UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

Northeastern University
College of Engineering

Cooperative Positioning using Massive Differentiation of GNSS Pseudorange Measurements

Master Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Helena Calatrava

In partial fulfillment
of the requirements for the
MASTER'S DEGREE IN TELECOMMUNICATIONS ENGINEERING

Advisors: Pau Closas, Alba Pagès
Boston and Barcelona, September 2022

telecos
BCN

Contents

List of Figures 4

List of Tables 5

Notation 10

1 Introduction 11
1.1 General Information . . . 11
1.2 Thesis Outline . . . 11
1.3 Problem Statement . . . 12
1.4 State of the Art on Cooperative Positioning . . . 12
1.4.1 Hybrid Positioning Techniques . . . 12
1.4.2 Crowdsensing and Privacy Protection . . . 15
1.4.3 GNSS Cooperative Positioning . . . 16
1.4.4 Inter-Device Communication Systems . . . 17

2 Background on GNSS Code-Based Positioning 19
2.1 Introduction to GNSS . . . 19
2.2 Single Constellation Standard Positioning Point (SPP) Solution . . . 22
2.2.1 Pseudorange Measurements Model . . . 22
2.2.2 Geometric Range Linearization . . . 23
2.2.3 Observation Model . . . 25
2.2.4 Iterative LSE . . . 25
2.2.5 Dilution of Precision . . . 26
2.2.6 Weighted Least Squares Estimator . . . 27
2.3 Classic Single Differentiation of Pseudorange Measurements . . . 27
2.3.1 Differential GNSS (DGNSS) . . . 27
2.3.2 Mathematical Derivation . . . 28
2.4 Contribution and Hypothesis . . . 29
2.4.1 Novelty . . . 30

3 Proposed GNSS Cooperative Positioning Algorithm 31
3.1 Motivation of the UCSD and MUCSD algorithms . . . 31
3.2 User-Centric Single Differentiation (UCSD) . . . 32
3.2.1 Observation Model . . . 32
3.2.2 Error Covariance Matrix . . . 34
3.2.3 CRB . . . 34
3.2.4 Iterative LSE . . . 34
3.2.5 Motivation for the Extension to N Cooperative users . . . 35
3.3 Massive User-Centric Single Differentiation (MUCSD) . . . 35
3.3.1 Observation Model . . . 35
3.3.2 Error Covariance Matrix . . . 35
3.3.3 CRB . . . 37
3.3.4 Iterative LSE . . . 38

4 Methods **41**
4.1 General Setup Description . . . 41
4.1.1 Use of NEU's Discovery Cluster . . . 41
4.2 MassiveCoop-Sim Simulator . . . 42
4.2.1 GNSS Configuration . . . 42
4.2.2 User Distribution . . . 42
4.2.3 Mask Angle filtering . . . 46
4.2.4 Generation of Pseudorange Measurements . . . 47
4.2.5 Position Estimation . . . 48
4.2.6 Benchmark: CRB Calculation . . . 50
4.3 Experimental Setup . . . 51
4.3.1 Performance Evaluation Metric (RMSE) . . . 51

5 Results and Discussion **52**
5.1 Error Covariance Matrix . . . 52
5.1.1 UCSD ($\mathbf{\Sigma}_{m,n}$) . . . 52
5.1.2 MUCSD ($\mathbf{\Sigma}$) . . . 53
5.2 Position Estimation RMSE . . . 53
5.2.1 Experiment A . . . 54
5.2.2 Experiment B . . . 55
5.2.3 Experiment C . . . 59
5.2.4 Interpretation of results via CRB inspection . . . 60

6 Conclusion **61**
6.1 Final Results . . . 61
6.2 Future Research . . . 61
6.2.1 Use of Carrier Phase Measurements and Double Differentiation) . . . 62
6.2.2 Computational Efficiency Evaluation . . . 62
6.2.3 Centralization of the System . . . 63
6.2.4 Selection of a Communication Protocol . . . 63

References **64**

List of Figures

1 Diagram illustrating the concept of hybrid positioning using GNSS and 5G mmWave signals. . . . 13
2 Scenario with satellite blocking in urban canyons. . . . 14
3 Challenges posed by LOS path obstruction and multipath propagation in GNSS-only systems. . . . 15
4 Illustration of the factors contributing to the rise of Mobile Crowdsensing (MCS). . . . 16
5 Example scenario of the proposed CP method using GPS in [1]. . . . 18
6 Illustration of the rationale behind GNSS positioning and the effect of unknown receiver clock offset. . . . 20
7 GPS, Galileo and GLONASS navigational frequency bands. . . . 21
8 General architecture of a standard GNSS receiver. . . . 21
9 Direction of the Line of Sight (LOS) vector. . . . 23
10 Scenario of a system using DGNSS, UCSD or MUCSD. . . . 31
11 Location of the firstly generated user. . . . 43
12 Distribution of 101 users within the defined spherical space. . . . 44
13 Distribution of 1001 users within the defined spherical space. . . . 45
14 Distribution of 10001 users within the defined spherical space. . . . 46
15 Distribution of inter-user distances with respect to the firstly generated user. 47
16 Inter-user distances 2D norms. . . . 48
17 Inter-user distances 3D norms. . . . 49
18 Monte Carlo integration scheme. . . . 50
19 UCSD Error Covariance matrix for different values of cooperative user position error. . . . 52
20 Contribution of the pseudorange code error to the MUCSD error covariance matrix. . . . 53
21 Contribution of the pseudorange code error and the cooperative user position error to the MUCSD error covariance matrix. . . . 54
22 RMSE results obtained with Experiment A. . . . 55
23 RMSE results obtained with Experiment B.1. . . . 57
24 RMSE results obtained with Experiment B.2. . . . 58
25 RMSE results obtained with Experiment C. . . . 59

List of Tables

1 Error sources that techniques using Single Differentiation (SD) of pseudo-range measurements aim to mitigate, potential error size in meters and mitigation results. . . . 28
2 Estimators proposed in this research with the UCSD and MUCSD algorithms. 32
3 Experiments conducted to test the performance of the UCSD and MUCSD algorithms. . . . 32
4 Computational resources used for the execution of our .m scripts in the Discovery cluster from Northeastern University, Boston [2]. . . . 42
5 Main modules of the MassiveCoop-Sim simulator, which takes as a reference the simulator provided in [3]. . . . 42
6 Summary of the setup of the three experiments conducted in this research. 51
7 Experiment-dependent parameter values set for Experiment A. . . . 55
8 Experiment-dependent parameter values set for Experiment B.1. . . . 57
9 RMSE results obtained with Experiment B.2. . . . 58
10 Experiment-dependent parameter values set for Experiment B.2. . . . 58
11 Experiment-dependent parameter values set for Experiment C. . . . 60
12 Main conclusions extracted from the experiments conducted to test the performance of the UCSD and MUCSD algorithms. . . . 62

Revision History and Approval Record

Revision	Date	Purpose
0	15/04/2022	Document creation
1	25/05/2022	Document revision (Introduction and Background, by author)
3	15/07/2022	Document revision (Results, by author)
4	22/08/2022	Document revision (whole document, by Alba Pagès Zamora)
5	30/08/2022	Document revision (whole document, by Pau Closas)

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Helena Calatrava	helena.calat@gmail.com
Pau Closas	closas@northeastern.edu
Alba Pagès Zamora	alba.pages@upc.edu

Written by:		Reviewed and approved by:	
Date	15/04/2022	Date	01/09/2022
Name	Helena Calatrava	Name	Alba Pagès
Position	Project Author	Position	Project Supervisor

Abstract

With Differential GNSS (DGNSS), Single Differentiation (SD) of GNSS pseudorange measurements is computed with the aim of correcting harmful errors such as ionospheric and tropospheric delays. These errors can be mitigated to up to very few centimeters, which denotes a performance improvement with respect to the Standard Point Positioning (SPP) solution, widely used in GNSS receivers. However, with DGNSS it is necessary to have a very precise knowledge of the coordinates of a reference station in order to experience this performance improvement. We propose the Massive User-Centric Single Differentiation (MUCSD) algorithm, which is proven to have a comparable performance to DGNSS without the need of a reference station. Instead, N cooperative receivers which provide noisy observations of their position and clock bias are introduced in the model. The MUCSD algorithm is mathematically derived with an Iterative Weighted Least Squares (WLS) Estimator. The estimator lower bound is calculated with the Cramér-Rao Bound (CRB). Several scenarios are simulated to test the MUCSD algorithm with the MassiveCoop-Sim simulator. Results show that if the observations provided by the cooperative users have a noise of up to 10 meters, DGNSS performance can be obtained with $N = 10$. When observations are very noisy, the MUCSD performance still approaches DGNSS for high values of N.

Acronyms

3GPP	3rd Generation Partnership Project
5G	5th Generation
ARNS	Aeronautical Radio Navigation Service
BLE	Bluetooth Low Energy
CP	Cooperative Positioning
CRB	Cramér-Rao Bound
CS	Crowdsensing
CVS	Cooperative Vehicle Safety
DD	Double Difference
DGNSS	Differential Global Navigation Satellite Systems
DOP	Dilution of Precision
DSRC	Dedicated Short-Range Communications
ECEF	Earth-Centered, Earth-Fixed (Coordinates)
ENU	Local East, North, Up (Coordinates)
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
ITS	Intelligent Transportation Systems
IVRs	Inter-Vehicle Ranges
LOS	Line of Sight
LS	Least Squares
LSE	Least Squares Estimator
LTE	Long Term Evolution
MCS	Mobile Crowdsensing
MIMO	Multiple-input Multiple-output
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimator
MT	Mobile Terminal
NED	Local North, East, Down (Coordinates)

O-C	Observed minus Computed (measurement prefit)
PDF	Probability Density Function
PPP	Precise Point Positioing
RFID	Radio Frequency Identification
RNSS	Radio Navigation Satellite Systems
RSS	Received Signal Strength
RTCM	Radio Technical Commission for Maritime Services
RTK	Real Time Kinematic
SD	Single Difference
SNR	Signal-to-Noise Ratio
SPP	Standard Point Positioning
TDoA	Time Difference of Arrival
ToA	Time of Arrival
UERE	User Equivalent Range Error
UHF	Ultra High Frequency
V2X	Vehicle-to-Everything
WARTK	Wide Area Real Time Kinematic
WAVE	Wireless Access in Vehicular Environment
Wi-Fi	Wireless-Fidelity
WLAN	Wireless Local Area Network
WLS	Weighted Least Squares
WSN	Wireless Sensor Network

Notation

$\hat{x}$	Estimation of parameter x.
$f(x)\|_{x=a}$	Function $f(x)$ evaluated at $x = a$.
$\|x\|$	Absolute value (modulus) of scalar x.
$\|\mathbf{x}\|$	ℓ^2-norm of vector $\mathbf{x}$.
$\det(\mathbf{X})$	Determinant of matrix $\mathbf{X}$.
$\mathbf{I}_N$	Identity matrix with dimension $N \times N$.
$\mathbf{J}_{N\times M}$	Matrix of ones or all-ones matrix with dimension $N \times M$.
$\mathbf{X}^\mathsf{T}$	Transpose of matrix $\mathbf{X}$.
$\mathbf{X}^\dagger$	Moore-Penrose pseudoinverse of matrix $\mathbf{X}$.
$\otimes$	Kronecker or tensor product.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
$\mathcal{U}(\mu, \Sigma)$	Continuous uniform distribution with mean μ and covariance matrix $\boldsymbol{\Sigma}$.
$\mathbb{E}[\cdot]$	Statistical expectation.
$\mathcal{I}(\boldsymbol{\theta})$	Fisher Information Matrix (FIM) with respect to the parameter vector $\boldsymbol{\theta}$.
$\ln(\cdot)$	Natural logarithm (base e).
$\frac{\partial f(\mathbf{x})}{\partial x_i}$	Partial derivative of function $f(\mathbf{x})$ with respect to the variable x_i.
$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$	Gradient of function $f(\mathbf{x})$ with respect to the vector $\mathbf{x}$.
i.i.d.	Independent and Identically Distributed
r.v.	Random Variable
s.t.	Subject to

Boldface upper-case letters denote matrices and boldface lower-case letters denote column vectors.

1 Introduction

The introductory section of this thesis aims to first describe the thesis outline in order to provide a clear understanding on what is explained throughout the document and which topics are covered. The state-of-the-art section covers the topic of Cooperative Positioning (CP) using Global Navigation Satellite Systems (GNSS) and other technologies. As the background section contains very specific content that is highly linked to the conducted research, we have decided to separate it from the introductory section. Please, refer to Section 2 for an introduction to the mathematical derivations behind GNSS code-based positioning.

1.1 General Information

This thesis is being submitted in partial fulfillment of the requirements for the Master's degree in Telecommunications Engineering (MET) at Universitat Politècnica de Catalunya (UPC), Barcelona School of Telecommunications Engineering (ETSETB). The research has been conducted as part of an international mobility program at the Signal Processing Imaging Reasoning and Learning (SPIRAL) Laboratory (Department of Electrical and Computer Engineering, College of Engineering) of Northeastern University in Boston (United States, Massachusetts).

As it is mentioned in Section 4, it has been required to use the Discovery Cluster from Northeastern University due to the high computational cost of some of the experiments conducted with MATLAB R2021b. A more detailed description of this matter is later provided to the reader.

The domain of this thesis is cooperative device localization in GNSS-only systems.

1.2 Thesis Outline

At the beginning of the document, the list of the figures and tables is included and followed by the notation description used in the mathematical derivations found in Sections 2, 3 and 4.

Within Section 1, and after the thesis outline, a review of current literature on CP using several technologies, not only GNSS, is included. The challenges posed by data sharing between a massive number of users, such as data protection, privacy and high computational cost, are described.

In Section 2, an extensive and detailed background on the core ideas of GNSS code-based positioning is provided, focusing on the single-constellation Standard Positioning Point (SPP) solution and the use of Single Differentiation (SD) between receivers in Differential GNSS (DGNSS). This section allows the reader to gain in-depth knowledge to read Section 3, and also to understand the hypothesis and contribution of this research in more detail. Consequently, both the hypothesis and contribution are described at the end of Section 2.

In Section 3, the proposed User-Centric Single Differentiation (UCSD) and Massive User-

Centric Single Differentiation (MUCSD) algorithms are mathematically derived. Section 4 shows the setup used to conduct the final experiments of this research. Our simulator, which we refer to as MassiveCoop-Sim, is inspired by the simulator in [3] and is described in detail. Some parameter values that are part of the experimental setup are experiment-dependent. Therefore, details regarding these parameters are directly provided in Section 5, where the thesis results are described

The performance of the proposed techniques is evaluated in Section 5. The discussion is focused on comparing the UCSD and MUCSD algorithms to the well-known and widely used DGNSS technique.

Lastly, conclusions are drawn from the obtained results in Section 6. A list of possible (and motivating) future lines of research is provided at the very end of the document.

1.3 Problem Statement

Please refer to Section 2.4 to read about the purpose, hypothesis and contribution of this research. Section 2 allows the reader to gain the in-depth knowledge required to understand the problem statement in detail.

1.4 State of the Art on Cooperative Positioning

In this section we would like to emphasize the importance of hybrid positioning and also the importance of Cooperative Positioning (CP). Hybrid positioning is defined as the use of more than one technology with positioning purposes, while CP is defined as the use of measurements from multiple users when conducting a position estimation task. The availability of measurements from several users brings a new dimension of information into positioning algorithms and this is the reason why it is such a powerful tool. However, these algorithms pose challenges given their high computational cost and the loss of privacy in some applications.

Radio signals have been used with localization purposes for a long time, being the most prominent used technologies the Global Positioning System (GPS), cellular systems and Wireless Fidelity (Wi-Fi) [4]. In general, localization encompasses two steps. The first step consists on the processing of user measurements to obtain distance or angle information. The second step usually involves triangulation to determine the position of a user [4].

1.4.1 Hybrid Positioning Techniques

A wide range of current (and future) applications demand high positioning accuracy. This is the case of several 5G use cases, such as the ones that are mission-critical (public safety, emergency calls), location-based (augmented reality, wearables, vehicle sharing, tracking systems), Intelligent Transportation Systems (ITSs) (traffic monitoring, drone tracking, autonomous driving, V2X), eHealth-related (patient tracking inside and outside hospitals, location of emergency equipment) and Industry 4.0 applications [5]. The support of localization services in 5G communications is considered a separate use case that can be managed by relying on multiple technologies. In [5], they separate between three categories

of 5G localization services: the ones using 3GPP technologies exclusively, the ones using non-3GPP technologies exclusively and, lastly, the combination of the latter.

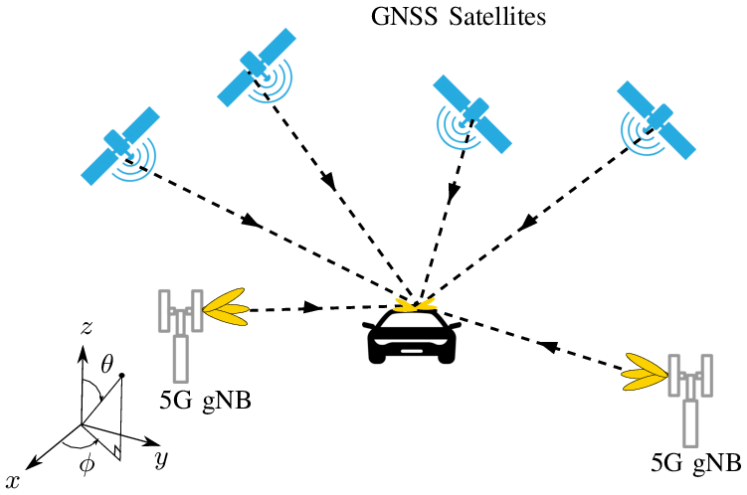

Figure 1: Diagram illustrating the concept of hybrid positioning using GNSS and 5G mmWave signals, from [6]. The users (vehicles) receive GNSS signals from the satellites and 5G signals from the 5G gNBs (radio nodes).

Wi-Fi [7], ZigBee, Radio Frequency Identification (RFID) and Bluetooth Low Energy (BLE) are some of the technologies used in hybrid positioning techniques and they are currently available for Wireless Local Area Networks (WLANs), Wireless Sensor Networks (WSN) and Internet of Things (IoT) applications. It is very interesting that, although they were initially designed with communication purposes only, they are now being used for localization as well. These technologies do not provide positioning capabilities by themselves, but the signals they transmit, if properly processed, offer different levels of localization accuracy [5]. For instance, accuracies at meter-level can be achieved using Wi-Fi fingerprinting techniques [7] [8]. This explains why Wi-Fi and BLE have been considered complementary localization technologies in LTE (Release 13). Their wide diffusion offers the possibility to enhance indoor positioning, which is specially challenging.

It is relevant to highlight that each localization method has its own benefits and drawbacks. Besides, accuracy levels depend on the technology we rely on [8]. Consequently, combining technologies, which is what hybrid techniques do, can be a good strategy. For example, GNSS positioning methods such as the Precise Point Positioning (PPP) and Real Time Kinematic (RTK) techniques provide centimeter-level accuracy when enough satellites are in view. The need of having enough satellites in view implies that GNSS systems perform better in outdoor conditions, where GNSS receivers are able to properly decode navigation messages. However, GNSS can also be of help in hybrid positioning systems that work both in outdoor and indoor environments [9]. Interestingly, numerous studies in current literature suggest that the integration of GPS and UWB is very powerful for indoor positioning [10]. This combination can also help in outdoor and indoor mixed scenarios [11].

As stated in [6], the GNSS performance drop due to the obstruction of Line of Sight (LOS)

paths (in urban canyons or underground tunnels, for instance) is generally overcome by augmenting GNSS systems with other dedicated subsystems like the one provided by future 5G mmWave networks. Hybridization of GNSS and 5G positioning systems is studied in current literature. The concept of hybrid positioning using GNSS and 5G signals is illustrated in Figure 1. In this scheme, the users (vehicles), receive information from the GNSS system and also from the radio nodes of the 5G cellular network.

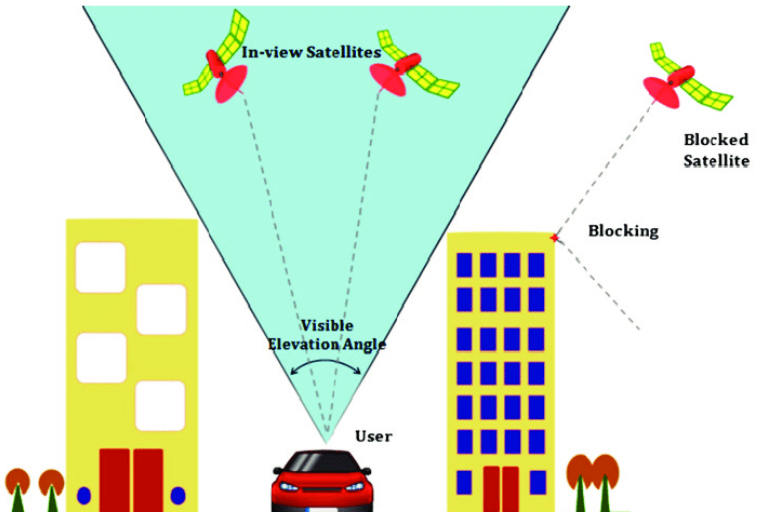

Figure 2: Scenario with satellite blocking in urban canyons, from [12]. Satellite signals that are outside the user visible elevation angle (blocked signals) are not received by the user and consequently the satellite sending that signal is not considered to be in view.

We would like to stress on the concept of satellite in view. The obstruction of LOS paths mentioned beforehand usually occurs due to not having enough in view satellites in the scenario, i.e., satellites sending signals within the user/receiver visible elevation angle. In Section 2, we justify the fact that four satellites in view are needed in order to properly estimate the position of a user. In Section 4, the concept of elevation angle is introduced. Figure 2 shows the diagram of an example scenario with two satellites in view and one blocked satellite. This diagram has been extracted from [12], where GNSS-only absolute positioning is studied under different sky view conditions. The challenge posed by the obstruction of LOS path (or NLOS) conditions is also illustrated in Figure 3 (b).

In urban environments, multipath can dramatically increase the positioning error in GNSS-only systems. Multipath is originated by reflected paths present in challenging scenarios such as urban canyons. This is illustrated in Figure 3 (a). There is a wide choice of literature on multipath and NLOS error mitigation, which can be targeted with hybrid positioning.

Multipath and NLOS errors are not an object of study in this research, as we are assuming that a minimum number of satellites in view is available in the scenario. However, we considered relevant to introduce the need of hybridization, as the technique that is proposed in this Master Thesis could benefit from it in further lines of research. This way, we could protect the receiver when not being under open-sky conditions.

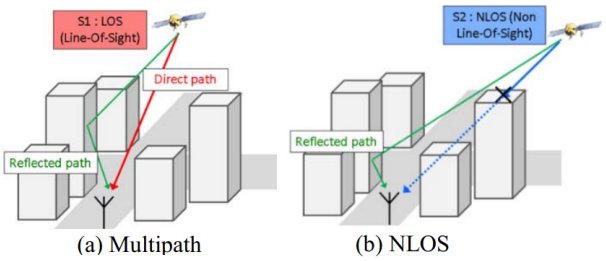

Figure 3: Challenges posed by LOS path obstruction and multipath propagation in GNSS-only systems, from [13]. In (b) the satellite is not in view and consequently the signal is not received within the user visible elevation angle, while in (a) the satellite is indeed in view but multipath effects increase position estimation error.

1.4.2 Crowdsensing and Privacy Protection

Crowdsensing (CS) is introduced in this section because it poses important challenges that will need to be addressed in future applications that require accurate knowledge of device location, for which GNSS Cooperative Positioning (CP) will be of great help. With CS, information is distributed across a large number of users rather than by a set of sensors in fixed locations. Current work focuses on Mobile CS, which assumes that a large group of individuals having devices that are capable of sensing will agree to collectively share their data [14]. In Figure 4, the factors contributing to the rise of Mobile CS are shown. In crowdsensing and crowdsourcing, data is collected collaboratively by exploiting the massive presence of geolocated devices that are connected to the Internet [15]. The introduction of IoT, crowdsensing, crowdsourcing and cloud computing is having important effects that promote innovation in applications that are part of the Industry 4.0.

Although advantageous, Mobile CS poses several challenges such as the limitation of resources like energy, bandwidth and computational power. For instance, if wanting to share GPS locations of several users in an outdoor scenario, the battery of the devices being used will probably drain. To fix this, Wi-Fi can be used for tracking, but the shared location will be less accurate [14]. It is relevant to highlight the high computational cost that cooperative algorithms have. This is the reason why several publications in the literature are trying to assess how critical is the loss of Quality of Service (QoS) due to the continuous generation of large amount of data in, for example, IoT applications using Mobile CS [16].

It is interesting to talk about CS in this section because it helps bringing up the concept of data protection and privacy. While data sharing is something beneficial for some applications, it can cause severe damages when phenomena such as the leakage of private data occur. In order to solve this problem, many privacy protection strategies based on different technologies have been proposed in state-of-the-art literature. The aim of these strategies is to secure the privacy of CS tasks and data [17]. For example, in [18], an encryption algorithm is proposed within a distributed structure that allows user to upload their sensory data and receive information without any concern regarding privacy disclo-

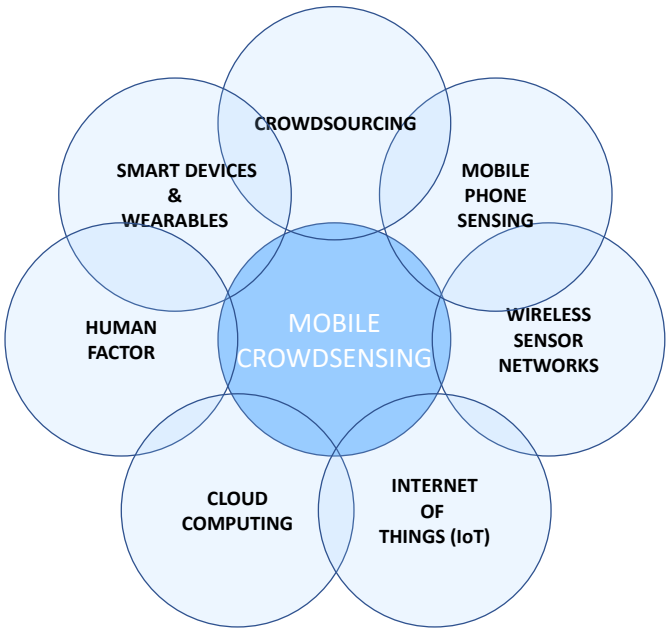

Figure 4: Illustration of the factors contributing to the rise of Mobile Crowdsensing (MCS). MCS assumes that a large group of individuals having devices that are capable of sensing will agree to collectively share their data [14].

sure. The algorithm presented in this publication is very interesting because it combines k-anonymity and blockchain.

1.4.3 GNSS Cooperative Positioning

CP exploits location information from additional measurements between users and thus increases localization accuracy. Several technologies suggested for future 5G networks would benefit from the access to accurate location of Mobile Terminals (MTs). In the light of this, it is conceivable that cooperative localization systems will be implemented in the future 5G networks [19]. Some authors have currently driven the further development of refined localization methods with the use of 5G Massive Multiple-input Multiple-output (MIMO) [4]. These large-scale antenna systems offer advantages not only in communications but also in localization given their high angular resolution.

As it has been previously mentioned, positioning techniques might rely on more than one technology. This also happens with some algorithms of CP that we find in the literature. In numerous publications, it has been proven that the use of GNSS data in CP algorithms is beneficial against GNSS impairments and it generally improves the accuracy [20] [21]. Moreover, the research about theoretical limits of CP is appealing in the GNSS field. For example, in [22], the Cramér-Rao Bound (CRB) for hybrid CP where GNSS information is combined with terrestrial range measurements is assessed. We consider this publication to be relevant because it provides a theoretical tool to evaluate the achievable positioning

accuracy for a given network configuration and can be used as a reference to compare the performance of other practical positioning algorithms.

Most of the CP methods are based on the assumption that the inter-vehicular ranges (IVRs) are available through a radio ranging method. The signal strength (RSS), time of arrival (ToA), or time difference of arrival (TDoA) can be measured for this purpose [23]. It is though important to take into account that the accuracy and feasibility of these methods is limited due to the communication medium constraints and the rapidly changing vehicular environment. Consequently, these are not always a reliable option for safety-critical systems.

GNSS-only cooperative systems are also endorsed in the state-of-the-art literature [24] [25]. GNSS systems are widely deployed for absolute and relative positioning as part of a wide range of ITS applications, given their global coverage, which is of high interest. Nevertheless, these systems have limited availability under some conditions, thus not being able to meet the strict requirements of many safety-critical applications [23]. To address the need of high accuracy and continuity of position information required by such applications, CP and sensor data fusion can help [26].

In [27], IVRs are measured by using raw GNSS observables, which is a similar approach to Differential GNSS (DGNSS), with the difference that both nodes are mobile and there is no presence of a reference or base station. The use of both code and carrier-phase measurements for GNSS-only CP is currently being investigated [28] [29]. Although the carrier-phase measurements are more precise, they are not the first option for CP in current literature. This is due to the fact that with carrier-phase measurements there is the need to resolve the carrier-phase ambiguity, which is computationally expensive.

We would like to focus on the algorithms presented in the publications [24] and [25]. These algorithms suggest that, by using only GPS pseudorange measurements, we can improve the positioning accuracy in cooperative vehicular localization systems, which is critical for cooperative vehicle safety (CVS) applications. The standard Double Differentiation (DD) pseudorange solution is adapted to low-end navigation level GPS receivers for its wide availability in ground vehicles. The Carrier to Noise Ratio (CNR) of raw pseudorange measurements are taken into account for noise mitigation, as they are used to build matrix $\mathbf{W}$ of the proposed Weighted Least Squares (WLS) estimator. The contribution of [25] is mentioned again in Section 2.4 when describing the novelty of the algorithms proposed in this Master Thesis.

1.4.4 Inter-Device Communication Systems

Our intention with this subsection is to emphasize the importance of the communication systems and standards used to exchange raw pseudorange information among neighbouring users in a cooperative scenario. Some studies suggest the use of Dedicated Short Range Communications (DSRC) for this matter [25].

DSRC are one-way or two-way short-range to medium-range wireless communication channels specifically designed for automotive use and a corresponding set of protocols and standards. The IEEE 802.11p is an approved amendment to the IEEE 802.11 standard

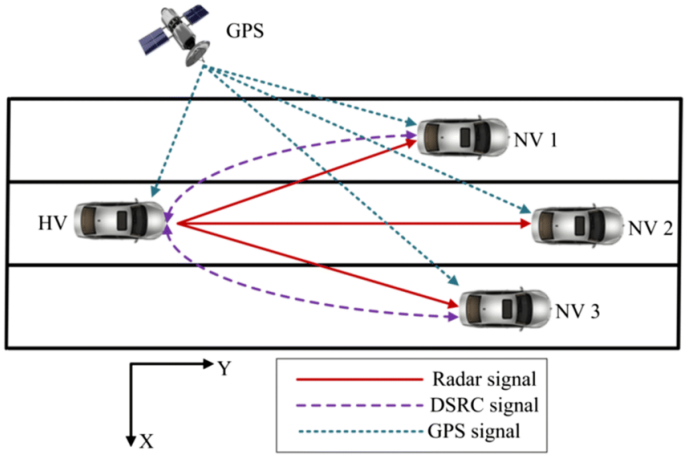

Figure 5: Example scenario of the proposed CP method using GPS in [1]. DSRC signals allow inter-vehicle communication, while GPS and radar signals are used to provide precise location information.

to add wireless access in vehicular environments (WAVE), and it can be used for DSRC communications. According to [1], CP has potential benefits from the emerging vehicular communications based on IEEE 802.11p DSRC, opening new opportunities that support vehicular applications. As GNSS is incorporated in modern vehicles, robust positioning approaches can be developed due to the availability of accurate position information. In Figure 5 we can see an example scenario of the proposed CP method using GPS in [1]. We see how this scenario makes use of DSRC signals that allow inter-vehicle communication. GPS and radar signals are used to provide precise location information to the algorithm proposed in the mentioned publication.

As stated in [30] (Chapter 5.8), for real-time users of DGNSS, a standard format for data communication developed by the Radio Technical Commission for Maritime Services (RTCM) is widely used for communication between the base station and the DGNSS receiver. The standard, named RTCM SC-104, was established in 1994. This is mentioned again in Section 2, as the concept of reference station or base station has not been introduced in the document yet. In [31], they suggest the use of RTCM corrections via DSRC to improve vehicle localization.

2 Background on GNSS Code-Based Positioning

This section focuses on GNSS code-based Positioning. We present how single-frequency code-based positioning is performed with the Standard Positioning Point (SPP) solution, which is widely used in GNSS receivers. The mathematical derivations presented in this section allow the reader to gain in-depth knowledge to move forward into the following sections. An introduction to Single Differentation (SD) techniques is provided, paying special tribute to Differential GNSS (DGNSS).

The Real-Time Kinematic (RTK) method is also mentioned. Nevertheless, RTK also makes use of carrier-phase measurements and this is out of the scope of this research. As it has been previously mentioned in Section 1, the use of carrier-phase measurements involves very high computational costs and consequently cooperative positioning algorithms that only use pseudorange mesurements are lighter. Furthermore, the mathematical derivations that include differentiation of carrier-phase measurements are more complex. Although computationally demanding, a study of GNSS Massive CP using these measurements would be of interest at least from a theoretical point of view. Therefore, this is proposed as a future line of research in Section 6.

Before getting into the mathematical derivations, we would also like to stress the fact that, in DGNSS, SD is performed between receivers and not between satellites. Nevertheless, SD between satellites has been explored in other literature such as in [32], where they try to solve the carrier-phase ambiguity with satellite-satellite SD. In RTK, differentiation is computed between both satellites and receiver (Double Differentiation).

2.1 Introduction to GNSS

In this subsection we are presenting very important concepts related to GNSS. The need of a fourth satellite when estimating the position of a user is justified with the introduction of the pseudorange expression, which includes the clock bias term. Next, a general description of available GNSS frequency bands is provided with the aim of bringing up the advantages provided by multi-frequency GNSS techniques. Finally, the general architecture of a standard GNSS receiver is presented in Figure 8. This is relevant in order to understand that the novel techniques proposed in this research are located at the very end of the receiver chain, i.e., the Navigation Module, providing user position and time as an output.

The concept of clock bias or clock offset and pseudorange must be introduced to the reader. GNSS receivers learn the satellite signal transmit time by inspecting the navigation message. Reception time is also known by the receiver from an internal clock, which has a considerably low precision in comparison to the one of satellite clocks. With the difference between the signal transmit and reception times, an estimated distance between satellite and receiver can be obtained. As GNSS signals are electromagnetic waves propagating at the speed of light, to compute the distance the difference in time is multiplied by the speed of light $c \approx 3e8$ m/s. The estimated distance provides the radius of the spherical surface centered at the satellite and containing the position of the user [33]. Position estimation would be possible with just three satellites in view if there was not a clock offset δt_m^k

between the receiver time (receiver m) and the GNSS time (from satellites $k = 1, \ldots, K$). All range measurements are shifted by an unknown term, which corresponds to this clock offset, and therefore they are referred to as pseudorange measurements or pseudorange observables (see Figure 6). Given the emergence of this unknown, an additional satellite is required to solve the navigation equations. Consequently, a total of four satellites in view is required to solve the position estimation problem.

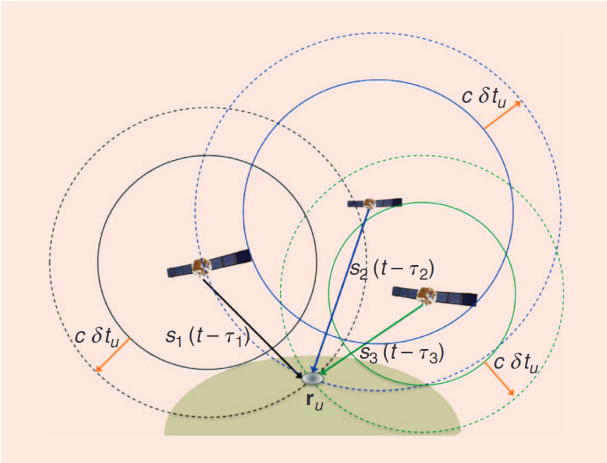

Figure 6: Illustration of the rationale behind GNSS positioning and the effect of unknown receiver clock offset, from [33].

GNSS satellites transmit signals in two or more frequencies that belong to the L band. According to the Institute of Electrical and Electronics Engineers (IEEE), the L band covers the radio spectrum frequencies from 1 to 2 gigahertz (GHz). These are at the top end of the Ultra High Frequency (UHF) band and at the lower end of microwave frequencies. Transmitted signals contain ranging codes and navigation data. The description of the characteristics of GNSS navigation signals is out of the scope of this research. We would like to, however, devote this paragraph to the frequency band allocation in Radio Navigation Satellite Systems (RNSS), which can be found in Figure 7, extracted from [34]. Let us first define GPS, Galileo and GLONASS as the GNSS systems owned by the United States, Europe and Russia, respectively. Two bands are worldwide available in the region allocated to the Aeronautical Radio Navigation Service (ARNS). The first one corresponds to the upper L band and contains GPS L1, Galileo E1 and GLONASS G1. The second one corresponds to the bottom of the lower L band and contains GPS L5 and Galileo E5, with E5a and L5 coexisting in the same frequencies. The remaining GPS L2, GLONASS G2 and Galileo E6 signals were only allocated to RNSS and correspond to intermediate bands, which are more vulnerable to interferences [34].

The previous paragraph is essential to understand the concept of multi-band (also called multi-frequency) GNSS techniques, which allow to simultaneously process signals from different frequency bands. These techniques bring several advantages. For instance, in situations where the system is suffering from interferences such as jamming or spoofing, it might happen that one of the frequency bands used by the receiver is interference-free. Also, the use of different bands allows to cancel ionosphere and troposphere delays.

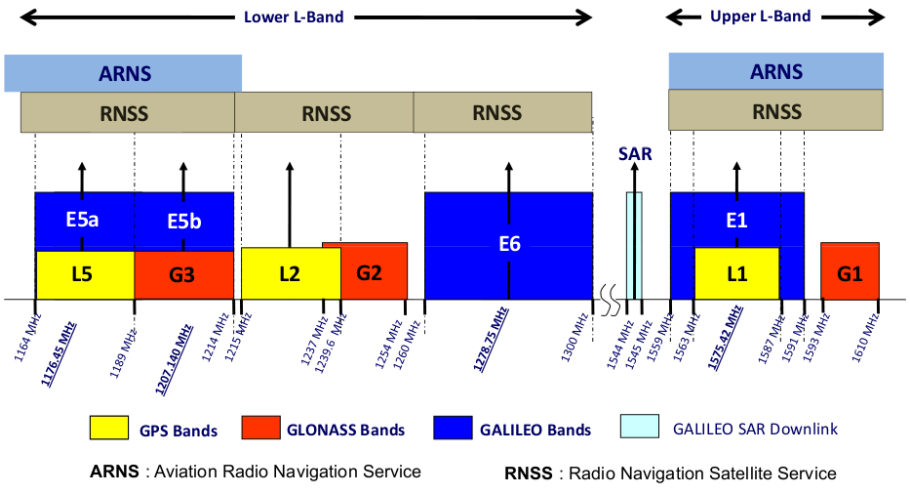

Figure 7: GPS, Galileo and GLONASS navigational frequency bands, from [34].

Furthermore, ambiguity resolution techniques may be improved with the use of more frequencies when combining GNSS measurements (narrow and wide-lane combinations) [35]. All in all, the more measurements are available, the better performance we can achieve. Also, some signals such as L5 have better transmission power, providing better Signal-to-Noise Ratio (SNR) values. In [36], they give insight into the design of GNSS multi-band antennas.

Finally, we would like to present which is the general architecture of a standard GNSS receiver (see Figure 8, extracted from [33]). It is of our interest to provide the reader with this image because we would like to highlight that the stage at which the algorithms proposed in this research are run is at the very end of the chain, at the navigation module. The output of the proposed techniques corresponds to the final chain output, with all the information required by the user: its time and position. A description of the functionalities of each module within the architecture is out of the scope of this Master Thesis.

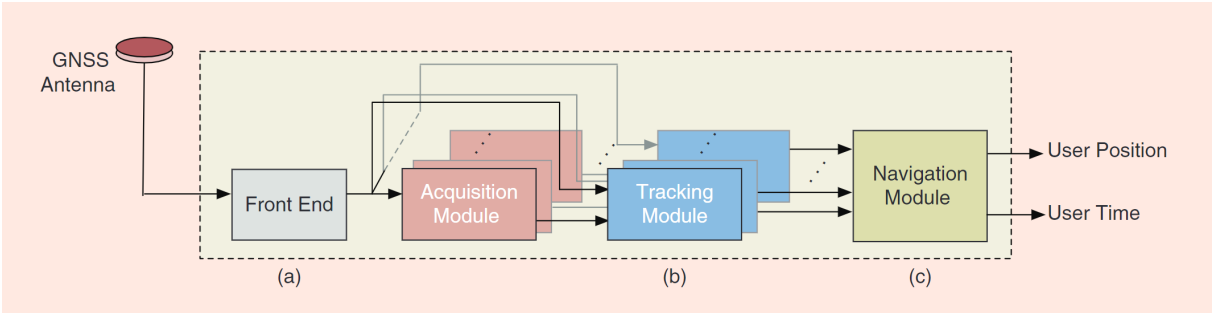

Figure 8: General architecture of a standard GNSS receiver, from [33], including the (a) front end and acquisition module, (b) the tracking module and (c) the navigation module, which outputs the user time and position.

A detailed description of the characteristics (i.e., number of satellites in orbit, coverage, bit rate) of the different GNSS constellations (GPS, Galileo, GLONASS, BeiDou, IRNSS)

is out of the scope of our study. The interested reader may find valuable information regarding this topic in several of the book references provided in this document, such as [3], [30], [37] and [38].

2.2 Single Constellation Standard Positioning Point (SPP) Solution

This subsection aims to describe how single-frequency code-based positioning is performed when computing the Standard Positioning Point (SPP) solution. It is possible to obtain the position coordinates of a receiver $\mathbf{p} = [x,\ y,\ z]^\mathsf{T}$ and its clock bias δt if pseudorange measurements of at least four satellites in view are available.

First of all, a comment regarding the notation used in this section is needed. We introduce two receivers (also called users), which we refer to as the reference user m and the cooperative user n. Expressions such as (1) are particularized for one specific user. For example, P_m^k denotes the pseudorange measurement between user m and satellite k. Although this section could also be understood without particularizing for a specific user, we have found interesting to introduce the notation from Section 4 early in the document.

2.2.1 Pseudorange Measurements Model

P_m^k and P_n^k are the pseudorange code measurements of the reference user m and the cooperative user n with respect to satellite k, being the satellites indexed by $k = 1, \ldots, K$. P_m^k and P_n^k are non-linear functions of the parameter vectors $\boldsymbol{\gamma}_m$ and $\boldsymbol{\gamma}_n$, being $\boldsymbol{\gamma} = \left[\mathbf{p}^\mathsf{T}, c\delta t\right]^\mathsf{T}$. The pseudorange measurements of receiver m with respect to satellite k can be expressed as

$$P_m^k = \rho_m^k + c\left(\delta t_m - \delta t^k\right) + \eta_m^k, \tag{1}$$

being

$$\eta_m^k = c\Delta T_m^k + c\Delta I_m^k + \epsilon_m^k, \tag{2}$$

where ΔT_m^k and ΔI_m^k are the non-dispersive tropospheric delay and the frequency-dependent ionospheric delay terms, respectively. These errors can be estimated and subtracted from the pseudorange measurements in order to mitigate their impact on the error term η_m^k. The term ρ_m^k is defined in (3) and corresponds to the geometric range between receiver m and satellite k.

On the one hand, single-frequency users rely on ionospheric correction models such as Klobuchar in the case of GPS and Nequick in the case of Galileo. On the other hand, multi-frequency users are able to cancel the ionospheric delay with the so-called ionosphere-free combination of GNSS measurements [39]. Single-frequency users do not rely on ionospheric corrections when DGNSS augmentation techniques are used, such as Classical DGNSS or Real Time Kinematic (RTK). With Single Differentiation (SD) and Double Differentiation (DD) techniques in short baseline scenarios the tropospheric and ionospheric delay terms are cancelled and consequently there is no need to estimate them.

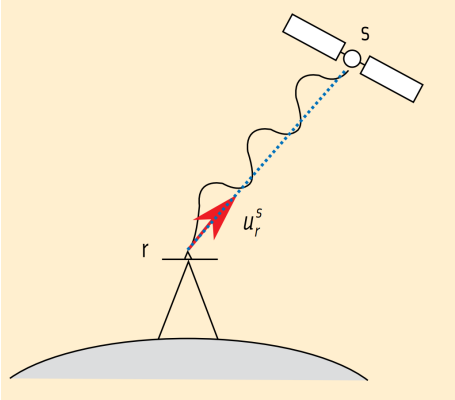

Figure 9: Direction of the Line of Sight (LOS) vector, from [40]. In this figure, u_r^S corresponds to the LOS vector from receiver r to satellite S. In our notation, we denote $\mathbf{LOS}_m^k$ as the LOS vector from receiver m to satellite k.

In (2), the error term $\epsilon_m^k \sim \mathcal{N}(0, \sigma_m^2)$ takes into account errors from various sources such as multipath, ephemeris and relativistic effects. It is considered that fore a same receiver the realizations of this error are independent and identically distributed ($i.i.d.$) for different satellites.

2.2.2 Geometric Range Linearization

The non-linearity of the pseudorange expression in (1) comes from the geometric range ρ_m^k, which does not include the clock biases and can be defined as

$$\rho_m^k = \left\|\mathbf{p}_m - \mathbf{p}^k\right\| = \sqrt{(x_m - x^k)^2 + (y_m - y^k)^2 + (z_m - z^k)^2}, \tag{3}$$

where $\mathbf{p}_m$ and $\mathbf{p}^k$ are the receiver and satellite position Earth-Centered, Earth-Fixed (ECEF) coordinates, respectively. Coordinate systems used in GNSS are later defined in Section 4. The satellite coordinates are sent by the satellite to the receiver in the navigation message. The geometric range measurements and clock biases can be put together in the function $\mathbf{h}(\boldsymbol{\gamma}_m)$, which already contains the contribution of the K satellites, as we have one available pseudorange measurement per satellite in view. The linearization of this function with respect to the vector $\boldsymbol{\gamma}_m$ can be obtained by applying the first Taylor expansion as

$$\mathbf{h}(\boldsymbol{\gamma}_m) = \begin{bmatrix} \rho_m^1 + c(\delta t_m - \delta t^1) \\ \vdots \\ \rho_m^K + c(\delta t_m - \delta t^K) \end{bmatrix} \approx \mathbf{h}(\boldsymbol{\gamma}_{0,m}) + \mathbf{H}(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_{0,m}), \tag{4}$$

where $\boldsymbol{\gamma}_{0,m}$ is the value of $\boldsymbol{\gamma}_m$ at which the function $\mathbf{h}(\boldsymbol{\gamma}_m)$ is linearized. The calculation of the $\mathbf{H}$ matrix can be performed as

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}\right|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}_{0,m}} = \left.\begin{bmatrix} \frac{\partial h^1(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} \\ \vdots \\ \frac{\partial h^K(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} \end{bmatrix}\right|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}_{0,m}} \left.\begin{bmatrix} \frac{\partial (\rho_m^1 + c(\delta t_m - \delta t^1))}{\partial \boldsymbol{\gamma}} \\ \vdots \\ \frac{\partial (\rho_m^K + c(\delta t_m - \delta t^K))}{\partial \boldsymbol{\gamma}} \end{bmatrix}\right|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}_{0,m}}, \tag{5}$$

where the subindex m in $\boldsymbol{\gamma}_m$ is avoided for simplicity. The gradient of $\mathbf{h}(\boldsymbol{\gamma})$ with respect to the vector $\boldsymbol{\gamma}$ can be calculated as

$$\mathbf{H} = \left.\begin{bmatrix} \frac{\partial h^1(\boldsymbol{\gamma})}{\partial x_m} & \frac{\partial h^1(\boldsymbol{\gamma})}{\partial y_m} & \frac{\partial h^1(\boldsymbol{\gamma})}{\partial z_m} & \frac{\partial h^1(\boldsymbol{\gamma})}{\partial c\delta t_m} \\ \vdots & & & \\ \frac{\partial h^K(\boldsymbol{\gamma})}{\partial x_m} & \frac{\partial h^K(\boldsymbol{\gamma})}{\partial y_m} & \frac{\partial h^K(\boldsymbol{\gamma})}{\partial z_m} & \frac{\partial h^K(\boldsymbol{\gamma})}{\partial c\delta t_m} \end{bmatrix}\right|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}_{0,m}}, \tag{6}$$

where the subindex m in $\boldsymbol{\gamma}_m$ is again avoided for simplicity. To obtain the values of the gradient in (6), we calculate the partial derivatives of $\mathbf{h}^k(\boldsymbol{\gamma})$ with respect to the position-related variables as

$$\begin{aligned} \frac{\partial h^k(\boldsymbol{\gamma})}{\partial x_m} &= \frac{\partial \left(\sqrt{(x_m - x^k)^2 + (y_m - y^k)^2 + (z_m - z^k)^2}\right)}{\partial x_m} \\ &= \frac{2(x_m - x^k)}{2\sqrt{(x_m - x^k)^2 + (y_m - y^k)^2 + (z_m - z^k)^2}} = \frac{-(x^k - x_m)}{\rho_m^k}. \end{aligned} \tag{7}$$

This result applies for the three coordinates x_m, y_m and z_m. It can be expressed as a function of the $\mathbf{LOS}_m^k$ vector, which if defined from the receiver m to the satellite k can be expressed as [41]

$$\mathbf{LOS}_m^k = \frac{[x^k - x_m,\, y^k - y_m,\, z^k - z_m]}{\rho_m^k}. \tag{8}$$

The definition of the LOS vector direction is important because it affects the order of the terms in the subtraction. In Figure 9, from [40], u_r^S corresponds to the LOS vector from receiver r to satellite S.

The partial derivative of $h^k(\boldsymbol{\gamma})$ with respect to the code bias is computed as

$$\frac{\partial h^k(\boldsymbol{\gamma})}{\partial c\delta t_m} = \frac{\partial \left(c(\delta t_m - \delta t^k)\right)}{\partial c\delta t_m} = 1, \tag{9}$$

which reminds us that actually the clock bias term is linear in the pseudorange expression. Taking all the aforementioned results into account, the $\mathbf{H}$ matrix can be expressed as

$$\mathbf{H} = \begin{bmatrix} -\mathbf{LOS}_m^1 & 1 \\ \vdots & \vdots \\ -\mathbf{LOS}_m^K & 1 \end{bmatrix}. \tag{10}$$

Finally, we can express the pseudorange code measurements of receiver m as

$$\mathbf{P}_m \approx \mathbf{h}(\boldsymbol{\gamma}_{0,m}) + \mathbf{H}(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_{0,m}) + \boldsymbol{\eta}_m = \begin{bmatrix} \rho_{m,0}^1 \\ \vdots \\ \rho_{m,0}^K \end{bmatrix} + \begin{bmatrix} -\mathbf{LOS}_m^1 & 1 \\ \vdots & \vdots \\ -\mathbf{LOS}_m^K & 1 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ c\delta t_m \end{bmatrix} + \begin{bmatrix} \eta_m^1 \\ \vdots \\ \eta_m^K \end{bmatrix}, \tag{11}$$

where $\delta x = x_m - x_0$, $\delta y = y_m - y_0$ and $\delta z = z_r - z_0$. There is no differential of δt because this term was already linear. Another way of justifying this is that the contribution of δt in $\mathbf{h}(\boldsymbol{\gamma}_0)$, δt_0, is cancelled with the term δt_0 found in $\delta\left(\delta t\right) = \delta t - \delta t_0$.

2.2.3 Observation Model

The expression in (11) can be rearranged in order to obtain a linear model which allows to estimate the position solution from the following linear equation

$$\mathbf{P}_m - \mathbf{h}(\boldsymbol{\gamma}_{0,m}) = \mathbf{H}(\boldsymbol{\gamma} - \boldsymbol{\gamma}_{0,m}) + \boldsymbol{\eta}_m. \tag{12}$$

This observation model resembles the ones presented in (25) and (30) for the two proposed algorithms. It is necessary to state in a very clear manner the expression of the observation model, and also the probability distribution of the observations, so that the LSE can be derived in a straightforward manner.

2.2.4 Iterative LSE

The presented linear model can be solved by a standard Iterative Least Squares Estimator (LSE) as

$$\hat{\boldsymbol{\gamma}}_m^{j+1} = \hat{\boldsymbol{\gamma}}_m^j + \left(\mathbf{H}^\mathsf{T}\mathbf{H}\right)^{-1}\mathbf{H}^\mathsf{T}\left(\mathbf{P}_m - \mathbf{h}(\hat{\boldsymbol{\gamma}}_m^j)\right). \tag{13}$$

The iterative solution can be initially approximated as $\hat{\boldsymbol{\gamma}}_{0,m} = 0$, meaning that the initial guess of the estimator is located at the center of the Earth. After convergence, this yields to the estimator $\hat{\boldsymbol{\gamma}}_m$. The matrix $\left(\mathbf{H}^\mathsf{T}\mathbf{H}\right)^{-1}\mathbf{H}^\mathsf{T}$ is known as the pseudo-inverse of $\mathbf{H}$ or LSE solution matrix and has a dimension of $4 \times K$, being K the number of satellites in view, for the single-constellation case described in this thesis.

In Section 3.3.4, some thinking regarding the LSE as part of the navigation solution is provided.

2.2.5 Dilution of Precision

The LSE solution matrix $\mathbf{H}$ depends on the relative geometry of the user and satellite positions that are involved in the computation of the estimator solution. The Dilution of precision (DOP) is a term used in satellite navigation to specify the error propagation as a mathematical effect of this relative geometry [30]. Although this metric is not further used to evaluate the proposed algorithm, we consider relevant to introduce it in the background section because it can be inferred from the LSE solution matrix, which is a very important matrix in most of the thesis mathematical derivations.

If assuming that the pseudorange errors are *i.i.d.*, jointly Gaussian and with zero mean, their covariance can be modeled, as a function of $\mathbf{H}$, as

$$cov(\delta\hat{\mathbf{x}}) = (\mathbf{H}^T\mathbf{H})^{-1}\sigma^2_{UERE} = \begin{pmatrix} \sigma^2_x & \sigma^2_{xy} & \sigma^2_{xz} & \sigma^2_{x\hat{t}} \\ \sigma^2_{xy} & \sigma^2_y & \sigma^2_{yz} & \sigma^2_{y\hat{t}} \\ \sigma^2_{xz} & \sigma^2_{yz} & \sigma^2_z & \sigma^2_{z\hat{t}} \\ \sigma^2_{x\hat{t}} & \sigma^2_{y\hat{t}} & \sigma^2_{z\hat{t}} & \sigma^2_{\hat{t}} \end{pmatrix}, \tag{14}$$

where $\delta\hat{\mathbf{x}}$ corresponds to the error of the solution parameters $\hat{\mathbf{x}} = (x, y, z, c \cdot \delta t)^T$, with the receiver position expressed in local East, North, Up (ENU) coordinates.

The term σ^2_{UERE} corresponds to the User Equivalent Range Error (UERE) variance, which is the umbrella term for error sources such as the satellite clock, upper atmosphere (ionosphere), receiver clock, satellite orbit, lower atmosphere (troposphere) and multipath. The DOP is a parameter without units used to describe the satellite geometry and UERE is the total sum of errors involved in satellite communication expressed as the unit of distance. These two metrics can be used to evaluate the performance of a positioning algorithm such as in [42].

The DOP gives a measure of the global quality of the obtained solution as the sum of the combinations of variances per component [38]. Some of the combinations that are commonly used to describe the expected precision per component are the Geometric DOP (GDOP), the Position DOP (PDOP), the Horizontal DOP (HDOP), the Vertical DOP (VDOP) and the Time DOP (TDOP). In cases where the position cannot be determined due to geometric singularities, the DOP values are infinitely large. For example, the GDOP can be computed as

$$\text{GDOP} = \frac{\sqrt{\sigma^2_x + \sigma^2_y + \sigma^2_z + \sigma^2_{\hat{t}}}}{\sigma_{\text{UERE}}}. \tag{15}$$

Moreover, there has been studies to investigate the use of the DOP metric to perform selection of satellites in GNSS positioning algorithms [43]. In general, increasing the number of satellites considered in the algorithm improves the positioning accuracy and increases availability. However, it reduces the positioning accuracy improvement rate and increases calculation loads. An appropriate satellite selection method is required and some literature proposes the use of HDOP and VDOP measurements with this purpose [44]. Other

metrics can be taken into account to perform satellite selection, such as the signal level [45].

2.2.6 Weighted Least Squares Estimator

If assuming that the UEREs of all satellites that contribute to the position solution are *i.i.d.*, the model could lead to a sub-optimal solution in the position estimation. In order to avoid this, the pseudoange errors are modeled as Gaussian with a covariance matrix $\mathbf{V}$, which can be described by different models based on different aspects such as the noise level and elevation of the satellites. From this matrix, a weighting matrix can be calculated as $\mathbf{W} = \mathbf{V}^{-1}$. If the weighting matrix is introduced to the LSE solution in (13), the new iterative solution is obtained from a Weighted Least Squares (WLS) estimator as

$$\hat{\boldsymbol{\gamma}}_m^{j+1} = \hat{\boldsymbol{\gamma}}_m^{j} + \left(\mathbf{H}^{\mathsf{T}}\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{W}\left(\mathbf{P}_m - \mathbf{h}(\hat{\boldsymbol{\gamma}}_m^{j})\right). \tag{16}$$

In the case of $\mathbf{V} = \sigma^2_{UERE}\mathbf{I}_K$, the presented solution is again the one found in (13), without the need of a weighting matrix $\mathbf{W}$. For single-frequency positioning algorithms, the pseudo-range errors are dominated by the residual ionospheric delays. When the ionospheric errors are estimated using methods as Klobuchar or NeQuick, residual errors are highly correlated for all satellites involved in the solution. This is the reason why previous literature has emphasized on the modeling of matrix $\mathbf{V}$, which is a relevant object of study [46].

In most cases, the strength of each satellite signal is monitored in order to model $\mathbf{W}$ [33]. The interested reader may find a valuable summary on signal strength monitoring techniques in [47]. In our implementation, we are assuming that we have complete knowledge of the weighting matrix in the LSE estimator.

2.3 Classic Single Differentiation of Pseudorange Measurements

We use the term DGNSS to refer to the classical DGNSS algorithm, which is based on Single Differenciation (SD) of pseudorange measurements. In other literature, this term sometimes refers to other GNSS Augmentation systems such as RTK or Wide Area RTK (WARTK).

2.3.1 Differential GNSS (DGNSS)

In DGNSS applications, the position and clock bias of the reference receiver are unknown. However, it is assumed that they are known with high precision for another receiver which we refer to as reference station or base station. Ideally, the reference station is located near the receiver whose position wants to be estimated. This way, the ionospheric and tropospheric errors are similar between the two receivers and their contribution can be cancelled. The distance between the reference receiver and the reference station is known as baseline. When the baseline is under 10 km it is considered that the setup is under short baseline conditions. As the position and clock bias of the reference station are known,

the unknown parameter vector associated to the reference receiver can be successfully estimated when the pseudoranges of both users are combined.

The main objective of techniques using SD of pseudorange measurements such as DGNSS is to cancel, or at least mitigate, the sources of error listed in Table 1 (from [30], Chapter 5.8). The potential error size from these sources, as well as the resulting error after mitigating these sources with DGNSS are also provided in this table. We can see how the use of DGNSS is successful in the task of improving the accuracy of GNSS positioning algorithms.

Nevertheless, it is relevant to highlight that for a DGNSS implementation, the communication with a reference station that knows its coordinates with very high precision is needed. This is the weakness of DGNSS that we are trying to improve with the techniques proposed in this paper.

Table 1: Error sources that techniques using Single Differentiation (SD) of pseudorange measurements aim to mitigate (from [30], Chapter 5.8). An orientative value of the potential error size and mitigation with DGNSS is provided. Single-frequency receiver under short baseline scenario is assumed.

Error Source	Potential Error (in meters)	Error after SD (in meters)
Satellite Clock	2	0
Satellite Ephemeris Prediction	2	0.1
Ionospheric Delay	2-10	0.2
Tropospheric Delay	2.5	0.2

There are some sources of error that DGNSS techniques cannot mitigate because they are uncorrelated between receivers, or even between antennas of a same receiver. The latter is the case of errors introduced by multipath propagation. These errors are out of the scope of this research.

Lastly, and as previously mentioned in Section 1, real-time DGNSS users usually use an RTCM standard format for data communication with the reference station [30] (Chapter 5.8). This standard corresponds to the RTCM SC-104, which was established in 1994. RTCM defined data messages and also an interface between the DGNSS receiver and the base station.

2.3.2 Mathematical Derivation

In this section, the reference receiver is referred to as receiver m, while the reference or base station is referred to as receiver n. It is essential to not be mistaken by the nomenclature, as the reference receiver is not a reference station. The reference receiver coordinates are not known and our objective is to estimate them, while the coordinates of the reference or base station are known in a precise manner.

According to (11), and as it has been previously explained, the pseudorange code mea-

surements of the reference user m can be expressed as

$$P_m^k = \rho_m^k + c\left(\delta t_m - \delta t^k\right) + \eta_m^k, \tag{17}$$

being

$$\eta_m^k = c\Delta T_m^k + c\Delta I_m^k + \epsilon_m^k. \tag{18}$$

In the same manner, the pseudorange code measurements of the cooperative user n can be expressed as

$$P_n^k = \rho_n^k + c\left(\delta t_n - \delta t^k\right) + \eta_n^k. \tag{19}$$

In order to perform SD of the pseudorange code measurements of users m and n, we subtract these expressions as

$$\Delta P = P_m^k - P_n^k = r^k + c\left(\delta t_m - \delta t_n\right) + \eta_m^k - \eta_n^k, \tag{20}$$

where $r^k = \rho_m^k - \rho_n^k$. We can see that the satellite clock bias term is cancelled due to differentiation, as well as the ionospheric and tropospheric delay terms $c\Delta T_m^k + c\Delta I_m^k$. The cancellation of atmospheric delay terms happens in short baseline scenarios, meaning that the reference user and the base station are at most 10 km far from each other. With two receivers separated by 25 km, the differential ionospheric delay is not cancelled. We can assume in these cases the delay to be typically at 10-20 cm level. This difference can increase up to 1 m with 100 km of baseline [30] (Chapter 5.8). DGNSS accuracy is in the order of 1 meter (1 sigma) for users in the range of few tens of kilometers from the reference station.

2.4 Contribution and Hypothesis

As it has been stated in this section, DGNSS uses Single Differentiation (SD) of pseudorange code measurements to enhance GNSS by correcting error sources listed in Table 1. As seen in this table, the error mitigation provided by DGNSS is very powerful because it reduces high errors to up to very few centimeters. This is the reason why access to DGNSS correction information makes differential GPS and GNSS receivers much more accurate than other receivers. A simple graphic proposal for a DGNSS scenario can be found in Figure 10 (left subfigure).

Having understood how beneficial DGNSS is for GNSS receivers, it is time to introduce the contribution that we aim to provide with this research. Our objective with this thesis is to derive and implement a method that provides a performance enhancement comparable to the one provided by DGNSS without the need of DGNSS corrections, or equivalently, without the need of precise knowledge about the position of a reference station.

The hypothesis posed in this research states that it is possible to increase GNSS accuracy with corrections provided by N cooperative users that have partial knowledge of their positions. We will assume that we are provided with noisy observations of their position and clock bias vectors $\boldsymbol{\gamma}_n$.

2.4.1 Novelty

We consider that the algorithms proposed in this research are novel. There is a large amount of literature on the use of GNSS pseudorange measurements for cooperative positioning in scenarios with N users. However, most of these publications do not explore the use of differentiation (single or double) of measurements between the N users or, if they do, the position estimation algorithms are more complex and require more steps, such as the one proposed in [25].

In [25], they propose a CP algorithm that uses differentiation of pseudorange measurements between N users. However, their approach is different to ours, as it requires the implementation of two algorithms. With the first algorithm, they introduce a ranging technique based on Double Differentiation (DD) of pseudorange measurements, which is used to detect inter-user distances. The second algorithm relies on the inter-user distances estimated with the first algorithm and also on inaccurate pseudorange measurements. A constrained nonlinear optimization problem is formulated by maximizing the PDF of the GPS fix error constrained to a road space constraint.

The inputs of the technique proposed in [25] match the ones of the Massive User-Centric Single Differentiation (MUCSD) algorithm that is proposed for this Master Thesis. These inputs are, first, noisy GPS fix values (which we refer to as noisy position and clock bias observations) and, second, inaccurate GNSS pseudorange measurements. The technique proposed in [25] is more complex because it needs the implementation of two separate algorithms and also uses DD instead of SD, which has a higher computational cost. Furthermore, the position estimation solution that we are proposing is closer to the navigation filter solution in standard GNSS receivers, with a Weighted Least Squares (WLS) estimator.

The first objective of this Master Thesis is to find whether it is possible to achieve DGNSS performance by increasing the value of N. The second objective is to, once DGNSS performance is achieved, try to find how the noise in cooperative user positions affects the performance of our technique. With this purpose, it is required to perform several experiments.

3 Proposed GNSS Cooperative Positioning Algorithm

The contribution of this research is hereafter developed. In this section, the proposed algorithms are mathematically derived.

3.1 Motivation of the UCSD and MUCSD algorithms

We first propose the User-Centric Single Differentiation (UCSD) (see Figure 10, center) algorithm, which differs from Differential GNSS (DGNSS) (see Figure 10, left) because the coordinates of the additional user, or station, are known with an error instead of with very high precision. With UCSD and DGNSS we achieve the cancellation of the tropospheric and ionospheric delays in short baseline scenarios. However, with UCSD the performance is worse due to the coordinates of the additional user being noisy, which increases the estimation error.

To solve this, the Massive User-Centric Single Differentiation (MUCSD) algorithm is proposed (see Figure 10, right), where N cooperative users are introduced to the scenario. The diversity in users increases available information and thus the algorithm performance. With MUCSD, it is shown that it is possible to obtain accurate positioning results if the positioning error of the cooperative users is not too high.

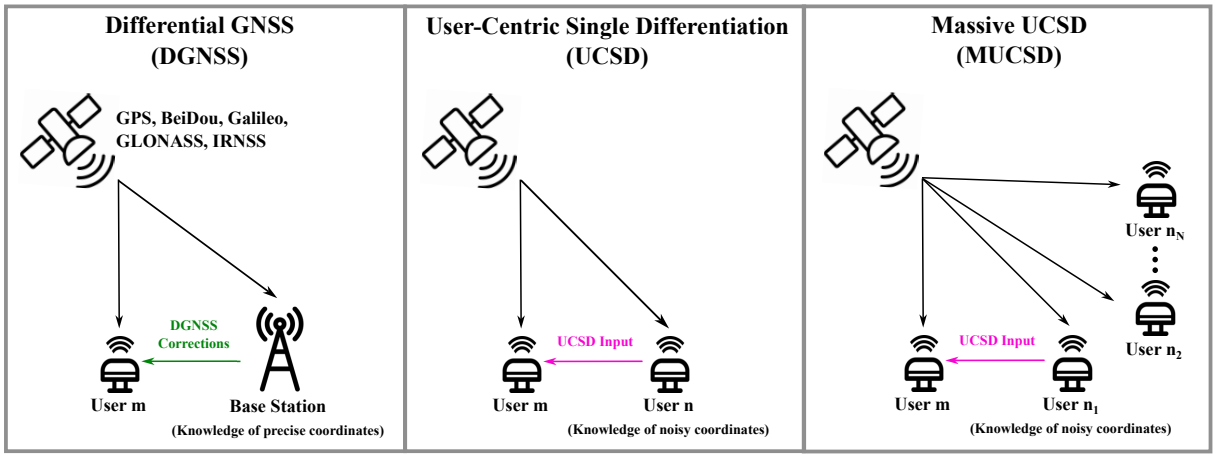

Figure 10: Simple description of (from left to right) a standard DGNSS scenario and the scenarios proposed for a system using User-Centric Single Differentiation (UCSD) and Massive UCSD (MUCSD), respectively.

With the UCSD algorithm, we propose to perform Single Differentiation (SD) of pseudorange measurements between a reference user m and an additional user n. The aim of the algorithm is to estimate the position and clock bias of user m, $\boldsymbol{\gamma}_m = [\mathbf{p}_m^\mathsf{T}, c\delta t_m]^\mathsf{T}$. The proposed technique differs from DGNSS because with UCSD we do not have a reference or base station, whose position and clock bias are known with high precision. Instead, noisy observations are available as $\bar{\boldsymbol{\gamma}}_n = [\bar{\mathbf{p}}_n^\mathsf{T}, c\bar{\delta t}_n]^\mathsf{T}$, where the bar operator denotes that the values do not correspond to the actual position and clock bias of user n. Taking this into account, we can state that the UCSD is tackling a more challenging problem than DGNSS because with UCSD we are missing the high precision provided by a base or

reference station. As the observations are noisy and not precise, the station, which would correspond to user n, is now referred to as a cooperative user. In Figure 10, we can see a very simple description of the differences between a DGNSS system scenario and the setup for the proposed UCSD.

It is relevant to highlight that with the UCSD and MUCSD algorithms the objective is to estimate the parameter vector $\boldsymbol{\gamma}_m$. It is not of our interest to estimate the position and clock bias of the N cooperative users in this section. This is the reason why we refer to the proposed technique as Massive User-Centric Single Differentiation.

An Iterative Least Squares Estimator (LSE) is proposed for each algorithm, together with their respective variance lower bounds, which have been calculated with the Cramér–Rao bound (CRB). In Table 2, the two estimators are listed together with a quick link to the most relevant equations of their mathematical derivation.

Table 2: Estimators proposed with the UCSD and MUCSD algorithms, which correspond to the main contribution of this research. A link to the observation model, error covariance matrix, Iterative LSE and variance lower bound (CRB) of the estimators is provided in this table.

Notation	Observation Model	Error Cov	Iterative LSE	CRB
$\hat{\boldsymbol{\gamma}}_m^{\text{UCSD}}$	Eq. (25)	Eq. (26)	Eq. (28)	Eq. (27)
$\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}$	Eq. (30)	Eq. (35)	Eq. (45)	Eq. (41)

Table 3: Description of the experiments conducted to test the performance of the UCSD and MUCSD algorithms.

ID	Tested Algorithm	Experiment-dependent parameter values
A	UCSD	Table 7
B	MUCSD	Tables 8 and 10
C	MUCSD	Table 11

3.2 User-Centric Single Differentiation (UCSD)

In this subsection, the model for UCSD, with 1 cooperative receiver, is mathematically derived. The UCSD algorithm estimator is denoted as $\hat{\boldsymbol{\gamma}}_m^{\text{UCSD}}$ and it is defined in Section 3.2.4. The lower bound on the variance of the proposed estimator can be found in Section 3.2.3.

3.2.1 Observation Model

The observation model for the SD between the reference receiver and a cooperative receiver that provides noisy observations of its state vector is formulated in this subsection. According to (11), the pseudorange code measurements of the reference user m can be expressed as

$$\mathbf{P}_m = \mathbf{h}(\boldsymbol{\gamma}_m) + \boldsymbol{\eta}_m \approx \mathbf{h}(\boldsymbol{\gamma}_{0,m}) + \mathbf{H}(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_{0,m}) + \boldsymbol{\eta}_m. \tag{21}$$

where $\boldsymbol{\gamma}_m = [\mathbf{p}_m^{\mathsf{T}}, c\delta t_m]^{\mathsf{T}}$ contains the unknown parameters to be estimated and $\boldsymbol{\gamma}_{0,m}$ corresponds to the point at which the measurements $\mathbf{P}_m$ are linearized when applying the first-order Taylor expansion. The error terms $\eta_m^k = c\Delta T_m^k + c\Delta I_m^k + \epsilon_m^k$ and $\epsilon_m^k \sim \mathcal{N}(0,\, \sigma_m^2)$ have been previously defined in Section 2.

As it has been previously stated, it is assumed that the position and clock bias of the cooperative user n are known with an error. This error can be modeled as $\boldsymbol{\eta}_{\boldsymbol{\gamma}_n} \sim \mathcal{N}(0,\, \sigma_{\boldsymbol{\gamma}_n}^2\, \mathbf{I}_4)$, with dimension 4×1. The noisy observations of user n can be expressed as $\bar{\boldsymbol{\gamma}}_n = [\bar{\mathbf{p}}_n^{\mathsf{T}}, c\delta\bar{t}_n]^{\mathsf{T}} = \boldsymbol{\gamma}_n + \boldsymbol{\eta}_{\boldsymbol{\gamma}_n} \sim \mathcal{N}(\boldsymbol{\gamma}_n,\, \sigma_{\boldsymbol{\gamma}_n}^2\, \mathbf{I}_4)$, where $\boldsymbol{\gamma}_n$ is the true value of the parameter vector. The pseudorange code measurements of the cooperative user n can therefore be approximated by the first-order Taylor expansion as

$$\mathbf{P}_n = \mathbf{h}(\boldsymbol{\gamma}_n) + \boldsymbol{\eta}_n \approx \mathbf{h}(\bar{\boldsymbol{\gamma}}_n) + \mathbf{H}(\boldsymbol{\gamma}_n - \bar{\boldsymbol{\gamma}}_n) + \boldsymbol{\eta}_n = \mathbf{h}(\bar{\boldsymbol{\gamma}}_n) + \mathbf{H}\boldsymbol{\eta}_{\boldsymbol{\gamma}_n} + \boldsymbol{\eta}_n, \tag{22}$$

where $\mathbf{P}_n$ is linearized with respect to the point $\bar{\boldsymbol{\gamma}}_n$ so that the error term $\boldsymbol{\eta}_{\boldsymbol{\gamma}_n}$ shows in the equation and its contribution to the covariance matrix can be calculated in a straightforward manner.

For simplicity, it can be assumed that the LOS vectors are approximately the same in the case of the two receivers due to the geometry of the problem. Therefore, it is considered that the $\mathbf{H}$ matrices in (21) and (22) are approximately the same. Considering this approximation and combining these two equations, the SD of code measurements between receivers m and n can be expressed as

$$\Delta\boldsymbol{\rho}_{m,n} = \mathbf{P}_m - \mathbf{P}_n \approx \mathbf{h}(\boldsymbol{\gamma}_{0,m}) + \mathbf{H}(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_{0,m}) - \mathbf{h}(\bar{\boldsymbol{\gamma}}_n) + \boldsymbol{\epsilon}_{m,n}, \tag{23}$$

where the tropospheric and ionospheric delays have been cancelled in the error term $\boldsymbol{\epsilon}_{m,n} = \boldsymbol{\epsilon}_m - \boldsymbol{\epsilon}_n - \boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n}$, being $\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n} = \mathbf{H}\boldsymbol{\eta}_{\boldsymbol{\gamma}_n} \sim \mathcal{N}(0,\, \sigma_{\boldsymbol{\gamma}_n}^2 \mathbf{H}\mathbf{H}^{\mathsf{T}})$. The cancellation occurs due to the differentiation between two users in a short baseline scenario (distances under 10 km).

The components of vector $\boldsymbol{\epsilon}_{m,n}$ are

$$\boldsymbol{\epsilon}_{m,n} = \begin{bmatrix} \epsilon_m^1 \\ \vdots \\ \epsilon_m^K \end{bmatrix} - \begin{bmatrix} \epsilon_n^1 \\ \vdots \\ \epsilon_n^K \end{bmatrix} - \begin{bmatrix} -\mathbf{LOS}_m^1 & 1 \\ \vdots & \vdots \\ -\mathbf{LOS}_m^K & 1 \end{bmatrix} \begin{bmatrix} \eta_{x_n} \\ \eta_{y_n} \\ \eta_{z_n} \\ \eta_{c\delta t_n} \end{bmatrix} = \begin{bmatrix} \epsilon_m^1 \\ \vdots \\ \epsilon_m^K \end{bmatrix} - \begin{bmatrix} \epsilon_n^1 \\ \vdots \\ \epsilon_n^K \end{bmatrix} - \begin{bmatrix} \epsilon_{\boldsymbol{\gamma}_n}^1 \\ \vdots \\ \epsilon_{\boldsymbol{\gamma}_n}^K \end{bmatrix}, \tag{24}$$

where $\epsilon_m^k \sim \mathcal{N}(0,\, \sigma_m^2)$ and $\epsilon_n^k \sim \mathcal{N}(0,\, \sigma_n^2)$.

A linear observation model can be built from (23) as follows

$$\mathbf{y}_{m,n} = \Delta\boldsymbol{\rho}_{m,n} - \mathbf{h}(\boldsymbol{\gamma}_{0,m}) + \mathbf{h}(\bar{\boldsymbol{\gamma}}_n) = \mathbf{H}(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_{m,0}) + \boldsymbol{\epsilon}_{m,n}. \tag{25}$$

These observations can be modeled as $\mathbf{y}_{m,n} \sim \mathcal{N}(\boldsymbol{\mu}_{m,n}, \boldsymbol{\Sigma}_{m,n})$, where $\boldsymbol{\mu}_{m,n} = \mathbf{H}\boldsymbol{\gamma}_m$ if the term $\mathbf{H}\boldsymbol{\gamma}_{m,0}$ is moved to the left side of (25). Also, $\boldsymbol{\Sigma}_{m,n} = \mathbb{E}\left[\boldsymbol{\epsilon}_{m,n}\boldsymbol{\epsilon}_{m,n}^{\mathsf{T}}\right]$, which corresponds to the error covariance matrix.

3.2.2 Error Covariance Matrix

When computing the SD of measurements between two users, the variance of the resulting code error term $\boldsymbol{\epsilon}_m - \boldsymbol{\epsilon}_n$ is the sum of each code error variance. It is considered that $\epsilon_m^k \sim \mathcal{N}(0, \sigma_m^2)$, $\epsilon_n^k \sim \mathcal{N}(0, \sigma_n^2)$ and $\sigma_m = \sigma_n$. Also, it is assumed that the error terms are *i.i.d.* between receivers and also between satellites of a same receiver. Therefore, the error covariance matrix can be expressed as

$$\boldsymbol{\Sigma}_{m,n} = \boldsymbol{\Sigma}_m - \boldsymbol{\Sigma}_n - \boldsymbol{\Sigma}_{\gamma_n} = 2\sigma^2\mathbf{I}_K + \sigma_{\gamma_n}^2\mathbf{H}\mathbf{H}^{\mathsf{T}}. \tag{26}$$

Heatmap figures created from $\boldsymbol{\Sigma}_{m,n}$ for different values of σ_{γ_n} and $\sigma = 1$ meter are shown in Figure 19.

3.2.3 CRB

The CRB can be used to find the lower bound on the variance of the unbiased estimator of each element in $\boldsymbol{\gamma}_m$. The estimator of the i-th element is denoted as $\left[\hat{\boldsymbol{\gamma}}_m^{\text{UCSD}}\right]_i$, and its CRB corresponds to the $[i,i]$ element of the FIM inverse as

$$\text{var}\left(\left[\hat{\boldsymbol{\gamma}}_m^{\text{UCSD}}\right]_i\right) \geq \text{CRB}\left(\left[\hat{\boldsymbol{\gamma}}_m^{\text{UCSD}}\right]_i\right) = \left[\left(\mathcal{I}^{\text{UCSD}}(\boldsymbol{\gamma}_m)\right)^{-1}\right]_{ii} = \left[(\mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}_{m,n}^{-1}\mathbf{H})^{-1}\right]_{ii}. \tag{27}$$

The mathematical derivation that leads to the expression in (27) can be inferred from the one provided in Section 3.3.3 for the MUCSD algorithm. The definition of the Fisher Information Matrix (FIM) is also provided later. In this case, the FIM is denoted as $\mathcal{I}^{\text{UCSD}}$ because it is calculated with a likelihood function that describes the observation model in (25), which only contains the contribution of one user pair. Therefore, it differs from the FIM denoted as $\mathcal{I}^{\text{MUCSD}}$ from (40).

3.2.4 Iterative LSE

As explained in Section 3.3.4, due to the non-linearity in the pseudorange measurement expression, it is necessary to solve the estimator iteratively by linearizing the function at some initial guess $\boldsymbol{\gamma}_{0,m}$. A new estimate can be computed as indicated in (28), where $\mathbf{W}_{m,n} = \boldsymbol{\Sigma}_{m,n}^{-1}$. The matrix $\mathbf{W}_{m,n}$ changes at every iteration because it depends on matrix $\mathbf{H}$, which also depends on the iteration. More details regarding the Iterative LSE derivation can be found for the MUCSD algorithm in Section 3.3.4. The mathematical derivation that leads to (28) can also be inferred from that section.

$$\hat{\boldsymbol{\gamma}}_m^{j+1,\text{UCSD}} = \begin{bmatrix} \hat{\mathbf{p}_m}^{j+1} \\ c\delta\hat{t}_m^{\,j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_m^j \\ 0 \end{bmatrix} + (\mathbf{H}^{j\mathsf{T}}\mathbf{W}_{m,n}^j\mathbf{H}^j)^{-1}\mathbf{H}^{j\mathsf{T}}\mathbf{W}_{m,n}^j\mathbf{y}_{m,n}^j \tag{28}$$

3.2.5 Motivation for the Extension to N Cooperative users

The results and discussion provided in Section 5.2.1 suggest that there is a great motivation to extend this model to its massive version with $N \geq 1$.

3.3 Massive User-Centric Single Differentiation (MUCSD)

In this subsection, the model for the MUCSD algorithm, with N cooperative receivers, is mathematically derived. The MUCSD algorithm estimator is denoted as $\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}$ and it is defined in Section 3.3.4. The lower bound on the variance of the proposed estimator can be found in Section 3.3.3.

3.3.1 Observation Model

The parameter vector to estimate is $\boldsymbol{\gamma}_m$, which is associated to the reference receiver m. The vector $\boldsymbol{\gamma}_n$ is associated to a cooperative receiver n, where $n = 1, \ldots, N$, being N the number of available cooperative receivers and thus the number of available user pairs to perform massive differential positioning. Consequently, the total number of users in the scenario is $N + 1$. The model in (25) can be expanded to the case of N user pairs in a straightforward manner as

$$\begin{aligned} \mathbf{y} = \begin{bmatrix} \mathbf{y}_{m,1} \\ \mathbf{y}_{m,2} \\ \vdots \\ \mathbf{y}_{m,N} \end{bmatrix} &= \begin{bmatrix} \Delta\boldsymbol{\rho}_{m,1} \\ \Delta\boldsymbol{\rho}_{m,2} \\ \vdots \\ \Delta\boldsymbol{\rho}_{m,N} \end{bmatrix} - \mathbf{1}_{N\times 1} \otimes \mathbf{h}(\boldsymbol{\gamma}_{0,m}) + \begin{bmatrix} \mathbf{h}(\bar{\boldsymbol{\gamma}}_1) \\ \mathbf{h}(\bar{\boldsymbol{\gamma}}_2) \\ \vdots \\ \mathbf{h}(\bar{\boldsymbol{\gamma}}_N) \end{bmatrix} + \mathbf{A}\boldsymbol{\gamma}_{0,m} \\ &= \mathbf{A}\boldsymbol{\gamma}_m + \begin{bmatrix} \boldsymbol{\epsilon}_{m,1} \\ \boldsymbol{\epsilon}_{m,2} \\ \vdots \\ \boldsymbol{\epsilon}_{m,N} \end{bmatrix}, \end{aligned} \tag{29}$$

where $\mathbf{A} = \mathbf{1}_{N\times 1} \otimes \mathbf{H}$. Again, due to the geometry of the problem, it can be assumed that the $\mathbf{H}$ matrices are approximately the same for all the users. Considering that $\bar{\boldsymbol{\Gamma}}_{1:N} = \begin{bmatrix} \bar{\boldsymbol{\gamma}}_1 & \bar{\boldsymbol{\gamma}}_2 & \ldots & \bar{\boldsymbol{\gamma}}_N \end{bmatrix}^{\mathsf{T}}$, the extended observation model can be expressed as

$$\mathbf{y} = \Delta\boldsymbol{\rho} - \mathbf{1}_{N\times 1} \otimes \mathbf{h}(\boldsymbol{\gamma}_{0,m}) + \mathbf{h}(\bar{\boldsymbol{\Gamma}}_{1:N}) = \mathbf{A}\left(\boldsymbol{\gamma}_m - \boldsymbol{\gamma}_{0,m}\right) + \boldsymbol{\epsilon}. \tag{30}$$

These observations can be modeled as $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = \mathbf{A}\boldsymbol{\gamma}_m$ if the term $\mathbf{A}\boldsymbol{\gamma}_{0,m}$ is moved to the left side of the expression in (30). The error covariance matrix corresponds to $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathsf{T}}]$. The mathematical expression used to compute this matrix is derived in the following subsection, as it differs from the one obtained with the UCSD algorithm in Section 3.2.2.

3.3.2 Error Covariance Matrix

With MUCSD, the measurements of the cooperative users are combined with the measurements of the reference user N times, as we have a total of N user pairs. This adds

more correlation to the error covariance matrix $\mathbf{\Sigma}$, which now is colored not only by the noise term $\boldsymbol{\epsilon}_{\boldsymbol{\gamma}_n} = \mathbf{H}\boldsymbol{\eta}_{\gamma_n} \sim \mathcal{N}(0, \sigma^2_{\boldsymbol{\gamma}_n}\mathbf{H}\mathbf{H}^\mathsf{T})$ but also by the combination of the pseudorange code errors of users m and n. In the following lines, we are calculating which is the contribution of this correlation to the error covariance matrix.

First, the resulting error term from the extended observation model in (30) must be properly defined. The error vector $\boldsymbol{\epsilon}$ can be expressed as

$$\boldsymbol{\epsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_m \\ \vdots \\ \boldsymbol{\epsilon}_m \end{bmatrix} - \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_N \end{bmatrix} - \begin{bmatrix} \mathbf{H}\boldsymbol{\eta}_{\gamma_1} \\ \vdots \\ \mathbf{H}\boldsymbol{\eta}_{\gamma_N} \end{bmatrix} = \mathbf{1}_{N\times 1} \otimes \boldsymbol{\epsilon}_m - \boldsymbol{\epsilon}_{1:N} - (\mathbf{I}_N \otimes \mathbf{H})\boldsymbol{\eta}_{\gamma_{1:N}}. \tag{31}$$

Therefore, the error covariance matrix has dimension $NK \times NK$ and can be calculated as

$$\begin{aligned} \mathbf{\Sigma} = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\mathsf{T}] = \mathbb{E}[(\mathbf{1}_{N\times 1} \otimes \boldsymbol{\epsilon}_m)(\mathbf{1}_{N\times 1} \otimes \boldsymbol{\epsilon}_m)^\mathsf{T}]\ ^{(a)} \\ - \mathbb{E}[\boldsymbol{\epsilon}_{1:N}\boldsymbol{\epsilon}_{1:N}^\mathsf{T}]\ ^{(b)} \\ - \mathbb{E}[((\mathbf{I}_N \otimes \mathbf{H})\boldsymbol{\eta}_{\gamma_{1:N}})((\mathbf{I}_N \otimes \mathbf{H})\boldsymbol{\eta}_{\gamma_{1:N}})^\mathsf{T}]\ ^{(c)}. \end{aligned} \tag{32}$$

Each of the three terms that appear in (32) are derived in the following lines. They describe the contribution to $\mathbf{\Sigma}$ of the pseudorange code error of the reference user m, in (32) (a), the pseudorange code error of the cooperative receivers $n = 1, \ldots, N$, in (32) (b), and the error from the coordinates provided by the cooperative users, in (32) (c).

Regarding the first term, (32) (a), and considering that the code error of the reference receiver m is modeled as $\boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \sigma^2_m\mathbf{I}_K)$, we have that

$$\begin{aligned} \mathbb{E}[(\mathbf{1}_{N\times 1} \otimes \boldsymbol{\epsilon}_m)(\mathbf{1}_{N\times 1} \otimes \boldsymbol{\epsilon}_m)^\mathsf{T}] = \mathbb{E}[(\mathbf{1}_{N\times 1} \otimes \boldsymbol{\epsilon}_m)(\mathbf{1}_{N\times 1}^\mathsf{T} \otimes \boldsymbol{\epsilon}_m^\mathsf{T})] = \\ \mathbb{E}[(\mathbf{1}_{N\times 1}\mathbf{1}_{N\times 1}^\mathsf{T}) \otimes (\boldsymbol{\epsilon}_m\boldsymbol{\epsilon}_m^\mathsf{T})] = \mathbf{J}_{N\times N} \otimes \mathbb{E}[\boldsymbol{\epsilon}_m\boldsymbol{\epsilon}_m^\mathsf{T}] = \mathbf{J}_{N\times N} \otimes \sigma^2_m\mathbf{I}_K, \end{aligned} \tag{33}$$

where $\mathbf{J}_{N\times N}$ is the all-ones matrix with dimension $N \times N$. Regarding the second term, (32) (b), and if it is assumed that the code errors are *i.i.d.* between cooperative receivers and also between satellites of a same receiver, the error vector $\boldsymbol{\epsilon}_n$ can be modeled as $\boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2_n\mathbf{I}_K)$. It is also assumed that σ^2_n has the same value for all the cooperative receivers. Consequently, the term (32) (b) can be developed as $\mathbb{E}[\boldsymbol{\epsilon}_{1:N}\boldsymbol{\epsilon}_{1:N}^\mathsf{T}] = \sigma^2_n\mathbf{I}_{NK}$, which matches with the dimension of $\mathbf{\Sigma}$ being $NK \times NK$.

Finally, regarding the third term and assuming that the position errors of the different cooperative users are identically distributed, which means that the accuracy of their positioning algorithms is the same ($\sigma^2_{\gamma_n}$ has the same value for all cooperative users), and also that they are independent, the third term, (32) (c), can be developed as

$$
\begin{aligned}
&\mathbb{E}[((\mathbf{I}_N \otimes \mathbf{H})\boldsymbol{\eta}_{\gamma_{1:N}})((\mathbf{I}_N \otimes \mathbf{H})\boldsymbol{\eta}_{\gamma_{1:N}})^{\mathsf{T}}] \\
&= \mathbb{E}\left[\mathbf{H}\boldsymbol{\eta}_{\gamma_1} \quad \cdots \quad \mathbf{H}\boldsymbol{\eta}_{\gamma_N}\right]^{\mathsf{T}}\left[\mathbf{H}\boldsymbol{\eta}_{\gamma_1} \quad \cdots \quad \mathbf{H}\boldsymbol{\eta}_{\gamma_N}\right] \\
&= \mathbb{E}[(\mathbf{I}_N \otimes \mathbf{H})\boldsymbol{\eta}_{\gamma_{1:N}}\boldsymbol{\eta}_{\gamma_{1:N}}^{\mathsf{T}}(\mathbf{I}_N^{\mathsf{T}} \otimes \mathbf{H}^{\mathsf{T}})] \\
&= (\mathbf{I}_N \otimes \mathbf{H})\mathbb{E}[\boldsymbol{\eta}_{\gamma_{1:N}}\boldsymbol{\eta}_{\gamma_{1:N}}^{\mathsf{T}}](\mathbf{I}_N^{\mathsf{T}} \otimes \mathbf{H}^{\mathsf{T}}) \\
&= \sigma_{\gamma_n}^2 \mathbf{I}_N \otimes \mathbf{H}\mathbf{H}^{\mathsf{T}}.
\end{aligned}
\tag{34}
$$

Taking into account all these expressions, and if $\sigma_m = \sigma_n = \sigma$, the error covariance matrix of the Massive SD model can be expressed as

$$
\boldsymbol{\Sigma} = \sigma^2(\mathbf{J}_{N\times N} \otimes \mathbf{I}_K + \mathbf{I}_{NK}) + \sigma_{\gamma_n}^2 \mathbf{I}_N \otimes \mathbf{H}\mathbf{H}^{\mathsf{T}}. \tag{35}
$$

Heatmap figures created from $\boldsymbol{\Sigma}$ for $K = 6$ satellites, $N = 3$ cooperative users, $\sigma = 5$ and $\sigma_{\gamma_n} = [0, 5]$ meters are shown in Figures 20 and 21.

3.3.3 CRB

The observation model that we have is the one in (30) and the parameter vector to estimate corresponds to $\boldsymbol{\gamma}_m$. The lower bound on the estimator $\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}$ is derived by computing the inverse of the Fisher Information Matrix (FIM), which is calculated as

$$
\mathcal{I}^{\text{MUCSD}}(\boldsymbol{\gamma}_m) = -\mathbb{E}\left[\frac{\partial^2 \ln f(\mathbf{y}|\boldsymbol{\gamma}_m)}{\partial \boldsymbol{\gamma}_m \partial \boldsymbol{\gamma}_m^{\mathsf{T}}}\right], \tag{36}
$$

where the likelihood function that describes the joint probability of the observation data $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ modeled in (30) as a function of the parameter vector $\boldsymbol{\gamma}_m$ can be expressed as

$$
f(\mathbf{y}|\boldsymbol{\gamma}_m) = \frac{1}{\sqrt{(2\pi)^N \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{A}\boldsymbol{\gamma}_m)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{A}\boldsymbol{\gamma}_m)\right). \tag{37}
$$

The log-likelihood function is calculated as

$$
\begin{aligned}
\ln f(\mathbf{y}|\boldsymbol{\gamma}_m) &= C - \frac{1}{2}(\mathbf{y} - \mathbf{A}\boldsymbol{\gamma}_m)^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{A}\boldsymbol{\gamma}_m) \\
&= C - \frac{1}{2}(\mathbf{y}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y} + \boldsymbol{\gamma}_m^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}\boldsymbol{\gamma}_m - 2\mathbf{y}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}\boldsymbol{\gamma}_m),
\end{aligned}
\tag{38}
$$

where $C = -\frac{1}{2}\ln\left((2\pi)^N \det(\boldsymbol{\Sigma})\right)$ is a constant that does not influence the CRB derivation. If we compute the first derivative of the log-likelihood function with respect to the parameter vector $\boldsymbol{\gamma}_m$, we obtain

$$\frac{\partial \ln f_{\boldsymbol{\gamma}_m}(\mathbf{y})}{\partial \boldsymbol{\gamma}_m} = -\frac{1}{2}\left(2\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}\boldsymbol{\gamma}_m - 2\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{y}\right). \tag{39}$$

The 4×4 FIM with respect to the parameter vector $\boldsymbol{\gamma}_m$, denoted as $\mathcal{I}(\boldsymbol{\gamma}_m)^{\text{MUCSD}}$, is therefore obtained as

$$\mathcal{I}^{\text{MUCSD}}(\boldsymbol{\gamma}_m) = -\mathbb{E}\left[\frac{\partial^2 \ln f(\mathbf{y}|\boldsymbol{\gamma}_m)}{\partial \boldsymbol{\gamma}_m \partial \boldsymbol{\gamma}_m^{\mathsf{T}}}\right] = \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}. \tag{40}$$

As a result, a lower bound can be placed on the variance of the estimator of each element in $\boldsymbol{\gamma}_m$. The estimator of element i in $\boldsymbol{\gamma}_m$ is denoted as $\left[\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}\right]_i$. The CRB for element i corresponds to the $[i,i]$ element of the FIM inverse as

$$\text{var}\left(\left[\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}\right]_i\right) \geq \text{CRB}\left(\left[\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}\right]_i\right) = \left[\left(\mathcal{I}^{\text{MUCSD}}(\boldsymbol{\gamma}_m)\right)^{-1}\right]_{ii} = \left[(\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A})^{-1}\right]_{ii}. \tag{41}$$

According to [48] (p. 35), if the measurements provided by each pair of users are *i.i.d.*, the CRB of each element may be calculated as

$$\text{CRB}\left(\left[\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}\right]_i\right) = \frac{1}{N}\text{CRB}\left(\left[\hat{\boldsymbol{\gamma}}_m^{\text{UCSD}}\right]_i\right) = \frac{1}{N}\left[(\mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}_{m,n}^{-1}\mathbf{H})^{-1}\right]_{ii}, \tag{42}$$

which corresponds to the CRB for the UCSD algorithm (see the expression in (27)) inversely weighted by the number of cooperative users in the scenario. As the measurements provided by each pair of users are not independent because the measurements of the reference user are present in each SD combination. Consequently, the CRB must be computed following the expression in (41).

3.3.4 Iterative LSE

The estimator $\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}}$ takes into account all the cooperative users added to the scenario and can be obtained from the full matrix model found in (30). As this observation model is linear, we can use a Least Squares Estimator (LSE) to estimate $\boldsymbol{\gamma}_m$ as

$$\hat{\boldsymbol{\gamma}}_m^{\text{MUCSD}} = (\mathbf{1}_{N\times 1} \otimes \mathbf{H})^{\dagger}\mathbf{y} = \mathbf{A}^{\dagger}\mathbf{y}, \tag{43}$$

where $(\cdot)^{\dagger}$ refers to the pseudoinverse, or the Moore-Penrose inverse, which can be defined as $\mathbf{A}^{\dagger} = (\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}$. This expression can be obtained by minimizing the cost function of $\boldsymbol{\gamma}_m$, which we refer to as $\mathcal{J}(\boldsymbol{\gamma}_m)$. As the observations are Gaussian and the model has been linearized, the LSE can also be obtained by maximizing the likelihood function of $\boldsymbol{\gamma}_m$. This is equivalent to setting the expression in (39) to zero.

As it has been previously stated, the covariance matrix of the observation model error is colored as $\boldsymbol{\Sigma} = \sigma^2(\mathbf{J}_{N\times N} \otimes \mathbf{I}_K + \mathbf{I}_{NK}) + \sigma_{\gamma_n}^2 \mathbf{I}_N \otimes \mathbf{H}\mathbf{H}^{\mathsf{T}}$. As a consequence, a Weighted Least Squares (WLS) problem needs to be solved and the estimator of $\boldsymbol{\gamma}_m$ can be computed as

$$\hat{\boldsymbol{\gamma}}_m^{\mathrm{MUCSD}} = (\mathbf{A}^\mathsf{T}\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{W}\mathbf{y}, \tag{44}$$

where $\mathbf{W} = \boldsymbol{\Sigma}^{-1}$. Due to the non-linearity in the pseudorange measurement expression, this is a non-linear Least Squares problem. Consequently, it is necessary to solve the estimator iteratively by linearizing the function at some initial guess $\boldsymbol{\gamma}_{0,m}$. A new estimate can be computed as

$$\hat{\boldsymbol{\gamma}}_m^{j+1,\mathrm{MUCSD}} = \begin{bmatrix} \hat{\mathbf{p}_m}^{j+1} \\ c\delta\hat{t}_m^{\ j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_m^j \\ 0 \end{bmatrix} + (\mathbf{A}^{j\mathsf{T}}\mathbf{W}^j\mathbf{A}^j)^{-1}\mathbf{A}^{j\mathsf{T}}\mathbf{W}^j\mathbf{y}^j, \tag{45}$$

where $\mathbf{p}^j$ corresponds to the position estimated in the previous iteration. The matrix $\mathbf{W}^j$ changes at every iteration because it depends on matrix $\mathbf{A}$, which also depends on the iteration. The LOS vectors in matrix $\mathbf{A}^j$ are calculated considering the distances between the satellites and the estimated position $\mathbf{p}^j$. Also, $\mathbf{y}^j$ is the Observed minus Computed (O-C) value, also called prefit, of the observations and is calculated as (see also (30))

$$\begin{aligned} \mathbf{y}_j &= \Delta\boldsymbol{\rho} - \mathbf{1}_{N\times 1} \otimes \mathbf{h}(\boldsymbol{\gamma}_{0,m}^j) + \mathbf{h}(\bar{\boldsymbol{\Gamma}}_{1:N}) + \mathbf{A}\boldsymbol{\gamma}_{0,m}^j \\ &= \begin{bmatrix} \Delta\boldsymbol{\rho}_{m,1} \\ \vdots \\ \Delta\boldsymbol{\rho}_{m,N} \end{bmatrix} - \mathbf{1}_{N\times 1} \otimes \mathbf{h}(\boldsymbol{\gamma}_{0,m}^j) + \begin{bmatrix} \mathbf{h}(\bar{\boldsymbol{\gamma}}_1) \\ \vdots \\ \mathbf{h}(\bar{\boldsymbol{\gamma}}_N) \end{bmatrix} + \mathbf{A}\boldsymbol{\gamma}_{0,m}^j \\ &= \begin{bmatrix} \Delta\rho_{m,1}^1 \\ \vdots \\ \Delta\rho_{m,1}^K \\ \vdots \\ \Delta\rho_{m,N}^1 \\ \vdots \\ \Delta\rho_{m,N}^K \end{bmatrix} - \begin{bmatrix} \mathbf{h}^1(\boldsymbol{\gamma}_{0,m}^j) \\ \vdots \\ \mathbf{h}^K(\boldsymbol{\gamma}_{0,m}^j) \\ \vdots \\ \mathbf{h}^1(\boldsymbol{\gamma}_{0,m}^j) \\ \vdots \\ \mathbf{h}^K(\boldsymbol{\gamma}_{0,m}^j) \end{bmatrix} + \begin{bmatrix} \mathbf{h}^1(\bar{\boldsymbol{\gamma}}_1) \\ \vdots \\ \mathbf{h}^K(\bar{\boldsymbol{\gamma}}_1) \\ \vdots \\ \mathbf{h}^1(\bar{\boldsymbol{\gamma}}_N) \\ \vdots \\ \mathbf{h}^K(\bar{\boldsymbol{\gamma}}_N) \end{bmatrix} + \mathbf{A}\boldsymbol{\gamma}_{0,m}^j. \end{aligned} \tag{46}$$

The value of the geometric range of the reference user m, $\rho^{1:K}(\boldsymbol{\gamma}_{0,m}^j)$, is updated at every iteration with the last estimated value of $\mathbf{p}^j$. The value of the geometric range of the cooperative users, $\rho^{1:K}(\hat{\boldsymbol{\gamma}}_{1:N})$, is the same over all the WLS iterations because the information regarding the position of the cooperative users is not updated. To compute these geometric ranges, the small angle approximation is applied to the rotation of the Earth during the signal transit time as suggested in (8.35) from [3].

The LSE, regardless of any probabilistic assumption made about the observations, corresponds to the value of $\boldsymbol{\gamma}$ that minimizes the cost function $\mathcal{J}(\boldsymbol{\gamma})$. The method is equally valid for Gaussian as well as non-Gaussian noise [48]. Nevertheless, the performance of the method depends on the properties of the corrupting noise and the modeling errors. If the observations belong to a Gaussian distribution and are *i.i.d.* and linear, the LSE is equivalent to the Maximum Likelihood Estimator (MLE), which is asymptotically efficient. If the observations are Gaussian but not *i.i.d.*, the WLS is equivalent to the MLE.

For the proposed observation model, the described WLS estimator is equivalent to the MLE and therefore it is asymptotically efficient. It has been proven that the implemented estimator is asymptotically efficient because the mean tends to zero and the variance to the CRB throughout the Monte Carlo iterations. Also, due to the initial model being non-linear, the model has been linearized and consequently it is needed to iterate the LS solution. This is why we are working with an Iterative WLS algorithm.

4 Methods

A general description of the setup used to conduct the experiments of this Master Thesis is provided at the beginning of this section. In Table 4, the computational resources used for the execution of our script are listed. After that, details regarding the simulator that has been employed, which we have named MassiveCoop-Sim, are provided to the reader. In Table 5, we can find a list of the modules implemented in our simulator. Finally, important aspects in relation to the conducted experiments are given in the last subsection.

4.1 General Setup Description

The experiments described in this thesis have been conducted with the MATLAB version 0.9.11 (R2021b) [49]. Due to the high computational cost of some of the experiments, it has been required to use the Discovery Cluster from Northeastern University, in Boston, which is operated by the university's Research Computing team. The Discovery cluster is located in the Massachusetts Green High Performance Computing Center in Holyoke, MA [2].

We have built a simulator which we refer to as MassiveCoop-Sim. This simulator is able to generate a scenario with N users uniformly distributed within a spherically defined space. It also computes the GNSS measurements of the N users in order to estimate their positions. The main contribution of this research is included in an additional module of the simulator, which aims to estimate the user positions in an enhanced manner with respect to the widely known DGNSS technique.

The simulator in [3] is taken as a reference for the implementation of the initial modules used in MassiveCoop-Sim. The initial modules cover the simulation of the scenario and the acquisition of GNSS pseudorange measurements, for which the user and satellite positions are needed in order to calculate the LOS vectors.

An accurate description of the modules that belong to MassiveCoop-Sim is later provided. In this section we are only referring to the parameter values and simulator features that are common for all the final experiments. The parameter values that are experiment-dependent are provided together with their description and results in further sections.

4.1.1 Use of NEU's Discovery Cluster

A description of the computational resources used for the execution of our .m scripts in the Discovery cluster can be found in Table 4. It should be noted that the resources used are not particularly powerful. This is because our purpose when using the cluster was to be able to run several .m scripts in parallel in order to save time. An evaluation of the computational efficiency of the code is not an object of study of this research. However, this matter is included in Section 6 as it would be relevant if testing the algorithm in a real scenario, for future stages of the project.

Table 4: Computational resources used for the execution of our .m scripts in the Discovery cluster from Northeastern University, Boston [2].

Description	Size
RAM	10 GB
CPUs	1
Cores	2
Partition Size	Small
Parallel Run	Activated

4.2 MassiveCoop-Sim Simulator

The main modules of the MassiveCoop-Sim simulator are listed in Table 5. In the following subsections, details about these modules are provided so that the reader can understand which has been the procedure followed to implement the algorithms proposed in this Master Thesis.

Table 5: Main modules of the MassiveCoop-Sim simulator, which takes as a reference the simulator provided in [3].

Module ID	Purpose
1	Build the N-users scenario.
3	Generate GNSS pseudorange measurements.
4	Benchmark (CRB calculation).
5	Estimate user positions with novel technique.
6	Plotting and Debugging.

4.2.1 GNSS Configuration

The GNSS configuration is first defined with parameters such as the number of satellites in the constellation and the orbital radius and inclination angle of the satellites. GNSS receivers can be set to ignore satellite signals with an angle above the horizon below a user-defined threshold, which corresponds to the mask angle and is also a setting of our simulator. The procedure followed to calculate which are the satellites in view for each user is further described, as some notation and knowledge needs to be introduced before.

We have set the number of satellites in the constellation to 30, the satellite orbital radius to 2.656175e7 meters and their inclination angle to 55°. These values have been taken from the GNSS configuration of one of the setups provided in the simulator in [3], as the geometry proposed by these setups has proven to be realistic. The mask angle has been set to 15°, being the most typical mask angle in GPS receivers between 10° and 15°.

4.2.2 User Distribution

It is essential to define a scenario that allows to test the proposed technique in the final experiments. For this research, we implement a specific module in our simulator to build

the defined scenario. First of all, some constants must be defined, such as the total number of users in the scenario and the maximum distance between them, which we refer to as $\mathbf{\Delta}_{\mathrm{MAX}}$. Also, the proposed estimation techniques are evaluated under several values of standard deviation of the code error. These standard deviation values can be selected as well. A constant with the number of Monte Carlo realizations is also defined. As the experiments have been run using a cluster, a large enough number of Monte Carlo realizations has been selected for every case (1e4).

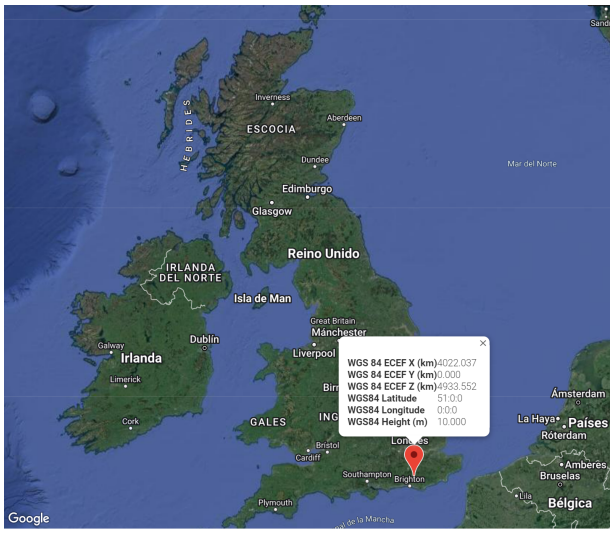

Figure 11: Location of the firstly generated user, which is used as a reference for the distribution of the rest of the users within the available space. For the selected setup, the position of this user is $\mathbf{p} = [4.022036955287312e6,\ 0,\ 4.933552391696703e6]$ meters in ECEF coordinates [37].

We consider relevant to describe the manner used to distribute the users within the available space, which we have defined to be spherical in order to make the scenario more realistic. The position of a firstly generated user is taken as a reference. For the selected setup, the position of this user is $\mathbf{p} = [4.022036955287312e6,\ 0,\ 4.933552391696703e6]$ meters in Earth-Centered, Earth-Fixed (ECEF) or Geocentric system coordinates [37]. As shown in Figure 11, the values of these coordinates match with a location nearby the city of Brighton, in the United Kingdom. We have considered interesting to select a real location and define it in this document in order to show the applicability of our experiments in a real life use case. We are assuming that users are uniformly distributed within a sphere, being its center the firstly generated user with coordinates $\mathbf{p}$. The impact of the values in this vector is later described. When necessary, the ionospheric and tropospheric conditions are set to values that are suitable for our experiments, which do not necessarily match the conditions experienced at the defined user location. Although unrealistic, this allows to properly test the performance of the proposed technique when mitigating such atmospheric effects.

The simulator in [3] contains several .csv files with user profiles including their positions at different time instants. For our setup, only a first position $\boldsymbol{\gamma}$ is extracted from one of these files. The positions associated to the rest of users are calculated afterwards with our simulator. The values extracted from the .csv file are actually in Local north, east,

down (NED) coordinates, and they are converted to ECEF with a function also provided by the simulator in [3]. As it can also be seen in Figure 11, the latitude and longitude of the first user location is 51° and 0 °, respectively, being 10 meters high. In the modules adapted for our simulator, we are only working with ECEF coordinates. However, the NED coordinates are always used when calculating the contribution that Earth rotation has on the GNSS pseudorange measurements.

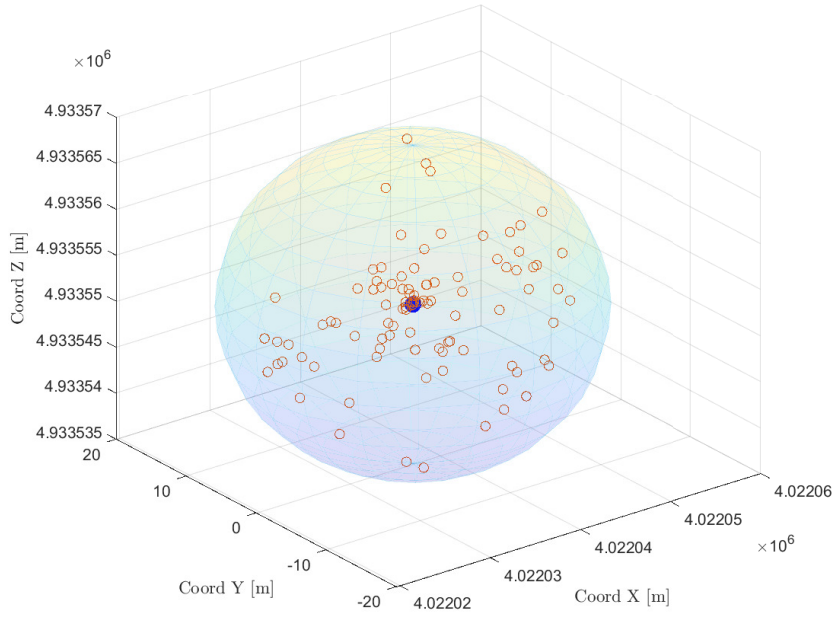

Figure 12: Distribution of 101 users within the defined spherical space for $\boldsymbol{\Delta}_{\mathrm{MAX}} = [10, 10, 10]$ m. In this case, the blue marker can be seen due to the low density of users.

One of the modules added to our simulator allows to plot the distribution of users along the defined spherical space. A sphere with center $c = \boldsymbol{\gamma}$ and radius $r\,(\boldsymbol{\Delta}_{\mathrm{MAX}})$ is defined in space. This means that the coordinates of the sphere center correspond to the position of the firstly generated user $\boldsymbol{\gamma}$. The vector $\boldsymbol{\Delta}_{\mathrm{MAX}}$ contains the maximum distance that a user can have with respect to the firstly generated user in the three components of the cartesian coordinates system, meaning that the maximum Euclidean distance between a user position $\boldsymbol{\gamma}_n$ and the firstly generated user $\boldsymbol{\gamma}$ is $\|\boldsymbol{\Delta}_{\mathrm{MAX}}\|$. For example, for a realistic setup where $\boldsymbol{\Delta}_{\mathrm{MAX}} = [10, 10, 10]$ m, we have that the Euclidean distance between a user that has been positioned in the farthest location and the firstly generated user corresponds to $\|\boldsymbol{\Delta}_{\mathrm{MAX}}\| = 10\sqrt{3} \approx 17.32$ m.

One of the aims of this module in our simulator is to generate the position vectors of N additional users, so that in the end we have a total of $N+1$ position vectors. To compute a random point in a sphere, we first generate three random values, which correspond to the three cartesian coordinates, and store them in a vector $\mathbf{u} = (x, y, z)$. For example, these can be three realizations of a normal distribution with $\mu = 0$ and $\sigma = 1$. After that, a normalization constant is calculated as $\alpha = \frac{r}{\|\mathbf{u}\|}$, being r a randomly generated number

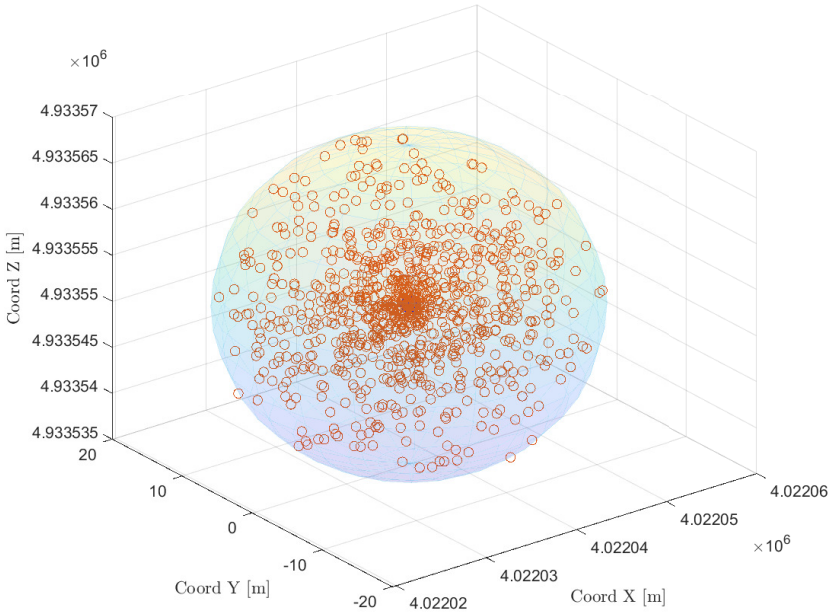

Figure 13: Distribution of 1001 users within the defined spherical space for $\boldsymbol{\Delta}_{\text{MAX}} = [10, 10, 10]$ m. The firstly generated user with position coordinates $\boldsymbol{\gamma}$ is highlighted with a blue marker (not visible due to the high user density). The rest of the 1000 users are highlighted with markers of orange edge color.

between 0 and 1 following an uniform distribution. Finally, the three components of $\mathbf{u}$ are multiplied by the normalization constant and also by the desired sphere radius, which in our case corresponds to $\|\boldsymbol{\Delta}_{\text{MAX}}\|$. The center of the sphere must also be added up to this result. In Figure 13 we can see the distribution of 1001 users within the defined spherical space. The firstly generated user with position coordinates $\boldsymbol{\gamma}$ is highlighted with the blue marker. The rest 1000 users are highlighted with markers of orange edge color. It can be observed that the users are uniformly distributed within the defined sphere.

Another term to consider for further analysis is what we call the *inter-user distance* $d_{m,n}$ between users m and n. As previously mentioned, the maximum value of $d_{1,n}$ for any value of n corresponds to $\|\boldsymbol{\Delta}_{\text{MAX}}\|$. In Figure 15, we can see the distribution of $d_{1,n}$ for $N = 1000$. It is shown that $d_{1,n} \sim \mathcal{U}(0,\ \|\boldsymbol{\Delta}_{\text{MAX}}\|)$ for big values of N. The distribution is less flat due to the high value assigned to the parameter *bins* when plotting the histogram, but it is considered that the distribution is uniform. The total number of inter-user distances that can be computed in a scenario with N users corresponds to $\binom{N}{2} = \frac{N!}{2!(N-2)!} = \frac{N(N-1)}{2}$ (combination without repetition of N total elements and sample size 2).

Also, in Figures 16 and 17 we can see the inter-user distances in a newly generated scenario with 4 users. The distances between users $d_{m,n}$ for $m, n \neq 1$ can be higher than $\|\boldsymbol{\Delta}_{\text{MAX}}\|$. More specifically, $d_{m,n} \leq \|2\boldsymbol{\Delta}_{\text{MAX}}\| \approx 34.64$ m. Figure 17 shows the values of the 3D norms, while Figure 16 shows the values of the 2D norms in the planes defined by the

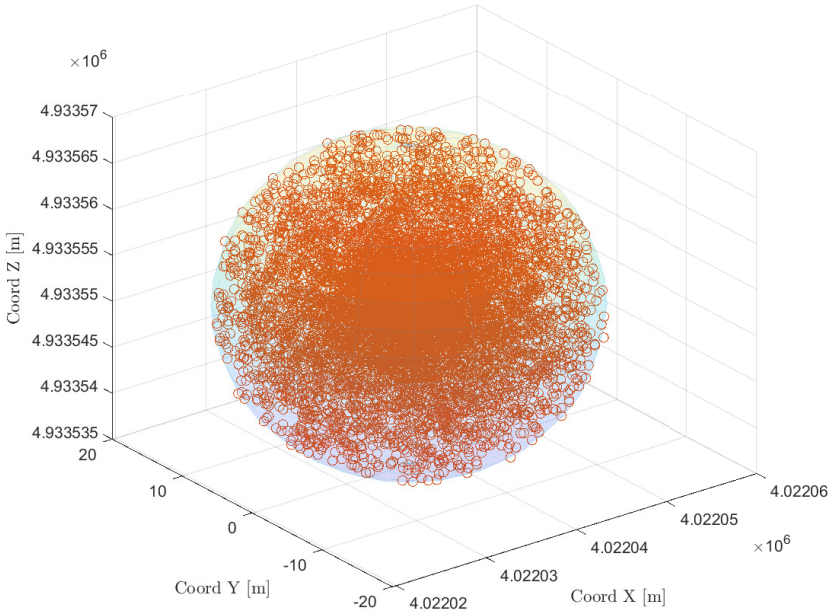

Figure 14: Distribution of 10001 users within the defined spherical space for $\mathbf{\Delta}_{\text{MAX}} = [10, 10, 10]$ m. The density of users in this scenario is extremely high.

three possible combinations of cartesian coordinates (XY, XZ, YZ). For these figures, a scenario with lower number of users has been simulated so that the distribution of users is more illustrative.

4.2.3 Mask Angle filtering

In the simulator, the user positions are used to compute the euclidean distances between the users and the satellites in view, which play a role in the calculation of the LOS vectors needed to compute the GNSS pseudorange measurements. It is assumed that all the users considered for the experiments are under open-sky conditions and have commonly in view enough satellites to solve the navigation equations (4 satellites). Having this minimum number of common satellites in view is a necessary condition to perform single differentiation of pseudorange measurements. Given the position of a user $\mathbf{p} = [x,\ y,\ z]^{\mathsf{T}}$, we can calculate which satellites are in view for this user. To calculate whether a satellite k with position $\mathbf{p}^k$ is in view for user n, we first compute the ECEF-to-NED coordinate transformation matrix given the latitude λ and longitude φ of user n (see (47) [41]). Then we determine the ECEF LOS vector from user m to satellite k as indicated in 8, which we refer to as $\mathbf{LOS}_m^k$. This vector is converted to NED coordinates as well. With the $\mathbf{LOS}_m^k$ vector in NED coordinates, we can describe the direction of each GNSS satellite from the user antenna by calculating its elevation θ and azimuth ψ angles as indicated in (48) and (49), respectively. In these equations, the unit vectors in NED coordinates are denoted as $[\hat{\mathbf{e}}, \hat{\mathbf{n}}, \hat{\mathbf{u}}]$. Finally, it is determined whether the satellite elevation angle is above the masking angle. If it is, this satellite is considered to be in view of the user.

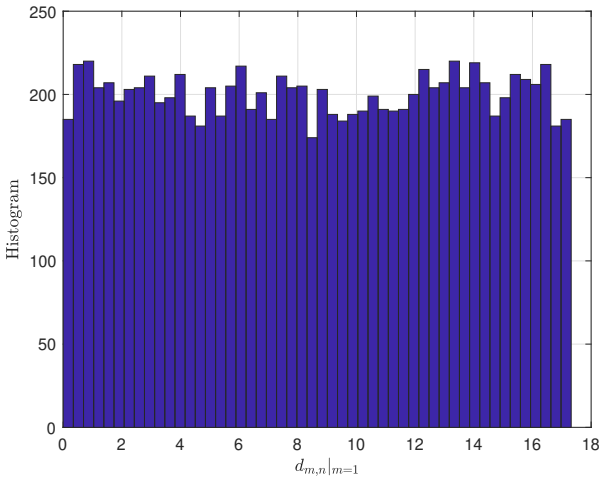

Figure 15: Distribution of $d_{1,n}$ for $N = 10000$ and $\mathbf{\Delta}_{\text{MAX}} = [10, 10, 10]$, showing that $d_{1,n} \sim \mathcal{U}\left(0,\ \|\mathbf{\Delta}_{\text{MAX}}\|\right)$ for big values of N. For this setup, $\|\mathbf{\Delta}_{\text{MAX}}\| = 17.32$ m.

$$\begin{bmatrix} e \\ n \\ u \end{bmatrix} = \begin{pmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\cos\lambda\sin\varphi & -\sin\lambda\sin\varphi & \cos\varphi \\ \cos\lambda\cos\varphi & \sin\lambda\cos\varphi & \sin\varphi \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5) \tag{47}$$

$$\theta = \arcsin\left(\mathbf{LOS}_m^k \cdot \hat{\mathbf{u}}\right) \tag{48}$$

$$\psi = \arctan\left(\frac{\mathbf{LOS}_m^k \cdot \hat{\mathbf{e}}}{\mathbf{LOS}_m^k \cdot \hat{\mathbf{n}}}\right) \tag{49}$$

4.2.4 Generation of Pseudorange Measurements

To compute the GNSS pseudorange code measurements, we follow the expression in (3). We are able to compute the pseudorange measurements because we have the knowledge of the position of each user and also of the pseudorange code error standard deviation, which is a setting of the simulator. When it is assumed that the position of the additional/cooperative users is known with an error with standard deviation σ_{γ_n}, this error is not included in the position used to compute the pseudorange measurement. This means that the only error included in the calculation of the code measurement is the code measurement error itself, regardless of the experimental setup. The term σ_{γ_n}, if different from zero, is used when computing the prefit of the position estimator. The same applies for the clock bias error. This is explained in more detail later, as the scope of this subsection is to talk about the genertion of pseudorange measurements exclusively.

The number of GNSS measurements computed in a code execution corresponds to $(N + 1) \times K \times M \times S$, being N the number of additional/cooperative users added to the scenario, K the number of satellites in view, M the number of Monte Carlo realizations that are set and S the number of values of code error standard deviation under which

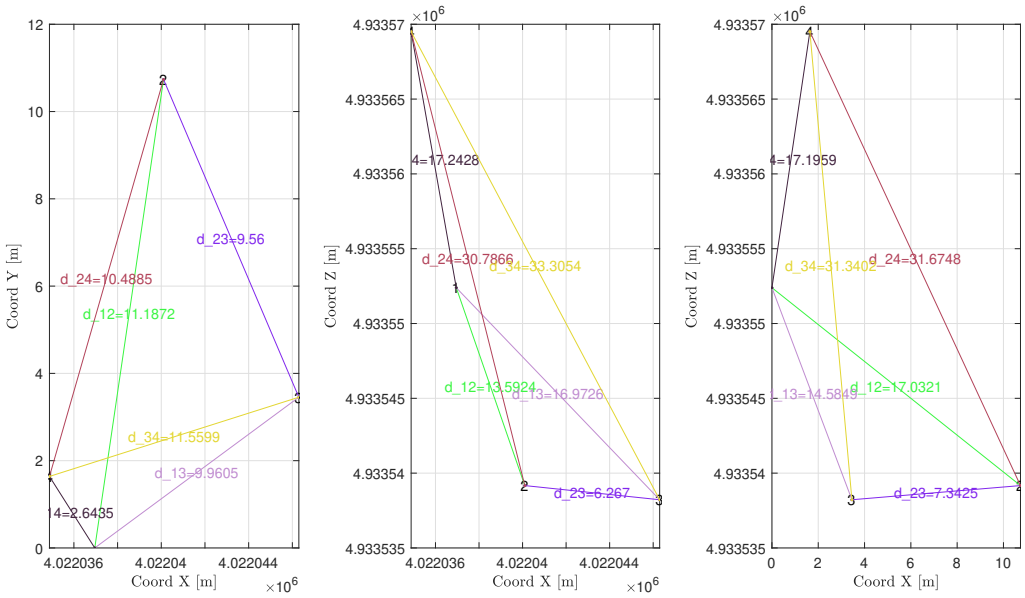

Figure 16: 2D norms of the inter-user distances $d_{m,n}$ in a simulated scenario with $N = 4$.

the proposed algorithms are tested. This means that, for instance, in a scenario with 6 satellites in view, 20 user pairs (therefore we take into account that we have a total of 21 users), 1000 Monte Carlo realizations and 10 values of code error standard deviation under study, we will generate 1.26e6 GNSS pseudorange measurements. It is interesting to mention this calculation as it gives an idea of how massive the scenario posed by the proposed techniques is.

Actually, this big magnitude is mostly given by the high amount of Monte Carlo realizations. In Figure 18, we can see a graphic that illustrates the additional dimension provided by the use of multiple Monte Carlo iterations in the MassiveCoop-Sim simulator. As indicated in this figure, the other dimensions are given by the pseudorange code error standard deviation σ and also by N and K. The values N and K are grouped together because we assume that for each user we need to calculate as many pseudorange measurements as satellites in view in the scenario.

For the generation of GNSS pseudorange measurements, the small angle approximation is applied to the rotation of the Earth during the signal transit time as suggested in (8.35) from [3]. This way, the output GNSS measurements are closer to the ones we would obtain in a real scenario under the same conditions. This matrix is used to increase the accuracy of the satellite position when computing the Euclidean distance between the satellite and the user.

4.2.5 Position Estimation

The main contribution of this research is included in the module that tackles the user position estimation. This module is able to work with both of the algorithms presented

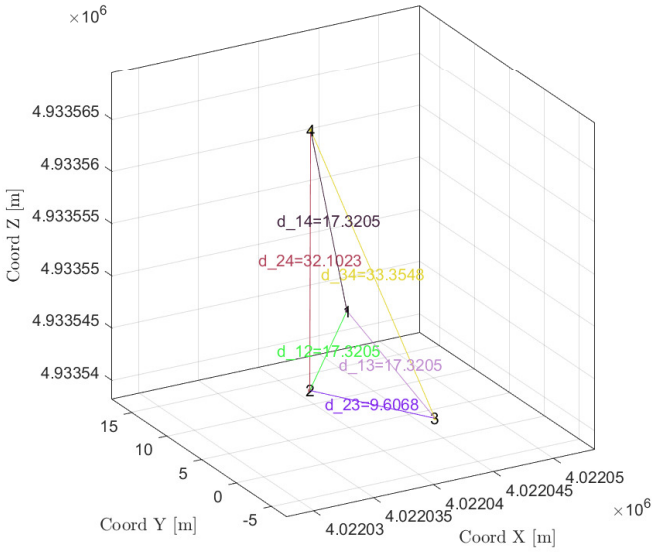

Figure 17: 3D norms of the inter-user distances $d_{m,n}$ in a simulated scenario with $N = 4$.

in next section: the User-Centric Single Differentiation (UCSD) algorithm and the Massive User-Centric Single Differentiation (MUCSD) algorithm. The proposed techniques make use of Single Differentiation (SD) of pseudorange code measurements. Thus, the first step to be performed in the position estimation is the computation of the difference of pseudorange measurements of the users. The number of GNSS measurements single differentiations computed in a code execution corresponds to $N \times K \times M \times S$. For example, in a scenario with 6 satellites in view, 20 user pairs (therefore we take into account that we have a total of 21 users), 1000 Monte Carlo realizations and 10 values of code error standard deviation under study, we will calculate 1.2e6 differentiations of pseudorange measurements. Although this sounds computationally expensive, this step is only subtracting the values that have been already computed in the previous step.

The step that is key corresponds to the actual estimator of the module. As the Weighted Least Squares (WLS) estimator is derived from a full matrix observation model that contains the SD of all the available user pairs (see (30)), we do not need to compute an Iterative LSE solution for each SD value obtained. The amount of LSE solutions that are to be computed is simply $M \times S$. If we had not integrated Monte Carlo to our system, we would be obtaining only one solution per value of code error standard deviation tested. Thanks to the integration of Monte Carlo in the MassiveCoop-Sim simulator, we can properly evaluate the performance of the proposed techniques.

Nevertheless, it is important to avoid being mistaken with regard to the computational expense of the estimation module of the MassiveCoop-Sim simulator. Although the number of position solutions is not affected by the number of additional/cooperative users introduced in the scenario N or the number of satellites K, these parameters have a very

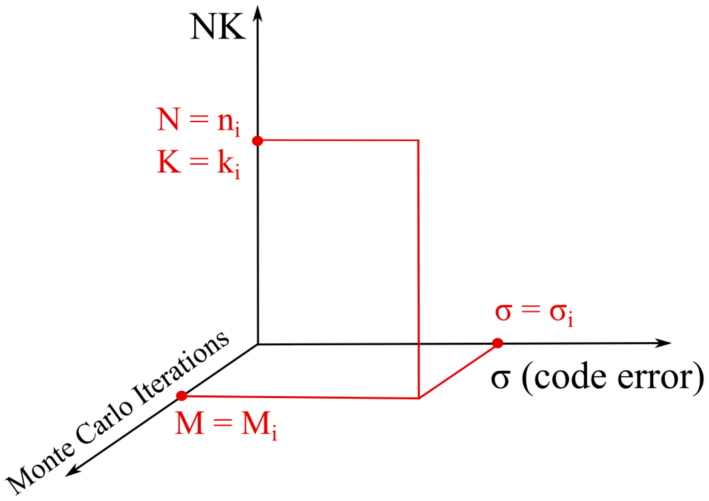

Figure 18: Graphic that illustrates the additional dimension provided by the use of Monte Carlo iterations in the MassiveCoop-Sim simulator. The other dimensions are given by the pseudorange code error standard deviation, the number of additional users added to the scenario N and the number of satellites in view K.

strong impact on the dimensions of the matrices used by these algorithms. The computational expense of the UCSD and MUCSD algorithms is strongly impacted by the error of the position of the cooperative users σ_{γ_n}. When this error is high, the algorithm struggles more to converge and the running time increases. Doubtlessly, the number of users introduced to the scenario is a parameter that strongly affects the computational expense of the MUCSD algorithm. Overall, the four parameters have an impact on the algorithm efficiency. It is of our interest to study in depth how to implement the proposed UCSD and MUCSD in an efficient manner, as this will be necessary if wanting to test them in a real scenario. In Section 6, this has been propoed as a future line of research.

4.2.6 Benchmark: CRB Calculation

The objective of this subsection is to stress the fact that in order to verify that results obtained with the LSE are correct, we have also computed the CRB of the proposed estimation problem. In the final experiments, we have plotted RMSE results obtained with the LSE together with the CRB, in order to show that the estimated position error matches the CBR. Mathematical derivation of the CRB is provided in Sections 3.2.3 and 3.3.3.

The CRB can be used with benchmarking purposes when comparing our results to the ones obtained with other techniques, in further lines of research. The CRB equations are also of our interest because they help when interpreting how errors introduced in the model affect the position estimation RMSE. Errors introduced in the model are the ones that help the modeling of noisy observations of the position and clock bias of cooperative users, and also GNSS pseudorange code errors.

For example, in Section 5.2.2, we describe how, depending on the ratio between σ and $\sigma^2_{\gamma_n}$, the term that dominates the covariance matrix $\mathbf{\Sigma}$ changes, and how this has an impact on the comparative between the MUCSD algorithm and DGNSS.

4.3 Experimental Setup

Three experiments have been conducted in this Master Thesis. Table 6 provides a link to the figures and tables of the results and setup of each experiment. The difference between Experiment B.1 and B.2 relies on the value of the error in the noisy observations of the cooperative users. The standard deviation of this error is higher for Experiment B.2 (10 meters), while in Experiment B.2 we assume that we have perfect knowledge of the cooperative user positions.

Table 6: Summary of the setup of the three experiments conducted in this research. A link to the corresponding result figures and experimental setup is provided in this table.

ID	Section	Experimental Setup	Results
A	5.2.1	Table 7	Figure 22
B	5.2.2	Table 8 and Table 10	Figure 23, Table 9 and Figure 24
C	5.2.3	Table 11	Figure 25

Experiment A is conducted with $N = 1$, which means that the executed algorithm is USCD (no massive scenario), while experiments B and C are conducted for higher values of N. The objective of Experiment A is to introduce the motivation of extending the UCSD model into the massive case, with the MUCSD. With Experiment B.1 and B.2 we see that bringing N users to the scenario allows us to have DGNSS performance without the need of a reference station, for realistic error in the noisy observations in the cooperative users. Finally, we see that for very pessimistic scenarios, with very high value of the noise in the position of cooperative users, our algorithm would need a too large value of N to provide this improvement in the performance. Table 12 summarizes conclusions extracted from the conducted experiments.

4.3.1 Performance Evaluation Metric (RMSE)

The metric used to evaluate the performance of the proposed algorithms is the position and clock bias estimation Root-Mean-Square Error (RMSE). Result figures provide values of estimation RMSE of both position cartesian coordinates and clock bias, which has been computed as

$$\mathrm{RMSE}\left(\hat{\boldsymbol{\gamma}}_m\right) = \sqrt{\mathrm{MSE}\left(\hat{\boldsymbol{\gamma}}_m\right)} = \sqrt{\mathbb{E}\left[\|\hat{\boldsymbol{\gamma}}_m - \boldsymbol{\gamma}_m\|^2\right]}, \tag{50}$$

where $\boldsymbol{\gamma}_m = \left[\mathbf{p}_m^{\mathsf{T}}, c\delta t_m\right]^{\mathsf{T}}$, being $\mathbf{p}_m = [x_m,\ y_m,\ z_m]^{\mathsf{T}}$ the cartesian coordinates of the reference user m and δt_m its clock bias.

5 Results and Discussion

Experimental results are provided in this section. First, heatmap figures of the error covariance matrices from the UCSD and MUCSD models are provided. After that, position estimation RMSE values for Experiments A, B and C are shown. The lower bounds on the variance of the proposed estimators, which have been calculated with the CRB, are also presented.

5.1 Error Covariance Matrix

Heatmap figures created from the error covariance matrices of the UCSD and MUCSD algorithms are given in this subsection for different scenarios. The UCSD error covariance matrix, $\mathbf{\Sigma}_{m,n}$ is defined in (26), while the MUCSD error covariance matrix, $\mathbf{\Sigma}$ is defined in (35).

5.1.1 UCSD ($\mathbf{\Sigma}_{m,n}$)

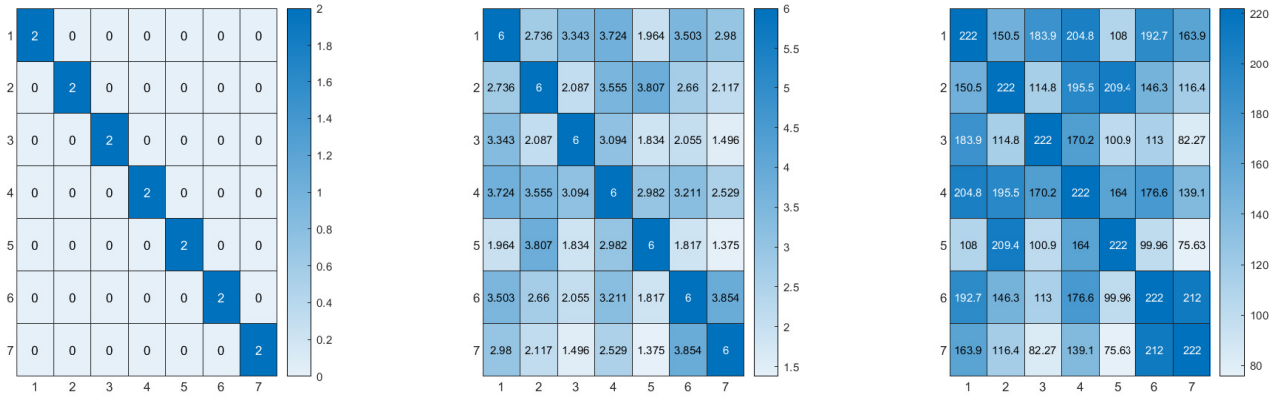

Figure 19: Error covariance matrix derived for the UCSD algorithm, with expression $\mathbf{\Sigma}_{m,n} = 2\sigma^2\mathbf{I}_K + \sigma^2_{\gamma_n}\mathbf{H}\mathbf{H}^\mathsf{T}$, for different values of σ_{γ_n}. From left to right: $\sigma_{\gamma_n} = 0$, $\sigma_{\gamma_n} = 1$ and $\sigma_{\gamma_n} = 10$. We see that the diagonal pattern of $\mathbf{\Sigma}_{m,n}$ declines with the increase of σ_{γ_n} with respect to σ.

In Figure 19, we can see the contribution of the pseudorange code error and the error given by the noisy observations of user n to the covariance matrix $\mathbf{\Sigma}_{m,n}$. From left to right, we have values of $\sigma_{\gamma_n} = 0$, $\sigma_{\gamma_n} = 1$ and $\sigma_{\gamma_n} = 10$ in meters. In the case with $\sigma = 1$ and $\sigma_{\gamma_n} = 0$, we see how $\mathbf{\Sigma}_{m,n}$ is a diagonal matrix, which is useful for computational efficiency when implementing the UCSD algorithm and also can bring simplicity to its mathematical derivation. By increasing the value of σ_{γ_n}, the error covariance matrix loses its diagonal pattern and it becomes more colored with the increase of σ_{γ_n} with respect to σ. It would be interesting to focus on the computational efficiency of the UCSD algorithm implementation and, when doing that, the diagonal or non-diagonal pattern of the $\mathbf{\Sigma}_{m,n}$ matrix for high values of σ_{γ_n} should be taken into account.

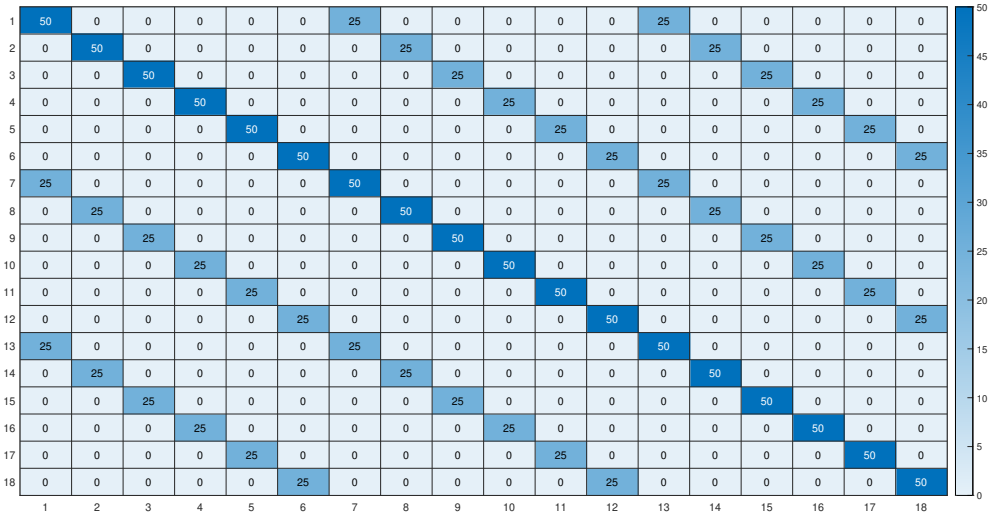

Figure 20: Heatmap of $\mathbf{\Sigma}$ showing the contribution of the pseudorange code error to the MUCSD error covariance matrix for $K = 6$, $N = 3$, $\sigma = 5$ and $\sigma_{\gamma_n} = 0$. $\mathbf{\Sigma}$ is a Toeplitz matrix.

5.1.2 MUCSD ($\mathbf{\Sigma}$)

In this subsection, two figures with the heatmap of matrix $\mathbf{\Sigma}$ are provided for different scenarios. The contribution of the pseudorange code error to the error covariance matrix is shown in Figure 20 for $K = 6$, $N = 3$ and $\sigma = 5$. This contribution is computed as $\sigma^2(\mathbf{J}_{N\times N} \otimes \mathbf{I}_K + \mathbf{I}_{NK})$, or equivalently, calculating the $\mathbf{\Sigma}$ matrix for $\sigma_{\gamma_n} = 0$. For this configuration, it can be seen that $\mathbf{\Sigma}$ is a Toeplitz matrix.

When $\boldsymbol{\eta}_{\boldsymbol{\gamma}_n}$ is also considered, the resulting error covariance matrix $\mathbf{\Sigma}$ is colored as shown in Figure 21 for $\sigma_{\gamma_n} = 10$. For low values of σ with respect to σ_{γ_n}, $\mathbf{\Sigma}$ can be assumed to be a block diagonal matrix, as in the case of Figure 21. However, in cases where σ has high values with respect to σ_{γ_n}, the $\mathbf{\Sigma}$ matrix cannot be assumed to be block diagonal. Instead, $\mathbf{\Sigma}$ would be a block Toeplitz matrix. The fact that the error covariance matrix follows either a block diagonal or a block Toeplitz pattern is beneficial for the computational efficiency of the MUCSD algorithm, given the matrix sparse nature. We consider relevant to highlight that $\mathbf{\Sigma}$ is a symmetric matrix and that consequently its inverse is also symmetric, which means that $\mathbf{\Sigma}^{-1} = (\mathbf{\Sigma}^{-1})^{\mathsf{T}}$. This property might be helpful for simplifying further mathematical analysis and also when implementing the MUCSD algorithm.

5.2 Position Estimation RMSE

Position estimation RMSE values obtained with the UCSD and MUCSD algorithms are provided in this subsection. General details regarding the three conducted experiments are given in Table 6. A final discussion on the interpretation of the results via the inspection of the CRB expression is provided.

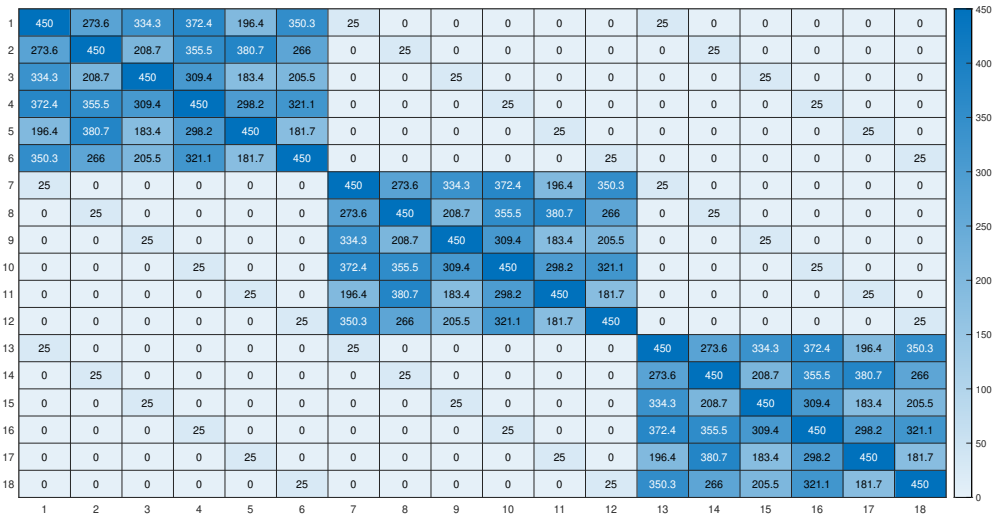

Figure 21: Heatmap of $\mathbf{\Sigma}$ for $K = 6$, $N = 3$, $\sigma = 5$ and $\sigma_{\gamma_n} = 10$, computed as $\mathbf{\Sigma} = \sigma^2(\mathbf{J}_{N\times N} \otimes \mathbf{I}_K + \mathbf{I}_{NK}) + \sigma^2_{\gamma_n}\mathbf{I}_N \otimes \mathbf{H}\mathbf{H}^\mathsf{T}$. For values of σ under 10 meters, $\mathbf{\Sigma}$ is a block diagonal matrix. For high values of σ with respect to σ_{γ_n}, $\mathbf{\Sigma}$ is a block Toeplitz matrix.

5.2.1 Experiment A

In this section, results obtained with the implementation of the UCSD algorithm for different values of σ_{γ_n} are provided in Figure 22. This corresponds to Experiment A from Table 3. The experiment-dependent parameter values set for this experiment are listed in Table 7.

We can see how the results with DGNSS are better, as the ionospheric and tropospheric delays are fully cancelled because we are assuming a short-baseline scenario. With UCSD, these delays are also completely cancelled, but the variances of the code track error of the users are added up, giving worse performance. Even in the case where $\sigma_{\gamma_n} = 0$, the UCSD gives a worse performance than DGNSS. We see how the performance of the UCSD algorithm is inversely proportional to the value of σ_{γ_n}. For $\sigma_{\gamma_n} = 10$, the UCSD is slightly worse than for $\sigma_{\gamma_n} = 0$, mostly for high values of σ. However, when it is assumed that the position of the cooperative receivers is known with 100 meters of variance, which means that the user m is provided with very noisy observations, the performance of the UCSD algorithm is extremely low.

Results shown in Figure 22 are not surprising as in this experiment we are not exploiting the ability of our simulator to include additional users in the scenario, as we do with the MUCSD algorithm. Given the obtained results, and in a natural manner, we can reach the conclusion that this performance would improve extending the model to N cooperative users. This is the reason why the UCSD algorithm is upgraded into its massive version.

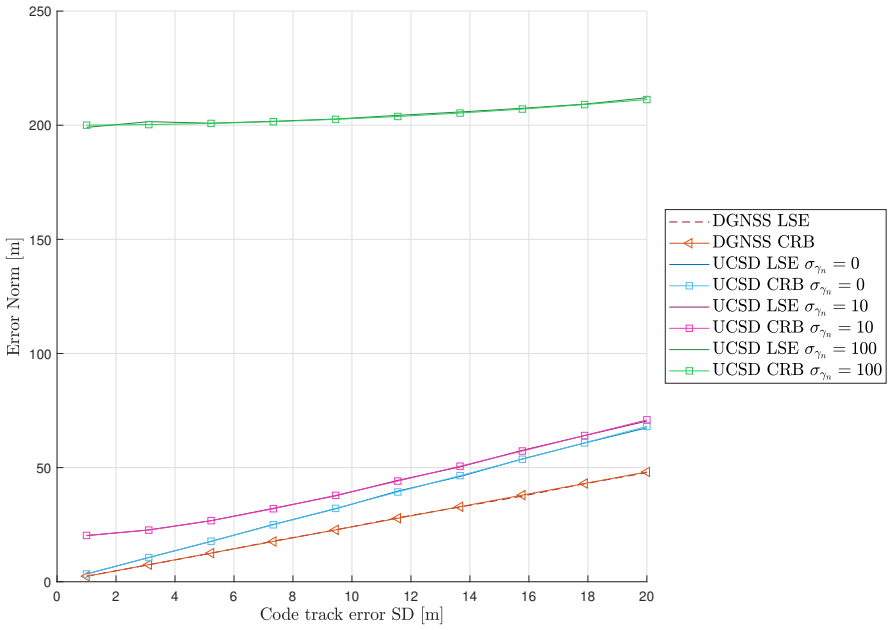

Figure 22: RMSE results obtained with Experiment A. The experiment-dependent parameter values set for this experiment are listed in Table 7. The code track error corresponds to the pseudorange measurement error, with standard deviation σ.

Table 7: Experiment-dependent parameter values set for Experiment A. N corresponds to the number of cooperative users introduced in the experiment, K corresponds to the number of satellites in view given the experiment geometry, M corresponds to the number of Monte Carlo realizations set in the simulator and S corresponds to the amount of σ values tested (exact values are also given in a vector).

Parameter	Value
Algorithm	UCSD
N	0
K	7
M	1e4
S	10
$\boldsymbol{\sigma}$	[1, 3.1, 5.2, 7.3, 9.4, 11.5, 13.6, 15.7, 17.8, 20] (in meters)
$\boldsymbol{\sigma_{\gamma_n}}$	[0, 10, 100] (in meters)

5.2.2 Experiment B

The aim of this experiment is to test whether by expanding the model in UCSD to N users we can get a performance that is comparable to the one of a system using DGNSS. For this, we have conducted Experiments B.1 and B.2.

Results obtained with Experiment B.1 can be found in Figure 23. The experiment-dependent parameter values user in Experiment B.1 can be found in Table 8. With the results in this figure we can state that the MUCSD achieves the goal of increasing the positioning performance by bringing more users into the scenario. We see that for $\sigma_{\gamma_n} = 0$, which is the scenario where we assume that we have perfect knowledge of the coordinates of the cooperative users, we start seeing a performance of the MUCSD for N=2. When N=50, meaning that the number of users introduced in the scenario is considerably high, the MUCSD performance is comparable to the one of a DGNSS system. By inspecting the numerical values, we see that the difference in performance is of around 1 meter maximum. It is assumed that if N is further increased, this meter difference will become closer to zero.

Results obtained with Experiment B.2 can be found in Figure 24. The experiment-dependent parameter values user in Experiment B.2 can be found in Table 10. In this experiment, we are testing the performance of the MUCSD algorithm for a relatively high value of $\sigma_{\gamma_n} = 10$ meters. This means that in this scenario we are assuming that the reference user m is given coordinates from the cooperative receivers that can be wrong up to 10 meters. As we considered that this would be a challenging scenario for our algorithm, we used higher values of N for this experiment.

As seen in Figure 24, when $\sigma_{\gamma_n} = 10$, we are not able obtain a good performance by introducing only one cooperative user to the scenario. However, when increasing the amount of cooperative users to 25 we already see an improvement of the performance. In Figure 24, results for N=25, N=50 and the DGNSS scenario are very close. To properly inspect the results, we have provided numerical values in Table 9. The numerical results in Table 9 suggest that the obtained position errors are higher for the MUCSD algorithm than for DGNSS.

The conclusion of Experiment B.2 is that the MUCSD algorithm is actually working towards providing a performance that is comparable to the one provided by DGNSS. Moreover, it is very clear how the performance of MUCSD is better with N=50 than with N=25. This induces to think that if N had been increased even more, the DGNSS would be met by MUCSD even for low values of σ.

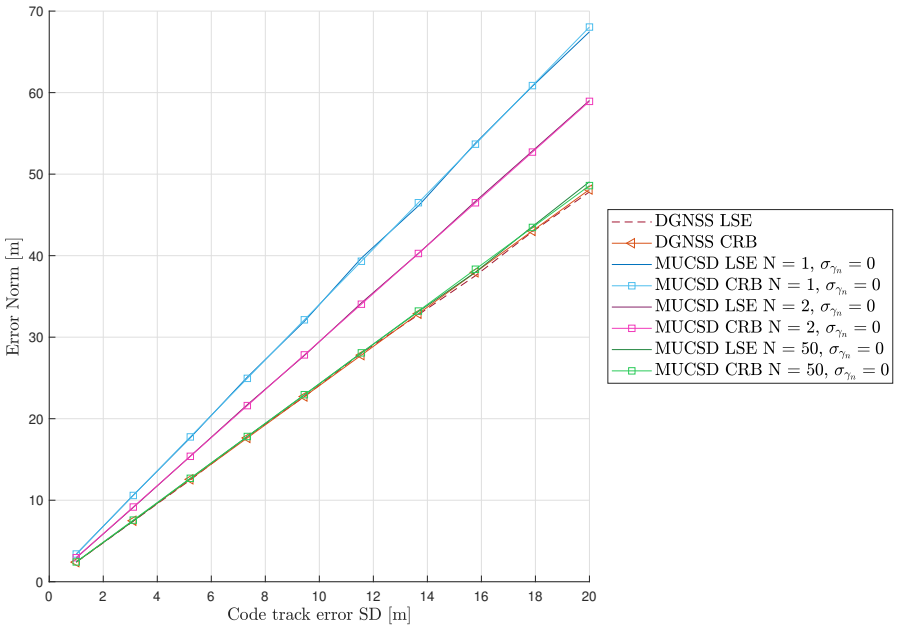

Figure 23: RMSE results obtained with Experiment B.1. The experiment-dependent parameter values set for this experiment are listed in Table 8. The code track error corresponds to the pseudorange measurement error, with standard deviation σ.

Table 8: Experiment-dependent parameter values set for Experiment B.1.

Parameter	Value
Algorithm	MUCSD
N	1, 2, 50
K	7
M	1e4
S, $\boldsymbol{\sigma}$	Same as Experiment A (see Table 7)
$\boldsymbol{\sigma_{\gamma_n}}$	0 (in meters)

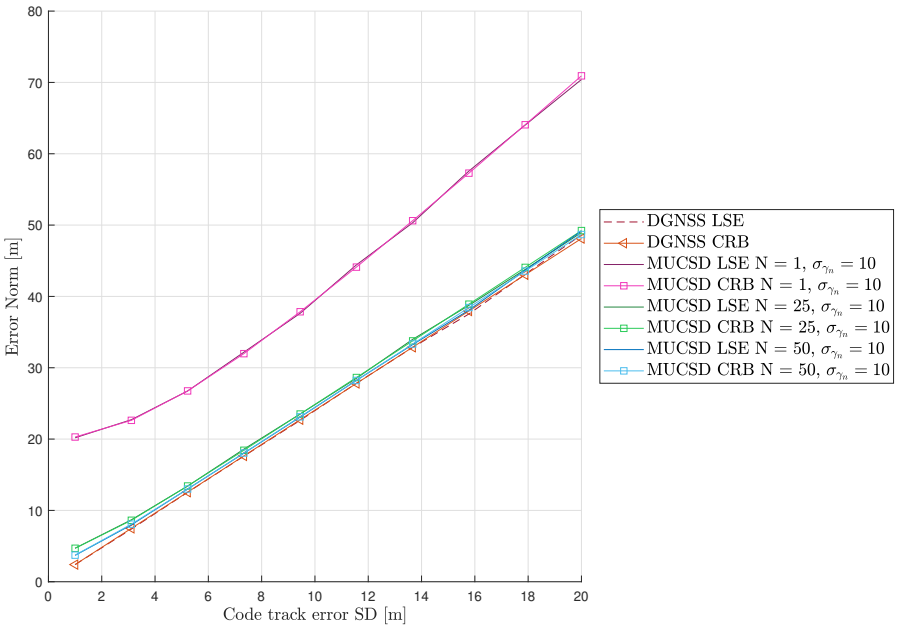

Figure 24: RMSE results obtained with Experiment B.2. The experiment-dependent parameter values set for this experiment are listed in Table 10. As the three configurations in orange, red and blue color give a non-distinguishable performance in this figure, numerical results are provided in Table 9. The code track error corresponds to the pseudorange measurement error, with standard deviation σ.

Table 9: RMSE results obtained with Experiment B.2 for the configurations that give similar performance in Figure 24.

σ	2	4	6	8	10	12	14	16	18
Error for N = 25	6.6	10.6	15.3	20	24.8	29.7	34.6	39.25	44.1
Error for N = 50	5.75	10.1	14.9	19.6	24.4	29.3	34.1	38.8	43.8
DGNSS	4.75	9.6	14.45	19.2	24	28.9	33.6	38.2	43.2

Table 10: Experiment-dependent parameter values set for Experiment B.2.

Parameter	Value
Algorithm	MUCSD
N	1, 25, 50
K	7
M	1e4
S, $\boldsymbol{\sigma}$	Same as Experiment A (see Table 7)
$\boldsymbol{\sigma_{\gamma_n}}$	10 (in meters)

5.2.3 Experiment C

Results of Experiment C can be found in Figure 25, together with the experiment-dependent parameters used in Table 11. With this experiment, we clearly see how the MUCSD cannot achieve the performance of DGNSS when σ_{γ_n} is very high.

We consider that these results make sense and they are not discouraging. This is because, in the proposed scenario, we have $\sigma_{\gamma_n} = 100$ meters, which means that the reference user m is given N positions of cooperative users which can be wrong up to 100 meters. Observations that are corrupted by such a high error are of course not contributing in a positive way to the MUCSD algorithm. However, it is relevant to highlight the improving performance of MUCSD with the increase of N. The results in Figure 25 suggest that for an extremely high value of N, the performance of MUCSD would be comparable to the one of DGNSS.

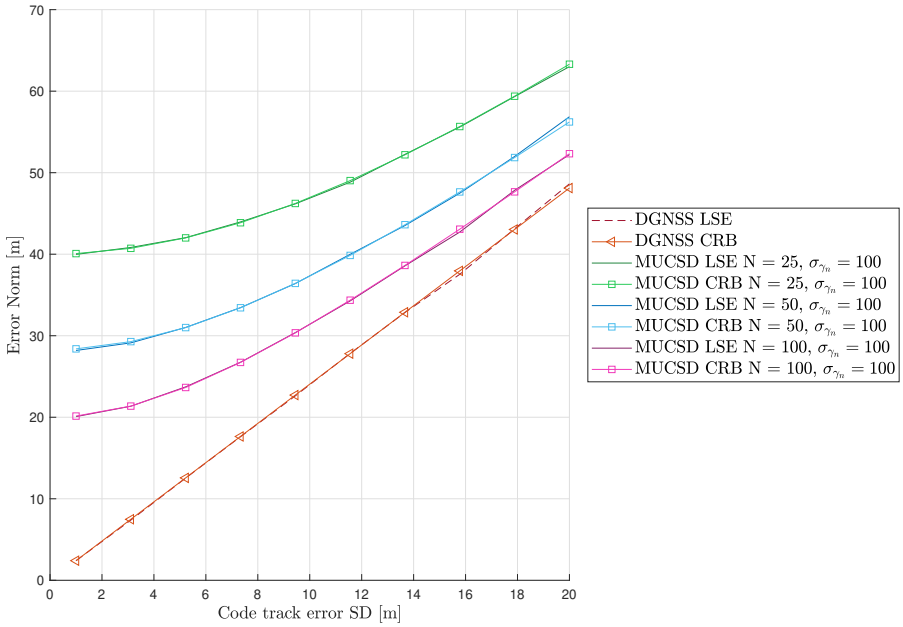

Figure 25: RMSE results obtained with Experiment C. The experiment-dependent parameter values set for this experiment are listed in Table 11. The code track error corresponds to the pseudorange measurement error, with standard deviation σ.

Table 11: Experiment-dependent parameter values set for Experiment C.

Parameter	Value
Algorithm	MUCSD
N	25, 50, 100
K	7
M	1e4
S, $\boldsymbol{\sigma}$	Same as Experiment A (see Table 7)
$\boldsymbol{\sigma}_{\boldsymbol{\gamma}_n}$	100 (in meters)

5.2.4 Interpretation of results via CRB inspection

The difference between DGNSS and MUCSD becomes lower with the increase of the code error standard deviation σ. This can be understood by inspecting the CRB expression in (41), which is proportional to the error covariance matrix $\mathbf{\Sigma}$ defined in (35).

On the one hand, when the value of σ is high, the term $\sigma^2(\mathbf{J}_{N\times N} \otimes \mathbf{I}_K + \mathbf{I}_{NK})$ dominates the expression, meaning that the pseudorange inaccuracy corresponds to the main error source of the model. In this case, $\mathbf{\Sigma}$ is a block diagonal matrix. Consequently, the performance of DGNSS becomes more similar to the one of the MUCSD algorithm. This can be understood by acknowledging that DGNSS includes the contribution of the pseudorange errors but not the one of the noisy observations from the cooperative users.

On the other hand, when the value of σ is lower in comparison to the value assigned to $\sigma^2_{\gamma_n}$, the term $\sigma^2_{\gamma_n}\mathbf{I}_N \otimes \mathbf{H}\mathbf{H}^\mathsf{T}$ becomes dominant in the error covariance matrix. As a consequence, the cooperative user position error has a strong impact on MUCSD performance and it differs more from the one provided by DGNSS. This happens when $\mathbf{\Sigma}$ is a block Toeplitz matrix.

It is very beautiful to understand how the CRB expression, which contains the error covariance matrix of the proposed model, can explain the relationship between the widely used DGNSS technique and the algorithm we are proposing in this Master Thesis.

6 Conclusion

In the first subsection of the conclusion, we review the previously described results in order to list which are the most important findings. A summary of the findings of the research can be found in Table 12. Finally, some motivating future lines of research for this thesis are proposed by the authors.

6.1 Final Results

The main conclusions extracted from the experiment results that have been previously described can be found in Table 12. With Experiment A, we have seen that the UCSD algorithmn is not able to give a performance comparable to DGNSS. Although the ionospheric and tropospheric errors are equally cancelled for short-baseline scenarios, the variance of the code error of the users is added up. Consequently the LSE estimation error increases.

With Experiment B, we have seen that the MUCSD algorithm is able to provide a performance comparable to DGNSS when the number of cooperative receivers in the scenario is increased. The amount of additional users required in the setup depends on how noisy the observations provided by these users are. In Experiment B.1, we assumed that the observations were not noisy at all. This is the reason why we see such an increase of performance with N=2 with respect to N=1 (almost 10 meters less of error for high values of $\sigma = 18$ m). Furthermore, we see with the MassiveCoop-Sim simulator that when $N = 10$, the MUCSD algorithm is able to counteract the addition of the variances of cooperative users and the performance is comparable to DGNSS. When the observations of the cooperative users are assumed to be noisy, with Experiment B.2, we still get a good performance for high values of N.

Finally, with Experiment C, we have seen that for very high values of $\boldsymbol{\gamma_n}$ our algorithm is not able to provide DGNSS performance even with 50 users having been introduced into the scenario. However, we consider that this makes sense, as it would not make sense to implement a cooperative positioning algorithm where we receive observations that might be up to 100 meters wrong.

In conclusion, we have proven the hypothesis stated in Section 2.4, as we have provided a novel algorithm for Massive Single Differentiation that achieves DGNSS performance without the need of having a precise knowledge of the coordinates of other receivers. Instead, we can work with noisy observations of these receivers, always maintaining the noise under a threshold that will depend on the power of the scenario (proportional to the value of N). The higher the value of N, the higher this noise can be so that the MUCSD algorithm performs as a system with DGNSS.

6.2 Future Research

In this section, we have described the lines of research that are to be explored in the future. We first propose to improve the performance of the UCSD and MUCSD algorithms in terms of estimation error by including the use of carrier-phase measurements and Double Differentition (DD), as the well-known Real Time Kinematic (RTK) algorithm does.

Table 12: Main conclusions extracted from the experiments conducted to test the performance of the UCSD and MUCSD algorithms, which are listed in Table 3

ID	Conclusion
A	The UCSD algorithm does not provide DGNSS performance.
B	The MUCSD algorithm provides DGNSS performance for realistic values of $\boldsymbol{\gamma_n}$.
C	The MUCSD algorithm does not provide DGNSS performance for unrealistically high values of $\boldsymbol{\gamma_n}$.

Afterwords, we propose several lines of research which are mostly related to implementing the UCSD and MUCSD algorithms in a system that allows us to test its performance in a realistic scenario. These lines of research would be a computational efficiency evaluation, the centralization of the proposed system into the cloud and the exploration of communication protocols to be used in our system, as feedback between users is needed in a massive cooperative positioning algorithm.

6.2.1 Use of Carrier Phase Measurements and Double Differentiation)

Traditional SPP algorithms use only GNSS pseudorange code measurements of the receiver and give positioning accuracy at meter level. The same happens with systems implementing DGNSS, as although most of the error sources are cancelled, the performance is still at the level of meters. It is necessary to use carrier-phase measurements in order to achieve high accuracy positioning results. The Precise Point Positioning (PPP) and RTK techniques make use of both pseudorange and carrier-phase measurements, leading to very accurate cm-level performances. Consequently, we would like to continue this research by proposing an algorithm that also makes use of carrier-phase measurements. With RTK, the key procedure is the Double Differentiation (DD) of GNSS mesurements between receivers and satellites. When following this line of research, we would like to compare the estimation error obtained with our technique to the one provided by the RTK algorithm.

DD is, from a computational perspective, more demanding than SD. However, it would be very interesting for the authors to include this analysis, from a theoretical and mathematical point of view.

6.2.2 Computational Efficiency Evaluation

Although having listed the computational resources used in order to run the .m scripts of the MassiveCoop-Sim simulator, we have not conducted a study on the computational efficiency of the implementation of the UCSD and MUCSD algorithms. This would be essential when wanting to bring these algorithms into a real scenario in order to test it in a realistic environment. When doing this, we should also consider the implications of implementing these algorithms on Real-Time Systems (RTS). With RTS, the processing of data and occurring events implies critical time constraints that have not been taken into account in this document.

6.2.3 Centralization of the System

With the proposed UCSD and MUCSD algorithms, all the computational load is being taken by the reference m. Considering the high dimensional data processed by the algorithms, mostly when a large number of users N are introduced into the scenario, it would be of our interest to consider the possibility of centralizing the system. For example, an interesting line of research would be to describe a realistic scenario where the MassiveCoop-Sim simulator operates in a cloud.

6.2.4 Selection of a Communication Protocol

As it has been introduced in Section 1.4, in cooperative positioning techniques, the presence of communication between devices is key. This happens mostly in systems where different technologies are used (what we call, hybrid systems). However, as we have seen, in only-GNSS systems such as the ones using DGNSS, there is the need of communication between terminals. In the case of DGNSS, as it has been explained beforehand, DGNSS corrections are sent from the referene or base station to the user. It is therefore necessary to define which communication protocol is used when sending, in this case, the DGNSS corrections.

Having understood the need of a communicatios protocol in a cooperative system, and after comprehending the core of the proposed UCSD and MUCSD algorithms, we can state that our system requires a protocol so that the N users in the scenario can send their measurements to the reference user m, to the cloud, or even between them. In some literature presented in Section 1.4, Dedicated Short-Range Communications (DSRC) are mentioned, as users are assumed to be at short distances from each other [25]. This would actually be the case of a scenario like the one proposed in Figure 10 when the baseline distances are short. In [25], they state that with the assistance of DSRC, the implementation of CP algorithms does not necessarily rely on any specialized sensors or infrastructures, making it an inexpensive and practical solution. The interested reader may refer to Section 1.4.4 to read more about this matter.

References

[1] Md Anowar Hossain, Ibrahim Elshafiey, and Abdulhameed Al-Sanie. Cooperative vehicle positioning with multi-sensor data fusion and vehicular communications. *Wireless Networks*, 25, 04 2019.

[2] NEU. Northeastern University's Discovery Cluster - Research Computing. `https://rc-docs.northeastern.edu/en/latest/welcome/welcome.html`, 2019.

[3] P.D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. GNSS/GPS. Artech House, 2013.

[4] Fuxi Wen, Henk Wymeersch, Bile Peng, Wee Peng Tay, Hing Cheung So, and Diange Yang. A survey on 5G Massive MIMO localization. *Digital Signal Processing*, 94:21–28, 2019. Special Issue on Source Localization in Massive MIMO.

[5] Nadezhda Chukhno, Sergio Trilles, Joaquín Torres-Sospedra, Antonio Iera, and Giuseppe Araniti. D2D-Based Cooperative Positioning Paradigm for Future Wireless Systems: A Survey. *IEEE Sensors Journal*, 22(6):5101–5112, 2022.

[6] Zohair Abu Shaban, Gonzalo Seco-Granados, Craig Benson, and Henk Wymeersch. Performance Analysis for Autonomous Vehicle 5G-Assisted Positioning in GNSS-Challenged Environments, 04 2020.

[7] Yuan Zhuang, You Li, Longning Qi, Haiyu Lan, Jun Yang, and Naser El-Sheimy. A Two-Filter Integration of MEMS Sensors and WiFi Fingerprinting for Indoor Positioning. *IEEE Sensors Journal*, 16(13):5125–5126, 2016.

[8] Muhammad Khan, Nasir Saeed, Arbab Ahmad, and Chankil Lee. Location Awareness in 5G Networks Using RSS Measurements for Public Safety Applications. *IEEE Access*, PP:1–1, 09 2017.

[9] Pedro Fernandez, José Santa, and Antonio Skarmeta. Hybrid Positioning for Smart Spaces: Proposal and Evaluation. *Applied Sciences*, 10:4083, 06 2020.

[10] Kian Meng Tan and Choi Look Law. GPS and UWB Integration for indoor positioning. In *2007 6th International Conference on Information, Communications and Signal Processing*, pages 1–5, 2007.

[11] Kun Zhang, Chong Shen, Qun Zhou, Haifeng Wang, Qian Gao, and Yushan Chen. A combined GPS UWB and MARG locationing algorithm for indoor and outdoor mixed scenario. *Cluster Computing*, 22, 05 2019.

[12] Kamil Maciuk. GPS-only, glonass-only and combined GPS+GLONASS absolute positioning under different sky view conditions. *Tehnicki Vjesnik*, 25:933–939, 06 2018.

[13] Kamijo Shunsuke, Gu Yanlei, and Li-Ta Hsu. GNSS/INS/On-board Camera Integration for Vehicle Self-Localization in Urban Canyon. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2533–2538, 2015.

[14] Raghu K. Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.

[15] Virginia Pilloni. How Data Will Transform Industrial Processes: Crowdsensing, Crowdsourcing and Big Data as Pillars of Industry 4.0. *Future Internet*, 10:24, 03 2018.

[16] Jinwei Liu, Haiying Shen, and Xiang Zhang. A Survey of Mobile Crowdsensing Techniques: A Critical Component for the Internet of Things. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6, 2016.

[17] Mengyao Peng, Hui Lin, and Xiaoding Wang. A Reinforcement Learning-based Task Classification Mechanism for Privacy-Enhanced Mobile Crowdsensing Strategy. In *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, pages 100–107, 2021.

[18] Tao Peng, Jierong Liu, Jianer Chen, and Guojun Wang. A Privacy-Preserving Crowdsensing System with Muti-blockchain. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1944–1949, 2020.

[19] Ping Zhang, Jian Lu, Yan Wang, and Qiao Wang. Cooperative localization in 5G networks: A survey. *ICT Express*, 3(1):27–32, 2017.

[20] G.M. Hoang, B. Denis, J. Härri, and D. T.M. Slock. Mitigating unbalanced GDoP effects in range-based vehicular Cooperative Localization. In *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 659–664, 2017.

[21] Bin Huang, Zheng Yao, Xiaowei Cui, and Mingquan Lu. Dilution of Precision Analysis for GNSS Collaborative Positioning. *IEEE Transactions on Vehicular Technology*, 65(5):3401–3415, 2016.

[22] Federico Penna, Mauricio A. Caceres, and Henk Wymeersch. Cramér-Rao Bound for Hybrid GNSS-Terrestrial Cooperative Positioning. *IEEE Communications Letters*, 14(11):1005–1007, 2010.

[23] Deyue Zou, Weixiao Meng, Shuai Han, Kai He, and Zhongzhao Zhang. Toward Ubiquitous LBS: Multi-Radio Localization and Seamless Positioning. *IEEE Wireless Communications*, 23(6):107–113, 2016.

[24] Daiqin Yang, Fang Zhao, Kai Liu, Hock Beng Lim, Emilio Frazzoli, and Daniela Rus. A GPS Pseudorange Based Cooperative Vehicular Distance Measurement Technique. In *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2012.

[25] Kai Liu, Hock Beng Lim, Emilio Frazzoli, Houling Ji, and Victor C. S. Lee. Improving Positioning Accuracy Using GPS Pseudorange Measurements for Cooperative Vehicular Localization. *IEEE Transactions on Vehicular Technology*, 63(6):2544–2556, 2014.

[26] Dorota Grejner-Brzezinska, Charles Toth, Terry Moore, John Raquet, Mikel Miller, and Allison Kealy. Multisensor Navigation Systems: A Remedy for GNSS Vulnerabilities. *Proceedings of the IEEE*, 104:1–15, 03 2016.

[27] Khurram Mazher, Muhammad Tahir, and Khurram Ali. GNSS pseudorange smoothing: Linear vs non-linear filtering paradigm. In *2016 IEEE Aerospace Conference*, pages 1–10, 2016.

[28] Eric Richter, Marcus Obst, Robin Schubert, and Gerd Wanielik. Cooperative relative localization using Vehicle-To-Vehicle communications. In *2009 12th International Conference on Information Fusion*, pages 126–131, 2009.

[29] Fabian de Ponte Müller, Alexander Steingass, and Thomas Strang. Zero-Baseline Measurements for Relative Positioning in Vehicular Environments. 12 2013.

[30] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2011.

[31] Keyvan Ansari, Charles Wang, Lei Wang, and Yanming Feng. Vehicle-to-Vehicle Real-Time Relative Positioning Using 5.9 GHz DSRC Media. In *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, pages 1–7, 2013.

[32] Michael Gabor and R. Nerem. Satellite-Satellite Single-Difference Phase Bias Calibration As Applied to Ambiguity Resolution. *Navigation*, 49:223–242, 12 2002.

[33] Gonzalo Seco-Granados, José López-Salcedo, David Jiménez-Baños, and Gustavo López-Risueño. Challenges in Indoor Global Navigation Satellite Systems: Unveiling its core features in signal processing. *IEEE Signal Processing Magazine*, 29(2):108–131, 2012.

[34] J.M. Juan Zornoza J. Sanz Subirana and M. Hernández-Pajares. Navipedia Article: GNSS Signal. `https://gssc.esa.int/navipedia/index.php/GNSS_signal`, 2011.

[35] Manuel Pajares, Jose Juan, Jaume Sanz, and Oscar Colombo. Application of ionospheric tomography to real-time GPS carrier-phase ambiguities Resolution, at scales of 400–1000 km and with high geomagnetic activity. *Geophysical Research Letters - GEOPHYS RES LETT*, 27:2009–2012, 07 2000.

[36] Alexander E. Popugaev and Rainer Wansch. *Multi-Band GNSS Antenna*, pages 69–75. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[37] Lev Rapoport Alfred Leick and Dmitry Tatarnikov. *GPS Satellite Surveying, Fourth Edition, Geodesy Chapter*. John Wiley and Sons, Ltd, 2015.

[38] E. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications*. Artech House mobile communications series. Artech House, 2005.

[39] J.M. Juan Zornoza J. Sanz Subirana and M. Hernández-Pajares. Navipedia Article: Combination of GNSS Measurements. `https://gssc.esa.int/navipedia/index.php/Combination_of_GNSS_Measurements`, 2011.

[40] Daniele Borio, Letizia Lo Presti, Mark Petovello, Gérard Lachapelle, and Dennis Odijk. What does geometry-based and geometry-free mean in the context of GNSS? *Inside GNSS*, pages 22–27, 3 2008.

[41] J.M. Juan Zornoza J. Sanz Subirana and M. Hernández-Pajares. *GNSS Data Processing Volume I: Fundamentals and Algorithms*. European Space Agency (ESA Communications), Noordwijk, the Netherlands, 2013.

[42] Oguz Isik, Ju-Hyeon Hong, Ivan Petrunin, and Antonios Tsourdos. Integrity Analysis for GPS-Based Navigation of UAVs in Urban Environment. *Robotics*, 9:66, 08 2020.

[43] S. Jyothirmaye, V.Satya Srinivas, and B. Ramu. Fast Satellite Selection Techniques and DOPs for Multi-GNSS Positioning. In *2019 URSI Asia-Pacific Radio Science Conference (AP-RASC)*, pages 1–4, 2019.

[44] Haifu Ji, Lihua Ma, Guoxiang Ai, and Meng Wang. *The Distributions of HDOP and VDOP in GNSS and a Corresponding New Algorithm of Fast Selecting Satellites*, volume 160, pages 411–421. 01 2012.

[45] Hongpyo Kim, Jinhyeok Jang, Jun-Pyo Park, Gyu-In Jee, and Young Lee. Satellite Selection Method According to Signal Levels of Multi-Constellation GNSS. pages 3746–3752, 10 2018.

[46] Rodrigo Leandro and Marcelo Santos. An Empirical Approach for the Estimation of GPS Covariance Matrix of Observations. 01 2005.

[47] Emanuela Falletti, Marco Pini, Letizia Lo Presti, and Davide Margaria. Assessment on low complexity C/No estimators based on M-PSK signal model for GNSS receivers. *2008 IEEE/ION Position, Location and Navigation Symposium*, pages 167–172, 2008.

[48] *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*. Pearson Education, 1997.

[49] MATLAB. *Version number 46, 9.11.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2021.