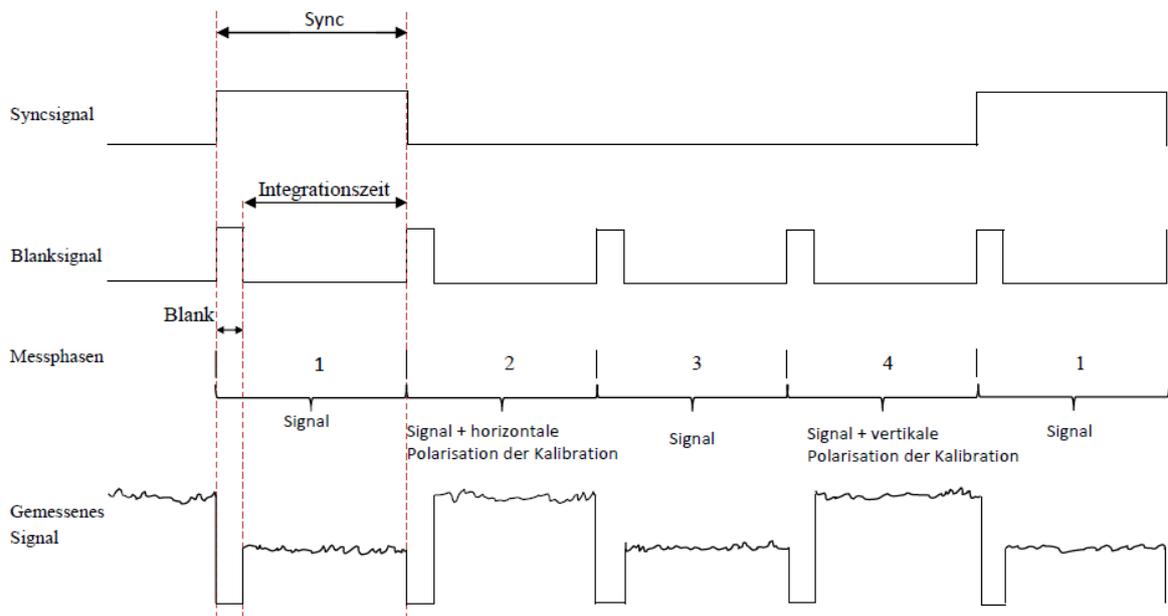


Abbildung 15: Messdatenaufnahme des Empfangssystems am 100-Meter-Radioteleskop

Im Rahmen dieser Arbeit wird nur eine Phase verwendet, da aufgrund verschiedener Phasen eine Datenserialisierung entwickelt werden müsste. Die Verarbeitung der Messdaten dieser Phase erfolgt als reine Empfangsleistung ohne aufgeschaltetes Kalibrationssignal. Das spätere Kalibrieren der Messung ist dem Benutzer durch das User Interface gegeben. Dazu kann die Rauschdiode manuell oder durch ein kleines Programm während jeder Messung einmalig geschaltet werden.

Zum Durchführen radioastronomischer Messungen benötigt das Spektrometer, das Kontinuum-Backend und der Empfänger jeweils Blank-/Syncsignale. Da die Bauteile zur selben Zeit die Messung beginnen und beenden sollen, müssen diese synchronisiert werden. Die Umsetzung dieses zeitlichen Abgleichs ist in Abbildung 16 dargestellt. Das Spektrometer fungiert als Master, welchen der Benutzer konfiguriert. Empfangseinheit und Kontinuum-Backend sind als Slaves eingebunden und erhalten die Einstellungen des FFTS. Aufgrund dieser Konfiguration dient das Spektrometer als Blank/Sync-Generator. Dies hat den Vorteil, dass dieser die Takte vorgibt und somit Empfänger, Kontinuum-Backend und Spektrometer miteinander synchronisiert. Infolgedessen kann auf die programmtechnische Synchronisation der Bauteile verzichtet werden.

Im unteren Teil der Abbildung 16 ist der Datenfluss der Empfangseinheit an die Backends dargestellt. Das HF-Signal wird über einen Powersplitter, welcher in dem Schaubild als Kreis

dargestellt ist, jeweils an das Spektrometer und an das Kontium-Backend übermittelt. Im Zeitbereich richtet zunächst die Detektoreinheit H200 das Signal proportional gleich, bevor das Kontinuum-Backend dieses A/D wandelt.

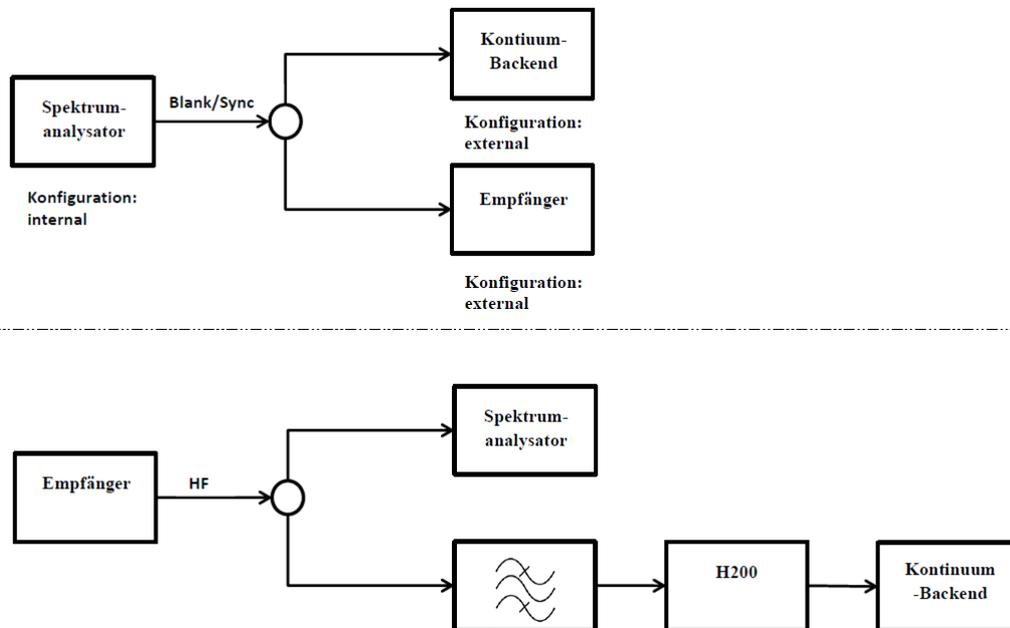


Abbildung 16: Oberes Blockschaltbild zeigt Backend- und Empfängersynchronisation, Spektrometer als Master, Kontinuumbackend und Empfänger als Slave ausgeführt. Unteres Blockschaltbild zeigt, wie die HF-Daten über einen Powersplitter an die Backends übertragen werden.

Beide Backends sind über UDP im Institutsnetzwerk angebunden und in der SCPI-Befehlssprache konfigurierbar. Aufgrund des verbindungslosen Netzwerkprotokolls werden die Parametrisierungen infolge einer erfolgreichen Datenübertragung gespiegelt und als Echo in einer Art Shake-Hands zurückgeführt.

Zur Veranschaulichung der Backendkonfiguration, durch die SCPI-Befehlssprache, ist in Listing 1 ein Minimalbeispiel⁴¹ dargestellt. Der Befehlsteil besteht aus den drei Teilen Ort, Name des Zielgeräts und der jeweiligen Anweisung, welche durch das Trennzeichen : voneinander separiert werden. Über die Angaben Ort und Namen der Zielgeräte besteht die Möglichkeit, dass auch bei einem Broadcast jedes System validieren kann, ob der Auftrag

⁴¹Der Übersichtlichkeit halber wird im Laufe dieser Arbeit lediglich auf die prägnanten Abschnitte des Kontrollprogramms in Form von Minimalbeispielen eingegangen. Der vollständige Quellcode ist auf der beigefügten CD hinterlegt.

für dieses Gerät bestimmt ist. Eine Adressierung der vorhandenen Backends geschieht über die jeweiligen Kommandos *pbe_a* (Kontinuum-Backend) bzw. *4mffts_a* (Spektrometer). Der folgende Bezeichner kommandiert in diesem Fall das Starten der Messdatenaufnahme. Aus einer erfolgreichen Parametrisierung resultiert ein Echo des Befehls, mit angehangener Zeitmarke.

```

1 apex:pbe_a:start #start Kontinuum-Backend
2 effbg:4mffts:start #start FFTS
3 #ECHO
4 APEX:PBE_A:START 2018-07-03T14:29:53.564+0000 # Rueckgabewert mit Timestamp
5 EFFBG:4MFFTS:START 2018-07-03T14:29:53.564+0000 # Rueckgabewert mit Timestamp

```

Listing 1: Beispiel zum Starten des Spektral- und Kontinuum-Backends mittels SCPI; Programm spiegelt die Parametereingabe und erzeugt ein Echo mit Zeitmarke

Zur Backendparametrisierung sind die zwei Funktionen *pbebackend()* und *fitwriter()* in das User Interface implementiert, siehe Listing 2. Diese bieten dem Benutzer die Möglichkeit, mehrere Kommandos gleichzeitig zu übertragen, welche das Programm in das SCPI-Format transformiert und sukzessiv übermittelt. In Bezug auf die Empfangssystemsynchroisation werden die Backends mit dem Kommando *cmdmode* konfiguriert. Wie bereits thematisiert, fungiert das Spektrometer als Taktgeber (*cmdmode internal*), welchem das Pocket-Backend (*cmdmode external*) folgt. Aus diesem Grund werden lediglich dem FFTS Blank- (*cmdsynctime 200000 μ s*) und Synczeiten (*cmdsynctime 2000 μ s*) kommandiert. Die Anzahl der Phasen (*cmdnumphases 1*) wird auf eins gesetzt. *Cmdusedchannels* bzw. *cmdnumspecchan* definieren die Kanäle, welche in diesem Fall die horizontale und vertikale Polarisation darstellen. Bauartbedingt sind dies bei Kontinuum-Messungen ein Messwert pro Integrationstakt und im Frequenzbereich 8192 Spektralkanäle pro Integrationstakt. Das Ausführen der Anweisung *configure* führt die Backendparametrisierung mit den vorangegangenen Konfigurationen durch.

```

1 #Konfiguration des Kontinuumbackends
2 pbebackend('cmdusedchannels 1 1','cmdnumphases 1','cmdmode external','configure')
3 #Konfiguration des Spektrometers
4 fitwriter('band1:cmdnumspecchan 8192','band2:cmdnumspecchan 8192','cmdusedsections
  1 1','cmdmode int','cmdsynctime 200000','cmdblanktime 2000','cmdnumphases 1','
  configure')

```

Listing 2: Backendkonfiguration über das User Interface

Startet das Radioteleskop eine astronomische Messung, beginnt die Backenddatenübertragung. Dieser Datentransfer ist netzwerkbasierend als TCP/IP im IEEE-Format definiert, welches Tabelle 4 veranschaulicht. Die ersten 64 Bit stellen die Konfigurationsparameter der Backends und des Datenformats dar. Die folgenden Daten definieren zunächst die Po-

larisation sowie den Kanal, auf welchen die zugehörigen Parameter der Empfängerleistung folgen. Im Fall des Kontinuum-Backends resultiert daraus ein Messwert pro Takt, welchen das Spektrometer in 8192 Spektralkanäle darstellt. Aufgrund der Verwendung zweier Polarisationen (vertikale und horizontale), wird dasselbe Format (Polarisation → Kanal → Messdaten) nochmals angefügt. Die Synczeit dient als Taktfrequenz der Datenübertragung. Da nur eine Phase in Verwendung ist, wird mit jedem Takt die Messreihe erweitert. Das implementierte Datenformat dient der Verarbeitung der Rohdaten. Eine Thematisierung dessen erfolgt in Kapitel 3.4.

Aufbau der TCP/IP-Daten:	Spektrometer	Kontinuum-Backend
Header	data[0:4]	data[0:4]
Zeichenkodierung	data[4:8]	data[4:8]
Länge der Daten	data[8:12]	data[8:12]
Name des Backends	data[12:20]	data[12:20]
Zeitmarke	data[20:52]	data[20:52]
Nummer der Phase	data[52:56]	data[52:56]
Backendsection	data[56:60]	data[56:60]
Blockingfaktor	data[60:64]	data[60:64]
Messdaten:		
Polarisation 1	data[64:68]	data[64:68]
Kanal 1	data[68:72]	data[68:72]
Messdaten 1	data[72:32840]	data[72:76]
Polarisation 2	data[32840:32844]	data[76:80]
Kanal 2	data[32844:32848]	data[80:84]
Messdaten 2	data[32848:65616]	data[84:88]

Tabelle 4: Stellt den Datenaufbau der übertragenen TCP/IP-Daten der Backends dar

Steuerrechner, Empfänger und Backends sind über das Zeitsignal IRIG-B angeschlossen, welches mittels eines GPS-Satellitenempfängers mit integriertem Zeitcodegenerator erzeugt wird. Als standardisierter Zeitcode dient dieser verschiedenen Komponenten zur Zeitsynchronisation. Die Parameter, welche die Regeleinheit zur Verfügung stellt sowie die Daten aus den Backends, sind mit Zeitmarken des IRIG-B Zeitsignals versehen. Vor dem Hintergrund, dass Zeiten und daraus resultierend Positionen eine hohe Genauigkeit benötigen, werden alle aufgenommenen Parameter mit Zeitmarken des IRIG-B Zeitsignals behaftet, woraus eine programmtechnische Zeitsynchronisation resultiert. Mit dieser Thematik befasst sich unter anderem Kapitel 3.5.4.

3.4 Entwicklung eines Rohdatenformats

Die Entwicklung des Rohdatenformats diente dem Zweck, große Datenmengen einer radioastronomischen Messung zu verarbeiten und in einem möglichst einfach auszulesendes Datenformat zu speichern.

Daten, die bei einer Messreihe aufgenommen wurden, sind in ein JSON-Format strukturiert.⁴² Aufgrund der großen Datenmenge der Backends, werden die Messsektionen zusätzlich mit dem Python Modul Pickle, welches transiente Daten in persistente Daten (in ASCII-Zeichenkodierung) wandelt, serialisiert. Die unterschiedliche Datenspeicherung hat den Hintergrund, dass die Messwerte des Backends Empfangsleistungen sind, die zunächst ohne Umrechnung durch menschliche Betrachtung keine Aussage haben. Im Gegensatz dazu können die Teleskopparameter, wie beispielsweise der Messmodus oder das beobachtete Objekt, im Rohdatenformat geprüft und bewertet werden.

Vor der Serialisierung der Messdaten in persistente Daten, fand eine Transferierung diese in das folgende Format statt:

```
[[Zeitmarke 1, Messwerte 1][Zeitmarke 2, Messwerte 2] ... [Zeitmarke N, Messwerte N]]
```

Es fiel die Entscheidung, die Daten in einem mehrdimensionalen Array zu definieren und mittels der Python Bibliothek Numpy zu serialisieren. Numpy bietet eine einfache Handhabung großer mehrdimensionaler Arrays und zudem die Möglichkeit durch vorkompilierte Funktionen für mathematische und numerische Routinen Berechnungen durchzuführen. Infolge einer Kalibrierung kann die Empfangsleistung in eine physikalische Einheit transformiert werden. Das Array ist wie folgt aufgebaut: Jede Abtastung resultiert aus einer Zeitmarke und einem Messwert. Für das Kontinuum-Backend ist dies pro Zeitmarke ein Messwert, für das Spektrometer 8192 Spektralkanäle pro Abtastung.

⁴²siehe Anhang A.2, JSON-Keys des Rohdatenformats

Listing 3 stellt ein Minimalbeispiel zum Auslesen der Parameter aus dem Rohdatenformat, einer FFTS-Messreihe, dar.

```
1 value = open(os.environ["savepathffts"], "r") # lese aus File bash-Variable
2 teleskop = json.loads(value.read()) # laden der Rohdaten in das JSON-Format
3 freq = teleskop["frequency"] # Array mit frequenz ueber die Messzeit
4 AZ = teleskop["multicposaz"] # Array mit der Azimutposition ueber die Messzeit
5 # Deserialisierung Pickle
6 backendsection1 = pickle.loads(teleskop["backendsection1"]) # Messdaten der
    horizontalen Polarisisation
7 # Arithmetisches Mittel der Empfangsleistung ueber den Messzeitraum
8 arithm = np.mean(backendsection1,axis = 0)[1:]
```

Listing 3: Beispiel zum Auslesen des Rohdatenformats, einer FFTS-Messreihe

Zunächst wird die Datei ausgelesen. Der Speicherort der Messung ist als Bash-Variable ausgeführt und bietet jedem Nutzer die Möglichkeit, diesen im Vorhinein zu deklarieren. Die Rohdaten werden in das JSON-Format transformiert. Infolgedessen besteht, wie in Listing 3 gezeigt, die Möglichkeit, über definierte Bezeichner⁴³ die Daten der Azimutpositionen sowie der Frequenzen, welche aus einer Messsektion resultieren, auszulesen. Zeile 8 zeigt einen großen Vorteil der Verwendung des Moduls Numpy. Mit Ausführung des Befehls `np.mean()` erfolgt die Bestimmung eines arithmetischen Mittels über die Messwerte und ein Anhängen dieses an das Array `arithm`. Numpy bearbeitet die Daten wie folgt: Die Berechnung beginnt mit dem zweiten Wert in der Liste, da der erste als Zeitmarke definiert ist. Das Programm startet mit Messwert 1, in welchem 8192 Spektalkanäle gelistet sind. Daraus wird der erste Spektralkanal entnommen. Danach folgt eine Betrachtung von Messwert 2, aus welchem erneut der erste Spektralkanal entnommen wird. Dieser Zyklus läuft bis zum Messwert N fort. Das arithmetische Mittel aus den gebildeten Daten wird in dem Array `arithm` als erster Messwert gespeichert. Durch die Entnahme des zweiten Spektralkanals der Messwerte 1 bis N kann aus diesen Daten abermals ein arithmetisches Mittel berechnet werden. Eine Wiederholung des Vorgehens sowie das Anhängen an die Liste `arithm` erfolgt bis zum Spektralkanal 8192.

⁴³siehe Anhang A.2, JSON-Keys des Rohdatenformats

3.5 Kontrollprogramme

Das Blockschaltbild 17 stellt den Aufbau des Kontrollprogramms aus Abbildung 13 dar. Dieses ist in fünf Unterprogramme aufgeteilt, welche die vier primären Aufgabenbereiche des Kontrollprogramms widerspiegeln:

1. User Interface

→ 4mcontrol.py

2. Kontrollprogramm

→ scriptcontrol.py

3. Aktorenkommunikation

→ regler.c

4. Backendkommunikation

→ datawriter.py

→ fftsdatawriter.py

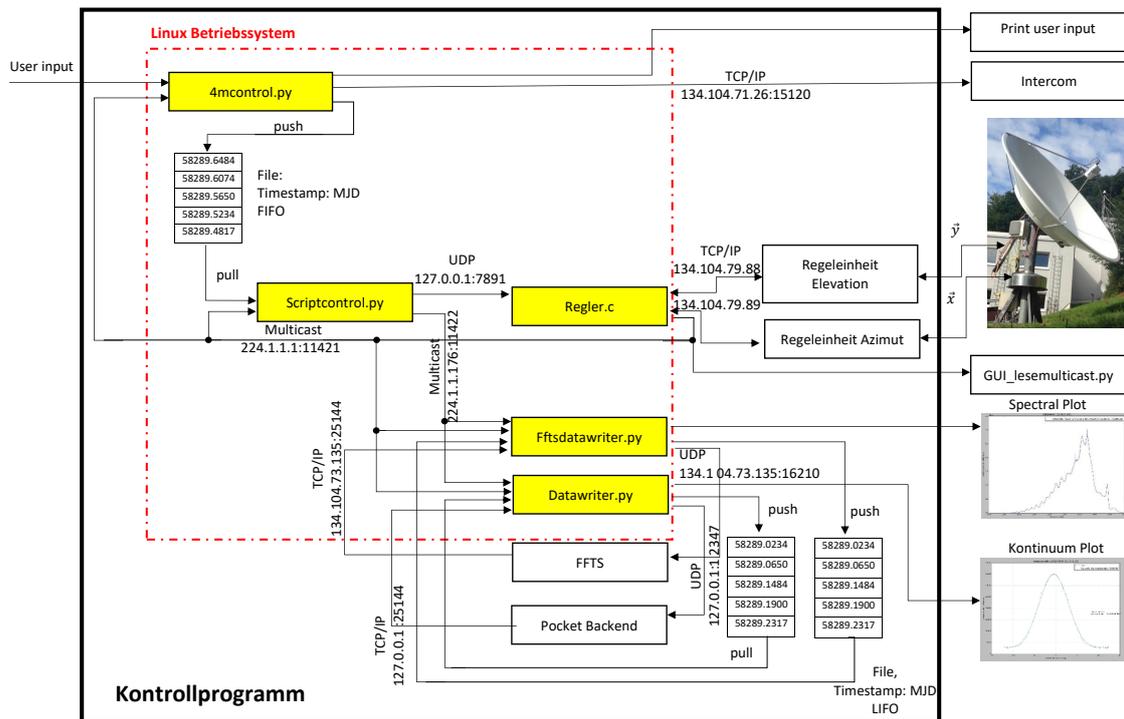


Abbildung 17: Übersicht über das Kontrollprogramm. Gelb markierte Blöcke sind die Unterprogramme, in welchen sich der Entwurf und die Implementierung des Kontrollsystems befinden.

3.5.1 4mcontrol.py

Die Applikation **4mcontrol.py** wurde als Schnittstelle zwischen Benutzer und Kontrollprogramm entwickelt, in welchem das skriptbasierte User Interface aus Kapitel 3.2 implementiert ist. Die Anwendung ist mit zwei Eingängen ausgestattet, welche zum einen dem Benutzer die Möglichkeit der Parametrisierung des Teleskops und der Backends bieten und zum anderen Teleskoppparameter wie Geschwindigkeit, Position usw. über einen Multicast entgegennehmen.

Zudem kommuniziert das Kontrollprogramm über das Netzwerkprotokoll TCP/IP mit dem Intercom, welches ein Steuern der Rauschdiode in der Fokuseinheit sowie Bandumschaltungen in der Direct Detection Unit ermöglicht.⁴⁴

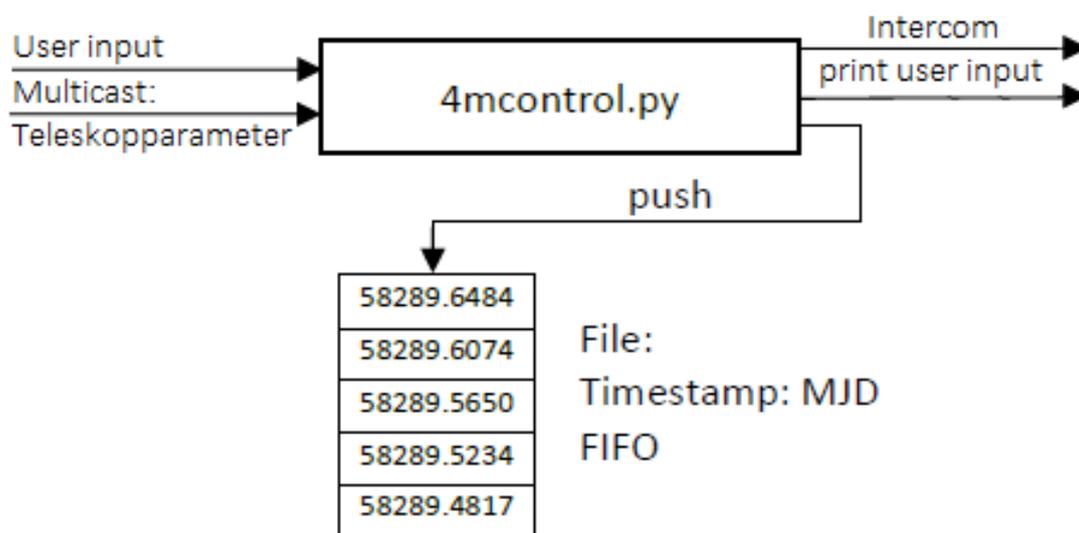


Abbildung 18: Ausschnitt aus Abb.17 zeigt die In- und Outputs des Programms 4mcontrol.py

Die Benutzerkonfigurationen werden in einen Ordner gespeichert, welche mit einem eindeutigen Dateinamen, in der astronomischen Zeiteinheit MJD,⁴⁵ definiert sind. Infolge jeder Teleskoppametrisierung wird das Register erweitert und fungiert als Interface zwischen der Benutzereingabe und dem Programm **scriptcontrol.py**.

⁴⁴siehe Abb. 4 Blockschaltbild des L-Band Empfängers

⁴⁵MJD = JD - 2400 000,5 (2400 000,5 JD = 1858 November 17,0^h UT = 0,0 MJD; vgl. [Keller 2016, S.76])

3.5.2 scriptcontrol.py

Entwickelt wurde die Anwendung **scriptcontrol.py** als Kontrollprogramm, das die Benutzereingabe abstrahiert, radioastronomische Positionen in das lokale Koordinatensystem transformiert und infolgedessen das Teleskop auf die gewünschte Position ausrichtet bzw. radioastronomische Messmodi durchführt. Zudem definiert das Programm den Zeitpunkt des Startens und Stoppens einer Messung.

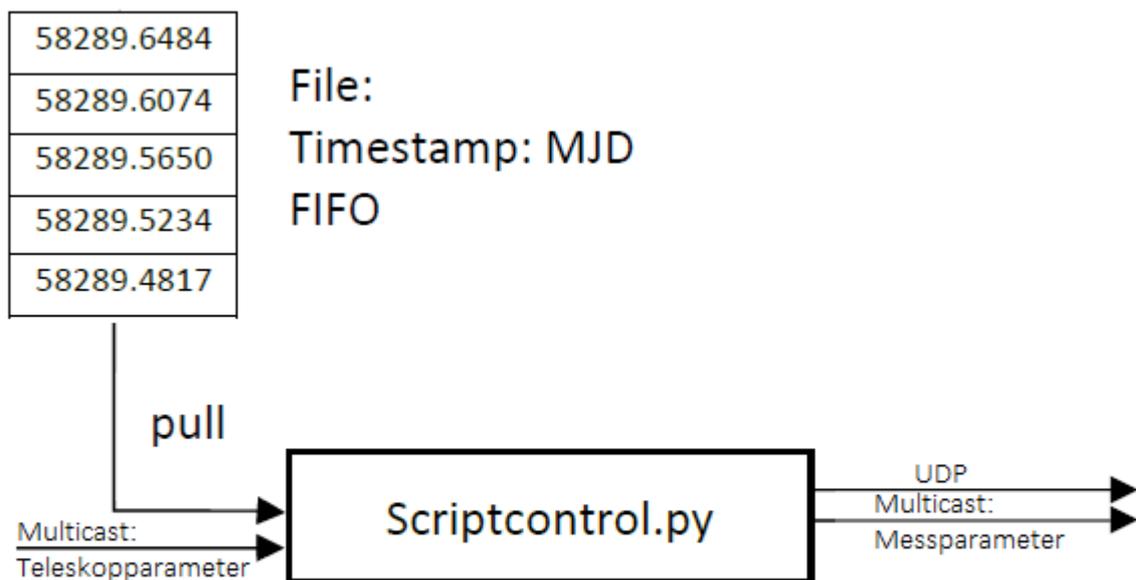


Abbildung 19: Ausschnitt aus Abb.17 zeigt die In- und Outputs des Programms **scriptcontrol.py**

Das Programm ist mit zwei externen Eingängen ausgeführt. Zunächst wird überprüft, ob Daten in dem im Vorhinein definierten Speicherort (Interface: **4mcontrol.py** → **scriptcontrol.py**) vorhanden sind. Ist dies der Fall, werden diese nach dem FIFO-Prinzip verwaltet. Zudem ist die Anwendung mit einem Multicast verbunden, welcher aktuelle Teleskopparameter wie beispielsweise Winkelstellung der beiden Achsen und Rotationsgeschwindigkeiten im Netzwerk zur Verfügung stellt. Außerdem erzeugt die Anwendung einen Multicast mit Messparametern des Teleskops. In diesem wird u.a. signalisiert, ob sich das Teleskop auf Soll-Position befindet und resultierend daraus, ob eine Messung gestartet werden kann. Der UDP-Datentransfer stellt das Interface zwischen dem C-Programm **regler.c** her. Dieses Protokoll enthält die umgerechneten Parametervorgaben um das Teleskop

zu positionieren.

Das Programm **scriptcontrol.py** ist, wie in Abbildung 20 zu sehen, in vier Threads aufgeteilt, welche eine parallele Programmabarbeitung gewährleisten. Die Funktion *recv()* liest die User-Parameter aus, konvertiert die Rohdaten in das JSON-Format⁴⁶ und stellt diese den beiden Threads *multicastmetadata()* und *send()* zur Verfügung. Das Interface zwischen den parallel arbeitenden Programmen wurde als Queue implementiert, welches als Datenstruktur zur Zwischenspeicherung von Objekten dient. Mittels dieses Puffers können parallel laufende Prozesse, welche mit unterschiedlichen Taktraten arbeiten, kommunizieren.

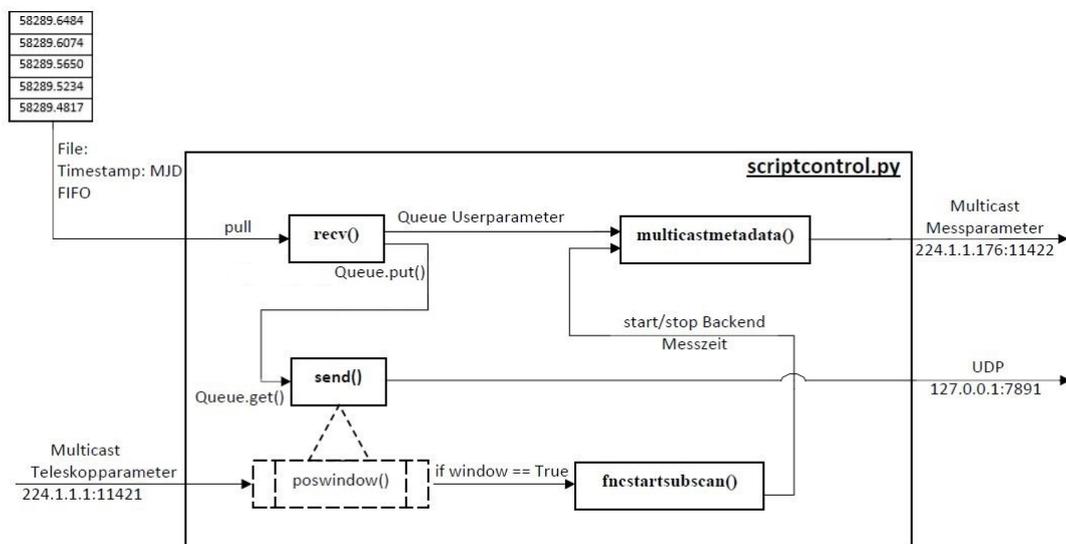


Abbildung 20: Aufbau des Pythonprogramms **scriptcontrol.py**; die vier fett markierten Funktionen sind Threads, die eine parallele Programmabarbeitung ermöglichen.

Der Thread *send()* liest die parametrisierten Benutzer-Eingaben aus der Queue des Threads *recv()*. Zunächst validiert das Programm, auf welches der vier Koordinatensysteme aus Kapitel 2.6 sich der Beobachter bezieht. Unter der Beachtung des Äquinoktiums, J2000.0 oder B1950.0, können die Koordinaten in das horizontale Koordinatensystem transformiert und somit die Teleskopparameter in Azimut und Elevation berechnet werden. Die Koordinatentransformation wird mithilfe des Python Softwarepakets Astropy durchgeführt. Dies ist ein speziell für die Astronomie entwickeltes Programm, mit welchem unter anderem astronomische Koordinatensysteme transformiert werden können. Zudem besteht die Möglichkeit, astronomische Objekte mit ihrer Bezeichnung dem Programm zu übermitteln.

⁴⁶siehe Anhang A.2 JSON-Keys: User Interface

In Listing 4 wird der Anschaulichkeit halber ein Minimalbeispiel der Parametertransformation von äquatorialen Koordinaten zu horizontalen Koordinaten dargestellt. Mithilfe der Funktion `SkyCoord()` werden die vorgegebenen Winkel des Benutzers einem Koordinatensystem zugeordnet. Diese Funktion benötigt zwei Winkelangaben, den Frame, welcher das Bezugssystem definiert sowie die Einheit, in welcher die beiden Winkel definiert sind. In diesem Beispiel hat der Benutzer sich für das äquatoriale Koordinatensystem entschieden, welches unter Astropy ICRS gelistet ist.⁴⁷ Als Bezugsrahmen der Parametervorgabe wurde die Winkeleinheit Grad definiert. Zum Ausrichten des Radioteleskops benötigt dieses Winkelvorgaben im horizontalen Koordinatensystem. Das Kommando `transform_to(AltAz())` kann unter Zuhilfenahme der aktuellen Zeit sowie des Standorts eine Matrizentransformation durchführen, woraus die gewünschten lokalen Winkel Azimut und Elevation resultieren.

```
1 class location():
2     LAT = 50.5251
3     LONG = 6.88195
4     height = 344
5     loc = EarthLocation(LONG, LAT, height)
6 Rektaszension = 84
7 Deklination = 22.5
8 coord = SkyCoord(Rektaszension, Deklination, frame='icrs', unit='deg')
9 altaz = coord.transform_to(AltAz(obstime=Time.now(), location=location.loc))
10 AZ = altaz.az.deg
11 EL = altaz.alt.deg
12 # Azimut = 228.86 deg
13 # Elevation = 54.3 deg
```

Listing 4: Minimalbeispiel zum Transformieren des äquatorialen Koordinatensystems in das horizontale Koordinatensystem

Wie in Kapitel 2 beschrieben, werden dem Azimutaktor Rotationspositionen vorgegeben. Da die Elevationsbewegungen jedoch translatorisch sind, müssen diese mittels trigonometrischer Funktionen umgerechnet werden.⁴⁸ Die Parametereingaben sind als Tupel definiert, welche aus den kommandierten Positionen kombiniert mit einer physikalischen Einheit bestehen. Beispielsweise muss einer übertragenen Position immer eine Winkeleinheit beigelegt werden.⁴⁹ Über ein Pointing-Modell wurde zudem eine Abhängigkeit der Elevationsachse über den Azimutwinkel definiert und eine Funktion interpoliert, welche diese kompensiert. Eine vertiefende Betrachtung des Themas erfolgt in Kapitel 5.

⁴⁷Das ICRS (International Celestial Reference System) ist ein weltweit vereinbartes Bezugssystem von kartesischen Koordinaten des Sonnensystems. Es wird sich auf das Äquinoktium J2000.0 bezogen. vgl. [United States Naval Observatory 2018]

⁴⁸Umrechnung vgl. [Schweikert 2017, S.10f.]

⁴⁹vgl. Kapitel 3.2

Zusätzlich ist das Kontrollsystem so gestaltet, dass nicht nur Koordinaten parametrisiert werden können, sondern auch radioastronomische Messmodi. Diese bieten die Möglichkeit, astronomische Objekte zu verfolgen oder einen Cross-Scan durchzuführen. Der Cross-Scan führt relative Bewegungen, in Azimut oder Elevation, über ein Objekt bzw. eine Position durch.

Die beschriebene Routine, der Funktion *send()*, ist in einer Schleife implementiert. Übermittelt der Thread *recv()* neue Daten, werden diese verarbeitet, andernfalls bleiben die alten Parameter gesetzt. Eine Positionsparametrisierung in horizontalen Koordinaten hält das Teleskop über eine Regelung möglichst präzise auf den vorgegebenen Azimut- und Elevationswinkel. Bei der Wahl eines anderen Koordinatensystems ist das Teleskop ständig in Bewegung und kompensiert die Erdrotation.

Parametrisierte Messaufträge werden zusätzlich zu der gewünschten Position mit einer Zeiteinheit (Dauer der Messung) übertragen. Das Unterprogramm des Threads *send()* → *poswindow()* ist über einen Multicast mit dem Programm **regler.c** verbunden, welcher durchgehend Teleskopparameter, wie beispielsweise den aktuellen Lageistwert des Teleskops, übermittelt.⁵⁰ Diese Routine führt einen Ist-/Sollwertvergleich durch. Infolge einer Teleskopausrichtung wird dies der Funktion *fnstartsubscan()* signalisiert, welches eine radioastronomische Messung initialisiert und ein Kommando an die Anwendung *multicastmetadata()* übermittelt. Nach Ablauf der Messzeit wird ein Interrupt ausgelöst, welcher über *multicastmetadata()* dem Programm **datawriter.py** bzw. **fftsdatawriter.py** übermittelt wird und die Kontinuum- bzw. Spektralmessung beendet.

Kommandiert der Beobachter einen Cross-Scan, wird zunächst die Entscheidung getroffen, ob die Durchführung in der Azimut- oder Elevationsachse stattfindet. Die Position kann der Benutzer in Azimutalenkoordinaten oder mit dem Bezeichner des zu beobachtenden Objekts definieren. Als Benutzereingabe erfolgt zudem die Vorgabe von Messdauer und Fahrbereich (Länge des Scans in Azimut oder Elevation). Der Winkelbereich bestimmt, in welchem relativen Winkel über das Objekt gefahren werden soll. In der Abbildung 21 ist ein Beispiel zur Parametervorgabe eines Cross-Scans dargestellt.

⁵⁰vgl. Anhang A.2 JSON-Keys: regler.c

```

In [1]: crossscan(10000,'sun',lengthx=(15,'d'),offsety=(5,'min'))
crossscan sun (15, 'd') ('0', 'd') 10000 ('cmdintegrationtime 10000',) ('0', 'd')
) ('0', 'd')

In [2]: send()

JSON-String
=====
{"decl": "", "pbebackend": "('cmdintegrationtime 10000',)", "VLAM": "0", "offx":
"0", "SOLL-posel": "0", "unitoffx": "d", "loc_LAT": 50.5251, "SOLL-posaz": "0",
"BET": "0", "VBET": "0", "unitcrosslengthx": "d", "unitcrosslengthy": "d", "loc
height": 344, "unitaz": "d", "unitel": "d", "system": "crossscan", "loc_LONG":
6.88195, "source": "sun", "ra": "", "coordsys": "J2000.0", "unitoffy": "min", "i
ntegrationtime": "10000", "crosslengthel": "0", "fitswriter": "('stop',)", "unit
ra": "h", "galacb": "", "crosslengthaz": "15", "galacl": "", "velposaz": "Schlei
chgang", "velposel": "Schleichgang", "LAM": "0", "command": "false", "unitdec":
"d", "offy": "5"}
=====

In [3]: █

```

Abbildung 21: Beispiel-Parametereingabe mittels des User Interface für einen Cross-Scan über ein Messobjekt (Sonne)

Über das Messobjekt, in diesem Fall die Sonne, erfolgt die Bestimmung der aktuellen Position in Azimut- und Elevationskoordinaten. Zudem ist die Länge des Cross-Scans mit 15° definiert. Diese wird halbiert und bestimmt, dass das Teleskop sich im Azimutwinkel $7^\circ 30'$ vor den aktuellen Standort der Sonne positionieren soll. Beachtet werden muss allerdings die Sinusoidal-Projektion aus Kapitel 2.7.

$$EL = 62,894^\circ \quad (19)$$

$$\text{offsety} = 5' \quad (20)$$

$$e = EL + \text{offsety} = 62,894^\circ + 5' = 62,977^\circ \quad (21)$$

$$\Delta AZ_{\text{Projektion}} = \frac{l_{\text{cross-1/2}}}{\cos(e)} = \frac{7,5^\circ}{\cos(62,977^\circ)} = \pm 16,507^\circ \quad (22)$$

$$\varphi_v = \frac{t_v}{t_m} \cdot \frac{l_{\text{cross-1/2}}}{\cos(e)} \quad (23)$$

$$\varphi_v = \frac{3 \text{ s}}{10 \text{ s}} \cdot \frac{7,5^\circ}{\cos(62,977^\circ)} = \pm 4,952^\circ \quad (24)$$

$$\Delta AZ = \Delta AZ_{\text{Projektion}} + \varphi_v = \pm 21,459^\circ \quad (25)$$

In Formel 19 wurde ein Elevationswinkel der Sonne von $62,894^\circ$ angenommen.⁵¹ Unter Beachtung des beispielhaften Offsets, errechnet sich die Elevationsposition $62,977^\circ$. Resultierend aus der Sinusoidal-Projektion wird die relative Bewegung des Teleskops von $\pm 16,507^\circ$ über die Sonne definiert.

Vor dem Hintergrund, dass das Teleskop während der Messung eine konstante Bewegung durchführen muss, wurde eine Vorhalte- bzw. Nachhaltezeit von $t_v = 3 \text{ s}$ implementiert. Unter Berücksichtigung dieser errechnet sich ein relativer Winkel von $\pm 21,459^\circ$.

Bei vorgegebener Messzeit und relativem Bewegungswinkel, besteht die Möglichkeit, die Rotationsgeschwindigkeit zu bestimmen:

$$\omega = \frac{\varphi}{t} \quad (26)$$

$$t_v = 3 \text{ s} \quad (27)$$

$$\omega = \frac{2 \cdot \Delta AZ}{2 \cdot t_{v+t}} = \frac{2 \cdot 21,459^\circ}{2 \cdot 3 \text{ s} + 12 \text{ s}} = 2,384^\circ/\text{s} \quad (28)$$

In diesem Fall bewegt sich das Radioteleskop während der Messung mit einer Rotationsgeschwindigkeit von $2,384^\circ/\text{s}$. Der Benutzer kann über verschiedene Messzeiten bestimmen, wie schnell die rotatorische Bewegung über das Messobjekt durchgeführt werden soll. Infolgedessen werden mehr oder weniger Messwerte während einer Messreihe aufgenommen.

Ist die Parametrisierung abgeschlossen und die Koordiantentransformation in die Azimut- und Elevationskoordinaten durchgeführt, übermittelt der Programmteil *send()* die Daten über das Netzwerkprotokoll UDP an das Programm **regler.c**.

Außerdem ist **scriptcontrol.py** mit einem zweiten Datenausgang ausgeführt. Der Programmteil *multicastmetadata()* versendet Messparameter⁵² über einen Multicast im Institutsnetzwerk. Dieser Datenstrom enthält alle wichtigen Informationen in Bezug auf die Messung wie beispielsweise den Messmodus, welches Objekt gemessen wird sowie Interrupts, welche

⁵¹Bei einem Beam des Radioteleskops von $3,7^\circ$ sind Positionsangaben mit drei Nachkommastellen nicht prägnant. Es soll lediglich die Möglichkeit aufzeigen, dass eine Parametervorgabe in verschiedenen Einheiten möglich ist.

⁵²im Anhang A.2 sind alle übermittelten Teleskopparameter mit Beschreibung dargestellt.

ein Starten bzw. Stoppen der astronomischen Messung signalisieren. Zudem werden die Backendkonfigurationen des Benutzers übermittelt und von den Programmen **fftsdatawriter.py** und **datawriter.py** verarbeitet. Die Parameter sind wie bei der Übertragung **4mcontrol.py** → **scriptcontrol.py** im JSON-Format definiert.⁵³

3.5.3 regler.c

Das C-Programm dient als Interface zwischen Software und Hardware (Regeleinheiten Azimut und Elevation), welches in die drei Threads *Regler*, *Consumer* und *Multicast* aufgeteilt ist. Die netzwerkbasieren UDP-Parametrisierungen nimmt das Programm **regler.c** entgegen. Die Schnittstelle zwischen **Regler.c** und der Regeleinheit ist von Bosch Rexroth vorgegeben und wird über TCP/IP ausgeführt.

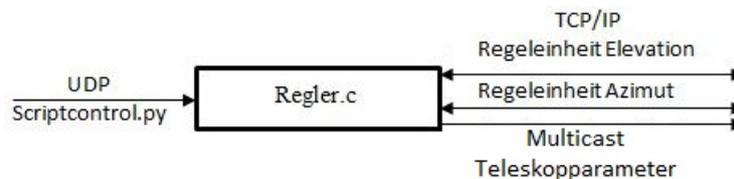


Abbildung 22: Interface zwischen Kontrollsystem und Regeleinheiten: Azimut und Elevation

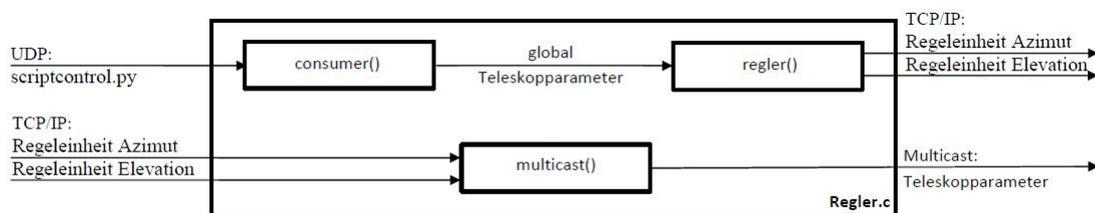


Abbildung 23: Einblick in das Programm regler.c

Im ersten Thread, *consumer()*, werden die Daten des UDP-Transfers (**scriptcontrol.py** → **Regler.c**) verarbeitet und an den Programmteil *regler()* übermittelt. Das Programm **scriptcontrol.py** hat die Werte bereits in die gewünschte Form transformiert. Somit werden beispielsweise die Positionen der Azimutrotationsachse in Grad, die Positionen der Elevationsachse jedoch in Millimeter an die Regeleinheit übertragen.

Für die Hardwarekommunikation (**Regler.c** ↔ Regeleinheiten) stellt Bosch Rexroth Bibliotheksfunktionen zur Verfügung. Listing 5 veranschaulicht die Parametrisierung der beiden

⁵³Multicast-Parameter siehe Anhang A.2

Regeleinheiten. Die Vorgabe der Werte wird mit der Funktion *ealMotionMovementMoveAbsolute()* durchgeführt, welche ein Interface zu den Regeleinheiten herstellt. Die Routine erwartet eine Position, Geschwindigkeit und Beschleunigung. Beide Regeleinheiten, Azimut und Elevation, kommunizieren mit dem zugehörigen Motor. Sie richten, resultierend aus der Benutzereingabe, das Teleskop aus und justieren über eine Regelung ständig neu. Aus diesem Grund wird die beschriebene Routine für jeweils eine Rotationsachse aufgerufen.

```

1 // Programmteil Regler
2 //ealMotionMovementMoveAbsolute(EALHandle, slaveIndex, position, velocity,
   acceleration, deceleration, jerk);
3 resultaz = ealMotionMovementMoveAbsolute(&handleaz, 0, posaz, velaz, accaz, decaz,
   0); //Parametervorgabe Azimut
4 resultele = ealMotionMovementMoveAbsolute(&handleele, 0, posele, velele, accel, decel
   , 0); // Parametervorgabe Elevation

6 //Programmteil Multicast
7 //read parameter
8 ealMotionGetActualPosition(&handle, 0, &actposaz); //aktuelle Position Azimut
9 ealMotionGetActualPosition(&handle, 0, &actposele); //aktuelle Position Elevation

```

Listing 5: Ausschnitt aus dem Programmcode Regler.c; Kommandos der Bosch Rexroth Bibliothek

Im dritten Programmteil *multicast()* werden die aktuellen Parameter, welche die Regeleinheit zur Verfügung stellen, in ein JSON-Format gepackt und über einen Multicast im Institutsnetzwerk verteilt.⁵⁴ Die Auslesung der Encoderwerte der Motoren kann beispielsweise mit der Funktion *ealMotionGetActualPosition()* erfolgen.

3.5.4 datawriter.py und fftsdatawriter.py

Die Python Programme **datawriter.py** und **fftsdatawriter.py** parametrisieren und empfangen Messdaten der beiden angebundnen Backends, Kontium-Backend (**datawriter.py**) und Spektrometer (**fftsdatawriter.py**). Zudem werden Teleskopparameter, Messoptionen und Messwerte in das Rohdatenformat⁵⁵ zusammengeführt und als Messergebnis visualisiert. Abbildung 24 stellt die Implementierung in einem Blockschaltbild dar.

⁵⁴Tabelle im Anhang A.2 mit JSON-Keys und Beschreibung

⁵⁵vgl. Kapitel 3.4

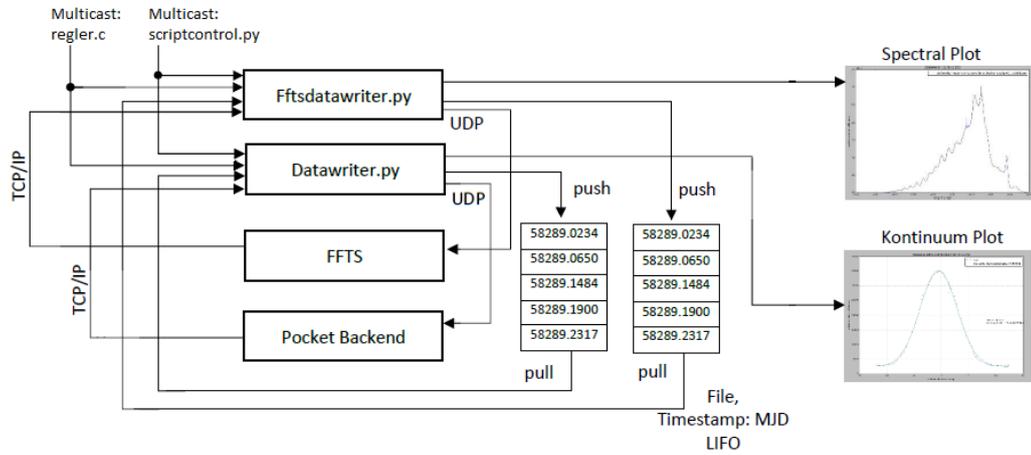


Abbildung 24: Blockschaltbild: Parameterübernahme aus Multicast, Backendkommunikation, Speichern der Rohdaten in MJD und Visualisierung der Messungen

Abbildung 25 stellt in einem Blockschaltbild die sieben implementierten Threads sowie das Interface zur Außenwelt des Algorithmus **datawriter.py** dar.

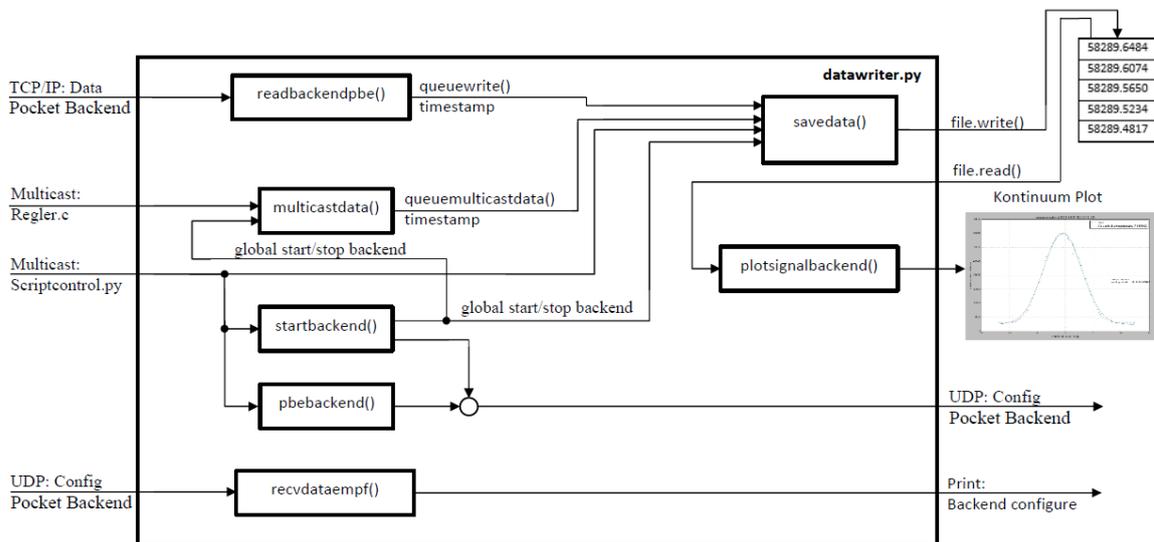


Abbildung 25: Blick in den Programmteil datawriter.py; die sieben Funktionen sind einzelne Threads, welche kombiniert das Backend konfigurieren, Messdaten aufnehmen, diese als Rohdatenformat speichern und visualisieren.

Die Backendkonfiguration ist im Programmteil *pbebackend()* implementiert. Der Multi-

cast⁵⁶ des **scriptcontrol.py** übermittelt die Konfigurationen, welche in das SCPI-Format konvertiert und anschließend mittels UDP an das Backend übertragen werden. Resultierend aus einer erfolgreichen Übertragung, empfängt das Programm *recvdataempff()* ein Echo der Backendparametrisierung und gibt dies auf der Konsole wider. So kann überprüft werden, ob die Daten angekommen sind bzw. ob der Befehl akzeptiert wurde.

Zudem ist der Multicast mit der Funktion *startbackend()* verbunden, welcher Informationen sendet, ob eine Messung gestartet oder beendet wurde. Folglich wird dem Backend sowie dem Thread *savedata()* das Kommando zum Starten bzw. Beenden einer Messung signalisiert.

Der Programmteil *readbackendpbe()* stellt das Interface über TCP/IP zum Backend her und verarbeitet die Empfangsleistung, welche das Empfangssystem dem Kontinuum-Backend übermittelt hat. Die Sockets der TCP/IP-Verbindung sind im Blocking-Mode. Somit wird die Funktion nur bearbeitet, wenn das Backend Messdaten bereitstellt. Wie in Kapitel 3.3 beschrieben, sind die Parameter im EEEI-Format definiert und werden nach Vorgabe der Tabelle 4 Kapitel 3.3 ausgelesen und als Puffer (Datenübertragung: *readbackendpbe()* → *savedata()*) in eine Queue geschrieben. Alle Messwerte, welche das Backend zur Verfügung stellt, sind mit einer Zeitmarke aus dem IRIG-B Zeitsignal versehen.

Multicastdata() ist mit dem Multicast des Programms **regler.c** verbunden, welcher Teleskopparameter,⁵⁷ wie beispielsweise die aktuelle Position, als Rohdaten zur Verfügung stellt. Vor dem Hintergrund, dass unter anderem die Elevationsachse eine transversale Bewegung ausführt, ist zunächst eine Umrechnung in eine Rotationsbewegung notwendig. Das Signalisieren einer Messung bewirkt, dass die empfangenen Parameter über die Messdauer sukzessive in ein Array geschrieben werden. Jedem neuen Parameter ist eine Zeitmarke beigelegt, welche synchron an den Thread *savedata()* übermittelt wird.

Im Thread *savedata()* werden alle Daten, die während der Messung aufgenommen wurden, in das Rohdatenformat aus Kapitel 3.4 fusioniert und in der eindeutigen Zeiteinheit MJD gespeichert. Der Speicherort ist als Bash-Variable deklariert, weshalb vor einer Messsektion eine Definition durch den Beobachter erfolgen muss.

Bei jeder Messung überprüft *plotsignalbackend()*, ob neue Daten im Speicherort hinterlegt

⁵⁶siehe Anhang A.2 JSON-Keys Messparameter → userparameter

⁵⁷siehe Anhang A.2 JSON-Keys: regler.c

sind. Aufgrund des Speichernamens in MJD, ist immer die Visualisierung der aktuellsten Messreihe möglich. Schreibt das Programm *savedata()* in ein neues File, wird der aktuelle Plot beendet, gewartet bis das Schreiben der Datei abgeschlossen ist und darauffolgend ein neues Diagramm erstellt. Die Messdaten des Backends sowie die Teleskopparameter wurden über zwei verschiedene Datenströme mit jeweiligen Zeitmarken in dem Rohdatenformat gespeichert. Infolge der eindeutigen Zuordnung ist eine Synchronisation dieser möglich und als Interpolation im Quellcode implementiert. Außerdem sind in dem Rohdatenformat Messparameter,⁵⁸ wie beispielsweise die Art der Messung, hinterlegt. Für den Fall, dass die Messung als ein Cross-Scan deklariert ist, wird als Messergebnis eine Glockenkurve erwartet und über die aufgenommene Empfangsleistung eine Gaußkurve approximiert. Daraus resultierend ist die Halbwertsbreite sowie die Amplitude der Funktion bestimmbar. Das Programm ist so implementiert, dass die berechnete Position des Objekts als Nullpunkt definiert wird und die Bewegung über dieses im negativen Winkelbereich startet und im positiven Winkelbereich endet. Im Idealfall ist das Maximum der gemessenen Glockenfunktion, an der Position 0, also an der errechneten Position des Objekts. Die Distanz zwischen dem Peak der Glockenfunktion und dem Nullpunkt wird als Pointing-Offset angegeben. Ist die Messreihe als eine einfache Messung im Kontinuum deklariert, wird auf der Abszisse die Zeit in MJD dargestellt.

Der Programmteil **fftsdatawriter.py** basiert auf demselben Aufbau. Der einzige Unterschied ist, dass für Messungen im Frequenzbereich gewöhnlich keine Cross-Scans durchgeführt werden. Für Spektralmessungen wird in *On-Off*-Verfahren beobachtet, d.h. es erfolgt eine Integration einer gewissen Zeit auf der Quelle (*On*) und die gleiche Zeit auf einer Position neben dieser (*Off*). Aus der Differenz wird das Spektrum berechnet als $\frac{On-Off}{Off}$.

⁵⁸siehe Anhang A.2 JSON-Keys: Messparameter

3.6 Regeleinheiten

Das Radioteleskop ist mit zwei Servomotoren und Regeleinheiten, jeweils für die Azimut- sowie die Elevationsachse, ausgeführt. Diese sind mit statischen IP-Adressen im Netzwerk verbunden und stellen über TCP/IP bidirektionale Verbindungen her. Wie in Kapitel 3.5.3 bereits thematisiert, kann unter Zuhilfenahme von C-Bibliotheksfunktionen das Interface zwischen dem Linux Betriebssystem und der Regeleinheit hergestellt werden.

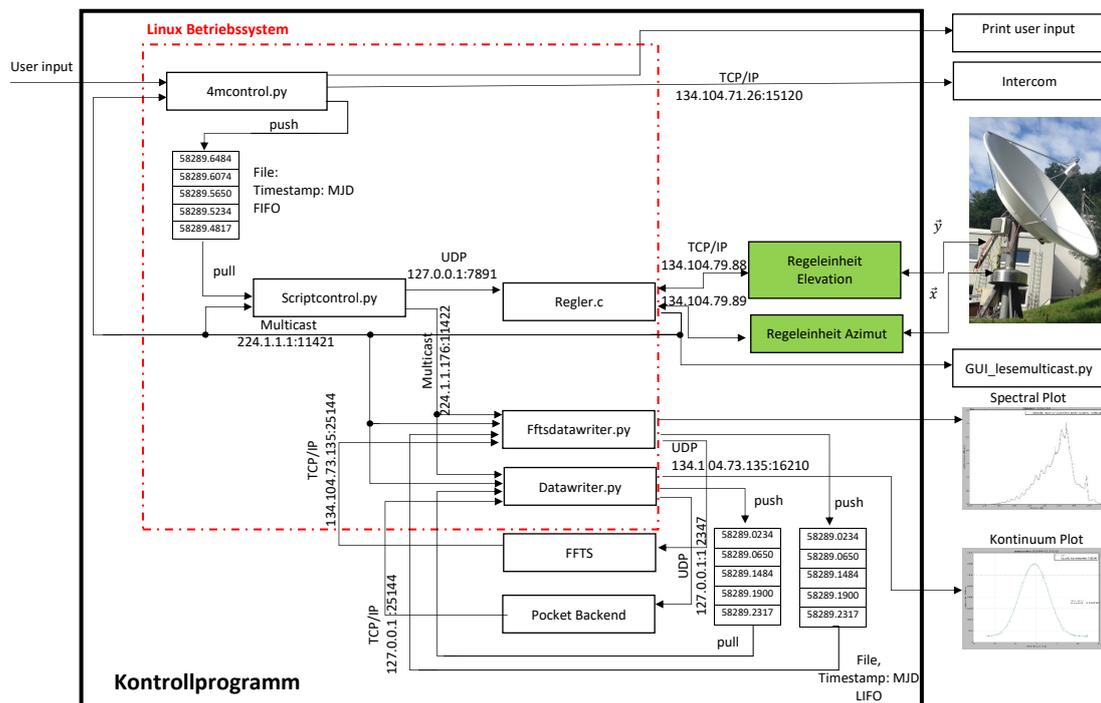


Abbildung 26: Einblick in das Kontrollprogramm. Grün markierte Blöcke sind die Regeleinheiten. Diese stellen das Interface zwischen Software und Hardware her.

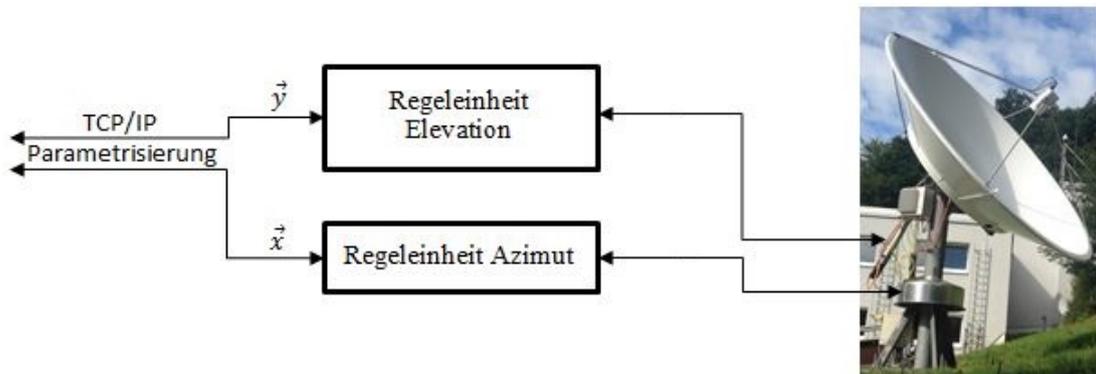


Abbildung 27: Ausschnitt des Kontrollprogramms. Dieses visualisiert die Kommunikation zwischen den Regelunits und den Motoren.

Abbildung 26 und 27 visualisieren die Einbindung der Regelunits in das Kontrollprogramm, welche als Interface zwischen der Softwareprogrammierung und der Hardware dienen. Die Maßeinheiten werden, wie in Formel 29 bzw. 30 definiert, an die Regelunits übertragen, die Parametrisierungen verarbeitet und infolgedessen das Teleskop auf die gewünschte Position ausgerichtet. Die Ethernetverbindung ist bidirektional aufgebaut, mithilfe welcher Teleskopparameter über Encoder ausgelesen und über die Regelunit an das Programm **regler.c** übertragen werden.

$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ t \\ F \end{bmatrix} = \begin{bmatrix} deg \\ 1/min \\ 1/s^2 \\ s \\ Nm \end{bmatrix} \quad (29)$$

$$\vec{y} = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ t \\ F \end{bmatrix} = \begin{bmatrix} mm \\ mm/min \\ mm/s^2 \\ s \\ Nm \end{bmatrix} \quad (30)$$

Bosch Rexroth stellt die Inbetriebnahme-Software *IndraWorks* zur Verfügung. Über diese können beispielsweise die Regelung der Motoren eingestellt oder Endlagen bestimmt werden. Auf die Inbetriebnahme sowie die Hardwarekomponenten, welche Motoren, Winkelwertgeber und Regeleinheiten verwendet werden, wird in dieser Arbeit nicht weiter eingegangen und kann in der Bachelorarbeit: *Automatisierung eines 4-Meter-Radioteleskops im Azimut, basierend auf dem Linux Betriebssystem Ubuntu*⁵⁹ studiert werden.

⁵⁹Schweikert [2016]

4 Messdatenaufnahme

4.1 Erprobung des Kontinuum-Empfangssystems

4.1.1 Cross-Scan-Messung über Elevationsdrehachse

Abbildung 28 zeigt eine der ersten radioastronomischen Messungen der Sonne im Kontinuum. Auf Basis der Messreihen besteht die Möglichkeit, die Implementierung des Steuerprogramms zu verifizieren.

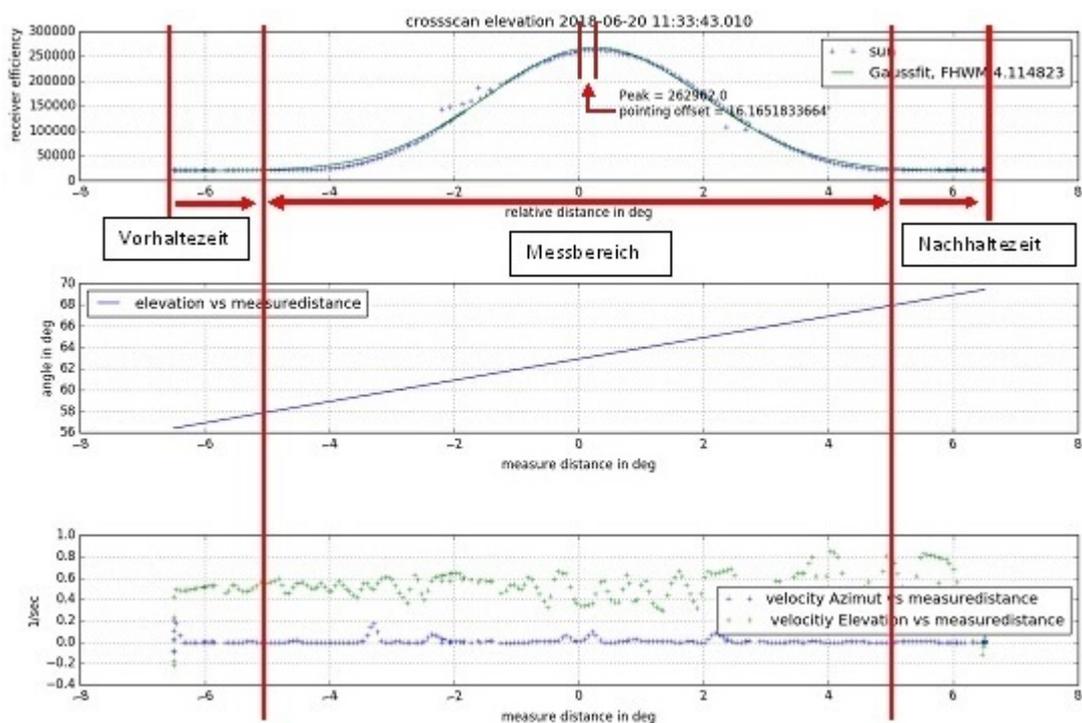


Abbildung 28: Cross-Scan in Elevation über die Sonne; dargestellt ist die Empfangsleistung über den Winkelbereich der Messung, zudem die Teleskopbewegung während der Messung.

Zunächst kann die Aussage getroffen werden, dass die mechanische und programmtechnische Ansteuerung des Teleskops funktioniert. Das System hat die Benutzereingaben verarbeitet und die aktuelle Position des Objekts in astronomische Koordinaten bestimmt. Diese wurden in das lokale Horizontal-Koordinatensystem transformiert. Die beiden Akteure positionieren den Parabolspiegel und stellen mithilfe einer Regelung die benötigte Genauigkeit her. Als Benutzereingabe wurde der Messmodus Cross-Scan, $\pm 5^\circ$ in Elevation, über die Sonne kommandiert. Cross-Scan bedeutet, dass das Teleskop vor ein Messobjekt ausgerichtet wird, mittels einer linearen Bewegung über dieses eine Messung durchführt

und anschließend die Positionierung sowie die astronomische Messung beendet. Wie in Darstellung 28 zu sehen, ist die relative Bewegung infolge der implementierten Vor- und Nachhaltezeit größer als der kommandierte Wert.⁶⁰ Diese Zeiten dienen dem Zweck, dass das Teleskop beschleunigen bzw. verzögern kann, ohne dass dies mit in die Messung eingeht.

Zudem ist ersichtlich, dass infolge der Vorgaben (Dauer der Messung und Länge des Cross-Scans) eine korrekte Rotationsgeschwindigkeit berechnet wird. Das Teleskop führt während der Messung eine gleichförmige Bewegung über das gewünschte Objekt durch und harmonisiert mit dem Empfangssystem. Aus der radioastronomischen Beobachtung resultiert eine symmetrische Glockenkurve, welche weder gekippt, gestaucht, noch abgeschnitten ist. Beim Betrachten der Geschwindigkeiten ist zu erkennen, dass sich das Teleskop während der Messung in der Regelung befindet und infolgedessen permanent auf der optimalen Position gehalten wird.

Der Pointing-Offset ist mit $16'$ für eine Messung mit einem 4-Meter-Radioteleskop.⁶¹ hinreichend genau. Außerdem besteht die Möglichkeit einer additiven Korrektur des Pointings über die Benutzeroberfläche, womit eine Ausrichtung des Teleskops nochmals justiert werden kann. Die Halbwertsbreite der Messung von $4,11^\circ$ ist, mit dem Beam von $3,74^\circ$ des Parabolspiegels⁶² und einem scheinbaren Durchmesser der Sonne von $32'$,⁶³ ein erwartbarer Wert.

Zudem kann die Aussage getroffen werden, dass das Steuerprogramm mit dem Kontinuum-backend kommuniziert. Die parametrisierten Konfigurationen wurden verarbeitet, eine Beobachtung zum gewünschten Zeitpunkt gestartet bzw. gestoppt, die Messdaten an das Kontrollprogramm übermittelt und als Resultat visualisiert.

4.1.2 Cross-Scan-Messung über Azimutdrehachse

Zum Verifizieren der Azimutachse wurde die gleiche Messung nochmals im Azimutrotationsbereich durchgeführt.

Die relative Positionierung von $\pm 5^\circ$ wird mit dem Korrekturterm aus Kapitel 2.7 (Sanson-Flamsteed Projektion) berechnet. Infolge eines Elevationswinkels von $62,88^\circ$ muss eine relative Bewegung von $\pm 10,97^\circ$ durchgeführt werden.

⁶⁰vgl. Kapitel 3.5.2

⁶¹benötigte Auflösung für ein 4-Meter-Radioteleskop: $\phi = 22'26''$ siehe [Schweikert 2017, S.6] \sim FWHM/10

⁶²vgl. Kapitel 2.4

⁶³vgl. [Unsöld u. Baschek 2005, S.171]

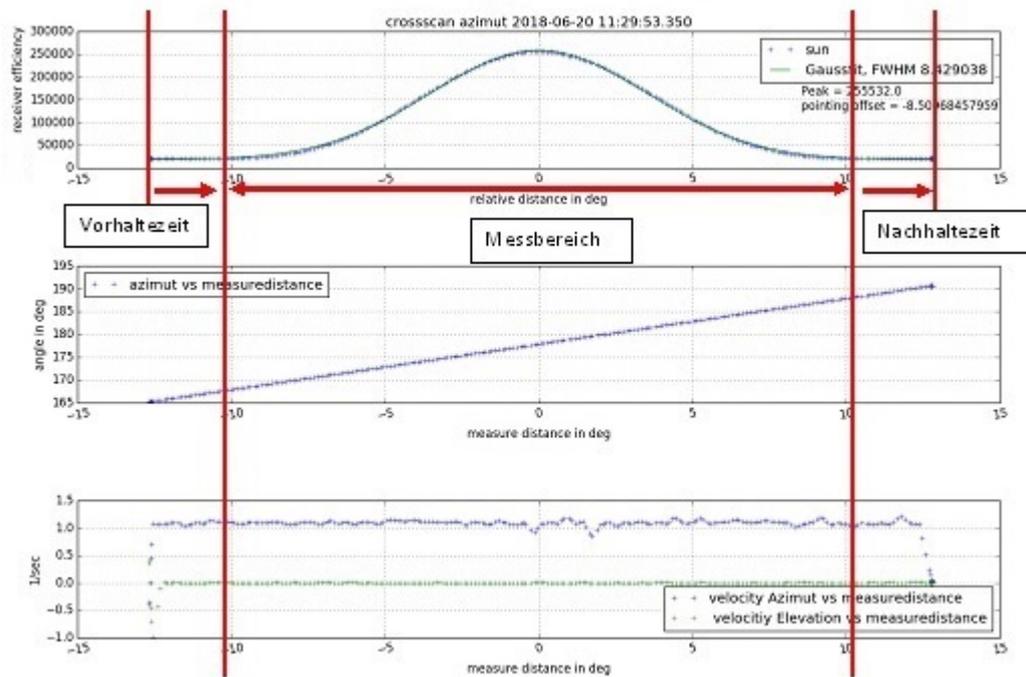


Abbildung 29: Cross-Scan in Azimut über die Sonne; dargestellt ist die Empfangsleistung über den Winkelbereich der Messung, zudem die Teleskopbewegung während der Messung.

Die radioastronomische Beobachtung aus Darstellung 29 ist vergleichbar mit dem vorherigen Elevations-Cross-Scan. Die Glockenkurve ist symmetrisch, zudem weder gekippt noch gestaucht bzw. gestreckt oder abgeschnitten. Außerdem ist die Rotationsgeschwindigkeit des Teleskops im gesamten Messbereich konstant. Das Backend wird zum richtigen Zeitpunkt gestartet und gestoppt. Der Pointing-Offset von $-8,5'$ ist hinreichend genau. Wie zu erwarten ist die relative Teleskopbewegung $\pm 10,97^\circ$, addiert mit der Vor- und Nachhaltezeit.

Bei Betrachtung der Halbwertsbreite fällt allerdings auf, dass diese mehr als doppelt so groß ist wie erwartet. Rechnerisch sollte sie im Bereich des vorherigen Elevations-Cross-Scans liegen. Eine solche Abweichung ist nicht hinnehmbar und muss nochmals evaluiert werden. Damit wird sich Kapitel 4.1.3 vertiefend befassen.

4.1.3 Empirische Evaluation der Halbwertsbreite in Elevation und Azimut

Dieses Kapitel untersucht die Problematik der unterschiedlichen Halbwertsbreiten der Messungen in Azimut und Elevation. Zunächst wurde die Konstanz der vorangegangenen Messungen, unter Zuhilfenahme verschiedener Cross-Scans der Sonne in Azimut und Elevation über den gesamten sichtbaren Winkelbereich, überprüft. Diese Messdatenaufnahme ist in Abbildung 30 aufgezeigt. Es ist zu erkennen, dass die Halbwertsbreiten über einen großen Winkelbereich jeweils konstant bleiben sowie die Parameter der Elevation erwartbar sind. Die Halbwertsbreiten im Azimutrotationsbereich jedoch Unstimmigkeiten aufweisen.

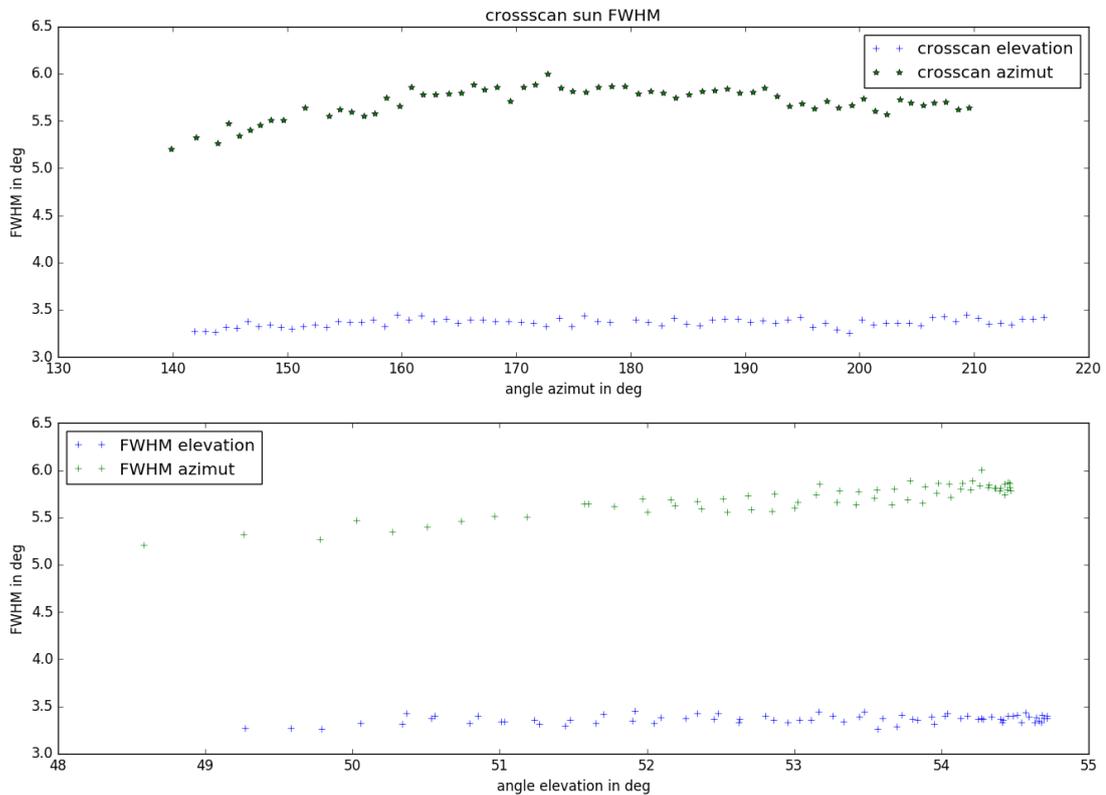


Abbildung 30: Halbwertsbreite der Elevations- (blau) und der Azimutscans (grün) bei Messungen der Sonne. Gemessen wurde über den gesamten Sichtbereich der Quelle. Halbwertsbreite über Azimutwinkel (oberes Schaubild) und über Elevationswinkel (unteres Schaubild) aufgetragen.

Drei mögliche Ursachen wurden in Bezug auf dieses Phänomen untersucht. Zum einen kann es an einem Fehler im Empfangssystem liegen, zum anderen an einem Fehler in der Programmierung des Kontrollsystems. Des Weiteren besteht die Möglichkeit, dass der Fokus nicht richtig ausgerichtet ist und daher der Fokus-Punkt nicht getroffen wird.

Zunächst findet eine Betrachtung des Empfangssystems statt. Da der Empfänger als Kreuzdipol ausgeführt ist, bewirkt eine Drehung der Empfangseinheit um 90° ein Tauschen der vertikalen und horizontalen Polarisation in Bezug auf den Dipol. Im Fehlerfall wird infolgedessen erwartet, dass die Halbwertsbreite des Elevations- und Azimutscans wechseln.

Darstellung 31 und 32 zeigt zwei Messreihen, welche Cross-Scans, in Azimut und Elevation, mit und ohne gedrehtem Empfangssystem darstellen. Gezeigt sind zwei Kanäle: Kanal 1 (horizontale Polarisation) und Kanal 2 (vertikale Polarisation)⁶⁴, jeweils ausgeführt als Azimut- und Elevations-Cross-Scan. Im Falle einer Fehlfunktion des Empfangssystems wurde erwartet, dass sich infolge einer Drehung des Kreuzdipols die Halbwertsbreite des Elevations- und Azimutscans vertauscht. Es ist ersichtlich, dass diese mechanische Änderung keine messtechnische Auswirkung mit sich zieht. Daraus kann die Feststellung getroffen werden, dass das Empfangssystem einwandfrei funktioniert und die beiden Polarisationen homogene Messdaten liefern.

⁶⁴vor der Drehung des Empfängers (in Abbildung 31 und 32 → initial position)

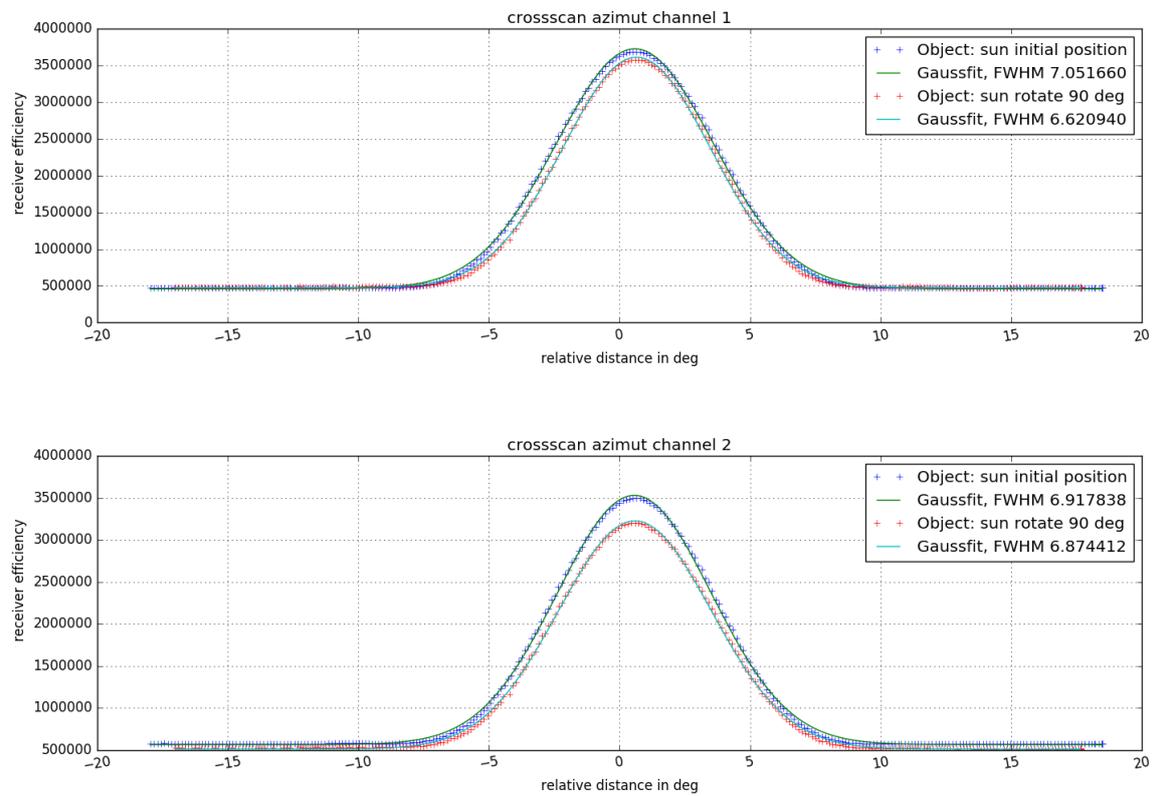


Abbildung 31: Cross-Scan über die Sonne in Azimut, einmal mit Polarisation 1 (oberes Bild) und Polarisation 2 (unteres Bild); Messung in Grundstellung (blau) und mit 90° gedrehtem Empfangssystem (rot).

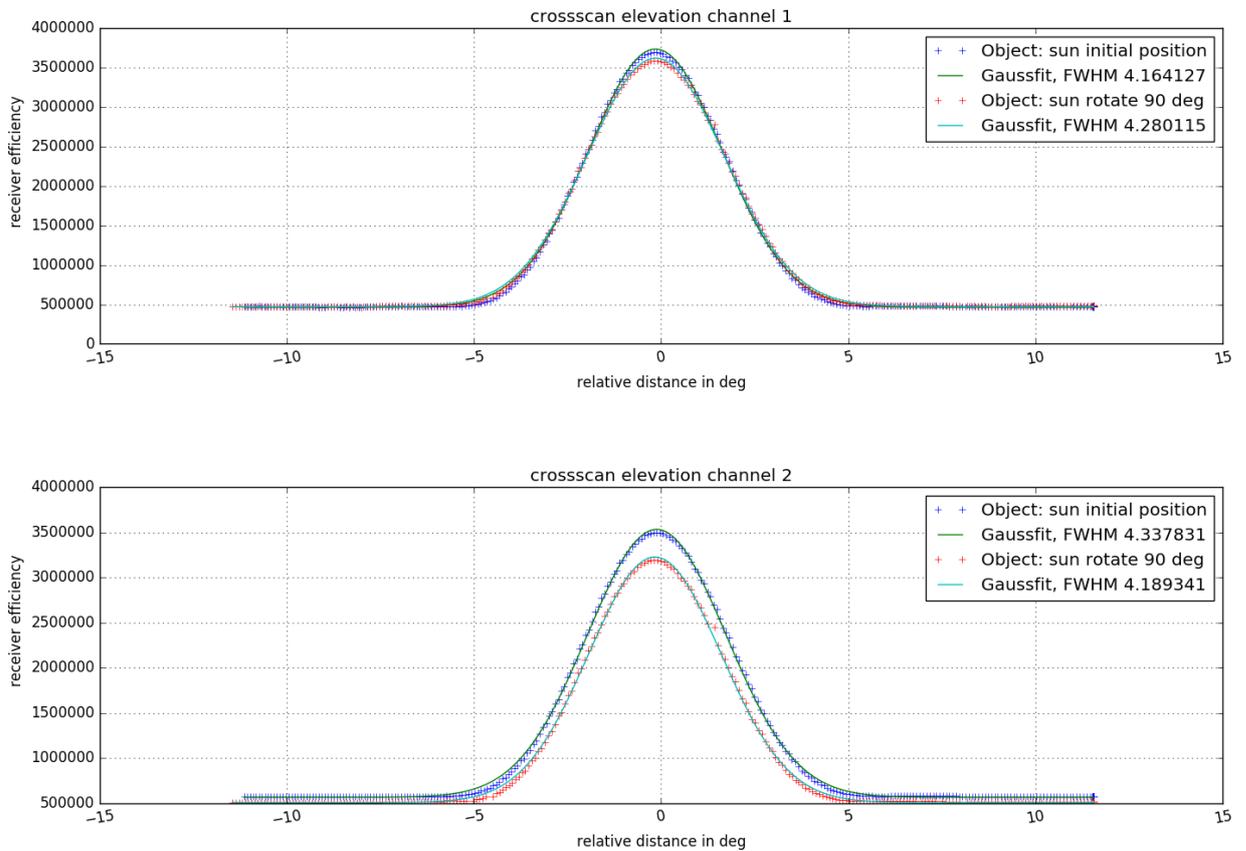


Abbildung 32: Cross-Scan über die Sonne in Elevation, einmal mit Polarisation 1 (oberes Bild) und Polarisation 2 (unteres Bild); Messung in Grundstellung (blau) und mit 90° gedrehtem Empfangssystem (rot).

Vor dem Hintergrund, dass in der Projektarbeit⁶⁵ bereits der Fokus bestimmt wurde, ist ein Fehler in der Softwareimplementierung am wahrscheinlichsten.

Eine Betrachtung der Sinusoidal-Projektion aus Kapitel 2.7 stellt die Lösung dar. Infolge einer Auswertung von relativen Bewegungen im Azimut muss die Transformation aus Kapitel 2.7 zurückgerechnet werden.

Diese Rücktransformation wurde in das Kontrollprogramm implementiert. Das Diagramm 33 stellt jeweils einen Cross-Scan über die Sonne in Azimut und Elevation dar. Wie zu erkennen, ist die Halbwertsbreite nahezu identisch, da eine Abweichung von Azimut zu Elevation

⁶⁵siehe Schweikert [2017]

von 10% innerhalb der Messunsicherheit liegt.⁶⁶ In diesem Fall liegt die Differenz bei 6% und infolgedessen im gewünschten Toleranzfenster. Außerdem sind Pointing-Offset sowie Amplitude annähernd deckungsgleich, was auf eine hohe Güte der Positionierung sowie der Fokusausrichtung schließen lässt.

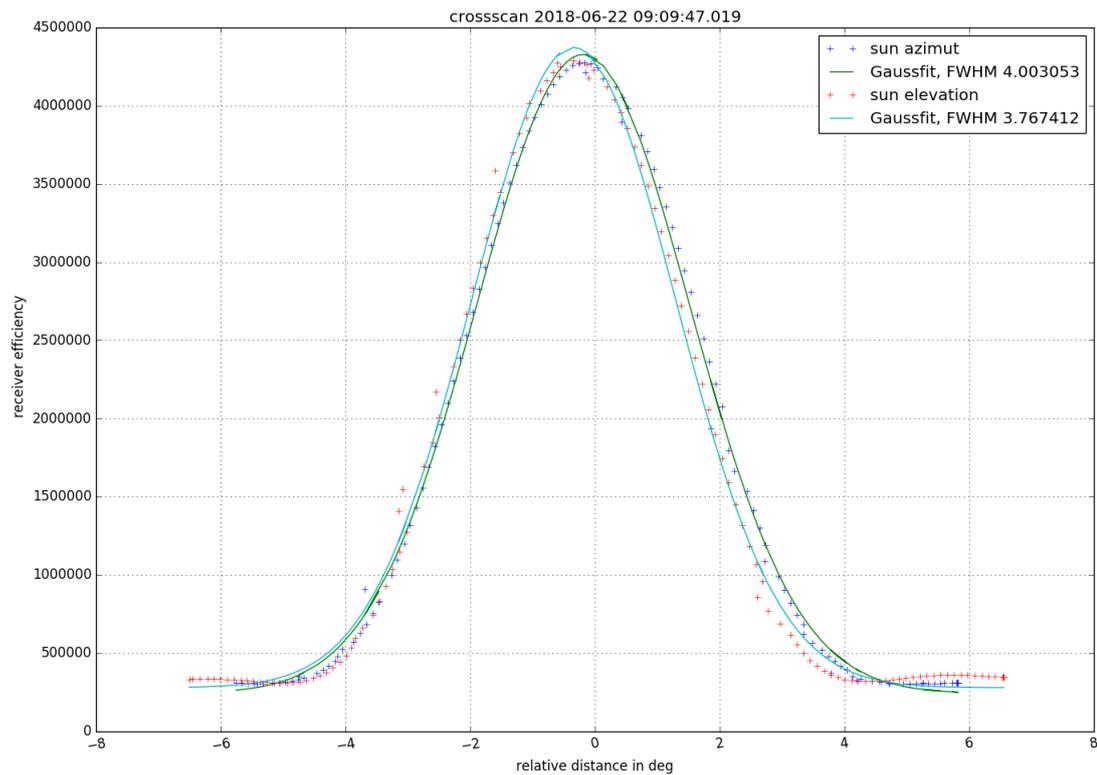


Abbildung 33: Cross-Scan über die Sonne in Elevation (rot) und in Azimut (blau) mit rücktransformierter Sinusoidal-Projektion.

⁶⁶Erfahrungswert der Max-Planck-Instituts für Radioastronomie; Dr. Hafok

4.2 Erprobung des Spektrometers

4.2.1 Spektralmessung des Supernova-Überrests Cassiopeia A

Wie in Kapitel 3.3 thematisiert, ist das Radioteleskop außerdem mit einem Spektrometer verbunden. Zum Testen des FFTS wurde eine radioastronomische Beobachtung des Supernova-Überrests Cassiopeia A im Spektralbereich durchgeführt, welche das Diagramm 34 darstellt. Es kann die Aussage getroffen werden, dass die Konfiguration sowie das Starten und Stoppen parametrisierter Messungen funktioniert. Im Rahmen radioastronomischer Beobachtungen im Frequenzbereich wird ein Objekt bzw. eine Position im Kosmos über einen definierten Zeitraum, wie in diesem Fall ein Supernova-Überrest in der Milchstraße, betrachtet und im Rohdatenformat⁶⁷ gespeichert. Vor der Visualisierung wurde ein arithmetisches Mittel über die Messreihe gebildet, mit dem Vorteil, dass zufällige Messabweichungen gedämpft und konsistente Frequenzen verstärkt werden können. Das Spektrometer besitzt 8192 Spektralkanäle und stellt einen Frequenzbereich von 0 Hz bis 1500 MHz dar. Der Bandpass des Empfangssystems ist mit einer Breite von 1200 MHz bis 1500 MHz ausgeführt. In Abbildung 34 ist die Empfangsleistung logarithmisch aufgetragen. Der Bandpass des Systems ist gut zu erkennen. In diesem Bereich wird die Empfangsleistung dargestellt und außerhalb stark gedämpft.

⁶⁷siehe Kapitel 3.4

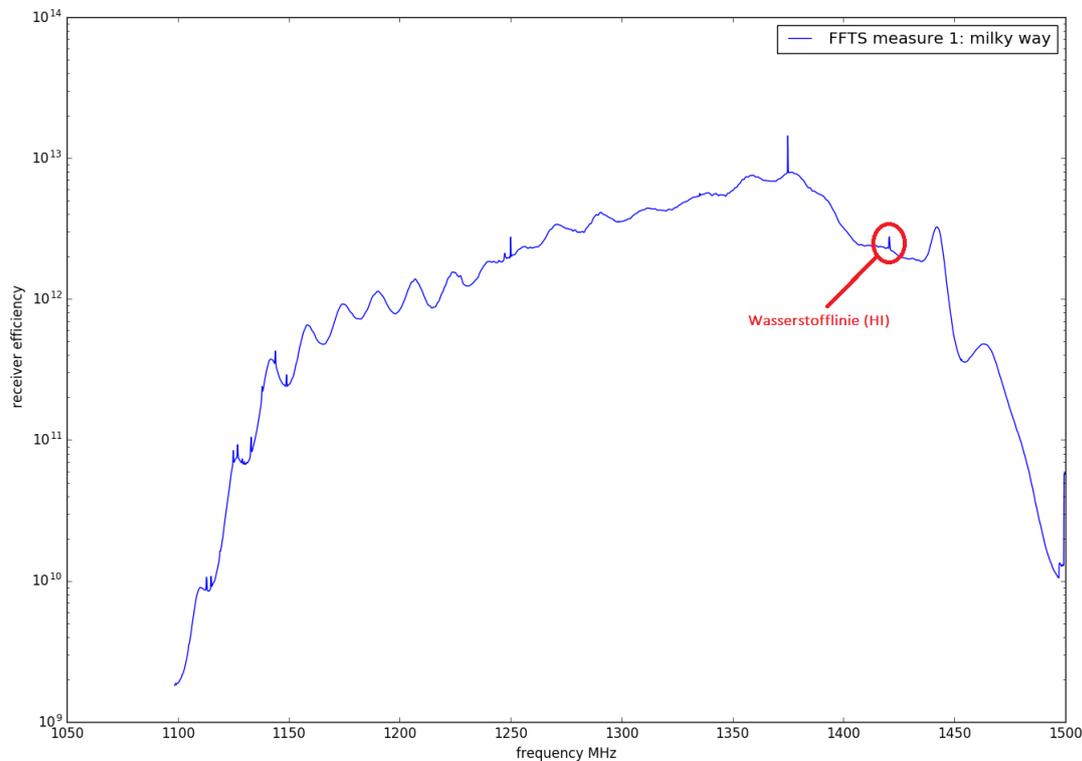


Abbildung 34: FFTS-Messung des Supernova-Überrests Cassiopeia A in der Milchstraße, Frequenzbereich von 1100 MHz bis 1500 MHz dargestellt.

4.2.2 Spektrallinie des atomaren Wasserstoffs (Spinflip-Übergang)

Abbildung 35 stellt die Aufzeichnung zweier radioastronomischer Messreihen dar, zum einen eine *On*-Messung (grüne Kennlinie), in diesem Fall das Objekt Cassiopeia A, dieses befindet sich innerhalb der Milchstraße sowie eine *Off*-Messung. Die Erde befindet sich innerhalb der Milchstraße, somit wird bei der *Off*-Messung aus dieser rausgeschaut. Die intrinsische Beobachtung zeigt einen deutlichen Ausschlag im Bereich einer Frequenz von 1421 MHz, wohingegen die extrinsische Messung dies in einem geringeren Maße aufweist. Die Emissionslinie aus Abbildung 35 zeigt Spin-Flip-Übergänge, bei diesen handelt es sich um den Hyperfeinstrukturübergang zwischen anti-paralleler und paralleler Spinausrichtung im neutralen Wasserstoffatom.⁶⁸ Durch den Übergang eines atomaren Wasserstoffs wird ein Energiebetrag frei, welcher sich in einer Frequenz von 1420 MHz bzw. einer Wellenlänge von 21 cm äußert. Es kann die Aussage getroffen werden, dass sich innerhalb der Milch-

⁶⁸vgl. [Kalvius 2013, S.82f.]

straße vermehrt atomarer Wasserstoff befindet. Der Ausschlag der *Off*-Messung ist damit zu erklären, dass wir aus der Milchstraße heraus schauen und somit Wasserstoff messen. Vor dem Hintergrund, dass die Übergangswahrscheinlichkeit äußerst gering ist und sich im extragalaktischen Kosmos kein HI befindet, ist in diesem Fall der Ausschlag geringer.

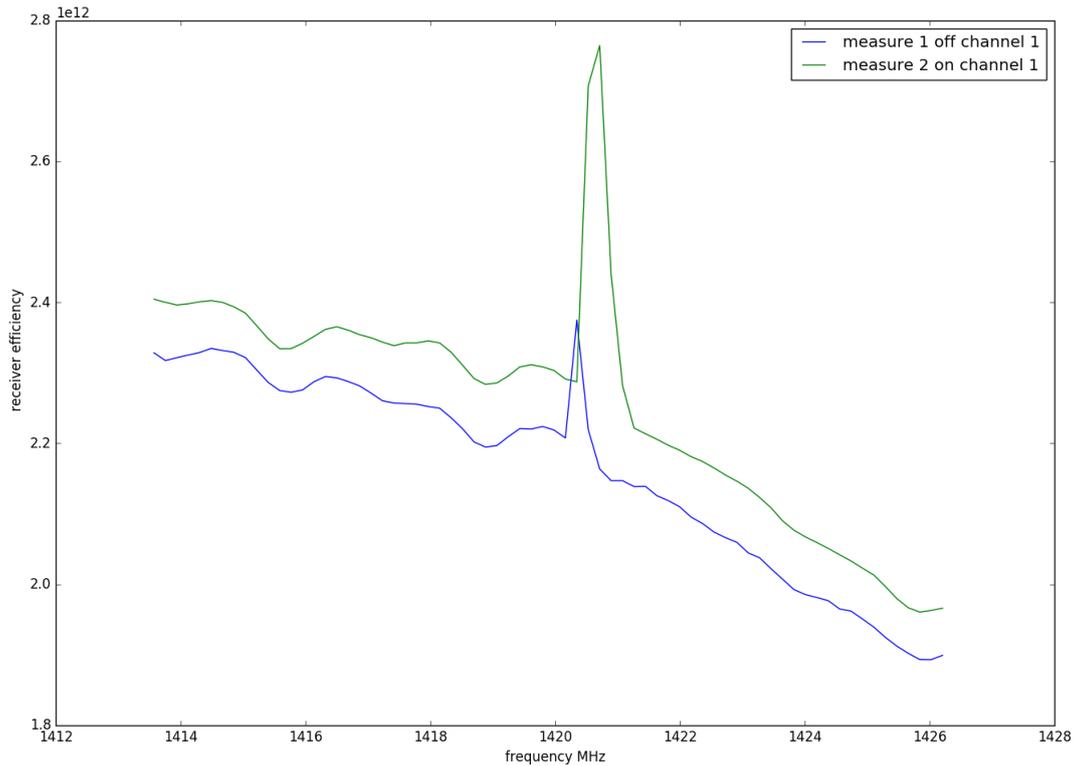


Abbildung 35: Spektralmessung der Objektes Cassiopeia A innerhalb der Milchstraße (grün) und aus dieser hinaus (blau).

5 Entwicklung eines Pointing-Modells

Das Pointing-Modell dient dem Zweck, die Positioniergenauigkeit eines Radioteleskops infolge radioastronomischer Messungen zu verifizieren. Die Durchführung von Cross-Scans bietet die Möglichkeit, den Pointing-Offset⁶⁹ zu bestimmen. Infolge mehrerer Cross-Scans über verschiedene Objekte und Winkel kann die Positionierung über einen großen Winkelbereich validiert werden.

5.1 Betrachtung der mechanischen und radioastronomischen Teleskopcharakteristik

In der Projektarbeit, *Einbau, Justage und Inbetriebnahme eines L-Band Empfangssystems in ein 4-Meter-Radioteleskop*⁷⁰ fand eine Verifizierung der mechanischen Charakteristik des Radioteleskops statt. Über einen Winkelaufnehmer ließ sich eine Korrelation zwischen Elevations- und Azimutwinkel nachweisen. Aus diesem Grund wurde eine Funktion interpoliert, welche diese Abhängigkeit kompensiert. Resultierend aus der vorangegangenen Softwareimplementierung besteht die Möglichkeit einer näheren Untersuchung der radioastronomischen Teleskopcharakteristik. Zur Gewährleistung einer möglichst guten Positioniergenauigkeit des Radioteleskops über den gesamten Himmel, wird die Ausrichtung infolge des Winkelwertaufnehmers und über den radioastronomischen Pointing-Offset verglichen.

Infolge verschiedener radioastronomischer Messungen ist aufgefallen, dass der Pointing-Offset eine große Messabweichung aufzeigt. Aus diesem Grund ist die in Abbildung 36 (links) dargestellte Messreihe aufgenommen worden. Diese Grafik zeigt den Pointing-Offset verschiedener Cross-Scans in Elevation über die Sonne, im gesamten Sichtbereich eines Tages. Der resultierende Pointing-Offset ist über den Elevations- und Azimutwinkel dargestellt.

⁶⁹Die Distanz zwischen dem Peak der Glockenkurve und der wahren Position des Objekts wird als Pointing-Offset angegeben.

⁷⁰Schweikert [2017]

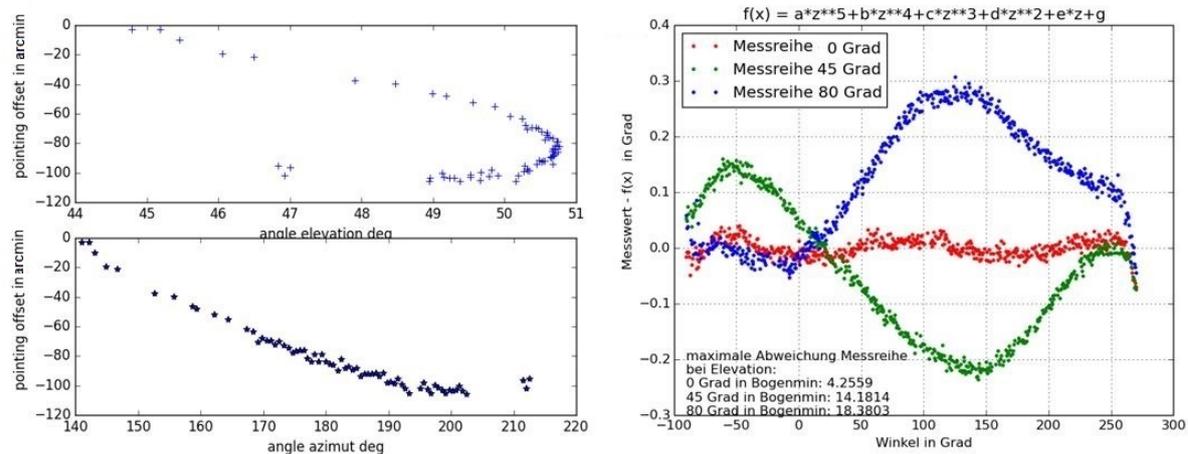


Abbildung 36: Abbildung links: radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne über Elevations- und Azimutwinkel. Resultierend aus verschiedenen Elevations-Cross-Scans. Abbildung rechts: mechanische Teleskopcharakteristik, relative Abweichungen der approximierten Funktion zu den Messreihen der Elevationwinkel 0° , 45° , 80° über den Azimutwinkel.⁷¹

Resultierend aus dieser Messreihe ist eine Abweichung von fast 2° über einen Azimutrotationsbereich von 70° feststellbar. Im Rahmen der Projektarbeit⁷² wurde hingegen im Winkelbereich von 360° , siehe Abbildung 36 (rechts), ein maximaler Messfehler von $18'$ ermittelt.

Da beide Positionsbestimmungsmethoden beim Gegenüberstellen nicht korrelieren, wurde im Folgenden erneut eine radioastronomische Messung durchgeführt, jedoch ohne die implementierte Interpolationsfunktion.⁷³ Im Diagramm 37 ist auf der linken Seite die Messung aus Abbildung 36 (linkes Schaubild) dargestellt. Die rechte Grafik zeigt das Messergebnis ohne Interpolationsfunktion. Beim Vergleich der beiden Diagramme ist zu erkennen, dass sich die Genauigkeit der Positionierung erhöht, wenn die Interpolationsfunktion nicht implementiert ist. Dadurch lässt sich eine Überkompensation der mechanischen Teleskopcharakteristik bestimmen.

⁷¹[Schweikert 2017, S. 23]

⁷²Schweikert [2017]

⁷³vgl. [Schweikert 2017, S.23], bzw. siehe Abb. 36, rechtes Diagramm $f(x)$

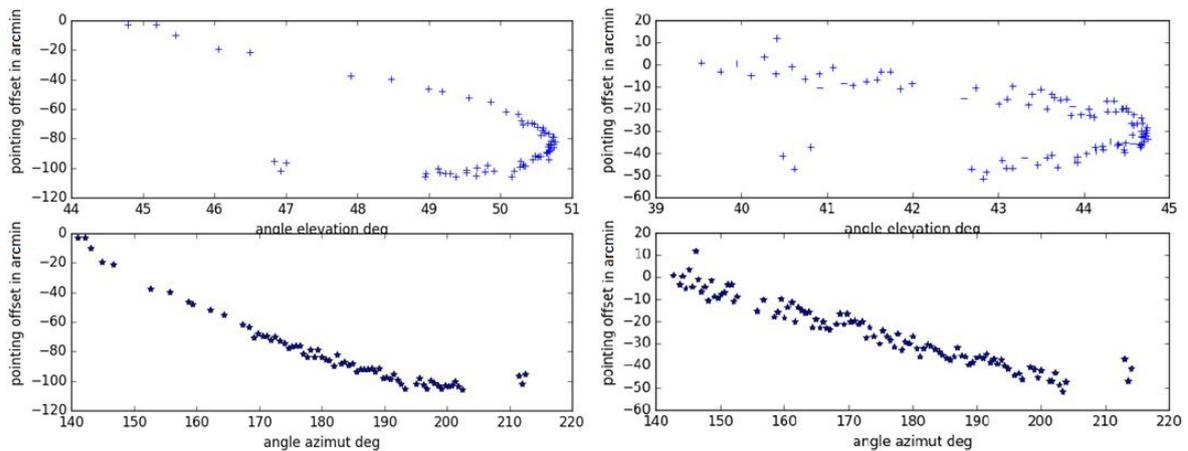


Abbildung 37: Abbildung links: radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, verschiedener Elevations-Cross-Scans, über Elevations- und Azimutwinkel, mit der durch Winkelaufnehmer bestimmten Interpolation. Abbildung rechts: radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, verschiedener Elevations-Cross-Scans, über Elevations- und Azimutwinkel, ohne der durch Winkelaufnehmer bestimmten Interpolation.

In der Projektarbeit⁷⁴ wurde der Winkelaufnehmer am Rand des Parabolspiegels befestigt, da eine Anbringung direkt am Apex-Punkt nicht möglich war. Eine nicht hinreichend hohe Steifigkeit des Spiegels könnte zu der Ungenauigkeit der Positionierung geführt haben. Da die Projektarbeit das Ziel hatte, das Teleskop so auszurichten, dass Objekte im Kosmos detektiert werden können und dies offensichtlich der Fall ist, wird sich mit der Eruiierung des Effekts nicht beschäftigt. Vor diesem Hintergrund erfolgt die Bestimmung der Genauigkeit des Radioteleskops durch wiederholte Cross-Scans.

⁷⁴Schweikert [2017]

5.2 Pointing-Modell Elevation

Abbildung 38 zeigt den Pointing-Offset mehrerer Elevations-Cross-Scans über den Elevations- und Azimutwinkel ohne Interpolation. Mit einer Messabweichung von circa einem Grad genügt die Positioniergenauigkeit des Radioteleskops nicht den Anforderungen. Es wird mindestens eine Präzision von $22'26''$ benötigt.⁷⁵ Vor dem Hintergrund, dass der Winkel der Elevation doppeldeutige Pointing-Offsets besitzt, kann diese Abweichung nicht interpoliert werden. Eine Steigerung des Azimutwinkels führt zu einer progressiven Zunahme der Messabweichung. Daraus kann geschlossen werden, dass die beiden Rotationsachsen korrelieren. Bei Betrachtung des Pointing-Offsets über den Azimutwinkel ist die Aussage zu treffen, dass diese mit einer negativen Geradengleichung bei steigendem Azimutwinkel fällt.

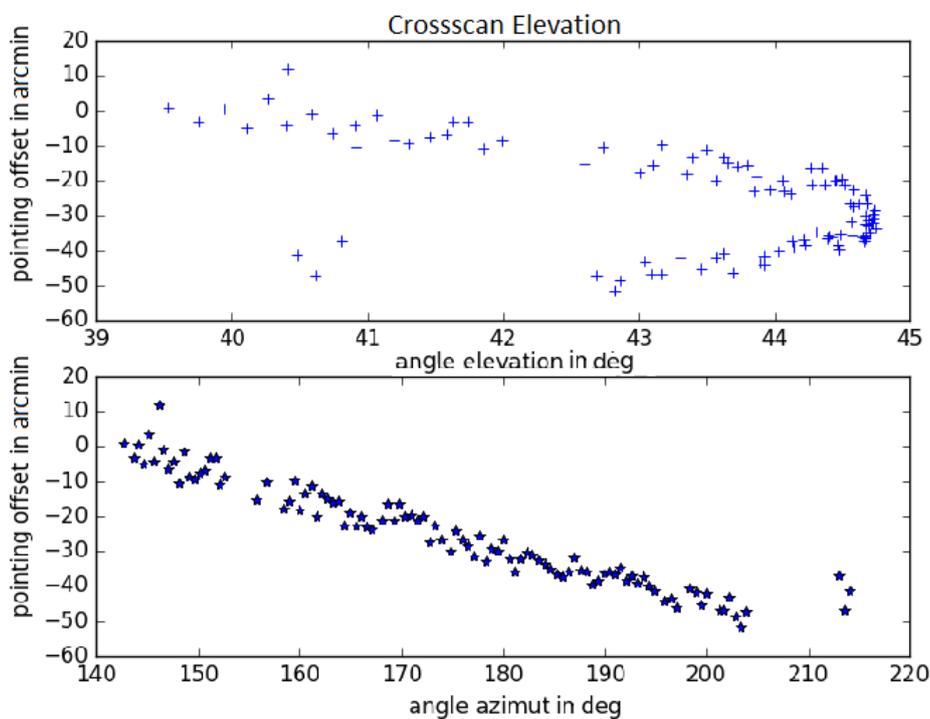


Abbildung 38: Radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, verschiedener Elevations-Cross-Scans, über Elevations- und Azimutwinkel, ohne der durch Winkelaufnehmer bestimmten Interpolation.

⁷⁵vgl. [Schweikert 2017, S.6] \sim FWHM/10

Resultierend aus den Messwerten wurde folgende Geradengleichung interpoliert und in das Kontrollprogramm eingebunden:

$$fitEL(AZ) = -0,0122678 \cdot AZ + 1,72495 \quad (31)$$

Mittels der genannten Funktion erfolgte nochmals eine radioastronomische Messung über die Sonne. Wie in Abbildung 39 zu erkennen, ist der Pointing-Offset mit einer maximalen Abweichung von knapp 17' Bogenminuten für ein 4-Meter-Radioteleskop hinreichend genau. Zudem ist in dem Graph keine Funktion zu bestimmen, welche interpoliert werden könnte und infolgedessen eine signifikante Verbesserung der Positioniergenauigkeit aufweisen würde. Eine Bewertung der Aussagekraft dieser Messreihen erfolgt in Kapitel 5.4.

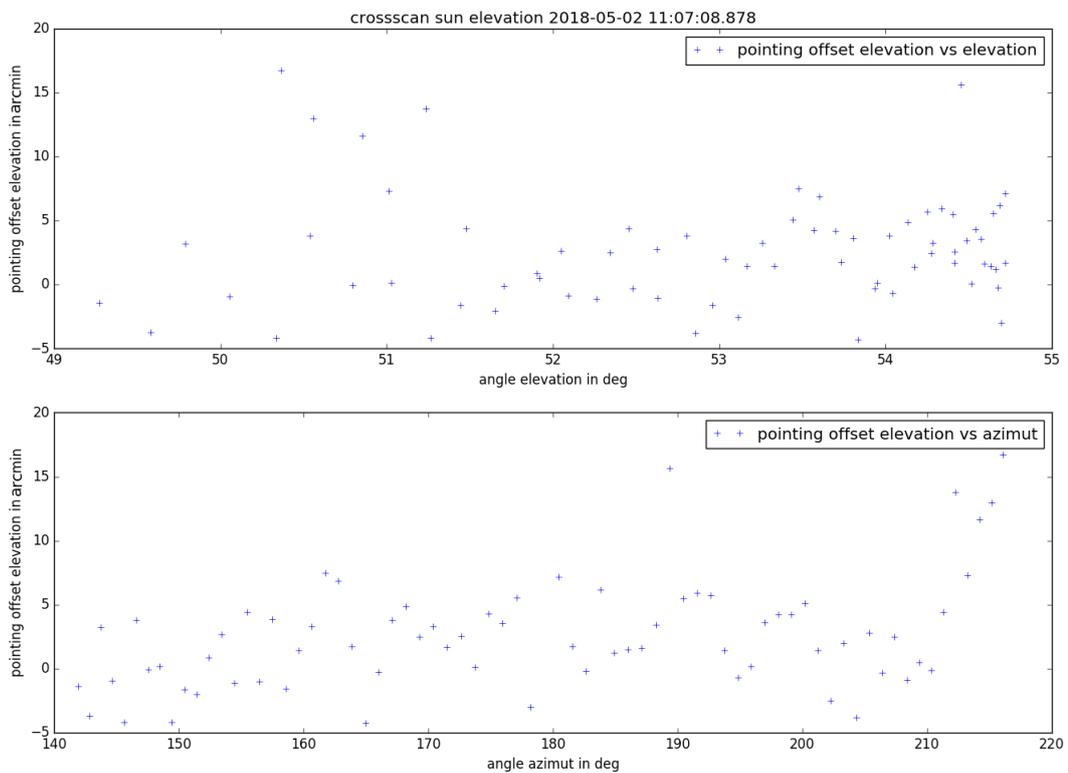


Abbildung 39: Pointing-Offset verschiedener Elevations-Cross-Scans mit interpolierter Geradengleichung.

5.3 Pointing-Modell Azimut

Nicht nur die Präzision der Elevationsachse muss betrachtet werden, sondern auch die Positioniergenauigkeit der Azimutalbewegung. Aus diesem Grund fand das gleiche Messverfahren in der Azimutrotationsachse Anwendung. Es wurden Azimut-Cross-Scans, über einen ganzen Tag verteilt im Sichtbereich des Teleskops, über die Sonne durchgeführt. Die resultierenden Pointing-Offsets sind in Abbildung 40, über die Elevation und den Azimut, dargestellt.

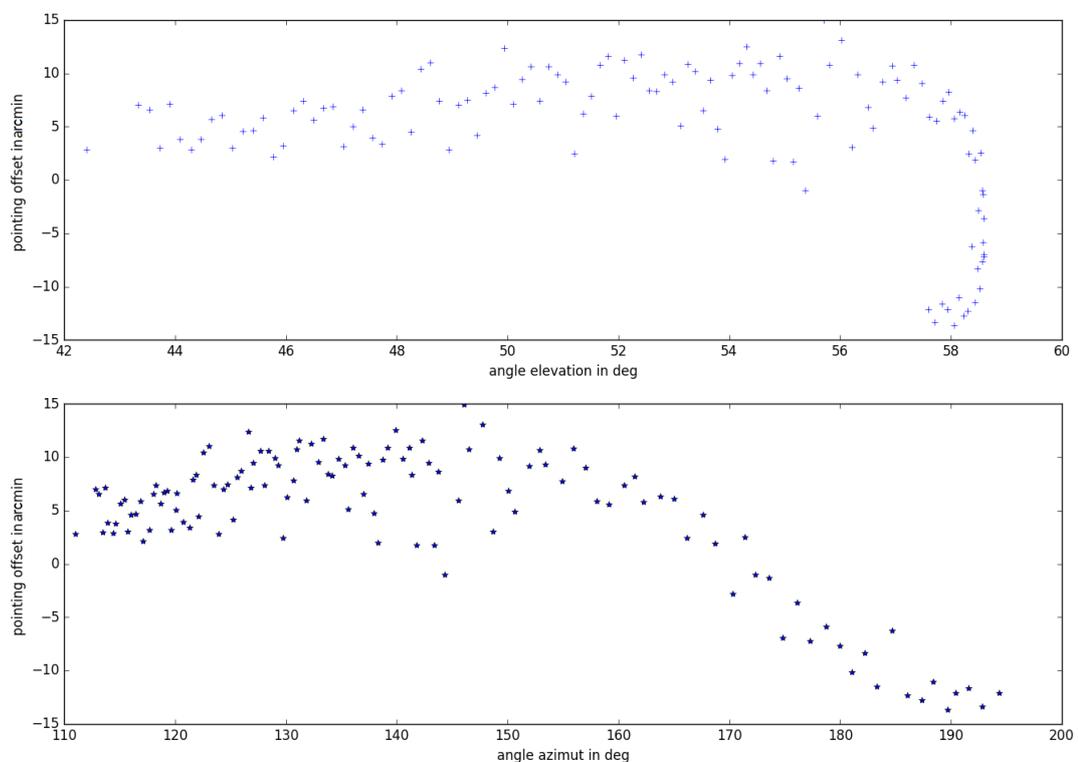


Abbildung 40: Radioastronomische Teleskopcharakteristik, Pointing-Offset der Sonne, mehrere Azimut-Cross-Scans, über Elevations- und Azimutwinkel.

Es ist zu erkennen, dass die Präzision der Azimutachse bereits hohe Güte aufweist. Mit einer maximalen Abweichung von $15'$ ist die Positionierung innerhalb des Toleranzbereichs.⁷⁶ Aus diesem Grund wird zunächst auf eine Interpolation der Messwerte verzichtet. Es besteht zwar die Möglichkeit der Annahme, dass der Pointing-Offset als Cosinusfunktion interpolierbar ist, jedoch würde diese Gewichtung zu keiner maßgeblichen Verbesserung

⁷⁶benötigte Auflösung $22'26''$ vgl. [Schweikert 2017, S.6] \sim FWHM/10

der Genauigkeit beitragen. Zudem erfolgte die Messung lediglich im Bereich von 110° - 200° (Azimut) und 42° - 49° (Elevation). Bei Betrachtung des Winkelbereichs von 160° bis 200° ist der Pointing-Offset monoton fallend. Jedoch kann im höheren Winkelbereich keine Aussage getroffen werden, ob dieses Verhalten weiter anhält, stagniert oder wieder steigt. Bei höheren Azimutwinkeln befand sich die Sonne unterhalb des Horizonts, weswegen keine weiteren Messungen möglich waren. Außerdem ist eine Bewertung der Positioniergenauigkeit des Radioteleskops unterhalb des Horizonts irrelevant. Zur Bestimmung eines größeren Azimutalrotationsbereichs muss auf einen höheren Sonnenstand gewartet oder andere Objekte beobachtet werden. Aus diesem Grund kann keine quantitative Aussage über den weiteren Verlauf des Pointing-Offsets außerhalb des gemessenen Winkelbereichs getroffen werden.

5.4 Evaluation des Pointing-Modells

Problematisch in Bezug auf die Positionsbestimmung über den Pointing-Offset ist der eingeschränkte Messbereich. Es kann immer nur die Position an dem Punkt bestimmt werden, wo sich die Quelle am Horizont, welcher in Bezug auf den Standort des 4-Meter-Radioteleskops in Abbildung 41 dargestellt ist, befindet. Der rot markierte Kasten ist der Bereich, in dem die Interpolation der Positionen definiert wurde. Außerhalb dieses Fensters ist momentan keine Aussage über die Positionsgenauigkeit des Radioteleskops möglich.

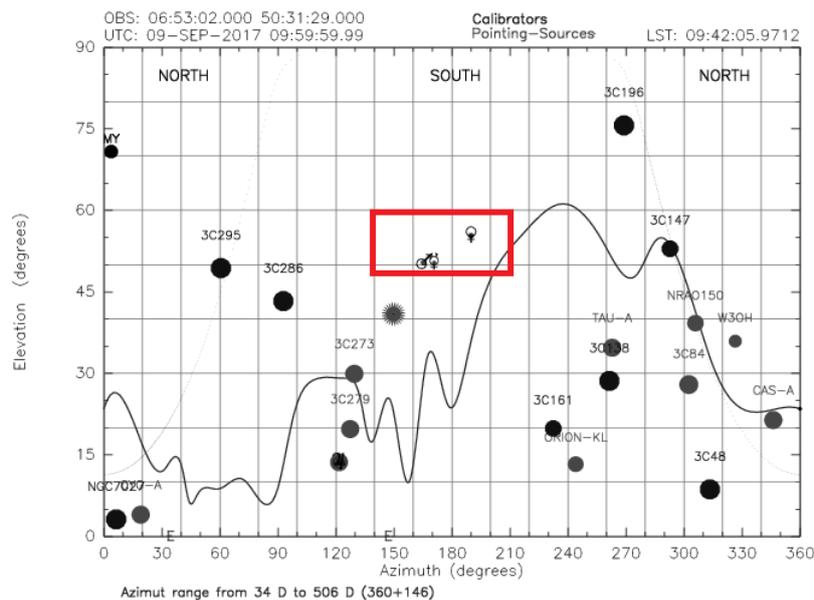


Abbildung 41: Horizont am Standort des 4-Meter-Radioteleskops, Elevations- über Azimutwinkel, rot markiert ist der Winkelbereich in dem die Interpolation aus dem Pointing-Offset definiert ist.⁷⁷

Ziel ist es über verschiedene Radioquellen im Kosmos das Fenster der Messdateninterpolation, in welchem die Positionsgenauigkeit bestimmt ist, zu vergrößern. In Abbildung 42 ist der Sonnenverlauf des Standorts Bad Neuenahr-Ahrweiler dargestellt. Durch Messungen über einen Zeitraum von einem halben Jahr, kann mittels des Pointing-Offsets, die Genauigkeit in diesem Winkelbereich erweitert werden. Zusätzlich sind über diese Beobachtungen nicht nur die Optimierung der Positionsgenauigkeit des Teleskops möglich, sondern es kann ebenso auch eine Bestimmung der Sonnenaktivität im 21 cm Frequenzbereich, über einen längeren Zeitraum, erfolgen. Zudem wäre eine Erueirung über die Nachweisbarkeit von Sonnenaktivität, mit dem 4-Meter-Radioteleskop, möglich.

⁷⁷[Schweikert 2017, S. 8]

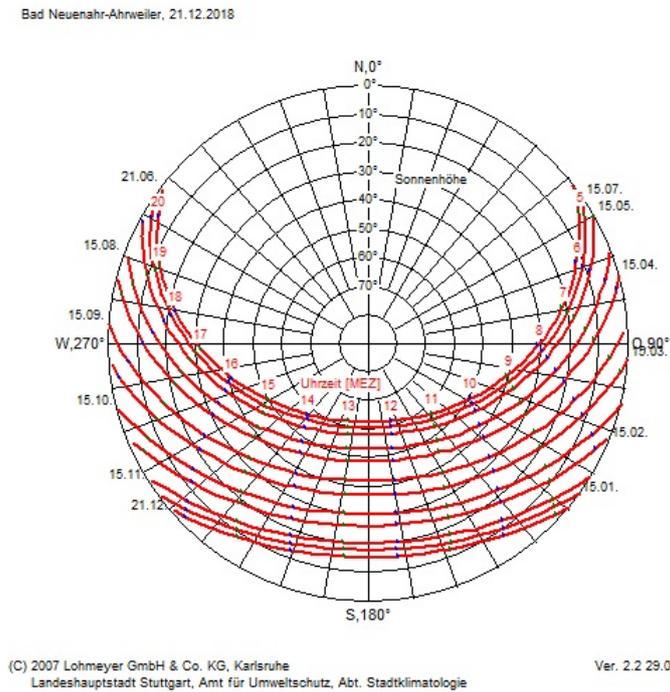


Abbildung 42: Sonnenstand über Bad Neuenahr-Ahrweiler im Laufe eines Jahres.⁷⁸

Testweise sind bereits weitere Objekte mit dem 4-Meter-Radioteleskop beobachtet worden (siehe folgendes Kapitel 6). Eine systematische Untersuchung steht hier noch aus.

⁷⁸Amt für Umweltschutz - Abteilung Stadtklimatologie [2018]

6 Radioastronomische Messungen

Das folgende Kapitel beschäftigt sich mit den ersten radioastronomischen Messungen im Kontinuum. Es soll getestet werden, welche Quellen mit dem 4-Meter-Radioteleskop sichtbar sind und ob die gemessenen Werte mit den theoretischen Parametern übereinstimmen.⁷⁹

$$t_{\text{cal}} = 20 \text{ K} \quad (32)$$

$$\Gamma = 0,0036 \text{ K} \quad (33)$$

$$T_{\text{cal}} = \frac{t_{\text{cal}}}{h_{\text{cal}}} \quad (34)$$

$$T_{\text{sys}} = T_{\text{cal}} \cdot h_{\text{bl}} \quad (35)$$

$$T_{\text{A}} = T_{\text{cal}} \cdot A \quad (36)$$

$$S = \frac{T_{\text{A}}}{\Gamma} \quad (37)$$

In Abbildung 43 ist ein Cross-Scan über Taurus A (Krebsnebel) dargestellt. Dieses Objekt liegt im Sternbild Stier und ist ein Überrest einer Supernova-Explosion, welche starke Kontinuum-Strahlung emittiert und im L-Band detektierbar ist.⁸⁰ Taurus A wurde mittels dieser Messung eindeutig detektiert. Zur Kalibration des Signals wurde die Rauschdiode additiv geschaltet. Über die bekannte Stärke t_{cal} , vgl. Kapitel 2.5, kann mit Hilfe der Höhe des Sprungs h_{cal} die Kalibrationstemperatur T_{cal} bestimmt werden. Diese Beziehung ermöglicht eine Bestimmung der System- T_{sys} und Antennentemperatur T_{A} . Um das Messergebnis mit anderen Radioteleskopen vergleichen zu können, wird die Antennentemperatur in eine Flussdichte S transformiert. Die Systemtemperatur des Empfängers hat eine Temperatur von 141 K, welche unter anderem von dem ungekühlten Empfänger (Eigenrauschen) kommt sowie den Spillover des Parabolspiegels, mit welchem beispielsweise der Boden oder der Wald zusätzlich thermische Strahlung in die Antenne einspeist. Die ermittelten Werte, aus Tabelle 5, sind mit den zuvor angenommenen konsistent.

⁷⁹vgl. Formeln Kapitel 2.5

⁸⁰vgl. Max-Planck-Institut für Radioastronomie [2018]

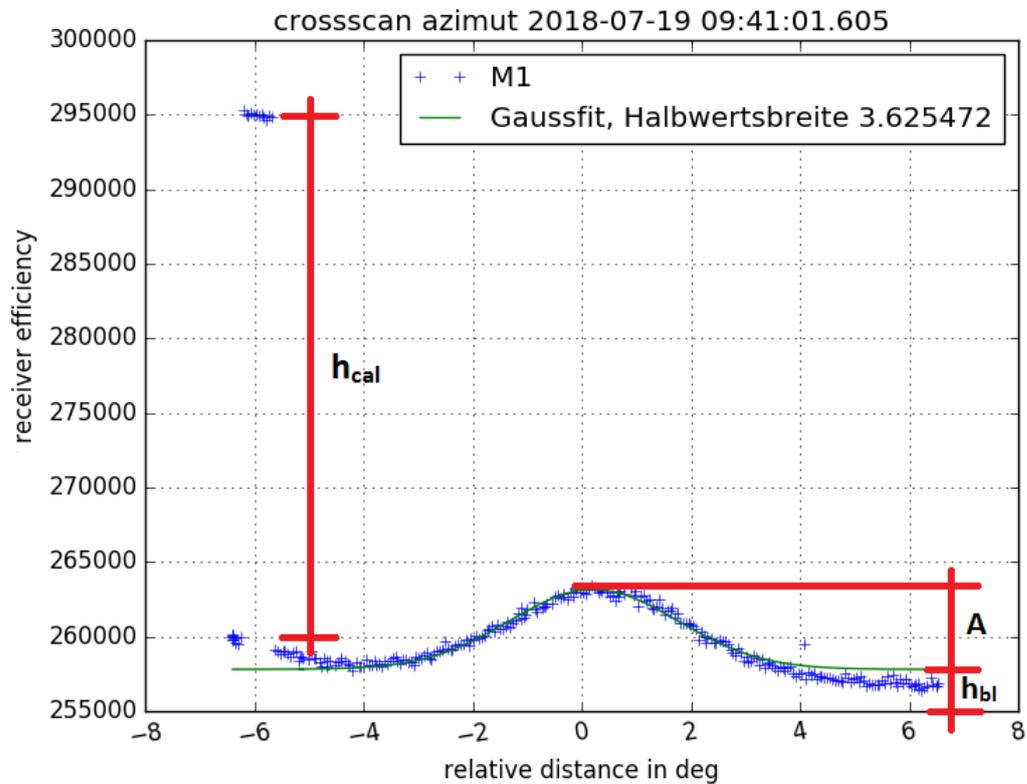


Abbildung 43: Taurus A (Krebsnebel)

Quelle:	Taurus A (M1 bzw. Krebsnebel)
Cross-Scan über Achse:	Azimut
Halbwertsbreite:	$3,6^\circ$
Baseline:	$h_{bl} = 257807cts$
Amplitude	$A = 5353cts$
Kalibration:	$h_{cal} = (295300 - 258694)cts = 36606cts$
Kalibrationstemperatur:	$T_{cal} = 0,000\,546\text{ K}/cts$
Systemtemperatur:	$T_{sys} = 141,3\text{ K}$
Antennentemperatur:	$T_A = 2,9\text{ K}$
Flussdichte:	$S = 812,4\text{ Jy}$

Tabelle 5: Parameter des gemessenen Objekts Taurus A (Krebsnebel)

Die selben Messungen wurden nochmals an anderen Quellen, die im L-Band sichtbar sein müssten, durchgeführt.⁸¹ Die wichtigsten Parameter sind in Tabelle 6 dargestellt.

⁸¹Messungen mit Parameter und Diagramm siehe Anhang A.3

Quelle:	Taurus A	Sonne	Sonne	Mond	Mond	Cassiopeia A	Cygnus A
Cross-Scan über Achse:	Azimut	Azimut	Elevation	Azimut	Elevation	Elevation	Azimut
Halbwertsbreite:	3,6°	4,0°	3,9°	3,4°	3,2°	3,3°	4,2°
Flussdichte S/Jy:	812,4	423963,7	476848	874,2	906,5	1766,7	1652,8

Tabelle 6: Parameter der gemessenen Objekte

Über die ermittelte Flussdichte kann ein Vergleich der Messwerte des 4-Meter-Radioteleskops zu anderen Teleskopen gezogen werden. In Abbildung 44 sind starke Radioquellen mit dem Strahlungsfluss über die Frequenz eingetragen. Der gemessene Strahlungsfluss des 4-Meter-Radioteleskops mit 21 cm Wellenlänge wurde in rot eingezeichnet. Wie zu erkennen, liegen die gemessenen Werte nahe an dem vorgegebenen Strahlungsfluss. Da, wie beispielsweise in Abbildung 43 ersichtlich, ein Pointing-Offset besteht, wird die Quelle nicht mit dem Maximum des Beams getroffen und damit verringert sich die Amplitude. Es müssten mehrere Messreihen aufgenommen und in Azimut und Elevation über die Objekte durchgeführt werden, woraus resultierend der Pointing-Offset minimiert wird. Zudem ist eine Fehlerrechnung zur Bestimmung der Messabweichung notwendig. An dieser Stelle war es jedoch zunächst von Interesse, ob die Messwerte plausibel und die Objekte mit dem Radioteleskop detektierbar sind. Die wichtigste Erkenntnis ist, dass die erwarteten Quellen im L-Band sichtbar sind. Der Parabolspiegel ist exakt genug, um diese Quellen zu messen und die Positionierung ist hinreichend genau, damit sich das Teleskop auf die gewünschte Position ausrichtet. Dies bietet die Möglichkeit, das Pointing-Modell aus Kapitel 5 weiter zu verfeinern und über einen großen Winkelbereich am Horizont zu bestimmen. Betrachtet man die Halbwertsbreite der Sonne in Azimut und Elevation sowie die des Mondes, sind die jeweiligen Halbwertsbreiten innerhalb der 10%-Unsicherheit.⁸² Somit liegen die Messreihen innerhalb der gewünschten Grenzwerte.

⁸²Erfahrungswert der Max-Planck-Instituts für Radioastronomie; Dr. Hafok

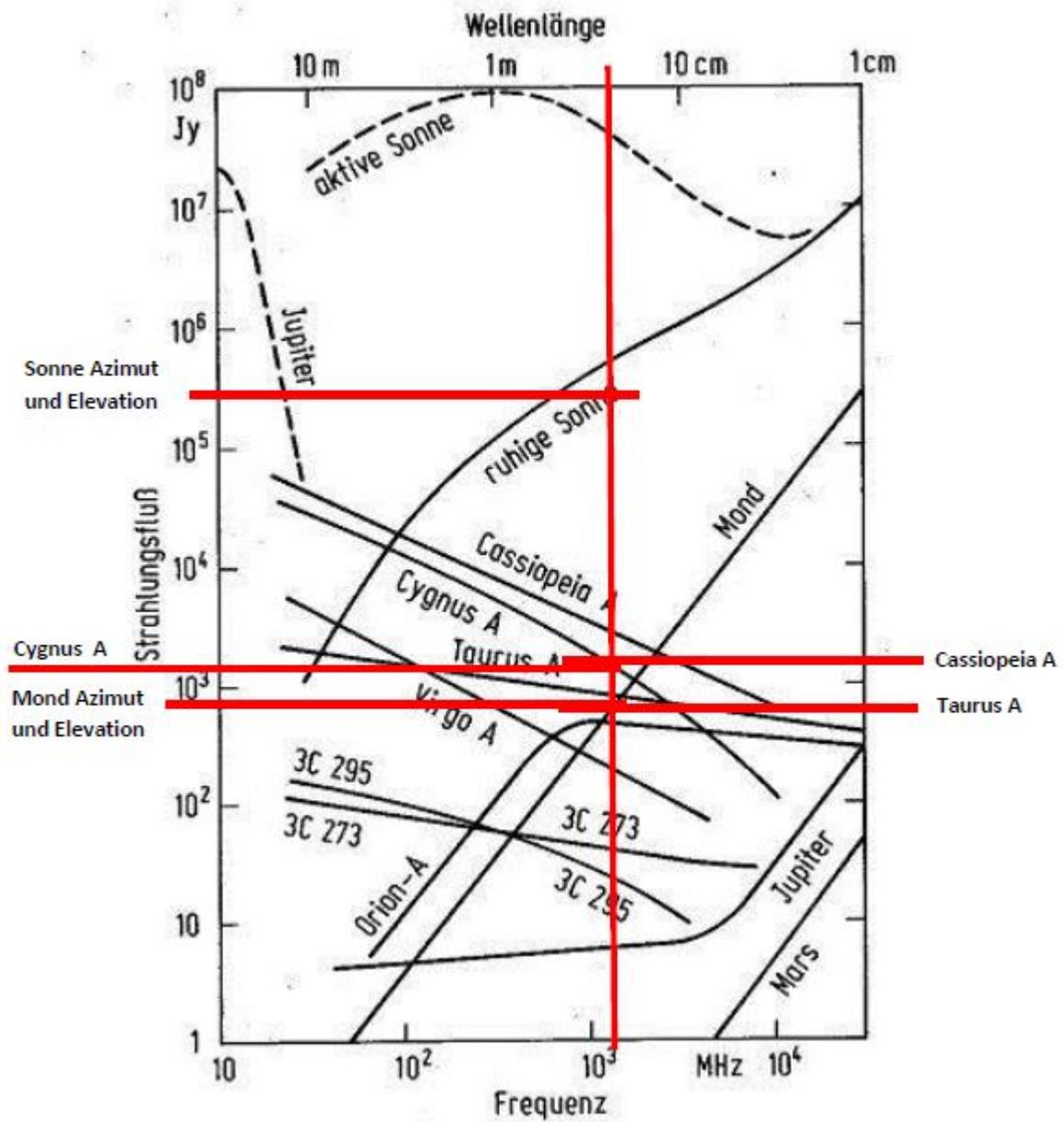


Abbildung 44: Starke Radioquellen: Strahlungsfluss in Abhängigkeit von der Frequenz der Strahlung. Rot markiert ist der gemessene Strahlungsfluss mit dem 4-Meter-Radioteleskop.⁸³

⁸³Hamm [2006]

7 Ergebnis

7.1 Fazit

Im Rahmen dieser Arbeit erfolgte eine Thematisierung des Entwurfs und der Implementierung eines Linux-basierten Kontrollsystems für ein 4-Meter-Radioteleskop. Darunter fielen die Aspekte der Entwicklung eines User Interfaces sowie die Implementierung eines Kontrollsystems, welches Benutzereingaben abstrahiert und als Resultat das Radioteleskop ausrichtet bzw. eine Beobachtung durchführt. Außerdem wurde die Einbindung der beiden Backends behandelt und wie eine langfristige Speicherung der Messdaten in einem Rohdatenformat zu realisieren ist. Die Überprüfung der Softwareimplementierung geschah auf Basis radioastronomischer Messungen. Des Weiteren erfolgte ein Vergleich der mechanischen und radioastronomischen Teleskopcharakteristik, was zur Erstellung eines Pointing-Modells führte.

Das radioastronomische Kontrollprogramm basiert auf dem Linux Betriebssystem Ubuntu, welches in die vier primären Aufgabenbereiche: User Interface, Kontrollprogramm, Aktoren- sowie Backendkommunikation aufgeteilt ist.

Ein skriptbasiertes User Interface bietet dem Benutzer die Möglichkeit, das Teleskop kommandobasiert zu parametrisieren. Mit dem positiven Aspekt, dass diese Aufträge in kleine Programme eingebunden werden können und beispielsweise Objekte über einen längeren Zeitraum zu beobachten bzw. astronomische Messaufträge zu wiederholen sind. Es wurde großen Wert auf eine intuitive Bedienung des Teleskops gelegt. Unter Zuhilfenahme der vier Hauptfunktionen *set*, *get*, *send()* und *help()* kann das System gesteuert werden. Außerdem bietet das Interface die Möglichkeit die Backends sowie die Empfängereinheit zu konfigurieren. Dem Beobachter ist die Option geboten, Positionen in den drei thematisierten horizontalen, äquatorialen und galaktischen Koordinatensystemen zu parametrisieren sowie verschiedene Messmodi zu kommandieren.

Abgeleitet von der Benutzereingabe bestimmt das Kontrollprogramm, auf welches Koordinatensystem und Äquinoktium sich bezogen wird und berechnet die Positionen in das horizontale Koordinatensystem des Radioteleskops. Der Algorithmus wurde so implementiert, dass das System in Folge der Koordinatensystemtransformation ausgerichtet wird und nach Erreichen der Soll-Position mit der radioastronomischen Messung beginnt. Je nach Messauftrag führt das Teleskop eine Rotationsbewegung über das Objekt durch

bzw. betrachtet eine Position im Kosmos. Das Programm ist so entworfen, dass dieses die Bewegung der Erde kompensiert und infolgedessen auf den gewünschten Punkt im Kosmos ausgerichtet bleibt.

Beim Erreichen der Soll-Position wird signalisiert, dass mit der radioastronomischen Messung begonnen werden kann. Für diesen Zweck sind das Kontinuum-Backend sowie das Spektrometer eingebunden. Sie bieten die Möglichkeit, radioastronomische Messungen im Zeit- bzw. im Frequenzbereich durchzuführen. Die Taktung beider Backends sind mit der Empfangseinheit synchronisiert. Es besteht die Option, Beobachtungen im Zeit- bzw. Frequenzbereich einzeln oder gleichzeitig durchzuführen.

Die Interprozesskommunikation ist netzwerkbasierend realisiert. Infolge der Koordinatensystemtransformation des Kontrollprogramms, sendet dieses die Daten über das Netzwerkprotokoll UDP an das Steuerungsprogramm, welches der Aktorkommunikation dient. Das Steuerungsprogramm ist in der Hochsprache C entwickelt und stellt unter Zuhilfenahme einer, vom Hersteller der Hardware, bereitgestellten Bibliothek die Schnittstelle zwischen der Soft- und Hardware her. Daraus resultierend werden die beiden Motoren in Azimut und Elevation auf die gewünschte Position ausgerichtet. Die Messparameter, welche u.a. Beobachtungsdauer und Backendkonfigurationen bereitstellen sowie die Teleskopparameter mit u.a. der aktuellen Position und Geschwindigkeit, werden in zwei Multicast Datenströmen im Netzwerk verteilt. Somit kann jeder innerhalb des Institutsnetzwerks diese betrachten.

Infolge der großen Datenmengen, welche während einer Beobachtung aufgenommen werden, seien es die radioastronomischen Messwerte, welche die Backends zur Verfügung stellen oder die Teleskopparameter, wie beispielsweise die aktuelle Position oder Rotationsgeschwindigkeit, war es notwendig ein Rohdatenformat zu entwickeln. Die Kombination der Datenserialisierung im JSON-Format und partiell unter Numpy als mehrdimensionales Array, stellt eine optimale Ausnutzung dar. Im JSON-Format gespeicherte Daten können über einen Algorithmus oder durch menschliche Betrachtung evaluiert werden. Die großen Datenmengen der Backends, welche lediglich unter Zuhilfenahme von mathematischen Operationen oder infolge von Visualisierungen auswertbar sind, können unter Numpy möglichst effektiv deserialisiert und verarbeitet werden.

Im Anschluss an die Softwareimplementierung folgte die Evaluierung des Algorithmus. Es kann die Aussage getroffen werden, dass das Kontrollsystem problemlos funktioniert. Die

Benutzereingabe wird abstrahiert und anhand einer Koordinatensystemtransformation das Radioteleskop auf die gewünschte Position ausgerichtet bzw. der Messmodus ausgeführt. Außerdem startet und stoppt die Messung zum richtigen Zeitpunkt und speichert die gewünschten Parameter im Rohdatenformat. Auf jede Messdatenaufnahme folgt eine Visualisierung. Die in der Theorie ermittelte Halbwertsbreite eines Cross-Scans in der Elevations- sowie der Azimutachse ist stimmig. Außerdem wurde die Anbindung des Spektrometers getestet. Infolge dieser Erprobung konnte die Spektrallinie des atomaren Wasserstoffs in der Milchstraße über den Spinflip-Übergang nachgewiesen werden. Resultierend daraus ist die Aussage zu treffen, dass die Einbindung des Spektrometers erfolgreich ist sowie der Nachweis, dass sich vermehrt atomarer Wasserstoff in der Milchstraße befindet.

Im Laufe dieser Arbeit konnte beobachtet werden, dass die mechanischen und radioastronomischen Teleskopcharakteristiken nicht korrelieren. Infolge einer Messdatenaufnahme über einen Winkelwertempfänger und die daraus interpolierte Kompensation aus der Projektarbeit⁸⁴ führte zu einer Überkompensation der radioastronomischen Teleskopcharakteristik. Es wurde eine Abhängigkeit der Elevationsachse über den Azimutwinkel festgestellt und infolgedessen ein Pointing-Modell über die Elevationsachse interpoliert. Zudem wurde die Positioniergenauigkeit der Azimutachse betrachtet. Im Winkelbereich der radioastronomischen Messung konnte die Aussage erfolgen, dass die Positioniergenauigkeit hinreichend ist und eine Interpolation keine signifikante Verbesserung bringen würde. Problematisch ist diese Aussage jedoch dahingehend, dass die Bestimmung des Pointing-Modells nur in einem kleinen Winkelbereich stattfand. Um eine qualitative Aussage treffen zu können, muss ein größerer Winkelbereich betrachtet werden.

Erste Tests zur Detektion anderer radioastronomischer Quellen wurden durchgeführt. Es stellte sich heraus, dass die hellsten Objekte im gemessenen Frequenzbereich problemlos sichtbar sind. Somit kann zunächst die Aussage getroffen werden, da die Detektion erstmals in Vordergrund steht, dass das Radioteleskop eine hinreichende Positioniergenauigkeit besitzt. Da infolgedessen ein Beobachten dieser Objekte einen großen Winkelbereich abdeckt und aus diesem Grund das Pointing-Modell in einem großen Winkelbereich erweiterbar ist. Außerdem wurden die Empfangsleistung in eine physikalische Einheit umgewandelt. Die berechnete Flussdichte liegt im erwarteten Bereich, somit sind alle Annahmen zur Kalibration konsistent. Für eine exakte Bestimmung müssen die Beobachtungen nochmals erfolgen, der Pointing-Offset einer additiven Korrektur unterzogen und der Cross-Scan in Azimut und Elevation erneut ohne Pointing-Offset durchgeführt werden.

⁸⁴vgl. Schweikert [2017]

7.2 Ausblick

Nachdem die Funktionsfähigkeit des Teleskops nachgewiesen wurde, sind weitere systematische Tests notwendig. Aufgrund der begrenzten Zeit dieser Arbeit und da der Entwurf und die Implementierung des Kontrollsystems des Radioteleskops im Vordergrund stand, konnte das Pointing-Modell nur in einem kleinen Winkelbereich definiert werden, somit kann keine qualitative Aussage über die Positioniergenauigkeit des Teleskops außerhalb dieses Winkelbereichs getroffen werden. Jedoch wurde dargelegt, dass alle wichtigen radioastronomischen Quellen im L-Band mit dem Radioteleskop detektierbar sind. Somit besteht die Möglichkeit, das Pointing-Modell in einem größeren Bereich zu definieren. Zudem wäre das Hystereseverhalten des Teleskops in Betracht zu ziehen. Dieses kann infolge von Cross-Scans geschehen, welche ein Objekt nacheinander von beiden Seiten überfährt und den Pointing-Offset vergleicht. Daraus resultierend kann im Azimut das Zahnflankenspiel sowie in der Elevationsachse die Ungenauigkeit des Kugelgewindetribs bestimmt und kompensiert werden. Wie in dieser Arbeit aufgezeigt, würden Langzeitmessungen über die Sonne einen großen Rotationsbereich abdecken. Zusätzlich sind über diese Messungen nicht nur die Optimierung der Positionsgenauigkeit des Teleskops möglich. Ebenso kann auch eine Bestimmung der Sonnenaktivität, im 21 cm Frequenzbereich, über einen längeren Zeitraum erfolgen und infolgedessen ist eine Eruiierung über die Nachweisbarkeit von Sonnenaktivitäten, mit dem 4-Meter-Radioteleskop, möglich. Damit das Teleskop möglichst effektiv ausgenutzt wird, wäre eine Überlegung dahingehend, ob ein Algorithmus implementiert wird, welcher überprüft, ob Quellen im Sichtbereich des Radioteleskops stehen und daraufhin automatisiert Messungen durchführt, annehmbar.

Außerdem besteht die Möglichkeit, die Rauschdiode, im Falle eines Cross-Scans, automatisch vor Überfahren des Objekts zu schalten. Infolgedessen müsste der Benutzer dieses nicht manuell durchführen bzw. über ein kleines Programm im User Interface implementieren.

Vor dem Hintergrund, dass die beiden Positionsbestimmungsmethoden der mechanischen und der radioastronomischen Teleskopcharakteristik nicht korrelieren, wäre eine Betrachtung dieses Effekts interessant. Es wäre in Betracht zu ziehen, ob der Parabolspiegel infolge einer Kippung in der Elevationsachse verformt wird. Zum Eruiieren des Effekts wäre die Erstellung eines Finite-Elemente-Modells des 4-Meter-Radioteleskops möglich.

Literatur

Amt für Umweltschutz - Abteilung Stadtklimatologie 2018

AMT FÜR UMWELTSCHUTZ - ABTEILUNG STADTKLIMATOLOGIE: *Sonnenstand*. https://www.stadtklima-stuttgart.de/index.php?klima_sonnenstand. Version: 2018, Abruf: 18.07.2018

Chéreau 2017

CHÉREAU, Fabien: *Astronomieprogramm unter GNU General Public License zur Simulation eines Planetariums*. 2017. – Stellarium 0.16.1

Hamm 2006

HAMM, Michael: *Die stärksten Radioquellen am Himmel*. https://www.mpifr-bonn.mpg.de/412925/michael_hamm. Version: 2006, Abruf: 17.07.2018

Kalvius 2013

KALVIUS, Georg M.: *Physik der Atome, Moleküle und Kerne - Wärmestatistik*. Auflage 5. Oldenbourg : Oldenbourg Wissenschaftsverlag GmbH, 2013. – ISBN 978–3–486–59898–8

Keller 2016

KELLER, Hans-Ulrich: *Kompendium der Astronomie*. Auflage 5. Stuttgart : Franckh Kosmos Verlag, 2016. – ISBN 978–3440148174

Keller 2007

KELLER, R.: *Kalibration*. <http://www3.mpifr-bonn.mpg.de/div/electronic/index.html>. Version: 2007, Abruf: 07.09.2018

Kraus 1997

KRAUS, A.: *Kurzzeitvariabilität aktiver Galaxienkerne in totaler und polarisierter Flußdichte*. Bonn : Max-Planck-Institut für Radioastronomie, 1997. – Inaugural-Dissertation

Kraus 2018

KRAUS, A.: *Calibration of the Effelsberg 100m telescope*. Max-Planck-Institut für Radioastronomie, 2018. – Interner Bericht

Kraus 1986

KRAUS, J.D.: *Radio Astronomy*. Auflage 2. Powell, Ohio : Cygnus-Quasar Books, 1986. – ISBN 1–882484–00–2

Kraus 1988

KRAUS, J.D.: *Antennas*. Auflage 2. Ohio State University : McGraw-Hill Book Company, 1988. – ISBN 0-07-0354 22-7

Max-Planck-Institut für Radioastronomie 2018

MAX-PLANCK-INSTITUT FÜR RADIOASTRONOMIE: *Taurus A*. <https://www.mpifr-bonn.mpg.de/525761/taurus-a>. Version: 2018, Abruf: 23.07.2018

Montenbruck 2001

MONTENBRUCK, Oliver: *Grundlagen der Ephemeridenrechnung*. Auflage 6. Heidelberg : Verlag Sterne und Weltraum, 2001. – ISBN 3-87973-941-2

Schweikert 2016

SCHWEIKERT, Sebastian: *Automatisierung eines 4-Meter-Radioteleskops im Azimut, basierend auf dem Linux Betriebssystem Ubuntu*. Effelsberg : Hochschule Koblenz in Kooperation mit Max-Planck-Institut für Radioastronomie, 2016. – Bachelorarbeit

Schweikert 2017

SCHWEIKERT, Sebastian: *Einbau, Justage und Inbetriebnahme eines L-Band Empfangssystems in ein 4-Meter-Radioteleskop*. Effelsberg : Hochschule Koblenz in Kooperation mit Max-Planck-Institut für Radioastronomie, 2017. – Projektarbeit

Sternwarte-Singen 2018

STERNWARTE-SINGEN: *praezession astronomisch*. http://www.sternwarte-singen.de/images/praezession_astronomisch_grafik900x680_vss2013.gif. Version: 2018, Abruf: 26.07.2018

United States Naval Observatory 2018

UNITED STATES NAVAL OBSERVATORY: *International Celestial Reference System*. http://aa.usno.navy.mil/faq/docs/ICRS_doc.php. Version: 2018, Abruf: 11.07.2018

Unsöld u. Baschek 2005

UNSÖLD, Albrecht ; BASCHEK, Bodo: *Der neue Kosmos*. Auflage 7. Leipzig : Springer Verlag, 2005. – ISBN 978-3-662-06530-3

Wikipedia 2016

WIKIPEDIA: *Meridian auf der Himmelskugel für einen Beobachter auf der Nordhalbkugel und geozentrischer Horizont (parallel zum Horizont des Beobachters)*. <https://upload.wikimedia.org/wikipedia/commons/8/84/MeridianGeozentrisch.svg>. Version: 2016, Abruf: 09.09.2018

A Anhang

A.1 Funktionen des User Interfaces

- help()
 - aufrufen aller Funktionen
- send()
 - übertragen der Parameter an das Kontrollprogramm
- start_messung()
 - übertragen der Parameter an das Kontrollprogramm und setzen der Parameterübergabe auf True
- Backendkonfiguration
 - pbebackend(*inputs)
 - * Konfiguration Kontinuum-Backend
 - fitwriter(*inputs)
 - * Konfiguration Spektrometer
- Parameter auslesen
 - get_position()
 - * lese aktuelle Azimut- und Elevationsposition
 - get_state_axis()
 - * lese Zustand der beiden Achsen Azimut und Elevation
 - get_status()
 - * ausgeben ob Teleskop für Parameterübergabe bereit ist oder nicht
 - get_actual_velocity()
 - * ausgeben der aktuellen Geschwindigkeit
 - get_actual_acceleration()
 - * ausgeben der aktuellen Beschleunigung
 - get_actual_torque()

- * ausgeben der aktuellen Drehmomente
- `get_position_limit()`
 - * ausgeben der implementierten Endlagen von Azimut und Elevation
- `get_offset()`
 - * ausgeben des parametrisierten Offsets
- astronomische Messmodi
 - `track(messobjekt, integration = None, offsetx = (offx,unitoffx), offsety = (offy,unitoffy))`
 - * verfolgen eines Messobjekts
 - `crossscan(integration, messobjekt = None, az = (posaz,unitaz), el = (posel,unitel), lengthx = (lengthaz,unitlenx), lengthy = (lengthel,unitleny), offsetx = (offx,unitoffx), offsety = (offy,unitoffy))`
 - * durchführen eines Cross-Scans
- Parameter setzen
 - `set_position_azel(integration, el = (posel,unitel), offsety = (offy,unitoffy), az = (posaz,unitaz), offsetx = (offx,unitoffx), velaz = None, velel = None)`
 - * setze Positionen im horizontalen Koordinatensystem
 - `set_position_Ra_Dec(integration = 0, ra = (rec, unitrec), dec = (decl, unitdec), coorsys = 'J2000.0')`
 - * setze Positionen im äquatorialen Koordinatensystem
 - `set_position_galactic(integration = 0, lng = (l, unitl), br = (b, unitb), coorsys = 'J2000.0')`
 - * setze Positionen im galaktischen Koordinatensystem
 - `set_status(status)`
 - * True/False, setzen ob Teleskop Befehle entgegen nimmt oder nicht
 - `set_offset(offaz = None, offel = None)`
 - * setzen der Offsetparameter Azimut und Elevation
 - `set_velocity_az(vel)`
 - * setzen der Rotationsgeschwindigkeit der Azimutachse
 - `set_velocity_el(vel)`

- * setzen der Rotationsgeschwindigkeit der Elevationsachse
- Intercom Parametrisierung
 - setup_tgen(blank = None, sync = None, pha = None)
 - * setzen der Blank, Synczeit und der Phase
 - reset_ioswitches()
 - * setzen der I/O-Schalter am Empfänger
 - cal_on()
 - * einschalten der Kalibration
 - cal_off()
 - * ausschalten der Kalibration
 - ask_intercom(cmd)
 - * parametrisieren des Intercom über SCPI-Befehle

A.2 Programm Interfaces

JSON-Keys	
User Interface:	
command	Freigabe des Teleskops True/False
source	Messobjekt
loc_height	Höhe des Teleskop-Standortes NHN
loc_LAT	Geographische Breite des Teleskop-Standortes
loc_LONG	Geographische Länge des Teleskop-Standortes
system	Koordinatensystem
ra	Rektaszension
decl	Deklination
unitra	Einheit Rektaszension
unitdec	Einheit Deklination
galacl	Galaktische Länge
galacb	Galaktische Breite
velposaz	Rotationsgeschwindigkeit Azimut
velposel	Rotationsgeschwindigkeit Elevation
SOLL-posaz	Azimut
SOLL-posel	Elevation
coordsys	Äquinoktium
offx	Offset Azimutrotationsachse
offy	Offset Elevationsrotationsachse
unitaz	Einheit Azimut
unitel	Einheit Elevation
unitoffx	Einheit Offset Azimut
unitoffy	Einheit Offset Elevation
crosslengthaz	Länge des Cross-Scans in der Azimutrotationsachse
crosslengthel	Länge des Cross-Scans in der Elevationsrotationsachse
unitcrosslengthx	Einheit des Cross-Scans in der Azimutrotationsachse
unitcrosslengthy	Einheit des Cross-Scans in der Elevationsrotationsachse
pbebackend	Konfigurationen Kontinuum-Backend
fitwriter	Konfigurationen FFTS-Spektrometer
integrationtime	Messdauer

Tabelle 7: JSON-Keys, des User Interface

JSON-Keys Messparameter:	
userparameter	Backendkonfiguration: Kontium-Backend und Spektrometer
referenceval	aktuelle Position des zu messenden Objekts
measuretime	aktuelle Messdauer
crosslenaz	Länge des Cross-Scans im Azimutwinkel
crosslenel	Länge des Cross-Scans im Elevationwinkel
timestampsavedata	Zeitmarke welche als Speichername dient
window	gibt an ob sich das Teleskop auf der Soll-Position befindet
startpbebackend	Signal zum Starten und Stoppen der Backends

Tabelle 8: JSON-Keys, welche über den Multicast als Messparameter verteilt werden

JSON-Keys regler.c:	
time	aktuelle Zeitmarke
posaz	aktuelle Azimutposition in Grad
posel	aktuelle Elevationsposition in mm
state	Zustand der Regeleinheit Azimut
stateel	Zustand der Regeleinheit Elevation
torqueaz	aktuelles Drehmoment der Azimutachse in Nm
torqueel	aktuelles Drehmoment der Elevationsachse in Nm
poslimitnegaz	Software Positionsendlage negativ Azimut
poslimitposaz	Software Positionsendlage positiv Azimut
poslimitnegel	Software Positionsendlage Minimum Elevation
poslimitposel	Software Positionsendlage Maximum Elevation
velocityaz	aktuelle Rotationsgeschwindigkeit Azimut in 1/min
velocityel	aktuelle Rotationsgeschwindigkeit Elevation mm/min
accelerationel	aktuelle Rotationsbeschleunigung Azimut in 1/sec ²
accelerationaz	aktuelle Rotationsgeschwindigkeit Elevation mm/min ²

Tabelle 9: JSON-Keys, welche über den Multicast aus dem Programm regler.c übertragen werden

UDP Parameter:	physikalische Einheiten
Starten und Stoppen der Parametervorgabe an die Regeleinheit	True/ False
Position Elevation	mm
Geschwindigkeit Elevation	mm/min
Position Azimut	deg
Rotationsgeschwindigkeit Azimut	1/min

Tabelle 10: Parameter, welche von scriptcontrol.py an das Programm regler.c übertragen werden.

JSON-Keys des Rohdatenformats:	
backendsection1	Backenddaten, mit vertikaler Polarisation und Zeitmarke
backendsection2	Backenddaten, mit horizontale Polarisation und Zeitmarke
multicposaz	Azimutwinkel während der Messung in Grad
multicposelev	Elevationswinkel während der Messung in Grad
multicvelaz	Rotationsgeschwindigkeit Azimut in 1/min
multicvelev	Rotationsgeschwindigkeit Azimut in 1/min
multictimestamp	Zeitmarke des Multicast
frequency	nur bei FFTS-Messung, Frequenzen während der Messung
object	welches Objekt beobachtet wurde
crosslenaz	Breite des Cross-Scans in Azimut in Grad
crosslenel	Breite des Cross-Scans in Elevation in Grad
cross	relativer Winkel des Cross-Scans
option	Messmodi
messung	Koordinatensystem

Tabelle 11: JSON-Keys des Rohdatenformats

A.3 Radioastronomische Messungen

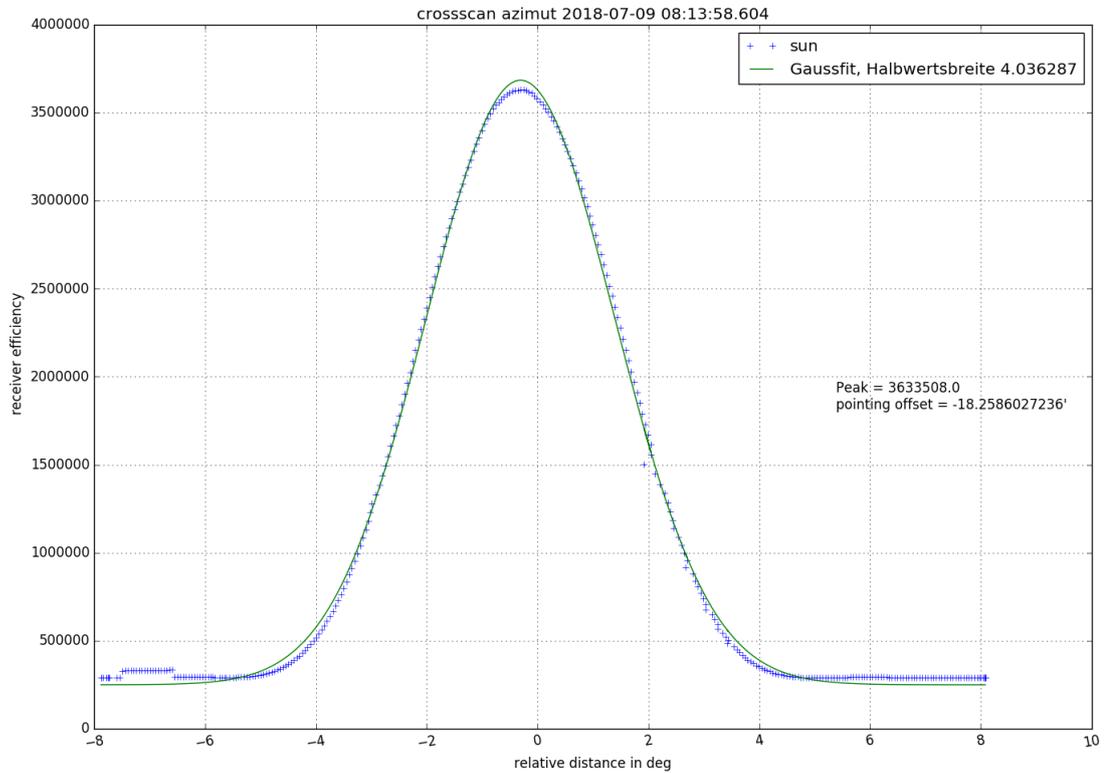


Abbildung 45: Cross-Scan der Sonne in Azimut. Vor überqueren der Quelle wurde die Rauschdiode ein- und ausgeschaltet.

Quelle:	Sonne
Cross-Scan über Achse:	Azimut
Halbwertsbreite:	$4,0^\circ$
Baseline:	$h_{bl} = 249076cts$
Amplitude	$A = 3435532cts$
Kalibration:	$h_{cal} = (333491 - 288472)cts = 45019cts$
Kalibrationstemperatur:	$T_{cal} = 4,4 \cdot 10^{-4} K/cts$
Systemtemperatur:	$T_{sys} = 110,7 K$
Antennentemperatur:	$T_A = 1526,3 K$
Flussdichte:	$S = 423\,963,7 Jy$

Tabelle 12: Messwerte einer Sonnenmessung im Azimut

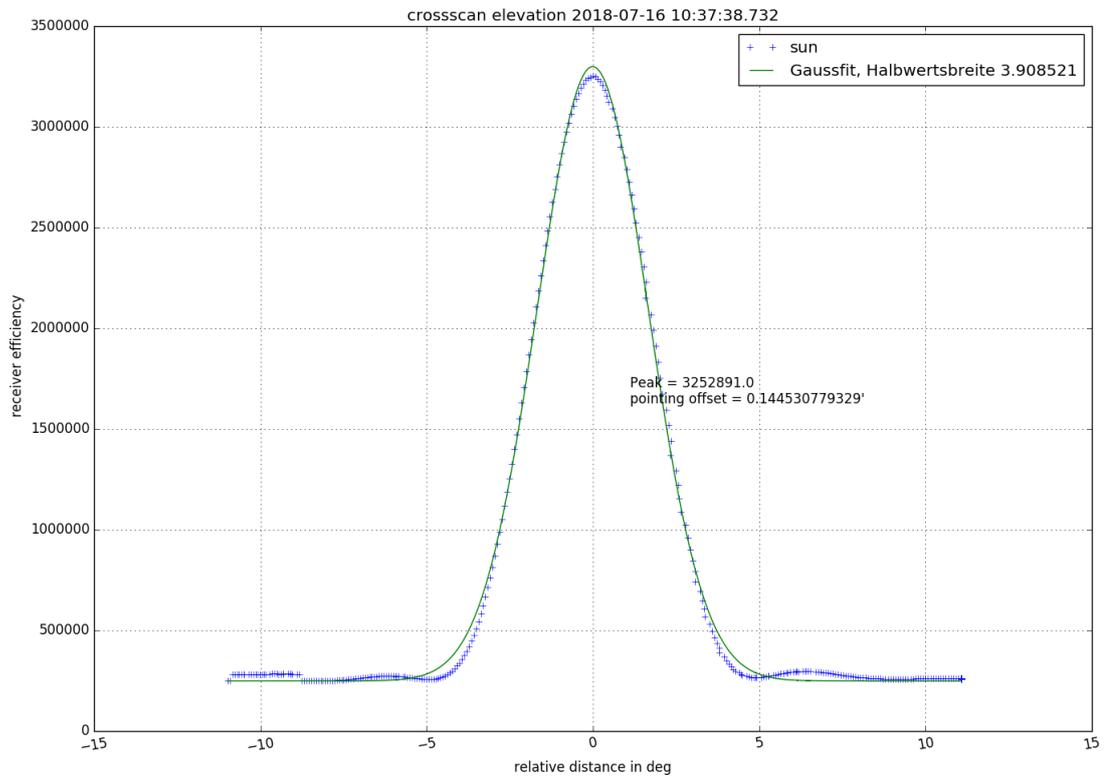


Abbildung 46: Elevations-Cross-Scan der Sonne

Quelle:	Sonne
Cross-Scan über Achse:	Elevation
Halbwertsbreite:	$3,9^\circ$
Baseline:	$h_{bl} = 247325cts$
Amplitude	$A = 3051266cts$
Kalibration:	$h_{cal} = (283161 - 247612)cts = 35549cts$
Kalibrationstemperatur:	$T_{cal} = 5,63 \cdot 10^{-4} \text{ K/cts}$
Systemtemperatur:	$T_{sys} = 139,3 \text{ K}$
Antennentemperatur:	$T_A = 1717 \text{ K}$
Flussdichte:	$S = 476\,848 \text{ Jy}$

Tabelle 13: Messwerte einer Sonnenmessung in Elevation

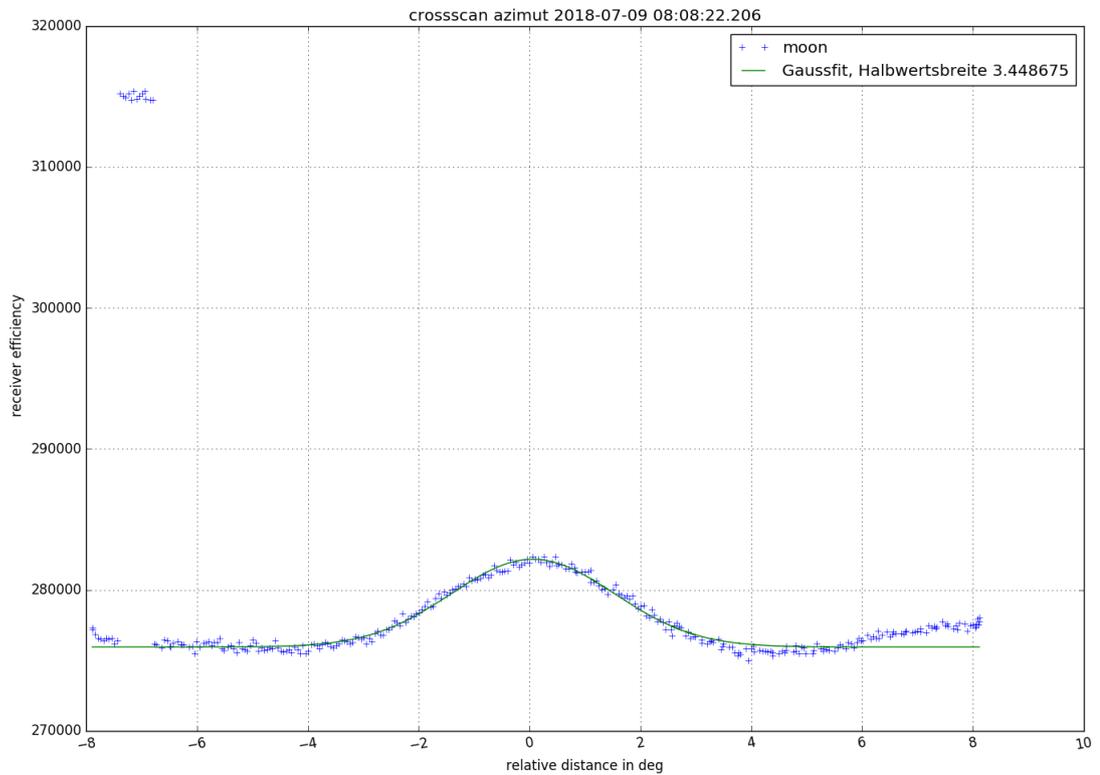


Abbildung 47: Azimut-Cross-Scan des Mondes

Quelle:	Mond
Cross-Scan über Achse:	Azimut
Halbwertsbreite:	$3,4^\circ$
Baseline:	$h_{bl} = 275953cts$
Amplitude	$A = 6220cts$
Kalibration:	$h_{cal} = (315405 - 275879)cts = 39526cts$
Kalibrationstemperatur:	$T_{cal} = 5,06 \cdot 10^{-4} \text{ K/cts}$
Systemtemperatur:	$T_{sys} = 139,6 \text{ K}$
Antennentemperatur:	$T_A = 3,1 \text{ K}$
Flussdichte:	$S = 874,2 \text{ Jy}$

Tabelle 14: Messwerte einer Mondmessung im Azimut

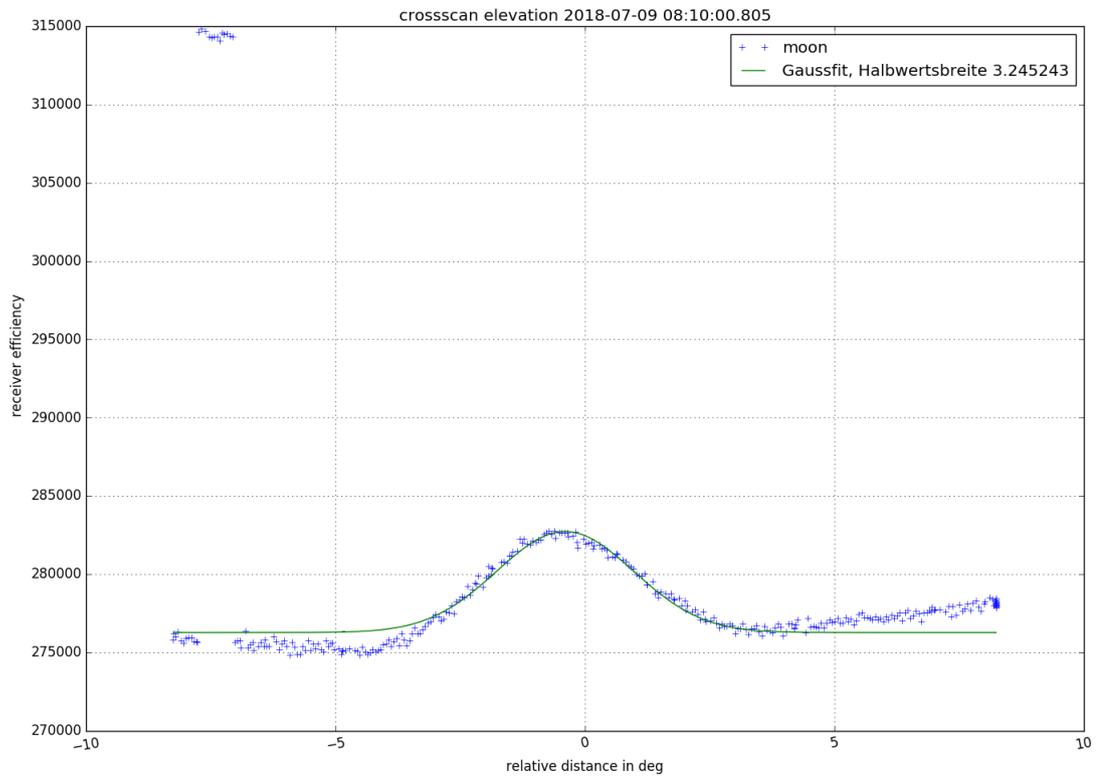


Abbildung 48: Elevations-Cross-Scan des Mondes

Quelle:	Mond
Cross-Scan über Achse:	Elevation
Halbwertsbreite:	$3,2^\circ$
Baseline:	$h_{bl} = 276277cts$
Amplitude	$A = 6451cts$
Kalibration:	$h_{cal} = (314855 - 275321)cts = 39534cts$
Kalibrationstemperatur:	$T_{cal} = 5,06 \cdot 10^{-4} \text{ K/cts}$
Systemtemperatur:	$T_{sys} = 139,8 \text{ K}$
Antennentemperatur:	$T_A = 3,3 \text{ K}$
Flussdichte:	$S = 906,5 \text{ Jy}$

Tabelle 15: Messwerte einer Mondmessung in Elevation

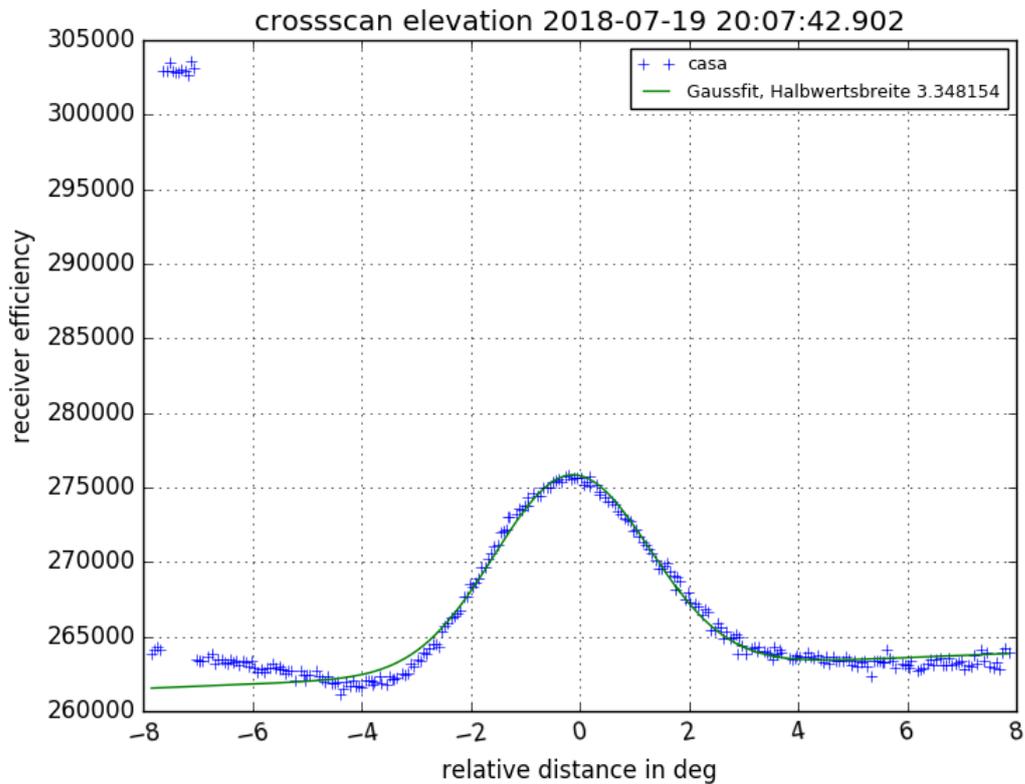


Abbildung 49: Elevations-Cross-Scan der Supernova Cassiopeia A

Quelle:	Cassiopeia A
Cross-Scan über Achse:	Elevation
Halbwertsbreite:	$3,3^\circ$
Baseline:	$h_{bl} = 263421\text{cts}$
Amplitude	$A = 12369\text{cts}$
Kalibration:	$h_{cal} = (30295 - 263948)\text{cts} = 38948\text{cts}$
Kalibrationstemperatur:	$T_{cal} = 0,018\,832\text{ K/cts}$
Systemtemperatur:	$T_{sys} = 5319,83\text{ K}$
Antennentemperatur:	$T_A = 5,15 \cdot 10^{-4}\text{ K}$
Flussdichte:	$S = 1766,7\text{ Jy}$

Tabelle 16: Messwerte der Supernova Cassiopeia A

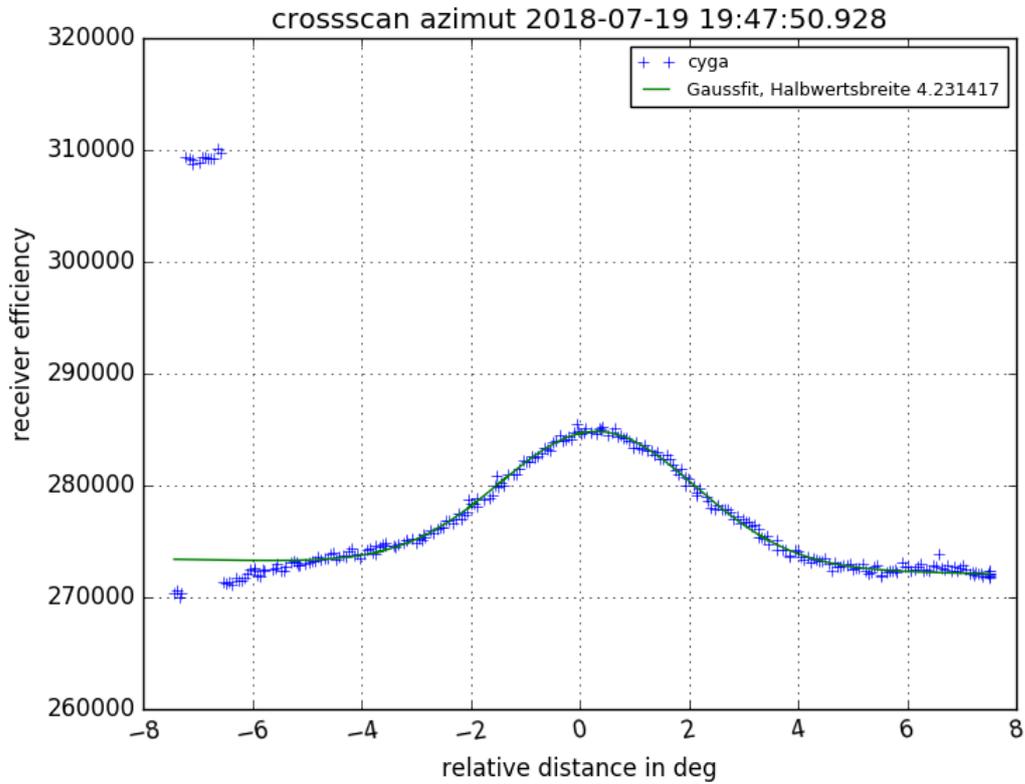


Abbildung 50: Azimut-Cross-Scan der Cygnus A

Quelle:	Cygnus A
Cross-Scan über Achse:	Azimut
Halbwertsbreite:	$4,2^\circ$
Baseline:	$h_{bl} = 273448\text{cts}$
Amplitude	$A = 11380\text{cts}$
Kalibration:	$h_{cal} = (309310 - 271035)\text{cts} = 38276\text{cts}$
Kalibrationstemperatur:	$T_{cal} = 5,23 \cdot 10^{-4} \text{ K/cts}$
Systemtemperatur:	$T_{sys} = 142,9 \text{ K}$
Antennentemperatur:	$T_A = 6,0 \text{ K}$
Flussdichte:	$S = 1652,8 \text{ Jy}$

Tabelle 17: Messwerte Cygnus A