



French Civil Aviation University (ENAC)
7 avenue Édouard Belin - CS 31055 Toulouse Cedex 4
Tél. +33 (0)5 62 17 40 00 - www.enac.fr

ASNAT21

PVT Computation

Presented on January 2022 by

Ishani Khanra

Tutors:

Dr. Anne-Christine Escher

Contents

1	INTRODUCTION	5
1.1	Context	5
1.2	Objective	5
1.3	Overview	5
2	DATA COLLECTION AND CONVERSION	6
2.1	Data Collection	6
2.2	Data Conversion	7
3	IMPLEMENTATION ON MATLAB	8
3.1	Data extraction	8
3.2	Satellite Position Computation	8
3.2.1	Clock corrections	10
3.2.2	Computation of SV position in ECEF at transmission time	10
3.2.3	Computation of SV position in ECEF at reception time	11
3.3	Receiver position computation	11
3.4	Tropospheric Correction	12
3.5	Ionospheric Correction	12
4	ANALYSIS OF RESULTS	14
4.1	Number of visible satellites	14
4.2	Raw pseudorange measurements	15
4.3	Signal-to-noise ratio	15
4.4	Doppler frequency	16
4.5	Satellite vehicle position	17
4.6	Satellite Clock correction	18
4.7	Ionospheric range correction	18
4.8	Tropospheric range Correction	19
4.9	Receiver position	20
4.9.1	Receiver position in x,y,z coordinates	20
4.9.2	Receiver position on map using latitude and longitude	20
4.10	Dilution of precision	21
4.10.1	GDOP	21
4.10.2	VDOP	21
4.10.3	HDOP	21
4.11	Estimation Errors	22
4.11.1	X-coordinate estimated position error	22
4.11.2	Y-coordinate estimated position error	23
4.11.3	Z-coordinate estimated position error	23
4.11.4	Estimated receiver position on map	24
4.12	Code-Minus-Carrier	25
5	CONCLUSION	26

List of Figures

2.1	Data collection in open field-receiver, antenna and laptop with u-center	6
2.2	U-center suring data collection[1]	7
2.3	RTK-convertor	7
3.1	Flowchart of satellite vehicle position computation for each satellite	9
3.2	Flowchart of satellite clock range correction implementation	10
3.3	Flowchart of Sagnac correction implementation	11
3.4	Flowchart of LSE implementation	12
3.5	Flowchart of tropospheric correction implementation[6]	12
3.6	Flowchart of ionospheric correction implementation[5]	13
3.7	Flowchart receiver position computation	13
4.1	Number of visible satellites evolution with time	14
4.2	Raw pseudorange measurements evolutions with time	15
4.3	SNR measurements evolution with time	16
4.4	Doppler frequency measurements evolution with time	16
4.5	Satellite position for open field data	17
4.6	Satellite position for terrace data	17
4.7	Satellite clock range correction measurements evolution with time	18
4.8	Ionospheric range error measurements evolution with time	19
4.9	Tropospheric range error measurements evolution with time	19
4.10	Receiver position from implementation on map	20
4.11	GDOP evolution with time	21
4.12	VDOP evolution with time	22
4.13	HDOP evolution with time	22
4.14	X-coordinate estimated position error at all epochs	23
4.15	Y-coordinate estimated position error at all epochs	23
4.16	Z-coordinate estimated position error at all epochs	24
4.17	3D estimated position of receiver	24

4.18 Code minus carrier evolution with time	25
---	----

Chapter 1

INTRODUCTION

1.1 Context

Navigation and positioning has reached milestones with the technology of global navigation satellite systems, commonly known as GNSS. It refers to a collection of satellites orbiting around the earth providing position, navigation, velocity and timing data to GNSS receivers placed on the earth. The constellation of GNSS includes "GPS (Global Positioning System) developed by the United States, GLONASS (GLObal NAVigation Satellite System) developed by Russia, GALILEO (Europe's Satellite Navigation System), China's COMPASS/Bei-Dou, India's Regional Navigation Satellite System (IRNSS or NavIC) and Japan's Quasi-Zenith Satellite System (QZSS)." [4]. GNSS is used widely across the world for several important applications like surveillance, civil aviation, military, space, road, rail, maritime applications. The extended uses have been in the fields of telecommunications, precision agriculture, safety devices, mobile phones and mining. They provide reliable and accurate position solution to the user and hence find its use in various devices around the world.

1.2 Objective

The project focuses on the computation of position, velocity and time computation of the GNSS receiver. This involves implementing the data processing part of the receiver on MATLAB. The signal captured by the receiver is a raw data that needs conversion to an appropriate readable format for further processing. The project involves the understanding of the collected data and what each component signifies. It then focuses on the position computation of the receiver by implementation of MATLAB algorithm from the course. Finally, the impact of corrections on the receiver position is studied. The goal of the project thus involves the understanding of the whole process starting with the capture of the signal till the estimation of receiver position and the analysis of the results.

1.3 Overview

This report would be divided into four stages. The first stage is the collection of data with the GNSS receiver. The second stage captures the receiver position estimation and corrections by MATLAB implementation. Then, the report will focus on the results from second part and their impact and analysis. Lastly, a conclusion would be drawn.

Chapter 2

DATA COLLECTION AND CONVERSION

This section focuses on the collection of data in different situation for implementation of PVT algorithm and the analysis of the information received by the receiver from the GPS satellite. Then, we would highlight the conversion of the raw data using appropriate techniques in order to read the data and understand it's components. These parameters would be further be used to implement the positioning algorithm, that would be discussed in chapter 3.

2.1 Data Collection

For the purpose of this project, two sets of data were collected. The first data was collected for a period of two hours from the balcony of the residence. This data is a static data and would help to analyse an environment surrounded by building, making it more susceptible to multipath. The second data was also a static measurement taken for a period of one hour. It was taken at the center of the football field where there is much less possibility of multipath as the buildings are not very near the receiver. However, since the receiver had been placed on the ground, multipath due to ground may be observed.



Figure 2.1: Data collection in open field-receiver, antenna and laptop with u-center

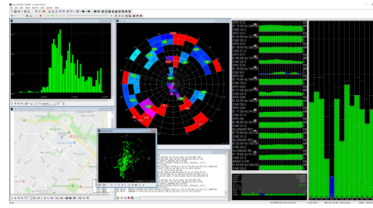


Figure 2.2: U-center suring data collection[1]

The data collection was done using U-Blox EVK-M8T GNSS evaluation receiver kit. Prior to the collection of data, U-center software interface 21.09 version was used. The receiver module was connected to the port of the the computer. The selection of port was done in the software. Once it was successfully connected to the software, a green icon could be seen on the bottom right corner. This is the signal that the receiver has been interfaced with the software and is ready for data collection. Once it is connected, we can observe that satellites are getting detected and they are appearing in the form of a graph and sky plots as seen in 2.1. Before starting to record the data, we perform some software configuration steps. This can be found on the "message view" tab from menu view. First, we should revert to default configuration, next we select the communication port and protocol and then we select the output messages. Before we start recording, we should open the packet console from the menu view and check the messages. Then we hit the record button and make sure the computer does not goto sleep mode. Finally, we stop the data collection after the desired time and store the data in a folder.

2.2 Data Conversion

The data obtained from the above process is the raw data and if we open the folder, we can observe that it is not readable. To convert it in a readable format like RINEX, we use RTKCONV version 2.4.2 software. RINEX stands for Receiver Independent Exchange Format, which is used in geodesy and helps to interchange data obtained from raw measurements from the satellites. The output from the conversion is two files- ".obs" which contains the observation file and the ".nav" which contains the navigation data. This data is readable with values in a specific format. The next section will focus on how this data is read in MATLAB to be used for PVT algorithm implementation.

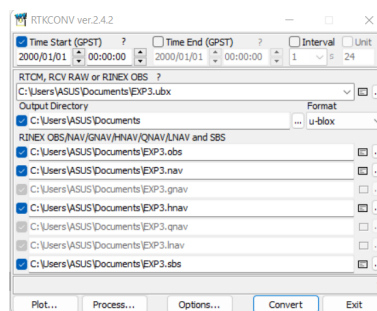


Figure 2.3: RTK-convertor

Chapter 3

IMPLEMENTATION ON MATLAB

3.1 Data extraction

The data obtained from the process discussed in section 2.2 gives us the raw measurements in ".obs" and ".nav" format. These files are imported on MATLAB and have to be extracted. In order to implement the PVT algorithm and find the estimated receiver or user position, we need to arrange the data contained in these files in separate variables. We use different matrices to extract these data and store them in an organised manner for ease of use and understanding. The data formats provided by our tutor was helpful in understanding what is contained in each of these files. This helped us to use the correct data as inputs to the modules and functions in the algorithm.

The ".obs" file is the file containing information in the form of header and data. The header contains information regarding the approximate position of the receiver, starting and ending time of observation. This position is required as a reference during the PVT solution using least square estimation. The other important information in the data part are the PRN number, the date and time of observation, number of satellites tracked at each epoch, the raw pseudorange measurements in meters, the carrier phase measurement in number of cycle, the doppler frequency in Hz and the carrier to noise ratio in dBHz.

The ".nav" file contains the navigation information in the form of header and data. The header portion contains information ionospheric alpha and beta parameters, the delta-UTC parameters. The data part contains the ephemeris data like the satellite vehicle's PRN number and it's health. The other information are satellite vehicle clock bias parameters, time of ephemeris in GPS time, group delay parameter, eccentricity, semi-major axis, time of clock and many more. These will be used for finding the 3D satellite vehicle position.

3.2 Satellite Position Computation

Computation of the satellite position in x,y and z coordinates for each satellite is the first step to compute the receiver position. This process is divided into several steps that would be discussed in this section. First we need to initialize certain constants which are given in Table 3.1.

Next, we reallocate certain variables to empty matrices. Next we design two for loops that would help in the position computation of each satellite. The first for loop is for each PRNs and the next loop is for each satellite vehicle in that epoch. We have the following flowchart that explains the further process.

Parameter	Values
Speed of light	299792458
Value of PI[m/s]	3.1415926535898
Earth's gravitational parameter [m^3s^{-2}]	$3.986005 * 10^{14}$
GPS L1 frequency(MHz)	1575.42
Earth's rotation rate (rad/sec)	$7.2921151467 * 10^{-5}$
Maximum number of epochs	10000
Force on earth	$-4.442807633 * 10^{-10}$

Table 3.1: Table of constants

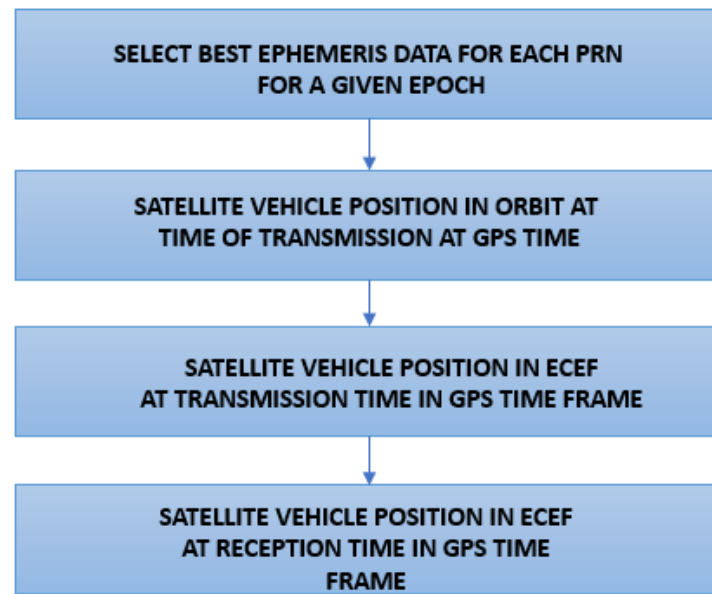


Figure 3.1: Flowchart of satellite vehicle position computation for each satellite

From the above flowchart, we see that the first step is to select the best ephemeris for each PRN at a given epoch. There may be more than one ephemeris data for the same PRN or satellite vehicle since our data collected is quite long. We need to choose the one which fits best. This selection is based on the lowest difference between the time of applicability of the ephemeris data and observation epoch. Next, we need to find the satellite vehicle's position in the orbit at the time of transmission in GPS time frame.

3.2.1 Clock corrections

The clock correction module can be represented in the following format.

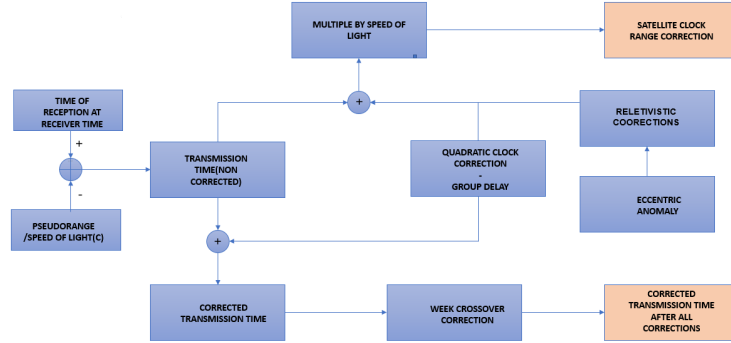


Figure 3.2: Flowchart of satellite clock range correction implementation

The first correction that needs to be implemented is the transmission time of the satellite vehicle in GPS time. It is given by the formula,

$$t_{tx}^{GPS} = t_{rx}^{GPS} - \rho/c \quad (3.1)$$

where, t_{rx} is the time of reception at GPS time and ρ is the pseudorange measurements taken from observation file(.obs). The quadratic clock correction also has to be computed for the satellite. These corrections are needed as the satellite clock might have some small errors associated with the clock of the satellite.

$$t_q = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 \quad (3.2)$$

where, a_{f0} is the SV clock offset, a_{f1} is the SV clock drift and a_{f2} is the SV clock drift rate. t_{oc} is the clock data reference time in seconds. These information are obtained from ephemeris information of nav file.

The satellite vehicle PRN code phase time offset between Receiver Time and GPS Time Scales in seconds is given by

$$\Delta t = t_q + t_r - t_{tGD} \quad (3.3)$$

$$\Delta t = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + t_r - t_{tGD} \quad (3.4)$$

In the above equation, t_{tGD} is the group delay correction. This delay results from delay between L1 P(Y) and L2 P(Y) based on the measurements made by the SV contractor when it was manufactured and is calculated by control segment. It is again found in ephemeris information of nav file. The relativistic correction is not added initially because for this computation, the knowledge of the value of eccentric anomaly has to be known. It is thus applied once this information has been found. It arises due to the fact that between the transmission and the reception of the signal, the earth has rotated.

$$t_r = Fe\sqrt{a}\sin E_k \quad (3.5)$$

where e is the eccentricity, a is the semi-major axis, E_k is the eccentric anomaly and $F = 4.442807633 \times 10^{-10}$. Finally, we apply the week cross over corrections depending on whether it is the beginning or end of the week.

3.2.2 Computation of SV position in ECEF at transmission time

This computation consists of three parts-finding the eccentric and true anomaly, the rotation arguments and the 3D position for each satellite.

The computation for true and eccentric anomaly consists of taking the semimajor axis from the ephemeris data and consequent computation of corrected mean motion, time from ephemeris epoch, mean anomaly, eccentric anomaly by iteration and the true anomaly.

Then, we compute the rotation arguments like the argument of latitude, correction argument of latitude, correction radius, correction inclination, corrected argument of latitude, corrected inclination and the corrected longitude of node.

Finally, we calculate the 3D position of each satellite vehicle. For this, using the computed parameters before, we calculate the orbital plane x-position, orbital plane y-position, ECEF x-coordinate, ECEF y-coordinate and ECEF z-coordinate.

Once the eccentric anomaly has been computed, the relativistic corrections can be applied to the clock correction model to find the clock range error.

3.2.3 Computation of SV position in ECEF at reception time

In the previous step, we have computed the SV position in ECEF at transmission time. This has to be converted to the position at reception time. For this, we construct a rotation matrix with estimated satellite positions, the corrected pseudorange and the mean angular velocity of the earth which is a constant. This is also known as the Sagnac effect correction. This is mainly due to the fact that between the time the signal is emitted from the satellite and the time it is received on the earth, the Earth has rotated. To account for this, we need this correction.

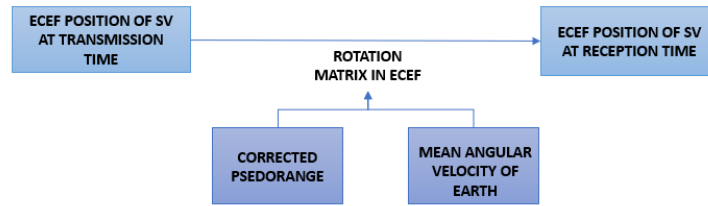


Figure 3.3: Flowchart of Sagnac correction implementation

3.3 Receiver position computation

This module computes the receiver's position with n measurement equations and four unknowns. The n measurement equations are the equations for each satellite tracked and the unknowns are the x, y, z coordinates and the receiver clock bias. The measurement equations are non-linear gaussian. Least square estimation will be used for the solution. The LSE is applied twice in our implementation—once before applying the ionospheric and tropospheric corrections and one at the end with all the corrections. At the first time, a rough estimate of the receiver's position is obtained. This is for input into the ionospheric corrections. The second time, it is applied with all corrections. This gives the final 3D position of the receiver. First, a measurement matrix is built from partial derivatives of the geometric distance between the satellite and the receiver for each satellite. Next, a pseudoinverse matrix is computed. This matrix is factored with the measurement equation. We obtain the difference in estimation with the previous iteration. The modulus of estimation and number of iterations is checked. This imposes a threshold. Also, it is known that for most epochs, the cost function converges towards zero within 10 iterations and thus we take a random value for the maximum number of iterations (16 in our case). Therefore, this loop runs until the modulus of estimations is greater than 10^{-5} and the number of iterations. This gives us the receiver position. The same explanation is given in the flowchart below:

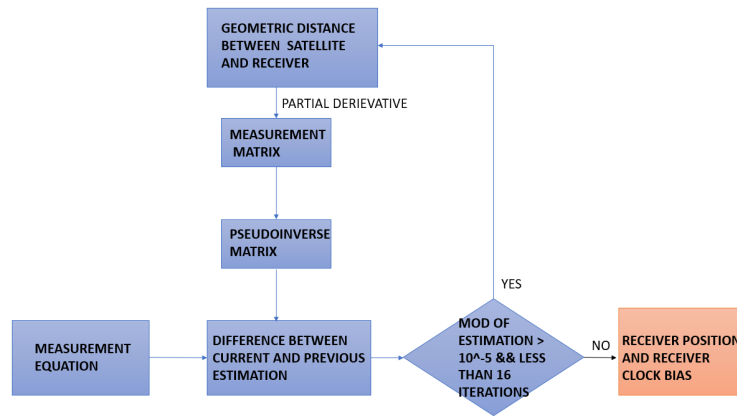


Figure 3.4: Flowchart of LSE implementation

3.4 Tropospheric Correction

The UNB3 model is used for this correction as we are using GPS satellites. The process of ionospheric correction can be given by the following flowchart

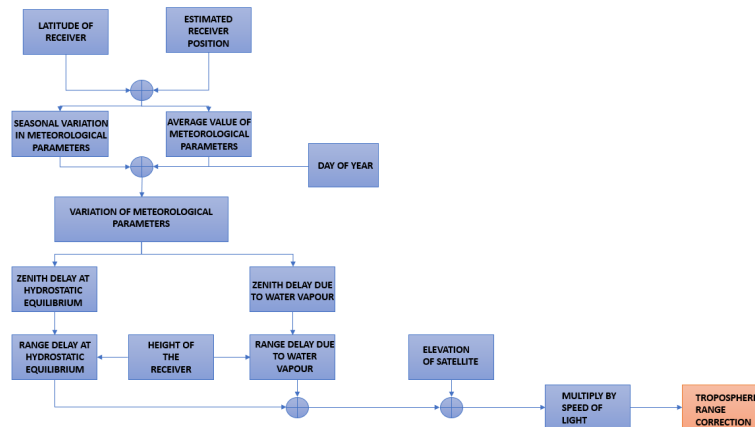


Figure 3.5: Flowchart of tropospheric correction implementation[6]

The parameters needed for tropospheric corrections are the day of the year which can be found from the .nav and .obs files. We need the elevation, latitude and height of satellites and the first estimation of receiver position. For this, we implement the least square estimation module without corrections to find an initial estimate. We compute the delay according to the average and seasonal variations in meteorological parameters. The content of water vapour and hydrostatic equilibrium is important for this correction. These parameters along with the elevation mapping function obtained from elevation calculation gives the tropospheric range correction.

3.5 Ionospheric Correction

The Klobuchar model is used for this correction again as we are using GPS satellites. The process of ionospheric correction can be given by the following flowchart

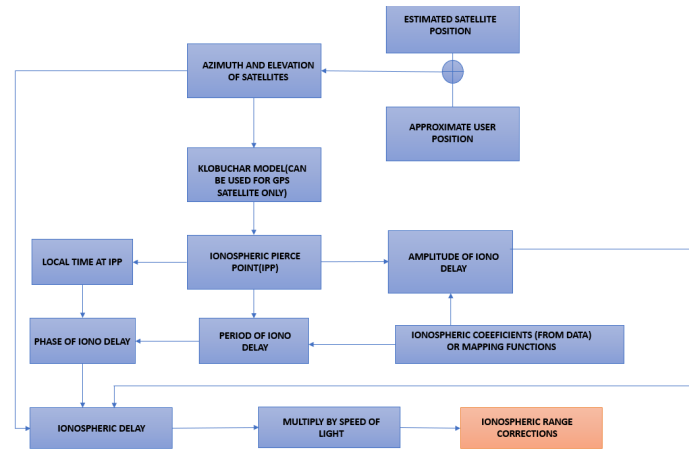


Figure 3.6: Flowchart of ionospheric correction implementation[5]

The most important parameters needed are the elevation and azimuth of the satellites. This module was provided by our tutor. Also, we need the estimated position of the satellites and the first estimate of the receiver as well. Hence, we implement the least square module to find an estimate of the x,y,z coordinates by implementation of least square estimation(without any corrections) like we did for tropospheric corrections. The ionospheric components can be found in the .nav file. These parameters are given by iono alpha and beta parameters. It is based on the principle of a grid based structure of the ionosphere where the signal passing through this grid to reach the receiver on the earth is taken into account. This point of the signal incident on the ionosphere is called IPP or ionospheric pierce point. The amplitude and phase of this ionospheric delay is calculated with the ionospheric coefficients-alpha and beta(obtained from .nav file). Then the slant factor gives the delay of signal through the ionosphere and finally on multiplying with speed of light, we get the ionospheric range correction.

In conclusion, the process of finding the receiver position can be summarized in the following flowchart given below.

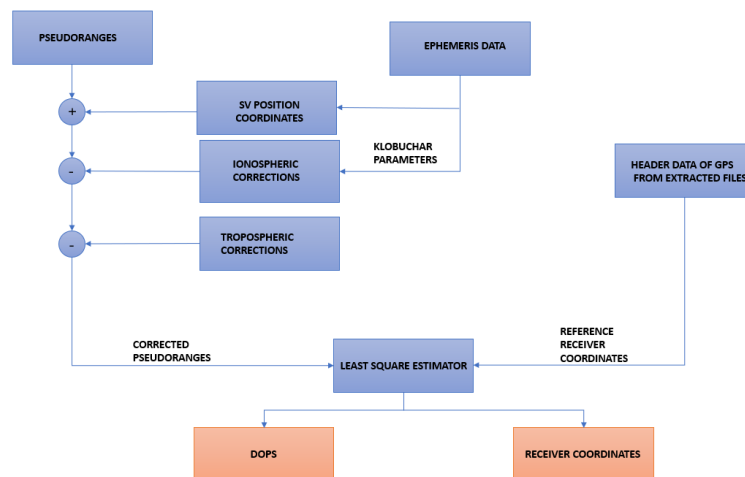


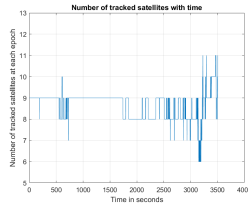
Figure 3.7: Flowchart receiver position computation

Chapter 4

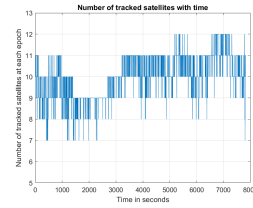
ANALYSIS OF RESULTS

This section presents the results from the implementation of the project and the analysis for these obtained results. As discussed before, two sets of data were collected-one from the balcony of the residence for about two hours and another in the center of the football field.

4.1 Number of visible satellites



(a) Number of satellites for open field data



(b) Number of satellites for terrace data

Figure 4.1: Number of visible satellites evolution with time

1. Open-field data

4.1a shows the number of satellites tracked in an open field over time of about 1 hour or 3500 epochs. The maximum number of satellites tracked was 11 and the minimum was 6. The number of satellites tracked does not vary continuously since data was collected where there is low chances of interference and signal blockage from buildings. However, between 6300-6400 epochs, the receiver is seen to track lower number of satellites.

2. Terrace data

4.1b represents the number of satellites tracked by the receiver over time of 7900 epochs for data taken from the balcony of the residence. The maximum number of satellites tracked was 12 over a time of a little over 2 hours. The minimum tracked satellites is 7 (for example between 400-500 seconds). The number is seen to vary within this range greatly. This can be justified with the environment in which the data was collected. It had a high risk of blocking of signals due to presence of obstacles like buildings. The receiver may lose the signals from satellites (especially low elevation satellites).

4.2 Raw pseudorange measurements

The pseudorange represents the pseudo distance between the satellite and the receiver. It is a function of elevation angle and helps us to understand whether the satellite is rising, setting or is overhead. This is mainly due to the fact earth is spherical. We can say that when the elevation angles decreases, the distance between the satellite and receiver increases.

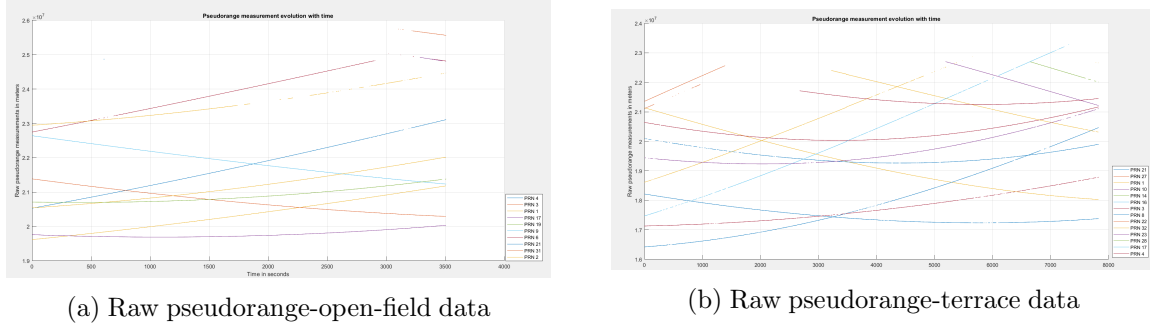


Figure 4.2: Raw pseudorange measurements evolutions with time

1. Open-field data

4.2a represents the raw pseudorange measurement in meters over 3500 epochs. As some of the lines seem to increase, it represents the satellites which are going away from the receiver or we can say those satellites are setting. The lines that are decreasing show the satellites that are coming into the view of the receiver. The satellites with lower values on this graph show that the satellites are overhead at that time epoch. Since the time of data collection is shorter, we see shorter curves and mostly curves that are increasing. At the end of the curve, two new satellites are seen to approach the receiver which can be observed from the two short lines at the end of the graph.

2. Terrace data

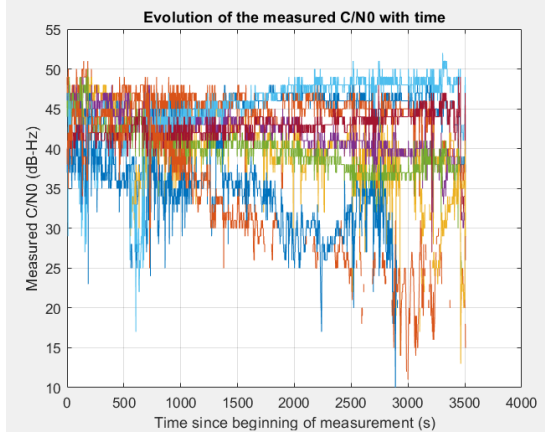
4.2b represents the raw pseudorange measurement over 7900 epochs. The raw pseudorange measurements are given in meters. Since the data collection is longer, we see longer curves. Also, the number of satellites rising and setting are more prominent in this graph.

4.3 Signal-to-noise ratio

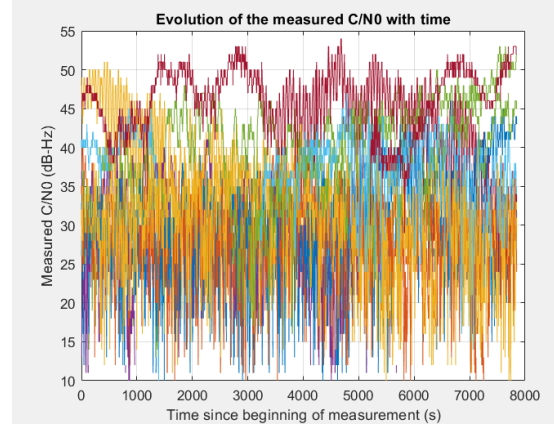
The graphs represent the ratio between the power of the signal (generally denoted C) and the spectral density level of the noise (N_0) for the two types of data collected. The variations represent the strength of the signal received from different satellites. It can be affected by such as geometric range, elevation of satellite, presence of obstacles, receiver and antenna gain. The effects due to receiver may be neglected due to as we use same receiver for all the measurements. A high value of CNR is desired and they provide good position solution. It should also be noted that high elevation satellites are less affected by multipath and their signal to noise ratio is greater.

1. Open-field data

4.3a represents the signal to noise ratio in dBHz over 3500 epochs. For the open field data, the plot is seen to have much less variations over time. This is because data was collected in an environment that did not have much obstacles. The CNR varies between 25 to 50 dBHz till 2800 epochs and the CNR variation drops to 10 dBHz after 2800 epochs. The last part however shows this drop and greater variation. This maybe due to tracking of an unhealthy satellite or satellites at low elevation angles.



(a) SNR measurements for open-field data



(b) SNR measurements for terrace data

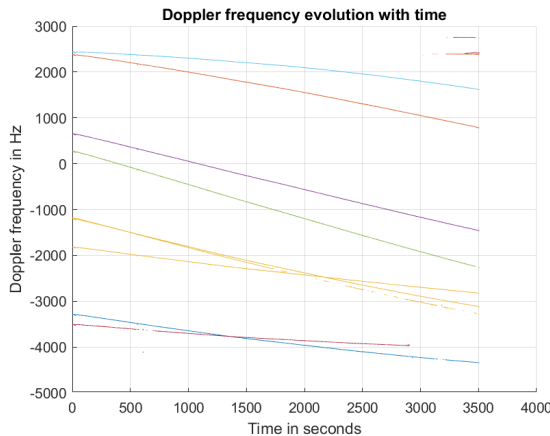
Figure 4.3: SNR measurements evolution with time

2. Terrace data

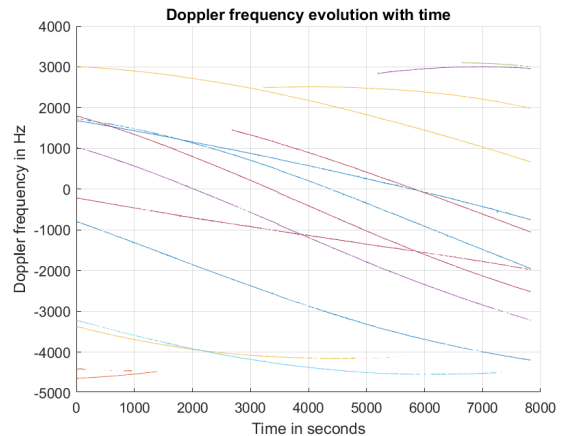
4.3b represents the signal to noise ratio in dBHz over 7900 epochs. The variation in signal to noise ratio for terrace data is quite large. This can be justified by interference and blockage from obstacles arising from the environment in which data was collected. The value varies from 10 to 50 dBHz.

4.4 Doppler frequency

Doppler frequency represents the change in frequency as an object moves towards or away from another object. The satellites are moving over time and are coming towards the receiver and then away from it. In the two graphs presented below, we can see that the change in frequency can be positive or negative. When the satellite is moving closer to the receiver, a positive value of Doppler frequency is seen and a negative value is seen when these satellites are moving away from the receiver. It can also be observed that the frequencies are positive first and then it becomes negative. This is because the satellites move towards and away from the receiver. The frequency is zero when satellite is overhead.



(a) Doppler frequency for open-field data



(b) Doppler frequency for terrace data

Figure 4.4: Doppler frequency measurements evolution with time

1. Open-field data

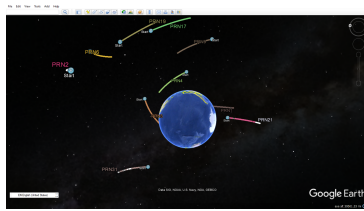
4.4a represents the doppler frequency in Hz over 3500 epochs. For the open field data, the values vary between -4200 Hz to 2800 Hz.

2. Terrace data

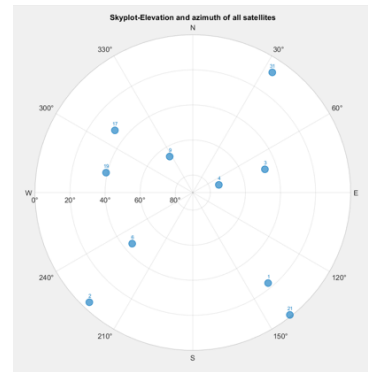
4.4b represents the doppler frequency in Hz over 7900 epochs. The values vary between -4600 Hz to 3000 Hz in the more challenging environment.

4.5 Satellite vehicle position

The satellite positions computed in x,y and z coordinates were converted into latitude, longitude and altitude. The position of these satellites was plotted using google maps and Matlab and the results were obtained to visualize the estimated trajectory of all the tracked satellites. The second plot is a sky-plot of the azimuth and elevation of all the satellites. The range for azimuth value is 0 to 360 degrees and for elevation is 0 to 90 degrees. It shows the position of satellites with respect to the receiver and hence helps us to visualize the geometry of the satellites. For the open field data, we see less number of satellites tracked as compared to the terrace data. This is because the open field data was collected for shorter duration of time.

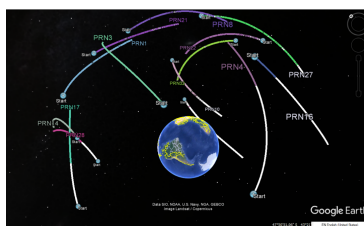


(a) Satellite position on google earth view

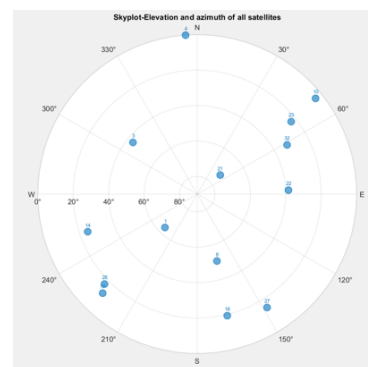


(b) Elevation and azimuth of satellites

Figure 4.5: Satellite position for open field data



(a) Satellite position on google earth view



(b) Elevation and azimuth of satellites

Figure 4.6: Satellite position for terrace data

4.6 Satellite Clock correction

The satellite clock error is represented in the figures below. Clock errors of the satellite contributes to the maximum error in range. This is proved from the results represented in the figures. For both the data, the range error can reach a value upto 180km(as seen from the graph). Thus, this correction is very important in PVT computation. Clock range error is not dependent on the geometry of the satellites. This error is due mainly due the satellite oscillator not being synchronized to the GPS time. The broadcast navigation message provides the error as a few coefficients and is updated every two hours.

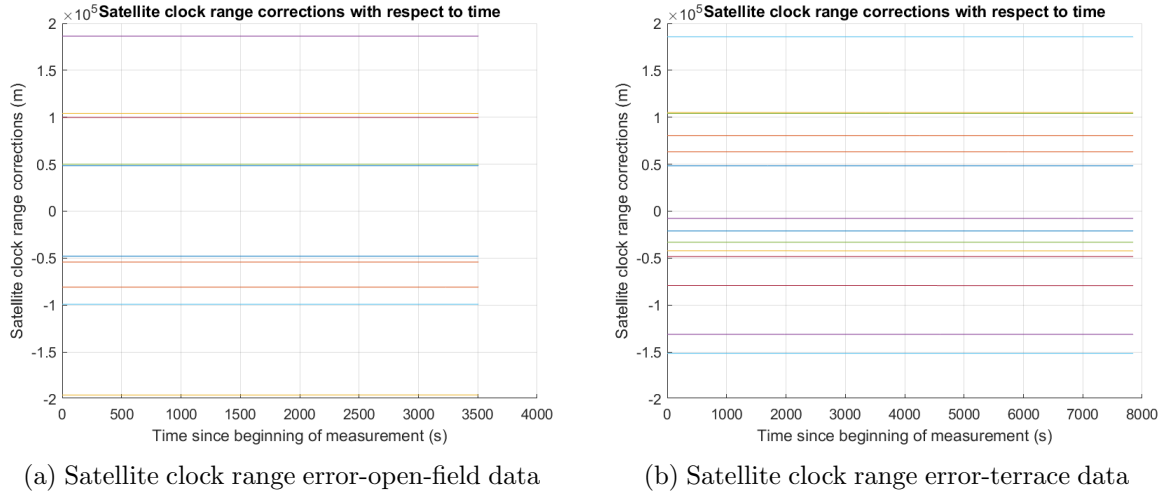


Figure 4.7: Satellite clock range correction measurements evolution with time

1. Open-field data

4.7a represents the satellite range error in meters over 3500 epochs. For the open field data, the values vary between -200km to 180 km.

2. Terrace data

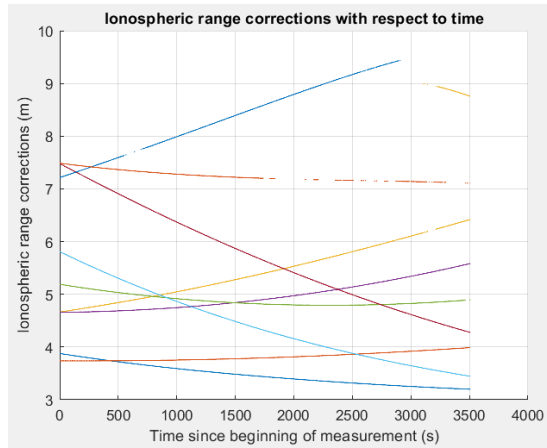
4.7b represents the satellite range error in meters over 7900 epochs. The values vary between -150km to 180km in the more challenging environment. The values are more or less similar in both cases, as it depends on the satellite's clock and not the environment.

4.7 Ionospheric range correction

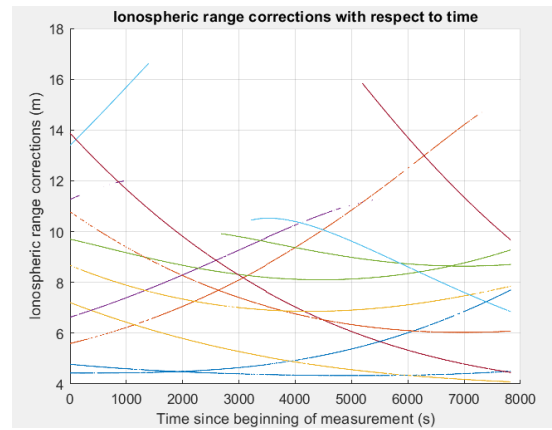
The figures given below represent the error in measurements due to the presence of ionospheric which gives rise to ionospheric range error. The ionosphere is a layer of the atmosphere that extends from 60-1000km above the earth's surface. The presence of charged particles(ions) interfere with the signal and gives rise to errors. The ionospheric errors are a functions of the elevation of the satellites. The satellites that have high elevation(overhead and near the receiver) have lower ionospheric range errors than the ones having low elevation. The increasing lines represent the setting of satellites and the decreasing lines represents their rising. The errors are of the orders tens of meters and hence have to be taken into account if we want a precise position.

1. Open-field data

4.8a represents the ionospheric range error in meters over 3500 epochs. For the open-field data varies from upto 9.4m. The error is lesser than the error in terrace data because the



(a) Ionospheric range error-open-field data



(b) Ionospheric range error-terrace data

Figure 4.8: Ionospheric range error measurements evolution with time

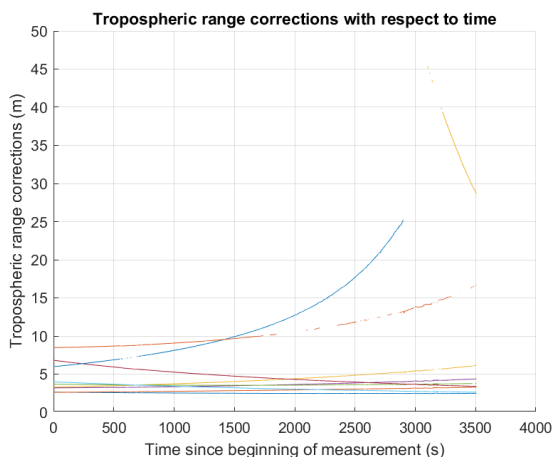
measurements were taken in the evening. In the evening the ionospheric noise is lesser because there the brightness or intensity of the sun decreases.

2. Terrace data

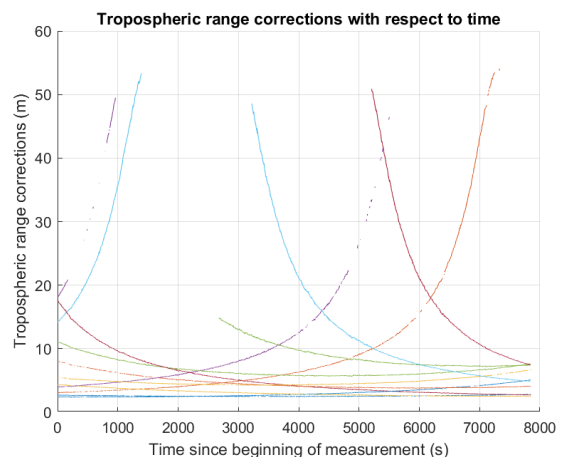
4.8b represents the ionospheric range error in meters over 7900 epochs. The range error varies between upto 16.5 m. in the more challenging environment. This data was collected in the afternoon and thus the ionospheric error is larger.

4.8 Tropospheric range Correction

The figures given below represent the error in measurements due to the presence of tropospheric which gives rise to tropospheric range error. The troposphere is a layer of the atmosphere that extends from above the earth's surface for about 15km. The errors are mainly due to presence of clouds, water vapour and gases. The satellites with low error are the satellites that are overhead and the errors that are increasing represent setting satellites.



(a) Tropospheric range error-open-field data



(b) Ionospheric range error-terrace data

Figure 4.9: Tropospheric range error measurements evolution with time

1. Open-field data

4.9a represents the tropospheric range error in meters over 3500 epochs. For the open-field data varies from upto 10m approximately for overhead satellites. Since data was collected for a short duration of time, we see that most errors are constant and do not increase. We however see the error rising to 25m at 2900 seconds for one satellite(indicating a setting satellite) and 40m for the satellite that appears at the end(a satellite that is rising).

2. Terrace data

4.9b represents the tropospheric range error in meters over 7900 epochs. The range error varies between upto 54 m in the more challenging environment. Approximately for most of the satellite that are overhead, the errors are seen to be below 10m. The curves represent the rising and setting of satellite and how it affects tropospheric error.

4.9 Receiver position

4.9.1 Receiver position in x,y,z coordinates

For the data which was collected from the center of the football field at ENAC, the 3D coordinates are $4.627644350154365e+06$, $1.199482478024442e+05$ and $4.373195722401632e+06$.

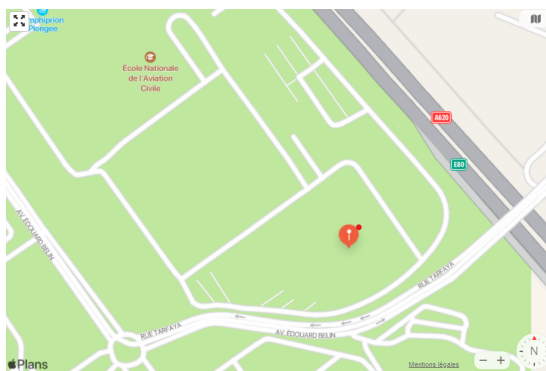
For the data which was collected from the terrace of the residence, the 3D coordinates are $4.627559781477445e+06$, $1.197694799742799e+05$ and $4.373306728568713e+06$.

4.9.2 Receiver position on map using latitude and longitude

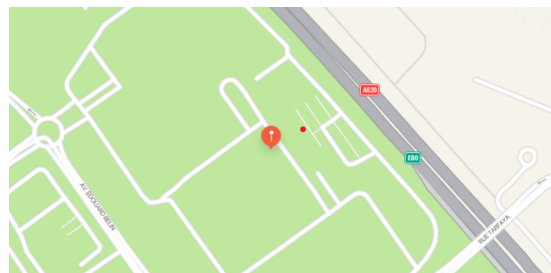
For the data which was collected from the center of the football field at ENAC, the latitude is 43.563248704525020 degrees north and longitude is 1.484773194474531 degrees east.

For the data which was collected from the terrace of the residence, the latitude is 43.564527282533630 degrees north and longitude is 1.482576182312251 degrees east.

The receiver position in latitude and longitude were plotting using GPS coordinates website[3] and the results are presented below. We observe drift from actual position of the receiver from which the measurement was taken and thus, the position errors will be discussed in the following sections.



(a) Receiver position from implementation-Open field



(b) Receiver position from implementation-Terrace

Figure 4.10: Receiver position from implementation on map

4.10 Dilution of precision

DOP represents the geometry of satellites visible to the receiver. It provides relation between the measurement or position errors. The values varies as satellites change their position, orientation and their number with respect to the receiver. The ideal value is less than one and a high DOP denotes bad geometry.

4.10.1 GDOP

The geometric dilution of precision affects the position error estimation. A good DOP represents good geometry meaning the satellites are spread around the receiver and the value will be small. On the other hand, a bad DOP represents bad geometry meaning the satellites are close to one another or not well spread or lies on one side and will have a large value. The GDOP is divided into PDOP(position DOP) and TDOP(time DOP). PDOP has two components VDOP(vertical DOP) and HDOP(horizontal DOP). For the open field data, the GDOP is between 1.8 to 5.3 which is a moderate DOP and will affect the position error moderately. For terrace data, the GDOP is between 1.1 to 2.7 which is excellent DOP value again showing it will not affect position error. This can be justified as the terrace data was collected for a longer period of time(2 hours)

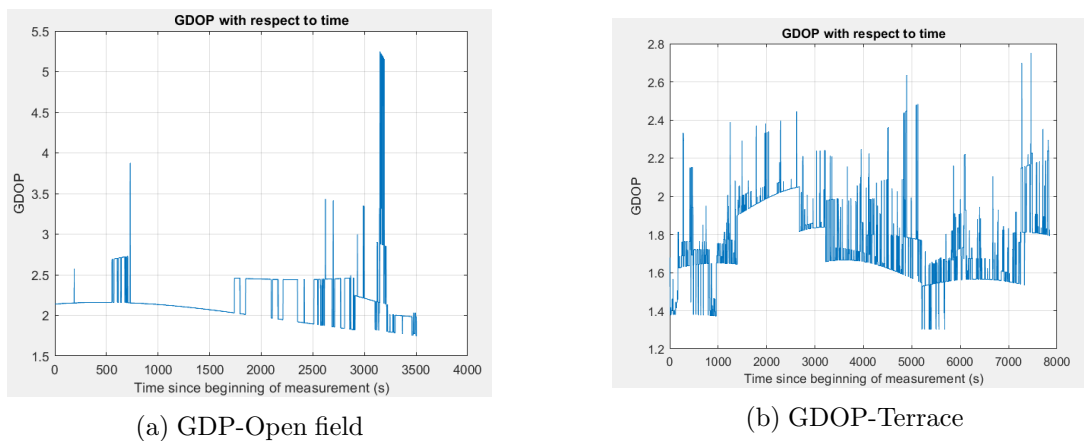


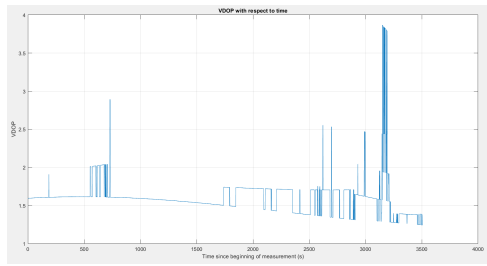
Figure 4.11: GDOP evolution with time

4.10.2 VDOP

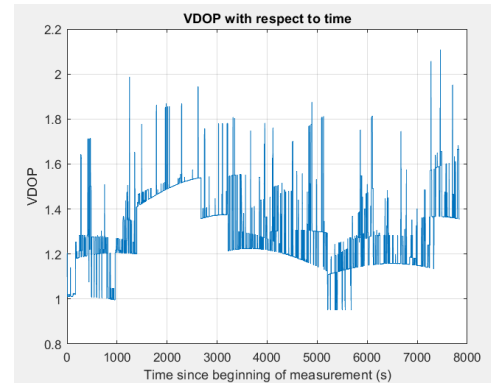
4.12 represents the vertical dilution of precision affects the vertical position estimation. The lower the value, the better the geometry and lower the position error. For the open field data, the VDOP is between 1.3 to 3.7 which is a good DOP and wont affect the position error much. For terrace data, the VDOP is between 1 to 2.1 which is excellent DOP value again showing it will not affect position error.

4.10.3 HDOP

4.13 represents the horizontal dilution of precision affects the horizontal position error. The lower the value, the better the geometry. For the open field data, the HDOP is between 0.95 to 1.73 which is an excellent DOP value and wont affect the position error. For terrace data, the HDOP is between 0.7 to 1.23 which is excellent DOP value again showing it will not affect position error.

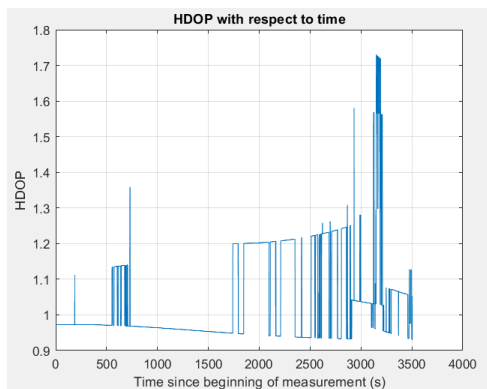


(a) VDOP-Open field

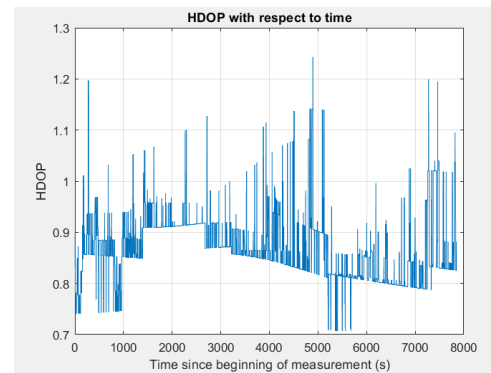


(b) VDOP-Terrace

Figure 4.12: VDOP evolution with time



(a) HDOP-Open field



(b) HDOP-Terrace

Figure 4.13: HDOP evolution with time

4.11 Estimation Errors

The estimation error has been calculated for each coordinate with the difference between the calculated results (from implementation) and the reference position (from the .obs file/observation data). The results explain the quality of the position estimate and the errors with respect to the reference position.

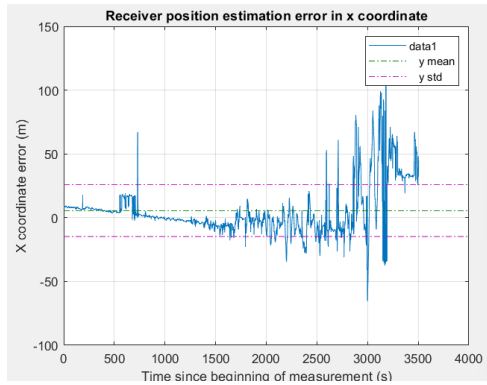
4.11.1 X-coordinate estimated position error

1. Open-field data

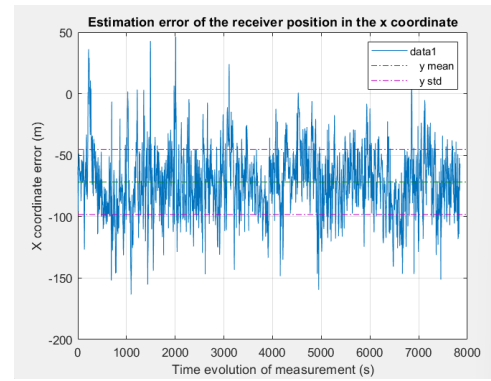
4.14a denotes the x-position error of the open field data. As we can see from the plot, the mean is -7.323 and standard deviation is 18.38m. The value ranges between -25m to 11m. This denotes poor position estimation in x coordinate because the probability of having our estimate equal to the true x position is very low.

2. Terrace data

4.14b denotes the x-position error of the terrace data. The mean is -71.88 and standard deviation is 26.44m. The value ranges between -105m to -45m. This denotes very poor position estimation in x coordinate because the probability of having our estimate equal to the true x position is very low. The challenging environment in which the data was taken is the reason for having such large errors.



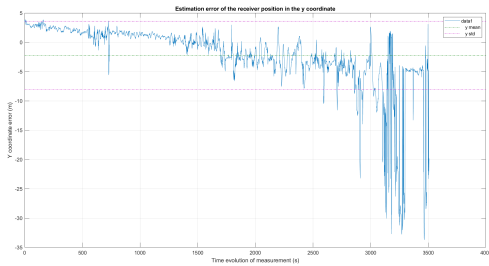
(a) X-coordinate estimation error-Open field



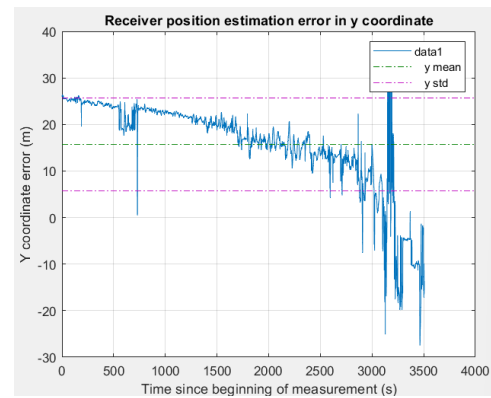
(b) X-coordinate estimation error-Terrace

Figure 4.14: X-coordinate estimated position error at all epochs

4.11.2 Y-coordinate estimated position error



(a) Y-coordinate estimation error-Open field



(b) Y-coordinate estimation error-Terrace

Figure 4.15: Y-coordinate estimated position error at all epochs

1. Open-field data

4.15a denotes the y-position error of the open field data. As we can see from the plot, the mean is -2.227m and standard deviation is 5.809m. The value ranges between -7m to 3m. This denotes poor position estimation in y coordinate because the probability of having our estimate equal to the true y position is very low. However, the y-error is lower than that observed in the x-coordinate error.

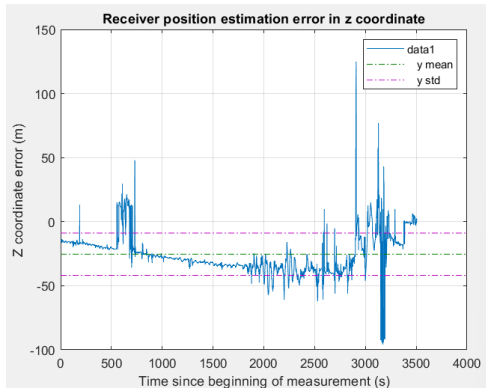
2. Terrace data

4.15b denotes the y-position error of the terrace data. The mean is 11.8m and standard deviation is 13.85m. The value ranges between -2m to 24m. This denotes poor position estimation in y coordinate because the probability of having our estimate equal to the true y position is very low. Again, the y-error is lesser than that observed in the x-coordinate error. The error is much higher due to the challenging environment where the signals encounter multipath and blockage.

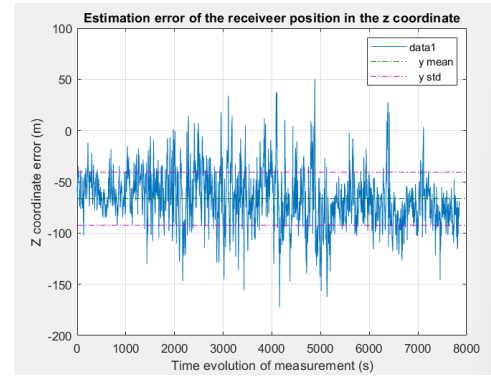
4.11.3 Z-coordinate estimated position error

1. Open-field data

4.16a denotes the z-position error of the open field data. As we can see from the plot, the



(a) Z-coordinate estimation error-Open field



(b) Z-coordinate estimation error-Terrace

Figure 4.16: Z-coordinate estimated position error at all epochs

mean is -11.07m and standard deviation is 10.09m. The value ranges between -7m to 3m. This denotes poor position estimation in z coordinate because the probability of having our estimate equal to the true z position is very low.

2. Terrace data

4.16b denotes the z-position error of the terrace data. The mean is -66.47 and the standard deviation is 25.83. The value ranges between -45m to -92m. This denotes poor position estimation in z coordinate because the probability of having our estimate equal to the true z position is very low. The more challenging environment in this case produces more errors due to multipath than the data collected in open field.

4.11.4 Estimated receiver position on map



(a) Estimated receiver position-Open field



(b) Estimated receiver position-Terrace

Figure 4.17: 3D estimated position of receiver

1. Open-field data

4.17a the estimated receiver positions that have been computed for all the epochs. It is observed that with all the errors the position is drifted from the center of the football field where the data was actually taken. X error: 7.323m, Y error: 2.227m and Z error: 11.07m.

2. Terrace data

4.17b the estimated receiver positions that have been computed for all the epochs. It is observed that with all the errors the position is drifted from the terrace of Louis Bleriot residence where the data was actually taken. X error: 71.88m, Y error: 13.85m and Z error: 66.47m. The estimated position is shown at Z block(ziegler) and the errors are upto 72m.

It should also be noted that the plots have been taken with google earth which has an error by default of 1-3m which is cumulated with the results presented in the 3D plots. Therefore, the 2D plots of individual x,y,z errors are more reliable.

4.12 Code-Minus-Carrier

The CMC plot includes the twice the ionosphere error, multipath, noise, and ambiguity in the pseudorange of each satellite. It gives a measure of how they affect our measurement. The ambiguity term is continuous and can be eliminated using a mean estimator. However, the other terms have to be accounted for. For both the plots, we see that the values are centered around zero. There are variations mainly due to the ionosphere and the multipath.

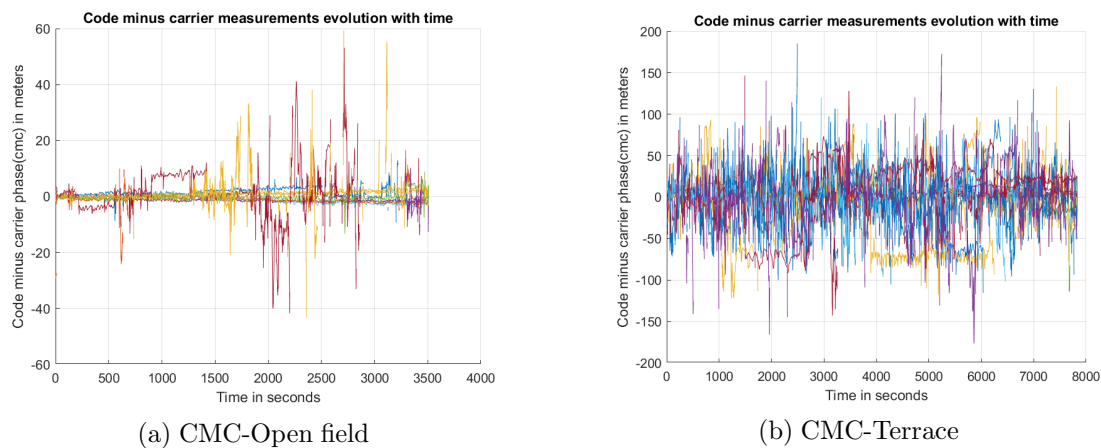


Figure 4.18: Code minus carrier evolution with time

1. Open-field data

4.18a shows that the CMC ranges between -40 to 60m. The values have a mean of zero but slight variations are seen for some satellites. Since our environment did not have many obstacles, we can say these errors are due to ionosphere or noise. Ionospheric errors can be either due to the solar activity at the time data was collected, due to low elevation or unhealthy or loosely tracked satellites.

2. Terrace data

4.18b shows that the CMC value ranges between -170m to 190m and is centered about zero. This data has been taken in the afternoon when solar activity is at its peak and hence shows larger values than that of the open-sky. Also, due to the challenging environment, we see the value is greater because it is more affected by multipath and blockage of signals due to buildings.

Chapter 5

CONCLUSION

The algorithm for the estimation of receiver position has been implemented successfully. Results from two datasets have been presented in the report to account for position estimation errors in different environments. An error up to 11m has been observed for data collected on an open football field. On the other hand, the error increases drastically up to 72m in the case of data collected in the building. Thus, we observe how much effect multipath may have on position errors.

Corrections such as satellite clock error, ionospheric error, and tropospheric errors have been accounted for in the algorithm. It is observed that satellite clock errors are responsible for most of the errors. For ionospheric correction, the Klobuchar model has been implemented and for tropospheric correction, UNB3 model has been used. The ionospheric and tropospheric errors are of the magnitude of a few tens of meters. The least square estimation is used for receiver position computation and the DOPs. DOPs for both the data sets are good and thus, they do not affect the position error much.

Finally, better results may be possible to achieve by changes in the algorithm like discarding satellites that are loosely tracked or are unhealthy. Adopting a model for noise and multipath may also reduce errors in a challenging environment.

Bibliography

- [1] <https://www.raspberryme.com/utiliser-u-center-pour-se-connecter-au-gps-sur-raspberry-pi/>
- [2] <https://www.coordonnees-gps.fr/>
- [3] <https://earth.google.com/web/@0,0,0a,22251752.77375655d,35y,0h,0t,0r>
- [4] <https://www.everythingrf.com/community/what-are-the-different-gnss-constellations?>
- [5] https://gssc.esa.int/navipedia/index.php/Klobuchar_Ionospheric_Model
- [6] https://gssc.esa.int/navipedia/index.php/Tropospheric_Delay