UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Contributions to high accuracy snapshot GNSS positioning

Xiao Liu

Ph.D Thesis

Aerospace Science and Technology Doctoral Program

Contributions to High Accuracy Snapshot GNSS Positioning

Xiao Liu

Advisors:

Prof. Dr. Jaume Sanz Subirana and Dr. Adria Rovira-Garcia

Research group of Astronomy and Geomatics

Department of Mathematics and Department of Physics

Universitat Politècnica de Catalunya

Dr. Adrià Gusi-Amigó

Albora Technologies

September 18, 2022

Barcelona, Spain

To my family and friends

Acknowledgments

Firstly, I would like to express my sincere gratitude to my academic supervisors Prof. Jaume Sanz Subirana and Dr. Adrià Rovira-Garcia. This thesis would not have been possible were it not for their insightful comments and suggestions. This PhD has been an enjoyable journey thanks to their continuous assistance at every stage of this research project.

The same gratitude goes to my industrial mentors Dr. Miguel Angel Ribot Sanfelix and Dr. Adrià Gusi-Amigó for all the technical discussions and communications of ideas. Their immense knowledge and experience in the GNSS industry has inspired me to keep exploring new algorithms.

I am also grateful to my colleagues at Albora Technologies, cooperating with them really made my job a lot easier. The SRTK software library was developed with a great amount of their effort. I enjoyed every moment of this industrial experience thanks to their expertise and professionalism. Particularly, great thanks to Prof. Pau Closas for his patience in co-authoring the academic papers before this dissertation.

I would also like to thank the help and accompany provided by the fellow researchers at group of Astronomy and GEomatics (gAGE), especially to Cristhian, who has always been there offering help even before I started this PhD.

In addition, I would like to thank all my friends in Barcelona who made this city my home. Especially to Xinjie Guo, thank you for your help in life and supporting me during the covid pandemic; and to Julieta, I really appreciate your confidence in me and the continuous encouragements.

I would like to acknowledge Albora Technologies for also providing the hardware for snapshot data collection. I would like to thank Everis Aeroespacial y Defensa, S.L.U. for lending the Septentrio receiver and antenna for this research. Equally, I would like to acknowledge the Institut Cartogràfic i Geològic de Catalunya for their support by enabling

multiple simultaneous Virtual Reference Station stream connections.

This research was funded by Albora Technologies and Universitat Politècnica de Catalunya with the industrial PhD grant (number DI 082) from the Generalitat de Catalunya. This research would not have been possible without their support. This research was partially funded by the Spanish Ministry of Science and Innovation project RTI2018-094295-B-I00.

Finally, I would like to express my deepest gratitude to my family. To my parents, thank you for the unconditional support and trust, it was never easy but you have sacrificed so much just to earn me the opportunity to chase my dream, your endless support will be forever appreciated. To my uncle and aunt, thank you for accepting me and patiently nurturing me with the best educational resources, those years changed my life.

Table of Contents

List of Tables

List of Figures

Abstract

Snapshot positioning is the technique to determine the position of a Global Navigation Satellite System (GNSS) receiver using only a very brief interval of the received satellite signal. In recent years, this technique has received a great amount of attention in the GNSS community thanks to its unique advantages in power efficiency, Time To First Fix (TTFF) and economic costs for deployment. However, the state of the art algorithms regarding snapshot positioning were based on code measurements only, which unavoidably limited the positioning accuracy to meter level. The present PhD research aims at achieving high-accuracy (centimetre level) snapshot positioning by properly utilizing carrier phase measurements. Two technical challenges should be tackled before such level of accuracy can be achieved, namely, inaccurate estimation of satellite transmission time and the so-called Data Bit Ambiguity (DBA) issue.

The first challenge is essentially originated from the lack of absolute timing accuracy in the receiver, as only the coarse time information is available from an external assistance module and its error can be up to a few seconds. Applying a conventional Coarse Time Filter (CTF) can increase this timing accuracy to the millisecond level. However, this is still not enough for carrier-phase based positioning since the satellite position errors introduced by such timing errors range up to one meter, which certainly impedes the carrier phase Integer Ambiguity Resolution (IAR). A method is proposed to set a global time tag and correspondingly construct the pseudoranges with full period corrections. This method guarantees the satellite transmission times are accurate to the nanosecond level. Thus, the corresponding accurate satellite coordinates enable the high accuracy snapshot navigation filter.

The second challenge is caused by the fact that snapshot measurements are generated based on the results of the correlation between the received signal and the local replicas. Multiple replicas are typically produced in snapshot positioning following the Multi Hypothesis (MH) acquisition architecture. It may happen that more than one local replica

(i.e. hypothesis) result in the maximum correlation energy. Hence, we need to identify the actual secondary codes or data bit symbols encoded in the received signal, i.e. to resolve the DBA. Particularly, when the local replica is generated with exactly opposite symbols to the actual ones, the resulting carrier phase measurement contains a Half Cycle Error (HCE) and impedes also the IAR step. A method has been proposed in this PhD to resolve the DBA issue for pilot signals with encoded secondary codes. This method attempts to form a consensus among all satellites regarding their secondary codes, under the assistance of the flight time differences, which are available as a side product of the conventional CTF. A different approach has been developed for data signals. It amends the carrier phase HCEs one after another by an iterative satellite inclusion procedure. This approach uses the Real Time Kinematics (RTK) LAMBDA Ratio Factor (LRF) as an indicator to evaluate the potential existence of the HCEs.

The present PhD focuses on implementing the so-called Snapshot RTK (SRTK) technique. As in the classic RTK technique, SRTK cancels most of the measurement errors through the Double-Differenced (DD) process. The workflow details of SRTK are explained incorporating the aforementioned new algorithms. Several experiments were performed based on real world signal recordings and the results confirm the feasibility of obtaining SRTK fix solutions. The performance of SRTK is numerically demonstrated under different parameters of signal bandwidth, integration time and baseline distance. The SRTK fix rates can reach more than 90% in most of the scenarios, with centimetre-level positioning errors observed in the fixed solutions.

It can be concluded that upon the implementation of the global time tag method, high accuracy snapshot positioning becomes feasible with the SRTK technique and its performance varies depending on the SRTK configuration. The algorithms developed for the DBA issue and carrier phase HCEs also prove to effectively improve the performance of SRTK.

Chapter 1

Introduction

1.1 GNSS Background

The term Global Navigation Satellite System (GNSS) is defined as the collection of all Satellite Navigation (SATNAV) systems and their augmentations [1]. Currently, there exist four GNSSs with global coverage: the U.S. Global Positioning System (GPS), the European Galileo system, the Chinese BeiDou Navigation Satellite System (BDS) and the Russian Federation GLObal NAvigation Satellite System (GLONASS). these GNSSs are sometimes referred to as core constellations. Other systems provide navigation services to a dedicated region only, such as India's Navigation with Indian Constellation (NAVIC) and Japan's Quasi-Zenith Satellite System (QZSS).

1.1.1 GNSS Segments

A GNSS consists of the following three main segments:

1. the **space segment**, which contains all the satellite vehicles that are continuously transmitting navigation signals. These signals include the ranging codes and navigation messages that are modulated to carrier sinusoidal signal at a given frequency. The navigation message contains the necessary elements, such as the ephemeris data that are required to compute satellite position, among many other parameters.

2. the **control segment**, which consists of a global network of ground facilities that track the GPS satellites, monitor their transmissions, perform analyses, and send commands and data to the satellites in the constellation.

3. the **user segment** that is composed of the GNSS receivers [2]. These devices operate passively (i.e., receive only) by processing the received GNSS signals and providing estimates of Position Velocity and Time (PVT) information to the user.

The space and control segments are mainly maintained by the constellation owners and not accessible to the general public. In contrast, the user segment is mainly controlled by the GNSS receiver manufacturers and it is the main focus of the present work.

1.1.2 GNSS Market

Many fundamental technologies that are being developed today rely on the continuous service of GNSS. The GNSS market report [3] published by the Europe Union Space Program Agency (EUSPA) listed the market segments where GNSS could potentially play an important role, including consumer solution devices, road transportation and automotive, manned aviation, drones, maritime, rail, agriculture, among many others. Each segment contains a great variety of applications. For example, applications of consumer solution devices include personal tracking devices, wearables, digital cameras, etc. Other advanced technologies such 5G, Artificial Intelligence (AI), Internet of Things (IoT) are being used in combination with GNSS to better serve the day-to-day life of end users.

Snapshot positioning is the technique to determine the position of a Global Navigation Satellite System (GNSS) receiver using only a very brief interval of the received satellite signal. The high accuracy snapshot positioning technique presented in this work mainly concerns the IoT related applications of the consumer solution devices segment. As shown in the market report, the user needs in this sector are firmly increasing over recent years. Numerically, the Compounded Annual Growth Rate (CAGR) of personal tracking devices

and Low-Power Asset Tracking devices are estimated to be 23.7% and 24.2% respectively. This growth trend is expected to continue until the year 2029.

1.1.3 Basic Principles

From the perspective of the user segment, the basic principle behind GNSS positioning is a *biased passive trilateration* process [4]. Such definition involves three key terms. First, the term "passive" is used since the positioning device only receives the GNSS signal. The collected signal is then processed to generate a time delay measurement that can be further converted to distance measurements (i.e., range) by multiplying the speed of light. To be more precise, this measurement is formed based on the time difference of arrival of synchronized signals. Because these signals are delayed or advanced with respect to the true geometric distances between the transmitting satellites and the receiver, we refer to them as pseudoranges, as we will see in Chapter 2. With more satellite signals been processed, more pseudorange measurements can be obtained. Secondly, the term "trilateration" is the core operation in the positioning algorithm. This operation makes use of the pseudorange measurements as the main input and estimates the user coordinates based on the differences of its distances from the known satellite positions. Finally, the term "biased" is used due to the fact that all pseudorange measurements share a common bias caused by the inaccuracy of the receiver clock.

As for the space and control segment, they have their own responsibilities to maintain the system operational according to the expected performances requirements. As already mentioned, in order to ensure that the user segment functions properly, the satellites in the constellation must have clocks that are synchronized to a common time with great accuracy. This is guaranteed by the use of high accuracy atomic clocks onboard the space vehicles, and correction parameters containing the bias, drift and acceleration about these satellite clocks are distributed timely. Besides, in order to accurately estimate the user coordinates, it is necessary to have a precise knowledge about the satellite coordinates at the time of

signal transmission. For this purpose, satellite orbit parameters, known as the ephemeris parameters, are continuously being monitored and provided to the users by the so-called navigation messages that are modulated to the carrier signal. These fundamental data, sometimes along with other more precise orbit and clock products, are also distributed by other communication channels such as internet.

1.1.4 GNSS Performance Parameters

Some parameters are used to evaluate the GNSS performance, the most typical ones are as follows [3]:

1. Availability, the percentage of time the Position Navigation or Timing (PNT) solution that can be computed by the user. Two types of availability are usually considered: the system availability that is determined by the GNSS Interface Control Documents (ICD) and the overall availability that takes into account the receiver performance and the user's environment.

2. Accuracy, the difference between true and computed solution. As formulated in [2], the following different types of accuracy measures are usually used in GNSS:

   (a) Formal accuracy: denoted as the diagonal terms of the covariance matrix of the navigation solution. It provides a characterisation of the quality and uncertainty of the coordinates and time estimates, instead of their actual errors.

   (b) Predicted accuracy: typically represented by the Dilution Of Precision (DOP) and the variance of the GNSS measurements. The DOP is only linked to the satellite geometry and represents a ratio factor between the measurement precision and that in positioning.

   (c) Measured accuracy: the actual navigation solution errors. This accuracy measure is computed by comparison of the obtained estimated to a set of ground truth data.

3. Continuity, the ability of a system to perform its function (deliver PNT services with the required performance levels [5]) without interruption once the operation has started.

4. Integrity, the ability of the system to provide warnings to users when it should not be used. It is the probability of a user being exposed to an error larger than the alert limits without timely warning

Other parameters include: Robustness, Authentication, Time To First Fix (TTFF), Power consumption, Resiliency, etc. Among them, the most valued ones for snapshot GNSS positioning, and as is for the present work, are the accuracy, TTFF, and power consumption.

1.2 Snapshot Positioning

Snapshot positioning is defined as the technique to determine the position of a GNSS receiver using only a very brief interval of the received satellite signal, where the sampling time ranges from 2 to 100 milliseconds (ms) [6]. This technique is also referred to as the one-shot GNSS receiver [7]. Compared to a conventional GNSS receiver that may require tens of seconds of signal tracking before a first position can be computed, snapshot positioning technique shows more advantages in applications that require timely positioning results or when energy consumption is significantly limited.

As pointed out in the IoT white paper published by EUSPA [8], traditional GNSS technologies do not align well with the stringent constraints of battery-powered IoT devices which are often expected to function intermittently for multiple years without charging. Multiple techniques were developed aiming at tackling this issue, including receiver duty cycling, ephemeris extension and prediction, Assisted GNSS (AGNSS), etc. While these techniques already contributed to the reduction of power consumption in the receiver, snapshot positioning technique takes a further step forward as it determines the position by using only a minuscule interval of a GNSS signal. Furthermore, snapshot positioning technique fits

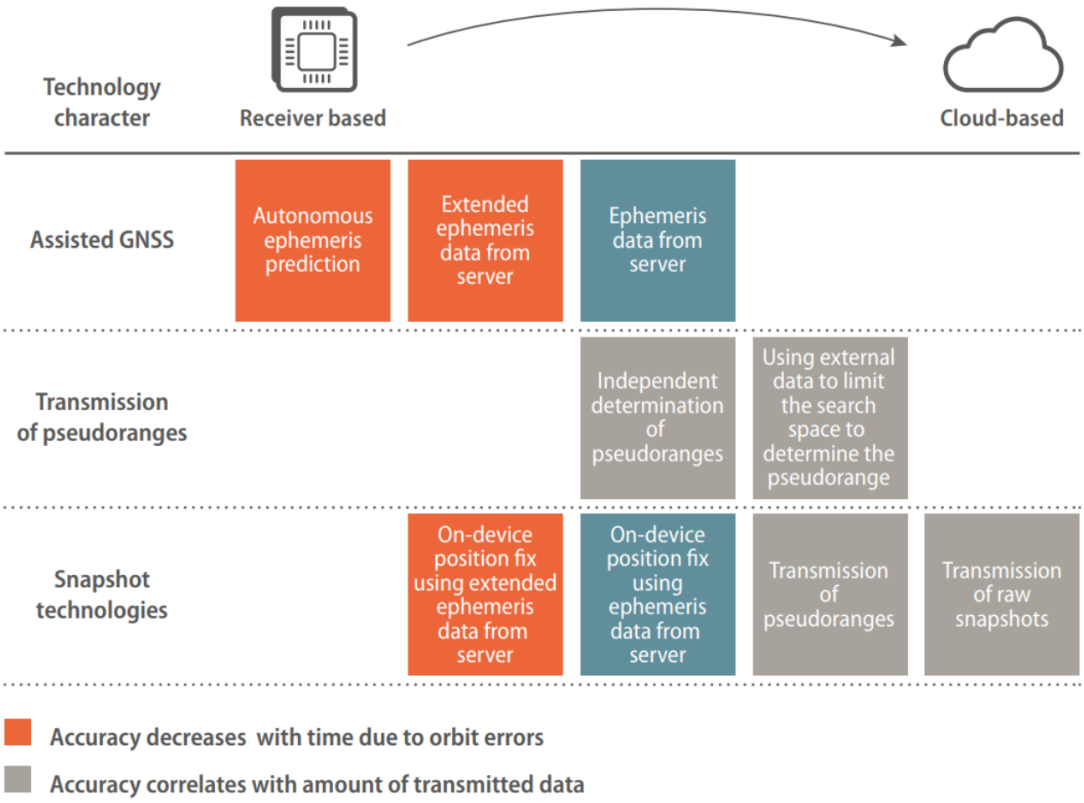

Figure 1.1: From receiver to cloud: GNSS optimisation techniques

perfectly for cloud based processing, which relies on the "outsourcing" of the majority of the computation workload to the cloud infrastructure. This change of strategy further reduces energy consumption in a significant way.

Figure 1.1 illustrates the most commonly used strategies aiming at reducing energy consumption using cloud-based technology. The blocks located on the left side of the figure are mainly receiver-based techniques, while those on the right side rely more on cloud-based techniques. As it can be seen in the last row, in order to take full advantages of cloud based strategy, the ultimate form of snapshot technology is based on the transmission of raw snapshots. This implies that the baseband signal processing and navigation filters are all implemented on the cloud. [9] confirmed that cloud based snapshot solution is up to more than an order of magnitude more efficient compared to conventional GNSS approaches in terms of energy consumption.

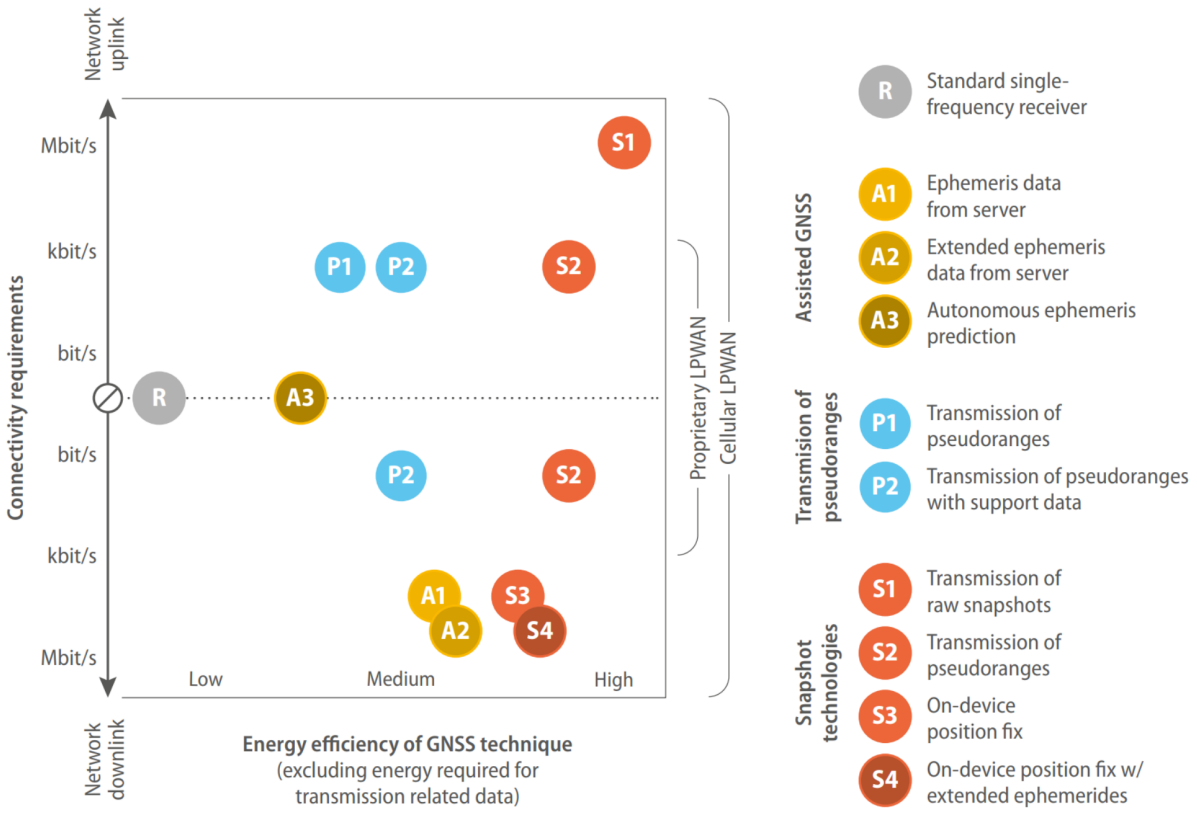

Figure 1.2: Connectivity requirements vs energy efficiency

The limitation in data connectivity is also a critical factor when shifting computation tasks from receiver to the cloud. An overview of connectivity and energy efficiency limitations for different positioning strategies is shown in Figure 1.2 [8]. As it can be seen, denoted by S1 circle, transmission of raw snapshots enjoys the highest energy efficiency in the positioning module. Although a relatively high up-link connectivity requirement needs to be fulfilled, no down-link connectivity is needed since the receiver only uploads the collected binary bits to the cloud and do not download any data. The present dissertation is focused on processing of the transmitted raw snapshots on the cloud.

1.3 Motivation

The increasing need of GNSS positioning modules for mass market IoT applications has been a key factor that drives the present research on snapshot positioning algorithms, especially in terms of:

1. **Energy efficiency**. As already mentioned in Section 1.2, the decrease in power consumption is the main advantage of snapshot positioning compared to conventional techniques. It motivates a systematic review of the currently most commonly used algorithms in snapshot technology.

2. **Positioning accuracy**. Accuracy measures were rarely addressed for snapshot positioning since currently only code delay based algorithms are developed [10]. It is thus encouraged to explore snapshot carrier phase measurements and develop new approaches for the navigation filter in order to obtain high accuracy PVT results.

3. **Economic cost**. In order to better serve the general public, the overall cost of deploying snapshot positioning service must be reduced to the minimum while satisfying the desired performance requirements. This motivates a detailed analyse of the performance and the cost. Not only the hardware cost of the snapshot receiver should be considered, but also the charges due to data transmission and other high precision data services.

Among these factors, advantages of snapshot positioning in terms of energy efficiency and economic cost have already been demonstrated in some current researches [8, 9]. It is the snapshot positioning accuracy that still requires more in-depth investigation as currently only meter-level accuracy can be achieved. The innovations and new algorithms presented in this contribution are motivated by the enormous potential applications of this technique, which have previously been obstructed by the fact that no high accuracy snapshot positioning algorithms are available so far.

1.4 Objectives

This thesis work is completed with the following objectives:

- Study and implement a functioning positioning engine incorporating state of the art GNSS algorithms.
- Consolidate the currently existed algorithms developed for snapshot GNSS positioning, including both signal processing and navigation filter data processing.
- Research and implement new methods for generating accurate snapshot GNSS observables, including time-tagged code delay, Doppler frequency offset and snapshot carrier phase, etc.
- Integrate snapshot observables to classic high accuracy navigation filters to achieve high accuracy snapshot positioning.
- Evaluate the performance of the state of the art snapshot algorithms together with the newly proposed methods.

1.5 Contributions

Along with the present dissertation, several other publications are made, including [10] and [11]. Besides, two related patent applications are filed and pending to be granted at the time of thesis writing. Furthermore, as part of an industrial PhD program, the algorithms involved in the present research were implemented and integrated to the code library of Albora Technologies. The author mainly contributed to the development of observable generation module and the specially designed high accuracy navigation filter.

1.6 Thesis Outline

This thesis contains six chapters and they are organised in this way: The present chapter has briefly introduced the background about GNSS and pointed out the motivations of developing snapshot positioning, as well as the objectives of this thesis.

Chapter 2 provides some details about the traditional GNSS techniques, including the baseband signal processing module and different types of data processing methods.

Chapter 3 provides a thorough walk-through of the conventional snapshot positioning algorithms. It focuses on the hypotheses construction in the acquisition module and the full period ambiguity resolution.

Chapter 4 explains the first main technical challenge faced by high accuracy snapshot positioning and provides a solution for it. The high accuracy snapshot positioning workflow is demonstrated and two sets of experiments are presented. Results from the first experiment confirm the feasibility of obtaining centimetre level positioning solutions under zero-baseline configuration and the results of the second experiment show more insights about the performance under different scenarios.

Chapter 5 tackles the second main challenge that addresses the potential errors in snapshot carrier phase measurements. Two different methods are proposed, respectively aiming at data signals and pilot signals. The improvements after applying these methods are demonstrated based on real world signals.

Chapter 6 concludes the performance of the above mentioned new methods and provides some suggestions and outlook for possible contributions on the snapshot positioning research in the future.

Chapter 2

Traditional GNSS Techniques

Since the start of the development of GNSS technologies in the 1960s, many researches have been done and contributed to the convenient use of these technologies [1, 12]. Thanks to the fast development of these technologies, they can be found in more and more modern applications such as drones, unmanned vehicles, etc. This chapter serves as a walk through of the GNSS techniques that are commonly used nowadays and provides some more detailed introduction to some important concepts that are used in the following chapters.

2.1 GNSS Signals

GNSS signals are the electromagnetic waves continuously transmitted by GNSS satellites in two or more frequencies in L band. These signals mainly consist of three components:

- Ranging code: also named as Pseudo-Random Noise (PRN) code, or primary code. It is used for the receiver to identify the satellite and determine the travel time from satellite to the receiver antenna.
- Navigation data: the binary bits of necessary information for each satellite, including data related to its position (ephemeris and almanac), time of transmission, satellite clock bias parameters, health status and other complementary contents.
- Carrier: Radio frequency sinusoidal signal at a given frequency in which the previous

two components are modulated.

For the primary codes, they can be characterized by the modulation scheme, code length (bits), code period (ms) and code bit rate (Hz), which is also known as the chip rate. For instance, the GPS L1 C/A signals employ the Gold codes of 1023 bits, the corresponding primary code period is 1 ms and the chip rate is 1.023 MHz. Such primary codes are modulated through the Binary Phase Shift Keying (BPSK) scheme.

Navigation messages are typically composed of frames that each carries a different set of information. Each frame may consists of several subframes. Some signals have fixed structures of such messages, while others employ more flexible formats. It is important to point out that many modern signals employ separate data and pilot components and the latter does not contain any navigation message data bits.

For the carrier signal, currently there are a few center frequencies that are used for different GNSS constellations. For example, the L1 frequency band, which is located at 1575.42 MHz, is used by GPS L1 C/A, Galileo E1, BDS B1C and SBAS signals; the L5 frequency band that is located at 1176.45 MHz holds GPS L5, Galileo E5a, BDS B2a and other signals from regional constellations such as the NAVIC; the L2 frequency band located at 1227.6 MHz hosts GPS L2 signals, etc.

Beside these parts, some other components such as overlay codes are also present in the composition of the so-called Signal In Space (SIS). These codes, also named as secondary codes, effectively extend the length of the short primary codes by modulating each duration of primary code period with bits that are well documented in the signal specification documents [13]. With the knowledge of the secondary code bit values, the receivers can exploit longer coherent integration times in their processing. A SIS can be characterized by the properties related to the above mentioned signal components together with some other parameters [14], such as the modulation type, transmitted power, etc.

Since the primary codes, navigation message data bits and secondary codes are of particular importance for the present PhD research, the characteristics of some of the most

commonly used signals are listed in Table 2.1.

Table 2.1: Basic information about the most commonly used GNSS signals. The rows colored in green represent the pilot signals while the others contain navigation message data bits.

	primary code		
	Chip rate [MHz]	Code length	Code period [ms]
GPS L1 C/A	1.023	1023	1
GPS L5Q	10.23	10230	1
Galileo E1B	1.023	4092	4
Galileo E1C	1.023	4092	4
Galileo E5a-I	10.23	10230	1
Galileo E5a-Q	10.23	10230	1
Beidou B1C (data)	1.023	10230	10
Beidou B1C (pilot)	1.023	10230	10
	Data bits		
	Bit rate [Hz]	Bit duration [ms]	
GPS L1 C/A	50	20	
GPS L5Q	N/A		
Galileo E1B	250	4	
Galileo E1C	N/A		
Galileo E5a-I	50	20	
Galileo E5a-Q	N/A		
Beidou B1C (data)	100	10	
Beidou B1C (pilot)	N/A		
	Secondary code		
	Code length	Code period [ms]	Notes
GPS L1 C/A	N/A		
GPS L5Q	20	20	20-bit Neuman-Hofman code
Galileo E1B	N/A		
Galileo E1C	25	100	$CS25_1$ code
Galileo E5a-I	20	20	$CS20_1$ code
Galileo E5a-Q	100	100	$CS100_{1-50}$ code
Beidou B1C (data)	N/A		
Beidou B1C (pilot)	1800	18000	Truncated Weil code

2.2 GNSS Receiver

The GNSS receiver, as a basic unit of the user segment of GNSS, is the device that acquires the SIS and computes the PVT solution based on the algorithms running inside. Traditional receivers have a similar architecture that can generally be illustrated in Figure 2.1 that is

extracted from [15].

First of all, the electromagnetic waves are captured by the antenna, which is usually accompanied by a preamplifier. The resulting Radio Frequency (RF) signal is then passed to the RF front-end to be down-converted and filtered in order to obtain the Intermediate Frequency (IF) signal that can be accepted by the Analogue to Digital Converter (ADC) module, as denoted by the third block. Note that both the RF front-end and the ADC require a local oscillator as the input signal. The output at this stage is a time series of samples that are typically represented in binary format. They are then fed to a baseband processing module that aims at generating the raw GNSS measurements for each satellite signal acquired. These raw measurement data consist of code pseudoranges, carrier phases and Doppler frequency offsets. They are finally processed in the PVT processing block to compute the navigation solutions that are finally provided to the GNSS users.

The most extensively discussed blocks for the GNSS community are the baseband processing and PVT computation blocks [16, 1], the other functional blocks shown in Figure 2.1 are common procedures for any wireless applications. In terms of the baseband signal processing block, the performance characteristics are mostly influenced by the receiver design choice between the open-loop batch processing architecture and the closed-loop sequential processing architecture [17].

The close-loop architecture is most commonly chosen for traditional GNSS receivers. It consists of multiple channels that each encapsulates the blocks for satellite acquisition, signal tracking, and demodulation of the navigation message for each signal and for each satellite being processed. While these steps are being processed sequentially for each channel, the desired measurements can be estimated continuously during the tracking step that is built on a close-loop architecture.

The open-loop approach, however, correlates the input signal with a batch of replicas in parallel. Each replica is constructed based on different code delay and Doppler parameter candidates, which form the so-called search space. Note that the open-loop work flow is sim-

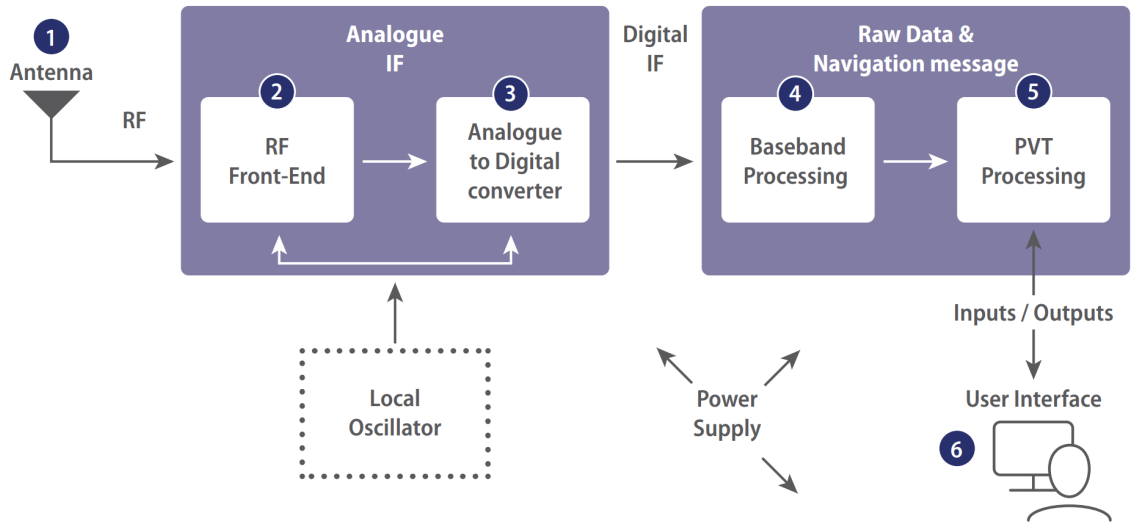

Figure 2.1: GNSS receiver functional block diagram [15]

ilar to the acquisition step of the close-loop approach but with more refined search space and different estimation techniques. More details are described in section 2.2.1. This approach has received much less attention from the GNSS community although it proves to be more advantageous in some applications [17].

As for the PVT processing module, also named as a navigation filter, it holds the main algorithms for the estimation of the navigation solution, including Standard Point Positioning (SPP), Real Time Kinematics (RTK), Precise Point Positioning (PPP), etc. All these procedures and implementation methods are introduced in the following subsections.

2.2.1 Acquisition

The first step of the baseband signal processing block is known as satellite acquisition. This step collects the IF digital bits that contain the information of the incoming signal and performs the following two tasks [18]:

- Identify the satellites that are mixed in the signal received at the antenna.
- Roughly determine the estimates of the code phase and the Doppler frequency offset for the observed satellites.

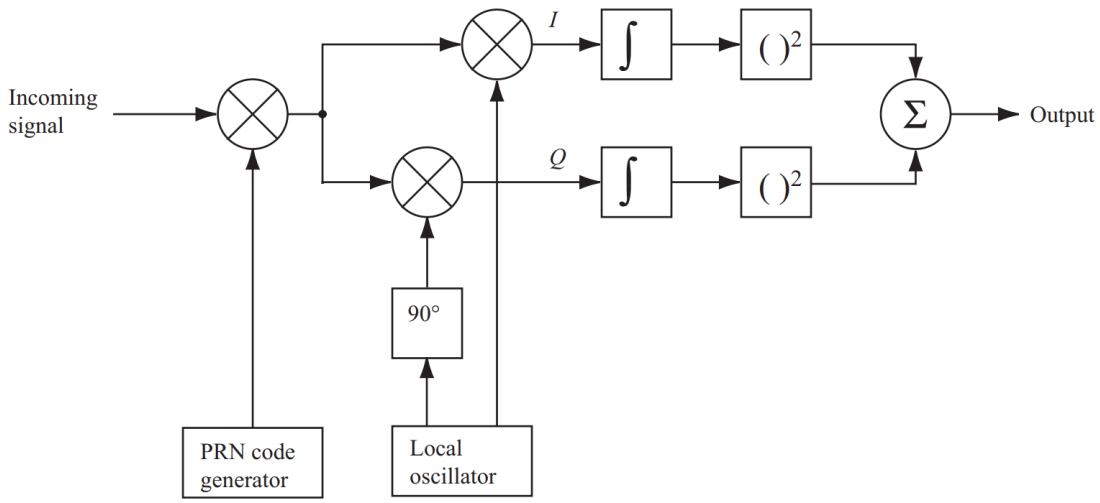

Figure 2.2: Block diagram of the serial search algorithm for acquisition

There are several different ways of implementing the acquisition step, but the most basic workflow is based on a serial search which is presented in Figure 2.2. As it can be seen, the incoming signal samples are correlated with a primary code replica generated locally. This output signal is then multiplied by a carrier signal generated by a local oscillator in two branches. One branch is the In-phase branch, noted by **I**, and the other is the Quadrature branch, noted by **Q**, which has applied a 90 degrees shift in the phase of the locally generated carrier signal. Both the I and Q branch results are integrated over a period defined by the coherent integration time, typically denoted by t_{coh}, and then squared and summed up to form the final correlation magnitude value.

To successfully perform the acquisition task, the receiver needs to rely on the following two correlation properties of the GNSS satellite primary code sequences [18]:

- The correlations between primary codes of two different satellites are close to zero. This is the root reason that allows identifying each satellite available in the obtained signal. In practice we generate the primary codes locally for all the satellites and only the channels corresponding to visible satellites present a correlation response at the end of the acquisition process.

- The auto-correlation output of the PRN sequence is nearly zero when the lag is over one chip. The pattern of the Auto Correlation Function (ACF) when the lag is below one chip depends on the signal modulation method, but the output magnitude is always at the maximum when the lag is zero. For example, for GPS L1 C/A signal, a triangular ACF profile can be observed between $-\frac{1}{1.023}\ \mu s$ and $\frac{1}{1.023}\ \mu s$ of offset. For BOC signals, their ACF have profiles with more local peaks, which complicates the acquisition process as the correct detection of the main peak remains challenging in the presence of noise.

As it can be expected, due to the second property of the PRN sequences, it is necessary to sweep through the whole PRN sequence and generate local replicas with all different code phases in order to get a high magnitude of acquisition output. This refers to the search in code phase space. Another similar search procedure is applied in order to find the proper Doppler frequency offset value. This is performed by adding different frequency steps to the IF in the local carrier generation process that is controlled by the Numerically Controlled Oscillator (NCO). These two search steps are independent from each other and result in a two dimensional search space that is extensively explained in various literature about GNSS baseband signal processing [18, 1]. Since the sequential search will result in a huge amount of combinations due to the large number of code phase and Doppler offset candidates, parallel search methods have been introduced in order to reduce the computational burden. These methods are usually based on the Fast Fourier Transform (FFT) technique, which is a practical fast implementation of the Digital Fourier Transform (DFT) for computers.

To better grasp the idea of the acquisition process, it is important to understand the mathematical model behind these correlations. Starting from the digital samples of the incoming Radio Frequency (RF) signals which were originally down-converted to IF, it can be mathematically represented as [19]:

$$r[n] = \sum_{i=1}^{L} A_i c_i \left[n - \tau_i\right] d_i \left[n - \tau_i\right] \cos\left[2\pi F_D^i n + \phi_i\right] + \eta[n] \tag{2.1}$$

where:

- L is the total number of satellites that broadcast the received signal
- A_i is the amplitude of the i-th useful signal
- c_i is the primary code (or PRN code) sequence, which is known a priori
- d_i represents the navigation message data bits, which are absent from the pilot component of some modern signals
- τ_i is the code phase at the satellite transmission time, in ms
- $F_D^i = f_{IF} + f_D^i$ is the sum of the intermediate frequency magnitude and the Doppler offset, in Hz
- ϕ_i is the carrier phase measurement at the satellite transmission time, in rad
- $\eta[n]$ is the signal noise after the front-end processing

These digital samples $r[n]$ are fed to the correlation process with local replicas of spreading codes and carrier signals, which generates for each signal a Cross Ambiguity Function (CAF) that can be defined by:

$$Y(\tau, F_D) = \frac{1}{N} \sum_{n=0}^{N-1} r[n] c[n-\tau] \exp\{-j2\pi F_D n\} \tag{2.2}$$

where:

- N is the total number of samples, which depends on the sampling frequency and the total length of the correlation time
- $r[n]$ is the received signal samples that is expressed in Equation (2.1)
- $c[n-\tau]$ is the local replicas of the primary code
- $\exp\{-j2\pi F_D n\}$ represents the local carrier signal with the frequency set to F_D

As it can be seen in Equation (2.2), τ and F_D are the two key variables that determine the magnitude of CAF output. The target of the acquisition process is to find a set of such parameters that lead to a distinctive peak of the CAF output. In practice, the complex correlation with the local carrier signal is implemented by the multiplication of the two orthogonal components of this carrier: one branch is multiplied by a sine wave and the other branch multiplied by a cosine wave. In general, the CAF results of these two branches of local replicas can be represented by:

$$\begin{aligned} Y_I(\tau, F_D) &= \frac{1}{N}\sum_{n=0}^{N-1} r[n]\cos(2\pi F_D n)\, c[n-\tau] \\ Y_Q(\tau, F_D) &= \frac{1}{N}\sum_{n=0}^{N-1} r[n]\sin(2\pi F_D n)\, c[n-\tau]. \end{aligned} \tag{2.3}$$

Y_I and Y_Q stand for the In-phase and Quadrature components of the correlation results. With these two components represented in this way, Equation (2.2) can be rewritten as:

$$Y(\tau, F_D) = Y_I(\tau, F_D) + jY_Q(\tau, F_D) \tag{2.4}$$

The magnitudes of the CAF outputs are computed with multiple tentative values in the code phase and Doppler offset dimensions. A detection procedure has to be applied in order to decide whether we have achieved a successful acquisition of the satellite. Once the acquisition of the satellite is confirmed, the final estimates of the two critical parameters are determined by identifying the peak magnitude, which can be represented by:

$$< \hat{\tau}, \hat{F_D} >= argmax\{Y_I^2(\tau, F_D) + Y_Q^2(\tau, F_D)\} \tag{2.5}$$

There are usually different methods that can be applied for this process, the simplest one is the serial search acquisition method that applies a brute force search on all the grid points of the two dimensional search space. However, as it can be expected, this method can

be quite computationally expensive when there is a large amount of grid points to search on. This search size depends on the search bin margins and the separations of two adjacent candidates, which is also usually named as the bin width. When a more refined result is desired, the bin width needs to be reduced and this results in a larger number of candidates to be searched on. Therefore, the trade-off between the acquisition output precision and the computational efficiency needs to be well considered. Fortunately, there are other different algorithms developed that largely accelerates this process. These algorithms rely on the implementation of the FFT that performs a transformation from time domain to frequency domain and an Inverse FFT (IFFT) process that transform the frequency domain operation results back to time domain. The use of such techniques is customary in modern GNSS receivers, more details can be found in various literature [18, 19, 20, 21].

Following the obtainment of the optimal estimates of the code phase and Doppler offset, the carrier phase of the signal at satellite transmission time can be represented using the formula of the complex angle:

$$\hat{\phi} = arctan(\frac{Y_Q\left(\hat{\tau}, \hat{F_D}\right)}{Y_I\left(\hat{\tau}, \hat{F_D}\right)}) \tag{2.6}$$

Typically the carrier phase measurements are not calculated in the acquisition stage. Instead, they are obtained in the tracking loops where finer Doppler frequency offset are estimated and thus result in more accurate and continuous carrier phase measurements. However, for special applications such as snapshot processing, they are computed in the acquisition step, more details are given in Chapter 3.

2.2.2 Tracking

The tracking step is a continuous estimation process following the acquisition step. It is recalled that the acquisition step requires the generation of a local replica signal on each channel in order to have a rough estimate of the code phase and Doppler frequency offset.

These replica signals are composed of local carrier signals at a tentative frequency and the primary codes and other encoded data bits. Due to the movement of GNSS satellites and possible user dynamics, the Doppler frequency offsets on each channel of the baseband processing module are continuously varying. Similarly, the code phase on each channel is also constantly changing since it depends on their propagation times and the signal reception time.

The main purpose of signal tracking is to match the local replicas with the changes of the Doppler frequencies and the code phases of each signal being processed. By continuously tracking the signal, we can avoid the computationally expensive search procedure as required in the acquisition step and at the same time obtain finer estimates of the main GNSS measurements: the code phase, Doppler offset and carrier phase. Additionally, after stable signal tracking is established, the data bits encoded in the received signal can be demodulated.

The tracking procedure can be divided into carrier tracking and code tracking. Carrier tracking is usually implemented in the form of a Phase Lock Loop (PLL) or a Frequency Lock Loop (FLL), while for code tracking a Delay Lock Loop (DLL) is typically applied [22]. The principles behind these tracking loops are similar: achieving the maximum consistency with the received signal by adjusting the code delay and Doppler offset parameters of locally generated replicas.

Figure 2.3 illustrates the basic architecture of GNSS signal tracking. There are mainly three components in the tracking process for each channel, namely, the correlator, the carrier tracking loop and the code tracking loop. Starting from the left side of the figure is the correlator part, which collects the digital samples of the received signals $r[n]$ and performs two stages of the so-called wipe off process. The first stage of wipe off is to eliminate the carrier signal by the multiplication with a locally generated sinusoidal signal at the frequency of $\hat{F_D}$, which was initially obtained in the acquisition process. The second stage is to wipe off the encoded primary codes, by the multiplication with a local replica of the PRN sequence with a code phase of $\hat{\tau}$.

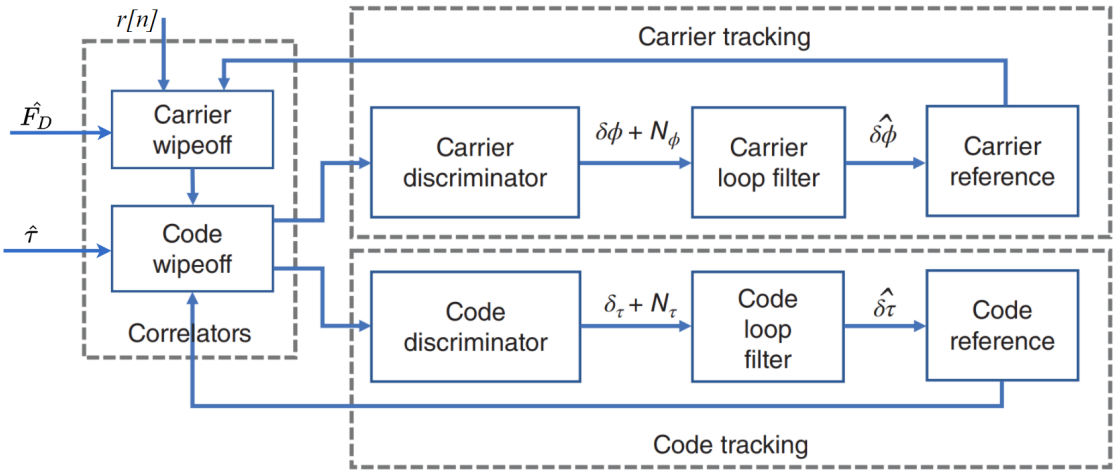

Figure 2.3: GNSS signal tracking architecture [4]

Ideally, the acquisition results are exactly equal to the actual carrier Doppler offset and code phase. Then, the two stages of correlation process should totally wipe off the carrier wave and the primary codes and result in only the accumulated signal amplitude over the correlation interval. The signs of these amplitude values are actually the remaining data bits in the received signal such as the navigation message bits or secondary codes. However, perfect correlations at this stage almost never happen since the acquisition results only provide coarse estimates, and that is the main reason why tracking loops need to be applied.

The upper part of the right side of Figure 2.3 depicts the carrier tracking loop. As it can be seen, the correlation results first pass through a carrier discriminator block. The discriminator is an estimator that provides a measure of the parameter discrepancy between the local signal and the received signal. In the case of carrier tracking, it estimates the phase error (for PLL) or the Doppler offset error (for FLL) of the locally generated carrier signal. The PLL and FLL estimates always contain noise terms, thus a carrier loop filter is used here in order to reduce the impact of such noise on the accuracy of the discriminator outputs. After that, these estimates are used to generate a new carrier reference that is then fed back to the correlator process for the subsequent carrier wipe off.

The code tracking loop is performed in parallel to the carrier tracking loop with a sim-

ilar procedure. The code discriminator estimates the code phase discrepancy between the currently used local code reference bits and the actual PRN sequence in the received signal. After being filtered by the code loop filter, a refined primary code reference sequence is generated based on the magnitude of this discrepancy and then fed back to the correlator for the following code wipe off process.

When these tracking loops reach a steady status, the discriminator outputs, i.e., the local reference carrier and code errors should be relatively low, although their magnitude still depend on the loop filter design choices. Meanwhile, the updated code phase and carrier phase estimates can be safely taken from the locally generated reference signals. More details about tracking loop implementations can be found in various literature [1, 22, 23]. Since the main focus of the present PhD is on snapshot positioning which adopts the open-loop signal processing architecture, the signal tracking is not further discussed.

2.2.3 Measurement Generation

Section 2.2.2 presented how continuous estimates of code phase and carrier phase are obtained through tracking loops. These measurements, however, can not be directly used for the positioning algorithms because they are generated under different time scales. The tracking loops for different satellites could be established at epochs that are very different from each other, even though the signal reception time is the same for all. In other words, the code phase value obtained at this stage for each satellite is only a fractional value that is between zero and one full code period (lasting, for instance, 1 ms in GPS L1 C/A signal).

In order to generate a set of full pseudoranges that are properly time tagged, some further steps must be taken. Only these full pseudoranges can correctly reveal the geometric distance relationships among all satellites. The computation of these pseudoranges is dependant on the determination of the misalignment of the time scales within which each code phase value is measured. This process is equivalent to the determination of the transmission times of different satellites. To better understand this procedure, the following paragraphs use the

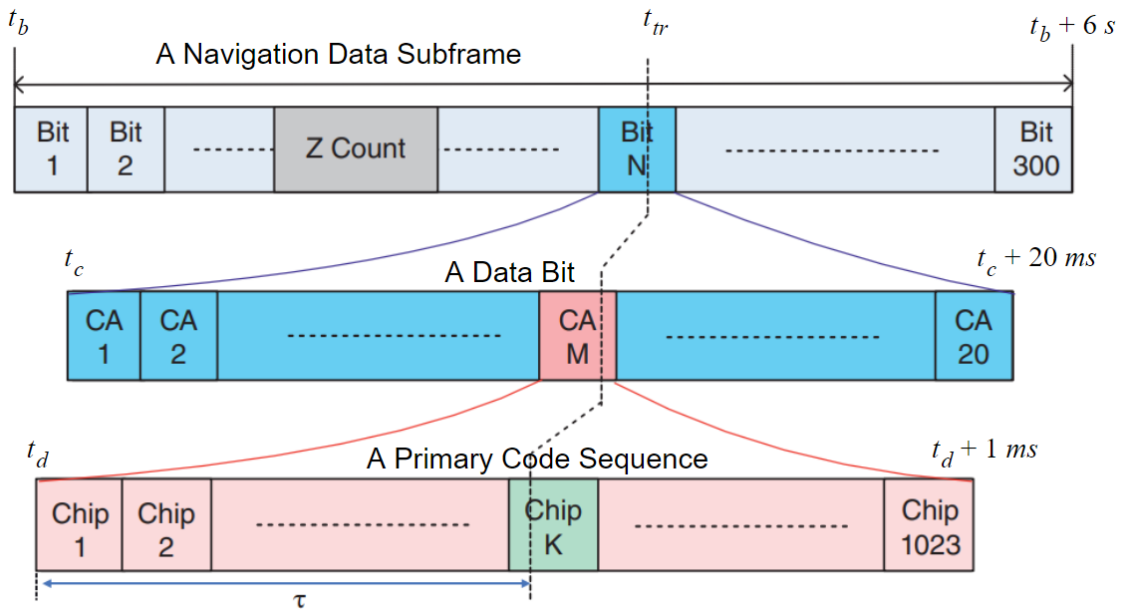

Figure 2.4: Illustration of computation of satellite transmission time of a GPS satellite [4]. The three rows respectively represent: a subframe containing 300 bits (gray, t_b to $t_b + 6\ s$), a bit lasting 20 primary code periods (blue, t_c to $t_c + 20\ ms$) and a primary code sequence with 1023 chips (pink, t_d to $t_d + 1\ ms$).

GPS L1 C/A signal as an explanatory example.

Figure 2.4 depicts the procedure to compute a GPS satellite transmission time by exploring different levels of codes modulated in this signal. As it can be seen, in order to find the transmission time t_{tr} of the currently received signal sample, it is necessary to first have the knowledge of a highly accurate absolute basis time, denoted as t_b. Then, the current bit index inside a full subframe is also required. In the example shown in Figure 2.4 this index is denoted as N. Note that this example shows only GPS L1 C/A signal, for which a full subframe contains totally 300 bits lasting 6 s, thus N is a value between 1 and 300.

Besides, to further increase the precision of this transmission time, the index of the current primary code period should also be computed, as denoted by M in this example. Note that M is a value between 1 and 20 since there exists 20 primary code periods in a GPS L1 C/A signal navigation data bit that lasts 20 ms. Finally, by the addition of the code phase τ that was previously obtained with the DLL, the satellite transmission time can

be obtained. This computation can be explained by:

$$t_{tr} = t_b + (N-1) \times 20 \times 10^{-3} + (M-1) \times 10^{-3} + \tau \tag{2.7}$$

where the necessary parameters M, N and t_b are respectively found in the processes of **bit synchronisation**, **frame synchronisation** and **navigation message data decoding**. These processes are further explained in the following paragraphs. The constants in Equation (2.7), 20×10^{-3} and 10^{-3}, respectively represent the 20 ms duration of each data bit and the 1 ms primary code period.

The first step towards the full pseudorange computation is to achieve bit synchronisation, which aims at finding the edges of the primary code period during which the received signal was transmitted. To represent it in the second row of Figure 2.4, this step is equivalent to find the nearest vertical line on the left side of t_{tr}, i.e., the starting edge of CA M that is shaded in pink. Then, the information about the received signal sample location in the current bit can be obtained by referring to these detected edges.

The bit synchronisation process is usually achieved by detecting the positions of bit transitions. It is recalled that when the DLL is successfully established, only the navigation data bits are left in the correlator output after the two stages of the aforementioned wipe off process. Figure 2.5 depicts such explanatory example of the resulting values of data bits sampled every 1 ms with indices that repeat from 1 to 20. Note that initially the starting index position (red 1) has been randomly chosen and the correction to it needs to be determined in this step. The blank boxes represent a bit value of +1 and the boxes shaded in gray represents data bits of -1.

In this example, we can see that the transition edge from blank boxes to shaded boxes is between index 3 and 4. It indicates that the values of the first 3 ms belong to one navigation data bit and the following 20 values belong to the next data bit. To avoid the false detection of bit transitions caused by the correlator output noise, an algorithm based on counting the

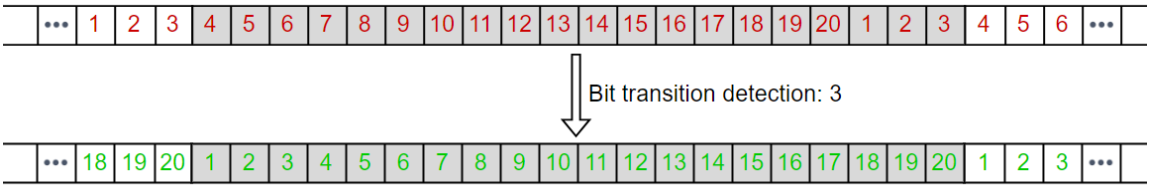

Figure 2.5: Bit synchronisation based on detecting sign transitions in the values of the correlator output after signal tracking. The blank boxes represent a bit value of +1 and the boxes shaded in gray represents data bits of -1. Red and green numbers are the indices before and after correction based on detected bit edge.

number of changes of adjacent values was proposed in [23] for more robust edge detection. Once these data bit edges are found, the starting index position is corrected (now green 1) to these edges. Then, the M values of Equation (2.7) for any further epochs are now equivalent to these corrected index values (green indices).

The next step is the frame synchronisation. The purpose of this step is to determine the location of the current navigation data bit in the scale of a full subframe of the navigation message. For the example shown in Figure 2.4, this refers to the determination of the N value of Equation (2.7). Since the bit synchronisation process has determined the edges of each data bit, we now can obtain a continuous stream of these bits. In order to find the start of a subframe, we need to explore the content of the navigation messages. Again following with the GPS L1 C/A example, the first word of each subframe is the telemetry word, also known as the TLM word. The first 8 bits inside this word are the synchronisation bits, which are always 10001011. The starting edge of a subframe is thus detected by searching this fixed bit sequence in the continuous stream that was obtained following the bit synchronisation process.

Besides, since the carrier phases are unknown at the initial stage of data demodulation, the first decoded bit is assigned with a value randomly, either 1 or 0, and then the following bits are decoded in relation to the initial bit. Thus, the decoded bits can also be totally opposite to the actual bits, as a results, a search of the reversed synchronisation bits (i.e.,

01110100) should also be performed. If such reversed synchronisation bits are detected in the obtained data stream, then all the values in this stream should be reversed to ensure that they correspond to the actual transmitted bits. In addition, it must be noted that there is still a low possibility that the same bits could appear in other locations of the subframe. If we assume equal probability of 0 and 1 to be encoded (thus both 50%), then the probability of detecting the two possible 8-digit synchronisation sequences (actual or its reverse) is $2 \cdot (0.5)^8 = 0.78\%$. This is a probability that should not be ignored since subframe start can be wrongly detected from time to time. In order to improve the reliability of this detection, other data bits in the navigation message such as the parity check bits should also be explored [23].

Finally, the data bit stream should also be decoded to obtain the necessary information needed for positioning, such as the satellite ephemeris or clock correction parameters. The primary task of this decoding step is to obtain the transmission times of these subframes, which correspond to the timing basis of t_b in Equation (2.7) as shown in Figure 2.4. More specifically, this time is extracted from the Time Of Week (TOW) information data bits that are encoded in the Hand Over Word (HOW) that appears as the second word of each subframe. The TOW marks the number of seconds from the start of the current GPS week.

At the present moment in the processing, the signal transmission times t_{tr} for each satellite being tracked can already be computed using Equation (2.7). The pseudoranges are then computed based on these times, together with a receiver clock t_{rx} that is common for all satellites:

$$P = c \cdot (t_{rx} - t_{tr}) \tag{2.8}$$

where $c = 299792458\ m/s$ is the speed of light.

For carrier phase measurements, the procedures mentioned above can not be applied since there is no such structure for the carrier wavelength that is similar to the different levels of codes. Thus there is no means to identify how many carrier cycles passed by since the start of the subframe. Thus, each measurement is biased by a different integer number

of cycles, which are known as the carrier phase integer ambiguities. Fortunately, once the carrier tracking is successfully established, a counter can be initiated to record the number of cycles passed by at the NCO for each channel. Note that since the Doppler frequency offsets are different for different satellites, such numbers of elapsed cycles are also different even though the time gap between two epochs is identical for all satellites. By accumulating these elapsed cycles to the previous epoch, the resulting carrier phase measurements can reflect the actual geometric distance differences while maintaining the relationship between the initial carrier phase integer ambiguities.

As can be expected, these measurements are ultra precise, typically millimetre-level, but are impacted by the existence of the unknown integer ambiguities. The high precision feature of carrier phase measurements can be utilised for high accuracy positioning only when these ambiguities are correctly resolved.

2.2.4 SPP Navigation Filter

The above mentioned baseband signal processing steps have generated the three basic measurements of GNSS receivers: Doppler frequency offset, code pseudorange and carrier phase. For SPP applications, carrier phase measurements are usually omitted due to the existence of the so-called integer ambiguities and only pseudorange measurements are used in the navigation filter. The mathematical model of pseudoranges can be written as [2]:

$$P_{f_i} = \rho + c(dt_{rec} - dt^{sat}) + T + \alpha_i I + D_{rec,i} + D_i^{sat} + MP_i + \varepsilon_i \tag{2.9}$$

where:

- ρ represents the geometric range between the satellite and the receiver antenna phase centers, in m
- dt_{rec} and dt^{sat} are the receiver and satellite clock bias, in s

- c is the speed of light in vacuum, in m/s
- T and I are the slant delays occurred at the Troposphere and Ionosphere, in m
- α_i is a constant that depends on the frequency f_i
- $D_{rec,i} + D_i^{sat}$ is the Differential Code Bias (DCB) caused by the instrumental delays from the receiver and satellite. It is frequency-dependent as well, in m
- MP_i is the multipath effect of code pseudorange at frequency f_i, in m
- ε_i is the receiver noise on code measurements, in m

It is important to point out that the receiver position unknowns $\mathbf{r}_{rec} = [x_{rec}, y_{rec}, z_{rec}]^T$ are inside the geometric ranges ρ, which is not a linear function of these unknowns as it is computed by Equation (2.10):

$$\rho = \left\| \mathbf{r}^{sat} - \mathbf{r}_{rec} \right\| = \sqrt{(x^{sat} - x_{rec})^2 + (y^{sat} - y_{rec})^2 + (z^{sat} - z_{rec})^2} \tag{2.10}$$

Where $\mathbf{r}^{sat} = [x^{sat}, y^{sat}, z^{sat}]^T$ is the vector of satellite position coordinates computed at the signal transmission time. As it can be seen, it contains square and square root operations. In order to simplify the numerical operations, a Taylor expansion is applied to the original function and after omitting the residual terms with orders higher than one. Then the geometric range can be decomposed into a sum of a constant computed near the position of an initial guess and a linear multiplication of the position increment and the vector of derivative coefficients. This is represented by Equation (2.11).

$$\rho^j = \rho_0^j + \frac{x_0 - x^j}{\rho_0^j} dx + \frac{y_0 - y^j}{\rho_0^j} dy + \frac{z_0 - z^j}{\rho_0^j} dz \tag{2.11}$$

Where ρ_0^j is the geometric range computed at an approximate position $\mathbf{r}_{rec,0}$ with coordinates of (x_0, y_0, z_0) and (dx, dy, dz) represent the position solution increment that is actually the unknowns to be solved in the navigation equation. The superscript j stands for the satellite

index and (x^j, y^j, z^j) are the coordinates of satellite j. When the unknown dx, dy, dz values are solved, they can be summed to the a priori approximate position in order to obtain a refined receiver position solution. The computation of the position increment has to be iterated several times in order to have an enough accurate solution, especially when the initial guess is far away from the actual receiver position. It is common to define the Line Of Sight (LOS) unit vector for satellite j as:

$$\mathbf{1}^j = \left[\frac{x^j - x_0}{\rho_0^j}, \frac{y^j - y_0}{\rho_0^j}, \frac{z^j - z_0}{\rho_0^j} \right] \tag{2.12}$$

Inserting Equation (2.12) into Equation (2.11) leads to the simplified linear equation:

$$\rho^j = \rho_0^j - \mathbf{1}^j \cdot d\mathbf{r}_{rec} \tag{2.13}$$

where $d\mathbf{r}_{rec} = [dx, dy, dz]^T$ represents the vector of increments of receiver position unknowns.

As for other pseudorange error terms expressed in Equation (2.9), they are carefully modeled in order to reduce their effects on the navigation filter to the minimum. The satellite clock offset dt^{sat} can be computed using the clock parameters that are transmitted together with the ephemeris data. For single-frequency users, the satellite DCB terms D_i^{sat} are computed using the equation:

$$D_i^{sat} = \mathrm{TGD}_i^{sat} \tag{2.14}$$

Where TGD_i is the Total Group Delay (TGD) data that is also transmitted in the navigation message. The Ionosphere delay I is usually computed based on prediction models, the most commonly used models are the Klobuchar model [24] and the NeQuick model [25]. Note that satellite DCB and Ionosphere delays are inversely proportional to the square of carrier frequency, thus for dual frequency receivers, these delay terms can be totally removed by forming the so-called Ionosphere-Free combination based on this property [26]. The frequency

dependent coefficient α_i is computed by the following equation:

$$\alpha_i = \frac{40.3}{f_i^2} 10^{16} \tag{2.15}$$

The troposphere delay T can be compensated also by models such as the one used in SBAS [27].

By correcting the error terms mentioned above as well as replacing ρ by its Taylor expansion as expressed in Equation (2.11), Equation (2.9) can be rewritten as:

$$P_{f_i}^j - \rho_0^j + c \cdot dt^{sat} - \alpha_i I - D_i^{sat} - T = \frac{x_0 - x^j}{\rho_0^j} dx + \frac{y_0 - y^j}{\rho_0^j} dy + \frac{z_0 - z^j}{\rho_0^j} dz + c \cdot \delta t_{rec} + MP_i + \varepsilon_i \tag{2.16}$$

where δt_{rec} represents the sum of the unknown receiver clock error dt_{rec} and the receiver side instrumental delay $D_{rec,i}$, while ε_i is now redefined as the lump sum of the pseudorange noise and all the residual error terms after applying their correction models.

The left side of the Equation (2.16) is usually named as the prefit residuals, or the "Observed - Computed" ("O-C") measurement. For satellite indexed by j, this value is denoted by the parameter b^j, thus forming the following equation:

$$b^j = P_{f_i}^j - \rho_0^j + c \cdot dt^j - \alpha_i I - D_i^j - T \tag{2.17}$$

The right side of Equation (2.16) contains the four unknowns that need to be estimated in the navigation filter together with their corresponding coefficients. With each satellite having one pseudorange measurement, and hence one prefit residual, with more than four satellites the linear system represented by Equations (2.17) is over-determined. In order to obtain a positioning solution with minimum squared error, the Least Square Estimation (LSE) method is usually used to estimate the position and clock offset. By applying Equation (2.16) to all the satellites available in a particular epoch, the following matrix form of the

navigation filter can be obtained:

$$\begin{bmatrix} b^1 \\ \vdots \\ b^n \end{bmatrix} = \begin{pmatrix} \frac{x_0-x^1}{\rho_0^1} & \frac{y_0-y^1}{\rho_0^1} & \frac{z_0-z^1}{\rho_0^1} & c \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_0-x^n}{\rho_0^n} & \frac{y_0-y^n}{\rho_0^n} & \frac{z_0-z^n}{\rho_0^n} & c \end{pmatrix} \begin{bmatrix} d\mathbf{r}_{rec} \\ \delta t_{rec} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{2.18}$$

the following matrix form notations are usually used in order to simplify the navigation equations:

$$\mathbf{b} = \mathbf{G} \begin{bmatrix} d\mathbf{r}_{rec} \\ \delta t_{rec} \end{bmatrix} + \varepsilon \tag{2.19}$$

where:

- $\mathbf{b}$ is the vector of computed prefit residuals from all available satellites
- $\mathbf{G}$ is the (n $\times$ 4) geometry matrix that contains the satellite to receiver directional unit vectors
- ε is the vector of error terms that include the observable noises, residual errors due to imperfect modelling and multipath effects.

by solving this navigation filter using the LSE method, the resulting navigation solution can be expressed as:

$$\begin{bmatrix} \hat{d}\mathbf{r}_{rec} \\ \hat{\delta} t_{rec} \end{bmatrix} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{b} \tag{2.20}$$

with $\hat{d}\mathbf{r}_{rec}$ obtained, it is used to update the initial solution $\mathbf{r}_{rec,0}$ based on the equation:

$$\mathbf{r}_{rec} = \mathbf{r}_{rec,0} + \hat{d}\mathbf{r}_{rec} \tag{2.21}$$

The covariance matrix corresponding to the LSE estimation is computed by:

$$\mathbf{P} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1} \tag{2.22}$$

Equation (2.21) has completed an update from the initial guess solutions $\mathbf{r}_{rec,0}$ to the new solutions of $\mathbf{r}_{rec}$. However, as mentioned when linearising the geometric distance ρ, if $\mathbf{r}_{rec,0}$ is too far away from the actual receiver position, the Taylor expansion terms used in Equation (2.11) may not be enough accurate. Thus, in these scenarios, in order to ensure that the navigation Equation (2.16) properly projects the position solution increments to the prefit residuals, few iterations of such updates using Equation (2.21) need to be performed with the updated position solution of one iteration being used as the initial guess of the next iteration.

Equation (2.20) is developed under the assumption that all the elements in vector $\mathbf{b}$ should be trusted to the same extent, which requires that the pseudorange measurements from different satellites to have the same error variance. This assumption, however, can not be satisfied most of the time. Satellites from different positions may experience different atmospheric delays and the geometric distance difference can also introduce different impacts on the received signal strength for these satellites. Besides, the multipath effect tends to be greater for low elevation satellites. Thus, different weights are usually assigned to different satellites in the LSE in order to obtain a more reliable final solution, this refers to the Weighted Least Squares (WLS) solution. With this in mind, the SPP navigation solution can be modified as:

$$\begin{bmatrix} \hat{d\mathbf{r}}_{rec} \\ \hat{\delta t}_{rec} \end{bmatrix} = \left(\mathbf{G}^T\mathbf{W}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{W}\mathbf{b} \tag{2.23}$$

where $\mathbf{W}$ stands for the weighting matrix for all the measurements used in the navigation filter. The corresponding covariance matrix can be expressed as:

$$\mathbf{P} = \left(\mathbf{G}^T\mathbf{W}\mathbf{G}\right)^{-1} \tag{2.24}$$

Usually, the weighting matrix is taken as the inverse of the measurement covariance matrix $\mathbf{R}$, that is $\mathbf{W} = \mathbf{R}^{-1}$. As it can be seen, in order to obtain accurate WLS navigation

solutions, it is critical to have proper estimates of the measurement covariance matrix $\mathbf{R}$. To achieve this, normally we need to consider all the error sources, but there are some simplified stochastic models [28] that computes the variances values for different satellites based on their elevation angles only.

For traditional GNSS applications, the receiver device typically works in a continuous mode, which means the navigation solution needs to be computed epoch by epoch. Thus, the positioning solutions from the previous epoch may be used as an initial guess of the current navigation filter, since the position change between two consecutive epochs is usually not large. In this case, the navigation solutions usually converge after one single iteration. This method refers to the Sequential Least Square Adjustment [29]. In practice, a Kalman Filter [30] is also frequently applied for continuous measurement inputs.

The key difference of a Kalman filter compared to the LSE method is the inclusion of the state transition in the estimation process. This inclusion takes advantages of the information obtained from previous epochs and thus results in a more reliable output solution. Since it makes use of the previous status together with the current measurements, the resulting solutions are typically more smooth compared to the LSE-based methods that only take into consideration the latter part. The Kalman filter has two basic steps: Prediction and Update. These two steps lead to each other in an iterative way as the output of one step is the input of the other. The Kalman filter starts with an initial input of a rough estimate of the unknowns and its variance covariance matrix, it is firstly used in the prediction step, and the output estimates are computed every time the update step is performed. These two steps are described in more details below:

1. The **prediction** step is based on the state transition equation that can be expressed as:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^{-} &= \Phi_k \hat{\mathbf{x}}_k \\ \mathbf{P}_{k+1}^{-} &= \Phi_k \mathbf{P}_k \Phi_k^{\mathrm{T}} + \mathbf{Q}_k \end{aligned} \tag{2.25}$$

   where:

- $\hat{\mathbf{x}}_{k+1}^{-}$ represents the predicted states, which includes receiver position $d\mathbf{r}_{rec}^{-}$ and clock bias δt_{rec}^{-}
- $\mathbf{P}_{k+1}^{-}$ represents the covariance matrix of the predicted states at epoch $k+1$
- Φ_k is the state transition matrix at the epoch k and is defined by the internal links between the unknowns of the state vector
- $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ are the state vector and its covariance matrix obtained at epoch k
- $\mathbf{Q}_k$ is the process noise matrix, which is determined by the amount of trust we have on the transition model applied to the states

2. The **update** step is more similar to the navigation equations used before in the LSE method. It merges the information in the measurements collected at the current epoch into the navigation system. The update step can be formulated as:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^{-}\mathbf{G}_k^{\mathbf{T}}\left[\mathbf{G}_k\mathbf{P}_k^{-}\mathbf{G}_k^{\mathbf{T}} + \mathbf{R}_k\right]^{-\mathbf{1}} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^{-} + \mathbf{K}_k\left[\mathbf{y}_k - \mathbf{G}_k\hat{\mathbf{x}}_k\right] \\ \mathbf{P}_k &= \left[\mathbf{I} - \mathbf{K}_k\mathbf{G}_k\right]\mathbf{P}_k^{-} \end{aligned} \tag{2.26}$$

where:

- $\mathbf{R}_k$ is the covariance matrix of the current measurements
- $\mathbf{K}_k$ is the Kalman gain matrix computed based on weighing the covariance matrices of the predicted states and the confidence on the current measurements
- $\mathbf{y}_k - \mathbf{G}_k\hat{\mathbf{x}}_k$ is called innovation and represents the amount of new information coming into the system from external measurements at the epoch k
- $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ are the state vector and its covariance matrix obtained after the update step and are fed to the prediction step of the next epoch $(k+1)$ of state estimate.

For positioning engines based on Kalman filter, it is important to assign proper values to the $\mathbf{Q}_k$ and $\mathbf{R}_k$ matrices, the optimal estimates can be obtained only when they are

well configured. By adjusting these matrices we are essentially modifying the trust on the prediction and the update steps. This can be seen in the Kalman gain term $\mathbf{K}_k$. When the currently obtained measurements are considered to be less trustful, $\mathbf{R}_k$ matrix should be configured with large values and result in a small Kalman gain value, which lead to the updated states $\hat{\mathbf{x}}_k$ almost unchanged from the predicted states $\hat{\mathbf{x}}_k^-$, which indicates that the prediction step has taken the main role in deciding the final outputs. Contrariwise, if the measurements are known to be very precise, the $\mathbf{R}_k$ matrix should be filled with entries of smaller values. Then, with the relatively large values in the $\mathbf{Q}_k$ matrix, the Kalman gain can be expected to be almost equal to $\mathbf{G}_k^-$. This results in the fact that the $\hat{\mathbf{x}}_k$ values are almost only controlled by the measurements $\mathbf{y}_k$, that is, rarely being influenced by the predicted states $\hat{\mathbf{x}}_k^-$.

Since Kalman filter considers information coming from previous states, it is not an option for snapshot positioning since such information is not available due to the nature of extremely short recordings. Thus, Kalman filter is not further discussed in this thesis.

2.2.5 Real Time Kinematics

RTK is the positioning technique that makes use of carrier phase measurements and aims at achieving centimeter level accuracy. RTK is developed on the basis of the concept of Differential GNSS (DGNSS), which takes advantages of the fact that the measurement errors of GNSS receivers are usually correlated temporally and spatially. That is, the range errors originated from the inaccuracies of satellite clocks, ephemeris, Ionosphere and Troposphere are very similar for two nearby receivers. Thus, they can be largely cancelled out when computing the differences of GNSS measurements collected simultaneously by such two receivers. The differencing procedure between two receivers and two satellites forms the so-called Double-Differenced (DD) measurements, as illustrated in Figure 2.6. It can be seen that continuous GNSS measurements collected from a base station are required in order to implement RTK in the moving rover receiver.

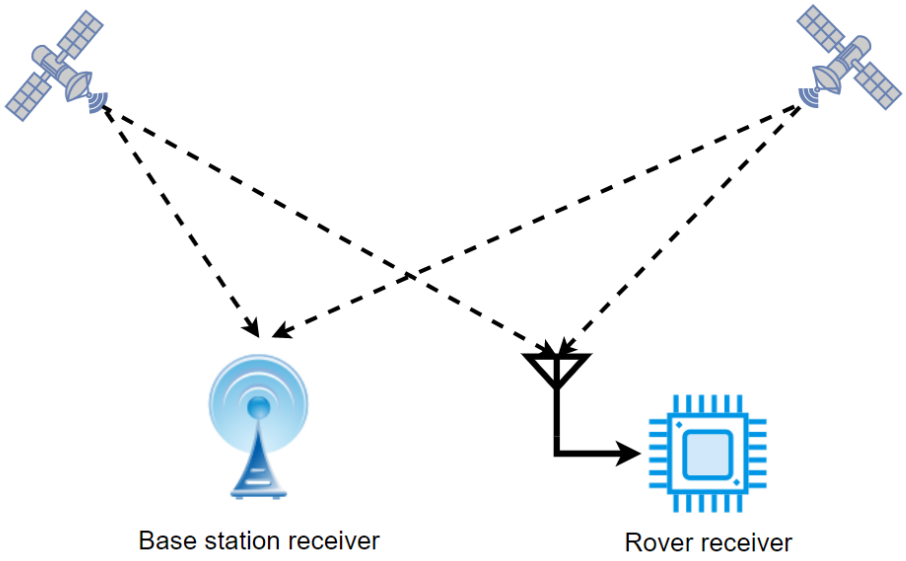

Figure 2.6: Double Differences built up between a moving (rover) receiver, a permanent base station and two satellites

The main reason that RTK shows a great advantage over SPP in terms of positioning accuracy is the use of carrier phase measurements, which are highly precise. These carrier phase measurements can be modelled as:

$$\Phi_{f_i} = \rho + c(dt_{rec} - dt^{sat}) + T - \alpha_i I + d_{rec,i} + d_i^{sat} + B_i + \lambda_i w + mp_i + \epsilon_i \tag{2.27}$$

where:

- $B_i = \lambda_i N_i$ is the ambiguity term, with λ_i and N_i respectively representing the wavelength and the integer ambiguity cycles for frequency i.
- $d_{rec,i}$ and d_i^{sat} are the Uncalibrated Phase Delay (UPD) of from the satellite and the receiver, respectively.
- $\lambda_i w$ represents the carrier phase offset introduced by the satellite to receiver rotation and the circular polarisation of the GNSS signal, i.e., wind-up.
- ϵ_i is the noise of carrier phase measurements.

The main difference of these phase measurements from pseudoranges is that the noise term ϵ_i is about two orders of magnitude lower than the code noise ε_i. Besides, the existence of carrier phase ambiguity terms B_i and the opposite sign for the Ionosphere are also the key differences.

After performing the DD procedure as shown in Figure 2.6, the resulting measurements are free from most of the error terms of Equation (2.27) since they are cancelled out. This can be seen from the mathematical model of the DD pseudoranges and DD carrier phases:

$$P_{rb}^{(kl)} = \rho_{rb}^{(kl)} + \alpha_i I_{rb}^{(kl)} + T_{rb}^{(kl)} + \varepsilon_{rb}^{(kl)} \tag{2.28}$$

$$\Phi_{rb}^{(kl)} = \rho_{rb}^{(kl)} + B_{rb}^{(kl)} - \alpha_i I_{rb}^{(kl)} + T_{rb}^{(kl)} + \epsilon_{rb}^{(kl)}, \tag{2.29}$$

Where:

- The subscripts r and b represent the rover and base station receiver respectively.
- The notation $(\cdot)_{rb}^{(kl)}$ represents the quantity of the parameter in the bracelet after performing the DD between the rover and base receivers and between satellites with indexes of k and l.
- $B_{rb}^{(kl)} = \lambda \cdot N_{rb}^{(kl)}$ represents the DD carrier phase integer ambiguity term.

Besides the carrier phase ambiguities, the remaining error terms in the DD measurements are the residual DD Ionosphere and Troposphere slant delays. Atmosphere delays are highly correlated when the rover and base receivers are located within a very short range (also known as the RTK baseline). Thus, for short baseline scenarios, these two residual DD delays are typically very small and are sometimes omitted for simplicity, which results in:

$$P_{rb}^{(kl)} = \rho_{rb}^{(kl)} + \varepsilon_{rb}^{(kl)} \tag{2.30}$$

$$\Phi_{rb}^{(kl)} = \rho_{rb}^{(kl)} + B_{rb}^{(kl)} + \epsilon_{rb}^{(kl)}, \tag{2.31}$$

It is noticed that this simplified DD model is valid for baseline distances up to few tens of kilometers.

In fact, the purpose of applying the DD process is to remove as much as possible the error terms in carrier phase measurements, so that the ambiguity values can be estimated and fixed to a set of integer values. High accuracy positioning estimates are available only when these integers are correctly determined.

In order to develop a navigation equation that can be solved by methods similar to those mentioned in Section 2.2.4, the linearisation step must be done on the DD geometric range also. After the first differencing step (between rover and base receivers), the SD geometric range for satellite k can be expressed as:

$$\rho_{rb}^{(k)} = \rho_{r}^{(k)} - \rho_{b}^{(k)} = \rho_{r,0}^{(k)} - \rho_{b,0}^{(k)} - (\mathbf{1}_{r}^{(k)} \cdot d\mathbf{r}_r - \mathbf{1}_{b}^{(k)} \cdot d\mathbf{r}_b) \tag{2.32}$$

The term $\mathbf{1}_{r}^{(k)}$ and $\mathbf{1}_{b}^{(k)}$ can be assumed as equal considering that the geometric range between the GNSS satellites and the receiver is significantly larger than the baseline distance. This component can then be simplified as:

$$\rho_{rb}^{(k)} = \rho_{rb,0}^{(k)} - \mathbf{1}_{r}^{(k)} \cdot (d\mathbf{r}_r - d\mathbf{r}_b) = \rho_{rb,0}^{(k)} - \mathbf{1}_{r}^{(k)} \cdot d\mathbf{r}_{rb} \tag{2.33}$$

With $d\mathbf{r}_{rb} = d\mathbf{r}_r - d\mathbf{r}_b$ representing the increment of baseline vector. Similarly, the component for satellite l can be expressed as:

$$\rho_{rb}^{(l)} = \rho_{rb,0}^{(l)} - \mathbf{1}_{r}^{(l)} \cdot d\mathbf{r}_{rb} \tag{2.34}$$

Then, DD measurements can be formed by the subtraction of SD measurements of two different satellites, which can be expressed as:

$$\rho_{rb}^{(kl)} = \rho_{rb}^{(k)} - \rho_{rb}^{(l)} \tag{2.35}$$

By substituting Equation (2.33) and (2.34) to Equation (2.35), the following final form of the linearised DD geometric range can be obtained:

$$\rho_{rb}^{(kl)} = \rho_{rb,0}^{(kl)} - (\mathbf{1}_r^{(k)} - \mathbf{1}_r^{(l)}) \cdot d\mathbf{r}_{rb} \tag{2.36}$$

Then, by inserting Equation (2.36) to Equation (2.30) and (2.31) the linear DD measurement equations can be represented as:

$$y_{P,rb}^{(kl)} = -(\mathbf{1}_r^{(k)} - \mathbf{1}_r^{(l)}) \cdot d\mathbf{r}_{rb} + \varepsilon_{rb}^{(kl)} \tag{2.37}$$

$$y_{\Phi,rb}^{(kl)} = -(\mathbf{1}_r^{(k)} - \mathbf{1}_r^{(l)}) \cdot d\mathbf{r}_{rb} + B_{rb}^{(kl)} + \epsilon_{rb}^{(kl)}, \tag{2.38}$$

The left hand side of Equations (2.37) and (2.38) are the DD prefits of pseudorange and carrier phase measurement values respectively, which can also be computed as:

$$y_{P,rb}^{(kl)} = P_{rb}^{(kl)} - \rho_{rb,0}^{(kl)} \tag{2.39}$$

$$y_{\Phi,rb}^{(kl)} = \Phi_{rb}^{(kl)} - \rho_{rb,0}^{(kl)} \tag{2.40}$$

By building up the linear DD equations of pseudorange and carrier phase as presented in Equation (2.37) and (2.38) for each satellite, the following compact form of navigation filter can be obtained:

$$\mathbf{y}_{DD} = \begin{bmatrix} G & 0 \\ G & \lambda \end{bmatrix} \begin{bmatrix} d\mathbf{r}_{rb} \\ \mathbf{n}_{DD} \end{bmatrix} + \begin{bmatrix} \varepsilon_{DD} \\ \epsilon_{DD} \end{bmatrix} \tag{2.41}$$

Where:

- $\mathbf{y}_{DD} = \begin{bmatrix} \mathbf{y}_{DD,P} \\ \mathbf{y}_{DD,\Phi} \end{bmatrix}$ is the vector with all the DD prefit values of pseudorange and carrier phase measurements concatenated together. Here $\mathbf{y}_{DD,P} = [y_{P,rb}^{(12)}, \ldots, y_{P,rb}^{(1m)}]^T$ are the DD pseudorange prefit residuals while $\mathbf{y}_{DD,\Phi} = [y_{\Phi,rb}^{(12)}, \ldots, y_{\Phi,rb}^{(1m)}]^T$ are the DD

carrier phase prefit residuals. Note that typically the reference satellite is chosen as the one with highest elevation angle, it is here indexed as 1 for simplicity. m is the total number of satellites and thus the length of this vector is $2(m-1)$.

- $\mathbf{n}_{DD} = [N_{rb}^{(12)}, \ldots, N_{rb}^{(1m)}]^T$ is the vector of only DD carrier phase integer ambiguities. Its length is m-1.
- $G = \begin{bmatrix} -(\mathbf{1}_r^{(1)} - \mathbf{1}_r^{(2)}) \\ \vdots \\ -(\mathbf{1}_r^{(1)} - \mathbf{1}_r^{(m)}) \end{bmatrix}$ is the RTK geometry matrix that is composed of the LOS vector difference between each satellite pair. The matrix size is (m-1, 3).
- $\begin{bmatrix} \varepsilon_{DD} \\ \epsilon_{DD} \end{bmatrix}$ is the vector of DD measurement noise terms for both pseudoranges and carrier phases, its length is also $2(m-1)$.

The next step is to solve the navigation filter presented in Equation (2.41). This can be done by applying the LSE or a Kalman filter similar to those used in the SPP navigation filter as described in 2.2.4. If there is only a single epoch of data, as in the case of snapshot positioning, the LSE based method should be chosen as there are no previous estimates available. In these scenarios, the SPP positioning solution based on this set of measurements are computed first. Then, these SPP coordinates are used as an initial guess to estimate the increment of the baseline vector $d\mathbf{r}_{rb}$. Regardless of which method was applied, the navigation solutions can be obtained with the following form:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{r}} \\ \hat{\mathbf{n}}_{DD} \end{bmatrix}; Q_{\hat{\mathbf{x}}} = \begin{bmatrix} Q_{\hat{\mathbf{r}}} & Q_{\hat{\mathbf{r}}\hat{\mathbf{n}}} \\ Q_{\hat{\mathbf{r}}\hat{\mathbf{n}}}^T & Q_{\hat{\mathbf{n}}} \end{bmatrix} \tag{2.42}$$

where:

- $\hat{\mathbf{x}}$ is the estimates of the unknown state vector that includes the baseline vector $\hat{\mathbf{r}} = \mathbf{r}_{rb,0} + \hat{d\mathbf{r}}_{rb}$ and the DD carrier phase ambiguities $\hat{\mathbf{n}}_{DD}$. It is noted that $\mathbf{r}_{rb,0}$ represents

the baseline vector computed using the initial guess of the rover receiver position and the known base receiver position.

- $Q_{\hat{\mathbf{x}}}$ represents the covariance matrix of the obtained parameter estimates.

This solution is named as float solution since the integer property of the DD carrier phase ambiguities $\mathbf{n}_{DD}$ has not been explored yet. In this regard, the ambiguities estimated in Equation (2.42) are called float ambiguities. In order to achieve high accuracy positioning, the correct integers inside the DD carrier phase measurements must be determined correctly. Only under such circumstances a so-called fixed solution can be obtained. The details about such Integer Ambiguity Resolution (IAR) procedure is presented in the following section.

2.2.6 Carrier Phase IAR

The previous discussion reveals how float solutions are generated without considering the integer constraints of the DD carrier phase ambiguities. This section focuses on how these carrier phase ambiguities can be fixed to the correct integers. There are mainly two approaches to solve this issue, the first method is to determine the ambiguity integer for each satellite pair independently (i.e. one after another separately). In contrast, the second method is to resolve this integer set all together as a group [31].

2.2.6.1 Satellite Independent IAR

The primary step of the first method is to obtain the so called Carrier Minus Code (CMC) DD measurements that can be computed based on the subtraction of Equation (2.38) from Equation (2.37):

$$y_{\Phi,rb}^{(kl)} - y_{P,rb}^{(kl)} = B_{rb}^{(kl)} + \varepsilon_{DD,CMC} \tag{2.43}$$

As it can be seen in Equation (2.39) and (2.40), the DD prefit residuals $y_{P,rb}^{(kl)}$ and $y_{\Phi,rb}^{(kl)}$ are computed by subtracting the same term $\rho_{rb,0}^{(kl)}$ to the DD pseudorange and carrier phase

measurements, the above equation can then be written as:

$$\Phi_{rb}^{(kl)} - P_{rb}^{(kl)} = \lambda \cdot N_{rb}^{(kl)} + \varepsilon_{DD,CMC} \tag{2.44}$$

The CMC combination completely removes the components related to geometric ranges from the original measurements and leaving only the DD carrier phase ambiguities and measurement noise, as shown on the right side of Equation 2.44. However, this removal is obtained at the price of merging pseudorange noises to the carrier phase ones, which results in a CMC measurement noise level that can be much larger than one wavelength, which is around 20 cm. The basic principle for the first method is to divide the CMC measurements by their wavelengths and then individually round off these values to their nearest integers as shown in Equation (2.45):

$$N_{rb}^{(kl)} = \left[\frac{\Phi_{rb}^{(kl)} - P_{rb}^{(kl)}}{\lambda} \right]_{roundoff} \tag{2.45}$$

However, the correctness of these integer estimates can not be guaranteed because of the enlarged magnitude of the CMC measurement noise $\varepsilon_{DD,CMC}$. In order to obtain reliable integers directly using such round-off computations, the error induced by thermal noise should be less than half of one wavelength. To evaluate the reliability of these rounded integers, we assume that the standard deviation of pseudorange measurement noise is one meter for both the rover and base receiver and the noise of carrier phase measurements are millimeter level that can be ignored compared to that of pseudoranges. For signals in the L1 band, their wavelengths are 19.05 cm. Consider that the pseudorange measurements between receivers and between satellites are all independent to each other, the standard deviation of DD pseudorange noise, and thus $\varepsilon_{DD,CMC}$, can reach 2 m, which is equivalent to more than 10 cycles of noise in the results computed by Equation (2.45). This is not acceptable for finding the correct integer in a single epoch.

There are two extra operations that can be applied in order to obtain more reliable integer

estimates based on Equation (2.45):

1. Reducing the noise level of the CMC measurements based on a smoothing process.

2. Create a virtual wavelength that is larger based on a linear combination of wavelengths from different frequencies.

The first operation aims at reducing the uncertainty of the numerator part of Equation (2.45) while the second operation seeks to enlarge the denominator part. The first operation relies on the fact that carrier phase integer ambiguities remain unchanged as long as the tracking loops are not interrupted, this indicates that the DD integer ambiguities $N_{rb}^{(kl)}$ also remain as constants if both the base station and rover receivers are locked in nominal tracking status. This operation, however, requires a collection of measurements across a wide time span since they are highly correlated over time. While traditional receivers adopting the close-loop architecture can perform a moving average of all the measurements collected in the designed time window, this operation is not applicable to snapshot receivers due to the nature of short duration of signal recordings.

The second operation explores the linear relationships between the wavelengths of different carrier frequencies. In this process, a virtual wavelength and the corresponding virtual measurements containing integer numbers of such new wavelengths are constructed based on original measurements collected from at least two different GNSS carrier frequencies [31, 32]. The Wide Lane (WL) combination [33] is the most commonly used operation for building such measurements with enlarged wavelengths. Such combination was initially constructed based on measurements of L1 and L2 frequency bands. With the availability of the new signals transmitted on L5 band, there are more similar linear combinations that can be built, such as the Extra Wide Lane (EWL) combination that explores the linear relationship between signals from L2 and L5 bands. The ambiguity integers of these combined measurements are fixed one after another following a cascading architecture and finally lead to the resolution of all the individual carrier phase measurements [34]. When measurements

from more than two frequencies are used for IAR, this technique is named as Three Carrier Ambiguity Resolution (TCAR) or Multiple Carrier Ambiguity Resolution (MCAR) [35, 36]. More details about the comparison of these methods are described in [37].

There are some drawbacks of this method as it intends to achieve IAR for each satellite based on measurement from the chosen satellite only. The main disadvantage is that it typically takes a long duration of signal recording before the correct ambiguity integers can be found for each satellite pair and thus yield a high accuracy solution. Despite the linear combinations of measurements from multiple frequencies can alleviate this issue, it requires the receiver to be capable of processing signals from these frequency bands, which implies the upgrades of antennas, receiver hardware and software. In order to avoid the long waiting times, a different approach is presented in the following subsection.

2.2.6.2 Collective IAR: LAMBDA

The main purpose of resolving the ambiguity integers in the DD carrier phase measurements is to obtain a high accuracy estimate of the baseline vector, since they are estimated jointly as shown in Equation (2.41). As mentioned before, $m-1$ carrier phase DD measurements can be constructed based on the m satellites that are available. Together with the 3 unknowns included in the baseline vector, it totals $m+2$ unknowns to be estimated. This is without considering the fact that only 3 of these unknowns are actually independent. For example, if 3 DD carrier phase integers are known, then an accurate baseline solution can be computed based on the 3 unambiguous but precise carrier phase measurements, which can then be used to deduce all the rest of the ambiguity integers. Similarly, if the 3 baseline parameters are known, then all the $m-1$ integers can be deduced as well [31].

This implies that there is a correlation among integer unknowns from all the satellite pairs, which has not been taken into account in the previous methods, since all these integers are estimated separately. In fact, with more than 4 satellites available, we can build up an over-determined navigation filter with redundant measurements. That is, the number of

DD carrier phase measurements is greater than the number of independent unknowns of the system.

Based on this property, IAR methods were developed by counting on all the available measurements together and the integer ambiguities are solved jointly as a set. These methods basically are based on solving Equation (2.41) using the Integer Least Squares (ILS) method. The main issue about ILS is that, as pointed out in [38], it typically does not have a closed form analytical solution. To tackle this issue, an integer search procedure with the aim of finding the minimum sum of squared residuals is typically used. This has inspired some of the most commonly used methods, including the Fast Ambiguity Search Filtering (FASF) method [39, 40], the Fast Ambiguity Resolution Approach (FARA) [41], the null space method [42], the Least-squares AMBiguity Decorrelation Adjustment (LAMBDA) [43], etc.

Among all the methods mentioned above, the LAMBDA method was proved to perform with highest efficiency [42, 37, 44]. The complete LAMBDA method can be generally divided into the following 4 steps [45]:

1. Obtain float solution. This step is already explained in section 2.2.5 as it is performed by a typical WLS without considering the integer properties of the DD ambiguity terms. The resulting float solutions, including the estimates of the unknowns and covariance matrix of these parameters, are expressed in Equation (2.42).

2. Integer estimation. This step aims at finding an optimal set of integers based on the float solutions according to a designed criteria.

3. Acceptance test. This is an optional step that decides whether or not to accept the integer solution computed in the previous step.

4. Obtain fixed solution. With the update from real-valued float ambiguities to the actual integer ambiguities, an update on the baseline solution is also performed correspondingly to obtain the final high accuracy positioning results.

The DD carrier phase ambiguities $\hat{\mathbf{n}}_{DD}$ in the float solution obtained after step 1 usually contain large uncertainties, usually up to tens of cycles, due to the inclusion of the less accurate pseudorange measurements in navigation filter as presented in Equation (2.41). Besides, there is a high correlation among these DD ambiguity parameters since the same reference satellite is used in their computation. These properties can be observed in their covariance matrix $Q_{\hat{\mathbf{n}}}$.

The key of the LAMBDA method is at step 2, where a decorrelation process is applied to the ambiguity parameters and then the set of integers is selected by the chosen strategy. There are different strategies that can be used in the search process, including the Integer Rounding (IR), Integer Bootstrapping (IB) and the ILS. IR and IB will not be discussed further due to their inferior efficiency. As for ILS, the main principle is based on the evaluation of the so-called closeness factor, which can be computed by:

$$C = (\mathbf{n} - \hat{\mathbf{n}})^T \mathbf{Q}_{\hat{\mathbf{n}}}^{-1} (\mathbf{n} - \hat{\mathbf{n}}) \tag{2.46}$$

where the vector $\mathbf{n}$ is a candidate set composed of integers for each DD carrier phase measurement, it can be seen as the coordinates of a grid point in a $m-1$ dimensional space. Note that for simplicity, the float solution $\hat{\mathbf{n}}_{DD}$ is expressed as $\hat{\mathbf{n}}$ by omitting the DD subscript. Due to the high correlation among the ambiguities, a constant closeness factor C will lead to a very elongated ellipsoid in the parameter space, that is, the number of possible candidates that we should search on in one direction is much larger than the other direction. This has a negative impact on the search efficiency as a lot of integer candidates need to be put into consideration. The decorrelation process is applied to alleviate this issue by a re-parametrization step. That is, instead of estimating the original ambiguity integers, we estimate another set of integers that can be transformed from the original integers but with much less correlation among the parameters, so that we can largely reduce the search space and increase computation efficiency. After the optimal integers for the new parameters are

identified, they are inverted back to the original parameters by a inverse transformation. The transformation matrix Z must satisfy the following requirements:

- Z must be composed of integers so that for any integer vector entry, the results are integers as well.
- Z must be invertible.
- Z^{-1} must have integer entries as well to ensure that the inverted parameters remain as integers.

The transformation process of the integer candidate vector $\mathbf{n}$ and the float ambiguities $\hat{\mathbf{n}}$ can be expressed as:

$$\mathbf{z} = \mathbf{Z}\mathbf{n}, \quad \hat{\mathbf{z}} = \mathbf{Z}\hat{\mathbf{n}} \tag{2.47}$$

where $\mathbf{z}$ and $\hat{\mathbf{z}}$ are the re-parametrization results of the integer candidate and the float ambiguities. The covariance of the new parameters resulted from float ambiguities can be computed as:

$$\mathbf{Q}_{\hat{\mathbf{z}}} = \mathbf{Z}\mathbf{Q}_{\hat{\mathbf{n}}}\mathbf{Z}^T \tag{2.48}$$

The objective of the decorrelation process is to obtain new parameters that are totally uncorrelated, which means $\mathbf{Q}_{\hat{\mathbf{z}}}$ should be a diagonal matrix. However, in practice, a perfect decorrelation is not possible because the $\mathbf{Z}$ matrix must satisfy the constraints of having only integer entries. Despite this obstacle, [44, 46] put more details about the process of obtaining the $\mathbf{Z}$ matrix that drives $\mathbf{Q}_{\hat{\mathbf{z}}}$ to be as near diagonal as possible. Based on the above transformation, the closeness factor for any $\mathbf{z}$ vector candidate can be now computed by:

$$C_{\hat{\mathbf{z}}}(\mathbf{z}) = (\mathbf{z} - \hat{\mathbf{z}})^T \mathbf{Q}_{\hat{\mathbf{z}}}^{-1} (\mathbf{z} - \hat{\mathbf{z}}) \tag{2.49}$$

Then, an evaluation process should be performed discretely for all the possible $\mathbf{z}$ vector

candidates in a search space that is defined by:

$$(\mathbf{z}-\hat{\mathbf{z}})^T\mathbf{Q}_{\hat{\mathbf{z}}}^{-1}(\mathbf{z}-\hat{\mathbf{z}}) \leq \chi^2 \tag{2.50}$$

where χ^2 is the constant that controls the volume of the search space. This value should be large enough to ensure the inclusion of the actual integer grid point in the search space, but at the meanwhile should be as small as possible to reduce the number of candidates in order to have a high computation efficiency. More details about the selection of the χ^2 value and analysis about the search volume can be found in [47, 48]. By searching through all these candidates, the one that returns the smallest closeness factor is picked out and appointed as the potential final integer solution.

Usually an acceptance test is also implemented in order to verify if the integer set obtained in the search process could be trusted. This test improves the reliability of the IAR process and brings more confidence to the final baseline solution [49]. Numerous different acceptance test approaches have been proposed in previous researches, including the difference test [50], the projector test [51], and the most commonly use ratio test [52, 53]. The ratio test is based on the computation of the ratio between the closeness factor of the second best integer set and that of the best integer set, which can be expressed by:

$$R = \frac{C_{\hat{\mathbf{z}}}(\mathbf{z}')}{C_{\hat{\mathbf{z}}}(\mathbf{z})} = \frac{(\mathbf{z}'-\hat{\mathbf{z}})^T\mathbf{Q}_{\hat{\mathbf{z}}}^{-1}(\mathbf{z}'-\hat{\mathbf{z}})}{(\mathbf{z}-\hat{\mathbf{z}})^T\mathbf{Q}_{\hat{\mathbf{z}}}^{-1}(\mathbf{z}-\hat{\mathbf{z}})} \tag{2.51}$$

where $\mathbf{z}'$ is the second best integers set obtained from the search process. The resulting ratio R is named as the LAMBDA Ratio Factor (LRF) and by comparing them with a predefined threshold value, we can decide whether or not the best integer set should be accepted.

Once the best set of re-parameterized integers passes the ratio test, they are transformed back to obtain the original integer vector $\check{\mathbf{n}}$ by the $\mathbf{Z}^{-1}$ matrix. They are then used to derive

the final high accuracy baseline solution by:

$$\check{\mathbf{r}} = \hat{\mathbf{r}} - Q_{\hat{\mathbf{r}}\hat{\mathbf{n}}} Q_{\hat{\mathbf{n}}}^{-1} (\hat{\mathbf{n}} - \check{\mathbf{n}}) \tag{2.52}$$

It must be pointed out that the threshold value R used in the ratio test is usually determined based on previous experiences, it is very popular to set this value to 2.5 or 3 in typical RTK processing [54]. However, there is a major problem of setting the threshold R to a constant value, depending on the model strength and the measurement qualities, either the false alarm rate or the failure rate can be unacceptably large [52].

A method based on a fixed failure rate was developed to tackle this issue [52, 55]. The threshold value in this method is determined by referring to a lookup table that was previously generated based on analysis of a large amount of data. Such lookup table depends on the model strength, i.e., number of ambiguities, frequencies, constellations, and noise level, etc., A tailor-made lookup table can be created by each user with their specific receivers in use, which could be applicable to snapshot positioning.

2.2.6.3 Summary

Although two approaches of achieving IAR were explained above, most of the time only the second method is used due to its high efficiency and reliability. Especially for the scenarios where only a short period of measurements is available and an instant IAR is desired, such as in the cases of snapshot positioning. In this dissertation, only the LAMBDA method is applied in the IAR process of the following high accuracy data processing.

2.2.7 PPP

PPP is another approach to obtain high accuracy positioning solutions by processing both the pseudorange and carrier phase measurements altogether. Different from RTK, PPP does not require real-time data streams of observables coming from nearby base stations

or any Virtual Reference Station (VRS) networks. In contrast, PPP requires orbit and clock correction parameters in real-time. As it can be expected, no DD measurements are computed and the error terms in the original observables have to be removed individually with the help of other products and models with the so-called State Space Representation (SSR). Typically, the following conditions should be satisfied in order to implement the PPP algorithm:

- Access to precise estimates of GNSS satellites orbits and clock errors. Those are usually computed in an external data processing center using raw data collected from stations distributed all over the world, thus these data is valid for all users globally.

- Measurements from at least two frequencies are required in order to have the capability to remove the ionospheric delay.

- The other error terms in Equation (2.9) and (2.27) should be modelled with accuracy at the centimeter level. Then, they can be either removed by an approximation or estimated as part of the unknowns by exploiting the model details. Particularly, some error terms that are usually neglected in SPP should be taken into consideration [2], such as the carrier phase windup effect, Antenna Phase Center (APC) variations, tidal motions on earth, the wet troposphere residual.

- A long period of data collection and processing is needed in order to decorrelate (i.e. separate) the ambiguity terms, which are treated as real-valued numbers, from other unknowns such as the receiver position, receiver clock and troposphere parameters. It can take up to one hour for the positioning solution to converge to decimeter/centimeter level with a single constellation [56], although this time can be slightly shortened by processing multiple GNSS constellations.

Although being a high accuracy positioning technique, the long convergence time of PPP has limited its application in many fields. It is especially contrary to the purpose of snapshot

positioning, which is designed to function with extremely short periods of data collection. Therefore, PPP technique is not further discussed as this dissertation focuses on snapshot positioning.

2.2.8 PPP-RTK

Section 2.2.5 presented the details of the RTK technique that depends upon the continuous input of GNSS measurements from a nearby reference station, whereas section 2.2.7 mentioned that PPP relies on external correction data in order to have precise models of measurement errors. A hybridisation of these two techniques, named as PPP-RTK, was first introduced in [57] in order to integrate the benefits from the two sides and limit their drawbacks as much as possible.

To better understand PPP-RTK, it is worth first analyzing what are the main drawbacks for RTK and PPP. The main disadvantage for RTK is the necessity of a data stream from another receiver that is constantly available and located not too far from the user receiver (typically less than few tens of kilometers). Besides, the RTK correction data stream produced by the reference receiver contains the rapidly changing pseudoranges and carrier phase measurements, this can cause a huge burden on the server and transmitter side due to the high bandwidth demand for data communication. These constraints have limited the users to adopt the RTK technique only in local regions such as around cities or smart farms. As for PPP, the biggest drawbacks are the long convergence time and the requirement of tracking two frequencies. The latter, complicates the hardware, including the antenna and front-end module.

The navigation filter of the PPP-RTK technique was developed mainly on the basis of PPP in the sense that only undifferenced measurements are used and the error terms are removed individually [57, 58]. In other words, both PPP and PPP-RTK utilise State Space Representation (SSR) that broadcast a single stream containing separated error components for all user receivers. Its main difference from standard PPP is that the error parameters are

generated based on signals collected by a finer distributed Continuously Operating Reference Station (CORS), similar to that of RTK. There are also more types of parameters in the PPP-RTK products, such as the precise ionosphere parameters, which enables this technique to be used for single-frequency receivers as well, and the Uncalibrated Phase Delay (UPD) (also named as phase bias) parameters, which allows cancelling the fractional part of the carrier phase ambiguity terms and lead to a possible IAR with undifferenced measurements [59]. In contrast, RTK adopts the Observable Space Representation (OSR) strategy where all error terms are cancelled out together as the transmitted code and carrier phase measurements data contains the lump sum of all these error terms.

2.2.9 AGNSS

As mentioned in section 2.2.3, traditional GNSS positioning engine requires the decoding of navigation message symbols that are modulated in the collected signal. Especially the bits that contain the information of signal transmission time and satellite and orbit parameters that are necessary for the computation of satellite positions. However, this implies a long waiting time since these navigation data are transmitted at a low bit rate.

For example, the GPS L1 C/A signal carries the navigation message with a bit rate of 50 Hz. A full frame lasts 30 seconds and include 5 subframes that each last 6 ms. Even in the most optimistic scenario, where the first 3 subframes that contain the satellite clock and ephemeris data are received first, it still takes at least 18 seconds of data collection. In contrast, worse scenario arises in practice when the signal strength is weak and the error rate increases in the demodulation process, resulting in an even longer waiting time. This certainly does not lead to a satisfying TTFF estimate.

Assisted GNSS (AGNSS) was developed in order to tackle this issue. This technique relies on the assistance data that is obtained from external resources, typically a communication module is required and the data are collected from internet. A very successful application of this technique is the smartphone positioning, where a mobile communication module is

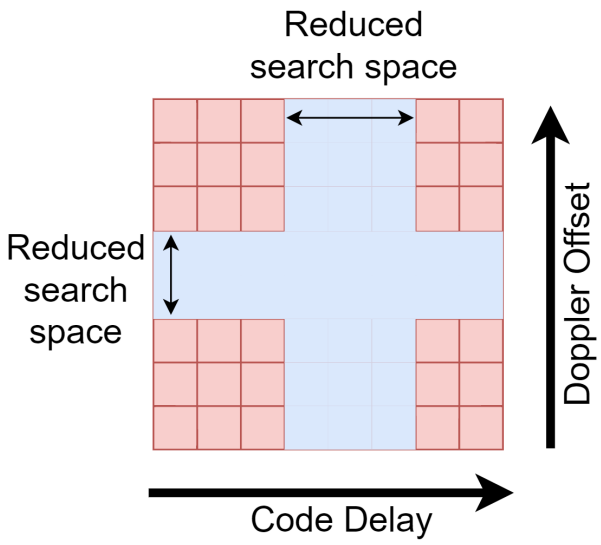

Figure 2.7: Reduced acquisition search space based on AGNSS

present by default and retrieves assistance data that are later fed to the GNSS module of the device.

Besides the satellite ephemeris and clock parameters, the assistance data should include other information such as Ionosphere correction parameters, rough knowledge about signal user position and time as well. With these data available, the AGNSS technique improves the receiver performance in the following ways:

- **Reducing the search space**. The knowledge about time and receiver position provides a rough estimate about the transmission time and thus the code phase of the received signal. Similarly, the Doppler frequency offset can be roughly estimated based on the computation of satellite motion using the information available in the assistance data. These coarse information accelerate the acquisition process by removing the parameter candidates that do not fall in the region of the rough estimates. As illustrated in 2.7, only a portion of the whole search space is considered and thus the time required for computation is reduced.

- **Preventing data demodulation**. Since the necessary navigation data is already obtained through the communication module, there is no need to wait for the demod-

ulation process. As long as the raw measurements are generated, the navigation filter can start processing them directly.

Chapter 3

Snapshot GNSS Techniques

In recent years, snapshot positioning technique have attracted great attention in the GNSS community [21, 60, 61] thanks to its advantages of lower cost and lower power consumption compared to conventional GNSS receivers. Section 1.2 mentioned that only a very brief interval of the received satellite signal is used for snapshot positioning [6]. This has introduced some technical difficulties in the implementation of such snapshot positioning. In terms of the overall workflow, both the baseband processing and PVT computation functional blocks require a special design tailored to the short duration of collected snapshot signals. For the baseband signal processing block, contrary to traditional close loop sequential signal processing workflow, snapshot positioning follows the open-loop batch processing architecture. Although received less attention from the GNSS community, the open-loop architecture shows advantages in several aspects [17], especially in the capability to quickly generate GNSS observables. As for the PVT computation module, extra steps are required to ensure that complete measurements are successfully generated. In particular, the navigation filter is essentially different from the traditional ones due to the fact that only coarse time information is available.

The overall workflow of snapshot GNSS positioning is illustrated in Figure 3.1. It starts with the GNSS signals being captured by the antenna connected to the snapshot receiver hardware. This part is equivalent to the RF front end of traditional receivers and converts the analog signals to digital bits that are down converted to the baseband. The resulting

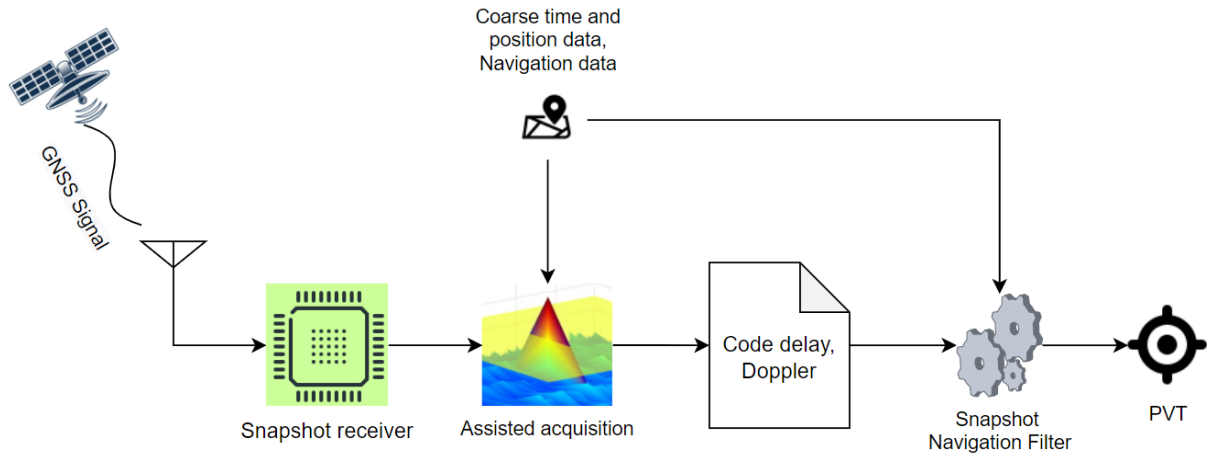

Figure 3.1: Overall workflow of conventional snapshot positioning

digital bits are then fed to the assisted acquisition module and the estimates of code phase and Doppler offset of the current snapshot signal are generated with the help of assistance data. Such assistance data are further used in the snapshot navigation filter to compute the PVT solution based on the estimates obtained from the acquisition module.

The remaining part of this chapter provides an insight about the typical techniques used for snapshot positioning. Section 3.1 introduces the details of the baseband signal processing module designed for snapshot data. The resulting snapshot measurements and related parameters are described in section 3.2. Sections 3.3 and 3.4 provide further details about the snapshot PVT computation module. The former describes the solution to the full period ambiguity issue, which is typically faced by receivers employing an open-loop architecture. The latter outlines the Coarse Time Filter (CTF) that is typically applied to measurements without accurate time tags, i.e., when TOW information is not precise enough. Note that this chapter focuses on conventional snapshot position techniques targeting meter-level accuracy. The more innovative algorithms about high accuracy snapshot positioning are provided in chapter 4 and 5.

3.1 Baseband Processing

The baseband processing module based on an open-loop architecture typically applies an assisted Multi Hypothesis (MH) acquisition process, without further deploying a signal tracking step. As its name suggests, this acquisition step differs from the traditional acquisition steps in the following characteristics:

- The acquisition search space is reduced with the extensive use of assistance data, including the coarse time and position information, as mentioned in section 2.2.9.
- The acquisition process makes multiple hypotheses about the data symbols that are encoded in the received signal in order to ensure the maximum energy in the CAF results.
- Carrier phase estimates are computed along with the code phase, in case they are required by the following PVT computation module.

Section 2.2.1 introduced the mathematical details applied in traditional acquisition modules. However, the previous discussion does not take into consideration the bit transitions that could happen in the middle of the signal that is used in the correlation operation. This situation happens due to the existence of the navigation data bits d_i in the collected signal, as represented in Equation (2.1). These data bits could experience sign changes and they usually last longer than the PRN code (also known as the primary code) period. For example, GPS L1 C/A signal contains navigation data that spans over 20 ms for each bit, while its PRN code period is just 1 ms. This ensures that there is always a sequence of at least 10 PRN code periods free from bit sign transitions [62].

The total integration time used in the correlation operation can be represented by the multiplication of the coherent integration time t_{coh} and the non-coherent integration factor N_{nc}. When the receivers experience a harsh environment with, for instance, severe obstruction of signal, the receiver should be capable of acquiring the signals with high sensitivity.

In order to achieve high sensitivity, relatively long duration of signal should be used, which implies greater t_{coh} and N_{nc}.

The coherent integration is the ideal correlation that makes use of all the data in the duration of t_{coh} with 100% gain efficiency [63]. The correlation result of coherent integration is represented in Equation (2.4) by the complex number $Y(\tau, F_D)$. It contains the correlation energy computed as $Y_I^2(\tau, F_D) + Y_Q^2(\tau, F_D)$, which was also used for the detection of the satellites in Equation (2.5).

The non coherent integration sums such correlation energy magnitude computed for all the N_{nc} pieces of coherently integrated signals. This way of processing is introduced mainly in order to reduce the impact of possible bit transitions occurring in the middle of the total integration time. It also helps to reduce the negative influences of potential mismatches of Doppler frequency [63, 64]. The settings of t_{coh} and N_{nc} are the design parameters that can be controlled by the receiver manufacturers. For snapshot receivers that usually have very limited total length of signal collection, it is important to efficiently use all the available data.

This dissertation tackles the algorithmic challenges faced by acquisition with fully coherent integration. The longer the coherent integration time, the higher sensitivity is possible and ultimately achieving a better positioning performance. This choice of adopting fully coherent integration also indicates that $N_{nc} = 1$ and t_{coh} is identical to the total snapshot length that is used for acquisition. Even though the non-coherent strategy is also applicable for snapshot positioning, it will not be further discussed.

The main challenge of extending the coherent integration time is the existence of the navigation data bits or secondary code symbols. It is recalled that these bits and symbols are not yet known for the receiver at the time of signal reception, unless provided by some external assistance services, such as the one mentioned in [65]. In order to tackle this challenge, the MH acquisition technique was developed. This method modifies the local replicas by multiplying the PRN sequence with a set of designed data symbols and results

in the following CAF computation equation:

$$\begin{aligned} Y_I(\tau, F_D) &= \frac{1}{N}\sum_{n=0}^{N-1} r[n]\cos(2\pi F_D n)\, c[n-\tau]d[n-\tau] \\ Y_Q(\tau, F_D) &= \frac{1}{N}\sum_{n=0}^{N-1} r[n]\sin(2\pi F_D n)\, c[n-\tau]d[n-\tau]. \end{aligned} \tag{3.1}$$

Where $d[n-\tau]$ represents the designed data symbols based on a hypothesis of the actually encoded values. Depending on the length of the integration time and the duration of one bit and one symbol, the total number of hypotheses N_{hyp} also varies.

Regarding the construction of all these hypotheses of data symbols, there are two scenarios that should be considered differently depending on the structure of the GNSS signals. The first case is for the signals with encoded navigation messages, whose data bits are generally unknown at the snapshot acquisition stage, unless provided by some external assistance services [65]. This type of signals are named as the **data signals** in this dissertation. The second case is for the **pilot signals** that do not contain navigation messages. However, these more modern signals usually apply a secondary code sequence above the primary codes and the exact sequence of such secondary codes are known beforehand as they are defined in their ICDs. The construction of hypotheses of $d[n-\tau]$ for these two scenarios are discussed in the following two subsections, respectively.

3.1.1 Data Signals

Due to the fact that navigation messages are generally unknown when no related assistance is provided, the MH acquisition module has to blindly generate all the possible hypotheses that could be encountered. That is, all the possible data bits combinations that could have been modulated to the received signal should be taken into consideration.

An illustrative example for this scenario can be generated with the popular GPS L1 C/A signal. Figure 3.2 shows the actual data bit values of GPS L1 C/A signal for a 20 ms

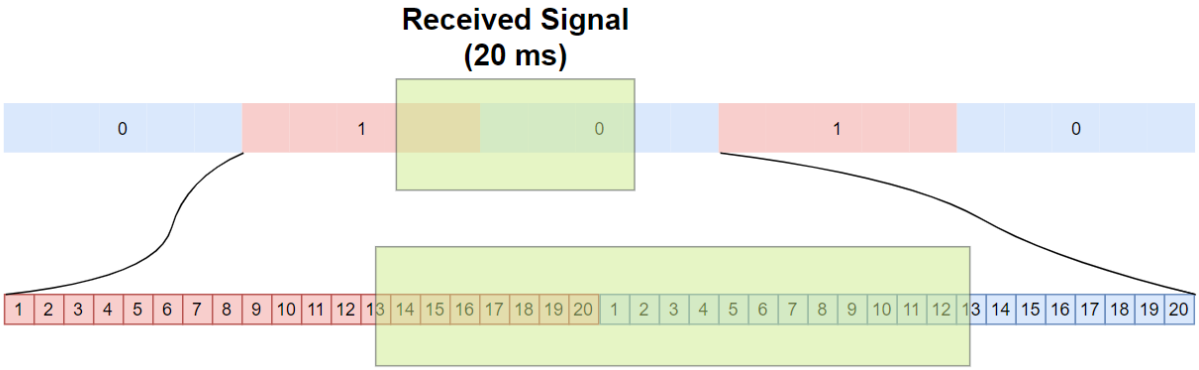

Figure 3.2: Example of actual encoded data bits of GPS L1 C/A signal for a 20 ms snapshot signal. Blue boxes represent bits of 0 and the pink boxes represent bits of 1.

snapshot. In the upper panel, the blue boxes represent bits of 0 and the pink boxes represent bits of 1, each of these bits has a duration of 20 ms, as depicted by the length of these boxes. In the lower panel, the region near the received signal is shown with more details, with each data bit represented by 20 symbols that each lasts 1 ms. Each symbol is represented by a box that covers a whole primary code period and the numbers inside the box are the index of the symbols counted from the start of the current bit.

The total number of encoded data bits N_b that needs to be considered can be computed by:

$$N_b = \text{ceil}\left\{\frac{t_{coh}}{T_b}\right\} + 1 \tag{3.2}$$

Where T_b represents the duration of one data bit. For the example shown in Figure 3.2, $t_{coh} = 20\ ms$ as we assume that the snapshot acquisition module runs in fully coherent mode. As a result there are a total of 2 bits that should be considered. This is shown in the green area of Figure 3.2, in which the received snapshot signal indeed spans over 2 bits that are encoded by [1, 0]. Note that the computation shown in Equation (3.2) includes an addition of 1, this is in order to guarantee that the whole duration of the collected signal can be correlated properly.

For a sequence with N_b bits, a total of 2^{N_b} combinations of data bits are possible to appear in the signal within the considered correlation time. For the example shown in Figure 3.2,

besides the correct combination of [1, 0], 3 other combinations could also have been made as [1, 1], [0, 1], [0, 0]. Regarding the construction of hypotheses, two properties about the data bits must be pointed out first:

- The duration of one data bit is always an integer number of times the duration of one primary code period. The ratio between them is denoted as N_r. For example, $N_r = 20$ for GPS L1 C/A signal.
- The edges of the encoded data bits are also well aligned to the primary code edges. That is, the beginning of any data bit is always synchronised to a PRN code edge.

For these reasons, the hypotheses can be made for each symbol that has a duration of one primary code period. This means that each data bit combination should be represented by a longer sequence of symbols that has a unit length of one primary code period, similar to that shown in the lower panel of Figure 3.2. For the correct data bit combination of this example, i.e., [1, 0], there are 20 hypotheses that can be made since the bit transition from 1 (pink box) to 0 (blue box) can happen at any of the PRN code edges. The assigned symbols for the v-th code period under the u-th hypothesis can be represented by Equation (3.3):

$$H_v^{(u)} = \begin{cases} 1, & v \leq u \\ 0, & v > u \end{cases}. \tag{3.3}$$

All the hypotheses of symbols for this example with the [1, 0] combination are listed in Figure 3.3. Similarly, the wrong data bit combination of [0, 1] will result in the hypotheses results of:

$$H_v^{(u)} = \begin{cases} 0, & v \leq u \\ 1, & v > u \end{cases}. \tag{3.4}$$

The combinations of [0, 0] and [1, 1] will result in hypotheses represented by $H_v^{(u)} = 0$ and $H_v^{(u)} = 1$ respectively. They have simpler expressions as there is no bit transition during the entire correlation time under these two combinations. These expressions are concluded

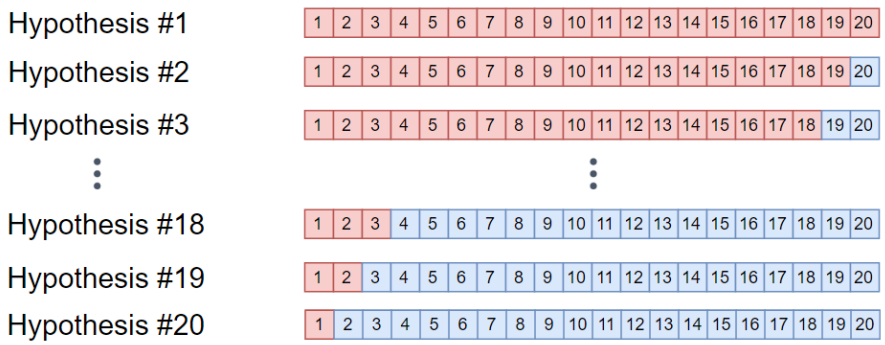

Figure 3.3: Hypotheses of data bit symbols based on the [1, 0] bit combination for a 20 ms snapshot GPS L1 C/A signal

Table 3.1: Hypotheses expressions and indices of different bit combinations for the example of 20 ms GPS L1 C/A snapshot signal.

Bit combination	Hypotheses expression	Hypotheses indices range
[1, 0]	$H_v^{(u)} = \begin{cases} 1, & v \leq u \\ 0, & v > u \end{cases}$.	1∼20
[0, 1]	$H_v^{(u)} = \begin{cases} 0, & v \leq u \\ 1, & v > u \end{cases}$.	21∼40
[0, 0]	$H_v^{(u)} = 0$	41∼60
[1, 1]	$H_v^{(u)} = 1$	61∼80

altogether in Table 3.1.

The total number of hypotheses now can be computed by:

$$N_{hyp} = 2^{N_b} \cdot N_r \tag{3.5}$$

For the case of the explanatory example, $N_{hyp} = 2^2 \cdot 20 = 80$. The Hypothesis Index (HI) ranges for all the combinations are also listed in Table 3.1. Note that for the combinations where bit transitions are absent, the data symbols under all the hypotheses of such combination are identical. For example, the data symbols for the combination [0, 0] are always a series of 0s, no matter the HI value, the same goes for the [1, 1] combination. In practice, a single hypothesis is enough for each of such combinations, instead of 20, thus resulting in the total number of hypotheses to be less than 80 (42 for this example as the 4). However,

in this dissertation, it is decided to keep the consistency of assigning N_r hypotheses for each combination, this results in a simple conversion from HI to the u index value, which can be expressed as:

$$u = HI \mod N_r \tag{3.6}$$

As it can be seen, u is the module-N_r result of the HI value.

In general, the expression of the assigned data bits $H_v^{(u)}$ can become more complicated when longer integration times are considered. This is due to the fact that multiple bit transitions can, and likely will, take place for integration times longer than one single bit period. The total number of hypotheses N_{hyp} correspondingly grows in such scenarios.

With all the hypotheses about the encoded symbols been made, the CAF of snapshot signals with 20 ms of coherent integration time can be represented as:

$$\begin{aligned} Y_I(\tau, F_D, u) &= \sum_{v=1}^{20} H_v^{(u)} \cdot Y_{Iv}(\tau, F_D), \\ Y_Q(\tau, F_D, u) &= \sum_{v=1}^{20} H_v^{(u)} \cdot Y_{Qv}(\tau, F_D). \end{aligned} \tag{3.7}$$

Where Y_{Iv} and Y_{Qv} are respectively the In-phase and Quadrature components of CAF for one primary code period as computed by Equation (2.3).

Similar to the traditional acquisition modules, by searching for the maximum magnitude of the CAF, the optimal code phase and Doppler offset parameters can be determined. The main difference of MH acquisition is that the optimal HI of encoded symbols is also detected. It is possible that there are more than one hypothesis that can lead to the CAF peak magnitude, this will be further explained in the following chapters. Besides, an interpolation step needs to be implemented for both the code phase and Doppler offset in order to obtain sufficiently precise measurements for the navigation filter [17], this results in the final estimates of code phase ($\hat{\tau}$) and Doppler shift ($\hat{F}_D$). In addition, although not needed for conventional snapshot positioning, the carrier phase at the beginning of the collected signal under the

optimal hypothesis can be computed by the following equation based on these interpolated code phase and Doppler offset estimates:

$$\hat{\varphi} = \arctan\left(\frac{Y_Q(\hat{\tau}, \hat{F}_D, u)}{Y_I(\hat{\tau}, \hat{F}_D, u)}\right). \tag{3.8}$$

3.1.2 Pilot Signals

Pilot signals do not contain navigation messages bits as the data signals that are mentioned in section 3.1.1. This effectively simplifies the process of construction of symbol hypotheses. However, the secondary codes that are usually encoded on top of the PRN codes should be considered. The construction of hypotheses about such secondary codes can be treated in the same manner as for the navigation data bits mentioned in the previous subsection. The main difference is that the exact sequence of the complete secondary code symbols is already known in advance, as they are defined in the ICD documents of the GNSS constellations. This implies that as long as the transmission time of the signal is known, all the corresponding secondary code symbols can be deduced. In addition, as it can be seen in Table 2.1, the duration of one secondary code symbol is equal to one full primary code period. This allows reducing the number of hypotheses and the corresponding computation workload. This approach consists of making hypotheses based on the position of the secondary code symbols over which the recorded signal started, i.e., the Secondary Code Index (SCI). With the hypothetical SCI and the total duration of the signal, all the corresponding symbols can be deduced.

Figure 3.4 shows an example of the possible secondary code symbols modulated in Galileo E1C signal for a 20 ms snapshot. The square boxes represent the 25 symbols of a full secondary code sequence, among which the shaded ones represent the symbols of ones and the blank ones represent symbols of zeros. The numbers inside these boxes show their corresponding SCI values. The collected snapshot signal, whose span is represented by the green box, starts from the box with SCI equal to 9. Similar to the cases for data signals, the

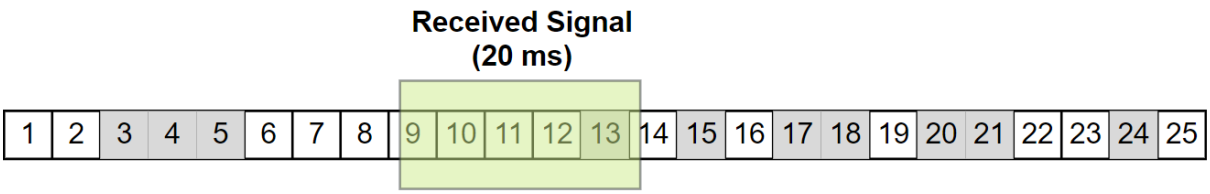

Figure 3.4: Example of secondary code symbols of Galileo E1C signal for a 20 ms snapshot signal, the shaded boxes represent the symbols of ones and the blank ones represent symbols of zeros. The numbers inside these boxes show their corresponding SCI values.

number of symbols required can be computed based on Equation 3.9.

$$N_S = \text{ceil}\left\{\frac{t_{coh}}{T_S}\right\} + 1 \tag{3.9}$$

where T_S is the duration of one secondary code symbol. For this example, $N_S = 6$ is required for the 20 ms of coherent integration time since the secondary code symbol lasts 4 ms for Galileo E1C signal. With the a priori knowledge that the $CS25_1$ code is applied, the actual symbols for the whole snapshot duration can be deduced as [0, 0, 0, 0, 1, 0]. When attempting to acquire this signal, a total of $2^{N_S} = 64$ hypotheses are supposed to be made if conventional method of enumerating all possible sign combinations is applied [66], just as in the data signal scenarios. However, by exploiting the fact that it is a pilot signal with known secondary code sequence, only 25 hypotheses are needed and each hypothesis corresponds to a unique SCI value. The total number of hypotheses based on this new approach is equivalent to the secondary code length N_{SC}.

For signals with extremely long secondary code sequences, such as the BDS B1C signal, which contains 1800 symbols and lasts 18 seconds, it is possible to truncate this sequence to shorter ones located near the rough transmission time that can be obtained with the assistance of the coarse time information. For that, we need to ensure that the actual secondary code symbols are fully comprised within this truncated sequence by setting a proper window size (e.g., 200 ms).

After computing the correlations for all the SCI hypotheses in the MH acquisition process,

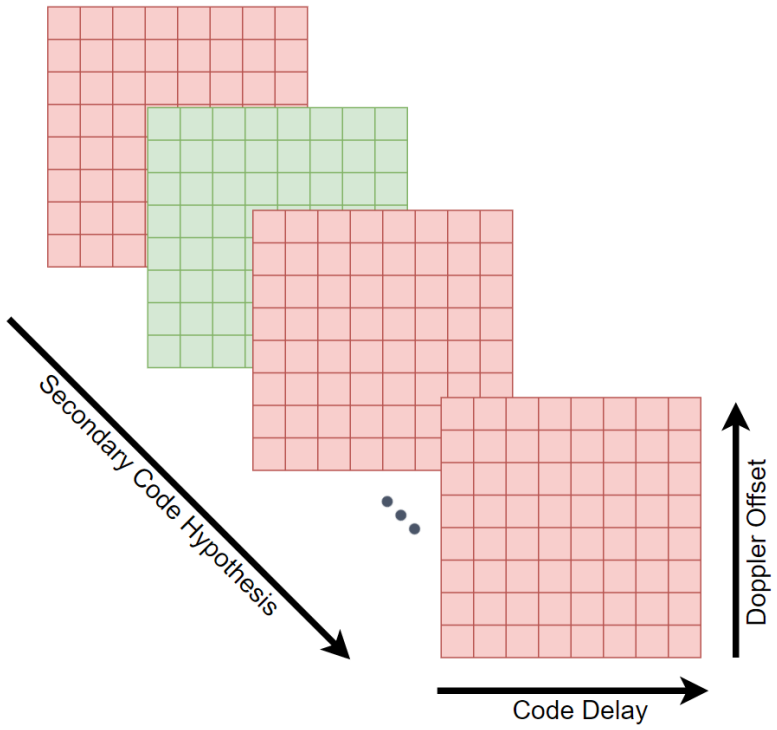

Figure 3.5: Three-dimensional search space of MH acquisition for snapshot signals.

CAF results with 3 dimensions are obtained, representing code phase, Doppler shift and the SCI value respectively, as illustrated in Figure 3.5. It results in a multi-layer structure due to the discrete integer SCI values. A search and detect procedure is implemented in order to find the best estimates of the parameters. Note that a local CAF energy peak can be found within each layer, but since the wrong SCI values most of the time lead to wrong encoded symbols, only the ones that have the maximum energy (shown in green) among all layers lead to the final acquisition output. It should be pointed out that when the snapshot signal length is relatively short, it it possible that the maximum CAF energy exist in more than one layer, as their magnitudes are identical. This problem is referred to as the Data Bit Ambiguity (DBA) issue and will be discussed with more details in Chapter 5.

3.2 Snapshot Measurements

Snapshot Measurements are the results obtained based on the search of maximum CAF magnitude computed in the MH acquisition process aforementioned in section 3.1. There are mainly the following 4 types of measurements that can be generated from snapshot signals:

- code phase τ. Code phase is the principle measurement required to compute the user PVT in the navigation engine. These code phase values are fractions of one code period that essentially show the time difference between the epoch when the received signal is transmitted and the starting epoch of the code period in which the transmission epoch locates. These fractional values greatly differ from the typical pseudorange measurements generated by traditional receivers based on close-loop architecture, as presented in Section 2.2.3. These differences will be further discussed in Section 3.3.

- Doppler Offset F_D. The Doppler offset is originated from the receiver to satellite relative motion, and the estimated frequency shift accounts also for the receiver clock drift.

- Carrier phase ϕ. Contrary to the traditional GNSS receivers that generate the carrier phase measurements based on continuous accumulation of cycles of the local oscillator, snapshot carrier phase measurements are fractional values of one full cycle that is computed based on Equation (2.6).

- u index. The hypotheses constructed using the approaches mentioned in section 3.1.1 and 3.1.2 lead to different CAF magnitudes. The identification of the maximum magnitude leads to the correct HIs. For data signals, the u value is computed based on these HIs with Equation (3.6), whereas for pilot signals this refers to the SCI value.

An important remark is that the first three measurements (τ, F_D and ϕ) are not accurately time-tagged since there is no accurate information about absolute timing at the stage

when the GNSS signals are received by the snapshot receiver. Besides, it is not possible for the local clock to be synchronised with the satellite clock in such a short time. In contrast, measurements generated by traditional receivers are time-tagged with a nanosecond accuracy.

For snapshot receivers that are being developed nowadays, only the code phase and Doppler offset measurements are used for positioning. This limits the resulting accuracy to the meter level accuracy. The other two types of measurements (ϕ and u) mainly concern the potential high accuracy algorithms. In the present dissertation, a data group including all of these 4 parameters is defined as a 'set' of snapshot measurements. For each satellite, the MH acquisition process may result in several sets of measurements, each set corresponds to the results under a certain HI, even though their code phase and Doppler offset parameters are identical.

3.3 Full Period Ambiguity

Traditional GNSS receivers compute full pseudorange measurements based on Equation (2.8) within Section 2.2.3. The receiver time t_{rx} is a value generated by the local clock and it contains a common clock bias (i.e. an offset) that is estimated in the navigation filter. This means that even if the pseudoranges are computed with a t_{rx} value that is inaccurate, the navigation filter can still obtain an accurate positioning solution while fixing this receiver time inaccuracy. In fact, it is the signal transmission time t_{tr} that is critical in order to generate valid pseudoranges. However, these transmission times are obtained for traditional receivers in the processes of bit synchronisation, frame synchronisation and navigation message data decoding, which are not available for snapshot signal due to their short duration.

Section 3.2 mentioned that the code phase measurements obtained from the MH acquisition results are only the fractional values. In order to be used for positioning, each of these fractional values need to be specially compensated in order to retrieve the pseudo

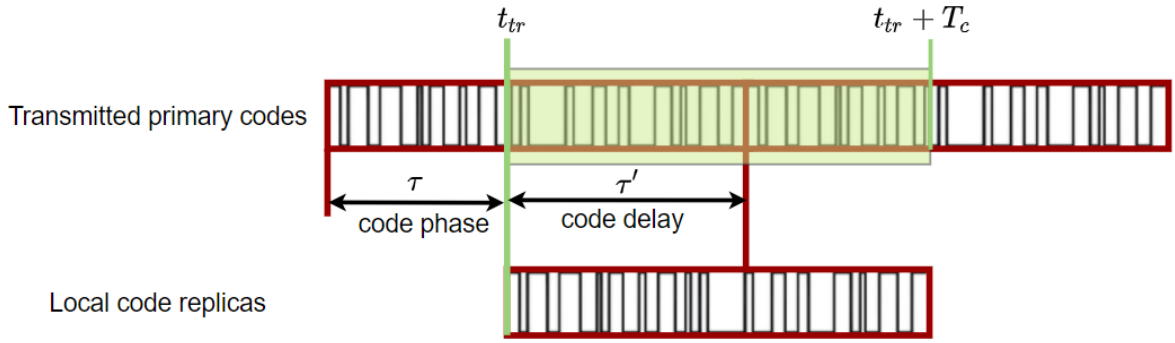

Figure 3.6: Illustration of relationship between code phase and code delay, $t_[tr]$, τ, τ' and T_c respectively represent the satellite transmission time, code phase, code delay and the primary code period.

distance between the user receiver and each different satellite, that is, to generate the full pseudoranges for each satellite.

For the convenience of the following computations, the code delay parameter τ' is defined as the remaining part of one primary code period after the code phase τ, as illustrated in Figure 3.6. The part shaded in green represents the first code period length of the collected signal that is transmitted at t_{tr}. Each full primary code period is bounded by a red box and as it can be seen, the code phase at transmission time t_{tr} is τ. Whenever the code phase is estimated during the acquisition process, the code delay can be directly obtained by $T_c - \tau$, with T_c representing the duration of the primary code. For instance, if the code delay of a GPS L1 C/A signal is estimated as $\tau = 0.3\ ms$, then the corresponding code delay $\tau' = 0.7\ ms$ since the primary code period $T_c = 1\ ms$.

The GNSS ICD defines that the primary code edges are well aligned with the system time scale. This implies that for any transmission time t_{tr}, the code phase τ can be determined by computing its remainder to the primary code period T_c, following the relationship:

$$t_{tr}^{(k)} = \tau^{(k)} + X_c^{(k)} \cdot T_c \tag{3.10}$$

where $X_c^{(k)}$ is an integer that corresponds to the transmission time and k stands for the

satellite index. In the present dissertation, the GPS time is applied as the default time scale, and to represent any specific epoch in time, a tuple of Week number (W_n) and TOW is used.

Similarly, any receiver time can be modelled by:

$$t_{rx} = \tau_r + Y_c \cdot T_c \tag{3.11}$$

where Y_c is an integer that corresponds to the receiver time and τ_r is the fractional part of the chosen time. As mentioned before, the selection of the receiver time does not impact the accuracy of PVT computation. Hence, the t_{rx} values can be chosen at the edge of the primary codes so that τ_r is equal to zero and thus simplifies the following calculation. According to Equation (2.8), the full pseudoranges of satellite k can be computed as:

$$P^{(k)} = c \cdot (t_{rx} - t_{tr}^{(k)}) = c \cdot (\tau'^{(k)} + N_c^{(k)} \cdot T_c) \tag{3.12}$$

Where $\tau'^{(k)}$ is the code delay for satellite k and $N_c^{(k)}$ is defined as the full period integer:

$$N_c^{(k)} = Y_c - X_c^{(k)} - 1 \tag{3.13}$$

that sums up all the integer parts from Equations (3.10) and (3.11).

The new expression of Equation (3.12) shows that for each satellite, there is an integer number of full code periods that should be estimated in order to obtain the full pseudoranges. Figure 3.7 shows the relationships of such integers for all satellites. The red vertical lines represent the different transmission times for each satellite and the green line represents the common reception time for all of them. The time difference between them are the flight times which can be expressed as $\tau'^{(k)} + N_c^{(k)} \cdot T_c$ and by multiplying the speed of light, the full pseudoranges are obtained as described in Equation (3.12). The main remaining issue is that the integer $N_c^{(k)}$ is still unknown for each different satellite. The process of finding these integer numbers is referred to as resolving the full period ambiguities, or code ambiguities.

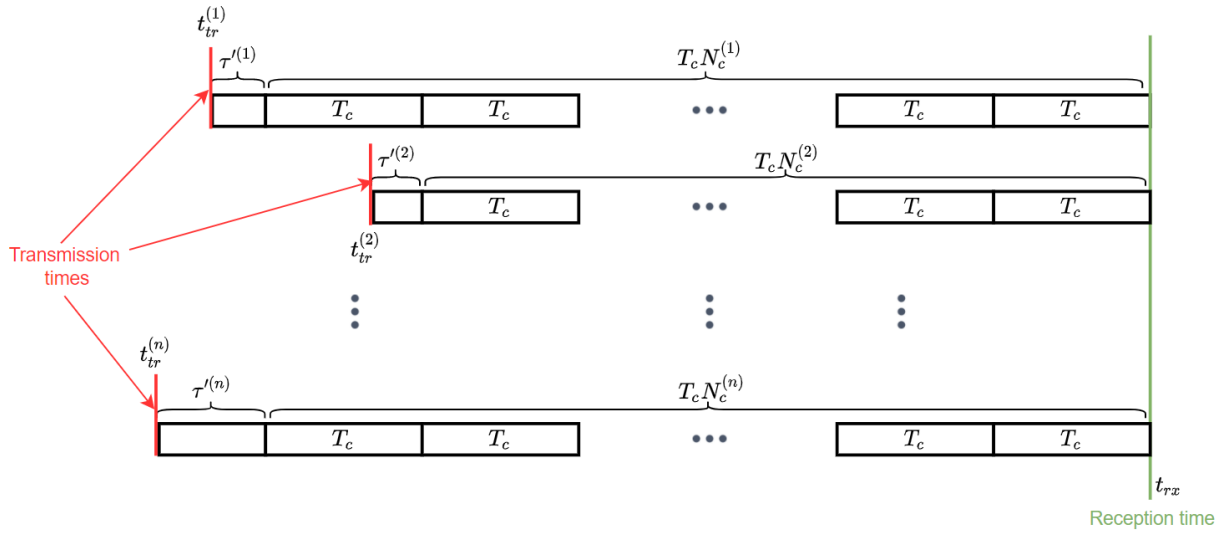

Figure 3.7: Full code period ambiguities for all satellites due to different transmission times.

Many previous researches focused on GPS L1 C/A signal only and thus also termed this process as resolving the 1-ms ambiguities [67, 68, 69].

The key to find the proper integer $N_c^{(k)}$ for each satellite is by analyzing the flight times. The primary component of the flight time stems from the geometric distance ρ between the receiver position at the reception time and the satellite position at the transmission time. While other factors such as the atmospheric effects and hardware delays may also impact the total flight time, the influence is well modelled [2] and the residual errors after applying standard corrections are at meter level in distance, which is equivalent to nanoseconds of flight time difference and can be safely omitted since their magnitudes are orders lower than a code period.

In order to quantitatively evaluate this difference in distance after correcting these minor effects from the full pseudorange, a parameter $d^{(k)}$ can be defined as:

$$d^{(k)} = P^{(k)} - M^{(k)} = P^{(k)} - \alpha_i I - T + c \cdot dt^{(k)} \tag{3.14}$$

where $M^{(k)}$ represents the sum of all modelled terms for satellite k. As it can be seen on the right side of Equation (3.14), the impacts coming from Ionosphere, DCB, Troposphere and

satellite clock errors are all corrected using well-known models. Based on the pseudorange model presented in Equation (2.9), this distance can be further modelled as:

$$d^{(k)} = \rho^{(k)} + c \cdot \delta t_{rec} + s^{(k)} \tag{3.15}$$

where $s^{(k)}$ is a lump sum of the pseudorange multipath error, receiver noise term and the residual errors after applying the correction models for the minor effects as performed in Equation (3.14). The magnitude of $s^{(k)}$ is usually at meter level, although it depends on the measurement quality and model accuracy. Since $\rho^{(k)} = \left\|\mathbf{r}^{(k)} - \mathbf{r}_{rec}\right\|$, assuming the coarse estimates of the receiver position $\mathbf{r}_{c,rec}$ and satellite positions $\mathbf{r}_c^{(k)}$ are known, by linking up Equation (3.14) and (3.15) and replacing $P^{(k)}$ based on Equation (3.12), the following equation is obtained to estimate the full period integers $N_c^{(k)}$:

$$N_c^{(k)} \cdot T_c + \tau'^{(k)} = \frac{\left\|\mathbf{r}_c^{(k)} - \mathbf{r}_{c,rec}\right\|}{c} + \delta t_{rec} + \frac{M^{(k)} + s^{(k)}}{c} \tag{3.16}$$

The only unknowns in this equation are the full period integers $N_c^{(k)}$, the receiver clock bias δt_{rec} that is common to all satellites and the small residual terms $s^{(k)}$. By putting the code delay estimates $\tau'^{(k)}$ and the primary code period T_c to the right side of the equation, an estimate of the number of full code periods can be obtained as:

$$n_c^{(k)} = \left(\frac{\left\|\mathbf{r}_c^{(k)} - \mathbf{r}_{c,rec}\right\|}{c} - \tau'^{(k)} + \delta t_{rec} + \frac{M^{(k)} + s^{(k)}}{c}\right)/T_c \tag{3.17}$$

Since the magnitude of $\frac{s^{(k)}}{c}$ is within few nanoseconds, orders of magnitudes lower than T_c, in principle, the integer $N_c^{(k)}$ could be obtained by rounding the $n_c^{(k)}$ estimate to the nearest integer. However, in practice, this rounding can not be performed due to the existence of the receiver clock bias δt_{rec}. In order to overcome the estimation difficulties brought by this unknown bias, a reference satellite can be chosen and a differencing step can be performed to cancel out this common value. The result of this step represents the difference of number

of full periods between two different satellites, which, as it results from integer subtraction, should still be an integer. By assigning the reference satellite with index 0, the integer difference can be computed by:

$$N_c^{(k0)} = N_c^{(k)} - N_c^{(0)} = \text{round}\left(n_c^{(k)} - n_c^{(0)}\right) \tag{3.18}$$

After applying Equation (3.18) to all satellites, all the SD integer numbers of full code periods are obtained. In order to construct the full pseudoranges, a random integer can be taken for $N_c^{(0)}$. As mentioned in [67], the expected flight time of a GPS satellite ranges from 64 ms to 89 ms, according to the geometric distance. With the chosen $N_c^{(0)}$ value, the full pseudoranges can be computed based on Equation (3.12) by:

$$P^{(k)} = c \cdot (\tau'^{(k)} + (N_c^{(k0)} + N_c^{(0)}) \cdot T_c) \tag{3.19}$$

Since $N_c^{(0)}$ is a randomly chosen integer, a common integer bias exists among all the resulting pseudoranges. This, however, does not impact the accuracy of PVT computation since these common integer bias parts are assimilated into the common receiver clock offset, which is estimated in the navigation filter. It is the integer differences $N_c^{(k0)}$ that contain the information necessary for snapshot positioning.

The success of the above computation of full pseudoranges relies on the set of information listed below that can not be obtained by the snapshot GNSS receiver alone. This means that the assistance module is required to provide these data by gathering them from external sources. The necessary information includes:

- Satellite ephemeris data. In order to compute the satellite positions at any epoch, the ephemeris parameters must be available.

- Satellite clock bias parameters. They are usually included in the ephemeris data, but some organizations (e.g., the International GNSS Service (IGS)) provide high accuracy

clock parameters through some separate products as well.

- Parameters for Ionosphere and Troposphere models. These data are required to compute the atmospheric effects compensations $M^{(k)}$ as needed in Equation (3.17).

- Coarse estimate of receiver position $\mathbf{r}_{c,rec}$.

- Coarse estimate of the transmission time. This absolute timing information is required to compute the coarse positions of the satellites $\mathbf{r}_c^{(k)}$ following Equation (3.17).

The first three items are continuously transmitted to the users through the broadcast navigation messages by each GNSS constellation of satellites. However, due to the limitations in the signal duration, snapshot receivers rely on other channels to obtain these data, such as internet. The quality of these parameters should be guaranteed by the organizations providing them. However, the coarse estimates of receiver position and reception time can not always be accurate enough since it depends on the method to obtain these information. Some of the typical methods to compute the coarse receiver position are:

- assume the coarse receiver position as the coordinates of a nearby cell towers transmitting the assistance data. The communication data packages usually contain the coordinates of the nearby cell towers, these coordinates are typically within 10 km of the communication device.

- use the last estimated position stored in the receiver memory. In most use cases the coordinates remain within kilometers to the last receiver position, unless the receiver has been switched off for a long time.

- store the coordinates of a fixed location in the receiver, for instance the center of a city. Many applications, such as asset tracking, operates only in a certain area. Thus the object been tracked is usually located within tens of kilometers to the center of the area in nominal operational conditions.

As for the coarse timing information, similar strategies can be applied. Typically the time tag provided by the communication module receiving assistance data contains error that is lower than few seconds. In order to ensure that the SD code period integer results computed based on Equation (3.18) are rounded to the correct integers, the following condition should be satisfied:

$$|e_c^{(k)} - e_c^{(0)}| \leq \frac{1}{2} T_c \cdot c \tag{3.20}$$

Where $e_c^{(k)}$ and $e_c^{(0)}$ stands for the total range errors introduced by the inaccuracies of the coarse position and time information for satellite k and the reference satellite respectively. Note that the other residual error terms are at nanosecond level and are thus neglected. The $e_c^{(k)}$ error can be expressed as:

$$e_c^{(k)} = \left\| \mathbf{r}_c^{(k)} - \mathbf{r}_{c,rec} \right\| - \left\| \mathbf{r}^{(k)} - \mathbf{r}_{rec} \right\| \tag{3.21}$$

where $\mathbf{r}^{(k)}$ represents the actual position of satellite k at the transmission time and $\mathbf{r}_{rec}$ represents the actual receiver location. As mentioned in [67], the maximum range rate of a GPS satellite is ± 800 m/s, the satellites from other constellations present an even smaller range rate due to their different orbit heights and inclination angles, except for the GLONASS satellites whose maximum range is ± 900 m/s. This implies that even in the worst scenarios, the range errors introduced by coarse time inaccuracy of one minute can reach 48 km for one GPS satellite.

The reference satellite for the computation in Equation (3.18) should be the satellite with the highest elevation angle [67]. It is also noted that when the two satellites in consideration are in the same part of the sky, $e_c^{(k)}$ and $e_c^{(0)}$ are affected in a similar way. This largely eases the fulfillment of the condition listed on the right side of Equation (3.20). A technique of switching reference satellite is proposed in [67] in order to alleviate the limits of the coarse position and time errors, such changes of reference satellites ensures that it is always the two closest satellites being considered. A rule of thumb of error margins for GPS satellites

is also given as 100 km of coarse position error and 1 min for the coarse time error, since the GPS primary code period is just 1 ms, which is equivalent to approximately 150 km of total error margin.

Other techniques were developed in order to further increase the boundary of the tolerable coarse position and time errors, including an instantaneous positioning algorithm based on Doppler measurements only [70]. This technique guarantees a positioning result with error up to a few kilometers, which is completely acceptable for rounding to the correct integer, as long as the coarse time error is not enormous (i.e., less than 3 minutes). Based on this technique, the coarse time Doppler navigation algorithm was developed to further expand the tolerance on initial time error to 1.5 h [61].

3.4 Coarse Time Filter

The full pseudorange measurements generated with Equation (3.19) contain a common integer bias for all the satellites due to the randomly selected $N_c^{(0)}$ integer value. Thus, only the differences between these full pseudoranges (essentially flight times) are accurately known. Although the common integer bias can be assimilated to the common receiver clock bias and does not impact the PVT computation workflow, the navigation filter lacks an accurate knowledge about the absolute times. The main problem brought by this absence of information is that the computed satellite positions could suffer from large errors since they relay on accurately known transmission times. Since the satellites are moving at different velocity and towards different directions, the amount of satellite position errors resulted from wrong transmission times are different from each other. Thus, these errors can not be absorbed into the common clock bias term in the navigation filter.

The inaccurate estimations of satellite transmission times can be observed in Equation (3.10) as the integer part $X_c^{(k)}$ can be expressed by Equation (3.22) based on previous analysis

of Equation (3.13) and (3.18).

$$X_c^{(k)} = Y_c - 1 - (N_c^{(k0)} + N_c^{(0)}) \tag{3.22}$$

Assuming the $N_c^{(k0)}$ integer is correctly determined thanks to a successful full period ambiguity resolution process described in section 3.3, the right side of Equation (3.22) still contains two uncertain parameters:

- Y_c, a value that is chosen as the integer part of the receiver time. This value is typically originated from the coarse time information provided by the assistance module, and is the only source of absolute time at the stage before a navigation solution is generated. The error of Y_c can range up to few seconds.
- $N_c^{(0)}$, a random integer flight time chosen to construct the full pseudoranges. Since the actual flight times of GNSS satellites are always between 64 ms to 89 ms, any value chosen between this range should lead to only a few milliseconds of error.

The range errors caused by these time uncertainties are illustrated by the blue arrows as shown in Figure 3.8, where two different scenarios are shown: satellite 1 is rising while satellite 2 is setting. $r^{(1)}$ and $r^{(2)}$ represent the positions of the two satellites computed at their actual transmission times, and their actual ranges to the receiver antenna are shown in green arrows. In contrast, $r_c^{(1)}$ and $r_c^{(2)}$ represent the satellite positions when the computations are done at a wrong transmission time that is caused by the coarse time error, the corresponding ranges are shown by red arrows. As it can be seen, the actual range is larger than the false range for satellite 1 and shorter for satellite 2.

The following equation can be used to compute the range errors introduced by the coarse time error:

$$\delta\rho^{(k)} = \rho^{(k)}(t_{tr}^{(k)} + \delta t_c) - \rho^{(k)}(t_{tr}^{(k)}) \tag{3.23}$$

where:

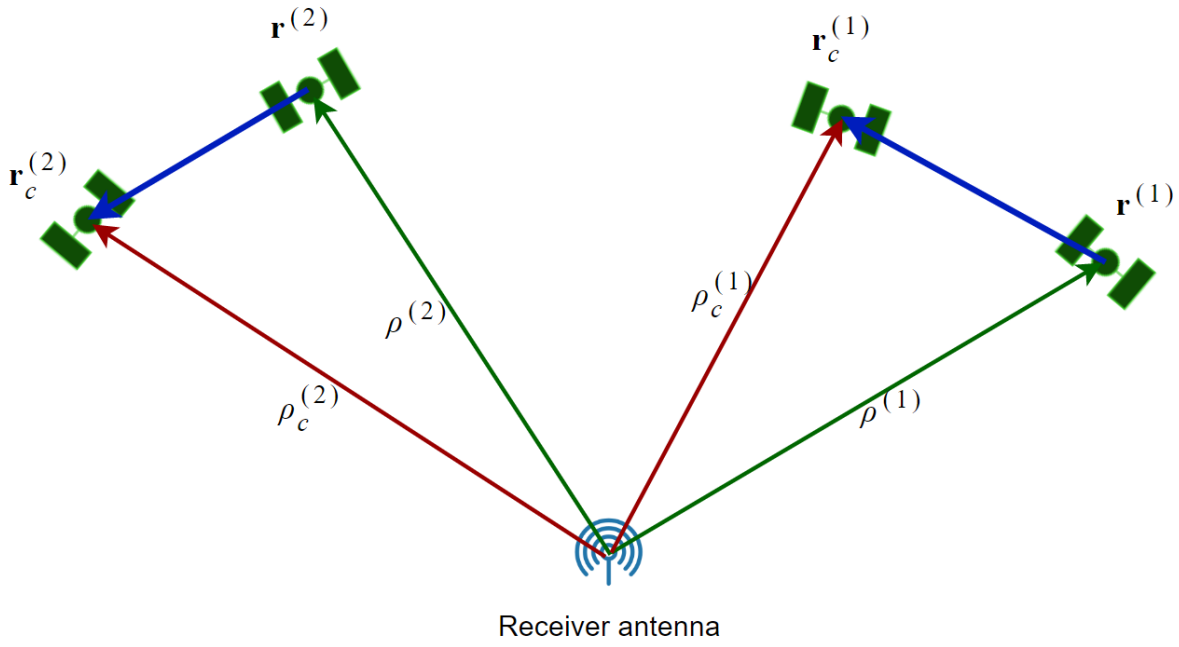

Figure 3.8: Different range errors caused by a common coarse time error. The blue arrows represent the satellite position errors introduced by the coarse time error.

- $\delta\rho^{(k)}$ is the range error introduced by δt_c, in m
- $\rho^{(k)}(t)$ is the range from the receiver to satellite k at epoch t, in m
- δt_c stands for the coarse time error attributable to the assistance module, in s

In the example shown in Figure 3.8, $\delta\rho^{(1)}$ is negative and $\delta\rho^{(2)}$ is positive. In order to avoid these range errors to deteriorate the computation of PVT solution, the range rate of each satellite can be used to compensate these errors, that is:

$$\delta\rho^{(k)} = v^{(k)} \cdot \delta t_c \tag{3.24}$$

where $v^{(k)}$ representing the pseudorange rate of satellite k, in m/s. More specifically, such range rates can be computed by:

$$v^{(k)} = \left(\mathbf{e}^{(k)} \cdot \mathbf{V}^{(k)} - \dot{\delta}_t^{(k)}\right) \tag{3.25}$$

where:

- $\mathbf{V}^{(k)}$ is the relative velocity between the receiver and satellite.
- $\mathbf{e}^{(k)}$ is the unitary vector to project $\mathbf{V}^{(k)}$ to the direction of LOS.
- $\dot{\delta}_t^{(k)}$ represents the satellite clock error rate in unit of length/time

By including these compensations to the full pseudoranges generated for snapshot signals, the navigation equation presented in Equation (2.16) needs to be updated and can now be expressed as:

$$b^k = \frac{x_0 - x^k}{\rho_0^k} \cdot dx + \frac{y_0 - y^k}{\rho_0^k} \cdot dy + \frac{z_0 - z^k}{\rho_0^k} \cdot dz + c \cdot \delta t_{rec} + v^{(k)} \cdot \delta t_c \tag{3.26}$$

where:

- b^k is the prefit residual that is computed following Equation (2.17) by using the coarse time t_c and coarse receiver position $\mathbf{r}_{c,rec}$
- $(x_0,\, y_0,\, z_0)$ represent the initial guess of the receiver coordinates, for CTF, the coarse receiver position $\mathbf{r}_{c,rec}$ is used.
- $(x^j,\, y^j,\, z^j)$ are the coordinates of satellite k computed at the epoch $t_{tr}^{(k)} + \delta t_c$

The resulting position estimates $d\mathbf{r}_{rec}$ are thus the three-dimensional coordinates vector from the coarse position to the actual receiver position.

By applying Equation (3.26) to all the satellites, the following matrix form of navigation equations can be established:

$$\begin{bmatrix} b^1 \\ \vdots \\ b^n \end{bmatrix} = \begin{pmatrix} \frac{x_0 - x^1}{\rho_0^1} & \frac{y_0 - y^1}{\rho_0^1} & \frac{z_0 - z^1}{\rho_0^1} & c & v^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \\ \frac{x_0 - x^n}{\rho_0^n} & \frac{y_0 - y^n}{\rho_0^n} & \frac{z_0 - z^n}{\rho_0^n} & c & v^{(n)} \end{pmatrix} \begin{bmatrix} d\mathbf{r}_{rec} \\ \delta t_{rec} \\ \delta t_c \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{3.27}$$

As it can be seen, the vector of unknowns contains five elements following the inclusion of the coarse time error δt_c. Similar to the representation shown in Equation (2.19), the above navigation equation can be simplified as:

$$\mathbf{b} = \mathbf{H} \begin{bmatrix} d\mathbf{r}_{rec} \\ \delta t_{rec} \\ \delta t_c \end{bmatrix} + \varepsilon \tag{3.28}$$

Where $\mathbf{H}$ is the geometry matrix for CTF with size n $\times$ 5, containing the satellite to receiver unitary LOS vectors, the speed of light constant and the observed pseudorange rate v for each satellite.

By solving the navigation filter shown in Equation (3.28) using, for instance, the WLS method, the resulting coarse time navigation solution can be estimated as:

$$\begin{bmatrix} \hat{d\mathbf{r}}_{rec} \\ \hat{\delta} t_{rec} \\ \hat{\delta} t_c \end{bmatrix} = \left(\mathbf{H}^T \mathbf{W} \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{W} \mathbf{b} \tag{3.29}$$

where $\mathbf{W}$ represents the weighting matrix for all the measurements used in CTF. Note that the hats in this equation means that they are the optimal estimates. The final reception time of the CTF solution t_{ctf} can then be computed by:

$$t_{ctf} = t_c - \hat{\delta} t_c - \hat{\delta} t_{rec} \tag{3.30}$$

while the position solution can be obtained by:

$$\mathbf{r}_{rec} = \mathbf{r}_{c,rec} + \hat{d\mathbf{r}}_{rec} \tag{3.31}$$

The main difference of the coarse time navigation filter and the traditional SPP filter

is the inclusion of the fifth element corresponding to the coarse time error δt_c. As it can be seen in Equation (3.24), in order to have an accurate estimate on this unknown, some special attention should be put on the range rate $v_{(k)}$. If the $v_{(k)}$ values used in the H are not precise enough, the resulting PVT solution is deteriorated.

Equation 3.25 shows that the magnitudes of the range rates are determined mainly by the following quantities:

- Satellite velocity. Different satellites travel at different velocities at their signal transmission epochs. These velocities could introduce a range rate of up to 900 m/s [67, 4] when projected to the LOS.

- Receiver velocity. The receiver velocity is identical for all satellites and thus the coarse time error will result in an position errors that is identical as well. However, since the LOS vectors $\mathbf{e}^{(k)}$ are different among satellites, the range errors projected to each satellite are different, unless the receiver is static with zero velocity.

- Satellite clock drift. The satellite onboard oscillators are atomic clocks that are designed to run steadily for a long time without any large satellite error variations. The nominal satellite clock error rate is at the order of 2^{-35} s/s, which is equivalent to a range rate of 1 cm/s, and can thus be neglected considering the magnitude of the range rates introduced by satellite velocities.

- Receiver clock drift. The receiver clocks have lower quality compared to the satellite clocks and may present some clock error drift. The coarse time error could result in a common receiver clock bias that can be absorbed to the fourth element of the unknown vector δt_{rec}. Thus, this parameter does not impact the positioning performance.

Note that the Doppler frequency offset F_D mentioned in Section 3.2 is formed with the above mentioned quantities. As a result, the range rate can be computed based on such Doppler

measurements, after an appropriate change of scale factor from cycle/s to meter/s:

$$v^{(k)} = F_D^{(k)} \cdot \lambda \tag{3.32}$$

In some researches, the range rates are obtained by directly projecting satellite velocities computed based on the ephemeris parameters to their LOS vectors [71]. Such procedure assumes that the receiver velocity is zero as in static scenarios [67, 68]. In the present dissertation, the range rates are obtained based on Equation (3.32). Thus, since the precision of Doppler measurements are determined by the settings of MH acquisition module, the final coarse time navigation solution accuracy also depends on these settings.

3.5 Typical Performance

The previous sections have presented the conventional snapshot positioning techniques. In order to evaluate the performance of these previously developed methods, a software dedicated to snapshot data processing has been developed during the PhD in cooperation with Albora Technologies. Based on some real-world data sets collected and processed with this software, this section provides an overview of the typical positioning and timing results of the conventional snapshot positioning techniques.

3.5.1 Timing Results

The raw GNSS signals were captured by the Labsat 3 record and playback device [72] that is connected to a high-end multi-frequency GNSS antenna (Septentrio PolaNt-x) [73]. The output from the Labsat recorder are the I & Q digital bits that are down converted to baseband. Note that the original data file contains a continuous signal that lasts for 90 s, but in order to processing these data in snapshot mode, these data are truncated to shorter duration pieces. More specifically, 450 snapshot files are cut from this data set with each

snapshot separated by 200 ms. In addition, some more details about these snapshot data are listed:

- Sampling frequency: the original signals were captured by Labsat with a sampling rate of 58 MHz, but for the following data processing and performance evaluation, the data sets are further down sampled to 14.5 MHz.

- Snapshot length: Only the first 40 ms of each truncated snapshots are used, the signals are processed in fully coherent integration mode, i.e., $T_{coh} = 40\ ms$

- Signal selection: Only GPS L1 C/A signals are processed for this experiment.

- Total number of satellites used in the navigation filter: 12

- The elevation angles of the used satellites range between 7.7 and 76.9 degrees, while their C/N0 values range between 36.2 and 43.43 dB Hz.

In order to evaluate the accuracy of the coarse time estimate, they need to be compared with a reliable reference data. The following two steps were performed in order to obtain such a high quality timing reference:

- The reference data are initially obtained based on the timing solutions computed with accurately time tagged snapshot measurements. These measurements are free from the transmission time estimation inaccuracies presented in Equation (3.22). These more advanced measurements are generated following the algorithms described in chapter 4. The accuracy of timing solutions estimated with these measurements are generally at the nanosecond level, considering that the pseudorange accuracy is at meter level. Thus, these accurately time tagged snapshot measurements can be used as a reference for this experiment.

- A further validation of the precision of these reference data is done by taking advantages of the Labsat clock stability during the short signal collection time. The Temperature

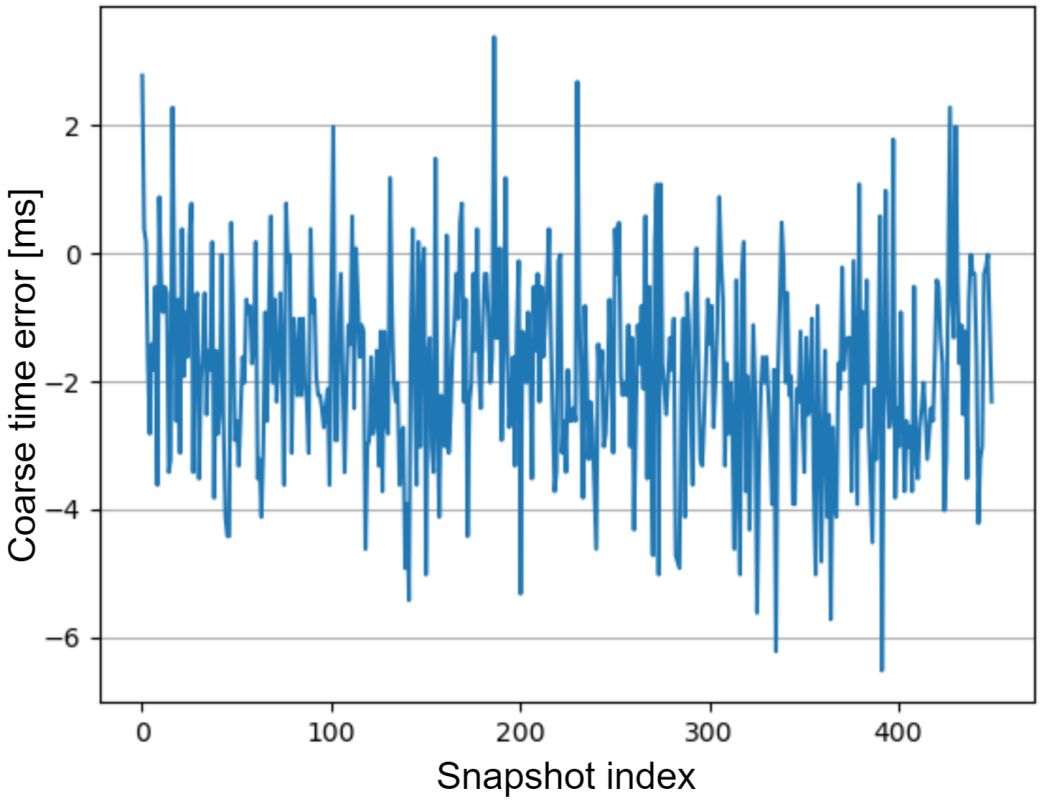

Figure 3.9: Coarse time errors of the 450 snapshot taken from the 90 s Labsat data collected on 2020-05-19

Compensated Crystal Oscillator (TCXO) onboard the Labsat recorder guarantees the clock drift to be within ± 0.1 ppm, which is equivalent to a maximum of 0.009 ms of clock error for the signals collected during 90 s. Such an error is at least two orders lower than the typical time errors generated by a CTF. Since the above mentioned reference data is highly consistent with the TCXO clock time (with less than 0.001 ms of discrepancies), the effectiveness of this data set is validated.

It can be seen from Figure 3.9 that in general the timing errors are at the level of few milliseconds, with the maximum error reaching -6.4 ms. Considering that only GPS satellites are used in this experiment, better timing solutions can be expected when more satellites from other GNSS constellations are to be incorporated into the CTF.

Figure 3.10: Antenna location in the north campus of UPC for snapshot data collection

3.5.2 Positioning Results

As for the positioning results, real-world data were collected using a snapshot receiver developed by Albora Technologies. The receiver was connected to the same antenna as in section 3.5.1. It was located in an open environment at the north campus of Universitat Politècnica de Catalunya (UPC) in Barcelona, Spain, as shown in Figure 3.10. The snapshots were taken roughly every 5 minutes and the collected digital bits were transmitted to the cloud server for PVT computation. The exact time gap between two snapshots slight varies due to the delays of data upload and cloud processing. Here are some information about the collected data:

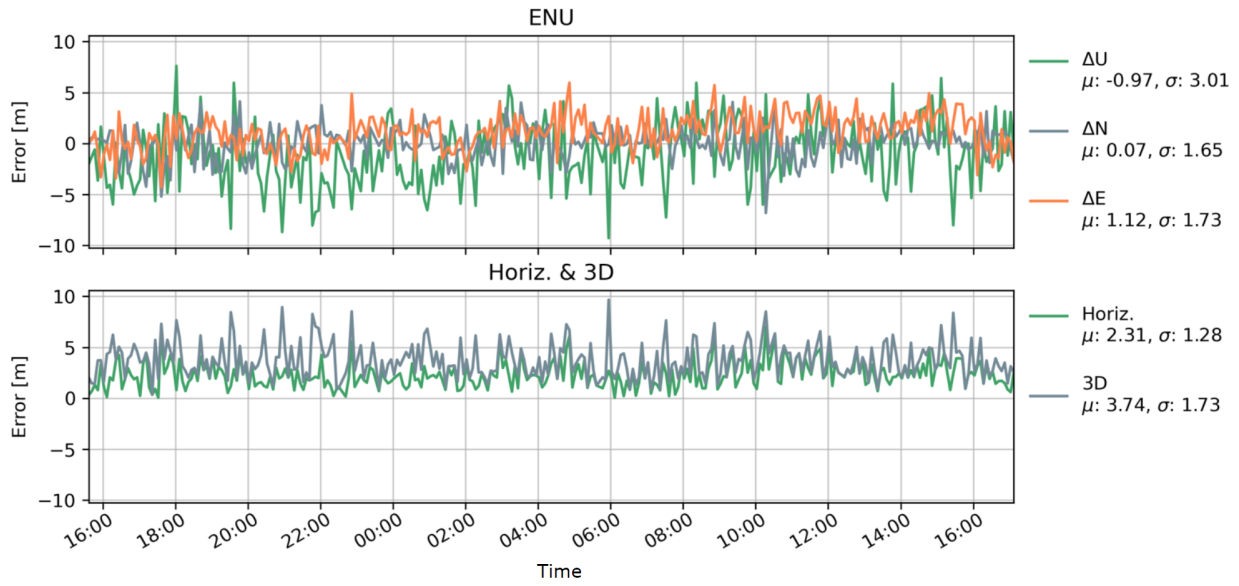

Figure 3.11: Conventional snapshot positioning errors of 25 hours of data collected on 2021-02-01

- With a total of 300 snapshots taken, the time span of these collected data exceeds 25 hours.
- The collected I & Q digital bits are sampled at 6.36 MHz.
- The data were processed with coherent integration time of 40 ms.
- Only GPS L1 C/A and Galileo E1C signals are used in this experiment.

The positioning errors in different directions are depicted in Figure 3.11. The mean errors and its standard deviation were also computed, as represented by μ and σ. It can be seen that in general the positioning errors are at the meter level. The upper panel represents the errors in east, north and up directions (respectively represented by ΔE, ΔN and ΔU). As expected, the main error component is the vertical direction with a standard deviation of 3 m. The lower panel shows that the horizontal errors is generally less than 5 m while the maximum 3D error is below 10 m. It can be concluded that for this experiment, the accuracy of conventional snapshot positioning is at the same level as traditional close loop receivers processing under the SPP mode, both at meter level.

Chapter 4

High Accuracy Snapshot Positioning

The conventional snapshot positioning techniques introduced in Chapter 3 produce positioning results with meter level accuracy only. This level of accuracy suffices some applications, such as for asset tracking and fleet management. Nowadays, the benefits brought by snapshot positioning techniques, especially in terms of power efficiency and economic cost, have attracted more applications to adopt this method of positioning. Along with classical applications that require meter level accuracy, others demand high accuracy positioning results. For example, millimeter level accuracy is required for geodetic displacement monitoring system for infrastructures such as bridges, dams, buildings, etc. Traditionally, a network of high-end GNSS receivers and antennas need to be deployed, together with additional infrastructure to power the receivers. It is interesting to evaluate the feasibility of obtaining high accuracy positioning results based on snapshot receivers, so that these low-cost devices can replace the expensive networks of traditional geodetic receivers for such applications.

Thanks to the advantages in power consumption, snapshot receivers can reduce the complexity of building new power infrastructures as they can rely on batteries or solar panels. Besides, snapshot techniques are usually performed in cooperation with cloud computing techniques in order to explore its full potential of power saving features. That is, the digital bits generated by the snapshot receivers are uploaded to the cloud and all the following signal processing and PVT computation tasks are performed on the cloud platform.

As for the choice of positioning algorithms, PPP is a high-accuracy GNSS positioning

technique, as introduced in Chapter 2, that requires a long convergence time before reaching highly-accurate positioning results. The convergence time is at the order of few tens of minutes, which is against the nature of snapshot recordings since they are designed to be as short as possible so that the TTFF is short and small data size is transmitted to the cloud. Thus, the present research focus on integrating snapshot data processing with RTK-based positioning engine. The technique resulting from this combination is termed as the Snapshot RTK (SRTK).

SRTK is the first technique aiming at achieving high-accuracy positioning based on snapshot data. In order to succeed obtaining a SRTK fix, two necessary conditions must be satisfied, respectively:

1. Snapshot measurements must be accurately time tagged so that satellite transmission times can be computed with enough accuracy.

2. DBA issues must be resolved so that there is only an unique set of measurements that can be generated for each satellite. This step also ensures that Half Cycle Error (HCE) does not appear in the carrier phase measurements.

In the present chapter, the focus is put on tackling the first condition and on verifying the feasibility of SRTK. The SRTK workflow and the strategy proposed for generation of time-tagged snapshot measurements are introduced respectively in section 4.1 and 4.2. Some feasibility experiments were performed and the results are shown in section 4.3 to confirm the first success of SRTK fix. Finally, by adjusting some related design parameters for more real world snapshot recordings, the SRTK performance is analysed in section 4.4. The methods related to the second condition are discussed in Chapter 5 with more details.

4.1 SRTK Workflow

Figure 3.1 depicted the overall architecture of the SRTK processing engine. Now a cloud-based SRTK engine is being applied for the experiments performed in this dissertation,

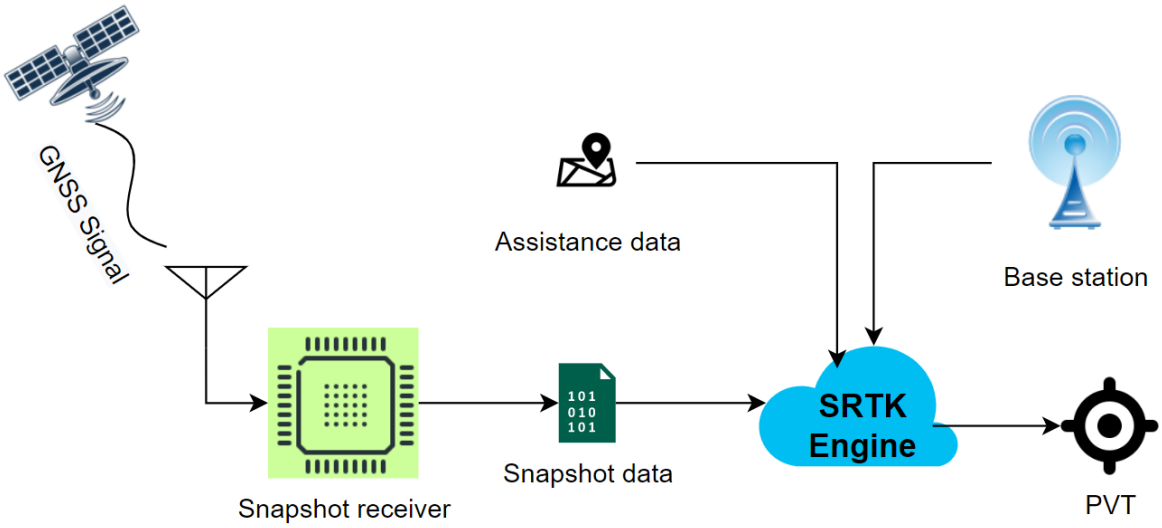

Figure 4.1: General architecture of cloud-based SRTK positioning.

its overall workflow is firstly illustrated in Figure 4.1. Similar to the conventional snapshot positioning workflow, the snapshot digital bits are generated by the snapshot receiver. These data are transmitted to a cloud platform which constantly collects all necessary assistance data required for snapshot positioning, including coarse time and position data, ephemeris parameters, etc. The main difference with respect to conventional snapshot positioning is that, for RTK processing, measurements collected from base stations are also required in order to compute the DD observables. All these data are gathered and processed by the SRTK engine on the cloud platform to obtain high accuracy positioning solutions.

The more specific steps inside the SRTK engine are depicted in Figure 4.2. Starting with the input of snapshot digital bits, the MH acquisition module generates a set of snapshot measurements for each satellite in view. From there, the SRTK engine workflow can be divided into two paths. In the uppermost path, the code delay and Doppler measurements are firstly used in a conventional CTF with the assistance of the coarse time and position information and ephemeris data, as described in section 3.4. This step, shaded in pink color, aims at obtaining an absolute reception time information about the collected snapshot data. The satellite transmission times can also be roughly estimated based on such information.

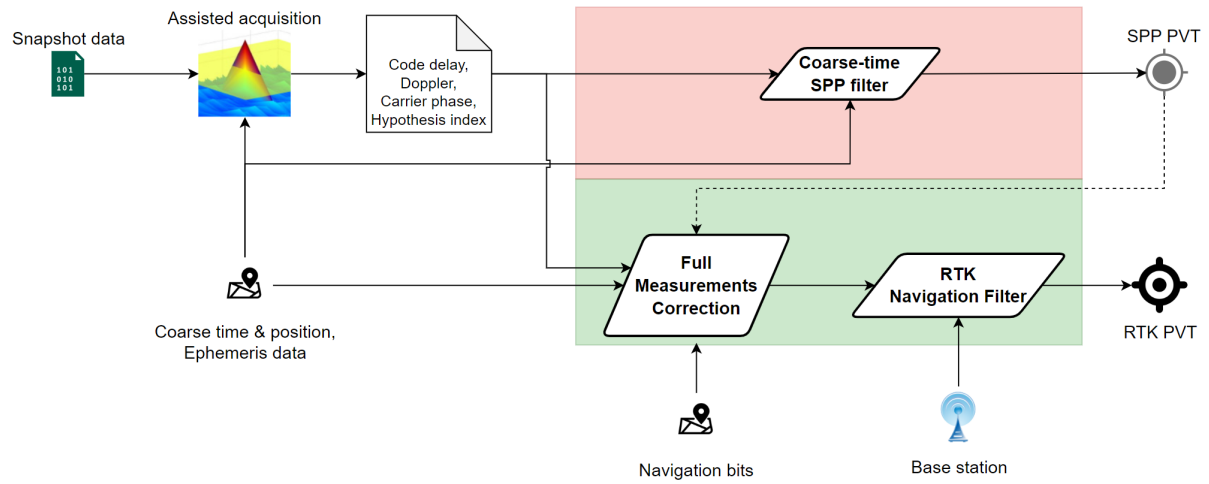

Figure 4.2: SRTK engine data processing workflow

As shown by the dashed line, the PVT solution of the CTF is then used in a full measurement correction process in the bottom panel. This process, also assisted by the coarse information, is designed to construct full pseudorange measurements that are accurately time tagged based on the set of fractional values obtained by the acquisition module. The resulting complete measurements are then fed to a RTK navigation filter together with the measurements collected from nearby base stations. Finally, this navigation filter generates a new PVT solution that could be highly accurate. Note that the full measurement correction process can also correct carrier phase HCEs if related navigation data bits are available on the cloud platform, this will be further discussed in Chapter 5.

It can be seen that the main difference between the SRTK engine and the conventional snapshot positioning engine is the inclusion of the blocks shaded in green color. While the RTK navigation filter is a common module that is specifically discussed in section 2.2.5, the main innovative step is the full measurement correction module. This unique process designed for SRTK is discussed in the following section with more details as it is the core of the whole workflow.

4.2 Full Measurement Correction

4.2.1 Motivation

The main motivation of adopting the CTF instead of a traditional SPP navigation filter for snapshot data resides in the lack of accurate knowledge about the absolute transmission time of the signal. In the case such time can not be accurately determined, the satellite coordinates computed at these erroneous times present large errors. The fifth element in the CTF geometry matrix is designed to compensate the range errors $\delta\rho^{(k)}$ caused by such wrong satellite positions.

Such compensations based on range rate (or Doppler offset) have been proved to work properly for conventional snapshot receivers when only code measurements are used and meter-level accuracy is desired. For high-accuracy applications, however, this method fails due to the stringent requirement about the satellite position accuracy. As it was shown in section 3.5.1, the timing accuracy of conventional snapshot CTF is at millisecond level even though the coarse time error δt_c is estimated in the filter. Thus, a residual range error may still exist in each full pseudorange generated based on Equation (3.19). These range errors are too large for a positioning engine aiming at fixing carrier phase ambiguities. For example, for GPS satellites, 1 ms of absolute timing error implies that the full pseudorange may be prone to errors of up to 0.8 m, and, that the DD range error can reach 1.6 m in the worst scenarios. The magnitude of such range errors may be acceptable for meter-level positioning accuracy. However, in contrast, it is much larger than 20 cm of the L1 signal wavelength and therefore will totally corrupt the possibility of carrier phase IAR and deny the RTK solution to be fixed. As a result, a different approach of measurement generation is required for SRTK in order to correct these range errors caused by timing inaccuracies. These corrected measurements are expected to reflect the actual satellite transmission times with nanosecond level accuracy.

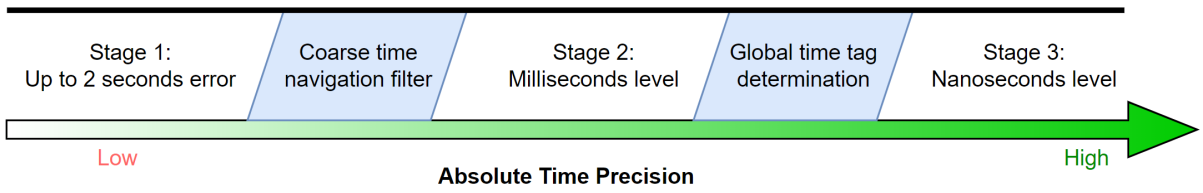

Figure 4.3: 3 stages of absolute time precision evolution in the SRTK positioning engine

An additional motivation of correcting the measurements to indicate accurate satellite transmission times is that when external sources of navigation data bits are available, smaller time errors will result in less possibility of retrieving the wrong bit from these distributed data set.

4.2.2 Global Time Tag Determination

The full pseudoranges of the acquired satellites are computed, according to Equation (3.12), based on the differences of the receiver time t_{rx} and the transmission times for each satellite. The receiver time t_{rx}, as defined by Equation (3.11), is common to all satellites. Since the pseudoranges can be generated with any time tag, usually they are chosen at epochs with full integers of primary code periods for simplicity of pseudorange computation. That is, $\tau_r = 0$ and the time tag is located at a primary code edge. Such receiver time is defined as a global time tag in this dissertation. The key, however, is to ensure the accuracy of satellite transmission times.

Before explaining the details of the method to obtain high accuracy transmission times and corresponding full pseudoranges, it is important to overview the accuracy of the receiver time that is available in the system. As illustrated in Figure 4.3, the absolute time accuracy in the SRTK filter gradually improves throughout three stages. In the first stage, the time information comes from the assistance data, typically from a cellular network, and it may include errors of up to 2 s [67]. Then, the second stage of time accuracy improves to the millisecond level after the CTF generates a time solution by estimating the coarse time error.

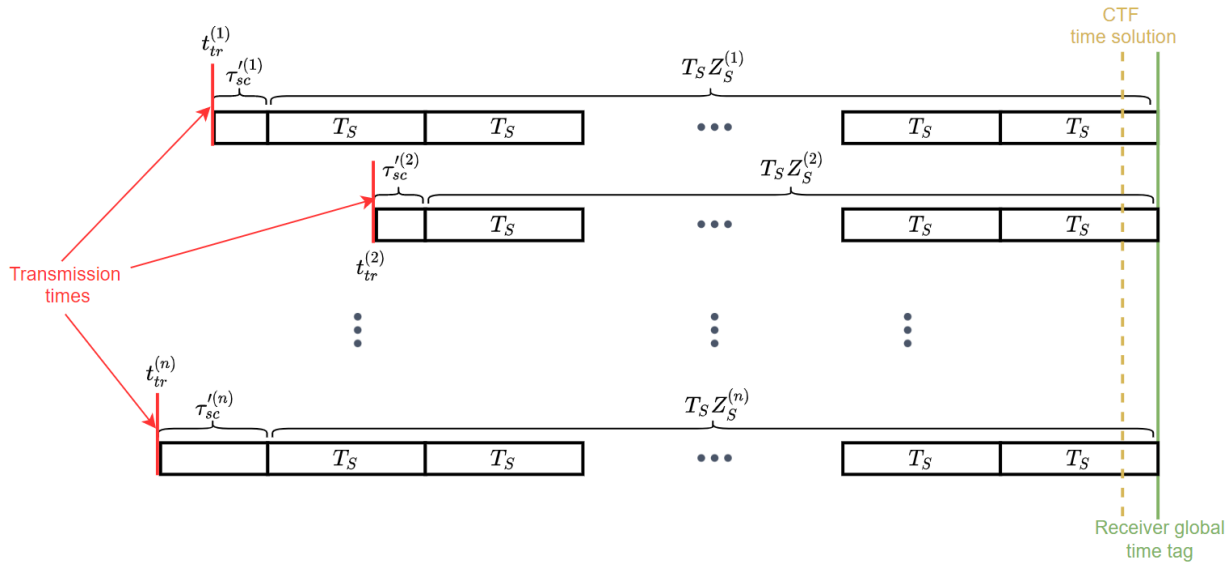

Figure 4.4: Relationship between the global time tag and other time parameters.

Finally, the third stage aims at an accuracy equivalent to that of the code delay estimates, i.e., nanosecond level. The present section focuses on describing the process from stage two to stage three.

According to Equation (3.10), the satellite transmission times can reach nanosecond level only when the integer number of codes X_c can be correctly determined. However, as shown in Equation (3.22), these integers suffer from the uncertainties of integer Y_c brought by the selection of receiver time tag, as well as the reference flight time integer $N_c^{(0)}$. The problem can be now redefined as to find a pair of these two values that returns the actual magnitude of $Y_c - N_c^{(0)}$. Note that this pair of values is not unique, if the receiver time is chosen to be later, the correct transmission time integer X_c can still be determined if the pseudorange measurements of all satellites are increased by the same amount.

The method to solve this problem is, in short, to determine a proper global time tag and then correspondingly adjust the pseudorange measurements to it. Figure 4.4 presents the relationships of different time parameters that are involved in this method. This figure is similar to Figure 3.7 that is used to illustrate the process of full period ambiguity resolution for fractional code delay measurements. As in both Figures, the satellite transmission epochs

$t_{tr}^{(k)}$ are represented by the red vertical lines on the left side. However, in Figure 4.4, each black box represents the duration of one data bit or a secondary code period, denoted by T_S, instead of a primary code period T_c. This shows that the determination of global time tag is performed at symbol level, instead of at code level, in order to ensure the correct rounding process.

Analogous to the code delay quantity $\tau'^{(k)}$, which was defined as a fraction of a primary code period shown in Figure 3.7, the secondary code delay, denoted by $\tau_{sc}'^{(k)}$, is now defined as the time delay between the transmission time and the latest edge of data bit or secondary code period. This value can be obtained with high accuracy based on the MH acquisition results based on Equation (4.1).

$$\tau_{sc}'^{(k)} = \tau'^{(k)} + u^{(k)} \cdot T_c \tag{4.1}$$

where $\tau'^{(k)}$ and $u^{(k)}$ are the code delay and hypothesis index that can be obtained from the set of snapshot measurements for satellite k. Note that the index u also represents the SCI value in the cases of pilot signals.

The detailed process of generating accurately time-tagged measurements is:

1. Compute the signal reception time t_{ctf} using the conventional CTF. This time is denoted by the yellow dash line in Figure 4.4. Note that t_{ctf} is the only source of absolute time available in the receiver at this step and its accuracy is at millimeter level.

2. Estimate the actual flight times based on the geometric ranges computed using assistance data and the corrections of other modeling terms $M^{(k)}$ such as satellite clock errors and atmospheric propagation delays. For this step, Equation (4.2) can be applied.

$$t_{f\_rough}^{(k)} = \frac{\left\| \mathbf{r}_c^{(k)} - \mathbf{r}_{c,rec} \right\| + M^{(k)}}{c} \tag{4.2}$$

As already mentioned in the full period ambiguity resolution process described in

section 3.3, the $t^{(k)}_{f\_rough}$ estimate errors are guaranteed to be less than 0.5 ms.

3. Compute the rough transmission times for each satellite based on the coarse receiver time t_{ctf}. Equation (4.3) can be applied

$$t^{(k)}_{tr\_rough} = t_{ctf} - t^{(k)}_{f\_rough} \tag{4.3}$$

These rough transmission times are accurate to millisecond level.

4. Add the secondary code delays $\tau^{\prime(k)}_{sc}$ to the rough transmission times. In principle, if the milliseconds of errors did not exist, this summation should result in a global time tag t_g located at the edge of a data bit or a secondary code period. Fortunately, errors at the level of the milliseconds do not impede the successful retrieval of these edges. The computation consists of a rounding process represented by Equation (4.4)

$$t_g = \text{round}\left(\frac{t^{(k)}_{tr\_rough} + \tau^{\prime(k)}_{sc}}{T_S}\right) \cdot T_S. \tag{4.4}$$

This step can be successfully implemented thanks to the fact that T_S is typically one order of magnitude larger than the aforementioned timing inaccuracies. For example, $T_S = 100ms$ for Galileo E1C signals as it correspond to the duration of a secondary code period, thus, a few milliseconds of error in $t^{(k)}_{tr\_rough}$ is tolerable for the rounding process.

5. Find the actual integer number of secondary code periods or data bit duration, which is required for the construction of full pseudoranges paired with the chosen global time tag.

$$Z^{(k)}_S = \text{round}\left(\frac{t_g - t^{(k)}_{tr\_rough} - \tau^{\prime(k)}_{sc}}{T_S}\right). \tag{4.5}$$

6. Compute the final pseudorange measurements P_{final} for each satellite that correspond

to the time tag t_g based on Equation (4.6).

$$P_{final}^{(k)} = c \cdot (\tau_{sc}^{\prime(k)} + Z_S^{(k)} \cdot T_S). \tag{4.6}$$

It is important to mention that the time tag t_g detected at step 4 and the integer values $Z_S^{(k)}$ computed at step 5 correspond to one pair of successful solutions that guarantee high accuracy transmission times to be obtained. However, such pair is not the only possible combination. For example, if the global time tag is chosen at $t_g' = t_g + T_S$ instead of the t_g obtained by Equation (4.4), then based on Equation (4.5), the resulting pseudoranges will contain integer values of $Z_S^{\prime(k)} = Z_S^{(k)} + 1$ that compensate this common bias without affecting the transmission time accuracy. As a result, the pair of t_g' and $Z_S^{\prime(k)}$ is valid as well for high-accuracy applications. In fact, Equation (4.4) is applied mainly because the actual receiver time is usually within one cycle of T_S after the satellite transmission time. For example, if the transmission time TOW of a Galileo E1C signal is at 259169.2193, then the optimal t_g candidate would be 259169.3 since the flight times are between 64 ms to 89 ms.

4.3 Zero Baseline SRTK Feasibility

The time-tagged pseudoranges constructed with the method presented in the previous section can be used in combination with the snapshot carrier phase measurements by the RTK navigation filter to achieve high accuracy snapshot positioning. The present section introduces some experiments performed under the "zero baseline" configuration, which indicates that most of the measurement error terms listed in Equation (2.9) and (2.27) are identical for the rover and base receivers since they are connected to the same antenna. This configuration eases the RTK data processing work flow and prevents the DD measurements from being impacted by the residual errors such as atmosphere and antenna phase center variation.

4.3.1 Experiment Goals

As mentioned in [74], it is actually the pseudorange measurement noise level, instead of carrier phase measurement quality, that plays the dominate role in impacting the possibility of achieving RTK fixes. The noise of full pseudoranges is essentially originated from the snapshot code delay measurement $\tau^{(k)}$. Such noise level is determined by the signal bandwidth and integration time [64, 75, 76], which can be evaluated by its Cramér-Rao Bounds (CRB). CRB is proportional to $\propto \frac{1}{(C/N_0)BT_{\text{int}}}$, where T_{int}, B and C/N_0 variables denote the coherent integration time, signal bandwidth and carrier to noise ratio, respectively. It can be seen that code delay error decreases when the integration time T_{int} or bandwidth B increases. While C/N_0 parameter depends on the surrounding environment, the other two parameters can be controlled by adjusting the snapshot receiver settings.

The experiments in this section were performed with three goals.

1. The primary goal was to perform carrier phase IAR for the first time using snapshot signals so as to confirm the feasibility of obtaining a SRTK fix.

2. The collected data were processed also to explore the relationship of code noise level and IAR fix rates of snapshot signal under different bandwidth and integration times.

3. Different settings of data processing were used in order to analyse the influences of multi-constellation and multi-frequency data on IAR fix rate.

4.3.2 Experiment Processing Options

The snapshot data set used in the present subsection is the same as the one mentioned in section 3.5.1, collected by a Labsat recorder. For RTK processing, the measurements generated with these snapshots are termed as rover measurements. As for the base station, a high-end GNSS receiver (Septentrio PolaRx-5e [77]) is used, it provides measurements at a 1 Hz rate. Note that under "zero baseline" configuration, the Labsat recorder and the base

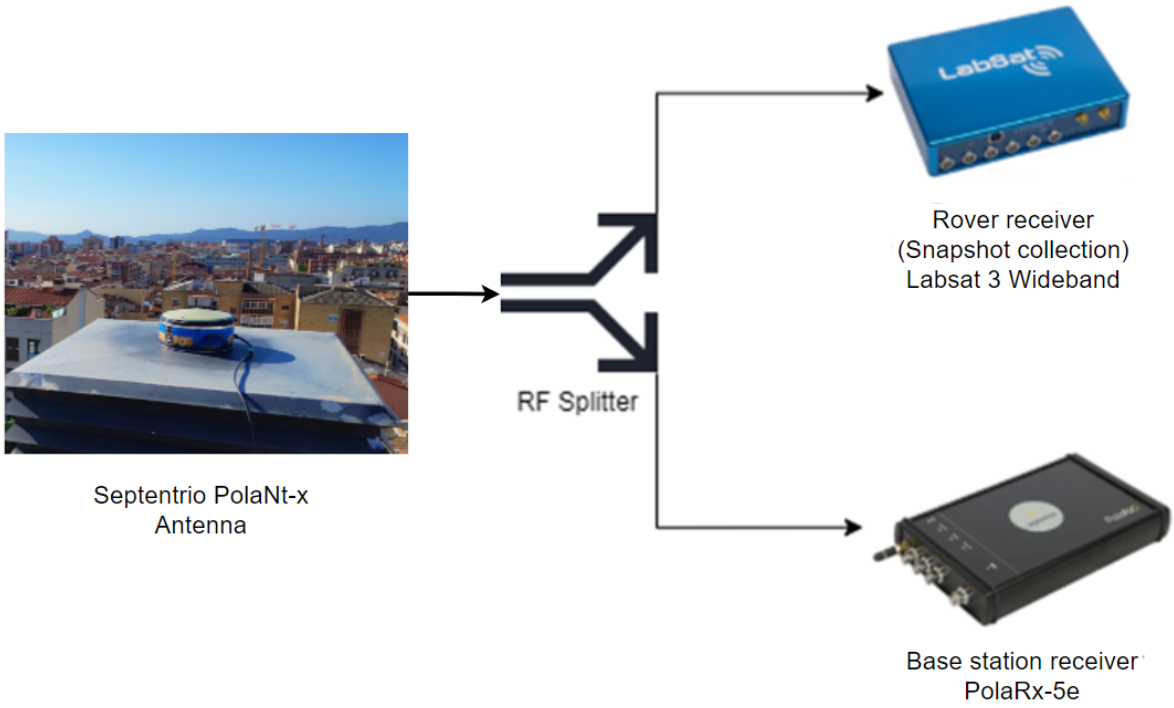

Figure 4.5: Zero baseline configuration with antenna located near Sabadell for snapshot data collection.

station receiver are connected to the same antenna (Septentrio PolaNt-x MF [73]) through a RF splitter that ensures that the same GNSS signal is received by them. The antenna is statically located on a roof with open environment near Sabadell, Spain, as it can be seen in Figure 4.5.

As for data processing, the snapshot signal and data processing software used in section 3.5.1 was expanded to include the new implementations of time tag and full measurement generation. This upgraded software was able to export these snapshot measurements in RINEX format that is very commonly used in the GNSS industry, the base station receiver was able to produce measurements in this format as well. As for the RTK navigation filter, the open source tool RTKLIB [78] was applied. All the baseband signal processing and SRTK filter data processing as shown in Figure 4.2 were performed in post-processing mode. Note that as mentioned in the beginning of this chapter and will also be discussed more thoroughly in Section 5.1, some snapshot carrier phase measurements may be prone to

potential HCEs, these situations are avoided by making use of navigation message data bits archived in the data base of German Research Centre for Geosciences (GFZ) [65] and by applying the methods mentioned in Chapter 5. As a result, all the snapshot carrier phases are valid (absent from HCEs) for RTK processing. The fact that they are fractional values of a cycle does not represent a problem since the integer parts of the DD carrier phases are estimated as unknown parameters in the RTK filter.

Since each snapshot data only produces one set of measurements, the RTK navigation filter was set to run in instantaneous mode, which means the estimation and IAR were performed using only measurements from one single epoch. In contrast, traditional receivers usually generate carrier phase measurements containing integer ambiguities that remain as constants across multiple epochs until the so-called cycle slip occurs and introduces a discontinuity. This fact allows some RTK filters to process measurements across several epochs in order to better estimate the unchanged integer ambiguities as constant values. The fact that only fractional parts of carrier phase measurements are available in the snapshot means that their integer parts need to be estimated at each epoch, no matter if the snapshots are taken continuously.

In order to achieve SRTK fix, multiple sets of full measurements were generated by processing the collected snapshot data with different coherent integration times and different bandwidth settings. Namely, integration times of 20 ms, 40 ms, 60 ms, 80 ms and 100 ms were used while signal bandwidths of 11.2 MHz, 14 MHz, 15.4 MHz, 17.5 MHz, 19.6 MHz, and 24.5 MHz were applied. The changes of bandwidth were realised by down-sampling the original 58 MHz snapshot digital bits to lower sampling frequencies since such decreases of sampling rate effectively reduces the bandwidth as well. Two quantities are used to indicate the SRTK performance: the LAMBDA Ratio Factor (LRF) and the fix rate. The fix rate is equivalent to the percentage of snapshots with LRF greater than a predefined threshold value. For the experiments described in this section, this threshold value is set to 3.

In order to assess the noise level in the code measurements, the DD code measurement

values were saved before being fed to the RTK navigation filter. As shown in Equation (2.37), these values should be equivalent to the DD code noise when the baseline $d\mathbf{r}_{rb}$ is equal to zero. Thus, from these stored data, the Rooted Mean Square Error (RMSE) of DD code measurement caused by noise can be concluded for all satellites and all epochs following Equation (4.7).

$$RMSE(t) = \sqrt{\sum_{k=2}^{n} (P_{rb}^{1k}(t))^2/(n-1)} \tag{4.7}$$

where n is the total number of satellites and $P_{rb}^{1k}(t)$ represents the DD pseudorange computed at epoch t between satellite k and the reference satellite, denoted as $k = 1$. Thus, the counter starts at $k = 2$ and a total of $n - 1$ DD pseudoranges are available. Note that by default the satellites were sorted based on their elevation angle and the one with largest elevation is used as the reference satellite.

In order to address the influences of multi-constellation and multi-frequency data on the IAR fix rate, snapshot measurements were fed to the navigation filter but this time with different constellations and frequency selection. This experiment only considered GPS and Galileo constellations and signals from the L1 and L5 frequency bands.

4.3.3 Experiment Results

First of all, the SRTK performance with GPS L1 C/A signals was verified. Figure 4.6 depicts results for the data set using 11.2 MHz bandwidth and different integration times. The corresponding carrier-to-noise density (C/N_0) values of these satellite measurements are plotted in Figure 4.7. The variation of calculated LRF values are represented in the upper panel of Figure 4.6. As it can be seen, for all the integration time settings used, there are always some snapshots that result in LRF values higher than the predefined threshold, which means the carrier phase IAR were successful at these epochs. This implies that RTK-level positioning accuracy results can be obtained for snapshot signals. The bottom panel of Figure 4.6 shows the RMS of DD pseudoranges. The blue curve, which represents the signal

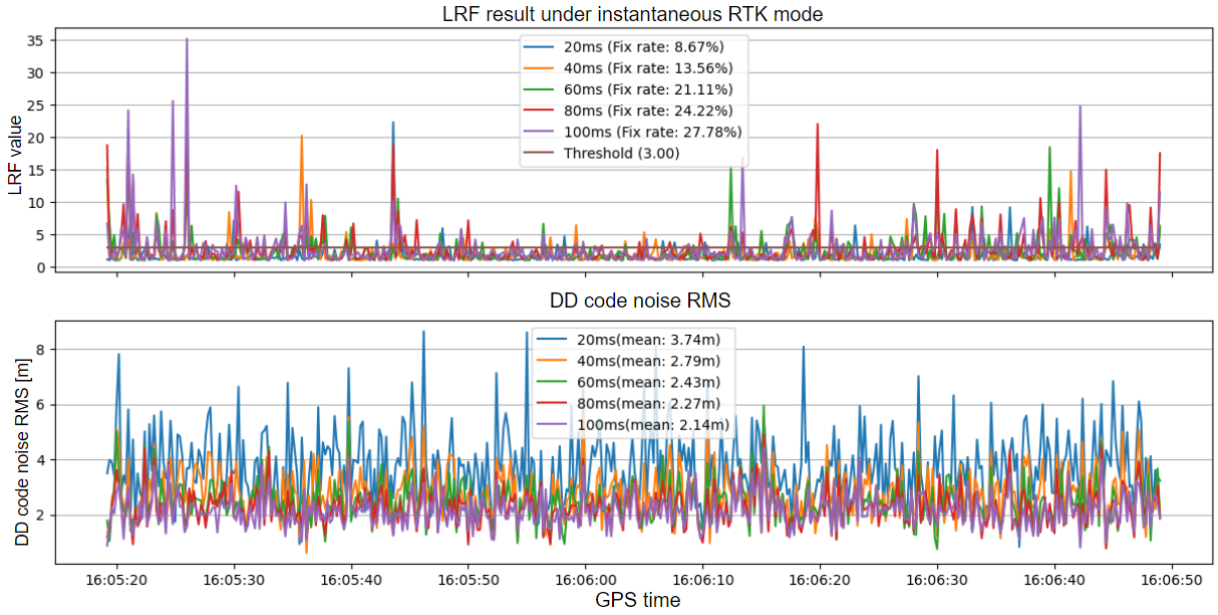

Figure 4.6: Time plot of LRF and DD code noise RMS for 11.2 MHz bandwidth under zero baseline.

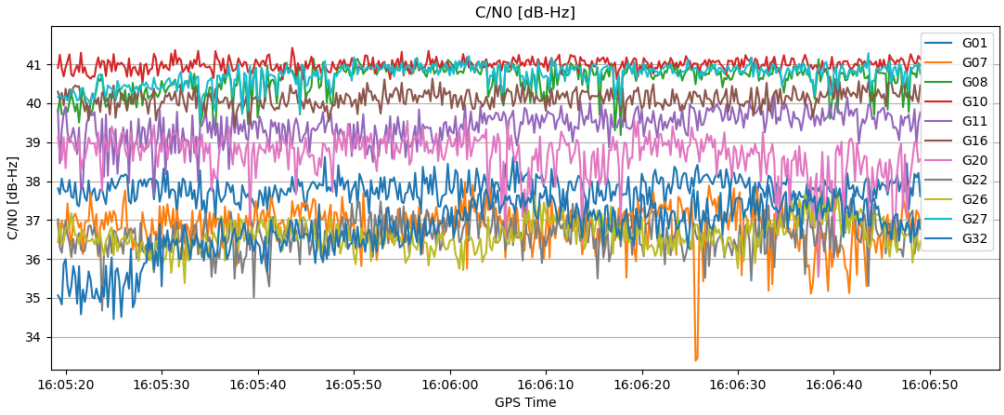

Figure 4.7: C/N_0 of GPS satellites in the collected snapshot data.

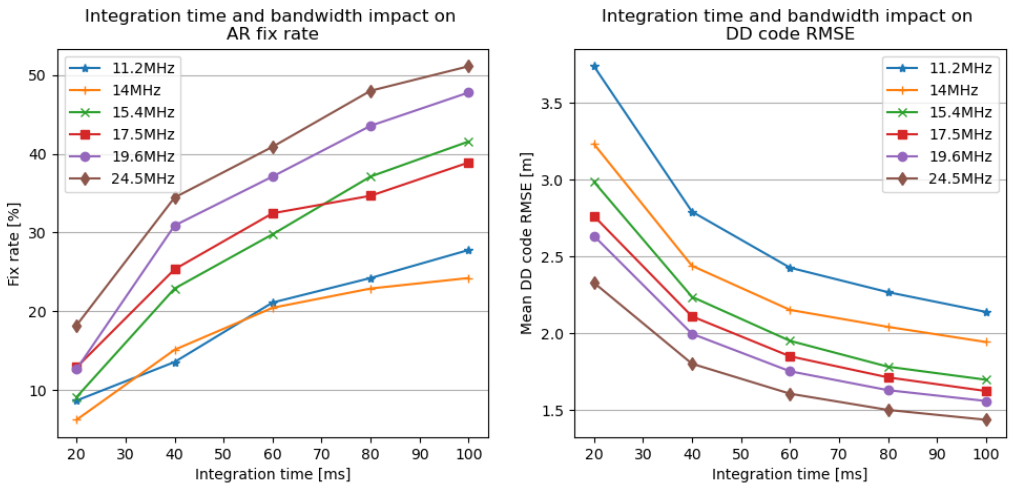

Figure 4.8: Integration time and bandwidth impact on fix rate and DD code noise RMS under zero baseline

with integration time of 20 ms, shows a larger DD code noise RMS compared to the purple curve that represents an integration time of 100 ms. The SRTK fix rate and the mean RMS values for the whole set of 450 snapshots are also computed and shown in the legends of the two subplots respectively. As expected, the longer the integration time, the higher fix rate can be achieved and the lower the DD pseudorange noise. This pattern can be observed more clearly in Figure 4.8 where impacts from different bandwidths are shown as well.

Figure 4.8 confirms the increase of the fix rate and the reduction of code measurement noise when the signal is acquired with a larger bandwidth. The lowest DD code noise RMS value obtained in these date sets is less than 1.5 m, which corresponds to the fix rate of 51%. Although about half of the snapshots can manage to obtain a RTK fix, such fix rate is rather low, this could be explained by the fact that only GPS L1 C/A signals are used to compute the results presented here.

It is recalled that for the previous analysis, GPS L1 C/A signals measurements were generated with 5 different integration time and 6 different bandwidth settings, which means a total of 30 sets of measurements are available, with 450 epochs included in each of them. In order to explore the code noise impact on IAR fix rate, DD code RMSEs of all satellites are calculated for each of these 13500 snapshots and plotted on the left side of Figure 4.9.

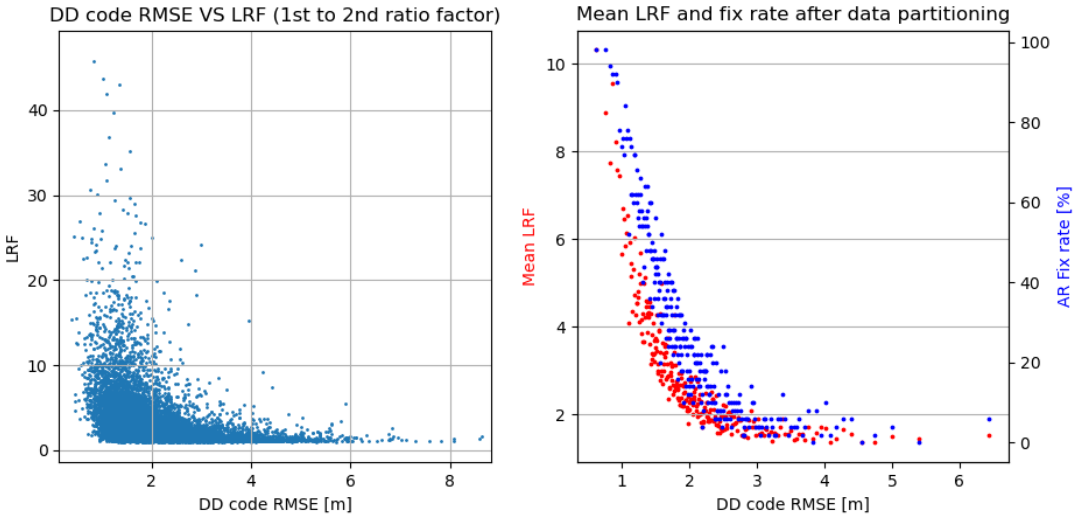

Figure 4.9: Relation ship between DD code RMSE, IAR fix rate and LRF

A basic trend can be observed in Figure 4.9: the smaller DD code RMSE becomes, the LRF tends to be higher and thus a higher fix rate can be expected. In order to show more clearly the relationship between the DD code quality and the fix rate, a simple partitioning is performed on these data points. They were sorted by their DD code RMSE magnitudes in an ascending order and divided into 270 groups with the sorted index, with each group containing 50 snapshots and the maximum DD code RMSE of the current group is smaller than the minimum of the next group. Then, the mean LRF and fix rate of each group is calculated and plotted on the right side of Figure 4.9. The trend matches the raw data points on the left, which shows a decrease in fix rate and mean LRF magnitude when the code RMSE goes higher. From this plot, it can also be seen that in order to achieve a 90% fix rate, the DD code measurement RMSE should be smaller than 1 m, which correspond to a mean LRF between 6 and 8.

Finally, in order to investigate the improvements brought by using Multi-Constellation Multi-Frequency (MCMF) signals from GPS and Galileo, different combinations of signals and constellations were applied to the navigation filter and the results are shown in Table 4.1. Snapshot signals with 6 different bandwidths were processed while the identical integration

time of 100 ms was set for all cases for generating these experiment results.

Table 4.1: IAR fix rate with different combinations of constellation and frequency bands, with 100 ms integration time under zero baseline

Bandwidth [MHz]	GPS L1CA	GAL E1C	GAL E5A	GPS L1CA& GAL E1C
11.2	27.78%	36.67%	23.56%	100%
14.0	24.22%	43.11%	69.78%	100%
15.4	41.56%	45.33%	90.89%	100%
17.5	39.11%	50.00%	99.56%	100%
19.6	48.00%	50.22%	100%	100%
24.5	51.33%	57.33%	100%	100%
Bandwidth [MHz]	GPS L1CA& GPS L5	GAL E1C& GAL E5A	MCMF	
11.2	97.33%	100%	100%	
14.0	98.67%	100%	100%	
15.4	99.78%	100%	100%	
17.5	99.78%	100%	100%	
19.6	100%	100%	100%	
24.5	100%	100%	100%	

When only using GPS L1 C/A signals, as analysed before, the maximum fix rate that can be reached among all these data sets is 51.33%, it is slightly higher when only Galileo E1C signal is used, although the maximum fix rate is still only 57.33%, which is achieved at the bandwidth of 24.5 MHz. However, Galileo E5A signal brings an improvement in terms of fix rate, except for the cases when a bandwidth of 11.2 MHz is used, in other cases of bandwidths of 19.6 MHz and 24.5 MHz, the fix rates are 100%. Note that the nominal bandwidth of Galileo E5A signal is 20 MHz, it can be expected that their performance is worse when a lower bandwidth is used. The case with GPS L5 signal only is not listed here because there were only 5 GPS satellites observed in this data set transmitting signals in L5 frequency band, and no Integer Ambiguity Resolution (IAR) fix has been achieved for this case.

However, when both GPS and Galileo constellations are used simultaneously, the result shows that IAR is not a problem anymore, as a 100% fix rate is obtained for all cases. As for single-constellation and multi-frequency processing, the output fix rate is also almost 100%

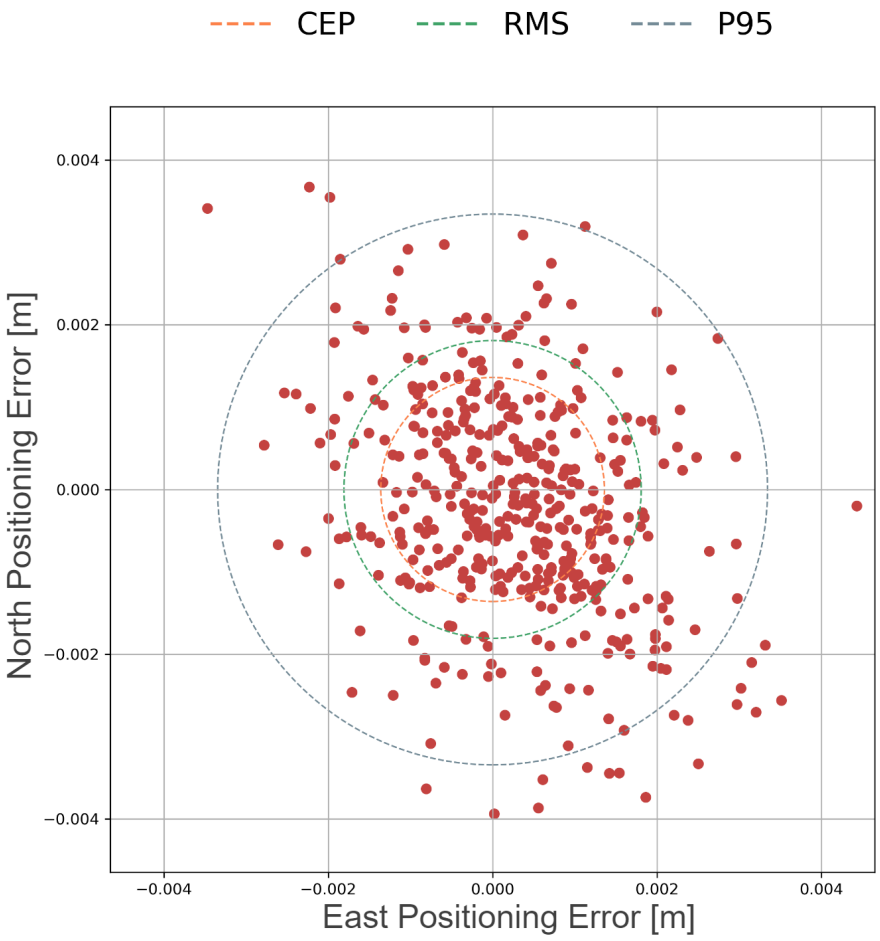

Figure 4.10: Horizontal errors of SRTK fixed positioning solution under zero baseline, using GPS L1 C/A and GPS L5 signal at 24.5 MHz bandwidth with 100 ms integration time

for all scenarios, except for the cases when GPS L1CA and L5 are used with bandwidths lower than 17.5 MHz, nevertheless, these values are still higher than 97%. The column for Multi-Constellation Multi-Frequency (MCMF) represents that case when all four signals are used, and 100% fix rate can be achieved as well in this case.

The SRTK fixed positioning results in zero baseline configuration are shown in Figure 4.10, using GPS L1 C/A and GPS L5 signal at 24.5 MHz bandwidth with 100 ms integration time. The positioning errors in north and east are all less than 5 mm. The Circular Error Probable (CEP) circle, shown in orange color, contains half of the positioning points inside, and it has a radius of 1.4 mm. The RMSE in the horizontal dimension is presented as the green circle and has a magnitude of 1.8 mm. The P95 circle, which represents the error

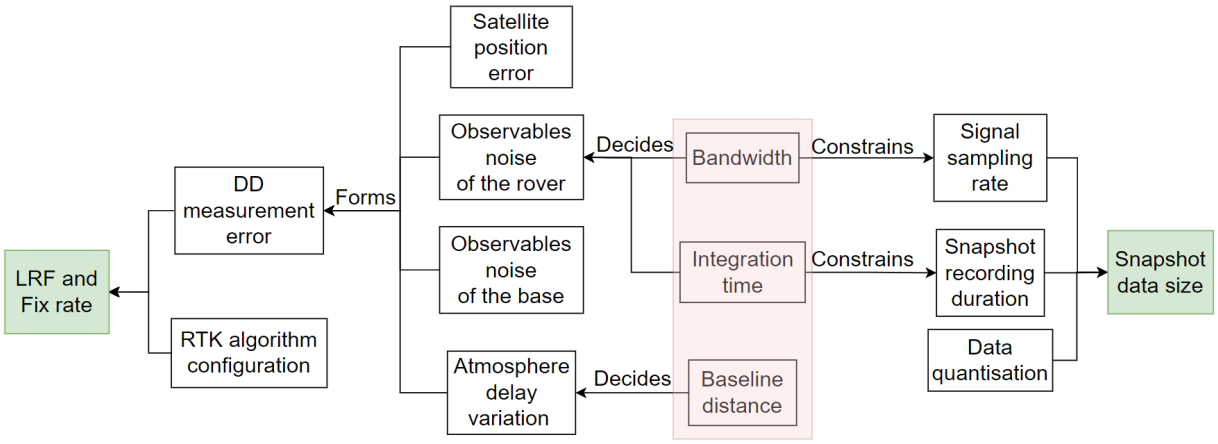

Figure 4.11: Relationships between the parameters of interest for SRTK data processing

magnitude that 95% of the data points reside in, has a radius of 3.4 mm. Note that these results correspond to a zero baseline experiment with a high-end reference receiver. Overall it shows that the SRTK positioning accuracy can reach millimeter level under such conditions.

4.4 SRTK Performance

The experiments performed in section 4.3 only considered zero baseline configuration and mainly intended to prove the feasibility of the SRTK technique. This section extends the previous experiments to different baseline distances and provides a deeper insight into the performance of SRTK in terms of the IAR fix rate. Besides, this performance is analysed from a more practical point of view as the total data size of the captured snapshot data is also considered.

4.4.1 Experiment Goals

The main objective of the experiments performed in this section is to explore the impact of the following three parameters on the SRTK performance: snapshot signal bandwidth, integration time, and baseline distance, which are shaded in red color in Figure 4.11. This figure shows the relationships between all the parameters of interest. Starting from these

three parameters, it can be explored from the left and right directions to find their impacts on the SRTK fix rate and the snapshot data size respectively.

Looking leftwards, it can be seen that the observables noise of the rover is determined by the signal bandwidth and the integration time, while the baseline distance impacts the magnitude of the atmosphere delay variation between the rover and the base receiver. The rover observable noise, together with other error terms such as reference receiver measurement noise and atmosphere delay residual errors, contribute to the final DD measurement errors. Note that for snapshot receivers, additional satellite position errors caused by timing inaccuracies may exist if the measurements are not generated with the method mentioned in section 4.2.

Looking rightwards, the bandwidth puts a constraint on the signal sampling rate in order to avoid the signal aliasing problems, that is, the sampling rate must be higher than the Nyquist rate, which is usually twice the bandwidth [79]. Furthermore, the integration time sets a minimum value of the total duration of the captured signal. These two parameters, together with the quantization factor, directly determine the minimum snapshot data size as it can be computed by:

$$S = 2 \cdot Q \times Fs \times T + A, \tag{4.8}$$

where:

- S is the size of the snapshot data, in bits
- Q represents the constant quantization parameter, in bits per sample
- Fs stands for the sampling rate, in Hz
- T is the total integration time, in s
- A stands for the size of additional data, in bits

The additional data size term A is usually very small (less than 1 kB) since only very limited amount of metadata is needed along side the snapshot signal data bits, such as rover

receiver identifier number. The multiplication by the constant 2 is due to the different I and Q streams of the snapshot signal.

From a commercial point of view, one of the major costs of adopting SRTK service arises from the transmission of raw GNSS data from the user end to the remote cloud platform. The mobile network data plan chosen for this service will depend on the minimum size of the data packet needed for each snapshot. The smaller the snapshot, the cheaper this service is. It is therefore of great interest to explore the minimum data size of a snapshot required for obtaining a given IAR fix rate.

4.4.2 Experiment Processing Options

A total of 240 snapshots were recorded for this section using the snapshot receiver designed by Albora Technologies, the same one as used for the results shown in section 3.5.2. The total duration of signal collection is 4 hours as each snapshot is separated by 1 min from the adjacent one. The receiver is also connected to the same high-end antenna shown in Figure 3.10. The snapshots were sent to the cloud platform to be processed by the software containing the SRTK positioning engine as presented in Figure 4.2. The navigation message data bits archived in the data base of GFZ are used in the MH acquisition module to ensure that HCEs do not appear in snapshot carrier phase measurements.

The base station data were collected using the VRS service provided by Institut Cartogràfic i Geològic de Catalunya (ICGC). Several streams of VRS data were collected simultaneously by setting a series of reference points with different baseline distances to the snapshot receiver. The original data contains signals from L1 band, however, only GPS L1 C/A and Galileo E1C signals were used to generate the following results. Although BDS B1C signals are also in the L1 frequency band, the corresponding measurements were not provided by the VRS service for reference data collection, thus BDS constellation was not processed. GLONASS constellation was discarded as well due to their inferior signal strength and requirements on the RF front-end. An elevation mask of 10 degrees was applied to fil-

ter out the low elevation satellites. The RTK navigation filter was running in instantaneous mode, which means the computations were performed based on observable data from a single epoch. The solution is regarded as fixed when the computed LRF value is greater than the threshold that is set to 2. The RTK filter considers only the short baseline DD measurement models represented by Equation (2.30) and (2.31) by neglecting the residual DD Ionosphere and Troposphere slant delays.

The experiment results were generated by processing the snapshot data set with all combinations of the following settings:

- Integration time (ms): $\{40, 60, 80, 100\}$
- Signal bandwidth (MHz): $\{6.36, 7.95, 10.6, 12.72, 15.9, 21.2, 25.44, 31.8\}$
- Baseline distance (km): $\{0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$

4.4.3 Experiment Results

First of all, Figure 4.12 presents the number of satellites and the DOP changes during the 4 hours of data collection. As it can be seen, more than 15 satellites are used in the navigation filter for most of the snapshots.

To assess the fix rate and mean LRF results, two plots can be generated for each integration time used, with the horizontal axis representing the varying settings of baseline distance and the vertical axis representing the SRTK fix rate and mean LRF value respectively. As an example, Figure 4.13 and 4.14 are generated for all the bandwidths used when the integration time is set to 100 ms. As it can be seen, the two plots follow a similar pattern, as it can be clearly observed that generally both the mean LRF value and SRTK fix rate drop when the baseline distance increases. This makes sense since the LRF values directly determine the possibility of obtaining a SRTK fix. For other integration times, similar trends can be obtained as well.

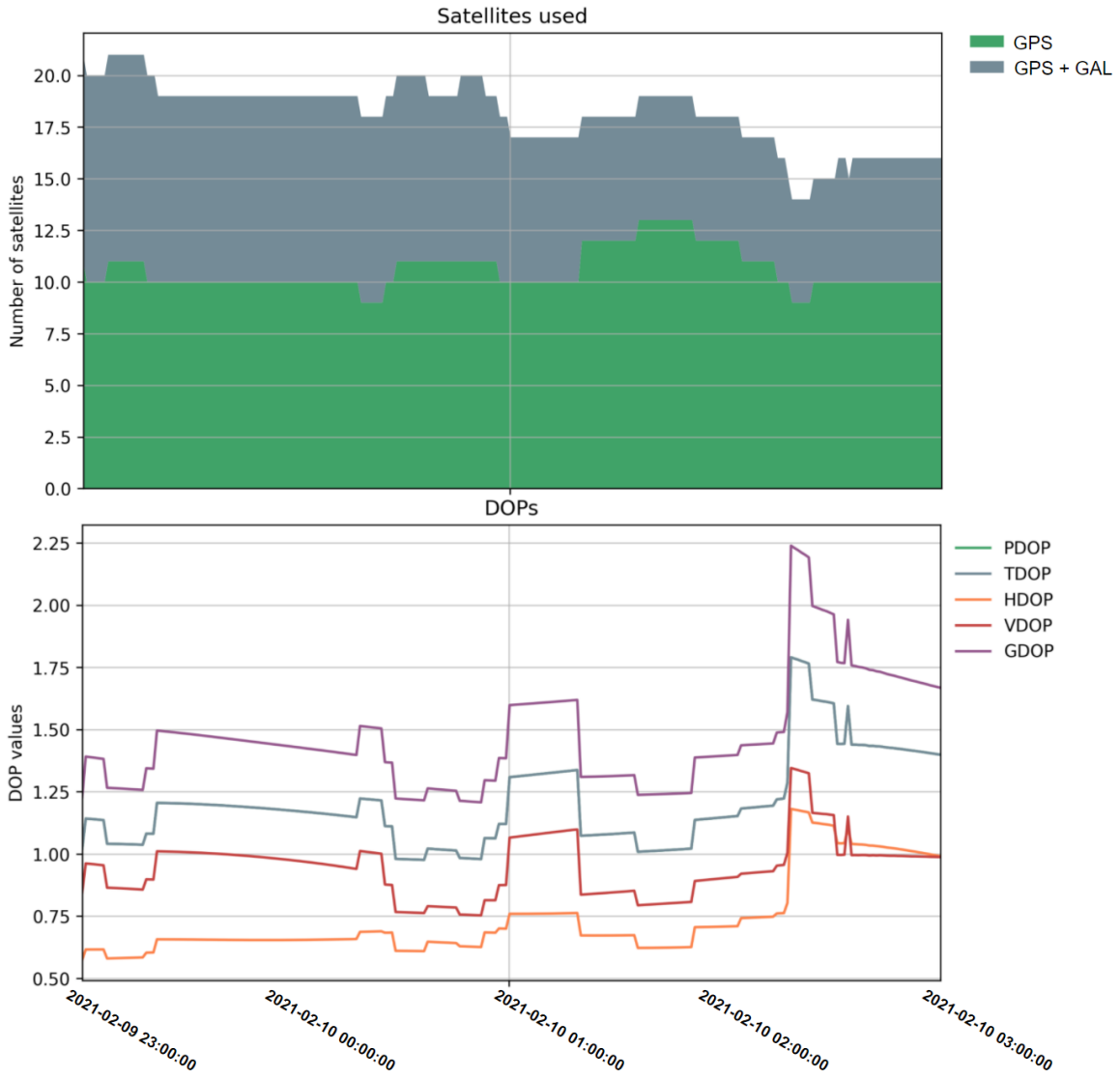

Figure 4.12: Number of satellites used (top) and the DOP values (bottom) of the 4 hours snapshot data collected on 2021-02-10

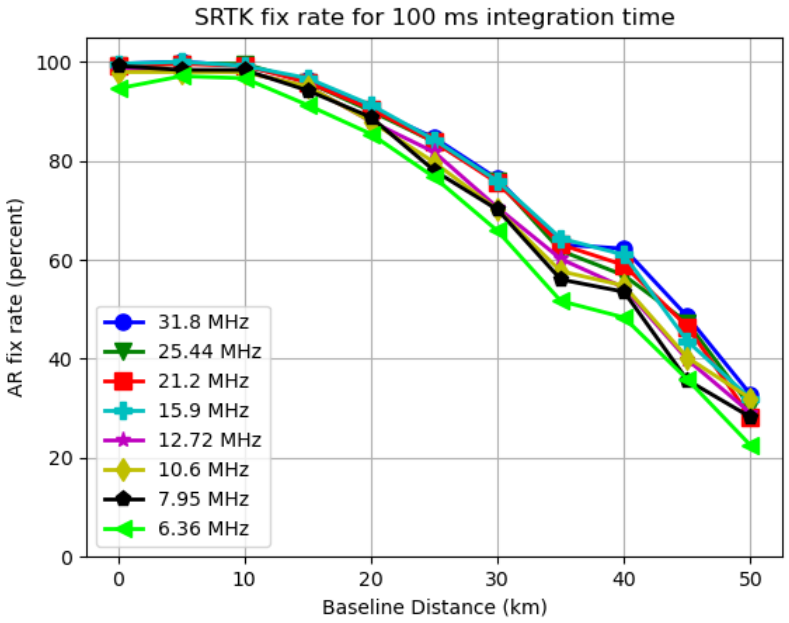

Figure 4.13: SRTK fix rate for different bandwidths when 100 ms integration time under different baselines

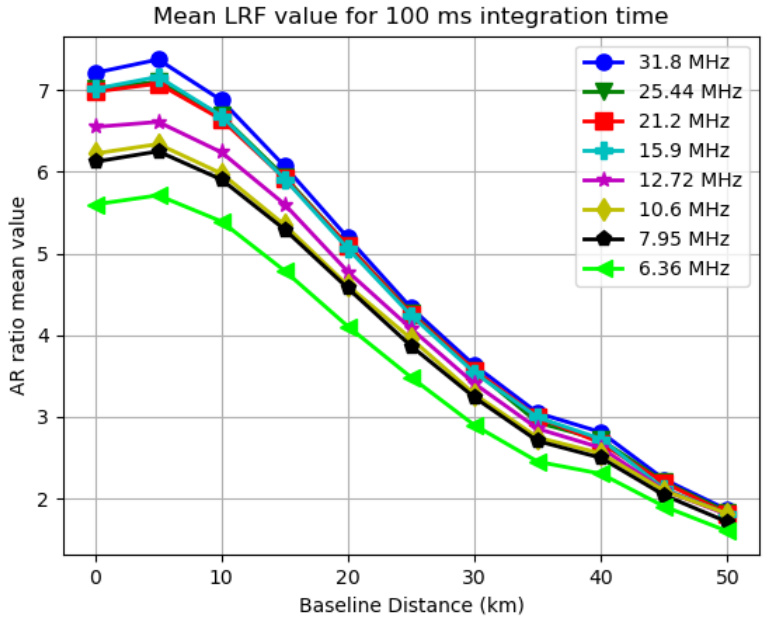

Figure 4.14: MeanLRF value for different bandwidths when 100 ms integration time under different baselines

Besides, when the baseline distance is less than 10 km, the decreasing trend does not appear to be quite obvious as the SRTK fix rate curve is almost flat and the mean LRF variation is also small. As the baseline distance increases, the SRTK fix rate drops more rapidly, which could be caused by the simple DD measurement modeling adopted in the RTK filter. More specifically, with 100 ms of integration time, and a baseline shorter than 15 km, more than 90% of snapshots generated valid RTK fixed solutions, regardless of the signal bandwidths used. In contrast, when the baseline distance increased to 50 km, the fix rate dropped to approximately 30%, with an average LRF value below 2.

In addition, Figure 4.13 depicts that the bandwidth also impacts the fix rate. The green line representing 6.36 MHz of sampling rate is lower than the one for 31.8 MHz by approximately 5% for most baseline distance cases. However, this impact seems much less significant compared to influences of baseline distance. As also shown in Figure 4.14, the mean LRF value is only slightly decreased from 7.2 to 5.8 for short baselines. This variation is even less noticeable under longer baseline scenarios.

Table 4.2: RTK fix rate (in percentage) for different integration times at different baselines.

	5 km	10 km	15 km	20 km	25 km	30 km	35 km	40 km	45 km	50 km
40 ms	92.53	90.87	87.14	82.16	74.69	66.39	53.11	51.45	40.66	31.54
60 ms	95.44	95.02	90.46	84.65	78.01	70.54	55.6	46.89	37.34	28.63
80 ms	98.34	98.76	94.19	88.38	80.91	74.69	59.34	56.02	43.98	27.80
100 ms	100	99.17	96.27	90.46	84.65	76.35	63.07	62.24	48.55	32.78

Table 4.3: Mean LRF values for different integration times at different baselines.

	5 km	10 km	15 km	20 km	25 km	30 km	35 km	40 km	45 km	50 km
40 ms	5.92	5.56	4.99	4.35	3.69	3.06	2.51	2.38	2.03	1.73
60 ms	6.58	6.18	5.49	4.67	3.9	3.23	2.64	2.32	2.04	1.73
80 ms	7.15	6.72	5.93	4.99	4.1	3.48	2.81	2.55	2.14	1.74
100 ms	7.38	6.88	6.07	5.19	4.33	3.63	3.05	2.81	2.23	1.86

The fix rate and meanLRF results for different integration times at different baselines are listed in Table 4.2 and 4.3, using a signal bandwidth of 31.8 MHz. Both tables show

that the performance of longer integration time is generally better compared to those shorter ones, although its impact is not as significant as the baseline distance. By looking at Table 4.2, an 8% improvement is observed when increasing the integration time from 40 ms to 100 ms, while this improvement becomes less noticeable when the baseline distance grows larger than 40 km. Table 4.2 also shows that the fix rate can still reach 90% for all the integration times used when the baseline distance is smaller than 10 km, and a fix rate of 80% can be assured for 20 km baselines. Results in Table 4.3 further confirm that the mean LRF improves when longer integration time is used. An overall 20% improvement in the meanLRF value is observed when integration time increases from 40 ms to 100 ms, except for the scenarios with baseline distances over 40 km.

Table 4.4: Snapshot data size (top value, in kB) and SRTK fix rate (bottom value, in percentage) for different integration times and bandwidths at 15 km baseline.

	31.8 MHz	25.44 MHz	21.2 MHz	15.9 MHz	12.72 MHz	10.6 MHz	7.95 MHz	6.36 MHz
40 ms	636	508.8	424	318	254.4	212	159	127.2
	87.14	92.95	87.14	89.21	93.36	88.38	82.16	78.84
60 ms	954	763.2	636	477	381.6	318	238.5	190.8
	90.46	91.29	87.97	90.46	90.04	88.38	86.72	82.99
80 ms	1272	1017.6	848	636	508.8	424	318	254.4
	94.19	94.19	92.95	94.19	90.87	90.46	90.87	87.55
100 ms	1590	1272	1060	795	636	530	397.5	318
	96.27	95.85	95.85	96.68	95.02	95.02	94.19	91.25

Finally, the data size corresponding to the chosen settings are represented in Table 4.4 together with the corresponding SRTK fix rates. To show the relationship between the SRTK fix rate and the data size more clearly, all their values at 15 km baseline distance are presented in Figure 4.15. Note that results from different integration times are represented by different colored markers to show that snapshot data size increases along with the integration time.

In general, larger snapshot sizes result in a higher possibility of fixing SRTK ambiguities. In this regard, the smallest data size used in this section is 127.2 kB and result in a 78.84% fix rate, while the largest data size used is 1590 kB, which is 12.5 times of the smallest

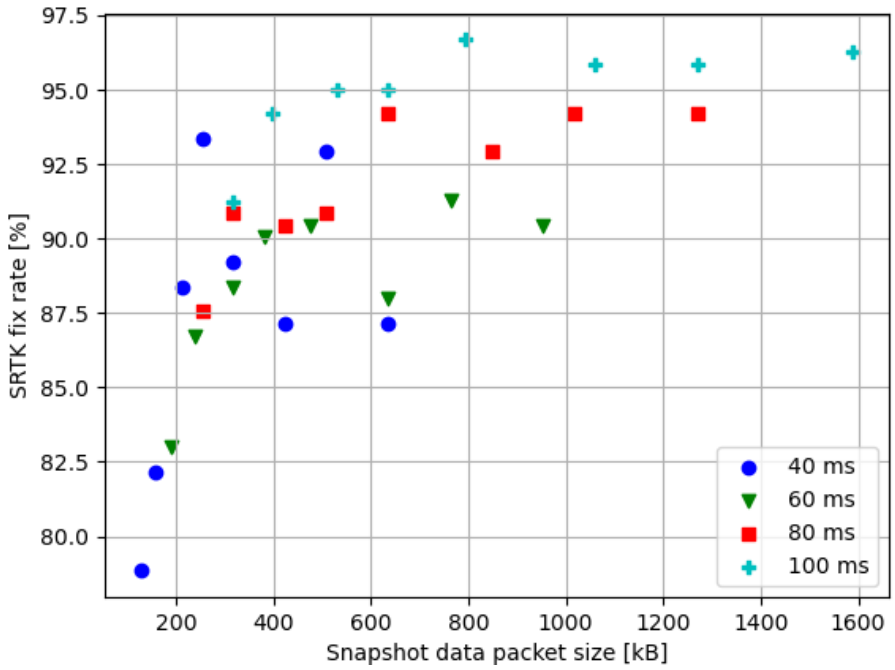

Figure 4.15: SRTK fix rates in relation to the snapshot sizes at 15 km baseline for different integration times.

data size, brings a fix rate of 96.27%. There is an obvious trade-off to consider between the data size and the expected accuracy. Assuming a minimum acceptable fix rate of 90%, the smallest data size able to achieve this goal in the test scenario is 254.4 kB, by using 12.72 MHz of bandwidth and 40 ms of integration time, for an observed fix rate of 93.36%.

The errors in the estimated coordinate are depicted in Figure 4.16 and Figure 4.17 to show the accuracy performance of SRTK when carrier phase ambiguities are fixed. These results are generated using 100 ms of integration time, 31.8 MHz signal bandwidth and 5 km baseline distance, for which a 100% fix rate was achieved, as shown in Table 4.2.

Figure 4.16 shows a small bias in the solutions, respectively -0.11 cm, 0.93 cm and 0.48 cm for East, North and Up directions. These biases might be a result of the inaccuracies in the ground truth, which was computed by averaging RTK fixed solutions from a closed-loop receiver (u-blox F9P). The corresponding standard deviations are 0.29 cm, 0.41 cm and 0.59 cm. It can be also seen in Figure 4.17 that the position points are generally located inside a 2 cm x 2 cm region with a calculated CEP of 1.012 cm, as denoted by the red circle. The

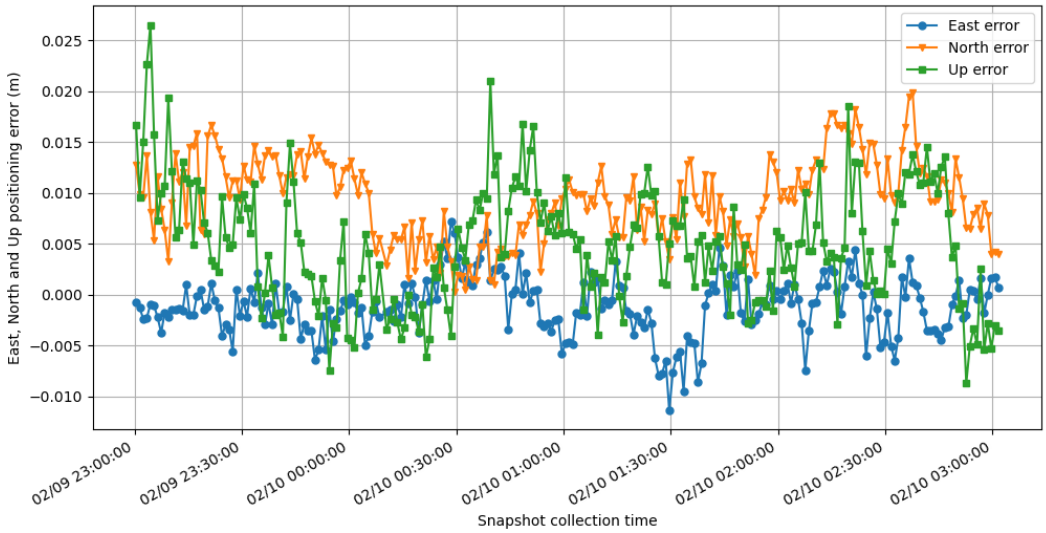

Figure 4.16: East (blue), North (orange) and Up (green) errors of SRTK fixed solutions, based on 100 ms of integration time, 31.8 MHz signal bandwidth and 5 km baseline distance

horizontal positioning RMSE is 1.066 cm and the 3D RMSE is 1.309 cm. The 95th percentile (P95) errors in the horizontal domain (denoted by the green circle) and 3D errors are 1.619 cm and 1.921 cm, respectively.

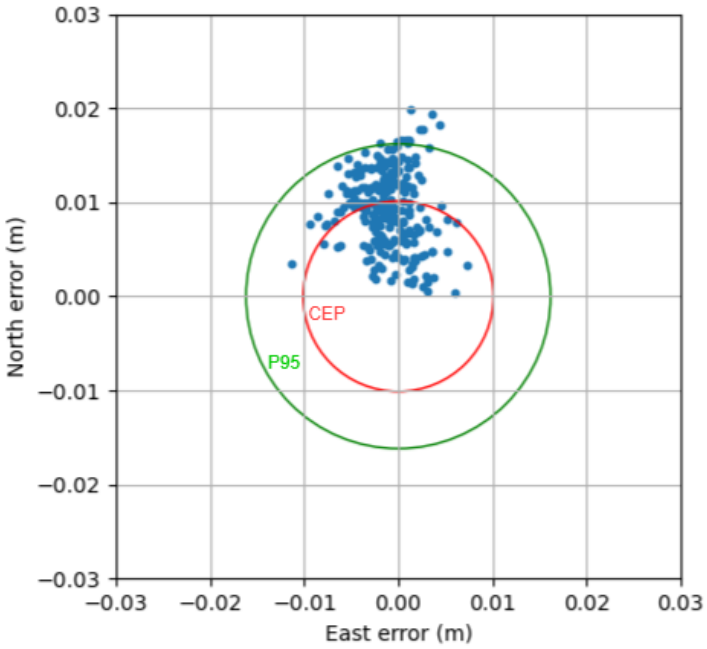

Figure 4.17: Horizontal positioning errors in East and North directions, based on 100 ms of integration time, 31.8 MHz signal bandwidth and 5 km baseline distance

Chapter 5

Data Bit Ambiguity

DBA is defined as the phenomenon when more than one of the data bit hypothesis candidates can bring about the detection of an acquisition energy peak. It is one of the main challenges faced by the SRTK technique that must be dealt with so that a unique set of snapshot measurements is obtained. The present chapter is dedicated to such DBA issues and describes some methods that can be applied to avoid, or at least to alleviate, the destructive influences that DBA poses on the SRTK performance.

Section 5.1 describes the fundamental reasons behind such phenomenon and the resulting HCEs that may appear in the carrier phase measurements. Due to the differences in signal structure of pilot signals and data signals, two new methods are introduced in this chapter, respectively aiming at these two types of signals. The first method is based on exploring the possible consensus of SCI values that can be formed among all satellites, the full description and performance analysis for this method is presented in section 5.2. The second method is an iterative process that focuses on the change of LRF result after each attempt of HCE correction. This method is discussed with more details in section 5.3.1.

5.1 Introduction

The main difference of snapshot baseband processing module from the traditional close-loop receivers is the application of the MH acquisition. As shown in Equation (3.1), the two

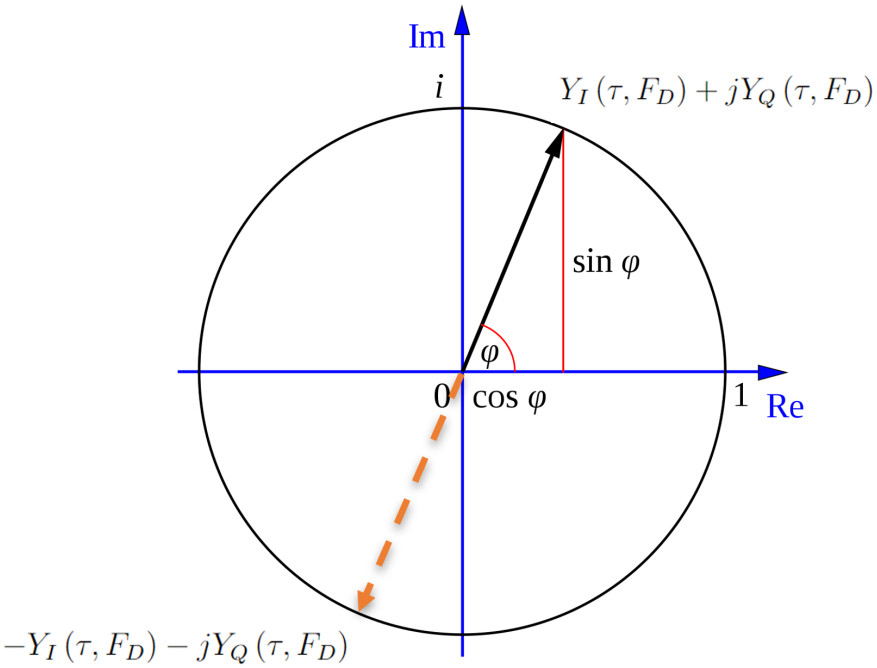

Figure 5.1: The ambiguous situation with two carrier phase measurements separated by half a cycle.

branches of correlation results are computed including the multiplication of the data symbols $d[n-\tau]$ from one of the predefined hypotheses. It is recalled that the estimation of the acquisition results are based on evaluating the correlation energy computed as $Y_I^2(\tau, F_D) + Y_Q^2(\tau, F_D)$. When the hypothesis symbols $d[n-\tau]$ are exactly the same as the actual symbols encoded in the collected signal, the maximum correlation energy is obtained. However, this is not the only situation that leads to the correlation energy peak. In fact, when the symbols of $d[n-\tau]$ are exactly opposite to the actual symbols, the same maximum energy magnitude can be obtained while the complex output of the correlation output can be represented as $-Y_I - jY_Q$, instead of the nominal output $Y_I + jY_Q$.

The carrier phase error caused by this ambiguous situation is plotted in Figure 5.1. Assuming that the black arrow represents the nominal complex correlation result (normalised), then the orange dashed line arrow shows the results obtained when multiplying symbols with exactly opposite signs. It can be seen that there is a 180 degrees offset between the complex angles of these two sets of results, such carrier phase offsets are referred to as the HCE.

Note that the SRTK performance will not be impacted if such carrier phase HCEs exist in all satellites, this is because they will be cancelled out when performing measurement differencing between satellites. The problem, however, is when only part of the satellites present such errors.

5.2 Pilot Signals

5.2.1 Current Limitations

Thanks to the fact that pilot signals do not contain navigation message data bits and the whole sequence of secondary code symbols is known beforehand, the MH acquisition module is able to generate the SCI values as one of the snapshot measurements. The SCI values are important for the SRTK processing mainly for the following two reasons:

1. to compute an accurate secondary code delay $\tau_{sc}^{\prime(k)}$ based on Equation (4.1), which will leave a resolution of a full secondary code period T_S for the navigation filter to determine the global time tag of the observables, as described in section 4.2.2. Only when accurate SCI values are known, the satellite transmission times can be properly computed based on the integer round-off procedure and the satellite position errors are within an acceptable range for the RTK navigation filter.

2. to ensure that the secondary code symbols are estimated correctly and the carrier phases are free from potential HCEs.

For pilot signals, DBA issues do not exist when the snapshot signal is long enough. This is because there is no other subsets of the known sequence that contains exactly the same symbols or with exactly the opposite signs, and thus the MH acquisition module produces a unique set of measurements that correspond to the actual SCI hypothesis. The DBA issue gradually appears when the snapshot length decreases, as discussed in the following example.

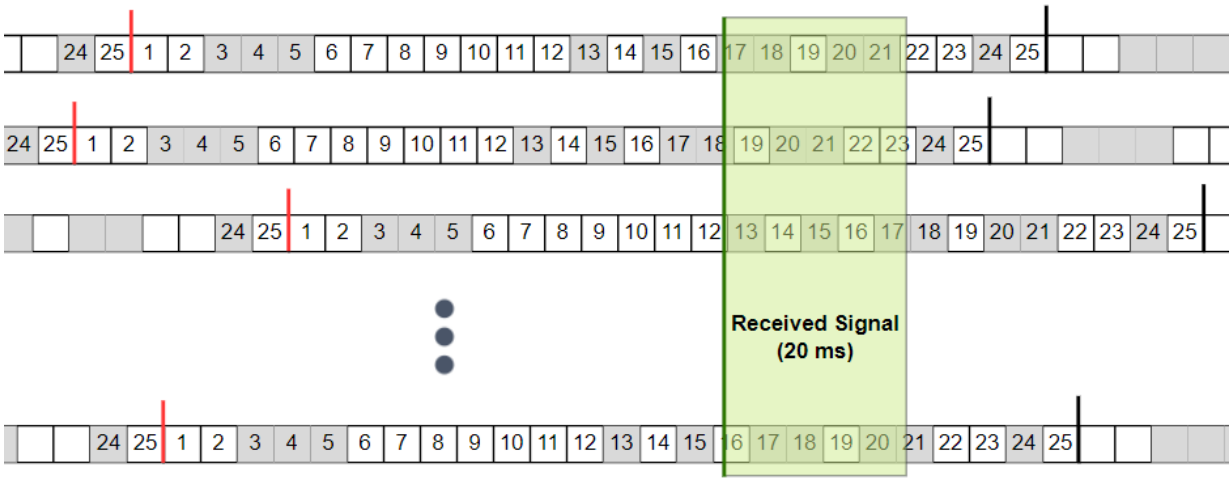

Figure 5.2: Different Galileo E1C signal SCIs for different satellites (represented as different rows) in a 20 ms snapshot signal (depicted in green). The vertical lines represent the start (red) and end (black) of a full Galileo E1C signal secondary code period of 100 ms.

Figure 5.2 presents different SCIs of different satellites in a 20 ms snapshot recording. The vertical lines represent the start (red) and end (black) of a full Galileo E1C signal secondary code period of 25 bits, lasting 100 ms. Each row represents one satellite and each square box carries one secondary code symbol that lasts 4 ms, where the shaded ones represent bit 0s and blank ones represent bit 1s. It is recalled that such secondary codes are identical for all satellites, as shown by the fact that all rows share the same shade pattern. The number inside each box represents their SCI, counted from the secondary code edge (marked in red). In this example, the actual SCI values for these satellites at the start of the received snapshot signal are $[17, 18, 12, \ldots, 16]$. The CAF energy peak should be found under these hypotheses.

However, because the snapshot data is so short, the CAF peak can be detected for some other particular SCI hypotheses as well. Such incorrect (i.e. ambiguous) cases can be divided into two groups based on their potential consequences on the secondary code delay $\tau_{sc}^{\prime(k)}$ and the carrier phase measurements:

1. The ambiguous SCI lead to secondary code symbols that are exactly **the same as** the

actual ones. In these scenarios, the false SCI values only disrupt the computation of the secondary code delay $\tau_{sc}^{\prime(k)}$ but not the carrier phase measurement.

2. The ambiguous SCI lead to secondary code symbols that are exactly **opposite to** the actual ones. In these scenarios, the false SCI values disrupt the computation of the secondary code delay $\tau_{sc}^{\prime(k)}$ and at the same time introduce HCEs to the carrier phase measurements.

Figure 5.3 shows all the 4 possible ambiguous cases for Galileo E1C pilot signal with a received signal length of 20 ms. Six secondary code symbols (i.e. $N_S = 6$) are required following Equation (3.9), with $t_{coh} = 20\ ms$ and $T_S = 4\ ms$. The coloured (green, blue and pink) rectangles represent the locations of the local replica secondary codes according to their SCI hypotheses.

The top row depicts the ambiguous scenario 1, where SCI indexes of 6 and 7 lead to exactly the same secondary code symbols, both with the sequence of [1 1 1 1 1 1]. In contrast, the other three rows depict the ambiguous scenario 2; the pairs of ambiguous SCI values all lead to exact opposite secondary code symbols, and thus are shaded by different colors. For instance, the second ambiguous case has SCI of 3 (blue) and 25 (pink) that correspond to the sequences of [0 0 0 1 1 1] and [1 1 1 0 0 0] respectively. In these cases, the carrier phase measurements will contain a HCE if the acquisition result sets are chosen with the wrong SCI value.

There is an upper limit in the duration of received signal for which DBA issues can still appear. When the snapshot length is long, a unique set of acquisition result can be found instead of multiple ambiguous sets. In these long-snapshot scenarios, the DBAs are automatically solved in the MH acquisition process. This way of solving DBA issues is named as the **satellite independent method** in this thesis since the SCI determination process is only related to one particular satellite, in contrast to the **consensus-based method** proposed later. Also, when the snapshot is shorter than this limit, it does not mean that DBA will definitely happen. For the example shown in Figure 5.2, only the last row (SCI of

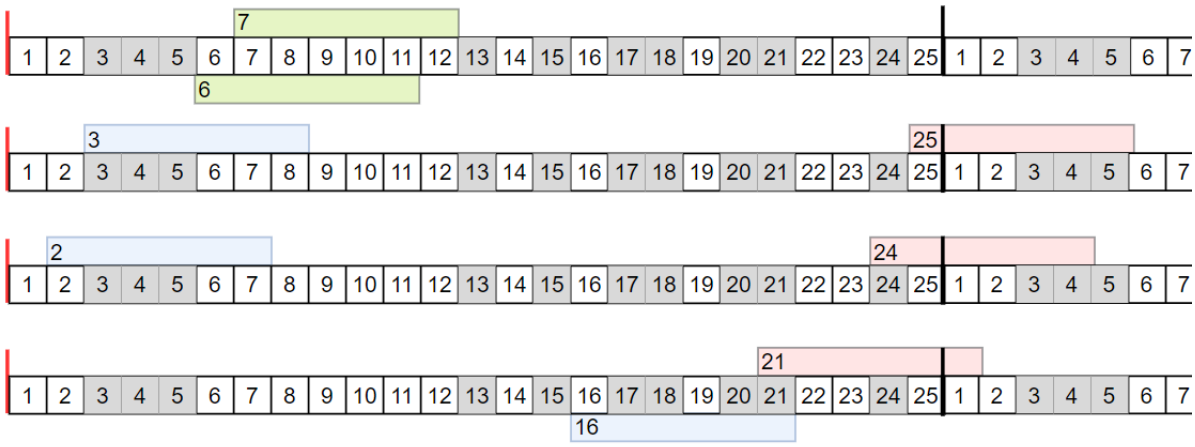

Figure 5.3: Four ambiguous cases that are possible for 20 ms of Galileo E1C signal. The first row show two SCI hypotheses lead to exactly the same secondary code symbols (in green), whereas the other three rows show different SCIs leading to exact opposite secondary code symbols (blue and pink) and hence can cause HCE in the corresponding carrier phase measurements. The vertical lines represent the start (pink) and end (black) of a full Galileo E1C signal secondary code period of 100 ms.

16) could be impacted by the DBA issue since Figure 5.3 has shown that [2, 3, 6, 7, 16, 21, 24, 25] are the only 8 candidates potentially exposed to such issue in this example.

These 8 SCI candidates among all the 25 possible values implies a probability of 32% that a particular satellite can not obtain a unique acquisition result set. Table 5.1 presents the results of such probability calculations for other snapshot lengths as well as their theoretical number of ambiguous SCIs (denoted by N_{amb}). Table 5.1 also includes other details such as N_S following Equation (3.9), and the maximum N_{hyp}, which is defined as the maximum number of SCI candidates that lead to the same secondary code sequence. Note that the minimum coherent integration time tested is 4 ms to ensure at least one full primary code period is included.

Table 5.1 shows that the DBA issue vanishes in MH acquisition results when the collected Galileo E1C signal has a duration greater than 24 ms. Only under this condition it can be assured that the resulting snapshot carrier phase measurements are free from HCEs. In the light of these results, a new method has been developed in order to alleviate such limitations and ensure the absence of DBAs in carrier phase measurements for integration times shorter

Table 5.1: Number of ambiguous SCIs for Galileo E1C pilot signal when acquired independently.

T_{coh} **[ms]**	N_S	N_{amb}	**Possibility of DBA**	**Maximum** N_{hyp}
(0, 4]	2	25	100%	12
(4, 8]	3	25	100%	4
(8, 12]	4	25	100%	3
(12, 16]	5	17	68%	2
(16, 20]	6	8	32%	2
(20, 24]	7	2	8%	2
(24, ∞)	≥ 8	0	0%	1

than 24 ms.

Whenever the DBA issue appears, no matter which type, it blocks the SRTK engine as the global time tags and highly accurate pseudoranges can not be determined correctly. The actual SCI value must be picked out from the ambiguous set and the snapshot measurements should be filtered to remain only the correct set of data.

The SCI identification process is known as the Data Bit Ambiguity Resolution (DBAR) and the overall workflow including this procedure is presented in Figure 5.4. As it can be seen, the MH acquisition results contain multiple ambiguous sets for each satellite, depicted in the leftmost table with red texts. The superscripts of the measurements denote the satellite, ranging from 1 to N, whereas the subscripts from 1 to M^i stand for the SCI of the acquisition result sets for satellite i, where M^i is the total number of ambiguous sets of satellite i. Since the M^i value is different for different satellites, this table could have a different number of rows for each column.

Note that for each satellite, the code delay measurements are identical for all the ambiguous sets, thus, the full period ambiguity resolution process is not impacted by the existence of DBAs. The key in Figure 5.4 is that the transmission time differences among satellites are also computed in this process as they are a critical input for the DBAR process. Ideally, the DBAR process should filter the MH acquisition results and generate a unique set of solution for each satellite, just like in the nominal cases where DBA issue does not exist. Then, these filtered outputs (shown in the table with green texts) are fed to the SRTK engine together

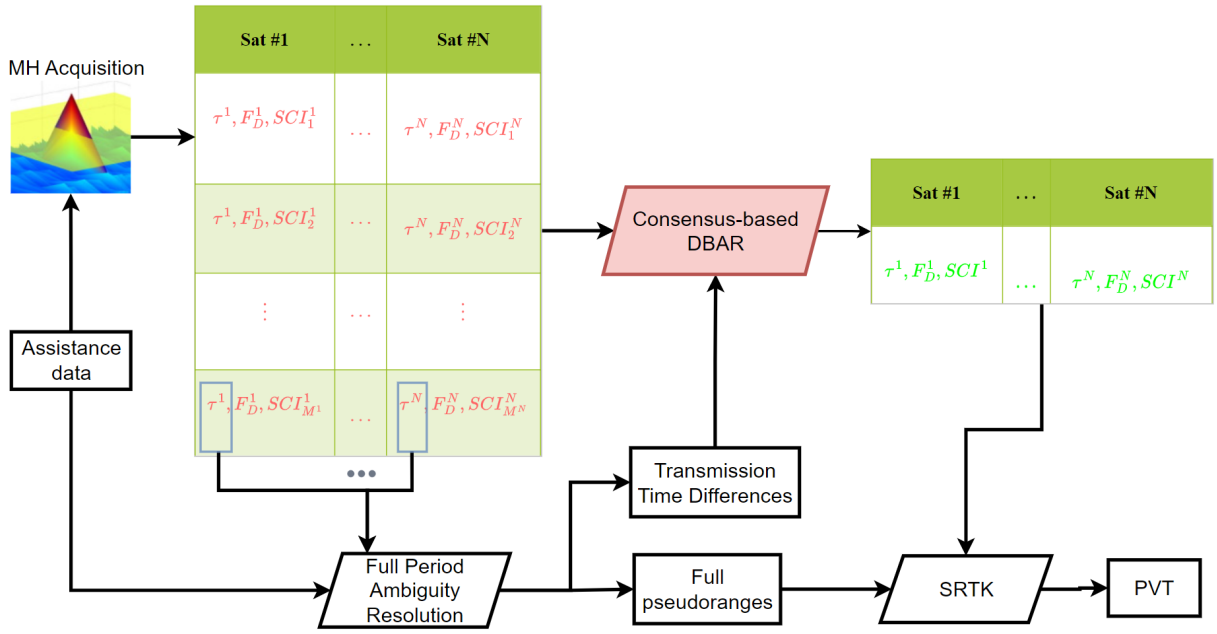

Figure 5.4: The overall SRTK workflow with the DBAR procedure for short snapshots.

with the full pseudoranges to obtain the PVT results.

5.2.2 Consensus-based Method

The consensus-based solution is based on the fact that the full period integer differences $N_c^{(k0)}$ can be accurately determined following Equation (3.18) as long as the coarse time and position errors are within acceptable ranges, as described in section 3.3. The transmission time differences can also be computed based on the $N_c^{(k0)}$ values since the reception time is common for all satellites. Since the a priori knowledge about the whole secondary code sequence is available, the mappings from transmission times to the encoded secondary codes are already known, thus, the expected relationships between SCI values of different satellites can be obtained as well. That is, if $N_c^{(k0)} = 1$, then it should be expected that $SCI^{(0)} - SCI^{(k)} = 1$, where $SCI^{(0)}$ and $SCI^{(k)}$ represent the SCI value for the reference satellite and satellite k, respectively. These relationships then work as a constraint for the SCI values obtained from the MH acquisition results.

The fundamental rule behind these restrictions is that all the satellites share the same

signal reception time. For any individual satellite k, its reception time can be represented as t_{rx} by Equation 5.1, as a summation of the transmission time $t_{tr}^{(k)}$ and the flight time $t_f^{(k)}$.

$$t_{rx} = t_{tr}^{(k)} + t_f^{(k)} \tag{5.1}$$

where $t_{tr}^{(k)}$ and $t_f^{(k)}$ are the transmission time and flight time of satellite k, respectively. The former is represented in Equation 3.11 as the combination of the integer part Y_c and the fractional part τ_r of a primary code period T_c, i.e., $t_{rx} = \tau_r + Y_c \cdot T_c$. Equation 3.11 can be re-defined concerning the secondary code period T_S as:

$$t_{tr}^{(k)} = \tau^{(k)} + SCI^{(k)} \cdot T_c + X_S^{(k)} \cdot T_S \tag{5.2}$$

where $X_S^{(k)}$ is an integer that corresponds to the transmission time, analogous to $X_c^{(k)}$ but with a lower resolution in time (T_S instead of T_c, e.g., 100 ms instead of 4 ms for Galileo E1C signal). The signal flight time $t_f^{(k)}$ can be represented by $\tau'^{(k)} + N_c^{(k)} \cdot T_c$ according to Equation 3.12. Then, Equation 5.1 can be re-written as:

$$t_{rx} = \tau^{(k)} + SCI^{(k)} \cdot T_c + X_S^{(k)} \cdot T_S + \tau'^{(k)} + N_c^{(k)} \cdot T_c \tag{5.3}$$

Notice that it was defined in Figure 3.6 that $\tau^{(k)} + \tau'^{(k)} = T_c$, i.e, their sum is actually equal to one full primary code period T_c, the above equation can be further simplified as:

$$t_{rx} = X_S^{(k)} \cdot T_S + (N_c^{(k)} + SCI^{(k)} + 1) \cdot T_c \tag{5.4}$$

The reception time can be computed using Equation 5.4 for two different satellites, such as a reference satellite indexed as 0 and another indexed as k, the following equation can be obtained:

$$X_S^{(0)} \cdot T_S + (N_c^{(0)} + SCI^{(0)} + 1) \cdot T_c = X_S^{(k)} \cdot T_S + (N_c^{(k)} + SCI^{(k)} + 1) \cdot T_c \tag{5.5}$$

Note that $N_c^{(k)}$ can be expressed as $N_c^{(0)} + N_c^{(k0)}$, where $N_c^{(0)}$ is the random integer chosen for the reference satellite and $N_c^{(k0)}$ is the full period integer difference that is accurately determined based on section 3.3. After further simplification, the final equation linking SCI values and $N_c^{(k0)}$ results can be obtained:

$$SCI^{(k0)} = X_S^{(k0)} \cdot T_S/T_c - N_c^{(k0)} \tag{5.6}$$

Where $X_S^{(k0)} = X_S^{(k)} - X_S^{(0)}$ is the difference between integers of secondary code periods of their transmission times. Most of time $X_S^{(k0)}$ is zero unless the transmission times are located in two different T_S periods. The term $SCI^{(k0)} = SCI^{(k)} - SCI^{(0)}$ is the SCI difference between the two satellites. Equation 5.6 implies that the difference in full period integers $N_c^{(k0)}$ can be used to deduce $SCI^{(k0)}$, which reveals the differences between the SCI values of different satellites.

The new consensus-based method can be generally divided into five steps:

1. The MH acquisition generates all the SCIs that lead to a acquisition peak for each satellite, including the ones with correct secondary codes and the ones with exactly opposite signs. For each satellite, only one candidate is correct and the target is to identify it as other candidates could lead to HCEs.

2. Perform the full period ambiguity resolution and store the $N_c^{(k0)}$ results. Note that the reference satellite is always indexed as 0.

3. Shift all potential SCI candidates (computed in step 1) for each satellite by an amount corresponding to $N_c^{(k0)}$ obtained in step 2 and obtain the modulus of shifted indexes for each satellite (in order to remove the $X_S^{(k0)}$ term).

4. Find a unique common integer value among the modulus of all shifted indexes (computed in step 3). This process can be achieved by a weighted voting procedure, which is described in the following paragraphs. In this way, the SCI ambiguities for each

satellite are resolved.

5. Shift back the unique integer values obtained in step 4 according to their flight time differences (by the same amount as in step 3) and retrieve the actual SCI values for each satellite. Finally, filter out other measurement values that correspond to the wrong SCI candidates

Figure 5.5 provides an overall example of the consensus method, using Galileo E1C pilot signals (recall that one full Galileo E1C signal secondary code period contains 25 bits). The integration time T_{coh} is 12 ms and thus, $N_S = 4$ following Equation (3.9) with a 100% probability of DBA according to Table 5.1. As in the previous discussion, each row represents one satellite.

The top right corner at Figure 5.5-(a) shows an example of the MH acquisition results. The red square boxes represent three ambiguous SCI hypotheses that lead to a CAF energy peak, while only one of them is the correct one. Notice that according to Table 5.1, we can have a maximum $N_{hyp} = 3$, i.e., among all the ambiguous sets, each set contain no more than three SCI values with $T_{coh} = 12\ ms$.

The aim of the consensus-based method is to resolve those DBAs and find the actual SCI of each satellite with the help of the flight time difference information $N_c^{(k0)}$, which is provided at the top left corner. These differences are computed based on Equation (3.18) using the second satellite as the reference, following the Step 2 previously described.

Step 3 shifts all the ambiguous SCI candidates for each satellite by an amount that corresponds to their flight time differences and update the SCI values, as shown in Figure 5.5-(b). That is, the first, third, and N-th satellites are shifted by $N_c^{(10)} = 5$, $N_c^{(30)} = 7$, and $N_c^{(N0)} = 8$. In contrast, the second satellite has been shifted by a full secondary code period (i.e. 25 bits) in order to better illustrate the consistency among satellites under the secondary code rollover, which is taken into account by the modulus operation.

Step 4 finds a common integer of 4, based on the consensus among these shifted SCIs (shaded in green on the right side of Figure 5.5-(b).), while the other SCI candidates are

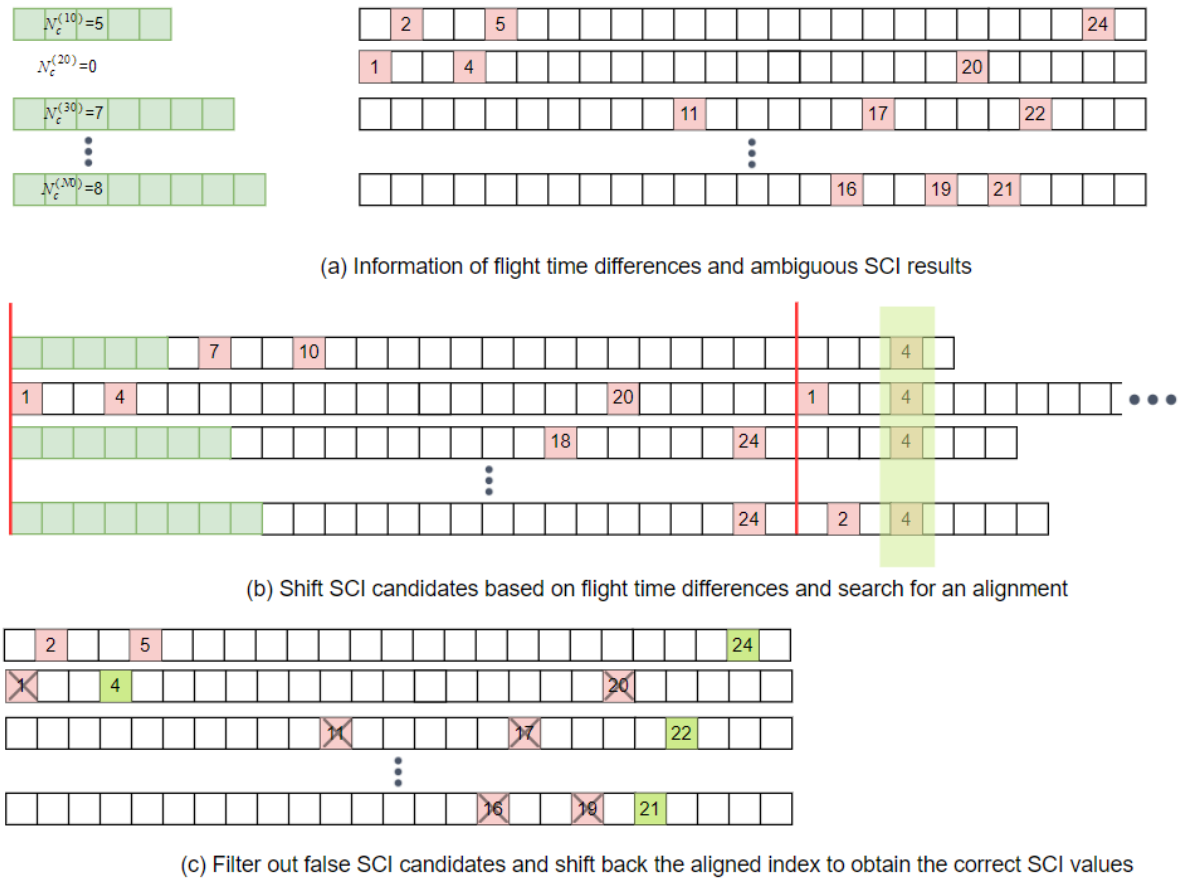

Figure 5.5: Example of DBA resolution based on transmission time consensus.

considered as false and are discarded. These false candidates are marked by a cross in Figure 5.5-(c).

Step 5 finally shifts back this common index with the same amount as $N_c^{(k0)}$, we can find the actual SCI values for each satellite. In the example shown at Figure 5.5-(c), we obtain SCIs as [24, 4, 22, . . . , 21]. The carrier phase measurements built based on these SCI values are free from HCEs.

In many actual recordings, the acquisition process could provide results that are prone to errors when the received signal strength is weak. When such errors exist in SCI results, it is possible that the proposed method can not find a common shifted SCI value from all the satellites. In this case, the consensus is destroyed by a minority that shows an anomaly due to false acquisition. In order to still obtain a robust solution in these scenarios, it is

proposed to perform a voting procedure to form the final consensus among all the satellites. Each satellite participates with a different weight, which can be based on, for instance, its energy magnitude obtained in the CAF. This procedure brings two advantages:

1. The voting weight leverages the different reliability of different satellites and this results in a final solution that agrees more with the satellites with higher correlation peaks.

2. This step ensures that a unique solution can still be found even when a minority of satellites show anomalies and interrupt the procedure of finding the common SCI.

Even though the proposed method brings great benefits for signals with short coherent integration times, it is still not possible to totally solve all the DBA issues, especially when the number of acquired satellites is reduced. The effectiveness of this method depends on three factors:

1. The diversity in the estimated satellite transmission times,

2. Total number of satellites,

3. The number of ambiguous SCI hypotheses for each satellite. N_{hyp}

The first factor depends on the satellite-receiver geometry. It determines the flight time of each satellite and since the reception time is common to all satellites, it results in the differences in the transmission times and SCI values among satellites. If all acquired satellites have very similar geometric distance to the receiver and result in almost the same flight times, the shifted SCIs as described in step 4 of section 5.2.2 would tend to be identical. Under such circumstance, the procedure of weighted voting could fail to find a unique index at step 5 and the acquisition result remains ambiguous. The second factor impacts the solution mainly because the more satellites participate in the voting, the higher chance that a unique solution can be obtained. The third factor is determined by the number of secondary codes encoded in the snapshot signal N_S, which is higher for longer coherent integration times and lower for shorter signals. For these reasons, unsuccessful DBAR could still happen when

the integration time is short and there is a limited number of satellites acquired. We have performed experiments under these scenarios and the results are presented in section 5.2.3.

5.2.3 Experiment Results

In order to show the improvements of the proposed consensus-based method compared to the conventional satellite independent method for SCI determination, the following experiment campaign was performed based on real snapshot GNSS signal recordings. The original snapshot data set used for evaluating the performance is the same as that used in section 4.4, which contains 240 snapshots across a duration of 4 hours. The difference with respect to the experiments performed in section 4.4 is that only Galileo E1C signals are used since this method mainly concerns pilot signals.

Since the traditional satellite independent method can already solve DBA issues when the snapshot length is longer than 24 ms, the experiments in this section were focused on snapshot lengths of {4, 8, 12, 16, 20} ms. The acquisition module was running in full coherent integration mode using the whole length of the truncated snapshot data. In order to get the true values of the SCI values of each satellite, the data set was processed in advance with long integration time of 40 ms using the satellite independent method to ensure that the detected SCI values are reliable. The results of the new method were then compared to these true values in order to evaluate its performance variation under different snapshot lengths.

The snapshot processing results are shown in Figure 5.6. The upper panel represents with green dots the results where successful DBARs are achieved a single set of acquisition output is filtered out, while results with red crosses show that the SCI ambiguities remain after the filtering process. The lower panel of Figure 5.6 shows the number of Galileo satellites that are acquired.

There are two metrics that should be taken into account when evaluating the performance of the consensus-based method:

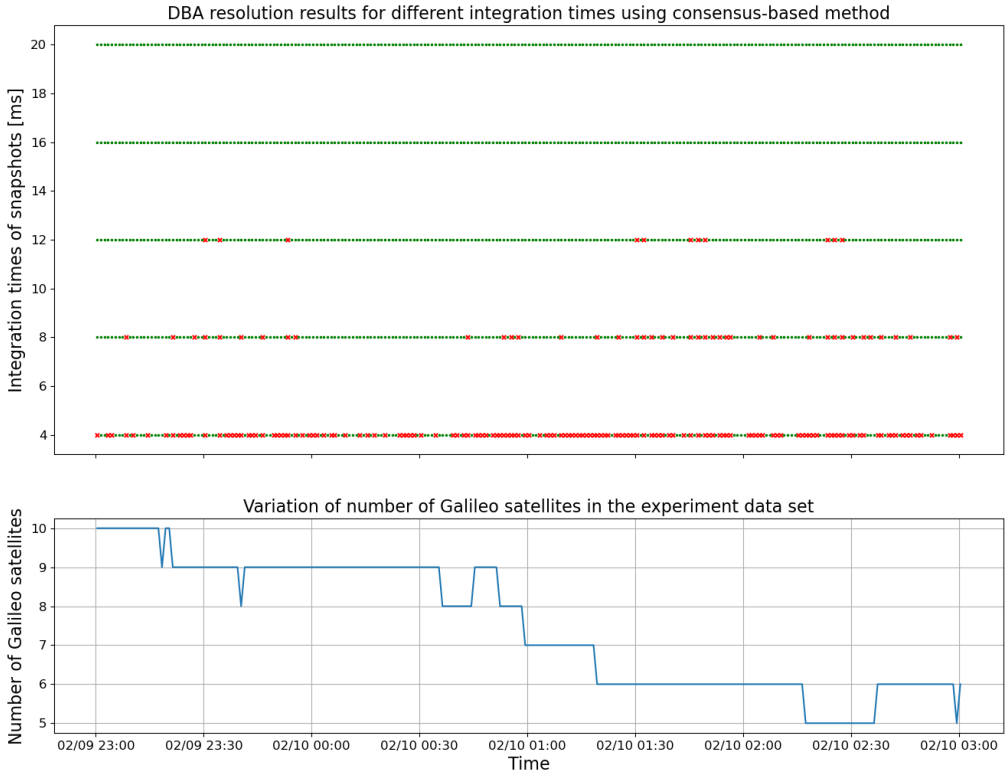

Figure 5.6: Snapshots with successful (green) and failed (red) DBAR procedure after applying the consensus-based method for different integration times.

1. Percentage of **Uniqueness** P_u. This value represents the success rate of achieving the DBA resolution. It shows the percentage of snapshots with unique set of acquisition results, as described in step 5 of the previously mentioned procedure.

2. Percentage of **Correctness** P_c of the final filtered acquisition results. It shows the rate of resulting unique sets of SCI values that are the same as the true values, for all satellites. A simple way to evaluate this is to verify the correctness of the shifted SCI value, which is computed in step 3 of the procedure. In this way, we only need to make the comparison once, instead of comparing SCI values for each satellites.

Table 5.2 lists P_u and P_c for different integration times. We can see that by applying the proposed consensus-based method, a unique solution can be obtained for all (i.e. $P_u = 100\%$) snapshots longer than 12 ms. As for the performance in terms of the correctness, the P_c values show that basically all the actual SCI values are filtered out correctly as long as a unique SCI set can be found. The only exception is when the snapshot is less than 4 ms, a few snapshots (7.4%) have been fixed to the wrong SCI values. This might be caused by the false acquisitions due to the short integration time and since only 2 secondary code symbols are involved, the larger number of hypothesis ($N_{hyp} = 12$ in Table 5.1) brings difficulties to the successful achievement of DBAR.

T_{coh} **[ms]**	P_u	P_c
(0, 4]	39.42%	92.6%
(4, 8]	82.37%	100%
(8, 12]	95.44%	100%
(12, 16]	100%	100%
(16, 20]	100%	100%
(20, ∞]	100%	100%

Table 5.2: Performance of consensus-based method in terms of uniqueness and correctness.

Figure 5.7 compares the results of the consensus-based method with the traditional satellite independent method. We can see that the latter (denoted by the blue line with cross markers) needs at least 24 ms of snapshot length to achieve $P_u = 100\%$, while the former,

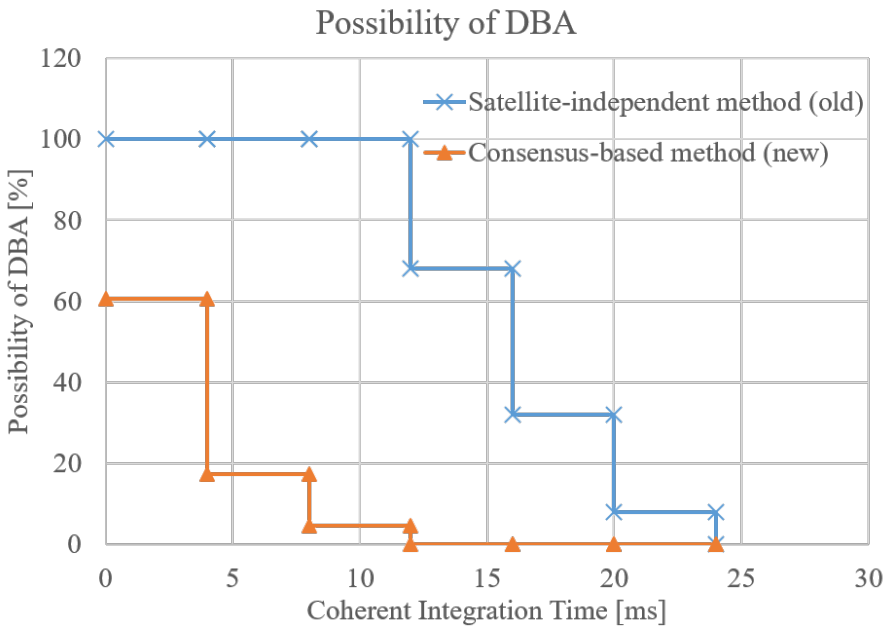

Figure 5.7: Probability of DBA issue (computed as $1 - P_u$) to happen for Galileo E1C signal using satellite independent method (blue) and proposed consensus-based method (orange).

as confirmed by Table 5.2 also, requires only 12 ms. When the snapshot length goes even shorter, the new method still shows superiority as it results in a much lower possibility of DBA.

5.3 Data signals

5.3.1 Current Limitations

The consensus-based method mentioned above can only be applied to pilot signals due to the following reasons:

- The a priori knowledge about the exact secondary code sequence does not exist for data signals, since the navigation message data bits are constantly changing and thus, there is no direct link between the transmission time and the encoded data bits that can be utilised.

- Even if the time-tagged navigation data bit streams are provided through external

assistance and the potential transmission time candidates can be deduced based on the u index values MH acquisition results, it would be still difficult to form a final consensus among all the satellites due to the limited diversity in transmission time differences compared to the bit duration. For example, GPS L1 C/A and Galileo E5a-I signals both employ navigation data bits that last 20 ms (i.e. modulated at 50 Hz). Considering that the transmission times are separated by a maximum of 30 ms due to the difference in geometric ranges from the receiver to different satellites, it is highly possible that the transmission epochs of many satellites are located in a time period that is covered by the same bit. This fact severely reduces the probability to find a unique solution.

For these reasons, it is not practical to attempt to resolve the DBA issues for data signals. In order to make use of the data signal measurements in the navigation filter, the potential HCE errors must be excluded from the snapshot carrier phase measurements.

5.3.2 Iterative Amendment Method

An iterative method has been developed by evaluating the LRF values computed with different subsets of measurements. Section 2.2.6.2 presented how the LRF values are computed and it is known that when the number of measurements increases in the navigation filter, the LRF value typically increases. However, if any carrier phase measurement contains HCE, the corresponding DD carrier phase measurement is contaminated by the same amount as well. Such erroneous measurement severely impacts the ratio test and the LRF value decreases to a value close to 1, which indicates that the second best solution of integer set is almost the same as the best solution in terms of their "closeness" to the float solutions. By observing the changes of the LRF value after including one measurement, it is possible to determine if the newly added measurement is affected by the HCE. The proposed method is based on iterative inclusions of satellites, and, once a HCE-impacted measurement is detected, it corrects the carrier-phase measurements by adding half a cycle and then proceed to the next

satellite inclusion until completing the iteration of such error detection.

As an initial step, the first LRF value (denoted by R_0) should be computed based on all measurements that are guaranteed to be free from HCEs. Pilot signal measurements, thanks to the consensus-based DBAR method mentioned before, can be used as a starting point, before any satellites with navigation data bits are iteratively included into the navigation filter.

The overall workflow of the proposed iterative method is presented in Figure 5.8. Starting from the initial LRF value R_0, one satellite measurement with potential HCE is included to the navigation filter and the corresponding new LRF value R_1 is computed. On another parallel branch, the newly included carrier phase measurement is intentionally added by a half cycle bias, obtaining a parallel LRF value of R_1'. The decision is made by comparing R_1 and R_1' values: if R_1 is larger, then we assume that the included measurement was free from HCE and vice versa. If the decision is that HCE exists in the newly added carrier phase measurement, it is directly amended by adding half a cycle and used as a base measurement for the next iteration of computation for R_2 and R_2'. The same procedure is repeated for other satellites and the index k in the Figure denotes the results after the k^{th} satellite is included. In this way all the measurements can be amended one by one, resulting in a final solution with an increased number of satellites included into the navigation filter.

The key in the proposed process is the decision making step. In some cases, the LRF value does not change much with added satellites, in some other cases, the LRF value even slightly decrease when a HCE-free carrier phase is included due to the bad quality of this newly added measurement. Thus, the simple strategy based on direct comparison of LRF, as shown in Figure 5.8, may not always lead to accurate HCE detection. Some additional conditions can be added to make the decision making process more accurate and robust to anomalies. These more advanced strategies are left for future research and the remaining part of this section focuses on the performance of the simple strategy shown in Figure 5.8.

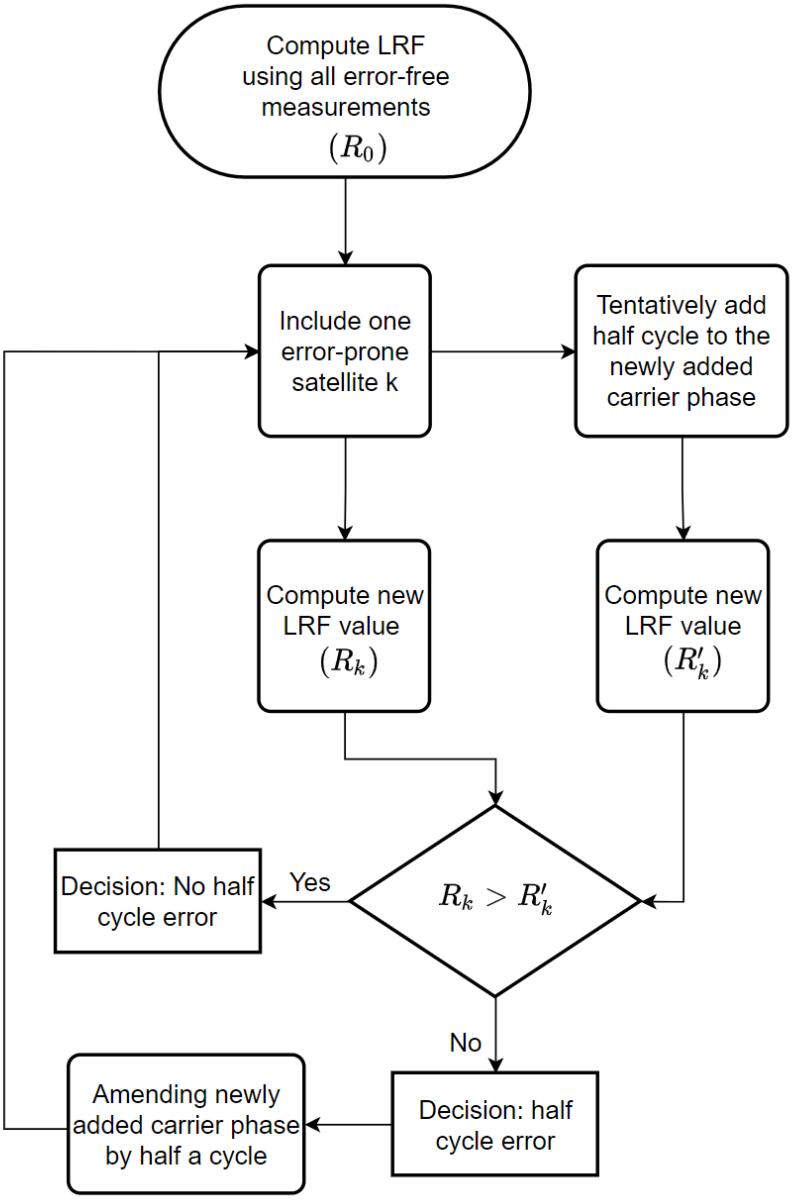

Figure 5.8: Workflow of HCE amendment based on iterative satellite inclusion. The decision making is based on direct comparison of the LRF values computed with and without adding satellites contaminated by HCE.

5.3.3 HCE Amendment Performance

The same data set as in Section 4.4 is used for the evaluation of the proposed method, consisting on a zero baseline configuration shown in Figure 4.5. Only GPS L1 C/A and Galileo E1C signals are processed for the experiments presented in this section. The Galileo E1C signals are used as the base measurements to compute the R_0 value. GPS L1 C/A signals are prone to possible HCEs and are included to the solution one by one according to the iterative method.

In order to verify the effectiveness of the proposed method, the actual carrier phase measurements are computed with the assistance of the actual navigation data bits that are collected from the data base of GFZ [65]. By post-processing the collected data with the knowledge of such data bits, the DBA issues can now be prevented from happening at the acquisition stage, resulting in snapshot carrier phases that are free from HCEs. These carrier phase values are then used as the ground truth. Then, additional HCEs are added to the carrier phase measurements of all GPS satellites on purpose for the evaluation of the proposed method.

Figure 5.9 shows an example of changes in the LRF after including one correct (.e., HCE-free) and one wrong carrier phase measurement containing a HCE. The green line shows the LRF value change when all the inclusions are free from HCEs, while the red branches show the results when the included measurements contain such errors. For this example, $R_0 = 1.08$ is computed using 9 Galileo satellites, whereas another 9 GPS satellites are included to the navigation filter one by one and their resulting LRF values are computed.

The LRF values (both computed with and without HCEs) and GDOP values after iterative GPS satellite inclusions are presented in the following table:

The result shows that if the included satellite carrier phase is correct, the LRF value basically increases, as shown by the green line and the R_k column in Table 5.3. In contrast, once a HCE is present in the newly included measurement, the LRF value drops drastically

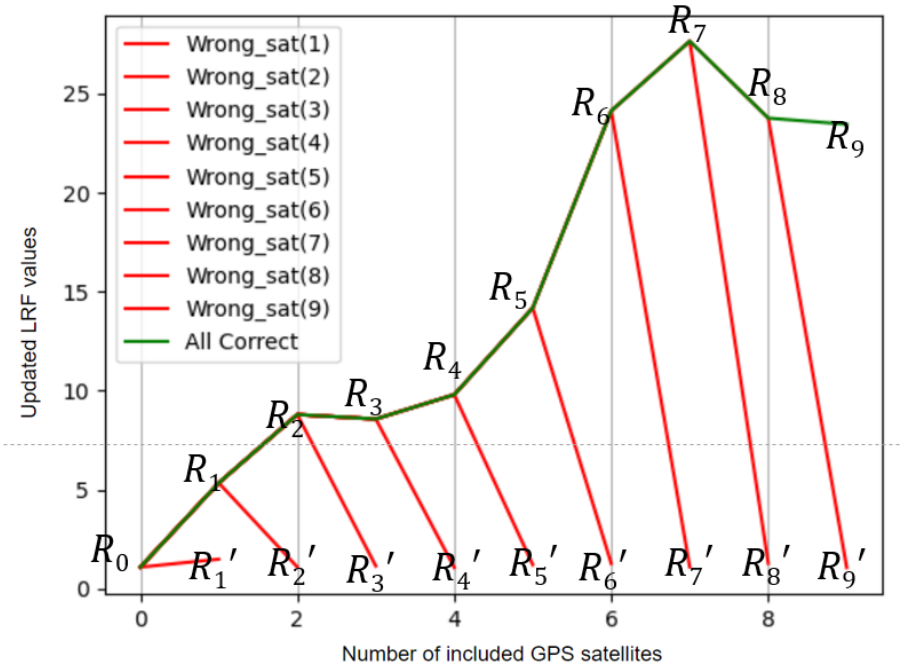

Figure 5.9: Satellite inclusion result for one snapshot with successful carrier phase amendment. Green line represents the LRF results after correct satellite inclusion, red lines represent the LRF values after including one satellite containing HCE.

to near 1, as indicated by the red branches and the R'_k column of Table 5.3. Note that the wrong addition is under the basis that all previous inclusions are correct and the pre-inclusion LRF value is computed without any HCEs. It can also be noticed that the GDOP follows a decreasing trend to be about 4 times better (from 1.55 to 0.37) when all GPS L1 C/A satellites are included, compared to the scenario before any inclusion, indicating a better estimated positioning accuracy. In this example, all the red branches reduce the LRFs and therefore are below the green line, which means that the proposed method can successfully detect and correct all the carrier phase HCEs. Unfortunately, such successful cases do not happen all the time, among all 239 snapshots only 103 snapshots have all their GPS satellite HCEs detected and amended correctly. This indicates a success rate of around 43.1% by simply applying the proposed method.

There are some other scenarios where the proposed method does not perform optimally, which points out that better strategies should be envisaged. Figure 5.10 shows an example

Number of inclusions k	LRF(correct) R_k	LRF(wrong) R'_k	GDOP
1	5.33	1.49	1.55
2	8.79	1.09	1.5
3	8.57	1.15	1.33
4	9.8	1.07	0.75
5	14.18	1.21	0.53
6	24.13	1.27	0.52
7	27.65	1.09	0.5
8	23.76	1.28	0.41
9	23.45	1.06	0.37

Table 5.3: LRF and GDOP variation results for one snapshot based on the iterative inclusion method.

where after the second inclusion, the resulting R_2 value computed without half cycle is smaller than its counterpart R'_2. As pointed out by the vertical red arrow, the red branch extends to the upper side of the green line. In this case the proposed method will fail to detect the HCE.

This situation is contrary to what is expected as it seems that the LRF computed with measurements containing a HCE is better than that with only correct measurements. Such LRF evolution could be caused by the large noise of the newly included measurement, although the root causes still need to be investigated more thoroughly.

Besides, it is noticed that the LRF value before the measurement inclusion impacts the successful identification of the HCEs as well. That is, there is a relationship between the probability of a successful repair of the k^{th} measurement and the pre-inclusion LRF value R_{k-1}. In order to evaluate how much impact this value has on the success rate, the following statistics are computed: For all the 239 snapshots processed, each time a satellite is included, a decision is made based on the proposed method. Thus, a total of 2026 decisions were made for all the data, among which 222 decisions were wrong. All these wrong decisions are made based on a relatively small LRF from previous measurements, when the previous LRF values are large, the follow up decisions about the HCE are generally correct. Figure 5.11 shows a distribution of the pre-inclusion LRF values for all the 222 wrong decisions (false detections)

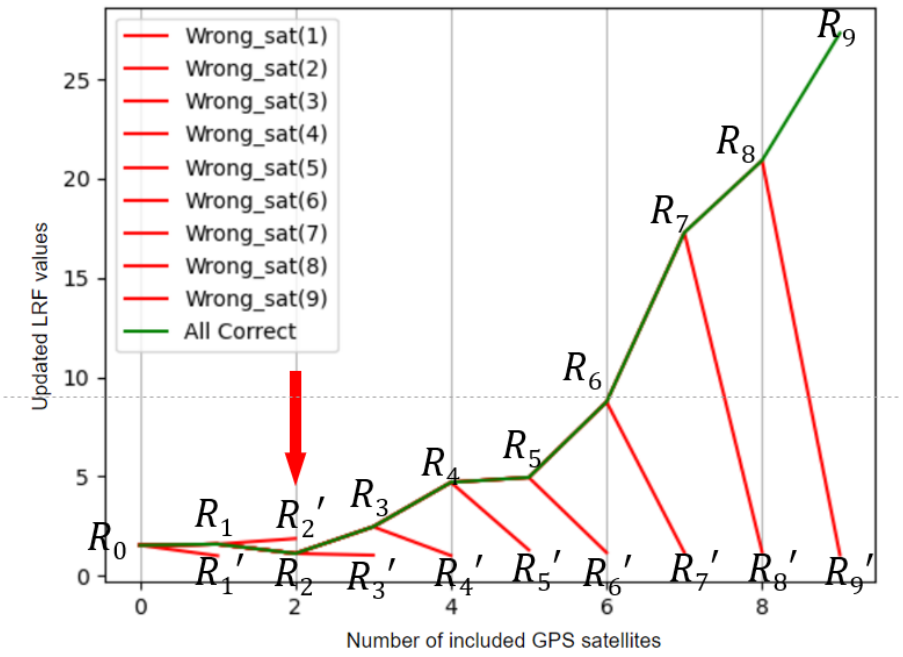

Figure 5.10: Result of the scenario where proposed method fails to detect the HCE that exists in the second satellite. Green line represents the LRF results after correct satellite inclusion, red lines represent the LRF values after including one satellite containing HCE. The red arrow shows when the proposed iterative inclusion method fails.

that were made.

It can be seen that when the LRF is already greater than 3, it is very likely that a correct decision can be made by the proposed method for the follow up satellite inclusion. The success rate for this scenario is about 99.03%.

In addition, the order of satellite inclusions also has an impact on the final success rate. If the more reliable satellites are included first, there is a higher chance that the following satellite can be corrected and the LRF value can be more reliably increased. For this, the same data set was used but the satellites are sorted in descending order in terms of their elevation angles. In this way, the satellites with higher elevation angles are included first to ensure a higher chance of correct identification and correction. The inclusion results after this modification were slightly better as a total of 113 snapshots managed to be corrected, equivalent to a success rate of 47.3% for the proposed method.

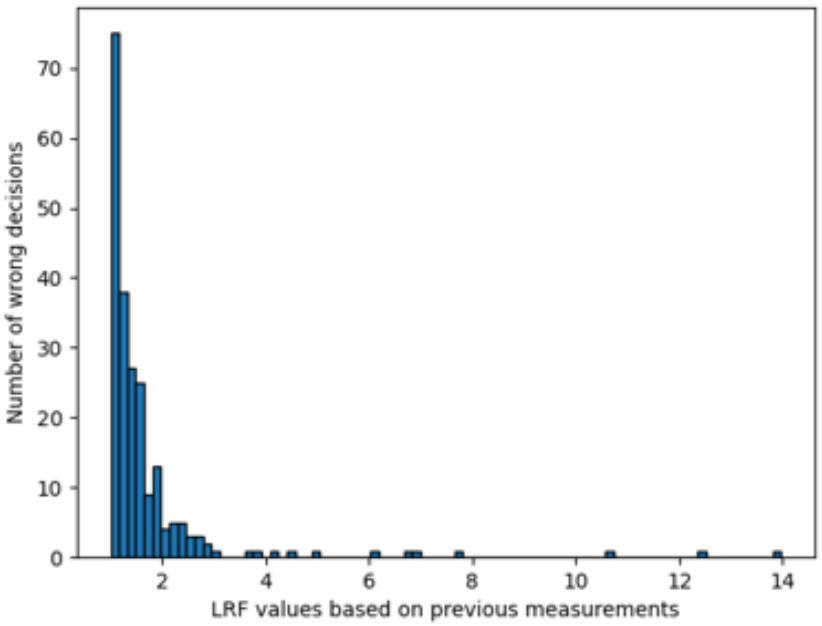

Figure 5.11: Histogram of pre-inclusion LRF values for all the false decisions been made.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The present PhD dissertation has put the focus on the implementation of high accuracy snapshot GNSS positioning techniques. Conventional snapshot positioning algorithms only reach meter level accuracy, since only snapshot code delay measurements are used. These conventional methods, including its signal processing and measurement data processing, have been reviewed and consolidated. The MH acquisition module is described and special care is taken in the hypotheses construction process. The snapshot pseudorange measurement generation process is presented by extending the typically used 1-ms code ambiguity to the more general full period ambiguity resolution procedure, which enlarges the amount of signals this method can be applied for. Besides, the fractional carrier phase generation for snapshot signals has been also described. The general performance of conventional snapshot technique has been analysed based on processing different sets of real-world snapshot collections. The results of Section 3.5.1 show that under nominal conditions, i.e., 40 ms of coherent integration time and 14.5 MHz of sampling rate, the absolute timing error of the conventional CTF solution is within few milliseconds, with a maximum error of 6.4 ms. As for the positioning accuracy, Section 3.5.2 showed that the horizontal errors are below 5 m while the vertical error presents a standard deviation of 3 m, being the maximum 3D error below 10 m. The

fundamentals established by the conventional snapshot positioning techniques are of vital importance for the following more advanced data processing.

My research encountered with two main challenges blocking high accuracy snapshot positioning results. Namely, the generation of full pseudorange measurements that are highly-accurate in terms of satellite transmission times and the presence of possible carrier phase HCEs that are caused by the DBA issues.

For the first challenge, Chapter 4 proposed a dedicated solution based on global time tag determination. The zero-baseline experiment results proved, for the first time, that SRTK is feasible and high-accuracy positioning results can be obtained based on snapshot GNSS recordings. Further experiments were performed and the SRTK performance was evaluated in more details, under different scenarios of processing parameters. The following conclusions can be drawn based on the results obtained:

- Increasing the coherent integration time used in the MH acquisition step improves the code measurements quality and as a result, increases the SRTK fix rate. The same principle applies for the bandwidth parameter. The higher it is, the more feasible the IAR becomes. Note that the bandwidth parameter is adjusted by changing the sampling rate of the collected snapshot signal. Thus, the increase of SRTK fix rate is at the price of a larger size of data to be transmitted to the cloud server and a corresponding higher computational burden. The trade-off between these parameters should be well considered when developing the snapshot receiver.

- When processing with only single constellations, Galileo signals perform slightly better than GPS signals according to the zero baseline results. By using multiple constellations together, a 100% fix rate can be achieved for all the data sets that have been processed under zero baseline mode.

- The availability of snapshot measurements from multiple frequencies dramatically increases the fix rate.

- Non-zero baseline SRTK positioning is also achievable. In fact, the navigation filter used for the experiments applied a short baseline model, i.e., ignoring the residual DD atmospheric delays. In this case, the baseline distance has a more significant impact on the fix rate than the bandwidth and integration time when the distance is larger than 10 km.

- Test results in Section 4.4 have also shown that a snapshot with the size of 255 kB could result in a potential fix rate of more than 93% with 15 km baseline distance.

- 100% fix rate was achieved with 100 ms of integration time and 31.8 MHz of sampling rate for 5 km baseline distance. When the snapshot carrier phase ambiguities are correctly fixed, centimetre level positioning accuracy can be obtained, as shown in Figure 4.17.

For the second challenge, different solutions are proposed for pilot signals and data signals. The pilot signals contain known sequence of secondary code bits and hence HCEs are eliminated from the source by resolving the DBA issues using a consensus-based method. This method exploits the inherent relationship between the flight time differences and the SCI values that can be obtained from the MH acquisition results. Real-world snapshot recordings were processed based on the proposed method to evaluate its performance compared to the traditional workflow where the DBA of each satellite is solved independently. Galileo E1C was chosen as the signal to be experimented and the results showed that the consensus-based method largely improves the capability to solve DBA issues. The consensus-based method guaranteed correct identification of SCI values for all satellites whenever the coherent integration time was longer than 12 ms. It remained partially effective even for signals as short as 4 ms. In contrast, 24 ms were needed by the traditional method based on the independent DBA removal. This better capability of obtaining correct SCI values effectively expanded the scope where high precision positioning can be achieved with snapshot data, thanks to the more genuine carrier phase measurements that are generated in the acquisition process.

For data signals, the consensus-based method can not be applied due to the lack of knowledge about the exact sequence of the encoded bits. In order to still obtain SRTK fixed solutions, another method is proposed focusing on detecting and amending the HCEs that still exists in the snapshot carrier phase measurements after the MH acquisition process. This method is based on observing the change of the SRTK LRF values after iteratively including the potentially erroneous satellite carrier phase into the navigation filter. This iterative method successfully detected and amended all the HCEs that were included in 43.1% of the snapshots that were tested. The other part of the snapshots in the data set were only partially fixed, which indicates room for improvement in the proposed method:

- First, the order by which the measurements are processed in the sequential inclusion algorithm can be improved: when satellites with higher elevation angles (and hence, lower measurement noise) are included first, the overall success rate increases from 43.1% to 47.3%.

- Second, enhanced decision making performance can be obtained when considering the pre-inclusion LRF values. When such value is already larger than 3, the proposed method obtained more than 99% success rate.

After the HCE identification and correction procedure, more satellites can be used to determine the user position and as a result, smaller GDOP values and thus better positioning solutions can be obtained.

6.2 Future Work

The present PhD work has initialised the investigations into algorithms dedicated to high-accuracy snapshot GNSS positioning, resulting in only the initial results of this subject. Some further investigations can potentially improve the performance of the proposed methods and algorithms, some recommendations for future researches in this subject are given below:

- For the non-zero baseline scenarios, the SRTK performance were only analysed using GPS L1 C/A and Galileo E1C signals. Following the experiments performed in this research, more signals could be processed to statistically characterize the SRTK performance in a robust manner.

- The consensus-based method still does not guarantee 100% of DBAR for Galileo E1C signals, especially when snapshots are shorter than 8 ms. It is recommended to further improve the performance of this algorithm by using satellites from other constellations and signals. Besides, a more extensive use of time assistance and a narrower window can be applied to filter out the SCI hypothesis with wrong transmission times.

- The proposed iterative HCE detection and amendment scheme can be refined in order to obtain a self-contained method that tackles HCEs with higher efficiency and effectiveness than the one proposed in the present work. Possible improvements may include carrier phase residual data analysis and multiple measurements inclusion simultaneously. Additionally, a better inclusion sequence can be investigated to not only consider the elevation angle, but also other parameters such as the signal to noise ratio.

Acronyms

ACF	Auto Correlation Function
ADC	Analogue to Digital Converter
AGNSS	Assisted GNSS
AI	Artificial Intelligence
APC	Antenna Phase Center
BDS	BeiDou Navigation Satellite System
BOC	Binary Offset Carrier
BPSK	Binary Phase Shift Keying
CAF	Cross Ambiguity Function
CAGR	Compounded Annual Growth Rate
CEP	Circular Error Probable
CMC	Carrier Minus Code
CORS	Continuously Operating Reference Station
CRB	Cramér-Rao Bounds
CTF	Coarse Time Filter
DBA	Data Bit Ambiguity
DBAR	Data Bit Ambiguity Resolution
DCB	Differential Code Bias
DD	Double-Differenced
DFT	Digital Fourier Transform

DGNSS Differential GNSS

DLL Delay Lock Loop

DOP Dilution Of Precision

EUSPA Europe Union Space Program Agency

EWL Extra Wide Lane

FARA Fast Ambiguity Resolution Approach

FASF Fast Ambiguity Search Filtering

FFT Fast Fourier Transform

FLL Frequency Lock Loop

GDOP Geometric Dilution Of Precision

GLONASS GLObal NAvigation Satellite System

GNSS Global Navigation Satellite System

GPS Global Positioning System

HCE Half Cycle Error

HI Hypothesis Index

HOW Hand Over Word

IAR Integer Ambiguity Resolution

IB Integer Bootstrapping

ICD Interface Control Documents

ICGC Institut Cartogràfic i Geològic de Catalunya

IF Intermediate Frequency

IFFT Inverse FFT

IGS	International GNSS Service
ILS	Integer Least Squares
IoT	Internet of Things
IR	Integer Rounding
LAMBDA	Least-squares AMBiguity Decorrelation Adjustment
LOS	Line Of Sight
LRF	LAMBDA Ratio Factor
LSE	Least Square Estimation
MCAR	Multiple Carrier Ambiguity Resolution
MCMF	Multi-Constellation Multi-Frequency
MH	Multi Hypothesis
NAVIC	Navigation with Indian Constellation
NCO	Numerically Controlled Oscillator
OSR	Observable Space Representation
PLL	Phase Lock Loop
PNT	Position Navigation or Timing
PPP	Precise Point Positioning
PRN	Pseudo-Random Noise
PVT	Position Velocity and Time
QZSS	Quasi-Zenith Satellite System

RF	Radio Frequency
RMS	Rooted Mean Square
RMSE	Rooted Mean Square Error
RTK	Real Time Kinematics
SATNAV	Satellite Navigation
SBAS	Satellite-based Augmentation System
SCI	Secondary Code Index
SD	Single-Differenced
SIS	Signal In Space
SPP	Standard Point Positioning
SRTK	Snapshot RTK
SSR	State Space Representation
TCAR	Three Carrier Ambiguity Resolution
TCXO	Temperature Compensated Crystal Oscillator
TGD	Total Group Delay
TOW	Time Of Week
TTFF	Time To First Fix
UPC	Universitat Politècnica de Catalunya
UPD	Uncalibrated Phase Delay
VRS	Virtual Reference Station
WL	Wide Lane

WLS Weighted Least Squares

Bibliography

[1] E. Kaplan and C. J.Hegarty. *Understanding GPS/GNSS: Principles and Applications, Third Edition*. 3rd. USA: Artech House, Inc., 2017. ISBN: 1630810584.

[2] J. Sanz Subirana, J.M. Juan Zornoza, and M. Hernández-Pajares. *GNSS DATA PROCESSING, Volume I: Fundamentals and Algorithms*. Vol. 1. ESTEC, PO Box 299, 2200 AG Noordwijk the Netherlands: ESA Communications, May 2013. ISBN: 978-92-9221-886-7.

[3] European GNSS Agency (GSA). *GNSS market report*. Luxembourg: Publications Office of the European Union, Oct. 2019. ISBN: 978-92-9206-043-5. DOI: 10.2878/031762. URL: https://www.gsa.europa.eu/system/files/reports/market_report_issue_6_v2.pdf.

[4] J. Morton et al. *Position, Navigation, and Timing Technologies in the 21st Century, Volumes 1 and 2: Integrated Satellite Navigation, Sensor Systems, and Civil Applications, Set*. John Wiley & Sons, 2020.

[5] *Global Navigation Satellite System (GNSS) Manual, First Edition*. English. International Civil Aviation Organization. 2005. URL: https://www.icao.int/Meetings/PBN-Symposium/Documents/9849_cons_en[1].pdf.

[6] K. Van Dierendonck and O. Al-Fanek. "What Is Snapshot Positioning and What Advantages Does It Offer?" In: *Inside GNSS* (Dec. 2018). URL: https://insidegnss.com/what-is-snapshot-positioning-and-what-advantages-does-it-offer/.

[7] G. López-Risueño and G. Seco-granados. "Measurement and Processing of Indoor GPS Signals Using One-Shot Software Receiver". In: *Proceedings of the IEEE Vehicular Technology Conference*. Springer, 2005.

[8] European GNSS Agency (GSA). *Power-efficient positioning for the Internet of Things*. Luxembourg: Publications Office of the European Union, June 2020. ISBN: 978-92-9206-048-0. DOI: 10.2878/437669. URL: https://www.gsa.europa.eu/sites/default/files/uploads/gsa_internet_of_things_white_paper.pdf.

[9] V. Lucas-Sabola et al. "Efficiency Analysis of Cloud GNSS Signal Processing for IoT Applications". In: *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2017)* (Mar. 2017). DOI: 10.33012/2017.15237.

[10] X. Liu et al. "RTK Feasibility Analysis for GNSS Snapshot Positioning". In: *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The*

Institute of Navigation (ION GNSS+ 2020) (Sept. 2020), pp. 2911 –2921. DOI: 10.33012/2020.17768.

[11] X. Liu et al. "Cloud-Based Single-Frequency Snapshot RTK Positioning". In: *Sensors* 21.11 (May 2021), p. 3688. ISSN: 1424-8220. DOI: 10.3390/s21113688. URL: http://dx.doi.org/10.3390/s21113688.

[12] D. Gebre-Egziabher and S. Gleason. *GNSS Applications and Methods*. GNSS technology and applications series. Artech House, 2009. ISBN: 9781596933309. URL: https://books.google.es/books?id=juXAE3SHQroC.

[13] ARINC Research Corp. *Navstar GPS Space Segment / Navigation User Interfaces (IS-GPS-200)*. 2021. URL: https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf.

[14] "SATNAV SIGNALS". In: *Engineering Satellite Based Navigation and Timing: Global Navigation Satellite Systems, Signals, and Receivers*. John Wiley & Sons, Ltd, 2015. Chap. 3, pp. 37–101. ISBN: 9781119141167. DOI: https://doi.org/10.1002/9781119141167.ch3. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119141167.ch3.

[15] European GNSS Agency (GSA). *GNSS User Technology Report*. Luxembourg: Publications Office of the European Union, Oct. 2020. ISBN: 978-92-9206-049-7. DOI: 10.2878/565013. URL: https://www.euspa.europa.eu/sites/default/files/uploads/technology_report_2020.pdf.

[16] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle. *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo and more*. Springer Vienna, 2008. ISBN: 9783211730126.

[17] N. Linty. "Snapshot Estimation Algorithms for GNSS Mass-Market Receivers". In: *PhD dissertation. Politecnico Di Torino, Torino, Italy* (Apr. 2015).

[18] K. Borre et al. "A Software-Defined GPS and Galileo Receiver". In: Birkhauser Boston, 2007. Chap. 5, pp. 69–73.

[19] D. Borio. "A Statistical Theory for GNSS Signal Acquisition". PhD thesis. Politecnico di Torino, Sept. 2014.

[20] M. Foucras. "Performance Analysis of Modernized GNSS Signal Acquisition". PhD thesis. Ecole Nationale d'Aviation Civile (ENAC), Feb. 2015.

[21] D. Jiménez-Baños et al. "Innovative Techniques for GPS Indoor Positioning Using a Snapshot Receiver". In: *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)* (2006), pp. 2944–2955. URL: https://www.ion.org/publications/abstract.cfm?articleID=6834.

[22] M.S. Braasch and A.J. van Dierendonck. "GPS receiver architectures and measurements". In: *Proceedings of the IEEE* 87.1 (1999), pp. 48–64. DOI: 10.1109/5.736341.

[23] B.W. Parkinson, J.J. Spilker, and P. Enge. *Global Positioning System, Vols I and II, Theory and Applications*. Reston, VA, USA: American Institute of Aeronautics, 1996.

[24] J.A. Klobuchar. "Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users". In: *Aerospace and Electronic Systems, IEEE Transactions on* AES-23.3 (1987), pp. 325–331. ISSN: 0018-9251. DOI: 10.1109/TAES.1987.310829. URL: http://dx.doi.org/10.1109/TAES.1987.310829.

[25] R. Prieto-Cerdeira et al. "The European Way: Assessment of NeQuick Ionospheric Model for Galileo Single-Frequency Users". In: *GPS World* 25.6 (2014), pp. 53–58. ISSN: 1048-5104. URL: http://gpsworld.com/innovation-the-european-way.

[26] D. Odijk. "Ionosphere-Free Phase Combinations for Modernized GPS". In: *Journal of Surveying Engineering* 129.4 (2003), pp. 165–173. DOI: 10.1061/(ASCE)0733-9453(2003)129:4(165).

[27] RTCA SC-159. *Radio Technical Commission for Aeronautics: Special Comittee No.159. Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment. RTCA Document 229-D*. Washington, USA, Dec. 2006.

[28] D. Prochniewicz, K. Wezka, and J. Kozuchowska. "Empirical Stochastic Model of Multi-GNSS Measurements". In: *Sensors* 21.13 (2021). ISSN: 1424-8220. DOI: 10.3390/s21134566. URL: https://www.mdpi.com/1424-8220/21/13/4566.

[29] Y. Wu and J. Guo. "Single point positioning with sequential least-squares filter and estimated real-time stochastic model". In: *Geo-spatial Information Science* 11.1 (2008), pp. 13–16. DOI: 10.1007/s11806-007-0162-5.

[30] R.E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45. URL: {https://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf}.

[31] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2011. ISBN: 9780970954428.

[32] P. J. G. Teunissen. "An analytical study of ambiguity decorrelation using dual frequency code and carrier phase". In: *Journal of Geodesy* 70 (1996), pp. 515–528.

[33] R. Hatch. "The synergism of GPS code and carrier measurements". In: *International Geodetic Symposium on Satellite Doppler Positioning*. Vol. 2. Jan. 1983, pp. 1213–1231.

[34] J. Jung, P. Enge, and B. Pervan. "Optimization of cascade ambiguity resolution with three civil GPS signals". In: *Proceedings of ION GPS 2000*. salt Lake City, Utah, USA, Sept. 2000, pp. 2191–2201.

[35] U. Vollath et al. "Analysis of the three-Carrier Phase ambiguity resolution (TCAR) for precise relative positioning in GNSS-2". In: *Proceedings of the Institute of Navigation GNSS+* (Sept. 1998), pp. 417–426.

[36] W. Werner and J. Winkel. "TCAR and MCAR Options with Galileo and GPS". In: *The Institute of Navigation* (2003). URL: https://www.ion.org/publications/abstract.cfm?articleID=5255.

[37] P. J. G. Teunissen, P. Joosten, and C.C.J.M. Tiberius. "A Comparison of TCAR, CIR and LAMBDA GNSS Ambiguity Resolution". In: *Proceedings of ION GPS 2002*. Portland, Oregon, USA, Sept. 2002, pp. 2799–2808.

[38] A. Hassibi and S. Boyd. "Integer parameter estimation in linear models with applications to GPS". In: *IEEE Transactions on Signal Processing* 46.11 (1998), pp. 2938–2952. DOI: 10.1109/78.726808.

[39] D. Chen. "Development of a Fast Ambiguity Search Filtering (FASF) Method for GPS Carrier Phase Ambiguity Resolution". In: *PhD dissertation, Department of Geomatics Engineering, University of Calgary* (1994).

[40] D. Chen and G. Lachapelle. "A Comparison of the FASF and Least- Squares Search Algorithms for on-the-Fly Ambiguity Resolution". In: *Annual of Navigation* 42 (1995), pp. 371–390.

[41] E. Frei and G. Beutler. "Rapid static positioning based on the fast ambiguity resolution approach FARA: theory and first results". In: *Manuscripta Geodaetica* 15 (Jan. 1990), pp. 325–356.

[42] U. Fernández-Plazaola et al. "The Null method applied to GNSS three-carrier phase ambiguity resolution". In: *Journal of Geodesy* 78 (Sept. 2004), pp. 96–102. DOI: 10.1007/s00190-004-0376-3.

[43] P. J. G. Teunissen. "The least-squares ambiguity decorrelation adjustment: A method for fast GPS integer ambiguity estimation". In: *Journal of Geodesy* 70 (Nov. 1995), pp. 65–82. DOI: 10.1007/BF00863419.

[44] P. J. G. Teunissen, P.J. De Jonge, and C.C.J.M. Tiberius. "Performance of LAMBDA Method for Fast GPS Ambiguity Resolution." In: *Navigation* 44.3 (1997), pp. 373–383.

[45] D. Psychas and S. Verhagen. *LAMBDA software package: Python implementation, Version 1.0*. Mathematical Geodesy and Positioning, Delft University of Technology, Nov. 2019.

[46] P. J. G. Teunissen. "GPS Carrier Phase Ambiguity Fixing Concepts". In: *GPS for Geodesy*. Ed. by P. J. G. Teunissen and Alfred Kleusberg. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 319–388. ISBN: 978-3-642-72011-6. DOI: 10.1007/978-3-642-72011-6_8. URL: https://doi.org/10.1007/978-3-642-72011-6_8.

[47] P. J. G. Teunissen, P. Joosten, and C.C.J.M. Tiberius. "The Volume of the GPS Ambiguity Search Space and its Relevance for Integer Ambiguity Resolution". In: *Proceedings of the 9th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1996)* (Sept. 1996), pp. 889 –898.

[48] P. J. G. Teunissen, P. Joosten, and C.C.J.M. Tiberius. "On the Spectrum of the GPS DD-Ambiguities". In: *Proceedings of the 7th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1994)* (Sept. 1994), pp. 115 –124.

[49] P.J.G. Teunissen and S. Verhagen. "GNSS Ambiguity Resolution: When and How to Fix or not to Fix?" In: *VI Hotine-Marussi Symposium on Theoretical and Computational Geodesy*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 143–148. ISBN: 978-3-540-74584-6.

[50] P. J. G. Teunissen and P.J. de Jonge. "Fast positioning using the LAMBDA-method". In: *Proceedings of the 4th International Symposium on Differential Satellite Navigation Systems (DSNS95)* (Apr. 1995), pp. 1 –8.

[51] S. Han. "Quality-control issues relating to instantaneous ambiguity resolution for real-time GPS kinematic positioning". In: *Journal of Geodesy* 71 (Jan. 1997), pp. 351–361. DOI: 10.1007/s001900050103.

[52] S. Verhagen and P. J. G. Teunissen. "The ratio test for future GNSS ambiguity resolution". In: *GPS solutions* 17.4 (2013), pp. 535–548.

[53] P. J. G. Teunissen and S. Verhagen. "GNSS Phase Ambiguity Validation: A Review". In: *Proceedings Space, Aeronautical and Navigational Electronics Symposium SANE2007, Japan* 107 (Jan. 2007), pp. 1–6.

[54] T. Takasu and A. Yasuda. "Kalman-Filter-Based Integer Ambiguity Resolution Strategy for Long-Baseline RTK with Ionosphere and Troposphere Estimation". In: *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)* (2010), pp. 161 –171. URL: https://www.ion.org/publications/abstract.cfm?articleID=9143.

[55] L. Wang and S. Verhagen. "A new ambiguity acceptance test threshold determination method with controllable failure rate." In: *Journal of Geodesy* 89 (Apr. 2015), 361–375. DOI: 10.1007/s00190-014-0780-2.

[56] A. Rovira Garcia et al. "Fast Precise Point Positioning: A System to Provide Corrections for Single and Multi-Frequency Navigation". In: *Navigation - Journal of The Institute of Navigation* 63 (Sept. 2016), pp. 231–247. DOI: 10.1002/navi.148.

[57] G. Wabbena, M. Schmitz, and A. Bagge. "PPP-RTK: Precise Point Positioning Using State-Space Representation in RTK Networks". In: *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)* (2005), pp. 2584 –2594.

[58] A. El-Mowafy. "Precise Real-Time Positioning Using Network RTK". In: *Global Navigation Satellite Systems*. Ed. by Shuanggen Jin. Rijeka: IntechOpen, 2012. Chap. 7. DOI: 10.5772/29502. URL: https://doi.org/10.5772/29502.

[59] D. Odijk, P. J. G. Teunissen, and A. Khodabandeh. "Single-frequency PPP-RTK: Theory and experimental results". In: vol. 139. Oct. 2014, pp. 571–578. ISBN: 978-3-642-37221-6. DOI: 10.1007/978-3-642-37222-3_75.

[60] O. Badia Solé and T.I. Ioan. "Enhancement study of GPS snapshot techniques". PhD thesis. Master Thesis, Aalborg University, Denmark, 2011. URL: https://projekter.aau.dk/projekter/files/43403136/MasterThesis.pdf.

[61] I. Fernández-Hernández and K. Borre. "Snapshot Positioning without Initial Information". In: *GPS Solut.* 20.4 (2016), 605–616. ISSN: 1080-5370. DOI: 10.1007/s10291-016-0530-4. URL: https://doi.org/10.1007/s10291-016-0530-4.

[62] K. Sun and L. Lo Presti. "Bit Sign Transition Cancellation Method for GNSS Signal Acquisition". In: *Journal of Navigation* 65.1 (2012), 73–97. DOI: 10.1017/S0373463311000543.

[63] C. Yang et al. "Comparative Study of Coherent, Non-Coherent, and Semi-Coherent Integration Schemes for GNSS Receivers (Preprint)". In: *Proceedings of the 63rd Annual Meeting of The Institute of Navigation (2007)* (Apr. 2007), pp. 572–588.

[64] T. Pany et al. "Coherent integration time: The longer, the better". In: *Inside GNSS* 4 (Nov. 2009), pp. 52–61.

[65] G. Beyerle et al. "A data archive of GPS navigation messages". In: *GPS Solutions* 13 (Sept. 2009), pp. 35–41. DOI: 10.1007/s10291-008-0095-y.

[66] G. Corazza et al. "Galileo Primary Code Acquisition Based on Multi-hypothesis Secondary Code Ambiguity Elimination". In: *Proceedings of the Institute of Navigation GNSS+* 2 (Jan. 2007), pp. 2459 –2465.

[67] F. Van Diggelen. *A-GPS: Assisted GPS, GNSS and SBAS*. Artech House, 2009. Chap. 4, pp. 61–102.

[68] K. Muthuraman, J. Brown, and M. Chansarkar. "Coarse Time Navigation: Equivalence of Algorithms and Reliability of Time Estimates". In: *Proceedings of the 2012 International Technical Meeting of The Institute of Navigation* (Jan. 2012), pp. 1115–1138.

[69] W. Yoo, L. Kim, and H. Lee. "A coarse-time positioning method for improved availability". In: *GPS Solutions* 24 (2019), p. 2. DOI: 10.1007/s10291-019-0919-y.

[70] F. Van Diggelen. *A-GPS: Assisted GPS, GNSS and SBAS*. Artech House, 2009. Chap. 8, pp. 239–273.

[71] J. Zhang et al. "GPS Satellite Velocity and Acceleration Determination using the Broadcast Ephemeris". In: *Journal of Navigation* 59.2 (2006), 293–305. DOI: 10.1017/S0373463306003638.

[72] *LabSat 3 Wideband. https://www.labsat.co.uk/index.php/en/products/labsat-3-wideband*. 2019.

[73] *Septentrio PolaNt-x MF*. https://www.septentrio.com/en/products/antennas/polant-x-mf. Accessed: 2021-04-15.

[74] D. Medina et al. "Compact CRB for Delay, Doppler and Phase Estimation - Application to GNSS SPP & RTK Performance Characterization". In: *IET Radar, Sonar & Navigation* 14.10 (2020), pp. 1537–1549.

[75] D. Dardari, E. Falletti, and M. Luise. *Satellite and terrestrial radio positioning techniques: a signal processing perspective*. Academic Press, 2011.

[76] L. Weill. "C/A Code Pseudoranging Accuracy - How Good Can It Get?" In: *Proceedings of the 7th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1994)*. Salt Lake City, UT, Sept. 1994, pp. 133–141. URL: https://www.ion.org/publications/abstract.cfm?articleID=3940.

[77] *Septentrio PolaRx5e*. https://www.septentrio.com/en/products/gnss-receivers/reference-receivers/polarx5e. Accessed: 2022-05-30.

[78] T. Takasu. "RTKLIB: Open Source Program Package for RTK-GPS". In: *Free and Open Source Software for Geospatial Conference*. Tokyo, Japan, Nov. 2009. URL: http://gpspp.sakura.ne.jp/paper2005/foss4g_2009_rtklib.pdf.

[79] J. Bao-Yen Tsui. "Fundamentals of Global Positioning System Receivers: A Software Approach, Second Edition". In: John Wiley & Sons, Inc., 2004. Chap. 6, pp. 117–118.