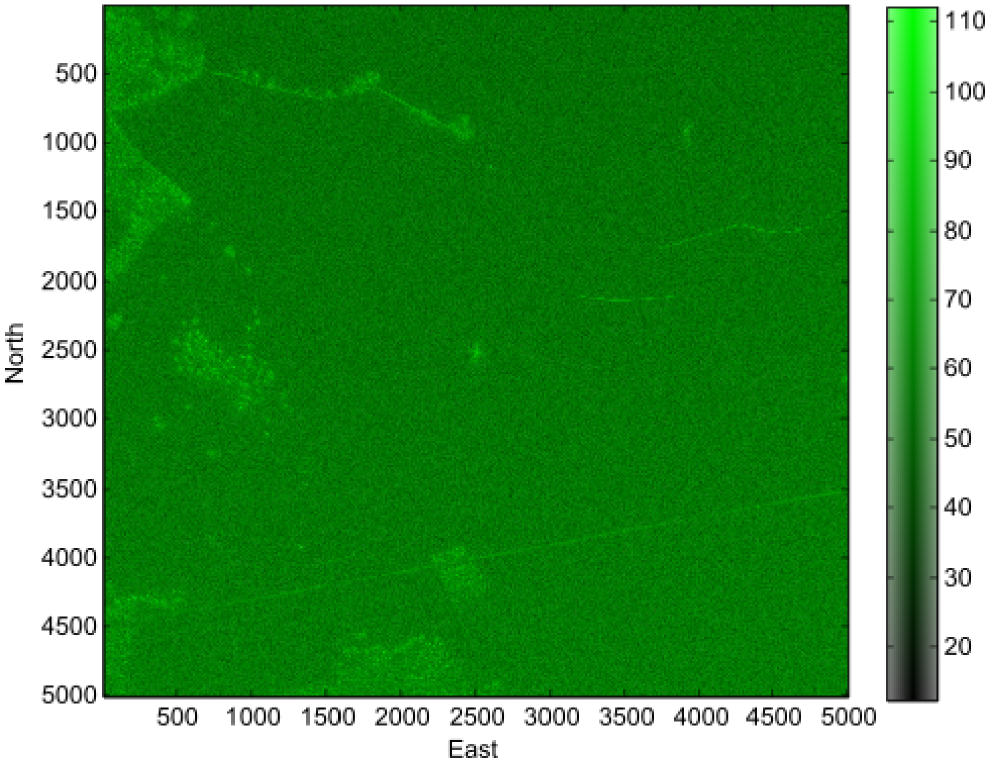
CHALMERS



Evaluate and develop high-performance GPS navigation using free GPS software

Master's Thesis in Engineering Physics

RONNY VIDEKULL

Department of Earth and Space Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

THESIS FOR THE DEGREE IN MASTER OF SCIENCE

Evaluate and develop high-performance GPS navigation using free GPS software

Ronny Videkull

Department of Earth and Space Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Evaluate and develop high-performance GPS navigation using free GPS software

Ronny Videkull

Master of Science Thesis in cooperation with Saab AB
Department of Earth and Space Sciences
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
SAR image created using positions produced by RTKLIB

Chalmers Reproservice
Gothenburg, Sweden 2015

Abstract

Synthetic Aperture Radar (SAR) mounted on an aircraft allows us to create high resolution images of landscapes. In order to obtain such images it is crucial to accurately determine the flight-path of the instrument's platform. For centimeter-level accurate positioning of the aircraft, a differential positioning technique based on Global Navigation Satellite Systems (GNSS) is used. This method, also known as Real-Time Kinematic (RTK) positioning, utilizes carrier phase signals and relies on a second (fixed) "base station" with well known coordinates. In this thesis the accuracy of the open-source GNSS library RTKLIB was evaluated. A comparison with results from a commercial software has shown that RTKLIB is able to provide similar positioning performance. In general it could be concluded that the accuracy of RTKLIB is sufficient for the given SAR application. Moreover, a batch-driven processing chain utilizing RTKLIB was realized in order to minimize manual interactions. The thesis was conducted at Saab AB.

Sammanfattning

Syntetisk aperturradar (SAR) monterat på ett flygplan kan användas för att skapa högupplösta bilder av landskap. För att skapa sådana bilder är det nödvändigt att noggrant bestämma instrumentplattformens flygbana. Centimeternivå på noggrannheten uppnås genom att använda en differentiell positioneringsteknik baserat på Global Navigation Satellite Systems (GNSS). Denna metod, känd som Real-Time Kinematic (RTK) positionering, använder bärvågsmätning och behöver en annan (fixerad) basstation vars position är väl känd position. I denna avhandling utvärderas noggrannheten hos den öppna GNSS programvaran RTKLIB. Genom att jämföra med resultat från en kommersiell programvara har det visats att RTKLIB klarar av att ge liknande positioneringsprestanda. I allmänhet kan slutsatsen dras att noggrannheten hos RTKLIB är tillräcklig för den angivna SAR tillämpningen. Dessutom har en processeringskedja som använder RTKLIB realiserats i syfte att minimera manuella interaktioner. Avhandlingen genomfördes vid Saab AB.

Acknowledgments

First I would like to thank Saab AB for giving me the opportunity to carry out this thesis. Thanks to all people at Saab I have come in contact with for answering all my questions, especially my supervisors Anders Åhlander and Patrik Dammert. I would also like to thank my examiner Thomas Hobiger for his support and ability to always point me in the right direction when I needed it. A special thanks goes to Tomoji Takasu for answering all my questions regarding RTKLIB and fixing the baseline bug so fast.

Ronny Videkull, Gothenburg 2015-05-21

List of symbols and abbreviations

C/A	Coarse Acquisition
CDMA	Code Division Multiple Access
DGPS	Differential GPS
DOP	Dilution Of Precision
ECEF	Earth-Centered Earth-Fixed
EKF	Extended Kalman Filter
ENU	East, North, Up
GDOP	Geometric Dilution Of Precision
GLONASS	Globalnaja nawigazionnaja sputnikowaja sistema
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPST	GPS Time
GRS	Geodetic Reference System
HDOP	Horizontal Dilution Of Precision
IMU	Inertial Measurement Unit
LAMBDA	Least-squares AMBiguity Decorrelation Adjustment
LLI	Loss of Lock Indicator
MEMS	Micro-Electro-Mechanical Systems
NTRIP	Networked Transport of RTCM via Internet Protocol
PRN	Pseudo Random Noise
RINEX	Receiver INdependent EXchange Format
RMS	Root Mean Square

RTCM	Radio Technical Commission for Maritime Services
RTK	Real-Time Kinematic
SAR	Synthetic-Aperture Radar
TAI	International Atomic Time
TDOP	Time Dilution Of Precision
UERE	User Equivalent Range Error
UT1	Universal Time
UTC	Coordinated Universal Time
VDOP	Vertical Dilution Of Precision
VHF	Very High Frequency
WGS	World Geodetic System

Contents

Chapter 1

Introduction

This thesis was conducted at the Airborne Surveillance Systems business unit in the business area Electronic Defence Systems at Saab AB. Saab is one of the world's premiere suppliers of solutions for surveillance, threat detection and location, platform and force protection, as well as avionics. Centre for this knowledge is the business area Electronic Defence Systems.

1.1 Background

One of the projects the division works on is a product called CARABAS[40]. In CARABAS a Synthetic-aperture radar (SAR), a form of radar, is used to create images of landscapes. SAR can be mounted on a moving platform and the distance the SAR device travels over a target creates a large synthetic antenna aperture. The radar under consideration operates in the VHF band, using wavelengths of 1-15 m, as opposed to traditional microwave SAR. Thanks to the long wavelengths it is possible for the radar to penetrate leaves and branches of a forest and see what lies underneath. In doing so, larger objects placed in a forest can be seen, for example cars and trucks, which would be hidden by the forest if microwaves would be used.

In order to obtain high resolution images it is very important to accurately determine the flight-path of the instrument's platform. To achieve the needed accuracy a differential positioning technique known as Real-Time Kinematic (RTK) positioning is used. RTK positioning is a technique based on Global Navigation Satellite Systems (GNSS). It utilizes the carrier phase signals and relies on a second (fixed) "base station" with well known coordinates [22]. This can increase the positioning accuracy of ordinary GPS positioning from meters to centimeters [7].

RTK can be used both in real-time applications and in post-processing. In real-time applications a data link must be set up between the base station and the rover. For post-processing the link is not needed, only the collected data. The latter is being studied in this thesis.

1.2 Purpose

Currently a commercial software is used to perform the differential GPS processing. It has good accuracy but it requires a lot of manual interactions, which is undesirable. Since it is a commercial software there is a cost attached to it, meaning that if Saab sells a product utilizing that software the customer would have to buy the software as well. The purpose of this thesis is to find another processing solution which reduces the manual interactions while fulfilling the accuracy requirements.

1.3 Objective

The goal of this thesis is

- Evaluate the accuracy of the open source program package for GNSS positioning called RTKLIB.
- Produce a batch-driven, stand-alone processing chain of RTKLIB which has a minimum of manual interactions.

The stated requirements for producing high resolution SAR images are a positioning accuracy of 5 cm (RMS) and an angular accuracy of 1° (RMS) [27] for the aircraft position and orientation. The requirement for the vertical positioning is not explicitly stated, but it is not as tightly defined as the horizontal.

Chapter 2

Global Positioning System (GPS)

GPS, which stands for Global Positioning System, has its origin in radio navigation systems. The main application of GPS is to be able to find location and time information anywhere on Earth where there is a free line of sight to four or more GPS satellites under all weather conditions. It originated in the United States military in the 1960s and the decision to make it free for civilan purposes was made after the incident with the Korean Air Lines Flight 007 [23, p. 248].

The first satellite was launched in 1978 and the system was fully operational in the mid 1990s [38]. The satellites orbit at an altitude of approximately 20200 km and are arranged into six equally-spaced orbital planes inclined 55° from the equator, see Figure 2.1. This constellation ensures that at least four satellites are visible at least 15° above the horizon anywhere in the world at any given time [11]. Originally the GPS constellation consisted of 24 satellites, but expansions have been made and as of January 2015 there were 30 active GPS satellites in orbit. Each satellite circles the Earth twice a day [9].

GPS consists of three segments, the space segment, control segment, and user segment:

- The space segment consists of the satellites. They broadcast radio signals to users and receive commands from the control segment.
- The control segment monitors the space segment and sends commands and data to the satellites.
- The user segment consists of the receivers that record and interpret the radio signals broadcasted by the satellites.

The satellites are monitored by five base stations which transmit ephemerides and clock adjustments [41, p. 2]. Ephemeris data are used to calculate the position of the satellite. The location of a receiver is determined by measuring the ranges between the receiver and at least four simultaneously observed satellites.

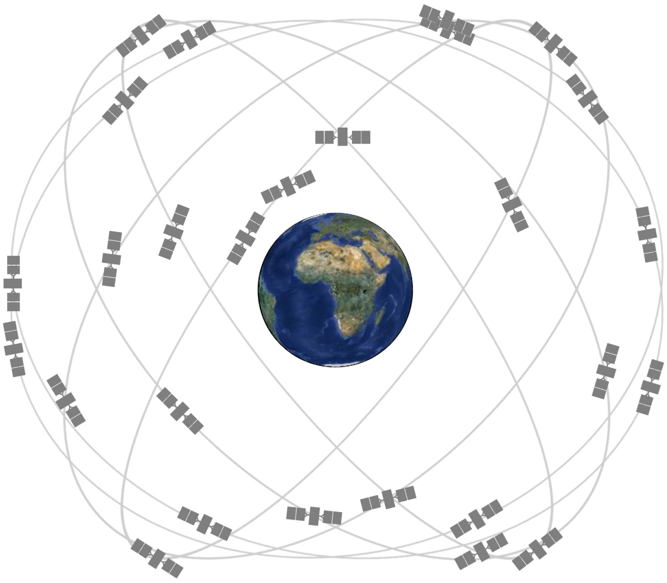


Figure 2.1: GPS constellation [39].

The term Global Navigation Satellite System (GNSS) can be used for a satellite system with global coverage, and is often used when talking about satellite navigation without specifying the system. GPS was the first GNSS. The Russian counterpart, Globalnaja nawigazionnaja sputnikowaja sistema (GLONASS), was the second. Other systems that do not yet have full global coverage and are in various states of completion are the European Galileo and the Chinese BeiDou/COMPASS. This thesis will only consider GPS.

2.1 Operation principle

The GPS system is based on time. A broadcasted GPS satellite signal consists of, among other things, a message with the time of transmission and a pseudorandom noise (PRN) code. The PRN code is a sequence of ones and zeros. When a receiver receives the signal it will compare the received code with an identical coded signal generated internally. The replica code is shifted in time until it achieves correlation with the received code. The time shift corresponds to the propagation time of the signal. This is illustrated in Figure 2.2, where t_e is the signal emission time, t_r is the signal reception time and τ is the propagation time. The receiver also keeps track of the phase of the signal, and measures the frequency shift of the signal caused by the Doppler effect.

By multiplying the propagation time with the speed of light the distance between the satellite and user can be computed. This is not the complete story as the signal does not

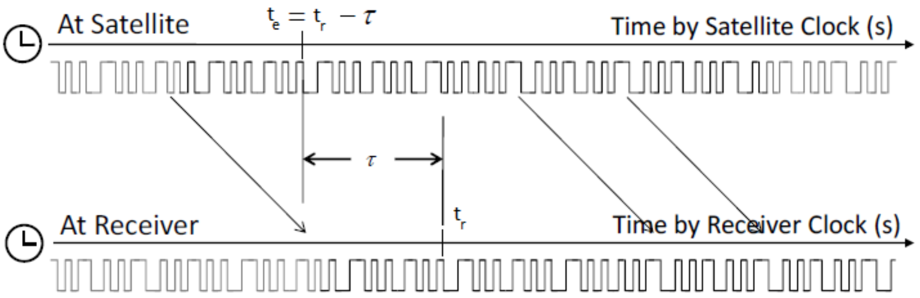


Figure 2.2: Use of replica code to determine signal propagation time [31].

travel through vacuum, and a more complete model is presented in Section 2.3.1.

2.2 Signals

The satellites use up to three frequencies to transmit data, all located in the L-band. The frequencies are known as L1, L2 and L5 and they are 1575.42 MHz, 1227.60 MHz and 1176.45 MHz respectively. Usually the word "carrier" is used when talking about the GPS signals, which comes from that the signals are modulated with some data in order to carry information. The three different carrier frequencies are generated by multiplying the fundamental frequency 10.23 MHz by 154, 120 and 115. The fundamental L-band frequency is produced by atomic clocks aboard the satellite. The carrier frequency and clock rates on the satellites are offset to compensate for relativistic effects [8]. For an observer on a satellite the fundamental frequency would appear to be 10.2299999954326 MHz. Without the offset of the clock rate the atomic clocks aboard the satellites would tick faster than identical clocks on the ground by 38 microseconds per day. This would lead to accumulating positioning errors of about 10 kilometers per day [2].

Superimposed on the carrier frequencies are pseudo random noise (PRN) codes and navigation data. The latter in the form of satellite ephemerides, ionospheric modeling coefficients, status information, system time and satellite clock corrections.

All satellites transmit three PRN ranging codes [8] known as the C/A-code, P-code and Y-code:

Coarse/Acquisition (C/A) code is a relatively short code and it is modulated onto the L1 carrier. The length of the code is 1 millisecond at a chipping rate of 1.023 million chips per second. The broadcast of the C/A-code is known as the Standard Positioning Service (SPS) and it is available to civilian users. Because the C/A-code is relatively short it is used for acquisition of the P-code.

Precision (P) code is a lot longer than the C/A-code. It is a 7-day sequence and it is modulated on the L1, L2 and L5 carriers, allowing for the removal of the first order effects of the ionosphere. The chipping rate of this code is 10.23 million chips per second, i.e. ten times higher than the C/A code. The broadcast of the P-code is known as the Precise Positioning Service (PPS).

Y-code is the name that is used when the P-code is encrypted, also known as the P(Y)-code. The Y-code is the result of combining the P-code and a W encryption code known only to authorized users. One of the main reason for encrypting the P-code is to protect against spoofing, the act of deceiving a GPS receiver by broadcasting counterfeit GPS signals. Therefore the encryption is known as anti-spoofing.

Newer satellites might also transmit additional PRN ranging codes, cf. [8]. The codes are unique for each satellite and they are predictable so that receivers can replicate them. This makes it possible for a receiver to distinguish individual signals received on the same frequency and the receiver can therefore identify from which satellites the signals originate by using something called Code Division Multiple Access (CDMA), cf. [18].

2.3 GPS observables

There are three basic GPS observables known as code pseudoranges, carrier phases and Doppler measurements [41].

2.3.1 Code pseudorange

The pseudorange is the distance between the satellite at the time of signal transmission and the receiver antenna at the time of signal reception. In the simplest of cases the signal travels only through vacuum and there are no clock errors. Then the pseudorange P_r^s between receiver r and satellite s would be easily determined by

$$P_r^s(t_r, t_e) = (t_r - t_e)c, \tag{2.1}$$

where c is the speed of light, t_r is the signal reception time of the receiver and t_e is the signal emission time of the satellite. This is the geometric distance between the satellite at t_e and the receiver at t_r. The value of t_e is known from the navigation message, and the value of t_r is found by shifting the internal replica of the code in time until it achieves correlation with the received code, see Figure 2.2. The width of the correlation function is inversely proportional to the chipping rate of the signal. For the C/A code the correlation produces a peak $1\,\mu$s wide. Multiplying this with the speed of light one finds that this correspond to about 300 m. For the P-code, with ten times higher chipping rate, the correlation peak would be ten times shorter and therefore the corresponding distance would be about 30 meters. The peak of the correlation function can be determined to about 1% of the width [41, p. 39], this gives ranging accuracy of 3 m and 0.3 m for the C/A and P-code, respectively.

Denoting the satellite vector $\mathbf{r^s}(t) = (x^s(t), y^s(t), z^s(t))$ and the receiver vector $\mathbf{r_r}(t) = (x_r(t), y_r(t), z_r(t))$, the pseudorange can be expressed as the distance between their vectors,

$$\begin{aligned} P_r^s(t_r, t_e) &= ||\mathbf{r^s}(t_e) - \mathbf{r_r}(t_r)|| \\ &= \sqrt{(x^s(t_e) - x_r(t_r))^2 + (y^s(t_e) - y_r(t_r))^2 + (z^s(t_e) - z_r(t_r))^2}, \end{aligned} \tag{2.2}$$

This geometric distance will be denoted $\rho_r^s(t_r, t_e)$.

However we will never experience the simple case as the signals will travel through a medium which is not vacuum and both the satellite and receiver clock will have errors. Introducing clock errors the pseudorange can be represented as

$$\begin{aligned} P_r^s(t_r, t_e) &= ((t_r + \delta t_r) - (t_e + \delta t^s))c \\ &= (t_r - t_e)c + (\delta t_r - \delta t^s)c \\ &= \rho_r^s(t_r, t_e) + (\delta t_r - \delta t^s)c, \end{aligned} \tag{2.3}$$

where δt_r and δt^s denotes the clock offsets of the receiver and satellite, respectively.

Adding ionospheric and tropospheric delay and other errors, denoted by I, T and ε_p, the pseudorange model can be written as

$$P_r^s(t_r, t_e) = \rho_r^s(t_r, t_e) + (\delta t_r - \delta t^s)c + I_r^s + T_r^s + \varepsilon_p. \tag{2.4}$$

Among the errors hidden in ε_p are earth tide effects, loading tide effects, multipath and relativistic effects [41]. This is the pseudorange model used in RTKLIB [32].

To calculate the position of one receiver at least four visible GPS satellites are needed. This is because the receiver clock bias needs to be estimated along with the three dimensional position. The rest of the parameters are modeled. The tropospheric delay T_r^s will be modeled using the Saastamoinen model [26]. The ionospheric delay I_r^s will be modeled using Klobuchar model [15] which uses the broadcasted ionospheric parameters.

The need of at least four satellites can be illustrated using the pseudo range equation. For simplicity the tropospheric and ionospheric delay terms are ignored, and similarly the additional errors are ignored. Expanding ρ_r^s the pseudo range can be written

$$P_r^s = \sqrt{(x^s - x_r)^2 + (y^s - y_r)^2 + (z^s - z_r)^2} + (\delta t_r - \delta t^s)c. \tag{2.5}$$

The satellite position (x^s, y^s, z^s) and satellite clock error (δt^s) are stated in the navigation message and therefore known. The other four terms, three position components and one clock error, are unknown. The system of equations for a receiver denoted r and the satellites

1, 2, 3 and 4 can be written

$$\begin{aligned}
P_r^1 &= \sqrt{(x^1 - x_r)^2 + (y^1 - y_r)^2 + (z^1 - z_r)^2} + (\delta t_r - \delta t^1)c \\
P_r^2 &= \sqrt{(x^2 - x_r)^2 + (y^2 - y_r)^2 + (z^2 - z_r)^2} + (\delta t_r - \delta t^2)c \\
P_r^3 &= \sqrt{(x^3 - x_r)^2 + (y^3 - y_r)^2 + (z^3 - z_r)^2} + (\delta t_r - \delta t^3)c \\
P_r^4 &= \sqrt{(x^4 - x_r)^2 + (y^4 - y_r)^2 + (z^4 - z_r)^2} + (\delta t_r - \delta t^4)c.
\end{aligned} \tag{2.6}$$

This system of nonlinear equations can be solved iteratively.

2.3.2 Carrier phase

It is not possible to measure the exact number of carrier cycles between a satellite and a receiver. It is however possible to keep track of the change in cycles since the start of a measurement. When a receiver initially locks on to a satellite the number of cycles between the satellite and the receiver is not known, but as long as the receiver maintains a lock on the signal that number stays constant. Then the fractional phase is measured continuously and the phase difference for each measurement is added to a tally keeping track of the change in cycles. This kind of measurement will have the correct fractional phase, but will have an ambiguous amount of full cycles wrong. To find the correct number of full cycles the ambiguity parameters can be modeled. The phase measurement can be made on all the carrier frequencies as it completely disregards the information contained within the signal, only the phase of the signal is needed.

The phase can be measured with a precision better than 1% of the wavelength [41, p. 39], meaning the positioning accuracy significantly improves from using only the code. For the L1 carrier, whose wavelength is about 19 cm, this corresponds to an accuracy of about 1.9 mm and 2.4 mm for L2 whose wavelength is about 24 cm. To fully exploit the phase measurements one must correct for propagation effects of the signal which is why the carrier phase is not usually used in standard navigation systems.

Once again starting in the simple and unrealistic case of the signal travelling through vacuum and no clock errors we can write the measured phase ϕ_r^s between receiver r and satellite s as

$$\phi_r^s(t_r, t_e) = \phi_r(t_r) - \phi^s(t_e) + N_r^s. \tag{2.7}$$

Here t_r is the signal reception time of the receiver, t_e is the signal emission time of the satellite, ϕ_r is the phase of the receiver's oscillator and ϕ^s is the phase of the emitted signal from the satellite. N_r^s denotes the number of full cycles between satellite s and receiver r, and that is the parameter which should be determined.

Assuming the received satellite signal and the reference carrier of the receiver have the frequency f, the phases can be written as

$$\phi_r(t_r) = f(t_r - t_0) + \phi_{r,0} \tag{2.8}$$

and

$$\phi^s(t_e) = f(t_e - t_0) + \phi^{s,0}, \tag{2.9}$$

where t_0 is the initial time, $\phi_{r,0}$ and $\phi^{s,0}$ are the initial phases of the receiver oscillator and the initial satellite carrier phase of the emitted signal respectively.

In a more realistic case both the receiver and satellite clock has errors. Using the same notation for those errors as used for the pseudorange the phases can be written

$$\phi_r(t_r) = f(t_r + \delta t_r - t_0) + \phi_{r,0} \tag{2.10}$$

and

$$\phi^s(t_e) = f(t_e + \delta t^s - t_0) + \phi^{s,0}. \tag{2.11}$$

Using this the carrier phase becomes

$$\begin{aligned} \phi_r^s(t_r, t_e) &= (f(t_r + \delta t_r - t_0) + \phi_{r,0}) - (f(t_e + \delta t^s - t_0) + \phi^{s,0}) + N_r^s \\ &= f(t_r - t_e) + f(\delta t_r - \delta t^s) + (\phi_{r,0} - \phi^{s,0}) + N_r^s. \end{aligned} \tag{2.12}$$

To express the carrier phase using the wavelength instead of the frequency the relation

$$c = f\lambda \quad \Rightarrow \quad f = c/\lambda \tag{2.13}$$

is used, resulting in

$$\phi_r^s(t_r, t_e) = \frac{c}{\lambda}(t_r - t_e) + \frac{c}{\lambda}(\delta t_r - \delta t^s) + (\phi_{r,0} - \phi^{s,0}) + N_r^s. \tag{2.14}$$

Defining the phase range, Φ_r^s, as the carrier phase multiplied by the carrier wavelength we get

$$\begin{aligned} \Phi_r^s(t_r, t_e) &= \lambda \phi_r^s(t_r, t_e) \\ &= c(t_r - t_e) + c(\delta t_r - \delta t^s) + \lambda(\phi_{r,0} - \phi^{s,0}) + \lambda N_r^s \\ &= \rho_r^s(t_r, t_e) + c(\delta t_r - \delta t^s) + \lambda(\phi_{r,0} - \phi^{s,0}) + \lambda N_r^s. \end{aligned} \tag{2.15}$$

So far only the clock errors have been introduced, but there are also other error sources. As for the pseudorange there is a delay because of the ionosphere and troposphere. Unique errors for the carrier phase are hidden in a term denoted $d\Phi_r^s$, which includes the receiver and satellite antenna phase offsets, receiver and satellite antenna phase center variations, station displacement, phase wind-up effect and relativity correction on the satellite clock. Once again other errors will be stored in a separate term, this time denoted ε_Φ for the phase range. Putting all of this together Equation (2.15) becomes

$$\Phi_r^s(t_r, t_e) = \rho_r^s(t_r, t_e) + c(\delta t_r - \delta t^s) + \lambda(\phi_{r,0} - \phi^{s,0}) + \lambda N_r^s - I_r^s + T_r^s + d\Phi_r^s + \varepsilon_\Phi, \tag{2.16}$$

which is the model used in RTKLIB [32]. Note that the sign of the error from the ionosphere is opposite of that used in the pseudo range model. This is because the ionosphere delays the code signal transmission and advances the phase signal transmission [21].

2.3.3 Doppler measurements

Because of the relative motion of the satellite and receiver the received signal frequency at the receiver might not be the same as the frequency of the signal when it left the satellite. This phenomena is known as the Doppler effect or the Doppler shift, and the result is that when the distance between the satellite and receiver is decreasing the frequency of the signal received is increasing and vice versa.

The observable known as the Doppler count D is the integration of the frequency shift over the interval of the epoch [14], plus the value from the previous epoch. The measurement for the Doppler count is strongly connected to the carrier phase measurement. A predicted value of D is used to predict the phase change, and then the phase change is compared with the measured value to get the precise value of the Doppler frequency shift. To get the original prediction of D the receiver makes use of almanacs [5], which are approximate orbital data parameters for all satellites.

2.4 Reference systems

In order to formulate the mathematics needed for acquiring the position of a receiver it is necessary to choose a time and coordinate reference system in which the states of both the satellite and receiver can be represented.

2.4.1 Time systems

A large part of GPS systems are based on keeping a very exact track of the time. There exist many similar time systems with small differences. Time systems usually referred to when talking about GPS are Coordinated Universal Time (UTC), Universal Time (UT1), GPS system time (GPST) and International Atomic Time (TAI):

UTC is the time standard that most people come in contact with on a daily basis. All the time zones in the world are expressed using UTC, where the local times are derived as a number of positive or negative offsets from UTC. There are two components used to determine UTC; TAI and UT1. The fact that UTC is based on UT1 is the origin of leap seconds. Because of the leap seconds UTC is not a uniform time scale. It is the International Bureau of Weights and Measures (BIPM) that is responsible for maintaining UTC [36].

UT1 is a time standard that is based on the Earth's rotation angle with respect to the Sun. Because of the variations in Earth's rotation UT1 is a nonuniform time scale. The responsibility for definitely determining UT1 falls to the International Earth Rotation and Reference System Service (IERS). It is the IERS that decides when to add leap seconds to the UTC to make sure that the difference between UTC and UT1 is less than 0.9 s.

GPST is the time standard that is used by the GPS satellites. It is a uniform time scale. It is usually expressed as a GPS week number and the time of week (TOW) in seconds. The gps week number is the number of weeks since the start epoch on January 6, 1980 at 00:00:00 (UTC) [16]. Since GPST is a uniform time scale the only difference with UTC is the number of leap seconds since the start epoch, which at the time of this writing is 16 seconds. It is defined by the GPS Control segment and is based on a set of atomic clocks at the Monitor Stations on board the satellites [30].

TAI is a time standard based on continuous counting of the SI second and therefore a uniform time scale. It is determined using the weighted average from over 400 atomic clocks worldwide [1]. The time difference between TAI and UTC is the total amount of leap seconds plus the initial offset of ten seconds between the two systems since before leap seconds were introduced in 1972. As of this writing the time difference between TAI and UTC is 35 seconds. It is worth noting that TAI is always ahead of GPST with 19 seconds. BIPM is responsible for maintaining TAI as well [36].

2.4.2 Coordinate systems

There are many coordinate systems that can be used to describe positions. When describing a position on the Earth's surface it is convenient to use a coordinate system that rotates with the Earth, known as an Earth-centered Earth-fixed (ECEF) coordinate system, see Figure 2.3. The origin is found at the Earth's mass center, and its *xy*-plane coincides with the Earth's equatorial plane. The x-axis points in the direction of the Greenwich meridian, or 0° longitude. The z-axis points in the same direction as the mean rotational axis of the Earth. The y-axis is directed to complete a right-handed system, which makes it point toward the 90°E longitude. Since the coordinate system rotates with the Earth the coordinates of a fixed point does not change.

The reference system currently used by GPS is known as the World Geodetic System 1984 (WGS84), which is an ellipsoidal coordinate system based on ECEF coordinates. Originally WGS84 used the GRS80 reference ellipsoid, and during the years since it was introduced refinements have been made to the system [19]. The position of a receiver is often given in latitude, longitude and height using WGS84. The Earth's shape in the model is such that the cross-sections of the Earth parallel to the equatorial plane are circular, and cross-sections of the Earth normal to the equatorial plane are ellipsoidal. In Figure 2.4 the *x*,*y* plane of the reference ellipsoid is shown. Note that the geodetic and geocentric latitude are different. The geocentric latitude is determined around the centre, and the geodetic latitude is determined by the angle between the normal of the ellipsoid and the plane of the equator.

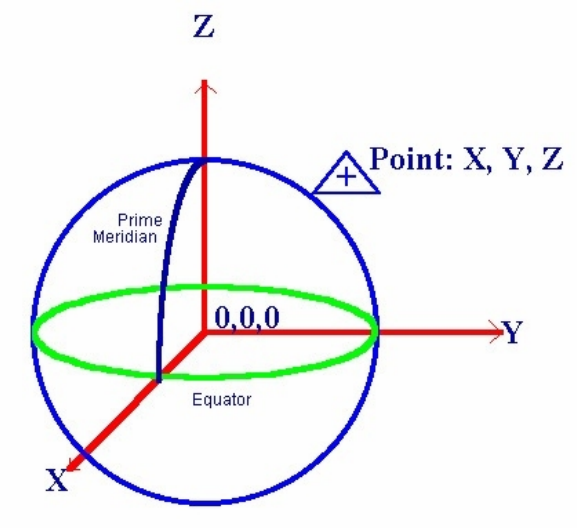


Figure 2.3: ECEF coordinate system [4].

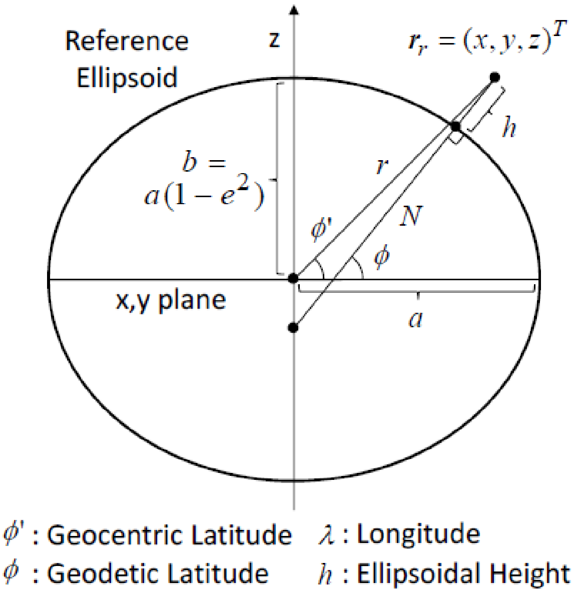


Figure 2.4: Reference ellipsoid, cross-section normal to equatorial plane [32]. a is the semi-major axis, b is the semi-minor axis and e is the eccentricity.

For some applications it is sufficient to use a simpler coordinate system such as local east, north, up (ENU) coordinates. In such a system the coordinates are formed by projecting positions to a plane which is tangential to the surface of the Earth in a reference point. Usually the x-axis points in the east direction, y in the north direction and z straight up. Because of the shape of the earth this type of coordinate system is only valid for short distances from the reference point. In Figure 2.5 a ENU coordinate system is presented together with an ECEF system.

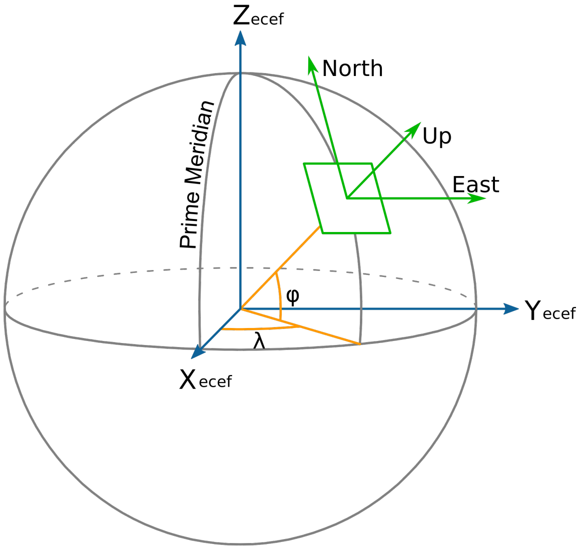


Figure 2.5: ECEF and ENU coordinates [32].

2.5 Differential positioning

In order to improve the positioning accuracy of ordinary GPS a differential positioning technique called Real-Time Kinematic, abbreviated RTK, can be used. RTK is based on the use of the carrier phase. Two GPS receivers are needed, one which is fixed on a known location and the other which might be moving around. The two receivers are called base station and rover, respectively. If the two receivers are close to each other it is possible to eliminate most errors that are common to both receivers. This is done by forming linear combinations of the measurements, i.e. forming differences of measurements from two receivers observing the same satellite. That process is referred to as a single difference. Using a pair of single differences one can form the double difference. Using double difference most of the error sources are removed [24], except for the multipath which can be mitigated but not eliminated [14].

There are no exact numbers for how close the receivers should be to each other for eliminating errors. A good rule of thumb is that they should be no further apart than 20 km [25], but the number varies with atmosphere conditions.

2.5.1 Single difference formation

Single difference can be formed when two receivers observe the same satellite, see Figure 2.6 where receivers r and b observe satellite s. It can be defined as

$$SD^s_{br}(O) = O^s_r - O^s_b, \tag{2.17}$$

where O is the original observable, subscripts b and r correspond to base station and rover receiver, respectively, and superscript s denotes the satellite from which the signal originated.

Using the pseudorange from Equation (2.4) the SD can be formed for the receivers b and r locked on satellite s as

$$\begin{aligned} P^s_{br} &= P^s_r - P^s_b \\ &= (\rho^s_r + (\delta t_r - \delta t^s)c + I^s_r + T^s_r + \varepsilon_{p,r}) - (\rho^s_b + (\delta t_b - \delta t^s)c + I^s_b + T^s_b + \varepsilon_{p,b}) \\ &= (\rho^s_r - \rho^s_b) + (\delta t_r - \delta t_b)c + (\delta t^s - \delta t^s) + (I^s_r - I^s_b) + (T^s_r - T^s_b) + (\varepsilon_{p,r} - \varepsilon_{p,b}) \\ &= \rho^s_{br} + c\delta t_{br} + I^s_{br} + T^s_{br} + \varepsilon_{p,br}. \end{aligned} \tag{2.18}$$

Using the same technique for the carrier phase range from Equation (2.16) we get

$$\begin{aligned} \Phi^s_{br} &= \Phi^s_r - \Phi^s_b \\ &= (\rho^s_r + c(\delta t_r - \delta t^s) + \lambda(\phi_{r,0} - \phi^{s,0}) + \lambda N^s_r - I^s_r + T^s_r + d\Phi^s_r + \varepsilon_{\Phi,r}) - \\ &\quad\ (\rho^s_b + c(\delta t_b - \delta t^s) + \lambda(\phi_{b,0} - \phi^{s,0}) + \lambda N^s_b - I^s_b + T^s_b + d\Phi^s_b + \varepsilon_{\Phi,b}) \\ &= \rho^s_{br} + c\delta t_{br} + \lambda\phi_{br,0} + \lambda N^s_{br} - I^s_{br} + T^s_{br} + d\Phi^s_{br} + \varepsilon_{\Phi,br}. \end{aligned} \tag{2.19}$$

Note that for both the pseudorange and carrier phase range the satellite clock error, δt^s, is removed because it is independent of where the receivers are. For the carrier phase range the satellite initial carrier phase is also removed.

2.5.2 Double difference formation

Double difference can be formed when two receivers observe the same two satellites, see Figure 2.7 where receivers r and b observe satellites s and z. It can be defined as

$$DD^{sz}_{br} = SD^z_{br}(O) - SD^s_{br}(O), \tag{2.20}$$

where superscripts s and z represents different satellites.

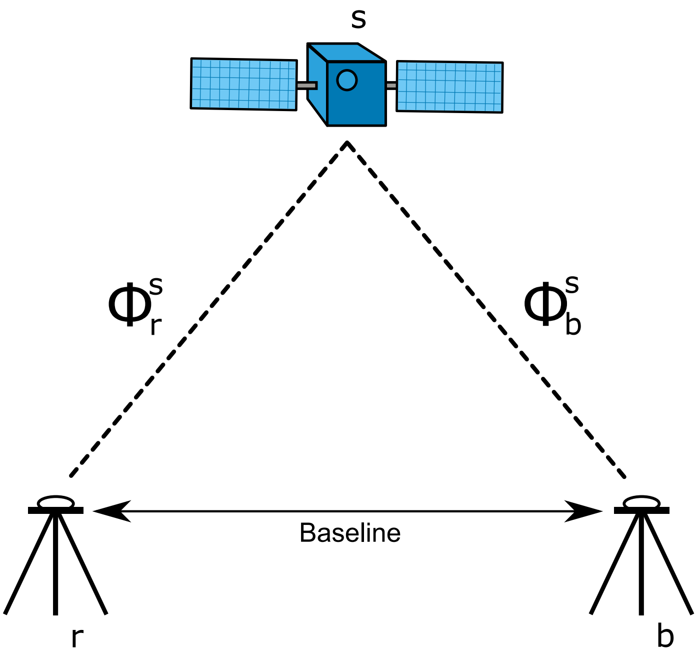


Figure 2.6: Single differencing.

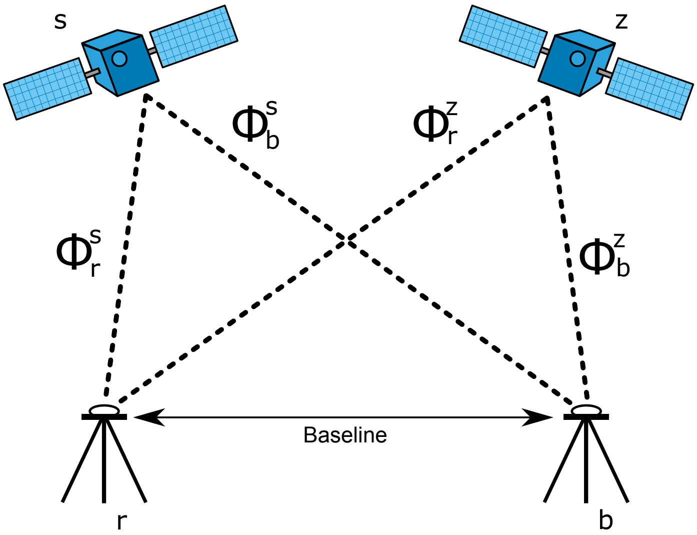


Figure 2.7: Double differencing.

Using the same technique as for the single differencing the double difference of the pseudoranges and carrier phase ranges can be formed as

$$\begin{aligned} P^{sz}_{br} &= P^{z}_{br} - P^{s}_{br} = \rho^{sz}_{br} + c\delta t^{sz}_{br} + I^{sz}_{br} + T^{sz}_{br} + \varepsilon^{sz}_{p_{br}} && (2.21) \\ \Phi^{sz}_{br} &= \Phi^{sz}_{r} - \Phi^{sz}_{b} = \rho^{sz}_{br} + c\delta t^{sz}_{br} + \lambda\phi^{sz}_{br,0} + \lambda N^{sz}_{br} - I^{sz}_{br} + T^{sz}_{br} + d\Phi^{sz}_{br} + \varepsilon_{\Phi_{br}} && (2.22) \end{aligned}$$

Some of the terms can be canceled out. First off the receiver clock offsets are constant and independent of satellites, thus we can write

$$\delta t^{sz}_{br} = \delta t^{z}_{br} - \delta t^{s}_{br} = (\delta t^{z}_{r} - \delta t^{z}_{b}) - (\delta t^{s}_{r} - \delta t^{s}_{b}) = (\delta t^{z}_{r} - \delta t^{s}_{r}) + (\delta t^{s}_{b} - \delta t^{z}_{b}) = 0. \quad (2.23)$$

The same goes for the initial carrier phases of the receivers:

$$\phi^{sz}_{br,0} = \phi^{z}_{br,0} - \phi^{s}_{br,0} = (\phi^{z}_{r,0} - \phi^{z}_{b,0}) - (\phi^{s}_{r,0} - \phi^{s}_{b,0}) = (\phi^{z}_{r,0} - \phi^{s}_{r,0}) + (\phi^{s}_{b,0} - \phi^{z}_{b,0}) = 0. \quad (2.24)$$

For short baselines the base station and rover are close enough to each other that the weather conditions are very similar. In such cases the ionospheric and tropospheric errors are highly correlated and one assumes that they are eliminated in the double differencing process,

$$I^{sz}_{rb} = I^{z}_{rb} - I^{s}_{rb} \approx 0 \quad (2.25)$$

and

$$T^{sz}_{rb} = T^{z}_{rb} - T^{s}_{rb} \approx 0. \quad (2.26)$$

That means that for a short baseline the double difference pseudorange and carrier phase range can be written as

$$\begin{aligned} P^{sz}_{br} &= \rho^{sz}_{br} + \varepsilon^{sz}_{p_{br}}, && (2.27) \\ \Phi^{sz}_{br} &= \rho^{sz}_{br} + \lambda N^{sz}_{br} + d\Phi^{sz}_{br} + \varepsilon^{sz}_{\Phi_{br}}. && (2.28) \end{aligned}$$

This is the double difference measurement model used in RTKLIB for baselines shorter than 10 km [32], and this is the category all flights in this thesis will fall under. For longer baselines the ionospheric and tropospheric errors would not cancel out and the double difference pseudorange and carrier phase range would then be

$$\begin{aligned} P^{sz}_{br} &= \rho^{sz}_{br} + I^{sz}_{br} + T^{sz}_{br} + \varepsilon^{sz}_{p_{br}}, && (2.29) \\ \Phi^{sz}_{br} &= \rho^{sz}_{br} + \lambda N^{sz}_{br} + d\Phi^{sz}_{br} - I^{sz}_{br} + T^{sz}_{br} + \varepsilon^{sz}_{\Phi_{br}}. && (2.30) \end{aligned}$$

2.6 DOP

A set of satellites which are more spread out in the sky will provide a more accurate position than a set of satellites that are close to each other. This concept is known as dilution of precision (DOP), or geometric dilution of precision (GDOP). Ideally DOP is as low as possible, representing a better positional precision caused by the wider angular separation between the satellites. This can be illustrated by a simple example where a receiver measures the ranges to two transmitters. In this example the receiver and the transmitters are all located on a single plane in space, see Figure 2.8, and there are some uncertainties in the measurements. The range from transmitter 1 to the receiver is illustrated with a bold red circular line, and the thinner red lines show the uncertainties. For transmitter 2 a green line is used. The position of the receiver is at the intersection of the green and red lines. Because of the uncertainties it is not a single intersection, but a small area.

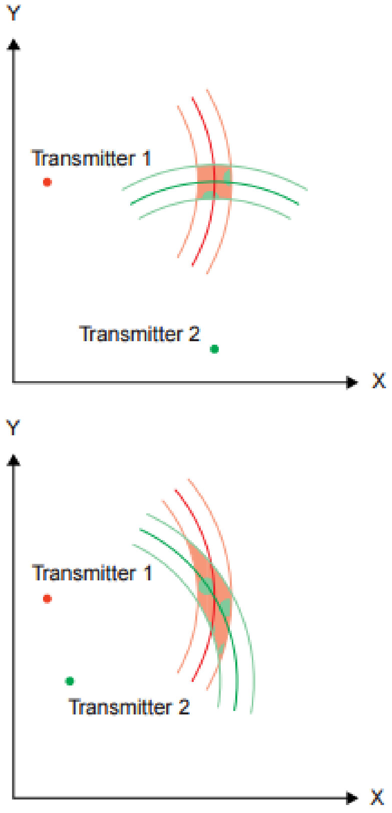


Figure 2.8: Illustrating DOP. In the top picture the transmitters are placed with wide angles with respect to the observer. In the bottom picture transmitter 2 is moved closer to transmitter 1, increasing the DOP. [17]

In the top plot the transmitters are 90° apart, seen from the receiver. This results in a relatively small region in which the receiver must lie. In the bottom plot the transmitters are closer together, especially in the *X*-direction. This results in a larger uncertainty region for the receiver, with the largest uncertainty in the Y-direction, and this shows that the precision in the bottom plot is diluted in comparison to that of the top plot.

This is a simple case in two dimensions, but the same concept is applicable for positioning of a receiver in three dimensions using satellites, where there are both positioning and timing errors. Generally the DOP value gets smaller when more satellites are used for a solution. Instead of examining the quality of the overall solution it is possible to look at specific components, such as the horizontal coordinate, vertical coordinate or the clock offset. They are known as the HDOP, VDOP and TDOP respectively. They can be expressed as

$$\begin{aligned} HDOP &= \frac{\sqrt{\sigma_E^2 + \sigma_N^2}}{\sigma} \\ VDOP &= \frac{\sigma_U}{\sigma} \\ TDOP &= \frac{\sigma_T}{\sigma}, \end{aligned} \tag{2.31}$$

and they are related to the GDOP according to

$$GDOP^2 = HDOP^2 + VDOP^2 + TDOP^2. \tag{2.32}$$

In the equations σ_E, σ_N and σ_U are the standard deviations of the receiver position in the east, north and up components. σ_T is the standard deviation of the receiver clock offset estimate and σ is the total user equivalent range error (UERE) which can be around 5 m for the standard positioning service [17].

2.7 RTKLIB algorithms

2.7.1 Extended Kalman Filter

RTKLIB employs the Extended Kalman filter (EKF) for obtaining position solutions. EKF is the nonlinear version of the Kalman filter, meaning that it works on nonlinear systems as opposed to the regular Kalman filter which only works on linear systems.

The regular Kalman filter can be broken into three basic steps:

Prediction Using a process noise model, the filter "predicts" parameters at next data epoch. A state transition matrix projects state vector (parameters) forward to the next epoch.

Kalman gain The Kalman Gain is the matrix that allocates the differences between the observations at time $t + 1$ and their predicted value at this time based on the current values of the state vector according to the noise in the measurements and the state vector noise.

Update This is the step in which the new observations are "blended" into the filter and the covariance matrix of the state vector is updated. The filter has now been updated to time $t + 1$ and measurements from $t + 2$ can be added and so on until all the observations have been added.

In order to initiate the filter it starts with an apriori state vector and covariance matrix.

The EKF works in much the same way as the Kalman filter. As stated above the state model does not have to be a linear function, instead it is enough that is differentiable. The model will for each time step be linearized using a Taylor expansion. For the details please refer to the RTKLIB manual [32] and [33].

In a standard Kalman filter the stochastic parameters obtained during the filter run are not optimum because they do not contain information about the deterministic parameters obtained from future data. A smoothing Kalman filter runs the filter forward and backwards in time, taking the full average of the forward filter at the update step with the backwards filter at the prediction step.

2.7.2 Integer Ambiguity Resolution

If the initial integer ambiguity value for each satellite-receiver pair could be determined the carrier phase measurements could be corrected to create a very precise receiver-satellite distance measurement. A solution using the corrected carrier phase observation is known as an "ambiguity-fixed" solution. In RTKLIB those are identified by the value of the quality flag Q. For a fixed solution the value of Q would be 1.

By using EKF the estimated rover position and the single-differenced carrier-phase ambiguities are obtained. The next step in order to improve the accuracy is to resolve the float carrier-phase ambiguities into integer values, refer to [33]. Eventually the integer ambiguity can be found by solving the integer least square (ILS) problem

$$\check{\boldsymbol{N}} = \min_{\boldsymbol{N}\in\mathbb{Z}}((\boldsymbol{N} - \hat{\boldsymbol{N}})^T \boldsymbol{C}_N^{-1} (\boldsymbol{N} - \hat{\boldsymbol{N}})), \tag{2.33}$$

where $\check{\boldsymbol{N}}$ is the best integer vector, $\hat{\boldsymbol{N}}$ is the double-differenced carrier-phase ambiguities. $\boldsymbol{C}$ is the covariance matrix of the system noise.

To solve this ILS problem RTKLIB employs LAMBDA [34] and MLAMBDA [3]. Then the improved rover position is obtained by solving

$$\check{\boldsymbol{r}} = \hat{\boldsymbol{r}} - \boldsymbol{C}_{RN} \boldsymbol{C}_N^{-1} (\hat{\boldsymbol{N}} - \check{\boldsymbol{N}}). \tag{2.34}$$

For more details please refer to the RTKLIB manual [32] and [33].

2.7.3 Cycle slip

Once a receiver has found and locked on to a signal from a satellite the unknown number of full wavelengths, ambiguity, between the satellite and the receiver stays constant. If there is a signal blockage, causing the receiver to temporarily lose the lock on a signal from a satellite, a cycle slip occurs. Once the receiver re-establishes the signal lock a new ambiguity will exist and must be solved for separately from the original ambiguity.

In RTKLIB cycle slips are detected by loss of lock indicator (LLI) in the input measurement data [10]. If dual-frequency measurements are available cycle slips can also be detected by monitoring jumps of the geometry-free linear combination of L1 and L2 carrier-phase.

One effect of cycle slips is that the position of the receiver might be offset by one or more wavelengths, corresponding to about 20 cm for L1, once the new ambiguity is found. This is a large jump when compared to the restrictions for the creation of high resolution SAR images and therefore it is crucial to identify and try to prevent cycle slips.

Chapter 3

GPS processing software

In this chapter the software products used to process GPS data for this thesis will be given a brief introduction.

3.1 RTKLIB

RTKLIB is a set of open source programs written by Tomoji Takasu. It is written in ANSI C (C89) and can be run on multiple OS environments. It can be used for logging, converting, downloading, streaming and processing GNSS data. RTKLIB supports data communication via serial I/O, TCP/IP connection and Networked Transport of RTCM via Internet Protocol (NTRIP). Various data formats are supported including RTCM 3.1 [29] and raw data from common GPS receivers. The parts of RTKLIB used in this work are the ones used for post-processing RTK and converting raw GPS data to the Receiver Independent Exchange Format (RINEX).

From the version 2.2.0 RTKLIB has been distributed under the GPLv3 license.

3.2 Waypoint

The software originally used in the CARABAS system is GrafNav and GrafMov from Waypoint Software [12]. GrafNav is used for flight paths and GrafMov for baselines. Instead of using RINEX files Waypoint uses their own observation format.

Chapter 4

SAR image acquisition

The main steps involved in creating an image using CARABAS is presented in Figure 4.1. This thesis focuses on the steps which has to do with GPS processing, shown on the top row.

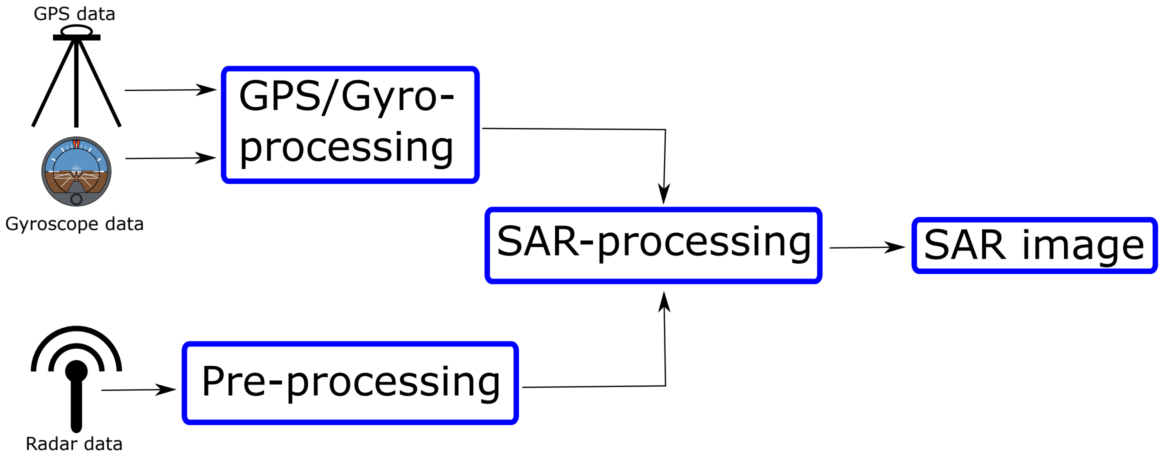


Figure 4.1: Processing steps for creation of a CARABAS III image.

GPS data from three antennas are used during the process. Two of the antennas are mounted on a helicopter and the third antenna is placed on the ground at a known position. The GPS data is the starting point of the processing together with the radar data and the gyroscope data. After acquiring GPS data they are passed through a software in order to determine the flight path and orientation of the aircraft. This is where RTKLIB is used. In the SAR-processing step a so called pass file is used to determine what data should be used. One piece of information found in the pass files is the time for each pass made during the flight, i.e. the start and stop times for each synthetic aperture.

4.1 SAR principle

During data collection for a single SAR image radar pulses are emitted in order to illuminate a target scene, and the echo of each pulse is recorded. CARABAS makes use of a radar that operates in the VHF band where the wavelengths are about 1-15 m. The radar antenna moves relative to the target scene as the data is collected and it is possible to distinguish the different pulses and identify them. The collected data basically shows a recording from multiple antenna locations which is what forms the synthetic antenna aperture. The final resolution of an image created using this technique is finer than what would be possible with the given physical antenna aperture [28, p. 3].

Consider the case when the radar pulses are sent from an antenna located on an airplane flying in a straight line over a target scene with three stationary objects. This is illustrated in Figure 4.2. The blue, red and green squares represents three objects at different locations.

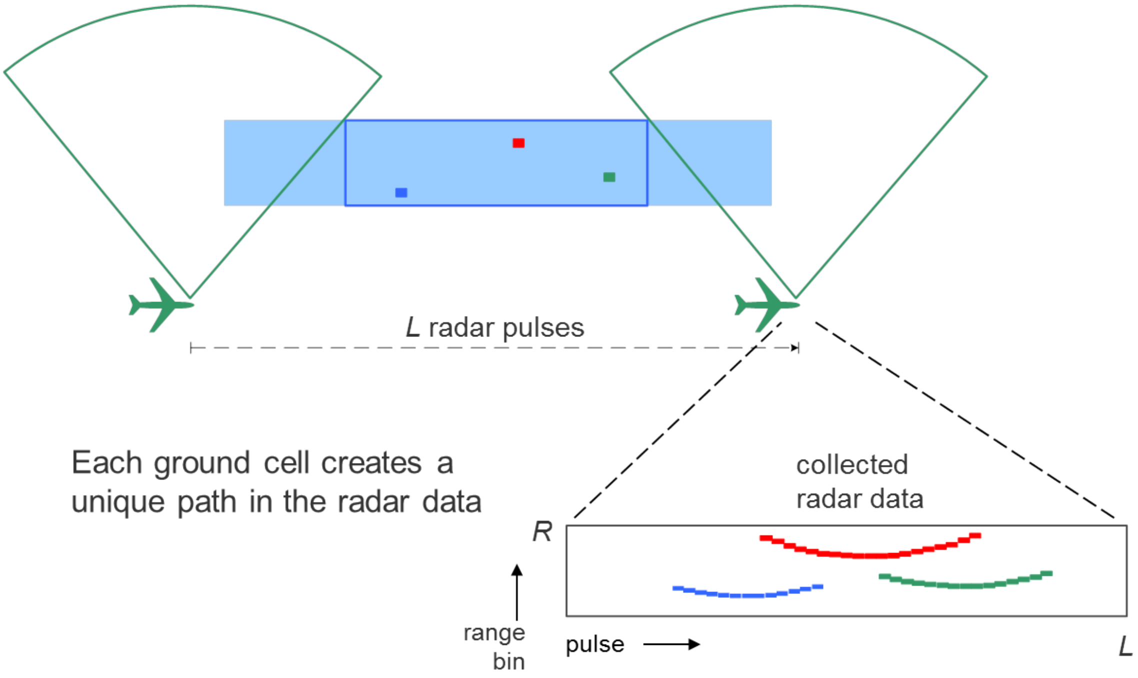


Figure 4.2: Illustration of radar data collection for one SAR image. Image courtesy Anders Åhlander.

As the plane passes the target scene the relative distance to the objects changes. This leads to each object creating a unique path in the radar data, as seen in the rectangle on the bottom right of the Figure.

In order to combine the radar data from all the pulses to create the final image a technique called Fast Factorized Backprojection[35] is used. Briefly described two neighbouring pulses are combined to create a sub-aperture, then two neighbouring sub-apertures are combined to create a bigger sub-aperture and so on until the all sub-apertures have been combined to create the final aperture.

As an example of how it might look Figure 4.3 presents one recording and the resulting image from a flight over Linköping.

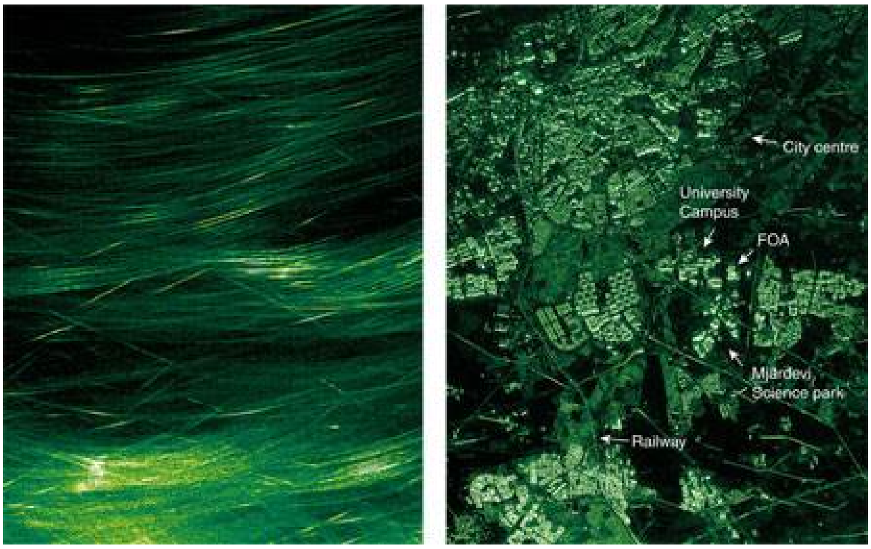


Figure 4.3: Recording to the left and resulting image to the right.

Chapter 5

Method

In this chapter the measurement and setup used will be described, together with the statistics used for comparing the two softwares.

5.1 Data collection

A small helicopter is used for collecting data in the CARABAS system, see Figure 5.1. Two GPS antennas are attached to the tail boom of the helicopter with a distance of approximately 1.2 m. From now on they will be referred to as the front and rear antennas.

A third GPS antenna is positioned stationary on the ground and functions as a base station. The base station updates at 5 Hz, and the front and rear antennas at 10 Hz. After a flight the flight path and orientation of the helicopter should be determined by post processing the collected GPS data from all antennas.

5.1.1 Flight path

The data from the front and rear antennas can be used to determine their positions during the flight. This is made by combining their data with that from the base station using RTK. The positions of one antenna are given in latitude, longitude and height based on the WGS84 datum. The most crucial part of the flight path is the part flown when creating the SAR images which is mostly made on somewhat straight lines. Obviously only position data from those parts are the ones that are used in the SAR image processing, but in this thesis positions from the whole flight path will be used as well for evaluating the performance of RTKLIB.

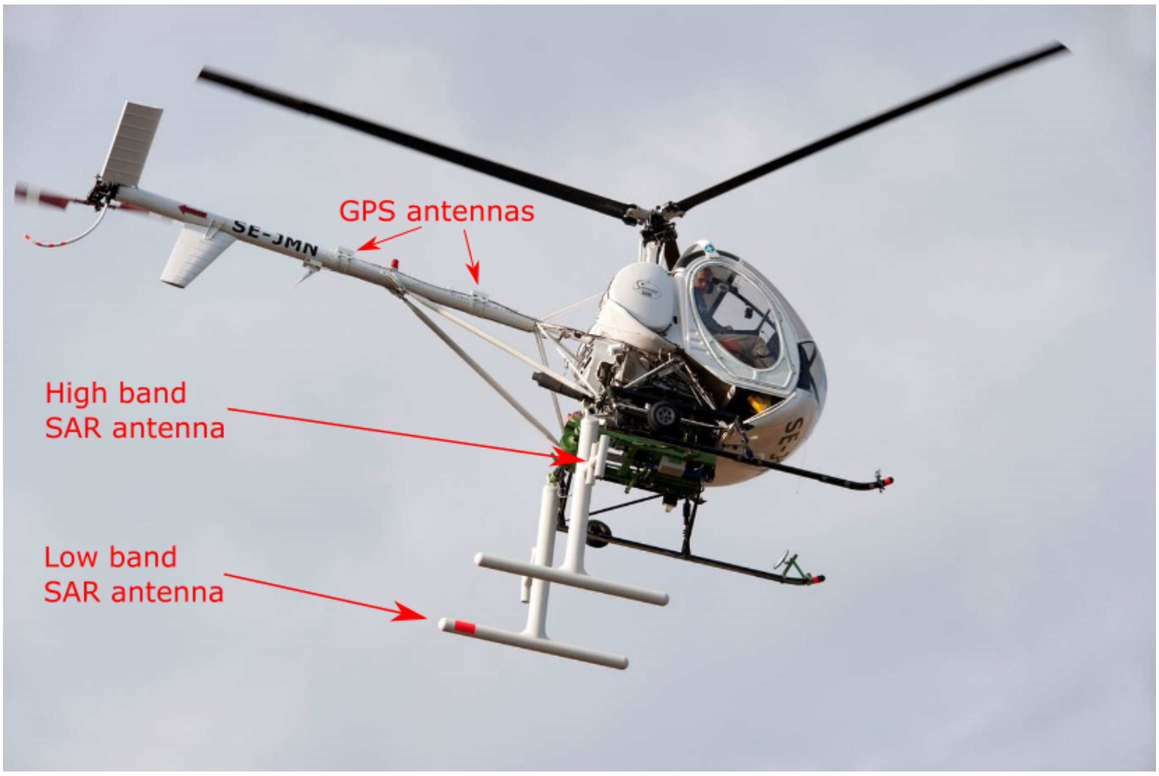


Figure 5.1: Picture of the helicopter used for collecting data. The placement of the high band and low band radar antennas are shown, as well as the GPS antennas. Image courtesy Saab AB.

5.1.2 Orientation

To find the orientation of the helicopter the moving baseline between the front and rear antennas should be determined as well. It is called a moving baseline since both of the antennas might be moving, and therefore the baseline between them will not be the same for all epochs during the flight. Note that for regular RTK positioning one of the antennas are placed on a fixed location.

The coordinate system used is a local ENU defined such that the origin is found at the start of the vector which in this thesis will be the rear antenna. The y-axis coincides with true north and the x-axis coincides with true east. The z-axis is defined vertically. As for the flight path only parts of the data are actually used in the SAR image processing, but baseline data from the whole flight will be used for evaluating RTKLIB.

Describing the rotation of an aircraft one usually uses the terms roll, pitch and yaw. They refer to rotations about certain axes of the aircraft, see Figure 5.2. The roll axis passes through the aircraft from the nose to the tail, the pitch axis passes through the aircraft from

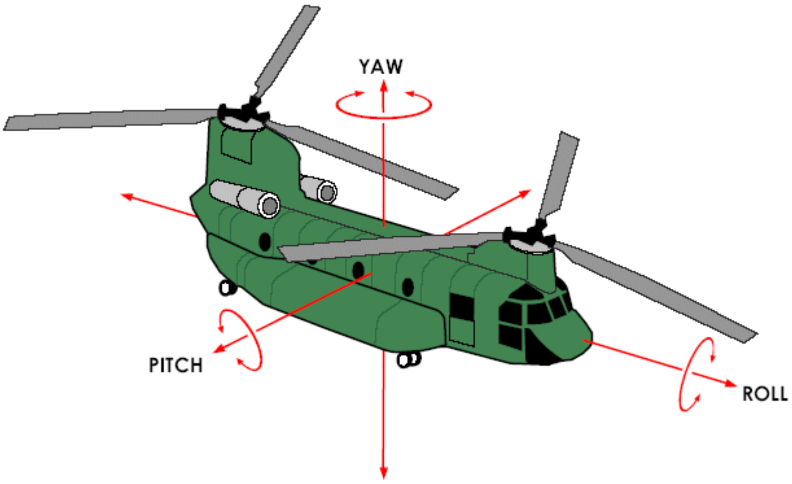


Figure 5.2: Roll, pitch and yaw axes [20].

the left side to the right side of the aircraft perpendicular to the roll axis. The yaw axis is perpendicular to the plane created by the roll and pitch axes.

Since only two antennas are attached to the helicopter in the CARABAS system the full orientation can not be determined without extra inputs. Therefore a MEMS IMU gyroscope is used for the third unknown angle. The gyroscope is calibrated during flights by performing "calibration manoeuvres", which are done by changing the heading of the helicopter at regular intervals [27].

5.2 Receivers

Two Javad Delta receivers were mounted on the tail boom of the helicopter. They can receive and process multiple signal types [13]. For the flights they provide measurement data for post processing at 10 Hz. For the flights compared in this thesis the only recorded data comes from GPS satellites, but they could be used to receive signals from GLONASS and Galileo as well.

The base station used is a dual-frequency Novatel receiver. It provides measurement data for post processing at 5 Hz. Both the receivers mounted on the helicopter and the base station provide data from the L1 and L2 frequencies.

5.3 Measurements

In order to evaluate RTKLIB data from flights made in October 2014 will be used. During aquisition of SAR data the helicopter flew at heights of 30, 250, 350 and 800 m respectively. The data will be processed using the command line tool rnx2rtkp with settings that are presented in Appendix C. Since the true flight path and moving baselines are not known the output from the currently used software will be taken as a reference. This means that it will be impossible to know if the results are completely valid, only how the result from RTKLIB differs from that of the other software. It should be noted that the result from RTKLIB could be closer to the correct value, or it could be further away from the correct value than the reference.

5.4 Statistics

To compare the different positioning results the mean difference, standard deviation of the difference, root mean square error of the difference, weighted mean difference and maximum difference will be studied. Assume there are N values, and the reference values and RTKLIB values are denoted $\hat{y}_i$ and y_i ($i \in 1,N$), respectively. Then the mean difference, $\bar{d}$, is simply defined as

$$\bar{d} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i). \tag{5.1}$$

The standard deviation of the differences, σ_d is defined as

$$\sigma_d = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((\hat{y}_i - y_i) - \bar{d})^2}. \tag{5.2}$$

The root mean square error (RMSE) of the difference is defined as

$$(RMSE) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \tag{5.3}$$

or equivalently

$$(RMSE)^2 = \bar{d}^2 + \sigma_d^2. \tag{5.4}$$

The mean difference gives information about how much the result differ from the reference on average, and the standard deviation gives information about the amount of variation about the mean difference. The RMSE of the difference instead describes the variation about 0.

The weighted mean difference uses the formal errors corresponding to each output position value. This way output values which are more uncertain, i.e. the ones with a large formal error, will not contribute the same amount to the result as values with smaller formal errors. It can be defined according to

$$\bar{d}_w = \frac{\sum_{i=1}^{N} w_i(\hat{y}_i - y_i)}{\sum_{i=1}^{N} w_i}, \tag{5.5}$$

where

$$w_i = \frac{1}{\sigma_i^2} \tag{5.6}$$

and

$$\sigma_i = \sqrt{\sigma_{\hat{y}_i}^2 + \sigma_{y_i}^2}. \tag{5.7}$$

Here $\sigma_{\hat{y}_i}$ is the formal error corresponding to $\hat{y}_i$ and σ_{y_i} is the formal error corresponding to y_i.

Chapter 6

Results

In this chapter the results from the comparisons of flight paths and moving baselines between RTKLIB and Waypoint will be presented. The accuracy of RTKLIB is evaluated using the result from Waypoint as a reference. The same input GPS data was used by both softwares. Since some settings of Waypoint are hidden a perfect comparison is impossible to do. The value of the quality flag, Q, which describes whether the integer ambiguities are resolved or not in RTKLIB and Waypoint was also studied. The value of Q can be either 1 or 2, where 1 means that the integer ambiguities are properly resolved and 2 means that they are not. The value of Q is determined in both programs using a statistic test.

In order to keep this chapter from overflowing with tables and plots most results can be found in Appendix A. The tables and figures in this chapter are chosen by how they represent the general results.

6.1 Flight paths

A typical flight path is presented in Figure 6.1. The color of the marker corresponds to different values of the quality flag Q. Green parts of the plot correspond to where the integer ambiguities are properly resolved, and the yellow parts where they are not resolved but the solution is still found by carrier-based relative positioning. The straight lines making up the square show where the helicopter flew when collecting radar data for the SAR images. Those parts of the flight path are known as test runs. It is for the test runs that the accuracy of RTKLIB needs to fulfill the stated requirements. The rest of the flight path consists of the helicopter repositioning itself for the test runs, and making some calibration movements for the gyroscopes on board.

The output from RTKLIB and GrafNav was converted from geodetic coordinates (latitude, longitude and height) to Cartesian coordinates with origin in 58° 15′ 0.324″N 15° 34′ 57.468″E 151.5 m. That is a point close to where all flights are made (mostly within 5 km),

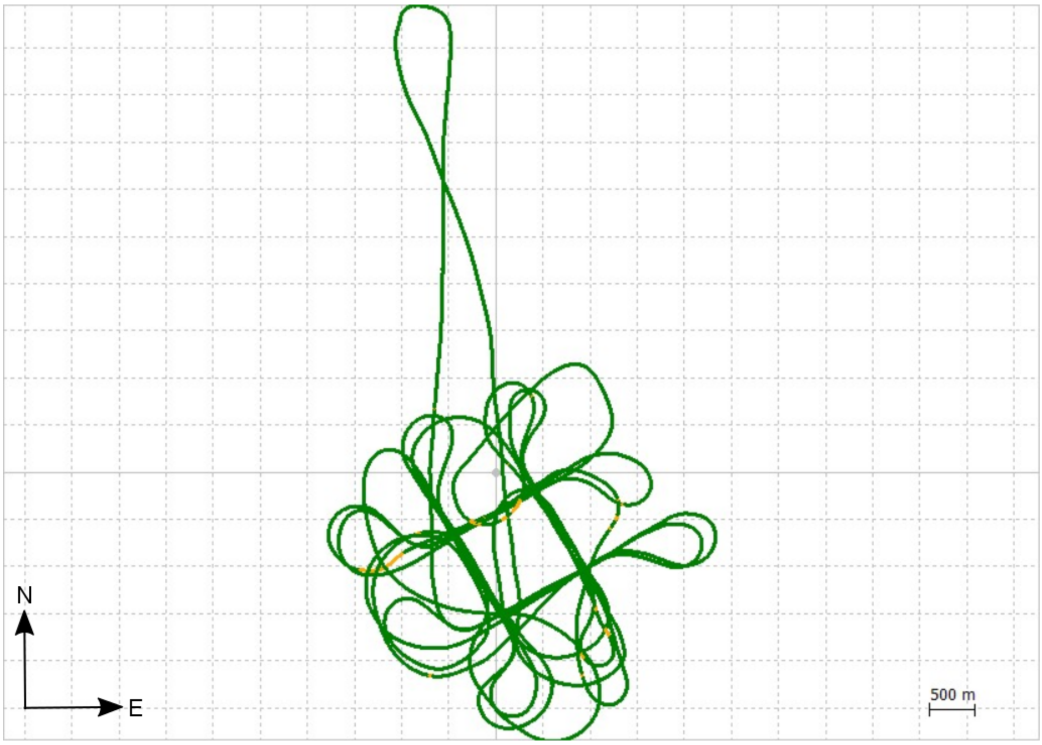


Figure 6.1: Example of a flight path.

ensuring that the local coordinate system is valid. Conversions were made using provided MATLAB functions from the CARABAS project signal processing chain.

The results from the comparisons will be presented using tables and plots. Each table will contain columns for mean difference, standard deviation of the difference, RMS of the difference, weighted mean difference and maximum difference. In the caption of each table additional information is found:

$qRat_R$ is the fix ratio of the RTKLIB positioning results, i.e. the ratio between the number of epochs where the integer ambiguity is fixed, and the total number of epochs.

$qRat_G$ is the corresponding fix ratio of the GrafNav positioning results.

nValues is the total number of values compared.

At the beginning of each caption is a string used for identifying from where the data originates. The first three characters describe the antenna, which can be either "AFT" or "FWD". The following four numbers describe the date of the flight. The final letter describes the flight number of the day, starting at "a" for the first flight and then "b" for the second and so on. From "AFT1001a" one can see that the antenna used was the "AFT"-antenna, the date flown was October 1st , and that it was the first flight of the day.

6.1.1 2014-10-03

The results for the second flight of October 3rd can be found in Table 6.1, and the corresponding Figure 6.2. RTKLIB has a slightly higher fix ratio than GrafNav for the compared values. The mean difference is less than one centimeter in both the x and y-direction, and as expected the difference is a bit larger for the z-direction. Note that the positioning requirement is fulfilled.

Table 6.1: FWD1003b, $qRat_R$ = 0.988, $qRat_G$ = 0.944, nValues = 19865

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	6.3	16.6	17.8	5.3	52.5
Y (mm)	-5.9	24.0	24.7	-2.5	83.1
Z (mm)	-36.5	34.7	50.4	-36.3	171.6

In the plots on can see that the differences in the horizontal directions are almost centered around 0 m. The histograms shows that there is an offset of about 0.05 m in the vertical direction. All in all this represents a good result as the differences are small and RTKLIB had a high fix ratio.

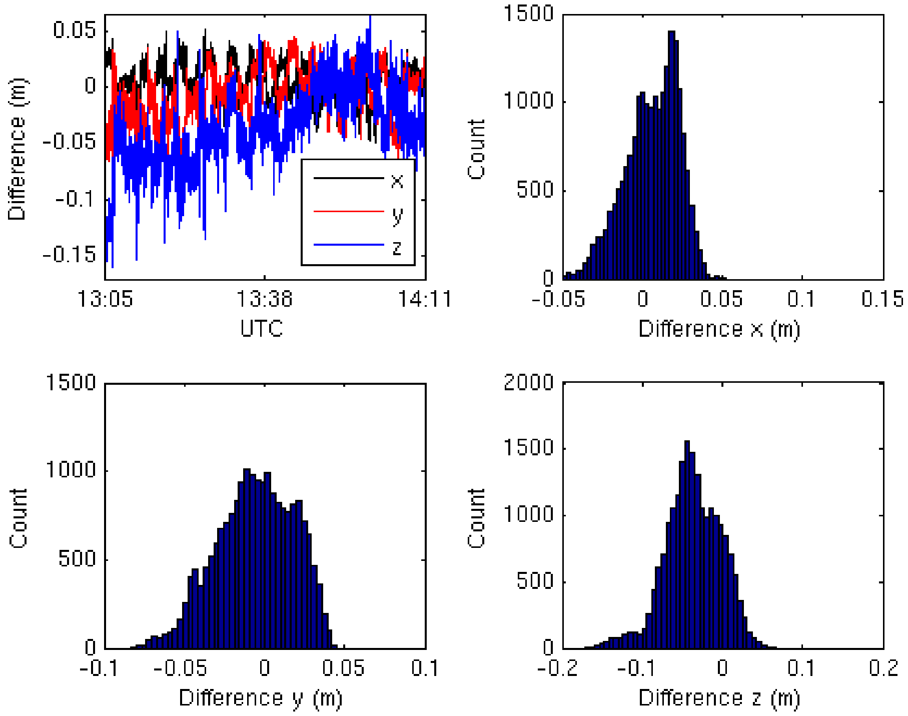


Figure 6.2: Plot and histogram displaying differences in x-, y- and z-directions for FWD1003b.

6.1.2 2014-10-06

The results for the third flight of October 6th can be found in Table 6.2 and Figure 6.3. For this flight RTKLIB has a significantly lower fix ratio than GrafNav. This is caused by some cycle slips, which GrafNav has a certain algorithm [12, p. 4] for correcting that RTKLIB does not have a counterpart for.

As expected the vertical direction is the one with the largest differences again. The resulting positions in the x-directions are a little bit better than the positions in the y-direction.

Table 6.2: FWD1006c, $qRat_R = 0.695$, $qRat_G = 0.942$, nValues = 13745

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	4.2	63.7	63.7	5.2	190.7
Y (mm)	21.7	84.5	87.3	20.3	315.4
Z (mm)	-0.6	172.1	172.1	-6.7	528.7

The result of the cycle slips can easily be seen in the plot where the differences jump at different times. In the histograms this is seen by the smaller peaks that are not centered around 0 m. Note that the accuracy requirements are not fulfilled for this flight.

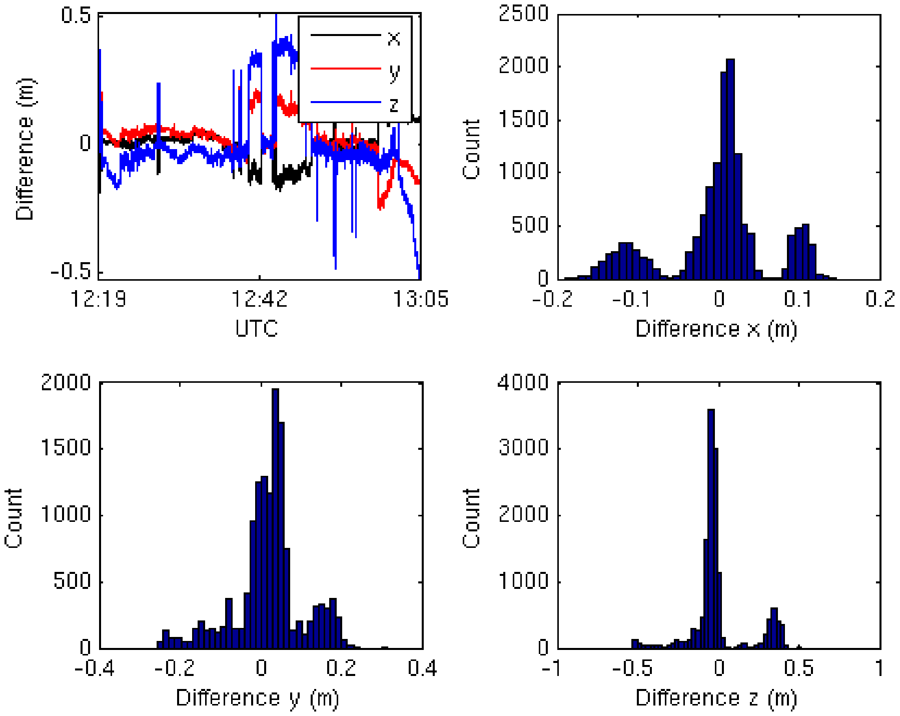


Figure 6.3: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c.

6.2 Individual test runs, flight paths

The positioning data used for creating SAR images will be extracted using external information where time tags are used for setting the start and stop time of the radar collection. The part of the flight where data is collected for one image is called a test run. In this section the results from a few test runs will be presented to give a better understanding of how the positioning of RTKLIB performs during the crucial radar data collection. The caption of the tables contains additional information:

h is the height above the ground flown by the helicopter.

l is the distance flown while collecting radar data.

At the beginning of each caption is a string used for identifying the test run.

6.2.1 2014-10-01, AFT1001a

In Table 6.3 and Figure 6.4 the results from one of the test runs from October 1st are presented. The data comes from the AFT-antenna on the first flight of the day. The helicopter flew at 250 m above the ground and both RTKLIB and GrafNav fixed the integer ambiguity resolution during the whole test run. Once again the largest differences are found in the vertical direction. It is worth noting that the maximum difference is still very small, only 58.9 mm.

Table 6.3: 15_E0904211, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 209, h = 250 m, l = 1015 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-6.5	3.5	7.4	-6.5	15.2
Y (mm)	-22.8	7.7	24.1	-22.9	37.2
Z (mm)	-15.3	20.5	25.5	-15.3	58.9

Looking at the histograms it is obvious there is a small offset between the solutions, and the offset is the smallest for the x-direction.

This test run illustrates well the behaviour for most test runs compared to full flights. The differences between RTKLIB and GrafNav is generally smaller during the test runs, and the fix ratio is usually very high for RTKLIB, cf. Table A.1.

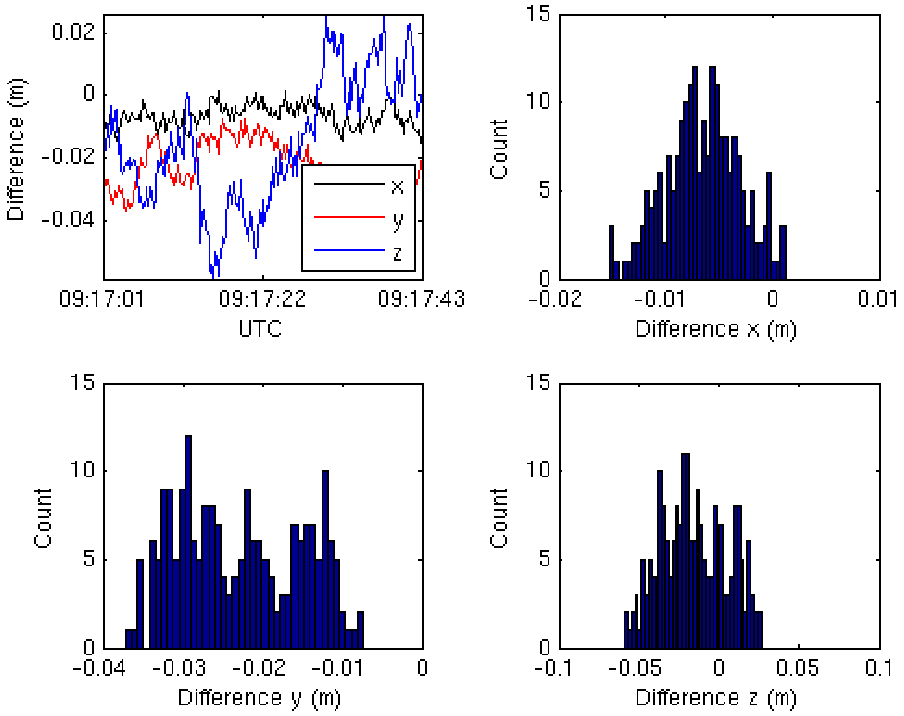


Figure 6.4: Plot and histogram displaying differences in x-, y- and z-directions for AFT1001a test run 15_E0904211.

6.2.2 2014-10-06, FWD1006c

In Table 6.4 and Figure 6.5 the results from one of the test runs from October 6th are presented. In this test run a cycle slip occurred which caused the integer ambiguity to be reinitialized by RTKLIB. After the reinitialization the position relative to that produced by GrafNav was shifted which can be seen by the jump at about 12:44:18 in the plot. A short while later RTKLIB corrects the integer ambiguity and the differences jump back to smaller values.

Table 6.4: 13_E0915508, $qRat_R$ = 0.721, $qRat_G$ = 0.446, nValues = 258, h = 250 m, l = 492 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-37.8	60.0	70.9	-34.6	153.3
Y (mm)	58.2	74.8	94.8	55.3	194.0
Z (mm)	102.1	166.0	194.9	91.9	406.7

By studying the position file produced by RTKLIB it can be seen that one satellite is the cause of the loss of the integer ambiguity. Using the solution status file, which is output by RTKLIB if the corresponding parameter is set, the satellite could be identified. It was found to be a satellite at a low elevation angle, slightly above 15°. Note that all this information is independent of the GrafNav results, meaning it is possible to identify the cause of similar problems without reference data.

By excluding the satellite from the processing the result significantly improved for the studied test run, see Figure 6.6. After excluding the satellite this test run fulfills the requirements on the accuracy. This is however not a perfect way of solving this kind of problems that can occur. By removing one satellite from the processing other test runs were affected, and even though most of the test runs improved there were two for which the result degraded. But it is a technique that can be used if no other solution is found for fixing the cycle slips.

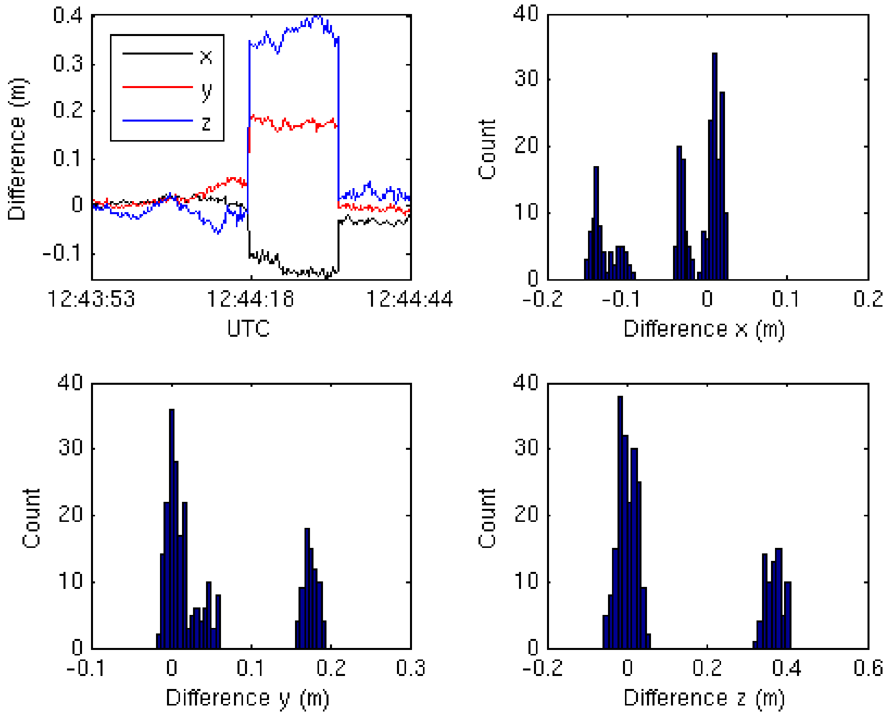


Figure 6.5: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 13_E0915508.

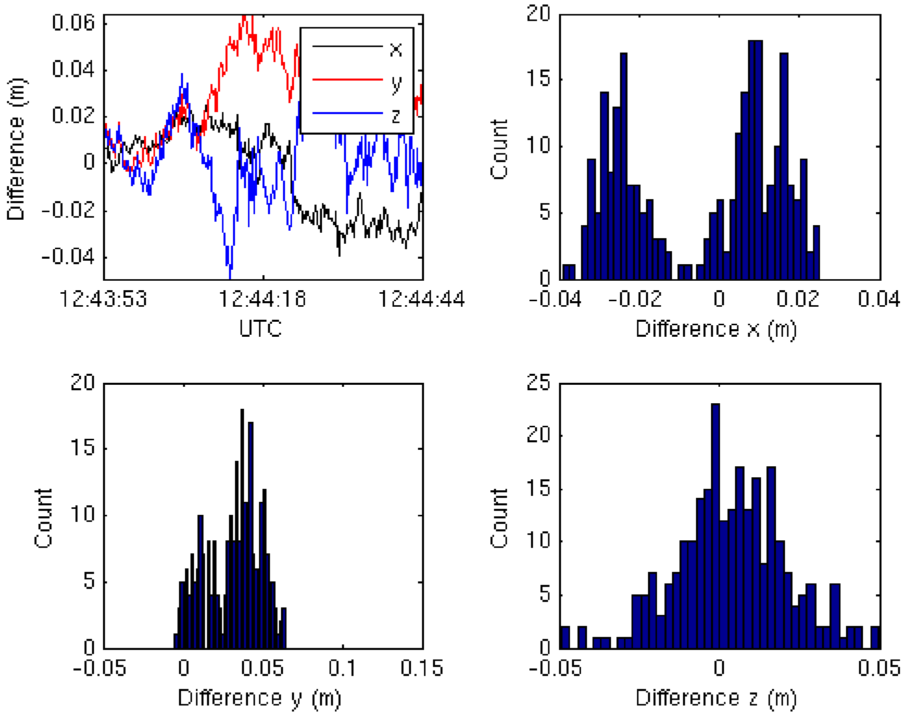


Figure 6.6: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 13_E0915508, with satellite G15 excluded from the solution.

6.3 Orientation

A typical set of moving baselines from one flight is presented in Figure 6.7, where the horizontal projections of the baseline vectors are shown. The plot consists of one marker for the x and y coordinates at each epoch. It looks like a circle because the helicopter is rotating around more than 360° during the flight. The origin of the moving baselines was set to be the AFT-antenna, meaning that each marker corresponds to the relative position of the FWD-antenna with respect to the AFT-antenna. From the plot it can be found that the baseline lengths is approximately 1.18 m.

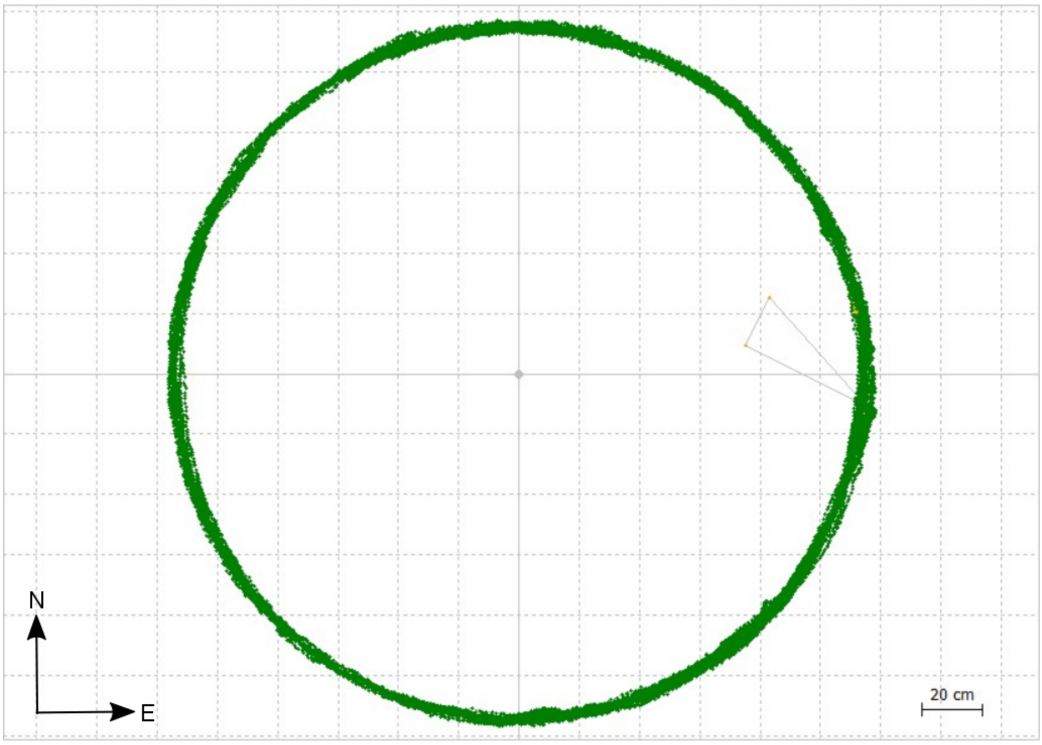


Figure 6.7: Set of moving baselines from all the epochs in one flight.

The results from the comparisons will be presented using tables and plots for the moving baselines as well, with basically the same information as the flight paths. The output used as a reference does not provide formal error, meaning the weighted mean difference will not be presented in the tables regarding moving baselines. Additionally there are two angles compared. The first angle is the azimuth which is defined as the angle between the true north and the projection of the baseline onto the horizontal plane. The second angle, θ, is the actual angle between the reference baseline and the new baseline. Also, note once again that the values used as a reference does not have to be the true positions.

6.3.1 2014-10-03

In Table 6.5 and Figure 6.8 the moving baseline results for the second flight of October 3rd are presented. These are the moving baselines corresponding to the flight path presented above for October 3rd. As for the flight path the fix rate is slightly higher for RTKLIB. The mean differences is on the millimeter scale, and the largest difference is 64.4 mm. The standard deviation of the compared moving baseline angles is lower than 1°, thus fulfilling the stated requirement.

Table 6.5: Baseline1003b, $qRat_R$ = 0.999, $qRat_G$ = 0.996, nValues = 39719

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	1.2	3.0	3.2	19.7
Y (mm)	0.5	5.3	5.3	35.0
Z (mm)	-1.2	9.3	9.4	64.4
Az (°)	-0.008	0.185	0.185	1.611
θ (°)	0.394	0.293	0.491	3.286

In the plots we can see the difference in x-, y- and z-directions for the different epochs, all of which are centered around 0 m. The angle differences are also presented. When there are spikes in one of the three Cartesian coordinates a corresponding spike is found in the angle differences.

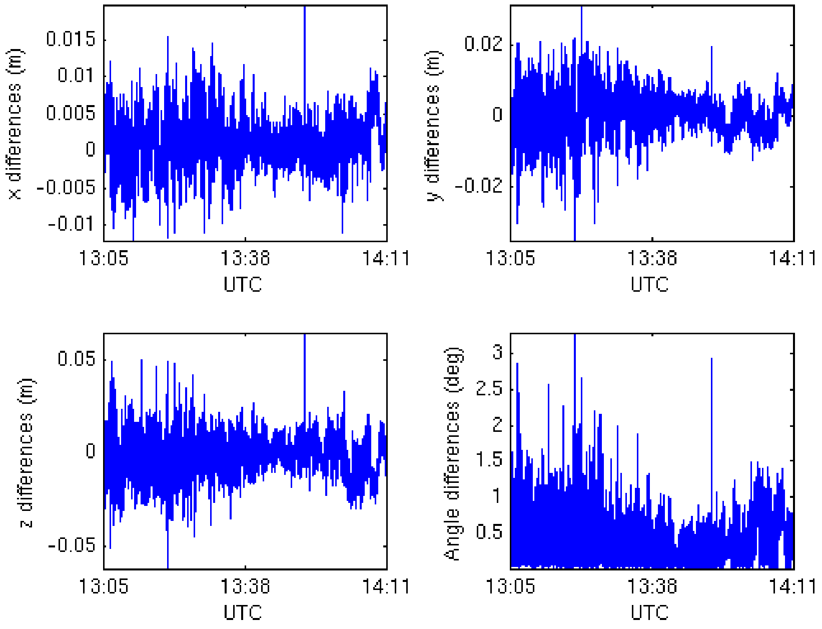


Figure 6.8: Plots displaying differences in x-, y-, z-directions and angles 1003b.

6.3.2 2014-10-06

In Table 6.6 and Figure 6.9 the moving baseline results for the third flight of October 6th are presented. These are the moving baselines corresponding to the flight path presented above for October 6th. As opposed to the flight path RTKLIB does not have a significantly lower fix rate. It is in fact slightly higher than that of GrafMov. Note though that the RMS and standard deviation of the differences between the angles are large, above 2°.

Table 6.6: Baseline1006c, $qRat_R = 0.979$, $qRat_G = 0.952$, nValues = 27328

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-1.4	34.0	34.0	3769.9
Y (mm)	0.2	14.6	14.6	415.2
Z (mm)	-1.1	91.2	91.2	7821.4
Az (°)	0.174	0.666	0.688	26.195
θ (°)	1.052	2.244	2.479	73.453

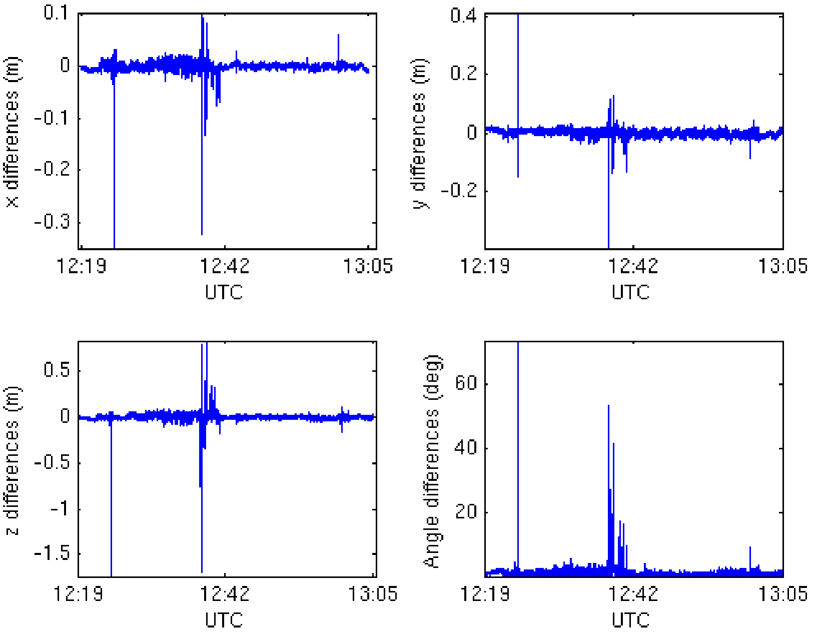


Figure 6.9: Plot displaying differences in x-, y-, z-directions and angles for 1006c.

A large spike can be seen early on in the flight. By comparing the east and north components of the moving baselines it is clearly seen that the spike is caused by a large jump in the position by GrafMov, see Figure 6.10. This illustrates that it is not always RTKLIB that is the cause of large differences, sometimes it is the reference positions that are off.

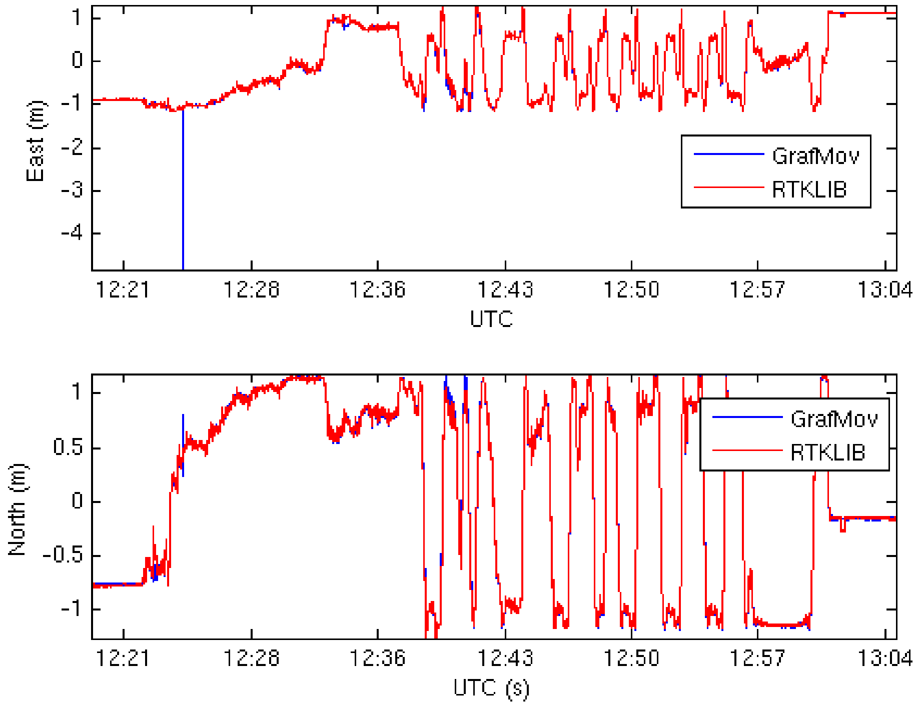


Figure 6.10: Plot displaying in east and north components for BASELINE1006c.

6.4 Individual test runs, orientation

6.4.1 2014-10-01, BASELINE1001a

In Table 6.7 and Figure 6.11 the results from one of the test runs from October 1st are presented. These are the moving baselines corresponding to the flight path test run presented earlier. The stated accuracy requirements for the baselines are clearly fulfilled for this test run.

Table 6.7: 15_E0904211, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 418, h = 250 m, l = 1015 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-5.1	2.4	5.7	12.6
Y (mm)	6.9	3.8	7.8	16.3
Z (mm)	5.5	8.0	9.7	29.3
Az (°)	-0.196	0.098	0.219	0.501
θ (°)	0.366	0.234	0.434	1.278

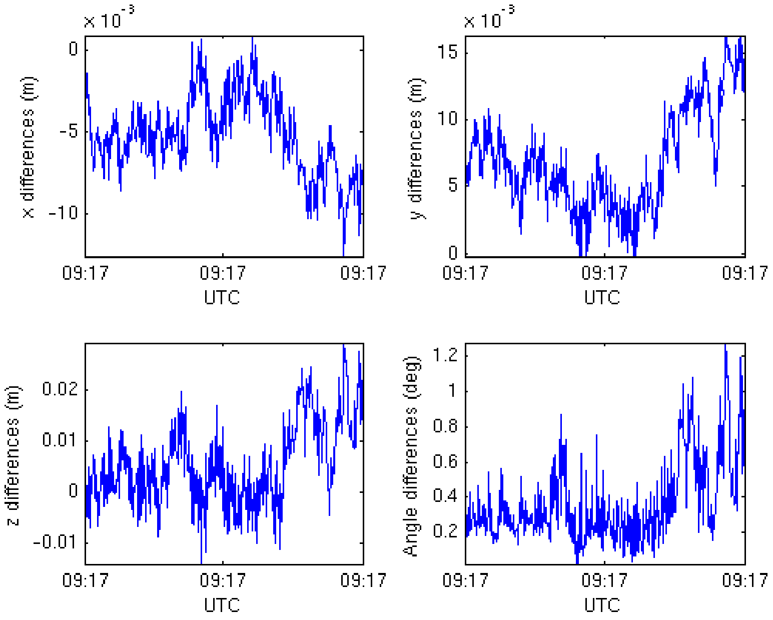


Figure 6.11: Plot displaying differences in x-, y-, z-directions and angles for 1001a test run 15_E0904211.

6.4.2 2014-10-06, BASELINE1006c

In Table 6.8 and Figure 6.12 the results from one of the test runs from October 6th are presented. These are the moving baselines corresponding to the flight path test run presented earlier where there was a cycle slip present. Note that the fix ratio is 1 for RTKLIB, a clear indication that the cycle slip did not affect the baselines.

Table 6.8: 13_E0915508, $qRat_R$ = 1.00, $qRat_G$ = 0.907, nValues = 516, h = 250 m, l = 492 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	4.7	4.7	6.7	29.0
Y (mm)	-1.9	7.5	7.8	22.2
Z (mm)	-9.9	15.8	18.6	75.2
Az (°)	-0.013	0.293	0.293	1.130
θ (°)	0.736	0.529	0.906	3.819

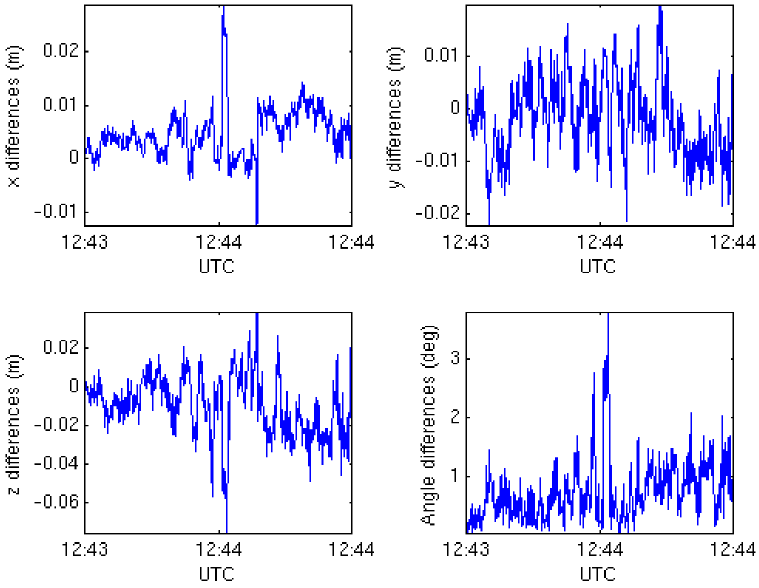


Figure 6.12: Plot displaying differences in x-, y-, z-directions and angles for 1006c test run 13_E0915508.

6.5 Summary of test runs

There are a total of 26 test runs presented in this thesis. For the flight path data 23 of them fulfill the stated requirements and 3 did not fulfill them. This is for the data created using the default configuration files presented in Appendix C. For those that failed it was possible to modify the settings to fix individual test runs, however as stated earlier this affected other test runs as well. The standard deviation between the reference position and RTKLIB was used for judging the test runs as a small global shift of the flight path does not affect the SAR images significantly. This means that a small offset in positions would not really alter the images, but that offset would make the values of the RMS in the tables increase.

For the orientation there were 22 test runs that fulfilled the requirements, and 2 that did not fulfill the requirements. This leaves 2 test runs, A.48 and A.49, where it depends on whether the azimuth (horizontal projection) or actual angle of the moving baselines affects the image quality. This depends on how the calibration maneuver for the gyroscope calibration is made, and since the calibration is usually made by turning the helicopter the azimuth is the more important angle. Then 24 test runs fulfilled the requirements, and 2 failed. Whereas for the full angle 22 fulfilled the requirements and 4 failed.

6.6 Quality flag

The goal of studying the quality flag was to find out whether the helicopter movements somehow affected the value of Q. Therefore the relation between the velocity of the helicopter and the quality flag was analysed. The values of Q for the first flight of October 7th using the FWD-antenna are presented in Figure 6.13. For this flight the fix ratio was 96 %, i.e. for 96 % of the epochs the integer ambiguities were fixed, and for the rest they were not.

The absolute velocities for that flight are presented in Figure 6.14, and the corresponding velocity components in Figure 6.15. Apart from the outlier in the y component of the velocity around 09:40 the solution looks good. Comparing the epochs where the value of the quality flag is 2 with the velocities no discernible pattern can be seen. It does not seem like a high velocity would affect the Q-value, at least not for the velocities reached by the helicopter.

The relation between the length of the baseline from base station to the helicopter was also studied, see Figure 6.16. At first glance it might seem like a short baseline cause problems here, but the reason for the large amount of Q=2 values is simply that for most of the flight the helicopter was close to the base station. I.e. there are a lot more data points in the region 100-400 m than in other regions.

In most of the data available from other flights the distance between the reference station and the helicopter is below 10 km, and the maximum length was 20 km. For these

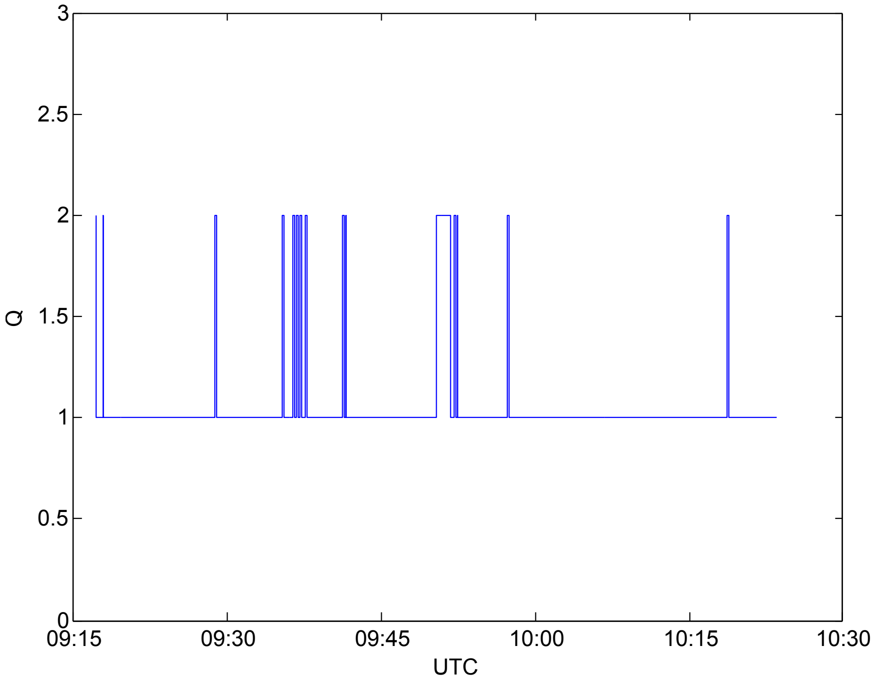


Figure 6.13: Plot displaying the values of the quality flag Q for FWD1007a.

relatively short distances no clear signs were found that the quality flag was affected.

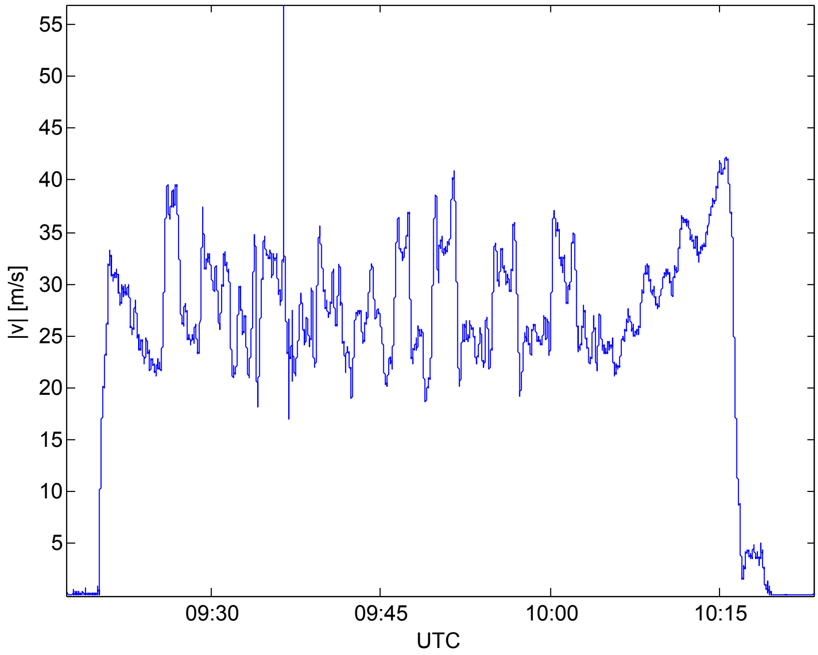


Figure 6.14: Plot displaying the absolute velocities for FWD1007a.

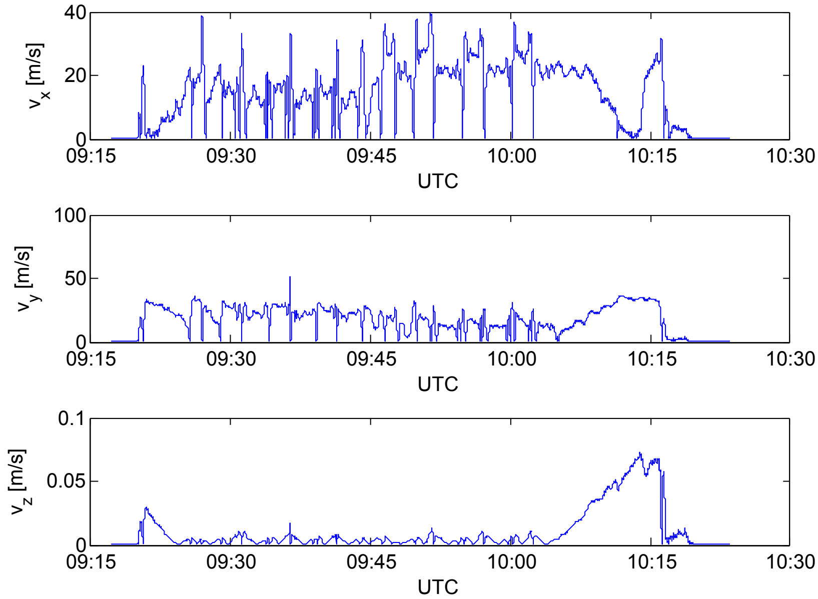


Figure 6.15: Plot displaying the velocity components for FWD1007a.

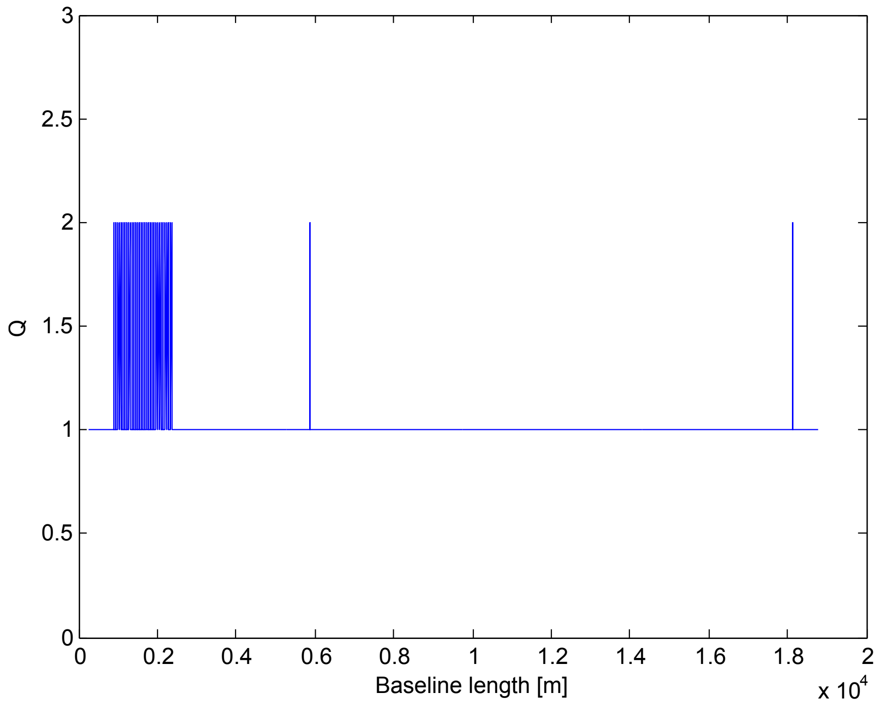


Figure 6.16: Plot displaying the Q values for different baseline lengths for FWD1007A.

6.7 Final image

The absolute comparison between RTKLIB and the Waypoint software is made by comparing the resulting CARABAS images created using the positioning data from both softwares. For this flight data from Febraury 8th 2013 was used. This particular flight was chosen since the image would contain an artificial calibration reflector.

In Figure 6.17 the SAR image created using the original software is presented. The corresponding image created using RTKLIB is found in Figure 6.18. The color of the pixels follows a logarithmic scale which describes the intensity of each pixel relative to other pixels. A logarithmic scale is used since otherwise large part of the picture would be completely black. The *x*-axis corresponds to the east direction, and the *y*-axis to the north direction. Comparing the two images they are close to identical. No information is lost using RKTLIB, and there is even a small improvement on the image in the area close to the reflector as it appears slightly sharper. The reflector can be found in the center of the images.

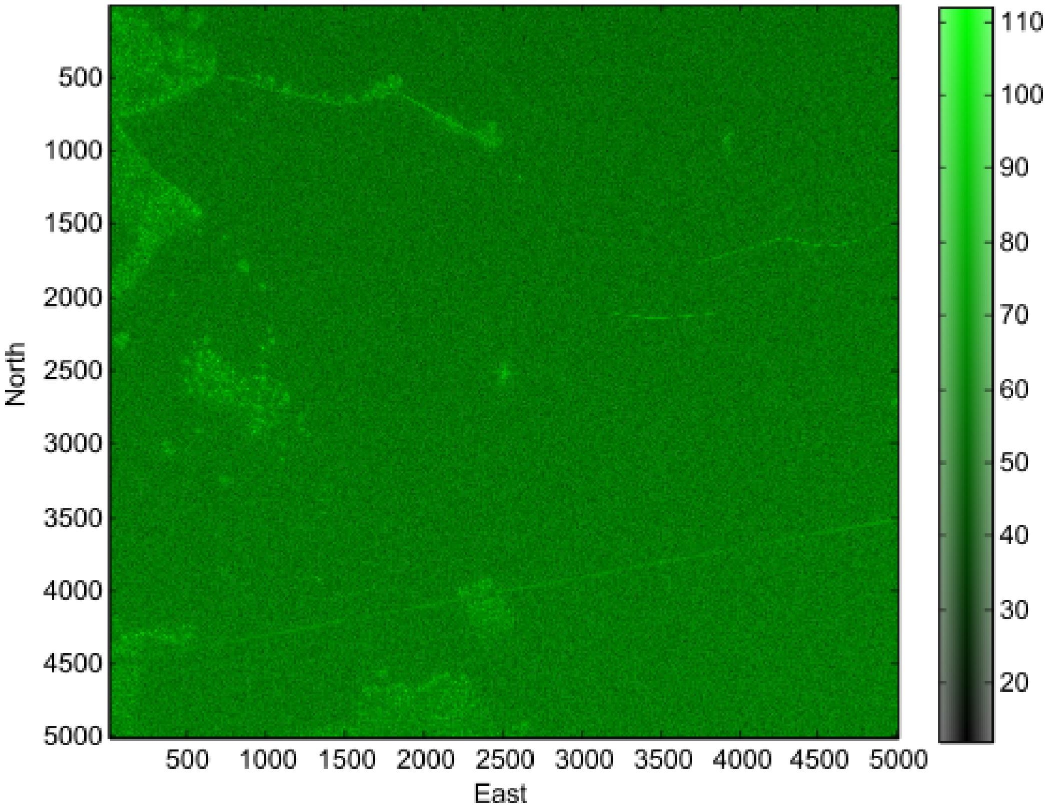


Figure 6.17: CARABAS image created using positioning data from Waypoint.

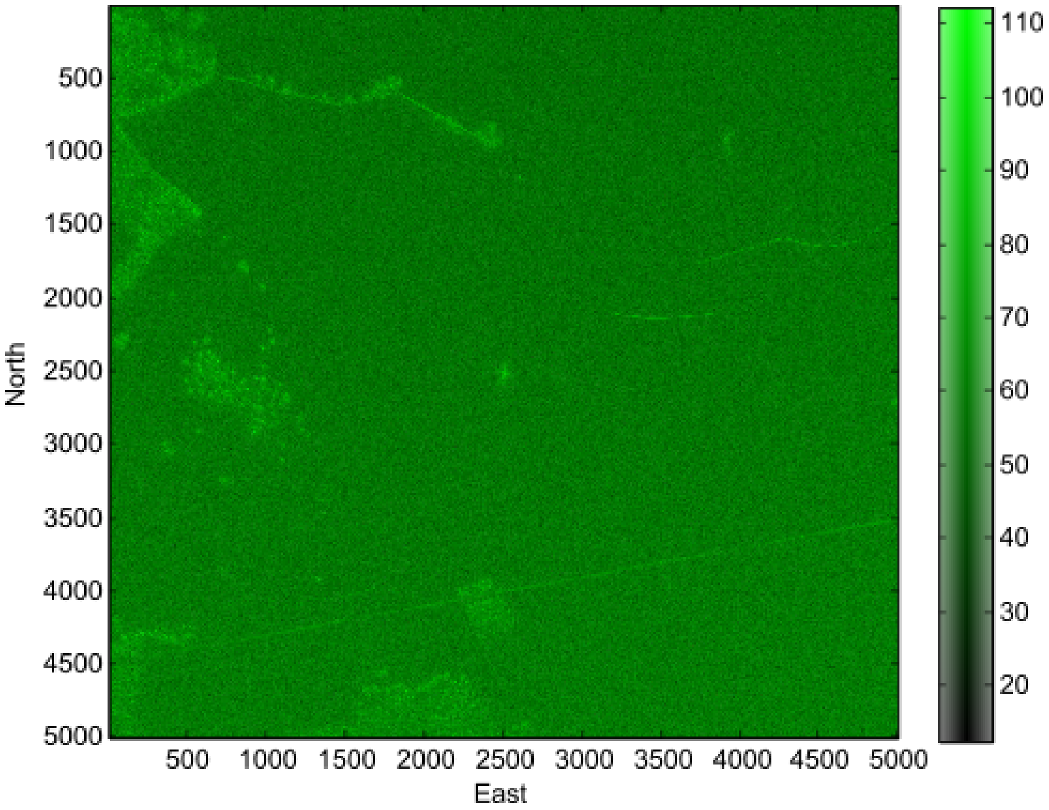


Figure 6.18: CARABAS image created using positioning data from RKTLIB.

In Figure 6.19 the area around the reflector has been enlarged. This image is made using the Waypoint output. In Figure 6.20 the same area can be seen where the positioning data comes from RTKLIB. Note the area to the right and left of the reflector where the output using RTKLIB improved upon that from Waypoint.

These images serve as a final proof that the accuracy of RTKLIB satisfies the requirements for the CARABAS system. Since the true positions are not known and the accuracy of the reference software is not known either one should keep in mind that the numbers in the tables presented in this report might not reflect the actual positions. However, as the final images are almost identical when using positions from RTKLIB and Waypoint we have shown that RTKLIB is accurate enough.

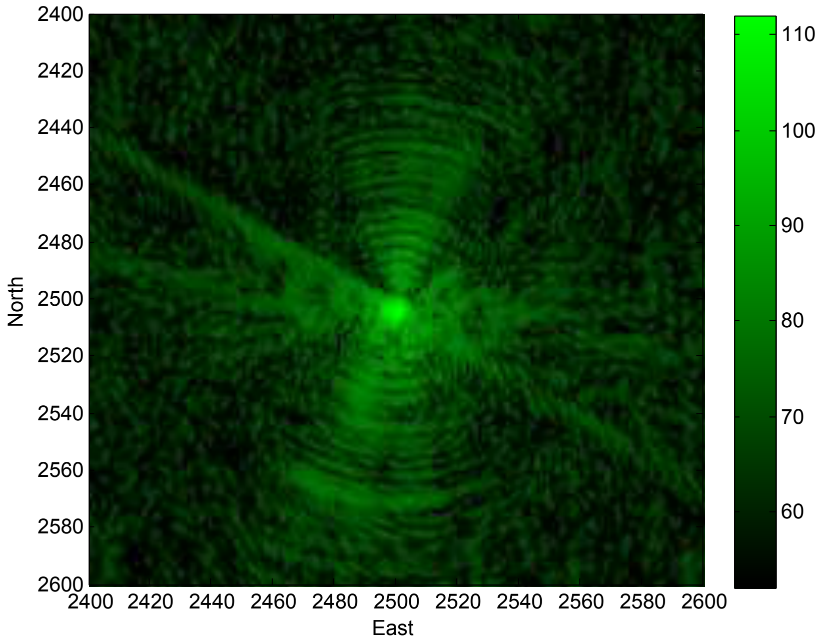


Figure 6.19: Closer look on the reflector in the CARABAS image created using positioning data from Waypoint.

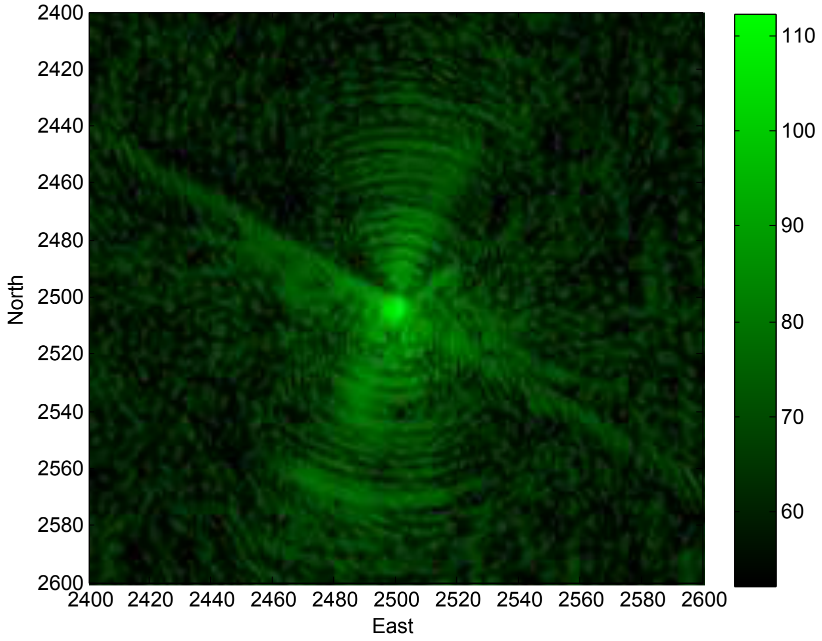


Figure 6.20: Closer look on the reflector in the CARABAS image created using positioning data from RTKLIB.

Chapter 7

Summary

One of the major objectives of this thesis was to create a batch-driven processing chain utilizing RTKLIB with minimum manual interaction. This was realized using the command-line interface of RTKLIB together with scripts written in C-shell and MATLAB-functions. The main script takes two arguments, the folder where to find raw rover data and the folder where to find the raw base station data. If all files are available the output from the script will be position files in a format that can be used by the rest of the processing chain for creating a SAR image. Once the program is started no further input is required which is a great improvement from using Waypoint Software where one operator had to actively sit in front of the program during the GPS processing which could take about one hour. The requirement for running the scripts is that all files are placed in the correct folders, and that SWEPOS data is available if needed. The machine on which the scripts are run also obviously need C-shell, MATLAB and RTKLIB. For more information see Appendix B

All results in this thesis where generated using the above mentioned scripts. A demonstration was made on the computer used in the field for creating CARABAS images. This demonstration proved that the scripts worked as intended, and showed the ease of installing RTKLIB as well as setting up everything needed.

The version 2.4.2 of RTKLIB was used in this thesis, and it was compiled without linking to LAPACK of Intel MKL libraries. By recompiling RTKLIB and linking to either library the time for processing GPS data will decrease if the computer used for the processing has more than one core. For time critical applications this provides a great and simple way speeding up the application.

For the flights studied in this thesis RTKLIB was generally on the same level of accuracy as the Waypoint software. There were however some drawbacks that have not yet been mentioned. In order to determine the base station position one must often use data from fixed reference stations. In Sweden it is possible to utilize the SWEPOS network which consists of about 300 fixed reference stations. Both softwares used are able to do this. However the processing options for GrafNav fits this type of positioning better than the counterpart in RTKLIB. Using GrafNav it is possible to provide data from multiple reference stations, something that is not possible in RTKLIB. Another drawback using RTKLIB for this kind of processing is that the data available from SWEPOS consists of RINEX files with data for one hour. There is no utility in RTKLIB for merging multiple RINEX observation files into one file. Because of this the processing of the base station position will have to be done one time for every hour that it collected data. If the base station started collecting data at 09:00 and stopped at 13:00 this would result in having to do the same processing four times in order to accurately determine the base station position. This means that the Kalman filter used for each processed hour has no information about the other states at the other hours. A simple remedy for this can be found. There exist freely available software for performing the merging of RINEX observation files, for example GPSTk [37] and TEQC [6]. Both GPSTk and TEQC could easily be implemented in the batch-driven processing chain created during this thesis for processing the GPS data using RTKLIB.

In the process of evaluating RTKLIB a bug was discovered when studying the baseline data. The bug caused the baselines to be several kilometers long, even though the baseline in reality was around 1.2 m. The creator of RTKLIB was then contacted and he found and fixed the bug a short time thereafter.

A good indication that the processing worked well is the fix ratio. For a good flight the ratio percentage is in the upper 90s. Adjusting the parameters in the configuration files will modify the behaviour of RTKLIB and therefore alter the ratio. For most flights the configuration files presented in Appendix C is sufficient. For certain flights where the fix ratio is lower one might need to adjust the values to better accommodate the environment at the measurements and the receivers used. This was used for the flight on October 6th in this thesis where there were cycle slips in some of the test runs. By modifying the parameters the cycle slips were repaired for some test runs. Though no perfect solution was found which fixed all cycle slips this could be used in the CARABAS system for improving the image quality of specific images.

Chapter 8

Conclusions and outlook

It has been shown that RKTLIB satisfies the accuracy requirements stated for creating high-resolution SAR images in the CARABAS system. The positioning accuracy was found to be within 5 cm and the moving baseline vectors within 1° when compared to reference data. SAR images were created using the output from RTKLIB for an absolute confirmation that the positioning accuracy was fulfilled. Using an image with a reflector at a known position it was shown that RTKLIB improved on the result with the compared Waypoint software. Assuming that the commercial software has an accuracy of 5 cm, RTKLIB should be on the same level or better.

A processing chain utilizing RTKLIB to create the position files used by the CARABAS system was created using C-shell and MATLAB, where the only required input from a user is information about where raw receiver data can be found on the computer where the processing will take place.

By studying how the quality flag was affected by the movements of the helicopter it was found that the speeds reached by the helicopter did not affect the integer ambiguity. Neither did the distances between the helicopter and the reference station, which for the provided flights always were less than 20 km.

Moving onwards there is still much that might be worth studying regarding the GPS processing for the CARABAS system. First and foremost the real-time performance of RTKLIB should be analyzed. This would be a natural step in the process of moving on from creating SAR images after a flight to create them while flying. Although no clear indications were found that the helicopter cabin affected one of the antennas more than the other it might be a good idea to study further. Modifying the filters used for updating positions in the single positioning mode of RTKLIB to use the characteristics of an aircraft would be a good start to see if it would be possible to eventually get rid of the reference station in the CARABAS system.

It should also be kept in mind that the future support of RTKLIB is not guaranteed to keep the same level at all times since it is a free software. For example if a bug is found it will probably only be fixed for the next release, which might be months away. The continuity aspect is also worth considering, it is always a risk that RTKLIB will not be maintained in 5-10 years. However, being open source makes it possible for users to fix bugs themselves, since modifications can be made as long as the conditions for the distribution conditions are met.

Bibliography

[1] Elisa Felicitas Arias. "Bureau International des Poids et Mesures (BIPM) - Time Department -". http://iag.dgfi.badw.de/fileadmin/IAG-docs/Travaux2013/08_BIPM.pdf, 2013. Accessed: 2015-05-11.

[2] Neil Ashby. Relativistic effects in the global positioning system. In *Gravitation and Relativity: At the Turn of the Millenium. 15th International Conference on General Relativity and Gravitation*, pages 231–258, 1997.

[3] X-W Chang, X Yang, and T Zhou. MLAMBDA: a modified LAMBDA method for integer least-squares estimation. *Journal of Geodesy*, 79(9):552–565, 2005.

[4] Peter H. Dana. "Earth Centered, Earth Fixed X,Y,Z". http://www.colorado.edu/geography/gcraft/notes/coordsys/gif/ecefxyz.gif, 1994. Accessed: 2015-05-12.

[5] Peter H. Dana. "Global Positioning System Overview". http://www.colorado.edu/geography/gcraft/notes/gps/gps.html, 1994. Accessed: 2015-03-17.

[6] Louis H Estey and Charles M Meertens. TEQC: the multi-purpose toolkit for GPS/-GLONASS data. *GPS solutions*, 3(1):42–49, 1999.

[7] Yanming Feng, Jinling Wang, et al. GPS RTK performance characteristics and analysis. *Positioning*, 1(13), 2008.

[8] United States Government. "GLOBAL POSITIONING SYSTEM WING (GPSW) SYSTEMS ENGINEERING & INTEGRATION". http://www.gps.gov/technical/icwg/IS-GPS-200E.pdf, June 2010. Revision E.

[9] United States Government. "Constellation Arrangement". http://www.gps.gov/systems/gps/space/, January 2015. Accessed: 2015-03-04.

[10] Werner Gurtner and Lou Estey. RINEX: The Receiver Independent Exchange Format Version 2.11, 2007.

[11] Elizabeth Howell. "Navstar: GPS Satellite Network". `http://www.space.com/19794-navstar.html`, February 2013. Accessed: 2015-05-07.

[12] Waypoint Consulting Inc. *GrafNav/GrafNet, GrafNav Lite, GrafMov, Inertial Explorer*, September 2004.

[13] "JAVAD GNSS, Inc". *Delta GNSS Receiver, Operator's Manual, version 1.4*, May 2009.

[14] Elliott D. Kaplan. *Understanding GPS: Principles and Applications*. Artech House, second edition, 2005.

[15] John A Klobuchar. Ionospheric time-delay algorithm for single-frequency gps users. *Aerospace and Electronic Systems, IEEE Transactions on*, (3):325–331, 1987.

[16] Richard B Langley. Time, clocks, and gps. *GPS World*, 2(10):38–42, 1991.

[17] Richard B Langley. Dilution of precision. *GPS world*, 10(5):52–59, 1999.

[18] Jhong Sam Lee and Leonard E. Miller. *CDMA Systems Engineering Handbook*. Artech House Publishers, 1998.

[19] FG Lemoine, DE Smith, L Kunz, R Smith, EC Pavlis, NK Pavlis, SM Klosko, DS Chinn, MH Torrence, RG Williamson, et al. The Development of the Joint NASA GSFC and NIMA Geopotential Model. In *Gravity, geoid and marine geodesy*, pages 461–469. Springer, 1997.

[20] Tech mp website. `http://www.www.tech-mp.com/vtol.sw.htm/`. Accessed: 2015-05-21.

[21] Navipedia. Ionospheric Delay — Navipedia, 2013. Accessed 2015-05-28.

[22] Navipedia. Real Time Kinematics — Navipedia, 2014. Accessed 2015-05-20.

[23] Scott Pace et al. *Global Positioning System: Assessing National Policies*. Rand Publishing, 1996.

[24] Benjamin W. Remondi. *Using the Global Positioning System (GPS) phase observable for relative geodesy: Modelling, processing, and results*. PhD thesis, University of Texas at Austin, Center for Space Research, Austin, TX, 1984.

[25] Anette RietDorf, Christopher Daub, and Peter Loef. Precise positioning in real-time using navigation satellites and telecommunication. In *Proceedings of The 3rd Workshop on Positioning and Communication (WPNC'06)*, 2006.

[26] J Saastamoinen. Atmospheric correction for the troposphere and stratosphere in radio ranging satellites. *The use of artificial satellites for geodesy*, pages 247–251, 1972.

[27] Gary Smith-Jonforsen. Carabas 3 generation 1 motion compensation performance evaluation. Unpublished, August 2013.

[28] Mehrdad Soumekh. *Synthetic aperture radar signal processing*. New York: Wiley, 1999.

[29] RTCM Standard. 10403.1 for Differential GNSS (Global Navigation Satellite Systems) Services-Version 3. *Radio Technical Commission for Matitime Services*, 2006.

[30] J. Sanz Subirana et al. "Time Reference in GNSS — Navipedia". `www.navipedia.net/index.php/Time_Reference_in_GNSS`, 2011. Accessed: 2015-05-11.

[31] Tomoji Takasu. "GNSS Precise Point Positioning with RTKLIB Part 1", 2011.

[32] Tomoji Takasu. *RTKLIB ver. 2.4.2 Manual*, April 2013.

[33] Tomoji Takasu and Akio Yasuda. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In *International Symposium on GPS/GNSS*, pages 4–6. International Convention Centre Jeju, Korea, 2009.

[34] Peter JG Teunissen. A new method for fast carrier phase ambiguity estimation. In *Position Location and Navigation Symposium, 1994., IEEE*, pages 562–573. IEEE, 1994.

[35] Lars MH Ulander, Hans Hellsten, and Gunnar Stenstrom. Synthetic-aperture radar processing using fast factorized back-projection. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(3):760–776, 2003.

[36] BIPM website. "BIPM work programme: Time". `http://www.bipm.org/en/bipm/tai`. Accessed: 2015-05-11.

[37] GPSTk website. `http://www.gpstk.org/`. Accessed: 2015-05-21.

[38] NASA website. "Global Positioning System History". `https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html#.VVyUHfmqpBc`, August 2013. Accessed: 2015-05-20.

[39] NASA website. "Global Positioning System". `https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS.html#.VUthGvmqpBd`, April 2014. Accessed: 2015-05-07.

[40] Saab website. "CARABAS". `saab.com/air/sensor-systems/ground-imaging-sensors/carabas`. Accessed: 2015-05-07.

[41] Guochang Xu. *GPS: Theory, Algorithms and Applications*. Springer, second edition, 2007.

Appendix A

Additional results

Here additional results from the comparison between RTKLIB and GrafNav/GrafMov will be presented.

A.1 Flight paths

A.1.1 2014-10-01

Table A.1: AFT1001a, $qRat_R$ = 0.987, $qRat_G$ = 0.930, nValues = 25689

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-4.9	17.1	17.8	-4.9	299.4
Y (mm)	-27.5	28.6	39.7	-27.4	614.4
Z (mm)	-1.3	65.7	65.7	-8.6	1931.0

Table A.2: AFT1001b, $qRat_R$ = 0.999, $qRat_G$ = 0.964, nValues = 3902

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	0.4	16.3	16.3	0.9	58.3
Y (mm)	14.7	18.6	23.7	13.6	84.4
Z (mm)	-27.2	31.6	41.7	-23.7	146.6

Note on A.2 the output from RTKLIB is split up into b and c for the corresponding output b from GrafNav. This is because GrafNav merges two observation files into one, whereas RTKLIB creates one solution for each file.

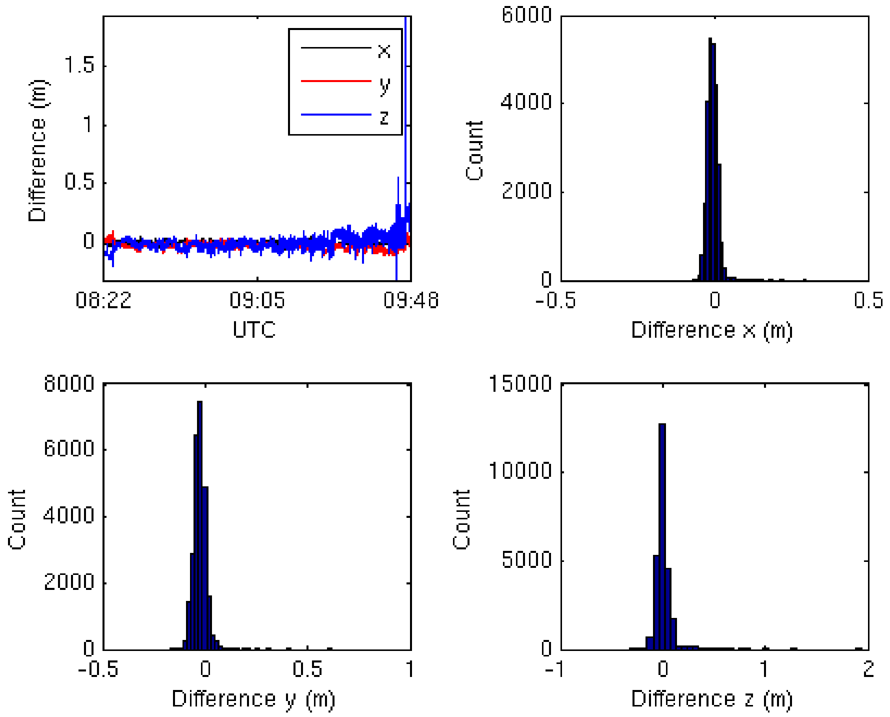


Figure A.1: Plot and histogram displaying differences in x-, y- and z-directions for AFT1001a.

Table A.3: AFT1001b (the rest), $qRat_R$ = 0.763, $qRat_G$ = 0.831, nValues = 10571

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	31.2	87.7	93.1	24.5	443.8
Y (mm)	39.9	67.3	78.2	36.0	448.4
Z (mm)	-53.2	153.7	162.6	-44.8	806.9

Table A.4: FWD1001a, $qRat_R$ = 0.897, $qRat_G$ = 0.765, nValues = 25466

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	12.3	43.3	45.1	4.6	422.5
Y (mm)	-40.4	44.3	59.9	-34.6	231.8
Z (mm)	44.5	103.2	112.4	11.6	1076.0

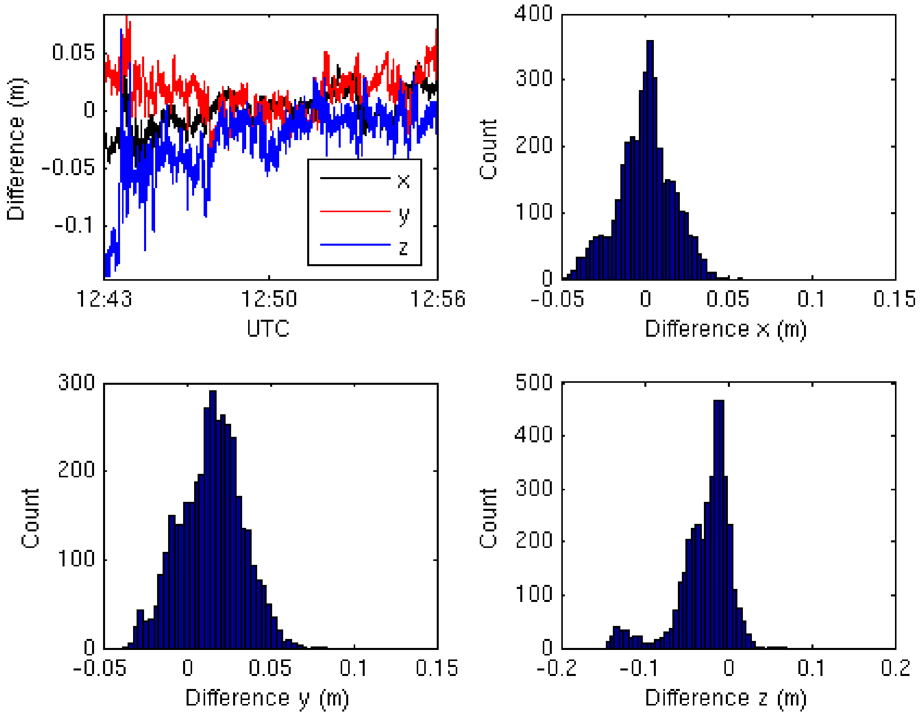


Figure A.2: Plot and histogram displaying differences in x-, y- and z-directions for AFT1001b.

Table A.5: FWD1001b, $qRat_R$ = 0.995, $qRat_G$ = 0.888, nValues = 3859

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-0.8	18.8	18.8	-1.1	52.9
Y (mm)	13.8	23.3	27.0	12.6	65.4
Z (mm)	-26.1	20.2	32.9	-25.5	97.2

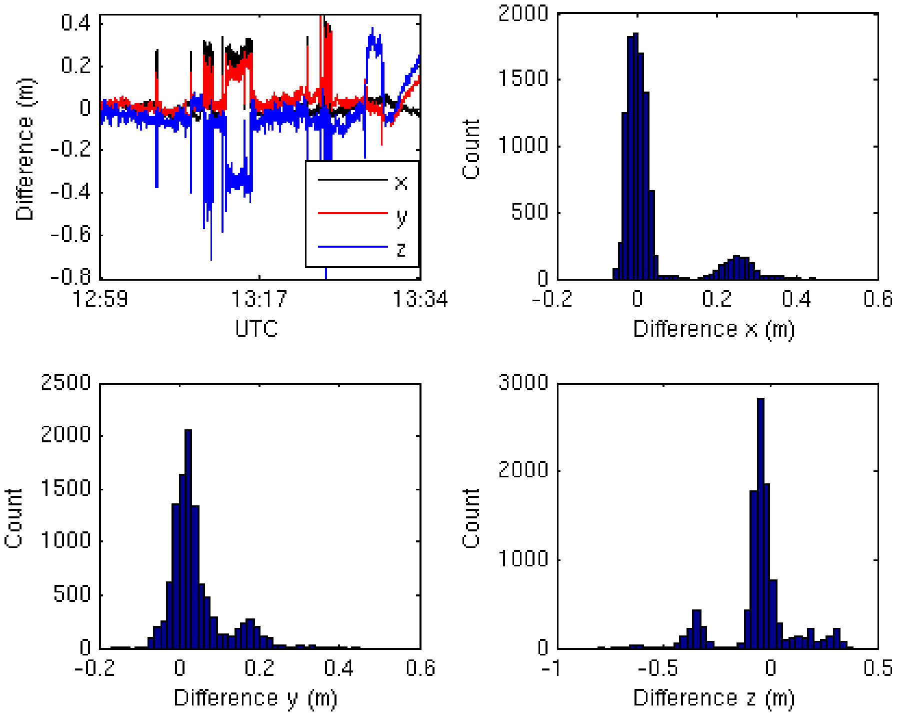


Figure A.3: Plot and histogram displaying differences in x-, y- and z-directions for AFT1001b.

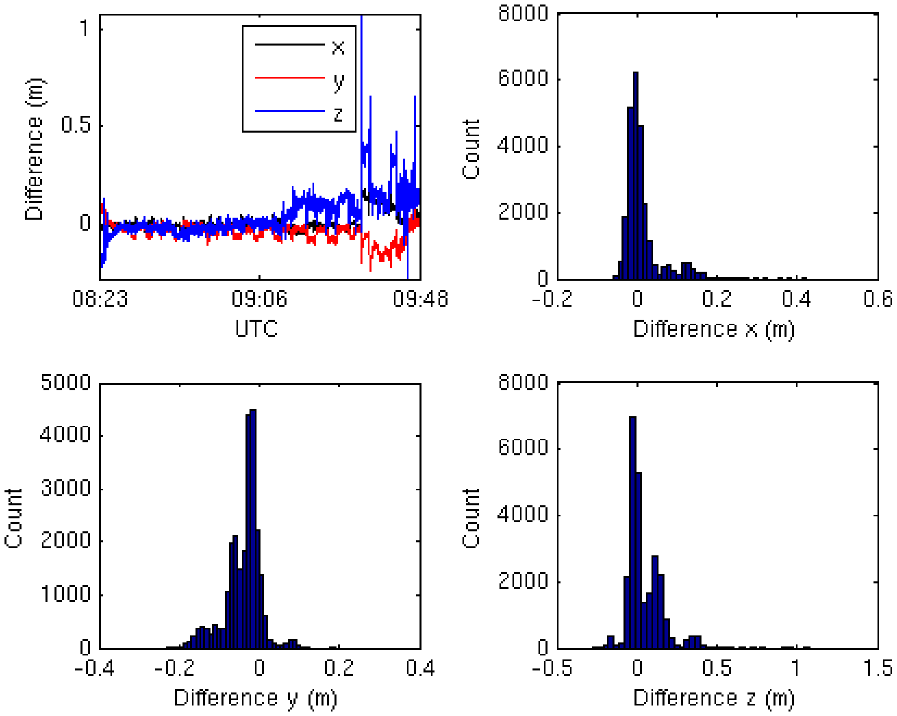


Figure A.4: Plot and histogram displaying differences in x-, y- and z-directions for FWD1001a.

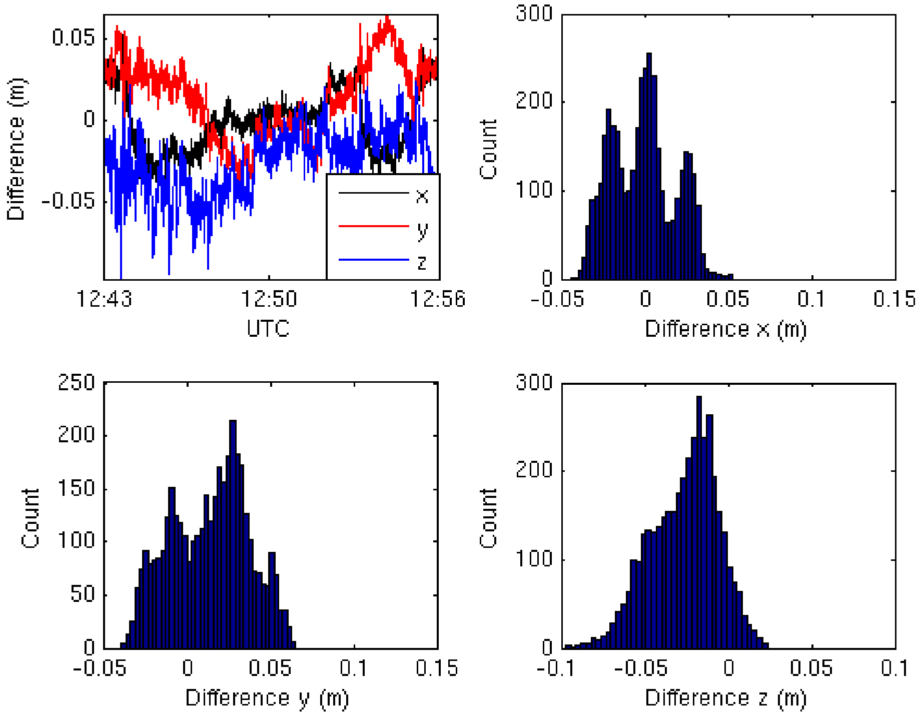


Figure A.5: Plot and histogram displaying differences in x-, y- and z-directions for FWD1001b.

A.1.2 2014-10-03

Table A.6: AFT1003a, $qRat_R$ = 0.660, $qRat_G$ = 0.940, nValues = 20667

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	13.8	24.6	28.2	14.3	118.7
Y (mm)	-15.9	35.4	38.8	-16.0	200.5
Z (mm)	-87.7	52.9	102.4	-85.2	298.0

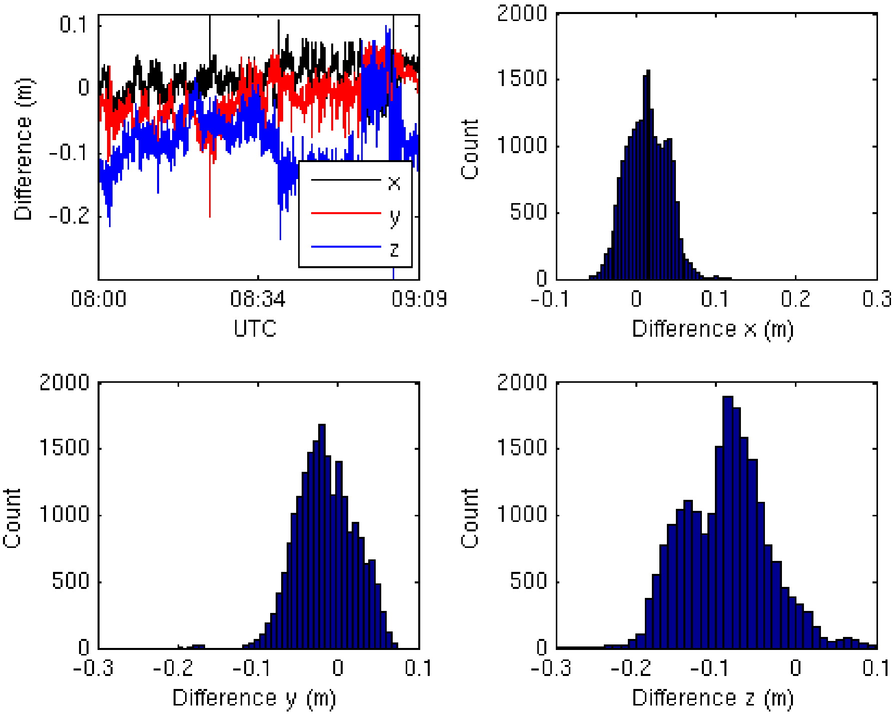


Figure A.6: Plot and histogram displaying differences in x-, y- and z-directions for AFT1003a.

Table A.7: AFT1003b, $qRat_R$ = 0.961, $qRat_G$ = 0.925, nValues = 19976

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	6.4	140.6	14.1	1.8	4432.7
Y (mm)	8.1	64.1	6.5	8.5	1901.8
Z (mm)	4.5	703.9	70.4	-15.9	22275.3

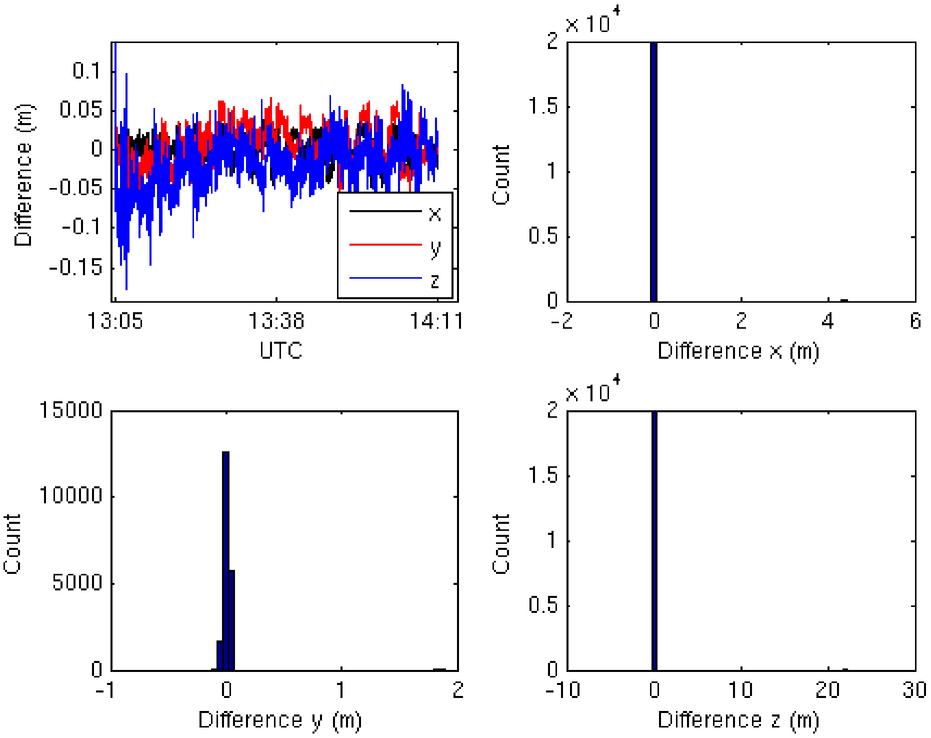


Figure A.7: Plot and histogram displaying differences in x-, y- and z-directions for AFT1003b.

Table A.8: FWD1003a, $qRat_R$ = 0.643, $qRat_G$ = 0.965, nValues = 20126

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	24.3	39.7	46.5	25.4	143.2
Y (mm)	-24.5	64.6	69.1	-24.9	258.9
Z (mm)	-45.5	133.7	141.2	-57.7	387.6

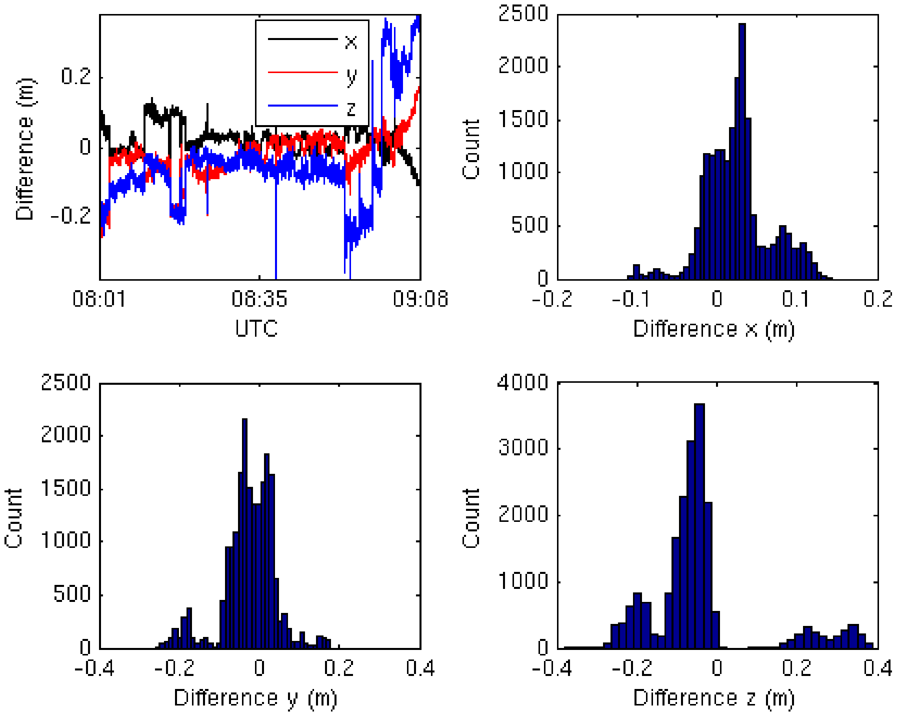


Figure A.8: Plot and histogram displaying differences in x-, y- and z-directions for FWD1003a.

A.1.3 2014-10-06

Table A.9: AFT1006c, $qRat_R$ = 0.513, $qRat_G$ = 0.852, nValues = 13746

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	41.2	55.0	68.7	35.1	161.8
Y (mm)	12.4	54.9	56.2	17.8	689.0
Z (mm)	229.0	372.5	437.3	180.2	1166.1

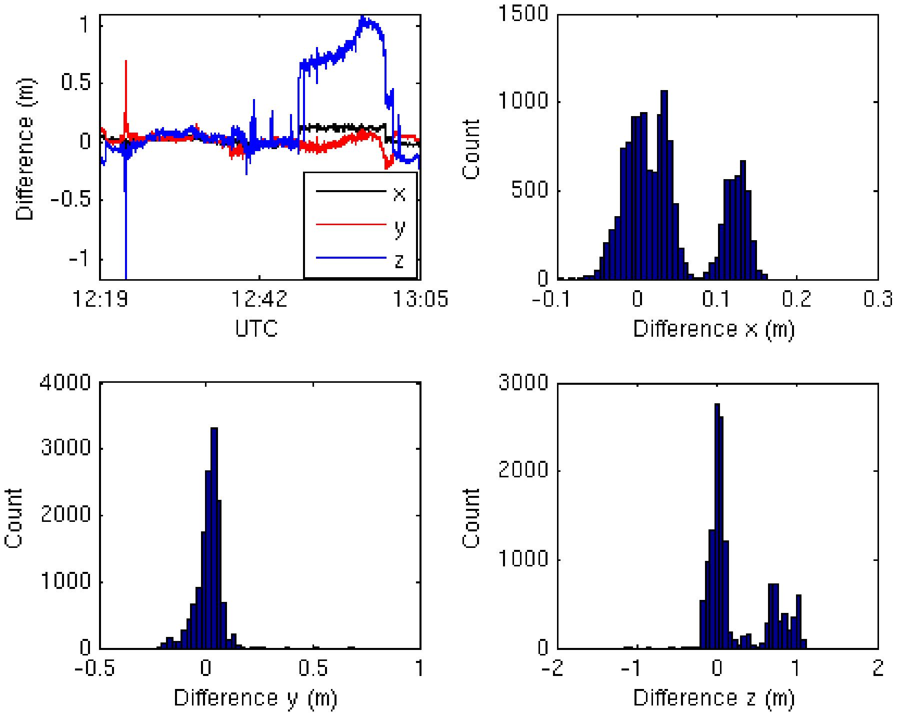


Figure A.9: Plot and histogram displaying differences in x-, y- and z-directions for AFT1006c.

A.1.4 2014-10-07

Table A.10: AFT1007a, $qRat_R$ = 0.533, $qRat_G$ = 0.923, nValues = 19867

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-121.4	176.0	213.8	-106.6	642.2
Y (mm)	-106.3	176.5	206.1	-85.8	963.8
Z (mm)	-66.4	442.6	447.6	24.9	2478.7

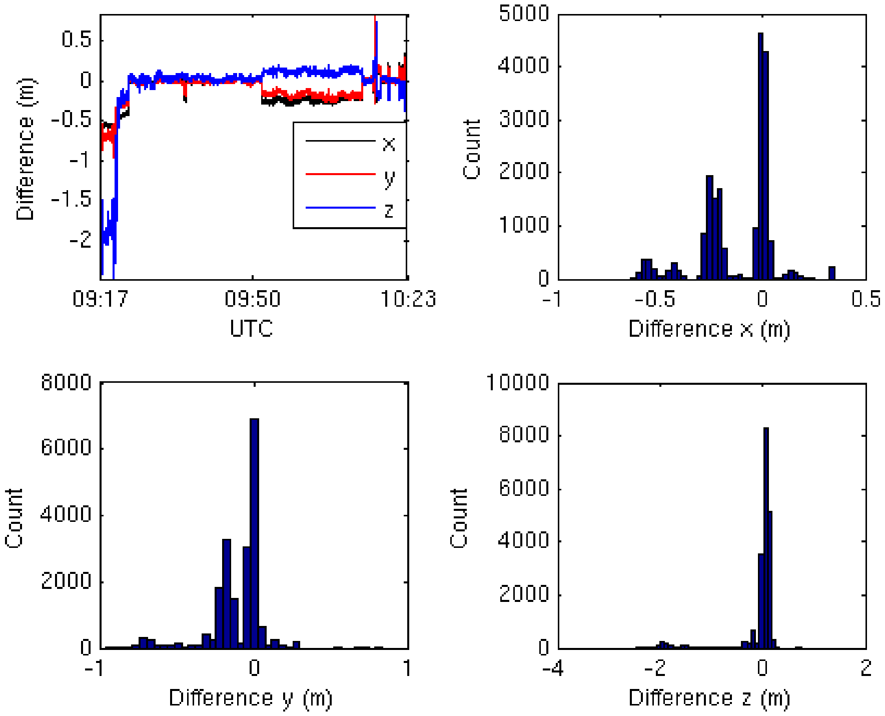


Figure A.10: Plot and histogram displaying differences in x-, y- and z-directions for AFT1007a.

Table A.11: AFT1007b, $qRat_R$ = 0.139, $qRat_G$ = 0.944, nValues = 17808

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	26.9	41.2	49.2	28.8	211.0
Y (mm)	150.1	72.5	166.7	153.2	256.3
Z (mm)	-347.9	145.7	377.2	-349.1	532.6

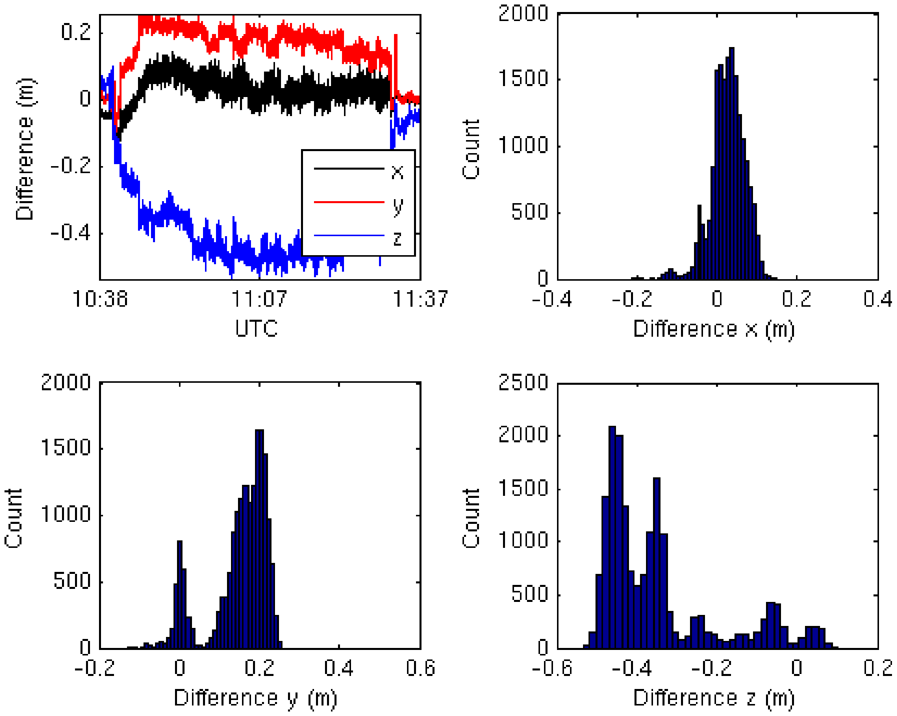


Figure A.11: Plot and histogram displaying differences in x-, y- and z-directions for AFT1007b.

Table A.12: FWD1007a, $qRat_R$ = 0.960, $qRat_G$ = 0.780, nValues = 19728

Variable	Mean diff	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-30.4	30.9	43.4	-33.0	343.5
Y (mm)	-75.8	63.6	99.0	-75.1	3496.2
Z (mm)	-108.3	93.4	143.0	-96.1	5770.1

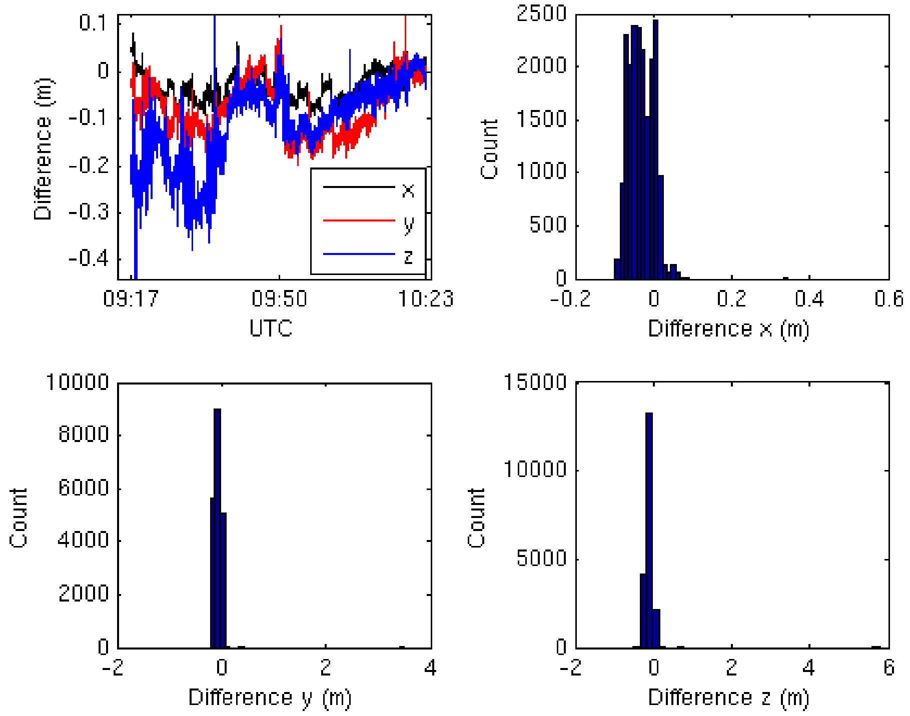


Figure A.12: Plot and histogram displaying differences in x-, y- and z-directions for FWD1007a.

Table A.13: FWD1007b, $qRat_R$ = 0.973, $qRat_G$ = 0.833, nValues = 17306

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	3.1	20.4	20.7	4.1	79.2
Y (mm)	-23.8	23.3	33.3	-24.5	90.3
Z (mm)	-54.2	26.9	60.5	-54.0	195.8

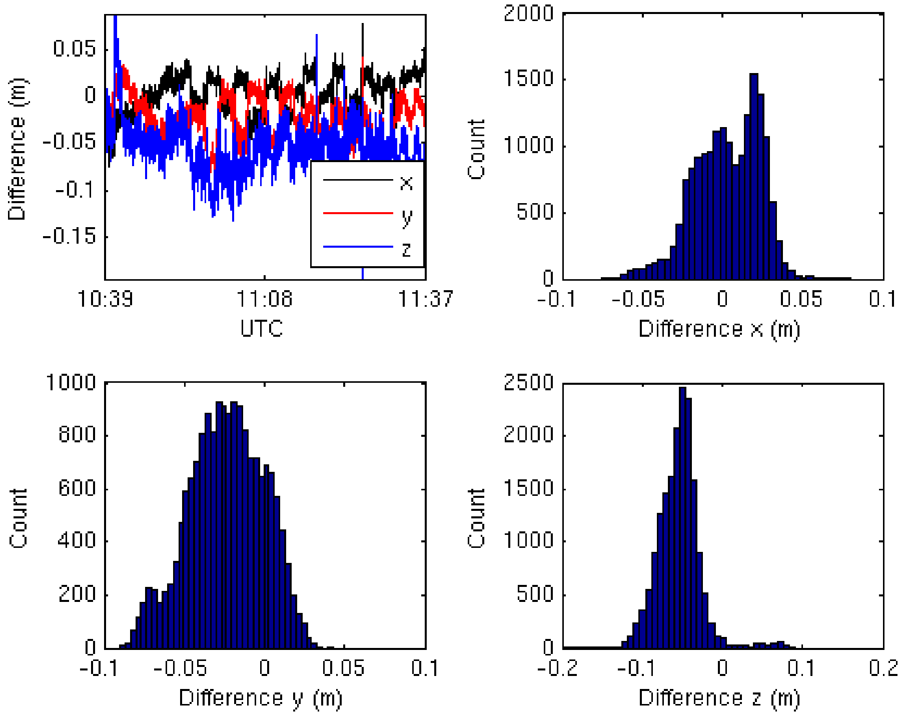


Figure A.13: Plot and histogram displaying differences in x-, y- and z-directions for FWD1007b.

A.2 Individual test runs, flight paths

A.2.1 2014-10-01, AFT1001a

Table A.14: 16_E0904212, $qRat_R = 1.00$, $qRat_G = 0.908$, nValues = 173, h = 250 m, l = 1015 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	2.8	2.6	3.8	2.8	11.4
Y (mm)	-30.9	7.7	31.9	-30.9	54.7
Z (mm)	4.5	16.8	17.4	4.4	41.4

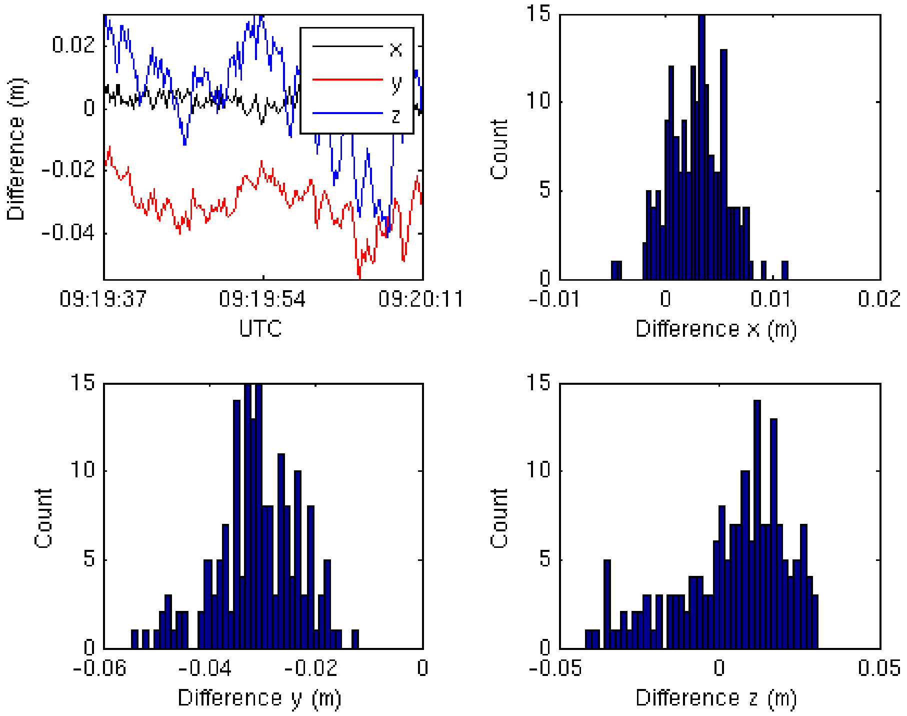


Figure A.14: Plot and histogram displaying differences in x-, y- and z-directions for AFT1001a test run 16 E0904212.

Table A.15: 17_E0904213, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 216, h = 250 m, l = 1013 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	10.7	8.9	13.9	10.7	31.3
Y (mm)	-43.7	8.2	44.5	-43.7	60.2
Z (mm)	-3.3	27.1	27.3	-3.2	63.6

Table A.16: 18_E0904214, $qRat_R$ = 1.00, $qRat_G$ = 0.394, nValues = 160, h = 250 m, l = 973 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	11.5	3.5	12.0	11.5	23.5
Y (mm)	-58.5	13.1	59.9	-58.5	85.7
Z (mm)	15.6	38.8	41.8	15.6	91.0

Table A.17: 19_E0904215, $qRat_R$ = 1.00, $qRat_G$ = 0.963, nValues = 215, h = 250 m, l = 1038 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-26.3	8.8	27.8	-2.6	46.8
Y (mm)	-30.8	5.6	31.3	-3.1	44.5
Z (mm)	76.6	17.8	78.6	76.2	118.4

Table A.18: 26_E0904218, $qRat_R$ = 1.00, $qRat_G$ = 0.965, nValues = 57, h = 30 m, l = 353 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	1.9	4.6	5.0	1.9	15.5
Y (mm)	-60.9	15.0	62.7	-60.9	87.8
Z (mm)	38.0	34.3	51.2	38.0	83.5

Table A.19: 27_E0904219, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 103, h = 30 m, l = 469 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-28.6	5.8	29.2	-28.6	38.8
Y (mm)	-21.5	7.4	22.7	-21.5	38.8
Z (mm)	102.1	16.7	103.4	102.1	126.4

Table A.20: 28_E0904220, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 57, h = 30 m, l = 323 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	1.5	3.5	3.8	1.5	8.1
Y (mm)	-73.5	7.4	73.9	-73.5	90.0
Z (mm)	72.7	14.8	74.2	72.7 94.6	

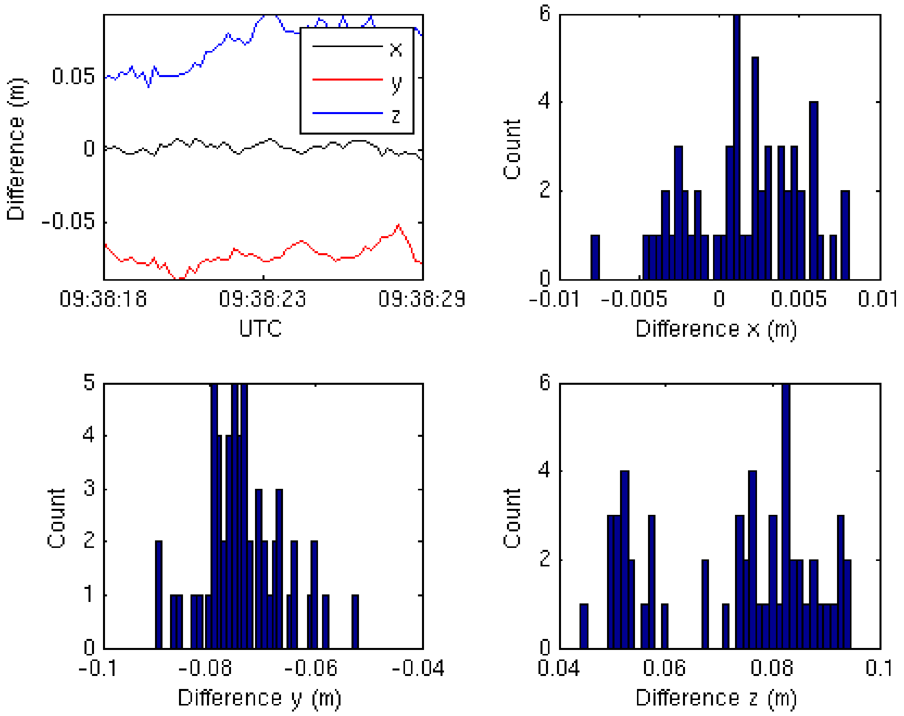


Figure A.15: Plot and histogram displaying differences in x-, y- and z-directions for AFT1001a test run 28 E0904220.

Table A.21: 29_E0904221, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 76, h = 30 m, l = 368 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-5.7	6.1	8.4	-5.7	16.0
Y (mm)	-56.2	7.3	56.7	-56.2	74.0
Z (mm)	67.8	15.9	69.7	67.9	98.3

A.2.2 2014-10-06, FWD1006c

Table A.22: 10_E0914220, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 40, h = 30 m, l = 242 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	13.6	2.9	13.9	13.6	19.7
Y (mm)	4.6	4.8	6.7	4.6	14.9
Z (mm)	-11.5	10.2	15.4	-11.5	30.8

Table A.23: 11_E0914221, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 51, h = 30 m, l = 240 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-14.1	4.5	14.8	-14.1	24.7
Y (mm)	23.9	6.3	24.7	23.9	37.1
Z (mm)	-17.5	9.6	20.0	-17.5	39.0

Table A.24: 12_E0914222, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 46, h = 30 m, l = 239 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	11.6	3.7	12.2	11.6	17.9
Y (mm)	-16.4	5.2	17.2	-16.4	25.6
Z (mm)	-38.4	10.8	39.9	-38.4	57.9

Table A.25: 13_E0914223, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 51, h = 30 m, l = 238 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-10.4	3.5	11.0	-10.4	19.7
Y (mm)	13.7	5.4	14.7	13.7	23.2
Z (mm)	-45.7	14.3	47.9	-45.7	67.

Table A.26: 14_E0914218, $qRat_R = 1.00$, $qRat_G = 0.674$, nValues = 43, h = 30 m, l = 239 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	11.7	4.0	12.4	11.7	18.6
Y (mm)	-5.8	4.1	7.1	-5.8	14.1
Z (mm)	-39.1	7.8	39.9	-39.1	51.7

Table A.27: 15_E0914219, $qRat_R = 1.00$, $qRat_G = 1.00$, nValues = 51, h = 30 m, l = 235 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-11.3	3.6	11.9	-11.3	20.6
Y (mm)	10.8	7.0	12.9	10.8	23.6
Z (mm)	-38.5	7.7	39.2	-38.5	54.1

Table A.28: 8_E0914218, $qRat_R = 0.00$, $qRat_G = 1.00$, nValues = 40, h = 30 m, l = 249 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-108.8	7.0	109.1	-108.8	123.9
Y (mm)	129.1	4.9	129.2	129.1	139.1
Z (mm)	384.1	9.9	384.2	384.1	413.8

Table A.29: 10_E0915505, $qRat_R = 1.00$, $qRat_G = 1.00$, nValues = 248, h = 250 m, l = 1781 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	27.1	4.2	27.5	27.1	37.8
Y (mm)	25.6	7.3	26.6	25.6	43.4
Z (mm)	-56.0	14.0	57.7	-56.0	87.8

Table A.30: 11_E0915506, $qRat_R = 0.949$, $qRat_G = 1.00$, nValues = 276, h = 250 m, l = 1576 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-11.6	26.8	29.2	-9.5	129.5
Y (mm)	-21.7	46.3	51.1	-25.7	220.2
Z (mm)	-31.9	64.7	72.1	-37.9	290.2

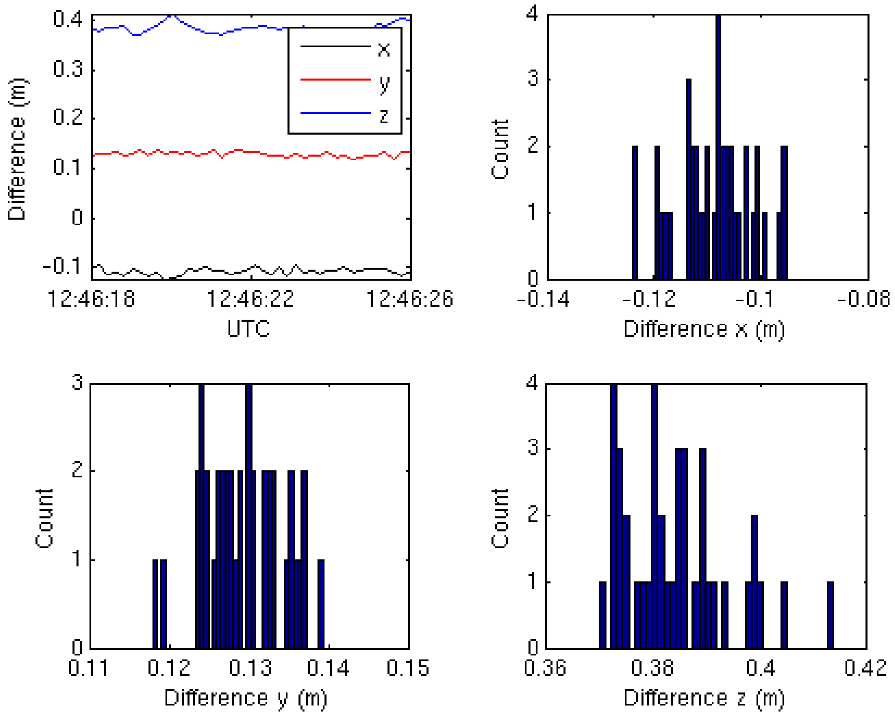


Figure A.16: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 8 E0914218.

Table A.31: 12_E0915507, $qRat_R$ = 0.00, $qRat_G$ = 0.984, nValues = 252, h = 250 m, l = 841 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-101.2	17.8	102.7	-100.6	140.2
Y (mm)	178.4	24.9	180.1	179.6	221.0
Z (mm)	336.1	14.0	336.4	335.7	367.6

Table A.32: 14_E0915509, $qRat_R$ = 0.00, $qRat_G$ = 0.981, nValues = 257, h = 250 m, l = 132 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-117.5	14.1	118.3	-117.9	155.8
Y (mm)	147.0	17.9	148.1	146.0	208.3
Z (mm)	359.2	27.1	360.2	358.2	431.8

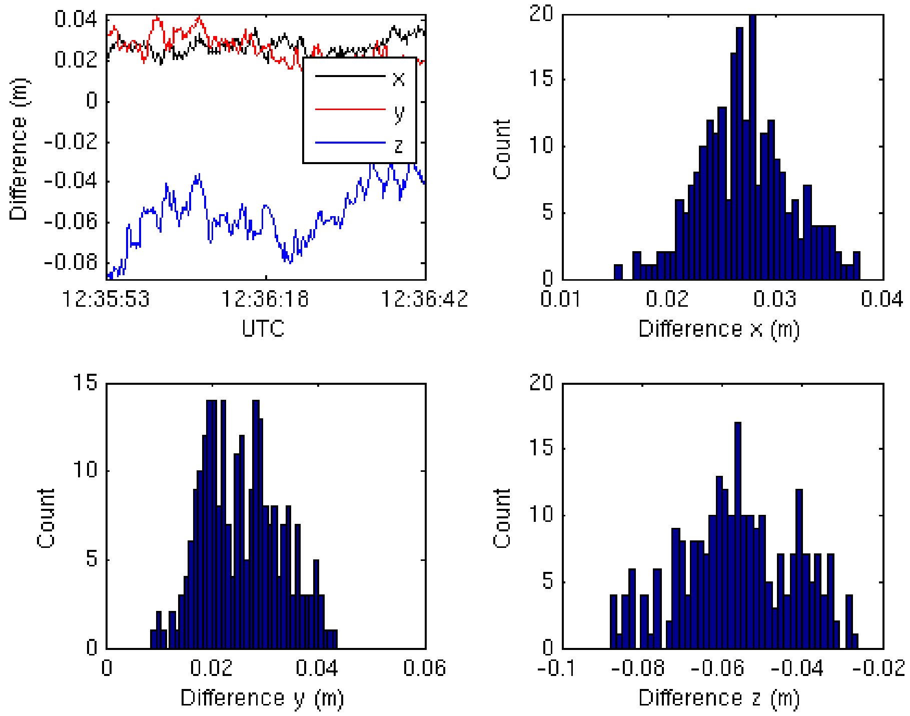


Figure A.17: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 10 E0915505.

Table A.33: 15_E0915510, $qRat_R$ = 0.371, $qRat_G$ = 1.00, nValues = 245, h = 250 m, l = 394 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	-61.2	56.0	83.0	-61.3	127.4
Y (mm)	57.6	43.0	71.9	58.1	114.6
Z (mm)	171.6	140.0	221.5	174.0	309.8

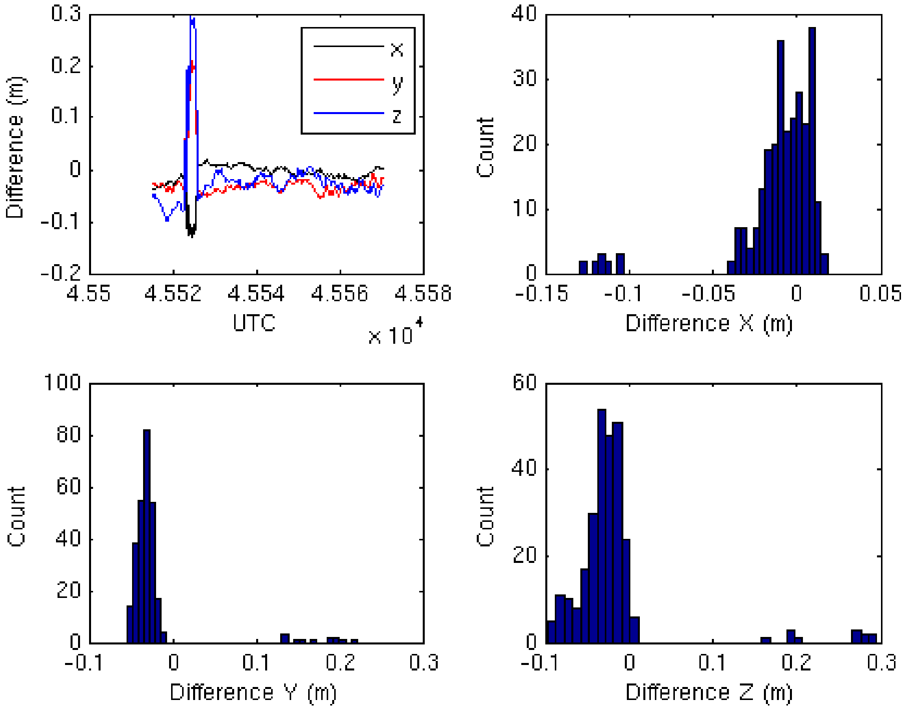


Figure A.18: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 11 E0915506.

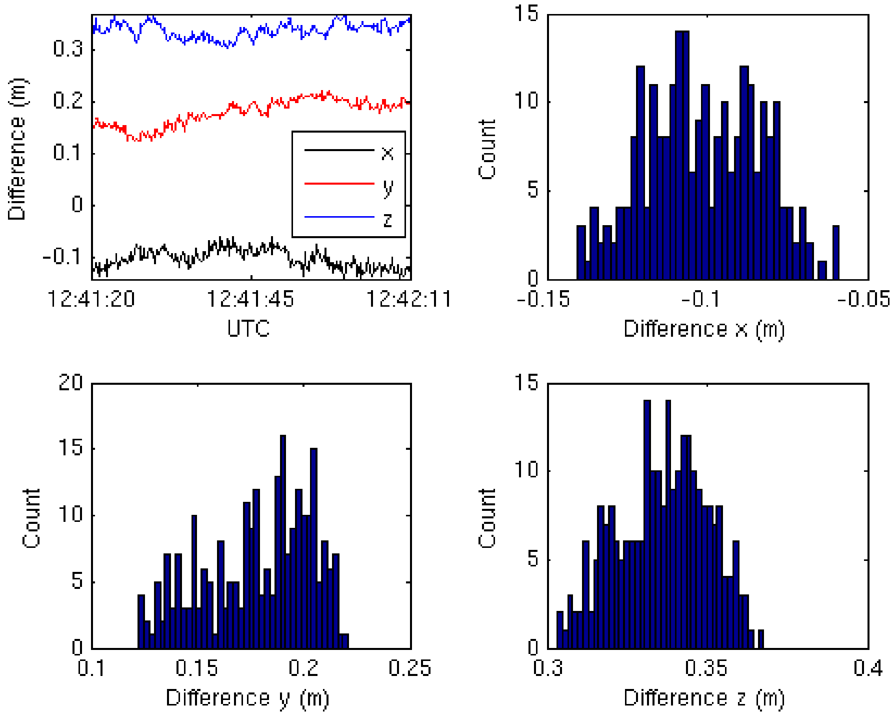


Figure A.19: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 12 E095507.

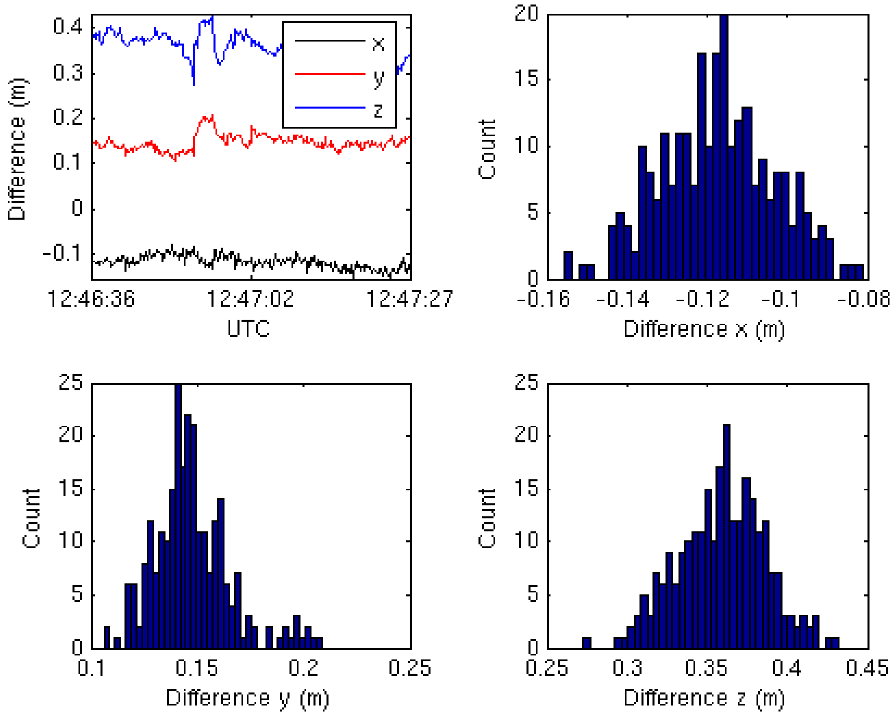


Figure A.20: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 14 E0915509.

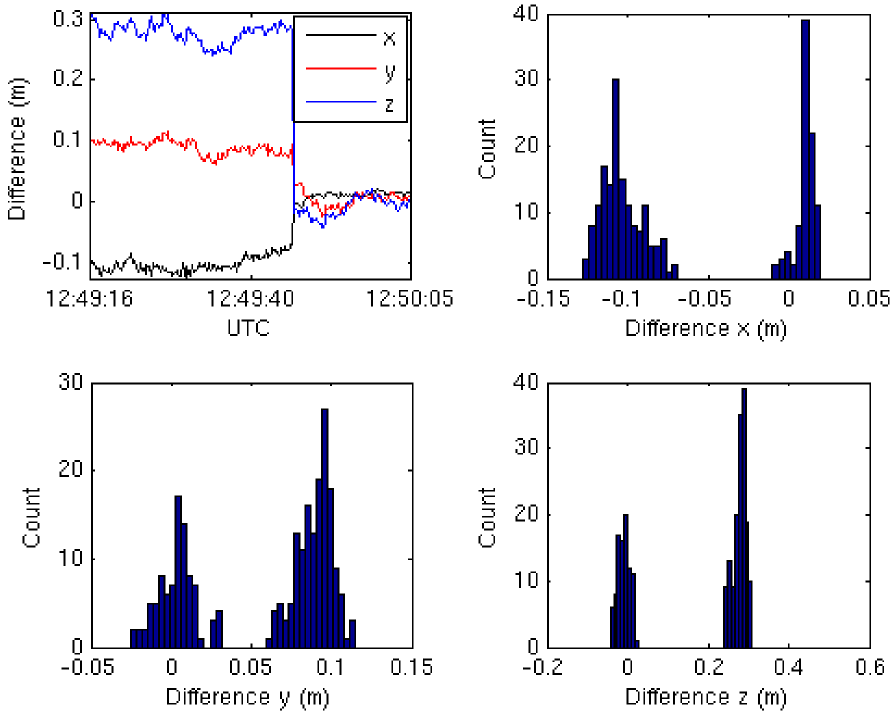


Figure A.21: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 15 E0915510.

Table A.34: 16_E0915511, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 256, h = 250 m, l = 1065 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	4.6	12.2	13.0	4.6	28.3
Y (mm)	-13.2	9.9	16.6	-13.2	43.2
Z (mm)	-35.5	17.9	39.7	-35.5	88.8

Table A.35: 17_E0915512, $qRat_R$ = 0.971, $qRat_G$ = 0.543, nValues = 280, h = 250 m, l = 1377 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	8.4	15.0	17.2	8.0	87.0
Y (mm)	-11.6	18.9	22.2	-11.5	119.2
Z (mm)	-49.9	47.7	69.0	-47.6	345.4

Table A.36: 8_E0915503, $qRat_R$ = 1.00, $qRat_G$ = 0.992, nValues = 262, h = 250 m, l = 1483 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	18.9	4.6	19.5	18.9	28.6
Y (mm)	48.7	10.1	49.7	48.7	77.4
Z (mm)	-43.2	13.0	45.1	-43.2	72.7

Table A.37: 9_E0915504, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 284, h = 250 m, l = 1931 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	W.Mean diff.	Max diff.
X (mm)	36.0	9.1	37.1	36.0	49.8
Y (mm)	53.0	7.5	53.5	53.0	71.3
Z (mm)	-13.9	7.9	16.0	-13.9	29.5

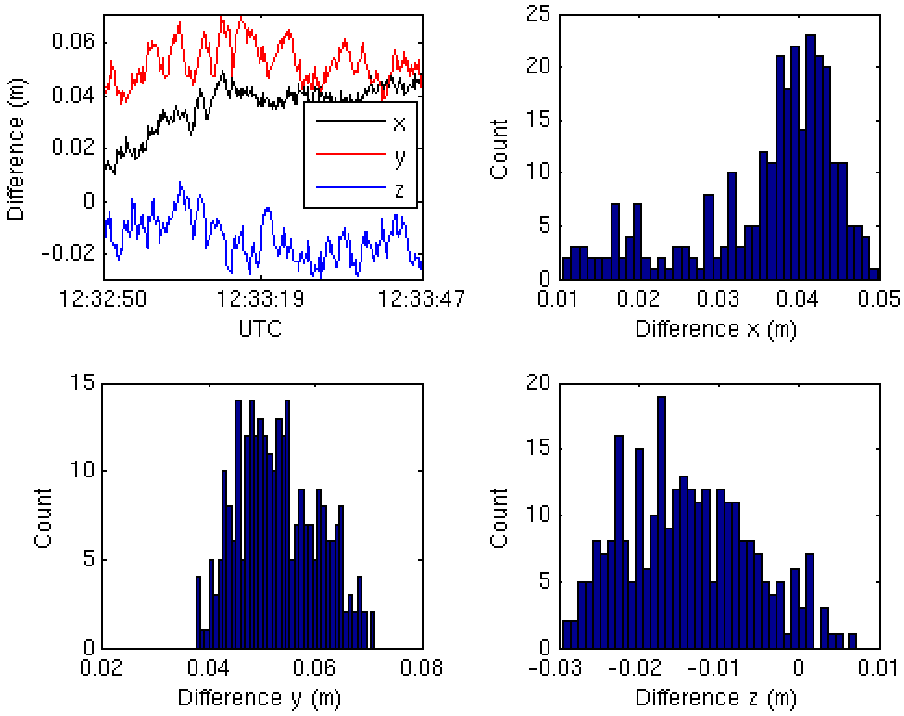


Figure A.22: Plot and histogram displaying differences in x-, y- and z-directions for FWD1006c test run 9 E0915504.

A.3 Orientation

A.3.1 2014-10-01

Table A.38: Baseline1001a, $qRat_R$ = 0.999, $qRat_G$ = 0.995, nValues = 49546

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-0.0	7.0	7.0	157.5
Y (mm)	1.0	10.6	10.6	130.4
Z (mm)	1.7	29.7	29.8	388.3
Az (°)	-0.053	0.453	10.456	5.444
θ (°)	0.782	1.280	1.500	16.032

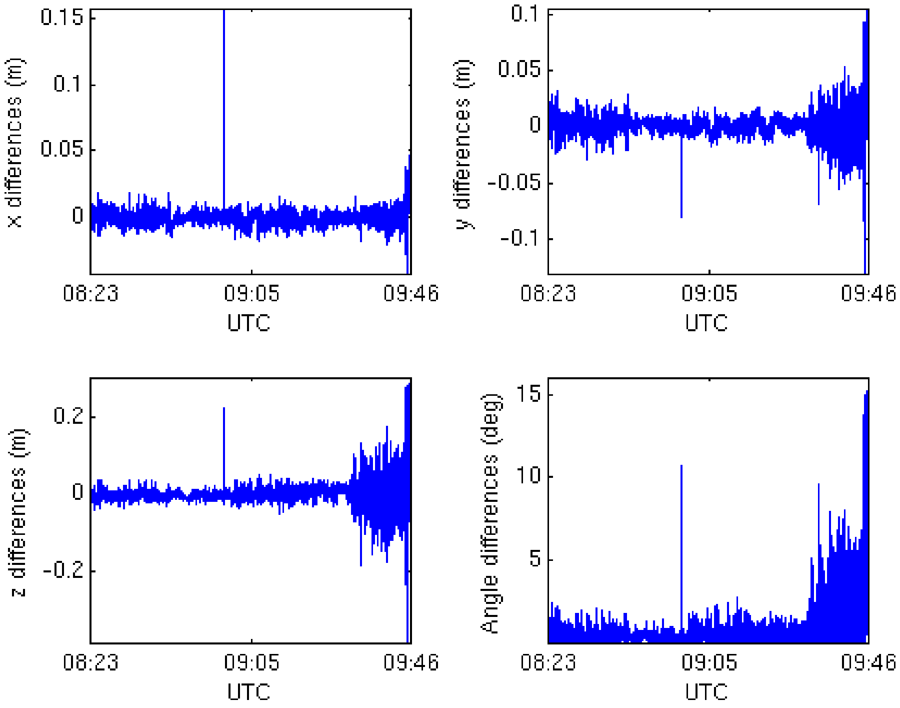


Figure A.23: Plot displaying differences in x-, y-, z-directions and angles for 1001a.

Table A.39: Baseline1001b, $qRat_R$ = 0.993, $qRat_G$ = 0.997, nValues = 7715

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	1.0	13.7	13.7	167.3
Y (mm)	-0.0	11.2	11.2	141.4
Z (mm)	6.4	59.2	59.5	743.2
Az (°)	-0.010	0.666	0.666	10.056
θ (°)	0.907	2.969	3.105	38.641

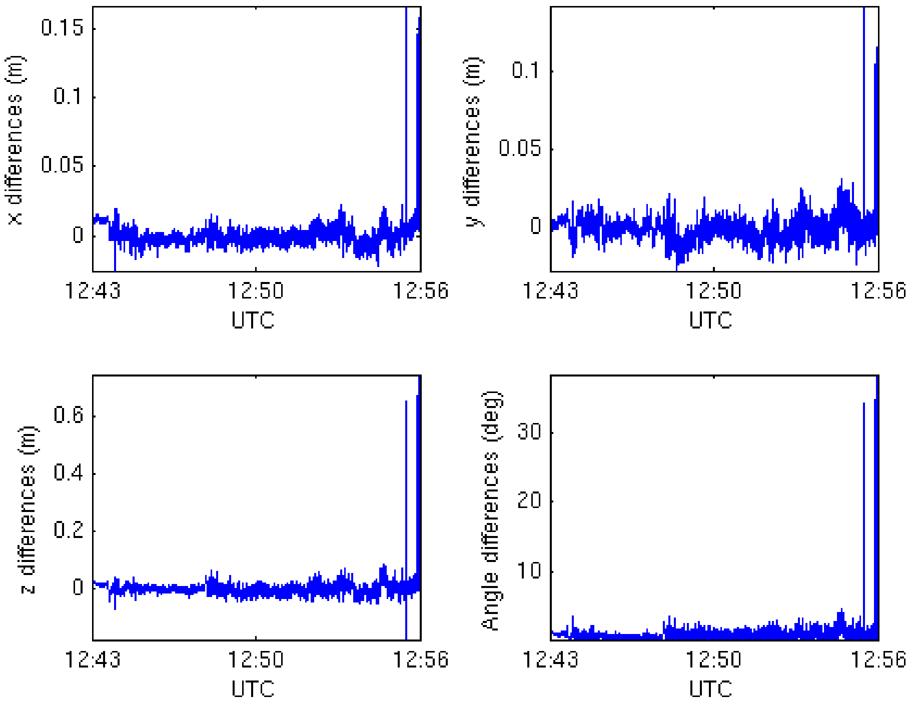


Figure A.24: Plot displaying differences in x-, y-, z-directions and angles for 1001b.

A.3.2 2014-10-03

Table A.40: Baseline1003a, $qRat_R$ = 0.978, $qRat_G$ = 0.992, nValues = 36660

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	2.3	21.6	21.7	405.0
Y (mm)	-0.1	21.9	21.9	313.4
Z (mm)	4.0	35.3	35.5	394.2
Az (°)	-0.003	1.370	1.370	24.071
θ (°)	0.779	2.047	2.190	25.377

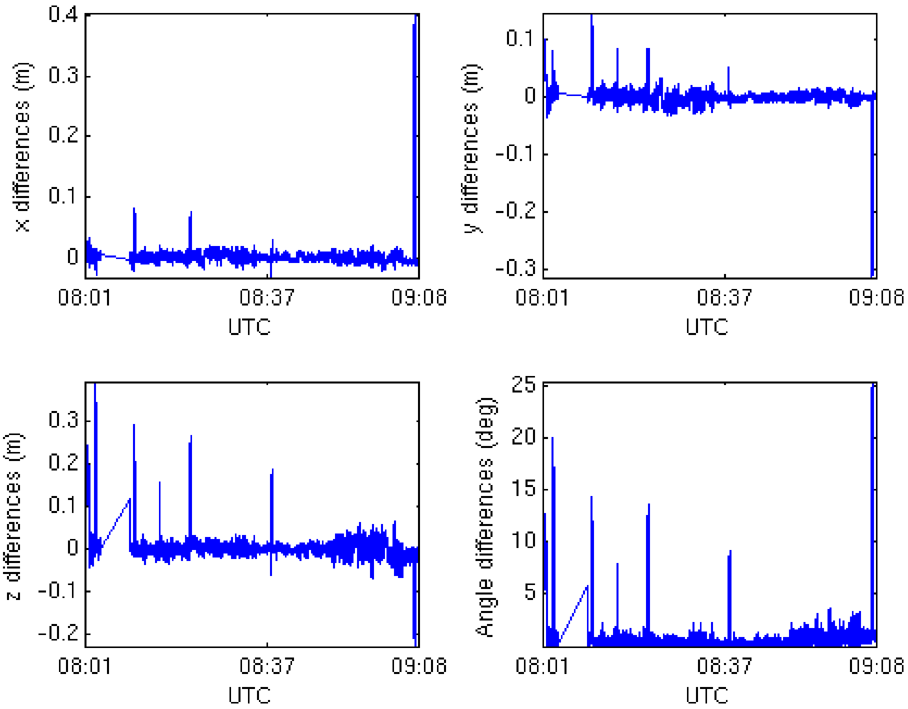


Figure A.25: Plot displaying differences in x-, y-, z-directions and angles for for 1003a.

A.3.3 2014-10-07

Table A.41: Baseline1007a, $qRat_R$ = 0.937, $qRat_G$ = 0.991, nValues = 27591

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	3.3	35.9	36.0	402.2
Y (mm)	2.9	50.1	50.2	977.6
Z (mm)	32.1	199.5	202.0	4262.1
Az (°)	0.311	2.117	2.140	61.178
θ (°)	2.400	8.504	8.837	100.115

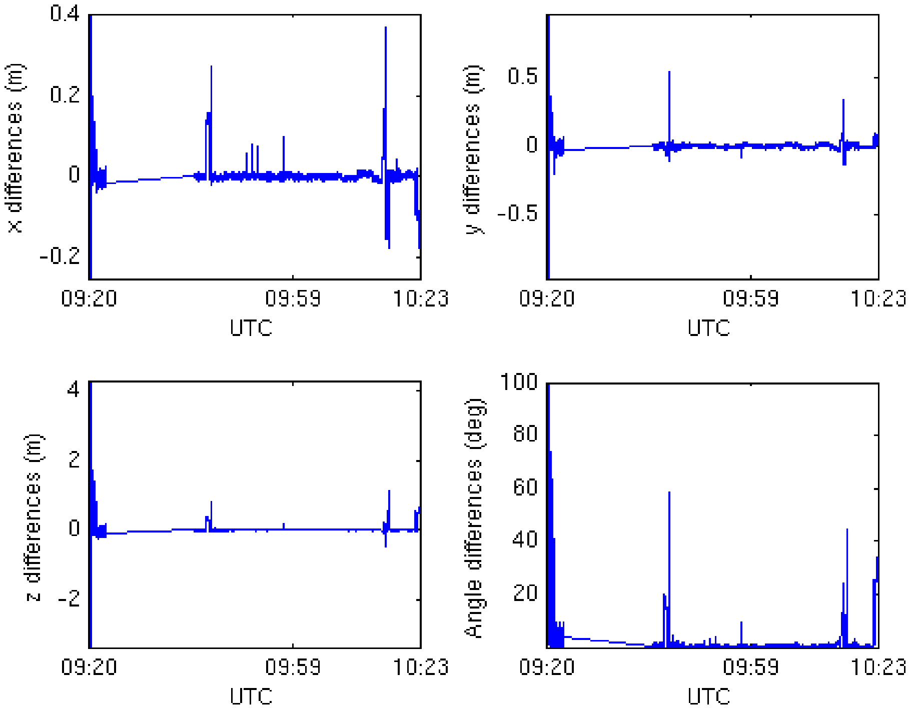


Figure A.26: Plot displaying differences in x-, y-, z-directions and angles for 1007a.

Table A.42: Baseline1007b, $qRat_R$ = 1.00, $qRat_G$ = 0.973, nValues = 34584

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	0.5	6.0	6.0	779.1
Y (mm)	1.2	6.0	6.1	55.0
Z (mm)	-1.7	18.8	18.8	2175.3
Az (°)	0.012	0.347	0.347	41.578
θ (°)	0.584	0.592	0.831	69.513

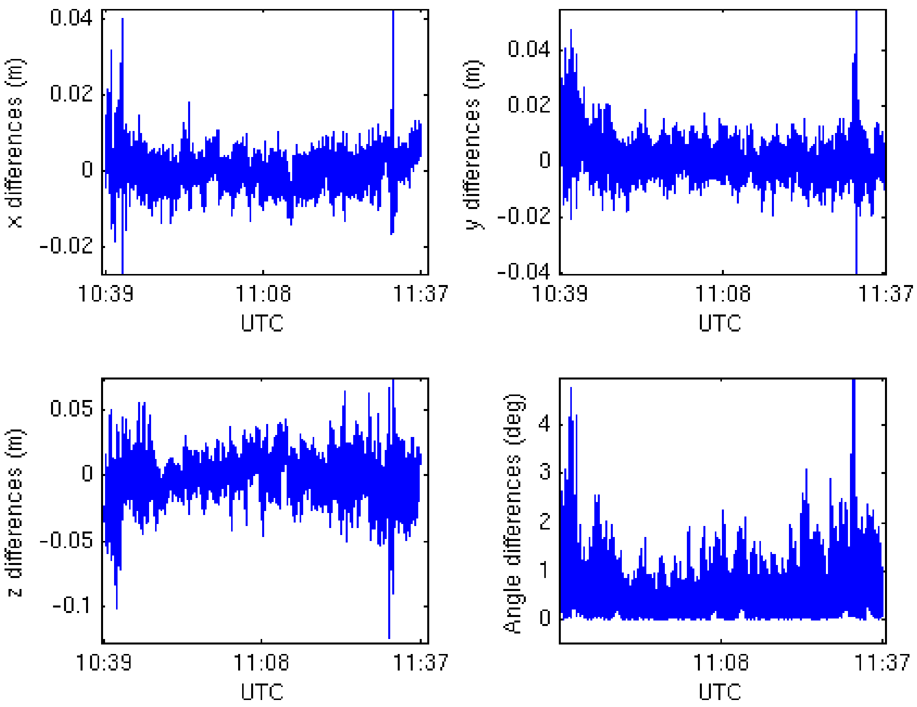


Figure A.27: Plot displaying differences in x-, y-, z-directions and angles for 1006b.

A.4 Individual test runs, orientation

A.4.1 2014-10-01, BASELINE1001a

Table A.43: 16_E0904212, $qRat_R = 1.00$, $qRat_G = 1.00$, nValues = 347, h = 250 m, l = 1015 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-0.8	1.8	1.9	6.1
Y (mm)	-3.7	2.7	4.6	11.0
Z (mm)	7.0	5.3	8.8	20.9
Az (°)	0.042	0.090	0.100	0.290
θ (°)	0.353	0.219	0.415	0.975

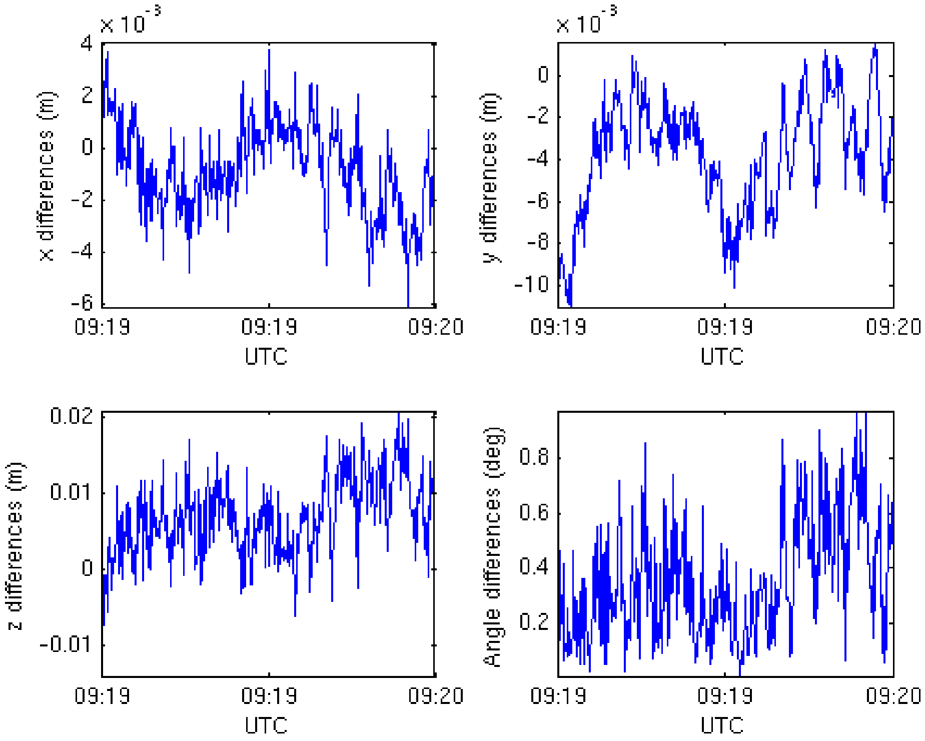


Figure A.28: Plot displaying differences in x-, y-, z-directions and angles for 1001a test run 16 E0904212.

Table A.44: 17_E0904213, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 434, h = 250 m, l = 1013 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-3.5	2.4	4.3	9.4
Y (mm)	5.9	2.6	6.5	12.0
Z (mm)	15.4	8.3	17.5	35.7
Az (°)	-0.114	0.113	0.161	0.414
θ (°)	0.688	0.335	0.765	1.569

Table A.45: 18_E0904214, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 320, h = 250 m, l = 973 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	0.6	1.8	1.8	5.6
Y (mm)	-1.0	3.4	3.5	7.8
Z (mm)	1.0	7.1	7.1	25.2
Az (°)	-0.034	0.082	0.089	0.244
θ (°)	0.296	0.212	0.364	1.155

Table A.46: 19_E0904215, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 431, h = 250 m, l = 1038 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-0.9	2.1	2.3	6.6
Y (mm)	2.9	1.9	3.5	8.5
Z (mm)	12.9	4.6	13.7	24.2
Az (°)	-0.019	0.101	0.103	0.297
θ (°)	0.581	0.208	0.618	1.094

Table A.47: 26_E0904218, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 113, h = 30 m, l = 353 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-0.3	1.7	1.8	4.5
Y (mm)	-2.4	7.4	7.8	16.8
Z (mm)	-7.3	23.3	24.4	58.1
Az (°)	0.038	0.125	0.130	0.307
θ (°)	0.996	0.737	1.239	2.947

Table A.48: 27_E0904219, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 205, h = 30 m, l = 469 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	2.7	3.5	4.4	10.7
Y (mm)	6.8	13.8	15.3	37.1
Z (mm)	19.7	45.0	49.2	121.2
Az (°)	0.264	0.449	0.521	1.226
θ (°)	1.916	1.191	2.256	5.748

Table A.49: 28_E0904220, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 113, h = 30 m, l = 323 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-3.6	1.9	4.1	9.3
Y (mm)	-11.8	7.3	13.9	29.3
Z (mm)	-33.6	22.6	40.5	96.7
Az (°)	0.146	0.093	0.174	0.415
θ (°)	1.805	1.056	2.091	4.947

Table A.50: 29_E0904221, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 153, h = 30 m, l = 368 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-8.2	2.3	8.5	17.0
Y (mm)	-12.4	7.5	14.5	38.4
Z (mm)	-50.2	23.5	55.4	123.6
Az (°)	-0.472	0.144	0.494	0.935
θ (°)	2.214	0.964	2.415	5.063

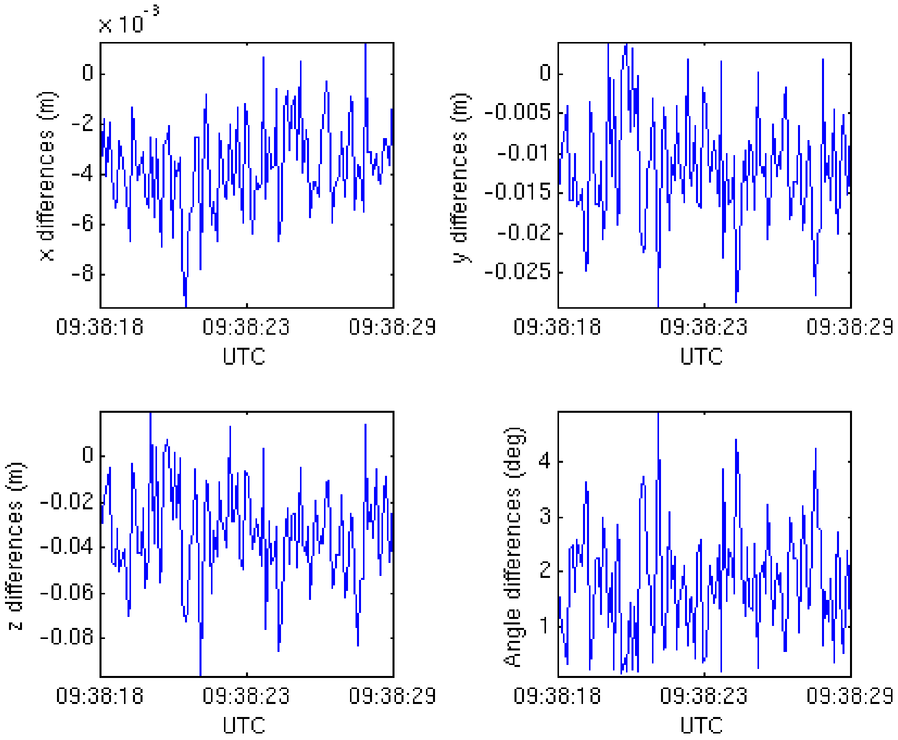


Figure A.29: Plot displaying differences in x-, y- and z-directions for 1001a test run 28 E0904220.

A.4.2 2014-10-06, BASELINE1006c

Table A.51: 10_E0914220, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 97, h = 30 m, l = 242 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	3.4	1.2	3.6	6.1
Y (mm)	-8.4	3.4	9.0	18.8
Z (mm)	-8.9	4.7	10.1	18.6
Az (°)	0.045	0.091	0.101	0.356
θ (°)	0.538	0.241	0.589	1.027

Table A.52: 11_E0914221, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 153, h = 30 m, l = 240 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-1.5	1.4	2.0	4.8
Y (mm)	7.0	3.3	7.8	15.1
Z (mm)	7.9	5.1	9.4	19.4
Az (°)	0.187	0.106	0.215	0.475
θ (°)	0.405	0.172	0.440	0.890

Table A.53: 12_E0914222, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 93, h = 30 m, l = 239 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-0.0	1.8	1.8	4.9
Y (mm)	-2.9	4.7	5.5	15.1
Z (mm)	-0.7	5.0	5.1	13.1
Az (°)	0.066	0.074	0.099	0.206
θ (°)	0.243	0.160	0.291	0.742

Table A.54: 13_E0914223, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 102, h = 30 m, l = 238 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	1.3	1.6	2.0	5.2
Y (mm)	1.9	4.3	4.7	12.8
Z (mm)	6.5	6.8	9.4	23.3
Az (°)	0.113	0.119	0.164	0.415
θ (°)	0.384	0.235	0.450	1.074

Table A.55: 14_E0914218, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 85, h = 30 m, l = 239 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	1.4	2.3	2.7	6.1
Y (mm)	-6.3	4.4	7.7	16.7
Z (mm)	-5.9	5.9	8.4	16.9
Az (°)	0.086	0.107	0.137	0.320
θ (°)	0.445	0.196	0.487	0.961

Table A.56: 15_E0914219, $qRat_R$ = 1.00, $qRat_G$ = 0.990, nValues = 102, h = 30 m, l = 235 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-2.5	2.1	3.3	8.9
Y (mm)	1.3	5.2	5.3	11.3
Z (mm)	-14.5	8.0	16.5	34.9
Az (°)	-0.046	0.155	0.162	0.409
θ (°)	0.731	0.324	0.800	1.604

Table A.57: 8_E0914218, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 79, h = 30 m, l = 249 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	1.1	1.4	1.8	4.2
Y (mm)	-6.3	5.4	8.3	17.4
Z (mm)	-5.0	6.4	8.1	21.1
Az (°)	0.095	0.104	0.141	0.398
θ (°)	0.379	0.309	0.489	1.235

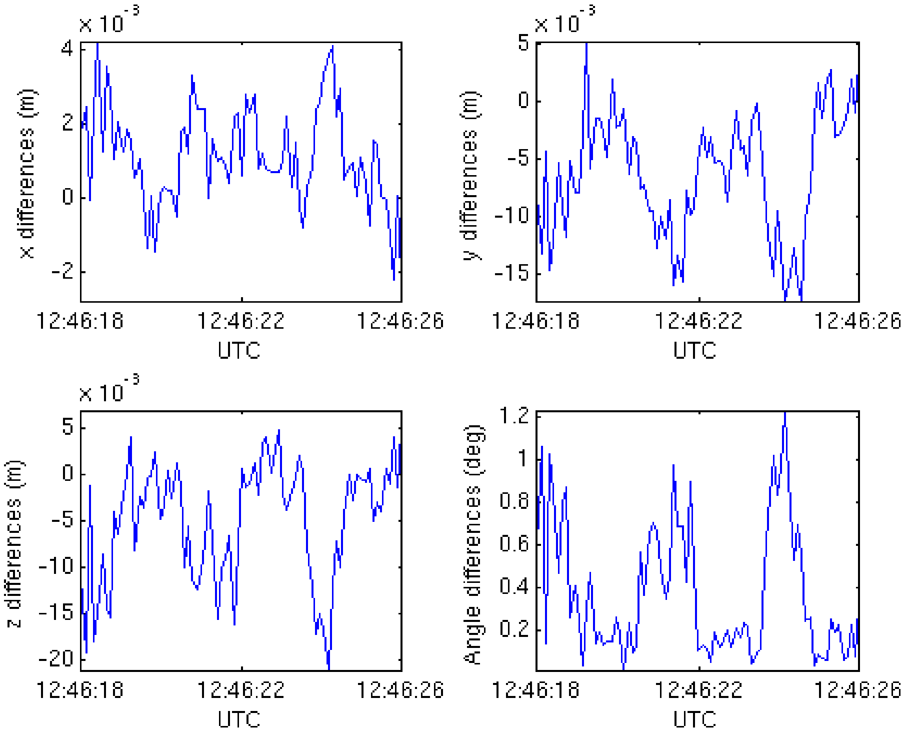


Figure A.30: Plot displaying differences in x-, y- and z-directions for 1006c test run 8 E0914218.

Table A.58: 10_E0915505, $qRat_R$ = 1.00, $qRat_G$ = 1.00, nValues = 497, h = 250 m, l = 1781 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	4.6	10.5	11.5	26.5
Y (mm)	-4.5	10.8	11.7	33.3
Z (mm)	0.6	35.9	35.9	97.0
Az (°)	0.345	0.165	0.382	0.758
θ (°)	1.337	0.817	1.567	3.963

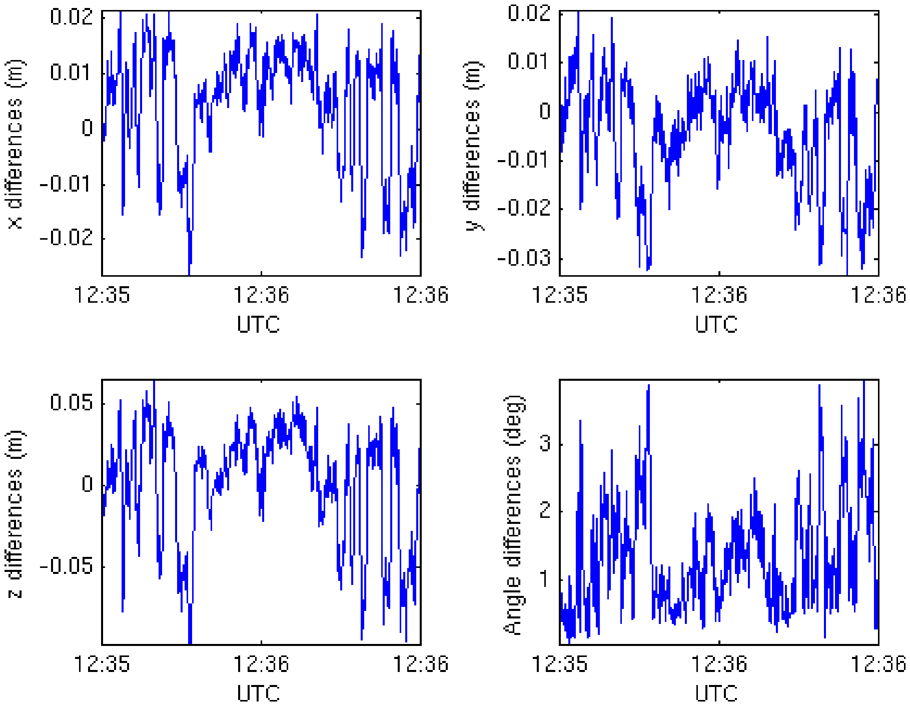


Figure A.31: Plot displaying differences in x-, y- and z-directions for 1006c test run 10 E0915505.

Table A.59: 11_E0915506, $qRat_R$ = 0.202, $qRat_G$ = 0.885, nValues = 494, h = 250 m, l = 1576 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-29.4	56.2	63.4	323.6
Y (mm)	-29.2	62.9	69.4	395.5
Z (mm)	-21.4	224.9	225.9	1696.7
Az (°)	1.448	3.552	3.836	26.195
θ (°)	7.059	8.333	10.921	53.012

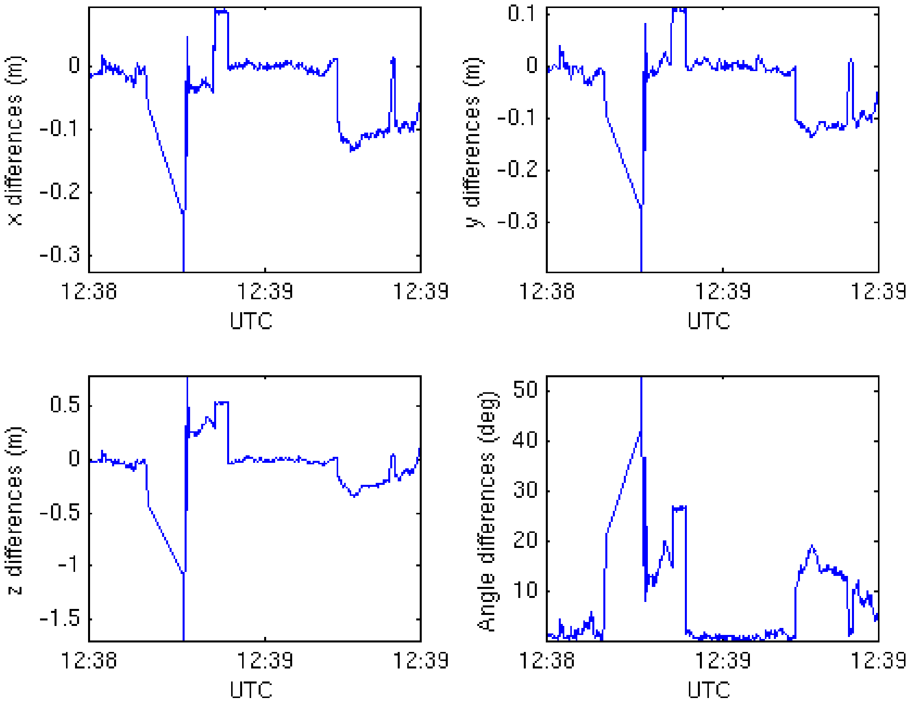


Figure A.32: Plot displaying differences in x-, y-, z-directions and angles for 1006c test run 11 E0915506.

Table A.60: 12_E0915507, $qRat_R$ = 0.584, $qRat_G$ = 0.836, nValues = 505, h = 250 m, l = 841 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-7.3	16.4	17.9	70.7
Y (mm)	-10.3	28.6	30.4	131.0
Z (mm)	-9.6	27.9	29.5	177.0
Az (°)	0.490	1.380	1.465	6.100
θ (°)	1.106	1.537	1.894	9.332

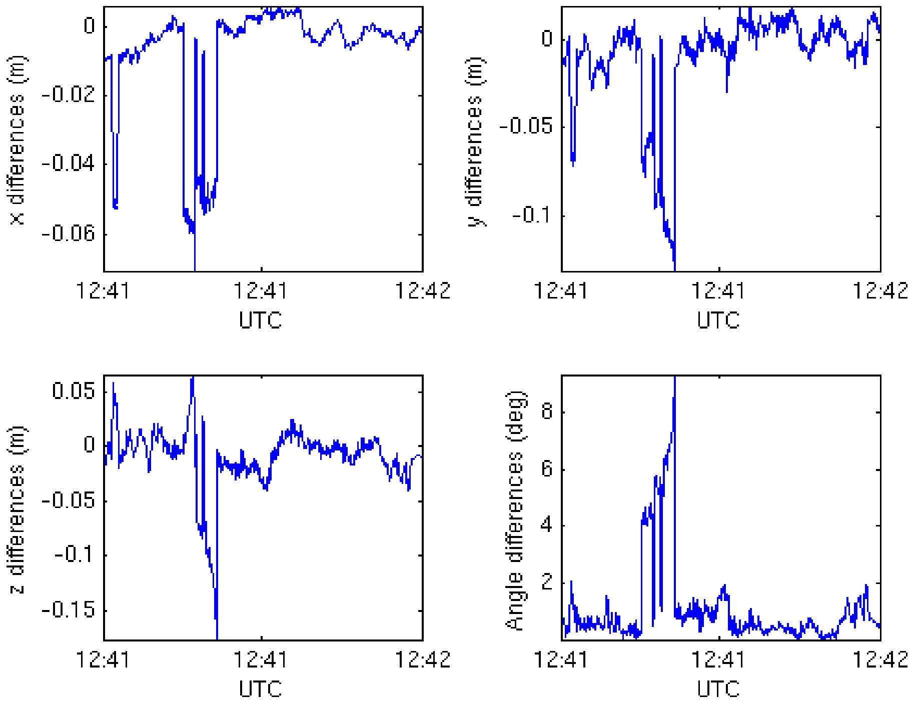


Figure A.33: Plot displaying differences in x-, y-, z-directions and angles for test run 12 E0915507.

Table A.61: 14_E0915509, $qRat_R$ = 1.00, $qRat_G$ = 0.984, nValues = 514, h = 250 m, l = 132 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-0.2	3.2	3.2	11.0
Y (mm)	-2.7	8.8	9.2	27.9
Z (mm)	-5.2	10.9	12.1	30.0
Az (°)	0.039	0.226	0.229	0.868
θ (°)	0.545	0.336	0.640	1.780

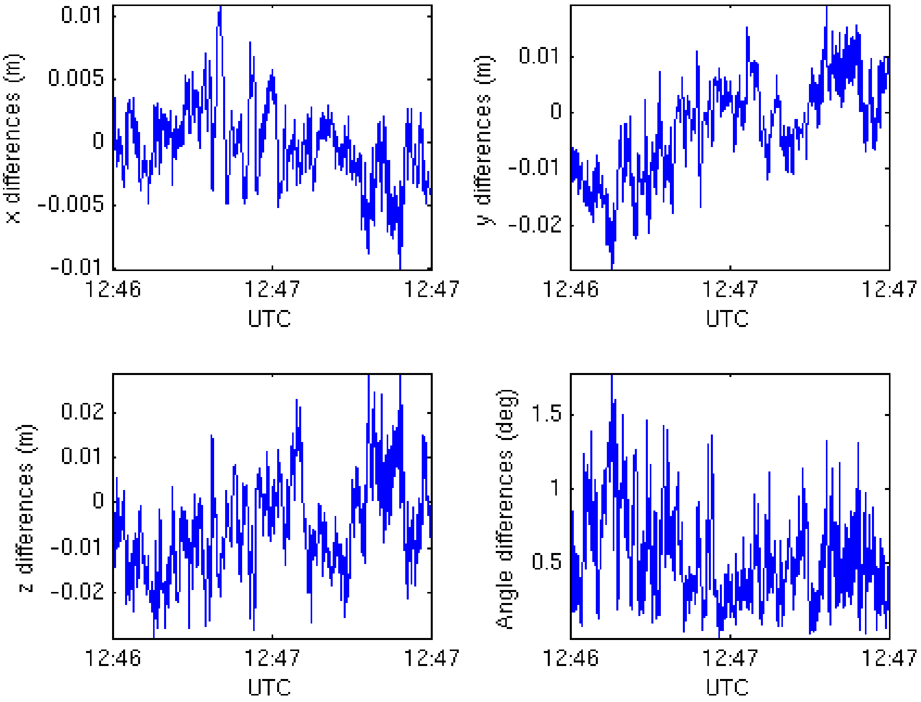


Figure A.34: Plot displaying differences in x-, y-, z-directions and angles for 1006c test run 14 E0915509.

Table A.62: 15_E0915510, $qRat_R$ = 1.00, $qRat_G$ = .998, nValues = 489, h = 250 m, l = 394 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	-1.4	2.8	3.2	8.7
Y (mm)	-0.1	8.3	8.3	23.2
Z (mm)	-3.3	10.3	10.8	37.0
Az (°)	0.115	0.248	0.273	1.216
θ (°)	0.516	0.311	0.603	1.697

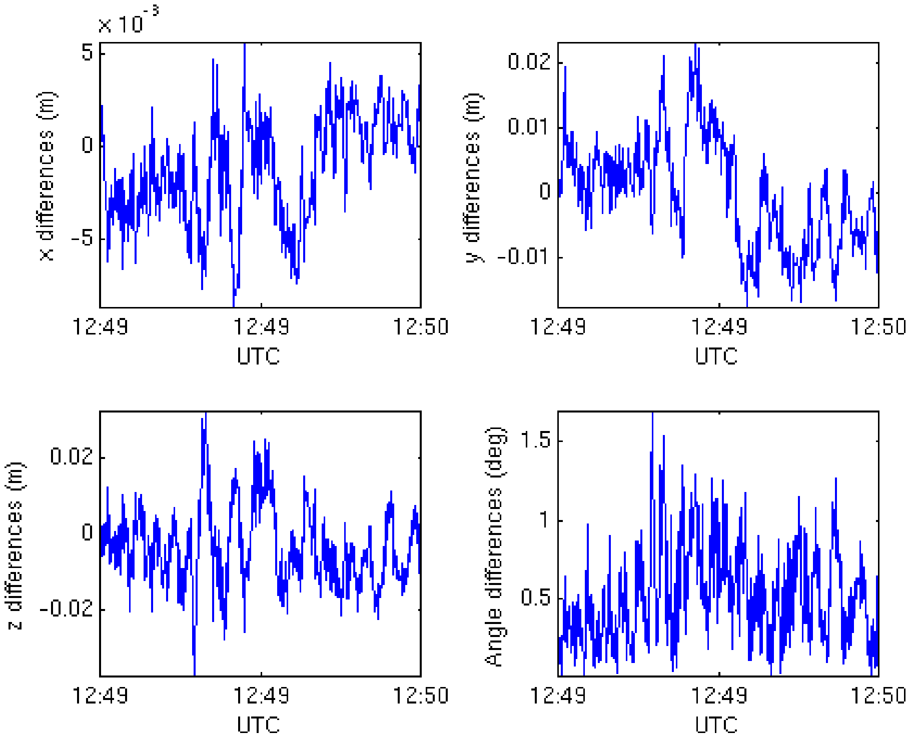


Figure A.35: Plot displaying differences in x-, y- and z-directions for 1006c test run 15 E0915510.

Table A.63: 16_E0915511, $qRat_R = 1.00$, $qRat_G = 1.00$, nValues = 511, h = 250 m, l = 1065 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	0.0	2.0	2.0	8.6
Y (mm)	-2.5	6.7	7.1	21.4
Z (mm)	0.9	8.7	8.7	23.2
Az (°)	0.030	0.183	0.186	1.060
θ (°)	0.397	0.260	0.475	1.151

Table A.64: 17_E0915512, $qRat_R = 1.00$, $qRat_G = 0.928$, nValues = 559, h = 250 m, l = 1377 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	1.3	2.5	2.8	11.8
Y (mm)	-7.2	6.2	9.5	22.1
Z (mm)	-3.7	9.2	10.0	50.5
Az (°)	0.008	0.170	0.170	0.739
θ (°)	0.472	0.316	0.568	2.597

Table A.65: 8_E0915503, $qRat_R = 1.00$, $qRat_G = 1.00$, nValues = 524, h = 250 m, l = 1483 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	3.2	5.5	6.3	17.2
Y (mm)	9.7	6.4	11.6	24.5
Z (mm)	14.4	21.5	25.9	69.5
Az (°)	0.284	0.302	0.415	1.109
θ (°)	1.015	0.706	1.236	3.389

Table A.66: 9_E0915504, $qRat_R = 1.00$, $qRat_G = 0.996$, nValues = 569, h = 250 m, l = 1931 m

Variable	Mean diff.	Std. of diff.	RMS of diff.	Max diff.
X (mm)	1.8	8.5	8.7	28.1
Y (mm)	-0.2	9.9	9.9	33.2
Z (mm)	-3.5	30.9	31.1	119.3
Az (°)	0.052	0.284	0.289	1.054
θ (°)	1.114	0.832	1.390	5.272

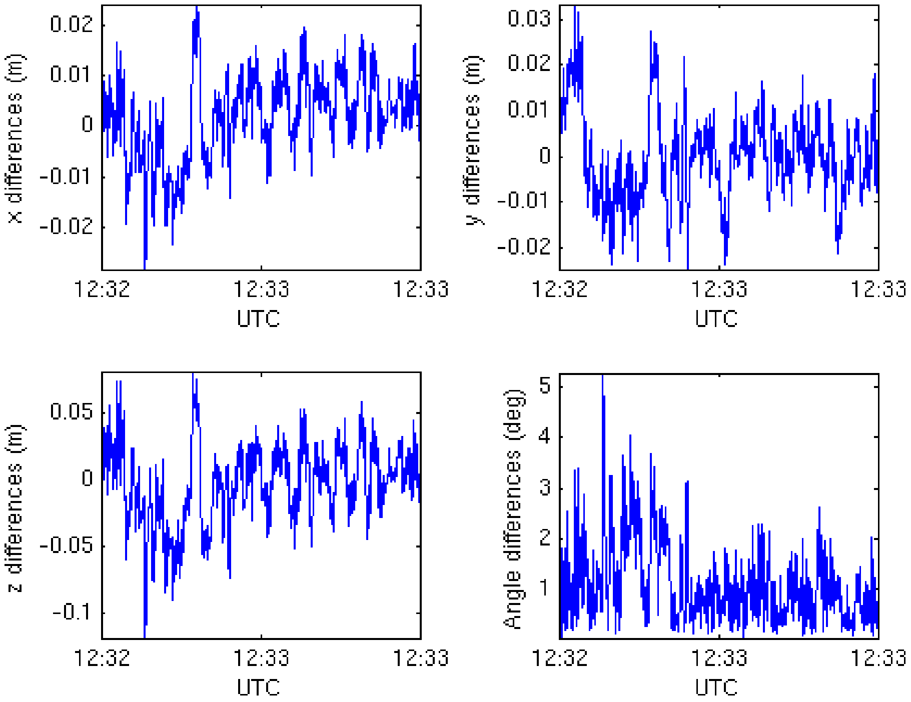


Figure A.36: Plot displaying differences in x-, y-, z-directions and angles for test run 9 E0915504.

Appendix B

Script overview

In this chapter the scripts created for the processing chain utilizing RTKLIB will be briefly presented. Raw data from three antennas are needed in order to be able to create the files with flight path and moving baseline data for the SAR image processing. If the base station position is known that information should be put in a txt-file. If the base station position is not known data from the SWEPOS network can be used to determine the position and create the corresponding txt-file. For this the SWEOPOS data should placed in the correct folder and the main script will then find out how it should proceed.

In Figure B.1 a short overview of the main steps in the processing chain are presented. Script files written in C-shell and MATLAB function files are used in the processing. The processing is started by providing the folders with raw rover and raw base station data to the main script. From there the main script will call `converter.csh` for converting the raw data into RINEX. For this `convbin` from RTKLIB is used.

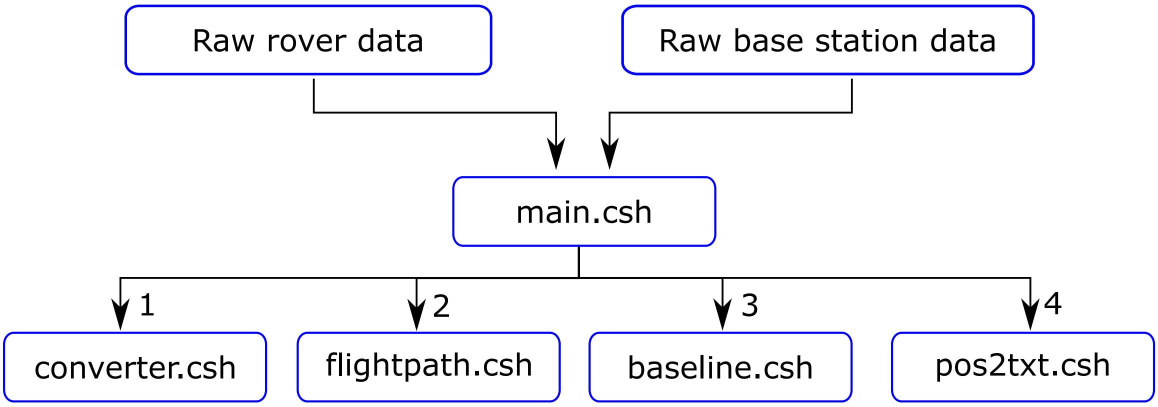


Figure B.1: Main steps in the GPS processing chain.

The next step is to create position files with flight path data which is done in the script called `fligthpath.csh`. Then the base station position is needed. If there exists a file named `basePosition.txt` the position therein will be used. Otherwise the behaviour is controlled by setting the corresponding variables in the main script. The default behaviour is to use SWEPOS data for determining the base station position, but it is also possible to use the approximate position stated in the RINEX header, or use the average of the single point solutions by RTKLIB.

Once the flight path files are created it is time to create the moving baseline files using the script `baseline.csh`. Then information is needed about how far apart the antennas on the helicopter are mounted. This information should be placed in a file named `inst.txt`. If this file does not exist the moving baseline files will be created once without constraints for the baseline. Those files will be used to determine the baseline constraint and create the needed file. Once the setup file exists the baseline files will be created using the baseline constraint. For creating both the flight path and baseline files `rnx2rtkp` from RTKLIB is used.

The last step is to convert the pos-files created by RTKLIB into txt-files with the format that is used be the rest of the SAR processing. This is done using `pos2txt.csh`

By modularizing the code as have been done it is easy for users to execute only part of the processing if needed. For example if it is needed to recreate the baseline data with different settings this can be done without starting from scratch and converting the raw data and creating flight path files.

B.1 Main files

main.csh
Sets all major variables needed by other scripts.

converter.csh
Converts raw GPS data to RINEX.

baseposition.csh
Creates a file with information about the base station position. Needs SWEPOS data.

flightpath.csh
Creates AFT/FWD.pos-files.

baseline.csh
Creates BASELINE.pos-files.

pos2txt.csh
"Wrapper" for corresponding m-file.

B.2 Minor files

checkObsTimes.csh
"Wrapper" for m-file.

checkObsTimes.m
Checks whether 2 observation files match in time.

pos2txt.m
Converts the format of the files created by RTKLIB to the format needed by the SAR processing.

readPos.m
Reads data from pos-files produced by RTKLIB.

readHeader.m
Reads header information from file produced by RTKLIB.

setupBaselineConstraints.m
Creates baseline constraint file.

setupBasePosition.m
Creates base station position file.

Appendix C

Configuration files

In this chapter the configuration files used for the different modes of rnx2rtkp are presented. For explanation of the configuration file refer to [32].

C.1 flightpath.conf

This configuration file is used to create position files for the flightpaths. When the base station position is used the variable "ant2-postype" is overriden by using the flag "-l".

```
# rtkpost options (2015/02/12 07:17:01, v.2.4.2)

pos1-posmode       =kinematic  # (0:single,1:dgps,2:kinematic,3:static,4:movingbase,5:fixed,6:ppp-kine,7:ppp-static)
pos1-frequency     =l1+l2      # (1:l1,2:l1+l2,3:l1+l2+l5,4:l1+l2+l5+l6,5:l1+l2+l5+l6+l7)
pos1-soltype       =combined   # (0:forward,1:backward,2:combined)
pos1-elmask        =15         # (deg)
pos1-snrmask_r     =off        # (0:off,1:on)
pos1-snrmask_b     =off        # (0:off,1:on)
pos1-snrmask_L1    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L2    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L5    =0,0,0,0,0,0,0,0,0
pos1-dynamics      =off        # (0:off,1:on)
pos1-tidecorr      =off        # (0:off,1:on,2:otl)
pos1-ionoopt       =brdc       # (0:off,1:brdc,2:sbas,3:dual-freq,4:est-stec,5:ionex-tec,6:qzs-brdc,7:qzs-lex,8:vtec_sf,9:vtec_ef,10:gtec)
pos1-tropopt       =saas       # (0:off,1:saas,2:sbas,3:est-ztd,4:est-ztdgrad)
pos1-sateph        =brdc       # (0:brdc,1:precise,2:brdc+sbas,3:brdc+ssrapc,4:brdc+ssrcom)
pos1-posopt1       =on        # (0:off,1:on)
pos1-posopt2       =on        # (0:off,1:on)
pos1-posopt3       =off        # (0:off,1:on)
pos1-posopt4       =off        # (0:off,1:on)
pos1-posopt5       =on        # (0:off,1:on)
pos1-exclsats      =           # (prn ...)
pos1-navsys        =1          # (1:gps+2:sbas+4:glo+8:gal+16:qzs+32:comp)
```

```
pos2-armode        =fix-and-hold # (0:off,1:continuous,2:instantaneous,3:fix-and-hold)
pos2-gloarmode     =on           # (0:off,1:on,2:autocal)
pos2-bdsarmode     =on           # (0:off,1:on)
pos2-arthres       =3
pos2-arlockcnt     =0
pos2-arelmask      =0            # (deg)
pos2-arminfix      =10
pos2-elmaskhold    =0            # (deg)
pos2-aroutcnt      =5
pos2-maxage        =30           # (s)
pos2-syncsol       =off          # (0:off,1:on)
pos2-slipthres     =0.05         # (m)
pos2-rejionno      =30           # (m)
pos2-rejgdop       =30
pos2-niter         =1
pos2-baselen       =0            # (m)
pos2-basesig       =0            # (m)
out-solformat      =llh          # (0:llh,1:xyz,2:enu,3:nmea)
out-outhead        =on           # (0:off,1:on)
out-outopt         =on           # (0:off,1:on)
out-timesys        =gpst         # (0:gpst,1:utc,2:jst)
out-timeform       =tow          # (0:tow,1:hms)
out-timendec       =5
out-degform        =deg          # (0:deg,1:dms)
out-fieldsep       =
out-height         =ellipsoidal # (0:ellipsoidal,1:geodetic)
out-geoid          =internal     # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
out-solstatic      =all          # (0:all,1:single)
out-nmeaintv1      =0            # (s)
out-nmeaintv2      =0            # (s)
out-outstat        =off          # (0:off,1:state,2:residual)
stats-eratio1      =100
stats-eratio2      =100
stats-errphase     =0.003        # (m)
stats-errphaseel   =0.003        # (m)
stats-errphasebl   =0            # (m/10km)
stats-errdoppler   =10           # (Hz)
stats-stdbias      =30           # (m)
stats-stdiono      =0.03         # (m)
stats-stdtrop      =0.3          # (m)
stats-prnaccelh    =10           # (m/s^2)
stats-prnaccelv    =10           # (m/s^2)
stats-prnbias      =0.0001       # (m)
stats-prniono      =0.001        # (m)
stats-prntrop      =0.0001       # (m)
stats-clkstab      =5e-12        # (s/s)
ant1-postype       =llh          # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant1-pos1          =90           # (deg|m)
ant1-pos2          =0            # (deg|m)
ant1-pos3          =-6335367.6285 # (m|m)
ant1-anttype       =ACC53G1215A_XT1
ant1-antdele       =0            # (m)
ant1-antdeln       =0            # (m)
ant1-antdelu       =0            # (m)
ant2-postype       =rinexhead    # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant2-pos1          =0            # (deg|m) Note these values will be overriden
ant2-pos2          =0            # (deg|m) during call to rnx2rtkp using flag -l
ant2-pos3          =0            # (m|m)
ant2-anttype       =ASH701975.01B
ant2-antdele       =0            # (m)
```

```
ant2-antdeln       =0          # (m)
ant2-antdelu       =0          # (m)
misc-timeinterp    =off        # (0:off,1:on)
misc-sbasatsel     =0          # (0:all)
misc-rnxopt1       =
misc-rnxopt2       =
file-satantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-rcvantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-staposfile    =
file-geoidfile     =
file-ionofile      =
file-dcbfile       =
file-eopfile       =
file-blqfile       =
file-tempdir       =
file-geexefile     =
file-solstatfile   =
file-tracefile     =
```

C.2 flightpath_single.conf

This configuration file is used to create position files for the flightpaths where the base station position is set as the average of single point solutions. When the base station position is used the variable "ant2-postype" is overriden by using the flag "-l". The only difference between this file and the one shown above is the value of "ant2-postype".

```
# rtkpost options (2015/02/12 07:17:01, v.2.4.2)

pos1-posmode       =kinematic  # (0:single,1:dgps,2:kinematic,3:static,4:movingbase,5:fixed,6:ppp-kine,7:ppp-static)
pos1-frequency     =l1+l2      # (1:l1,2:l1+l2,3:l1+l2+l5,4:l1+l2+l5+l6,5:l1+l2+l5+l6+l7)
pos1-soltype       =combined   # (0:forward,1:backward,2:combined)
pos1-elmask        =15         # (deg)
pos1-snrmask_r     =off        # (0:off,1:on)
pos1-snrmask_b     =off        # (0:off,1:on)
pos1-snrmask_L1    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L2    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L5    =0,0,0,0,0,0,0,0,0
pos1-dynamics      =off        # (0:off,1:on)
pos1-tidecorr      =off        # (0:off,1:on,2:otl)
pos1-ionoopt       =brdc       # (0:off,1:brdc,2:sbas,3:dual-freq,4:est-stec,5:ionex-tec,6:qzs-brdc,7:qzs-lex,8:vtec_sf,9:vtec_ef,10:gtec)
pos1-tropopt       =saas       # (0:off,1:saas,2:sbas,3:est-ztd,4:est-ztdgrad)
pos1-sateph        =brdc       # (0:brdc,1:precise,2:brdc+sbas,3:brdc+ssrapc,4:brdc+ssrcom)
pos1-posopt1       =on        # (0:off,1:on)
pos1-posopt2       =on        # (0:off,1:on)
pos1-posopt3       =off        # (0:off,1:on)
pos1-posopt4       =off        # (0:off,1:on)
pos1-posopt5       =off        # (0:off,1:on)
pos1-exclsats      =           # (prn ...)
pos1-navsys        =1          # (1:gps+2:sbas+4:glo+8:gal+16:qzs+32:comp)
```

```
pos2-armode        =fix-and-hold # (0:off,1:continuous,2:instantaneous,3:fix-and-hold)
pos2-gloarmode     =on         # (0:off,1:on,2:autocal)
pos2-bdsarmode     =on         # (0:off,1:on)
pos2-arthres       =3
pos2-arlockcnt     =0
pos2-arelmask      =0          # (deg)
pos2-arminfix      =10
pos2-elmaskhold    =0          # (deg)
pos2-aroutcnt      =5
pos2-maxage        =30         # (s)
pos2-syncsol       =off        # (0:off,1:on)
pos2-slipthres     =0.05       # (m)
pos2-rejionno      =30         # (m)
pos2-rejgdop       =30
pos2-niter         =1
pos2-baselen       =0          # (m)
pos2-basesig       =0          # (m)
out-solformat      =llh        # (0:llh,1:xyz,2:enu,3:nmea)
out-outhead        =on         # (0:off,1:on)
out-outopt         =on         # (0:off,1:on)
out-timesys        =gpst       # (0:gpst,1:utc,2:jst)
out-timeform       =tow        # (0:tow,1:hms)
out-timendec       =5
out-degform        =deg        # (0:deg,1:dms)
out-fieldsep       =
out-height         =ellipsoidal # (0:ellipsoidal,1:geodetic)
out-geoid          =internal   # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
out-solstatic      =all        # (0:all,1:single)
out-nmeaintv1      =0          # (s)
out-nmeaintv2      =0          # (s)
out-outstat        =off        # (0:off,1:state,2:residual)
stats-eratio1      =100
stats-eratio2      =100
stats-errphase     =0.003      # (m)
stats-errphaseel   =0.003      # (m)
stats-errphasebl   =0          # (m/10km)
stats-errdoppler   =10         # (Hz)
stats-stdbias      =30         # (m)
stats-stdiono      =0.03       # (m)
stats-stdtrop      =0.3        # (m)
stats-prnaccelh    =10         # (m/s^2)
stats-prnaccelv    =10         # (m/s^2)
stats-prnbias      =0.0001     # (m)
stats-prniono      =0.001      # (m)
stats-prntrop      =0.0001     # (m)
stats-clkstab      =5e-12      # (s/s)
ant1-postype       =llh        # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant1-pos1          =90         # (deg|m)
ant1-pos2          =0          # (deg|m)
ant1-pos3          =-6335367.6285 # (m|m)
ant1-anttype       =ACC53G1215A_XT1
ant1-antdele       =0          # (m)
ant1-antdeln       =0          # (m)
ant1-antdelu       =0          # (m)
ant2-postype       =single     # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant2-pos1          =0          # (deg|m) Note these values will be overriden
ant2-pos2          =0          # (deg|m) during call to rnx2rtkp using flag -l
ant2-pos3          =0          # (m|m)
ant2-anttype       =ASH701975.01B
ant2-antdele       =0          # (m)
```

```
ant2-antdeln       =0           # (m)
ant2-antdelu       =0           # (m)
misc-timeinterp    =off         # (0:off,1:on)
misc-sbasatsel     =0           # (0:all)
misc-rnxopt1       =
misc-rnxopt2       =
file-satantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-rcvantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-staposfile    =
file-geoidfile     =
file-ionofile      =
file-dcbfile       =
file-eopfile       =
file-blqfile       =
file-tempdir       =
file-geexefile     =
file-solstatfile   =
file-tracefile     =
```

C.3 baseline.conf

This configuration file is used to create the position files for the baselines without constraints. To set baseline constraints the value of the variables "pos2-baselen" and "pos2-basesig" are changed to appropriate values.

```
# rtkpost options (2015/02/27 12:42:04, v.2.4.3)

pos1-posmode       =movingbase # (0:single,1:dgps,2:kinematic,3:static,4:movingbase,5:↩
    fixed,6:ppp-kine,7:ppp-static)
pos1-frequency     =l1+l2      # (1:l1,2:l1+l2,3:l1+l2+l5,4:l1+l2+l5+l6,5:l1+l2+l5+l6+l7↩
    )
pos1-soltype       =combined    # (0:forward,1:backward,2:combined)
pos1-elmask        =20          # (deg)
pos1-snrmask_r     =off         # (0:off,1:on)
pos1-snrmask_b     =off         # (0:off,1:on)
pos1-snrmask_L1    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L2    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L5    =0,0,0,0,0,0,0,0,0
pos1-dynamics      =off         # (0:off,1:on)
pos1-tidecorr      =off         # (0:off,1:on,2:otl)
pos1-ionoopt       =brdc        # (0:off,1:brdc,2:sbas,3:dual-freq,4:est-stec,5:ionex-tec↩
    ,6:qzs-brdc,7:qzs-lex,8:vtec_sf,9:vtec_ef,10:gtec)
pos1-tropopt       =saas        # (0:off,1:saas,2:sbas,3:est-ztd,4:est-ztdgrad)
pos1-sateph        =brdc        # (0:brdc,1:precise,2:brdc+sbas,3:brdc+ssrapc,4:brdc+↩
    ssrcom)
pos1-posopt1       =off         # (0:off,1:on)
pos1-posopt2       =off         # (0:off,1:on)
pos1-posopt3       =off         # (0:off,1:on)
pos1-posopt4       =off         # (0:off,1:on)
pos1-posopt5       =off         # (0:off,1:on)
pos1-exclsats      =            # (prn ...)
pos1-navsys        =1         # (1:gps+2:sbas+4:glo+8:gal+16:qzs+32:comp)
pos2-armode        =continuous # (0:off,1:continuous,2:instantaneous,3:fix-and-hold)
pos2-gloarmode     =off         # (0:off,1:on,2:autocal)
```

```
pos2-bdsarmode     =off         # (0:off,1:on)
pos2-arthres       =3
pos2-arlockcnt     =0
pos2-arelmask      =0           # (deg)
pos2-arminfix      =10
pos2-elmaskhold    =0           # (deg)
pos2-aroutcnt      =5
pos2-maxage        =30          # (s)
pos2-syncsol       =off         # (0:off,1:on)
pos2-slipthres     =0.05        # (m)
pos2-rejionno      =30          # (m)
pos2-rejgdop       =30
pos2-niter         =2
pos2-baselen       =0   # (m)
pos2-basesig       =0   # (m)
out-solformat      =enu         # (0:llh,1:xyz,2:enu,3:nmea)
out-outhead        =on          # (0:off,1:on)
out-outopt         =on          # (0:off,1:on)
out-timesys        =gpst        # (0:gpst,1:utc,2:jst)
out-timeform       =tow         # (0:tow,1:hms)
out-timendec       =2
out-degform        =deg         # (0:deg,1:dms)
out-fieldsep       =
out-height         =ellipsoidal # (0:ellipsoidal,1:geodetic)
out-geoid          =internal    # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
out-solstatic      =all         # (0:all,1:single)
out-nmeaintv1      =0           # (s)
out-nmeaintv2      =0           # (s)
out-outstat        =off   # (0:off,1:state,2:residual)
stats-eratio1      =100
stats-eratio2      =100
stats-errphase     =0.003       # (m)
stats-errphaseel   =0.003       # (m)
stats-errphasebl   =0           # (m/10km)
stats-errdoppler   =10          # (Hz)
stats-stdbias      =30          # (m)
stats-stdiono      =0.03        # (m)
stats-stdtrop      =0.3         # (m)
stats-prnaccelh    =10          # (m/s^2)
stats-prnaccelv    =10          # (m/s^2)
stats-prnbias      =0.0001      # (m)
stats-prniono      =0.001       # (m)
stats-prntrop      =0.0001      # (m)
stats-clkstab      =5e-12       # (s/s)
ant1-postype       =llh         # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant1-pos1          =90          # (deg|m)
ant1-pos2          =0           # (deg|m)
ant1-pos3          =-6335367.6285 # (m|m)
ant1-anttype       =ACC53G1215A_XT1
ant1-antdele       =0           # (m)
ant1-antdeln       =0           # (m)
ant1-antdelu       =0           # (m)
ant2-postype       =single      # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant2-pos1          =0           # (deg|m)
ant2-pos2          =0           # (deg|m)
ant2-pos3          =0           # (m|m)
ant2-anttype       =ACC53G1215A_XT1
ant2-antdele       =0           # (m)
ant2-antdeln       =0           # (m)
ant2-antdelu       =0           # (m)
```

```
misc-timeinterp    =off        # (0:off,1:on)
misc-sbasatsel     =0          # (0:all)
misc-rnxopt1       =
misc-rnxopt2       =
file-satantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-rcvantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-staposfile    =
file-geoidfile     =
file-ionofile      =
file-dcbfile       =
file-eopfile       =
file-blqfile       =
file-tempdir       =
file-geexefile     =
file-solstatfile   =
file-tracefile     =
```

C.4 static.conf

This configuration file is used for finding the base station position when data from SWEPOS stations are available.

```
# rtkpost options (2015/02/12 07:17:01, v.2.4.2)

pos1-posmode       =static  # (0:single,1:dgps,2:kinematic,3:static,4:movingbase,5:fixed↩
    ,6:ppp-kine,7:ppp-static)
pos1-frequency     =l1+l2      # (1:l1,2:l1+l2,3:l1+l2+l5,4:l1+l2+l5+l6,5:l1+l2+l5+l6+l7↩
    )
pos1-soltype       =combined   # (0:forward,1:backward,2:combined)
pos1-elmask        =15         # (deg)
pos1-snrmask_r     =off        # (0:off,1:on)
pos1-snrmask_b     =off        # (0:off,1:on)
pos1-snrmask_L1    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L2    =0,0,0,0,0,0,0,0,0
pos1-snrmask_L5    =0,0,0,0,0,0,0,0,0
pos1-dynamics      =off        # (0:off,1:on)
pos1-tidecorr      =off        # (0:off,1:on,2:otl)
pos1-ionoopt       =brdc       # (0:off,1:brdc,2:sbas,3:dual-freq,4:est-stec,5:ionex-tec↩
    ,6:qzs-brdc,7:qzs-lex,8:vtec_sf,9:vtec_ef,10:gtec)
pos1-tropopt       =saas       # (0:off,1:saas,2:sbas,3:est-ztd,4:est-ztdgrad)
pos1-sateph        =brdc       # (0:brdc,1:precise,2:brdc+sbas,3:brdc+ssrapc,4:brdc+↩
    ssrcom)
pos1-posopt1       =on        # (0:off,1:on)
pos1-posopt2       =on        # (0:off,1:on)
pos1-posopt3       =off        # (0:off,1:on)
pos1-posopt4       =off        # (0:off,1:on)
pos1-posopt5       =off        # (0:off,1:on)
pos1-exclsats      =           # (prn ...)
pos1-navsys        =1          # (1:gps+2:sbas+4:glo+8:gal+16:qzs+32:comp)
pos2-armode        =fix-and-hold # (0:off,1:continuous,2:instantaneous,3:fix-and-hold)
pos2-gloarmode     =on         # (0:off,1:on,2:autocal)
pos2-bdsarmode     =on         # (0:off,1:on)
pos2-arthres       =3
pos2-arlockcnt     =0
```

```
pos2-arelmask      =0          # (deg)
pos2-arminfix      =10
pos2-elmaskhold    =0          # (deg)
pos2-aroutcnt      =5
pos2-maxage        =30         # (s)
pos2-syncsol       =off        # (0:off,1:on)
pos2-slipthres     =0.05       # (m)
pos2-rejionno      =30         # (m)
pos2-rejgdop       =30
pos2-niter         =1
pos2-baselen       =0          # (m)
pos2-basesig       =0          # (m)
out-solformat      =llh        # (0:llh,1:xyz,2:enu,3:nmea)
out-outhead        =on         # (0:off,1:on)
out-outopt         =on         # (0:off,1:on)
out-timesys        =gpst       # (0:gpst,1:utc,2:jst)
out-timeform       =tow        # (0:tow,1:hms)
out-timendec       =5
out-degform        =deg        # (0:deg,1:dms)
out-fieldsep       =
out-height         =ellipsoidal # (0:ellipsoidal,1:geodetic)
out-geoid          =internal   # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
out-solstatic      =all        # (0:all,1:single)
out-nmeaintv1      =0          # (s)
out-nmeaintv2      =0          # (s)
out-outstat        =off        # (0:off,1:state,2:residual)
stats-eratio1      =100
stats-eratio2      =100
stats-errphase     =0.003      # (m)
stats-errphaseel   =0.003      # (m)
stats-errphasebl   =0          # (m/10km)
stats-errdoppler   =10         # (Hz)
stats-stdbias      =30         # (m)
stats-stdiono      =0.03       # (m)
stats-stdtrop      =0.3        # (m)
stats-prnaccelh    =10         # (m/s^2)
stats-prnaccelv    =10         # (m/s^2)
stats-prnbias      =0.0001     # (m)
stats-prniono      =0.001      # (m)
stats-prntrop      =0.0001     # (m)
stats-clkstab      =5e-12      # (s/s)
ant1-postype       =llh        # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant1-pos1          =90         # (deg|m)
ant1-pos2          =0          # (deg|m)
ant1-pos3          =-6335367.6285 # (m|m)
ant1-anttype       =ASH701975.01B
ant1-antdele       =0          # (m)
ant1-antdeln       =0          # (m)
ant1-antdelu       =0          # (m)
ant2-postype       =rinexhead  # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant2-pos1          =0          # (deg|m) Note these values will be overriden
ant2-pos2          =0          # (deg|m) during call to rnx2rtkp using flag -l
ant2-pos3          =0          # (m|m)
ant2-anttype       =*
ant2-antdele       =0          # (m)
ant2-antdeln       =0          # (m)
ant2-antdelu       =0          # (m)
misc-timeinterp    =off        # (0:off,1:on)
misc-sbasatsel     =0          # (0:all)
misc-rnxopt1       =
misc-rnxopt2       =
file-satantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-rcvantfile    =/proj/Carabas_PDF2/Personal/u025078_Ronny/training/pcv/ngs08.atx
file-staposfile    =
file-geoidfile     =
file-ionofile      =
file-dcbfile       =
file-eopfile       =
file-blqfile       =
file-tempdir       =
file-geexefile     =
file-solstatfile   =
file-tracefile     =
```