

# **Inbetriebnahme einer freien Software zur Satellitenbahnvorhersage und Ansteuerung einer Hochleistungsantenne**

**STUDIENARBEIT  
über das dritte Studienjahr**

im Studiengang  
Elektrotechnik, Nachrichten- und Kommunikationstechnik

an der DHBW Ravensburg  
Campus Friedrichshafen

von

Sarah Brückner,  
Maximilian Stiefel und  
Hannes Bohnengel

15. Juli 2016

Bearbeitungszeitraum:	Oktober - Dezember 2015
	April - Juni 2016
Betreuer:	Dipl.-Ing. (DH) Hardy Lau

# Kurzfassung

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean porttitor mi purus, sit amet efficitur velit semper sit amet. Mauris et pulvinar nunc, id maximus metus. Suspendisse convallis sapien nisi, sed maximus quam gravida eu. Duis faucibus elit non nunc posuere dignissim. Sed ullamcorper fringilla felis ac mollis. Nunc blandit tristique auctor. Praesent elementum dictum nulla et vulputate. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Integer sed rutrum lacus. Morbi tincidunt dui at augue molestie rhoncus. Curabitur sagittis sed sapien quis vestibulum. Cras dictum sem quam, nec tincidunt augue tempor et. Morbi consectetur, dui id scelerisque consequat, velit tortor gravida eros, ac hendrerit massa magna sit amet lacus. Quisque mattis nulla diam, sed efficitur mi fermentum vitae. Nam vestibulum iaculis rhoncus.

## Abstract

  Translation of „Kurzfassung“ comes here...

# Erklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2015.

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema:

**Inbetriebnahme einer freien Software zur  
Satellitenbahnvorhersage und Ansteuerung einer  
Hochleistungsantenne**

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Friedrichshafen, den 12. Juli 2016

---

Sarah Brückner

---

Maximilian Stiefel

---

Hannes Bohnengel

# Inhaltsverzeichnis

<b>Formelgrößen und Einheiten</b> . . . . .	<b>I</b>
<b>Abkürzungen</b> . . . . .	<b>II</b>
<b>1 Einleitung</b> . . . . .	<b>1</b>
<b>2 Projektmanagement</b> . . . . .	<b>3</b>
2.1 Zeitplan . . . . .	4
2.2 Anforderungsdefinition . . . . .	4
2.3 Arbeitspakte . . . . .	5
2.4 Gantt-Diagramm . . . . .	7
<b>3 Einführung</b> . . . . .	<b>9</b>
3.1 Amateurfunksatelliten . . . . .	9
3.2 Aufbau der Bodenstation . . . . .	13
3.2.1 Antennen . . . . .	13
3.2.2 Rotoren . . . . .	14
3.2.3 Funkgerät ICOM IC-9100 . . . . .	15
<b>4 Hintergründe</b> . . . . .	<b>16</b>
4.1 Bahnmechanik . . . . .	16
4.1.1 Die Keplerschen Gesetze . . . . .	16
4.1.2 Die Bahnelemente . . . . .	25
4.1.3 Von den Bahnelementen zum Vektor . . . . .	33
4.1.4 Bahnmodelle . . . . .	37
4.2 Doppler-Effekt . . . . .	39
<b>5 GPredict</b> . . . . .	<b>42</b>
5.1 Übersicht . . . . .	42
5.2 Grafische Oberfläche . . . . .	43
5.2.1 Grundansichten . . . . .	44
5.2.2 Weitere Ansichten . . . . .	46

5.2.3 Modul Pop-Up Menü . . . . .	47
5.2.4 GPredict-Einstellungen . . . . .	53
5.3 HamLib-Programmierschnittstelle . . . . .	55
5.3.1 Übersicht . . . . .	55
5.3.2 Parameter-Konfiguration . . . . .	57
5.3.3 Verwendung . . . . .	58
5.4 Inbetriebnahme unter Windows . . . . .	59
5.4.1 Zusätzliche Software . . . . .	59
5.4.2 Test der Bedienfelder . . . . .	60
5.4.3 Test der Kommunikation zum Funkgerät . . . . .	61
5.4.4 Test der Kommunikation zu den Rotoren . . . . .	62
5.4.5 Konfiguration des Funkgeräts . . . . .	63
5.4.6 Konfiguration der Rotoren . . . . .	64
5.4.7 Modifizierung des HamLib-Quellcodes . . . . .	65
5.4.8 Kompilieren der modifizierten HamLib für Windows . . . . .	65
5.4.9 Probelauf und Testergebnisse . . . . .	67
<b>6 Zusammenfassung und Ausblick . . . . .</b>	<b>69</b>
<b>Abbildungsverzeichnis . . . . .</b>	<b>III</b>
<b>Tabellenverzeichnis . . . . .</b>	<b>V</b>
<b>Literatur- und Quellenverzeichnis . . . . .</b>	<b>VI</b>
<b>Anhang . . . . .</b>	<b>VIII</b>

# Formelgrößen und Einheiten

Formelzeichen	Einheit	Abkürzung	Physikalische Größe
$r$	Meter	$m$	Länge
$v = \dot{r}$	Meter pro Sekunde	$\frac{m}{s}$	Geschwindigkeit
$a = \dot{v} = \ddot{r}$	Meter pro Quadratsekunde	$\frac{m}{s^2}$	Beschleunigung
$A$	Quadratmeter	$m^2$	Fläche
$M, m$	Kilogramm	$kg$	Masse
$T, t$	Sekunden	$s$	Zeit

# Abkürzungen

**API** Application Programming Interface

**AOS** Acquisiton of Signal

**BE** Backend

**LOS** Loss of Signal

**GPL** General Public License

**CW** Continious Wave

**SGP4** „Simplified General Pertubations Model No. 4“

**SDP4** „Simplified Deep Space Pertubations Model No. 4“

**NORAD** „North American Aerospace Defense Command“

**TLE** „Two-line Elements“

**API** Application Programming Interface

**AOS** Acquisiton of Signal

**BE** Backend

**CI-V** Communication Interface-V

**GPL** General Public License

**LOS** Loss of Signal

**PTT** Push-to-Talk

**CW** Continious Wave

# 1 Einleitung

In der Vergangenheit wurden teure Satelliten nur zu Forschungs- und Verteidigungszwecken entwickelt und gebaut. Heutzutage nimmt der Geschäftsanteil mit kommerziellen Satelliten stetig zu. Möchte der Fussball-Fan ein Europameisterschaftsspiel in Echtzeit verfolgen können oder der Reisende sich zu seinem Ziel navigieren lassen, so sind Satellitensysteme unverzichtbar.

Eine entscheidende Rolle bei der Satellitenforschung haben die Amateurfunksatelliten gespielt. Neue Techniken konnten ohne ein kommerzielles Geschäft oder eine wissenschaftliche bzw. militärische Mission zu gefährden mit Amateurfunksatelliten getestet werden. Die ständige Beobachtung der Satelliten wurde durch die Vielzahl der Amateurfunker weltweit sichergestellt.

Um Satelliten in der Nachrichtentechnik nutzen zu können, wird eine Bodenstation benötigt. Mit entsprechender Hard- und Software können Satelliten verfolgt werden. Auch eine Kommunikation über und mit Objekten im Orbit ist durch ein entsprechendes Equipment möglich.

Diese Studienarbeit beschreibt die Inbetriebnahme einer freien Software zur Satellitenbahnvorhersage und Ansteuerung einer Hochleistungsantenne. Dies schließt die Nachführung der Antenne und damit die Steuerung von Rotoren mit ein. Die freie Software soll die bestehende Software SatPC32 ersetzen. SatPC32 ist eine proprietäre Software und dient der Bodenstation als Referenz. Da SatPC32 nicht mehr weiterentwickelt wird und es darüber hinaus keinen Support für diese Software gibt, wurde der alternative Weg, einer freien Software für die Bodenstation in Friedrichshafen, gewählt. Diese freie Software ist GPredict. GPredict ist quelloffen und beherrscht es mit Hilfe der „Two-line Elements“ (TLE) die Satellitenposition vorherzusagen. Des Weiteren wird von GPredict bzw. der darunter liegenden HamLib eine Vielzahl verschiedener Hardware-Einheiten zur Ausrichtung der Antenne und zur Kommunikation unterstützt. Die Ausgabe der Satellitenposition auf einer grafischen Oberfläche beherrscht GPredict problemlos. Eine Kommunikation mit Transpondern im Orbit aufzubauen ist mit GPredict kinderleicht.

Diese Studienarbeit beschreibt GPredict und soll dem Leser einen leichten Einstieg in die

Bedienung dieser Software ermöglichen. Außerdem beleuchtet diese Arbeit den physikalischen Hintergrund der Satellitenbahn-Vorhersage und damit einhergehend die Ursache der Dopplerverschiebung der Mittenfrequenz bei der Kommunikation mit einem Satelliten. Ein weiterer inhaltlicher Grundpfeiler dieser Studienarbeit ist die Dokumentation des gesamten Projektprozess.

## 2 Projektmanagement

Schon Thomas Carlyle (1795–1881) erkannte die Wichtigkeit von strukturierten und organisiertem Vorgehen als er sagte:

“Unsere Hauptaufgabe ist nicht, zu erkennen, was unklar in weiter Entfernung liegt, sondern zu tun, was klar vor uns liegt”.

In einem Projekt ist das strukturierte und organisierte Vorgehen der klare Weg zu einem erfolgreichem Ziel. Daher wird sich in dieser Arbeit dem Projektmanagement bedient um die Antennennachführung für Satelliten in die richtige Richtung zu lotzen. Dabei lehnt sich das Management an das V-Modell, welche den Ablauf von Software-, als auch von Hardwareentwicklungsprozessen beschreibt. Dieses Modell soll einem Projekt die Richtung weisen, jedoch werden die einzelnen Schritte vom Projektmanager selbst definiert. Ein Vorgehensmodell wie dieses legt folgende Prozesse fest:

- die Aktivitäten die durchzuführen sind,
- die Reihenfolge des Arbeitsablaufes,
- die Definition von Ergebnissen,
- die Fertigstellungskriterien,
- die Ressourcen die vorhanden sind
- und die anzuwendenden Standards/Werkzeuge.

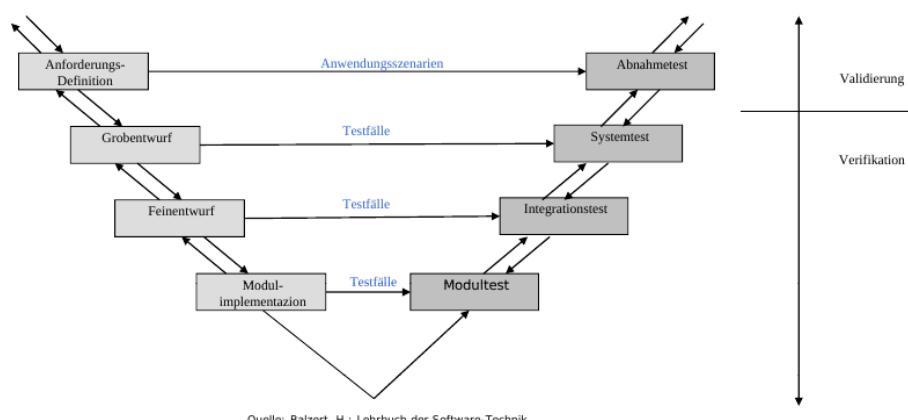


Abbildung 2.1: V-Modell, Quelle: [1]

Eine wichtige Rolle spielt die Qualitätssicherung, die das V-Modell sicher stellt. In diesem Modell sind die Verifikation und die Validation ein fester Bestandteil. Verifikation bedeutet, die Sicherstellung dass das entwickelte Produkt mit den Spezifikationen übereinstimmt. Die Validation ist die Eignung des Produkts bezogen auf seinen Einsatzzweck. Durch die Sicherstellung beider Qualitätsmerkmale wird das Projekt erfolgreich zu seinem Ziel, die Antennennachführung für Satelliten, geführt. Aus diesem Grund ist das V-Modell die richtige Vorangehensweise für dieses Projekt.

## 2.1 Zeitplan

Bevor ein Projekt starten kann, ist wichtig zu definieren, wie viel Zeit für die Durchführung zu Verfügung steht. Für diese Studienarbeit wird drei Studierenden einen Zeitrahmen von zwei Semester zugeschrieben. Beginn der Studienarbeit ist der 14. Oktober 2015 und endet am 15. Juli 2016. Für die Studierenden der Kurse TEN, TEK und TLE ist im Studienplan jeder Mittwoch von 14 Uhr bis 17.15 Uhr Studienarbeitszeit vorgesehen. Dies ist eine grobe Richtlinie für die Durchführung der Studienarbeit und kann nach Bedarf angepasst werden.

## 2.2 Anforderungsdefinition

Inbetriebnahme einer quelloffenen, kostenfreien Software zur Antennennachführung für das Verfolgen von erdnahen Satelliten. Die zur Antennennachführung benötigten Schnittstellen werden von der Hochschule zur Verfügung gestellt und sollen für diese Arbeit verwendet werden. Die Software soll die Satellitenposition mit Hilfe der Keplerelemente berechnen und ausgeben. Die Ausgabe soll grafisch erfolgen und anzeigen, wann der gewählte Satellit verfügbar ist. Ist der gewählte Satellit verfügbar, soll die Antenne in die richtige Position nachgeführt werden.

## 2.3 Arbeitspakete

Arbeitspakete brechen das Projekt in einzelne Aufgaben runter und werden an einen zuständigen Mitarbeiter zugeteilt. Ein Arbeitspaket ist granular, das heißt es kann quantitativ und qualitativ von einem Mitarbeiter bearbeitet werden. Außerdem ist ein Arbeitspaket zeitlich begrenzt und dient der Projektfortschrittskontrolle. Daher ergibt sich ein detaillierter Projektlauf und ein Projektstrukturplan entsteht. In einem Projektstrukturplan sind alle im Projekt durchzuführenden Arbeiten aufgelistet und thematisch gegliedert. Für diese Projektarbeit ergibt sich folgender Projektstrukturplan 2.2:



Abbildung 2.2: Projektstrukturplan

Die Studienarbeit ist in vier Themen aufgeteilt: Physikalische Grundlagen, Technische Umsetzung, Projektmanagement und Dokumentation. Fragen wie, "Was ist unter den Keplerelementen zu verstehen und wie kann aus ihnen Azimut und Elevation bestimmt werden?", müssen bei dem Arbeitspaket Physikalischer Hintergrund beantwortet werden. Diese Beschreibung ist grundlegend zum Verständnis der zu in Betrieb nehmenden Software für die Satelliten Verfolgung und Antennennachführung. Das Inbetriebnehmen der Software wird in der "Technischen Umsetzung" durchgeführt. Dabei muss im ersten Schritt eine geeignete Software validiert werden. Die Inbetriebnahme setzt auch das Wissen über die Hard- und Softwareumgebung voraus, welche zu verwenden ist. Es folgt die Konfiguration und Test der geeigneten Software mit den definierten Schnittstellen.

Das Projektmanagement definiert die einzelnen Teilaufgaben und koordiniert diese in einem zeitlich festgelegten Rahmen um zu einem ergebnisorientierten Projektabschluss zu gelangen. Dabei ist auch das Erstellen einer Präsentation Teilaufgabe des Projektmanagements.

Die Dokumentation dieser Studienarbeit beschreibt den gesamten Projektentwicklungsprozess. Dabei wurden die Themen nach den Arbeitspaketen aufgeteilt. Der zugewiesene Student des Arbeitspaketes hat daher die Aufgabe, seine Erkenntnisse zu dokumentieren. Allgemein wurde sich auf das Drucksatzsystem zur Textformatierung L<sup>A</sup>T<sub>E</sub>X geeinigt. Daher ist für die schriftliche Ausarbeitung dieser Studienarbeit eine fortlaufende Dokumentenoptimierung nötig. Um das Arbeiten von drei Studierenden an einem Dokument zu ermöglichen wird sich an dem freien Softwaretool "Git" zur verteilten Versionsverwaltung bedient. Je vier Termine zu einem "Merge" (engl., Zusammenführen) der aufgeteilten Ausarbeitungen werden zur Versionsverwaltung genutzt und dabei Änderungen von Formatierung diskutiert und zusätzlich Arbeitspaketfortschritte erörtert.

Durch das Aufgliedern des Projekts in einzelne Arbeitspakete wurden die Prozesse, Aktivitäten die durchzuführen sind, die Reihenfolge des Arbeitsablaufes durch die definierte Zeitvorgabe, die Ergebnisdefinition, die vorhandenen Ressourcen sowie die anzuwendenden Werkzeuge (L<sup>A</sup>T<sub>E</sub>X und Git) definiert. In der folgenden Grafik 2.3 sind alle für die Studienarbeit benötigten Arbeitspakete mit Titel (Name), Verantwortlichem (Responsible), Zuweisung (Allocation) und Arbeitszeit (Estimate) aufgeführt.

Name	Responsible	Allocation	Estimate	Constraint Start	Constraint End
Physikalische Grundlagen	Maximilian Stiefel	Maximilian Stiefel	8,0h	14.10.2015 11:12	11.11.2015 11:12
Recherche	Maximilian Stiefel	Maximilian Stiefel	10,0h	14.10.2015 11:13	08.06.2016 11:13
Beschreibung	Maximilian Stiefel	Maximilian Stiefel	8,0h	14.10.2015 11:13	08.06.2016 11:13
Keplerelemente	Maximilian Stiefel	Maximilian Stiefel	5,0h	18.05.2016 11:14	01.06.2016 11:14
Vorhersagemodelle	Maximilian Stiefel	Maximilian Stiefel	5,0h	11.05.2016 11:14	08.06.2016 11:14
Datenformate	Maximilian Stiefel	Maximilian Stiefel	8,0h	11.05.2016 11:14	08.06.2016 11:14
Doppler-Shift	Maximilian Stiefel	Maximilian Stiefel	15,0h	14.10.2015 11:12	13.07.2016 11:12
Amateurfunktssatelliten	Maximilian Stiefel	Maximilian Stiefel			
Dokumentation	Maximilian Stiefel	Maximilian Stiefel			
Technische Umsetzung	Hannes Bohnengel				
Recherche	Hannes Bohnengel	Hannes Bohnengel,Maximilian Stiefel	5,0h	11.11.2015 11:23	09.12.2015 11:23
Bibliothek Orekit	Hannes Bohnengel	Hannes Bohnengel	5,0h	13.04.2016 11:24	27.04.2016 11:24
Open Source Programme	Hannes Bohnengel	Hannes Bohnengel			
Inbetriebnahme GPredict	Hannes Bohnengel	Hannes Bohnengel	10,0h	13.04.2016 11:25	15.06.2016 11:25
Konfiguration von GPr...	Hannes Bohnengel	Hannes Bohnengel			
Inbetriebnahme Schnittste...	Hannes Bohnengel	Hannes Bohnengel	8,0h	13.04.2016 11:25	27.04.2016 11:25
Konfiguration der Schn...	Hannes Bohnengel	Hannes Bohnengel			
Integration	Hannes Bohnengel	Hannes Bohnengel	15,0h	27.04.2016 11:25	15.06.2016 11:25
Integration GPredict m...	Hannes Bohnengel	Hannes Bohnengel	10,0h	15.06.2016 11:26	22.06.2016 11:26
Test des Systems	Hannes Bohnengel	Hannes Bohnengel			
Beschreibung	Hannes Bohnengel	Sarah Brückner	8,0h	25.11.2015 11:26	29.06.2016 11:26
Hard- und Software-U...	Hannes Bohnengel	Hannes Bohnengel	15,0h	14.10.2015 11:23	13.07.2016 11:23
Dokumentation	Hannes Bohnengel	Hannes Bohnengel			
Projektmanagement	Sarah Brückner				
Anforderungsanalyse	Sarah Brückner	Sarah Brückner	5,0h	14.10.2015 11:37	20.04.2016 11:37
Arbeitspakete	Sarah Brückner				
Definition	Sarah Brückner	Sarah Brückner	5,0h	14.10.2015 11:38	20.04.2016 11:38
Zuweisung	Sarah Brückner	Sarah Brückner	2,0h	14.10.2015 11:38	20.04.2016 11:38
Zeitplanung	Sarah Brückner				
Gantt-Diagramm erstel...	Sarah Brückner	Sarah Brückner	5,0h	20.04.2016 11:38	04.05.2016 11:38
Protokollführung	Sarah Brückner	Sarah Brückner	8,0h	14.10.2015 11:37	05.07.2016 11:37
Präsentation	Sarah Brückner	Sarah Brückner	10,0h	01.06.2016 11:37	01.06.2016 11:37
Dokumentation	Sarah Brückner	Sarah Brückner	15,0h	14.10.2015 11:38	13.07.2016 11:38
Dokumentenoptimierung	Sarah Brückner				
Git-Merge1	Sarah Brückner	Hannes Bohnengel,Sarah Brückner,Maximilian Stiefel	2,0h	17.05.2016 14:00	17.05.2016 16:00
Git-Merge2	Sarah Brückner	Hannes Bohnengel,Sarah Brückner,Maximilian Stiefel	2,0h	30.05.2016 12:00	30.05.2016 14:00
Git-Merge3	Sarah Brückner	Hannes Bohnengel,Sarah Brückner,Maximilian Stiefel	2,0h	10.06.2016 11:30	10.06.2016 13:30
Git-Merge4	Sarah Brückner	Hannes Bohnengel,Sarah Brückner,Maximilian Stiefel	2,0h	15.06.2016 09:00	15.06.2016 11:00

Abbildung 2.3: Arbeitspakete

Der Arbeitspaketverantwortliche besitzt die Verantwortung für die Arbeitspaketdurchführung in Qualität, Zeit und Kosten. Dieser Steuert und hat die Kontrolle aller Arbeiten in seinem Arbeitspaket. Ebenso Koordiniert der Verantwortliche seine im Arbeitspaket beteiligten Mitarbeiter. Die Berichterstattung über den Status im Arbeitspaket folgt bei einem der “Merges”.

## 2.4 Gantt-Diagramm

Um die einzelnen Arbeitspakete koordinieren zu können werden aus diesen ein Gantt-Diagramm erstellt. Hier werden die einzuhaltenden Endtermine der einzelnen Arbeiten sowie die Parallelisierung von unabhängig zueinander liegenden Arbeiten geplant. Die Erstellung eines Gantt-Diagramms erfolgt mit dem Programm “Plan” zur Projektplanung und -verwaltung. In der Grafik 2.4 wird die Parallelisierung der Arbeitspakete deutlich.

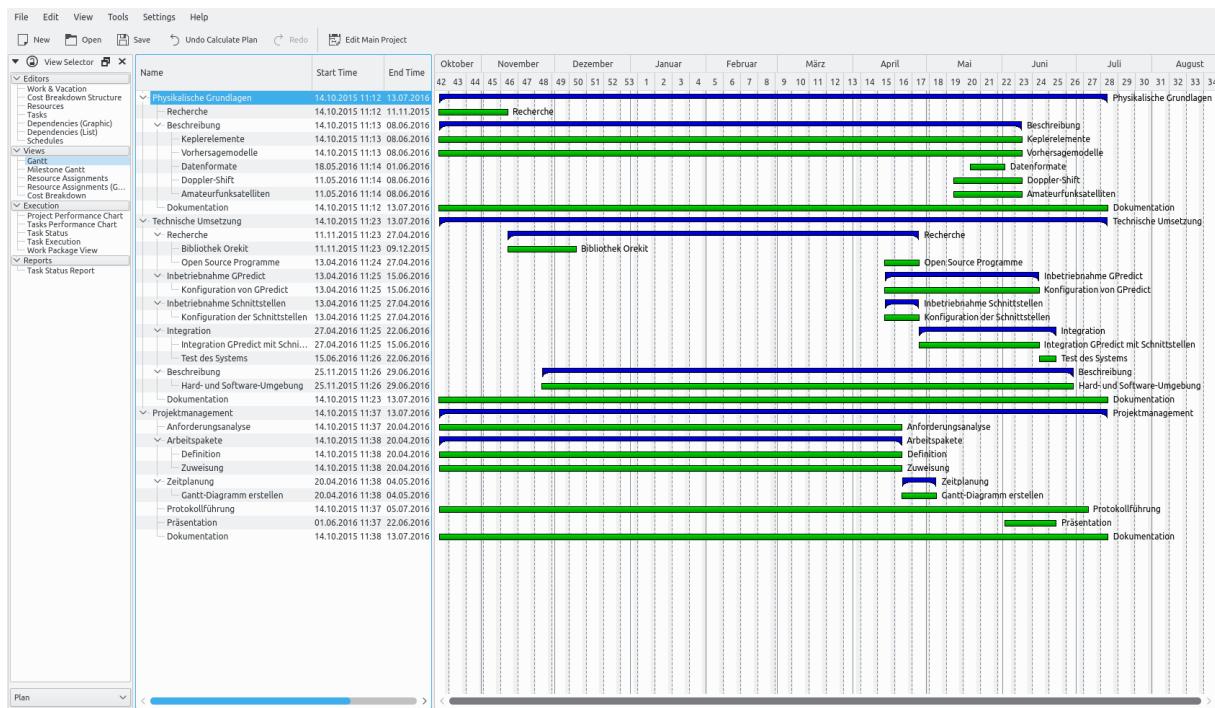


Abbildung 2.4: Gantt-Diagramm

Die drei Einheiten Physikalische Grundlagen, Technische Umsetzung und Projektmanagement können parallel bearbeitet werden. Begonnen wird mit einer umfassenden Recherche über die Keplerelmente und deren Berechnung. Parallel dazu kann über die Realisierung der Software für die Satellitenverfolgung recherchiert werden. Die dadurch gewonnenen Erkenntnisse führen zu der Anforderungsdefinition. Analog zum V-Modell kann nun der nächste Schritt eingeleitet werden. Dieser entspricht in dieser Studienarbeit der Inbetriebnahme von der Software “GPredict”. Zu diesem Arbeitspaket gehören die Aufgaben Konfiguration der Software und Konfiguration der Schnittstellen. Darüber hinaus muss sich der Verantwortliche umfassend mit der Software auseinander setzen um im nächsten Schritt die Integration beider Komponenten, Software und Hardware, durchzuführen. Ist eine funktionsfähige Kommunikation beider Komponenten bereitgestellt, können Tests, wie das Verfolgen von Satelliten, geprüft werden. Bei den Tests wird unterschieden in Modultest, das Testen der einzelnen Komponenten auf ihre Funktionsfähigkeit, Integrationstest, Kommunikationsfähigkeit der Software mit der Hardware und Systemtest, das Verfolgen von Satelliten und Antennennachführung. Alle Erkenntnisse die in den Arbeitspaketen gewonnen wurden müssen dokumentiert werden. Daraus schließt sich das letzte Arbeitspaket, die Dokumentation und die damit einhergehende Dokumentenoptimierung.

# 3 Einführung

## 3.1 Amateurfunksatelliten

Seit den sechziger Jahren sind Amateurfunker an der Raumfahrt beteiligt. Nur vier Jahre nach dem Start des ersten Satelliten Sputnik 1 wurde der erste Amateurfunksatellit OSCAR-1 am 12. Dezember 1961 erfolgreich gestartet. OSCAR bedeutet Orbiting Satellite Carrying Amateur Radio und dieser Satellit war 22 Tage in seinem Orbit. Dabei konnten 570 Funkamateure in 28 Ländern ihre Beobachtungen zum OSCAR-Projekt melden. In dieser Studienarbeit soll die Bodenstation erdnahe Satelliten verfolgen können. Allgemein teilt man die Satelliten in GEO, MEO, HEO und LEO Satellitenbahnen ein [2].

- GEO (Geostationary Orbit): Umlaufbahn in 35790 km Höhe und Umlaufzeit von einem Tag. Empfangsfenster für die Bodenstation: Immer. Dazu gehören unter anderen die Satelliten Astra und Meteosat.
- MEO (Medium Earth Orbit): Umlaufbahn in 6000-36000 km Höhe und Umlaufzeit von 5-12 Stunden. Empfangsfenster für die Bodenstation: 2-4 Stunden. Darunter fallen die Navigationssatelliten wie GPS, Galileo und GLONASS.
- HEO (Highly Elliptical Orbit): Umlaufbahn in 200-15.000 km bzw. 50.000-400.000 km und Umlaufzeit von  $\sim$  12 Stunden. Empfangsfenster für die Bodenstation: 8 Stunden. Satelliten in diesem Orbit eignen sich zur Versorgung von Polargebieten, da die geostationären Satelliten oberhalb von 82° Elevation nicht mehr zu empfangen sind. Beispiele für HEO Satelliten sind AO10, AO13, AO40.
- LEO (Low Earth Orbit): Umlaufbahn in 200-1500 km Höhe und Umlaufzeit von 1.5-2 Stunden. Empfangsfenster für die Bodenstation: unter 15 Minuten. Dazu gehören die Amateurfunksatelliten, welche der Satellitenkommunikation zwischen Funkamateuren und auch zu experimentellen Zwecken dienen, wie AO-7 und AO-51. Außerdem befindet sich die ISS auf dieser niedrigen Erdumlaufbahn.

Amateurfunksatelliten kommunizieren im 2-Meter- und 70-Zentimeterband [3] und ermöglichen einen internationalen Sprech- und Datenfunk. Außerdem senden diese Satelliten auch Messwerte der Betriebsdaten des Satelliten. Diese Satelliten werden meist von

Hochschulen oder Amateurfunkvereinigungen gebaut und mit weiteren Satelliten an Board einer Rakete in das All geschossen. Dabei handelt es sich um kleine, leichte Satelliten, auch "Cubesat" genannt.

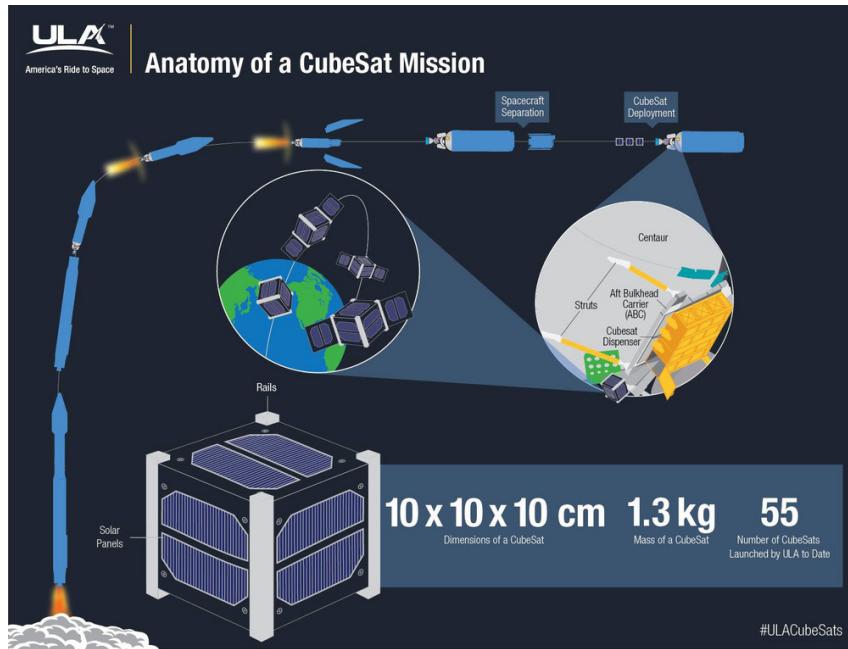


Abbildung 3.1: Cubesat, Quelle: [4]

Die Grafik 3.1 zeigt einen beispielhaften Cubesat und wie dieser von einer Rakete in das All geschossen wird. Im Vergleich, der ASTRA 1L Satellit mit einer Spannweite von 20 Meter und einem Gewicht von 4,5 Tonnen. Erdnahe Satelliten umkreisen die Erde ständig und haben daher nur ein beschränktes Zeitfenster, in dem man mit einem solchen Satelliten arbeiten kann. Davor befindet sich der Satellit durch die Erdkrümmung hinter dem Horizont und ist daher unerreichbar. Daher ist es für die Bodenstation wichtig zu wissen wann der Satellit am Horizont auftaucht um die Antenne in die richtige Position nach zuführen. Nach der korrekten Nachführung folgt die Kontaktaufnahme. Dafür besitzt jeder Satellit einen Transponder um Daten zu empfangen und wieder abzustrahlen. Dabei wird der Datenstrom ohne qualitative Veränderung weitergeleitet. In der Tabelle 3.1 sind die Transpodermodi der Amateurfunksatelliten gelistet. Der Modus gibt an, in welchem Frequenzbereich der Satellit hört und auf welchem er sendet. Ein Transponder in dem Modus VU empfängt im 2-m-Band und sendet im 70-cm-Band. Ein Signal, das der Satellit zur Erde schickt nennt man "Downlink" und von der Basisstation zum Satelliten "Uplink". Einige Cubesats senden in Continious Wave (CW) das Rufzeichen sowie ihre Telemetriedaten aus und dienen der Funktionskontrolle.

Tabelle 3.1: Abkürzungen der Transpondermodi, Quelle: [5]

Bezeichnung	Band
V	2-m-Band
U	70-cm-Band
L	23-cm-Band
S	13-cm-Band
X	3-cm-Band

Um Informationen zu empfangen oder zu senden, gibt es einige Möglichkeiten im Amateurfunk. Es werden im Folgenden die wichtigsten Betriebsarten vorgestellt.

- SSB – Single Side Band: Eine von den Amateurfunkern eingeführten Betriebsart für das Fernsprechen. Diese ist eine leistungsfähigere und effizientere Weiterentwicklung der Amplitudenmodulation. Bei der Amplitudenmodulation werden zwei Seitenbänder mit den Informationen und ein Träger benötigt. Daher ist ein AM-Signal sehr breit und störanfällig. Anders bei der SSB Betriebsart. Hier wird auf den Träger und dem zweiten Seitenband verzichtet. Trotz dessen kann ein gutes Sprachsignal übertragen werden.
- FM – Frequenzmodulation: Bei der Frequenzmodulation wird die Information in eine Frequenz gepackt und die Amplitude bleibt dabei konstant. FM ermöglicht eine qualitativ gute, störungsfreie Übertragung von Sprache.
- Packet Radio – Datenübertragung: Diese Betriebsart überträgt digitale Daten in dem kurze Datenpakete ausgesendet werden und am Empfänger wieder zusammen gesetzt werden. Dadurch können Funkamateure mit ihren üblichen UKW-Funkgeräten untereinander Daten austauschen. Für die Kommunikation zwischen Amateurfunksatelliten und den Bodenstationen wird vorwiegend das Packet-Radio-Protokoll mit FSK-Modulation (Frequenzumtastung) verwendet.

Die 1969 gegründete Organisation AMSAT ist eine Vereinigung von Funkamateuren, welche Raumfahrtsatelliten entwickeln und kostengünstige Raumfahrtmissionen realisieren. Unter anderem dokumentiert AMSAT auch alle Amateurfunksatelliten und deren Status. In der Grafik 3.2 sind die Satelliten mit dem aktuellen Statusbericht (04. Juni 2016)

Name	Jun 4	Jun 3	Jun 2	Jun 1	May 31	May 30
CUTE-1	1	1	1	1	1	1
NUSAT-1	1	1 1	1	2	111	1
QUFTI-1	1	11	1	1	1	1
UKube-1		1			11	2
LilacSat-2	1 1 2	2 1 1	1 2	2	11 1 1 1 2	1 1 1 1 2
UWE-3						1
[A] AO-7	1 1		1	1		11
[B] AO-7	23 1 1111 1 1211	1 12223221 311 1111 1	11 21 3321	112 2211		
XI-V				1		
RS-15	2	2	2	2	2	2
AO-16	1					
LO-19						1
FO-20	1					
FO-29	4232 11 111112 1 224122211111 2134221 1 211112511 12223131					
XW-2A	1 1 3 1 1 2 114 11 1 213 1 1 1 1 12 11 1 23 222 1					
XW-2B	2 1 1 22 1 1 1 13 1111 221 1 1 1212 2 11 25 211					
XW-2C	112 1 1 1 1 1 1 1 1132 3 1211 21311 1 1 121 131 1 231 11					
XW-2D			1		1	2 1
XW-2F						
NO-44	1			1	12	11
SO-50	131 22 21 111 2 52 11 21 21 221 21 53 22 22 244 1					
NO-62	1	1				
AO-73	151 211 1361 1111321 11 111 11 2 432 1 2 35111					
EO-79			1			12
NO-84	1 1 1 1			1 1		1
AO-85	2341 1311 321 1 11 42322 16 5611 134 4224 324 25335					
IO-86	1					
AO-98	1			1		
Delfi-C3	11 11 1 1 1 11 11 1 1 1					11
ISS-FM					2	
XI-IV	1	1	1	1	1	1
DUCHIFAT1	1		1			1
ISS-DATA	11221 11 33 1221 4 321 11111 4					

Abbildung 3.2: AMSAT Live OSCAR Satellite Status, Quelle: [6]

gelistet. Diejenigen Satelliten mit blauer Markierung sind aktiv, wohingegen die roten Satelliten kein Signal zu dem jeweiligen Zeitpunkt von sich gegeben haben. Gelb bedeutet, dass nur die Telemetriedaten empfangen wurden und orange, dass Konflikte bestehen. Für die Studienarbeit sind diese Informationen sehr nützlich, um Satelliten tracken zu verfolgen.

## 3.2 Aufbau der Bodenstation

In diesem Abschnitt werden die von der Bodenstation der DHBW Ravensburg zur Verfügung stehende Komponenten vorgestellt.

### 3.2.1 Antennen

Die Sendeleistung der Satelliten liegt typischerweise unter 1 Watt. Im 2-m-Band wäre es möglich mit einer Rundstrahlantenne ein Signal zu empfangen, doch auf den höheren Bändern wird aufgrund der kürzeren Wellenlänge und damit einhergehende höheren Streckendämpfung eine Richtantenne benötigt. Da für die Kommunikation über Satelliten beide Polarisationsarten, vertikal und horizontal, benötigt werden, kommen Kreuzyagi-Antennen zum Einsatz. In der unteren Grafik ist von links nach rechts eine Kreuzyagi-Antenne für das 70-cm-Band, eine Kreuzyagi-Antenne für das 2-m-Band und zwei Yagi-antennen, welche von  $90^\circ$  um eine  $\frac{\lambda}{4}$  Wellenlänge auf dem Tragrohr versetzt sind, für das 23-cm-Band, zu sehen.

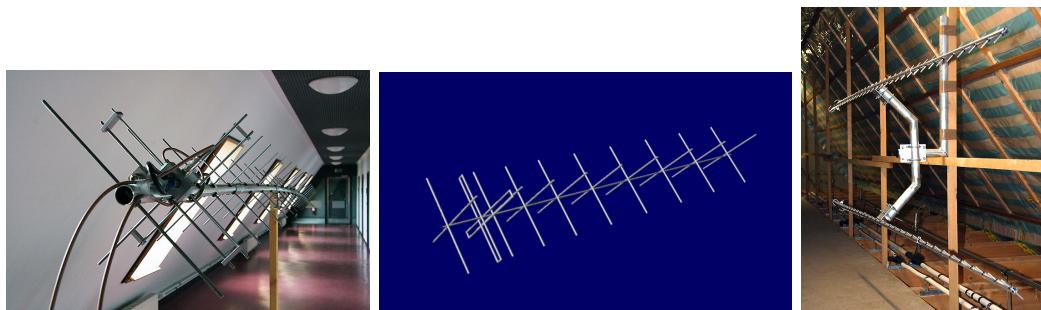


Abbildung 3.3: 436CP42UG 70-cm, WX220 2-m und 23CM35 23-cm, Quelle: [7]

Dazu gehört ein Mast sowie Vorverstärker und Polarisationsschalter. Die Antennen für das 23- und 70-cm-Band kommen von der Firma M<sup>2</sup> und die Antenne für das 2-m-Band von der Firma WIMO. Yagiantennen bestehen aus einem Faltdipol, einem Reflektor und mehreren Direktoren. Aus der Vorlesung Hochfrequenztechnik [8] ist bekannt, dass ein Reflektor ca. 5 % länger als  $\frac{\lambda}{2}$  ist, somit oberhalb seiner Resonanzfrequenz betrieben wird und daher induktiv wirkt. Anders ist es beim Direktor. Dieser wird von Direktor zu Direktor um ca. 5 % kürzer und wird im unteren Resonanzfrequenzbereich betrieben. Der Reflektor erhöht den Antennengewinn durch Reflexion der einfallenden Wellen auf dem

Dipol. Die Direktoren beeinflussen die Abstrahlcharakteristik der Antenne, da die keulenförmige Ausprägung stärker wird. Bei der linken Kreuzyagi-Antenne sieht man in der Grafik den Reflektor und dahinter, den Faltdipol. Darauf folgen die Direktoren. Folgende Antennengewinne werden von den Antennen bereitgestellt: 436CP42UG (70-cm): 18.90 dBi, 23CM35 (23-cm): 20.94 dBi und WX220 (2-m): 35.00 dBi.

### 3.2.2 Rotoren

Um die Richtantennen steuern zu können werden Rotoren sowie dessen Steuergeräte von der Bodenstation zur Verfügung gestellt.

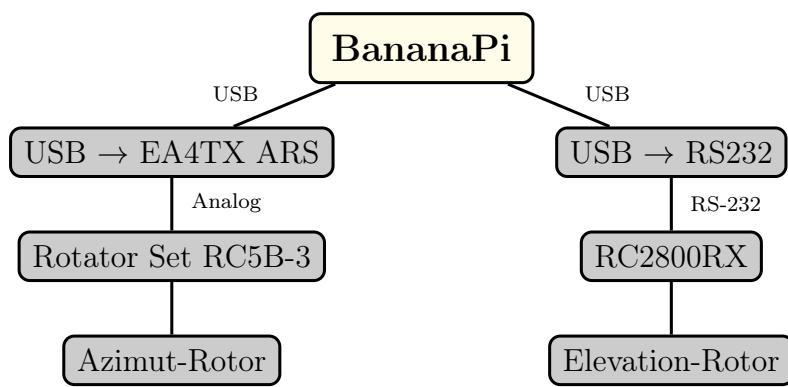


Abbildung 3.4: Rotorsteuerung

In dem Organigramm 3.4 stellt der BananaPi, ein Einplatinencomputer, die Rotorsteuerung als Einheit in dem DHBW Netzwerk bereit. Über die ARSVCOM Schnittstelle kann auf ihn zugegriffen werden. Die Bodenstation verfügt über zwei unterschiedliche Steuergeräte für den Elevationsrotor und den Azimutrotor. Dieser BananaPi emuliert zwei einheitliche Steuergeräte und setzt die Kommandos des ARS Protokolls um. Über USB sind die Steuergeräte für den Azimut- und Eleavtions-Rotor an den BananaPi angeschlossen. Das Steuegerät RC2800PXEL für den Elevationsrotor von der Firma M<sup>2</sup> arbeitet mit der seriellen Schnittstelle RS-232. Daher wird ein USB → RS232 Adapter benötigt. Das analoge Steuergerät RC5B-3-P von Create für den Azimutrotor benötigt einen Umsetzer von dem digitalen in den analogen Bereich. Dies geschieht über das Antennen-Rotor System ARS-USB von EA4TX.

### 3.2.3 Funkgerät ICOM IC-9100



Abbildung 3.5: Funkgerät ICOM IC-9100, Quelle: [7]

Die letzte zentrale Komponente der Bodenstation ist das Funkgerät und ermöglicht den Funkbetrieb. Das IC-9100 Funkgerät der Firma ICOM ist ein Multiband Transceiver mit Doppelempfangsmöglichkeit, da der IC-9100 über drei unabhängige Empfängerschaltungen verfügt. Die Sendebandbreite ist durch wählbare Hochpass- und Tiefpassfrequenz einstellbar und das Funkgerät verfügt über eine Sendeleistung von 100 Watt. Eine für die Bodenstation wichtige Funktion ist der Funkbetrieb über Amateurfunksatelliten. Im einstellbaren Satellitenbetrieb synchronisiert der IC-9100 die Sende- und Empfangsfrequenz und stimmt beide mit der gleichen Abstimmschrittweite ab. Das Funkgerät kann in den Satelliten Modi B (435 MHz Uplink, 145 MHz Downlink) und J ( 145 MHz Uplink, 435 MHz Downlink) betrieben werden. Die Kommunikation über den PC mit dem Funkgerät erfolgt über die Software Netcom. Das Netcom-Modul ist mit dem ICOM IC-V Bus verbunden und bietet neben einer USB-Schnittstelle mit emulierter Soundkarte auch eine Netzwerkschnittstelle über die man mit dem Netcom Manager (der wiederum eine viruelle serielle Schnittstelle zur Verfügung stellt) mit dem Funkgerät kommunizieren kann. Das IC-9100 kann die wichtigsten Amateurfunkbänder, von Kurzwelle bis Ultra-High-Frequency (UHF), erfassen.

# 4 Hintergründe

## 4.1 Bahnmechanik

### 4.1.1 Die Keplerschen Gesetze

Seit der Antike galt die Erklärung der Bewegung der Planeten und die Vorhersage dieser als eine große Herausforderung. Theorien von Ptolemaios mit seinem geozentrischen Weltbild und Kopernikus mit seinem heliozentrischen Weltbild führten bereits im 16. Jahrhundert zu brauchbaren Modellen zur Vorhersage der Planetenbewegungen. Diese Modelle unterlagen jedoch Ungenauigkeiten, „die in mit Instrumenten des 16. Jahrhunderts bereits messbaren Breichen lagen“ (siehe S. 20 in [9]). Der mathematische Aufwand hinter diesen



Abbildung 4.1: Johannes Kepler (1571-1630), Quelle: [10]

Modellen war enorm. Selbst das kopernikanische Weltbild, dass einige Vereinfachungen mit sich brachte, bediente sich der Überlagerung einer Vielzahl von Kreisbewegungen, um das Verhalten der Planeten zu erklären. Resignierend zog sich zu der Zeit die katholische Kirche und mit ihr viele Gelehrte auf den Standpunkt zurück, dass „die Frage, welche der Theorien die korrekte sei, [...] schlicht unbeantwortbar“ wäre (siehe S. 21 in [9]).

Ein deutscher Mathematiker und Astronom, Johannes Kepler, war hier anderer Auffassung. Er war überzeugter Kopernikaniker und stand im Dienste des Kaisers Rudolph II. Schließlich gelang es ihm aus seinen Beobachtungen drei einfache Gesetze herzuleiten. Seine Gesetze führten zu Vorhersagen der Planetenbewegungen nie da gewesener Präzisi-

on, welche er seinem Dienstherr widmend in den Rudolphinischen Tabellen niederschrieb. Steiner und Schlagerl schreiben in Ihrem Buch „Raumflugmechanik“, dass ohne die Vorarbeit Keplers keine Weltraumtechnik je existiert hätte (vgl. S. 21 in [9]). Die drei Gesetze lauten:

1. Keplersches Gesetz: Die Planeten umlaufen die Sonne auf elliptischen Bahnen. In einem der Brennpunkte dieser Ellipsen befindet sich die Sonne.
2. Keplersches Gesetz: Die Linie von der Sonne zu einem Planeten überstreicht in gleichen Zeiten gleiche Flächen.
3. Keplersches Gesetz: Die Quadrate der Umlaufzeiten zweier Planeten verhalten sich zueinander so wie die Kuben der großen Halbachsen ihrer Bahnellipsen.

Kepler starb 1630 und damit 12 Jahre vor Isaac Newtons (1642-1726) Geburt. Mit seinen Werken hinterließ Kepler Newton alles, um das Gravitationsgesetz später herleiten zu können.

### **Das erste Keplersche Gesetz**

Durch die Annahme Planeten bewegen sich auf Ellipsen anstatt auf Kreisen, brach Kepler ein tausende Jahre altes Paradigma. Das mit der Ellipse war allerdings nicht seine Idee. Bereits im 11. Jahrhundert nahm ein arabischer Gelehrter namens Al-Zarkali (1029-1087) eine elliptische Bahn an, um die unregelmäßige Bewegung des Merkurs erklären zu können. Kepler kannte diese Idee durch die Lehren des Mathematikers und Astronomen Peuerbach (1423-1461), welcher die Ellipsen-Theorie im Abendland verbreitete.

Zunächst soll die Ellipse an sich betrachtet werden. Die einfachste Möglichkeit eine Ellipse zu konstruieren besteht darin zwei Nägel in einer Holzplatte mit einem Stück Schnur mit einer Schlaufe zu verbinden. Das Stück Schnur muss länger sein als der Abstand zwischen beiden Nägeln. Nimmt man nun einen Bleistift und drückt ihn in der Schlaufe gegen die Schnur, kann man die beiden Nägel mit Kontakt der Bleistiftspitze zum Holzbrett umrunden. Hält man die Schnur konstant auf Spannung, so ergibt sich eine Ellipse. Darüber hinaus muss der Punkt auf welchem die Schlaufe am Bleistift anliegt höher sein, als der Abschluss der Nagelköpfe. Im übertragenden Sinne beschreibt die folgende Mengendefini-

tion dieses Experiment mit Bezug zu Abbildung 4.2.

$$E = \{P | \overline{F_1P} + \overline{F_2P} = 2a = \text{konstant}\} \quad (4.1)$$

$F_1$  und  $F_2$  heißen Brennpunkte der Ellipse.  $M$  ist der Mittelpunkt der Ellipse.  $S_1$  und  $S_2$  sind die Haupt-,  $S_3$  und  $S_4$  die Nebenscheitel. Die Strecke  $\overline{MS_1}$  ist gleich der Strecke  $\overline{MS_2}$ .

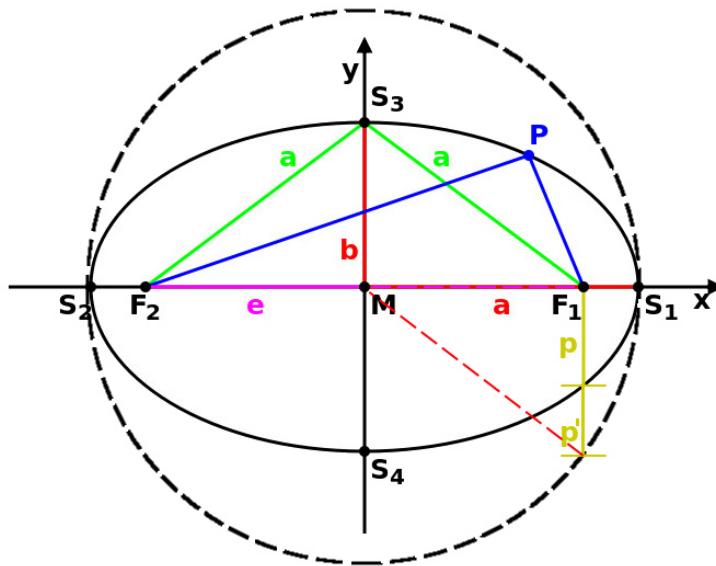


Abbildung 4.2: Ellipse, Quelle: [11] zusätzlich eigener Überarbeitung

Man spricht bei der Länge dieser Strecke von der **großen Halbachse a**. Beide Strecken ergeben zusammen die Hauptachse  $\overline{S_1S_2}$ . Analog gibt es hierzu die Nebenachse, welche durch die Strecke  $\overline{S_3S_4}$  bestimmt wird. Die **kleinen Halbachsen** sind  $\overline{MS_3}$  und  $\overline{MS_4}$ . Diese haben die Längen **b**. Eine Ellipse kann auch als Stauchung eines Kreises mit dem Faktor  $\frac{b}{a}$  angesehen werden.

Die numerische Exzentrizität  $e'$  ist ein Maß für die Schlankheit der Ellipse. Sie ist definiert als

$$e' = \frac{e}{a} \quad (4.2)$$

Je größer die lineare Exzentrizität  $e$  im Verhältnis zu der großen Halbachse  $a$  wird, desto schlanker wird die Ellipse, da die Brennpunkte weiter vom Mittelpunkt entfernt sind. Das Wort numerisch gibt bei der Exzentrizität  $e'$  an, dass diese sich auf eine andere Größe (die große Halbachse) bezieht. Für eine Ellipse gilt  $0 < e' < 1$ . Für den Fall  $e' = e = 0$  hat die Ellipse die selbe Erscheinung wie ein Kreis, da die Brennpunkte  $F_1$  und  $F_2$  im Mittelpunkt  $M$  liegen. Für  $e' = 1$  entartet die Ellipse zu einer Geraden, da die kleine Halbachse  $b$  zu

0 wird. Um das zu zeigen wird die obige Gleichung noch mal herangezogen.

$$e'^2 = \frac{e^2}{a^2} = \frac{a^2 - b^2}{a^2} = 1 - \left(\frac{b}{a}\right)^2 \quad (4.3)$$

Des Weiteren besitzt jede Ellipse einen Halbparameter  $p$ . Geht man davon aus, dass es einen Abstand  $p'$  gibt, welcher  $p$  bis zu einer die Ellipse umschließende Kreislinie verlängert, so gelten folgende Gleichungen

$$\frac{p}{p'} = \frac{b}{a} \quad (4.4)$$

Mit dem Satz eines alten Freudes folgt

$$p' = \sqrt{a^2 - e^2} \quad (4.5)$$

Jetzt ist klar, dass gilt

$$p = \frac{b}{a} \cdot p' = \frac{b}{a} \sqrt{a^2 - e^2} = \frac{b^2}{a} \quad (4.6)$$

Was nun noch fehlt ist „eine Gleichung, also eine analytische Beschreibung der Punkte einer Ellipse“ (siehe S.22 in [9]). Eine solche Gleichung ergibt sich mit dem Schnitt einer Ebene mit einem Kegel. Die Neigung der Schnittebene zur Kegelgrundfläche sei  $\alpha$ . Der Öffnungswinkel des Kegels sei  $\epsilon$ . Jetzt passiert etwas, dass das räumliche Denkvermögen herausfordert. In den Kegel wird eine (Dandelin'sche) Kugel eingeschrieben. Diese Kugel berühre die Ebene im Punkt  $F$  und tangiere den Kegelmantel entlang eines Breitenkreises. Es ist einzusehen, dass der Punkt  $F$  auf der Hauptachse der Ellipse liegt. Der Schnittpunkt der entstehenden Ellipse und der Normale zur Hauptachse im Punkt  $F$  ist der Punkt  $Q$ .  $P$  stellt einen beliebigen Punkt auf der Ellipse dar. Die Verbindungsgeraden zwischen  $F$  und  $P$  hat zur Hauptachse die Neigung  $\theta$ . Der Abstand zwischen  $F$  und  $P$  ist  $r$ .  $s$  und  $s_0$  sind die Abstände der Punkte  $P$  und  $Q$  zum Berührkreis der Kugel gemessen entlang des Kegelmantels.

Man sieht nun, dass die Größen  $r$ ,  $s$  und  $\theta$  von  $P$  abhängig sind. Es interessiere nun die mathematische Funktion  $r(\theta)$ , welche die Ellipsenbahn beschreibe (vgl. S.23 in [9]). Betrachtet wird nun zunächst die zweidimensionale Zeichnung rechts oben in Abb. ??.

Mit ein bisschen Nachdenken sieht man, dass

$$(s_0 - s) \cdot \cos(\epsilon) = r \cdot \cos(\theta) \cdot \cos(\alpha) \quad (4.7)$$

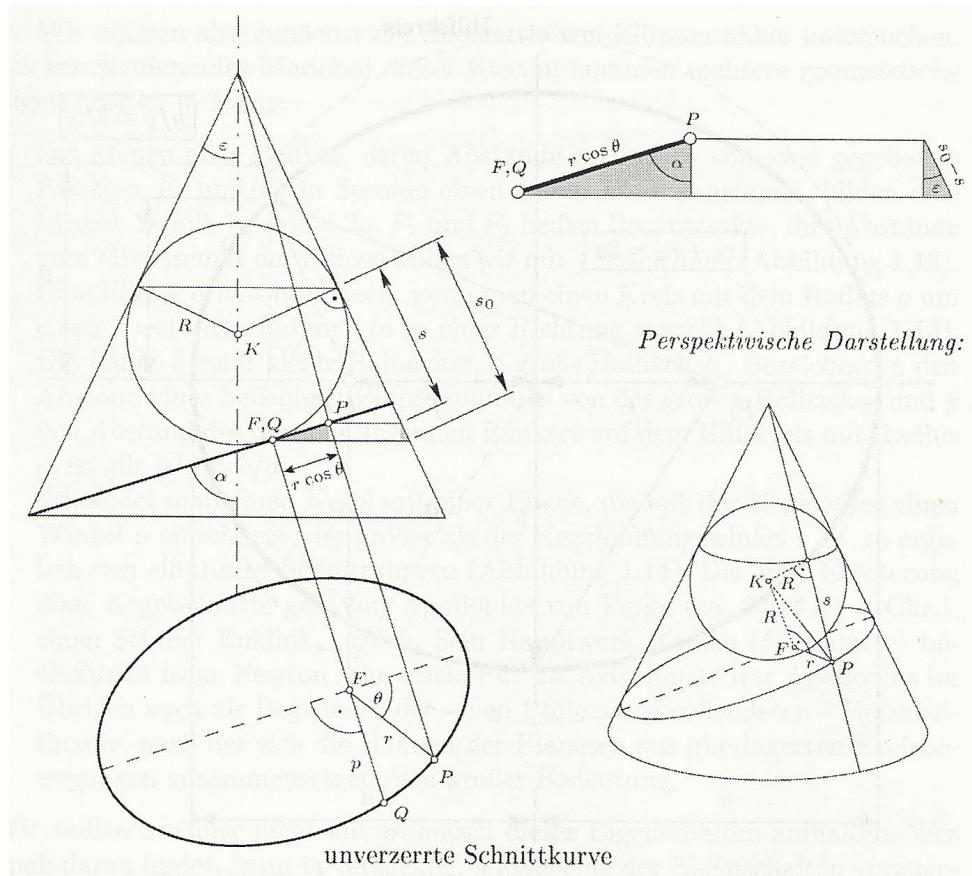


Abbildung 4.3: Kegelschnitt, Quelle: S.24 in [9]

Doch woher kommt der Ausdruck  $r \cdot \cos(\theta)$ ? Hierzu werfe man einen Blick auf die zweidimensionale Abbildung der Schnittfläche/Ellipse links unten im Bild. Dieses Bild setze man nun in Relation zum Bild darüber. Der Abstand  $r \cdot \cos(\theta)$  lässt sich nun auf die Hauptachse der Ellipse projizieren.  $r$ , die Projektionslinie für P die Hauptachse und F bilden nun eine rechtwinkliges Dreieck aus. Der Rest ist Trigonometrie.

In der Darstellung rechts unten in Abb. ?? ist folgende Beziehung auffindbar.

$$R^2 + s^2 = R^2 + r^2 \quad (4.8)$$

Das bedeutet, dass  $s$  durch  $r$  in Gleichung 4.7 ersetzt werden kann.

$$(s_0 - r) \cdot \cos(\epsilon) = r \cdot \cos(\theta) \cdot \cos(\alpha) \quad (4.9)$$

Ausmultiplizieren ergibt

$$s_0 \cos(\epsilon) - r \cos(\epsilon) = r \cos(\theta) \cos(\alpha) \quad (4.10)$$

Sortieren führt zu

$$s_0 \cos(\epsilon) = r \cos(\theta) \cos(\alpha) + r \cos(\epsilon) \quad (4.11)$$

Ausklammern und auflösen bringt

$$r = \frac{s_0 \cos(\epsilon)}{\cos(\theta)\cos(\alpha) + \cos(\epsilon)} \quad (4.12)$$

Die entstandene Gleichung 4.12 kann nun noch durch die Zusammenhänge  $p = s_0$  (Halbparameter) und  $e' = \frac{\cos(\alpha)}{\cos(\epsilon)}$  vereinfacht werden (vgl. S. 24 in [9]). Hierzu dividiert man Gleichung 4.12 durch  $\cos(\epsilon)$ .

$$r = \frac{s_0}{1 + \frac{\cos(\alpha)}{\cos(\epsilon)} \cos(\theta)} = \frac{p}{1 + e' \cos(\theta)} \quad (4.13)$$

Fertig ist die mathematische Version des ersten Keplerschen Gesetzes.

## Das zweite Keplersche Gesetz

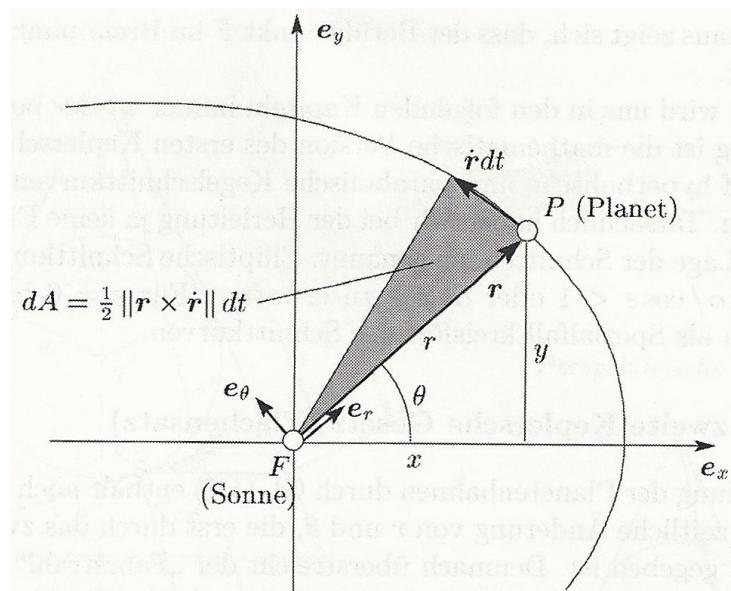


Abbildung 4.4: Kegelschnitt, Quelle: S.26 in [9]

Gleichung 4.13 liefert noch keine Aussage über die zeitliche Änderung von  $r$  und  $\theta$ . Um eine Aussage über die zeitliche Änderung dieser Variablen treffen zu können, kann Keplers zweites Gesetz herangezogen werden: Die Fläche, welche die Verbindungs linie zwischen Sonne und einem Planeten überstreckt ist zeitlich konstant. Die Fläche  $\Delta A$ , die in einem Zeitintervall  $\Delta t$  durch strichen wird ist genau

$$\Delta A = \frac{1}{2} \left| \vec{r} \times \dot{\vec{r}} \right| \Delta t + F(\Delta t^2) \quad (4.14)$$

Die beiden Vektoren  $r$  und  $\dot{r}\Delta t$  spannen eine Fläche auf, welche ein Parallelogramm beschreibt. Das Kreuzprodukt ergibt einen Vektor dessen Länge dieser Fläche entspricht.

Die Hälfte davon ist die Fläche des Dreiecks, die gesucht wird. Der Ausdruck  $\dot{r}\Delta t$  ist dabei sehr ungenau und beschreibt eigentlich nur die Änderung des Vektors  $r$ . Aus diesem Grund kommt noch der Fehlerterm  $F$  hinzu, der die Krümmung der Ellipse berücksichtigt. Bezieht man sich im nächsten Schritt auf infinitesimale Elemente, die wirklich gegen Null gehen, so erreicht man die gewünschte Genauigkeit. Der Fehlerterm wird überflüssig.

$$dA = \frac{1}{2} \left| \vec{r} \times \dot{\vec{r}} \right| dt \quad (4.15)$$

Man führt nun eine für jeden Planeten individuelle Konstante  $h$  ein, da sich das Verhältnis  $\frac{dA}{dt}$  nicht verändern darf.

$$2dA = h \cdot dt \quad (4.16)$$

Um das mathematische Äquivalent zu dem sprachlich formulierten zweiten Gesetz zu erhalten, soll wie beim ersten Gesetz eine Abhängigkeit zu  $r$  und  $\theta$  hergestellt werden. Zu diesem Zweck wird die Gleichung einer Koordinatentransformation in Zylinderkoordinaten unterworfen (vgl. S. 25 f. in [9]). Es gilt also

$$x = r \cos(\theta), \quad y = r \sin(\theta) \quad \text{und} \quad z = z.$$

Gemäß der Definition von Zylinderkoordinaten darf man jetzt den Vektor  $\vec{r}$  auch anders schreiben:

$$\vec{r} = r \vec{e}_r + z \vec{e}_z \quad (4.17)$$

wobei folgendes generell über Zylinderkoordinatensysteme bekannt ist:

$$\vec{e}_r = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix}, \quad \vec{e}_\theta = \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{pmatrix} \quad \text{und} \quad \vec{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Die Vektoren stehen allesamt senkrecht aufeinander, was man sieht, wenn man das Skalarprodukt bildet. Das liegt daran, dass das Skalarprodukt über die Summe der Längen der Vektoren multipliziert mit dem Kosinus des Winkels den sie einschließen definiert wird, welcher bei  $\frac{\pi}{2}$  bekanntlich Null ist. Da auch die Ableitung des Vektors  $\vec{r}$  (Geschwindigkeit) gesucht ist beginnt man zu differenzieren. Man wende hier zunächst die Summenregel, dann auf den ersten Ausdruck noch die Produkt- und die Kettenregel, da  $\vec{e}_r$  von  $\theta$  abhängt und diese wiederum von  $t$ . Es folgt

$$\dot{\vec{r}} = \dot{r} \vec{e}_r + r \dot{\vec{e}}_r + \dot{z} \vec{e}_z \quad (4.18)$$

Setzt man nun

$$\dot{\vec{e}_r} = \dot{\theta} \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{pmatrix} = \dot{\theta} \vec{e}_\theta$$

in Gleichung 4.18 ein, so ergibt sich

$$\dot{\vec{r}} = \dot{r} \vec{e}_r + r \dot{\theta} \vec{e}_\theta + \dot{z} \vec{e}_z \quad (4.19)$$

Man wählt nun die z-Achse geschickt, so dass diese senkrecht auf der Trägerebene der Ellipse steht (vgl. S.26 in [9]). Durch diesen Schachzug gilt für die zu betrachtenden Gleichungen  $z = 0$ . Jetzt fällt Gleichung 4.15 in sich zusammen

$$\frac{dA}{dt} = \frac{1}{2} \left| \vec{r} \times \dot{\vec{r}} \right| = \frac{1}{2} \left| r \vec{e}_r \times (\dot{r} \vec{e}_r + r \dot{\theta} \vec{e}_\theta) \right| \quad (4.20)$$

Durch die Bilinearität des Kreuzprodukts folgt

$$\frac{dA}{dt} = \frac{1}{2} \left| \dot{r} r (\vec{e}_r \times \vec{e}_r) + r r \dot{\theta} (\vec{e}_r \times \vec{e}_\theta) \right| \quad (4.21)$$

Die Tatsache, dass das Kreuzprodukt eines Vektors mit sich selbst den Nullvektor ergibt und dem Umstand, dass  $\vec{e}_r \perp \vec{e}_\theta \perp \vec{e}_z$  ist, führt zu

$$\frac{dA}{dt} = \frac{1}{2} \left| r^2 \dot{\theta} \vec{e}_z \right| = \frac{1}{2} r^2 \dot{\theta} = h = \text{konstant} \quad (4.22)$$

### Das dritte Keplersche Gesetz

Auf sein drittes Gesetz soll Kepler besonders stolz gewesen sein (vgl. S.27 in [9]). Nach seinem Gesetz stehen die Halbachsen  $a_1$  und  $a_2$  und die Umlaufzeiten  $T_1$  und  $T_2$  in dem Zusammenhang

$$\frac{T_1^2}{T_2^2} = \frac{a_1^3}{a_2^3} \quad (4.23)$$

Die Umlaufzeit lässt sich nun leicht mit der Konstanten  $h$ , welche für das zweite Keplersche Gesetz eingeführt wurde ausdrücken. Es wird hierfür die gesamte Ellipsenfläche ( $A_{\text{Ellipse}} = ab\pi$ ) und konsequenterweise dann auch die gesamte Umlaufzeit  $T$  angenommen. Eingesetzt in Gleichung 4.22 ergibt sich

$$2ab\pi = hT. \quad (4.24)$$

Daraus folgt

$$T = \frac{2ab\pi}{h}. \quad (4.25)$$

Eingesetzt in Gleichung 4.23 ergibt sich

$$\frac{T_1^2}{T_2^2} = \frac{a_1^3}{a_2^3} = \frac{a_1^2 b_1^2 h_2^2}{a_2^2 b_2^2 h_1^2}. \quad (4.26)$$

Bringt man nun  $a_1^2 b_1$  und  $a_2^2 b_2^2$  einmal durch Division und einmal durch Multiplikation eins weiter nach links, so kann man die Beziehung aus Gleichung 4.6 ausnutzen und schreiben

$$\frac{a_1 b_2^2}{a_2 b_1^2} = \frac{h_2^2}{h_1^2} \quad (4.27)$$

Oben und unten multipliziert man jetzt noch jeweils mit  $1/a_1$  und  $1/a_2$ .

$$\frac{b_2^2/a_2}{b_1^2/a_1} = \frac{h_2^2}{h_1^2} = \frac{p_2}{p_1} \quad (4.28)$$

Das Resultat ist ein „Zusammenhang zwischen den Halbparametern  $p_1$ ,  $p_2$  und den Bewegungskonstanten  $h_1$ ,  $h_2$ “ (siehe S.27 in [9]). Diese Größen sind aus dem zweiten Keplerschen Gesetz bekannt.

Die Bewegungskonstante  $h$  ist eine wichtige Größe, um die Umlaufzeit  $T$  und auch um die Mittlere Anomalie  $M(t)$  bzw. die Kreisfrequenz  $n$  dieser zu bestimmen. Um an diese Größe zu gelangen leitet man nach S.87 in [12] aus der Zentrifugalkraft und der Gravitationskraft für einen Sonderfall das dritte Keplersche Gesetz her. Man tut so als bewege sich ein Körper mit der Masse  $m_{Sat}$  auf einer Kreisbahn mit dem Radius  $a$ , was für eine (numerische und lineare) Exzentrizität von 0 zutrifft. Da der Körper nicht ins Schwerezentrum im Mittelpunkt der Bahn abstürzt, muss die Gravitationskraft gleich der Zentrifugalkraft sein. Es folgt

$$F_Z = F_G \quad (4.29)$$

$$m_{Sat} \cdot n^2 \cdot a = G \cdot \frac{m_{Sat} M_{Erde}}{a^2} \quad (4.30)$$

$$a^3 \left( \frac{2\pi}{T} \right)^2 = G M_{Erde} \quad (4.31)$$

Aus dieser Gleichung folgt durch ein paar geschickte Umformungen, die aus Platzgründen an dieser Stelle nicht niedergeschrieben wurden, und Umstellen nach  $T$

$$T = \frac{2 a^2 \pi}{\sqrt{G M_{Erde} a}} \quad (4.32)$$

Im Zähler steht die doppelte von der Verbindungsstrecke zwischen Schwerezentrum und Satellit in einer Umlaufperiode  $T$  überschrittenen Fläche. Durch die Verwandschaft der

obigen Gleichung mit Gleichung 4.25, kann man schlussfolgern, dass die Bewegungskonstante in diesem Fall sich zu

$$h = \sqrt{G M_{Erde} a} \quad (4.33)$$

ergibt. Die Berechnung der mittleren Anomalie  $M(t)$  stellt nun kein Problem mehr da.

### 4.1.2 Die Bahnelemente

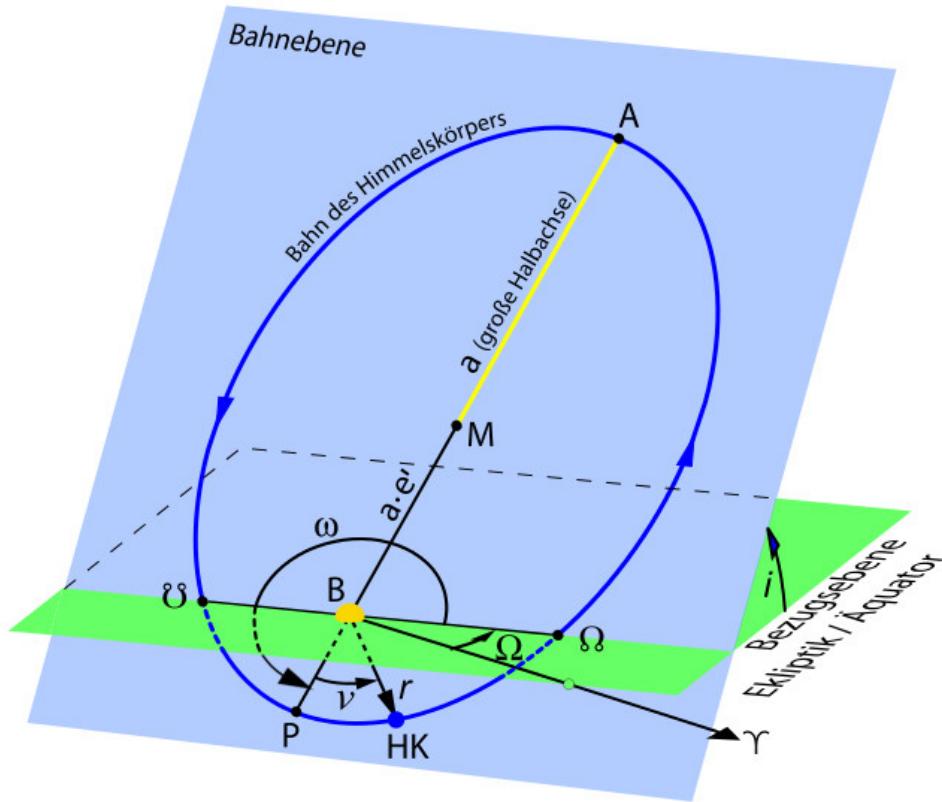


Abbildung 4.5: Bahnelemente, Quelle: [13], Grafik ist eigens modifiziert

Die Bahnelemente dienen der Beschreibung einer Bewegung eines Himmelskörpers auf einer Umlaufbahn (meist einer Ellipse). Dieser Körper unterliegt den Keplerschen Gesetzen. Wird die Bewegung eines Himmelskörpers durch äußere Einflüsse (z.B. Gravitationskraft der Sonne) nicht gestört, so kann sie durch sechs Größen beschrieben werden. Diese Größen sind die Bahnelemente. Zwei Bahnelemente beschreiben die **Form der Bahn**, drei legen die **Lage der Bahn** im dreidimensionalen Raum fest und ein Bahnelement gibt an zu welcher **Zeit** sich der Himmelskörper wo auf der Bahn befunden hat.

Diese Bahnelemente reichen in der Praxis nicht aus, um die Position eines Himmelskörpers z.B. eines Satelliten mit einem Vorhersagemodell berechnen zu können. Aus diesem

Grund werden die Bahnelemente meist um von Vorhersagemodellen benötigten Informationen ergänzt. Im Folgenden werden die Bahnelemente in Ihrer Bedeutung anhand der Abbildung 4.5 erläutert.

### Gestalt der Bahn

Um die Gestalt der Bahn zu beschreiben wird die **numerische Exzentrizität (1)**  $e'$  und die Angabe der Länge der **großen Halbachse (2)**  $a$  benötigt. Diese beiden Größen werden ausführlich im Absatz 4.1.1 beschrieben. Es braucht nicht mehr, um eine elliptische Bahn zu definieren. Zur Erinnerung: Die große Halbachse  $a$  ist die Länge der Strecke vom Mittelpunkt zu den Hauptscheiteln. Die (lineare) Exzentrizität  $e$  gibt die Schlankeit der Ellipse an. Wobei die numerische Exzentrizität  $e'$  auf die große Halbachse normiert ist.

Die Abhängigkeit der Umlaufzeit eines Satelliten von den Bahndimensionen (v.a. Länge der großen Halbachse ) und die elliptische Bahnform sind der „quadratischen Abnahme der Gravitationsbeschleunigung [geschuldet]“ (siehe S. 87 in [12]).

$$\ddot{\vec{r}} = -\frac{GM_{Erde}}{r^2} \cdot \frac{\vec{r}}{r} \quad (4.34)$$

$GM_{Erde} \approx 398600,4 \frac{\text{km}^3}{\text{s}^2}$  ist dabei das konstante Produkt aus Gravitationskonstante und Masse der Erde. Das Stichwort lautet hier Impulserhaltungssatz. In einem geschlossenen System kann Impuls weder vernichtet noch erschaffen werden. Bis zum Perigäumsdurchgang nimmt ein Satellit Impuls von der Erde über die Kraft, die zum Erdmittelpunkt hin wirkt auf. Danach wird Impuls an die Erde abgegeben → der Satellit verlangsamt sich. Nach dem Apogäumsdurchgang fängt der Satellit wieder an schneller zu werden. Er nimmt wieder Impuls auf.

### Lage der Bahn

Es erscheint sinnvoll, dass die Lage im Raum als nächstes festgelegt wird. Auch scheint es sinnig, dass man drei Bahnelmente benötigt, um die Lage der Bahn im dreidimensionalen Raum festzulegen.

Man stelle sich nun vor, dass die Ellipse, auf welcher der Satellit sich bewegt, in einer Ebene liege. Diese Ebene schneide man mit einer Referenzebene (z.B. Schnittebene durch den Erdäquator). Die **Inklination (3)**  $i$  ist nun ein Winkel, „um den die Bahnebene

gegenüber der [Referenzebene] geneigt ist“ (siehe S.88 in [12]). Um die Nebenachse der Ellipse herum kann durch diese Festlegung keine Rotation mehr stattfinden. Die Ellipse, die ja in der Ebene liegt, kann sich jetzt noch in dieser Ebene um eine zur Ebene orthogonalen Achse drehen. Um das zu unterbinden wird ein weiterer Winkel, das **Argument des Perigäums (4)**  $\omega$ , eingeführt. Das Argument des Perigäums ist der Winkel, den die Knotenlinie und die Perigäumsrichtung (Hauptachse) einschließen. Das Perigäum liegt am Ende der Hauptachse, auf welches sich der Satellit zu bewegt, nachdem er den aufsteigenden Knoten passiert hat. Der aufsteigende Knoten ist der Punkt in dem der Satellit die Referenzebene auf seiner elliptischen Umlaufbahn (von Süd nach Nord) durchstößt. Die Knotenlinie ist die Schnittgerade der Ellipse mit der Referenzebene. Darüber hinaus gibt es das Apogäum, welches nach dem absteigenden Knoten durchlaufen wird, der am anderen Ende der Nebenachse beheimatet ist. Eine andere Definition für das Perigäum ist der Punkt des geringsten Abstands des Satelliten zur Erde. Analog lässt sich das Apogäum als Punkt des größten Abstands des Satelliten zur Erde festlegen. Die Ellipse dreht sich nun nicht mehr in der Ebene. Eine Sache wurde jetzt noch nicht betrachtet. Die Ebene kann sich immer noch entlang Ihrer Hauptachse drehen. Die **Rektaszension des aufsteigenden Knotens (5)**  $\Omega$  gibt nun den Winkel an, den die Knotenlinie und der Vektor vom Ellipsenmittelpunkt zum Frühlingspunkt  $\gamma$  einschließen. Doch was ist der Frühlingspunkt? „Als Frühlingspunkt (auch Widderpunkt) wird in der Astronomie der Schnittpunkt des Himmelsäquators mit der Ekliptik bezeichnet, an dem die Sonne zum Frühlingsanfang der Nordhalbkugel (= Herbstanfang der Südhalbkugel) steht“ (siehe [14]). Die Derehung um die Hauptachse ist jetzt also auch nicht mehr möglich. Der Bahnlage wurden somit alle drei Freiheitsgrade genommen.

Die Bewegung eines Satelliten erfolgt in einer nicht veränderbaren Bahnebene mit dem Bahnmittelpunkt gleich dem Erdmittelpunkt. Dies impliziert das erste Keplersche Gesetz. Der Grund hierfür ist „dass die Anziehung der Erde (in erster Näherung) immer zum Erdmittelpunkt gerichtet ist“ (siehe S. 87 in [12]). Zu keinem Zeitpunkt gibt es somit einen Kraftvektor, der senkrecht zum Ortsvektor  $\vec{r}$  wirkt und somit die Bahnebene verschieben könnte (Annahme: Mittelpunkt des Koordinatensystems ist der Erdmittelpunkt). Daraus resultiert wiederum, dass eine einmal eingeschlagene Bahnebene ohne eine äußere Kraft nicht wieder verlassen werden kann.

## Zeitlicher Bezug und Kepler-Gleichung

Bis jetzt ist alles recht schön und anschaulich. Es ist klar wo die Bahnebene im Raum ist und wie diese aussieht. Wir haben durch das zweite Keplersche Gesetz eine Gleichung, die uns die Position des Satelliten in Abhängigkeit des Winkels  $\theta$  verrät. Diese ist

$$r = \frac{p}{1 + e' \cos(\theta)} \quad (4.35)$$

$\theta$  ist der Winkel, den die Verbindungsgeraden zwischen einem Brennpunkt und dem Satellit mit der Hauptachse einschließt. Es gibt allerdings noch ein Haar in der Suppe. Aufgrund des Flächensatzes ist die Winkelgeschwindigkeit des Satelliten nicht konstant (bzw. nur für eine in der Natur nicht vorkommende Kreisbahn konstant).  $\theta$  ist eine Funktion von  $t$ . Die sog. wahre Anomalie  $\theta(t)$  schwankt um die mittlere Anomalie  $M$  (nicht zu verwechseln mit der Erdmasse). Man stößt nun auf das Zweikörperproblem (Keplerproblem). Dies ist die Angabe der wahren Anomalie  $\theta(t = t_x)$  zu einem vorgegebenen Zeitpunkt  $t_x$ . Wie kommt die mittlere Anomalie nun zustande? Zunächst sollte ein genauer Blick auf Abbildung 4.6 geworfen werden. Die Abbildung stellt eine Momentaufnahme einer Situation dar. Wie schon bei früheren Betrachtungen wird die Ellipse (Orbit) in einen Umkreis

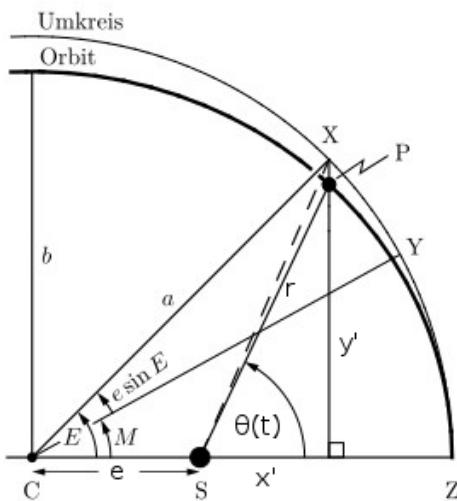


Abbildung 4.6: Kepler-Gleichung, Quelle: [15], Grafik ist eigens modifiziert

eingeschrieben. Ein fiktiver (mittlerer) Satellit  $Y$  laufe entlang des Umkreises um den Punkt  $C$  (Mittelpunkt der Ellipse und des Kreises, kein Ellipsenbrennpunkt). Der wahre Satellit  $P$  bewege sich entlang der Ellipse. Die Lage des fiktiven Satelliten  $Y$  wird zum Zeitpunkt  $t$  mittels des Winkels  $M$  dargestellt. Hierbei wird unterstellt, dass der Satellit

zum Zeitpunkt  $t_P$  im Perigäum (oft auch Periapsis genannt)  $M = 0$  war. Es gilt nun

$$M = 2\pi \frac{t - t_P}{T} = n(t - t_P) \quad (4.36)$$

Hierbei ist  $n$  die Winkelgeschwindigkeit und  $T$  die Umlaufzeit des Satelliten um das Schwerzentrum.  $S$  ist der Erdmittelpunkt (Schwerzentrum). Um nun die Kurve zu der Ellipse und dem tatsächlichen Satellit zu erhalten wird das zweite Keplersche Gesetz für diesen konkreten Fall herangezogen. In gleichen Zeiträumen werden gleiche Flächen von den Fahrstrahlen der Satelliten unabhängig von der Bahnform (egal ob elliptisch oder kreisförmig mit  $e' = e = 0$ ) überstrichen. Hier spricht man von einer bijektiven Abbildung der Ellipsen- und Kreisflächensegmente. Letztere sind leicht zu berechnen. Es wird also praktisch ein Umweg über den Umkreis gegangen. Das zweite Keplersche Gesetz angewandt auf den vorliegenden Fall, führt bezüglich der Zeitspanne einer Überstreichung der gesamten Fläche zu

$$\frac{A_{Kreis}}{A_{Ellipse}} = \frac{\pi a^2}{\pi a b} \quad (4.37)$$

Nach dem zweiten Keplerschen Gesetz ist dieses Verhältnis dasselbe für eine infinitesimale Fläche. So gilt

$$\frac{A_{CYZ}}{A_{SPZ}} = \frac{A_{Kreis}}{A_{Ellipse}} = \frac{a}{b} \quad (4.38)$$

Kepler führte als eine Hilfsgröße die exzentrische Anomalie  $E$  ein. Hierzu wird  $P$  auf den Hilfskreis projiziert. Es entsteht der Punkt  $X$ . Der Winkel zwischen Hauptachse und  $X$  ist die exzentrische Anomalie. Durch den in Gleichung 4.38 hergeleiteten Zusammenhang kann folgende Aussage getroffen werden.

$$A_{SXZ} = \frac{a}{b} A_{SPZ} \quad (4.39)$$

Setzt man Gleichung 4.38 in Gleichung 4.39 ein, so ergibt sich

$$A_{SXZ} = A_{CYZ} \quad (4.40)$$

Mit der Gleichung 4.40 ist ein indirekter Zusammenhang zwischen der mittleren Anomalie (Punkt  $Y$ ) und der exzentrischen Anomalie (Punkt  $X$ ) gefunden. Der Fahrstrahl  $\overline{CY}$  in der Zeit einer Periode  $U$  eine Fläche von  $\pi a^2$  überstreich, so überstreich er in einer Zeit  $t$  eine Fläche, die um den Faktor  $\frac{M}{2\pi}$  kleiner ist. Es folgt

$$A_{CYZ} = \frac{M}{2\pi} \pi a^2 = \frac{a^2}{2} M \quad (4.41)$$

Analog gilt dies für den Fahrstrahl  $\overline{CX}$ .

$$A_{CXZ} = \frac{a^2}{2} E \quad (4.42)$$

Weiter geht's mit noch mehr Geometrie. Die Fläche  $A_{CXZ}$  kann in zwei Teilflächen zerlegt werden.

$$A_{CXZ} = A_{CSX} + A_{SXZ} \quad (4.43)$$

Die Fläche  $A_{CSX}$  kann als die Fläche eines Dreiecks aufgefasst werden. Dabei ist  $a \cdot e'$  die Basis und  $a \sin(E)$  die Höhe des Dreiecks.

$$A_{CSX} = \frac{1}{2} \cdot a \cdot e' \cdot a \sin(E) = \frac{a^2}{2} e' \sin(E) \quad (4.44)$$

Als letzten Schritt setze man nun noch Gleichung 4.40 (unter Beachtung von Gleichung 4.41), Gleichung 4.42 und Gleichung 4.44 in Gleichung 4.43 ein.

$$\frac{a^2}{2} E = \frac{a^2}{2} e' \sin(E) + \frac{a^2}{2} M \quad (4.45)$$

Mit ein paar Tricks aus der Grundschule entsteht die **Kepler-Gleichung** wie sie im Buche steht.

$$E - e' \sin(E) = M = n \cdot (t - t_P) \quad (4.46)$$

Diese Gleichung kann nicht nach  $E$  aufgelöst werden und so keine schöne Funktion  $E(t)$  gewonnen werden. Doch Gott sei dank, schenke er Newton die Eingebung des Newton-Verfahrens zur Nullstellensuche. Das Newton-Verfahren ist im Grunde eine rekursive Folge, wobei jedes Folgeglied allgemein mit

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (4.47)$$

zu bestimmen ist. Man stellt also Gleichung 4.46 entsprechend um und leite ab.

$$f(E) = E - e' \sin(E) - M \stackrel{!}{=} 0 \quad (4.48)$$

$$f'(E) = 1 - e' \cos(E) \quad (4.49)$$

Es folgt die rekursive Folge für die Kepler-Gleichung.

$$E_{n+1} = E_n - \frac{E_n - e' \sin(E_n) - M}{1 - e' \cos(E_n)} \quad (4.50)$$

Als Startwert wird in [12]  $E_0 = M$  empfohlen. Die rekursive Folge Verhält sich dabei so, dass sie gegen einen entsprechenden Wert für die exzentrische Anomalie  $E$  konvergiert. Wenn sich entsprechende Nachkommastellen nach gewisser Genauigkeit nicht mehr

ändern, so hat man die Nullstelle gefunden. Mit diesem Wert für  $E(t = t_x)$  kann die wahre Anomalie  $\theta(t = t_x)$  bestimmt werden. Zu beachten ist, dass in die Kepler-Gleichung natürlich vor der Anwendung des Newton-Verfahrens auch der gewünschte Zeitpunkt  $t_p$  eingesetzt wird.

Wie bestimmt man nun die wahre Anomalie, also den Winkel, der zu dem Zeitpunkt gehört aus welchem wir dann wiederum einen Vektor  $r$  gewinnen können? Hierzu werden die Größen  $x'$  und  $y'$  in Abbildung 4.6 herangezogen. Man sucht nun nach einer Beziehung zwischen diesen Größen und bekannten Größen aus dem Fundus der Bahnelemente die Form der Bahn betreffend (dazu gehören natürlich auch abgeleitete Größen). Es ist nachvollziehbar, dass

$$x' = r \cos(\theta) = \cos(E) a - e = a [\cos(E) - e'] \quad (4.51)$$

ist. Für  $y'$  muss man etwas tiefer in die Trickkiste greifen. Man geht aus von Gleichung 4.4. Diese stelle man nach  $p$  um.

$$p = p' \frac{b}{a} \quad (4.52)$$

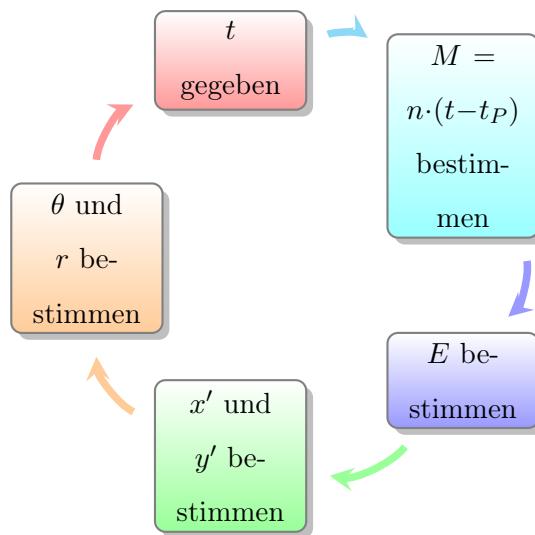
Für  $p'$  kann der Term  $\sin(E) a$  eingesetzt werden. Mit dem Satz des Pythagoras kann  $b$  auch mit den Größen  $a$  und  $e$  ausgedrückt werden. Es folgt

$$p = y' = \sin(E) a \frac{\sqrt{a^2 - e^2}}{a} = \sin(E) a \sqrt{1 - e'^2} \quad (4.53)$$

Aus diesen beiden Größen kann unmittelbar die wahre Anomalie  $\theta$  zu einem gegebenen Zeitpunkt bestimmt werden.

$$\theta = \arctan \left( \frac{y'}{x'} \right) = \arctan \left( \frac{\sin(E) a \sqrt{1 - e'^2}}{a (\cos(E) - e')} \right) = \arctan \left( \frac{\sin(E) \sqrt{1 - e'^2}}{\cos(E) - e'} \right) \quad (4.54)$$

Nun ist also der zeitliche Bezug bekannt. Hier nochmal eine kleine Darstellung wie vorzugehen ist.



Führt man den Vorgang wie oben gezeigt iterativ aus, so erhält man einen zweidimensionalen Vektor  $\vec{r}$ . Wenn man nun scharf nachdenkt, so fällt einem auf, dass man nochmal eine Information benötigt. Es braucht einen Zeitpunkt  $t_P$ . Dies muss ein Zeitpunkt sein, bei welchem der Satelliten im Perigäum stand. Als Analogie könnte man die Phase eines Signals in der Elektrotechnik angeben. Genau diesen Zeitpunkt liefern die Bahnelemente als zeitlichen Bezug.

Als Zusammenfassung hier nochmal eine Übersicht aller Bahnelemente.

- Gestalt der Bahn
  - die große Halbachse  $a$
  - die numerische Exzentrizität  $e'$
- Lage der Bahn
  - die Inkination  $i$
  - die Rektaszension des aufsteigenden Knotens  $\Omega$
  - das Argument des Perigäums  $\omega$
- Zeitlicher Bezug
  - die Epoche (wann ist der Sattelit im Perigäum gewesen)  $t_P$

### 4.1.3 Von den Bahnelementen zum Vektor

Im nächsten Schritt trägt man alle Informationen aus den Bahnelementen zusammen, um einen Vektor  $\vec{r}$  im dreidimensionalen kartesischen Koordinatensystem zu erhalten. Der Ursprung des Koordinatensystems soll im Erdzentrum liegen und der Vektor  $\vec{r}$  soll ein Ortsvektor des Satelliten sein. Die x-y-Ebene dieses Koordinatensystems ist der Äquator. Das Ergebnis der bisherigen Berechnungen ist der Vektor

$$\vec{r}' = a \cdot \begin{pmatrix} \cos(E) - e' \\ \sin(E) \sqrt{1 - e'^2} \\ 0 \end{pmatrix} \quad (4.55)$$

Die Satellitenbahn liegt nach der obigen Gleichung in der äquatorialen Ebene. Der Erdmittelpunkt liegt im Brennpunkt der Ellipse. Anstatt die Bahnebene zu drehen. Dreht man nun das Koordinatensystem. Allerdings in die mathematisch entgegengesetzte Richtung. Hierfür können natürlich nur die Winkel verwendet werden, welche in den Bahnelementen für die Lage der Bahn mitgeliefert werden. Bei der Drehung wird auf Drehmatrizen zurückgegriffen. Eine gute Veranschaulichung dieser Drehungen findet man *in dieser Animation*.

Der gesuchte Vektor ergibt sich nun zu

$$\vec{r} = \mathbf{R}_Z(-\Omega) \mathbf{R}_X(-i) \mathbf{R}_Z(-\omega) \cdot a \cdot \begin{pmatrix} \cos(E) - e' \\ \sin(E) \sqrt{1 - e'^2} \\ 0 \end{pmatrix} \quad (4.56)$$

Hierbei stellen  $\mathbf{R}_X$  und  $\mathbf{R}_Z$  folgende Drehmatrizen dar (nachzuschlagen auf S.89 in [12]).

$$\mathbf{R}_X(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \quad (4.57)$$

$$\mathbf{R}_Z(\alpha) = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.58)$$

Eine Frage, die sich aufdrängen mag ist: Warum kann die Drehung der Ellipse in der Bahnebene (Argument des Perigäums  $\omega$ ) einfach auf eine Drehung der z-Achse zurückgeführt werden? Man kann es sich auch so vorstellen, dass zunächst die Ellipse noch in der äquatorialen Ebene liegt, die z-Achse dann um  $-\omega$  gedreht wird, dann die x-Achse um

$-i$  und letztendlich die z-Achse nochmal um  $-\Omega$  gedreht wird.

Nach [12] ergibt sich nach der Ausführung der Drehungen der Vektor

$$\vec{r} = r \cdot \begin{pmatrix} \cos(\omega + \theta) \cos(\Omega) - \sin(\omega + \theta) \cos(i) \sin(\Omega) \\ \cos(\omega + \theta) \sin(\Omega) + \sin(\omega + \theta) \cos(i) \cos(\Omega) \\ \sin(\omega + \theta) \sin(i) \end{pmatrix} \quad (4.59)$$

Durch Ableitung nach der Zeit kommt man wie in [12] angegeben auf

$$\vec{v} = \mathbf{R}_Z(-\Omega) \mathbf{R}_X(-i) \mathbf{R}_Z(-\omega) \cdot \frac{\sqrt{GM_{Erde}} a}{r} \cdot \begin{pmatrix} -\sin(E) \\ \cos(E) \sqrt{1 - e'^2} \\ 0 \end{pmatrix} \quad (4.60)$$

Gleichung 4.59 lässt sich so interpretieren, dass der Term  $\sqrt{\frac{GM_{Erde}}{r^2 a}}$  die zeitliche Ableitung der exzentrischen Anomalie  $E$  sein muss, da die Kettenregel der Differentialrechnung greift und die Rotationsmatrizen keine zeitlichen Abhängigkeiten aufweisen.

Mit den Gleichungen 4.59 und 4.60 lässt sich nun die Antenne fast ausrichten und nachführen. Der nächste Schritt ist die Berücksichtigung der Position der Bodenstation auf der Erde. Um sich das Leben nicht unnötig schwer zu machen, verwendet geografische Koordinaten und gewinnt daraus einen Ortsvektor für das kartesische Koordinatensystem im Erdmittelpunkt. Es wird hierbei vernachlässigt, dass die Erde keine Kugel sondern annähernd ein Ellipsoid ist. Ganz korrekt würde man es machen, wenn man sich des World Geodetic System 1984 (WGS84) bedient. Laut [16] können durch diese Vernachlässigung Ortsverschiebungen (der Bodenstation) von bis zu 20km auftreten. Es soll jedoch zunächst möglichst einfach verstanden werden, wie ein Programm wie GPredict vorgeht und auch zur Überprüfung der Daten, die GPredict liefert reicht die Genauigkeit aus, die mit dieser Vernachlässigung erreicht wird. Im ersten Schritt zur **Beschreibung der Bahn im erdfesten System** muss das Koordinatensystem nochmal an der z-Achse gedreht werden. Das liegt daran, dass die x-y-Ebene zwar im Äquator liegt, die x-Achse jedoch nicht auf den nullten Längengrad, sondern auf den Frühlingspunkt zeigt. Bei dieser Achsentransformation wird auch die Rotation der Erde mit berücksichtigt. Die x-Achse wird in den sog. Meridian von Greenwich gezogen. Die Drehung der z-Achse wird mit dem Winkel  $\Theta$  vollzogen (siehe Abbildung 4.7. Dieser Winkel wird dabei als „Sternenzeit“ bezeichnet und häufig im Stundenmaß (1h entspricht  $15^\circ$ ) ausgedrückt“ (siehe S. 89 in [12]). Für diese

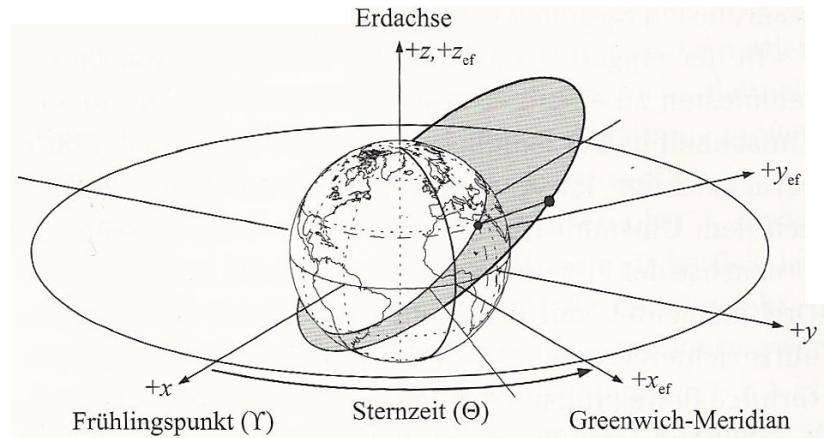


Abbildung 4.7: Sternzeit, Quelle:S.89 in [12]

einfacheren Angelegenheiten kann so in [12] beschrieben die Sternzeit nach folgender Gleichung eruiert werden.

$$\Theta = 280,4606^\circ + 360,9856473^\circ \cdot d \quad (4.61)$$

In dieser Gleichung stellt  $d$  die Tage seit dem 1. Januar 2000 dar. Mit gegebener UT1 kann so die genaue Sternzeit berechnet werden. Mit der Sternzeit ergibt sich nun die letzte Drehtransformation.

$$\vec{r}_{Greenwich} = \mathbf{R}_Z(\Theta) \cdot \vec{r} \quad (4.62)$$

Dadurch ist die erdfeste Position des Satelliten aus den inertialen Koordinaten bestimmt worden. Nach [12] ergibt sich die erste zeitliche Ableitung zu

$$\vec{v}_{Greenwich} = \mathbf{R}_Z(\Theta) \cdot \vec{v} - \begin{pmatrix} 0 \\ 0 \\ \omega_{Erde} \end{pmatrix} \times \vec{r}_{Greenwich} \quad (4.63)$$

Die Interpretation von Gleichung 4.63 ist, dass sich die Geschwindigkeit  $\vec{v}_{Greenwich}$  im erdfesten System vom inertialen System durch einen Term unterscheidet, der von der Drehgeschwindigkeit  $\omega_{Erde} = 7,29212 \cdot 10^{-5} \frac{rad}{s}$  und dem Abstand des Satelliten von der Erde abhängt.

Liegt eine Bodenstation von der die Funkverbindung zum Satellit aufgebaut werden soll nicht zufällig auf dem Schnittpunkt zwischen dem nullten Längengrad und dem Satellitenortsvektor, so muss der Vektor  $\vec{r}_{Greenwich}$  für die Antenne selbstverständlich angepasst werden. Das grobe Vorgehen ist aus der analytischen Geometrie bekannt. Der Vektor

$\vec{r}_{Greenwich}$  stellt den Ortsvektor zum Satelliten dar. Der Vektor  $\vec{r}_{Bodenstation}$  stellt den Ortsvektor zur Bodenstation dar. Um einen Vektor von der Bodenstation zum Satellit zu erhalten rechne man die Spitze minus den Anfang.

$$\vec{r}_{Bodenstation, Satellit} = \vec{r}_{Satellit} - \vec{r}_{Bodenstation} \quad (4.64)$$

wobei

$$\vec{r}_{Satellit} = \vec{r}_{Greenwich}$$

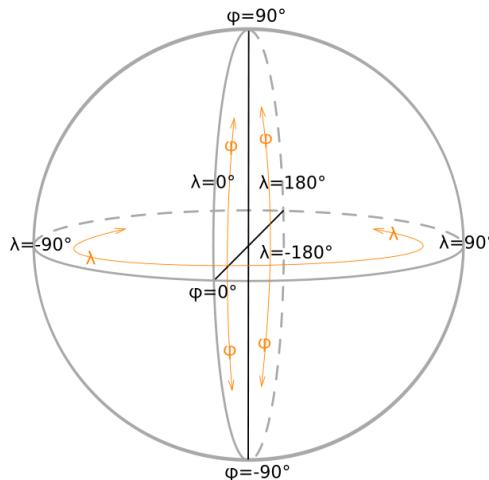


Abbildung 4.8: Geografische Koordinaten, Quelle: [16]

Für die Berechnung des Vektors  $\vec{r}_{Bodenstation}$  werfe man zunächst einen Blick auf das dem populären Kugelkoordinatensystem nicht ganz unähnlichen geografischen Koordinatensystem (Abbildung 4.8). Die Koordinaten der Bodenstation im geografischen Koordinatensystem lassen sich ganz leicht ohne vor Ort zu sein mittels *OpenStreetMap* herausfinden. Dabei entspricht der erste Wert dem Breiten- der zweite dem Längengrad. Der mittlere Erddurchmesser  $d_{Erde}$  beträgt nach [17] in etwa 12.742km.

Man stelle sich nun die Bodenstation als einen Punkt auf der Kugeloberfläche vor. Schneidet man entlang des Längengrades der Bodenstation durch die Kugel, so lässt sich mit einfacher Trigonometrie die Information des Breitengrades in das kartesische Koordinatensystem übertragen. Die z-Komponente des gesuchten Ortsvektors  $\vec{r}_{Bodenstation}$  ist der Sinus des Breitengrades multipliziert mit dem halben Erddurchmesser. Die Ankathete des Dreiecks, dass sich jetzt ausgebildet hat spielt bei der Betrachtung von x- und y-Komponente eine wichtige Rolle. Diese Ankathete hat die Länge des Kosinus des Breitengrads multipliziert mit dem halben Erddurchmesser. Um die x- und die y-Komponente

zu ermitteln schneidet man entlang des Äquators durch die Erdkugel hindurch und wirft einen Blick auf die Schnittfläche. Die eben betonte Ankathete definiert mit der x-Achse ein weiteres rechtwinkligen Dreieck. Die Ankathete des ersten Schnitts wurde zur Hypotenuse dieses Dreiecks. Über den Längengrad und die bekannte Länge der Hypotenuse können nun x- und y-Komponente leicht bestimmt werden. Die x-Komponente folgt aus dem Kosinus des Längengrades multipliziert mit der Hypotenuse. Die y-Komponente ist eine Folge der Multiplikation des Sinus des Längengrades mit der Hypotenuse. Aus diesen einfachen und vom Kugelkoordinatensystem bekannten Denkansätzen folgt der Ortsvektor der Bodenstation.

$$\vec{r}_{Bodenstation} = \frac{d_{Erde}}{2} \begin{pmatrix} \cos(\text{latitude}) \cos(\text{longitude}) \\ \cos(\text{latitude}) \sin(\text{longitude}) \\ \sin(\text{latitude}) \end{pmatrix} \quad (4.65)$$

Mit diesem Vektor ist das letzte Fragment gefunden, um die Antenne über die beschriebene Vorgehensweise auszurichten und nachzufahren. Es sei darauf hingewiesen, dass die hier beschriebene Vorehensweise nicht mit einem Bahnmodell wie dem „Simplified General Perturbations Model No. 4“ (SGP4) mithalten kann vor allem, da bis zu diesem Punkt noch keine Bahnstörungen berücksichtigt wurden. Dieses Kapitel dient lediglich dem Verständnis.

#### 4.1.4 Bahnmodelle

Eines der populärsten Bahnmodelle ist das **SGP4**. Das Bahnmodell wurde einst für das „North American Aerospace Defense Command“ (NORAD) entwickelt. Diese Institution hat u.a. die Aufgabe der Überwachung erdnaher Flugkörper. NORAD bestimmt in regelmäßigen Zeitabständen die Bahnelemente aller katalogisierten Satelliten. Diese Informationen sind in einem bestimmten Rahmen auch zivilen Nutzern zugänglich (z.B. über *Celestrak*). Hiervon profitieren eben solche Amateurfunk-Bodenstationen wie die der DHBW Friedrichshafen. Die Distribution der NORAD-Bahnelemente „erfolgt in einem zweizeiligen Datenformat, was zu dem Beinamen TLE geführt hat“ (siehe S. 95 in [12]). Das Format der TLE ist in Abbildung 4.9 gut beschrieben. Es ist zu beachten, dass es sich bei den Bahnelementen, welche von NORAD geliefert werden um mittlere Bahnelemente handelt. Diese sind im Gegensatz zu den tatsächlichen (oskulierenden) Bahnelementen nicht für eine Auswertung mit dem oben beschriebenen Keplerschen Bahnmodell

verwendbar, sondern nur für die Verwendung in dem analytischen Modell mit welchem sie generiert wurden verwertbar. Das SGP4 entstand um 1970 und basiert auf einem analyti-

Zeile	Spalten	Erläuterung	Beispiel	Bedeutung
1	1	Zeilennummer (= 1)	1	1. Zeile
	3-7	NORAD-Katalognummer (Beispiel 16609)	25554	ISS (Zarya-Modul)
	8	Klassifizierung	U	nicht klassifiziert
	10-17	Internationale COSPAR-Satellitennummer (yynnnaaa), bestehend aus Jahr (yy), Startnummer (nnn) und Objekt (aaa)	98067A	Erstes Katalogobjekt zum 67. Start des Jahres 1998
	19-20	Epoche der Bahnelemente (Jahr)	06	2006
	21-32	Epoche Tag und Tagesbruchteile (UTC)	287.64456019	14. Oktober, 15:28:10
	34-43	Änderungsrate der mittleren Bewegung (in [rev/d <sup>2</sup> ]) oder ballistischer Koeffizient B	.00008182	0,00008182
	45-52	Änderungsrate (2. Ordnung) der mittleren Bewegung (in [rev/d <sup>3</sup> ]) (Dezimalpunkt zwischen Spalten 45 und 46; Exponent in Spalte 51-52)	00000-0	0,0
	54-61	Luftwiderstandsterm B* (in [1/R <sub>⊕</sub> ]); (Dezimalpunkt zwischen Spalten 54 und 55; Exponent in Spalte 60-61)	53355-4	0,53355 · 10 <sup>-4</sup>
	63	Bahnmodell	0	SGP4-Modell
	65-68	Laufende Nummer des Datensatzes	860	860
	69	Prüfsumme der Zeichenwerten (modulo 10)	9	9
2	1	Zeilennummer (= 2)	2	2. Zeile
	3-7	NORAD-Katalognummer (Beispiel 16609)	25554	ISS (Zarya-Modul)
	9-16	Bahnneigung (in [°])	051.6354	51,6354°
	18-25	Rektaszension des aufsteigenden Knotens (in [°])	292.0281	292,0281°
	27-33	Exzentrizität (Dezimalpunkt zwischen Spalten 26 und 27)	0013277	0,0013277
	35-42	Argument des Perigäums (in [°])	096.0881	96,0881°
	44-51	Mittlere Anomalie (in [°])	057.9543	057,9543°
	53-63	Mittlere Bewegung (in [rev/d])	15.76874518	15,76874518°/d
	64-68	Umlaufsnummer	45200	45200
	69	Prüfsumme der Zeichenwerten (modulo 10)	8	8

Abbildung 4.9: TLE, Quelle: S.96 in [12]

schen Model von Lane und Crawford. Deren Modell verband zu deren Zeit die gravitativen Störungen mit einem analytischen Modell zur Beschreibung des Luftwiderstands. Für den operationellen Einsatz haben Lane und Hoots die vollständige Theorie vereinfacht und so in die SGP4 überführt.

Aufer dem SGP4 verwendet GPredict auch das „Simplified Deep Space Pertubations Model No. 4“ (SDP4). „Es erfasst zusätzlich Störungen durch die Gravitationskraft von Sonne und Mond, sowie [Informationen], die bei Bahnen mit einer Umlaufzeit von 12 h (GPS)

und 24 h ([bei ]geostationäre[n] Satelliten) von Bedeutung sind“ (siehe S. 96 in [12]). Das SDP4 kommt bei Satelliten mit einer Umlaufzeit von mehr als 225 Minuten zum Zug. Derartige Satelliten bewegen sich auf sog. „Deep Space“-Bahnen.

Aus Abbildung 4.10 ist ein Gefühl für die Genauigkeit des SGP4 entnehmbar. Nahe an der Epoche liegt die Abweichung im Bereich von wenigen Kilometern. Die Abweichung kommt von mehreren kurzperiodischen Störungen, die das Bahnmodell unterschlägt. Innerhalb von mehreren Tagen nimmt die Abweichung mit quadratischer Abhängigkeit zu. Laut [12] ist der Grund dieses Fehlers die nicht vorhersagbare Atmosphärendichte. Bei höhren Flugbahnen sind die Fehler deswegen allgemein deutlich kleiner. Eine ziehmlich genaue Bahnprediktion über mehrere Wochen ist so, it nicht unrealistisch. Als Fazit lässt sich sagen, dass die Bahnelemente mit den GPredict arbeitet regelmäßig aktualisiert werden sollten, um eine optimale Leistungsfähigkeit der Bodenstation zu erreichen.

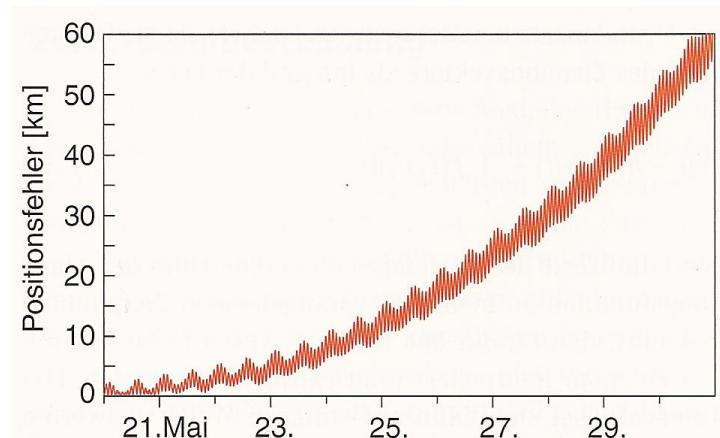


Abbildung 4.10: Vorhersagegenauigkeit des Satelliten CHAMP im Mai 2001, Quelle: S.96 in [12]

## 4.2 Doppler-Effekt

*Der folgende Abschnitt entstammt der Seminararbeit der Autoren eben dieser Arbeit vom 16. November 2015 aus der Vorlesung Radartechnik gelesen von Jürgen Göbel und der DHBW Friedrichshafen.*

Ein Beobachter, der sich relativ zu einem Wellensender bewegt, registriert eine andere Frequenz als die von der Quelle gesendete Frequenz. Bei der Näherung von Sender und

Empfänger scheint die Frequenz erhöht und bei einer Entfernung verkleinert. Dieses Phänomen wurde 1842 von C.J.Doppler entdeckt.

Um ein grundlegendes physikalisches Verständnis für den Doppler-Effekt zu entwickeln, gehe man zunächst von einem akustischen Beispiel aus: Ein Rennwagen fahre auf einem Beobachter mit der Geschwindigkeit  $v_R = 70 \frac{m}{s}$  zu. Der Beobachter habe dabei keine Eigengeschwindigkeit  $v_B = 0$  relativ zum Bezugssystem Rennstrecke. Im Spektrum des vom Auto emittierten akustischen Signals sei ein Sinussignal der Frequenz  $f_R = 1kHz$  enthalten. Aus Sicht des Rennwagenfahrers hat das emittierte Signal die Wellenlänge  $\lambda_R = \frac{c_S}{f_R} = \frac{340 \frac{m}{s}}{1kHz} = 0,34m$ . Aus Sicht des Beobachters sieht das Ganze etwas anders aus. Man nehme an, der Rennwagen emittiere zunächst einen Wellenberg. Bis der zweite Wellenberg emittiert wird ist der Rennwagen allerdings schon ein Stück  $\delta s_R$  auf den Beobachter zu gefahren. Diese Bewegung relativ zum Beobachter hat zur Folge, dass sich die Wellenlänge um genau dieses Stück verkürzt bzw. verlängert, wenn der Rennwagen sich vom Beobachter fortbewegt. Die Wellenlänge aus Sicht des Beobachters ist nun wie folgt zu beschreiben.

$$\lambda_B = \lambda_R - \delta s_R = \lambda_R - v_R \cdot T_R = \lambda_R - \frac{v_R}{f_R} \quad (4.66)$$

Mit den gegebenen Werten ergibt sich eine neue Wellenlänge  $\lambda_B = 0,27m$ . Da  $\lambda_B \sim \frac{1}{f_B}$  ergibt sich auch eine neue Frequenz von  $f_B \approx 1260Hz$ . Nun geht man noch ein Stück weiter, so dass man eine Funktion  $f_B(f_R, v_R)$  erhält. Hierzu wird Gleichung 4.66 herangezogen. Es muss nur noch die Beziehung  $\lambda = \frac{c_S}{f}$  einmal für die Wellenlänge des Rennwagens und einmal für die Wellenlänge des Beobachters eingesetzt werden. Dann ist die Gleichung nach  $f_B$  aufzulösen. Das Ergebnis ist dann

$$f_B = \frac{f_R}{1 - \frac{v_R}{c_S}} \quad (4.67)$$

Entfernt sich der Rennwagen ist für die Geschwindigkeit des selbigen einfach ein negativer Wert einzusetzen. **Achtung:** Soll der Doppler-Effekt auf eine sich in einem Medium fortbewegende Welle angewandt werden, ist von den Gleichungen her immer zwischen bewegtem Sender, bewegtem Empfänger und bewegtem Sender, sowie bewegtem Empfänger zu unterscheiden. Die oben aufgeführten Gleichungen beziehen sich ausschließlich auf den Fall, dass sich der Sender bewegt.

Elektromagnetische Wellen brauchen wie seit dem 20 Jahrhundert allgemein bekannt kein

Medium, um sich fortzupflanzen. Bewegen sich Sender oder Empfänger relativ zueinander, so kommt es auch bei elektromagnetischen Wellen zum Dopplereffekt. Aufgrund dessen, dass die Ausbreitung nicht auf das Medium angewiesen ist, spielt es keine Rolle, ob sich Empfänger, Sender oder beide bewegen. Die Frequenzverschiebung ist nur noch von der Relativgeschwindigkeit abhängig. Das Relativitätsprinzip besagt nun, dass sich jeder Übertragungsteilnehmer als ruhend betrachten darf. (Aus physikalischer Sicht kann man tatsächlich behaupten, dass jemand einem vor der Diskothek „in die Faust gelaufen“ ist). Dank diesen Prinzips ist der obige akustische Fall (siehe Gleichung 4.67) universell direkt auf eine Übertragung mit elektromagnetischen Wellen umsetzbar.

Das ist allerdings nur die halbe Wahrheit. Es gibt leider keine absolute Zeit. Die Gleichzeitigkeit von räumlich getrennten Ereignissen ist eine Frage der Relativgeschwindigkeit des Beobachters.

Der relativistische Effekt der Zeitdilatation ist beim Doppler-Effekt mitzubetrachten, da die Periodendauer des emittierten Signals aus Sicht des Beobachters auf der Erde aufgedehnt wird. Somit wird die Wellenlänge des Sendesignals  $\lambda_S$  größer und die Frequenz des selbigen  $f_S$  kleiner. Mit dieser Erkenntnis folgt nun die tatsächliche Gleichung für den Dopplereffekt.

$$f_E = \frac{f_S}{1 - \frac{v_r}{c_L}} \cdot \frac{1}{\gamma} = \frac{f_S \sqrt{1 - \left(\frac{v}{c_L}\right)^2}}{1 - \frac{v_r}{c_L}} \quad (4.68)$$

Hierbei ist  $f_E$  die vom Empfänger empfangene Frequenz,  $v_r$  die radiale Relativgeschwindigkeit von Sender und Empfänger und  $c_L$  die Lichtgeschwindigkeit. Die tangentialen Anteile der Bewegung interessieren den Beobachter im nicht relativistischen Teil der Gleichung natürlich nicht. Es sind nur die radialen Anteile für den Doppler-Effekt relevant. Filtert man also den tangentialen Anteil mittels des Kosinus aus der Relativgeschwindigkeit aus, so kann man für die Geschwindigkeit einfach die betragsmäßige Relativgeschwindigkeit einsetzen.

$$f_E = \frac{f_S \sqrt{1 - \left(\frac{v}{c_L}\right)^2}}{1 - \frac{v}{c_L} \cdot \cos(\alpha)} \quad (4.69)$$

Bewegt sich nun der Sender rein tangential zum Empfänger würde man keinen Doppler-Effekt erwarten. Aufgrund der Relativgeschwindigkeit von Sender und Empfänger tritt jedoch die Zeitdilatation nach wie vor auf. Man spricht in diesem Fall vom transversalen Doppler-Effekt.

# 5 GPredict

Eine sehr detaillierte Beschreibung der Software GPredict ist unter [18] in englischer Sprache verfügbar. Dieses Kapitel stellt eine kompakte Beschreibung mit ausschließlich für dieses Projekt relevanten Informationen dar. Insofern können gewisse Parallelen nicht ausgeschlossen werden, wobei darauf hingewiesen sei, dass der Mehrwert dieser Zusammenfassung in der Kürze und Relevanz im Vergleich zum englischen Pendant liegt.

## 5.1 Übersicht

GPredict ist eine freie Software zur Satellitenverfolgung und Orbitvorhersage und steht als Quellcode oder bereits fertig kompiliertes Programm für Windows, Mac OS und Linux zur Verfügung. Die Software ist in C geschrieben und unter der GNU General Public License (GPL) lizenziert. Somit kann sie frei verändert und an die entsprechenden Nutzervoraussetzungen angepasst werden. Im Rahmen dieser Projektarbeit wird Version 1.3 von GPredict verwendet (verfügbar unter [19]).

In Abbildung 5.1 ist das Prinzip eines Satellitenverfolgungsprogramms zu sehen (die blauen Blöcke stellen hierbei die Funktionalität des Programms dar). Zunächst wird an Hand der Keplerschen Bahnelemente und dem aktuellen Zeitpunkt die absolute Position des Satelliten berechnet. Daraufhin wird der Vektor, der von der Bodenstation zum Satelliten zeigt, bestimmt. Nun können Azimut und Elevation dieses Vektors für die Ansteuerung der Antenne verwendet werden.

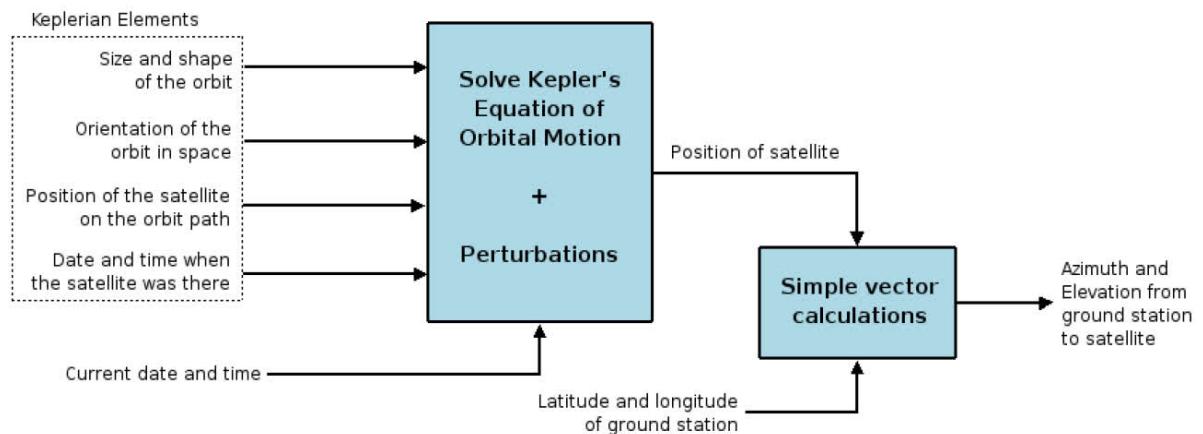


Abbildung 5.1: Prinzip eines Satellitenverfolgungsprogramms, Quelle: [18]

Zur Berechnung der Satellitenposition wird auf den NORAD SGP4/SDP4 Algorithmus zurückgegriffen. Um hierfür zu jedem Zeitpunkt die aktuellen Kepler-Elemente des zu verfolgenden Satelliten zu kennen, gibt es unter GPredict die Möglichkeit einer automatischen Aktualisierung über HTTP, FTP oder aus dem lokalen Verzeichnis.

Bei GPredict ist im Gegensatz zu anderen Satellitenverfolgungsprogrammen wie SatPC32 kein Limit an zu verfolgenden Satelliten und Bodenstationen gegeben. Durch die Verwendung von Modulen kann außerdem unkompliziert zwischen verschiedenen Konfigurationen gewechselt werden. Die Orbitvorhersage eines Satelliten lässt sich sowohl grafisch als auch tabellarisch darstellen, wobei durch die Einstellungen verschiedenster Parameter eine sehr individuelle Anzeige erreicht werden kann.

## 5.2 Grafische Oberfläche

In Abbildung 5.2 ist die grafische Oberfläche von GPredict zu sehen. In der Standardkonfiguration ist dort zunächst die Kartenansicht bzw. *Map View* (oben), die Polarisicht bzw. *Polar View* (links unten) und die Einzelsatellitenansicht bzw. *Single-Satellite View* (rechts unten) zu sehen.

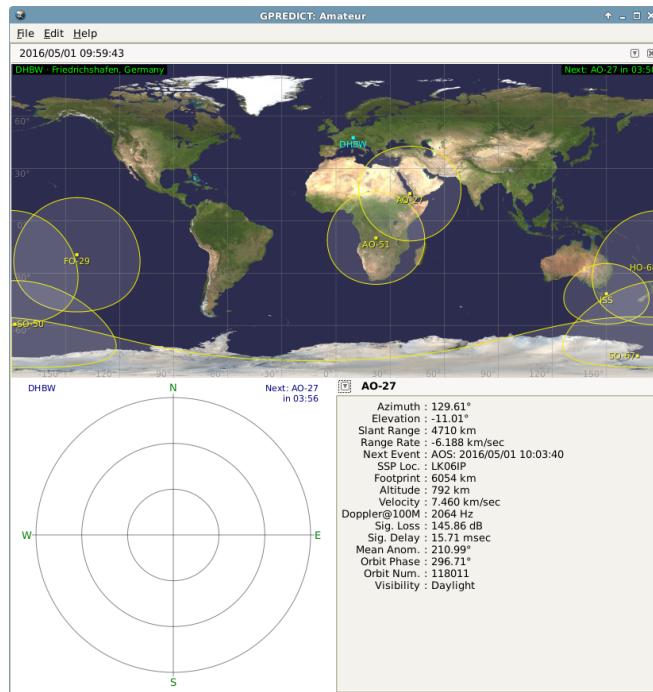


Abbildung 5.2: Standardoberfläche von GPredict

### 5.2.1 Grundansichten

Zu den oben genannten Ansichten kommen noch zwei Weitere hinzu, die Listenansicht bzw. *List View* und eine Ansicht für bevorstehende Durchläufe, die sogenannte *Upcoming Passes View*. Im Folgenden werden die verschiedenen Ansichten genauer beschrieben:

#### Map View

Diese Ansicht besteht, wie in Abbildung 5.2 zu sehen, aus einer Weltkarte auf der die aktuellen Standorte der für das aktuelle Modul ausgewählten Satelliten zu sehen ist. Das heißt der Punkt auf dem der entsprechende Satellit senkrecht bezogen auf den Erdmittelpunkt steht. Außerdem ist um diesen Punkt die Fläche umrahmt, von der der Satellit von der Erde aus sichtbar ist. Mit einem Rechtsklick auf einen Satellitennamen kann außerdem die Option *Ground Track* aktiviert werden, mit welcher die Spur des Satelliten für mehrere Orbits angezeigt wird.

#### Polar View

Die *Polar View* (siehe Abbildung 5.2) stellt eine Draufsicht auf die Bodenstation dar, bei der die Polarachse den Azimutwinkel darstellt und die Radialachse den Elevationswinkel. Mit einem Rechtsklick auf einen Satelliten lässt sich mit der Option *Show sky track* aktivieren, dass die Spur des entsprechenden Satelliten angezeigt wird. Zusätzlich wird das aktuelle Modul links oben angezeigt, der nächste sichtbare Satellit (rechts oben) und die genauen Werte für Azimut und Elevation (links unten) sobald sich der Mauszeiger auf der *Polar View* befindet.

#### Single-Satellite View

In dieser Ansicht (siehe Abbildung 5.2) werden detaillierte Informationen zu einem ausgewählten Satelliten angezeigt, z.B. Azimut, Elevation, Entfernung der direkten Sichtverbindung (*Slant Range*), Höhe, Geschwindigkeit, Doppler-Verschiebung oder Signaldämpfung. Mit einem Klick auf das -Symbol links neben dem Satellitennamen kann zwischen den für dieses Modul ausgewählten Satelliten gewechselt werden.

## List View

Die Listenansicht zeigt eine tabellarische Auflistung aller für das aktuelle Modul ausgewählten Satelliten mit verschiedenen Details, mit je einem Satelliten pro Zeile. In Abbildung 5.3 ist die Listenansicht mit allen verfügbaren Details zu sehen. Mit einem Klick auf eine entsprechende Kategorie lässt sich das Sortierkriterium ändern. Falls hier ein variables Kriterium wie die Geschwindigkeit eingestellt wird, ändert sich die Sortierreihenfolge mit der eingestellten Auffrischrate (*Refresh Rate*). Die Bezeichnung des jeweiligen Details ist in dieser Ansicht abgekürzt, z.B. Az für *Azimut*. Unter den Moduleinstellungen beim Reiter *List View* kann ausgewählt werden, welches Detail angezeigt wird. Dort ist außerdem erkenntlich für was die entsprechenden Abkürzungen stehen.

PREDICT: Amateur																								
Satellite	Catnum	Az	El	Dir	Ra	Dec	Range	Rate	Next Event	Next AOS	Next LOS	Lat	Lon	SSP	Footp	Alt	Vel	Dop	Loss	Del	MA	Phase	Orbit	Vis
AO-16	20439	130.84°	1.44°	↑	77.85°	-24.94°	3123	-6.192	AOS: 2016/05/14 09:16:53	2016/05/14 09:16:53	2016/05/14 09:30:54	28.24°	31.38°	KL58QF	6061	795	7.455	2066	142.29	10.42	101.79°	143.14°	37540	D
AO-27	22825	302.43°	-33.05°	↓	246.64°	-5.76°	8274	4.923	AOS: 2016/05/14 10:30:59	2016/05/14 10:30:59	2016/05/14 10:38:17	32.75°	-94.79°	EM220R	6065	796	7.459	-1642	150.75	27.60	61.58°	86.60°	18040	D
AO-51	28375	164.60°	-61.90°	↑	175.50°	-73.32°	12103	-3.178	AOS: 2016/05/14 09:44:59	2016/05/14 09:44:59	2016/05/14 09:58:52	-74.56°	61.59°	MB05TK	6094	804	7.439	1060	154.06	40.37	177.00°	248.90°	62552	D
AO-7	7530	5.05°	5.12°	↑	193.90°	47.24°	4020	-0.030	LOS: 2016/05/14 09:23:02	2016/05/14 10:58:15	2016/05/14 09:23:02	77.98°	21.90°	KQ07WV	7909	1459	7.141	10	144.48	13.41	48.14°	67.70°	89887	D
FO-29	24278	75.48°	11.81°	↑	113.96°	18.46°	2967	-2.583	LOS: 2016/05/14 09:28:48	2016/05/14 10:55:02	2016/05/14 09:28:48	48.38°	43.41°	LN180J	7242	1187	7.207	862	141.85	9.90	81.95°	115.24°	97493	D
GO-32	25397	168.75°	-70.51°	↑	191.92°	-66.53°	12869	-1.410	AOS: 2016/05/14 09:55:13	2016/05/14 09:55:13	2016/05/14 10:03:27	-81.05°	141.42°	QA080W	6226	843	7.429	470	154.59	42.93	135.40°	190.41°	92704	D
HO-68	36122	37.82°	-62.98°	↑	183.47°	-24.63°	12691	-2.765	AOS: 2016/05/14 09:47:30	2016/05/14 10:04:23	2016/05/14 10:04:23	-4.54°	161.54°	RI05SK	7280	1201	7.254	922	154.47	42.33	198.82°	279.60°	30815	D
ISS	25544	351.52°	-78.11°	↑	203.47°	-35.89°	12897	-1.162	AOS: 2016/05/14 09:58:13	2016/05/14 09:58:13	2016/05/14 10:01:27	-25.14°	-166.90°	AG64NU	4465	412	7.663	388	154.61	43.02	155.26°	218.34°	99971	E
SO-50	27607	146.43°	-77.73°	↓	188.75°	-57.31°	13075	1.227	AOS: 2016/05/14 10:07:21	2016/05/14 10:07:21	2016/05/14 10:20:51	-64.63°	158.75°	QC95II	5459	632	7.561	-409	154.73	43.61	31.20°	43.87°	72027	E

Abbildung 5.3: List View

## Upcoming Passes View

Die *Upcoming Passes View* (siehe Abbildung 5.4) zeigt alle Satelliten des aktuellen Moduls, deren Azimut und Elevation, sowie die Zeit bis zum nächsten Verschwinden des Satelliten, dem sogenannten *Loss of Signal (LOS)* bzw. dem nächsten Auftauchen, auch *Acquisition of Signal (AOS)* genannt. Wie bei der *List View* ist es auch hier möglich nach den verschiedenen Spalten zu sortieren.

Satellite	Az	El	AOS/LOS
AO-27	29.48°	21.85°	05:04
AO-16	182.57°	-49.98°	25:16
FO-29	170.06°	-60.09°	26:43
AO-7	157.48°	-65.08°	32:06
GO-32	339.04°	-64.81°	55:41
HO-68	237.62°	-52.19°	01:08:45
SO-50	152.56°	-31.48°	01:15:43
AO-51	47.69°	-72.15°	09:34:23
ISS	114.72°	-65.28°	12:46:49

Abbildung 5.4: Upcoming Passes View

### 5.2.2 Weitere Ansichten

Bei allen Ansichten kann durch einen Klick auf den Satellitennamen ein kleines Pop-Up Menü geöffnet werden, welches den entsprechenden Satellitennamen, die Option *Show next pass* und die Option *Future passes* anzeigt. Bei einem Klick auf den Satellitennamen öffnet sich ein Fenster mit dem Titel *Satellite Info*, wie in Abbildung 5.5 zu sehen. Dort sind unter dem Reiter *Orbit Info* verschiedene Informationen zum Satellitenorbit und unter dem Reiter *Transponders* die verfügbaren Transponder zu sehen.

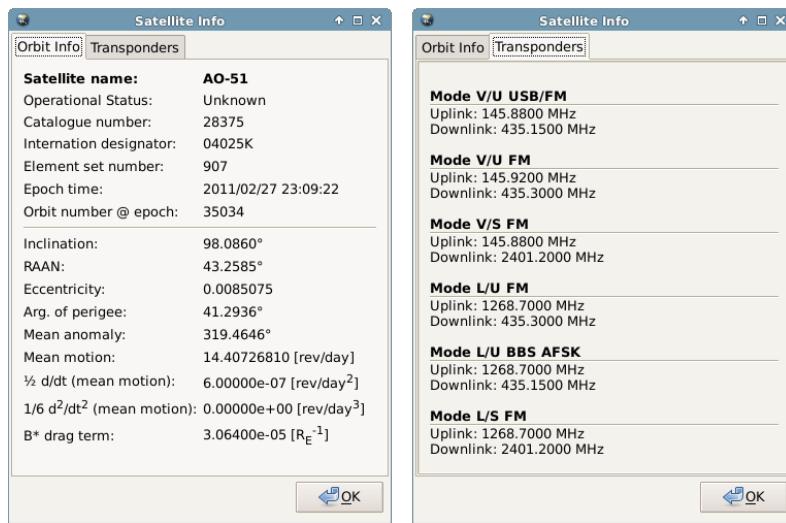


Abbildung 5.5: *Satellite Info*

Mit einem Klick auf die Option *Show next pass* gelangt man zu einer Übersicht über den nächsten Durchlauf des entsprechenden Satelliten. Die Details sind tabellarisch, als Polaransicht und als Verlauf des Azimut- und Elevationswinkels über der Zeit zu sehen (siehe Abbildung 5.6).

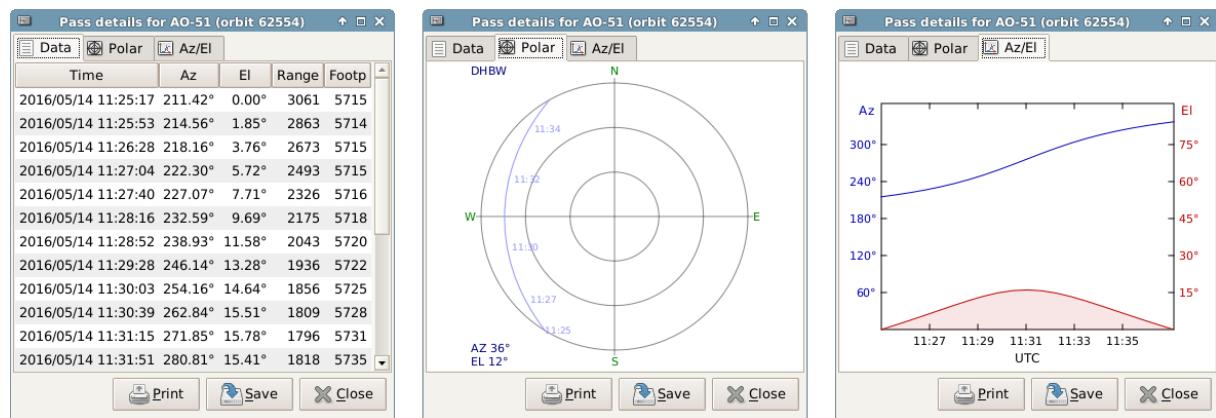


Abbildung 5.6: *Pass Details*

Die Option *Future passes* öffnet ein Fenster, in welchem die nächsten Durchläufe des entsprechenden Satelliten tabellarisch dargestellt sind (siehe Abbildung 5.7). Hierbei ist die Anzahl der darzustellenden Durchläufe in den GPredict-Einstellungen unter *Predict* als *Number of passes to predict* einstellbar.

Upcoming passes for AO-51					
AOS	LOS	Duration	Max El	AOS Az	LOS Az
2016/05/14 11:25:17	2016/05/14 11:37:14	00:11:56	15.78°	211.40°	330.90°
2016/05/14 21:48:41	2016/05/14 22:01:36	00:12:55	17.95°	28.00°	151.46°
2016/05/14 23:27:06	2016/05/14 23:41:46	00:14:39	66.49°	10.74°	203.90°
2016/05/15 01:06:48	2016/05/15 01:17:51	00:11:03	11.27°	355.88°	256.29°
2016/05/15 09:05:45	2016/05/15 09:18:52	00:13:06	29.84°	135.28°	355.11°
2016/05/15 10:44:19	2016/05/15 10:57:50	00:13:30	35.33°	188.03°	339.49°
2016/05/15 21:09:47	2016/05/15 21:19:39	00:09:51	7.03°	39.63°	125.08°
2016/05/15 22:47:15	2016/05/15 23:01:58	00:14:43	56.91°	16.72°	183.81°
2016/05/16 00:26:29	2016/05/16 00:39:40	00:13:11	22.68°	2.30°	234.24°
2016/05/16 08:27:10	2016/05/16 08:38:40	00:11:30	14.77°	113.22°	1.23°

Abbildung 5.7: *Upcoming Passes*

### 5.2.3 Modul Pop-Up Menü

Um das Modul Pop-Up Menü zu öffnen, klickt man ganz rechts oben im GPredict-Fenster auf das -Symbol. Im daraufhin erscheinenden Pop-Up Menü ist es möglich die Positionierung eines Moduls innerhalb des GPredict-Fensters einzustellen, ein Modul zu kopieren, zu löschen, zu schließen oder genauer zu konfigurieren. Außerdem sind dort weitere Funktionen, welche im Folgenden genauer beschrieben werden, zugänglich.

Wie in Abbildung 5.8 zu sehen, bietet die Funktion *Sky at a glance* eine Übersicht darüber, wann welche Satelliten innerhalb der nächsten acht Stunden sichtbar sind. Dieser Zeitraum lässt sich in den GPredict-Einstellungen bei *Predict* unter dem Reiter *Sky at a Glance* zwischen einer und 24 Stunden einstellen.

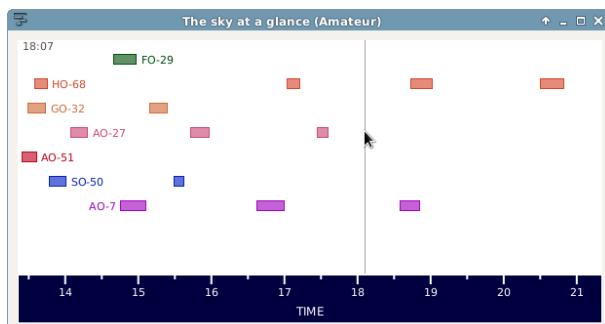


Abbildung 5.8: *The sky at a glance*

Über die Funktion *Time Controller* (siehe Abbildung 5.9) lässt sich die Zeit, auf die sich die Berechnungen von GPredict beziehen, ändern. Hierbei ist standardmäßig das aktuelle Datum und die aktuelle Uhrzeit eingestellt. Außerdem kann hier die Geschwindigkeit, mit der die eingestellte Zeit fortschreitet, auf maximal ein Hundertfaches erhöht werden. Die eingestellte Zeit wird im GPredict-Fenster ganz links oben im ausgewählten Format angezeigt. Mit dem Schieberegler kann die Zeit zwischen  $-2,5$  und  $+2,5$  Stunden eingestellt werden.



Abbildung 5.9: *Time Controller*

Klickt man auf *Configure*, öffnet sich ein Fenster wie in Abbildung 5.10 zu sehen. Hier lassen sich die zu verfolgenden Satelliten und die Bodenstation für das aktuelle Modul auswählen. Außerdem gelangt man mit einem Klick auf das Feld *Properties* in die Modul-Einstellungen. Diese gelten im Gegensatz zu den in den allgemeinen GPredict-Einstellungen zu findenden Modul-Einstellungen nur für das aktuelle Modul. Auf Seite 54 sind nähere Infos zu den Modul-Einstellungen zu finden.

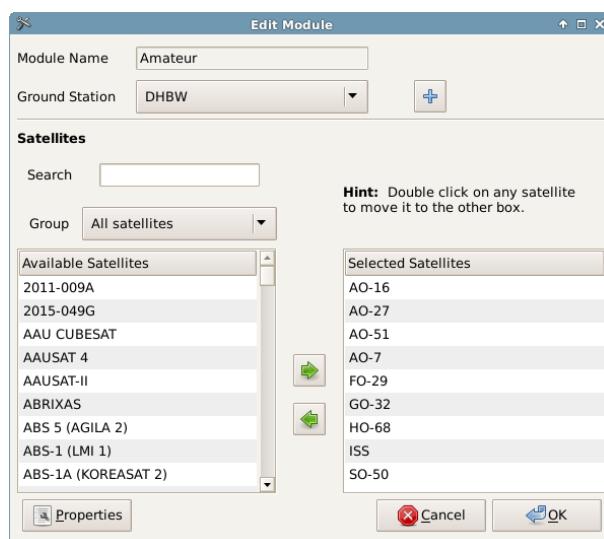


Abbildung 5.10: *Edit Module*

Hinter der Funktion *Antenna Control* (siehe Abbildung 5.11) verbirgt sich ein Bedienfeld zur Steuerung der Antennenrotoren. Bevor dieses geöffnet werden kann, muss zunächst in den GPredict-Einstellungen unter *Interfaces* mindestens eine Schnittstelle zur Rotorensteuerung konfiguriert werden (siehe Abschnitt 5.4.6).

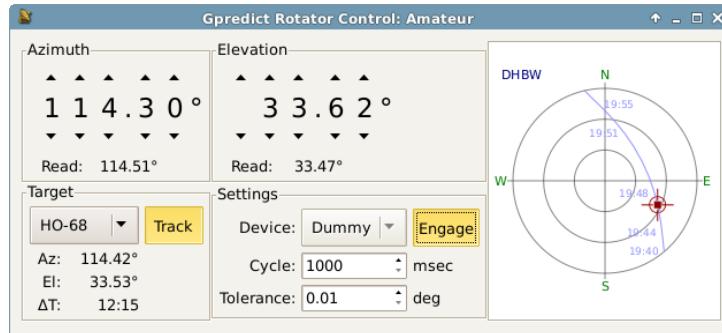


Abbildung 5.11: Rotorensteuerungs-Bedienfeld

Das Bedienfeld beinhaltet zum Einen eine Polaransicht, auf der die Spur und der aktuelle Ort des zu verfolgenden Satelliten (dargestellt durch ein Viereck) und die gegenwärtige Ausrichtung der Antenne (dargestellt durch ein Fadenkreuz) zu sehen ist. Zum Anderen sind folgende vier Bereiche verfügbar:

- **Azimuth:** In diesem Feld lässt sich die Ausrichtung der Antenne in Azimut-Richtung steuern, vorausgesetzt, dass die *Track*-Funktion nicht aktiviert ist. Am unteren Ende des Feldes wird unter *Read* der aktuelle Azimut-Winkel der Antenne angezeigt. Ist keine Verbindung zum Rotor aufgebaut, wird hier „---“ angezeigt. Liegt ein Verbindungsproblem vor, erscheint „ERROR“.
- **Elevation:** In diesem Feld lässt sich die Ausrichtung der Antenne in Elevations-Richtung steuern, vorausgesetzt, dass die *Track*-Funktion nicht aktiviert ist. Am unteren Ende des Feldes wird unter *Read* der aktuelle Elevations-Winkel der Antenne angezeigt. Ist keine Verbindung zum Rotor aufgebaut, wird hier „---“ angezeigt. Liegt ein Verbindungsproblem vor, erscheint „ERROR“.
- **Target:** Hier lässt sich der zu verfolgende Satellit auswählen. Es stehen hierbei nur die für das aktuelle Modul ausgewählten Satelliten zu Verfügung. Aktiviert man die Schaltfläche *Track*, wird der ausgewählte Satellit verfolgt. Unter dem Satellitennamen werden die jeweiligen Winkel in Echtzeit dargestellt und hinter  $\Delta T$  wird die Zeit bis zum nächsten AOS bzw. LOS angezeigt.

- **Settings:** Hier lässt sich die in den GPredict-Einstellungen festgelegte Schnittstelle zur Kommunikation mit den Rotoren auswählen. Mit einem Klick auf die Schaltfläche *Engage* wird die Verbindung zu dieser Schnittstelle aufgebaut bzw. unterbrochen. Unter *Cycle* kann dabei der Zyklus eingestellt werden, in welchem Kommandos an die Rotor-Schnittstelle gesendet und Winkelwerte von dieser abgefragt werden. Ein sinnvoller Wert liegt hierbei zwischen zwei und fünf Sekunden. Bei *Tolerance* wird die tolerierte Differenz zwischen abgefragtem und eingestelltem Winkel eingetragen. Sobald diese überschritten wird, wird ein Kommando an die Rotor-Schnittstelle geschickt. Hierbei sollte sowohl die Winkelauflösung der Rotorsteuerung, als auch die Keulenbreite der Antenne berücksichtigt werden. Nach fünf aufeinanderfolgenden Fehlern bei der Kommunikation mit den Rotoren, wird die Verbindung automatisch unterbrochen.

In Tabelle 5.1 sind alle möglichen Kombinationen der Schaltflächen *Track* und *Engage* und deren Auswirkung beschrieben.

Track	Engage	Beschreibung
Inaktiv	Inaktiv	Es werden weder Kommandos an die Rotoren gesendet, noch wird die aktuelle Ausrichtung der Antenne ausgelesen. Die aktuellen Winkel des zu verfolgenden Satelliten werden nicht in die Winkelsteuerungs-Eingabefelder übertragen.
Aktiv	Inaktiv	Die aktuellen Winkel des zu verfolgenden Satelliten werden in die Winkelsteuerungs-Eingabefelder übertragen. Allerdings werden keine Kommandos an die Rotoren geschickt und die aktuelle Ausrichtung der Antenne wird nicht ausgelesen.
Aktiv	Aktiv	Die aktuellen Winkel des zu verfolgenden Satelliten werden in die Winkelsteuerungs-Eingabefelder übertragen und diese werden an die Rotoren geschickt. Die aktuelle Ausrichtung der Antenne wird ausgelesen.
Inaktiv	Aktiv	Die Winkel, die in den Winkelsteuerungs-Eingabefelder eingestellt sind, werden an die Rotoren gesendet und die aktuelle Ausrichtung der Antenne wird ausgelesen.

Tabelle 5.1: Betriebsmodi des *Antenna Control*-Bedienfelds, Quelle: [18]

Um dem Funkgerät die entsprechenden Up- und Downlink-Frequenzen inklusive Korrektur der Doppler-Verschiebung zu übermitteln, wird das *Radio Control*-Bedienfeld (siehe Abbildung 5.12) verwendet. Dieses kann nur geöffnet werden, wenn mindestens eine Funkgerät-Schnittstelle in den GPredict-Einstellungen unter *Interfaces* konfiguriert ist (siehe Abschnitt 5.4.5).



Abbildung 5.12: Funkgerät-Steuerung

Neben einer Anzeige die die Zeit bis zum nächsten AOS bzw. LOS darstellt (ganz unten im Bedienfeld in fettgedruckten Buchstaben) kann das *Radio Control*-Bedienfeld in folgende vier Bereiche untergliedert werden:

- **Downlink:** In diesem Bereich kann die Frequenzeinstellung für die Downlink-Frequenz vorgenommen werden. Außerdem ist die aktuelle Dopplerfrequenz und ein Feld namens *LO*, welches für eine Offsetfrequenz steht, die im Funkgerät eingestellt werden kann, steht. Die korrigierte Frequenz welche nach *Radio* angezeigt wird, setzt sich damit beispielsweise wie folgt zusammen:  $f_{Radio} = f_{Satellite} + f_{Doppler} - f_{LO}$
- **Uplink:** Dieser Bereich beinhaltet die Frequenzeinstellungen für die Uplink-Frequenz und besitzt die gleichen Eigenschaften wie der Downlink-Bereich.
- **Target:** Hier kann der zu verfolgende Satellit und der zu verwendete Transponder eingestellt werden. Mit einem Klick auf *Track* wird die Doppler-Verschiebung des ausgewählten Satelliten bei den Frequenzeinstellungen im Downlink- und Uplink-Bereich korrigiert. Eine wichtige Rolle spielt hierbei die Schaltfläche *T* (*Tune*), da die Frequenzen des Transponders nur durch einen Klick auf diese in die Frequenzeinstellungen übertragen werden. Sollte der Transponder beispielsweise nur einen Down-

link besitzen, wird auch nur die Frequenz im Downlink-Bereich eingestellt. Mit der Schaltfläche *L (Lock)* lässt sich die Differenz der Downlink- und Uplink-Frequenz sperren, das heißt Änderungen an der einen wirken sich auch auf die andere Frequenz aus. Dies ist nicht für Transponder möglich, die nur einen Up- oder Downlink besitzen.

- **Settings:** In diesem Bereich lassen sich bis zu zwei Funkgeräte auswählen die in den GPredict-Einstellungen unter *Interfaces* eingerichtet wurden. Das obere Gerät stellt das Primäre dar und kann für Up- und Downlink verwendet werden. Das Untere kann als sekundäres Gerät verwendet werden, welches nur für den Uplink verwendet werden kann. Mit einem Klick auf *Engage* wird die Kommunikation zwischen GPredict und Funkgerät(en) aufgebaut. Hierbei kann im Feld neben *Cycle* der Zyklus eingetragen werden, nach welchem Kommandos an das Funkgerät geschickt werden sollen.

In Tabelle 5.2 ist eine Übersicht über die verschiedenen Betriebsmodi des *Radio Control*-Bedienfelds zu sehen.

Track	Engage	Beschreibung
Inaktiv	Inaktiv	Es wird keine Korrektur der Doppler-Verschiebung durchgeführt, keine Befehle an das Funkgerät gesendet und die aktuelle Frequenz des Funkgeräts nicht ausgelesen.
Aktiv	Inaktiv	Die Korrektur der Doppler-Verschiebung wird durchgeführt, es werden aber weder Befehle an das Funkgerät geschickt noch wird die aktuelle Frequenz ausgelesen.
Aktiv	Aktiv	Die Doppler-Verschiebung wird korrigiert und die eingestellte Frequenz wird zum Funkgerät geschickt. Die aktuelle Frequenz des Funkgeräts wird ausgelesen.
Inaktiv	Aktiv	Die Korrektur der Doppler-Verschiebung wird nicht ausgeführt. Die eingestellte Frequenz wird an das Funkgerät geschickt und die dort eingestellte Frequenz wird ausgelesen.

Tabelle 5.2: Betriebsmodi des *Radio Control*-Bedienfelds, Quelle: [18]

Wenn die *Engage*-Schaltfläche aktiviert ist, wird die Korrektur der Doppler-Verschiebung durchgeführt, egal ob der ausgewählte Satellit sichtbar ist oder nicht.

### 5.2.4 GPredict-Einstellungen

Um in die GPredict-Einstellungen zu gelangen, klickt man links oben unterhalb der Titelleiste auf *Edit* und anschließend auf *Preferences*. Nun öffnet sich ein Fenster bei dem standardmäßig die Erste der vier Kategorien zu sehen ist, die Kategorie *General* (siehe Abbildung 5.13). Nach einer Veränderung in den GPredict-Einstellungen muss GPredict zunächst neugestartet werden, damit diese Änderung wirksam wird. Überall wo die Schaltfläche *Reset* zu finden ist, können mit einem Klick auf diese, die Standard-Einstellungen wiederhergestellt werden. Bei den meisten Optionen erscheint ein kleines Informations-Fenster, sobald man mit dem Mauszeiger einige Sekunden über der entsprechenden Option verharrt.

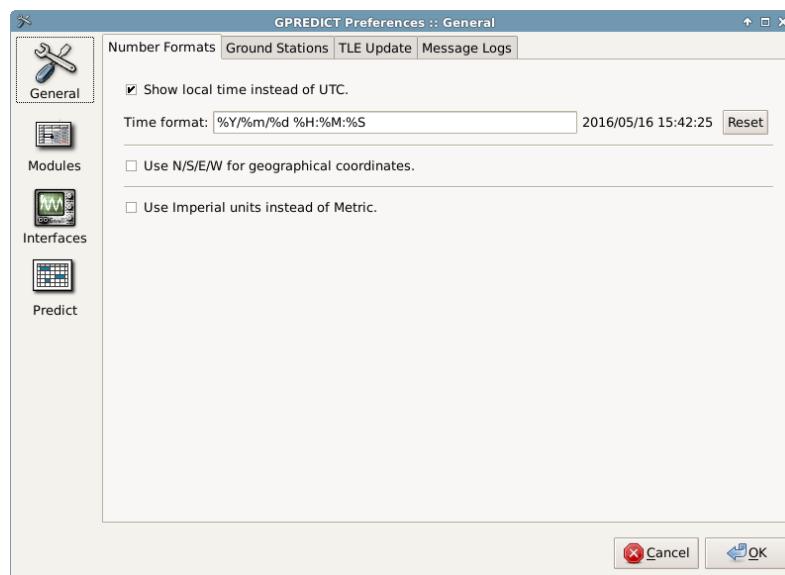


Abbildung 5.13: GPredict-Einstellungen, Kategorie General

Die Kategorie *General* ist in folgende vier Reiter untergliedert:

- **Number Formats:** Hier lassen sich Einstellungen zum Zeitformat, zu den geografischen Koordinaten und zu den Längen- und Geschwindigkeits-Einheiten vornehmen.
- **Ground Stations:** Unter diesem Reiter können beliebig viele Bodenstationen eingerichtet werden. Mindestens eine muss jedoch zu jeder Zeit vorhanden sein.

- **TLE Update:** Hier können Einstellungen zur Aktualisierung der Kepler-Elemente vorgenommen werden.
- **Message Logs:** Hier können Einstellungen bzgl. des GPredict-Protokolls vorgenommen werden. Unter *Log browser* im Menü *File* kann dieses dargestellt werden.

Zur Kategorie *Modules* gelangt man über zwei Wege. Der erste führt über die allgemeinen GPredict-Einstellungen, wie in diesem Abschnitt beschrieben. Der Zweite führt über das Modul Pop-Up Menü, wie auf Seite 48 beschrieben. Der Unterschied ist, dass bei den Modul-Einstellungen in den GPredict-Einstellungen die Standardwerte für alle Module eingestellt werden können und beim zweiten Weg nur die für das entsprechende Modul. Die Modul-Einstellungen sind in folgende sechs Reiter untergliedert:

- **Layout:** Bei diesem Reiter lässt sich konfigurieren, welche Ansichten in welcher Anordnung im GPredict-Fenster zu sehen sind. Hierbei kann aus verschiedenen Anordnungen ausgewählt werden oder eine völlig benutzerspezifische Anordnung erstellt werden (Hierfür wird an dieser Stelle auf [18] Seite 67f verwiesen).
- **Refresh Rates:** Hier lässt sich die Zeitspanne auswählen, nach welcher GPredict die Positions-Berechnungen periodisch für jeden Satelliten durchführt, der für das geöffnete Modul ausgewählt wurde. Außerdem lässt sich einstellen mit welchem ganz-zahligen Vielfachen dieser Zeitspanne die einzelnen Ansichten aufgefrischt werden.
- **List View:** Hier lassen sich die Parameter der *List View* auswählen.
- **Map View:** Hier lassen sich Darstellungsoptionen für die *Map View* konfigurieren.
- **Polar View:** Hier lassen sich Darstellungsoptionen für die *Polar View* konfigurieren.
- **Single Sat View:** Hier lassen sich die Parameter der *Single Sat View* auswählen.

Die Kategorie *Interfaces* ist in zwei Reiter untergliedert, *Radios* und *Rotators*. Dort wird jeweils eine Liste der bereits eingerichteten Funkgeräte bzw. Rotoren angezeigt, welche standardmäßig leer ist. Die Anzahl der Geräte ist dabei nach oben nicht beschränkt. Die detaillierte Einrichtung ist in Abschnitt 5.4.5 und 5.4.6 genauer beschrieben.

Beim Öffnen der Kategorie *Predict* wird der Reiter *Pass Conditions* angezeigt. Unter diesem lassen sich folgende Parameter einstellen:

- **Minimum elevation:** Dieser Parameter gibt an, ab welcher Elevation von einem Durchlauf eines Satelliten ausgegangen wird. Das heißt, übersteigt die maximale Elevation diesen Wert, wird dieser Durchlauf in die *Upcoming Passes*, die *Pass Details* und die Ansicht *The sky at a glance* aufgenommen.
- **Number of passes to predict:** Anzahl der angezeigten zukünftigen Durchläufe.
- **Passes should occur within:** Dieser Parameter definiert den Zeitraum, in welchem zukünftige Durchläufe von GPredict berücksichtigt werden. So tritt entweder erst die Anzahl der angezeigten zukünftigen Durchläufe ein oder der Zeitraum in welchem zukünftige Durchläufe berücksichtigt werden sollen.
- **Time resolution:** Hier kann die Zeitauflösung eingetragen werden, mit der der nächste Durchlauf in den *Pass Details* dargestellt wird. Je geringer die Auflösung, desto mehr Einträge werden angezeigt.
- **Number of entries:** Mit diesem Parameter kann die Anzahl der Einträge des nächsten Durchlaufs in den *Pass Details* eingestellt werden. Hierbei ist zu beachten, dass diese Einstellung eine höhere Priorität als die Zeitauflösung besitzt.
- **Twilight threshold:** Dieser Parameter gibt an, ab welcher Elevation ein Satellit als sichtbar gilt. Für die Satellitenverfolgung spielt dieser Parameter keine Rolle.

Unter den Reitern *Multiple Passes* und *Single Pass* lassen sich die Parameter auswählen, die in den Ansichten *Upcoming Passes* und *Pass Details* dargestellt werden. Beim Reiter *Sky at a Glance* können Darstellungsoptionen für die Funktion *The sky at a glance* eingestellt werden.

Eine detailliertere Beschreibung der GPredict-Einstellungen kann aus [18] Seite 23ff entnommen werden.

## 5.3 HamLib-Programmierschnittstelle

### 5.3.1 Übersicht

Da es keinen einheitlichen Kommunikationsstandard für die zahlreichen Funkgeräte und Rotoren unterschiedlicher Hersteller gibt, ist für die Verwendung von GPredict eine applicationsspezifische Programmierschnittstelle oder auch Application Programming Inter-

face (API) erforderlich. Mit den *Ham Radio Control Libraries* (englisch für Amateurfunk-Kontrollbibliotheken), kurz HamLib, steht dem Benutzer eine solche API zur Verfügung. HamLib ist unter der GPL lizenziert und ist unter [20] in der Version 3.0.1 als Download für Linux und Windows kostenlos verfügbar. Wie in Abbildung 5.14 zu sehen, ermöglicht HamLib einer Software wie GPredict die Kommunikation mit verschiedenen Funkgeräten und Rotoren, in dem es für jedes dieser Geräte einen eigenen Treiber bzw. ein eigenes Backend (BE) zur Verfügung stellt.

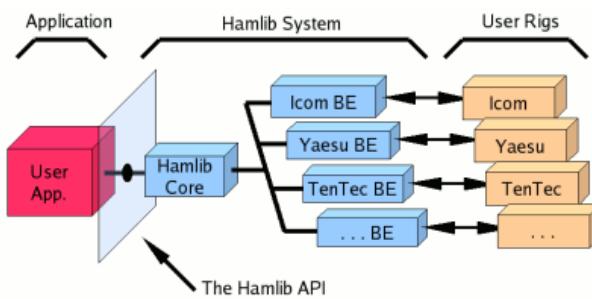


Abbildung 5.14: HamLib Design, Quelle: [21]

Dabei verwendet man entweder den Quellcode um eine benutzerspezifische Anwendung zu erstellen oder man greift auf die bereits fertig kompilierten Programme zurück, welche im Folgenden aufgelistet sind:

- **rigctl:** Ein Kommandozeilenprogramm um Befehle über die Kommandozeile an das Funkgerät zu senden (Unter Windows: *rigctl.exe*).
- **rotctl:** Ein Kommandozeilenprogramm um Befehle über die Kommandozeile an die Antennenrotoren zu senden (Unter Windows: *rotctl.exe*).
- **rigctld:** Ein Kommandozeilenprogramm um Befehle über das TCP/IP-Protokoll an das Funkgerät zu senden (Unter Windows: *rigctld.exe*).
- **rotctld:** Ein Kommandozeilenprogramm um Befehle über das TCP/IP-Protokoll an die Antennenrotoren zu senden (Unter Windows: *rigctld.exe*).

Hierbei steht „*rot*“ für „Rotator“ (deutsch: Rotor), „*rig*“ für „Rig“ (deutsch: Amateurfunkgerät) und das „*d*“ am Ende von *rigctld* und *rotctld* für „Deamon“ (deutsch: Hintergrundprozess).

### 5.3.2 Parameter-Konfiguration

Für die erfolgreiche Nutzung der oben genannten Programme müssen einige Optionen und Parameter berücksichtigt werden. Hierfür ist es notwendig die erforderlichen Informationen zu den Rotoren und zum Funkgerät zu kennen. Um eine Übersicht der möglichen Optionen und Befehle zu erhalten, kann das jeweilige Programm einfach mit der Option **-h** oder **--help** ausgeführt werden.

Die erste zentrale Information ist die *Hamlib ID*. Hierfür gibt man folgenden Befehl in die Kommandozeile ein und erhält eine Liste mit allen unterstützten Funkgeräten (für *rigctl.exe* und *rotctl(d).exe* analog durchführbar):

```
rigctl.exe -l
```

Im Folgenden sind die beiden Zeilen dargestellt, die für das in diesem Projekt verwendete Funkgerät, sowie für die verwendete Rotorschnittstelle relevant sind. Hierbei steht der erste Eintrag für die *Hamlib ID*, der Zweite für den Hersteller, der Dritte für die Version und der Vierte für den Test-Status:

368	Icom	IC-9100	0.7	Untested
601	Yaesu	GS-232A	0.3	Beta

Als Nächstes muss die Schnittstelle zum Funkgerät bzw. zur Rotorensteuerung angegeben werden. Bei der im Rahmen dieses Projekts verwendeten Software (siehe Abschnitt 5.4.1) werden hierfür serielle Schnittstellen verwendet. Wie diese konfiguriert werden, ist im folgenden Auszug aus der Hilfe, die bei Eingabe der Option **-h** oder **--help** erscheint (für *rigctl.exe*), zu sehen:

```
-r, -rig-file=DEVICE      select radio model number. See model list
-s, -serial-speed=BAUD   set serial speed of the serial port
-C, -set-conf=PARM=VAL    set config parameters
```

Um eine Liste der möglichen Konfigurationsparameter (**config parameters**) auszugeben, führt man *rigctl(d).exe* bzw. *rotctl(d).exe* mit der Option **-L** oder **--show-conf** aus. Die genauen Werte sind in Abschnitt 5.4.1 aufgeführt.

Im Testbetrieb ist die Ausgabe von Statusmeldungen (z.B. Warnungen oder Fehler) oft sehr hilfreich. Hierfür lässt sich mit der Option **-v** die Intensität der Ausgabe einstellen.

Bei `-v` werden nur grobe Fehler ausgegeben, wohingegen bei `-vvvvv` alle verfügbaren Statusmeldungen der API ausgegeben werden. Bei Weglassen der Option erfolgt keine Ausgabe. Mit diesem Wissen sieht die Ausführung des Programms `rotctl.exe` im Rahmen dieses Projekts wie folgt aus:

```
rotctl.exe -vvvv -m 601 -r COM10 -s 9600 -C stop_bits=1,data_bits=8
```

Da beim Funkgerät IC-9100 das Icom-spezifische Protokoll CI-V (das „V“ steht hierbei für die römische Zahl) für die Fernsteuerung verwendet wird, kommt bei der Verwendung von `rigctl.exe` noch die CI-V-Adresse als Parameter hinzu. Somit sieht die Ausführung des Programms `rigctl.exe` wie folgt aus:

```
rigctl.exe -vvvv -m 368 -r COM5 -c 0x7C -s 19200 -C stop_bits=1,data_bits=8
```

### 5.3.3 Verwendung

Nachdem das Programm `rigctl.exe` bzw. `rotctl.exe` mit den unter Abschnitt 5.3.2 dargestellten Parametern gestartet wurde, hat man die Möglichkeit Befehle an das Funkgerät bzw. an die Rotorensteuerung zu senden. Eine Übersicht der Befehle lässt sich der Ausgabe der Programmausführung mit dem Parameter `-h` bzw. `--help` entnehmen.

Da nicht jedes Funkgerät bzw. nicht jede Rotorensteuerung alle möglichen Befehle unterstützt, ist es wissenswert welche Befehle von den in diesem Projekt verwendeten Geräten unterstützt werden. Unter Angabe der *Hamlib ID* und der Option `-u` bzw. `--dump-caps` kann eine Übersicht der *capabilities* (Fähigkeiten), z.B. der unterstützten Befehle oder der Standard-Parameter ausgegeben werden:

```
rigctl.exe -m 368 -u
rotctl.exe -m 601 -u
```

Um die Verwendung dieser Kommandozeilenprogramme zu vereinfachen und um gleichzeitig die notwendige Konfiguration festzuhalten, wurde für die Programme `rigctl`, `rigctld`, `rotctl` und `rotctld` jeweils ein Batch-Skript erstellt (siehe Anhang Anhang B; Anhang C; Anhang D; und Anhang E:). Bei der Verwendung unter Linux können diese ohne großen Aufwand in Bash-Skripte umgeschrieben werden.

## 5.4 Inbetriebnahme unter Windows

### 5.4.1 Zusätzliche Software

Sowohl das Funkgerät als auch die Rotorensteuerung werden über das Hochschul-Netzwerk angesteuert. Für die Kommunikation mit den notwendigen Hardwarekomponenten (Banana Pi und *NetCOM 211*) stehen bereits Programme für Windows zur Verfügung. In Abbildung 5.15 ist eine Übersicht zu sehen, wie GPredict mit dem Funkgerät und den Rotoren kommuniziert. Hierbei sind reine Software-Komponenten in blau und Hardware-Komponenten in rot dargestellt. Der Banana Pi liegt zwar als Hardware-Komponente vor, jedoch ist die Software selbst erstellt worden und kann jederzeit angepasst werden, deshalb ist er in blau-rot dargestellt.

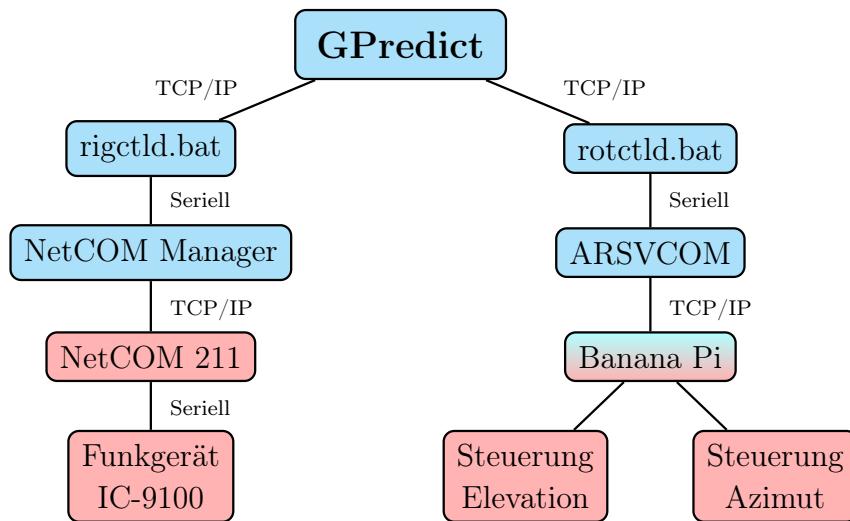


Abbildung 5.15: Kommunikation zwischen Hard- und Software

Hierbei kommuniziert GPredict über eine IP-basierte Schnittstelle nur mit den HamLib-Programmen *rigctld.exe* und *rotctld.exe*. Diese werden unter Berücksichtigung der notwendigen Parameter als Batch-Skripte ausgeführt und müssen solange eine Kommunikation zwischen GPredict und Funkgeräten bzw. Rotoren bestehen soll, aktiv sein.

Die Software *NetCOM Manager* stellt eine virtuelle serielle Schnittstelle (COM5) zur Verfügung, auf die von *rigctld.exe* zugegriffen werden kann. Das *NetCOM 211*-Modul ist wiederum per serieller Schnittstelle mit dem Funkgerät verbunden und ermöglicht die Kommunikation über das Netzwerk mit der Software *NetCOM Manager*. Somit muss man sich lediglich im Hochschul-Netzwerk befinden, um mit dem Funkgerät kommunizieren zu

können. Wird hierbei die Möglichkeit einer VPN-Verbindung wahrgenommen, kann quasi von jedem beliebigen Ort mit Internetanbindung mit dem Funkgerät kommuniziert werden.

Für die Kommunikation mit den Rotoren, wird auf die Software *ARSVCOM* zurückgegriffen. Diese emuliert eine Rotorensteuerung vom Typ Yaesu *RS232A*, wofür ebenfalls ein virtueller, beliebig einstellbarer COM-Port erstellt wird. Der COM-Port und der Rotorensteuerungs-Typ werden als Parameter bei *rotctl(d).exe* angegeben. Die Software *ARSVCOM* kommuniziert über das Netzwerk mit dem Banana Pi, an welchem wiederum die Steuerungseinheiten des Azimut- und Elevations-Rotors angeschlossen sind (siehe Abschnitt 3.2.2).

#### 5.4.2 Test der Bedienfelder

Um sich mit den Bedienfeldern zur Funkgerät- bzw. Rotorensteuerung vertraut zu machen, kann auf zwei sogenannte *Dummy*-Schnittstellen zugegriffen werden. Diese sind sowohl bei *rigctl.exe*, als auch bei *rotctl.exe* unter der *Hamlib ID* 1 erreichbar. In Abbildung 5.16 ist eine beispielhafte Konfiguration zu sehen, die in den *Interface*-Einstellungen zur Kommunikation mit den *Dummy*-Schnittstellen vorgenommen werden kann.

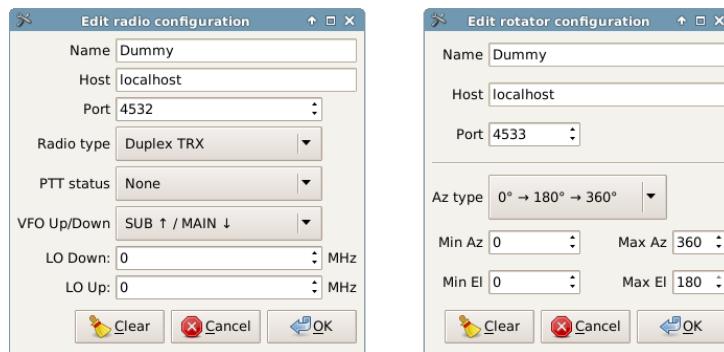


Abbildung 5.16: Schnittstellenkonfiguration für das Dummy-Interface

Für diesen Test wurden für die Programme *rigctld.exe* und *rotctld.exe* ebenfalls eigene Batch-Skripte erstellt (siehe Anhang Anhang F: und Anhang G:). Die hierfür verwendeten Optionen unterscheiden sich lediglich im Port, für *rigctld.exe* ist standardmäßig der Port 4532 und für *rotctld.exe* der Port 4533 vorgesehen. Dieser kann jedoch nach Belieben angepasst werden.

Sobald ein Fehler bei der Kommunikation zwischen den Bedienfeldern und den HamLib-Programmen fünf Mal in Folge aufgetreten ist, wird aus Sicherheitsgründen die Kommunikation automatisch abgebrochen.

### 5.4.3 Test der Kommunikation zum Funkgerät

Um die Kommunikation zwischen HamLib und dem Funkgerät zu testen, werden zunächst einzelne Kommandos mit Hilfe des Skripts *rigctl.bat* an das Funkgerät geschickt. Somit kann sichergestellt werden, ob das HamLib-Backend für das Funkgerät IC-9100 richtig implementiert wurde. Hierbei kann nicht davon ausgegangen werden, dass ein reibungsloser Betrieb möglich ist, da als Test-Status des HamLib-Backends für das Funkgerät IC-9100 „Untested“ angegeben ist (siehe Seite 57).

In der Betriebsanleitung des Funkgeräts IC-9100 (siehe [22]) ist auf Seite 183ff eine Übersicht der Befehlsstruktur und der zulässigen Kommandos des ICOM-spezifischen Protokolls CI-V dargestellt. Zur Kontrolle, ob die HamLib-Befehle auch den richtigen CI-V-Kommandos entsprechen, wird mit dem Windows-Programm *Serial Port Monitor* (hierfür kann auch ein anderes Programm verwendet werden) der serielle Port COM5 abgehört, während über *rigctl.bat* verschiedene Befehle an das Funkgerät geschickt werden.

Für die folgenden grundlegenden Befehle konnte eine Übereinstimmung der von HamLib verwendeten mit den in [22] zu findenden CI-V Befehlen festgestellt und damit eine korrekte Funktionalität sichergestellt werden:

- F: `set_freq` (Setzen der Frequenz des *Main*-Bandes)
- f: `get_freq` (Auslesen der Frequenz des *Main*-Bandes)
- M: `set_mode` (Setzen des Modus und der entsprechenden Bandbreite)
- m: `get_mode` (Auslesen des Modus und der zugehörigen Bandbreite)

Eine Übersicht der gültigen Eingabeparameter von *rigctl(d).exe* kann über die Option `--dump-caps` ausgegeben werden (siehe Abschnitt 5.3.3).

Da das Funkgerät in den meisten Fällen bei der Kommunikation mit Satelliten im Satellitenmodus betrieben werden muss, ist es essentiell, dass dieser über *rigctl(d).exe* aktiviert werden kann. Beim Test des entsprechenden Befehls (siehe Tabelle 5.3), wurde jedoch festgestellt, dass das CI-V-Kommando, dass an das Funkgerät geschickt wird, nicht dem Korrekten aus [22] entspricht. Für das fertig kompilierte Programm *rigctl(d).exe* gibt es zwar den Befehl `send_cmd` mit dem laut [21] ASCII-Zeichen geschickt werden können, dieser konnte im Rahmen dieses Projektes jedoch nicht erfolgreich verwendet werden. Es konnte hierbei auch nach Testen verschiedener Teile des CI-V-Kommandos der Satellitenmodus am Funkgerät nicht aktiviert werden. Somit ist es nicht möglich einen „rohen“ Befehl zu versenden, ohne einen Eingriff in den Quellcode von HamLib vorzunehmen. In Tabelle 5.3 ist der HamLib-Befehl zur Aktivierung des Satellitenmodus, das verwendete CI-V-Kommando und das tatsächliche CI-V-Kommando dargestellt. Die Ausschnitte, die nicht übereinstimmen, sind rot markiert.

HamLib-Befehl <i>rigctl(d).exe</i>	CI-V (Verwendet)	CI-V (Tatsächlich)
U SATMODE 1	FE FE 7C E0 <b>1A 07</b> 01 FD	FE FE 7C E0 16 5A 01 FD

Tabelle 5.3: Fehlerhaftes CI-V Kommando beim Aktivieren des Satellitenmodus

#### 5.4.4 Test der Kommunikation zu den Rotoren

Beim Test der Bedienelemente kann der Ausgabe des Skripts *rotctld-dummy.bat* bei Verwendung der höchsten Ausgabeintensität `-vvvvv` entnommen werden, dass lediglich die Befehle `set_pos` und `get_pos` beim Verfolgen eines Satelliten verwendet werden. Diese beiden Befehle konnten mit dem Skript *rotctl.bat* erfolgreich für die Steuerung der beiden Rotoren getestet werden. Grundsätzlich sind bei der Steuerung der Rotoren einige Aspekte zu beachten, beispielsweise, dass die Rotorgeschwindigkeit reduziert wird, sobald sich der Ist-Winkel in gewisser Nähe zum Soll-Winkel befindet. Diese Punkte werden jedoch alle entweder von den Steuerungsmodulen oder von der Software *ARSVCOM* übernommen, sodass in GPredict lediglich die jeweiligen Winkel ausgelesen und gesetzt werden müssen.

### 5.4.5 Konfiguration des Funkgeräts

Dem Abschnitt 3.2.3 kann entnommen werden, dass das Funkgerät IC-9100 vollduplexfähig ist und bei Betrieb im Satellitenmodus nur im *Sub-Band* senden und im *Main-Band* empfangen darf. Diese Informationen können in den *Interface-Einstellungen* wie in Abbildung 5.17 zu sehen, eingestellt werden.

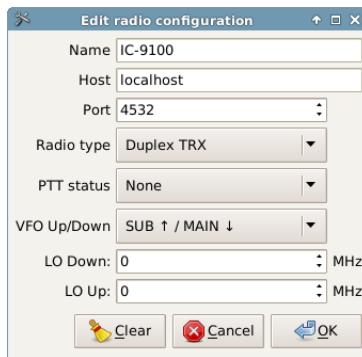


Abbildung 5.17: Funkgerät-Konfiguration

Bei *Host* muss die IP-Adresse angegeben werden unter der aktive *rigctld.exe*-Prozess zu erreichen ist. Hier ist standardmäßig *localhost* eingetragen, was für den Computer steht, auf dem GPredict ausgeführt wird. Da im Rahmen dieses Projekts GPredict und *rigctld.bat* immer auf dem gleichen Computer ausgeführt werden, wird sowohl unter GPredict als auch beim Skript *rigctld.bat* die IP-Adresse 127.0.0.1 (*localhost*) eingestellt. Hierbei ist darauf zu achten, dass die Verwendung der Zeichenkette *localhost* unter Windows bei der Verwendung von *rigctld.exe* nicht funktioniert. Die Angabe des Ports der *Interface-Einstellungen* muss ebenfalls dem in *rigctld.bat* eingestellten Port entsprechen. Die Felder *LO Down* und *LO Up* ermöglichen die Berücksichtigung eines Transverters, welcher zur Veränderung der Empfangs- oder Sendefrequenz eingesetzt wird. Somit kann beispielsweise mit einem 28 MHz-Empfänger ein Signal von einem Satellit der mit 144 MHz sendet, empfangen werden, in dem  $144\text{ MHz} - 28\text{ MHz} = 116\text{ MHz}$  bei *LO Down* eingetragen wird. Beim Feld *PTT status* wird *None* eingetragen, da bei vollduplexfähigen Funkgeräten die Abfrage des Push-to-Talk (PTT)-Status nicht nötig ist (siehe [18] Seite 58).

### 5.4.6 Konfiguration der Rotoren

Bei der Konfiguration der Rotorensteuerung ist die Angabe der IP-Adresse und des Ports analog wie bei der Funkgerät-Konfiguration vorzunehmen, bis auf die Ausnahme, dass hier ein anderer Port eingestellt werden muss (siehe Abbildung 5.18).

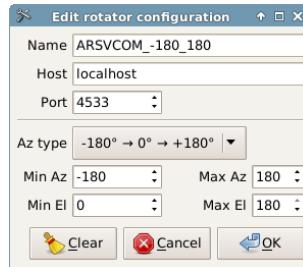


Abbildung 5.18: Konfigurationen der Rotorensteuerung

Der Elevations-Rotor lässt sich von  $0^\circ$  bis  $180^\circ$  auslenken, dementsprechend wird die Einstellung bei *Min El* und *Max El* gewählt. Der Azimut-Rotor lässt sich um  $360^\circ$  auslenken und besitzt einen Anschlag bei  $180^\circ$  (Süden), das heißt er lässt sich von  $0^\circ$  (Norden) aus jeweils um  $180^\circ$  in Richtung Süden drehen. Deshalb wird beim Feld *Az type* der Bereich von  $-180^\circ$  bis  $180^\circ$  eingestellt, woraufhin automatisch bei *Min Az*  $-180^\circ$  und bei *Max Az*  $180^\circ$  eingetragen wird.

Passiert ein Satellit, wie in Abbildung 5.19 beispielhaft zu sehen, die Linie zwischen dem Mittelpunkt des Polardiagramms und der südlichen Richtung (also den Anschlag des Azimut-Rotors), ist ein Rotorflip notwendig.

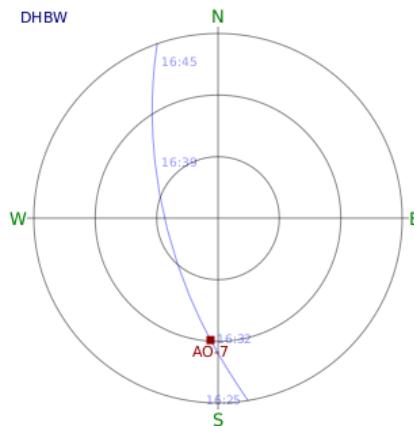


Abbildung 5.19: Verlauf eines Satelliten über den Anschlag des Azimut-Rotors

Bei einem Rotorflip wird der Elevations-Rotor über die  $90^\circ$ -Schwelle gekippt, womit der Anschlag des Azimut-Rotors um  $180^\circ$ , also nach Norden, verlegt wird. Somit ist es möglich einen Satelliten mit dem in Abbildung 5.19 zu sehenden Verlauf zu verfolgen.

### 5.4.7 Modifizierung des HamLib-Quellcodes

Um fehlerlos einen Rotorflip durchführen zu können, ist es notwendig die Quelldatei, in der das Backend für die verwendete Rotorschnittstelle implementiert ist, anzupassen. Dafür wurden insgesamt drei Änderungen in der Datei `gs232.c`, die unter dem Pfad `hamlib-3.0.1/gs232a/` des HamLib-Quellcodes zu finden ist, vorgenommen. Die modifizierte Datei ist im Anhang Anhang H: zu finden.

Die erste Änderung wurde in der Funktion `gs232_rot_set_position()` in Zeile 141 vorgenommen. Hier wurde die folgende Zeile eingefügt um sicherzustellen, dass der Azimutwert, der an den Rotor geschickt wird, immer zwischen  $0^\circ$  und  $360^\circ$  liegt:

```
if (az < 0.0) az += 360.0;
```

Die zweite Änderung wurde in der Funktion `gs232_rot_get_position()` in Zeile 171 vorgenommen. Durch Einfügen der folgenden Zeile, wird sichergestellt, dass der vom Rotor gelesene Azimutwert immer zwischen  $-180^\circ$  und  $180^\circ$  liegt, sodass er der Rotorkonfiguration von GPredict entspricht:

```
if (*az > 180.0) *az -= 360.0;
```

Die dritte Änderung befindet sich in Zeile 224. Hier wurde bei den Fähigkeiten des Rotor-Backends der minimale Azimutwert von  $0^\circ$  auf  $-180^\circ$  abgeändert, wie es im Folgenden zu sehen ist:

```
.min_az = -180.0,
```

### 5.4.8 Kompilieren der modifizierten HamLib für Windows

Um diese Anpassungen des Quellcodes im Programm `rotctl(d).exe` nutzen zu können, ist es erforderlich den Quellcode von HamLib nachdem die Änderungen vorgenommen wurden für Windows zu kompilieren. Wie dabei vorgegangen werden muss, ist im Folgenden schrittweise aufgeführt. Hierfür wurde ein Rechner, auf dem Debian 8.5 (64 bit) installiert ist, verwendet.

1. Zuerst lädt man den HamLib-Quellcode, der unter [20] verfügbar ist, herunter, entpackt das Archiv *hamlib-3.0.1.tar.gz* und verschiebt den darin enthaltenen Ordner *hamlib-3.0.1* ins Verzeichnis */home/<username>/builds/*. Das bei Schritt 5 erwähnte Skript bezieht sich auf diesen absoluten Pfad.
2. Als nächstes lädt man das Archiv *libusb-win32-bin-1.2.4.0.zip* herunter (siehe [23]) und entpackt es ebenfalls in das im vorherigen Schritt genannte Verzeichnis.
3. Da der Quellcode für die Verwendung unter Windows kompiliert werden soll, müssen unter Linux entsprechende Softwarekomponenten installiert sein. Diese können mit folgenden Befehlen installiert werden:

```
$ sudo apt-get update
$ sudo apt-get install mingw32 wine
```

4. Nun navigiert man in einer Bash-Kommandozeile in den Ordner *hamlib-3.0.1* der im ersten Schritt erstellt wurde und führt folgenden Befehl aus:

```
$ ./configure -host=i586-mingw32msvc
```

5. Anschließend markiert man das Bash-Skript *build-win32.sh* im Verzeichnis *./scripts/* mit folgendem Befehl als ausführbar:

```
$ chmod +x build-win32.sh
```

6. Als letztes führt man das Bash-Skript *build-win32.sh* mit folgendem Befehl aus und findet nach erfolgreichem Erstellungsvorgang das Archiv *hamlib-win32-3.0.1.zip* im Ordner *hamlib-3.0.1*. In diesem Archiv ist nun die vollständige HamLib für Windows enthalten.

```
$ ./build-win32.sh hamlib-3.0.1
```

Nähere Details zu den oben genannten Schritten können aus folgenden Dateien des HamLib-Quellcodes (im Ordner *hamlib-3.0.1*) entnommen werden:

- **INSTALL** (Abschnitt „MS Windows“)
- **scripts/README.build-win32** (gesamte Datei)

### 5.4.9 Probelauf und Testergebnisse

Für den korrekten Betrieb von GPredict ist es notwendig folgende Softwarekomponenten zu starten:

- Die beiden Skripte *rigctld.bat* und *rotctld.bat*
- Die Software ARSVCOM
- Die Standard-Benutzeroberfläche von GPredict
- Die Bedienfelder für die Funkgerät- und Rotorensteuerung

Es konnte durch mehrere Probelaufe der grundsätzliche Betrieb von GPredict sichergestellt werden. Dazu gehören das erfolgreiche Ausrichten der Antenne, also des Azimut- und Elevations-Rotors, sowie die Übermittlung der korrekten Frequenz (inklusive Korrektur der Doppler-Verschiebung) zum Funkgerät. Allerdings muss an dieser Stelle auf gewisse Einschränkungen hingewiesen werden, die den reibungslosen Betrieb der Bodenstation mit GPredict noch nicht ganz ermöglichen. Diese sind im Folgenden näher erläutert:

Möchte man, nachdem die Kommunikation zwischen GPredict und dem Funkgerät erfolgreich aufgebaut ist und diese auch schon eine gewisse Zeit ohne Unterbrechung läuft, von einem Transponder des aktuellen Satelliten auf einen anderen Transponder wechseln, bei dem die *Down*- und *Uplink*-Frequenz genau vertauscht sind, bricht die Kommunikation zum Funkgerät ab. Das hat damit zu tun, dass es nicht erlaubt ist, im Satellitenmodus des Funkgeräts die gleiche Frequenz im *Sub*- und *Main*-Band einzustellen. Dafür gibt es beim CI-V Protokoll einen speziellen Befehl, der einen Tausch der Frequenzen im *Sub*- und *Main*-Band veranlasst. Dieser Befehl wird nicht von dem HamLib-Programm *rigctld.exe* geschickt. Beim Test unter SatPC32 ist dieser Wechsel möglich und es konnte nachgewiesen werden, dass dort dieser Befehl gesendet wird. Um diese Funktionalität in GPredict zu implementieren, ist ein Eingriff in den Quellcode nötig.

Wie bereits unter Abschnitt 5.4.3 auf Seite 62 beschrieben, ist der Befehl zur Aktivierung des Satellitenmodus des Funkgeräts in HamLib fehlerhaft implementiert. Außerdem wurde festgestellt, dass dieser Befehl beim Kommunikationsaufbau auch nicht vom Programm *rigctld.exe* an das Funkgerät geschickt wird (bei SatPC32 konnte dies nachgewiesen werden). Somit ist es notwendig, das Funkgerät manuell in den Satellitenmodus zu setzen

um eine fehlerfreie Kommunikation zu gewährleisten. Um diese Funktionalität zu implementieren, führt ebenfalls kein Weg daran vorbei, den Quellcode von HamLib anzupassen und diesen anschließend neu zu kompilieren.

Der letzte Punkt, der einem reibungslosen Betrieb von GPredict im Weg stand, hat mit der Ansteuerung des Azimut-Rotors zu tun. Hier war es notwendig einen Eingriff in den HamLib-Quellcode vorzunehmen (siehe Abschnitt 5.4.7) und diesen daraufhin neu zu kompilieren (siehe Abschnitt 5.4.8). Ohne diesen Eingriff kam es zu einer fehlerhaften Ansteuerung des Azimut-Rotors beim Verfolgen eines Satelliten. Ein Test nach der Modifizierung des HamLib-Quellcodes lieferte zufriedenstellende Ergebnisse.

# 6 Zusammenfassung und Ausblick

In dieser Studienarbeit, Inbetriebnahme einer freien Software zur Satellitenbahnvorhersage und Ansteuerung einer Hochleistungsantenne, wurde eine freie Software validiert und in Betrieb genommen. Durch eine umfassende Recherche wurde die freie Software GPredict als eine sehr gute Alternative zu der Referenzsoftware SatPC32 verifiziert.

GPredict ist ein Satelliten-Tracking Programm und ermöglicht eine Anbindung an die Antennenrotoren sowie an das Funkgerät. Unter der Verwendung der TLE berechnet GPredict die jeweilige Satellitenbahn. Um zu verstehen, wie GPredict eine Bahnvorhersage berechnet, wurden die physikalischen Hintergründe theoretisch beschrieben und die Berechnungen der Keplerelemente hergeleitet.

Um GPredict mit der Hardwareumgebung anzubinden, wurde eine entsprechende Konfiguration von GPredict vorgenommen. Die Funkgerät- und Antennensteuerung erfolgt mittels der Hamlib-Bibliotheken. Dafür wurden eigene Batch-Skripte geschrieben um die Verwendung der verfügbaren Kommandozeilenprogramme zu vereinfachen.

Für die Validierung der Anforderungsdefinition, wurden definierte Tests durchgeführt. Dabei handelte es sich um Modultests, Integrationstests und Systemtests.

Zusammenfassend wurde die Studienarbeit gemäß dem V-Modell bearbeitet und strukturiert zu einem Ergebnis gebracht. Dieses Ergebnis korreliert mit der Anforderungsdefinition und bietet eine Grundlage für weitere Studienarbeiten an der Bodenstation DHBW Ravensburg Campus Friedrichshafen. Dabei wäre eine weitere Aufgabe die Implementierung eines Befehls zum Tausch von Sub- und Main-Band.

Im Rahmen dieser Studienarbeit wurden die Inhalte der Nachrichtentechnik und des technischen Managements reflektiert und ein neues Gebiet der Bahnmechanik erschlossen. Für die Unterstützung und Betreuung dieser Studienarbeit möchten wir uns recht herzlich bei Herrn Hardy Lau bedanken.

# Abbildungsverzeichnis

2.1	V-Modell, Quelle: [1]	3
2.2	Projektstrukturplan	5
2.3	Arbeitspakete	7
2.4	Gantt-Diagramm	8
3.1	Cubesat, Quelle: [4]	10
3.2	AMSAT Live OSCAR Satellite Status, Quelle: [6]	12
3.3	Antennen	13
3.4	Rotorsteuerung	14
3.5	Funkgerät ICOM IC-9100, Quelle: [7]	15
4.1	Bahnelemente	16
4.2	Ellipse	18
4.3	Kegelschnitt	20
4.4	Kegelschnitt	21
4.5	Bahnelemente	25
4.6	Kepler-Gleichung	28
4.7	Sternzeit	35
4.8	Geografische Koordinaten	36
4.9	TLE	38
4.10	Vorhersagegenauigkeit	39
5.1	Prinzip eines Satellitenverfolgungsprogramms, Quelle: [18]	42
5.2	Standardoberfläche von GPredict	43
5.3	<i>List View</i>	45
5.4	<i>Upcoming Passes View</i>	45
5.5	<i>Satellite Info</i>	46
5.6	<i>Pass Details</i>	46
5.7	<i>Upcoming Passes</i>	47
5.8	<i>The sky at a glance</i>	47
5.9	<i>Time Controller</i>	48
5.10	<i>Edit Module</i>	48

5.11 Rotorensteuerungs-Bedienfeld . . . . .	49
5.12 Funkgerät-Steuerung . . . . .	51
5.13 GPredict-Einstellungen, Kategorie <i>General</i> . . . . .	53
5.14 HamLib Design, Quelle: [21] . . . . .	56
5.15 Kommunikation zwischen Hard- und Software . . . . .	59
5.16 Schnittstellenkonfiguration für das Dummy-Interface . . . . .	60
5.17 Funkgerät-Konfiguration . . . . .	63
5.18 Konfigurationen der Rotorensteuerung . . . . .	64
5.19 Verlauf eines Satelliten über den Anschlag des Azimut-Rotors . . . . .	64

Alle nicht eigens nachgewiesenen Abbildungen stammen von den Autoren.

# Tabellenverzeichnis

3.1	Abkürzungen der Transpondermodi	11
5.1	Betriebsmodi des <i>Antenna Control</i> -Bedienfelds, Quelle: [18]	50
5.2	Betriebsmodi des <i>Radio Control</i> -Bedienfelds, Quelle: [18]	52
5.3	Fehlerhaftes CI-V Kommando beim Aktivieren des Satellitenmodus	62

# Literatur- und Quellenverzeichnis

- [1] P. D. I. habil. Klaus-Peter Fähnrich, *Softwaretechnik - Vorgehensmodelle*, Deutsch. Universität Leipzig, Institut für Informatik, Betriebliche Informationssysteme, 2009.
- [2] W. F. Inc. (2015). Satellitenorbit, Adresse: [http://www.linkfang.de/wiki/Satellitenorbit#Highly\\_Elliptical\\_Orbit\\_.28HE0.29](http://www.linkfang.de/wiki/Satellitenorbit#Highly_Elliptical_Orbit_.28HE0.29) (besucht am 04.06.2016).
- [3] ——, (2015). Amateurfunksatellit, Adresse: <https://de.wikipedia.org/wiki/Amateurfunksatellit> (besucht am 24.05.2016).
- [4] C. Lab. (2016). ULA Releases Application for University CubeSat Competition, Adresse: <http://www.cubesat.org/> (besucht am 24.05.2016).
- [5] M. Bopp. (2013). Amateurfunk über Satelliten, Adresse: <http://www.dd1us.de/Downloads/Amateurfunk%20ueber%20Satelliten%2020130406%203v1.pdf> (besucht am 04.06.2016).
- [6] R. A. S. Corporation. (2013). AMSAT Live OSCAR Satellite Status Page, Adresse: <http://www.amsat.org/status/> (besucht am 04.06.2016).
- [7] H. Lau. (2015). Satelliten Bodenstation, Adresse: <http://dk0te.dhbw-ravensburg.de/cgi-bin/navi?m=SAT> (besucht am 08.06.2016).
- [8] P. D. I. A. Linkohr, *Hochfrequenztechnik*, Deutsch. DHBW Ravensburg, 2016.
- [9] W. Steiner und M. Schlagerl, *Raumflugmechanik*, Deutsch, 1. Aufl. Berlin Heidelberg: Springer Verlag, 2004.
- [10] W. F. Inc. (2016). Johannes Kepler, Adresse: [https://en.wikipedia.org/wiki/Johannes\\_Kepler](https://en.wikipedia.org/wiki/Johannes_Kepler) (besucht am 05.05.2016).
- [11] ——, (2016). Ellipse, Adresse: <https://de.wikipedia.org/wiki/Ellipse> (besucht am 19.06.2016).
- [12] W. Ley, K. Wittmann, W. Hallmann u.a., *Handbuch der Raumfahrttechnik*, Deutsch, 3. Aufl. München: Carl Hanser Verlag, 2008.
- [13] W. F. Inc. (2016). Bahnelement, Adresse: <https://de.wikipedia.org/wiki/Bahnelement> (besucht am 01.05.2016).
- [14] ——, (2016). Frühlingspunkt, Adresse: <https://de.wikipedia.org/wiki/Fr%C3%BChlingspunkt> (besucht am 25.05.2016).
- [15] ——, (2016). Kepler-Gleichung, Adresse: <https://de.wikipedia.org/wiki/Kepler-Gleichung> (besucht am 27.05.2016).

- [16] ——, (2016). Geografische Koordinaten, Adresse: [https://de.wikipedia.org/wiki/Geographische\\_Koordinaten](https://de.wikipedia.org/wiki/Geographische_Koordinaten) (besucht am 31.05.2016).
- [17] ——, (2016). Erdradius, Adresse: <https://de.wikipedia.org/wiki/Erdradius> (besucht am 31.05.2016).
- [18] *GPredict User Manual*. Adresse: <https://sourceforge.net/projects/gpredict/files/Gpredict/1.3/gpredict-user-manual-1.3.pdf/download> (besucht am 01.05.2016).
- [19] *Sourceforge.net: GPredict 1.3*. Adresse: <https://sourceforge.net/projects/gpredict/files/Gpredict/1.3/> (besucht am 01.05.2016).
- [20] *HamLib Download*. Adresse: <https://sourceforge.net/projects/hamlib/> (besucht am 31.05.2016).
- [21] *HamLib Manual*. Adresse: <http://hamlib.sourceforge.net/manuals/hamlib.html> (besucht am 13.05.2016).
- [22] *IC-9100 Instruction Manual*. Adresse: [http://www.icom-australia.com/products/amateur/ic-9100/Amateur\\_IC-9100\\_Instruction\\_Manual.pdf](http://www.icom-australia.com/products/amateur/ic-9100/Amateur_IC-9100_Instruction_Manual.pdf) (besucht am 24.05.2016).
- [23] *Libusb-win32 Download*. Adresse: <https://sourceforge.net/projects/libusb-win32/> (besucht am 10.07.2016).

# Anhang A: CD

## Inhalt

- Studienarbeit: *T3X00.pdf*
- Präsentations-Handout zur Studienarbeit: *T3X00-Handout.pdf*
- Skripte:
  - *rigctl*.bat
  - *rigctl*.bat
  - *rotctl*.bat
  - *rotctl*.bat
  - *rigctl-dummy*.bat
  - *rotctl-dummy*.bat
- Modifizierte Datei: *gs232.c*
- Kompilierter HamLib-Quellcode für Windows: *hamlib-3.0.1.zip*

## Anhang B: Skript: *rigctld.bat*

rigctld.bat

1/1

```
1  @rem Dies ist ein Skript zur Ausfhrung von rigctld.exe.
2  @title Batch Skript: rigctld.exe
3  @echo off
4
5  echo =====#
6  echo ##
7  echo # Batch Skript zur Ausfehrung von #
8  echo ##
9  echo # rigctld.exe #
10 echo ##
11 echo =====#
12 echo.
13 echo # Ausfehrung von rigctld.exe mit folgenden Parametern:
14 echo.
15 set mycmd=rigctld.exe -vvvv -m 368 -r COM5 -c 0x7C -s 19200 -C stop_bits=1,data_bits=8,write_delay=
5,post_write_delay=40,timeout=1000 -T 127.0.0.1 -t 4532
16 echo %mycmd%
17 echo.
18 echo =====#
19 echo.
20 echo # Ausgabe von rigctld.exe:
21 echo.
22 %mycmd%
23
```

# Anhang C: Skript: *rigctl.bat*

---

**rigctl.bat** 1/1

```
1  @rem Dies ist ein Skript zur Ausf&uuml;hrung von rigctl.exe.
2  @title Batch Skript: rigctl.exe
3  @echo off
4
5  echo =====#
6  echo ##
7  echo # Batch Skript zur Ausf&uuml;hrung von #
8  echo ##
9  echo # rigctl.exe #
10 echo ##
11 echo =====#
12 echo.
13 echo # Ausf&uuml;hrung von rigctl.exe mit folgenden Parametern:
14 echo.
15 set mycmd=rigctl.exe -vvvv -m 368 -r COM5 -c 0x7C -s 19200 -C stop_bits=1,data_bits=8,post_write_delay=40
16 echo %mycmd%
17 echo.
18 echo =====#
19 echo.
20 echo # Ausgabe von rigctl.exe:
21 echo.
22 %mycmd%
23
```

## Anhang D: Skript: *rotctld.bat*

---

**rotctld.bat** 1/1

```
1  @rem Dies ist ein Skript zur Ausfhrung von rotctld.exe.
2  @title Batch Skript: rotctld.exe
3  @echo off
4
5  echo =====#
6  echo ##
7  echo # Batch Skript zur Ausfehrung von #
8  echo ##
9  echo # rotctld.exe #
10 echo ##
11 echo =====#
12 echo.
13 echo # Ausfehrung von rotctld.exe mit folgenden Parametern:
14 echo.
15 set mycmd=rotctld.exe -vvvv -m 601 -r COM10 -s 9600 -C stop_bits=1,data_bits=8,min_az=-180,max_az=
16 180,min_el=0,max_el=180 -T 127.0.0.1 -t 4533
16 echo %mycmd%
17 echo.
18 echo =====#
19 echo.
20 echo # Ausgabe von rotctld.exe:
21 echo.
22 %mycmd%
23
```

# Anhang E: Skript: *rotctl.bat*

---

**rotctl.bat** 1/1

```
1  @rem Dies ist ein Skript zur Ausf&uuml;hrung von rotctl.exe.
2  @title Batch Skript: rotctl.exe
3  @echo off
4
5  echo =====#
6  echo ##
7  echo # Batch Skript zur Ausf&uuml;hrung von #
8  echo ##
9  echo # rotctl.exe #
10 echo ##
11 echo =====#
12 echo.
13 echo # Ausf&uuml;hrung von rotctld.exe mit folgenden Parametern:
14 echo.
15 set mycmd=rotctl.exe -vvvv -m 601 -r COM10 -s 9600 -C stop_bits=1,data_bits=8,min_az=-180,max_az=
16 180,min_el=0,max_el=180
16 echo %mycmd%
17 echo.
18 echo =====#
19 echo.
20 echo # Ausgabe von rotctld.exe:
21 echo.
22 %mycmd%
23
```

# Anhang F: Skript: *rigctld-dummy.bat*

```
rigctld-dummy.bat
1  @rem Dies ist ein Skript zur Ausfhrung von rigctld.exe mit der Dummy-Schnittstelle
2  @title Batch Skript: rigctld.exe - Dummy
3  @echo off
4
5  echo =====#
6  echo ##
7  echo # Batch Skript zur Ausfehrung von #
8  echo ##
9  echo # rigctld.exe - Dummy #
10 echo ##
11 echo =====#
12 echo.
13 echo # Ausfehrung von rigctld.exe mit folgenden Parametern:
14 echo.
15 set mycmd=rigctld.exe -vvvv -m 1 -T 127.0.0.1 -t 4532
16 echo %mycmd%
17 echo.
18 echo =====#
19 echo.
20 echo # Ausgabe von rigctld.exe:
21 echo.
22 %mycmd%
23
```

1/1

# Anhang G: Skript: *rotctld-dummy.bat*

---

**rotctld-dummy.bat**

1 *@rem Dies ist ein Skript zur Ausführung von rotctld.exe mit der Dummy-Schnittstelle.*  
2 *@title Batch Skript: rotctld.exe - Dummy*  
3 *@echo off*  
4  
5 **echo =====**  
6 **echo #**  
7 **echo #** Batch Skript zur Ausfuehrung von **#**  
8 **echo #**  
9 **echo #** rotctld.exe - Dummy **#**  
10 **echo #**  
11 **echo =====**  
12 **echo.**  
13 **echo #** Ausfuehrung von rotctld.exe mit folgenden Parametern:  
14 **echo.**  
15 **set mycmd=rotctld.exe -vvvv -m 1 -T 127.0.0.1 -t 4533**  
16 **echo %mycmd%**  
17 **echo.**  
18 **echo =====**  
19 **echo.**  
20 **echo #** Ausgabe von rotctld.exe:  
21 **echo.**  
22 **%mycmd%**  
23

1/1

# Anhang H: Modifizierte Datei: gs232.c

gs232.c

1/4

```
1  /*
2   *  Hamlib Rotator backend - GS-232
3   *  Copyright (c) 2001-2010 by Stephane Fillion
4   *  Copyright (c) 2009 by Jason Winnigham
5   *
6   *  This library is free software; you can redistribute it and/or
7   *  modify it under the terms of the GNU Lesser General Public
8   *  License as published by the Free Software Foundation; either
9   *  version 2.1 of the License, or (at your option) any later version.
10  *
11  *  This library is distributed in the hope that it will be useful,
12  *  but WITHOUT ANY WARRANTY; without even the implied warranty of
13  *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
14  *  Lesser General Public License for more details.
15  *
16  *  You should have received a copy of the GNU Lesser General Public
17  *  License along with this library; if not, write to the Free Software
18  *  Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
19  *
20  */
21
22 #ifdef HAVE_CONFIG_H
23 #include "config.h"
24 #endif
25
26 #include <stdio.h>
27 #include <stdlib.h>
28 #include <string.h> /* String function definitions */
29 #include <unistd.h> /* UNIX standard function definitions */
30 #include <math.h>
31
32 #include "hamlib/rotator.h"
33 #include "serial.h"
34 #include "misc.h"
35 #include "register.h"
36
37 #include "gs232a.h"
38
39 #define EOM "\r"
40 #define REPLY_EOM "\r"
41
42 #define BUFSZ 64
43
44 /**
45  * gs232_transaction
46  *
47  * cmdstr - Command to be sent to the rig.
48  * data - Buffer for reply string. Can be NULL, indicating that no reply is
49  *        needed, but answer will still be read.
50  * data_len - in: Size of buffer. It is the caller's responsibility to provide
51  *        a large enough buffer for all possible replies for a command.
52  *
53  * returns:
54  *        RIG_OK - if no error occurred.
55  *        RIG_EIO - if an I/O error occurred while sending/receiving data.
56  *        RIGETIMEOUT - if timeout expires without any characters received.
57  *        RIG_REJECTED - if a negative acknowledge was received or command not
58  *                      recognized by rig.
59  */
60 static int
61 gs232_transaction (ROT *rot, const char *cmdstr,
62                     char *data, size_t data_len)
63 {
64     struct rot_state *rs;
65     int retval;
66     int retry_read = 0;
67     char replybuf[BUFSZ];
68
69     rs = &rot->state;
70
71     transaction_write:
72
73     serial_flush(&rs->rotport);
74
75     if (cmdstr) {
```

## gs232.c

```

76         retval = write_block(&rs->rotport, cmdstr, strlen(cmdstr));
77         if (retval != RIG_OK)
78             goto transaction_quit;
79     }
80
81     /* Always read the reply to know whether the cmd went OK */
82     if (!data)
83         data = replybuf;
84     if (!data_len)
85         data_len = BUFSZ;
86
87     memset(data, 0, data_len);
88     retval = read_string(&rs->rotport, data, data_len, REPLY_EOM, strlen(REPLY_EOM));
89     if (retval < 0) {
90         if (retry_read++ < rot->state.rotport.retry)
91             goto transaction_write;
92         goto transaction_quit;
93     }
94
95 #if 0
96     /* Check that command termination is correct */
97     if (strchr(REPLY_EOM, data[strlen(data)-1]) == NULL) {
98         rig_debug(RIG_DEBUG_ERR, "%s: Command is not correctly terminated '%s'\n", __FUNCTION__, data);
99         if (retry_read++ < rig->state.rotport.retry)
100             goto transaction_write;
101         retval = -RIG_EPROTO;
102         goto transaction_quit;
103     }
104 #endif
105
106     if (data[0] == '?') {
107         /* Invalid command */
108         rig_debug(RIG_DEBUG_VERBOSE, "%s: Error for '%s': '%s'\n",
109                  __FUNCTION__, cmdstr, data);
110         retval = -RIG_EPROTO;
111         goto transaction_quit;
112     }
113
114     retval = RIG_OK;
115 transaction_quit:
116     return retval;
117 }
118
119
120
121 /*
122  * write-only transaction, no data returned by controller
123 */
124 static int
125 gs232_wo_transaction (ROT *rot, const char *cmdstr,
126                         char *data, size_t data_len)
127 {
128     return write_block(&rot->state.rotport, cmdstr, strlen(cmdstr));
129 }
130
131
132 static int
133 gs232_rot_set_position(ROT *rot, azimuth_t az, elevation_t el)
134 {
135     char cmdstr[64];
136     int retval;
137     unsigned u_az, u_el;
138
139     rig_debug(RIG_DEBUG_TRACE, "%s called: %f %f\n", __FUNCTION__, az, el);
140
141     if (az < 0.0) az += 360.0; /* added to ensure proper use with DHBW Azimut-Rotor */
142     u_az = (unsigned)rint(az);
143     u_el = (unsigned)rint(el);
144
145     sprintf(cmdstr, "W%03u %03u" EOM, u_az, u_el);
146     retval = gs232_wo_transaction(rot, cmdstr, NULL, 0);
147
148     if (retval != RIG_OK)
149         return retval;
150

```

## gs232.c

```
151     return RIG_OK;
152 }
153
154 static int
155 gs232_rot_get_position(ROT *rot, azimuth_t *az, elevation_t *el)
156 {
157     char posbuf[32];
158     int retval;
159
160     rig_debug(RIG_DEBUG_TRACE, "%s called\n", __FUNCTION__);
161
162     retval = gs232_transaction(rot, "C2" EOM, posbuf, sizeof(posbuf));
163     if (retval != RIG_OK || strlen(posbuf) < 10)
164         return retval;
165
166     /* parse */
167     if (sscanf(posbuf+2, "%f", az) != 1) {
168         rig_debug(RIG_DEBUG_ERR, "%s: wrong reply '%s'\n", __FUNCTION__, posbuf);
169         return -RIG_EPROTO;
170     }
171     if (*az > 180.0) *az -= 360.0; /* added to ensure proper use with DHBW Azimut-Rotor */
172     if (sscanf(posbuf+7, "%f", el) != 1) {
173         rig_debug(RIG_DEBUG_ERR, "%s: wrong reply '%s'\n", __FUNCTION__, posbuf);
174         return -RIG_EPROTO;
175     }
176
177     rig_debug(RIG_DEBUG_TRACE, "%s: (az, el) = (%.1f, %.1f)\n",
178             __FUNCTION__, *az, *el);
179
180     return RIG_OK;
181 }
182
183 static int
184 gs232_rot_stop(ROT *rot)
185 {
186     int retval;
187
188     rig_debug(RIG_DEBUG_TRACE, "%s called\n", __FUNCTION__);
189
190     /* All Stop */
191     retval = gs232_wo_transaction(rot, "S" EOM, NULL, 0);
192     if (retval != RIG_OK)
193         return retval;
194
195     return RIG_OK;
196 }
197
198
199 /* **** */
200 /*
201 * Generic GS232 (not A, not B) rotator capabilities.
202 */
203
204 const struct rot_caps gs232_rot_caps = {
205     .rot_model = ROT_MODEL_GS232,
206     .model_name = "GS-232",
207     .mfg_name = "Yaesu/Kenpro",
208     .version = "0.1",
209     .copyright = "LGPL",
210     .status = RIG_STATUS_BETA,
211     .rot_type = ROT_TYPE_AZEL,
212     .port_type = RIG_PORT_SERIAL,
213     .serial_rate_min = 150,
214     .serial_rate_max = 9600,
215     .serial_data_bits = 8,
216     .serial_stop_bits = 1,
217     .serial_parity = RIG_PARITY_NONE,
218     .serial_handshake = RIG_HANDSHAKE_NONE,
219     .write_delay = 0,
220     .post_write_delay = 0,
221     .timeout = 400,
222     .retry = 3,
223
224     .min_az = -180.0, /* changed from 0.0 to -180.0 to ensure proper use with DHBW Azimut-Rotor */
225     .max_az = 450.0, /* vary according to rotator type */
226 }
```

```
gs232.c
226     .min_el =     0.0,
227     .max_el =    180.0,
228
229     .get_position = gs232_rot_get_position,
230     .set_position = gs232_rot_set_position,
231     .stop =        gs232_rot_stop,
232 };
233
234
235 /* **** */
236 /*
237 * F1TE Tracker, GS232 without position feedback
238 *
239 * http://www.f1te.org/index.php?option=com_content&view=article&id=19&Itemid=39
240 */
241
242 const struct rot_caps f1tetracker_rot_caps = {
243     .rot_model =      ROT_MODEL_F1TETRACKER,
244     .model_name =    "GS232/F1TE Tracker",
245     .mfg_name =      "F1TE",
246     .version =        "0.1",
247     .copyright =     "LGPL",
248     .status =         RIG_STATUS_BETA,
249     .rot_type =       ROT_TYPE_AZEL,
250     .port_type =     RIG_PORT_SERIAL,
251     .serial_rate_min = 150,
252     .serial_rate_max = 9600,
253     .serial_data_bits = 8,
254     .serial_stop_bits = 1,
255     .serial_parity =  RIG_PARITY_NONE,
256     .serial_handshake = RIG_HANDSHAKE_NONE,
257     .write_delay = 0,
258     .post_write_delay = 0,
259     .timeout = 400,
260     .retry = 0,
261
262     .min_az =     0.0,
263     .max_az =    360.0, /* vary according to rotator type */
264     .min_el =     0.0,
265     .max_el =    180.0,
266
267     .get_position = NULL, /* no position feedback available */
268     .set_position = gs232_rot_set_position,
269 #if 0
270     .stop =        gs232_rot_stop,
271 #endif
272 };
273
274
```