

POLITECNICO DI TORINO



ICT for Geomatics: Navigation and Maps

Labs Report



Professor:

Fabio Dovis

Marco Piras

Student:

Lorenzo Bellone

A.Y.2018-2019

Contents

1	Lab1: Evaluation of the PVT solution	2
1.1	Introduction	3
1.2	Recursive Least Mean Square	3
1.3	The Dataset	3
1.4	Task A-1	4
1.5	Task A-2	5
1.6	Task A-3	7
1.7	Task A-4	9
2	Lab 2: Computation of PVT using real data	13
2.1	Introduction	14
2.2	Task B-2	14
2.3	Task B-3	15
2.4	Task B-4	16
	Bibliography	19

Chapter 1

Lab1: Evaluation of the PVT solution

1.1 Introduction

In the first lab it is required to evaluate the position of an user given two datasets: the first one containing the pseudoranges evaluated between the user and a specific satellite, the second one containing the true position of the satellites. Moreover, the data are provided for different GNSS constellations: GPS, Galileo, GLONASS and BeiDou.

Given these types of data, we already know that the PVT (position, velocity and time) solution [1] can be computed with one simple equation 1.1, called Least Mean Square (LMS):

$$\Delta x = (H^T H)^{-1} H^T \Delta \rho \quad (1.1)$$

In equation 1.1, Δx represents the difference between the true position of the user and an approximated position and $\Delta \rho$ is the difference between the approximated pseudorange (geometrical distance between the approximated position and the satellite) and the real one. We also know how to compute the matrix H :

$$H = \begin{matrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 \\ a_{x,2} & a_{y,2} & a_{z,2} & 1 \\ a_{x,3} & a_{y,3} & a_{z,3} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,J} & a_{y,J} & a_{z,J} & 1 \end{matrix} \quad (1.2)$$

The coefficients of matrix H , $a_{x,j}$, are unitary vectors steering from the approximation point towards the position of the j^{th} satellite.

1.2 Recursive Least Mean Square

One possible way to obtain a more precise PVT solution is to apply the LMS not just once, but in an iterative algorithm. At each iteration the approximated point is updated with the solution obtained at the previous step. Typically, this algorithm converges within ten iterations.

Given a time instant n and the value of the measured pseudoranges at that time instant, ρ_n , the algorithm follows the steps below:

1. Initialization of the approximation point $\hat{x}_n^0 = (0, 0, 0, 0)$.
2. Evaluation of the difference between the range vector geometrically measured from the current position x_n^k and true position of the satellite and the vector of measured pseudoranges $\Delta \rho_n^k = \hat{\rho}_n^k - \rho_n$.
3. Computation of the k^{th} PVT solution $\Delta \hat{x}_n^k = ((H_n^k)^T H_n^k)^{-1} (H_n^k)^T \Delta \rho_n^k$.
4. Updating of the current position $\hat{x}_n^k = \hat{x}_n^{k-1} + \Delta \hat{x}_n^k$. Go to step 2.

It is important to point out that the matrix H must be updated at each iteration, since the approximated position also changes.

1.3 The Dataset

As already mentioned above, the aim of this lab is to implement the Recursive Least Mean Square algorithm in order to obtain the true position of an user. Matlab (version

R2018b) was used for the implementation of the algorithm.

We were provided with the values of the pseudoranges measured with different satellite constellations (GPS, Galileo, GLONASS, BeiDou) for six locations, together with the true position of the satellites. For each satellite we had several values of the pseudoranges since the data collections included 3600 seconds of satellites observation from a static position.

The number of visible satellites was not constant over the observation time, hence, also the number of rows of the matrix H should be changed at each iteration of the algorithm. The raw measurements of the pseudoranges were already corrected of all the predictable contributions to the error, but after the correction there is still the random contribution to the error, modelled by the User Equivalent Range Error (σ_{UERE}) [2].

For every position, two types of measurements were provided. In the first case (NominalUERE folder), all the measured pseudoranges had the same σ_{UERE} . The second one (RealisticUERE folder) showed a more realistic case, in which the σ_{UERE} is satellite-dependent.

1.4 Task A-1

The first task required to check the satellites visibility at each time instant for all the constellations (Loaded from the NominalUERE folder). In the provided data structure, every time a satellite is not visible, the corresponding value of pseudorange is a NaN (not a number). Figure 1.1, 1.2 and 1.3 show some examples of plots where the number of visible satellites and the measured pseudoranges are displayed along the time.

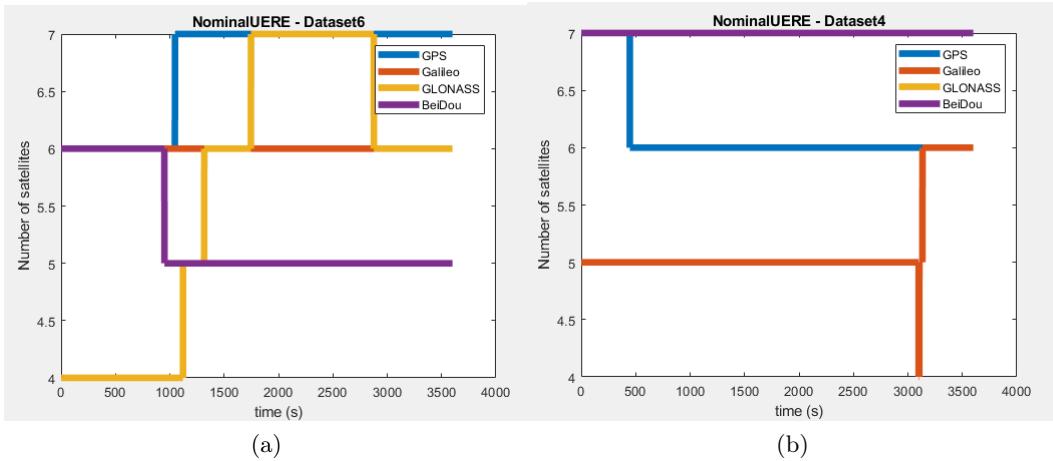


Figure 1.1: On the left (a): number of visible satellites along the time for the dataset6 from NominalUERE folder.
On the right (b): number of visible satellites along the time for the dataset4 from NominalUERE folder.

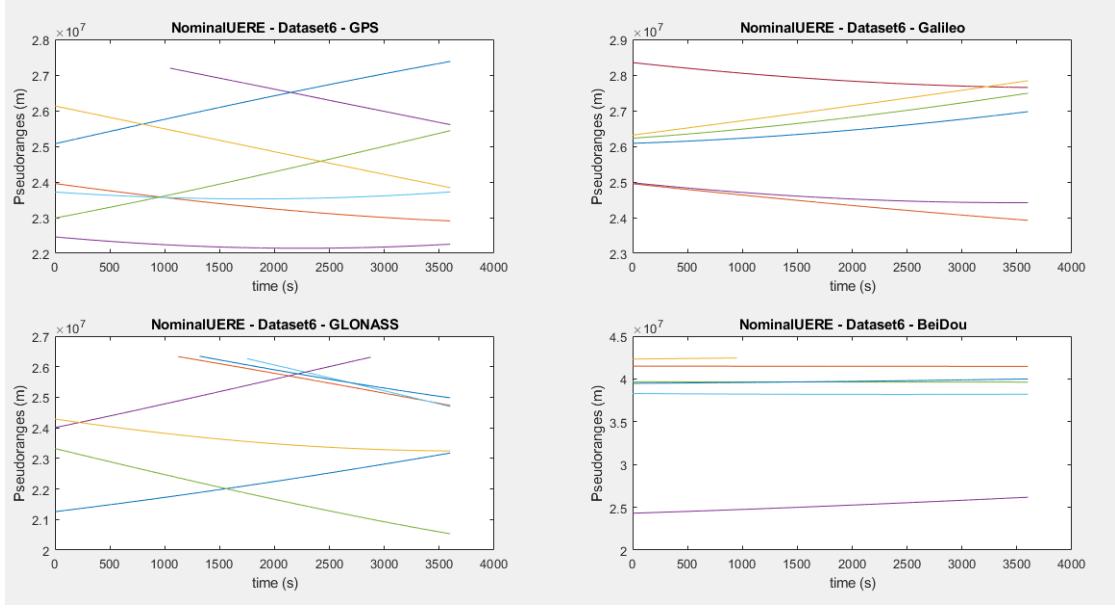


Figure 1.2: Measured pseudoranges along the time for dataset 6 from NominalUERE folder.

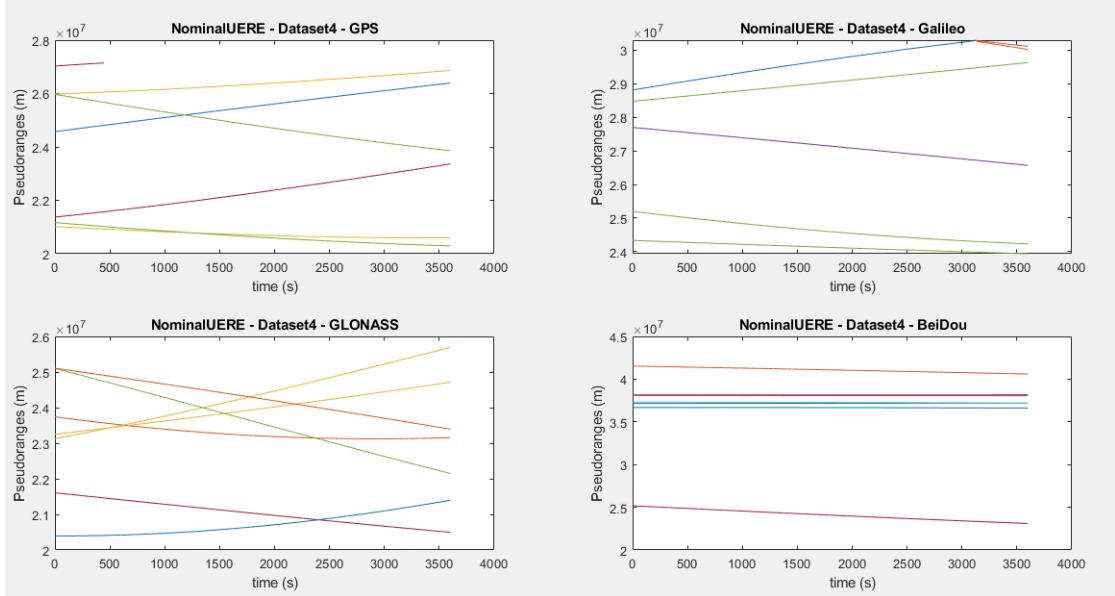


Figure 1.3: Measured pseudoranges along the time for dataset 4 from NominalUERE folder.

It is important to observe that GPS and Galileo constellations provide the highest and most stable number of visible satellites for dataset 6 while BeiDou and GLONASS constellations provide the same quality of service for dataset 4. This is a first indication of the possible datasets' localizations.

1.5 Task A-2

In the second task, we had to implement the Recursive LMS algorithm described above with the use of Matlab. At each time instant a position x_n is obtained. Then, thanks to the provided function “ecef2lla.m”, the ECEF coordinates (Earth-centred, Earth-fixed) are converted into LLA coordinates (Latitude-Longitude-Altitude). The obtained coordinates are passed to the function “writeKML_GoogleEarth.m” in order to create a KML

file, with which it is possible to verify the final position on Google Earth.



Figure 1.4: Estimation of the user state at each time instant. Dataset1 from NominalUERE folder. Location: Cape Town. Constellation: BeiDou.

In figure 1.4 it is quite easy to see that, even if all the predictable contributions to the errors were already been corrected, there is still a contribution to the position error due to the σ_{UERE} that can not be neglected. Other examples are shown in figure 1.5.



Figure 1.5: Estimation of the user state at each time instant. On the left (a): Dataset6 from NominalUERE folder, location: Turin. On the right (b): Dataset4 from NominalUERE folder, location: Shanghai. Constellation: GPS.

At the end of the second task, all the six provided datasets were identified with the corresponding city and with a more or less precise estimation of the user state, as in the previous examples:

Dataset	Location
dataset_1_20180328T122038	Cape Town (South Africa)
dataset_2_20180328T122158	Helsinki (Finland)
dataset_3_20180328T121914	Longyearbyen (Svalbard Islands)
dataset_4_20180328T121804	Shanghai (China)
dataset_5_20180328T121529	Stanford (California)
dataset_6_20180328T121701	Turin (Italy)

Table 1.1

1.6 Task A-3

In the third task, the standard deviation of the position error had to be computed over the time. In the case in which the pseudorange measurements are assumed as an ergodic and stationary process, the position error strictly depends on two quantities: the Geometric Dilution of Precision (GDOP) and the σ_{UERE} . When σ_{UERE} is not satellite-dependent, as in the case of NominalUERE dataset, the standard deviation of the position error can be obtained through the equation 1.3.

$$\sigma_x = \sqrt{\sigma_{Xu}^2 + \sigma_{Yu}^2 + \sigma_{zu}^2 + \sigma_{but}^2} = GDOP \cdot \sigma_{UERE} \quad (1.3)$$

While σ_{UERE} represents the residual error contribution to the pseudorange measurements, the GDOP only depends on the position of the satellites with respect to the user state (equation 1.4).

$$GDOP = \sqrt{\text{tr} \left\{ (\mathbf{H}^T \mathbf{H})^{-1} \right\}} \quad (1.4)$$

The direct calculation of the σ_{UERE} is not possible since multiple realizations of the pseudoranges for the same time instant are not available. However, supposing that the pseudorange measurements can be considered as an ergodic random process, we can remove their deterministic variation along the time taking the second derivative of the process. According to this approach, we will obtain a new process with zero mean, whose standard deviation can be thought as an approximation of the σ_{UERE} .

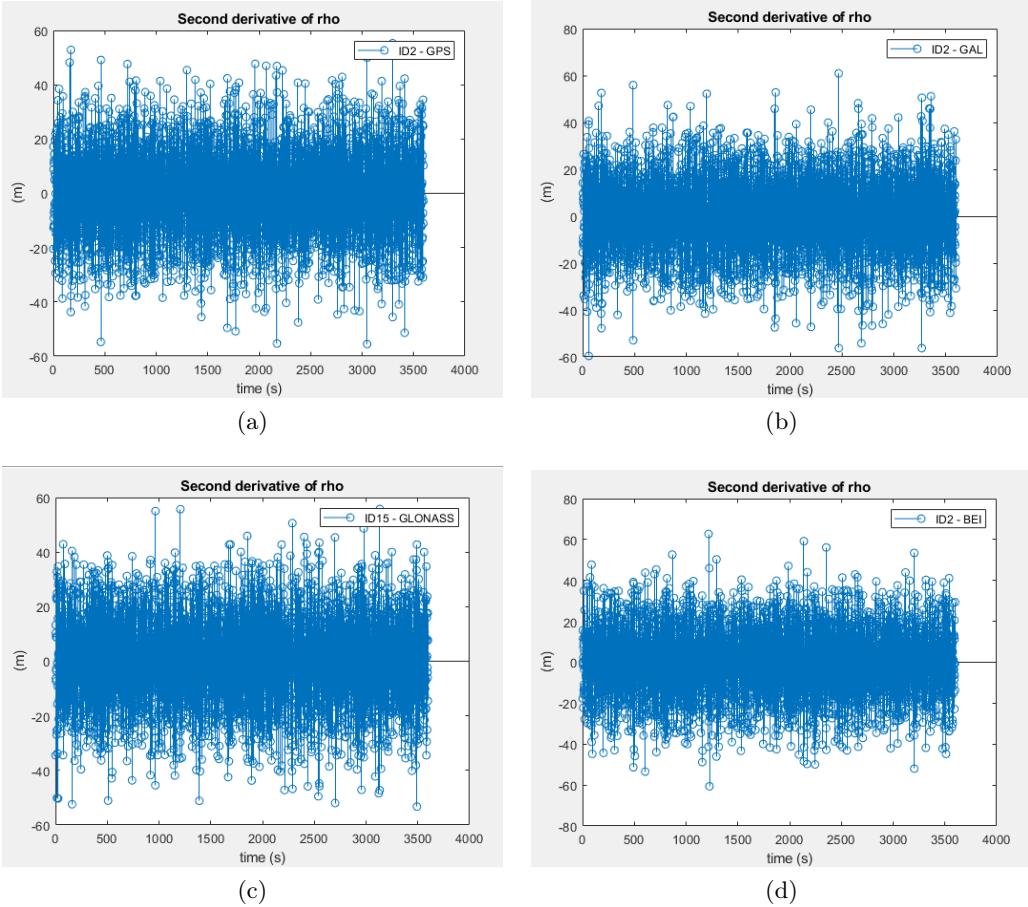


Figure 1.6: $d^2\rho/dt^2$ evaluated from the same dataset (dataset 6) for the four different constellations. GPS on the top left, Galileo on the top right, GLONASS in the lower left, BeiDou in the lower right.

Figure 1.6 shows four examples of the second derivative applied on the measured pseudoranges. For each satellite constellation just one satellite was considered. This operation was also performed with all the other visible satellites and datasets.

After this step, the standard deviation of the position error can be computed in two ways: in the theoretical case we evaluate the σ_{UERE} as the standard deviation of the processes shown in figure 1.4, and then, we multiply it by the GDOP. In the real assumption, we have to compute the standard deviation on the obtained positions over time along the x, y and z directions. Thus, we can just have a single value for the σ_x , while one value for each time instant is computed when equation 1.3 is applied. In figure 1.7 a comparison between the ideal computation of the error (in red) and the real computation (in blue) is shown. The two values are very different, but this is due to the strong assumption on the pseudorange measurements.

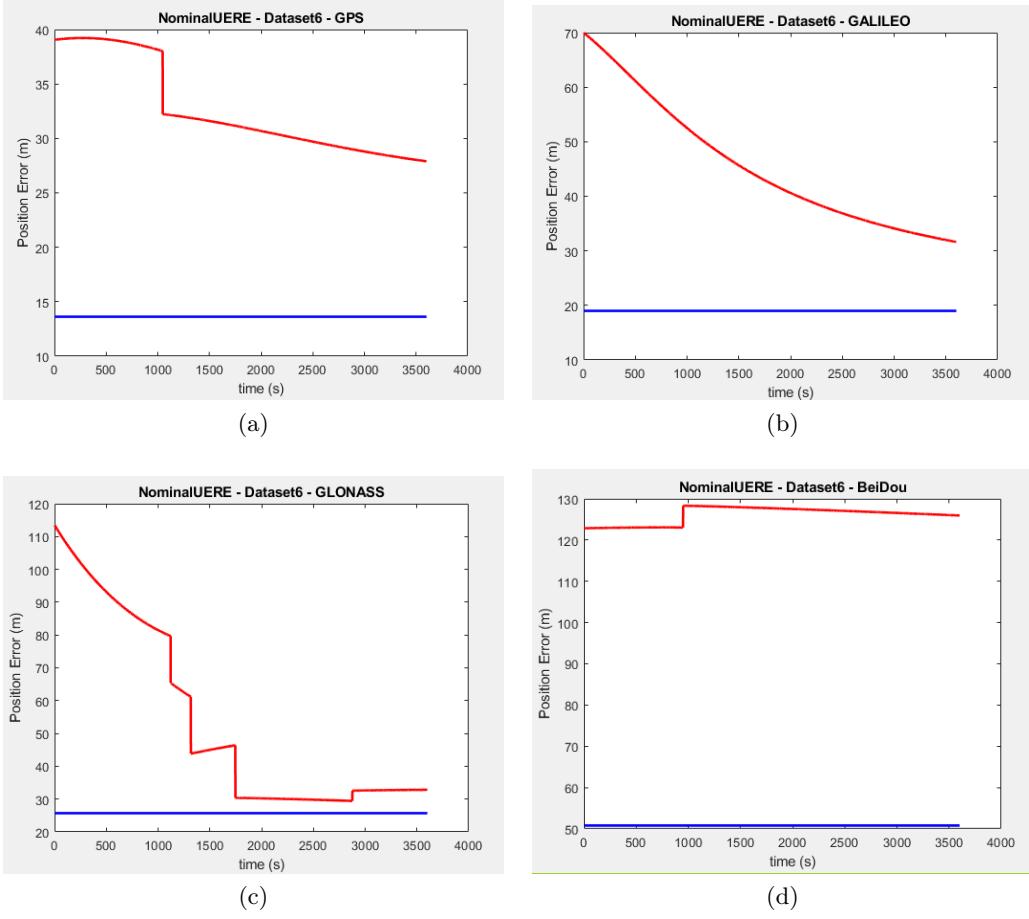


Figure 1.7: Position Error over the time. Same dataset and constellations of figure 1.4

The quality of the obtained solutions is directly observable from figure 1.7. On the Y-labels we have the quantity of uncertainty related to the user state. When the theoretical error evaluation (red line) sharply changes, it means that a new satellite has become visible (if the uncertainty decreases) or that a visible satellite has gone away (if the uncertainty increases). Instead, the slow variations of the position error are due to the movement of the satellites around the earth over the time instants.

The real position error is constant, since it is the standard deviation of all the user positions evaluated over time. This value gives us the real uncertainty on the user state in terms of meters, that is to say, the range in which the 68% of user positions are localized.

1.7 Task A-4

In real cases, the measured pseudoranges can be characterized by different values of the σ_{UERE} , according to the considered satellite. The covariance matrix of the pseudorange errors (\mathbf{R}) is still diagonal, but each entry is characterized by a different value.

In this case, a more accurate algorithm can be applied to obtain the PVT solution. Some measurements may be known to be more accurate than others, so we can give more relevance to the pseudoranges with a small variance. This approach is called Weighted Least Square (WLMS) algorithm, characterized by the introduction of a positive definite weighting matrix \mathbf{W} . In order to give less weighting to the most uncertain measurements, \mathbf{W} is set according to equation 1.5.

$$\mathbf{W} = \mathbf{R}^{-1} \quad (1.5)$$

The WLMS algorithm is slightly different from the Recursive LMS:

1. Initialization of the approximation point $\hat{\mathbf{x}}_n^0 = (0, 0, 0, 0)$.
2. Evaluation of the difference between the range vector geometrically measured from the current position \mathbf{x}_n^k and true position of the satellite and the vector of measured pseudoranges $\Delta\rho_n^k = \hat{\rho}_n^k - \rho_n$.
3. Computation of a new weighted geometrical matrix $\bar{\mathbf{H}}_n^k = \left((\mathbf{H}_n^k)^T \mathbf{W} \mathbf{H}_n^k \right)^{-1} (\mathbf{H}_n^k)^T \mathbf{W}$.
4. Computation of the k^{th} PVT solution $\Delta\hat{\mathbf{x}}_n^k = \bar{\mathbf{H}}_n^k \Delta\rho_n^k$.
5. Updating of the current position $\hat{\mathbf{x}}_n^k = \hat{\mathbf{x}}_n^{k-1} + \Delta\hat{\mathbf{x}}_n^k$.

The implementation of the WLMS algorithm was still implemented on Matlab. The results were computed for all the datasets and constellations (exploiting this time the RealisticUERE folder). Only some of the obtained results will be shown, in order to facilitate the comparison between the two different algorithms.

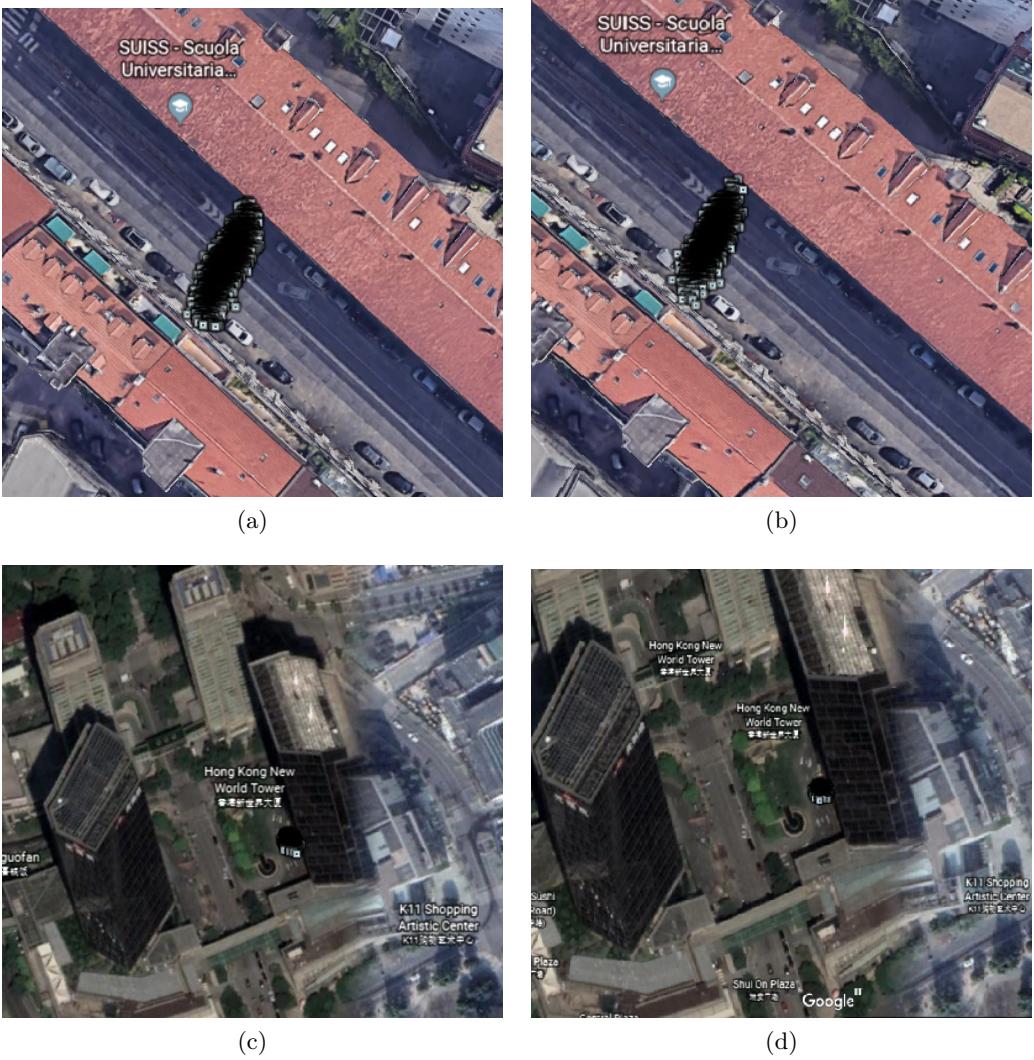


Figure 1.8: Comparison between LMS and WLMS. On the top left: Dataset 6 from RealisticUERE folder processed with LMS algorithm. On the top right: Dataset6 from RealisticUERE folder processed with WLMS algorithm. Constellation: GPS. To the lower left: Dataset 4 from RealisticUERE folder processed with LMS algorithm. To the lower right: Dataset 4 from RealisticUERE folder processed with WLMS algorithm. Constellation: BeiDou.

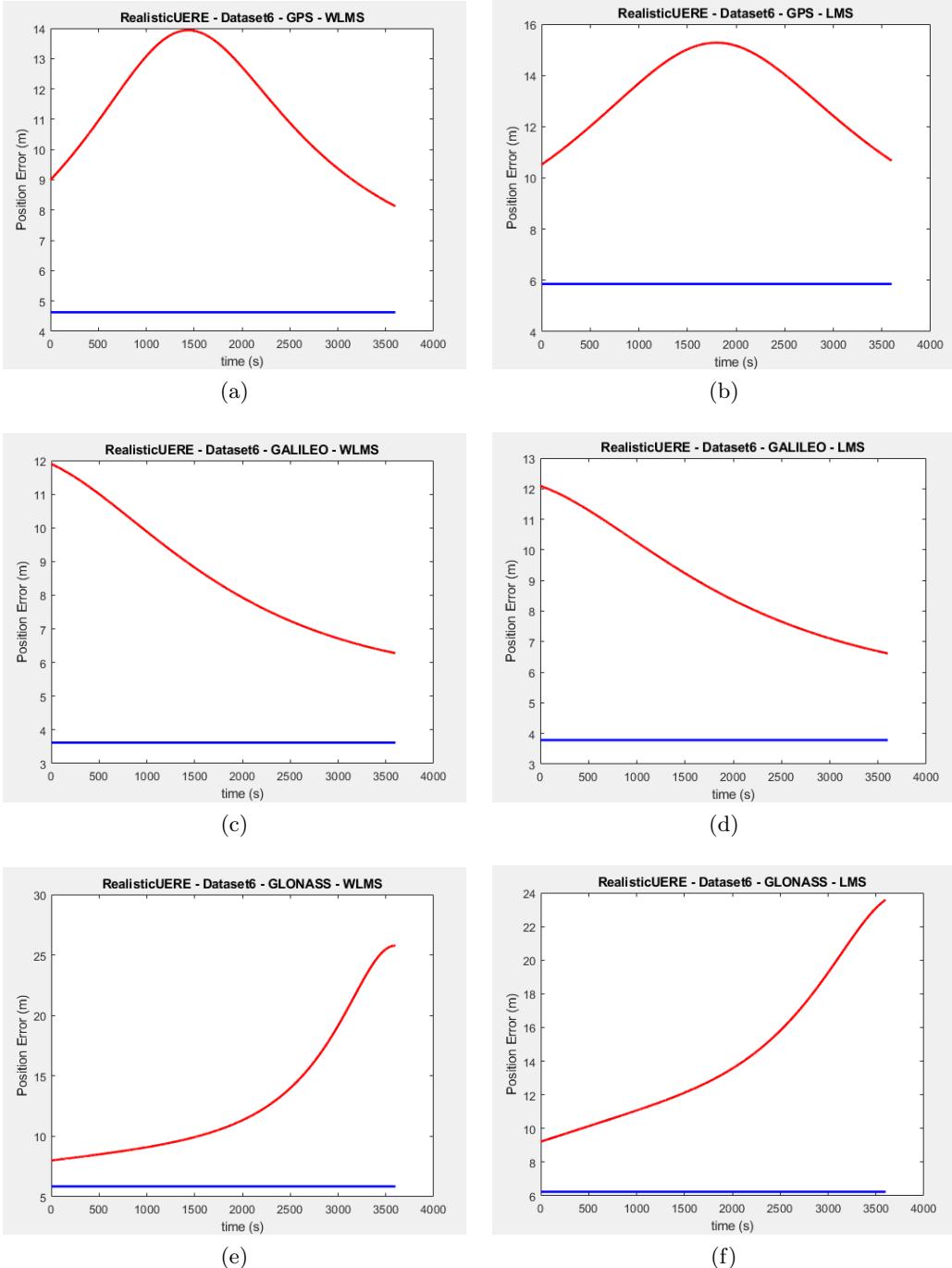
Figure 8 shows how the application of the WLMS algorithm can improve the PVT solution. In both datasets, this improvement is not graphically observable, since the position error decreases just of few meters. In both cases the results seem very similar. This is due to the fact that the constellations exploited were GPS for Turin and BeiDou for Shanghai. These constellations are optimized to be fully operational with these types of locations. In general, all the PVT solutions elaborated with WLMS present a more accurate position estimation, even though the algorithm is more complex from a computational point of view.

The standard deviation of the pseudorange error can be computed, for each satellite, according to the same methods described in Task A-3. This time the σ_{UERE} is no more a constant for all the satellites, thus, the theoretical standard deviation of the position error must be computed in a different way. From the covariance matrix of the pseudorange errors, we can obtain the covariance matrix of the position errors through equation 1.6.

$$\text{cov}(\delta \mathbf{x}) = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{R} \mathbf{W}^T \mathbf{H} (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \quad (1.6)$$

As a result, the position error will be given by equation 1.7.

$$\sigma_x = \sqrt{\text{tr}\{\text{cov}(\delta x)\}} \quad (1.7)$$



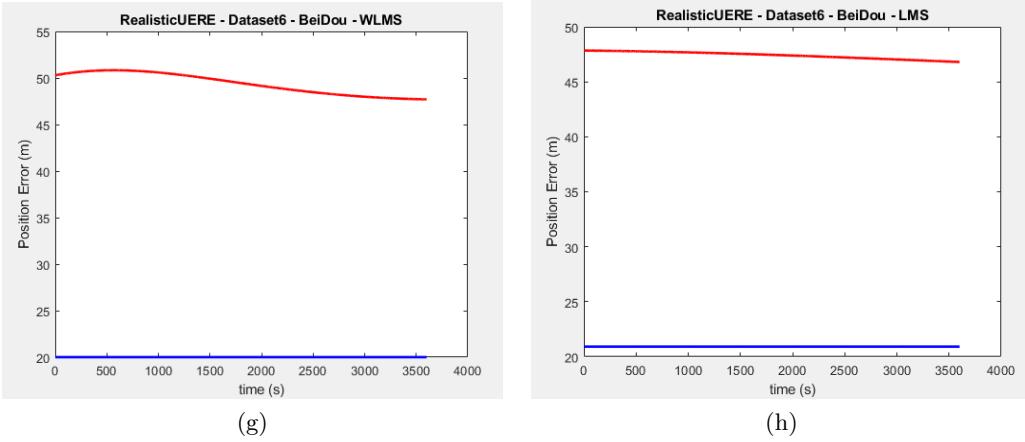


Figure 1.9: Comparison between position errors obtained with WLMS algorithm (on the left) and position errors obtained with recursive LMS algorithm (on the right). Each plot presents the ideal evaluation of the position error (in red) and the real one (in blue).

In figure 1.9 it can be seen that the position error evaluated through the WLMS is always smaller than the position error obtained through the LMS, even if the difference is not so evident. However, a decrease of few meters in the position error might mean a significant improvement in the quality of the service .

Chapter 2

Lab 2: Computation of PVT using real data

2.1 Introduction

In the second lab it is required to acquire the raw pseudoranges measurements through an Android app (GNSS logger) and to find the user state with them.

While in the first lab we worked with already corrected pseudoranges measurements, this time the values we have are still affected by all the possible contributions to the errors. The GPS ICD (Interface Control Documents) provides a general algorithm that is typically used to remove all the predictable contributions to the errors.

The raw measurements were collected at Politecnico di Torino, in three different types of locations. In the first one the GNSS service had to be entirely operative; in the second one the service had to be almost denied; in the third one the service had to be completely denied. The raw measurements were evaluated for five minutes for each data collection.

2.2 Task B-2

As already said, after the measure of the pseudoranges, these values have to be corrected of all the predictable contributions to the error. The PVT solution was evaluated through the MATLAB Google code that can be downloaded from: <https://github.com/google/gps-measurement-tools/tree/master/opensource>.

This script does not only provide an algorithm that removes the predictable contributions to the error of the pseudoranges, but also the PVT solution obtained through the WLMS (Weighted Least Square) algorithm. One of the outputs of this code represents the estimated positions along the time, together with the position error. An example is shown in figure 2.1.

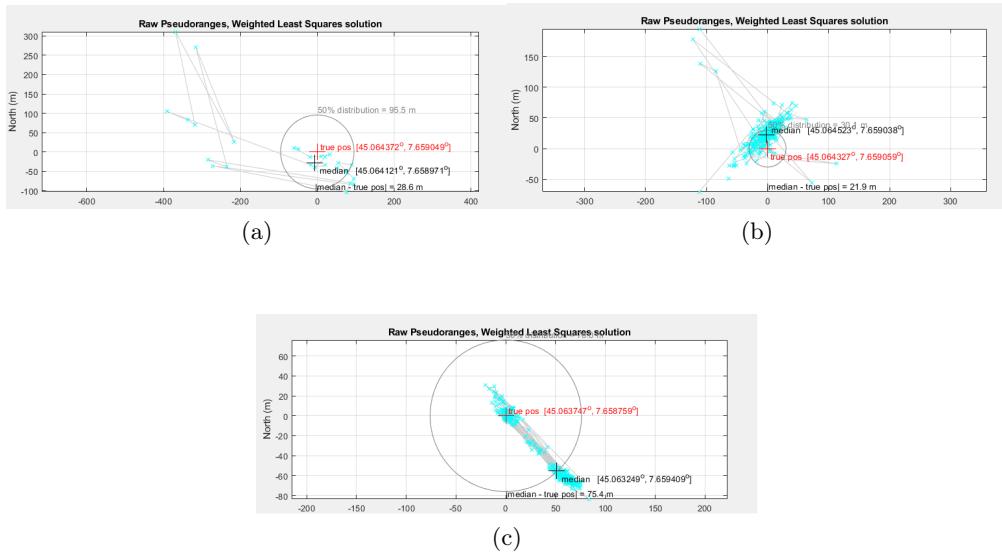


Figure 2.1: Comparisons between the true and the estimated positions for the three cases: Completely denied on the top left (a); almost denied on the top right (b); fully operative in the lower centre (c).

The three subplots in figure 1 show three different qualities of service. The performance of the positioning procedure is evaluated by the median among all the points and the distance between the median and the true position. Apparently, the case in which the service is denied gives a better performance, but there is to say that the number of user position estimations are very few and some of them are far away (more than 200m) from the real position. In the case in which the service is fully operative there are a lot of estimations, but many of them are located about 80m away from the true position.

This significant error is due to the location of the acquisition (near to “General Motors” building), where the multipath effect can not be neglected.

2.3 Task B-3

Another output of the script developed by Google is the evaluation of the pseudoranges of our acquisitions. They are shown in figure 2.2.

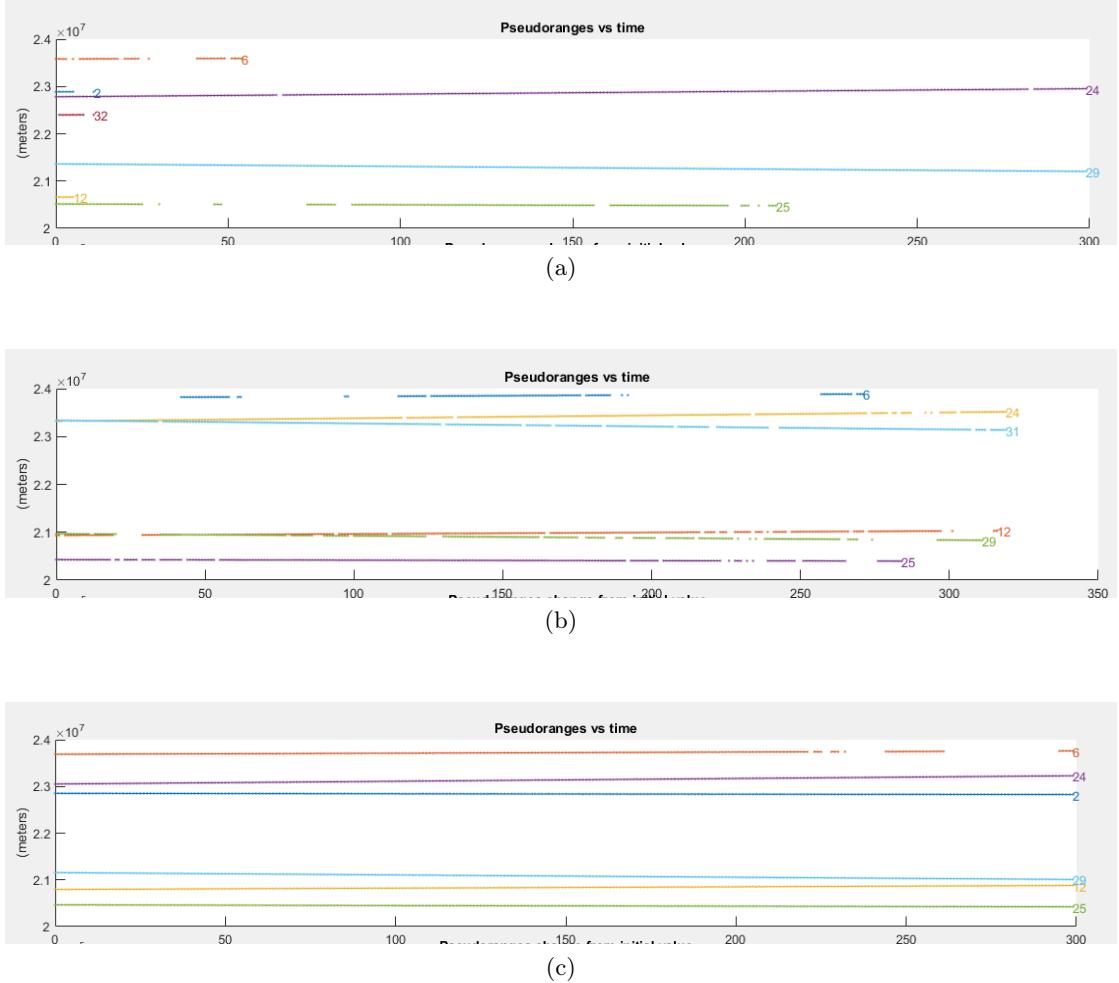


Figure 2.2: The evaluated pseudoranges vs time for three types of acquisition: completely denied (a), almost denied (b) and fully operative (c).

In figure 2 it is possible to observe how many satellites are visible at each type of acquisition. When the service is completely denied, only two satellites are always visible; this is the reason why the number of positions estimated in this case is very small. The number of satellites increases when the service is almost denied, but it is easy to notice an intermittent behaviour of their visibility. Finally, when the service is fully operative, the number of visible satellites, equal to six, is almost constant along the time, which means that a position can be evaluated at every time instant.

The script returns also the values of C/N_0 along the time, that is another measure to evaluate the performance of the positioning procedure. In the GNSS world, the signal-to-noise ratio (SNR) is computed with the ratio between the power of the signal and the power spectral density of the noise at the input of the receiver. This formulation is useful since it is independent from the receiver implementation.

The computed C/N_0 vs time are shown in figure 2.3.

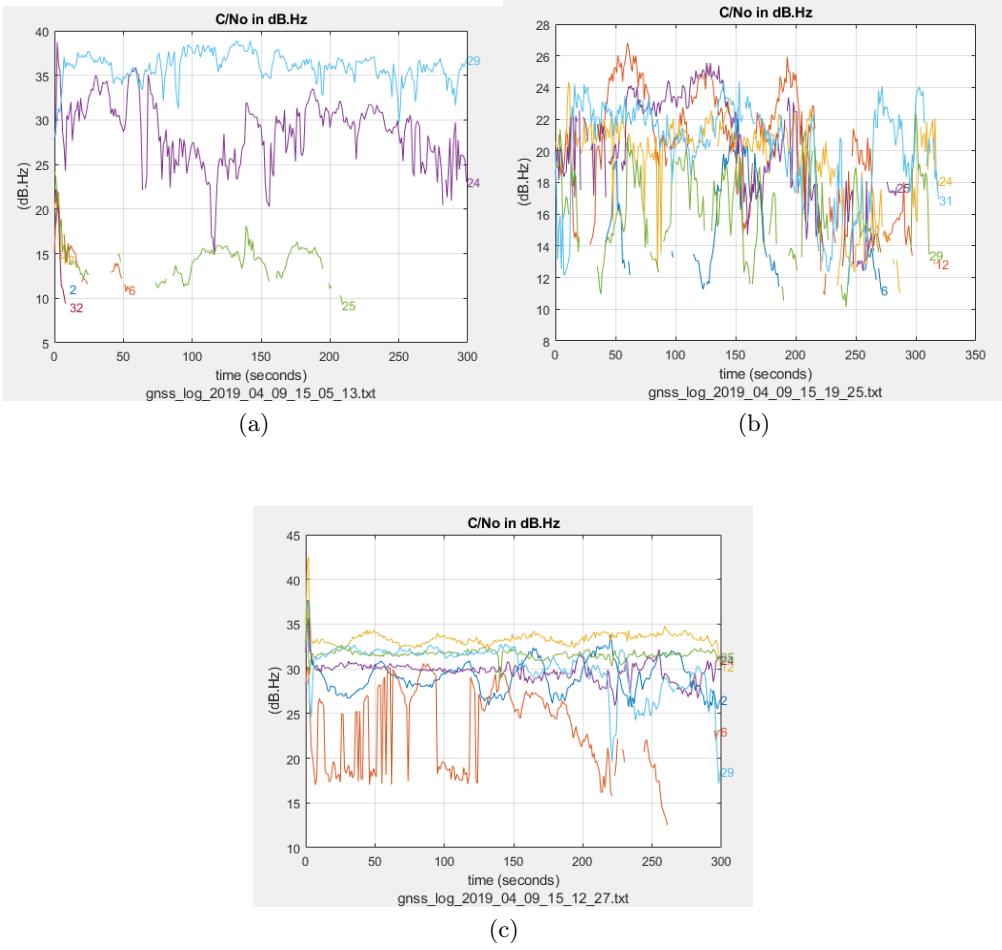


Figure 2.3: C/N_0 evaluated for the three different cases: Completely denied (a), almost denied (b) and fully operative (c).

In each subplot of figure 3, every curve associated with a different colour refers to the received signal from a different satellite. In the first two cases (completely and almost denied) all the signals show either large variations in the C/N_0 ratio or a complete absence of signal, which lead to a bad quality on the position estimations. In the case in which the service is fully operative, most of the signals show stable values in the C/N_0 ratio, with a good amplitude (about 30 dB-Hz on average). Signals from satellites number 6 and number 9 have the largest variations. This means that the multipath effect seen before is probably caused by the received signals from these two satellites.

2.4 Task B-4

As in task A-4 from the previous lab, given the values of the pseudoranges along the time, it is possible to evaluate the residual error contribution to the pseudorange measurements, σ_{UERE}^j , for each satellite. The direct calculation of the σ_{UERE}^j is not possible since multiple realizations of the pseudoranges for the same time instant are not available. However, supposing that the pseudorange measurements can be considered as an ergodic random process, we can remove their deterministic variation along the time taking the

second derivative of the process. According to this approach, we will obtain a new process with zero mean, whose standard deviation can be thought as an approximation of the σ_{UERE}^j .

In figure 2.4 three examples of this method are shown, one for each case (completely denied, almost denied and fully operative) associated with the same satellite.

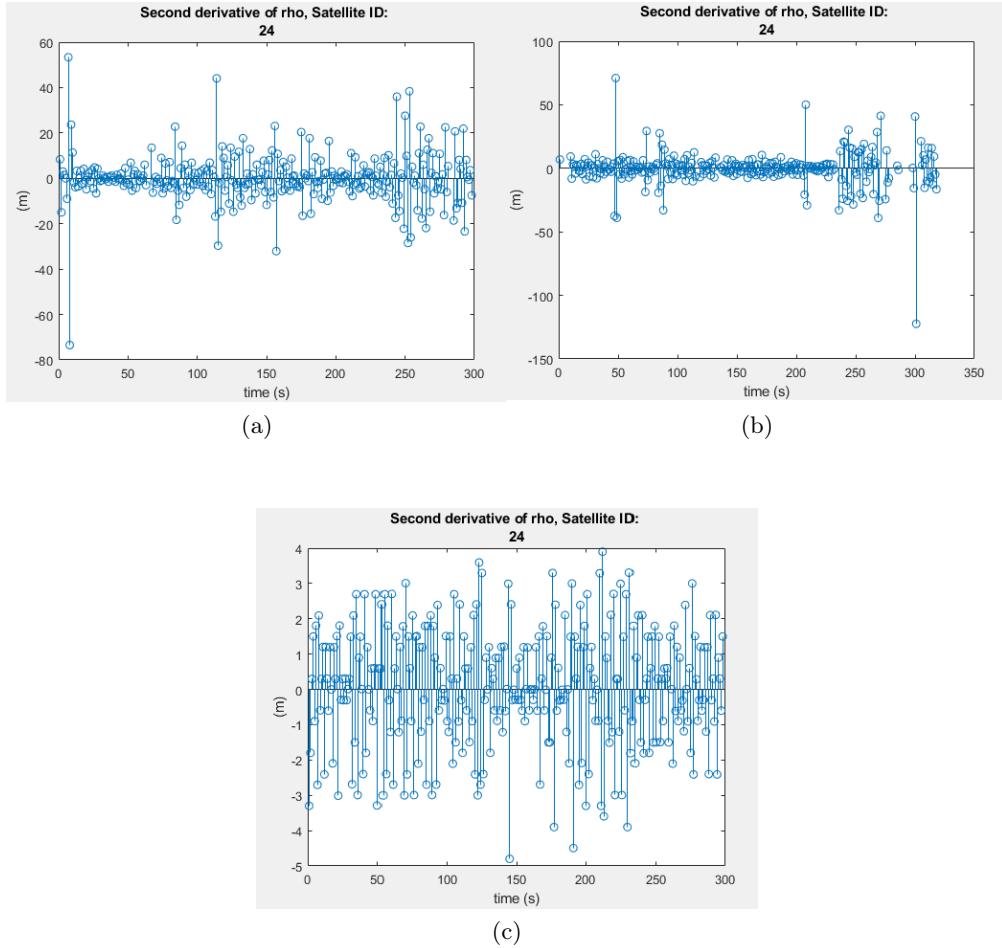


Figure 2.4: Second derivative applied to the measured pseudoranges. The same satellite (# 24) for each case: completely denied (a), almost denied (b) and fully operative (c).

At the end of this procedure, the σ_{UERE}^j can be evaluated for each case. The results are reported in Table 1.1.

Satellite ID	Case	σ_{UERE}^j
2	Fully Operative	4.508
6	Fully Operative	62.286
12	Fully Operative	5.0139
24	Fully Operative	1.703
25	Fully Operative	1.238
29	Fully Operative	3.907
6	Almost Denied	24.788
12	Almost Denied	15.324
24	Almost Denied	13.955
25	Almost Denied	14.591
29	Almost Denied	23.705
31	Almost Denied	16.655
2	Completely Denied	12.301
6	Completely Denied	35.526
12	Completely Denied	17.108
24	Completely Denied	10.994
25	Completely Denied	16.927
29	Completely Denied	4.317
32	Completely Denied	57.561

Table 2.1: σ_{UERE}^j evaluated for each visible satellite in all the three cases.

The residual error associated to the pseudorange is relatively small when the service is fully operative. As already seen in figure 3, the satellite that gives an abnormal uncertainty in this case is the number 6. The average σ_{UERE}^j tends to increase for almost and completely denied services. These results are perfectly consistent with the analysis of pseudoranges and C/N_0 discussed in Task B-3.

Bibliography

- [1] Teresa Ferreira. *GNSS Receivers General Introduction* — Navipedia. https://gssc.esa.int/navipedia/index.php/GNSS_Receiver_General_Introduction. 2011.
- [2] Wikipedia contributors. *Error analysis for the Global Positioning System* — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Error_analysis_for_the_Global_Positioning_System&oldid=887082067. [Online; accessed 20-May-2019]. 2019.