Faculty of Civil and Industrial Engineering
Department of Civil, Constructional and Environmental Engineering

Master Degree in Transport Systems Engineering

Raw GNSS Measurements Provided By Android 7 Smartphones: Preliminary Results Of Vehicles Precise PVT Estimation

Candidate: Edoardo Fornaciari
Number: 1384016

Advisor: Augusto Mazzoni

Academic Year 2017-2018

I think this is the best advice:
always think about
how you could do things better and
question yourself.

Elon Musk

Contents

Contents **iv**

1 Introduction **3**

1.1 GNSS Architecture 4

1.2 Satellite Constellations 5

1.3 GNSS Signals 8

2 Observation Techniques **11**

2.1 Code Pseudorange Measurements 12

2.2 Phase Pseudorange Measurements 13

2.3 GNSS Observable Errors 13

3 Data Acquisition with a Smartphone **17**

3.1 Location API 23 17

3.1.1 Location API in Android 7 19

3.2 GNSS Raw Measurements description 20

3.2.1 GNSS Time Generation 20

3.3 The RINEX Format 23

4 Measurements Processing **29**

4.1 Preliminary Tests with GNSS Raw Analysis Software for Android Developers 29
4.1.1 From the GNSS Raw Analysis Software for Android Developers to the RINEX format 33
4.1.2 Post - Processing GNSS data with RTKLIB . . 34
4.2 The Single Point Positioning 36
4.2.1 Preliminary Tests 36
4.2.2 The Geo++ RINEX Logger 39
4.2.3 Tests with Geo++ 40
4.3 The Relative Positioning 53

5 The Variometric Approach 63
5.1 The Static Approach 66
5.1.1 The Duty - Cycle 69
5.1.2 The Static Approach during Burst Period . . . 72
5.2 The Kinematic Approach 75
5.2.1 The Flight Test 77

6 Conclusions 87

Appendix A 89
.1 Python Codes 89
.2 VADASE Configuration 92

Bibliography 96

List of Figures

1.1 GNSS Architecture [1] 5
1.2 Type of Satellites: GPS IIR-M (top left), Glonass-M (top right), Galileo IOV (bottom left) and Beidou-M (bottom right). [1] 8
1.3 GPS, Glonass, Galileo and Beidou navigational frequency bands. [1] 9

2.1 Basic concept of range measurement [8] 12
2.2 Observable Errors [8] 16

3.1 Location API in Android API Level 23 [10] 18
3.2 Location API in Android API Level 24/25/26[10] . . . 20
3.3 Pseudorange generation from Android methods[9] . . . 22
3.4 RINEX Example obtained in February 28

4.1 Android Features Pre-A8 Update - November 2017 [12] 30
4.2 Samsung Galaxy S8 Scheme [13] 31
4.3 GNSS Raw Analysis Software Interface [12] 33
4.4 Preliminaty Output [12] 34
4.5 RINEX Formar Header 35

4.6 RTK U.I. [14] 35
4.7 Single Point Test - 30^th November 37
4.8 Single Epoch vs. NMEA - 30^th November [19] 39
4.9 Geo++ Logger User Interface [16] 40
4.10 1^{st} Configuration [19] 41
4.11 2^{nd} Configuration [19] 41
4.12 First Configuration - Devices Positions 43
4.13 Geodedic Coordinates of a point P [6] 45
4.14 System oriented with orthogonal axes, East, Nord, Up [6] 47
4.15 R1/T2 Single Positioning - GPS 51
4.16 R1/T2 Single Positioning - Mean Satellites 51
4.17 R2/T2 Single Positioning - GPS 52
4.18 R2/T2 Single Positioning - Mean Satellites 52
4.19 The Relative Positioning [8] 54
4.20 Double Difference Scheme [8] 55
4.21 R1T2 - R2T2 Fix Distance 61

5.1 Processing Scheme - VADASE 65
5.2 The Duty - Cycle 70
5.3 R1T3 - G+E - LOOCV ON - Velocities 71
5.4 R1T3 - G+E - LOOCV ON - Burst Period Velocities . 72
5.5 R2T3 - G+E - LOOCV ON - Burst Period Velocities . 73
5.6 Processing Scheme - Kin VADASE 76
5.7 Tecnam P92 - General View 77
5.8 Inside view of the cokpit with the S8 fixed, approaching Civitavecchia, flying over Santa Marinella 78
5.9 Number of Satellites, G+E, LOOCV ON 79

5.10 Route of Flight in single positioning 80
5.11 Route near the Santa Severa Nord airfield - Single Point Positioning 80
5.12 P92 - Performance 82
5.13 Flight Velocities 83
5.14 Flight Velocities during Take Off 83
5.15 Flight Velocities - Turning Point 84
5.16 Flight Velocities during the landing 84
5.17 SPP Vs. Kin-VADASE - Airfield 85
5.18 SPP Vs. Kin-VADASE - Port of Civitavecchia 86

List of Tables

4.1 Measurements Analyzed with Excel on February 27 . . 44
4.2 Measurements Analyzed with Python on February 27 . 49
4.3 Static Positioning - All Solutions - February 27 57
4.4 Static Positioning - Single Solutions - February 27 . . . 57
4.5 Static Positioning - All Solutions/Geodedical Coordinates February 27 58
4.6 Static Positioning - Single Solution/Geodedical Coordinates February 27 58
4.7 Static Positioning - All Solutions (Mean): Distances between receivers 59
4.8 Static Positioning - Single Solutions (Mean): Distances between receivers 59
4.9 Distance Difference Between All and Single Solutions . 60
4.10 Distance Between Devices - Median / Mean 60

5.1 VADASE Results - Static Positioning - February 27 . . 68
5.2 VADASE Results - Burst Period - February 27 73

Abstract

In recent years, the use of Global Navigation Satellite Systems (GNSS) positioning has increased significantly. Its applications range from civil engineering use, control and construction of infrastructures, to the transport sector for navigation. Positioning in real time for navigation can be captured at high performance using techniques such as Real Time Kinematic (RTK) and Precise Point Positioning (PPP). These techniques allow precision ranging from one to a few centimeters. Despite these high performances, the application of these techniques requires high costs, which reach almost ten thousand euros per receiver.
The main objective of this work is to research and evaluate performance in precise positioning of the chip contained in a smartphone. This is to analyse and verify if it is possible to achieve high performance even with an economic chip of about fifty euro at single L1 frequency. The work was based, in a first phase, on analysis with post-processing techniques in conjunction with a reference GNSS network, considering all satellite constellations, able to be picked up by the smartphone. Subsequently, the second phase of the work was based on the variometric approach, using the

VADASE (Variometric Approach for Displacement Analysis Stand-alone Engine) algorithm, using the carrier phase but without external data coming from a GNSS network.
This innovative work was possible thanks to the release of the version of the Android 7 operating system. With this O.S. Google, has made it possible to acquire raw data, especially the carrier phase, from the smartphone. However, raw data is not available directly in a standard format and for this reason, before steps one and two, the work focused on reading and converting into a standard format.
The analyses were carried out, also studying the accuracy of the solution considering the various satellite constellations (GPS, GLONASS, Galileo, Beidou). In fact, more satellites generally improve the accuracy of the solution. About the variometric approach, the combinations between GPS and Galileo were studied in particular.
The tests were carried out in static mode and also in cinematic mode, with the aim of studying the measurements of the smartphone in a moving vehicle. If the results were satisfactory, numerous low cost, high-performance alternatives could be opened, and this could revolutionize private transportation and beyond.

Chapter 1

Introduction

A GNSS involves a constellation of satellites orbiting Earth, continuously transmitting signals that enable users to determine their three-dimensional (3D) position with global coverage. The basic observable in a GNSS is the time required for a signal to travel from the satellite (transmitter) to the receiver. This travel time, multiplied by the speed of light, provides a measure of the apparent distance (pseudorange) between them.

For many years, the only fully operational GNSS system was the US Global Positioning System (GPS). The Russian GLObal NAvigation Satellite System (Glonass) was restored to full operation in December 2011. The Chinese BeiDou and European Galileo systems are currently under development, although Beidou started an initial operating service (Phase II) in late December 2011.

The positioning principle is based on solving an elemental geometric problem, involving the distances (ranges) of a user to a set of at least four GNSS satellites with known coordinates. These ranges and

satellite coordinates are determined by the user's receiver using signals and navigation data transmitted by the satellites; the resulting user coordinates can be computed to an accuracy of several metres. However, centimetre-level positioning can be achieved using more advanced techniques [1].

1.1 GNSS Architecture

The GNSS basically consists of three main segments: the space segment, which comprises the satellites; the control segment (also referred to as the ground segment), which is responsible for the proper operation of the system; and the user segment, which includes the GNSS receivers providing positioning, velocity and precise timing to users.

The main functions of the space segment are to generate and transmit code and carrier phase signals, and to store and broadcast the navigation message uploaded by the control segment. These transmissions are controlled by highly stable atomic clocks onboard the satellites. The GNSS space segments are formed by satellite constellations with enough satellites to ensure that users will have at least four satellites in view simultaneously from any point on Earth's surface at any time.

The control segment (also referred to as the ground segment) is responsible for the proper operation of the GNSS. Its basic functions are:

- to control and maintain the status and configuration of the satellite constellation;
- to predict ephemeris and satellite clock evolution;

- to keep the corresponding GNSS time scale (through atomic clocks);
- to update the navigation messages for all the satellites.

The user segment is composed of GNSS receivers. Their main function is to receive GNSS signals, determine pseudoranges (and other observables) and solve the navigation equations in order to obtain the coordinates and provide a very accurate time [1]. The basic elements of a generic GNSS receiver are: an antenna with preamplication, a radio frequency section, a microprocessor, an intermediate-precision oscillator, a feeding source, some memory for data storage and an interface with the user. The calculated position is referred to the antenna phase centre.

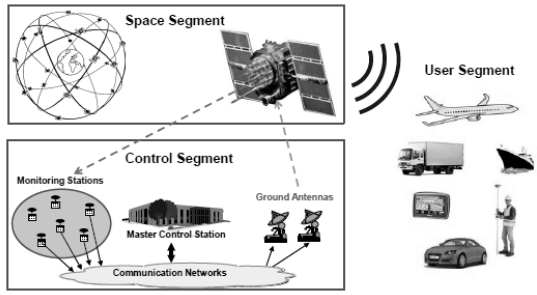

Fig. 1.1: GNSS Architecture [1]

1.2 Satellite Constellations

Satellites have various structures and mechanisms to keep them in orbit, communicate with the control segment and broadcast signals

to receivers. The satellite clocks are one of the critical components of GNSSs. For this reason, satellites are equipped with very high-stability atomic clocks (rubidium, caesium, hydrogen)[1].

The GPS satellites are arranged in six equally spaced orbital planes surrounding Earth, each with four slots occupied by baseline satellites. This 24-slot arrangement ensures there are at least four satellites in view from virtually any point on the planet.2 The satellites are placed in a Medium Earth Orbit (MEO) orbit, at an altitude of 20 200 km and an inclination of 55 relative to the equator. Orbits are nearly circular, with an eccentricity of less than 0:02, a semi-major axis of 26 560 km and a nominal period of 11 hours, 58 minutes and 2 seconds (12 sidereal hours), repeating the geometry each sidereal day [5].

The nominal Glonass constellation consists of 24 MEO satellites deployed in three orbital planes with eight satellites equally spaced in each plane. The orbits are roughly circular, with an inclination of about 64,8, and at an altitude of 19 100 km with a nominal period of 11 hours, 15 minutes and 44 seconds, repeating the geometry every eight sidereal days. Due to funding problems, the number of satellites decreased from the 24 available in 1996 to only 6 in 2001. In August 2001, the Russian government committed to recover the constellation and to modernise the system, approving new funding. A total of 24 operational satellites plus 2 in maintenance were again available in December 2011, restoring the full constellation [4].

The planned Galileo constellation in Full Operational Capability (FOC) phase consists of 27 operational and 3 spare MEO satellites at an altitude of 23 222 km and with an orbit eccentricity of 0:002. Ten satellites will occupy each of three orbital planes inclined at an angle of 56 with respect to the equator. The satellites will be spread

around each plane and will take about 14 hours, 4 minutes and 45 seconds to orbit Earth, repeating the geometry each 17 revolutions, which involves 10 sidereal days. This constellation guarantees, under nominal operation, a minimum of six satellites in view from any point on Earth's surface at any time, with an elevation above the horizon of more than 10. The Galileo Deployment Plan has two main phases: (1) the In-Orbit Validation (IOV) phase with a reduced constellation of four operational satellites and their related ground infrastructure (2012); and (2) the FOC that involves the deployment of the remaining ground and space infrastructure, including an intermediate initial operational capability phase (by 2014 - 2016) with 18 satellites in operation (the 4 IOV satellites plus 14 others). Completion of the FOC phase is expected by 2019 - 2020 [3].

The Beidou (Compass) constellation (Phase III) will consist of 35 satellites, including 5 Geostationary Orbit (GEO) satellites and 30 non-GEO satellites in a nearly circular orbit. The non-GEO satellites include 3 Inclined Geosynchronous Satellite Orbit (IGSO) ones, with an inclination of about 55, and 27 MEO satellites orbiting at an altitude of 21 528km in three orbital planes with an inclination of about 55 and with an orbital period of about 12 hours and 53 minutes, repeating the ground track every seven sidereal days. The GEO satellites, orbiting at an altitude of about 35 786 km, are positioned at 58:75E, 80E, 110.5E, 140E and 160E, respectively, and are expected to provide global navigation service by 2020. The previous Phase II involves a reduced constellation of four MEO, five GEO and five IGSO satellites to provide regional coverage of China and surrounding areas. The initial Phase II operating service with 10 satellites started on 27 December 2011 [2].

Fig. 1.2: Type of Satellites: GPS IIR-M (top left), Glonass-M (top right), Galileo IOV (bottom left) and Beidou-M (bottom right). [1]

1.3 GNSS Signals

GNSS satellites continuously transmit navigation signals at two or more frequencies in L band. These signals contain ranging codes and navigation data to allow users to compute both the travel time from the satellite to the receiver and the satellite coordinates at any epoch. The main signal components are described as follows:

- Carrier: Radio frequency sinusoidal signal at a given frequency;
- Ranging code: Sequences of zeros and ones which allow the receiver to determine the travel time of the radio signal from the satellite to the receiver. They are called PRN sequences or PRN codes;
- Navigation data: A binary-coded message providing information on the satellite ephemeris (pseudo-Keplerian elements or satellite position and velocity), clock bias parameters, almanac (with a reduced-accuracy ephemeris data set), satellite health status and other complementary information.

The allocation of frequency bands is a complex process because multiple services and users can fall within the same range. That is, the same frequencies can be allocated for different purposes in different countries. The ITU is a United Nations agency coordinating the shared global use of the radio spectrum. It involves, for instance, television, radio, cell (mobile) phone, radar satellite broadcasting, etc., and even microwave ovens. The ITU divides the electromagnetic spectrum into frequency bands, with different radio services assigned to particular bands.

There are two bands in the region allocated to the Aeronautical Radio Navigation Service (ARNS) on a primary basis worldwide. These bands are especially suitable for Safety-of-Life (SoL) applications because no other user of this band is allowed to interfere with the GNSS signals. These correspond to the upper L band (1559 1610 MHz), containing the GPS L1, Galileo E1, Glonass G1 and Beidou B1 bands, and to the bottom of the lower L band (1151 1214 MHz) where the GPS L5, Glonass G3, Galileo E5 and Beidou B2 bands are located. It is schematized below:

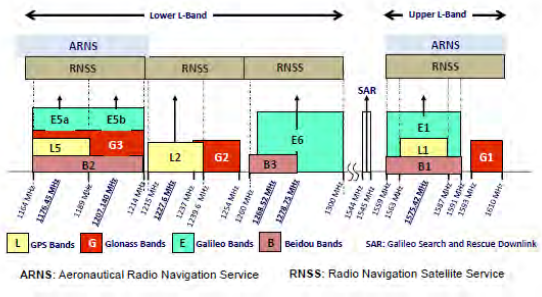

Fig. 1.3: GPS, Glonass, Galileo and Beidou navigational frequency bands. [1]

The remaining GPS L2, Glonass G2, Galileo E6 and Beidou B3 signals are in the 1215.6 - 1350MHz bands. These bands were allocated to radio location services (ground radars) and RNSS on a primary basis, so the signals in these bands are more vulnerable to interference than the previous ones [1].

Chapter 2

Observation Techniques

The basic concept of GNSS is to measure the signal traveling time between artificial satellite and receiver. By multiplying this time by the light velocity (c), we get the range between the satellite and the receiver.

$$Range = c \cdot (t_R - t^S) = \Delta t_R^S \cdot c \tag{2.1}$$

The time or phase measurement performed by the receiver is based on the comparison between the received signal at the antenna of the receiver and the generated reference signal by the receiver. The two signals are affected by the clocks errors. Therefore, the range measured is not true and it is called pseudorange. Since the signal travels through the atmospheric layers, further noise should be modeled in order to compute the precise range. [1]

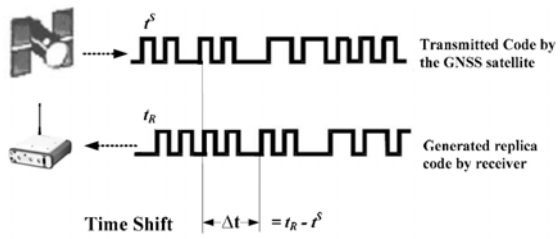

Fig. 2.1: Basic concept of range measurement [8]

2.1 Code Pseudorange Measurements

Code correlation technique is used to measure the time difference between the received and generated replica code. The range could be formulated as follows:

$$R_R^S = c \cdot [(t_R - \delta_R) - (t^S - \delta^S)] \tag{2.2}$$

where δ^S is the satellite clock offset and δ_R is the receiver clock offset. An high stability atomic clock is generally used on board of the satellite, so δ^S is small and could be modeled by a polynomial with the coefficients being transmitted in the navigation message. However, the receiver clock offset δ_R is large and is treated as unknown to be estimated in the function: [6] [8]

$$R_R^S = c \cdot \Delta t + c \cdot (\delta^S - \delta_R) = \rho + c \cdot \Delta\delta \tag{2.3}$$

where ρ is the true distance between satellite and receiver and its expressed by the vector in reference geocentric coordinate system as:

$$\rho = \sqrt{(X^S - X_R)^2 + (Y^S - Y_R)^2 + (Z^S - Z_R)^2} \tag{2.4}$$

2.2 Phase Pseudorange Measurements

Phase pseudo range is based on the measurements of phase difference between the received and generated signal $\Delta\varphi_R^S$ at the receiver. The received carrier is Doppler shifted due to the motion of satellite.

In order to calculate the range using phase measurement, we have to add to $\Delta\varphi_R^S$ the number of cycles between the satellite and the receiver, which is an ambiguous value and is often called ambiguity (N). By considering the initial phase errors of the satellite and receiver due to their clocks, the mathematical model of phase pseudo range can be expressed by:

$$\Delta\varphi_R^S + N = -\frac{f}{c} \cdot \rho - f\delta^S + f\delta_R \tag{2.5}$$

If we rearrange the above equation and use $\Phi = -\Delta\varphi_R^S$ and $\Delta\delta = \delta^S - \delta_R$, the it becomes similar to the code pseudo range equation, but with the additional the ambiguity value (N):

$$\lambda \cdot \Phi = \rho + c \cdot \Delta\delta + \lambda \cdot N \tag{2.6}$$

where λ is the wave length. [6] [8]

2.3 GNSS Observable Errors

The code and phase measurements are affected by noise and errors due to the propagation of signals through atmospheric layers and due to the noise measurements. These errors can described briefly as below:

- Satellite clock error: This can be modeled by the polynomial

coefficients transmitted in the navigation message with respect to a reference time.

$$\delta^S = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 \tag{2.7}$$

- Orbital error: This can be eliminated by differential positioning. Precise orbits could be obtained in near real time via Internet from the services centers such as International GNSS Service (IGS).

- Ionospheric error: This error is modeled or eliminated by using the linear combination of two or multiple frequencies. The relation between the ionospheric effect on the future GNSS (L5, L2 and L1 for GPS; E5a, E5b and E1 for GALILEO) using the triple frequency could be written as follows:

$$\lambda_1 \cdot \Phi_1 = \rho + c \cdot \Delta\delta + \lambda_1 \cdot N - I_{L1} \tag{2.8}$$

$$\lambda_2 \cdot \Phi_1 = \rho + c \cdot \Delta\delta + \lambda_2 \cdot N - \frac{f_1^2}{f_2^2} I_{L1} \tag{2.9}$$

$$\lambda_3 \cdot \Phi_3 = \rho + c \cdot \Delta\delta + \lambda_3 \cdot N - \frac{f_1^2}{f_3^2} I_{L1} \tag{2.10}$$

where, Ionosphere = I_{L1}. The effect of ionosphere on GNSS measurement is of special interest in solving the ambiguity number N [1]. Having multiple frequency can give more advantages for ionosphere models to estimate the first and second order effect of the ionosphere. Moreover, it allows more possibilities in

ambiguity resolution process. Ionosphere could also be modeled using the ionospheric coefficient transmitted by the navigation message. [8]

- The troposphere: This consists of two layers: Wet layer (up to 10 km above the surface of ground), and dry layer from 10 to 40 km above the ground. Troposphere causes a delay in both the code and carrier observations. Since it is not frequency dependent, it cannot be canceled out by using dual frequency measurements but it can, however, be successfully modeled. Tropospheric models depend on empirical models by considering all values of temperature, pressure, relative humidity and mapping function. Examples of such models are the Hopfield, and Saastamoninen models. [6]

- Receiver clock error: This is due to using non-precise clock in the receiver (quartz clock), which causes offset and drift in the receiver clock and GNSS reference time. This error is treated as unknown in the pseudo range computations. The clock receiver error could be eliminated in double difference equation as shown in the follow section.

- Multipath: This is caused by multiple reflections of the signals at the receiver or at the satellite due to multiple paths taken by the signal to arrive to the destination. The best way to reduce multipath phenomenon is to choose the site away from reflection surface (such as buildings, cars, trees, etc), and by appropriate antenna design. Carrier phase are less affected by multipath propagation than code ranges, because multipath is frequency

dependent. The multipath error could reach to a one meter level. The elimination of multipath is possible by selecting an antenna that takes advantages of the signal polarization. [8]

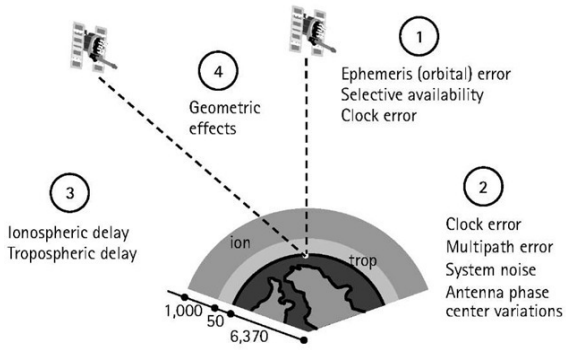

Fig. 2.2: Observable Errors [8]

Chapter 3

Data Acquisition with a Smartphone

The Android operating system has ad-hoc interface called API[1] that allows users to access the systems functionalities. Access to the GNSS measurements in the Android devices were added in API level 24 embedded in Android N using the GnssMeasurement and GNSSClock classes. [9]

3.1 Location API 23

Figure 3.1 shows the structure of the Location API with Android 23 system (version 6 or lower) drivers, the configuration with the GNSS chipset, and the handles of the data transmission between the chipset and the OS.

Different user applications can access the GNSS Data using the

[1] API = Application Programming Interface

framework of Location API. Until the arrival of Android API 23, the only data that could be accessed were GPS satellite information (C/No, azimuth, elevation, and any satellite used in the PVT), NMEA sentences, and PVT solutions with the proper time stamp.

Users are not able to configure the GNSS chipset, but they can send basic configuration commands such as those to restart/start the GNSS chipset or clean the assisted data. All the configuration settings including GNSS constellation priorities and the different PVT algorithms are driven by the chipset. [10]

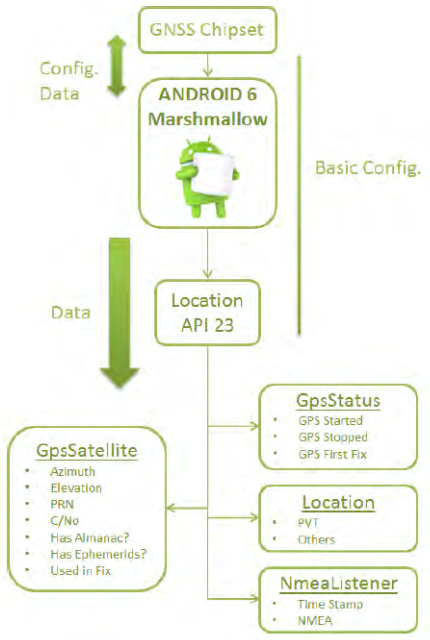

Fig. 3.1: Location API in Android API Level 23 [10]

3.1.1 Location API in Android 7

Location API level 24 provides raw GNSS measurements, GNSS clocks, and GNSS measurements. Hereafter, they are referred to as raw measurements. Figure 3.2 shows the Location API level 27 (Android 7). From API 24 (Android 7), developers have access to the following Android classes:

- GNSS Clock: it contains the following GNSS raw and computed information.
  - Receiver time (used to compute the pseudorange)
  - Clock bias
- GNSS Navigation Message: it contains the following GNSS raw and computed information.
  - Bits of the Navigation Message (all the constellations)
  - Navigation Message status
- GNSS Measurement: it contains the following GNSS raw and computed information.
  - Received Satellite Time (used to compute the pseudorange)
  - Code
  - Phase

While this data is directly received from the GNSS chipset, we do not have direct access to the chipset.

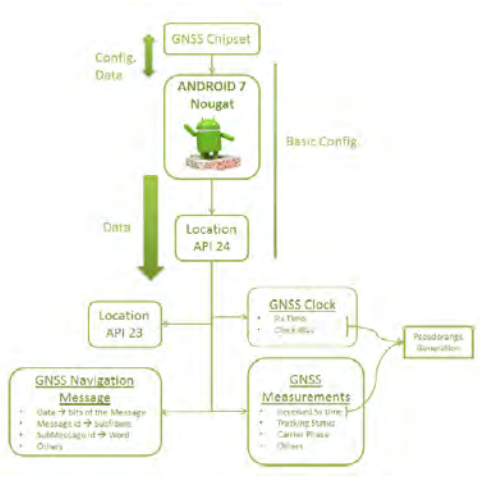

Fig. 3.2: Location API in Android API Level 24/25/26[10]

3.2 GNSS Raw Measurements description

Android does provide a straightforward pseudorange and GNSS time. To obtain these, GNSS engineering is required. This paragraph explains how to obtain typical GNSS parameters from the raw measurements.

3.2.1 GNSS Time Generation

Time and distance units provided by Android are expressed in nanosecond and the measurements are split in different parameters accessible through different methods described as follow:

- getTimeNanos() gets the GNSS receiver internal hardware clock value in nanoseconds. Its a counter of the time the Android device has been powered on.

- getFullBiasNanos() gets the difference between hardware clock inside GPS receiver and the true GPS time since January 6, 1980 (GPS time start reference). This value is available if the receiver has estimated the GPS time. If the computed time is for a non-GPS constellation, the time offset of that constellation to GPS has to be applied to fill this value. This value should be used only once, at the beginning, allowing the receiver clock to estimate additional drift.

- getBiasNanos() gets the clock sub-nanosecond bias and can be used to express time sub-nanosecond precision.

- getTimeOffsetNanos() gets the time offset at which the measurement was taken in nanoseconds. The measurement time can be expressed as the sum of TimeNanos and TimeOffsetNanos.

- getReceivedSvTimeNanos() gets the received GNSS satellite time, at the measurement time, in nanoseconds. It estimates the time of transmission of the pseudorange. This value depends of the sync state that can be achieved, per each satellite. For GPS the sync state starts form C/A Code Lock, then Bit Sync and SubFrame Sync and ends when the time of week (TOW) is fully decoded; for Galileo the sync state starts with E1BC code lock, E1C Code Lock an 2nd Code Lock, Page lock and ends when the Galileo TOW is fully decoded. Transitions form Galileo E1B can E1C channels should be taken into account.

These methods allow to estimate the reception time and the emission time with respect the user time and satellite transmitted time;

combining these times its possible to generate the pseudorange as defined in the RINEX format. The following scheme (Figure 3.3) shall be followed to generate a reliable pseudorange:

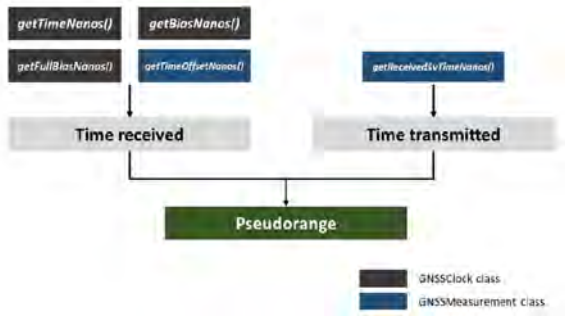

Fig. 3.3: Pseudorange generation from Android methods[9]

The Carrier Phase and Doppler can get immediately using the following methods:

- getAccumulatedDeltaRangeMeters() gets the accumulated delta range since the last channel reset, in meters. This value is directly proportional to the carrier phase and positive value indicates that the SV is moving away from the receiver. Is important to validate this value with the method getAccumulatedDeltaRangeState() that indicates whether getAccumulatedDeltaRangeMeters() has been reset or there is a cycle slip, indicating a loss of lock. The carrier-phase tracked is, as default, L1 (1575.42 MHz).

- getPseudorangeRateMetersPerSecond() gets the pseudorange-rate at the timestamp in m/s. The value is directly proportional to the Doppler Shift, it does not include the receiver and satellite clock frequency errors and a positive 'uncorrected' value indicates that the SV is moving away from the receiver.

Finally with getCn0DbHz() is it possible to get carrier-to-noise density, in dB-Hz, of the tracked signal at the antenna input. Combining these methods, it is possible to create a multiconstellation (GPS, Galileo, GLONASS and Beidou) single frequency (L1) mixed RINEX observable file, version 2.11 or 3.0x.

3.3 The RINEX Format

The first proposal for the Receiver Independent Exchange Format RINEX was developed by the Astronomical Institute of the University of Berne for the easy exchange of the GPS data to be collected during the first large European GPS campaign EUREF 89, which involved more than 60 GPS receivers of 4 different manufacturers. The governing aspect during the development was the following fact: Most geodetic processing software for GPS data use a well-defined set of observables: [11]

- The carrier-phase measurement at one or both carriers (actually being a measurement on the beat frequency between the received carrier of the satellite signal and a receiver-generated reference frequency).

- The pseudorange (code) measurement, equivalent to the difference of the time of reception (expressed in the time frame of the receiver) and the time of transmission (expressed in the time frame of the satellite) of a distinct satellite signal.

- The observation time being the reading of the receiver clock at

the instant of validity of the carrierphase and/or the code measurements.

Usually the software assumes that the observation time is valid for both the phase and the code measurements, and for all satellites observed. Consequently, all these programs do not need most of the information that is usually stored by the receivers: They need phase, code, and time in the above mentioned definitions, and some station-related information like station name, antenna height, etc. Up till now two major format versions have been developed and published:

- The original RINEX Version 1 presented at and accepted by the 5th International Geodetic Symposium on Satellite Positioning in Las Cruces, 1989.
- RINEX Version 2 presented at and accepted by the Second International Symposium of Precise Positioning with the Global Positioning system in Ottawa, 1990, mainly adding the possibility to include tracking data from different satellite systems (GLONASS, SBAS)

Several subversions of RINEX Version 2 have been defined:

- Version 2.10: Among other minor changes allowing for sampling rates other than integer seconds and including raw signal strengths as new observables.
- Version 2.11: Includes the definition of a two-character observation code for L2C pseudoranges and some modifications in the GEO NAV MESS files.

- Version 2.20: Unofficial version used for the exchange of tracking data from spaceborne receivers within the IGS LEO pilot project.

The upcoming European Navigation Satellite System Galileo and the enhanced GPS with new frequencies and observation types, especially the possibility to track frequencies on different channels, ask for a more flexible and more detailed definition of the observation codes. To improve the handling of the data files in case of mixed files, i.e. files containing tracking data of more than one satellite system, each one with different observation types, the record structure of the data record has been modified significantly and, following several requests, the limitation to 80 characters length has been removed. As the changes are quite significant, they lead to a new RINEX Version 3. The new version also includes the unofficial Version 2.20 definitions for spaceborne receivers. The major change asking for a version 3.01 was the requirement to generate consistent phase observations across different tracking modes or channels, to apply $\frac{1}{4}$-cycle shifts prior to RINEX file generation, if necessary, to facilitate the processing of such data. [11] [1]

The RINEX version 3.00 format consists of three ASCII file types:

- 1. Observation data File
- 2. Navigation message File
- 3. Meteorological data File

Each file type consists of a header section and a data section. The header section contains global information for the entire file and is placed at the beginning of the file. The header section contains header

labels in columns 61-80 for each line contained in the header section. These labels are mandatory and must appear exactly as given in these descriptions and examples. The format has been optimized for minimum space requirements independent from the number of different observation types of a specific receiver or satellite system by indicating in the header the types of observations to be stored for this receiver and the satellite systems having been observed. In computer systems allowing variable record lengths the observation records may be kept as short as possible. Trailing blanks can be removed from the records. There is no maximum record length limitation for the observation records. Each Observation file and each Meteorological Data file basically contain the data from one site and one session. Starting with Version 2 RINEX also allows to include observation data from more than one site subsequently occupied by a roving receiver in rapid static or kinematic applications. Although Version 2 and higher allow to insert header records into the data field it is not recommended to concatenate data of more than one receiver (or antenna) into the same file, even if the data do not overlap in time. If data from more than one receiver must be exchanged, it would not be economical to include the identical satellite messages collected by the different receivers several times. Therefore, the navigation message file from one receiver may be exchanged or a composite navigation message file created containing nonredundant information from several receivers in order to make the most complete file. The format of the data records of the RINEX Version 1 navigation message file was identical to the former NGS exchange format. RINEX version 3 navigation message files may contain navigation messages of more than one satellite system (GPS, GLONASS, Galileo, SBAS). It is useful to remember the

basic definitions needed to read the format: GPS observables include three fundamental quantities that need to be defined: Time, Phase, and Range.

The time of the measurement is the receiver time of the received signals. It is identical for the phase and range measurements and is identical for all satellites observed at that epoch. For single-system data files it is by default expressed in the time system of the respective satellite system. Else the actual time can (for mixed files must) be indicated in the Start Time header record. [11]

The pseudo-range (PR) is the distance from the receiver antenna to the satellite antenna including receiver and satellite clock offsets (and other biases, such as atmospheric delays):

$$PR = d + c \cdot (\delta_R - \delta^S + \lambda) \tag{3.1}$$

so that the pseudo-range[1] reflects the actual behavior of the receiver and satellite clocks. The pseudo-range is stored in units of meters.

The phase is the carrier-phase measured in whole cycles. The half-cycles measured by squaring-type receivers must be converted to whole cycles and flagged by the respective observation code. The phase changes in the same sense as the range (negative doppler). The phase observations between epochs must be connected by including the integer number of cycles. The observables are not corrected for external effects like atmospheric refraction, satellite clock offsets, etc. If necessary, phase observations are corrected for phase shifts needed to guarantee consistency between phases of the same frequency and satellite

[1] where d = distance, δ_R = receiver clock offset, δ^S = satellite clock offset, λ = other biases

system based on different signal channels.

The image shows an example of a RINEX format obtained by a measurement obtained in February:

```
> 2018  2 27 14  5 24.9997983  0 17
G10  24371161.396       52742.1621        -451.619          20.277
G12  21510311.221     -245177.2890        1939.141          29.514
G13  22360454.280      484280.5840       -3785.196          30.064
G15  21248975.840      328108.2430       -2580.465          35.772
G17  22382695.282      196970.5380       -1612.154          28.974
G19  21386985.598      -13867.3940          68.752          32.859
G24  20656245.978     -102650.2790         786.648          31.721
R26  21557188.869     -320527.6010        2528.184          29.797
R09  21870830.539      382181.2771       -3051.060          23.389
R11  21967843.079     -244081.5160        1943.927          27.905
R01  21239897.825      -46363.1740         321.967          30.854
R19  21824615.133      156601.0390       -1278.253          26.718
R07  23580129.316      614095.1330       -4984.951          29.932
R08  20885918.779      287142.1140       -3147.766          26.952
C11  23499397.606      -40843.5130         395.468          28.360
C20  26311381.310     -102572.0070        1098.284          28.171
E19  26487516.574     -134912.6371        1098.438          25.370
> 2018  2 27 14  5 25.9997983  0 17
G10  24371250.135       53192.6841        -454.481          20.565
G12  21509949.672     -247115.9440        1938.359          28.904
G13  22361180.077      488065.6390       -3785.316          30.358
G15  21249466.600      330688.6580       -2580.556          35.747
G17  22383021.157      198582.8920       -1612.471          29.073
G19  21386980.501      -13936.2940          68.450          32.755
G24  20656096.682     -103436.9020         786.612          31.070
R26  21556703.804     -323055.6190        2527.912          29.776
R09  21871428.925      385229.1031       -3047.389          23.299
R11  21967466.240     -246025.8150        1944.144          27.648
R01  21239853.156      -46685.2020         321.838          30.857
R19  21824821.690      157879.4710       -1278.671          26.800
R07  23581060.771      619079.8690       -4984.697          30.016
R08  20886480.890      290289.9400       -3147.922          26.349
C11  23499318.161      -41238.9750         394.917          28.143
C20  26311180.149     -103670.7050        1098.338          28.485
E19  26487314.514     -136011.4421        1099.911          24.832
```

Fig. 3.4: RINEX Example obtained in February

This useful format is the key to postprocessing a receiver's data and producing more accurate data, adding information such as weather conditions models at the time of measurement. For this reason, the next phase of the analysis of the measurements focused on the search for an instrument able to read the measurements of the device and create a file with RINEX format for the subsequent postprocesses. [11] [1]

Chapter 4

Measurements Processing

In this chapter, the process of analysis and measurements will be described in its entirety. The first studies and the first analyzes began in the second half of November, when the approach and the tools to be analyzed were decided in a similar trial and error process. For this reason all the processes will be described, from the exploratory to the final ones.

4.1 Preliminary Tests with GNSS Raw Analysis Software for Android Developers

The first studies on raw measurements were performed using the Android developer software: GNSS Raw Analysis Software. The software package includes basic desktop software and a smartphone application: The GNSS Logger APK [1]. To get GNSS output with the

[1] Android Package

sample app, the device must support raw GNSS measurements. All the live tests carried out in this work were performed using a Samsung Galaxy S8 [13] device. However, not all android devices are in fact compatible with this type of software and analysis. What allows compatibility with this innovative feature, as well as the characteristics of the hardware, is also the operating system. In Fig.4.1 is possible to observe the type of compatible devices, with consideration also of the operating system: [12]

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
Huawei Mate 10	8.0	no	yes	yes	yes	GPS GLONASS
Huawei Mate 10 Pro	8.0	no	yes	yes	yes	GPS GLONASS QZSS
Sony Xperia XZ1	8.0	no	no	no	yes	GPS GLONASS GALILEO BeiDou
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

Fig. 4.1: Android Features Pre-A8 Update - November 2017 [12]

It is visible that the Samsung S8 shows a great transparency in being able to analyze also the navigational message, and trace almost the whole of the GNSS constellations. In particular, the table distinguishes the version of the device: The Exynos. For this name, all the devices produced for the market in Europe, the Middle East and Africa are identified (EMEA). The device used, is included in the list of the following models: From G950F to G955F. Once the actual compatibility

was checked, the first analysis could be carried out.

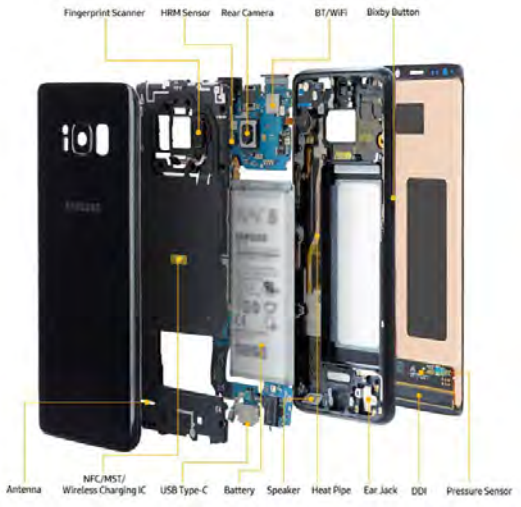

Fig. 4.2: Samsung Galaxy S8 Scheme [13]

The process consists of these steps: The first step allows to record data using the APK, the process has the duration of the time of the measurement itself. The second step is to download the file containing the measurements obtained from the device, directly on the desktop, opening it with the GNSS Raw Analysis software. The software allows to analyze the measurements made, thanks to the support of the Matlab environment. This allows to analyze certain and pre-established characteristics of the measurements, such as:

- The Skyplot: Graph showing the position of the satellites during the observation period.
- The signal quality of the satellites.
- For each satellite, the time plot of carrier to noise density (C/No).

- The pseudoranges.

- The weighted least squares position results obtained from the raw pseudoranges. The weighting is done using the reported uncertainty of each measurement, which is part of the raw measurement API spec.

- The errors of each pseudorange for each measurement.

- The errors of each pseudorange rate for each measurement.

Various tests were carried out during the first winter months and this approach made it possible to understand the preliminary quality and potential of the device. In fact, the S8 shows an ability to receive the signal from numerous satellites, in particular the GLONASS and GPS constellations. In these early tests, observations were made in various configurations and conditions, such as the influence of the antenna for the data network with the quality of position measurements and background applications.These tests showed a slight improvement in measurements with the active data network, but this fact needs further study in the future to confirm this anlysis. The tool offered by Android developers [12] therefore, is useful for preliminary analysis, however, being a beta software, it is not suitable for more advanced analysis. For this purpose, it was necessary to obtain a format[1] that would allow the study with a greater level of detail than the Android analysis software.

[1]Chapter 3 - RINEX Format

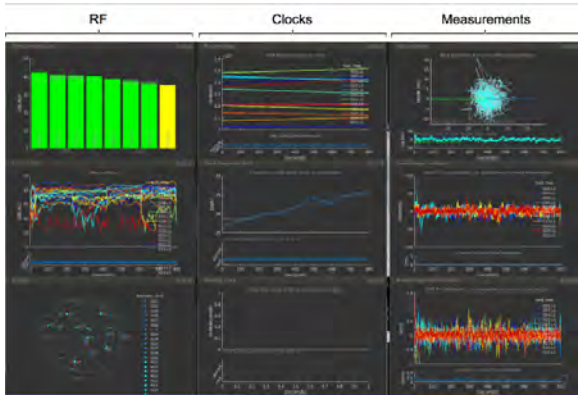

Fig. 4.3: GNSS Raw Analysis Software Interface [12]

4.1.1 From the GNSS Raw Analysis Software for Android Developers to the RINEX format

The next step for the analysis of the measurements of the device in question, was to order and make the file of the observations clear and uniform, converting it, to as much as possible, in a RINEX format. The file and, therefore, the output of the APK GNSS Logger is a format readable only by the software for Android developers. Opening it and analysing it with any text editor is difficult if not for the original destination software.From the image it is possible to observe the GNSS Logger output file opened by a reading software:

It is noted that the file is difficult to read immediately and for this reason it was decided to use the Matlab software, for format conversion.

The script created on Matlab within a PhD reaserch of the Geodesy and Gematics Division of Sapienza [9] briefly transforms and reorders the data contained in the original file in a RINEX format. The tests carried out in the months from November to January, followed this

Fig. 4.4: Preliminaty Output [12]

procedure. Once the device registered, the output file was opened to Matlab and converted into a RINEX format.

The script created in Matlab, however, could not be considered complete: since from the original file, it extrapolated and rearranged the information related to the GPS constellation only. This obviously limited the analysis of the measurements to the American constellation alone, leaving out all the others. However, we proceeded to the analysis of the measurements considering only the GPS system, looking for a solution or tool able to analyse all the measurements and converting them into a complete RINEX 3.0 format. The RINEX format is observed in Fig.4.5

4.1.2 Post - Processing GNSS data with RTKLIB

Once the RINEX format is obtained, it is possible to start post processing analyses, such as: Single Point Positioning, Relative Positioning, Kinematic etc. For this reason, a software suitable for this type of analysis is necessary. RTKLib is an open source program pack-

```
     3.03           OBSERVATION DATA    M: Mixed            RINEX VERSION / TYPE
Geo++ RINEX Logger  Geo++                20180227 140506 UTC PGM / RUN BY / DATE
************************************************************COMMENT
This file was generated by the Geo++ RINEX Logger App       COMMENT
for Android devices (Version 1.2.3). If you encounter       COMMENT
any issues, please send an email to android@geopp.de        COMMENT
Note: Tracking signals and phase shifts by best knowledge   COMMENT
************************************************************COMMENT
Geo++                                                       MARKER NAME
GEODETIC                                                    MARKER TYPE
RINEX Logger user   Geo++                                   OBSERVER / AGENCY
ce041714f39d4a1702  samsung             SM-G950F           REC # / TYPE / VERS
ce041714f39d4a1702  SM-G950F                               ANT # / TYPE
  4635407.4452  4158179.2825  1379894.4913                 APPROX POSITION XYZ
        0.0000        0.0000        0.0000                 ANTENNA: DELTA H/E/N
G    4 C1C L1C D1C S1C                                      SYS / # / OBS TYPES
R    4 C1C L1C D1C S1C                                      SYS / # / OBS TYPES
E    4 C1B L1B D1B S1B                                      SYS / # / OBS TYPES
C    4 C2I L2I D2I S2I                                      SYS / # / OBS TYPES
J    4 C1C L1C D1C S1C                                      SYS / # / OBS TYPES
  2018     2    27    14     5   24.9997983     GPS         TIME OF FIRST OBS
G                                                           SYS / PHASE SHIFTS
R                                                           SYS / PHASE SHIFTS
E                                                           SYS / PHASE SHIFTS
C                                                           SYS / PHASE SHIFTS
J                                                           SYS / PHASE SHIFTS
                                                            END OF HEADER
```

Fig. 4.5: RINEX Formar Header

age for GNSS positioning. The project is packaged developed and maintained by Tomoji Takasu from the Tokyo University of Marine Science and Technology in Japan. It supports standard and precise positioning algorithms with: GPS, GLONASS, Galileo, QZSS, BeiDou and SBA[14].

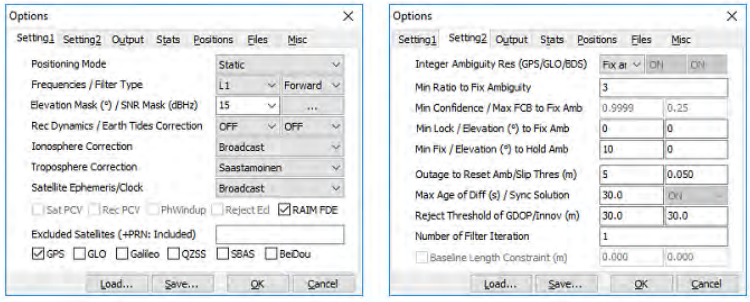

Fig. 4.6: RTK U.I. [14]

4.2 The Single Point Positioning

The basic concept of point position depends on a least square adjustments performed on at least four code observations. The code observation equation[1] is [6]:

$$P_R^S(t) = \rho_R^S(t) + c(dt_R(t)) + I_R^S(t) + T_R^S(t) \tag{4.1}$$

The unknowns in the above equation are X, Y, Z and the clock error $\Delta\delta$. The distance[2] between satellite and receiver is [6]:

$$\rho_R^S = \sqrt{(X_R - X^S)^2 + (Y_R - Y^S)^2 + (Z_R - Z^S)^2} \tag{4.2}$$

4.2.1 Preliminary Tests

Measurements of single point positioning carried out in the last months of 2017 are now shown. These tests were performed with the Matlab script and therefore the results only consider the GPS constellation. The first test was carried out on November 30 afternoon, at one of the internal courtyards of the Faculty of Engineering. These tests were performed with recordings of about five minutes[3]. The observations were then processed using the RTKpost software in single point positioning. It is important to underline that in order to compute the orbits and clocks of the satellites (known terms of the code equation 4.1) it is necessary to know the navigational message of the

[1]Where ρ is the distance between satellite and receiver, c is the speed of light in the vacuum, I is the Ionospheric Delay, T is the Tropospheric Delay

[2]Where R are the receiver cartesian coordinates, S are the satellite cartesian coordinates

[3]About 300 epochs

same. At present, no Android application is capable of obtaining the navigational files, for this reason all the processing in this work the navigational files have been taken from permanent stations in the area. It is observed that the positions (one per epoch) of the S8 device is not precise and has some points that are far from the actual position of several tens of meters as in Fig.4.7. Remember that in these tests only the constellation of American satellites is considered.

Fig. 4.7: Single Point Test - 30^th November

Thanks to the Google Earth software[19] it was possible postprocessing the measurements obtained from the device, then transformed (through the steps shown in the previous paragraphs) into RINEX format and converted through RTKPost into KML format, which is an XML notation for expressing geographic annotation and visualization within Internet-based, two-dimensional maps and three-dimensional Earth browsers. Considering that at the time of the measurements,

the device was fixed, it is deduced that the position has numerous inaccuracies.

On the same day, another test was also performed. This time the device was in motion, following the pedestrian path University Metro station, near the Colosseum. In this test, at first, it was decided to analyse it as the previous example and then, after obtaining the RINEX file, converted to KML. However, the path was also measured with another android application: GPS NMEA. This application records an internal GPS solution in NMEA format. NMEA 0183 is a combined electrical and data specification for communication between marine electronics such as echo sounder, sonars, anemometer, gyrocompass, autopilot, GPS receivers and many other types of instruments. It has been defined by, and is controlled by, the National Marine Electronics Association (US). It is observed in Fig.4.8 that the tracing obtained from the NMEA application is certainly more precise than the single-point measurement, however, the second one is the most accurate even if less precise, as it is more faithful to the path actually followed. The internal solutions, in fact, more precise and less accurate, is little faithful to the real path. The figure shows in blue the path traced by GPS NMEA and in red the one by GNSS Logger postprocessed.

To continue with the analysis of the measurements, we searched for a logger able to effectively record all the constellations of positioning, as listed in the compatibility chart of the smartphone written by the Android developers. This, with the purpose of comparing the following studies, the precision and therefore the difference between the different constellations. Although the device, in the previous studies, has recorded for every period, more than four satellites (the minimum), for an accurate study it is necessary to consider all the constellations.

Fig. 4.8: Single Epoch vs. NMEA - 30^th November [19]

This was possible thanks to a new logger.

4.2.2 The Geo++ RINEX Logger

This logger, in addition to being the first to record measurements directly in RINEX format, supports all the constellations that support the same S8 smartphone: GPS, GLONASS, GALILEO and BDS. This logger, as a first feature, distinguishes the satellites in their constellations. Once the application is open, it begins to search for visible satellites. Of all these, consider those traceable and consequently those synchronized, that is, consider the measurements with the best quality. The RINEX format is 3.03, which supports BDS and QZSS. Once tested and finally verified the registration of all the synchronized constellations, we can proceed with the most advanced and post processed studies with RTKLib.

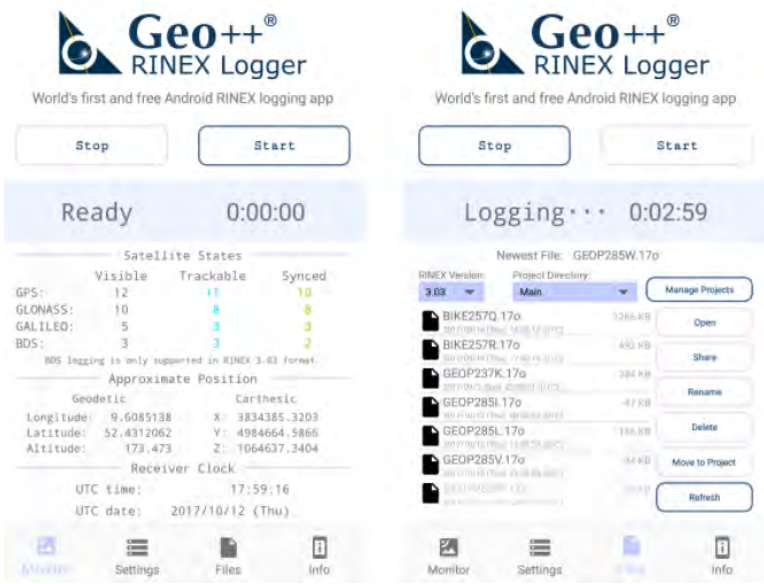

Fig. 4.9: Geo++ Logger User Interface [16]

4.2.3 Tests with Geo++

The tests were carried out on February 27th 2018, two S8 devices were used on the roof of the hydraulics and geodesy building of the Sapienza Engineering Faculty. In the first measurement, the devices were positioned at exactly 2 meters. The acquisition with Geo ++ Logger was carried out at the same time and same duration for the two smartphones (about ten minutes).

The first registration was therefore, made following this configuration as in Fig.4.10. The second and third, however, the two devices are placed at 2.15 m. as in Fig.4.11

Also for these tests, at first, a single point positioning approach through RTKpost was carried out. Therefore for each observation

Fig. 4.10: 1^{st} Configuration [19]

Fig. 4.11: 2^{nd} Configuration [19]

epoch, for each device and for each recording it was possible to estimate the position of the device in terms of X,Y,Z Geocentric Cartesian Coordinates. The following data was calculated with Excel, for each recording and for each device the data were processed considering different combinations of satellites[1]:

[1]Conf.1 = GPS, Conf.2 = GPS + Galileo, Conf.3 = GPS + Galileo + GLONASS + BEIDOU

- 1 GPS
- 2 GPS + Galileo
- 3 GPS + Galileo + GLONASS + BEIDOU

Obtained all these configurations, the following outputs have been calculated with excel:

- 1 - The Mean of the positions X, Y, Z;
- 2 - The Mean of the observed satellites;
- 3 - The standard deviation of the positions X, Y, Z;

For convenience in showing the results, the recordings are identified in T1, T2, T3 and the receivers in R1, R2. The data obtained are interesting and show that even if the device at that time recorded and synchronized a single Galileo satellite, however, from the elaboration made it possible to understand that this satellite does not seem to have influenced the position of the device. The real difference in results is obtained by considering all the constellations captured by Android.

These data show a clear difference regarding the quality of the signal acquired by the two devices. The standard deviations show how the second device, in all three recordings, has received fewer satellites and therefore a consequent worsening of the precision that is instead greater for the first device. It should be added that the configuration of the two devices is different: The first device has the antenna of the active data network, while the second is not. This may have influenced the result.

After this first simple processing, which showed a general idea of how the devices recorded, it is necessary to find another, more powerful tool to perform calculations. For this reason, it is necessary to create a simple script that can analyse the data in a short time, bypassing the slow and cumbersome process of the spreadsheet. Later it will illustrate the main functions used and some features of the same script.

Fig. 4.12: First Configuration - Devices Positions

Tab. 4.1: Measurements Analyzed with Excel on February 27

T/R	Conf.	MEAN-X(m)	MEAN-Y(m)	MEAN-Z(m)	DEV.ST-X(m)	DEV.ST-Y(m)	DEV.ST-Z(m)	MEAN-N Sat.
T1/R1	1	4642364.16	1028712.73	4236881.07	40.81	12.57	25.69	5.7
T1/R1	2	4642364.16	1028712.73	4236881.07	40.81	12.57	25.69	5.7
T1/R1	3	4642358.64	1028710.60	4236881.23	14.38	6.76	9.97	13.3
T2/R1	1	4642360.96	1028714.93	4236879.54	36.48	13.45	22.57	6.0
T2/R1	2	4642360.96	1028714.93	4236879.54	36.48	13.45	22.57	6.7
T2/R1	3	4642357.04	1028712.94	4236877.83	31.77	12.41	22.89	13.6
T3/R1	1	4642360.57	1028717.28	4236882.74	28.25	17.35	17.13	6.0
T3/R1	2	4642360.57	1028717.28	4236882.74	28.25	17.35	17.13	6.3
T3/R1	3	4642360.57	1028717.28	4236882.74	28.25	17.35	17.13	7.3
T1/R2	1	4642323.00	1028702.58	4236860.33	136.09	29.53	70.98	4.6
T1/R2	2	4642323.00	1028702.58	4236860.33	136.09	29.53	70.98	4.6
T1/R2	3	4642365.57	1028711.10	4236882.35	59.19	14.62	33.30	10.0
T2/R2	1	4642366.49	1028720.77	4236884.92	32.05	14.20	24.28	6.0
T2/R2	2	4642366.49	1028720.77	4236884.92	32.05	14.20	24.28	6.0
T2/R2	3	4642368.09	1028721.32	4236886.23	14.47	6.52	15.67	13.0
T3/R2	1	4642381.63	1028725.46	4236892.15	48.72	20.31	26.88	6.0
T3/R2	2	4642381.63	1028725.46	4236892.15	48.72	20.31	26.88	6.5
T3/R2	3	4642381.63	1028725.46	4236892.15	48.72	20.31	26.88	7.5

4.2.3.1 Data Analysis with Python

Python is an interpreted high-level programming language for general purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. The script created analyses the files initially processed with RTKPost, which, contains the solutions for each epoch. First, the script calculates the means of X, Y, Z. At the same time the mean of the synchronized satellites used in the processing, and then the standard deviation of the X, Y, Z is calculated. Practically, the script performs the calculations of the previous paragraph in less than two seconds, greatly simplifying the spreadsheet process, drastically reducing the calculation time. The results expressed are in X, Y, Z or Geocentric Cartesian Coordinates. It is possible thanks to support functions, to calculate also the above-mentioned outputs, in Geodetical Coordinates. [17]

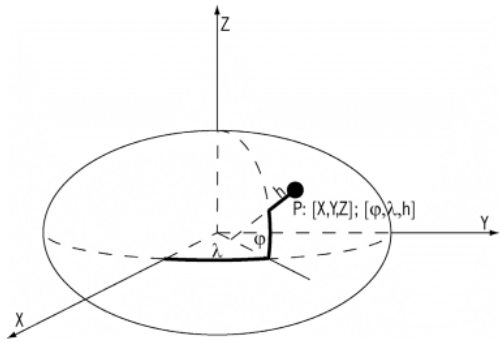

Fig. 4.13: Geodedic Coordinates of a point P [6]

Where λ is the geodetic longitude, ϕ is the geodetic latitude and

h ellipsoidal height. The relationship between Cartesian and geodesic coordinates of P is given by:

$$X_P = (N + h_p)cos\varphi_P cos\lambda_P \tag{4.3}$$

$$Y_P = (N + h_p)cos\varphi_P sin\lambda_P \tag{4.4}$$

$$Z_P = [N(1 - e^2) + h_P]sin\varphi_P \tag{4.5}$$

$$N = \frac{a}{\sqrt{(1 - e^2 sin^2\varphi_P)}} \tag{4.6}$$

Where N is the East-West curvature radius. At first, some auxiliary quantities are computed:

$$e_b^2 = \frac{a^2 - b^2}{b^2}, \rho = \sqrt{X^2 + Y^2}, \Psi = arctan(\frac{Z}{\rho\sqrt{1 - e^2}} \tag{4.7}$$

Then the Geodedic coordinates are given by:

$$\lambda = arctan(\frac{Y}{X}), \rho = arctan(\frac{(Z + e_b^2 bsin(\Psi)^3)}{(\rho - e^2 acos(\Psi)^3)}) \tag{4.8}$$

$$N = \frac{a}{\sqrt{(1 - e^2 sin^2\varphi_P)}}, h = \frac{\rho}{cos(\varphi)} - N \tag{4.9}$$

The process is generally iterative and can be formulated with Python language. In a similar way, through a support function it is possible to express the results obtained in a geocentric Cartesian system in a local

coordinate system Est, Nord, Up. In other words, you can define a system oriented with respect to the horizontal plane, with orthogonal axes East, North and Up.

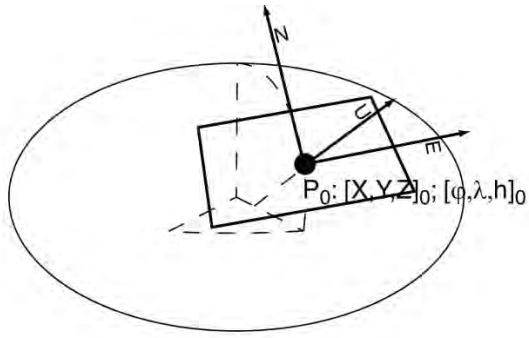

Fig. 4.14: System oriented with orthogonal axes, East, Nord, Up [6]

Given the geocentric baselines from P_0 to another point P:

$$\Delta x_{P0,P} = \begin{vmatrix} X_P - X_0 \\ Y_P - Y_0 \\ Z_P - Z_0 \end{vmatrix} \tag{4.10}$$

The local coordinates (respect a local sustem with origin in 0) of P are given by:

$$\begin{vmatrix} E \\ N \\ U \end{vmatrix} = R_0 \begin{vmatrix} X_P - X_0 \\ Y_P - Y_0 \\ Z_P - Z_0 \end{vmatrix}, R_0 = \begin{vmatrix} -sin\lambda_0 & cos\lambda_0 & 0 \\ -sin\varphi_0 cos\lambda_0 & -sin\varphi_0 sin\lambda_0 & cos\varphi_0 \\ cos\varphi_0 cos\lambda_0 & cos\varphi_0 sin\lambda_0 & sin\varphi_0 \end{vmatrix} \tag{4.11}$$

Performed these transformations, useful for computational and interpretative convenience, it is possible through the Matplot library, to

graph all the data useful for the analysis. At this moment, all the analyses will be carried out and processed with Python.

4.2.3.2 The Single Point Positioning - Results analysis

The difference between Excel and Python appears to be marked only in the execution time of the calculations. In Tab.4.2, the results in E, N, Up and the respective position in Latitude, Longidudine and height of the final solutions are shown[1].

[1]Same Configurations, See Paragraph 4.2.3 Conf.1 = GPS, Conf.2 = GPS + Galileo, Conf.3 = GPS + Galileo + GLONASS + BEIDOU

Tab. 4.2: Measurements Analyzed with Python on February 27

T/R	Conf.	LAT.	LONG.	Height (m)	DEV.ST-E (m)	DEV.ST-N (m)	DEV.ST-U (m)
T1/R1	1	41.8935860	12.4944164	102.25	6.84	14.94	46.95
T1/R1	2	41.8935860	12.4944164	102.25	6.84	14.94	46.95
T1/R1	3	41.8936223	12.4944058	98.00	6.82	7.68	14.98
T2/R1	1	41.8935917	12.4944507	99.26	7.39	11.76	42.67
T2/R1	2	41.8935917	12.4944507	99.26	7.39	11.76	42.67
T2/R1	3	41.8936059	12.4944375	94.94	6.31	8.59	39.33
T3/R1	1	41.8936124	12.4944794	101.49	11.77	10.71	33.71
T3/R1	2	41.8936124	12.4944794	101.49	11.77	10.71	33.71
T3/R1	3	41.8936124	12.4944794	101.49	11.77	10.71	33.71
T1/R2	1	41.8937019	12.4944044	56.85	6.25	41.60	150.33
T1/R2	2	41.8937019	12.4944044	56.85	6.25	41.60	150.33
T1/R2	3	41.8935884	12.4943936	103.86	6.80	18.73	65.95
T2/R2	1	41.8935877	12.4945050	107.81	8.15	9.07	40.79
T2/R2	2	41.8935877	12.4945050	107.81	8.15	9.07	40.79
T2/R2	3	41.8935864	12.4945073	109.93	4.69	6.56	20.69
T3/R2	1	41.8935412	12.4945207	124.39	10.60	17.61	55.45
T3/R2	2	41.8935412	12.4945207	124.39	10.60	17.61	55.45
T3/R2	3	41.8935412	12.4945207	124.39	10.60	17.61	55.45

Once the data were computed in numerical form, Considering the East, North, Up, positions it is now possible to observe the trend of the acquisition for each single epoch. Another interesting thing is to observe how the device synchronized the satellites for each epoch and then display the average of the recording. Graphic processing was performed for all devices, all recordings and every single constellation combination considered. However, considering that the measurements obtained with GPS and subsequently in GPS + Galileo, show no differences, the following graphs will be shown considering only the American Constellation. In the following graphs: Fig.4.15,4.16,4.17,4.18 showing for each period, the average of the acquired satellites and positions, the second recording (T2) will be shown for each S8 device. What results from these observations is that the quality of the measurements is worse in the Up and this is easily visible from the position graphs 4.15 and in the Tab.4.2, while the acquired GPS satellites are stable throughout the acquisition as in Fig.4.16,4.18. Summing up, it can be defined, that the Up varies with maxima that reach also the 120m, the East 30m and the North 50m.

The interesting thing that has been noticed in the various configurations of the constellations, is that, in the configuration where the totality of the GNSS constellations are considered, the RTKPost software, returns the file with fewer epochs. This is due to a problem in the RTKPost software, which in erasing the .pos file (which will then be analyzed in Python) erases many observation epochs. The cause seems to be the Russian constellation GLONASS.

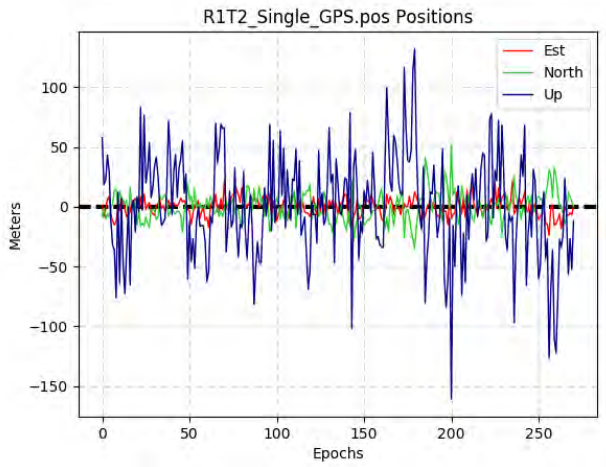

Fig. 4.15: R1/T2 Single Positioning - GPS

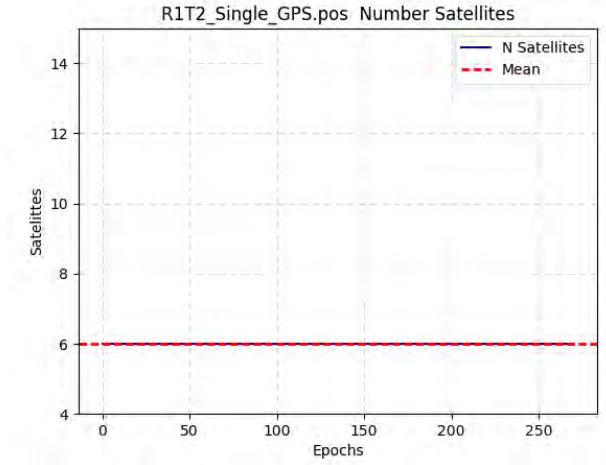

Fig. 4.16: R1/T2 Single Positioning - Mean Satellites

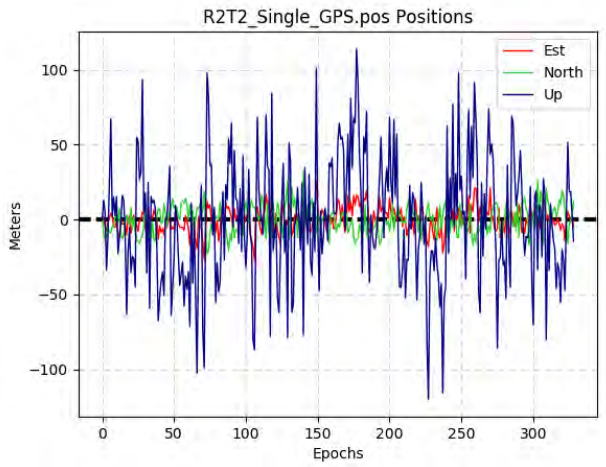

Fig. 4.17: R2/T2 Single Positioning - GPS

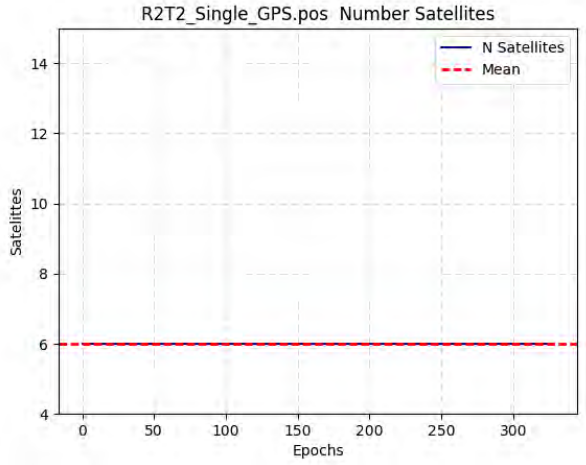

Fig. 4.18: R2/T2 Single Positioning - Mean Satellites

4.3 The Relative Positioning

Once the data were processed in single positioning and the first approximations of positioning of the two devices were drawn up, the analysis was carried out in the case of relative positioning. Two receivers make simultaneous observations of the same satellites. The observations are conveniently combined and processed to estimate the base (3D vector) between the two receivers. The accuracy can depend on: the type of receivers (type of observable acquirable); from the distance between the receivers (from <10 Km to > 500 Km); from the survey method (duration of the stay on the points); and from the data processing approach (real time, post processing). Speaking of precision, it goes from: 1-2 meters (relative on the codes in real time): precision navigation, up to the best of the centimeter (double frequency, static prolonged). The method requires the simultaneous use of at least two receivers; one of these must be on a point (reference station) of coordinates known a priori; all this makes the survey methodology less complex than traditional topographic techniques, however, data processing can be complex and burdensome [6]. In Fig.4.19 the relative positioning is summarized.

The relative position, therefore, is carried out using combinations of observations, said double differences involving the two receivers. Before describing double differences, the concept of single difference is now shown.

Consider two R1 and R2 receivers, which have made observations of a satellite S in the same epoch t. The single difference is the difference between the observations of the receiver forward and of the back. The single difference is essentially a new observation constructed by

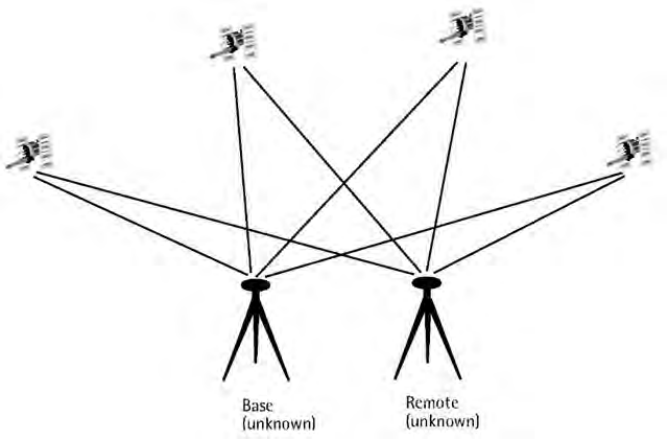

Fig. 4.19: The Relative Positioning [8]

two originals, where the loss of noise is offset by some advantages: in the single difference the clock error of the satellite no longer appears, contributing to the error balance; In addition, the differentiation significantly reduces the effects of tropospheric disturbance and ephemeris errors and attenuates ionospheric error. Finally, in the case of the phases, the initial fractional phase of the satellite disappears. Described the functioning of the single difference, the concept of double difference can now be defined.

The double differences are the difference of two simultaneous single differences referring to the same pair of receivers and to two distinct satellites S1 and S2. The use of single or double differences therefore does not allow absolute positioning but only the relative one; further, the electronic noise of observation propagates, increasing and correlating, in the differentiation of the observations. On the other hand, the advantage offered by the double differences is such that in the last thirty years they have become the fundamental observable for relative

positioning: first the receiver's clock incognites are eliminated, but this is not the main point because for this purpose observations are also sacrificed. The major good is the complete elimination of the initial fractional phases: in fact, now only the entire term of ambiguity remains, to which the so-called estimation and fixing algorithms can be applied. In Fig.4.20 double difference is shown.

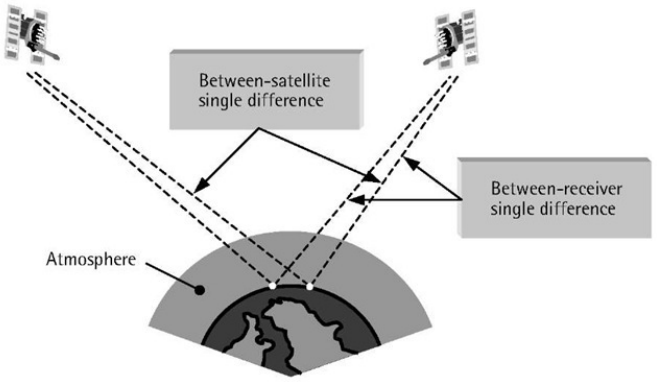

Fig. 4.20: Double Difference Scheme [8]

It is clear that for the relative positioning analysis, it is important to consider the ambiguities. RTKPost uses a complicated mathematical formula or algorithm to calculate the exact number of radio wavelengths between the satellites and the base station antenna, a process known as ambiguity resolution, and yield either a fixed or float solution. In a fixed solution, the number of wavelengths is a integer number, and the algorithm is constrained to yield a integer number. In a float solution, the algorithm does not yield an acceptable fixed solution, so the ambiguity is allowed to be a decimal or floating-point number.

For these tests, two S8 devices were used fixed at a known distance,

in three different recordings. In a first case, the positions of the single devices were analysed with respect to the Permanent Station M0SE[1] , then the position of the device was analysed considering an S8 as Rover and the other as Permanent Station. The pre-set positions are the same as those used in the single positioning study of the previous paragraph.

For relative positioning, in addition to the RINEX file of the device and of the Navigational File, it is necessary the RINEX file of the Permanent Station, in this case that of the M0se. In the post processing phase, therefore, the analysis was carried out by setting the Static condition and considering both cases of ambiguity fixing: both fix and float. In a first phase of processing, the solution output was compared both by making the single solution elaborate by RTK and by processing different solutions from time to time. The single solution chosen by the RTKPost software seems to be different from the average of the total ones. The software, in the case of a single solution, has shown to exclude some of the recording epochs, compromising the final position, so it was decided to consider all the solutions for each epoch and then elaborate the final position in another computation, considering the average of these. In the following tables 4.3,4.4 is shown the difference between the calculations with the average of all solutions and of the single one. Considering also the conversion into Geodetic coordinates, Tab.4.5,4.5.

[1]Permanent Station of Sapienza University, `http://www.epncb.oma.be/_networkdata/siteinfo4onestation.php?station=m0se`

Tab. 4.3: Static Positioning - All Solutions - February 27

R/T	AMB.	X - ecef (m)	Y - ecef (m)	Z - ecef (m)	sdx (m)	sdy (m)	sdz (m)
R1/T1	FIX	4642368.37	1028713.38	4236884.62	6.47	1.22	2.92
R1/T1	FLOAT	4642368.54	1028713.43	4236884.86	0.48	0.08	0.06
R2/T1	FIX	4642369.90	1028712.21	4236886.33	15.00	3.61	7.43
R2/T1	FLOAT	4642371.56	1028712.58	4236886.95	10.77	2.41	4.24
R1/T2	FIX	4642369.78	1028717.39	4236886.25	1.16	0.60	0.78
R1/T2	FLOAT	4642369.87	1028717.45	4236886.33	0.27	0.09	0.21
R2/T2	FIX	4642370.20	1028720.98	4236887.29	1.06	0.54	0.98
R2/T2	FLOAT	4642370.03	1028721.06	4236887.44	0.54	0.24	0.43
R1/T3	FIX	4642365.79	1028718.92	4236886.31	2.34	1.49	1.35
R1/T3	FLOAT	4642366.03	1028719.00	4236886.43	0.26	0.13	0.14
R2/T3	FIX	4642379.51	1028723.70	4236892.14	5.26	1.93	2.61
R2/T3	FLOAT	4642379.47	1028723.61	4236892.09	2.14	0.70	1.09

Tab. 4.4: Static Positioning - Single Solutions - February 27

R/T	AMB.	X - ecef (m)	Y - ecef (m)	Z - ecef (m)	sdx (m)	sdy (m)	sdz (m)
R1/T1	FIX	4642319.40	1028703.08	4236856.96	0.01	0.00	0.01
R1/T1	FLOAT	4642369.73	1028713.54	4236885.02	0.08	0.04	0.05
R2/T1	FIX	4642374.40	1028713.54	4236887.82	0.03	0.01	0.01
R2/T1	FLOAT	4642374.49	1028713.53	4236887.89	0.07	0.03	0.07
R1/T2	FIX	4642370.10	1028717.53	4236886.43	0.03	0.01	0.02
R1/T2	FLOAT	4642370.13	1028717.51	4236886.48	0.08	0.04	0.05
R2/T2	FIX	4642370.02	1028720.88	4236886.99	0.03	0.01	0.02
R2/T2	FLOAT	4642369.68	1028720.89	4236887.19	0.10	0.05	0.07
R1/T3	FIX	4642366.68	1028719.11	4236886.98	0.01	0.01	0.01
R1/T3	FLOAT	4642366.43	1028719.20	4236886.66	0.10	0.06	0.07
R2/T3	FIX	4642376.77	1028722.89	4236890.66	0.03	0.02	0.02
R2/T3	FLOAT	4642376.58	1028722.77	4236890.61	0.14	0.08	0.09

Tab. 4.5: Static Positioning - All Solutions/Geodedical Coordinates February 27

ROVER	AMBIGUITIES	LATITUDE	LONGITUDE	HEIGHT (m)	REG. N
R1	FIX	41.8935843	12.4944132	107.78	1
R1	FLOAT	41.8935848	12.4944133	108.07	1
R2	FIX	41.8935883	12.4943954	109.85	1
R2	FLOAT	41.8935701	12.4943989	114.44	1
R1	FIX	41.8935808	12.4944574	110.92	2
R1	FLOAT	41.8935816	12.4944572	110.67	2
R2	FIX	41.8935815	12.4944978	112.12	2
R2	FLOAT	41.8935834	12.4944991	112.11	2
R1	FIX	41.8936035	12.4944851	107.93	3
R1	FLOAT	41.8936028	12.4944854	108.19	3
R2	FIX	41.8935559	12.4945055	122.57	3
R2	FLOAT	41.8935559	12.4945045	122.49	3

Tab. 4.6: Static Positioning - Single Solution/Geodedical Coordinates February 27

ROVER	AMBIGUITIES	LATITUDE	LONGITUDE	HEIGHT (m)	REG. N
R1	FIX	41.8936997	12.4944196	52.07	1
R1	FLOAT	41.8935788	12.4944115	109.06	1
R2	FIX	41.8935701	12.4943994	114.33	1
R2	FLOAT	41.8935701	12.4943989	114.44	1
R1	FIX	41.8935808	12.4944574	110.92	2
R1	FLOAT	41.8935810	12.4944571	110.97	2
R2	FIX	41.8935807	12.4944970	111.78	2
R2	FLOAT	41.8935840	12.4944981	111.66	2
R1	FIX	41.8936026	12.4944850	109.05	3
R1	FLOAT	41.8936018	12.4944867	108.67	3
R2	FIX	41.8935630	12.4945032	119.45	3
R2	FLOAT	41.8935640	12.4945022	119.26	3

To obtain a numerical report, the table in which the distances between the two receivers are calculated is now shown, considering the results in both cases as in Tab.4.7,4.8.

Tab. 4.7: Static Positioning - All Solutions (Mean): Distances between receivers

ROVER	AMBIGUITES	REG. N	DIST. REF. (m)	DIST. (m) REC.	Δ (m)
1 & 2	FIX	1	2	2.57	0.57
1 & 2	FLOAT	1	2	3.77	1.77
1 & 2	FIX	2	2.15	5.18	3.03
1 & 2	FLOAT	2	2.15	5.19	3.04
1 & 2	FIX	3	2.15	3.75	1.60
1 & 2	FLOAT	3	2.15	3.77	1.62

Tab. 4.8: Static Positioning - Single Solutions (Mean): Distances between receivers

ROVER	AMBIGUITES	REG. N	DIST. REF. (m)	DIST. (m) REC.	Δ (m)
1 & 2	FIX	1	2	63.93	61.93
1 & 2	FLOAT	1	2	5.56	3.56
1 & 2	FIX	2	2.15	3.39	1.24
1 & 2	FLOAT	2	2.15	3.48	1.33
1 & 2	FIX	3	2.15	11.39	9.24
1 & 2	FLOAT	3	2.15	11.46	9.31

It is interesting that the results change, and all the solutions obtained in static positioning in which all the solution are considered (mean), seem to be the most accurate[1]. In Tab.4.9 the difference is shown.

We have seen that, for now, the solutions obtained from the average, result to be those that produce better and more realistic outputs. It is interesting to try to calculate the final position using the median instead of the mean. Consider now, the case of the second recording,

[1]This is because the RTKPost software returns as one final solution one that may not be the best.

Tab. 4.9: Distance Difference Between All and Single Solutions

ROVER	AMBIGUITES	REG. N	Δ (m)
1 & 2	FIX	1	61.35
1 & 2	FLOAT	1	1.79
1 & 2	FIX	2	1.78
1 & 2	FLOAT	2	1.70
1 & 2	FIX	3	7.63
1 & 2	FLOAT	3	7.69

T2, with the aim of obtaining an estimate of the final distance between the two receivers better and therefore closer to reality. In Tab.4.10 the diistance between Devices calculated from the median and the mean is shown.

Tab. 4.10: Distance Between Devices - Median / Mean

Output	T2	Distances	Δ	Ref. (m)
Median	FIX	4.07	1.92	2.15
Median	FLOAT	4.08	1.93	2.15
Mean	FIX	5.18	3.03	2.15
Mean	FLOAT	5.19	3.04	2.15

As can be clearly seen from the results Tab.4.10, it is clear that the solution obtained, once all the solutions from RTKPost have been obtained, the median is the best, the mean is the latter. For a simpler interpretation of the results obtained, the final position obtained from the analysis can be observed thanks to the KML format Fig.4.21. The distance turns out to be in Fix is 4.07m, while the reference distance was 2.15m[1].

[1]As in Fig.4.11

Fig. 4.21: R1T2 - R2T2 Fix Distance

At the end of this analysis, which can be defined as preliminary, regarding the static positioning, the results obtained demonstrate a good performance of the two devices, in line with the expected results, considering the nature of the geolocation chipset. The precision in some cases, remains less than five meters and in some cases, Tab.4.7. even the two, this limited of course only to tests performed on February 27th. These analyzes have made it possible to understand in advance the performance of the devices with positioning and post-process techniques already consolidated over the years. The real challenge is to exploit a completely new approach, to fully exploit the potential of the devices. This variometric approach, born in the Department of Geodesy of the Sapienza University, will be illustrated in the next chapter.Tab.4.7.

Chapter 5

The Variometric Approach

After analysing the behaviour of the device in the classic methods of positioning, it is interesting to propose another type of approach and analysis. The variometric approach is based on time single differences of carrier phase observations continuously collected using a standalone GPS receiver on standard GPS broadcast products (orbits and clocks) available in real-time. The least squares estimation of the 3-D velocities is based upon the entire set of variometric Equations which can be written for two generic consecutive epochs (t and t + 1). The number of variometric equations depends on the number of satellites common to the two epochs. At least four satellites are necessary in order to estimate the four unknown parameters for each consecutive epoch couple. Differently from other data processing schemes (differential positioning (DP) and PPP[1]), the variometric approach does not require phase ambiguity resolution and it is also able to work with single-frequency data only, as the case of the S8 devices. Overall, one receiver works

[1]Precise Point Positioning

in standalone mode and the epoch-by-epoch displacements (which are equivalent to velocities) are estimated. Then, velocities are integrated (and derivative calculated) over the time interval of interest to retrieve the comprehensive receiver displacements (and accelerations).[20]

The first implementation of the variometric approach was proposed in 2011 in the VADASE (Variometric Approach for Displacement Analysis Stand-alone Engine) software, as an innovative solution to estimate in real-time rapid movements of GPS receivers in a global reference frame. To prove the effectiveness of the approach, the algorithm was implemented in a desktop application capable to process standard Receiver Independent Exchange format (RINEX) files containing observations and ephemeris acquired by a GNSS receiver. Its validity was proved in the GPS seismology field through the application to the catastrophic Tohoku-Oki earthquake (Mw 9.0, March 11, 2011), when VADASE was the first approach capable of computing accurate displacements caused on 2 International GNSS Service (IGS) Japanese stations (MIZU and USUD), immediately after the availability of data. VADASE was also applied to the Emilia earthquake (Mw 6:1, May 20, 2012) in case of small displacements.[21]

The variometric algorithm was conceived to detect, in real-time, fast and short duration displacements occurring to a single GNSS receiver. However, the original implementation in the VADASE software focused on seismology and monitoring applications, where the initial coordinates are known with high accuracy (better than 0.5 m, which is normally the case for reference stations receivers or monitoring markers) and the receiver is expected to undergo limited movements (up to few meters) around its starting position.

The initial receiver position P_0 was simply retrieved from the RINEX

header and used to compute, for all available N epochs, the known terms of the variometric Equations in order to derive the displacements between epoch t and t + 1. Following the described processing design it appears evident how the performances of VADASE were significantly decreasing in kinematic applications, where the errors in the functional model computed using initial receiver position P_0 grew proportionally to the receiver movements.[20],[21] In Fig.5.1 the processing scheme of VADASE.

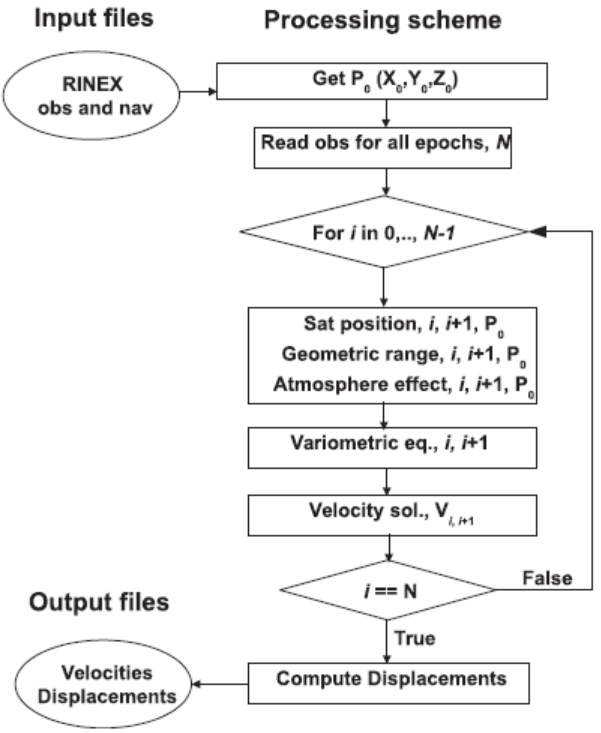

Fig. 5.1: Processing Scheme - VADASE [20]

5.1 The Static Approach

The potential of the VADASE was used for the analysis in the static case, previously described in the relative positioning. Each device has been analysed in every recording (T) with different configurations: analysing only the GPS and then subsequently also in combination with Galileo. Another configuration was made considering the cross-validation, in particular the LOOCV, called Leave-one-out cross-validation. These are the configurations used for the analysis of each device, for each period with VADASE:

- GPS - LOOCV OFF
- GPS - LOOCV ON
- GPS + Galileo - LOOCV OFF
- GPS + Galileo - LOOCV ON

As explained in the previous paragraph, the reference position P_0 expressed in X, Y, Z has been inserted in the RINEX header position. The position derived from the median of the studies in the previous chapter was used (Static Case). Obviously only the frequency L1 was considered, with the ionospheric model active.

Once the outputs calculated by VADASE are obtained, it is possible to process them through Python, seeing the behaviour of the device. In particular, what is processed is:

- Mean E,N,Up
- Mean N Satellites

- Standard Deviation E,N,Up

In the Tab.5.1 it can see the results of the analysis obtained with VADASE in the case of the previous chapter of the Relative Positioning.

The results obtained, show us how small the oscillations E, N, Up are. In some cases, as in the third recording, there are averages of a few millimeters, all obtained with a single-frequency device chipset. The results are excellent and in particular, during graphic processing of the results it is possible to observe a phenomenon already hypothesized and already present on other smartphone devices: The Duty Cycle.

Tab. 5.1: VADASE Results - Static Positioning - February 27

R1T1								
CONSTELLATION	LOOCV	Mean N Sat	East Mean	North Mean	Up Mean	Dev. East	Dev. North	Dev. Up
GPS	OFF	6.28	0.018	-0.087	-0.084	0.088	0.304	0.433
GPS	ON	6.49	-0.006	-0.092	-0.214	0.275	0.304	1.659
GPS + GALILEO	OFF	6.28	0.018	-0.087	-0.084	0.088	0.304	0.433
GPS + GALILEO	ON	6.49	-0.006	-0.092	-0.214	0.275	0.304	1.659
R1T2								
CONSTELLATION	LOOCV	Mean N Sat	East Mean	North Mean	Up Mean	Dev. East	Dev. North	Dev. Up
GPS	OFF	7.39	-0.02	0.024	-0.155	0.141	0.171	0.675
GPS	ON	7.23	-0.021	0.022	-0.146	0.13	0.164	0.532
GPS + GALILEO	OFF	8.1	-0.023	0.028	-0.168	0.153	0.191	0.708
GPS + GALILEO	ON	7.86	-0.023	0.029	-0.167	0.154	0.19	0.692
R1T3								
CONSTELLATION	LOOCV	Mean N Sat	East Mean	North Mean	Up Mean	Dev. East	Dev. North	Dev. Up
GPS	OFF	6.97	0.012	0.001	0.01	0.119	0.148	0.358
GPS	ON	6.9	0.011	0.003	0.008	0.115	0.106	0.349
GPS + GALILEO	OFF	7.32	0.004	0.008	-0.017	0.124	0.149	0.409
GPS + GALILEO	ON	7.19	0.004	0.011	-0.02	0.121	0.117	0.376
R2T1								
CONSTELLATION	LOOCV	Mean N Sat	East Mean	North Mean	Up Mean	Dev. East	Dev. North	Dev. Up
GPS	OFF	5.44	0.026	0.054	-0.276	0.171	0.57	1.69
GPS	ON	5.78	0.025	0.023	-0.179	0.138	0.389	0.804
GPS + GALILEO	OFF	5.44	0.026	0.054	-0.276	0.171	0.57	1.69
GPS + GALILEO	ON	5.78	0.025	0.023	-0.179	0.138	0.389	0.804
R2T2								
CONSTELLATION	LOOCV	Mean N Sat	East Mean	North Mean	Up Mean	Dev. East	Dev. North	Dev. Up
GPS	OFF	6.46	-0.028	0.054	-0.224	0.154	0.18	0.816
GPS	ON	6.23	-0.005	0.031	-0.099	0.133	0.158	0.639
GPS + GALILEO	OFF	6.46	-0.028	0.054	-0.224	0.154	0.18	0.816
GPS + GALILEO	ON	6.23	-0.005	0.031	-0.099	0.133	0.158	0.639
R2T3								
CONSTELLATION	LOOCV	Mean N Sat	East Mean	North Mean	Up Mean	Dev. East	Dev. North	Dev. Up
GPS	OFF	6.91	-0.011	0.085	-0.397	0.205	0.502	1.769
GPS	ON	6.79	-0.004	0.053	-0.309	0.187	0.345	1.384
GPS + GALILEO	OFF	7.51	0.01	0.045	-0.266	0.656	1.042	2.232
GPS + GALILEO	ON	7.29	0.007	0.014	-0.191	0.628	1.023	2.15

5.1.1 The Duty - Cycle

Many devices have a dynamic demand, for example, smartphones and embedded sensing devices. These devices have an idle or sleep power consumption of the order of micro-watts and their active peak power consumption can be of the order of watts. The difference between these two models, idle and active, can be several orders of magnitude. The battery and power delivery systems must support these wide power demand bandwidths [22].

Given a periodic event, the duty cycle is the ratio of the duration of the active state T_p, to te total duration of the cycle T:

$$D = \frac{T_P}{T} \tag{5.1}$$

The power spent during time T can be determined as the power consumption of the states P_p and P_i for the active and idle states. The power consumption of the duty state is significantly larger than for the idle state. The average consumption is given by:

$$P_avg = \frac{P_P T_P + P_i T_i}{T} \tag{5.2}$$

The duty cycle is very important for wireless communications that tipically have at least three states: idle, transmitting, and receiving. The latter two states are the high-power states of wireless radios. The overall power can be optimized by making the duty cycles short so that the device spends most of its time in a lower-power state and minimizes the time in the high-power state. For example the radio could have longer intervals between transmissions and receptions and keep the active periods as short as possible. The Fig.5.2, summarizes

the meaning of duty cycle:

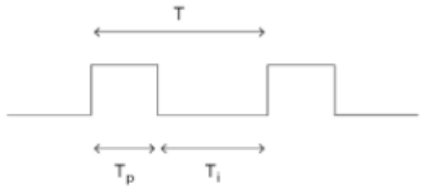

Fig. 5.2: The Duty - Cycle
[22]

A modern smartphone consists of many subsystems that have their own duty cycles. The geolocation system is one of these. As mentioned, one of the major challenges for smartphone manufacturers is to increase the battery life [22]. Since continuous use of a GNSS receiver drains the battery, the GNSS chip manufacturers use the duty cycle. The duty cycle of the power allows the receiver to track GNSS data continuously only for a fixed period, called burst period, before shutting down the main routines inside the chipset. The carrier-phase differential techniques require the resolution of integer ambiguities before the estimation of a precise navigation solution. These ambiguities are constant and estimable so-long as a receiver maintains lock on each signal's carrier phase. However, any interruption in signal data collection introduces integer discontinuities in the ambiguities: in presence of several interruptions, ambiguities are very difficult to estimate. Professional and, now, even mass-market receivers continuously track signals and specific algorithms can be applied to resolve carrier-phase ambiguities [9]. Typically, the power-saving duty cycling wakes up the receiver once a second for only few milliseconds per second and unknown additional cycles offsets from the true phase arise at the beginning of each

duty cycled measurement intervals: as consequence, the receiver is not capable of measuring the full cycle changes in phase that may occur between duty-cycles interval. Considering an interval of 1 second, the burst period can be for example 200 msec: the user can still access to the measurements every second but what happens in the remaining 800 msec is not known. There is however an exception to this process: the receiver remains continually active while decoding the navigation message. From a cold start, it takes minutes to decode the full message, leaving the users to track continuously the carrier-phase. All this is evident in the elaborated graph obtained from pyton where the measurement E, N, Up is shown with respect to the recording periods: (From 0 to 120 epoch = Burst Period) as in Fig.5.3.

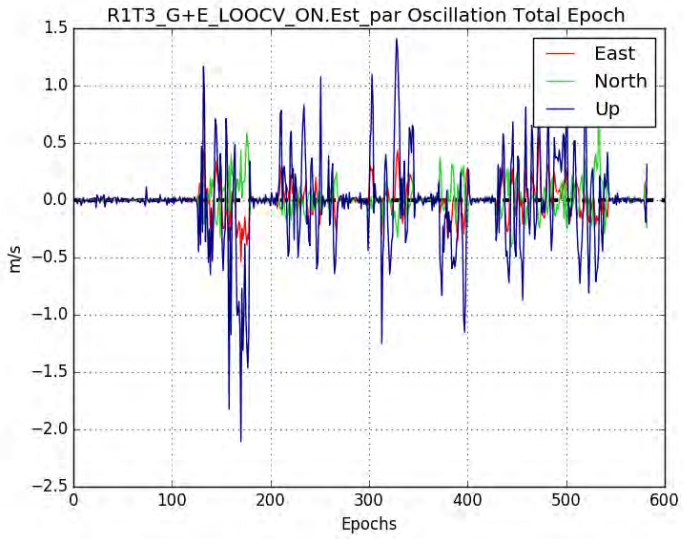

Fig. 5.3: R1T3 - G+E - LOOCV ON - Velocities

5.1.2 The Static Approach during Burst Period

The data obtained from the VADASE therefore, once considered the duty cycle phenomenon, have been filtered. For filtered, we mean that all those epochs in which the device was in the burst period have been analysed separately. The results obtained after filtering will increase the accuracy considerably. The comparison between the two devices in the respective best registrations R1T3, R2T3 is shown below, Fig.5.4,5.5.

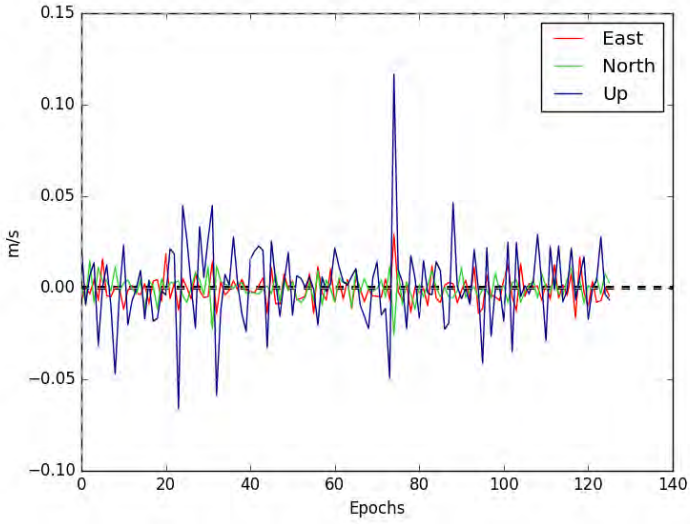

Fig. 5.4: R1T3 - G+E - LOOCV ON - Burst Period Velocities

In the cases taken into consideration it is understood that the duration of the burst period changes from device to device. The duty cycle is managed by microsensors inside the smartphone that deal, as explained in the previous paragraph, to manage the energy in the subsystems [22]. However, the duration of this period is different and not

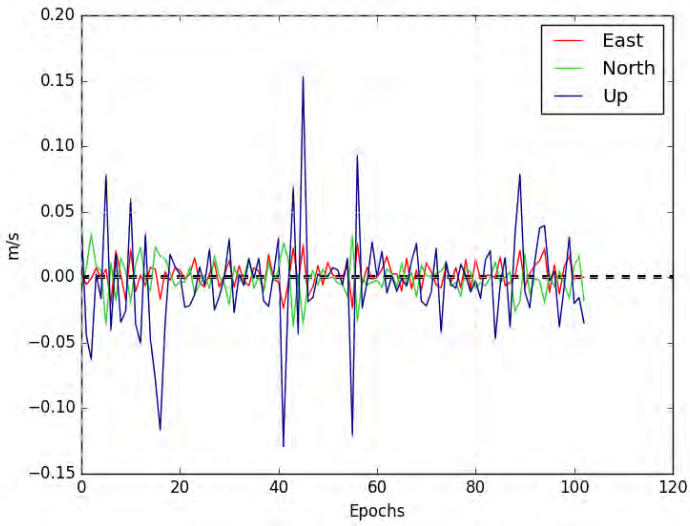

Fig. 5.5: R2T3 - G+E - LOOCV ON - Burst Period Velocities

fixed for each device. To observe the potential of VADASE the analysis was performed by calculating and considering only the filtered parts of the recordings in the burst period. The results[1] are shown in Tab.5.2.

Tab. 5.2: VADASE Results - Burst Period - February 27

R/T	Mean Sat.	East Mean	North Mean	Up Mean	Dev. East	Dev. North	Dev. Up
R1T1	6.7	0.002	-0.001	0.002	0.026	0.031	0.061
R1T2	7.9	0.000	0.001	0.004	0.007	0.008	0.017
R1T3	7.8	-0.001	0.000	0.001	0.008	0.007	0.022
R2T1	5.8	0.020	0.047	-0.197	0.085	0.231	0.613
R2T2	6.6	0.001	0.000	0.001	0.009	0.012	0.051
R2T3	7.6	0.002	-0.001	-0.004	0.010	0.013	0.038

The analysis of the table allows us to see that the results are variable according to the number of observations processed, by the number of synchronized satellites. In fact, the most uncertain measurements are

[1]The configuration includes GPS + Galileo and LOOCV ON

those where the average of the satellites was less than six Tab.5.2. The solutions that present a greater number of synchronized satellites, and therefore a greater coverage, return the results of averages very close to zero and standard deviations in East, North, Up which in plan are below the centimeter and the Up, lower at two centimeters. This noise, is very close to the noise of the variometric approach applied to phase observations applied to the geodetic receivers. (2/3 mm for the plan and a centimeter for the Up compponent). It can therefore be seen that the phase observations taken by the S8 are of good quality, therefore consistent, which give results not far from the professional geodynamic devices. The variometric approach was used in static mode to do a quality analysis of phase observation. Once we understood that the phase was consistent, we decided to use it in a purely transport application.

5.2 The Kinematic Approach

The goal of Kin-VADASE is to retrieve accurate kinematic parameters of moving receivers so that the variometric algorithm can be exploited also in GNSS navigation applications. The original implementation of the variometric approach for GNSS seismology and monitoring purposes was meant to detect velocities and displacements of permanent stations, which were modelled as receivers in pseudo-static conditions. As such, the initial coordinates were supposed to be known with high accuracy and they were constantly used in the variometric equations. In Kin-VADASE initial coordinates, are continuously updated by the epoch-by-epoch variometric solutions, accounting for the estimated displacement with respect to the previous position. If the initial coordinates (starting position of the moving vehicle) are not known, they are estimated based on code observations.[20][21]

In order to satisfy the kinematic requirements, the initial scheme used for implementing VADASE software was revised and updated. Initial coordinates P_0 are taken from the RINEX observation file header, or, if not available, are estimated for the first epoch based on code observations. Then, at each epoch i a new receiver position P_i, derived on the basis of the previous position and of the estimated velocity, is used to compute the variometric equations. Additionally, in order to satisfy requirements related to navigation, Kin-VADASE allows direct visualization of the receivers trajectory on Google Earth platform by providing a suited Keyhole Markup Language (KML) output file.[20]

The principal differences with respect to VADASE are highlighted in blue as in Fig.5.6. The receiver position P_i used to compute Equation at epoch i is updated on the basis of the estimated velocity.

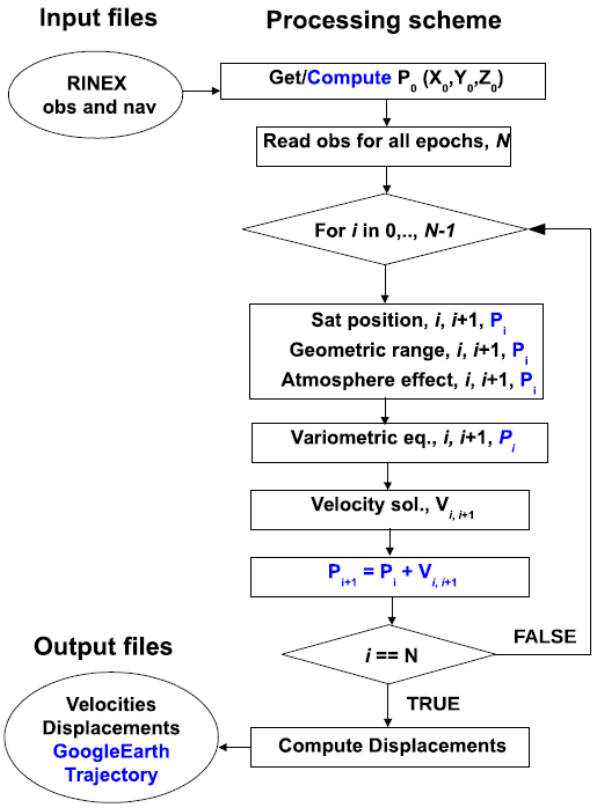

Fig. 5.6: Processing Scheme - Kin VADASE [20]

5.2.1 The Flight Test

The kinematic analysis of the Kin-VADASE was carried out on 23^{rd} March 2018, in a sunny but windy morning[1]. The vehicle used is an ultralight aircraft of Tecnam, the P-92 Echo Light. The objective is to analyze the performance of the S8 device on a vehicle with the Variometric approach, with speeds exceeding 140 km / h. Compared to the tests on land vehicles, in fact, the air test, allows us to better analyze the measurements on the Up component, which in the case of land, remains of minimal interest. In Fig.5.7 it is possible to observe the external characteristics of the aircraft.

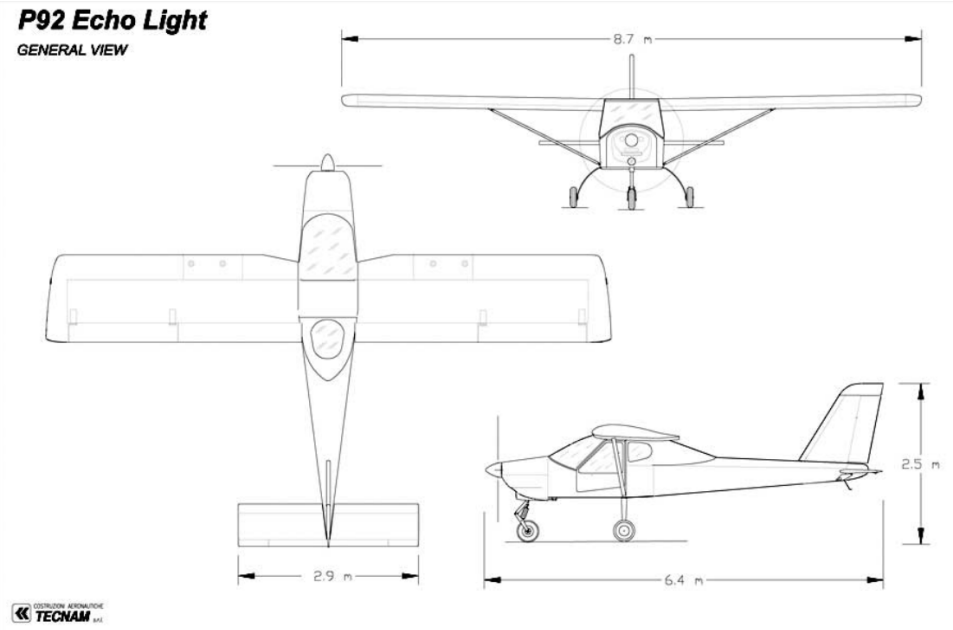

Fig. 5.7: Tecnam P92 - General View
[23]

[1]About 22 knots of Mistral

The test was performed, trying to replicate with precision, the behavior of a user, in this case a private pilot, positioning the Samsung S8 fixed on the cockpit. The duration of the acquisition is about thirty minutes (about 1800 epochs) and includes all phases of the flight: From the ignition of the engine in the apron to the parking of the vehicle immediately after landing. This is to observe all the phases and related speeds analyzed by the Kin-VADASE in a complete flight.

Fig. 5.8: Inside view of the cokpit with the S8 fixed, approaching Civitavecchia, flying over Santa Marinella

During the flight, there was good satellite coverage during the entire duration, this is evident in the Fig.5.9, which allows to observe a high average of the synchronized satellites, which in this study are limited to GPS and Galileo.

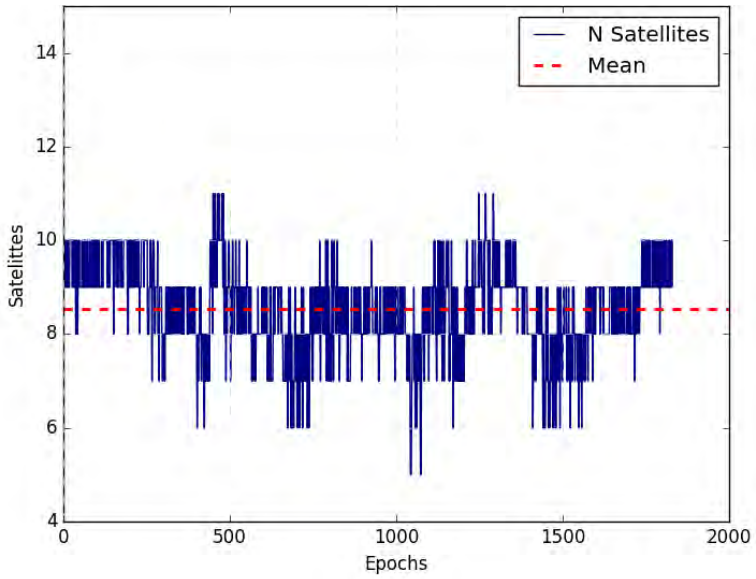

Fig. 5.9: Number of Satellites, G+E, LOOCV ON

As a first step, the flight in Single Positioning has been postprocessed, using the RTKPost software, as the other tests previously performed. After that, the KML file was obtained, which allowed to obtain the route performed Fig.5.10.

It is possible to observe, by enlarging the route, Fig.5.11, towards the Santa Severa Nord airfield, as the positions for each epoch are not coherent with the route actually followed. This is easily observable from the different heights that occur in the following two epochs, in fact they present a difference of several meters between consecutive epochs.

Both for the single positioning study and for the subsequent variometric approach, the analysis configuration including GPS + Galileo was used. Once the preliminary study in single positioning was com-

Fig. 5.10: Route of Flight in single positioning

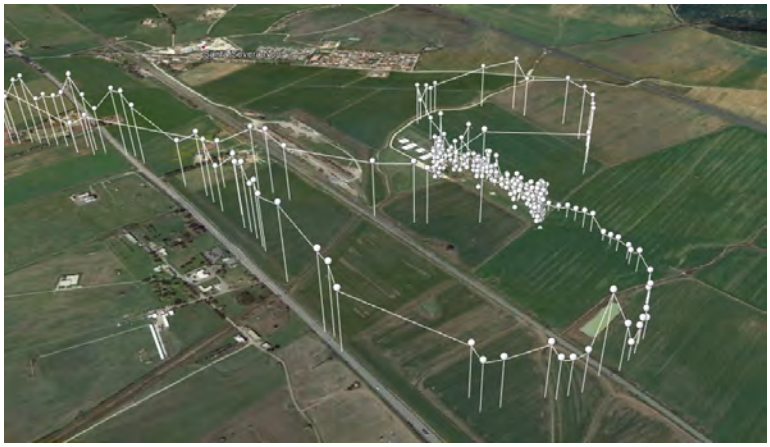

Fig. 5.11: Route near the Santa Severa Nord airfield - Single Point Positioning

pleted, the flight analysis began, with the approach of Kin - VADASE. The procedure follows the same of the study in a static way. The difference is obviously in the activation of the kinematic mode settings. Thanks to python plots it is possible to observe the velocities during

the entire flight, and therefore observe the different phases. All this can be seen in the graph showing all the velocities. From the Fig.5.13 it is possible to observe the various phases of the flight:

- The first phase, Fig.5.14 concerns the movement from the plane parking to the waiting area before the entrance of the runway indicated by the hold position signs. The reduced speeds on the taxiway are visible, about 15 km/h., This up to the hundredth registration epoch.

- The second phase, Fig.5.14 concerns waiting in the entrance area to the runway, waiting for the take-off permit. During this phase the aircraft is stopped with the engine running. About 200 epochs.

- The third phase concerns take-off, Fig.5.14. Thanks to the strong Mistral and the set flap configuration, the plane took off quickly, about 20 epochs, up to a stable altitude of about 1000 feet. It is interesting to observe how the graph shows the speeds in line with the characteristics of the aircraft as illustrated in the flight manual Fig.5.12, about 100 km/h for take-off. Immediately after take-off, the maneuver to remove and exit the airfield area towards the coast is also visible, heading towards the castle of Santa Severa.

- The fourth phase, Fig.5.15 concerns the "turning point", maneuver, which took place near the port of Civitavecchia. About 40 epochs.

- The fifth and last phase, Fig.5.16 concerns approach and landing.

Max Cruise Speed (KTAS)	190 km/h – 103 kts
Stall Speed (Flaps Down Power Off) KCAS	63 km/h – 34 kts
Practical ceiling	3660 mt – 12000 ft
Take off run	100 mt – 328 ft
Take off distance	230 mt – 755 ft
Landing Run	100 mt – 328 ft
Landing Distance	250 mt – 820 ft
Rate of climb	885 ft/min – 4,5 m/sec
Range	270 NM – 500 km

Fig. 5.12: P92 - Performance
[23]

From the graph it is evident how the speeds converge to the zero axis at the moment of the "touch" at the epoch 1740. From here it follows the taxiing to the parking area and to the shutdown of the engine.

In the graphs, it is evident that in many epochs the Up component is not continuous. This is because the weather conditions of flight were not among the best at the time of the air test. During the entire duration of the flight in fact, the air was turbulent, this fact has influenced the aircraft, unleashing numerous accelerations along all the components, many times, higher than 2 m/s.

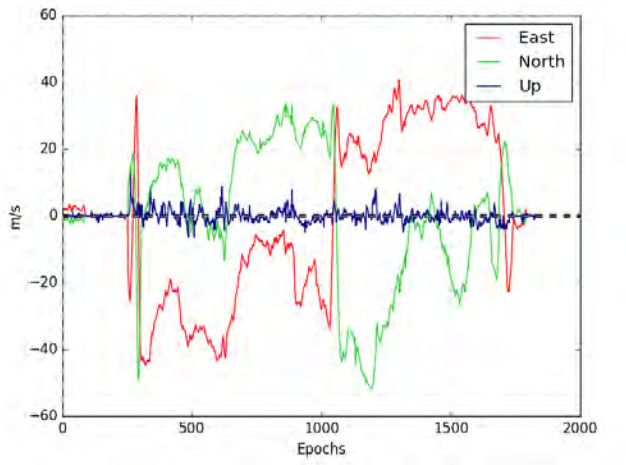

Fig. 5.13: Flight Velocities

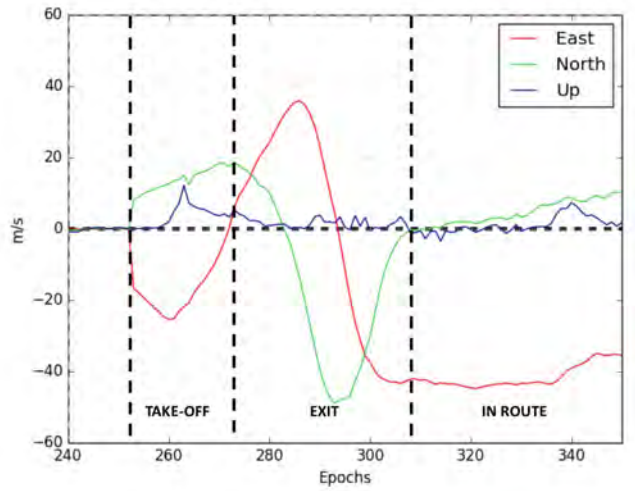

Fig. 5.14: Flight Velocities during Take Off

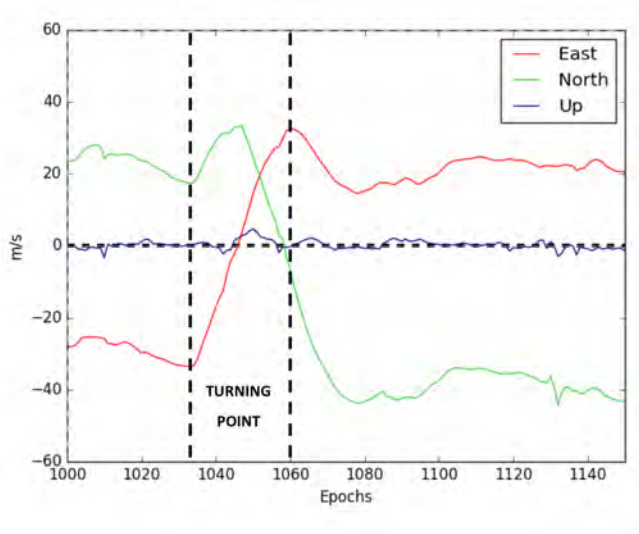

Fig. 5.15: Flight Velocities - Turning Point

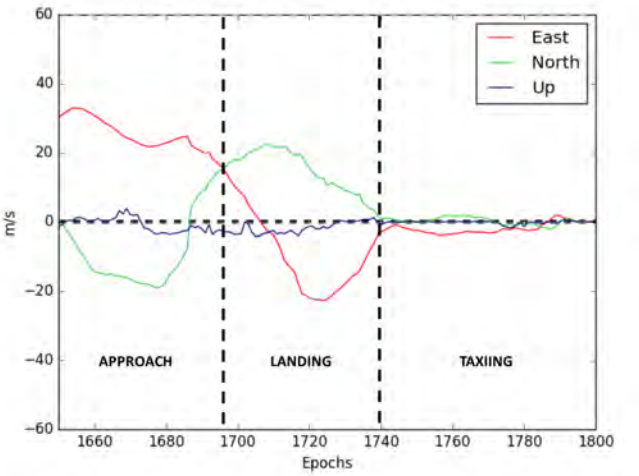

Fig. 5.16: Flight Velocities during the landing

This acquisition, which lasted for about thirty minutes, also showed the importance and functionality of the smartphone's internal microsensors. In fact, the device did not enter the dutycycle phase, maintaining the computational effort always in burst period (as in the static case), this because, except for short periods such as waiting for entry on the track, the device was always in motion, requiring the OS to always update the position. [22]

The next step of the analysis allowed to obtain a comparison with the route obtained in KML single positioning. The Kin-VADASE has indeed returned a set of points obtained in an output file for each single epoch. These points transformed into KML have returned a route much more accurate than the first, without the obvious aberrations returned by the classic single point positioning technique, which uses a standard gnss device.

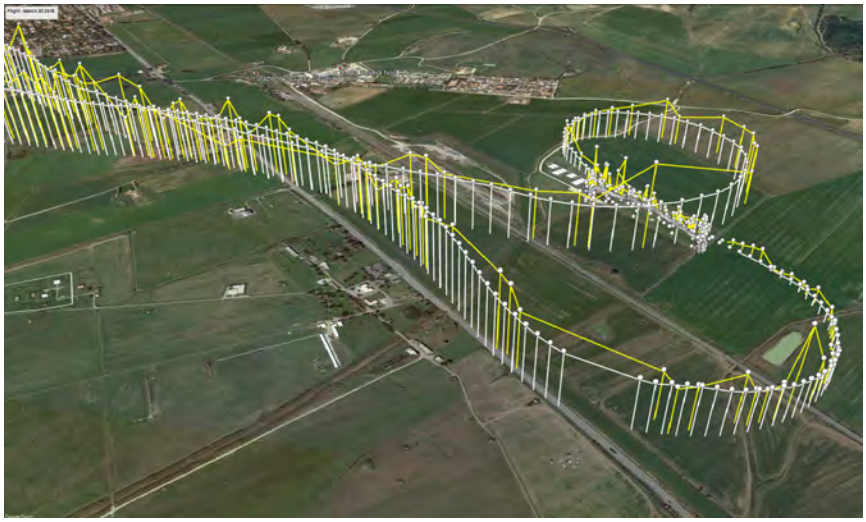

Fig. 5.17: SPP Vs. Kin-VADASE - Airfield

Fig. 5.18: SPP Vs. Kin-VADASE - Port of Civitavecchia

In Fig.5.17 5.18 it is shown the differences between the two routes: the white represents the one obtained from the Kin-VADASE and the yellow one in SPP. The fact that the duty cycle has not come into operation, causing continuous movement of the device, means that in the future, this can be exploited to the maximum for the high-precision positioning of a device in a vehicle.

Chapter 6

Conclusions

In the previous chapters we have seen how this analysis followed an "Working in Progress" approach. This is because, on the one hand, the availability of measurements on a smartphone is recently released[1], on the other hand because, the availability of tools for obtaining the measurements have not been optimized, considering this as a novelty.

It is therefore clear that the entire measurement process has been dealt with the awareness that perhaps the results could be disappointing as this was the first study of the capabilities of the Samsung home smartphone.

This was demonstrated by the initial difficulty and lack of appropriate tools to achieve a standard of collected measurements. The RINEX format has been tested with different methods and tools, such as matlab scripts and applications that have not been satisfactory due to a lack of completeness, such as the failure to register all the GNSS constellations.

[1]The Android 7, Nougat, was officially released on August 22, 2016, with Nexus devices being the first to receive the update. The LG V20 was the first smartphone released with Nougat [12]

Despite all these initial efforts, in the end, the release of the Geo ++ app on the market, has allowed to speed up the studies and to deepen the various positioning techniques carried out.

The good results obtained in the single positioning proved to be in line with what was expected from a single-frequency chipset, as well as the static positioning study. However, this has been proven with code observations. The real innovative challenge was to use a new approach, the variometric approach, previously applied only to multi-frequency geodetical receivers, which instead use phase observations.

For this reason, in a first phase we tried to understand if the analyses carried out in the static field were effective with the phase observations being consistent. The results obtained, show a clear consistency of the measurements. In the case of good coverage of satellites (R1T3) a dispersion of the velocities, corresponds lower than one centimeter in the planimetric components and less than two centimeters in the up component, values that are not far from the values of geodedical receivers. This allowed us to test this approach on a vehicle where it had never been tested, an ultra-light aircraft. The route obtained through the Kin-VADASE demonstrates how, with this approach, it is possible to have an excellent measurement with a smartphone device and a low-cost single-frequency chipset. Not only this, the speed chart also allows analyzing the quality of the flight, describing all the maneuvers with the corresponding speeds.

For this reason a variometric approach, applied to a mass market device, like a smartphone, would allow to have very precise measurements in the context of navigations, in movements that a person makes every day, from his car to his ultralight plane or boat or bicycle, practically in all the vehicles in his possession.

Appendix A

In this appendix, the codes written in Python and examples of the various configurations are shown, used in the analysis of the measurements.

.1 Python Codes

- Transformation Function in Geodedical System

```
1
2  def XYZ2GEO(X0,Y0,Z0):
3
4      a=6378137 #Semi-minor axis
5      f=1/298.257223563 #Crush
6
7      e2=f*(2-f)
8      e=sqrt(e2) #Eccentricity
9
10     b=a-a*f #Semi-major axis
11
12     eb2=(a**2-b**2)/(b**2)
13     ro=sqrt(X0**2+Y0**2)
14     psi=arctan(Z0/(ro*sqrt(1-e2)))
15     lam0=arctan(Y0/X0)
16     fi0=arctan((Z0+eb2*b*sin(psi)**3)/(ro-e2*a*cos(psi)**3))
17
18     N=a/sqrt(1-e2*sin(fi0)**2)
19     h0=ro/cos(fi0)-N
20
21     fideg=fi0/np.pi*180
22     lamdeg=lam0/np.pi*180
23     h=h0
24
25     return(fideg,lamdeg,h)
```

- Transformation Function in East, North, Up System

```
1
2  def XYZ2ENU(X,Y,Z,X0,Y0,Z0,fi0,lam0):
3
4      mat1=zeros((3,3), float)
5      mat1[0,0]=-sin(lam0)
6      mat1[0,1]=cos(lam0)
7      mat1[0,2]=0
8      mat1[1,0]=-sin(fi0)*cos(lam0)
9      mat1[1,1]=-sin(fi0)*sin(lam0)
10     mat1[1,2]=cos(fi0)
11     mat1[2,0]=cos(fi0)*cos(lam0)
12     mat1[2,1]=cos(fi0)*sin(lam0)
13     mat1[2,2]=sin(fi0)
14
15     mat2=zeros((3,1),float)
16     mat2[0,0]=X-X0
17     mat2[1,0]=Y-Y0
18     mat2[2,0]=Z-Z0
19
20     mat3=mat1.dot(mat2)
21
22     E=mat3[0,0]
23     N=mat3[1,0]
24     U=mat3[2,0]
25
26     return (E,N,U)
```

- Reading File .pos

```
1
2  filedaleggere = raw_input ("Digitare File da Leggere: -> ")-> raw_input
3
4  # Columns Selection from File Rinex .pos
5  X,Y,Z,Q,nsat = genfromtxt(filedaleggere,unpack=True,
6  usecols=(2,3,4,5,6), skip_header=25)
7
8  # Mean & Standard Deviation
9  mediaX=np.mean(X)
10 mediaY=np.mean(Y)
11 mediaZ=np.mean(Z)
12 mediansat=np.mean(nsat)
13
14 devX=np.std(X)
15 devY=np.std(Y)
16 devZ=np.std(Z)
17
18 medianX=np.median(X)
19 medianY=np.median(Y)
20 medianZ=np.median(Z)
21
22 lat,lon,quota=XYZ2GEO(mediaX,mediaY,mediaZ)
```

- Transformation in Radians and vector cycle E, N, Up

```
1
2  # Radians Transformations
3  latrad=lat/180*np.pi
4  lonrad=lon/180*np.pi
5
6  E=[0.0]*len(X)
7  N=[0.0]*len(X)
8  U=[0.0]*len(X)
9
10 # Vector Cycle E,N,UP
11 for i in range(len(X)):
12 E[i],N[i],U[i]=XYZ2ENU(X[i],Y[i],Z[i],mediaX,mediaY,mediaZ,latrad,lonrad)
13
14 mediaE=np.mean(E)
15 mediaN=np.mean(N)
16 mediaU=np.mean(U)
17
18 devE=np.std(E)
19 devN=np.std(N)
20 devU=np.std(U)
```

- Example of a graphic plot

```
1
2  # Graphic Plot (Epocs, Meters)
3  #plt.figure(figsize=(22,10)) #Changing Plot Dimension
4  plt.title(filedaleggere+" Positions")
5
6  axes = plt.gca()
7  #axes.set_xlim([0,400])
8  plt.axhline(0, linestyle="dashed", color="black", linewidth=3,)
9  plt.plot(E, color="red", label="Est", linewidth=1)
10 plt.plot(N, color="limegreen", label="North", linewidth=1)
11 plt.plot(U, color="navy", label="Up", linewidth=1)
12 plt.xlabel("Epochs")
13 plt.ylabel("Meters")
14 plt.grid(True)
15 plt.rc('grid', linestyle="--", color="lightgrey")
16 plt.legend()
17 plt.savefig(filedaleggere+"_plot.png")
18 plt.close()
```

.2 VADASE Configuration

- VADASE - Configuration File Example

```
1
2   **** VADASE CONFIGURATION FILE ****
3
4  ## COMBINATION ##
5  L1
6  ## CONSTELLATION(S) ##
7  G+E
8  ## OBSERVATION(S) ##
9  L1C+L2W
10 ## SATELLITE(S) TO BE EXCLUDED ##
11 NULL
12 ## TROPOSPHERE MODEL ##
13 4
14 ## FILTER DATA IN DIFF_VAR ##
15 0
16 ## SAGNAC EFFECT ##
17 1
18 ## COMPUTE RCV CLK OFFSET ##
19 1
20 ## IONOSPHERE MODEL ##
21 1
22 ## LOOCV ##
23 1
24 ## KIN MODE ##
25 1
26 ## SAT HEALTH ##
27 0
28 ## TOE CONTROL ##
29 9000
30 ## WEBAPP ##
31 0
32 ## UNVERBOSE ##
33 0
34 ## LOOCVALPHA ##
35 5
```

- VADASE Kin Point output, 23 March 2018 - Example

```
1
2  465132 4642253.0225 985297.6017 4247176.0273
3  465133 4642253.3423 985299.9447 4247175.2147
4  465134 4642253.5788 985302.2522 4247174.1961
5  465135 4642254.2553 985304.3816 4247173.8214
6  465136 4642254.4914 985306.5817 4247173.1349
7  465137 4642254.6798 985308.6838 4247172.3981
8  465138 4642254.7801 985310.7145 4247171.9174
9  465139 4642254.8597 985312.5837 4247171.4757
10 465140 4642255.1070 985314.3609 4247170.7113
11 465141 4642255.3958 985316.2545 4247169.9305
12 465142 4642255.8540 985318.3325 4247168.8971
13 465143 4642255.9736 985320.6395 4247167.9562
14 465144 4642256.0709 985322.9973 4247167.3202
15 465145 4642256.2211 985325.3118 4247166.6454
16 465146 4642256.6017 985327.6744 4247166.1070
17 465147 4642257.0886 985329.8775 4247165.2646
18 465148 4642257.4213 985331.9582 4247164.2309
19 465149 4642257.7223 985333.8834 4247163.1365
20 465150 4642258.1540 985335.7639 4247162.1881
21 465151 4642258.2992 985337.5892 4247161.6638
```

References

[1] Sanz Subirana, J., Juan Zornoza, J., Hernandez-Pajares, M. GNSS Data Processing: Volume I: Foundamentals and Algorithms. *European Space Agency, ESA Communications, ESTEC*, May 2013. v, 4, 5, 6, 8, 9, 10, 11, 14, 25, 28

[2] Cao, C., Jing, G., Luo, M. COMPASS Satellite Navigation System Development. *PNT Challenges and Opportunities Symposium, Stanford, California,USA*, November 2008. 7

[3] Galileo SIS ICD 2010 Galileo Open Service Signal In Space Control Document. `http://ec.europa.eu/enterprise/policies/satnav/galileo/open-service/index_en.htm`. 7

[4] GLONASS ICD Technical Report v.5.1., 2008 6

[5] GPS/SPS-PS 2008 Global Positioning System Standard Positioning Service. Performance Standard. `http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf`.. 6

[6] Biagi, L. I Fondamentali del GPS. *Geomatics Workbooks,Vol.8*, 2009. vi, 12, 13, 15, 36, 45, 47, 53

[7] Davis, P., Blewitt, G. Methodology for global geodetic time series estimation: a new tool for geodynamics. *Journal of Geophysical Research, Vol 105*, May 2010.

[8] Global navigation Satellite System (GNSS) Education, Priceton University v, vi, 12, 13, 15, 16, 54, 55

[9] Pirazzi, G., Mazzoni, A., Biagi, L., Crespi, M. Preliminary performance analysis with a GPS+Galileo enabled chipset embedded in a smartphone. *30th International Technical meeting of the Satellite Division of ION, Portland, Oregon*, September 2017. v, 17, 22, 33, 70

[10] Navarro, M., Bernhardt, N., kirchner, M., Redenkiewicz, J., Sunkevic, M. Assessing Galileo Readiness in Anfroid Devices Using Raw Measurements. *30th International Technical meeting of the Satellite Division of ION, Portland, Oregon*, September 2017. v, 18, 20

[11] Gurtner, W. RINEX, The Receiver Independent Exchange Format Version, 3.01. *Astronomical Institute, University of Bern*, June 2009. 23, 25, 27, 28

[12] Raw GNSS measurements Android Developers Website - Google. `https://developer.android.com/guide/topics/sensors/gnss`. v, 30, 32, 33, 34, 87

[13] Galaxy S8 Samsung Website. `http://www.samsung.com/it/smartphones/galaxy-s8/`. v, 30, 31

[14] RTKLIB: An Open Source Program Package for GNSS Positioning RTKLIB Website. http://www.rtklib.com/. vi, 35

[15] Matlab Mathworks Website. https://it.mathworks.com/products/matlab.html.

[16] Geo++ GMBH. http://www.geopp.de/. vi, 40

[17] Python Python Software Foundation US. https://www.python.org/. 45

[18] Stazioni Permanenti GNSS Regione Lazio. http://www.regione.lazio.it/rl_sitr/?vw=contenutidettaglio&id=132.

[19] Google Earth Pro Software Google. https://www.google.com/intl/it/earth/desktop/. vi, 37, 39, 41

[20] Branzanti, M. Colosimo, G. Mazzoni, A. Variometric approach for real-time GNSS navigation: First demostration of Kin-VADASE capabilities. *Advances in space research, ScienceDirect*, September 2016. 64, 65, 75, 76

[21] Colosimo, G. Crespi, M. Mazzoni, A. Real-time GPS seismology with a stand-alone receiver: A preliminary feasibility demostration. *Journal of geophysical research, Vol.116*, November 2011. 64, 65, 75

[22] Tarkoma, S. Siekinen, M. Lagerrspetz, E. Xiao, Y. Smartphone Energy Consuption: Modeling and Optimization. *Cambridge University Press*, 2014. 69, 70, 72, 85

[23] P92 - Echo Light Tecnam. https://www.tecnam.com/it/aircraft/p92-echo-light/. 77, 82