Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Design and Implementation of a GNSS Simulator Supporting Advanced Simulations of Mobile Robot Motion in Outdoor Environments

Bc. Václav Truhlařík

Supervisor: Mgr. Martin Pecka, PhD.
May 2024

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Truhlařík Václav** Personal ID number: **492376**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Design and Implementation of a GNSS Simulator Supporting Advanced Simulations of Mobile Robot Motion in Outdoor Environments

Master’s thesis title in Czech:

Návrh a implementace simulátoru systému satelitní navigace pro pokročilé simulace pohybu mobilních robotů v terénu

Guidelines:

The goal of the thesis is to create a practical advanced simulator of GNSS receivers that could be used in connection with Gazebo or Gazebo Classic simulators and ROS robotics framework.
The student should first review existing approaches to this problem and design a suitable API that would allow practical configuration of simulated experiments and useful output data.
Further, the student should review existing sources of satellite orbits, atmosphere parameters and other required data for the simulation and select those sources that would cover all required inputs for the simulation (if possible; otherwise usage of generative models is also allowed).
The student should implement a basic signal propagation model taking into account satellite orbits and their uncertainties, atmosphere effects and basic RF receiver influences. Usage of publicly available libraries for the computations is expected in this stage.
Next, the student should implement a method that simulates signal obstruction by buildings, objects, obstacles or plants (at least direct visibility, but simulation of multipath effects would be welcome).This method should utilize a generic description of a simulated world to get information about the obstacles.
Finally, the student should experimentally verify quality of the simulated results by comparing them with data captured by a real GNSS receiver on a scene that can also be modeled in the Gazebo simulator. If there are multiple sources of orbit data or other important parameters, the experimental validation should also examine the influence of individual sources to the resulting quality.

Bibliography / sources:

[1] Teunissen, P. J., & Montenbruck, O. (Eds.). (2017). Springer handbook of global navigation satellite systems (Vol. 10, pp. 978-3). Cham, Switzerland: Springer International Publishing.
[2] https://g ssc.esa.int/navipedia
[3] Open-source GNSS computation libraries: https://github.com/tomojitakasu/RTKLIB , https://github.com/sgl-ut/gnsstk , https://github.com/sgl-ut/gnsstk-apps , https://github.com/commaai/laika , https://geoscienceaustralia.github.io/ginan , https://github.com/ctu-vras/gnss-info/
[4] Public sources of relevant data: https://celestrak.org/NORAD/elements/ , https://files.igs.org/pub/station/general/igs_satellite_metadata.snx , https://satdb.ethz.ch , http://ftp.aiub.unibe.ch/CODE_MGEX/CODE/2023/
[5] Septentrio Knowledge Base (https://customersupport.septentrio.com/s/topiccatalog)

Name and workplace of master’s thesis supervisor:

Mgr. Martin Pecka, Ph.D. Vision for Robotics and Autonomous Systems FEE

Name and workplace of second master’s thesis supervisor or consultant:

Ing. Václav Navrátil, Ph.D. Department of Radioelectronics FEE

Date of master’s thesis assignment: **02.02.2024** Deadline for master's thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

Mgr. Martin Pecka, Ph.D. Supervisor’s signature	prof. Dr. Ing. Jan Kybic Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Acknowledgements

My thanks belong to Mgr. Martin Pecka, PhD. for professional guidance, patience and willingness he provided during work on this thesis. Additionally, I would like to thank Ing. Václav Navrátil, Ph.D. for specialized assistance in satellite navigation issues.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses. I have used ChatGPT as a programming advisory for general C++ questions.

Prague, date 24.05.2024 Václav Truhlařík

Abstract

This thesis is centered around newly proposed GNSS simulator based on satellite visibility. This simulator utilises knowledge about Global Navigation Satellite Systems and functionality of real life GNSS receiver. Virtual receiver configured in this simulator is not able to obtain data broadcasted from the satellites, hence an already collected ephemerides are downloaded from selected data center. Position of satellites acquired from these navigation files are checked by Gazebo simulator for direct or obstructed visibility from the receiver. This ray casting occurs in 3D model world with obstacles of different signal permeability. Simulator calculates observations from visible satellites and stores them in output RINEX file. An advantage of this simulator lays in the ability to take in consideration the real 3D structure of the environment. Other simulators usually use only elevation mask for filtering possible satellites.

Keywords: GNSS, ROS, Visibility based GNSS simulator, Gazebo, satellite, ray-casting, pseudorange

Supervisor: Mgr. Martin Pecka, PhD.
Department of Cybernetics,
Resslova 307/9,
Praha 2

Abstrakt

Tato práce je zaměřena okolo nově navrhnutého GNSS simulátoru založeného na viditelnosti satelitů. Tento simulátor využívá znalosti o Globálních Navigačních Satelitních Systémech a funkcionality skutečného GNSS přijímače. Virtuální přijímač nakonfigurován v tomto simulátoru není schopen získávat data vysílaná ze satellitů, proto jsou z vybraného data centra staženy již sesbírané efemeridy. Pozice satelitů získané z těchto navigačních souborů jsou prověřeny Gazebo simulátorem pro přímou nebo zastíněnou viditelnost od přijímače. Toto vysílání paprsků probíhá v 3D modelovaném světě s překážkami s různými propustnostmi signálu. Z viditelných satelitů pak simulátor spočítá pozorováná data a uloží je do výstupního RINEX souboru. Výhoda tohoto simulátoru leží ve schopnosti uvažování skutečné 3D struktury prostředí. Ostatní simulátory využívají pouze minimální elevace k odfiltrování možných satelitů.

Klíčová slova: GNSS, ROS, GNSS simulátor založený na viditelnosti, Gazebo simulátor, satelit, vrhání paprsků, pseudovzdálenost

Překlad názvu: Návrh a implementace simulátoru systému satelitní navigace pro pokročilé simulace pohybu mobilních robotů v terénu

Contents

1 Introduction 1

Part I
Real world

2 Global Navigation Satellite Systems 5
2.1 Structure 5
2.1.1 Segments 5
2.1.2 Orbits 6
2.2 Current GNSSs 6
2.3 Signals 8
2.4 Coordinate systems 10

3 GNSS Receiver 13
3.1 Types 13
3.1.1 Division by features 13
3.1.2 Division by usage 14
3.2 Hardware 15
3.2.1 Antenna 15
3.2.2 Radio Frequency Front End 16
3.2.3 Baseband Signal Processing 17
3.2.4 Applications Processing 17
3.3 Operational modes 18
3.3.1 Stand-alone 18
3.3.2 Satellite-based augmentation 19
3.3.3 Differential 19
3.3.4 Real Time Kinematics 19

4 RINEX 21
4.1 Navigation Data 21
4.2 Observation Data 22

Part II
Model

5 GNSS simulators 27
5.1 Types 27
5.2 Sky-plot TLE 28
5.3 RxPlanner by Septentrio 28
5.4 Visibility based GNSS receiver simulator 29

6 Satellite information 33
6.1 Data centers 33
6.2 GNSSTK 34

7 Gazebo 37
7.1 World 37
7.1.1 Obstacles 37
7.2 Plugin 38
7.2.1 Transformation 39
7.2.2 Visibility rays 39

8 Observations 41
8.1 Pseudorange 42
8.2 Carrier phase 43
8.3 Doppler shift 43
8.4 Signal strength 44

Part III
Experiments

9 Data collection 49
9.1 Real world 49
9.2 Simulation 50

10 Comparison 53
10.1 Number of satellites 53
10.2 Observations 53
10.3 Position 55

11 Conclusion 61
11.1 Future improvements 62

Bibliography 63

Figures

2.1 GNSS architecture[4] 6
2.2 Satellite orbits 7
2.3 Main components of GPS L1 C/A signal.[14] 9
2.4 Coordinate systems: ECEF (blue), ENU (green) and LLA (yellow and green *up* vector).[18] 11

3.1 GNSS antenna radiation pattern. 16
3.2 Correlation process.[25] 18
3.3 Generic diagram of a channel in the receiver processing block.[26] . . 18
3.4 Real-Time Kinematic (RTK) overview[33] 20

4.1 Keplerian elements and satellite orbit[38] 22

5.1 Sky Plot sky view 29
5.2 Rx Planner sky view. Dark red - GPS, light purple - Galileo 30
5.3 Workflow scheme of the simulator from this thesis 31

7.1 Example model of yard at CTU at the Charles Square 38
7.2 Detail of the communication network for the simulator. Legend: solid line - interfunctional communication, dashed line - ROS messages on */topic*, dotted line - ROS parameters set before simulation . . 40

8.1 Iterative time of transmit estimation 43
8.2 Doppler effect in 1 dimension. . . 44
8.3 Fit of least squares method for C/N_0 approximation. 45

9.1 Robot with antenna and with clear view at sky. 50
9.2 Real world setup for the experiments 50
9.3 Setup for receiver simulation in Gazebo. 51
9.4 Simulation setup for the experiments 51

10.1 Sky views for the experiment with clear view 54
10.2 Sky views for the experiment under the tree 54
10.3 Comparison plot of pseudoranges from simulation and from real GNSS module for 3 visible GPS satellites 55
10.4 Differences between pseudoranges from simulation and from real GNSS module for 3 visible GPS satellites 56
10.5 Comparison plot of carrier phase measurements from simulation and from real GNSS module 57
10.6 Differences of carrier phase measurements from simulation and from real GNSS module 57
10.7 Comparison plot of Doppler shifts from simulation and from real GNSS module 58
10.8 Differences of Doppler shifts from simulation and from real GNSS module 58
10.9 Comparison plot of $\mathrm{C/N_0}$ from simulation and from real GNSS module 59
10.10 Position visualization using Open Street Map with ground truth, measurements and covariance. 60

Tables

2.1 Frequencies utilised by the GNSSs[12] 8

4.1 Observation Code Components[39] 23

8.1 Supported GNSS shortcuts and signal bands 42

10.1 Obtained positions in ECEF coordinates and differences between them 56

10.2 Positions in latitude, longitude, altitude 57

Chapter 1
Introduction

GNSS localization is the backbone of navigation and mapping by mobile robots. This thesis describes what Global Navigation Satellite Systems (GNSSs) encapsulate and how exactly they provide data necessary for position acquisition. In real world GNSS signals travel from satellites in Medium-altitude Earth orbits to ground segment's receivers. GNSS receiver is in of itself complex device specialized in obtaining as many broadcast signals as possible. From these signals, navigation data in a form of ephemerides are obtained and used for calculation of useful observations such as pseudorange, carrier phase, doppler shift and $\mathrm{C/N_0}$ signal integrity.

Additionally, it is useful to have an option to get these data without the necessity of real receiver module. Several types of GNSS simulators can be used to obtain positioning data, or to collect additional observation. New Visibility based GNSS receiver simulator is proposed as a part of this thesis. Main advantage of this simulator is utilization of Gazebo simulator with modeled 3D environment, that can provide more realistic filtering of satellites based on their visibility.

Satellite broadcast is not model in the simulator, so the collected ephemerides are downloaded from selected data center. C++ library GNSS Toolkit is used to process the navigation data and to find position and velocity of the satellites. Robot Operating System (ROS) node communicates these positions to the Gazebo plugin, where the satellite visibility is checked by casting a ray in its direction.

Data about visible satellites are used to calculate observations containing pseudorange, carrier phase, doppler shift and carrier to noise ratio ($\mathrm{C/N_0}$). All these and other useful data about the simulated receiver are stored in RINEX observation file, which is the output of the simulation together with the navigation file from data center.

Implementation of this newly proposed Visibility based GNSS receiver simulator is stored on public GitHub[1].

[1] https://github.com/ctu-vras/gnss-info/tree/simulation

Part I

Real world

Chapter 2

Global Navigation Satellite Systems

GNSSs are satellite systems that provide position and time measurements on Earth. In basic principle, receiver needs data from four separate satellites to compute its position and time by using time of flight method. Necessity for at least 4 satellites com from the nature of the task for specifying XYZ 3D position and time (3+1). Unsurprisingly, with more tracked satellites higher precision of the position calculations is expected. [1]

2.1 Structure

2.1.1 Segments

Basic structure of every Global Navigation Satellite System can be divided into three segments: space, control and user. A scheme of these segments and how they are connected can be seen in Figure 2.1[2, p. 16][3, p. 5]

- Space
  This segment consists of constellations of satellites orbiting the Earth at around 20 to 23 thousands kilometers above the sea. These satellites broadcast messages with data about their IDs, time, orbit and status necessary for position calculations by ground segments.

- Control
  First ground segment is responsible for maintaining the satellites, their paths and health. Consisting of globally placed stations, antennas and a master control station, it computes navigational data and communicate them back to the satellites.

- User
  The last and biggest segment describes end-user equipment used for receiving and identifying satellite signals such as antennas, receivers and computing units that process the signals. It encapsulates all civilian, military, ground or sea based and even in-space receivers.

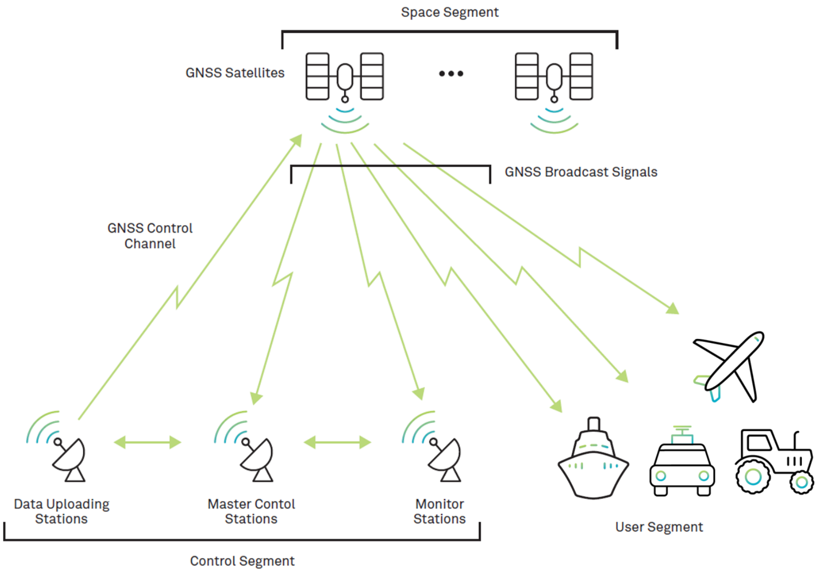


Figure 2.1: GNSS architecture[4]

2.1.2 Orbits

Satellites for navigation systems typically utilize three different orbits divided by their position with respect to Earth. Mutual position of these types of orbit can be seen in Figure 2.2.

- Medium-altitude Earth Orbit (MEO)
  In these orbits, satellites are evenly distributed around the Earth providing global coverage. MEO orbits are inclined 55 to 65° with respect to the equator, they are almost circular and they can contain several evenly spaced satellites each. Satellites in Medium-altitude orbits are flying at around 20,000 km above the surface.
- Inclined Geosynchronous Orbit (IGSO)
- Geostationary Orbit (GEO)
  Speed of the satellites in these synchronous orbits matches the rotational speed of the Earth. IGSO and GEO satellites basically hover above one place on Earth, resulting in only local coverage. GEO satellites, additionally, orbit the Earth around the equator (with inclination of 0°) and they have practically perfectly circular paths. In order to match the rotation of the Earth, synchronous satellites usually maintain altitude of 35,000 km.[5]

2.2 Current GNSSs

There are currently six satellite networks in use. GPS, GLONASS, Galileo and BeiDou are the four Global systems, and the two Regional Navigation

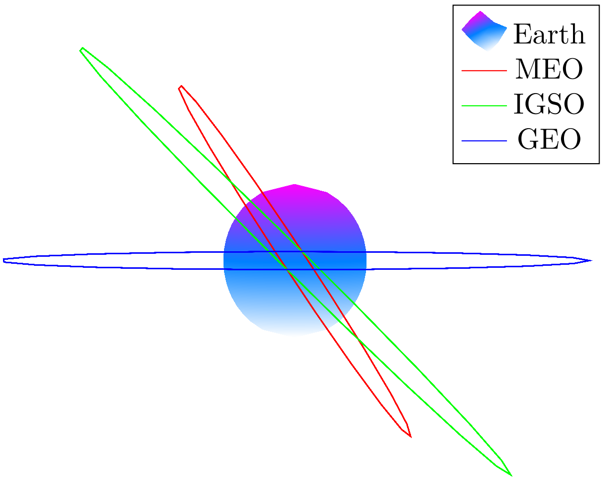


Figure 2.2: Satellite orbits

Satellite Systems (RNSS) are QZSS and IRNSS/NavIC.[2, p. 16-19][3, p. 5][6]

- GPS
  The Global Positioning System (GPS) is the first and most known GNSS. Although, it was originally build for the US military, it is from its early beginning available for any civilian with GPS receiver. The constellation of 24 satellites in 6 MEO planes (with inclination of 56°) with 4 satellites was completed in 1995.

- GLONASS
  As a direct competition to GPS, *Global'naya Navigatsionnaya Sputnikovaya Sistema* or GLONASS was put in use in 1996 and is currently operated by Russian army. This system consists of 24 satellites in 3 MEO planes (with inclination of 65°) each with 8 satellites and has been free to use by public domain since 2007.

- Galileo
  A joint effort of the European Commission and the European Space Agency planned a European based navigational system Galileo and started launching usable satellites in 2011. As of May 2024, there are 30 satellites placed in 3 MEO planes. After the commissioning of the latest 2 launches, there should be 27 operational and 3 spare satellites.[7]

- BeiDou
  The initial constellation developed by China consisted of 3 GEO satellites launched in 2003. A global system was declared operational in 2011 with addition of MEO and IGSO satellites. As of May 2024, there are 7 operational GEO satellites, 10 IGSO and 27 MEO satellites. [8]

- QZSS
  Quasi-Zenith Satellite System (QZSS) is a Japanese system designed to always have at least 1 satellite near the zenith above (directly above)

GNSS	Name	Frequency [MHz]
GPS	L1	1575.42
	L2	1227.6
	L5	1176.45
Galileo	E1	1575.42
	E5a	1176.45
	E5	1191.795
	E5b	1207.14
	E6	1278.75
BeiDou	L1	1575.42
	B1	1561.098
	B2a	1176.45
	B2	1191.795
	B2b	1207.14
	B3	1268.52
QZSS	L1	1575.42
	L2	1227.6
	L5	1176.45
	L6	1278.75
IRNSS	S	2492.028
	L5	1176.45

Table 2.1: Frequencies utilised by the GNSSs[12].

Japan. QZSS is operated as a 4 satellite (3 IGSO, 1 GEO) constellation since 2018 and it is trying to augment the the satellite visibility in urban canons and mountainous terrain often found in Japan.[9]

- IRNSS
  Indian Regional Navigation Satellite System (IRNSS) was developed by Indian government and later renamed to NavIC (Navigation with Indian Constellation) a Hindi word for navigator. It consists of 3 GEO and 4 IGSO satellites providing coverage over India and surrounding region up to 1500 km from its boundary.[10][11]

2.3 Signals

All GNSSs operate in the L-band frequency band laying between 1 and 2 GHz. This choice of frequency is low enough to allow user antennas to be as simple as possible, and simultaneously it is high enough to not be significantly delayed by ionospheric influence. Exact frequencies utilised by each GNSS can be seen in Table 2.1 Signals broadcasted by the satellites travel at the speed of light ranging from 299,792,458 m/s in vacuum to slightly slower in the atmosphere.

Even though, all satellites transmit on the same radio frequencies, it is possible to clearly distinguish them. Each satellite modulates its signal by

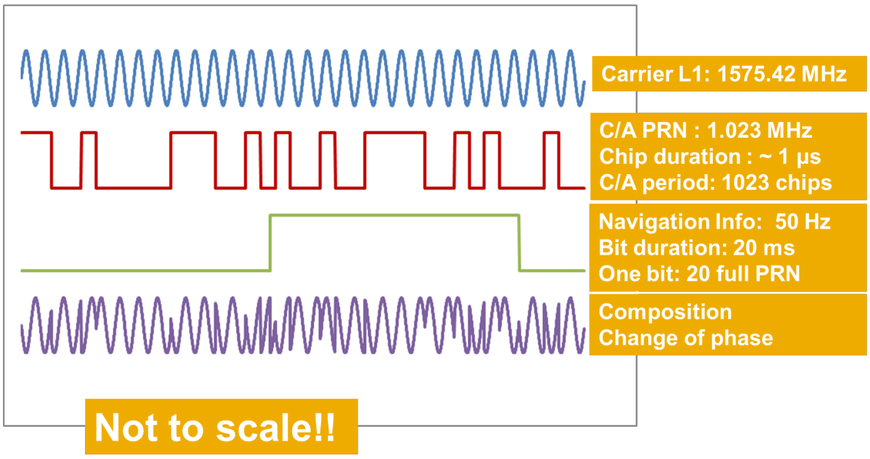


Figure 2.3: Main components of GPS L1 C/A signal.[14]

a unique pseudorandom code (PRN). This transmission scheme is called Code Division Multiple Access (CDMA). Receivers know the PRN code for each satellite and are able to track them separately (more about that in chapter 3). Final modulating component of the signal is Navigation data message containing all information needed for position and time determination. An example, how each component of the GPS L1 C/A signal looks like and how the final signal is structured can be seen in Figure 2.3.[13]

Satellites transmit their navigation messages on all possible bands. Such navigation message contains information about:

- Date and time of the satellite
- Satellite status and health
  Satellite is transmitting out-of-service message when its being repaired or is in any way unusable.
- Satellite ephemeris data
  This data allows receiver to precisely compute satellites position. More about that in chapter 4.
- Almanac data
  This consist of coarse orbit and status information about each satellite in the GNSS's constellation.

On its way from the satellite in space to the receiver at the surface of the Earth, signals have to pass through the Earth's atmosphere. As the wave passes through the different layers of the atmosphere, the radio signals are refracted and are being delayed with respect to traveling in straight line. Biggest delay is being obtained in the ionosphere layer spanning from 50 to 1000 km above the surface. Radiation from the sun ionise gas and released electrons influence the electromagnetic wave. Because these influences are frequency-dependent (delay depends on $\frac{1}{f^2}$), receivers can resolve their effect by using signals from 2 frequency bands.

Another layer, troposphere, can delay all frequencies at the same scale, so its effect can not be eliminated the same way as ionosphere can be. However, as it is dependant on local temperature, pressure and humidity, it can be modeled and compensated for. Final phenomenon causing incorrect measurements occur when the signal rays reflect on obstacles such as buildings or trees. This is referred to as multipath propagation.[3, p. 13-18]

2.4 Coordinate systems

To correctly describe satellite and planet motion, accurate coordinate systems are a necessity. There are several systems for different purposes.

- Celestial Reference System (CRS)
  Also known as Earth Centred Inertial (ECI) has its origin in the Earth's center of mass (Geocenter) and its XY plane is the mean Equator plane. X axis of this system is pointing from the Geocenter to the Vernal equinox, a place the equator plane and ecliptic plane intersects. Z axis is defined by the direction of the earth mean rotation pole and Y axis is completing the right handed orthogonal system. Because of the Earth motions, all the terms defining the CRS are used as they were on January 1, 2000 at 12:00 (J2000.0 epoch). Since the axes are set by a point independent of the Earth's rotation, the system stays intact as the planet spins.[15]

- Earth-Centred, Earth-Fixed (ECEF)
  ECEF system otherwise known as Terrestrial Reference System also has its origin in the Geocenter, however its axes are set differently. Unlike ECI system, ECEF from its definition is fixed on the Earth, meaning it will spin as the planet spins. Z axis of this system is in direction of the Conventional Terrestrial Pole (CTP), which is calculated as an average of the poles between years 1900 and 1905. Additionally, X axis instead to the Vernal equinox points towards the intersection of the equator and the Greenwich meridian.[16] World Geodetic System WGS-84 is an example of the ECEF system with additional constants defining the Earth's ellipsoid.[17]

- East, North, Up (ENU)
  Working with Earth-centred system is not often convenient for applications near the Earth's surface. For these instances a Local Tangent Plane (LTP) systems are used. There is an infinite amount of these systems, because a specific one can be constructed for each point on the surface, however, all follow the same definition for axes. Z axis is set from the point on the Earth upwards as perpendicular to the local ellipsoidal surface. X axis has the direction to the geographic East and y axis points towards the North. Visualization of an ENU system is in Figure 2.4.[2, p. 611]

- Latitude, Longitude, Altitude (LLA)
  Another coordinate system focused on surface level applications is a slight

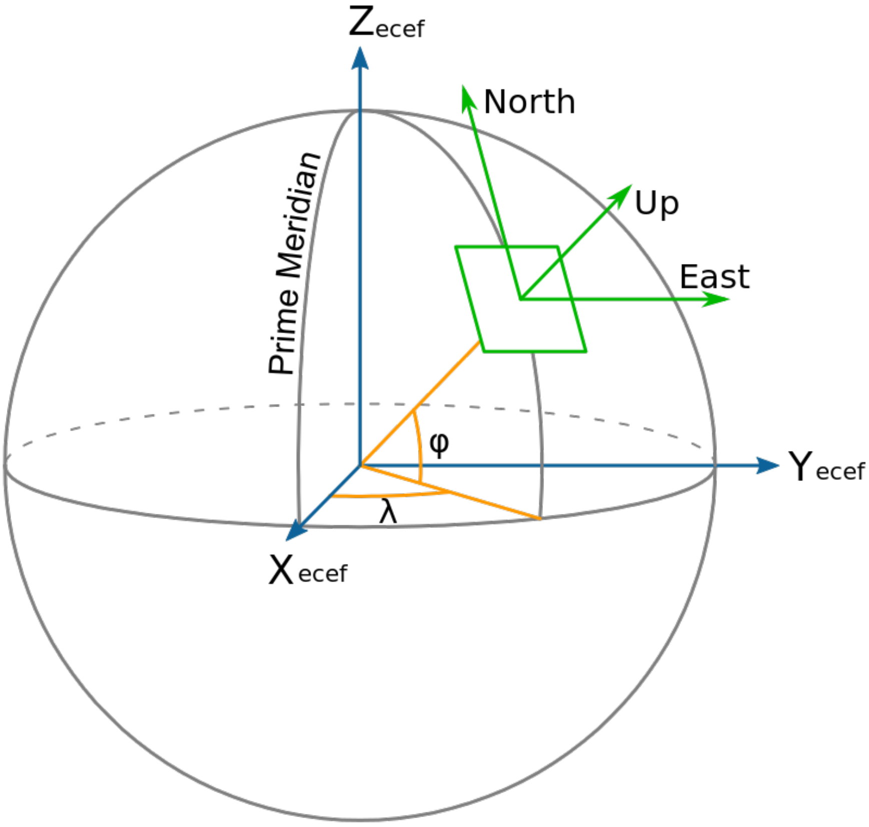


Figure 2.4: Coordinate systems: ECEF (blue), ENU (green) and LLA (yellow and green *up* vector).[18]

variation of Spherical coordinate system. These spherical coordinates defined upon Cartesian ECEF coordinates uses angles and distance to the point of interest. Latitude ϕ is defined as an angle from the XY plane of ECEF system. Longitude λ corresponds to the angle from the Greenwich meridian and altitude is set a the height above the sea level. Visual representation of the LLA system can be seen in Figure 2.4.[2, p. 31-32]

Chapter 3

GNSS Receiver

To obtain a position, the GNSS receiver needs at least 4 satellites. However, the solution will improve with each additional satellite. As mentioned before, receiver is able to track individual satellites by aligning their PRN codes with their replicas it has stored inside.[3, p. 19][19]

3.1 Types

In this section, there is a simple overview of the types of GNSS receivers. Receiver type is usually tight to the final application. Multi-constellation receiver can improve availability, on the other hand to improve accuracy, receiver with carrier-based technology can be more useful.[20]

3.1.1 Division by features

- Multi-Constellation
  These types of receivers can take signals from multiple navigational constellations. Basically, all satellites broadcast very similar set of core information. This presents an opportunity when for example GPS satellites are being obstructed by a building, receiver can track other constellations with at the moment more visible satellites.

- Multi-Frequency
  Instead of several GNSSs, multi-frequency receivers can obtain data signals from the same satellite on multiple frequency bands. This processing can lead to suppression of ionospheric delays and so to getting better accuracy.

- Augmented
  These receivers can store additional data from other monitoring receivers or from previous computations. Based on this data, they can make better correction to the satellite position computations. These augmentations can come from other geostationary satellites called Satellite Base Augmentation Systems (SBAS) or from the ground segment over radio or internet called Ground Based Augmentation Systems (GBAS).

- Differential
  Differential receivers utilize similar scheme as augmentation receivers. They can obtain corrections from 1 or more reference stations and then calculate more precise positions. These correction can be communicated via radio or internet (more in section 3.3).

- Assisted
  Receivers of this type utilize assistance systems to obtain more data and improve their functions. First improvement comes in the form of downloading almanac and ephemeris data from the internet, so the receiver can initialize faster without depending on satellite broadcast. Additionally, receiver can offload heavier computations on the server allowing it to focus on taking in more data.

- Software-defined Receivers
  These receivers place the processing into a digital forms, however, they still use hardware antennas. This comes with benefits such as using any microprocessor as a receiver or being able to store more data on cloud. Although, there are drawbacks in needing more processing power.[21]

3.1.2 Division by usage

- Navigation Receivers Handheld
  Basic handheld receivers make use of integrated antenna, integrated battery, keyboard and some display. Nowadays, any smartphone is this type of receiver. These receivers usually have software for waypoint navigation.

- Navigation Receivers Non-Handheld
  Non-handheld receiver is, in most cases, black box with interfaces for different uses, such as marine, land or avionics applications. Because these receivers are used in larger structures like airplanes, it significantly less matter how large they are or how much power they consume.

- Engines, OEM Modules, Chips and Dies
  These are all semifinished products without housing, power supply, control or display. They are usually used in specific market like car navigation systems. These receivers provide flexible hardware and software interfaces, being able to output all raw data.

- Time Transfer Receivers
  Timing receivers utilize more precise clock power by rubidium crystal. Specific software can determine precise clock offset and synchronize to Coordinated Universal Time (UTC). In the beginning, timing receivers used only 1 channel to calculate 1 unknown (time), so the user had to put into them the antenna coordinates via world geodetic system (WGS) 84.

- Geodetic Receivers
  These receivers have multisystem and multifrequency capabilities producing the most precise positions. Surveying receivers can obtain even carrier-phase measurements with millimeter level accuracy. Additionally, modern receivers have radio or internet modules that can provide RTK correction data. More about RTK in subsection 3.3.4.

- Space Receivers
  Placing GNSS receivers on-board of satellites or other space object come with several difficulties. Doppler shifts in range of tenths of kHz are caused by the high orbital velocities of about 8,000 m/s. However, if the receiver is further from Earth than the MEO or even GEO satellites, it has to rely only on the satellites broadcasting on the side-lobes from the other side of the Earth.

- Attitude Determination Receivers
  Attitude determination receivers use multiple antennas that know the precise offsets between them. While positioning, receiver is able to calculate pitch, roll and yaw of the platform. This type of receivers is utilized on terrestrial and spaceborn platforms.[2, p. 397-399]

3.2 Hardware

Even though receivers are different for each GNSS and for each application, there is overarching structure they have to follow. This general receiver structure consists of 4 building blocks capable of full signal processing.

3.2.1 Antenna

Antenna is responsible for capturing the L-band signal from satellites and converting it to electronic signal. It is also essential for out-of-band interference rejection. Current antennas focus on reception of signals coming directly from above (+3 dBic gain). Although, they try to capture as much signal from the side as well, under 5° elevation sharp pattern roll-off is utilised to stop receiving signals reflected from the ground. General GNSS antenna pattern is in Figure 3.1. There are two types of antennas based on their power. Passive ones do not require any power making them optimal for mobile devices. However, they have to be very closely mounted to the front end part of the receiver, since they do not have any amplification. Where it is not possible to mount the antenna directly on the receiver (applications on roof for example), an active antenna has to be used. By obtaining power directly from the receiver, active antenna can amplify the received signal to compensate for longer cable, where Signal to Noise Ratio (SNR) or Carrier to Noise (C/N_0) could be degraded. Phase centre of the antenna is actually where the receiver determines the position solution. Unfortunately, this phase centre is not where the physical centre of the antenna is and actually is not at single place at all. Its position is highly dependent on the direction of the

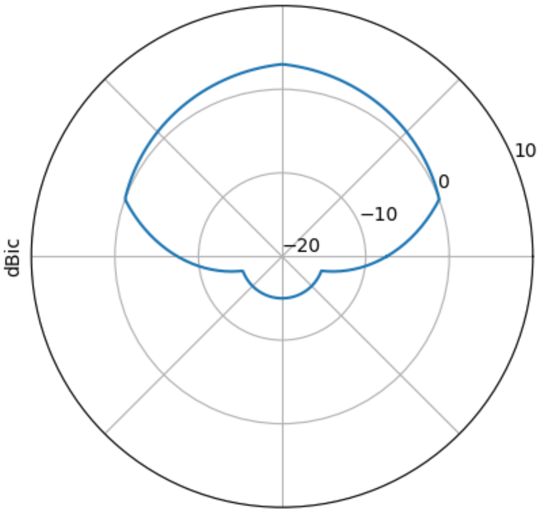


Figure 3.1: GNSS antenna radiation pattern.

receiving signal as well as on its frequency. Finally depending on the size of the antenna, this difference between phase centres can be as big as 10 cm.

There are many antenna configurations. For example patch antenna build as a low-profile microstrip is often used in handheld devices. On the other hand, choke ring antenna is surrounded by several conductive rings preventing multipath and low elevation signals to pass. Parabolic antennas utilizing the reflective dish to direct the radio waves are able to receive signals only from a particular direction. This makes them suitable for control segment but not for the users.

Another way how to look at antennas division is by their application. Larger antennas are usually used with fixed but highly accurate receivers. Choke ring antennas that support multi-frequency and multi-constellation reception are the most common. Rover antennas have to be smaller and they typically require a trade-off between accuracy and portability. Most common use cases are land survey, forestry, construction or robotics. Finally handheld receiver antennas are almost always patch antennas used in smartphones or portable navigation devices.[22][2, p. 373-376]

3.2.2 Radio Frequency Front End

The signal coming from the antenna is then filtered, down-converted and digitized in the front end part of the receiver. Firstly, filtering part of the front end is responsible for signal amplification, keeping the noise figure low and rejecting out-of-band interference. This is usually achieved by Low-Noise Amplifiers (LNA) and filter stages.

After the filters it is needed to convert the radio frequency (RF) signal to the digital signal used for further processing. Although it is possible to directly sample the RF signal, it would need the analog-digital converter (ADC) to run at RF (1-2 GHz) which would be very costly. Other way could be using the frequency folding as an advantage and by undersampling the higher-frequency content will be aliased into all of the lower Nyquist zones, however, this is also very impractical. That is why downsampling converters are used. The first method of downsampling, heterodyne, utilizes several stages that down-converts the signal into intermediate frequency (IF) signal and filters it by a low pass filter. This conversion presents the possibility to use cheaper filters and subsequently cheaper ADC. Another way called homodyne mixing offers conversion directly between the RF and baseband, however, it comes with some drawbacks. The filters used in this mixer would have to be of much higher quality and thus more costly and converting directly to baseband places harmonics near zero, making them difficult to filter out.

To output the desired digital signal, analog-digital converter (ADC) is used. Basic structure of the ADC needs an oscillator to set the sampling frequency, sample and hold circuit, threshold comparator and an encoder to convert the comparator result to bits. The most important thing is to correctly choose the sampling frequency. According to the Nyquist–Shannon sampling theorem, the sampling frequency f_s needs to be at least double of the baseband frequency B, $f_s > 2B$. Otherwise aliasing occurs and signal can not be reconstructed. Some receivers have automatic gain control (AGC), which is conditioning the amplitude of the analog signal to match the maximum and minimum thresholds of the ADC, making it more efficient.[23][2, p. 376-383]

3.2.3 Baseband Signal Processing

Usually, GNSS receivers assign 1 channel to each satellite it can see and they utilize code and carrier tracking loops. These loops are continuously estimating the code delay by aligning the PRNs and the carrier phase (or its instantaneous value, the Doppler frequency) by shortening or stretching the receiver's inner frequency. Basic diagram of such channel for tracking satellite is in Figure 3.3. Using correlation method, receiver is able to tell which satellite it is tracking. The correlation process can be seen in Figure 3.2. After synchronization with the incoming signal, receiver can store the time of acquirement. By comparing this time and the time stored in navigation message, it can use this difference to calculate range to the satellite.[19][24]

3.2.4 Applications Processing

The application processing block is responsible for extracting observables and navigation data from the signal. The least the receiver can provide is position, velocity and time (PVT) information.[27] How precise can this information be is explained in section 3.3. PVT of the receiver is computed from acquired observations, which are: Pseudorange, Doppler shift, Carrier phase and Signal strength. More about the observation in chapter 8.

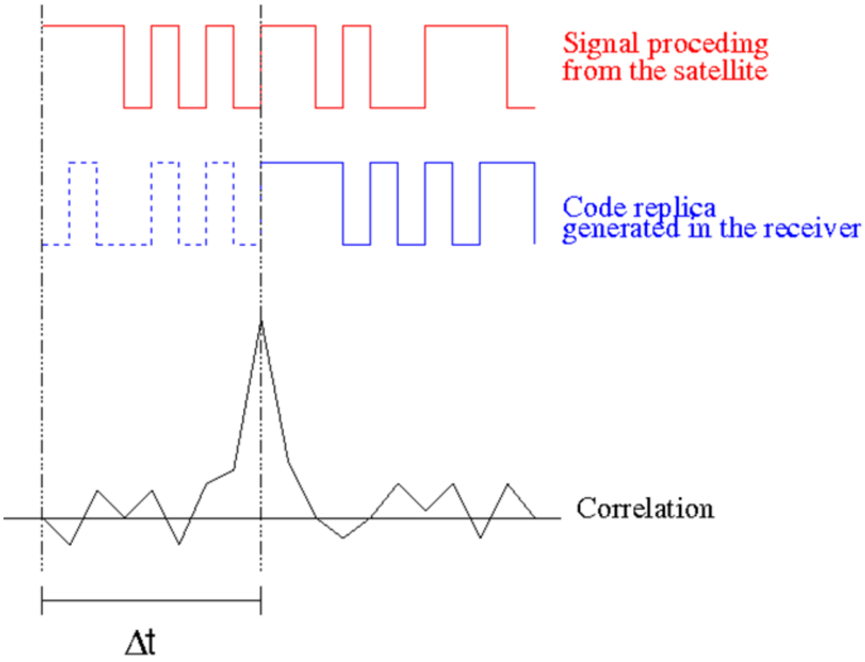


Figure 3.2: Correlation process.[25]

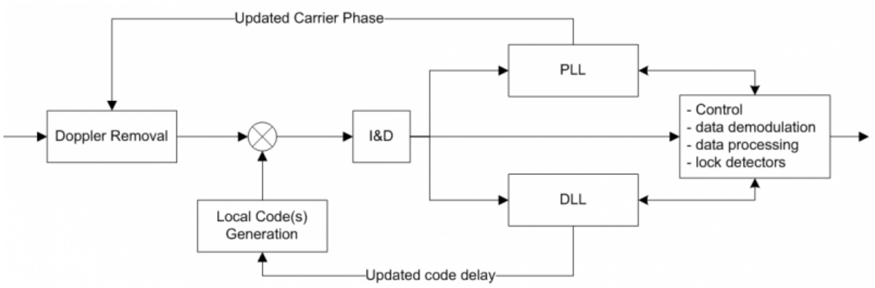


Figure 3.3: Generic diagram of a channel in the receiver processing block.[26]

3.3 Operational modes

Based on the number of satellites, their mutual positions and on visibility conditions, GNSS modules operate in several modes. Each mode having its own level of precision and computational complexity.

3.3.1 Stand-alone

The simplest and most commonly used operational mode is Stand-alone. While in this mode GNSS module collects only the positioning data the satellites provide and there are no corrections made to erroneous measurements. Based just on the basic data, calculations done in this mode can provide only 2 to 5 meters accuracy. This uncertainty can be overlooked for example in basic automotive applications, however for localization of mobile robots that are usually smaller than lower bound of the accuracy range it is not enough.[28][29]

3.3.2 Satellite-based augmentation

Next step in refinement of the data obtained by satellites is use of Satellite-Based Augmentation systems. These systems use measurements collected by precisely located stations that put together all measured errors. Computed corrections are then send to geostationary satellites from where they are broadcast to moving GNSS modules on vehicles or in mobile robots. Although, the geostationary satellites provides corrections, still the nature of the data transfer blocks better position estimates then those with one meter accuracy.[30][29]

3.3.3 Differential

Similarly to Satellite-based augmentation, Differential GNSS explore the premise that any GNSS receivers in close enough proximity experience the same atmospheric errors. There are several stationary receivers placed at exactly specified locations, that collects data from the satellites, compares them to the knowledge of their own position and calculates potential corrections necessary to achieve similarity. Then these stations broadcast these corrections via radio or internet (NTRIP) to other moving GNSS modules in their proximity. Moving receivers or otherwise called rovers, can (based on their equipment) connect to the stations from up to three hundred seventy kilometers, however, closer to the station the corrections are more accurate and can lead to position estimates with up to thirty centimeters precision.[31][32][29]

3.3.4 Real Time Kinematics

The most advanced enhancing method providing centimeter accuracy is called Real Time Kinematics (RTK). Again this method utilises second stationary receiver placed on predetermined coordinates denoted as base and the mobile receiver called rover we use for the data collection. Similarly to the differential mode, RTK presume that having these two receivers in perimeter under thirty kilometers provides same atmospheric ambiguities that can be omitted by the base station. A scheme of RTK setup can be seen in figure 3.4

However, to provide the centimeter level accuracy, there are several mathematical computations necessary. The pseudo-range measurements take in consideration the time when the data message was sent t_s and when it was received t_r and together with the presumption that the signal travels at the speed of light in atmosphere c we get this equation

$$d = c \cdot (t_r - t_s). \tag{3.1}$$

This measurements can provide just meter level accuracy because even ten nanosecond inaccuracy in the clock of the receiver or the satellite can lead to three meters error.

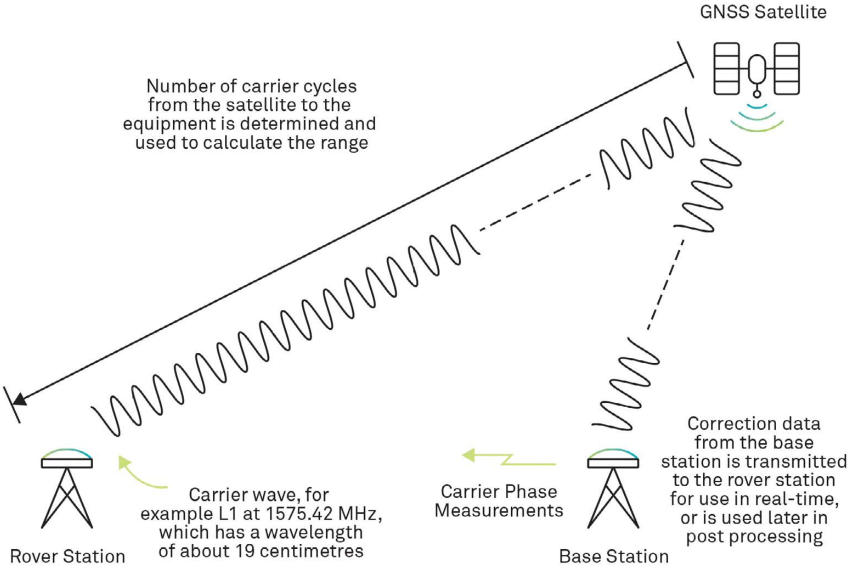

GNSS Satellite
Number of carrier cycles from the satellite to the equipment is determined and used to calculate the range
Correction data from the base station is transmitted to the rover station for use in real-time, or is used later in post processing
Carrier Phase Measurements
Carrier wave, for example L1 at 1575.42 MHz, which has a wavelength of about 19 centimetres
Rover Station
Base Station

Figure 3.4: Real-Time Kinematic (RTK) overview[33]

More advanced way of determining the distance between the satellite and the receiver is the carrier phase measurement. As the name implies with this measurement rover can calculate where exactly it is in the sine wave λ_ϕ in which form the signal travels. However, for the distance calculation it is necessary to know how many whole sine waves N are there between the satellite and the receiver. With the knowledge of the wave length λ (19 cm for L1 frequency band) the rover can calculate the distance by following equation

$$d = N\lambda + \lambda_\phi. \tag{3.2}$$

These calculations called ambiguity resolution process are run inside the rover receiver as well as inside the base station and for all connected satellites. The goal is to find solution for N in whole number of wave cycles, the so called **fix** positioning. If no such solution has been found yet, it will use the **float** positioning result instead.[34]

Chapter 4
RINEX

Receiver Independent Exchange format (RINEX) was developed by Astronomical Institute of the University of Berne and is used as interchangeable format for GNSS data. Each RINEX file has header part with global information and data part. RINEX utilizes mainly 2 types of ASCII files, Navigational data and Observation data. There are currently 4 major versions of the RINEX format. The first version from 1989 was able to capture only GPS information. Other GNSSs were added in second and third refinements. The newest fourth version is able to accommodate even the ionospheric corrections or Earth orientation. To contain other types of information, additional RINEX-like formats are used, like:

- IONEX - format for storing ionosphere models
- ANTEX - stores Phase Center Offsets (PCOs) and Phase Center Variations (PCVs) of geodetic GNSS antennas
- RINEX CLOCK - exchanges information about satellite and receiver clock offsets

Several other data formats are widely used in GNSS applications. Standard Product 3 (SP3) format stores calculated positions, clock offsets and potentially velocities of satellites specified in its header. Almanac data formats such as YUMA, SEM or Two Line Element (TLE), stores the most bare bone information needed to obtain satellite position. Lastly, for communication of corrections between reference stations and receivers RTCM format proposed by Radio Technical Commission for Maritime and its internet version Networked Transport for RTCM via Internet Protocol (NTRIP) are used.[35]

4.1 Navigation Data

RINEX file navigation data is actually a collection of ephemeris data broadcasted by each satellite. Navigation messages with these data can be collected by any receiver and subsequently utilized to estimate its position. The header section of the navigation file is in the later versions identical for all GNSSs. Data necessary for valid header contain:

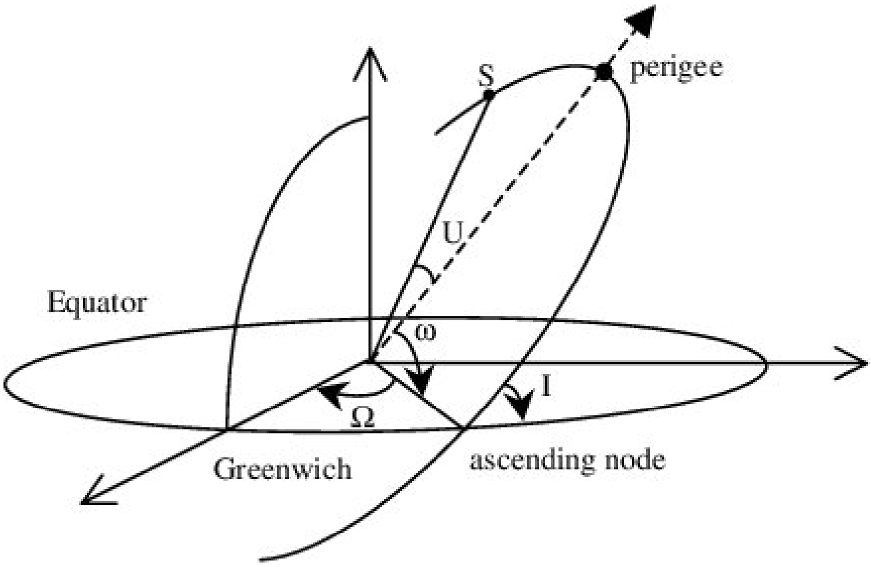


Figure 4.1: Keplerian elements and satellite orbit[38]

- Version of RINEX used for this file
- Type of file, being navigation or observation
- GNSS designation or mixed designation if information about satellites from more than 1 constellation is present
- Program and authority responsible for the creation of this file
- Time of the creation of this file
- and End of header marking

Data portion of the navigation file is populated by records for each satellite with floating point numbers. Number of contained data and format of the records is specific for each constellation. However, the dataset for each satellite has to consist of six Keplerian orbital elements, three rate parameters, three pairs of sinusoidal corrections and reference ephemeris time.[36, p. 37]

The six Keplerian parameters describe the satellites orbit as well as its position on it. In the Figure 4.1, there can be seen 3 of these elements: argument of ascending node Ω, argument of perigee ω and inclination i. Remaining parameters are: orbital semi-major axis a, eccentricity e and mean anomaly M. True anomaly, distance between the satellite and its perigee, can be seen in the Figure 4.1 as U.[37]

4.2 Observation Data

RINEX observation file contains already processed information from the signals received from satellites. This type of files can be done for separated GNSSs or have data from all constellations mixed together. Similar to the

observation	C	L	D	S
type	pseudorange	carrier phase	doppler	signal strength
band / frequency	1, 2, ..., 9			
attribute	tracking mode or channel, eg. C, P, I			

Table 4.1: Observation Code Components[39]

navigation file, this file is divided into two parts. In header section, however, there are many additional designations as the data is dependent on the receiver itself. Nature of the marker, receiver type or antenna type all have to be specified. Estimated position of the receiver as well as the offset of the antenna to this position is needed. The main difference comes in storing the observations:

- Pseudorange - travelling time of the signal multiplied by the speed of light gives the estimated range between receiver and satellite
- Carrier phase - two orders of magnitude more precise measurements of the range, but unfortunately, ambiguous by an unknown integer number of wavelengths
- Doppler shift - time derivative of its carrier phase, this shift is determined by the relative radial velocity of the satellite and receiver
- Signal strength - on a scale from 1 to 9 it is based on carrier to noise ratio (C/N$_0$)

In order to correctly pair the measurements, satellite and carrying signal, observation codes for each constellation have to be stored in the header. From RINEX version 3 they follow a 3 character structure. Each observation code is build by following Table 4.1. For example *L1C* can be decoded as carrier phase measurement from civilian signal C/A on L1 frequency band. In the data section of observation file, each satellite has that many measurements as there are observation codes, and in the same order, for its constellation in the header.[36, p. 12-26]

Part II

Model

Chapter 5

GNSS simulators

GNSS simulators possess the ability to control conditions, simulate all possible signals and they are consistently repeatable. Simulators model receiver and satellite motions, signal characteristics and other effects influencing the propagation of the signal.

5.1 Types

- Live Sky View
  This type of system focuses on reproduction of live outdoor signals and on transferring them to the indoors testing facilities. Key values of this simulator are utilization of the real world signals as well as being relatively cheap. However, re-radiation of the RF signals can cause unpredictable harm to other tests running on the receiver. Additionally, user does not have any control over the simulation environment and the repeatability of the testing is impossible due to the constant movements of the satellites.

- Pseudolites
  A pseudo satellite device is responsible for representing the real satellite. These transmitters are used more so as an augmentation devices. However, time synchronization and the idea of stationary transmitter are major difficulties for conventional receivers.

- RF record and playback systems (RPS)
  RPSs hold similar functionality as the live sky viewer, but they present another opportunity in the form of storing the signals. The device records all the RF energy in the bandwidth, downconverts it, digitizes and stores in simple media storage. Subsequently the data can be converted back to the analog signal and replayed on the original frequency. This solution offers consistent repeatability with real world signals.

- Multi channel RF constellation simulators Although there are even single channel simulators, they are so useful since the receivers can connect to multiple channels and constellation at the same time. Multi constellation simulators present solution for both these problems. Such simulator

> should provide faithful representation of all present GNSS constellations and it should allow user to control the simulation environment.[40]

Most of these simulators are kept behind unknown paywall and the free open source platforms do not excel in the number of features. In the following section, there are briefly described two free simulators focused on calculating visibility of the satellites based on location and time.

5.2 Sky-plot TLE

Open source repository gnss-info made by Mgr. Martin Pecka Ph.D encapsulates software for obtaining information about GNSS satellites and for subsequent visualization of the visible ones. This project consists of 3 Robot Operating System (ROS) nodes with support of two custom libraries based on GNSSTK C++ library. Node publish_all_satellites utilizes IGSSatelliteMetadata library which it uses to obtain satellite information from the International GNSS Service (IGS). Subsequently list with information about satellites at the specified time instance is published as custom ROS message on /satellite topic.

Second ROS node publish_sky_view make use of the second custom library OrbitalDataManager, that is able to download almanac information about all satellites received on the /satellite topic. Data obtained by the manager are in the form of TLE and GNSSTK library is used to calculate the Earth-Centric Earth-Fixed (ECEF) positions of the satellites. Based on parameterized angle of elevation and subscribed position of the view point, it determines which satellites are visible. The list of visible satellites with additional information needed for the sky plot is subsequently published on /sky_view topic.

Finally, the main node sky_plot written in Python subscribes to all the data and produces visual of the sky view. After loading information about all satellites and subsequently the position of visible satellites, this node plots them in polar coordinates as an image with different color for each constellation. This image can be visualized by the RVIZ visualization tool. Example of sky view from the yard at Charles Square can be seen in Figure 5.1.[41]

5.3 RxPlanner by Septentrio

RxPlanner is a part of RxTools package developed by Septentrio, that can be downloaded for free. RxTools is a set of graphical user interface (GUI) applications for monitoring and configuring GNSS receivers as well as for data analysis. Similar to the Sky plot, RxPlanner is a software tool that visualise visibility of satellites from a predefined place and over some period of time. Simulator does not need a physical receiver, however, it requires applicable ephemeris data. When the currently stored data does not apply for the desired time period, RxPlanner downloads new set.

Main advantage of this simulator is its graphical interface, where user can easily pick the viewing point on the map. Starting time can be set to a

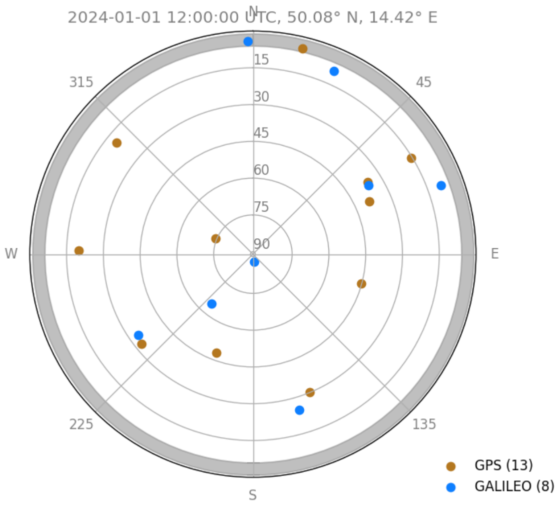


Figure 5.1: Sky Plot sky view

specific date and hour and the duration of the simulation is configurable from 1 to 24 hours. User can also select which constellations or even specific satellites should be processed, however, only GPS, GLONASS and Galileo systems are supported. Based on these settings, RxPlanner can produce several visualizations of number of visible satellites, estimated quality of the position measurements or the sky view with visible satellites and their paths over the time interval. Example of this sky view is for comparison in Figure 5.2.[42]

5.4 Visibility based GNSS receiver simulator

Since neither of these free simulators take in consideration the actual 3D representation of the targeted scene, a new Visibility based GNSS receiver simulator is proposed as a part of this thesis. There are few commercial simulators offering the ability to simulate GNSS receiver in a 3D model environment based on real conditions, but none open source solution was found by the time of writing this thesis. The new proposed simulator should work with the virtual environment modeled in Gazebo, where by using ray-casting technique visibility of the satellites is obtained (section 7.2). Output of this simulator is in the form of 2 RINEX files, one with navigation data downloaded from BKG data center (section 6.1, and second one with calculated observations based on the position of the virtual receiver and visible satellites (chapter 8). A flowchart scheme of the simulator and its inputs and outputs is in Figure 5.3. Communication processes between ROS nodes running on the simulator can be seen in Figure 7.2.

Implementation of this newly proposed Visibility based GNSS receiver sim-

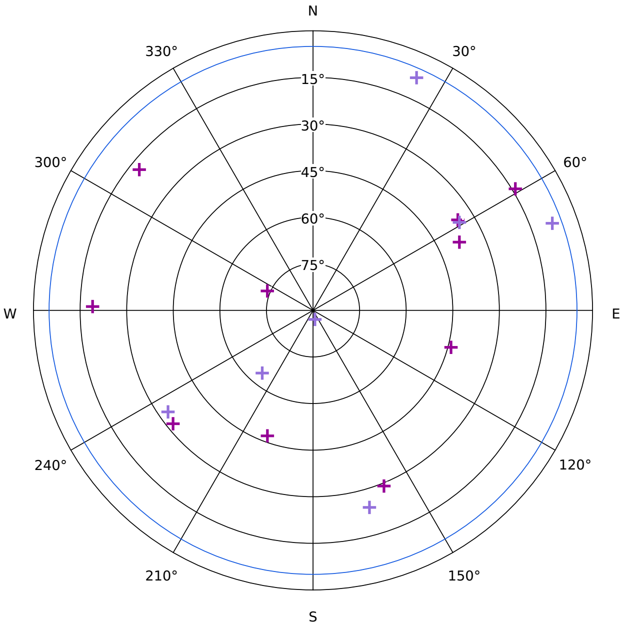


Figure 5.2: Rx Planner sky view. Dark red - GPS, light purple - Galileo

ulator can be found on public GitHub[1]. Installation of this simulator follows this basic installation steps for any ROS environment. Detailed steps are described in the README in the repository. Whole simulator can be launched by one launch file by calling **roslaunch gnss_info simulation.launch**. This ROS launch starts all necessities for the simulator containing:

- gnss simulator - ROS node for downloading ephemerides, calculating observations and storing them
- simulation_world - Gazebo world with 3D model and plugin for checking satellite visibility
- sky_view - ROS node from the Sky-Plot simulator used for visualization of sky view with satellites
- rviz - Visualization tool with image from sky_view node

[1] https://github.com/ctu-vras/gnss-info/tree/simulation

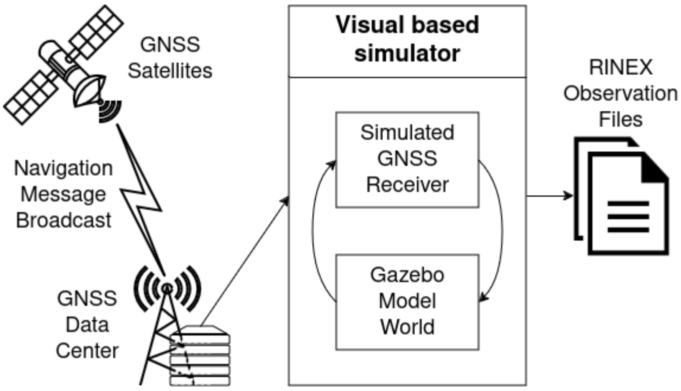


Figure 5.3: Workflow scheme of the simulator from this thesis

Chapter 6

Satellite information

As discussed in the chapter 5, there are many different simulators. However, they can be divided based on their data acquisition. First type of simulators, like live sky view or RPSs, utilizes real data broadcasted by the satellites. Second type acquire the necessary data from data centers with collected information about most of the satellites, typically just in one file.

6.1 Data centers

For this thesis, several different data centers were utilised as the project went through its development, starting with the Federal Institute of Technology in Zurich (ETHZ), that is being used as a database for the Sky plot simulator. Database, started in 2023 by Space Geodesy Group from the institute, stores orbital data in the form of Two Line Elements (TLE). TLEs are shorter and less accurate version of almanacs. The original data is being pulled from CelesTrak center and then made accessible by an Application Programming Interface (API). CelesTrak is an online database run by Space Data Center (SCD) in Hawaii. Its main objective is to monitor orbits of over 750 satellites for close approaches with other satellites or space debris.[43][44]

As the scope of this thesis requires more precise data than TLEs a suitable database for data acquisition is Crustal Dynamics Data Information System (CDDIS). CDDIS serves as the central databank for the National Aeronautics and Space Administration (NASA) and as a global archive for the IGS since 1992. There are stored GNSS products and highly accurate ephemerides for free usage for example by geophysicists. Data from over 450 stations are published on a daily basis. More so, additional data is being archived on an hourly basis from 300 sites and on sub-hourly basis from 160 stations. Although, the downloading of the files is free, user needs to be logged in, which complicates things for autonomous programs.[45]

A GitHub repository of a Comma AI company developing an open source driver assistance system openpilot also stores files with GNSS ephemerides. Even thought the files are actually the same as in the CDDIS database, there are stored files only for the GPS constellation.[46] As discussed in chapter 8 the simulator described in this thesis should and can work with other GNSSs. Finally, this requirement for multi GNSS ephemeris files was met by the

Federal Agency for Cartography and Geodesy (BKG) in Germany. BKG stands as a regional data center for Europe and collects data from world-wide reference stations. The ephemerides and other GNSS products are generated by the IGS and they are being stored on HTTP and FTP servers for free downloading. BKG also provides some form of NTRIP corrections.[47]

6.2 GNSSTK

Acquired data needs to be read and processed. The GNSS Toolkit (GNSSTK) is an open-source C++ library run by Space and Geophysics Laboratory (SGL) at The University of Texas at Austin. Main focus lies in making the GNSS data processing accessible for satellite navigation community and for GNSS application development. GNSSTK library offers implementation of models and algorithms found in navigation systems. As part of this thesis, the toolbox is used as the main tool for reading downloaded data files in RINEX format, storing them in adequate data structure and for accessing the data from this structure.

Basic workflow for processing navigation files consists of 5 classes provided by GNSSTK.

- Stream
  Formatted File Stream is a root class responsible for input/output operations with files. Internal record number variable automatically increments as the data is being read and resets when new file is opened. Extension classes intended to read text files such as RINEX Navigation Stream utilize this record number in augmented form as a line count.

- Header
  Most of the data files in GNSS have general header for all following data in the file. Based on a Formatted File Data (FFData) class, header classes stores usually the first object read/written from the stream. Common elements present in the header are time of the file creation, program and its agency, that created the file and basic information about the file type.

- Data
  Second structure created upon the FFData is data class specific for a type of file, for example RINEX Navigation Data. Data classes are responsible for processing the information read from the stream and for storing it in adequate variable types.

- Factory
  Navigation Data Factory serves as an internal storage for navigation file opened and decoded by the previous classes. Main feature of the class is an instance of map structure, that can be organised and searched based on satellite id and other factors.

- Library
  Navigation Library is the overarching structure for storing data factories.

This library serves as the entry point for looking up navigation message data. Primary function getXVT searches for data about the desired satellite in all available factories and returns calculated position, velocity and time of the measurement in the ECEF coordinate system.[48]

Chapter 7

Gazebo

Gazebo is a 3D simulator with a toolbox of development libraries and cloud services. The main scope of the simulator is to accurately simulate set of robots in indoor or outdoor environments. Gazebo offers high fidelity physics and sensors simulations. Additional realness for the simulations is presented in the form of rich library of robot models, environments and a variety of sensors. Gazebo is ideally suited for testing robotics algorithms, designing robots or performing testing in realistic scenarios, as it provides easy communication with ROS and its nodes. As the part of this thesis, gazebo serves as a physics engine for ray casting calculations.[49][50]

7.1 World

First part of each Gazebo simulation is 3D world. As mentioned above, there are many high quality models available from the first installation of the Gazebo and much more models are easily downloadable. There are 2 complimentary ways how to design worlds in Gazebo. First is to open the Gazebo GUI and insert models, shapes, and environments, that can be reshaped and transformed. This way has the advantage of being able to see the results in real time. Second, less appealing, way is to open the world file (XML format) or start from scratch in a code editor. Although this way the user cannot easily see the 3D representation of the code, it can provide a benefit in form of being precise or when the user needs to generate many relatively simple models.

Gazebo model of the simulating environment is set in coordinates with respect to the Gazebo origin. However, this origin can be set to a specific place on the Earth, meaning the model is approximately in the same place as the real world counterpart. Usually the environment is simplified just to flat ground plane and rectangular obstacles. This simulator does not come with predefined models and it is up to the user to provide them.

7.1.1 Obstacles

There are just 2 types of obstacles supported for this simulator. Impenetrable structures such as buildings, walls or the ground plane completely blocks

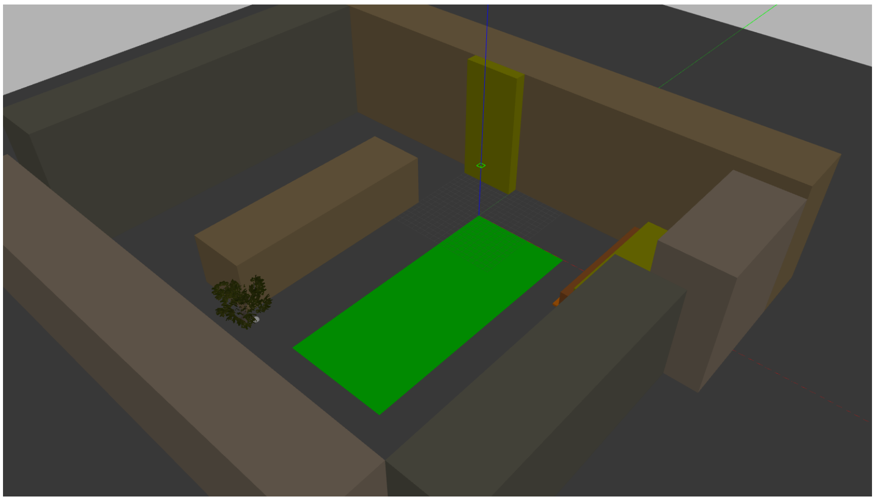


Figure 7.1: Example model of yard at CTU at the Charles Square

signals traveling from satellites. All these structures should be part of one static model, where each link represents one obstacle. Additional models of trees, bushes or other semi-passable objects can be added separately. These models can let the physics ray shape through, however, it decreases the signal strength. An example model of the gazebo environment can be seen in Figure 7.1.

7.2 Plugin

Plugin in Gazebo is code compiled as library that can be inserted into the simulation and that have almost full control over the simulation through C++ classes. Plugins can be inserted and removed from the running Gazebo system. There are several types of plugins each with different scope of control over Gazebo. Models are usually attached to the part of Gazebo of the same scope, for example world plugin is attached to the world instance in Gazebo. Developer chooses the type of plugin based on the desired scope. Gazebo utilizes hierarchical structure for objects, which means sensors are part of the models and models are part of the worlds.[51]

Gazebo plugin used as a part of work on this thesis is a World type plugin, meaning it is attached to the whole world described in section 7.1. As all Gazebo plugins, this one is written in C++ as a extended class of the World Plugin class. The main function *Load*, that is being called when the plugin is loaded to the simulator, focuses on the class initialization as well as on an initialization for a ROS node part of the plugin. ROS node instance for this plugin is crucial for communication with other nodes running on the system (as described in chapter 8). Plugin as a whole has several tasks running asynchronously as they are call upon by the ROS, however, the overall structure can be divided by its focus into 2 parts.

7.2.1 Transformation

Functions from GNSSTK library returns the satellites positions in ECEF coordinate system, however, in order to be able to model the environment in Gazebo more easily, the whole Gazebo world is transferred to the real world position of the environment. This transformation is done by substituting the latitude ϕ and longitude λ coordinates of the desired point, that would become the origin for the Gazebo ENU frame, into the Equation 7.1, where (x_0, y_0, z_0) are ECEF coordinates of this now ENU origin point.

$$\begin{bmatrix} E \\ N \\ U \end{bmatrix} = \begin{bmatrix} -sin\lambda & cos\lambda & 0 \\ -cos\lambda sin\phi & -sin\lambda sin\phi & cos\phi \\ cos\lambda cos\phi & sin\lambda cos\phi & sin\phi \end{bmatrix} \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \right) \tag{7.1}$$

When the plugin is loaded into the Gazebo simulation, robot starting position in ENU frame is obtained from ROS parameters */robot_position/x*, */y* and */z*. Using inverse operations from Equation 7.1, robot position in ECEF coordinates is published on a ROS topic */robot_position_ecef* for subsequent calculations in chapter 8.

7.2.2 Visibility rays

In order to correctly receive GNSS signal a Line of Sight (LOS) needs to be established between the receiver and broadcasting satellite. For this an instance of Ray Shape from the Gazebo physics engine is created at the plugin initialization. After receiving custom ROS message SatellitePositions on */satellite_positions* topic, a check is run for each satellite in the message, whether there is LOS to receiver. Position of the robot in ENU coordinates is set a the starting point of the ray and current satellite position (after transformation from ECEF to ENU) is set as the end point. Name of the first obstructing entity (if any) and distance to it is filled upon calling ray shape's member function GetIntersection. If no entity is in the ray's path, satellite is assumed to be visible from the receiver position and it would provide healthy data. Such satellite can be sent back to the GNSS simulator in the same ROS message format on */visible_satellite_positions* topic. However, if the encountered entity is semi-passable, like a tree, the ray shape can continue from the point of collision (+10 cm) in the original direction. This repeats until a static wall entity or no entity is encountered. Every time, it collides with tree-like object, an α coefficient is increased by 1 (from 0 to 10), which is an equivalent of 1 dbHz decrease in signal strength.

Additionally, if the satellite is visible its azimuth and elevation with respect to the robot position is calculated. With other necessary pieces of information such as frame_id, time stamp and robot position as reference point, it all fills custom SkyView message. These messages are publish on */sky_view* topic, where the sky_plot node from Sky Plot simulator is subscribed to. As discussed in section 5.2, this node can produce images of the sky with relative satellite positions and color them based on which system they belong to. For

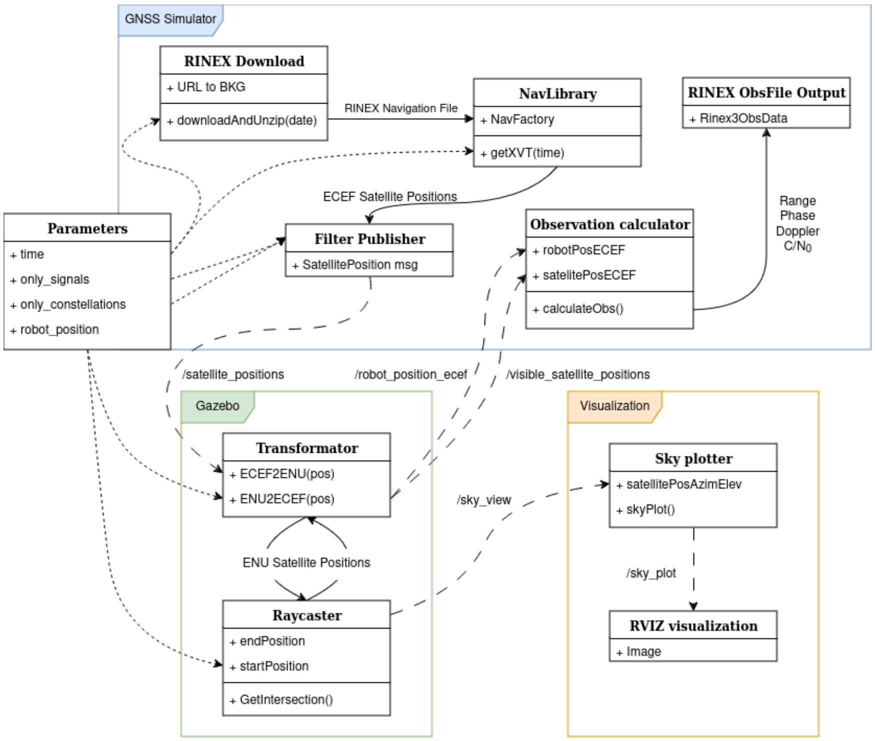


Figure 7.2: Detail of the communication network for the simulator. Legend: solid line - interfunctional communication, dashed line - ROS messages on */topic*, dotted line - ROS parameters set before simulation

better understanding, the communication network between nodes can be seen in Figure 7.2.

Chapter 8
Observations

Second ROS node running with the simulator is responsible for usage of GNSSTK library as well as for downloading correct navigation files and communication with Gazebo simulator. This simulator runs in real time from the starting time obtained as a ROS parameter */time* at the start up. Suitable time window, where the simulator is able to obtain satellite information, is from 2017 until current date, based on the selected data center. As discussed in section 6.1 the BKG data center in Germany is best suited for this application since it is free, open without login and it collects all necessary data. URL string for download is formatted with year and day of the year derived from the starting time. Only one navigation file with ephemerides for one day is downloaded, which means the simulation runtime is limited to one hour after the last data in the file.

Other ROS parameters like */only_signals* and */only_systems* when set determine the ability of simulated receiver to process them. If not set, receiver will not filter out any signals. Supported signals and navigation systems with correct formats are described in Table 8.1. Signals need to be set as a list of *shortcut*+"_"+*band* systems as list of *shortcuts*.

After obtaining the initial parameters and downloading necessary data, a method triggered by a ROS timer at 2 Hz frequency repeatedly requests the satellite positions from the navigation library at current simulation time. If the satellite's constellation is in the filtered systems and the navigation library is able to find its position at that time, all necessary information about the satellite is added to the SatellitePositions message. This message contains information about each filtered satellite necessary for processing it via Gazebo simulator and for subsequent readmission.

Output of the simulator are 2 RINEX files. First navigation file is downloaded from the data center and is outputted as is. Both header and data part of the second Observation file are filled up by the simulator. In addition to the basic information, such as what program under what agency and when outputted this file, header of RINEX observation file needs estimated robots position in ECEF coordinates and antenna offset to this position. This simulator works on the basis, that robot's position is known with respect to the model environment in Gazebo. This local ENU position is transformed by the Gazebo engine into the ECEF frame and published for this node

Name	Shortcut	Frequency band names
GPS	GPS or G	L1, L2, L5
Galileo	GAL or E	E1, E5, E5a, E5b, E6
BaiDou	BDS or C	L1, B1, B2, B2a, B2b, B3
QZSS	QZS or J	L1, L2, L5, L6
IRNSS	IRN or I	S, L5

Table 8.1: Supported GNSS shortcuts and signal bands

on */robot_position_ecef* topic. Actually, received position is the antenna's position, because inside the simulation the receiver and antenna are one dimensionless point, hence the offset is vector of zeros.

However, the most important part of the observation header is a dictionary with vector of observables for each supported navigation system. These observations consists of pseudorange to the satellite, carrier phase measurement, doppler shift calculation and perceived signal strength. Inside of this simulator, all of these calculations are frequency dependent. Meaning if utilization of all 3 frequency bands of GPS is enabled in the simulation, there will be 12 observables assigned to the GPS in the header dictionary. This initialization of the output observation file and its header are done just once on the start of the simulation. Data part of the file is gradually filled as the information about satellite visibility is obtained on the */visible_satellite_positions* topic.

8.1 Pseudorange

As discussed more thoroughly in section 3.3, pseudorange measurements are the most important information the GNSS receiver needs for estimating its position. Usually, pseudorange is computed based on the time of flight of the received signal. The time of transmit is not known to the simulator, but it is not possible to just calculate a range to satellite position obtained at receive time.

True transmit time necessary for correct satellite position can be estimated by iterative algorithm visually described in Figure 8.1. As a setup, position of satellite is obtained at the receive time, however, from the distance to the receiver Δt_1 estimating the time of flight is calculated. Iterative recalculation of the satellite position and delta time leads to the estimated time of flight. From this time, the true satellite position at transmit time cam be obtained, taking the rotation of Earth in consideration. Raw range can be then calculated as a difference between the satellite and receiver positions. Additional corrections are made from satellite's clock offset and from the relativity corrections mitigating the time dilatation.

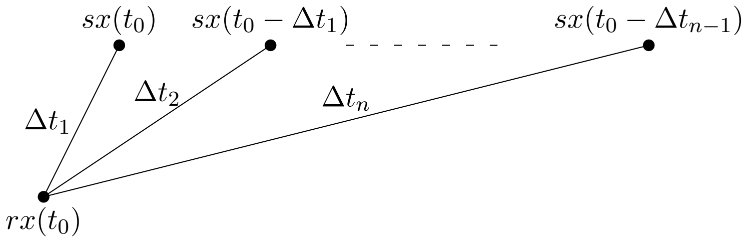


Figure 8.1: Iterative time of transmit estimation

8.2 Carrier phase

Carrier phase is measured as a more precise alternative to the pseudorange measurement. In the real system, carrier phase is measurements are based around the wave-like nature of the signal. However, the wave-like propagation of the signal from satellite is not modeled in this simulator. Simple solution how to calculate the number of cycles the carrier traveled can be estimated by dividing the previously calculated pseudorange by the wave length of the signal. Wave lengths for all possible frequency bands are known to the simulator.

In the RINEX observation file, the shift in this number of cycles needs to be stored. To satisfy this constraint, the first number of cycles calculated for each satellite and for each frequency band is stored and subsequently subtracted from all following observations with these properties. As long as the satellite is visible during the simulation, the loss of lock bit is not set in the output file. When locked on, the receiver can calculate the distance to the satellite more precisely, as described in subsection 3.3.4.

8.3 Doppler shift

The Doppler shift in frequency is the difference between the carrier frequency transmitted from the satellite and the frequency perceived by the receiver on the arrival. This shift is dependant on the relative radial velocity of the satellite and receiver. Generally, when the two objects (transmitter and receiver) are moving closer to each other, the observer perceives higher frequency than the sender actually transmit on. This effect in 1D space is visually explained in Figure 8.2. Based on the movement of the source to right, the red receiver would observe higher frequency than the green receiver.

Virtual receiver modeled in the simulator from this thesis is not able to measure the Doppler frequency as the real receiver, which shifts its frequency until it matches the receiving one. Instead simulator uses this general formula for Doppler effect in 3D

$$\frac{f'}{f} = \frac{c - \overrightarrow{v_r}\,\overrightarrow{n}}{c - \overrightarrow{v_s}\,\overrightarrow{n}}, \tag{8.1}$$

where f' stands for the shifted Doppler frequency, c is for the speed of light

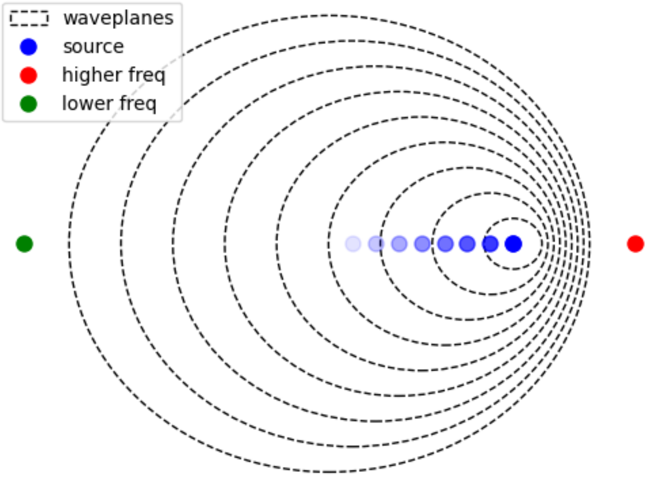


Figure 8.2: Doppler effect in 1 dimension.

(speed of the wave propagation) and $\overrightarrow{n}$ stands for the relative vector from the satellite to the receiver. Finally to obtain the shift in frequencies that is stored in RINEX output file, the actual transmitting frequency f is subtracted from the calculated Doppler frequency f'.

8.4 Signal strength

In real life, the signal strength is influenced by many sources, mainly by the environment around the receiver. In the RINEX observation file, this strength is indicated by Carrier to Noise ratio in dbHz units. This unit needs to be set in the header of the file, otherwise, the signal strength is given on a scale from 1 (poor) to 9 (strong).[36, p. 25]

For the proof of concept of this simulator simple transformation function from the pseudorange to the strength based on logarithmic scale is used. Final formula utilizing the logarithmic function goes like

$$s = (-635.64 * ln(range) + 10754.07) - \alpha. \tag{8.2}$$

Real measurements under clear sky view were taken in order to determine correct constants for the logarithmic function and to omit the α parameter. Least squares method was used as the optimization algorithm. Obtained constants should be recalculated for each antenna in use. Best fit for the antenna used in experiments (Part III) can be seen in Figure 8.3.

Additionally,strength calculated by the logarithmic function can be decreased if the ray in Gazebo simulation hits an obstacle (like a tree), that lets it through but lowers the signal strength. When the ray can travel

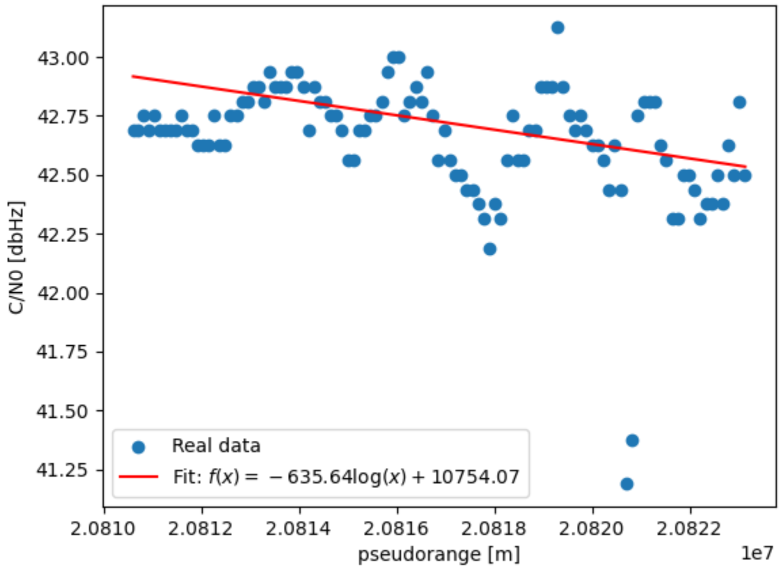


Figure 8.3: Fit of least squares method for C/N_0 approximation.

uninterrupted all the way to the satellite, the α coefficient is set to 0, however, with every semi-passable obstacle the coefficient increases by an arbitrary amount 1 (max 10).

Part III

Experiments

Chapter 9
Data collection

As any other simulator, the one implemented as a part of this thesis needed to be tested against real data. There were several experiments that were based on real life scenarios with focus on the extreme cases, where GNSS receivers usually have problems. All experiments were conducted on the in-block yard at Czech Technical University in Prague at Charles Square. This chapter focuses on the data collection and the description of the extreme cases. The comparison itself is described in detail in the next chapter, chapter 10.

9.1 Real world

All the real data was collected by tracked mobile robot Charlie equipped with dual antenna GNSS module from Septentrio. Since the goal of these experiments is to compare data simulated from one receiver only, there is no need to collect the data with two antennas, hence only the main antenna was mounted on the robot (as seen in Figure 9.1 in white circle). There was just another 1 experiment, where the robot had clear vies at the sky, however, at that place robot repeatedly shown higher uncertainty in its position. An idea behind this second placement was to uncover what causes these uncertainties.

During remaining three experiments, robot always had at least partial obstruction in its view of sky. In Figure 9.2 the setup for these 3 scenarios is presented. In the first 2, the robot's view is partially and then fully obstructed by a tree, which could present unpredictable satellite tracking. The last figure shows the most challenging setup, where the robot's antenna is placed under thick concrete ledge. It is expected that under these conditions, robot should be able to see only few satellites if any.

Data measurements are collected from the Septentrio module in special Septentrio binary format (SBF). For further comparisons, these data files are converted to RINEX observation files using Septentrio SBF Converter. This piece of software is part of the RXTools package as described in section 5.3. Several settings can be changed for the outputted RINEX file. For these experiments only the RINEX version is set to 3.04 in order to correspond to the version of files from the simulator.

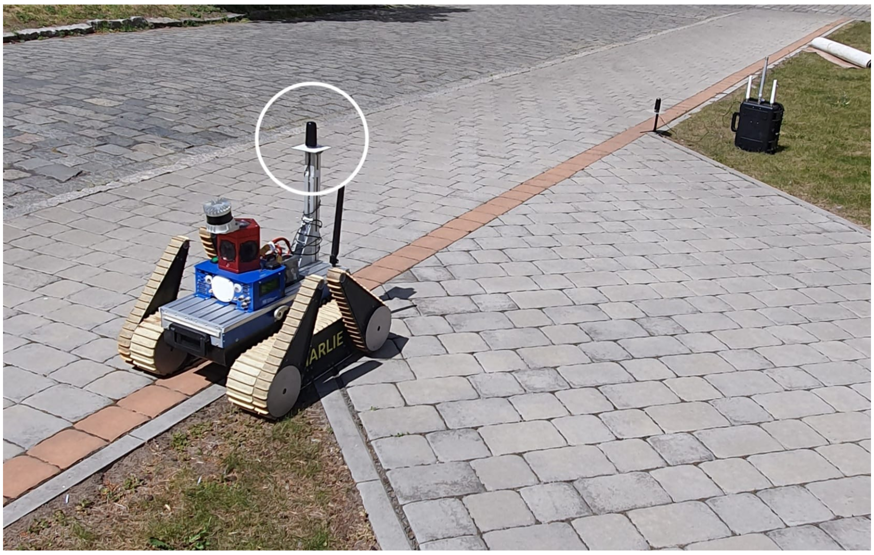
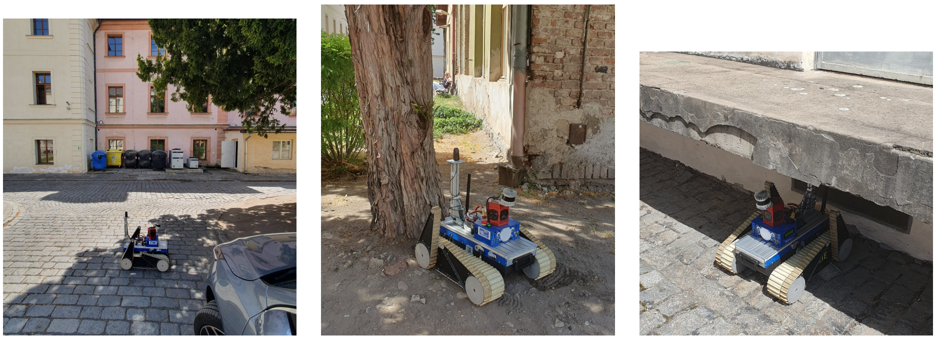


Figure 9.1: Robot with antenna and with clear view at sky.



(a) : Partially under a tree **(b) :** Fully under a tree **(c) :** Under a ledge

Figure 9.2: Real world setup for the experiments

9.2 Simulation

For the simulation part of the experiments, robot's starting position was configured to be on the same place as in the real world. A virtual replica of the yard at Charles Square was modeled and placed in Gazebo world. Origin of this world was transformed to be in ECEF coordinates at the same place as the real yard is. Robot's position inside this model ENU coordinate frame is set by ROS parameters. Estimated position where exactly robot should be placed was obtained from previous geodetic measurements conducted on the yard. The height coordinate of the simulated receiver is set to 70 centimeters above the model surface, similar to the antenna in the real world. For every experiment, starting time of the real data collecting is set as a ROS parameter for the simulator. Length of the simulation was set to equal the real world measuring.

Setup for the experiments with clear view at the sky is in Figure 9.3.

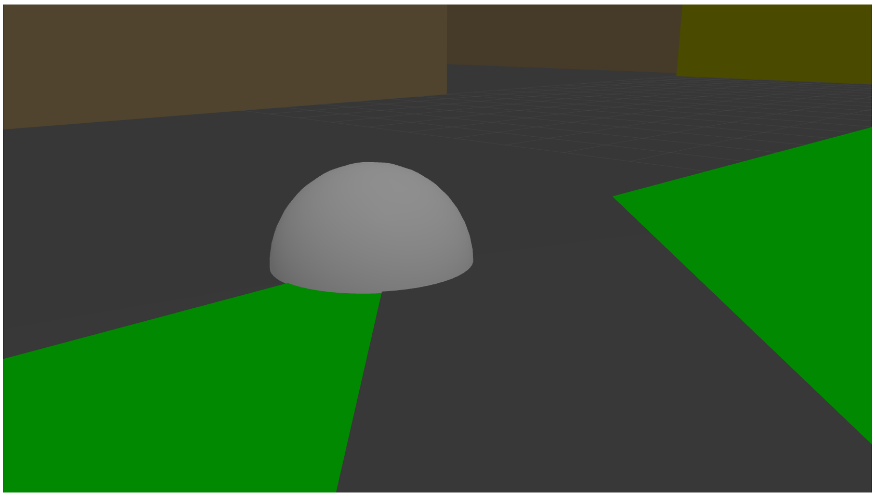
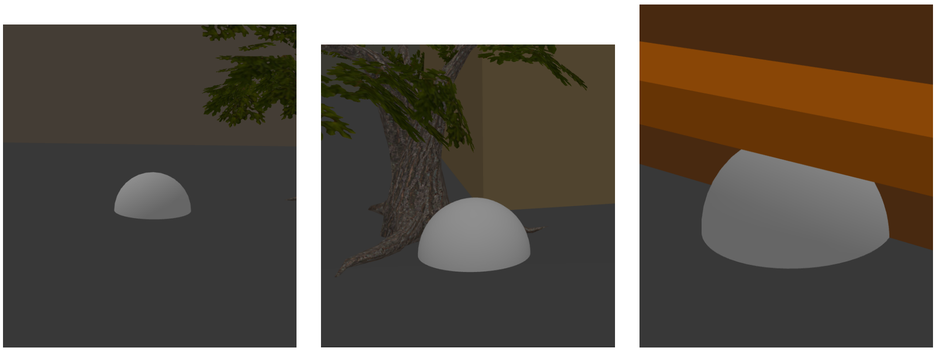


Figure 9.3: Setup for receiver simulation in Gazebo.



(a) : Partially under a tree **(b) :** Fully under a tree **(c) :** Under a ledge

Figure 9.4: Simulation setup for the experiments

Virtual receiver is modeled as a sphere just for visualization. Simulations with obstructed view in Figure 9.4 place the receiver under obstacles of variable types. Wall kind obstacles completely blocks the ray cast between satellite and receiver. However, a tree obstacle can let the ray through, even though it decreases the signal strength for each collision.

RINGO software for multi-GNSS pre-processing is used to read the generated RINEX files for interactive visualization. RINGO RINEX viewer is used to generate HTML files, that can be open in web browser to easily visualize and compare collected data. Another RINGO tool can be used to convert the RINEX observation data into CSV data format.[52]

Chapter 10

Comparison

Observation data from simulation as well as from the Septentrio module were converted into RINEX format and subsequently into HTML and CSV using the RINGO software.

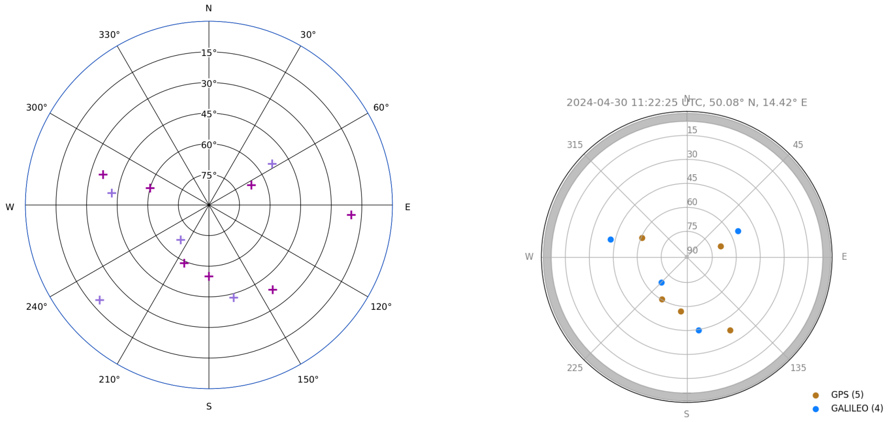
10.1 Number of satellites

Comparison of number of visible satellites and their general positions were conducted by visual inspection of the sky views. Real data from the Septentrio module were read by SBF Analyzer from the RxTools toolbox and sky view was generated. Data from the simulator shows only GPS and Galileo satellites, because the GNSSTK library has some issues with time conversion in other systems. However, from the sky views in Figure 10.1 and in Figure 10.2, it is clear that the simulator can see same satellites in the same positions as the real module. As expected, less satellites are marked as visible in the Gazebo simulation, because no reflected or diffracted signals are taken into consideration.

For the simulation under the wall ledge, no satellites were outputted as visible. In this case, any comparison with the real data can not be made. Real Septentrio module was able to capture some satellites, because it can catch even highly diffracted signals from the ground.

10.2 Observations

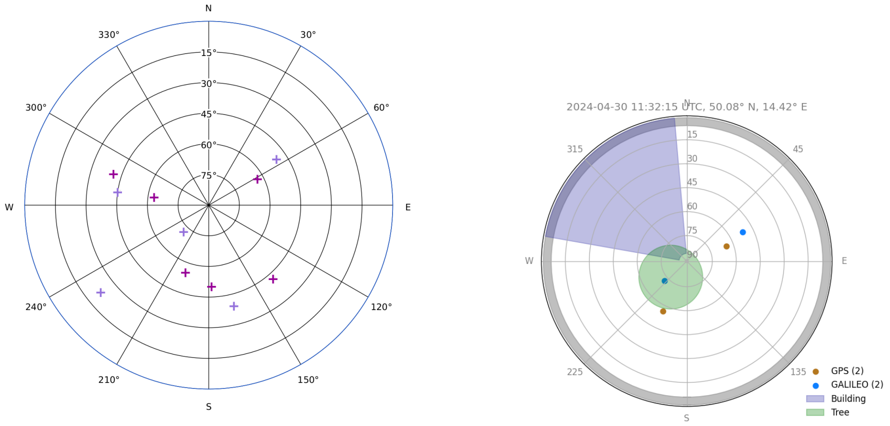
From the experiments with clear view or obstructed by the tree, real and simulated data were converted into HTML and CSV formats. The HTMLs can be visualized in web browser and can undergo visual comparison, however for this part, numeric comparisons of both real and simulated data were made. All following figures with comparisons are from the experiment with clear sky view. Other experiments with sky view obstructed by the tree follow the same principles and display similar results. Pseudoranges obtained from the simulation running under clear sky view are not exactly the same as from the real GNSS module, however, this offset can be attributed to the clock offset of the Septentrio module. In Figure 10.3 and Figure 10.4, it can be seen that



(a) : Real data from Septentrio. Dark red - GPS, light purple - Galileo

(b) : Data from the simulator

Figure 10.1: Sky views for the experiment with clear view



(a) : Real data from Septentrio. Dark red - GPS, light purple - Galileo

(b) : Data from the simulator

Figure 10.2: Sky views for the experiment under the tree

the difference between pseudoranges is the same for each displayed satellite. The rising tendency of the differences in time is caused by the drift in the clock of the real receiver.

Additional comparisons of doppler shift and carrier phase follow the same principle. The differences between simulated value and real value are the same for all visible satellites (Figure 10.6). This can be attributed to the clock offset of the receiver. Since the simulator is starting measuring phase cycles from 0, the comparison is done only after shifting the real measurements to start from 0 as well. Final comparison is plotted in Figure 10.5.

Comparison of Doppler shifts can be seen in Figure 10.7 and Figure 10.8. Even though, the real GNSS module produced more complex observations of Doppler shift, it is clear that the shift between simulated and real system is the same. This is due to the simplicity of Doppler shift calculation in the simulator and the more complex task of centering of received signal by

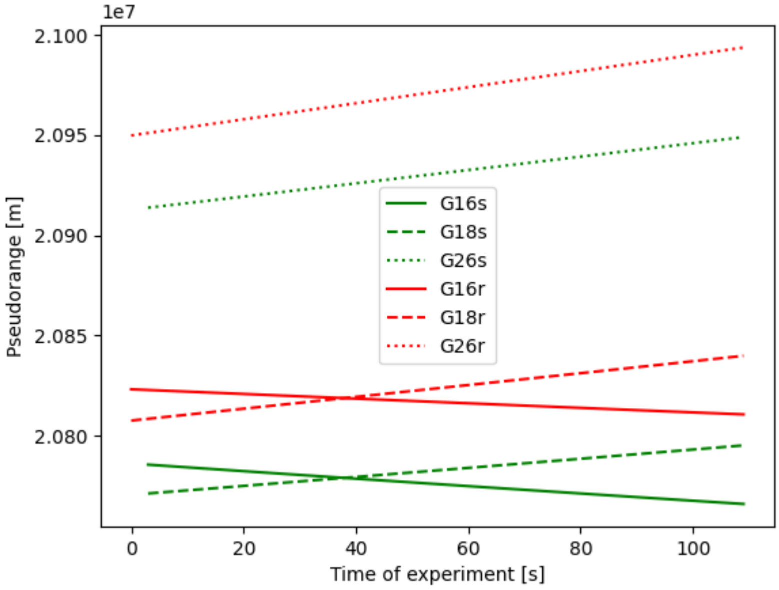


Figure 10.3: Comparison plot of pseudoranges from simulation and from real GNSS module for 3 visible GPS satellites

correlation in the real system.

Carrier to noise (C/N_0) comparison is in Figure 10.9, where the simulated signal strength is clearly not detailed enough as the values from the real module. This can be explored in the future improvements on this project.

10.3 Position

The best telling rating of the simulator is to compare obtained position to the position acquired from the real module, as well as the ground truth acquired by geodetic GNSS receiver. Obtained positions and calculated differences are depicted in Table 10.1. To calculate the positions, GNSSTK class PRSolution was used. This class takes pseudoranges to at least 4 satellites, navigation ephemerides and time of observation on its input and calculates solution for ECEF position of the receiver and its time offset to used GNSS times.

Positions in Table 10.1 were computed from pseudoranges to 7 satellites (4 GPS and 3 Galileo) observed during the clear sky view experiment. Smaller distance between simulated position and ground truth can be attributed to the sterile nature of the simulator as well as to the initial position of the virtual receiver that could be shifted in beneficial direction. However, when converted to latitude, longitude and altitude (LLA), the only difference in position are in altitude (Table 10.2). Visual comparison of all 3 positions with covariance can be seen in Figure 10.10.

For the experiment where the sky view is partially obstructed by the tree, position calculated from the data collected by real module shifts rapidly up

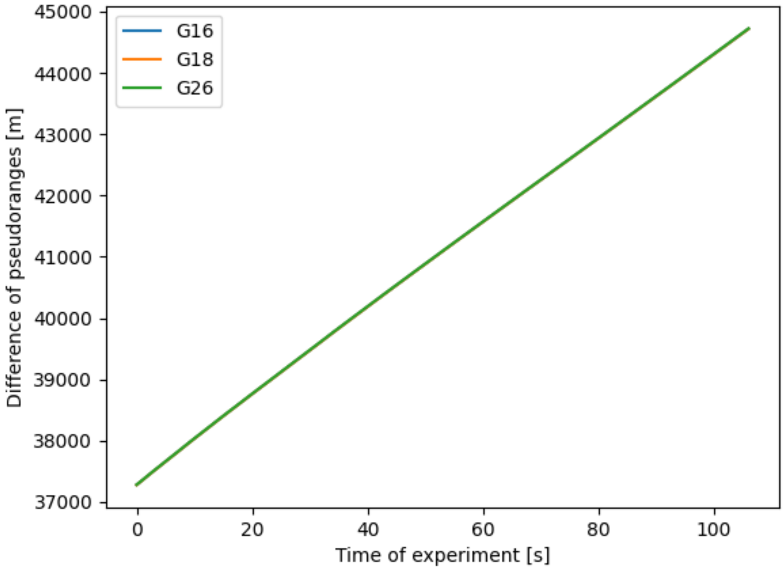


Figure 10.4: Differences between pseudoranges from simulation and from real GNSS module for 3 visible GPS satellites

	Ground truth	**Simulator**	**Real module**
ECEF pos. [m]	3972335 1021247 4868439	3972333 1021245 4868439	3972339 1021248 4868440
Diff. GT [m]	0	2.304	8.931
Diff. Sim [m]	2.304	0	9.890
Diff. Real [m]	8.931	9.890	0

Table 10.1: Obtained positions in ECEF coordinates and differences between them

to 25 meters from the ground truth. Simulator is able to keep the distance to the ground truth position in the scope of few meters. This is caused by the function of the simulator to liter out obstructed satellites. Although, the real module is able to track more satellites, the obtained position can be degraded by lower quality signals. During other two experiments (fully under the tree and under the wall), the simulator was not able to track enough satellites to compute its position.

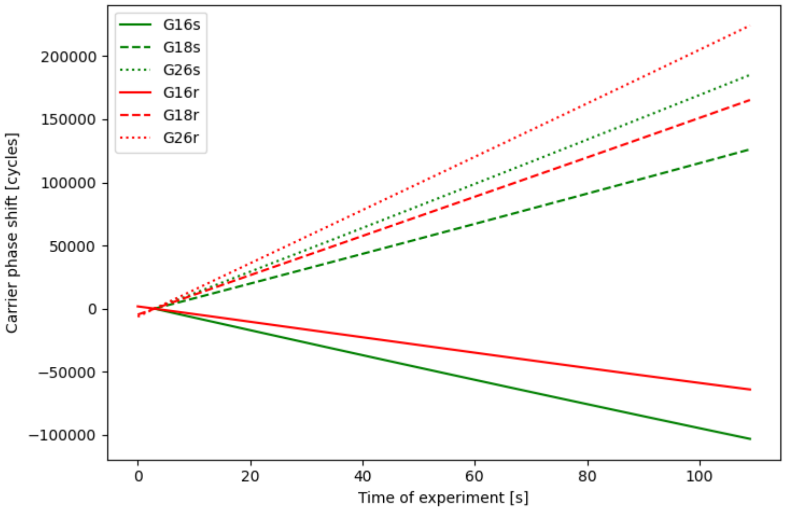


Figure 10.5: Comparison plot of carrier phase measurements from simulation and from real GNSS module

Position	Latitude [deg]	Longitude [deg]	Altitude [m]
Ground truth	50.07640222	14.41791814	249.735
Simulator	50.07642	14.41791	**248.831171**
Real module	50.07642	14.41793	**258.598459**

Table 10.2: Positions in latitude, longitude, altitude

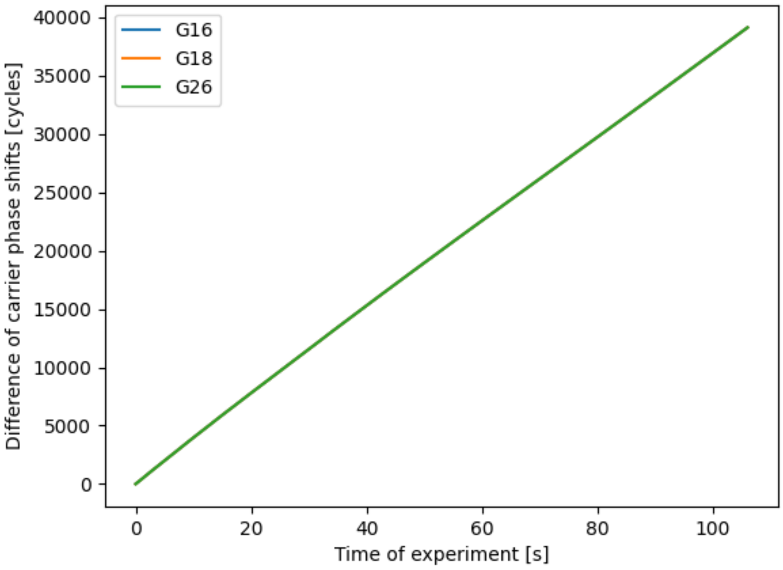


Figure 10.6: Differences of carrier phase measurements from simulation and from real GNSS module

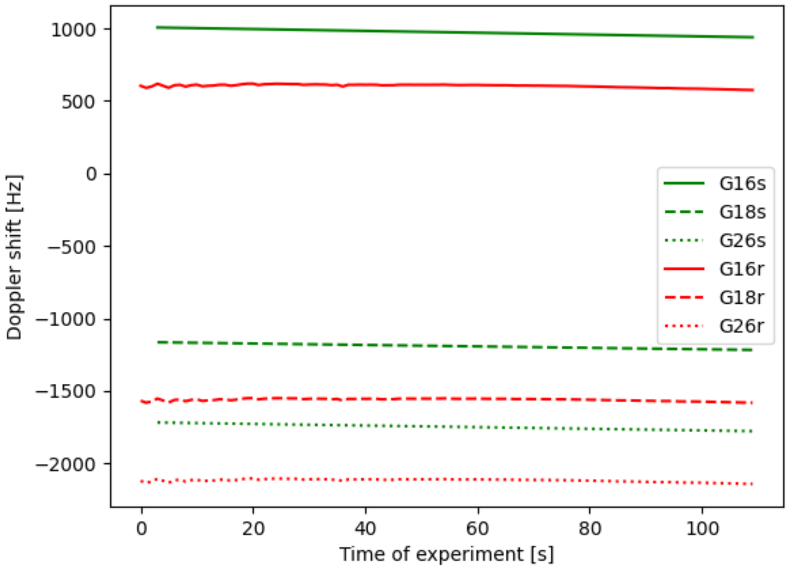


Figure 10.7: Comparison plot of Doppler shifts from simulation and from real GNSS module

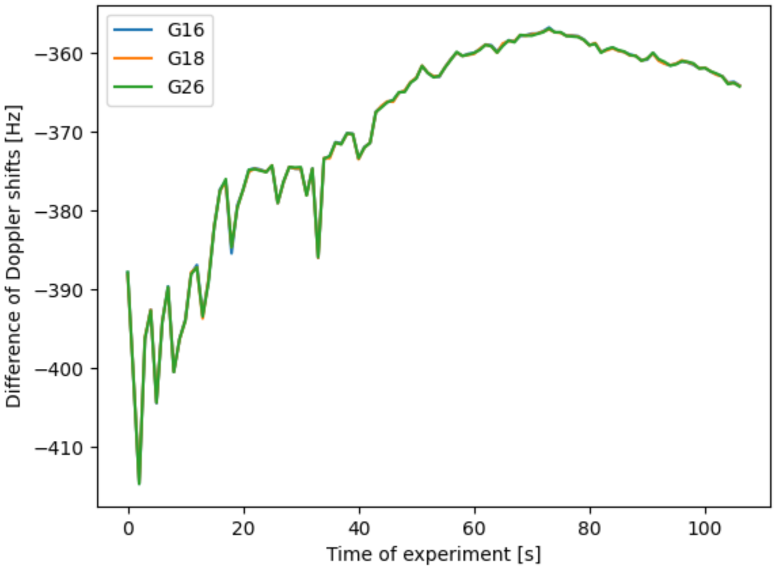


Figure 10.8: Differences of Doppler shifts from simulation and from real GNSS module

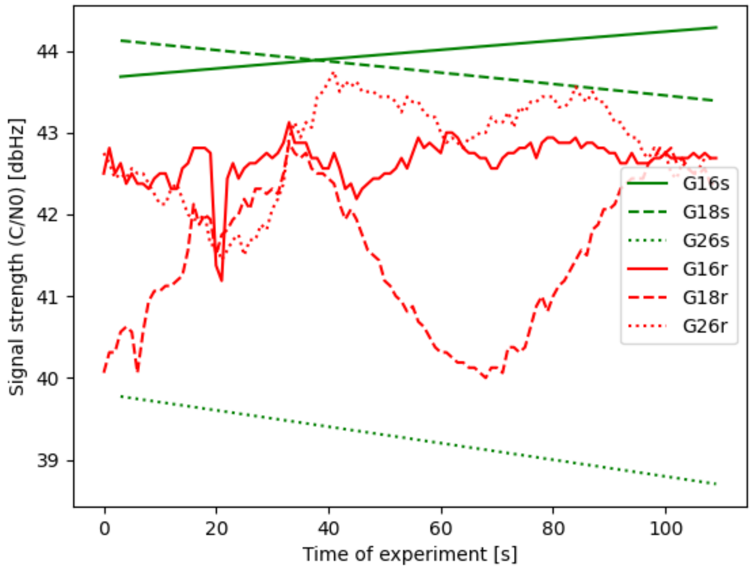


Figure 10.9: Comparison plot of C/N$_0$ from simulation and from real GNSS module

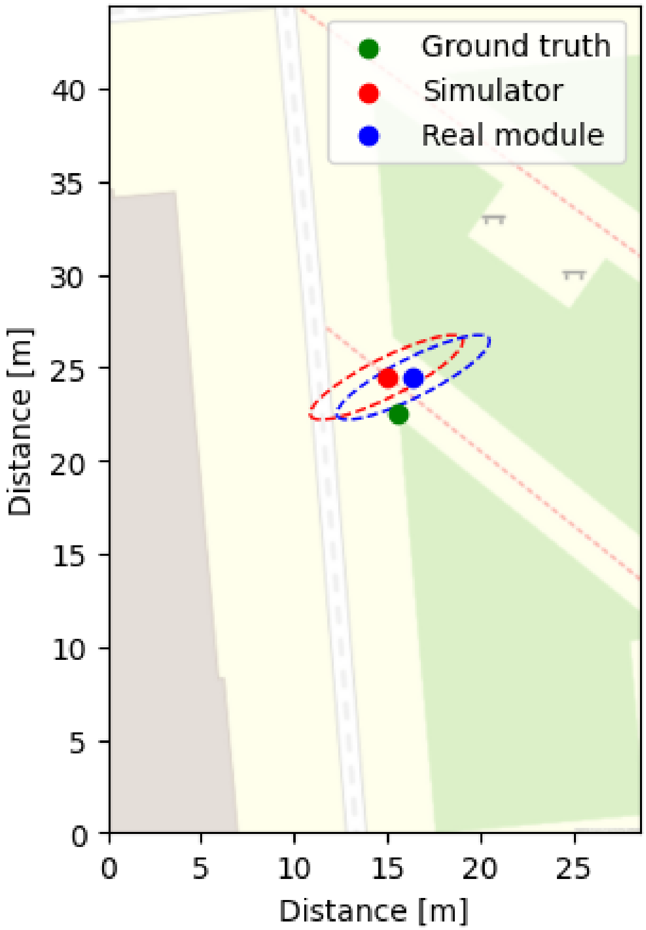


Figure 10.10: Position visualization using Open Street Map with ground truth, measurements and covariance.

Chapter 11
Conclusion

Visibility based GNSS receiver simulator was proposed with focus on satellite visibility. The simulator utilises open source C++ library GNSS Toolkit and free data center BKG in Germany. Navigation file downloaded from the database is used for obtaining satellite positions during the simulation. Emphasis is placed on time of flight consideration and the Earth's rotation when computing the satellite position and pseudorange.

Main advantage lies in the use of Gazebo simulator connected to the main simulation node via Robot Operation System (ROS). This implementations presents great opportunity for inclusion of this simulator to many robotic platforms. Gazebo simulator contains model of simulated environment, for example yard at CTU at Charles square. Supply of models for other environments is up to the user of this simulator. In order to simplify initialization of the virtual receiver, the whole Gazebo world is translated and transformed to the real coordinates of the environment. Visibility rays from the receiver simulated position to the calculated satellite position are cast and based on collisions (if any) satellite visibility is decided. There are two types of obstacles, that can cause satellite's loss of sight. Hard impenetrable obstacles like ground or walls completely stops the visibility ray, however, partially passable obstacles like trees can let the signal through but highly degraded.

Output of this simulator are calculated observations stored in RINEX observation file. The most important observation, pseudorange, is the calculated distance the signal travel from satellite to the receiver. Carrier phase measurement and Doppler shift from the carrier frequency are other observations calculated from the pseudorange and from the satellite velocity respectively. Simulator calculates the final observation, Carrier to noise (C/N_0), based on the pseudorange and number of semi-passable obstacles in the signal's way. From these observations, position of the virtual receiver could be estimated, as long as there are enough visible satellites. This obtained position is comparable to the the position obtained by the real life GNSS module Septentrio.

11.1 Future improvements

Right now the simulation does not consider the atmospheric effects on the signal propagation. Ionospheric corrections should be downloadable from data centers like the ones described in this thesis and model of the troposphere could be implemented with the use of the GNSSTK library. Implementation of these influences should have effect on more precise calculations of the pseudorange measurements.

Other improvements could be made to the virtual environment in Gazebo, more detailed models of objects could be used. Mainly, the free example models of trees that come with Gazebo are insufficient for more complex simulations. Better tree models could be downloaded from free databases of 3D models or modeled in 3D modeling tools.

Additionally, better way to estimate signal strength (C/N_0), could be achieved by utilisation of Fresnel zone, which is an ellipsoid between the transmitter and receiver, where most of the signal strength lies. This could be managed either by sending more visibility rays and calculate the percentage of them that collide or even by modeling the Fresnel ellipsoid itself and compute the obstructed volume and change the C/N_0 accordingly.

Bibliography

[1] L. Jurišica, F. Duchoň, D. Kaštan, and A. Babinec, “High precision GNSS guidance for field mobile robots”, *International Journal of Advanced Robotic Systems*, vol. 5, no. 9, Nov. 2012, ISSN: 1729-8814. DOI: 10.5772/52554. [Online]. Available: https://doi.org/10.5772/52554.

[2] P. J. G. Teunissen and O. Montenbruck, Eds., *Springer Handbook of Global Navigation Satellite Systems*. Springer International Publishing, 2017, ISBN: 978-3-319-42926-7. [Online]. Available: https://doi.org/10.1007/978-3-319-42928-1.

[3] Dr. R. Thirsk, *An introduction to GNSS*. NovAtel, 2023, ISBN: 978-0-9813754-1-0. [Online]. Available: https://www.calameo.com/read/0019157965c9cf7056211.

[4] Dr. R. Thirsk, *Gnss architecture*, Reproduced as Figure 2 in [3], 2023.

[5] R. Lea, “What is a geosynchronous orbit?”, *Space.com*, Dec. 2022. [Online]. Available: https://www.space.com/29222-geosynchronous-orbit.html.

[6] prof. Ing. M. Štroner Ph.D, *Globální navigační satelitní systémy (GNSS)*, 2022. [Online]. Available: https://k154.fsv.cvut.cz/wp-content/uploads/2022/01/GNSS_obs.pdf.

[7] “Galileo satellites”. (2024), [Online]. Available: https://www.esa.int/Applications/Satellite_navigation/Galileo/Galileo_satellites.

[8] “Fundamental PNT service”. (2024), [Online]. Available: http://www.csno-tarc.cn/en/system/constellation.

[9] “Overview of the Quasi-Zenith Satellite System (QZSS)”. (2024), [Online]. Available: https://qzss.go.jp/en/overview/services/sv01_what.html.

[10] “IRNSS - Indian Regional Navigation Satellite System”. (2023), [Online]. Available: https://www.ursc.gov.in/navigation/irnss.jsp.

[11] “IRNSS Programme”. (2023), [Online]. Available: https://www.isro.gov.in/IRNSS_Programme.html.

[12] "GNSS Constellations, Radio Frequencies and Signals". (2024), [Online]. Available: https://www.tallysman.com/gnss-constellations-radio-frequencies-and-signals/.

[13] "GNSS signal". (2014), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=GNSS_signal.

[14] J. C. Ramón, *GPS Signals*, Reproduced as Figure 1 in [13], 2014.

[15] "Conventional Celestial Reference System". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Conventional_Celestial_Reference_System.

[16] "Conventional Terrestrial Reference System". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Conventional_Terrestrial_Reference_System.

[17] "Reference Frames in GNSS". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Reference_Frames_in_GNSS.

[18] Mike1024, *Coordinate systems diagram*, Reproduced as Figure in https://commons.wikimedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_relationships.svg, 2010.

[19] "GNSS Receivers General Introduction". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=GNSS_Receivers_General_Introduction.

[20] "Generic Receiver Description". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Generic_Receiver_Description.

[21] "The Different Types of GNSS Receivers Available on the Market Today". (2020), [Online]. Available: https://castnav.com/the-different-types-of-gnss-receivers-available-on-the-market-today/.

[22] "Antennas". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Antennas.

[23] "Front End". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Front_End.

[24] "Baseband Processing". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Baseband_Processing.

[25] T. Ferreira, *PRN Correlation*, Reproduced as Figure 2 in [24], 2011.

[26] T. Ferreira, *Receiver channel diagram*, Reproduced as Figure 1 in [24], 2011.

[27] "System Design Details". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=System_Design_Details#Applications_Processing.

[28] "What is the difference between Standalone, Differential GPS and Real-time Kinematic?" (2023), [Online]. Available: https://support.marxact.com/article/112-what-is-the-difference-between-standalone-differential-gps-and-real-time-kinematic.

[29] "Real Time Kinematic (RTK)". (), [Online]. Available: https://www.vboxautomotive.co.uk/index.php/en/component/content/article/2-products/109-real-time-kinematic-rtk.

[30] "What is SBAS?" (2022), [Online]. Available: https://www.euspa.europa.eu/eu-space-programme/egnos/what-sbas.

[31] M. Chivers, "Differential GPS Explained", *ArcUser*, 2003. [Online]. Available: https://www.esri.com/about/newsroom/arcuser/differential-gps-explained/.

[32] "What is D-GPS and how does it work?" (2024), [Online]. Available: https://www.spectx.nl/blog/what-is-dgps/.

[33] Dr. R. Thirsk, *Real-Time Kinematic (RTK)*, Reproduced as Figure 43 in [3], 2023.

[34] Emlid. "How RTK works", YouTube. (Feb. 2022), [Online]. Available: https://www.youtube.com/watch?v=ieearzWTCZw.

[35] "Interfaces and Protocols". (2013), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Interfaces_and_Protocols#Almanac_Related_Formats.

[36] "RINEX - The Receiver Independent Exchange Format". (2018), [Online]. Available: https://files.igs.org/pub/data/format/rinex304.pdf?_gl=1*kb2e0w*_ga*MTYyODUwMDkzNi4xNzExNjI3NTY0*_ga_Z5RH7R682C*MTcxNjU2NDkyMS4xOC4wLjE3MTY1NjQ5MjEuNjAuMC4w&_ga=2.24922106.1346401969.1716564922-1628500936.1711627564.

[37] "Two-body Problem". (2011), [Online]. Available: https://gssc.esa.int/navipedia/index.php?title=Two-body_Problem.

[38] K. Zhang, *Keplerian elements and satellite orbit*, Reproduced as Figure 1 in [53], 2006.

[39] *Observation Code Components*, Reproduced as Table 3 in [36], 2018.

[40] "Choosing a GNSS Simulator". (2019), [Online]. Available: https://www.spirent.com/assets/eb_choosing-a-gnss-simulator.

[41] M. Pecka, *Gnss-info*, https://github.com/ctu-vras/gnss-info, 2023.

[42] "RxTools - User Manual". (2023), [Online]. Available: https://www.septentrio.com/system/files/support/rxtools_v23.0.1_user_manual.pdf.

[43] "About the Satellite Database". (2023), [Online]. Available: https://satdb.ethz.ch/about/.

[44] "CelesTrak". (2022), [Online]. Available: https://celestrak.org/webmaster.php.

[45] “Global Navigation Satellite System (GNSS) Overview”. (2024), [Online]. Available: https://cddis.nasa.gov/Techniques/GNSS/GNSS_Overview.html.

[46] A. Shihadeh, *Openpilot*, https://github.com/commaai/openpilot, 2023.

[47] “The BKG GNSS Data Center”. (2024), [Online]. Available: https://igs.bkg.bund.de/.

[48] D. Barber, *Gnsstk*, https://github.com/SGL-UT/gnsstk, 2024.

[49] “How it works”. (2024), [Online]. Available: https://gazebosim.org/home.

[50] “Beginner: Overview”. (2014), [Online]. Available: https://classic.gazebosim.org/tutorials?tut=guided_b1.

[51] “Plugins 101”. (2014), [Online]. Available: https://classic.gazebosim.org/tutorials?tut=plugins_hello_world.

[52] “RINGO: A RINEX pre-processing software for multi-GNSS data”. (2023), [Online]. Available: https://terras.gsi.go.jp/software/ringo/en/.

[53] Z. Jason, K. Zhang, R. Grenfell, and R. Deakin, “GPS Satellite Velocity and Acceleration Determination using the Broadcast Ephemeris”, *THE JOURNAL OF NAVIGATION*, vol. 59, pp. 293–305, May 2006. DOI: 10.1017/S0373463306003638.