ANALYSIS OF THE FEASIBILITY OF GNSS APPLICATIONS FOR HARSH ENVIRONMENTS.

Politecnico di Torino

1859

Master thesis of Mechatronic Engineering

Carlos Saro De Aldecoa

S275813

Supervisor:

Prof. Paolo Dabove

Co-supervisor:

Vincenzo Di Pietra

Abstract

This project is about an analysis of the behavior of GNSS systems for its possible inclusion in applications related to mechatronics. To carry out this research, only open-source programs, mobile phones and low-cost portable receivers will be used.

With this approach, citizens can have powerful tools to collect data and generate information without depending on external sensors. This procedure, called citizen sensing, gives citizens information about the quality of their living environment.

By using this innovative method, the possibilities of introducing GNSS technology in mechatronic systems can increase due to the data collection with solutions that are already in use by most of the population (like mobile phones).

The procedure will start in selecting the receivers that best fit the defined requirements and collect some sets of data with them in different environments. Then study different GNSS algorithms of post-processing, verify which one of them have a better behavior and compare their performance with real data.

Through these comparisons, errors can be calculated and classified depending on the range of precision that they have. The idea is to be able to differentiate the quality of the sensors used during the project and choose the post-processing methods that best fit for these navigation solutions.

Once all the computation is done, the final step is to select which one of the receivers, environment and algorithms can make GNSS systems to be included in which mechatronic applications.

Contents

Figures

Tables

1. State of Art

Satellite positioning systems are increasing its importance in the recent years. There is no doubt in the multiple applications that this technology has in multiple fields, for example, aviation, maritime positioning, safety, etc...

According to the report "*Outcome of the European GNSS' User Consultation Platform*" of the European GNSS Agency, the installed devices for this aim will increase a lot in the next years. The metrics of this growth can be seen in the graphic below, which is also shared in the same report.

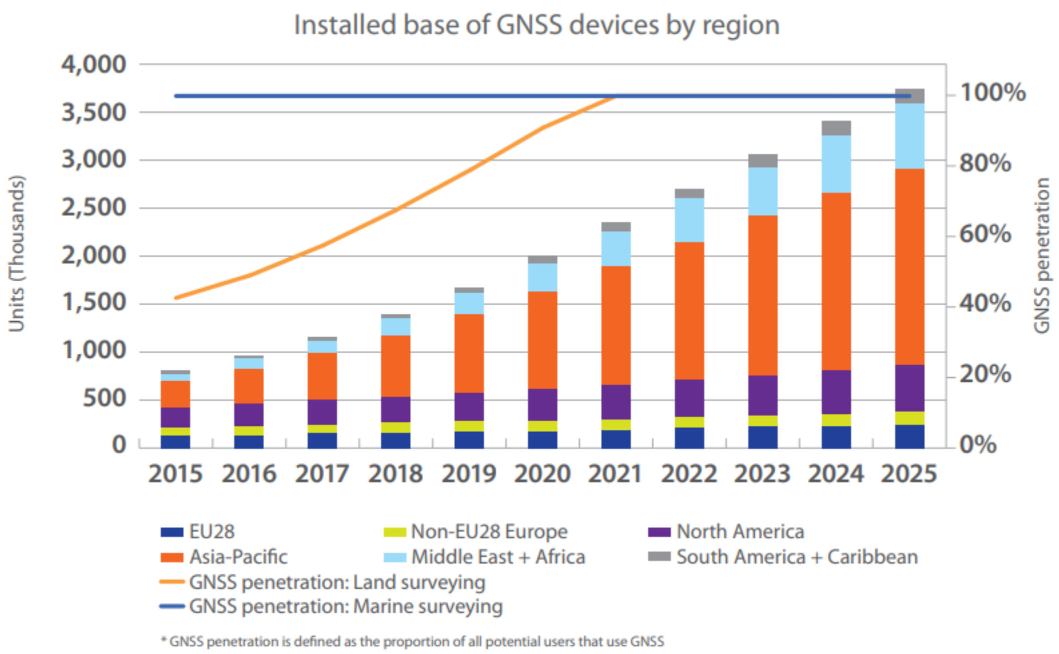


Figure 1 – Installed base of GNSS devices by region [1]

And that is why important organizations have been investing a lot in this technology in the recent years. In the article "*The economic impact on the UK of a disruption to GNSS*" of the company London Economics, it is assured that the UK government has invested almost £1.2bn (€1.5bn) since 2000 to develop the European GNSS infrastructure, with the aim of accomplish greater performance and resilience, and foster the lucrative downstream applications market, which, in their opinion, provides significant benefits to users and the rest of society.

But it also results interesting its importance in mechatronic systems. A large variety of modern devices require precise and global positioning for their correct functionality. This is the case for example of autonomous cars, in which accuracy in their location is essential. As the organization EUSPA (European Union Agency for the Space Program) publishes in their article "*Central role for robust GNSS in autonomous driving*", localization of autonomous vehicles will be enabled mainly by GNSS technologies.

In the same report, the engineer in Geodesy and Cartography Alberto Fernandez-Wyttenbach assures that although it will have to be complemented with other technologies, GNSS is a most important in this field. Another aspect that it's highlighted is the barrier that cost can suppose in commercial purposes.

The same happens with other devices like drones, since in them a precise navigation system is essential to assure total control. In the research "*GNSS Interference in Unmanned Aerial Systems*" of the organization Septentrio Satellite Navigation, it is highlighted how in case of losing the GNSS position, UAVs can be stabilized with their inertial sensor, but will be unable to navigate towards its landing spot without intervention of a human operator.

As previously said, an important point in the viability of using this technology in mechatronic systems, apart from its performance, is its cost. In the investigation "*Evaluation of Low-Cost, Centimeter-Level Accuracy OEM GNSS Receivers*" from the department of aerospace engineering and mechanics of the University of Minnesota, they demonstrated that receivers with a price between 500 and 1000 euros a centimeter-level accuracy can be achieved.

In this study, they also use GNSS receivers with lower costs, as is it the case of the model NV08C-RTK of NVS Technologies (around 450 €), the u-blox NEO-M8P (around 200 €) and the model S2525F8 of Skytraq (around 160 €). But with those devices, they could not achieve an accuracy of less than a meter.

In this thesis, a set of low-cost GNSS receivers will be analyzed with the idea of proving their validity on mechatronic systems. Regarding the importance of accuracy in this field, this could result very interesting in order to demonstrate in which conditions it may be feasible to consider this solution to solve the problem of precise location.

On the other hand, the price of all GNSS receivers used supposes an advantage and makes it easier to introduce them in any electronic device that need to be localized in real time, of course, if the desired range of precision is reached.

2. Objectives

This thesis is about to study the behavior of Global Navigation Satellite Systems in different environments. The principle aim of the research is to test a set of devices and be able to prove their validity for mechatronic applications by assuring a good behavior in terms of precise localization. For this purpose, several minor objectives have been set to achieve the desired result step by step.

Firstly, it is necessary to understand how GNSS systems work and some theory behind all the components that take part in them. It is also very important to study the most common issues in this field to try to avoid them as much as possible. Another important point to be treated is about the algorithms used to achieve a navigation solution.

Once the capabilities of GNSS systems are clear, a study of which conditions are necessary to meet to be able to introduce them in which mechatronic applications is required.

The next objective of the analysis relies on trying to replicate the conditions defined before by post processing real data collected with different low-cost GNSS receivers in different environments.

After achieving the navigations solutions by processing the input data, it is required to evaluate in detail those results and compare them with each other. At this point, it is also essential to determine whether the requirements settled are fulfilled or not.

With this strategy, the main goal can be achieved by reviewing the results of the last step. If the requirements defined are severe enough that assure an acceptable range of accuracy, every combination of measurements capable of achieving them is validated to be introduced in the mechatronic systems for which those requirements were specified.

One important issue that GNSS systems have is the bad behavior in indoor environments, where other positioning solutions are much more precise. In this case, all the possibilities are going to be outdoor to not deal with that problem.

An important application that can be interesting are autonomous vehicles. They can be also divided into the ones used to transport people in the same roads than normal cars, and another kind used to transport heavy appliances thought private roads or lands.

For the first application, as the security requirements are very restrictive and the means that will be in use for this project does not permit high-accuracy GNSS receivers, only the second application is realistic to target.

Another field that can include this kind of positioning systems are drones. As they are outdoor and most of them do not precise from too severe requirements in terms of accuracy. That is why a range of error of one or two meters can be good enough.

GNSS positioning is also interesting in air transport. Although in civil airplanes the necessary security cannot be assured, in other air vehicles like helicopters or Unmanned Aerial Vehicles (UAVs) these systems can be introduced.

3. Introduction

In this chapter, a brief introduction of all the theory behind GNSS systems is explained. The idea is to treat only the necessary terms for the aim of this thesis in a simple way. Starting from the logic used to compute a navigation solution and then continuing explaining the parts that conform the principal Global Navigation Satellite Systems.

3.1. GNSS Positioning

A Global Navigation Satellite System is formed of a constellation of Satellites Orbiting Earth. These satellites are continuously transmitting signals that are processed to determine user´s 3D position with global coverage.

There are several Global Navigation Satellite Systems. The first one being fully operational was de United States GPS (Global Positioning System) but nowadays there are some more. Russian GLONASS, Chinese BeiDou and European Galileo are the other examples of GNSS. [3]

3.2. Basic Principle

This chapter is thought to be a simplification of how GNSS work to be able to understand it with an easy explanation and then be able to continue from this base.

As previously said, the main goal of those systems is to determine user 3D positioning. The way of calculating it is based in measuring travel time of at least four different signals. With this data and knowing the coordinates of the satellites, the user can determine his position.

The next image represents the logic of this system in 2D dimension because it is simpler to represent it with a scheme. It can be seen that only three satellites are needed in this case.

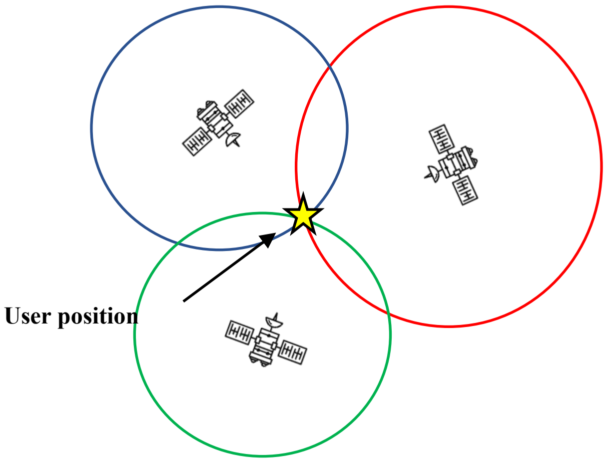


Figure 2 – GNSS basic principle

For this purpose, satellites transmit different signals. Roughly, these signals can be divided in two, Navigation Data and Observation Data.

Navigation Data (also known as ephemeris) is meant to give information about satellite coordinates. On the other hand, Observation Data is the one used to calculate the travel time of the signal and be able to approach user position.

3.3. GNSS Architecture

A GNSS system is formed of three main segments: the space segment, the control segment, and the user segment.

3.4. Space Segment

The space segment consists of GNSS satellites, orbiting about 20,000 km above the earth. Each GNSS has its own constellation of satellites, arranged in orbits to provide the desired coverage, as illustrated in the next figure.

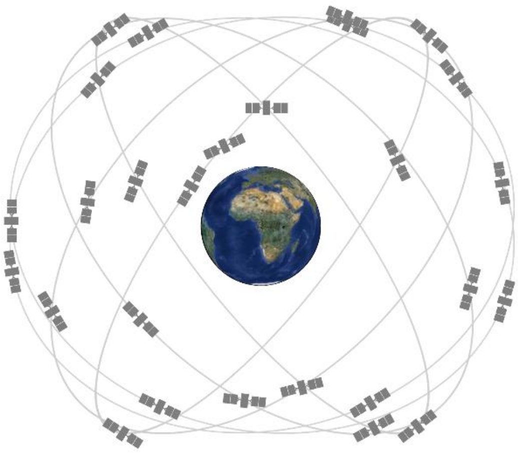


Figure 3 - GNSS Space segment [12]

The main function of this segment is to transmit observation signals (code and carrier phase) and to store and broadcast navigation messages uploaded by the control segment. To be able to achieve the desired global coverage, user must have at least four satellites in view simultaneously from any point of the earth at any time. [10]

As previously said, each GNSS has its own constellation. The main characteristics of some of them will be described below.

GPS satellites are placed in the four slots that divide each six equally orbital plane. The orbit has an altitude of 20200 km and an inclination of 55° relative to the equator. The eccentricity of less than 0.02 makes the orbit almost circular and with a nominal period of 11 hours, 58 minutes, and 2 seconds. [12]

GLONASS satellites are deposited in three orbits equally spaced in each of them. In this case the inclination relative to the equator is of about 64.8° and the altitude 19100 km what makes a nominal period of 11 hours 15 minutes and 44 seconds. These constellations repeat the same geometry every eight sidereal days. [16]

Galileo satellites are placed in three orbital planes inclined 56° with respect to the equator. They take 14 hours, 4 minutes, and 45 seconds to orbit earth. Their altitude is 23222 km and the eccentricity 0.002. [3]

3.4.1. Satellites

Satellites are a key part of the Space segment. They have various mechanism and structures that puts them in orbit, make them able to communicate with the control segment and broadcast signals to receivers. As an accurate satellite time is a critical factor to avoid big errors in the position solution, they are equipped with atomic clocks.

To explain the main characteristics of each of them, the easiest way in to divide in constellations.

In the case of GPS satellites, they are divided into Blocks, from Block I to the newest Block III. Starting from the Block I, they weighted around 850 kgs and were planned to operate for 4,5 years. They were able to provide positioning during 3 or 4 days without contacting with the control segment. [10]

The second generation of satellites include Block II, Block IIA, Block IIR, Block IIR-M and Block IIF. They weight between 1500 and 2000 kgs, had a longer average lifetime (from 7,5 to 15 years) and provide positioning for at least 180 days with no contact with the control segment.

Block III satellites importantly improve navigation messages with a better interoperability with other satellites and control segment.

Regarding GLONASS satellites, these can be divided into generations. The zero generation or prototypes were placed in orbit in 1982. But the first true GLONASS satellites (First Generation) were not launched until 1985. Their mass was of about 1250 Kg and they were expected to operate from 2 to 3,5 years (although many of them reached 4,5 years). [10]

It was in 2001 when the first second generation of GLONASS satellites was sent into orbit. This new development introduced a larger lifetime (up to 7 years) and some improvements like better propulsion system or the addition of caesium clocks (which are more stable). In the third generation, these satellites reduced their weight to 750 Kg and enlarged lifetime to 10-12 years.

Galileo satellites were launched in different phases, the experimental one was done from 2005 until 2008. It included satellites GIOVE-A and GIOVE-B and its objective was to validate technologies an assured the feasibility of the project. [10]

After this, it came the Galileo IOV Phase, which was aimed to prove space, ground and user segment. In this phase four operational satellites complemented the other two experimental ones. They weight around 700 Kg and were equipped with a powerful transmitter to broadcast precise navigation data. [3]

Galileo started offering Early Operational Capability (EOC) on 15 December 2016, providing initial services with a weak signal. Then, they reached Full Operational Capability (FOC) in 2019. [10]

The full Galileo constellation it is planned to consist of 24 active satellites, which is expected by 2021. The next generation of satellites will begin to become operational after 2025 and replace the older ones, which could then be reused for backup capabilities.

3.5. Control Segment

The main objective of the control segment (or ground segment) is to assure the proper operation of the global navigation satellite system. It is capable of this duty by doing some important functions:

- Correct and keep the proper time scale of the system.
- Predict and update navigation data and clock evolution.
- Control the status of the constellation.

3.6. User Segment

User segment includes the equipment of people who receive GNSS signals and uses them to compute their position and time information. [13] There are a lot of devices that can receive GNSS data, some examples can be phones, GNSS Stations, Computers, GNSS High accuracy receivers, u-blox, etc.

3.7. GNSS Signals

The GNSS satellites continuously transmit signals in two or more frequencies in L band. These signals contain ranging codes and navigation data to allow the users to compute the travelling time from satellite to receiver and the satellite coordinates at any epoch. The principal components of these signals can be divided in:

Carrier: Radio frequency sinusoidal signal at a given frequency.

Ranging code: Sequences of 0s and 1s, which allow the receiver to determine the travel time of the signal from satellite to receiver. They are called Pseudo-Random Noise (PRN) sequences or PRN codes. [3]

Navigation data: A binary-coded message providing information on the satellite ephemeris (orbit elements or satellite position and velocity), clock parameters, almanac (with a reduced accuracy ephemeris data set), satellite health status, and other complementary information.

In the next figure it can be seen an example of the wave forms of a GPS signal in the L1 band.

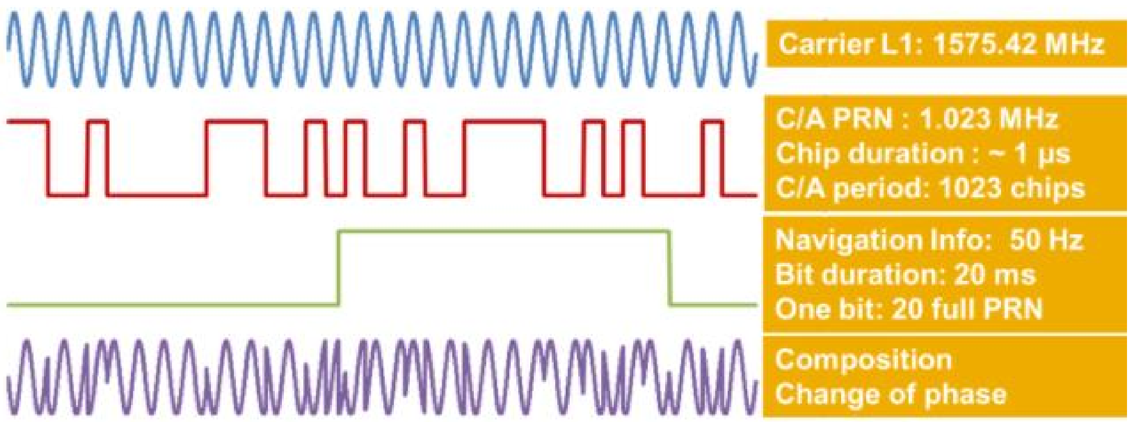


Figure 4 – GNSS signals [10]

About the allocation of frequency bands, they cannot be easily divided as some of them can be in the same range with a different purpose depending on the country. To understand and see the frequencies that are used worldwide for the GNSS signals, a graphic is shown below.

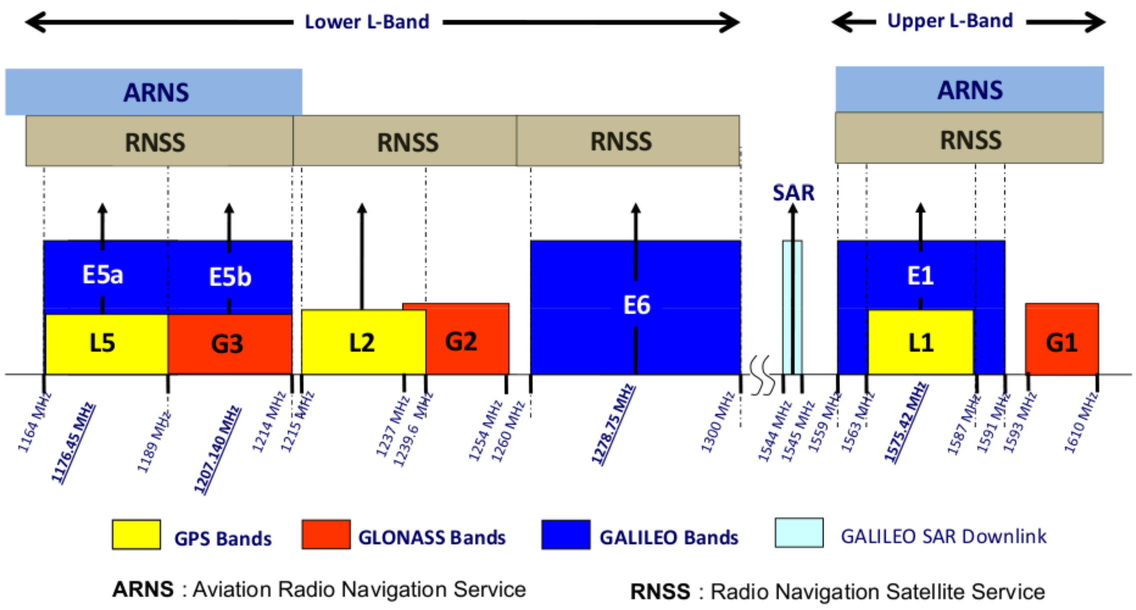


Figure 5 - GNSS Frequency bands [10]

3.8. RINEX files

As previously said the main goal of a navigation receiver is to compute a navigation solution. To calculate the position with those several measurements, one or more satellite constellations, are needed. All this intermediate data is used not only for the solution computation but also in a multitude of applications.

Due to the growing interest in all data measured by user segment receivers and stations, it was necessary to unify this data by creating standards and permit exchanging data in a common way between all the interested entities (such as companies, manufactures, universities or users).

Receiver Independent Exchange format (RINEX) is a data interchange format for raw satellite navigation system data. It was developed by the Astronomical Institute of the University of Berne for the exchange of GPS data to be collected during the first large European GPS campaign [7]. This development was based on the fact that most GNSS processing software used a mainly the next defined set of observables:

- The carrier-phase measurement at one or both carriers
- The pseudorange (code) measurement, equivalent to the difference of the time of reception and the time of transmission of the signal.
- The observation time being the reading of the receiver clock at the instant of validity of the carrier-phase and/or the code measurements. [8]

At the present time three major format versions have been developed:

- The original RINEX Version 1 presented at and accepted in 1989.
- RINEX Version 2 presented at and accepted in 1990, mainly adding the possibility to include tracking data from different satellite systems (GLONASS, SBAS).
- RINEX Version 3, currently under revision. It is planned to include the new global navigation satellite systems Galileo and BeiDou. [8]

3.8.1. Observation RINEX

The observations are the measurements made by receivers using the signals of the GNSS satellites. They include three fundamental quantities: Time, pseudorange and carrier phase but they also give information of other parameters like doppler and SNR (Signal noise ratio).

The structure of every observation RINEX files can be divided into Header and measurements, at the same time divided in epochs.

In the header there could be information about the RINEX version, the kind of data (observation or navigation), receiver, time of the measurements or the order that follow the measurements in each row of the file. An example of a RINEX version 2.11 header can be seen below. [14]

```
     2.11           OBSERVATION DATA    M (MIXED)           RINEX VERSION / TYPE
BLANK OR G = GPS,  R = GLONASS,  E = GALILEO,  M = MIXED    COMMENT
gLAB                gAGE                    17-MAR-10 12:14 PGM / RUN BY / DATE
EXAMPLE OF A MIXED RINEX FILE                               COMMENT
MRKR                                                        MARKER NAME
9080.1.34                                                   MARKER NUMBER
gAGE                UPC: Technical University of Catalonia  OBSERVER / AGENCY
THIS FILE IS PART OF THE gLAB TOOL SUITE                    COMMENT
FILE PREPARED BY: ADRIA ROVIRA GARCIA                       COMMENT
PLEASE EMAIL ANY COMMENT OR REQUEST TO: glab.gage @ upc.edu COMMENT
IR2200716006        ASHTECH UZ-12       CQ00                REC # / TYPE / VERS
482                 AOAD/M_T        NONE                    ANT # / TYPE
  4789028.4701   176610.0133  4195017.0310                  APPROX POSITION XYZ
        0.9030        0.0000        0.0000                  ANTENNA: DELTA H/E/N
     1     1                                                WAVELENGTH FACT L1/2
     1     2     3   G14   G18   G19                        WAVELENGTH FACT L1/2
     7    L1    L2    P1    P2    C1    S1    S2            # / TYPES OF OBSERV
    30.000                                                  INTERVAL
  2010     3     5     0     0    0.0000000     GPS         TIME OF FIRST OBS
  2010     3     5    23    59   30.0000000     GPS         TIME OF LAST OBS
     1                                                      RCV CLOCK OFFS APPL
    15                                                      LEAP SECONDS
    14                                                      # OF SATELLITES

   G07   815   815   815   815   815   815   815            PRN / # OF OBS
   G09   246   246   246   246   246   246   246            PRN / # OF OBS
   G12   687   687   687   687   687   687   687            PRN / # OF OBS
   G13   762   762   762   762   762   762   762            PRN / # OF OBS
   G15   454   454   454   454   454   454   454            PRN / # OF OBS
   G20   599   599   599   599   599   599   599            PRN / # OF OBS
   G21   636   636   636   636   636   636   636            PRN / # OF OBS
   G26   210   210   210   210   210   210   210            PRN / # OF OBS
   G31   874   874   874   874   874   874   874            PRN / # OF OBS
   G32   457   457   457   457   457   457   457            PRN / # OF OBS
   R11   907                     907   907                  PRN / # OF OBS
   R19   348                     348   348                  PRN / # OF OBS
   R23   936                     936   936                  PRN / # OF OBS
   S24   198                     198   198                  PRN / # OF OBS
                                                            END OF HEADER
```

Figure 6 – RINEX 2.11 Observation file header [14]

Every epoch of measurements reflects the year, week, day, hour, minute and second in which the signal is received. The time is identical for the phase and range measurements and is identical for all satellites observed in the same epoch.

The pseudorange is the distance from the receiver antenna measured in meters. This quantity equals the speed of light in vacuum times the apparent time travel of the code signal. This time is calculated as the difference of reception (receiver frame) and the time of transmission (satellite frame).

The carrier phase is measured in cycles (being a measurement on the beat frequency between the received carrier of the satellite signal and a receiver generated reference frequency). It changes in the same sense as the range. The phase observations between epochs must be connected by including the integer number of cycles.

In the observation data file, the pseudorange is measured in meters. There are three types C1 (C/A code or standard on the frequency L1), P1 (code P or Precise in L1) and P2 (code P in L2).

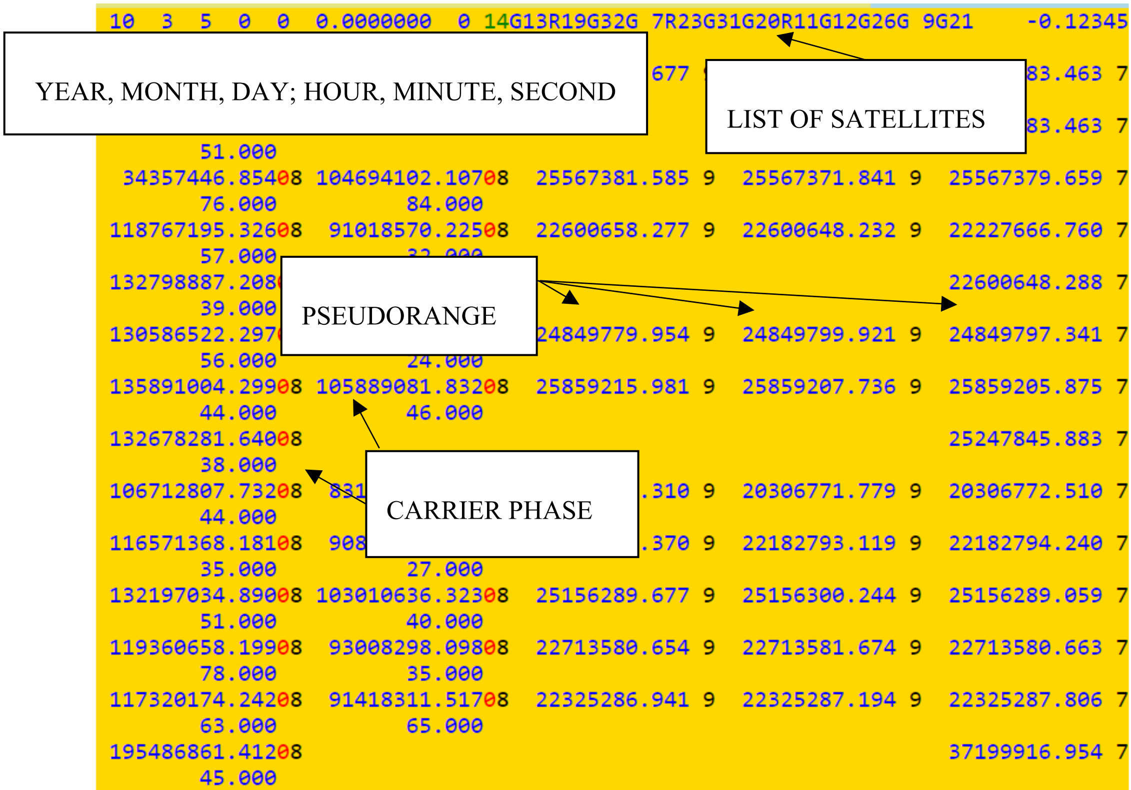


Figure 7 – RINEX 2.11 Observation file measurements [14]

The doppler is an additional observable that measures the quickness that the satellite is moving with respect the receiver. Its signed is defined positive for approaching satellites.

It is worth to say that depending on the RINEX version the order and measurements displayed may change. The example, as written in the header, is a version 2.11.

3.8.2. Navigation RINEX

The navigation RINEX message provides information of the satellite orbit elements, satellite position and velocity, clock parameters, almanac, satellite health status, and other complementary information. [7]

It also can be divided in two parts, header, and measurements (at the same time divided in satellites).

In the header, there could be information about the RINEX version (in this case 2.11), the kind of data (in this case navigation), or the receiver position. An example of a RINEX header can be seen bellow.

```
     2.11           N: GPS NAV. MESSAGE                     RINEX VERSION / TYPE
gLAB                gAGE / UPC          17-MAR-10 12:14     PGM / RUN BY / DATE
EXAMPLE OF A NAVIGATION RINEX FILE                          COMMENT
THIS FILE IS PART OF THE gLAB TOOL SUITE                    COMMENT
FILE PREPARED BY: ADRIA ROVIRA GARCIA                       COMMENT
PLEASE EMAIL ANY COMMENT OR REQUEST TO: glab.gage @ upc.edu COMMENT
     0.1676D-07     0.2235D-07    -0.1192D-07    -0.1192D-07ION ALPHA
     0.1208D+06     0.1310D+06    -0.1310D+06    -0.1966D+06ION BETA
   0.133179128170D-06   0.107469588780D-12   552960     1025DELTA-UTC: A0,A1,T,W
    15                                                      LEAP SECONDS
                                                            END OF HEADER
```

Figure 8 – RINEX 2.11 Navigation file header [14]

The measurements in the navigation files are divided in different groups. Each group of measurements is referred to the same satellite and provide data about orbit, position, velocity, and other measurements to be able to compute accurately satellite coordinates.

This data is sent with a different frequency than the observation data (which use to be every second). In this case, the ephemeris can be provided every two, four, six or even more hours. The idea is to use the information from two hours before the reception of the message until two hours later. Even though this is the most common practice, there is a measurement (known as fit interval) whose purpose is to indicate the usable time of the data.

An example of the measurements of a navigation data is shown below.

```
 6 99  9  2 19  0  0.0 -.839701388031D-03 -.165982783074D-10  .000000000000D+00
     .910000000000D+02  .934062500000D+02  .116040547840D-08  .162092304801D+00
     .484101474285D-05  .626740418375D-02  .652112066746D-05  .515365489006D+04
     .409904000000D+06 -.242143869400D-07  .329237003460D+00 -.596046447754D-07
     .942817490922D+00  .326593750000D+03  .206958726335D+01 -.638312302555D-08
     .307155651409D-09  .000000000000D+00  .102500000000D+04  .000000000000D+00
     .000000000000D+00  .000000000000D+00  .000000000000D+00  .910000000000D+02
     .406800000000D+06  .000000000000D+00
```

Figure 9 – RINEX 2.11 Navigation data measurements [14]

Where the first number indicates the satellite PRN, then it goes the time of the satellite and after it all the other measurements represent different information, as previously said, about position, velocity, orbit, almanac, etc...

3.8.3. RINEX Version 3.X

In the last chapter all the examples that can be found are RINEX v2.11. Although it is still in use, due to its limitations RINEX v3 was created in 2006/2007 to provide generic and systematic support for all GNSS constellations.

The new RINEX format resulted very important as it allow multi constellation, which increase in a big amount, the number of measurements in the file. This upgrade improves the quality of the navigation solutions and give users more coverage worldwide. [8]

But this is not the only change changes with respect previous versions as it can be seen in the next example of an observation RINEX version 3 measurements. Where, for example, the format is not equal.

```
> 2010 03 05 00 00 00.0000000  0 14      -0.123456789012
G13 121367582.20508  94572134.49208  23095489.677 9  23095481.949 9  23095483.463 7        42.000          40.000
R19 134357446.85408  23095483.463 7        51.000
G32  34357446.85408 104694102.10708  25567381.585 9  25567371.841 9  25567379.659 7        76.000          84.000
G07 118767195.32608  91018570.22508  22600658.277 9  22600648.232 9  22227666.760 7        57.000          32.000
R23 132798887.20808  22600648.288 7        39.000
G31 130586522.29708 101755719.98608  24849779.954 9  24849799.921 9  24849797.341 7        56.000          24.000
G20 135891004.29908 105889081.83208  25859215.981 9  25859207.736 9  25859205.875 7        44.000          46.000
R11 132678281.64008  25247845.883 7        38.000
G12 106712807.73208  83152833.16108  20306772.310 9  20306771.779 9  20306772.510 7        44.000          46.000
G26 116571368.18108  90834826.58108  22182792.370 9  22182793.119 9  22182794.240 7        35.000          27.000
G09 132197034.89008 103010636.32308  25156289.677 9  25156300.244 9  25156289.059 7        51.000          40.000
G21 119360658.19908  93008298.09808  22713580.654 9  22713581.674 9  22713580.663 7        78.000          35.000
G15 117320174.24208  91418311.51708  22325286.941 9  22325287.194 9  22325287.806 7        63.000          65.000
S24 195486861.41208  37199916.954 7        45.000
```

Figure 10 - RINEX v3 Observation file [14]

3.9. Pseudorange simplified model

GNSS receivers collect data at specified intervals (for example every second), generally instructed by the receiver user. The receiver clock time is used to know exactly when the measurement is sampled, the term that remains unknown is the time of the satellite clock when the signal was transmitted.

The actual observation of the satellite can be written as:

$$P^S = (T - T^S)\,c \qquad [1]$$

Where T is the known time, T^S is the satellite time of transmission and c is the speed of light in vacuum.

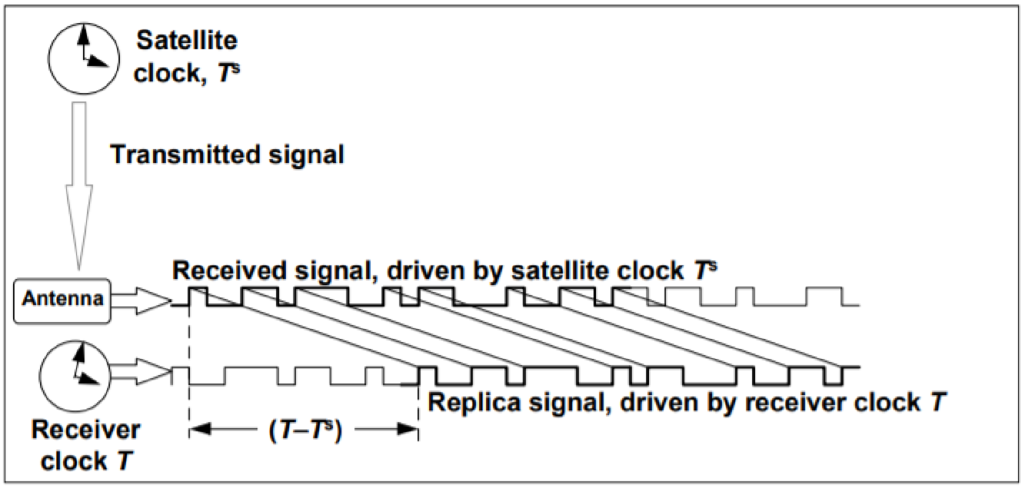


Figure 11 - Pseudorange simplified model [2]

An upgrade of the equation above can be done by adding a clock bias to each of the time in the last equation explained. With this change applied to it, the next equation can be achieved after a several calculations. [16]

$$P^S(t\,,\,t^S) = \sqrt{(x^S(t^S) - x(t))^2 + (y^S(t^S) - y(t))^2 + (z^S(t^S) - z(t))^2} \qquad [2]$$

The Navigation message allows us to compute the satellite position (x^S , y^S , z^S) and the satellite clock bias τ^S. At this point there are only four terms that need to be calculated. That is why a minimum of four satellites need to be monitored to be able to compute a navigation solution.

3.10. Carrier-phase simplified model

The carrier phase measurement is a measure of the range between a satellite and receiver expressed in units of cycles of the carrier frequency.

These carrier phase measurements are much more precise than the code measurements (typically two orders of magnitude more precise), but they are ambiguous by an unknown integer number of wavelengths (λN).

The carrier phase measurements $\phi_L = \lambda_L \; \phi_L$ can be modelled as:

$$\phi_L = \rho + c\,(dt_r - dt_s) + \lambda_L N + B_\phi \qquad [3]$$

Where ρ Is the geometric range between the receiver and the satellite, *c* is the speed of light, λ is the wavelength of the signal, N is the integer ambiguity and B corresponds to carrier phase bias, which include several terms each of them associated with delays or noises. [16]

The principal delays that include the term B_ϕ are:

- Multipath error
- Receiver noise.
- Ionospheric delay.
- Tropospheric delay.
- Instrumental delays.

Those terms are going to be analyzed in the next chapter, as the correct modelling of them result in a much more precise navigation solution.

4. Sources of error

Going deeper into the study and analysis of the GNSS signals, to be able to obtain accurate positioning, it is a key factor to understand the sources of errors of the different measurements to try to avoid them.

This errors that make the measurements imprecise can be divided in two different groups, measurement modelling and measurement features and noise.

4.1. Measurement features and noise

To solve the GNSS positioning problem, combinations of pseudorange and carrier phase can be used. As they are not the same measurements, they also behave differently.

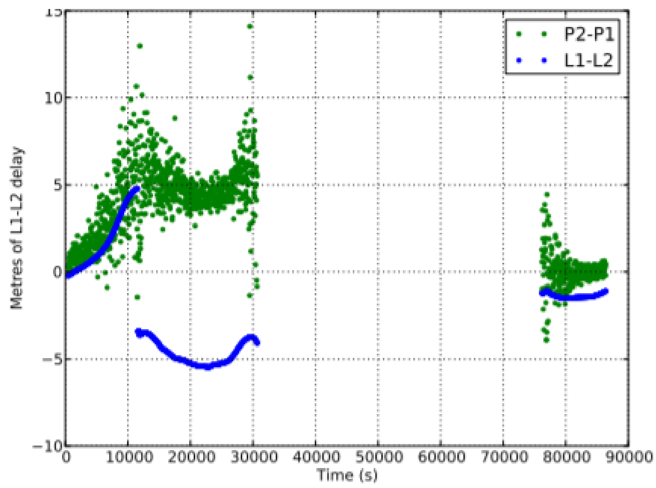


Figure 12 – Range and Carrier-phase behavior [10]

As it can be seen in the last picture, carrier phase (in blue) seems to be very precise but not very accurate. On the contrary, range measurements (in green) are accurate but not precise.

This noise depends on factors like, design of the antenna, signal power or the correlation process. They are independent of each receiver and cannot be avoid with differential techniques or by combinations of measurements.

4.2. Receiver noise

Because of the wave form, the receiver noise can be smoothed with a low-pass filter. By this correction carrier phase measurements can be reduced at a level of few millimeters. In the case of pseudorange, it can be reduced to about tens of centimeters.

4.3. Multipath

Multipath is the phenomenon that occurs when the signal arrives at the antenna by different ways. This use to occur when the antenna is surrounded of reflecting structures and it is more probable when the satellite that emits the signal has a low elevation. In the next figure an example of multipath can be seen.

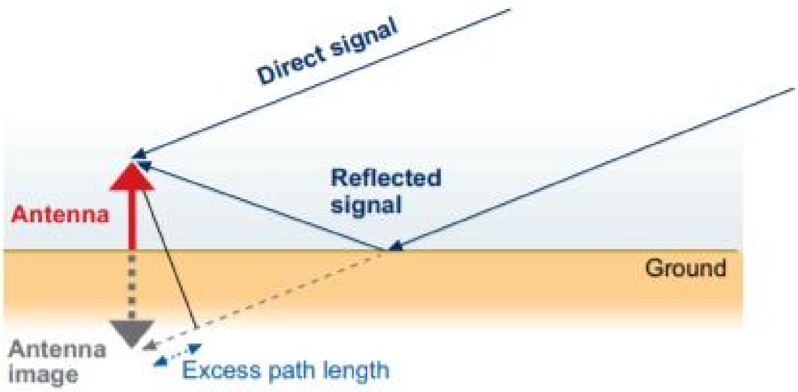


Figure 13 - Multipath representation [10]

There are some ways to avoid this phenomenon to happen all of them referred to improve the environment of the antenna. It can be done for example by moving it away from reflecting structures or by attenuation signals with low-elevation directions.

4.4. Cycle slip

A cycle slip is a discontinuity in a receiver's phase lock on a satellite's signal. The principal causes can be a power loss, a very low signal-to-noise ratio, a failure of the receiver software or a malfunctioning satellite oscillator. It can also be caused by severe ionospheric conditions. Most common, however, are obstructions such as buildings, trees, etc., that are so solid they prevent the satellite signal from being tracked by the receiver. After a cycle slip when the satellite reappears, the tracking resumes.

Pseudorange measurements are not as affected by cycle slips as are carrier phase measurements. On the other hand, carrier phase positioning accuracy suffers if cycle slips are not detected and repaired. A cycle slip causes the critical component for successful carrier phase positioning.

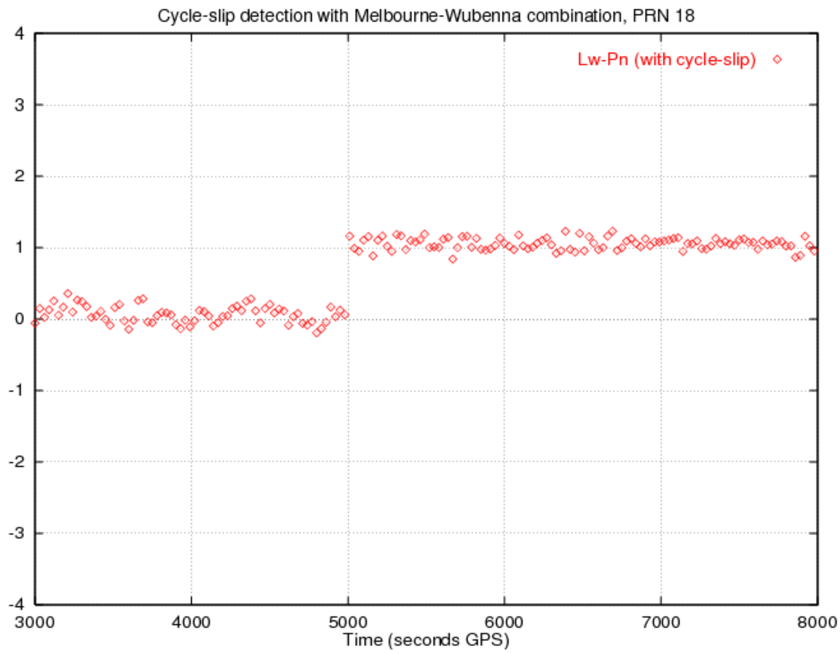


Figure 14 – Cycle slip representation [4]

Because of the importance of the detection of cycle slips to be able to avoid them, some heuristic methods are used to this aim. Some examples of this methods are the double-differences or triple-differences.

4.5. Measurement modelling

As previously explained, carrier phase and pseudorange measurements contain some additional time delays that without a correct treatment can cause an imprecise solution.

In the next chapters the different corrections that are made for each measurement are presented. These techniques are a key of the correct functionality of satellite systems. And they are divided in the different known delays.

4.5.1. Clock corrections

Clock delays are a cause of a bad clock synchronization referring to the GNSS time scale. They can be explained separately in two terms. There is the receiver clock offset and the satellite clock offset.

There can be difference between to identical clocks, one placed in the receiver and the other in the satellite, are due to the general relativity (because the different gravitational potential) and the speed relativity (due to the different speed).

To model them, there are two components that can be calculated separately. A constant component that depends only on the major component of the semi-major axis of the satellite orbit and a periodic component depending on the eccentricity of the orbit.

4.5.2. Instrumental delays

Possible sources of these delays are antennas, cables, as well as different filters used in receivers and satellites. These instrumental delays affect both, pseudorange and carrier phase measurements. [10]

The receiver instrumental delay is assimilated in the receiver clock. So, being common for all satellites, it is assumed to be zero and is included in the receiver clock estimate.

4.5.3. Ionospheric delay

The ionosphere is the zone of the terrestrial atmosphere that extends itself from about 60 km until more than 2.000 km in high. As it name says, it contains a partially ionized medium, as result of the X and UV rays of Solar Radiation and the incidence of charged particles. [3]

The propagation speed of the GNSS electromagnetic signals in the ionosphere depends on its electron density (see below), which is typically driven by two main processes: during the day, sun radiation causes ionization of neutral atoms producing free electrons and ions. During the night, the recombination process prevails, where free electrons are recombined with ions to produce neutral particles, which leads to a reduction in the electron density.

4.5.4. Tropospheric delay

The troposphere is the lowest layer of atmosphere, and it contains 75% of the atmosphere's mass and 99% of the total mass of water vapor and aerosols. It is non-ionized and non-dispersive medium with respect to radio waves up to 15 GHz. [3]

The troposphere affects the GNSS signals in the way that signals are both delayed and refracted. This delay is divided, based on physical parameters, into hydrostatic delay, also known as dry delay, and wet delay. The hydrostatic delay is caused by dry gases and particles in the troposphere, and it is about eighty to ninety percent of the total tropospheric delay.

Hydrostatic delay can be precisely determined from surface pressure measurements using empirical models. The tropospheric wet delay is due to water vapor content in the troposphere, and it is difficult to be precisely modeled, because the water vapor in the troposphere is not well mixed.

The determination of the tropospheric zenith wet delay (ZWD) cannot be consistently modeled with millimeter precision by any existing empirical model. So, the ZWD is one of the accuracies limiting factors in GNSS positioning. The ZWD precise estimation is important for high precision applications, such as Network Real Time Kinematic (RTK) and precise point positioning (PPP). In addition, the ZWD values calculated from GNSS measurements can be used in Numerical weather prediction. [10]

5. Data Measuring

5.1. GNSS receivers

GNSS Receivers process the Signals In Space (SIS) transmitted by the satellites, being the user interface to any Global Navigation Satellite System (GNSS). Although the information provided by a generic GNSS receiver can be used by a wide range of Applications, most of them rely on the receiver's navigation solution.

Receivers can determine the user position by executing the GNSS algorithms with the measured signals as inputs. Because the satellites are always in motion, the receiver has to continuously acquire and track the signals from the satellites in view, in order to compute an uninterrupted solution, as desired in most applications.

5.2. Base Stations

A base station (also known as reference station) is a station where a GNSS receiver is installed at a known location. The data from these stations can be used for GNSS receivers to augment the system with another set of data apart from the one provided from the user device.

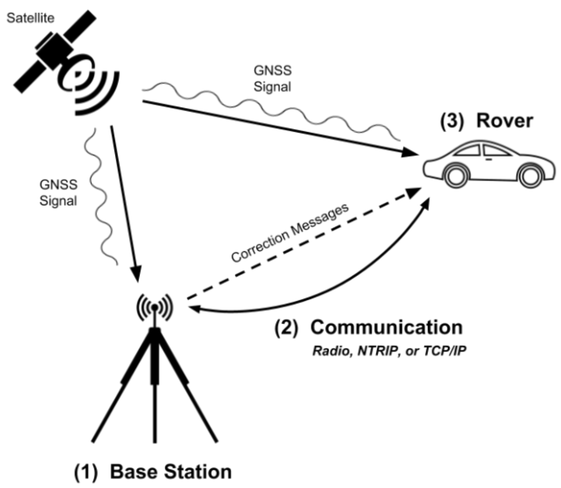


Figure 15 – Representation of the use of a Base station [12]

By this augmentation, accuracy can be improved to a centimeter level (10cm-50cm, depending on the closeness of the Station with respect the rover).

5.3. u-blox

u-blox are low-cost GNSS receivers of small size. What is interesting from this device is the good performance that they have for their price (they can be obtained from 15€). Part from good performance, u-blox are very portable and can be added easily to a lot of other systems.

In the case of this project, u-blox are used to work as a base station and augment the set of data provided from mobile phones. The aspect that u-blox have is shown in the next figure.

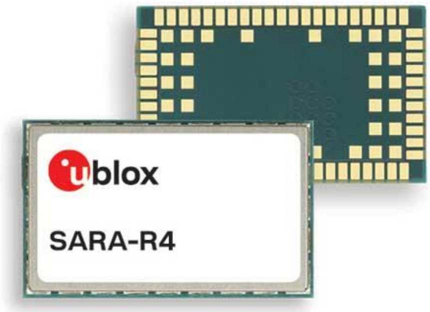


Figure 16 - u-blox [17]

The model used for collecting data measurements in this project was the u-blox M8T. It has support for GPS, BeiDou, GLONASS and Galileo but allow users to track currently satellites from three different constellations. It measures are 12.2 × 16.0 × 2.4 mm which makes it a portable device. [17]

5.4. Mobile phones as GNSS receivers

The last measurements used in the analysis are provided from common mobile phones. Nowadays, mostly all phones include a set for GNSS positioning so that means that can be used as receivers.

But there is a problem when the desired data is desired to be unprocessed, because not all phones permit it. The possibility of using raw data from a phone remains in the characteristics of each of them. This is a bottleneck of the project as not all phones are designed to do that. Users need to analyze the data sheet of the phone and see if it can afford it.

To facilitate this duty, android developers' web site (https://developer.android.com/) has a table in which they show several phones supporting the desired characteristics. At the time of writing these lines, this list of phones was the following.

Model	Android version	Automatic Gain Control	Navigation messages	Accumulated delta range	HW clock	Global systems
HTC U11 Plus	8.0	no	no	no	yes	GPS GLONASS
HTC U11 Life	8.0	no	no	no	yes	GPS GLONASS
Huawei Mate 10	8.0	no	yes	yes	yes	GPS GLONASS
Huawei Mate 10 Pro	8.0	no	yes	yes	yes	GPS GLONASS QZSS
Google Pixel 2 XL	8.0	yes	no	no	yes	GPS GLONASS GALILEO BeiDou QZSS
Google Pixel 2	8.0	yes	no	no	yes	GPS GLONASS GALILEO BeiDou QZSS
Sony Xperia XZ1	8.0	no	no	no	yes	GPS GLONASS
Samsung Note 8 (Exynos)	7.1	no	yes	yes	yes	GPS GLONASS GALILEO BeiDou
Samsung Note 8 (QCOM)	7.1	no	no	no	yes	GPS GLONASS GALILEO BeiDou
LG V30	7.1.2	no	no	no	yes	GPS GLONASS
Moto X4 2017	7.1	no	no	no	yes	GPS GLONASS

Essential PH-1	7.1	no	no	no	yes	GPS GLONASS
Moto Z2	7.1	no	no	no	yes	GPS GLONASS
HTC U11	7.1	no	no	no	yes	GPS GLONASS
OPPO R11	7.1	no	no	no	yes	GPS GLONASS GALILEO BeiDou
Huawei Honor 9	7.0	no	yes	yes	yes	GPS GLONASS
Samsung S8 (Exynos)[1]	7.0	no	yes	yes	yes	GPS GLONASS GALILEO BeiDou QZSS
Samsung S8 (QCOM)[2]	7.0	no	no	no	yes	GPS
Huawei P10	7.0	no	yes	yes	yes	GPS GLONASS GALILEO BeiDou QZSS
Huawei P10 Lite	7.0	no	no	no	yes	GPS
Huawei Honor 8	7.0	no	yes	yes	yes	GPS GLONASS BeiDou
Huawei Mate 9	7.0	no	yes	yes	yes	GPS GLONASS BeiDou
Huawei P9	7.0	no	yes	yes	yes	GPS GLONASS BeiDou
Google Pixel XL	7.0	no	no	no	yes	GPS
Google Pixel	7.0	no	no	no	yes	GPS
Nexus 6P[3]	7.0	no	no	no	no	GPS
Nexus 5X[3]	7.0	no	no	no	no	GPS

Figure 17 – List of phones able to provide raw GNSS data [19]

Apart from the ones listed, there are more phones capable of saving raw GNSS data, but the only way of proving that utility is to check it by yourself.

5.5. Phones used in the measurement campaigns

There are a lot of phones that can be used as GNSS receivers, for collecting satellite signals for this project, the models used are the following:

- Xiaomi Pocophone F1: Single frequency receiver tracking GPS (L1 frequency), GLONASS (G1), Galileo (E1) and BeiDou (B1).

  This cell phone costs about 320 euros, and acts in this project as a single frequency GNSS receiver. It went on sale in 2019 and has a size of 6.18 inches.

Figure 18 - Xiaomi Pocophone F1

- Xiaomi MI8: Dual Frequency receiver tracking GPS (L1 + L5 frequencies), GLONASS (G1), Galileo(E1 + E5a) and BeiDou (B1).

  This phone costs about 520 euros, went on sale in 2018 and has a size of 6.21 inches. It is equipped with the receiver called *Broadcom BCM47755.* It is the first dual frequency GNSS receiver for smartphones. This new chipset is expected to significantly reduce errors in urban environments, thus allowing location-based applications to offer a better consumer experience.

-

Figure 19 - Xiaomi MI8

- Samsung A50: Single frequency receiver tracking GPS (L1 frequency), GLONASS (G1), Galileo (E1) and BeiDou (B1).

  This last receiver costs about 300 euros, went on sale in 2019 and has a size of 6.4 inches. It is single frequency so the characteristics seem similar that the ones found in Xiaomi Pocophone.

-

Figure 20 - Samsung A50

5.6. Phone applications

Apart from looking for a phone that can solve the needs of the project, it is also necessary to be able to convert that raw data into RINEX format files. At this point is when phone apps are important. It is not very difficult to find an application that converts raw data into RINEX files in the APP store, the only thing to do is search them by the word “RINEX” or something similar and a brunch of options will appear.

In this case only two Apps will be used. Both are summarized below:

5.6.1. Geo++ RINEX Logger

Geo++ RINEX Logger is an easy application that can generate RINEX observation files using the phone as a receiver. The way this App works is just waiting until the screen is fully charged (because if not, maybe first epochs do not receive proper data) and the press the start button.

Once the user desire to end the measurement, the only thing to do is pressing the stop button. After it, the application stores the RINEX file in a path that can be found in the settings of the App.

Figure 21 - GEO++ App Interface

5.6.2. RINEX on

RINEX on is an application developed by a company called Nottingham Scientific Ltd (nowadays part of GMV group) that works similarly than the last one explained.

The principal advantage is that after the user stops measuring, it is possible to send an email with the RINEX file.

6. Data processing

Once the data collected from the Satellites is saved, there are some algorithms than can be applied to compute a navigation solution. As those algorithms require a heavy computation, they are usually done by programs constructed for that duty.

Probably the best known is RTKLIB software, which will be explained in the next chapter.

6.1. RTKLIB Software

RTKLIB is an open-source program for GNSS positioning developed by Tomoji Takasu and freely available from *http://www.RTKLIB.com/*. It consists of a portable program library and several application programs utilizing the library.

It supports standard and precise positioning algorithms with GPS, GLONASS, Galileo or Beidou. Having the possibility to process de data in real-time or post-processing. RTKLIB also supports RINEX files as an input.

For this thesis, the version 2.4.3 of RTKLIB will be the one used. The goal of this projects remains in doing and analysis of the feasibility of GNSS applications in harsh environments. To do so, only the post processing module will be needed. The procedure of the analysis is based in obtaining the measurements of GNSS data both in harsh environments (for example in a spot with high building surrounding or reflecting materials) and friendly environments (like open sky spots) comparing the accuracy and precision of the position solution.

For this aim, after the measurement when can compute the solution at any time by post-processing it. The Interface of RTKLIB post processing module in shown in the next image to be able to understand it better.

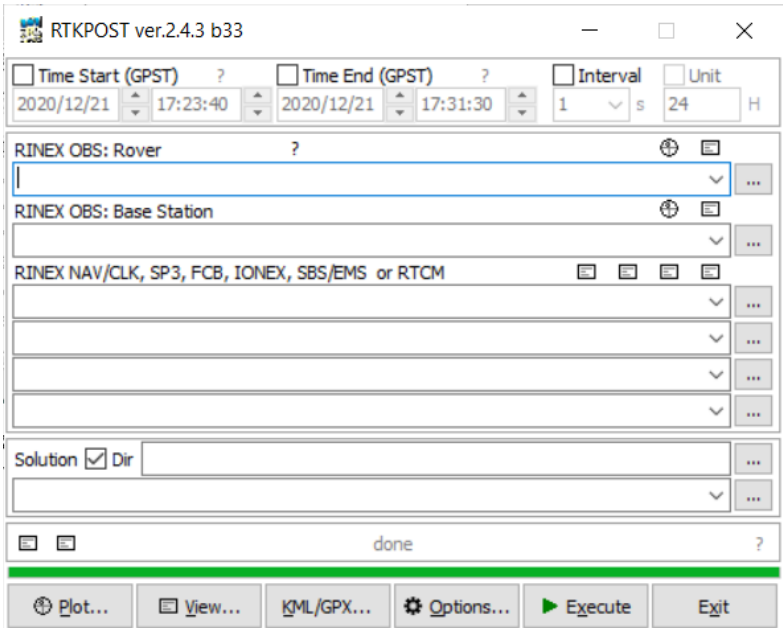


Figure 22 – RTKLIB main screen

In this picture it can be seen the main screen of RTKPOST. In this program window the user can introduce the inputs, for example, the RINEX observation files and the RINEX navigation files and just Execute the program to obtain a positioning solution.

6.1.1. Inputs

In the main screen is where input files are introduced. There are some different input files that need to be explained. The first of them is the rover.

A rover is any mobile GNSS receiver that is used to collect or update data in the field, typically at an unknown location. In our case a rover is a mobile phone that can measure GNSS signals. It worth to add that any kind of receiver that provides observation data can be used as a rover. A combination with an u-blox receiver user as a rover will be analyzed to review the quality of the results

In the case of the base station, both u-blox and permanent GNSS stations will be used. The principal advantage that permanent stations have is that their location is known, so the errors can be computed easily.

Navigation data is also a necessary input for any kind of navigation solution. This data is provided also from u-blox and base stations.

It is important to highlight that although RTKLIB is able to process a lot of input formats, only RINEX files will be used.

6.1.2. Post-Processing

Post-processing, as its name indicates, is a method of computation of GNSS data with measurements that have already been stored. The importance of this way of working remains in the goal of improving the solution and analyzing the possible problems that can occur.

During this project, the same stored GNSS measurements are used to compute the solution with different methods and combinations. The idea is to be able to understand which input files have better data and which of them give the user a properly level of accuracy in the results.

6.1.3. Position solution

The solution file will be stored in the path that the user gives to the program. This information can also be shown pressing the button "View".

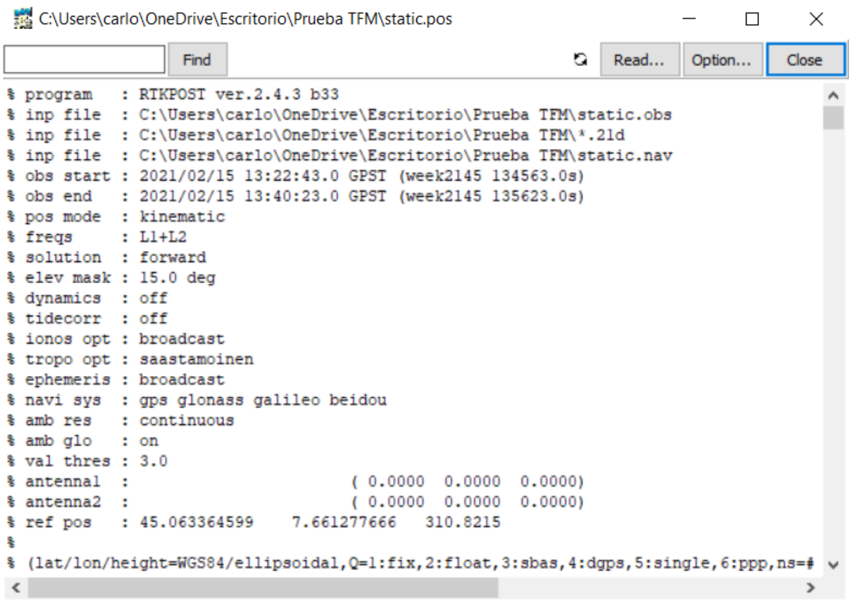
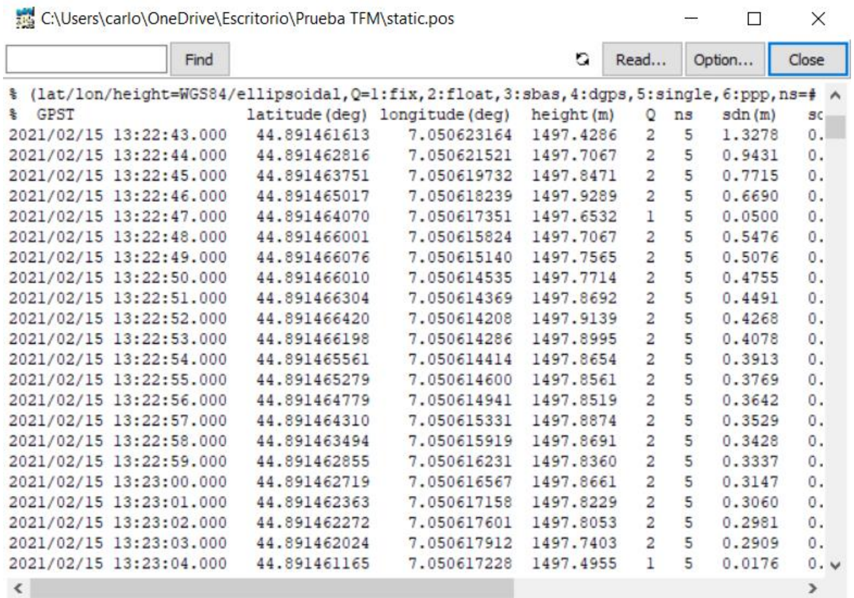


Figure 23 – RTKLIB position solution file header

The lines starting with "%" are header lines. The header lines contain some additional information or processing options. The header of the file provides some information of the way this colution was computed. There can be found the input files path, solution mode selected, satellite mask, frequency used, etc..

After the header it comes the position solutions, each row is the position solution of an epoch. In every one of them, diferent data such as latitud, longitud or height can be found. An example of this file is shown below.




```
% (lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,ns=#
%  GPST                  latitude(deg) longitude(deg)  height(m)   Q  ns   sdn(m)   sc
2021/02/15 13:22:43.000   44.891461613    7.050623164  1497.4286   2   5   1.3278   0.
2021/02/15 13:22:44.000   44.891462816    7.050621521  1497.7067   2   5   0.9431   0.
2021/02/15 13:22:45.000   44.891463751    7.050619732  1497.8471   2   5   0.7715   0.
2021/02/15 13:22:46.000   44.891465017    7.050618239  1497.9289   2   5   0.6690   0.
2021/02/15 13:22:47.000   44.891464070    7.050617351  1497.6532   1   5   0.0500   0.
2021/02/15 13:22:48.000   44.891466001    7.050615824  1497.7067   2   5   0.5476   0.
2021/02/15 13:22:49.000   44.891466076    7.050615140  1497.7565   2   5   0.5076   0.
2021/02/15 13:22:50.000   44.891466010    7.050614535  1497.7714   2   5   0.4755   0.
2021/02/15 13:22:51.000   44.891466304    7.050614369  1497.8692   2   5   0.4491   0.
2021/02/15 13:22:52.000   44.891466420    7.050614208  1497.9139   2   5   0.4268   0.
2021/02/15 13:22:53.000   44.891466198    7.050614286  1497.8995   2   5   0.4078   0.
2021/02/15 13:22:54.000   44.891465561    7.050614414  1497.8654   2   5   0.3913   0.
2021/02/15 13:22:55.000   44.891465279    7.050614600  1497.8561   2   5   0.3769   0.
2021/02/15 13:22:56.000   44.891464779    7.050614941  1497.8519   2   5   0.3642   0.
2021/02/15 13:22:57.000   44.891464310    7.050615331  1497.8874   2   5   0.3529   0.
2021/02/15 13:22:58.000   44.891463494    7.050615919  1497.8691   2   5   0.3428   0.
2021/02/15 13:22:59.000   44.891462855    7.050616231  1497.8360   2   5   0.3337   0.
2021/02/15 13:23:00.000   44.891462719    7.050616567  1497.8661   2   5   0.3147   0.
2021/02/15 13:23:01.000   44.891462363    7.050617158  1497.8229   2   5   0.3060   0.
2021/02/15 13:23:02.000   44.891462272    7.050617601  1497.8053   2   5   0.2981   0.
2021/02/15 13:23:03.000   44.891462024    7.050617912  1497.7403   2   5   0.2909   0.
2021/02/15 13:23:04.000   44.891461165    7.050617228  1497.4955   1   5   0.0176   0.
```

Figure 24 - RTKLIB position solution file data

After longitude, latitude and height, it comes a flag, with a value of the following meaning.

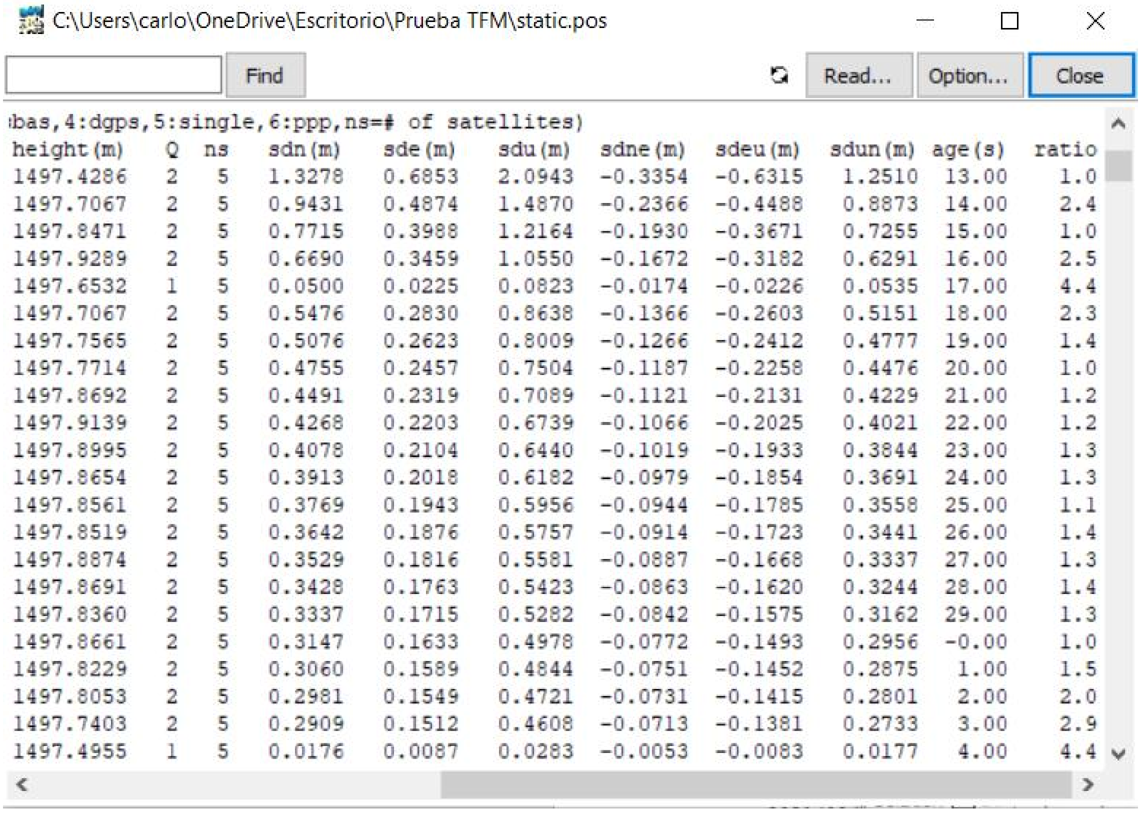
1: Fixed, solution by carrier-based relative positioning and the integer ambiguity is properly resolved.

2: Float, solution by carrier-based relative positioning but the integer ambiguity is not resolved.

3: Reserved.

4: DGPS, solution by code-based DGPS solutions or single point positioning with SBAS corrections.

5: Single, solution by single point positioning.



height(m)	Q	ns	sdn(m)	sde(m)	sdu(m)	sdne(m)	sdeu(m)	sdun(m)	age(s)	ratio
1497.4286	2	5	1.3278	0.6853	2.0943	-0.3354	-0.6315	1.2510	13.00	1.0
1497.7067	2	5	0.9431	0.4874	1.4870	-0.2366	-0.4488	0.8873	14.00	2.4
1497.8471	2	5	0.7715	0.3988	1.2164	-0.1930	-0.3671	0.7255	15.00	1.0
1497.9289	2	5	0.6690	0.3459	1.0550	-0.1672	-0.3182	0.6291	16.00	2.5
1497.6532	1	5	0.0500	0.0225	0.0823	-0.0174	-0.0226	0.0535	17.00	4.4
1497.7067	2	5	0.5476	0.2830	0.8638	-0.1366	-0.2603	0.5151	18.00	2.3
1497.7565	2	5	0.5076	0.2623	0.8009	-0.1266	-0.2412	0.4777	19.00	1.4
1497.7714	2	5	0.4755	0.2457	0.7504	-0.1187	-0.2258	0.4476	20.00	1.0
1497.8692	2	5	0.4491	0.2319	0.7089	-0.1121	-0.2131	0.4229	21.00	1.2
1497.9139	2	5	0.4268	0.2203	0.6739	-0.1066	-0.2025	0.4021	22.00	1.2
1497.8995	2	5	0.4078	0.2104	0.6440	-0.1019	-0.1933	0.3844	23.00	1.3
1497.8654	2	5	0.3913	0.2018	0.6182	-0.0979	-0.1854	0.3691	24.00	1.3
1497.8561	2	5	0.3769	0.1943	0.5956	-0.0944	-0.1785	0.3558	25.00	1.1
1497.8519	2	5	0.3642	0.1876	0.5757	-0.0914	-0.1723	0.3441	26.00	1.4
1497.8874	2	5	0.3529	0.1816	0.5581	-0.0887	-0.1668	0.3337	27.00	1.3
1497.8691	2	5	0.3428	0.1763	0.5423	-0.0863	-0.1620	0.3244	28.00	1.4
1497.8360	2	5	0.3337	0.1715	0.5282	-0.0842	-0.1575	0.3162	29.00	1.3
1497.8661	2	5	0.3147	0.1633	0.4978	-0.0772	-0.1493	0.2956	-0.00	1.0
1497.8229	2	5	0.3060	0.1589	0.4844	-0.0751	-0.1452	0.2875	1.00	1.5
1497.8053	2	5	0.2981	0.1549	0.4721	-0.0731	-0.1415	0.2801	2.00	2.0
1497.7403	2	5	0.2909	0.1512	0.4608	-0.0713	-0.1381	0.2733	3.00	2.9
1497.4955	1	5	0.0176	0.0087	0.0283	-0.0053	-0.0083	0.0177	4.00	4.4

Figure 25 - RTKLIB position solution file data

The column ns reflects the number of satellites used to compute the solution. The parameters sdn, sde and sdu are the estimated standard deviations of the solution assuming a priori error model and error parameters by the positioning options. The last letter of sdn, sde or sdu means N (north), E (east) or U (up).

The absolute value of sdne, sdeu or sdun means square root of the absolute value of NE, EU or UN component of the estimated covariance matrix. The sign in this number indicates the sign of the covariance.

The column age (or age difference) means the time difference between the observation data epochs of the rover receiver and the base station in seconds.

Finally, ratio factor of ″ratio-test″ means the ratio of the squared sum of the residuals with the second-best integer vector to with the best integer vector.

Apart from this file, RTKLIB also provides the user with some graphical information, which can be found pressing the “Plot” button.

The plots have some diferrences depeding on the solution mode that had been configured. In the next figures there can be seen two of the plots that can be obtained from single solution.

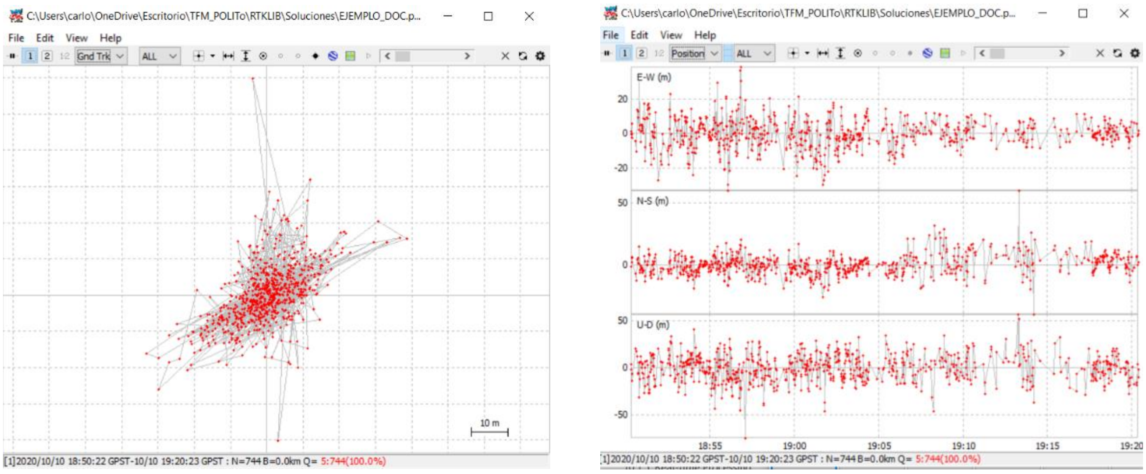


Figure 26 - RTKLIB position solution plots

6.1.4. GNSS Algorithms

<u>Single Point Positioning:</u>

Probably the simplest way of computing a position solution with the input data summarized above is by Least Squares Algorithm. The method is a standard approach to approximate the solution of overdetermined systems (when there are more equations than unknowns) by minimizing the sum of the squares of the residuals made in the results of every single equation.

RTKLIB employs an iterated weighted least square estimation. Depending on the measurements the computation can be done as Linear LSE or non-linear LSE.

Linear LSE -> In the case of Linear equations they can be represented by this form:

$$y = Hx + v \qquad [4]$$

The idea of the algorithm is to minimize the cost function given by the sum of the squared measurement errors represented by the *v*. The estimated unknown parameter vector by the weighted LSE can be obtained by:

$$\hat{X} = (H^T\, W\, H)^{-1}\, H^T\, W\, y \qquad [5]$$

The weight matrix W for the LSE is often given as:

$$W = diag(\sigma_1^{-2}, \sigma_2^{-2}, \ldots, \sigma_m^{-2}) \qquad [6]$$

Where σ_I is the standard deviation of i-th measurement error.

Non-linear LSE -> In case that the measurements are not given as linear models, the measurement equations can be written by a general non-linear vector function as:

$$y = h(x) + v \qquad [7]$$

By applying weighted LSE the normal equation for this non-linear systems is:

$$\hat{X} = X_0 + (H^T\, W\, H)^{-1}\, H^T\, W\, (y - h(X_0)) \qquad [8]$$

If the initial parameters X_0 are not enough near the true values, we can iteratively improve the estimated parameters like:

$$\hat{X} = X_0 \qquad [9]$$

$$\hat{X}_{i+1} = X_i + (H^T\, W\, H)^{-1}\, H^T\, W\, (y - h(X_i)) \qquad [10]$$

If the iteration is converged, we can obtain the final estimated parameters as:

$$\lim_{n\to\infty} X_i \qquad [11]$$

For this method, it is only necessary observation data from the rover and navigation data from any station or device.

Kinematic and Static Positioning modes:

RTKLIB employs extended Kalman Filter (EKF) to obtain the final solution in Static and Kinematic modes in conjunction with GNSS signal measurement models and troposphere and ionosphere models.

By using EKF a state vector X and its covariance matrix can be estimated with the measurement vector y.

$$\hat{X}_{k+1}(-) = {H_k}^{k+1}\, \hat{X}_k\, (+) \quad [12]$$

$$P_{k+1}(-) = {F_k}^{k+1}\, P_k\, (+)\, {F_k}^{k+T}\, {Q_k}^{k+1} \quad [13]$$

where $\widehat{X}_k$ and P_k are the estimated state vector and its covariance matrix at the epoch time kt. (-) and (+) indicates before and after measurement update of EKF. h(x), H(x) and R_k are the measurements model vector, the matrix of partial derivatives and the covariance matrix of measurement errors, respectively.

For the combinations in which the master station and rover are relatively near each other, with a short length (<10 km) baseline between the rover r and the base-station b, the following double differenced measurement equations are generally used for the phase-range and pseudorange. In these equations, the satellite and receiver clock biases, and the ionospheric and tropospheric effects and other minor correction terms are almost eliminated by using double differenced technique.

$$\phi_{rb,i}{}^{jk} = \rho_{rb}{}^{jk} + \lambda_i\, (B_{rb,i}{}^{j} - B_{rb,i}{}^{i}) + \phi_{r,i}{}^{s} + \varepsilon_{\phi} \quad [14]$$

$$P_{rb,i}{}^{jk} = \rho_{rb}{}^{jk} + \varepsilon_{\phi} \quad [15]$$

Where ϕ is the carrier-phase, B is the single-differenced carrier-phase biases and ρ is the geometric range (computed from the position of the receivers and satellites).

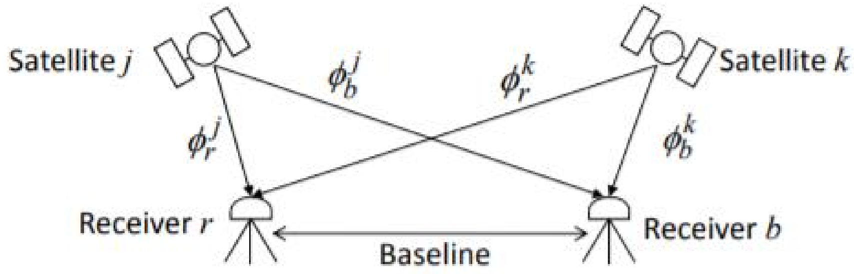


Figure 27 – Dual differences terms schema [4]

By solving the EKF formulas with these equations, the estimated rover antenna position, velocity and float SD carrier-phase biases the epoch time k_t are obtained. The differences between Kinematic and Static modes, is the way they handle the EKT time update.

In each case, the arrays ${F_k}^{k+1}$ and ${Q_k}^{k+1}$ of the equation [13] are replaced with different values depending on the solution mode selected.

Once the estimated states obtained in the EKF measurement update, the float carrier-phase ambiguities can be resolved into integer values in order to improve accuracy and convergence time. By a sequence of equations that is out of the scope of this study a factor is computed from this resolution. Then it is compared to a threshold to determine if the FIXED solution can be obtained, on the other case, the FLOAT solution will appear in the position file and this improvement is not done.

7. GNSS Data analysis

7.1. First analysis

The first analysis that was carried out, was formed of two collections of data. The idea was to collect some measurements in an open sky environment with almost no obstacles and other ones in a harsh environment such as a street of a city with buildings around.

Both measurements were collected with the same phone, to be able to compare the behavior of the same receiver with different environments. The two measurements were taken in Madrid.

Figure 28 - Open sky environment

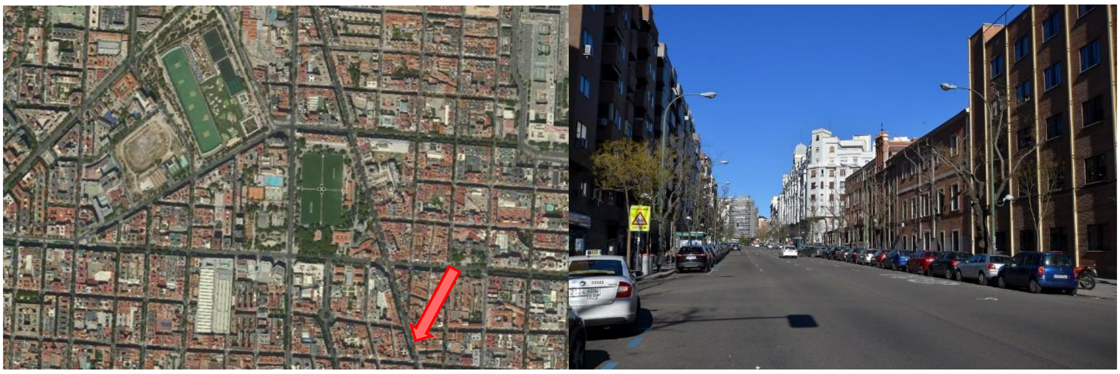


Figure 29 - Harsh environment

7.2. Measurements

Once the measurements where collected, a first review can be done. In the case of the favorable environment, observation data from more than 8 satellites are present in most of the epochs. An example of one of them is shown below.

```
> 2020 10 10 18 50 18.4344740  0 27
C10    40433289.413           -857.563     26.300
C23    25776909.214           2790.402     33.300
C24    22806775.995          -1908.566     34.200
C25    22200704.169           1047.592     38.800
C26    26312050.743          -2988.500     22.100
E01    24319103.032           1728.043     41.100
E04    24629712.602           1741.991     36.600
E19    23572770.907           -994.253     31.400
E21    23951214.516           -995.175     27.900
E27    27478041.645          -2801.501     27.900
E31    28273925.866           2979.600     38.900
G05    23370139.386          -3625.913     42.600
G12    25242142.725           3984.162     37.500
G13    20509652.260           -910.866     27.100
G15    20338038.166            979.632     39.800
G17    24464982.642           2252.200     23.900
G18    24936000.663          -1988.349     27.700
G20    23151718.996           2152.432     36.400
G24    21803794.245           2228.699     38.800
G28    22113485.250          -1968.472     27.600
G30    24782968.005          -3021.200     23.000
R05    21625124.835          -3611.610     38.700
R06    19080341.652            308.371     32.000
R07    21696086.609           3605.156     29.300
R09    20892474.038           2729.023     43.500
R15    22310450.694          -3754.485     33.400
R16    19209555.798           -828.281     31.900
```

There can be found 25 different satellites in that epoch, at first sight is seems a good amount of collected data. The letters before the satellite PRN mean the constellation, G means GPS, R means GLONASS E is for Galileo and C for BeiDou satellites.

In the case of the harsh environment, the satellites in view were quite a less number. One epoch of this RINEX data is also shown.

> 2020 10 12 15 14 43.4310322 0 22

C21	25357274.334	2395.800	19.400
E01	27472378.778	-1400.200	13.700
E04	27366595.510	-2220.300	14.400
E05	24564433.306	1104.500	32.500
E09	21984435.302	-896.200	20.400
E24	26844661.337	2795.850	15.200
E31	25571746.158	1122.135	32.300
E36	25633006.349	-2858.350	22.000
G02	21999039.392	-1176.993	28.700
G05	21360974.315	2073.290	35.900
G06	23129075.584	-3658.390	29.700
G07	21332143.274	-1190.200	18.700
G09	22794291.950	-2757.550	25.200
G13	23075305.108	2515.500	18.300
G30	20715889.298	831.420	35.500
R05	23179553.739	-2926.750	18.700
R06	19785518.780	-514.750	21.700
R07	21135968.983	2036.306	35.200
R09	21757599.137	3783.163	30.300
R15	22531980.744	-3275.700	19.300
R16	20032222.791	559.150	17.400
R23	23874835.105	2499.000	12.000

Although the satellites in view are a bit less, the second measurement seems to be good enough to obtain a suitable solution.

To confirm the quality of the data collected, RINEX data must be post processed with RTKLIB software to get the precision of the solution.

7.3. Position solution

As previously said, to make the comparison, an execution of RTKLIB post processing module will be done. As the phones only receive observation data, Navigation RINEX files were obtained from the same permanent station.

The processing method selected is the single one, no master station will be needed. At this point, the procedure is just checking the results and see how much both differ from each other.

7.4. Open-Sky Environment

To show clearer the solutions with both executions, the results will be shown in a graphical way. In the case of an open-sky environment data, this graphics are shown below.

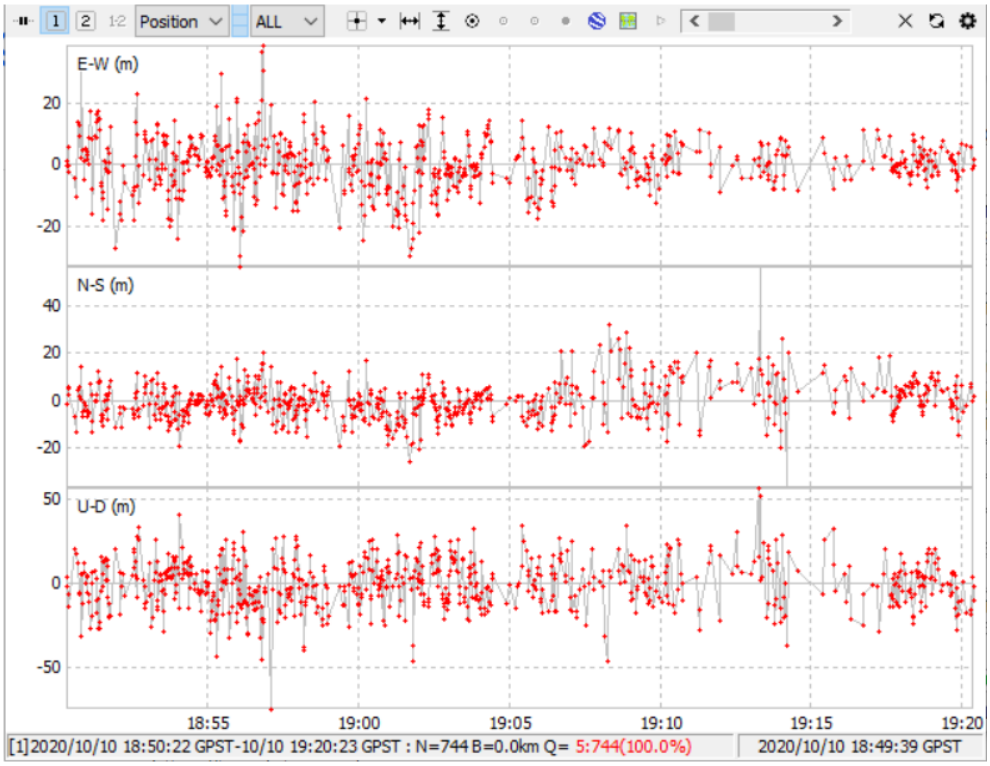


Figure 30 - RTKLIB solution plot in an open sky environment

The errors in the position solution file seem to be not very usable for our purposes during the hole execution (the errors are in the range of +-20 meters). Anyway, considering that the type of solution is single, and the results can be improved with the use of a master station, a further check must be made.

At this point, kinematic option in RTKLIB will be chosen. As previously said, this kind of solution needs a master station to be computed. So RINEX observation data from the nearest GNSS station of the National Spanish Geographic Institute (ign.es) will be used to post-process the measurements.

When trying to execute the tool to get the position solution with this second form, no results were obtained. To check if this problem occurred only with this combination of data, nothing will be done until doing the analysis of the other environment.

7.5. Harsh environment

The same strategy will be followed in the case on a harsh environment. So, the graphic of the position solution during the hole execution is shown below.

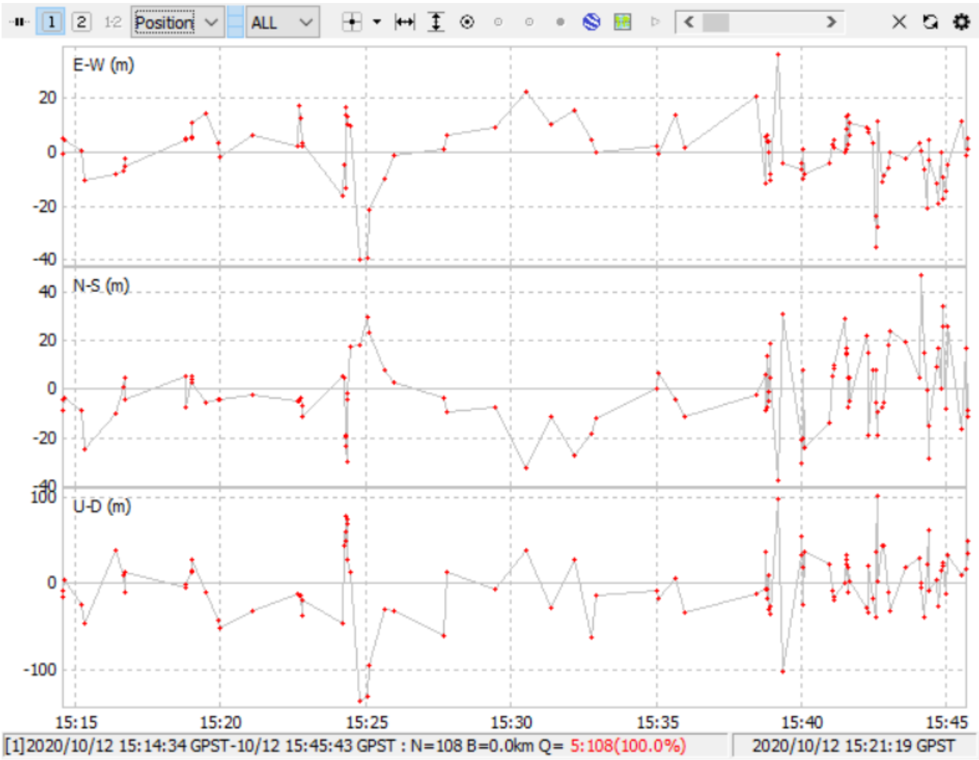


Figure 31 - RTKLIB solution plot in a harsh environment

The first difference that can be easily seen between both representations is the lack of computed data of the second execution compared with the first one (each red point corresponds to an epoch with a position solution computed).

There are several periods of 2 or 3 minutes without solution, for example in a period between epochs at 15:35 and 15:40. Even though the errors in both graphics seem to be within the same range of values, this epochs without solution could lead to many problems when thinking of applying this technology in mechatronic systems.

Following the same steps as before, the kinematic type of solution will be tried. But the result is the same, no solution was computed.

This is a bottle neck of the hole analysis because none of the data measured is giving a usable solution with the techniques used. To be able to keep with the study, this problem must be solved.

7.6. Issues encountered

During the previous executions, a principal issue was encountered, no kinematic solution could be computed to improve and minimize the errors. RTKLIB software provides the users with a file that traces the errors when a solution in an epoch cannot be computed. This is the next step trying to understand the reason why this is happening.

When looking at this file, the same message is repeated in a lot of epochs:

2 19:09:27.43: no double-differenced residual

This warning means that the tool is not able to compute the kinematic aproach, which is based in double differenced observables, as explained in the chapter 2.1.8.1.

$$\phi_{rb,i}{}^{jk} = \rho_{rb}{}^{jk} + \lambda_i \left(B_{rb,i}{}^{j} - B_{rb,i}{}^{i}\right) + \phi_{r,i}{}^{s} + \varepsilon_{\phi} \qquad [16]$$

$$P_{rb,i}{}^{jk} = \rho_{rb}{}^{jk} + \varepsilon_{\phi} \qquad [17]$$

Where:

- ϕ refers to carrier phase measurements.
- B is the single-differenced carrier-phase biases
- λ is the carrier nominal wavelength.
- ε_{ϕ} are the measurement noise components, including multipath and other effects.
- ρ is the geometrical range between the satellite and the receiver, computed as a function of the satellite (xSat,ySat,zSat) and receiver (xRx,yRx,zRx)

It can be then supposed that there is one or more terms in the equation that is not found by the software.

It is logic to think that the most probable source of error at this time is the receiver used as a rover, the mobile phone, not only because its main application is not receive satellite data, but also because the second option is a national GNSS Station that is been prove that provides valid navigation data at the period analyzed.

A good method of verifying this theory is to compare the Rover RINEX file with the one downloaded from the Station measurements of the same epoch and the same satellite are shown:

Rover:

```
> 2020 10 10 18 55 14.4344718 0 2
G05 23574278.153              -3662.467  40.000
```

GNSS Station:

```
20    10    10    18    55    14.0000000    0
18G05G12G13G15G17G18G19G20G23G24G28G30R05R06R07R09R15R16
123883093.625 7   96532303.26847   23574166.540   23574167.260   43.050   43.000
```

A lot of differences can be detected is just a line of measurements. The first one and less important is the different format of the file. Always that both have all the necessary information this difference does not affect the results.

The second one that appears is the time of the observation. The epoch of observation of the rover is almost half a second different than the one of the GNSS station. As the double-differences algorithm compare those to input observation files, if they are not referred to the same epoch, the comparison could be useless.

But there is one more difference between them, Phone RINEX file does not have carrier phase measurements.

7.7. Solution proposed

Once the study of the issues encountered is done and the error localized, a solution can be proposed.

The first attempt is to try to collect new measurements with the same phone (Xiaomi Pocophone F1). But after doing two more tries with the same result, it can be concluded that another receiver needs to be used.

So, after trying with some other phones, surprisingly this issue came more times than the expected ones. But anyway, at the end, two phones that met the necessary requirements were found. Xiaomi MI8 and Samsung A50.

To sum up, the solution proposed is simple, just changing the receiver to be able to collect carrier phase measurements and try to improve the single method position solution with a master station.

8. Complete measurements

Choosing those new phones as rovers, a complete bunch of measurements can be collected with all the necessary terms in the RINEX files.

The two new rovers are, as previously said, two mobile phones. A Xiaomi MI8 and a Samsung A50. A difference between them can be highlighted, the first rover (Xiaomi) is a dual frequency receiver L1 and L5 and the second one only received data of the L1 frequency band.

Dual frequency GPS receivers offer two major advantages over single frequency equipment. The first one is that ionospheric errors that are inherent in all GPS observations can be modelled and significantly reduced by combining satellite observations made on those different frequencies. Apart from that, observations on two frequencies allow faster ambiguity resolution times.

Usually both these advantages will derive in a better behavior in terms of precision and accuracy. So, it can be expected Xiaomi MI8 to have a better performance than the Samsung model.

Apart from Samsung and Xiaomi, another receiver was used at the same time to collect not only observation, but also navigation data. This receiver is the low-cost u-blox M8T.

This GNSS receiver that support measurements from GPS, Galileo GLONASS and BeiDou constellations in a frequency band. It supports three different constellations to be monitored at the same time.

With all those devices, along with permanent stations, a big number of combinations can be done. In this way, the results can be compared to check which of the devices with which solution mode have a better performance.

8.1. Kinematic solution combinations

In this chapter, the results of each combination processed will be shown with the execution of kinematic type of solution. This is because the use of a master station will get lower errors.

It is interesting to group the position solution representations changing only one input file in the executions. This method permits easily differentiate the quality on the changed input file.

- Rover: Xiaomi, Master: Torino Permanent Station, Navigation: Torino Permanent Station, Position mode: Kinematic

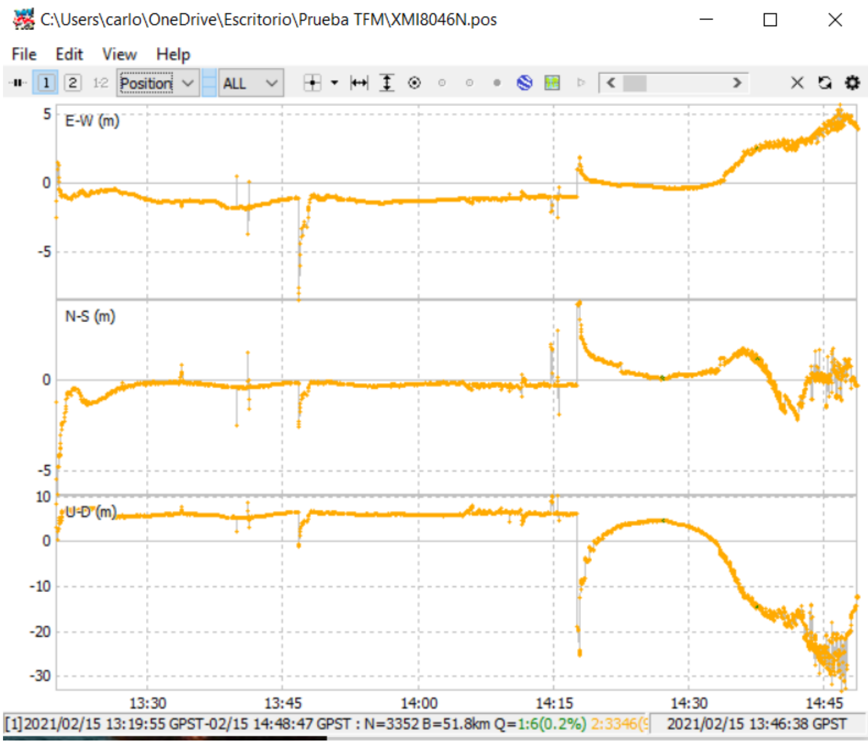


Figure 32 - RTKLIB Kinematic solution plot (Xiaomi -Torino permanent station)

- Rover: Samsung, Master: Torino Permanent Station, Navigation: Torino Permanent Station, Position mode: Kinematic

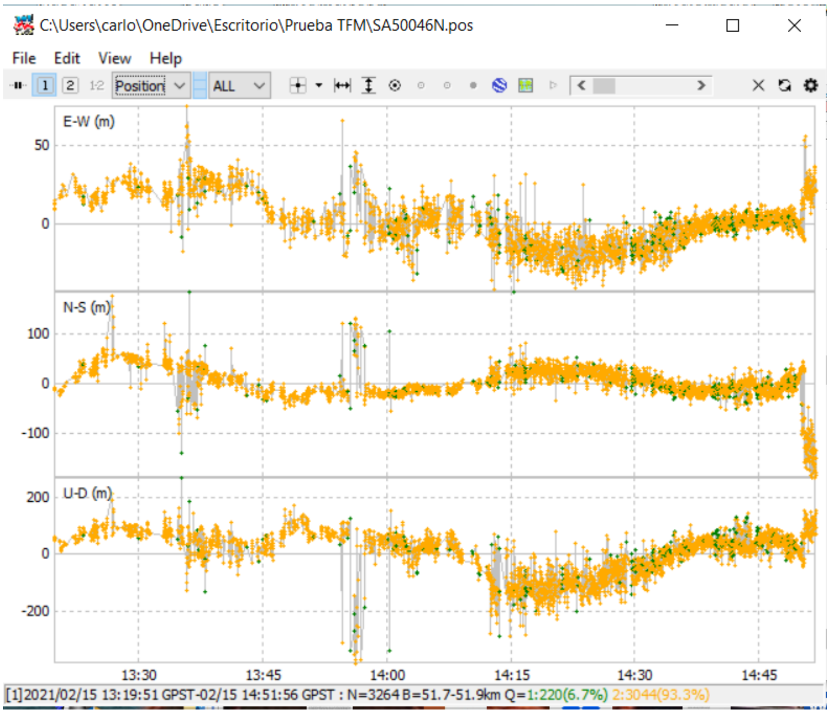


Figure 33 - RTKLIB Kinematic solution plot (Samsung -Torino permanent station)

- <u>Rover</u>: u-blox, <u>Master</u>: Torino Permanent Station, <u>Navigation</u>: Torino Permanent Station, <u>Position mode</u>: Kinematic

Figure 34 - RTKLIB Kinematic solution plot (u-blox -Torino permanent station)

A conclusion that can lead from this group of data is clear, the errors when using the Samsung phone as a rover give higher range of errors than when using the other ones. As it can be seen, this combination gives non-reliable results. With a low accuracy and precision during the hole period of measurements.

With respect to the Xiaomi phone and the u-blox receiver, the results seem to be quite similar at first epochs. In the case of the Xiaomi, it does not keep the same quality during the hole execution.

Continuing with the next group, the combinations are:

- Rover: Xiaomi, Master: u-blox, Navigation: u-blox Position mode: Kinematic

Figure 35 - RTKLIB Kinematic solution plot (Xiaomi - u-blox)

- Rover: Samsung, Master: u-blox, Navigation: u-blox, Position mode: Kinematic

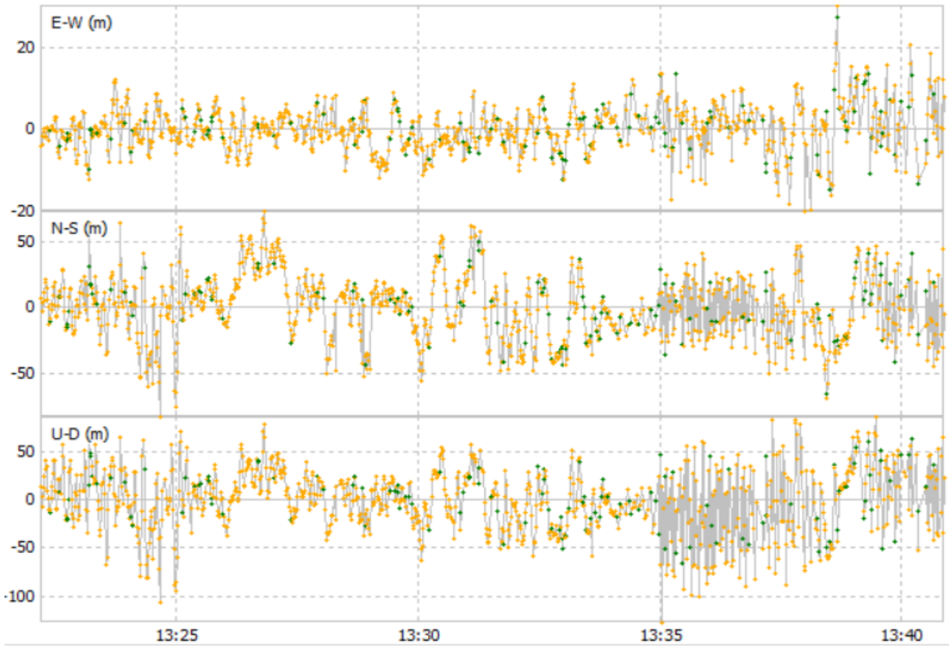


Figure 36 - RTKLIB Kinematic solution plot (Samsung - u-blox)

Once again, in the second combination, it is remarkable the difference between the Xiaomi phone and Samsung as receivers. Being the second much less accurate than the first one.

Another conclusion that can be demonstrated with this second approximation is that the u-blox can be perfectly chosen to use to collect good navigation and observation data. This means that a low-cost device can offer users acceptable position solutions, which is an important milestone for GNSS systems to be included in mechatronic applications.

8.2. Static solution combinations

Another comparison that results interesting to achieve the goal of improving the performance of data post-processing, is to compare which kind of solution obtains better results, kinematic or static.

As previously explained, the static solution algorithm is quite different from the kinematic one. Its purpose is to have a good performance when the receiver is not in motion, which is the case that we are analyzing. That is why we expect a better behavior of the tool when activating this option.

As done in the last chapter, different combinations will be shown.

- Rover: Xiaomi, Master: u-blox, Navigation: u-blox, Position mode: Static

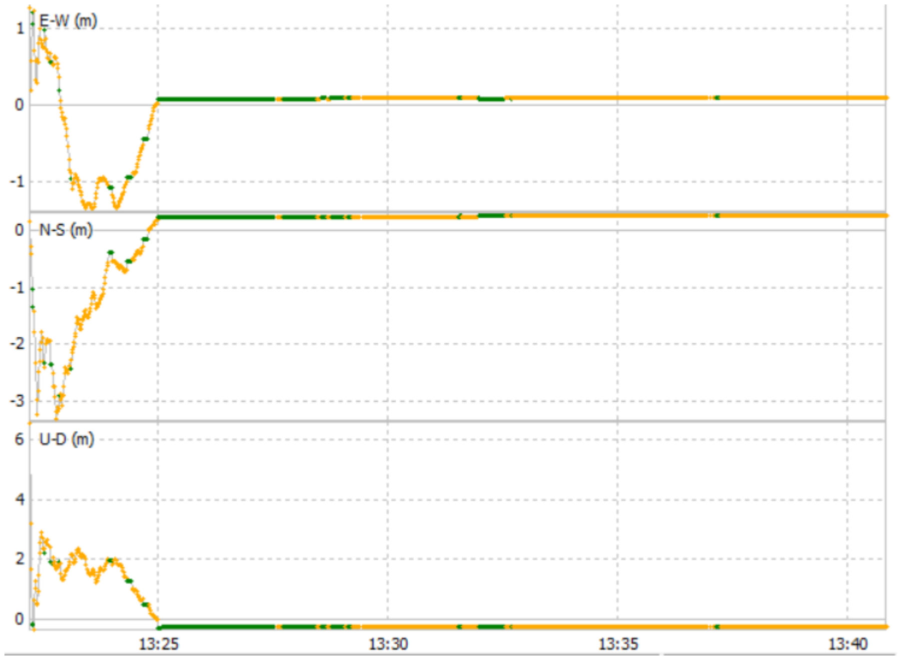


Figure 37 - RTKLIB Static solution plot (Xiaomi -u-blox)

- Rover: Xiaomi, Master: Torino permanent station, Navigation: Torino permanent station, Position mode: Static

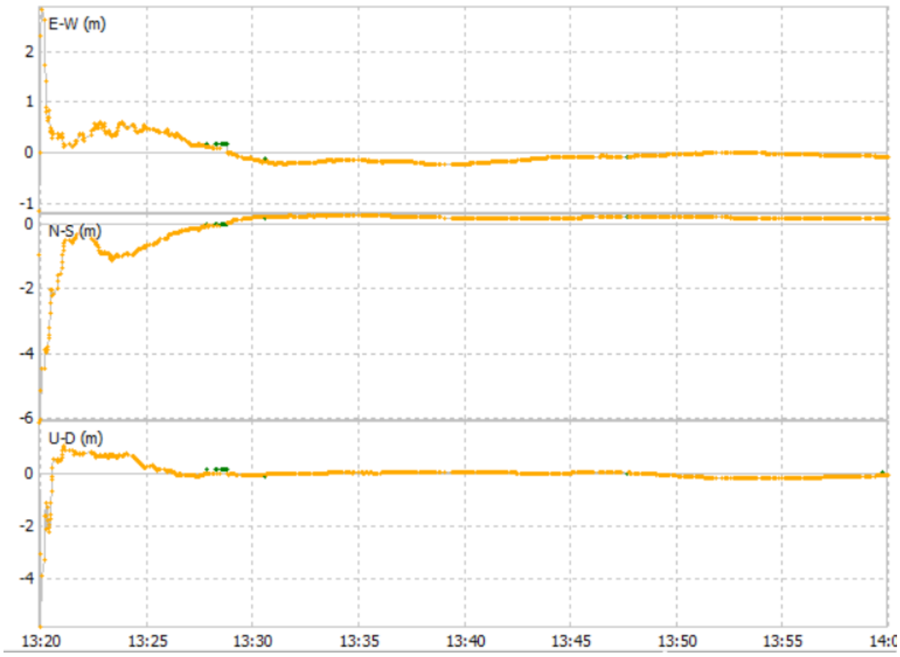


Figure 38 - RTKLIB Static solution plot (Xiaomi – Torino station)

The first approach can be used to prove the expected behavior of Static position mode compared to the kinematic one in this kind of measurements. In these two combinations it is shown that the solution is very accurate once it has been stabilized.

It is also proven that the quality of the u-blox measurements, which have a similar accuracy in their results than the permanent Station of Torino.

- Rover: Samsung, Master: u-blox, Navigation: u-blox, Position mode: Static

Figure 39 - RTKLIB Static solution plot (Samsung - u-blox)

- Rover: u-blox, Master: Torino permanent station, Navigation: Torino permanent station, Position mode: Static

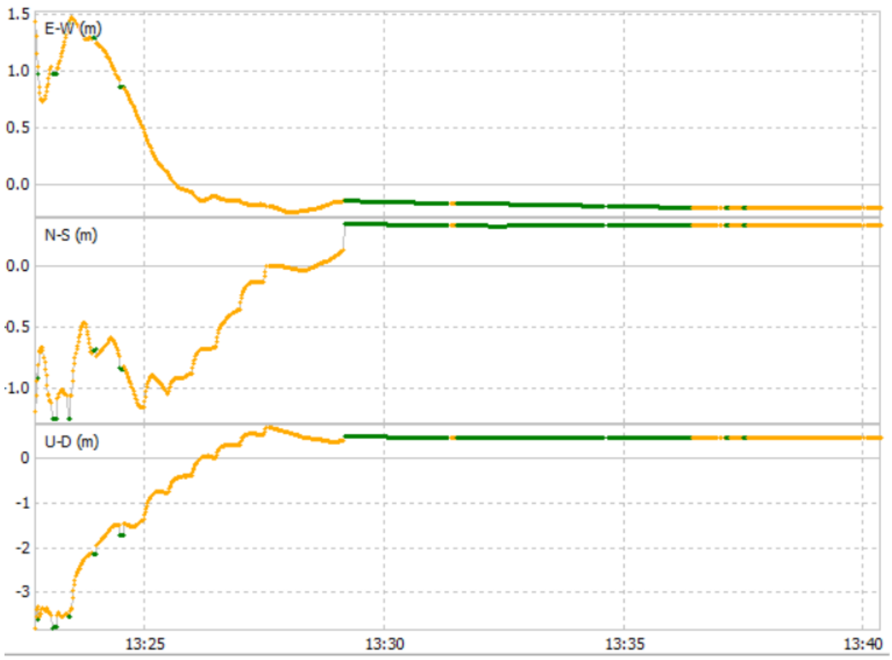


Figure 40 - RTKLIB Static solution plot (Samsung – Torino station)

With all those Static solution combinations, it can be clearly seen that the results improve in terms of lower errors and more stable solutions. At this point the results using the

Xiaomi phone can be considered useful for the applications presented at the beginning of this study.

To assure this first conclusion, a further analysis must be done. In this case, the only way to study the results is from doing a visual inspection of RTKLIB graphic. As this is not very scientific and does not completely prove the validity of the inferences explained, more comparisons will be added to the test.

To this aim, a script has been programmed to use the position solution files as inputs and compare their results with real number and clearer graphics.

8.3. Comparison script

The idea of this script is mostly computing the real errors that this combinations of receivers and solutions are giving. Once those errors are clear, a threshold is set to see which percentage of epochs are inside. Depending on the threshold value and the percentage of epoch with an error below it, it can be considered which devices are valid for which mechatronic applications.

The programing language chosen to develop the script is Batch language, the one of windows command line. As this language does not really complete the requirements needed, Cygwin have been also installed.

Cygwin is an open-source tool that permits extend Batch language with Unix commands, which result in a much more complete language and permits user to carry on multiple functions in terms of file comparison.

Another extension that needs to be installed before execution the script is GNUPLOT, a free tool that is capable of plotting graphics from any kind of text file in an easy way.

The procedure of the script is reading to different inputs, with two different combinations of position solution files, plot both in the same graphic, compute their differences and plot them with a threshold. After this plot a percentage of epochs inside that threshold is also calculated. To clarify this summary the script will be explained part by part.

The first important thing to say is that the inputs chosen to be compared, where the ones that resulted more interesting from the combinations done before. The script can perform four different comparisons at the same time, so the inputs will be eight position solution combinations.

The first comparison is based on contrasting both phones used as rovers. Comparing Xiaomi and Samsung phones using both u-blox as base station and navigation files.

The second part of the script compares kinematic position mode with respect to the Static one. The first combination is done with Xiaomi as a rover and u-blox as master, and the last one using u-blox observation files combined with Torino permanent station ones.

To finish, the program takes the two combinations with better performance and compares them. The idea is to approximate the real error and to determine the quality of the receivers. The way the script is programmed is presented in the next lines to understand how the comparisons are made.

The first part of the script is only saving the paths that are going to be used, apart from linking GNUPLOT and Cygwin to the script with the function *set_environment_variables* to be able to start using those tools.

```
cls
@echo off
COLOR F0
::-----------------------------------------------------------------------------------
::-----------------------------------------------------------------------------------
::------------- Script To compare the solutions of UBlox and both phones-------------
::------------------------Author: Carlos Saro De Aldecoa-----------------------------
::------------------------------Number: S275813--------------------------------------
::-----------------------------------------------------------------------------------
::-----------------------------------------------------------------------------------
call :set_environment_paths
::-----------------------------------------------------------------------------------
set CURRENT=%~dp0
set INPUT_TORI_TORI=%CURRENT%MasterTori_NavTori
set OUTPUT_TORI_TORI=%INPUT_TORI_TORI%\Comparison_Results
set INPUT_UBLOX_UBLOX=%CURRENT%MasterUBlox_NavUBlox
set OUTPUT_UBLOX_UBLOX=%INPUT_UBLOX_UBLOX%\Comparison_Results
set INPUT_UBLOX_STATIC_VS_KINEMATIC=%CURRENT%StaticVSKinematic_Ublox_Tori
set OUTPUT_UBLOX_STATIC_VS_KINEMATIC=%INPUT_UBLOX_STATIC_VS_KINEMATIC%\Comparison_Results
set INPUT_XIAOMI_STATIC_VS_KINEMATIC=%CURRENT%StaticVSKinematic_Xiaomi_Ublox
set OUTPUT_XIAOMI_STATIC_VS_KINEMATIC=%INPUT_XIAOMI_STATIC_VS_KINEMATIC%\Comparison_Results
```

Figure 41 - First part of the comparison script

After that, the script repeats the same structure four times, to compute the different comparisons. Although the hole code will be added as an annex to the thesis, at this point only one of those comparisons will be explained.

```
@echo ----------------------------------------------------------------------------------
@echo ----------------------------------------------------------------------------------
@echo ------------------------------START OF THE COMPUTATION----------------------------
@echo ---------------------ROVER: Xiaomi and UBlox MASTER: Tori------------------
@echo ----------------------------------------------------------------------------------
@echo ----------------------------------------------------------------------------------

@echo First it is necesary to compute RTKLIB post procesing with the input parameters desired.^
 The outputs have to be saved in the path: %INPUT_TORI_TORI%
@echo .
@echo .
@echo Press intro when the RTKLIB outputs are saved in the path %INPUT_TORI_TORI%

@echo .
@echo .
@echo The comparison starts by a change of format of the files to be able to process them:
@echo .
dos2unix %INPUT_TORI_TORI%/UBlox_Tori_Tori.pos && dos2unix %INPUT_TORI_TORI%/Xiaomi_UBlox_UBlox.pos
@echo .
:: Change the epoch value into a comparable value
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}'^
 %INPUT_TORI_TORI%/UBlox_Tori_Tori.pos > %OUTPUT_TORI_TORI%\UBlox_Tori_Tori.tmp
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}'^
 %INPUT_TORI_TORI%/Xiaomi_UBlox_UBlox.pos > %OUTPUT_TORI_TORI%\Xiaomi_UBlox_UBlox.tmp
```

Figure 42 - Second part of the comparison script

The purpose of this second part of the program is to prepare the inputs to be easily processed. The instruction dos2unix is applied to the position solution files to change their format from Windows to Unix, which is necessary because of the nature of the programming language (thought for Linux operating system).

After this format change, there is and instruction call gawk, which is simulating bash awk. Awk is an instruction that permits read files and operate with their parameters very easily.

Every awk instruction has three different parts:

- BEGIN: Is used to initialize variable if necessary, it is just read one time at first.
- BODY: This second part is the one that is used to read the files from beginning to end. It is only necessary to specify the conditions to indicate which parts of the input file need to be read and apply the desired the operations to them.
- END: This last part is only read one time at the end of the program. It can be used to put a final comment or to operate with the variables processed in the middle loop.

It is worth to highlight that is not necessary to include all the parts in each program.

The first awk that appears in the code just prints the desired columns to be analyzed, without the header, in a temporary output file.

```
@echo In order to compare in a better way the results, the geographic coordinates will^
 be proyected into East North and Up (ENU).
@echo .
call :LatLonToENU %OUTPUT_TORI_TORI%\UBlox_Tori_Tori.tmp %OUTPUT_TORI_TORI%\ENU_UBlox_Tori_Tori.tmp
call :LatLonToENU %OUTPUT_TORI_TORI%\Xiaomi_UBlox_UBlox.tmp %OUTPUT_TORI_TORI%\ENU_Xiaomi_UBlox_UBlox.tmp

@echo .
@echo To be able to compare proterly these files, each row of them have to refer to the same epoch.
@echo .
call :OrganizeFiles %OUTPUT_TORI_TORI%\ENU_UBlox_Tori_Tori.tmp %OUTPUT_TORI_TORI%\UBlox_Tori_Tori_Tmp.tmp^
 134562
call :OrganizeFiles %OUTPUT_TORI_TORI%\ENU_Xiaomi_UBlox_UBlox.tmp %OUTPUT_TORI_TORI%\Xiaomi_UBlox_UBlox_Tmp.tmp^
 134562
```

Figure 43 - Third part of the comparison script

As the input file of the script in the RTKLIB output, the position coordinates are measured in latitude, longitude, and height. To make the comparison more graphic from the user point of view, they will be projected into ENU coordinates (east, north, up). To do so, a function must be applied, and it is called *LatLonToENU*.

It is worth to explain a little bit of theory at this point. The idea of this transformation is no other than giving the user a more intuitive way of comparison.

The reference system used by RTKLIB in the position solution files, is the geographic coordinate system. In this system all positions or earth surface can be determined by three components.

- Latitude: The latitude of a point on earth is the angle between the equatorial plane and the line that crosses that point with the center of the earth. All the points with the same latitude form a parallel plane with the equator. Its values can be from -90 to +90 degrees.
- Longitude: The longitude of a point on earth in the angle between the reference meridian and the meridian that passes through that point. The values can be from 0 to 180 degrees.
- Altitude: The altitude of a point on earth is the elevation of that point with respect the sea level.

On the other hand, ENU coordinates represent a reference frame that projects the three-dimensional components into a plane. It is commonly used in aviation.

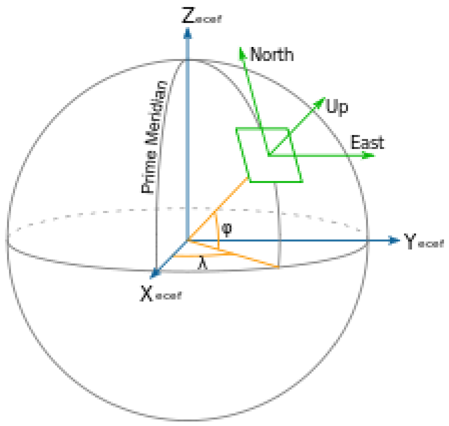


Figure 44 - Schema of ENU coordinates

In the case of height or up component, it is the same as the altitude described before. But to compute the other two of them, a sequence of operations must be computed.

To convert from geodetic coordinates to local tangent plane (ENU) coordinates is a two-stage process:

1. Convert geodetic coordinates to ECEF coordinates
2. Convert ECEF coordinates to local ENU coordinates

In the next part of the code, this process is done and all the equations used to achieve this conversion can be found.

```
:LatLonToENU
:: Parameter 1 -> Input file
:: Parameter 2 -> Output file
:: --------------------------------
gawk 'BEGIN {^
        Pi = 3.14159265359;^
        H = 32;^
        Major_Axis = 6378137;^
        Minor_Axis = 6356752.31424518;^
        Ecce = 0.0818191908426203;^
        Ecce2 = 0.0820944379496945;^
        EcceEl2 = 0.00673949674227624;^
        CurRadius = 6399593.62575849;^
    }^

    {^
        LatRad = $3 * Pi / 180;^
        LonRad = $4 * Pi / 180;^
        MeridianH = (6 * H) - 183;^
        DLambda = LonRad - ((MeridianH * Pi)/180);^
        A = cos(LatRad) * sin(DLambda);^
        Xi = 0.5 * log((1 + A)/(1 - A));^
        PreEta = (((sin(LatRad) / cos (LatRad))) / cos(DLambda)) - LatRad;^
        Eta = atan2(sin(LatRad) / cos(LatRad),cos(DLambda)) - LatRad;^
        Ni = (CurRadius/(sqrt(1 + EcceEl2 * cos(LatRad) * cos(LatRad)))) * 0.9996;^
        Zeta = (EcceEl2 / 2) * Xi * Xi * cos(LatRad) * cos(LatRad);^
        A1 = sin(2 * LatRad);^
        A2 = A1 * cos(LatRad) * cos(LatRad);^
        J2 = LatRad + (A1 / 2);^
        J4 = ((3 * J2) + A2) / 4;^
        J6 = (5 * J4 + A2 * (cos(LatRad) * cos(LatRad))) / 3;^
        Alfa = 3 / 4 * EcceEl2;^
        Beta = (5 / 3) * Alfa * Alfa;^
        Gamma = (35 / 27) * Alfa * Alfa * Alfa;^
        Fi = 0.9996 * CurRadius * (LatRad - (Alfa * J2) + (Beta * J4) - (Gamma * J6));^
        East = (Xi * Ni * (1 + (Zeta / 3))) + 500000;^
        North = Eta * Ni * (1 + Zeta) + Fi;^
        printf("%%s %%s %%f %%f %%s %%s\n",$1,$2,East,North,$5,$6);^
}' %~1 > %~2

goto :EOF
```

Figure 45 – "LatToENU" function of the comparison script

Here it can be seen a more complete example of an awk program. In the BEGIN instruction, all the constants are settled, and then, used in the main part to process the coordinates of all epochs in the input file. The result is another temporary file that is used as an input for the next step.

Once this coordinate transformation is done, the files have to be organized to be able to compare them properly. This is because the RTKLIB position solution files are not of the same length and there are some epochs in each of them that do not appear in it, because the tool could not compute a solution of them.

The way of comparison that will be done is to put both position solution files in the same one to be able to easily compare two columns of it. But this is only possible if each the rows refer to the same epoch in those two files. That is why in the organize function the epochs without solution will be filled as zeros.

```
:OrganizeFiles
:: Parameter 1 -> Input file
:: Parameter 2 -> Output file
:: Parameter 3 -> Initial Epoch
:: --------------------------------
gawk 'BEGIN {^
        Epoch=%~3;^
    }^
    {^
    if (Epoch ^=^= $2) {^
        printf("%%s %%s %%s %%s %%s\n",$2,$3,$4,$5,$6);^
        Epoch++;^
    }^
    else {^
        EpochsDif=$2-Epoch;^
        for (i=0;i^< EpochsDif;i++){^
            printf("%%s %%s %%s %%s %%s\n",Epoch,0,0,0,0);^
            Epoch++;^
        }^
        printf("%%s %%s %%s %%s %%s\n",$2,$3,$4,$5,$6);^
        Epoch++;^
    }^
}' %~1 > %~2

goto :EOF
```

Figure 46 - "OrganizeFiles" function of the comparison script

Once both files are organized in the correct manner, they can be pasted in another file which will contain ten columns with the desired information. Epoch, East, North, Height and the type of solution achieved, 1 referred to Fixed and 2 to Float solutions. This file will be used by the script to plot all the graphics and it will also be found as an output file called *CompareEpochs.dat.*

```
call :plotDifColors  %OUTPUT_TORI_TORI%\CompareEpochs.dat %OUTPUT_TORI_TORI%\EAST_UBloxVSXiaomi.png "East Comparison" "East(m)"^
 2 346067 346070 134562 135622
@echo .
@echo .
call :plotDifColors  %OUTPUT_TORI_TORI%\CompareEpochs.dat %OUTPUT_TORI_TORI%\NORTH_UBloxVSXiaomi.png "North Comparison" "North"^
 3 4972738 4972744 134562 135622
@echo .
@echo .
call :plotDifColors  %OUTPUT_TORI_TORI%\CompareEpochs.dat %OUTPUT_TORI_TORI%\HEI_UBloxVSXiaomi.png "Height Comparison" "Height"^
 4 1497 1504.3 134562 135622
@echo .
```

As previously said, the file CompareEpochs.dat is used by the script to plot the different comparison graphics. This task can be achieved by the function *PlotDifColors,* which takes as input apart from the file, the desired legends to be printed and the axis values.

```
:plotDifColors
:: Parameter 1 -> Input file containing the data to plot
:: Parameter 2 -> Path of the figure
:: Parameter 3 -> Title
:: Parameter 4 -> Y label
:: Parameter 5 -> Column of Ublox data
:: Parameter 6 -> Column of Xiaomi data
:: Parameter 7 -> Minimum valueY
:: Parameter 8 -> Maximum valueY
:: Parameter 9 -> Minimum valueX
:: Parameter 10 -> Maximum valueX
:: ---------------------------------
set GNUPLOT_FILE=temp.gpl
set VAR_FILE=%~1
set FIGURE_NAME=%~2
set /a SEC_COL=%~5 + 5


@echo set title "%~3" > %GNUPLOT_FILE%
@echo set autoscale >> %GNUPLOT_FILE%
@echo set datafile separator " " >> %GNUPLOT_FILE%
@echo set grid >> %GNUPLOT_FILE%
@echo set ylabel "%~4" >> %GNUPLOT_FILE%
@echo set xlabel "Epoch (sec)" >> %GNUPLOT_FILE%
@echo set xrange [%~8:%~9] >> %GNUPLOT_FILE%
@echo set yrange [%~6:%~7] >> %GNUPLOT_FILE%
@echo set term png size 1024,768 >> %GNUPLOT_FILE%
@echo set output "%FIGURE_NAME:\=/%" >> %GNUPLOT_FILE%
@echo plot "%VAR_FILE:\=/%" u 1:($5==2?$%~5:1/0) w p lc rgb "#F9AF07" title "UBlox Float Solution",^
 "%VAR_FILE:\=/%" u 1:($5==1?$%~5:1/0) w p lc rgb "#569A0C" title "UBlox Fixed Solution",^
 "%VAR_FILE:\=/%" u 1:($10==2?$%SEC_COL%:1/0) w p lc rgb "#41EDEC" title "Xiaomi Float Solution",^
 "%VAR_FILE:\=/%" u 1:($10==1?$%SEC_COL%:1/0) w p lc rgb "#AE02FD" title "Xiaomi Fixed Solution" >> %GNUPLOT_FILE%
gnuplot -persist < %GNUPLOT_FILE:\=/% > nul 2>&1
@echo "<FIGURE %FIGURE_NAME%>"
del /q %GNUPLOT_FILE%
goto :EOF
```

Figure 47 - "plotDifColor" function of the comparison script

The tool that is used to plot is GNUPLOT, and it have its own programing language. So, the script will send the commands to an intermediate file that will be processed by GNUPLOT to generate the outputs graphics. The most interesting part of the function is that can detect the value of the column that specifies if the solution is fixed or float and plot a different color in each case. This makes sense as the proper way of comparing the same epoch is when the solution achieved with different inputs are both fixed or both floats.

```
call :computeDiferences %OUTPUT_TORI_TORI%\CompareEpochs.dat %OUTPUT_TORI_TORI%\SolutionDiferences.dat 1 -1
call :computePercentages %OUTPUT_TORI_TORI%\SolutionDiferences.dat %OUTPUT_TORI_TORI%\Percentages.dat

call :plotDiferences  %OUTPUT_TORI_TORI%\SolutionDiferences.dat %OUTPUT_TORI_TORI%\DIF_EAST_UBloxVSXiaomi.png^
 "Diferences between Ublox and Xiaomi" "East" 2 -10 10 134562 135622
@echo .
@echo .
call :plotDiferences  %OUTPUT_TORI_TORI%\SolutionDiferences.dat %OUTPUT_TORI_TORI%\DIF_NORTH_UBloxVSXiaomi.png^
 "Diferences between Ublox and Xiaomi" "North" 3 -10 10 134562 135622
@echo .
@echo .
call :plotDiferences  %OUTPUT_TORI_TORI%\SolutionDiferences.dat %OUTPUT_TORI_TORI%\DIF_HEIGHT_UBloxVSXiaomi.png^
 "Diferences between Ublox and Xiaomi" "Height" 4 -10 10 134562 135622
@echo .
@echo .
del /q %OUTPUT_TORI_TORI%\*.tmp
@echo .
@echo .
@echo The comparison between both phones and the Ublox as rovers is done
@echo The results can be found in the path: %OUTPUT_TORI_TORI%
```

Figure 48 - Fourth part of the comparison script

In the picture above, it can be seen the last part of the comparison. It can be in turn divided in another two parts, computation, and plot. In this case the computation done is simply a subtraction of two terms, each of them of one of the solutions plotted before. Apart from that a threshold is settled at this time.

To check if the measurements a good enough, once the threshold is defined, the percentage of epochs that meet the requirement in terms of east, north and height coordinates is also computed and printed in an output file called *Percentages.dat.*

The last function that is performed by the script, is plotting the calculations done just before. In this case, the function *PlotDifferences* is in charge of doing it.

```
:plotDiferences
:: Parameter 1 -> Input file containing the data to plot
:: Parameter 2 -> Path of the figure
:: Parameter 3 -> Title
:: Parameter 4 -> Y label
:: Parameter 5 -> Column of Ublox data
:: Parameter 6 -> Column of data
:: Parameter 7 -> Minimum valueY
:: Parameter 8 -> Maximum valueY
:: Parameter 9 -> Minimum valueX
:: Parameter 10 -> Maximum valueX
:: --------------------------------
set GNUPLOT_FILE=temp.gpl
set VAR_FILE=%~1
set FIGURE_NAME=%~2


@echo set title "%~3" > %GNUPLOT_FILE%
@echo set autoscale >> %GNUPLOT_FILE%
@echo set datafile separator " " >> %GNUPLOT_FILE%
@echo set grid >> %GNUPLOT_FILE%
@echo set ylabel "%~4" >> %GNUPLOT_FILE%
@echo set xlabel "Epoch (sec)" >> %GNUPLOT_FILE%
@echo set xrange [%~8:%~9] >> %GNUPLOT_FILE%
@echo set yrange [%~6:%~7] >> %GNUPLOT_FILE%
@echo set term png size 1024,768 >> %GNUPLOT_FILE%
@echo set output "%FIGURE_NAME:\=/%" >> %GNUPLOT_FILE%
@echo plot "%VAR_FILE:\=/%" u 1:($5==2?$%~5:1/0) w p lc rgb "#F9AF07" title "Diferences in Float Solutions",^
 "%VAR_FILE:\=/%" u 1:($5==1?$%~5:1/0) w p lc rgb "#569A0C" title "Diferences in Fixed Solutions",^
 "%VAR_FILE:\=/%" u 1:7 w l linewidth 3 lc rgb "#FF0000" notitle,^
 "%VAR_FILE:\=/%" u 1:6 w l linewidth 3 lc rgb "#FF0000" title "Threshold"  >> %GNUPLOT_FILE%
gnuplot -persist < %GNUPLOT_FILE:\=/% > nul 2>&1
@echo "<FIGURE %FIGURE_NAME%>"
del /q %GNUPLOT_FILE%
goto :EOF
```

Figure 49 - "plotDiferences" function of the comparison script

Now that the script purpose is completed understood, the next step is to prove its functionality with real data.

9. Second Analysis

To make a more precise comparison, and to be able to reaffirm the previous ones and draw new conclusions, the same collection of measurements that were processed at the beginning of this section will be used as inputs of the script. As before, the most representative combinations have been chosen.

9.1. Xiaomi VS Samsung as rovers

The first case to be analyze is probably the most realistic one. In these combinations all the devices are portable and do not need from any master station to compute a position solution in kinematic or static mode.

Both executions are done with the u-blox as the master station and navigation data and, as the title says, Samsung and Xiaomi phone as rovers. The solution will be computed in static mode.

The first graphs that are going to be shown are the plot of the three coordinates, east, north and height of both executions.

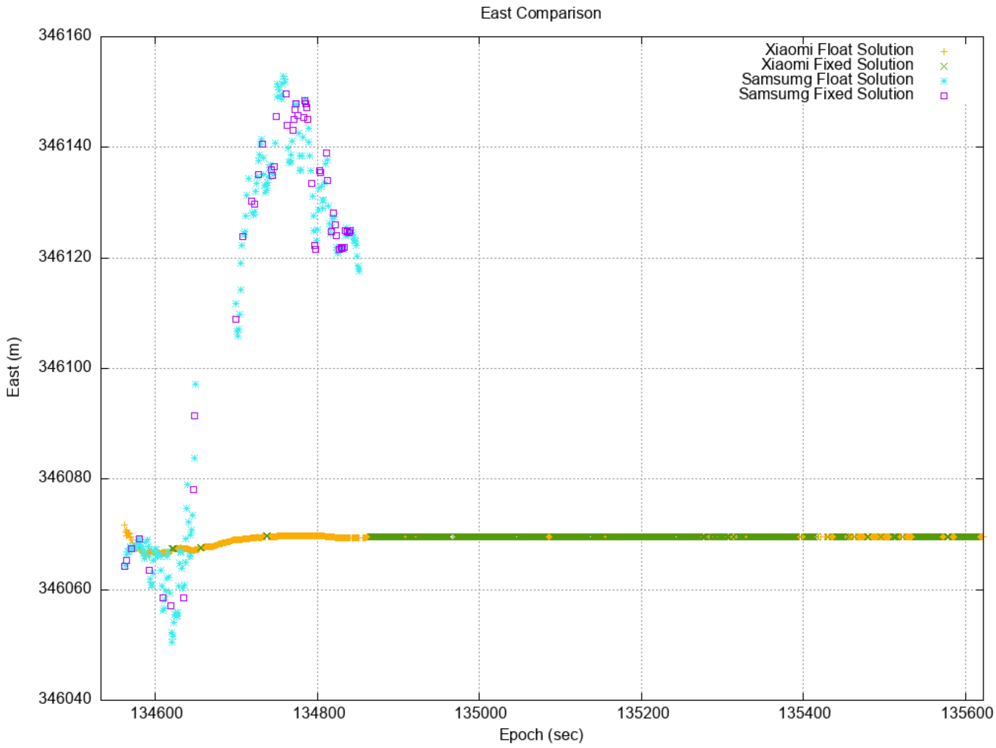


Figure 50 – Simulation 1: East comparison of Xiaomi and Samsung as rovers

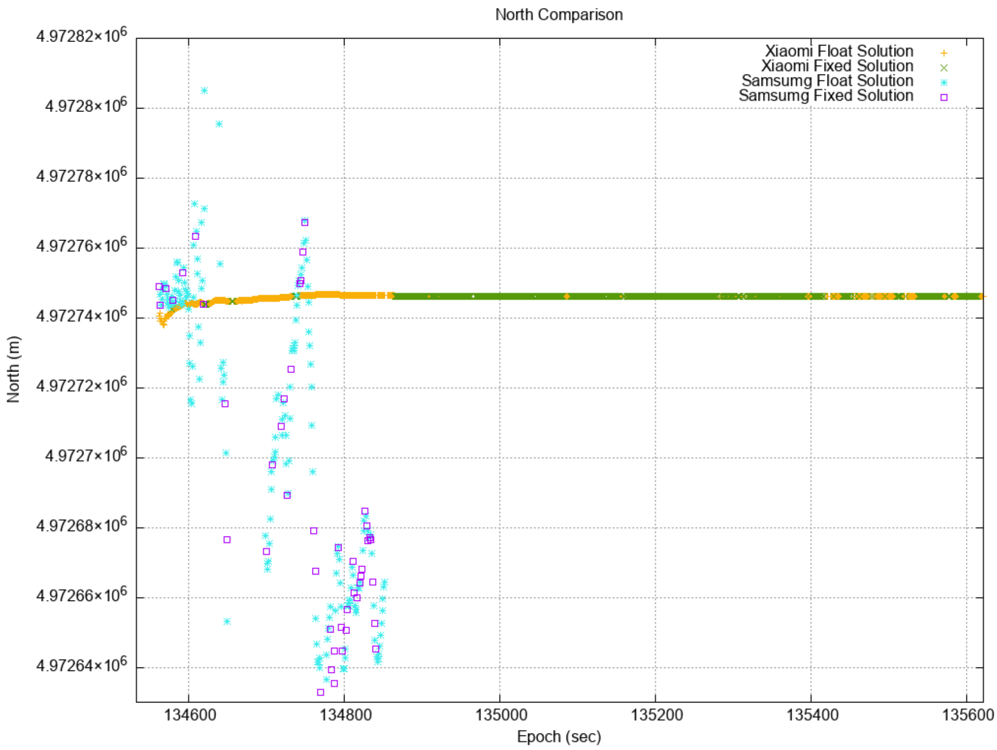


Figure 51 - Simulation 1: North comparison of Xiaomi and Samsung as rovers

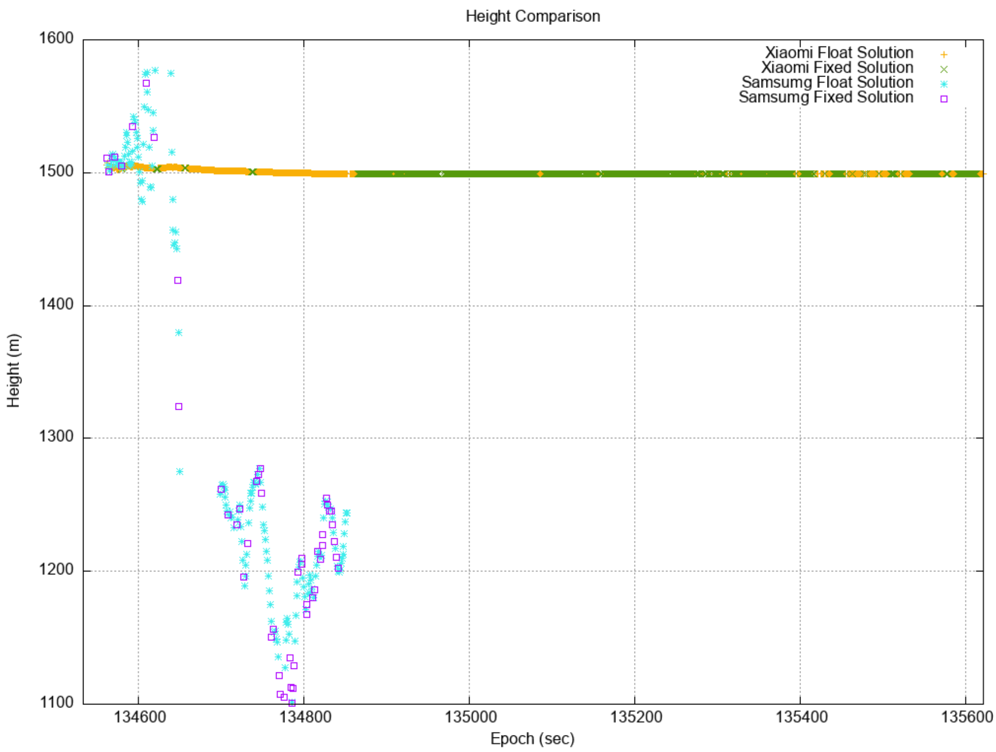


Figure 52 - Simulation 1: Height comparison of Xiaomi and Samsung as rovers

The first reasoning that can be deducted from the graphs is the lack of data that the Samsung solution have. In most of the epochs of the observation period, RTKLIB is not able to compute a solution in static mode.

Apart from this first issue, in the epochs that there is solution, the results have big errors and vary a lot. Because of that bad behavior, a high threshold must be set to see if errors are mostly inside it. It was decided to put a 20 meters threshold just to check if Samsung receiver could be useful for some applications in which accuracy is not a critical factor.

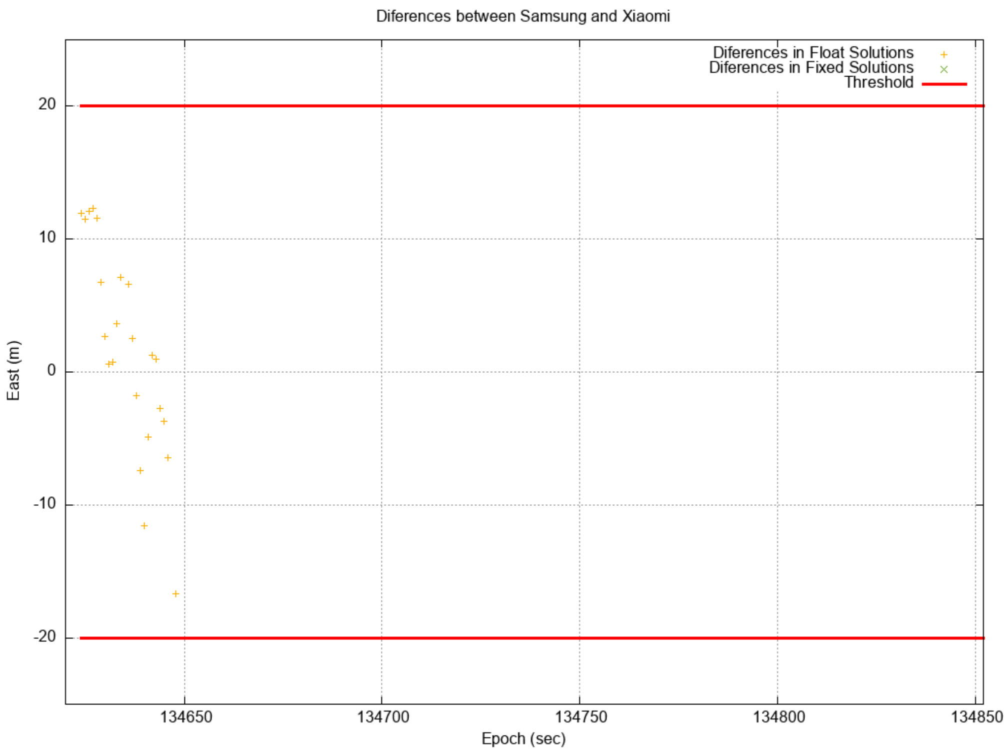


Figure 53 - Simulation 1: East differences between Xiaomi and Samsung as rovers

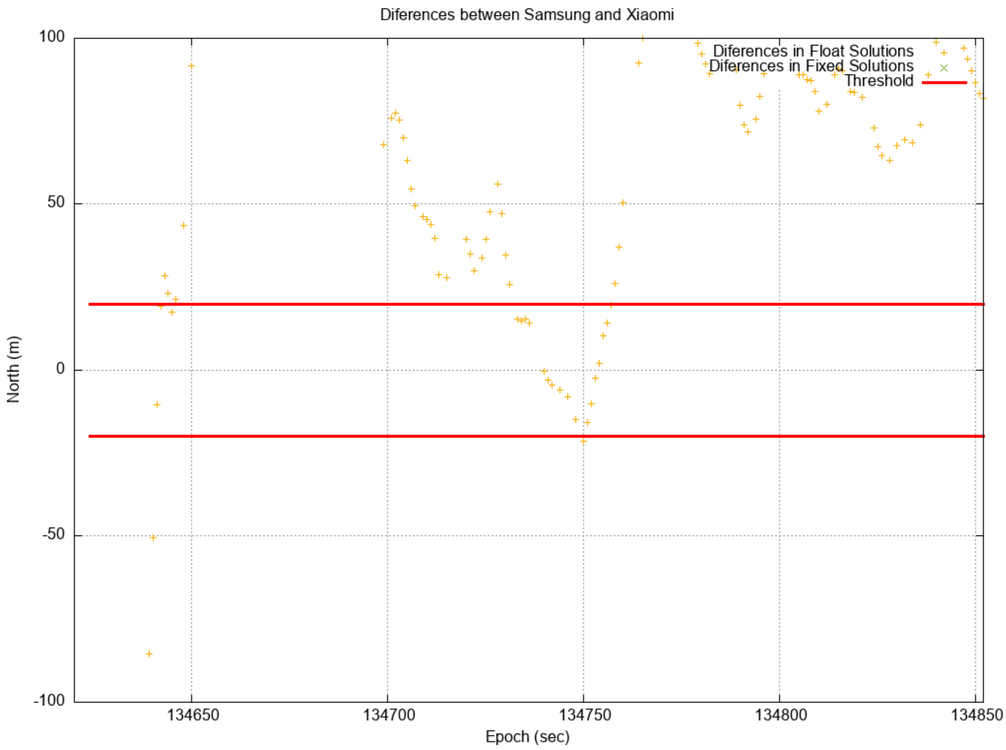


Figure 54 - Simulation 1: North differences between Xiaomi and Samsung as rovers

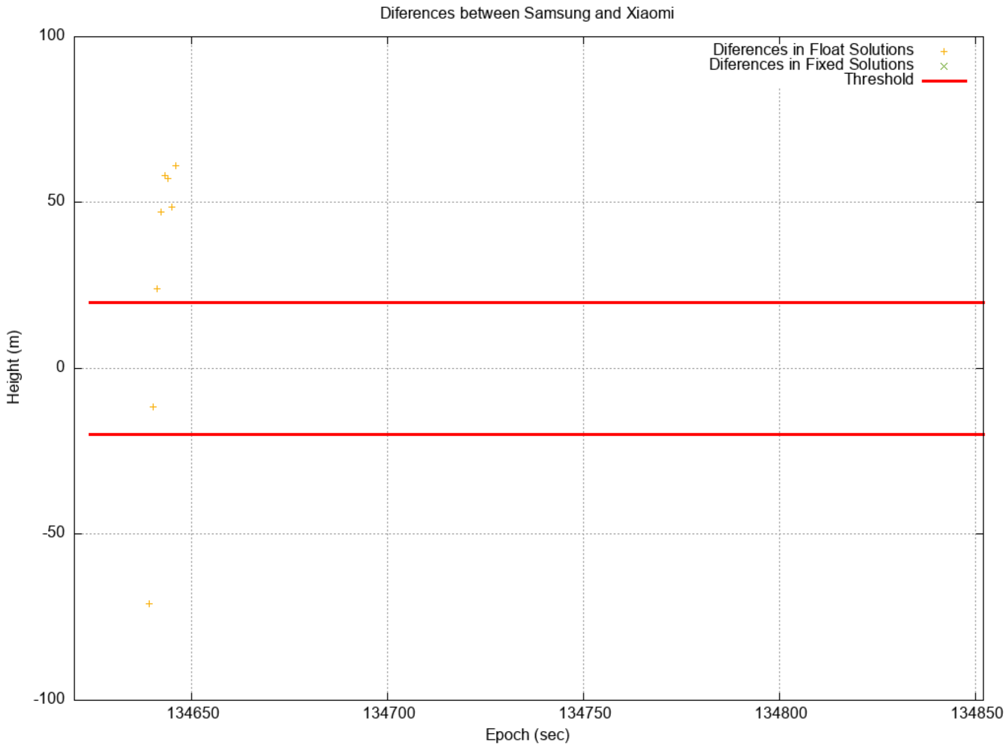


Figure 55 - Simulation 1: Height differences between Xiaomi and Samsung as rovers

Apart from visual inspection it is also important to compute real data to prove if indeed the conclusions drawn are true or not. The next table summarizes the percentage of epochs inside the threshold of 20 meters.

East Percentage	North percentage	Height percentage
18,6 %	16,2 %	1.6 %

Table 1 - Simulation 1: Percentages of first comparison

An important conclusion can be deducted from this second comparison, Samsung phone as a GNSS receiver does not have an accurate performance. Not only was not able to compute solution in the hole period but also the ones computed were not precise.

The main differences between Xiaomi and Samsung receivers, is that the first one can receive signals from two frequency bands, L1 and L5, which does not happen in the second one, that only is capable of receive L1 signals.

This difference could be an important factor to explain the reason why there is such a contrast in the behavior of both tools. Of course, there can be another important factor, like the quality of the materials, that are not in the scope of this analysis.

9.2. Kinematic VS Static with Xiaomi as rover

In this case, the comparison will rely on the solution modes of the Xiaomi receiver. The navigation and Observation data used apart is from the u-blox. This solution it is also using input date from portable devices, as is the case of Xiaomi phone (used as a rover) and the u-blox (as the base station).

During the first approach, doing a visual inspection to the RTKLIB plots, it was concluded that the behavior of the Static solution mode gave a better performance than the Kinematic mode. At this point, it results interesting to see how much those post-processing methods differ from each other.

Samsung phone has been discarded in this study as the measured data cannot be considered useful. That is why only Xiaomi and u-blox will be used as rovers.

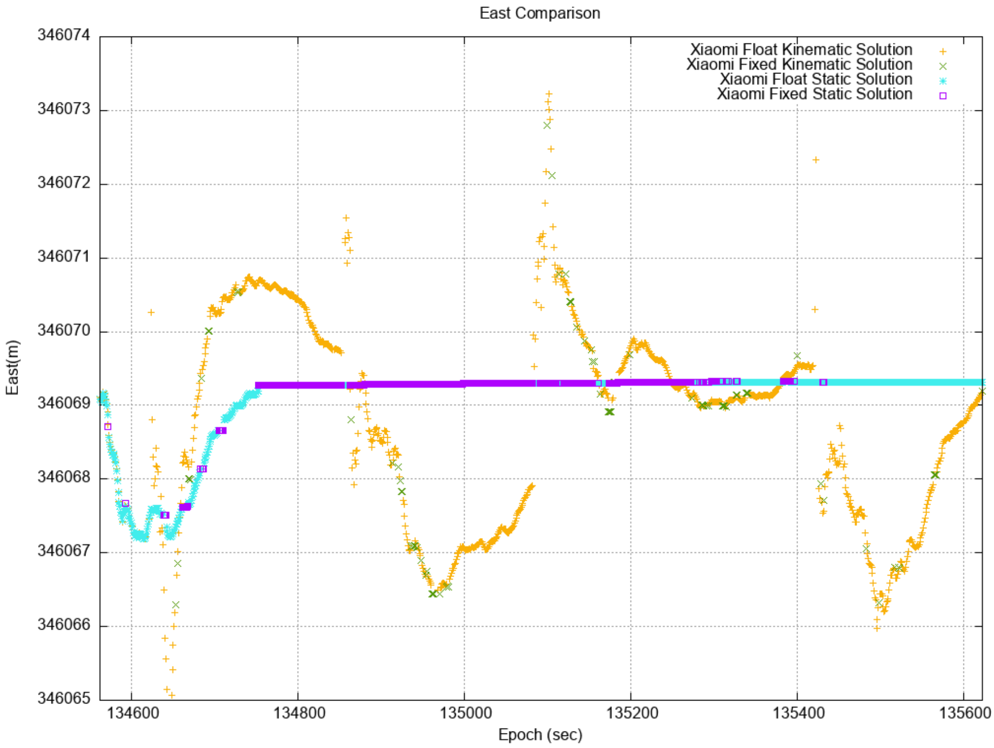


Figure 56 - Simulation 1: East kinematic and static solutions using Xiaomi

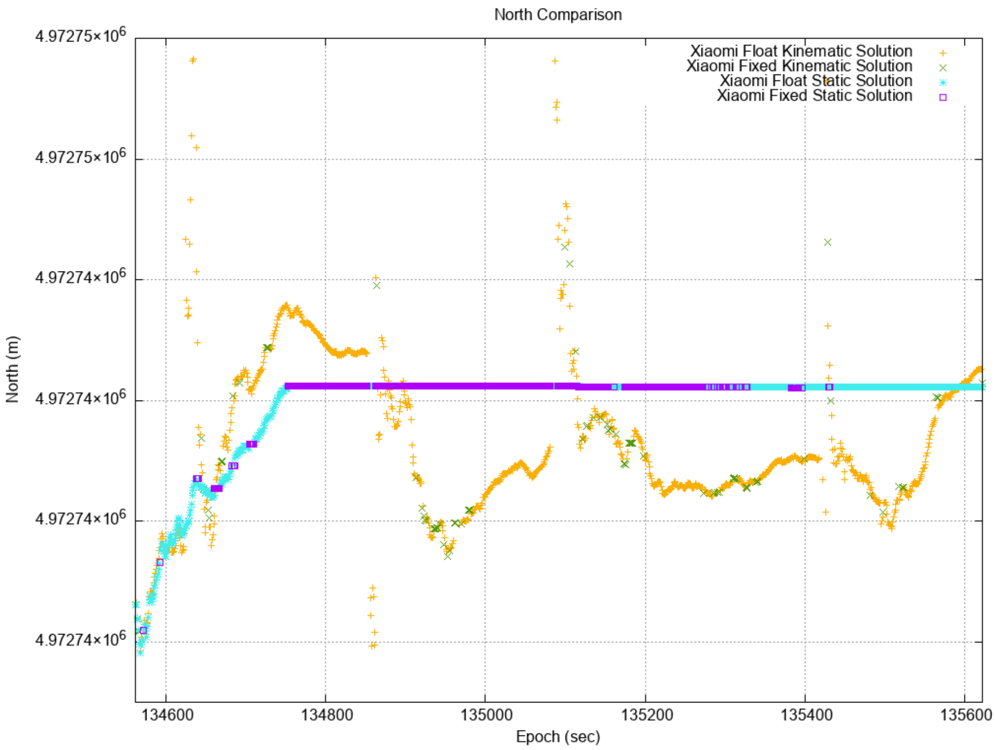


Figure 57 - Simulation 1: North kinematic and static solutions using Xiaomi

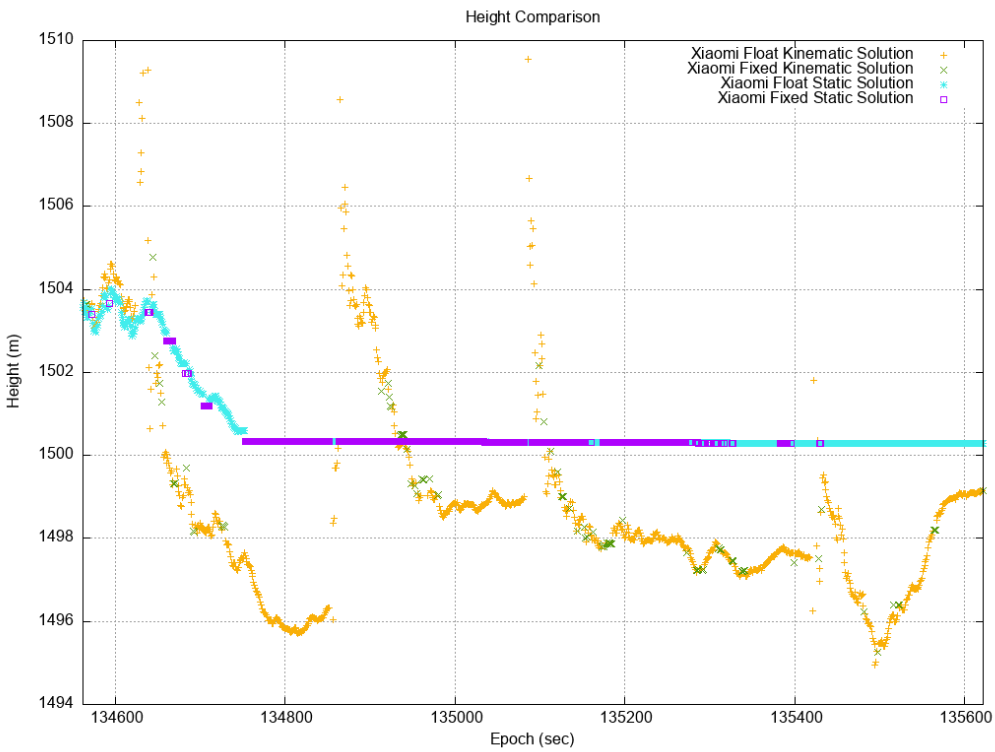


Figure 58 - Simulation 1: Height kinematic and static solutions using Xiaomi

With these graphs, the first conclusion can be reaffirmed. Static solution behaves better that the Kinematic one, at least when using the Xiaomi receiver.

Apart from the first epochs in which the tool is converging to a stable state, the rest of the solution in the static mode do not suffer from any variation. This means that the error is very low.

This approach can be used to prove the quality of the input data provided by the Xiaomi phone and the u-blox. In this case, the criteria to see if the solutions are good enough if the same than in the first comparison, so threshold is set in 2 meters.

The epochs compared this time are only the ones that have the same type of solution, since it does not make much sense to compare a fix with a float solution.

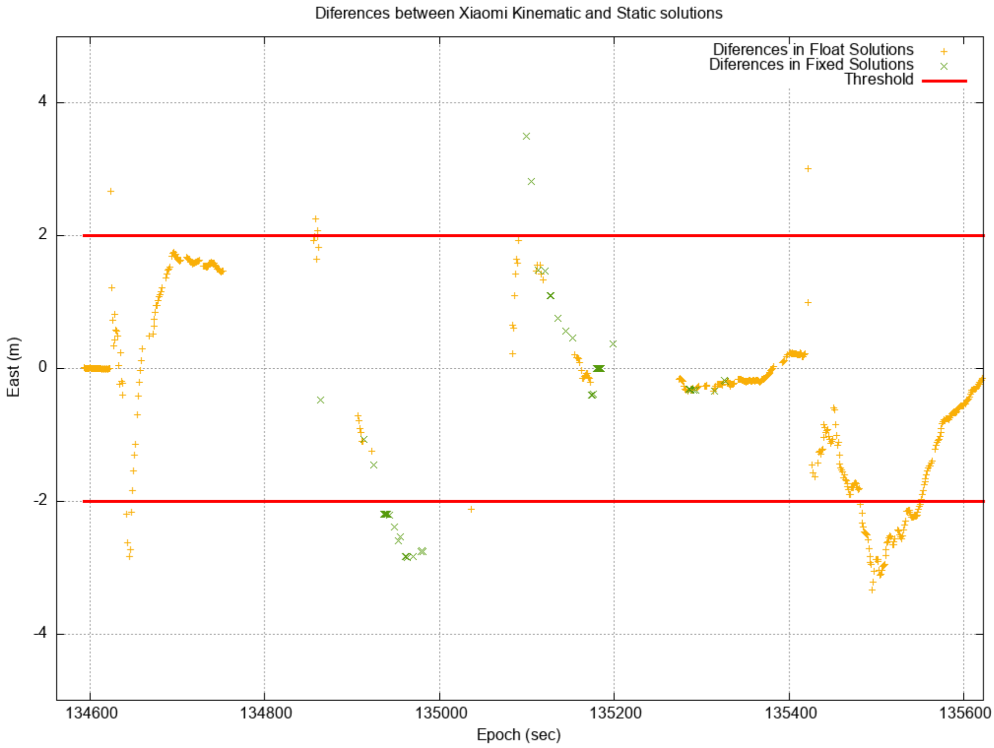


Figure 59 - Simulation 1: East differences between solutions using Xiaomi

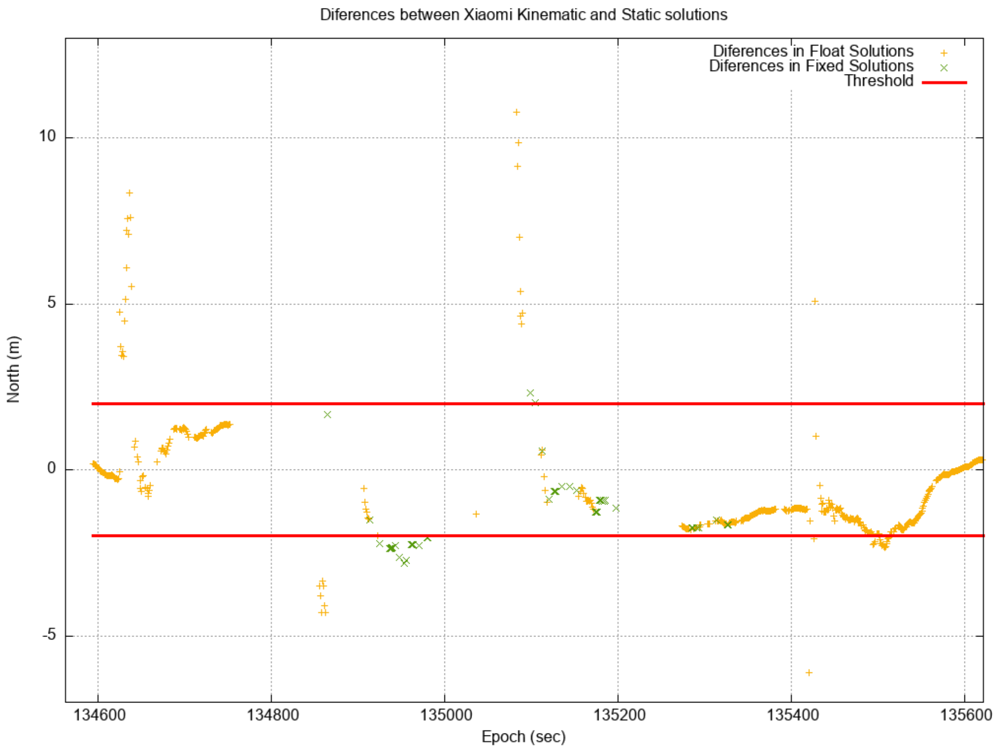


Figure 60 - Simulation 1: North differences between solutions using Xiaomi

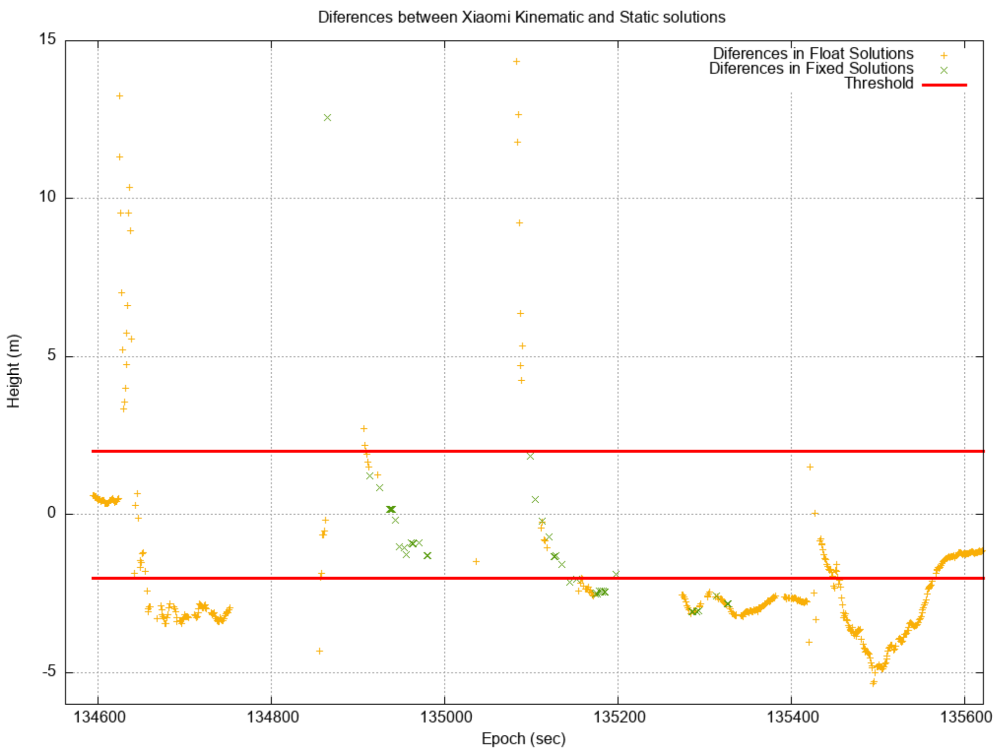


Figure 61 - Simulation 1: Height differences between solutions using Xiaomi

East Percentage	North percentage	Height percentage
84,2 %	88,9 %	41,4 %

Table 2 – Simulation 1: Percentages of second comparison

Even thought it was clearly seen that the static solution is better, the truth is that the errors when using the kinematic mode are still low. In the case of east and north coordinates errors, they can still be useful for some mechatronic purposes. Apart from the errors, Kinematic solution does not have a stable behavior and varies a lot during the period of measurement. This issue can be a barrier when it comes to it´s application to other fields.

As in the first comparison, the most unprecise component is the height coordinate, which is a known difficulty of GNSS systems.

9.3. Kinematic VS Static with u-blox as rover

The last combination in terms of solution mode that was analyzed in detail, was the use of u-blox as a Rover. As before, it is expected to obtain better results with the Static solution, but the importance at this point is to also measure the differences between the accuracy of each of them and the possible applications that they have.

The threshold selected at this time is also two meters, keeping again with the criteria defined above.

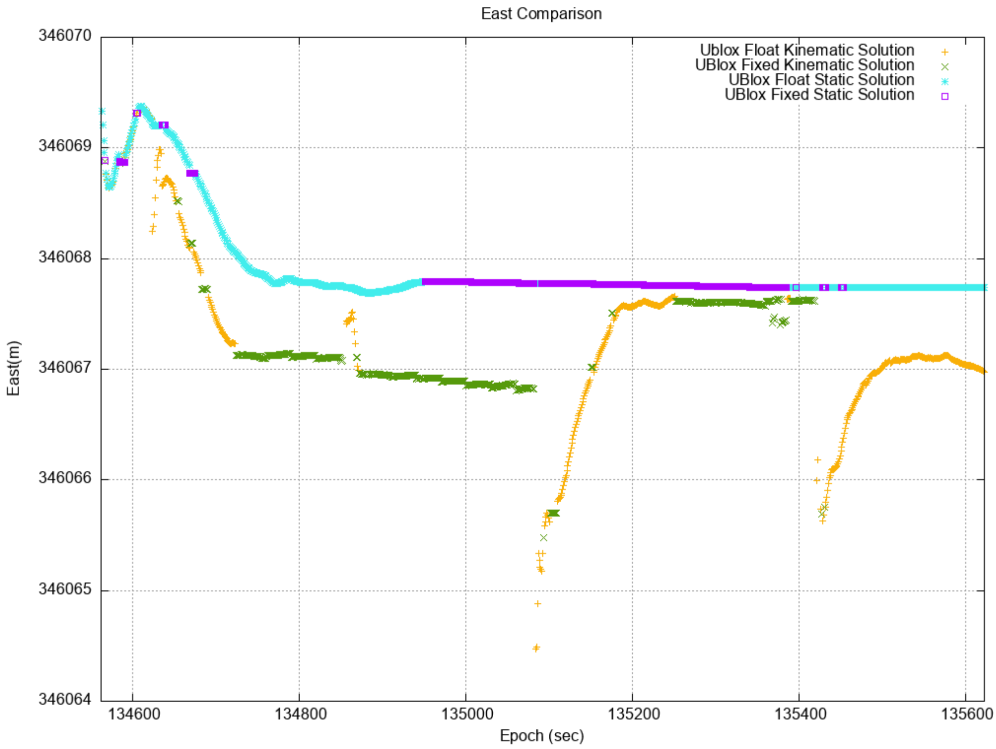


Figure 62 - Simulation 1: East kinematic and static solutions using u-blox

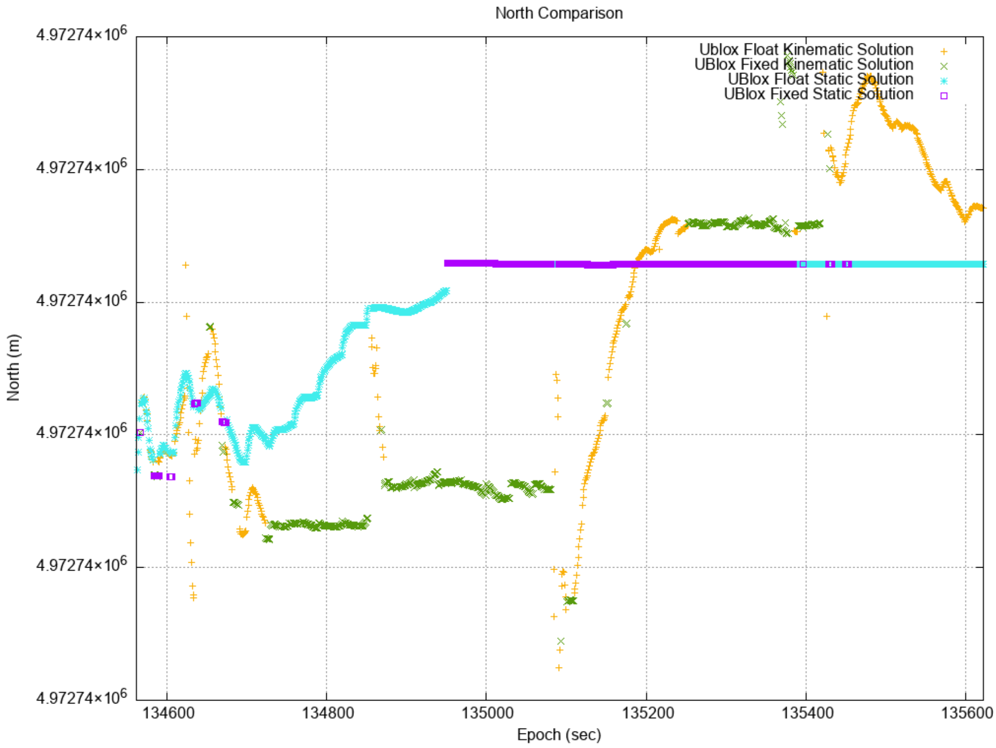


Figure 63 - Simulation 1: North kinematic and static solutions using u-blox

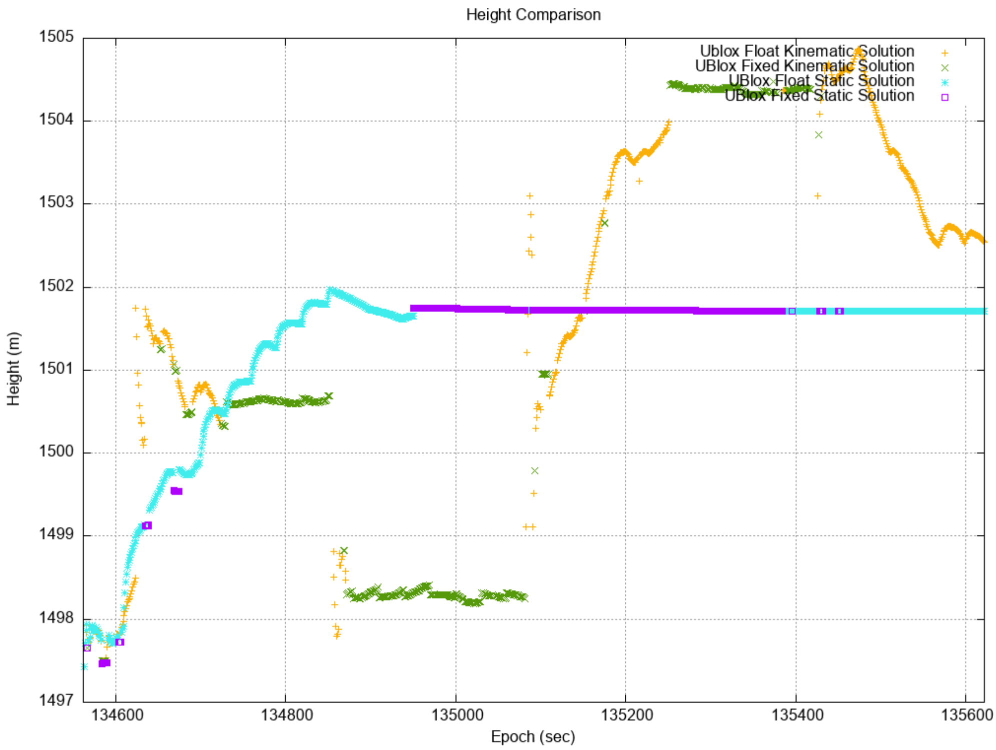


Figure 64 - Simulation 1: Height kinematic and static solutions using u-blox

The conclusion could be the same in this second Kinematic versus Static comparison. In all the coordinates, the static solution is behaving better. It seems obvious that this is the best approach to be used. To finish and check how much do these combinations differ, lest keep on with the last part of the script.

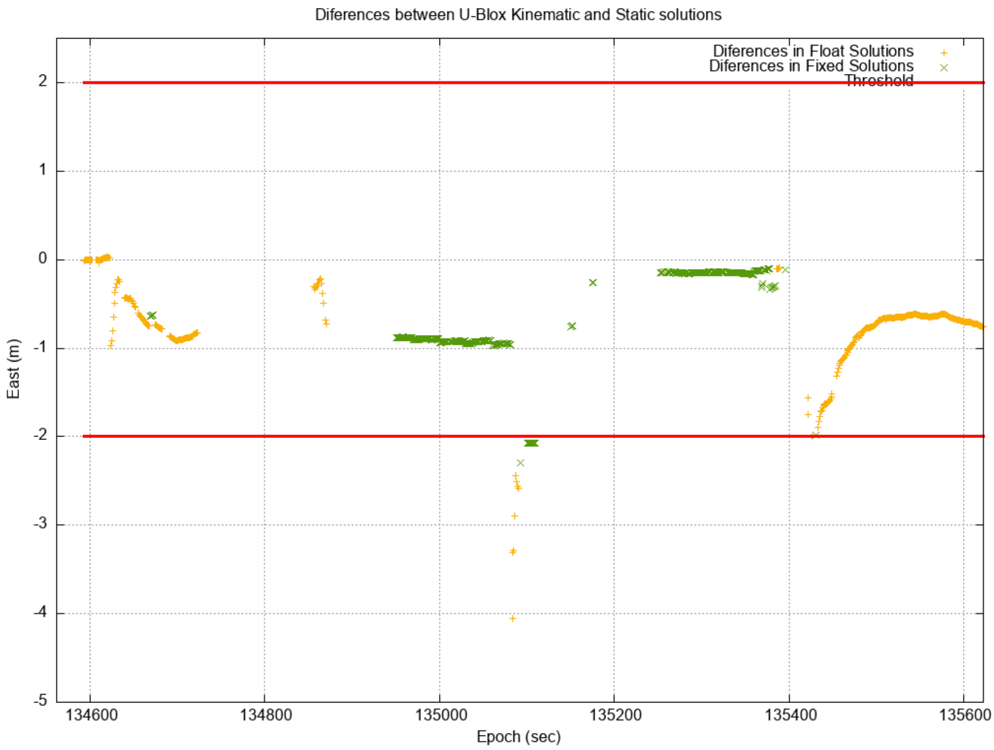


Figure 65 - Simulation 1: East differences between solutions using u-blox

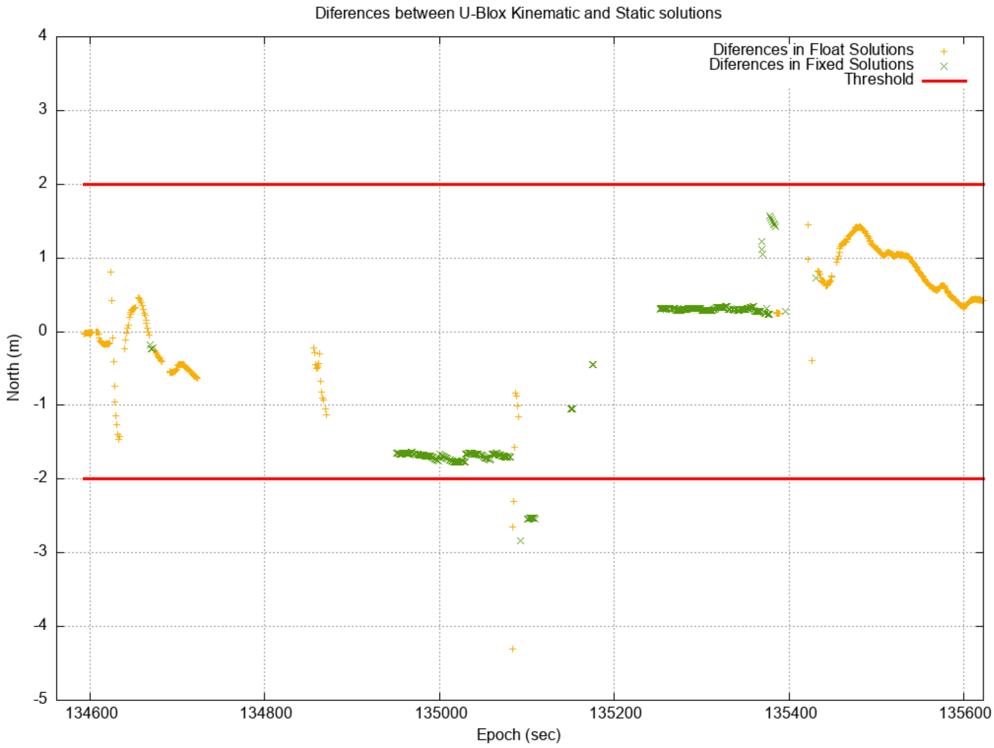


Figure 66 - Simulation 1: North differences between solutions using u-blox

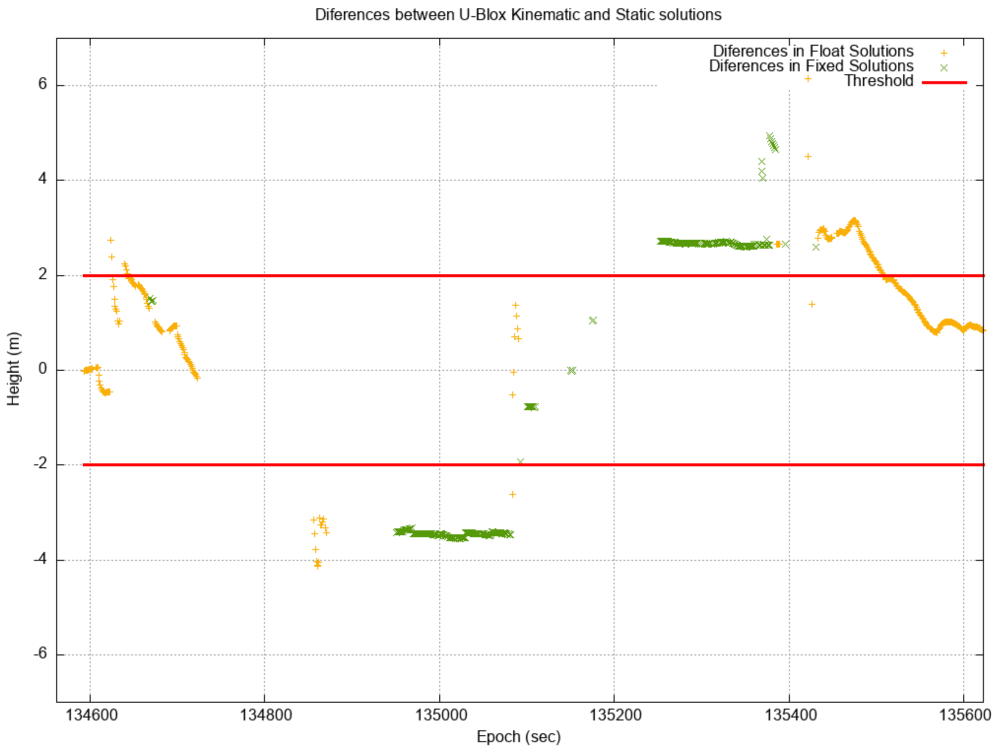


Figure 67 - Simulation 1: Height differences between solutions using u-blox

Leaving aside the height coordinate, u-blox Kinematic solution is mostly inside the threshold. The next table will show if it passes the criterion proposed.

East Percentage	North percentage	Height percentage
97,2 %	97,9 %	42,8 %

Table 3 - Simulation 1: Percentages of third comparison

With this data, the last sentence can be validated. East and North component are sufficiently accurate and can be useful for mechatronic purposes. It can be also reaffirmed the fact that Height coordinate is much more imprecise, and it cannot be assured the reliability of this devices for GNSS positioning.

The other statement that can be reaffirmed is the variability of the kinematic solution, which is changing a lot it´s values as in the last comparison.

9.4. Xiaomi VS u-blox as rovers

The last comparison is between Xiaomi and u-blox used as rovers. It has been seen that those two components are the ones that obtained a better performance, not only in terms of errors but also in stability.

This comparison it is very important to be able to achieve one of the objectives determined at the beginning of this study. The possibility of including GNSS systems in mechatronic applications rely on assuring an error of less than one meter in eighty percent of the epochs.

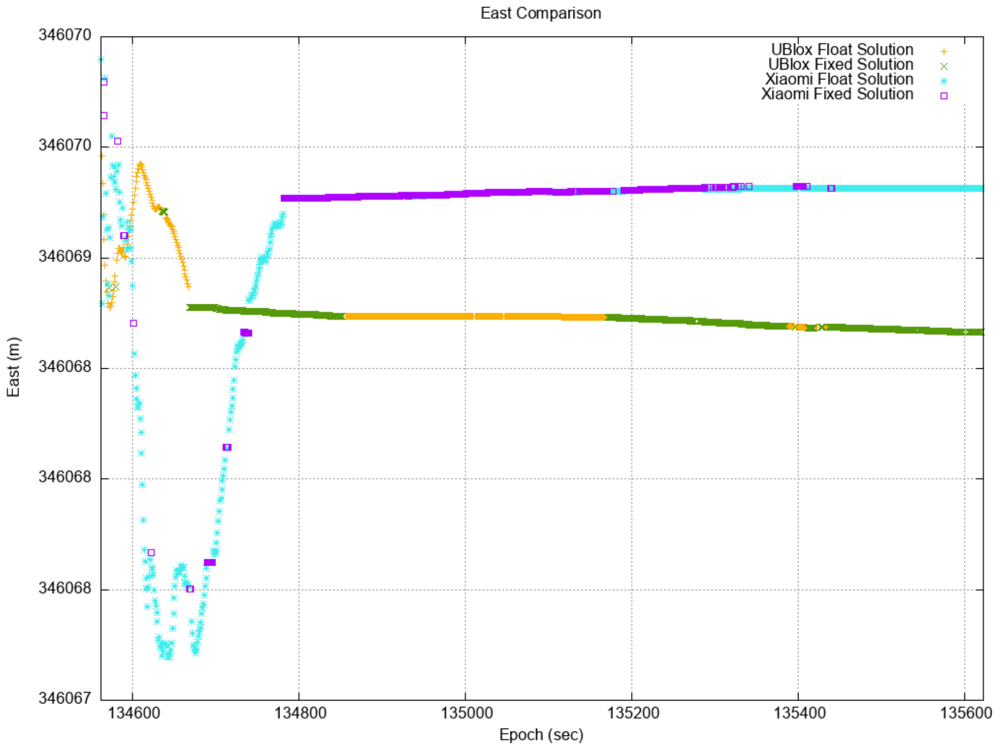


Figure 68 - Simulation 1: East comparison of Xiaomi and u-blox as rovers

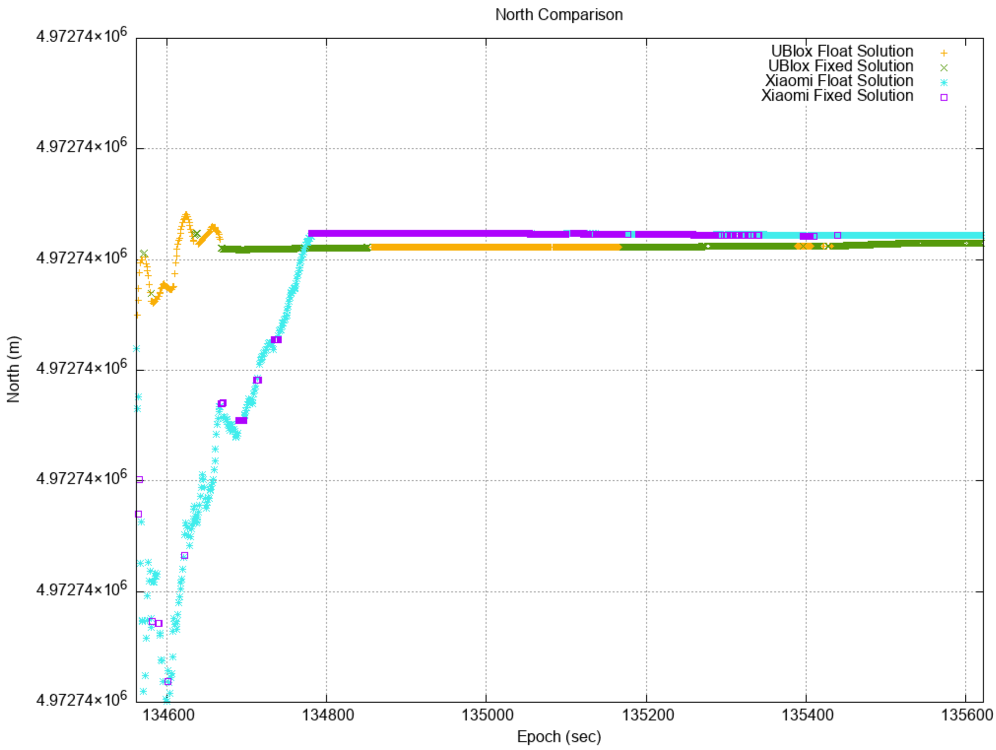


Figure 69 - Simulation 1: North comparison of Xiaomi and u-blox as rovers

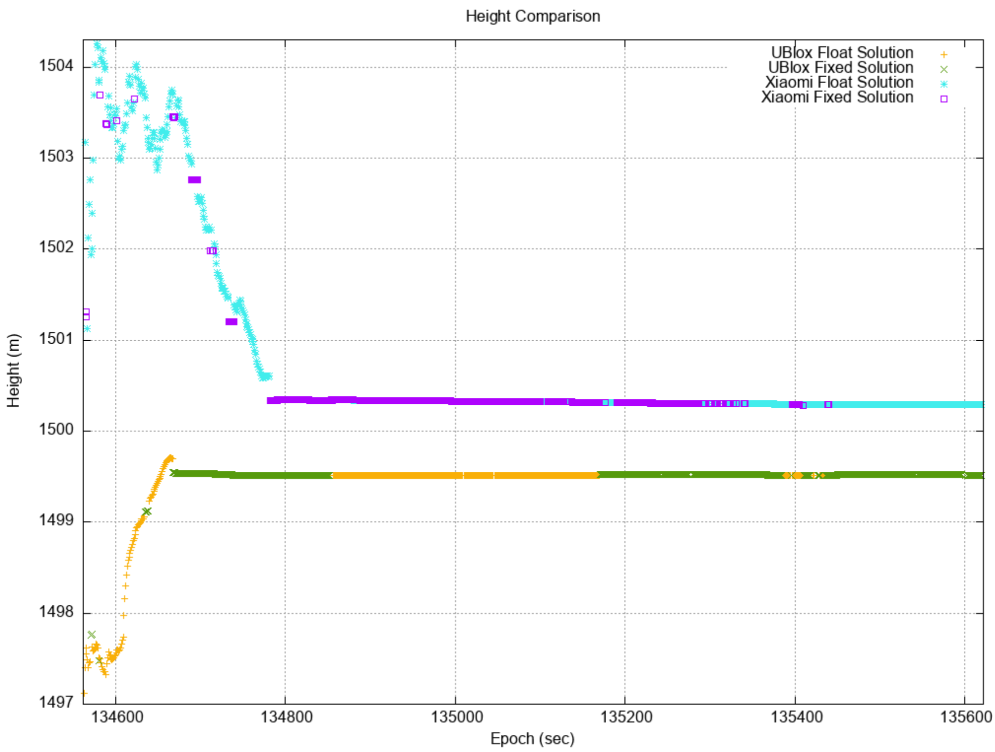


Figure 70 - Simulation 1: Height comparison of Xiaomi and u-blox as rovers

As usually, the software needs some time to stabilize the solution, so the first epochs vary more and seem less accurate than the rest. It is interesting the good behavior that the combination of Xiaomi and u-blox are giving for static solution mode. The accuracy achieved (very similar to the one using u-blox and Torino permanent Station) it is not a cause of the quality of the receivers, as the permanent station is, of course, much more technologically advanced, but probably because the closeness of the two portable devices.

The next graphs will help to see more clearly the differences between the solutions. In this case, a threshold of one meter have been defined. It was already determined the conditions that these data must meet to pass the defined criteria.

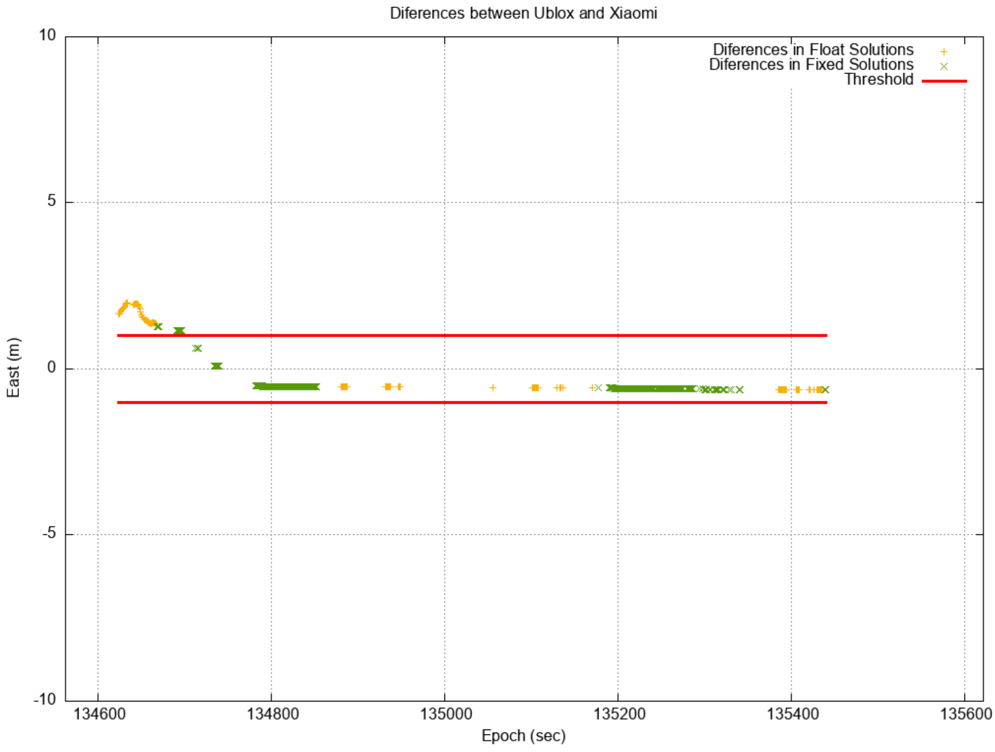


Figure 71 - Simulation 1: East differences between Xiaomi and u-blox as rovers

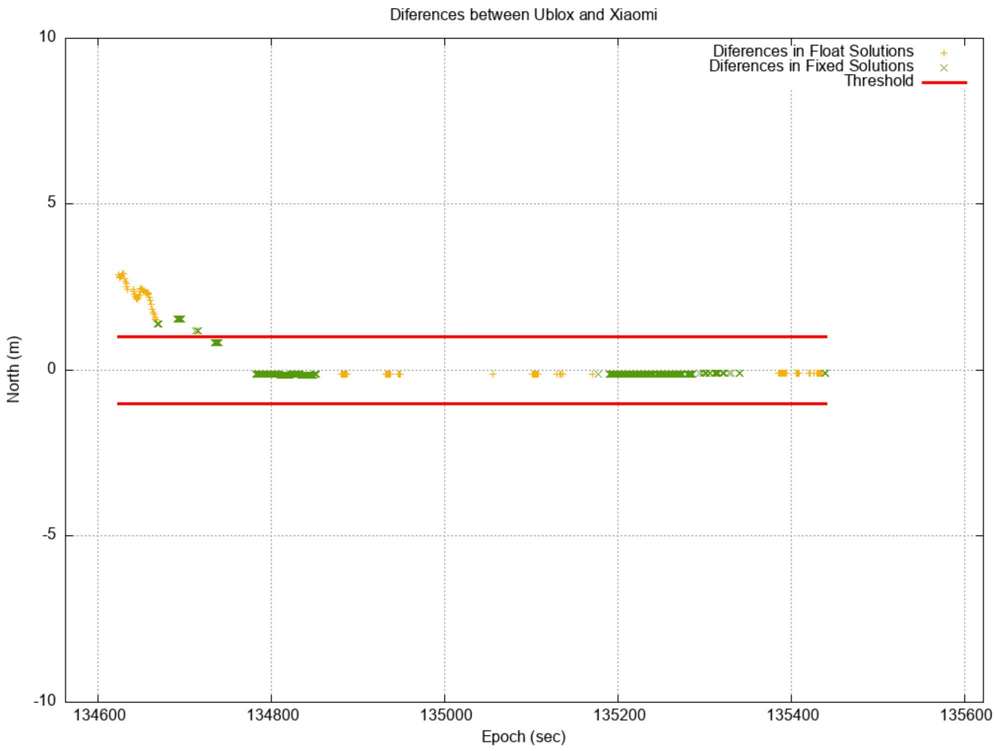


Figure 72 - Simulation 1: North differences between Xiaomi and u-blox as rovers

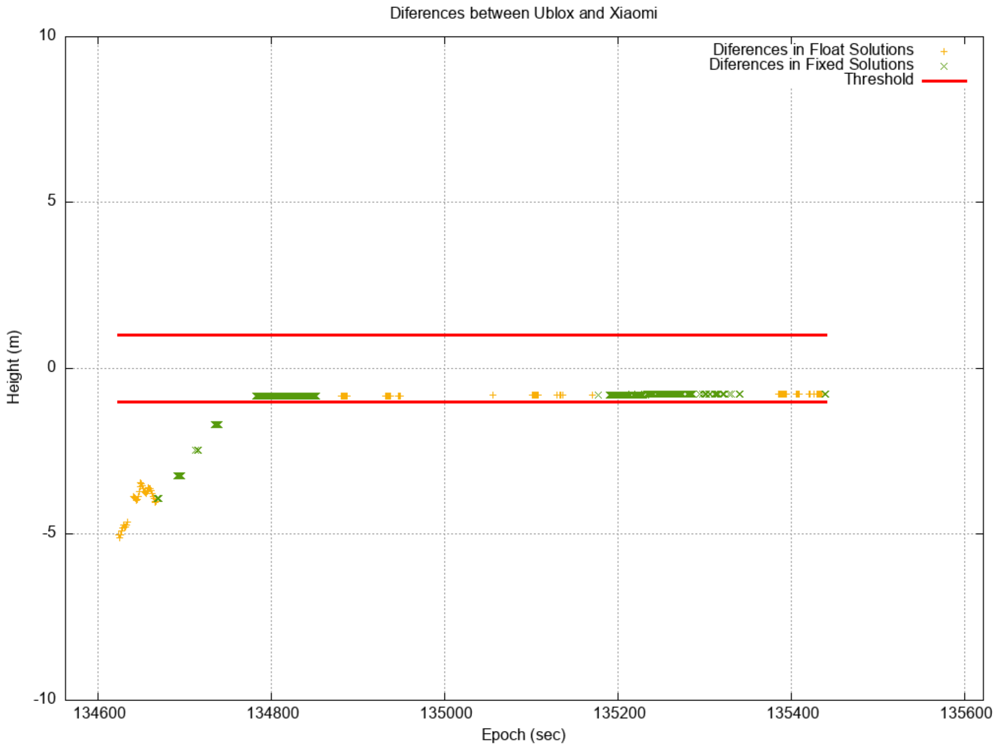


Figure 73 - Simulation 1: Height differences between Xiaomi and u-blox as rovers

East Percentage	North percentage	Height percentage
83,5 %	82,5 %	80,5 %

Table 4 - Simulation 1: Percentages of fourth comparison

If u-blox combination with Torino permanent station is considered as the most accurate solution that can be achieved, it can be assured that regarding the east and north component the Xiaomi phone used as a receiver have a good performance. Being most of the position solution errors less than 1 meter.

It worth to highlight that this percentages can be improved even more if the time of stabilization is not considered. It can be seen in the graphic that only the first epochs do not fill inside the one-meter threshold.

10. Third analysis

The first measurement campaign was deeply analyzed, and a lot of conclusion could be derived from it. The next step, and the objective of this chapter, is testing another set of data from the same receivers to see whether last statements can be reaffirmed.

The idea is the same as before, but this time the RTKLIB plots are not going to be shown. With the data provided by the script, those graphics do not give any further information.

As before, the chapter will be divided into the comparison of the most significant combinations.

10.1. Xiaomi VS Samsung as rovers

The first comparison will rely on both phones as rovers and u-blox providing observation and navigation data, both solutions in static mode. This approach is one of the most interesting ones, as all the devices used to compute a solution are portable.

In the last set of measurements, Samsung receiver was discarded from any mechatronic application, so it is interesting to see if this statement remains.

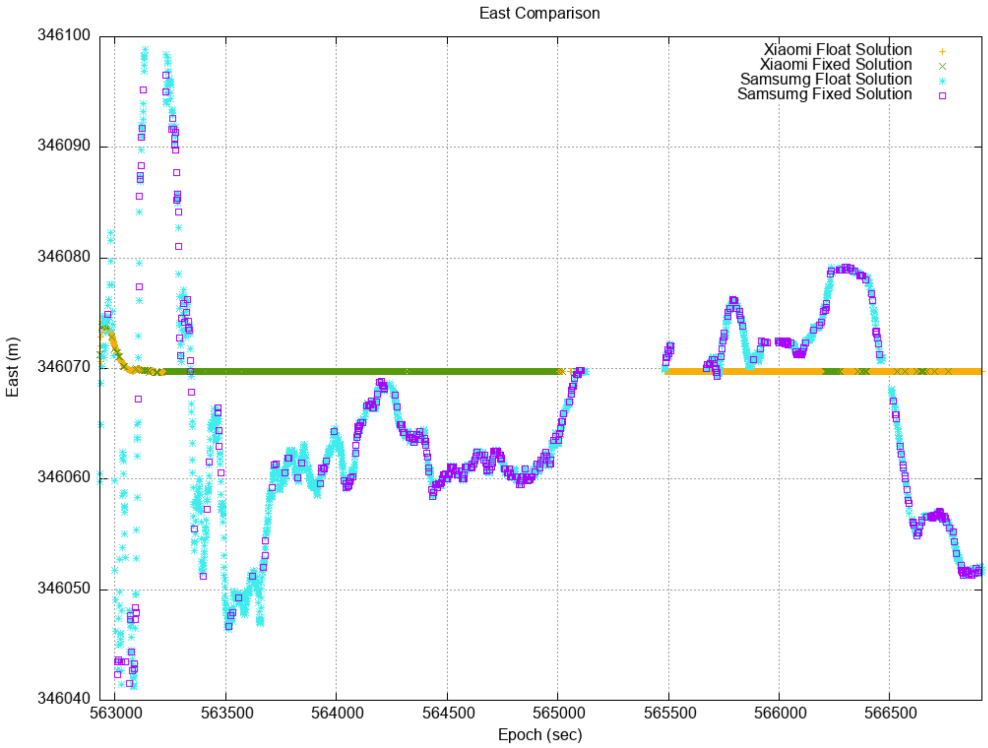


Figure 74 - Simulation 2: East differences between Xiaomi and Samsung as rovers

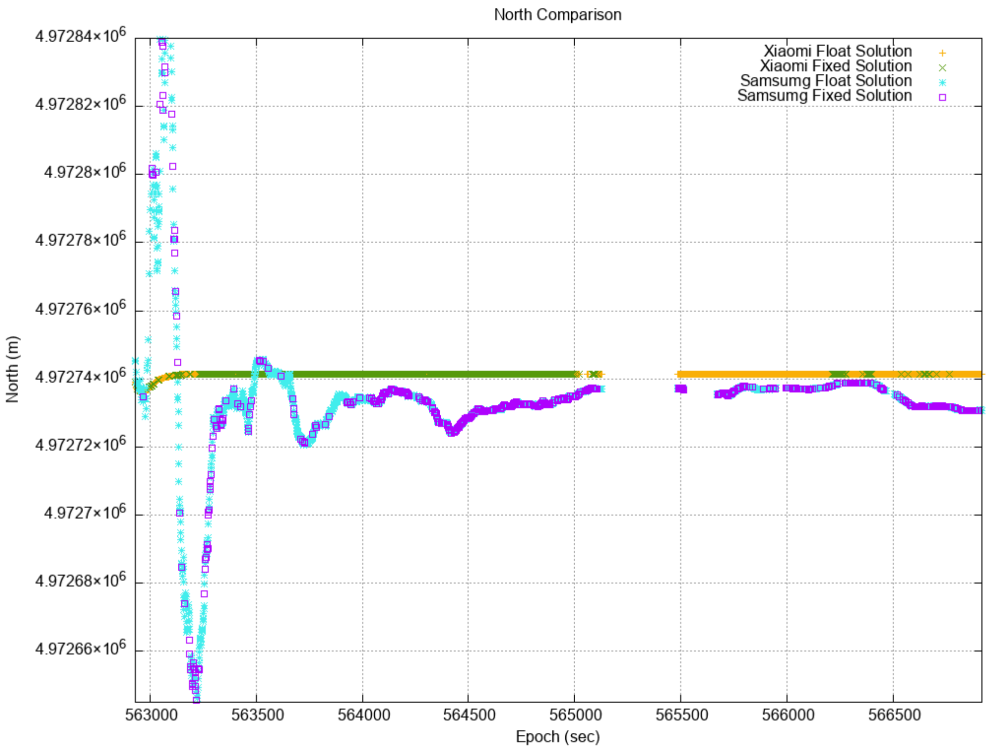


Figure 75 - Simulation 2: North differences between Xiaomi and Samsung as rovers

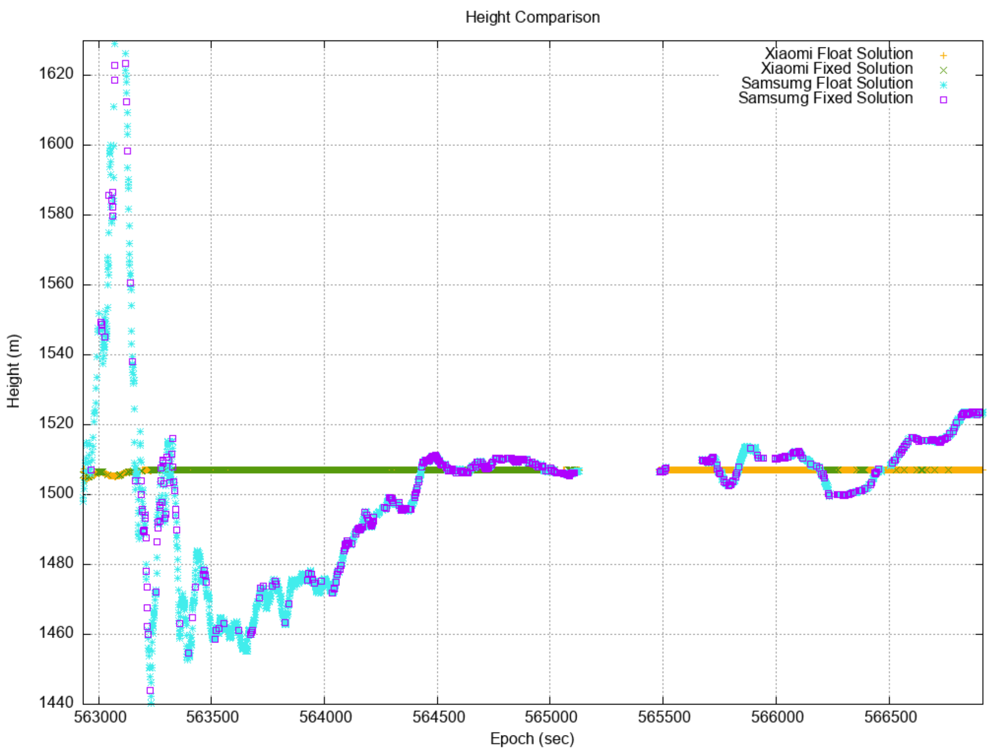


Figure 76 - Simulation 2: Height differences between Xiaomi and Samsung as rovers

Before looking at the real differences of both solutions, the error are clearly above 2 meters. This means that Samsung phone is not providing sufficient quality data again. The threshold will be 10 meters this time, to check whether it is reliable inside that range of error.

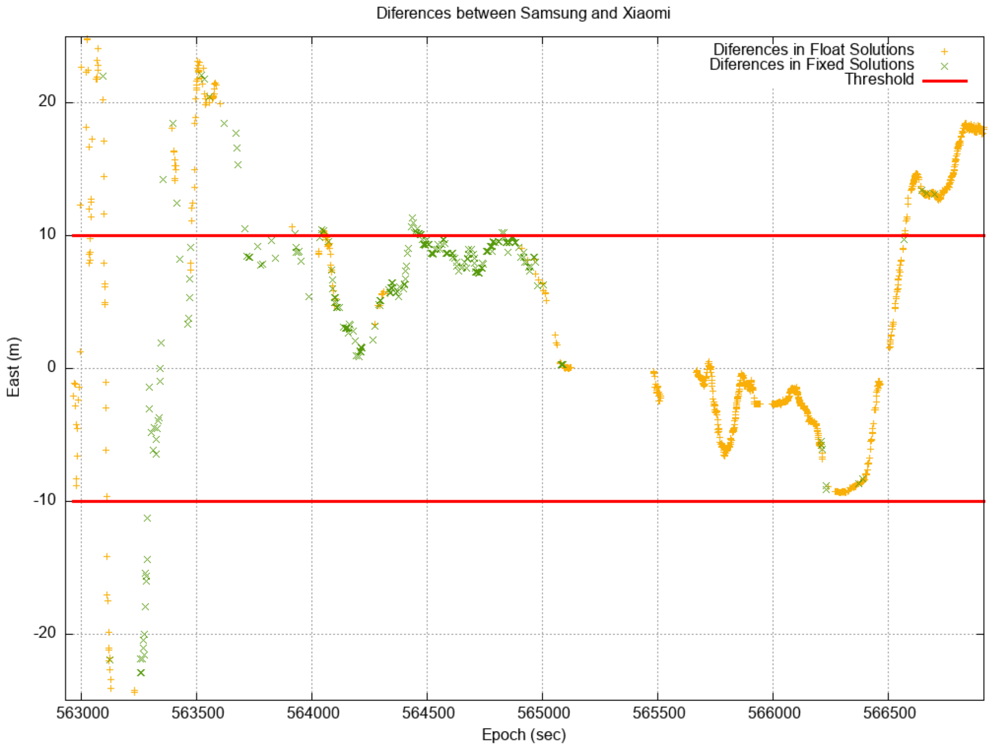


Figure 77 - Simulation 2: East differences between Xiaomi and Samsung as rovers

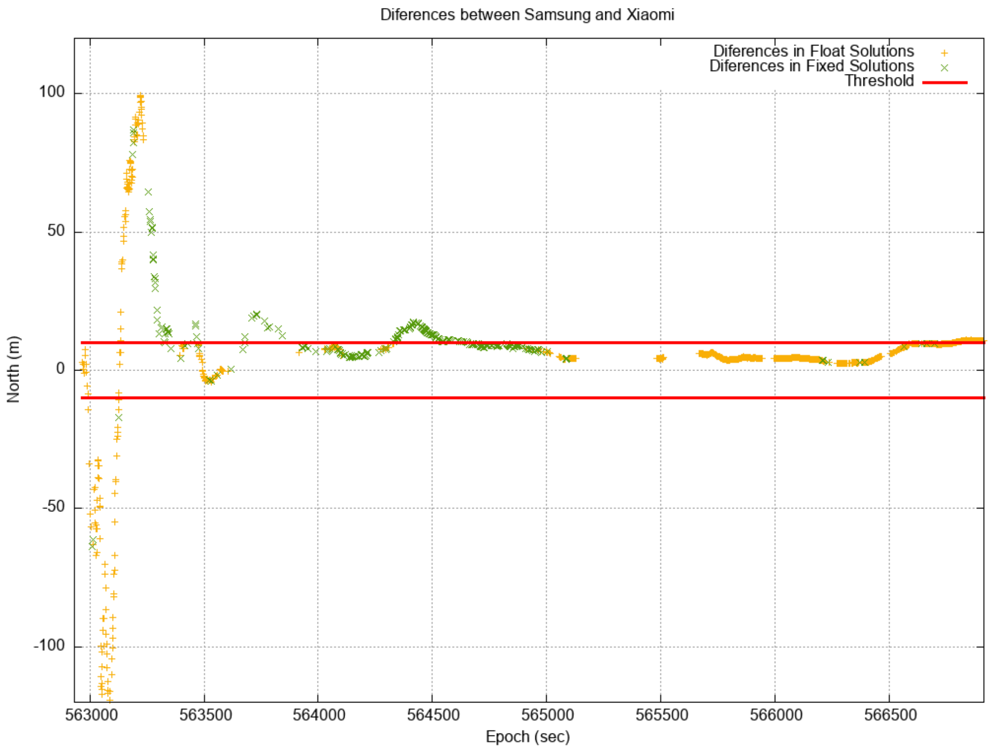


Figure 78 - Simulation 2: North differences between Xiaomi and Samsung as rovers

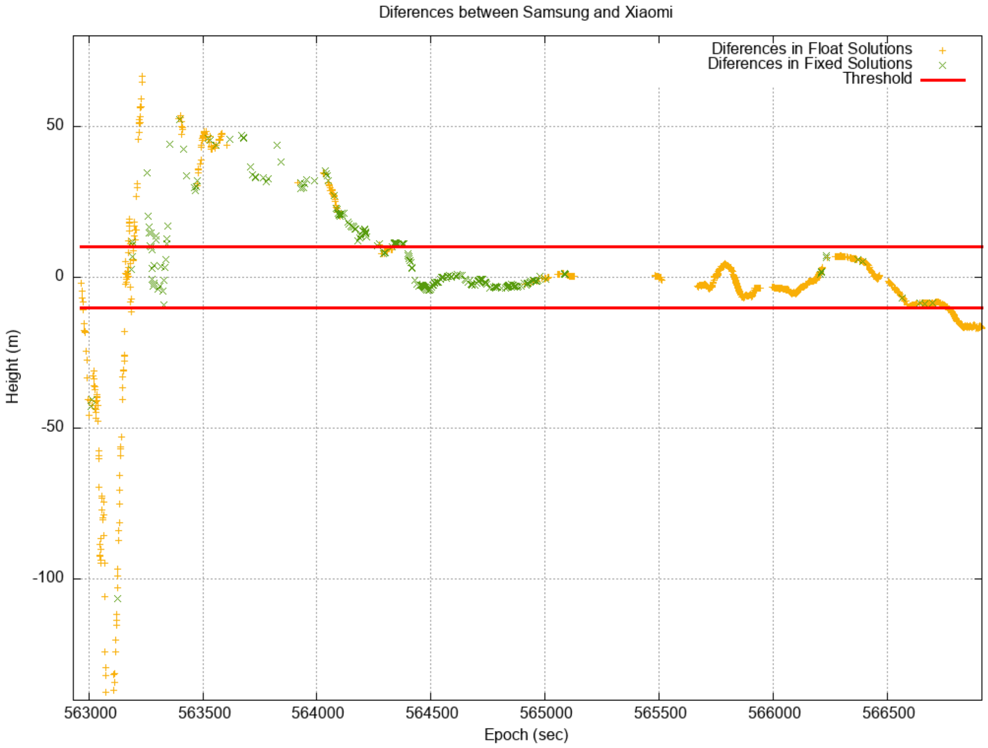


Figure 79 - Simulation 2: Height difference between Xiaomi and Samsung as rovers

East Percentage	North percentage	Height percentage
65,7 %	73,2 %	68,8 %

Table 5 - Simulation 2: Percentages of first comparison

Considering the values of the table, the combination using the Samsung cannot ensure even a 10-meter error. The conclusion derived from the last analysis is now approved and from this point on that phone will not be included in more comparisons. It was demonstrated that it does not have any of the applications of which this study is looking for.

Another interesting aspect that is clearly seen in the graphics, is the lack of data in the middle of the comparison. This fact is not surprising because during the measurement, the u-blox receiver was restarted, and no data could be saved until it was switched on again.

To finish this set of data, the behavior of the Xiaomi is again very stable and, as in the last measurement campaign, its errors are low.

The next step is to prove the idea of static solution mode as being more accurate that kinematic. For that purpose, Xiaomi and u-blox will be compared with themselves in those two modes on the next to analysis.

10.2. Kinematic VS Static with Xiaomi as a rover

This comparison is not only for proving the performance of both solution modes, but also an important way of checking the quality of the Xiaomi performance.

At this point, one of the starting receivers was discarded, so the only portable combination that can pass the criteria and be usable in other applications depends on the Xiaomi and the u-blox.

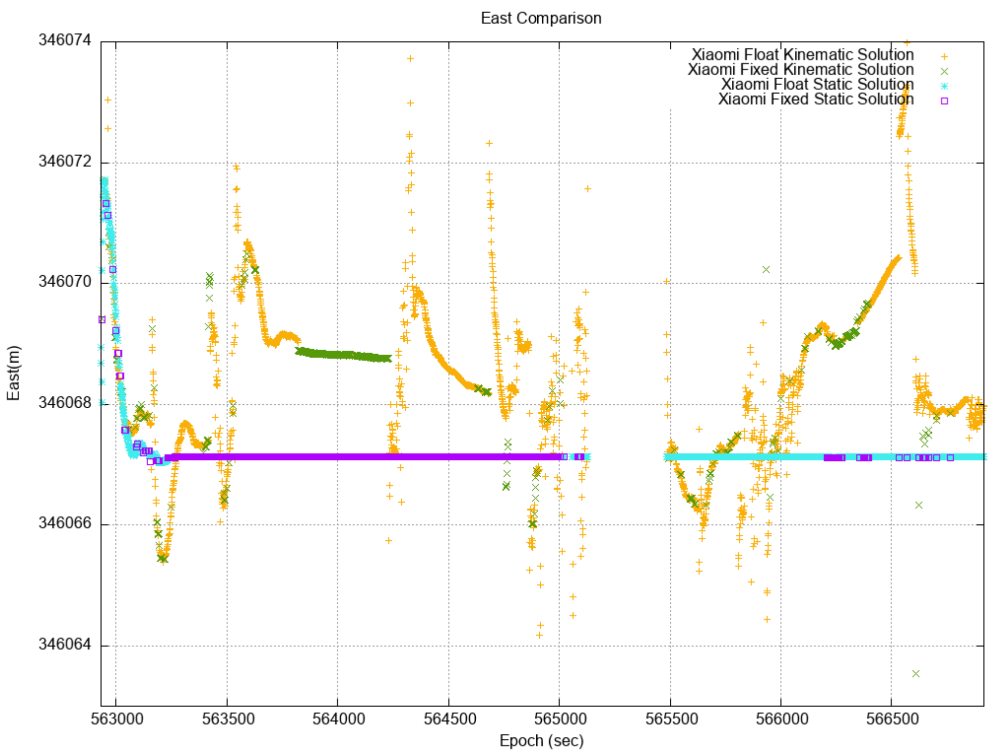


Figure 80 - Simulation 2: East comparison of solutions using Xiaomi

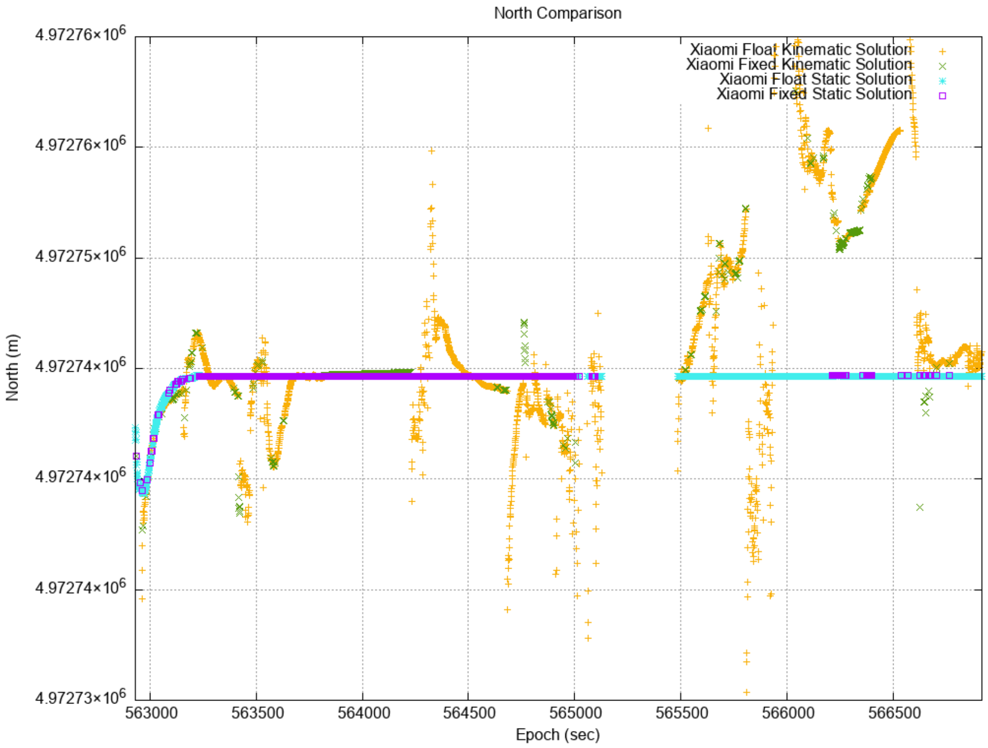


Figure 81 - Simulation 2: North comparison of solutions using Xiaomi

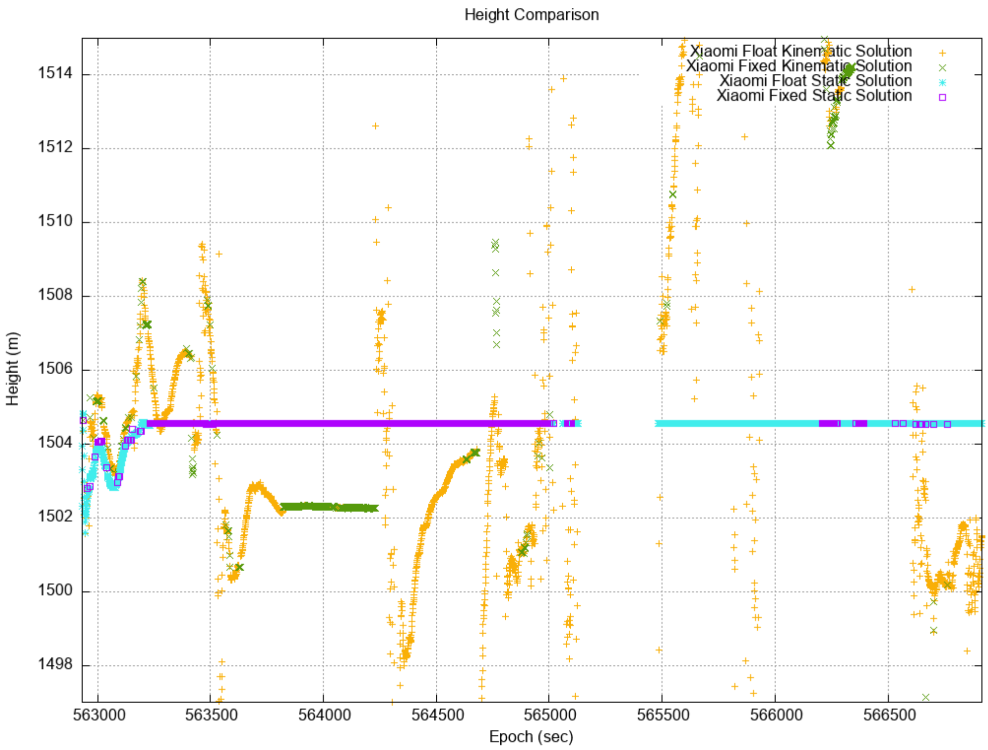


Figure 82 - Simulation 2: Height comparison of solutions using Xiaomi

Once again, the behavior of the Static solution is much better. But in this graphics, Kinematic solution performance get worse with respect the previous study. Let´s see whether it meets the level of precision required.

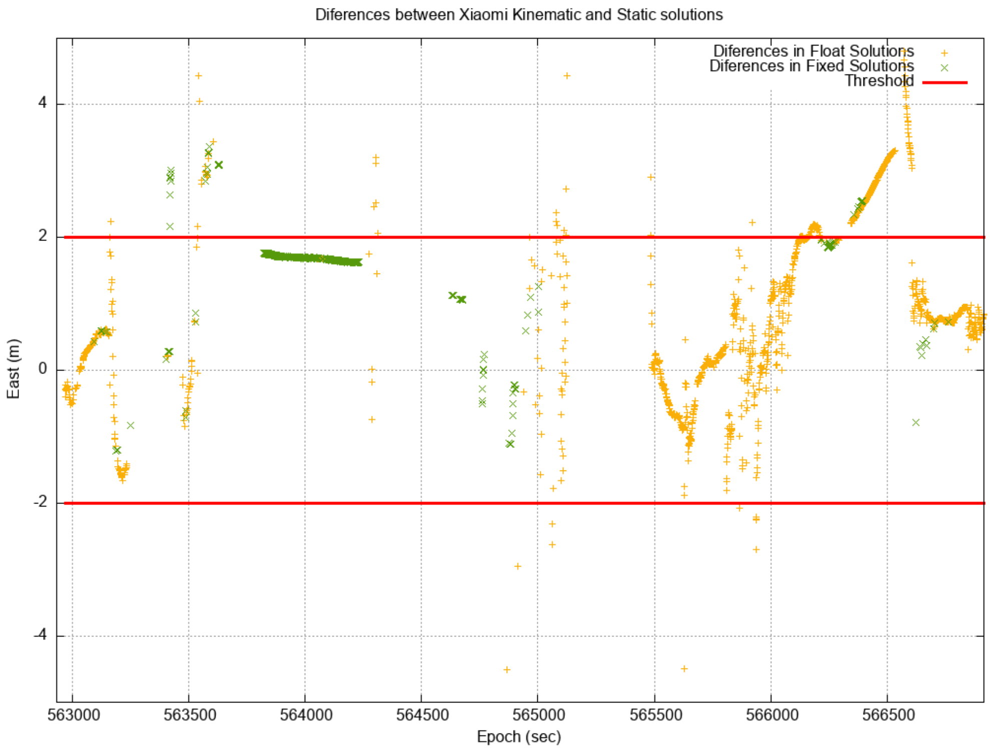


Figure 83 - Simulation 2: East differences between solutions using Xiaomi

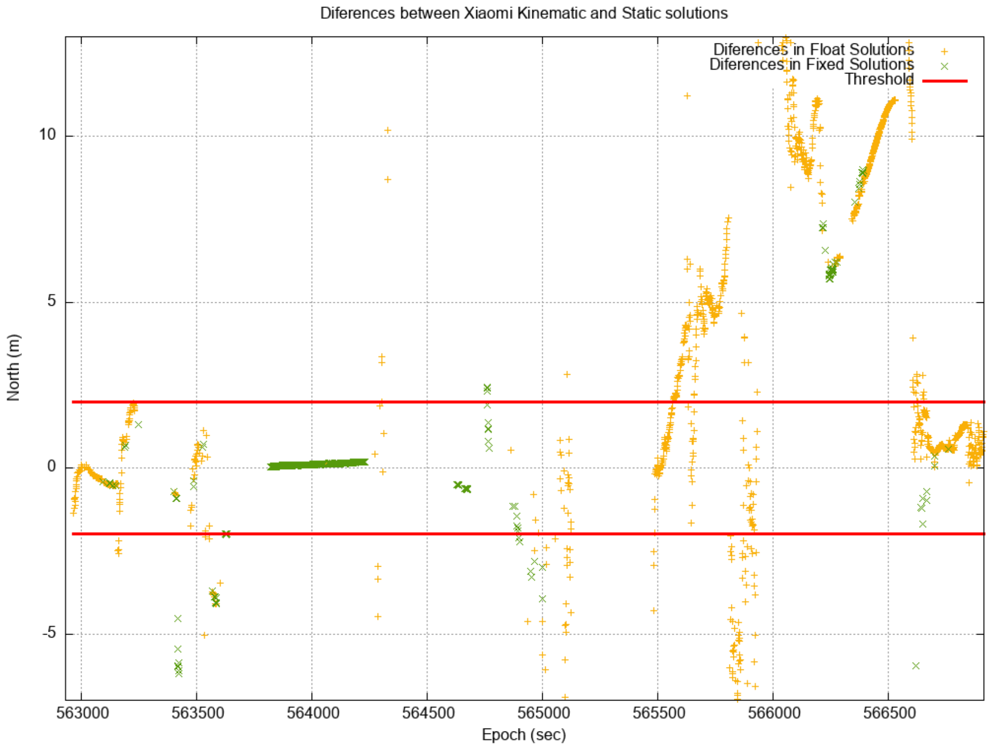


Figure 84 - Simulation 2: North differences between solutions using Xiaomi

Figure 85 - Simulation 2: Height differences between solutions using Xiaomi

East Percentage	North percentage	Height percentage
77,0 %	78,7%	43,7 %

Table 6 - Simulation 2: Percentages of second comparison

Although for east and north coordinates the solution is very close of the 80% percent, in any of them nor in the height one this solution is good enough. This suppose a step back in the statement derived from the other collection, because in that case, even though Kinematic solution was unstable, the errors were inside the defined threshold, what does not happen this time.

10.3. Kinematic VS Static with u-blox as a rover

With four comparisons regarding the solution mode, some patterns can be understood, and the conclusion may already have a certain reliability. Apart of it, is does not have to be highlighted again the importance of the u-blox measurements for the purpose of the thesis.

In the case of this comparison, it does not show any different behavior than the other ones comparing solution modes. That is why the plots are not going to be added as they do not contribute with anything new.

This performance is more like the one saw in the other review, where the range of error is mostly bellow two meters in East and North components. Apart from that, the Static solution is again constant during the hole period.

East Percentage	North percentage	Height percentage
100 %	64,4 %	99,9 %

Table 7 - Simulation 2: Percentages of third comparison

In this combinations, Kinematic performance in north coordinate does not meet the requirement. Due to the high variations and the percentages in some coordinates of some combination solutions Kinematic mode cannot be considered reliable for these kind of measurements in which the receivers are not in motion.

10.4. Xiaomi VS u-blox as rovers

Finally, the last comparison will determine one of the most important conclusions of this research. The possibility of applying the combination of Xiaomi and u-blox together assuring a low error (less than 1 meter) in at least eighty percent of the epochs.

To achieve this purpose, or course, static solution will be used. The value that will be considered as the real position of the user and the one from which the errors will be computed is the arithmetic average between this combination compared with the one using u-blox as a rover and Torino permanent Station as the Master Station and providing navigation data.

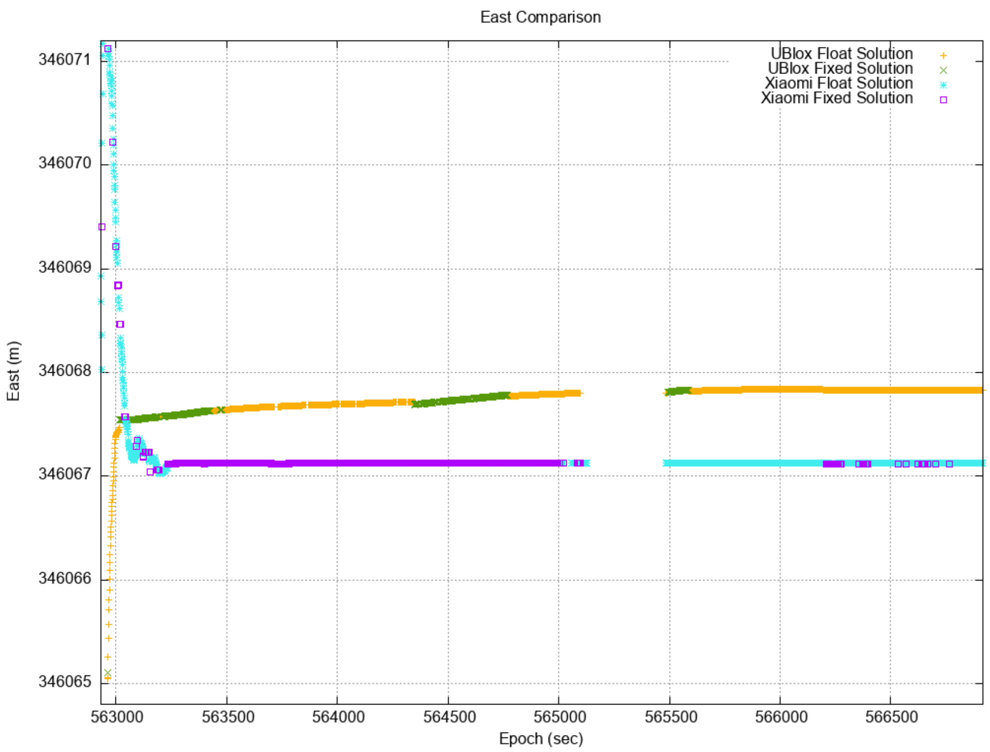


Figure 86 - Simulation 2: East comparison of Xiaomi and u-blox as rovers

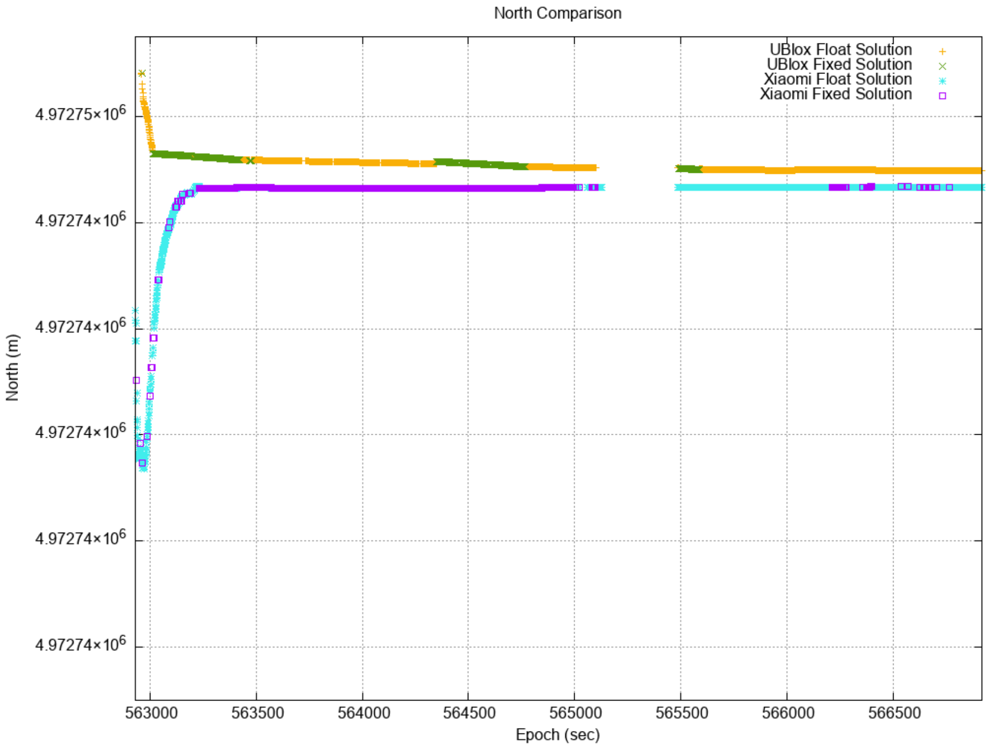


Figure 87 - Simulation 2: North comparison of Xiaomi and u-blox as rovers

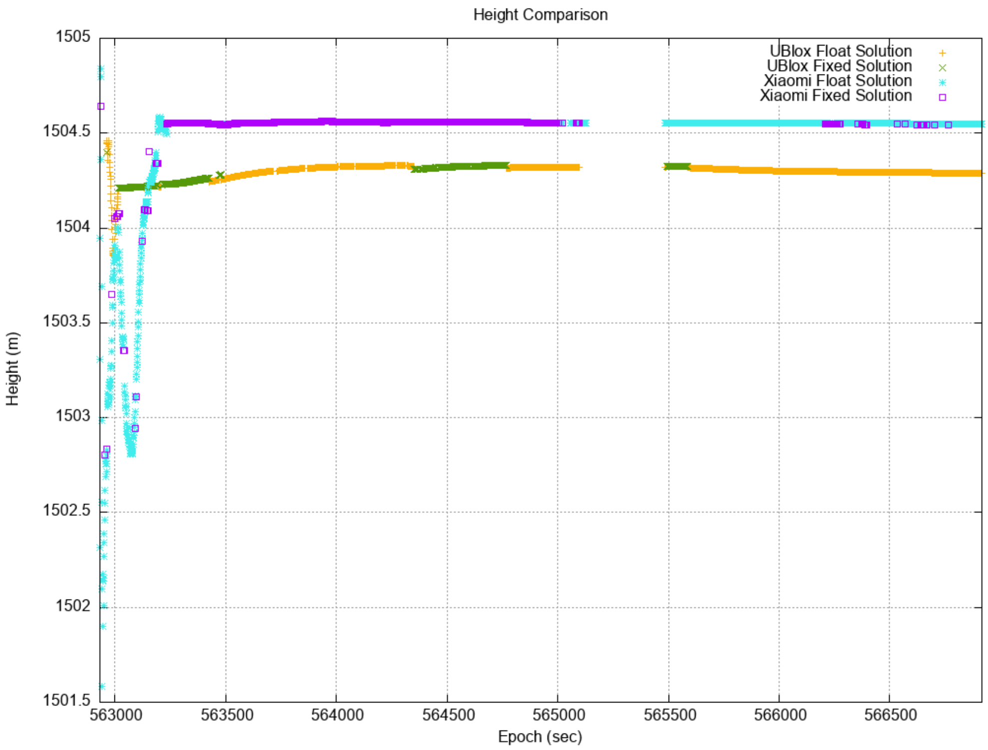


Figure 88 - Simulation 2: Height comparison of Xiaomi and u-blox as rovers

Even though these plots were shown separately, in this last comparison the low variability and the similarity of both solutions stands out even more. With this information it can be affirmed that the errors remain apparently low but is not possible to determine if the data can be considered useful. To do so, this time the Threshold is determined to be of only 1 meter, keeping the percentage of epochs with a lower error in an eighty percent.

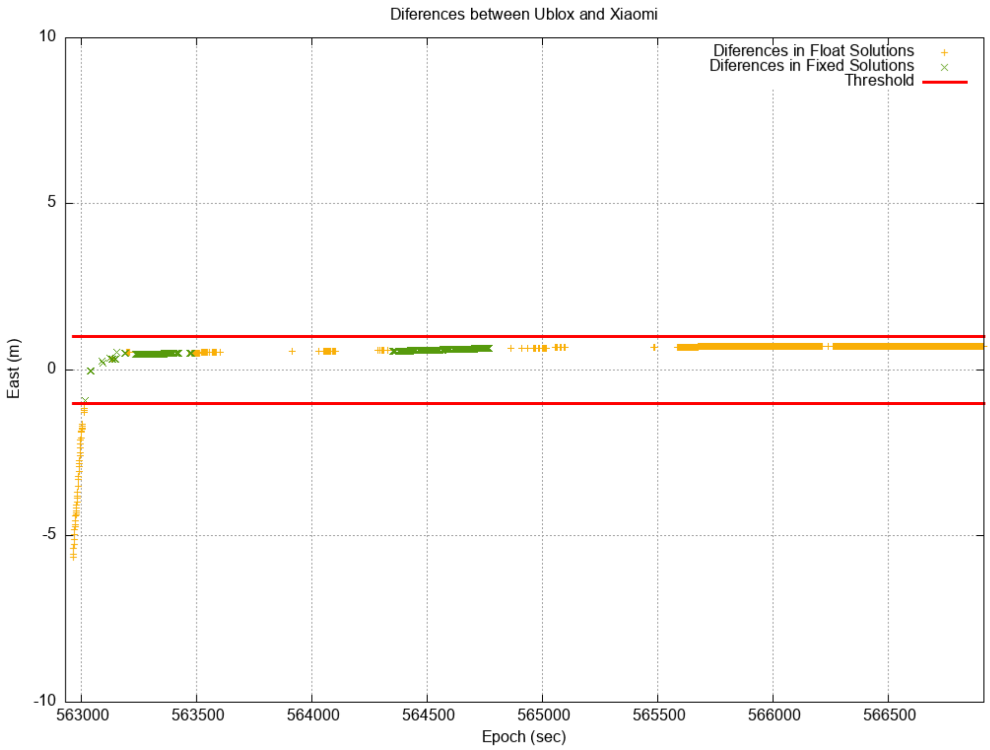


Figure 89 - Simulation 2: East differences between Xiaomi and u-blox as rovers

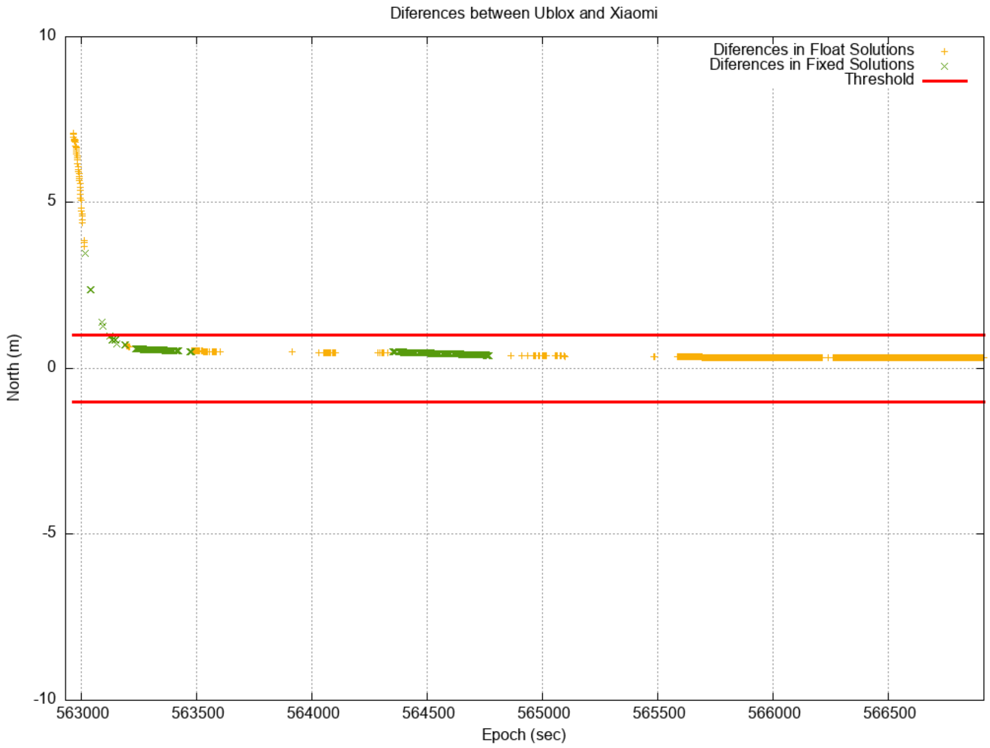


Figure 90 - Simulation 2: North differences between Xiaomi and u-blox as rovers

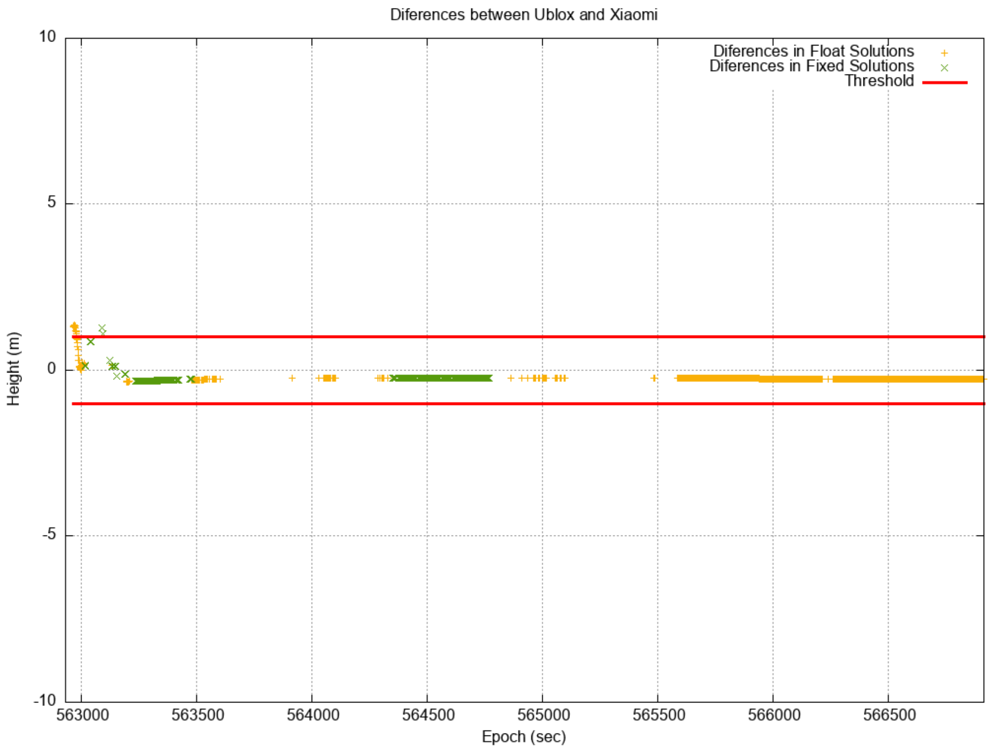


Figure 91 - Simulation 2: Height differences between Xiaomi and u-blox as rovers

East Percentage	North percentage	Height percentage
99,5 %	99,3 %	99,8 %

Table 8 - Simulation 2: Percentages of fourth comparison

This last comparison is with no doubt the one with less range of error. The solutions meet the criteria for each coordinate, what permits assuring that the error will be less than one meter in at least eighty percent of the epochs. In this case the percentages are much higher than that number and without considering the first epochs in which the solution is reaching its stabilization, all errors are below 1 meter.

After the deep study of the performance of the receivers, the next procedure is to validate the conclusions provided during the review of the data generated.

11. Validity of the results

This chapter is about collecting all the inferences subtracted from the previous comparisons. As a summary, a table collecting all the results achieved is shown:

Combinations	Threshold	East Percentage	North percentage	Height percentage
Xiaomi VS Samsung as rovers	20 m	18,6 %	16,2 %	1.6 %
Kinematic VS Static with Xiaomi as rover	2 m	84,2 %	88,9 %	41,4 %
Kinematic VS Static with u-blox as rover	2 m	97,2 %	97,9 %	42,8 %
Xiaomi VS u-blox as rovers with static solution mode	1 m	83,5 %	82,5 %	80,5 %
Xiaomi VS Samsung as rovers	10 m	65,7 %	73,2 %	68,8 %
Kinematic VS Static with Xiaomi as rover	2 m	77,0 %	78,7%	43,7 %
Kinematic VS Static with u-blox as rover	2 m	100 %	64,4 %	99,9 %
Xiaomi VS u-blox as rovers with static solution mode	1 m	99,5 %	99,3 %	99,8 %

Table 9 - Summary of the computed results

The first a probably one of the most unexpected ones, is the uselessness of the measurements done with the Samsung phone. Even though it could be suspected when looking at RTKLIB plots of chapter 2.2.2., with the further analysis the error could be measured and showed clearly that it cannot be assured even a 20 meter of precision in a reasonable percentage of epochs. With this order of error, it is impossible to do a consistent positioning system applicable to any mechatronic purpose.

The second important conclusion derived from the analysis, is the better performance that the Static Solution have with respect the Kinematic one in this kind of measurements in which the receivers are not in motion. The most important difference is the higher stability that the Static solution have, it only varies at first epochs and then maintain almost the same solution over the whole period.

About the kinematic solution, although it has been seen that it does not have big errors in the result, its most negative point is indeed this high variability. Thinking again in the main goal of this test, this issue supposes a barrier for most of mechatronic systems, in which a loss of control cannot be permitted.

Another statement that can be taken is the difficulty that GNSS systems have with height coordinates in some of the solutions. But even thought this issue appeared in some of the results, in the case of static solutions done with the best performance receivers, this problem disappeared.

So, to finish, the last and probably the most important conclusion that the script clarified was the validity of the results when using the measurements from the Xiaomi MI5 phone and the u-blox M8T receiver.

One of the most stable solutions that could be obtained was the one using those two devices. Although Torino permanent station is a receiver much more technologically advanced, the proximity of the u-blox with respect the phone at the time of measuring has a big impact on the quality of the result obtained.

But apart from the time when they were used together, their behavior of their collected data with other receivers was always satisfactory, what verifies again their efficiency.

12. Conclusions

To start the conclusion of the thesis, it is interesting to remind the main goal of this study. As said in the chapter 2, the principal objective was to study the behavior of Global Navigation Satellite Systems in different environments and prove the validity of this technology in mechatronic systems.

To support this possible application, devices commonly used by the majority of citizens (such as cell phones) and others that are easy to acquire (like u-blox) were utilized. The idea of this method, which is called citizen sensing, is to give the population the possibility to use the data they collect. In this way, by taking advantage of commonly used equipment, it is possible to facilitate the inclusion of GNSS technology using these devices in mechatronics-related applications.

After all the computations done during the research it was demonstrated that an error of less than a meter can be assured with these tools in at least eighty percent of epochs. Even though this result could be easily improved if the stabilization period is not considered, it is still more than acceptable.

With this range of error, GNSS systems can be introduced in a lot of mechatronic systems. This is the case of drones, in which apart from the autonomous landing, an accuracy of less than a meter is sufficient to control the device during the flight. The same happens with agriculture robots, which can be controlled by GNSS technology during their use. Of course, there are many more examples where this technology could fit, but it has not been deemed necessary to show them all.

In the case of autonomous vehicles, although GNSS is the core technology for providing location, due to the severe requirements to assure safety of the passengers, it would be necessary to complement it with other sensors.

To summarize, it can be said that the analyzed GNSS devices and algorithms can be combined with mechatronics systems in open sky environments, if they do not require a centimeter-level precision and if they are not critical for safety. Moreover it is possible to conclude that the main objective of this research has been fulfilled.

14. Bibliography

[1] Agency, E. G. (n.d.). *Outcome of the European GNSS' User Consultation Platform.*

[2] Blewitt, G. (n.d.). *Basics of the GPS Technique: Observation Equations.* Department of Geomatics, University of Newcastle.

[3] European Space Agency. (n.d.). *Navipedia*. Retrieved from https://gssc.esa.int/navipedia/index.php/Main_Page

[4] European Space Agency. (n.d.). *The European Space Agency*. Retrieved from http://www.esa.int

[5] EUSPA. (n.d.). *Central role for robust GNSS in autonomous driving.*

[6] Gebre-Egziabher, D. (2018). *Evaluation of Low-Cost, Centimeter-Level Accuracy OEM GNSS Receivers.*

[7] GPS Joint program Office. (2004). *IS-GPS-200.* El Segundo, CA.

[8] Gurtner, W. (2009). *The Receiver Independent Exchange Format.* Bern.

[9] Instituto Geográfico Nacional. (n.d.). *Red Geodésica Nacional de Estaciones de Referencia GNSS*. Retrieved from https://www.ign.es/web/ign/portal/gds-gnss-estaciones-permanentes

[10] J.Sanz Subirana, J. J. (2013). *GNSS Data processing.* Contactivity bv, Leiden, the Netherlands.

[11] London Economics. (2017). *Economic impact to the UK of a disruption.*

[12] National Aeronautics and Space Administration. (n.d.). *Global Positioning system | NASA*. Retrieved from https://www.nasa.gov/

[13] *Novatel*. (2021, 07 01). Retrieved from https://novatel.com/an-introduction-to-gnss/chapter-1-gnss-overview/section-

1#:~:text=The%20space%20segment%20consists%20of,as%20illustrated%20in%20Figure%203

[14] Research group of Astronomy and GEomatics (gAGE). (n.d.). *gAGE: Research group of Astronomy and Geomatics*. Retrieved from https://gage.upc.edu/

[15] The EUREF Permanent Network (EPN). (n.d.). *EUREF Permanent GNSS Network*. Retrieved from https://www.epncb.oma.be/

[16] The Interstate Aviation Committee. (n.d.). *Information and analysis center for positioning, navigation and timing*. Retrieved from https://www.glonass-iac.ru/en/

[17] U-blox. (n.d.). *U-blox official web page*. Retrieved from https://www.u-blox.com/

[18] US Army Corps of Engineers. (2003). *NAVSTAR Global Positioning System Surveying.* Washington DC.

[19] Web Android developers. (n.d.). Retrieved from https://developer.android.com/

[20] Wim De Wilde, G. C.-M. (n.d.). *GNSS Interference in Unmanned Aerial Systems.*

A. Annex

A.1. Programing code

```
cls
@echo off
COLOR F0
::-----------------------------------------------------------------------------------
::-----------------------------------------------------------------------------------
::------------- Script To compare the solutions of u-blox and both phones-------------
::------------------------Author: Carlos Saro De Aldecoa-----------------------------
::------------------------------Number: S275813--------------------------------------
::-----------------------------------------------------------------------------------
::-----------------------------------------------------------------------------------
call :set_environment_paths
::-----------------------------------------------------------------------------------
set CURRENT=%~dp0
set INPUT_TORI_TORI=%CURRENT%MasterTori_NavTori
set OUTPUT_TORI_TORI=%INPUT_TORI_TORI%\Comparison_Results
set INPUT_U-BLOX_U-BLOX=%CURRENT%Masteru-blox_Navu-blox
set OUTPUT_U-BLOX_U-BLOX=%INPUT_U-BLOX_U-BLOX%\Comparison_Results
set INPUT_U-BLOX_STATIC_VS_KINEMATIC=%CURRENT%StaticVSKinematic_u-blox_Tori
set OUTPUT_U-BLOX_STATIC_VS_KINEMATIC=%INPUT_U-BLOX_STATIC_VS_KINEMATIC%\Comparison_Results
set INPUT_XIAOMI_STATIC_VS_KINEMATIC=%CURRENT%StaticVSKinematic_Xiaomi_u-blox
set OUTPUT_XIAOMI_STATIC_VS_KINEMATIC=%INPUT_XIAOMI_STATIC_VS_KINEMATIC%\Comparison_Results

@echo -----------------------------------------------------------------------------------
@echo -----------------------------------------------------------------------------------
@echo -------------------------------START OF THE COMPUTATION----------------------------
@echo -----------------ROVER: Xiaomi and u-blox MASTER: Tori and U-BLOX-------------
@echo -----------------------------------------------------------------------------------
@echo -----------------------------------------------------------------------------------

@echo First it is necesary to compute RTKLIB post procesing with the input parameters desired.^
 The outputs have to be saved in the path: %INPUT_TORI_TORI%
@echo .
@echo .
@echo Press intro when the RTKLIB outputs are saved in the path %INPUT_TORI_TORI%
```

```
@echo .
@echo .
@echo The comparison starts by a change of format of the files to be able to process them:
@echo .
dos2unix %INPUT_TORI_TORI%/u-blox_Tori_Tori.pos && dos2unix
%INPUT_TORI_TORI%/Xiaomi_u-blox_u-blox.pos
@echo .
:: Change the epoch value into a comparable value
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}'^
 %INPUT_TORI_TORI%/u-blox_Tori_Tori.pos > %OUTPUT_TORI_TORI%\u-
blox_Tori_Tori.tmp
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}'^
 %INPUT_TORI_TORI%/Xiaomi_u-blox_u-blox.pos >
%OUTPUT_TORI_TORI%\Xiaomi_u-blox_u-blox.tmp

@echo In order to compare in a better way the results, the geographic coordinates will^
 be proyected into East North and Up (ENU).
@echo .
call :LatLonToENU %OUTPUT_TORI_TORI%\u-blox_Tori_Tori.tmp
%OUTPUT_TORI_TORI%\ENU_u-blox_Tori_Tori.tmp
call :LatLonToENU %OUTPUT_TORI_TORI%\Xiaomi_u-blox_u-blox.tmp
%OUTPUT_TORI_TORI%\ENU_Xiaomi_u-blox_u-blox.tmp

@echo .
@echo To be able to compare proterly these files, each row of them have to refer to the
same epoch.
@echo .
call :OrganizeFiles %OUTPUT_TORI_TORI%\ENU_u-blox_Tori_Tori.tmp
%OUTPUT_TORI_TORI%\u-blox_Tori_Tori_Tmp.tmp^
 134562
call :OrganizeFiles %OUTPUT_TORI_TORI%\ENU_Xiaomi_u-blox_u-blox.tmp
%OUTPUT_TORI_TORI%\Xiaomi_u-blox_u-blox_Tmp.tmp^
 134562

@echo Once this is done, we can proceed and compare both files:
@echo .
@echo .
paste -d " " %OUTPUT_TORI_TORI%\u-blox_Tori_Tori_Tmp.tmp
%OUTPUT_TORI_TORI%\Xiaomi_u-blox_u-blox_Tmp.tmp >^
 %OUTPUT_TORI_TORI%\CompareEpochs.dat

call :plotDifColors %OUTPUT_TORI_TORI%\CompareEpochs.dat
%OUTPUT_TORI_TORI%\EAST_u-bloxVSXiaomi.png "East Comparison" "East(m)"^
 2 346067 346070 134562 135622
@echo .
@echo .
call :plotDifColors %OUTPUT_TORI_TORI%\CompareEpochs.dat
%OUTPUT_TORI_TORI%\NORTH_u-bloxVSXiaomi.png "North Comparison" "North"^
```

```
 3 4972738 4972744 134562 135622
@echo .
@echo .
call :plotDifColors %OUTPUT_TORI_TORI%\CompareEpochs.dat
%OUTPUT_TORI_TORI%\HEI_u-bloxVSXiaomi.png "Height Comparison" "Height"^
 4 1497 1504.3 134562 135622
@echo .
@echo .
call :computeDiferences %OUTPUT_TORI_TORI%\CompareEpochs.dat
%OUTPUT_TORI_TORI%\SolutionDiferences.dat 1 -1
call :computePercentages %OUTPUT_TORI_TORI%\SolutionDiferences.dat
%OUTPUT_TORI_TORI%\Percentages.dat

call :plotDiferences %OUTPUT_TORI_TORI%\SolutionDiferences.dat
%OUTPUT_TORI_TORI%\DIF_EAST_u-bloxVSXiaomi.png^
 "Diferences between u-blox and Xiaomi" "East" 2 -10 10 134562 135622
@echo .
@echo .
call :plotDiferences %OUTPUT_TORI_TORI%\SolutionDiferences.dat
%OUTPUT_TORI_TORI%\DIF_NORTH_u-bloxVSXiaomi.png^
 "Diferences between u-blox and Xiaomi" "North" 3 -10 10 134562 135622
@echo .
@echo .
call :plotDiferences %OUTPUT_TORI_TORI%\SolutionDiferences.dat
%OUTPUT_TORI_TORI%\DIF_HEIGHT_u-bloxVSXiaomi.png^
 "Diferences between u-blox and Xiaomi" "Height" 4 -10 10 134562 135622
@echo .
@echo .
del /q %OUTPUT_TORI_TORI%\*.tmp
@echo .
@echo .
@echo The comparison between both phones and the u-blox as rovers is done
@echo The results can be found in the path: %OUTPUT_TORI_TORI%

@echo .
@echo --------------------------------------------------------------------------------
@echo --------------------------------------------------------------------------------
@echo -----------------------ROVER: Xiaomi and Samsung  MASTER: u-blox--------------------
@echo --------------------------------------------------------------------------------
@echo --------------------------------------------------------------------------------
@echo .
@echo .
@echo Before Executing RTKLIB we have to compute the solution fix position mean values
to introduce it as the reference position of the u-blox.
@echo The results will be saved in %INPUT_TORI_TORI%/MeanFixValues.dat and will
have to be introduce in the program manually.

call :meanPosValue %INPUT_TORI_TORI%/u-blox_Tori_Tori.pos
```

```
@echo .
@echo .
@echo Press intro when the RTKLIB outputs are saved in the path %INPUT_U-BLOX_U-BLOX%

@echo .
@echo .
@echo The comparison starts by a change of format of the files to be able to process them:
@echo .
dos2unix %INPUT_U-BLOX_U-BLOX%/Xiaomi_u-blox_u-blox.pos && dos2unix %INPUT_U-BLOX_U-BLOX%/Samsung_u-blox_u-blox.pos


@echo .
@echo .
:: Change the epoch value into a comparable value
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}' %INPUT_U-BLOX_U-BLOX%/Xiaomi_u-blox_u-blox.pos > %OUTPUT_U-BLOX_U-BLOX%\Xiaomi_u-blox_u-blox.tmp
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}' %INPUT_U-BLOX_U-BLOX%/Samsung_u-blox_u-blox.pos > %OUTPUT_U-BLOX_U-BLOX%\Samsung_u-blox_u-blox.tmp

@echo In order to compare in a better way the results, the geographic coordinates will be proyected into East North and Up (ENU).
@echo .
@echo .
call :LatLonToENU %OUTPUT_U-BLOX_U-BLOX%\Xiaomi_u-blox_u-blox.tmp %OUTPUT_U-BLOX_U-BLOX%\ENU_Xiaomi_u-blox_u-blox.tmp
call :LatLonToENU %OUTPUT_U-BLOX_U-BLOX%\Samsung_u-blox_u-blox.tmp %OUTPUT_U-BLOX_U-BLOX%\ENU_Samsung_u-blox_u-blox.tmp

@echo To be able to compare proterly these files, each row of them have to refer to the same epoch.

call :OrganizeFiles %OUTPUT_U-BLOX_U-BLOX%\ENU_Xiaomi_u-blox_u-blox.tmp %OUTPUT_U-BLOX_U-BLOX%\Xiaomi_u-blox_u-blox_Tmp.tmp 134562
call :OrganizeFiles %OUTPUT_U-BLOX_U-BLOX%\ENU_Samsung_u-blox_u-blox.tmp %OUTPUT_U-BLOX_U-BLOX%\Samsung_u-blox_u-blox_Tmp.tmp 134562

@echo Once this is done, we can proceed and compare both files:
@echo .
@echo .
paste -d " " %OUTPUT_U-BLOX_U-BLOX%\Xiaomi_u-blox_u-blox_Tmp.tmp %OUTPUT_U-BLOX_U-BLOX%\Samsung_u-blox_u-blox_Tmp.tmp > %OUTPUT_U-BLOX_U-BLOX%\CompareEpochs.dat

:: Here all de plots are made
```

```
call :plotDifColors2 %OUTPUT_U-BLOX_U-BLOX%\CompareEpochs.dat %OUTPUT_U-BLOX_U-BLOX%\EAST_SamsungVSXiaomi.png "East Comparison" "East" 2 346040 346160 134533 135622
@echo .
@echo .
call :plotDifColors2 %OUTPUT_U-BLOX_U-BLOX%\CompareEpochs.dat %OUTPUT_U-BLOX_U-BLOX%\NORTH_SamsungVSXiaomi.png "North Comparison" "North" 3 4972630 4972820 134533 135622
@echo .
@echo .
call :plotDifColors2 %OUTPUT_U-BLOX_U-BLOX%\CompareEpochs.dat %OUTPUT_U-BLOX_U-BLOX%\HEI_SamsungVSXiaomi.png "Height Comparison" "Height" 4 1100 1600 134533 135622
@echo .
@echo .
call :plotDiferences %OUTPUT_U-BLOX_U-BLOX%\SolutionDiferences.dat %OUTPUT_U-BLOX_U-BLOX%\DIF_EAST_SamsungVSXiaomi.png "Diferences between Samsung and Xiaomi" "East" 2 -25 25 134562 134852
@echo .
@echo .
call :plotDiferences %OUTPUT_U-BLOX_U-BLOX%\SolutionDiferences.dat %OUTPUT_U-BLOX_U-BLOX%\DIF_NORTH_SamsungVSXiaomi.png "Diferences between Samsung and Xiaomi" "North" 3 -100 100 134562 134852
@echo .
@echo .
call :plotDiferences %OUTPUT_U-BLOX_U-BLOX%\SolutionDiferences.dat %OUTPUT_U-BLOX_U-BLOX%\DIF_HEIGHT_SamsungVSXiaomi.png "Diferences between Samsung and Xiaomi" "Height" 4 -100 100 134562 134852
@echo .
@echo .

call :computeDiferences %OUTPUT_U-BLOX_U-BLOX%\CompareEpochs.dat %OUTPUT_U-BLOX_U-BLOX%\SolutionDiferences.dat 20 -20
call :computePercentages %OUTPUT_U-BLOX_U-BLOX%\SolutionDiferences.dat %OUTPUT_U-BLOX_U-BLOX%\Percentages.dat
del /q %OUTPUT_U-BLOX_U-BLOX%\*.tmp

@echo The comparison between both phones and the u-blox as rovers is done
@echo The results can be found in the path: %OUTPUT_U-BLOX_U-BLOX%
@echo .
@echo .

@echo .
@echo ------------------------------------------------------------------------------------
@echo ------------------------------------------------------------------------------------
@echo ----------------------------------------Xiaomi----------------------------------------
@echo -----------------------------STATIC and KINEMATIC Solutions------------------------
@echo ------------------------------------------------------------------------------------
@echo .
```

```
@echo .
@echo Press intro when the RTKLIB outputs are saved in the path %INPUT_XIAOMI_STATIC_VS_KINEMATIC%

@echo The comparison starts by a change of format of the files to be able to process them:
@echo .
dos2unix %INPUT_XIAOMI_STATIC_VS_KINEMATIC%/Xiaomi_u-blox_Kinematic.pos && dos2unix %INPUT_XIAOMI_STATIC_VS_KINEMATIC%/Xiaomi_u-blox_Static.pos
@echo .
:: Change the epoch value into a comparable value

gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}' %INPUT_XIAOMI_STATIC_VS_KINEMATIC%/Xiaomi_u-blox_Kinematic.pos > %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Kinematic.tmp
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}' %INPUT_XIAOMI_STATIC_VS_KINEMATIC%/Xiaomi_u-blox_Static.pos > %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Static.tmp

@echo In order to compare in a better way the results, the geographic coordinates will be proyected into East North and Up (ENU).
@echo .
call :LatLonToENU %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Kinematic.tmp %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\ENU_Xiaomi_u-blox_Kinematic.tmp
call :LatLonToENU %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Static.tmp %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\ENU_Xiaomi_u-blox_Static.tmp

@echo .
@echo To be able to compare proterly these files, each row of them have to refer to the same epoch.
@echo .

call :OrganizeFiles %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\ENU_Xiaomi_u-blox_Kinematic.tmp %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Kinematic_Tmp.tmp 134533
call :OrganizeFiles %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\ENU_Xiaomi_u-blox_Static.tmp %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Static_Tmp.tmp 134533


@echo Once this is done, we can proceed and compare both files:
@echo .
@echo .
paste -d " " %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Kinematic_Tmp.tmp %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Xiaomi_u-blox_Static_Tmp.tmp > %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\CompareEpochs.dat
```

```
:: Here all de plots are made
@echo .
call :plotDifColors3 %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\EAST_StaticVSKinematic.png "East Comparison" "East(m)" 2 346065 346074 134562 135622
@echo .
@echo .
call :plotDifColors3 %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\NORTH_StaticVSKinematic.png "North Comparison" "North" 3 4972737 4972748 134562 135622
@echo .
@echo .
call :plotDifColors3 %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\HEI_StaticVSKinematic.png "Height Comparison" "Height" 4 1494 1510 134562 135622
@echo .


call :computeDiferences %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\SolutionDiferences.dat 2 -2
call :computePercentages %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\Percentages.dat

call :plotDiferences %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\DIF_EAST_StaticVSKinematic.png "Diferences between Xiaomi Kinematic and Static solutions" "East" 2 -5 5 134562 135622
@echo .
@echo .
call :plotDiferences %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\DIF_NORTH_StaticVSKinematic.png "Diferences between Xiaomi Kinematic and Static solutions" "North" 3 -7 13 134562 135622
@echo .
@echo .
call :plotDiferences %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\DIF_HEIGHT_StaticVSKinematic.png "Diferences between Xiaomi Kinematic and Static solutions" "Height" 4 -6 15 134562 135622


del /q %OUTPUT_XIAOMI_STATIC_VS_KINEMATIC%\*.tmp
```

```
@echo .
@echo .

@echo ------------------------------------------------------------------------------
@echo ------------------------------------------------------------------------------
@echo ----------------------------------------u-blox------------------------------------
@echo ----------------------------STATIC and KINEMATIC Solutions------------------------
@echo ------------------------------------------------------------------------------
@echo .
@echo .
@echo Press intro when the RTKLIB outputs are saved in the path %INPUT_U-
BLOX_STATIC_VS_KINEMATIC%

@echo .
@echo .
@echo The comparison starts by a change of format of the files to be able to process them:
@echo .
dos2unix %INPUT_U-BLOX_STATIC_VS_KINEMATIC%/u-blox_Tori_Kinematic.pos &&
dos2unix %INPUT_U-BLOX_STATIC_VS_KINEMATIC%/u-blox_Tori_Static.pos
@echo .
:: Change the epoch value into a comparable value

gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}'
%INPUT_U-BLOX_STATIC_VS_KINEMATIC%/u-blox_Tori_Kinematic.pos >
%OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\u-blox_Tori_Kinematic.tmp
gawk '{if (NR ^> 25) {printf("%%s %%i %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6)};}'
%INPUT_U-BLOX_STATIC_VS_KINEMATIC%/u-blox_Tori_Static.pos > %OUTPUT_U-
BLOX_STATIC_VS_KINEMATIC%\u-blox_Tori_Static.tmp

@echo In order to compare in a better way the results, the geographic coordinates will be
proyected into East North and Up (ENU).
@echo .
call :LatLonToENU %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\u-
blox_Tori_Kinematic.tmp %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\ENU_u-
blox_Tori_Kinematic.tmp
call :LatLonToENU %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\u-
blox_Tori_Static.tmp %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\ENU_u-
blox_Tori_Static.tmp

@echo .
@echo To be able to compare proterly these files, each row of them have to refer to the
same epoch.
@echo .

call :OrganizeFiles %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\ENU_u-
blox_Tori_Kinematic.tmp %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\u-
blox_Tori_Kinematic_Tmp.tmp 134533
```

```
call :OrganizeFiles %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\ENU_u-blox_Tori_Static.tmp %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\u-blox_Tori_Static_Tmp.tmp 134533


@echo Once this is done, we can proceed and compare both files:
@echo .
@echo .
paste -d " " %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\u-blox_Tori_Kinematic_Tmp.tmp %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\u-blox_Tori_Static_Tmp.tmp > %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\CompareEpochs.dat

:: Here all de plots are made
@echo .
call :plotDifColors4 %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\EAST_StaticVSKinematic.png "East Comparison" "East(m)" 2 346064 346070 134562 135622
@echo .
@echo .
call :plotDifColors4 %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\NORTH_StaticVSKinematic.png "North Comparison" "North (m)" 3 4972740 4972745 134562 135622
@echo .
@echo .
call :plotDifColors4 %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\HEI_StaticVSKinematic.png "Height Comparison" "Height (m)" 4 1497 1505 134562 135622
@echo .


call :computeDiferences %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\CompareEpochs.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\SolutionDiferences.dat 2 -2
call :computePercentages %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\Percentages.dat

call :plotDiferences %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\DIF_EAST_StaticVSKinematic.png^
 "Diferences between u-blox Kinematic and Static solutions" "East" 2 -5 2.5 134562 135622
@echo .
@echo .
```

```
call :plotDiferences %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\DIF_NORTH_StaticVSKinematic.png^
 "Diferences between u-blox Kinematic and Static solutions" "North" 3 -5 4 134562 135622
@echo .
@echo .
call :plotDiferences %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\SolutionDiferences.dat %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\DIF_HEIGHT_StaticVSKinematic.png^
 "Diferences between u-blox Kinematic and Static solutions" "Height" 4 -7 7 134562 135622

del /q %OUTPUT_U-BLOX_STATIC_VS_KINEMATIC%\*.tmp
@echo .
@echo .
@echo End of comparison
goto :EOF


:plotDifColors
:: Parameter 1 -> Input file containing the data to plot
:: Parameter 2 -> Path of the figure
:: Parameter 3 -> Title
:: Parameter 4 -> Y label
:: Parameter 5 -> Column of u-blox data
:: Parameter 6 -> Column of Xiaomi data
:: Parameter 7 -> Minimum valueY
:: Parameter 8 -> Maximum valueY
:: Parameter 9 -> Minimum valueX
:: Parameter 10 -> Maximum valueX
:: --------------------------------
set GNUPLOT_FILE=temp.gpl
set VAR_FILE=%~1
set FIGURE_NAME=%~2
set /a SEC_COL=%~5 + 5


@echo set title "%~3" > %GNUPLOT_FILE%
@echo set autoscale >> %GNUPLOT_FILE%
@echo set datafile separator " " >> %GNUPLOT_FILE%
@echo set grid >> %GNUPLOT_FILE%
@echo set ylabel "%~4" >> %GNUPLOT_FILE%
@echo set xlabel "Epoch (sec)" >> %GNUPLOT_FILE%
@echo set xrange [%~8:%~9] >> %GNUPLOT_FILE%
@echo set yrange [%~6:%~7] >> %GNUPLOT_FILE%
@echo set term png size 1024,768 >> %GNUPLOT_FILE%
@echo set output "%FIGURE_NAME:\=/%" >> %GNUPLOT_FILE%
@echo plot "%VAR_FILE:\=/%" u 1:($5==2?$%~5:1/0) w p lc rgb "#F9AF07" title "u-blox Float Solution", "%VAR_FILE:\=/%" u 1:($5==1?$%~5:1/0) w p lc rgb "#569A0C" title "u-blox Fixed Solution", "%VAR_FILE:\=/%" u 1:($10==2?$%SEC_COL%:1/0) w p lc rgb
```

```
"#41EDEC" title "Xiaomi Float Solution", "%VAR_FILE:\=/%" u 1:($10==1?$%SEC_COL%:1/0) w p lc rgb "#AE02FD" title "Xiaomi Fixed Solution" >> %GNUPLOT_FILE%
GNUPLOT -persist < %GNUPLOT_FILE:\=/% > nul 2>&1
@echo "<FIGURE %FIGURE_NAME%>"
del /q %GNUPLOT_FILE%
goto :EOF

:plotDifColors2
:: Parameter 1 -> Input file containing the data to plot
:: Parameter 2 -> Path of the figure
:: Parameter 3 -> Title
:: Parameter 4 -> Y label
:: Parameter 5 -> Column of u-blox data
:: Parameter 6 -> Column of Xiaomi data
:: Parameter 7 -> Minimum valueY
:: Parameter 8 -> Maximum valueY
:: Parameter 9 -> Minimum valueX
:: Parameter 10 -> Maximum valueX
:: ---------------------------------
set GNUPLOT_FILE=temp.gpl
set VAR_FILE=%~1
set FIGURE_NAME=%~2
set /a SEC_COL=%~5 + 5


@echo set title "%~3" > %GNUPLOT_FILE%
@echo set autoscale >> %GNUPLOT_FILE%
@echo set datafile separator " " >> %GNUPLOT_FILE%
@echo set grid >> %GNUPLOT_FILE%
@echo set ylabel "%~4" >> %GNUPLOT_FILE%
@echo set xlabel "Epoch (sec)" >> %GNUPLOT_FILE%
@echo set xrange [%~8:%~9] >> %GNUPLOT_FILE%
@echo set yrange [%~6:%~7] >> %GNUPLOT_FILE%
@echo set term png size 1024,768 >> %GNUPLOT_FILE%
@echo set output "%FIGURE_NAME:\=/%" >> %GNUPLOT_FILE%
@echo plot "%VAR_FILE:\=/%" u 1:($5==2?$%~5:1/0) w p lc rgb "#F9AF07" title "Xiaomi Float Solution", "%VAR_FILE:\=/%" u 1:($5==1?$%~5:1/0) w p lc rgb "#569A0C" title "Xiaomi Fixed Solution", "%VAR_FILE:\=/%" u 1:($10==2?$%SEC_COL%:1/0) w p lc rgb "#41EDEC" title "Samsumg Float Solution", "%VAR_FILE:\=/%" u 1:($10==1?$%SEC_COL%:1/0) w p lc rgb "#AE02FD" title "Samsumg Fixed Solution" >> %GNUPLOT_FILE%
GNUPLOT -persist < %GNUPLOT_FILE:\=/% > nul 2>&1
@echo "<FIGURE %FIGURE_NAME%>"
del /q %GNUPLOT_FILE%
goto :EOF

:plotDifColors3
:: Parameter 1 -> Input file containing the data to plot
```

```
:: Parameter 2 -> Path of the figure
:: Parameter 3 -> Title
:: Parameter 4 -> Y label
:: Parameter 5 -> Column of u-blox data
:: Parameter 6 -> Column of Xiaomi data
:: Parameter 7 -> Minimum valueY
:: Parameter 8 -> Maximum valueY
:: Parameter 9 -> Minimum valueX
:: Parameter 10 -> Maximum valueX
:: ---------------------------------
set GNUPLOT_FILE=temp.gpl
set VAR_FILE=%~1
set FIGURE_NAME=%~2
set /a SEC_COL=%~5 + 5


@echo set title "%~3" > %GNUPLOT_FILE%
@echo set autoscale >> %GNUPLOT_FILE%
@echo set datafile separator " " >> %GNUPLOT_FILE%
@echo set grid >> %GNUPLOT_FILE%
@echo set ylabel "%~4" >> %GNUPLOT_FILE%
@echo set xlabel "Epoch (sec)" >> %GNUPLOT_FILE%
@echo set xrange [%~8:%~9] >> %GNUPLOT_FILE%
@echo set yrange [%~6:%~7] >> %GNUPLOT_FILE%
@echo set term png size 1024,768 >> %GNUPLOT_FILE%
@echo set output "%FIGURE_NAME:\=/%" >> %GNUPLOT_FILE%
@echo plot "%VAR_FILE:\=/%" u 1:($5==2?$%~5:1/0) w p lc rgb "#F9AF07" title "Xiaomi Float Kinematic Solution", "%VAR_FILE:\=/%" u 1:($5==1?$%~5:1/0) w p lc rgb "#569A0C" title "Xiaomi Fixed Kinematic Solution", "%VAR_FILE:\=/%" u 1:($10==2?$%SEC_COL%:1/0) w p lc rgb "#41EDEC" title "Xiaomi Float Static Solution", "%VAR_FILE:\=/%" u 1:($10==1?$%SEC_COL%:1/0) w p lc rgb "#AE02FD" title "Xiaomi Fixed Static Solution" >> %GNUPLOT_FILE%
GNUPLOT -persist < %GNUPLOT_FILE:\=/% > nul 2>&1
@echo "<FIGURE %FIGURE_NAME%>"
del /q %GNUPLOT_FILE%
goto :EOF

:plotDifColors4
:: Parameter 1 -> Input file containing the data to plot
:: Parameter 2 -> Path of the figure
:: Parameter 3 -> Title
:: Parameter 4 -> Y label
:: Parameter 5 -> Column of u-blox data
:: Parameter 6 -> Column of Xiaomi data
:: Parameter 7 -> Minimum valueY
:: Parameter 8 -> Maximum valueY
:: Parameter 9 -> Minimum valueX
:: Parameter 10 -> Maximum valueX
:: ---------------------------------
```

```
set GNUPLOT_FILE=temp.gpl
set VAR_FILE=%~1
set FIGURE_NAME=%~2
set /a SEC_COL=%~5 + 5


@echo set title "%~3" > %GNUPLOT_FILE%
@echo set autoscale >> %GNUPLOT_FILE%
@echo set datafile separator " " >> %GNUPLOT_FILE%
@echo set grid >> %GNUPLOT_FILE%
@echo set ylabel "%~4" >> %GNUPLOT_FILE%
@echo set xlabel "Epoch (sec)" >> %GNUPLOT_FILE%
@echo set xrange [%~8:%~9] >> %GNUPLOT_FILE%
@echo set yrange [%~6:%~7] >> %GNUPLOT_FILE%
@echo set term png size 1024,768 >> %GNUPLOT_FILE%
@echo set output "%FIGURE_NAME:\=/%" >> %GNUPLOT_FILE%
@echo plot "%VAR_FILE:\=/%" u 1:($5==2?$%~5:1/0) w p lc rgb "#F9AF07" title "u-blox Float Kinematic Solution",^
 "%VAR_FILE:\=/%" u 1:($5==1?$%~5:1/0) w p lc rgb "#569A0C" title "u-blox Fixed Kinematic Solution",^
 "%VAR_FILE:\=/%" u 1:($10==2?$%SEC_COL%:1/0) w p lc rgb "#41EDEC" title "u-blox Float Static Solution",^
 "%VAR_FILE:\=/%" u 1:($10==1?$%SEC_COL%:1/0) w p lc rgb "#AE02FD" title "u-blox Fixed Static Solution" >> %GNUPLOT_FILE%
GNUPLOT -persist < %GNUPLOT_FILE:\=/% > nul 2>&1
@echo "<FIGURE %FIGURE_NAME%>"
del /q %GNUPLOT_FILE%
goto :EOF

:plotDiferences
:: Parameter 1 -> Input file containing the data to plot
:: Parameter 2 -> Path of the figure
:: Parameter 3 -> Title
:: Parameter 4 -> Y label
:: Parameter 5 -> Column of u-blox data
:: Parameter 6 -> Column of data
:: Parameter 7 -> Minimum valueY
:: Parameter 8 -> Maximum valueY
:: Parameter 9 -> Minimum valueX
:: Parameter 10 -> Maximum valueX
:: --------------------------------
set GNUPLOT_FILE=temp.gpl
set VAR_FILE=%~1
set FIGURE_NAME=%~2


@echo set title "%~3" > %GNUPLOT_FILE%
@echo set autoscale >> %GNUPLOT_FILE%
@echo set datafile separator " " >> %GNUPLOT_FILE%
```

```
@echo set grid >> %GNUPLOT_FILE%
@echo set ylabel "%~4" >> %GNUPLOT_FILE%
@echo set xlabel "Epoch (sec)" >> %GNUPLOT_FILE%
@echo set xrange [%~8:%~9] >> %GNUPLOT_FILE%
@echo set yrange [%~6:%~7] >> %GNUPLOT_FILE%
@echo set term png size 1024,768 >> %GNUPLOT_FILE%
@echo set output "%FIGURE_NAME:\=/%" >> %GNUPLOT_FILE%
@echo plot "%VAR_FILE:\=/%" u 1:($5==2?$%~5:1/0) w p lc rgb "#F9AF07" title
"Diferences in Float Solutions",^
 "%VAR_FILE:\=/%" u 1:($5==1?$%~5:1/0) w p lc rgb "#569A0C" title "Diferences in Fixed
Solutions",^
 "%VAR_FILE:\=/%" u 1:7 w l linewidth 3 lc rgb "#FF0000" notitle,^
 "%VAR_FILE:\=/%" u 1:6 w l linewidth 3 lc rgb "#FF0000" title "Threshold" >>
%GNUPLOT_FILE%
GNUPLOT -persist < %GNUPLOT_FILE:\=/% > nul 2>&1
@echo "<FIGURE %FIGURE_NAME%>"
del /q %GNUPLOT_FILE%
goto :EOF

:OrganizeFiles
:: Parameter 1 -> Input file
:: Parameter 2 -> Output file
:: Parameter 3 -> Initial Epoch
:: --------------------------------
gawk 'BEGIN {^
                Epoch=%~3;^
        }^
        {^
        if (Epoch ^=^= $2) {^
                printf("%%s %%s %%s %%s %%s\n",$2,$3,$4,$5,$6);^
                Epoch++;^
        }^
        else {^
                EpochsDif=$2-Epoch;^
                for (i=0;i^< EpochsDif;i++){^
                        printf("%%s %%s %%s %%s %%s\n",Epoch,0,0,0,0);^
                        Epoch++;^
                }^
                printf("%%s %%s %%s %%s %%s\n",$2,$3,$4,$5,$6);^
                Epoch++;^
        }^
}' %~1 > %~2

goto :EOF

:SaveFloatAndFix
:: Parameter 1 -> Input file
:: Parameter 2 -> Output file1
:: Parameter 3 -> Output file2
```

```
:: --------------------------------
gawk '{if ($5 ^=^= 2) {^
                printf("%%s %%s %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6);^
        }^
}' %~1 > %~2

gawk '{if ($5 ^=^= 1) {^
                printf("%%s %%s %%s %%s %%s %%s\n",$1,$2,$3,$4,$5,$6);^
        }^
}' %~1 > %~3
goto :EOF


:computeDiferences
:: Parameter 1 -> Input file
:: Parameter 2 -> Output file
:: Parameter 3 -> Rover1
:: Parameter 4 -> Rover2
:: Parameter 5 -> Master
:: Parameter 6 -> Navigation
:: --------------------------------
gawk 'BEGIN {^
                printf("%%s EPOCH  EAST  NORTH  HEIGHT  Q  Threshold\n","%%");^
        }^
        {^
        if (NF ^> 4 ^&^& NR ^> 60) {^
                if ($2 ^!^= 0 ^&^& $7 ^!^= 0) {^
                        if ($5 ^=^= 1 ^&^& $10 ^=^= 1) {^
                                printf("%%s    %%.4f    %%.4f    %%.4f    %%s    %%s
%%s\n",$1,$2-$7,$3-$8,$4-$9,1,%~3,%~4);^
                        }^
                        if ($5 ^=^= 2 ^&^& $10 ^=^= 2) {^
                                printf("%%s    %%.4f    %%.4f    %%.4f    %%s    %%s
%%s\n",$1,$2-$7,$3-$8,$4-$9,2,%~3,%~4);^
                        }^
                }^
        }^
}' %~1 > %~2


goto :EOF

:computePercentages
:: Parameter 1 -> Input file
:: Parameter 2 -> Output file

:: --------------------------------
gawk 'BEGIN {^
                EastEpoch=0;^
```

```
			NorthEpoch=0;^
			HeightEpoch=0;^
			TotalEpochs=0;^
			EastPer=0;^
			NorthPer=0;^
			HeightPer=0;^
}^
{^
		TotalEpoch ^+^= 1;^
		if (sqrt($2*$2) ^< $6){^
			EastEpoch ^+^= 1;^
		}^
		if (sqrt($3*$3) ^< $6){^
			NorthEpoch ^+^= 1;^
		}^
		if (sqrt($4*$4) ^< $6){^
			HeightEpoch ^+^= 1;^
		}^
}^
END{^
		EastPer = EastEpoch/TotalEpoch;^
		NorthPer = NorthEpoch/TotalEpoch;^
		HeightPer = HeightEpoch/TotalEpoch;^
		printf(":: EPOCH  E(Per) N(Per) H(Per)\n");^
		printf("   %%s    %%s     %%s     %%s\n",$1,EastPer,NorthPer,HeightPer);^
}' %~1 > %~2

goto :EOF

:LatLonToENU
:: Parameter 1 -> Input file
:: Parameter 2 -> Output file
:: ---------------------------------
gawk 'BEGIN {^
			Pi = 3.14159265359;^
			H = 32;^
			Major_Axis = 6378137;^
			Minor_Axis = 6356752.31424518;^
			Ecce = 0.0818191908426203;^
			Ecce2 = 0.0820944379496945;^
			EcceEl2 = 0.00673949674227624;^
			CurRadius = 6399593.62575849;^
		}^

		{^
			LatRad = $3 * Pi / 180;^
			LonRad = $4 * Pi / 180;^
			MeridianH = (6 * H) - 183;^
			DLambda = LonRad - ((MeridianH * Pi)/180);^
```

```
			A = cos(LatRad) * sin(DLambda);^
			Xi = 0.5 * log((1 + A)/(1 - A));^
			PreEta = (((sin(LatRad) / cos (LatRad))) / cos(DLambda)) - LatRad;^
			Eta = atan2(sin(LatRad) / cos(LatRad),cos(DLambda)) - LatRad;^
			Ni = (CurRadius/(sqrt(1 + EcceEl2 * cos(LatRad) * cos(LatRad)))) *
0.9996;^
			Zeta = (EcceEl2 / 2) * Xi * Xi * cos(LatRad) * cos(LatRad);^
			A1 = sin(2 * LatRad);^
			A2 = A1 * cos(LatRad) * cos(LatRad);^
			J2 = LatRad + (A1 / 2);^
			J4 = ((3 * J2) + A2) / 4;^
			J6 = (5 * J4 + A2 * (cos(LatRad) * cos(LatRad))) / 3;^
			Alfa = 3 / 4 * EcceEl2;^
			Beta = (5 / 3) * Alfa * Alfa;^
			Gamma = (35 / 27) * Alfa * Alfa * Alfa;^
			Fi = 0.9996 * CurRadius * (LatRad - (Alfa * J2) + (Beta * J4) - (Gamma *
J6));^
			East = (Xi * Ni * (1 + (Zeta / 3))) + 500000;^
			North = Eta * Ni * (1 + Zeta) + Fi;^
			printf("%%s %%s %%f %%f %%s %%s\n",$1,$2,East,North,$5,$6);^
}' %~1 > %~2


goto :EOF


:meanPosValue
:: Parameter 1 -> Input file
:: ---------------------------------
gawk 'BEGIN {^
	Lon=0;^
	Lat=0;^
	Hei=0;^
	Epoch=0;^
	}^
	{^
	if (NR ^> 25 ^&^& $6 ^=^= 1) {^
		Lat+=$3;^
		Lon+=$4;^
		Hei+=$5;^
		Epoch++;^
		}^
	}^
	END {^
		MeanLon=Lon/Epoch;^
		MeanLat=Lat/Epoch;^
		MeanHei=Hei/Epoch;^
		printf("MEAN LATITUDE: %%.9fMEAN LONGITUDE: %%.9f	MEAN
HEIGHT: %%.9f\n",MeanLat,MeanLon,MeanHei);^
```

```
                    printf("FIX SOLUTION EPOCHS %%s\n",Epoch);^
}' %~1 > %INPUT_TORI_TORI%/MeanFixValues.dat

::
==================================================================
==============
:set_environment_paths
::
==================================================================
==============
:: DESCRIPTION: Sets the environment paths (cygwin, GNUPLOT)              =
::
==================================================================
==============
:: %~1 = set_environment_paths                                    =
::
==================================================================
==============
:: Setting cygwing path
set newPATH=%PATH:;C:\cygwin64\bin=%
set PATH=C:\cygwin64\bin;%newPATH%
set newPATH=%PATH:;C:\cygwin\bin=%
set PATH=C:\cygwin\bin;%newPATH%
set CYGWIN=nodosfilewarning

:: Setting GNUPLOT path
set newPATH=%PATH:;C:\Program Files (x86)\GNUPLOT\bin=%
set PATH=%newPATH%;C:\Program Files (x86)\GNUPLOT\bin
REM set newPATH=%PATH:;C:\Program Files\GNUPLOT\bin=%
REM set PATH=%newPATH%;C:\Program Files\GNUPLOT\bin
goto :EOF
```