

ICT4SS - Lab Exercise #1, 02 April 2019

Lab Exercise #1: Evaluation of PVT

Laboratory on Least Mean Square positioning solution



POLITECNICO
DI TORINO
Dipartimento
di Elettronica
e Telecomunicazioni

Least Mean Square Solution Computation



- In the most general case the PVT solution is given by the LMS equation

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

- One possible way to solve for the PVT is to use the LMS algorithm not applied a single time, but iterated in order to provide a more precise solution
- At each iteration, the estimated position is expected to be closer to the real user location

Recursive Least Mean Square

- For each time instant n , a number of iterations $k = 1, \dots, K$ is performed for the same measurements vector $\boldsymbol{\rho}_n$

INITIAL POSITION/APPROXIMATION POINT	
Initialization	$\hat{\mathbf{x}}_n^0$
Procedure for k^{th} iteration	$\Delta \hat{\boldsymbol{\rho}}_n^k = \hat{\boldsymbol{\rho}}_n^k - \boldsymbol{\rho}_n$
	$\Delta \hat{\mathbf{x}}_n^k = ((\mathbf{H}_n^k)^T \mathbf{H}_n^k)^{-1} (\mathbf{H}_n^k)^T \Delta \hat{\boldsymbol{\rho}}_n^k$
	$\hat{\mathbf{x}}_n^k = \hat{\mathbf{x}}_n^{k-1} + \Delta \hat{\mathbf{x}}_n^k$
RANGE VECTOR GEOMETRICALLY EVALUATED FROM $\hat{\mathbf{x}}_n^0$ AND EPHEMERIS	
VECTOR OF MEASURED PSEUDORANGES	

- \mathbf{H} is updated at each iteration k
- The initial approximation point, $\hat{\mathbf{x}}_n^0$, could be the same at each time n
- K has to be chosen in order to have “stable” solution (generally $K < 10$)

Recursive Least Mean Square

- $\hat{\mathbf{x}}_n^k = [\hat{x}_n^k \quad \hat{y}_n^k \quad \hat{z}_n^k \quad \hat{b}_n^k]$ is the estimated position/state
- The number of visible satellites J is not constant over observation time n
- $\Delta \hat{\rho}_n$ is the vector of the J_n pseudorange differences (between measured and calculated) at the k -th iteration
- At each iteration k , \mathbf{H}_n^k is the geometrical matrix obtained from the previous estimated position:

Satellite coordinates are fixed for any k

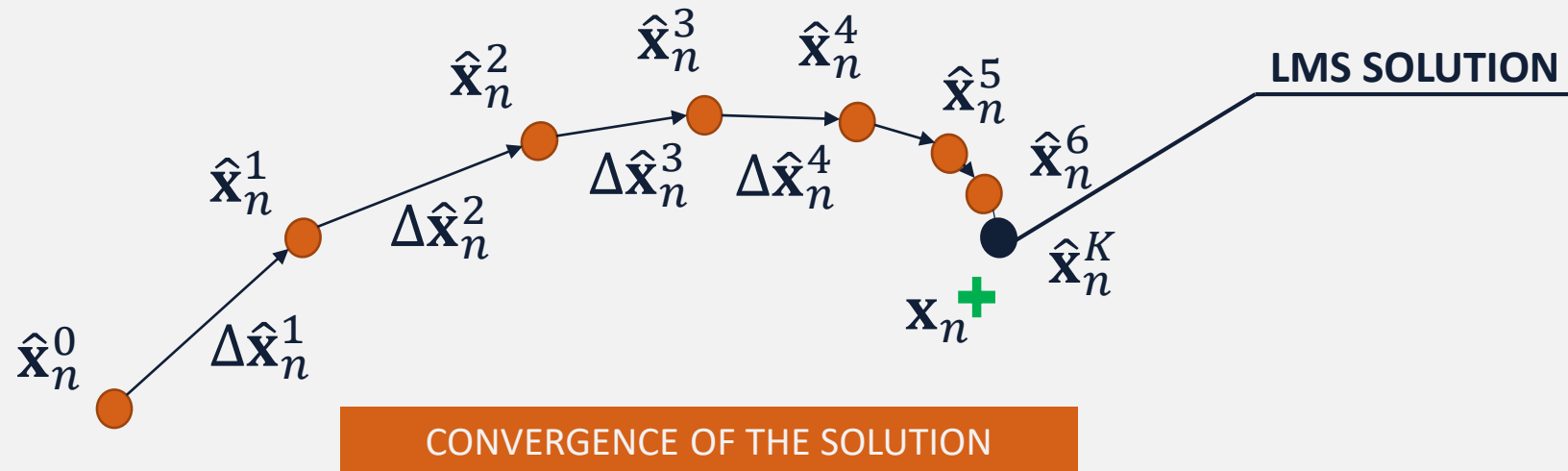
$$\mathbf{H}_n^k = \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 \\ a_{x,2} & a_{y,2} & a_{z,2} & 1 \\ a_{x,3} & a_{y,3} & a_{z,3} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,J_n} & a_{y,J_n} & a_{z,J_n} & 1 \end{bmatrix} \quad a_{x,j} = \left[\frac{x_{j,n} - \hat{x}_n^k}{\hat{r}_{j,n}} \right], \quad a_{y,j} = \dots, \quad a_{z,j} = \dots$$

$$\hat{r}_{j,n} = \sqrt{(x_{j,n} - \hat{x}_n^k)^2 + (y_{j,n} - \hat{y}_n^k)^2 + (z_{j,n} - \hat{z}_n^k)^2}$$

Recursive Least Mean Square

- Adding $\Delta\hat{\mathbf{x}}_n^k$ to the estimated position at the previous iteration, a new estimation of the position is obtained

$$\hat{\mathbf{x}}_n^k = \hat{\mathbf{x}}_n^{k-1} + \Delta\hat{\mathbf{x}}_n^k$$



From raw to corrected pseudorange



- The measurement of the pseudoranges must be corrected of all the predictable contributions to the errors
- After the correction there will be still a random contribution that is modeled by the User Equivalent Range Error (UERE)

$$UERE \sim \mathcal{N}(0, \sigma_{UERE})$$

Different errors for different satellites

Basic LMS algorithm attributes the same relevance to all the measurements but actually pseudoranges are not equally precise.

- In real cases, each pseudorange may be characterized by a different value of standard deviation, $\sigma_{j, \text{UERE}}$
- They can still be considered uncorrelated such that:

$$\mathbf{R} = \text{Cov}(\boldsymbol{\rho}) = \begin{bmatrix} \sigma_{1, \text{UERE}}^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_{2, \text{UERE}}^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_{3, \text{UERE}}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_{J, \text{UERE}}^2 \end{bmatrix}$$

Different errors for different satellites

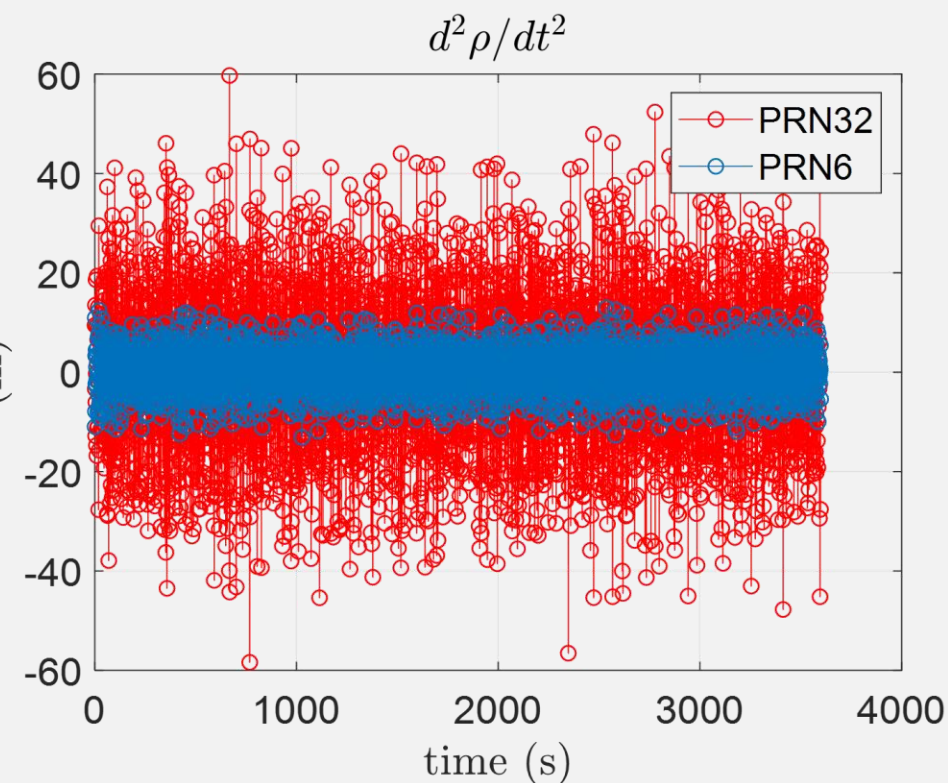
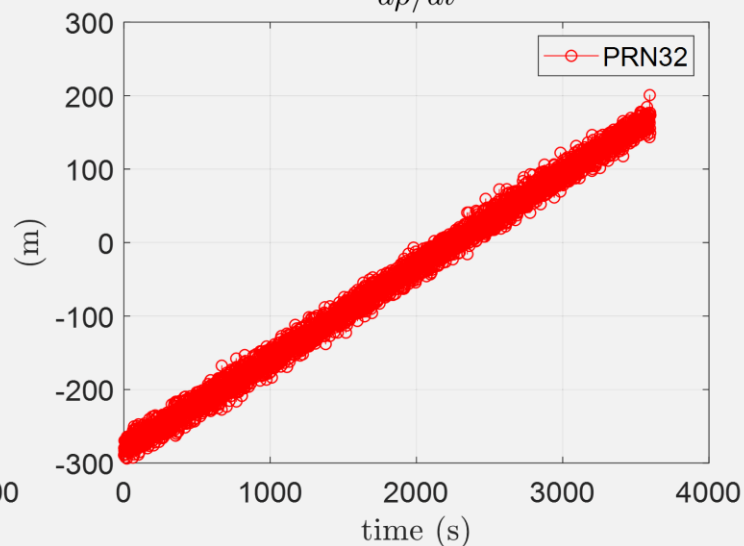
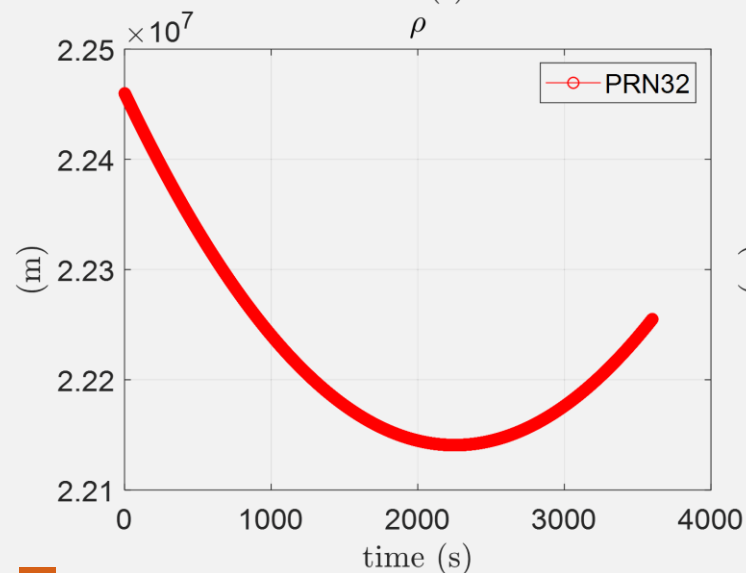
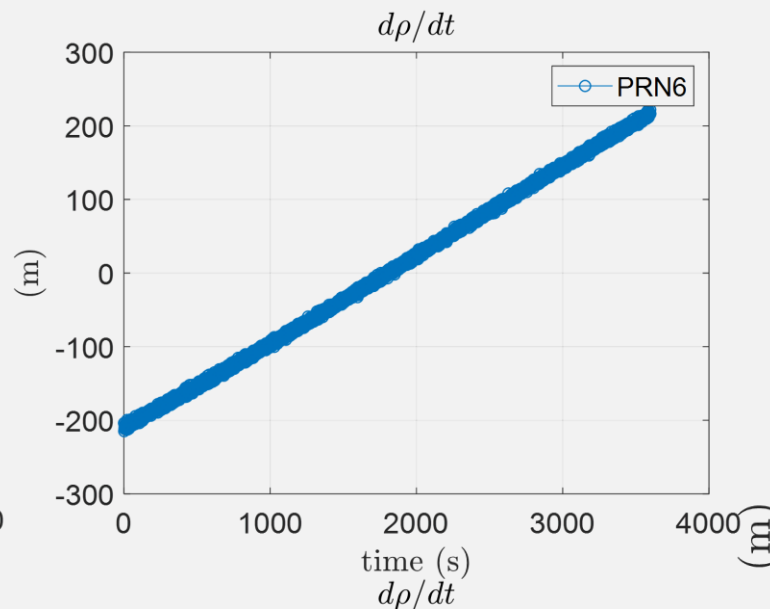
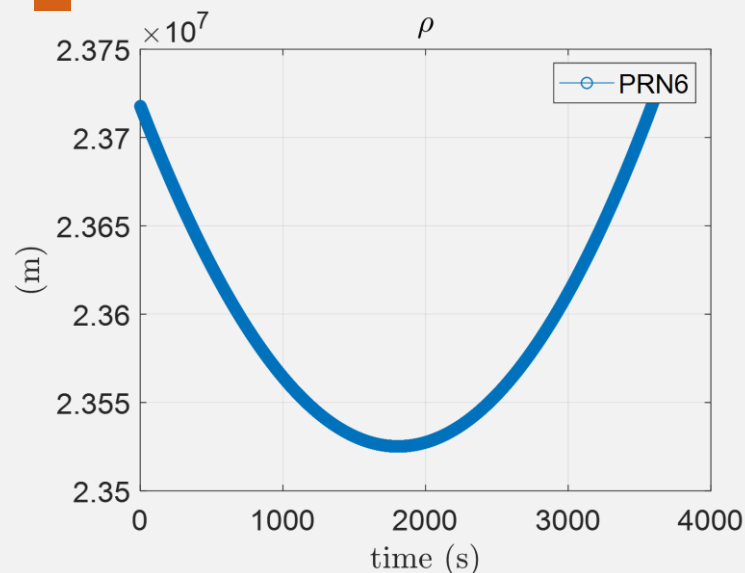


- The direct calculation of the covariance matrix is not possible since the $\rho_{j,n}$ are time dependent (multiple realizations for the same instant n are not available)
- We can estimate the $\sigma_{j, \text{UERE}}$ for each pseudorange ρ_j analyzing the error on the pseudorange itself along the time (supposing it is ergodic)
- Dependency on time must be removed
- Since pseudorange measurement follows a quadratic trend, the second order derivative can be applied to remove the variation along the time



Use MATLAB function `diff(X, order)`

Pseudorange de-trending by differentiation



Weighted Least Mean Square



- A second approach foresees to solve the system by means of a **Weighted Least Squares** (WLS) algorithm, characterized by the introduction of the weighting matrix **W**, which is a positive definite matrix
- Since some measurements may be known to be more accurate than others, the measurement accuracy is known to be characterized by the inverse of the measurement errors covariance matrix **R**
- It is natural to select $\mathbf{W} = \mathbf{R}^{-1}$ to give the least weighting to the most uncertain measurements.

Uncorrelated measurements

- The weight matrix can be estimated from the measurements, thus designing a new **weighted geometrical matrix** $\bar{\mathbf{H}}_n^k$
- For each time instant n , a number of iterations $k = 1, \dots, K$ is performed for the same measurements vector $\boldsymbol{\rho}_n$

Initialization	$\hat{\mathbf{x}}_n^0$
Procedure for k^{th} iteration	$\Delta \hat{\boldsymbol{\rho}}_n^k = \hat{\boldsymbol{\rho}}_n^k - \boldsymbol{\rho}_n$
	$\bar{\mathbf{H}}_n^k = \left((\mathbf{H}_n^k)^T \mathbf{W} \mathbf{H}_n^k \right)^{-1} (\mathbf{H}_n^k)^T \mathbf{W}$
	$\Delta \hat{\mathbf{x}}_n^k = \bar{\mathbf{H}}_n^k \Delta \hat{\boldsymbol{\rho}}_n^k$
	$\hat{\mathbf{x}}_n^k = \hat{\mathbf{x}}_n^{k-1} + \Delta \hat{\mathbf{x}}_n^k$

ICT4SS - Lab Exercise #1, 02 April 2019

Lab Exercise #1: Evaluation of PVT

Laboratory on Least Mean Square positioning solution



POLITECNICO
DI TORINO
Dipartimento
di Elettronica
e Telecomunicazioni

Variables for GPS constellation



USER and SATELLITE STATE VECTORS (position and clock bias)

Unknown User Position

$$\mathbf{x}_n = [x_n \quad y_n \quad z_n \quad b_{n,GPS}]$$

Known Satellite Position

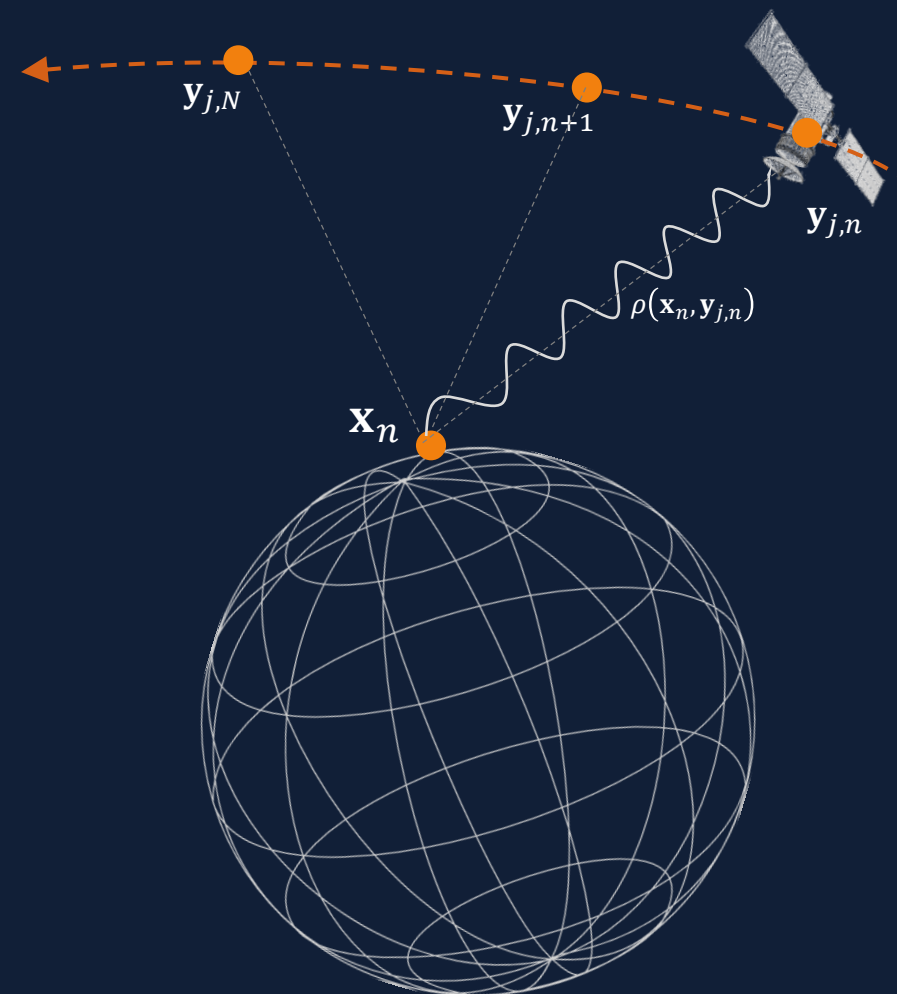
$$\mathbf{y}_{j,n} = [x_{j,n} \quad y_{j,n} \quad z_{j,n}]$$

PSEUDORANGE EQUATION

Measured range distance from the satellite

$$\rho(\mathbf{x}_n, \mathbf{y}_{j,n}) = \sqrt{(x_n - x_{j,n})^2 + (y_n - y_{j,n})^2 + (z_n - z_{j,n})^2} + b_{n,GPS}$$

- $n \in (1, 2, \dots, N)$ is the time index
- $j \in (1, 2, \dots, J)$ is the satellite identifier
- Data collections include 3600 seconds of satellites observations from a static position \mathbf{x}_n : $\mathbf{x}_n = \mathbf{x}_{n+1} = \dots = \mathbf{x}_N$, where $N = 3600$



Observables Data Structures



PSEUDORANGE MEASUREMENTS (GPS)

RHO.GPS				
	$n = 1$	$n = 2$..	$n = N$
GPS.PRN_1	$\rho(\mathbf{x}_1, \mathbf{y}_{1,1})$	$\rho(\mathbf{x}_2, \mathbf{y}_{1,2})$...	$\rho(\mathbf{x}_N, \mathbf{y}_{1,N})$
GPS.PRN_2	$\rho(\mathbf{x}_1, \mathbf{y}_{2,1})$	$\rho(\mathbf{x}_2, \mathbf{y}_{2,2})$...	$\rho(\mathbf{x}_N, \mathbf{y}_{2,N})$
GPS.PRN_3	$\rho(\mathbf{x}_1, \mathbf{y}_{3,1})$	$\rho(\mathbf{x}_2, \mathbf{y}_{3,2})$...	$\rho(\mathbf{x}_N, \mathbf{y}_{3,N})$
...
GPS.PRN_J	$\rho(\mathbf{x}_1, \mathbf{y}_{J,1})$	$\rho(\mathbf{x}_2, \mathbf{y}_{J,2})$...	$\rho(\mathbf{x}_N, \mathbf{y}_{J,N})$

SATELLITE POSITIONS FROM EPHEMERIS (GPS)

SV_ECEF.GPS		Ex: SV Position State Vector History			
SV ID	\mathbf{Y}_j	\mathbf{Y}_1	x	y	z
GPS.PRN_1	\mathbf{Y}_1	$n = 1$	$x_{1,1}$	$y_{1,1}$	$z_{1,1}$
GPS.PRN_2	\mathbf{Y}_2	$n = 2$	$x_{1,2}$	$y_{1,2}$	$z_{1,2}$
GPS.PRN_3	\mathbf{Y}_3	$n = 3$	$x_{1,3}$	$y_{1,3}$	$z_{1,3}$
...
GPS.PRN_J	\mathbf{Y}_J	$n = N$	$x_{1,N}$	$y_{1,N}$	$z_{1,N}$

- The number of visible satellites is not constant over observation time n
- The folder named «NominalUERE» contains pseudorange measurements with the same σ_{UERE}
- The folder named «RealisticUERE» contains pseudorange measurements with satellite-dependent σ_{UERE}
- Pseudorange measurements can be considered an ergodic random process over short time periods
- It is possible to select different constellations using strings GPS, GLO, BEI, GAL in the data structure

Lab 1 session A



TASK

A-1

Load data from the *NominalUERE* folder. Check the satellites visibility at each time instant n for all the constellations. Plot the number of the visible satellites as well as the measured pseudoranges along the time.

TASK

A-2

Choose a dataset and a constellation (e.g. GPS, Galileo) and develop a Least Mean Square (LMS) positioning algorithm and estimate the user state, $\hat{\mathbf{x}}_n^K$, at each time instant n .
Verify the position on Google Earth: convert from ECEF to LLA and use *writeKML_GoogleEarth.m*.

TASK

A-3

Compute the standard deviation of the position error over time $n = 1, 2, \dots, N$ and compare the quality of the obtained solution for different datasets and constellations. Motivate the results.

TASK

A-4

Estimate the $\sigma_{j, \text{UERE}}$ for all satellites using the dataset from the *realisticUERE* folder. Implement the **Weigthed Least Mean Square (WLMS)** repeating Tasks 2 and 3.
Compare the performance of LMS and WLMS on *realisticUERE* dataset.

Lab 1 session B



TASK

B-1

Acquire raw pseudorange measurements through the app GNSS logger for Android Smartphone and log them as a txt file (ex. gnss_log_2018_10_19_11_15_47.txt)

TASK

B-2

Compute the positioning solution running the MATLAB Google code you find at <https://github.com/google/gps-measurement-tools/tree/master/opensource>

TASK

B-3

Comment on the outputs for the pseudorange and C/No measurements

TASK

B-4

Estimate $\sigma_{j, \text{UERE}}$ as in task A-4

OPTIONAL TASKS



TASK

B-5

Replace the weighting strategy of the Google PVT with the one developed for the **Weigthed Least Mean Square (WLMS)** in Task A-4

TASK

B-6

Replace the PVT computation with your own code developed in the session A

Simulation hint



For each epoch $n = 1:N$

- Find the available pseudorange measurements
- Build the measurement vector $\boldsymbol{\rho}_n = [\rho_{1,n} \quad \cdots \quad \rho_{J,n}]$
- Retrieve the corresponding satellites coordinates $\mathbf{y}_{j,n}$
- Compute PVT solution (K iterations)

end

How to use refMap

- Install the Mapping toolbox
- `refMap(mean(latitude), mean(longitude), 'Here I am!');`

