



Stefan Laller, BSc.

Single Point Positioning with Android Raw Data on a Smartphone

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Geomatics Science

submitted to

Graz University of Technology

Supervisor

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

Institute of Geodesy

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Acknowledgements

First of all, I would like to thank my supervisor Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Manfred Wieser for his excellent support with this master thesis. Despite his stressful job, he always had time for me whenever I knocked on his door.

My thanks also go out to Dipl.-Ing Thomas Moder and Dipl.-Ing Roman Wilfinger for their valuable assistance. They always had an open ear for questions and saved me a lot of time with useful tips.

Special thanks to all my student colleagues for a great time and their support. I would like to thank Uroš Bokan, Mathias Duregger, Gernot Kainz, and Maximilian Schachner-Nedherer in particular for their great friendship and for always being there for me.

Furthermore, I would like to thank Mag. Petra Horvath for proofreading this master thesis.

Last but not least, I have to thank my parents and my sister for their support throughout my education. Thank you for giving me the opportunity for such an education.

Above all, I would like to thank my father for his mathematical expertise and his useful advice during my studies. Thank you for so many exam preparations when you spent a lot of time studying with me. Without you, all of this would not have been possible. Thanks for everything you taught me.

In memory of my father.

Graz, October 2018

Stefan Laller

Abstract

Today, Global Navigation Satellite Systems (GNSS) are an indispensable tool for determining the position of objects. Looking to the future, one can already imagine that these will become of ever-increasing importance. Not only in geodetic surveying, but also in everyday life, they are indispensable. With these navigation satellite systems, easy positioning, such as of a vehicle or a smartphone, is possible in a short amount of time.

This master thesis describes a satellite-based positioning of a smartphone. The year 2016 is considered a historic milestone, because Google has been providing the raw data of the built-in GNSS chip directly from the Android version 7.0 (Nougat) since May 2016. This is considered a revolution in the development and improvement of the positional accuracy of mobile phones. Access to the raw data offers the user a multitude of possibilities. Thus, a wide variety of algorithms can be performed, such as Precise Point Positioning (PPP) or Relative Positioning between two devices. Moreover, an integration with inertial sensors is possible. Even a multi-GNSS evaluation on the smartphone can be realized much more easily using the raw data. Thus, several Global Navigation Satellite Systems can be utilized, and individual satellites can be sorted out.

A significant part of this master thesis deals with the investigation and analysis of Android raw data. In order to achieve this, they are compared from several different smartphones.

In addition, an Android app has been developed that uses real-time Android raw data and broadcast ephemerides to compute a Single Point Positioning (SPP) based on GPS.

Kurzfassung

Globale Satellitennavigationssysteme (GNSS) stellen heutzutage ein unumgängliches Werkzeug für die Positionsbestimmung von Objekten dar. Blickt man in die Zukunft, lässt sich bereits erahnen, dass diese einen immer höheren Stellenwert einnehmen werden. Nicht nur bei geodätischen Vermessungen, sondern auch im alltäglichen Leben sind diese nicht mehr wegzudenken. Mithilfe dieser Satellitennavigationssysteme ist eine einfache Positionierung, beispielsweise eines Fahrzeuges oder eines Smartphones, in kurzer Zeit möglich.

In dieser Masterarbeit wird eine satelliten-basierende Positionierung eines Smartphones erläutert. Das Jahr 2016 gilt als historischer Meilenstein, denn seit Mai 2016, stellt Google direkt die Rohdaten des eingebauten GNSS Chips ab der Android Version 7.0 (Nougat) zur Verfügung. Dies gilt als Revolution in der Entwicklung und Verbesserung der Positionsgenauigkeit von Mobiltelefonen. Der Zugriff auf die Rohdaten bietet dem Nutzer eine Vielzahl von Möglichkeiten. So können unterschiedlichste Algorithmen, wie beispielsweise eine präzise Einzelpunktbestimmung (engl. Precise Point Positioning (PPP)) oder eine relative Positionierung zwischen zwei Geräten durchgeführt werden. Weiters ist eine Integration mit Trägheitssensoren möglich. Aber auch eine Multi-GNSS Auswertung am Smartphone kann mithilfe der Rohdaten besser realisiert werden. Somit können mehrere globale Satellitennavigationssysteme verwendet und einzelne Satelliten aussortiert werden.

Ein großer Teil der Masterarbeit befasst sich mit der Untersuchung und der Analyse der Android Rohdaten. Hierfür werden diese von mehreren unterschiedlichen Smartphones miteinander verglichen.

Zusätzlich wurde eine Android App entwickelt, welche in Echtzeit mithilfe dieser Android Rohdaten und den Broadcast Ephemeriden eine GPS-basierende Einzelpunktbestimmung (engl. Single Point Positioning (SPP)) berechnet.

Contents

List of Figures	X
List of Tables	XI
1 Introduction	1
1.1 Motivation	1
1.2 Objective	1
1.3 Structure	2
2 Basic Concept	3
2.1 Introduction	3
2.2 GPS Architecture	4
2.2.1 Control Segment	4
2.2.2 Space Segment	5
2.2.3 User Segment	5
2.3 Coordinate Systems	5
2.3.1 ECEF System	6
2.4 Time Systems	10
2.4.1 UTC Time	10
2.4.2 GPS Time	10
2.5 Single Point Positioning	11
2.5.1 Basic Principle of SPP	11
2.5.2 Observation Model	12
2.5.3 Receiver Clock Error	13
2.5.4 Satellite Clock Error	13
2.5.5 Relativistic Effect	15
2.5.6 Time Group Delay (TGD)	15
2.5.7 Atmospheric Effect	16
2.5.8 Global Positioning System (GPS) Broadcast Ephemerides	23
2.5.9 Satellite Positions	25

2.5.10	Least-squares Adjustment	26
2.5.11	Position Accuracy	28
3	Android Raw Measurements	30
3.1	Introduction	30
3.2	Possible Mobile Devices	31
3.3	Used Test Equipment	32
3.3.1	Huawei Honor 8	32
3.3.2	Huawei P10	32
3.3.3	Samsung S8 (Exynos)	33
3.3.4	Smartphone Global Navigation Satellite System (GNSS) Chip - Broadcom BCM4774	33
3.4	Structure	34
3.5	GNSS Analysis Tool	39
4	Data Processing with Android	43
4.1	Introduction	43
4.2	Single Point Positioning (SPP)	44
4.2.1	Pseudorange Generation	44
4.2.2	Data Filtering	45
4.2.3	Weighting	46
4.2.4	Implemented Algorithm	48
4.3	Android Application	51
4.3.1	Software Structure	51
4.3.2	Software Design	54
5	Results and Evaluation	57
5.1	Test Measurement	58
5.1.1	Carrier-to-noise Density and Elevation	60
5.1.2	Position Solution	61
5.1.3	Comparison to Other Tools	65
5.2	Multiple Test Measurements	66
5.3	Android Raw Data Analysis and Comparison	69
5.3.1	Number of Satellites	70
5.3.2	Data Filtering	71
5.3.3	Carrier-to-noise Density (C/N_0)	72
5.3.4	Single Point Positioning	73
5.4	Comparison of Multiple Test Measurements	76
5.5	SPP Real Time Solution on Android	77

6 Conclusion and Future Tasks	81
A Appendix	83
A.1 Constants	83
A.2 Reference Coordinates	84
Bibliography	85

List of Figures

2.1	Cartesian and ellipsoidal coordinates [cf. 9, p. 278, Fig. 8.1]	6
2.2	Gauß-Krüger projection © BEV 15.05.2015 [12, Abbildung 10]	8
2.3	Minimal GNSS constellation	11
2.4	Atmospheric layers	16
2.5	Ray path through the atmosphere [cf. 17, p. 109, Fig. 5.10]	17
2.6	GPS broadcast ephemerides - header	23
3.1	Start window - “GNSS Analysis Tool“	39
3.2	“GNSS Analysis Tool“ - part 1	40
3.3	“GNSS Analysis Tool“ - part 2	41
3.4	“GNSS Analysis Tool“ - part 3	42
4.1	App icon	51
4.2	Program structure - flowchart	53
4.3	App - screenshots	54
4.4	App - measuring	55
4.5	Output file - position data	56
4.6	Output file - raw data	56
5.1	Measurement situation	58
5.2	Number of visible satellites	59
5.3	Android raw data	60
5.4	Comparison of all options	61
5.5	Position of all epochs	63
5.6	Time series of the difference to the reference point	64
5.7	Comparison to other tools - mean position	65
5.8	Comparison with all evaluation tools - time series	66
5.9	Measurement situation - comparison	69
5.10	Comparison of the visible GPS satellites	70
5.11	Data filtering - amount of satellites	71
5.12	Comparison - Carrier-to-noise density	72

5.13	Mean position of all three smartphones	73
5.14	Comparison - mean position other tools	75
5.15	App - measuring	77
5.16	Real time - positions	78
5.17	Real time - height	79
5.18	Time series of the formal accuracy	80

List of Tables

2.1	Meteorological parameters for the tropospheric delay: average [cf. 17, p. 124, Table 5.1]	22
2.2	Meteorological parameters for the tropospheric delay: seasonal variations [cf. 17, p. 124, Table 5.1]	23
2.3	GPS broadcast ephemerides - data [cf. 17, p. 57, Table 3.8]	24
3.1	Suitable mobile devices [3]	31
3.2	Constellations	38
5.1	Numeric comparison of all options	62
5.2	Comparison of multiple test measurements	67
5.3	Comparison with other tools - mean position	67
5.4	Comparison with other tools - standard deviation	68
5.5	Comparison of all three smartphones	74
5.6	Comparison of all three smartphones processed with other tools	74
5.7	Comparison of all three smartphones of different measurement campaigns	76
5.8	Measurement campaign real time - numerical results	78
A.1	Constants	83
A.2	Gauß-Krüger projection: parameters [9, p. 287]	83
A.3	Reference coordinates - World Geodetic System 1984 (WGS84)	84

List of Acronyms

API	Application Programming Interface
BIPM	International Bureau of Weights and Measures
DOP	Dilution of Precision
ECEF	Earth Centered Earth Fixed
FCN	Frequency Channel Number
GLONASS	Globalnaja Nawigazionnaja Sputnikowaja Sistema
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPST	GPS Time
IMU	Inertial Measurement Unit
IPP	Ionospheric Pierce Point
LBS	Location-based Service
MEO	Medium Earth Orbit
OSN	Orbital Slot Number
PPP	Precise Point Positioning
PRN	Pseudo-Random Noise
QZSS	Quasi-Zenith Satellite System
SAR	Search and Rescue
SBAS	Satellite-based Augmentation Systems
SPP	Single Point Positioning
TAI	International Atomic Time

TGD	Time Group Delay
TOW	Time of Week
UT	Universal Time
UTC	Universal Time Coordinated
WGS84	World Geodetic System 1984

Chapter 1

Introduction

1.1 Motivation

The number of smartphone users is increasing rapidly from year to year. More and more people are using a mobile device. Not only because they enable constant reachability, but also because of their ever-increasing performance and functions, they are an indispensable part of today's life. There are a variety of applications for smartphones, which would have been unthinkable only a few years ago.

The development of smartphones has not stopped when it comes to smartphone positioning. By virtue of the increasing demands of various applications, the positional accuracy of the smartphone is becoming more and more accurate. Thus, apps which rely on the position of the mobile phone access the position solution of the mobile internal GNSS chip. However, in some applications, such as navigation for blind people, a higher position accuracy than that of the internal chip is required. The ability to access the raw data allows the use of better positioning techniques.

1.2 Objective

The aim of this master thesis is the investigation and the analysis of Android raw measurements. This includes the structure of the data and the parameters which are provided for the user. Moreover, raw measurements from different smartphones are to be compared and tested.

Furthermore, an Android application is to be created which processes the raw data directly from the smartphone. In this application, a Single Point Positioning (SPP) will be implemented. The computation of the smartphone position, which is based only on GPS observations, is to be conducted in real time.

1.3 Structure

This master thesis is divided into a total of seven chapters. The first part, the introduction, deals with the motivation, a short objective, and the structure of the paper.

The second chapter presents the most basic theoretical aspects of satellite-based positioning and other mathematical theory.

The Android raw data are described in detail in chapter three. Moreover, the test equipment used in this thesis will be briefly introduced.

Chapter four deals with GNSS processing in Android and gives an overview of the resulting software and its structure.

The analysis of the raw data and the results of the position determination are presented in chapter five. Furthermore, a comparison of the different smartphones is provided.

The last two chapters of this master thesis provide a conclusion and outlook on future tasks, as well as an attachment, which contains additional information.

Chapter 2

Basic Concept

Satellite-based positioning is the determination of the position of a receiver observing GNSS satellites. In this chapter, an introduction and the mathematical basics of a satellite-based positioning are provided. Furthermore, the general architecture of the GPS is examined in section 2.2.

2.1 Introduction

Global Navigation Satellite Systems (GNSS) are an essential tool nowadays, and especially in the future. The principle usage of GNSS is to determine the position of an object. Consequently, the importance of satellite systems is reflected in numerous applications, for instance navigation systems and surveying systems, as well as positioning applications on smartphones. It is also a free service, which every user can take advantage of, regardless of the weather.

In recent years, the progress in GNSS positioning has been enormous. Not only the quality of the signals has been improving, but their availability is also increasing. One reason for the better performance is that there are more and more satellites in space, which can be used to determine the user's positions.

The American Global Positioning System (GPS), the European system Galileo, or the Russian Globalnaja Nawigazionnaja Sputnikowaja Sistema (GLONASS) are only a few constellations, which dominate the GNSS market.

There exist many different techniques, which are based on satellite positioning of a receiver. One of them is the Single Point Positioning (SPP), which is being examined in this master thesis.

2.2 GPS Architecture

The Global Positioning System (GPS) can be split up into three segments. It consists of the control segment, the space segment, and the user segment. They are described in the following three sections.

2.2.1 Control Segment

The control segment is responsible for controlling the whole system. This part of the GPS architecture consists of one master control station, six monitor stations and four ground antennas.

Master Control Station

The master control station is located in Colorado Springs and is the most important component. This station collects the tracking data from the monitor stations and calculates the parameters of the satellite orbits and the clocks. The calculated parameters are sent to one ground antenna. Then the data is transmitted to the satellites, and the satellites can send their broadcast ephemerides with the navigation message [9, p. 324 f.].

Monitor Stations

The operational control system includes six monitor stations, which have the task to continuously measure pseudoranges of all visible satellites in both frequencies. The smoothed measurement data is then sent to the master control station. The monitor stations are located at Hawaii, Colorado Springs, Ascencion Island (South Atlantic Ocean), Diego Garcia (Indian Ocean), Kwajalein (North Pacific Ocean), and Cape Canaveral (Florida)[9, p. 325].

Ground antennas

The main task of the ground antennas is to transmit data, for instance clock information or ephemerides, to the satellites. Therefore, the most important equipment is the ground antenna. The calculations are performed at the master control station. The four stations are located at Ascencion Island, Diego Garcia, Kwajalein and Cape Canaveral [9, p. 325].

2.2.2 Space Segment

The second part of the GPS architecture is the space segment. It can be divided into the constellation and the satellites.

Constellation

The GPS constellation consists of 24 satellites, and is full operational. There also exist additional spare satellites. They are divided into six orbital planes, which are nearly circular and have an inclination of 55° to the equator. GPS satellites fly in a Medium Earth Orbit (MEO) and have an altitude of about 20 200 *km*. Because of this constellation, it is possible to measure at least four satellites simultaneously [9, p. 322 f.].

Satellites

Every satellite has several high-stability atomic clocks (rubidium, caesium, and hydrogen), computers and a lot of auxiliary equipment on board, for instance power supply, as well as systems to keep and move the satellites in the orbit or to broadcast signals to the receiver [17, p. 9].

2.2.3 User Segment

The last component is the user segment, which consists of the GPS receivers that receive the electromagnetic signals from the satellites to determine the position of the user [17, p. 18].

2.3 Coordinate Systems

One of the most important topics in connection with GNSS positioning is the coordinate system.

The definition of a coordinate system requires an origin, and moreover the direction of at least two axes is necessary. The third axis is orthogonal to the other two axes. The system, which is used for the investigations, is the Earth Centered Earth Fixed (ECEF) system. This coordinate system is suitable for fixing a point on the earth's surface.

2.3.1 ECEF System

This section deals with the definition of the global ECEF coordinate system. It rotates with the earth, and the origin is defined in the center of the earth's mass. The X -axis is in the direction of the Greenwich meridian, and the Z -axis coincides with the earth's mean rotational axis. The Y -axis is orthogonal to the other two axes and is within the equatorial plane so that the coordinate system represents a right-handed system. In Figure 2.1, the cartesian and the ellipsoidal representations of a point are shown:

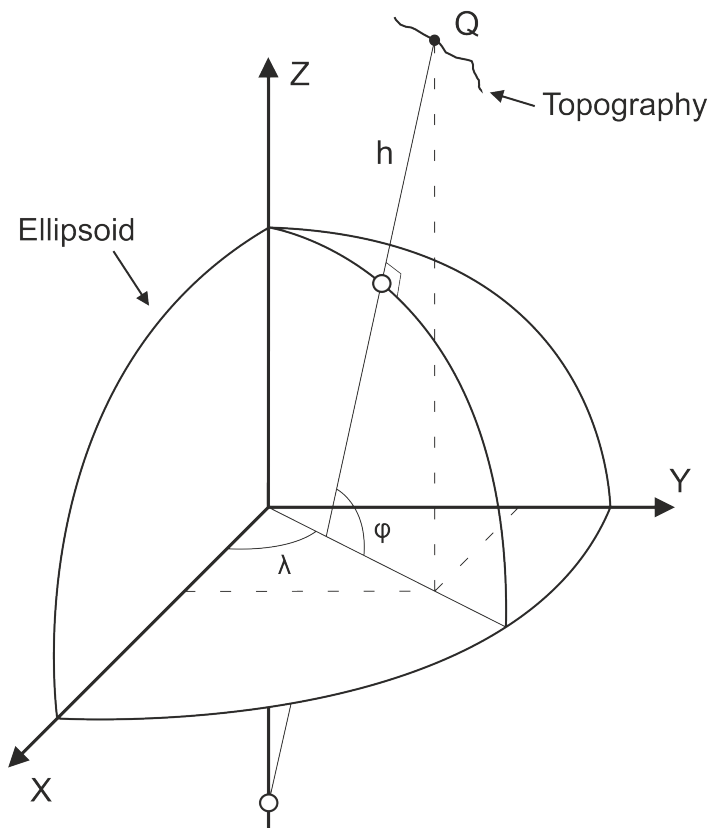


Figure 2.1: Cartesian and ellipsoidal coordinates [cf. 9, p. 278, Fig. 8.1]

The cartesian representation of a point is given by the X - , Y - , and Z -coordinates. The ellipsoidal coordinates are more convenient for civilian usage. The definition of a point in the ellipsoidal coordinate system is achieved with the ellipsoidal latitude φ , the longitude λ , and the height h . For this definition, an ellipsoid has to be specified with the two semiaxis a and b . The center of the ellipsoid must coincide with the origin of the coordinate system, and moreover the reference meridian, respectively the rotational axis of the ellipsoid, has to coincide with the X - and the Z - axis of the cartesian coordinate system.

Transformation: Ellipsoidal \Leftrightarrow Cartesian

In this section, the transformation between the two realisations of the mentioned system is shown. This is one of the most important aspects of geodesy. In [9, p. 277 ff.] both transformations are described in detail. Only the formulas which are necessary are included here.

The transformation from the ellipsoidal coordinates φ , λ , and h to the cartesian coordinates can be done with equation 2.1:

$$\begin{aligned} X &= (N + h) \cos \varphi \cos \lambda \\ Y &= (N + h) \cos \varphi \sin \lambda \\ Z &= \left(\frac{b^2}{a^2} N + h \right) \sin \varphi \end{aligned} \tag{2.1}$$

N is the radius of the ellipsoidal curvature in the prime vertical,

$$N = \frac{a^2}{\sqrt{a^2 \cos^2 \varphi + b^2 \sin^2 \varphi}}. \tag{2.2}$$

The parameters a and b are the semiaxis of the WGS84-ellipsoid. The numeric values of these two parameters are listed in A.1.

The inverse approximate transformation requires more auxiliary quantities, but they are closed formulas to evaluate. The computation of the ellipsoidal coordinates from the cartesian ones can be done with equation 2.3:

$$\begin{aligned} \varphi &= \arctan \frac{Z + e'^2 b \sin^3 \theta}{p - e'^2 a \cos^3 \theta} \\ \lambda &= \arctan \frac{Y}{X} \\ h &= \frac{p}{\cos \varphi} - N \end{aligned} \tag{2.3}$$

with

$$\theta = \arctan \frac{Z a}{p b} \tag{2.4}$$

$$p = \sqrt{X^2 + Y^2} \tag{2.5}$$

and the first and second numerical eccentricity:

$$e^2 = \frac{a^2 - b^2}{a^2} \quad (2.6)$$

$$e'^2 = \frac{a^2 - b^2}{b^2} \quad (2.7)$$

These formulas are only valid if the center of the ellipsoid lies in the origin of the coordinate system. The rotational axis of the ellipsoid has to coincide with the Z -axis, and the reference meridian has to coincide with the Greenwich meridian.

Projection: Gauß-Krüger plane

The Gauß-Krüger projection [9, p. 284 ff.] is a transverse Mercator projection. It consists of 120 zones with 3 degree bands. In the middle of one band, the central meridian λ_0 is projected without scale distortion. Austria is located at the following bands:

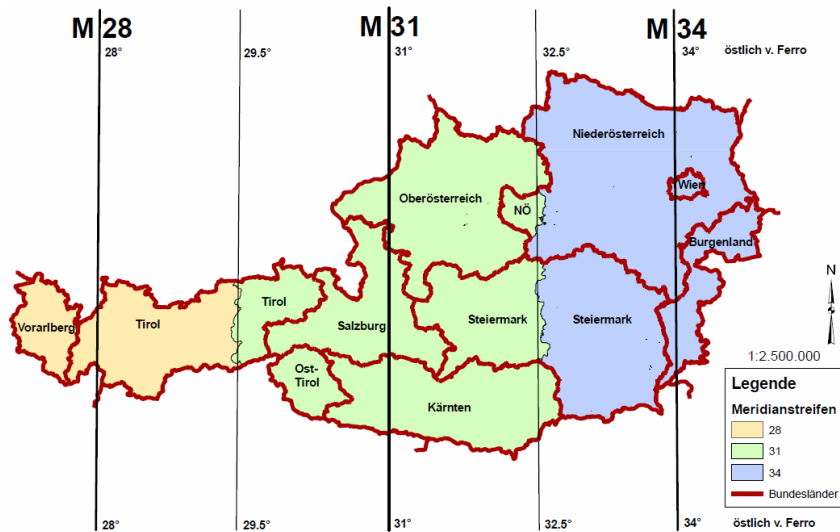


Figure 2.2: Gauß-Krüger projection © BEV 15.05.2015 [12, Abbildung 10]

For the investigations in this master thesis, only M34 is relevant. The formulas which are used for the calculation of the x and y coordinates from the ellipsoidal coordinates φ and λ are presented in equations 2.8 and 2.9. In this master thesis, the designations of x and y are reversed in contrast to [9] so that x is north and y is east.

$$\begin{aligned}
 x &= B(\varphi) + \frac{t}{2}N \cos^2 \varphi l^2 \\
 &+ \frac{t}{24}N \cos^4 \varphi (5 - t^2 + 9\eta^2 + 4\eta^4)l^4 \\
 &+ \frac{t}{720}N \cos^6 \varphi (61 - 58t^2 + t^4 + 270\eta^2 - 330t^2\eta^2)l^6 \\
 &+ \frac{t}{40320}N \cos^8 \varphi (1385 - 3111t^2 + 543t^4 - t^6)l^8 + \dots
 \end{aligned} \tag{2.8}$$

$$\begin{aligned}
 y &= N \cos \varphi l + \frac{1}{6}N \cos^3 \varphi (1 - t^2 + \eta^2)l^3 \\
 &+ \frac{1}{120}N \cos^5 \varphi (5 - 18t^2 + t^4 + 14\eta^2 - 58t^2\eta^2)l^5 \\
 &+ \frac{1}{5040}N \cos^7 \varphi (61 - 479t^2 + 179t^4 - t^6)l^7 + \dots
 \end{aligned} \tag{2.9}$$

with the arc length of the meridian:

$$B(\varphi) = \alpha[\varphi + \beta \sin 2\varphi + \gamma \sin 4\varphi + \delta \sin 6\varphi + \epsilon \sin 8\varphi + \dots] \tag{2.10}$$

and further quantities:

N ... radius of ellipsoidal curvature in prime vertical (see Eq. 2.2)

$\eta = \frac{\sqrt{a^2 - b^2}}{b} \cos \varphi$... auxiliary variable

$t = \tan \varphi$... auxiliary variable

λ_0 ... geographical longitude of the central meridian (e.g. M34 with respect to Greenwich (16° 20') or with respect to Ferro (34°))

$l = \lambda - \lambda_0$... geographical longitude difference to central meridian

The parameters α , β , γ , δ , and ϵ are taken from [9, p. 287] for the GRS-80 ellipsoid. The numeric values are also provided in A.1. In this master thesis the Gauß-Krüger projection is only used for the representation and the comparison of the WGS84 - coordinates.

2.4 Time Systems

The time is one of the most important components of satellite-based positioning. The reason for that is that the pseudorange from the satellite to the receiver is calculated from the runtime of the signal. Due to the principle of one way measurements, both the satellite as well as the receiver require very accurate clocks. To compute the time difference between the transmitted signal and the received signal, both clocks have to be the same time scale. The relevant time systems for this master thesis are described in the next sections.

2.4.1 UTC Time

Every time scale is based on a periodic process, like the Earth's rotation, Earth revolution or Atomic oscillators. Universal Time Coordinated (UTC) is an atomic time defined as a combination of the Universal Time (UT) and the International Atomic Time (TAI). This time scale is generated by the International Bureau of Weights and Measures (BIPM) in Paris and is obtained from about 250 caesium clocks and hydrogen maser clocks. The reason for introducing this time scale is that the UT time scale is not completely linear. If the difference between UTC and UT becomes bigger than 0.9 seconds, a leap second is introduced [17, p. 39 f.].

2.4.2 GPS Time

GPS Time (GPST) is the reference time system for the Global Positioning System (GPS). It is a continuous scale that started on the 6th of January in 1980 at 00:00 UTC. It is based on atomic clocks and does not include any leap seconds. GPST has an offset of 19 seconds with respect to the International Atomic Time (TAI). It has been represented in the form of a week number and the seconds of the actual week since January 6th, 1980. At the beginning of the 1024th week, a rollover takes place. That means that the week number is set to zero. The reason for that is that there are only 10 bits reserved in the navigation message. At midnight on August 21-22, 1999, the first GPS rollover took place [9, p. 315].

2.5 Single Point Positioning

The positioning technique which is investigated in this thesis is the Single Point Positioning (SPP). In the next section, the basic principle of the SPP method is described. The other sections contain all steps in detail.

2.5.1 Basic Principle of SPP

The basic principle of satellite-based positioning is to measure the runtime of a transmitted signal. The SPP solution is based on code measurements between the GNSS satellite and the receiver. This has to be done on four visible satellites simultaneously. The reason for observing four satellites is that four unknown parameters have to be estimated. Not only the three unknown coordinates of the receiver, but also the receiver clock error has to be determined. The objective is to determine these four parameters. In many cases, more than four satellites are visible, and consequently there are more observations than unknown parameters. In this case, a least-squares adjustment is applied. The next picture shows the minimal constellation of satellite-based positioning:

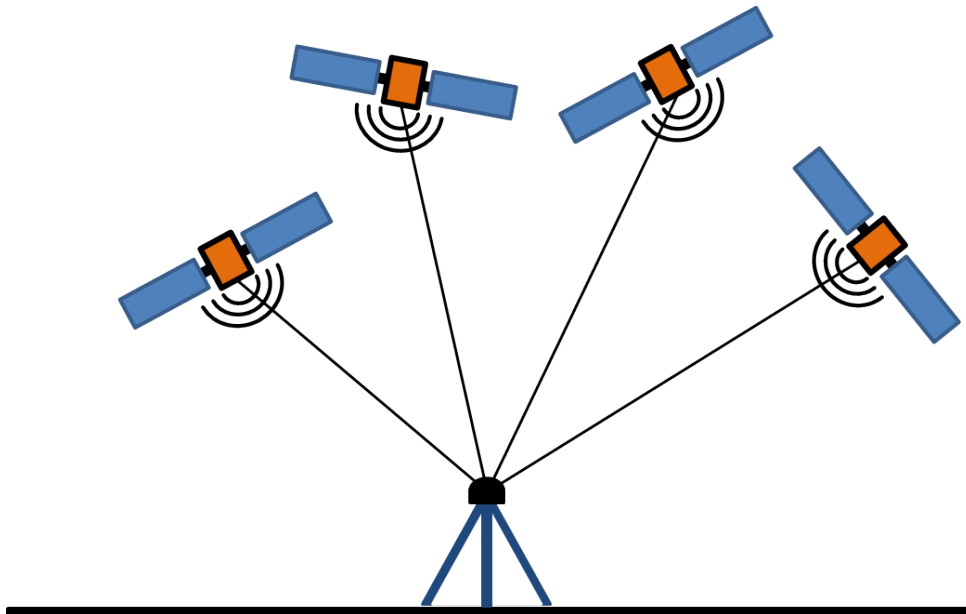


Figure 2.3: Minimal GNSS constellation

The observations between the satellite and receiver are pseudoranges. A pseudorange is the true geometric distance between transmitter and receiver, afflicted with systematic errors. These errors are listed in the following sections.

After the calculation of the satellite positions and with the pseudorange from each satellite, the user's position can be determined.

The satellite's position can be calculated with the ephemerides. There exist different types of ephemerides, for example the precise ephemerides or the broadcast ephemerides. In this master thesis, the broadcast ephemerides are used for the computation of the GPS satellite positions in the ECEF coordinate system.

These ephemerides are described in section 2.5.8, and the algorithm for calculating the positions of the satellites is provided in section 2.5.9.

2.5.2 Observation Model

An observation model connects the unknown parameters with the observations. In this case, the unknown parameters are the receiver's coordinates and the receiver clock error. The observations are the pseudoranges between the satellites and the receiver. The simplest form of the code pseudorange observation model is given in equation 2.11 [9, p. 161]:

$$R_r^s(t) = \rho_r^s(t) + c\Delta\delta_r^s(t) \quad [m] \quad (2.11)$$

In equation 2.11, $\Delta\delta_r^s(t)$ represents a combined clock error of the satellite clock and the receiver clock. A more detailed observation model includes more interfering terms. In equation 2.12, the considered errors are included [17, p. 139].

$$R_r^s(t) = \rho_r^s(t) + c[\delta_r(t) - \delta^s(t)] + \Delta I_r^s(t) + \Delta I_r^s(t) + c TGD \quad [m] \quad (2.12)$$

with

$R_r^s(t)$... observed pseudorange [m]

$\rho_r^s(t)$... geometric distance [m]

$\delta_r(t)$... receiver clock error [s]

$\delta^s(t)$... satellite clock error [s]

c ... speed of light [m s⁻¹]

$\Delta T_r^s(t)$... tropospheric delay [m]

$\Delta I_r^s(t)$... ionospheric delay [m]

TGD ... time group delay [s]

The used numerical value of c , the speed of light, is given in A.1. The shown observation model 2.12 is used for the least-squares adjustment and the estimation of the smartphone position.

2.5.3 Receiver Clock Error

The receiver clock error $\delta_r(t)$ in equation 2.12 is the difference between the internal clock of the receiver and the system time. In order to consider this clock error, it must be estimated together with the receiver's position in the least-squares adjustment.

2.5.4 Satellite Clock Error

The satellite clock error is the difference between the internal clock of the satellite and the system time. This error differs from satellite to satellite. If it also had to be estimated in the least-squares adjustment, there would always be more unknowns than observations. Therefore, it is modeled with an polynomial of second degree. The equation 2.13 shows how the satellite clock error is calculated for each satellite and each epoch [9, p. 161 f.]:

$$\delta^s(t) = a_0 + a_1(t - t_c) + a_2(t - t_c)^2 \quad [s] \quad (2.13)$$

with

a_0	... satellite clock bias [s]
a_1	... satellite clock drift [-]
a_2	... satellite clock drift rate [s^{-1}]
t	... time of transmission [s]
t_c	... time of clock [s]

The parameters a_0 , a_1 and a_2 are broadcast with the navigation message. At this step, it should be mentioned that this calculation method is not exact. Consequently, a small amount of random error remains, which is why a further term of the satellite clock error should be considered. In equation 2.13, the relativistic effect has to be taken into account. Therefore, the satellite clock correction in seconds has to be computed with equation 2.14 [9, p. 162]:

$$\delta^s(t) = a_0 + a_1(t - t_c) + a_2(t - t_c)^2 + \delta^{rel} \quad [s] \quad (2.14)$$

The relativistic effect δ^{rel} is described in more detail in the next section.

2.5.5 Relativistic Effect

GNSS satellites fly at a height of about 20 000 *km* and at a velocity of about 3.9 *km s⁻¹*, so the relativistic effect has to be considered.

The gravitational potential of the earth, as well as the velocity of the satellite influence the on-board clocks. This means that a clock in a satellite runs slower than a clock on the earth's surface. This effect has also been taken into account for the computation of the satellite clock error. Equation 2.15 can be used for calculating this effect [9, p. 145 ff.]:

$$\delta^{rel} = -2 \frac{\sqrt{\mu}}{c^2} e \sqrt{a} \sin E \quad [s] \quad (2.15)$$

μ	... geocentric gravitational constant [$\text{m}^3 \text{s}^{-2}$] [9, p. 27 f.]
c	... speed of light [m s^{-1}]
e	... eccentricity [-]
a	... semi-major axis [m]
E	... eccentric anomaly [rad]

The numerical values for the geocentric gravitational constant μ and the speed of light c are provided in A.1. Now the computed relativistic effect δ^{rel} can be substituted in equation 2.14.

2.5.6 Time Group Delay (TGD)

The Time Group Delay or Total Group Delay (TGD) is an instrumental delay, which occurs in the satellite. This can be caused by the antenna, a cable error or different filters. This kind of instrumental error is also included in the receiver. The difference to the satellite instrumental error is that in the receiver, the error is the same for all observations, so it is contained in the receiver clock error.

The TGD correction is included in the broadcast ephemerides and can be directly used for each satellite. It cancels out in a combination of two-frequency measurements [17, p. 106].

2.5.7 Atmospheric Effect

For a satellite-based positioning of a receiver, the atmosphere is one of the most influential factors. GNSS satellites fly at a height of about 20 000 *km* above the earth's surface, so the transmitted signal has to travel through the atmosphere.

The atmosphere is not homogeneous, and consequently it can be divided into different layers according to their physical properties. Figure 2.4 shows the different layers of the earth's atmosphere, which are of interest for GNSS applications:

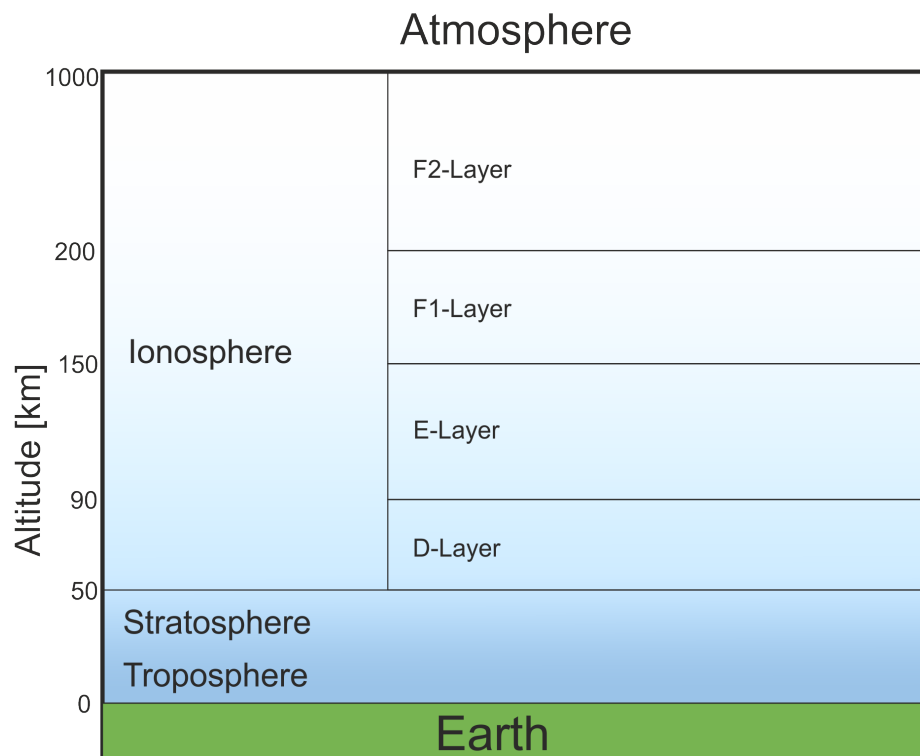


Figure 2.4: Atmospheric layers

On the basis of the electromagnetic structure, the atmosphere is divided into the so-called neutral atmosphere and the ionosphere. The GNSS community sums up the stratosphere and the troposphere as the neutral atmosphere but calls it troposphere [9, p. 65].

Snell's law says that if the electromagnetic wave goes through many layers with different refractivity, the signal path is curved. Therefore, the rays have an extended path between the satellite and the receiver [17, p. 109]. This phenomenon is shown in figure 2.5:

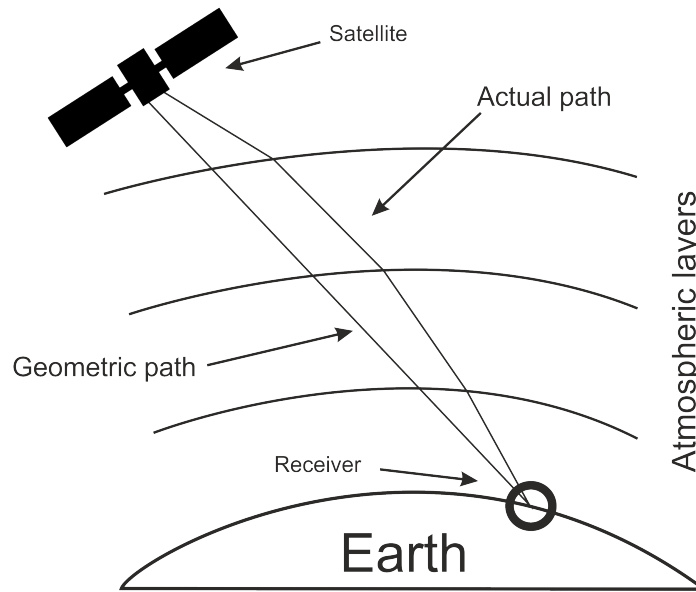


Figure 2.5: Ray path through the atmosphere [cf. 17, p. 109, Fig. 5.10]

For GNSS-specific applications, only the tropospheric and the ionospheric effect are relevant. For the computation of the position of the receiver, both effects can be approximated with models. The ionosphere and troposphere layers are described in the next two paragraphs.

Ionospheric delay

The ionospheric layer reaches from approximately 50 to 1000 km above the earth's surface. It is categorized in many small layers, starting with the D-layer, which varies with the sunlight. During night time, it is nearly totally deionized. The D-layer starts at an altitude of about 50 km and goes up to 90 km. The next layer, the E-layer, is situated roughly 90 km to 150 km above the earth's surface. The F-layer, which is divided into the F1-layer and the F2-layer, starts at 150 km and ends at a height of about 1000 km. The first 50 km are called F1-layer, and the rest represents the F2-layer. The F2-layer has the highest density of electrons, especially from 10 am. to 2 pm. local time it is increasing rapidly. The lowest electron density appears in the morning [9, p. 65 f.].

For precise applications, these layers have to be considered for the calculation of the ionospheric delay. The ionosphere is an dispersive medium, which means that it is independent of the frequency of the signal. Due to this property, the ionospheric effect can be eliminated if the receiver measures with two frequencies.

In this master thesis, only single frequency measurements are obtained, and consequently a model for the approximation of the ionospheric delay is necessary. The NeQuick model or the Klobuchar model, for instance are possibilities for modelling the ionospheric delay. For these investigations, the Klobuchar model was used for the approximation of the ionospheric delay. In this case, the ionospheric layer is also assumed as one layer for reasons of simplification. The next section deals with the Klobuchar model in detail.

Klobuchar model

The Klobuchar model [17, p. 116 ff.] is only one of many models which approximate the ionospheric delay in the signal path. This model uses the eight ionospheric parameters, α_n , and β_n , which are sent in the broadcast ephemerides data (see section 2.5.8). The input parameters of the algorithm are:

- User's approximate latitude φ_u
- User's approximate longitude λ_u
- Elevation angle E of satellite
- Azimuth A of satellite
- Klobuchar coefficients α_n and β_n

In the first step, the earth-centred angle has to be computed with equation 2.16:

$$\psi = \frac{\pi}{2} - E - \arcsin\left(\frac{R_E}{R_E + h} \cos E\right) \quad (2.16)$$

In equation 2.16, R_E is the radius of the earth and is defined with $R_E = 6378 \text{ km}$. For GPS, the height of the ionospheric layer, which is assumed as a thin layer, is set at $h = 350 \text{ km}$.

The next two formulas, 2.17 and 2.18, are used for the calculation of the latitude Φ_I and the longitude λ_I of the Ionospheric Pierce Point (IPP). The IPP is the point where the signal path between the satellite and the receiver intersects the thin ionospheric layer.

$$\Phi_I = \arcsin(\sin \varphi_u \cos \psi + \cos \varphi_u \sin \psi \cos A) \quad (2.17)$$

$$\lambda_I = \lambda_u + \frac{\psi \sin A}{\cos \Phi_I} \quad (2.18)$$

In the next step, the geomagnetic latitude Φ_m of the IPP has to be computed. In order to achieve this, equation 2.19 can be used:

$$\Phi_m = \arcsin(\sin \Phi_I \sin \Phi_P + \cos \Phi_I \cos \Phi_P \cos(\lambda_I - \lambda_P)) \quad (2.19)$$

with

$$\Phi_P = 78.3^\circ \quad \dots \text{ latitude of the geomagnetic pole}$$

$$\lambda_P = 291.0^\circ \quad \dots \text{ longitude of the geomagnetic pole}$$

Furthermore, the local time at the IPP has to be calculated with equation 2.20:

$$t = 43200 \lambda_I / \pi + t_{GPS} \quad (2.20)$$

where $0 \leq t < 86400$. To get t within this span, 86400 has to be subtract if $t \geq 86400$, and if $t < 0$, 86400 has to be added. It should also be mentioned that t_{GPS} are only the seconds within the actual day.

The amplitude A_I of the ionospheric delay can be computed with equation 2.21 and the period P_I of the delay with equation 2.22. If $A_I < 0$, then A_I should be set to zero, and if $P_I < 72\,000$, it should be set to 72 000.

$$A_I = \sum_{n=0}^3 \alpha_n \left(\frac{\Phi_m}{\pi} \right)^n \quad (2.21)$$

$$P_I = \sum_{n=0}^3 \beta_n \left(\frac{\Phi_m}{\pi} \right)^n \quad (2.22)$$

Now the phase X_I of the ionospheric delay can be calculated with the following equation:

$$X_I = \frac{2\pi(t - 50400)}{P_I} \quad (2.23)$$

With the next formula 2.24, the slant factor F can be determined:

$$F = \left[1 - \left(\frac{R_E}{R_E + h} \cos E \right)^2 \right]^{-0.5} \quad (2.24)$$

The final time delay I_1 of the ionospheric effect can be computed with:

$$I_1 = \begin{cases} [5 \cdot 10^{-9} + A_I \cos X_I] \cdot F & |X_I| < \pi/2 \\ 5 \cdot 10^{-9} \cdot F & |X_I| \geq \pi/2 \end{cases}$$

The ionospheric delay is given in seconds and is valid for the GPS L1 frequency. The computed correction has to be multiplied with the speed of light, which has to be done in equation 2.25:

$$\Delta I_r^s(t) = cI_1 \quad (2.25)$$

This delay could be substituted in equation 2.12.

Tropospheric delay

The neutral atmosphere, or simply troposphere, is the lowest layer in the atmosphere and extends from the earth's surface to a height of approximately 50 km. It is non-dispersive for frequencies up to 30 GHz. This implies that if a dual frequency method is used, the tropospheric delay cannot be eliminated. Temperature, pressure, and partial water vapor pressure are parameters of the function of the refractive index of this layer [9, p. 65]. Therefore, the compensation of the effect of the troposphere with a model requires these parameters approximately at the user's position.

There exist several models which can be used to approximate the tropospheric delay, for instance the Hopfield or Saastamoinen model. In this master thesis, the Hopfield model is used. A detailed description of this model is provided in the next paragraph.

Hopfield model

The Hopfield model [9, p. 128 ff.] is used for the approximation of the tropospheric delay. The tropospheric delay can be divided into a dry and a wet component. Approximately 90 % of the tropospheric refraction occur from the dry component $\Delta_d^{Trop}(E)$, and only 10 % from the wet component $\Delta_w^{Trop}(E)$. Equation 2.26 shows the connection of the two components and the total tropospheric delay $\Delta T_r^s(E)$:

$$\Delta T_r^s(E) = \Delta_d^{Trop}(E) + \Delta_w^{Trop}(E) \quad [m] \quad (2.26)$$

The input parameters of the model are:

- Temperature T
- Partial pressure of water vapor e
- Atmospheric pressure p
- Elevation angle E of satellite

The temperature, partial pressure of water vapor, and the atmospheric pressure have to be known for the user's location. Therefore, table 2.1 and 2.2 in the next paragraph are used in this master thesis.

The dry and wet component in equation 2.26 can be computed with the next two equations:

$$\Delta_d^{Trop}(E) = \frac{10^{-6}}{5} \frac{N_{d,0}^{Trop} h_d}{\sin \sqrt{E^2 + 6.25}} \quad (2.27)$$

$$\Delta_w^{Trop}(E) = \frac{10^{-6}}{5} \frac{N_{w,0}^{Trop} h_w}{\sin \sqrt{E^2 + 2.25}} \quad (2.28)$$

with

$$h_d = 40\,136 + 148.72(T - 273.16) \text{ [m]} \quad (2.29)$$

$$h_w = 11\,000 \text{ [m]} \quad (2.30)$$

The dry component $N_{d,0}^{Trop}$, which is required for equation 2.27 on the earth's surface, can be computed with formula 2.31:

$$N_{d,0}^{Trop} = 77.64 \frac{p}{T} \quad (2.31)$$

The wet component $N_{w,0}^{Trop}$ can be calculated with equation 2.32:

$$N_{w,0}^{Trop} = -12.96 \frac{e}{T} + 3.718 \cdot 10^5 \frac{e}{T^2} \quad (2.32)$$

The last step is to evaluate equation 2.26 and put the result directly in equation 2.12.

Parameter interpolation

In order to obtain the temperature, the partial pressure of water vapor, and the atmospheric pressure at the user's location, the parameters have to be interpolated from standard values from table 2.1 and 2.2. For the purpose of simplification, the parameter interpolation is taken from [17, p. 123 f.] in this master thesis. The next three equations were used to compute the numerical values for these parameters:

$$T(\varphi, D) = T_0(\varphi) - \Delta T(\varphi) \cos \left[\frac{2\pi(D - D_{min})}{365.25} \right] \quad (2.33)$$

$$e(\varphi, D) = e_0(\varphi) - \Delta e(\varphi) \cos \left[\frac{2\pi(D - D_{min})}{365.25} \right] \quad (2.34)$$

$$p(\varphi, D) = p_0(\varphi) - \Delta p(\varphi) \cos \left[\frac{2\pi(D - D_{min})}{365.25} \right] \quad (2.35)$$

For northern latitudes, D_{min} is 28, and for southern latitudes it is 211. D is the day of the year, counting from January 1st. The parameters $T_0(\varphi)$, $e_0(\varphi)$ and $p_0(\varphi)$ were linearly interpolated for the user's latitude from table 2.1.

Table 2.1: Meteorological parameters for the tropospheric delay: average [cf. 17, p. 124, Table 5.1]

Latitude [°]	$p_0[mbar]$	$T_0[K]$	$e_0[mbar]$
15 or less	1013.25	299.65	26.31
30	1017.25	294.15	21.79
45	1015.75	283.15	11.66
60	1011.75	272.15	6.78
75 or more	1013.00	263.65	4.11

The parameters $\Delta T(\varphi)$, $\Delta e(\varphi)$, and $\Delta p(\varphi)$ were also linearly interpolated for the user's latitude from table 2.2.

Table 2.2: Meteorological parameters for the tropospheric delay: seasonal variations [cf. 17, p. 124, Table 5.1]

Latitude [°]	$\Delta p[mbar]$	$\Delta T[K]$	$\Delta e[mbar]$
15 or less	0	0	0
30	-3.75	7.00	8.85
45	-2.25	11.00	7.24
60	-1.75	15.00	5.36
75 or more	-0.50	14.50	3.39

If the latitude is below 15° or above 75°, the numerical values for the parameters T , e , and p can be taken directly [17, p. 124] from table 2.1, respectively from table 2.2.

2.5.8 GPS Broadcast Ephemerides

Broadcast ephemerides are computed and uploaded to the satellites from the master station of the control segment. They contain general information, orbital information, and satellite clock information. The orbit message is described by the Keplerian parameters and their variation. The satellite clock information is described by coefficients for modelling the satellite clock error [9, p. 50].

The broadcast ephemerides are also necessary for the calculation of the GPS satellite positions in the ECEF system. The parameters which are used in this master thesis are listed in table 2.3. The parameters of the ionospheric correction are included into the header of the file of the broadcast ephemerides. The next two paragraphs deal with the header and data section of the navigation data.

Header section

A schematic design of a header of the broadcast ephemerides file is given in figure 2.6:

```

| 2          NAVIGATION DATA          RINEX VERSION / TYPE
CCRINEXN V1.6.0 UX  CDDIS          06-FEB-18 23:30  PGM / RUN BY / DATE
IGS BROADCAST EPHEMERIS FILE          COMMENT
0.8382D-08 -0.7451D-08 -0.5960D-07  0.5960D-07  ION ALPHA
0.8806D+05 -0.3277D+05 -0.1966D+06  0.1966D+06  ION BETA
0.0000000000000D+00-0.444089209850D-14  405504  1987 DELTA-UTC: A0,A1,T,W
18          LEAP SECONDS
          END OF HEADER
    
```

Figure 2.6: GPS broadcast ephemerides - header

The parameters ION ALPHA and ION BETA in lines four and five in the header are used in equations 2.21 and 2.22 for the calculation of the ionospheric delay. All other parameters are provided in the data section.

Data section

All parameters which are used in this master thesis are listed in table 2.3. Most of them are necessary for computing the satellite positions. Moreover, a short description of each parameter is included.

Table 2.3: GPS broadcast ephemerides - data [cf. 17, p. 57, Table 3.8]

Parameter	Description	Unit
PRN	Satellite ID	-
t_c	Time of clock	second
$t_{o \ sec}$	Ephemerides reference epoch within the current week	second
$t_{o \ week}$	Week number of ephemerides reference epoch	week
a_0	Satellite clock bias	second
a_1	Satellite clock drift	second/second
a_2	Satellite clock drift rate	second/second ²
C_{rc}, C_{rs}	Orbital radius correction	meter
C_{uc}, C_{us}	Latitude argument correction	radiant
C_{ic}, C_{is}	Inclination correction	radiant
Δn	Mean motion difference	radiant/second
M_0	Mean anomaly at reference epoch	radiant
e	Eccentricity	-
\sqrt{a}	Square root of semi-major axis	$\sqrt{\text{meter}}$
Ω_0	Longitude of the ascending node at the beginning of the week	radiant
i_0	Inclination of reference epoch	radiant
ω	Argument of perigee	radiant
$\dot{\Omega}$	Rate of node's right ascension	radiant/second
\dot{i}	Rate of inclination angle	radiant/second
TGD	Time group delay	second

The time of clock t_c is necessary for modelling the clock error polynomial in equation 2.14. The time stamp of the ephemerides reference epoch t_{oe} is given by $t_{o \ sec}$ and $t_{o \ week}$ in the navigation message. At this time, the parameters are given. The usage of the values in table 2.3 is described in section 2.5.9 for the calculation of the satellite coordinates in the ECEF system.

2.5.9 Satellite Positions

In this section, all steps for computing the satellite positions are described in detail and can be found in [17, p. 57 f.].

First of all, the time difference between the reference epoch of the ephemerides and the time of transmission have to be determined. Both time parameters are related to the beginning of the GPS week.

$$t_k = t - t_o \text{ sec} \quad (2.36)$$

The time difference has to be checked if $t_k > 302400$ s, than 604800 s has to be subtract and if $t_k < -302400$ s, 604800 has to be added.

In the next step, the mean anomaly M_k has to be computed for t_k . This is done in equation 2.37. μ is the geocentric gravitational constant [9, p. 27 f.] and is given in A.1.

$$M_k = M_0 + \left(\frac{\sqrt{\mu}}{\sqrt{a^3}} + \Delta n \right) t_k \quad (2.37)$$

The eccentric anomaly E_k has to be computed with the formula 2.38. The only way to solve this equation is iteratively. In this master thesis, the iteration ends, if the difference of two consecutive iterations is smaller than 10^{-8} .

$$M_k = E_k - e \sin E_k \quad (2.38)$$

Furthermore, the true anomaly v_k is computed with equation 2.39.

$$v_k = \arctan \left(\frac{\sqrt{1 - e^2} \sin E_k}{\cos E_k - e} \right) \quad (2.39)$$

With Equation 2.40, the argument of the latitude u_k can be determined:

$$u_k = \omega + v_k + c_{uc} \cos 2(\omega + v_k) + c_{us} \sin 2(\omega + v_k) \quad (2.40)$$

The radial distance r_k is computed with the next formula:

$$r_k = a(1 - e \cos E_k) + c_{rc} \cos 2(\omega + v_k) + c_{rs} \sin 2(\omega + v_k) \quad (2.41)$$

The inclination i_k of the orbital plane can be calculated with equation 2.42:

$$i_k = i_0 + \dot{i}t_k + c_{ic} \cos 2(\omega + v_k) + c_{is} \sin 2(\omega + v_k) \quad (2.42)$$

Equation 2.43 can be used for the computation of the longitude of the ascending node λ_k .

In this equation, ω_E is the earth's rotation rate. The numerical value is given in A.1.

$$\lambda_k = \Omega_0 + (\dot{\Omega} - \omega_E) t_k - \omega_E t_{o \text{ sec}} \quad (2.43)$$

The last step is to compute the ECEF coordinates of the satellite. In order to achieve this, three rotations around u_k , i_k , and λ_k are necessary. The next equation shows the rotation in detail:

$$\begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix} = \mathbf{R}_3(-\lambda_k) \mathbf{R}_1(-i_k) \mathbf{R}_3(-u_k) \begin{bmatrix} r_k \\ 0 \\ 0 \end{bmatrix} \quad (2.44)$$

with

$$\mathbf{R}_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}, \quad \mathbf{R}_3(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.45)$$

Now, the ECEF coordinates from the satellites are known and can be used for further calculation steps.

2.5.10 Least-squares Adjustment

In order to estimate the receiver's coordinate and the receiver clock error, a least-squares adjustment [cf. 11] is necessary. In this master thesis, the Gauß-Markov model is used. If the relation between the observations and unknown parameters is not linear, it has to be linearised. The general linear approach in matrix form is given in equation 2.46:

$$\mathbf{l} = \mathbf{A}\mathbf{x} + \mathbf{e} \quad (2.46)$$

with

- $\mathbf{l}_{[n \times 1]}$... observations
- $\mathbf{A}_{[n \times m]}$... design matrix
- $\mathbf{x}_{[m \times 1]}$... unknown parameters

$\mathbf{e}_{[n \times 1]}$... residuals

In this section, n is the number of observations and m is the number of unknown parameters. The system in equation 2.46 is singular if the number of observations is smaller than the number of unknown parameters. If there are more observations than unknown parameters, these parameters have to be estimated so that equation 2.47 becomes a minimum:

$$\|\mathbf{e}\|^2 = \sum_{i=1}^n e_i^2 \rightarrow \min \quad (2.47)$$

The estimation of the sought parameters can be achieved with equation 2.48:

$$\Delta \hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \Delta \mathbf{l} \quad (2.48)$$

with

$$\Delta \mathbf{l} = \mathbf{l} - \mathbf{l}_0 \quad (2.49)$$

and

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \quad (2.50)$$

In the case of a non-linear model, only the improvent vector $\Delta \hat{\mathbf{x}}$ can be computed. To get the absolute parameters, the following equation has to be used:

$$\hat{\mathbf{x}} = \mathbf{x}_0 + \Delta \hat{\mathbf{x}} \quad (2.51)$$

\mathbf{x}_0 is the vector with the approximate values, respectively with the values from the previous iteration step. $\Delta \mathbf{l}$ is the difference vector between the real observations and the computed observations. The computed observations \mathbf{l}_0 can be obtained from the approximate values of the \mathbf{x}_0 vector.

In the design matrix \mathbf{A} , the linear connection of the observations and the parameters are given. If the relation is not linear, a Taylor series has to be used for linearization, and furthermore approximate values are necessary.

In case not all observations should have the same influence on the position solution, a weighted least squares adjustment can be used. The procedure is exactly the same, but instead of equation 2.48, the following equation 2.52 is used:

$$\Delta \hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{P} \Delta \mathbf{l} \quad (2.52)$$

The difference is the weighting matrix \mathbf{P} . How this matrix is generated in this master thesis is shown in section 4.2.3. The covariance matrix of the parameters is given in equation 2.53:

$$\mathbf{Q} = \sigma^2 (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \quad (2.53)$$

In this master thesis $\sigma^2 = 1$ is used for the computations. The detailed description how the least-squares adjustment is used for Single Point Positioning (SPP) is given in section 4.2.4.

2.5.11 Position Accuracy

The position accuracy is one of the most important parts of a position solution. In this master thesis, the formal accuracy [17, p. 148] is used for every epoch in the output file (see Fig. 4.5) of the project's Android application.

The formal accuracy indicates the quality of the estimation of the parameters for both the position and the receiver clock and can be calculated from the covariance matrix \mathbf{Q} . This matrix includes the formal accuracy from the cartesian coordinates in the ECEF system and from the receiver clock error. In a more useful expression, it is shown in east, north, and up components. Therefore, the geometric part of the covariance matrix \mathbf{Q}_{xyz} has to be transformed with the following equation:

$$\mathbf{Q}_{enu} = \mathbf{R}^T \mathbf{Q}_{xyz} \mathbf{R} \quad (2.54)$$

with

$$\mathbf{R} = \begin{bmatrix} -\sin \lambda & -\sin \varphi \cos \lambda & \cos \varphi \cos \lambda \\ \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \sin \lambda \\ 0 & \cos \varphi & \sin \varphi \end{bmatrix} \quad (2.55)$$

φ and λ are the ellipsoidal coordinates of the estimated position.

The formal accuracy from each component can be computed of elements from \mathbf{Q}_{enu} with the next three equations 2.56 to 2.58:

$$\sigma_{east} = \sqrt{\mathbf{Q}_{enu11}} \quad (2.56)$$

$$\sigma_{north} = \sqrt{\mathbf{Q}_{enu22}} \quad (2.57)$$

$$\sigma_{up} = \sqrt{\mathbf{Q}_{enu33}} \quad (2.58)$$

The standard deviation of the mean position in the Gauß-Krüger plane, which are analysed and compared in chapter 5, is described hereinafter. For more information see [11].

First of all, the error vectors \mathbf{v}_x and \mathbf{v}_y have to be determined. In these vectors the differences to the mean x -component and the mean y -component are included:

$$\mathbf{v}_x = \begin{bmatrix} \bar{x} - x_1 \\ \bar{x} - x_2 \\ \vdots \\ \bar{x} - x_n \end{bmatrix}, \quad \mathbf{v}_y = \begin{bmatrix} \bar{y} - y_1 \\ \bar{y} - y_2 \\ \vdots \\ \bar{y} - y_n \end{bmatrix} \quad (2.59)$$

n describes the number of epochs. The standard deviation for both components can be computed with the following two formulas:

$$\sigma_x = \sqrt{\frac{1}{n-1} \mathbf{v}_x^T \mathbf{v}_x} \quad (2.60)$$

$$\sigma_y = \sqrt{\frac{1}{n-1} \mathbf{v}_y^T \mathbf{v}_y} \quad (2.61)$$

As further parameter for the accuracy, the standard deviation from Helmert is used and can be computed with equation 2.62:

$$\sigma_{Helmert} = \sqrt{\sigma_x^2 + \sigma_y^2} \quad (2.62)$$

Chapter 3

Android Raw Measurements

3.1 Introduction

In May 2016, Google announced that the access of GNSS raw measurements from the version Android 7 (Nougat) would be allowed. Since then, users of a suitable mobile device have been able to use the raw measurements. Prior to that, only the position, velocity, and time information which had been processed by the GNSS chip in the smartphone were available for users. Consequently, only a meter-level accuracy of the chip's internal positioning was achieved [1].

The provision of the raw measurements is a revolution in the development of smartphone positioning. The usage of these observations has a lot of advantages, as the performance of the smartphone positioning is improving, especially the possibility of other positioning techniques, for instance a relative positioning between two smartphones or a PPP. It also opens the possibility to use code pseudoranges, Doppler, and carrier phase measurements for computing the user's position. Thus, a better positional accuracy can be achieved [1]. Moreover, the multi-GNSS solutions can be optimised much more easily [7]. Galileo, Beidou, GLONASS, or Quasi-Zenith Satellite System (QZSS) are further GNSS constellations which can be taken advantage of. If more than one constellation is used, the number of GNSS signals can be rapidly increased.

Additionally, the GNSS data can be fused with Inertial Measurement Unit (IMU) measurements from the smartphone's sensors [6].

Furthermore, there also exist two-frequency measurements on smartphones. The Xiaomi Mi 8, for instance, has the Broadcom BCM47755 chip embedded. In the case of Galileo, the E1/L1 and the E5/L5 frequency, for example, can be tracked simultaneously with this chip. Due to the L5 frequency positioning, navigation is much easier, especially in difficult environments like cities. Moreover, the multipath effect can be reduced because of the L5 frequency [19].

3.2 Possible Mobile Devices

The Android raw measurements are not available on all mobile devices. Therefore, some possible devices are included in table 3.1. In the following table, the model of the mobile device and the Android version are provided, as well as the available GNSS constellations.

Table 3.1: Suitable mobile devices [3]

Model	Android version	GPS	GLONASS	Galileo	Beidou	QZSS
HTC U11 Plus	8.0	✓	✓			
HTC U11 Life	8.0	✓	✓			
Huawei Mate 10	8.0	✓	✓			
Huawei Mate 10 Pro	8.0	✓	✓			✓
Google Pixel 2 XL	8.0	✓	✓	✓	✓	✓
Google Pixel 2	8.0	✓	✓	✓	✓	✓
Sony Xperia xZ1	8.0	✓	✓	✓	✓	
Samsung Note 8 (Exynos)	7.1	✓	✓	✓	✓	
Samsung Note 8 (QCOM)	7.1	✓	✓	✓	✓	
LG V30	7.1.2	✓	✓			
Moto X4 2017	7.1	✓	✓			
Essential PH-1	7.1	✓	✓			
Moto Z2	7.1	✓	✓			
HTC U11	7.1	✓	✓			
OPPO R11	7.1	✓	✓	✓	✓	
Huawei Honor 9	7.0	✓	✓			
Samsung S8 (Exynos)	7.0	✓	✓	✓	✓	✓
Samsung S8 (QCOM)	7.0	✓				
Huawei P10	7.0	✓	✓	✓	✓	✓
Huawei P10 Lite	7.0	✓				
Huawei Honor 8	7.0	✓	✓		✓	
Huawei Mate 9	7.0	✓	✓		✓	
Huawei P9	7.0	✓	✓		✓	
Google Pixel XL	7.0	✓				
Google Pixel	7.0	✓				
Nexus 6P	7.0	✓				
Nexus 5X	7.0	✓				
Nexus 9 (non cellular vers.)	7.1	✓	✓			

The three smartphones which are used for the investigations in this master thesis are described in the next section.

3.3 Used Test Equipment

For the examination of the raw GNSS measurements in this master thesis, three different smartphones are used. A short description of the three mobile devices is given in the next three sections. For the generation of the raw measurement file, the application “GNSSLogger“ from Google [20] was used.

3.3.1 Huawei Honor 8

The main investigations are performed with the Huawei Honor 8. Some of the technical specifications of this smartphone are provided in the following list:

- Manufacturer: Huawei
- Model: Honor 8
- Android version: 7.0
- GNSS Positioning Chipset: BCM4774 [10]
- GNSS constellations: GPS, GLONASS, Beidou
- Measurements: Pseudorange, Doppler, Carrier phase (Accumulated delta range)[3]

3.3.2 Huawei P10

The second test device is the Huawei P10. The technical specifications are listed below:

- Manufacturer: Huawei
- Model: P10
- Android version: 8.0
- GNSS Positioning Chipset: BCM4774 [5]
- GNSS constellations: GPS, GLONASS, Galileo, Beidou, QZSS
- Measurements: Pseudorange, Doppler, Carrier phase (Accumulated delta range)[3]

3.3.3 Samsung S8 (Exynos)

The third smartphone which is used for the comparison of the raw measurements is the Samsung S8. The technical specifications are as follows:

- Manufacturer: Samsung
- Model: Galaxy S8 Exynos, G950F
- Android version: 8.0
- GNSS Positioning Chipset: BCM4774 [4]
- GNSS constellations: GPS, GLONASS, Galileo, Beidou, QZSS
- Measurements: Pseudorange, Doppler, Carrier phase (Accumulated delta range)[3]

3.3.4 Smartphone GNSS Chip - Broadcom BCM4774

In all three smartphones described above, the same GNSS chip is installed. The Broadcom 4774 is a chip which supports the satellite constellations Galileo, GPS, GLONASS, Satellite-based Augmentation Systems (SBAS), QZSS, and Beidou. Although the GNSS chip can process these constellations, not all GNSS systems are available to the user in every smartphone. The phone's internal position is improved by this chip with respect to the previous generation [2].

3.4 Structure

In this section, the general structure of the Android raw measurements is shown. In the next paragraphs, all parameters of the raw data are provided in detail. The informations of the following paragraphs are from [3], otherwise the source is given explicitly.

ElapsedRealtime

The ElapsedRealtime is the time since the system was booted. It is a continuous time, includes deep sleep, and runs when the CPU is in power saving mode as well.

Time

The field Time or more precisely “TimeNanos“ in the raw measurements is referred to the internal hardware clock of the receiver in the smartphone. This value is expressed in nanoseconds. It has to be used for the calculation of the pseudorange.

Leapsecond

The leap seconds are related to the clock’s time in seconds.

TimeUncertainty

This term refers to the time’s uncertainty in nanoseconds. “TimeUncertainty“ is an absolute value and is often zero or not included in the measurement file.

FullBias

This is the time difference between the clock of the GPS receiver in the smartphone and the GPST since January 6, 1980. The FullBias is also expressed in nanoseconds.

Bias

The field “BiasNanos“ in the Android raw measurements is the hardware clock’s bias in sub-nanoseconds.

BiasUncertainty

“BiasUncertainty“ is the clock’s bias uncertainty and is also provided in nanoseconds.

Drift

This term refers to the drift of the clock of the GPS receiver in the mobile device. The unit in which this parameter is given is nanoseconds per second.

DriftUncertainty

This value describes the clock's drift uncertainty in nanoseconds per second.

HardwareClockDiscontinuity

This value serves as a counter for the hardware clock discontinuity. This counter should always be the same as in the previous epoch. If this parameter changes from one epoch to another, it indicates that the hardware clock is not continuous. In this case, all time parameters from the class `GnssClock` should be reset.

Svid

The satellite vehicle ID is an identification number for each satellite. In most constellations, it is the Pseudo-Random Noise (PRN). In [3] it is also denoted as Pseudo-Random Number. For GLONASS, it is the Frequency Channel Number (FCN) and the Orbital Slot Number (OSN). The values of the different constellations of this parameter are:

- GPS: 1- 32
- SBAS: 120 - 151, 183 - 192
- GLONASS: 1 - 24 as the OSN (if this is known, OSN is preferred), and 93 - 106 as the FCN ($[-7 \text{ to } 6] + 100$)
- QZSS: 193 - 200
- Galileo: 1 - 36
- Beidou: 1 - 37

TimeOffset

This refers to the time offset between the measurement time and the field "Time." It is an individual time offset for the observation and is expressed with an accuracy of sub-nanoseconds. This offset is used for the computation of the received time of the signal.

State

The state parameter in the Android raw measurements shows the sync state for each satellite. For example the constant value of the tracking state, which has Time of Week (TOW) decoded is 8 [3]. This state is checked in the data filtering of this master thesis for the right tracking status. The parameter "State" of each measurement is bitwise compared to these constant values. For further state constants see [3].

ReceivedSvTime

The “ReceivedSvTime“ is the received satellite time at the measurement time in nanoseconds. For GPS and QZSS, this value is relative to the beginning of the actual GPS week. If the observation is from a GLONASS satellite, this value is the time of day. In the case of a Beidou satellite or a Galileo satellite respectively, it is relative to the Beidou time of week or the Galileo time of week. This value is relative to the SBAS time if the measurement is from an SBAS satellite.

ReceivedSvTimeUncertainty

This parameter is the estimated error of the received satellite time for each satellite. It is also provided in nanoseconds.

Cn0

The Carrier-to-noise power density ratio C/N_0 is a quality parameter of the received signal. It is a combination of the carrier power C and the noise power density N_0 [9, p. 86]. In the raw data this value is expressed in dBHz and has typically a range of 10-50 dBHz [3].

PseudorangeRate

The pseudorange rate is given in meters per second and is not corrected by the receiver and satellite clock frequency error. A positive rate represents a satellite which is moving away from the receiver, and a negative value shows that the satellite is coming closer to the receiver. With this parameter, the doppler shift can be computed. In order to achieve this, equation 3.1 can be used [3]:

$$DopplerShift = -\frac{PseudorangeRate}{k} \quad [\text{Hz}] \quad (3.1)$$

The constant k , which represents the wavelength, can be calculated with the following equation [7]:

$$k = \frac{c}{f_c} \quad (3.2)$$

with

c ... speed of light [$m s^{-1}$]

f_c ... signal central frequency (e. g.
L1 = 1575.42 MHz for GPS) [Hz]

PseudorangeRateUncertainty

The field “PseudorangeRateUncertainty“ is the uncertainty of the pseudorange’s rate and is given in meters per second.

AccumulatedDeltaRangeState

The validity of the “AccumulatedDeltaRange“ is represented by this state value. It shows if the “AccumulatedDeltaRange“ is reset or a cycle slip has happened.

AccumulatedDeltaRange

This value stands for the accumulated delta range since the last reset was done. It is expressed in meters. The carrier phase measurement is provided as this value. With equation 3.3, the connection between the carrier phase and the accumulated delta range is determined:

$$AccumulatedDeltaRange = -k \cdot CarrierPhase \quad [m] \quad (3.3)$$

k is the wavelength again and can be calculated with equation 3.2.

AccumulatedDeltaRangeUncertainty

This value indicates the uncertainty of the accumulated delta range. It is also represented in meters.

CarrierFrequency

The next field of the Android raw measurements is the “CarrierFrequency.“ This field shows the carrier frequency of the signal which is tracked. In the case of GPS, it could be L1 (1575.45 MHz), L2 (1227.60 MHz), or L5 (1176.45 MHz). If this parameter is not included in the measurement file, the measurement refers to the central frequency (e. g. L1 at GPS).

CarrierCycles

This is the number of full cycles which occur between the GNSS satellite and the receiver. This value was added in the Application Programming Interface (API) level 24 and is not supported in the API level 28. Instead of this field, the “AccumulatedDeltaRange“ has to be used.

CarrierPhase

The “CarrierPhase“ is the fractional part of a full carrier phase. This field is also not supported. Again, “AccumulatedDeltaRange“ has to be used for the computations.

CarrierPhaseUncertainty

Instead of the “CarrierPhaseUncertainty,” the “AccumulatedDeltaRangeUncertainty” is to be used.

MultipathIndicator

This parameter is an indicator for each measurement for the multipath state. It is depending on whether an observation is influenced or not.

SnrIn

The SNR is the signal-to-noise ratio and is provided in decibel.

ConstellationType

This field of the Android raw measurements indicates which constellation type the current observation is from. In the following table 3.2, the possible constellations are listed with their indicator [3]:

Table 3.2: Constellations

Constellation	Indicator
Beidou	5
GALILEO	6
GLONASS	3
GPS	1
QZSS	4
SBAS	2
Unknown	0

Agc

The last field is the “Agc” or Automatic Gain Control level. It is expressed in decibel and is used as an amplifier to regulate the incoming signal. It can be used to indicate if a signal is disturbed or not.

3.5 GNSS Analysis Tool

The Android raw measurements can be logged with the Android application “GNSSLogger“ from Google, for instance. To analyse these raw measurements, the desktop application “GNSS Analysis Tool“ (also released by Google) can be used [20]. In this master thesis, the results of the self-calculated position is compared with the results of the “GNSS Analysis Tool v.2.4.0.0.“ The starting window of this program is shown in the following picture:

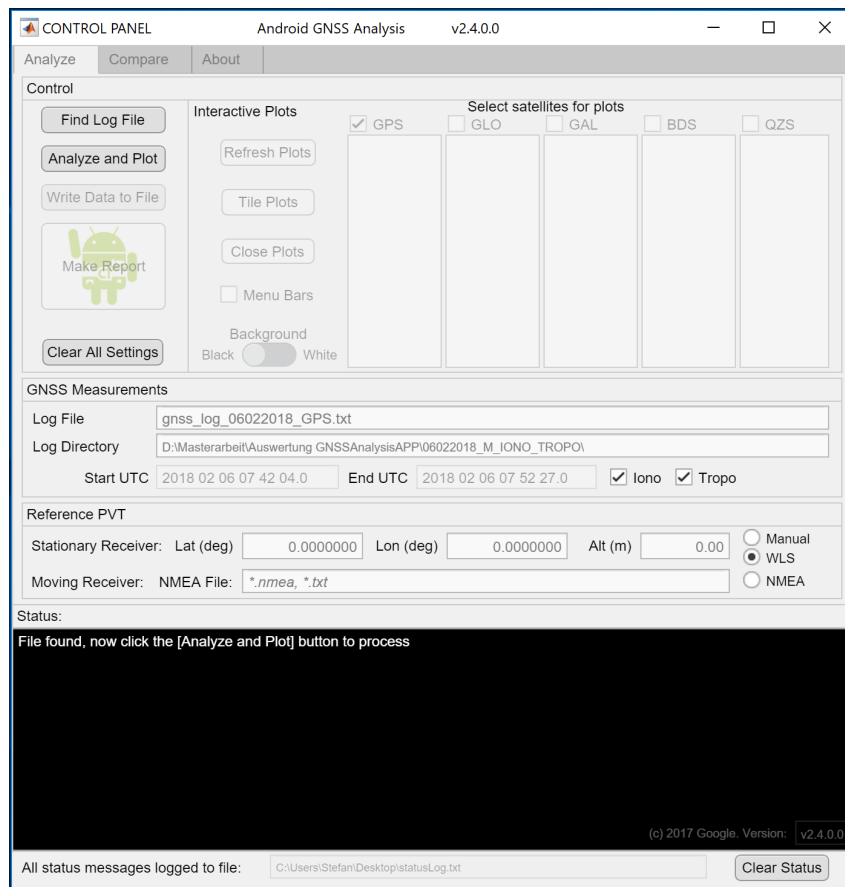


Figure 3.1: Start window - “GNSS Analysis Tool“

In this part of the program, all settings for the computation can be selected. First of all, the logged measurement file has to be chosen. After that, the satellite constellations, the atmospheric correction, and many other settings can be selected. With the button “Analyze and Plot,“ the actual measurement file can be processed. When the data is processed, the program creates some plots of the results. In addition, the graphical representation of the results and an output file with the positions are created. Furthermore, the comparison of several data files is possible. The program can process GPS, GLONASS, Galileo, Beidou, and QZSS signals and is also suitable for multiple-frequencies.

Furthermore, is also possible to create a report of the analysis of the dataset [3]. The program’s graphical representation of the Android raw measurements are shown in figures 3.2, 3.3, and 3.4:

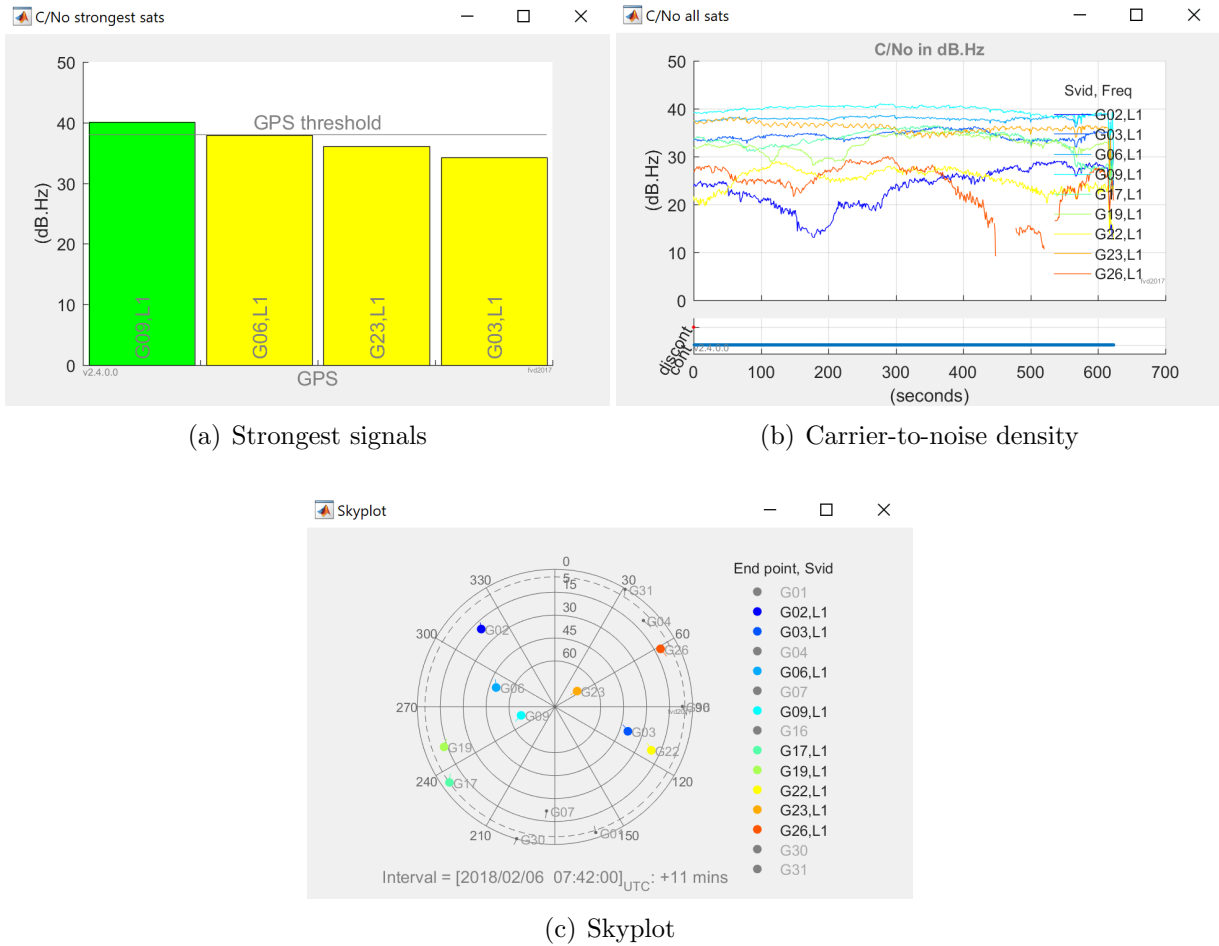
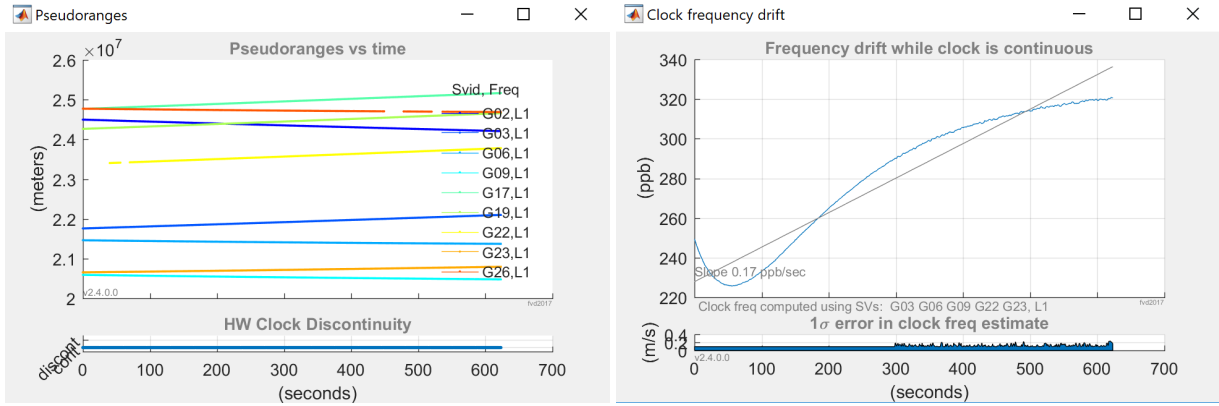


Figure 3.2: “GNSS Analysis Tool“ - part 1

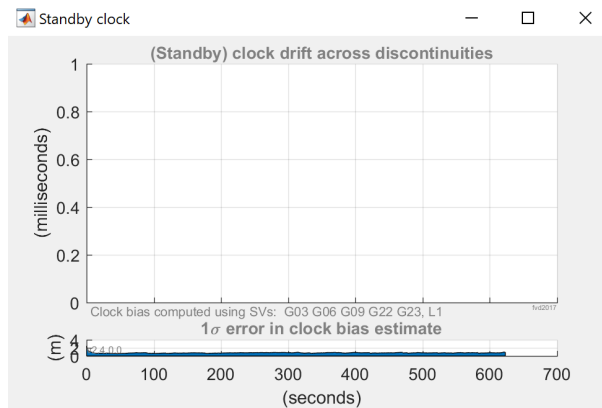
In the subfigures above, the first plot shows the four satellites with the strongest C/N_0 . This is portrayed for all constellations in the measurement file. In this example, only GPS can be seen because the analyzed data file does not include any other constellations. Figure 3.2(b) presents the Carrier-to-noise density of all visible satellites. In figure 3.2(c) there is a skyplot from all satellites which are visible from the user’s position, given [3].

The next three plots from the “GNSS Analysis Tool“ which include the clock information are shown in the following figure:



(a) Pseudoranges

(b) Frequency drift of the receiver clock

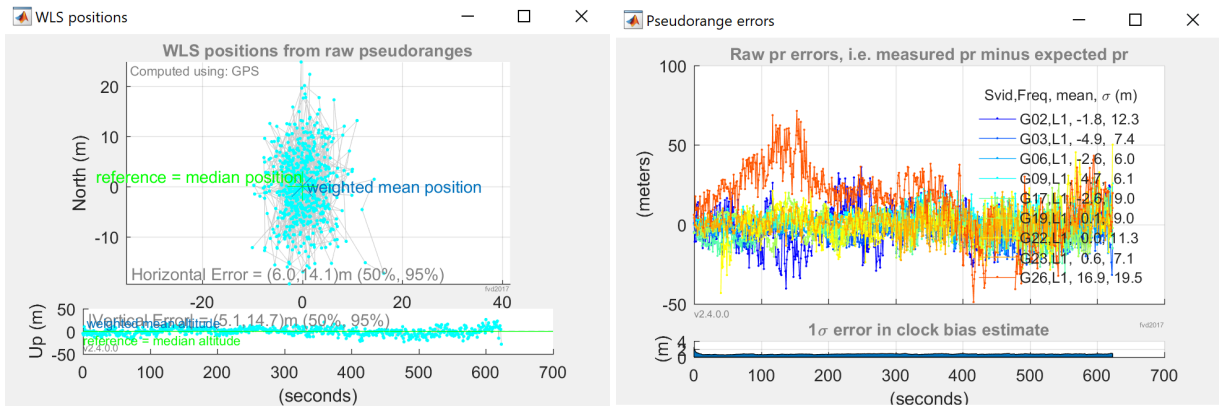


(c) Offset of the standby clock

Figure 3.3: “GNSS Analysis Tool“ - part 2

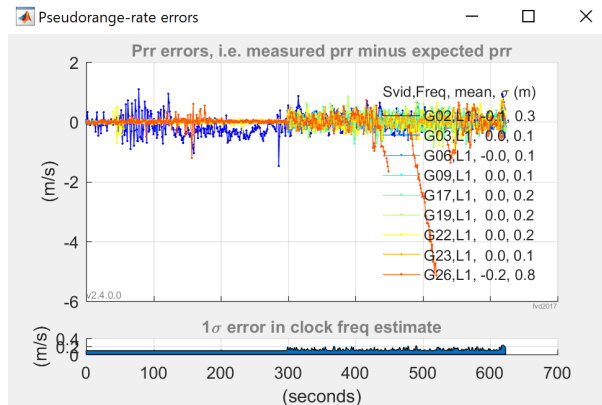
Figure 3.3(a) portrays the pseudoranges from each satellite. They are represented by a time series over the whole measurement time. The clock frequency offset of the receiver is plotted in figure 3.3(b), and the offset of the standby clock is shown in figure 3.3(c) for each epoch [3]. In the analysed data set the clock discontinuities are zero.

The measurement plots of the “GNSS Analysis Tool“ are presented in the following figure:



(a) Weighted least-squares positions

(b) Residual errors of the pseudorange



(c) Residual errors of the pseudorange-rate

Figure 3.4: “GNSS Analysis Tool“ - part 3

In the figures above, all information with respect to the measurements is included. The first plot shows the results of the computed weighted least-squares adjustment. The weighting is done with uncertainty of each measurement [3].

The second and third plot shows the errors of the pseudoranges, respectively the pseudorange-rate measurements for each satellite.

For more information about the “GNSS Analysis Tool“ see [3] and [20].

Chapter 4

Data Processing with Android

4.1 Introduction

A smartphone application which can process GNSS raw data has numerous possibilities. Knowing the smartphone's position, respectively the position of the user, is very important in many situations, for example the management of the Search and Rescue (SAR) emergency operations or simply to navigate the user to their target. Furthermore, the user's location is very important in a Location-based Service (LBS) beside the information and communication. In such applications the positional accuracy, the reliability, and the availability of the user's position play an important role [8].

For the reasons mentioned above, smartphone positioning is a hot topic today, but also in future for the development of positioning methods.

The goal of this master thesis is to create an application for a Single Point Positioning (SPP) with Android raw measurements on the smartphone.

This chapter deals with the detailed implementation of the GNSS processing algorithm with Android. A thorough description of the SPP algorithm and how to process the Android raw measurements is provided. The calculation of the pseudorange or the data filtering, as well as the weighting methods are explained in this chapter.

The Android application was created with the software "Android studio 3.1.4." A short overview of the software structure is also included. Furthermore, this chapter contains some screen shots of the final Android application. The output files of the application are described as well.

4.2 Single Point Positioning (SPP)

This section provides a detailed description of the SPP algorithm and how it is implemented in the software.

4.2.1 Pseudorange Generation

The pseudoranges are not directly included in the raw measurement data. They can however be calculated with the provided parameters. A detailed description of how the pseudoranges from Android raw measurements are calculated can be found in [7].

The basic principle of a pseudorange between the receiver and the satellite is presented in the next equation:

$$R_r^s(t) = (t_{RX} - t_{TX}) \cdot c \quad [m] \quad (4.1)$$

with

c	... speed of light [$m \ s^{-1}$]
t_{RX}	... measurement time [s]
t_{TX}	... received satellite time at the measurement time [s]

Essentially, it is the time difference between the transmitted and the received time. This difference is multiplied by the speed of light in order to get the pseudorange in meters. From the Android raw measurements, the time of transmission t_{TX} is given by the parameter “ReceivedSvTime.” It is important that the tracking status equals TOW decoded (see sec. 4.2.2).

In the first step, the measurement time $t_{RX_{GNSS}}$ has to be computed [7]:

$$t_{RX_{GNSS}} = [Time + TimeOffset] - [FullBias(1) + Bias(1)] \quad [ns] \quad (4.2)$$

In equation 4.2, all parameters which are needed can be found in the raw measurement file. They are described in section 3.4.

Note: In equation 4.2, only the first “FullBias“ and “Bias“ of the entire measurement period are used for the computation of $t_{RX_{GNSS}}$. The reason for that is that the receiver clock has a drift. Consequently, this parameter is set to constant, in this case with the first entry in the measurement file, in order to estimate this drift together with the receiver clock error in the least-squares adjustment. For further information about this issue see [7] and [15].

The calculated time in equation 4.2 is the full GNSS time. Normally the GNSS receivers has the GPST as a default GNSS reference time [7]. The time parameter “ReceivedSvTime“ or t_{TX} is TOW, and consequently relative to the beginning of the current week. Therefore, the measurement time t_{RX} also has to relate to the beginning of the week. This can be achieved with equation 4.3 [7]:

$$t_{RX} = t_{RX_{GNSS}} - WeekNumber \cdot 604800 \cdot 10^9 \quad [ns] \quad (4.3)$$

The week number can be obtained from the “FullBias“ from the Android raw measurements. t_{TX} and t_{RX} must be converted from nanoseconds to seconds. Finally, the pseudorange can be computed with equation 4.1.

4.2.2 Data Filtering

The data filtering of the Android raw measurements is a critical point in the estimation of the smartphone position. The obtained bad GNSS signals have to be correctly sorted out.

In this master thesis, the parameter “ReceivedSvTimeUncertainty“ is used for filtering the data. The reason why this parameter is used for the filtering algorithm is that only pseudoranges are used as observations, and the “ReceivedSvTimeUncertainty“ indicates if the parameter “ReceivedSvTime,“ which is needed for the calculation of the pseudorange, is reliable. In some received signals, this parameter has a value of 1 000 000 000 nanoseconds. In comparison, “normal“ measurements have a range of approximately 5 to 100 nanoseconds. Thus, measurements of such a high value have to be removed from the computations.

There are more parameters which can be used to filter the data. Further possibilities are for example the “PseudorangeRateUncertainty“ or the “AccumulatedDeltaRangeUncertainty.“ It is only useful to filter the observations with these parameters when Doppler and Carrier phase measurements are used for the calculation of the user position.

A further filter which is implemented in the created Android software serves to cancel out all measurements of which the tracking status is wrong. In some cases, the receiver is locked in the C/A code, for instance (for further synchronisation stages see [7]), and consequently the entire C/A codes are unknown. Therefore, the valid range of the “ReceivedSvTime“ is from 0 to 1 millisecond. That occurs especially with weak signals, for instance satellites with a low elevation angle or when the signal is affected by multi-path.

The correct lock of the code is achieved if the TOW is decoded so that the valid range is between 0 and 1 week. The problem of the wrong lock of the code leads to an ambiguous pseudorange of which only the fractional part is known. There exist some advanced algorithms to solve this problem, however in this master thesis, these observations were excluded from the computations. For further information on this issue see [7] or [13].

As a last filter of the software, all GNSS constellations except the Global Positioning System (GPS) are excluded from the computations. As a result, a pure GPS solution is obtained.

4.2.3 Weighting

Weighting in the determination of a receiver position is one of the most important part of the whole process. This is based on the fact that not all measurements which are obtained from the receiver have the same quality. Therefore, it is very useful that not all observations influence the solution equally. There are many options when it comes to dealing with the different quality of the observations. The easiest way is that all measurements have the same weight. A further possibility is that the observations are categorised in bad and good measurements. In this case, a threshold for a good measurement is set, and all observations which are under this threshold are eliminated and not used for the positioning solution.

A much better alternative is to give each measurement an individual weight so that all measurements influence the solution at different strengths.

In this master thesis, different options were included in the investigations. The three options which are implemented in the Android App are shown in the next three sections:

Equal weights

This kind of weighting is the simplest form. All observations influence the solution equally. The weighting matrix \mathbf{P} has the following form:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (4.4)$$

Weighting by elevation angle

The weighting of the measurements with the elevation angle of each satellite is a further possibility. In fact, the signals from a satellite with a lower elevation angle have covered a longer distance through the atmosphere, so the influencing factor of the ionosphere and troposphere is much higher than from satellites with a larger elevation angle. If the satellite is exactly in the zenith, the signal which is transmitted from the satellite to the receiver travels the shortest way through the atmosphere. The weighting matrix is also a diagonal matrix and has the following structure:

$$\mathbf{P} = \begin{bmatrix} \sin E_1 & 0 & \cdots & 0 \\ 0 & \sin E_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sin E_n \end{bmatrix} \quad (4.5)$$

Weighting by signal strength

The third option for weighting in the Android application is the weighting by the signal strength. In order to achieve this, the Carrier-to-noise-density (C/N_0), which is directly provided in the Android raw measurements, can be used. The \mathbf{P} matrix has the following appearance:

$$\mathbf{P} = \begin{bmatrix} C/N_{0_1} & 0 & \cdots & 0 \\ 0 & C/N_{0_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C/N_{0_n} \end{bmatrix} \quad (4.6)$$

The Carrier-to-noise-density can be used directly for the weighting matrix.

4.2.4 Implemented Algorithm

In this section, a detailed description of the implemented positioning algorithm is provided. The calculations are directly done on the smartphone in real time.

First of all, data filtering, which is described in section 4.2.2, is performed for each measurement. Consequently, only the observations from the GPS satellites with the right tracking status and a plausible “ReceivedSvTimeUncertainty“ are used for the computation of the smartphone position. If there are fewer than four satellites, the epoch is skipped.

The next step is to compute the pseudoranges from all observations. The pseudorange generation is described in section 4.2.1. Especially the numerical aspect has to be considered.

The calculation of the receiver position is described in the next paragraph.

Receiver position

For the estimation of the smartphone position and the receiver clock error, approximate values of these parameters are necessary. The parameter vector \mathbf{x}_0 with these values has the following structure:

$$\mathbf{x}_0 = \begin{bmatrix} x_r^0 \\ y_r^0 \\ z_r^0 \\ \delta_r(t) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (4.7)$$

The vector \mathbf{x}_0 is used for the calculations of the first iteration step instead of \mathbf{x} in the least-squares adjustment.

First, the satellite positions and the satellite clock error are calculated. In order to achieve this, the broadcast ephemerides from section 2.5.8 are used in the formulas in section 2.5.9 for the determination of the satellite positions. The modelling of the satellite clock error is performed with the formulas in section 2.5.4, as well as with the relativistic effect in section 2.5.5.

The coordinates of the satellites, which are computed with the formulas in 2.5.9, cannot be applied directly to the positioning algorithm. They must be rotated by the influence of the rotation of the earth during the runtime of the signal. Consequently, the following transformation matrix has to be used:

$$\mathbf{R} = \begin{bmatrix} \cos \omega_E dt & -\sin \omega_E dt & 0 \\ \sin \omega_E dt & \cos \omega_E dt & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

ω_E is the earth rotation rate. The numerical value which is used for the investigations can be found in A.1. dt describes the runtime of the signal from the satellite to the receiver. The final rotated satellite positions \mathbf{X}_{ECEF}^{rot} can be calculated with equation 4.9:

$$\mathbf{X}_{ECEF}^{rot} = \mathbf{R}^T \mathbf{X}_{ECEF} \quad (4.9)$$

With those satellite coordinates, the design matrix \mathbf{A} can be created. However, the coordinates of the smartphone, respectively the approximate coordinates, are also necessary. The following SPP algorithm can also found in [17, p. 139 ff.]. With the functional relation

$$\rho_r^s(t) = \sqrt{(x^s(t) - x_r)^2 + (y^s(t) - y_r)^2 + (z^s(t) - z_r)^2} \quad (4.10)$$

the geometric range of each satellite to the receiver is given. For the least-squares adjustment, all correction terms have to be taken into account. In order to fill the design matrix, the next equation 4.11 has to be linearised:

$$R_r^s(t) = \rho_r^s(t) + c[\delta_r(t) - \delta^s(t)] + \Delta T_r^s(t) + \Delta I_r^s(t) + c TGD^s(t) \quad (4.11)$$

The correction terms $c\delta^s(t)$, $\Delta T_r^s(t)$, $\Delta I_r^s(t)$, and $TGD^s(t)$ come from a model, respectively from the broadcast ephemerides, and will be considered later. The equation which is used for the design matrix is given in formula 4.12:

$$R_r^s(t) = \sqrt{(x^s(t) - x_r)^2 + (y^s(t) - y_r)^2 + (z^s(t) - z_r)^2} + c\delta_r(t) \quad (4.12)$$

Now, equation 4.12 is derived based on the parameters which should be estimated. In this case, these are the three components of the receiver's position x_r , y_r , and z_r , as well as the receiver clock error $\delta_r(t)$.

Consequently, the design matrix has the following structure:

$$\mathbf{A} = \begin{bmatrix} \frac{\partial R_r^1(t)}{\partial x_r} & \frac{\partial R_r^1(t)}{\partial y_r} & \frac{\partial R_r^1(t)}{\partial z_r} & \frac{\partial R_r^1(t)}{\partial \delta_r} \\ \frac{\partial R_r^2(t)}{\partial x_r} & \frac{\partial R_r^2(t)}{\partial y_r} & \frac{\partial R_r^2(t)}{\partial z_r} & \frac{\partial R_r^2(t)}{\partial \delta_r} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial R_r^n(t)}{\partial x_r} & \frac{\partial R_r^n(t)}{\partial y_r} & \frac{\partial R_r^n(t)}{\partial z_r} & \frac{\partial R_r^n(t)}{\partial \delta_r} \end{bmatrix} \quad (4.13)$$

For each visible satellite which represents one line in the matrix, the following mathematical derivatives of equation 4.12 have to be determined:

$$\begin{aligned} \frac{\partial R_r^s(t)}{\partial x_r} &= -\frac{x^s(t) - x_r}{\rho_r^s(t)} & \frac{\partial R_r^s(t)}{\partial z_r} &= -\frac{z^s(t) - z_r}{\rho_r^s(t)} \\ \frac{\partial R_r^s(t)}{\partial y_r} &= -\frac{y^s(t) - y_r}{\rho_r^s(t)} & \frac{\partial R_r^s(t)}{\partial \delta_r} &= c \end{aligned} \quad (4.14)$$

In the last column of the design matrix, the derivative is the constant value c , which is the speed of light in vacuum. By virtue of numerical stability for the calculations, this value is set to 1. This factor is taken into account at the end of all matrix operations.

In matrix form, the functional relation can be expressed as:

$$\begin{bmatrix} R_r^1(t) - \rho_r^1(t) - D^1(t) \\ R_r^2(t) - \rho_r^2(t) - D^2(t) \\ \vdots \\ R_r^n(t) - \rho_r^n(t) - D^n(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial R_r^1(t)}{\partial x_r} & \frac{\partial R_r^1(t)}{\partial y_r} & \frac{\partial R_r^1(t)}{\partial z_r} & \frac{\partial R_r^1(t)}{\partial \delta_r} \\ \frac{\partial R_r^2(t)}{\partial x_r} & \frac{\partial R_r^2(t)}{\partial y_r} & \frac{\partial R_r^2(t)}{\partial z_r} & \frac{\partial R_r^2(t)}{\partial \delta_r} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial R_r^n(t)}{\partial x_r} & \frac{\partial R_r^n(t)}{\partial y_r} & \frac{\partial R_r^n(t)}{\partial z_r} & \frac{\partial R_r^n(t)}{\partial \delta_r} \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \\ \delta_r \end{bmatrix} \quad (4.15)$$

with the correction term

$$D^s(t) = -c\delta^s(t) + \Delta T_r^s(t) + \Delta I_r^s(t) + c TGD^s(t) \quad (4.16)$$

In equation 4.15, the left hand side corresponds with the vector dl in section 2.5.10. The geometric range $\rho_r^n(t)$ is calculated with the approximate values. With the non-linear least-squares adjustment, only the improvements of the starting values can be calculated. The non-linear least-squares adjustment is described in detail in section 2.5.10. The assumption for the variance factor of the observations with $\sigma_0^2 = 1$ was chosen for all investigations in this master thesis.

The absolute position, which is estimated, can be calculated using the following equation:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} x_r^0 \\ y_r^0 \\ z_r^0 \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad (4.17)$$

The computation of the receiver position is an iterative process with the abort condition that in case two consecutive iterations provide a receiver position, it does not differ more than 10 *cm*. The approximate coordinates for the first epoch are provided in formula 4.7, for further epochs the approximate position is set with the position of the previous epoch.

4.3 Android Application

In this section, the resulting software for the Single Point Positioning (SPP) on a mobile device is presented, as well as a brief description of the software structure.

In Figure 4.1, the icon of the project’s Android application is shown. The app is named “Smartphone-SPP” and runs only on mobile devices with the software platform Android. The minimum required API level is 24. Some of the mobile devices on which the application runs are provided in section 3.2.

There are no additional files or programs necessary to run the app. The only requirement is that the smartphone is connected to the internet. The internet connection is required in order to download the broadcast ephemerides from the server.



Figure 4.1: App icon

4.3.1 Software Structure

In this section, a short overview of the resulting Android application is given. The software is written in the program language Java. To create the app, the software “Android studio 3.1.4” was used. It is based on an object-oriented approach with several classes, as well as two classes from the Android Location API, `GnssClock` and `GnssMeasurement`. The class `GnssClock` represents the time measurements of the smartphone’s clock. The second class includes the satellite observations as well as computed values [3]. A list of the implemented classes of the project with a short description is provided in the following paragraphs:

`MainActivity`

This is the main class of the program and conducts the entire process of the app. It extends from `AppCompatActivity` and is the application’s entry point. Moreover, the callbacks `OnMapReadyCallback` and `SharedPreferences.OnSharedPreferenceChangeListener` are implemented in this class. The callback `OnMapReadyCallback` is included in order to prepare the Google map for plotting the positions in the app. `SharedPreferences.OnSharedPreferenceChangeListener` is implemented that changes in the settings are registered when the app is already running.

NavdataDownload

The task of this class is to communicate with a server of the Graz University of Technology and download the broadcast ephemerides. From that server, the current ephemerides are downloaded at the beginning of the measurement process. Additionally, after every 300 epochs, the latest GPS ephemerides are downloaded again.

Measurement

In this class, the received raw measurements are stored for further computations. The generation of the pseudorange also takes place in this part of the program. Furthermore, a flag is set to determine whether a measurement is valid or not. If an observation passes the filter which is described in section 4.2.2, the flag is set to `true`, otherwise it is set to `false`.

Satellite

The downloaded broadcast ephemerides are structured using instances of this class. In addition, the eight ionospheric parameters alpha and beta are stored to a vector. Although the actual ephemerides from all GPS satellites are downloaded, an object of this class is created only for visible satellites.

Epoch

The computations of the Single Point Positioning (SPP) are conducted in this class. The input are the valid raw measurements from the class `Measurement` and the ephemerides from the class `Satellite`, as well as the stored settings, and the approximate coordinates. An iterative least-squares adjustment is performed to calculate the smartphone position for each epoch.

SettingsActivity & SettingsFragment

These classes manage all settings which are chosen by the user. This includes the settings of the atmospheric correction, the weighting method, and whether or not output files should be created.

GnssDataLogger

This part of the program is responsible for obtaining the Android raw measurements from the phone's chip. This includes both the information of the `GnssClock` and the `GnssMeasurement` class.

The constructor of the `MainActivity` class runs when the app is started, and prepares all settings, and collects all information which is necessary for further computation steps.

Moreover, the access to the required permission for the location information on the phone takes place in the constructor. Furthermore, a `BroadcastReceiver` was used to obtain the Android raw measurements from the class `GnssDataLogger` via intent for usage in the program.

The following flowchart shows the procedure of the program schematically:

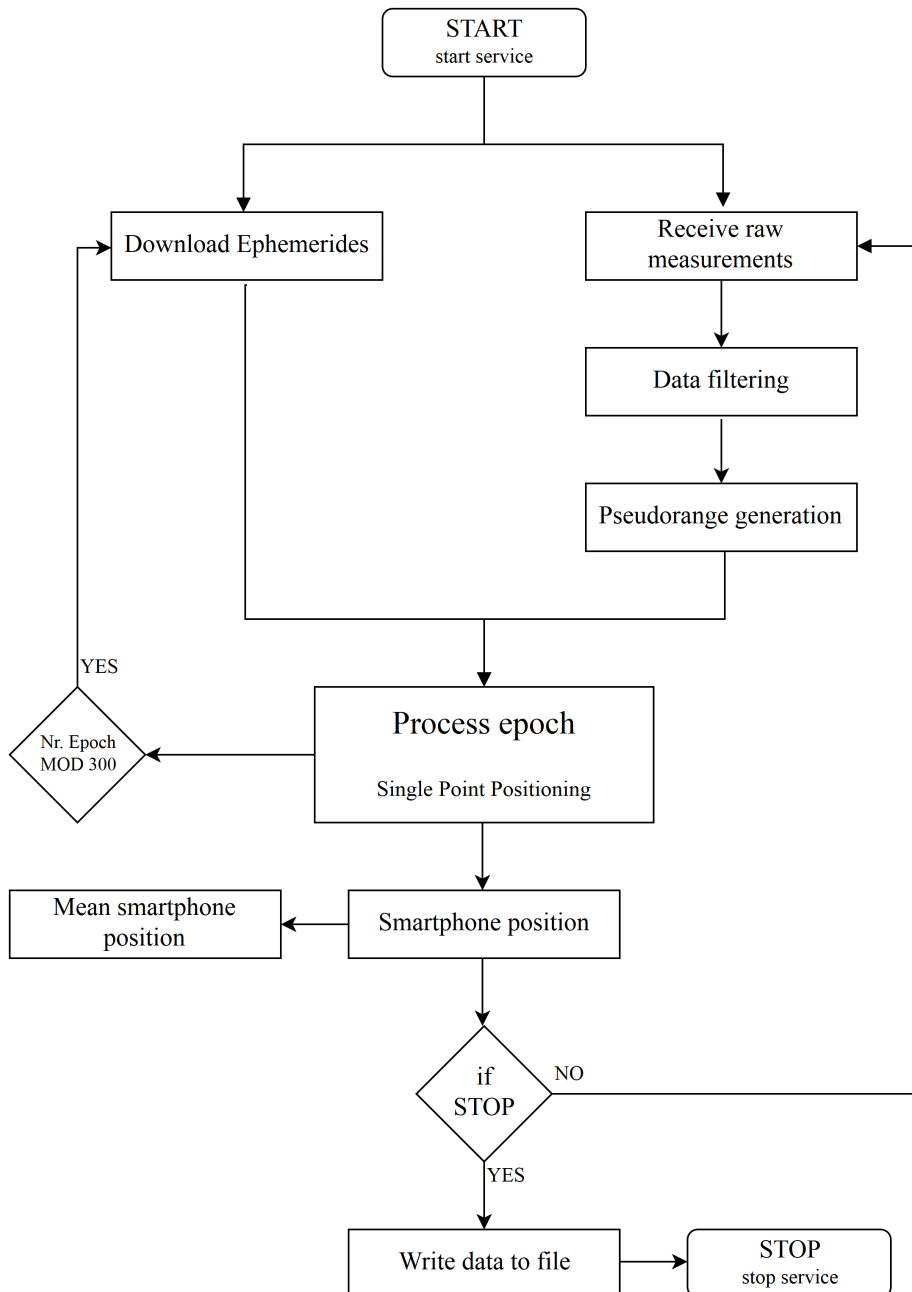


Figure 4.2: Program structure - flowchart

4.3.2 Software Design

The starting window and the setting window of the app are presented in figure 4.3:

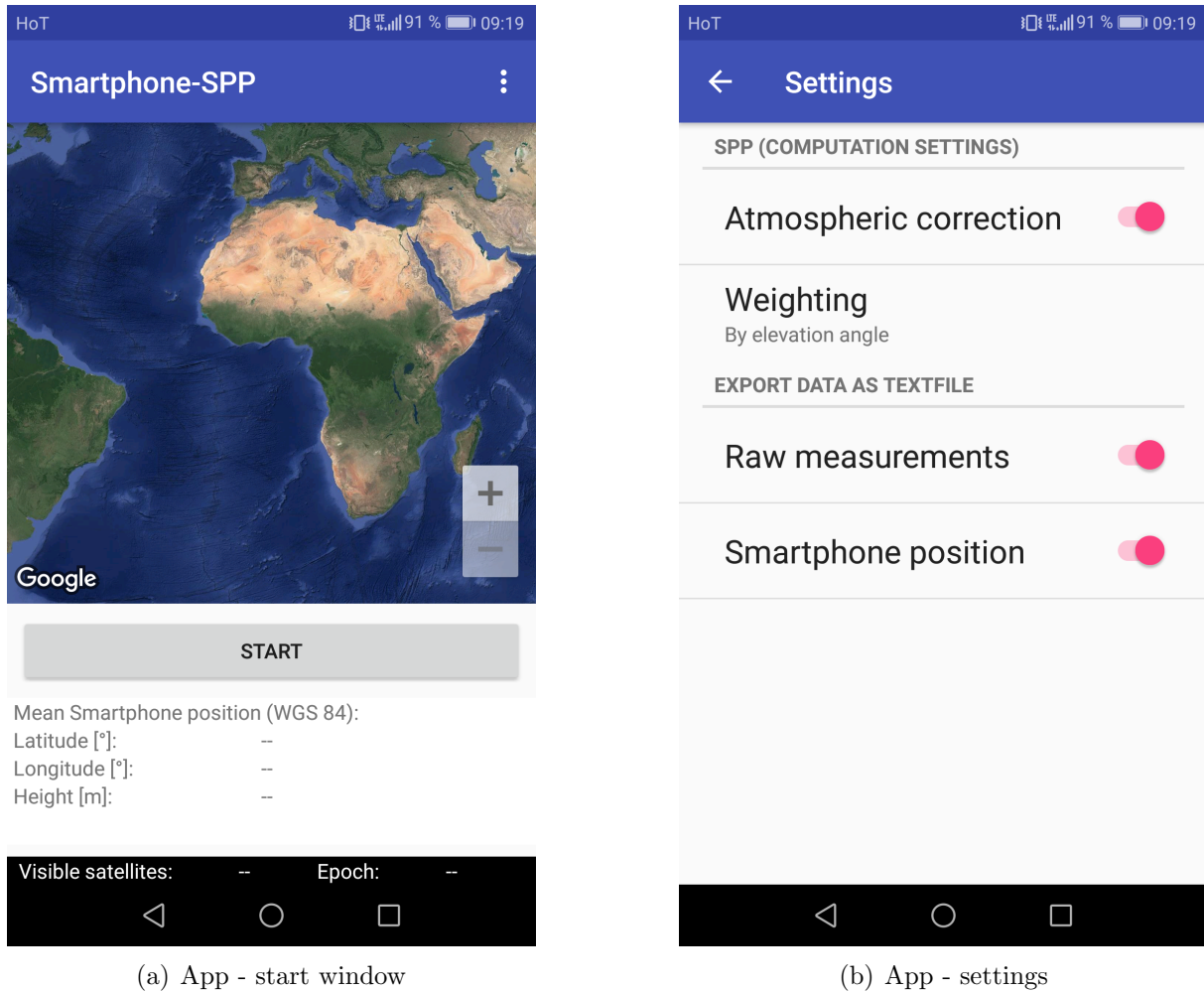


Figure 4.3: App - screenshots

As can be seen in the left picture, there are several pieces of information which are provided for the user. First of all, there is a graphical representation of the world. This is visualized through a Google Map API. Directly below the “START“ button, the mean smartphone position of all epochs is shown. The geographic latitude and longitude are expressed in degrees, and the ellipsoidal height is given in meters. These values refer to the WGS84 coordinate system. Moreover, the visible satellite number and the number of epochs are provided.

The right figure shows the settings menu of the application. There, the user can choose if an atmospheric correction should be attached as well as the weighting method. All three weighting methods which are described in section 4.2.3 can be used for the computations.

As a last option, users can decide whether the Android raw measurements which are used for the computations or the calculated smartphone positions should be written to an output file. These two files are saved on the local storage of the phone.

In the following screenshot, the app is in action. The computed positions in real time are shown on the map of the app.

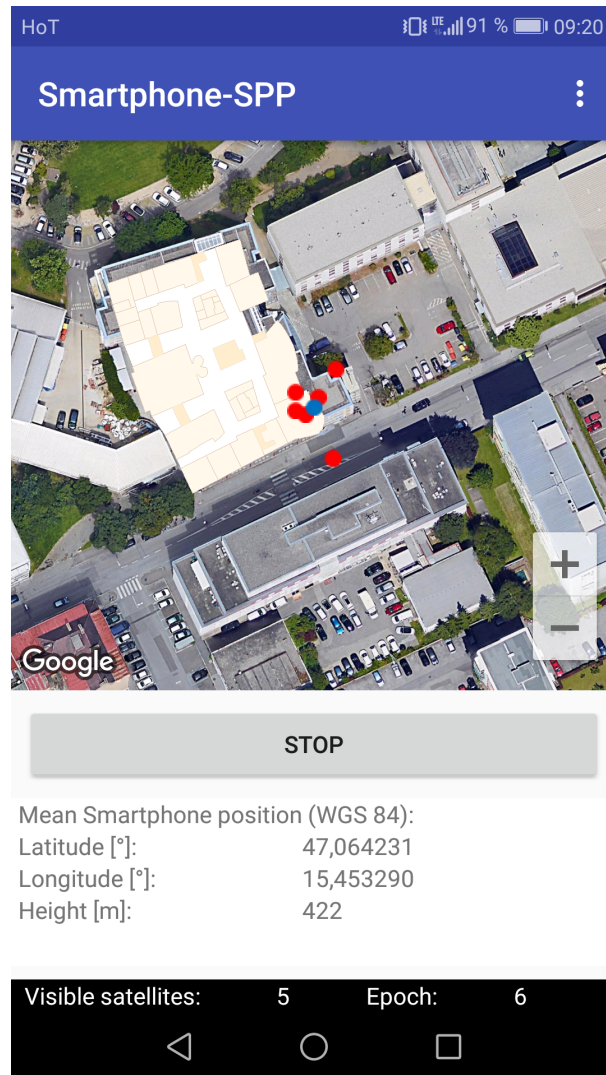


Figure 4.4: App - measuring

The red marker represents the position solution from one single epoch. The blue marker in the map indicates the mean smartphone position. The map window is focused at this point.

The structure of the output file of the position data is shown in the following figure:

```

1 # Smartphone-SPP with Android raw measurements
2 #
3 # Reference system:           WGS84
4 # Ionospheric correction:     Klobuchar model
5 # Tropospheric correction:    Hopfield model
6 # Weighting:                 Elevation angle
7 #
8 # Pos,TimeUTC,LatitudeDegree,LongitudeDegree,AltitudeMeter,North_
9 #
10 Pos,07:20:08,47.064253,15.453305,422,0.05,0.05,4.78
11 Pos,07:20:09,47.064223,15.453231,437,0.05,0.05,4.78
12 Pos,07:20:10,47.064264,15.453231,428,0.05,0.05,4.78
13 Pos,07:20:11,47.064215,15.453264,422,0.05,0.05,4.78
14 Pos,07:20:12,47.064314,15.453360,429,0.05,0.05,4.78
15 Pos,07:20:13,47.064119,15.453351,395,0.05,0.05,4.78

```

Figure 4.5: Output file - position data

In the header, which is indicated by the symbol “ #, “ all relevant information is included. Moreover, a description of the data section is provided in line eight. The first entry is “Pos,“ which indicates that this is a position. The second parameter is the time given in UTC. The geographical latitude and longitude are provided in degree as well as the ellipsoidal height in meters. The formal errors of these three components are the last entries in the file. The formal accuracy of φ and λ are given in arcseconds, and that of the height is expressed in meters.

In figure 4.6, the Android raw data which are used for the computation of the Single Point Positioning are shown:

```

1 # Smartphone-SPP with Android raw measurements
2 #
3 # GPS raw measurements
4 #
5 # Raw,ElapsedRealtimeMillis,TimeNanos,LeapSecond,TimeUncertaintyNanos,Ful
6 #
7 Raw,4536906,14885000000,,,-1222759211115311318,0.000000,12.646649,,0,8,0
8 Raw,4536908,14885000000,,,-1222759211115311318,0.000000,12.646649,,0,10,
9 Raw,4536909,14885000000,,,-1222759211115311318,0.000000,12.646649,,0,20,
10 Raw,4536911,14885000000,,,-1222759211115311318,0.000000,12.646649,,0,21,
11 Raw,4536912,14885000000,,,-1222759211115311318,0.000000,12.646649,,0,27,
12 Raw,4537888,15885000000,,,-1222759211115311023,0.000000,12.591418,,0,8,0
13 Raw,4537890,15885000000,,,-1222759211115311023,0.000000,12.591418,,0,10,

```

Figure 4.6: Output file - raw data

The data section of the output file includes all parameters which are described in 3.4. The units of all parameters are exactly the same as in the “GNSSLogger“ output file (see sec. 3.4). In the header a description of the data section is given.

Chapter 5

Results and Evaluation

The results of this master thesis are described in this chapter, and the different investigations are divided into the following sections. The majority of the raw data which are presented in this chapter was recorded with the “GNSSLogger“ application. The rest is from the project’s own Android app.

For the comparison of the Android raw measurements, a total of three smartphones are used. The main investigations were made with the Huawei Honor 8. The other test devices are the Huawei P10 and the Samsung Galaxy S8.

The evaluation of the positioning algorithm and the analysis of the Android raw measurements are carried out with two additional programs. The first one is RTKLib (ver. 2.4.2), and the second one is GNSS Analysis Tool, which is provided by Google and is described in section 3.5. In the GNSS analysis program, the recorded raw measurements can be loaded directly. The output file of this program is a .kml file. To load this file in Matlab, the function from [18] is used. In contrast, RTKLib cannot load the Android raw data. Therefore, a Matlab script was created to transform the Android raw measurement file to the standard file RINEX. All processing programs (with the exception of the smartphone’s internal solution) only use GPS observations.

All figures in this chapter were created with the software Matlab. In some figures, a Google Maps API is plotted in the background. To plot this, the Matlab function from [22] is used. Furthermore, some screenshots from the Android application are presented. Additionally, all results from sections 5.1 to 5.4 were computed with Matlab in post processing. The results which are computed in real time are presented in section 5.5.

The test measurements were performed at a pillar on the roof of the Graz University of Technology. The reference coordinates from this pillar were provided by the Institute of Geodesy. The numerical values from this point are given in A.2.

5.1 Test Measurement

In this section, one measurement campaign is presented in detail. The measurement campaign took place on the 6th of February, 2018 on the rooftop of the Graz University of Technology at Steyrergasse 30. This measurement was accomplished with the Huawei Honor 8 smartphone. Figure 5.1 shows the measurement situation:



Figure 5.1: Measurement situation

The measuring period was approximately 10 minutes, and the measurement data was recorded with a frequency of 1 Hz. The Android raw data from this measurement campaign were logged with the “GNSSLogger“ application. Figure 5.2 shows the number of visible GPS satellites throughout the whole measurement period.

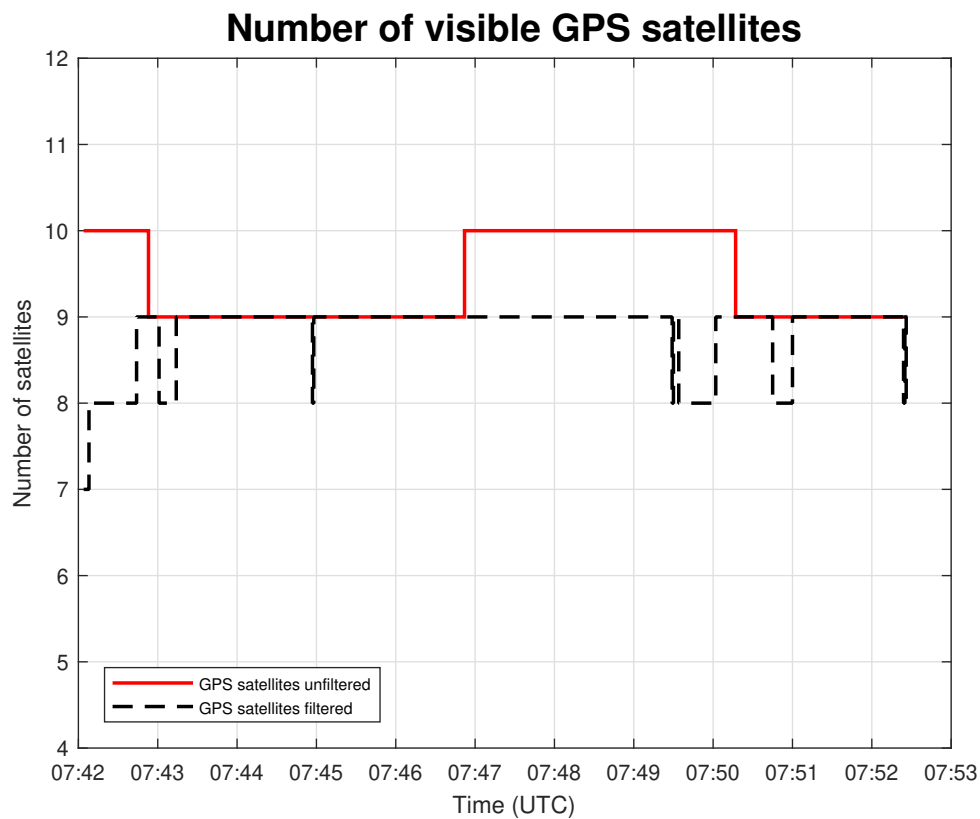


Figure 5.2: Number of visible satellites

In the diagram above, the number of visible GPS satellites before and after the filtering process are visualised. As can be seen, in the first seconds of the measurement campaign, three satellites were filtered out. For about half of the measurement period, no satellite is removed from the computations.

Generally, it can be said that a total number of eight satellites throughout nearly the whole measurement period is very good in relation to an overdetermined system.

Other measurement campaigns show that it is also possible that the difference between the two lines in figure 5.2 can be higher. It can reach up to three or even four satellites, which were filtered out of the computation steps.

5.1.1 Carrier-to-noise Density and Elevation

In the next figure, the Carrier-to-noise density (left plot) and the elevation angle (right plot) for each visible GPS satellite are plotted:

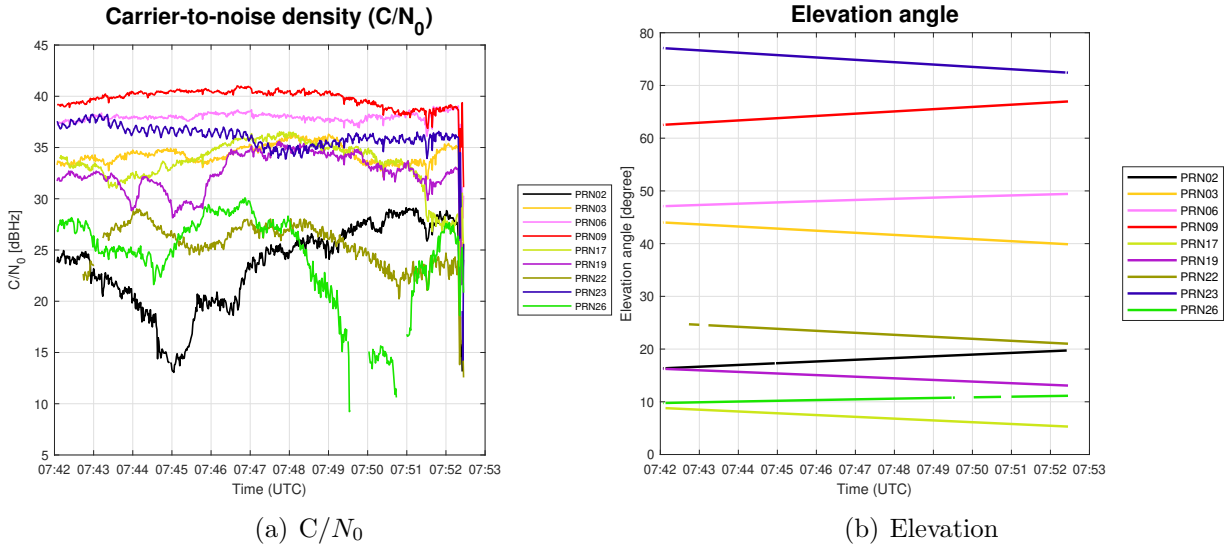


Figure 5.3: Android raw data

The Carrier-to-noise density ratio gives us information about the received signal strength. As can be seen in figure 5.3, the range of the Carrier-to-noise density goes from 10 to approximately 40 dBHz. A value of 10 dBHz implies a very bad signal quality. Signals over 25 to 30 dBHz are good signals for the position estimation and could achieve a higher weight if weighting with the Carrier-to-noise density is performed.

As far as comparing the C/N_0 with the elevation angle is concerned, it can be said that in most cases, if the elevation of a satellite is higher, this applies to the Carrier-to-noise density as well, in contrast to satellites with a low elevation.

The gaps in the elevation angle, respectively in the Carrier-to-noise density ratio, indicate that the smartphone filtered this satellite out of the raw data in these epochs.

It can also be said that the C/N_0 from all satellites goes down at the end of the measurement campaign. The reason for that is that in the last few seconds, the signals were shaded by the user when he turned off the app which writes the raw data to a file.

5.1.2 Position Solution

For the investigations in this master thesis, a total of four solutions of the position were created and contrasted. The following methods were calculated:

- 1. Option: SPP (GPS)
- 2. Option: SPP (GPS) + Weighting (by elevation)
- 3. Option: SPP (GPS) + Atmospheric corrections
- 4. Option: SPP (GPS) + Weighting (by elevation) + Atmospheric corrections

A graphical comparison of the mean position in the Gauß-Krüger plane of all options is given in the next figure:

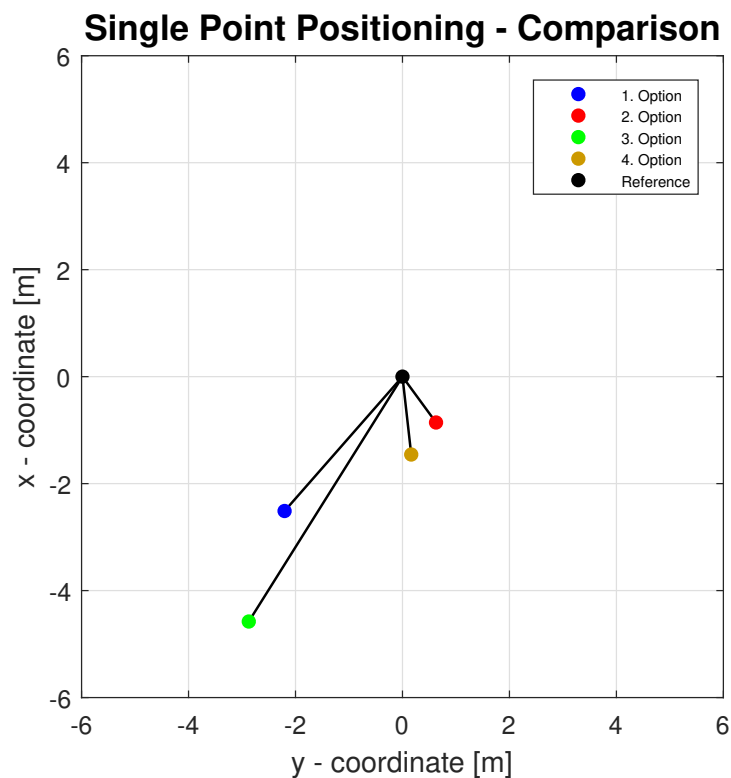


Figure 5.4: Comparison of all options

As can be seen in the figure above, the second option provides the best solution with respect to the reference position. Although this option provides the best solution in this case, the fourth option is used for further investigations.

The reason for that is that further measurement campaigns show that the fourth option is slightly better.

Figure 5.4 shows that the influence of a weighting at this measurement campaign is significant, while the influence of the atmosphere is not so high. One reason why the atmospheric influence is minor is that the tropospheric, and especially the ionospheric activities at the time of measurement were not very high. A further possibility is that the least-squares adjustment absorbed the influence. At this measurement campaign, the consideration of the atmosphere produces even a slightly worse position solution. One reason could be that the atmosphere is only approximated by models.

The second and fourth options were weighted by the elevation angle and not by the Carrier-to-noise density. The investigations showed that the weighting by the elevation angle leads to better results. This weighting method could be problematic if the smartphone positioning is performed in a city with tall buildings around the measurement point. The reason for that is that signals of satellites with a high elevation angle could also be influenced by multipath, and therefore these signals are not optimal for the positioning algorithm. Since satellites with a high elevation could also have a bad signal quality, the weighting with the Carrier-to-noise density ratio is more useful.

As a result, the experienced user has the possibility to weight the measurements with the carrier-to-noise density in the Android application.

In the following table, the numeric comparison of the mean positions of all options is provided. The results are presented in the Gauß-Krüger projection and pertain to the reference position. The standard deviations (see equation 2.60 to 2.62) of the mean position are given as well.

Table 5.1: Numeric comparison of all options

Option	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
1	-2.6	-2.2	3.4	10.1	6.5	12.0
2	-0.9	0.6	1.1	9.1	4.5	10.2
3	-4.7	-2.9	5.5	9.6	6.1	11.4
4	-1.5	0.2	1.5	8.9	4.3	9.9

In table 5.1, it can be seen that the results of the different options in relation to the standard deviation are similar. The standard deviation shows that the second and fourth options lead to a similar solution and are better than the other two options.

In the next figure, the WGS84 positions of option four from all epochs are plotted. The smartphone position coordinates are represented in the geographic longitude and latitude. Moreover, the reference point and the mean position over the entire measurement time are shown in figure 5.5. For the background, a Google Map API is used.

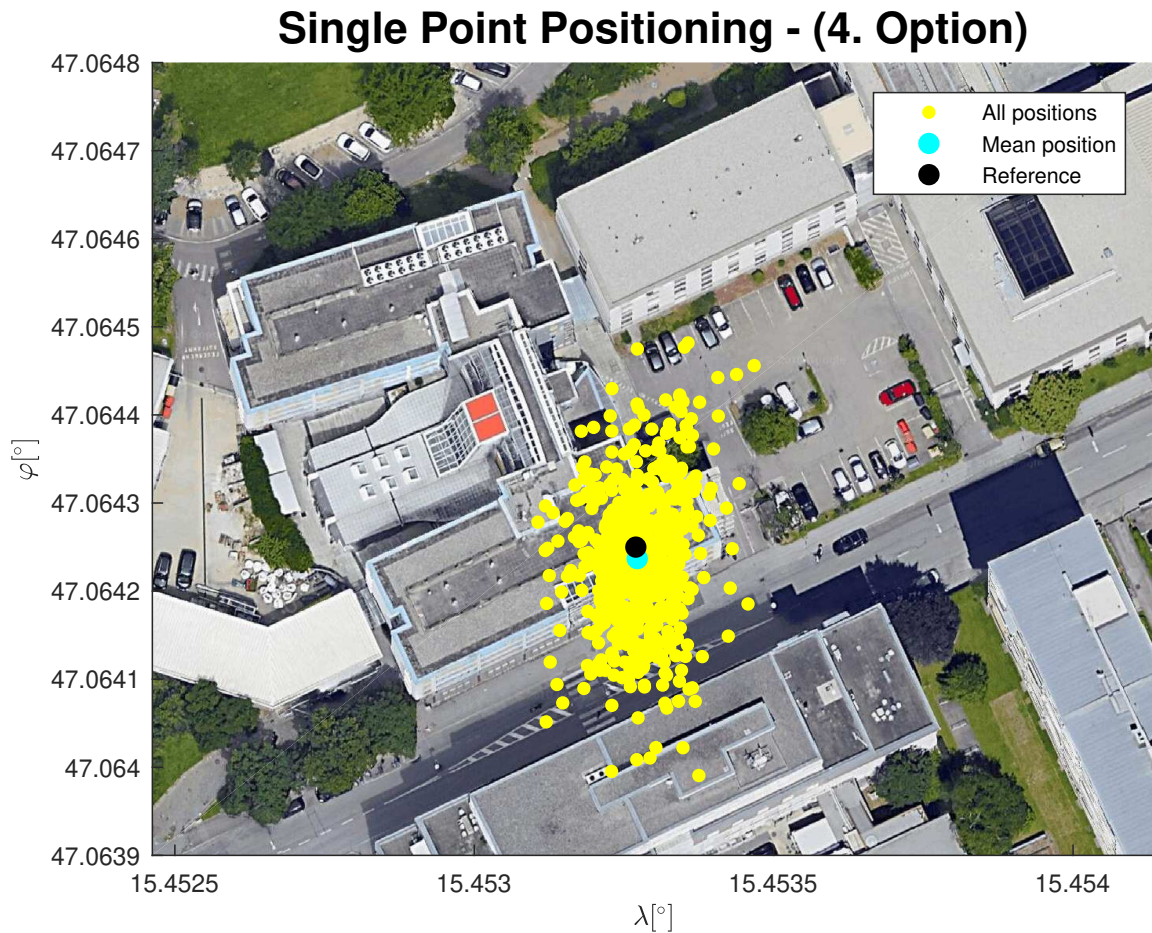


Figure 5.5: Position of all epochs

It can be seen that the spread of the positions is worse in the north-south component than in the east-west component. This fact can also be observed in the time series in the next graph. A reason for that can be the geometry of the monitored satellites and possible shadowing of buildings in the surrounding area. This can also be observed in other measurement campaigns.

A further aspect to be considered is that the mean position over the whole time span matches the true position very well.

All test measurements show that with code pseudoranges, only a meter-level accuracy can be achieved. This fact is also a result of the investigations in [1].

Figure 5.6 shows the difference between the self-calculated position and the reference position at each epoch for every component:

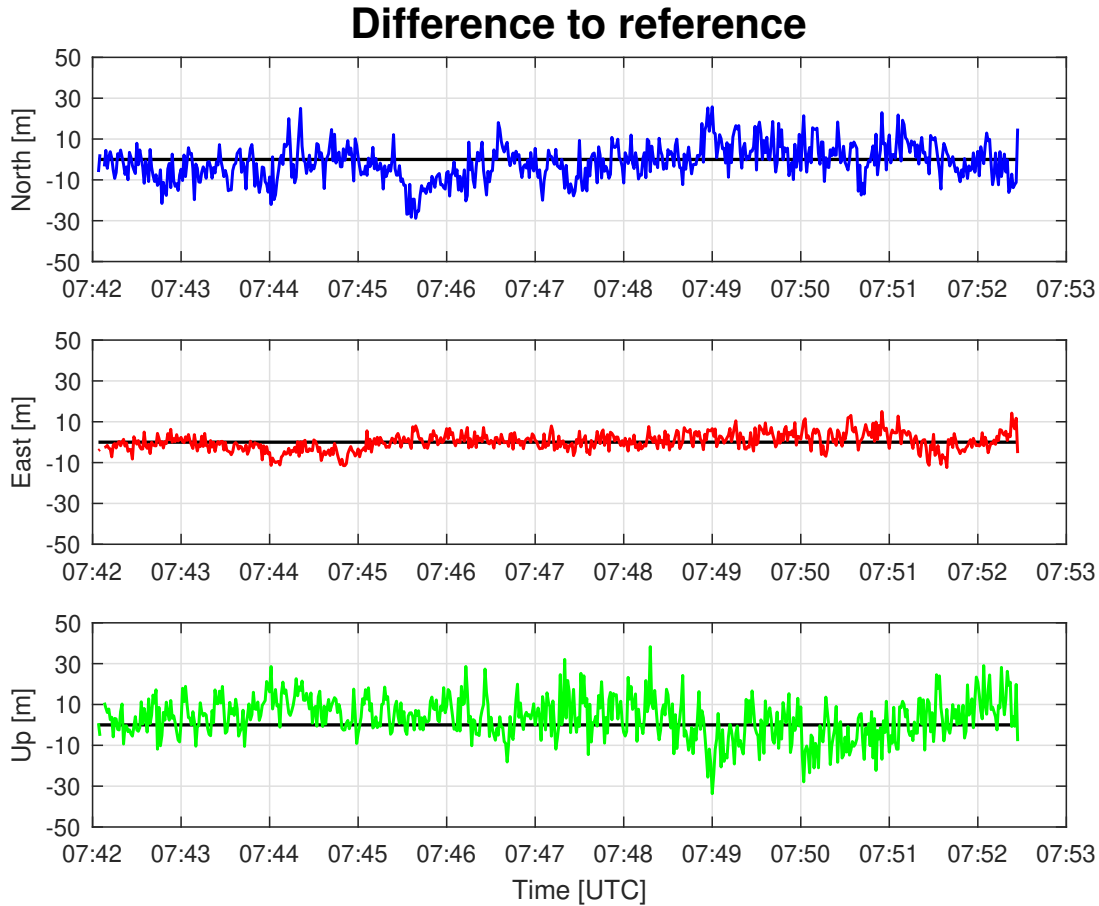


Figure 5.6: Time series of the difference to the reference point

As the figure above demonstrates, the difference between the reference and the self-calculated position solutions changes. The east component shows a better performance than the north and up component. The position solution fluctuates in a range of about ± 10 meters in the second component. The difference to the reference point in the north and up component reaches 30 meters. A reason why the east component is better than the north component could be the satellite geometry or the shadowing of buildings. These can also be seen in other analysis programs, for instance in RTKLib. Moreover, the higher differences in the up component are a typical effect of GNSS measurements. Reasons for that could be the atmospheric influence, which is not completely eliminated through the model. Further aspects are the satellite geometry and the direction which the signals comes from.

In the entire time series, no clear trend is visible.

5.1.3 Comparison to Other Tools

For the evaluation of the implemented positioning algorithm, other tools which can process GNSS data were used. RTKLib and the Analysis Tool from Google were relied on for the verification of the project's own SPP algorithm. In addition, the smartphone's internal solution from the GNSS chip is plotted in the next figure:

Single Point Positioning - Comparison to other tools

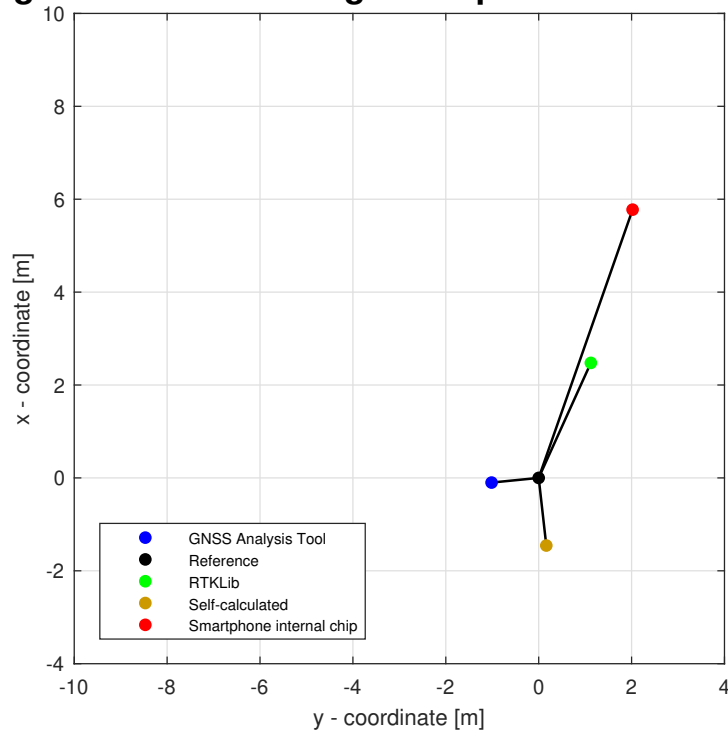


Figure 5.7: Comparison to other tools - mean position

As can be seen, the self-calculated position is better than both the solutions from RTKLib and the phones's internal positioning solution. In figure 5.7, the GNSS Analysis Tool provides the best solution in relation to the reference position. This program weights the measurements through the uncertainty of the measurement [3].

In the following figure, the residuals of the different tools to the reference point are plotted. The absolute differences are calculated from the x - and y - coordinates in the Gauß-Krüger plane.

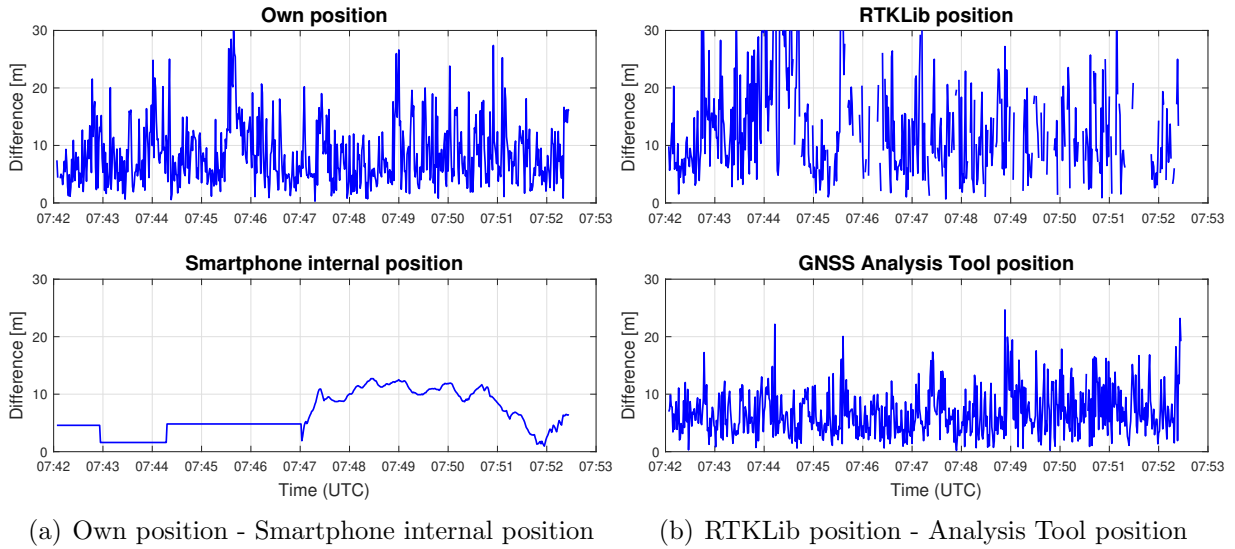


Figure 5.8: Comparison with all evaluation tools - time series

In the subfigure above, it can be seen that the time series of the smartphone’s internal solution has the smoothest form. This indicates that the GNSS chip in the phone perhaps perform a position filtering to achieve such a smooth position throughout the entire measurement time. It also seems that the smartphone’s GNSS chip holds the same position over a longer time span, as can be seen in the first five minutes. Furthermore, the smartphone’s internal GNSS chip uses more sensors to determine the position [3].

The other three time series show a similar structure. The fluctuation range is up to 20 or 30 meters in all three time series. Due to the results of the comparison between the project’s own positioning algorithm and other tools, it can be concluded that the project’s own positioning has been correctly implemented.

5.2 Multiple Test Measurements

In this section, the results of several measurement campaigns are listed in detail. In all measurements, a period of approximately ten minutes was chosen, and as test equipment, the Huawei Honor 8 was selected. In the following table, the results of the four different options which are described in section 5.1.2 are compared to each other. Again, Δd_{ref} is the absolute difference to the reference position in the Gauß-Krüger plane.

Table 5.2: Comparison of multiple test measurements

Measurement date	Δd_{ref} [m]			
	Option 1	Option 2	Option 3	Option 4
Feb 6th, 2018	3.4	1.1	5.5	1.5
Apr 9th, 2018	10.9	1.0	3.4	1.7
May 14th, 2018	4.5	4.0	3.5	3.3
May 16th, 2018	10.0	10.2	8.6	8.8
May 23th, 2018	1.6	0.9	1.5	2.0
May 30th, 2018	6.2	1.3	2.7	1.2
Jul 10th, 2018	7.1	6.3	5.7	5.1

As can be seen in table 5.2, the results of the different measurement campaigns vary. Moreover, the influence of the atmosphere as well as the weighting differs. Although some options yield similar results, all in all it can be said that the fourth possibility is the most favourable one, as was to be expected.

In the measurement on April 9th, 2018, the influence of the weighting was extreme in contrast to the measurement on May 16th, 2018, and the weighting results in a worse solution than without weighting. In this case, the satellites with the highest Carrier-to-noise density ratio (40 to 45 dBHz) have an elevation angle of just 40 to 50 degrees. Consequently, those measurements have only a minor influence on the solution, although the signal strength is very good.

A closer look at table 5.2, said that the influence of the atmosphere decreases when the solution is weighted by the elevation angle. This is based on the fact that satellites with a low elevation angle receive a lower weight and consequently have only a minor influence at the position solution. The reason for that is that those satellites have a higher atmospheric correction.

In table 5.3, the fourth option is compared to the position solution of other tools.

Table 5.3: Comparison with other tools - mean position

Measurement date	Δd_{ref} [m]			
	Phone's GNSS Chip	RTKLib	Analysis Tool	Option 4
Feb 6th, 2018	6.1	2.7	1.0	1.5
Apr 9th, 2018	1.2	2.8	0.4	1.7
May 14th, 2018	0.4	3.4	3.5	3.3
May 16th, 2018	1.1	3.1	1.8	8.8
May 23th, 2018	6.8	6.6	2.6	2.0
May 30th, 2018	3.2	0.8	3.2	1.2
Jul 10th, 2018	2.9	6.3	4.0	5.1

As can be seen in table 5.3, the position solution differs from measurement to measurement. The smartphone’s internal position, for instance, achieves the best solution at 0.4 meters, and the worst position at 6.8 meters to the reference point. Reasons for that can be for instance the different geometry of the satellites or the atmospheric conditions, as well as the signal quality. A similar result can be observed in the position solutions from RTKLib and the Analysis Tool. Furthermore, the self-calculated position is very good in some cases but worse in others.

A reason for the different solutions of the tools can be that every tool has its specific filtering algorithm of the data, and in addition perhaps a slightly different implementation of the Single Point Positioning. In fact, also the weighting can cause such differences. The self-calculated position is weighted by the elevation angle, and the Analysis Tool is weighted by the uncertainty of the measurements [3], for instance. There is no information on how the phone’s internal GNSS chip or RTKLib weights the measurements for their positioning algorithm. The investigations of the different weightings of the project’s own position show that in some cases, the difference is enormous.

In the following table, the standard deviation after Helmert of all tools are presented:

Table 5.4: Comparison with other tools - standard deviation

Measurement date	$\sigma_{Helmert}$ [m]			
	Phone’s GNSS Chip	RTKLib	Analyse tool	Option 4
Feb 6th, 2018	4.0	15.2	7.8	9.9
Apr 9th, 2018	5.9	26.8	14.2	21.0
May 14th, 2018	4.3	9.5	9.3	9.8
May 16th, 2018	2.4	9.0	9.1	14.7
May 23th, 2018	4.0	18.5	9.1	11.0
May 30th, 2018	2.6	13.4	7.9	8.5
Jul 10th, 2018	4.5	31.2	8.8	8.9

Table 5.4 shows that the standard deviation differs from tool to tool as well. What can be seen instantly is that RTKLib has the worst standard deviations at the most measurements. A detailed look at the position solutions from April 9th and July 10th shows that the spread of the position solution is bigger than in other measurement campaigns.

The standard deviation from the smartphone’s internal position provides the best values at every measurement campaign. This is a clear indication that the GNSS chip in the phone perhaps carries out a position filter.

Option four also has a standard deviation on the same scale as the other tools.

5.3 Android Raw Data Analysis and Comparison

In this section, the results of three different smartphones are compared. The Android raw measurements of the different mobile devices were measured simultaneously at the “same“ reference point. The measurement campaign took place on the 24th of August, 2018 at the rooftop of the University of Technology. The measurement situation is shown in the next picture:



Figure 5.9: Measurement situation - comparison

As can be seen, the smartphones do not have precisely the same measurement point. The only possibility to measure exactly the same point is to use an external antenna with a signal splitter.

As these investigations focus on a Single Point Positioning, this difference can be neglected.

The measurement period of all three smartphones was approximately 10 minutes and took place at the same position as in section 5.1. The results from all smartphones are compared to each other in the following sections.

5.3.1 Number of Satellites

This section deals with a detailed investigation of the number of visible GPS satellites on all three mobile devices. In subfigure 5.10, the satellite number is visible:

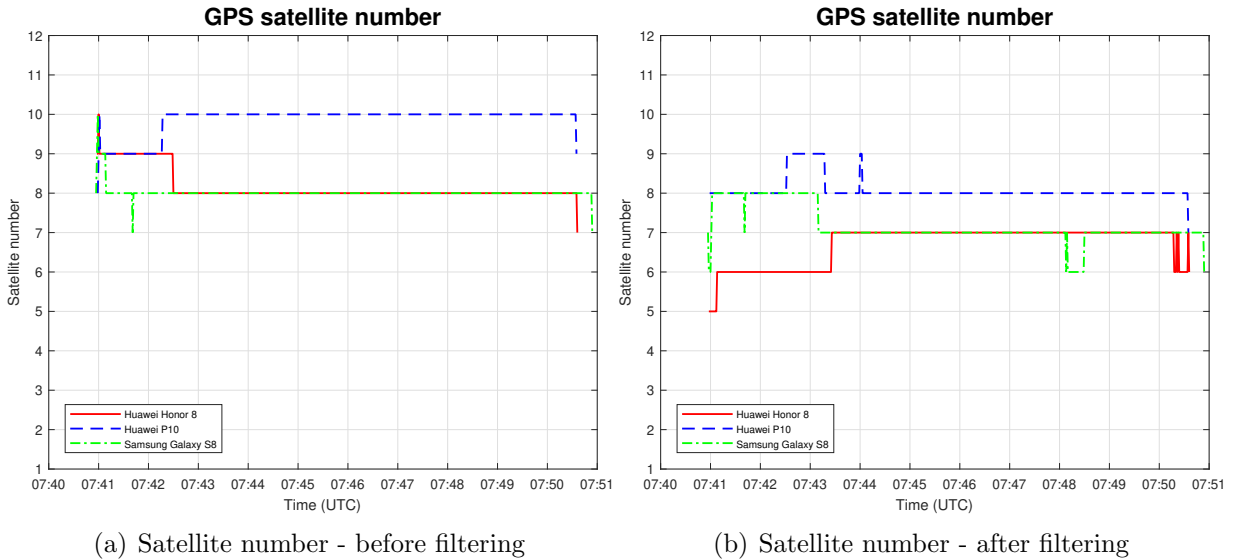


Figure 5.10: Comparison of the visible GPS satellites

The difference between the unfiltered number of satellites and the filtered number of satellites is clearly visible when both subfigures are compared to each other.

In fact, the left figure shows that the number of visible satellites differs among the three smartphones. The Huawei P10 has two more satellites than the other two smartphones most of the time.

A reason why the smartphones do not have the same number of satellites could be a different tracking status. Consequently, the Huawei P10 could track two more satellites at the same position and the same time as the other two smartphones. As can be seen in section 5.3.3, the smartphones receive the same satellite signals with different signal strengths. Therefore, if the strength of the signal is low, it could be possible for one smartphone to track this satellite, but for an other smartphone to fail.

In figure 5.10, the influence of the filter becomes obvious in the number of satellites. It can be seen that the most satellites are filtered out from the Huawei P10. Furthermore, after the filtering process, the Huawei P10 has the highest number of satellites throughout the majority of the measurement time.

5.3.2 Data Filtering

In the next subfigure, the left plot represents the number of satellites which are excluded because of their “ReceivedSvTimeUncertainty,“ and the right plot shows those which have been removed due to their wrong tracking status:

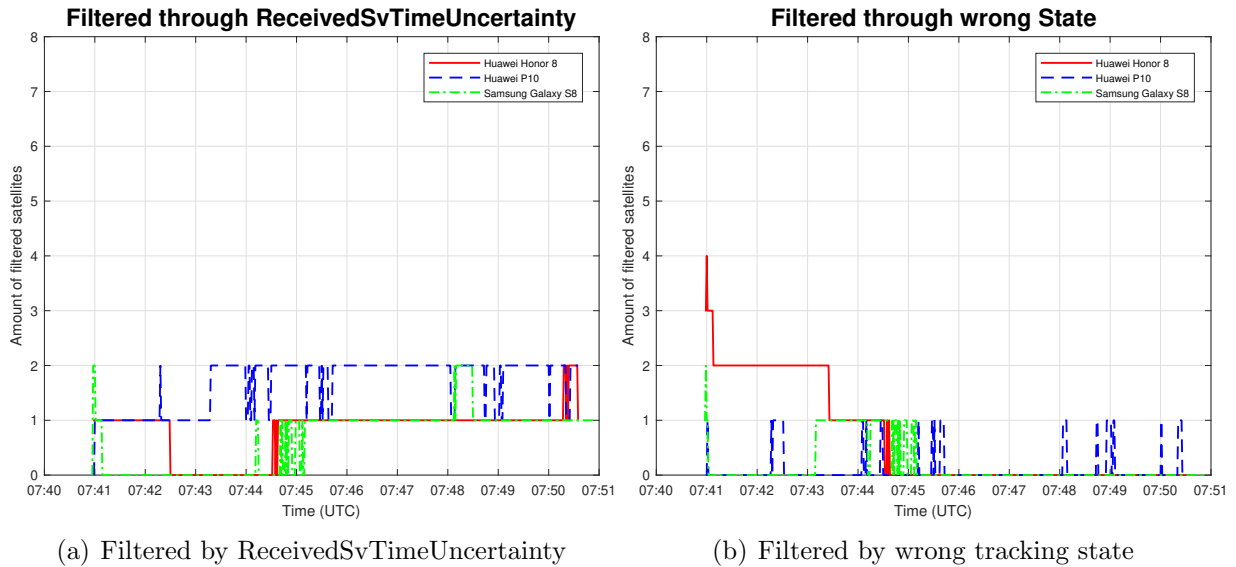


Figure 5.11: Data filtering - amount of satellites

In the left plot, the measurements with a “ReceivedSvTimeUncertainty“ of 1 000 000 000 nanoseconds are filtered out of the raw data. In the right figure, satellites with the wrong tracking status are cancelled out.

As can be seen, the number of filtered satellites strongly fluctuates. It is also visible that most of the time, one or two satellites are eliminated through the filter. This occurs in all three smartphones. In figure 5.11(b), it can be seen that the Huawei Honor 8 has four satellites with the wrong tracking state at the beginning of the measurement period. After a few minutes, this number goes down to two and later to one satellite. It should also be mentioned that observations could also have a bad “ReceivedSvTimeUncertainty“ and a wrong tracking status. These observations are included in 5.11(a).

5.3.3 Carrier-to-noise Density (C/N_0)

The Carrier-to-noise density of all smartphones is plotted in the next subfigures. Four satellites were chosen indiscriminately:

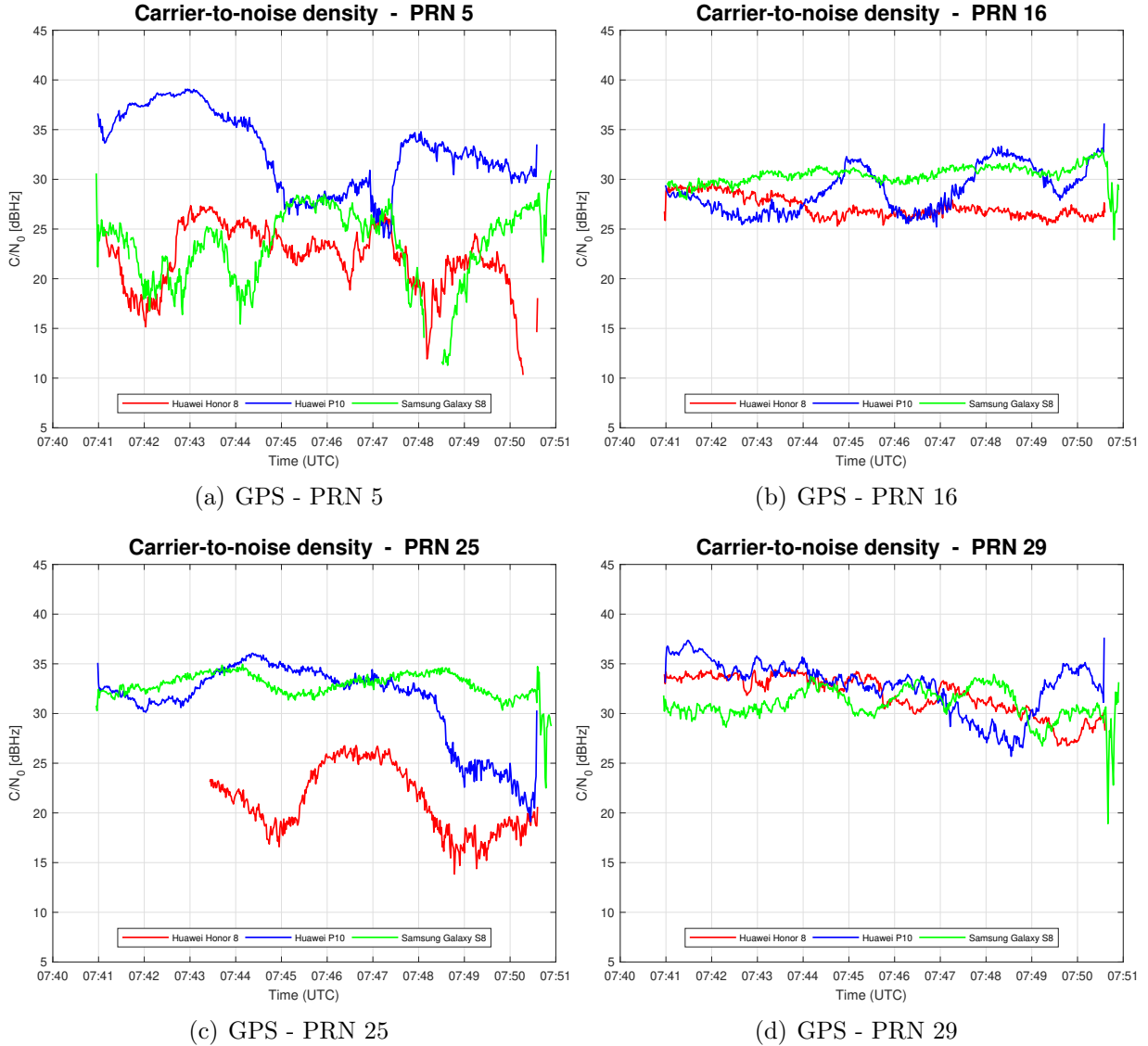


Figure 5.12: Comparison - Carrier-to-noise density

In figure 5.12, it can be seen that the Carrier-to-noise density can vary from smartphone to smartphone. Differences up to 10 or 20 dBHz can occur (see. 5.12(c)). By looking at this parameter, no statement which smartphones provides the best values can be made, however. For instance, the Huawei P10 gets a better signal quality from the satellite with the PRN number 5 (see. 5.12(a)), but for the PRN number 16, the Samsung Galaxy S8 provides the best signal. A reason for that can be the smartphone's internal tracking of the signals, which differs from model to model, or the antenna. The differences in the Carrier-to-noise density between different devices are also noticed in [14] and [16].

5.3.4 Single Point Positioning

In this part, the computed positions of all three smartphones are compared to each other. For the representation of the mean position, the Gauß-Krüger coordinates are chosen. In figure 5.13, the three smartphone positions and the reference point are plotted.

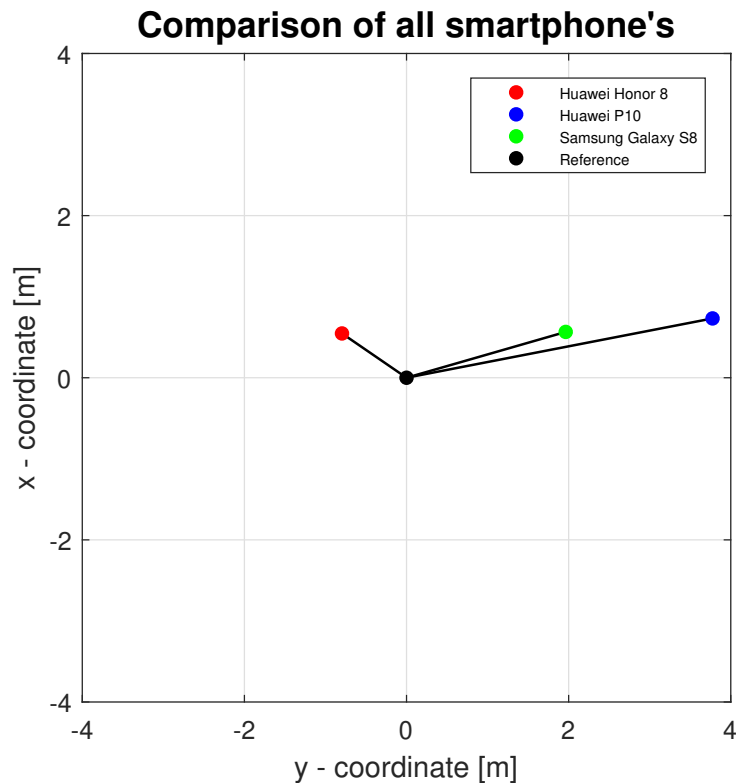


Figure 5.13: Mean position of all three smartphones

In figure 5.13, it can be seen that the difference between the smartphone position is approximately 2 to 4 meters. Furthermore, the absolute offset to the reference point is in the same range. Especially the position solutions from the Huawei P10 and the Samsung Galaxy S8 match very well in this measurement campaign, but further test measurements show that the position of the three smartphones can also differ much more. An overview of multiple measurement campaigns is provided in section 5.4.

The numerical results are presented in the following table. It should also be mentioned that the computed positions are all weighted by the elevation angle, and the atmospheric corrections are considered as well.

Table 5.5: Comparison of all three smartphones

Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	0.5	-0.8	1.0	9.7	8.2	12.7
Huawei P10	0.7	3.8	3.8	5.5	5.3	7.6
Samsung Galaxy S8	0.6	2.0	2.0	5.6	5.0	7.5

As can be seen in the table above, the three mobile devices provide, a similar Δd_{ref} , especially the Huawei Honor 8 and the Samsung Galaxy S8. Moreover, $\sigma_{Helmert}$ is nearly identical for the Huawei P10 and the Samsung Galaxy S8.

The Android raw measurements from all three smartphones were also processed with other tools. The results from RTKLib and the Analysis Tool from Google, as well as the phone's internal position, are provided in the next table.

Table 5.6: Comparison of all three smartphones processed with other tools

Phone's GNSS Chip						
Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	0.9	0.8	1.2	1.1	2.2	2.5
Huawei P10	2.9	-0.2	2.9	4.4	0.3	4.4
Samsung Galaxy S8	1.3	-1.6	2.1	1.3	1.5	1.9
RTKLib						
Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	0.2	-6.8	6.8	16.8	12.3	20.8
Huawei P10	1.2	-1.0	1.5	8.4	6.1	10.3
Samsung Galaxy S8	0.9	-1.2	1.5	6.9	7.4	10.1
Analyse tool						
Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	0.6	-4.4	4.4	6.7	6.8	9.6
Huawei P10	0.7	0.5	0.9	4.5	4.2	6.2
Samsung Galaxy S8	0.7	-0.3	0.8	5.5	4.8	7.3

In table 5.6, it can be seen that the results for Δd_{ref} are very similar for both the Huawei P10 and the Samsung Galaxy S8 for all processing tools. Especially the solution from RTKLib of the Huawei Honor 8 is completely different in comparison to the other smartphones.

Generally, it can be said that the absolute difference in comparison with the self-calculated position is in the same range. It is also obvious that RTKLib again has the worst standard deviation in comparison to other tools, respectively the self-calculated position.

A graphical representation of the mean position of the different programs is provided in the next subfigure:

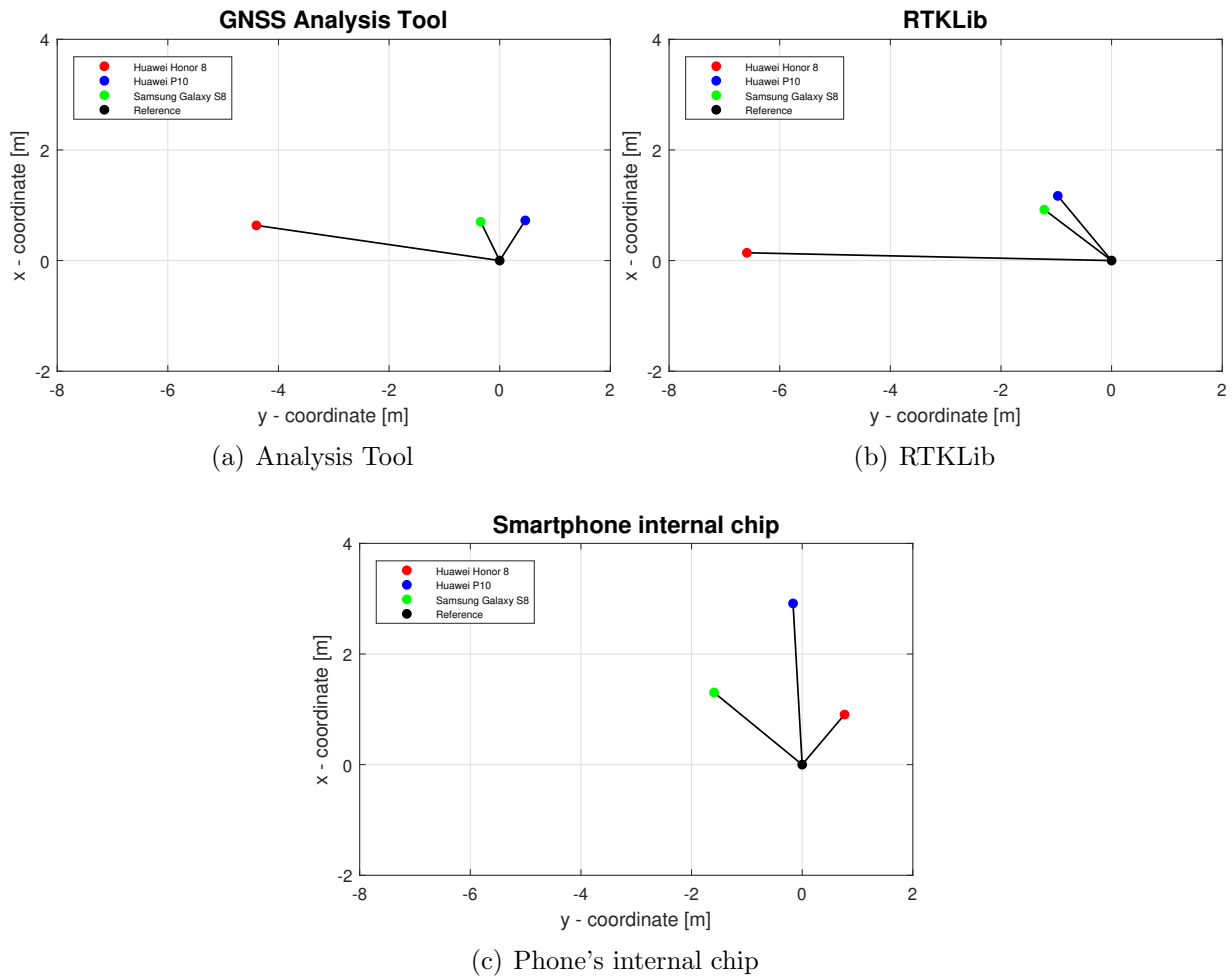


Figure 5.14: Comparison - mean position other tools

In figure 5.14, it can be seen that the Analysis Tool and RTKLib provide very similar positions for all three smartphones. Especially the location of the position is nearly identical. The position of the phone's internal chip differs from the other two tools. Particularly the position of the Huawei Honor 8 differs by several meters.

5.4 Comparison of Multiple Test Measurements

In this section, the results of multiple test measurements in which all three smartphones are compared to each other are presented. Again, only the fourth option from section 5.1.2 is included:

Table 5.7: Comparison of all three smartphones of different measurement campaigns

July 19th, 2018						
Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	3.4	2.8	4.4	10.1	8.0	12.9
Huawei P10	8.8	5.1	10.2	14.2	11.6	18.3
Samsung Galaxy S8	14.2	1.2	14.3	18.4	8.9	20.4
August 20th, 2018						
Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	-2.7	2.8	3.9	7.3	7.5	10.4
Huawei P10	-4.1	-1.5	4.3	9.4	10.9	14.4
Samsung Galaxy S8	-1.7	0.8	1.9	5.3	5.0	7.2
August 24th, 2018: Measurement 1						
Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	0.5	-0.8	1.0	9.7	8.2	12.7
Huawei P10	0.7	3.8	3.8	5.5	5.3	7.6
Samsung Galaxy S8	0.6	2.0	2.0	5.6	5.0	7.5
August 24th, 2018: Measurement 2						
Mobile device	$\Delta x_{ref}[m]$	$\Delta y_{ref}[m]$	$\Delta d_{ref}[m]$	$\sigma_x[m]$	$\sigma_y[m]$	$\sigma_{Helmert}[m]$
Huawei Honor 8	-1.4	6.2	6.3	11.9	10.7	16.1
Huawei P10	-4.6	3.3	5.6	11.2	13.7	17.7
Samsung Galaxy S8	0.2	5.6	5.6	7.4	8.5	11.3

As can be seen in the table above, the differences between the reference position and the self-calculated position are approximately in the same range in all mobile devices. July 19th, 2018, when all smartphones have extremely different positions, can be considered an exception. At this measurement campaign, the calculated position differs by several meters, also in the case of other processing tools.

In conclusion, when comparing the three smartphones, it cannot be determined which phone provides the best solution.

There are numerous influencing factors on the phones. For instance, the heat of the phone itself can influence the clock error rate of the mobile device [20]. The built-in antenna also has a big influence on the results. Even the strength of the signals and the number of visible satellites differ from phone to phone at one measurement campaign.

5.5 SPP Real Time Solution on Android

In this section, the results of the project's Android application are provided. The measurement campaign took place on the 5th of October, 2018 at the rooftop of the University of Technology of Graz. The computations of the following smartphone positions were done in real time.

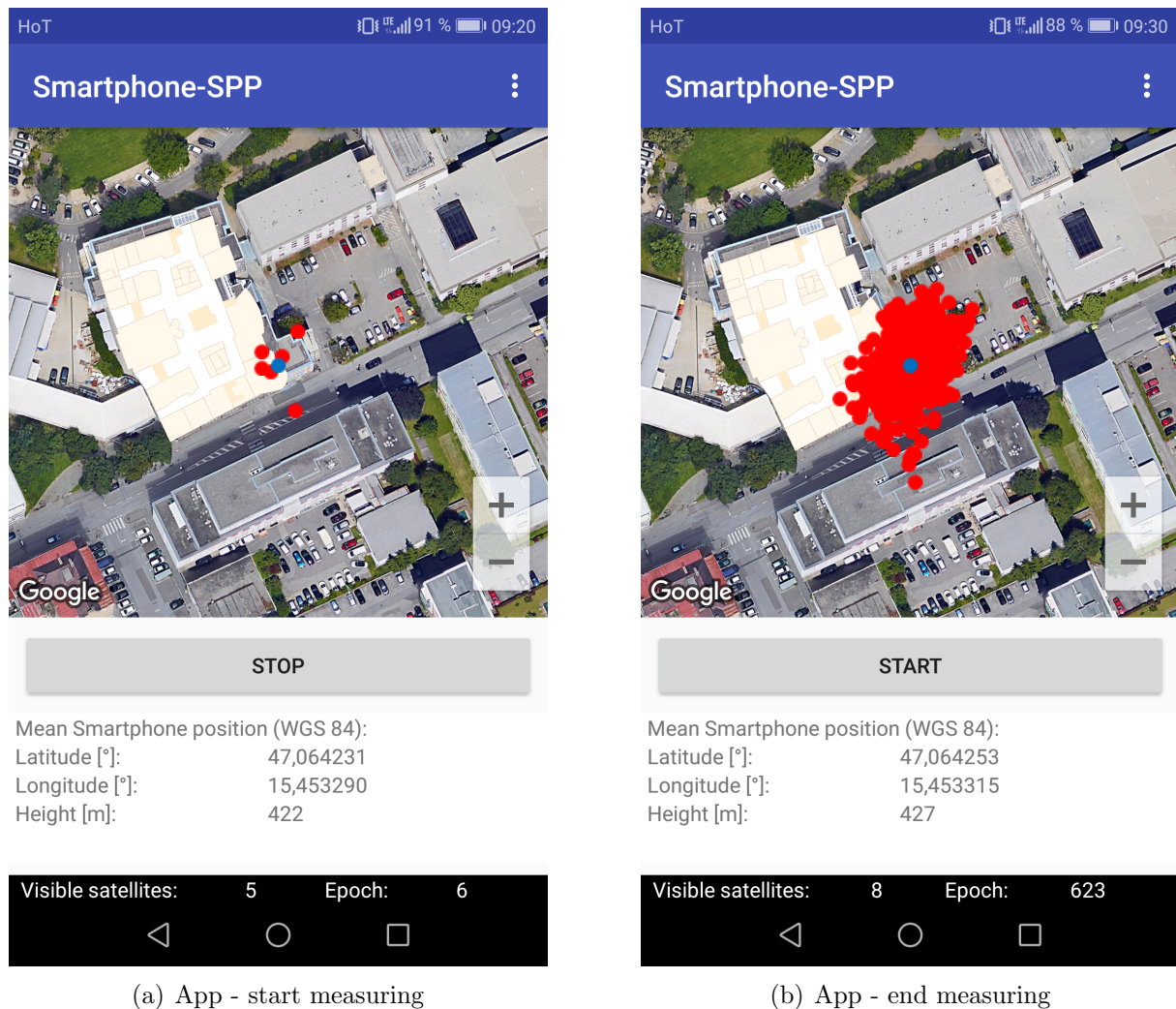


Figure 5.15: App - measuring

In this test measurement, the same reference point was chosen as in the other measurement campaigns. The atmospheric correction was activated, and the elevation angle was used as weighting option. The measurement time was approximately ten minutes.

In the following figure, the position solution, which was computed in real time, and the reference point are plotted. The positions in the following picture are the ellipsoidal WGS84 coordinates and the background is again from Google Maps:

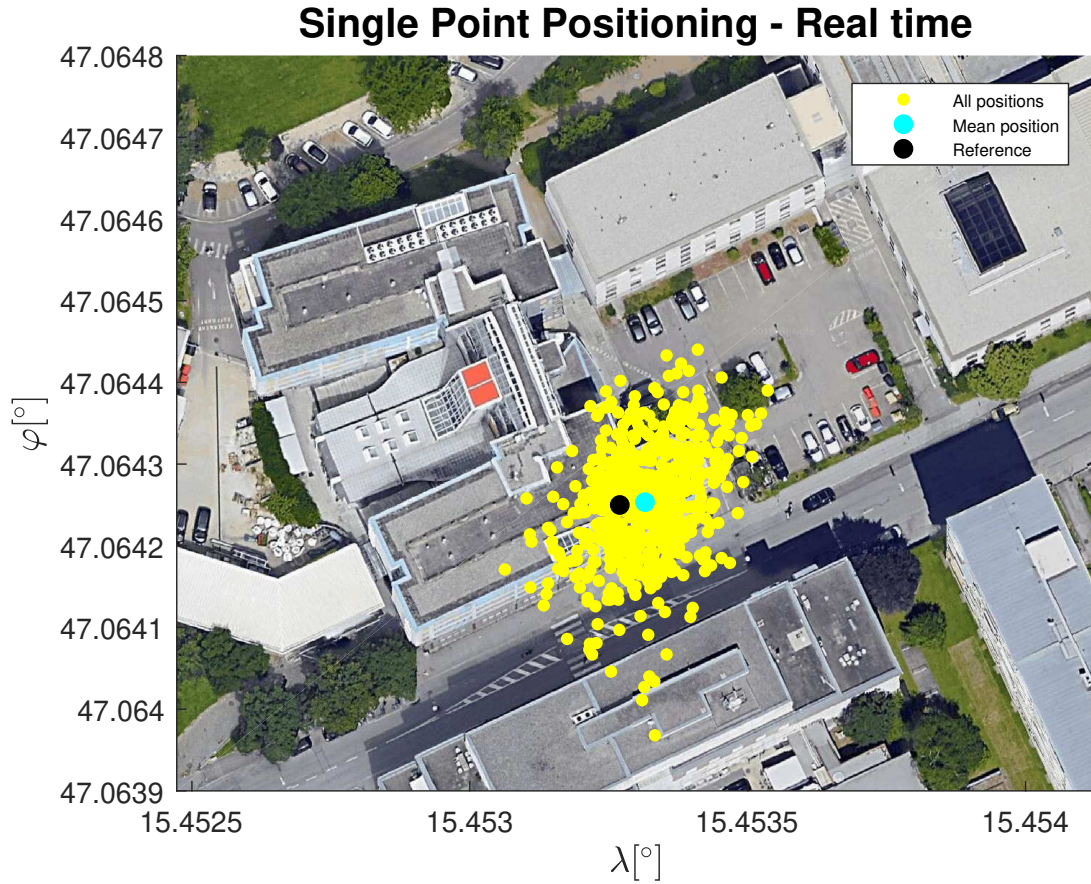


Figure 5.16: Real time - positions

In figure 5.16, the positions from all epochs are plotted, as well as the reference position and the mean position. An absolute difference of $\Delta d_{ref} = 3.5 \text{ m}$ between the Gauß-Krüger coordinates of the reference and the mean point has occurred. As the previous results show that this difference is in the same range. Further numerical results are provided in table 5.8.

Table 5.8: Measurement campaign real time - numerical results

Mobile device	$\Delta x_{ref} [m]$	$\Delta y_{ref} [m]$	$\Delta d_{ref} [m]$	$\sigma_x [m]$	$\sigma_y [m]$	$\sigma_{Helmert} [m]$
Huawei Honor 8	0.3	3.5	3.5	7.8	5.7	9.6

Moreover, the results for the standard deviation are in the same range as in the previous results.

The ellipsoidal height is presented in the following figure. The portrayed values are relative to the reference height.

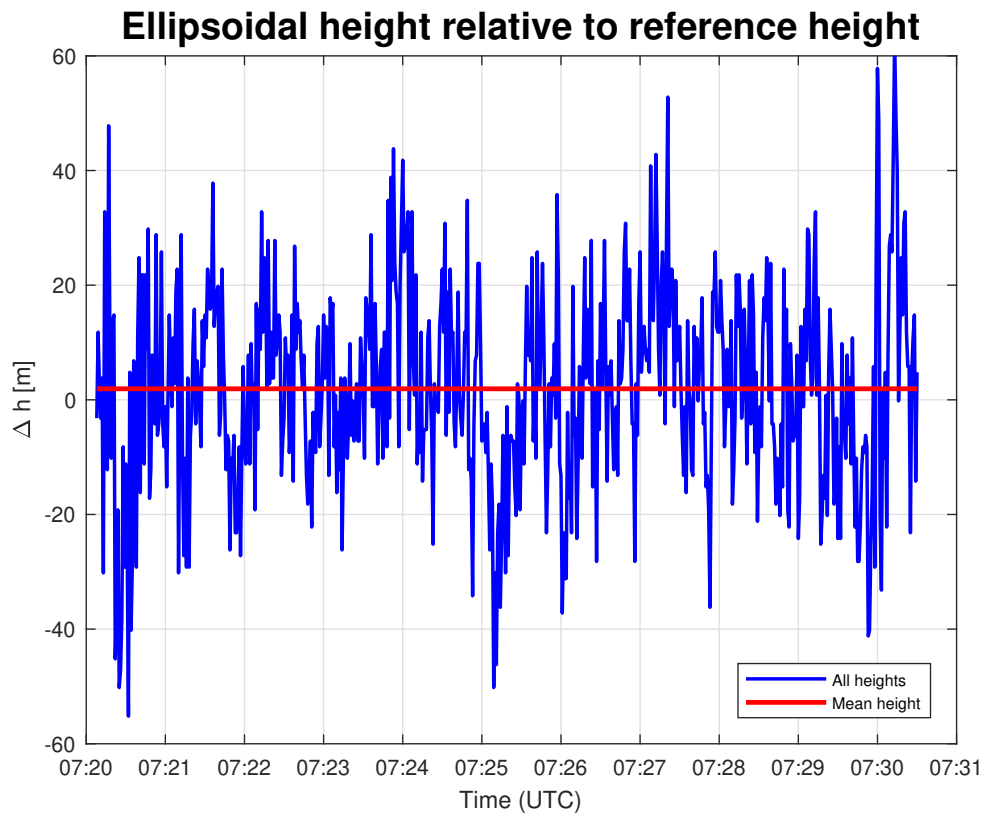


Figure 5.17: Real time - height

As can be seen in figure 5.17, the computed ellipsoidal height has a range of approximately ± 40 m. This underlines the inaccurate determination of the height with satellite-based positioning. In spite of that, the mean height is about two to three meters away from the reference height. Consequently, the mean height matches the reference height very well.

The formal errors of the measured points are plotted in the next subfigure. At this point, it should be mentioned that the first two components are presented in arcseconds, and the third, respectively the up component, is provided in meters. These values were computed as described in section 2.5.11.

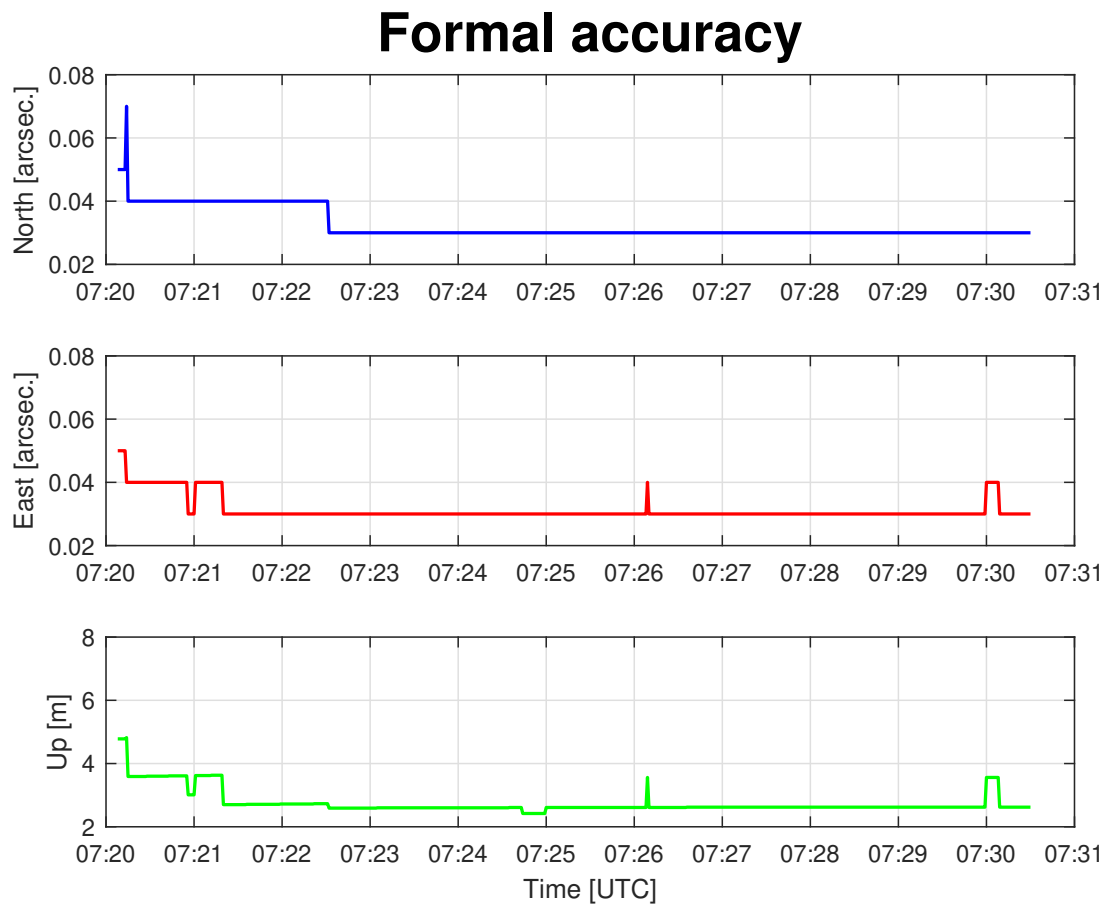


Figure 5.18: Time series of the formal accuracy

As can be seen, the formal accuracy of the north and east directions are in the same range. A fact that can be observed in all three components is that after about two minutes of measuring, the errors remain nearly constant. After that time they are about 0.03 arcseconds for the first two coordinates. This corresponds to about 0.9 meters. In the height component, the formal error is approximately 3 meters.

A reason why the formal errors at the beginning are higher is that there are fewer satellites which are used for the calculations. A further fact which can be seen in the time series are the two peaks in the east and up component. A look at the satellite number shows that for this short time period there is one less satellite.

Chapter 6

Conclusion and Future Tasks

To summarise the results of this master thesis, it can be said that the Android raw measurements are the right step to a better performance of the smartphone positioning. In this project, a Single Point Positioning was investigated and implemented in an Android application. The results show that with code observations, only a meter-level accuracy can be achieved.

The investigations of all three smartphones showed that there can be big differences in the received signals. This can be seen in the number of visible satellites, as well as in the Carrier-to-noise density of the received satellite signals. The positions of the different mobile devices match very well most of the time, but it can also occur that they provide very different positions at some measuring periods.

In comparison with the other processing tools, the self-calculated position is on the same accuracy level. However, several test measurements showed that the positioning performance can differ. This behavior can also be observed in the other software tools as well. In contrast to this, the smartphone's internal position solution yields very good results in nearly all test measurements. A reason for that could be that the smartphone's internal chip uses more sensors, like accelerometer or geomagnetic field sensor, to determine the position [3].

In relation to the created Android application, it needs to be mentioned that the same performance as in post processing is achieved. It should also be said that an internet connection is necessary. That should not be a problem, however, because nowadays, an internet connection is possible for smartphones nearly everywhere and at any time.

For future tasks, it can be said that this topic opens a lot of possibilities in smartphone positioning. For instance, relative positioning between two or more mobile devices can be realised through the access to the raw measurements. Moreover, the fact that two-frequency measurements are possible [19] makes us confident.

A further improvement of the smartphone position can be achieved with a multi-GNSS solution. With such a high number of satellites, positioning in difficult environments would also be possible. Finally, a better position solution would be possible if in addition to the code measurements, Doppler and carrier-phase measurements were used for the computations as well. At this point, it should be mentioned that the carrier-phase measurement cannot be used optimally yet because of the fact that not all Android versions provide a continuous measuring in order to save battery. This is called duty cycling. In the Android P version it is possible to turn off this [21].

Furthermore, in addition to the GNSS sensor, other sensors, for instance inertial sensors, will bring an improvement of the position accuracy.

Appendix A

Appendix

A.1 Constants

The following constants, which are used for the investigations, are listed in the table below:

Table A.1: Constants

Parameter	Numerical value	Unit	Description
a	6378137	m	Big semi-major axis (WGS84 ellipsoid)
b	6356752.3141	m	Small semi-major axis (WGS84 ellipsoid)
μ	$3\,986\,004.418 \cdot 10^8$	$m^3 s^{-2}$	Geocentric gravitational constant
ω_E	$7.292115 \cdot 10^{-5}$	$rad\ s^{-1}$	Earth rotation rate
c	299792458	$m\ s^{-1}$	Speed of light

The parameters which are necessary for the Gauß-Krüger projection are provided in table A.2:

Table A.2: Gauß-Krüger projection: parameters [9, p. 287]

Parameter	Numerical value
α	$6\,367\,449.1458\ m$
β	$-2.51882793 \cdot 10^{-3}$
γ	$2.64354 \cdot 10^{-6}$
δ	$-3.45 \cdot 10^{-9}$
ϵ	$5 \cdot 10^{-12}$

A.2 Reference Coordinates

The reference coordinates which are used for the investigations in this master thesis are listed in the following table:

Table A.3: Reference coordinates - WGS84

Parameter	Numerical value	Unit
X	4195390.4810	m
Y	1159800.6800	m
Z	4646944.4996	m

Bibliography

- [1] S. Banville and F. Van Diggelen. Precision GNSS for Everyone - Precise Positioning Using Raw GPS Measurements from Android Smartphones. *GPS World*, November 2016; p.43-48.
- [2] BROADCOM. BCM4774, 2018. [<https://www.broadcom.com/products/wireless/gnss-gps-socs/bcm4774/#overview>; visited on Aug 23rd, 2018].
- [3] Android Developer. Raw GNSS Measurements, updated Aug 7th, 2018. [<https://developer.android.com/guide/topics/sensors/gnss>; visited on Aug 23rd, 2018].
- [4] M. Fortunato. Initial Tests of the XIAOMI MI 8 dual Frequency GNSS Smartphone, Jul 6th, 2018. [<https://www.nsl.eu.com/nsl-jcms/about-nsl/nsl-blog/15-products-and-services/55-xiaomi-mi8>; visited on Aug 25th, 2018].
- [5] GALILEO GNSS. Huawei's New P10 and P10 Plus smartphones support Galileo, Apr 11th, 2017. [<https://galileognss.eu/huaweis-new-p10-and-p10-plus-smartphones-support-galileo/>; visited on Aug 23rd, 2018], .
- [6] GALILEO GNSS. How to access raw GNSS measurements on smartphones, Dec 12th, 2016. [<http://galileognss.eu/how-to-access-raw-gnss-measurements-on-smartphones/>; visited on Sep 1st, 2018], .
- [7] GSA GNSS Raw Measurements Task Force. Using GNSS raw measurements on Android devices. *European Global Navigation Satellite Systems Agency*, 2017.
- [8] B. Hofmann-Wellenhof, K. Legat, and M. Wieser. *Navigation: Principles of Positioning and Guidance*. Springer Wien NewYork, 2003.
- [9] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle. *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo and more*. Springer Wien NewYork, 2007.
- [10] MyFixGuide. Huawei Honor 8 Teardown, Aug 15th, 2016. [<http://www.myfixguide.com/manual/huawei-honor-8-teardown/>; visited on Aug 23rd, 2018].

BIBLIOGRAPHY

- [11] W. Niemeier. *Ausgleichsrechnung - Statistische Auswertemethoden, 2. Auflage*. Walter de Gruyter Berlin New York, 2008.
- [12] Otter Otter, J., N. Höggerl, E. Imrek, G. Stangl, and E. Zahn. 3-D Referenzsysteme in Österreich. *BEV - Bundesamt für Eich- und Vermessungswesen*, 2015.
- [13] M. Petovello. GNSS Solutions: Coarse Time Positioning. *InsideGNSS*, March/April 2012; p.28-31.
- [14] S. Riley, H. Landau, V. Gomez, N. Mishukova, W. Lentz, and A. Clare. Positioning with Android: GNSS observables, Jan 17th, 2018. [<http://gpsworld.com/positioning-with-android-gnss-observables/>; visited on Aug 27th, 2018].
- [15] ROKUBUN. GNSS carrier phase from Nexus 9, Jul 24th, 2017. [<https://rokubun.cat/2017/07/24/gnss-carrier-phase-nexus-9/>; visited on Aug 30th, 2018].
- [16] RUKUBUN. First look at Android N GNSS raw measurements, Jun 30th, 2016. [<https://rokubun.cat/2016/06/30/android-n-preview-gnss-measurements/>; visited on Sep 1st, 2018].
- [17] J. Sanz Subirana, J.M. Juan Zornoza, and M. Hernández-Pajares. *GNSS DATA PROCESSING, Volume I: Fundamentals and Algorithms*. ESA Communications, 2013.
- [18] J. Slegers. MathWorks - kml2struct (version 1.1.0.0), updated Jun 8th, 2012. [<https://de.mathworks.com/matlabcentral/fileexchange/35642-kml2struct>; visited on Oct 8th, 2018].
- [19] GPS World Staff. Dual-frequency GNSS smartphone hits the market, Jun 11th 2018. [<http://gpsworld.com/dual-frequency-gnss-smartphone-hits-the-market/>; visited on Aug 25th, 2018].
- [20] F. Van Diggelen and M. Khider. GNSS Analysis Tools from Google, Apr 4th, 2018. [<http://insidegnss.com/gnss-analysis-tools-from-google/>; visited on Aug 25th, 2018].
- [21] F. Van Diggelen, R. Want, and W. Wang. How to achieve 1-meter accuracy in Android, Jul 3rd, 2018. [<http://gpsworld.com/how-to-achieve-1-meter-accuracy-in-android/>; visited on Sep 1st, 2018].
- [22] B. Zohar. MathWorks - plot_google_map (version 2.0.0.0), updated Apr 9th, 2018. [https://de.mathworks.com/matlabcentral/fileexchange/27627-zoharby-plot_google_map; visited on Oct 5th, 2018].