POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Geoinformatica
Scuola di Ingegneria Civile, Ambientale e Territoriale

INTEGRATION OF GNSS AND 5G FOR PRECISE URBAN POSITIONING

Relatore:
Prof. Ludovico Giorgio Aldo Biagi

Tesi di Laurea di:
Marianna Alghisi 946826

Anno Accademico 2020-2021

Contents

Abstract

Positioning services for people and devices has become a necessity for nowadays society: the computation of precise position, both in single epoch and in real time, is in fact a fundamental aspect of many applications, such as guidance and control for people and transport, industrial applications, mass market, public security, assistive technologies and recreational purposes. From about 30 years outdoor positioning has been provided by the Global Navigation Satellite System (GNSS).

At the beginning, positioning was based only on GPS constellation, using expensive and bulky dedicated receivers. Thanks to the technological development and the growing interest in the sector, over the years, the number of constellations has increased, constituting the GNSS. In addition, also the availability of receivers has increased, diversifying quality and pricing: for instance, nowadays, smartphones are equipped with cheap and miniaturized GNSS devices.

Nowadays, thanks to the satellite coverage, we can consider open sky positioning an achieved goal worldwide. However, in urban environment, the obstructions caused by buildings degrade the positioning accuracy, even making it impossible in some configurations.

This thesis aims to be the beginning of a complete analysis of the possible integration of the 5G signals to GNSS positioning in urban environment. The work is structured as follows and can be divided into two main phases:

1 Implementation of a software to perform GNSS point positioning.
  According to the available literature, we developed an open source Python library that is able to perform point positioning, starting from the computation of the ephemeris (position, velocity and clock offset of each satellite in each epoch) to the estimate of the position of the receiver by following the standard formulas and Least Squares ap-

proach. The software aims to be a complete solution that allows the user to perform point positioning starting from raw data, offering all the necessary functions for reading files, computing ephemerides and estimating unknowns. The software implementation, although it is not a research topic, is necessary for the future development of the thesis and to provide a complete open source software available to everyone.

2 Geometric analysis of the advantages obtainable by integrating the 5G signals with GNSS in urban scenario, under the hypothesis of having Time of Arrival (ToA) observations from 5G.
This second phase of work consists in an experimental study of the actual visibility of the GNSS satellites in urban environment: the analysis is carried out through the simulation of 4 scenarios of urban environments with buildings of different heights. The height of the buildings is decisive to differentiate the two classes: scenarios with tall buildings (24 meters high), which we will refer to as urban canyons, and scenarios with residential buildings (9 meters high), which will represent residential streets. Subsequently,the visibility of the satellites is analyzed for each scenario: signals that are not blocked by the presence of buildings are identified with a masking process. At the end, it is shown how the geometry and the number of satellites in view influence the positioning performance and it is studied how the introduction of 5G network for positioning can improve and ensure the performance of positioning in urban environment. This geometrical analysis is performed on the results obtained from the computation of Diluition of Precision (DOP) indices on both GNSS-only and GNSS+5G configurations.

0.1 Structure of the thesis

The content of the document is structured and divided as follows.

The first chapter provides an introduction to the different technologies involved in the exploitation of the work: the various GNSS constellations and 5G.

The second chapter focuses on software development: the entire section is dedicated to the description of the software for positioning. The formulas implemented for determining the position, speed and clock offset of the

satellites and the procedure for estimating the position of the receiver are analyzed in detail. An analysis of the software architecture is also provided.

With the third chapter we enter in the simulation phase of this work: the urban environment simulation scenarios are described in their characteristics and simulation methods, the satellites masking method is introduced (identifies, among all the satellites in sight, those actually visible whose signal is not blocked by the presence of buildings), the results of the satellites visibility analysis for each scenario are proposed and discussed.

The fourth chapter describes the evaluation method applied to assess the geometric quality of the satellites in sight: the analysis is performed by introducing simplified PDOP and HDOP indices. The chapter introduces and analyzes the concept of hybrid positioning GNSS+5G and discusses in detail the results obtained for each scenario by evaluating the improvements introduced by hybridizing GNSS with 5G.

The conclusions summarize what has been discusses and highlight the limits and the future development directions.

Chapter 1

Introduction to GNSS and 5G

Positioning service for people and devices has become a necessity for nowadays society, the detection of the exact position, both in single epoch and in real time, is in fact a fundamental aspect of many services of daily use. This service is made available by the Global Navigation Satellite System (GNSS), which includes all the constellations of satellites, whose signal is used by the receivers to estimate their position.
Consequently, it is evident that the performance of the service is closely linked to the visibility of the satellites, which is strongly compromised in an urban environment due to the presence of buildings and infrastructures.

The following chapter aims to give an overview of the technologies involved in this work: GNSS and the involved constellations and 5G technology.

1.1 Global Navigation Satellite System

With GNSS we refer to Global Navigation Satellite System, which is a terrestrial, sea or air navigation system, that uses a network of artificial satellites to determine the position of a receiver. All the satellites belongs to four different constellations, administered by different authorities:

- GPS
  Global Positioning System managed by USA government, globally fully operating since 1991.
- GLONASS
  Russian global Positioning System, operated by the Russian Space

Forces (VTK), fully operating since 2011.

- GALILEO
  European constellation, not all the satellites are operative.

- BeiDou
  Chinese navigation satellite system.

1.1.1 GPS

GPS stands for Global Positioning System, the project dates back to 1960s, but the first development was in 1970s at the US Department of Defense. In 1978 the first satellite was launched, from 1993 the Initial Operational Capability IOC) phase started: the system was operational H24 but in experimental phase. In 1995 the FOC (Full Operational Capability) of the constellation was officially declared.

According to [2](Alfred Leick Ph.D et Al., 2015) and GPS.gov website, the satellites of the GPS constellation fly in medium Earth orbit (MEO) at an altitude of approximately 20,200 km. The constellation counts 30 operational satellites, divided in 6 different orbital planes, belonging to different blocks that were released in different times and summarized in table 1.1, note that the first block of satellites is not included as it is no longer in orbit.

LEGACY SATELLITES		MODERNIZED SATELLITES		
Block IIA	Block IIR	Block IIR-M	Block IIF	Block III/IIIF
0 operative	7 operative	7 operative	12 operative	4 operative
Launched	Launched	Launched	Launched	First launch
in 1990-1997	in 1997-2004	in 2005-2009	in 2010-2016	in 2018

Table 1.1: Present and Future generations of GPS satellites.

All the satellites of the constellation compose the so-called Space Segment. However, GPS System structure is composed of a three module architecture: Space Segment (SS), Control Segment (CS) and the User Segment (US). CS consists of a global network of ground facilities that track the GPS satellites, monitor their transmissions, perform analyses, and send commands and data to the constellation. US is composed by million of receiver that performs positioning and timing function.

GPS signals include ranging signals and navigation messages. Regarding

navigation messages, they include all the parameters required for the computation of the satellite's position and velocity and the clock offset with respect to GPS time and they also contain the almanac, that gives status information of the constellation and the details regarding the ionospheric model.
As concerns ranging signals, they are used for the estimate of the time distance between the satellite and the receiver.

GPS satellites have an on-board oscillator that produces three sinusoidal carrier phases and five binary codes, summarized in table 1.2.

Carrier Phases	Binary Codes
L1	C/A - Coarse Acquisition
L2	PR P - Precision Code transformed in
L5 (from Block IIF)	Y - encrYpted in 1994
	Navigation Data
	PR L2C (from Block IIR)
	PR M (from Block IIR)

Table 1.2: GPS signals.

GPS system offers two fundamental services, according to [11](GMV, 2011):

- Standard Positioning Service - SPS
  Consists in the standard positioning and timing service provided on GPS L1, L2 and L5 frequencies and available to all GPS users. The L1 frequency contains a coarse acquisition (C/A) code and a navigation data message. The L2 frequency contains a CM-code and CL-code signals whereas I5-code and Q5-code signals are transmitted in L5 frequency.

- Precise Positioning Service - PPS
  Highly accurate military positioning, velocity and timing service broadcasted at the GPS L1 and L2 frequencies. Both frequencies contain a precision (P) code ranging signal with a navigation data message that is reserved for authorized use by the use of cryptography.

For what concerns GPS precision in positioning, it strongly depends on the operative modalities applied, however, it can be safely said that the performances are slightly below 10m for SPS and below 5 meters for PPS.

1.1.2 GLONASS

The acronym GLONASS stands for GLObal Navigation Sputnik System, and is the Russian global navigation satellite system, administered by the Russian Space Forces. The development of the project started during the Cold War in the early 60's, in opposition to the American GPS.

The first satellites were launched in 1982 and activated in 1983. The entire system was made fully operational in 1995 with the completion of the constellation, following some postponements. The innovation and maintenance of the system was severely compromised by the bad Russian economical situation of the period, in fact in 2002 the whole constellation consisted in only 7 operational satellites, consequently the navigation service was practically unavailable. Starting from 2001 the restoration of the system became a government priority and then, after completing the federal modernization program, the system became again fully operational in 2011.

According to Glonass IAC (Information Analytical Center), the constellation comprehend a total of 31 satellites, of which 24 are fully operational and 4 are for emergency purposes. The 24 operational satellites are equally distributed in three orbital planes, roughly circular, with an orbital altitude of about 19100 kilometers.

A summary of the status and releases of the different blocks of satellites making up the constellation is provided in the table 1.3.

GLONASS	GONASS-M	GLONASS-K	GLONASS-K2
1982-2005	2003-2016	2011-2018	2017+
Decommissioned	In use	In-orbit validation	In development

Table 1.3: GLONASS satellites generations.

For what concerns the signal transmitted by GLONASS satellites, according to [10](GMV, 2011), there two types of signals: open standard-precision signals L1OF/L2OF, and obfuscated high-precision signal L1SF/L2SF. Codes, as in GPS, are obtained with an encoding process applied on the binary phase-shift keying applied to the signals.

GLONASS provides two different services:

- SPS - Standard Positioning Service (or Standard Accuracy Signal service)

It is an open service for civilian purposes, free of charge for worldwide users. The navigation signal is provided in frequency band G1 and, starting from GLONASS-M, it was added a second civilian code G2.

- PPS - Precise Positioning Service (or High Accuracy Signal service) Restricted to military and authorized users.

Like for GPS, Glonass is composed by three segments: space (SS, discussed above), ground (GS) and user segment (US). GS is composed of a System Control Center and a network of Command and Tracking Stations located in Russian territories. US consists in all the users receivers able to compute the positions, velocity and time of the Glonass satellites with the corresponding navigation signals.

In conclusion, it is important to remind that until 2006 the precision in the estimate of the position of the receiver for civilian purposes was about 30 meters, while for military purposes it was around 10 meters. Starting from 2007 the whole system was made available for everyone. Nowadays, Glonass performance has improved and reached the GPS performance.

1.1.3 GALILEO

Galileo is the European GNSS and provides improved positioning and timing services. Realized on the footsteps of GPS, it offers a reliable alternative to Glonass and GPS itself because it remains under civilian control. The program is designed to be compatible with all existing and planned GNSS and inter-operable with GPS and Glonass.

Galileo dates back its initial services to December 2015 and it is not yet fully operational. The Galileo program was structured according to the conventional three main phases: In-Orbit Validation (IOV), Initial Operational Capability (IOC) and Full Operational Capability (FOC) phases. The constellation actually counts 22 in-orbit satellites, 18 of them in FOC and 4 in IOV.
When Galileo will be fully operational, there will be 30 satellites in Medium Earth Orbit (MEO) at an altitude of 23222 kilometres, divided in three orbital planes. The satellites will cover all the Earth surface and each orbit will take about 14 hours.

The satellites transmits three signals: E1, E6 and E5, the last one is further sub-divided in signals E5a and E5b. For a deeper analysis on GALILEO

signals we remind to [16] (J.A Avila Rodriguez, 2011).

GALILEO system expands the range of services intended for users, reporting from [8] (GMV, 2011):

- Open Service - OS
  Positioning service accurate to one metre, open to every user typology. In particular it targets the mass market and is intended for motor vehicle navigation and location-based mobile telephone services.

- High Accuracy Service - HAS
  Service complementing the OS by providing an additional navigation signal and added value services in a different frequency band. The HAS signal can be encrypted in order to control the access to the Galileo HAS services.

- Public Regulated Service - PRS
  Service restricted to government-authorised users, for sensitive applications which require a high level of service continuity. It is intended for security and strategic infrastructure.

- Search and Rescue Service - SAR
  Galileo's worldwide search and rescue service will help to forward distress signals to a rescue coordination centre by detecting emergency signals transmitted by beacons and relaying messages to them.

The system architecture follows the setup already seen for GPS and Glonass: three segments for Space, Ground and User.

In conclusion, once the system will be fully operative, the reachable performance will be different for each service. For a further analysis we remind to [9](GMV, 2011).

1.1.4 BeiDou

BeiDou Navigation Satellite System (BDS) is the new Chinese global positioning solution, which became fully operational in June 2020. We refer to it as a 'new system' and it is composed by three different systems: BeiDou-1 (or Compass), BeiDou-2 and BeiDou-3.
BeiDou-1 constellation, now completely abandoned, was born for safety reasons and consisted of 4 geostationary satellites capable of serving positioning only within the Chinese territory, in order to offer itself as an independent

alternative to GPS for military and emergency purposes.
BeiDou-2 consists of the second generation of satellites with a total of 15 vehicles in geosynchronous orbit, i.e. at about 36,000 kilometers in height, and only 3 in MEO at 21500 kilometers in height. With this second generation satellite system, coverage is extended to most of the territories of Asia and Pacific islands. Use cases expanded to industrial technologies for final users, for instance cars and smartphones sold in China regularly use the BeiDou system for positioning.

As seen for GPS, Glonass and Galileo; BeiDou final Architecture is composed of Space Segment, Ground Segment and User Segment. Regarding the final configuration of the Space Segment reached with BeiDou-3, it consists in an hybrid constellation of 35 operational satellites (5 of them are currently retired for testing) that cover the whole Earth, 3 of them are in geostationary orbit (GEO), 3 in inclined geosynchronous orbit (IGSO) and 24 in medium Earth orbit (MEO). All the space vehicles can communicate with each other and can operate without ground station support.

the entire constellation is composed, according to available information dating back to July 2020, of a total of 42 satellites belonging to the third and second blocks: 30 of BeiDou-3 and 12 of BeiDou-2.

Regarding the Ground Segment, it consists in several ground stations including master control stations, time synchronization stations and monitoring stations.

For what concerns the User Segment, it consists in various kinds of receivers, such as terminals, chips, antennae, application system and services.

BeiDou-3 provides 5 public signals: B1I, B1C, B2a, B2b and B3I in 3 different frequency bands of B1, B2 and B3.

BeiDou offers global and regional services. Regarding the global services we have:

- Open Service (OS)
  As seen for GPS and GALILEO, also BeiDou offer to the user free navigation solutions for civilian purposes, with an accuracy of 10 meters.

- Authorized Service (AS)

There is little information for this type of service, but it offers high reliable use even in complex situations

Regarding the regional services:

- Wide Area Augmentation service
  Aim to achieve the accuracy of one meter by including 30 stations that broadcast corrections to specific receivers at regional scales.

- Short Message service
  Allow to exchange messages between the User and the Station.

All the information reported in this paragraph were obtained through the official BEIDOU NSS page, the page of the Test and Assessment Research Center of China Satellite Navigation Office, which manages the whole system, from the workshop [5](R. Chengqi, 2012) and from the article on advances in BeiDou system [15](Rui Li et Al., 2020).

1.2 5G

With the acronym 5G we intend the 5th Generation of mobile communication technology, whose standards have been designed to provide a greater capacity and higher data speeds with respect to the previous generation LTE (Long Term Evolution), in order to meet the needs of an increasingly dynamic and connected society.

The incessant technological development in all sectors has made a definitive change in communication systems requirements. Referring to [1](Ahmad et Al, 2020), the services provided by the 5th Generation will be able to support the evolution of IoT (Internet of Things), bringing the number of connected devices up to 50 billion by 2030, in fact 5G will enhance the wireless connection between different devices (D2D, Device to device) such as sensors, actuators and electronic appliances by supporting mNTC (massive Machine Type Communication). In addition, another key innovation introduced by 5G is the URLLC (Ultra-Reliable Low-Latency Communication): Low Latency allows to optimize a network for the processing of large amounts of data in real time and, thanks to a new QoS (Quality of Service), 5G provides a network with instantaneous and intelligent systems; these advancements interest connected and autonomous cars, aerial vehicles, remote control of robots, industry automation (Industry Revolution 4.0), remote surgery and smart grid applications. Moreover, 5G will also improve eMBB

(enhanced Mobile BroaBband) that will supply high bandwidth internet access for wireless connectivity, large-scale video streaming, and virtual reality.

The diffusion of 5G started in 2019, but the implementation of all its potential is still under study. In fact, all the functionalities of 5G are carried out by specific enablers, each enabler has its own features, and the combinations of enablers define 5G technology. 5G has numerous enablers, among these, we will briefly describe those specifically related to the purpose of our work: integration of 5G signal in point positioning in urban environment.

1.2.1 mmWave

mmWave is an acronym for millimiter Wave and it consists in an ultra-high-frequency band that ranges from 24.25 GHz to 52.6 GHz, according to [6](J.A. Del Peral-Rosado, 2018), able to solve spectrum scarcity issues in the architecture. Thanks to the bands it is possible to provide high data rates ad ultra-high capacity and large bandwidth with low latency. The attenuation on losses is very important because can satisfy the demand over urban areas and allows the implementation of new applications. Among all applications, we find the possibility of using mmWave for high-accuracy positioning, in order to compensate for the problems related to positioning in harsh environments such as urban canyons.
This specific functionality allows us to lay the foundations for a hypothesis of integration of the 5G network in satellite positioning systems in all those contexts where visibility is strongly compromised. It therefore becomes the basis of our work: to analyze the geometric improvements obtainable from a hybrid system composed of GNSS and 5G signals.

1.2.2 massive MIMO

With MIMO we intend the Multiple-Input Multiple-Output technology, which consists of a system with multiple inputs and outputs that uses a system of multiple antennas both on the emitter and on the receiving side to improve communication performance, exploiting the concept of positive interference of multipath. In massive MIMO data rates are increased and interference are reduced by implementing large-scale and advanced antennas arrays, that can be controlled horizontally and vertically.

An example of a network infrastructure is provided in figure 1.1, according to [1]. The type of infrastructure proposed is called cell-free and is

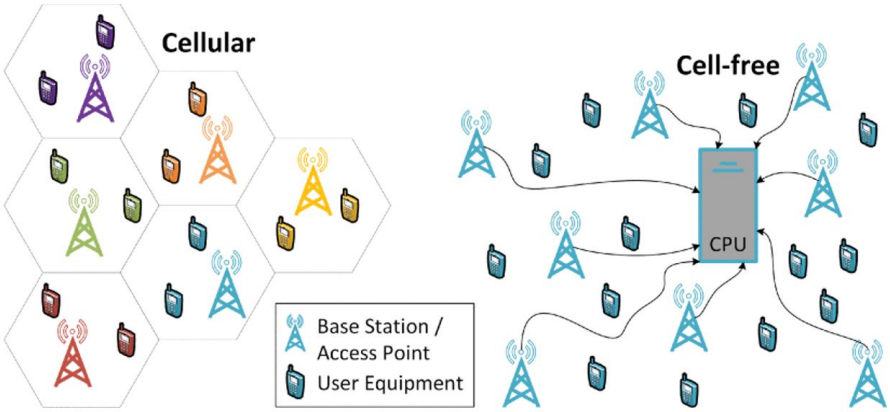

Figure 1.1: Comparison between cellular and cell-free massive MIMO systems [1]

based on the concept of distributing APs that serve the users with the signal encoding/decoding taking place in a CPU.

Chapter 2

Software development

The following chapter focuses on the description of the software's goals, components and specific requirements. Aim of the software is to allow the user the possibility to perform four main tasks:

- Rinex file reader, conversion of the information contained in the file in Python objects, that allow to perform plotting and analysis.
- Computation of GPS satellites' Earth-fixed position, velocity and clock offset, starting from Rinex navigation data.
- Estimation of receiver's position in global cartesian reference system by performing least squares point positioning on code observation equations.
- Provide a series of functions that allow to carry out the main transformations between reference systems.

The chapter is organized as follows: a first paragraph is intended to define the software environment, functional requirements and fundamental aspects. Then, the modules implemented in order to reach the software's goals.

2.1 Software Specification

The software is completely written in Python and it has been developed in Spyder-Anaconda environment.
The following packages are required:

- pandas

- numpy
- math
- matplot
- geoRinex

The packages can be installed through Anaconda terminal by launching the command 'conda install -name of package-' or 'pip install -name of package-', followed by the name of the package to be installed.

The software is organized in four modules, each one contains multiple functions, competing to the same macro area of topic, aimed to achieve the same goal. In particular, each function can be called from the others, whether they are in the same module or belong to different modules, by following this syntax:

```
import nameOfModule
var = nameOfModule.nameOfFunction(args, ...)
```

Following, a brief summary of the modules' content.

1) *Rinexreader.py* contains the functions implemented for reading precise ephemerides Rinex file (.SP3 format) and for reading the Ionospheric Correction paramethers contained in navigation Rinex files.

2) $transformation.py$ contains the functions for the principal conversions between the different Coordinate Systems, described in Appendix A.

3) $function.py$ contains all the functions involved in the computation of the position, velocity and clock offset of the satellites.

4) *pointPositioning.py* contain the function that perform Point Positioning on code observation of the in-sight satellites.

In 'main.py' we have an example of use of the principal functions contained in the different modules, in order to show to the user how to import and use them in the script to achieve the set goals.

2.2 Rinex reader

First goal of the software is to provide the user the possibility of reading and loading data from Rinex files in the working environment, using proper structures to facilitate and speed up the plotting and analysis of large amounts

of data.

Since different Rinex file formats exist, the software aims to offer a complete solution for reading Rinex NAV, OBS and SP3; by integrating new code to already available solutions.

Regarding Rinex Navigation and Observation files, we decided to implement the reading through the functionalities offered by GeoRinex package. The package must be installed in the working environment and imported in the script. GeoRinex offers the possibility, through the function 'geoRinex.load(...)', to convert Navigation and Observation Rinex files (including Hatanaka compressed OBS) in xarray.DataSet objects. Both Rinex 3.x and Rinex 2.x are supported. Below, an example of use of the 'load' function is provided:

```
import geoRinex as gr
ds = gr.load('path to Rinex') #Loaded as xarray
dsToDF = obs.to_dataframe() #Conversion to DataFrame object
```

Key aspect of the load function is its flexibility in data selection: thanks to a series of parameters that can be passed to the function, it is possible to select the data of interest of the user. For instance, it is possible to set time bounds to load only specific data with 'tlim' parameter, an example is given below:

```
start = '2021-05-26T12:59'
stop = '2021-05-26T13:13'
ds = gr.load(my.rnx', tlim=[start, stop])
```

Moreover, other useful parameters that can be passed to the function for loading OBS Rinex are the parameters 'meas' and 'use'. The first one allows to read from OBS Rinex only the observations specified by the user, in order to speed up the program and save memory, for example in our work we decided to elect only the code observation values, as shown below:

```
meas = ['C1', 'C2']
```

The 'use' parameter allows to read from NAV and OBS Rinex only the information related to selected satellite systems. The following example shows how to employ the 'use' parameter for the selection of Galileo and GPS infrmation.

```
use = ['E', 'G']
```

Regarding the SP3, that are the Rinex files containing the precise ephemerides, we decided to implement an ad-hoc function to read and convert them in pandas.DataFrame structures, in order to speed up the process and fit at best software's goals.
The function, contained inside *RINEXreader.py* module, takes as input the path to the Rinex containing the ephemeris, reads all the lines and stores the collected information (satellite position and clock offset) in a table, indexed by the time and the name of the satellite.

RINEXreader.py module also contains two other functions, developed to extract specific information from Rinex files:

- $RINEXreader.getIonoParams(path)$ takes as input the OBS Rinex and returns the alpha and beta parameters, required to compute the Ionosphere correction term in Point Positioning with Klombuchar model.
- $Rinexreader.getRecStartPos(path)$ takes as input the OBS Rinex and returns the approximate position of the receiver contained in the observation file, used to inizialize the iteration cycle in Point Positioning.

2.3 SV position and velocity computation

Key aspect of the software is to provide a module that allows the user to compute satellites' position and satellites' velocity for a given epoch, according to the navigational parameters retrieved from Rinex navigation file. The module is developed and tested to be able to determine the positions and velocities of the satellites belonging to GPS constellation. The formulas shown in the following refers to the GPS system and can be extended to Galileo, by applying the respective corrections. The implemented formulas refer to those distributed by GPS Enterprise through the technical document IS-GPS-200 [3] and to the book containing GPS theory [2] (GPS Satellite Surveying, Fourth Edition, Alfred Leick Ph.D,2015).

All the functions implemented to compute and validate the satellites' positions, velocity and clock offset are contained inside $functions.py$ module.

GPS navigation Rinex are read and loaded in Python working environment through GeoRinex load function, (python package, see 2.2). The parameters are stored in a DataFrame indexed by the time and the name of the satellite. The required parameters change every two hours and are transmitted from

the satellite to the receiver. This means that, starting from a certain time and for the next two hours, the formulas for the position computation will be applied using the same parameters, which means that the only variable value is the time. From here on we will refer with the term 'time slot' to the time interval during which the parameters doesn't change.

Focusing on GPS, for each satellite we have 12 time slots per day, starting at midnight. By convention, the start of a time slot is taken at the round hour (i.e. '2021-05-26 00:00:00', '2021-05-26 02:00:00', '2021-05-26 04:00:00').

Since errors can occur in the registration of the timestamp of the start of a time slot (i.e. '2021-05-26 15:59:44' is rather than '2021-05-26 16:00:00'), the function '$functions.fixTime(...)$' is applied to the DataFrame field containing the 'time' to correct the wrong timestamps and round them to the closest reference value.

The user has the possibility to establish the time range within which he wants to compute the position and velocities of the in sight satellites on the receiver.

The list of the in-sight satellites for each epoch is obtained through the OBS Rinex file of the same day: all the observation are loaded through GeoRinex (see 2.2), then, with a loop cycle on all the epochs, the available satellites are evaluated according to the recorded observations.

The function $functions.satPar(...)$ is available to assign to each satellite, for a given epoch, the navigational parameters according to the time slot they belong to. In particular, if a satellite is in-sight, but there are no navigation data of the time slot, the parameters of the subsequent time slot, if available within the next two hours, will be used to determine its position.

Position and velocity are computed by the function $functions.satPosVel(...)$, that takes as input the epoch, the satellite and the associated parameters and returns a DataFrame containing the measurements (satellite's Earth Fixed position in global cartesian reference system and velocity). All the parameters and formulas implemented are described in sections 2.3.1, 2.3.2, 2.3.3.

In conclusion, to evaluate the consistency of the obtained results, the soft-

ware provides two functions that allow to compare the values of the computed ephemerides with the a-posteriori precise ephemerides distributed by IGS and to check the precision of the computed velocities. A deeper description of the implemented algorithms is provided in section 2.3.4

2.3.1 GPS navigation parameters

The following paragraph shows all the navigational parameters required for the computation of GPS positions, velocities and clock offsets.

Table 2.1 contains all the GPS navigation parameter transmitted by the navigation message and loaded from the NAV Rinex.
Table 2.2 shows all the constant parameters used in the computations.

$SVclockBias$	s
$SVclockDrift$	s/s
$SVclockRate$	s/s^2
c_{rs}	m
Δn	rad/s
M_0	rad
c_{uc}	rad
e	
c_{us}	rad
$\sqrt{A}$	$\sqrt{m}$
t_{oe}	s
c_{ic}	rad
Ω_0	rad
c_{is}	rad
i_0	rad
c_{rc}	m
ω	rad
$\dot{\Omega}$	rad/s
$IDOT$	rad/s

Table 2.1: GPS SV's navigation paramethers

2.3.2 Formulas for SV's position and velocity computation

The following paragraph shows the formulas and procedures implemented for the computation of satellite's Earth-fixed position and velocity. Table 2.3 focuses on the formulas regarding the satellite's positions, the obtained

$\mu = 3.986005 * 10^(14)$	WGS 84 value of the earth's gravitational constant for GPS user.	m^3/s^2
$c = 2.99792458 * 10^8$	Speed of light.	m/s^2
$\dot{\Omega}_e = 7.2921151467 * 10^{-5}$	WGS 84 value of the earth's rotation rate.	rad/s

Table 2.2: Constant parameters

results are in $[meters]$. Table 2.4 contains the equations required to achieve the satellite's velocities in the three axes X,Y,Z. The results are given in $[m/s]$.

In particular, referring to the line 2 in table 2.3, we report, according to [3], the corrections applied for t_k evaluation. In the formula t is GPS system time at time of transmission, t_k shall be the actual total time difference between the time t and the epoch time t_{oe}, and must account for beginning or end of week crossovers. That is, if t_k is greater than 302,400 seconds, subtract 604,800 seconds from t_k. If t_k is less than -302,400 seconds, add 604,800 seconds to t_k.

In addition, is important to highlight that the process shown in line 8 of table 2.3 to compute the eccentric anomaly E_k is iterative: the value of E_k depends on the value of the variable itself computed with the previous iteration; in addition, to initialize the loop, we set E_{k-1} equal to the computed value of mean anomaly M_k. We decided to implement the iteration with a loop repeated 100 times.

1	t	GPS system time at time of transmission.	s
2	$t_k = t - t_{oe}$	Time from ephemeris reference epoch.	s
3	$A = (\sqrt{A})^2$	Semi-major axis	m
4	$n_0 = \sqrt{\frac{\mu}{A^3}}$	Computed mean-motion.	rad/s
5	$n = n_0 + \delta n$	Corrected mean motion.	rad/s
6	$M_k = M_0 + n * t_k$	Mean anomaly.	rad
7	$E_0 = M_k$	Initial value for eccentric anomaly.	rad
8	$E_k = M_k + e \sin E_{k-1}$	Eccentric anomaly, computed by iteration.	rad
9	$\nu_k = \arctan(\sqrt{\frac{1+e}{1-e}} \tan \frac{E_k}{2})$	True anomaly	/
10	$\Phi_k = v_k + \omega$	Argument of Latitude	rad
11	$\delta u_k = c_{us} \sin 2\Phi_k + c_{uc} \cos 2\Phi_k$	Argument of Latitude Correction	rad
12	$\delta r_k = c_{rs} \sin 2\Phi_k + c_{rc} \cos 2\Phi_k$	Radius Correction	rad
13	$\delta i_k = c_{is} \sin 2\Phi_k + c_{ic} \cos 2\Phi_k$	Inclination Correction	rad
14	$u_k = Phi_k + \delta u_k$	Corrected Argument of Latitude	rad
15	$r_k = A(1 - e \sin E_k + \delta r_k$	Corrected radius	m
16	$i_k = i_0 + \delta i_k + (IDOT)t_k$	Corrected Inclination	rad
17	$x'_k = r_k \cos u_k$ $y'_k = r_k \sin u_k$	Positions in orbitals	m
18	$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e)t_k - \dot{\Omega}_e t_{oe}$	Corrected longitude of ascending node.	rad
19	$x_k = x'_k \cos u_k - y'_k \cos i_k \cos \Omega_k$ $y_k = x'_k \sin \Omega_k - y'_k \cos i_k \cos \Omega_k$ $z_k = y'_k \sin(i_k)$	Earth fixed positions.	m

Table 2.3: Formulas to get SV Eart-fixed positions

2.3.3 SV's clock offset

The clock offset describes the difference between the satellite's clock and the GPS time and it's required in order to perform Point Positioning.

$$dts = t_s - t_{GPS}$$

The offset can be accurately computed through a 2nd order polynomial:

$$dts = SVclockBias + t_k SVclockRate + t_k^2 SVclockRate$$

$\dot{E}_k = \frac{n}{e \cos E_k}$	Eccentric anomaly rate.
$\dot{\nu}_k = \dot{E}_k \frac{\sqrt{1-e^2}}{1-e \cos E_k}$	True Anomaly Rate.
$\frac{di_k}{dt} = (IDOT) + 2\dot{\nu}_k(c_{is} \cos 2\Phi_k - c_{ic} \sin 2\Phi_k)$	Corrected Inclination Angle Rate.
$\dot{u}_k = 2\dot{\nu}_k(c_{us} \cos 2\Phi_k - c_{uc} \sin 2\Phi_k)$	Corrected argument of Latitude Rate.
$\dot{r}_k = eA\dot{E}_k + 2\dot{\nu}_k(c_{rs} \cos 2\Phi_k - c_{rc} \sin 2\Phi_k)$	Corrected Radius Rate.
$\dot{x}'_k = dotr_k \cos u_k - r_k \dot{u}_k \sin u_k$	In-plane X velocity.
$\dot{y}'_k = dotr_k \sin u_k - r_k \dot{u}_k \cos u_k$	In-plane X velocity.
$\dot{x}_k = -\dot{x}'_k \dot{\Omega}_k \sin \Omega_k + \dot{x}'_k \cos \Omega_k - \dot{y}'_k \sin \Omega_k \cos i_k$ $-y'_k(\dot{\Omega}_k cos\Omega_k \cos i_k - \frac{di_k}{dt} \sin \Omega_k \sin i_k$	Earh-fixed X velocity m/s
$\dot{y}_k = -\dot{x}'_k \dot{\Omega}_k \cos \Omega_k + \dot{x}'_k \sin \Omega_k - \dot{y}'_k \cos \Omega_k \cos i_k$ $-y'_k(\dot{\Omega}_k sin\Omega_k \cos i_k - \frac{di_k}{dt} \cos \Omega_k \sin i_k$	Earth-fixed Y velocity m/s
$\dot{z}_k = \dot{y}'_k \sin i_k + y'_k \frac{di_k}{dt} \cos i_k$	Earth-fixed Z velocity m/s

Table 2.4: Formulas to compute SV's Earth-fixed velocities

Where $SVclockBias$, $SVclockDrift$ and $SVclockRate$ are parameters contained in GPS navigation data and t_k is the time from ephemeris reference epoch.

2.3.4 SV position and velocity check

The consistency of the obtained results is checked by two ad hoc functions, that compare the computed values with the expected ones.

Expected values for satellites' position and clock offset are retrieved from the a-posteriori precise ephemeris distributed by IGS. Since GeoRinex package doesn't offer a function able to read precise ephemerides' SP3 Rinex file, an ad hoc function has been developed, that takes as input the path of the SP3 Rinex file and returns a DataFrame containing information on time, name of the satellite and the respective position and clock offset, see Paragraph 2.2.

The function $functions.checkSatPos(...)$ computes the difference in absolute value between the computed values and the ones retrieved from IGS, then the function computes the mean, the standard deviation and evaluates the maximum values of the new series of data.

The module has been tested on the data collected by Milano's PS (Permanent Station) on May 26th 2021. Position and speed of the satellites in sight were computed for the whole day and the accuracy of the results was assessed with the developed functions. Table 2.5 shows the obtained statistical results. Since all the reported values are in meters, we can definitely consider the errors negligible, since the satellites orbit at a minimum distance of 500km from Earth's surface.

	Implemented Formula	Mean [m]	Max [m]
X	$abs(x_{s_{comp}} - x_{s_{IGS}})$	0.9	3.4
Y	$abs(y_{s_{comp}} - y_{s_{IGS}})$	0.8	3.4
Z	$abs(z_{s_{comp}} - z_{s_{IGS}})$	0.8	3.1
ts	$abs(t_{s_{comp}} - t_{s_{IGS}})$	$1.7 * 10^{-9}$	$1.1 * 10^{-9}$

Table 2.5: Results of analysis on difference between computed and expected values for SV's position, results cover the whole day of May 26th 2021

A more empirical method has been implemented in order to evaluate the consistency of the computed velocities, since their precise values are not contained in Rinex precise ephemerides files.

The realization of the function is based on the hypothesis of correctness of the data obtained so far: we assume, in fact, that the positions of the satellites computed with the $functions.satPosVel(...)$ function, in light of the statistical results obtainable with $functions.checkSatPos(...)$, are correct.

Fuction $functions.checkSatvel(...)$, for each epoch t, computes the position of the satellites in the selected epoch and the position of the satellites in the subsequent 3 seconds ($t_{next} = t + 3s$). The theoretical values of velocity's components are computed as follow:

$$\mathbf{v}_{TH} = \frac{\mathbf{x}_{comp}(t+3) - \mathbf{x}_{comp}}{3s}$$

The obtained values $\mathbf{v}_{TH}$ are then compared with the ones obtained with the function $functions.satPosVel(...)$. Then, the same statistical analysis performed on positions is computed also for the velocities: the absolut of the difference between the expected and the theoretical value is computed and the mean, the standard deviation and the maximum value are computed on the data series defined above. For continuity, the results of the accuracy analysis on velocities, shown in table 2.6, have been computed on the data collected on May 26th 2021.

	Implemented Formula	Mean [m/s]	Max [m/s]
V_X	$abs(Vx_{comp} - Vx_{TH})$	0.3	0.6
V_Y	$abs(Vy_{comp} - Vy_{TH})$	0.2	0.6
V_Z	$abs(Vz_{comp} - Vz_{TH}$	0.4	0.7

Table 2.6: Results of analysis on difference between computed and expected values for SV's velocity, results cover the whole day of May 26th 2021

2.4 Point Positioning

With point positioning we intend the estimate of the absolute position of the receiver in single epoch by code observations. The implemented solution is based on the Least Squares Algorithm, which is the standard approach to approximate the solution of an over-determined linear system in which we have to manage errors.

The unknowns are the coordinates and the clock offset with respect to GPS time of the receiver:

$$\mathbf{X}_R = \begin{bmatrix} x_R \\ y_R \\ z_R \\ dtr \end{bmatrix}$$

The code observation are retrieved from OBS Rinex and loaded through GeoRinex's load function. Point positioning is performed on C1, that has the following observation equation:

$$C1_R^S(t) = \rho_R^S(t) + c * (dtr(t) - dts(t)) + I_R^S(t) + T_R^S(t) + dt_{rel}^S(t)$$

Where:

- $C1_R^S(t)$ is the code observation for the considered epoch and satellite.
- $\rho_R^S(t)$ is the distance between the receiver and the satellite.
- $dtr(t)$ is the clock offset of the receiver, with respect to GPS time.
- $dts(t)$ is the clock offset of the satellite, with respect to GPS time.
- $I_R^S(t)$ is the ionospheric delay, computed with the Klobuchar model, that depends on receiver's position and meteorological conditions of the considered epoch. This error is due to the free electrons present in the atmosphere; it increases as the satellite's elevation decreases.

- $T_R^S(t)$ is the tropospheric delay, computed with the Saastamoinen model. Due to the air and the water vapor content in the atmosphere from 0 to 40km.

- $dt_{rel}^S(t)$ is the relativistic effect, that is due to the eccentricity of the satellite's orbit, computed according to the Special Relativity Theory.

For every observation, we assume that position, velocity and clock offset of the respective satellite are known; in our implementation, we decided to use the values computed with the function described in paragraph 2.1.
Since Point Positioning is an iterative process, the system requires to be initialized at the starting point with approximated values for the unknowns. The initial values for the receiver's coordinates are obtained from the Rinex observation file. The initial receiver clock offset is set to 0.

$$\tilde{\mathbf{x}}_R = \begin{bmatrix} \tilde{X}_R \\ \tilde{Y}_R \\ \tilde{Z}_R \end{bmatrix} \qquad\qquad dtr_0 = 0$$

2.4.1 Code Observations

Code observations, on which point positioning is performed, are retrieved from the Rinex Observation file of a receiver, for a selected day. The following paragraph focuses on the description of how code observations are retrieved.

The receiver identifies the in-view satellite through the C/A code, then it performs an electronic correlation between the internal replica of the code and the code received from the satellite, measuring the delay of the received code with respect to the one internally generated. The correlator estimates the shift between the two codes and converts it in time. Therefore, the receiver estimates the delay observed from the receiver to the satellite.

$$\Delta T_R^S(t) = t_R(t) - t^S(t - \tau_R^S)$$

Where τ_R^S is the travel time of the signal between the satellite and the receiver ($\simeq 66ms$).
Remembering that:

$$t^S(t - \tau_R^S) = t - \tau_R^S + dt^S(t - \tau_R^S)$$
$$t_R(t) = t_R(t) + dt_R(t)$$

We introduce the hypothesis that the clock offset of the satellite dt^S does not change during the travel time: $dt^S(t - \tau_R^S) = dt^S(t)$.

By sobstitution, we obtain:

$$\Delta T_R^S(t) = \tau_R^S + dt_R(t) - dt^S(t)$$

The observed delay depends on both the receiver's clock and the satellite's clock.

By multiplying the time observation $\Delta T_R^S(t)$ by the signal travel speed in vacuum we obtain the code observation. Since it is an electronic measurement, it is subject to electronic noise ν. Modern receivers have ν in the order of $20 - 30cm$ for C/A and $20 - 30cm$ for P(Y).

However, the code observation is not only affected by electronic noises, but also by disturbances, that are all those phenomena that alter the metric content of the observations.

2.4.2 Atmospheric disturbances and corrections

In absence of the atmosphere, the signal travels at speed of light in vacuum c. In this condition we can evaluate the distance between as follow:

$$\rho_R^S = \tau_R^S * c$$

However, in the Earth's atmosphere, the speed of propagation and the travel itself of the signal undergoes variations due to the physical state of the medium crossed, which cause the so-called disturbance or atmospheric delay. We need to consider that:

- The signal, crossing the atmosphere, follows a curved path, longer than the geometric distance, named Minimum Time Path (according to Fermat Law).
- The speed propagation varies from point to point, according to the atmospheric condition

$$\tau_R^S * c = \rho_R^S + \Delta_R^S$$

Where Δ_R^S is the atmospheric effect in unit lenghts:

$$\Delta_R^S = \int_\rho (\eta(x) - 1)dx$$

In particular, the atmospheric layers that interact the most with the satellite's signals are the Ionosphere and the Troposphere. In the following, the ionospheric and tropospheric disturbances and the models implemented for their correction will be described.

Ionospheric correction

Ionospheric disturbance is due to the presence of ions and free electrons in the layer of atmosphere between 100 to 1000 kilometers in altitude, which interfere with the propagation of the signal.
The signal propagation speed ν varies according to the local refraction index η:

$$\nu(r) = \frac{c}{\eta(r)}$$
$$\eta_{code}(r) = 1 + \frac{A}{f^2} N_e(r)$$

Where:

- $A = 40.3 m^3/s^{-2}$ constant value.
- f is the frequency of the signal.
- $N_E(r)$ is the local electrons density in $electrons/m^3$

The total ionospheric delay can be expressed as:

$$I_R^S = \int_\rho \frac{A}{f^2} N_e(x) dx = \frac{A}{f^2} \int_\rho N_e(x) dx$$
$$I_R^S = \frac{A}{f^2} TEC_R^S$$

Where TEC_R^S is the Total Electron Content for the layer passed by the signal on the path between the satellite and the receive.
The ionospheric delay:

- Increases with solar radiation and solar intensity (we expect higher values for the daytime).
- Increases as the elevation of the satellite decreases.

The computation of the ionospheric correction parameter (in m) is implemented following the Klombuchar model, that is a mathematical model of the Ionosphere, that allow to evaluate I_R^S according to latitude, longitude and time of the receiver. The implemented function depends on eight parameters α_i and β_i (with $i = 0, 1, 2, 3$) that are contained in the header of the Rinex Observation file.

The module *pointPositioning.py* contains the function *ionoCorrection*(...) that takes as input the above mentioned parameters and returns the ionospheric correction.

Tropospheric correction

The tropospheric delay is due to air and water vapor presents in the atmosphere layer between the Earth surface and 40 kilometers in altitude. The local refraction index is given by:

$$\eta(r) = 1 + k_1 \frac{P(r)}{T(r)} + k_2 \frac{e(r)}{T(r} + k_3 \frac{e(r)}{T^2(r)}$$

Where k_1, k_2, k_3 are constant values, $P(r)$ and $T(r)$ are the local pressure and temperature in Kelvin). The final tropospheric delay is given by the integration of $\eta(r)$. It is minimum for dry air and cold weather and increases with hot and humid air.

In the software, the estimate of tropospheric correction is implemented following the Saastamoinen model, that is a mathematical model of the troposphere. The function *saastamoinenModel*(...) contained in *pointPositioning.py* module takes as input the elevation and the height of the satellite with respect to the receiver. Pressure, temperature and humidity are modeled in height, according to Bergas formulas.

2.4.3 Relativistic effect

In the computation of the parameters of the observation equation, we must consider another term related to the satellite's clock offset: the relativistic effect. In fact, $dt^S(t)$ is given by:

$$dt^S(t) = dt_0^S(t) + dt_{rel}^S(t)$$

Where, the first term is the clock offset computed according to the formulas shown in chapter 2.3.2, while $dt_{rel}^S(t)$ is the relativistic term, computed according to Relativistic Theory and due to the eccentricity of satellite's orbit.

$$dt_{rel}^S(t) = -2 * \mathbf{x}^S(t) * \dot{\mathbf{x}}^S(t)/c$$

This value ranges between 0 and 5 meters and depends exclusively on the ephemeris of the satellite.

2.4.4 Linearization of the observation equations

Since the Least Squares method is applicable only to a set of linear equations, the processing of GPS observations involves the linearization of the observation equations with respect to the coordinates and the clock offset of the receiver.

Starting point of the linearization process is the general notation for the distance between the satellite and the receiver:

$$\rho_R^S(t) = \tilde{\rho}_R^S(t) + \tilde{\mathbf{e}}_R^S(t) * \delta\mathbf{x}_R(t)$$

Where:

- $\tilde{\rho}_R^S(t) = \sqrt{(X^S(t) - \tilde{X}_R(t))^2 + (Y^S(t) - \tilde{Y}_R(t))^2 + (Z^S(t) - \tilde{Z}_R(t))^2}$
  Approximated distance between the satellite and the receiver.
- $\tilde{\mathbf{e}}_R^S(t) = (\mathbf{x}^S(t) - \tilde{\mathbf{x}}_R(t)) * \frac{1}{\tilde{\rho}_R^S}$
  Unitary vector.
- $\delta\mathbf{x}_R(t) = \begin{bmatrix} X_R - \tilde{X}_R \\ Y_R - \tilde{Y}_R \\ Z_R - \tilde{Z}_R \end{bmatrix}$
  Difference between the receiver's correct position and the receiver's approximated coordinates.

In the evaluation of the distance between the receiver and the satellite we must consider that the observed distance, is the one between the receiver at epoch t and the satellite at epoch $t - \tau$:

$$\rho_R^S(t) = ||\mathbf{x}_R(t) - \mathbf{x}^S(t - \tau)||$$

Where τ is the travel time of the signal between the satellite and the receiver. τ has an approximated value of $66ms$, during the travel time the satellite moved about $1000m/s * 0.066s = 66m$.
In addition, during the travel time also the Earth performs a rotation around its axis equivalent to $\tau\dot{\Omega}_E$, where $\dot{\Omega}_E$ is the angular velocity of Earth's rotation.
Furthermore, also the true reception time t is actually unknown, because we only know the receiver's clock reception time t_R and $t = t + dt_R$.
The satellite's position computation becomes:

$$\begin{gathered} \mathbf{x}^S(t - \tau_R^S) = \mathbf{x}^S(t_R + dt_R(t) - \tau_R^S) \\ = \mathbf{x}_{EPH}^S(t_R) + \dot{\mathbf{x}}_{EPH}^S(t_R) * (dt_R(t) - \tau_R^S) \\ -\tau_R^S \begin{bmatrix} 0 & 0 & \dot{\Omega_E} \end{bmatrix}^T \times \mathbf{x}_{EPH}^S(t_R) \end{gathered}$$

Where $\mathbf{x}^S_{EPH}$ represent the vector containing the ephemerides computed with the formulas seen in paragraph 2.3.2.

Proceeding with the substitution of the previous terms we obtain the following formula to be linearized:

$$\rho^S_R(t) = ||\mathbf{x}_R(t_R + dt_R(t)) - \mathbf{x}^S(t_R + dt_R(t) - \tau^S_R||$$
$$= ||\tilde{\mathbf{x}}_R(t_R + dt_R(t) + \delta\mathbf{x}_R(t_R + dt_R(t)) - \mathbf{x}^S_{EPH}(t_R) - \dot{\mathbf{x}}^S_{EPH}(t_R)(dt_R(t) - \tau^S_R) + \tau^S_R \left[\omega\times\right] \mathbf{x}^S_{EPH}(t_R)||$$

Remembering that:

$$\tau^S_R = \frac{\rho^S_R(t)}{c}$$

The process proceeds with the linearization with respect τ^S_R, dt_R and the correction con coordinates receiver. The obtained final equation for ρ^S_R is the following:

$$\rho^S_R(t) = K^S_{1R}(t_R) * \tilde{\rho}^S_R(t_R) + \tilde{\mathbf{e}}^S_R(t_R) * \delta\mathbf{X}_R(t_R) + K^S_{2R}(t_R) * dtr(t_R)$$

Where:

- Not significant terms have been neglected (terms smaller than $5 \times 10^{-4}m$).
- $\delta\mathbf{X}_R = \begin{bmatrix} X_R - \tilde{X}_R \\ Y_R - \tilde{Y}_R \\ Z_R - \tilde{Z}_R \end{bmatrix}$
  This vector depends on the true value of the receiver's coordinate.
- $\tilde{\rho}^S_R = \sqrt{(\tilde{X}_R - X^S)^2 + (\tilde{Y}_R - Y^S)^2 + (\tilde{Z}_R - Z^S)^2)}$
  Approximated distance between the receiver and the satellite, computed using the approximated values of receiver's coordinates obtained in the previous iteration. At first iteration, the used starting values are obtained from Rinex obseravation file.
- $\tilde{\mathbf{e}}^S_R = (\mathbf{x}^S - \tilde{\mathbf{x}}_R) * \frac{1}{\tilde{\rho}^S_R}$
  Unitary vector.

- $\alpha^S_{1R} = -(\tilde{\mathbf{e}}^S_R \cdot (\dot{\mathbf{x}}^s + [\omega\times]\mathbf{x}^S))$
- $\alpha^S_{2R} = -(\tilde{\mathbf{e}}^S_R \cdot \dot{\mathbf{x}}^s)$

- $K_{1R}^S = 1 - \alpha_{1R}^S/c - (\alpha_{1R}^S)^2/c^2$
- $K_{2R}^S = -\alpha_{2R}^S + \alpha_{1R}^S * \alpha_{2R}^S/c$

The obtained formula for $\rho_R^S(t)$ is substituted in the code observation equation, in this way we obtain a set of n linear equations, where n is the number of in-sight satellites.

After the replacing and rearranging the terms we obtain the following equation:

$$C1_R^S = [K_{1R}^S * \tilde{\rho}_R^S + I_R^S + T_R^S + dt_{rel}^S - c * dts] + \begin{bmatrix} \tilde{\mathbf{e}}_R^{S^T} * K_{1R}^S & (K_{2R}^S + c)/c \end{bmatrix} \begin{bmatrix} \delta \mathbf{X}_R \\ c* \end{bmatrix}$$

2.4.5 Least Squares implementation

The final formula for the observation equation, obtained by the linearization with respect to the position and the clock offset of the receiver, provides a set of n equations in 4 unknowns. Taking up the notation used in the previous chapter we can rewrite the system in a more compact notation, by introducing to the deterministic model of Least Squares:

$$\mathbf{y} = \mathbf{b} + A\hat{\mathbf{x}}$$

and the stochastic model of Least Squares:

$$\mathbf{y}_0 = \mathbf{b} + A\hat{\mathbf{x}} + \underline{\epsilon}$$

Where:

- $\mathbf{y}$ is a $[n \times 1]$ vector, that contains the observable, which are theoretical quatities.
- $\mathbf{y}_0$ is a $[n \times 1]$ vector, containing all the code observations retrieved from the n in-view satellites, these correspond to the observable plus $\underline{\epsilon}$, which is the term that represents the observation noise that affects the observables .
- $\mathbf{b}$ is the $[n \times 1]$ vector containing all the known terms.
- A is the matrix describing the coefficients of the unknown terms, it has dimension $[n \times 4]$, the first three columns describe the geometry of the satellite with respect to the position of the receiver, the fourth column contains the time coefficients.

$$A = \begin{bmatrix} e_{RX}^{S1} * K_{1R}^{S1} & e_{RY}^{S1} * K_{1R}^{S1} & e_{RZ}^{S1} * K_{1R}^{S1} & (K_{1R}^{S1} + c)/c \\ e_{RX}^{S2} * K_{1R}^{S2} & e_{RY}^{S2} * K_{1R}^{S2} & e_{RZ}^{S2} * K_{1R}^{S2} & (K_{1R}^{S2} + c)/c \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ e_{RX}^{Sn} * K_{1R}^{Sn} & e_{RY}^{Sn} * K_{1R}^{Sn} & e_{RZ}^{Sn} * K_{1R}^{Sn} & (K_{1R}^{Sn} + c)/c \end{bmatrix}$$

- $\hat{x}$ is the $[4 \times 1]$ vector containing the unknown terms, the ones that are going to be estimated in the iterative procedure.

$$\hat{\mathbf{x}} = \begin{bmatrix} X_R - \tilde{X}_R \\ Y_R - \tilde{Y}_R \\ Z_R - \tilde{Z}_R \\ c * dtr \end{bmatrix}$$

Least Squares solution in single epoch

The following paragraph will illustrate the formulas implemented for the estimation of the unknowns in single epoch, according to Least Squares theory.

The developed function *pointPositioning*(...), contained in *pointPositioning.py* module, takes as input a DataFrame containing the satellite's data to be processed and the path to the stored Rinex observation file.
Each row of the DataFrame associates to each satellite, for each epoch, its position, speed and code observation C1.

The algorithm processes data grouped according to the epoch considered and iterate the process up to a maximum value of 20 times, unless the system reaches before the convergence set to the millimeter.

First of all, an initial skimming on the in-view satellites is performed: for each satellite, azimuth and elevation are computed with respect the receiver's position (at the first iteration the coordinates are taken by the observation Rinex, for the following the coordinates estimated by the previous iteration are used), all the satellites having an elevation smaller than a specific cutoff angle editable by the user (we set it to 5°) are discarded, since the signal could be too disturbed by the surrounding environment or by the atmosphere.

Then, the vector $\mathbf{b}$ and the A matrix are computed through a loop on the table containing all the satellites' parameters.

The vector $\mathbf{b}$ containing the known terms is subtracted from the vector $\mathbf{y}_0$ containing the code observations.

$$\Delta\mathbf{y} = \mathbf{y}_0 - \mathbf{b}$$

Next, a new matrix required for the estimation process is introduced in the algorithm, the Q matrix is defined as follow:

$$Q = I * \eta_{el} = \underset{[n\times m]}{IdentityMatrix} * \eta_{el}$$

Where η is a weight vector of dimension $[n \times 1]$ containing the parameters η^{sat^i}:

$$\eta^{sat^i} = \frac{1}{\sin(elevation^{sat^i})}$$

$$Q = \begin{bmatrix} \eta^{sat^1} & 0 & 0 & \dots & 0 \\ 0 & \eta^{sat2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \eta^{satn} \end{bmatrix}$$

In this way, in the Least Squares process, the satellites with higher elevation will gain more weight in determining the solution, therefore those subject to less disturbance.

Once having defined all the matrices, the process moves on as follow:

$$N = A^T Q^{-1} A$$
$$\hat{\mathbf{x}}_R = N^{-1} A^T Q^{-1} \Delta\mathbf{y}$$

$$X_R = \hat{x}_R[0] + \tilde{X}_R$$
$$Y_R = \hat{x}_R[1] + \tilde{Y}_R$$
$$Z_R = \hat{x}_R[2] + \tilde{Z}_R$$
$$dtr = \hat{x}_R[3] * c$$

Once the iteration cycle has been completed over a specific epoch, the values of the estimated unknowns are saved in a DataFrame and the procedure moves on to the next epoch to be processed. At the end, the function returns the whole DataFrame, that contains the computed position and the clock offset of the receiver in the different times.

2.4.6 Obtained results and considerations on errors

The following paragraph provides an example of the results obtained by performing the implemented point positioning script. We decided to test the module on data dating back to May 26, 2021 from Milan Permanent Station (PS) and from Zimmerwald PS.

Navigation and Observation Rinex have been downloaded for Milan PS from SPIN3 GNSS website (*Servizio di Posizionamento Interregionale GNSS* for Piemonte, Lombardia and Valle d'Aosta - https://www.spingnss.it/). On the other hand, the Zimmerwald NAV and OBS Rinex were obtained through BKG GNSS Data Center (*Bundesamt fur Kartographie und Geodasie* - https://igs.bkg.bund.de/).

Satellites' ephemerides and velocities have been computed according to the methods described in paragraph 2.3.

We decided to perform Point Positioning every minute, from $26/5/2021-00:00:00$ to $27/5/2021-00:00:00$.

Starting with the analysis of the results obtained with Milan's data, figure 2.1 show the number of the in view satellites used to perform point positioning for each epoch. The plot shows only the satellites whose elevation is greater than the fixed cutoff angle (in our case 5°). The number of in view satellites ranges between a minimum of 7 and a maximum of 13: it can be deduced that for each epoch the number of available satellites is sufficiently high and allows to perform correctly the Least Squares algorithm since it is always greater than the number of unknowns to be estimated.

The function *pointPositioning*(...) return a DataFrame containing the position X_R, Y_R, Z_R in global cartesian and the clock offset of the receiver for each epoch. Table 2.7 shows the mean value and the standard deviation of each field of the DataFrame obtained with the processing of Milan's PS data.

	X_R	Y_R	Z_R	dt_R
Mean	4421978.9	718518.7	4524920.9	$3.1 * 10^{-4}$
Std	2.3	1.5	2.9	$2.7 * 10^{-4}$

Table 2.7: Mean and standard deviation on results of $pointPositioning$ function on Milan PS

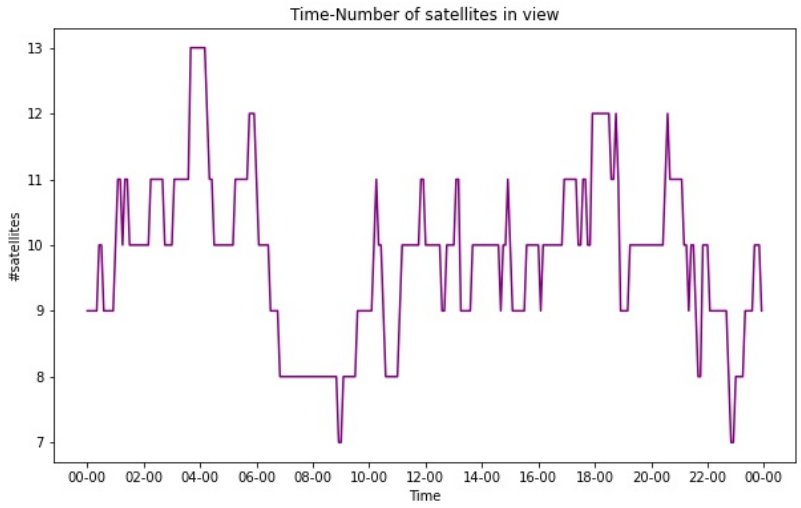

Figure 2.1: Plot of the number of in view satellites during the day of analysis May 26th 2021 over Milan PS

From a first statistical analysis, focusing on the standard deviation, we can already assume the goodness of the results obtained since the variability, intended as the dispersion of the data around their average value, is reasonably low.

In order to have a clearer idea on the consistency of the obtained results, we converted the estimated global cartesian coordinates of the receiver in local cartesian with respect to the correct position of the receiver (known a-priori because the receiver is located in a PS.

Figures 2.5, 2.6, 2.7 show the trend over time of the estimated local cartesian coordinates of the Milan receiver with respect the true coordinates, given in the following in global geodetic:

$$P_{0_{Milan}} = \begin{bmatrix} 45°28'42.127'' \\ 9°13'45.167'' \\ 191.125m \end{bmatrix}$$

Table 2.8 contains the mean, standard deviation and maximum in absolute value of East, North and Up coordinates of Milan case study. From the analysis of the statistical results and from the plot of the local coordinates shown in the figures, we observe that the estimates of the position of the receiver are placed around zero, with maximum values that do not exceed

	East	North	Up
Mean	0.5	1.8	-0.6
Std	1.4	2.2	3.0
Max	4.9	9.1	8.3

Table 2.8: Statistical analysis if Point Positioning results in local cartesian

10 meters. Considering that the accuracy of the GPS in the position handled through code observations is around 10 meters for the SPS (Standard Positioning Service - see 1.1.1) we can consider the results obtained more than satisfying.

The software was tested also on Zimmerwald's PS data, in order to verify the linearity of the results of the process. Figures 2.2, 2.3, 2.4 show the results obtained in terms of local cartesian cooordinates from the processing of the Zimmerwald PS data during the day May 26th 2021. The transformation in local cartesian was evaluated with respect to the coordinates of the Zimmerwald PS written below.

$$P_{0_{Zimmerwald}} = \begin{bmatrix} 46°56'37.527'' \\ 7°27'54.097'' \\ 956.451m \end{bmatrix}$$

The results obtained for the Zimmerwald case study confirm the previous ones seen for Milan: the coordinates in local cartesian are more than optimal, since they vary between -3 and +3 as regards the East and North components, while the Up varies between -6 and + 6. Also in this case the accuracy of the results can be considered excellent.

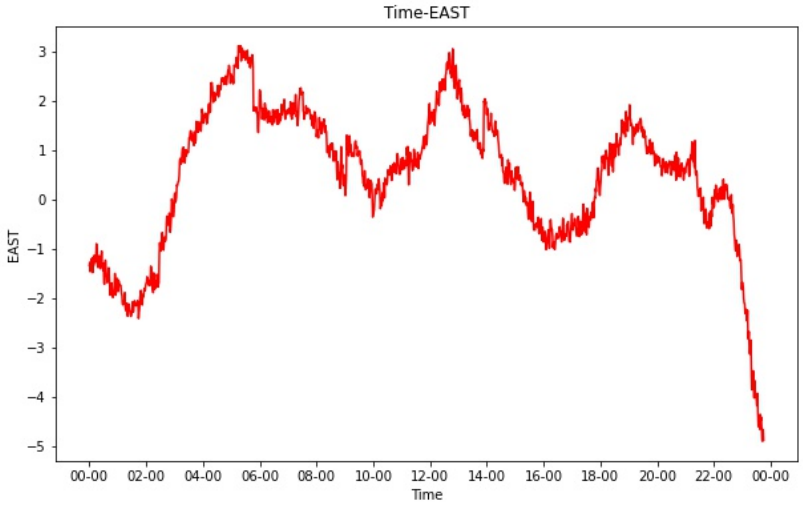

Figure 2.2: Plot of the trend of Milan's EAST component on May 26th 2021.

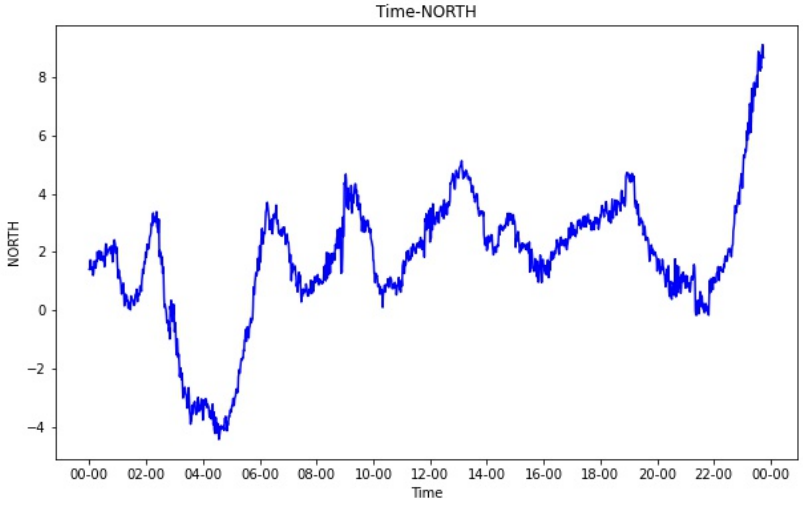

Figure 2.3: Plot of the trend of Milan's NORTH component on May 26th 2021.

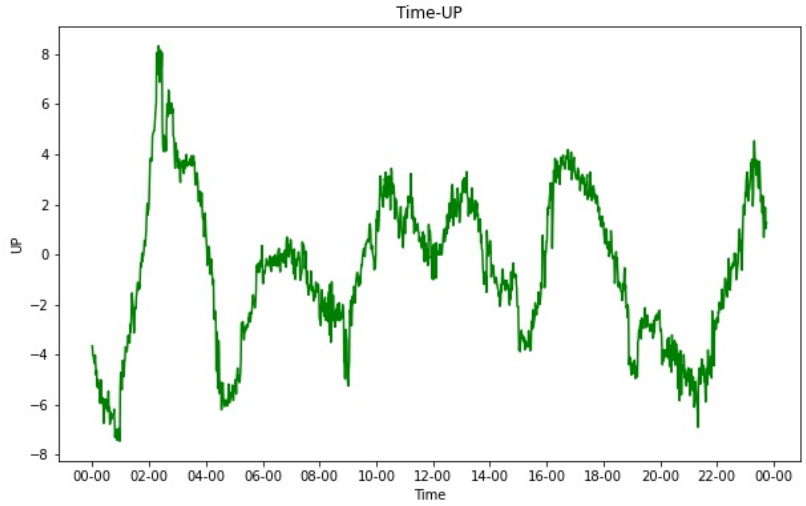

Figure 2.4: Plot of the trend of Milan's UP component on May 26th 2021.

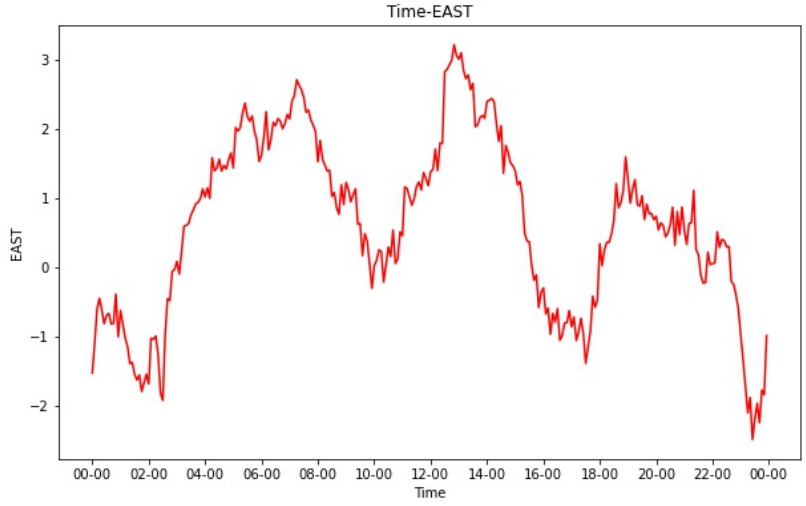

Figure 2.5: Plot of the trend of Zimmerwald's EAST component on May 26th 2021.

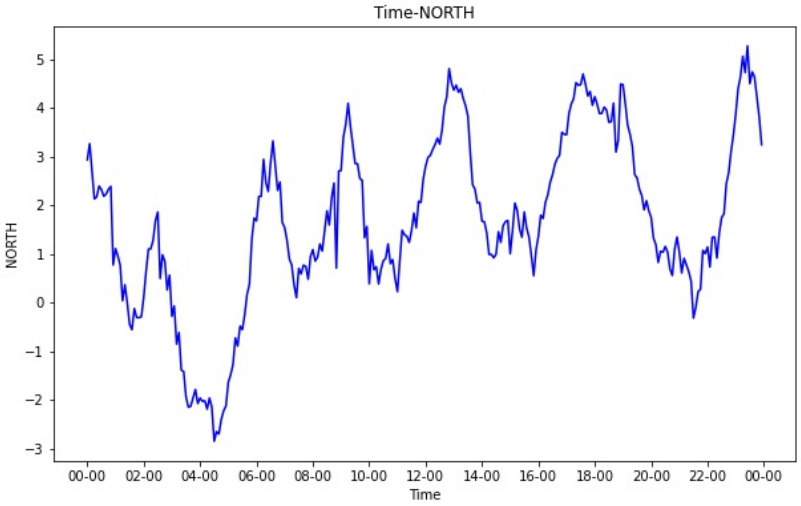

Figure 2.6: Plot of the trend of Zimmerwald's NORTH component on May 26th 2021.

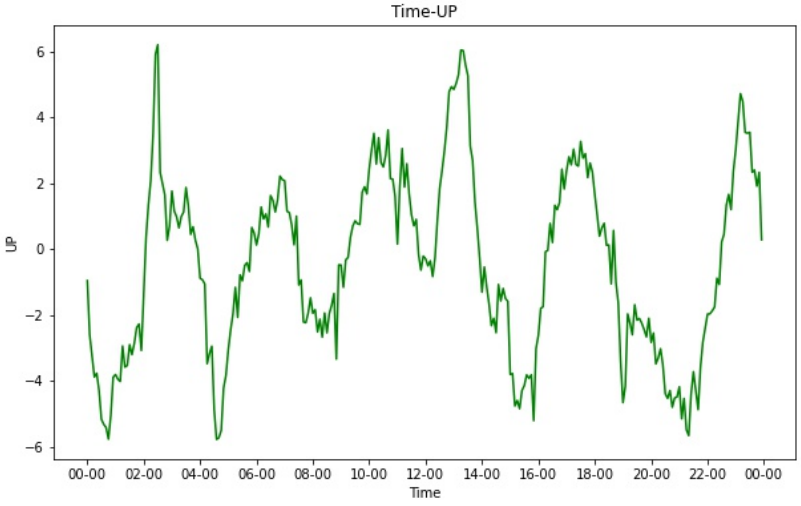

Figure 2.7: Plot of the trend of Zimmerwald's UP component on May 26th 2021.

Chapter 3

Simulation

The results obtained in the previous chapter, refer to the observations collected by a permanent station placed at a height of about 30 meters above the ground on the roof of a building. It therefore can be assumed that there are no obstacles between the GNSS antenna and the satellites, whose signals can be blocked or diverted by the presence of obstacles, such as buildings. All the satellites with elevation greater than 0 are considered in Line Of Sight (LOS). The proposed conditions describe well what happens in the positioning of a ground-level receiver in open field.

However, performance of Global Navigation Satellite System (GNSS) positioning in urban environments is hindered by poor satellite availability because there are many man-made and natural objects in urban environments that obstruct satellite signals.

Key point of this work is to provide an analysis of the GNSS visibility in a urban environment. To reach this goal we decided to develop a software that, given the positions of the satellites in different urban scenarios created ad-hoc for simulating the urban environment, is able to distinguish the satellites in LOS from those NLOS, whose signal is blocked by the presence of the buildings represented by the considered scenario.

Purpose of this chapter is to describe the methods and processes implemented for the realization of the simulation of the different urban scenarios and for the satellite masking process.

This step is fundamental in order to achieve the ultimate goal of this thesis work: provide an analysis of the visibility conditions of satellites in the pro-

posed scenarios and the possible improvement that could be obtained, from a geometrical point of view, in positioning in urban environment with the introduction of street level 5G signal transmitters.

In the paragraphs below will be defined: the four considered urban scenarios, the formulas implemented, methods for reading the precise ephemerides and the algorithm developed to reject satellites whose signal is blocked by the presence of buildings. In conclusion, the results of the masking process are shown.

3.1 Street Simulation

Starting point of the simulation is the realization of the four urban street scenarios. All the roads have the same width of 9 meters, comprising a dual carriageway and sidewalks on both sides of the streets. On the streets there are buildings of constant height, consistent with the hypothesis of performing the simulation in a urban canyon. The receiver is placed exactly in the middle of the road, along the traffic line. The developed scenarios are:

1 Street from East to West with buildings 24 meters high.

2 Street from East to West with buildings 9 meters high.

3 Street from North to South with buildings 24 meters high.

4 Street from North to South with buildings 9 meters high.

Each street have a total extension of 20 kilometers, in order to ideally represent an infinitely extended road, from -10 up to +10 kilometers with respect to the receiver that is placed exactly in the middle. The streets are implemented by a grid of points, whose latitude and longitude correspond to the constant planimetric position of the edges of the road:

- from -10 up to -5 kilometers points have been displaced every 100 meters.
- from -5 up to -2 kilometers points have been displaced every 50 meters.
- from -2 up to +2 kilometers points have been displaced every 1 meter.
- from 2 up to 5 kilometers points have been displaced every 50 meters.
- from 5 up to 10 kilometers points have been displaced every 100 meters.

In this way, the dot grid becomes more dense the closer to the receiver, as the elevation angle of the height of the building increases.

The grids representing the four different scenarios have been realized with a Python script. The whole procedure can be reassumed in 6 fundamental steps:

1 Definition of the horizontal coordinates of the beacons in local cartesian (with Up = 0) with respect to the receiver's position.

2 Conversion of the local cartesian in global cartesian (see Appendix A.3)

3 Conversion of global cartesian to geodetic (see Appendix A.1).

4 Setting of the actual height of the beacons as: $h_{ground} + h_{building}$.

5 Back conversion of the new coordinates from geodetic to global cartesian (see Appendix A.2.

6 Back conversion of global cartesian coordinates to local cartesian with respect to the receiver's position.

First of all, the points of the roads were defined in local cartesian coordinates, with respect to the position of the receiver.
For the simulation we decided to place the receiver in the permanent station of Milan, in Piazza Leonardo, at ground level.

$$P_R = \begin{bmatrix} 45°28'42.127'' \\ 9°13'45.167'' \\ 160.609m \end{bmatrix}$$

The local cartesian grid follows the proposed schema: taking as example the street running from East to West, the North component is +4.5m, for the North buildings of the streets, or -4.5m, for the South buildings. For each side of the street, the East coordinate ranges from -10000m up to 10000m according to the rule defined above. The Up component is initially set to 0 (ground level) for both sides of the streets.

The second step consists in the transformation of the points from local to global cartesian. The transformation consists in the product between the coordinates and a rotation matrix, whose terms depends only on the receiver's position. The rotation matrix reported in the following refers to the one described in Appendix A.3 and is the one required for the direct

transformation from global to local cartesian, since we are implementing the opposite transformation (from local to global cartesian) the matrix will be trasported in the final transformation formula.

$$lat_R = \text{latitude of the receiver}$$
$$lon_R = \text{longitude of the receiver}$$
$$R = \begin{bmatrix} -\sin lon_R & -\cos lon_R & 0 \\ -\sin lat_R \cos lon_R & -\sin lat_R \sin lon_R & \cos lat_R \\ \cos lat_R \cos lon_R & \cos lat_R \sin lon_R & \sin lat_R \end{bmatrix}$$

The transformation is performed with the following formula:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = R^T \begin{bmatrix} E_i \\ N_i \\ U_i \end{bmatrix} + \begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix}$$

Then, the obtained grid in global cartesian is converted in geodetic reference system, according to the formula shown in Appendix A in section A.1.
At this point, we have obtained a grid of points whose latitude and longitude can be considered fixed, the geodetic heights correspond to the distance of the road ground with respect to the surface of the reference ellipsoid. The fourth step consists in setting the value of geodetic heights by adding the desired building height to the value itself.
Taking for example a street configuration with buildings 24 meters height, we have:

$$P_{geodetic}^{final} = \begin{bmatrix} Latitude_i \\ Longitude_i \\ Height_i + 24 \end{bmatrix}$$

The final grid, containing the correct coordinates of the points representing the buildings, is then converted in global cartesian, referring to the formula shown in Appendix A.2. Then, we move to the last step of the procedure that consists in the conversion of the coordinates in local cartesian with respect to the position of the receiver, this last configuration is the one that we're going to use in the analysis phase:

$$P_{i,LC}^{final} = \begin{bmatrix} E_i^{final} \\ N_i^{final} \\ U_i^{final} \end{bmatrix} = R[P_{i,cartesian}^{final} - P_{R,cartesian}]$$

Where R is the rotation matrix defined above, $P_{i,cartesian}^{final}$ are the points converted from Global Geodetic to Global Cartesian (according to the formulas present in Appendix A, paragraph A.2) and $P_{R,cartesian}$ are the coordinates

of the receiver in Global Cartesian.

For the sake of completeness, the final configuration of the East-West road is shown as an example: figure 3.1 shows the road on the map of Milan (realized with QGIS), while figure 3.2 shows a three-dimensional view of the coordinates of the points representing the heights of the buildings for the road from East to West in local cartesian reference system. As we move away from the receiver the Up component of the buildings decrease, due to the Earth's curvature, while the Up of the points are with respect to a horizontal plane tangent to the Earth's surface and passing through the receiver. In the example are represented buildings with an height of 24 meters, we observe a minimum value in Up of $16.2m$, consequently we have a maximum deviation of about $7.8m$.

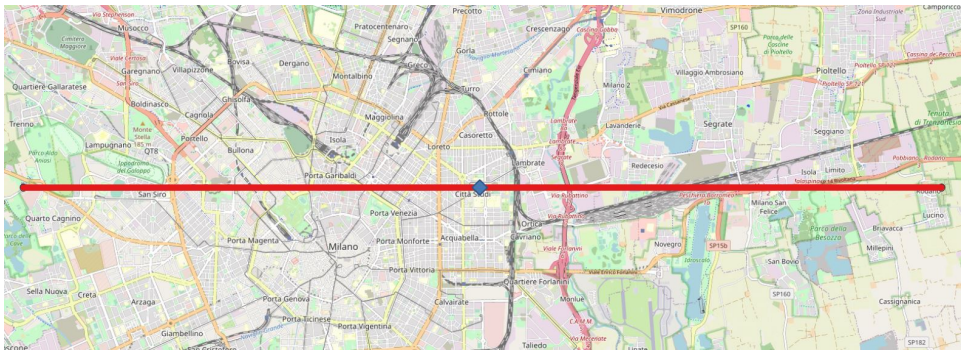

Figure 3.1: 2D plot on QGIS of the Global Geodetic coordinates of the point of the street running from East to West.

3.2 Satellite Masking

Once the four urban environment simulation scenarios have been created, the analysis proceeds with the evaluation of the satellites actually in LOS with the receiver. In fact, the presence of buildings obstructs the reception of the signals.

In our simulation, the receiver is placed in the middle of the road (see Figure 3.1) and has the latitude and longitude of the antenna of Milan Permanent Station, the height is set to the ground level.

$$P_R = \begin{bmatrix} 45°28'42.127'' \\ 9°13'45.167'' \\ 160.609m \end{bmatrix}$$

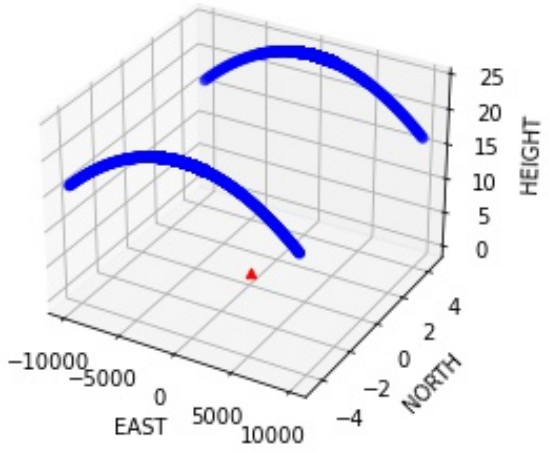

Figure 3.2: 3D plot of the Local Cartesian coordinates of the buildings, for the street running from East to West, with respect to the position of the receiver.

3.2.1 Theoretically in view satellites

First step is to obtain the positions of all the satellites that are theoretically in-view for the receiver (that rise beyond the horizon).

We decided to consider four different constellations of satellites in our analysis:

- GPS - prefix G
- Glonass - prefix R
- Galileo - prefix E
- Beidou - prefix C

At this point it is necessary to introduce a clarification regarding the satellites: we decided to consider in our simulation all the operational satellites present in the sky. The ability to receive signals depends on the technology of the receiver: older receivers will not be able to receive signals from the latest generation satellites. For example, the case of the BeiDou satellites is reported: the PS of Milan receives only signals from the constellations of GPS and Glonass, on the contrary the PS of Zimmerwald receives signals from GPS, Glonass, Galileo and BeiDou. In particular, the signals received

by BeiDou are those operational since before April 2018 (in total 23 out of the 42 currently operational).

Since the software developed and described in chapter 2.3 focuses on the computation of position and velocity of the satellites belonging only to GPS constellation, we decided to retrieve the positions from ESA's a-posteriori precise ephemerides, distributed by ESA's navigation office. This product furnish the position and the clock offset of each satellite of all the available constellations and it is available at *http://navigation-office.esa.int/products/gnss-products/*. The positions of the satellites are available every 5 minutes through the day and are a sufficiently large sample of data for our analysis.

Precise ephemerides are read with the developed function $readSP3(...)$ belonging to the module $RINEXreader$ described in paragraph 2.2. The function returns a DataFrame containing the position of all the satellites.

In order to narrow down to the theoretically in-view satellites for the receiver, the satellites coordinates have been transformed from global cartesian to local cartesian reference system, with respect to receiver's known position. For the conversion formulas we remind to Appendix A, paragraph A.3. For each satellite, starting from local cartesian coordinates, elevation and azimuth angles in degrees have been evaluated as follow:

$$d_H = HorizontalDistance = \sqrt{E^2 + N^2}$$

if $d_H < 0.1$	if $d_H > 0.1$
$Azimuth = 0$	$Azimuth = \arctan \frac{E}{N}$
$Elevation = 90$	$Elevation = \arctan \frac{U}{OD}$

Once having obtained the values of Azimuth and Elevation, a first scam on satellites' DataFrame is obtained by selecting only those having an Elevation grater than 0 with respect the receiver. In this way, we obtain a data structure containing the list of all in-view satellites, their coordinates in LC, azimuth and elevation for each epoch belonging to the time range. Figure 3.3 shows the number of satellites in view for each constellation available per instant of time. Instead, figure 3.4 shows the same results by introducing a cutoff of 5 degrees on the elevation of the satellites.

3.2.2 Masking and LOS satellites definition

Next step is to load the grid of points representing the buildings simulation in local cartesian and compute for each point their azimuth and elevation

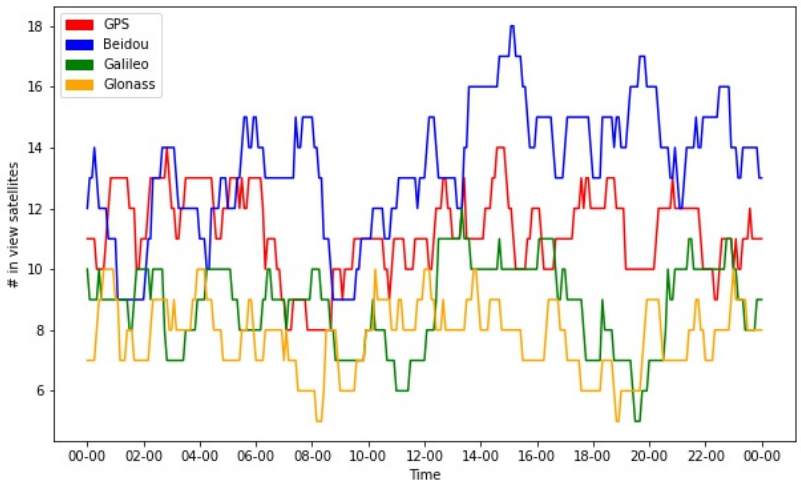

Figure 3.3: Number of in view satellite per constellation per epoch.

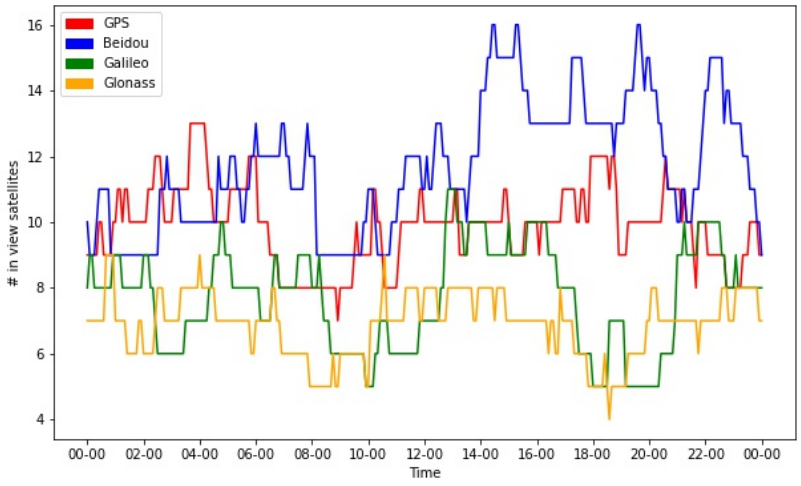

Figure 3.4: Number of in view satellite per constellation per epoch with a cutoff of 5°.

with respect to the receiver's position, following the procedure described above. The buildings configuration represented by the grid constitutes the actual mask through which we will filter the visible satellites.

We expect that the signals of all the satellites, whose elevation is smaller than the elevation of the buildings surrounding the receiver, do not reach the receiver, because they are blocked by the constructions.

To perform this task, a loop has been implemented which, for each satellite theoretically in-sight, assigns it the building point with the closest azimuth. Then, the elevation of the satellite is compared with the one of the point:

- If the satellite's elevation is greater than the building, than the satellite is considered actually in-view, since there are no obstacles on the signal's path.
- If the satellite's elevation is smaller than the building, the satellite's signal is blocked by the building, therefore, the satellite is discarded.

3.2.3 Masking results

The masking process aims to simulate the effective visibility of satellites in urban environment, in particular in the worst case of an urban canyon (corresponding to the simulation with buildings 24 meters high).
In open sky condition, or in absence of buildings, considering all the available satellites' constellations (GPS, Galileo, Glonass and BeiDou), we have an average number of 41 satellites in-view, with a peak of 51 satellites and a minimum of 31. In particular, in our analysis, we consider in-view all the satellites with elevation greater than zero, computed with respect to the position of the receiver.

The introduction of roadside buildings leads to a notable decrease in the number of satellites in sight, as expected, this number decreases with the height of the buildings. Table 3.1 shows in the fist 3 columns the maximum, minimum and average values of in sight satellites for the 4 configurations. The last three columns ($n_s < 4$, $n_s < 5$, $n_s < 6$) reports the number of epochs in which the number of in LOS satellites is smaller than a reference value, in particular if n_s is smaller then 4 then it is impossible to perform point positioning, corresponds to the lowest level of redundancy required. The same analysis was also carried out by introducing a cutoff of 5 degrees on the elevation of the satellites, but the results are not reported as they do

not change significantly.

	Mean	Max	Min	$n_S < 4$	$_S < 5$	$n_S < 6$
East-West - 9m height	$\simeq 13$	20	8	0	0	0
East-West - 24m height	$\simeq 5$	10	1	64	128	189
North-South - 9m height	$\simeq 5$	16	4	0	5	18
North-South - 24m height	$\simeq 4$	8	0	157	209	250

Table 3.1: Satellites in LOS availability and critical number of epochs in urban environment

A first analysis shows that, in some epochs, the visibility of the satellites is strongly compromised in the simulation of a urban canyon (buildings height equal to 24 meters).

Figures 3.5, 3.6, 3.7, 3.8 show a comparison between the number of satellites theoretically in-sight (in open sky condition, always plotted in red) and the actual ones, obtained as a consequence of the buildings masking effect. Each chart represent the masking both at 9 and 24 meters for each road type (East-West and North-South). We observe a significant decrease in the number of available satellites in a highly urbanized environment and, as we expected, the number of satellites actually in-sight decreases as the high of the buildings increases.

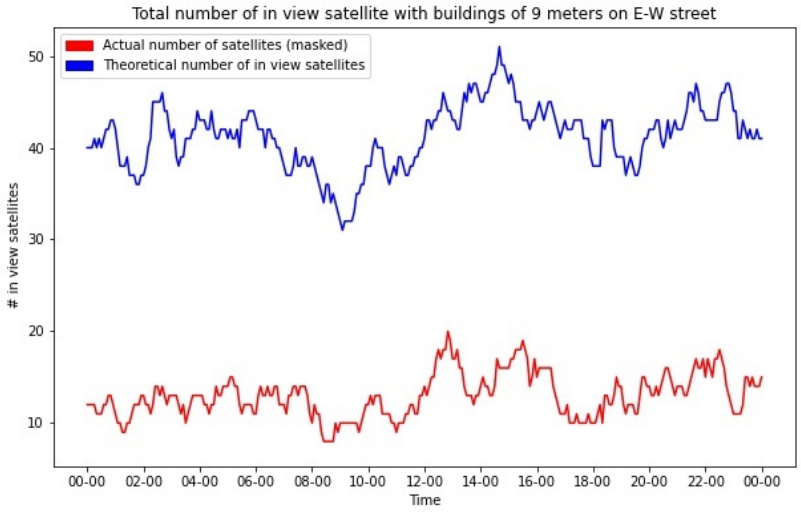

Figure 3.5: Masking effect on in view satellites with buildings height of 9 meters on a street running East to West.

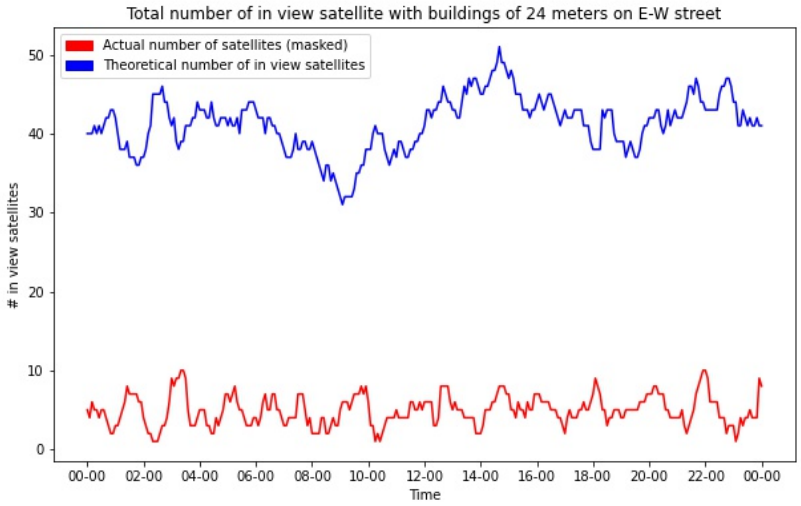

Figure 3.6: Masking effect on in view satellites with buildings height of 24 meters on a street running East to West.

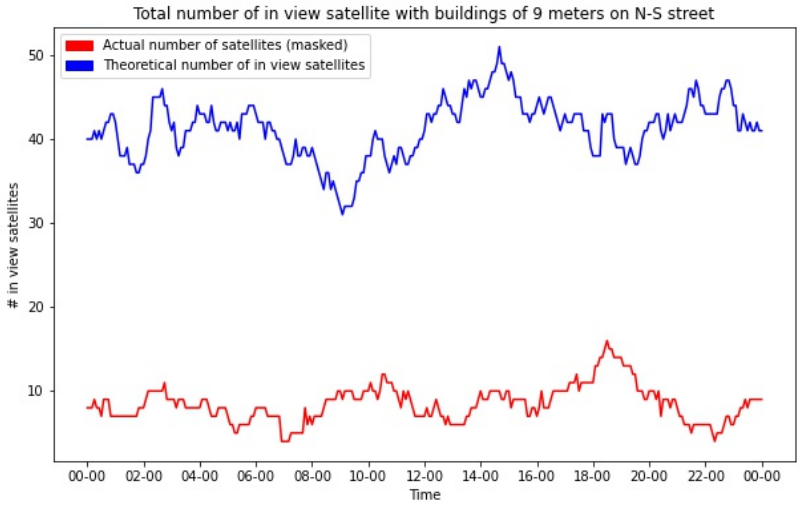

Figure 3.7: Masking effect on in view satellites with buildings height of 9 meters on a street running North to South.

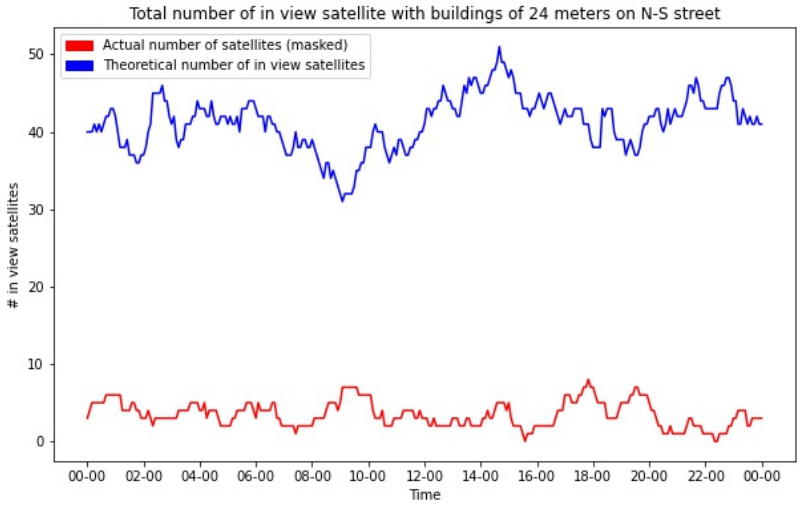

Figure 3.8: Masking effect on in view satellites with buildings height of 24 meters on a street running North to South.

Chapter 4

5G integration

From what emerges from the visibility study carried out in urban canyons and described in the previous chapter, the actual number of in LOS satellites capable of sending their signal to the street-level receiver in urban environment is considerably lower than the number of available satellites (over the horizon). This condition, severely affects the possibility to successfully perform point positioning in single epoch with an acceptable margin of error, in addition, for many epochs it is even not sufficient for positioning purposes (see table 3.1). In fact, the urban environment provides perfect conditions for non-line-of-sight (NLOS) situations, since the buildings block the signal [13](Julia Brebler, Marcus Obst, 2017). The presence of buildings therefore, not only blocks the signal of the satellites (NLOS), but also causes the multiple reflection of the GNSS signal (Multipath effect). A deeper analysis on Multipath Effect is provided in [14](T. Kos et Al., 2010). A schematic representation of NLOS and Multipath phenomena is provided in Figure 4.1.

Therefore, Global Navigation Satellite System (GNSS) is not suitable for the dense urban or indoor environments as the satellite signals are very weak and disturbed. In addition, this problem has been detected also in airports [12](Audrey Guilloton et Al., 2012).

Nowadays, Positioning of devices in wireless networks continues to become more important for future applications and use cases like emergency services, vehicular use cases, location-based services market (LBS) and for the transition to smart cities.

Meanwhile, Positioning is an important application of the fifth-generation (5G) communication system. Millimiter wave (mmWave) technology is one

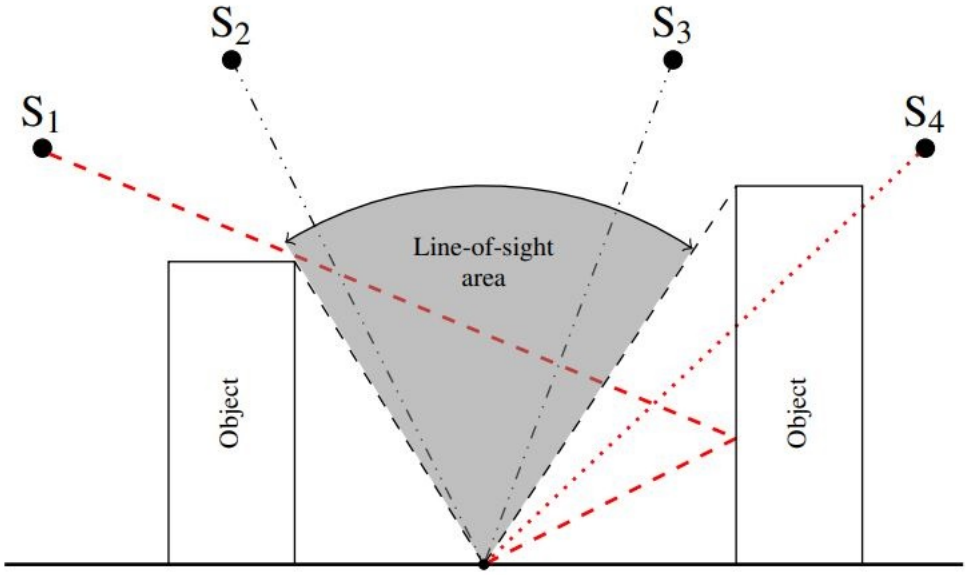

Figure 4.1: NLOS and multipath effect in urban environment.

of the key properties of the 5G communication networks and will be able to provide extremely high localization accuracy, a deeper analysis on mmWave positioning application is given in [17] (Solaranta et Al., 2017).

A possible solution to the problems of GNSS positioning in urban scenario is given by the GNSS integration with the 5G communication system. The goal is to provide an hybrid solution, where 5G observations can be added to point positioning GNSS solutions to improve positioning performance.

Goal of our work is to provide an analysis of the geometrical improvement obtained in the configuration with 5G integration with respect to the receiver. We assume that 5G signal is transmitted by a network of beacons, placed on the sides of the street at a fixed height with respect the ground. We assume that the positions of the beacons is known and that each outdoor point is reached by at least four 5G signals.

In the developed simulation, we assume that the 5G signal is able to transmit time observations, therefore to communicate the Time of Arrival (ToA) to the receiver, which corresponds to the absolute instant of time in which the signal transmitted by the beacon reaches the receiver.

Consistently with what have been said so far, we assume that the estimation of the receiver's position in single epoch will be performed by applying Least Squares Method to the available observation equation. In addition, we assume that the 5G observations can be integrated in the Least Squares process, in such a way to increase the available observations and therefore the redundancy of the system.

In the following, the methods to evaluate the quality of the configuration geometry and Positioning accuracy will be discussed, as well as the implementation of the beacons' network. Then, the evaluation methods will be applied on the original configuration (only GNSS, see chapter 3) and on the integrated configuration (GNSS+5G). In addition, we remind that the evaluation of the quality index will be performed on the satellites actually in LOS, identified with the masking method proposed in paragraph 3.2.2.

4.1 Precision indexes - PDOP and HDOP

In our simulation, Point Positioning is performed in single epoch and the estimate of receiver's position is obtained by applying Least Squares algorithm to m observation equations. The accuracy in estimating the position of a receiver depends not only on the errors present in the individual observations, discussed in Chapter 2, but also on the geometry of the available satellites with respect to the receiver.

Since our analysis is aimed at evaluating the geometry of the positioning system, we decided to compute, for each observation epoch, the values of the Positional Dilution Of Precision (PDOP) and the Horizontal Dilution of Precision (HDOP) indexes.

With Dilution of Precision (DOP), in satellite navigation, we intend the effect of the satellite navigation geometry on the precision of positioning. In other words, it measures the strength of the satellite configuration. The higher the value of DOP, the worse is the signals geometry configuration. DOP are computed from final covariance matrix and are:

1 HDOP: Horizontal DOP, geometric accuracy index for planimetric positioning.

2 VDOP: Vertical DOP, geometric accuracy index for height estimate.

3 PDOP: Positional DOP, sum of HDOP and VDOP, geometric index of accuracy of the estimated position.

4 TDOP: Time DOP, accuracy index for the estimate of the receiver offset with respect to the time of the satellite.

5 GDOP: Global DOP, sum of PDOP and TDOP, represent the overall accuracy of the system.

Remembering the final covariance matrix of the estimated coordinates and clock offset in Least Square algorithm:

$$C_{\hat{x}\hat{x}} = \begin{bmatrix} \sigma_x^2 & \sigma_{yx} & \sigma_{zx} & \sigma_{tx} \\ \sigma_{xy} & \sigma_y^2 & \dots & \dots \\ \sigma_{xz} & \dots & \sigma_z^2 & \dots \\ \sigma_{xt} & \dots & \dots & \sigma_t^2 \end{bmatrix} = \hat{\sigma}_0^2 N^{-1}$$

Where $N^{-1} = Q_{\hat{x}\hat{x}}$, that is the cofactor matrix, containing the proportionality factors that express the relative accuracy of different estimates. $Q_{\hat{x}\hat{x}}$ in Least Squares process is defined as follow:

$$Q_{\hat{x}\hat{x}} = (A^T Q^{-1} A)^{-1} = \begin{bmatrix} q_x & \dots & \dots & \dots \\ q_{xy} & q_y & \dots & \dots \\ q_{xz} & q_{yz} & q_z & \dots \\ q_{xt} & q_{yt} & q_{zt} & q_t \end{bmatrix}$$

A is the design matrix of the system, containing all the coefficients that multiply the unknowns and has dimension $[m \times 4]$. Q is the cofactor matrix of the observations and has dimensions of $[m \times m]$. In our simulation we decided to put the cofactor matrix Q equal to the identity matrix.

$$A = \begin{bmatrix} e_X^{S^1} & e_Y^{S^1} & e_Z^{S^1} & 1 \\ e_X^{S^2} & e_Y^{S^2} & e_Z^{S^2} & 1 \\ \dots & \dots & \dots & 1 \\ \dots & \dots & \dots & 1 \\ e_X^{S^m} & e_Y^{S^m} & e_Z^{S^m} & 1 \end{bmatrix}$$

Focusing on A, the value of the elements belonging to the first three columns depends only on the geometry of the satellites with respect to the receiver, being $e_j^{S^i}$ the elements of the unitary vector computed as follow for each satellite in LOS.

$$\mathbf{e}_R^{S^i} = \begin{bmatrix} e_X^{S^i} \\ e_Y^{S^i} \\ e_Z^{S^i} \end{bmatrix} = \begin{bmatrix} (\tilde{X}_R - X^{S^i}) \\ (\tilde{Y}_R - Y^{S^i}) \\ (\tilde{Z}_R - Z^{S^i}) \end{bmatrix} * \tilde{\rho}_R^S$$

Where:

- $\tilde{X}_R, \tilde{Y}_R, \tilde{Z}_R$ are the approximate coordinates of the receiver.
- $\tilde{\rho}_R^S$ is the approximate distance between the receiver and the satellite.
- $X^{S^i}, Y^{S^i}, Z^{S^i}$ are the coordinate of the satellite obtained from the precise ephemerides.

In our implementation, measurements of the indexes were computed every 5 minutes over a full day. The positions of the satellites were obtained from the precise ephemerides transmitted by ESA navigation office. For the reading process of the ephemeris we remind to paragraph **??**. The in LOS satellite ware obtained by applying the masking method introduced and discussed in paragraph 3.2.2. The position of the receiver is considered fixed and coincides with the one established for the construction of the building grid and the 5G beacons network.

$$P_R = \begin{bmatrix} 45°28'42.127'' \\ 9°13'45.167'' \\ 160.609m \end{bmatrix}$$

Since the coordinates of the receiver and the satellites are known for each epoch, we can compute the cofactor matrix $Q_{\hat{x}\hat{x}}$ a-priori, even if we don't have any information on observation values, and evaluate PDOD and HDOP indexes as follow:

$$PDOP = \sqrt{q_x + q_y + q_z}$$
$$HDOP = \sqrt{q_x + q_y}$$

DOP values are dimensionless, according to DOP value we can classify the quality of satellites configuration geometry. Reference values for the classification are shown in table 4.1.

Value	Class
$1 - 5$	Good
$5 - 10$	Moderate
$10 - 20$	Fair
> 20	Poor

Table 4.1: Reference values for DOP

4.2 Simulation of 5G beacons

The simulation of the integration of the 5G signal is performed on the four urban scenarios introduced in chapter 3:

1 Street from East to West with buildings 24 meters high.

2 Street from North to South with buildings 24 meters high.

3 Street from East to West with buildings 9 meters high.

4 Street from North to South with buildings 9 meters high.

First step of the process it to introduce in the scenarios the points representing the network of beacons transmitting the 5G signal to the receiver. We assume that the receiver, from it's position, acquires the signal from four beacons, arranged in pairs on the two opposite sides of the road at a distance of 200 meters. The receiver is placed in the exact center with respect to the beacons and all the beacons have a 20 meters height with respect to the ground.

Since the simulation was carried out on two different roads, it was necessary to simulate two different networks: one for the street running from East to West and another for the North to South street.

The procedure applied to compute the exact position of the beacons is the same applied for the realization of the points representing the buildings on the street:

1 Definition of the horizontal coordinates of the beacons in local cartesian (with Up = 0) with respect to the receiver's position.

2 Conversion of the local cartesian in global cartesian (see Appendix A.3)

3 Conversion of global cartesian to geodetic (see Appendix A.1).

4 Setting of the actual height of the beacons as: $h_{ground} + 20m$.

5 Back conversion of the new coordinates from geodetic to global cartesian (see Appendix A.2.

6 Back conversion of global cartesian coordinates to local cartesian with respect to the receiver's position.

Figure 4.2: Map representing the E-W street with the integration of 5G beacons

In figures 4.2 and 4.3 it is possible to see the final configuration for the East-West street and for the North-South street produced via QGIS. Yellow points represent the street side buildings, the blue triangle represents the receiver's position and the green diamonds correspond to the beacons' network.

4.3 Results of DOP analysis

As mentioned before, PDOP value is a geometric index of accuracy of the estimated position; therefore, it indicates the positional precision in space of 3D coordinates. HDOP, on the other hand, indicates the accuracy of the estimate of the receiver's planimetric position, that is typically required for navigation in urban environment. The following paragraph shows the results of the analysis carried out. The following analysis is of purely geometric interest: in the following discussion the problems related to the offset of the receiver with respect to the GNSS and 5G clocks are not considered as we focus on a geometric analysis on the signals configuration. For each scenario, the PDOP and HDOP of each epoch were computed under the following conditions:

- Open sky - GNSS: ideal condition in absence of buildings, all satellites with elevation greater than 0 are in LOS and involved DOP computation.
- Urban canyon - GNSS: presence of buildings in the environment that block the satellites' signal, the number of in LOS satellites in determined with the masking method proposed in paragraph 3.2.2. DOP is computed only with the available satellites.
- Urban canyon - GNSS + 5G: DOP is computed for the hybrid system, which comprehend the the available satellites in LOS and the signal received from the 5G network proposed in previous chapter.

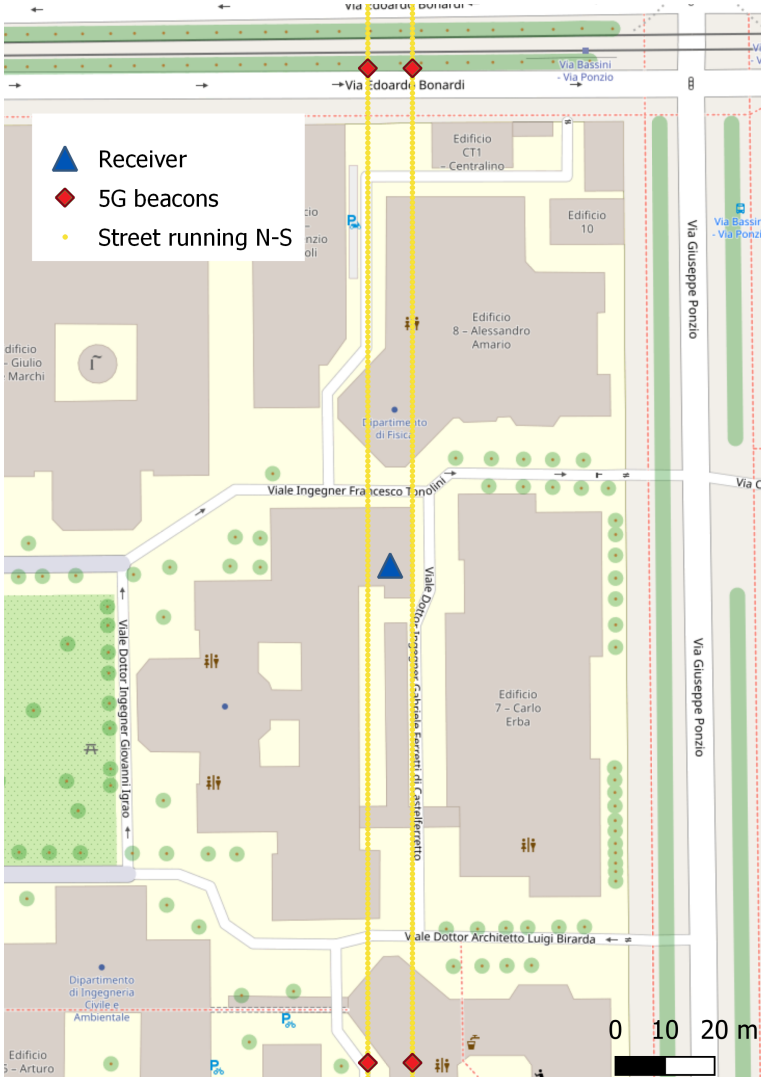

Figure 4.3: Map representing the N-S street with the integration of 5G beacons

4.3.1 Street East-West - buildings height 24m

The first scenario discussed is one of the most critical: the receiver is placed in the center of a 9 meter wide road running from East to West, overlooked by buildings 24 meters high. As showed in figure 3.6, the number of in LOS satellites is significantly reduced with respect to open sky condition.

PDOP analysis

Figure 4.4 shows the comparison of the PDOP daily behaviour in case of open sky (blue line) and in urban canyon with the satellites masked by the buildings (red line). In ideal conditions of open sky, the graph of the PDOP can be considered constant with minimal variations with respect to the average value of 0.5, in fact we have a standard deviation of 0.02.
On the contrary, if we consider the presence of buildings, the PDOP rises considerably, obtaining an average value of about 8.8, with peaks reaching $50 - 60$ and a standard deviation of 7.8.

In addition, for the masked configuration of satellites, we are able to compute the PDOP only for a smaller number of epochs with respect to the total. In fact, considering that the whole time series includes 289 epochs, PDOP values are available only for 225 epochs, this means that in the remaining 64 epochs we do not have a sufficient number of satellites in LOS to be able to evaluate the PDOP and, consequently, to perform precise Point Positioning in single epoch. Moreover, in 52 epochs ($\approx 23\%$ of available epochs) the PDOP reaches a value greater than 10, among these 14 epochs have a value greater than 20, classifying the configuration of the satellites as 'poor', referring to the classification shown in table 4.1.

From the results obtained from the first comparison it can be deduced that the precision of positioning in an urban environment is strongly compromised. The simulation goes on with the evaluation of the PDOP in the same scenario with the integration of the four 5G signals received from the network of beacons introduced in paragraph 4.2. The results are showed in figure 4.5, which compares the PDOPs obtained from the hybrid configuration GNSS+5G (green line) with those obtained from the masked GNSS configuration discussed above (red line).

It is evident that in the hybrid configuration the PDOP behaviour remains on average similar to GNSS-only, however the previously reached peaks are considerably dampened. A sharp decrease in the maximum values leads to a decrease in the average value which falls to 5.9. In addition, the introduction

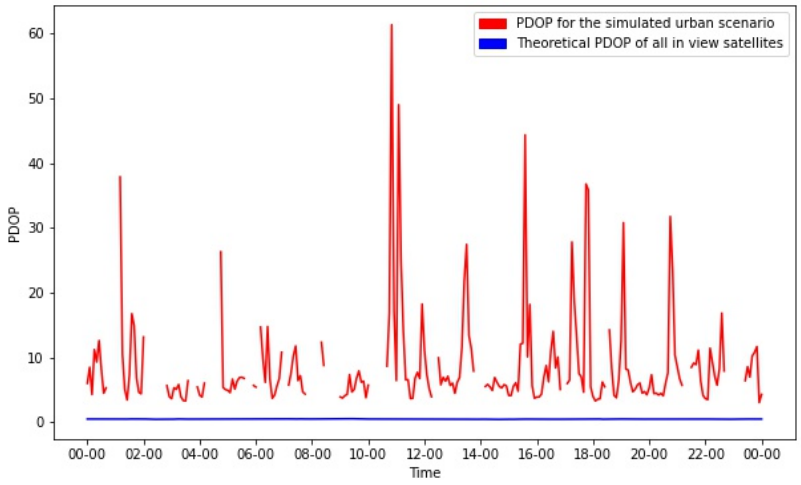

Figure 4.4: PDOP daily graph for East-West street with buildings 24m high, comparing open sky and GNSS-only configurations (May 26th 2021)

of 4 new observation equations in the system makes positioning available for all the epochs.

In hybrid configuration we observe that the maximum value reached by the PDOP is 11.4, and it exceed the reference value of 10.0 only 17 epochs ($\approx 5.8\%$ of analyzed epochs). We can therefore catalog the geometry of the received signals as 'good' or 'moderate', according to the reference value reported in Table 4.1.

Table 4.2 provides a summary of what has been said so far, showing the number of epochs in which a certain class of PDOP is reached and the number of epochs in which it cannot be computed for both the GNSS only and 5G + GNSS configuration.

	Good	Moderate	Fair	Good	NO PDOP
GNSS only	58	115	38	14	64
GNSS+5G	117	155	17	0	0

Table 4.2: Number of epochs per PDOP class over a time series of 289 epochs, East-West street with buildings 24m high on May 26th 2021

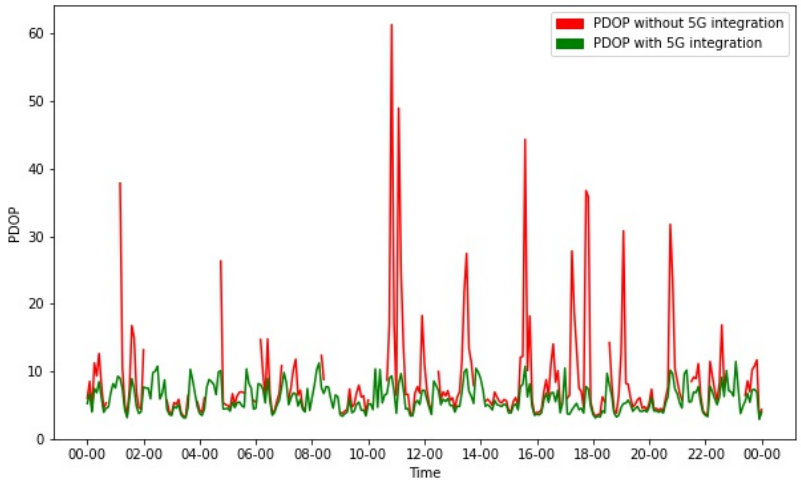

Figure 4.5: PDOP daily graph for East-West street with buildings 24m high, comparing GNSS-only and GNSS+5G configurations (May 26th 2021)

HDOP analysis

What emerges from the HDOP computation confirms the above-mentioned results of the PDOP. HDOP was evaluated for all the available epochs, the obtained values do not differ significantly with respect to PDOP. Figures 4.6, 4.7 show HDOP behaviour throughout the day, the first shows the comparison between open sky and only GNSS configurations, the second compares only GNSS and hybrid GNSS+5G configurations. The plots are very similar to those obtained for PDOP, from these results we deduce that the error related to the Up component of the estimate is not relevant and negligible with respect to the errors related to horizontal positioning.
In addition, from a further analysis on the computed cofactor matrices, it emerges that the biggest error is the one that corresponds to the North component, several orders of magnitude greater than the one related to the East component. This is due to the fact that the street represented in the scenario is oriented from East to West ideally infinitely extended and there are no sufficient information to estimate correctly the component orthogonal to the street direction.

Tables 4.3 and 4.4 report as example the computed values of PDOP, HDOP and the respective cofactor matrix at $00:15:00$ for the configuration of GNSS-only and 5G+GNSS, the error in the cofactor matrix related to the

	Only-GNSS
PDOP	11.2
HDOP	11.1
Cofactor matrix	$\begin{bmatrix} 1 & 3 & -1 \\ 3 & \mathbf{123} & -15 \\ -1 & -15 & 2 \end{bmatrix}$

Table 4.3: PDOP, HDOP and cofactor matrix for only-GNSS configuration at $00:15:00$ *- EW 24m*

	GNSS+5G
PDOP	7.4
HDOP	7.3
Cofactor matrix	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \mathbf{54} & -6 \\ 0 & -6 & 1 \end{bmatrix}$

Table 4.4: PDOP, HDOP and cofactor matrix for hybrid GNSS+5G configuration at $00:15:00$ *- EW 24m*

North component is highlighted in red and it is evident that the crucial factor in the computation of PDOP and HDOP.

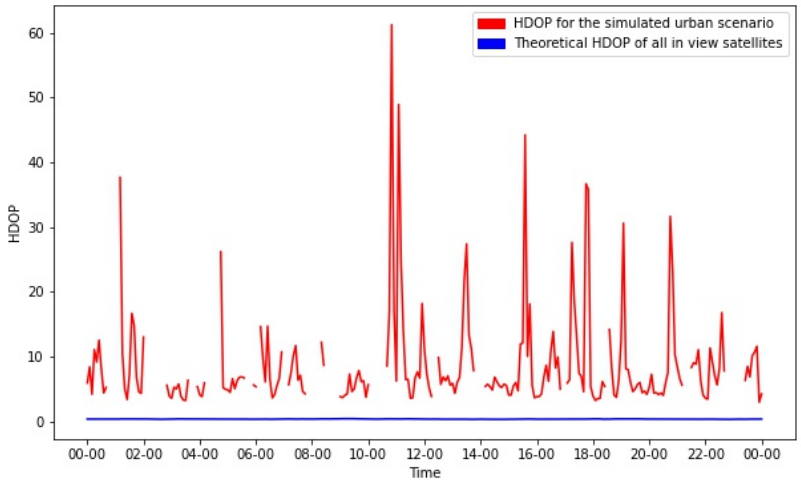

Figure 4.6: HDOP daily graph for East-West street with buildings 24m high, comparing open sky and GNSS-only configurations (May 26th 2021)

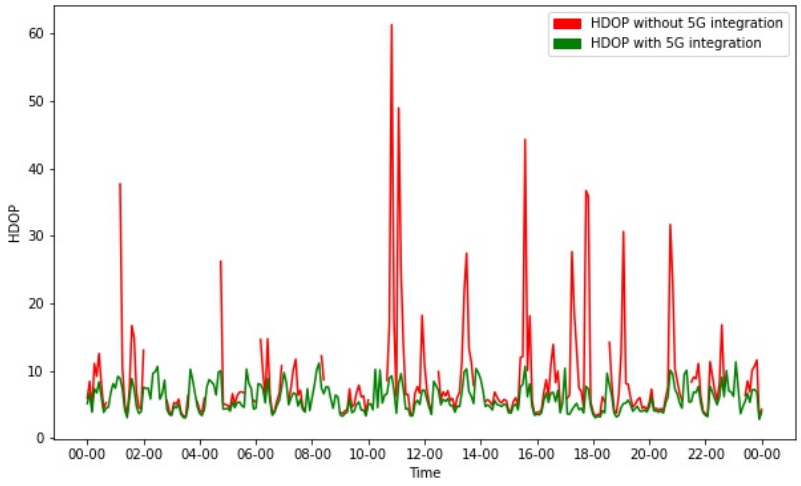

Figure 4.7: HDOP daily graph for East-West street with buildings 24m high, comparing GNSS-only and GNSS+5G configurations (May 26th 2021)

4.3.2 Street East-West - buildings height 9m

The second scenario discussed represents a 9 meter wide road running from East to West with the height of the buildings set to 9 meters above the ground. The setting of the scenario is the same as the previous one, but

the change in the height of the buildings involves a radical change in the obtained results for PDOP and HDOP evaluation. The scenario in question therefore simulates an average built-up urban environment, which represents an intermediate situation between the urban canyon and the open-field.

Already from the screening carried out on satellites, analyzed in chapter 3.2.2 with reference to graph 3.5 and table 3.1, it is evident that the number of satellites in LOS, which varies from a minimum of 8 to a maximum of 20, is high enough to allow a good point positioning for all the epochs in the time series.
The analysis, therefore, focuses on the quality of the geometric configuration of the satellites and on the possible improvement in signal reception brought by the introduction of a 5G network.

PDOP analysis

Figure 4.8 shows the behaviour of the PDOP computed for the open-field (blue line) and only-GNSS masked (red line) configurations. It is evident that the presence of the 9-meter buildings minimally affects the quality of the PDOP. In fact, in the configuration of GNSS only, the average value of the PDOP is 1.7, with a maximum value of 2.8 and standard deviation equal to 0.3.
The obtained results, which can be considered excellent, allow to classify the satellite's configuration as 'good' (best obtainable result in the proposed scale) for the whole time range, referring to the standard values for DOP classification in Table 4.1. In addition, considering that for the ideal condition of open-field (with all the satellites in LOS) we obtain a mean value of 0.5 for the PDOP, the obtained results for the proposed scenario are very close to the ideal ones.

Following the proposed working scheme, we proceeded with the evaluation of the PDOP for the hybrid system. Figure 4.9 shows the comparison between the PDOP behaviour of the only-GNSS (red line) and hybrid GNSS+5G (green line) configurations. We observe an overall minimum decrease in the PDOP value (the average value of the decrease is 0.1285), which does not substantially change the result.

Table 4.5 shows the number of epochs in which a certain class of PDOP is reached and the number of epochs in which it cannot be computed for both the GNSS only and 5G + GNSS configuration.

	Good	Moderate	Fair	Good	NO PDOP
GNSS only	289	0	0	0	0
GNSS+5G	289	0	0	0	0

Table 4.5: Number of epochs per PDOP class over a time series of 289 epochs, East-West street with buildings 9m high on May 26th 2021

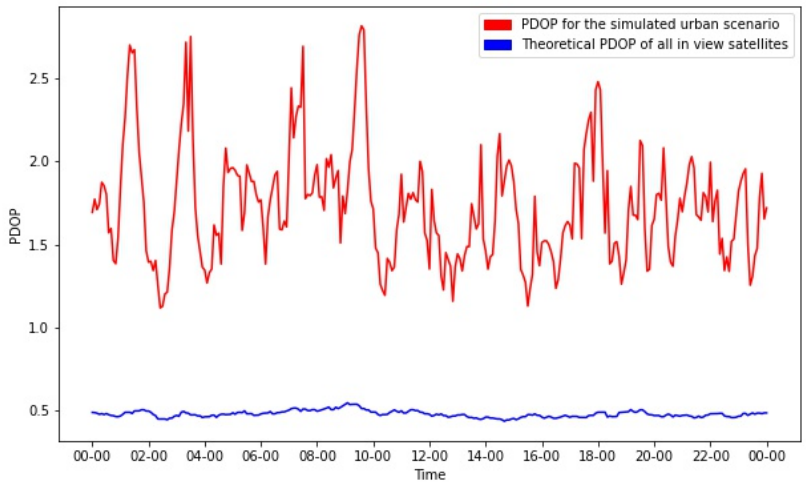

Figure 4.8: PDOP daily graph for East-West street with buildings 9m high, comparing open sky and GNSS-only configurations (May 26th 2021)

From this we can deduce that the introduction of the 5G signal in an environment with sufficient satellite coverage does not significantly improves the geometrical scenario.

HDOP analysis

For the sake of completeness, the results of the HDOP computation for the entire time interval are proposed in the following paragraph. The HDOP daily behaviour, shown in figures 4.10 and 4.11, provide further confirmation of the stability and efficiency of the satellite constellations. In fact, considering the GNSS-only configuration (represented with the red line) we have an average value of 1.685 for the HDOP, that further decrease in the Hybrid GNSS+5G configuration to 1.556.

As seen in the previous cases, the introduction of 5G beacons allows to reduce the peaks connected to bad satellite arrangements, however, since

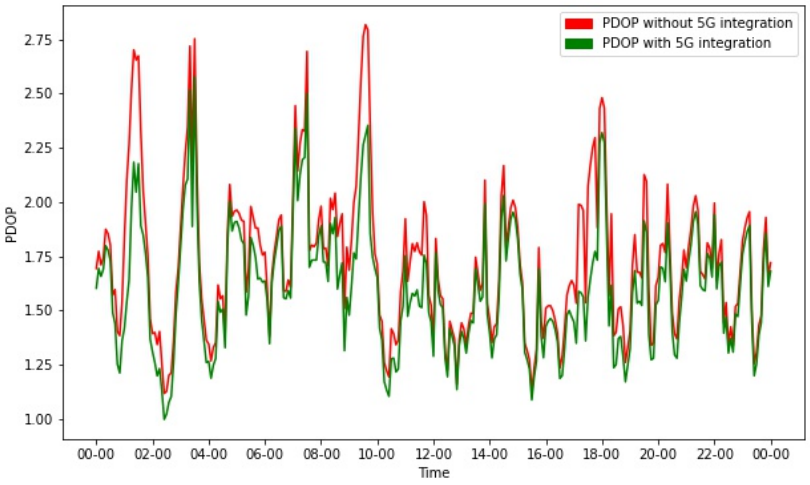

Figure 4.9: PDOP daily graph for East-West street with buildings 9m high, comparing GNSS-only and GNSS+5G configurations (May 26th 2021)

the results obtained from the GNSS-only configuration are already optimal, the inclusion of 5G networks in a medium urbanized environment turns out to be useless.

4.3.3 Street North-South - buildings height 24m

Then, we move on to the examination of the scenario which, from the results obtained, turned out to be the most problematic: the receiver is placed in the center of a 9 meter wide road running from East to West for 20 km, overlooked by buildings 24 meters high.

Starting from the identification of satellites in LOS described in paragraph 3.2.2, from the results obtained shown in Figure 3.8, we observe that for many epochs (157 over 289, $\simeq 54\%$ of the total number of analyzed epochs) the number of in-view satellites falls below the threshold value of 4 (minimum number of observations required for Point Positioning in single epoch). In particular, the average number of in-sight satellites is 3.5. Therefore, in this particular road configuration, representative of the urban canyon, it is not possible to perform Point Positioning in single epoch with continuity.

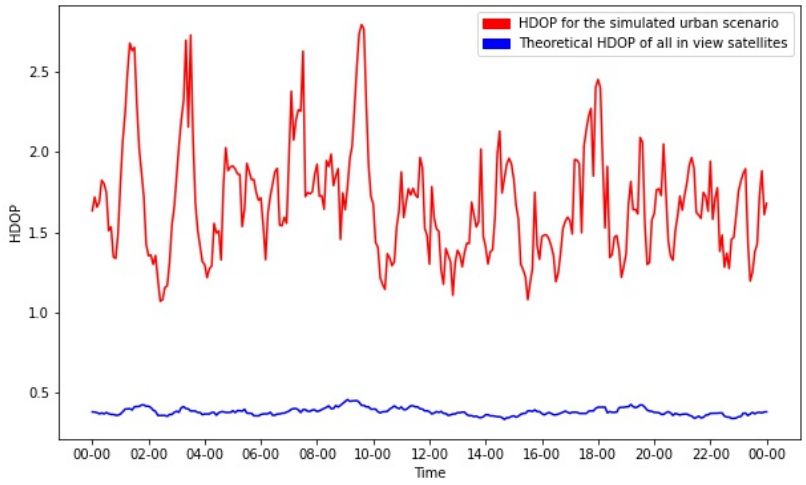

Figure 4.10: HDOP daily graph for East-West street with buildings 9m high, comparing open sky and GNSS-only configurations (May 26th 2021)

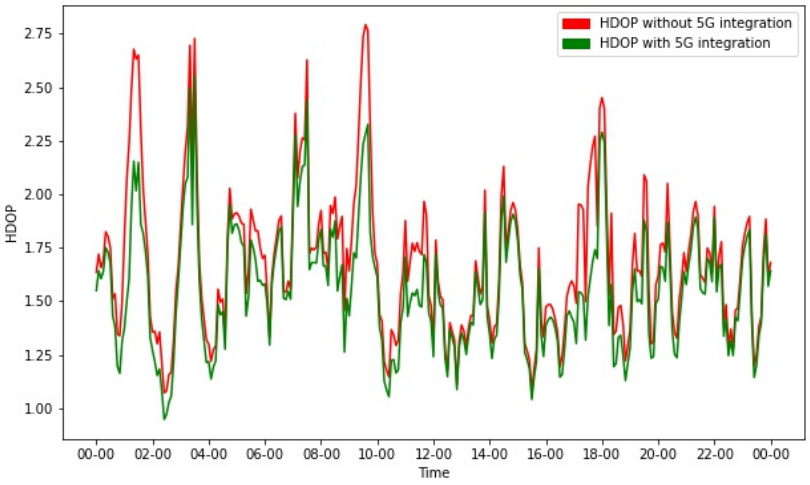

Figure 4.11: HDOP daily graph for East-West street with buildings 9m high, comparing GNSS-only and GNSS+5G configurations (May 26th 2021)

PDOP analysis

Despite the problems that already emerged on the visible number of satellites, we proceeded with the computation of the PDOP for the whole time

range. As expected, it was possible to evaluate the PDOP only on the epochs with a sufficient number of in LOS satellites (132 over 289, $\simeq 46\%$).

Figure 4.12 shows the PDOP's behaviour of the masked configuration of satellites (red line), the average is 9.2 for the available values, PDOP ranges between a minimum of 3.2 and a maximum of 44.8 with a standard deviation for the distribution equal to 7.4. From the graph it is evident that there is no continuity in the computation of the PDOP, in addition, moreover, 31 epochs over the 132 available ($\simeq 23\%$) have a PDOP value greater than 10, therefore a classification equal to fail or poor, according to classification classes reported in Table 4.1. We remind to table 4.6 for a synthetic representation of the number of epochs belonging to each class.

The integration of the satellite signal with the 5G network is therefore an obligatory step, to allow not only to improve the geometry of the available signals, but, above all, to have in each epoch a sufficient number of signals to perform Point Positioning.

The results of PDOP evaluation for the Hybrid configuration are shown in Figure 4.13. The addition of the 4 signals coming from the 5G beacons present in the network allows to evaluate the PDOP in all the epochs and consequently to perform Point Positioning and to significantly reduce the value of the PDOP in the most critical points. The average value of the PDOP for the hybrid configuration is 7.2 ranging between a minimum of 2.968 and a maximum of 11.6. The new distribution has a standard deviation of 2.5, which makes us understand that the variability of the values also decreases and they are all closer to the average value. The threshold value of 10 is exceeded only in 10 epochs: let consider that these correspond to the epochs in which we were not previously able to compute the PDOP. The improvements for each class are shown in table 4.6.

Therefore, despite the PDOP value does not allow us to evaluate the configuration as 'good' in all epochs, the geometry significantly improves.

	Good	Moderate	Fair	Good	NO PDOP
GNSS only	27	74	20	11	157
GNSS+5G	256	23	10	0	0

Table 4.6: Number of epochs per PDOP class over a time series of 289 epochs, North-South street with buildings 24m high on May 26th 2021

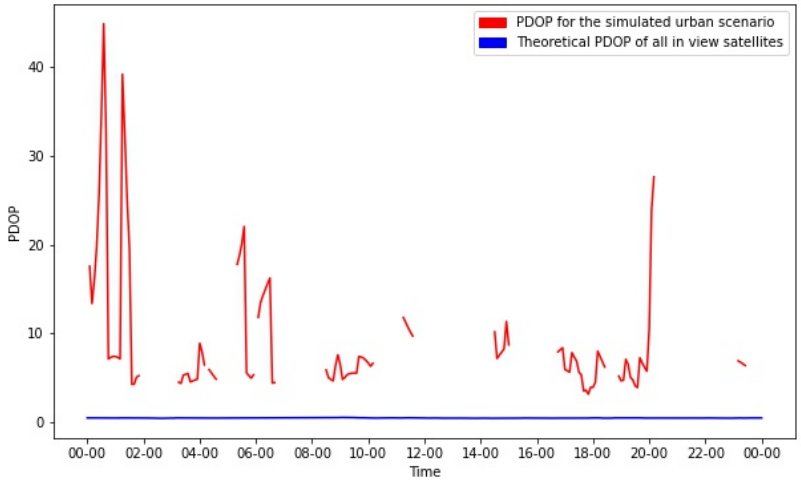

Figure 4.12: PDOP daily graph for North-South street with buildings 24m high, comparing open sky and GNSS-only configurations (May 26th 2021)

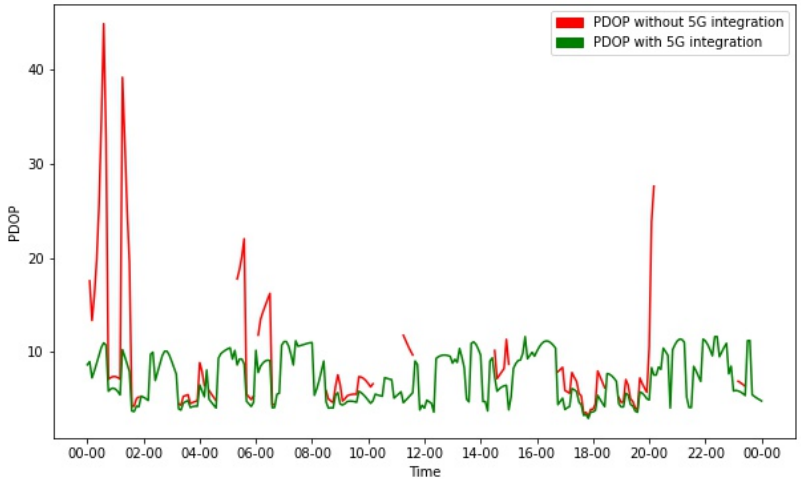

Figure 4.13: PDOP daily graph for North-South street with buildings 24m high, comparing GNSS-only and GNSS+5G configurations (May 26th 2021)

HDOP analysis

The next step consists in computing the HDOP for the available epochs, what emerges from the obtained results confirms what has been said so far: the HDOP values do not differ significantly with respect to PDOP, as seen for the street running from East to West with buildings' height set to 24m. Figures 4.14, 4.14 show HDOP behaviour throughout the day, the first shows the comparison between open sky and only GNSS configurations, the second compares only GNSS and hybrid GNSS+5G configurations. The plots are very similar to those obtained for PDOP: from these results we deduce that the error related to the Up component of the estimate is not relevant and negligible with respect to the errors related to 2D position.

As expected, from a check on the cofactor matrices, in this case the biggest error is the one corresponding to the East component, while the error of the North component is negligible, due to the fact that the street is oriented from North to South and there are no sufficient information to estimate correctly the East component.

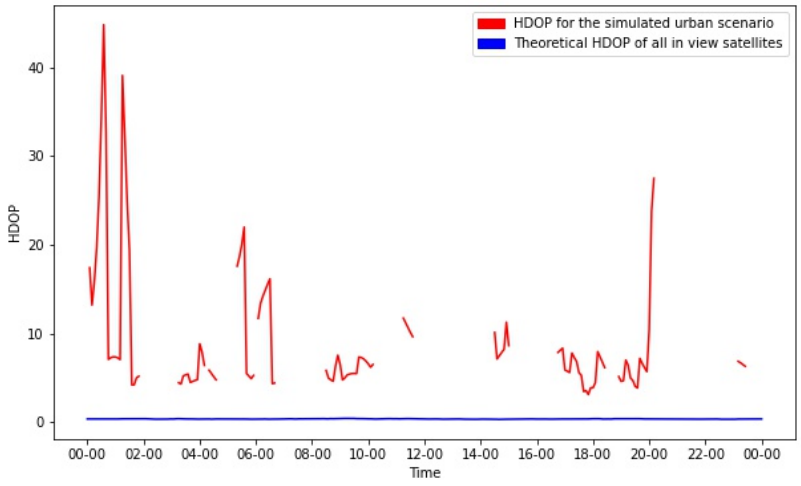

Figure 4.14: HDOP daily graph for North-South street with buildings 24m high, comparing open sky and GNSS-only configurations (May 26th 2021)

As shown for the first scenario, Tables 4.7, 4.8 report an example of PDOP, HDOP and cofactor matrices evaluated respectively for the GNSS-only and Hybrid GNSS+5G configurations at the same time 00 : 15 : 00.

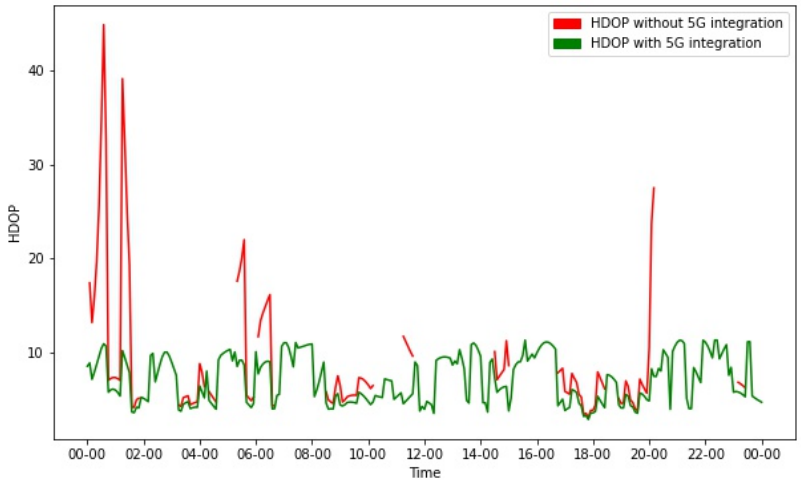

Figure 4.15: PDOP daily graph for North-South street with buildings 24m high, comparing GNSS-only and GNSS+5G configurations (May 26th 2021)

	Only-GNSS
PDOP	16.0
HDOP	15.6
Cofactor matrix	$\begin{bmatrix} \mathbf{387} & 16 & 32 \\ 165 & 1 & 2 \\ 32 & 2 & 3 \end{bmatrix}$

Table 4.7: PDOP, HDOP and cofactor matrix for only-GNSS configuration at $00:15:00$ *- NS 24m*

4.3.4 Street North-South - buildings height 9m

The last scenario proposed represents a 9 meter wide street running from North to South, with buildings' height set to 9 meter. As seen previously in paragraph 4.3.2, this scenario refers to a medium urbanized environment, where the presence of buildings blocks the signal of some satellites in sight, but does not compromise the reception of a sufficient number of signals in order to be able to perform Point Positioning.

Referring to the visibility graph in Figure 3.7, we observe that the average number of satellites in LOS is 8, ranging between 4 and 16, therefore, it is always possible to perform Point Positioning and to compute DOP indexes for all the epochs of the time range.

	GNSS+5G
PDOP	7.4
HDOP	7.4
Cofactor matrix	$\begin{bmatrix} \mathbf{63} & 1 & 5 \\ 1 & 0 & 0 \\ 5 & 0 & 1 \end{bmatrix}$

Table 4.8: PDOP, HDOP and cofactor matrix for hybrid GNSS+5G configuration at $00:15:00$ *- NS 24m*

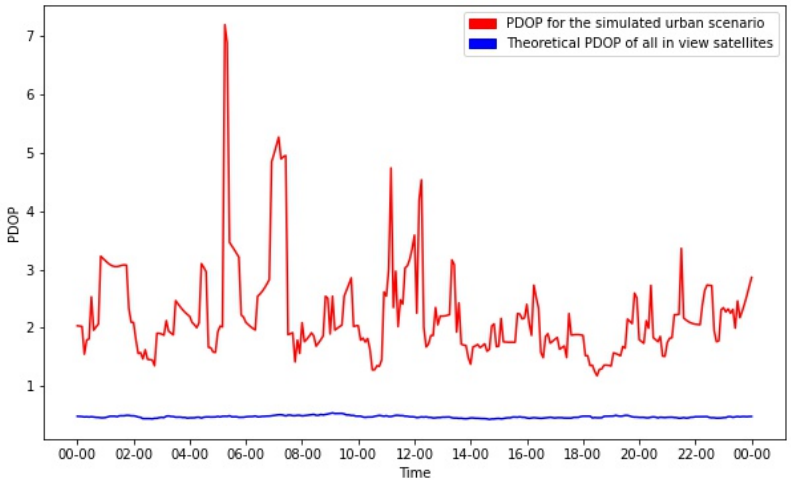

Figure 4.16: PDOP daily graph for North-South street with buildings height 9m in open sky and only GNSS masked configurations

PDOP analysis

Figures 4.16 and 4.17 show a comparison between the PDOP computed for all the satellites in-view in open-field conditions (blue line) with the PDOP computed with GNSS-only configuration (red line) and Hybrid GNSS+5G configuration (green line). Already from a first analysis on the results obtained from the GNSS-only configuration, we observe an average value of 2.2 for the PDOP, which is considered an excellent results. In addition, the PDOP value ranges between a minimum of 1.2 and a maximum of 7.2, all the results are below the threshold value of 10 and allow to classify the configuration geometry as 'good' for most of the epochs. See the resume of the classification in table 4.9.

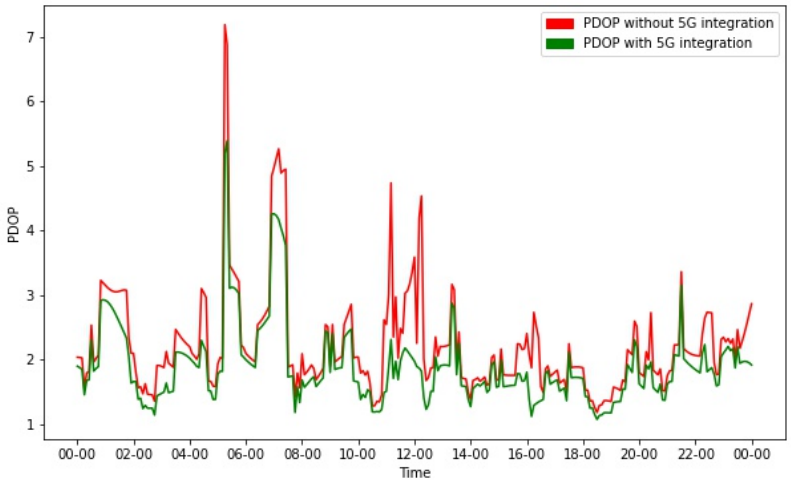

Figure 4.17: PDOP daily graph for North-South street with buildings height 9m only GNSS masked configurations and hybrid GNSS+5G configuration

The integration of the four 5G signals in the configuration implies a gen-

	Good	Moderate	Fair	Good	NO PDOP
GNSS only	285	4	0	0	0
GNSS+5G	289	0	0	0	0

Table 4.9: Number of epochs per PDOP class over a time series of 289 epochs, North-South with buildings 9m high on May 26th 2021

eral reduction of the PDOP value for all the epochs, which becomes more significant where PDOP values were higher. With the Hybrid configuration PDOP's average values decreases to 1.9, but, in general, the obtained reduction is of little relevance compared to the improvements we can observe in densely urbanized environments such as urban canyons, represented by the first and third scenarios.

HDOP analysis

The results obtained from the computation of the HDOP and from the comparison with the relative PDOP, confirm the results previously discussed and obtained in the road configuration of buildings of 9 m height facing the street running from East to West.

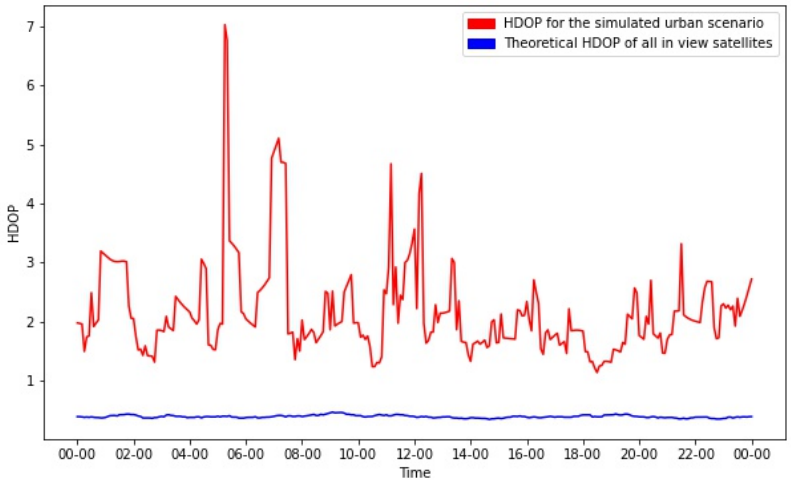

Figure 4.18: HDOP daily graph for North-South street with buildings height 9m in open sky and only GNSS masked configurations

The average value of the HDOP drops from 2.2 in GNSS-only configuration to 1.8 in the Hybrid GNSS+5G configuration. The decrease obtained is very insignificant, especially considering that we are already working with good index values.

From the analysis of the covariance matrices it emerges that, also in this case, the prevailing error is, as expected, the one linked to the East component, that contributes the most in the evaluation of the final index value.

4.4 Considerations on clock offsets

The results analyzed above focus on the evaluation of the geometry, since the indices considered (PDOP and HDOP) highlight the influence of the geometry of the signals on the three-dimensional and two-dimensional positioning error.

In particular, it should be emphasized that the discussion does not consider the contribution to the system ill conditioning introduced by the estimate of the receiver clock offset.

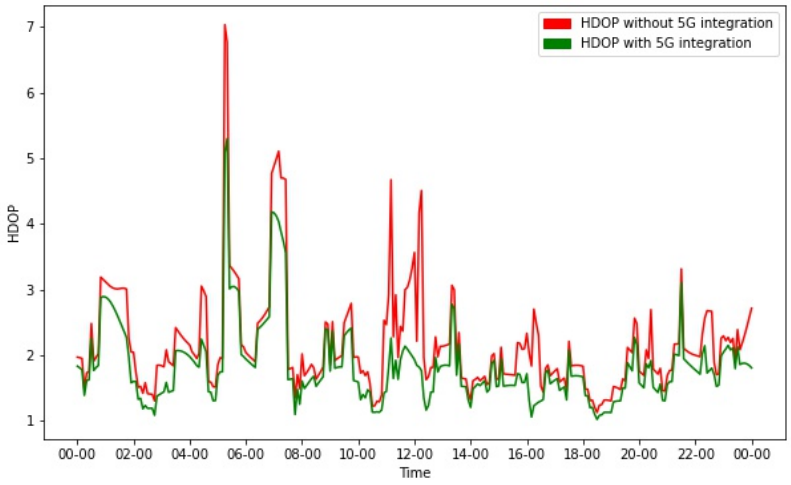

Figure 4.19: HDOP daily graph for North-South street with buildings height 9m only GNSS masked configurations and hybrid GNSS+5G configuration

When performing point positioning, as seen in paragraph 2.4, the number of unknowns to estimate depends on several factors. To the three positional unknowns, all the unknowns clock offsets of the receiver must be added: for each constellation of satellites used for the positioning process (GPS, Glonass, BeiDou, Galileo), it is necessary to estimate the time difference between the clock of the receiver with respect to the time scale of the specific constellation. Therefore, as seen in paragraph 2.4, if we only use the GPS constellation we will have a total of 3 + 1 unknowns. If instead we use all the available constellations, it is necessary to introduce 3 + 4 unknowns: 3 for the position and 4 corresponding to the clock offsets of the receiver with respect to the 4 constellations.

However, with at a certain accuracy level, we can assume that we know the offsets between the different constellations. This implies that the number of unknown offsets is reduced to one, since the relationship with the others is known.

Finally, it should be noted that the implementation of the Hybrid GNSS + 5G solution involves the introduction of an additional clock offset. In fact, working on the hypothesis that the clocks of the beacons of the 5G network are reciprocally synchronized, we have at least the offset of the re-

ceiver to be estimate.

The introduction of new unknowns to be estimated undoubtedly increases the error contained in the covariance matrix, affecting positioning performance. However, by increasing the density of beacons in urban canyons, where the reception of signals is strongly compromised, it is possible to overcome the problem introduced by the increase in unknowns. In particular, the problem related to clock offsets will be subject of future research.

Chapter 5

Conclusions and future perspectives

Regarding the development of the software for GNSS positioning (chapter 2), the results obtained by applying Least Squares process are satisfactory, exactly in line with the expectations. At present, only GPS processing has been implemented, but in few months we will have a complete GNSS software. Indeed, this work was undertaken with the aim of creating an open source Python library for GNSS positioning. The work will proceed with the implementation of the orbits computation of the other constellations (Galileo, Glonass and BeiDou), in order to integrate the observations in the Least Squares model developed for the estimation of the unknowns.

Regarding the urban environment simulation and the integration of 5G with GNSS for positioning, it is appropriate, in light of the obtained results, to differentiate according to the considered scenario. Remembering that the 4 proposed scenarios are infinitely extended roads with East-West and North-South direction with buildings 9 or 24 meters high, it emerges that the most critical results are those related to the urban canyons, represented by roads with buildings 24 meters high: the availability of satellites is significantly reduced, affecting the positioning performance. In fact, in many epochs the number of satellites does not satisfy the minimum redundancy requirement (smaller than 4), reaching up to 0 satellites in LOS and therefore not making positioning possible at all for. In particular, this happens for 22% of the epochs in East-West and for 53% in North-South. For both North-South and East-West roads with tall buildings, the computed PDOP and HDOP indices have unstable behaviour in time-series: they vary in the different categories ('good', 'fair', 'moderate' and 'bad'). For instance, 23% of the

PDOPs that can be computed, both for the North-South and East-West configurations, are in 'bad' and 'moderate' classes. The situation is therefore extremely critical and it is impossible to meet the demands of precise GNSS positioning. The integration of the GNSS signal with a 5G network turns out to be a possible solution, as it guarantees positioning and, consequently, the computation of the indices, allowing to obtain DOP indices, which mainly belongs to 'good' and 'fair' classes.

In the scenarios with buildings 9 meters high, which correspond to residential streets, the geometry and visibility of the satellites improves and always the positioning is possible, as we always have a sufficient number of satellites in LOS. The computation of PDOP and HDOP indices is in fact satisfactory for each epoch and the introduction of the 5G network does not entail decisive improvements. All satellites configurations are classified as 'good'.

This work aims to continue towards a more complete analysis of the actual possibility of integration of GNSS and 5G, first of all investigating the problems related to the clock offsets, both of the satellites and of the 5G network, focusing on the estimation of the time unknowns and on the clock synchronization of a 5G network.

Appendix A

Transformation between different Reference System

The objective of this appendix is to include all the implemented formulas necessary for the transformation of the coordinates in the different reference systems, required in the software and simulation computeation.

A.1 From Global Cartesian To Geodetic

$$\text{Starting point: } P_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \text{ m}$$

Constant definition:

a	6378137m
e	0.0818191908426215

b	$a(\sqrt{1-e^2})$
e_b^2	$\frac{a^2-b^2}{b^2}$
r	$\sqrt{X_R^2+Y_R^2}$
Longitude	$\arctan \frac{Y_i}{X_i}$
ψ	$\arctan \frac{Z_i}{r\sqrt{1-e^2}}$
Latitude	$\arctan \frac{Z_i+e_b^2*b*(\sin\psi)^3}{r-e^2a(\cos\psi)^2}$
N	$\frac{a}{\sqrt{1-e^2(\sin Latitude)^2}}$
h	$\frac{r}{\cos Latitude} - N$

$$P_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \rightarrow \begin{bmatrix} \varphi \\ \lambda \\ h \end{bmatrix}$$

A.2 From Global Geodetic to Global Cartesian

$$\text{Starting point: } P_X = \begin{bmatrix} \varphi \\ \lambda \\ h \end{bmatrix} \text{ m}$$

Constant definition:

a	6378137m
e	0.0818191908426215

N	$\frac{a}{\sqrt{e^2(\sin Latitude)^2}}$
X_i	$(N+H)\cos Latitude * \cos Longitude$
Y_i	$(N+H)\cos Latitude * \sin Longitude$
Z_i	$(N(1-e^2)+h)\sin Latitude$

$$P_i = \begin{bmatrix} Latitude_i{}^\circ \\ Longitude_i{}^\circ \\ H_i m \end{bmatrix} \rightarrow \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

A.3 Global Cartesian to Local Cartesian

Given an origin point P_0 for the interest area, the coordinates of the other points P_i are expressed as East, North, Up (ellipsoidal) with respect to the origin.

$$\text{Origin: } P_0 = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix}$$

$$P_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

First steep consist in the transformation of P_0 in Global Geodetic (see A.1), to get the values of latitude and longitude in degrees of the origin(lon_0, lat_0), required for the computation of the rotation matrix R.

$$R = \begin{bmatrix} -\sin lon_0 & -\cos lon_0 & 0 \\ -\sin lat_0 \cos lon_0 & -\sin lat_0 \sin lon_0 & \cos lat_0 \\ \cos lat_0 \cos lon_0 & \cos lat_0 \sin lon_0 & \sin lat_0 \end{bmatrix}$$

$$\begin{bmatrix} E_i \\ N_i \\ U_i \end{bmatrix} = R(P_i - P_0)$$

Appendix A. Transformation between different Reference System

List of Tables

Bibliography

[1] W. S. H. M. W. Ahmad, N. A. M. Radzi, F. S. Samidi, A. Ismail, F. Abdullah, M. Z. Jamaludin, and M. N. Zakaria. 5g technology: Towards dynamic spectrum sharing using cognitive radio networks. *IEEE Access*, 8:14460–14488, 2020.

[2] Dmitry Tatarnikov Ph.D Alfred Leick Ph.D, Lev Rapoport Ph.D. *GPS Satellite Surveying, Fourth Edition.* John Wiley and Sons, Inc, 2015.

[3] Claudina Tiznado Anthony Flores. Is-gps-200l. 14/5/2020.

[4] Ludovico Biagi. *I fondamenti del GPS.* Geomatics Workbooks, 2006.

[5] R Chengqi. Development of the beidou navigation satellite system. In *Global navigation satellite systems. Report of the Joint Workshop of the National Academy of Engineering and the Chinese Academy of Engineering. Washington, DC*, 2012.

[6] José A. Del Peral-Rosado, Jani Saloranta, Giuseppe Destino, José A. López-Salcedo, and Gonzalo Seco-Granados. Methodology for simulating 5g and gnss high-accuracy positioning. *Sensors*, 18(10), 2018.

[7] Giuseppe Destino, Jani Saloranta, Gonzalo Seco-Granados, and Henk Wymeersch. Performance analysis of hybrid 5g-gnss localization. pages 8–12, 2018.

[8] GMV. Galileo general introduction. 2011.

[9] GMV. Galileo performances. 2011.

[10] GMV. Glonass general introduction. 2011.

[11] GMV. Gps general introduction. 2011.

[12] Audrey Guilloton, Anne-Christine Escher, and Damien Koenig. Multipath study on the airport surface. pages 355–365, 04 2012.

[13] Marcus Obst Julia Brebler. Gnss positioning in non-line-of-sight context – a survey for technological innovation. *ASTES Journal*, 2(3):722–731, 2017.

[14] Tomislav Kos, Ivan Markezic, and Josip Pokrajcic. Effects of multipath reception on gps positioning performance. pages 399–402, 2010.

[15] Rui Li, Shuaiyong Zheng, Ershen Wang, Jinping Chen, Shaojun Feng, Dun Wang, and Liwen Dai. Advances in beidou navigation satellite system (bds) and satellite navigation augmentation technologies. *Satellite Navigation*, 1(1):1–23, 2020.

[16] J.A Avila Rodriguez. Galileo signal plan. 2011.

[17] Jani Saloranta and Giuseppe Destino. Reconfiguration of 5g radio interface for positioning and communication. pages 898–902, 2017.