



Patrick Kriegl, BSc

Evaluierung eines proprietären Datenformats für den Austausch von GNSS Rohdaten anhand der Software RTKLIB

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Geomatics Science

eingereicht an der

Technischen Universität Graz

Betreuer

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Manfred Wieser

Institut für Geodäsie

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich im Laufe meines Studiums unterstützt haben.

Ein besonderer Dank gilt Prof. Dipl.-Ing. Dr.techn. Manfred Wieser für die Betreuung und Unterstützung im Laufe dieser Arbeit, sowie für die Ermöglichung der Mitarbeit als studentischer Projektmitarbeiter im Zuge von Forschungsprojekten am Institut für Geodäsie.

Außerdem danke ich im Besonderen Dipl.-Ing. Daniel Koch, der mir im Laufe der Arbeit mit viel Zeiteinsatz für offene Fragen zur Verfügung stand und mir immer mit hilfreichen Informationen weiterhalf.

Besonders bedanken möchte ich mich auch beim gesamten Team des Instituts für Geodäsie für die moralisch und fachliche Unterstützung im Zuge dieser Arbeit. Ein herzlicher Dank gilt auch allen Studienkollegen meines Jahrgangs und Freunden, insbesondere meinem Mitbewohner für die unvergessliche Studienzeit in Graz.

Schließlich möchte ich mich vor allem bei meiner Familie, im Speziellen bei meinen Eltern und Großeltern, die mir diese Ausbildung ermöglicht haben, für die Unterstützung bedanken.

Kurzfassung

Für die kostengünstige Bestimmung von Position, Geschwindigkeit und Bewegungsrichtung eines Objektes werden in der Navigation hauptsächlich low-cost GNSS, INS, UWB oder Barometerdaten verwendet. Ziel dieser Arbeit ist es, die entstehende Datenflut für robuste Navigationsanwendungen aufzubereiten und zu evaluieren. Dazu werden ein proprietäres Datenmanagement für die einzelnen Komponenten, sowie eine eigens konzipierte Systemarchitektur für die Datenübertragung der Navigationsdaten zum Einsatz kommen.

Der stetig wachsende low-cost GNSS-Markt bietet unterschiedlichste GNSS-Empfänger für jegliche kostengünstige Navigationsanwendungen an. Verantwortlich für die Rohdatenübertragung, welche vor allem bei RTK Navigationsanwendungen zwingend notwendig ist, ist eine eigens in dieser Arbeit entwickelte innovative Systemarchitektur. Diese besteht aus einer eigenen Datenübertragung, einem proprietären Navigation-Rohdatenformat und einer Weiterentwicklung einer vorhandenen Software für die verbesserte Positionierung. Herkömmliche, standardisierte Formate reizen nicht das gesamte Potential von low-cost Empfängern für Navigationsanwendungen aus, welche jedoch für Anwendungen mit hohen Genauigkeitsansprüchen erforderlich sind. Unter Verwendung aller GNSS-Rohdaten eines low-cost GNSS-Empfängers wie z.B. Informationen über mögliche Cycle Slip, Uhrparameter und Informationen über die Phasenmessung kann die Prozessierung deutlich verbessert und beschleunigt werden. Des Weiteren kann ein deutlicher Genauigkeitsgewinn im Vergleich mit der Verwendung herkömmlicher Übertragungsformate erzielt werden.

Anhand eigener Testszenarien wird das erstellte Navigation-Datenformat getestet und die erzielten Ergebnisse werden mit Positionslösungen, welche mit herkömmlichen Formaten wie RTCM erzielt wurden, gegenübergestellt. Die Vorteile erweiterter Rohdatenformate kann anhand dieser Testszenarien deutlich gezeigt werden.

Abstract

For the cost-effective estimation of position, speed and movement of an object, low-cost GNSS, INS, UWB or barometer data are mainly used in navigation applications. A specially designed innovative exchange data format for all installed sensors has to be developed and implemented in an own system architecture. Within this architecture, robust navigation applications with low-cost sensors are feasible.

Currently, the associated low-cost GNSS market mainly offers GPS or GLONASS compatibility for receivers. Responsible for raw-data broadcast within a mobile network are a specially designed innovative data-format, an own data-broadcast architecture and an enhanced positioning software.

Common standardized formats for raw-data broadcast are not able to extract the full extent of data from low-cost receivers. It is absolutely mandatory to use the full extent of receiver data for robust navigation applications.

Based on specially designed testing scenarios the newly created data format has to be tested and analyzed. Those achieved results have to be compared with other position results, like positions which are achieved from the standardized data format RTCM. All results have to be analyzed and compared with this conventional solution.

Akronyme

A/D	Analog to Digital
AGC	Automatic Gain Control
A-GPS	Assisted Global Positioning System
C/A	Coarse / Acquisition
C/N0	Carrier to Noise Ratio
CPU	Central Processing Unit
DD	Double Differences
DGPS	Differential GPS
DoF	Degree of Freedom
DLL	Delay Lock Loop
DOP	Dilution of Precision
DSP	Digital Signal Processor
ECEF	Earth-Centered, Earth-Fixed
ECSF	Earth-Centered, Space-Fixed
EGNOS	European Geostationary Navigation Overlay Service
EKF	Extended Kalman Filter
eRTK	enhanced Real Time Kinematic
ESA	European Space Agency
GLONASS	Globalnaja Nawigazionnaja Sputnikowaja Sistema
GNSS	Global Navigation Satellite System
GOBS	GNSS Observation
GPS	Global Positioning System
GPST	Global Positioning System Time
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IC	Integrated Circuit
ICD	Interface Control Document
IF	Intermediate Frequency
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LAMBDA	Least-Squares Ambiguity Decorrelation Adjustment
LLI	Loss of Lock Indicator
ILS	Integer Least Square
LSB	Least Significant Bit
MEMS	Micro-Electro-Mechanical System
MSB	Most Significant Bit
NMEA	National Marine Electronic Association
NTRIP	Networked Transport of TCM via Internet Protocol
PLL	Phase Lock Loop
PPP	Precise Point Positioning
PPS	Pulse Per Second

PRN	Pseudo Random Noise
PVT	Position Velocity Timing
RF	Radio Frequency
RINEX	Receiver Independent Exchange Format
RMSE	Root-Mean-Square Error
RPAS	Remotely Piloted Aircraft Systems
RTCM	Radio Technical Commission for Maritime Services
RTK	Real Time Kinematic
SBAS	Satellite Based Augmentation System
SD	Single Differences
SIS	Signal In Space
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
SPP	Single Point Positioning
TCP/IP	Transmission Control Protocol/Internet Protocol
TD	Triple Differences
UART	Universal Asynchronous Receiver Transmitter
UMTS	Universal Mobile Telecommunications System
UWB	Ultra Wide Band
WAAS	Wide Area Augmentation System

Inhaltsverzeichnis

1. Einleitung	1
2. Navigationssensoren	3
2.1 Inertiale Sensoren.....	3
2.1.1 MEMS Technologie.....	4
2.2 GNSS	6
2.2.1 Receiverdesign	6
2.2.2 high-cost Empfänger	9
2.2.3 low-cost GNSS-Empfänger	10
2.2.4 u-blox M8T	10
2.3 Navio Board	12
2.4 Systemarchitektur	14
3. GNSS Systeme und Formate.....	15
3.1 Grundlagen	15
3.2 Rohdatenformate	19
3.3 RINEX.....	23
3.4 RTCM	23
3.5 GOBS	26
3.6 Fazit.....	30
4. Übertragungskonzepte.....	31
4.1 Grundlagen.....	31
4.2 TCP/IP	31
4.3 NTRIP	32
4.4 Analyse der Übertragungsformate	35
5. Relative Positionierung	36
5.1 Grundlagen.....	36
5.2 Trägerphasenmessung.....	37
5.3 Single-differences	40
5.4 Double-differences.....	41
5.5 Triple-differences	42
5.6 Koordinatensysteme	43

5.6.1	ECEF	43
5.6.2	ECSF	45
5.6.3	Lokales Horizontsystem	46
5.7	Algorithmus.....	47
6.	RTKLIB	48
6.1	Grundlagen.....	48
6.2	Zugehörige Anwendungen.....	49
6.3	RTK-Definition	50
6.4	RTK-Algorithmus	51
6.4.1	EKF Formulierung.....	51
6.4.2	DD-Ansatz.....	52
6.4.3	Beobachtungsgewichtung	55
6.4.4	Integer Ambiguitätenlösung.....	56
7.	eRTK.....	58
7.1	Grundlagen.....	58
7.2	Systemarchitektur eRTK.....	58
7.3	eRTK-RTKNAVI.....	61
7.3.1	GOBS in eRTK.....	62
7.3.2	eRTK – EKF.....	64
7.3.3	eRTK – Gewichtungen	67
7.3.4	Cycle Slip Detektion	69
8.	Ergebnisse und Evaluierung	72
8.1	Messaufbau	72
8.2	Versuchsreihen.....	74
8.3	TestszENARIO – Höhe.....	75
8.3.1	Messergebnisse	76
8.3.2	Fazit.....	80
8.4	TestszENARIO – Lage1	81
8.4.1	Messergebnisse	82
8.4.2	Fazit.....	87
8.5	TestszENARIO – Lage2	87
8.5.1	Messergebnisse	89
8.4.2	Fazit.....	93

9 Zusammenfassung und Ausblick	94
Literaturverzeichnis	96
Abbildungsverzeichnis	99
Tabellenverzeichnis	102
Anhang A	103

Kapitel 1

Einleitung

Der stetig wachsende low-cost Navigationsmarkt bietet, zusammen mit neuen kostengünstigen Navigationsanwendungen aller Art, stets neue rohdatenfähige GNSS-Empfänger, MEMS-Sensorik und embedded Plattform Systeme für präzise Positionierungsmethoden an. Insbesondere die herkömmliche Verwendung der GNSS-Rohdaten, wie der Trägerphasenmessung im low-cost GNSS-Bereich, ist nicht mehr alleine ausschlaggebend für eine robuste Navigationsanwendung.

Durch diese Art an kostengünstigen Empfänger bedarf es somit einer speziellen Prozessierung für eine hochgenaue und stabile Positionslösung, eines speziellen innovativen Datenmanagements und einer speziellen Datenhandhabung, sowie einer eigens konzipierten Systemarchitektur für die Datenübertragung aller Navigationsdaten.

Die Grundidee besteht darin, alle Sensordaten über das Internet zugänglich zu machen und für eine weitere Prozessierung aufzubereiten. Somit können alle Daten mitgeloggt werden, bzw. auch in Echtzeit prozessiert werden.

Verantwortlich für die GNSS-Rohdatenübertragung, welche z.B. bei RTK zwingend notwendig ist, ist eine eigens in dieser Arbeit entwickelte, innovative Systemarchitektur, bestehend aus (i) einer mobilen-Datenübertragung, (ii) einer eigenen Server-Caster-Client Struktur, (iii) eines proprietären Navigation-Rohdatenformates und (iv) einer eigens definierten Schnittstelle zur Synchronisierung der Daten in Echtzeit.

Außerdem wird das zurzeit wachsende europäische Satellitensystem Galileo in das Format implementiert, wodurch es möglich ist, drei Satellitensysteme (GPS, GLONASS und Galileo) simultan mit low-cost GNSS-Hardware im Einfrequenzbereich zu empfangen und dadurch die Positionslösung deutlich zu stabilisieren und im low-cost Bereich innovativ zu gestalten.

Einsatz findet diese Art von low-cost GNSS Sensoren in Kombination mit low-cost Inertialsensorik in immer aktueller werdenden RPAS-Anwendungen, in welchen Navigation im kostengünstigen Bereich eine bedeutende Rolle einnimmt. Seit 2012 wird am Institut für Geodäsie, Arbeitsgruppe für Navigation, kontinuierlich an einer robusten und allgemeingültigen RTK-Software namens „eRTK“ geforscht, wobei low-cost Einfrequenz-Hardwarekomponenten zum Einsatz kommen. Durch die laufende Entwicklung, eng gekoppelt an den low-cost GNSS-Markt, können derzeit die Systeme GPS und GLONASS für die RTK-Processing herangezogen werden. Diese Matlab-basierende Software soll nun in eine erweiterte C/C++-basierende Software eingebunden werden und für die Positionierung herangezogen werden. Somit ist es möglich, echtzeitfähige Positionslösungen plattformunabhängig zu entwickeln und in sogenannte embedded platforms einzubinden.

Anhand eigener Testszenarien wird das erstellte Navigations-Datenformat sowie die eigene Systemarchitektur getestet, sowie neue Sensoren werden implementiert. Die erzielten Ergebnisse werden mit Positionslösungen, welche mit herkömmlichen Formaten wie RTCM erzielt wurden, gegenübergestellt und evaluiert. Abschließend werden die Vorteile aber auch Nachteile der Entwicklung aufgezeigt.

Kapitel 2

Navigationssensoren

Der Begriff der Navigationssensoren ist weit gedehnt und geht von Sensoren im handelsüblichen Smartphone bis hin zu hochpreisigen Navigationssensoren von Raketen und Flugzeugen. Klassisch versteht man darunter Sensoren, welche zur Bestimmung von Position, Geschwindigkeit und Raumlage eines Objektes verwendet werden. Navigationssensoren lassen sich laut Hofmann-Wellenhof et al. (2003) grob in folgende Bereichen gliedern:

- Dead Reckoning Sensoren,
- Sensoren für terrestrische Radionavigation,
- Sensoren für satellitenbasierte Navigation,
- Inertiale Sensoren
- und kamerabasierte Sensoren.

Auch eine kostenspezifische Einteilung lässt sich über solche Sensoren vornehmen. Hierzu unterscheidet man immer zwischen low-cost und high-cost Produkten, wobei low-cost Sensoren aller Art im Bereich von wenigen Euro bis zu 1 000 Euro liegen. Der high-cost Bereich beschreibt alles darüber und definiert sich nach oben hin offen.

Da im Zuge dieser Arbeit hauptsächlich inertielle low-cost Sensoren, sog. MEMS für die Datenaufbereitung und low-cost GNSS-Sensoren für die Prozessierung verwendet wurden, wird in nachfolgenden Unterkapitel nur auf diese Art von Sensoren näher eingegangen. Informationen über andere Sensoren können von Wendel (2011) und Moder (2011) entnommen werden.

2.1 Inertielle Sensoren

Inertielle Sensoren definieren sich durch die Bestimmung der Position und Geschwindigkeit mit Sensorik, welche den Newton'schen Bewegungsgesetzen

folgen. Diese Bewegung ist definiert durch Translation und Rotation, wodurch zwei verschiedene Typen von Sensoren, Akzelerometer und Gyroskope, welche sehr kurzzeitstabil sind (wenige Minuten), zum Einsatz kommen. Eine IMU besteht im grundlegenden aus drei Beschleunigungssensoren (Akzelerometern) und drei Gyroskopen, welche jeweils drei orthogonale Raumrichtungen messen (siehe Abbildung 2.1: Aufbau einer IMU, mit 3 Akzelerometern (Rechtecke) und 3 Gyroskopen (Zylinder)).

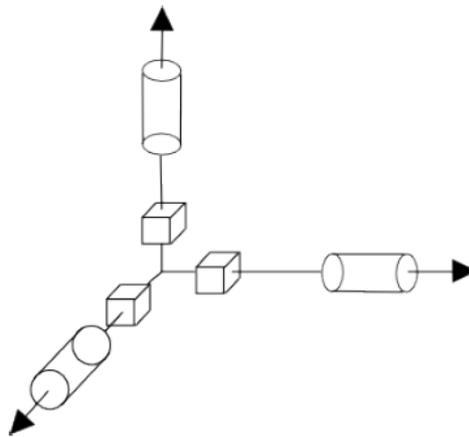


Abbildung 2.1: Aufbau einer IMU Hofmann-Wellenhof, et al. (2003)

Seitdem jedoch MEMS den Einzug in die inertielle Sensorik gefunden haben, dominieren diese den low-cost Massenmarkt im inertialen Bereich (z.B. Smartphones, Tablets, Smartwatches usw.) und werden nachfolgend kurz beschrieben. Für eine detaillierte Beschreibung des Prinzips der inertialen Sensorik wird auf Titterton et al. (2004) und Hofmann-Wellenhof et al., (2003) verwiesen.

2.1.1 MEMS Technologie

Die Bezeichnung „MEMS Technologie“ beschreibt keinen inertialen Sensor an sich, sondern eine elektronische Bauartweise von IC-Schaltungen. Durch diese Bauart lassen sich die inertialen Sensoren deutlich verkleinern, vergünstigen und sind sehr stromsparend. Diese Bauart weist jedoch eine massiv schlechtere Messgenauigkeit als hochpreisige mechanische inertielle Sensoren, auf. MEMS Sensoren bestehen grundlegend aus MEMS Akzelerometern, MEMS Gyroskopen und MEMS Magnetometern. Für eine detaillierte Beschreibung der MEMS Technologie wird auf Titterton et al., (2004) und Moder, (2011) verwiesen.

MEMS Akzelerometer

Anhand von Siliziumtechnologien können mechanische Elemente wie (i) Probemasse und (ii) Pendel auf kleinstem Raum verbaut werden. Nach Titterton et al., (2004) können MEMS Akzelerometer hauptsächlich durch die Art der Beschleunigungsmessung unterschiedlich konstruiert und eingeteilt werden. Entweder wird die Auslenkung einer Probemasse, oder die Änderung der Frequenz eines vibrierenden Bauteils, gemessen.

MEMS Gyroskop

Das Prinzip der Gyroskopmessung beruht auf der Bestimmung der Coriolisbeschleunigung. Die Bedingung des Auftretens der Corioliskraft ist jedoch, dass sich das Objekt selbst in Bewegung befindet. Somit lässt man es in eine bestimmte Richtung vibrieren. Titterton, et al. (2004) zeigt, dass eine Bauform mit einfachen Oszillatoren für die MEMS Technologie verwendet wird. Abbildung 2.2 zeigt den Einsatz von Gyroskopen, abgebildet nach einer Genauigkeitsskala. Zu sehen ist, dass MEMS Sensoren die deutlichen höchsten Fehler, wie (i) Bias und (ii) Skalierung aufweisen.

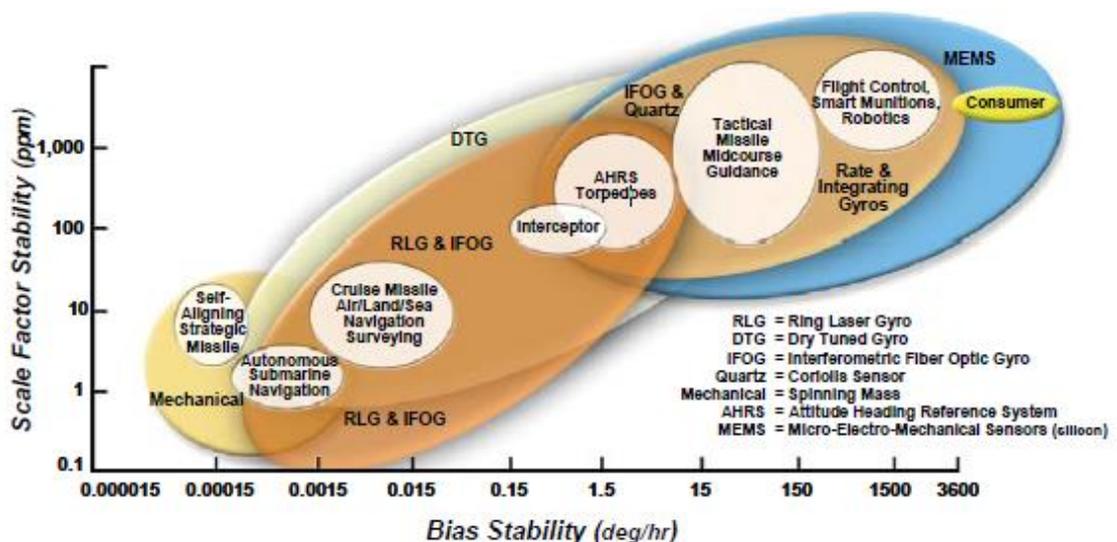


Abbildung 2.2: Überblick inertielle Sensoren mit Fehlergrößen RTO of NATO (2011)

2.2 GNSS

GNSS wird verwendet, um die dreidimensionale Position von Objekten bestimmen zu können, und weist eine gute Langzeitstabilität auf. Nähere Informationen über das System und das Funktionsprinzip können den Kapiteln 3 und 5 entnommen werden. Dieses Kapitel beschreibt den grundlegenden Aufbau eines GNSS-Empfängers und gibt einen Überblick über die low-cost Empfänger in diesem Bereich.

2.2.1 Receiverdesign

GNSS-Empfänger sind verantwortlich für die Verarbeitung der verschiedenen Signale des L-Bandes (SIS), siehe Abbildung 3.1, welche das Raumsegment, die Satelliten, Richtung Erde absenden. Hierbei überträgt jeder Satellit ein kontinuierliches Signal im 1,5 - 2 GHz-Frequenzbereich mit einem periodisch modulierten Codesignal. Der Empfänger zielt darauf ab, dieses Signal zu empfangen und mittels Korrelationstechnik mit einer Replika zu vergleichen (Akquisition). In weiterer Folge ist der Empfänger verantwortlich, die empfangenen Signale zu demodulieren und kontinuierlich zu tracken, sowie eine Positionslösung zu bestimmen. Der generelle Aufbau eines GNSS-Empfängers ist laut ESA (22.09.2016) in Abbildung 2.3 und Tabelle 2-1 dargestellt.

Tabelle 2-1: Receiverdesign

Komponente	Output	Beschreibung
Antenne	RF-Signal	Empfangen der Signale
Front End	Digitale binäre Werte	Konvertierung, Filterung, A/D
Baseband Processing	Messung	Signalprozessierung
Application processing	PVT	Positionsbestimmung

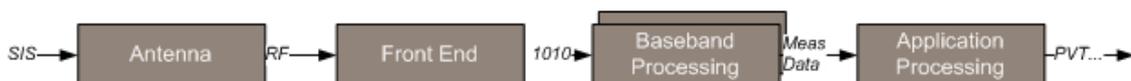


Abbildung 2.3: Receiverdesign laut ESA (22.09.2016)

Nachfolgend wird kurz auf die einzelnen GNSS-Empfängerkomponenten näher eingegangen, wobei das RF Front End und das Baseband Processing das

Grundgerüst eines jeden GNSS-Empfängers bilden. Je nach Anforderung der gewünschten Applikation, spielt der Application Processor jedoch nur eine untergeordnete Rolle, da nicht in jeder Anwendung eine direkte Positionslösung des Empfängers gewünscht ist. Wie in Abbildung 2.4 zu sehen ist, ist auch eine weitere Einteilung laut Berglez (2013) möglich, wobei nur die Bezeichnung der Komponenten geringfügig abweicht. Ein detaillierterer Überblick über den Aufbau eines GNSS-Empfängers kann auch Boore, et al. (2007) entnommen werden.

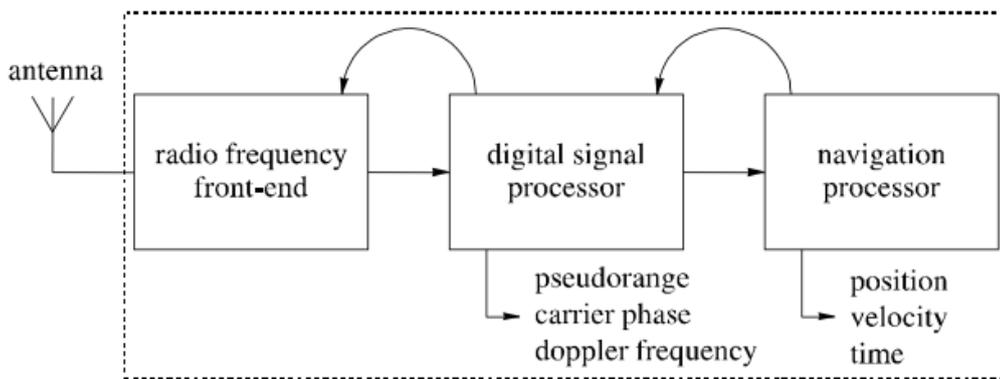


Abbildung 2.4: Receiverdesign Alternative

Der grundlegende Aufbau jedes hardwarebasierten GNSS-Empfängers ist der gleiche. Unterschieden wird hauptsächlich zwischen high-cost und low-cost Systemen, sowie zwischen rohdatenfähigen Empfängern und Empfängern, welche nur eine Positionslösung liefern. Eine weitere Einteilung der Empfänger kann auch noch nach sog. Software-based Receiver Boore et al., (2007) getroffen werden, auf welche in dieser Arbeit jedoch nicht eingegangen wird.

Antenne

Der Zweck einer Antenne liegt darin, das eingehende GNSS-Signal zu empfangen und dieses in Form von elektrischen Signalen an das RF Front End weiterzugeben. GNSS-Antennen existieren in verschiedensten Bauweisen, Preisklassen und Größen. Dies variiert je nach Anforderung an Frequenzen, Funktionalität und Genauigkeit der Messung an sich.

RF Front End

Das RF Front End besteht meist aus (i) low-noise amplifier (LNA), um das leistungsschwache empfangene Signal zu verstärken, sowie aus (ii) einigen Filtern, um Störeinflüsse des Signals, wie Multipath, zu minimieren. Das empfangene Signal wird danach mittels Vermischung mit einer RF Frequenz in eine abgewandelte Zwischenfrequenz, IF, transformiert. Das IF-Signal wird nun mittels Sampling Methoden in ein digitales Signal zu diskreten Zeitpunkten umgewandelt. Ein AGC sorgt für eine fehlerfreie Quantisierung des Signals zu diskreten Werten. Dieses digitale IF-Signal wird nun an den DSP für das Tracking und die Akquisition weitergegeben. Nähere Informationen zu einem RF-Front End können aus Boore et al., (2007) entnommen werden.

Digital Signal Processor

Der DSP dient zur Verarbeitung des digitalen IF-Signals, zur Akquisition, zum Tracking und zur Bestimmung der GNSS-Messungen wie Pseudorange, Trägerphase oder Doppler. Ein allgemeines Blockschaltbild eines DSP ist in Abbildung 2.5 dargestellt.

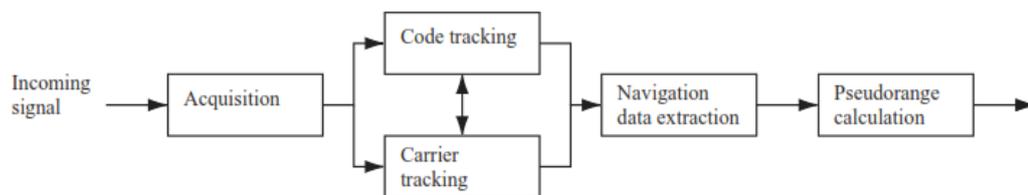


Abbildung 2.5. DSP Prinzip Boore et al., (2007)

Grundlegend vergleicht das Akquisition-Modul das digitale IF-Signal mit Replika Codes von jedem Satelliten. So können die Messungen einem bestimmten Satelliten zugeordnet werden. Wurde ein Satellit getrackt, wird dieser dekodiert, um sicherzustellen, dass der Satellit nicht verloren geht. Low-cost Empfänger waren hier lange Zeit Grenzen gesetzt, da nicht genügend Kanäle für das Tracken von mehreren Satelliten zur Verfügung standen. Mittlerweile stehen aber mehrere 100 Kanäle für das Tracking in low-cost Empfängern zur Verfügung, wie in Realini (2009) gezeigt wurde. High-cost Systeme, im Speziellen software-defined

Receiver, sind in der Lage, mehr als 1000 Kanäle gleichzeitig zu tracken. Der DSP stellt somit das Herzstück eines jeden GNSS-Empfängers dar und liefert als Endprodukt die Rohdatenmessungen zugeordnet zu jedem Satelliten. Nähere Informationen zu einem DSP können in Berglez (2013) und Realini (2009) nachgelesen werden.

Navigation Processor

Der Navigation Processor dient zur Analysierung der Rohdatenmessungen und zur Bestimmung der Position des Empfängers. Die meisten low-cost Empfänger haben hierbei jedoch ein Blackbox-Verhalten, wie in Realini (2009) gezeigt wurde. Da für eine weitere Verarbeitung in dieser Arbeit nur die Rohdaten zur Verwendung kommen, wird hierauf nicht weiter eingegangen.

2.2.2 high-cost Empfänger

Für hochgenaue Anwendungen werden üblicherweise geodätische Empfänger und Antennen verwendet, welche mehrere Trägerphasen auswerten können. Diese hochpreisigen Systeme, welche vor allem in (i) der Aeronautik, (ii) der herkömmlichen Vermessung, sowie (iii) für Monitoring von Objekten zum Einsatz kommen, sind deutlich robuster gegen Störungseinflüsse von außen und liefern eine stabilere Positionslösung als low-cost Komponenten. Preislich liegen diese Art von Systemen im Bereich von mehreren 1000,- Euro. Bekannteste Hersteller im high-cost Bereich sind u.a. Leica, Topcon, Javad, Ashtech, NovAtel, Trimble.



Abbildung 2.6: High-cost Empfänger (Topcon Hyper II und Javad Sigma TRE)

2.2.3 low-cost GNSS-Empfänger

Der GNSS-Markt erlebt vor allem im Bereich der low-cost Sensorik ein großes Wachstum. Die Verwendung von Multisystemen wie (i) GPS, (ii) GLONASS, (iii) Beidou und (iv) Galileo ist auch bei low-cost Sensorik von großer Nachfrage betroffen und erlebt einen gewaltigen Entwicklungsschub. Somit gibt es immer mehr Hersteller mit günstigen Multisystemempfängern auf den Markt. Der Aspekt der Rohdatenfähigkeit von low-cost Empfängern soll hier deutlich erwähnt werden, da für eine weitere Prozessierung die Rohdaten der GNSS-Messung ausschlaggebend sind, der low-cost Bereich jedoch vor allem auf geschlossene GNSS-Empfänger ohne Rohdatenausgabe baut. Da im Zuge dieser Arbeit vor allem der Aspekt der low-cost Sensorik mit Rohdatenausgabe im Vordergrund steht, wird nachfolgend in Abschnitt 2.2.4 der verwendete Empfänger vorgestellt.

2.2.4 u-blox M8T

Wie bereits zuvor erwähnt, werden im Laufe der Arbeit nur low-cost Hardware-Komponenten verwendet. Dies hat den entscheidenden Nachteil zu high-cost Systemen, dass nur eine Trägerphase zur Auswertung zur Verfügung steht. Ebenfalls sind bei einer kostengünstigen GNSS-Antenne deutliche Qualitätsverluste gegenüber einer hochpreisigen geodätischen Antenne zu verzeichnen. Der im Zuge dieser Arbeit verwendete low-cost GNSS-Empfänger u-blox M8T der Herstellerfirma u-blox AG¹ ist in Abbildung 2.7 dargestellt.



Abbildung 2.7: u-blox Evaluation Kit und u-blox Chip

¹ U-blox AG, Halbleiterbausteinhersteller, Thalwil Schweiz

Nähere Spezifikationen zum Empfänger können der Tabelle 2-2 entnommen werden.

Tabelle 2-2: u-blox M8T

Empfänger	u-blox NEO M8T
Updaterate	max. 2 Hz
Stromversorgung	5.0 VDC
Anzahl Kanäle	72 (1 Frequenz)
Firmware	EXT CORE 3.01
Modul	NEO-M8T PrecisionTiming Rohdatenfähig
Genauigkeit der Rohdaten	Halbe Wellenlänge L1 Träger (9,5cm)
Kosten [EUR] Chip	99,- (Stand August 2016)
Evaluation Kit	280,- (Stand August 2016)
GNSS	GPS, GLONASS, Galileo, Beidou siehe Kombinationen

Wie in Tabelle 2-2: u-blox M8T gezeigt wird ist ein Empfangen aller zurzeit aktiven GNSS-Systeme möglich. Aufgrund der limitierten Anzahl an Frequenzbändern und Kanälen, wie auch in Abschnitt 2.2.4 gezeigt wird, können nicht alle Systeme simultan betrieben werden. Die möglichen Kombinationen, welche sich aus Tests im Laufe dieser Arbeit herausgestellt haben, sind in Abbildung 2.8 dargestellt, wobei jedes System auch einzeln messbar ist.

	GPS	GALILEO	GLONASS	BeiDou
Kombinationen	✓ □	✓ □	-	-
	✓	✓	✓	-
	✓	✓	-	✓
	✓	-	✓	-
	✓	-	-	✓
	-	✓	✓	-
	-	✓	-	✓
	-	-	✓	✓
	-	-	-	-

Abbildung 2.8: u-blox M8T GNSS-Kombinationen

2.3 Navio Board

Navigationen Anwendungen fordern Applikationen und Plattformen, welche möglichst viele Sensoren miteinander verknüpfen. Vor allem im low-cost Bereich sind spezielle embedded platforms mit mehreren Navigationssensoren rar gesät. Insbesondere Navigationsanwendungen für RPAS fordern leichte und kostengünstige Multisensorsysteme. Somit wurde im Zuge von Forschungsprojekten ein Multisensorsystem der Firma emlid² für Forschungszwecke verwendet. Die Firma spezialisiert sich auf Multisensorsysteme für RPAS, basierend auf der linuxbasierten Einplatinenplattform Raspberry PI 2. Das Multisensorsystem Navio 2 wurde in Kombination mit Raspberry PI Modell 2 im Zuge dieser Arbeit verwendet. Ein Überblick der Spezifikationen wird in Tabelle 2-3: Navio Spezifikationen geliefert.

Tabelle 2-3: Navio Spezifikationen

Platform	Navio 2
MPU9250	9DOF MEMS IMU, Chip 1
LSM9DS1	9DOF MEMS IMU, Chip 2
MS5611	Barometer
u-blox M8N	GNSS-Empfänger Achtung: keine Rohdaten!
14 PWM	Servo Outputs
Power Module	Stromausgabe
UART, I2C, SPI	Schnittstellen
Sonstiges	Kann der Homepage der Firma Emlid entnommen werden.

Das Navio 2 stellt jedoch keinen rohdatenfähigen GNSS-Empfänger bereit, sondern den Empfänger u-blox M8N, welcher nur Positionslösungen bereitstellt. Abbildung 2.9: Änderung des GNSS-Empfängers zeigt das Blockschaltbild des

² Emlid Limited, Hong Kong

M8N und des M8T Empfängers, welche identische Pinouts besitzen und somit ausgetauscht werden können.

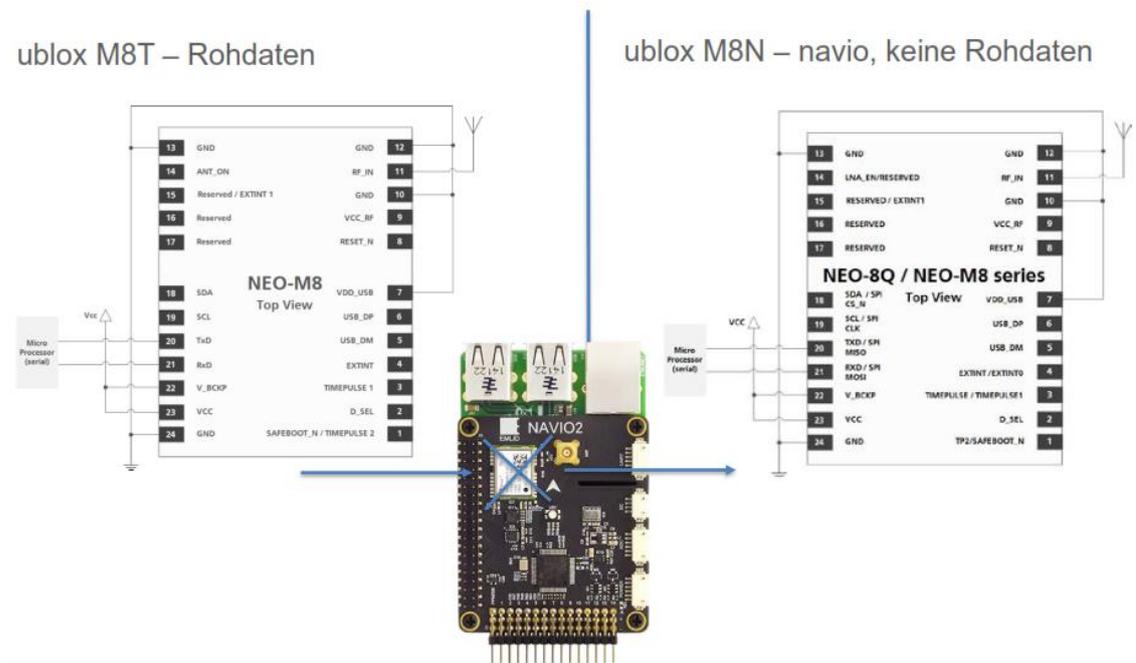


Abbildung 2.9: Änderung des GNSS-Empfängers

Durch die Kompatibilität der Pinouts der beiden GNSS-Chips ist ein Austausch des Empfängers möglich. Hierzu wurde mittels Heißluftflöten der u-blox M8N entfernt und durch den rohdatenfähigen u-blox M8T ersetzt. Ein genereller Aufbau des Navio 2 Boards ist der Abbildung 2.10 zu entnehmen.

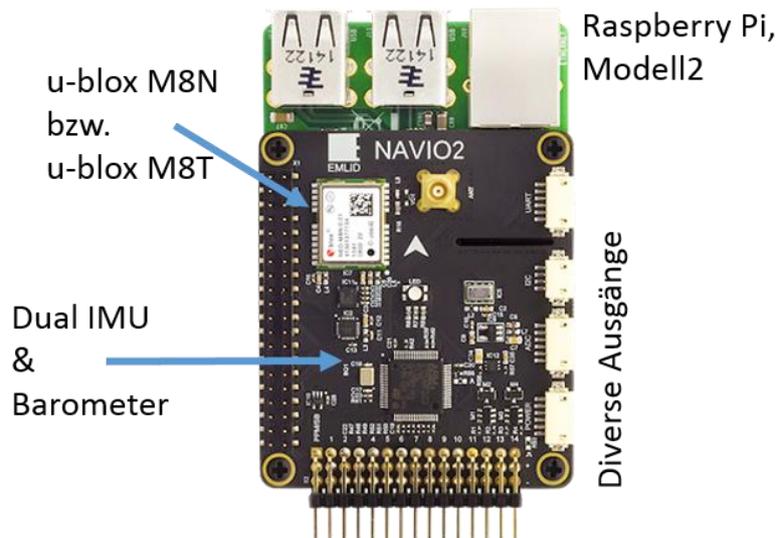


Abbildung 2.10: Navio 2: Aufbau

Abbildung 2.10 zeigt das Navio 2 Board aufgesteckt auf einen Raspberry PI 2, welcher für den Betrieb des Boards notwendig ist. Ausgestattet ist das adaptierte Navio 2 Board nun mit einer DUAL 9DOF IMU, einem Barometer und dem ausgetauschten rohdatenfähigen u-blox M8T GNSS-Chip. Dieser kann einzeln erworben werden und besitzt laut Abbildung 2.9 die selbe Pinbelegung, wodurch ein Ansprechen des Chips mit dem Platinenlayout des Navio 2 Boards möglich ist.

2.4 Systemarchitektur

Die in den Kapiteln zuvor erwähnten Sensoren beschreiben nun die Hardware-Systemarchitektur dieser Arbeit. Ziel ist es, die inertialen Daten ins Internet zu streamen und die GNSS-Daten zu übertragen, sowie im relativen RTK Modus auszuwerten, siehe Kapitel 5 und 6.

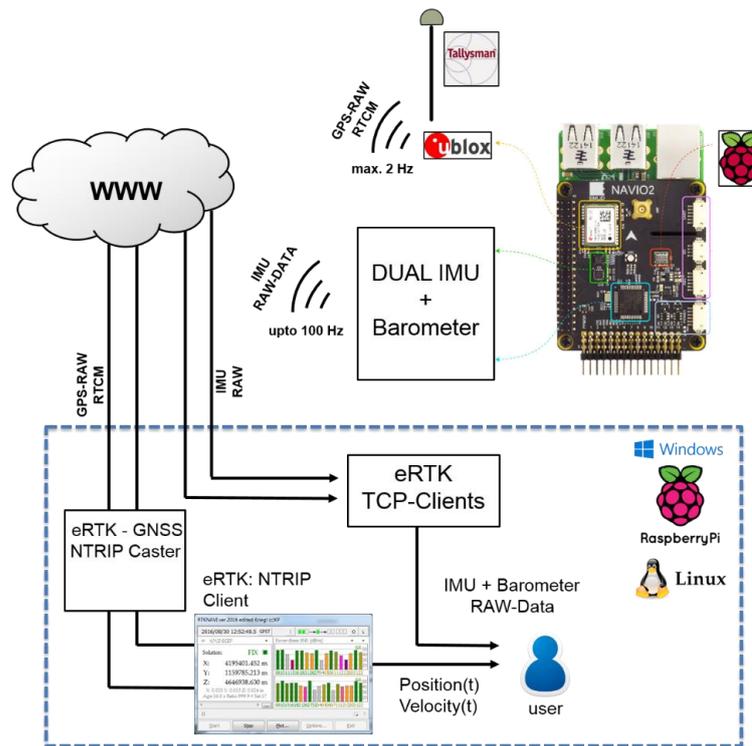


Abbildung 2.11: Systemarchitektur Navio

Die Beschreibung der GNSS-Datenübertragung und Prozessierung kann den Abschnitten 3.5, 4 und 5 entnommen werden.

Kapitel 3

GNSS Systeme und Formate

3.1 Grundlagen

GNSS dient zur Positionsbestimmung in Bezug auf einen Empfänger mit Hilfe von Satellitensignalen. Diese Art der globalen Satellitensysteme bietet theoretisch jederzeit und überall auf der Erdoberfläche eine ausreichende Anzahl von mindestens vier Sichtverbindungen zu Satelliten, welche für eine Positionsbestimmung notwendig ist.

Zurzeit werden hauptsächlich (i) GPS der Vereinigten Staaten und (ii) GLONASS von Russland für die globale Abdeckung verwendet. Allerdings befinden sich weitere Systeme im Aufbau und werden bereits verwendet, sind jedoch für die globale Abdeckung mit noch zu wenig Satelliten im Raumsegment ausgestattet. Diese Systeme sind (i) das europäische System Galileo sowie (ii) das chinesische System BeiDou/Compass. Weitere Systeme zur Integritätssteigerung wie SBAS (EGNOS, WAAS) stehen auch zur Verfügung. Eine prinzipielle Übersicht der vorhanden GNSS Systeme kann Abbildung 3.1 entnommen werden.

Parameter	GPS	GLONASS	Galileo	Compass (Meo)
First launch	22-Feb-78	12-Oct-82	28-Dec-05	13-Apr-07
FOC	17-Jul-95	18-Jan-96	2012	2013
Services	-military -civil	-military -civil	-commercial -open	- authorized - open/commercial
Number of SV	31	24	27	27
Orbital planes	6	3	3	3
Inclination	55°	64.8°	56°	55°
Semi-major Axis [Km]	26560	25508	29601	27840
Period	11h58m	11h15m	14h05m	12h50m
Coordinate frame	WGS-84	PZ-90	GTRF	Beijing 1954
Time system	GPST	UTC (SU)	GST	China UTC
Coding	CDMA	FDMA	CDMA	CDMA
Frequencies [MHz]	L1:1575.42 L2:1227.60 L5:1176.45	G1:1602 G2:1246 G3:TBD	E1:1575.42 E5a:1176.45 E5b:1207.14 E6:1278.75	B1-2:1589.74 (E1) B-1:1561.1 (E2') B2:1207.14 (E5b) B3:1268.52 (E6)

Abbildung 3.1: GNSS Systeme nach Hofmann-Wellenhof et al.,(2001)

Durch in Hofmann-Wellenhof et al., (2008) gezeigte Fehlereinflüsse, wie (i) atmosphärische Einflüsse, (ii) Uhrenfehler und insbesondere (iii) Multipath, erreicht man mit einer Einzelpunktbestimmung eine Genauigkeit von mehreren Metern. Mit differentiellen Ansätzen wie DGPS und der relativen Positionierung samt RTK kann eine deutliche Genauigkeitssteigerung in den cm-Bereich erreicht werden, siehe Kapitel 5, 6 und Hofmann-Wellenhof et al., (2008).

Das Prinzip der Messung basiert auf einer Laufzeitmessung, um die gemessene Strecke zum Satelliten zu berechnen, sowie auf einer Phasen- und Dopplermessung. Betrachtet man die Darstellung in Abbildung 3.2 eines Empfängers auf der Erde und eines Satelliten, so ergibt sich folgender funktioneller Zusammenhang.

$$\rho = |\boldsymbol{\rho}^s - \boldsymbol{\rho}_r|, \quad 3-1$$

wobei der Vektor $\boldsymbol{\rho}_r$ die Unbekannte darstellt. Das Skalar bzw. der Vektor ρ und $\boldsymbol{\rho}^s$ ergeben sich aus der Laufzeitmessung des Signals bzw. der bekannten Satellitenposition im Orbit.

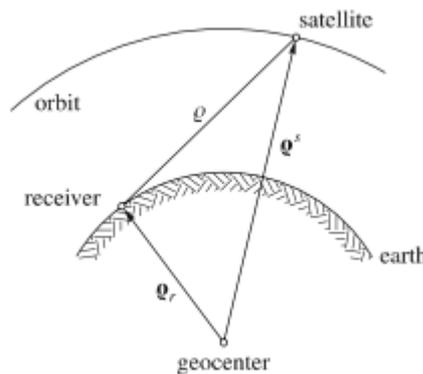


Abbildung 3.2: Prinzip der Satellitenpositionierung von Hofmann-Wellenhof et al.,(2001)

Der Systemaufbau eines GNSS Systems gliedert sich laut Global Positioning System Directorate (2013) in drei Bereiche:

- **Raumsegment**

Wie in Hofmann-Wellenhof et al., (2001) gezeigt wird, müssen mindestens 24 Satelliten für eine globale Abdeckung die Erde umkreisen. Die ausgesendeten elektromagnetischen Wellen bestehen aus einer Trägerwelle im L-Band,

auf welcher ein PRN-Code sowie Navigationsnachrichten moduliert sind. Der PRN Code dient zur Bestimmung der Satellitennummer und die Navigationsnachrichten geben Auskunft über Bahn- und Uhrenparameter des Satelliten, wobei jedes System leicht andere Spezifikationen der Signalaufbereitung verwendet. Nähere Information können dem Global Positioning System Directorate (2013) entnommen werden. Ausgesendet werden diese Signale mit einer Leistung von rund 22 W, siehe Zogg (2002).

- **Kontrollsegment**

Das Kontrollsegment, auch als Bodensegment bekannt, dient zur Steuerung und Überwachung des gesamten Systems. Dazu wurden weltweit Bodenstationen verteilt, welche für die Orbitbestimmung der Satelliten verantwortlich sind. Die Orbitparameter werden dann mittels Up-Link Stationen an die Satelliten gesendet und in die Navigationsnachricht implementiert.

- **Usersegment**

Das Benutzer- oder Usersegment teilt sich in verschiedene Kategorien ein. Diese Kategorien beziehen sich auf verwendete Empfänger, gewünschten Service sowie gewünschte Genauigkeit. Empfangen werden die Signale mit einer Leistung von nur rund -160 dBm, was einer Leistung von 10^{-16} W entspricht, siehe Kaplan (2006).

Messprinzip

Wie in Formel 3-1 gezeigt, wird keine direkte geometrische Strecke gemessen, sondern eine Pseudostrecke R , welche um den Uhrenfehler verfälscht ist. Unterschieden wird hier nach Laufzeit- oder Phasenmessung. Ein einfaches Modell zur Beschreibung der Pseudostrecke lautet,

$$R_r^s(t) = |\boldsymbol{\rho}^s(t) - \boldsymbol{\rho}_r(t)| + \Delta\rho(t) = \rho_r^s(t) + c\delta_r^s(t), \quad 3-2$$

wobei $\Delta\rho$ auftretende Fehler, wie Uhrenfehler und atmosphärische Einflüsse entlang der Messung beschreibt. Beachtet man nur den Uhrenfehler, welcher bei einer Distanz von rund 20.000 km bei Lichtgeschwindigkeit (Parameter c) den

größten Einfluss hat, so ergibt sich das oben angeführte vereinfachte Modell. δ beschreibt hier den kombinierten Uhrenfehler, welcher sich wieder in Empfänger- und Satellitenuhrenfehler aufteilt. Der Satellitenuhrenfehler ist aber aufgrund der Navigationsnachricht bekannt. Daraus ergeben sich laut Formel 3-2 vier Unbekannte: der Empfängeruhrenfehler, sowie die drei Positionskomponenten, welche in p_r enthalten sind. Somit muss zu jedem Zeitpunkt eine Sichtverbindung zu vier Satelliten vorhanden sein, um sich positionieren zu können. Diese Art der Positionierung, Einzelpunktlösung (SPP), mit Codebeobachtungen ist die einfachste, jedoch auch die ungenaueste Methode mit einer Abweichung von mehreren Metern zur wahren Position. Die in dieser Arbeit verwendete Technik der relativen Positionierung in Echtzeit wird in Kapitel 5 und 6 beschrieben. Nähere Informationen zum Messprinzip der satellitengestützten Positionierung kann Hofmann-Wellenhof et al., (2008) und Kaplan (2006) entnommen werden.

Datenformate

Da die Übertragung und Aufbereitung der GNSS-Daten ein großer Aspekt dieser Arbeit ist, wird in den nachfolgenden Kapiteln auf die unterschiedlichen GNSS-Datenformate eingegangen. Grundlegend unterscheidet man nach Rohdaten, welche empfängerspezifisch sind, und standardisierten Daten unterteilt in GNSS-Formaten. Diese wiederum unterteilen sich in Formaten für Echtzeitanwendungen und Formaten für das Postprocessing.

Das Ziel ist es, ressourcenschonend Rohdaten, Korrekturdaten oder Positionsdaten an jegliche Benutzer zu übertragen. Für den Bereich der GNSS-Datenübertragung für Benutzer ist meist eine Einweg-Datenübertragung ausreichend. Die grundlegende Aufteilung der Daten lautet nach Hofmann-Wellenhof et al., (2008)

- **Rohdaten**

Empfängerspezifische, binäre Rohdaten werden hauptsächlich zur Datenaufzeichnung verwendet und beinhalten alle Daten, welche der Empfänger ausgibt.

- **RTCM**

Dies ist ein standardisiertes Format für die Echtzeitübertragung von Roh- und Korrekturdaten an Benutzer bzw. anderer Empfänger.

- **RINEX**

Dies ist ein Datenformat zur Auswertung von GNSS-Signalen im Postprocessing. Es beinhaltet definierte Rohdaten des Empfängers.

- **NMEA**

Dies ist ein standardisiertes ASCII Datenformat, um Positionslösungen zu speichern und zu übertragen.

3.2 Rohdatenformate

Mit der Rohdatenausgabe in offengelegten, binären Formaten ist es möglich, Zugriff auf alle Empfängerdaten zu bekommen. Im Allgemeinen stellen Hersteller ein eigens definiertes Rohdatenformat offengelegt zur Verfügung. Schließlich ist es möglich, Daten zu dekodieren und somit Zugriff auf alle GNSS-Rohdaten der Empfänger zu erhalten, um eine Prozessierung in einer externen Software durchzuführen. Zum einen kann eine Konvertierung zum standardisierten RINEX Format durchgeführt werden, zum anderen ist es möglich, alle zugänglichen Rohdaten in eine Software zu implementieren und dadurch die Positionslösung zu verbessern. Der in dieser Arbeit verwendete u-blox M8T Empfänger greift auf das offengelegte binäre Format *ubx* u-blox AG (2014) zurück, welches von allen u-blox Empfänger verwendet wird. Das für den benutzten Empfänger u-blox M8T verwendete Format wird nachfolgend beschrieben.

UBX-Protokoll

Das proprietäre Protokoll erlaubt die Ausgabe aller Empfängerdaten im binären Format und kann nach der Protokollspezifikation dekodiert werden. Das Protokoll verwendet eine 8 Bit Kodierung und wird mittels Checksumme auf Vollständigkeit überprüft. Die Class- und Message-ID wird für die Unterscheidung und Einteilung der Nachrichtentypen (Messages) eingesetzt. Die grundlegende Paketstruktur nach u-blox AG (2014) ist in Abbildung 3.3 dargestellt.

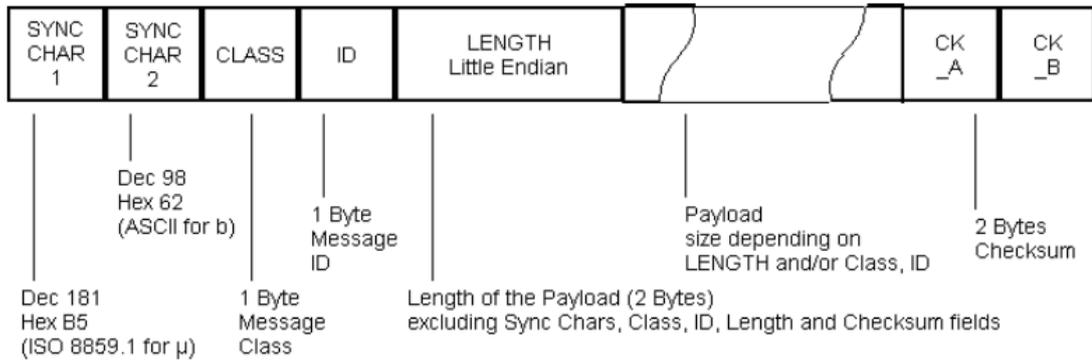


Abbildung 3.3: UBX-Protokollstruktur nach u-blox AG (2014)

Die Felder Class und ID dienen zur Definition des Message Type, und werden nachfolgend erklärt. Die eigentlichen Daten werden dann im Payload Feld eingefügt, diese variieren je nach Message. Die Class ID, welche im UBX-Protokoll definiert sind, können der nachfolgenden Tabelle entnommen werden. Jede Class wird dann wiederum in Messages aufgeteilt. Die wichtigsten u-blox Classes sind NAV und RXM, in welchen Satelliteninformationen und Positionslösungen bzw. die Rohdaten implementiert sind.

Tabelle 3-1: UBX-Class IDs

Name	Class	Beschreibung
NAV	0x01	Positionslösungen PVT, Heading, DOP, Satelliteninformationen
RXM	0x02	Rohdaten, Navigationsnachrichten Ephemeriden
INF	0x04	Error-Informationen, Sonstiges
ACK	0x05	Bestätigung der Konfigurationen
CFG	0x06	Konfigurationsnachrichten
UPD	0x09	Firmwarenachrichten, Firmware-Updates
MON	0x0A	CPU-Load, Empfängerstatus
AID	0x0B	Aiding Messages, Almanach, A-GPS
TIM	0x0D	Zeitinformationen, PPS Output
MGA	0x13	Mult-GNSS Daten (keine Rohdaten)
LOG	0x21	Logging Messages, sonstige Informationen

Die in dieser Arbeit verwendeten Messages für eine erweiterte Rohdatenprozessierung werden in nachfolgender Tabelle 3-2 aufgelistet und beschrieben. Für eine Übersicht aller vorhandenen Messages siehe u-blox AG (2014).

Tabelle 3-2: UBX-Messages

Message	Class ID	Beschreibung
UBX-NAV		
NAV-Clock	0x01 0x22	Uhrenlösungen, Bias Drift, GPST Time, Leapseconds
NAV-Sol	0x01 0x06	Positionslösungen, Genauigkeiten, Geschwindigkeiten
NAV-Sat	0x01 0x35	Satelliteninformationen, Azimuth, Elevation, DOP
UBX-RXM		
RXM-RAWX	0x02 0x15	Multi-GNSS Rohdatenmessungen
RXM-SFRBX	0x02 0x13	Raw Subframe Daten (Ephemeriden), siehe ICD der einzelnen System z.B. (Global Positioning System Directorate, 2013)

Der Aufbau der einzelnen Messages ist u-blox AG (2014) zu entnehmen und die für die Rohdatenmessung verwendete Message RXM-RAWX wird in nachfolgender Tabelle 3-3 erklärt.

Tabelle 3-3: UBX RXM-RAWX nach u-blox AG (2014)

Message	RXM-RAWX					
Description	Multi-GNSS Rohdatenmessungen					
Firmware	Supported on: • u-blox M8 firmware version 2.30 & 3.01 (für rohdatenfähige Empfänger)					
Type	Periodic/Polled (z.B. bei 2 Hz alle 0.5 Sekunden)					
Comment	Beinhaltet alle Multi-GNSS Rohdatenmessungen. Der Aufbau kann der unten angeführten Struktur entnommen werden.					
Message Structure	<i>Header</i>	<i>Class</i>	<i>ID</i>	<i>Length (Bytes)</i>	<i>Payload</i>	
	0xB5 0x62	0x02	0x15	16 + 32*numMeas	see below	
Payload Contents:						
Byte Offset	Number	Scaling	Name	Einheit	Beschreibung	
0	R8	-	rcvTow ⁽¹⁾	s	Gemessene GPS-Sekunden in aktueller GPS Woche. Für die Struktur der GPS Zeit siehe (Global Positioning System Directorate, 2013)	
8	U2	-	Week ⁽¹⁾	weeks	Gemessene GPS-Woche	

10	I1	-	leapS	s	GPS Leap-Seconds (GPS-UTC)
11	U1	-	numMeas	-	Anzahl an Messungen
12	X1	-	recStat	-	Trackingstatus des Empfängers
13	U1[3	-	reserved1	-	Reserved
Start der wiederholten Blöcke (Anzahl: numMeas viele)					
16 + 32*N	R8	-	prMes	m	Pseudorangemessung. GLONASS Interfrequency Channel wird intern angebracht.
24 + 32*N	R8	-	cpMes	cycles	Phasenmessung in cycles. Messgenauigkeit auf ½ Cycle
32 + 32*N	R4	-	doMes	Hz	Dopplermessungen. Positiv für steigende und negative für absteigende Satelliten
36 + 32*N	U1	-	gnssId	-	GNSS identifier
37 + 32*N	U1	-	svId	-	Satellite identifier
38 + 32*N	U1	-	reserved2	-	Reserved
39 + 32*N	U1	-	freqId	-	GLONASS Frequenzchannel (range from 0 to 13)
40 + 32*N	U2	-	locktime	ms	Zeit einer gelockten Phasensmessung (maximum 64500ms)
42 + 32*N	U1	-	cno	dBHz	Signal-Rausch-Verhältnis
43 + 32*N	X1	0.01*2 ⁿ	prStdev	m	Geschätzte Standardabweichung der Pseudorangemessung
44 + 32*N	X1	0.004	cpStdev	cycles	Geschätzte Standardabweichung der Phasenmessung in Cycles. Ohne dem Scaling (Bitwerte 1 - 15)
45 + 32*N	X1	0.002*2 ⁿ	doStdev	Hz	Geschätzte Standardabweichung der Dopplermessung
46 + 32*N	X1	-	trkStat	-	Tracking status bitfield. Informationen über Cycle Slip
47 + 32*N	U1	-	reserved3	-	Reserved
Ende des wiederholten Blockes					

⁽¹⁾ Zeitparameter beziehen sich immer auf GPST.

3.3 RINEX

Das RINEX Format wurde am Institut für Astronomie an der Universität Bern für den einheitlichen Austausch von GPS Daten entwickelt. Das Format wurde laut Hofmann-Wellenhof et al., (2008) im Jahr 1989 das erste Mal vorgestellt und seitdem stets weiterentwickelt, auch den Umständen verschuldet, das im Laufe der Jahre neue Satellitennavigationssysteme und neue Frequenzbänder entwickelt wurden. Die grundlegende Idee ist, Daten, welche für eine Prozessierung benötigt werden, standardisiert im ASCII Format darzustellen. Im Allgemeinen handelt es sich dabei um (i) die Codemessung, (ii) Phasenmessung, (iii) Dopplermessung sowie (iv) dem Signal-Rausch-Verhältnis SNR, alle zugeordnet zu Satellit und Zeitpunkt. Aufgeteilt werden RINEX Files in ein Observation File mit den zuvor erwähnten Rohdaten und einem Navigation File, welche die Bahnparameter für alle sichtbaren Satelliten beinhalten.

Tabelle 3-4: RINEX Files

File Type	Containing Information
Observation Data File	GNSS Measurements
GPS Navigation Message File	Ephemeris (Orbit information)
Meteorological Data File	Pressure, Temperature etc.

3.4 RTCM

Für den Echtzeit-Austausch von Korrektur- bzw. Rohdaten zwischen Empfängern oder Navigationsanwendungen, wie sie bei DGPS oder RTK notwendig sind, wurde 1985 von der US Radio Technical Commission for Maritime Services, siehe RTCM SPECIAL COMMITTEE NO. 104 (2006), ein standardisiertes Format, kurz RTCM, entwickelt. Im Laufe der Jahre entstanden mehrere Versionen, wobei das aktuellste die Version RTCM 3.1 ist. Im Zuge dieser Arbeit wurde das Format RTCM 3.1 für die Übertragung von RTK-Nachrichten für GPS und GLONASS verwendet, RTCM SPECIAL COMMITTEE NO. 104 (2006). Das Referenzdokument beschreibt die Nachrichten und Techniken zur Übertragung von RTK-Rohdaten und Navigationsnachrichten.

Je nach Version von RTCM werden unterschiedliche Funktionen der Übertragung unterstützt.

Tabelle 3-5: RTCM Versions

Version	Beschreibung
RTCM 2.0	Code Korrekturen for DGPS
RTCM 2.1	Code + Phase Korrekturen für RTK
RTCM 2.2	... + GLONASS Kompatibilität
RTCM 2.3	... + Antennen Parameter
RTCM 3.0	... + Network RTK, GNSS and RTK raw data

RTCM 3.x

Das RTCM Format wird in sogenannte Message Types aufgeteilt, von denen zurzeit 63 verschiedene Botschaften verwendet werden. Jeder Message Type beginnt mit zwei festgeschriebenen Datenwörtern (30 Bit), welche sich im sogenannten Header befinden. Dieser dient zur Überprüfung der Übertragung und zur Bestimmung des Message Types, siehe RTCM SPECIAL COMMITTEE NO. 104 (2006). Die Parität dient zur Überprüfung auf Korrektheit (Länger der Message) und das MSB bzw. LSB dient zur Definition der Bitanordnung pro Message.

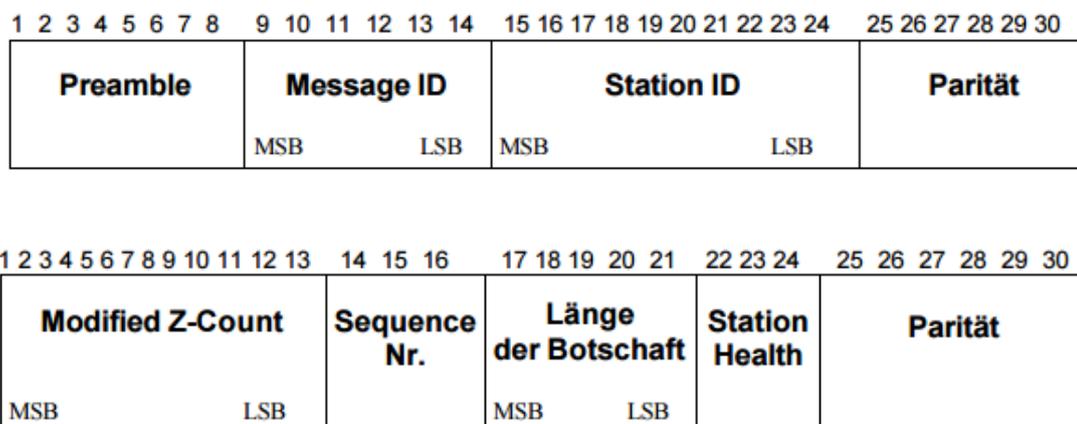


Abbildung 3.4: RTCM Header

Die Message ID beschreibt die Art der Daten, welche übertragen werden. Für den RTK-Modus stehen laut RTCM SPECIAL COMMITTEE NO. 104 (2006) folgende Messages zur Verfügung.

Group Name	Sub-Group Name	Message Type
Observations	GPS L1	1001
		1002
	GPS L1 / L2	1003
		1004
	GLONASS L1	1009
		1010
GLONASS L1 / L2	1011	
	1012	
Station Coordinates		1005
		1006
Antenna Description		1007
		1008
Network RTK Corrections	Network Auxiliary Station Data Message	1014
	Ionospheric Correction Differences	1015
	Geometric Correction Differences	1016
	Combined Geometric and Ionospheric Correction Differences	1017
Auxiliary Operation Information	System Parameters	1013
	Satellite Ephemeris Data	1019
		1020
	Unicode Text String	1029
Proprietary Information		Currently assigned message numbers 4088 – 4095

Abbildung 3.5: RTCM RTK-Messages nach RTCM SPECIAL COMMITTEE NO. 104 (2006)

Eine detaillierte Auflistung der RTK Parameter ist der nachfolgenden Tabelle 3-6 zu entnehmen.

Tabelle 3-6: RTCM 3.0, RTK Messages nach RTCM SPECIAL COMMITTEE NO. 104 (2006)

Message Type	Inhalt
1004	GPS Code, Phasenmessung, LLI und SNR
1006	ECEF Koordinaten des Referenzpunktes mit Antennenhöhe
1008	Antennenparameter
1012	GLONASS Code, Phasenmessung, LLI und SNR

Die RTCM Version ab RTCM 3.0 implementiert hauptsächlich RTK Parameter wie Rohdaten oder Netzinformationen. Die Rohdaten, sprich Phasenmessung, Codemessung, SNR oder LLI werden übertragen. Sonstige Messung, wie unter

anderem die Dopplermessung, sind jedoch nicht standardisiert in das Format implementiert.

3.5 GOBS

Das im Zuge von Projektarbeiten am Institut für Geodäsie entwickelte GNSS Observation Format (GOBS), ist ein eigens konzipiertes proprietäres Rohdatenformat, welches speziell für den u-blox M8T nach Kriegl et al., (2015) und Koch et al., (2015) konzipiert wurde.

Ziel dieses Formates ist es, alle Rohdaten des u-blox M8T laut Abschnitt 3.2 in ein vereinheitlichtes Format zu implementieren. Das im Zuge einer Matlab-basierenden RTK-Software entstandene Format wird hier entsprechend verbessert und erweitert sowie in eine verbesserte RTK-Software nach RTKLIB eingebunden.

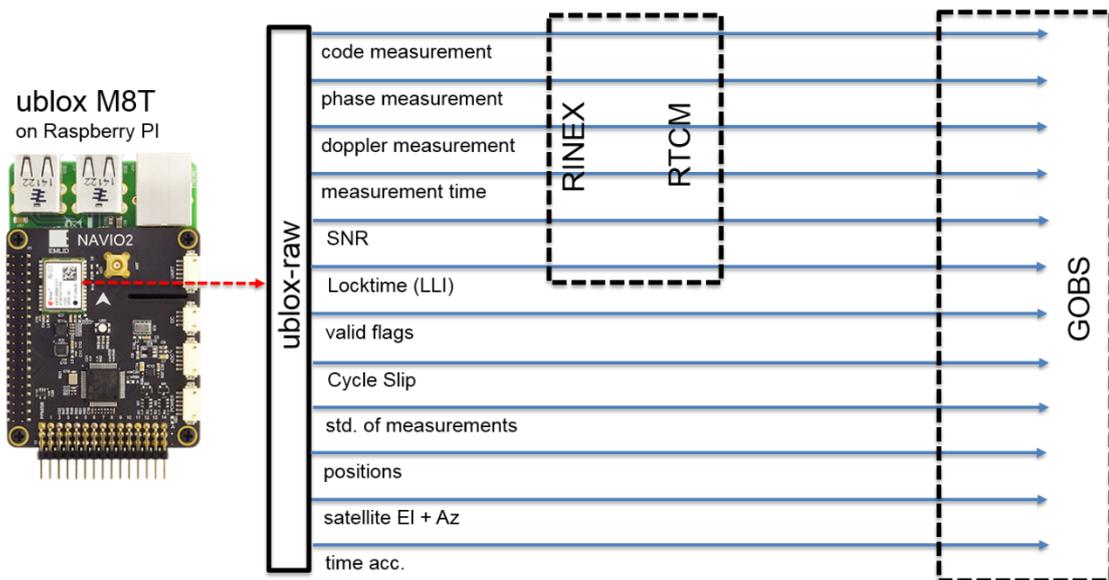


Abbildung 3.6: Struktur von GOBS

Die detaillierten Bedeutungen der einzelnen Parameter können der Tabelle 3-3 entnommen werden. Der Aufbau des Formates ist dem Anhang A zu entnehmen. Der wichtigste Qualitätsgewinn durch das Einbeziehen aller vorhandenen Parameter des GOBS-Formates, wie LLI, valid-flags oder Cycle Slip Parameter, wird nachfolgend gezeigt und erläutert.

- **LLI und Locktime:**

Der standardisierte Qualitätsparameter Loss of Lock Indicator LLI einer GNSS-Messung gibt Auskunft über den Phasenlock, DLL und PLL, einer GNSS-Messung. Vor allem in urbanen und anderen stark abgeschattet Messgebieten treten im Zuge von GNSS-Messungen, insbesondere bei der Phasenmessungen, Probleme beim Tracking der Trägerphase auf. Der GNSS-Empfänger verliert den Lock der Phasenmessungen und die Ambiguitätenlösung ist nicht mehr gültig, siehe Abschnitt 6.4.4. Dadurch entstehen Cycle Slip in der Messung, siehe Abschnitt 7.3.4.

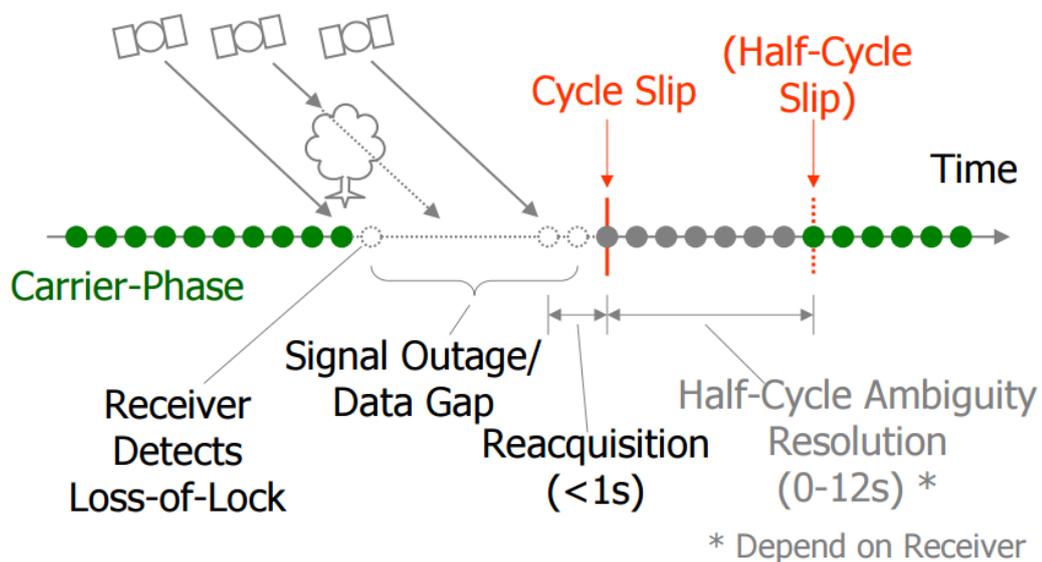


Abbildung 3.7: Cycle Slip nach Cycle Slip Detection and Fixing by MEMS-IMU/GPS Integration for Mobile Environment RTK-GPS (2008)

Abbildung 3.7 zeigt das Grundprinzip von auftretenden Cycle Slip. In abgeschatteten Gebieten geht das GNSS-Tracking verloren. Außerdem ist nach einer Requisition die Phasenmessung nicht mehr gültig, da die DLL und PLL keinen Lock haben, und Cycle Slips treten auf. Nach dem Auftreten benötigt die RTK Software laut Cycle Slip Detection and Fixing by MEMS-IMU/GPS Integration for Mobile Environment RTK-GPS (2008) mehrere Minuten, um einen Cycle Slip zu korrigieren. Empfänger bieten hierzu einen LLI Flag an, um einen möglichen Cycle Slip zu erkennen. Vor allem bei low-cost Empfängern ist das Tracking eines Satelliten jedoch so sensitiv, dass dieser LLI sehr anfällig für fehlerhafte Informationen ist.

Der u-blox M8T Empfänger bietet hierzu eine Locktime des Signals, anstatt des LLI, als Ausgabe an (u-blox proprietär: 0-64500 ms). Sobald das Signal wieder akquiriert wurde, zählt ein Counter von 0 bis max. 64500 ms (64,5 s). Somit kann bzw. sollte ein Signal erst nach gewisser Zeit herangezogen werden. Im Zuge dieser Arbeit stellte sich ein Wert um 3000 ms als geeignet heraus, um Cycle Slip behaftete Messung auszuschließen bzw. diese frühzeitig detektieren zu können. Grundlegend besteht ein LLI aus einem Wert, welcher Auskunft über die Fehleranfälligkeit der Messung gibt, dargestellt in Tabelle 3-7.

Tabelle 3-7: LLI Spezifikation nach IGS (2013)

LLI	1	2	3
Loss of lock indicator	Lost lock between previous and current observation: Cycle Slip possible	Half-cycle ambiguity/slip possible	Bit 1 and 2 set or under spoofing
Reaction	dismiss satellite	Cycle Slip Detektion	dismiss satellite

Da der LLI-Flag jedoch nicht im *UBX* Format implementiert ist, berechnet er sich im GOBS Format aus der Locktime, den valid-Flags, Cycle Slip Parametern und der Standardabweichung der Phasenmessung.

- **Valid-Flags**

Die Valid-Flags geben eine grundlegende Auskunft darüber, ob der Phasen-, bzw. der Codemessung vertraut werden kann. Dies wird u-blox proprietär berechnet, was jedoch nicht nachvollzogen werden kann. Sollte der Flag ungültig gesetzt sein, wird die Phasenmessung automatisch verworfen und der LLI auf 1 gesetzt.

- **Cycle Slip**

Die u-blox proprietären Cycle Slip Informationflags geben zusätzliche Auskunft über mögliche Cycle Slip bzw. Half Cycle Slip. Sollte der Cycle Slip Flag aktiv sein und sonstige Parameter wie Locktime oder Valid Flags nicht auf einen möglichen Fehler hinweisen, wird der LLI auf 2 gesetzt. Der Half Cycle Slip Flag wird zurzeit nicht beachtet, dass dieser u.a. für den Pilot und Data Channel wegfällt.

- **Standardabweichung der Beobachtungen**

U-blox bietet Auskunft über die geschätzte Standardabweichung der jeweiligen Rohdatenmessung, wie Code-, Phasen- oder Dopplermessung. Im Zuge dieser Arbeit wird jedoch nur die Standardabweichung der Phasenmessung für eine weitere Verarbeitung verwendet. Grundlegend setzt dich der Wert aus vier Bits (Dezimal 1-15) zusammen, welcher durch einen Skalierungsfaktor zur Standardabweichung abgewandelt wird. Für die Verwendung in GOBS werden jedoch nur die Bit-Werte herangezogen, wobei bei Werten größer 10 der Satellit verworfen wird und der LLI auf 1 gesetzt wird. Ist der Wert im Bereich 4-13, wird je nach Zusammensetzung der anderen Qualitätsparameter der LLI auf 2 oder 3 gesetzt, um mögliche Cycle Slips zu detektieren. Außerdem wird dieser Wert für eine erweiterte Gewichtung der Phasenmessung im Zuge der RTK-Prozessierung herangezogen, siehe Abschnitt 7.3.3.

- **Positionen, Azimut + Elevation und Uhrparameter**

Die sonstigen Werte werden herangezogen, um eine effizientere RTK-Prozessierung zu gewährleisten. Außerdem können diese Werte als Kontrollmöglichkeit mit den in der RTK-Prozessierung berechneten Werten, wie, Position, Empfänger-Uhrparameter (Bias und Drift) sowie Azimut und Elevation herangezogen werden. Sollte z.B. die RTK-Position mehrere 10 Meter von der u-blox evaluierten Position abweichen, kann die errechnete Position verworfen und ein spezielles Error-Handling eingesetzt werden.

3.6 Fazit

Grundlegend wird im Zuge von GNSS-Datenformaten in (i) Standardisierte Formate, (ii) empfängerspezifische Formate sowie (iii) in Postprocessing und (iv) Echtzeit-Formaten untergliedert. Eine Zusammenfassung der Formate kann der Tabelle 3-8 entnommen werden.

Tabelle 3-8: Formate Fazit

Anmerkung	RINEX 2.xx – 3.xx	RTCM 2.xx - 3.xx	UBX	GOBS
Verwendung	Postprocessing	Echtzeit	Postprocessing + Echtzeit	Postprocessing + Echtzeit
Messungen	C, CP, D, SNR	C, CP	alle	alle
Qualität	LLI, SNR	LLI, SNR	alle	alle + erweitert
Standard	Ja	Ja	Nein	Nein
Empfänger-abhängig	Nein	Nein	Ja	Ja, nur u-blox M8T

mit:

C ... Code

CP ... Carrier Phase

D ... Doppler

LLI ... Loss of Lock Indicator

SNR ... Signal to Noise Ratio

Kapitel 4

Übertragungskonzepte

4.1 Grundlagen

Die Vernetzung aller möglicher Daten im sogenannten Internet Of Things macht auch vor der Navigationssparte keinen Halt. Für etwaige Anwendungen ist es erforderlich, Rohdaten, Korrekturdaten und Positionsdaten von Navigationsempfängern über das Internet zu verteilen und für externe Anwendungen zur Verfügung zu stellen oder zu speichern. Vor allem im Bereich der relativen GNSS-Positionierung werden bereits Roh- sowie Korrekturdaten von nationalen Betreibern wie dem Vermessungsamt über das Internet verteilt. Diese Daten, meist im RTCM-Format, können dann von high-cost Systemen empfangen werden und eine genauere Positionierung ist möglich, siehe Kapitel 5 und 6. Als Beispiel wird hier der Austrian Positioning Service APOS des Bundesamtes für Eich- und Vermessungswesen genannt. Aber auch im low-cost Bereich ist es möglich, eigene Korrekturnetze aufzubauen.

Verantwortlich für die Rohdatenübertragung, wie sie bei RTK zwingend notwendig ist, ist eine in dieser Arbeit entwickelte Systemarchitektur, bestehend aus einer UMTS-Datenübertragung, eines proprietären GNSS-Rohdatenformates GOBS und einer eigens definierten Schnittstelle zur Synchronisierung der Daten in Echtzeit. Für das Streamen von Navigationsdaten über das Internet stehen mehrere standardisierte Protokolle zur Verfügung. Nachfolgend werden die beiden bedeutendsten Protokolle zur Verteilung von Navigationsdaten via Internet, TCP/IP und NTRIP näher dargestellt.

4.2 TCP/IP

Die grundlegende Kommunikation mittels TCP/IP Standard basiert nach (Meinel, et al., 2013) auf Paketen, wobei ein Paket einen Teil der zu sendenden Daten beinhaltet. Prinzipiell sind die Protokolle TCP (Transmission Control Protocol)

und IP (Internet Protocol) die beiden Techniken, welche für eine korrekte Datenübertragung genutzt werden. Das angewandte, in Tabelle 4-1 dargestellte Layer-Prinzip macht es möglich, definierte Anwendungen gewissen Layern zuzuordnen. TCP/IP definiert sich nach Meinel et al., (2013) durch vier Layer.

Tabelle 4-1: TCP/IP Layer

Application
Transport
Network
Link

Der Link-Layer dient zur Kontrolle und Verwaltung der Interfaces zwischen Sender und Empfänger. Der übergeordnete Network Layer, welcher dem IP-Standard zugeordnet ist, verwaltet das Packet-Routing und die Adressverwaltung. Der TCP-Standard regelt den Transport-Layer, welcher den Übertragungsfluss (packet flow) der Übertragung innehat.

Der Top-Layer, Application, beinhaltet die einzelnen Services und Funktionen einer TCP/IP Übertragung, z.B. HTTP, SMTP oder auch den für die GNSS-Rohdatenübertragung verantwortlichen NTRIP-Standard. Nähere Information zur Übertragung via TCP/IP Technik kann Meinel et al., (2013) entnommen werden.

4.3 NTRIP

Das nach BKG (2013) entwickelte Networked Transport of RTCM via Internet Protocol NTRIP ist ein Protokoll zur Übertragung von GNSS-Daten aller Art über das Internet. Entwickelt wurde es im Zuge von Referenzsystemmessungen (EUREF) vom Bundesamt für Kartographie und Geodäsie Deutschland in Zusammenarbeit mit der Universität Prag.

Dieses Protokoll ermöglicht es, mehreren Nutzern gleichzeitig Korrekturdaten sowie Rohdaten von mehreren Referenzstationen zur Verfügung zu stellen. Es ist auch möglich, eigene Referenzstationen mit diesem Protokoll zu betreiben. Der grundlegende Aufbau des NTRIP Standards besteht laut GNSS-

Echtzeitorbitkontrolle auf Basis Internet-transferierter (NTRIP) RTCM-Datenströme (2005) aus vier Elementen:

- NTRIP-Source,
- NTRIP-Server,
- NTRIP-Caster,
- NTRIP-Client.

Die NTRIP Datenquelle NTRIP-Source, welche zumeist aus einem GNSS-Rohdatenempfänger besteht, generiert die GNSS-Daten und leitet diese an einen NTRIP-Server weiter. Der NTRIP-Server stellt eine Verbindung zu dem NTRIP-Caster der als Broadcaster fungiert, her und leitet die Daten an diesen weiter. Dieser NTRIP-Caster verteilt die Daten an verschiedene NTRIP-Clients. Das Senden und Empfangen der Daten basiert auf dem TCP/IP-Protokoll. Ein grundlegender Aufbau des NTRIP-Konzepts nach BKG (2013) kann der Abbildung 4.1 entnommen werden.

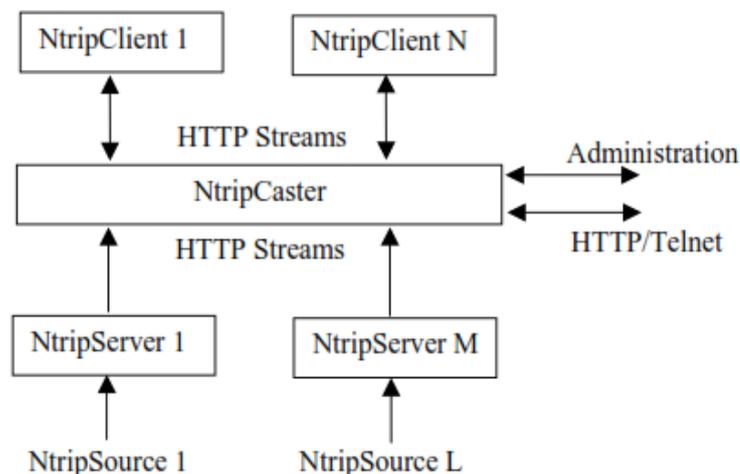


Abbildung 4.1: NTRIP-Struktur nach RTCM SPECIAL COMMITTEE NO. 104

Der NTRIP-Caster ist das Herzstück jeder NTRIP-Datenübertragung. Er stellt sicher, dass die korrekten Daten einer bestimmten Referenzstation an die einzelnen User weitergegeben werden. Die eigentliche Übertragung der Daten verwendet das auf dem IP basierten für NTRIP abgewandelten HTTP-Protokoll. Somit ist es möglich, mehrere User gleichzeitig mit dem gleichen Port zu verbinden.

NTRIP Caster

Der NTRIP-Caster, welcher mittels IP-Adresse und speziellen Port erreichbar ist, verwendet sogenannte Mountpoints, um einzelne Referenzstationen voneinander zu unterscheiden. So verbindet sich jede Referenzstation mit einem bestimmten Mountpoint am NTRIP-Caster. Mehrere Benutzer, NTRIP-Clients, können sich dann gleichzeitig mit dem gewünschten Mountpoint verbinden. Sollte keine bestimmte Referenzstation mittels Mountpoint ausgewählt werden, ist es dem User auch möglich, selbst einen Mountpoint zu wählen. Dazu sendet der NTRIP-Client seine NMEA-Näherungsposition und der Caster sendet die Daten der am nächsten liegenden Referenzstation. Dank der HTTP-Verbindung kann ständig eine Wartung mittels Fernadministration des NTRIP-Casters erfolgen. Eine vollständige NTRIP Beschreibung ist RTCM SPECIAL COMMITTEE NO. 104 zu entnehmen.

4.4 Analyse der Übertragungsformate

Tabelle 4-2 beschreibt eine Analyse der einzelnen Datenformate via NTRIP oder TCP/IP. Es wurde eine 10-minütiges Testszenario der Datenübertragung analysiert, wobei (i) das konzipierte Format GOBS, (ii) das standardisierte Format RTCM, (iii) das erweiterte Format RTCM mit GOBS und (iv) die empfangerspezifischen Rohdatenformate eines inertialen Sensors übertragen wurden. Das standardisierte Format RTCM benötigt hier die deutlich geringste Datenrate, was zum einem an einem ausgeklügelten Datenformat, siehe RTCM SPECIAL COMMITTEE NO. 104 (2006), zum anderem jedoch auch daran liegt, dass nur ein Bruchteil der vorhandenen Rohdaten übertragen werden. Wird das Format um alle Daten, welche im GOBS Format vorhanden sind, erweitert, steigert sich die Datenrate auch hier und nähert sich dem GOBS-Format an. Die Rohdaten des inertialen Sensors benötigen eine deutliche größere Datenrate. Dies ist der höheren Updaterate (GNSS 2 Hz und inertial 50 Hz) geschuldet. Übertragen werden hierbei nur die 9DOF-Parameter sowie Temperatur und barometrischer Druck im klassischem TCP/IP Modus.

Tabelle 4-2: Übertragungsanalyse

Daten	Modus	Datarate [kbps]	Mode
Stream I/O ublox	GOBS (RAW)	15 - 30	NTRIP
	RTCM	4 - 6	NTRIP
	RTCM - GOBS	10 - 20	NTRIP
Stream I/O IMU+Baro	RAW	~150	TCP

Daten	Modus	Datenmenge nach 10 min [MByte]
Stream I/O ublox	GOBS (RAW)	~ 1.9
	RTCM	~ 0.4
	RTCM - GOBS	~ 1.4
Stream I/O IMU+Baro	RAW	~ 11.3

Kapitel 5

Relative Positionierung

5.1 Grundlagen

Die in GNSS-Anwendungen am häufigsten eingesetzte Technik ist grundsätzlich die relative Positionierung. Dazu wird über mindestens einen koordinatenmäßig bekannten Punkt (Referenzpunkt A) ein rohdatenfähiger GNSS-Empfänger aufgebaut. Die Bestimmung des Neupunktes B geschieht dann relativ zu diesem Referenzpunkt, entweder in Echtzeit (RTK) oder im Postprocessing, siehe Abbildung 5.1

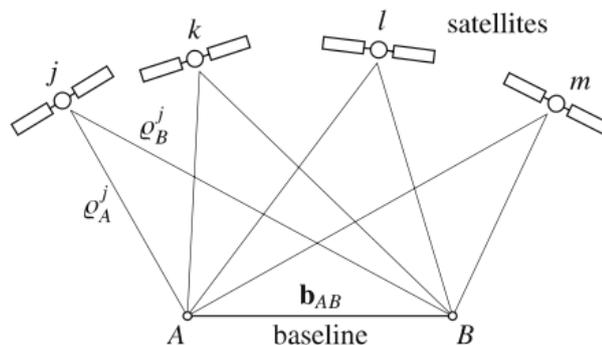


Abbildung 5.1: Prinzip der relativen Positionierung nach Hofmann-Wellenhof et al., (2008)

Die meisten Fehlereinflüsse nach Hofmann-Wellenhof et al., (2008), wie Atmosphärenfehler oder i.a. satellitenabhängige Fehler, können dadurch eliminiert werden. Multipath Einflüsse bleiben jedoch vorhanden und müssen im Zuge der Prozessierung entsprechend beachtet werden, siehe Abschnitt 7.3.4.

Für eine präzise Positionsbestimmung, wie der relativen Positionierung oder dem Precise Point Positioning, ist die Verwendung der Trägerphasenmessung essenziell. Durch eine ständige Messung der Trägerphase wird immer das Phasenreststück gemessen, die Anzahl der unbekanntes Cycles ist unbekannt aber im Allgemeinen über eine ständige Messung konstant, siehe Abschnitt 5.2.

5.2 Trägerphasenmessung

Wie in Kapitel 2.2.1 beschrieben, ist die im DSP angesiedelte Phasenregelschleife PLL für eine ständige Phasenmessung verantwortlich. Damit ist die Messung der Änderung der Phasenmessung zwischen dem Trägersignal und einem im Empfänger erzeugten Referenzsignal möglich.

Nach Hofmann-Wellenhof et al., (2008) kann die Trägerphasenmessung wie folgt hergeleitet werden:

$$\begin{aligned}\varphi^s(t) &= f^s t - f^s \frac{p}{c} - \varphi_0^s, \\ \varphi_r(t) &= f_r t - \varphi_{0r},\end{aligned}\tag{5-1}$$

wobei $\varphi^s(t)$ die Phase des vom Satelliten im Empfänger mit einer Frequenz f^s ankommenden Signals zum Zeitpunkt t beschreibt. $\varphi_r(t)$ beschreibt somit die Phase des im Empfänger generierten Referenzsignals mit der Frequenz f_r zum Zeitpunkt t mit den in der Tabelle angegebenen Einheiten.

Tabelle 5-1: GNSS-Einheiten

Größe	Symbol	Einheit GNSS	Einheit SI
Frequenz	f	cycles / sec	1 /sec = Hz
Wellenlänge	λ	m / cycle	m
Phase	φ	cycle	rad
Zeit	t, δ	sec	sec

Die initialisierten Phasenmessungen φ_{0r} und φ_0^s werden durch Uhrenfehler beschreibbar und sind in Formel 5-2 dargestellt, vgl Hofmann-Wellenhof et al., (2008).

$$\begin{aligned}\varphi_0^s &= -f^s \delta^s, \\ \varphi_{0r} &= -f_r \delta_r,\end{aligned}\tag{5-2}$$

Durch Differenzbildung, der in Formel 5-1 eingeführten Phasen ergibt sich die Beat Phase. Diese Beat Phase ist die eigentliche Messgröße der Trägerphasenmessung, jedoch nur als Phasenreststück. Die Anzahl der ganzen Wellenlängen

zwischen Satellit und Empfänger ist stets unbekannt. Die Messung des Phasenreststücks ist jedoch sehr genau und liegt bei 1/100 bis 1/1000 der Wellenlänge. Als Beispiel kann bei GPS L1 ($f_0=1575.42$ MHz, mit $\lambda = 19.0$ cm) das Phasenreststück mit einer Auflösung von rund 2 mm gemessen werden, wobei die Codemessung nur mit einer Auflösung von rund 3 m gemessen wird, siehe (Kaplan, 2006). Somit ergibt sich die Beat-Phase $\varphi_r^s(t)$ mit,

$$\begin{aligned}\varphi_r^s(t) &= \varphi^s(t) - \varphi_r(t), \\ &= -f^s \frac{\rho}{c} + f^s \delta^s - f_r \delta_r + (f^s - f_r)t.\end{aligned}\tag{5-3}$$

Die Abweichung der Frequenzen f^s und f_r von der Nominalfrequenz f beträgt laut (Hofmann-Wellenhof, et al., 2008) nur rund 10^{-3} Hz. Dies führt aufgrund der Signalausbreitung zu einem Fehler von ca. 10^{-4} Cycles in der Phasenmessung und kann somit vernachlässigt werden. Somit ergibt sich die vereinfachte Form

$$\varphi_r^s(t) = -f \frac{\rho}{c} - f \Delta \delta.\tag{5-4}$$

Wird ein Empfänger zum Zeitpunkt t_0 eingeschaltet, wird das Phasenreststück ($0 \leq \varphi \leq 2\pi$) gemessen. Die Anzahl der ganzen Wellenlängen ist nicht bekannt, ist aber in der Annahme, dass das Signal die gesamte Messdauer gelockt ist, zeitlich konstant. Diese unbekannte Anzahl an Cycles N wird Integer Ambiguität genannt, da man weiß, dass nur ganzzahlige Cycles vorkommen können. Die Beat Phase zum Zeitpunkt t ist somit

$$\varphi_r^s(t) = \Delta\varphi_r^s|_{t_0}^t + N,\tag{5-5}$$

wobei $\Delta\varphi_r^s$ die Messgröße definiert. Eine geometrische Darstellung ist der Abbildung 5.2 zu entnehmen.

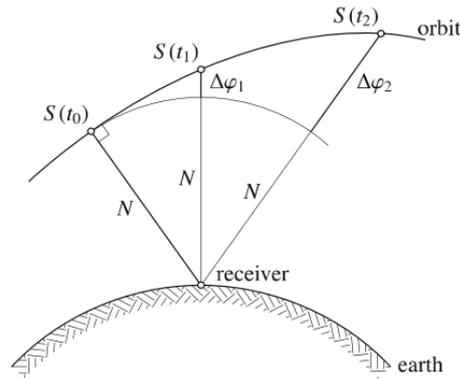


Abbildung 5.2: Geometr. Interpretation der Phasenmessung nach Hofmann-Wellenhopf, et al., (2008)

Die Kombination der Formel 5-4 und 5-5 führt zur vereinfachten Gleichung der Trägerphasenmessung,

$$\Phi(t) = \frac{1}{\lambda} \rho(t) + \frac{c}{\lambda} \Delta\delta(t) + N. \quad 5-6$$

$\Phi(t)$...	Phasenmessung (Messgröße) [cycles]
λ	...	Wellenlänge (L1... 19.2 cm) [m]
c	...	Vakuumlichtgeschwindigkeit [m/s]
$\rho(t)$...	geometrische Strecke Empfänger zu Satellit [m]
$\Delta\delta$...	kombinierter Uhrenfehler (Satellit-Empfänger) [s]
N	...	Phasenambiguität [cycles]

Der hauptsächliche Unterschied zur Codemessung, Formel 3-2, ist somit die Ambiguität N . Nach Hofmann-Wellenhopf et al., (2008) muss erwähnt werden, dass das Vorzeichen der resultierenden Phasenmessung nicht definiert ist und empfängerabhängig ist. Jedoch wird inkrementiert und die Phasenmessung muss über die Zeit größer werden. Nach Koch (2012) verfolgt der Empfänger die Änderung der Trägerphase und erhöht bzw. senkt den Integerzähler, wenn die gemessene Phase den Messbereich eines Cycles verlässt. Kommt es jedoch durch Abschattung zur Unterbrechung des Locks eines Satelliten (Tracking geht verloren), muss der Integerzähler des Empfängers reinitialisiert werden. Tritt dies ein, so ist die Phasenmessung einem Sprung – einem Cycle Slip – unterworfen, siehe Abbildung 3.7. Im Zuge der relativen Positionierung wird die Ambiguität N als zusätzliche Unbekannte geführt. Werden alle Fehlereinflüsse nach Hofmann-

Wellenhof et al., (2008) in die Formel 5-6 angebracht, ergibt sich folgende Schreibweise der Trägerphasenmessung

$$\Phi_r^s(t) = \frac{1}{\lambda_{L1}} \rho_r^s(t) + \frac{c}{\lambda_{L1}} (\delta_r(t) - \delta^s(t)) + N_r^s - \Delta\rho_{iono}^s(t) + \Delta\rho_{tropo}^s(t) + \epsilon_r(t) + \varepsilon(t), \quad 5-7$$

mit folgenden Parametern:

$\Phi_r^s(t)$...	Phasenmessung (Messgröße) [cycles]
λ	...	Wellenlänge (L1... 19.2 cm) [m]
c	...	Vakuumlichtgeschwindigkeit [m/s]
$\rho(t)$...	geometrische Strecke Empfänger zu Satellit [m]
$\delta_r(t) - \delta^s(t)$...	kombinierter Uhrenfehler (Empfänger-Satellit) [s]
N_r^s	...	Phasenambiguität [cycles]
$\Delta\rho_{iono}^s$...	Fehlereinflüsse der dispersiven Ionosphäre (Beschleunigung des Signals, dadurch negativ)
$\Delta\rho_{tropo}^s$...	Fehlereinflüsse der Troposphäre
$\epsilon_r(t) + \varepsilon(t)$...	Multipath und relativistische Fehler

Für die nachfolgende Berechnung der relativen Positionierung wird nur die Phasengleichung herangezogen. Die relativen Fehlerterme werden vernachlässigt und die Darstellung und Formulierung bezieht sich auf Hofmann-Wellenhof, et al., (2008).

5.3 Single-differences

Im Zuge von Einfachdifferenzen (single-differences) sind zwei Empfänger (A und B) und ein Satellit (j) involviert. Die Gleichungen,

$$\begin{aligned} \Phi_A^j(t) &= \frac{1}{\lambda} \rho_A^j(t) + \frac{c}{\lambda} (\delta_A(t) - \delta^j(t)) + N_A^j, \\ \Phi_B^j(t) &= \frac{1}{\lambda} \rho_B^j(t) + \frac{c}{\lambda} (\delta_B(t) - \delta^j(t)) + N_B^j, \end{aligned} \quad 5-8$$

werden dann durch das Bilden einer linearen Kombination zur Lösung der Einfachdifferenzen verwendet. Durch diese Differenzbildung erlischt der Satellitenuhrenfehler $\delta^j(t)$

$$\Phi_{AB}^j(t) = \frac{1}{\lambda} \rho_{AB}^j(t) + \frac{c}{\lambda} \delta_{AB}(t) + N_{AB}^j \quad 5-9$$

5.4 Double-differences

Im Vergleich zu Einfachdifferenzen werden bei Doppeldifferenzen (double-differences) neben zwei Empfängern (A und B) auch noch ein zweiter Satellit (j und k) hinzugezogen. Somit benötigt man bereits zwei Einfachdifferenzen, einmal zu einem Satelliten j und einmal zu einem Satelliten k. Die Formeln,

$$\begin{aligned} \Phi_{AB}^j(t) &= \frac{1}{\lambda} \rho_{AB}^j(t) + \frac{c}{\lambda} \delta_{AB}(t) + N_{AB}^j, \\ \Phi_{AB}^k(t) &= \frac{1}{\lambda} \rho_{AB}^k(t) + \frac{c}{\lambda} \delta_{AB}(t) + N_{AB}^k, \end{aligned} \quad 5-10$$

werden dann linear kombiniert. Es ist immer eine simultane Messung mit zwei Empfängern zu zwei Satelliten notwendig. Laut

$$\Phi_{AB}^{jk}(t) = \frac{1}{\lambda} \rho_{AB}^{jk}(t) + N_{AB}^{jk}, \quad 5-11$$

wird der kombinierte Empfängeruhrenfehler $\delta_{AB}(t)$ dadurch eliminiert. Diese Elimination ist die Ursache für die häufige Verwendung dieser Methode. Ein weiterer, in den Gleichungen nicht dargestellter Vorteil, ist die nahezu gesamte Elimination der atmosphärischen Einflüsse. Da laut Kaplan (2006) bei nahegelegenen Stationen (A und B in einer Entfernung bis zu ~ 10 km) gleiche atmosphärische Einflüsse herrschen.

Im Zuge der Auswertung der Positionen wurden nur doppeldifferenzierte Beobachtungen (Phasen- und Codemessungen) verwendet.

5.5 Triple-differences

Bei Dreifachdifferenzen werden Doppeldifferenzen von zwei Epochen linear kombiniert um damit die zeitunabhängige Phasenambiguität N_{AB}^{jk} zu eliminieren. Aufgrund des entstehenden hohen Rauschen dieser Methode, siehe Hofmann-Wellenhof et al., (2008), findet diese Methode in der Praxis kaum Anwendung und wurde auch im Laufe dieser Arbeit nicht verwendet. Näheres zu dieser Methode kann Hofmann-Wellenhof et al., (2008) entnommen werden.

5.6 Koordinatensysteme

Zur Beschreibung der Position von Empfängern aber auch von Satelliten ist es notwendig, auf standardisierte Bezugssysteme zurückzugreifen. Per Definition wird zwischen einem Bezugssystem und einem Bezugsrahmen unterschieden. Das Bezugssystem, z.B. International Terrestrial Reference System, beschreibt die theoretische Definition des Bezugsrahmens. Dieser Rahmen beschreibt die Realisierung in der Natur z.B. International Terrestrial Reference Frame. Nachfolgend werden die im Zuge für GNSS-Positionierung verwendeten Bezugsrahmen beschrieben. Im deutschsprachigen Raum wird jedoch kein Unterschied zwischen Rahmen und System gesehen. Die Definitionen und Formulierungen wurden aus Hofmann-Wellenhop et al., (2008) übernommen.

5.6.1 ECEF

Das erdfeste Äquatorsystem rotiert mit der Erde mit und dient zur Beschreibung von Punkten auf der Erdoberfläche. Die Koordinatenbestimmung mittels GNSS liefert immer Punkte im erdfesten Äquatorsystem. Das Geozentrum beschreibt den Mittelpunkt des Systems, die X-Achse (X_1) liegt in der Äquatorebene und zeigt Richtung Greenwich-Meridian, die Z-Achse (X_3) zeigt in Richtung der Erdrotationsachse und die Y-Achse (X_2) bildet darauf ein orthogonales Rechtssystem. Die Definition kann der Abbildung 5.3 entnommen werden.

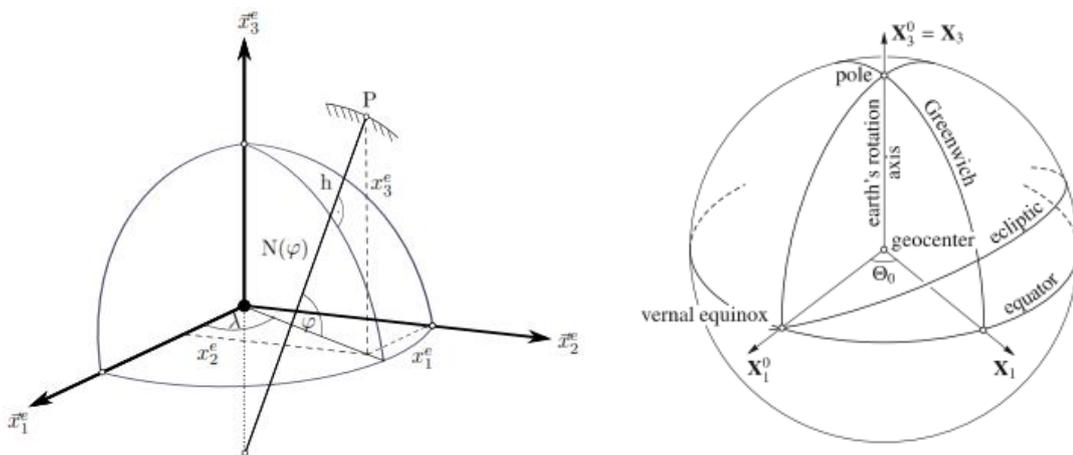


Abbildung 5.3: ECEF, kartesisch und ellipsoidisch nach Hofmann-Wellenhop et al., (2008)

Abbildung 5.3 zeigt außerdem den geometrischen Zusammenhang zwischen ellipsoidischen Koordinaten (φ , λ , h) und den kartesischen Koordinaten (X , Y , Z). Ellipsoidische Koordinaten benötigen im Gegensatz zu kartesischen Koordinaten ein Referenzellipsoid, welches nach Hofmann-Wellenhof et al., (2008) dem World Geodetic System 84 (WGS84) entspricht.

Transformation zwischen kartesischen- und ellipsoidischen Koordinaten

Nach Hofmann-Wellenhof et al., (2008) erfolgt die Umrechnung von ellipsoidischen- zu kartesischen Koordinaten nach den Formeln

$$\begin{aligned} X &= (N + h)\cos\varphi \cos\lambda , \\ Y &= (N + h)\cos\varphi \sin\lambda , \\ Z &= \left(\frac{b^2}{a^2}N + h\right) \sin\varphi , \end{aligned} \tag{5-12}$$

wobei N den Normalkrümmungsradius darstellt und in Formel 5-13 angeführt ist.

$$N = \frac{a^2}{\sqrt{a^2\cos^2\varphi + b^2\sin^2\varphi}} , \tag{5-13}$$

Einer Transformation von kartesischen zu ellipsoidischen Koordinaten liegt nach (Hofmann-Wellenhof, et al., 2008) eine Approximation zu Grunde und wird nach den Formeln

$$\begin{aligned} \varphi &= \arctan \frac{Z + e'^2 b \sin^3\theta}{p - e^2 a \cos^3\theta} \\ \lambda &= \arctan \frac{Y}{X} , \\ h &= \frac{p}{\cos\varphi} - N , \end{aligned} \tag{5-14}$$

Beschrieben, wobei die Hilfsgrößen e' , e , θ und p Hofmann-Wellenhof, et al., (2008) entnommen werden können.

5.6.2 ECSF

Das raumfeste Äquatorkoordinatensystem stellt nach Hofmann-Wellenhof et al., (2008) ein quasi-inertiales System dar, welches sich im Zustand der Ruhe oder einer gleichförmigen Bewegung befindet (inertiales System). Da es jedoch Beschleunigungen durch die Bewegung um die Sonne unterliegt, ist es ein quasi-inertiales System. Das Geozentrum beschreibt den Mittelpunkt des Systems, die X-Achse (X_1^0) liegt in der Äquatorebene und zeigt Richtung Frühlingspunkt (vernal equinox), die Z-Achse (X_3^0) zeigt in Richtung der Erdrotationsachse und die Y-Achse (X_2^0) bildet darauf ein orthogonales Rechtssystem, siehe Abbildung 5.3. Dieses nicht rotierende System wird benötigt, um Satellitenbahnen mit Hilfe der Kepler-Parameter zu beschreiben, siehe Abbildung 5.4.

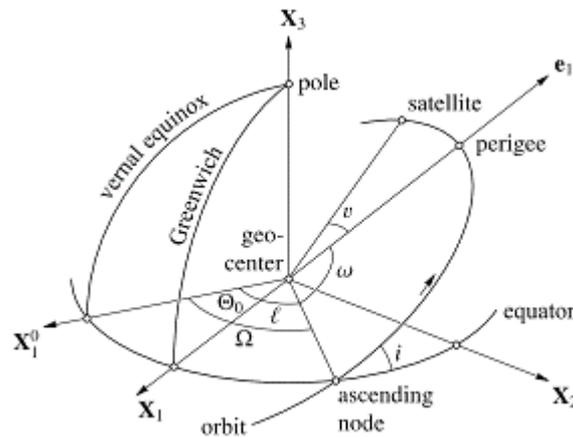


Abbildung 5.4: Keplerbahnparameter nach Hofmann-Wellenhof et al., (2008)

Mit den Parametern nach Tabelle 5-2 Hofmann-Wellenhof et al., (2008).

Tabelle 5-2: Keplerparameter

Parameter	Beschreibung/Name
Ω	Rektaszension des aufsteigenden Knoten
i	Inklination der Orbitebene
ω	Argument des Perigäums
a	Große Halbachse der Orbitellipse
e	Numerische Exzentrizität
T_0	Perigäum Durchgangszeit.

Die Transformationen zwischen erdfesten und raumfesten Koordinatensystem kann Hofmann-Wellenhof et al., (2008) entnommen werden.

5.6.3 Lokales Horizontsystem

Das lokale Horizontsystem ist ein Koordinatensystem, welches sich mit der Erde in Rotation befindet und den Mittelpunkt im Topozentrum (P), dem Standpunkt, oder einen sonstigen beliebigen Punkt, definiert hat. Die Z-Achse (d^e) zeigt Richtung Nadir oder Zenit. Die X-Achse (n^e) zeigt Richtung Norden und die Y-Achse (e^e) ist orthogonal dazu und zeigt Richtung Osten. Die Komponenten werden dadurch meist als north-east-up (oder north-east-down) bezeichnet. Geodätische Messungen zu Satelliten (Azimut oder Elevation) werden in diesem System ausgedrückt. Auch Geschwindigkeiten beziehen sich zumeist auf das lokale Horizontsystem. Eine grafische Darstellung kann Abbildung 5.5 entnommen werden, vgl. Hofmann-Wellenhof et al., (2003)

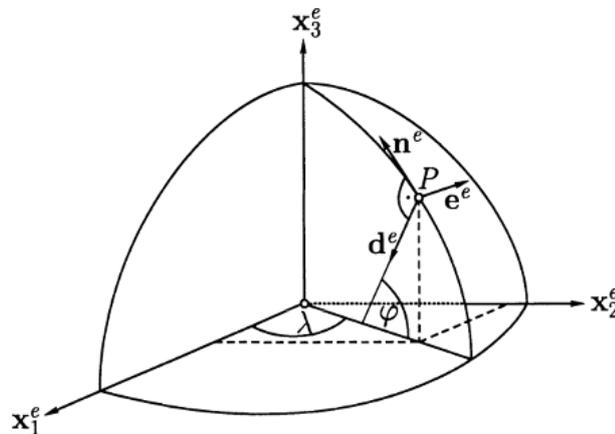


Abbildung 5.5: Lokales Horizontsystem nach Hofmann-Wellenhof et al., (2003)

Für die benötigten Transformationen zwischen den einzelnen Systemen wird auf Hofmann-Wellenhof et al., (2008) und Hofmann-Wellenhof et al., (2003) verwiesen. Die genannten drei Koordinatensysteme sind die für die GNSS-Positionierung verwendeten Bezugssysteme.

5.7 Algorithmus

Der Algorithmus zur Durchführung einer relativen Positionierung kann Hofmann-Wellenhof et al., (2008) oder Kaplan (2006) entnommen werden. Der Im Zuge dieser Arbeit verwendete Algorithmus zur RTK Positionierung wird nach der Formulierung von Takasu (2013) im Kapitel 6 beschrieben.

Kapitel 6

RTKLIB

6.1 Grundlagen

RTKLIB nach Takasu (2013) ist eine Sammlung von Open-Source C/C++-basierten Anwendungen für GNSS Positionierung und GNSS Auswertung. Die Anwendungen sind sowohl für Postprocessing als auch für Echtzeitanwendungen konzipiert und unterstützen alle GNSS Systeme und Auswertetechniken. Im Zuge dieser Arbeit wurde die Technik der relativen Positionierung in Echtzeit, RTK (Real Time Kinematic) verwendet und wird in den nachfolgenden Kapiteln näher erläutert. RTKLIB besteht aus WINDOWS und UNIX kompilierten Anwendungen, wobei die Windows Anwendungen mit grafischer Oberfläche gestaltet wurden und die UNIX Anwendungen konsolenbasiert sind. Für die lizenzierte Verwendung der RTKLIB-Anwendungen wird auf die GPLv3 Lizenz und auf Takasu (2013) verwiesen.

Die unterschiedlichen Anwendungen können aus Takasu (2013) und Takasu (2013) bzw. dem Abschnitt 6.2 entnommen werden. Die im Zuge dieser Arbeit verwendete Anwendung der Echtzeitpositionierung RTKNAVI wird in Kapitel 7 näher beschrieben.

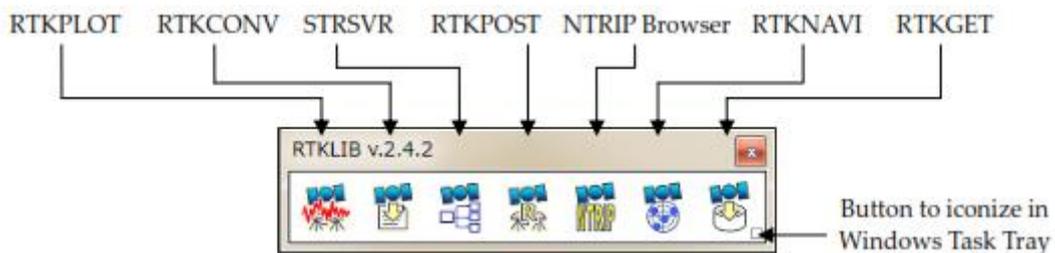


Abbildung 6.1: RTKLIB Menüführung nach Takasu (2013)

6.2 Zugehörige Anwendungen

Wie zuvor erwähnt, ist RTKLIB eine Sammlung von Softwarepaketen zur Auswertung, Analyse und Übertragung von GNSS-Daten. Implementiert sind mehrere mögliche Positionierungsmethoden, GNSS-Konstellationen, max. zwei Frequenzen und eine Vielzahl der gängigsten rohdatenfähigen GNSS-Empfängern im high-cost sowie im low-cost Bereich. Die einzelnen Programme sind:

RTKNAVI

RTKNAVI stellt das Echtzeit-Positionierungsmodul dar. Je nach Anforderung kann eine Positionierungsmethode (Einzelpunkt, PPP oder RTK) mit diversen Einstellungsmöglichkeiten gewählt werden. Es ist möglich, verschiedenste Eingangs- und Ausgangsquellen für die Aufbereitung der Daten zu verwenden.

RTKCONV

Dieses Modul stellt die Konvertierungssoftware da. Alle implementierten, rohdatenfähigen Empfänger können so in die verschiedenen RINEX Files konvertiert werden. Unterstützt werden alle GNSS-Konstellationen und RINEX-Formate. Außerdem können auch RTCM-Streams in das RINEX Format konvertiert werden.

RTKPOST

RTKPOST ist die Gegenkomponente zu RTKNAVI und wird für das Auswerten im Postprocessing verwendet. Unterstützt wird als Inputformat, im Gegensatz zu RTKNAVI, nur das RINEX Format. Sonstige Auswerte-Einstellungen gleichen dem Echtzeit-Modul RTKNAVI.

RTKPLOT

RTKLIB stellt auch eine eigene Visualisierungssoftware RTKPLOT zur Verfügung und unterstützt Google Earth³ als Visualisierungskomponente. Positionslösungen können so visualisiert und anhand von statistischen Parametern analysiert werden.

³ Google Earth, (c) Google Inc.

STRSVR

Dieses Modul STRSVR dient zur Anbindung von Empfängern und Datenstreams über diverse Schnittstellen wie z.B. (i) einer seriellen Schnittstelle, (ii) eines TCP/IP-Streams oder (iii) von NTRIP Daten. Das Modul dient somit auch zum Loggen der Rohdaten einzelner Empfänger und Datenstreams.

NTRIP Browser

Der NTRIP Browser erlaubt es, Verbindung zu allen bekannten NTRIP Caster herzustellen und wenn möglich, diese Daten zu streamen.

RTKGET

Für diverse GNSS-Anwendungen wie PPP werden zusätzliche Daten für eine Prozessierung benötigt, welche zumeist aus diversen Quellen über das Internet bezogen werden müssen. RTKGET erlaubt einen Verbindungsaufbau zu diesen Quellen und stellt die benötigten Daten wie (i) präzise Uhrenparameter, (ii) Orbitparameter und (iii) Ionosphärenmodelle zum Download bereit.

6.3 RTK-Definition

Bei der Positionierungsmethode RTK handelt es sich um eine relative Positionierung im Echtzeit-Modus. Somit werden in Echtzeit die Rohdaten von einer Basisstation und eines Rovers kombiniert und prozessiert. Für die Echtzeitfähigkeit bedarf es spezieller Algorithmen, da z.B. das Lösen der Phasenmehrdeutigkeit sehr rechenaufwändig ist. Hierzu werden spezielle on-the-fly Methoden eingesetzt, siehe Koch (2010). Grundlegend wird noch zwischen einem aktiven und einem passiven RTK-Modus unterschieden. Wobei bei einem aktiven System die Prozessierung und Positionsbestimmung am Rover stattfindet, was für Navigationsanwendung von Bedeutung ist. Bei einem passiven RTK-System findet die Prozessierung und Positionsbestimmung jedoch an der Basis oder einem externen Server statt, was vor allem für Monitoring-Zwecke ihren Einsatz findet, siehe Koch et al., (2015).

Nachfolgend wird der RTK-Algorithmus nach Vorlage der RTKLIB-Prozessierung Takasu (2013) beschrieben.

6.4 RTK-Algorithmus

Das Kapitel beschreibt die grundlegende Vorgangsweise der RTK-Prozessierung, wie sie in der Software RTKLIB ihre Anwendung findet.

Die Verwendung der Differenzbildungen wie SD, DD oder TD der einzelnen Messungen, wie sie bei RTK vorkommen, sind im Abschnitt 5.3 – 5.5 dargestellt. RTKLIB verwendet für den kinematischen als auch statischen RTK-Modus einen Extended Kalman Filter. Zum leichteren Verständnis der RTKLIB-Codeimplementierung wird die Nomenklatur des EKF von Takasu (2013) übernommen.

6.4.1 EKF Formulierung

Mit der Verwendung eines EKF kann ein Zustandsvektor \mathbf{x} für die unbekannt Parameter mit dessen Kovarianzmatrix \mathbf{P} in Abhängigkeit des Beobachtungsvektors \mathbf{y} beschreiben werden. Mit der Formulierung

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\tilde{\mathbf{x}}_k)),$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}(\tilde{\mathbf{x}}_k)) \tilde{\mathbf{P}}_k, \quad 6-1$$

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}^T(\tilde{\mathbf{x}}_k) (\mathbf{H}(\tilde{\mathbf{x}}_k) \tilde{\mathbf{P}}_k \mathbf{H}^T(\tilde{\mathbf{x}}_k) + \mathbf{R}_k)^{-1},$$

worin $\hat{\mathbf{x}}_k$ und \mathbf{P}_k den geschätzten Parametervektor oder auch Zustandsvektor mit Kovarianzmatrix zum Zeitpunkt t_k beschreiben. $\mathbf{h}(x)$, $\mathbf{H}(x)$ und \mathbf{R}_k beschreiben den Beobachtungsmodellvektor, die Matrix mit den partiellen Ableitungen und die Kovarianzmatrix der Messfehler. Unter der Annahme eines linearen Systemmodells kann das Timeupdate wie folgt definiert werden,

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{F}_k \hat{\mathbf{x}}_k$$

6-2

$$\tilde{\mathbf{P}}_{k+1} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_k,$$

worin F_k und Q_k die Übertragungsmatrix mit deren Kovarianzmatrix des Systemrauschens zum Zeitpunkt t_k zu t_{k+1} beschreiben.

6.4.2 DD-Ansatz

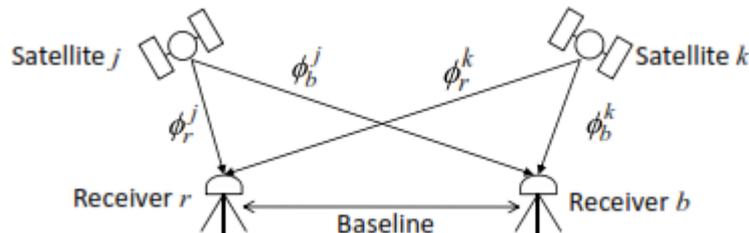


Abbildung 6.2: DD-Prinzip Takasu (2013)

Das Prinzip des DD Ansatzes ist in Abbildung 6.2 dargestellt, um den nachfolgenden Algorithmus besser beschreiben zu können. SD müssen immer vor DD erfolgen und sollen für die gleiche Messepoche für Rover und Basis durchgeführt werden. Da aber die Empfänger nie zum exakt gleichen Zeitpunkt synchronisiert werden, wählt RTKLIB immer die am nächsten liegende Basismessung zur Berechnung der SD bzw. DD aus. Der Zeitunterschied zwischen den Messungen wird als Age of Differential bezeichnet und sollte im idealen Fall bei 0 s liegen. Wird diese Differenz aufgrund von Übertragungsverzögerungen zu groß, verschlechtert sich die Positionslösung aufgrund von unterschiedlichen Uhrparametern wie dem Satelliten Uhrendrift. Für die grundlegende Bestimmung eines Referenzsatelliten wird der am höchsten stehende Satellit zu jeder Epoche herangezogen. Für die Prozessierung muss angemerkt werden, dass keine SD zwischen GPS und GLONASS Satelliten berechnet werden. Dies ist auf die unterschiedlichen Laufzeitverzögerungen der Systeme zurückzuführen.

Der unbekannte Parametervektor

$$\mathbf{x} = (r_r^T, B_{L1}^T)^T, \quad 6-3$$

definiert die zu bestimmenden Parameter, wobei B die SD GPS/GLONASS-Ambiguitäten (in Cycles pro Frequenzband) beschreibt und r_r , v definieren die Position bzw. die Geschwindigkeit.

Der Beobachtungsvektor \mathbf{y} wird durch DD-Phasen und Codemessungen beschrieben.

$$\mathbf{y} = (\Phi_1^T, R_1^T)^T, \quad 6-4$$

wobei Φ_1^T, R_1^T die Einfrequenz Phasen- bzw. die Codemessungen zu den Satelliten beschreiben.

$$\begin{aligned} \Phi_1 &= (\Phi_{rb}^{12}, \Phi_{rb}^{13}, \Phi_{rb}^{14}, \dots, \Phi_{rb}^{lm})^T, \\ R_1 &= (R_{rb}^{12}, R_{rb}^{13}, R_{rb}^{14}, \dots, R_{rb}^{lm})^T, \end{aligned} \quad 6-5$$

beschreiben nun den Aufbau des DD-Vektors im L1 Band. Die Indizes r,b stehen hierbei für Rover, Basis und die hochgestellten Indizes definieren die SD zwischen den einzelnen Satelliten. Somit ergeben sich bei einer Basislinie bis rund 10 km folgende Beobachtungsgleichungen. Satelliten- und Empfängeruhrenfehler, sowie atmosphärische Effekte, werden durch die DD-Technik eliminiert. Die Formeln

$$\begin{aligned} \Phi_{rb}^{ij} &= \rho_{rb}^{ij} + \lambda(B_{rb}^i - B_{rb}^j) + \varepsilon_\Phi, \\ R_{rb}^{ij} &= \rho_{rb}^{ij} + \varepsilon_R, \end{aligned} \quad 6-6$$

definieren nun den Input für den Beobachtungsmodellvektor $\mathbf{h}(\mathbf{x})$.

$$\begin{aligned} \mathbf{h}(\hat{\mathbf{x}}) &= (\mathbf{h}_{\Phi,L1}^T, \mathbf{h}_{R,L1}^T)^T, \\ \mathbf{h}_{\Phi,L1} &= \begin{pmatrix} \rho_{rb}^{12} + \lambda(B_{rb}^1 - B_{rb}^2) \\ \rho_{rb}^{13} + \lambda(B_{rb}^1 - B_{rb}^3) \\ \vdots \\ \rho_{rb}^{1m} + \lambda(B_{rb}^1 - B_{rb}^m) \end{pmatrix}, \quad \mathbf{h}_{R,L1} = \begin{pmatrix} \rho_{rb}^{12} \\ \rho_{rb}^{13} \\ \vdots \\ \rho_{rb}^{1m} \end{pmatrix} \end{aligned} \quad 6-7$$

Somit ergibt sich die Designmatrix $\mathbf{H}(\mathbf{x})$ wie folgt

$$\mathbf{H}(\hat{\mathbf{x}}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{pmatrix} -\mathbf{DE} & \mathbf{0} & \lambda \mathbf{D} \\ -\mathbf{DE} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad 6-8$$

wobei sich ρ_r^i und ρ_b^i als

$$\rho_r^i = \|\widehat{\mathbf{r}}_r - \mathbf{r}^i\|, \quad \rho_b^i = \|\widehat{\mathbf{r}}_b - \mathbf{r}^i\|, \quad 6-9$$

im ECEF-System beschreiben lassen.

Die Matrizen \mathbf{D} und \mathbf{E} beschreiben die SD-Matrix bzw. \mathbf{E} beschreibt die Matrix mit den e_r^i line-of-sight Vektoren von Rover zu dem Satelliten i

$$\mathbf{D} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 1 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & -1 \end{pmatrix}, \quad 6-10$$

$$\mathbf{E} = (e_r^1, e_r^2, \dots, e_r^m)^T$$

Die in Formel 6.1 definierte Kovarianzmatrix der Messfehler setzt sich aus

$$\mathbf{R}_k = \begin{pmatrix} \mathbf{D}\mathbf{R}_{\phi,L1}\mathbf{D}^T & \\ & \mathbf{D}\mathbf{R}_{R,L1}\mathbf{D}^T \end{pmatrix}$$

$$\mathbf{R}_{\phi,L1} = 2\text{diag}(\sigma_{\phi}^{1^2}, \sigma_{\phi}^{2^2}, \dots, \sigma_{\phi}^{m^2}) \quad 6-11$$

$$\mathbf{R}_{R,L1} = 2\text{diag}(\sigma_R^{1^2}, \sigma_R^{2^2}, \dots, \sigma_R^{m^2})$$

zusammen. Zustände kommen diese durch eine Varianzfortpflanzung, siehe (Koch, 2012). Die Gewichtung der Beobachtungen kann Abschnitt 7.3.3 entnommen werden.

Die in Formel 6.2 dargestellte Übertragungsmatrix \mathbf{F}_k und die Kovarianzmatrix des Systemrauschens \mathbf{Q}_k setzen sich für den kinematischen Positionierungsmodus in RTKLIB aus

$$\mathbf{F}_k = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & \\ & \mathbf{I}_{3 \times 3} & \\ & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix}, \quad 6-12$$

$$\mathbf{Q}_k = \begin{pmatrix} \infty_{3 \times 3} & & \\ & \mathbf{0}_{3 \times 3} & \\ & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix},$$

zusammen. Somit setzt RTKLIB, unter Berücksichtigung der numerischen Stabilität ($\infty_{3 \times 3}$ wird durch hohes Rauschen ersetzt), die Zustände bei jeder Epoche zurück auf den Initialisierungswert. Der Initialisierungswert wiederum ergibt sich aus einer SPP-Lösung, siehe Takasu et al., (2005). Im statischen Positionierungsmodus verwendet RTKLIB eine vereinfachte $\mathbf{F}_k = \mathbf{I}$ und $\mathbf{Q}_k = \mathbf{0}$ Methode.

Durch Lösen der EKF-Formel 6.1, erreicht man die Lösung der geschätzten Roverposition und der Float-Ambiguitäten. Um eine fixierte Lösung zu erhalten, wird auf Abschnitt 6.4.3 verwiesen.

6.4.3 Beobachtungsgewichtung

RTKLIB verwendet für die Gewichtung der einzelnen Beobachtungen ein elevationsbedingtes Gewichtungsschema. Die in Formel 6.11 dargestellten Standardabweichungen σ_{ϕ}^i und σ_R^i , der einzelnen Messungen ergeben sich somit aus

$$\sigma^2 = F^s R_r \left(a_{\sigma}^2 + \frac{b_{\sigma}^2}{\sin^2(EL_r^s)} \right) + \sigma_{eph}^2 + \sigma_{ion}^2 + \sigma_{trop}^2 + \sigma_{bias}^2, \quad 6-13$$

mit folgenden Parametern:

F^s	...	Satellitensystemfaktor (1:GPS+GAL, 1.5:GLO)
R_r	...	Code/Phase Fehlerratio (Phase/Code=300)
$a_{\sigma}^2, b_{\sigma}^2$...	Phasenfehler a und b in Meter (0.003)
σ_{eph}	...	Standardabweichung der Ephemeriden
σ_{ion}	...	Standardabweichung des Ionosphärenmodells
σ_{trop}	...	Standardabweichung des Troposphärenmodells
σ_{bias}	...	Standardabweichung des Code Bias
EL_r^s	...	Elevation [°] aus GOBS Format

Somit ergibt sich eine elevationsbedingte Gewichtung, welche grundlegend unterschieden wird nach (i) Art der Beobachtung und (ii) Art des GNSS-Systems. Es ergibt sich durch Formel 6.13 eine 300-fach höhere Gewichtung einer Phasenmessung im Vergleich zu einer Codemessung. Angebracht wird die Gewichtung durch die SD-Matrix auf die SD-Beobachtungen und in weiterer Folge durch eine DD-Matrix auf die DD-Beobachtungen, siehe Takasu (2013).

6.4.4 Integer Ambiguitätenlösung

Wie bereits in Kapitel 5.2 erwähnt, sind Phasenmessungen nicht eindeutig, müssen jedoch aus ganzzahligen Integer-Werten bestehen. Diese Integer-Eigenschaft geht jedoch bei der Schätzung des Basislinienvektors, siehe Formeln 6-1 bis 6-12, verloren. Die Wiederherstellung dieser Integer-Eigenschaft ist jedoch nicht trivial lösbar und fordert vor allem in der Echtzeitauswertung spezielle on-the-fly Algorithmen. Die im Zuge der RTKLIB Prozessierung verwendete Methode LAMBDA, nach Tiberius et al., (1996), wird nachfolgend kurz beschrieben. Um einen Überblick über die einzelnen Lösungsansätze zu bekommen, wird auf Hofmann-Wellenhof et al., (2008) verwiesen.

LAMBDA-Methode

Nach Takasu et al., (2005) und Tiberius et al., (1997) ist die LAMBDA Methode der effektivste Algorithmus, um die Integer-Eigenschaft der Ambiguitäten zu reproduzieren. In RTKLIB werden zuerst die Float-Solution aus Abschnitt 6.4.2, sowie die Kovarianzmatrix zu einem DD-Format konvertiert, vgl. Formel 6-3. Die Nomenklatur wird laut Takasu, et al., (2005) übernommen.

$$\hat{\mathbf{x}}'_k = \mathbf{G} \hat{\mathbf{x}}'_k = (\hat{\mathbf{r}}_r^T, \hat{\mathbf{N}}_r^T)^T,$$

$$\mathbf{P}'_k = \mathbf{G} \mathbf{P}_k \mathbf{G}^T = \begin{pmatrix} \mathbf{Q}_R & \mathbf{Q}_{NR} \\ \mathbf{Q}_{RN} & \mathbf{Q}_N \end{pmatrix}, \quad 6-14$$

$$\mathbf{G} = \begin{pmatrix} \mathbf{I}_3 & & \\ & \mathbf{D} & \\ & & \mathbf{D} \end{pmatrix},$$

worin $\hat{\mathbf{N}}_r$ die zu bestimmenden DD-Ambiguitäten beschreiben. \mathbf{Q}_N beschreibt die zugehörige Kovarianzmatrix der Integer Ambiguitäten. Um nun den Integer Ambiguitätenvektor bestimmen zu können, muss ein ILS-Problem nach

$$\tilde{\mathbf{N}} = \underset{N \in \mathbb{Z}}{\operatorname{argmin}}((\mathbf{N} - \hat{\mathbf{N}})^T \mathbf{Q}_N^{-1} (\mathbf{N} - \hat{\mathbf{N}})), \quad 6-15$$

gelöst werden. Als Lösungsansatz für dieses Problem wird die in RTKLIB implementierte LAMBDA Methode verwendet. Nach Tiberius et al., (1996) und

Teunissen et al., (1996) verwendet die LAMBDA-Methode eine verbesserte Methode, um den Suchraum im Ambiguitätenraum einzuschränken. Eine detaillierte Beschreibung der LAMBDA-Methode und die verwendete Methodik kann Teunissen et al., (1996) entnommen werden. Die Lösung dieser Methode wird anschließend mit einem Ratio-Test in RTKLIB validiert. Der Ratio-Test ergibt sich aus dem Verhältnis der gewichteten Residuen der erstbesten \check{N} zur zweitbesten \check{N}_2 Lösung. Der Threshold Wert R_{Thres} , welcher den Grenzwert zur Annahme einer Fixierung darstellt, stellt eine Einstellungsmöglichkeit in RTKLIB dar.

$$R = \frac{(\check{N}_2 - \hat{N})^T Q_N^{-1} (\check{N}_2 - \hat{N})}{(\check{N} - \hat{N})^T Q_N^{-1} (\check{N} - \hat{N})} > R_{Thres}, \quad 6-16$$

beschreibt die oben angeführte Validierung der Fixierung. Wird dieser angenommen, erhält man die fixierte Positionslösung nach

$$\begin{pmatrix} \check{r}_r \\ \check{v}_r \end{pmatrix} = \begin{pmatrix} \hat{r}_r \\ \hat{v}_r \end{pmatrix} - Q_{RN} Q_N^{-1} (\hat{N} - \check{N}). \quad 6-17$$

Sollte der Ratio-Test abgelehnt werden, liefert RTKLIB die Float-Lösung als Positionslösung.

Kapitel 7

eRTK

7.1 Grundlagen

Das Kapitel 7 beschreibt nun die eigentliche Entwicklung dieser Arbeit. Das in Kapitel 3.5 beschriebene Rohdatenformat GOBS soll nun mittels einer eigens entwickelten Systemarchitektur in die Software RTKLIB eingebunden und für die RTK-Prozessierung herangezogen werden. In weitere Folge soll das dynamische Modell der RTK-Prozessierung verbessert werden, sowie unterschiedliche Beobachtungsgewichtungen sollen anhand der GOBS-Daten implementiert werden. Diese weiterentwickelte Software von RTKLIB wird folgend als eRTK bezeichnet.

7.2 Systemarchitektur eRTK

Die eigens entwickelte Systemarchitektur, bestehend aus einer UMTS-Datenübertragung, des entwickelten proprietären GNSS-Rohdatenformates GOBS, einer Server-Caster-Client Struktur mit NTRIP-Funktionsweise und einer weiterentwickelten RTKLIB-Prozessierung eRTK, ist in Abbildung 7.1 dargestellt.

Die Standard-Systemarchitektur nach RTKLIB, siehe Abbildung 7.2, zeigt die RTK-Systemarchitektur, wie sie in RTKLIB implementiert ist. Hierbei werden jedoch nur standardisierte Formate wie RTCM für die Prozessierung herangezogen.

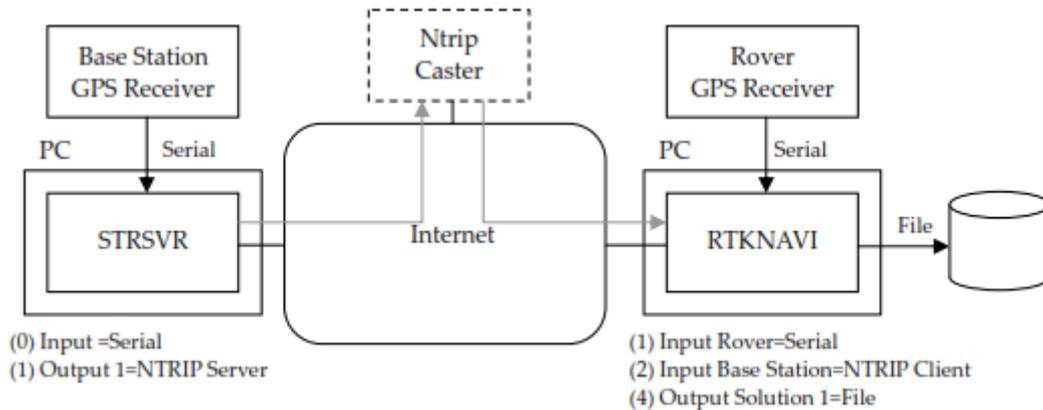


Abbildung 7.2: RTKLIB-Systemarchitektur Takasu (2013)

Die entwickelte Software eRTK, welche eine Weiterentwicklung des RTKLIB Moduls RTKNAVI darstellt, ist für eine Echtzeit-RTK Prozessierung mit dem entwickelten GOBS-Format konzipiert.

Um mit eRTK eine RTK-Prozessierung durchführen zu können, sind ein Basis- und ein Rover-Empfänger, ausgestattet mit jeweils einem u-blox M8T, sowie eine eigene Datenübertragung erforderlich. Das Softwarepaket eRTK, siehe Abbildung 7.3, liest die GOBS-Daten der Empfänger in Echtzeit ein undprozessiert, wenn möglich, eine fixierte RTK-Lösung in Echtzeit.

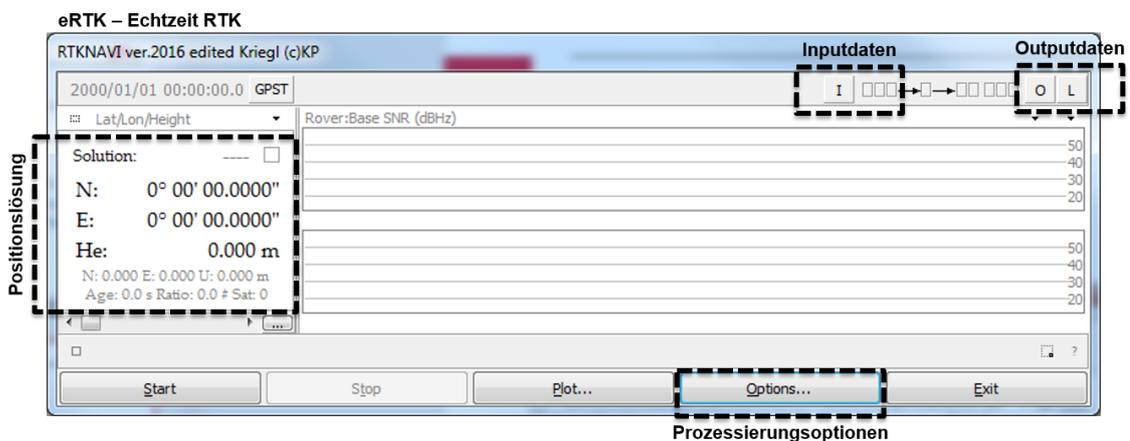


Abbildung 7.3: eRTK Menü

Für eine detaillierte Übersicht der Optionsmöglichkeiten und der grundlegenden Funktionsweise von eRTK bzw. RTKNAVI wird auf Takasu (2013) verwiesen.

7.3 eRTK-RTKNAVI

Die grundlegende Weiterentwicklung legt den Fokus auf

- das Einbinden aller Rohdaten (GOBS),
- eine verbesserte Gewichtung der GNSS-Messungen,
- eine verbesserte Cycle Slip Detektion sowie
- eine verbesserte Initialisierung.

Die eigentliche RTK-Prozessierung findet mit dem RTKLIB Algorithmus, siehe Abschnitt 6.4, statt. Abbildung 7.4 zeigt das Einbinden des GOBS Formates, welches in Abschnitt 3.5 definiert wurde, in die Software. Anstelle der standardisierten RTCM-Daten werden hierbei alle empfängerspezifischen Daten für die Prozessierung herangezogen. Die Datenübertragung erfolgt mittels UMTS/GSM-Datenstick im Mobilnetz von HoT.

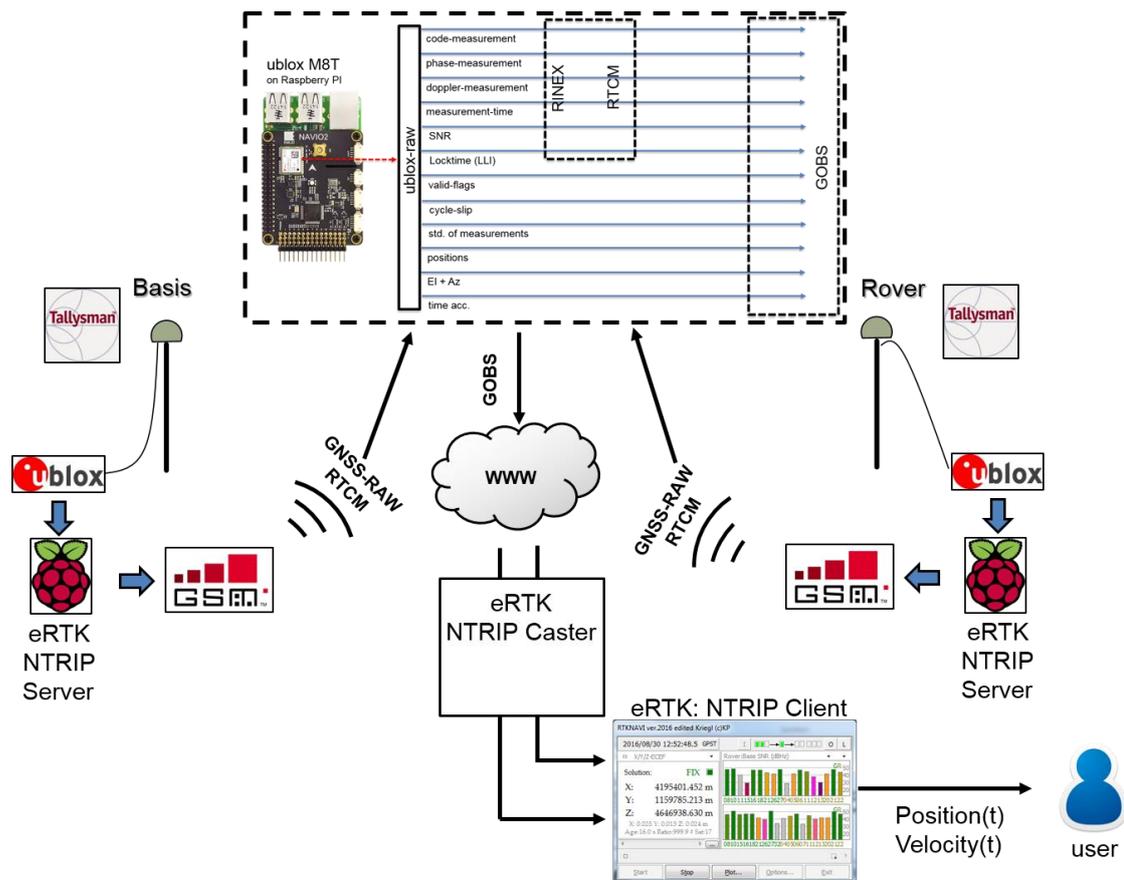


Abbildung 7.4: GOBS-eRTK

Um ein RTKLIB-Modul mit dem zugehörigen frei zugänglichen und erweiterbaren Source Code von RTKLIB kompilieren zu können, wird ein RAD Studio 10.1 Berlin laut embarcadero (2016) mit zugehörigem Embarcadero C/C++ Compiler, benötigt.

7.3.1 GOBS in eRTK

Das in Abschnitt 3.5 und Anhang A definierte Format GOBS implementiert alle möglichen empfängerspezifischen Daten des u-blox M8T. eRTK ermöglicht es, alle GOBS-Parameter für eine verbesserte RTK-Prozessierung heranzuziehen, sowie zu übertragen und mitzuloggen. Um einen mitgeloggtten Datensatz in das GOBS-Format konvertieren zu können, wurde das Konvertierungsmodul RTKCONV von RTKLIB entsprechend adaptiert. Mit dieser Lösung, dargestellt in Abbildung 7.5, ist es nun möglich, Datensätze nicht nur in das RINEX-Format zu konvertieren, sondern eine Konvertierung in das GOBS-Format für Postprocessing Analysen ebenso unterstützen. Eine Prozessierung im Postprocessing, bringt vor allem bei der Cycle Slip Detektion und Ambiguitätenlösung große Vorteile mit sich. So lassen sich etwa durch eine Vorfilterung oder Rückwärtsprozessierung zusätzliche Informationen über mögliche Cycle Slips gewinnen.

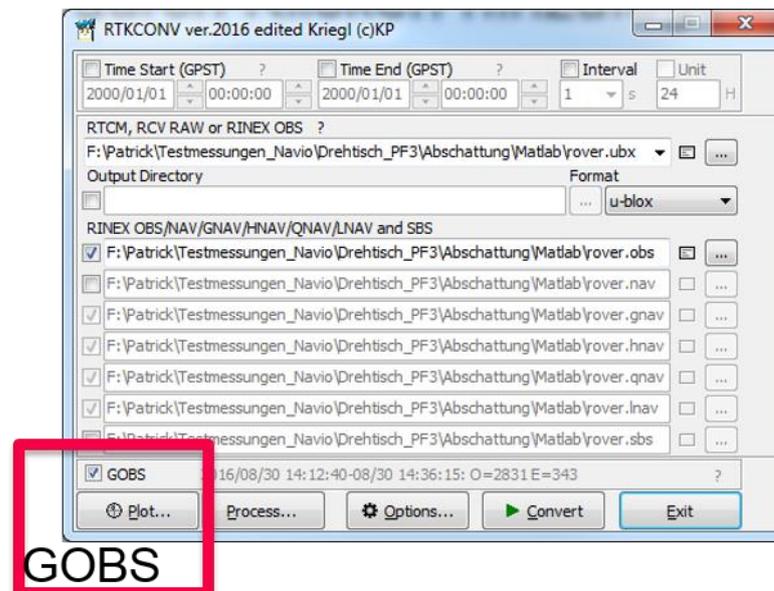


Abbildung 7.5: RTKCONV-GOBS

Vor allem bei low-cost Empfängern ist das Tracking eines Satelliten sensitiv, da ein häufiges Auftreten von Cycle Slips sehr wahrscheinlich ist. Somit zeigt es sich als schwierig, alle Cycle Slips mit herkömmlichen Methoden zu detektieren und zu reparieren. Die herkömmliche RTKLIB Version verwendet nach Takasu et al., (2005) den standardisierten LLI Indikator, um mögliche Cycle Slips im Einfrequenzbereich zu detektieren. Aufgrund der high-sensitiven Eigenschaft eines low-cost Empfängers, ist eine stabile Cycle Slip Detektion nach dieser Methode schwer realisierbar. Das GOBS Format berechnet den LLI aufgrund mehrerer Qualitätsparameter des Signals, wie (i) der Locktime des Signals (Zeit, seitdem ein Lock des Satelliten erzielt wurde), (ii) der Standardabweichung der Phasemessung und (iii) Valid-Flags der einzelnen Messungen, neu. Mit dieser verbesserten LLI Zusammenstellung kann eine stabilere Cycle Slip Detektion durchgeführt werden. Für die Definition eines LLI siehe Abschnitt 3.5.

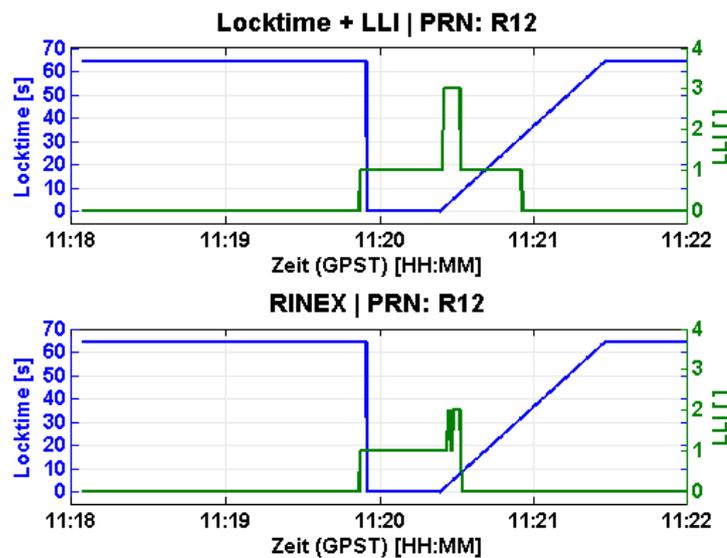


Abbildung 7.6: GOBS LLI - RINEX LLI

Abbildung 7.6 zeigt den Unterschied zwischen dem GOBS-LLI des GLONASS R12 Satelliten, und dem LLI Wert aus üblichen Formaten wie RINEX oder RTCM (untere Grafik). Dieser Satellit fällt kurz vor 11:20 Uhr aus dem Lock (Locktime = 0) und kann nach etwa 30 Sekunden wieder getrackt werden.. Zu sehen ist, dass der obige GOBS-LLI deutlich länger einen Wert größer 0 aufzeigt (Zeitraum 11:20 bis 11:21). Somit wird bei einer RINEX- oder RTCM-Prozessierung dieser Satellit zur Berechnung hinzugezogen, obwohl noch etwaige Cycle

Slips die Messung beeinflussen können (LLI = 0). Unter Verwendung des GOBS Formates wird dieser Satellit nicht zur Prozessierung hinzugezogen. Des Weiteren können Cycle Slips entsprechend erkannt und repariert werden.

7.3.2 eRTK – EKF

Der in Abschnitt 6.4 beschriebene Extended Kalman Filter ist bereits in RTKLIB realisiert, bietet jedoch begrenzte Möglichkeiten das dynamische Modell des Empfängers anzupassen. So wird für die übliche RTKLIB Lösung nur zwischen einem statischen und dynamischen System unterschieden. eRTK ermöglicht es, durch Einbindung eines adaptierten EKF (zwei dynamische Modelle), eine verbesserte RTK-Lösung, sowie eine schnellere Fixierung der Ambiguitäten zu erhalten. Dazu wurde folgendes Modell implementiert:

$$\mathbf{F}_k = \begin{pmatrix} \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \tau_r & \\ & \mathbf{I}_{3 \times 3} & \\ & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix}, \quad 7-1$$

$$\mathbf{Q}_k = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & \\ & \mathbf{Q}_v & \\ & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix},$$

worin

$$\mathbf{Q}_v = \mathbf{E}_r^T \text{diag}(\sigma_{ve}^2 \tau_r, \sigma_{vn}^2 \tau_r, \sigma_{vu}^2 \tau_r) \mathbf{E}_r \quad 7-2$$

und $\tau_r = t_{k+1} - t_k$ definiert sind. σ_v beschreibt die Standardabweichungen der Geschwindigkeit im lokalen Horizontsystem (north-east-up: $\sigma_{vn} - \sigma_{ve} - \sigma_{vu}$), siehe Abschnitt 5.6.3. Somit wird die Übertragungsmatrix \mathbf{F}_k und die Kovarianzmatrix des Systemrauschens \mathbf{Q}_k , im Gegensatz zum üblichen Modell, adaptiert, die restliche Modellierung erfolgt analog zu Abschnitt 6.4. Da im Bereich der RTK-Prozessierung zumeist statisch initialisiert wird und erst nach einer Fixierung eine Kinematik einsetzt, wurde in eRTK ein Static-Start Modus implementiert. RTKLIB erlaubt es, nur zwischen statischen und kinematischen Modus zu wählen, was diverse Einschränkungen mit sich bringt. Mit dem neu implementierten Static-

Start Modus kann nun eine schnellere Fixierung erzielt werden. Das Feature wurde laut Formeln 7-3 und 7-4 implementiert.

Statischer Modus solange eine Float-Lösung erzielt wird:

$$\mathbf{F}_k = \begin{pmatrix} \mathbf{I}_{3 \times 3} & & \\ & \mathbf{I}_{3 \times 3} & \\ & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix},$$

7-3

$$\mathbf{Q}_k = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & \\ & \mathbf{0}_{3 \times 3} & \\ & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix},$$

Umschalten des Modells nach erfolgreicher Fixierung auf

$$\mathbf{F}_k = \begin{pmatrix} \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \tau_r & \\ & \mathbf{I}_{3 \times 3} & \\ & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix},$$

7-4

$$\mathbf{Q}_k^{k+1} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & \\ & \mathbf{Q}_v & \\ & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix}.$$

Die dritte Spalte ist aufgrund der zeitinvarianten Ambiguitäten konstant, d.h. hier wird kein Rauschen modelliert. Durch diese Implementierung ergibt sich eine verbesserte Startphase im statischen Bereich, indem eine schnellere Fixierung erzielt werden kann. Die verbesserte Standardabweichung, aufgrund des dynamischen Modells, im Beispiel der Höhenkomponente, ist in Abbildung 7.7 dargestellt. Es ist ersichtlich, dass eine schnellere Fixierung (Standardabweichung der Höhe, s_{du} , geht gegen 0), sowie eine geringere Standardabweichung der Höhe mit dem Static-Start Modus erzielt werden kann.

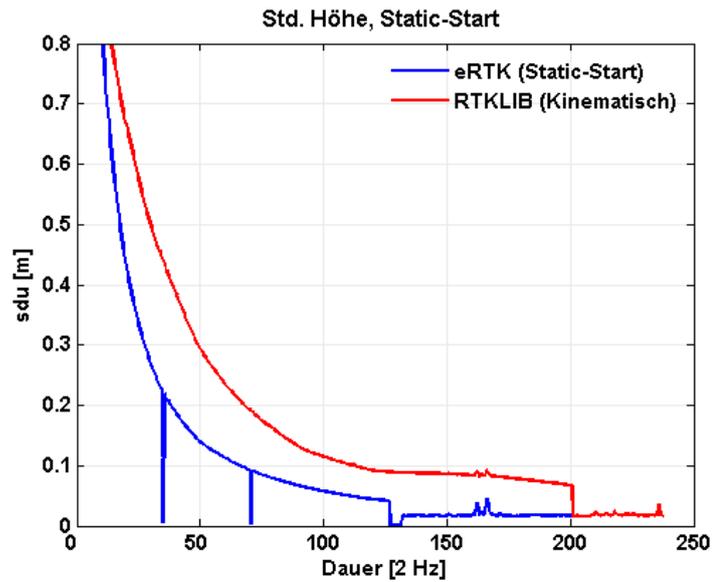


Abbildung 7.7: Static-Start – sdu

Der Static-Start Modus erlaubt es außerdem, eine stabilere Fixierung anzuwenden. Durch die geringeren Standardabweichungen kann eine falsche Fixierung nahezu ausgeschlossen werden. Der Modus wurde entsprechend der RTKNAVI Menüführung in den Optionen als eigener Positionierungsmodus eingebracht, siehe Abbildung 7.8.

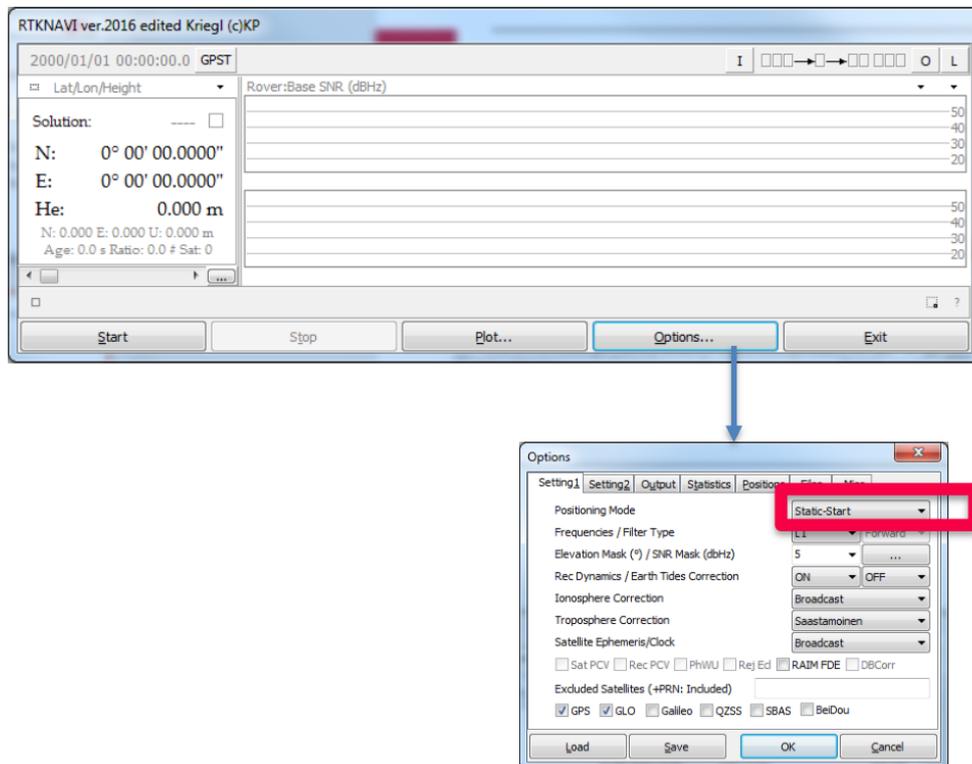


Abbildung 7.8: Static-Start Feature

7.3.3 eRTK – Gewichtungen

Neben der in Abschnitt 6.4.3 definierten elevationsbedingten Gewichtung wurden in eRTK mit Hilfe des GOBS-Formates, folgende, zusätzliche Gewichtungsoptionen implementiert: (i) eine SNR+elevationsbedingte Gewichtung, (ii) eine reine SNR-Gewichtung und (iii) eine abgeänderte SNR+elevationsbedingte Gewichtung mit den im GOBS-Format vorhandenen Standardabweichungsparametern. Die Implementierung und Formulierung wird nachfolgend dargestellt und beschrieben.

SNR Gewichtung

Nach Wieser (2001) eignet sich eine SNR-basierte Gewichtung für Phasenbeobachtungen besser als ein elevationbedingtes Modell. Nach Zidan (2006) liegt der Grund darin, dass sich die Standardabweichung der eigentlichen Phasenmessung des PLL eines GNSS-Empfängers aus einer Funktion von C/N0 und der Bandbreite der Trägerfrequenz zusammensetzt. Das nach Wieser (2001) entwickelte SIGMA- ϵ Modell beschreibt die Gewichtung der Phasenmessung

$$\sigma_{\phi}^2 = V_i + C_i \cdot 10^{-\frac{SNR}{10}} \quad 7-5$$

mit

σ_{ϕ}^2	...	Varianz der undifferenzierten Phasenmessung [m ²]
V_i, C_i	...	Modellparameter, antennen-empfängerspezifisch
i	...	Antennen / Empfängertyp
SNR	...	gemessenes SNR aus GOBS [dBHz]

als Parameter.

Für eine optimierte Modellparameterauswahl müssen laut Wieser (2001) die Parameter auf die DD-Residuen angepasst werden. Für diese Arbeit wurden leicht geänderte Werte nach Realini (2009) als Modellparameter angenommen.

Elevation + SNR Gewichtung

Da low-cost GNSS-Empfänger jedoch high-sensitivity Eigenschaften besitzen, kann eine reine SNR Gewichtung zu verfälschten Gewichtungen der Phasenmessung führen. Realini (2009) zeigt, dass eine Kombination von Elevation- und SNR-Informationen zu einer optimalen Gewichtung von Code- bzw. Phasenmessung im low-cost Bereich führen. Dies gelang durch Auswertung der Codemessungsfehler (RMSE) als Funktion, abhängig von der Elevation und des SNR. Realini (2009) zeigt auch, dass bei steigender Elevation eine Senkung des RMSE auftritt, was dem Prinzip einer Gewichtung nach Elevation entspricht. Durch Kombination mit der SNR-Gewichtung werden jedoch Satelliten, welche unter möglichem Multipath oder sonstigen Abschattungen liegen, geringer gewichtet als bei einer reinen Elevationsgewichtung. Das Modell wurde jedoch im Hinblick auf statische bzw. gering-kinematische Anwendungen implementiert und sollte für hochkinematische (>10 km/h) nicht ohne weitere Untersuchungen verwendet werden. Somit ergibt sich das nach Realini (2009) entwickelte Gewichtungsmodell

$$W(SNR, e) = \begin{cases} \frac{1}{\sin^2 e} \left(10^{-\frac{(SNR-s1)}{a}} \left(\left(\frac{A}{10^{-\frac{(s0-s1)}{a}}} - 1 \right) \frac{SNR - s1}{s0 - s1} + 1 \right) \right) & \dots SNR < s1 \\ 1 & \dots SNR \geq s1 \end{cases} \quad 7-6$$

mit den Parametern

W	...	Varianz der einzelnen Beobachtung
SNR	...	Elevation des zu gewichtenden Satelliten
e	...	Elevation des zu gewichtenden Satelliten
$s1$...	SNR Oberer-Threshold [50 dBHz]
$s0$...	SNR Unterer-Threshold [10 dBHz]
a, A	...	statistische RMSE Parameter nach Realini (2009).

Die inverse Gewichtungsfunktion ist in nachfolgender Abbildung 7.9 dargestellt und zeigt das Verhalten von Elevation zu SNR. Die Ergebnisse der unterschiedlichen Gewichtungen sind Kapitel 8 zu entnehmen.

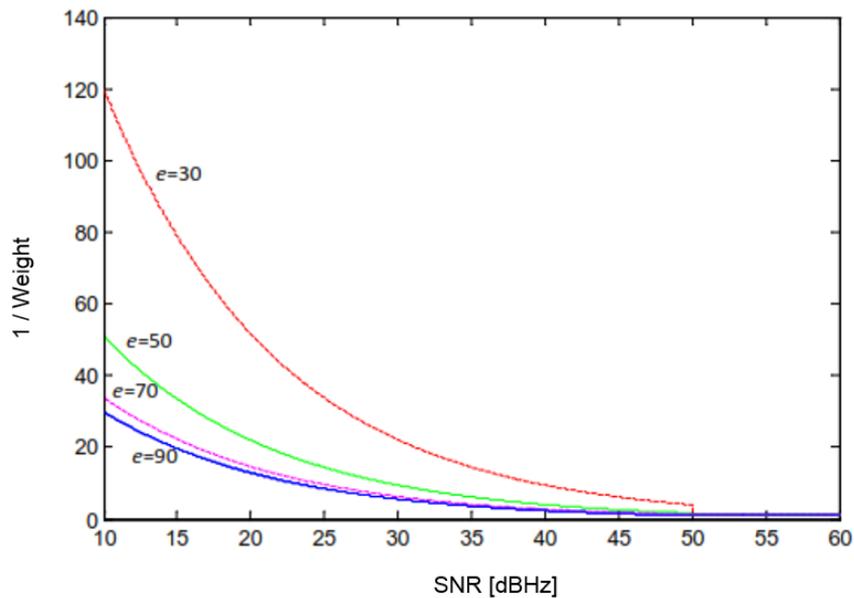


Abbildung 7.9: Gewichtung Realini, (2009)

Der Sprung der roten Linie ($e=30$) auf null, bei $\text{SNR}=50$, lässt sich durch den Threshold $s_1=50$ dBHz erklären.

7.3.4 Cycle Slip Detektion

Wie bereits in Abschnitt 3.5 und 7.3.1 beschrieben, verwendet RTKLIB den LLI im Einfrequenzbereich, um mögliche Cycle Slips zu detektieren. Da aber vor allem im low-cost Bereich kaum stabile LLI-Indikatoren zur Verfügung stehen, wird im Zuge der eRTK Entwicklung das GOBS-Format für eine Cycle Slip Detektion herangezogen und entsprechende Satelliten mit möglichen Cycle Slip werden ausgeschlossen. Grundlegend tritt ein Cycle Slip auf, wenn eine Unterbrechung der Phasenmessung oder ein sehr geringes SNR vorhanden ist, siehe Abbildung 7.10. Für eine stabile Positionslösung ist es essentiell, mögliche Cycle Slip zu detektieren und betroffene Satelliten auszuschließen, was die einfachste Methode für Realtime-Anwendungen darstellt.

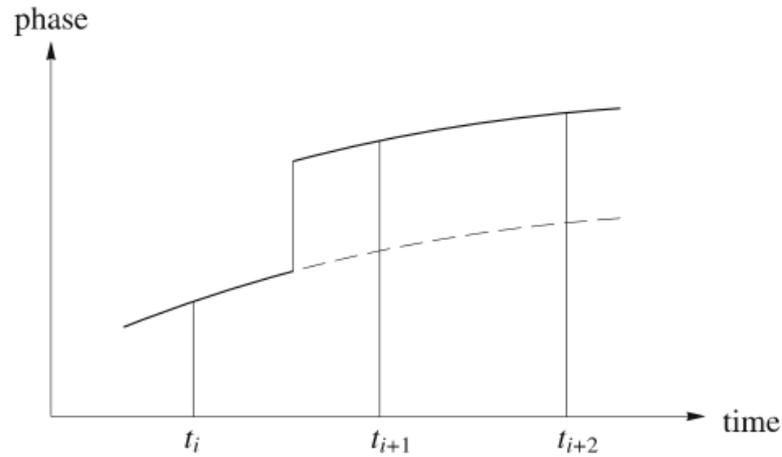
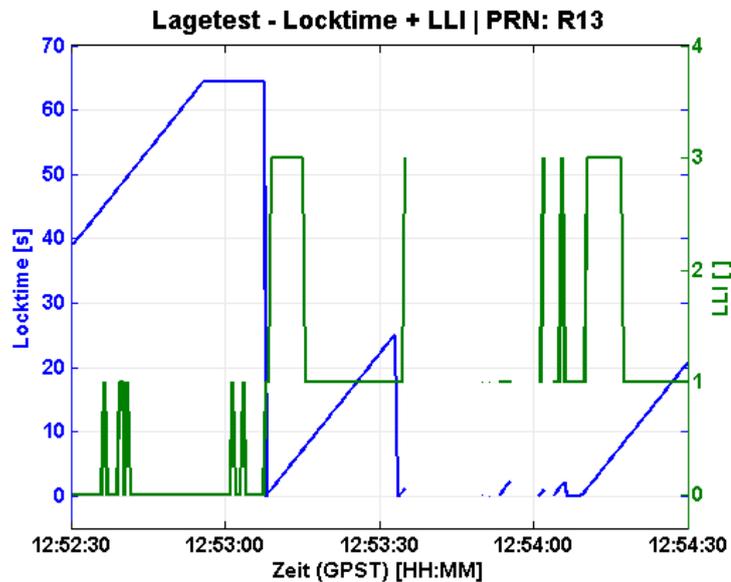


Abbildung 7.10: Darstellung eines Cycle Slip nach Hofmann-Wellenhof et al., (2008)

Um betroffene Satelliten auszuschließen, wird vor allem mit Hilfe der GOBS-Daten, die in Abschnitt 7.3.1 definiert wurden, ein erweiterter LLI bestimmt und zur Cycle Slip Detektion herangezogen. Dies ist im low-cost Bereich eine sehr stabile Methode, um Cycle Slips zu detektieren. Nach Hofmann-Wellenhof et al., (2008) können vor allem im high-cost Bereich mehrere Frequenzen zur Cycle Slip Detektion herangezogen werden, um dies zu erleichtern. Eine beispielhafte Darstellung der eRTK Methode kann Abbildung 7.11 entnommen werden.



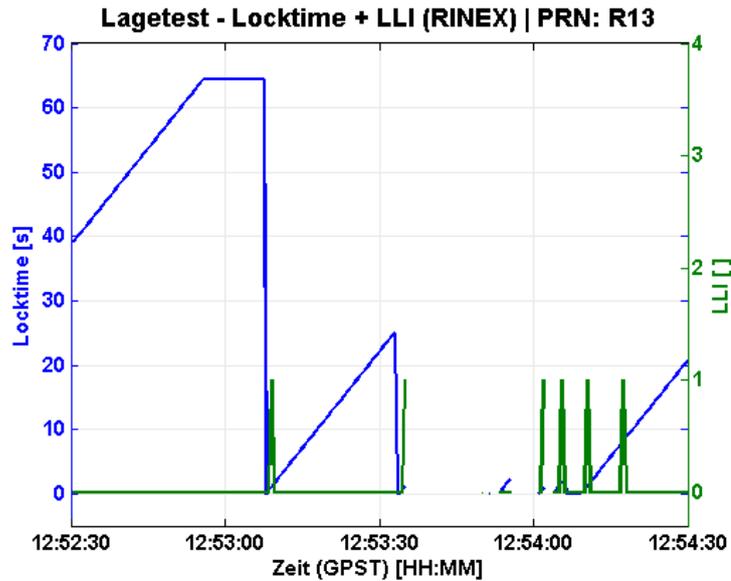


Abbildung 7.11: Cycle Slip Analyse eRTK, eRTK Bild 1,
RTKLIB Bild 2

Die Abbildung 7.11 zeigt den Unterschied der möglichen Cycle Slip Detektion anhand des LLI zwischen eRTK und der RTKLIB Version. RTKLIB verwendet für die Prozessierung nur den in der unteren Abbildung 7.11 dargestellten LLI. Dieser erfasst jedoch nicht alle ungelockten Phasenmessungen des Empfängers bzw. verwendet den betroffenen Satelliten nach einem Abbruch des Locks zu schnell wieder für die Prozessierung. Im oberen Teil der Abbildung 7.11 ist der erweiterte LLI des GOBS-Formates dargestellt. Hierfür werden die in Abschnitt 3.5 definierten Parameter für die erweiterte LLI-Berechnung herangezogen. Verliert die Phasenmessung ihren Lock bzw. herrscht eine Lockdauer von weniger als 30 Sekunden, so wird dem betroffenen Satelliten ein entsprechender LLI zugewiesen. Dies versichert der Prozessierung, dass nur Satelliten mit entsprechend gutem und langen Phasenlock verwendet werden und sich somit die Positionierung als stabil gestaltet. Für einen Überblick der erzielten Ergebnisse mit den zuvor angeführten Ergebnissen wird auf siehe Kapitel 8 verwiesen.

Kapitel 8

Ergebnisse und Evaluierung

Das entwickelte Softwarepaket eRTK, samt definierter Systemarchitektur, wird anhand spezieller Testszenarien evaluiert und getestet, sowie mit Ergebnissen der RTKLIB-Version gegenübergestellt. Zusätzlich werden noch Vergleiche zwischen high-cost-Positionslösungen, welche mit dem Topcon Hyper II im RTK-Modus erzielt wurden, mit der low-cost eRTK-Lösung durchgeführt. Die Tests werden nachfolgend näher beschrieben und sie wurden alle unter erschwerten Bedingungen, mit erhöhter Abschattung, durchgeführt. Um zwischen den high-cost und low-cost Aufbau konsistent zu bleiben, wurden die gleichen GNSS-Einstellungen und Updateraten für die Prozessierung verwendet. Anzumerken ist, dass alle Ergebnisse in Echtzeit realisiert wurden. Eine Spezifikation der Empfänger kann Abschnitt 8.1 entnommen werden.

8.1 Messaufbau

Für die Evaluierung der Systemarchitektur und der Softwareentwicklung eRTK wurden eigene Testszenarien am Dach des Geodäsiegebäudes konzipiert. Der grundlegende low-cost Systemaufbau besteht aus, einem Rover und Basisempfangler, ausgestattet mit u-blox M8T Empfänger samt Tallysmann TW340 GNSS-Antenne, einem UMTS-Datenstick für einen mobilen Internetempfang und einem Raspberry PI2 als Embedded-Plattform. Softwareseitig kam die Softwareentwicklung eRTK mit eigenen NTRIP-System, sowie die Standardsoftware RTKLIB zum Einsatz. Um nun auch Vergleiche mit geschlossenen high-cost RTK-Systemen durchführen zu können, wurde das Topcon Hyper II RTK-System für den Messaufbau verwendet. Hierfür wurde eine eigene Basisstation am Dach eingerichtet, welche die Korrekturdaten für RTK an den Rover übermittelt. Die verwendete GNSS-Hardware kann der Tabelle 8-1: GNSS-Messaufbau entnommen werden.

Tabelle 8-1: GNSS-Messaufbau

Spezifikation	ublox M8T	Topcon Hyper II
Frequenzen	L1	L1 und L2
GNSS	GPS+GLO	GPS+GLO
Updaterate	2 Hz	2 Hz
Solution	Echtzeit eRTK + RTKLIB	RTK Output [NMEA]
Datenübertragung	Mobil 3G (Netz: HoT)	Topcon Funk

Das Konzept des Messaufbaus ist nun in Abbildung 8.1 dargestellt und zeigt die Verwendung der low-cost Systemarchitektur im Vergleich mit einer high-cost RTK-Architektur.

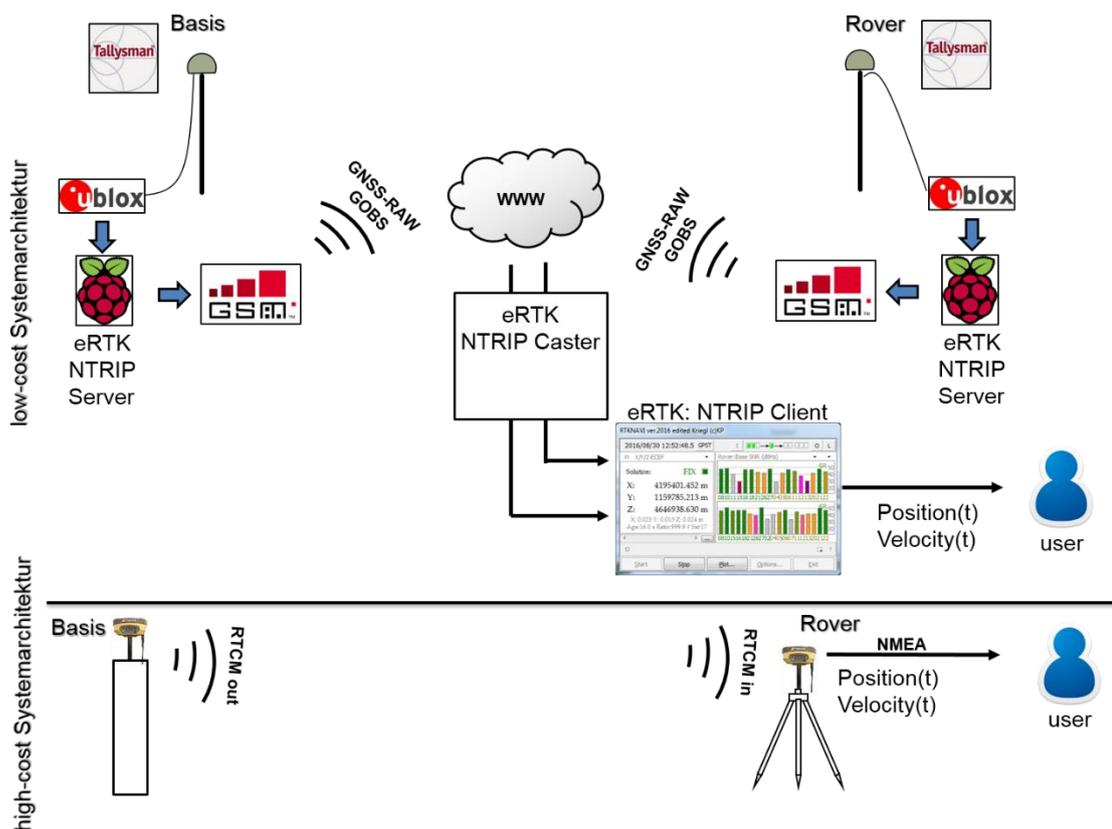


Abbildung 8.1: Messkonzept

8.2 Versuchsreihen

Der grundlegende Systemaufbau der Testmessungen gliedert sich in drei unterschiedliche Szenarien (i) einem Test der Höhenkomponente mit einer Treppenfunktion als Höhenänderung unter erschwerten Bedingungen, (ii) einem Lagetest durch eine Kreisbewegung, durchgeführt mit konstanter Drehbewegung mit dem institutseigenen Drehtisch RAK unter guten Bedingungen, sowie (iii) einem ähnlichen Lagetest unter erschwerten Bedingungen. Die Testmessungen wurden am 25.08.2016 und 30.08.2016 am Dach des Geodäsiegebäudes durchgeführt. Als Basisstationsparameter wurden die vom Institut von Geodäsie bestimmten Koordinaten der Messpfeiler verwendet, siehe Tabelle 8-2 und Abbildung 8.2.

Tabelle 8-2: Pfeilerkoordinaten INAS, ITRF2008 Epoche 2012/07/18

Basisstation	Pfeiler	ECEF X [m]	ECEF Y [m]	ECEF Z [m]	Antennenhöhe [m]
ublox M8T	5	4195390.479	1159800.683	4646944.500	0.125
Topcon	2	4195402.958	1159779.515	4646938.544	0.195

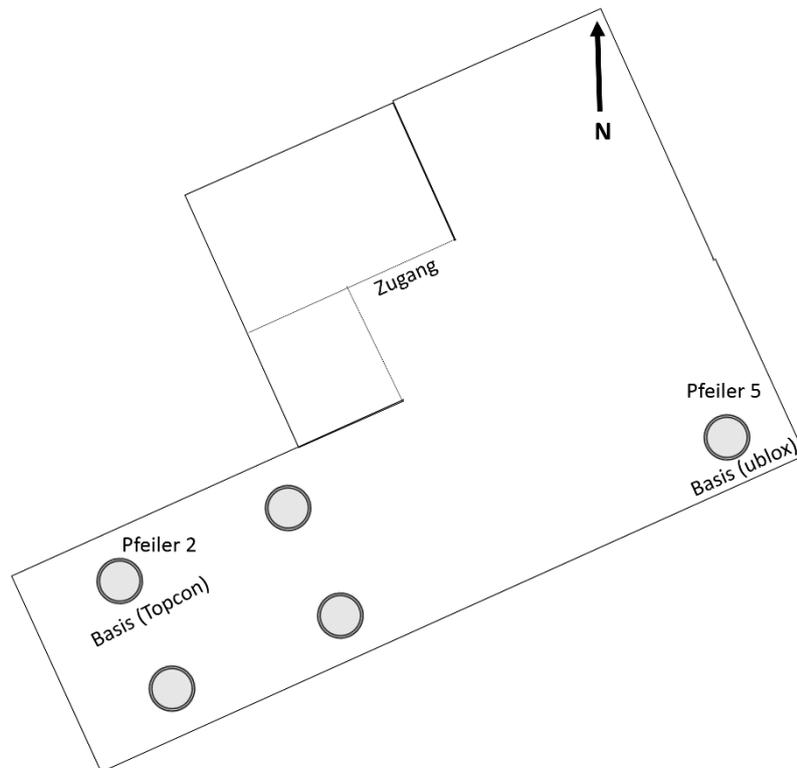


Abbildung 8.2: Skizze Basisstationen, Dach Geodäsiegebäude

8.3 Testscenario – Höhe

Da die Höhenkomponente im Vergleich zur Lage bei GNSS-Positionierungsmethoden aufgrund der Schnittgeometrie bei GNSS die ungenaueste ist, wird hierfür ein eigenes Testscenario verwendet. Somit kann gezeigt werden, dass auch die low-cost GNSS-Empfänger eine stabile und genaue Höhenkomponente liefern kann. Dazu wurde unter erschwerten Messbedingungen nah am Gebäude, eine Stufenweise Messung (10 cm Stufen) durchgeführt, siehe Abbildung 8.3. Diese Messbedingungen bringen vor allem sehr viel Multipatheffekte mit sich.

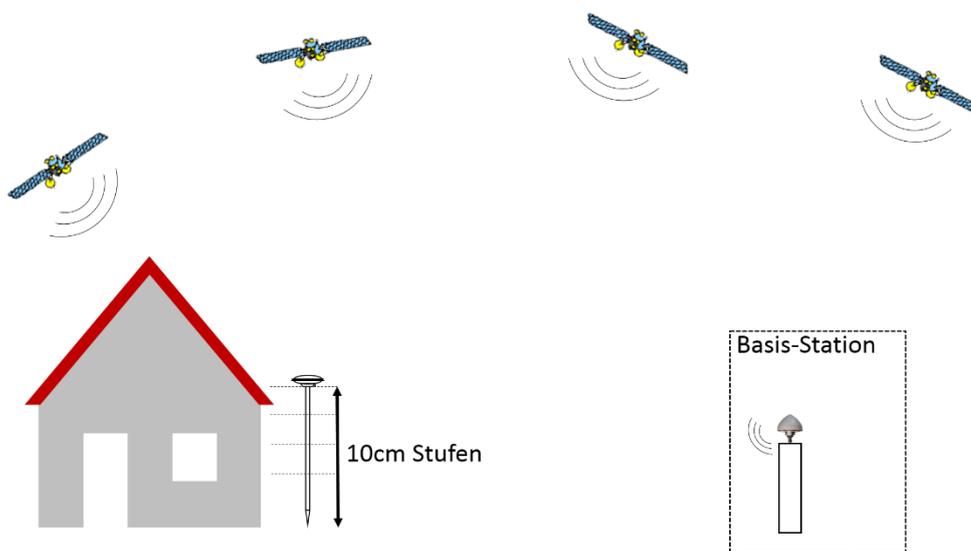


Abbildung 8.3: Skizze Höhentest

Die Ergebnisse der Höhenmessung können nun den nachfolgenden Kapiteln entnommen werden. Alle Ergebnisse wurden in Echtzeit erzielt, die Analyse der Rohdaten wurde mit den mitgeloggten Daten des Rovers durchgeführt. Dass für den Höhentest kein high-cost System verwendet wurde, hat den Grund, dass man die stufenweise Veränderung der Höhe exakt kennt und somit kein zwingender Vergleich mit einer high-cost Lösung notwendig ist. Dies ist dadurch beschreibbar, da der Messaufbau 10 cm Marken beinhaltet und somit von einer Höhe von 2 m über dem Boden acht exakte 10 cm Schritte nach unten durchgeführt werden können. Außerdem macht der Messaufbau dieses Tests eine synchrone Messung mit beiden Systemen nur schwer möglich.

8.3.1 Messergebnisse

Abbildung 8.4 zeigt den Messaufbau und die Positionslösungen der beiden Varianten. Aus der Lösung ist ersichtlich, dass eRTK eine deutlich stabilere und genauere Positionslösung liefert als das bei RTKLIB der Fall ist. Wie in weiteren Analysen ersichtlich wird, verliert RTKLIB, vor allem nach diversen Satellitenwechsel, die Fixierung aufgrund von auftretenden Cycle Slips. Insbesondere bei steigender Abschattung aufgrund von sinkender Höhe, liefert RTKLIB eine Cycle Slip behaftete, unstabile Positionslösung der Höhenkomponente.

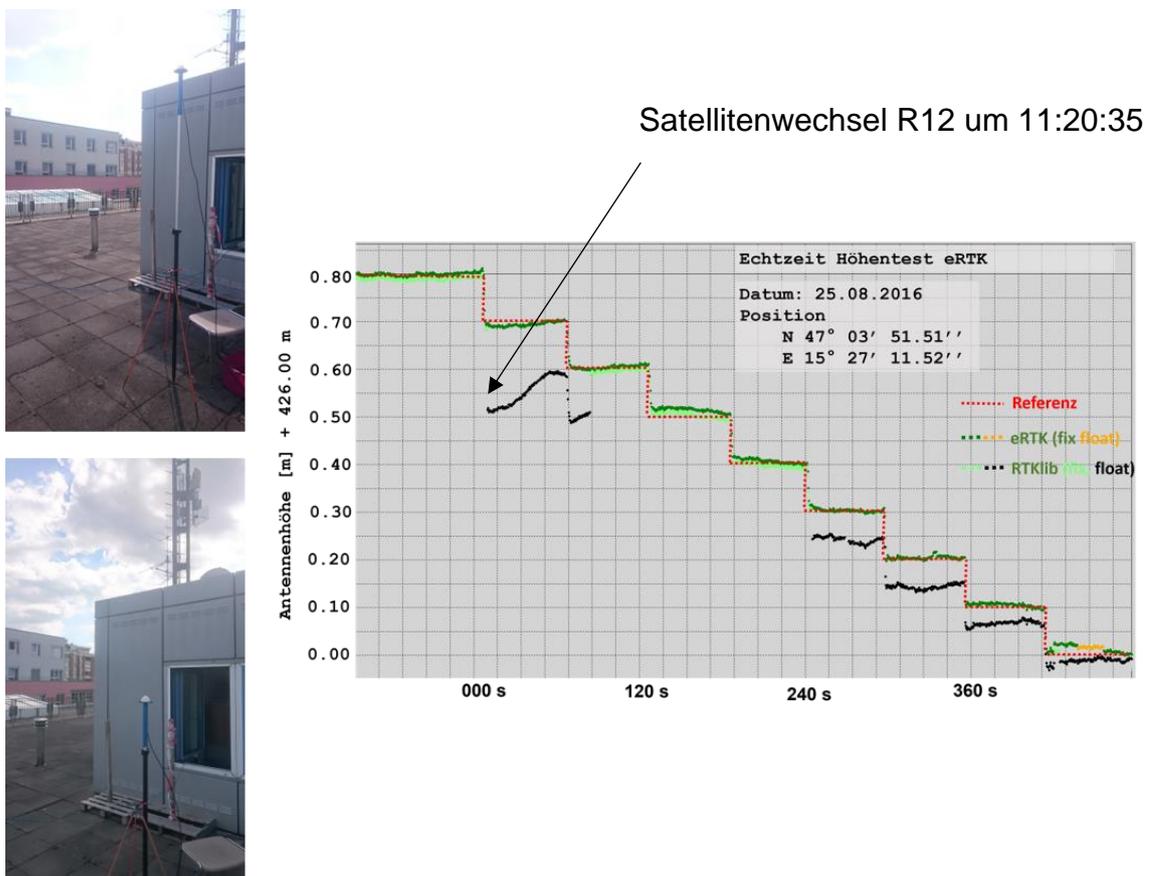


Abbildung 8.4: Messergebnisse Höhentest

Die empirische Standardabweichung der Höhenkomponente der eRTK Version ist in Abbildung 8.5 dargestellt und zeigt einen Anstieg der Standardabweichung mit sinkender Höhe des Testaufbaus. Dies korreliert stark mit der hohen Abschattung und dem damit hergehenden Wegfall von Satelliten.

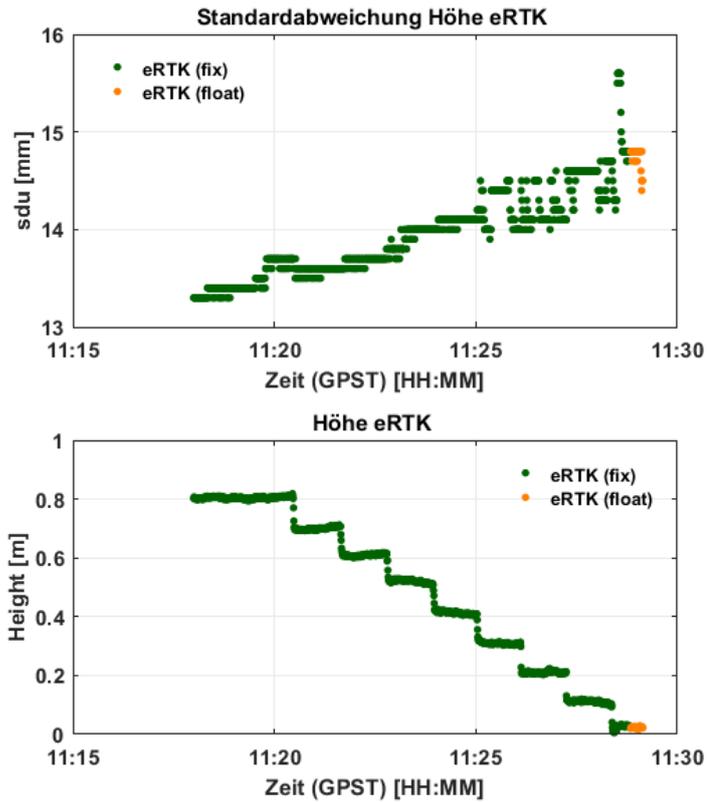


Abbildung 8.5: Standardabweichung der eRTK Höhenkomponente

Die erreichte Positionsgenauigkeit in der Höhenkomponente liegt damit in einem Bereich, welcher mit low-cost Empfänger für eine Höhenbestimmung unter erschwerten Bedingungen möglich ist. Um die erzielten Lösungen der eRTK und RTKLIB Version vergleichen zu können, werden die aufgezeichneten Rohdaten analysiert. Ins Auge gefasst werden vor allem die Zeitpunkte mit auftretenden Cycle Slip in der RTKLIB Lösung.

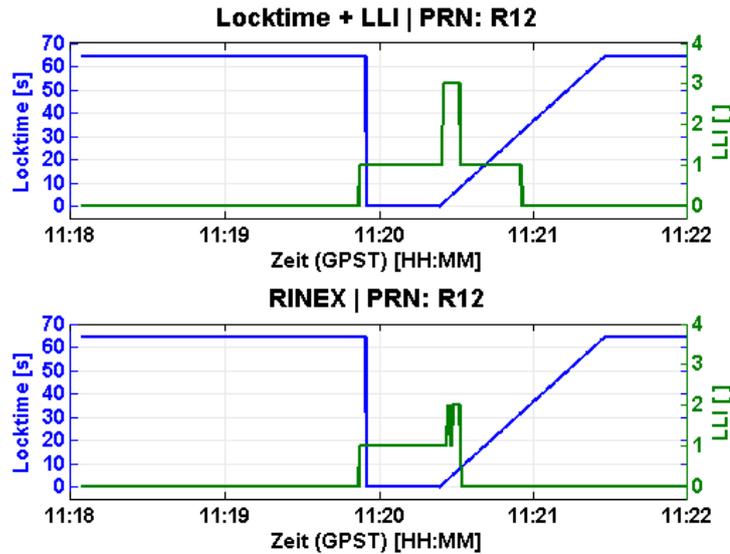


Abbildung 8.6: Locktime+LLI R12, Höhentest

Der in Abbildung 8.4 erwähnte Satellitenwechsel lässt sich nun auf einen kurzen Loss of Lock in der Phasenmessung zurückführen. Wie u.a. in Kapitel 3 erwähnt, hat RTKLIB nur die nach RINEX standardisierten Parameter zur Verfügung und nimmt den Satelliten nach erfolgreicher Akquisition wieder in den EKF, ohne auf etwaige Cycle Slip zu achten. Durch dieses zu rasche Hinzufügen des Satelliten in den EKF kommt es zu einem Sprung in der Positionslösung, da diese Messung mit Cycle Slip behaftet ist. Die Lösung der eRTK Version überprüft jedoch die Locktime, Standardabweichung der Phasenmessung und sonstige Qualitätsparameter aus dem GOBS-Format auf eine mögliche fehlerhafte Messung und schließt genau diesen Satelliten R12 aus der Positionsberechnung aus. In weiterer Betrachtung kann auch gezeigt werden, dass diesen betroffenen Satelliten ein geringes SNR zu Grunde liegt (10-35 dBHz). Dies steht auch in Korrelation mit dem empfängerspezifischen Wert der Standardabweichung von der Phasenmessung. Nach Zogg (2002) sollte ein low-cost GNSS-Empfänger für eine präzise und stabile RZK-Lösung Satelliten mit einem SNR von geringer als 30 dBHz von der Prozessierung ausgeschlossen werden. Dies ist ebenfalls in Abbildung 8.7 ersichtlich, worin bei einem SNR, welches unter die 30 dBHz Grenze fällt, die Standardabweichung der Phase deutlich zunimmt.

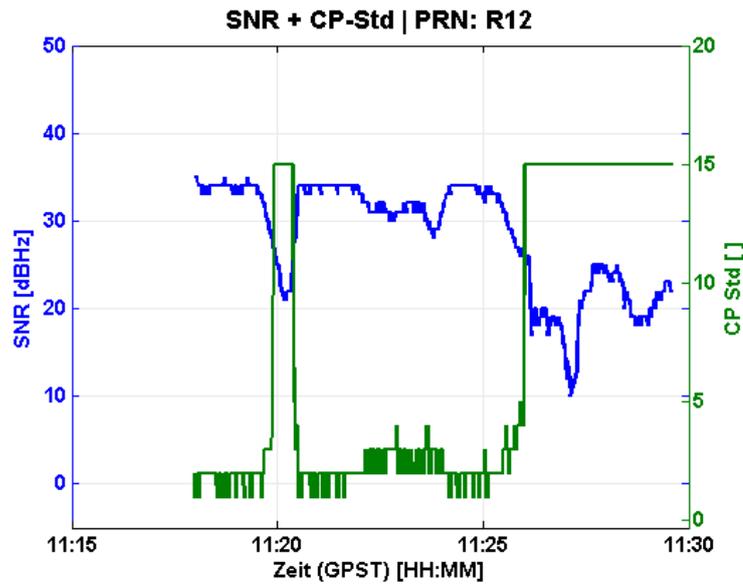


Abbildung 8.7: SNR+Std. R12, Höhentest

Auch die Gewichtung der einzelnen Satelliten im Zuge der RTK-Prozessierung, lässt sich damit gut steuern. Abbildung 8.8 zeigt den Unterschied der Gewichtung der eRTK und RTKLIB Methode.

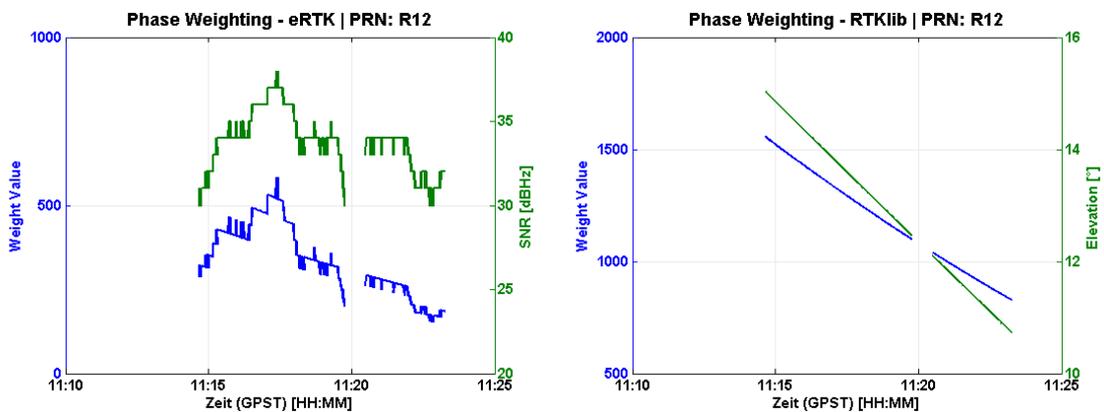


Abbildung 8.8 Gewichte R12, Höhentest

Die linke Abbildung zeigt nun eine SNR+elevationsbedingte Gewichtung, wie sie nach Abschnitt 7.3.3 in eRTK definiert wurde. Die rechte Abbildung zeigt eine reine Elevation-Gewichtung, wie sie RTKLIB verwendet. Gezeigt wird, dass bei geringen SNR Werten die Gewichte in eRTK um den Faktor 2-3 geringer sind als in RTKLIB. Die Elevation hat bei dieser Art der Gewichtung eine Trendwirkung.

Steigt der Satellit ab (sinkende Elevation), wie R12, so ist ein leichter Trend der Gewichte in eRTK ersichtlich.

8.3.2 *Fazit*

Durch die Analyse der Rohdaten in Zusammenhang mit den erzielten Positionslösungen kann gezeigt werden, dass (i) durch die Verwendung des GOBS-Formates mit zusätzlichen Rohdaten eine präzisere und stabilere Positionslösung erzielt wird, dass (ii) vor allem Cycle Slip dadurch in eRTK vermieden werden bzw. betroffene Satelliten ausgeschlossen werden, dass (iii) eRTK im Allgemeinen unter erschwerten Bedingungen bessere Ergebnisse als RTKLIB liefert und dass (iv) auch durch eine SNR+elevationsbedingte Gewichtung Verbesserungen durch die Verwendung des SNR unter erschwerten Bedingungen erzielt werden können.

8.4 TestszENARIO – 2D-Position Test 1

Neben dem gesondert durchgeführten Höhentest wurden zwei unterschiedliche Tests der Positionsgenauigkeit von eRTK durchgeführt. Der erste Test wurde unter guten Bedingungen am Messpfeiler 3 am Dach des Geodäsiegebäudes durchgeführt. Zum Einsatz kam ein Drehtisch RAK 200, welcher eine konstante Drehbewegung ermöglicht. Da die Mittelpunktkoordinaten sowie die Auslegerlänge von 1 m bekannt sind, hat man eine Across-Track Genauigkeitsinformation im Vergleich zur Referenztrajektorie zur Verfügung. Zum Einsatz kommt die bereits beschriebene low-cost Systemarchitektur sowie die high-cost RTK-Lösung von Topcon, siehe Abbildung 8.1: Messkonzept. Der grundlegende Aufbau der Messung ist in Abbildung 8.9: Messaufbau dargestellt.

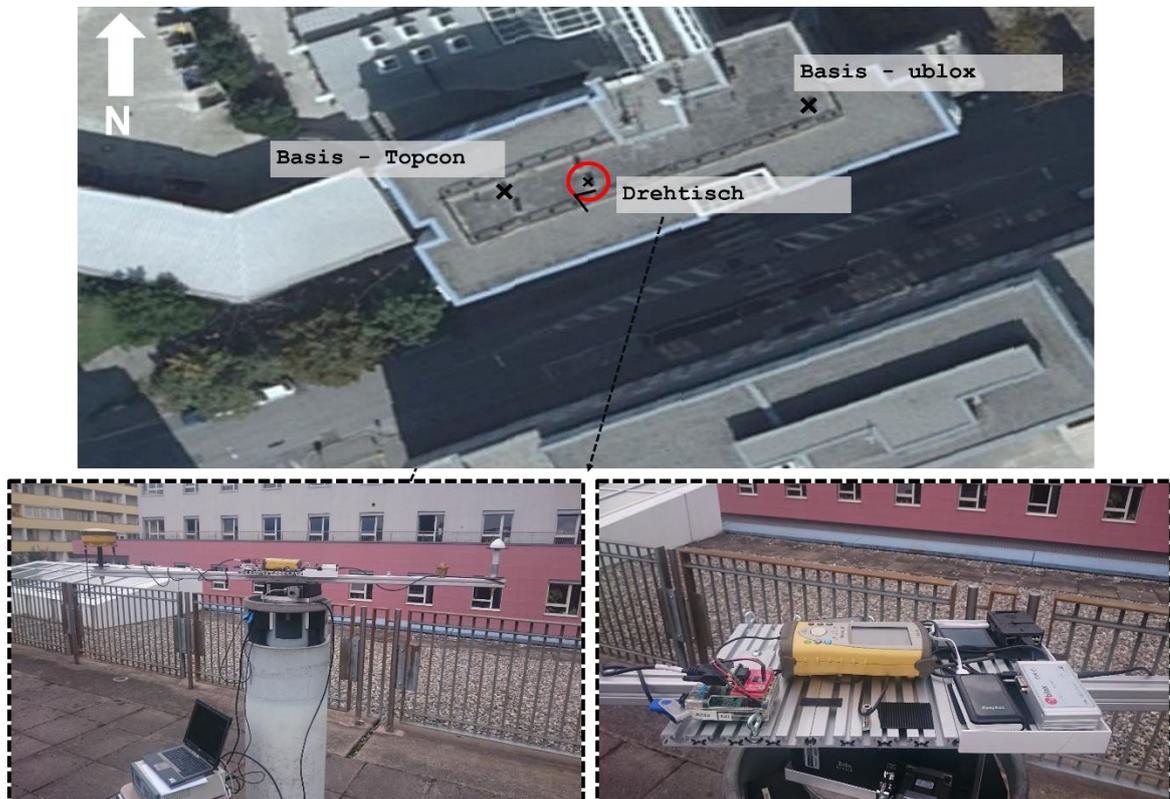


Abbildung 8.9: Messaufbau 2D-Position Test 1

Diese Testmethode erlaubt es, eine Aussage über die relative als auch über die absolute Positionsgenauigkeit zu treffen. Die erzielten Ergebnisse der rund 15 min langen Messung samt Analysen werden nachfolgend näher beschrieben.

8.4.1 Messergebnisse

Nachfolgende Abbildungen 8.10 und 8.11 zeigen die erzielten Positionslösungen der low-cost Systemarchitektur, dargestellt als Nord-Süd und Ost-West Komponente in einem lokalen Horizontsystem mit dem Ursprung im Messpfeiler 3.

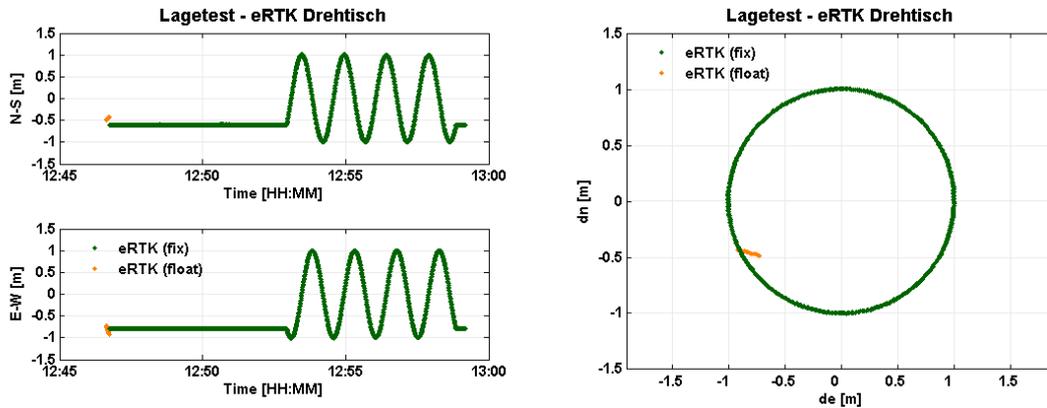


Abbildung 8.10 Positionslösung eRTK, Lagetest 1

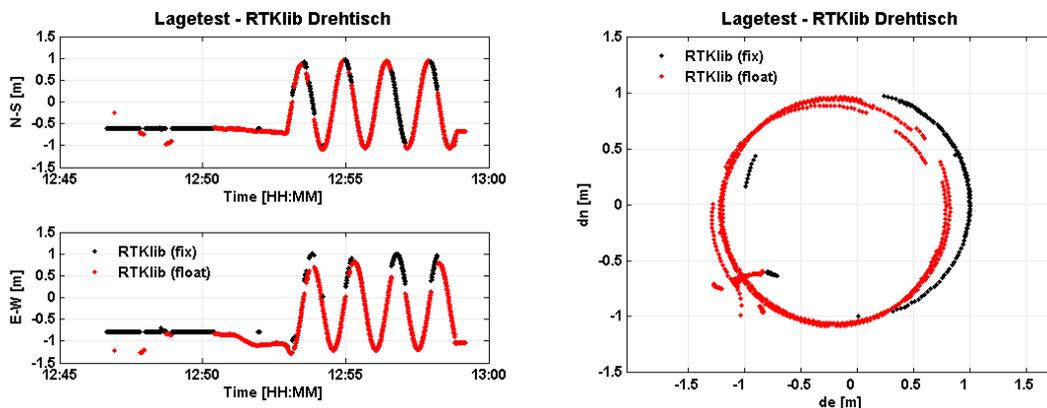


Abbildung 8.11 Positionslösung RTKLIB, 2D-Position Test 1

Die obere Abbildung zeigt die erzielte Trajektorienlösung der eRTK-Software. Durch den Einsatz des Static-Start Modus kann eine deutlich schnellere Fixierung als bei der RTKLIB Lösung erreicht werden. Insgesamt erzielt eRTK eine Fixierung von 93,6 % im Laufe der Messung, wobei RTKLIB eine Fixierung von nur 34,3 % ermöglicht, obwohl sehr gute Messbedingungen vorliegen. Durch das Verlieren der Fixierungen bei RTKLIB verschlechtert sich damit auch die erzielbare Genauigkeit der Positionslösung, was in Abbildung 8.12 dargestellt ist.

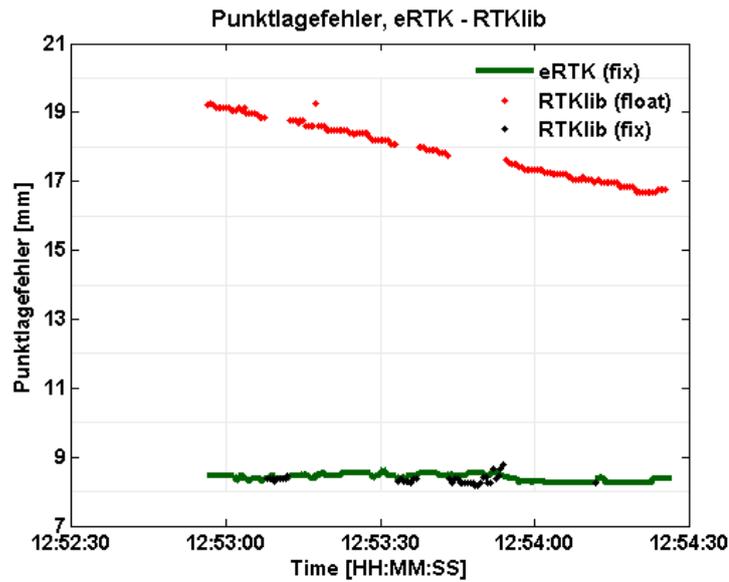


Abbildung 8.12: Punktlagefehler, 2D-Position Test 1

eRTK erreicht einen durchschnittlichen Punktlagefehler von 8.5 mm in der Lage. RTKLIB kommt nur an diesen Wert heran, wenn eine Fixierung vorliegt. In weiterer Folge wird auf einen Vergleich der low-cost Ergebnisse mit high-cost Lösungen von Topcon eingegangen. Dazu wird zur besseren Übersicht nur die erste Runde der Messung betrachtet.

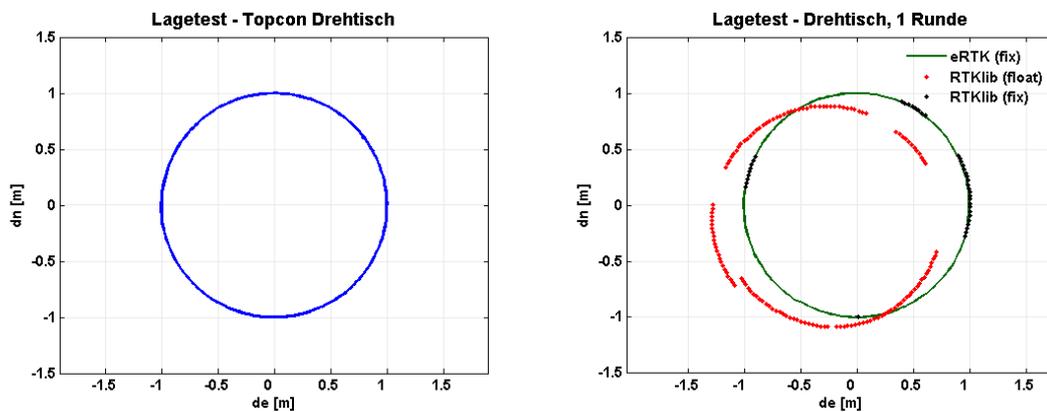


Abbildung 8.13 Trajektorie 1 Runde, 2D-Position Test 1

Abbildung 8.13 zeigt nun die erzielten Trajektorien, links die high-cost Lösung und rechts die beiden low-cost Lösungen. Die Topcon und eRTK Lösungen haben eine 100 % Fixierung und werden anhand der Abweichungen zur Solltrajektorie in Abbildung 8.14 näher betrachtet.

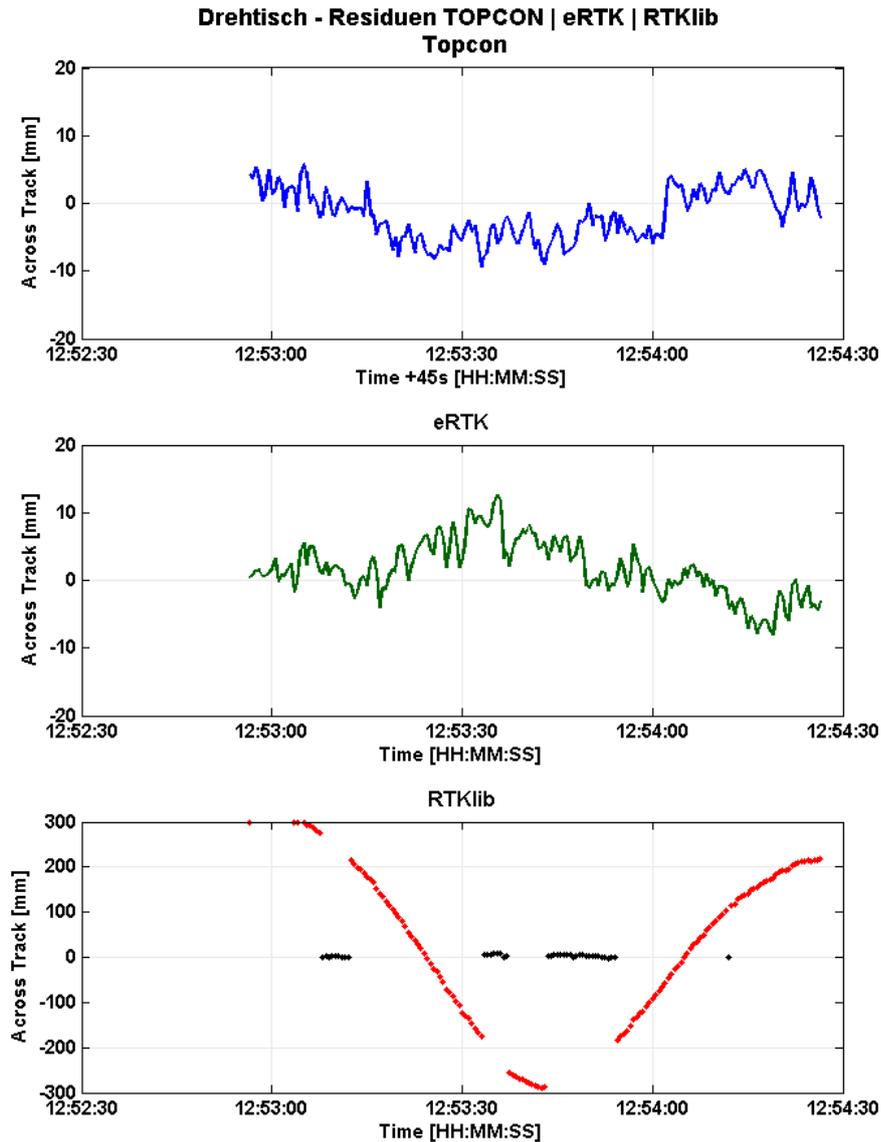


Abbildung 8.14: Residuen eine Runde, 2D-Position Test 1

Die Topcon und eRTK Trajektorie weichen maximal 10 mm von der Solltrajektorie ab, was nach Zogg (2002) dem Messrauschen von GNSS-Positionen im low-cost Bereich entspricht. Vor allem der geringe Unterschied der low-cost eRTK Lösung zur high-cost Positionslösung von Topcon soll hier hervorgehoben werden. Abbildung 8.13 zeigt die RTKLIB Lösung als eine Kreistrajektorie, welche nach Form und Größe die der wahren entspricht, jedoch aufgrund von Cycle Slip einen konstanten Offset in der Lösung aufweist. Betrachtet man nun die aufgezeichneten Rohdaten der low-cost Systemarchitektur im oben dargestellten Zeitraum, so lässt sich wieder feststellen, dass durch die Verwendung des GOBS-Formates

deutliche Vorteile in der Prozessierung entstehen. Hervorzuheben ist, dass Rohdaten der high-cost Lösung im RTK-Modus nicht zur Verfügung stehen und nur auf die Positionslösungen im NMEA-Format zugegriffen werden kann. Betrachtet man nun wieder die für die Phasenmessung verantwortlichen Qualitätsindikatoren wie Locktime und LLI, so lassen sich zwischen eRTK mit dem GOBS-Format und RTKLIB mit RINEX (bzw. RTCM) Format deutliche Unterschiede feststellen.

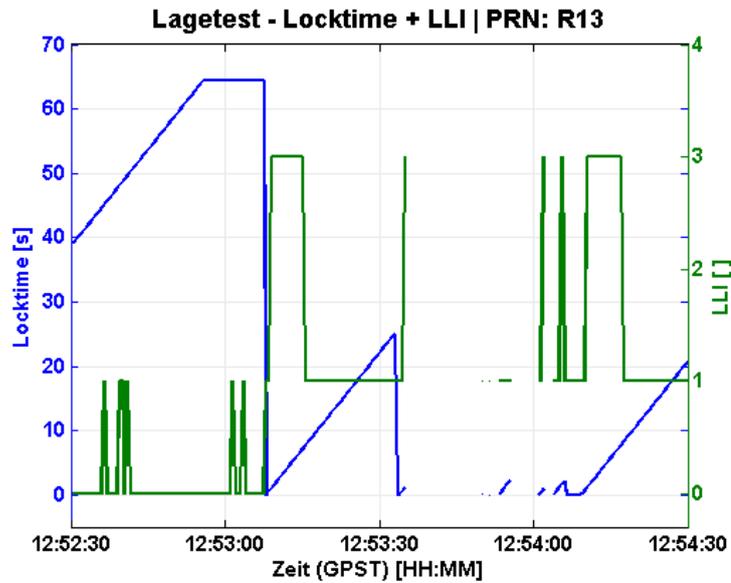


Abbildung 8.15: eRTK Rohdaten, 2D-Position Test 1

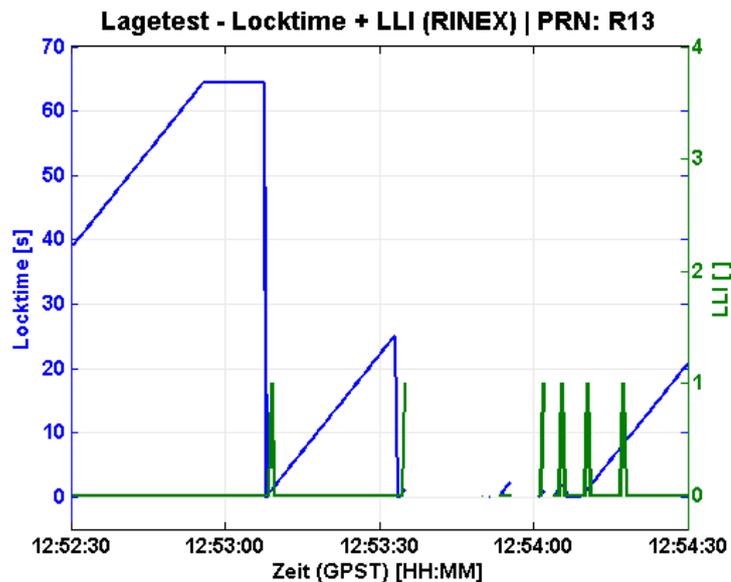


Abbildung 8.16: RTKLIB Rohdaten, 2D-Position Test 1

Vor allem der erweiterte LLI in eRTK bzw. mit GOBS führt zu einer verbesserten Prozessierung im RTK-Modus. eRTK schließt den Satelliten mit der PRN R13

durch die Einbeziehung aller Indikatoren aus, womit Cycle Slip, welche jedoch in der RTKLIB Lösung auftreten, vermieden werden.

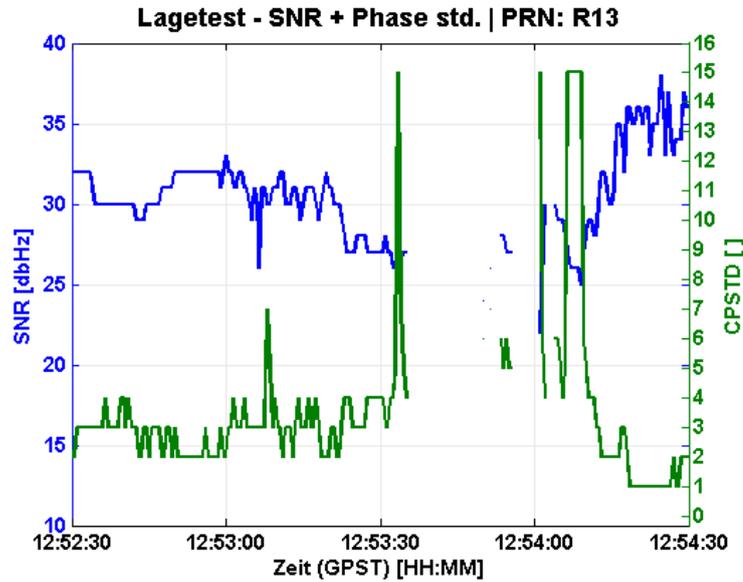


Abbildung 8.17: eRTK Rohdaten 2, Lagetest 1

Auch hier lässt sich wieder zeigen, dass der betroffene Satellit ein geringes SNR um 30 dBHz aufweist, bzw. kurzzeitig gar nicht getrackt werden kann. Die Standard RTKLIB-Prozessierung jedoch verwendet diesen Satelliten, sobald ein Tracking vorliegt, für die Prozessierung. Durch die schlechte Qualität der Phasemessung dieses Satelliten kommt es jedoch zu Cycle Slip, was die Ergebnisse erheblich verfälscht.

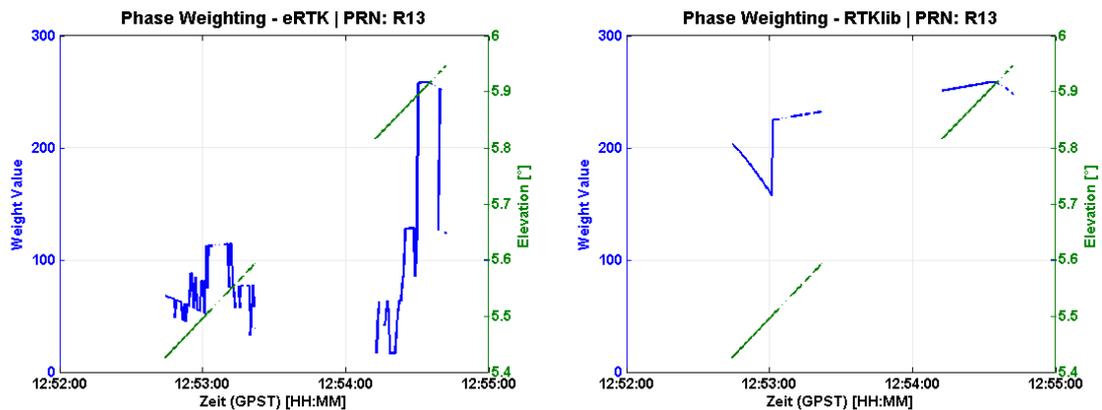


Abbildung 8.18 Gewichte R13, 2D-Position Test 1

Erweitert man nun die SNR+elevationsbedingte Phasengewichtung mit der im GOBS-Format vorhandenen Standardabweichung der Phasemessung, wird

eine neue Gewichtungsmethode erreicht. Die richtige Anpassung der Gewichtung ist jedoch sehr situationsbedingt. Wie in Abbildung 8.18 ersichtlich, bekommt der Satellit R13 ein um Faktor 2 schlechteres Gewicht in eRTK als in RTKLIB. Dies trifft vor allem auf Satelliten unter schlechten Bedingungen (geringe Elevation, geringes SNR) zu. Treten jedoch hauptsächlich solche Satelliten auf, d.h. nur wenige Satelliten haben ein gutes SNR ($> 35\text{dBHz}$), so kommt es zu einem deutlich höheren Rauschen in der Positionslösung. Cycle Slip werden im Vergleich zu RTKLIB dennoch vermieden und erkannt.

8.4.2 *Fazit*

Auch hier kann durch die Analyse der Rohdaten, in Zusammenhang mit den erzielten Positionslösungen, gezeigt werden, dass (i) durch die Verwendung des GOBS-Formates mit zusätzlichen Rohdaten eine präzisere und stabilere Positionslösung für kinematische Anwendungen auch unter guten Bedingungen erzielt werden kann, dass (ii) auch die high-cost Lösung von Topcon kaum bessere Ergebnisse erzielt als die eingesetzte entwickelte low-cost Systemarchitektur und dass (iii) die Gewichtung und deren Einfluss auf die Positionslösung stark abhängig ist von der jeweiligen Situation und Geometrie der Satellitenanordnung.

8.5 TestszENARIO – 2D-Position Test 2

Für die Durchführung eines kinematischen Lagetests wurde ein weiterer Versuchsaufbau mit dem Drehtisch RAK 200 durchgeführt. Dazu wurden mit dem high-cost Topcon RTK-System ein Punkt eingemessen, auf welchem ein Stativ mit dem Drehtisch aufgebaut wurde. Dabei führt die Drehbewegung von guten Bedingungen in einen stark abgeschatteten Bereich, siehe Abbildung 8.19 und 8.20.

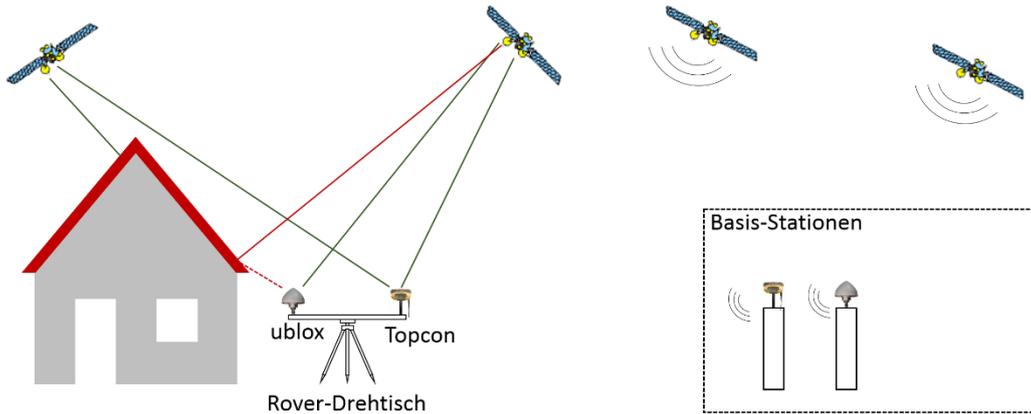


Abbildung 8.19: Aufbau Multipath

Da die Mittelpunktkoordinaten sowie die Auslegerlänge von 1 m bekannt sind, hat man eine Across-Track Genauigkeitsinformation als Referenztrajektorie zur Verfügung. Zum Einsatz kommt die bereits beschriebene low-cost Systemarchitektur sowie die high-cost RTK-Lösung von Topcon, siehe Abbildung 8.20.

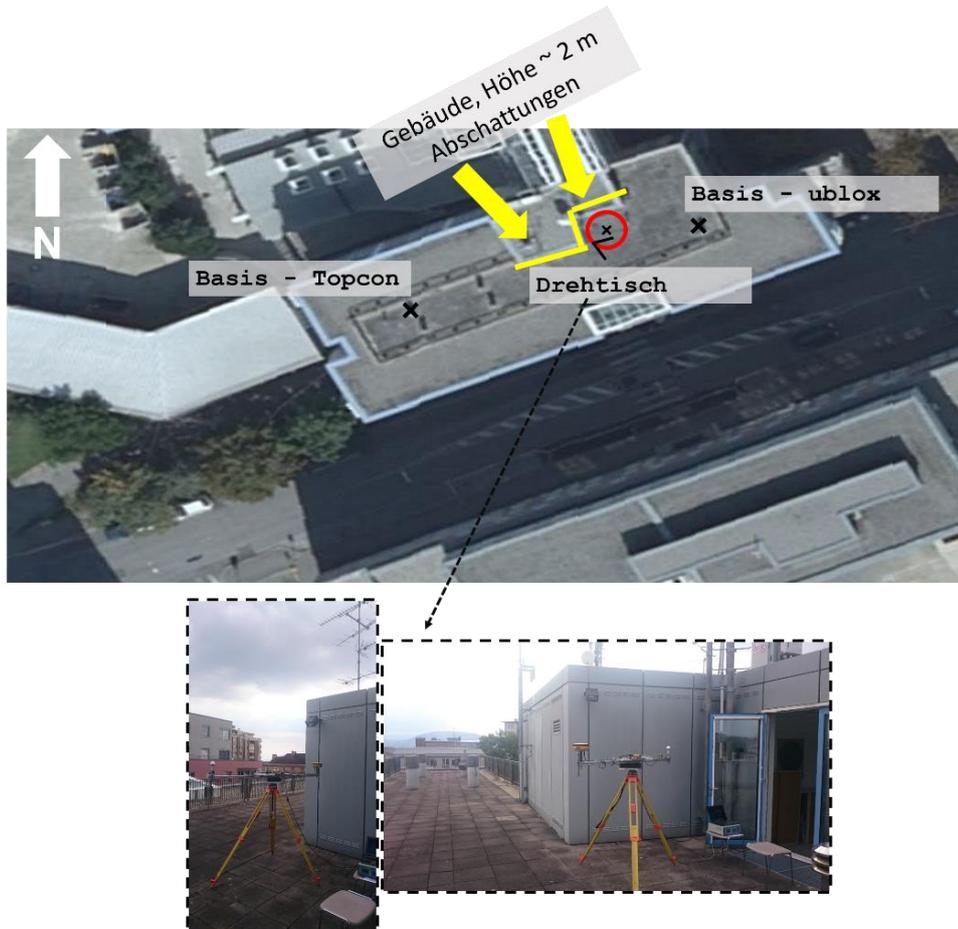


Abbildung 8.20: Systemaufbau 2D-Position Test 2

Mit diesem speziellen Messaufbau lässt sich die Systemarchitektur unter erschweren Bedingungen in der Kinematik testen. Außerdem ist mit viel Multipath zu rechnen und Vergleiche zwischen der low-cost Systemarchitektur und der high-cost Lösung können betrachtet werden.

8.5.1 Messergebnisse

Abbildung 8.21 zeigt die erzielten Trajektorien des oben beschriebenen Testaufbaus. Der obere Teil stellt die Lösung der low-cost Systemarchitektur mit eRTK und RTKLIB dar und der untere Teil beschreibt die high-cost Lösung von Topcon.

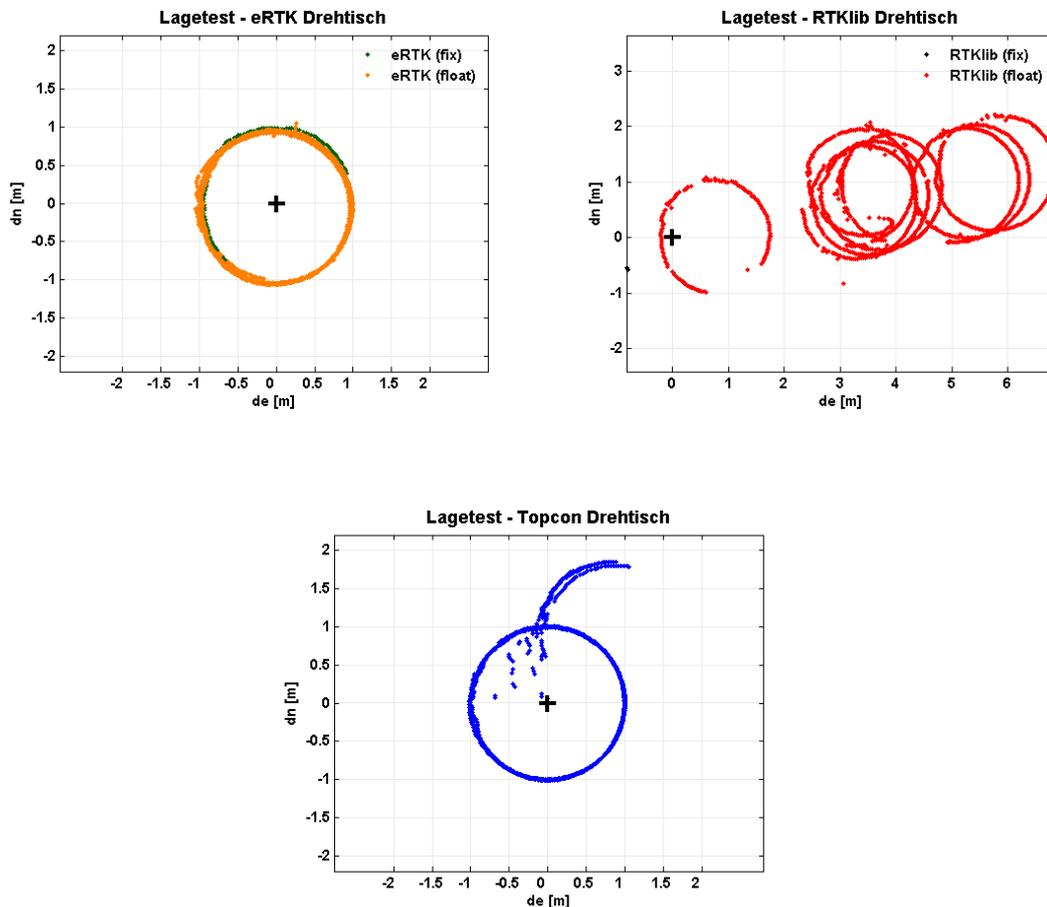


Abbildung 8.21 Positionslösung Gesamt, 2D-Position Test 2

Es ist deutlich ersichtlich, dass eine schlechtere Positionslösung als in Abschnitt 8.4, erzielt wird. Vor allem die Versetzung Richtung Osten (rechts) kann gezeigt werden. Dies ist auf die Multipath Effekte der Abschattung im linken Bereich (Westen) zurückzuführen, siehe Abbildung 8.20. Somit versetzt es die Trajektorie

aufgrund von Multipath und auftretenden Cycle Slip. Auch in der high-cost Lösung von Topcon treten deutliche Ausreißer in der Positionslösung auf, welche in eRTK durch Verwendung des GOBS-Formates vermieden werden. Aber auch mit der entwickelten Systemarchitektur samt eRTK kann nur eine Fixierung von rund 5.5 % erreicht werden. Die auftretende Float-Lösung (Ambiguity Ratio wurde nicht erreicht, siehe Formel 6-16) passt sich jedoch der Solltrajektorie gut an und weist den Abschattungen zugrundeliegend nur ein höheres Rauschen auf. Betrachtet man nun die Ursache der schlechten Positionslösungen in diesem Testaufbau, wird die stark variierende Anzahl an Satelliten erkennbar. Durch ein ständiges Verlieren und wieder Hinzugewinnen des Trackings einer Phasenmessung, wird der große Einfluss des GOBS-Formates mit eRTK ersichtlich. Unter solch erschwerten Bedingungen kann mit der entwickelten low-cost Systemarchitektur eine stabile und den Umständen entsprechende präzise Positionslösung, welche auch von high-cost Systemen nicht erzielt wird, garantiert werden.

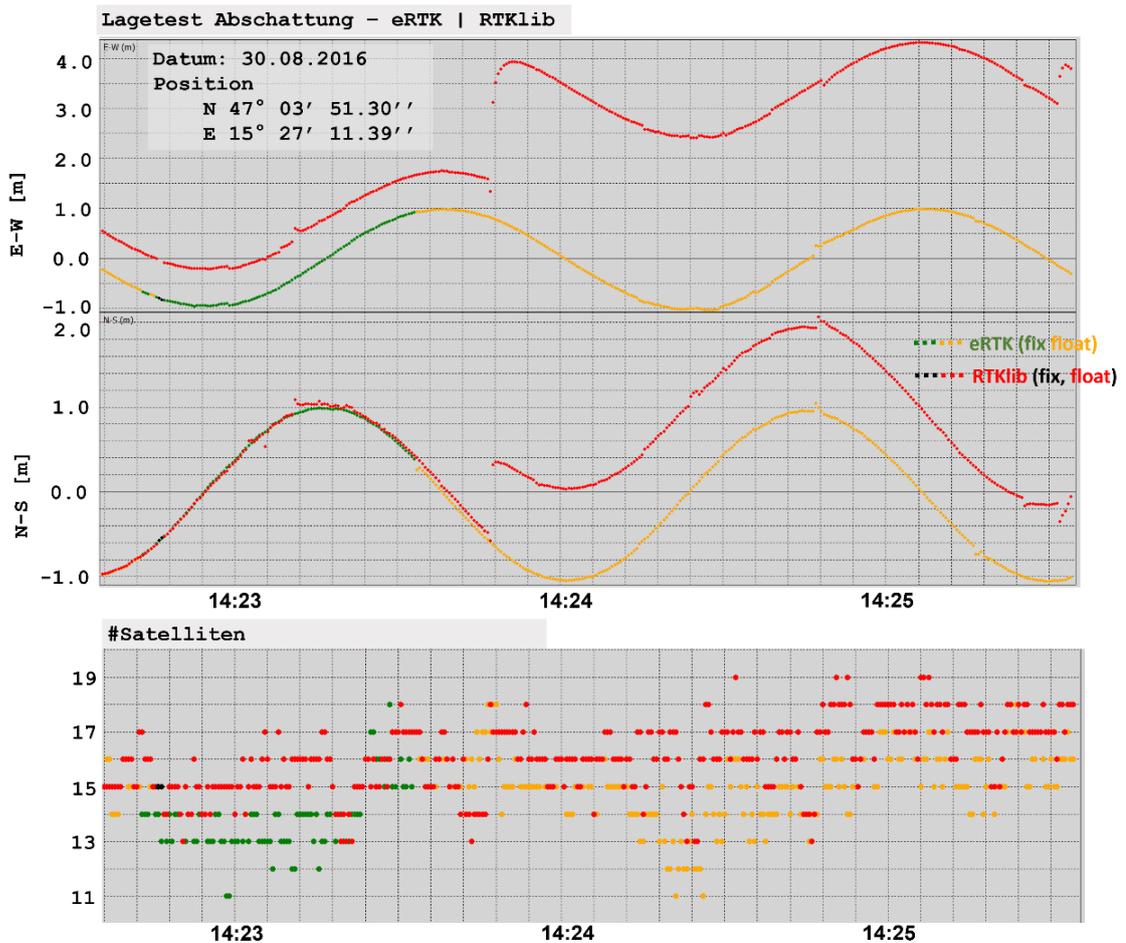


Abbildung 8.22: Positionslösung zwei Runden, 2D-Position Test 2

Abbildung 8.22 zeigt die Trajektorie samt Satellitenwechsel in einem Ausschnitt von zwei Runden, mit in Abbildung 8.23 zugehörigem, empirischen 2D-Punktla- gefehler nach Helmert. Durch die ständigen Wechsel der Satelliten, ist RTKLIB mit einem deutlich höheren Punktlagefehler behaftet. Der Punktlagefehler der eRTK Lösung liegt hingegen um robuste 8-10 mm.

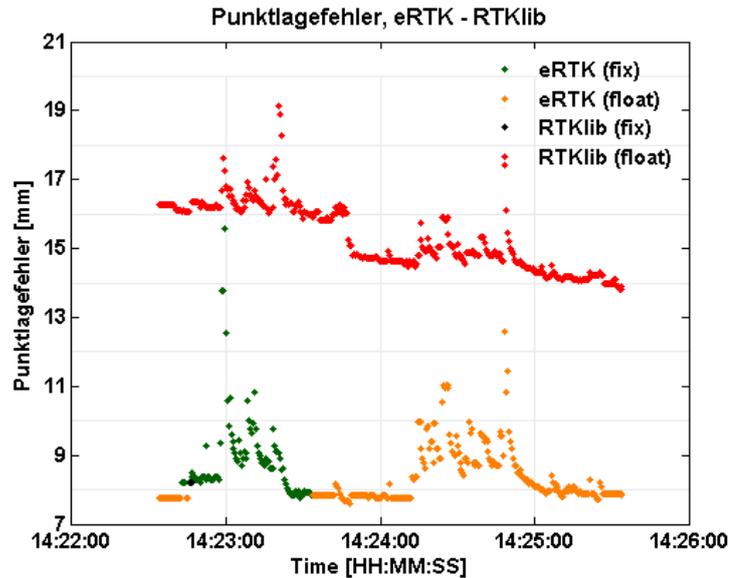


Abbildung 8.23: Punktlagefehler 2 Runden, Lagetest 2

Betrachtet man die Across Track Residuen der Trajektorien zur bekannten Soll- trajektorie, ist der deutliche Einfluss der Abschattung durch die Wand samt Mul- tipath ersichtlich. Einzig die low-cost Systemarchitektur um eRTK erlaubt eine stabile Positionslösung über den gesamten Messzeitraum, mit einer maximalen Abweichung zur Solltrajektorie von unter 10 cm. Durch das Auftreten der Cycle Slips in der low-cost Lösung von RTKLIB weicht diese Lösung mehrere Meter vom Soll ab. Auch die high-cost Lösung ist nach gewisser Zeit einigen Cycle Slip ausgesetzt.

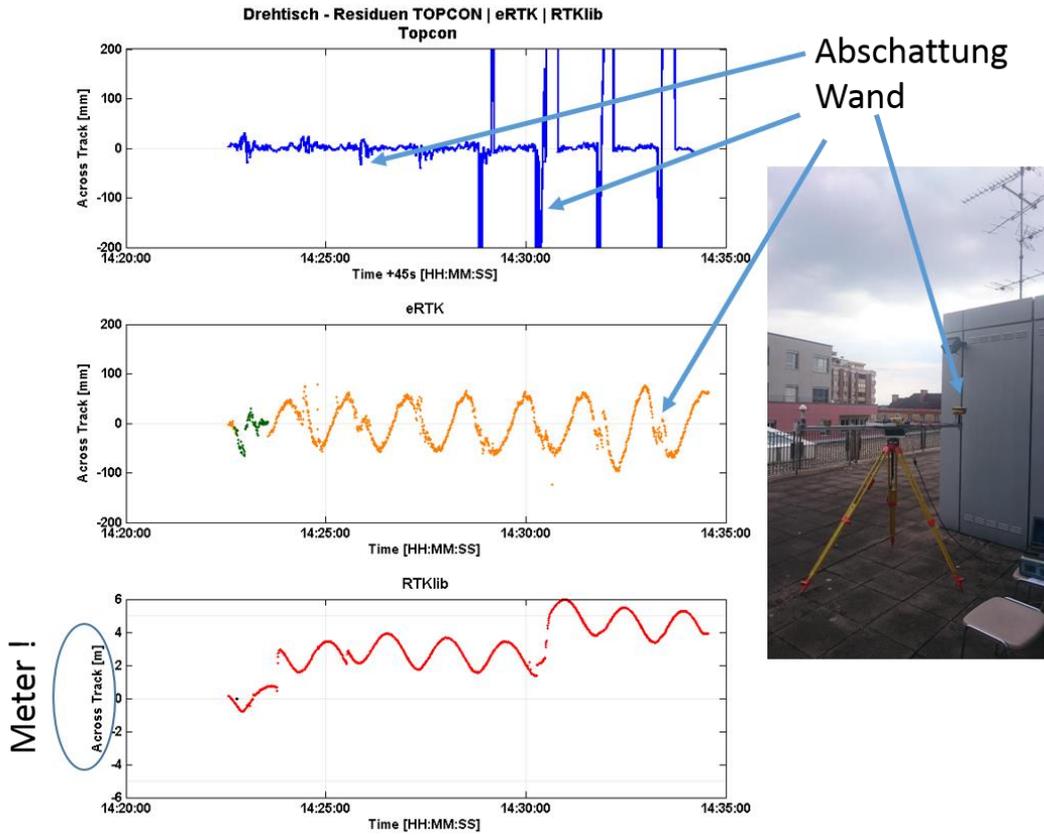


Abbildung 8.24: Residuen 2D-Position Test 2

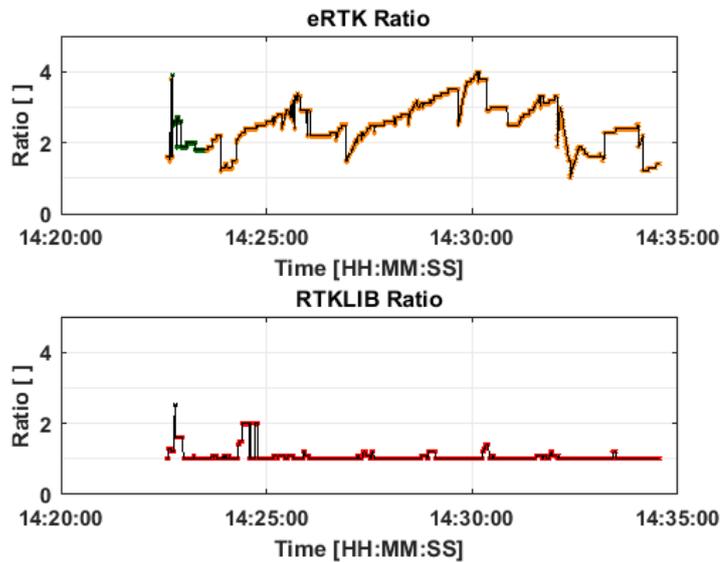


Abbildung 8.25: Ambiguity Ration

Das Ambiguity Ratio zeigt vor allem zu Zeitpunkten im stark abgeschatteten Bereich, einen Abfall an. Außerdem kann gezeigt werden, dass das Ratio für eine Fixierung zu gering ist, was die Float Lösung in beiden Varianten erklärt.

8.4.2 *Fazit*

Der Testaufbau unter erschwerten Messbedingungen zeigt, dass (i) die entwickelte low-cost Systemarchitektur um eRTK eine stabile und präzise Positionslösung liefert, obgleich auch keine optimale Fixierung zustande kommt, dass (ii) die high-cost RTK Lösung von Topcon schlechtere Ergebnisse liefert als die low-cost Systemarchitektur und dass (iii) eine spezielle Prozessierung aller Daten im low-cost Bereich essentiell für eine stabile Positionslösung ist.

Kapitel 9

Zusammenfassung und Ausblick

Ziel der vorliegenden Arbeit war es, eine eigene Systemarchitektur für ein low-cost GNSS-RTK System zu entwickeln und zu evaluieren.

Dazu wurde mithilfe eines eigens definierten Datenformates GOBS, für den GNSS-Datenaustausch, die RTK-Prozessierung im low-cost Bereich im Vergleich mit herkömmlichen Formaten wie RTCM oder RINEX deutlich stabilisiert und verbessert. Es konnte gezeigt werden, dass zusätzliche Qualitätsindikatoren, für die in RTK unabdingbare Phasenmessung von essentieller Bedeutung sind. Diese Qualitätsindikatoren kommen jedoch in herkömmlichen Datenformaten und Prozessierungsalgorithmen nicht zur Anwendung und erschweren hier eine stabile Positionslösung mit low-cost GNSS. Durch Weiterentwicklungen und Implementation der Systemarchitektur bzw. des GOBS-Formates in das vorhandene Softwarepaket RTKLIB entstand eine eigenständige Echtzeit-Positionierungssoftware eRTK. Somit konnte die Entwicklung in Echtzeit getestet und mit der herkömmlichen Lösung verglichen werden. Durch eigene Testmethoden konnten die Vorteile von eRTK gegenüber herkömmlichen Methoden gezeigt werden, wobei alles in Echtzeit durchgeführt wurde. Als Referenz wurde auch eine high-cost Lösung eingesetzt. Unter erschwerten Messbedingungen konnte auch diese Lösung nicht die gewünschten Ergebnisse erzielen. Einzig der eRTK-Entwicklung gelang es, eine stabile Positionslösung zu garantieren.

Durch die Verwendung aller zur Verfügung stehenden Messdaten entstehen jedoch auch Einschränkungen, wie (i) eine Erhöhung der Übertragungsdatensmenge und eine (ii) Abhängigkeit an die einzelnen Empfängerhersteller. Denn jeder Empfänger, auch unterschiedliche Modelle derselben Herstellerfirma, besitzen unterschiedliche Datenformate und müssen jeweils einzeln dekodiert und implementiert werden, was einen deutlichen Mehraufwand aufzeigt.

Ausblick

Das in dieser Arbeit bereits mitentwickelte Übertragungskonzept für sonstige low-cost Sensorik, wie INS oder UWB, kann in weiterer Folge für eine integrierte Lösung herangezogen werden und somit die Positionslösung deutlich steigern. Dieser Ansatz ist auch für stark anwachsende RPAS Anwendungen von großer Bedeutung, da GNSS-Positionen alleine keine Aussage über die Ausrichtung des Objektes im Raum geben.

Für Echtzeitanwendungen im RTK-Modus wird jedoch zumeist eine Abhängigkeit vom mobilen Internetempfang gegeben sein, welche für die Datenübertragung von Basis- und Roverdaten verantwortlich ist. Hier könnten etwaige PPP-Methoden oder sonstige Übertragungskonzepte für RTK getestet werden.

Auch das stetig wachsende europäische Satellitensystem Galileo kann als eigenständiges GNSS-System betrachtet werden und zur Prozessierung herangezogen werden, was jedoch auch in der Arbeit bereits implementiert wurde. Da zurzeit jedoch nur wenige Satelliten für eine Prozessierung zur Verfügung stehen, kann noch keine Aussage über eine Verbesserung durch die Verwendung von Galileo getroffen werden.

In ein paar Jahren jedoch, werden drei bis vier vollwertige GNSS Systeme (GPS, GLONASS, Galileo und Beidou) für die satellitengestützte Positionierung zur Verfügung stehen und für den low-cost Markt deutliche Genauigkeitssteigerungen für die Positionslösungen hervorrufen.

Literaturverzeichnis

Berglez, Phillip. 2013. *Development of a multi-frequency software-based GNSS receiver*. Graz : *Doctoral Thesis*, Technische Universität Graz, 2013.

BKG. 2013. *BKG Ntrip Client*. Frankfurt : Federal Agency for Cartography and Geodesy (BKG), 2013.

Boore, Kai, et al. 2007. *A Software-Defined GPS*. Boston : Birkhäuser, 2007.

Takasu, Tomoji und Yasuda, Akio. 2008 *Cycle Slip Detection and Fixing by MEMS-IMU/GPS Integration for Mobile Environment RTK-GPS*. Savannah, USA, 2008. ION GNSS 2008. S. 8.

embarcadero 2016. 10.15.2016.

<https://www.embarcadero.com/app-development-tools-store/rad-studio>.

10.15.2016.

ESA, Navipedia. 22.09.2016.

http://www.navipedia.net/index.php/Generic_Receiver_Description. s.l. : ESA, 22.09.2016.

Global Positioning System Directorate. 2013. *Interface Specification IS-GPS-200H*. US : Global Positioning System Directorate, 2013.

Opitz, Michael und Weber, Robert. 2005 *GNSS-Echtzeitorbitkontrolle auf Basis Internet-transferierter (NTRIP) RTCM-Datenströme*. 2005, VGI – Österreichische Zeitschrift für Vermessung und Geoinformation, S. 123-131.

Hofmann-Wellenhof, Bernhard, Lichtenegger, Herbert und Collins, James. 2001. *GPS Theory and Practice, Fifth Edition*. Wien : Springer, 2001.

Hofmann-Wellenhof, Bernhard, Lichtenegger, Herbert und Wasle, Elmar. 2008. *GNSS - Global Navigation Satellite Systems*. Wien : Springer, 2008.

Hofmann-Wellenhof, Bernhard, Wieser, Manfred und Legat, Klaus. 2003. *Navigation - Principles of Positioning and Guidance*. Wien : Springer-Verlag Wien New York, 2003.

Huber, Katrin. 2015. *Precise Point Positioning with Ambiguity Resolution for real-time applications, Doctoral Thesis*. Graz : Technische Universität Graz, 2015.

IGS. 2013. *RINEX Version 3.02*. Bern, Boulder : IGS, 2013.

- Kaplan, Elliot D. 2006. *Understanding GPS, principles and applications*. Boston : Artech House, 2006.
- Koch, Daniel. 2012. *Softwareentwicklung eines low-cost GNSS Vessel Navigation Systems*. Graz :Diplomarbeit, Technische Universität Graz, 2012.
- Koch, Daniel, Kriegl, Patrick und Dorn, Markus. 2015. *Low-cost but Robust and High-Precision RTK-Navigation*. Darmstadt : CERGAL, 2015.
- Kriegl, Patrick und Koch, Daniel. 2015. *Erweitertes GNSS-Rohdatenformat für low-cost GNSS-Empfänger*. Wildhaus, Switzerland : AHORN 2015, 2015.
- Meinel, Christoph und Sack, Harald. 2013. *Internetworking*. Berlin, Hiedelberg : Springer, 2013.
- Moder, Thomas. 2011. *Wesen und Nutzen inertialer MEMS*. Graz, Diplomarbeit : Technische Universität Graz, 2011.
- Peterzon, Martin. 2004. *Distribution of GPS-data via Internet*. Gävle : Reports in Geodesy and Geographical Information Systems, 2004.
- Realini, Eugenio. 2009. *goGPS free and constrained relative kinematic positioning with low cost receivers* . Como : *Doctoral Thesis* POLITECNICO DI MILANO , 2009.
- RTCM SPECIAL COMMITTEE NO. 104 . *Ntrip Networked Transport of RTCM via Internet Protocol*. Arlington : Radio Technical Commission For Maritime Services.
- RTCM 2006. *RTCM STANDARD 10403.1*. Arlington : Radio Technical Commission For Maritime Services, 2006.
- RTO of NATO. 2011. *Low-Cost Navigation Sensors*. s.l. : RTO/NATO, 2011.
- Takasu, T. 2013. *RTKLIB ver. 2.4.2. Manual*. s.l. : Takasu, T., 2013.
- Takasu, Tomoji. 2007-2013. RTKLIB. [Online] 2007-2013. [Zitat vom: 29. 09 2016.] <http://www.RTKLIB.com/>.
- Takasu, Tomoji und Yasuda, Akio. 2005. *Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB*. Tokyo : Laboratory of Satellite Navigation, Tokyo University of Marine Science and Technology, 2005.
- Teunissen, Peter J.G. und Kleusberg, Alfred. 1996. *GPS or Geodesy*. Berlin, Heidelberg : Springer, 1996.
- Tiberius, C.C.J.M, Teunissen, P.J.G. und De Jonge, P.J. 1997. *Performance of the LAMBDA method for Fast GPS Ambiguity Resolution*. Delft : Delft University, 1997.

- Tiberius, Christian und de Jonge, Paul. 1996. *The LAMBDA method for integer ambiguity estimation: implementation aspects*. Delft : TU Delft, 1996.
- Titterton, David und Weston, John. 2004. *Strapdown inertial navigation technology*. Reston, Va. : American Inst. of Aeronautics and Astronautics, 2004.
- u-blox AG. 2014. *u-blox M8 Receiver Description*. Thalwill : u-blox AG, 2014.
- Wendel, Jan. 2011. *Integrierte Navigationssysteme*. München : Oldenbourg Wissenschaftsverlag, 2011.
- Wieser, Andreas. 2001. *Robust and fuzzy techniques for parameter estimation and quality assessment in GPS*. Graz, Achen : Shaker Verlag, 2001.
- Xu, Guochang. 2003. *GPS, theory, algorithms and applications*. Berlin : Springer, 2003.
- Zidan, Nesreen I. 2006. *GNSS Receivers For Weak Signals*. Norwood : Artech House Inc., 2006.
- Zogg, Jean-Marie. 2002. *GPS Basics*. Thalwil : u-blox AG, 2002.

Abbildungsverzeichnis

Abbildung 2.1: Aufbau einer IMU	4
Abbildung 2.2: Überblick inertielle Sensoren mit Fehlergrößen	5
Abbildung 2.3: Receiverdesign.....	6
Abbildung 2.4: Receiverdesign Alternative.....	7
Abbildung 2.5. DSP Prinzip	8
Abbildung 2.6: high-cost Empfänger	9
Abbildung 2.7: u-blox Evaluation Kit und u-blox Chip.....	10
Abbildung 2.8: u-blox M8T GNSS-Kombinationen	11
Abbildung 2.9: Änderung des GNSS-Empfängers	13
Abbildung 2.10: Navio 2: Aufbau.....	13
Abbildung 2.11: Systemarchitektur Navio	14
Abbildung 3.1: GNSS Systeme	15
Abbildung 3.2: Prinzip der Satellitenpositionierung	16
Abbildung 3.3: UBX-Protokollstruktur.....	20
Abbildung 3.4: RTCM Header	24
Abbildung 3.5: RTCM RTK-Messages	25
Abbildung 3.6: Struktur von GOBS.....	26
Abbildung 3.7: Cycle Slip	27
Abbildung 4.1: NTRIP-Struktur.....	33
Abbildung 5.1: Prinzip der relativen Positionierung	36
Abbildung 5.2: Geometr. Interpretation der Phasenmessung	39
Abbildung 5.3: ECEF, kartesisch und ellipsoidisch	43
Abbildung 5.4: Keplerbahnparameter.....	45
Abbildung 5.5: Lokales Horizontsystem	46

Abbildung 6.1: RTKLIB Menüführung.....	48
Abbildung 6.2: DD-Prinzip.....	52
Abbildung 7.1: eRTK-Systemarchitektur	59
Abbildung 7.2: RTKLIB-Systemarchitektur.....	60
Abbildung 7.3: eRTK Menü	60
Abbildung 7.4: GOBS-eRTK	61
Abbildung 7.5: RTKCONV-GOBS	62
Abbildung 7.6: GOBS LLI - RINEX LLI.....	63
Abbildung 7.7: Static-Start – sdu.....	66
Abbildung 7.8: Static-Start Feature	66
Abbildung 7.9:Gewichtung	69
Abbildung 7.10: Darstellung eines Cycle Slip.....	70
Abbildung 7.11: Cycle Slip Anaylse eRTK	71
Abbildung 8.1: Messkonzept	73
Abbildung 8.2: Skizze Basisstationen, Dach Geodäsiegebäude.....	74
Abbildung 8.3: Skizze Höhentest	75
Abbildung 8.4: Messergebnisse Höhentest.....	76
Abbildung 8.5: Standardabweichung der eRTK Höhenkomponente	77
Abbildung 8.6: Locktime+LLI R12, Höhentest.....	78
Abbildung 8.7: SNR+STD R12, Höhentest.....	79
Abbildung 8.8 Gewichte R12, Höhentest	79
Abbildung 8.9: Messaufbau 2D-Position Test 1	81
Abbildung 8.10 Positionslösung eRTK, Lagetest 1	82
Abbildung 8.11 Positionslösung RTKLIB, 2D-Position Test 1	82
Abbildung 8.12: Punktlagefehler, 2D-Position Test 1	83
Abbildung 8.13 Trajektorie 1 Runde, 2D-Position Test 1	83

Abbildung 8.14: Residuen eine Runde, 2D-Position Test 1	84
Abbildung 8.15: eRTK Rohdaten, 2D-Position Test 1	85
Abbildung 8.16: RTKLIB Rohdaten, 2D-Position Test 1	85
Abbildung 8.17: eRTK Rohdaten 2, Lagetest 1	86
Abbildung 8.18 Gewichte R13, 2D-Position Test 1	86
Abbildung 8.19: Aufbau Multipath	88
Abbildung 8.20: Systemaufbau 2D-Position Test 2	88
Abbildung 8.21 Positionslösung Gesamt, 2D-Position Test 2	89
Abbildung 8.22: Positionslösung zwei Runden, 2D-Position Test 2	90
Abbildung 8.23: Punktlagefehler 2 Runden, Lagetest 2	91
Abbildung 8.24: Residuen 2D-Position Test 2.....	92
Abbildung 8.25: Ambiguity Ration	92

Tabellenverzeichnis

Tabelle 2-1: Receiverdesign.....	6
Tabelle 2-2: u-blox M8T	11
Tabelle 2-3: Navio Spezifikationen.....	12
Tabelle 3-1: UBX-Class IDs	20
Tabelle 3-2: UBX-Messages	21
Tabelle 3-3: UBX RXM-RAWX.....	21
Tabelle 3-4: RINEX Files.....	23
Tabelle 3-5: RTCM Versions	24
Tabelle 3-6: RTCM 3.0, RTK Messages	25
Tabelle 3-7: LLI Spezifikation	28
Tabelle 3-8: Formate Fazit	30
Tabelle 4-1: TCP/IP Layer.....	32
Tabelle 4-2: Übertragungsanalyse	35
Tabelle 5-1: GNSS-Einheiten.....	37
Tabelle 5-2: Keplerparameter.....	45
Tabelle 8-1: GNSS-Messaufbau	73
Tabelle 8-2: Pfeilerkoordinaten INAS, ITRF2008 Epoche 2012/07/18	74

Anhang A

Anhang A beschreibt das im Zuge von Projektarbeiten am Institut für Geodäsie entstandene Datenformat GOBS auf den Seiten 103 bis 115.

FORMAT GNSSObs
 SUFFIX .gobs
 VERSION 2.0 KP
 26.09.2016

Block	Column	Type	Information	Unit	Format
1	1	GPS_week	[www]	-	[] %4i
1	2*	GPS_seconds	[sssss.fff]	[0 < GPS_seconds < 604800]	[s] %6.3f
1	3*	PRN	[PP]	[0 < PRN < 34]	[] %02i
1	4*	Type	[T]	Rover=1; Base=2; otherwise=0;	[] %1i
GPS***=1; GAL***=3; COM***=5;			[]	%4i	
1	5*	GNSS	[G]	GLO***=2; QZS***=4;	
1	6*	Lock Time	[sssss]	[0 < Lock < max(64500ms)]	[ms] %5i
1	7*	LLI**	[L]	[0 < LLI < 1]	[] %1i
1	8*	Half-CC Valid	[c]	[0 < HCC < 1]	[] %1i
1	9*	CC-Valid	[c]	[0 < CC-phase < 1]	[] %1i

* Use a BLANK before (see example)
 ** Note: If this data is not available (on L1) at all, replace it by NaN.
 *** GPS; GLO=Glonass; GAL=Galileo; QZSS=QZS; Compass=COM;

Block	Column	Type	Information	Unit	Format
2	10*	OBS C1	[ccccccccc.ccc]	-	[m] %15.3f
2	12*	OBS L1	[111111111.111]	-	[cycles] %15.3f
2	14*	OBS D1	[ddddddddd.ddd]	-	[Hz] %15.3f
2	16*	OBS S1	[ss.sss]	-	[dBHz] %2.3f
2	18*	Sigma_C1**	[cccc.ccc]	-	[m] %6.3f
2	19*	Sigma_L1**	[cccc.ccc]	-Bit 1 to 15	[cycles] %6.3f
2	20*	Sigma_D1**	[cccc.ccc]	-	[Hz] %6.3f
2	21*	PR_valid**	[v]	pseudorange valid if value 1	[] %1i
2	22*	CP_valid**	[v]	carrier phase valid if value 2	[] %1i

* Use a BLANK before (see example)

** Note: If this data is not available (on L1) at all, replace it by NaN.

*** GPS; GLO=GLONASS; GAL=Galileo; QZSS=QZS; Compass=COM;

Block	Column	Type	Information	Unit	Format
3	23*	LS*	-	Leap Seconds (UTC-GPST)	[s] %5i
3	24*	Az*	-	Azimuth of observed SV	[°] %10.3f
3	25*	El*	-	Elevation of observed SV	[°] %10.3f
3	26*	rX*	-	received rcv (ublox) position X	[m] %15.5f
3	27*	rY*	-	received rcv (ublox) position Y	[m] %15.5f
3	28*	rZ*	-	received rcv (ublox) position Z	[m] %15.5f
3	29*	sR*	-	rcv (ublox) position acc.	[m] %8.3f
3	30*	c1B*	-	clock (ublox) bias	[ns] %8.3f
3	31*	c1D*	-	clock (ublox) drift	[ns] %8.3f

* Use a BLANK before (see example)

** Note: If this data is not available (on L1) at all, replace it by NaN.

*** GPS; GLO=GLONASS; GAL=Galileo; QZSS=QZS; Compass=COM;

FORMAT GNSSNav
 SUFFIX .gnav
 VERSION 2.0
 14.04.2016

GPS - NAVDATA BLOCK 0

Block	Column	Type	Information	Unit	Format
0	1*	GNSS (identifier	[G]	[]	%01i
0	2*	PRN	[PP]	[]	%02i
0	3*	Year (ToC)	[YYYY]	[]	%6i
0	4*	Month (ToC)	[MM]	[]	%02i
0	5*	Day (ToC)	[DD]	[]	%02i
0	6*	Hour (ToC)	[hh]	[]	%02i
0	7*	Minute (ToC)	[mm]	[]	%02i
0	8*	Second (ToC)	[sssss.fff]	[s]	%7.3f
0	9*	Clock Bias	[-]	[s]	%25.13e
0	10*	Clock Drift	[-]	[s/s]	%25.13e
0	11*	Clock Drift Rate	[-]	[s/s ²]	%25.13e

* Use a BLANK before (see example)

GPS - NAVDATA BLOCK I

Block	Column	Type	Information	Unit	Format
1	12*	IODE	Issue of Data, Ephemeris	[]	%25.13e
1	13*	CRS		[m]	%25.13e
1	14*	Δn		[rad/sec]	%25.13e
1	15*	M0		[rad]	%25.13e

* Use a BLANK before (see example)

GPS - NAVDATA BLOCK II

Block	Column	Type	Information	Unit	Format
2	16*	CUC		[rad]	%25.13e
2	17*	e	Eccentricity	[]	%25.13e
2	18*	CUS		[rad/sec]	%25.13e
2	19*	sqrt(a)		[sqrt(m)]	%25.13e

* Use a BLANK before (see example)

GPS - NAVDATA BLOCK III

Block	Column	Type	Information	Unit	Format
3	20*	Seconds (ToE)	[]	[sec of GPS]	%25i
3	21*	CIC	[]	[rad]	%25.13e
3	22*	OMEGA	[]	[rad]	%25.13e
3	23*	CIS	[]	[rad]	%25.13e

* Use a BLANK before (see example)

GPS - NAVDATA BLOCK IV

Block	Column	Type	Information	Unit	Format
4	24*	i0	[]	[rad]	%25.13e
4	25*	CRC	[]	[m]	%25.13e
4	26*	omega	[]	[rad]	%25.13e
4	27*	OMEGA DOT	[]	[rad/sec]	%25.13e

* Use a BLANK before (see example)

GPS - NAVDATA BLOCK V

Block	Column	Type	Information	Unit	Format
5	28*	IDOT	[]	[rad/sec]	%25.13e
5	29*	Codes on L2 channel	[]	[rad]	%25.13e
5	30*	Week (ToE)	[]	[]	%25i
5	31*	L2 P data flag	[]	[]	%25.13e

* Use a BLANK before (see example)

GPS - NAVDATA BLOCK VI

Block	Column	Type	Information	Unit	Format
6	32*	SV accuracy	[]	[m]	%25.13e
6	33*	SV health	[]	[MSB only]	%25.13e
6	34*	TGD	[]	[sec]	%25.13e
6	35*	IODC Issue of Data, Clock	[]	[]	%25.13e
6	36*	IODE3	[]	[]	%25.13e

* Use a BLANK before (see example)

GPS EXAMPLE (Structure of .gobs)

```
BLOCK 0 >> %01i %02i %6i %02i %02i %02i %02i %7.3f %25.13e %25.13e %25.13e
BLOCK 1 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 2 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 3 >> %25.13e %25.13e %25.13e %25i
BLOCK 4 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 5 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 6 >> %25.13e %25.13e %25.13e %25.13e %25.13e
```

GLONASS - NAVDATA BLOCK 0

Block	Column	Type	Information	Unit	Format
0	1*	GNSS (identifier	[G]	[]	%01i
0	2*	PRN	[PP]	[]	%02i
0	3*	Year (ToC)	[YYYY]	[]	%6i
0	4*	Month (ToC)	[MM]	[]	%02i
0	5*	Day (ToC)	[DD]	[]	%02i
0	6*	Hour (ToC)	[hh]	[]	%02i
0	7*	Minute (ToC)	[mm]	[]	%02i
0	8*	Second (ToC)	[sssss.fff]	[s]	%7.3f
0	9*	Clock Bias	[-]	[s]	%25.13e
0	10*	Relative Fq. Bias	[-]	[s/s]	%25.13e
0	11*	Message Frame Time	[-]	[s/s ²]	%25.13e

* Use a BLANK before (see example)

GLONASS - NAVDATA BLOCK I

Block	Column	Type	Information	Unit	Format
1	12*	Sat.Pos. X	[]	[Km]	%25.13e
1	13*	Sat.Vel. X	[]	[Km/sec]	%25.13e
1	14*	Sat.Acc. X	[]	[Km/sec ²]	%25.13e
1	15*	Sat.Health	[]	[]	%25.13e

* Use a BLANK before (see example)

GLONASS - NAVDATA BLOCK II

Block	Column	Type	Information	Unit	Format
2	16*	Sat.Pos. Y	[]	[Km]	%25.13e
2	17*	Sat.Vel. Y	[]	[Km/sec]	%25.13e
2	18*	Sat.Acc. Y	[]	[Km/sec ²]	%25.13e
2	19*	Frequency number	[]	[]	%25.13e

* Use a BLANK before (see example)

GLONASS - NAVDATA BLOCK III

Block	Column	Type	Information	Unit	Format
3	20*	Sat.Pos. Z	[]	[Km]	%25.13e
3	21*	Sat.Vel. Z	[]	[Km/sec]	%25.13e
3	22*	Sat.Acc. Z	[]	[Km/sec ²]	%25.13e
3	23*	Age of oper. information(days)	[]	[]	%25i

* Use a BLANK before (see example)

GLONASS - NAVDATA BLOCK VI

Block	Column	Type	Information	Unit	Format
3	24*	GLO IOD	reserved for processing	[]	0
3	25*	GLO TOE WEEK	reserved column for processing	[]	0
3	26*	GLO TOE SEC	reserved column for processing	[]	0

* Use a BLANK before (see example)

GLONASS - NAVDATA BLOCK V

Block	Column	Type	Information	Unit	Format
3	27*	SPARE	[NaN]	[]	%25.13e
3	28*	SPARE	[NaN]	[]	%25.13e
3	29*	SPARE	[NaN]	[]	%25.13e

* Use a BLANK before (see example)

GLONASS - NAVDATA BLOCK IV

Block	Column	Type	Information	Unit	Format
3	30*	SPARE	[NaN]	[]	%25.13e
3	31*	SPARE	[NaN]	[]	%25.13e
3	32*	SPARE	[NaN]	[]	%25.13e
3	33*	SPARE	[NaN]	[]	%25.13e
3	34*	SPARE	[NaN]	[]	%25.13e

* Use a BLANK before (see example)

GLONASS EXAMPLE (Structure of .gobs)

```
BLOCK 0 >> %01i %02i %6i %02i %02i %02i %02i %7.3f %25.13e %25.13e %25.13e
BLOCK 1 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 2 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 3 >> %25.13e %25.13e %25.13e %25i
BLOCK 4 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 5 >> %25.13e %25.13e %25.13e %25.13e
BLOCK 6 >> %25.13e %25.13e %25.13e %25.13e %25.13e
```