



END OF STUDIES PROJECT - FINAL REPORT

AEROSPACE SYSTEMS - NAVIGATION AND TELECOMMUNICATIONS

# PRECISE POSITIONING ON SMARTPHONES

Arnau Ochoa Bañuelos

COMPANY TUTOR: Arnaud Ginestet

UNIVERSITY TUTOR: Paul Thevenon

ÉCOLE NATIONALE DE L'AVIATION CIVILE

Toulouse, September 1, 2021



# Abstract

The capacity to obtain precise positioning using smartphones has become conceivable in recent years. With the release of Android 7 in 2016, GNSS raw measurements were made available to developers. Later, in 2018, the first smartphone with a dual-frequency chip was launched. Since then, the dual-frequency capability has become mainstream among middle to high-level devices, which usually also support at least four constellations. Furthermore, the Android GNSS API has made new information available with the later updates. For these reasons, and considering that smartphones have access to the Internet, it is plausible to aim at decimeter-level positioning accuracy on these devices. Nonetheless, smartphones also have some flaws when using them as GNSS receivers. Firstly, the GNSS antenna is small, and it is embedded inside the body of the smartphone. This causes the signals to be received at a low signal-to-noise ratio. Furthermore, the rest of electronics that are comprised inside the body of the device can cause small interference that further degrade the received signal. Secondly, smartphone manufacturers usually put low-cost GNSS chips in their smartphones, since the manufacturing cost of the device comprises many more components. This causes some parts of the chipset, such as the clock, to be lower quality when compared to the dedicated high-end GNSS receivers that are commonly used for high-accuracy positioning applications.

The objective of this project is to try to obtain the best accuracy from a set of GNSS measurements that are obtained with smartphones, aiming at the decimeter-level accuracy. This project is comprised within the Google Smartphone Decimeter Challenge, a competition organized by the Android's GNSS team. For this competition, Google released a pool of datasets containing GNSS and inertial measurements that were obtained with smartphones on different locations around the US Bay Area. The goal of this competition is to obtain the best possible accuracy from all the datasets, taking the mean of the 50% and the 95% horizontal accuracies among all traces. The results will be presented at ION GNSS+ 2021.

Our team for this competition is formed in collaboration by people from CS Group and ENAC. This report presents the solution that has been implemented in order to face this competition. One interesting characteristic of the data published by Google is that many traces contain measurements from more than one phone. For this reason, we decided to combine the measurements

from multiple receivers in a Real-Time Kinematic (RTK) implementation. This method allows to estimate position, velocity and attitude of the vehicle while at the same time reducing the noise of the estimation.

# Résumé

La capacité d’obtenir un positionnement précis avec des smartphones est devenue possible ces dernières années. Avec la sortie d’Android 7 en 2016, les mesures brutes GNSS ont été mises à la disposition des développeurs. Plus tard, en 2018, le premier smartphone doté d’un récepteur à double fréquence a été lancé. Depuis lors, la capacité à double fréquence est devenue courante parmi les appareils de niveau moyen à élevé, qui prennent généralement en charge au moins quatre constellations. De plus, l’API GNSS Android a rendu de nouvelles informations disponibles avec les dernières mises à jour. Pour ces raisons, et étant donné que les smartphones ont accès à Internet, il est plausible de viser une précision de positionnement au décimètre pour ces appareils. Néanmoins, les smartphones présentent également quelques défauts quand ils sont utilisés comme récepteurs GNSS. Premièrement, l’antenne GNSS est petite et elle est intégrée à l’intérieur du corps du smartphone. Cela provoque la réception des signaux à une faible puissance par rapport au bruit. De plus, le reste de l’électronique qui est compris à l’intérieur du corps de l’appareil peut provoquer de petites interférences qui dégradent davantage le signal reçu. Deuxièmement, les fabricants de smartphones intègrent généralement des récepteurs GNSS à bas prix dans leurs smartphones, car le coût de fabrication de l’appareil comprend beaucoup plus de composants. Cela entraîne une qualité inférieure de certaines parties de la puce, telles que l’horloge, par rapport aux récepteurs GNSS haut de gamme dédiés qui sont couramment utilisés pour les applications de positionnement de haute précision.

L’objectif de ce projet est d’essayer d’obtenir la meilleure précision à partir d’un ensemble de mesures GNSS obtenues avec des smartphones, visant la précision au niveau décimétrique. Ce projet s’inscrit dans le cadre du Google Smartphone Decimeter Challenge, un concours organisé par l’équipe GNSS d’Android. Pour ce concours, Google a publié un ensemble de données contenant des mesures GNSS et inertielles obtenues avec des smartphones à différents endroits de la baie de San Francisco, aux États-Unis. L’objectif de ce concours est d’obtenir la meilleure précision possible à partir de tous les données, en prenant la moyenne des précisions horizontales de 50% et 95% parmi toutes les traces. Les résultats seront présentés à ION GNSS+ 2021.

Notre équipe pour ce concours est formée en collaboration par des ingénieurs de CS Group et de l’ENAC. Ce rapport présente la solution qui a été mise en place pour faire face à cette concur-

rence. Une caractéristique intéressante des données publiées par Google est que de nombreuses traces contiennent des mesures provenant de plusieurs téléphones. Pour cette raison, nous avons décidé de combiner les mesures de plusieurs récepteurs dans une méthode Real-Time Kinematic (RTK). Cette méthode permet d'estimer la position, la vitesse et l'orientation du véhicule tout en réduisant le bruit de l'estimation.

# Acknowledgements

I would like to express my gratitude to Paul Thevenon and Jan Bolting for their patient guidance and useful recommendations on the development of this project. I also would like to thank Arnaud Ginestet for accepting me to participate on this internship, offering me the opportunity and all the facilities to work at CS Group, where I have felt like home. Finally, I am also very grateful to my family and friends for their personal support during the realization of this project.





# Contents

<b>Abstract</b>	<b>i</b>
<b>Résumé</b>	<b>iii</b>
<b>1 Subject of the internship</b>	<b>3</b>
<b>2 Introduction</b>	<b>5</b>
2.1 Objectives . . . . .	6
2.2 Work environment . . . . .	6
<b>3 Organization</b>	<b>7</b>
3.1 Internship environment . . . . .	7
3.2 Google Smartphone Decimeter Challenge . . . . .	8
<b>4 State of the art</b>	<b>11</b>
4.1 RTK positioning . . . . .	11
4.2 PPP . . . . .	12
4.3 GNSS/INS integration . . . . .	13
4.4 Multi-receiver RTK . . . . .	14
<b>5 Proposed solution</b>	<b>15</b>
<b>6 Mathematical Models</b>	<b>17</b>
6.1 GNSS observables . . . . .	17

6.2	SPPV EKF model . . . . .	20
6.2.1	State transition model . . . . .	20
6.2.2	Measurement model . . . . .	22
6.2.3	Prediction and update steps . . . . .	25
6.3	Sequential Kalman Filter . . . . .	26
6.4	Single-receiver RTK . . . . .	26
6.4.1	RTK Processing . . . . .	28
6.5	Multiple-receiver RTK . . . . .	32
6.5.1	Multiple-receiver geometry . . . . .	33
6.5.2	RTK processing . . . . .	34
<b>7</b>	<b>Measurement analysis</b>	<b>39</b>
7.1	Calculation of GNSS observables . . . . .	40
7.2	Comparison of GNSS measurement sources . . . . .	41
7.2.1	Reliability of the measurement uncertainties . . . . .	42
<b>8</b>	<b>Results</b>	<b>45</b>
8.1	Comparison between positioning methods . . . . .	45
8.2	Accuracy of multi-receiver RTK . . . . .	47
8.3	Use of carrier phase measurements . . . . .	47
8.4	Results of Google Smartphone Decimeter Challenge . . . . .	48
<b>9</b>	<b>Conclusion</b>	<b>51</b>
	<b>Acronyms</b>	<b>55</b>

# List of Figures

3.1	Example setup of phones on the vehicle dashboard [Fu20] . . . . .	9
3.2	Ground-truth system and lever arm example [Fu20] . . . . .	9
3.3	Timeline of the project. . . . .	10
6.1	Double differences as projections. $SV\ p$ and $SV\ q$ are the pivot and non-pivot satellites, and $m$ and $n$ are the station and the receiver [Kap06] . . . . .	30
6.2	Multi-receiver geometry . . . . .	33
6.3	Distribution of pseudorange rate innovations (blue) and LOS velocity error induced by lever arm (red). . . . .	37
7.1	Differences of the preprocessed measurements of a single satellite compared to the RINEX (top) and the GnssAnalysis (bottom). From left to right: pseudorange (meters), carrier phase (cycles) and Doppler (Hz). . . . .	42
7.2	Distributions of the pseudorange residuals obtained from the RINEX (left) and the preprocessed raw measurements (right). . . . .	42
7.3	Receiver clock bias estimated by the WLS (left) and difference of pseudorange between RINEX and preprocessed raw measurements (right). . . . .	43
7.4	Pseudorange residuals vs elevation-based standard deviation (left). Pseudorange residuals vs uncertainty-based standard deviation (right). . . . .	43
7.5	CDF of horizontal error from <i>magnitude</i> 's EKF using elevation-based variance (blue) vs uncertainty-based variance (red). . . . .	44
8.1	Estimated positions with SPP (blue), SPPV (red), single-receiver RTK (purple) and multi-receiver RTK (green). . . . .	46

8.2	Horizontal error CDFs of SPP (blue), SPPV (red), single-receiver RTK (purple) and multi-receiver RTK (green). . . . .	46
8.3	Comparison between single-receiver (blue, red, purple) and multi-receiver (green). . . . .	47
8.4	Horizontal error CDFs of <i>No-phase RTK</i> (blue), and RTK (red). . . . .	48
8.5	Scores obtained on all traces of the training set. . . . .	49
8.6	Final scores obtained with different techniques on the test dataset. . . . .	50

# List of Tables

3.1 Datasets of the Google Smartphone Decimeter Challenge . . . . . 9



# Chapter 1

## Subject of the internship

The subject of this internship is to participate in the Google Smartphone Decimeter Challenge. This competition, hosted by the Android GPS team, is being presented at the ION GNSS+ 2021 Conference. The objective of this competition is to develop a high precision positioning algorithm that uses GNSS measurements from Android smartphones. To participate in this competition, a team has been formed by CS Group and ENAC.





## Chapter 2

# Introduction

In May 2016, Google announced the availability of GNSS raw measurements on smartphones with the release of Android 7. From then, newer versions have been improving the availability of this data and have also included access to more information such as the Automatic Gain Control (AGC) or the code type of the signal. Newer phones have also added compatibility with other constellations apart from GPS with some devices supporting up to 5 different systems, namely GPS, GLONASS, Galileo, BeiDou and QZSS. Furthermore, thanks to the decreasing cost and the miniaturization of the hardware, some manufacturers included dual frequency chips on their higher-end products, the first being the Xiaomi Mi 8 which was launched on May 2018. During all these years, the interest on the Android GNSS raw measurements has been steadily increasing [Fu20].

This year 2021, Android’s GNSS team has launched the Google Smartphone Decimeter Challenge, which will be presented at ION GNSS+ 2021. For this competition, Google released a pool of 48 datasets containing inertial and GNSS raw measurements obtained from five different smartphones. These datasets were collected on highways and urban areas in the US San Francisco Bay Area. In addition, Satellite Space Representation (SSR) corrections and Observation Space Representation (OSR) corrections are provided from SwiftNavigation Inc.<sup>1</sup> and Verizon Inc.<sup>2</sup>. The goal of this competition is to develop an algorithm capable of obtaining the best possible accuracy on these datasets, setting decimeter-level positioning as an accuracy target. In addition, another pool of 73 datasets is provided which follow the same structure but also contain ground-truth files collected by a high-precision NovAtel system.

Next, the objectives of this internship are detailed and the work environment is described. The organization of this internship is depicted in Chapter 3. Then, Chapter 4 provides a review of the state of the art relevant to this project. Chapter 5 describes the solution that our team proposed

---

<sup>1</sup>[www.swiftnav.com/google-smartphone-decimeter-challenge](http://www.swiftnav.com/google-smartphone-decimeter-challenge)

<sup>2</sup>[webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=88&guccounter=1](http://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=88&guccounter=1)

in order to face this competition. The mathematical models of the methods implemented are described in Chapter 6. An analysis of the Android GNSS measurements is given in Chapter 7 and results are provided in Chapter 8. Finally, conclusions are outlined in Chapter 9.

## 2.1 Objectives

The main deliverables for this project are the following:

- **Extended abstract:** An extended abstract was sent on March 5 in order to opt for the presentation of the final paper at the conference. This abstract included a description of the proposed solution and some preliminary results.
- **Full results:** The results that are obtained with the test dataset must be uploaded before August 4.
- **Final paper:** Since our team was selected from the extended abstract, a paper will be presented at the ION GNSS+ 2021 conference. Which starts on September 20.

## 2.2 Work environment

My internship at CS Group has been done within the department of Flight Dynamics & Science, which is included in the Space Business Unit in the company. This department addresses all the matters that concern the chain of a satellite. From the mission to the control, going through the spatial mechanics, the simulations and including the activities of support to the qualification and operations. All these activities are categorized in: Space Mechanics, Control Center and Operations, Mission and Simulations, Navigation, and Products and R&D.

The people that provided support on the realization of this project are:

- **Arnaud Ginestet (CS Group):** Company tutor, expertise in GNSS.
- **Bryan Cazabonne (CS Group):** Experience in positioning on Android smartphones.
- **Matthieu Pascaud (CS Group):** Expertise in Android.
- **Paul Thevenon (ENAC):** University tutor, expertise in GNSS.
- **Jan Bolting (ENAC):** Expertise in GNSS.
- **Xiao Hu (ENAC):** Expertise in GNSS. Experience on multi-receiver RTK.

## Chapter 3

# Organization

### 3.1 Internship environment

As it has been stated in the introduction, this internship has been performed in a collaboration between CS Group and ENAC. CS Group is an information-technology service company which creates products for a wide variety of fields. One of these fields is the aerospace, developing software for the Attitude Control System of satellites or open source tools such as Orekit, a low level space dynamics library written in Java.

This internship at CS Group has been done within the department of Flight Dynamics & Science, which is included in the Space Business Unit in the company. This department addresses all the matters that concern the chain of a satellite. From the mission to the control, going through the spatial mechanics, the simulations and including the activities of support to the qualification and operations. All these activities are categorized in: Space Mechanics, Control Center and Operations, Mission and Simulations, Navigation, and Products and R&D.

CS Group also has long experience with the Android development. For this reason, the collaboration between CS Group and ENAC is born with the objective of participating on the Google Smartphone Decimeter Challenge.

The people that will provide support on the realization of this project are:

- **Arnaud Ginestet (CS Group):** Company tutor, expertise in GNSS.
- **Bryan Cazabonne (CS Group):** Expertise in Orekit.
- **Matthieu Pascaud (CS Group):** Expertise in Android.
- **Paul Thevenon (ENAC):** University tutor, expertise in GNSS.
- **Jan Bolting (ENAC):** Expertise in GNSS.

## 3.2 Google Smartphone Decimeter Challenge

The Google Smartphone Decimeter Challenge is a competition organized by the Navigation team at Android which aims at promoting the research on high accuracy positioning techniques from data that can be obtained from Android smartphones. For this competition, the organizers published a pool of datasets obtained from different smartphones in various measurement campaigns in the US San Francisco Bay Area. These datasets contain GNSS raw measurements as well as inertial measurements, including accelerometer, gyroscope and magnetometer measurements. Additionally, Google collaborated with SwiftNav Inc. and Verizon Inc. to provide Satellite Space Representation (SSR) and Observation Space Representation (OSR) correction data, which allow to implement Precise Point Positioning (PPP) and Real-Time Kinematic (RTK), respectively.

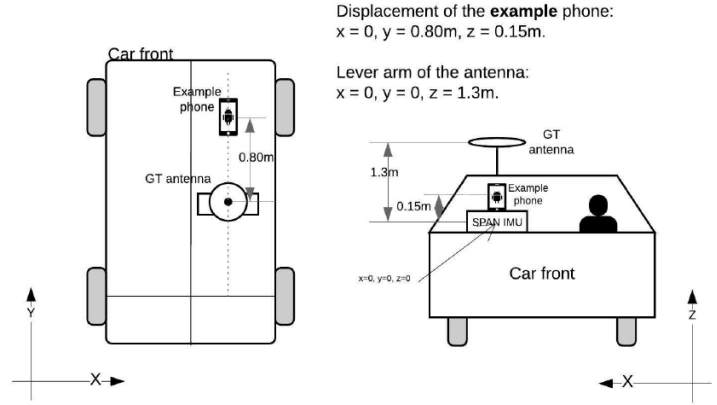
Initially, on December 23 2020, Google published a dataset containing measurements taken in 17 different campaigns between May and September 2020. These campaigns contained one or multiple traces from different smartphones. In total this dataset contained 39 different traces from 4 different smartphones. This dataset was provided as an example of the format of the data that would be later provided for the competition. This data contained also NMEA files with the ground-truth positions of the smartphones, which were obtained with a NovAtel SPAN system and a Waypoint Inertial Explorer via tightly coupling of IMU and GNSS measurements. The position provided by this reference system is then shifted by the relative position of the smartphones inside the car, so one ground-truth NMEA file is provided for each trace. An example setup of phones during a measurement campaign is shown in Fig. 3.1 and a diagram representing the ground-truth system in the vehicle can be seen in Fig. 3.2

In May 12 2021, when the competition started, the definitive data was published. This data is divided in two datasets: the *train* and the *test*. The *train* dataset contains 73 traces from 29 campaigns for which the ground-truth positions are provided. This dataset was provided so that the teams could test their algorithms and verify the results with the reference positions. The *test* dataset contains 48 traces from 19 campaigns for which no ground-truth was provided. Table 3.1 summarizes the details of these two datasets.

The *test* dataset is the one to be used to evaluate the results in the competition. The participating teams have to provide the estimated positions for these 48 traces in a *csv* file containing the names of the campaign and of the phone, the timestamp in milliseconds and the latitude and longitude of the estimated positions. The number of entries and the timestamps must match with an example submission file that is provided by the organizers of the competition. Then, the score of each team is computed by taking the mean between the 50 and the 95 percentiles of the horizontal error of each trace and then averaging this mean for all traces. This is



**Figure 3.1:** Example setup of phones on the vehicle dashboard [Fu20]



**Figure 3.2:** Ground-truth system and lever arm example [Fu20]

Dataset	Train	Test
Campaigns	29	19
Traces	73	48
Phones	Pixel4, Pixel4Modded, Pixel4XL, Pixel4XLModded, Pixel5, Mi8, SamsungS20Ultra	Pixel4, Pixel4Modded, Pixel4XL, Pixel4XLModded, Pixel5, Mi8, SamsungS20Ultra
Ground-truth	Yes	No

**Table 3.1:** Datasets of the Google Smartphone Decimeter Challenge

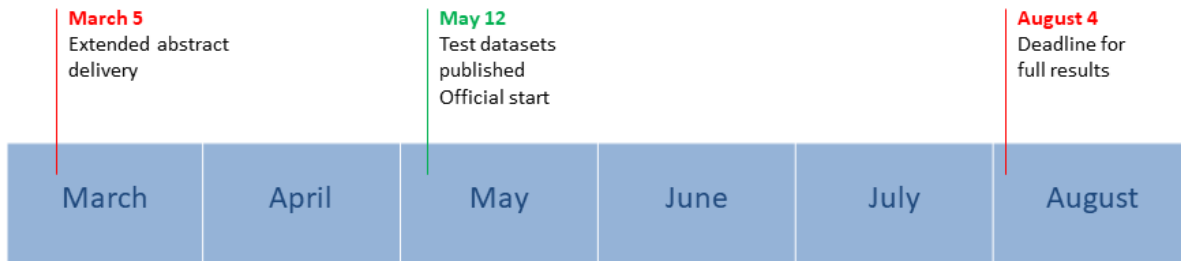
$$s = \text{mean}\{s_1, s_2, \dots, s_{48}\} \quad | \quad s_i = \text{mean}\{P_{50,t}, P_{95,t}\} \quad (3.1)$$

where  $s$  is the final score in meters, and  $P_{50,t}, P_{95,t}$  are the 50 and 95 percentiles of the horizontal error for the trace  $t$ .

The main milestones of this project are shown in the timeline of Fig. 3.3. The first milestone was on March 5, when our team submitted the extended abstract to opt for the option to be speakers at the ION GNSS+ 2021 conference. The acceptance of the speakers was based on the expected accuracy of the results and the theoretical innovation.

On April 14, our abstract was accepted among 20 other teams to participate on the ION conference. Later, on May 12, the competition was officially started with the release of all the

datasets, which were published on a dedicated page on the Kaggle website <sup>1</sup>. From that day, the teams were allowed to upload the result files on the same website. From these results two scores are computed. The first score uses 53% of the estimated positions to generate a public leaderboard showing the progress of all the participating teams. The other 47% of the data is used to compute a private score that is the one actually used to select the winners of the competition. Finally, on August 4, the competition was closed and the leaderboard with the private scores was published.



**Figure 3.3:** Timeline of the project.

---

<sup>1</sup><https://www.kaggle.com/c/google-smartphone-decimeter-challenge>

# Chapter 4

## State of the art

Different techniques have been long used to achieve high accuracy in the position computation. In order to achieve decimeter-level accuracy, the use of carrier phase measurements and the capability to obtain integer solutions for the ambiguities are two features that are required [Kap06]. The two solutions that are classically applied to solve this problem are RTK positioning and PPP. These techniques can also be enhanced with the integration of Inertial Navigation System (INS) measurements, which can provide improvements of accuracy, availability, continuity and integrity. Furthermore, other advanced techniques are being proposed such as the use of multiple receivers. In addition, other methods exist that can improve the accuracy when post-processing the measurements. One of the most well-known techniques is the Forward-Backward Smoothing. Next, an overview of these techniques is provided

### 4.1 RTK positioning

RTK is a technique that relies on differential GNSS measurements in order to resolve the carrier-phase ambiguities. This technique is based on the use of the carrier-phase measurements that are obtained in a station whose location is well known. These measurements, called Observation Space Representation (OSR), are then retrieved by the user receiver, and are used to compute double-differences with the GNSS measurements that are obtained locally. This method allows to eliminate the main sources of error in GNSS. From a high-level point of view, the RTK algorithm is based of the following principles:

- System-induced errors, such as satellite clock bias, are constant among different receivers. Thus, they can be eliminated by differentiation of two observations of the same satellite obtained from different receivers.
- Receiver-induced errors, such as receiver clock bias, are constant among different observa-

tions on the same receiver. Therefore, these errors can be eliminated by differentiation of two observations of different satellites obtained from the same receiver.

- Atmospheric-induced errors are approximately equal in the vicinity of a receiver. Therefore, they can be drastically reduced by differentiation of two observations of the same satellite obtained from different receivers.
- The noise of carrier measurements is much smaller than the one of the pseudorange measurements. However, the processing of carrier measurements is subject to the so-called carrier phase ambiguity, an unknown integer number of times the carrier wave length, that needs to be fixed in order to rebuild full range measurements from carrier ones [GMV11].
- The carrier phase ambiguities can be estimated using differential measurements by different techniques. Usually, the estimation of these ambiguities is performed using standard least-squares or Kalman filtering. The solution obtained is referred as the *float solution*. The resulting ambiguities, though, are highly correlated and have a poor precision. This problem can be solved with integer ambiguity resolution methods, for which the least-squares ambiguity decorrelation adjustment (LAMBDA) is usually considered the standard method [Teu17].

This method has also its limitations. RTK can only be used for distance of up to 10 or 20 kilometers from the base station [Teu17]. This limitation is caused by distance-dependent biases, mainly ionospheric refraction. Additionally, some convergence time is needed to estimate the phase ambiguities. This time depends on the processing algorithm and the distance between the base and the receiver. Moreover, a high-bandwidth communication channel is needed for real time applications, as the OSR data tends to be heavy.

## 4.2 PPP

PPP provides an attractive alternative to differential Global Navigation Satellite System (GNSS) techniques such as RTK. The PPP method uses un-differenced, multi-frequency, pseudorange and carrier-phase observations along with precise satellite orbit and clock products, usually called Satellite Space Representation (SSR). This technique can provide centimeter-level precision on standalone static or kinematic positioning. From a high-level point of view, the PPP algorithm is based of the following principles:

- System induced errors, such as orbit or satellite clock bias, are precisely estimated by a network of GNSS reference stations distributed worldwide. These precise satellite products are then broadcasted by entities such as the International GNSS Service (IGS). These



products can then be used by a receiver instead of the ones broadcasted in the navigation messages, which are less precise.

- The ionospheric delay is the major source of error due to the atmosphere, inducing up to 30 m delays. Since the ionosphere is a dispersive medium, the first-order and second-order terms of the ionospheric delay can be eliminated by the ionosphere-free combination of measurements observed at two distinct signal frequencies. The third-order term is negligible at the sub-mm level for GNSS frequencies [Teu17].
- Various models exist for estimating the tropospheric delay. The typical model that is used in most GNSS applications is the *Saastamonien* [Saa72]. However more precise models exist, which many times require external information input.
- PPP requires the introduction of additional initial phase ambiguity unknowns, which causes a fairly long initial convergence of PPP solutions that can be longer than 15 minutes [Teu17]. As with RTK, phase ambiguities are usually estimated as *float solutions* and, then, are fixed to integers.
- Additional error sources, such as antenna phase center offset or tidal effects, can also be modeled in a PPP system. These will improve even more the accuracy but require a higher complexity of the algorithm.

PPP provides some benefits with respect to the differential techniques such as RTK, mainly due to the fact that no reference stations are needed. This eliminates the restriction of the positioning to be applied inside an area delimited by the distance to the closest station. Furthermore, although some communication channel is required in order to retrieve the precise satellite orbit and clock products, less bandwidth is required in order to transmit SSR data compared to OSR data. However, as it has been previously stated, PPP requires a long time of convergence in order to obtain a float estimate of the carrier-phase ambiguities. In addition, this method also requires a cautious modeling and data screening for outliers and carrier-phase cycle slips, which is more challenging than for the DD approach [Teu17].

### 4.3 GNSS/INS integration

In navigation, multi-sensor data fusion consists in combining measurements from different sensors in order to obtain a more robust and accurate navigation solution. [Teu17]. The most typical sensors that are used are the Inertial Navigation System (INS), which are usually comprised by accelerometers and gyroscopes, but other sensors can also be used such as altimeters or magnetometers. When integrating INS and GNSS, the inertial measurements provide continuous navigation with low short-term noise. Nonetheless, the accuracy of an inertial navigation

solution decreases over time as the system integrates errors repeatedly [Gro08]. The accuracy degradation is even more significant when employing low-cost equipment for long periods of time. On the other hand, GNSS provide a navigation solution with high short-term noise and high long-term accuracy. Nevertheless, their output rate is much lower compared to the INS and performance and availability strongly depend on the signal propagation conditions.

For these reasons, the integration of inertial sensors and GNSS results is a very profitable symbiosis as it not only overcomes the difficulties that each sensor presents individually. GNSS measurements correct the drifted inertial solution, while INS smooths the noisy GNSS solution and provides complete navigation solution during signal outages. Therefore, the INS/GNSS integration results in an improved integrated system, which provides continuous navigation with high short- and long-term accuracy [Gro08]. The integration of the sensors is typically done using a Kalman filter. This estimation technique can provide optimal estimates of the instantaneous state of a linear system perturbed by Gaussian white noise, and update them in near real time or through post-processing [Kap06].

## 4.4 Multi-receiver RTK

In the recent years, various studies have researched the use of an array of GNSS receivers for attitude determination. Henkel and Günther proposed in [Hen13] a technique to estimate the attitude using two low-cost GPS receivers and INS measurements. Following this work, Henkel and Iafrancesco also proposed in [Hen14] an algorithm that estimates position and attitude by combining RTK from two receivers with INS measurements. Aghili and Salerno also worked on 3D attitude determination and position estimation of mobile robots using two RTK GPS receivers and IMU [Agh09] [Agh10]. Medina et al. also propose in [Med20] a method to recursively estimate the position and attitude using GNSS carrier phase observations from an array of receivers. Nonetheless, these techniques did not take into account the known geometry of the receivers to help improve the positioning performance.

Hu et al. propose in [Hu20b] [Hu21] [Hu20a] to consider the known geometry of multiple GNSS receivers as a constraint to improve the performance of an RTK as well as to estimate the attitude. This technique shows to improve the fixed rate, positioning accuracy and solution availability.

## Chapter 5

# Proposed solution

The Google Smartphone Decimeter Challenge datasets have a particularity that our team wants to exploit: it involves several smartphones which are located at a fixed location onboard the vehicle during one data collect. As such, the multiple sensors can be considered as an array of sensors, whose observations can be processed in a centralized manner. For example, by including the geometric relation between the different GNSS antennas of the multiple GNSS receivers, it is possible to establish an observation model that will include the attitude of the array in addition to its position.

These additional measurements add more redundancy for the state estimation process, and also increase robustness to signal blockage, multipath and cycle slips, thanks to some kind of spatial diversity: because the antennas are separated spatially, they may not be affected by the same propagation events. This spatial diversity gain increases with the spatial separation of the antennas. Due to the small separation in the Google data collect (on the order of tenths of cm), the effectiveness of this feature is not known in advance. However, it is expected that the multi-receiver solution can only be better than a single receiver solution.

This technique has been investigated previously by Xiao Hu at ENAC [Hu21] [Hu20a] [Hu20b] in the context of RTK in an extended Kalman filter. It has proved to provide increased performance, in terms of cycle slip detection, solution availability, fixed ambiguity solution availability and accuracy, providing some attitude information as well. The initial planning considered also the hybridization with INS, possibly combining the inertial measurements from multiple smartphones. Also we considered possibility to enhance the algorithm with additional post-processing techniques such as forward-backward smoothing. However, the characteristics of the smartphone measurements have required us to spend more time in the initial phases of the development so it has not been possible to include the GNSS/INS hybridization nor the forward-backward smoothing.

The development of this solution has been divided in various steps so that the complexity of the algorithm increases iteratively and the implementation can be performed in a more controllable way. These steps are the following:

1. **Review the existing code and tools:** Get familiar with the code that is already implemented from the research performed by Xiao Hu.
2. **Read and pre-process measurements:** The measurements provided by Google which come from the smartphones have to be read from the provided files, which have a specific format. Also, these measurements are very raw and have to be pre-processed in order to obtain the typical GNSS measurements (e.g. code pseudorange).
3. **Assess performance with basic positioning methods:** The GNSS measurements are evaluated using an already existing software called *magnitude*, which was developed by Prof. Paul Thevenon. The two techniques that are used with this software are a Weighted Least Squares (WLS) and a Kalman Filter (KF), both using only code-pseudorange observations to obtain Single-Point-Position estimates.
4. **Develop SPP solution:** Develop a Single-Point-Position (SPP) single-receiver solution. This solution combines GPS and Galileo dual-frequency and BeiDou single-frequency code pseudorange measurements.
5. **Develop SPPV solution:** Enhance the SPP solution by including Doppler measurements to obtain a Single-Point-Position-and-Velocity (SPPV) single-receiver solution. With this method, which is implemented using a Kalman Filter, the velocity estimation is used to predict the position in the next epoch, which smooths the noisiness introduced by the code pseudoranges.
6. **Include differential corrections:** Transform the existing implementation to perform differential corrections and obtain a RTK implementation. This method considers the existing code pseudorange and Doppler measurements as well as the carrier phase measurements with float ambiguity solution.
7. **Combine multiple receivers:** Transform the existing implementation to combine the measurements from multiple receivers.

## Chapter 6

# Mathematical Models

As explained in Chapter 5, the first two positioning methods that were implemented on this project are a SPP and a SPPV, both based on an Extended Kalman Filter (EKF). In this document, the model for the standalone SPP algorithm is not detailed as it is itself included in the SPPV model and it is also explained in the reference books [Kap06] [Teu17]. This chapter begins with the definition of the GNSS observables that are used in this project. Then, the description of the SPPV EKF is provided, which is also used to explain the behavior of the Extended Kalman Filter. A description of the sequential implementation of the EKF is provided next. Finally, the models for the single- and multi-receiver RTKs are provided.

### 6.1 GNSS observables

For the execution of this project, up to three different types of GNSS observables are used. These are the *code pseudorange*, *carrier phase* and *Doppler*. The code pseudorange is a measure of the apparent transit time of the signal from a satellite to the receiver [Mis06]. This is a biased measurement due to the fact that the satellite and the receiver clocks are not synchronized. Furthermore, the code pseudorange is also subject to other sources of error which are, mainly, the atmosphere (troposphere and ionosphere), instrumental delays (both in the satellite and in the receiver), relativistic effects, multipath and receiver noise. Thus, a code pseudorange observation can be modeled as

$$\rho_i = r_i + c(\delta t_u - \delta t_i) + I_i + T_i + \varepsilon_\rho \quad (6.1)$$

where  $\rho_i$  is the pseudorange observation from satellite  $i$ .  $r_i$  denotes the geometrical range between the receiver and the satellite  $i$ .  $c$  is the speed of light while  $\delta t_u$  is the bias of the receiver clock and  $\delta t_i$  is the bias of the satellite clock.  $I_i$  and  $T_i$  refer to the ionospheric and tropospheric

delays, respectively.  $\varepsilon_\rho$  denotes the pseudorange other errors, including multipath and receiver noise. Regarding the receiver clock delay,  $c\delta t_u$ , it is constant among all the measurements and, thus, it is added as an additional variable to estimate.

In general, some of the sources of error can be eliminated or mitigated. The satellite clock bias is estimated by the ground segment and is then broadcasted by the satellites in the navigation message so that the receiver can correct it. For the atmospheric effects, various models exist that allow to estimate the delays that are induced to the signal. Tropospheric delays are typically estimated with high accuracy using the Saastamoinen model [Saa72], which provides an estimate of the delay if the satellite is at the zenith. From that, the actual delay is estimated using a mapping function [Teu17]. The ionospheric delay can also be estimated using a model. Ionospheric models, though, require information of the state of the ionosphere, which is broadcasted on the navigation message. Different constellations offer distinct models, for example GPS uses the Klobuchar model [Klo87] while Galileo uses a variation of the NeQuick model [Teu17]. Alternatively, a receiver can implement a *iono-free* combination to mitigate the ionospheric delay. This method takes advantage of the fact that the ionosphere is a dispersive medium and, thus, the delay it induces depends on the frequency. With this technique, a linear combination of measurements at both frequencies is used to mitigate the first-order ionospheric error [Kap06].

The carrier phase is a much more precise measurement than that of the code pseudorange, for this reason it is almost essential for precise positioning applications. The carrier phase measurement is the difference between the phases of the receiver-generated carrier signal and the carrier received from a satellite at the instant of the measurement [Mis06]. To measure the carrier phase, a part of the receiver called Phase Lock Loop (PLL) locks with the satellite signal and measures the initial fractional phase difference between the received and the generated signals. From then on, the PLL tracks changes in this measurement, counting full cycles and keeping track of the fractional cycle at each measurement epoch. The receiver, though, can only keep track on the number of cycles since the signal is acquired. Therefore, there is an ambiguity in the actual number of cycles between the satellite and the receiver. This number of cycles has to be estimated during the positioning process. Usually, a float estimation of the ambiguity is estimated and, then, this estimation can be resolved into an integer solution in a process that is referred to as *integer ambiguity resolution*. Additionally, the carrier phase suffers from similar error sources as the code pseudorange. Thus, the model for the carrier phase measurement can be expressed as

$$\phi_i = r_i + c(\delta t_u - \delta t_i) - I_i + T_i + \lambda N_i + \varepsilon_\phi \quad (6.2)$$

where  $\phi_i$  is the pseudorange observation from satellite  $i$ .  $r_i$  denotes the geometrical range between the receiver and the satellite  $i$ .  $c$  is the speed of light while  $\delta t_u$  is the bias of the

receiver clock and  $\delta t_i$  is the bias of the satellite clock.  $I_i$  and  $T_i$  refer to the ionospheric and tropospheric delays, respectively. It must be noted that the error induced by the ionosphere affects to the phase measurement with the opposite sign [Kap06].  $N_i$  denotes the carrier phase ambiguity and  $\lambda$  is the wavelength of the carrier signal. Finally,  $\varepsilon_\phi$  refers to the other carrier phase errors, including multipath and receiver noise.

In (6.1) and (6.2) both clock biases are referred with respect to the system time. When combining observations from multiple constellations, one of the constellation is selected to set the reference time in the receiver. Therefore, an additional delay has to be accounted between the reference system time and the other constellations. Similarly, when using measurements from multiple frequencies an additional inter-frequency bias has to be considered. This bias is introduced by the different hardware delays that are introduced to different frequencies [Teu17]. Thus, the models for the pseudorange and the carrier phase (in meters) observations become

$$\rho_i = r_i + c(\delta t_u - \delta t_i + \delta t_{IS} + \delta t_{IF}) + I_i + T_i + \varepsilon_\rho \quad (6.3)$$

$$\phi_i = r_i + c(\delta t_u - \delta t_i + \delta t_{IS} + \delta t_{IF}) - I_i + T_i + \lambda N_i + \varepsilon_\phi \quad (6.4)$$

where  $\delta t_{IS}$  refers to the Inter-System bias, i.e. the bias between the time of the constellation  $S$  and the reference time; and  $\delta t_{IF}$  refers to the Inter-Frequency bias.

Due to the relative motion between the satellite and the receiver, a Doppler shift in the frequency of the signal carrier is observed at reception. This Doppler shift is measured either by time derivation of the carrier phase estimated by the PLL or directly by a Frequency Lock Loop (FLL). The atmospheric delays and hardware delays do not significantly contribute to the Doppler, as they vary slowly with time [Teu17]. The Doppler shift is directly related to the Line-Of-Sight (LOS) velocity between satellite and receiver as

$$\Delta f = f_c - f_L = \frac{\dot{r}_i}{c} f_L \quad (6.5)$$

where  $f_c$  and  $f_L$  are the frequency of the received signal and the frequency of the locally generated replica, respectively. The LOS velocity, or range rate, is denoted by  $\dot{r}_i$ . Then, for convenience, the Doppler observation can be expressed in terms of the pseudorange rate,  $\dot{\rho}_i$  as

$$\dot{\rho}_i = \dot{r}_i + c(\dot{\delta t}_u + \dot{\delta t}_{IF} + \dot{\delta t}_{IS} - \dot{\delta t}_i) \quad (6.6)$$

where  $\dot{\delta t}_u$ ,  $\dot{\delta t}_i$ ,  $\dot{\delta t}_{IF}$  and  $\dot{\delta t}_{IS}$  denote the clock drifts of the receiver and of the satellite  $i$ , the Inter-Frequency clock drift and the Inter-System clock drift, respectively. The range rate is defined

as the subtraction between the satellite velocity,  $\dot{\mathbf{s}}_i$ , and the receiver velocity,  $\dot{\mathbf{u}}$ , projected onto the  $\mathbf{e}_i$  vector from the receiver to the satellite,  $\mathbf{e}_i$ . This is

$$\dot{r}_i = (\dot{\mathbf{s}}_i - \dot{\mathbf{u}})^T \mathbf{e}_i \quad (6.7)$$

where  $\mathbf{e}_i$  is obtained from the satellite position,  $\mathbf{s}_i$ , and the receiver position  $\mathbf{u}$  as

$$\mathbf{e}_i = \frac{\mathbf{s}_i - \mathbf{u}}{r_i} \quad (6.8)$$

## 6.2 SPPV EKF model

The Kalman Filter is an iterative algorithm that estimates the state of a system by alternating two different steps. In the first step, a model that describes the evolution of the system is applied to predict the state of the system from one epoch to the next one. In the second step, a series of measurements that are related with the system state are used to update the prediction obtained in the first step. These two steps are named *prediction* and *update*, respectively. This algorithm assumes that the functions that are used in the prediction and in the update are linear. In many cases, though, these functions are non-linear and hence a variation of this algorithm is used, called Extended Kalman Filter. In this section, the model for the SPPV method is detailed, which is implemented with an EKF.

### 6.2.1 State transition model

The state transition model describes how the states or parameters of the system vary over time based on a specific model which, in this case, is linear. The state of the system is described by the state vector, which contains the unknowns of the system to be estimated. For the case of a multi-constellation SPPV model, the state vector at a given epoch  $n$  is given by

$$\mathbf{x}_n = \begin{bmatrix} \mathbf{u}^T & \dot{\mathbf{u}}^T & c\delta t_u & c\dot{\delta t}_u & c\delta \mathbf{t}_{IF}^T & c\dot{\delta \mathbf{t}}_{IF}^T & c\delta \mathbf{t}_{IS}^T & c\dot{\delta \mathbf{t}}_{IS}^T \end{bmatrix}^T \quad (6.9)$$

where  $\delta \mathbf{t}_{IF}$  is a  $N_{freq} - 1$  column vector that contains the Inter-Frequency clock biases and  $\delta \mathbf{t}_{IS}$  is a  $N_{sys} - 1$  column vector that contains the Inter-System clock biases. The state transition between two consecutive epochs is given by the following equation

$$\mathbf{x}_n = \mathbf{F}_{n-1} \mathbf{x}_{n-1} + \mathbf{w}_n \quad (6.10)$$



where  $\mathbf{x}_n$  is the state vector at epoch  $n$ ,  $\mathbf{F}_{n-1}$  refers to the state transition matrix of epoch  $n-1$  and  $\mathbf{w}_k$  denotes the process noise vector at epoch  $n$ . This process noise vector has its associated process noise covariance matrix, defined by

$$\mathbf{Q}_n = \mathbb{E}[\mathbf{w}_n \mathbf{w}_n^T] \quad (6.11)$$

The state transition matrix  $\mathbf{F}_n$  is defined from the specific state transition models of each state parameter. In the case of the SPPV method, the evolution of the receiver position is directly obtained by integration of the estimated velocity, which in discrete time is given by

$$\mathbf{u}_n = \mathbf{u}_{n-1} + \dot{\mathbf{u}}_{n-1} \Delta t \quad (6.12)$$

where  $\Delta t$  denotes the time span between two consecutive epochs. The remaining state parameters are modeled as a 1st-order Markov process. This results in

$$\begin{aligned} \dot{\mathbf{u}}_n &= \dot{\mathbf{u}}_{n-1} + \mathbf{w}_{\dot{\mathbf{u}},n} \\ c\delta t_{u,n} &= c\delta t_{u,n-1} + \mathbf{w}_{c\delta t_u,n} \\ c\dot{\delta t}_{u,n} &= c\dot{\delta t}_{u,n-1} + \mathbf{w}_{c\dot{\delta t}_u,n} \\ c\delta t_{IF,n} &= c\delta t_{IF,n-1} + \mathbf{w}_{c\delta t_{IF},n} \\ c\dot{\delta t}_{IF,n} &= c\dot{\delta t}_{IF,n-1} + \mathbf{w}_{c\dot{\delta t}_{IF},n} \\ c\delta t_{IS,n} &= c\delta t_{IS,n-1} + \mathbf{w}_{c\delta t_{IS},n} \\ c\dot{\delta t}_{IS,n} &= c\dot{\delta t}_{IS,n-1} + \mathbf{w}_{c\dot{\delta t}_{IS},n} \end{aligned} \quad (6.13)$$

where

$$\begin{aligned} \mathbf{w}_{\dot{\mathbf{u}},n} &\sim \mathcal{N}(0, \sigma_u^2 \mathbf{I}_3) \\ \mathbf{w}_{c\delta t_u,n} &\sim \mathcal{N}(0, \sigma_{c\delta t_u}^2) \\ \mathbf{w}_{c\dot{\delta t}_u,n} &\sim \mathcal{N}(0, \sigma_{c\dot{\delta t}_u}^2) \\ \mathbf{w}_{c\delta t_{IF},n} &\sim \mathcal{N}(0, \sigma_{c\delta t_{IF}}^2 \mathbf{I}_{N_{freq}-1}) \\ \mathbf{w}_{c\dot{\delta t}_{IF},n} &\sim \mathcal{N}(0, \sigma_{c\dot{\delta t}_{IF}}^2 \mathbf{I}_{N_{freq}-1}) \\ \mathbf{w}_{c\delta t_{IS},n} &\sim \mathcal{N}(0, \sigma_{c\delta t_{IS}}^2 \mathbf{I}_{N_{sys}-1}) \\ \mathbf{w}_{c\dot{\delta t}_{IS},n} &\sim \mathcal{N}(0, \sigma_{c\dot{\delta t}_{IS}}^2 \mathbf{I}_{N_{sys}-1}) \end{aligned} \quad (6.14)$$

where  $\mathbf{I}_N$  is the identity matrix of size  $N$ . Considering the time span between epochs to be constant, the resulting state transition matrix  $\mathbf{F}_n$  is also constant for all  $n$  and it is given by

$$\mathbf{F} = \begin{bmatrix}
\mathbf{I}_3 & \Delta t \mathbf{I}_3 & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 2(N_{freq}-1)} & \mathbf{0}_{3 \times 2(N_{sys}-1)} \\
\mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 2(N_{freq}-1)} & \mathbf{0}_{3 \times 2(N_{sys}-1)} \\
\mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{I}_2 & \mathbf{0}_{2 \times 2(N_{freq}-1)} & \mathbf{0}_{2 \times 2(N_{sys}-1)} \\
\mathbf{0}_{2(N_{freq}-1) \times 3} & \mathbf{0}_{2(N_{freq}-1) \times 3} & \mathbf{0}_{2(N_{freq}-1) \times 2} & \mathbf{I}_{2(N_{freq}-1)} & \mathbf{0}_{2(N_{freq}-1) \times 2(N_{sys}-1)} \\
\mathbf{0}_{2(N_{sys}-1) \times 3} & \mathbf{0}_{2(N_{sys}-1) \times 3} & \mathbf{0}_{2(N_{sys}-1) \times 2} & \mathbf{0}_{2(N_{sys}-1) \times 2(N_{freq}-1)} & \mathbf{I}_{2(N_{sys}-1)}
\end{bmatrix} \quad (6.15)$$

where  $\mathbf{0}_{3 \times 2}$  is an all-zeros matrix of size  $3 \times 2$ . The corresponding process noise matrix  $\mathbf{Q}$  is a diagonal matrix given by

$$\mathbf{Q} = \text{diag} \begin{bmatrix}
\mathbf{0}_{3 \times 1} \\
\sigma_u^2 \mathbf{1}_3 \\
\sigma_{\dot{c}dt_u}^2 \\
\sigma_{\dot{c}dt_u}^2 \\
\sigma_{\dot{c}dt_{IF}}^2 \mathbf{1}_{N_{freq}-1 \times 1} \\
\sigma_{\dot{c}dt_{IF}}^2 \mathbf{1}_{N_{freq}-1 \times 1} \\
\sigma_{\dot{c}dt_{IS}}^2 \mathbf{1}_{N_{sys}-1 \times 1} \\
\sigma_{\dot{c}dt_{IS}}^2 \mathbf{1}_{N_{sys}-1 \times 1}
\end{bmatrix} \quad (6.16)$$

### 6.2.2 Measurement model

The measurement model describes how the observations are related to the system states. A measurement vector,  $\mathbf{z}_n$ , containing all the observations is built. In this case, the measurement vector at epoch  $n$  contains all the pseudoranges,  $\boldsymbol{\rho}$ , and pseudorange rates,  $\dot{\boldsymbol{\rho}}$ , as

$$\mathbf{z}_n = \begin{bmatrix} \boldsymbol{\rho}^T & \dot{\boldsymbol{\rho}}^T \end{bmatrix}^T \quad (6.17)$$

Commonly, the measurement vector can be modeled as a function of the state vector  $\mathbf{z}_n$  as

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n) + \mathbf{v}_n \quad (6.18)$$

where  $\mathbf{h}_n$  is the function that relates the states with the measurements and  $\mathbf{v}_n$  denotes the measurement noise vector that describes the expected Gaussian noise of the measurements. This measurement has its associated measurement covariance matrix which is given by

$$\mathbf{R}_n = \mathbb{E}[\mathbf{v}_n \mathbf{v}_n^T] \quad (6.19)$$

In this case, the measurement vector contains the code pseudoranges and pseudorange rates of all of the constellations in use.

$$\mathbf{x}_n = \begin{bmatrix} \boldsymbol{\rho}^T & \dot{\boldsymbol{\rho}}^T \end{bmatrix}^T \quad (6.20)$$

where  $\boldsymbol{\rho}$  and  $\dot{\boldsymbol{\rho}}$  are column vectors containing all the code pseudoranges and pseudorange rates, respectively.

From the equations (6.3) and (6.6), the measurement equations, which describe the relation between the state and the measurements, can be deduced. Firstly, the function that describes the pseudorange measurements can be obtained by developing the geometric range  $r_i$  between the satellite and the receiver. Considering the pseudoranges are corrected by the satellite clock and the atmospheric effects, the function  $\mathbf{h}_{\rho,n}(\mathbf{x}_n)$  relates the corrected pseudoranges with the state vector as

$$\mathbf{h}_{\rho,n}(\mathbf{x}_n)_{[i]} = |\mathbf{s}_{i,n} - \mathbf{u}_n| + c\delta t_{u,n} + c\delta t_S \quad (6.21)$$

where the subscript  $[i]$  denotes the  $i$ -th row of the measurement function. This non-linear measurement function  $\mathbf{h}_{\rho,n}(\mathbf{x}_n)$  is then linearized by a 1st order Taylor approximation in order to obtain the Jacobian matrix of  $\mathbf{h}_{\rho,n}(\mathbf{x}_n)$ . This linearization is performed around the predicted state, denoted by  $\mathbf{x}_n^-$ .

$$\mathbf{h}_{\rho,n}(\mathbf{x}_n) \approx \mathbf{h}_{\rho,n}(\mathbf{x}_n^-) + \nabla \mathbf{h}_{\rho,n}(\mathbf{x}_n) \Big|_{\mathbf{x}_n = \mathbf{x}_n^-} (\mathbf{x}_n - \mathbf{x}_n^-) \quad (6.22)$$

The Jacobian matrix of  $\mathbf{h}_{\rho,n}(\mathbf{x}_n)$ , results in the pseudorange measurement matrix  $\mathbf{H}_{\rho,n}$

$$\mathbf{H}_{\rho,n} = \nabla \mathbf{h}_{\rho,n}(\mathbf{x}_n) \Big|_{\mathbf{x}_n = \mathbf{x}_n^-} \quad (6.23)$$

In order to simplify the notation, the subscript  $n$  denoting the epoch is dropped. However, it must be taken into account that the LOS vectors  $\mathbf{e}_i$  depend on the positions of the satellite  $i$  and the receiver and thus they will change on different epochs. Considering as an example that two frequencies and three constellations are being used, the measurement matrix can be defined as follows

$$\mathbf{H}_\rho = \begin{bmatrix} \mathbf{e}_1^T & \mathbf{0}_{1 \times 3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{e}_1^T & \mathbf{0}_{1 \times 3} & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{e}_S^T & \mathbf{0}_{1 \times 3} & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{e}_{N_{sat}}^T & \mathbf{0}_{1 \times 3} & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (6.24)$$

where the 9th column corresponds to the Inter-Frequency bias and the 11th and 12th columns correspond to the Inter-System biases of the two non-reference constellations. Thus, the rows that match the measurements from the non-reference frequency will contain a 1 on the column corresponding to that frequency, and the same logic is applied to the Inter-System bias. For example, in (6.24), the first and second rows correspond to measurements on two frequencies of the reference constellation and the third and fourth rows that are shown correspond to measurements on the reference frequency of two different non-reference constellations.

Regarding the pseudorange rate, the measurement function can be obtained from the equations (6.6) and (6.7). Considering that the pseudorange rate has been corrected by the satellite's clock drift, which is obtained from the navigation message, the measurement function  $\mathbf{h}_{\dot{\rho}}(\mathbf{x})$  can be written as

$$\mathbf{h}_{\dot{\rho}}(\mathbf{x})_{[i]} = (\dot{\mathbf{s}}_i - \dot{\mathbf{u}})^T \mathbf{e}_i + c\dot{\delta}t_u + c\dot{\delta}t_{IF} + c\dot{\delta}t_{IS} \quad (6.25)$$

Thus, the measurement matrix for the pseudorange rates,  $\mathbf{H}_{\dot{\rho}}$  can be defined as follows

$$\mathbf{H}_{\dot{\rho}} = \begin{bmatrix} \mathbf{0}_{1 \times 3} & \mathbf{e}_1^T & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{0}_{1 \times 3} & \mathbf{e}_1^T & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{1 \times 3} & \mathbf{e}_S^T & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{1 \times 3} & \mathbf{e}_{N_{sat}}^T & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.26)$$

Finally, the measurement matrix for the SPPV EKF writes

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_\rho \\ \mathbf{H}_{\dot{\rho}} \end{bmatrix} \quad (6.27)$$

The corresponding measurement covariance matrix is, then

$$\mathbf{R} = \begin{bmatrix} \sigma_\rho^2 \mathbf{I}_{N_{sat}} & \mathbf{0}_{N_{sat} \times N_{sat}} \\ \mathbf{0}_{N_{sat} \times N_{sat}} & \sigma_\rho^2 \mathbf{I}_{N_{sat}} \end{bmatrix} \quad (6.28)$$

### 6.2.3 Prediction and update steps

An iteration of the KF algorithm comprises two main steps, namely the prediction and the update. On the first step, the system state and its covariance are predicted from the estimates of the previous epoch. With this step, a first *a priori* estimation is obtained. On the second step, the predictions are updated with the measurements obtained at that epoch. After this step, an updated *a posteriori* estimation is obtained, which is used as an input for the prediction of the following epoch. The weight of the correction provided by the measurements in the update step is given by the so called Kalman gain, which is computed from the covariance of the prediction and the measurement covariance. These two steps are summarized in the following equations.

#### 6.2.3.1 Prediction step

$$\hat{\mathbf{x}}_n^- = \mathbf{F} \hat{\mathbf{x}}_{n-1}^+ \quad (6.29)$$

$$\mathbf{P}_n^- = \mathbf{F} \mathbf{P}_{n-1}^+ \mathbf{F}^T + \mathbf{Q}_{n-1} \quad (6.30)$$

where (6.29) is the equation to obtain the *a priori* state estimate of the current epoch,  $\hat{\mathbf{x}}_n^-$ , from the *a posteriori* state estimate of the previous epoch,  $\hat{\mathbf{x}}_{n-1}^+$ . Similarly, equation (6.30) is the prediction of the *a priori* state covariance  $\mathbf{P}_n^-$  from the *a posteriori* state covariance of the previous epoch,  $\mathbf{P}_{n-1}^+$ .

#### 6.2.3.2 Update step

$$\hat{\mathbf{x}}_n^+ = \hat{\mathbf{x}}_n^- + \mathbf{K}_n (\mathbf{z}_n - \mathbf{h}_n(\hat{\mathbf{x}}_n^-)) \quad (6.31)$$

$$\mathbf{P}_n^+ = (\mathbf{I} - \mathbf{K}_n \mathbf{H}_n) \mathbf{P}_n^- \quad (6.32)$$

$$\mathbf{K}_n = \mathbf{P}_n^- \mathbf{H}_n^T (\mathbf{H}_n \mathbf{P}_n^- \mathbf{H}_n^T + \mathbf{R}_n)^{-1} \quad (6.33)$$

where (6.31) is the equation to obtain the *a posteriori* state estimate,  $\hat{\mathbf{x}}_n^+$ , from the *a priori* state estimate,  $\hat{\mathbf{x}}_n^-$  and equation (6.32) is the *a posteriori* state covariance  $\mathbf{P}_n^+$  obtained from the *a priori* state covariance,  $\mathbf{P}_n^-$ . As previously stated, both updates rely on the Kalman gain  $\mathbf{K}_n$  which is computed as shown in equation (6.33). Finally, it is important to note that the term between parenthesis in (6.31) is usually referred to as innovation and it is denoted by  $\mathbf{y}_n$ :

$$\mathbf{y}_n = \mathbf{z}_n - \mathbf{h}_n(\hat{\mathbf{x}}_n^-) \quad (6.34)$$

### 6.3 Sequential Kalman Filter

Considering  $m$  as the number of measurements, the update step of the Kalman Filter requires the inversion of an  $m \times m$  matrix in order to compute the Kalman gain. This becomes a problem when a large number of measurements are used, as the matrix inversion is a computationally costly operation. To solve this problem, the Sequential Kalman Filter can be implemented, which do not require matrix inversion [Sim06]. This implementation assumes that the measurement covariance matrix  $\mathbf{R}$  is diagonal. Then, the measurement model can be expressed as

$$y_{n,i} = \mathbf{H}_{n[i,:]} \mathbf{x}_n + v_{n,i} \quad | \quad i = 1 \dots m \quad (6.35)$$

where  $y_{n,i}$  is the  $i$ -th measurement in  $\mathbf{y}_n$ ,  $v_{n,i}$  is the  $i$ -th element in  $\mathbf{v}_n$ , and  $\mathbf{H}_{n[i,:]}$  is the  $i$ -th row of  $\mathbf{H}_n$ . Then, instead of processing the measurements at time  $n$  as a vector, the Kalman Filter is implemented such that the update is performed one measurement at a time. Then, the equations (6.31) (6.32) (6.33) become operations between vectors which result in a scalar *a posteriori* estimation  $\hat{\mathbf{x}}_{n,i}^+$  obtained after processing the  $i$ -th measurement. This *a posteriori* estimation of the  $i$ -th measurement becomes the *a priori* estimation used to process the measurement  $i + 1$ . The *a posteriori* estimation of the  $m$ -th measurement becomes the estimation of the  $n$ -th epoch:

$$\begin{aligned} \hat{\mathbf{x}}_n^+ &= \hat{\mathbf{x}}_{n,m}^+ \\ \mathbf{P}_n^+ &= \mathbf{P}_{n,m}^+ \end{aligned} \quad (6.36)$$

### 6.4 Single-receiver RTK

The previous method is highly dependent on the quality of the measurement corrections. These corrections, though, aren't perfect and usually a residual error is left after correcting the measurements. This is specially true with the atmospheric errors which are more difficult to accurately model [Teu17]. For this reason, Differential GNSS (DGNSS) techniques are usually implemented for applications requiring a high accuracy. One of the most used DGNSS techniques is the RTK positioning. RTK is based in the use of measurements from a reference station whose position is well known to eliminate the common errors between station and receiver, namely the atmospheric and satellite errors. Considering the distance between the receiver and the reference station short enough (less than 20 km), the mentioned errors can be assumed to be

the same [Teu17]. Furthermore, RTK also utilizes phase measurements since the ambiguities can be resolved in a few minutes for short distances between receiver and reference station [Teu17]. Typically, a RTK algorithm uses only pseudorange and carrier phase measurements. Nevertheless, in the typical scenario that is encountered in this project the receiver is moving and, thus, it is interesting to include the non-differenced Doppler measurements to estimate the velocity. This is expected to indirectly improve the accuracy of the position estimation since the Doppler measurements are usually less noisy and do not have ambiguities. Therefore, the position estimations provided by the ranging measurements (i.e. pseudorange and carrier phase) will be smoothed by the velocity estimate thanks to the Kalman filter prediction.

The RTK algorithm begins by computing the double differences of the pseudorange and phase measurements. First, Single Differences (SDs) between receiver and station are computed, where the atmospheric and satellite errors are eliminated.

$$\rho_i^{SD} = \rho_{s,i} - \rho_{u,i} = r_{s,i} - r_{u,i} + c(\delta t_s - \delta t_u) + \varepsilon_{\rho,i}^{SD} \quad (6.37)$$

$$\phi_i^{SD} = \phi_{s,i} - \phi_{u,i} = r_{s,i} - r_{u,i} + c(\delta t_s - \delta t_u) + \lambda N_i^{SD} + \varepsilon_{\phi,i}^{SD} \quad (6.38)$$

where  $\rho_i^{SD}$  and  $\phi_i^{SD}$  are the SD code and carrier measurements, respectively. Sub-index  $s$  stands for reference station receiver and sub-index  $u$  stands for user receiver. The single-differenced ambiguities are denoted by  $N_i^{SD} = N_{s,i} - N_{u,i}$ . The residual errors after computing the single differences are all included in the terms  $\varepsilon_{\rho,i}^{SD}$  and  $\varepsilon_{\phi,i}^{SD}$ .

Then, in order to remove the receiver clock biases, Double Differences (DDs) are computed. Here, a satellite is taken as a pivot, and the measurements from the non-pivot satellites are subtracted to the measurement of the pivot satellite.

$$\rho_j^{DD} = \rho_p^{SD} - \rho_j^{SD} = r_{s,p} - r_{u,p} - r_{s,j} + r_{u,j} + \varepsilon_{\rho,j}^{DD} \quad (6.39)$$

$$\phi_j^{DD} = \phi_p^{SD} - \phi_j^{SD} = r_{s,p} - r_{u,p} - r_{s,j} + r_{u,j} + \lambda(N_p^{SD} - N_j^{SD}) + \varepsilon_{\phi,j}^{DD} \quad (6.40)$$

where  $j$  denotes the index of the DD measurement and is in the range  $1, \dots, N_{DD}$ . Following the same notation,  $\rho_j^{DD}$  and  $\phi_j^{DD}$  denote the double-differenced code and carrier measurements, respectively; and  $\varepsilon_{\rho,j}^{DD}$  and  $\varepsilon_{\phi,j}^{DD}$  contain the residual errors after computing the double-differences.

It is worth noting that two different approaches can be followed regarding the ambiguities in equation (6.40). One approach is to join  $N_p^{SD} - N_j^{SD}$  as single double-differenced ambiguities and include them into the state vector. However, following this approach, every time the pivot satellite changes all the glsdd ambiguities need to be re-estimated. This approach can be interesting to implement for applications in which the reception of GNSS signals is very stable and there is not much signal blocking. However, when this is not the case, it may be more interesting

to keep track of the single-differenced ambiguities and compute the double differences each time. Although this approach is a bit more complex in the implementation, it is the one selected for this project as it has been seen that the GNSS measurements obtained from smartphones often lose track of the signals.

Every time a new signal is acquired, either because a new satellite is in view or because the receiver has reacquired lock with the signal, an initial estimation of the ambiguity has to be obtained. In order to obtain these initial estimations, the Code-Minus-Carrier (CMC) is used.

$$CMC_i^{SD} = \rho_{SD,i} - \phi_{SD,i} = -\lambda N_i^{SD} + \varepsilon_{\rho,i}^{SD} - \varepsilon_{\phi,i}^{SD} \quad (6.41)$$

$$\hat{N}_{SD,i} = -\frac{CMC_{SD,i}}{\lambda} \quad (6.42)$$

where  $\hat{N}_{SD,i}$  is the estimated single-differenced ambiguity.

#### 6.4.1 RTK Processing

Since this RTK problem is linear, it is implemented with a basic Kalman Filter. In this case, the single-differenced ambiguities are included in the state vector as parameters to estimate. This time, however, there is no need to estimate the receiver clock bias as it is removed by differentiation.

Since RTK is a DGNSS method, the estimated position is relative to the reference station. Thus, actual the parameter to estimate is the baseline vector,  $\mathbf{b}_{us}$ , between the receiver position,  $\mathbf{u}$ , and the station's,  $\mathbf{s}$ .

$$\mathbf{b}_{us} = \mathbf{u} - \mathbf{s} \quad (6.43)$$

Then, the state vector can be written as

$$\mathbf{x}_n = \left[ \mathbf{b}_{us}^T \quad \dot{\mathbf{u}}^T \quad \delta \dot{t}_u \quad (\dot{\delta t}_{IF})^T \quad (\dot{\delta t}_{IS})^T \quad (\mathbf{N}^{SD})^T \right]^T \quad (6.44)$$

where  $\mathbf{N}^{SD}$  is a column vector containing the SD ambiguities of all the common satellites between the receiver and the station. Since this project uses multiple receivers and multiple frequency bands, one pivot satellite has to be selected for each pair of constellation and frequency; therefore  $\mathbf{N}^{SD}$  is of size  $N_{DD} = N_{meas} - N_{piv}$ , where  $N_{meas}$  is the number of carrier phase measurements available and  $N_{piv}$  is the number of pivot satellites (i.e. number of constellation-frequency pairs).

In this case, the measurement vector contains the DD pseudorange and carrier phase measurements as well as the non-differenced pseudorange rates



$$\mathbf{z}_n = \begin{bmatrix} \boldsymbol{\rho}^{DD^T} & \boldsymbol{\phi}^{DD^T} & \dot{\boldsymbol{\rho}}^T \end{bmatrix}^T \quad (6.45)$$

#### 6.4.1.1 State transition model

The state transition model for this RTK is very similar to the SPPV RTK. The estimated velocity of the receiver can be directly added to the receiver-station baseline to perform the prediction of the relative position. The rest of the state parameters are modeled as 1st-order Markov processes. Thus, the state transition model is

$$\begin{aligned} \mathbf{b}_{us_n} &= \mathbf{b}_{us_{n-1}} + \dot{\mathbf{u}}_{n-1} \Delta t \\ \dot{\mathbf{u}}_n &= \dot{\mathbf{u}}_{n-1} + \mathbf{w}_{\dot{\mathbf{u}},n} \\ c\dot{\delta}t_{u,n} &= c\dot{\delta}t_{u,n-1} + w_{c\dot{\delta}t_u,n} \\ c\dot{\delta}t_{IF,n} &= c\dot{\delta}t_{IF,n-1} + \mathbf{w}_{c\dot{\delta}t_{IF},n} \\ c\dot{\delta}t_{IS,n} &= c\dot{\delta}t_{IS,n-1} + \mathbf{w}_{c\dot{\delta}t_{IS},n} \\ \mathbf{N}_n^{SD} &= \mathbf{N}_{n-1}^{SD} + \mathbf{w}_{N^{SD},n} \end{aligned} \quad (6.46)$$

where

$$\begin{aligned} \mathbf{w}_{\dot{\mathbf{u}},n} &\sim \mathcal{N}(0, \sigma_u^2 \mathbf{I}_3) \\ w_{c\dot{\delta}t_u,n} &\sim \mathcal{N}(0, \sigma_{c\dot{\delta}t_u}^2) \\ \mathbf{w}_{c\dot{\delta}t_{IF},n} &\sim \mathcal{N}(0, \sigma_{c\dot{\delta}t_{IF}}^2 \mathbf{I}_{N_{freq}-1}) \\ \mathbf{w}_{c\dot{\delta}t_{IS},n} &\sim \mathcal{N}(0, \sigma_{c\dot{\delta}t_{IS}}^2 \mathbf{I}_{N_{sys}-1}) \\ \mathbf{w}_{N^{SD},n} &\sim \mathcal{N}(0, \sigma_{N^{SD}}^2 \mathbf{I}_{N_{DD}}) \end{aligned} \quad (6.47)$$

Then, the state transition matrix can be written as

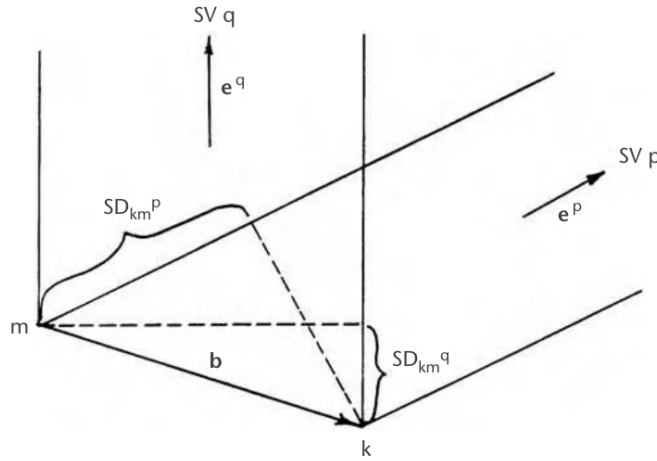
$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \Delta t \mathbf{I}_3 & 0 & \cdots & 0 \\ \mathbf{0} & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \quad (6.48)$$

and the corresponding process noise matrix  $\mathbf{Q}$  is given by the diagonal matrix

$$Q = \text{diag} \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \sigma_u^2 \mathbf{1}_{3 \times 1} \\ \sigma_{\dot{c}t_u}^2 \\ \sigma_{\dot{c}t_{IF}}^2 \mathbf{1}_{N_{freq}-1 \times 1} \\ \sigma_{\dot{c}t_{IS}}^2 \mathbf{1}_{N_{sys}-1 \times 1} \\ \sigma_{N^{SD}}^2 \mathbf{1}_{N_{DD} \times 1} \end{bmatrix} \quad (6.49)$$

#### 6.4.1.2 Measurement model

Considering the distance between the receiver and the station to be small enough, the double difference of the geometric ranges in equations (6.39) and (6.40) can be rewritten as the difference of the LOS vectors between the station and the two satellites, projected onto the baseline vector between station and receiver. Figure 6.1 from chapter 8 in [Kap06] shows this relationship.



**Figure 6.1:** Double differences as projections. *SV p* and *SV q* are the pivot and non-pivot satellites, and *m* and *n* are the station and the receiver [Kap06]

Then, the measurement equation for the DD pseudorange is

$$\mathbf{h}_{\rho^{DD}}(\mathbf{x})_{[j]} = (\mathbf{e}_{s,p} - \mathbf{e}_{s,j})^T \mathbf{b}_{us} \quad (6.50)$$

where  $\mathbf{e}_{s,p}$  and  $\mathbf{e}_{s,j}$  are the LOS unit vectors from the station to the pivot and non-pivot satellites, respectively. Similarly, the measurement equation for the DD carrier phase is

$$\mathbf{h}_{\phi^{DD}}(\mathbf{x})_{[j]} = (\mathbf{e}_{s,p} - \mathbf{e}_{s,j})^T \mathbf{b}_{us} + \lambda(N_p^{SD} - N_j^{SD}) \quad (6.51)$$

From these two equations and considering the same model of equation (6.25) for the pseudorange rates, the measurement matrix is defined as

$$\mathbf{H}_n = \begin{bmatrix} \mathbf{H}_{\rho^{DD},n} \\ \mathbf{H}_{\phi^{DD},n} \\ \mathbf{H}_{\dot{\rho},n} \end{bmatrix} \quad (6.52)$$

where

$$\mathbf{H}_{\rho^{DD},n_{[j:]}} = \begin{bmatrix} (\mathbf{e}_{s,p} - \mathbf{e}_{s,j})^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times N_{freq}-1} & \mathbf{0}_{1 \times N_{sys}-1} & \mathbf{0}_{1 \times N_{DD}} \end{bmatrix} \quad (6.53)$$

$$\mathbf{H}_{\phi^{DD},n_{[j:]}} = \begin{bmatrix} (\mathbf{e}_{s,p} - \mathbf{e}_{s,j})^T & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times N_{freq}-1} & \mathbf{0}_{1 \times N_{sys}-1} & \lambda & 0 & \cdots & 0 & -\lambda \end{bmatrix} \quad (6.54)$$

$$\mathbf{H}_{\dot{\rho},n_{[i:]}} = \begin{bmatrix} \mathbf{0}_{1 \times 3} & (\mathbf{e}_{s,p} - \mathbf{e}_{s,j})^T & \mathbf{a}_i^T & \mathbf{b}_i^T & \mathbf{0}_{1 \times N_{sat}} \end{bmatrix} \quad (6.55)$$

with the  $\lambda$  and  $-\lambda$  in (6.54) located at the columns that correspond to the pivot and non-pivot satellites, respectively. Vectors  $\mathbf{a}_i$  and  $\mathbf{b}_i$  contain a 1 at the column that corresponds to the frequency and the constellation of the  $i$ -th measurement, respectively.

The measurement covariance matrix for the DD measurements can be obtained from the covariance matrices of SD measurements. Considering that the noise of the receiver measurements is independent from that of the station measurements, the SD measurement covariance matrices can be obtained by simple addition of the measurement covariances of the two sources. Considering the measurement covariances to be the same in the receiver and in the station, the SD measurement covariance matrices can be defined as

$$\mathbf{R}_{\rho^{SD}} = 2\mathbf{R}_{\rho} \quad (6.56)$$

$$\mathbf{R}_{\phi^{SD}} = 2\mathbf{R}_{\phi} \quad (6.57)$$

Then, the DD measurement covariance matrices are given by

$$\mathbf{R}_{\rho^{DD}} = \mathbf{D}\mathbf{R}_{\rho^{SD}}\mathbf{D}^T \quad (6.58)$$

$$\mathbf{R}_{\phi^{DD}} = \mathbf{D}\mathbf{R}_{\phi^{SD}}\mathbf{D}^T \quad (6.59)$$

where  $\mathbf{D}$  is the double-differencing matrix where the location of the column full of 1's correspond to the pivot satellite and the location of the -1's matches the position of the non-pivot satellite:

$$\mathbf{D} = \begin{bmatrix} -1 & 0 & \cdots & 0 & 1 \\ 0 & -1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \quad (6.60)$$

The resulting covariance matrices  $\mathbf{R}_{\rho^{DD}}$  and  $\mathbf{R}_{\phi^{DD}}$  are non-diagonal, which results in a non-diagonal measurement covariance matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{\rho^{DD}} & \mathbf{0}_{N_{DD}} & \mathbf{0}_{N_{DD} \times N_{sat}} \\ \mathbf{0}_{N_{DD}} & \mathbf{R}_{\phi^{DD}} & \mathbf{0}_{N_{DD} \times N_{sat}} \\ \mathbf{0}_{N_{sat} \times N_{DD}} & \mathbf{0}_{N_{sat} \times N_{DD}} & \sigma_{\rho}^2 \mathbf{I}_{N_{sat}} \end{bmatrix} \quad (6.61)$$

As explained in section 6.3, this non-diagonal matrix presents a problem when implementing a sequential Kalman Filter update. One option to solve this issue is to perform a diagonalization of the resulting matrix by Cholesky decomposition, as it is explained in section 7.4 of [Pet79]. This option was tested during the realization of this project but the accuracy of the position estimation was reduced. This is not expected considering that the inclusion of the non-diagonal terms in the covariance matrix improves the modeling of the noise, which should improve the performance of the algorithm. Given the limited time for this project, this issue was faced by simply considering a diagonal measurement matrix and inflating the variances of the DD measurements.

$$\mathbf{R} = \begin{bmatrix} \sigma_{\rho^{DD}}^2 \mathbf{I}_{N_{DD}} & \mathbf{0}_{N_{DD} \times N_{DD}} & \mathbf{0}_{N_{DD} \times N_{sat}} \\ \mathbf{0}_{N_{DD} \times N_{DD}} & \sigma_{\phi^{DD}}^2 \mathbf{I}_{N_{DD}} & \mathbf{0}_{N_{DD} \times N_{sat}} \\ \mathbf{0}_{N_{DD} \times N_{DD}} & \mathbf{0}_{N_{DD} \times N_{DD}} & \sigma_{\rho}^2 \mathbf{I}_{N_{sat}} \end{bmatrix} \quad (6.62)$$

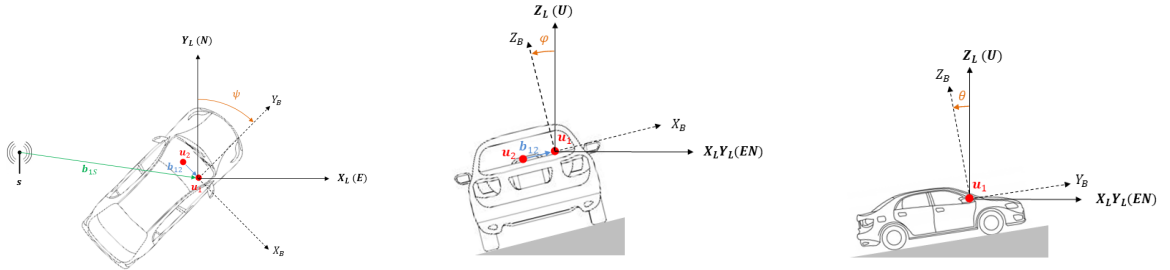
## 6.5 Multiple-receiver RTK

In many of the measurement campaigns, more than one smartphone was used at the same time. This opens the possibility to combine the measurements from the different receivers in order to improve the performance of the RTK, as well as provide estimation on the attitude. The technique implemented in this project is based on the research done by Hu et al. [Hu20b] [Hu20a] [Hu21]. This method requires the relative geometry of the receivers to be known in order to combine the measurements. This geometry is not directly provided for the Google Smartphone Decimeter Challenge, but it can be deduced from the ground-truth files as they

contain the position of each smartphone and the attitude of the vehicle at each epoch. Then, the geometries that are obtained from the *train* dataset, which contains the ground-truth, are extrapolated for the *test* dataset, for which no ground-truth is provided.

### 6.5.1 Multiple-receiver geometry

In order to combine the measurements from multiple smartphones, their relative position inside the vehicle needs to be considered. To do so, a body frame is defined with the origin centered in one of the smartphones, called the *master*. Then, the positions of the other smartphones, called the *slaves*, are defined relative to the master smartphone. Three attitude angles are defined: pitch  $\theta$ , roll  $\varphi$  and yaw  $\psi$ . The capacity to observe these attitude angles depends on the number and geometry of the smartphones in the car. In most of the campaigns the smartphones are aligned with the perpendicular axis of the vehicle, which does not allow the filter to observe the pitch angle. This, however, is not a problem for this project since the accuracy of the algorithm is evaluated in the horizontal error which is not very affected by an error in the pitch estimate. Figure 6.2 shows an example of a two-receiver geometry in the typical configuration that is found in the datasets. In this section, the master receiver is always indicated by the subscript 1 and the slave receivers are indicated by a subscript from 2 to  $N_{rx}$ .



**Figure 6.2:** Multi-receiver geometry

The relative position of the slave receivers is then transformed from body frame to the Earth-Centered, Earth-Fixed (ECEF) frame. This transformation can be understood as a two-step operation. First, a transformation is made from body frame to Local Tangent Plane (LTP), expressed in East-North-Up (ENU), and then a second transformation is made from LTP to ECEF. These two operations are made by the rotation matrices  $\mathbf{C}_{B2L}$  and  $\mathbf{C}_{L2E}$ , respectively. For the 2-smartphone example, this is

$$\mathbf{b}_{12}^E(\theta, \psi, \varphi) = \mathbf{C}_{L2E} \mathbf{C}_{B2L}(\theta, \varphi, \psi) \mathbf{b}_{12}^B \quad (6.63)$$

where  $\mathbf{b}_{12}^E$  and  $\mathbf{b}_{12}^B$  are the baseline between master and slave smartphones expressed in ECEF and in body frame, respectively. Section 2.4 in [Gro08] details how to build both rotation

matrices. However, the rotation matrix from body to LTP is shown next as it will be necessary in the following section.

$$\mathbf{C}_{B2L}(\theta, \varphi, \psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (6.64)$$

### 6.5.2 RTK processing

For this multi-receiver method, the attitude angles are included as state parameters, as well as the clock drifts and SD ambiguities of all receivers. Considering the example of two receivers, the state vector results in

$$\mathbf{x} = \begin{bmatrix} \mathbf{b}_{1s}^E \\ \theta \\ \varphi \\ \psi \\ \dot{\mathbf{u}}^E \\ \dot{\delta t}_1 \\ \dot{\delta t}_2 \\ \dot{\delta t}_{IF} \\ \dot{\delta t}_{IS} \\ \mathbf{N}_{1s}^{SD} \\ \mathbf{N}_{2s}^{SD} \end{bmatrix} \quad (6.65)$$

where  $\mathbf{N}_{ks}^{SD}$  is the SD ambiguity vector for the  $k$ -th smartphone.

Following the example of the two receivers, the measurement vector contains the DD measurements and the pseudorange rates from each receiver as

$$\mathbf{z}_n = \left[ \boldsymbol{\rho}^{DD^T}_1 \quad \boldsymbol{\rho}^{DD^T}_2 \quad \boldsymbol{\phi}^{DD^T}_1 \quad \boldsymbol{\phi}^{DD^T}_2 \quad \dot{\boldsymbol{\rho}}_1^T \quad \dot{\boldsymbol{\rho}}_2^T \right]^T \quad (6.66)$$

### 6.5.2.1 Transition model

The state transition model is built very similarly to the single-receiver RTK.

$$\begin{aligned}
\mathbf{b}_{1s_n}^E &= \mathbf{b}_{1s_{n-1}}^E + \dot{\mathbf{u}}_{n-1} \Delta t \\
\theta_n &= \theta_{n-1} + w_{\theta,n} \\
\varphi_n &= \varphi_{n-1} + w_{\varphi,n} \\
\psi_n &= \psi_{n-1} + w_{\psi,n} \\
\dot{\mathbf{u}}_n &= \dot{\mathbf{u}}_{n-1} + \mathbf{w}_{\dot{\mathbf{u}},n} \\
c\dot{\delta t}_{k,n} &= c\dot{\delta t}_{k,n-1} + \mathbf{w}_{c\dot{\delta t}_k,n} \\
c\dot{\delta t}_{IF,n} &= c\dot{\delta t}_{IF,n-1} + \mathbf{w}_{c\dot{\delta t}_{IF},n} \\
c\dot{\delta t}_{IS,n} &= c\dot{\delta t}_{IS,n-1} + \mathbf{w}_{c\dot{\delta t}_{IS},n} \\
\mathbf{N}_{ks,n}^{SD} &= \mathbf{N}_{ks,n-1}^{SD} + \mathbf{w}_{N_{ks}^{SD},n}
\end{aligned} \tag{6.67}$$

where

$$w_{\theta,n} \sim \mathcal{N}(0, \sigma_{\theta}^2), \quad w_{\varphi,n} \sim \mathcal{N}(0, \sigma_{\varphi}^2), \quad w_{\psi,n} \sim \mathcal{N}(0, \sigma_{\psi}^2) \tag{6.68}$$

Then, the state transition matrix can be built very similarly to the case of the single-receiver RTK in (6.48). This is, an identity matrix plus the time span  $\delta t$  in the off-diagonal terms that relate the velocity with the relative position. The process noise covariance matrix is also very similar to the single-receiver case in (6.49) including the attitude variances in the diagonal.

### 6.5.2.2 Measurement model

#### Pseudorange and carrier phase

The measurement model is defined such that the relative position of the master receiver is estimated. The measurement equation for the master receiver is defined as in the single receiver case of equations (6.50) and (6.51). Then, the measurement equations of the slave receivers are defined considering the baseline between smartphones as

$$\begin{aligned}
\mathbf{h}_{\rho_k,n}(\mathbf{x})_{[j:]} &= (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T (\mathbf{b}_{1s}^E - \mathbf{b}_{12}^E) \\
&= (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T \left( \mathbf{b}_{1s} - \mathbf{C}_{L2E} \mathbf{C}_{B2L}(\theta, \varphi, \psi) \mathbf{b}_{12}^B \right)
\end{aligned} \tag{6.69}$$

$$\begin{aligned}
\mathbf{h}_{\phi_k,n}(\mathbf{x})_{[j:]} &= (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T (\mathbf{b}_{1s}^E - \mathbf{b}_{12}^E) + \lambda N_p^{SD} - \lambda N_j^{SD} \\
&= (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T \left( \mathbf{b}_{1s} - \mathbf{C}_{L2E} \mathbf{C}_{B2L}(\theta, \varphi, \psi) \mathbf{b}_{12}^B \right) + \lambda (N_p^{SD} - N_j^{SD})
\end{aligned} \tag{6.70}$$

Equations (6.69) and (6.70) are clearly not linear due to the rotation matrix from body to LTP. Thus, they are linearized to obtain the measurement matrices as follows

$$\mathbf{H}_{\rho_k^{DD}, n[j:]} = \begin{bmatrix} (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T & h_{\theta,k} & h_{\varphi,k} & h_{\psi,k} & 0 & \dots & 0 \end{bmatrix} \quad (6.71)$$

$$\mathbf{H}_{\phi_k^{DD}, n[j:]} = \begin{bmatrix} (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T & h_{\theta,k} & h_{\varphi,k} & h_{\psi,k} & 0 & \dots & 0 & \boldsymbol{\lambda}_j^T \end{bmatrix} \quad (6.72)$$

where  $\boldsymbol{\lambda}_j$  is a vector containing  $\lambda$  and  $-\lambda$  at the positions corresponding to the pivot and non-pivot satellites, respectively; and

$$\begin{aligned} h_{\theta,k} &= (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T \mathbf{C}_{L2E} \left( \frac{\partial}{\partial \theta} \Big|_{\theta=\hat{\theta}} \mathbf{C}_{B2L} \right) \mathbf{b}_{1k}^B \\ h_{\varphi,k} &= (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T \mathbf{C}_{L2E} \left( \frac{\partial}{\partial \varphi} \Big|_{\varphi=\hat{\varphi}} \mathbf{C}_{B2L} \right) \mathbf{b}_{1k}^B \\ h_{\psi,k} &= (\hat{\mathbf{e}}_{s,p} - \hat{\mathbf{e}}_{s,j})^T \mathbf{C}_{L2E} \left( \frac{\partial}{\partial \psi} \Big|_{\psi=\hat{\psi}} \mathbf{C}_{B2L} \right) \mathbf{b}_{1k}^B \end{aligned} \quad (6.73)$$

From equation (6.64), the partial derivative of the  $\mathbf{C}_{B2L}$  with respect to  $\theta$  is given by

$$\frac{\partial}{\partial \theta} \mathbf{C}_{B2L} = \begin{bmatrix} 0 & \cos \psi \sin \varphi \cos \theta + \sin \psi \sin \theta & -\cos \psi \sin \varphi \sin \theta + \sin \psi \cos \theta \\ 0 & \sin \psi \sin \varphi \cos \theta - \cos \psi \sin \theta & -\sin \psi \sin \varphi \sin \theta - \cos \psi \cos \theta \\ 0 & \cos \varphi \cos \theta & -\cos \varphi \sin \theta \end{bmatrix} \quad (6.74)$$

The partial derivatives with respect to  $\psi$  and  $\varphi$  can be obtained similarly from equation (6.64).

### Pseudorange rate

Regarding measurement model for the pseudorange rate, the attitude rate needs to be considered as the velocities may differ between receivers due to the lever arm. If the simple example of Fig 6.2 is considered, the velocity of the slave receiver can be expressed as a function of the velocity of the master receiver and the angular rates,  $\omega_z$  and  $\omega_y$ , as

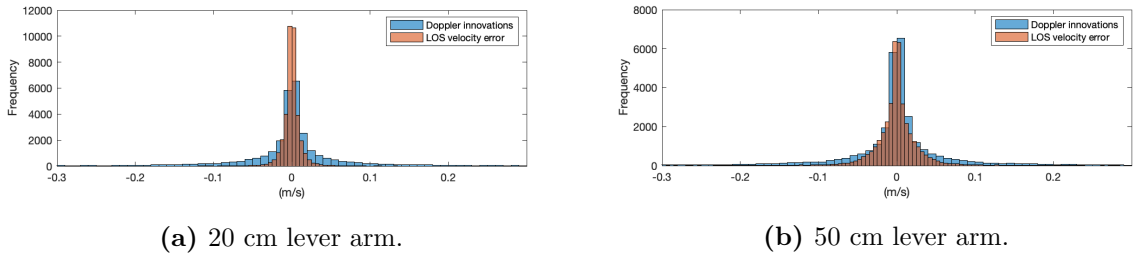
$$\dot{\mathbf{u}}_k^E = \dot{\mathbf{u}}_1^E + \mathbf{C}_{L2E} \mathbf{C}_{B2L} \|\mathbf{b}_{1k}\| \begin{bmatrix} 0 \\ \omega_z \\ \omega_y \end{bmatrix} \quad (6.75)$$

The effect of the lever arm in the velocity of the slave receiver relative to the master receiver has been studied and compared to the noise of the Doppler measurements. To do so, two examples



have been considered where two smartphones are located as in Figure 6.2 with a separation of 20 and 50 centimeters, which represent a normal and an worst case scenarios. Then, the distribution of the gyroscope measurements obtained in a urban area has been taken to compute the velocity error induced by the lever arm as in (6.75). This velocity is projected onto the LOS vectors of the example campaign so it can be compared to the Doppler measurement noise. As it can be seen in Figure 6.3, the error introduced by the lever arm in a worst-case scenario of 50 meter of separation falls inside the distribution of the Doppler noise. The error in the velocity introduced by the lever arm can then be considered as additional measurement noise by inflating the covariances of the Doppler measurements of the slave receivers.

With this approximation, the velocities of the receivers are considered to be equal which considerably simplifies the equations and removes the dependency on the attitude rate in equation 6.75. Considering this approximation, the measurement matrix for the pseudorange rates is the same for the master and the slave receivers, which is obtained from the same model in (6.25) used for the SPPV algorithm.



**Figure 6.3:** Distribution of pseudorange rate innovations (blue) and LOS velocity error induced by lever arm (red).

$$\mathbf{H}_{\dot{\rho}_k, n[i:]} = \begin{bmatrix} \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & (\mathbf{e}_{s,p} - \mathbf{e}_{s,j})^T & \mathbf{a}_i^T & \mathbf{b}_i^T & 0 & \dots & 0 \end{bmatrix} \quad (6.76)$$

Finally, the measurement matrix for the example of two receivers is

$$\mathbf{H}_n = \begin{bmatrix} \mathbf{H}_{\rho_1^{DD}, n} \\ \mathbf{H}_{\rho_2^{DD}, n} \\ \mathbf{H}_{\phi_1^{DD}, n} \\ \mathbf{H}_{\phi_2^{DD}, n} \\ \mathbf{H}_{\dot{\rho}_1, n} \\ \mathbf{H}_{\dot{\rho}_2, n} \end{bmatrix} \quad (6.77)$$



## Chapter 7

# Measurement analysis

The organizers of the Google Smartphone Decimeter Challenge provided the Android GNSS Raw measurements and the INS measurements in text files called *GnssLog*, which follow a similar format as CSV files. The GNSS Raw measurements that are provided by an Android smartphone are not directly the GNSS observables that are explained in section 6.1, so an initial mapping step has been implemented which computes the code pseudorange, carrier phase and Doppler measurements from the raw measurements. Additionally, the Android GNSS API also provides some parameters that indicate the reliability of these measurements. These parameters are used in this project to filter the measurements and to calculate the measurement covariances that are used in all the positioning methods that are described in Chapter 6. The complete list of raw measurements can be found in [Fu20] as well as in the API documentation website [Dev21]. A detailed guide on how to process Android Raw GNSS measurements can be found in ESA’s white paper [Eur17], which is the main reference that has been used for the implementation of this project. A summary of the calculation of GNSS observables is provided in the first section of this chapter.

As well as the Android Raw GNSS measurements, the organizers of the competition also provided the measurements in RINEX files. These files, however, do not contain all the information that is included in the Raw measurement files, in particular the uncertainties related to the GNSS measurements. RINEX files, however, are simpler to use since no preprocessing is required to obtain the GNSS measurements, which can be a source of errors. Furthermore the files can be read by the *magnitude* software. In order to evaluate which source is better to use, the GNSS observables obtained from preprocessing the raw measurements have been compared to the ones obtained from the RINEX files. A summary of this comparison is provided in this chapter. Then, an analysis of the reliability of the uncertainties provided by the Android Raw measurements is also given. In order to perform these analyses, a WLS and a EKF implementation from the *magnitude* software were used to obtain the clock bias estimations, the measurement residuals

and an assessment of the performance.

## 7.1 Calculation of GNSS observables

The main particularity of the Android Raw GNSS measurements is that they do not provide the pseudorange measurements directly, instead, Android provides the necessary parameters to compute them. These parameters include the internal hardware clock, the time of transmission of the signal and the bias of the receiver clock with respect to the reference GNSS system. While the reception time is obtained in the GNSS system time, the transmission time is referenced to the constellation of each measurement. Therefore, both times have to be set to the same reference before computing the time difference required to obtain the pseudoranges. For the execution of this project, measurements from up to three different GNSS constellations are used, namely the North-American GPS, the European Galileo and the Chinese BeiDou. Measurements from the Russian system GLONASS is also provided in most of the measurement files. However, the pseudoranges obtained following the description in [Eur17] were of the order of the millions of km, which is clearly wrong considering that the orbital altitude of GLONASS satellites is 19.140 km. Furthermore, since GLONASS is implemented over Frequency-Division Multiple Access, the use of Doppler measurements and the implementation of RTK becomes more complex. Therefore, this constellation was omitted in the execution of this project.

Following the steps described in section 2.4 of ESA's white paper [Eur17], the calculation of the pseudoranges can be summarized as

$$\rho = \frac{t_{Rx} - t_{Tx}}{10^9} c \quad (7.1)$$

where  $t_{Tx}$  and  $t_{Rx}$  are the transmission and reception times, respectively. The transmission time is obtained directly from the parameter *ReceivedSvTimeNanos* and it is referred to the system time of the constellation of the measurement. The reception time is computed from the hardware clock timestamp and the bias to the true GPS time. However, it is referred to the system time of the smartphone (generally GPS). The calculation of the reception time in GPS time can be summarized as the following expression

$$t_{Rx}^G = t_{clock} - t_{bias} \quad (7.2)$$

Equation (7.2) is actually a simplification of the complete calculation which is fully detailed in [Eur17]. The resulting time,  $t_{Rx}^G$  is then shifted so that it is referenced to the system time of the measurement, which provides the transmission time used in equation (7.1).

The raw parameters include uncertainties for the receiver clock time,  $\sigma_{t_{clock}}$ , the receiver clock bias,  $\sigma_{t_{bias}}$ , and the transmission time,  $\sigma_{t_{Tx}}$ . Then, the standard deviation of the pseudorange measurement can be estimated as

$$\sigma_{\rho} = c \sqrt{\sigma_{t_{clock}}^2 + \sigma_{t_{bias}}^2 + \sigma_{t_{Tx}}^2} \quad (7.3)$$

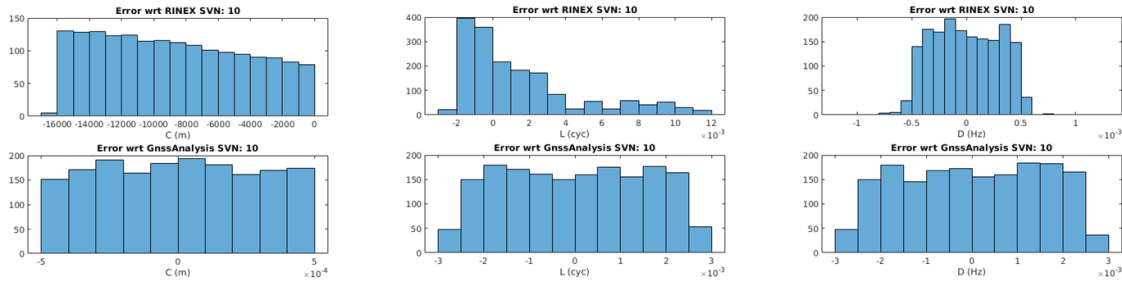
As per the carrier phase and the Doppler measurements, the first one can be directly obtained from the Accumulated Delta Range (ADR) which is provided in meters, and the second one is provided as the pseudorange rate in meters per second. Both these measurements have their associated uncertainties expressed in meters and meters per second, respectively. Therefore, they can be directly used as standard deviation estimations of the measurements.

## 7.2 Comparison of GNSS measurement sources

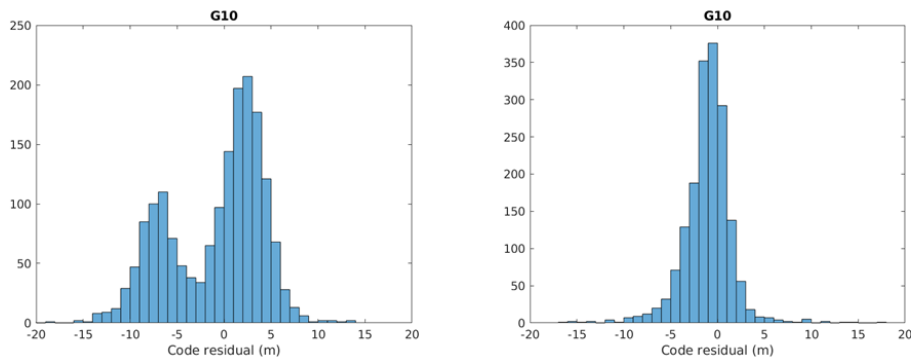
As it has been stated, the *GnssLog* files do not contain the same data as a RINEX file. In order to verify the preprocessing of GNSS observables from the raw measurements and compare them to the measurements from RINEX, the *GnssAnalysis* software was used. *GnssAnalysis* is a Desktop application developed by Android which allows the user to analyze the GNSS measurements obtained with a smartphone. Apart from a set of plots, this application can also provide a file with the preprocessed measurements. However, the code of this application is not accessible so it could not be used in our project to automate the computation of GNSS observables. Nevertheless, it could be used to obtain the observables of a single trace and compare them to the results of our preprocessing and the observables obtained from the RINEX. Figure 7.1 shows an example of the differences between the measurements generated by our preprocessing step with the measurements obtained from the RINEX files and the *GnssAnalysis* application. It can be seen that our calculations provide almost the same result as the *GnssAnalysis* for all the measurement types, with differences of half a millimeter on the pseudoranges or one hundredth of a cycle (less than 2 mm in L1) in the phase. However, the pseudoranges largely differ when compared to the ones obtained from the RINEX.

Figure 7.2 shows a comparison of the pseudorange residuals obtained from the WLS processing the RINEX and the preprocessed raw measurements. It can be seen that the residuals from the first source follow a bi-modal distribution, which is unexpected in the pseudorange measurements, as the noise is expected to be zero-mean.

Finally, the differences between the pseudoranges obtained from the RINEX and the ones obtained by processing the *GnssLog* are computed and it matches with the receiver clock bias computed with the WLS method. Figure 7.3 shows this comparison for a single satellite as an example. This shows that in the generation of the RINEX files from the raw measurements,



**Figure 7.1:** Differences of the preprocessed measurements of a single satellite compared to the RINEX (top) and the GnsAnalysis (bottom). From left to right: pseudorange (meters), carrier phase (cycles) and Doppler (Hz).

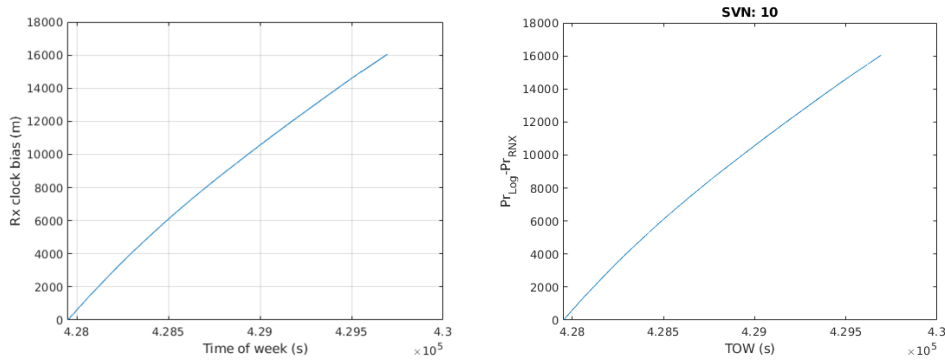


**Figure 7.2:** Distributions of the pseudorange residuals obtained from the RINEX (left) and the preprocessed raw measurements (right).

the receiver clock bias has been corrected in the pseudorange measurements. However, it is not corrected in the carrier phase measurements, since these measurements are almost the same between sources (Fig. 7.1). Although this difference could be solved by keeping two different estimations of the clock bias, it only makes the implementation more complex and prone to errors. For this reason, it was decided to use the GnsLog files as the source for the GNSS measurements.

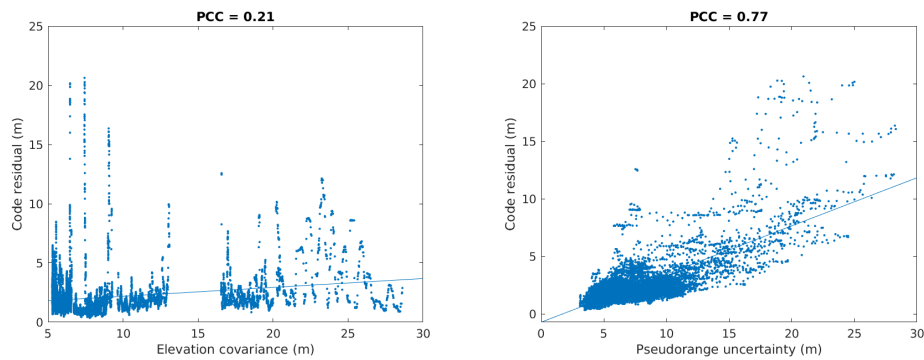
### 7.2.1 Reliability of the measurement uncertainties

As mentioned earlier in this chapter, one of the particularities of the Android Raw GNSS measurements is that an uncertainty of the measurement is provided for each parameter. However, the Android documentation does not detail how these uncertainties are obtained. To determine the reliability of the uncertainties, the standard deviation of the pseudoranges is computed from these uncertainties and it is compared to the pseudorange residuals obtained with the *magnitude* software. Figure 7.4 shows the moving standard deviation of the pseudorange residuals plotted against the moving mean of the standard deviation calculated from the uncertainties, which have

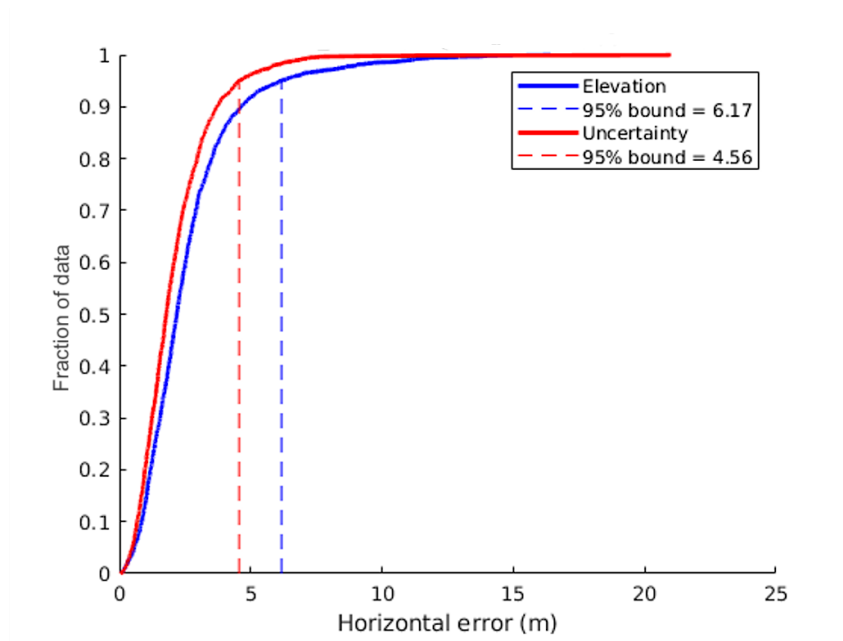


**Figure 7.3:** Receiver clock bias estimated by the WLS (left) and difference of pseudorange between RINEX and preprocessed raw measurements (right).

been computed from 1700 epochs of measurements of 9 satellites. In comparison, the same plot is shown for the standard deviation computed as the inverse of the sine of the elevation. It can be seen that the uncertainty-based standard deviation shows a high correlation with the noise in the measurements, with a Pearson Correlation Coefficient (PCC) close to 0.8. In contrast, the typical elevation-based estimation of the standard deviation shows no correlation with the actual noise in the measurements. Therefore, the uncertainties provided by Android seem to be much more reliable than the elevation-based model for estimating the measurement variance. For further evidence, Fig. 7.5 shows the Cumulative Distribution Functions (CDF) of the horizontal error of the *magnitude*'s KF estimation using the two different pseudorange variance estimations. It can be seen that the uncertainty-based variance provides better results, with a 95% accuracy of 4.56 m versus 6.17 from the elevation-based model. Thus, the measurement covariance matrices of all the methods used in this project are built using the uncertainties provided in the Android GnssLog files.



**Figure 7.4:** Pseudorange residuals vs elevation-based standard deviation (left). Pseudorange residuals vs uncertainty-based standard deviation (right).



**Figure 7.5:** CDF of horizontal error from *magnitude*'s EKF using elevation-based variance (blue) vs uncertainty-based variance (red).



# Chapter 8

## Results

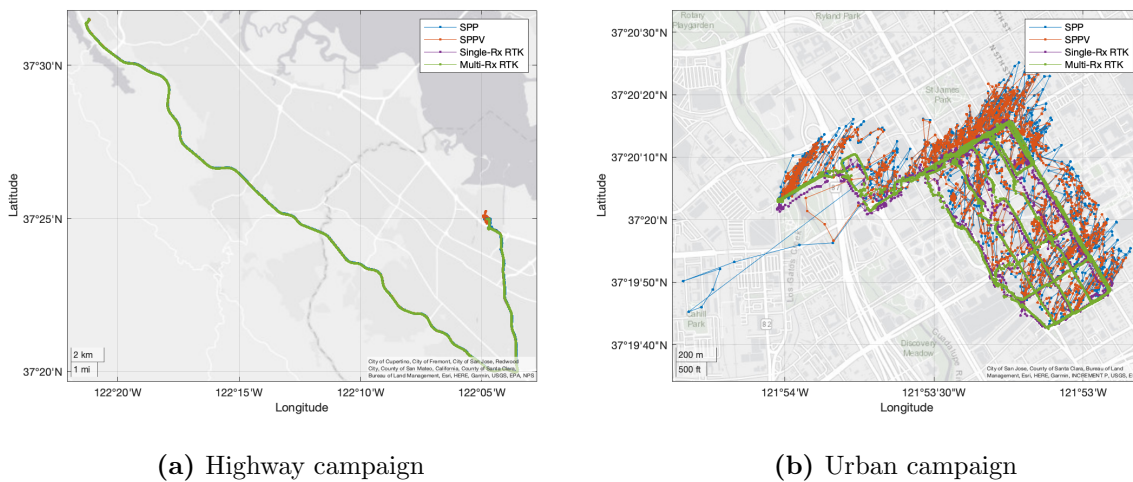
This chapter provides a study of the performance of the positioning methods used in this project. Results from two campaigns are provided as a representation of the two main scenarios found in the 29 *train* campaigns. The first campaign is an open-sky scenario which contains measurements from 3 smartphones (namely Pixel4, Pixel4XL and Pixel4XLModded) taken on a highway of the San Francisco Bay Area. The second campaign is an urban scenario with measurements from 2 smartphones (namely Pixel4 and SamsungS20Ultra) taken in the streets of San Jose, California. Additionally, this chapter also provides a summary of the results obtained in the Google Smartphone Decimeter Challenge.

### 8.1 Comparison between positioning methods

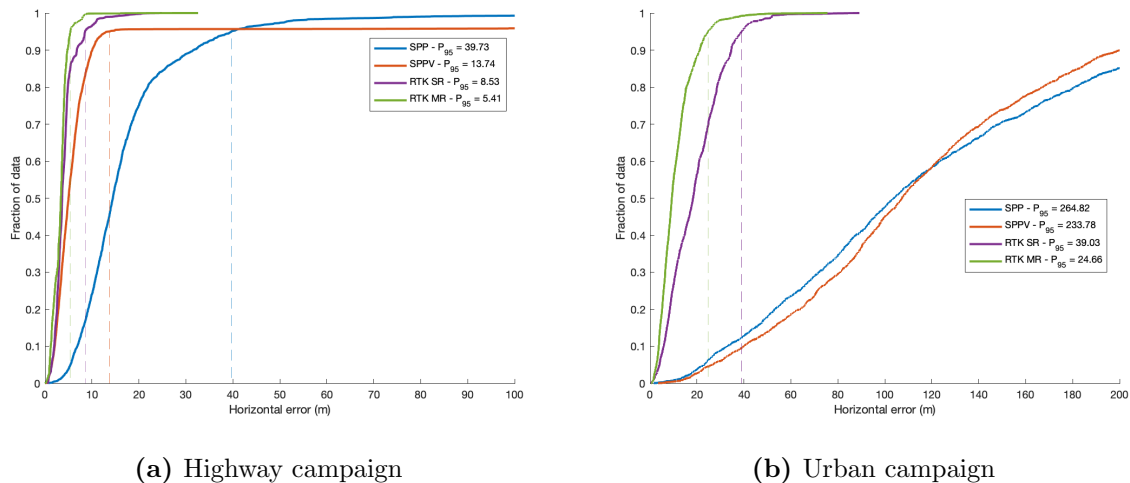
In this section, the performance of the three methods explained in Chapter 6 is compared. Additionally, the results obtained with a conventional SPP EKF are provided in order to observe the effect of using the Doppler measurements. This method consists in the standard EKF which only uses pseudorange measurements to estimate the position (see Section 22.5 in [Teu17]).

The trajectories estimated by these four positioning methods are shown in Figures 8.1a and 8.1b for the highway and the urban scenarios, respectively. In order to study the accuracy of the four different methods that are studied Figure 8.2 shows the Cumulative Distribution Functions (CDFs) and 95 percentiles of the horizontal errors in the highway and the urban campaigns. As it is expected, the accuracy of the positions is much better in the highway campaign, since the measurements are much less affected by multipath. Also, it can be seen that the RTK performs better than the SPP and SPPV methods thanks to the use of the differential corrections. However, it is interesting to see that the main improvement in the highway campaign comes from the inclusion of the Doppler measurements. A hypothesis for

this big improvement is that the inclusion of Doppler measurements in the EKF provide an indirect smoothing of the pseudorange measurements, since the Doppler is less affected by noise. This is specially significant when the measurements are taken by smartphones, since the noise is higher due to the miniaturized and low-cost hardware. On the urban scenario, however, the main improvement comes by far with the RTK method. This is also surprising because one would expect that the error in the urban scenario would be more driven by the multipath, which is not corrected on the differential measurements.



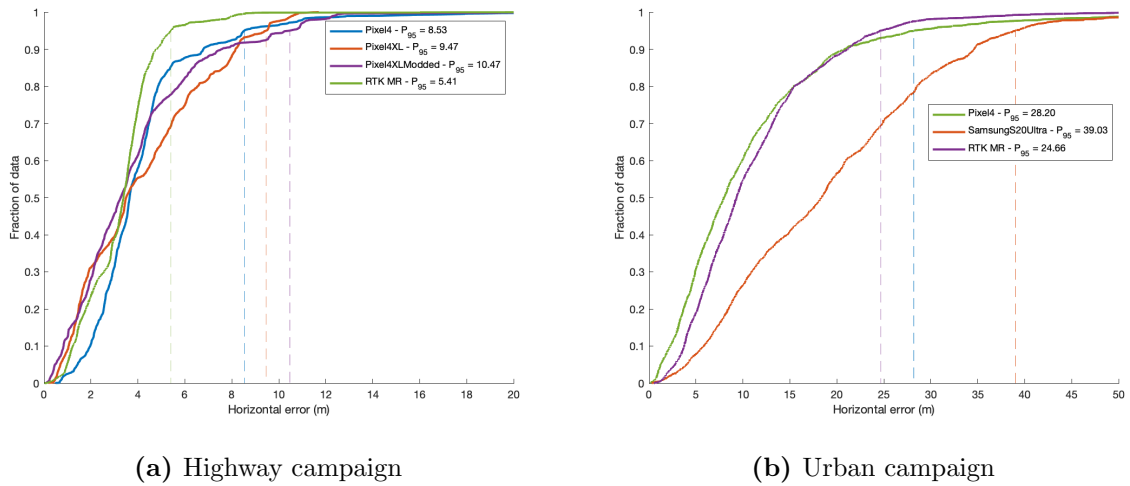
**Figure 8.1:** Estimated positions with SPP (blue), SPPV (red), single-receiver RTK (purple) and multi-receiver RTK (green).



**Figure 8.2:** Horizontal error CDFs of SPP (blue), SPPV (red), single-receiver RTK (purple) and multi-receiver RTK (green).

## 8.2 Accuracy of multi-receiver RTK

In order to study the effect of combining measurements from multiple receivers, the same two campaigns used in the previous section are taken as an example. In both campaigns, the positions of all smartphones are computed using the single-receiver RTK. The results are compared to the multi-receiver RTK, which combines all the measurements of that campaign. Figure 8.3 shows the horizontal errors for the two campaigns mentioned. In both cases, the error at the 95 percentile is lower with the multi-receiver method than using any of the single receivers. However, as it can be seen in the lower percentiles, this better performance of the multi-receiver RTK is not consistent during all the trace. For these tests, the measurement covariances of different smartphones were not weighted differently. However, a different weight could be given to the measurements of each smartphone according to the observed accuracy in the single-receiver method or on the measurements residuals, which could improve the performance of the multi-receiver RTK.



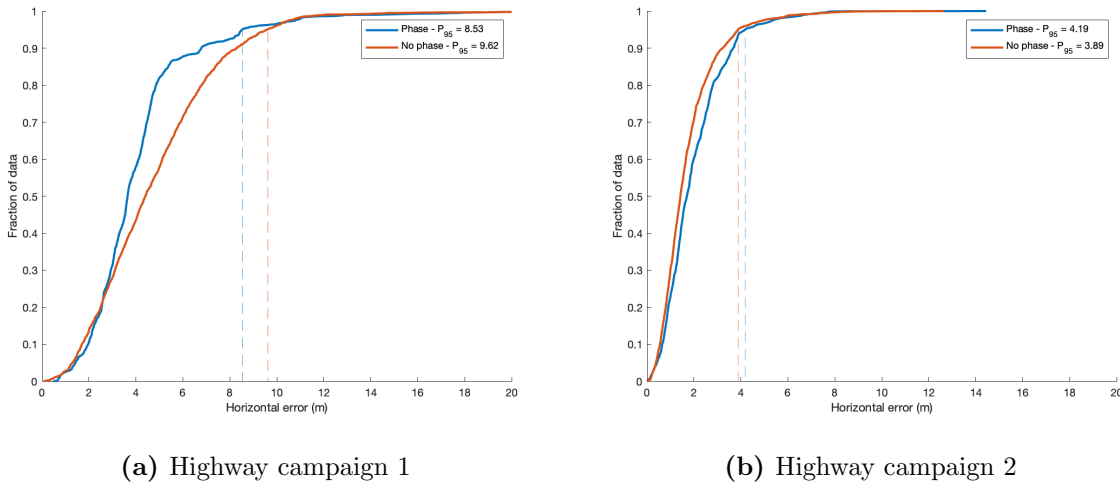
**Figure 8.3:** Comparison between single-receiver (blue, red, purple) and multi-receiver (green).

## 8.3 Use of carrier phase measurements

As explained in Chapter 6, carrier phase measurements are less affected by noise and thus are extensively used for precise positioning applications such as RTK. However, their use adds complexity due to the required estimation of ambiguities. Furthermore, these estimates are affected by the loss of lock of the receiver. Therefore, a good stability of the carrier phase tracking is desired, which usually requires more expensive hardware and external antennas. For this reason, carrier phase measurements on smartphones are not as reliable as they are when taken from dedicated GNSS receivers. In this section, the use of smartphone carrier phase

measurements on RTK is studied. To do so, the single-receiver RTK described Chapter 6 is compared to a *No-phase RTK* method that follows the same model as the single-receiver RTK of Chapter 6 for the pseudoranges and the pseudorange rates but which does not consider the carrier phase measurements.

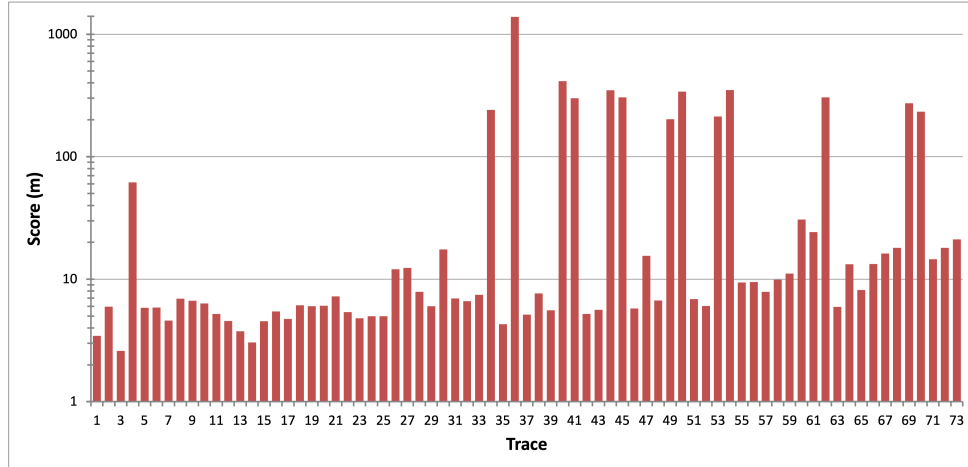
Figure 8.4a shows the horizontal errors for the single-receiver RTK and the *No-phase RTK* on the same highway campaign of the previous sections. It can be seen that the accuracy of the estimates improves when including carrier phase measurements, reducing the 95 percentile by more than a 10%. However, this is not always the case. In Figure 8.4b, the same result is shown for a different highway campaign. This time, the use of carrier phase measurements increases the 95 percentile of the horizontal error by about 7%.



**Figure 8.4:** Horizontal error CDFs of *No-phase RTK* (blue), and RTK (red).

## 8.4 Results of Google Smartphone Decimeter Challenge

The results obtained with the single-receiver methods described in this project (namely SPPV and RTK) have shown a largely varying accuracy on different traces. It has been observed that the evaluation of some traces provided errors of several hundreds of meters while other traces from the same campaign showed errors on the order of the few meters. However, these large errors were not always associated to the same smartphone model. Figure 8.5 show the scores obtained from evaluating all the traces of the training dataset with the single-receiver method. It can be seen that the majority of traces result in a score below 10 meters, however the score of 14 traces is over 50 meters. An analysis has been made to try to identify the causes of the worst scores, however they have not been identified and this problem has been left for future work. In order to continue with the development of this project, the measurements that resulted in scores over 50 meters have been discarded, and measurements from another receiver in the same



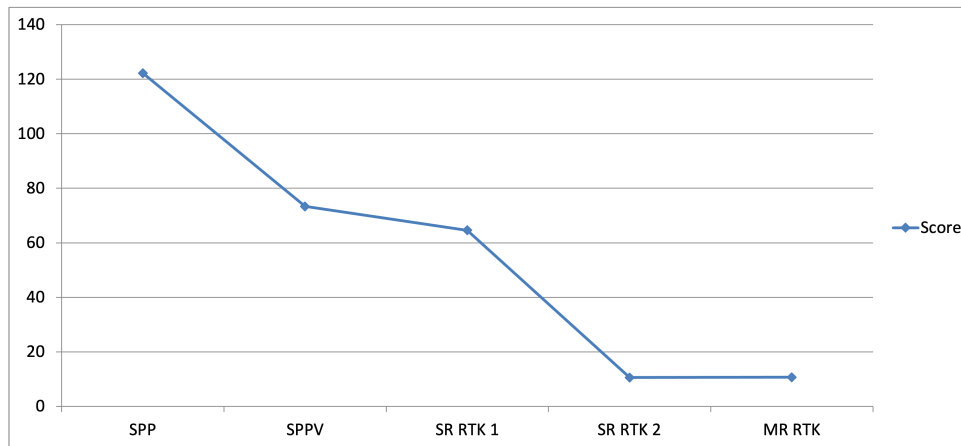
**Figure 8.5:** Scores obtained on all traces of the training set.

campaign have been used instead.

The individual scores obtained on each trace can not be obtained for the test dataset because the ground-truths are not provided. In order to improve the score on the Google Smartphone Decimeter Challenge, the results from the single-receiver RTK have been plotted over a map. These results have been visually inspected and the measurements that provided bad performances have been discarded when using the multi-receiver RTK.

When using the multi-receiver RTK on the train dataset, the geometry of the smartphones is computed from the ground-truth. However, this information is not available on the test dataset. To solve this, our team assumed that this geometry would not change between campaigns that were performed in close days and that contained the same smartphone models. Then, the geometries computed in the train dataset were assigned to the test campaigns by comparing the dates and smartphones used in each campaign. However, this assumption has turned out to be wrong. Although the multi-receiver RTK improved the accuracy of the estimations on the train dataset, the score obtained on the test dataset did not improve but became a few decimeters worse than the single-receiver RTK. In order to solve this problem, a model could be implemented which uses the differential GNSS measurements from multiple smartphones to estimate the geometry. Then, this method could be added as a previous step to the multi-receiver RTK.

Figure 8.6 shows the evolution of the scores obtained on the test dataset. The first method consists on the SPP method which only uses pseudorange measurements. The second method is the SPPV explained in Chapter 6. Both the third and the fourth methods refer to the single-receiver RTK. In the first of the two, all the estimates of each trace are computed from the measurements of that trace. In the second one, the results are evaluated and the traces that result in a visually bad accuracy are substituted by the results from another trace in the same



**Figure 8.6:** Final scores obtained with different techniques on the test dataset.

campaign. Finally, the last method refers to the multi-receiver RTK, for which the measurements that are discarded in the single-receiver are also not used.

## Chapter 9

# Conclusion

This internship has been mainly focused on the participation in the Google Smartphones Decimeter Challenge. The competition has served to research on techniques to provide precise positioning using Android Raw GNSS measurements. Three different techniques have been implemented for this project. The first technique uses a Kalman Filter to combine pseudorange and Doppler measurements, which provides an estimation of the position and velocity of the receiver. It has been shown that the inclusion of Doppler measurements notably improves the accuracy of the estimates. The second technique consists of an RTK with float ambiguity resolution, which also includes non-differential Doppler measurements to estimate the velocity. The use of differential GNSS measurements has shown an improvement in the performance of the estimation. However, it has been seen that the inclusion of carrier phase measurements does not always provide better results. Finally, the third technique combines simultaneous measurements from different smartphones in order to improve the performance of the RTK and also provide an estimation of the attitude. This technique assumes that the relative positions of the smartphones in the vehicle are known. This, however, is not the case for the test dataset used for evaluation in this competition. Then, the geometries obtained in the train dataset have been used with the test dataset. This extrapolation has not provided the improve in performance seen in with the train datasets, which indicates that the positions of the smartphones in the car changed between campaigns.

In this project, some subjects have been left for future research due to the limited time of the competition. Firstly, it would be interesting to further research on the nature of the Android GNSS measurements, specially on the use of the carrier phase since it has not provided the improvement in performance that can be observed in dedicated GNSS receivers. Secondly, the use of a non-diagonal covariance measurement matrix on the RTK methods should improve the performance of the estimates. Although the workaround implemented in this project still provides acceptable accuracies, it would be more rigorous to include the non-diagonal terms and provide

a better noise model to the Kalman Filter. Finally, the hybridization of the RTK methods with INS and Magnetometer measurements would reduce the covariance of the estimates because a more accurate transition model would be used. This could notably improve the accuracy of the estimates.







# Acronyms

**ADR** Accumulated Delta Range.

**AGC** Automatic Gain Control.

**CDF** Cumulative Distribution Function.

**CMC** Code-Minus-Carrier.

**DD** Double Difference.

**DGNSS** Differential GNSS.

**ECEF** Earth-Centered, Earth-Fixed.

**EKF** Extended Kalman Filter.

**ENU** East-North-Up.

**FLL** Frequency Lock Loop.

**GNSS** Global Navigation Satellite System.

**GPS** Global Positioning System.

**IGS** International GNSS Service.

**INS** Inertial Navigation System.

**KF** Kalman Filter.

**LAMBDA** least-squares ambiguity decorrelation adjustment.

**LOS** Line-Of-Sight.

**LTP** Local Tangent Plane.

**OSR** Observation Space Representation.

**PCC** Pearson Correlation Coefficient.

**PLL** Phase Lock Loop.

**PPP** Precise Point Positioning.

**RTK** Real-Time Kinematic.

**SD** Single Difference.

**SPP** Single-Point-Position.

**SPPV** Single-Point-Position-and-Velocity.

**SSR** Satellite Space Representation.

**WLS** Weighted Least Squares.

# Bibliography

- [Agh09] Farhad Aghili, Alessio Salerno, “Attitude determination and localization of mobile robots using two RTK GPSs and IMU”, *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pags. 2045–2052, oct. 2009, iSSN: 2153-0866.
- [Agh10] Farhad Aghili, Alessio Salerno, “3-D Localization of mobile robots and its observability analysis using a pair of RTK GPSs and an IMU”, *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pags. 303–310, jul. 2010, iSSN: 2159-6255.
- [Dev21] Android Developers, “Raw GNSS Measurements”, <https://developer.android.com/guide/topics/sensors/gnss>, 2021.
- [Eur17] European GNSS Supervisory Authority., *Using GNSS raw measurements on Android devices: white paper.*, Publications Office, LU, 2017.
- [Fu20] Guoyu Michael Fu, Mohammed Khider, Frank van Diggelen, “Android raw gnss measurement datasets for precise positioning”, *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pags. 1925–1937, 2020.
- [GMV11] GMV, “Navipedia. real Time Kinematics”, <https://gssc.esa.int/navipedia/index.php/Real\Time\Kinematics>, 2011.
- [Gro08] Paul D Groves, *Principles of GNSS, Inertial, and Multi-sensor Integrated Navigation Systems*, Vol. 39, Artech House, 1<sup>st</sup> ed., 2008.
- [Hen13] Patrick Henkel, Christoph Günther, “Attitude determination with low-cost gps/ins”, *26th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2013*, Vol. 3, pags. 2015–2023, 01 2013.
- [Hen14] Patrick Henkel, Michele Iafrancesco, “Tightly coupled position and attitude determination with two low-cost GNSS receivers”, *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pags. 895–900, aug. 2014, iSSN: 2154-0225.
- [Hu20a] Xiao Hu, Paul Thevenon, Christophe Macabiau, “Cycle Slip Detection and Repair Using an Array of Receivers with Known Geometry for RTK Positioning”, *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pags. 1123–1134, IEEE, Portland, OR, USA, apr. 2020.
- [Hu20b] Xiao Hu, Paul Thevenon, Christophe Macabiau, “Improvement of rtk performances using an array of receivers with known geometry”, *Proceedings of the 2020 International Technical Meeting of The Institute of Navigation*, pags. 440–453, 2020.

- [Hu21] Xiao Hu, Paul Thevenon, Christophe Macabiau, “Attitude determination and rtk performances amelioration using multiple low-cost receivers with known geometry”, *Proceedings of the 2021 International Technical Meeting of The Institute of Navigation*, pags. 439–453, 2021.
- [Kap06] Elliott D. Kaplan, C. Hegarty (eds.), *Understanding GPS: principles and applications*, Artech House mobile communications series, Artech House, Boston, 2<sup>nd</sup> ed., 2006, oCLC: ocm62128065.
- [Klo87] John A. Klobuchar, “Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users”, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-23, no. 3, pags. 325–331, may 1987, conference Name: IEEE Transactions on Aerospace and Electronic Systems.
- [Med20] Daniel Medina, Jordi Vilà-Valls, Anja Hesselbarth, Ralf Ziebold, Jesús García, “On the Recursive Joint Position and Attitude Determination in Multi-Antenna GNSS Platforms”, *Remote Sensing*, Vol. 12, no. 12, pags. 1955, jun. 2020.
- [Mis06] P. Misra, P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, Ganga-Jamuna Press, 2<sup>nd</sup> ed., 2006.
- [Pet79] Peter S. Maybeck, *Stochastic models, estimation and control. Volume 1*, Vol. 1, Academic press, New York, 1<sup>st</sup> ed., 1979.
- [Saa72] J. Saastamoinen, “Atmospheric Correction for the Troposphere and Stratosphere in Radio Ranging Satellites”, *The Use of Artificial Satellites for Geodesy*, Vol. 15, pags. 247–251, American Geophysical Union (AGU), 1972, eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/GM015p0247>.
- [Sim06] Dan Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*, Wiley-Interscience, Hoboken, N.J, 2006, oCLC: ocm64084871.
- [Teu17] Peter J.G. Teunissen, Oliver Montenbruck (eds.), *Springer Handbook of Global Navigation Satellite Systems*, Springer International Publishing, Cham, 2017.