POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

GNSS Interferometric Techniques for Attitude Determination

Tesi di Laurea Magistrale in
Aeronautical Engineering - Ingegneria Aeronautica

Author: **Rodrigo Herrero Mártil**

Student ID: 960836
Advisor: Prof. Alberto Luigi Michele Rolando
Academic Year: 2021-22

Abstract

GNSS attitude determination is one of the derived applications from satellite navigation and has been studied intensively and successfully implemented in the last years. This technology is based on differential range measurements between a network of antennas arranged on a vehicle. The measurements used are the very-precise phase measurements, which implies facing the ambiguity resolution problem, as well as cycle slip and multipath detection.

The aim of this thesis is to derive a software application for attitude determination from GPS data. The present documents describes the process followed until the final product: theoretical presentation, mathematical modeling, implementation and validation. The theoretical presentation acknowledges the issues to consider in GNSS attitude determination, with focus on the ambiguity resolution problem, which is analysed with detail. The structure and algorithms of the program are intended to work in a real-time application, although the current implementation works as a post-processing tool. The program contains algorithms for ambiguity resolution, cycle slip and multipath detection, and its performance has been tested in a variety of environments generated with simulated GPS data. The tests carried out in with the simulated data allowed the validation of the programs and the modules as well as the development of comparative analyses on the impact of errors intrinsic to satellite navigation on the case of attitude determination.

Keywords: attitude determination, GNSS, ambiguity resolution, interferometry.

Abstract in lingua italiana

La determinazione dell'assetto con GNSS è una delle applicazioni derivate della navigazione satellitare e è stata studiata in forma intensiva e implementata con successo negli ultimi anni. Questa tecnologia è basata su misure differenziali tra un network di antenne distribuite su un veicolo. Le misure utilizzate sono misure di fase molto precise, che implicano affrontare il problema della risoluzione dell'ambiguità e del rilevamento dei fenomeni di multipath e cycle slip.

Lo scopo di questa tesi è sviluppare una applicazione software per la determinazione degli angoli di assetto dai dati GPS. Il presente documento descrive il processo seguito fino al prodotto finale: presentazione teorica, modellazione matematica, implementazione e validazione. La presentazione teorica contiene le questioni a considerare nel calcolo di assetto con GNSS, con attenzione sul problema della risoluzione della ambiguità, che è analizzato in dettaglio. La struttura e gli algoritmi del programma sono stati costruiti con la intenzione di essere applicati in tempo reale, sebbene l'attuale implementazione funziona come una post-processing tool. Il programma contiene algoritmi per la risoluzione dell'ambiguità, rilevamento di multipath e cycle slips, e le sue prestazioni sono stati testati in una varietà di condizioni generate con dati simulati. Le prove eseguite con i dati simulati hanno permesso di validare il programma e i moduli nonché lo sviluppo di analisi comparativi tra l'impatto degli errori tipici della navigazione satellitare nel caso del calcolo di assetto.

Parole chiave: determinazione d'asetto, GNSS, risoluzione dell'ambiguitá, interferometria

Contents

1 | Introduction

1.1. Context and motivation

Since the beginning of operation of the first Global Navigation Satellite System (GNSS), the American Global Positioning System (GPS), the number of applications derived from this technology has not ceased to grow. This has been motivated by the decrease in the cost of GNSS sensors, continuous improvements introduced in the system, and the addition of other GNSS programs like the Russian GLONASS, the European Galileo and the Chinese Beidou.

One of the first improvements made to the standalone positioning system was the development of differential GNSS (DGNSS, DGPS in the case of GPS), which makes use of the knowledge of the position of one or more reference stations to estimate some of the errors that degrade the performance of the GPS technology and send corrections to receivers. These augmentation systems can provide service from a small area surrounding a single reference station to an extensive region, and they can be ground based (GBAS) or space-based (SBAS). One example of a GBAS with local scope is the Local-Area Augmentation System, used to enhance GPS performance to aid in aircraft landing. Examples of wide area SBAS are the European Geostationary Navigation Overlay System (EGNOS) and the American Wide Area Augmentation System (WAAS).

Another enhancement to the system was to incorporate the usage of carrier phase data. which dramatically increases the accuracy of the GPS solution. However, to use carrier phase measurements, an additional parameter needs to be estimated, which is the integer number of cycles between satellite and receiver, known as integer ambiguity.

With the use of both carrier phase measurements and differential (interferometric) techniques, a receiver with known position can determine the relative position of a set of multiple antennas mounted on a vehicle in a local navigation reference frame. This information, given the known position of the antennas in the body frame, can be processed to obtain the attitude of the vehicle. This particular application of the satellite navigation technology is the topic of this project.

1.2. Objectives

The main objective of this thesis is the development of a real-time attitude determination program. To achieve this purpose, a set of milestones or objectives has been defined:

- Definition of the theoretical framework of the GNSS attitude determination problem. Description of the GNSS technology and the particular characteristics thet make it usable for attitude estimation.
- Development of a sound mathematical model to serve as basis for practical applications. The model should include the attitude determination algorithm, as well as additional solutions for complementary problems like cycle-slip detection and correction and ambiguity resolution.
- Design and implementation of a practical application. The programming language chosen for the application is C++.

1.3. Methodology

The ultimate objective of this project is to develop a functioning library to process GNSS data and provide an attitude solution. In order to fulfil this goal, there are steps to follow which correspond to the objectives defines previoulsy; first present the theoretical aspects, then provide a mathematical model, finally implement the algorithms. These steps are presented sequentially in the following chapters.

The final result, the application, is intended to incorporate functioning scripts to address every problem that needs to be solved. This will mean that some of the theoretical presentations will be limited to the specific technology or method that will be later implemented. For example, there exist a wide variety of methods for cycle slip detection and for ambiguity resolution. Here only one cycle slip method will be commented and, in the case of ambiguity resolution, although a few methods will be presented, only the implemented one(s) will be thoroughly analysed.

1.4. Document structure

The present document is structured as follows. First, chapter 2 will introduce the GNSS technology by means of brief commentaries on the GNSS segments, signals, measurements and errors. chapter 4 contains the full description of the attitude determination problem, with sections on attitude estimation from vector observations, GNSS interferometric techniques, ambiguity resolution, cyle slip detection, etc. Next, chapter 4 contains the commentary on the implementation process, with descriptions of the structure of the program and the main algorithms. chapter 5 presents the results obtained after the processing of GPS measurements with the with the developed program. Finally, chapters 6 and 7 contain conclusions and recommendations derived during the development of this project.

2 | GNSS fundamentals

This chapter serves as introduction to the GNSS system, with descriptions of the most relevant aspects of the technology. section 2.1 briefly describes the three segments a GNSS system. section 2.2 defines the characteristics of GNSS signals that allow the construction of measurements, or observables, which are presented in section 2.3. Finally, section 2.4 is dedicated to the different errors the measurements are subject to.

2.1. Segments

The GNSS system is divided in three segments: the space segment, the ground/control segment and the user segment.

The satellite segment is the set of satellites, usually called constellation, which broadcasts signals to the control and user segment. GPS and Galileo satellites orbit in medium Earth orbits (MEOs) with around 60º inclination and with a period of two orbits per day, approximately [14]. Each satellite, from now on called Space Vehicle (SV), broadcasts two different signals: ranging codes and navigation data messages. Ranging codes are used by the receiver to estimate the time of transmission of the signal, while the navigation message contains information about the satellite orbital state.

The ground or control segment is composed by ground stations with different missions: monitor stations, control stations and uplink stations. Monitor stations receive ranging signals from the satellite and transmit them to the control stations, which calculate the navigation data message and the maneuvers to perform, if needed.

The user segment is the equipment used to pick up and process the signals broadcast by the satellites. The equipment consists on an antenna, a receiver, and ranging and navigation processors. The antenna collects the radio signal (electromagnetic) and converts it to an electrical signal, which is then demodulated by the receiver. The ranging processor determines the range from the antenna to each satellite, and finally the navigation processor calculates the GNSS solution, which is the estimation of the position, velocity and time (PVT).

Both the space and control segment are operated by the same organism. For GPS, the responsible is the Department of Defense through the U.S. Space Force; for Galileo, it is the European Union Agency for the Space Programme.

2.2. Signals

The majority of GNSS signals combine a carrier with a navigation data message, and a ranging code, using Bi-Phase Shift Key (BPSK) modulation . These information is used to construct the GNSS measurements, or observables. GNSS signals belong to the L band, which is subdivided in segments assigned to each particular constellation. The frequency bands relative to GPS and Galileo are presented in table 2.1. The reason a constellation broadcasts in different frequencies is to improve robustness and allow combination of signals between frequencies used for a variety of purposes: atmospheric error estimation, cycle slip detection, etc.

Table 2.1: Frequencies allocation for GPS and Galileo.

GPS	
L1	1575.42 MHz
L2	1227.6 MHz
L5	1176.45 MHz

Galileo	
E1	1575.420 MHz
E6	1278.750 MHz
E5	1191.795 MHz
E5a	1176.450 MHz
E5b	1207.140 MHz

The GNSS signal components are:

- Carrier: sinusoidal signal at a given frequency, used to derive the carrier phase observable.
- Ranging code: Pseudo-Random Noise sequences (PRN) used to determine the travel time between the satellite and the receiver and so to derive the code pseudorange observable.
- Navigation data: message containing information about the state of the satellite, such as ephemeris data (Kepler parameters), clock bias parameters, etc. This information is used mainly to determine the position of the satellite, which is necessary for the positioning solution.

2.3. Measurements

The GNSS measurements, usually called observables, are code pseudoranges, carrier phase and Doppler shift. In the following sections they will be explained and modeled with more detail.

2.3.1. Code pseudorange

The basic GNSS measurement is the travelling time, i.e., the time span from the emission of the signal from the satellite to the reception of the signal by the receiver. This measure is taken by means of a correlation process between the ranging code inside the GNSS signal and a replica code generated by the receiver. The process consists on a shifting in time of the replica code to the point of maximum correlation.

The pseudorange measure, obtained by multiplying the travelling time by the speed of light, is an estimate of the distance between the satellite and the receiver. The prefix pseudo- indicates that the measure does not represent a true distance, understood as the separation between two objects at the same time, but in rather represents the difference in the position of the satellite at emission time and the position of the antenna at reception time. Additionally, there are other effects that make it differ from the true range. In a simplified way, the expression for the code pseudorange is shown in eq. (2.1).

$$p = \rho + c(\delta t_r - \delta t^s) + I + T + k_r + k^s + \epsilon \tag{2.1}$$

where

p – Pseudorange measured by the receiver

ρ – True geometric distance between the user and the satellite

c – Speed of light

$\delta t_r, \delta t^s$ – Receiver and satellite clock errors

I – Ionospheric delay

T – Tropospheric delay

k_r – Receiver instrumental errors

k^s – Satellite instrumental errors

ϵ – Remaining measurement errors (mainly multipath and receiver noise)

2.3.2. Carrier phase

Apart from the code information, another available measurement is the carrier phase of the signal. This measurement can be used as an estimate of the distance between user and satellite and has a much higher precision. However, it is ambiguous by an integer number of cycles (one cycle is 2π radians in phase or one wavelength). The expression of the carrier phase observable is shown in eq. (2.2).

$$\phi = \rho + \lambda N + c(\delta t_r - \delta t^s) - I + T + k_r + k^s + \epsilon \tag{2.2}$$

where

ϕ – Carrier phase measured by the receiver

ρ – True geometric distance between the user and the satellite

λ – Signal wavelength

N – Integer ambiguity

c – Speed of light

$\delta t_r, \delta t^s$ – Receiver and satellite clock errors

k_r – Receiver instrumental errors

k^s – Satellite instrumental errors

I – Ionospheric delay

T – Tropospheric delay

ϵ – Remaining measurement errors (mainly multipath and receiver noise)

Once they achieve lock on a satellite, receivers can dynamically track the change on the number of carrier cycles [18]. This means that the term Φ in eq. (2.2) has a fractional part (direct measure) and an integer part (cycle count). However, before the receiver locks the satellite, there exists an unknown number of carrier cycles between them. These unknown integer is known as *carrier phase ambiguity* and needs to be determined so carrier phase measurements can be exploited. This *ambiguity resolution* process needs to be done every time the receiver loses track of the satellite, as this makes N change arbitrarily and causes range discontinuities. This loss of signal can be caused by the setting of a satellite, aggresive maneuvering by the user, or an obstruction in the Line of Sight between user and satellite [18]. The resolution of the ambiguity, as well as cycle slip detection and correction, is the key to high precision GNSS solutions and will be treated in detail in section 3.3 and section 3.4

2.3.3. Doppler shift

The Doppler shift of a signal corresponds to its time derivative, which must be estimated by the receiver for a successful tracking. Doppler shift is related to the relative velocity between the satellite's and receiver's antennas. This measurement will not be used for the computation of the attitude solution but rather for cycle slip detection.

2.4. Errors

The measurements made by the GNSS receiver are affected by a number of errors that can be associated to the emission, propagation, and reception processes. Emission errors are caused by the satellite and include satellite clock, ephemeris errors, and hardware satellite errors. Errors in the propagation, caused by the presence of the atmosphere, are mainly tropospheric delay and ionospheric delay/advance. Reception errors include hardware effects, receiver noise, interference and multipath. Additionally, relativistic effects must be taken into account. Note that the receiver clock error is not considered formally a measurement error because it can be determined as part of the GNSS solution along with the position.

Most of these errors can be canceled with differential techniques. A more detailed description of the remaining sources of errors for attitude determination will be presented in section 3.6.

Satellite clock error

The satellite clock error is monitored by the GNSS control segment and can be corrected via the navigation message. Satellites' Ephemeris data include three parameters to calculate the clock bias using a second order polynomial.

$$\delta t_s = a_0 + a_1 t_k + a_2 t_k^2 + \Delta t_{rel} \tag{2.3}$$

where Δt_{rel} is a correction due to relativistic effects.

Ephemeris error

Ehpemerides data for the satellites are estimated by the control segment and uplinked to the satellites to be rebroadcast to the user. Errors in the estimation of the satellite's position are in the range 1-6 m (Warren, 2002, cited in [18]), although these errors do not project significantly onto the Line of Sight (LOS) towards the Earth, with a resulting ranging error in the order of 0.8 m (Taylor, 2005, cited in [18]).

Atmospheric effects

The effects of the atmosphere on the measurements are divided into tropospheric effects and ionospheric effects. The ionosphere is a dispersive medium and causes the signal information (code and navigation data) to delay and the carrier phase to advance, in a phenomenon called *ionospheric divergence*, while the troposphere is non dispersive and causes both components to delay.

There exist various mathematical models in the literature to estimate this errors (some examples can be found in [18]), as well as multi-frequency methods to eliminate their effects on the measurements.

Relativistic effects

Both the general and the special relativity theories have an impact in GNSS measurements [18]. The error is periodic and can be compensated by the user with the expression (Dieter, 2003, as cited by [18])

$$\Delta t_{rel} = Fe\sqrt{a}\sin E_k \tag{2.4}$$

where

$$\begin{aligned} F &- \text{constant, equal to -4.442807633} \times 10^{-10} s/m^{1/2} \\ e &- \text{satellite orbital eccentricity} \\ a &- \text{semimajor axis of the satellite orbit} \\ E_k &- \text{eccentric anomaly of the satellite orbit} \end{aligned}$$

Receiver noise

Tracking loops in the receiver produce measurement errors, with thermal noise jitter and interference being the most important sources [18]. The effect of noise in the particular case of attitude determination will be assessed in the following chapter and also with a dedicates analysis when presenting results.

Hardware errors

Both satellite and receiver hardware introduce biases. GNSS signals of different frequencies are not synchronized at the time of transmission due to the different signal paths for each signal. This bias is estimated and broadcast on the navigation message under the *timing group delay* parameter, T_{GD}. Bias errors introduced by the receiver tend to be small compared to other sources and are often consequently ignored [18]. Signals are delayed as they travel through the antenna and the receiver's hardware. The delay is important for timing applications, but can be neglected in the positioning solution, as all signals are almost equally delayed, and therefore this delay will appear as a solution of the least-squares adjustment in the user clock bias estimate.

Multipath

Multipath consists on the reception of reflected or diffracted replicas of the desired original signal, causing a sort of delay which affects the decorrelation process. Multipath errors depend on the receiver location satellite elevation angle, receiver signal processing, antenna gain pattern and the characteristics of the transmitted signal. It also depend on the environment; multipath is prone to appear if reflective objects such as walls,buildings, trees and water surfaces, are close by.

The effect of multipath is one of the most important sources of errors. This is, in part, due to the fact that other effects have been mitigated with GNSS augmentations and modernization or allow precise estimation with mathematical models, while multipath and shadowing are more difficult to predict or model.

3 | GNSS-based attitude determination

This chapter is dedicated to the problem of attitude determination with GNSS technology. The basis of the problem lies on the precise calculation of one or more baseline vectors in two different reference frames; the baselines are formed by network of antennas mounted on the body of the vehicle. One reference frame is the body frame, attached to the vehicle; here the vectors between antennas are known a priori from the distribution of the fixed antennas. The other reference system is the local navigation frame; here the baseline vectors are calculated with the aid of GNSS. To define the local frame, the position of the reference receiver (origin of the baseline vector) must be known. Then, the GNSS baseline solution, given in the Earth Centered, Earth Fixed (ECEF) frame, can be transformed and the attitude obtained.

The precision required for the baseline solution makes it neccessary to use carrier phase observables, with much lower noise than code pseudoranges. The drawback of phase measurements is that they are ambiguous by an integer number of cycles, as shown previously in eq. (2.2). The problem of determining this ambiguity has been of interest in the satellite navigation scientific community, with an extensive family of methods and algorithms developed in the last decades.

Almost all methods developed for attitude determination are based on the double differencing carrier phase technique, with variations on the way attitude is parametrised (Euler angles, quaternions), and differing mainly in the way they tackle the ambiguity resolution problem. A comparative study of different implementations for attitude determination can be seen in [24].

The chapter is organised as follows. section 3.1 shows the attitude parametrisation as well as attitude reconstruction from the baseline vectors; section 3.2 explains the problem of the determination of the baselines; section 3.3 is dedicated to the problem of ambiguity resolution; section 3.4 addresses methods for cycle slip detection.

3.1. Attitude estimation from vector observations

As described in the overview, attitude determination consists on the determination of the baseline vectors between antennas in two different reference frames to then reconstruct the attitude of the vehicle, here parametrised with the Euler angles. Given vector $\mathbf{b_{ab}}$ between to antennas, the relationship of its coordinates between the body frame and the navigation frame can be represented by means of the rotation matrix, as shown in eq. (3.1).

$$\mathbf{b_{ab}^n} = \mathbf{R_{b\to n}} \mathbf{b_{ab}^b} \tag{3.1}$$

Where subscripts n and b indicate navigation and body frames respectively. The explicit description of the rotation matrix depends on the chosen navigation frame. Considering the East-North-Up navigation frame, the expression for R with the Euler angle attitude parametrisation is (eq. (3.2)):

$$\mathbf{R_{b\to n}} = \begin{bmatrix} \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\theta & -\cos\theta\sin\phi & -\cos\theta\cos\phi \end{bmatrix} \tag{3.2}$$

The solution to $\mathbf{R_{b\to n}}$ is not unique if only one baseline is used, as the rotation along the baseline axis is undetermined [14]. In this case, only the elevation and azimuth of the baseline in the navigation frame can be obtained, which do not generally match the Euler angles. Only in the particular case that the antennas are mounted along one of the body axes do elevation and azimuth correspond to two of the Euler angles, leaving the third undetermined. If a network of three or more antennas are used, then at least two baselines can be constructed and determined, making it possible to obtain the full attitude of the vehicle.

The problem of obtaining the rotation matrix between two reference frames from vector observations, also called attitude matrix $\mathbf{A}$, regardless of the specific attitude parameterisation used, is called Wahba's problem, and was first stated by Wahba in 1965 [31]. The problem consist on finding the orthogonal matrix A with determinant $+1$ that minimises the loss function in eq. (3.3).

$$L(A) = \frac{1}{2}\sum_i a_i |\mathbf{b}_i - \mathbf{A}\mathbf{r}_i|^2 \tag{3.3}$$

where $\mathbf{b}_i$ are unit vectors in the body frame, $\mathbf{r}_i$ are the corresponding vectors in the local navigation frame and a_i are positive weights. A number of algorithms have been proposed since the statement of the problem. Reference [22] presents a review on the most popular methods. Whenever three or more vectors are available, the usual approach is to carry out a least-squares adjustment. For the case of two baselines, there are simplifications that allow for closed-form solutions to eq. (3.3). Two of these are presented next in this section. For the case or more than two antennas, the attitude matrix can be obtained with a standard least-squares adjustment or with some of the techniques in [22].

Once the attitude matrix has been obtained, the attitude angles are obtained as shown in eq. (3.4).

$$\theta = \arcsin(A_{31}), \qquad \phi = \arctan\left(\frac{A_{32}}{A_{33}}\right), \qquad \psi = \arctan\left(\frac{A_{11}}{A_{21}}\right) \tag{3.4}$$

where

$$\mathbf{A} = \mathbf{R_{b\to n}}^\top \tag{3.5}$$

3.1.1. TRIAD

The TRIAD method was developed in 1964 [2] and later proved to be a closed form solution to Wahba's problem for a particular choice of the weights on the measured vectors. The input for the method are a couple of vectors measured in body frame, $\mathbf{b}_1$ and $\mathbf{b}_2$, and their expressions in some reference frame, $\mathbf{r}_1$ and $\mathbf{r}_2$. The attitude matrix to be determined is the one that transforms a vector from the reference frame to the body frame (eqs. (3.6) and (3.7)).

$$\mathbf{A}\mathbf{r}_1 = \mathbf{b}_1 \tag{3.6}$$

$$\mathbf{A}\mathbf{r}_2 = \mathbf{b}_2 \tag{3.7}$$

The idea behind TRIAD, "TRIaxial Attitude Determination", is that, if we have an orthogonal right-handed triad in the reference-frame and the corresponding triad in the body-frame ($\mathbf{v}$ and $\mathbf{w}$, respectively), the attitude matrix is (eq. (3.8)):

$$\begin{gathered}\mathbf{A} = [\mathbf{w}_1\mathbf{w}_2\mathbf{w}_3]\,[\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3]^\top = \mathbf{w}_1\mathbf{v}_1^\top + \mathbf{w}_2\mathbf{v}_2^\top + \mathbf{w}_3\mathbf{v}_3^\top \\ \mathbf{A}\mathbf{v}_i = \mathbf{w}_i\,, \quad i = 1, 2, 3.\end{gathered} \tag{3.8}$$

The triads $\mathbf{v}$ and $\mathbf{w}$ are formed from the pairs $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{b}_1$, $\mathbf{b}_2$, respectively, and the third vector is constructed as shown in eqs. (3.9) and (3.10).

$$\mathbf{r}_3 = (\mathbf{r}_1 \times \mathbf{r}_2)/|\mathbf{r}_1 \times \mathbf{r}_2| \tag{3.9}$$

$$\mathbf{b}_3 = (\mathbf{b}_1 \times \mathbf{b}_2)/|\mathbf{b}_1 \times \mathbf{b}_2| \tag{3.10}$$

There are now two ways to construct the attitude matrix (eqs. (3.11) and (3.12)), depending on which vector is emphasized.

$$\mathbf{A}_1 = \mathbf{b}_1\mathbf{r}_1^\top + \mathbf{b}_3\mathbf{r}_3^\top + (\mathbf{b}_1 \times \mathbf{b}_3)(\mathbf{r}_1 \times \mathbf{r}_3)^\top \tag{3.11}$$

$$\mathbf{A}_2 = \mathbf{b}_2\mathbf{r}_2^\top + \mathbf{b}_3\mathbf{r}_3^\top + (\mathbf{b}_2 \times \mathbf{b}_3)(\mathbf{r}_2 \times \mathbf{r}_3)^\top \tag{3.12}$$

The two expressions for the rotation matrix allow some adaptation of the algorithm depending on the expected error of the calculated baseline. For example, if the baseline calculation process returns a float solution for one baseline and a fixed solution for the other one, then the fixed solution is prioritised. If both solutions are fixed, the solution for the longest baseline can be prioritised. There exists also a third TRIAD estimator, which considers both measurements symmetrically. For more information on this method and its variations, see [23], [32].

3.1.2. FOAM

The Fast Optimal Attitude Matrix (FOAM) method allows for the selection of arbitrary weights for the observed vectors and therefore can be adapted to the characteristics of the observation baselines in a similar way as the TRIAD method but with more freedom, as the weights a_i can take any value. This method reduces to the TRIAD method whenever one of the weights is null, $a_1 = 0$, $a_2 = 0$ and when $a_1 = a_2$. The general case of the algorithm is exposed in detail in [21]. The case of two vector observations reduces to solving the set of equations (eqs. (3.13) and (3.14)):

$$\lambda = \sqrt{a_1^2 + a_2^2 + 2a_1a_2\left[(\mathbf{b}_1 \cdot \mathbf{b}_2)(\mathbf{r}_1 \cdot \mathbf{r}_2) + |\mathbf{b}_1 \times \mathbf{b}_2||\mathbf{r}_1 \times \mathbf{r}_2|\right]} \tag{3.13}$$

$$\begin{aligned}\mathbf{A} = \mathbf{b}_3\mathbf{r}_3^\top &+ (a_1/\lambda)\left[\mathbf{b}_1\mathbf{r}_1^\top + (\mathbf{b}_1 \times \mathbf{b}_3)(\mathbf{r}_1 \times \mathbf{r}_3)^\top\right] \\ &+ (a_2/\lambda)\left[\mathbf{b}_2\mathbf{r}_2^\top + (\mathbf{b}_2 \times \mathbf{b}_3)(\mathbf{r}_2 \times \mathbf{b}_3)^\top\right]\end{aligned} \tag{3.14}$$

where in the general case of n measured vectors λ represents the sum of the singular values of the matrix formed with the observations $\mathbf{B} = \sum_i a_i\mathbf{b}_i\mathbf{r}_i^\top$ and $\mathbf{b}_3$ and $\mathbf{r}_3$ are calculated as shown in eqs. (3.9) and (3.10).With more than two vector measurements, the equation for λ has to be solved numerically.

3.2. Baseline determination. GNSS Interferometric techniques

Inteferometric, or differential, techniques are key to precise baseline determination. With this techniques, the solution is not the absolute position of the receiver, but a relative position with respect to a reference. In the case of attitude determination, the reference is one of the antennas, and the relative solution is precisely the baseline vector from that antenna to another.

There are two types of GNSS multi-antenna systems for attitude determination: dedicated and non-dedicated. Dedicated systems use a common receiver designed to synchronise the signals from the antennas, while non-dedicated systems have a receiver associated to each antenna. The advantage of dedicated systems lies on the shared clock, which means that with single differencing the receiver clock can be eliminated. For non-dedicated systems, double-differencing is required to remove the effect of receiver clock bias. On the other hand, non-dedicated systems offer advantages in terms of cost-effectiveness and flexibility [7, p. 45]. Also, the most important effect in accuracy for attitude determination with GNSS is carrier phase multipath error, rather than receiver noise [20, p. 12].

The definition of single and double differences is done hereunder. The results are presented for the carrier phase observable, although the case of pseudoranges is equivalent with the exception that the ambiguity term is not present.

3.2.1. Single differences

Single differences are constructed with the carrier phase measure made by two receivers with a single satellite. This interferometer is shown in fig. 3.1, where the assumption that the LOS vectors to the satellite are parallel (reasonable for close antennas). The mathematical expressions for the carrier phase measure for receivers a and b with the satellite p are based in the model in eq. (2.2) and are shown in eqs. (3.15) and (3.16).

$$\phi_a^p = \rho_a^p + \lambda N_a^p + c(\delta t_a - \delta t^p) - I_a^p + T_a^p + k_a + k^p + \epsilon_a^p \tag{3.15}$$

$$\phi_b^p = \rho_b^p + \lambda N_b^p + c(\delta t_b - \delta t^p) - I_b^p + T_b^p + k_b + k^p + \epsilon_b^p \tag{3.16}$$

The single difference (SD) is constructed subtracting both equations, resulting in:

$$\phi^p_{ab} = \rho^p_{ab} + \lambda N^p_{ab} + c\delta t_{ab} + k_{ab} + \epsilon^p_{ab} \tag{3.17}$$

The expression in eq. (3.17) shows important results. First, the satellite clock bias, as well as the satellite instrumental errors, have been dropped. Also, the atmospheric effects I, T have disappeared on account of the fact that, if the receivers are colatitude and closely spaced (less than 50 km) [18], the effects of the troposphere and the ionosphere cancel out. New combined terms for the integer ambiguity, receiver clock biases, distances and remaining errors have appeared. The combined distance term represents the projection of the baseline between the two antennas into the LOS to the satellite. It can written as:

$$\begin{aligned} \rho^p_{ab} &= (\rho^p_b - \rho^p_a) \\ \rho^p_{ab} &= -\mathbf{1}^p_a \cdot \mathbf{x}_{ab} \end{aligned} \tag{3.18}$$

Where $\mathbf{1}^p_a$ is the Line of Sight vector between the receivers and the satellite. It has been assumed that $\mathbf{1}^p_a \simeq \mathbf{1}^p_b$.

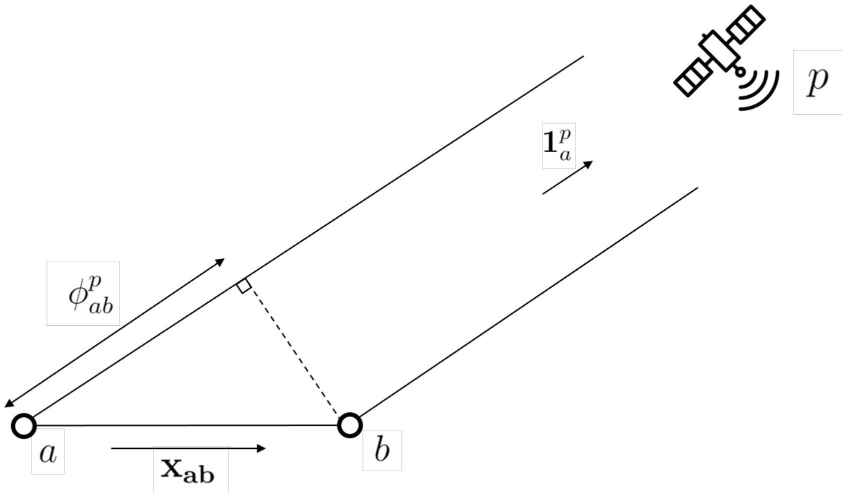


Figure 3.1: Interferometer with one satellite.

In the case of n satellites in view, a total of n single differences can be formed. The unknowns are the coordinates of the baseline vector and the combined receiver clock bias. This means that, in general, at least $four$ satellites are needed to solve the problem. However, taking into account the particular case of attitude determination, only *two* satellites are needed under some conditions [5]. First, the baseline length is known, which adds an additional equation to the system. Furthermore, if a dedicated system (see the introduction) is used, the term associated to the receiver clock error cancels. For the general case of non-dedicated systems, double differencing is required to cancel the receiver clock error effect.

3.2.2. Double differences

Double differences are constructed with the carrier phase measure made by two receivers with respect to two satellites. This is shown in fig. 3.2, maintaining the assumption of identical (parallel) LOSs for each satellite. The mathematical expressions for the single differences formed with receivers a and b with respect to satellites p and q are shown in eqs. (3.19) and (3.20).

$$\phi^p_{ab} = \rho^p_{ab} + \lambda N^p_{ab} + c\delta t_{ab} + k_{ab} + \epsilon^p_{ab} \tag{3.19}$$

$$\phi^q_{ab} = \rho^q_{ab} + \lambda N^q_{ab} + c\delta t_{ab} + k_{ab} + \epsilon^q_{ab} \tag{3.20}$$

The double difference is constructed subtracting both equations, resulting in:

$$\phi^{pq}_{ab} = \rho^{pq}_{ab} + \lambda N^{pq}_{ab} + \epsilon^{pq}_{ab} \tag{3.21}$$

In eq. (3.21) the combined receiver clock biases and the combined instrumental errors have canceled out. A new double difference ambiguity vector has appeared, as well as a combined system noise that takes into account the sources of error, mainly multipath and receiver noise [18]. Following the result shown in eq. (3.18), the combined distance term can be written as shown in eq. (3.22).

$$\begin{aligned} \rho^{pq}_{ab} &= (\rho^p_b - \rho^p_a) - (\rho^q_b - \rho^q_a) \\ \rho^{pq}_{ab} &= -\mathbf{1}^p_a \cdot \mathbf{x}_{ab} + \mathbf{1}^q_a \cdot \mathbf{x_{ab}} \\ \rho^{pq}_{ab} &= -(\mathbf{1}^p_a - \mathbf{1}^q_a) \cdot \mathbf{x}_{ab} \end{aligned} \tag{3.22}$$

The extension to the case where n satellites are available requires the choice of a *master* satellite, typically the one with a higher elevation. The single difference formed with it ϕ^1_{ab} is used to compute the $n-1$ double differences. The resulting system of equations, is shown in matrix form in eq. (3.23), where the result appearing in eq. (3.22) has been introduced.

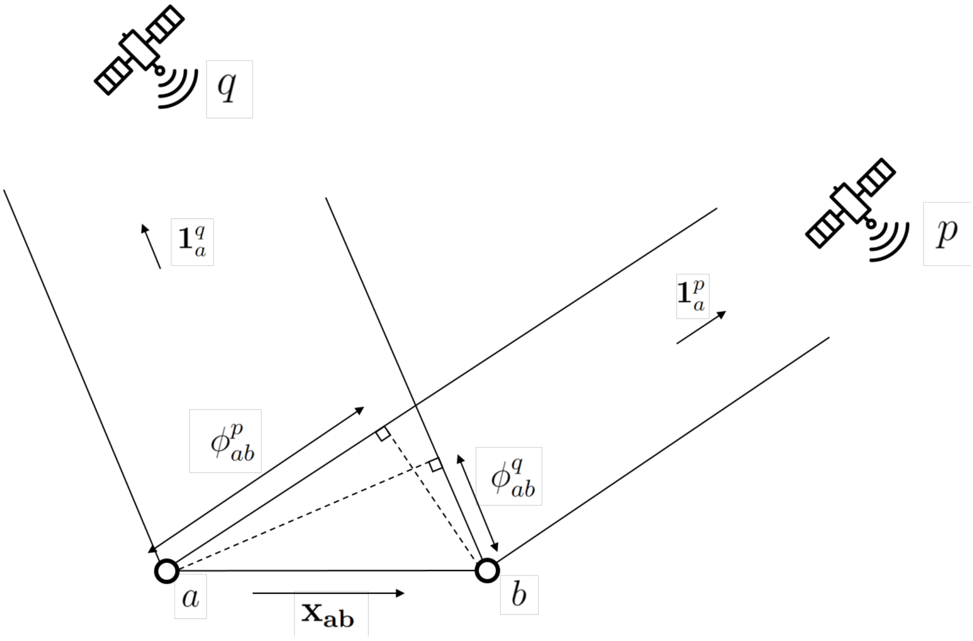


Figure 3.2: Interferometer with two satellites.

$$\begin{bmatrix} \phi_{ab}^{21} \\ \phi_{ab}^{31} \\ \vdots \\ \phi_{ab}^{N1} \end{bmatrix} = \begin{bmatrix} -(\mathbf{1}_a^2 - \mathbf{1}_a^1) \\ -(\mathbf{1}_a^3 - \mathbf{1}_a^1) \\ \vdots \\ -(\mathbf{1}_a^n - \mathbf{1}_a^1) \end{bmatrix} \cdot \mathbf{x}_{ab} + \lambda \cdot \begin{bmatrix} N_{ab}^{21} \\ N_{ab}^{31} \\ \vdots \\ N_{ab}^{n1} \end{bmatrix} + \begin{bmatrix} \epsilon_{ab}^{21} \\ \epsilon_{ab}^{31} \\ \vdots \\ \epsilon_{ab}^{n1} \end{bmatrix} \tag{3.23}$$

$$[\,\mathbf{\Phi}_{ab}\,] = [\,\mathbf{H}_{ab}\,]\,\mathbf{x}_{ab} + \lambda\,[\mathbf{N}_{ab}] + \epsilon_{ab} \tag{3.24}$$

The errors contained in ϵ, mainly due to multipath and receiver noise, are going to be neglected at his moment. Assuming the ambiguity vector $\mathbf{N_{ab}}$ is known, the only unknown appearing in the system in eq. (3.23) is the baseline vector $\mathbf{x_{ab}}$. If we have n satellites, we can construct $n-1$ double differences, and therefore, in principle, a minimum of *four* satellites in view are required for attitude determination, the same needed for carrier phase positioning. Again, if the constraint on the baseline length is considered, another equation can be incorporated to the system. Note that the equation of the baseline length is non-linear so, if it is to be added to the system, it needs to be linearised.

3.3. Ambiguity resolution

From the beginning of the processing of carrier phase measurements, the problem of solving the unknown integer cycles at the moment of acquisition has become of paramount importance. Most of the Ambiguity Resolution (AR) methods share a common structure, with a diagram shown in fig. 3.3. The solution is obtained through two consecutive steps. First, a *float solution* or *estimate* is obtained, where the ambiguities and the baseline vector are real-valued. This is usually done with a least-squares adjustment. Then, the integer nature of the ambiguities is exploited to derive a *fixed solution* which consists in an integer ambiguity vector and a corrected real-valued baseline vector. The second step makes use of the variance-covariance matrix of observations and contains a search technique. These two steps will be further developed further in this section. These methods make use of search approaches to fix the ambiguities to their integer values, and can be divided into two types depending on whether the search is performed in the coordinate domain of the ambiguity domain. The methods more popular at the present time are the latter. Another possible classification is whether the methods require measurements from various epochs, the so-called *multi-epoch* methods, or can instead derive an ambiguity solution with data from only one epoch, and are then considered *single-epoch* methods.

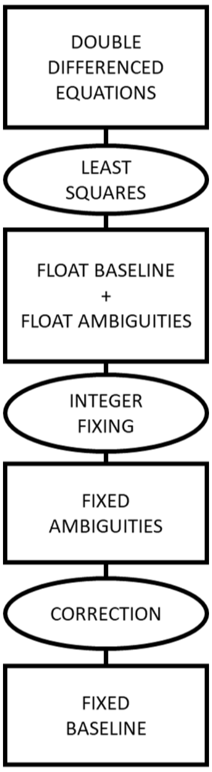


Figure 3.3: Ambiguity Resolution process.

The three most important integer estimators, that serve as a basis for the development of the AR techniques, are Integer Least Squares (ILS), Integer Boostrapping and Integer Rounding. These estimators serve as a basis for the more sophisticated methods that will be presented later in this section.

The discussion on ambiguity resolution is structured as follows. First; a review on the most important methods present in the literature will be carried out, followed by a more practical description of the AR process. This second part of the section includes the details of the calculation of the float and fixed solution, with the definition of the Integer Rounding, Bootstrapping and Least Squares estimators and also shows the specifics of one of the most used methods, the Least-squares AMBiguity Decorrelation Adjustment (LAMBDA), which will be the method implemented in the application.

3.3.1. Overview on ambiguity resolution methods

The most important methods developed to solve the ambiguity resolution problem are shown in table 3.1. The majority of them, with the exception of the Ambiguity Function Method (AFM) method, are based on the Integer Rounding, Bootstrapping and

Least-squares estimators. They present the two sequential float-fixed steps to obtain the ambiguity parameters. The difference between them lies on how they perform the search in the fixing process. Three of the methods from table 3.1, with different approaches to the problem, are briefly commented below.

Method		Reference(s)
Ambiguity Function Method	AFM	[6] [8]
Least Squares Ambiguity Search Technique	LSAST	[16]
Least-squares AMBiguity Decorrelation Adjustment	LAMBDA	[27]
Null method		[9]
Fast Ambiguity Search Filter	FASF	[4]
Fast Ambiguity Resolution Approach	FARA	[11]
Three Carrier Ambiguity Resolution	TCAR	[10]
Cascade Integer Resolution	CIR	[17]

Table 3.1: Overview of ambiguity resolution methods.

The Ambiguity Function Method (AFM), first presented in [6], was one of the first developed and is characteristic because, in contrast to the rest, performs the search step in the coordinate domain. It does not perform the float and fixed solution procedure and rather defines an Ambiguity Resolution Function (AFM) that reaches a maximum when the observed minus computed DD are equal. This function only depends on the coordinates of the candidates for the secondary receiver, located in a determined search space. In theory, the optimal position of the secondary receiver is the one with the correct set of ambiguities. The drawbacks of this method are that the computational efficiency depends on the size of the search space, which can result in long computational time, and that the method may have to discriminate between various maxima points. A modification of this method in order to improve its performance is presented in [8].

The Least Squares Ambiguity Search Technique (LSAST) is targeted to kinematic applications. In this method the double differenced observables are separated into two groups. The first group contains three DD well-posed observables that, with correctly fixed ambiguities, allow the computation of a potential position solution. Then, the potential position can be substituted in the remaining observation equations for the second group and its ambiguity component solved. If the estimated ambiguities for the second group lie outside some limits defined by code observations, the position candidate is rejected, as well as the first group's ambiguity components associated with it. The disadvantages of this method are that it is sub-optimal in general and that it was originally developed for dynamic applications, so it can be applied to a static scenario but it would not exploit the beneficial information that the user's position is constant [19].

The Fast Ambiguity Search Filter makes use of the Kalman filter for the estimation of the float solution, and defines sequential search ranges for the ambiguity parameters. These search ranges are determined in a recursive way making use of a priori geometric information and the influence of other presumably fixed integer ambiguities. This allows for a more efficient search space definition and consequently a reduction in computational time. Another method that implements a sequential space range definition is the Fast Ambiguity Resolution Approach (FARA).

The Three Carrier Ambiguity Resolution and the Cascade Integer Resolution methods make use of three carrier frequencies and are based on the boostrapping estimator. They both make use of a geometry-free widelane combination between frequencies in order to get a signal with a higher wavelength, which increases the success rate when applying the bootstrapping operator. However, this transformation loses the information on geometry of the measurements and is therefore a setback if the objective of the user is to obtain geometric information such as the baseline coordinates [26]. This means that an additional step with a geometry-based model, now including the fixed ambiguities, is required to obtain the desired solution.

3.3.2. Steps in Ambiguity Resolution: float and fixed solution

Float solution

To calculate the float solution, the unknown ambiguity vector is added to the unknowns in eq. (3.23). With n satellites available, this leaves a balance of $(n-1)$ equations to solve for $3+(n-1)$ unknowns. The alternatives to fix the under-determination of the system are either to include code measurements or to include data from additional epochs. The inclusion of pseudoranges should in principle make it possible to achieve a *single-epoch* float solution, with the setback of the inclusion of the noisier code measurements that could degrade the solution. The balance in this case is of $2(n-1)$ equations against $3+(n-1)$ unknowns. The system of equations for the single-epoch, code-phase approach is shown in eq. (3.25), where the error term has been dropped for simplicity.

$$\left[\begin{array}{c} p_{ab}^{21} \\ p_{ab}^{31} \\ \vdots \\ p_{ab}^{N1} \\ \hline \phi_{ab}^{21} \\ \phi_{ab}^{31} \\ \vdots \\ \phi_{ab}^{N1} \end{array}\right] = \left[\begin{array}{c|cccc} -(\mathbf{1}_a^2 - \mathbf{1}_a^1) & 0 & 0 & \dots & 0 \\ -(\mathbf{1}_a^3 - \mathbf{1}_a^1) & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -(\mathbf{1}_a^n - \mathbf{1}_a^1) & 0 & 0 & \dots & 0 \\ \hline -(\mathbf{1}_a^2 - \mathbf{1}_a^1) & \lambda & 0 & \dots & 0 \\ -(\mathbf{1}_a^3 - \mathbf{1}_a^1) & 0 & \lambda & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -(\mathbf{1}_a^n - \mathbf{1}_a^1) & 0 & 0 & \dots & \lambda \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{x}_{ab} \\ \hline N_{ab}^{21} \\ N_{ab}^{31} \\ \vdots \\ N_{ab}^{n1} \end{array}\right] \tag{3.25}$$

$$\left[\begin{array}{c} \mathbf{p}_{ab} \\ \hline \mathbf{\Phi}_{ab} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{H}_{ab} & 0 \\ \hline \mathbf{H}_{ab} & \lambda I \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{x}_{ab} \\ \hline \mathbf{N}_{ab} \end{array}\right] \tag{3.26}$$

The *multi-epoch* approach makes it necessary to wait for the solution with the advantage of using only the more-precise carrier phase measurements. In this other case, the balance is of $m(n-1)$ equations against $3m+(n-1)$ unknowns for the dynamic case (i.e., the baseline changes every epoch), where m represents the number of epochs of data considered. This is possible as long as there are no cycle slips and the receivers do not lose of lock with any satellite. The system of equations is:

$$\left[\begin{array}{c} \mathbf{\Phi}_{ab}^{1} \\ \mathbf{\Phi}_{ab}^{2} \\ \vdots \\ \mathbf{\Phi}_{ab}^{m} \end{array}\right] = \left[\begin{array}{cccc|c} \mathbf{H}_{ab}^{1} & 0 & \dots & 0 & \lambda I \\ 0 & \mathbf{H}_{ab}^{2} & \dots & 0 & \lambda I \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{H}_{ab}^{m} & \lambda I \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{x}_{ab}^{1} \\ \vdots \\ \mathbf{x}_{ab}^{m} \\ \hline \mathbf{N}_{ab} \end{array}\right] \tag{3.27}$$

where the superindex indicates the epoch $i = 1, .., m$.

The systems shown in eqs. (3.26) and (3.27) can be solved with the Least Squares method, yielding the float solution

$$\begin{bmatrix} \hat{\mathbf{x}}_{ab}^{m} \\ \hat{\mathbf{N}}_{ab} \end{bmatrix} \tag{3.28}$$

The *multi-epoch* approach can be also implemented with code and phase measurements. The most general system of equations, regardless of its characteristics (single- or multi-epoch, code-phase or phase based, single- or multi-frequency), will be reformulated as it is presented in the majority of the literature. The new expression is shown in eq. (3.29).

$$y = Aa + Bb + e \tag{3.29}$$

where y is the double-difference observation vector and a and b represent the unknown ambiguity and baseline vectors, respectively. B is the geometry matrix, and A is a matrix containing wavelengths. This new model separates the integer and real-valued unknowns.

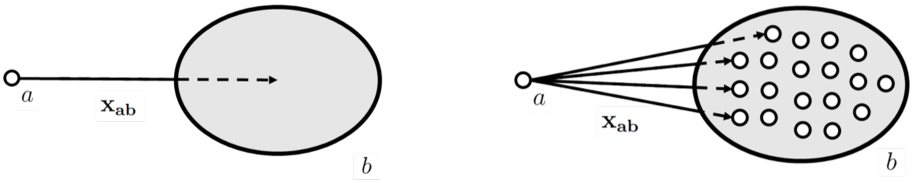
Fixed solution

As stated previously, the system for the float solution can be solved with a least-squares adjustment, preferably with a weighted least-squares approach that takes into account the different noise levels of code and phase measurements. The results are the float estimates and the variance-covariance matrix (eq. (3.30)).

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}, \quad \begin{bmatrix} Q_{\hat{a}} & Q_{\hat{a}\hat{b}} \\ Q_{\hat{b}\hat{a}} & Q_{\hat{b}} \end{bmatrix} \tag{3.30}$$

where the symbol (ˆ) refers to the floating solution. The fixed solution will be marked with the subscript (ˇ).

To obtain the fixed solution, the integer characteristic of the ambiguities is taken into account. This can be seen graphically, as shown in fig. 3.4. fig. 3.4a shows the confidence region of the float solution. If ambiguities were real numbers, the position of the secondary receiver could be any point inside the region; however, due to the integer nature of the ambiguities, only a finite number of points can be considered candidates for the baseline vector, as shown in fig. 3.4b. The solution fixing step consists on choosing the integer values for the ambiguities and then correcting the float baseline solution accordingly. This is shown graphically in fig. 3.5 for a two dimensional case (three satellites in view, two DD equations) where the DD ambiguity vector has only two components. The size of the region including the possible solutions is defined by the covariance matrix of the float solution. In the n-dimensional case, the size of the search space or condifence region would be a hyperellipsoid.



(a) Float

(b) Fixed

Figure 3.4: Sketch of possible baseline solutions.

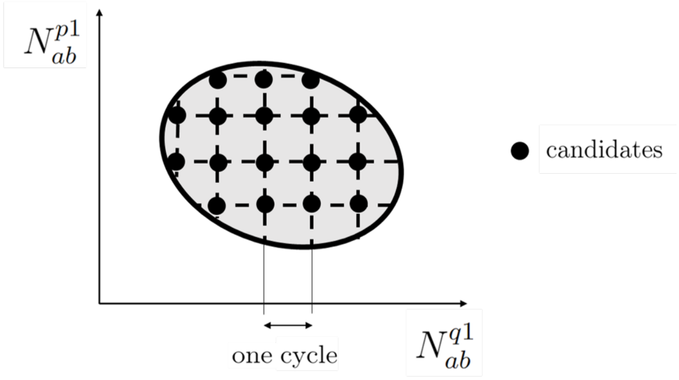


Figure 3.5: Integer fixing in a two dimensional case.

Mathematically, the process of finding an integer estimate from a real solution can be expressed as a mapping process:

$$\check{a} = S(\hat{a}) \tag{3.31}$$

Where $S : \mathbb{R}^n \mapsto \mathbb{Z}^n$ represents the mapping from the n-dimensional space of real numbers to the n-dimensional space of integers [30]. After the integer solution is obtained, the baseline float estimate is corrected (eq. (3.32)).

$$\check{b} = \hat{b}(\check{a}) = \hat{b} - Q_{\hat{b}\hat{a}} Q_{\hat{a}}^{-1} (\hat{a} - \check{a}) \tag{3.32}$$

The space of integers $\mathbb{Z}^n$ is discrete, while the space of real numbers $\mathbb{R}^n$ is continuous. This means that many float numbers will be mapped into the same integer. A *pull-in region* is defined as the set of floats that will result in the same integer after the mapping. This is expressed in mathematical terms in eq. (3.33). Each integer estimator is defined by its pull-in region [30].

$$S_z = \{x \in \mathbb{R}^n \mid \quad z = S(x)\}, \quad z \in \mathbb{Z}^n \tag{3.33}$$

For a more detailed discussion of the integer estimation process, see [30]. There exist a variety of methods depending on the choice of the mapping function S.

3.3.3. Integer estimators

Integer rounding

The simplest way to obtain a fixed solution is to simply round the float ambiguities to the nearest integer. This approach does not take into account the variances of the observables. The pull-in region for this method, $S_{z,R}$, is shown in eq. (3.34) [30].

$$S_{z,R} = \left\{x \in \mathbb{R}^n \mid \quad |x_i - z_i| \leq \frac{1}{2}\right\}, \quad z \in \mathbb{Z}^n \tag{3.34}$$

and represents an n-dimensional unit cube, centered at the integer. The two-dimensional case is a unit square, as shown in fig. 3.6.

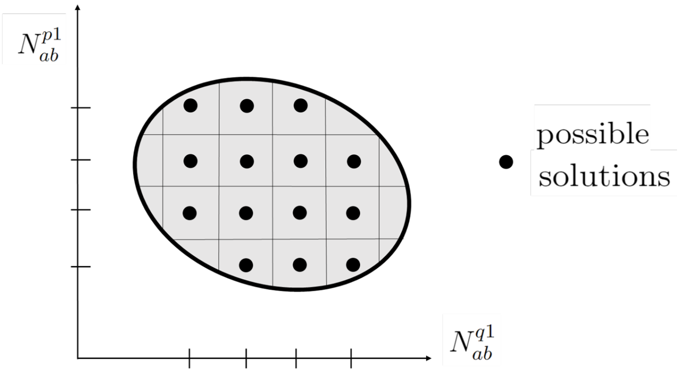


Figure 3.6: Pull-in region for the rounding estimator.

Integer bootstrapping

Integer bootstrapping, also called sequential integer rounding, takes the correlation between the components float ambiguity vector into consideration. It follows from a sequential least squares update. The process begins with rounding of the most precise ambiguity. Then, the remaining $(n-1)$ floating ambiguities are corrected by means of their correlation with the rounded ambiguity. The process is repeated now with the second-most precise float ambiguity, which is rounded to the nearest integer. Then, the remaining $(n-2)$ floating values are corrected again, this time by means of their correlation with the second ambiguity.

The process for this estimator is shown in eq. (3.35) [28], where the first ambiguity a_1 is assumed to be the most precise, and therefore the starting point.

$$
\begin{aligned}
\check{a}_1 &= [\hat{a}_1] \\
\check{a}_2 &= [\hat{a_{2|1}}] = [\hat{a}_2 - \sigma_{\hat{a}_2\hat{a}_1}\sigma^{-2}_{\hat{a}_1}(\hat{a}_1 - \check{a}_1)] \\
&\vdots \\
\check{a}_n &= [\hat{a}_{n|N}] = \left[\hat{a}_n - \sum_{i=1}^{n-1} \sigma_{\hat{a}_n\hat{a}_{i|I}}\sigma^{-2}_{\hat{a}_{i|I}}(\hat{a}_{i|I} - \check{a}_i)\right]
\end{aligned}
\tag{3.35}
$$

The term $\hat{a}_{i|I}$ represents the ith float ambiguity obtained after applying the corrections for the previous $I = \{i+1, ..., n\}$ sequentially rounded ambiguities.

The resulting integer values depend on the choice of the first ambiguity to round. Also, the estimator works better if applied to the decorrelated ambiguities [30]. The pull-in regions for the 2D case are shown in fig. 3.7.

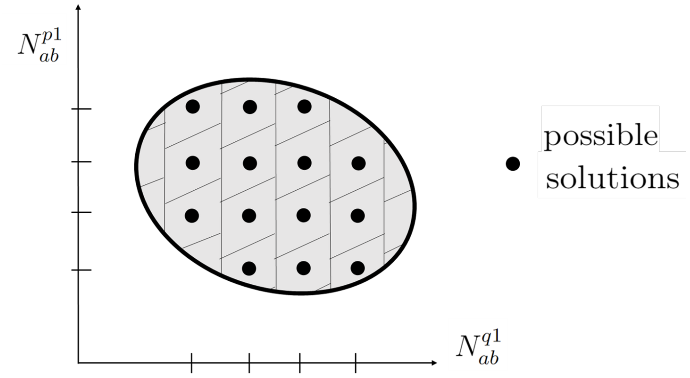


Figure 3.7: Pull-in region for the bootstrapping estimator.

Integer Least Squares

The ambiguity resolution problem can be formulated as an optimisation problem taking into account the integer constraint on the ambiguities. The problem is called *integer least-squares* (ILS) and belongs to the family of non-standard least-squares problems.

The problem is (eq. (3.36)).

$$\min_{z,\zeta} ||y - Az - B\zeta||^2_{Q_y}\,, \quad z \in \mathbb{Z}^n\,,\ \zeta \in \mathbb{R}^p \tag{3.36}$$

Following an orthogonal decomposition [30],

$$||y - Az - B\zeta||^2_{Q_y} = ||\hat{e}||^2_{Q_y} + ||\hat{a} - z||^2_{Q_{\hat{a}}} + ||\hat{b}(z) - \zeta||^2_{Q_{b|a}} \tag{3.37}$$

where $\hat{e}$ is the residual of the float solution and $\hat{b}(z)$ is the conditional baseline estimator eq. (3.32). in eq. (3.37), the first term comes from the adjustment of the float solution, the second term is the integer estimation and the third term is the baseline fixing. The integer least-squares method solves for the second term (eq. (3.38)).

$$\check{a} = \arg\min_{z\in\mathbb{Z}^n} ||\hat{a} - z||^2_{Q\hat{a}} \tag{3.38}$$

The solution of the ILS problem involves a search process, unlike the previous methods. The pull-in regions of ILS are determined by the variance-covariance matrix, or vc-matrix. An example in a two dimension problem is shown in fig. 3.8.

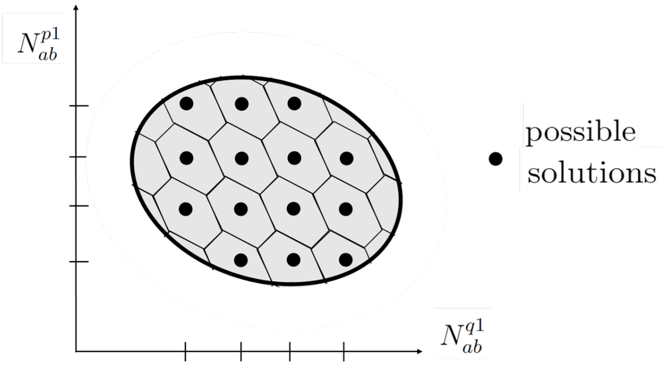


Figure 3.8: Pull-in region for the ILS estimator.

3.3.4. LAMBDA method

Developed by Teunissen [27] [25], the Least-Squares AMBiguity Decorrelation Adjustment is one of the most extended integer techniques for fast ambiguity resolution. It incorporates a decorrelation process before solving the integer least-squares problem. The reason for that is that the search space for ILS, defined as

$$\Omega_a = \{a \in \mathbb{Z}^n \mid (\hat{a} - a)^\top Q_{\hat{a}}^{-1}(\hat{a} - a) \leq \chi^2\} \tag{3.39}$$

is very elongated due to the high correlation between the ambiguities. This means that the search space may take a long time [30].

The decorrelation process transforms the search space into a more spherical one, thus maken the search more efficient. The transformation step is shown in eq. (3.40) and the transformed search space in eq. (3.41).

$$\hat{z} = Z^\top \hat{a}, \quad Q_{\hat{z}} = Z^\top Q_{\hat{a}} Z, \quad \check{a} = Z^{-\top} \check{z} \tag{3.40}$$

$$\Omega_z = \{z \in \mathbb{Z}^n \mid (\hat{z} - z)^\top Q_{\hat{z}}^{-1} (\hat{z} - z) \leq \chi^2\} \tag{3.41}$$

After a $LDL^\top$ decomposition of $Q_{\hat{z}}$, the right hand-side of eq. (3.41) can be rewritten as:

$$\sum_{i=1}^{n} \frac{(\hat{z}_{i|I} - z_i)^2}{\sigma_{i|N}^2} \leq \chi^2 \tag{3.42}$$

where $\hat{z}_{i|I}$ is the conditional least-squares estimator and $\sigma_{i|N}^2$ are the conditional covariances from matrix D. The n intervals of the search step are:

$$\begin{aligned} (\hat{z}_1 - z_1)^2 &\leq \sigma_1^2 \chi^2 \\ (\hat{z}_{2|1} - z_2)^2 &\leq \sigma_{2|1}^2 \left(\chi^2 - \frac{\hat{z}_1 - z_1}{\sigma_1^2}\right) \\ &\vdots \end{aligned} \tag{3.43}$$

The choice of the search space χ^2 is done with the help of the bootstrapped solution, which is a good approximation to ILS. The decorrelation step in the LAMBDA method can be applied to the float solution regardless of the specific estimator that is going to be applied later, increasing the success rate of integer rounding and bootstrapping. An estimation of this success for the LAMBDA method was proposed in [28] based on the covariance matrix of the float solution. In theory, the expression estimates the probability of obtaining the true ambiguities with bootstrapping. However, the result can be used as a lower bound for the Integer Least Squares estimator and as an upper bound for the Integer Rounding estimator. The expression is shown in eq. (3.44) where Φ is the cumulative normal distribution function and $\sigma_{i|I}$ is the conditional standard deviation of the ith ambiguity when the previous I ambiguities have been fixed.

$$P_{s,B} = \prod_{i=1}^{n} \left(2\Phi\left(\frac{1}{2\sigma_{i|I}}\right) - 1\right) \tag{3.44}$$

The LAMBDA method has been improved over the years with the introduction of the baseline length constraint into the problem, rather than as a validation criteria (C-LAMBDA method [3]) and with reformulations of the problem, as the Multivariate Constrained LAMBDA (MC-LMABDA) method [12, 13].

3.4. Cycle slip detection and correction

When a receiver loses lock on a satellite, and then reacquires the signal again, the phase measure is altered by an unknown number of cycles no matter how brief the loss of continuity is. This can be caused by excessive maneuvering, scintillation, an obstruction in the line of sight to the satellite, etc. The result of a loss of lock is an erroneous measure that will corrupt the attitude solution unless taken care of, making it crucial to detect when a cycle slip occurs. Once detected, the usual approach is to "ignore" the offending satellite for a determined number of epochs instead of trying to correct or repair the cycle slip. Once the satellite is accepted again, the ambiguity resolution process is applied to the satellite [18, p.422]. This makes it possible, considered the minimum number of satellites is maintained, to obtain a fixed solution during both the time-out period for the satellite and the ambiguity resolution process.

This is true unless the cycle slip occurs with the pivot or reference satellite. In this case the approach is to either restart the ambiguity resolution process for all satellites, or choose a new reference satellite and reconstruct the new double-differenced ambiguity vector.

A simple cycle slip detection methods is based on the two-frequencies *geometry-free* combination. This linear combination removes the gometry and the clocks and non-dispersive effects in the signal. The resulting measure has very low noise and presents a low change between epochs unless a cycle slip occurs, making it a suitable signal for cycle slip detection. It must be noted that with this combination signal noise (i.e., signal instabilities) is amplified. The most simple implementation would consists on just differentiating the signal between consecutive epochs and check that the change remains within a certain admissible threshold. A more complex and robust implementation would be to use a polynomial fit to predict the signal value, and then check the change. This second approach is more suitable when the sampling time is big and the natural change of the signal becomes comparable to the jump produced by a cycle slip.

For attitude determination the rate of the measurements is usually high ($f_s > 1$ Hz), and therefore the simple method without the polynomial fit is assumed be sufficient for correct cycle slip detection. The setback of this method is that, as it makes use of phase observables in two frequencies, it does not give information of on which frequency the cycle slip actually occurs. This is a problem when the the main GNSS algorithm only uses one frequency, because the cycle slip flag may be activated unnecessarily.

Another simple cycle slip detector makes use of the Doppler shift measure, which is an estimation of the rate of change of the phase observable. This allows for an estimation of the phase cycles for the next epoch. The estimation is checked against the actual measured cycles and, if the jump is too big, a cycle slip may have occurred. The setback of this method is that it depends on the resolution of the Doppler shift observable and the phase observable, making it hard to detect small cycle slips.

However, a small cycle slip passes through the detection algorithm it will be because of its small effect on the value of the phase observable, so it can be assumed that it will not deteriorate much the attitude solution, at least in the beginning. The solution may drift over time, as the geometry of the satellites and the receiver changes and the "skipped" cycle slip will start having an effect on the estimated attitude [20, p. 117]. This drift will happen also to the calculated baseline length, so it can be tracked and, if the calculated length diverges far from its known value, the ambiguity resolution process can be launched again. The same effect and solution procedure can be applied when the calculated ambiguities are not correct by a small number of cycles.

3.5. Multipath mitigation

As stated in the previous chapter, the most important errors that the double difference solution is subject to are receiver errors and multipath. There is little to be done regarding the receiver error once the hardware is fixed.

In the case of multipath, the development of mitigation techniques has been an active research field. One group of mitigation techniques focuses on the environment close to the antenna, by using special materials in the proximity of the antenna or by changing its placement with respect to the ground: close if there are few obstructions neighboring (e.g. in an open field) or high if there are obstacles close. The approaches based on receiver processing are classified as parametric when they estimate paremeters of multipath and correct for their effects, and nonparametric when they just employ discriminators that are less sensitive to multipath effects [18, p. 293].

Most of these receiver processing techniques act on the acquisition and tracking stages of the GNSS signal. This lies outside the scope of the present project and therefore none of this techniques are going to be implemented in the program. The approach taken is similar to the one used for cycle slips. Once the presence of multipath is detected, the offending satellite will not be used for a predetermined number of epochs, assuming that the motion of both the satellite and the vehicle will make the receiver acquire the true signal, without multipath effects, after some time. For the detection of multipath, the Doppler prediction mentioned in section 3.4 will be used again. Details of the specific implemented algorithm are shown in chapter 4.

3.6. Errors and performance

Unlike the rest of the error sources, multipath and receiver noise do not cancel with interferometric techniques. These two effects are uncorrelated between receivers and therefore the resulting error variance is the sum of the error variances of each receiver. From the two, multipath is considered to be the most important [18, p. 390].

Carrier phase measurements, vital for attitude determination, have the advantage of lower measurement noise (typically 1 mm) with respect to code measurements. After double differencing, the combined receiver noise can be estimated around 2 mm, while a typical value for multipath effects on marine and land environments is 1-2 cm [20].

An estimation of measurement and estimation errors is important because the variance in the observables has a direct effect on the solution of the systems of equations for baseline determination (eq. (3.23)), float solution eqs. (3.25) and (3.27) and integer ambiguity resolution. These systems need to be solved with a weighted least-squares adjustment, shown in appendix A, in order to get the covariance matrices of the estimates and later apply the integer ambiguity resolution techniques. The effect of variance estimation on ambiguity resolution was investigated in [1], where the estimated standard deviation for the phase observables was $\sigma_{L1} = 5.2$ mm, $\sigma_{L2} = 4.9$ mm for the L1 and L2 frequencies. Another estimation of these parameters is done in [3, p. 65] with results for the code and phase standard deviations equal to $\sigma_p = 30 - 50$cm, $\sigma_\phi = 3$ mm. These values are used in simulation-based experiments to generate the observables data.

Apart from the intrinsic errors in the estimation of the baseline, the accuracy of the attitude solution is also dependant on the baseline length. Considering a 1m baseline can be measured to 1cm precision, the standard deviation of the attitude solution is 10 mrad ($\sim 0.6^{\text{o}}$) [14, p. 289]. This means that for a given accuracy in the GNSS baseline solution, longer baselines provide more precise results, although for long baselines the effect of flexibility can be important.

The influence of multipath and receiver noise in the solution will be studied later in chapter 5.

4 | Development

This chapter is dedicated to the implementation of the different algorithms and techniques presented in chapter 3. The program is designed as a C++ application that gets as input GPS recordings from two or three antennas and produces as a result an attitude solution. Even though the input is not real-time data but a recording, the program is designed as to be able to work in real-time. This means that at each iteration, the application reads one epoch of data and produces the corresponding attitude solution.

The program has been developed in Eclipse IDE and makes use of the external library Eigen for the handling of matrices and vectors [15]. The generation of code documentation has been done with the help of the tool Doxygen [29].

It is structured as follows. section 4.1 describes the structure of the input data and the required output for the program and presents the main features and modules needed to fulfill the requirements. section 4.2 presents the structure of the program and the general working logic. section 4.3 shows in more detail the specific algorithms and implementation of the modules.

4.1. Input and output. Program features

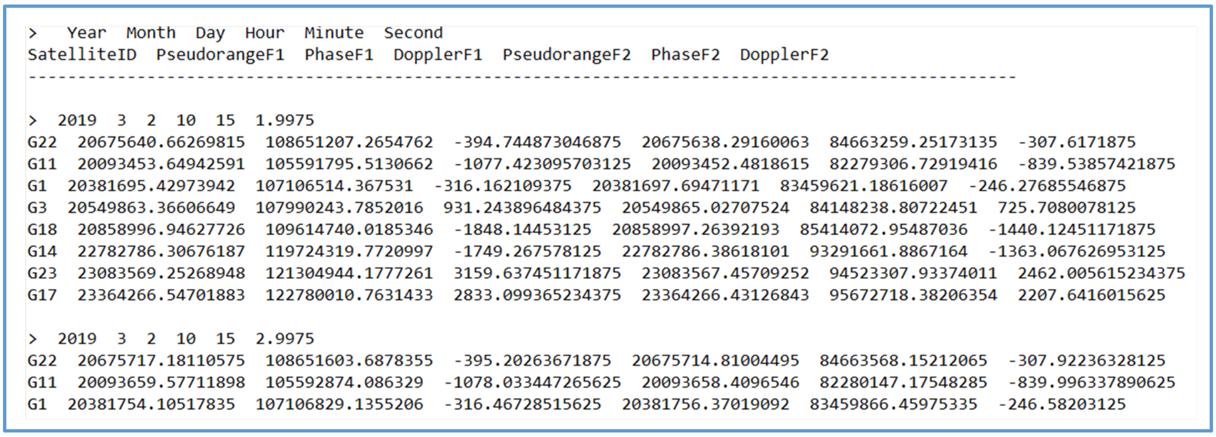
The program works with GNSS recordings of ephemerides and observables data. The different components of both inputs for the program (observables and ephemerides data) are shown in tables 4.1 and 4.2, and two examples of the files can be seen in figs. 4.1 and 4.2. From these two files, the program must produce an attitude solution at each epoch. The output varies in the case of two or three antennas. If recordings from two antennas are available, the attitude solution consists on two parameters: elevation and heading of the baseline vector. There is the possibility of introducing the body coordinates of the vector to correct for the “bias” that there may exist between the heading of the vehicle and the heading of the baseline vector. But, in presence of non-null roll or pitch, this corrected heading will not match the true heading of the vehicle unless the antennas are installed with a specific configuration section 3.1. With three antennas, the full attitude of the vehicle can be obtained. As additional output, a logger to record events has also been included. This events include cycle slips, multipath, launching of the ambiguity resolution process, etc. For a graphical representation of the results, an additional Python program has been developed, although it will not be presented here.

Observables data
Code pseudorange, frequencies L1 and L2
Carrier phase, frequencies L1 and L2
Doppler shift, frequencies L1 and L2

Table 4.1: Input observables data.

Ephemerides data
Time of ephemerides (reference epoch in week seconds)
Square root of the semi-major axis
Orbit eccentricity
Mean anomaly at reference epoch
Argument of perigee
Orbit inclination at reference epoch
Longitude of the ascending node at beginning of week
Mean motion difference
Rate of inclination angle
Rate of right ascension of node
Latitude corrections (two parameters)
Orbit radius corrections (two parameters)
Orbit inclination corrections (two parameters)
Clock parameters (offset, drift, drift rate))

Table 4.2: Input ephemerides data.

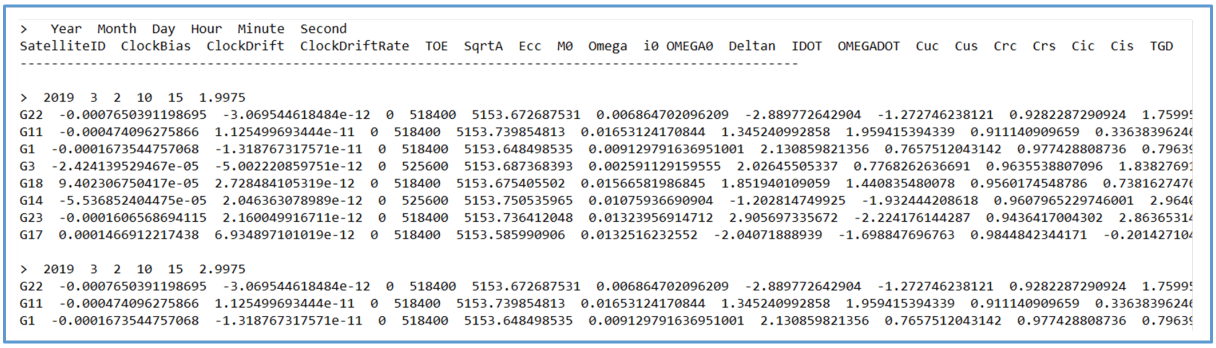


```
>   Year  Month  Day  Hour  Minute  Second
SatelliteID  PseudorangeF1  PhaseF1  DopplerF1  PseudorangeF2  PhaseF2  DopplerF2
----------------------------------------------------------------------------------------------------

>  2019  3  2  10  15  1.9975
G22  20675640.66269815  108651207.2654762  -394.744873046875  20675638.29160063  84663259.25173135  -307.6171875
G11  20093453.64942591  105591795.5130662  -1077.423095703125  20093452.4818615  82279306.72919416  -839.53857421875
G1  20381695.42973942  107106514.367531  -316.162109375  20381697.69471171  83459621.18616007  -246.27685546875
G3  20549863.36606649  107990243.7852016  931.243896484375  20549865.02707524  84148238.80722451  725.7080078125
G18  20858996.94627726  109614740.0185346  -1848.14453125  20858997.26392193  85414072.95487036  -1440.12451171875
G14  22782786.30676187  119724319.7720997  -1749.267578125  22782786.38618101  93291661.8867164  -1363.067626953125
G23  23083569.25268948  121304944.1777261  3159.637451171875  23083567.45709252  94523307.93374011  2462.005615234375
G17  23364266.54701883  122780010.7631433  2833.099365234375  23364266.43126843  95672718.38206354  2207.6416015625

>  2019  3  2  10  15  2.9975
G22  20675717.18110575  108651603.6878355  -395.20263671875  20675714.81004495  84663568.15212065  -307.92236328125
G11  20093659.57711898  105592874.086329  -1078.033447265625  20093658.4096546  82280147.17548285  -839.996337890625
G1  20381754.10517835  107106829.1355206  -316.46728515625  20381756.37019092  83459866.45975335  -246.58203125
```

Figure 4.1: Structure of the input observables data.



```
>   Year  Month  Day  Hour  Minute  Second
SatelliteID  ClockBias  ClockDrift  ClockDriftRate  TOE  SqrtA  Ecc  M0  Omega  i0 OMEGA0  Deltan  IDOT  OMEGADOT  Cuc  Cus  Crc  Crs  Cic  Cis  TGD
----------------------------------------------------------------------------------------------------

>  2019  3  2  10  15  1.9975
G22  -0.0007650391198695  -3.069544618484e-12  0  518400  5153.672687531  0.006864702096209  -2.889772642904  -1.272746238121  0.9282287290924  1.7599
G11  -0.000474096275866  1.125499693444e-11  0  518400  5153.739854813  0.01653124170844  1.345240992858  1.959415394339  0.911140909659  0.336383962
G1  -0.0001673544757068  -1.318767317571e-11  0  518400  5153.648498535  0.00912979163695100l  2.130859821356  0.7657512043142  0.977428808736  0.7963
G3  -2.424139529467e-05  -5.002220859751e-12  0  525600  5153.687368393  0.002591129159555  2.02645505337  0.7768262636691  0.9635538807096  1.8382769
G18  9.402306750417e-05  2.728484105319e-12  0  518400  5153.675405502  0.01566581986845  1.851940109059  1.440835480078  0.9560174548786  0.738162747
G14  -5.536852404475e-05  2.046363078989e-12  0  525600  5153.750535965  0.01075936690904  -1.202814749925  -1.932444208618  0.9607965229746001  2.964
G23  -0.0001606568694115  2.160049916711e-12  0  518400  5153.736412048  0.01323956914712  2.905697335672  -2.224176144287  0.9436417004302  2.8636531
G17  0.0001466912217438  6.934897101019e-12  0  518400  5153.585990906  0.0132516232552  -2.04071888939  -1.698847696763  0.9844842344171  -0.2014271

>  2019  3  2  10  15  2.9975
G22  -0.0007650391198695  -3.069544618484e-12  0  518400  5153.672687531  0.006864702096209  -2.889772642904  -1.272746238121  0.9282287290924  1.7599
G11  -0.000474096275866  1.125499693444e-11  0  518400  5153.739854813  0.01653124170844  1.345240992858  1.959415394339  0.911140909659  0.336383962
G1  -0.0001673544757068  -1.318767317571e-11  0  518400  5153.648498535  0.009129791636951001  2.130859821356  0.7657512043142  0.977428808736  0.7963
```

Figure 4.2: Structure of the input ephemerides data.

Apart from the GPS data, the application reads a set of parameters from an in put text file. These input file is read once at the beginning of the application and fixes the values of tolerances, maximum iterations, number of epochs to process, etc. The complete list of parameters is shown in table 4.3. The GPS time at the beginning of the day is necessary because the epoch present in the data files is of the form {Year, Month, Day, Hour, Minute, Second} and the program reads the epoch and calculates time only with the hour, minute and second parameters; the year, month and day equivalence in GPS time needs to be input to the program. The available methods for the calculation of attitude matrix are TRIAD and FOAM. The available methods for the float solution are single frequency code and phase, single frequency phase, and dual frequency code and phase. The fixing of the ambiguities can be done with integer rounding, bootstrapping, and integer least squares, always after applying the decorrelation adjustment from LAMBDA.

Input
Path to the folder with observables and ephemerides data
Number of antennas
Coordinates of the antennas in body frame
GPS time at the beginning of day
Measurement rate
Number of epochs to process
General
Tolerance in length for ambiguity vector rejection
Timeout for offending SV with cycle-slip
Timeout for offending SV with multipath
Method for attitude matrix calculation from vectors
Ambiguity resolution
Method for the float solution
Method for the fixed solution
Tolerance in length for solution acceptance
Maximum accumulated epochs
Variance of code measurements
Variance of phase measurements
Output
Standard or detailed output

Table 4.3: Configuration parameters for the program.

In order to match the requirements, the program needs to have modules dedicated to each of the individual problems presented in chapter 3, as well as a functioning logic to link them appropriately. These two aspects will be discussed in the following sections.

4.2. Program structure

As a c++ application, the program is structured in *classes*, with member functions and variables to better organize and solve tasks. The classes and their features are:

- `Positioning`. Class for positioning calculation. Gets code pseudoranges from `GNSS_Data` and obtains the user position and clock bias, needed for the attitude determination algorithm.

- `Attitude determination`. The main class for the attitude determination problem. Makes the calls to read and preprocess GPS data with `GNSS_Data` and passes it to the `Baseline` class. Stores the calculated attitude at each epoch. Two versions for two and three antennas.

- `Baseline`. Class with the information for one baseline (known length, body coordinates, ambiguity vector, etc). Receives the current-epoch observables and ephemerides data, accumulates it if necessary and calls `GNSS_Calculator` for the solution of the system of equations. Stores the calculated baseline vector at each epoch.

- `GNSS_Data`. Handles the observables and ephemerides data. When instructed, reads the data for the subsequent epoch, does the preprocessing (cycle slip and multipath detection, data sorting) and passes the needed observable (code, phase, L1/L2, etc) to `AttitudeDetermination` and `Positioning` classes.

- `GNSS_Calculator`. Static class that contains all the mathematical operations for solving the different systems of equations (eqs. (3.23), (3.25) and (3.27)), attitude matrix calculation methods (eqs. (3.11) to (3.14)), coordinate transformations, satellite position calculation, etc. This class needs the external library *Eigen* for matrix manipulation, least-squares adjustment, etc.

- `AmbiguityResolution`. This class contains the methods explained in section 3.3 for integer ambiguity resolution and is called from `GNSS_Calculator` to fix the solution.

- `readInput`. Class to read the input file with the parameters for the algorithm.

- `readData`. Class to read the GNSS recordings and extract the observables and ephemerides data.

- `writeOutput`. Class to write the attitude output and additional data of interest.

A flowchart representing the program's operation during one epoch is shown in fig. 4.3, where the initial configuration step is not present.

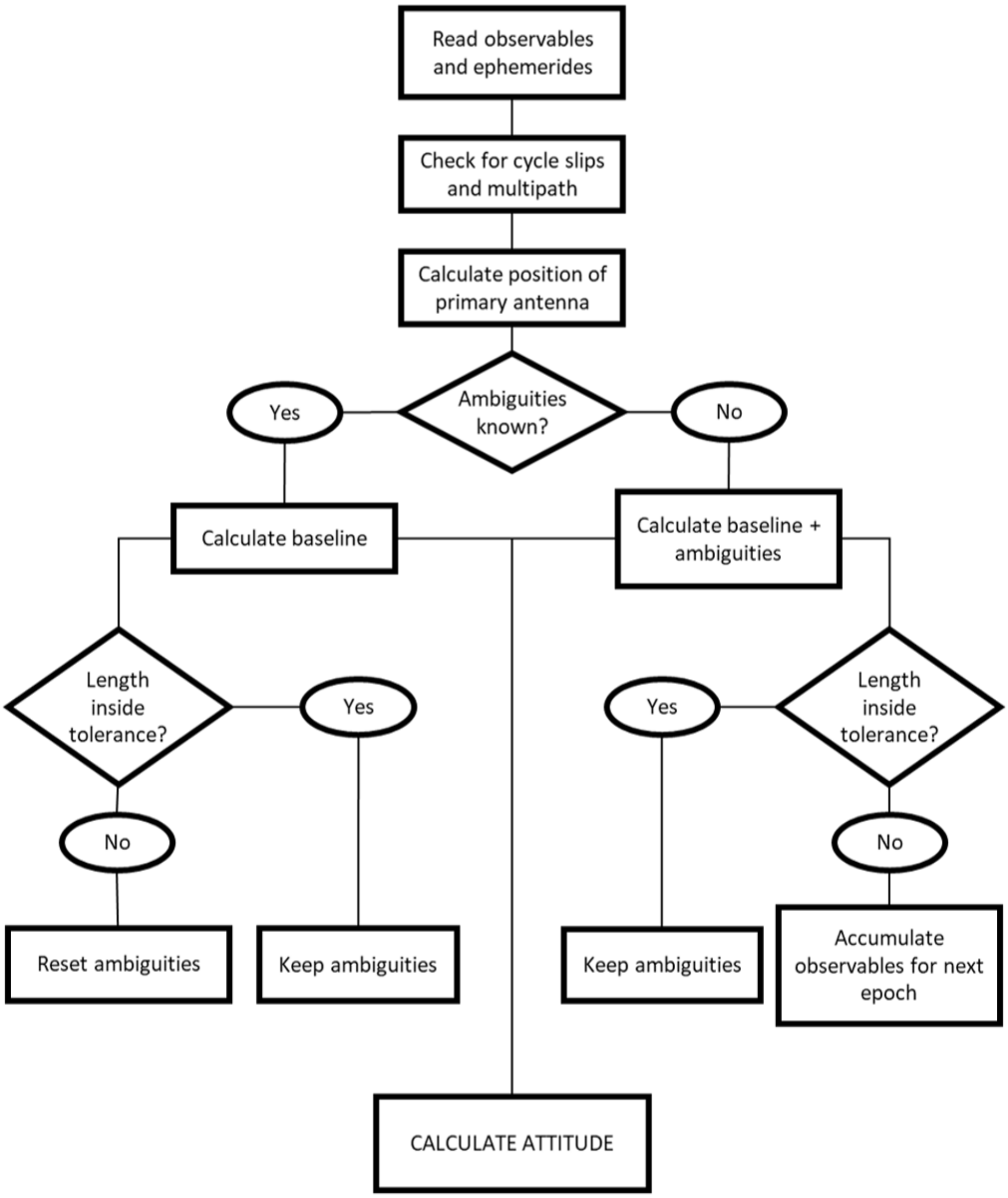

Read observables and ephemerides
Check for cycle slips and multipath
Calculate position of primary antenna
Ambiguities known?
Yes
No
Calculate baseline
Calculate baseline + ambiguities
Length inside tolerance?
Yes
No
Length inside tolerance?
Yes
No
Reset ambiguities
Keep ambiguities
Keep ambiguities
Accumulate observables for next epoch
CALCULATE ATTITUDE

Figure 4.3: Flowchart of the program.

As mentioned previously, the most important class is the `AttitudeDetermination` class, for which there are two versions; one for the calculation of elevation and heading from two antennas and another for the full attitude computation from three antennas. The dependencies of this class are shown in a graphical way in fig. 4.4, and a sequence diagram showing the calls between classes can be seen in fig. 4.5.

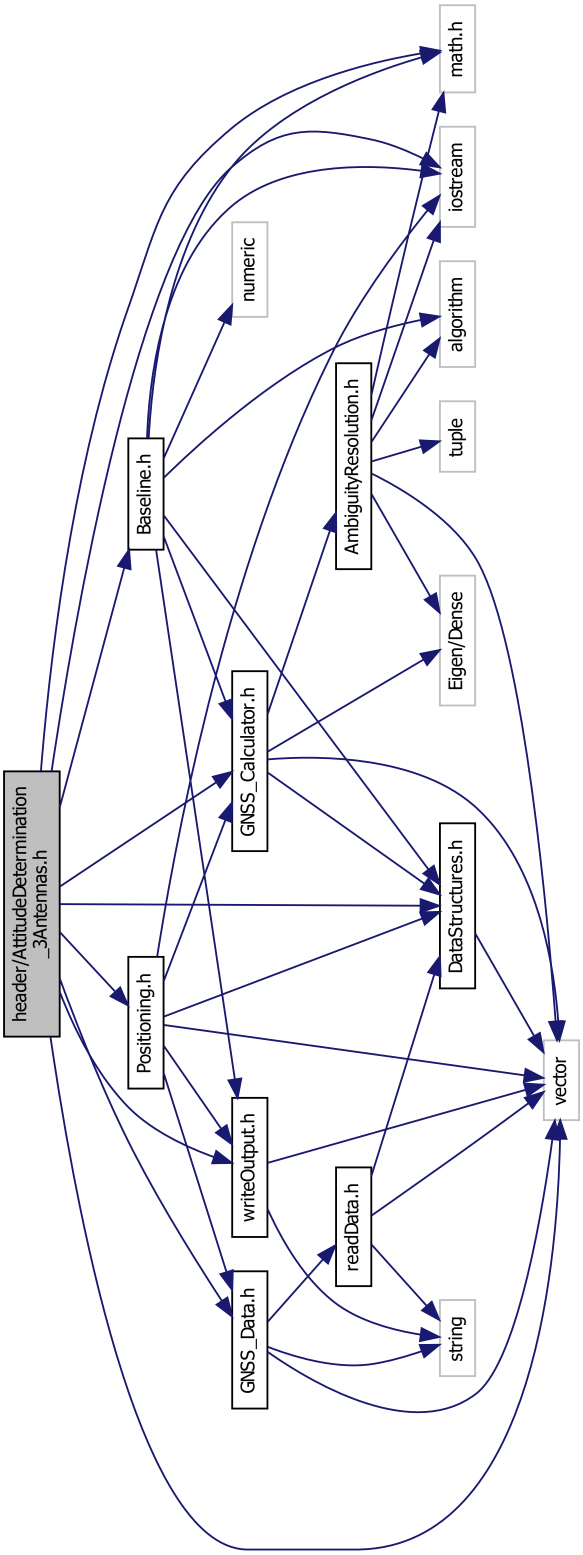


Figure 4.4: Dependency diagram for the `AttitudeDetermination` class.

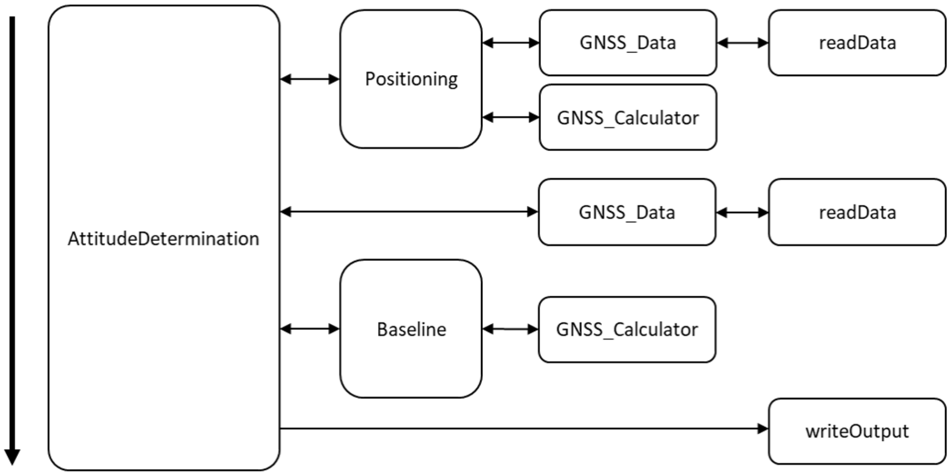


Figure 4.5: Sequence diagram for the program.

There is a bit of redundancy related to the `GNSS_Data` class, as it appears also in the class for positioning. This is because the `Positioning` class is intended to be able to do the calls for getting observables and ephemerides data on its own and work independently if needed.

4.3. Algorithms

4.3.1. Main script

The main script is called from the executable and reads the input, instantiates the `AttitudeDetermination` class and starts a `for` loop in the number of epochs to process. The script goes as follows (algorithm 4.1).

Algorithm 4.1 Main script

```
Call readInput class to read the program parameters
Pass through the program settings to the corresponding classes
for (number of epochs) do
    Calculate attitude with AttitudeDetermination
end for
```

4.3.2. Data handling and preprocessing

This feature is done by means of the dedicated `GNSS_Data` class. The function of the `GNSS_Data` class is to store the observables as a whole and then send only the useful observable to the rest of the program. The reading of the text file with the recorded data is done epoch by epoch by menas of a call to the `readData`. Apart from the member functions to read and the getters to obtain the observables, the class does also the preprocessing by means of the member functions `sortObservablesData`, `sortEphemeridesData`, `detectMultipath` and `detectCycleSlips`.

The data sorting consists on the matching between observables data from two antennas and between observables and ephemerides data. The receiver presents the SVs sorted by elevation, highest first, but the order may change, and at some epochs it may differ between recordings. The member functions dedicated take one set of observables as a reference (the one from the reference antenna) and sort the observables of the secondary antenna and ephemerides with the PRN.

4.3.3. Baseline and attitude determination

The process of attitude determination involves both the `AttitudeDetermination` and the `Baseline` classes. The logic of the attitude calculation method in the first one is shown in algorithm 4.2.

Algorithm 4.2 Attitude calculation.

```
Calculate position using Positioning class
Call GNSS_Data to read one epoch of data, do the preprocessing and to get code and
phase observables, ephemerides data and cycle slips or multipath flags
if (cycleslips are present) or (multipath is present) then
    Pass to the Baseline class(es) the corresponding cycle slip or multipath flag
end if
Call Baseline class(es) to calculate the vector(s) in ECEF frame
Get the calculated baseline(s)
Transform to navigation frame with member functions from GNSS_Calculator
Calculate elevation and heading, or full attitude, with member functions from
GNSS_Calculator
Write output with writeOutput
```

Step 6 in algorithm 4.2 calls a number of `Baseline` classes depending on the number of antennas. They store and calculate variables related to one baseline vector between 2 antennas. This means that if a network of only two antennas are used, only one `Baseline` class will be instantiated; if three antennas are used, two classes will be needed; and further on. Once the baselines have been obtained, a control step takes place to check whether the norm of the baseline coordinates lies sufficiently close to the known true length:

$$|L_{\text{calculated}} - L_{\text{true}}| \leq \epsilon_1 \tag{4.1}$$

where the value of the tolerance needs to be chosen. Regardless of the result of the control step, the attitude is obtained from the baseline coordinates in the local frame. If the FOAM method is used, the weights for the baseline vectors are obtained from their respective `Baseline` classes. These weights are function of the *state* of the baseline, in particular, they depend on the number of available satellites after the offending ones are discarded and on whether the coordinates correspond to the float or the fixed baseline solution.

From the various methods to caculate the attitude, only the chosen in the input file will be called. The structure is similar for all methods and is shown in algorithm 4.3.

Algorithm 4.3 Baseline calculation.

```
if (ambiguity vector is known) then
    if (cycle slips are present) or (multipath is present) then
        Remove data from offending satellite(s)
    end if
    Calculate baseline vector
    if (cycle slips are present) then
        Calculate unknown ambiguity components (the ones from the satellites subject
        to cycle slips)
    end if
else
    if (multipath is present) then
        Remove data from offending satellite(s)
    end if
    Calculate baseline and ambiguity vectors
end if
```

4.3.4. Ambiguity resolution

The float solution of the problem is done with methods in the classes `Baseline`, `GNSS_Calculator` and `AmbiguityResolution`. The first accumulates the observables data, times, main antenna positions, for the multi-epoch solution, and launches the float calculation with the calculator. When the float solution and its corresponding covariance matrix have been obtained, the fixing of the float solution is done with the dedicated `AmbiguityResolution` class, with structure shown in the UML diagram in fig. 4.6. The function to call is `fixFloatSolution( )` which extracts the float ambiguity vector and the baseline vector from the float solution, fixes the ambiguity with the prescribed method, and corrects the baseline float solution.

The acceptance metric for the ambiguity solution is the calculated baseline length. The equation for this is:

$$|L_{\text{calculated}} - L_{\text{true}}| \leq \epsilon_2 \tag{4.2}$$

where the value for the tolerance ϵ_2 has to be defined depending on the precision of the available measurements, and its different form the one of eq. (4.1)

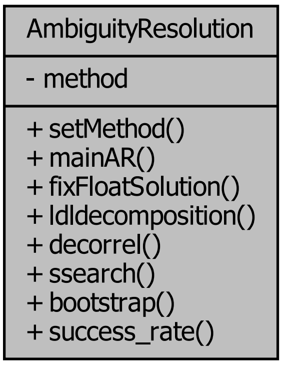


Figure 4.6: UML diagram for the `AmbiguityResolution` class.

4.3.5. Cycle slips and multipath handling

The approaches for cycle slip and multipath detection were presented in sections 3.4 and 3.5. The detection is done simultaneously, as shown in section 4.3.5. This algorithm is implemented as a member function of `GNSS_Data` and is called after the reading and storing of observables for the current epoch.

Algorithm 4.4 Cycle slip and multipath detection. `GNSS_Data` class.
.

```
for (number of satellites) do
    Check jump in the ionospheric residual
    if (jump > tolerance) then
        Cycle slip flag for the satellite
    end if
    Check difference between measured and Doppler-predicted phase observable
    if (difference > tolerance) then
        if (SV did NOT suffer a cycle slip) then
            Multipath flag for the satellite
        end if
    end if
end for
```

After the flags and the indexes of the offending SVs have been stored, they are passed to the `Baseline` class to do the appropriate manipulation of the observables when calling the solver.

In the standard case when ambiguities are known, the approach is that whenever multipath occurs, the observables relative to any offending satellite are removed for a prescribed time. If instead the incidence is a cycle slip, the observables are also removed and are instead used to calculate the unknown ambiguity component affected by the cycle slip once the baseline has been obtained from the rest of the measurements. After the prescribed time, measurements from that satellite are completely recovered and incorporated to the baseline calculation.

If multipath occurs in the middle of the ambiguity resolution process, the process is restarted without considering the measurements from the offending satellite. If it is a cycle slip, the ambiguity resolution process is simply restarted.

If cycle slips affect the pivot satellite used for the DD equations, the ambiguity resolution process has to be launched again for all satellites. If multipath affects the pivot satellite, it is removed and the ambiguity resolution process restarted; when the offending satellie is recovered, the ambiguity process is started again.

5 | Results

This chapter is dedicated to the presentation of results obtained after processing GPS recordings relative to different trajectories. Data has been generated with a GPS signal simulator and process with the program presented in chapter 4.

The first part of the analysis is relative to the validation of the program: first, ideal data with no noise, no multipath and no cycleslips will be processed; then, the application will be tested with another trajectory with incremental effects on the recordings to validate the cycle slip and multipath handling and begin assessing the effect of receiver noise. A third dataset for another trajectory will be processed with different levels of receiver noise for a comparative analysis. The last part of the chapter is dedicated to additional commentaries on the ambiguity resolution performance and the effect of the main antenna position.

5.1. Test environment

As said previously, the program has been used to process data generated with a simulator. The use of simulated data has the advantage that it can reproduce several environments and allows for a parametric analysis in different conditions, which makes this ideal for a first evaluation of the performance of the application. The simulator allows the introduction of multipath, cycle slips and user-defined levels of receiver noise, which will give the possibility to carry out the analyses present here.

The setback of using simulated data is that the program is not guaranteed to function satisfactorily in a real environment, where the real version of the simulated effects may be different.

5.2. Program validation. Attitude solution

The first part of this chapter is dedicated to the evaluation of the program and its different modules. Recordings of three trajectories will be processed to obtain the positioning and attitude solution. The first set of recordings are ideal, with no measurement error, while the second and third sets of recordings include receiver noise, multipath and cycle slips.

5.2.1. Dataset 1: ideal measurements

The first dataset used to validate the solution corresponds to a plane trajectory with null roll and pitch. The program calculates both the trajectory of the main antenna and the three Euler angles. The measurement rate is 10 Hz and the antenna coordinates in the body frame are shown in table 5.1, as well as a graphical representation of the antenna network and the definition of the baselines in fig. 5.1. The configuration of the antennas on the vehicle gives the possibility of forming baselines aligned with the x and y body axes, which is ideal as the two baselines are orthogonal. As said in previous chapters, when only one baseline is used, there exists the possibility of calculating both the angles of heading and elevation of that baseline, although they will not match any of the three Euler angles unless specific configurations. In this case, a baseline between antennas 1 and 2 would allow the calculation of true heading and roll, while a baseline between antennas 1 and 3 would allow the same for heading and pitch angle. The trajectory of the vehicle is shown in fig. 5.2, and the results for the heading angle are shown in fig. 5.3. The errors in the three attitude angles, are shown in fig. 5.4.

Antenna	x [m]	y [m]	z [m]
1	0	0	0
2	5	0	0
3	0	10	0

Table 5.1: Antenna body coordinates. Dataset 1.

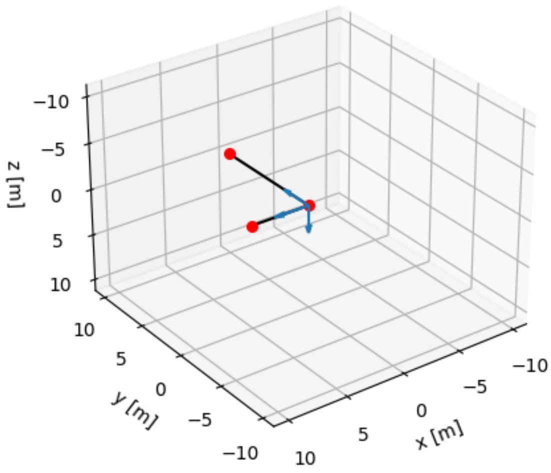


Figure 5.1: Receivers and baselines for the first dataset.

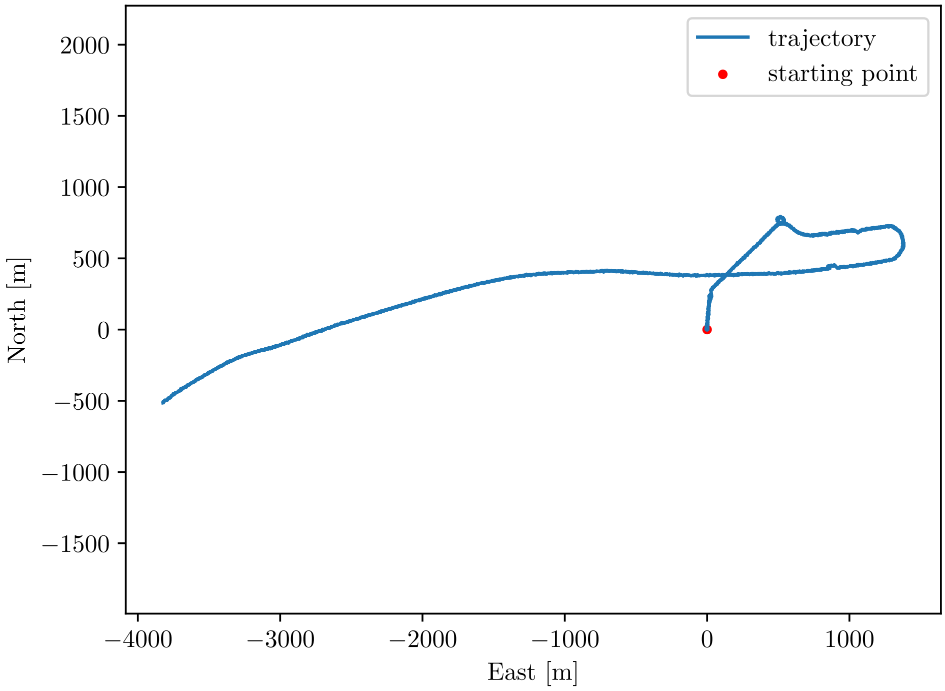


Figure 5.2: Dataset 1. Trajectory.

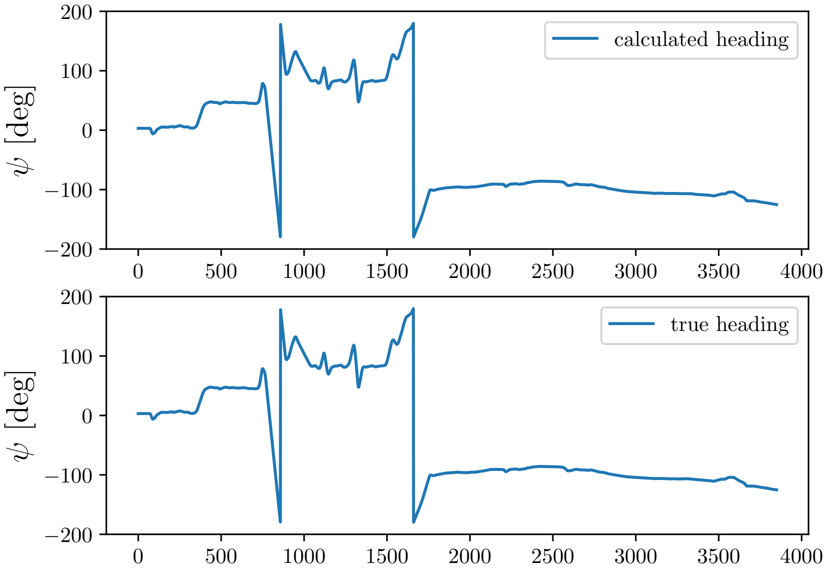
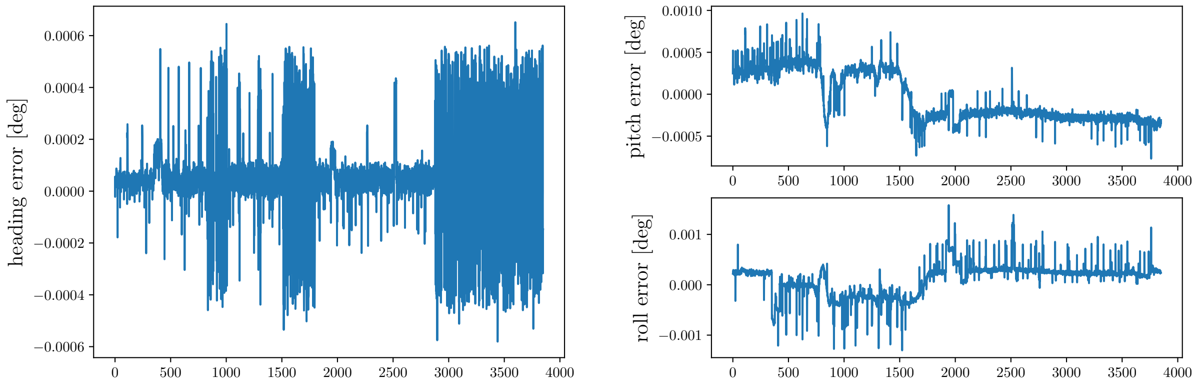


Figure 5.3: Dataset 1. Heading solution.



(a) Heading error

(b) Pitch and roll errors

Figure 5.4: Attitude solution errors.

As expected due to the absence of errors in the code and phase observables, the attitude solution is very precise in the three angles, and the time history of the heading angle matches the trajectory. The ambiguity was unknown at first but was fixed successfully at the first epoch with the LAMBDA method.

5.2.2. Dataset 2. Introduction of receiver noise, multipath, and cycleslips

The second validation analysis is targeted to the assessment of the cycle slip and multipath handling algorithms. Receiver noise is also introduced in the measurements and the measurement rate in this case is of 1 Hz. The coordinates of the three antennas are shown in table 5.2. Again, parting from the distribution of the three receivers on the vehicle, the baselines have to be defined. As said previously in this text, the accuracy of the attitude solution is proportional to the baseline length, so based on this the two baselines to form would be with antennas 1-3 and antennas 2-3. However, it is recommended to choose a network with good geometrical coverage, i.e, in which the directions of the baselines differ enough and are ideally orthogonal. With this information, the choice for the baselines is to use the pairs of antennas 1-2 and 1-3, as shown in fig. 5.5.

Antenna	x [m]	y [m]	z [m]
1	0	5	-20
2	0	-5	-20
3	10	0	-10

Table 5.2: Antenna body coordinates. Dataset 1.

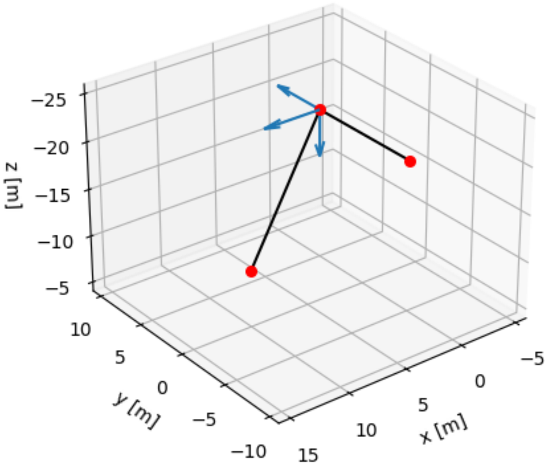


Figure 5.5: Receivers and baselines for the second dataset.

The initial and final points of the trajectory consists are [40° N, 2° E] and [40.1° N, 2.1° E] in geodetic coordinates, and the motion is carried out with a speed of 5 m/s. The trajectory is done without slip and with harmonic heading. The true solution for the heading is obtained as the sum of the course, which is 37.5° plus a sinusoidal of amplitude 20º and period of 30 seconds. The recordings correspond to a segment of this trajectory, in particular the first 250 seconds. The calculated trajectory of the vehicle is shown in fig. 5.6.

The purpose of this validation step is to study the effects of receiver noise, multipath and cycle slips; for this, the ambiguity vector will be known *a priori* in the analysis on receiver noise and multipath, and will only be calculated in the cycle slip analysis.

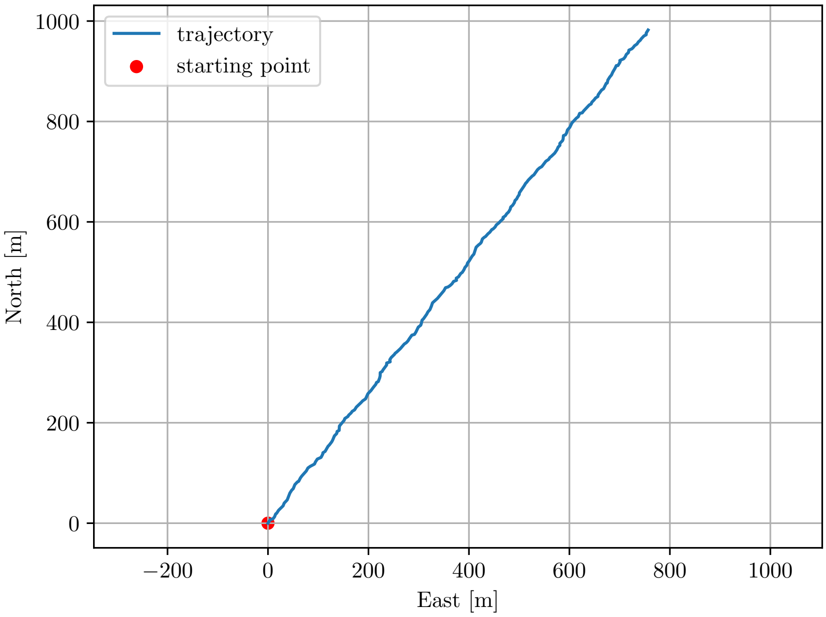


Figure 5.6: Dataset 2. Trajectory.

Receiver noise

The first effect introduced in the simulations include receiver noise, with values shown in table 5.3. This values are very high compared to those used in other attitude determination analyses present in the literature as shown in section 3.6. A discussion on receiver noise and its effect on results will be presented later in a following analysis, as the purpose of the present validation step is focused on the performance of the multipath and cycle slip handling.

σ_P [m]	σ_ϕ [m]
1.2	0.25

Table 5.3: Code and phase receiver noise. Dataset 2.

This levels of noise are very high, as after double-differencing the errors in position wouldbe around 2.5 metres for code and of 0.5 metres with phase observables. This accuracy is of various wavelengths for both L1 and L2 frequency, so the ambiguity resolution process is not expected to be successful. A further commentary on the ambiguity resolution performance with receiver noise will be presented later in this chapter. The heading solution in this first subset with only receiver noise is shown in fig. 5.7. The errors in the three Euler angles can be seen in fig. 5.8.

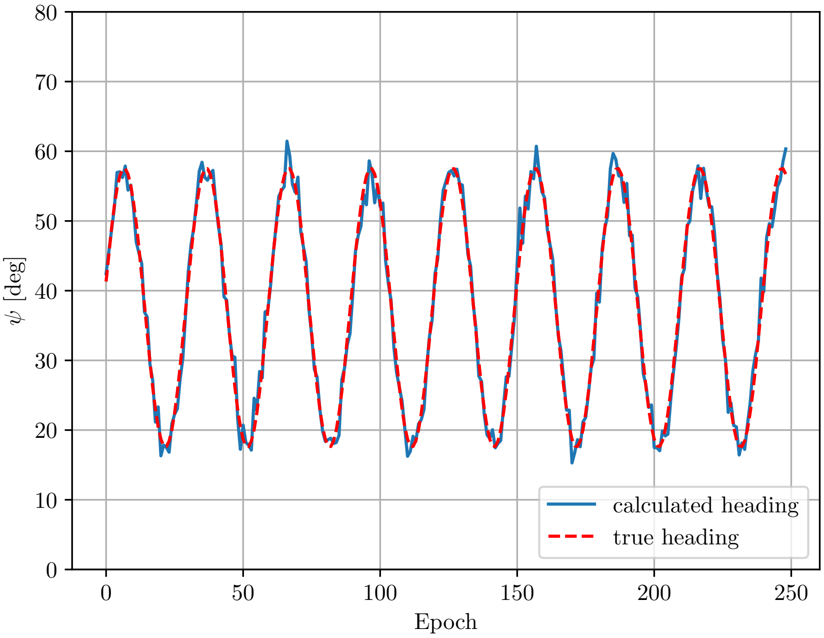
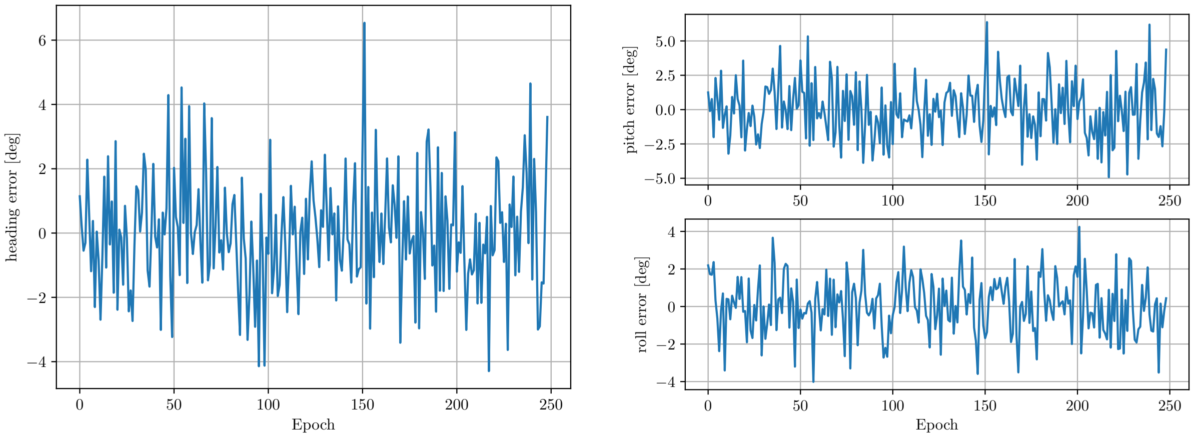


Figure 5.7: Dataset 2. Heading solution with receiver noise.



(a) Heading error

(b) Pitch and roll errors

Figure 5.8: Dataset 2. Attitude solution errors with receiver noise.

The RMS of the estimation of the Euler angles is shown in table 5.4. Results match what was expected, as with a positioning error of 0.5 metres with phase DDs, the angular resolution with a baseline of 15 metres would theoretically be $0.5/15 \simeq 0.03$ rad $\simeq 2$ degrees.

Angle	RMS [deg]
Heading	1.76
Pitch	1.49
Roll	2.07

Table 5.4: Dataset 2. Attitude errors with receiver noise..

Multipath

The second analysis has been done with this trajectory is done to check the multipath rejection algorithm presented in section 4.3.5. The receiver noise of the previous analysis is maintained. The attitue solution has been obtained first without the multipath rejection algorithm to check its influence in the calculated attitude angles; results can be seen in figs. 5.9 and 5.10. It can be seen how when multipath occurs, the solution presents important deviations. The multipath detection algorithm was based on the bias between the value for the carrier phase and a prediction done with the use of the Doppler observable. The value of this difference for this dataset is shown in fig. 5.11.

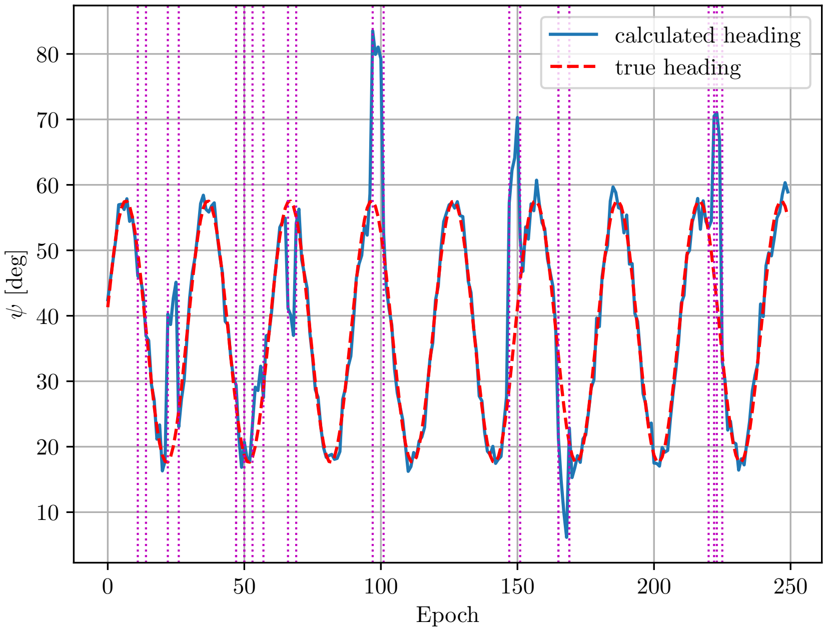


Figure 5.9: Dataset 2. Heading without multipath detection. The epochs were multipath occurs are marked with vertical lines.

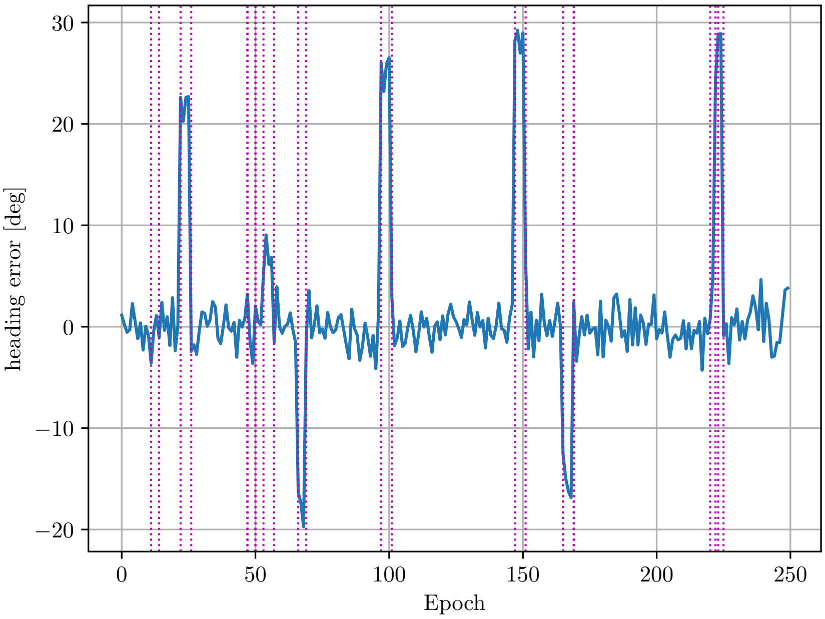


Figure 5.10: Dataset 2. Heading error without multipath detection. The epochs were multipath occurs are marked with vertical lines.

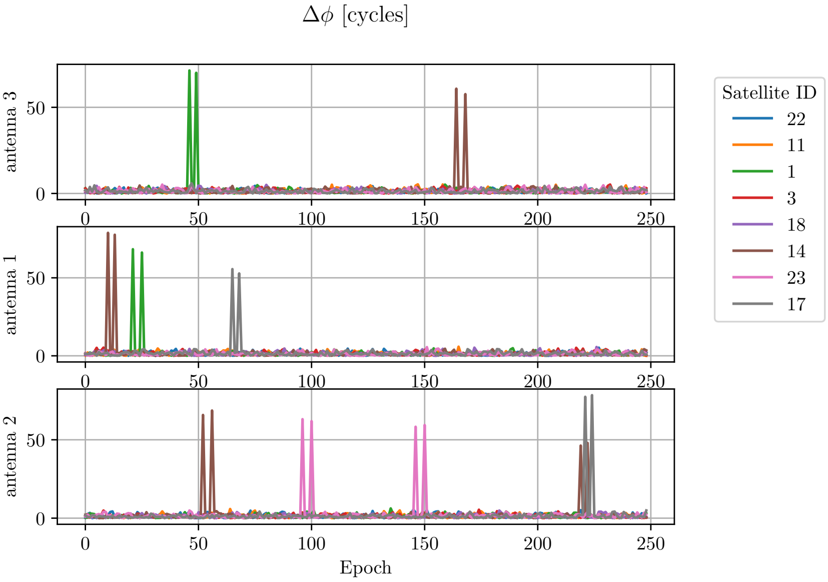


Figure 5.11: Dataset 2. Difference between measured and doppler-predicted carrier phase.

According to fig. 5.11, multipath lasts for a few seconds and then the signal returns to normal. This is of course the simulation of multipath; with real measurements, the situation would be different and the proposed simple multipath rejection algorithm may not be suitable. The solution after the observables suffering from multipath are rejected is shown in figs. 5.12 and 5.13, where the vertical lines mark the epochs where multipath has been detected. The error still increases at the epochs with multipath because there are less available satellites and therefore the geometrical resolution worsens. The numerical values for the errors are shown in table 5.5, where it can be seen that the RMS has suffered a slight increase.

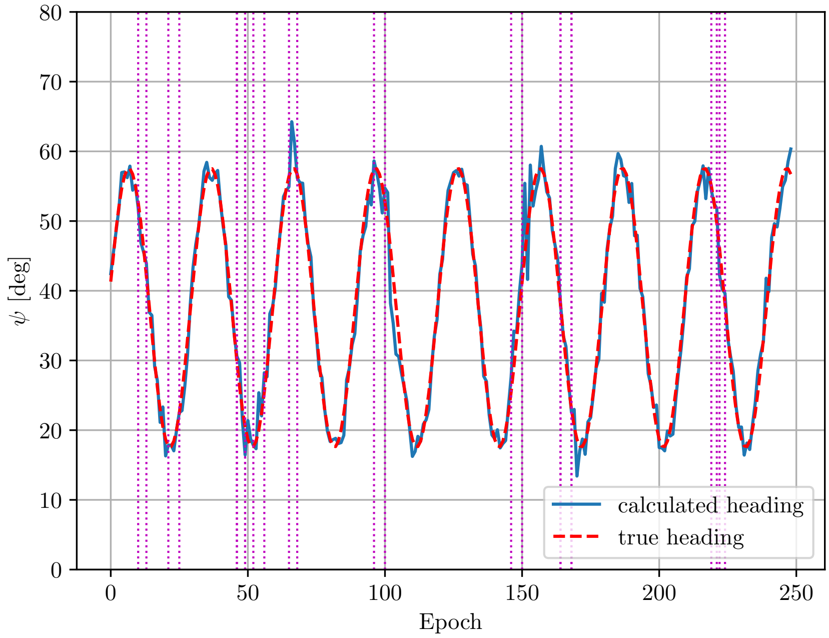
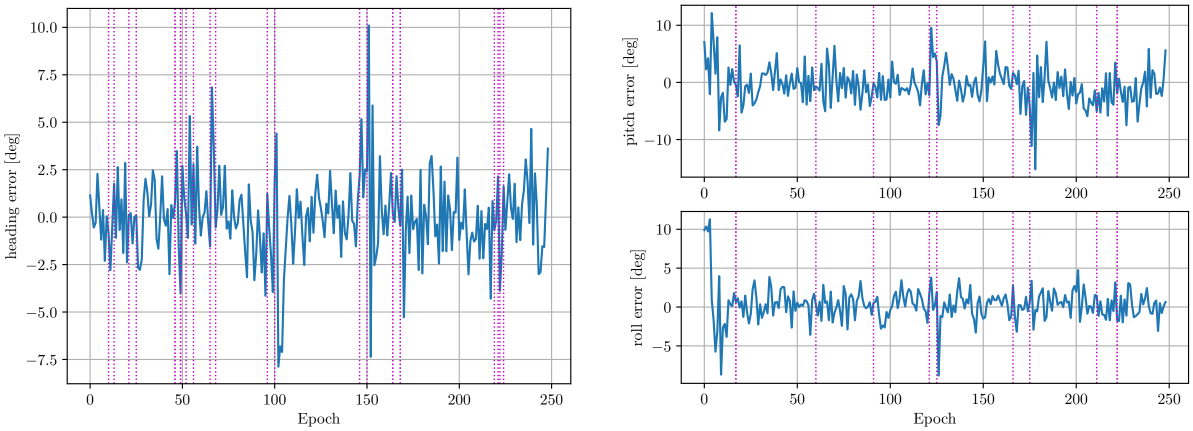


Figure 5.12: Dataset 2. Heading with multipath detection.



(a) Heading error

(b) Pitch and roll errors

Figure 5.13: Dataset 2. Attitude solution errors with multipath detection.

Angle	RMS [deg]
Heading	2.08
Pitch	1.56
Roll	2.39

Table 5.5: Dataset 2. Attitude errors with multipath.

Cycle slips

One last variation of the recordings, this time with cycle slips, has been simulated and processed with the developed application. The receiver noise parametres are again the ones shown in table 5.3. This time the ambiguity vector is not known and therefore needs to be calculated at the beginning.

First, results without the cycle slip handling algorithms are going to be shown and commented. They can be seen in figs. 5.14 and 5.15. Ambiguities are fixed after 14 epochs and, as there are not corrections for cycle slips, they remain fixed. Note that, as was previously commented, the ambiuity resolution process is very likely to fail with these levels of noise. It can be seen in the graphs how the effect of cycle slips in the attitude solution is mainly the introduction of a bias in the angles. The same effect is produced by an erroneous ambiguity vector.

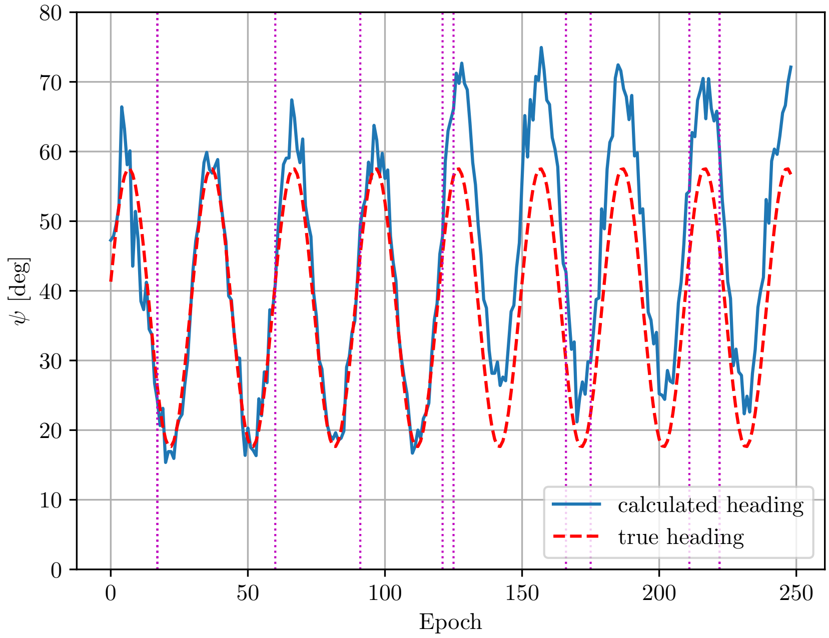
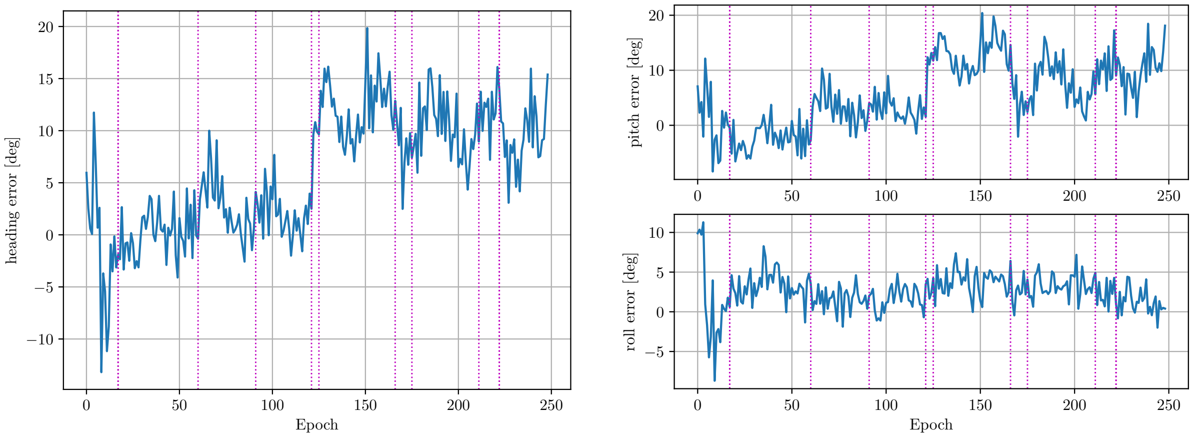


Figure 5.14: Dataset 2. Heading solution without cycle slip handling. Epochs where cycle slips occur are marked with vertical lines.



(a) Heading error

(b) Pitch and roll errors

Figure 5.15: Dataset 2. Attitude solution errors without cycle slip handling.

The cycle slip detection algorithm is based on the geometry-free combination. The evolution of this measure, shown in fig. 5.16, suffers jumps whenever a cycle slip occurs. Otherwise, the evolution in time of the combination is very slow. As expected, the most important cycle slip is associated to SV n22, which serves as pivot satellite in the construction of the double differences equations for both baselines; this explains how after epoch n $\sim$ 120 the solution deteriorates.

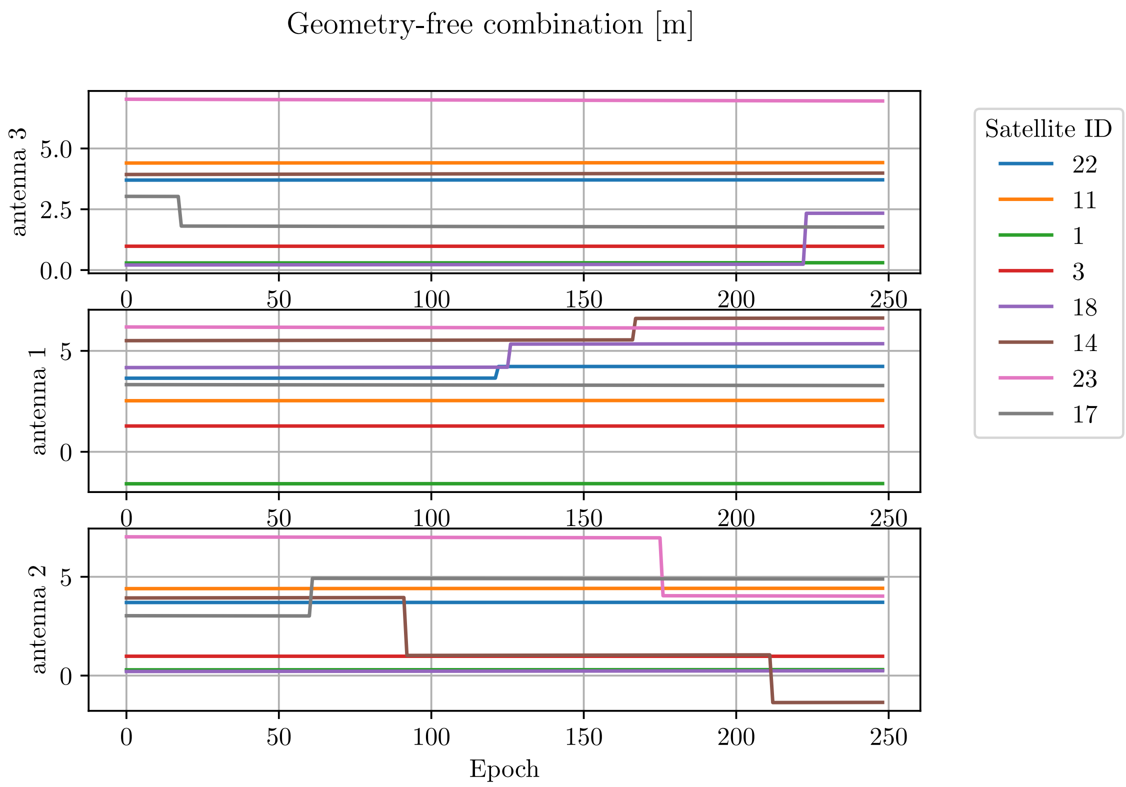


Figure 5.16: Dataset 2. Geometry-free combination.

The results with the implemented cycle slip handling algorithm are shown in figs. 5.17 and 5.18. As the ambiguites for the offending satellite are recalculated after a cycle slip, the bias that was present before has disappeared. When the cycle slip occurs with the pivot satellite n22 (epoch $\simeq$ 125), the ambiguity vector has to be fully calculated again. It can be seen how the error is bigger right after cycle slips occur. The reason for this is that the offending satellite is removed for a determined time and therefore there are less available measurements and the quality of the solution decreases. Finally, the error metrics for the solution in presence of cycle slips are shown in table 5.6. The error has grown significantly for two reasons; first, and most importantly, the ambiguities are not known at the beginning of the processing and therefore have to be calculated, probably incorrectly because of the high receiver noise; and second, because the removal of satellites slightly worsens the solution.

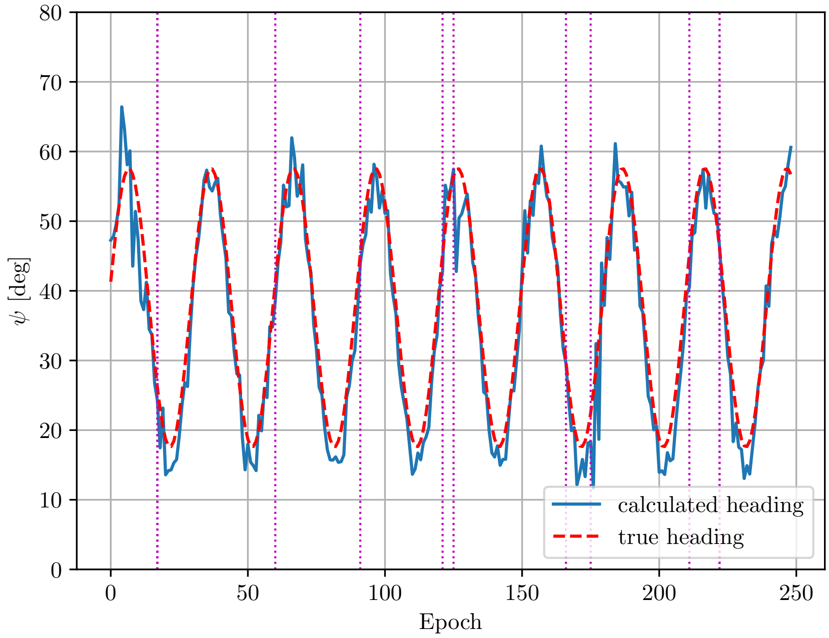
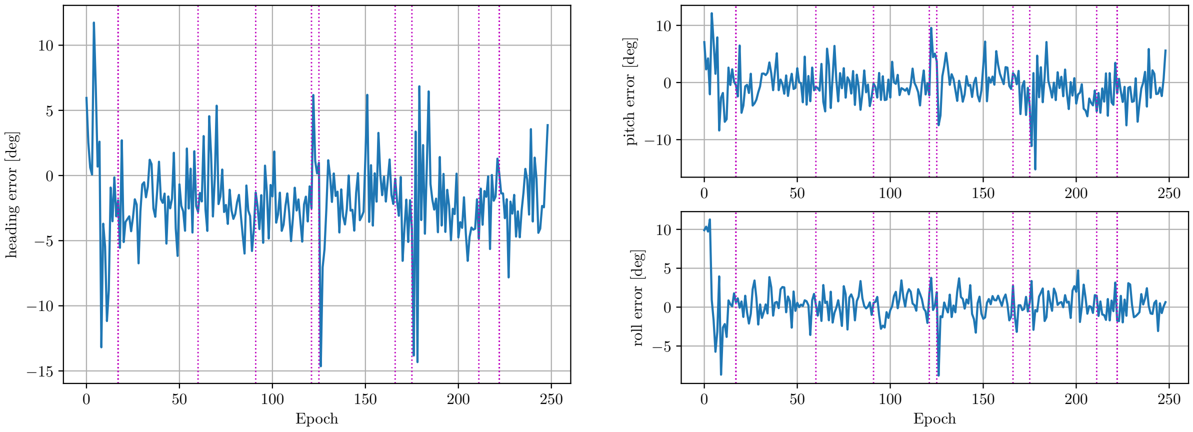


Figure 5.17: Dataset 2. Heading solution with cycle slip handling. Epochs where cycle slips occur are marked with vertical lines.



(a) Heading error

(b) Pitch and roll errors

Figure 5.18: Dataset 2. Attitude solution errors with cycle slip handling.

Angle	RMS [deg]
Heading	3.29
Pitch	2.42
Roll	4.56

Table 5.6: Dataset 2. Errors with cycle slips.

5.2.3. Dataset 3. Influence of receiver noise

A third trajectory has been simulated and processed with the program with focus on the effect of receiver noise. The different values of receiver noise used are shown in table 5.7. The three simulations contain cycle slips and multipath effects.

Subset	σ_P [m]	σ_ϕ [m]
1	0.3	0.005
2	0.6	0.03
3	1.2	0.25

Table 5.7: Dataset 3. Code and phase receiver noise

The trajectory of the vehicle is shown in fig. 5.19. Vehicle speed is 15 m/s and the total simulation time is 600 seconds, with measurement rate of 1 Hz. Heading matches the course of the vehicle and roll and pitch angles suffer a harmonic variation with amplitude 20°, period 30 second in pitch and amplitude 10° and period 10 seconds in roll.

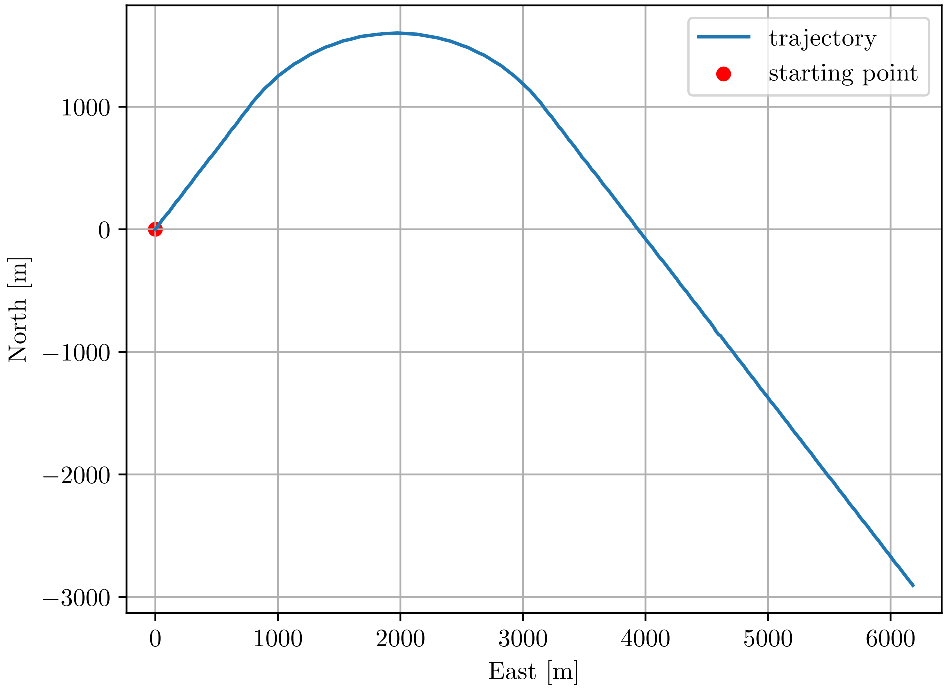


Figure 5.19: Dataset 3. Trajectory.

The analysis with the low, medium and high noise levels is presented hereunder. Note that even the medium noise is already very high compared with what is usual, according to the commentary made in section 3.6.

Subset 1. Low noise

The first simulation of this analysis has been carried out with standard deviations $\sigma_P = 0.3$ and $\sigma_\phi = 0.005$ metres for code and phase, respectively. The results are shown in fig. 5.20, the errors in fig. 5.21 and the error metrics in table 5.8. The most important deviations from the true solution happen at points where cycle slips and multipath affect two, three or more satellites, and whenever multipath occurs with the pivot satellite. The ambiguity resolution process, launched if multipath or cycle slips affect the pivot satellite or when the calculated length diverges too far from the true value, converges usually with data from one epoch with the LAMBDA method.

The results with this level of noise, which matches the typical values used in simulators in the literature, show high level of precision for the GPS attitude solution. The error could be further reduce with a more efficient cycle slip or multipath handling whenever the pivot satellite is involved. However, the ambiguity resolution process is fast and allows for a reset of the ambiguity vector whenever the quality of the solution, measured with the calculated baseline length, drifts more than the acceptable threshold.

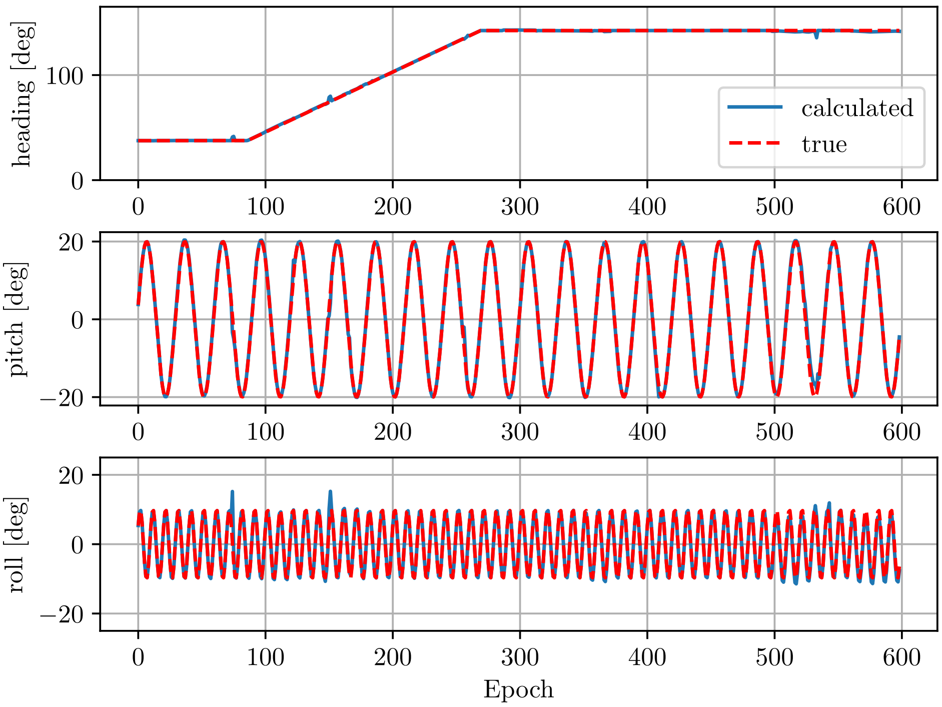


Figure 5.20: Dataset 3. Attitude results with low noise.

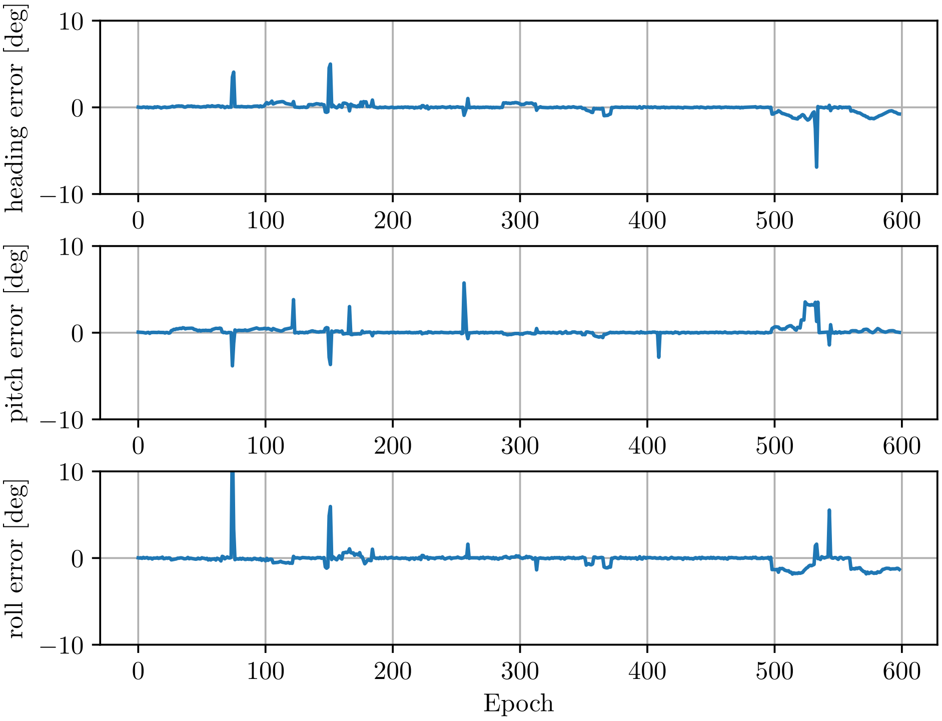


Figure 5.21: Dataset 3. Attitude errors with low noise.

Angle	RMS [deg]
Heading	0.423
Pitch	0.596
Roll	0.496

Table 5.8: Dataset 3. Errors with low noise.

Subset 2. Medium noise

Results with the second pair of values for receiver noise are shown in figs. 5.22 and 5.23. Here the ambiguity method that works better is the LAMBDA method, as the levels of noise are not very high. However, the process is not as fast as in the previous case and it takes several tries to obtain a solution with a corresponding length that lies inside the acceptance interval. Note that the wavelength corresponding to L1 frequency is $\lambda \simeq 20$ cm, so the float ambiguities with the levels of noise of this second group of measurements are likely to be away from the true ambiguities by several integers. While the ambiguity resolution process is taking place, as is the case at the first epochs of the recording, only a float solution of the attitude can be presented (see epochs 0-20 of the attitude and error graphs). The levels of noise of this analysis make the results highly dependant on the parametres chosen for the processing (table 4.3), such as the maximum accumulated epochs for the ambiguity solution and the tolerances for both the fixed solution and the maximum length deviation to declare the ambiguity vector erroneous and recalculate it.

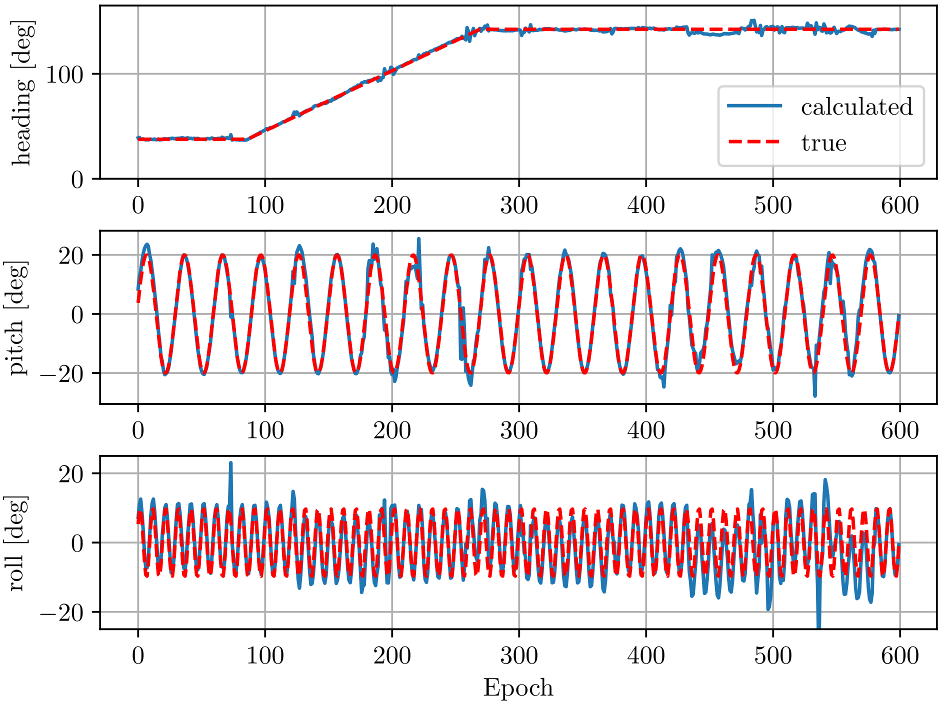


Figure 5.22: Dataset 3. Attitude with medium noise.

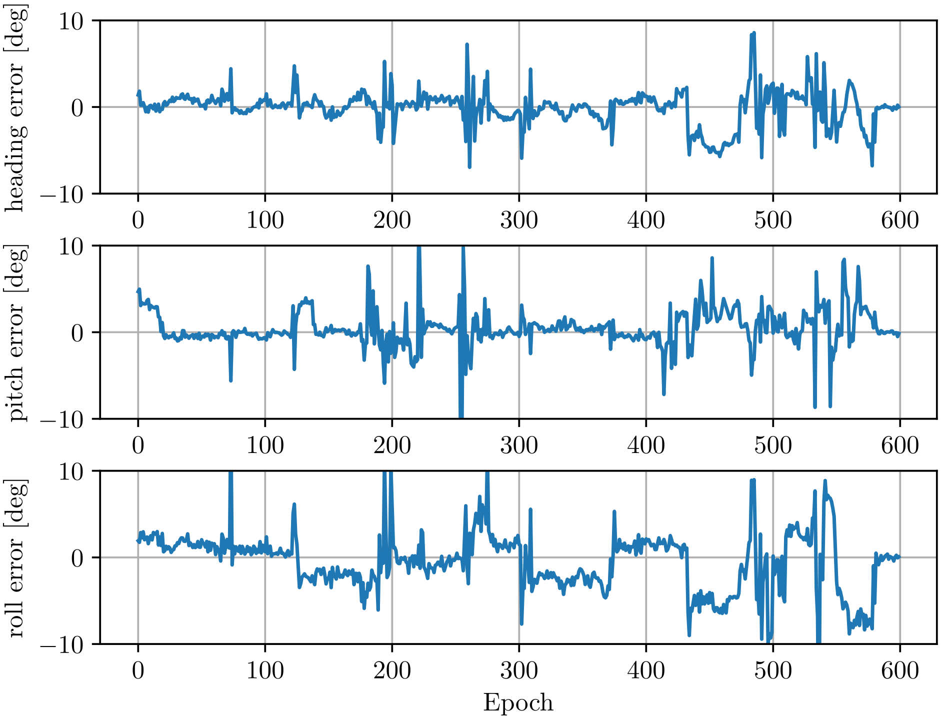


Figure 5.23: Dataset 3. Attitude errors with medium noise.

The RMS of the error for the three Euler angles is shown in table 5.9. The epochs where the error grows too much would usually be considered erroneous due to the high attitude change rate. The deviation of the solution has grown significantly in this case. Apart from the direct effect of measurement noise in the attitude solution with phase measurements, receiver noise affects the ambiguity resolution process greatly. This can be easily seen in the graphical results, where, apart from the random-like error, there exists some constant bias sustained for periods of time in the calculated attitude angles, explained by an incorrect ambiguity solution. When the solution is obtained with wrong ambiguities, the calculated angles and baseline length tend to drift over time. This makes it possible to detect, after some time, if ambiguities are wrong. However, the solution would still be degraded in the meantime.

Angle	**RMS [deg]**
Heading	2.00
Pitch	2.05
Roll	3.22

Table 5.9: Dataset 3. Errors with medium noise.

Subset 3. High noise

The final subset of recordings of this analysis on receiver noise corresponds to $\sigma_P = 1.2$ and $\sigma_\phi = 0.03$ metres for code and phase, respectively. These noise is the same as the one introduced in the simulations of Dataset 2. The calculated attitude and RMS of the error are shown in figs. 5.24 and 5.25 and table 5.10. The results present a high error, reaching 20° in some epochs, that would make the solution not suitable for a practical application. In reality, phase measurements are supposedly much more precise, so this level of receiver noise is only meaningful for a parametric analysis as this one.

It is interesting to see how the errors in the calculated roll angle are higher than those of pitch and heading. As commented in section 3.6, the longer is the baseline, the better the solution. With the present antenna geometry, roll motion is expected to have less resolution. Additionally, having one baseline aligned with the y body axis translates in a higher accuracy for the pitch angle. Also because of the geometry of the baseline between antennas 1 and 3 (pointing downwards from antenna 1, and aligned more in the direction of the x axis than the y axis), a motion in the roll axis will translate into a change in the heading of the baseline. Of course, the computation of the attitude matrix takes the orientation of the antennas into account and a change in the heading of the baseline would not necessarily mean a change in the heading of the vehicle. However, if the roll motion, calculated through the baseline coordinates, has important errors, it is likely that this lack of precision will affect the heading solution. The RMS of the attitude error can be seen in table 5.10.

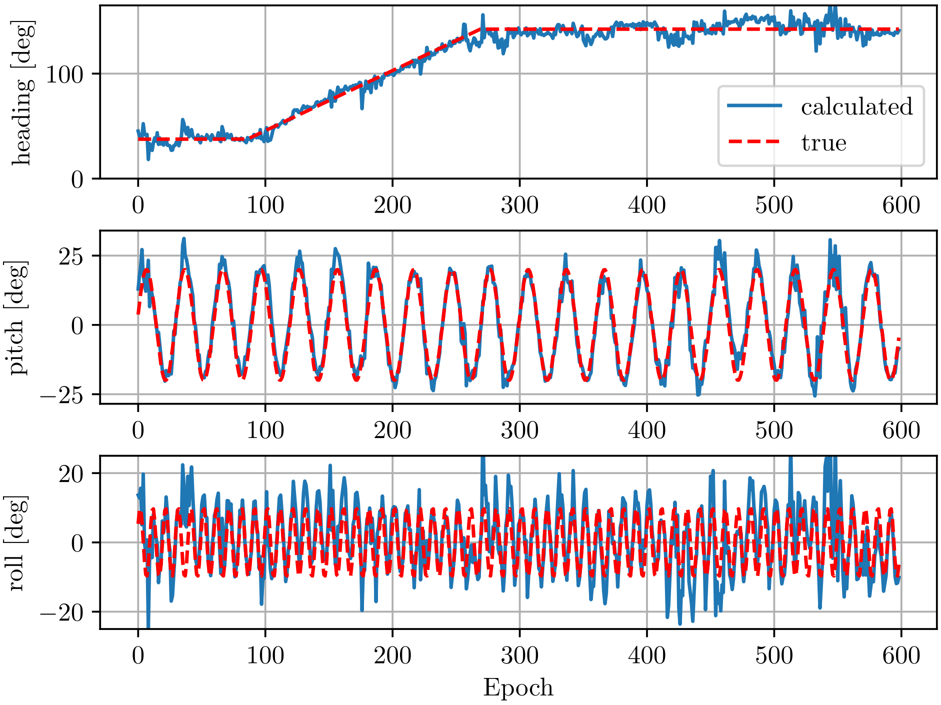


Figure 5.24: Dataset 3. Attitude with high noise.

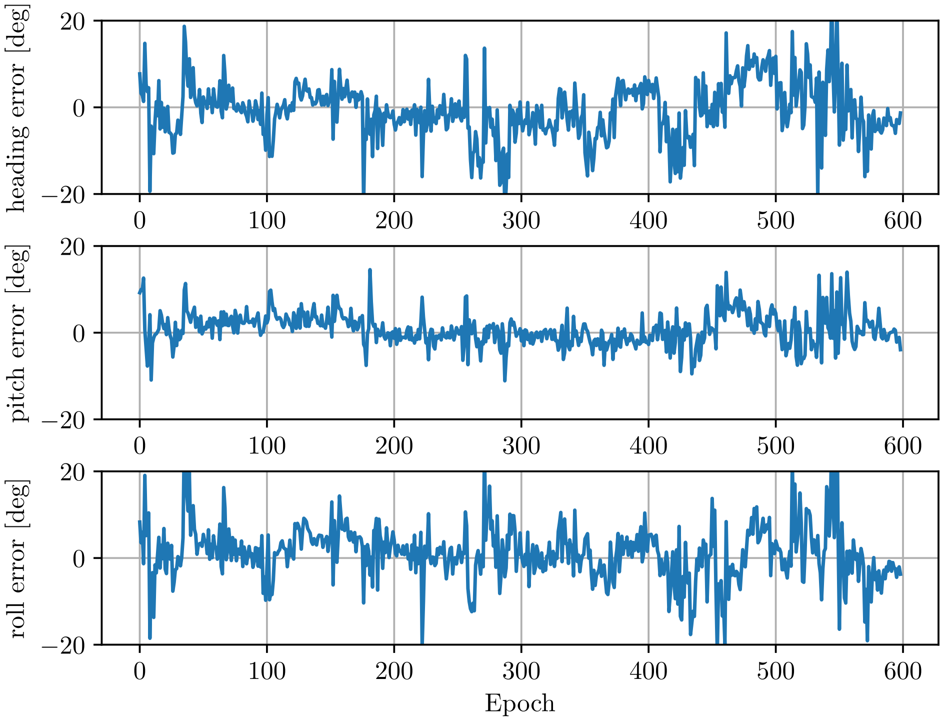


Figure 5.25: Dataset 3. Attitude errors with high noise.

Angle	RMS [deg]
Heading	2.9374
Pitch	1.933456
Roll	3.484504

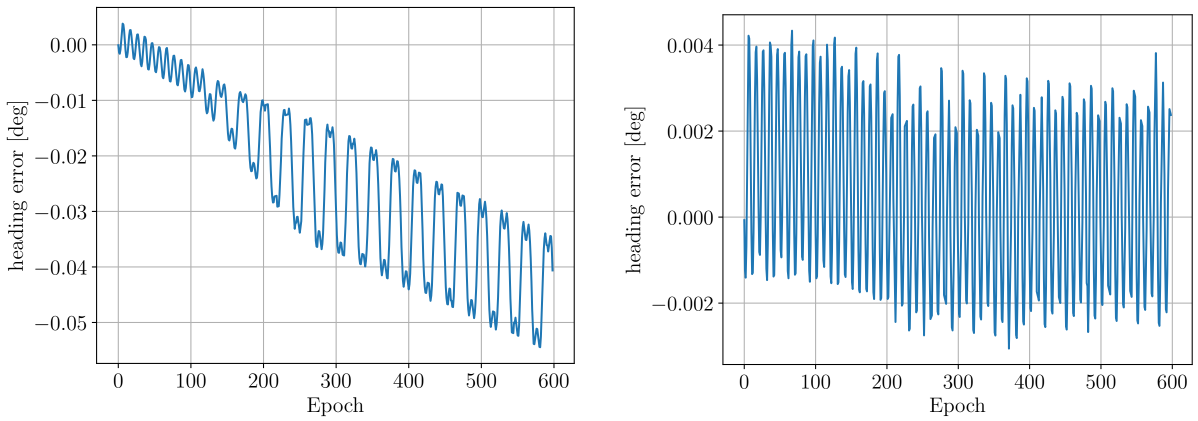
Table 5.10: Dataset 3. Errors with high noise.

5.3. Additional results

5.3.1. Effect of errors in the positioning solution

The position of the main or pivot antenna is one of the necessary data for the attitude determination problem. This information is use to construct the Line Of Sight vectors from the user to the satellites for the geometry matrices as well as for the definition of the local ENU reference frame. For the calculation of LOS vectors, the required precision is expected to be small, as the usual biases in the GNSS positioning solution, even without corrections, are negligible compared to the distance from user to satellite. For the setting of the local frame the reasoning is the same, as the error in latitude and longitude would need to be considerable in order to alter the orientation of the ENU triad.

With the purpose of demonstrating this, the attitude solution with the measurements from dataset 3 has been obtained using a constant user position equal to the position at the beginning of the simulation. The trajectory is still the same (see fig. 5.19), the only thing that changes is that the main antenna position input is constant. The measurements used have no receiver noise nor multipath nor cycle slips to isolate the effect of the main antenna position. The error in heading for the two cases (constant and changing position) is shown in fig. 5.26. The RMS of the error is shown in table 5.11.



(a) Constant position

(b) Changing position

Figure 5.26: Heading errors with different main antenna positions.

Angle	RMS [deg]
Heading	0.0285
Pitch	0.0215
Roll	0.0281

(a) Constant position

Angle	RMS [deg]
Heading	0.00202
Pitch	0.0174
Roll	0.00162

(b) Changing position

Table 5.11: Heading errors with different inputs for the main antenna positions.

Heading error tends to grow over time, as the vehicle moves away from the initial position. However, although errors are one order of magnitude higher, they are still very small even at the end, when the distance from the initial position is up to 6 km. This makes the calculation of the position from epoch to epoch unnecessary, as it could be updated on a lower rate without degrading the attitude solution. Nevertheless, this loss of precision, however small, may have an effect that has not been studied in the process of ambiguity resolution.

5.3.2. Ambiguity resolution performance

This analysis is dedicated to study the behaviour of the LAMBDA method. The calculated ambiguities with different levels of receiver noise will be compared with the true values, taking also into account the values for the success rate calculated as shown in eq. (3.44). This success rate is interpreted as a lower bound for the probability of correct ambiguity fixing with the LAMBDA method. The noise levels are those shown before in table 5.7.

The true value for the ambiguity vector is the one obtained with the lower noise level, as the corresponding baseline length matches the true one with centimetre accuracy and does not drift over time. The float solution has been obtained with single-epoch data and with five-epoch data, and with both code-carrier observables. Results are shown in tables 5.12 and 5.13. It is clear how if the float solution is obtained with data from more epochs, the success rate increases and the resulting ambiguities are closer to the true values. However, the computational time needed to solve the problem grows exponentially, as the number of equations of the system has an increasing number of equations.

Subset	Success rate	BL 1-2 ambiguities	BL 1-3 ambiguities
1	0.72	[-2, 7, 2, -7, -6, -1, 2]	[-2, 7, 2, -7, -6, -1, 2]
2	$2 \cdot 10^{-3}$	[-2, 7, 2, -7, -6, -2, 1]	[-3, 8, 0, -6, -3, -8, 0]
3	$6 \cdot 10^{-8}$	[-3, 6, 3, -8, -6, -8, -1]	[-4, 10, -0, -4, 0, -16, -2]

Table 5.12: Single epoch ambiguity resolution results. BL = Baseline.

Subset	Sucess rate	BL 1-2 ambiguities	BL 1-3 ambiguities
1	0.99	[-2, 7, 2, -7, -6, -1, 2]	[-2, 7, 2, -7, -6, -1, 2]
2	0.16	[-2, 7, 2, -7, -6, -1, 2]	[-3, 7, 1, -8,-7, -7, 0]
3	$1.6 \cdot 10^{-5}$	[-2, 7, 3, -8, -8, -3, 1]	[-2, 6, 2, -7, -11, -8, -6]

Table 5.13: Five epoch ambiguity resolution results. BL = Baseline.

6 | Conclusions

In the previous chapters the topic of attitude determination with GNSS has been presented, beginning with an exposition on the theoretical aspects and then working on a practical implementation of the technology, followed by a commentary on the results obtained. It has been proved that the GNSS technology is able to provide attitude results for vehicles with changing attitude if the conditions are not extremely non-favourable. The three main obstacles to tackle to obtain an efficient and accurate implementation of the GNSS interferometric technique, apart from successful ambiguity resolution, are receiver noise, cycle slips and multipath. Additionally, flexibility in the structure where the receivers are mounted may affect the solution, although this has not been taken into account in this project. A few remarks on what has been shown on this can be found next:

- It has been shown how receiver noise introduces a zero-mean error that can have a huge impact on the solution of the problem, especially in the aspect of ambiguity resolution. This effect can not be corrected after the receivers have been installed and therefore needs to be taken into account since the definition of the system, as the noise is one of the most important characteristics to check when looking for a commercial receiver.

- Cycle slips, and the ambiguity problem in general, are the setbacks of using carrier phase measurements to calculate the baselines. This disadvantage can not be avoided, as pseudorange measurements lack enough precision for a satisfactory attitude solution. It has been shown how the effect of cycle slips or wrong ambiguities is the drift in time of the solution, which can be controlled by monitoring the length of the calculated baseline. However, the need to include a more *active* cycle slip handling strategy has been demonstrated, as the time spent with wrong ambiguities can be long until this incidence is detected. The proposed solution for the cycle slip detection has proved to work satisfactorily in the cases processed with the application, although there exist in the literature a great variety of more complex algorithms or strategies that may provide better results. The good side of the ambiguity problem is that it is not so rigid, and the strategy for cycle slip detection and/or correction, as well as the ambiguity resolution method implemented, can always be enhanced or updated if the current solution is not satisfactory.

- The last effect considered in this project has been multipath. As shown in the results chapter, the reception of signals that present multipath effects results in big errors in the baseline coordinates and in the attitude solution. These errors last until the true non-reflected signal is acquired again. In this project, a simple method has been proposed to reject offending satellites in this aspect. However, multipath effects are very dependant on the environment of operation and on the placement of the receivers on the vehicle. Therefore, the simulated multipath effects should not be considered as a particular case and the proposed multipath mitigation method may need to be modified.

- On-the-fly ambiguity resolution is key to exploit the precision of carrier phase measurements. It has been commented how the interest in ambiguity resolution has resulted in a variety of methods developed in the literature with the objective of getting a fast and reliable solution. An implementation of the most popular method, LAMBDA, has been included in the application. It has been shown how the accuracy of the float solution, deteriorated by the effects commented earlier, is very important in the success of the ambiguity fixing step.

7 | Future work

From the beginning, the purpose of this project has been to develop an application to study the technology of GNSS attitude determination in a practical way. With this purpose, a C++ implementation of the mathematical model presented in chapter 3 has been carried out satisfactorily and tested in different conditions. On this aspect, a few suggestions can be derived:

- The program developed in this project is targeted to a real-time application. Although it has been used as a post-processing tool, its algorithms and flowchart are intended to work in real-time processing. The most important change to introduce would be to remove the `GNSS_Data` class, which stores and provides the observables and ephemerides data to the rest of the classes, with another routine that gets the data from the receivers and calls the attitude determination classes.

- Depending on the receiver quality, environment, placement of the antennas, flexibility of the structure, the effects commented previously will have stronger or weaker impact in the solution. This means that the program needs to be adapted to the characteristics of the specific situation where the attitude solution is to be obtained. The parameters of the standard deviations for code and phase, the thresholds for the acceptance of the ambiguity solution and for the rejection of the ambiguities, etc., have to be *tuned* in order to have an efficient performance and accurate solution.

Bibliography

[1] A. R. Amiri-Simkooei, S. Jazaeri, F. Zangeneh-Nejad, and J. Asgari. Role of stochastic model on gps integer ambiguity resolution success rate. *GPS solutions*, 20(1): 51–61, 2016.

[2] H. D. Black. A passive system for determining the attitude of a satellite. *AIAA journal*, 2(7):1350–1351, 1964.

[3] P. Buist. The baseline constrained lambda method for single epoch, single frequency attitude determination applications. In *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, pages 2962–2973, 2007.

[4] D. Chen and G. Lachapelle. A comparison of the fasf and least-squares search algorithms for on-the-fly ambiguity resolution. *Navigation*, 42(2):371–390, 1995.

[5] C. E. Cohen et al. Attitude determination. *Global positioning system: theory and applications*, 2, 1996.

[6] C. C. Counselman and S. A. Gourevitch. Miniature interferometer terminals for earth surveying: ambiguity and multipath with global positioning system. *IEEE Transactions on Geoscience and Remote Sensing*, (4):244–252, 1981.

[7] Z. Dai. On gps based attitude determination. 2013.

[8] C. Eling, P. Zeimetz, and H. Kuhlmann. Single-epoch ambiguity resolution for kinematic gnss positioning. In *Proceedings of the 3rd International Conference on Machine Control & Guidance, Stuttgart, Germany*, pages 27–29, 2012.

[9] U. Fernandez-Plazaola, T. Martín-Guerrero, J. Entrambasaguas-Munoz, and M. Martín-Neira. The null method applied to gnss three-carrier phase ambiguity resolution. *Journal of Geodesy*, 78(1):96–102, 2004.

[10] B. Forssell, M. Martin-Neira, and R. Harrisz. Carrier phase ambiguity resolution in gnss-2. In *Proceedings of the 10th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 1997)*, pages 1727–1736, 1997.

[11] E. Frei and G. Beutler. Rapid static positioning based on the fast ambiguity resolution approach. 1990.

[12] G. Giorgi and P. J. Teunissen. Carrier phase gnss attitude determination with the multivariate constrained lambda method. In *2010 IEEE Aerospace Conference*, pages 1–12. IEEE, 2010.

[13] G. Giorgi, P. Teunissen, S. Verhagen, and P. Buist. Improving the gnss attitude ambiguity success rate with the multivariate constrained lambda method. In *Geodesy for Planet Earth*, pages 941–948. Springer, 2012.

[14] P. D. P. D. Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. GNSS technology and application series. Artech House, Boston, second edition. edition, 2013 - 2013. ISBN 9781608070060.

[15] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[16] R. Hatch. Instantaneous ambiguity resolution. In *Kinematic systems in geodesy, surveying, and remote sensing*, pages 299–308. Springer, 1991.

[17] J. Jung, P. Enge, and B. Pervan. Optimization of cascade integer resolution with three civil gps frequencies. In *Proceedings of the 13th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 2000)*, pages 2191–2200, 2000.

[18] E. D. Kaplan and C. Hegarty. *Understanding GPS/GNSS: principles and applications*. Artech house, 2017.

[19] B. Li and Y. Shen. Global navigation satellite system ambiguity resolution with constraints from normal equations. *Journal of Surveying Engineering-asce - J SURV ENG-ASCE*, 136, 05 2010. doi: 10.1061/(ASCE)SU.1943-5428.0000017.

[20] G. Lu. *Development of a GPS multi-antenna system for attitude determination*. University of Calgary Calgary, Canada, 1995.

[21] F. L. Markley. Optimal matrix algorithm. *The Journal of the astronautical Sciences*, 41(2):261–280, 1993.

[22] F. L. Markley and D. Mortari. How to estimate attitude from vector observations. In *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, volume 103, pages 1979–1996, 1999.

[23] L. Markley. Attitude determination using two vector measurements. In *NASA Conference Publication*, pages 39–52. NASA, 1999.

[24] A. Raskaliyev, S. H. Patel, T. M. Sobh, and A. Ibrayev. Gnss-based attitude determination techniques—a comprehensive literature survey. *IEEE Access*, 8:24873–24886, 2020.

[25] P. Teunissen, P. De Jonge, and C. Tiberius. The lambda method for fast gps surveying. In *International Symposium "GPS Technology Applications" Bucharest, Romania*, 1995.

[26] P. Teunissen, P. Joosten, and C. Tiberius. A comparison of tcar, cir and lambda gnss ambiguity resolution. In *Proceedings of the 15th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 2002)*, pages 2799–2808, 2002.

[27] P. J. Teunissen. Least-squares estimation of the integer gps ambiguities. In *Invited lecture, section IV theory and methodology, IAG general meeting, Beijing, China*, pages 1–16, 1993.

[28] P. J. Teunissen. Success probability of integer gps ambiguity rounding and bootstrapping. *Journal of Geodesy*, 72(10):606–612, 1998.

[29] D. van Heesch. Doxygen. https://www.doxygen.nl, 1997.

[30] S. Verhagen. *The GNSS Integer Ambiguities: Estimation and Validation*. Publications on geodesy. NCG, Nederlandse Commissie voor Geodesie, Netherlands Geodetic Commission, 2005. ISBN 9789061322900.

[31] G. Wahba. A least squares estimate of satellite attitude. *SIAM review*, 7(3):409–409, 1965.

[32] X. Zhu, M. Ma, D. Cheng, and Z. Zhou. An optimized triad algorithm for attitude determination. *Artificial Satellites*, 52(3):41, 2017.

A | Additional equations and mathematical definitions

A.1. Calculation of satellite position

The position of the satellite is determined with the following set of equations. The input data are the ephemerides data and GPS time.

$$
\begin{aligned}
t_k &= t - t_{oe} \\
M_k &= M_0 + \left(\frac{\sqrt{\mu}}{\sqrt{a^3}}\right) t_k \\
E_k &= M_k + e \sin E_k \qquad\qquad \text{(to be solved iteratively)} \\
v_k &= \arctan \frac{\sqrt{1-e^2} \sin E_k}{\cos E_k - e} \\
u_k &= \omega + v_k + c_{uc} \cos 2(\omega + v_k) + c_{us} \sin 2(\omega + v_k) \\
r_k &= r(1 - e \cos E_k) + c_{rc} \cos 2(\omega + v_k) + c_{rs} \sin 2(\omega + v_k) \\
i_k &= i_0 + \dot{i}\ t_k + c_{ic} \cos 2(\omega + v_k) + c_{is} \sin 2(\omega + v_k) \\
\lambda_k &= \Omega_0 + (\dot{\Omega} - \omega_E)\ t_k - \omega_E\ t_{oe}
\end{aligned}
\tag{A.1}
$$

where t_k is the time at which the position need to be obtained and ω_E is the Earth's angular rotation speed.

To obtain the position in cartesian coordinates, a sequence of three rotations is applied:

$$
\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_3[-\lambda_k]\ R_1[-i_k]\ R_3[-u_k] \begin{bmatrix} r_k \\ 0 \\ 0 \end{bmatrix}
\tag{A.2}
$$

which gives

$$\begin{aligned} x &= r_k \cos(u_k) \cos(\lambda_k) - r_k \sin(u_k) \cos(i_k) \sin(\lambda_k) \\ y &= r_k \cos(u_k) \sin(\lambda_k) + r_k \sin(u_k) \cos(i_k) \cos(\lambda_k) \\ z &= r_k \sin(u_k) \sin(i_k) \end{aligned} \tag{A.3}$$

The satellite clock error is approximated with a second-order polynomial,

$$\delta t_s = a_0 + a_1 t_k + a_2 t_k^2 + \Delta t_{rel} \tag{A.4}$$

where a_0, a_1 y a_2 are the clock coefficients included in the ephemerides data 4.2 and Δt_{rel} is a correction due to relativistic effects,

$$\Delta t_{rel} = F e \sqrt{a} \sin E_k \tag{A.5}$$

here F is a constant, equal to $-4.442807633 * 10^{-10}$ s m$^{-1/2}$ and e, a y E_k are the eccentricity, the semi-major axis and the eccentric anomaly (calculated as in eq. (A.1)).

A.2. Least-squares adjustment

The method to solve the overdetermined systems present in this project (positioning algorithm, baseline solution, float ambiguity solution) is the Least-squares adjustment. The system to be solved is

$$\mathbf{A}\,\mathbf{x} = \mathbf{y} \tag{A.6}$$

A.2.1. Standard least-squares

The basic version of this methos is shown hereunder. Given the overdetermined system shown before in eq. (A.6), the solution can be obtained with the standard least-squares adjustment as:

$$\left[\mathbf{A}^\top \mathbf{A}\right] \mathbf{x}_{LS} = \mathbf{A}^\top \mathbf{y} \tag{A.7}$$

$$\mathbf{x}_{LS} = \left[\mathbf{A}^\top\,\mathbf{A}\right]^{-1}\,\mathbf{A}^\top\,\mathbf{y} \tag{A.8}$$

A.2.2. Weighted least-squares

As the system is overdetermined, the residual, is not going to be null. The standard adjustment minimizes the residual equitatively, while the weighted approach allows for a degree of *customization*. A weight matrix can be designed to prioritise the minimisation of the residuals of some equations against the rest. Given a weight matrix $\mathbf{W}$, the solution is obtained as

$$\left[\mathbf{A}^\top \mathbf{W}\,\mathbf{A}\right] \mathbf{x}_{LS} = \left[\mathbf{A}^\top \mathbf{W}\right]\, \mathbf{y} \tag{A.9}$$

$$\mathbf{x}_{LS} = \left[\mathbf{A}^\top \mathbf{W}\,\mathbf{A}\right]^{-1} \left[\mathbf{A}^\top \mathbf{W}\right] \mathbf{y} \tag{A.10}$$

A typical choice for the weight matrix is the inverse of the covariance matrix of the measurements, $\mathbf{W} = \mathbf{C}^{-1}$. In this case, the covariance matrix of the estimation can be obtained as:

$$\mathbf{C}_{LS} = \left[\mathbf{A}^\top \mathbf{W}\,\mathbf{A}\right]^{-1} \tag{A.11}$$

List of Figures

List of Tables

List of Algorithms

List of Symbols

p	—	measured pseudorange
ρ	—	distance between user and satellite
c	—	speed of light
δt_r	—	receiver r clock error
δt^s	—	satellite s clock error
I	—	ionospheric delay
T	—	tropospheric delay
k_r	—	receiver r instrumental errors
k^s	—	satellite s instrumental errors
ϵ	—	measurement errors
ϕ	—	carrier phase measured by the receiver
λ	—	signal wavelength
N	—	integer ambiguity
Δt_{rel}	—	satellite clock error due to relativistic effects
a	—	orbit semi-major axis
e	—	orbit eccentricity
E_k	—	orbit eccentric anomaly
ψ	—	heading angle
θ	—	pitch angle
ϕ	—	roll angle
$\mathbf{R_{b\rightarrow n}}$	—	rotation matrix from body frame to navigation frame
$\mathbf{A}$	—	rotation or attitude matrix
$\mathbf{b_i}$	—	vector in the body frame
$\mathbf{r_i}$	—	vector in the local navigation frame
$\mathbf{b_{ab}}$, $\mathbf{x_{ab}}$	—	baseline vector between antennas
$\mathbf{1}_a^p$	—	line of sight vector between receiver a and satellite p
$\mathbf{H}_{ab}$	—	geometry matrix of double-differences

Q	—	Covariance matrix
$\hat{a}$, $\check{a}$	—	float, fixed ambiguities
$\hat{b}$, $\check{b}$	—	float, fixed baseline
σ_P, σ_ϕ	—	standard deviation of code and phase noise
f_s	—	measurement rate
$\Phi(x)$	—	cumulative normal distribution function
S_z	—	pull-in region of an integer estimator
Z	—	transformation matrix of the LAMBDA method

List of Acronyms

AR	—	*Ambiguity Resolution*
ARF	—	*Ambiguity Resolution Function*
AFM	—	*Ambiguity Function Method*
BPSK	—	*Bi-Phase Shift Key*
CIR	—	*Cascade Integer Resolution*
DD	—	*Double Difference(d)*
DGNSS	—	*Differential Global Navigation Satellite System*
DGPS	—	*Differential Global Positioning System*
ECEF	—	*Earth Centered, Earth Fixed*
EGNOS	—	*European Geostationary Navigation Overlay Service*
ENU	—	*East North Up*
FARA	—	*Fast Ambiguity Resolution Approach*
FASF	—	*Fast Ambiguity Search Filter*
FOAM	—	*Fast Optimal Attitude Matrix*
GBAS	—	*Ground Based Augmentation System*
GLONASS	—	*Globalnaya Navigazionnaya Sputnikovaya Sistema, or Global Navigation Satellite System*
GNSS	—	*Global Navigation Satellite System*
GPS	—	*Global Positioning System*
IB	—	*Integer Bootstrapping*
ILS	—	*Integer Least-Squares*
IR	—	*Integer Rounding*
LAMBDA	—	*Least-squares AMBiguity Decorrelation Adjustment*
LSAST	—	*Least Squares Ambiguity Search Technique*
LOS	—	*Line Of Sight*
MEO	—	*Medium Earth Orbit*
PRN	—	*Pseudo-Random Noise*

PVT	—	*Position, Velocity and Time*
RMS	—	*Root Mean Square*
SBAS	—	*Satellite Based Augmentation System*
SD	—	*Single Difference(d)*
SV	—	*Space Vehicle*
TCAR	—	*Three Carrier Ambiguity Resolution*
TRIAD	—	*TRIaxial Attitude Determination*
WAAS	—	*Wide Area Augmentation System*